



SATBAYEV UNIVERSITY

УДК: 004.9:616(043)

На правах рукописи

МУКАШЕВА АСЕЛЬ КОПТЛЕУВНА

**Исследование и разработка информационной системы диагностики
сахарного диабета на базе инструментов BigData технологий**

6D070300 – Информационные системы

Диссертация на соискание степени
доктора философии (PhD), доктора по профилю

Научный консультант:
доктор PhD, ассоц. профессор
Сапарходжаев Н.П.

Зарубежный научный консультант:
доктор PhD, профессор Amy W. Aron

Республика Казахстан
Алматы, 2020

СОДЕРЖАНИЕ

НОРМАТИВНЫЕ ССЫЛКИ	4
ОПРЕДЕЛЕНИЯ	5
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	7
ВВЕДЕНИЕ	8
1 АКТУАЛЬНЫЕ ПРОБЛЕМЫ ДИАГНОСТИКИ И ПРОГНОЗИРОВАНИЯ САХАРНОГО ДИАБЕТА В СОВРЕМЕННОМ МИРЕ	13
1.1 Исследование распространенности сахарного диабета в мире	13
1.2 Сравнительный анализ современных медицинских информационных систем диагностики	21
1.3 Исследование источников и характеристик информационных ресурсов по сахарному диабету	25
1.4 Постановка цели и задачи исследования	26
Выводы по первому разделу	26
2 ИССЛЕДОВАНИЕ МЕТОДОВ ПРИМЕНЕНИЯ ТЕХНОЛОГИИ BIGDATA В СФЕРЕ МИРОВОГО ЗДРАВООХРАНЕНИЯ	27
2.1 Аналитика применения обработки больших данных во всех направлениях	27
2.2 Методы и алгоритмы, используемые в области аналитики медицинских данных	28
2.3 Применение информационных технологии для решения медицинских задач	32
Выводы по второму разделу	34
3 МОДЕЛИ И МЕТОДЫ ИССЛЕДОВАНИЯ ПРОГНОЗИРОВАНИЯ РОСТА ПАЦИЕНТОВ И АНАЛИЗ ДАННЫХ	35
3.1 Модели прогнозирования статистических данных на базе методов регрессионного анализа	35
3.2 Получение модели регрессионного анализа на основе алгоритмов машинного обучения	48
3.3 Анализ, обработка и визуализация больших данных	50
3.4 Методы поиска ключевых данных для обработки медицинских данных	64
Выводы по третьему разделу	68
4 РАЗРАБОТКА МОДЕЛЕЙ ДИАГНОСТИКИ СД С ПОМОЩЬЮ ИНСТРУМЕНТОВ BIGDATA	70
4.1 Математическая модель обработки и поиска больших данных	70
4.2 Разработка математической модели диагностирования сахарного диабета с помощью больших данных	74
4.3 Выбор платформы на базе инструментов BigData технологии	77
4.4 Конфигурирование и запуск компонентов системы Hadoop	80
4.5 Администрирование и обработка данных в среде Hadoop	83

4.5.1	Загрузка и обработка данных с помощью MapReduce	83
4.5.2	Администрирование приложений с помощью Spark	92
4.6	Программная реализация информационной системы	97
4.7	Создание и загрузка неструктурированных данных в MongoDB	99
	Выводы по четвертому разделу	104
	ЗАКЛЮЧЕНИЕ	105
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	108
	ПРИЛОЖЕНИЕ А – Разработка поисковой системы для диагностирования болезни	118
	ПРИЛОЖЕНИЕ Б – Разработка интерфейса ИС	123
	ПРИЛОЖЕНИЕ В – Процесс загрузки данных в MongoDB	131
	ПРИЛОЖЕНИЕ Г – Листинг программы	136
	ПРИЛОЖЕНИЕ Д – Акт внедрения	201
	ПРИЛОЖЕНИЕ Е – Авторское свидетельство	203

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей диссертации использованы ссылки на следующие стандарты:

«Инструкция по оформлению диссертации и автореферата», ВАК МОН РК, №377-3ж.

ГОСТ 7.1–2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.

ГОСТ 7.32–2017 – Система стандартов по информации, библиотечному и издательскому делу. «Отчет о научно-исследовательской работе. Структура и правила оформления».

ОПРЕДЕЛЕНИЯ

В настоящей диссертации используются следующие термины с соответствующими определениями:

База данных – систематически организованное или структурированное хранилище индексированной информации, позволяющее легко получать, обновлять, анализировать и выводить данные.

Большие данные – наборы данных, используемые для обозначения огромного объема данных, которые трудно обработать с использованием традиционных методов работы с базами данных и программного обеспечения. Также данная технология использует массивный параллелизм с помощью вычислительной машины.

Доказательная медицина – это подход к медицинской практике, предназначенный для оптимизации процесса принятия решений путем акцентирования использования доказательств из хорошо спланированных и хорошо проведенных исследований.

Извлечение информации – данные анализируются и просматриваются для извлечения соответствующей информации из источников данных по определенному шаблону. Большая часть извлечения данных происходит из неструктурированных источников данных и различных форматов данных.

Информационный поиск – алгоритм, который используется для извлечения информации, хранящейся в некоторой структуре данных или вычисленной в пространстве поиска проблемной области, либо с дискретными или с непрерывными значениями.

Кластер – представляет собой группу серверов и других ресурсов, которые действуют как единая система и обеспечивают высокую доступность и, в некоторых случаях, распределение нагрузки и параллельную обработку.

Корреляционный анализ – статистический инструмент, используемый для изучения близости отношений между двумя или более переменными.

Машинное обучение – категория алгоритмов, которая позволяет программным приложениям быть более точными в прогнозировании результатов без явного программирования. Основная предпосылка машинного обучения заключается в создании алгоритмов, которые могут принимать входные данные и использовать статистический анализ для прогнозирования выходных данных при обновлении выходных данных по мере появления новых данных.

Медицинская экспертная система – компьютерная программа, которая, обеспечивает поддержку принятия решений в виде точной диагностической информации или предлагает лечение или прогноз.

Парсер – задача программирования, которая разбивает заданную последовательность символов или значений на более мелкие части на основе некоторых правил. Используется в самых разных приложениях - от простого анализа файлов до крупномасштабной обработки естественного языка.

Распределенные вычисления – вычисления на распределенных автономных компьютерах, которые обмениваются данными только по сети.

Регрессионный анализ – статистический метод для исследования оценки взаимосвязи между зависимых и набором независимых объясняющих переменных. Также используется в качестве общего термина для различных методов анализа данных, которые используются в качественном методе исследования для моделирования и анализа многочисленных переменных. В методе регрессии зависимая переменная является предиктором или пояснительным элементом, а независимая переменная является результатом или ответом на конкретный запрос.

Сахарный диабет – представляет собой группу метаболических нарушений, характеризующихся высоким уровнем сахара в крови в течение длительного периода.

Эпидемиология – метод, используемый для поиска причин последствий для здоровья и заболеваний среди населения. Эпидемиологическая информация используется для планирования и оценки стратегий по предотвращению заболеваний и в качестве руководства для ведения пациентов, у которых заболевание уже развилось.

Эндокринология – изучение медицинских аспектов гормонов, в том числе заболеваний и состояний, связанных с гормональным дисбалансом, повреждением желез, вырабатывающих гормоны, или использованием синтетических или природных гормональных препаратов.

Экспертная система – интерактивная и надежная компьютерная система принятия решений, которая использует как факты, так и эвристики для решения сложных проблем принятия решений. Компьютерное приложение, которое решает самые сложные проблемы в конкретной области.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД	большие данные
ВОЗ	Всемирная организация здравоохранения
ВРП	Валовой региональный продукт
ВВП	Валовой внутренний продукт
ГОБМП	гарантированный объем бесплатной медицинской помощи
ИС	информационная система
ИТ	информационные технологий
ИУЖ	Идентификаторы уровня жизни
МДФ	Международная федерация диабета
МНК	Метод наименьших квадратов
МЗРК	Министерства здравоохранения Республики Казахстан
МЗСРК	Министерство здравоохранения и социального развития РК
МЭС	Медицинская экспертная система
НИЗ	неинфекционные заболевания
ООН	Организация Объединённых Наций
ОЭСР	Организации экономического сотрудничества и развития
РК	Республика Казахстан
СД	сахарный диабет
API	Application programming interface (Интерфейс прикладного программирования)
IBM	International Business Machines (Международные Бизнес Машины)
MSKCC	Memorial Sloan Kettering Cancer Center (Мемориальный центр Слоана Кеттеринга)
HDP	Hortonworks Data Platform (Платформа данных Hortonworks)
YDA	Youth Diabetes Action (Молодежное Диабетическое Действие)

ВВЕДЕНИЕ

Оценка современного состояния решаемой научно-технологической задачи. Первый Глобальный доклад Всемирной организацией здравоохранения (ВОЗ) по диабету демонстрирует, что число взрослых, живущих с диабетом, с 1980 года увеличилось почти в четыре раза до 422 миллионов взрослых. В 2012 году было зафиксировано, что от диабета умерло 1,5 миллиона человек. Осложнения НИЗ могут привести к инфаркту, инсульту, слепоте, почечной недостаточности и ампутации нижних конечностей [1]. Сахарный диабет (СД) признан самым быстрорастущим хроническим неинфекционным заболеванием (НИЗ) в мире. Общее количество людей с диабетом 2 типа растет в каждой стране [2].

Уровень заболеваемости СД в мире за последние два десятилетия вырос почти вдвое, т. е. ожирение и проблемы со здоровьем, которые он порождает, с каждым днем увеличивается распространение в развивающихся странах [3].

По официальной статистике ВОЗ только в 2014 году 8,5% взрослых в возрасте 18 лет и старше страдали диабетом. Также в 2016 году диабет стал прямой причиной 1,6 миллиона случаев смерти, а в 2012 году высокий уровень глюкозы в крови стал причиной еще 2,2 миллиона случаев смерти [4]. Только в 2016 году по официальным данным статистики ВОЗ в результате НИЗ произошло 41 миллион случаев смерти, что составляет 71% от общего числа 57 миллионов случаев смерти. Большинство таких летальных исходов были вызваны четырьмя основными НИЗ: сердечно-сосудистыми заболеваниями, 17.9 млн. летальных исходов, т.е. на них приходится 44% всех случаев НИЗ; рак, 9.0 миллиона летальных исходов, т.е. на них приходится 22% всех случаев НИЗ; хроническое респираторное заболевание, 3.8 миллиона смертей; 9%; диабет, 1.6 миллиона смертей; 4% [5].

Можно отметить тот факт, что чем больше улучшается благосостояние граждан и человеческий труд заменяется новыми изобретениями и технологиями, тем комфортнее становится жизнь человека. Это все являются предпосылками для развития СД второго типа, т.е. люди стали страдать от избыточного веса и ожирения. С 1980 года число людей, живущих с диабетом, увеличилось почти в четыре раза, достигнув отметки в 422 миллионов взрослых, причем большинство из них живут в развивающихся странах [6].

Основание и исходные данные для разработки темы. Основанием для разработки данной темы является выписка из протокола заседания кафедры «Информационные технологии» №3 от 30 сентября 2016 г. об утверждениях научных руководителей и тем докторских диссертации PhD докторантов специальности 6D070300 – Информационные Системы. Исходными данными для выполнения исследования являются статьи, доклады, учебники и монографии.

Обоснование необходимости проведения научно-исследовательской работы. В РК СД входит в число социально-значимых заболеваний, тем самым обуславливается высокий уровень распространенности данного заболевания среди населения, а также высокие затраты на диагностику и лечения СД. Важным направлением развития медицинского обслуживания населения становится разработка и внедрение различных проблемно-ориентированных информационных систем (ИС). Разрабатываемые ранее подобные системы, не охватывали весь объем разнородной информации, которая собирается при диагностировании и назначении хода лечения СД, а также они не использовали в качестве инструментария BigData технологии и облачные сервисы.

Сведения о планируемом научно-техническом уровне разработки, патентных исследованиях и выводы из них. В процессе выполнения работы был проведен анализ литературных данных и патентные исследования по проблеме информационных технологий (ИТ) диагностики СД и прогнозирования роста пациентов СД по РК.

Научно-технический уровень диссертационной работы обеспечен новизной и адекватностью полученных результатов, их практической значимостью и перспективностью использования. В результате проведенных исследований разработан метод прогнозирования роста пациентов с помощью регрессионного анализа, а также разработана платформа для создания ИС поддержки врача эндокринолога.

Сведения о метрологическом обеспечении диссертации. Все экспериментальные исследования, разработка методик и алгоритмов, решение расчетных задач были реализованы с использованием программных языков для разработки приложений, таких как, Python, Node.js, PHP, MongoDB, Hadoop, MapReduce, Spark, Elasticsearch, NoSQLBooster и Java. Для проверки адекватности результатов прогнозирования был проведен сравнительный анализ полученных результатов с помощью библиотеки scikit-learn языка Python с программным продуктом Excel компании Microsoft. Также во время исследования были использованы программные пакеты технологии BigData. Для апробации ИС была выбрана функционирующая клиника на базе МКТУ им. Х. А. Ясави, где ИС прошла тестирование для проверки работоспособности данной системы.

Актуальность темы диссертационного исследования. Распространенность СД в Казахстане носит характер эпидемии и приводит к большим финансовым затратам. Проведенный анализ современного состояния исследований по прогнозированию роста пациентов показал недостаточную проработанность в данной предметной области. Диагностирование и лечение диабета в настоящее время является одной из важных задач в медицине на современном этапе развития медицинских услуг. Разработанные методы и модели позволяют обоснованно улучшить диагностику и лечение диабета. С учетом высокой стоимости лечения диабета актуальным является прогнозирование затрат на основе

моделирования технико-экономических показателей в кластере лечения диабета. Использование реальных статистических данных, с помощью которого можно выявить наиболее эффективный метод для прогнозирования роста пациентов СД на фоне пассивного выявления для определения количества закупаемого инсулина по регионам. Объем спрогнозированного количества инсулина в свою очередь позволит запланировать финансовые расходы во избежание лишних затрат.

Актуальность подчёркивается отсутствием проблемно-ориентированных ИТ диагностики и выработки рекомендаций по лечению СД. В связи с этим потребность в исследовании и разработке ИС поддержки врача-эндокринолога по диагностике СД на базе инструментов BigData технологии высока. Применение современных ИТ в системе здравоохранения позволит улучшить качество оказываемых медицинских услуг с помощью технологий для проведения анализа, обработки и визуализации больших данных. Поэтому исследование этих вопросов определяет актуальность темы данной диссертационной работы.

Научная новизна темы исследования заключается в разработке структуры ИС для диагностики СД на основе BigData технологий и разработке системы моделей процессов поиска, извлечения, обработки, анализа и визуализации БД, а также в разработке статистической модели прогнозирования количества пациентов СД на основе регрессионного анализа с использованием технико-экономических показателей в сфере медицинских услуг по диабету.

Целью диссертационной работы является разработка метода прогнозирования роста пациентов СД по РК, а также разработка ИС поддержки врача-эндокринолога по диагностике СД на базе инструментов BigData технологии.

Объектом исследования являются процессы прогнозирования больных СД на последующие года с целью подготовки процедуры закупа необходимого количества инсулина и ИС диагностики пациентов в медицинских учреждениях с помощью инструментов технологии BigData.

Предмет исследования – ИС поддержки врача-эндокринолога по диагностике СД на базе инструментов BigData технологии, а также методы прогнозирования роста пациентов СД по Республике Казахстан.

Задачи исследований, их место в выполнении научно-исследовательской работы в целом. В соответствии с поставленной целью определены следующие задачи, которые необходимо решить в настоящей работе:

– выполнить анализ современного состояния применения методов прогнозирования и использования технологий BigData в сфере медицинского обслуживания;

– разработать метод прогнозирования роста пациентов с помощью методов регрессионного анализа и с применением библиотеки scikit-learn для определения количества закупаемого инсулина;

- разработать систему на основе алгоритмического и программного обеспечения по диагностике и сопровождению лечения СД;
- разработать платформу ИС с применением инструментов технологии BigData;
- проведение исследовательского анализа, обработки больших данных с помощью сводной статистики и графических представлений данных.

Таким образом, сформулирована цель и задачи исследования по прогнозированию возможного количества пациентов, которая позволит заранее запланировать количество закупаемого инсулина и других средств контроля диабета. Также проведение исследовательского анализа, обработки больших данных с целью выявления проверки гипотез, закономерностей, с помощью сводной статистики и графических представлений данных позволят повысить качество обработки больших массивов данных. В дальнейшем, планируется запустить данную ИС в открытом доступе для практического обучения методам диагностирования СД на основе полученных теоретических знаний студентами медицинских университетов.

Методологическая база исследований. Поставленные задачи решались путём проведения теоретических и практических исследований. В ходе решения поставленных задач использовались теории математической статистики, машинного обучения, математического анализа. Практическая реализация алгоритмов осуществлялась на основе методов объектно-ориентированного программирования и объектно-ориентированных баз данных.

Положения, выносимые на защиту. На защиту выносятся следующие положения:

- на основе множества допустимых решений статистических задач, полученные результаты позволяют спрогнозировать количество пациентов.
- была предложена модель системы по распределённым вычислениям, обеспечивающее эффективное выполнение разработанных алгоритмов;
- была разработана структурная модель процессов поиска, извлечения, обработки и анализа данных;
- была разработана и апробирована на основе системного подхода ИС для диагностики СД на базе инструментов BigData технологий.

Структура и объем диссертации. Диссертационная работа состоит из введения, четырех разделов основного содержания, заключения и приложений, библиографического списка из 139 наименований и содержит 117 страниц, 88 рисунков и 6 таблиц.

В рамках данной диссертационной работы было подготовлено и издано 14 исследовательских работ по рассматриваемой теме, из них:

- пять статей опубликованы в издательствах, отвечающих требованиям высшей аттестационной комиссии Министерства образования науки РК;
- восемь статей размещены в сборниках трудов международных конференций;

– одна статья в международном рецензируемом научном журнале.

Автор выражает огромную признательность врачу высшей категории, кандидату медицинских наук, Аканову Жанай Айкановичу за ценные советы при планировании исследования.

Помимо этого автор выражает благодарность своему наставнику – ныне покойному к.т.н., Омару Т.К., за поддержку во всех моих начинаниях.

Особую признательность автор выражает, научному консультанту, доктору PhD, ассоциированному профессору Сапарходжаеву Нурбеку Пажарбековичу за всестороннюю помощь и внимание в процессе написания диссертации.

1 АКТУАЛЬНЫЕ ПРОБЛЕМЫ ДИАГНОСТИКИ И ПРОГНОЗИРОВАНИЯ САХАРНОГО ДИАБЕТА В СОВРЕМЕННОМ МИРЕ

1.1 Исследование распространенности сахарного диабета в мире

В соответствии с определением, утвержденным ВОЗ, СД представляет собой состояние, когда фиксируется длительное повышение уровня сахара в крови человека. На повышение уровня сахара влияют много факторов. На сегодняшний день СД можно отнести к очень серьезному хроническому заболеванию, которое протекает незаметно, если не пройти специальное обследование на наличие сахара в крови.

Важным фактором таких рекомендаций является разработка единых норм и стандартов по диагностике и лечению диабета. Также ВОЗ предоставляют и разрабатывают такие стандарты, которые касаются всех различных заболеваний.

Быструю распространенность диабета заметили в странах со средним и низким уровнем дохода. Диабет можно лечить, а его последствия можно избежать или отсрочить с помощью диеты, физической активности, приема лекарств и регулярного обследования и лечения осложнений.

Повышенный уровень глюкозы в крови может со временем привести к плачевным последствиям, влекущим к серьезным повреждениям сердца, сосудов, глаз, почек и нервов. Если СД не контролировать и не следовать рекомендациям, то такие последствия неизбежны. На сегодняшний день более 400 миллионов человек живут с диабетом в соответствии с рисунком 1.1 [7].

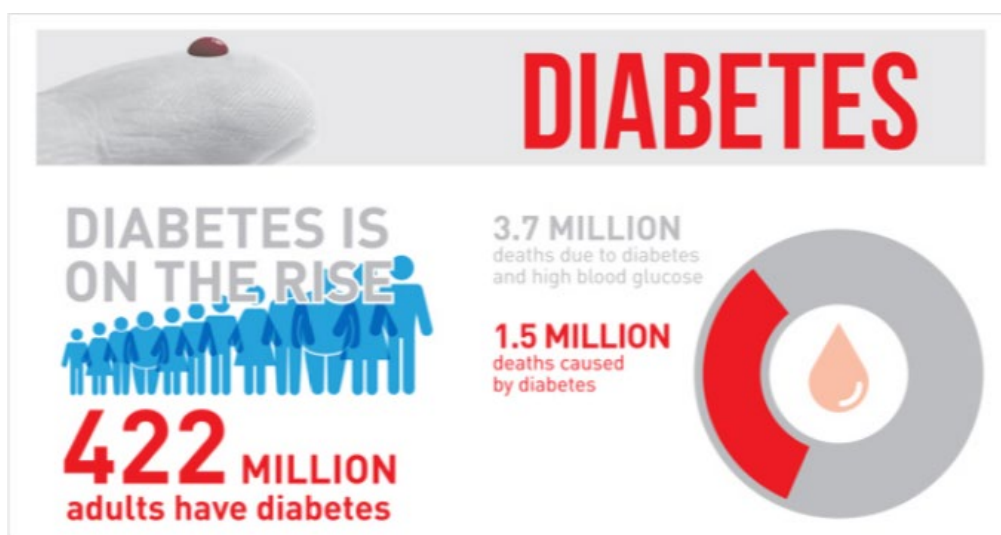


Рисунок 1.1 – Глобальный доклад о диабете [6].

По возможности, население должно проходить различные обследования и осмотры, чтобы выявить это заболевание или узнать о

предрасположенности человека. При диагностике этого заболевания у пациента необходимо будет сообщить, как правильно жить с этим диагнозом, то есть правильное питание и здоровый образ жизни. Чем дольше человек живет с необнаруженным диабетом, тем хуже могут быть его последствия для здоровья. Должен быть открытый доступ к первоначальной диагностике, такой как анализ уровня глюкозы, в крови, и должен быть доступен в учреждениях первичной медико-санитарной помощи. СД необходимо исследовать и изучать как глобальную социальную проблему, которая несет скрытые угрозы будущим поколениям. Поздняя диагностика и неадекватное лечение приведут к различным осложнениям для всего организма, что приведет к снижению качества жизни и ее продолжительности.

В регионах Юго-Восточной Азии и Западной части Тихого океана было диагностировано более 50 процентов, не диагностированных случаев. Также самые низкие доли не диагностированных случаев были зафиксированы в Северной Америке и Карибском бассейне 37,6 % и в Европе 37,8 %. Многие хронические заболевания быстро распространяются и находятся на более поздних стадиях в странах Южной Азии, чем в других регионах. У жителей Южной Азии диабет 2 типа развивается в более молодом среднем возрасте и прогрессирует быстрее, чем в других этнических группах [8].

В официальной статистике Международной Диабетической Федерации (МДФ) утверждается, что в 2015 году в семи странах было более десяти миллионов людей с заболеванием диабета: Китай, Индия, Соединенные Штаты Америки; Бразилия, Российская Федерация, Мексика и Индонезия. На сегодняшний день около 422 миллиона человек в возрасте от 20 до 79 лет болеют диабетом, в то время как 47 % смертей, связанных с диабетом, происходят в возрасте до 60 лет. К 2040 году каждый десятый - 642 миллиона человек - согласно прогнозам МДФ, будет иметь это состояние [9].

Опубликованные исследования международных торговых рынков, сопровождающиеся экономическим ростом, учитывают приток дешевых обработанных пищевых продуктов. Газированные напитки и продукты, использующие большое количество сахара, предлагают дешевую замену традиционной диете и подвергают население риску диабета. Однако, корреляционные исследования предполагают только тенденции и взаимосвязь между диабетом и диетой, а данные о населении не могут с научной точностью доказать, что сахар вызывает диабет [10].

СД является мировой проблемой. В десятку стран входят такие страны как: Китай, Индия, США, Бразилия, Мексика, Индонезия, Россия, Египет, Германия, Пакистан [11].

В исследованиях восьмого издания МФД 2017 года, посвященному диабету, были включены последние данные, содержащие текущую информацию и прогнозы по диабету во всем мире [12]. Благодаря деятельности МДФ, Всемирный день диабета охватывает миллионы людей во всем мире и объединяет диабетические общества из 160 стран, чтобы повысить осведомленность о диабете и его осложнениях. Генеральная

Ассамблея Организаций Объединённых Наций (ООН) в своей резолюции также предложила государствам-членам разработать национальные программы по борьбе с диабетом и уходу за людьми с этим заболеванием [13].

Важно отметить, что все меры предпринимаемые ВОЗ являются эффективными и общедоступными для медицинских организаций и различных правительственных структур. Пропаганда здорового образа жизни на высоких государственных уровнях, особенно в странах с низким и средним уровнем доходов, является необходимым.

Существующие показатели распространенности СД 2 типа позволяют проводить анализ и оценку оказываемой медицинской помощи и определить затраты на лечение. Сотрудники медицинских организаций и члены исследовательских организаций демонстрируя данные о заболеваемости, опираются на базы данных Всемирного банка включительно. Статистические показатели распространенности СД в мире определяют необходимость проведения прямых исследований для выявления настоящей картины распространенности патологии, которые были бы использованы в качестве научной основы для планирования профилактических медицинских мероприятий. Согласно статистическим данным, предоставленным официальным сайтом Всемирного банка только в 2017 году, распространенность диабета среди населения в возрасте от 20 до 79 лет составляет для примера по ОЭСР (Организации экономического сотрудничества и развития)– 8,206% [14].

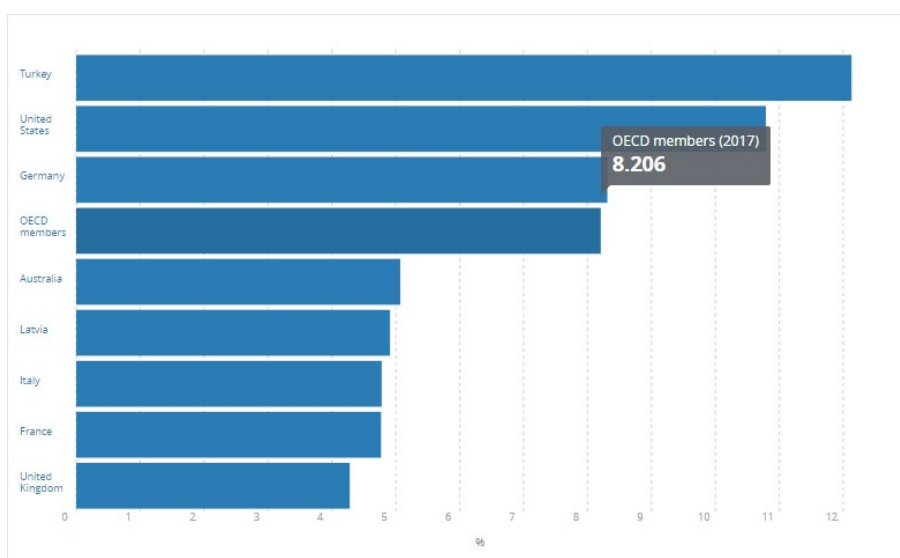


Рисунок 1.2 – Распространенность диабета среди населения в возрасте от 20 до 79 лет [15]

Можем заметить, что за последние 40 лет распространенность ожирения в мире значительно выросла, менее 1% в 1975 году до 6–8% и в 2016 году среди девочек и мальчиков - от 3% до 11% среди мужчин и от 6% до 15% среди женщин за тот же период времени [16]. СД раньше считали

болезнью запада, так как в развитых странах население очень часто употребляло жирную и быстрого приготовления пищу. Также малая активность и сидячий образ жизни соответственно повлияло на рост заболевания диабетом.

Население всей планеты растет с каждым днем и на Азию приходится 60% населения мира, страдающего диабетом. Быстрое экономическое развитие, урбанизация и изменения в состоянии питания можно увидеть в последние десятилетия в Азии. Эти изменения привели к колоссальному увеличению распространенности диабета в течение относительно короткого времени. Было уже отмечено, что в 1980 году менее 1% взрослых китайцев имели сахарный диабет, а к 2008 году распространенность достигла почти 10%. Проведенные исследования демонстрируют увеличение роста заболеваемости среди сельского населения Китая. Авторы утверждают, что на высокую распространенность огромное влияние оказало низкий уровень осведомленности о заболевании, а также о ходе правильного подобранного лечения и самоконтроля. Общественному здравоохранению необходимо рассмотреть как важное стратегическое направление лечение и профилактику СД [17].

Более 92 миллионов населения Китая уже имеет диабет, а также 148 миллионов были на уровне преддиабетического состояния. Такие большие цифры указывают на серьезность ситуации, так как Китай уже обгоняет Индию по количеству пациентов, страдающих от диабета. Однако уместно отметить тот факт, что уже в городских районах на юге Индии распространенность диабета достигла почти 20% [18]. Ежедневные стрессы также влияют на возникновение СД. В 2010 году в северном индийском мегаполисе проводилось исследование распространенности депрессии у диабетиков, которая достигала 41% [19].

Распределение полов по результатам исследований в Индии показывают противоречивые результаты, а некоторые исследования из северных регионов показывают женскую предрасположенность [20]. Другое исследование из южной Индии сообщило о более высокой распространенности среди мужчин [21].

Многие исследования противоречат друг другу, но показатели того, что цифры неуклонно растут можно увидеть в ежегодных отчетах всемирных организаций. СД в северной Индии растет угрожающими темпами. Информированность и осведомленность о диабете в Индии, особенно в северных сельских районах, очень скудны. Многочисленные факторы определяют тенденцию развития диабета в северной Индии, которую необходимо учитывать в первую очередь, чтобы добиться профилактики и лучшего контроля диабета [22].

Авторы данного исследования опубликовали результаты, где по сравнению с сильным полом женщины чаще соблюдают религиозные посты. Большое количество времени проводят в закрытых помещениях, не имея

возможности заниматься спортом, тем самым эти основные факторы препятствуют контролю диабета [23].

На сегодняшний день в Великобритании диагностированных больных уже 3,8 миллиона человек [24]. Важно отметить, что из 17 человек в Великобритании хоть у одного существует диабет – это может быть диагностированный или не диагностированный [25]. Предполагается, что к 2025 году в Великобритании уже пять миллионов человек будут страдать от диабета [26]. В данном исследовании приводятся данные, что более 400 человек каждый день, более 17 человек каждый час и около трех человек каждые десять минут узнают о своем диагнозе [27]. В одной из публикаций было отмечено, что исследователи, связанные с Лондонской школой гигиены и тропической медицины и ВОЗ, намеревались объяснить причины роста заболеваемости диабетом. Их результаты, опубликованные в журнале «Public Health Nutrition», сравнивают распространенность диабета в 173 странах, потребление продуктов питания и социально-экономическую ситуацию [28]. Проанализировав всю эту информацию, можем прийти к выводу, что есть взаимосвязь между увеличением числа городов, перемещением населения из сел в города и диабетом. Уровень жизни в городских условиях намного комфортабельнее и мало подвижнее чем в сельской местности, где физическая активность на первом месте. Следовательно, мы видим корреляцию между этими закономерностями, так как жизнь в городе располагает употреблению большого количества сахара.

Анализ проблемы распространения сахарного диабета в Казахстане. По официальным данным ВОЗ, в РК около 11,5% людей от общего количества населения страны страдают от СД; из которых представители женского пола составляют - 11,7%, а мужского пола – 11,3% [29].

Согласно материалам Национального регистра, в Казахстане зарегистрированных пациентов в 2018 году составило 317 597 человек. Общее количество взрослого населения страдающих от диабета – 314 407 человек, до 14 лет 2 379 человек, а также от 15 до 17 лет около 811 человек. В РК действующий регистр по сахарному диабету содержит данные только о больных, которым уже было диагностировано данное заболевание. Следует учитывать, что значительное количество пациентов наблюдается в частных медицинских организациях и эти данные не охвачены регистрацией [30].

По данным МДФ, в РК распространенность исследуемой патологии должна составлять 7,5%, где по их исследованиям имеется также 0,5 млн. больных с не диагностированным диабетом [31].

Смертность

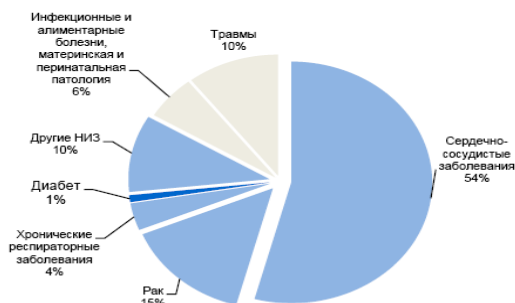
Количество случаев смерти от сахарного диабета

	Мужчины	Женщины
в возрасте 30–69 лет	350	430
в возрасте 70 лет и старше	190	440

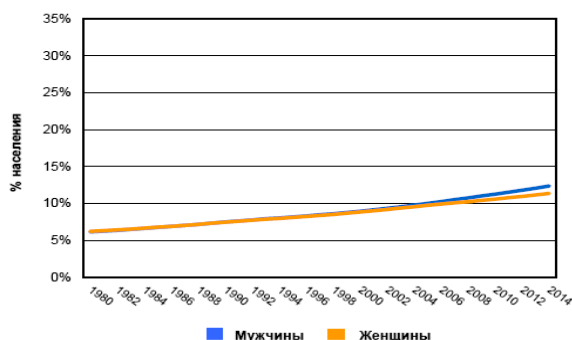
Количество случаев смерти, которые можно отнести к высокому содержанию сахара в крови

	Мужчины	Женщины
в возрасте 30–69 лет	4 090	2 340
в возрасте 70 лет и старше	2 970	5 550

Относительный показатель смертности (% от суммарного числа смертей, все возраста)



Тенденции стандартизированной по возрасту распространенности диабета



Распространенность сахарного диабета и сопутствующих факторов риска

	Мужчины	Женщины	Всего
Диабет	11.3%	11.7%	11.5%
Избыточный вес	59.8%	57.7%	58.7%
Ожирение	21.3%	25.4%	23.5%
Недостаточная физическая активность	18.8%	21.0%	20.0%

Рисунок 1.3 – Данные ВОЗ на 2017 г.

По данным исследователей на распространение диабета в стране имеют влияние такие факторы как расовая принадлежность, возраст пациента, наследственная предрасположенность. Также большое влияние имеют такие модифицируемые факторы, как уровень глюкозы и холестерина, ожирение и сердечно-сосудистая патология [32].

Ситуация с распространенностью диабета в Казахстане также вызывает беспокойство, так как согласно данным, предоставленным Общественным фондом «Казахстанское общество по изучению диабета» [33]. В Казахстане количество диагностированных людей с диабетом в 2018 году превышает 300 тысячи человек. В РК СД отводится в ранг социально значимых заболеваний, требующих системных решений, так что это требует разработки и принятия, адекватных мер по улучшению лечения заболеваний и увеличению продолжительности жизни граждан.

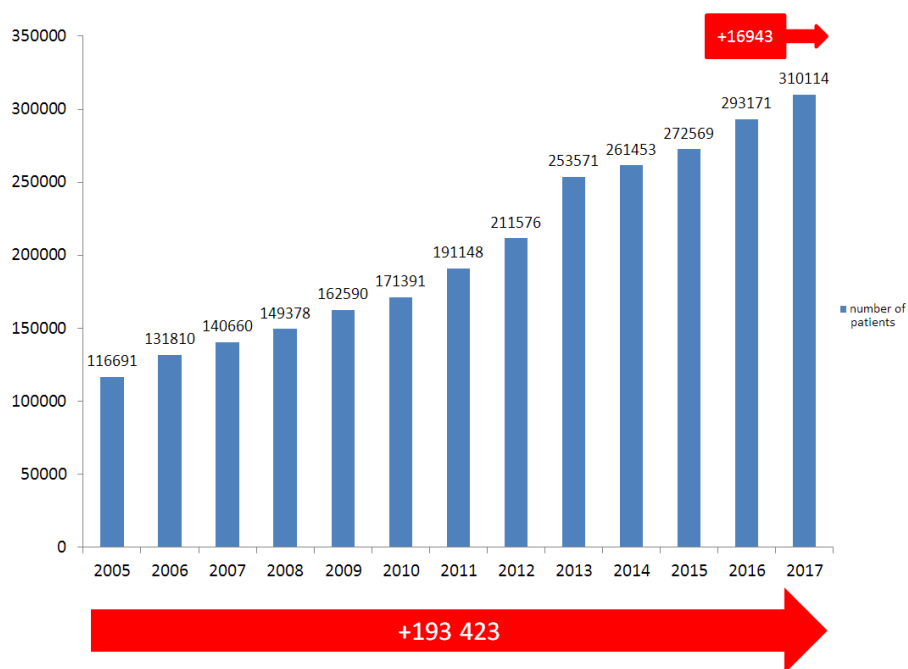


Рисунок 1.4. – Заболеваемость СД в РК составляет 1,7 % населения (18 157 078)

Существуют нормативные документы, законодательные акты для оказания эндокринологической помощи взрослому и детскому населению: Приказ Министерства здравоохранения РК (МЗРК) от 16 марта 2011 года № 145 «Об утверждении Правил проведения профилактических медицинских осмотров целевых групп населения» [34]; Приказ МЗРК от 5 апреля 2013 года № 211 «О внедрении Программы управления хроническими неинфекционными заболеваниями в пилотных регионах» [35]; Приказ МЗСРРК от 15 марта 2016 года № 203 «О некоторых вопросах управления хроническими неинфекционными заболеваниями в рамках пилотного проекта» [36]; Приказ МЗРК от 25 мая 2012 года №364 «Об утверждении Положения о деятельности организаций здравоохранения, оказывающих эндокринологическую помощь населению Республики Казахстан» [37]; Приказ и. о. МЗРК от 4 ноября 2011 года № 786 «Об утверждении перечня лекарственных средств и изделий медицинского назначения для бесплатного обеспечения населения в рамках государственного обеспечения бесплатной медицинской помощью на амбулаторном уровне» [38]; Приказ МЗРК от 27 апреля 2019 года № ҚР ДСМ-55 «О внесении изменений и дополнений в приказ Министра здравоохранения РК от 18 июля 2018 года № 434 «Об утверждении списка лекарственных средств, изделий медицинского назначения в рамках гарантированного объема бесплатной медицинской помощи и в системе обязательного социального медицинского страхования, закупаемых у Единого дистрибьютора на 2019 год» [39].

Несмотря на предпринимаемые меры на государственном уровне по снижению уровня заболеваемости СД в стране, нехватку врачей-эндокринологов в медицинских структурах можно заметить в статистическом

сборнике «Здоровье населения РК и деятельность организаций здравоохранения в 2015 году». Согласно приведенным данным заболеваемость населения по всем болезням на 2017 год всего составила 57896,9 человек на 100 тыс. населения. На 2017 год заболеваемость СД среди всего населения достигла 204,2 человек на 100 тыс. населения, а в 2016 году было зафиксировано 200,4 на 100 тыс. населения. Из них женщины в 2017 году 235,1 на 100 тыс. населения, а в 2016 было 237,9 на 100 тыс. населения. В данном сборнике указано, что обеспеченность населения медицинскими специалистами в области эндокринологии составляет в среднем 0,4 на 10 тыс. населения [40, 14] в соответствии с рисунком 1.5.



Рисунок 1.5 – Кадровая обеспеченность на 10 тыс. населения, 2015 г.

Следует отметить что, многочисленные программы скрининга населения для ранней диагностики и профилактики СД должны быть одним из безусловных приоритетов современного здравоохранения РК [41].

Совершенствование управления качеством медицинских услуг занимает важное место в контексте стратегического развития здравоохранения Казахстана до 2020 года [42-43].

Своевременное диагностирование и правильный образ жизни могут помочь людям, страдающим от этой болезни, помогая пациентам вести нормальную жизнь. Также можно с уверенностью сказать, что СД, это самое распространенное НИЗ. Его опасность в том, что люди уже болеющим этим недугом не знают об этом. Обнаруживают его только тогда, когда он уже находится на запущенной стадии.

СД вызывает такие осложнения, как сердечнососудистые заболевания, хроническую почечную недостаточность, инфаркт миокарда, слепоту, инсульт, ишемическую болезнь сердца, периферическую полинейропатию и тяжелые нарушения функционирования центральной нервной системы, приводящих к ранней инвалидизации и высокой летальности таких пациентов [44]. Любой гражданин страны имеет право получать бесплатную

медицинскую помощь за счет бюджетных средств и включает профилактические, диагностические и лечебные медицинские услуги, обладающие наибольшей доказанной эффективностью [45].

По инициативе МЗРК в июне 2018 года была озвучена новая модель гарантированного объема бесплатной медицинской помощи (ГОБМП), где планировалось перевести, включая СД из ряда хронических заболеваний из группы диспансерного учета под активное динамическое наблюдение. Данное нововведение позволяет пациенту принимать непосредственное участие в процессе контроля и улучшения своего самочувствия совместно с врачом. Позволяя снизить риски возникновения осложнений, инвалидизации.

Непосредственно сам пациент должен придерживаться определенного образа жизни, регулярно принимать необходимые лекарственные средства для снижения артериального давления, контроля уровня глюкозы в крови, контроля над бронхиальной астмой и т.д. [46]. СД является хроническим заболеванием и лечение на ранней стадии можно проводить путем внесения изменений в образ жизни людей. Естественно, эту болезнь нужно контролировать с помощью лекарственных средств на ранних стадиях. Если поддерживать диету и вести спортивный образ жизни, тогда можно не запустить рост этой болезни. На поздних стадиях помогает введение инсулина, которое контролируется медицинскими специалистами.

1.2 Сравнительный анализ современных медицинских информационных систем диагностики

В соответствии с целями данного исследования был произведен обзор разработанных медицинских диагностических ИС. В результате, проведенный анализ показал, что в РК такие разработки по диагностике диабета не проводились. На сегодняшний день исследования в области медицины по диагностике и лечению различных заболеваний обуславливается высокой потребностью в использовании новейших ИТ современными медицинскими учреждениями [47].

Разработка и внедрение ИТ в медицинскую отрасль позволит улучшить качество диагностирования и лечения болезни.

С помощью информатизации увеличивается возможность получения аналитических прогнозов и точной информации. Планирование клинической оценки ИС поддержки принятия решений, требует стратегии, которая охватывает различные аспекты клинической проблемы, технических трудностей интеграции и реализации программного и аппаратного обеспечения. Разработка и оценка диагностических систем поддержки принятия решений остается активной и сложной областью исследований.

Использование ИТ для улучшения ухода за пациентами продолжает оставаться первостепенной целью в секторе здравоохранения. Некоторые утверждают, что мы приближаемся к переломному моменту, когда можно ожидать неуклонного роста числа внедренных ИС здравоохранения и интенсивности их использования в различных условиях. Европейские страны

уже считаются лидерами в использовании электронных медицинских карт в первичной медико-санитарной помощи, где врачи уже более десяти лет используют электронные медицинские карты в своей повседневной практике [48].

В рамках исследования нами был проведен тщательный поиск в области применения инструментов технологии BigData в диагностике сахарного диабета. Для этих задач были использованы поисковые системы fips.ru, Google Patents для поиска по зарубежным патентам. Поисковой анализ продемонстрировал следующие результаты, где самое большое количество финансовых вложений является страна США.

Наиболее известным на сегодня случаем применения искусственного интеллекта в медицине является проект «International Business Machines (IBM) Watson Health». В 2013 году IBM и «Memorial Sloan Kettering Cancer Center (MSKCC)» в Нью-Йорке начали использовать Watson в качестве системы поддержки принятия решений для диагностики и назначения лечения в онкологии. Через MSKCC ежегодно проходит 30 тыс. пациентов. История болезни каждого пациента сравнивается с историями других пациентов с похожим анамнезом, а также со всеми известными медицинскими публикациями на предмет совпадения тех или иных признаков болезни. В результате пациент получает персонализированную диагностику и лечение, а когнитивные функции Watson продолжают улучшаться с каждым новым пациентом, потому что обучение идет непрерывно. Анализ 200 млн. цифровых документов занимает 3 секунды. В перспективе IBM Watson должен стать идеальным врачом-диагностом, который путём анализа симптомов и истории болезни пациента способен поставить диагноз и назначить самое вероятное лечение. При постановке диагнозов IBM Watson использует тот же самый алгоритм ответов на вопросы [49].

Авторы исследования [50] во время обзора существующих информационных медицинских систем с использованием Google Scholar обнаружили много интересных ИС диагностики болезней, которые были исследованы.

Как утверждается в исследовании [51] быстрые и точные медицинские диагнозы имеют решающее значение для успешного лечения заболеваний. Используя алгоритмы машинного обучения, и основываясь на результатах лабораторных анализов крови, исследователи создали две модели для прогнозирования гематологических заболеваний.

Авторы использовали в прогностической модели все доступные параметры анализа крови, а в другой использовался только уменьшенный набор, который обычно измеряется при поступлении пациента. Обе модели дали хорошие результаты, получив точность прогноза 0,88 и 0,86 при рассмотрении списка из пяти наиболее вероятных заболеваний и 0,59 и 0,57 при рассмотрении только наиболее вероятного заболевания. Клинический тест показал, что точность предложенных прогностических моделей была на

уровне точности специалистов по гематологии. Данное исследование впервые показало, что прогностическая модель машинного обучения, основанная только на анализах крови, может быть успешно применена для прогнозирования гематологических заболеваний. Этот результат и может открыть беспрецедентные возможности для медицинской диагностики.

Авторы работы [52] утверждают, что основанная на правилах система диагностики и лечения проблем с ушами, была разработана с использованием языка объектов SL5. Основными источниками знаний для этой управляемой системы для диагностики проблем слуха и слуха являются специализированные сайты о проблемах слуха. В настоящее время система, основанная на правилах, имеет 12 правил, которые охватывают некоторые проблемы со слухом и ушами, она может помочь больным с воспалением среднего уха.

В работе [53] исследователи разрабатывают медицинскую экспертную систему (МЭС) для диагностики и лечения гипертонии у беременных, которая будет использоваться в отделе репродуктивного здоровья Учебно-справочной больницы Мои в Элдорете, Кения. В ходе исследования было установлено, что в отделе репродуктивного здоровья ощущается острая нехватка специалистов-акушеров, что означает, что у специалистов также недостаточно знаний о диагностике и лечении гипертонии у беременных. Появилась необходимость разработки МЭС в качестве инструмента обмена экспертными знаниями, которая будет использоваться другими медицинскими работниками в отделе репродуктивного здоровья, которые не являются специалистами в диагностике и лечении гипертонии во время беременности.

Авторами работы [54] была исследована экспертная система (ЭС) для диагностики заболеваний кожи была разработана с использованием «CLanguage Integrated Production System» (CLIPS), чтобы помочь пользователю диагностировать следующие заболевания кожи. В этой статье была исследована причина болезней и даны методы лечения болезней, когда это было возможно.

Авторы научной работы [55] проанализировали и изложили проблемы из-за нехватки специалистов, практикующих врачей и медицинских учреждений. В попытке решить такие проблемы, в исследованиях были предприняты попытки спроектировать и разработать ЭС, которые могут дать рекомендации врачам и пациентам, чтобы облегчить диагностику и рекомендовать лечение пациентов. В обзоре [56] представлены комплексные исследования МЭС для диагностики различных заболеваний.

Авторы суммирует некоторые из основанных на правилах нечетких ЭС и медицинских диагностических систем на основе искусственной нейронной сети. Исследователи в работе [57] с помощью методов искусственного интеллекта, обзора литературы и методологий исследований разработали ИС для врачей в Гане. В дальнейшем могут использовать разработанную ЭС с помощью мобильных технологий для ускорения диагностики,

подтверждения собственного диагноза, предоставления рекомендаций по обнаруженному диагнозу и обеспечения консультации по определенным заболеваниям при диагностике у пациента.

В основе исследования лежит Учебная больница Корле-Бу, которая в настоящее время является третьей по величине больницей в Африке и ведущим национальным справочным центром больницы в Гане.

В исследовании авторами [58] описана конструкция предлагаемой системы, которая была разработана для помощи ортопеду в диагностике заболевания артритом по таким симптомам, как: боль при надавливании в суставе, воспаление, отмечаемое отеком сустава, скованность, особенно по утрам, потеря гибкости сустава. Ограниченная подвижность суставов, деформация суставов, потеря веса и усталость, неспецифическая лихорадка и крепитация. В предлагаемой работе представлен обзор болезни артрита, дана причина болезней и дано лечение по возможности. Язык SL5 Object Expert System использовался для проектирования и реализации предложенной ЭС.

На сегодняшний день ишемическая болезнь сердца является наиболее опасным заболеванием сердца, которое может привести к внезапной сердечной смерти. Тем не менее, диагностика является довольно дорогой и трудоемкой процедурой, которую пациент должен пройти.

Протокол методов обзора определяет лучшие методы и классификатор для идентификации данной болезни. Авторы работы [59] предлагают два рабочих процесса на основе двух наборов параметров для экземпляров А и В. Необходимо следовать соответствующей процедуре для будущей оценки процесса автоматической диагностики ишемической болезни сердца. Повышение доступности медицинской помощи включает предоставление возможностей цифровой диагностики людям, которые в ней нуждаются, особенно тем, кто не может попасть в кабинет врача. Это одна из главных тем цифрового здравоохранения.

В последние годы были достигнуты невероятные успехи в лечении и диагностике рака, а технологические достижения в диагностике рака позволили выявить его на ранней стадии. В Центре по лечению рака Уинтропа, Нью-Йоркского университета, ведущие специалисты в области диагностики используют самые современные технологии и методы для постановки первоначального диагноза или подтверждения предполагаемого диагноза. При выборе диагностических тестов врач будет учитывать возраст и состояние здоровья, тип предполагаемого рака, серьезность симптомов и результаты предыдущих тестов [60].

Одним из важнейших технологических трендов в области медицины и диагностики, в частности, стала персонализация. Благодаря возможности получать огромный массив данных по каждому пациенту врачи способны принимать информированные терапевтические решения. Точность и быстрота диагностики в этом процессе имеет большое значение: до 70% клинических решений основываются на данных диагностических исследований [61]. Следовательно, можно отметить, что внедрение ИТ

технологии в сферу медицины кардинальным образом вносят большой вклад в продление жизни и улучшения качества жизни больных.

1.3 Исследование источников и характеристик информационных ресурсов по сахарному диабету

В Послании Первого Президента страны Н.А. Назарбаева «Третья модернизация Казахстана: глобальная конкурентоспособность» соответственно ставятся задачи: качественного диагностирования и лечения широкого спектра болезней; формирования программы «Цифровой Казахстан» [62].

Согласно исследованию, из-за широкой распространенности диабета по всему миру, было создано множество диабетических ассоциаций, школ диабета, организаций по борьбе с диабетом. Число таких организаций переваливает за несколько сотен по всему миру: МДФ, Американская Диабетическая Ассоциация, Европейский фонд по изучению диабета, Альянс европейских исследований диабета, Диабетическая Ассоциация РК. Авторы утверждают, что такие специализированные организации предоставляют наиболее точную информацию и являются официальными источниками информации по диабету в определенных регионах [63].

Перечисленные организации имеют обратную связь друг с другом и имеют совместные программы, чтобы поощрять и поддерживать исследования в области диабета, быстро распространять полученные знания и облегчать их применение [64]. Одну из таких программ можно увидеть в сотрудничестве Европейского фонда по изучению диабета и Японского диабетического общества. Целью этой программы является содействие совместным исследованиям в области диабета. Эти программы финансируются и в рамках данных программ выделяются стипендии до 50000 евро [65].

Проведение организационных мероприятий по поддержке и финансирование исследований по профилактике, уходу и лечению диабета должны находить поддержку у государственных структур. Также необходимо увеличение сотрудничества врачей друг с другом. Принятия участия в международных конференциях по обмену опытом между медицинскими работниками и учеными исследующие более эффективные методы лечения должны иметь первостепенный характер.

Исследователи и другие сотрудники, работающие в сфере диабета должны иметь доступ ко всем источникам, информирующим о СД. Также немаловажную роль играет работа по осведомленности населения о профилактике и лечения диабета. Все эти предпосылки и нормы помогут в лечении этого хронического заболевания. Все эти параметры были описаны и предложены в Китайском Диабетическом сообществе [66], где одним из таких организаций является Youth Diabetes Action (YDA). Важным направлением YDA является в предоставлении услуг, которые содержат

программы поддержки лечения СД, а также медицинскую и финансовую поддержку [67].

1.4 Постановка цели и задачи исследования

Проведённый анализ современного состояния исследований по прогнозированию роста пациентов показал недостаточную ее проработанность, а также полное отсутствие разработок в области медицинских диагностических систем. В связи с этим целью данной работы является исследование прогнозирования роста пациентов СД по РК на каждый последующий год, а также создание на базе инструментов BigData технологии ИС поддержки врача-эндокринолога по диагностике СД и проведение анализа, обработки и визуализации больших данных. Поставленная цель определила задачи данного исследования:

- проведение анализа современного состояния по применению методов прогнозирования и использования технологий BigData в сфере медицинского обслуживания;

- разработка метода прогнозирования роста пациентов на основе методов регрессионного анализа, а также с применением библиотеки scikit-learn для определения количества закупаемого инсулина;

- разработка системы на основе алгоритмического и программного обеспечения по диагностике СД;

- разработка платформы ИС с применением инструментов технологии BigData;

- проведение исследовательского анализа, обработки больших данных с помощью сводной статистики и графических представлений данных.

Таким образом, данные сформулированные цель и задачи исследования по прогнозированию возможного количества пациентов позволит заранее запланировать количество закупаемого инсулина и других средств контроля диабета. Также, разработанная ИС на базе инструментов BigData технологий для поддержки врача-эндокринолога по диагностике СД помогает студентам, проходящим обучение на врачей-эндокринологов методам диагностирования СД.

Выводы по первому разделу

1. Были рассмотрены актуальные проблемы диагностики и исследованы причины распространенности СД в мире.

2. Произведен сравнительный анализ современных медицинских ИС диагностики, которые продемонстрировали успешные результаты при диагностировании болезней.

3. Были исследованы источники и характеристики информационных ресурсов по СД.

4. Сформулированы цель и задачи исследования.

Решению сформулированных задач посвящены следующие главы.

2 ИССЛЕДОВАНИЕ МЕТОДОВ ПРИМЕНЕНИЯ ТЕХНОЛОГИИ BIGDATA В СФЕРЕ МИРОВОГО ЗДРАВООХРАНЕНИЯ

2.1 Аналитика применения обработки больших данных во всех направлениях

В современном мире аналитика применения обработки больших данных (БД) - это процесс изучения больших наборов данных, которые содержат различные типы данных. БД позволяют обнаружить все скрытые шаблоны, неизвестные корреляции, тенденции рынка, предпочтения клиентов и другую полезную бизнес-информацию и не только. Затем аналитические результаты могут привести к более эффективному маркетингу, новым возможностям получения дохода, качественному обслуживанию клиентов, повышению эффективности работы, конкурентным преимуществам по сравнению с конкурирующими организациями и другим бизнес - преимуществам [68].

БД превратились в центральную тему в различных начинаниях и научных исследованиях. Это объясняется тем, что способность создавать, собирать, передавать, подготавливать и изучать исключительные меры различий в информации имеет практически широкое применение.

БД, с их потенциалом для получения ценной информации для улучшения процесса принятия решений, в последнее время вызвали значительный интерес со стороны, как ученых, так и практических исследователей. Аналитика БД все больше становится популярной практикой, которую многие организации применяют с целью получения ценной информации из БД. Аналитический процесс, включая развертывание и использование инструментов аналитики БД, рассматривается организациями как инструмент повышения операционной эффективности, хотя он обладает стратегическим потенциалом, направляет новые потоки доходов и получает конкурентные преимущества перед конкурентами [69].

Аналитика может быть разделена на 5 типов:

– описательная аналитика: самый простой класс исследований - тот, который позволяет вам объединять огромную информацию в более мелкие, более ценные порции данных [70];

– прогнозная аналитика: прогнозная аналитика может быть наиболее часто используемой категорией аналитики данных, поскольку она используется для выявления тенденций, корреляций и причинно-следственных связей. Категория может быть далее разбита на прогнозное моделирование и статистическое моделирование [71];

– предписательная аналитика: наряду с описательной и прогностической аналитикой предписательная аналитика, является одним из трех основных типов аналитических компаний, используемых для анализа данных. Этот тип аналитики иногда описывается как форма прогнозирующей аналитики, но немного отличается по своему фокусу. Целью предписывающей аналитики является выработка наилучших возможных

рекомендаций для ситуации, которая разворачивается, учитывая то, что аналитик может определить по имеющимся данным [72];

– диагностическая аналитика: это форма расширенной аналитики, которая анализирует данные или контент для ответа на вопрос «Почему это произошло?» и характеризуется такими методами, как детализация, обнаружение данных, анализ данных и корреляции. Диагностическая аналитика более глубоко изучает данные, чтобы попытаться понять причины событий и поведения [73];

– когнитивная аналитика: когнитивная аналитика объединяет ряд интеллектуальных технологий, таких как искусственный интеллект, алгоритмы машинного обучения, глубокое обучение и т.д., для применения интеллектуального интеллекта человека для выполнения определенных задач. В основном, этот тип аналитики вдохновлен тем, как человеческий мозг обрабатывает информацию, делает выводы и систематизирует инстинкты и опыт для обучения, такого как понимание не только слов в тексте, но и полного контекста того, что пишется или говорится.

Все эти интеллектуальные технологии делают когнитивное приложение более умным и более эффективным с течением времени, участь на его взаимодействии с данными и людьми [74]. Такие компании, как Google и Amazon, являются мастерами майнинга и анализа БД. Они используют знания, полученные в результате анализа БД, чтобы добиться превосходства над своими конкурентами. Система анализирует БД, такие как история покупок, покупательские привычки и модели покупок. Используя БД и прогнозную аналитику, они создали маркетинговую машину и создали чрезвычайно успешную бизнес-модель.

С ростом вычислительных мощностей, надежной инфраструктуры данных, быстрой разработки алгоритмов и необходимостью получать более полное представление о постоянно растущих объемах данных предприятия стремятся использовать аналитику БД как часть своего процесса принятия решений. Лица, принимающие решения, осознали, что при лучшем понимании может быть достигнута превосходная конкурентная позиция [75].

Организации из разных областей инвестируют в приложения БД для изучения больших наборов данных, чтобы выявить все скрытые шаблоны, неизвестные корреляции, тенденции рынка, предпочтения клиентов и другую полезную информацию. Область применения приложений БД включают в себя: производство, здравоохранение, медиа, Internet of Things правительство и т.д. [76].

2.2 Методы и алгоритмы, используемые в области аналитики медицинских данных

Медицинская обработка изображений с точки зрения цифровых данных дает важную информацию об анатомии и функциях органов в дополнение к выявлению состояний заболеваний. Поскольку размер и размерность данных увеличиваются, понимание зависимостей между данными и разработка

эффективных, точных и вычислительно эффективных методов требуют новых автоматизированных методов и платформ. Стремительный рост числа медицинских организаций, а также числа пациентов привел к более широкому использованию автоматизированных систем медицинской диагностики и поддержки принятия решений в клинических условиях. Многие области здравоохранения, такие как диагностика, прогноз и скрининг, могут быть улучшены с помощью использования ИТ. В данном исследовании обсуждаются последние разработки, которые направлены на использование больших объемов медицинских данных при объединении мультимодальных данных из разнородных источников [77]. Уровень данных, создаваемый в системах здравоохранения, не является тривиальным. На сегодняшний день отрасль здравоохранения отстала в использовании БД из-за ограниченной способности стандартизировать и консолидировать данные. Однако, теперь аналитика БД улучшила здравоохранение, предоставляя персонализированные лекарства и предписывающую аналитику. Актуальность сбора данных для прогностических целей очень важна, в основном чтобы увидеть, какие виды лечения более эффективны для конкретных состояний, для определения закономерностей, связанные с побочными эффектами лекарств, и получить другую важную информацию, которая может помочь пациентам и снизить затраты. Общая структура генерируемых данных приведена в соответствии с рисунком 2.1.



Рисунок 2.1 – Структура генерируемых данных в здравоохранении [78]

С увеличением роста использования ИТ в сфере здравоохранения объем данных растет быстрыми темпами в геометрической прогрессии. Медицинские данные собираются с различных устройств. Например, датчики, которые считывают параметры с пациентов. Также многие данные содержат различные фото изображения, истории болезней, медицинские записи и т.д. и получается огромный массив данных, который нуждается в

обработке. Назначение цели исследования данной диссертации - это разработка ИС, для диагностики СД. Главная цель диагностики СД – это точность и сто процентная точность диагноза. Для точной диагностики существуют различные виды определения типа СД, т.е. по первичным симптомам которые помогают определить тип СД. Медицинские анализы являются более достоверными в определении типа СД.

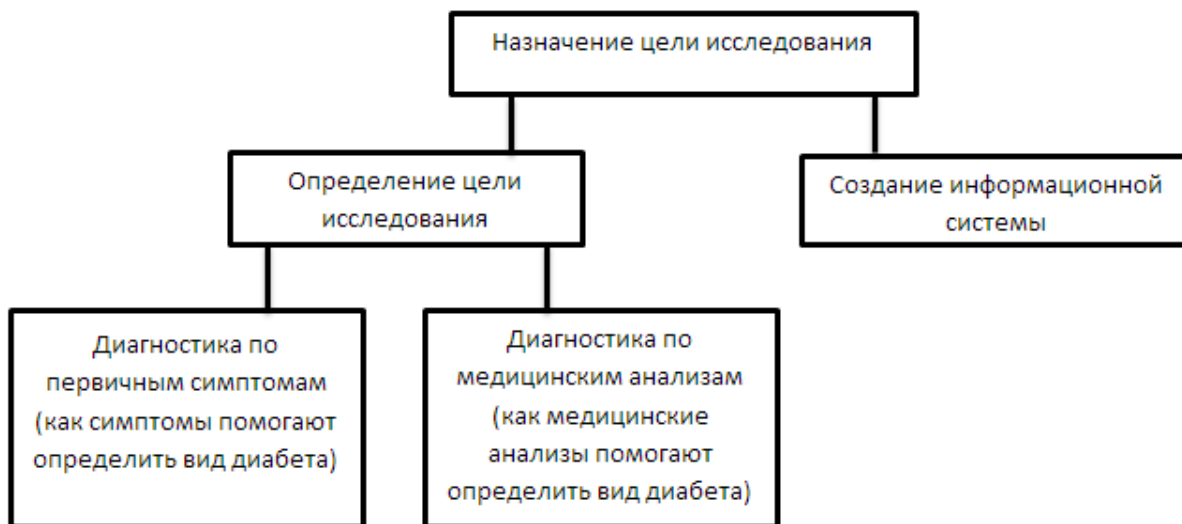


Рисунок 2.2 – Общая структура назначения цели исследования [78, с. 201]

На сегодняшний день медицинский сектор обслуживания пациентов производит большое количество больших массивов данных. Их количество увеличивается ежедневно. Информация хранится в базах данных и в целом не получают заслуженной обработки для применения этих данных в диагностировании различных болезней и хранении истории болезней не для следования стандартных типовых процедур хранения информации, а для самообучения системы при диагностировании болезней. Сбор данных из проверенных медицинских источников и интеграция этих данных в рабочий процесс в дальнейшем. Еще важным моментом является то, что данные будут собираться из различных источников в соответствии с рисунком 2.3.



Рисунок 2.3 – Общая структура сбора данных [78, с. 202]

Вид, в котором данные записываются и хранятся медицинским персоналом, является важным этапом в подготовки этих данных для вторичных целей. Подготовка данных являются важным этапом. Клинические данные, хранящиеся в системах «Электронная медицинская карта», обычно являются структурированными или неструктурированными:

1) структурированные данные соответствуют заданной модели данных и набору значений, ограничивая пользователей возможностью только вводить или выбирать предварительно определенные значения. Компьютеры могут легко обрабатывать структурированные данные, которые они часто хранят в базах данных. Данные, отправляемые медицинскими устройствами в ЕНР, обычно отправляют структурированные данные;

2) неструктурированные данные (также называемые «свободным текстом») не следуют заранее заданному набору значений, что позволяет пользователям вместо этого вводить повествовательную информацию о данных, используя свои собственные слова. Это средство записи данных предоставляет пользователю наибольшую свободу для записи, но поскольку одно и то же клиническое событие может быть задокументировано множеством способов, компьютеры не могут легко обрабатывать неструктурированные данные, что повышает вероятность ошибок.

Большая проблема в области здравоохранения состоит в том, что около 80% медицинских данных остаются неструктурированными и неиспользованными после их создания. Например, текст, изображение, сигнал и т. д. [79]. Такие типы данных трудно поддаются обработке и не могут быть использованы в медицинских ИС. Во многих медицинских организациях эти данные не используются и могут храниться в хранилищах в течение длительного времени [80]. Следующим этапом является подготовка данных, где данные будут загружаться в базу данных посредством API.



Рисунок 2.4 – Общая структура подготовки данных [78, с. 203]

Как видно из рисунка выше, собранная информация должна быть преобразована для извлечения из нее нужной информации. Для преобразования информации из неструктурированного вида в структурированный вид необходимо использовать инструменты для работы с данными различных форматов.

Первоначальной задачей разрабатываемой ИС является диагностика СД всех видов по симптомам и по анализам. Во-первых, необходимо составить список ключевых слов, которые соответствуют диагностике вида диабета. Во-вторых, необходимо ввести в базу данных перечень анализов и снимков для диагностики СД по анализам.

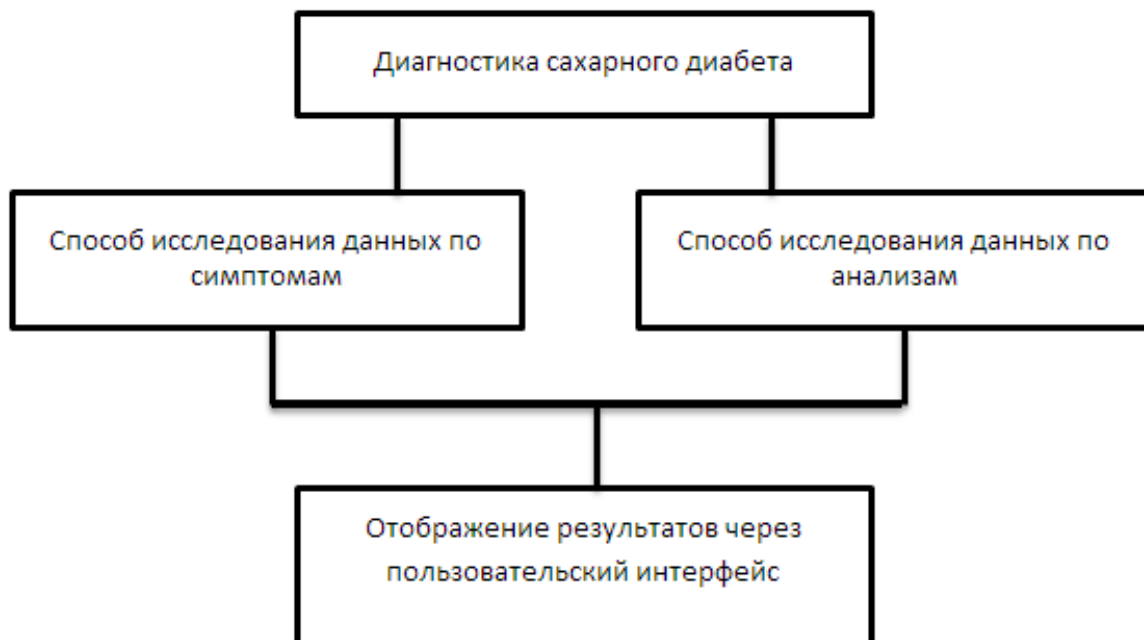


Рисунок 2.5 – Общая структура диагностики СД

После введения всех параметров результаты диагностики должны отображаться в пользовательском интерфейсе или автоматически сохранены в истории болезни пациента. При необходимости эти данные могут быть извлечены врачом для будущих сравнительных операции анализов пациентов.

2.3 Применение информационных технологии для решения медицинских задач

В соответствии с задачами данного исследования был осуществлен анализ рынка примененных ИТ для решения медицинских задач. Был осуществлен расширенный обзор применения технологии в решении медицинских задач, где лидирующие позиции занимает США. Разработанные медицинские ИС способны решать поставленные задачи. Многие исследования проводились на базе университетов. Ниже были рассмотрены и перечислены разработки ИС в сфере здравоохранения и с применением технологии BigData:

Опыт Массачусетского госпиталя. Согласно опубликованным исследованиям Массачусетский главный госпиталь разработал аналитическую систему QPID. Эта система предназначена помогать врачам, чтобы они не упускали важной информации о пациенте во время

прохождения лечения. В этом же исследовании можно увидеть непосредственное применение BigData в здравоохранении – это прогнозирование хирургического риска. Разработанная система способна выполнять автоматический поиск протоколов лечения, затем выводит на экран результаты с посчитанным красным, желтым или зеленым индикатором риска» [81].

Опыт Калифорнийского университета. Исследователями Калифорнийского университета в Дэйвисе был разработан алгоритм [82], в котором используются данные с электронных медицинских карт. Главная задача этого алгоритма – это помочь врачам выявить первые признаки сепсиса. Заражение крови в 40% случаев приводит к летальным исходам, так как симптомы на ранних стадиях заболевания обнаружить непросто.

Опыт университета Калифорнии. Исследование выявило прогностические модели сердечных заболеваний, выявление причин фибрилляции предсердий, сокращение госпитализации при сердечной недостаточности и определение влияния социальных сетей на здоровье сердца путем анализа до 1 миллиона участников в течение десяти лет. В данном исследовании были применены большие данные [83].

Опыт университета Джона Хопкинса. Согласно опубликованному исследованию, в результате экспериментов была создана база данных «Oncospace». В данной базе хранятся 3D-снимки опухолей тысяч пациентов страдающих от рака головы, шеи, предстательной и поджелудочной железы.

Истории болезней данных пациентов включают исчерпывающую информацию об особенностях аномалии, количестве полученной радиации в ходе терапии, сопутствующие заболевания, зафиксированные побочные эффекты, а также исход болезни. В ежедневной практике врачи онкологи изучают 3D-снимки опухолей и, основываясь на своих знаниях и на опыте своих коллег назначают лечение. Возможности базы данных «Oncospace» позволяют врачам получить доступ к историям болезней похожих на их случаи и рассчитать эффективность разных курсов лечения [84].

Опираясь на проведенный анализ, можно утверждать, что глобальный рынок систем персонального мониторинга и диагностики хронических заболеваний начал развиваться сравнительно недавно – согласно отчету американской Transparency Market Research [85], в 2012 году его объем составлял \$650 млн.

Темпы ежегодного роста рынка достигают 43,3% и в 2019-м он превысит \$8 млрд., говорится в исследовании. Согласно, их отчету в сфере мониторинга лидирующие позиции занимали кардиологические устройства, затем устройства контроля уровня глюкозы вышли в лидеры продаж. Устройства, не требующие постоянного прокола пальцев для тестирования крови, облегчают жизнь пациенту, чтобы чаще проверять уровень глюкозы и в случае его понижения или повышения принять соответствующие меры.

Это позволяет предотвратить случаи гипогликемии и гипергликемии, отодвигает развитие осложнений [86].

Выводы по второму разделу

1. На основе проведенного анализа применения БД в различных направлениях позволило обнаружить все скрытые шаблоны, неизвестные корреляции и другую полезную информацию. Таким образом, аналитические результаты привели к более эффективному маркетингу, качественному обслуживанию, повышению эффективности работы, конкурентным преимуществам в данном исследовании.

2. В результате были рассмотрены методы и алгоритмы, применяемые в области аналитики медицинских данных. Было выявлено, что быстро развивающаяся область большого анализа данных начинает играть ключевую роль в развитии практики в области здравоохранения и научных исследований. Тем самым были определены инструменты для накопления, управления, анализа и усвоения больших объемов разнородных, структурированных и неструктурированных данных, созданных с помощью современных систем здравоохранения.

3. В соответствии с задачами данного исследования был осуществлен анализ рынка используемых ИТ для решения медицинских задач.

4. Основные результаты главы опубликованы в научных работах [87-90].

3 МОДЕЛИ И МЕТОДЫ ИССЛЕДОВАНИЯ ПРОГНОЗИРОВАНИЯ РОСТА ПАЦИЕНТОВ И АНАЛИЗ ДАННЫХ

Разработанная ИС «Diabetes.kz» состоит из двух подсистем:

1. первая подсистема «Прогнозирование роста пациентов» связана с технико-экономическим планированием, где она отражает моделирование затратной части;

2. вторая подсистема «Определение статуса и сопровождение лечения» исследует вопросы определения статуса и сопровождения лечения пациента, где в перспективе требуются большие вычислительные нагрузки.

Общая структура ИС «Diabetes.kz» приведена на рисунке 3.1.

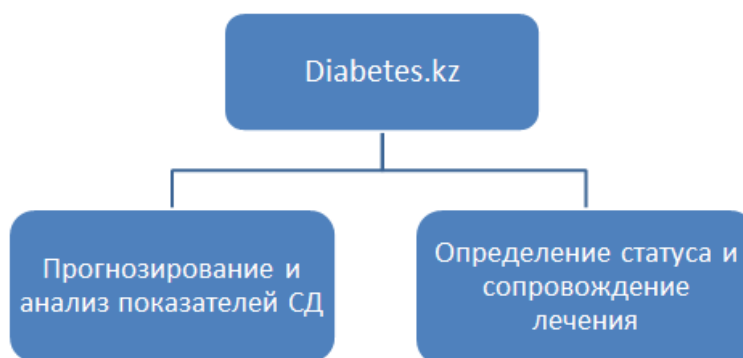


Рисунок 3.1 – Общая структура ИС «Diabetes.kz»

Логическая связь между подсистемами существует, так как лечение и сопровождение пациента напрямую влияет на результаты прогнозирования.

3.1 Модели прогнозирования статистических данных на базе методов регрессионного анализа

Комплексное использование алгоритмов и методов статистического поиска закономерностей и построения на их основе требуемых прогнозов. В данном исследовании были применены статистические данные регистра больных СД за последние пятнадцать лет по РК. Для проведения экспериментальных исследований данные по больными СД были предоставлены Общественным фондом «Казахстанское общество по изучению диабета» г. Алматы. На сегодняшний день статистические методы прогнозирования охватывают разработку, изучение и применение современных математико-статистических методов прогнозирования на основе объективных данных [91]. Обычно под прогнозированием понимается процесс предсказания будущего основанное на некоторых данных из прошлого, т.е. изучается развитие интересующего явления во времени [92]. Данные по валовому внутреннему продукту (ВВП), валовому региональному продукту (ВРП) и населению Казахстана взяты с официального сайта статистического агентства РК [93]. Целью исследования является построение моделей прогнозирования роста пациентов СД по РК с помощью методов

регрессионного анализа на фоне пассивного выявления. Выявить наиболее точный экспериментальный метод прогнозирования диабета. Для объекта исследования были использованы данные больных СД по РК с 2004 г. по 2018 г. В таблице 3.1 указывается количество больных диабетом по РК с 2004 г. по 2018г., где количество больных увеличилось с 114 тыс. человек до 326 тыс. (185.46%).

Стандартным инструментом в медицинских исследованиях сегодня являются модели регрессии [94]. Регрессионный анализ представляет собой набор статистических методов, которые служат основой для выведения выводов о взаимоотношениях между взаимосвязанными переменными и возможно, является наиболее используемым из всех методов анализа данных, поскольку эти методы применимы практически во всех областях исследований [95]. Применение методов на базе регрессионного анализа для построения прогностических моделей будут успешны, если мы имеем связь между двумя переменными. После проведенного анализа было выявлено, что существует связь между ростом ВРП и ростом больных СД. Также после проведенных расчетов корреляции и определения вида этой связи мы применили три вида регрессионного анализа для прогнозирования роста пациентов СД на фоне пассивного выявления. Это – линейная регрессия, полиномиальная регрессия, экспоненциальная регрессия. С помощью указанных выше трех уравнений регрессии экспериментальным путем зная величину одного из рассматриваемых признаков с высокой вероятностью точности определить величину другого спаренного с ним признака. Вычисление коэффициентов регрессии проводилось на основе построения линии регрессии, максимально приближенной к фактическому распределению признаков. Использовались различные виды линий тренда – линейная, полиномиальная трех степеней, и экспоненциальная. В каждом конкретном случае выбирался лучший вариант регрессионного уравнения, определяемый по величине критерия точности аппроксимации R^2 [96].

Статистические методы для изучения распределения переменной ответа по отношению к одной или нескольким объясняющим переменным известны как методы регрессии [97]. Задача регрессионного анализа заключается в построении математической модели, которая позволяет давать оценку значений зависимой переменной по значениям независимых переменных [98]. Многие методы были разработаны для определения различных параметрических отношений между переменной отклика и независимыми переменными. Эти методы обычно зависят от формы функции параметрической регрессии и распределения члена ошибки в регрессионной модели. Например, линейная регрессия, логистическая регрессия, пуассоновская регрессия, пробит-регрессия [99]. В данной схеме рассматривается применения трех методов регрессионного анализа для ситуации с одной зависимой и одной независимой переменной. Особое внимание уделяется проверке соблюдения необходимых условия для применения методов. При наличии линейной зависимости между двумя

переменными, можно применить методы регрессионного анализа для построения прогностических моделей. В качестве метода прогнозирования были применены методы статистического анализа. Как известно, регрессионные линейные уравнения хорошо интерпретируются и позволяют осуществить прогноз.

1. Наиболее распространенный метод – это применение линейной регрессии:

$$y = a + bx \quad (3.1.1)$$

где используется для прогнозирования от x до y . Уравнения регрессии имели такое математическое выражение в случае линейной регрессии:

$$y = 15915x + 78368, \text{ где } R^2 = 0,9804$$

Для каждого значения x уравнение дает наилучшее предсказуемое значение y , а все значения y образуют линию регрессии, которая равняется линии соответствия регрессии. В результате применения статистических данных, мы получили прогнозируемые данные на 2019 год по каждому региону страны. Полученные результаты, представлены в таблице 4, затем был построен график, где критерий достоверности аппроксимации, т.е. R^2 был выше 0,9.

Таблица 3.1 – Прогнозируемое количество пациентов с использованием линейной регрессии

№	Регионы	2019
1	Акмолинская область	16110
2	Актюбинская область	14107
3	Алматинская область	31222
4	Атырауская область	8882
5	Западно-Казахстанская область	9446
6	Жамбылская область	17172
7	Карагандинская область	31500
8	Костанайская область	22444
9	Кызылординская область	10453
10	Мангыстауская область	9399
11	Южно-Казахстанская область	39085
12	Павлодарская область	18612
13	Северо-Казахстанская область	17554
14	Восточно-Казахстанская область	34932
15	г. Астана	15254
16	г. Алматы	36837
17	Республика Казахстан	333010

Результаты линейной регрессии демонстрируют, что к 2019 году спрогнозировано 333010 пациентов.

2. Прогноз с использованием полиномиального регрессионного анализа:

$$y = ax^3 + bx^2 + cx^1 + d \quad (3.1.2)$$

где уравнение регрессии имели такое математическое выражение в случае полиномиальной регрессии [100-101]:

$$y = -38,378x^3 + 1487,6x^2 + 780,81x^1 + 113349, \text{ где } R^2 = 0,9964$$

Таблица 3.2 – Прогнозируемое количество пациентов с использованием полиномиальной регрессии

№	Регионы	2019
1	Акмолинская область	16187
2	Актюбинская область	15189
3	Алматинская область	30867
4	Атырауская область	9419
5	Западно-Казахстанская область	10225
6	Жамбылская область	19725
7	Карагандинская область	31569
8	Костанайская область	25198
9	Кызылординская область	11852
10	Мангыстауская область	10609
11	Южно-Казахстанская область	38867
12	Павлодарская область	20445
13	Северо-Казахстанская область	17137
14	Восточно-Казахстанская область	35993
15	г. Астана	16240
16	г. Алматы	40552
17	Республика Казахстан	350074

Результаты полиномиальной регрессии демонстрируют, что к 2019 году спрогнозировано 350074 пациентов.

3. Прогноз с использованием экспоненциального регрессионного анализа:

$$y = a * m^x = a * (e^{\ln(m)})^x = a * e^{x * \ln(m)} = a * e^{bx}, \text{ где } b = \ln(m) \quad (3.1.3)$$

где уравнение регрессии имели такое математическое выражение в случае экспоненциальной регрессии [102]:

$$y = 102666e^{0.0796x}, \text{ где } R^2 = 0,995$$

Таблица 3.3 – Прогнозируемое количество пациентов с использованием экспоненциальной регрессии

№	Регионы	2019
1	Акмолинская область	17956,62
2	Актюбинская область	16150,32
3	Алматинская область	36074,25
4	Атырауская область	10580
5	Западно-Казахстанская область	10817
6	Жамбылская область	18472
7	Карагандинская область	33915
8	Костанайская область	23805
9	Кызылординская область	13304
10	Мангыстауская область	11624
11	Южно-Казахстанская область	42957
12	Павлодарская область	20082
13	Северо-Казахстанская область	19236
14	Восточно-Казахстанская область	37296
15	г. Астана	17976
16	г. Алматы	39702
17	Республика Казахстан	369945,3

Результаты экспоненциальной регрессии демонстрируют, что к 2019 году спрогнозировано 369945 пациентов. Произведенные три вида уравнений регрессии имели высокий критерий достоверности аппроксимации. Следует отметить, что полиномиальный эксперимент показал более высокую точность. По РК линейная регрессия показала 333010 тыс., а полиномиальная регрессия 350074 тыс. и экспоненциальная 369945 тыс. больных. Получив эти данные, были построены графики регрессионных моделей.

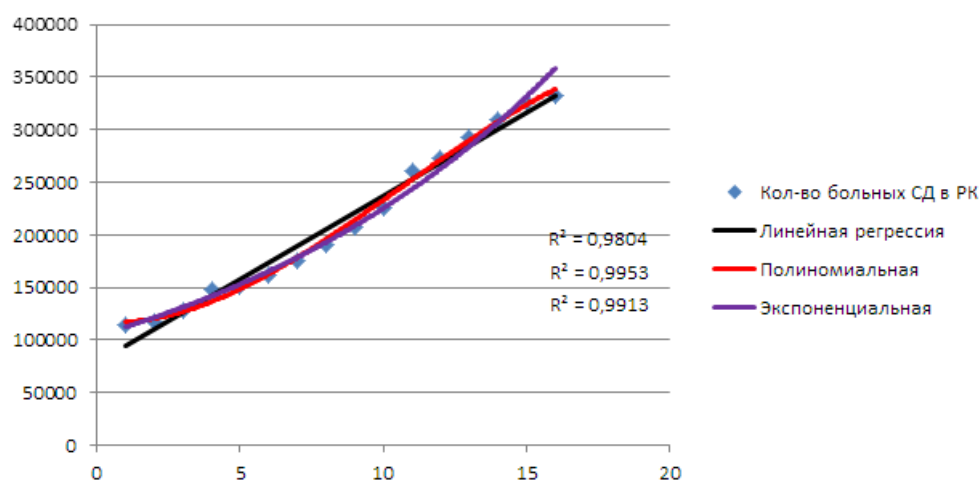


Рисунок 3.2 – График прогнозируемого количества пациентов с использованием методов регрессионного анализа [103]

Как показано на рисунке 3.2 все три вида уравнения регрессии имели высокий критерий достоверности аппроксимации, т.е. R^2 был выше 0,9. Полиномиальная регрессия показывает самый высокий коэффициент детерминации (R^2) в данной схеме.

Следующей целью данного исследования является усовершенствование экспериментальных расчетов для улучшения точности прогнозирования роста пациентов сахарным диабетом по г. Алматы и выявления факторов максимально влияющих на их рост [104].

Прогнозирование методом множественной линейной регрессии. Используя статистические данные по ВВП и ВРП на душу населения, индикаторы уровня жизни населения (ИУЖ), численность населения по г. Алматы появилась возможность получить прогнозируемые значения.

Все приведенные статистические данные о ВРП, ВВП, ИУЖ, численность населения по г. Алматы были получены из официального сайта Министерство национальной экономики РК за 2004 г. по 2018 г. Чтобы получить уравнение регрессии были применены показатели, такие как:

1) численность населения г. Алматы. В таблице 8 приведены данные роста населения за 15 лет;

2) следующим показателем был выбран ВРП – показатель, измеряющий валовую добавленную стоимость;

3) следующим показателем был выбран ВВП, макроэкономический показатель, отражающий денежную меру рыночной стоимости товаров и услуг, произведенных за определённый период времени. Согласно опубликованному исследованию, регрессионный анализ эксперимента показал, что прямые затраты на диабет тесно и положительно связаны с ВВП страны.

Таким образом, цель заключалась в выявлении новых фактических данных об экономическом воздействии диабета 2 типа, а также были исследованы влияния СД на рынок труда [105].

В данном исследовании было смоделировано экономическое бремя диабета в абсолютном виде и ВВП у лиц в возрасте 20–79 лет, используя эпидемиологические и демографические данные, а также последние прогнозы ВВП для 180 стран [106];

4) следующим показателем был выбран ИУЖ – это уровень благосостояния, комфорта, материальных благ и предметов первой необходимости, доступных им в определенной географической области, обычно в стране;

5) в качестве объекта исследования были использованы данные больных сахарным диабетом в г. Алматы.

Используя статистические данные по ВВП и ВРП на душу населения, ИУЖ, численность населения по г. Алматы появилась возможность получить прогнозируемые значения, для переменной которое показывает количество пациентов. Таким образом, y равен количеству зарегистрированных больных, x_1 – численность населения, x_2 – показатели ВВП, x_3 – показатели

ВРП, x_4 – данные ИУЖ, x_5 – года, где x_5 противоположен остальным факторам, чтобы компенсировать свободный член, т.е. с увеличением времени ожидается снижение пациентов.

Таблица 3.4 – Общее количество граждан за последние 16 лет

Год	Количество граждан в г. Алматы	ВРП	ВВП	ИУЖ	Кол-во больных СД по г. Алматы
2004	1 175 208	924	391004	12817	13616
2005	1 209 485	1219	501128	15787	15004
2006	1 247 896	1793	667212	19152	15543
2007	1 287 246	2049	829865	25226	16487
2008	1 324 739	2193	1024175	32984	17553
2009	1 361 877	2307	1056855	34282	19473
2010	1 390 610	2797	1336606	39014	19429
2011	1 413 152	3395	1705849	45918	21108
2012	1 449 366	3908	1847085	51860	23776
2013	1 474 849	4779	2113205	56453	25362
2014	1 506 899	5021	2294830	62271	29228
2015	1 641 444	5440	2330360	67321	29037
2016	1 702 766	6139	2639710	76575	32382
2017	1 751 308	6506	2943893	83053	35604
2018	1 801 993	6729	3261768	92703	37720

Приведенные в таблице 3.4 параметры уравнения, также могут определяться с применением метода наименьших квадратов (МНК), где используются параметры модели, в которых сумма квадратов отклонений фактических значений результативного признака минимизируется от теоретических, полученных по выбранному уравнению регрессии:

$$S = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - a_0 - a_1x_{1i} - a_2x_{2i} - \dots - a_mx_{mi})^2 \rightarrow \min \quad (3.1.4)$$

Рассматривая S в качестве функции параметров a_i и проводя математические преобразования, получаем систему нормальных уравнений с m неизвестными по числу параметров a_i .

$$\begin{cases} \sum y = na_0 + a_1 \sum x_1 + a_2 \sum x_2 + \dots + a_m \sum x_m, \\ \sum yx_1 = a_0 \sum x_1 + a_1 \sum x_1^2 + a_2 \sum x_2x_1 + \dots + a_m \sum x_mx_1, \\ \dots \\ \sum yx_m = a_0 \sum x_m + a_1 \sum x_1x_m + a_2 \sum x_2x_m + \dots + a_m \sum x_m^2. \end{cases} \quad (3.1.5)$$

Здесь n – число наблюдений, m – число факторов в уравнении регрессии, где решение системы позволяет получить значения параметров регрессии a_i . Для анализа общего качества уравнения линейной многофакторной регрессии используют множественный коэффициент детерминации R^2 , который рассчитывается по формуле [107]:

$$R^2 = \frac{\sigma^2}{\sigma_y^2} \quad (3.1.6)$$

где σ_y^2 – является дисперсией случайной величины y ; σ^2 – условная по x факторам дисперсия зависимой переменной, которая определяет долю вариации результативного признака, обусловленную изменением факторных признаков, входящих в многофакторную регрессионную модель. Уравнение множественной линейной регрессии рассчитывается по формуле [108]:

$$\hat{y} = a_0 + a_1x_1 + a_2x_2 + \dots + a_mx_m \quad (3.1.7)$$

где \hat{y} – теоретические значения результативного признака, полученные путем подстановки соответствующих значений факторных признаков в уравнение регрессии x_1, x_2, \dots, x_m – значения факторных признаков; a_1, a_2, \dots, a_m – коэффициенты регрессии. Для расчета параметров уравнения регрессии задаются параметры входного интервала Y , которые содержат данные по результативному признаку и параметры X содержащие факторные признаки. Данная модель получается путем МНК, где уравнение приняло вид:

$$\hat{y} = 2285283,22 + 0,0013 \cdot x_1 + 2,420 \cdot x_2 - 0,001 \cdot x_3 + 0,3683 \cdot x_4 - 1137,51 \cdot x_5$$

(0,016) (1,804) (0,007) (0,321) (823,705)

Приведенные стандартные ошибки соизмеримы с коэффициентами. Таким образом, стандартные ошибки, из данного уравнения не превышают значения коэффициентов. Показатель x_2 демонстрирует, что с ростом ВРП увеличивается рост больных и это доказывает тенденцию, что люди начали употреблять жиры, белки, сахара и т.д. На данном эксперименте полученное прогнозируемое значение 37184 единиц, он отличается от фактического на 535 единицы. Из уравнения видно, что с ростом численности населения количество пациентов вырастает на 0.001тыс. пациентов. С ростом ВРП на душу населения количество пациентов вырастает на 2,420 тыс. пациентов. А вот рост ВВП на душу населения снижает количество пациентов на 0,001 тыс. пациентов. Увеличение индикаторов уровня жизни населения влечет за собой рост количества пациентов на 0,3683 тыс. пациентов. Уравнение приняло вид:

$$\hat{y} = 2551826.23 + 0.0021x_1 + 2.17x_2 - 0.001x_3 + 0.38x_4 - 1271.07x_5$$

Полученные стандартные ошибки соизмеримы с коэффициентами, где прогнозируемое число пациентов СД на 2019 г. равняется 37074,98.

Проверка качества уравнения множественной регрессии на основе статистических критериев:

3. Критерий Фишера. Проверка качества выполняется проверкой гипотезы о равенстве нулю коэффициент детерминации рассчитанного по данным генеральной совокупности: R^2 или $b_1 = b_2 = \dots = b_m = 0$. Для проверки данного утверждения используют F-критерий Фишера, т.е. вычисляется фактическое значение F-критерия, через коэффициент детерминации R^2 , рассчитанный по данным наблюдения. По таблицам распределения Фишера-Снедекора находят критическое значение F-критерия ($F_{кр}$). Также задается коэффициент значимости α равный 0,05 и двумя числами степеней свободы $k_1=m$ и $k_2=n-m-1$.

$$R^2 = 1 - \frac{S_e^2}{\sum(y_i - \bar{y})^2} = \frac{9349044.109}{850773933.73} = 0.989$$

Проверяется гипотеза об общей значимости, т.е. гипотезу об одновременном равенстве нулю всех коэффициентов регрессии при объясняющих переменных: $H_0: R^2 = 0; \beta_1 = \beta_2 = \dots = \beta_m = 0; H_1: R^2 \neq 0$. Проверка этой гипотезы осуществляется с помощью F-статистики распределения Фишера. Если $F < F_{кр}$, то нет оснований для отклонения гипотезы H_0 .

$$F = \frac{R^2}{1-R^2} \cdot \frac{(n-m-1)}{m} = \frac{0.989}{1-0.989} \cdot \frac{15-5-1}{5} = 162.002$$

где табличное значение при степенях свободы $k_1 = 5$ и $k_2 = n-m-1 = 15 - 5 - 1 = 9$, $F_{кр}(5;9) = 3.48$. Поскольку фактическое значение $F > F_{кр}$, то коэффициент детерминации статистически значим и уравнение регрессии статистически надежно [109]. Получено подтверждение, что качество регрессионного уравнения по критерию Фишеру удовлетворено, т.е. R^2 выполняется, что свидетельствует о том, что выбранные критерии анализа.

2. t-критерий Стьюдента. В медицинских исследованиях этот метод является наиболее известным статистическим критерием, который используется для определения статистической значимости различий в средних значениях. Проводится проверка критического значения t-критерия Стьюдента при значимости 0,05 и числе степеней свободы $15 - 5 - 1 = 9$ равно 2,26 [110].

$$t_{крит}(n - m - 1; \alpha/2) = (0,05; 15 - 5 - 1) = 2,26$$

Доверительный интервал прогноза: $y_{min} = 37074,98 - 2,26 \cdot 1159,94 = 34451$, $y_{max} = 37074,98 + 2,26 \cdot 1159,94 = 39698,96$. Построим график, приведенный на рисунке 3.3.

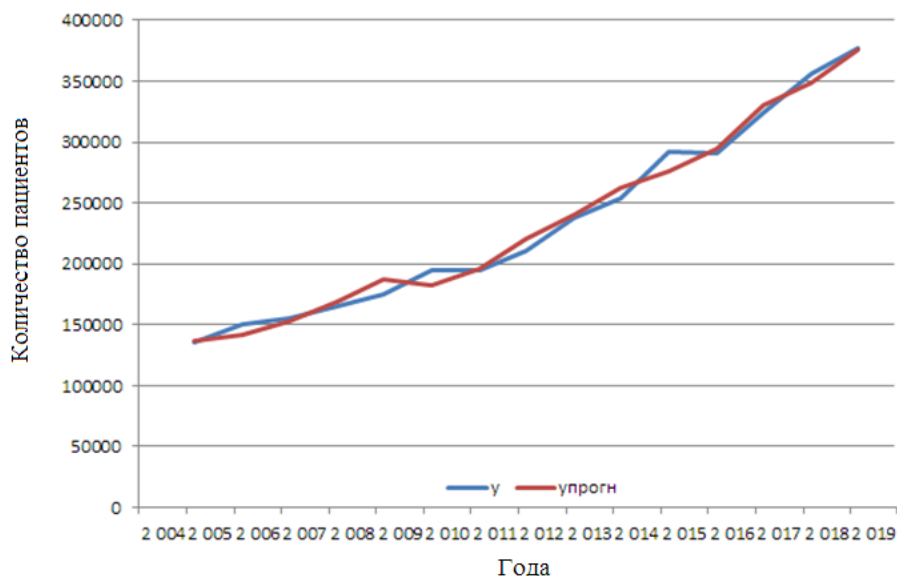


Рисунок 3.3 – График прогнозируемого роста пациентов.

Можно видеть, что фактические значения близки к предсказанным, то есть модель может использоваться для прогнозирования. Для того чтобы регрессионный анализ, основанный на МНК, давал наилучшие из всех возможных результаты, должны выполняться определённые условия Гаусса-Маркова:

1. Проверка выполнимости предпосылки МНК «Математическое ожидание случайного отклонения ε_i равно нулю для всех наблюдений»: строится график остатков по наблюдениям с добавлением линейного тренда, если тренд близок к оси абсцисс, то эта предпосылка выполняется и делается вывод, что случайное отклонение в среднем не оказывает влияния на зависимую переменную [111].

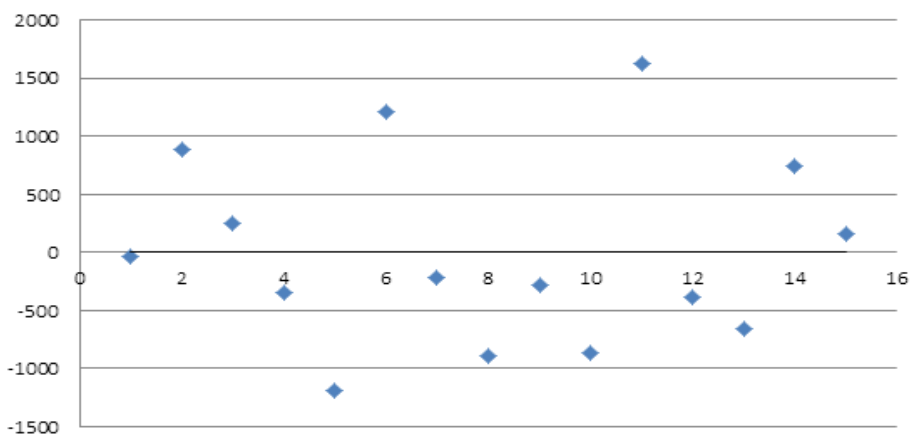


Рисунок 3.4 – График остатков наблюдения

2. В данном случае предпосылка выполняется, так как на графике получена горизонтальная полоса, то остатки ε_i представляют собой случайные величины и МНК оправдан, теоретические значения \hat{y} хорошо аппроксимируют фактические значения [112].

3. Проверка выполнимости предпосылки МНК «Отсутствие мультиколлинеарности». В качестве критерия мультиколлинеарности может быть принято соблюдение следующих неравенств:

$$r(x_j y) > r(x_k x_j); r(x_j y) > r(x_k x_j) \quad (3.1.8)$$

Если одно из неравенств не соблюдается, то исключается тот параметр x_k или x_j , связь которого с результативным показателем Y оказывается наименее тесной. В нашем случае все парные коэффициенты корреляции $|r| < 0.7$, что говорит об отсутствии мультиколлинеарности факторов. Построим корреляционную матрицу между факторными переменными [113].

Таблица 3.5 – Результаты корреляционной матрицы

	x1	x2	x3	x4	x5
x1	1				
x2	0,981066	1			
x3	0,979358	0,993367	1		
x4	0,990173	0,991178	0,996753	1	
x5	0,981631	0,990823	0,99383	0,994211	1

Коллинеарные факторы x_1, x_2, x_3, x_4 и x_5 . Так как коэффициенты межфакторной корреляции больше 0,7; можно сделать вывод, что все переменные коррелированы между собой. Между объясняющими переменными отсутствует сильная линейная зависимость.

4. Проверка выполнимости очередной предпосылки МНК «Модель является линейной относительно параметров»: на основе результатов этапа спецификации, так как специфицирована модель линейная относительно параметров, то данная предпосылка выполняется, т.е. модель является линейной относительно параметров [114].

5. Проверка нормальности распределения остаточной компоненты. Расчетное значение RS-критерия равно:

$$RS = \frac{\varepsilon_{max} - \varepsilon_{min}}{S_\varepsilon} \quad (3.1.9)$$

где $\varepsilon_{max} = 1627.4604$ - максимальное значение остатков, $\varepsilon_{min} = -1185.7661$ - минимальный уровень ряда остатков. S_ε – среднеквадратическое отклонение.

Таблица 3.6 – Проверка нормальности распределения остаточной компоненты

y	$y(x)$	$e_i = y-y(x)$	e^2
13616	13646.2557	-30.2557	915.4074
15004	14117.561	886.439	785774.1007
15543	15298.0211	244.9789	60014.6614
16487	16834.4161	-347.4161	120697.9465
17553	18738.7661	-1185.7661	1406041.2439
19473	18259.6361	1213.3639	1472251.9538
19429	19648.3854	-219.3854	48129.9537
21108	22002.0952	-894.0952	799406.2267
23776	24059.5098	-283.5098	80377.8067
25362	26227.3021	-865.3021	748747.7243
29228	27600.5396	1627.4604	2648627.3536
29037	29427.7448	-390.7448	152681.4987
32382	33045.0001	-663.0001	439569.1326
35604	34855.1943	748.8057	560709.9764
37720	37561.572	158.428	25099.4312
			9349044.4176

Несмещенная оценка среднеквадратического отклонения.

$$S_e = \sqrt{\frac{\sum e^2}{n-1}} = \sqrt{\frac{9349044,418}{15-1}} = 817,184$$

$$RS = \frac{1627,46 - (-1185,766)}{817,184} = 3,443$$

Расчетное значение RS-критерия попадает в интервал (2,7-3,7), следовательно, выполняется свойство нормального распределения. Таким образом, модель адекватна по нормальности распределения остаточной компоненты [115]. Таким образом, исходя из полученных значений, можно прийти к выводу, что при имеющихся статистических данных факторов ВРП, ВВП, ИУЖ и численности населения, можно спрогнозировать количество больных СД. Это свидетельствует, что выбранные факторы регрессионного анализа имеют большое значение для роста или снижения количества больных СД. В общем виде алгоритм модели прогнозирования и проверки общего качества уравнения представлен на рисунке 3.5.

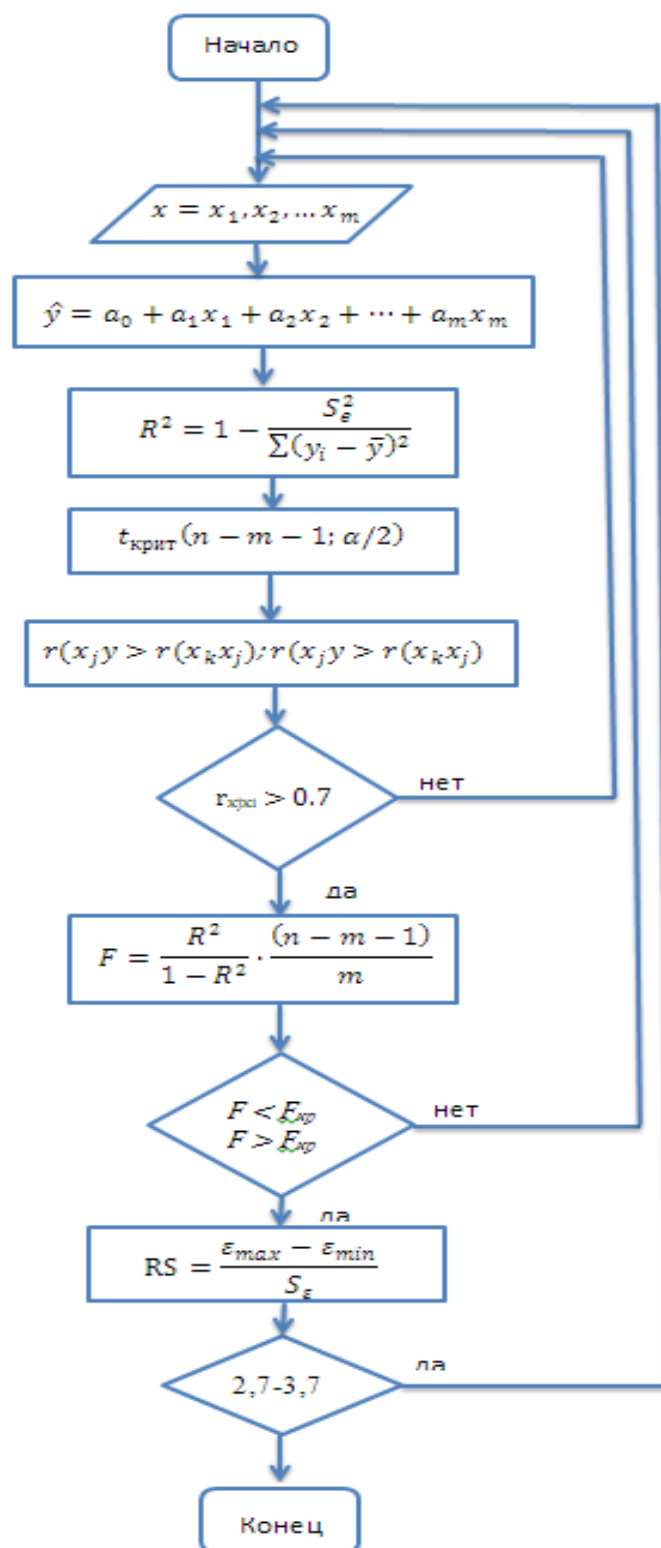


Рисунок 3.5 – Алгоритм проверки качества уравнения регрессии для прогнозирования роста пациентов

При моделировании отдельных компонентов, зная их законы распределений, можно получить прогнозные данные по количеству пациентов на будущий период. Полученные данные можно использовать для того чтобы заложить сумму для закупа медикаментов необходимых для

пациентов. В рамках исследования были получены статистические данные, которые свидетельствуют о росте распространенности СД в РК. Изучены многие работы по применению регрессионного анализа в прогностических целях. Пришли к заключению, что математические методы регрессионного анализа являются эффективными приемами для решения различных медицинских задач. Осуществили прогноз на 2019 год с использованием трех различных моделей предсказания регрессионного анализа. Все эти эксперименты показали увеличение роста больных, что вызывает большую обеспокоенность среди нас. Также произвели корреляцию и выявили взаимосвязь между ростом населения и количеством больных СД и увеличением ВРП и ростом больных СД.

После обработки полученных статистических данных и после проведения экспериментальных исследований можно прийти к выводу, что распространенность СД в РК растет. В очередной раз были убеждены, что для выполнения прогностических задач в медицине применение регрессионного анализа показывает высокий критерий достоверности. Также исходя из полученных результатов, можно сделать заключение, что важным вектором в предотвращении СД является раннее диагностирование болезни, и проведение профилактических мер по повышению осведомленности риска и осложнениях диабета у населения.

3.2 Получение модели регрессионного анализа на основе алгоритмов машинного обучения

Для реализации алгоритмов машинного обучения используется один из самых популярных языков программирования - Python. Python имеет хорошо документированную библиотеку `scikit-learn`, специализирующуюся на машинном обучении. Библиотека `scikit-learn` помогает решать поставленные задачи, такие как кластеризация, перекрестная проверка, наборы данных, уменьшение размеров, алгоритмические композиции, извлечение признаков, выбор признаков, оптимизация параметров алгоритма, множественное обучение.

Для более наглядной демонстрации работы с библиотекой рассмотрим пример распознавания количества пациентов с диабетом на основе данных, полученных из регистра статистических данных РК. Прежде всего, прежде чем начать работу по решению проблемы, необходима загрузка данных. Библиотека `scikit-learn` ориентирована на моделирование данных, а не на загрузку файла. Для загрузки данных лучше использовать библиотеку `Pandas`, поскольку она имеет удобную функциональность для ввода-вывода и обработки табличных данных, а также позволяет проводить первичный анализ данных. Код загрузки данных:

```
data = pd.read_csv('registr_bolnyh', header=None, na_values='?')  
X = data.drop([0], axis=1)  
Y = data[0]
```

Важным моментом является работа с массивами, где X - знаки, а Y - классы, где необходимым шагом является нормализация функций, поскольку

большинство алгоритмов машинного обучения основаны на градиентных методах. После загрузки необходимых данных можно использовать возможности алгоритмов машинного обучения. Библиотека `scikit-learn` реализует множество алгоритмов, таких как: логистическая регрессия, линейная регрессия, наивный байесовский метод, *k*-ближайшие соседи, деревья решений, метод опорных векторов и т. д. Библиотека `scikit-learn` может быть легко интегрирована в приложения за пределами традиционного диапазона статистического анализа данных, поскольку он опирается на научную экосистему Python.

Алгоритмы, реализованные на языке высокого уровня, могут использоваться в качестве строительных блоков для подходов, специфичных для случая использования, например, в медицинской визуализации [116]. Рассмотрим первый эксперимент с использованием алгоритма линейной регрессии:

```
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
x = x[:, np.newaxis]
y = y[:, np.newaxis]
model = LinearRegression()
model.fit(x, y)
y_pred = model.predict(x)
plt.scatter(x, y, s=10)
plt.plot(x, y_pred, color='r')
plt.show()
```

После обучения модели легко спрогнозировать количество пациентов в соответствии с входным атрибутом, используя метод прогнозирования. Рассмотрим второй эксперимент с использованием алгоритма полиномиальной регрессии:

```
import operator
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import PolynomialFeatures
x = x[:, np.newaxis]
y = y[:, np.newaxis]
polynomial_features = PolynomialFeatures(degree=3)
x_poly = polynomial_features.fit_transform(x)
model = LinearRegression()
model.fit(x_poly, y)
y_poly_pred = model.predict(x_poly)
rmse = np.sqrt(mean_squared_error(y, y_poly_pred))
r2 = r2_score(y, y_poly_pred)
print(rmse)
print(r2)
plt.scatter(x, y, s=10)
# sort the values of x before line plot
sort_axis = operator.itemgetter(0)
sorted_zip = sorted(zip(x, y_poly_pred), key=sort_axis)
```

```
x, y_poly_pred = zip(*sorted_zip)
plt.plot(x, y_poly_pred, color='m')
plt.show()
```

Рассмотрим третий эксперимент с использованием алгоритма экспоненциальной регрессии:

```
import numpy as np
>>> x = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15])
>>> y = np.array([114355, 117563, 128039, 147717, 151336, 162012, 175685, 190682,
207935, 226202, 261453, 272629, 293171, 310114, 326449])
>>> p1 = np.polyfit(x, np.log(y), 1)
#  $y \approx \exp(-0.401) * \exp(0.105 * x) = 0.670 * \exp(0.105 * x)$ 
# (^ biased towards small values)
print (p1)
#  $y \approx \exp(1.42) * \exp(0.0601 * x) = 4.12 * \exp(0.0601 * x)$ 
# (^ notsobiased)
```

В результате экспериментов были изучены основные особенности библиотеки scikit-learn, используемой для решения задач машинного обучения. После этого шага было проведено сравнение результатов описанных в разделе 3.1, где, полученные результаты были сравнены с результатами раздела 3.2. Было установлено, что все три метода в Python и Excel дают точные результаты, только в методе экспоненциальной регрессии была небольшая разница у 16 пациентов. В экспоненциальном методе результат, полученный в Excel, был равен 369945 пациентам, тогда как в Python это число было равно 369961. Поскольку это число низкое, можно сделать вывод, что применение методов регрессионного анализа для прогнозирования является довольно хорошей идеей, поскольку приближение в приоритете. Модель регрессии была бы более надежной, если бы статистические данные о людях с диабетом в Казахстане давались по месяцам, а не по годам. Следовательно, это одно из ограничений данного исследования.

В данном разделе были представлены модели и методы исследования прогнозирования роста пациентов СД по РК, которые включены в подсистему «Прогнозирование и анализ показателей СД».

В исследовании применены реальные статистические данные и технико-экономические показатели, которые позволили выявить факторы, влияющие на рост больных. Разработанные модели, используемые в качестве алгоритмического обеспечения подсистемы «Прогнозирование и анализ показателей СД» позволит сэкономить бюджетные средства при планировании закупок инсулина и других медикаментов, таким образом, полученные модели позволят осуществлять краткосрочные прогнозы.

3.3 Анализ, обработка и визуализация больших данных

Исследовательский анализ данных относится к критическому процессу проведения первоначальных исследований данных с целью выявления проверки гипотез, закономерностей, выявления аномалий и проверки

предположений с помощью сводной статистики и графических представлений данных.

В интеллектуальном анализе данных исследовательский анализ данных – это подход к анализу наборов данных для обобщения их основных характеристик, часто с помощью визуальных инструментов. Следовательно, исследовательский анализ данных используется для того чтобы перед задачей моделирования выяснить о том какую информацию несут эти данные. От визуального просмотра колонки цифр или целой электронной таблицы маловероятно определить важные характеристики данных. Исследовательские методы анализа данных были разработаны в качестве вспомогательного инструмента в этой ситуации. Исследовательский анализ данных обычно классифицируется по двум направлениям. Во-первых, каждый метод является либо неграфическим, либо графическим. Во-вторых, каждый метод является либо одномерным, либо многомерным.

Процесс сбора, хранения и обработки медицинских данных является основной проблемой для решения актуальных задач в области здравоохранения. Необходимо использовать современные инструменты информационных технологии, чтобы эффективно применять цифровые медицинские данные для обеспечения поддержки принятия решений в этой области. Одним из таких инструментов получивший широкую известность в области обработки и анализа данных является язык программирования Python. Этому способствует простота языка, а также большое разнообразие открытых библиотек, которые доступны. В данной работе реализованы примеры исследования и классификации данных с использованием некоторых библиотек на Python.

Для исследования, понадобится выбрать интересующий набор данных (DataSet). В качестве дата сетов были использованы медицинские данные, предоставленные ОФ «Казахстанское общество по изучению диабета» в рамках целевой программы «Бремя диабета 2019-2020 для РК». Набор данных включает в себя всю информацию об диспансерных больных всех граждан РК состоящие на учете с диагнозом «Сахарный диабет» с 31.12.2009 г. по 31.12.2019 г. Для решения поставленных задач использовалось программное решение Anaconda, который включает в себя все необходимые инструменты для решения задач по анализу и обработке данных. Дистрибутив Anaconda состоит из интерпретатора Python и содержит множество различных пакетов для обработки данных, включающих в себя популярные библиотеки такие как: NumPy, Pandas, SciPy, Matplotlib. Данный инструмент можно скачать с официального сайта для установки на ОС. После установки данного продукта можно включить Anaconda navigator, и далее выбрать с каким именно продуктом будет проводиться работа. Anaconda Navigator – это графический интерфейс пользователя на рабочем столе, включенный в дистрибутив Anaconda. Также позволяет запускать приложения и легко управлять пакетами, без использования командной строки. На рисунке 3.6 указан меню Anaconda navigator.

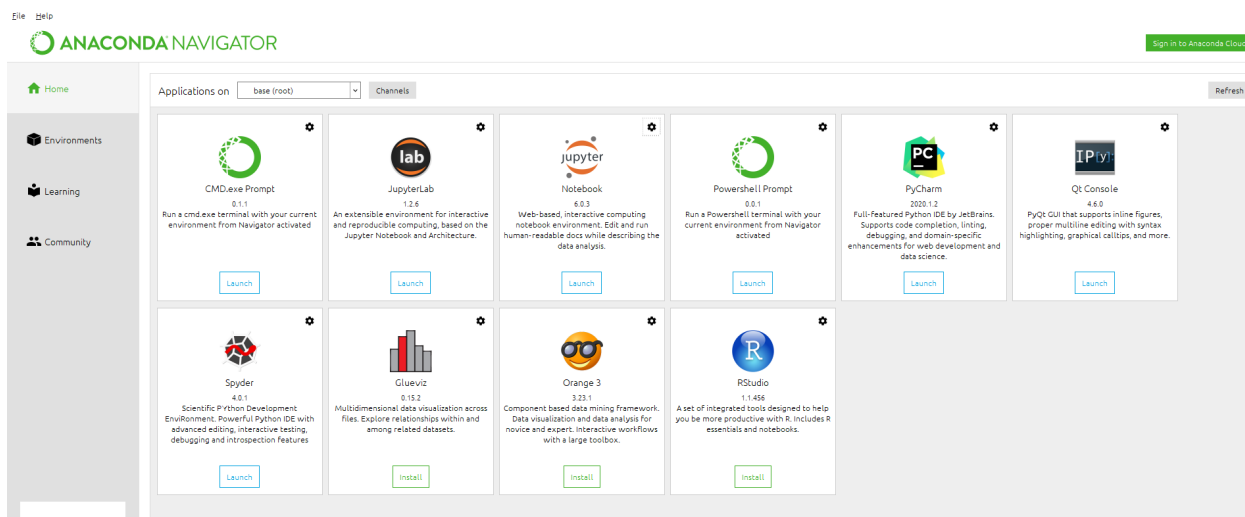


Рисунок 3.6 – Меню Anaconda navigator

Для обработки данных использовался инструмент Jupyter notebook для интерактивной обработки данных, который входит в пакет anaconda. После включения Jupyter notebook, данный инструмент откроется в браузере. И далее все необходимые манипуляции можно проводить в браузере. После успешного запуска можно увидеть правильную настроенную среду Jupyter.

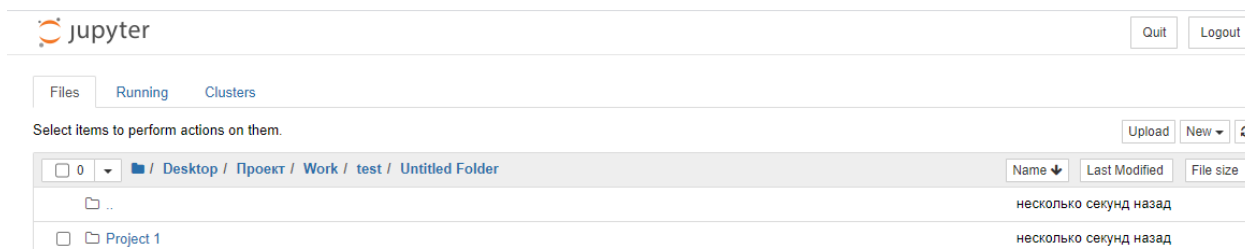


Рисунок 3.7 – Главное меню Jupyter

Следующим этапом является подготовка набора данных для загрузки, для этого необходимо реализовать анализ на подмножестве данных, используя библиотеку Pandas. Pandas позволяет обрабатывать, чистить, анализировать данные. В исследованиях данных с помощью Python также часто используют Pandas для работы с таблицами. Это библиотека для работы с таблицами данных и хорошо подходит для небольших и средних наборов данных. Также наиболее распространенным решением является использование Pandas вместе с другими решениями, такими как SQL, MongoDB, Elasticsearch. Библиотека Pandas – один из самых популярных инструментов Python для работы с данными, она поддерживает различные текстовые, бинарные и sql форматы файлов, в том числе .xlsx, .xls и .csv. Для работы с файлами Excel Pandas использует модули xlrd и xlwt. Для загрузки файлов в датасет необходимо импортировать необходимые библиотеки для исследовательского анализа данных.

```
In [1]: import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Рисунок 3.8 – Импорт библиотек

Обработка производилась над файлами в формате .xlsx, файлы были представлены в формате excel. Данные файлы поддерживаются библиотекой pandas, которая поставляется в пакете дистрибутива Anaconda, и позволяет обрабатывать данные.

на 31.12.2009.xlsx	01.07.2020 15:35	Лист Microsoft Ex...	6 048 КБ
на 31.12.2010.xlsx	02.07.2020 0:13	Лист Microsoft Ex...	7 238 КБ
на 31.12.2011.xlsx	02.07.2020 0:54	Лист Microsoft Ex...	8 453 КБ
на 31.12.2012.xlsx	02.07.2020 0:54	Лист Microsoft Ex...	9 418 КБ
на 31.12.2013.xlsx	02.07.2020 0:54	Лист Microsoft Ex...	10 489 КБ
на 31.12.2014.xlsx	02.07.2020 0:55	Лист Microsoft Ex...	11 515 КБ
на 31.12.2015.xlsx	02.07.2020 0:55	Лист Microsoft Ex...	12 394 КБ
на 31.12.2016.xlsx	02.07.2020 1:10	Лист Microsoft Ex...	13 692 КБ
на 31.12.2017.xlsx	02.07.2020 0:56	Лист Microsoft Ex...	14 711 КБ
на 31.12.2018.xlsx	02.07.2020 0:56	Лист Microsoft Ex...	15 865 КБ
на 31.12.2019.xlsx	02.07.2020 0:57	Лист Microsoft Ex...	18 289 КБ

Рисунок 3.9 – Перечень файлов об диспансерных больных граждан РК

Для загрузки файлов в датасет необходимо прописать загрузку в среде Jupyter notebook. На рисунке 3.10 указан способ загрузки данных в датасет df, который в дальнейшем позволяет обработать загруженные данные в датасет.

```
In [1]: import pandas as pd
from os import listdir

filepaths = [f for f in listdir("./test/") if f.endswith('.xlsx')]
df = pd.concat(map(pd.read_excel, [
    "./test/на 31.12.2009.xlsx", "./test/на 31.12.2010.xlsx",
    "./test/на 31.12.2011.xlsx", "./test/на 31.12.2012.xlsx", "./test/на 31.12.2013.xlsx",
    "./test/на 31.12.2014.xlsx", "./test/на 31.12.2015.xlsx", "./test/на 31.12.2016.xlsx",
    "./test/на 31.12.2017.xlsx", "./test/на 31.12.2018.xlsx", "./test/на 31.12.2019.xlsx"]]))
```

Рисунок 3.10 – Перечень файлов загруженных в датасет

Далее необходимо проверить, какое количество данных было загружено в датасет. На рисунке 3.11 приведены результаты количества строк и столбцов всех файлов, где общее количество строк превышает 2700000 записей.

```
In [47]: df.shape
Out[47]: (2705693, 8)
```

Рисунок 3.11 – Результаты подсчета всех строк и столбцов

На рисунке 3.12 можно получить информацию по столбцам, используя функцию count.

```
In [3]: df.count()
Out[3]: Регион          2705693
        МО             2703494
        Дата рождения  2705693
        ПОЛ           2705693
        национальность 2705693
        МКБ           2705693
        Дата взятия на учет 2705693
        код МО        2705693
        dtype: int64
```

Рисунок 3.12 – Подсчет строк в каждом столбце

Используя функцию head, можно посмотреть какие данные загружены в датасет. На рисунке 3.13 представлен способ использования функции head, он показывает первые 10 строк датасета.

```
In [48]: df.head(10)
Out[48]:
```

	Регион	МО	Дата рождения	ПОЛ	национальность	МКБ	Дата взятия на учет	код МО
0	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1957-05-16 00:00:00.000	жен	Казахи	E11.7	1997-01-01 00:00:00.000	03МС
1	Акмолинская область	ГОСУДАРСТВЕННОЕ КОММУНАЛЬНОЕ КАЗЕННОЕ ПРЕДПРИЯ...	1930-09-10 00:00:00.000	муж	Украинцы	E11.8	2004-01-01 00:00:00.000	04PJ
2	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасының ж...	1948-01-05 00:00:00.000	муж	Казахи	E11.8	2003-01-01 00:00:00.000	0FA5
3	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасының ж...	1933-04-05 00:00:00.000	муж	Казахи	E11.5	2006-06-20 00:00:00.000	01УК
4	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1972-08-14 00:00:00.000	муж	Русские	E11.8	1992-01-01 00:00:00.000	014A
5	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1951-09-21 00:00:00.000	жен	Немцы	E11.9	1996-07-21 00:00:00.000	08ТС
6	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1990-01-21 00:00:00.000	жен	Азербайджанцы	E10.4	2004-09-11 00:00:00.000	007L
7	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1938-05-09 00:00:00.000	муж	Казахи	E11.8	2008-05-06 00:00:00.000	007N
8	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1944-05-25 00:00:00.000	жен	Белорусы	E11.8	2008-01-01 00:00:00.000	007K
9	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1950-02-21 00:00:00.000	жен	Казахи	E11.7	2004-09-23 00:00:00.000	005B

Рисунок 3.13 – Первые 10 строк датасета.

Инструменты визуализации позволяют получить более детальный анализ при обработке данных, где предоставляется возможность обрабатывать данные различными методиками и библиотеками. Например, возможность обработки данных при помощи различных библиотек. В данном исследовании для визуализации данных использовалась библиотека matplotlib, позволяющая визуализировать поступающие данные, после обработки в Pandas. На рисунке 3.14 можно увидеть результаты обработки подсчета количества пациентов состоящих на учете с диагнозом СД по регионам РК. Также приведен подсчет количества пациентов находящихся в официальной статистике по всем регионам страны.

```
In [49]: import matplotlib.pyplot as plt
obl = df["Регион"].value_counts()
obl.plot(kind='barh', figsize = (12,8))

Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x2bbdd93b5c8>
```

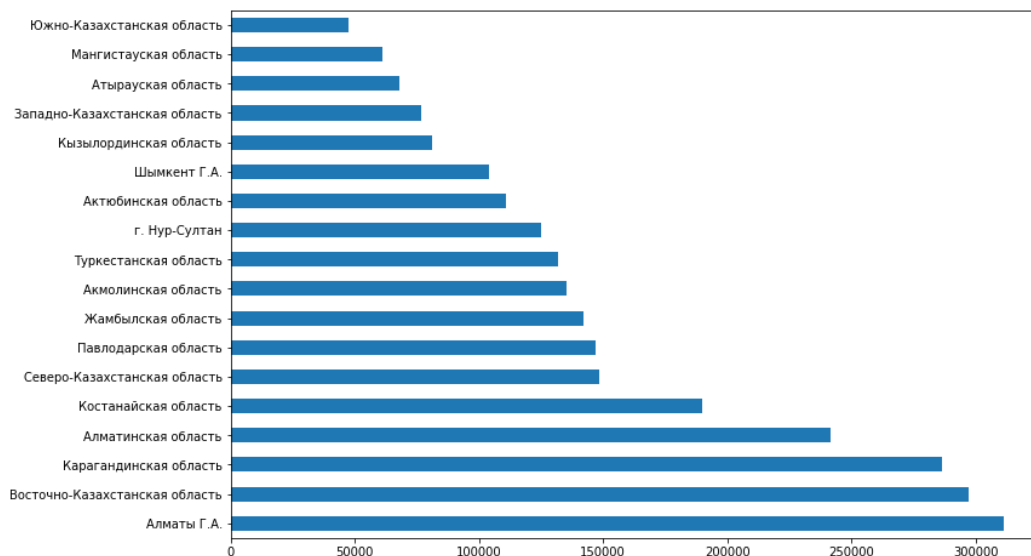


Рисунок 3.14 – Визуализация количества пациентов по регионам.

Также в результате обработки и визуализации были получены числовые значения количества пациентов с диагнозом СД по регионам РК.

```
In [43]: obl

Out[43]: Алматы Г.А. 311212
Восточно-Казахстанская область 297488
Карагандинская область 286442
Алматинская область 241780
Костанайская область 189781
Северо-Казахстанская область 148549
Павлодарская область 146734
Жамбылская область 142321
Акмолинская область 135296
Туркестанская область 131831
г. Нур-Султан 124775
Актюбинская область 110776
Шымкент Г.А. 104041
Кызылординская область 81184
Западно-Казахстанская область 76792
Атырауская область 68047
Мангистауская область 61018
Южно-Казахстанская область 47626
Name: Регион, dtype: int64
```

Рисунок 3.15 – Результаты числовых значений.

В продолжении обработки данных необходимо вывести график роста пациентов по РК. В результате обработки на рисунке 3.16 представлен график роста заболеваемости с 2009 г. по 2019 г. по РК.

```
In [39]: year = df["Год"].value_counts()
#year = year[:20]

year.plot()
```

Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x250ba993548>

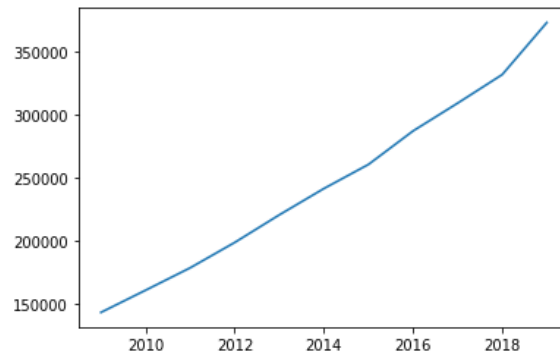


Рисунок 3.16 – График роста заболеваемости по РК.

Все файлы были обработаны и осуществлен подсчет количества всех зарегистрированных пациентов по РК с 2009 г. по 2019 г.

```
In [40]: year
```

```
Out[40]: 2019    373183
2018    331901
2017    309248
2016    287189
2015    260627
2014    241496
2013    220621
2012    198728
2011    178594
2010    160840
2009    143266
Name: Год, dtype: int64
```

Рисунок 3.17 – Числовые значения зарегистрированных пациентов по РК.

В результате обработки на рисунке 3.18 представлен график роста заболеваемости с 2009 г. по 2019 г. по г. Алматы.

```
In [44]: alm_year = akm["Год"].value_counts()
alm_year.plot()
```

Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x250be4da648>

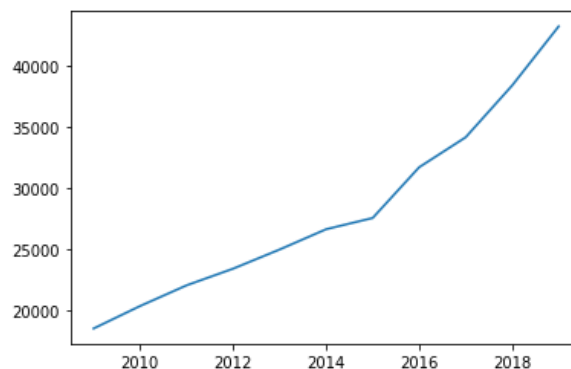


Рисунок 3.18 – График роста заболеваемости по г. Алматы.

Все файлы были обработаны и осуществлен подсчет количества всех зарегистрированных пациентов по г. Алматы с 2009 г. по 2019 г.

```
In [45]: alm_year
Out[45]: 2019    43221
         2018    38397
         2017    34166
         2016    31719
         2015    27576
         2014    26660
         2013    25004
         2012    23440
         2011    22081
         2010    20391
         2009    18557
Name: Год, dtype: int64
```

Рисунок 3.19 – Числовые значения зарегистрированных пациентов по г. Алматы.

Для более глубокого исследовательского анализа данных, используя два файла состоящие из двух частей за 2019 год, исследуем данные по регионам, полу, возрасту и т.д. Далее можно загрузить предоставленные excel файлы в DataFrame (data1 и data2).

```
In [2]: files = [f for f in os.listdir("./") if f.endswith(".xlsx")]
data1 = pd.read_excel(files[0], encoding = "cp1251", skiprows = range(0,7)).drop([0]).reset_index(drop=True)
data2 = pd.read_excel(files[1], encoding = "cp1251", skiprows = range(0,7)).drop([0]).reset_index(drop=True)
```

Рисунок 3.20 – Загрузка data1 и data2

Последняя строка data1 совпадают с первой строкой data2. Это позволяет нам объединить эти два отдельных DataFrames в один DataFrame (combined_data).

```
In [8]: combined_data = pd.concat([data1.drop([len(data1)-1]), data2]).reset_index(drop=True)
combined_data
```

Рисунок 3.21 – Объединение двух файлов

После объединения двух файлов, получаем 112626 строк и 41 столбец данных.

Out[8]:

	№ п/п	Регион, где пролечился больной	Медицинская организация, где пролечился больной	Диспансерный учет по данным ИС "ЭРДБ" с диагнозами E10-E14 (наличие в базе карты - 1, отсутствие - 0)	RpnID	Дата рождения	Дата смерти(РПН)	Возраст	Пол	Гражданство	...	Наименован операц
0	1	Ақмолинская область	Государственное коммунальное предприятие на пр...	0.0	394122278	28.01.1954	29.07.2019	65	Мужской	Казахстан	...	Ni
1	2	Ақмолинская область	Государственное коммунальное предприятие на пр...	1.0	394123525	21.04.1946	NaN	72	Женский	Казахстан	...	Ni
2	3	Ақмолинская область	Государственное коммунальное предприятие на пр...	0.0	405395196	10.12.1938	NaN	80	Женский	Казахстан	...	Ni
3	4	Ақмолинская область	Государственное коммунальное предприятие на пр...	1.0	405955008	09.03.1949	NaN	70	Женский	Казахстан	...	Ni
4	5	Ақмолинская область	Государственное коммунальное предприятие на пр...	1.0	394123538	21.11.1951	NaN	67	Женский	Казахстан	...	Ni
...
112621	72487	г.Нур-Султан	Государственное коммунальное предприятие на пр...	1.0	394918861	16.09.1963	NaN	56	Женский	Казахстан	...	Ni
112622	72488	г.Нур-Султан	Государственное коммунальное предприятие на пр...	0.0	394381753	15.12.1984	NaN	35	Мужской	Казахстан	...	Ni
112623	72489	г.Нур-Султан	Государственное коммунальное предприятие на пр...	1.0	5900000000002820	13.04.1999	NaN	20	Женский	Казахстан	...	Ni
112624	72490	г.Нур-Султан	Государственное коммунальное предприятие на пр...	0.0	406596836	06.10.1982	NaN	37	Мужской	Казахстан	...	Ni
112625	72491	г.Нур-Султан	Государственное коммунальное предприятие на пр...	1.0	394788134	09.08.1960	NaN	59	Мужской	Казахстан	...	Ni

112626 rows x 41 columns

Рисунок 3.22 – Объединенный DataFrame

Информация, хранящаяся в загруженных и объединенных файлах позволяет исследовать данные, где проходили лечение пациенты по РК. С помощью визуализации можно наблюдать результаты обработки на рисунке 3.23.

```
In [17]: order=combined_data["Регион, где пролечился больной"].value_counts().sort_values(ascending=False).index

sns.set_context("poster")
sns.set_style("whitegrid")
fig = plt.figure(figsize=(20,15));
sns.countplot(y='Регион, где пролечился больной',
              order=order,
              hue='Пол',
              data=combined_data)

plt.xlim([0,9000])
plt.title("По половым признакам");
```

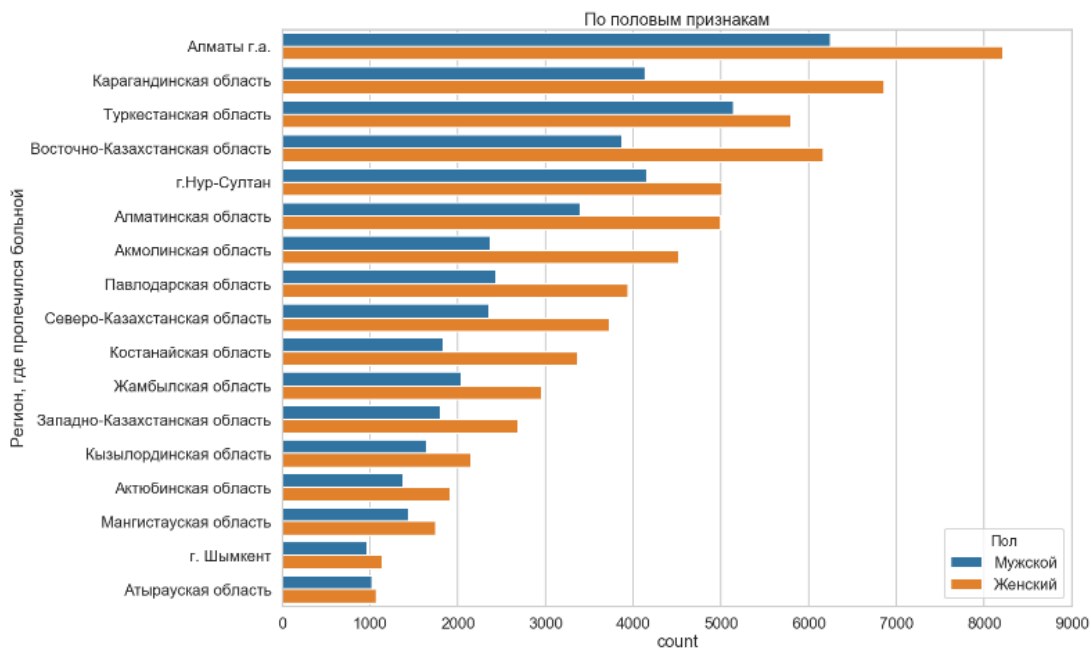


Рисунок 3.23 – Статистика регионов, где проходили лечение пациенты

В результате обработки можно увидеть, где проходили лечение пациенты, а также они были разделены по половому признаку. Также можно определить возраст пациентов проходивших лечение.

```
In [18]: sns.set_context("poster")
sns.set_style("whitegrid")
plt.figure(figsize=(20,15))
diag3 = sns.countplot(x = "Возраст",
                      order = range(0,100,5),
                      hue = "Пол", data=combined_data)
plt.ylim([0,2500])
plt.ylabel("Количество");
```

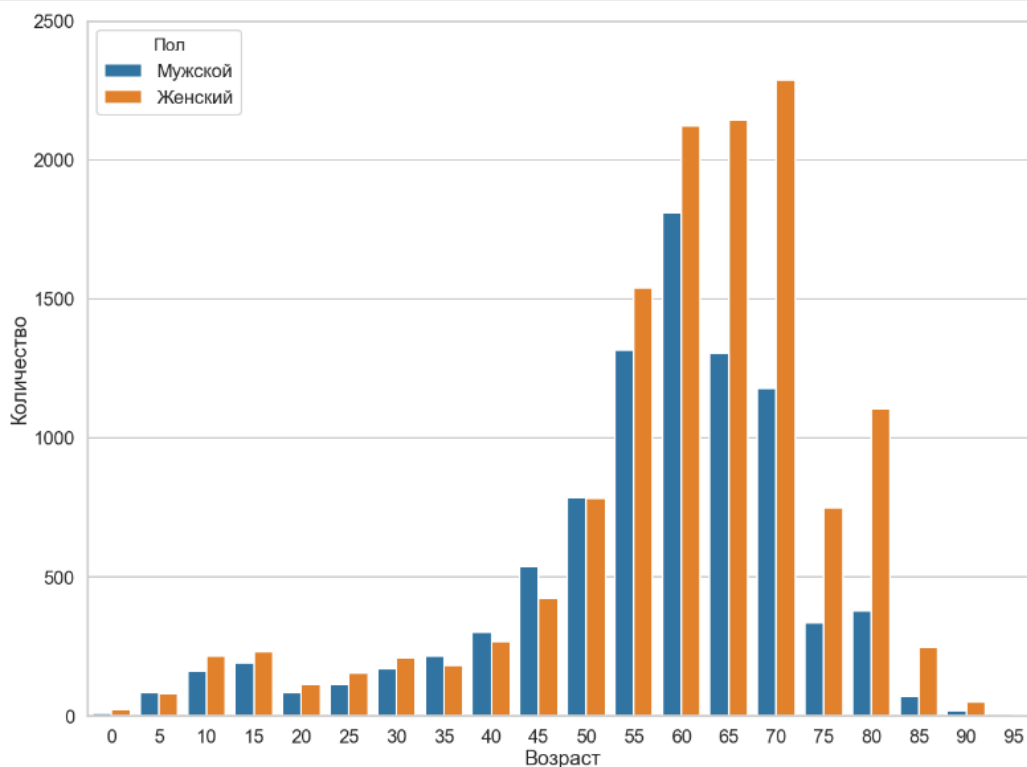


Рисунок 3.24 – Возраст пациентов

Следовательно, г. Алматы, Карагандинская, Туркестанская, Восточно-Казахстанская области, и г. Нур-Султан лидеры по количеству пациентов проходивших лечение среди регионов. Очевидно, что большая часть пациентов женщины, чем мужчины, и наибольшее количество женщин пациентов в возрасте 70 лет, тогда как для мужчин это значение меньше 60. Можно также отметить, что рост вероятности заболевания возникает после 40 лет и для мужчин и для женщин. Далее исследуем данные пациентов по гражданству, где в результате обработки можно увидеть, что большинство пациентов являются гражданами РК.

Out[23]:

Гражданство		Гражданство	
КАЗАХСТАН	111685	ИНДИЯ	3
РОССИЙСКАЯ ФЕДЕРАЦИЯ	315	СОЕДИНЁННЫЕ ШТАТЫ	3
НЕ УКАЗАНО	160	МОНГОЛИЯ	2
УЗБЕКИСТАН	128	ИЗРАИЛЬ	2
АЗЕРБАЙДЖАН	47	АМЕРИКАНСКОЕ САМОА	2
КЫРГИЗСТАН	32	ЛИТВА	1
АРМЕНИЯ	19	МАКЕДОНИЯ, БЫВШАЯ ЮГОСЛАВСКАЯ РЕСПУБЛИКА	1
КИТАЙ	19	ПАКИСТАН	1
УКРАИНА	17	ИОРДАНИЯ	1
ЛИЦО БЕЗ ГРАЖДАНСТВА	15	КАНАДА	1
Не указано	14	САУДОВСКАЯ АРАВИЯ	1
ТАДЖИКИСТАН	10	АФГАНИСТАН	1
ГРУЗИЯ	9	АВСТРАЛИЯ	1
ГЕРМАНИЯ	8	СОМАЛИ	1
ТУРЦИЯ	8	ОБЪЕДИНЁННЫЕ АРАБСКИЕ ЭМИРАТЫ	1
ТУРКМЕНИСТАН	6	МАЛАЙЗИЯ	1
МОЛДОВА, РЕСПУБЛИКА	5	ЕГИПЕТ	1
БЕЛАРУСЬ	3		

Рисунок 3.25 – Определение гражданства

Следовательно, 99% всех пациентов состоящие на учете являются гражданами РК. Также можно выяснить каким именно методом пациенты были зарегистрированы в едином реестре больных СД.

```
In [24]: combined_data.loc[:, "Кем направлен"] = combined_data.loc[:, "Кем направлен"].str.upper()

sns.set_context("poster")
sns.set_style("ticks")

sns.catplot(y = "Кем направлен",
            hue = "Планово",
            kind = "count",
            data=combined_data,
            legend = False,
            height = 10,
            aspect = 2)

plt.xlabel("Количество")
plt.legend(title="Направлено", loc='best', labels=['Экстренно', 'Планово']);
plt.xlim([0,45000]);
```

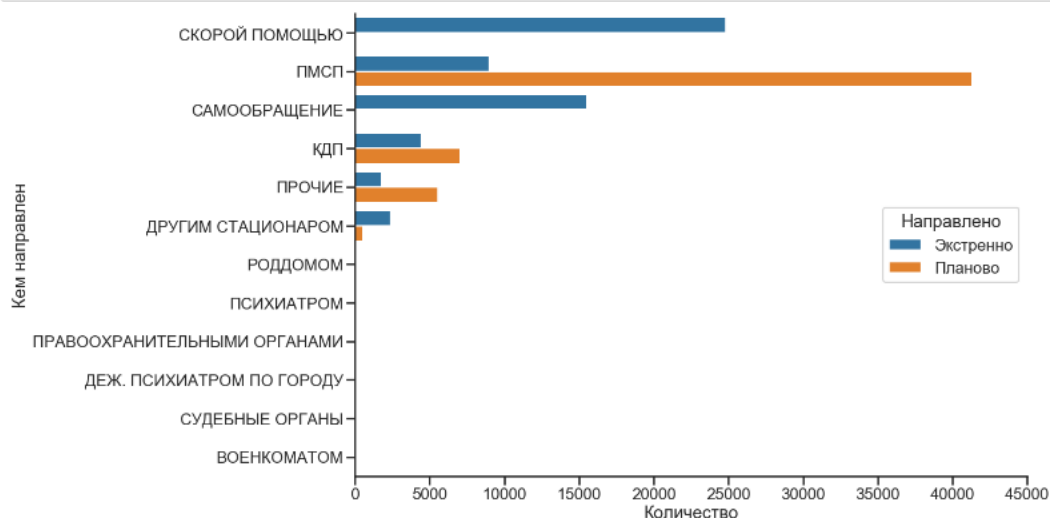


Рисунок 3.26 – Методы направления пациентов для регистрации

Согласно рисунку 3.26 скорая помощь и самообращение характеризуются экстренной доставкой пациента, тогда как первичная медико-санитарная помощь (ПСМП) и консультативно-диагностическая помощь (КДП) и прочие - по большей части были плановыми. Случаев с направлением от роддома, психиатра, правоохранительных органов, дежурного городского психиатра, судебного органа и военкомата единицы по отношению к предыдущим. Также можно вычислить 10 самых распространённых диагнозов. В результате вычисления можно увидеть заключительные диагнозы на рисунке 3.27.

Out[25]:

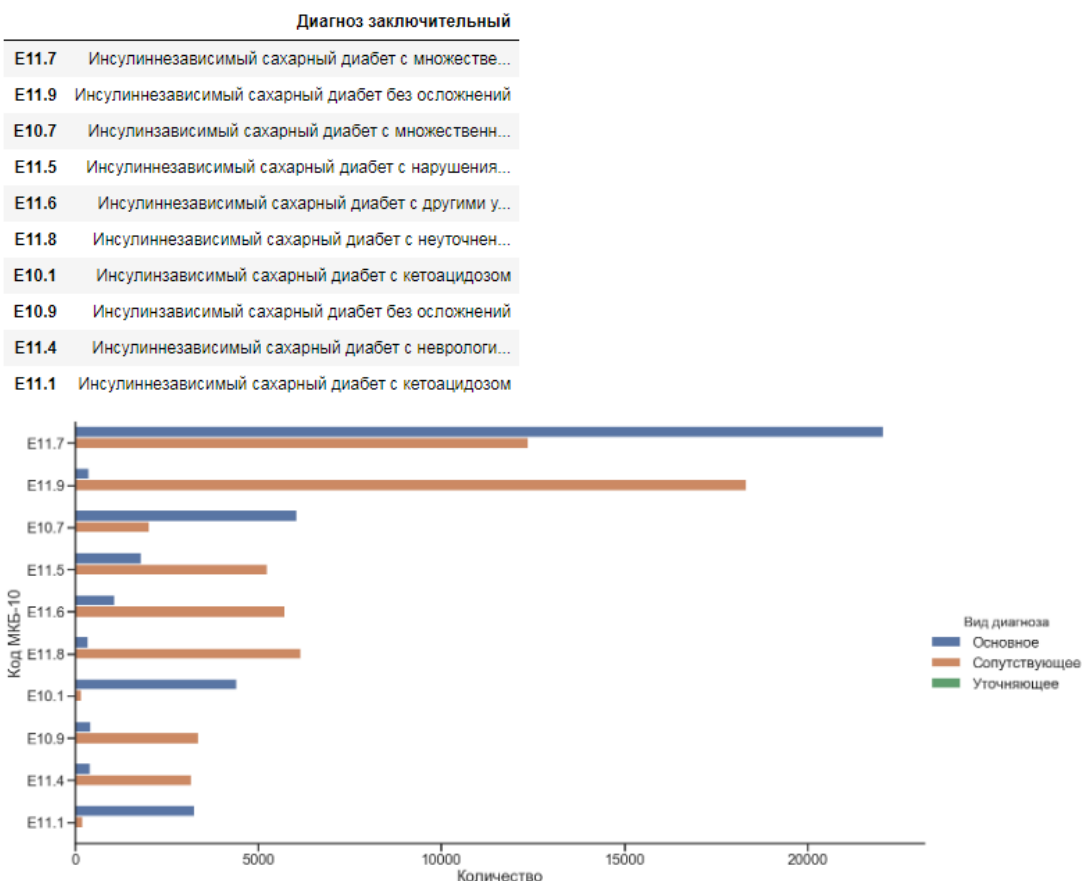


Рисунок 3.27 – Список распространенных диагнозов

Двумя самыми распространенными диагнозами являются инсулиннезависимый СД с множественными осложнениями и без осложнений, соответственно. В первом случае, зачастую, диагноз является основным, тогда как для второго случая, в подавляющем большинстве - сопутствующим. Остальные диагнозы и их виды также представлены на рисунке. Следующим примером для вычислений были использованы данные о количестве койко-дней в среднем на рисунке 3.28.

```

In [33]: plt.figure(figsize=(20,15))

sns.set(style = "ticks")
sns.set_context("poster")
sns.scatterplot(x = "Возраст",
                y = "Проведено койко-дней",
                hue="Исход лечения",
                hue_order=["Без перемен", "Улучшение", "Выздоровление", "Смерть", "Ухудшение"],
                data=combined_data)
plt.axhline(combined_data["Проведено койко-дней"].mean(), color = "k", label="Среднее значение койко-дней");
plt.legend();

```

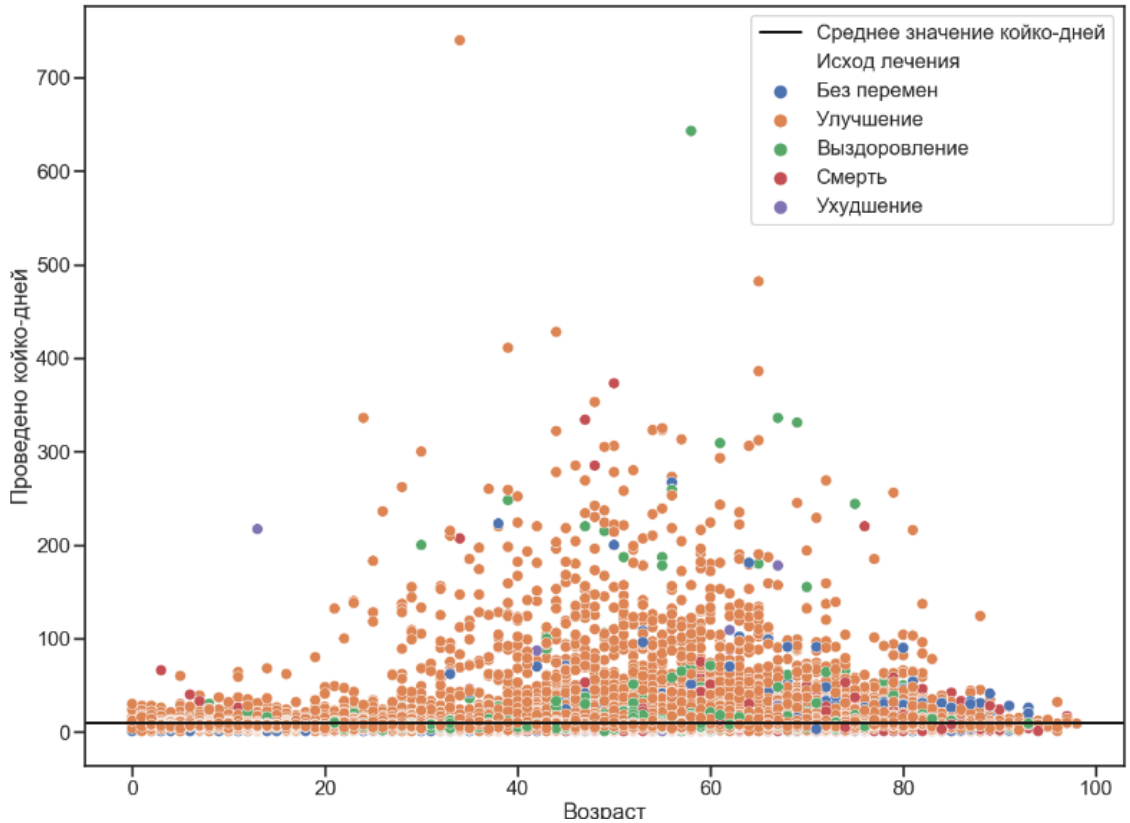


Рисунок 3.28 – Среднее значение проведенных койко-дней пациентами

Согласно рисунку 3.28 несмотря на некоторые значения в более 200 дней, средним значением койко-дней является значение около 10 (отмечено черной линией). Диаграмма показывает, что в возрасте от 30 до 70 лет вероятность провести более 50 койко-дней возрастает. Далее можно определить самые распространенные социальные статусы пациентов.

```
In [34]: #fig = plt.figure(figsize=(20,15));
combined_data.loc[combined_data["Категории льготника"]=="-", "Категории льготника"] = "Не предоставлен
о"

important_privileges = combined_data["Категории льготника"].value_counts().index[0:6]
important_status = combined_data["Социальный статус"].value_counts().index[0:10]
sns.set(style="whitegrid", context="poster")

sns.catplot(y="Социальный статус",
            order = important_status,
            hue = "Категории льготника",
            hue_order = important_privileges,
            kind = "count",
            data=combined_data,
            height = 10,
            aspect = 2);
sns.set(style = "white");
plt.xlabel("Количество");
```

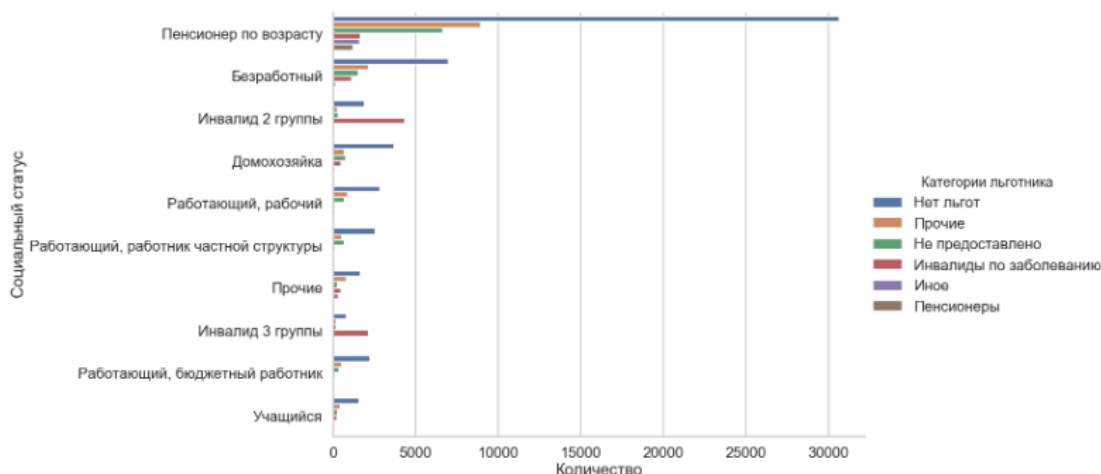


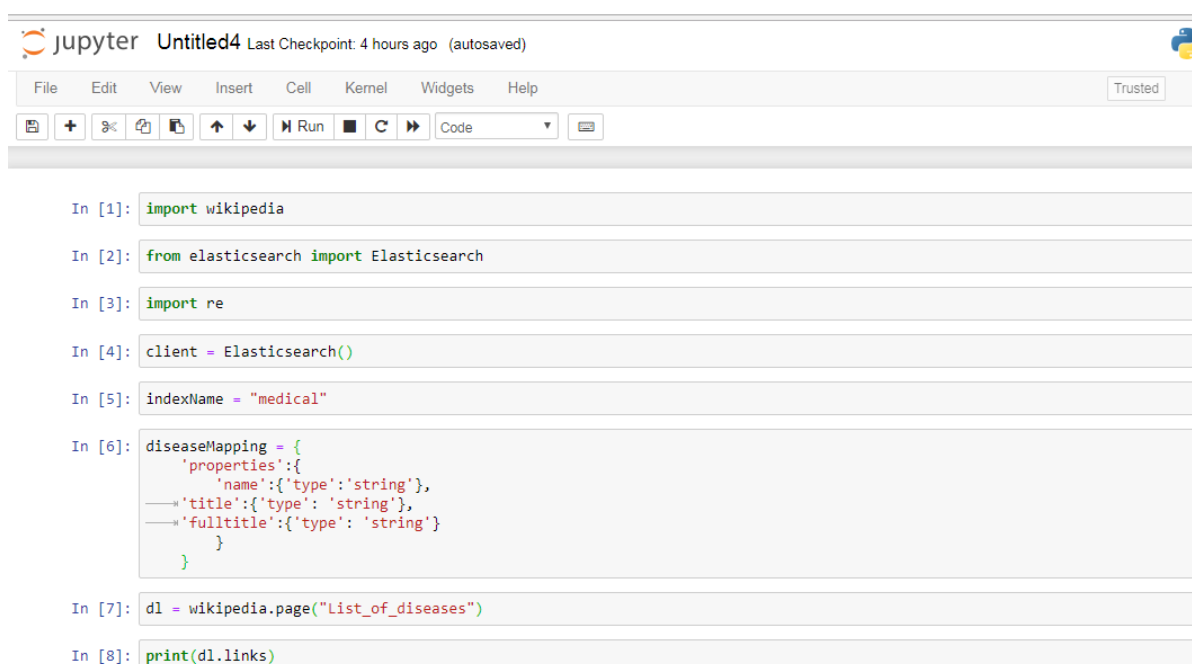
Рисунок 3.29 – Распространенные социальные статусы пациентов.

Согласно рисунку 3.29, 10 самых распространенных социальных статусов пациентов, где подавляющее большинство не имеют никаких льгот. Из пенсионеров лишь около четверти имеют ту или иную льготу. Среди инвалидов 2-й и 3-й группы лишь половина имеют льготы по инвалидности

3.4 Методы поиска ключевых данных для обработки медицинских данных

Процесс автоматизации диагностирования болезней является основной проблемой в современном мире. Для решения этой задачи является сбор внешних и внутренних медицинских данных. К внутренним данным можно отнести данные хранящиеся на электронных и бумажных носителях медицинских учреждений, а внешние включают все данные находящиеся в свободном доступе. В нашем случае для обработки и извлечения нужной информации, которые приведены в работе [78, с. 199], где можем воспользоваться медицинскими данными с Википедии. Для хранения и анализа больших массивов данных, поступающих с различных источников используется платформа Hadoop, но для обработки небольшого количества данных можно применить решения на базе NoSQL-решений. NoSQL-решения позволяют хранить и обрабатывать данные среднего объёма. К таким решениям, относится Elasticsearch, который является распределенной

системой поиска и аналитики с открытым исходным кодом для всех типов данных. Как и MongoDB, Elasticsearch представляет с собой хранилище документов, но также является поисковой системой для полнотекстового поиска. Данные получены из страницы Википедии со списком болезней в виде алфавитных списков. Для того чтобы обработать данные с Википедии необходимо через указанный доступный адрес скачать данные. Для этого создаем консольное приложение для диагностики всех болезней на основе данных с Википедии. Для реализации данного приложения был использован «Elasticsearch» и язык программирования Python. «Elasticsearch» также создает локальный сервер и выдает порт для правильной работы. Далее в среде Python набираем следующую команду, т.е. подключаемся к Википедии и «Elasticsearch».



```
In [1]: import wikipedia

In [2]: from elasticsearch import Elasticsearch

In [3]: import re

In [4]: client = Elasticsearch()

In [5]: indexName = "medical"

In [6]: diseaseMapping = {
    'properties':{
        'name':{'type':'string'},
        'title':{'type':'string'},
        'fulltitle':{'type':'string'}
    }
}

In [7]: dl = wikipedia.page("List_of_diseases")

In [8]: print(dl.links)
```

Рисунок 3.30 – Создание модели в Elasticsearch

На рисунке «Elasticsearch» сообщает, что индекс будет включать тип документа с именем «Disease» тем самым указываем тип каждого из полей. Данный файл содержит такие поля как: name, title, fulltext и имеют тип string . Далее необходимо получить страницу со списком всех болезней. Создаем массив для определения болезни и загружаем данные полученные с Википедии по болезням.

```
[u'Airborne disease', u'Contagious disease', u'Cryptogenic disease', u'Disease', u'Disseminated disease', u'Endocrine disease',
u'Environmental disease', u'Eye disease', u'Lifestyle disease', u'List of abbreviations for diseases and disorders', u'List of
cancer types', u'List of communication disorders', u'List of cutaneous conditions', u'List of diseases (0\u20139)', u'List of d
iseases (A)', u'List of diseases (B)', u'List of diseases (C)', u'List of diseases (D)', u'List of diseases (E)', u'List of disea
ses (F)', u'List of diseases (G)', u'List of diseases (H)', u'List of diseases (I)', u'List of diseases (J)', u'List of disea
ses (K)', u'List of diseases (L)', u'List of diseases (M)', u'List of diseases (N)', u'List of diseases (O)', u'List of disea
ses (P)', u'List of diseases (Q)', u'List of diseases (R)', u'List of diseases (S)', u'List of diseases (T)', u'List of diseases
(U)', u'List of diseases (V)', u'List of diseases (W)', u'List of diseases (X)', u'List of diseases (Y)', u'List of diseases
(Z)', u'List of disorders', u'List of eating disorders', u'List of fictional diseases', u'List of genetic disorders', u'List of
heart disorders', u'List of human disease case fatality rates', u'List of infectious diseases', u'List of intestinal diseases',
u'List of liver disorders', u'List of mood disorders', u'List of neurological disorders', u'List of notifiable diseases', u'List
of personality disorders', u'List of voice disorders', u'List of vulvovaginal disorders', u'Lists of diseases by year of disc
overy', u'Localized disease', u'Mental illness (alphabetical list)', u'Non-communicable disease', u'Organic disease', u'Outline
of exercise', u'Outline of health', u'Outline of nutrition', u'Progressive disease', u'Public health law', u'Rare disease', u'S
ystemic disease']
```

```
In [9]: diseaseListArray = []
```

```
In [10]: check = re.compile("List of diseases*")
```

```
In [11]: for link in dl.links:
    if check.match(link):
        try:
            diseaseListArray.append(wikipedia.page(link))
        except Exception,e:
            print str(e)
print(diseaseListArray[0].links)
```

Рисунок 3.31 – Ссылки со списком болезней в Википедии

Страница содержит большой массив ссылок, для этого при загрузке данных с Википедии массив был структурирован и сортирован.

```
In [12]: diseaseListArray
```

```
Out[12]: [<WikipediaPage 'List of diseases (0-9)'>,
<WikipediaPage 'Lists of diseases'>,
<WikipediaPage 'List of diseases (B)'>,
<WikipediaPage 'List of diseases (C)'>,
<WikipediaPage 'List of diseases (D)'>,
<WikipediaPage 'Lists of diseases'>,
<WikipediaPage 'List of diseases (F)'>,
<WikipediaPage 'Lists of diseases'>,
<WikipediaPage 'List of diseases (H)'>,
<WikipediaPage 'Lists of diseases'>,
<WikipediaPage 'List of diseases (J)'>,
<WikipediaPage 'List of diseases (K)'>,
<WikipediaPage 'List of diseases (L)'>,
<WikipediaPage 'List of diseases (M)'>,
<WikipediaPage 'List of diseases (N)'>,
<WikipediaPage 'List of diseases (O)'>,
<WikipediaPage 'List of diseases (P)'>,
<WikipediaPage 'List of diseases (Q)'>,
<WikipediaPage 'List of diseases (R)'>,
<WikipediaPage 'List of diseases (S)'>,
<WikipediaPage 'List of diseases (T)'>,
<WikipediaPage 'List of diseases (U)'>,
<WikipediaPage 'List of diseases (V)'>,
<WikipediaPage 'List of diseases (W)'>,
<WikipediaPage 'List of diseases (X)'>,
<WikipediaPage 'List of diseases (Y)'>,
<WikipediaPage 'List of diseases (Z)'>]
```

Рисунок 3.32 – Структурированный список болезней в Википедии

Список болезней необходимо проиндексировать для этого в массиве «Checklist» хранятся допустимые первые символы. Если болезнь не отвечает критерию, она пропускается. Из массива «Checklist» получаем отсортированные данные и загружаем в «Elasticsearch» под определенным индексом для более удобной обработки данных. Также учитывается тип документа, для проверки являются ли эти данные болезнью, а затем индексируем для более быстрой обработки.

```

In [13]: checkList = [ ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"],
                      ["A"], ["B"], ["C"], ["D"], ["E"], ["F"], ["G"], ["H"],
                      ["I"], ["J"], ["K"], ["L"], ["M"], ["N"], ["O"], ["P"],
                      ["Q"], ["R"], ["S"], ["T"], ["U"], ["V"], ["W"], ["X"], ["Y"], ["Z"]]

docType = 'diseases'

for diseaselistNumber, diseaselist in enumerate(diseaseListArray):
    for disease in diseaselist.links:
        try:
            if disease[0] in checkList[diseaselistNumber] and disease[0:3] != "List":
                currentPage = wikipedia.page(disease)
                client.index(index=indexName, doc_type=docType, id=disease, body = { "name":disease, "title":currentPage.title, "
            except Exception,e:
                print str(e)

```

Рисунок 3.33 – Индексирование и загрузка болезней в Elasticsearch

После индексирования можно проверить работу приложения. После обработки загруженных данных в «Elasticsearch» удалось произвести подсчет количество наименований болезней, которые равны 4899 строкам.

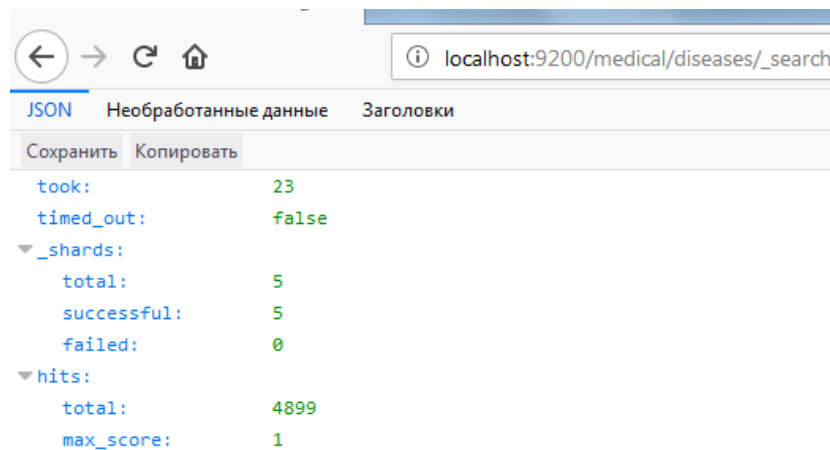


Рисунок 3.34 – Вывод данных в JSON

В данном примере разработано приложение, где были получены сведения о болезнях с Википедии и загрузили в базу данных «Elasticsearch» на локальный сервер. Осуществив сбор и подготовку медицинских данных теперь можно их исследовать. Для диагностики необходимо определиться с ключевыми симптомами, чтобы поисковая система могла определить вид болезни. Такие как: усталость, высокая температура, боль в суставах, сыпь, боль в груди. Предположим, что все эти симптомы присутствуют и, объявив их важными, перед поиском ставим знак «+» перед каждым симптомом. Результаты сортируются по рейтингу соответствия.

```
In [5]: from elasticsearch import Elasticsearch

client = Elasticsearch()
indexName = "medical"

docType = "diseases"
searchFrom = 0
searchSize = 3

searchBody = {
    "fields": ["name"],
    "query": {
        "simple_query_string": {
            "query": "+fatigue+fever+joint_pain+rash+chest_pain",
            "fields": ["fulltext", "title^5", "name^10"]
        }
    }
}

print(client.search(index=indexName, doc_type=docType,
                    body=searchBody, from_ = searchFrom, size=searchSize))
```

Рисунок 3.35 – Попытки поиска по симптомам

В применяемой переменной «SearchBody», которая содержит структуру данных JSON, указываются поля, которые хотите получить в возвращаемых результатах, в этом случае достаточно названия болезни.

```
{u'hits': {u'hits': [{u'_score': 0.010658317, u'_type': u'diseases', u'_id': u'Cutaneous lupus erythematosus', u'fields': {u'na
me': [u'Cutaneous lupus erythematosus']}, u'_index': u'medical'}, {u'_score': 0.010658317, u'_type': u'diseases', u'_id': u'Lup
us erythematosus', u'fields': {u'name': [u'Lupus erythematosus']}, u'_index': u'medical'}, {u'_score': 0.0056674588, u'_type':
u'diseases', u'_id': u'Systemic lupus erythematosus', u'fields': {u'name': [u'Systemic lupus erythematosus']}, u'_index': u'med
ical'}], u'total': 6, u'max_score': 0.010658317}, u'_shards': {u'successful': 5, u'failed': 0, u'total': 5}, u'took': 247, u'ti
med_out': False}
```

Рисунок 3.36 – Результаты поиска по пяти симптомам

Результаты поиска демонстрируют, что при достаточном количестве симптомов, вероятность диагностирования повышается, так как в данном случае при введении в поисковую систему пять симптомов волчанки (lupus), были достаточны для диагностирования этой болезни.

Рассмотрим пример для вывода совпадений по запросу с фильтром «Diabetes» в названии болезни. Чтобы получить список ключевых слов, соответствующих выбранному заболеванию, т.е. какие слова должны считаться значимыми. Для этого будет использоваться агрегирование значимых терминов. Следовательно, «Elasticsearch» профилирует итоговый набор, представляя ключевые слова, отличающего его от других данных. Тем самым «Filter» отбирает все болезни, в названии которых присутствует слово «Diabetes». Запрос фильтруется по полю «name» а значение остается только в том случае, если оно содержит слово «Diabetes». Результатом обработки листинга является количество возвращённых строк, которые равняются 28 значениям [78, с. 217]. Все результаты обработки листингов можно увидеть в приложении А.

Выводы по третьему разделу

1. Разработаны новые модели для решения задач выявления наиболее эффективных методов регрессионного анализа для прогнозирования роста пациентов СД на фоне пассивного выявления.

2. Разработаны эффективные алгоритмы моделирования методов регрессии с использованием библиотеки `scikit-learn`.

3. Произведен исследовательский анализ, обработки больших данных с целью выявления проверки гипотез, закономерностей, аномалий с помощью сводной статистики и графических представлений данных.

4. Исследована на базе NoSQL-решений система диагностирования болезни по симптомам с помощью распределенной системой поиска и аналитики «Elasticsearch»

5. Основные результаты раздела опубликованы в [103-104, 117].

4 РАЗРАБОТКА МОДЕЛЕЙ ДИАГНОСТИКИ СД С ПОМОЩЬЮ ИНСТРУМЕНТОВ BIGDATA

4.1 Математическая модель обработки и поиска больших данных

БД стали важной областью исследований благодаря появлению огромных данных, генерируемых из различных источников, и играют решающую роль во многих областях принятия решений и прогнозирования. Быстрое развитие различных инструментов БД, таких как Hadoop, Spark, в исследовательских и промышленных сообществах позволило распространять, передавать и обрабатывать огромное количество данных. Приложения БД используют методы анализа для эффективного анализа больших объемов данных. Однако выбор подходящих инструментов БД, основанных на пакетной и потоковой обработке данных и методах аналитики для разработки системы БД, затруднен из-за проблем обработки и применения БД. Практики и исследователи, разрабатывающие системы больших массивов данных, имеют недостаточную информацию о текущей технологии и требованиях, предъявляемых к платформе БД. В связи с этим существует высокая потребность в исследованиях в области БД. Тенденций исследований, связанных с обработкой БД, а также в использовании инструментов БД, позволяют обеспечивать эффективные решения в таких областях, как здравоохранение и медицинская помощь. Результаты таких исследований обеспечат понимание и знание существующих платформ БД и областей их применения, преимуществ и недостатков инструментов БД, методов анализа БД и их использования, а также новых исследовательских возможностей в будущем развитии систем БД.

В качестве объема БД указываются петабайты, эксабайты, зеттабайты, а также йоттабайты, а объем самым первым атрибутом БД [118]. Важность заключается в слове «большой», где необходимая информация могут быть обнаружены из анализа огромных массивов данных. Следовательно, предположим, что S – множество, где мощность множества S равна n , обозначаемая через [119], $|S| = n$. Если в S есть ровно n различных элементов, где n – неотрицательное целое число [120]. Пусть $S = \{x, b, z, e, f\}$, затем $|S| = 5$, когда n – неотрицательное целое число, S называется конечным. Таким образом, когда S не является конечным, тогда S называется бесконечным. Мощность бесконечного множества S обсуждается в дискретной математике, теории вещественных функций и теории мер, где для любого бесконечного множества существует множество с ещё большей мощностью – таким образом, не существует бесконечного множества наибольшей мощности. В моделировании сущностных отношений используется значение атрибута как наименьшую единицу для представления данных. Фактически, значение атрибута было наименьшей единицей для определения данных и манипулирования данными в системах баз данных. Значение атрибута по-прежнему является наименьшей единицей измерения для базы данных NoSQL или обработки веб-данных [121]. Поэтому также

используется значение атрибута как наименьшая единица БД. Значение атрибута может быть обозначено как v . Для примера рассмотрим, $v_1 =$ большие данные, $v_2 =$ данные, $v_3 =$ диабет, $v_4 =$ прогнозирование, $v_5 =$ система. С точки зрения поиска, эти значения можно рассматривать как ключевые слова, а с точки зрения математической лингвистики они являются самыми основными элементами для построения предложения путем конкатенации, следовательно, абзаца, страницы текста и т. д., по мере продвижения вверх по иерархии. Однако для задачи поиска можно не заботиться о некоторых значениях атрибутов для определенной группы, так называемых предлогов, поскольку все эти значения атрибутов можно рассматривать как незначимые элементы или компоненты. С теоретической точки зрения значением атрибута v может быть любое слово. Следовательно, можно ввести значение атрибута в виде последовательности $v_i, i = \{1, 2, 3, \dots, n, \dots\}$, что означает предел последовательности значений атрибутов v_i , когда i стремится к бесконечности. Пусть V - набор значений атрибутов, и предположим, что набор всех значений атрибутов идентифицируется как огромные массивы БД U . U - состоит из всех доступных данных, и в основном охватывает всю информацию то, с чем можно столкнуться в исследованиях БД [122]. Соотношение между V и U выглядит следующим образом:

$$V \subseteq U \quad (4.1.1)$$

V можно рассматривать как базовый элемент U . Требуется только конечная последовательность значений атрибутов $v_i, i = \{1, 2, 3, \dots, n, \dots\}$, чтобы составить предложение с помощью конкатенации, где n – целое число. Однако не можем использовать фиксированное целое число n и соответствующую последовательность значений атрибутов $v_i, i = \{1, 2, 3, \dots, n, \dots\}$, для представления того, что имеем, и будем иметь. По крайней мере, предоставляется бесконечное число значений атрибутов, чтобы составить объем БД, больше информации, большие знания, потому что число предложений теоретически бесконечно. Это подразумевает, что мощность или размер V равен $|V| = N_0$, где N_0 является мощностью всех целых чисел N , тогда U должен быть по крайней мере неисчислимой бесконечностью [123], как мощность всех вещественных чисел, потому что любое подмножество V может быть образовано значимым предложением, абзацем или текстом. Например, если осуществляется поиск «BigData» как v_i , то можно найти набор текстов или данных, состоящих из «BigData». Это означает, что элемент V соответствует множеству элементов U . Таким образом, имеем отношение между мощностью V и мощностью U :

$$|U| = 2^{|V|} \quad (4.1.2)$$

Исходя из вышперечисленного, $|U| = c$, c - это мощность множества вещественных чисел R и $|V| = N_0$, то есть

$$|U| = c = 2^{|V|} = 2^{N_0} \quad (4.1.3)$$

Соответственно, для примера пусть $u \in U$ будет документом в большом массиве данных, u может быть файлом .docx или отчет в формате pdf. Пусть $v \in U$ будут значением атрибута, v может быть такое слово, как «большие данные» или «диабет». Функция поиска, обозначаемая $s: V \rightarrow U$, определяется как:

$$s(v) = u \text{ если } v \in u \quad (4.1.4)$$

Если для поиска использовать «сахарный диабет», обозначенной как v , то поиск будет осуществляться как файл во всех областях, обозначенный как u , включая v . В более общем поисковая функция s может быть определена следующим образом:

$$s(v) = F(u) \quad (4.1.5)$$

где $F(u) = \{u_i | \text{если } v \in u_i, u_i \in U, i = \{0,1,2, \dots, m\}\}$. Поиск по одному ключевому слову соответствует, по крайней мере, картинке/документу или набору картинок/документов в качестве результатов поиска. При поиске «большие данные» в поисковых системах можно получить более 60 500 000 результатов. Следовательно, $F(v) \subseteq U$. Для $i = 0$, не получаем никаких исследованных результатов. Используя формулу (6) как определение наиболее общей функции поиска в том, что следует далее. Рассмотрев свойство функции поиска как операция, поскольку поиск в датасете с помощью поисковых систем - это операция, а поиск или запрос в реляционной базе данных с использованием структурированного языка запросов SQL, как и поисковая система, - это манипуляция данными [119, с. 7].

Пусть $v_1, v_2, v_3 \in V$, затем используя формулу 6, $s(v_1) = F(v_1), s(v_2) = F(v_2), s(v_3) = F(v_3)$, затем сохраняются следующие свойства функций поиска:

$$s(v_1 \vee v_2) = s(v_1) \cap s(v_2) = F(v_1) \cap F(v_2) \quad (4.1.6)$$

где \vee - это операция между v_1 и v_2 для повторного поиска с помощью поисковых систем. \vee как оператор обладает свойством ассоциации, то есть, $v_1 \vee (v_2 \vee v_3) = (v_1 \vee v_2) \vee v_3$. Эта операция аналогична конкатенации между элементами данных в лингвистике или формальном языке. Следует отметить, что на практике, осуществляя поиск по ключевым словам такие как

«большие данные», «сахарный диабет» и «прогнозирование» используя поисковую систему (24 июня 2020) и обнаруживаем, что s (сахарный диабет) = 12,900,000, а s (большие данные) = 78,200,000, и s (прогнозирование) = 6,230,000. Это удовлетворяет тому, что $s(\text{прогнозирование}) < \min \{s(\text{большие данные}), s(\text{сахарный диабет})\} = s(v_1 V v_2)$, где этот результат подразумевает, что семантическое объединение отличается от объединения в науке о количественных отношениях. Первое может оказаться более сложным, чем второе, т.е. больше не входим в него из-за ограничения сферы охвата.

Теперь дана последовательность значений атрибута $v_i, i = \{1, 2, 3, \dots, n, \dots\}$, где $s(v_1) = F(v_1), s(v_2) = F(v_2), \dots, s(v_n) = F(v_n)$ из этого следует вывод, что в конечном мире больших данных результат поиска по отношению к операции V равен:

$$s(v_1 V v_2 V \dots V v_n V \dots) = \prod_1^{\infty} F(v_i) = F(v_1) \cap F(v_2) \cap \dots \cap F(v_n) \quad (4.1.7)$$

Когда $n \rightarrow \infty$, следующее свойство поиска как операции V в веб-поиске удерживается:

$$s(v_1 V v_2 V \dots V v_n V \dots) = \prod_1^{\infty} F(v_i) \quad (4.1.8)$$

Уравнения (4.1.7) и (4.1.8) являются теоремами представления для поиска в больших массивах данных соответственно. Также это отражает наиболее общий закон поиска в цифровом пространстве. Для лучшего понимания можно рассмотреть пример, где предположим, пользователь в датасете ищет «аналитику БД» с помощью поисковых систем. Однако, не получив нужную информацию ему пришлось расширить пространство поиска с помощью «BigData». С математической точки зрения, мы можем обозначить «аналитику БД» как v_1 и «BigData» как v_2 . Тогда расширение пространство поиска означает, что он использует $v_1 \wedge v_2$ (своего рода пересечение в теории множеств) чтобы найти то, что он ожидает в датасетах. Тогда имеем следующие двойные результаты, соответствующие выражениям (4.1.7) и (4.1.8), основанные на двойственном принципе в дискретной математике. В конечном мире БД результат поиска по операции \wedge равен:

$$s(v_1 \wedge v_2 \wedge \dots \wedge v_n) = \prod_1^n F(v_i) = F(v_1) \cup F(v_2) \cup \dots \cup F(v_n) \quad (4.1.9)$$

Когда $n \rightarrow \infty$, следующее свойство поиска как операции \wedge в веб-поиске сохраняется:

$$s(v_1 \wedge v_2 \wedge \dots \wedge v_n \wedge \dots) = \prod_1^{\infty} F(v_i) \quad (4.1.10)$$

Уравнения (4.1.7) - (4.1.10) являются математической основой для поиска, структурирования и работы с БД. Следует отметить, что ограничиваем БД текстовыми данными и мультимедийными данными. На самом деле в науке и промышленности существует множество областей БД, где данные являются числовыми и т.д., и необходимо использовать аналогичные методы для работы с численными массивами данных [119, с. 9].

4.2 Разработка математической модели диагностирования сахарного диабета с помощью больших данных

Перспективность использования интеллектуальных методов и моделей подтверждается в системах определения информационных рисков, меры определенности информации и прочих параметров диагностики СД, однако, использование ИС при управлении мерами данных разного типа в облачных сервисах всё еще уступает место классическим математическим алгоритмам, что связано с ограниченностью применения стандартных технологий вычислений. Решение этой задачи видится во включении в ИС элемента экспертных оценок на основе имеющихся стандартизированных развернутых протоколов и данных экспертных отечественных протоколов диагностики [124].

Вводятся анализы и снимки ЭКГ, УЗИ ОБП и почек, УЗДГ сосудов нижних конечностей, флюорография, справка фтизиатра, рентгенография. После проверки всех полученных анализов и обследований врач-эндокринолог официально диагностируют СД пациенту, и принимает решение о назначении лечения и возможности госпитализации, если это будет необходимо. Однако, точность прогноза вероятности подтверждения диагноза по указанным параметрам даже с учетом допуска параллельных вычислений, может быть достигнута только за счет оптимизации неструктурированных данных. В разработанной системе используется последовательное выполнение этапов математического описания процесса работы модели:

Этап 1 - формирование множества входных и выходных параметров модели. Этап предусматривает использование разработанной модели интеграции параметров допуска вероятности диагноза по количеству критериев, что позволяет на основе анализа характеристик объекта $H = \{h_1 \dots h_5\}$ и характеристик параллельных вычислений по различным медицинским анализам $F = \{f_1 \dots f_f\}$ определить множество параметров допуска количества обращений, которые будут использованы в качестве входных параметров системы:

$$X = \{x_1, \dots, x_{N_x}\} \quad (4.2.1)$$

где N_x - количество входных параметров модели. Также определяется множество выходных параметров модели, которые будут свидетельствовать о наличии / отсутствии параллельных вычислений определенного типа:

$$Y = \{y_1, \dots, y_{N_y}\} \quad (4.2.2)$$

где N_y - количество выходных параметров.

Этап 2 - получение статистических данных. В результате анализа характеристик облачного сервиса в ядре диагностики $M = \{m_1 \dots m_k\}$ оценивается возможность регистрации параметров подтверждения диагноза, используемых для определения X и Y . Если оценка отрицательная, то диагноз не подтвержден, то есть:

$$M = \{m_{MPNN}\} \quad (4.2.3)$$

Этап 3 - представление экспертных знаний. Используя разработанный алгоритм применения продукционных правил для представления экспертных знаний в подключаемых модулях, оценивается возможность разработки модели подядра диагностики качественных результатов (данные УЗИ и т.п.). Если оценка отрицательная, то модель подядра исключается из дальнейшего рассмотрения:

$$m_{MPNN} \notin M \quad (4.2.4)$$

В противном случае:

$$m_{MPNN} \in M \quad (4.2.5)$$

Этап 4 - проверка множества допустимых видов признаков. Этап ориентирован на проверку множества допустимых клинических картин, которые не являются значимыми для подтверждения диагноза. Таким образом, для определения ε используется правило:

$$\text{Если } \delta_1 \leq \delta_2 \rightarrow \varepsilon = \delta_1 \text{ иначе } \varepsilon = \delta_2 \quad (4.2.6)$$

где δ_1 - минимально допустимая величина параллельных вычислений по признакам отклонений от нормы, δ_2 - минимально допустимая величина ложного распознавания диагноза, ε - минимально допустимая ошибка.

Этап 6 - определение минимального объема обучающей выборки. Данный этап ориентирован на определение минимально допустимого количества критериев оценки для постановки диагноза. Расчет происходит так:

$$P_{min} = 20N_x \quad (4.2.7)$$

где P_{min} - минимально допустимое количество примеров, N_x - количество входных параметров.

Этап 7 - проверка. Этап предусматривает сравнение допустимого параметра в выборке для идентификации ошибок вычислений:

$$t_{min}(m_j) \geq t_d \rightarrow m_j \notin M^{tn} \quad (4.2.8)$$

где M^{tn} - множество значимых показателей с допустимым отклонением, m_j - j -й элемент системы. Поскольку модели эффективности использования облачных сервисов для этого случая в качестве обучающих примеров используются неструктурированные данные, то для модели система формирования выборки равна продолжительности разработки продукционных правил. Для расчета $T_{j,max}$ используется выражение:

$$T_{j,max} = T_f - t_j \quad (4.2.9)$$

где t_j - срок обучения j - ой модели $m_j \notin M^{(tn)}$. Результатом выполнения этапа является сформированное множество:

$$\{T_{1,max}, \dots, T_{L,max}\} \quad (4.2.10)$$

где L - количество элементов $M^{(tn)}$.

Множество систем, которые целесообразно использовать для оценки параметров допуска количества обращений, формируется с помощью выражений (11-12).

$$\text{Если } T_{j,max} > T_d \rightarrow m_j^{(tn)} \notin M_z \quad (4.2.11)$$

$$\text{Если } T_{j,max} < T_d \rightarrow m_j^{(tn)} \in M \quad (4.2.12)$$

где M_z - множество систем (в данной системе - типов анализов), которые целесообразно использовать для оценки параметров допуска количества обращений. В результате реализации алгоритма формируется множество нейросетевых систем, которые целесообразно использовать для оценки параметров допуска количества обращений с целью распознавания параллельных вычислений. При этом, само функционирование диагностической системы на основе комплексной модели сбора данных в облачном сервисе должно быть подключенным к медицинскому лечебному протоколу, выбор которого должен выступать отдельным модулем унификации. Поэтому к этапу применения методологий выбора технологии построения ИС надо исследовать типы данных в моделях с акцентом на

применение ИТ для БД и построения проекта модели диагностирования диабета с помощью БД, учитывающей все подходы к разработке таких моделей.

4.3 Выбор платформы на базе инструментов BigData технологии

Среда распределенной обработки Hadoop с открытым исходным кодом, управляет обработкой и хранением данных для приложений с БД, работающими в кластерных системах. Также может обрабатывать и хранить различные формы структурированных и неструктурированных данных. Hadoop работает на кластерах обычных серверов и может масштабироваться для поддержки тысяч аппаратных узлов и огромных объемов данных. Он использует одноименную файловую систему, которая предназначена для обеспечения быстрого доступа к данным между узлами в кластере, а также отказоустойчивых возможностей, поэтому приложения могут продолжать работать случае сбоя отдельных узлов. Тем самым, Hadoop стала основополагающей платформой управления данными для аналитики БД после того, как она появилась на рынке [125].

В первую очередь, данная среда ориентирована на аналитическое использование, а ее способность обрабатывать и хранить различные типы данных делают ее особенно подходящей для приложений и анализа больших данных. БД обычно включают в себя большие объемы данных типа полу структурированных и неструктурированных форм информации, таких как записи интернет-трафика, журналы веб-серверов и мобильных приложений, сообщения в социальных сетях, электронные письма клиентов, данные из сенсорных датчиков, данные из датчиков технологии Internet of Things и т.д. [126].

Для разработки и настройки платформы кластера используется программный продукт «Apache Ambari» компании «Apache Software Foundation». «Ambari» позволяет управлять и контролировать кластером Hadoop, а также интегрировать Hadoop с существующей инфраструктурой предприятия.

Разрабатываемая ИС имеет следующее функциональное значение, которое состоит из следующих задач:

1. В кластере Hadoop, создается реляционная база данных, где будут обрабатываться структурированные данные;
2. В кластере Hadoop, создается база данных, где будут обрабатываться неструктурированные данные различных форматов;
3. В кластере Hadoop, создание распределенной файловой системы HDFS.

Исходя из задач указанных выше, а также после проведения сравнительного анализа, были выбраны следующие инструменты Hadoop:

- Hive. Это система хранения с открытым исходным кодом, которая запрашивает и анализирует большие наборы данных, хранящиеся в основном в файлах Hadoop;

- MongoDB. Это система управления базами данных с открытым исходным кодом, которая использует модель базы данных, ориентированную на документы, которая в свою очередь поддерживает различные формы данных [127];

- Oozie. Это серверная система планирования рабочих процессов для управления заданиями Hadoop;

- Sqoop. Это инструмент, предназначенный для передачи данных между Hadoop и серверами реляционных баз данных. Рисунок 4.1 показывает исследуемую архитектуру конфигурирования экосистемы Hadoop.

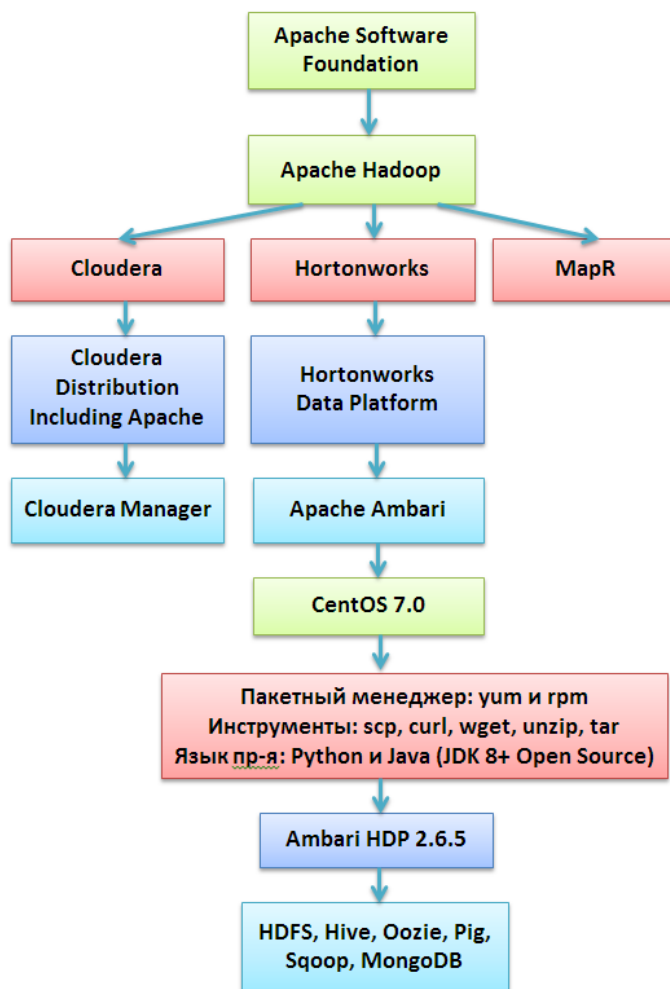


Рисунок 4.1 – Иерархическая архитектура экосистемы Hadoop

Компоненты Hadoop, состоят из трех уровней называемые стеками:

1. Core Hadoop состоит из HDFS, MapReduce;
2. Essential Hadoop состоит из Pig, Hive, ZooKeeper;
3. Hadoop Support: состоит из Hadoop, Hive, Oozie, Spark, Pig.

После проведенного анализа был выбран Hadoop Support, где собран весь пакет инструментов, который позволяет следить за работой Hadoop, а также есть возможность подключать его к более мощному кластеру. Для разработки и настройки платформы кластера используется программный

продукт «Apache Ambari» компании «Apache Software Foundation». «Ambari» позволяет управлять и контролировать кластером Hadoop [128], а также интегрировать Hadoop с существующей инфраструктурой предприятия. После проведения сравнительного анализа, были выбраны и установлены в среде платформы будущей экосистемы инструменты BigData технологий. Разрабатываемая ИС должна не только определять статус пациента, а также хранить все истории болезни пациента, чтобы врач удаленно мог иметь доступ к ней, и созданная база данных должна автоматически создавать репликацию (по умолчанию 3) и быть отказоустойчивой, иметь доступ к подключенным медицинским базам данных. Также система может быть использована студентами медицинских вузов в виде симулятора для повышения знаний при диагностировании диабета. Разрабатываемая ИС позволит облегчить работу врачей-эндокринологов, т.к. ими могут быть использованы не только собственные знания, но и мировые ресурсы базы знаний о сахарном диабете. Структура ИС диагностики СД на базе инструментов BigData технологий приведена в соответствии с рисунком 4.2.

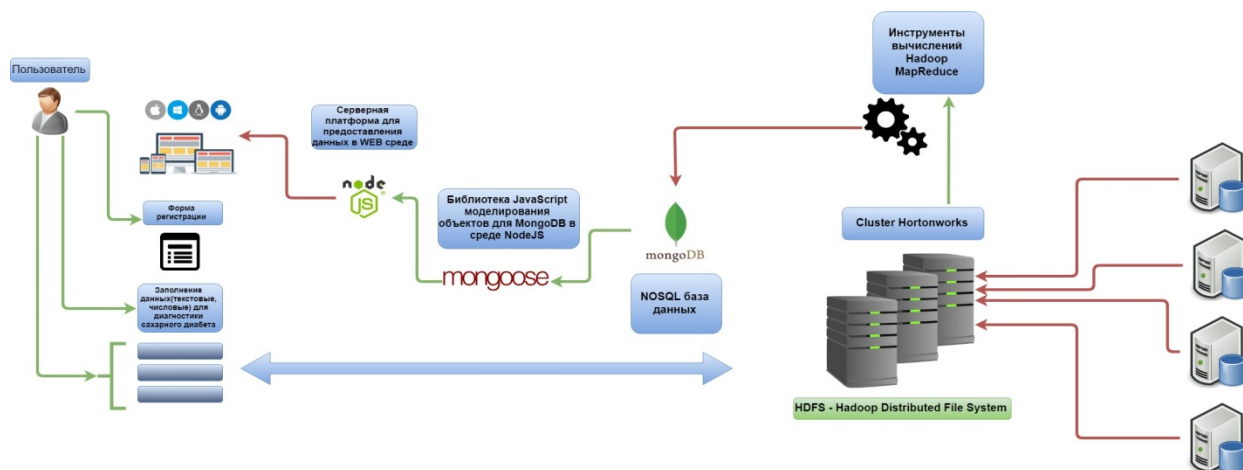


Рисунок 4.2 – Общая структура процесса взаимодействия компонентов системы

Концептуальная схема ИС, которая наглядно знакомит пользователя с функционированием информационно-имитационной системы. Также здесь подробно показаны взаимосвязи между модулями системы. На концептуальном уровне выделяют ИС по управлению данными или ориентированные на работу с данными. Актуальность таких ИС усилилась с появлением ИТ Big Data. Big Data как ИТ имеет такую формальную модель [129].

$$BD = (Vol_{BD}, I_P, A_{BD}, T_{BD}) \quad (4.3.1)$$

где Vol_{BD} - множество типов объемов; I_P - множество типов источников данных (диагностических критериев); A_{BD} - множество методик анализа БД;

T_{BD} - множество технологий обработки БД. На этапах обработки БД применяют следующие технологии:

$$T_{BD} = (T_{NoSQL}, T_{SQL}, T_{Hadoop}, T_V) \quad (4.3.2)$$

где T_{NoSQL} - технологии NoSQL баз данных; T_{Hadoop} - технологии обеспечения массивно-параллельной обработки; T_{SQL} - технологии обработки структурированных данных (базы данных SQL); T_V - технологии визуализации БД. Таким образом, происходит загрузка всех анализов в базу MongoDB, и они хранятся в кластере системы. Оптимальным вариантом, совместимым с двухъядерной структурой и поддержкой (базы данных SQL) является Hadoop, тогда как все неструктурированные данные (числовые, текстовые, снимки УЗИ и т.д.) загружаются в базу MongoDB. В связи с неструктурированностью данных особое внимание обратим на технологии обеспечения массивно-параллельной обработки, так как целью работы является исследование облачных вычислений и принятия проектного решения по выбору оптимальной модели постановки диагноза и лечения как двух параллельных облачных сервисов, компонентов ИС по управлению БД.

В настоящее время создание перспективных систем обеспечения информационной поддержки в лечении и диагностике СД ассоциируется с использованием интеллектуальных средств. Перспективность использования информационных автоматизированных моделей подтверждается в системах определения информационных рисков, связанных с неструктурированностью данных, мер определенности информации и прочих параметров диагностики, однако, использование ИС на основе двухъядерного компонента в облачном сервисе является вариантом решения данной проблемы и выступает отдельным модулем унификации по действующим протоколам установления диагноза и лечения СД.

4.4 Конфигурирование и запуск компонентов системы Hadoop

Для установки и развертки всей системы необходимо получить доступ к современному оборудованию, которое отвечает всем требованиям системы. В данной исследовательской работе была предоставлена возможность использовать возможности суперкомпьютера Сатпаев университета.

Техническое оборудование кластера. В данном исследовании использовалось оборудование суперкомпьютера Сатпаев университета. Был предоставлен удаленный доступ к нижеследующим компонентам:

- processor Intel Xeon x5550 2.67GHz 4 core (4 шт.);
- сервера компании FUJITSU PRIMERGY BX920 S1;
- RAM объемом каждая 16 GB (4 шт.);
- HDD по 150 Gb (4 шт.).

Физическая архитектура, приведенная на рисунке, включает в себя коммутаторы, маршрутизаторы, сетевые адаптеры. С помощью этих конфигурационных устройств посредством защищенного соединения SSH

ретранслируем IP адреса машин для удаленного подключения. Приведенная архитектура позволяет организовать взаимодействие всех четырех серверов с защищенным доступом к ним.

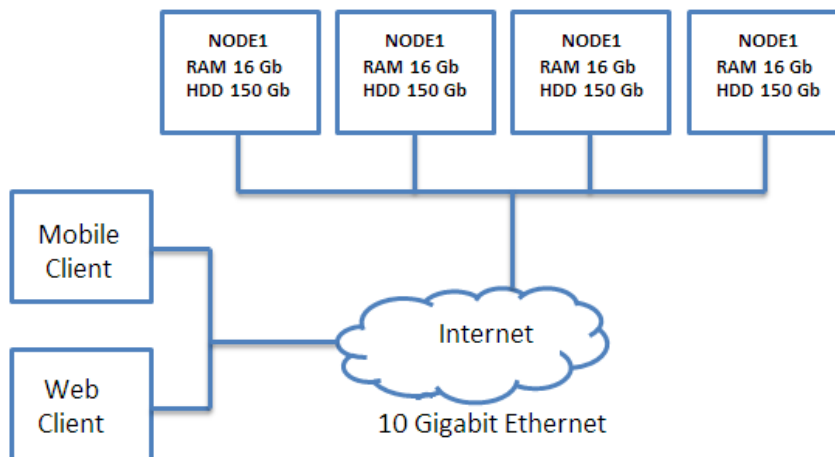


Рисунок 4.3 – Физическая архитектура сетевых подключений

Перечень программных средств. Выше была рассмотрена аппаратная часть сетевого соединения. В первую очередь для запуска работы всего кластера Hadoop предусматривает установку Unix-подобной операционной системы (ОС). После проведенного анализа, а также с учетом всех требуемых конфигурации выбор был остановлен на дистрибутиве CentOS 7.

Установка компонентов Hadoop с помощью Apache Ambari server Hortonworks Data Platform (HDP) 3.1.1 версии. Apache Ambari – это инструмент для обеспечения, управления и мониторинга кластеров Apache Hadoop. В состав Ambari входит набор REST API и интерфейс управления на основе браузера. Ambari server позволяет системным администраторам:

- 1) используя простой пошаговый мастер настройки и установки сервисов Hadoop на любом количестве хостов;
- 2) Ambari позволяет обрабатывать конфигурацию сервисов Hadoop для кластера;
- 3) Ambari обеспечивает централизованное управление для запуска, остановки и реконфигурации услуг Hadoop по всему кластеру;
- 4) Ambari обеспечивает мониторинг кластера Hadoop. Ambari предоставляет панель мониторинга за состоянием кластера Hadoop;
- 5) Ambari позволяет настроить систему оповещения администратора по электронной почте, в случае сбоев.

После запуска ambari-server нужно подключиться к серверу по http протоколу. После входа на экране появится Wizard для настройки сервера. Мы выбираем LaunchInstallWizard в соответствии с рисунком 4.4.

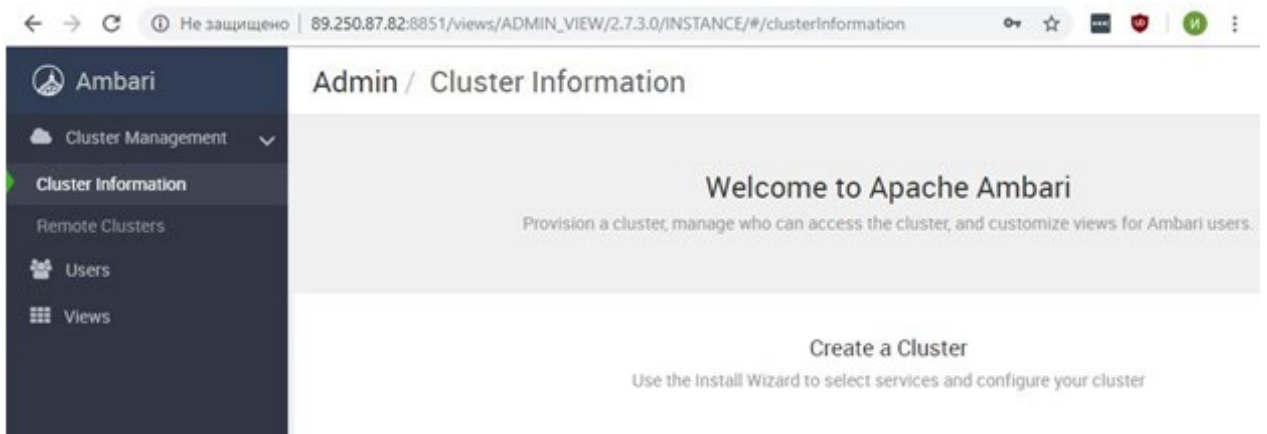


Рисунок 4.4 – Настройка и установка кластера HDP

Необходимо назвать имя кластера разрабатываемой системы, указать какую версию HDP нужно установить, и для какого именно дистрибутива необходимо установить кластер системы. После выбора ОС необходимо определиться с выбором для него репозитория. Далее необходимо указать какие машины будут подключены к серверу, также важной частью установки является необходимость указывать их имена. В нашем случае подключение будет происходить по защищенному протоколу SSH. Далее если все данные указаны правильно, то кластер проверит все сервера и подключиться к ним для дальнейшей установки. Если появляется зеленая полоса, то она указывает, что хост соединен с кластером, и на него можно установить инструменты BigData в соответствии с рисунком 4.5.

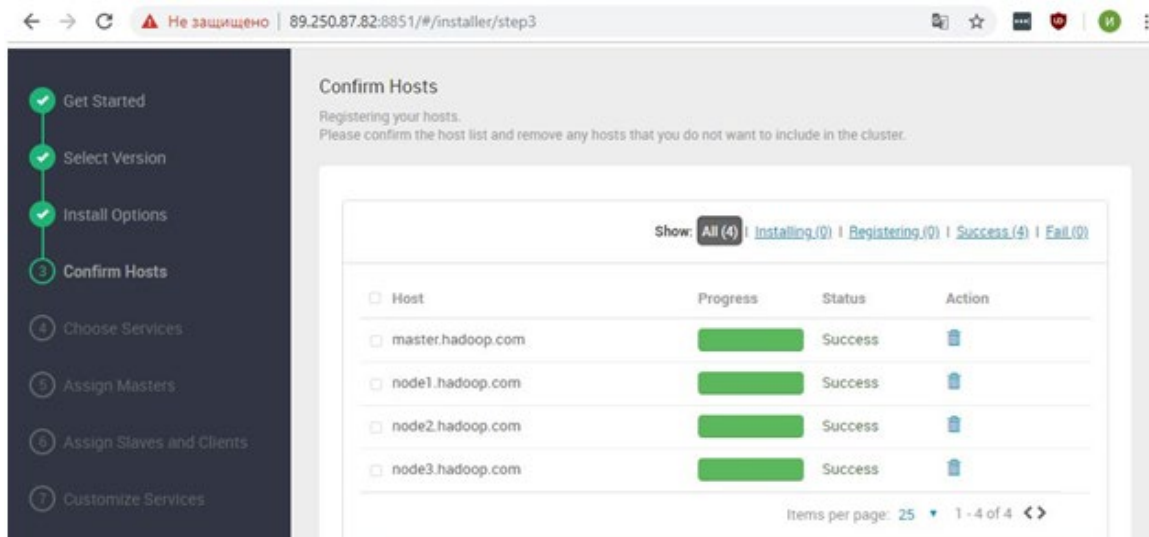


Рисунок 4.5 – Подключение хоста к кластеру

Необходимо выбрать какие пакеты будут установлены на узлы, которые будут подключены к кластеру. Далее выбираем нужные компоненты и необходимо просмотреть, нет ли ошибок в настройке кластера, и проверить все ли необходимые пакеты будут установлены. После необходимо выбрать

на каких машинах будут запущены сервисы для работы распределенной файловой системы, а также дополнительные сервисы. Затем необходимо настроить пароли для сервисов. После всех необходимых настроек кластер установит и настроит все необходимые инструменты на указанных машинах и можно увидеть окно готового кластера в соответствии с рисунком 4.6.

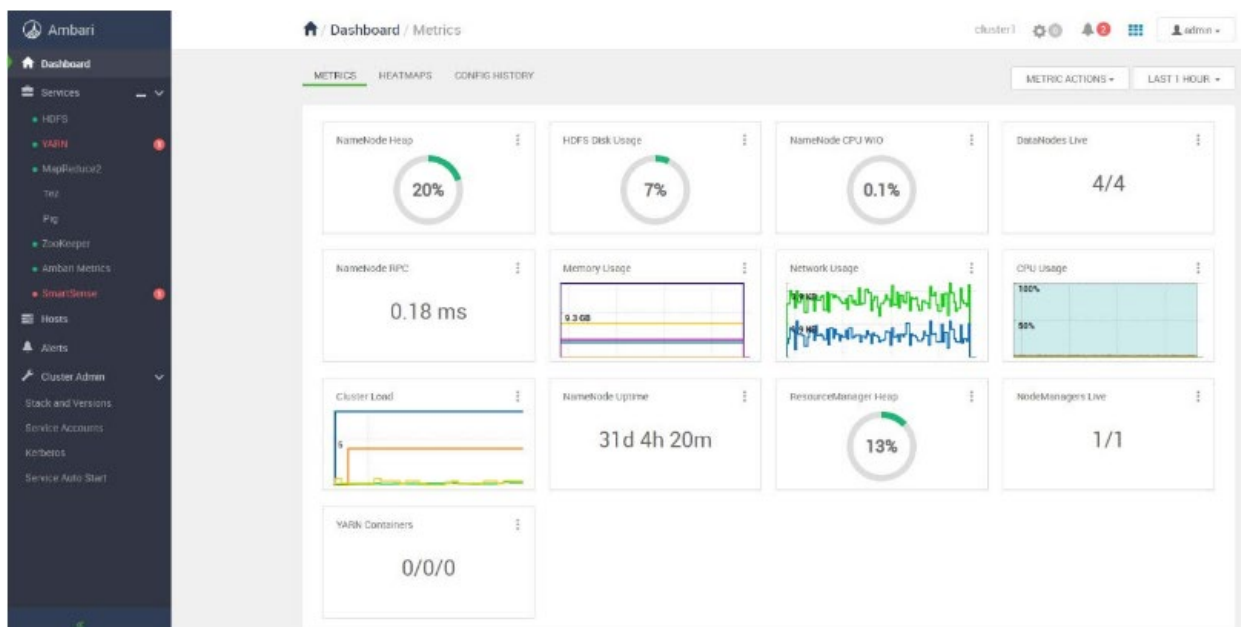


Рисунок 4.6 – Окно правильно настроенного кластера

После установки необходимых компонентов инструментов BigData технологий, такие как Hive, MongoDB, Pig, Oozie, Sqoop следующей основной задачей является создание и подключение различных базы данных для неструктурированных и структурированных данных и их взаимодействие между собой.

4.5 Администрирование и обработка данных в среде Hadoop

4.5.1 Загрузка и обработка данных с помощью MapReduce

В первых версиях Hadoop, основными компонентами были MapReduce и распределенная файловая система Hadoop (HDFS), а также Hadoop Common, набор общих утилит и библиотек. Для реализации модели распределенных вычислений MapReduce будет использоваться Java-фреймворк Hadoop. Hadoop – это фреймворк, позволяющий строить распределенные вычисления в кластерах для работы с огромными массивами данных. Основная задача MapReduce это использование функции map и reduce для разделения заданий обработки на несколько задач. Эти заданные задачи выполняются на узлах кластера, где хранятся данные, и затем для объединения того, что эти задачи дают согласованный набор результатов. Hadoop построен на Java и доступен через язык программирования Python для написания кода MapReduce [130].

MapReduce обычно разбивает входной набор данных на независимые блоки, которые обрабатываются шагом «Map» полностью параллельно. Фреймворк сортирует выходные данные с данного шага, которые затем переходят на шаг «Reduce», отвечающий за объединение данных с шага «Map». Обычно как входные, так и выходные данные задания хранятся в файловой системе. Фреймворк заботится о планировании задач, их мониторинге и повторном выполнении неудачных задач. Обычно вычислительные узлы и узлы хранения данных в вычислительном кластере совпадают, то есть MapReduce и распределенная файловая система Hadoop (HDFS) работают на одном и том же наборе узлов. Эта конфигурация позволяет платформе эффективно планировать задачи на узлах, где уже присутствуют данные, что приводит к очень высокой совокупной пропускной способности по всему кластеру.

Hadoop MapReduce - это вычисление, которое разбивает большие задания на отдельные задачи, которые могут выполняться параллельно на кластере серверов. Результаты задач могут быть объединены, чтобы вычислить окончательные результаты. Как говорилось ранее, MapReduce разбивает массив данных на независимые части, которые обрабатываются в отдельном узле вычислительного кластера. Поэтому нужно извлечь данные из.xlsx-файла и разбить их на части. Всего имеется порядка 60.000 записей, которые нужно посчитать, необходимо вести подсчет по определенным столбцам, по регионам и по возрасту.

Декомпозиция задачи. Создаем 3 класса необходимых для функционирования нашей программы:

1. TokenizerMapper – класс, расширяющий суперкласс Map<KEYIN,VALUEIN,KEYOUT,VALUEOUT> и сопоставляющий входные данные виды key/value с набором промежуточных пар key/value;

2. IntSumReducer – класс, расширяющий суперкласс Reducer<KEYIN,VALUEIN,KEYOUT,VALUEOUT> и уменьшающий набор промежуточных значений, имеющих общий ключ, до меньшего набора значений;

3. Main – главный класс, конфигурирующий работу фреймворка Hadoop.

Класс *Main*.

```

package Main;

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import MapReduce.*;

public class Main {

    public static void main(String[] args) throws Exception{
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(Main.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}

```

Рисунок 4.7 – Класс Main.

Класс *IntSumReducer*.

```

package MapReduce;

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class IntSumReducer extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}

```

Рисунок 4.8 – Класс IntSumReducer.

Класс *TokenizerMapper*.

```

package MapReduce;

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
        StringTokenizer itr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            context.write(word, one);
        }
    }
}

```

Рисунок 4.9 – Класс TokenizerMapper

Для запуска программы необходимо создать две папки «age» и «region», в которых будут храниться файлы с информацией из столбцов «возраст» и «регион». В каждой папке по 20 файлов, в каждом файле по 3000 записей. На первом шаге данные из .xlsx-файла нужно разделить на отдельные файлы с помощью Python-скрипта.

```

import openpyxl

path = r"C:\Users\egor-\OneDrive\Рабочий стол\lab\данные\region"

workbook = openpyxl.load_workbook(path)
sheet = workbook.active

entries_counter = 0
for i in range(20):

    regions = ""
    age = ""

    for j in range(3000):
        if entries_counter >= len(sheet["B"]): break

        regions += sheet["B"][entries_counter].value + "\n"
        age += sheet["H"][entries_counter].value + "\n"

    with open("file_region_" + str(i) + ".txt", "w") as file:
        file.write(regions)

    with open("file_age_" + str(i) + ".txt", "w") as file:
        file.write(age)

```

Рисунок 4.10 – Создание папок «age» и «region»

Далее нужно экспортировать недостающие Jar-файлы для MapReduce как показано на рисунке 4.11.

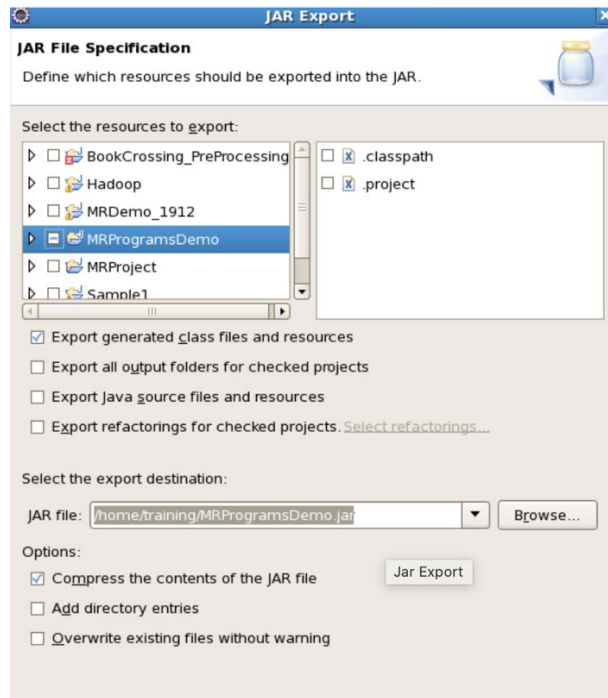


Рисунок 4.11 – Экспорт недостающих jar-файлов

Затем необходимо скомпилировать указанный проект и поместить его в jar-файл. Это можно сделать с помощью сборщика maven (mvn package).

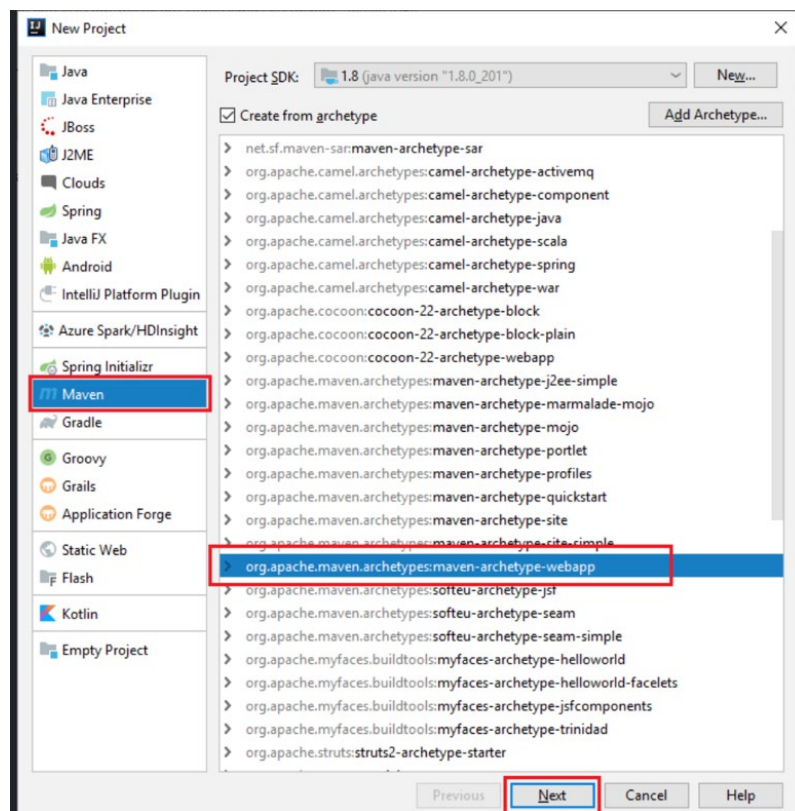


Рисунок 4.12 – Сборка maven пакета

Далее подготавливаем пакеты (.jar) проекта.

```
PS D:\Java_Repo\MapReduceForDistinctValue> mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building MapReduceForDistinctValue 0.0.1-SNAPSHOT
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ MapReduceForDistinctValue ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\Java_Repo\MapReduceForDistinctValue\src\main\resources
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ MapReduceForDistinctValue ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 4 source files to D:\Java_Repo\MapReduceForDistinctValue\target\classes
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ MapReduceForDistinctValue ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\Java_Repo\MapReduceForDistinctValue\src\test\resources
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ MapReduceForDistinctValue ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to D:\Java_Repo\MapReduceForDistinctValue\target\test-classes
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ MapReduceForDistinctValue ---
[INFO] Surefire report directory: D:\Java_Repo\MapReduceForDistinctValue\target\surefire-reports

-----
T E S T S
-----
Running com.bdp.mapreduce.distinct.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.045 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ MapReduceForDistinctValue ---
[INFO] Building jar: D:\Java_Repo\MapReduceForDistinctValue\target\MapReduceForDistinctValue-0.0.1-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
```

Рисунок 4.13 – Подготовка jar-пакета

Затем необходимо разрешить зависимость в файле pom.xml с помощью консоли cmd.

```
pom
<dependency>
<groupId>org.apache.hadoop</groupId>
<artifactId>hadoop-common</artifactId>
<version>2.7.1</version>
</dependency>
<!-- Hadoop Mapreduce Client Core -->
<dependency>
<groupId>org.apache.hadoop</groupId>
<artifactId>hadoop-mapreduce-client-core</artifactId>
<version>2.7.1</version>
</dependency>
<dependency>
<groupId>jdk.tools</groupId>
<artifactId>jdk.tools</artifactId>
<version>${java.version}</version>
<scope>system</scope>
<systemPath>${JAVA_HOME}/lib/tools.jar</systemPath>
</dependency>
```

Рисунок 4.14 – Добавление зависимостей в файл pom.xml

Далее нужно добавить в HDFS наши входные данные как показано на рисунке 4.15.


```
[root@NN ~]# hadoop fs -copyFromLocal/home/NN/HadoopRepo/MapReduce/resources/distinctvalue/sampleddata.txt /user/bdp/mapreduce/distinctvalue/input
```

Рисунок 4.15 – Команда для добавления данных в HDFS

Затем необходимо запустить приложение, как показано на рисунке 4.16.

```
[root@NN ~]# hadoop jar /home/NN/HadoopRepo/MapReduce/MapReduceForDistinctValue-0.0.1-SNAPSHOT.jar com.bdp.mapreduce.distinct.driver.DistinctValueDriver /user/bdp/mapreduce/distinctvalue/input /user/bdp/mapreduce/distinctvalue/output
```

Рисунок 4.16 – Запуск приложения

Приложение MapReduce успешно выполнилось, согласно рисункам 4.17 и 4.18

```
INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8020
INFO input.FileInputFormat: Total input files to process: 1
INFO mapreduce.JobSubmitter: number of splits:1
INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1584546032725_0001
INFO impl.YarnClientImpl: Submitted application application_1584546032725_0001
INFO mapreduce.Job: The url to track the job: http://localhost:8020/jobhistory/job/job_1584546032725_0001/
INFO mapreduce.Job: Running job: job_1584546032725_0001
INFO mapreduce.Job: Job job_1584546032725_0001 running in cluster
INFO mapreduce.Job: map 0% reduce 0%
INFO mapreduce.Job: map 100% reduce 0%
```

Рисунок 4.17 – Выполнение команды MapReduce

```
Reduce shuffle bytes=3558
Reduce input records=314
Reduce output records=139
Spilled Records=628
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=126
CPU time spent (ms)=1220
Physical memory (bytes) snapshot=503894016
Virtual memory (bytes) snapshot=4236754944
Total committed heap usage (bytes)=298319872
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=1669
File Output Format Counters
Bytes Written=2823
```

Рисунок 4.18 – Успешное выполнение программы

В результате получаем два файла «file_age» и «file_region», как показано на рисунке 4.19.

```
$ hadoop fs -ls /Woutputs  
  
Found 2 items  
  
-rw-r--r--  1 training supergroup      2020-07-28 03:36 /file_region.txt  
  
-rw-r--r--  1 training supergroup      2020-07-28 03:53 /file_age.txt
```

Рисунок 4.19 – Директория с выходными данными

Согласно рисунку 4.20, можно увидеть список файлов внутри которых содержатся файлы с данными из таблицы.

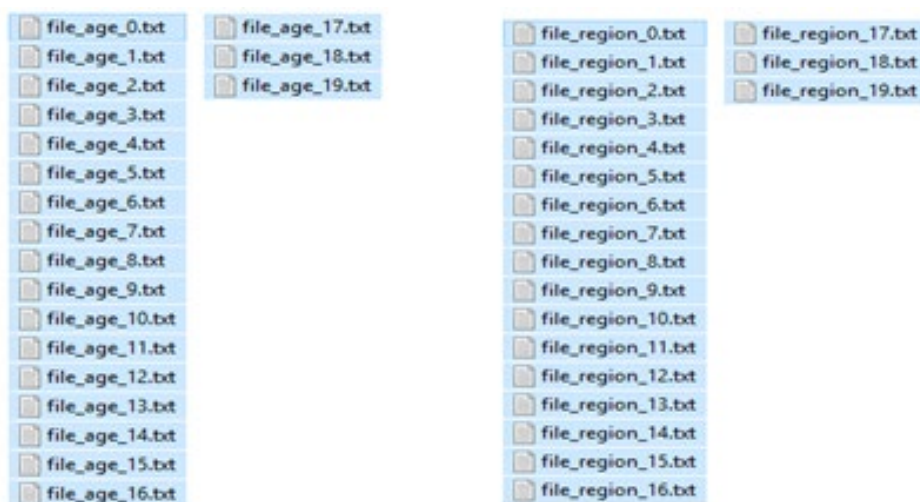


Рисунок 4.20 – Файлы с выходными данными

Используя данные по регионам можно вычислить возраст пациентов и их количество. Были извлечены данные из .xlsx-файла и разделены на части.

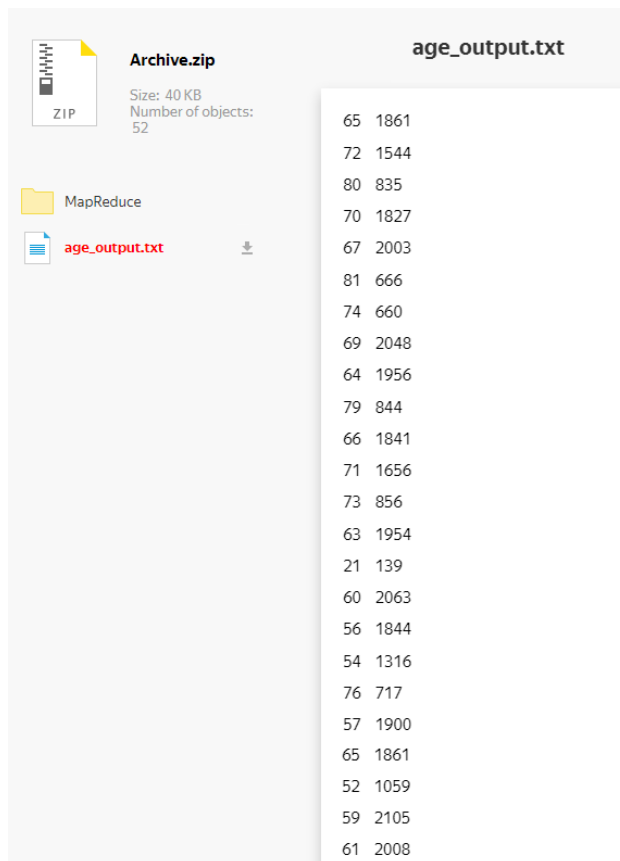


Рисунок 4.21 – Содержимое выходных файлов

В результате обработки содержимое выходных файлов позволило определить количество пациентов по возрасту. Далее можно увидеть результаты подсчета количества по столбцу «Регион» внутри которых содержатся файлы с данными из таблицы.

```

~$ hadoop fs -cat file_region.txt
1 Акмолинская область 6920
2 Акмолинская область 6721
3 Актюбинская область 3300
4 Алматинская область 8400
5 Атырауская область 2090
6 Западно-Казахстанская область 4491
7 Жамбылская область 5003
8 Карагандинская область 11004
9 Костанайская область 5203
10 Кызылординская область 3798
11 Мангистауская область 3197
12 Павлодарская область 6383
13 Северо-Казахстанская область 6088
14 Восточно-Казахстанская область 10045
15 Алматы г.а. 14466
16 Туркестанская область 10958
17 г. Шымкент 2115
18 г.Нур-Султан 9170

```

Рисунок 4.22 – Содержимое файла file_region.txt

В результате обработки содержимое выходных файлов позволило определить количество пациентов по региону.

4.5.2 Администрирование приложений с помощью Spark

Основной целью исследования является задача настройки и развертки универсальной инфраструктуры обучения работе с основными инструментами для работы с БД. К таким инструментам, как правило, относятся Hadoop и входящие в его состав инструменты, такие как HDFS, MapReduce, Spark и т.д. Одним из проектов Apache верхнего уровня является Spark, который также ориентирован на параллельную обработку данных в кластере, однако большое отличие заключается в том, что он работает в памяти. Hadoop читает и записывает файлы в HDFS, а Spark обрабатывает данные в оперативной памяти, используя концепцию, известную как «Resilient Distributed Datasets (RDD)», устойчивый распределенный набор данных. Spark может работать либо в автономном режиме, с кластером Hadoop, служащим источником данных. Spark и Hadoop проекты Apache с открытым исходным кодом, и для Hadoop требуется больше памяти на диске, а Spark требует больше оперативной памяти, а это означает, что настройка кластеров Spark может быть более дорогостоящей. Spark предназначен для улучшения, а не замены стека Hadoop. Spark был разработан для чтения и записи данных из HDFS, а также в другие системы хранения данных, такие как HBase и Amazon S3. Таким образом, пользователи Hadoop могут расширить возможности обработки данных, объединив Spark с Hadoop MapReduce, HBase и другими фреймворками больших данных [131]. Существуют три способа развертывания Spark в кластере Hadoop: автономный режим (Standalone), Over Yarn и Spark в MapReduce (SIMR) как показано на рисунке 4.23.

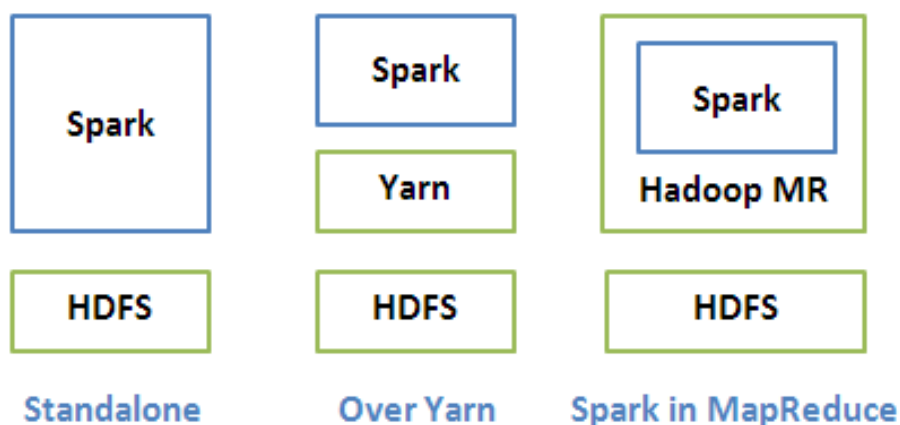
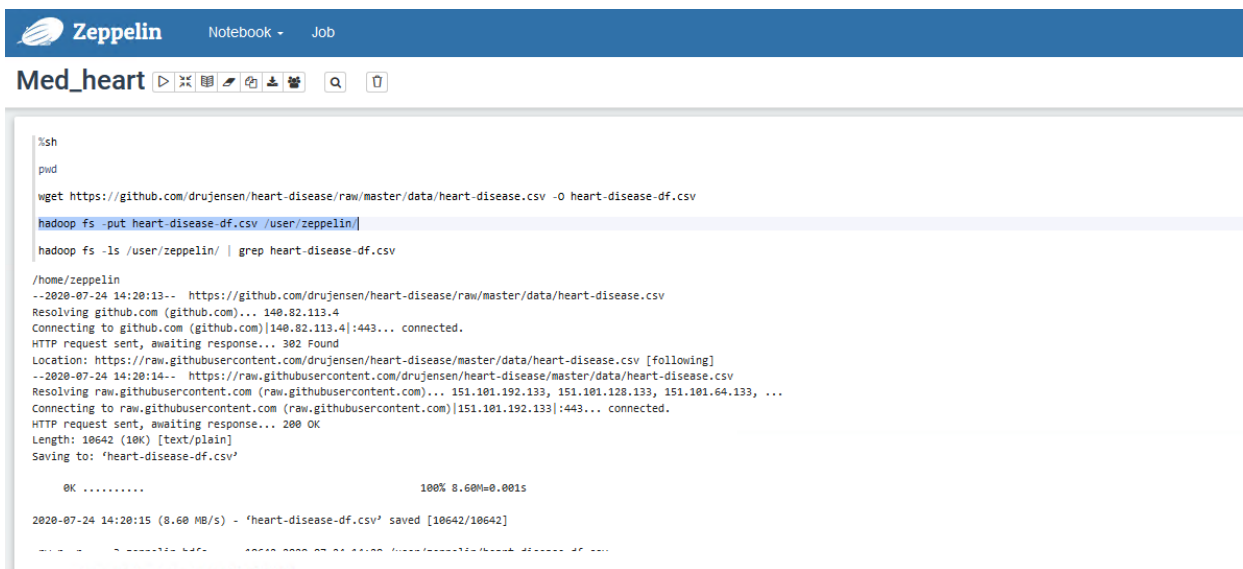


Рисунок 4.23 – Три способа развертки Spark в кластере Hadoop.

С помощью автономного развертывания можно статически распределять ресурсы на всех машинах в кластере Hadoop и запускать Spark одновременно с Hadoop MapReduce. Затем пользователь может выполнять произвольные задания Spark для своих данных HDFS. Также для развертки Hadoop Yarn можно просто запустить Spark на Yarn без предварительной установки или административного доступа. Это позволяет пользователям

легко интегрировать Spark в свой стек Hadoop и использовать преимущества полной мощности Spark, а также других компонентов, работающих поверх Spark. SIMR используется для запуска заданий Spark в дополнение к автономному развертыванию. С помощью SIMR можно начать работу с Spark и использовать его оболочку без какого-либо административного доступа [132, 133]. Apache Spark обладает обширными инструментами для обработки данных и все эти факторы позволяют использовать как эффективный инструмент для параллельной обработки множества данных в кластере Hadoop, и повысить скорость обработки.

Анализ данных с использованием Apache Spark. Для работы с данными были использованы датасеты онлайн сообщества «Kaggle» [134], где предоставляются различные датасеты необходимые для выполнения работы с данными. Для обработки был взят датасет о сердечно сосудистых заболеваниях, где были выбраны основные показатели анализов больных. Графическая оболочка Apache Zeppelin использовалась для взаимодействия со Spark, которая обеспечивает представление данных в удобном формате визуализации. Для этого в первую очередь нужно загрузить файлы в HDFS, и выполнить команду `//hadoop fs -put heart-disease-df.csv /user/zeppelin/`, как показано на рисунке 4.24.



```
%sh
pwd
wget https://github.com/drujensen/heart-disease/raw/master/data/heart-disease.csv -O heart-disease-df.csv
hadoop fs -put heart-disease-df.csv /user/zeppelin/
hadoop fs -ls /user/zeppelin/ | grep heart-disease-df.csv

/home/zeppelin
--2020-07-24 14:20:13-- https://github.com/drujensen/heart-disease/raw/master/data/heart-disease.csv
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/drujensen/heart-disease/master/data/heart-disease.csv [following]
--2020-07-24 14:20:14-- https://raw.githubusercontent.com/drujensen/heart-disease/master/data/heart-disease.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.192.133, 151.101.128.133, 151.101.64.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.192.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10642 (10K) [text/plain]
Saving to: 'heart-disease-df.csv'

OK .....                               100% 8.60M=0.001s

2020-07-24 14:20:15 (8.60 MB/s) - 'heart-disease-df.csv' saved [10642/10642]
```

Рисунок 4.24 – Загрузка файл в HDFS.

Далее полученный файл необходимо добавить в датасет для дальнейшей обработки, а также добавить библиотеки для обработки данных как показано на рисунке 4.25.

```

%pyspark
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
from pyspark.sql.context import SQLContext
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from pyspark.sql import SparkSession

df = spark.read.csv('heart-disease-df.csv', header=True)

Took 1 sec. Last updated by admin at July 24 2020, 2:27:43 PM. (outdated)

```

Рисунок 4.25 – Загрузка данных и библиотек.

Далее нужно получить количество строк, которое было добавлено в датасет, и просмотреть содержимое, как показано на рисунке 4.26.

```

%pyspark
df.count()

294

Took 0 sec. Last updated by admin at July 24 2020, 3:29:49 PM.

%pyspark
df.show()

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|age|sex|cp|trestbps|chol|fbs|restecg|thalach|exang|oldpeak|slope|ca|thal|num|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 28| 1| 2| 130| 132| 0| 2| 185| 0| 0| ?| ?| ?| 0|
| 29| 1| 2| 120| 243| 0| 0| 160| 0| 0| ?| ?| ?| 0|
| 29| 1| 2| 140| ?| 0| 0| 170| 0| 0| ?| ?| ?| 0|
| 30| 0| 1| 170| 237| 0| 1| 170| 0| 0| ?| ?| ?| 6|
| 31| 0| 2| 100| 219| 0| 1| 150| 0| 0| ?| ?| ?| 0|
| 32| 0| 2| 105| 198| 0| 0| 165| 0| 0| ?| ?| ?| 0|
| 32| 1| 2| 110| 225| 0| 0| 184| 0| 0| ?| ?| ?| 0|
| 32| 1| 2| 125| 254| 0| 0| 155| 0| 0| ?| ?| ?| 0|
| 33| 1| 3| 120| 298| 0| 0| 185| 0| 0| ?| ?| ?| 0|
| 34| 0| 2| 130| 161| 0| 0| 190| 0| 0| ?| ?| ?| 0|
| 34| 1| 2| 150| 214| 0| 1| 168| 0| 0| ?| ?| ?| 0|
| 34| 1| 2| 98| 220| 0| 0| 150| 0| 0| ?| ?| ?| 0|
| 35| 0| 1| 120| 160| 0| 1| 185| 0| 0| ?| ?| ?| 0|
| 35| 0| 4| 140| 167| 0| 0| 150| 0| 0| ?| ?| ?| 0|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

Took 1 sec. Last updated by admin at July 24 2020, 3:29:55 PM.

```

Рисунок 4.26 – Количество строк в датасете.

Также можно проверить какие именно поля были выбраны и названия их заголовков как показано на рисунке 4.27.

```

%pyspark
df.columns

['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'num']

Took 0 sec. Last updated by admin at July 24 2020, 2:27:51 PM.

```

Рисунок 4.27 – Наименование полей.

Далее необходимо определить, какая возрастная группа чаще всего подвержена сердечно сосудистым заболеваниям. На рисунке 4.28 получаем информацию обо всех возрастах участников исследования, и в дальнейшем визуализируем эти результаты.

```
%pyspark
#необходимо определить возрастную группу, которая максимально подвержена сердечно сосудистым заболеваниям
age_val = (df.groupBy('age').count().sort('age').cache())
age_val.count()

age_val_t = (age_val.toPandas().sort_values(by=['count'],ascending=False))
age_val_t

  age  count
26  54    25
20  48    19
24  52    17
27  55    15
21  49    15
18  46    13
25  53    12
22  50    12
15  43    12
11  39    11
13  41    11
19  47    10
28  56    10
30  58     9
23  51     9
31  59     8
..  ..     ~

Took 4 sec. Last updated by admin at July 24 2020, 2:27:57 PM. (outdated)
```

Рисунок 4.28 – Загрузка файл в HDFS.

На рисунке 4.29 представлена визуализация данных через инструмент seaborn для визуализации данных.

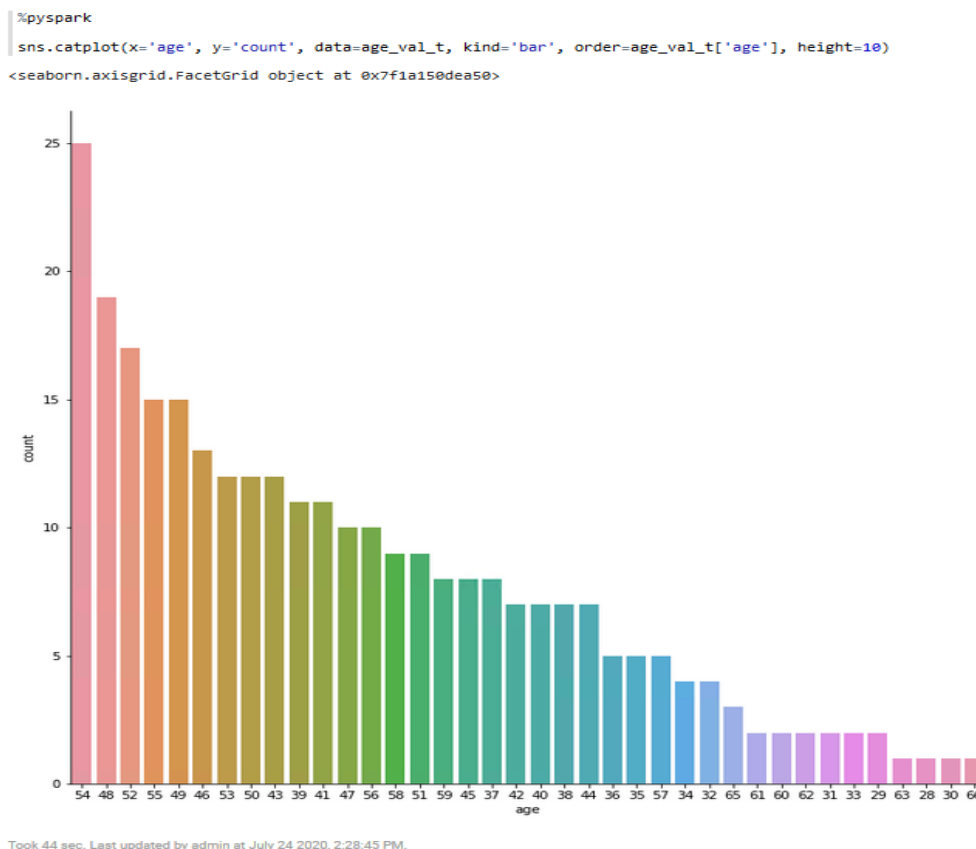


Рисунок 4.29 – Визуализация возрастной группы.

После того как была определена возрастная группа, необходимо получить данные о максимальных количествах сердечных сокращений у данной группы больных. В первую очередь нам нужно получить данные о больных, чей возраст равен 54 годам, и после этого посчитать у какого количества больных пиковые сокращения.

```
%pyspark
#В предыдущем анализе мы обнаружили что возрастная группа 54 года, чаще всего подвержена ССЗ, нужно определить максимально число сердечных сокращений
age_logs_val = df.filter(df["age"] == 54).cache()
thalach_log_files = (age_logs_val.groupBy("thalach").count().sort("count", ascending=False).limit(10))

thalach_val_t = (thalach_log_files.toPandas().sort_values(by=['count'],ascending=False))
thalach_val_t
```

thalach	count	
0	140	3
1	130	3
2	150	2
3	142	2
4	122	2
5	137	2
6	118	1
7	110	1
8	155	1
9	100	1

Took 1 sec. Last updated by admin at July 24 2020, 3:04:09 PM. (outdated)

Рисунок 4.30 – Количество пиковых сокращений.

Далее необходимо визуализировать эти данные, ниже на рисунке представлен результат визуализации.

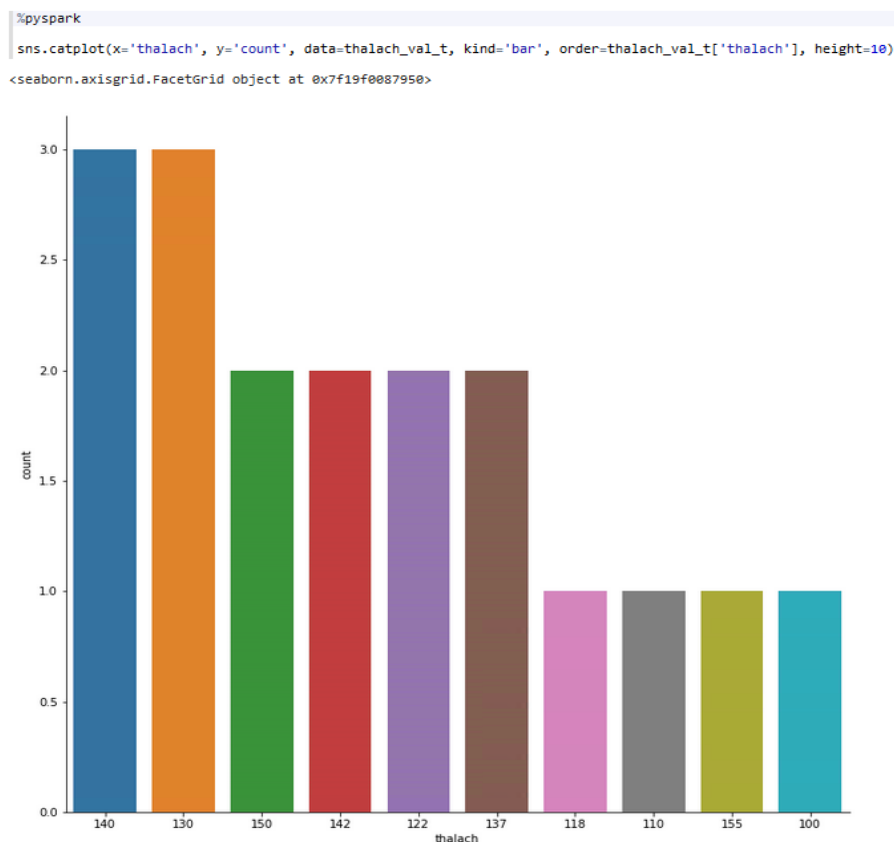


Рисунок 4.31 – Загрузка файл в HDFS.

Apache Spark обладает обширными инструментами для обработки данных и все эти факторы позволяют использовать как эффективный инструмент для параллельной обработки множества данных в кластере, и повысить скорость обработки. В данном эксперименте были использованы датасеты онлайн сообщества «Kaggle», о сердечно сосудистых заболеваниях, где были выбраны основные показатели анализов больных. Графическая оболочка Apache Zeppelin использовалась для взаимодействия со Spark, которая обеспечивает представление данных в удобном формате визуализации.

4.6 Программная реализация информационной системы

Разработанная система определяет статус больного, и позволяет врачу-эндокринологу следить за течением болезни. Тем самым, ИС позволяет обновлять историю болезни пациента с новыми поступающими данными. Созданная нереляционная база данных автоматически создает репликации и является отказоустойчивой, а также имеет доступ к подключенным медицинским базам данных. Также система может быть использована студентами медицинских вузов в виде симулятора для повышения знаний при диагностировании СД и сопровождения лечения больного. Разработанная ИС позволит облегчить работу врачей-эндокринологов, т.к. ими могут быть использованы не только собственные знания, но и мировые ресурсы базы знаний о сахарном диабете. Алгоритм диагностирования болезни приведен на рисунке 4.32.

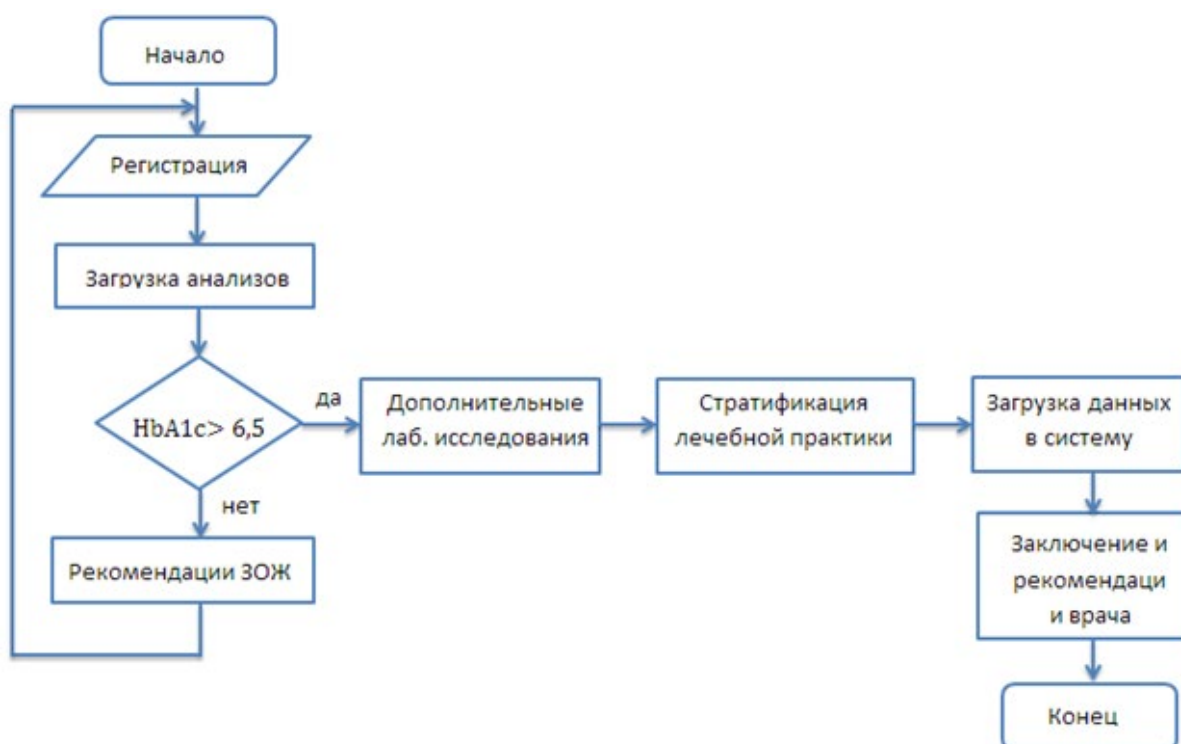


Рисунок 4.32 – Алгоритм функционирования системы

Следующим этапом является построение системы в виде удобного для пользователя интерфейса, где будут осуществляться различные операции по диагностике диабета, такие как:

- вводятся различные регистрационные данные пациента;
- создаются медицинские карты пациента;
- вводятся анализы, где каждый анализ состоит из различных числовых и текстовых параметров;
- если диагноз подтверждается, то для госпитализации и лечения нужно сдать дополнительно 12 видов анализов и обследований.
- подключение различных медицинских базы данных для владения полной информацией о СД всех видов, а также чтобы врач-эндокринолог был в курсе последних важных публикациях о диабете. Так как врач должен владеть всей актуальной информацией о методах лечения. Полная разработка интерфейса ИС подробно описана в Приложении Б.

Загрузка данных в MongoDB через пользовательский интерфейс. MongoDB – это нереляционная база данных, которая удобна в использовании, где нужна гибкость структуры и используются большие объемы данных различных форматов. Для работы по этой теме диссертации она отлично позволяет хранить истории болезни пациентов в масштабе не только медицинского учреждения, но и страны. Любая карта пациента может быть различного типа с множеством полей, и они будут постоянно увеличиваться. Для классических реляционных баз данных это выливается в весьма нетривиальную задачу горизонтального масштабирования, которая в MySQL решается через перенастройку сервера, а в PostgreSQL через специальную промежуточную таблицу. Горизонтальный рост и ввод новых узлов кластера сопряжен с большими трудностями и плохо автоматизируется для реляционных баз данных. Также классические базы данных очень плохо работают со смешанной нагрузкой, когда у вас запись/чтение примерно 1:1 и данных очень много. Это вызывает непрерывное перестроение индексов, и их использование имеет больше недостатков. Это тот тип нагрузки, при которой выбираемая подсистема низкого уровня в СУБД MySQL очень часто повреждается без возможности восстановления, тем самым вызывая ощутимый простой на реорганизацию структур данных.

После пошаговой установки системы управления данных с открытым исходным кодом MongoDB для неструктурированных данных появляется окно хранилища данных в соответствии с рисунком 4.33.

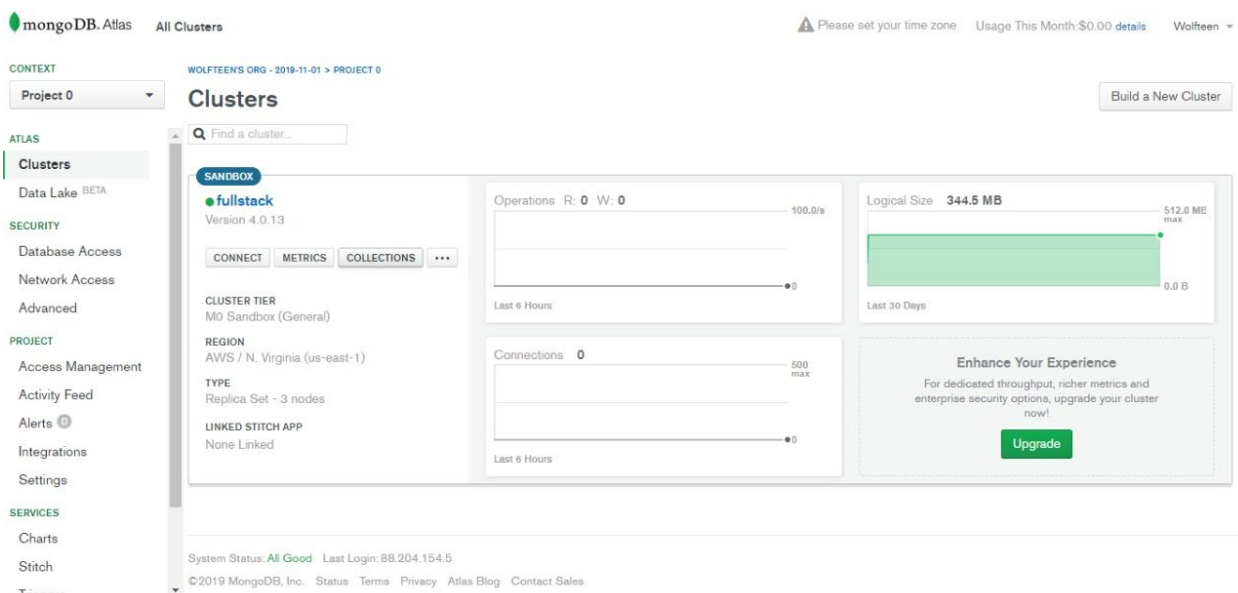


Рисунок 4.33 – Структура БД MongoDB

Процесс загрузки параметров, снимков различных форматов в базу данных MongoDB, подробно описана в Приложении В.

4.7 Создание и загрузка неструктурированных данных в MongoDB

Для загрузки неструктурированной информации различных форматов в MongoDB в режиме ручного управления необходимо установить следующие инструменты:

1. NoSQLBooster. NoSQLBooster для MongoDB - это кросс-платформенный инструмент GUI для MongoDB v2.6-4.2, ориентированный на оболочку, который предоставляет комплексные инструменты мониторинга серверов, свободный построитель запросов [135];

2. Putty. Putty – это бесплатный эмулятор терминала с открытым исходным кодом, последовательная консоль и приложение для передачи файлов по сети. Поддерживает сетевые протоколы, включая SSH, а также может подключаться к последовательному порту.

Установка NoSQLBooster и Putty. Для начала необходимо скачать Putty, затем установить его на сервере и указать параметры (номер порта, IP адрес, выбрать машину). Настроив все конфигурационные настройки, мы запускаем соединение с сервером. После того как прописывается IP адрес и номер порта выбираем защищенное соединение SSH, затем выбрав сервер в данном случае Node 1 можно запускать соединение. Получив приглашение, заходим под пользователем root и вводим пароль.

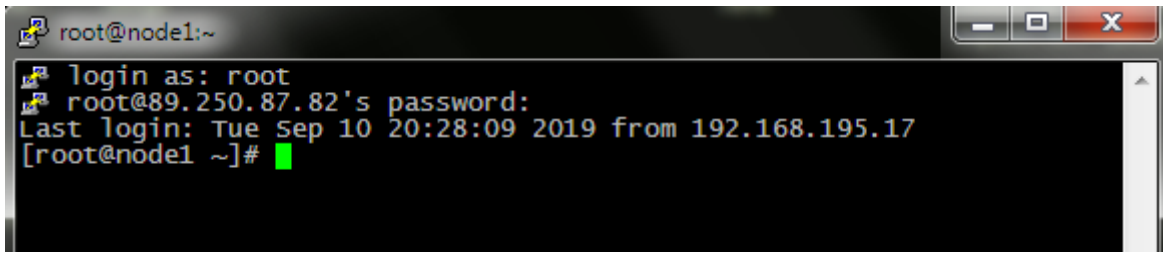


Рисунок 4.34 – Успешное соединение

Настроив защищенное соединение, можем устанавливать NoSQLBooster для MongoDB, предварительно загрузив его с официального сайта.

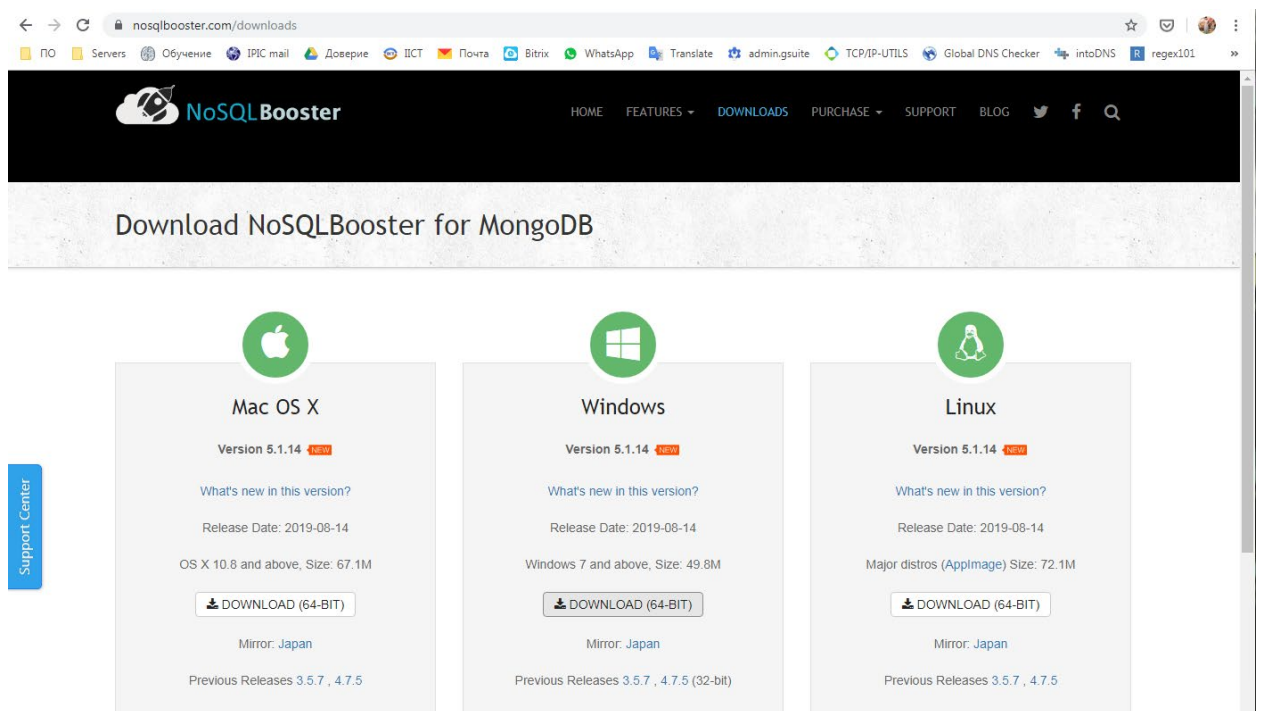


Рисунок 4.35 – Версия «NoSQLBooster for MongoDB» для Windows

После установки и настройки NoSQLBooster for MongoDB запускаем приложение, и появляется окно запросов, приведенное на рисунке 4.36.

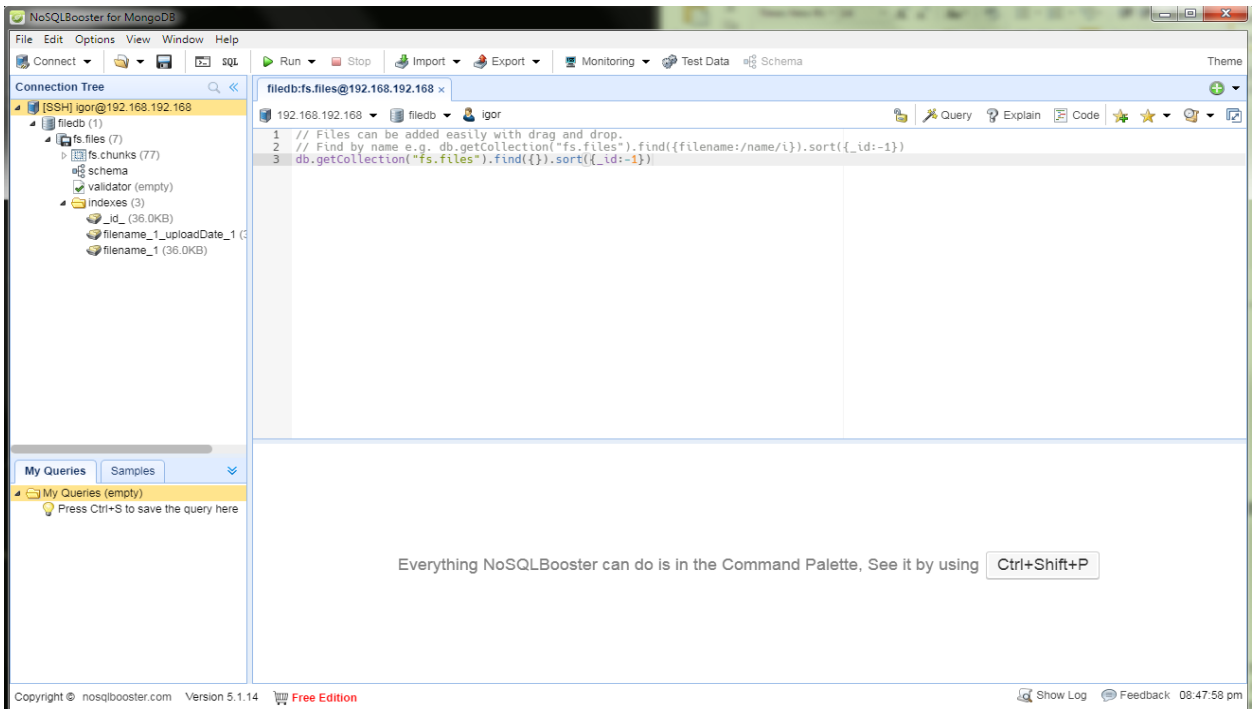


Рисунок 4.36 – Главное окно «NoSQLBooster for MongoDB»

После этого, методом ручного импорта производим загрузку всех неструктурированных данных различных форматов. Для начала была создана база данных «fs.files», где были загружены статьи, монографии, журналы форматов «.pdf, .doc, .txt.».

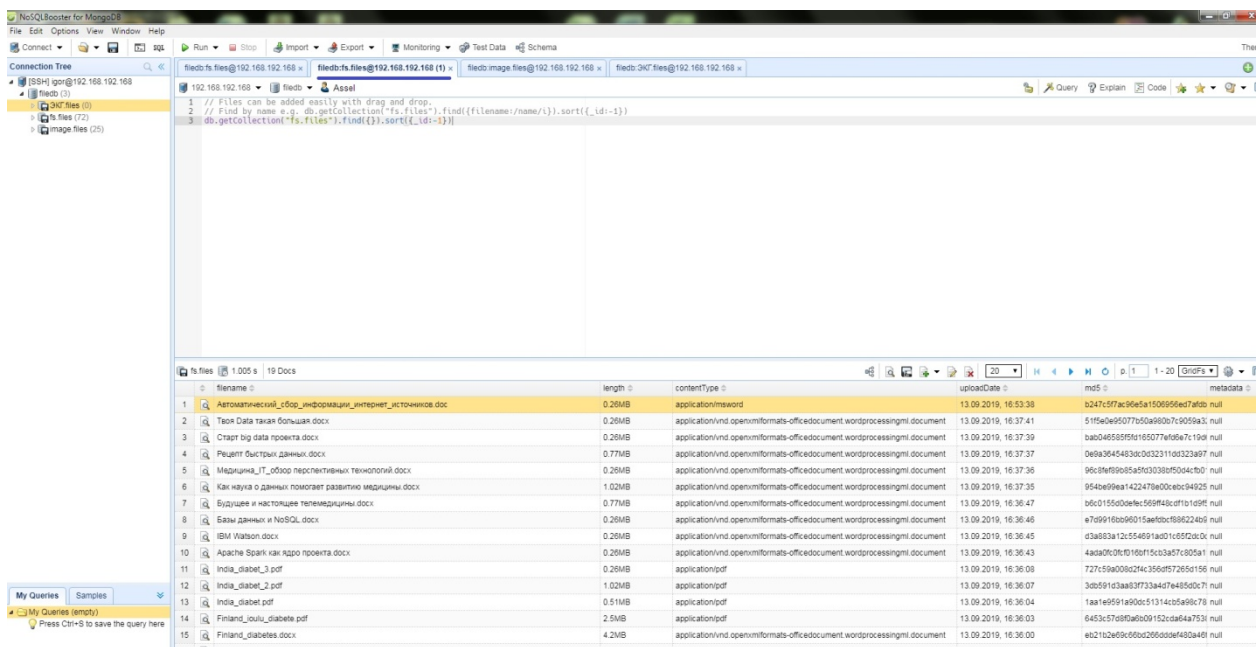


Рисунок 4.37 – Папка «fs.files»

Далее была создана база данных «УЗИ ОБП и почек.files» для размещения в ней снимков различных форматов.

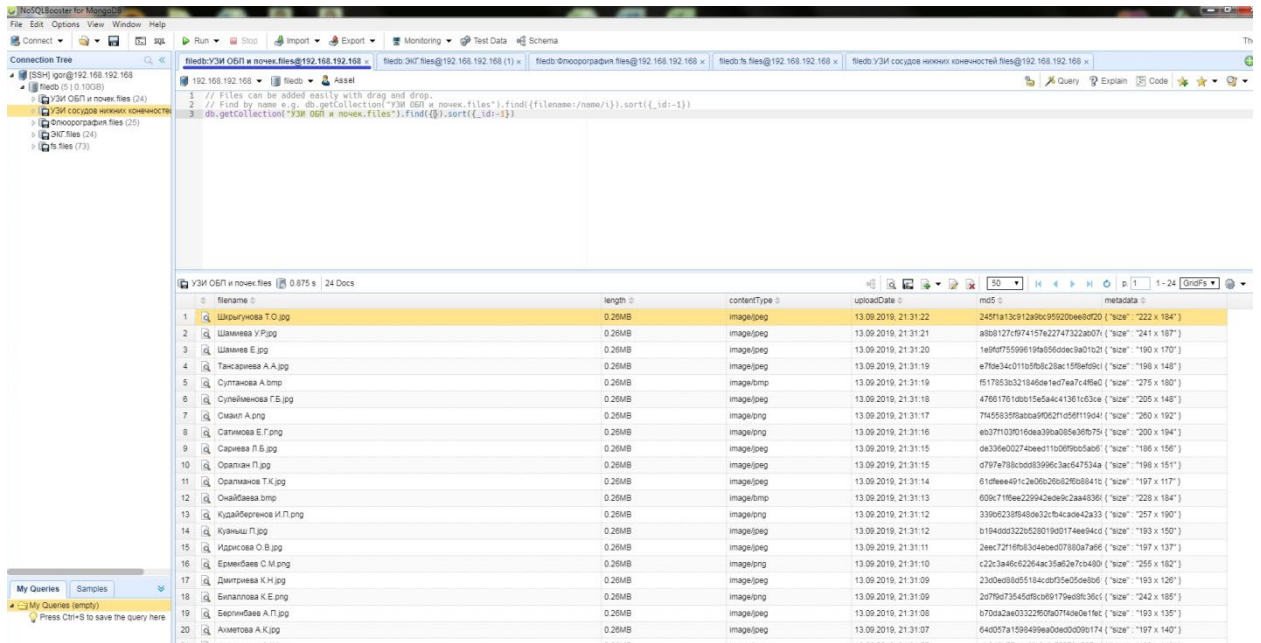


Рисунок 4.38 – Базы данных «УЗИ ОБП и почек.files»

Таким же образом была создана база данных «Флюорография.files».

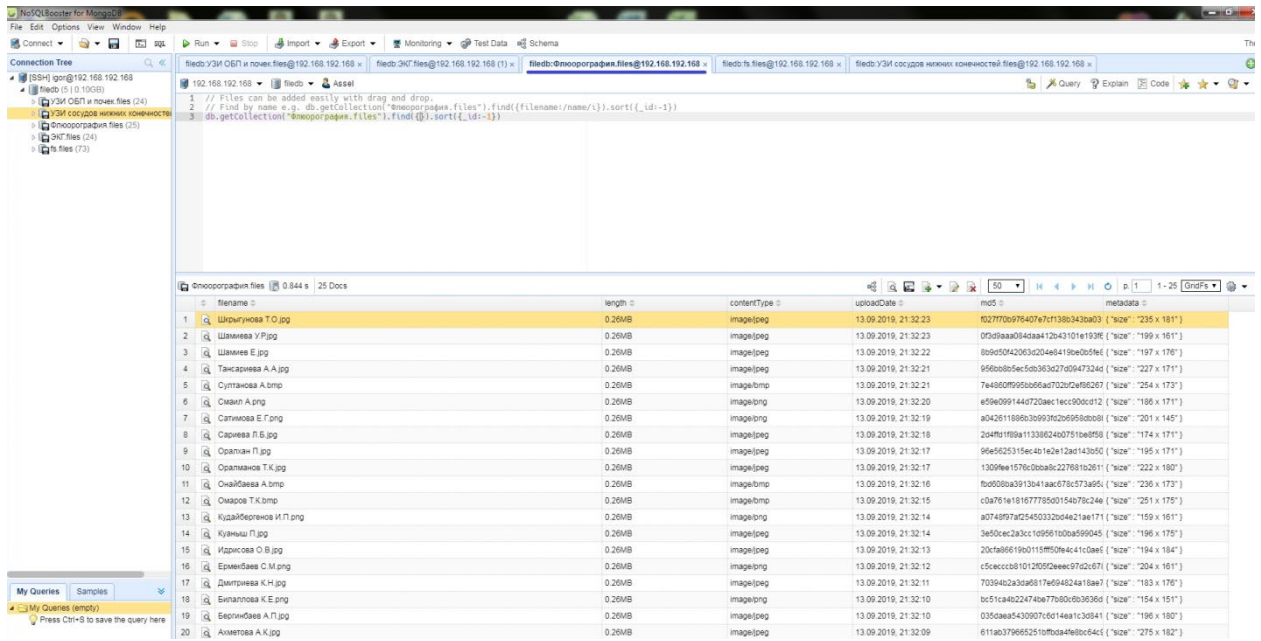


Рисунок 4.39 – Базы данных «Флюорография.files»

Далее была создана база данных «ЭКГ.files» для размещения в ней снимков различных форматов.

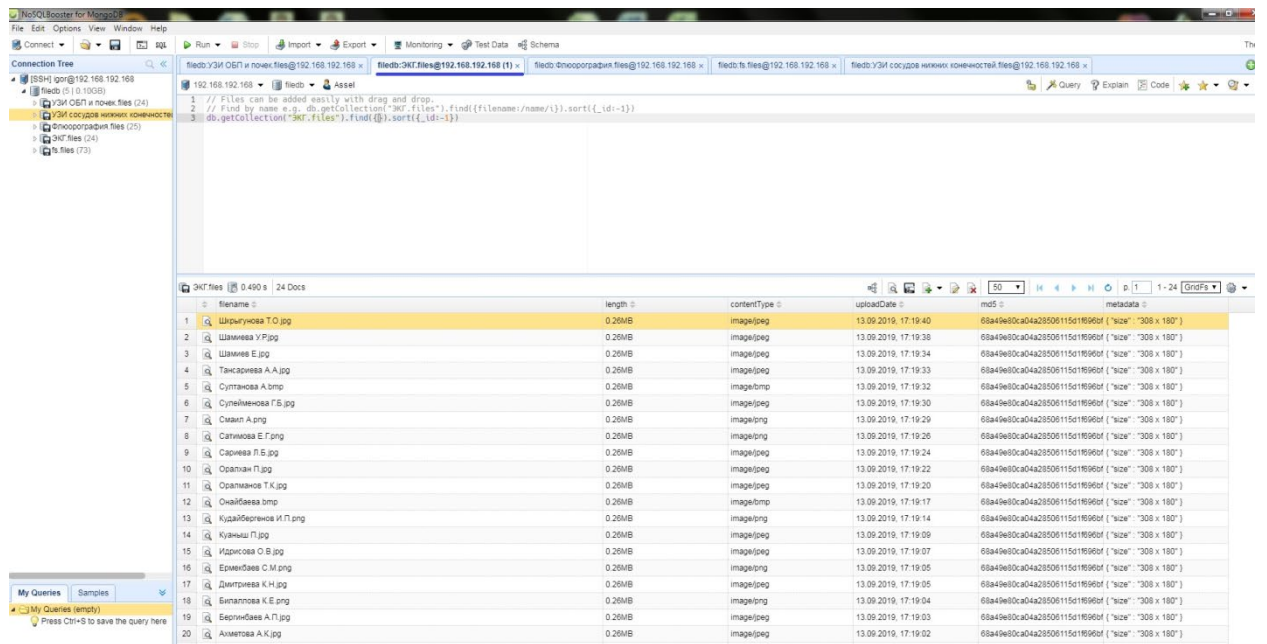


Рисунок 4.40 – Базы данных «ЭКГ.files»

Таким же образом была создана база данных «УЗИ сосудов нижних конечностей.files».

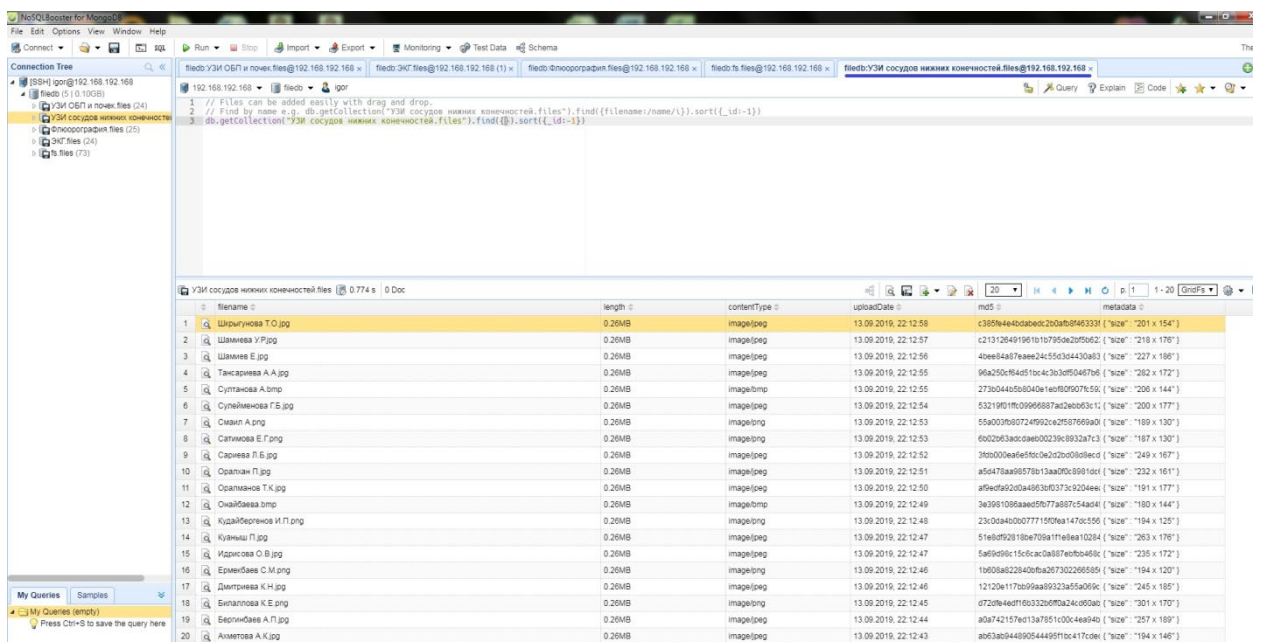


Рисунок 4.41 – Базы данных «УЗИ сосудов нижних конечностей.files»

Далее была создана база данных «FreeStyleLibre.files» для загрузки показателей уровня глюкозы в виде графиков полученные от элементов системы FreeStyleLibre.

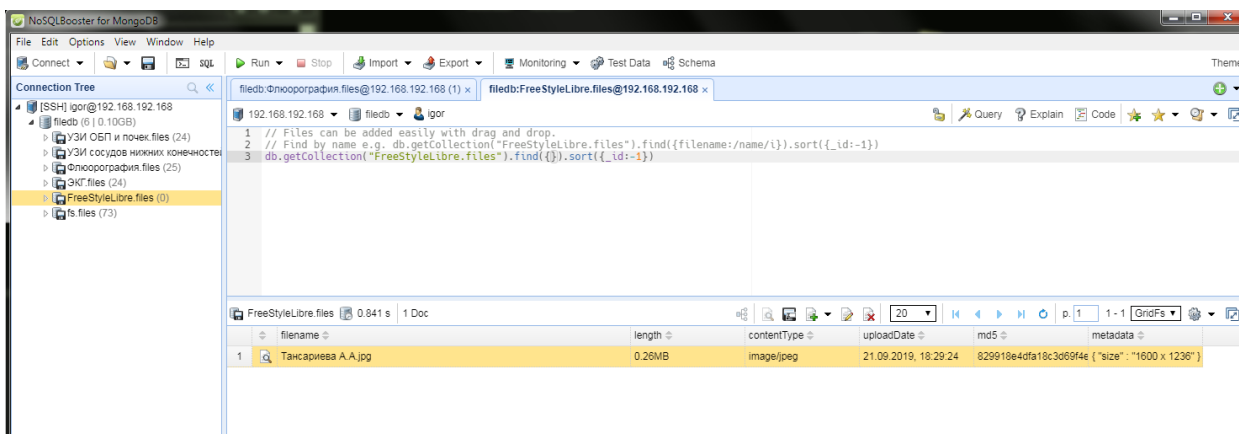


Рисунок 4.42 – Базы данных «FreeStyleLibre.files»

В соответствии с рисунком можно увидеть, что все папки были созданы и загружены снимки, графики, отчеты с указанными фамилиями пациентов различных форматов в ручном режиме.

Выводы по четвертому разделу

1. Исследована математическая модель обработки и поиска больших данных.
2. Разработана экосистема Nadoop на базе инструментов BigData технологии для распределенной системы обработки данных.
3. Разработана не реляционная база данных MongoDB для хранения неструктурированных медицинских данных.
4. В экосистеме Nadoop произведена обработка данных с помощью модели MapReduce и Spark.
5. Основные результаты раздела опубликованы в [136-139].

ЗАКЛЮЧЕНИЕ

Краткие выводы по результатам диссертационных исследований.

В данной диссертационной работе применены методы анализа данных, основанные на статистическом анализе для прогнозирования возможного количества пациентов с СД по РК; были проведены серии регрессионных уравнений, где были включены различные параметры для выявления факторов имеющих влияние на рост пациентов. В разработанной ИС реализованы функции хранения и обработки, репликации структурированных и неструктурированных данных, возможность управления системой в автоматическом и ручном режимах. По результатам исследования получены следующие результаты:

1) рассмотрены актуальные проблемы диагностики и исследованы причины распространенности СД в мире. Произведен сравнительный анализ современных медицинских ИС диагностики во всем мире, которые продемонстрировали успешные результаты при диагностировании болезней. Были исследованы источники и характеристики информационных ресурсов по СД;

2) на основе проведенного анализа применения БД в различных направлениях позволило обнаружить все скрытые шаблоны, которые привели к более эффективному маркетингу, качественному обслуживанию, повышению эффективности работы, конкурентным преимуществам в данном исследовании. Рассмотрены методы и алгоритмы, применяемые в области аналитики медицинских данных. Были определены инструменты для накопления, управления, анализа и усвоения больших объемов разнородных, структурированных и неструктурированных данных, созданных с помощью современных систем здравоохранения. Был осуществлен анализ рынка используемых ИТ для решения медицинских задач;

3) предложены и обоснованы новые методы для решения задач выявления наиболее эффективных методов регрессионного анализа для прогнозирования роста пациентов СД на фоне пассивного выявления по РК. Выявлена тесная взаимосвязь между количеством больных СД и ростом населения, увеличением ВРП, ВВП, ИУЖ по г. Алматы;

4) разработана информационная модель диагностирования СД с помощью БД. На основе математического аппарата показан перечень стандартных медицинских анализов, с помощью которых можно диагностировать диабет. При этом, установлены не только граничные критерии диагностики, но и обоснован механизм параллельных вычислений разнотипных данных для определения вероятности постановки диагноза.

5) разработана и настроена платформа ИС, где используется программный продукт Apache Ambari, который позволяет управлять и контролировать кластером Hadoop. В кластере Hadoop, создана нереляционная база данных, где используются и обрабатываются

неструктурированные данные различных форматов. Создана распределенная файловая система HDFS.

Оценка полноты решения поставленных задач. Поставленные перед диссертантом задачи были решены:

– выполнен анализ современного состояния применения методов прогнозирования и использования технологий BigData в сфере медицинского обслуживания.

– разработан метод прогнозирования роста пациентов с помощью методов регрессионного анализа и с применением библиотеки scikit-learn для определения количества закупаемого инсулина.

– разработана система на основе алгоритмического и программного обеспечения по диагностике СД.

– выполнены экспериментальные исследования обработки больших данных, проведен анализ полученных результатов и сформулированы выводы по работе в целом.

Рекомендации и исходные данные по конкретному использованию результатов. Предложенные алгоритмы расчета количества возможных пациентов на 2019 год по РК с помощью моделей регрессионного анализа реализованы на основе применения библиотек scikit-learn языка программирования Python. Разработанная модель прогнозирования количества пациентов СД с использованием технико-экономических показателей позволяет заранее запланировать количество медикаментов для закупки в бюджет. В следствии было подтверждено общее качество регрессионных уравнений, стандартным тестом Фишера, критерием Стьюдента, также были выполнены проверки выполнимости предпосылок МНК на условия теоремы Гаусса-Маркова. В дополнение к этому, была разработана структурная модель процессов поиска, извлечения, обработки и анализа данных и ИС для диагностики сахарного диабета на базе инструментов BigData технологии была апробирована на основе системного подхода. Практической ценностью информационной системы является:

- данная ИС позволяет управлять работой кластера Hadoop;

- каждый инструмент технологии BigData способствует созданию и подключению различных баз данных для неструктурированных и структурированных данных и их взаимодействию между собой;

- полная отказоустойчивость разработанной ИС, непревзойденная производительность систем хранения данных, пакет дополнительных услуг, разнообразие учета ресурсов и удобный портал самообслуживания;

- при одновременном подключении к серверу пользователей ИС работает асинхронно, то есть ставит приоритеты и распределяет ресурсы грамотнее, что повышает эффективность выполнения задачи;

- система может быть использована студентами медицинских вузов в виде симулятора для повышения знаний при диагностировании диабета.

Практическое применение данной ИС заключается в его возможной реализации для врачей-эндокринологов, т.к. ИС позволит облегчить работу

при определении статуса больного, в сопровождении лечения, а также для прогнозирования роста пациентов.

Оценка экономической эффективности технологии. На основе множества допустимых решений статистических задач, полученные результаты позволили спрогнозировать количество пациентов и пришли к выводу, что математические методы регрессионного анализа являются эффективными методами решения различных медицинских задач.

Как было описано в диссертационной работе, организация эндокринологической помощи в РК, направлена на снижение заболеваемости и инвалидизации пациентов с эндокринными заболеваниями. С целью повышения качества и активной продолжительности жизни граждан страны, вследствие чего проведение и участие в скринингах по социально-значимым заболеваниям, таким как СД несет общую экономическую выгоду для бюджета страны.

На основании полученных результатов испытаний, ИС определяет статус, а также хранит все истории болезни пациента. Программное обеспечение позволяет производить удаленный доступ через Rest API, и созданная база данных создает репликацию и является отказоустойчивой, что снижает риски потери данных. Также система может быть использована студентами медицинских вузов в виде симулятора для повышения знаний при диагностировании диабета. ИС существенно облегчает процесс диагностирования и ведения лечения болезни у пациента, тем самым увеличивая общую эффективность.

Оценка научного уровня выполненной работы в сравнении с лучшими достижениями в данной области. На основании выполненных исследований получены следующие новые результаты:

- разработана методика применения статистических моделей на основе регрессионного анализа в медицине;
- разработан метод прогнозирования технико-экономических показателей в сфере медицинских услуг по диабету;
- разработана структурная модель процессов поиска, извлечения, обработки и анализа данных;
- разработана структура ИС для диагностики СД на основе BigData технологий.

На основе детального анализа литературных источников, а также по результатам апробирования системы, можно прийти к выводу, что разработанная ИС для определения статуса СД выдает адекватные результаты, что говорит о том, что диссертационная работа отвечает современным требованиям науки и техники.

Результаты, полученные в ходе выполнения диссертационной работы подтверждены публикациями и апробированы в материалах и сборниках международных научно-практических конференций, что обосновывает достоверность полученных результатов и их научную значимость.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 ВОЗ. Вопросы здравоохранения: диабет. <https://www.who.int/diabetes/global-report/en/>. 01.06.2020.
- 2 Диабет во всем мире. <https://www.diabetesaustralia.com.au/diabetes-globally>. 01.06.2020.
- 3 Глобальные показатели диабета растут по мере распространения ожирения. <https://www.nytimes.com/2015/06/08/health/research/global-diabetes-rates-are-rising-as-obesity-spreads.html>. 01.06.2020.
- 4 ВОЗ. Ключевые факты. <https://www.who.int/news-room/fact-sheets/detail/diabetes>. 01.06.2020.
- 5 Неинфекционные заболевания и душевное здоровье. World health statistics 2018: monitoring health for the SDGs, sustainable development goals. Geneva: World Health Organization; 2018. Licence: CC BY-NC-SA 3.0 IGO.
- 6 ВОЗ. Всемирный день здоровья 2016: победить диабет. <https://www.who.int/campaigns/world-health-day/2016/en/>. 01.06.2020.
- 7 ВОЗ. Ключевые сообщения. Global report on diabetes. World Health Organization. ISBN 978 92 4 156525 7 (NLM classification: WK 810).
- 8 Время остановить распространение диабета во всем мире. <https://www.raconteur.net/healthcare/time-to-halt-worldwide-spread-of-diabetes>. 01.06.2020.
- 9 Диабет: мир в опасности. <https://geographical.co.uk/people/development/item/2235-the-world-at-risk>. 01.06.2020.
- 10 Глобальная эпидемия диабета, принесенная глобальным развитием. <https://www.theatlantic.com/health/archive/2012/07/the-global-diabetes-epidemic-brought-to-you-by-global-development/259305/>. 01.06.2020.
- 11 МДФ. Эпидемиология и исследования. IDF Diabetes Atlas 8th Edition. <https://www.idf.org/e-library/epidemiology-research/diabetes-atlas.html>. 01.06.2020.
- 12 МДФ. Факты и цифры о диабете. <https://www.idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html>. 01.06.2020.
- 13 Всемирный день борьбы с диабетом. <https://www.calend.ru/holidays/0/0/100/>. 01.06.2020.
- 14 Тажиева А.Е. Состояние и совершенствование организации амбулаторно-поликлинической помощи взрослым больным сахарным диабетом 2 типа в г. Алматы: дисс. на соискание степени доктора философии PhD. – Алматы, 2018. – 13 с.
- 15 Всемирный банк. Распространенность диабета. <https://data.worldbank.org/indicator/SH.STA.DIAB.ZS?end=2017&locations=IM-OE-TR-AU-FR-DE-GB-US-IT-LV&start=2017&view=bar>. 01.06.2020.
- 16 Lindsay M Jaacks, Stefanie Vandevijvere, An Pan, Craig J McGowan, Chelsea Wallace, Fumiaki Imamura, Dariush Mozaffarian, Boyd Swinburn, Majid Ezzati, The obesity transition: stages of the global epidemic, Lancet Diabetes Endocrinol 2019.

17 Wang Q, Zhang X, Fang L, Guan Q, Guan L, Li Q (2018) Prevalence, awareness, treatment and control of diabetes mellitus among middleaged and elderly people in a rural Chinese population: A cross-sectional study. PLoS ONE 13 (6): e0198343. <https://doi.org/10.1371/journal.pone.0198343>.

18 Ramachandran A, Mary S, Yamuna A, Murugesan N, Snehalatha C. High prevalence of diabetes and cardiovascular risk factors associated with urbanization in India. *Diabetes Care* 2008;31:893–898

19 Raval A, Dhanaraj E, Bhansali A, Grover S, Tiwari P. Prevalence and determinants of depression in type 2 diabetes patients in a tertiary care centre. *Indian J Med Res.* 2010;132:195–200.

20 Misra A, Pandey RM, Devi JR, Sharma R, Vikram NK, Khanna N. High prevalence of diabetes, obesity and dyslipidaemia in urban slum population in northern India. *Int J Obes Relat Metab Disord.* 2001;25:1722–9.

21 Ramachandran A, Snehalatha C, Vijay V, King H. Impact of poverty on the prevalence of diabetes and its complications in urban southern India. *Diabet Med.* 2002;19:130–5.

22 Gutch M, Mohd Razi S, Kumar S, Gupta KK. Diabetes mellitus: Trends in northern India. *Indian J Endocr Metab* 2014;18:731-4/.

23 Kalra S, Kalra B, Kumar A. Social stigma and discrimination: A care crisis for young women with diabetes in India. *Diabetes Voice.* 2009; 54:37–9.

24 Распространенность диабета 2018. <https://www.diabetes.org.uk/professionals/position-statements-reports/statistics/diabetes-prevalence-2018>. 01.02.2019.

25 Диабет в Великобритании: факты и статистика. <https://mrc.ukri.org/documents/pdf/diabetes-uk-facts-and-stats-june-2015/>. 01.02.2019.

26 Диабет в Великобритании. Знать диабет. Бороться с диабетом. https://www.diabetes.org.uk/resources-s3/2019-02/1362B_Facts%20and%20stats%20Update%20Jan%202019_LOW%20RES_EXTERNAL.pdf. 01.02.2019.

27 Диабет: факты и статистика. version 3. revised: march 2014 next review: march 2015. <https://www.diabetes.org.uk/resources-s3/2017-11/diabetes-key-stats-guidelines-april2014.pdf>. 01.02.2019.

28 Sanjay Basu, David Stuckler, Martin McKee, Gauden Galea. Nutritional determinants of worldwide diabetes: an econometric study of food markets and diabetes prevalence in 173 countries. *Public Health Nutrition:* 16(1), 179–186. doi:10.1017/S1368980012002881.

29 ВОЗ - Информация о странах с диабетом, 2016. https://www.who.int/diabetes/country-profiles/kaz_ru.pdf?ua=1. 01.02.2019.

30 Туякбаева А.С. Состояние проблемы и пути профилактики сахарного диабета //Центрально-Азиатский журнал по общественному здравоохранению. – Том 11. – №2 – Алматы. – 2012.-С.8.

31 Центрально-Азиатский диабетологический форум 2015 года //Здоровье Казахстана медицинская газета. - №3(34) – Алматы. – 2015. – С.55/

32 Даньярова Л.Б., Джунусбекова Г.А., Тундыбаева М.К., Ташманова А.Б., Шыман Ж.Ж. Актуальность проведения эпидемиологических исследований по распространенности сахарного диабета 2 типа и промежуточных гипергликемий в Казахстане //Медицина, №9 (171), Алматы. – 2016. С.52-55.

33 Казахстанское общество по изучению диабета. <https://www.kssd.site/>. 01.02.2020.

34 Приказ Министра здравоохранения Республики Казахстан от 16 марта 2011 года № 145. О внесении изменений в приказ и.о. Министра здравоохранения Республики Казахстан от 10 ноября 2009 года № 685 "Об утверждении Правил проведения профилактических медицинских осмотров целевых групп населения".

35 Приказ Министра здравоохранения Республики Казахстан от 5 апреля 2013 года № 211 «О внедрении Программы управления хроническими неинфекционными заболеваниями в пилотных регионах».

36 Приказ Министра здравоохранения и социального развития Республики Казахстан от 15 марта 2016 года № 203 «О некоторых вопросах управления хроническими неинфекционными заболеваниями в рамках пилотного проекта».

37 Приказ Министра здравоохранения Республики Казахстан от 25 мая 2012 года №364 «Об утверждении Положения о деятельности организаций здравоохранения, оказывающих эндокринологическую помощь населению Республики Казахстан».

38 Приказ и.о. Министра здравоохранения Республики Казахстан от 4 ноября 2011 года № 786 «Об утверждении перечня лекарственных средств и изделий медицинского назначения для бесплатного обеспечения населения в рамках государственного обеспечения бесплатной медицинской помощью на амбулаторном уровне».

39 Приказ Министра здравоохранения Республики Казахстан от 27 апреля 2019 года № ҚР ДСМ-55 «О внесении изменений и дополнений в приказ Министра здравоохранения Республики Казахстан от 18 июля 2018 года № 434 «Об утверждении списка лекарственных средств, изделий медицинского назначения в рамках гарантированного объема бесплатной медицинской помощи и в системе обязательного социального медицинского страхования, закупаемых у Единого дистрибьютора на 2019 год».

40 Статистический сборник «Здоровье населения Республики Казахстан и деятельность организаций здравоохранения в 2015 году» Статистический сборник.-Астана. 2016. <http://www.rcrz.kz/index.php/ru/statistika-zdravookhraneniya-2>.

41 Токтарова Н.Н., Базарбекова Р.Б., Досанова А. К. Распространенность сахарного диабета 2 типа среди взрослого населения Казахстана (результаты регистрового национального исследования NOMAD). Медицина. - Алматы, №6 (180), 2017.-с.49.

42 Новый Казахстан в новом мире. Послание Президента Республики Казахстан народу Казахстана, г. Астана, 28 февраля 2007 года // Казахстанская правда. – 2007. – 1 марта. – № 33 (25278).

43 Салимгерей А. А., Жандосова Ж. Ж., Маулен А. А., Социально-значимые болезни, влияющие на здоровье молодёжи Республики Казахстан (на основе официальных статистических и социологических исследований). Вестник КазНУ. Серия юридическая. №1 (77). 2016, с.72. ISSN 1563-0366.

44 Нажмутдинова Д.К., Урунбаева Д.А., Садыкова Н.Г., Рахимбердиева З.А., Эргашова Ш.Э. Оценка качества жизни больных сахарным диабетом 2 типа, осложненным диабетической полинейропатией. Медицина.-Алматы, №3 (177), 2017.-с.50.

45 Государственные услуги и информация онлайн. Гарантированный объем бесплатной медицинской помощи. Кто и как может получить гарантированный объем бесплатной медицинской помощи? http://egov.kz/cms/ru/articles/health_care/2Fgarant_obyiom_med_pomoshi. 01.02.2020.

46 Всемирный день борьбы с диабетом в Казахстане. Статистика и планы. <https://www.zakon.kz/4945901-vsemirnyy-den-borby-s-diabetom-v.html>. 01.02.2020.

47 Стратегия «Казахстан 2050»:новый политический курс состоявшегося государства. Послание Президента Республики Казахстан – Лидера Нации Н. А. Назарбаева народу Казахстана.–Астана, 14 декабря 2012 г.

48 Lau F., Kuziemyky S., Price M., Gardner J. A review on systematic reviews of health information system studies. J Am Med Inform Assoc. 2010 Nov-Dec; 17(6): 637–645. doi: 10.1136/jamia.2010.004838.

49 А вы за нас еще и лечить будете? Роботы в белых халатах. <https://www.forbes.ru/tehnologii/338785-vy-za-nas-eshche-i-lechit-budete-roboty-v-belyh-halatah>. 03.08.2019.

50 Bassem S. Abu-Nasser. Medical Expert Systems Survey. International Journal of Engineering and Information Systems, 2017, 1 (7), pp.218-224. fihal-01610722f.

51 Gunčar G., Kukar M., Notar M., Brvar M., Černelč P., Notar M., Notar M. An application of machine learning to haematological diagnosis. Scientific Reports volume 8, Article number: 411 (2018).

52 Abu Naser, S. S., & Al-Nakhal, M. A. (2016). A Ruled Based System for Ear Problem Diagnosis and Treatment. World Wide Journal of Multidisciplinary Research and Development, 2(4), 25-31.

53 Gudu, J., Gichoya, D., Nyongesa, P., & Muumbo, A. (2012). Development of a Medical Expert System as an Expert Knowledge Sharing Tool on Diagnosis and Treatment of Hypertension in Pregnancy. International Journal of Bioscience, Biochemistry and Bioinformatics, Vol. 2, No. 5.

54 Abu Naser, S. S., & Akkila, A. N. (2008). A Proposed Expert System for Skin Diseases Diagnosis. Journal of Applied Sciences Research; www.aensiweb.com/JASR/, 4(12), 1682-1693.

55 Singla J., Grover D., Bhandari M. Medical Expert Systems for Diagnosis of Various Diseases. International Journal of Computer Applications (0975 – 8887) Volume 93 – No.7, May 2014.

56 Nohria R., Medical Expert System-A Comprehensive Review. International Journal of Computer Applications (0975 – 8887) Volume 130 – No.7, November 2015.

57 Asabere, N. mMES: A Mobile Medical Expert System for Health Institutions in Ghana. International Journal of Science and Technology. Volume 2 No.6, June 2012. ISSN 2224-3577.

58 El-Mashharawi H., Alshawwa I., Elkahlout M., Abu-Naser S. An expert system for arthritis disease using SL5 Object. International Journal of Academic Health and Medical Research (IJAHMR) ISSN: 2000-007X Vol. 3 Issue 4, April-2019, Pages: 28-35.

59 Qurat-ul-ain Mastoi, Teh Ying Wah, Ram Gopal Raj, and Uzair Iqbal. Automated Diagnosis of Coronary Artery Disease: A Review and Workflow. Cardiology Research and Practice. Volume 2018, Article ID 2016282, 9 pages. doi.org/10.1155/2018/2016282.

60 Диагностические технологии и техники. <https://www.nyuwinthrop.org/diagnostic-technology-techniques>. 01.02.2020

61 «Современные технологии контроля диабета дают новый уровень свободы», Медицинские технологии, выпуск №2, 24 октября 2018. <https://plus.rbc.ru/news/5bcf76cb7a8aa9437e5daaed>. 01.02.2020.

62 Послание Президента Республики Казахстан Н. Назарбаева народу Казахстана. «Третья модернизация Казахстана: глобальная конкурентоспособность». - Астана 31 января 2017 г. http://www.akorda.kz/ru/addresses/addresses_of_president/poslanie-prezidenta-respubliki-kazahstan-nazarbaeva-narodu-kazahstana-31-yanvarya-2017-g. 01.02.2020.

63 Киштибаева А.Е., Шопабаева А.Р., Пичхадзе Г.М. Маркетинговое исследование распространения сахарного диабета в Республике Казахстан. // Вестник КазНМУ, №2(2), 2014. – С.390-393.

64 European Foundation for the Study of Diabetes. <http://www.europeandiabetesfoundation.org/>. 01.02.2020.

65 Стипендиальные программы взаимных исследований. <http://www.europeandiabetesfoundation.org/workshops/59-european-foundation-study-diabetes-efsd-and-japan-diabetes-society-jds-reciprocal>. 01.02.2020.

66 Что может сделать молодежная акция против диабета? http://www.yda.org.hk/?page_id=2. 01.02.2020.

67 Международная Диабетическая Федерация. Западная часть Тихого океана. Новости. International Diabetes Federation. IDF. Western Pacific. News.

<https://www.idf.org/our-network/regions-members/western-pacific/western-pacific/118-hiking-in-hong-kong-for-world-diabetes-day.html>. 01.02.2020.

68 Kuchipudi Sravanthi, Tatireddy Subba Reddy, Applications of Big data in Various Fields. Kuchipudi Sravanthi et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (5) , 2015, 4629-4632.

69 Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, Vishanth Weerakkody. Critical analysis of Big Data challenges and analytical methods. Journal of Business Research. Volume 70, January 2017, Pages 263-286.

70 Mashooque A. Memon, Safeullah Soomro, Awais K. Jumani, Muneer A. Kartio. Big Data Analytics and Its Applications. Annals of Emerging Technologies in Computing (AETiC) Vol. 1, No. 1, 2017.

71 Learn Data Science. 4 types of Data Analytics. <https://www.datascience.com/blog/4-types-of-data-analytics>.

72 Пять ключевых типов аналитики больших данных, которые должен знать каждый бизнес-аналитик. <https://www.carptechu.edu/blog/five-types-of-big-data-business-analytics>. 01.02.2020.

73 Диагностическая аналитика. <https://www.cornerstoneondemand.com/glossary/diagnostic-analytics>. 01.02.2020.

74 5 видов аналитики: предписывающая, прогнозирующая, диагностическая, описательная и когнитивная аналитика. <https://www.weirdgeek.com/2018/11/types-of-analytics/>. 01.02.2020.

75 Типы аналитики больших данных. <https://www.rishabhsoft.com/blog/what-is-big-data-analytics-and-types>. 01.02.2020.

76 Приложения для работы с большими данными в реальном времени в различных областях. <https://www.edureka.co/blog/big-data-applications-revolutionizing-various-domains/>. 01.02.2020.

77 Ashwin Belle, Raghuram Thiagarajan, S. M. Reza Soroushmehr, Fatemeh Navidi, Daniel A. Beard, Kayvan Najarian. Big Data Analytics in Healthcare. Biomed Res Int. 2015; 2015: 370194. doi: 10.1155/2015/370194.

78 Силен Д., Мейсман А., Али М. Основы Data Science и BigData. Python и наука о данных. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия программиста). ISBN 978-5-496-02517-1.

79 Kong HJ. Managing Unstructured Big Data in Healthcare System. Health Inform Res. 2019;25(1):1–2. doi:10.4258/hir.2019.25.1.1.

80 Pak HS. Unstructured data in healthcare [Internet] Fremont (CA): Healthcare Tech Outlook; c2018.

81 Big Data в здравоохранении: как и для чего медицина использует Большие Данные? <http://ru.datasides.com/big-data-medicine/>.

82 Электронная аналитика данных медицинской карты помогает выявить риск, последствия сепсиса. <https://ehrintelligence.com/news/ehr-data-analytics-help-detect-risk-outcomes-of-sepsis>. 01.02.2020.

83 Присоединяйтесь к исследованию, чтобы положить конец болезни сердца. <https://www.health-eheartstudy.org/>. 01.02.2020.

84 Johns Hopkins Medicine. www.hopkinsmedicine.org. 01.02.2020.

85 Исследования прозрачности рынка. <https://www.transparencymarketresearch.com/>. 01.02.2020.

86 Медицинские технологии. Тематическое приложение к ежедневной деловой газете РБК, 24 октября 2018 | № 185 (2909).

87 N. Saparkhojayev, A. Mukasheva, P. Saparkhojayev. The concept of monetization of IoT-based project: case of Medical System in Kazakhstan. The 15th International Scientific Conference Information Technologies and Management 2017 April 27 – 28, pp. 78-81, ISMA University, Riga, Latvia.

88 А.К. Мукашева, Н.П. Сапарходжаев. «Разработка системы обслуживания пациентов на основе технологии BigData на территории Республики Казахстан». МНПК «Математические методы и информационные технологии макроэкономического анализа и экономической политики» КазННТУ им. К.И.Сатпаева, стр. 182-187, 11-12 апреля 2017г. – с.182-187

89 Н.П. Сапарходжаев., А.К. Мукашева, «Анализ системы для диагностики сахарного диабета на основе технологии BigData» ISBN 978-601-323-111-2. Труды Международных Сатпаевских чтений «Инновационные решения традиционных проблем: инженерия и технологии», 12 апреля 2018г. И66 – Алматы: КазННТУ имени Сатпаева, 2018.-1254-1256.

90 N. Saparkhojayev, A.K. Mukasheva, «Introduction to BigData technology for diagnosis of diabetes». Information Technologies and Management, 2018 April 26 – 27, “Information Technologies, Management and Society”, ISMA University, Riga, Latvia.

91 С.И. Кифоренко, А.Б. Котова, Н.В. Лавренюк, Е.Ю. Иваськива, «Диагностика сахарного диабета. Прогрессивные информационные технологии» УСиМ, 2015, № 4.

92 Алгоритмы специализированной медицинской помощи больным сахарным диабетом/ Под редакцией И.И. Дедова, М.В. Шестаковой, А.Ю. Майорова. – 8-й выпуск. – М.: УП ПРИНТ; 2017.

93 Ministry of National Economy of the Republic of Kazakhstan. Committee on statistics. Official website: http://stat.gov.kz/faces/homePage?_adf.ctrl-state=2fn371p1u_4&lang=ru&_afLoop=8562077751869222. 01.02.2020.

94 Werner Vach: Regression Models as a Tool in Medical Research, 2013 by Taylor & Francis Group, LLC, ISBN-13: 978-1-4665-1749-3.

95 M. A. Golberg, H. A. Cho, Introduction to Regression Analysis, WIT Press, ISBN-13: 978-1853126246

96 Sineeveva L.V., Medvedeva N.N., Nikolaev V.G., Strelkovich N.N., Orlova I.I. Application of regression analysis methods in the biomedical researches. Bulletin of new medical technologies - 2013 - Т. XX, № 2 - p. 217

97 Per Kragh Andersen, Lene Theil Skovgaard. Regression with Linear Predictors. Springer New York Dordrecht Heidelberg London 2010, ISBN 978-1-4419-7169-2, DOI 10.1007/978-1-4419-7170-8.

98 Cherkashina Y.A., Application of regression analysis for solving diagnosis problem of children's health. Modern problems of science and education. 2015. № 1 (part 1).

99 Xin Yan, Xiaogang Su. Linear regression analysis. Theory and computing. June 2009. ISBN-13: 978-981-283-410-2. DOI 10.1142/6986.

100 Ton J. Cleophas., Aeilko H. Zwinderman. Regression Analysis in Medical Research for Starters and 2nd Levelers. Springer International Publishing AG 2018. ISBN 978-3-319-71936-8. DOI.org/10.1007/978-3-319-71937-5.

101 3 ways to calculate a polynomial in Excel. <https://4analytics.ru/trendi/3-sposoba-rascheta-polinoma-v-excel.html>. 01.02.2020.

102 Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining. Introduction to Linear Regression Analysis, 5th Edition. ISBN: 978-0-470-54281-1. April 2012. 672 Pages.

103 Mukasheva, A., Saparkhojayev, N., Akanov, Z. et al. Forecasting the Prevalence of Diabetes Mellitus Using Econometric Models. Diabetes Therapy (2019). <https://doi.org/10.1007/s13300-019-00684-1>.

104 А.К. Мукашева, Н.П. Сапарходжаев. «Прогнозирование распространенности диабета в республике Казахстан на основе методов регрессионного анализа». Вестник КазНУ, №5(135)2019г. – Алматы: КазНУ имени Сатпаева, 2019, стр. 290-297.

105 Seuring T, Archangelidi O, Suhrcke M. The Economic Costs of Type 2 Diabetes: A Global Systematic Review. Pharmacoeconomics. 2015;33(8):811–831. doi:10.1007/s40273-015-0268-9.

106 Christian Bommer, Vera Sagalova, Esther Heesemann, Jennifer Manne-Goehler, Rifat Atun, Till Bärnighausen, Justine Davies, Sebastian Vollmer. Diabetes Care May 2018, 41 (5) 963-970; DOI: 10.2337/dc17-1962.

107 Математические модели динамики экономических систем: монография / И.В. Трегуб. — Москва : РУСАЙНС, 2018. — 164 с.

108 Шанченко, Н. И., Эконометрика: лабораторный практикум/ Н. И. Шанченко – Ульяновск: УлГТУ, 2004. – 79 с.

109 Сергеева Ю.В., Эконометрика: учебное пособие. - Дзержинск: изд-во «Конкорд», 2015. – 126 с.

110 Каморников С. Ф. , Каморников С. С. Эконометрика : учеб. пособие. – М. : Интеграция, 2012. – 262 с.

111 Домбровский В.В. Эконометрика: учебник/ В.В. Домбровский. – М.: Нов. учебник, 2004. – 342 с. – ISBN 5-8393-0400-X.

112 Эконометрика: учебное пособие / И. В. Потахова. — Томск: факультет дистанционного обучения ТУСУРа, 2015. — 110 с.

113 Эконометрика / И. И. Исмагилов, Е. И. Кадочникова, А. В. Костромин. – Казань: Казан. ун-т, 2014. – 235 с.

- 114 Эконометрика. Конспект лекций / Е. И. Кадочникова; Каз. федер. ун-т. – Казань, 2013. – 106 с.
- 115 Эконометрика: Учеб. пособие / А.И. Новиков. - 2-е изд., испр. и доп. - М.: ИНФРА-М, 2011. - 144 с.
- 116 Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12 (2011) 2825-2830.
- 117 Н.П. Сапарходжаев, А.К. Мукашева. «Прогнозирование распространенности диабета в Республике Казахстан на основе методов регрессионного анализа». Вестник АУЭС, №3(46)2019, Алматы, стр. 100-107.
- 118 M. Minelli, M. Chambers and A. Dhiraj, Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses, Chinese edn. (Wiley & Sons, 2013).
- 119 Sun, Zhaohao & Wang, Paul. (2017). A Mathematical Foundation of Big Data. Journal of New Mathematics and Natural Computation. 13. 83-99. 10.1142/S1793005717400014.
- 120 Z. Sun and J. Xiao, Essentials of Discrete Mathematics, Problems and Solutions (Hebei University Press, Baoding, 1994).
- 121 C. Coronel, S. Morris and P. Rob, Database Systems: Design, Implementation, and Management, 11th edn. (Course Technology, Cengage Learning, Boston, 2015).
- 122 S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd edn. (Prentice Hall, Upper Saddle River, 2010).
- 123 T. Jech, Set Theory, The Third Millennium edn., Revised and Expanded (Springer, 2003), https://en.wikipedia.org/wiki/Cardinality_of_the_continuum.
- 124 Клинический протокол диагностики и лечения сахарный диабет. Утверждено на Экспертной комиссии по вопросам развития здравоохранения Министерства здравоохранения Республики Казахстан протокол № 10 от «04» июля 2014 года.
- 125 Apache Hadoop. <https://hadoop.apache.org/>. 03.06.2020.
- 126 Hadoop. [https:// searchdatamanagement.techtarget.com / definition /Hadoop](https://searchdatamanagement.techtarget.com/definition/Hadoop). 03.06.2020.
- 127 База данных для современных приложений. <https://www.mongodb.com/>.03.06.2020.
- 128 T. White, Hadoop: The Definitive Guide, Oreilly & Associates Inc; 3rd Edition, 2012.
- 129 Ramsingh J., Bhuvanewari V. An efficient Map Reduce-Based Hybrid NBC-TFIDF algorithm to mine the public sentiment on diabetes mellitus – A big data approach // Journal of King Saud University. 3 July 2018.
- 130 T. White, Hadoop: The Definitive Guide, Oreilly & Associates Inc; 4rd Edition, 2015.
- 131 Национальная библиотека им. Н. Э. Баумана. Apache Hadoop. https://ru.bmstu.wiki/Apache_Hadoop. 03.06.2020.

- 132 Spark. Spark core programming. Tutorials Point (I) Pvt. Ltd. 2015.
- 133 Learning Spark: Lightning-Fast Big Data Analysis. 1st Edition by Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia. ISBN-13: 978-1449358624.
- 134 Start with more than a blinking cursor. <https://www.kaggle.com/>. 03.06.2020.
- 135 Why Choose NoSQLBooster for MongoDB? <https://nosqlbooster.com/>. 03.06.2020.
- 136 N. Saparkhojayev, A.K.Mukasheva, «The development of information system of formation and use of information resources for evaluation of parameters and evaluation of recommendations based on BigData technology tools: work with MongoDB». International Conference on Cyber Security and Computer Science (ICONCS'18), Oct 18-20, 2018 Safranbolu, Turkey.
- 137 A. Mukasheva, N. Saparkhojayev, Z. Akanov, A. Algazieva, «Prevalence of diabetes in the republic of Kazakhstan based on regression analysis methods». ISBER. International Conference on Research in E-Learning & Distance Education, Social Sciences, Economics and Management. July 13-14, 2019 / Turkey.
- 138 А. К. Мукашева, Н. П. Сапарходжаев, И. А. Зимин, «Разработка информационной системы на основе технологии BigData», «Научное творчество: научно-исследовательская практика и приоритетные направления» / Материалы XXII республиканской научной конференции студентов и молодых ученых, посвященной 1150-летию Абу Насыр аль-Фараби: Том I. (23 апреля 2020 г), стр. 527-532, ISBN 978-601-285-096-3.
- 139 А. А. Куандыков, А. К. Мукашева, И. А. Зимин, «Обработка журналов веб серверов для получения статистики сайтов с помощью инструмента Apache Spark», МНПК «Модели инновационных решений повышения конкурентоспособности отечественной науки» (4 июня 2020 г, г. Челябинск). В 2 ч. Ч. 2 / - Уфа: OMEGA SCIENCE, 2020. – стр. 79-84, ISBN 978-5-907347-21-2.

ПРИЛОЖЕНИЕ А

Разработка поисковой системы для диагностирования болезни

//Листинг программы индексирования и загрузки болезней в Elasticsearch.

```
import wikipedia
from elasticsearch import Elasticsearch
import re
client = Elasticsearch()
indexName = "medical"
diseaseMapping = {
    'properties':{
        'name':{'type':'string'},
        'title':{'type':'string'},
        'fulltitle':{'type':'string'}
    }
}
dl = wikipedia.page("List of diseases")
print(dl.links)
diseaseListArray = []
check = re.compile("List of diseases*")

for link in dl.links:
    if check.match(link):
        try:
            diseaseListArray.append(wikipedia.page(link))
        except Exception,e:
            print str(e)
print(diseaseListArray[0].links)
checkList = [ ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"],
               ["A", ["B"], ["C"], ["D"], ["E"], ["F"], ["G"], ["H"],
               ["I"], ["J"], ["K"], ["L"], ["M"], ["N"], ["O"], ["P"],
               ["Q"], ["R"], ["S"], ["T"], ["U"], ["V"], ["W"], ["X"], ["Y"], ["Z"]]
docType = 'diseases'
for diseaselistNumber, diseaselist in enumerate(diseaseListArray):
    for disease in diseaselist.links:
        try:
            if disease[0] in checkList[diseaselistNumber] and disease[0:3] != "List":
                currentPage = wikipedia.page(disease)
                client.index(index=indexName, doc_type=docType, id=disease, body = {
                    "name":disease, "title":currentPage.title, "fulltext":currentPage.content})
            except Exception,e:
                print str(e)
```

//Листинг программы диагностики болезни по симптомам.

```
from elasticsearch import Elasticsearch
client = Elasticsearch()
indexName = "medical"
docType = "diseases"
searchFrom = 0
searchSize = 3
searchBody = {
```

```

"fields": ["name"],
"query": {
  "simple_query_string": {
    "query": '+fatigue+fever+"joint pain"+rash+"chest pain"',
    "fields": ["fulltext", "title^5", "name^10"]
  }
}
}
print(client.search(index=indexName, doc_type=docType,
  body=searchBody, from_ = searchFrom, size=searchSize))

```

//Результатом обработки листинга является диагностика трех видов волчанки.

```

{u'hits': {u'hits': [{u'_score': 0.010658317, u'_type': u'diseases', u'_id': u'Cutaneous lupus erythematosus', u'fields': {u'name': [u'Cutaneous lupus erythematosus']}}, u' index': u'medical'}, {u'_score': 0.010658317, u'_type': u'diseases', u'_id': u'Lupus erythematosus', u'fields': {u'name': [u'Lupus erythematosus]}}, u' index': u'medical'}, {u'_score': 0.0056674588, u'_type': u'diseases', u'_id': u'Systemic lupus erythematosus', u'fields': {u'name': [u'Systemic lupus erythematosus']}}, u' index': u'medical'}], u'total': 6, u'max score': 0.010658317, u' shards': {u'successful': 5, u'failed': 0, u'total': 5}, u'took': 247, u'timed_out': False}

```

//Листинг программы вывода совпадений по запросу с фильтром «diabetes» в названии болезни.

```

from elasticsearch import Elasticsearch
client = Elasticsearch()
indexName = "medical"

docType = "diseases"
searchFrom = 0
searchSize = 3

searchBody = {
  "fields": ["name"],
  "query": {
    "filtered": {
      "filter": {
        'term': {'name': 'diabetes'}
      }
    }
  },
  "aggregations": {
    "DiseaseKeywords": {
      "significant_terms": { "field": "fulltext", "size": 30
    }
  }
}
}
client.search(index=indexName, doc_type=docType,
  body=searchBody, from_ = searchFrom, size=searchSize)

```

//Результатом обработки листинга является количество возвращённых строк, которые равняются 28 значениям.

Out[2]:

```
{u'_shards': {u'failed': 0, u'successful': 5, u'total': 5},
 u'aggregations': {u'DiseaseKeywords': {u'buckets': [{u'bg_count': 9,
  u'doc_count': 9,
  u'key': u'bainein',
  u'score': 55.917091836734706},
 {u'bg_count': 9,
  u'doc_count': 9,
  u'key': u'passer',
  u'score': 55.917091836734706},
 {u'bg_count': 9,
  u'doc_count': 9,
  u'key': u'diabainein',
  u'score': 55.917091836734706},
 {u'bg_count': 10,
  u'doc_count': 9,
  u'key': u'siphon',
  u'score': 50.29323979591837},
 {u'bg_count': 7,
  u'doc_count': 7,
  u'key': u'sapidus',
  u'score': 43.49107142857143},
 {u'bg_count': 7,
  u'doc_count': 7,
  u'key': u'ndi',
  u'score': 43.49107142857143},
 {u'bg_count': 7,
  u'doc_count': 7,
  u'key': u'sapere',
  u'score': 43.49107142857143},
 {u'bg_count': 7,
  u'doc_count': 7,
  u'key': u'tasty',
  u'score': 43.49107142857143},
 {u'bg_count': 10,
  u'doc_count': 8,
  u'key': u'avp',
  u'score': 39.706122448979585},
 {u'bg_count': 6,
  u'doc_count': 6,
  u'key': u'insipidus',
  u'score': 37.27806122448979},
 {u'bg_count': 6,
  u'doc_count': 6,
  u'key': u'rehydrate',
  u'score': 37.27806122448979},
 {u'bg_count': 6,
  u'doc_count': 6,
  u'key': u'vaptanes',
  u'score': 37.27806122448979},
```


{u'bg_count': 6,
u'doc_count': 6,
u'key': u'defect.the',
u'score': 37.27806122448979},
{u'bg_count': 6,
u'doc_count': 6,
u'key': u'l.l',
u'score': 37.27806122448979},
{u'bg_count': 6,
u'doc_count': 6,
u'key': u'insipide',
u'score': 37.27806122448979},
{u'bg_count': 9,
u'doc_count': 7,
u'key': u'tasteless',
u'score': 33.770833333333336},
{u'bg_count': 7,
u'doc_count': 6,
u'key': u'orlistat',
u'score': 31.922011661807577},
{u'bg_count': 7,
u'doc_count': 6,
u'key': u'natriuresis',
u'score': 31.922011661807577},
{u'bg_count': 7,
u'doc_count': 6,
u'key': u'casr',
u'score': 31.922011661807577},
{u'bg_count': 7,
u'doc_count': 6,
u'key': u'gk',
u'score': 31.922011661807577},
{u'bg_count': 10,
u'doc_count': 7,
u'key': u'osmolarity',
u'score': 30.36875},
{u'bg_count': 53,
u'doc_count': 16,
u'key': u'polyuria',
u'score': 29.611089718906427},
{u'bg_count': 37,
u'doc_count': 13,
u'key': u'thirst',
u'score': 28.077185603971323},
{u'bg_count': 8,
u'doc_count': 6,
u'key': u'cidofovir',
u'score': 27.904974489795915},
{u'bg_count': 8,
u'doc_count': 6,
u'key': u'perceptible',
u'score': 27.904974489795915},

```

    {u'bg_count': 11,
      u'doc_count': 7,
      u'key': u'hydrochlorothiazide',
      u'score': 27.585227272727273},
    {u'bg_count': 12,
      u'doc_count': 7,
      u'key': u'thiazides',
      u'score': 25.265625},
    {u'bg_count': 12, u'doc_count': 7, u'key': u'flavor', u'score': 25.265625},
    {u'bg_count': 4,
      u'doc_count': 4,
      u'key': u'lada',
      u'score': 24.852040816326525},
    {u'bg_count': 4,
      u'doc_count': 4,
      u'key': u'idf',
      u'score': 24.852040816326525}],
  u'doc_count': 28}},
  u'hits': {u'hits': [{u'_id': u'Diabetes mellitus',
    u'_index': u'medical',
    u'_score': 1.0,
    u'_type': u'diseases',
    u'fields': {u'name': [u'Diabetes mellitus']}}],
    {u'_id': u'Diabetes insipidus, nephrogenic type 3',
      u'_index': u'medical',
      u'_score': 1.0,
      u'_type': u'diseases',
      u'fields': {u'name': [u'Diabetes insipidus, nephrogenic type 3']}}],
    {u'_id': u'Maturity onset diabetes of the young',
      u'_index': u'medical',
      u'_score': 1.0,
      u'_type': u'diseases',
      u'fields': {u'name': [u'Maturity onset diabetes of the young']}}],
  u'max_score': 1.0,
  u'total': 28},
  u'timed_out': False,
  u'took': 1959}

```

ПРИЛОЖЕНИЕ Б

Разработка интерфейса ИС

Для построения сервера был использован Node JS. JavaScript построен как цикл обработки событий, а Node.js построен из JavaScript. Node.js способен обрабатывать несколько запросов, и действовать будет как клиент в отношении сторонних сервисов, выполняя только один поток. JavaScript выполняет действие на стороне клиента, а Node — на сервере. С помощью Node.js можно писать полноценные приложения. Node умеет работать с внешними библиотеками, вызывать команды из кода на JavaScript и выполнять роль веб-сервера. С Node.js проще масштабироваться. При одновременном подключении к серверу тысяч пользователей Node.js работает асинхронно, то есть ставит приоритеты и распределяет ресурсы грамотнее. Node.js был выбран так как, нам необходимо, чтобы пользователи могли без затруднений заходить в систему и загружать свои данные. При построения сервера на Node.js мы можем использовать готовые библиотеки. К примеру, express необходим для прослушивания сервера, jsonwebtoken необходим для создания токена (см. рис. Б1).

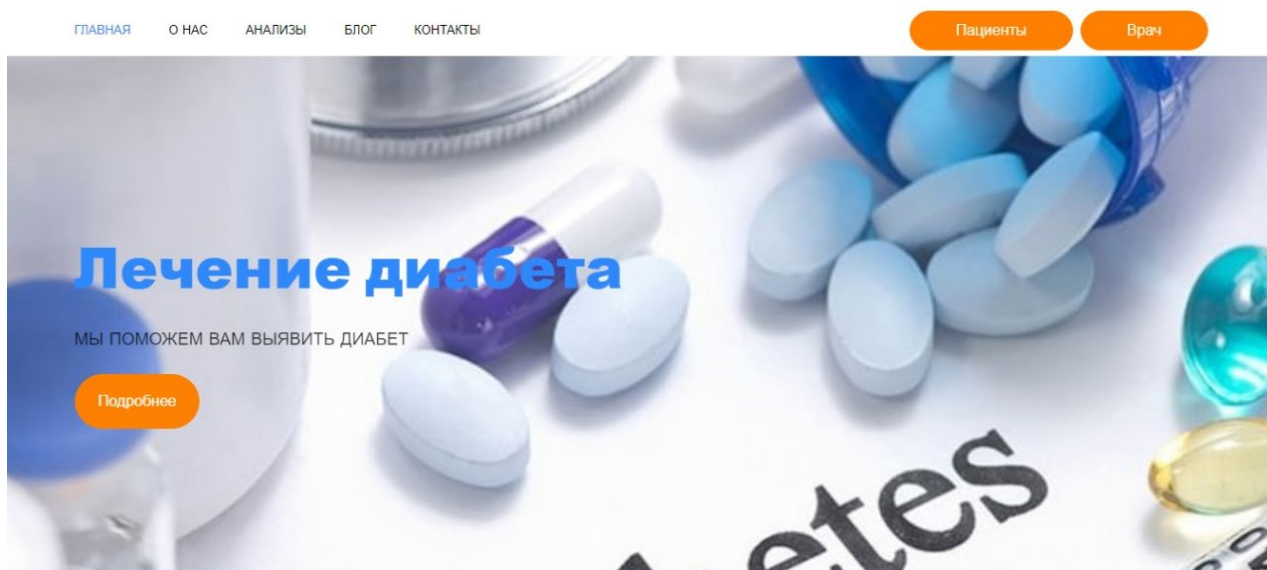


Рисунок Б1. Главное меню системы

Для начала вводятся регистрационные данные пациента (см. рис. Б2).

КОНСУЛЬТАЦИЯ

Регистрация

Имя	Фамилия
Отчество	Дата рождения <input type="checkbox"/>
Email	Пароль
Лечащий доктор	Адрес проживания
Пройти регистрацию	У Вас уже есть аккаунт?

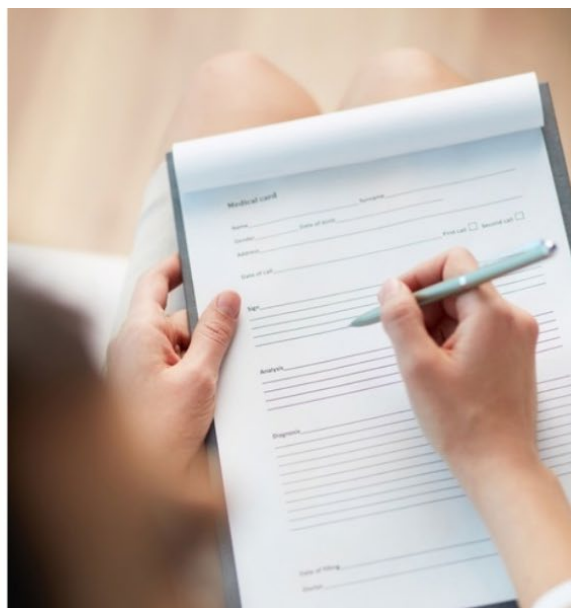


Рисунок Б2. Регистрация пациента

После процедуры регистрации необходимо завести логин и пароль для авторизации (см. рис. Б3).

Авторизация пользователей

ГЛАВНАЯ > АВТОРИЗАЦИЯ >

ВХОД В СИСТЕМУ

Логин	Пароль
<input type="text"/>	<input type="password"/>
Войти	

A close-up photograph showing a person's hands holding a white medical form on a clipboard. The person is using a light blue pen to write on the form. The form has various fields and checkboxes, including 'Medical card', 'Name', 'Date of birth', 'Address', 'Sex', 'Blood group', 'Allergies', and 'Date of filling'.

Рисунок Б3. Авторизация пользователя (пациента)

После регистрации и заведения аккаунта на пациента в базе создается история болезни. Для авторизации сотрудников (врача) необходимо быть зарегистрированным в отделе технического обслуживания медицинского учреждения, где разработчики заранее выдадут логин и пароль (см. рис. Б4).

Имя	Фамилия	Email	Лечащий врач
Сулейменова	Гаухар	suleimenova@mail.ru	Карасов Медет
Сергей	Иванов	ivanov@mail.ru	Карасов Медет
Абдибеков	Досан	abdibekov@mail.ru	Карасов Медет
Абдрахманова	Акерке	abrakhmanova@mail.ru	Карасов Медет
Алиярова	Сандугаш	aliyarova@mail.ru	Карасов Медет
Алмабаев	Санжар	almabaev@bk.ru	Карасов Медет
Альмуратова	Карина	almuratova@mail.ru	Карасов Медет
Асфендияр	Аяулым	asfendiyar@br.ru	Карасов Медет
Ахметова	Аксункар	ahmetova@bk.ru	Карасов Медет
Аюпова	Айсулу	auypova@bk.ru	Карасов Медет
Бергинбаев	Абзал	berginbayev@bk.ru	Карасов Медет

Рисунок Б4. Закрепление пациентов за лечащим врачом

Для того чтобы диагностировать СД у пациента необходимо направить пациента сдать анализы в клиничко-диагностической лаборатории. Далее из выпадающего меню вводятся анализы, где каждый анализ состоит из различных числовых и текстовых параметров (см. рис. Б5).

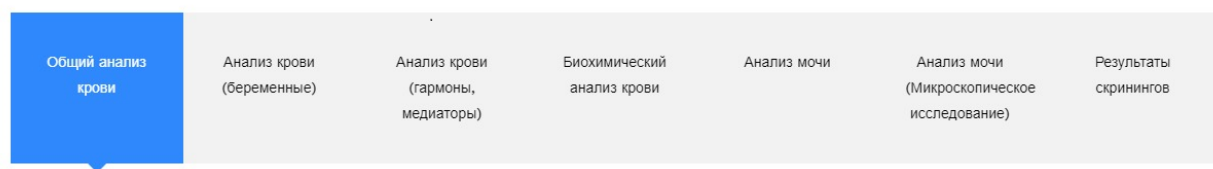


Рисунок Б5. Перечень анализов

Параметры, которые нужно ввести для общего анализа крови пациента (см. рис. Б6).

Общий анализ крови

Врачебный анализ, позволяющий оценить содержание гемоглобина в системе красной крови, количество эритроцитов, цветовой показатель, количество лейкоцитов, тромбоцитов.

Гемоглобин	Эритроциты	Цветной показатель
Средний объем эритроцита	Среднее содержание нб в :	Средняя концентрация нб в е
Распределение эритроцитов	Тромбоциты	Тромбоцит
Средний объем тромбоцита	Лейкоциты	Нейтрофилы
Нейтрофилы (абс. кол-во)	Эозинофилы	Эозинофилы (абс. кол-во)
Базофилы	Базофилы (абс. кол-во)	Моноциты
Моноциты (абс. кол-во)	Лимфоциты	Лимфоциты (абс. кол-во)
Отправить анализы		




Рисунок Б6. Общий анализ крови

Далее из выпадающего меню вводится биохимический анализ крови для беременных пациентов (см. рис. Б7).

Общий анализ крови	Анализ крови (беременные)	Анализ крови (гормоны, медиаторы)	Биохимический анализ крови	Анализ мочи
--------------------	----------------------------------	-----------------------------------	----------------------------	-------------

Анализ крови при беременности

Анализ при беременности – это необходимое условие для нормального развития, роста и рождения ребенка, а также сохранения здоровья будущей матери.

Кожа	Гликозилированный гемогло
Отправить анализы	




Рисунок Б7. Биохимический анализ крови для беременных

Далее из выпадающего меню вводится анализ крови на содержание гормонов и медиаторов (см. рис. Б8).

Анализ крови – содержание гормонов и медиаторов

С помощью анализа на гормоны можно выявить многие патологии, в их числе:

- дисфункции щитовидной железы (гипо- и гипертиреоз, зоб, тиреозит);
- бесплодие, нарушения менструального цикла;
- сахарный и несахарный диабет;
- опухолевые процессы;
- ожирение (дисфункция эндокринных клеток поджелудочной железы);
- заболевания гипоталамо-гипофизарной системы (болезнь Иценко-Кушинга, акромегалия и др.);
- надпочечниковая недостаточность.

ТТГ

Свободный Т4

Анти ТПО

Отправить анализы



Рисунок Б8. Анализ крови на содержание гормонов и медиаторов

Также вводится биохимический анализ крови (см. рис. Б9).

Биохимический анализ крови

Метод лабораторной диагностики, который позволяет оценить работу внутренних органов (печень, почки, поджелудочная железа, желчный пузырь и др.), получить информацию о метаболизме (обмен липидов, белков, углеводов), выяснить потребность в микроэлементах.

Общий белок

Мочевина

Глюкоза

АЛТ

АСТ

Билирубин (общий)

Отправить анализы

Рисунок Б9. Биохимический анализ крови

Далее из выпадающего меню вводится анализ мочи, который делится на вид исследования, т.е. физико-химические свойства (см. рис. Б10).

Анализ мочи

Одно из самых частых лабораторных исследований, которое проводится в диагностических целях. Он назначается при многих заболеваниях и включает в себя определение до 20 показателей, каждый из которых помогает в постановке правильного диагноза.

Количество	Цвет	Прозрачность
Относительная плотность	Реакция	Белок
Глюкоза	Билирубин	Уробилиноген
Кетоны	Нитриты	Лейкоциты в полоске
Эритроциты в полоске	Отправить анализы	

Рисунок Б10 Анализ мочи (физико-химические свойства)

Далее из выпадающего меню вводится анализ мочи, который делится на вид исследования, т.е. микроскопическое исследование (см. рис. Б11).

Общий анализ крови
Анализ крови (беременные)
Анализ крови (гормоны, медиаторы)
Биохимический анализ крови
Анализ мочи (Микроскопическое исследование)
Результаты скринингов

Анализ мочи

Эритроциты неизменные	Эритроциты измененные	Лейкоциты
Бактерии	Почкующие дрожжи	Эпителий плоский
Эпителий переходный	Эпителий почечный	Слизь
Цилиндры неклассифицируе	Цилиндры гиалиновые	Цилиндры лейкоцитарные
Цилиндры зернистые	Кристаллы трипельфосфата	Кристаллы оксалата кальция
Кристаллы мочевой кислоты	Отправить анализы	




Рисунок Б11. Анализ мочи (микроскопическое исследование)

После введенных числовых параметров система определяет статус на первом этапе в соответствии (см. рис. Б12).

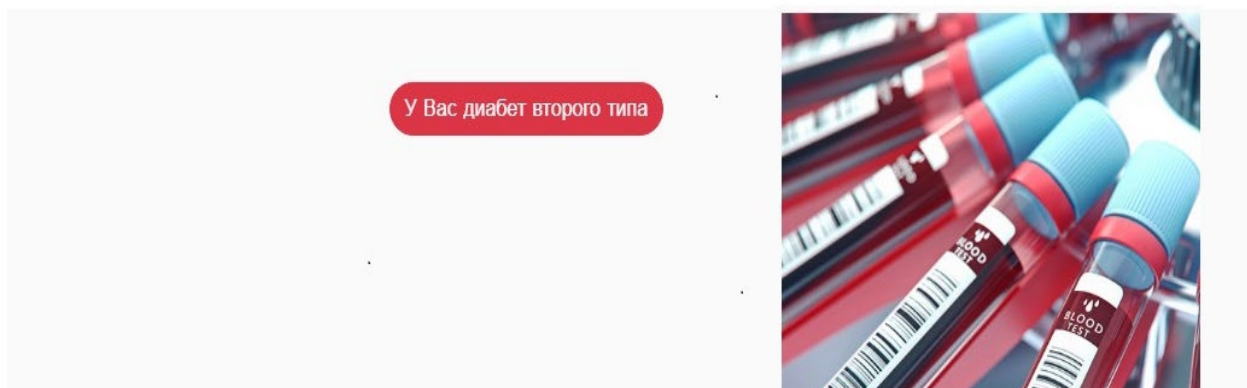


Рисунок Б12. Определение статуса больного

После введенных числовых параметров анализов если диагноз подтверждается, то для рекомендации лечения нужно сдать 12 видов анализов и обследований, которые нужно затем загрузить в историю болезни пациента (см. рис. Б13).

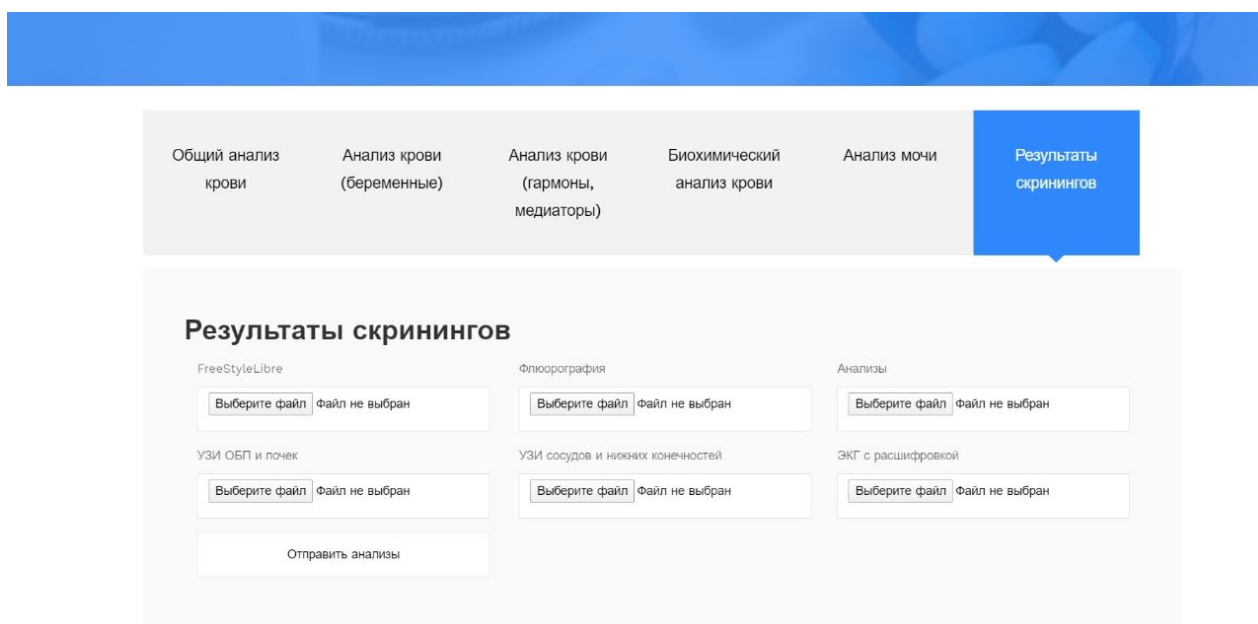


Рисунок Б13. Загрузка данных

Вводятся снимки ЭКГ, УЗИ ОБП и почек, УЗДГ сосудов нижних конечностей, флюорография, справка фтизиатра, рентгенография таких как:

- ОАК, ОАМ, микрореакция, кал на я/гл. - 10 дней;
- БАК (МТ, АСТ, общий белок, общий билирубин, мочевины, креатинин, холестерин, ЛПВП, ЛПНП, ГГТП, ЩФ, альбумин, триглицериды, калий, натрий) - 10 дней;

- гликозилированный гемоглобин – 3 месяца;
 - ЭКГ с расшифровкой – 3 недели. (При наличии подъема сегмента ST, отрицательных z. T – консультация кардиолога обязательно.);
 - УЗИ ОБП и почек – 1 месяц.
 - УЗДГ сосудов нижних конечностей – 6 месяцев.
 - консультация офтальмолога, с ОБЯЗАТЕЛЬНЫМ определением стадии ретинопатии;
 - консультация терапевта (+ справка об эпидокружении);
 - флюорография ОГК – 6 месяцев;
 - Если пациент перенес туберкулез – справка от фтизиатра – ОБЯЗАТЕЛЬНО;
 - при наличии трофических язв – рентгенография в 2х проекциях, для исключения костнодеструктивных изменений, бакпосев раневого отделяемого, консультация ангиохирурга;
 - выписка из амбулаторной карты, копия удостоверения личности.
- После проверки всех полученных анализов и обследований врач-эндокринолог официально диагностирует диабет пациенту, и принимает решение о назначении лечения и возможности госпитализации, если это будет необходимо. Справочная информация о владельце сайта (см. рис. Б14).

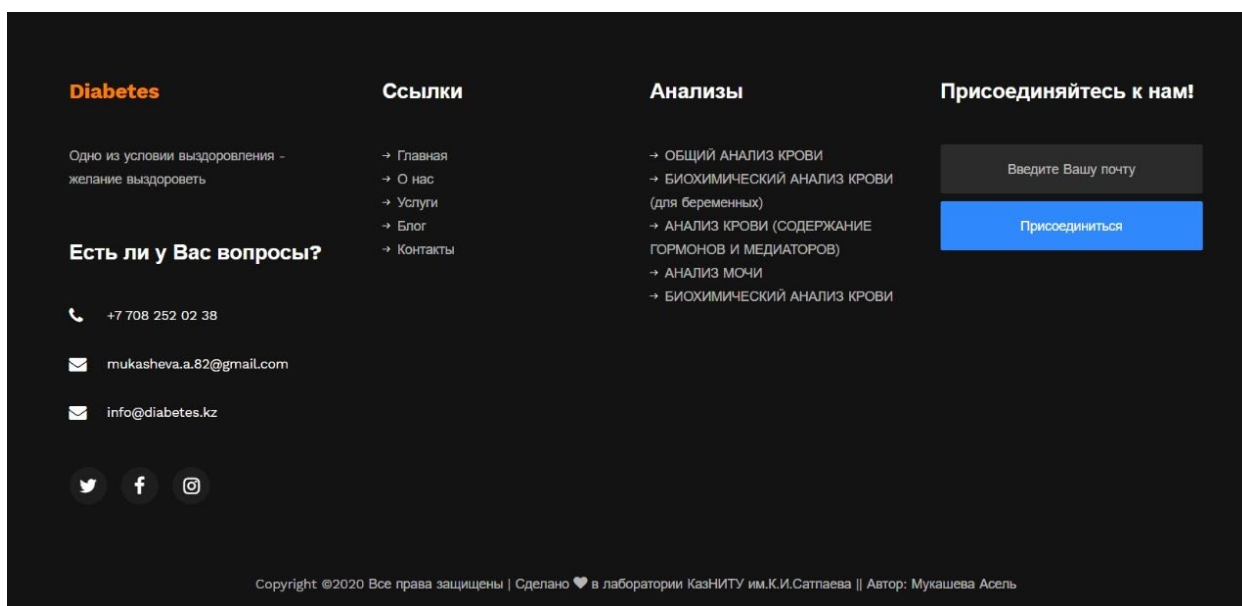


Рисунок Б14. Контактная информация

ПРИЛОЖЕНИЕ В

Процесс загружаемых данных в MongoDB

После процесса авторизации врача в главном меню системы в базе появляются загруженные данные (см. рис. В1).

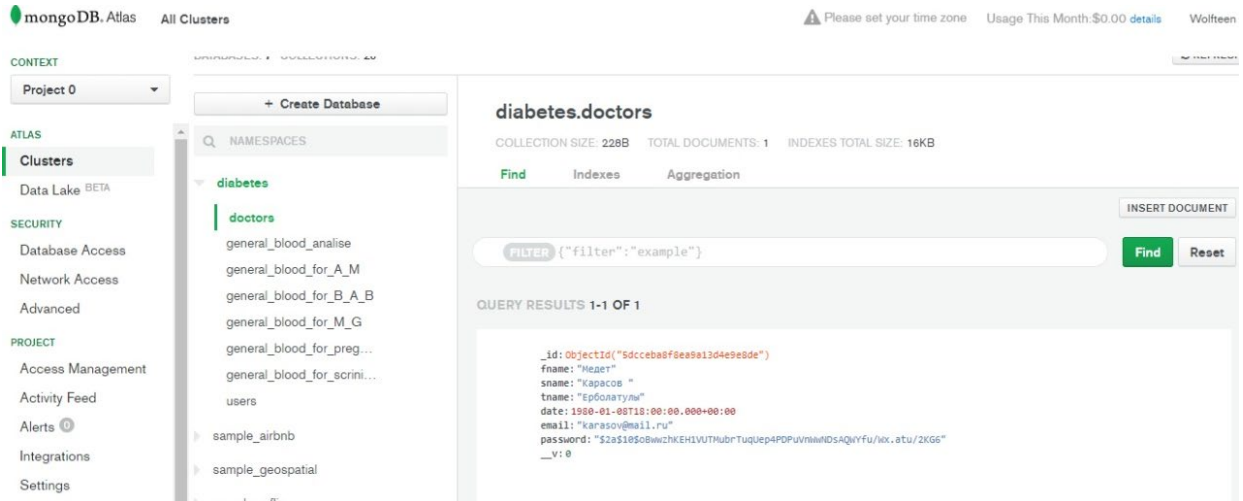


Рисунок В1. Регистрационные данные врача

Затем после процесса авторизации пользователя в главном меню системы в базе появляются загруженные данные (см. рис. В2).

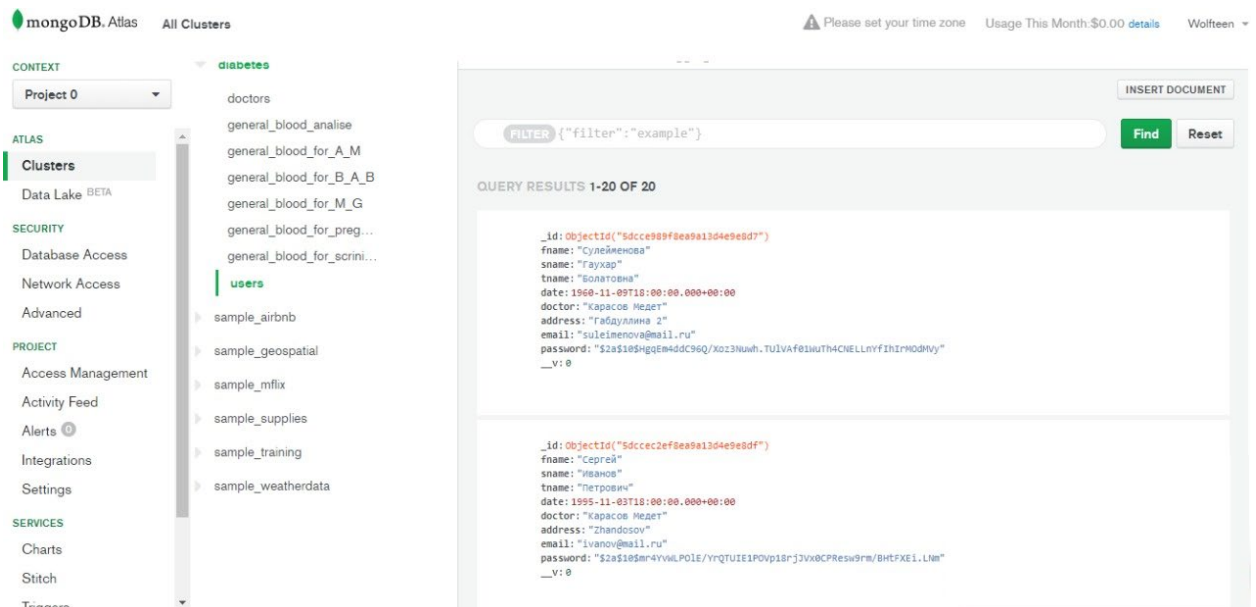


Рисунок В2. Регистрационные данные пользователя

Перечень анализов, которые необходимо ввести пользователем после регистрации в систему для диагностирования диабета (см. рис. В3).



Рисунок В3. Перечень анализов в MongoDB

Все виды анализов необходимо ввести и сохранить. Далее производится загрузка числовых параметров анализа «Общий анализ крови», где эти данные загружаются (см. рис. В4).

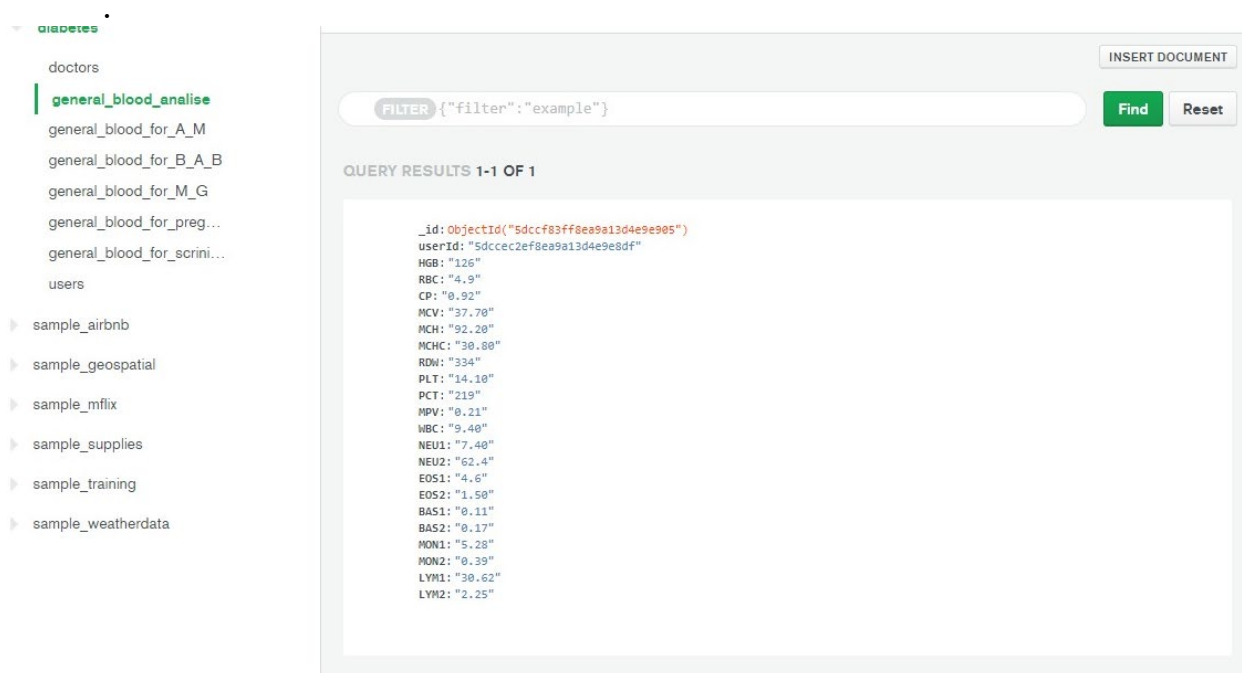


Рисунок В4. Числовые данные анализа «Общий анализ крови»

Затем идет загрузка числовых параметров анализа «Биохимический анализ крови», где эти данные загружаются в MongoDB (см. рис. В5).

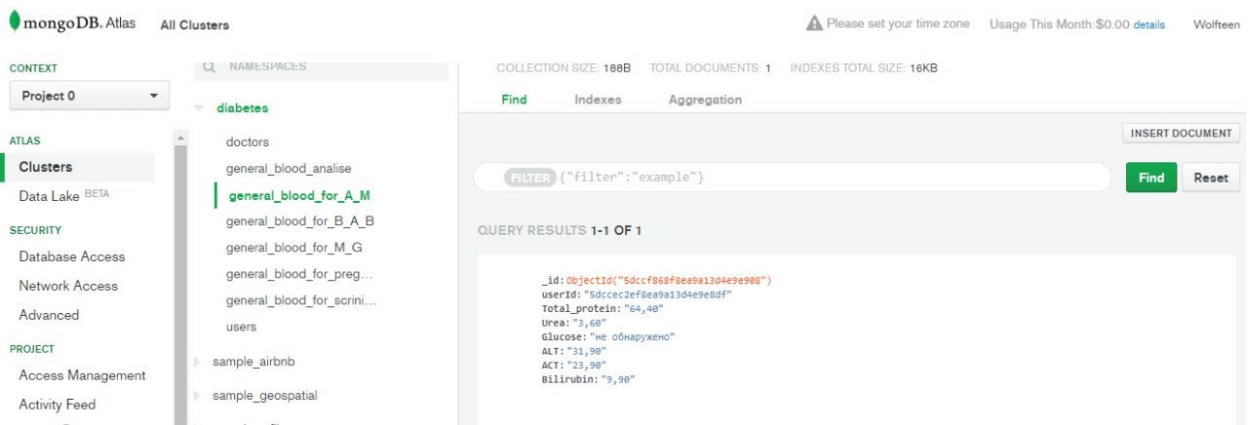


Рисунок В5. Числовые данные анализа «Биохимический анализ крови»

Далее идет загрузка числовых параметров анализа «Анализ мочи», где эти данные загружаются (см. рис. В6).

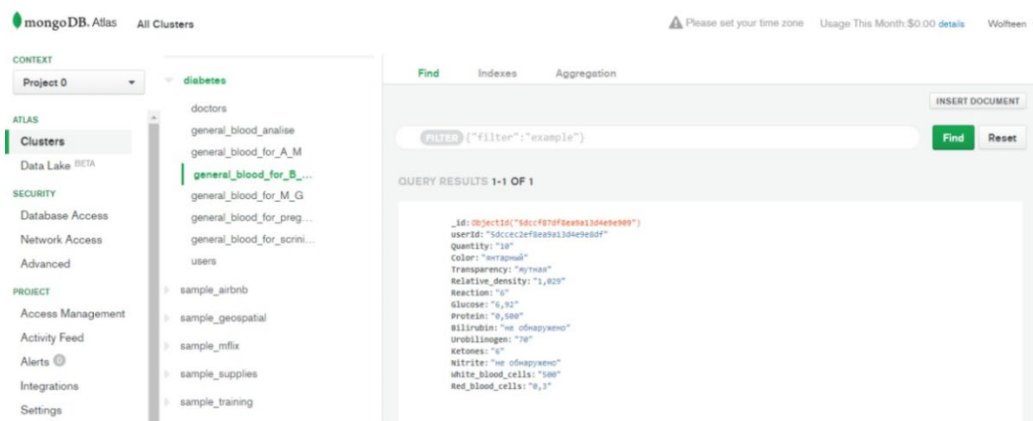


Рисунок В6. Числовые данные анализа «Анализ мочи»

Затем идет загрузка числовых параметров анализа «Анализ крови (содержание гормонов и медиаторов)», где эти данные загружаются в MongoDB (см. рис. В7).

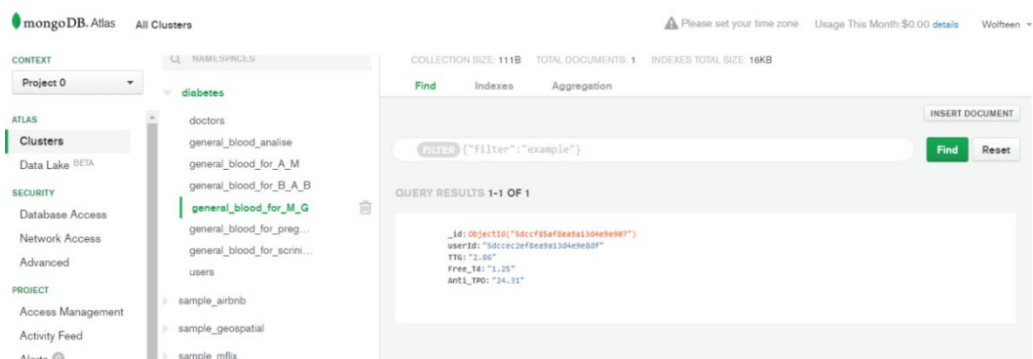


Рисунок В7. Числовые данные анализа «Анализ крови»

Далее идет загрузка числовых параметров анализа «Биохимический анализ крови (для беременных)», где эти данные загружаются (см. рис. В8).

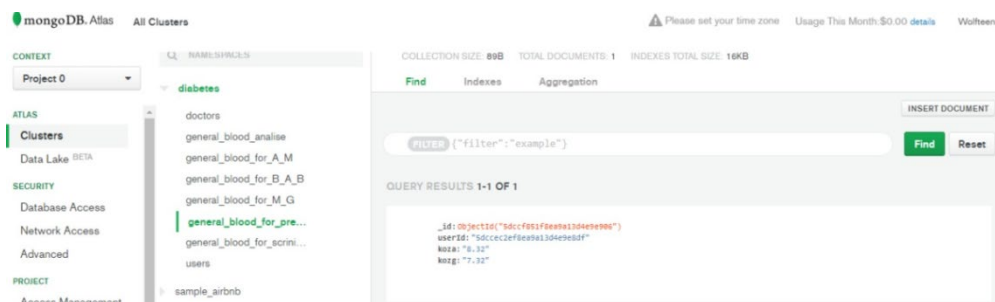


Рисунок В8. «Биохимический анализ крови (для беременных)»

Затем идет загрузка неструктурированных данных (снимки УЗИ сосудов нижних конечностей, УЗИ органов брюшной полости и почек, флюорография, электрокардиограмма, файлы датчиков FreeStyleLibre) различных форматов, где эти данные загружаются в MongoDB в историю болезни пользователя (см. рис. В9).

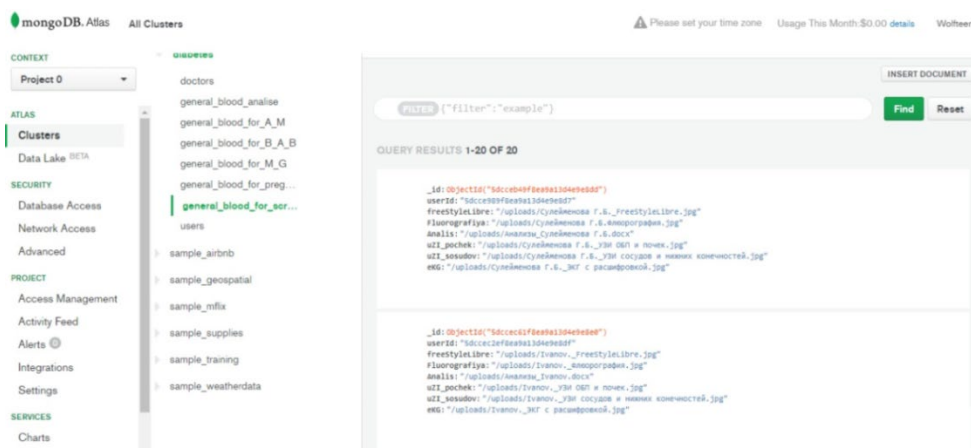


Рисунок В9. Загрузка неструктурированных данных

Загруженные параметры и снимки различных форматов хранятся в базе данных, где после их загрузки лечащий врач может наблюдать за состоянием пациента дистанционно (см. рис. В10).



Рисунок В10. Скрининг снимков, загруженных в MongoDB

Электронные файлы датчиков FreeStyleLibre могут загружаться с накопительной карты носителя, далее загружены в базу.

ПРИЛОЖЕНИЕ Г

Листинг программы

// В данном файле приведен пароль для подключения к MongoDB.

```
Keys.js
module.exports = {
  mongoURI: 'mongodb+srv://wolftteen:Wolfteen123@fullstack-nctpp.mongodb.net/diabetes',
  jwt: 'dev-jwt'
}
```

// В данном блоке реализована регистрация и авторизация пользователей сайта. При регистрации учитывается, зарегистрирована почта или нет, также при регистрации пароль кодируется с помощью bcrypt.js. В начале файла делаем соединение с базой данных, с моделями Mongoose User, Doctor.

```
Auth.js –controllers
const bcrypt = require('bcryptjs')
const jwt = require('jsonwebtoken')
const keys = require('../config/keys')
const User = require('../models/User')
const Doctor = require('../models/Doctor')
const error = require('../utils/errorHandler')
const ObjectId = require("mongodb").ObjectId;
const mongodb = require('mongodb')
const MongoClient = mongodb.MongoClient
module.exports.login = async function(req,res){
  const candidate = await User.findOne({email: req.body.email})
  if(candidate){
    const passwordResult = bcrypt.compareSync(req.body.password, candidate.password)
    if(passwordResult){
      const userId = candidate._id
      // if(candidate.role='doctor'){
      //   res.render('analysis')
      // }
      res.render('analysis', {userId:userId})
    }
  }
  else{
    const message = 'Введенный пароль неверен'
    res.render('login', {message:message})
  }
  }
  else{
    const message = 'Данный пользователь не найден'
    res.render('login', {message:message})
  }
}
module.exports.register = async function(req,res){
  const candidate = await User.findOne({email: req.body.email})
  if(candidate){
    const message = 'Выбранная почта занята'
    console.log(message)
    res.render('register', {message: message})
  }
}
```



```

else{
  const salt = bcrypt.genSaltSync(10)
  const password = req.body.password
  const user = new User({
    fname: req.body.fname,
    sname: req.body.sname,
    tname: req.body.tname,
    date: req.body.date,
    doctor: req.body.doctor,
    address: req.body.address,
    email: req.body.email,
    password: bcrypt.hashSync(password, salt)
  })
  try {
    user.save().then()
    res.render('login', {user:user})
  } catch (e) {
    errorHandler(res,e)
  }
}
}

```

// В данном файле реализован процесс введения различных анализов и загрузка в базу данных. Submit_f, submit_s, submit_t необходимы, чтобы получить данные, которые ввел пользователь. Затем, каждую контсанту разбиваем на значения, проверяем на корректность введенных значений, и если все правильно, то подключаемся к базе данных и загружаем. Если существует ошибка, то переводим на нашу страницу для ошибок Error handler.

```

module.exports.general_blood = function(req,res){
  const submit_f = req.body.submit_first_analisy
  const submit_s = req.body.submit_analysys_ber
  const submit_t = req.body.submit_analysys_G_M
  const submit_four = req.body.submit_analysys_B_A_K
  const submit_five = req.body.submit_analysys_A_M
  const submit_sixe = req.body.submit_analysys_scriningi
  if(submit_f){
    analisys = { userId: req.body.userId, HGB: req.body.HGB, RBC: req.body.RBC,
    CP:req.body.CP, MCV:req.body.MCV, MCH: req.body.MCH, MCHC: req.body.MCHC, RDW:
    req.body.RDW,
      PLT:req.body.PLT, PCT:req.body.PCT, MPV:req.body.MPV, WBC:req.body.WBC,
    NEU1:req.body.NEU1, NEU2: req.body.NEU2, EOS1:req.body.EOS1, EOS2:req.body.EOS2,
      BAS1: req.body.BAS1,BAS2: req.body.BAS2, MON1: req.body.MON1,
    MON2:req.body.MON2, LYM1: req.body.LYM1, LYM2: req.body.LYM2}
    MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
      const db = client.db('diabetes')
      db.collection('general_blood_analise').insertOne( analisys, function(err,result){
        if(err){
          res.send("Ошибка загрузки данных на сервер")
        }
      })
      console.log(analisys)
      res.render("general_blood", {analisys: analisys})
    })
  })
}

```

```

}
if(submit_s){
  analisys = {userId:req.body.userId, koza:req.body.koza, kozg: req.body.kozg}
  MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
    const db = client.db('diabetes')
    db.collection('general_blood_for_pregnant').insertOne( analisys, function(err,result){
      if(err){
res.send("Ошибка загрузки данных на сервер")
      }
      res.render("general_blood", {analisys: analisys})
    })
  })
}
if(submit_t){
  analisys = {userId:req.body.userId, TTG:req.body.TTG, Free_T4: req.body.Free_T4,
Anti_TPO: req.body.Anti_TPO}
  MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
    const db = client.db('diabetes')

    db.collection('general_blood_for_M_G').insertOne( analisys, function(err,result){
      if(err){
res.send("Ошибка загрузки данных на сервер")
      }
      res.render("general_blood", {analisys: analisys})
    })
  })
}
if(submit_four){
  analisys = {userId:req.body.userId, Quantity:req.body.Quantity, Color: req.body.Color,
Transparency: req.body.Transparency,
  Relative_density: req.body.Relative_density,Reaction: req.body.Reaction,Glucose:
req.body.Glucose,Protein: req.body.Protein,Bilirubin: req.body.Bilirubin,
  Urobilinogen: req.body.Urobilinogen,Ketones: req.body.Ketones,Nitrite:
req.body.Nitrite,White_blood_cells: req.body.White_blood_cells,Red_blood_cells:
req.body.Red_blood_cells
  }
  MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
    const db = client.db('diabetes')

    db.collection('general_blood_for_B_A_B').insertOne( analisys, function(err,result){
      if(err){
res.send("Ошибка загрузки данных на сервер")
      }
      res.render("general_blood", {analisys: analisys})
    })
  })
}
if(submit_five){
  analisys = {userId:req.body.userId, Total_protein:req.body.Total_protein, Urea:
req.body.Urea, Glucose: req.body.Glucose,
  ALT: req.body.ALT,ACT: req.body.ACT,Bilirubin: req.body.Bilirubin
  }
}

```

```

MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
  const db = client.db('diabetes')
  db.collection('general_blood_for_A_M').insertOne( analisys, function(err,result){
    if(err){
res.send("Ошибка загрузки данных на сервер")
    }
    res.render("general_blood", {analisys: analisys})
  })
})
}
if(submit_sixe){
  const freeStyleLibre =
(req.files['freeStyleLibre'][0].destination+"/"+req.files['freeStyleLibre'][0].originalname).substr
(8);
  const fluorografiya =
(req.files['Fluorografiya'][0].destination+"/"+req.files['Fluorografiya'][0].originalname).substr
(8);
  const analis =
(req.files['Analis'][0].destination+"/"+req.files['Analis'][0].originalname).substr(8);
  const uZI_pocek =
(req.files['UZI_pocek'][0].destination+"/"+req.files['UZI_pocek'][0].originalname).substr(8)
;
  const uZI_sosudov =
(req.files['UZI_sosudov'][0].destination+"/"+req.files['UZI_sosudov'][0].originalname).substr(
8);
  const eKG =
(req.files['EKG'][0].destination+"/"+req.files['EKG'][0].originalname).substr(8);
  console.log(freeStyleLibre)
  console.log(analis)
  console.log(uZI_pocek)
  console.log(uZI_sosudov)
  console.log(eKG)
  analisys = {userId:req.body.userId, freeStyleLibre:freeStyleLibre,
Fluorografiya:fluorografiya , Analis:analis,
  uZI_pocek:uZI_pocek ,uZI_sosudov:uZI_sosudov ,eKG: eKG
  }
  console.log(analisys)
  MongoClient.connect(keys.mongoURI, {useNewUrlParser:true,
useUnifiedTopology:true}, (error, client)=>{
  const db = client.db('diabetes')

  db.collection('general_blood_for_scriningi').insertOne( analisys, function(err,result){
    if(err){
res.send("Ошибка загрузки данных на сервер")
    }
    res.render("general_blood", {analisys: analisys})
  })
})
}
}
module.exports.general_blood_all = async function(req,res){
  const userId= req.params.id

```

```

MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
  const db = client.db('diabetes')
  db.collection('general_blood_analise').findOne({userId: userId }, function(err, analisys){
    res.render("general_blood", {analisys: analisys})
  })
})
}
module.exports.general_blood_two = async function(req,res){
  const userId= req.params.id
  MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
    const db = client.db('diabetes')
    db.collection('general_blood_for_pregnant').findOne({userId: userId }, function(err,
analisys){
      res.render("general_blood_two", {analisys: analisys})
    })
  })
}
module.exports.general_blood_three = async function(req,res){
  const userId= req.params.id
  MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
    const db = client.db('diabetes')
    db.collection('general_blood_for_M_G').findOne({userId: userId }, function(err, analisys){
      res.render("general_blood_three", {analisys: analisys})
    })
  })
}
module.exports.general_blood_four = async function(req,res){
  const userId= req.params.id
  MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
    const db = client.db('diabetes')
    db.collection('general_blood_for_A_M').findOne({userId: userId }, function(err, analisys){
      res.render("general_blood_four", {analisys: analisys})
    })
  })
}
module.exports.general_blood_five = async function(req,res){
  const userId= req.params.id
  MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
    const db = client.db('diabetes')
    db.collection('general_blood_for_B_A_B').findOne({userId: userId }, function(err,
analisys){
      res.render("general_blood_five", {analisys: analisys})
    })
  })
}
module.exports.general_blood_six = async function(req,res){
  const userId= req.params.id
  MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
    const db = client.db('diabetes')
    db.collection('general_blood_for_scriningi').findOne({userId: userId }, function(err,
analisys){
      res.render("general_blood_six", {analisys: analisys})
    })
  })
}

```

```

    })
  })
}

```

// В данном файле реализован процесс регистрации врача. Получаем данные, проверяем почту, если почта уже существует, выдается сообщение «Выбранная почта занята». При правильном заполнении данных, пароль кодируется «всурпт», затем данные загружаются в MongoDB.

```

module.exports.doctor_register = async function (req,res) {
  res.render("doctor_reg")
}
module.exports.doctor_reg = async function (req,res) {
  const candidate = await Doctor.findOne({email: req.body.email})
  if(candidate){
    const message = 'Выбранная почта занята'
    console.log(message)
    res.render('doctor_reg', {message: message})
  }
  else{
    const salt = bcrypt.genSaltSync(10)
    const password = req.body.password
    const doctor = new Doctor({
      fname: req.body.fname,
      sname: req.body.sname,
      tname: req.body.tname,
      date: req.body.date,

      email: req.body.email,
      password: bcrypt.hashSync(password, salt)
    })
    try {
      doctor.save().then()
      res.render('doctor_log', {doctor:doctor})
    } catch (e) {
      errorHandler(res,e)
    }
  }
}
module.exports.doctor_log = async function (req,res) {
  res.render("doctor_log")
}
module.exports.doctor_login = async function(req,res){
  const candidate = await Doctor.findOne({email: req.body.email})
  if(candidate){
    const passwordResult = bcrypt.compareSync(req.body.password, candidate.password)
    if(passwordResult){
      const userId = candidate._id
      MongoClient.connect(keys.mongoURI, {useNewUrlParser:true}, (error, client)=>{
        const db = client.db('diabetes')
        const analisys = db.collection('users').find().toArray(function(err, analisys){
          res.render('all_pacients', {analisys: analisys})
        })
      })
    }
  }
}

```

```

    }
    else{
        const message = 'Введенный пароль неверен'
        res.render('doctor_log', {message:message})
    }
}
else{
    const message = 'Данный пользователь не найден'
    res.render('doctor_log', {message:message})
}
}
const mongoose = require('mongoose')
module.exports.connect = function(req,res){
    mongoose.connect(keys.mongoURI, {useUnifiedTopology: true, useNewUrlParser: true,
    useCreateIndex: true,})
    .then(()=> res.status(200).json({message:'MongoDB connected.'}))
    .catch(error => res.status(200).json({message:'Error.'+error}))
}

```

// В данном файле реализован процесс «Passport.js- middleware», где проверяется, является ли пользователь нашего сайта авторизованным. Если нет то на страницы «analysis», «doctors» посетитель не получит доступа. Это необходимо для того чтобы злоумышленник не использовал «sql» инъекции.

```

const JwtStrategy = require('passport-jwt').Strategy,
    ExtractJwt = require('passport-jwt').ExtractJwt;
const keys = require('../config/keys')
const mongoose = require('mongoose')
const User = mongoose.model('users')
const opts = {
    jwtFromRequest: ExtractJwt.fromAuthHeaderAsBearerToken(),
    secretOrKey: keys.jwt
}
module.exports = passport =>{
    passport.use(
        new JwtStrategy(opts, async function(payload, done) {
            try{
                const user = await User.findById(payload.userId).select('email id')
                if(user){
                    done(null, user)
                }
                else{
                    done(null, false)
                }
            }
            catch(e){
                console.log(e)
            }
        })
    )
}
Upload.js – middleware

```

// В данном «Upload.js – middleware» файле получаем данные и загружаем в папку «uploads». В настройках можно изменить следующие значения: формат файла, размер файла, куда загрузить данные и под каким именем их необходимо загрузить. Для того чтобы не загружать данные с одинаковыми именами, было принято решение о необходимости перед началом файла, поставить дату и время с учетом до секунды.

```
const multer = require('multer')
const moment = require('moment')
const storage = multer.diskStorage({
  destination(req, file, cb){
    cb(null, 'uploads/')
  },
  filename(req, file, cb){
    const date = moment().format('DDMMYYYY-HH:mm:ss_SSS')
    cb(null, `${date}-${file.originalname}`)
  }
})
const fileFilter = (req, file, cb)=>{
  if(file.mimetype === 'image/png' || file.mimetype === 'image/jpeg' || file.mimetype === 'image/jpg'){
    cb(null, true)
  }
  else{
    cb(null, false)
  }
}
const limits = {
  fileSize: 1024*1024*5
}
module.exports = multer({
  storage,fileFilter,limits
})
```

// В данном блоке реализован процесс Mongoose, где строим модель врача, указываем тип для атрибутов и их свойств.

```
Doctor.js – model
const mongoose = require('mongoose')
const Schema = mongoose.Schema
const doctorSchema = new Schema({
  fname:{
    type: String,
    required: true
  },
  sname:{
    type: String,
    required: true
  },
  tname:{
    type: String
  },
  email:{
    type: String,
    required: true,
```

```

    unique: true
  },
  password: {
    type: String,
    required: true
  },
  date: {
    type: Date
  }
})
module.exports = mongoose.model('doctors', doctorSchema)

```

// В Mongoose строим модель пациента, указываем тип для атрибутов и их свойств.

```

user.js
const mongoose = require('mongoose')
const Schema = mongoose.Schema
const userSchema = new Schema({
  fname: {
    type: String,
    required: true
  },
  sname: {
    type: String,
    required: true
  },
  tname: {
    type: String
  },
  email: {
    type: String,
    required: true,
    unique: true
  },
  password: {
    type: String,
    required: true
  },
  date: {
    type: Date
  },
  doctor: {
    type: String
  },
  address: {
    type: String,
    required: true
  }
})
module.exports = mongoose.model('users', userSchema)

```

// Роутеры необходимы для того чтобы реализовать переход пользователя от одной страницы на другую. С помощью метода «get» пользователь может увидеть путь, на

котором он находится, когда пользователь смотрит свои анализы. При «post» запросе пользователь не видит путь, по которому он перешел, это необходимо при регистрации и авторизации, чтобы пользователь при отправке данных на сервер не имел возможности просмотреть введенные данные в URL.

analysys.js – router

```
const express = require('express')
const controller = require('../controllers/analysys')
const router = express.Router()
const passport = require('passport')
const upload = require('../middleware/upload')
router.get("/", passport.authenticate('jwt', {session:false}), (req,res)=>{
  res.render('analysys')
})
router.post("/general_blood", passport.authenticate('jwt', {session:false}),
controller.general_blood)
module.exports = router
```

auth.js – routes

```
const express = require('express')
const controller = require('../controllers/auth')
const router = express.Router()
const multer = require('multer')
const moment = require('moment')
const storage = multer.diskStorage({
  destination: function(req,file,cb){
    cb(null, './public/uploads')
  },
  filename: function(req,file,cb){
    cb(null, file.originalname)
  }
})
const limits = {
  fileSize: 1024*1024*5
}
const upload = multer({storage, limits})
router.get('/', (req,res)=>{
  res.render('index')
})
router.post("/login", upload.fields([
  { name: 'freeStyleLibre'}, { name: 'Fluorografiya'},
  {name:'Analis'}, { name: 'UZI_pochek'}, { name: 'UZI_sosudov'}, {name:'EKG'}]),
controller.login)
router.post("/register", upload.fields([
  { name: 'freeStyleLibre'}, { name: 'Fluorografiya'},
  {name:'Analis'}, { name: 'UZI_pochek'}, { name: 'UZI_sosudov'}, {name:'EKG'}]),
controller.register)
router.get("/register", upload.fields([
  { name: 'freeStyleLibre'}, { name: 'Fluorografiya'},
  {name:'Analis'}, { name: 'UZI_pochek'}, { name: 'UZI_sosudov'}, {name:'EKG'}]), (req,res)=>{
  res.render("register")
})
router.get("/login", upload.fields([
  { name: 'freeStyleLibre'}, { name: 'Fluorografiya'},
  {name:'Analis'}, { name: 'UZI_pochek'}, { name: 'UZI_sosudov'}, {name:'EKG'}]), (req,res)=>{
  res.render("login")
})
```

```

router.get("/login/analysis", upload.fields([ { name: 'freeStyleLibre'}, { name: 'Fluorografiya'},
{name:'Analis'}, { name: 'UZI_pochek'}, { name: 'UZI_sosudov'}, {name:'EKG'}]), (req,res)=>{
  res.render('analysis')
})
// http://localhost:5000/api/auth/login/
router.post("/analysis/general_blood", upload.fields([ { name: 'freeStyleLibre'}, { name:
'Fluorografiya'}, {name:'Analis'}, { name: 'UZI_pochek'}, { name: 'UZI_sosudov'},
{name:'EKG'}]), controller.general_blood)
router.get("/analysis/:id", controller.general_blood_all)
router.get("/blood_analysis/:id", controller.general_blood_two)
router.get("/blood_analysis_g/:id", controller.general_blood_three)
router.get("/b_blood_analysis/:id", controller.general_blood_four)
router.get("/Analysis_of_urine/:id", controller.general_blood_five)
router.get("/analysis_scrining/:id", controller.general_blood_six)
router.get("/doctor/register", controller.doctor_register)
router.post("/doctor/register", controller.doctor_reg)
router.get("/doctor/login", controller.doctor_log)
router.post("/doctor/login", controller.doctor_login)
router.get("/connect", controller.connect)
module.exports = router

```

// В данном блоке реализован процесс более быстрой загрузки веб приложения. Меню и «footer» сайта разделили с помощью HBS. Это необходимо для того чтобы, при обновлении сайт загружал только контент. В начале файла указан «title», это заголовок сайта, после идет тег мета, он нужен для того чтобы наш сайт правильно загружал текст на кириллице. Далее идет «css» стили и подключение к «JS».

```

header.hbs
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Diabetes - быстроеопределениедиабета</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link
href="https://fonts.googleapis.com/css?family=Work+Sans:100,200,300,400,500,600,700,800,9
00" rel="stylesheet">
<link rel="stylesheet" href="/css/open-iconic-bootstrap.min.css">
<link rel="stylesheet" href="/css/animate.css">
<link rel="stylesheet" href="/css/owl.carousel.min.css">
<link rel="stylesheet" href="/css/owl.theme.default.min.css">
<link rel="stylesheet" href="/css/magnific-popup.css">
<link rel="stylesheet" href="/css/aos.css">
<link rel="stylesheet" href="/css/ionicons.min.css">
<link rel="stylesheet" href="/css/bootstrap-datepicker.css">
<link rel="stylesheet" href="/css/jquery.timepicker.css">
<link rel="stylesheet" href="/css/flaticon.css">
<link rel="stylesheet" href="/css/icomoon.css">
<link rel="stylesheet" href="/css/style.css">
<link rel="stylesheet" href="/css/main.css">
</head>
<body>
Footer.hbs

```

```

<script src="/js/jquery.min.js"></script>
<script src="/js/jquery-migrate-3.0.1.min.js"></script>
<script src="/js/popper.min.js"></script>
<script src="/js/bootstrap.min.js"></script>
<script src="/js/jquery.easing.1.3.js"></script>
<script src="/js/jquery.waypoints.min.js"></script>
<script src="/js/jquery.stellar.min.js"></script>
<script src="/js/owl.carousel.min.js"></script>
<script src="/js/jquery.magnific-popup.min.js"></script>
<script src="/js/aos.js"></script>
<script src="/js/jquery.animateNumber.min.js"></script>
<script src="/js/bootstrap-datepicker.js"></script>
<script src="/js/jquery.timepicker.min.js"></script>
<script src="/js/scrollax.min.js"></script>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBVWaKrjvy3MaE7SQ74_uJiULglIj
Y0H2s&sensor=false"></script>
<script src="/js/google-map.js"></script>
<script src="/js/main.js"></script>
</body>
</html>

```

//Данные обо всех пользователях. В «container» создается таблица, в которую будут загружаться данные пользователей. Данная страница доступна только для врачей.

All_pacients.hbs

```

{{>header}}
<div class="container">
<table class="table">
<thead class="thead-dark">
<tr>
<th scope="col">Имя</th>
<th scope="col">Фамилия</th>
<th scope="col">Email</th>
<th scope="col">Лечащий врач</th>
</tr>
</thead>
<tbody>
  {{#each analisis}}
<tr>
<a href="#">
<td><a href="/analysis/{{_id}}">{{fname}}</a></td>
<td><a href="/analysis/{{_id}}">{{sname}}</a></td>
<td><a href="/analysis/{{_id}}">{{email}}</a></td>
<td><a href="#">{{doctor}}</a></td>
</a>
</tr>
  {{/each}}
</tbody>
</table>
</div>

```

//В данной странице пользователи могут загрузить свои анализы

```

Analisis.hbs
{{> header}}
<section class="hero-wrap hero-wrap-2" style="background-image: url('/images/bg_1.jpg');"
data-stellar-background-ratio="0.5">
<div class="overlay"></div>
<div class="container">
<div class="row no-gutters slider-text align-items-center justify-content-center">
<div class="col-md-9 ftco-animate text-center">
<h1 class="mb-2 bread">Анализы пользователей</h1>
<p class="breadcrumbs"><span class="mr-2"><a href="/">Главная <i class="ion-ios-
arrow-forward"></i></a></span><span>Анализы <i class="ion-ios-arrow-
forward"></i></span></p>
<p class="breadcrumbs"><span class="mr-2"><a href="/">Главная <i class="ion-ios-
arrow-forward"></i></a></span><span><a href="/analisis/{{userId}}">Просмотр
анализов <i class="ion-ios-arrow-forward"></i></a></span></p>
</div>
</div>
</div>
</section>
<section class="ftco-section" id="services">
<div class="container">
<div class="ftco-departments">
<div class="row">
<div class="col-md-12 nav-link-wrap">
<div class="nav nav-pills d-flex text-center" id="v-pills-tab" role="tablist" aria-
orientation="vertical">
<a class="nav-link ftco-animate active" id="v-pills-1-tab" data-toggle="pill" href="#v-
pills-1" role="tab" aria-controls="v-pills-1" aria-selected="true">Общий анализ крови</a>
<a class="nav-link ftco-animate" id="v-pills-2-tab" data-toggle="pill" href="#v-pills-2"
role="tab" aria-controls="v-pills-2" aria-selected="false">Анализ крови (беременные)</a>
<a class="nav-link ftco-animate" id="v-pills-3-tab" data-toggle="pill" href="#v-pills-3"
role="tab" aria-controls="v-pills-3" aria-selected="false">Анализ крови (гармоны,
медуаторы)</a>
<a class="nav-link ftco-animate"
id="v-pills-4-tab" data-toggle="pill" href="#v-pills-4" role="tab" aria-controls="v-pills-4"
aria-selected="false">Биохимический анализ крови</a>
<a class="nav-link ftco-animate" id="v-pills-5-tab" data-toggle="pill" href="#v-pills-5"
role="tab" aria-controls="v-pills-5" aria-selected="false">Анализ мочи</a>
<a class="nav-link ftco-animate" id="v-pills-6-tab" data-toggle="pill" href="#v-pills-6"
role="tab" aria-controls="v-pills-6" aria-selected="false">Результаты скринингов</a>
</div>
</div>
<div class="col-md-12 tab-wrap">
<div class="tab-content bg-light p-4 p-md-5 ftco-animate" id="v-pills-tabContent">
<div class="tab-pane py-2 fade show active" id="v-pills-1" role="tabpanel" aria-
labelledby="day-1-tab">
<div class="row departments">
<div class="col-lg-4 order-lg-last d-flex align-items-stretch">
<div class="img d-flex align-self-stretch" style="background-
image: url('/images/dept-1.jpg');"></div>

```

```

</div>
<div class="col-lg-8">
    <h2>Общий анализ крови</h2>
    <p> Врачебный анализ, позволяющий оценить содержание
    гемоглобина в системе красной крови, количество эритроцитов, цветовой показатель,
    количество лейкоцитов, тромбоцитов. </p>
<div class="row mt-5 pt-2">
<div class="col-lg-12">
<form action="analysis/general_blood" class="" method="POST">
<div class="d-md-flex">
<div class="form-group col-lg-4">
<input type="text" class="form-control" placeholder="Гемоглобин" name="HGB" required>
</div>
<div class="form-group col-lg-4 ml-1">
<input type="text" class="form-control" placeholder="Эритроциты" name="RBC"
required>
</div>
<div class="form-group col-lg-4 ml-1">
    <input type="text" class="form-control" placeholder="Цветной показатель"
name="CP" required>
</div>
<div class="form-group col-lg-4 ml-1">
    <input type="hidden" class="form-control" value="{{userId}}" name="userId"
required>
</div>
</div>
<div class="d-md-flex">
<div class="form-group col-lg-4">
    <input type="text" class="form-control" placeholder="Средний объем эритроцита"
name="MCV" required>
</div>
<div class="form-group col-lg-4 ml-md-1">
    <input type="text" class="form-control" placeholder="Среднее содержание Hb в
эритроците" name="MCH" required>
</div>
<div class="form-group col-lg-4 ml-md-1">
    <input type="text" class="form-control" placeholder="Средняя концентрация Hb в
эритроците" name="MCHC" required>
</div>
</div>
<div class="d-md-flex">
<div class="form-group col-lg-4">
    <input type="text" class="form-control" placeholder="Распределение эритроцитов
по объему" name="RDW" required>
</div>
<div class="form-group col-lg-4 ml-md-1">
    <input type="text" class="form-control" placeholder="Тромбоциты"
name="PLT" required>
</div>
<div class="form-group col-lg-4 ml-md-1">
    <input type="text" class="form-control" placeholder="Тромбокрит" name="PCT"
required>

```

```

</div>
  </div>
<div class="d-md-flex">
  <div class="form-group col-lg-4">
    <input type="text" class="form-control" placeholder="Средний объем тромбоцита"
name="MPV" required>
  </div>
  <div class="form-group col-lg-4 ml-md-1">
    <input type="text" class="form-control" placeholder="Лейкоциты"
name="WBC" required>
  </div>
<div class="form-group col-lg-4 ml-md-1">
  <input type="text" class="form-control" placeholder="Нейтрофилы"
name="NEU1" required>
</div>
</div>
  <div class="d-md-flex">
    <div class="form-group col-lg-4">
      <input type="text" class="form-control" placeholder="Нейтрофилы (абс. кол-
во)" name="NEU2" required>
    </div>
    <div class="form-group col-lg-4 ml-md-1">
      <input type="text" class="form-control" placeholder="Эозинофилы"
name="EOS1" required>
    </div>
    <div class="form-group col-lg-4 ml-md-1">
      <input type="text" class="form-control" placeholder="Эозинофилы (абс. кол-
во)" name="EOS2" required>
    </div>
  </div>
  <div class="d-md-flex">
    <div class="form-group col-lg-4">
      <input type="text" class="form-control" placeholder="Базофилы"
name="BAS1" required>
    </div>
    <div class="form-group col-lg-4 ml-md-1">
      <input type="text" class="form-control" placeholder="Базофилы (абс. кол-во)"
name="BAS2" required>
    </div>
    <div class="form-group col-lg-4 ml-md-1">
      <input type="text" class="form-control" placeholder="Моноциты" name="MON1"
required>
    </div>
  </div>
  <div class="d-md-flex">
    <div class="form-group col-lg-4">
      <input type="text" class="form-control" placeholder="Моноциты (абс. кол-
во)" name="MON2" required>
    </div>
    <div class="form-group col-lg-4 ml-md-1">
      <input type="text" class="form-control" placeholder="Лимфоциты"
name="LYM1" required>
    </div>
  </div>

```

```

</div>
    <div class="form-group col-lg-4 ml-md-1">
        <input type="text" class="form-control" placeholder="Лимфоциты (абс. кол-
во)" name="LYM2" required>
    </div>
</div>
    <div class="d-md-flex">
        <div class="form-group col-lg-4 ">
            <input type="submit" class="form-control" value="Отправить анализы"
name="submit_first_analisis" required>
        </div>
    </div>
</form>
</div>
</div>
</div>
</div>
</div>
<div class="tab-pane fade" id="v-pills-2" role="tabpanel" aria-labelledby="v-pills-day-2-
tab">
    <div class="row departments">
        <div class="col-lg-4 order-lg-last d-flex align-items-stretch">
            <div class="img d-flex align-self-stretch" style="background-
image: url(/images/dept-2.jpg);"></div>
        </div>
        <div class="col-lg-8">
            <h2>Анализ крови при беременности</h2>
            <p>Анализы при беременности – это необходимое условие для
нормального развития, роста и рождения ребенка, а также сохранения здоровья
будущей матери. </p>
        </div>
    </div>
<div
class="row mt-5 pt-2">
    <div
class="col-lg-12">
        <form action="analysis/general_blood" class="" method="POST">
            <div class="d-md-flex">
                <div class="form-group col-lg-4">
                    <input type="text" class="form-control" placeholder="Кожа" name="koza" required>
                </div>
                <div class="form-group col-lg-4 ml-1">
                    <input type="text" class="form-control" placeholder="Гликозилированный гемоглобин"
name="kozg" required>
                </div>
                <div class="form-group col-lg-4 ml-1">
                    <input type="hidden" class="form-control" name="userId" value="{{userId}}" required>
                </div>
            </div>
            <div class="d-md-flex">
                <div class="form-group col-lg-4 ">
                    <input type="submit" class="form-control" value="Отправить анализы"
name="submit_analisis_ber" required>
                </div>

```

```

</div>
</form>
</div>
</div>
</div>
</div>
</div>
<div class="tab-pane fade" id="v-pills-3" role="tabpanel" aria-labelledby="v-pills-
day-3-tab">
  <div class="row departments">
    <div class="col-lg-4 order-lg-last d-flex align-items-stretch">
      <div class="img d-flex align-self-stretch" style="background-
image: url(/images/dept-3.jpg);"></div>
    </div>
    <div class="col-md-8">
      <h2>Анализ крови - содержание гормонов и медиаторов</h2>
      <p>С помощью анализа на гормоны можно выявить многие
патологии, в их числе:<br />- дисфункции щитовидной железы (гипо- и гипертиреоз, зоб,
тиреоидит); <br />-бесплодие, нарушения менструального цикла; <br />-сахарный и
несахарный диабет; <br />-опухолевые процессы; <br />-ожирение (дисфункция
эндокринных клеток поджелудочной железы); <br />-заболевания гипоталамо-
гипофизарной системы (болезнь Иценко-Кушинга, акромегалия и др.); <br />-
надпочечниковая недостаточность.</p>
    <div class="row mt-5 pt-2">
      <div
class="col-lg-12">
        <form action="analysys/general_blood" class="" method="POST">
          <div class="d-md-flex">
            <div class="form-group col-lg-4">
              <input type="text" class="form-control" placeholder="ТТГ" name="TTG" required>
            </div>
            <div class="form-group col-lg-4 ml-1">
              <input type="text" class="form-control" placeholder="Свободный Т4" name="Free_T4"
required>
            </div>
            <div class="form-group col-lg-4">
              <input type="text" class="form-control" placeholder="Анти ТПО" name="Anti_TPO"
required>
            </div>
            <div class="form-group col-lg-4 ml-1">
              <input type="hidden" class="form-control" name="userId" value="{{userId}}" required>
            </div>
          </div>
          <div class="d-md-flex">
            <div class="form-group col-lg-4 ">
              <input type="submit" class="form-control" value="Отправить анализы"
name="submit_analysys_G_M" required>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>

```



```

        </div>
        </div>
    </div>
    <div class="tab-pane fade" id="v-pills-4" role="tabpanel" aria-labelledby="v-pills-
day-4-tab">
        <div class="row departments">
            <div class="col-lg-4 order-lg-last d-flex align-items-stretch">
                <div class="img d-flex align-self-stretch" style="background-
image: url(/images/dept-4.jpg);"></div>
            </div>
            <div class="col-md-8">
                <h2>Биохимический анализ крови</h2>
                <p>Метод лабораторной диагностики, который позволяет
оценить работу внутренних органов (печень, почки, поджелудочная железа, желчный
пузырь и др.), получить информацию о метаболизме (обмен липидов, белков, углеводов),
выяснить потребность в микроэлементах.</p>
            <div class="row mt-5 pt-2">
                <div class="col-lg-12">
                    <form action="analysis/general_blood" class="" method="POST">
                        <div class="d-md-flex">
                            <div class="form-group col-lg-4">
                                <input type="text" class="form-control" placeholder="Общий белок" name="Total_protein"
required>
                            </div>
                            <div class="form-group col-lg-4 ml-1">
                                <input type="text" class="form-control" placeholder="Мочевина" name="Urea" required>
                            </div>
                            <div class="form-group col-lg-4">
                                <input type="text" class="form-control" placeholder="Глюкоза" name="Glucose" required>
                            </div>
                            <div class="form-group col-lg-4 ml-1">
                                <input type="hidden" class="form-control" name="userId" value="{{userId}}" required>
                            </div>
                        </div>
                        <div class="d-md-flex">
                            <div class="form-group col-lg-4">
                                <input type="text" class="form-control" placeholder="АЛТ" name="ALT" required>
                            </div>
                            <div class="form-group col-lg-4 ml-1">
                                <input type="text" class="form-control" placeholder="АСТ" name="ACT" required>
                            </div>
                            <div class="form-group col-lg-4">
                                <input type="text" class="form-control" placeholder="Билирубин (общий)" name="Bilirubin"
required>
                            </div>
                        </div>
                        <div class="d-md-flex">
                            <div class="form-group col-lg-4 ">
                                <input type="submit" class="form-control" value="Отправить анализы"
name="submit_analysys_A_M" required>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>

```

```

</form>
</div>
</div>
</div>
</div>
</div>
<div class="tab-pane fade" id="v-pills-5" role="tabpanel" aria-labelledby="v-pills-
day-5-tab">
  <div class="row departments">
    <div class="col-lg-4 order-lg-last d-flex align-items-stretch">
      <div class="img d-flex align-self-stretch" style="background-
image: url(/images/dept-5.jpg);"></div>
    </div>
    <div class="col-md-8">
      <h2>Анализ мочи</h2>
      <p>Одно из самых частых лабораторных исследований,
которое проводится в диагностических целях. Он назначается при многих заболеваниях и
включает в себя определение до 20 показателей, каждый из которых помогает в
постановке правильного диагноза.</p>
    <div class="row mt-5 pt-2">
      <div
class="col-lg-12">
        <form action="analisys/general_blood" class="" method="POST">
          <div class="d-md-flex">
            <div class="form-group col-lg-4">
              <input type="text" class="form-control" placeholder="Количество" name="Quantity"
required>
            </div>
            <div class="form-group col-lg-4 ml-1">
              <input type="text" class="form-control" placeholder="Цвет" name="Color" required>
            </div>
            <div class="form-group col-lg-4">
              <input type="text" class="form-control" placeholder="Прозрачность" name="Transparency"
required>
            </div>
            <div class="form-group col-lg-4 ml-1">
              <input type="hidden" class="form-control" name="userId" value="{{userId}}" required>
            </div>
            </div>
          <div class="d-md-flex">
            <div class="form-group col-lg-4">
              <input type="text" class="form-control" placeholder="Относительная плотность"
name="Relative_density" required>
            </div>
            <div class="form-group col-lg-4 ml-1">
              <input type="text" class="form-control" placeholder="Реакция" name="Reaction" required>
            </div>
            <div class="form-group col-lg-4">
              <input type="text" class="form-control" placeholder="Белок" name="Protein" required>
            </div>
            </div>
          <div class="d-md-flex">

```

```

<div class="form-group col-lg-4">
<input type="text" class="form-control" placeholder="Глюкоза" name="Glucose" required>
</div>
<div class="form-group col-lg-4 ml-1">
<input type="text" class="form-control" placeholder="Билирубин" name="Bilirubin"
required>
</div>
<div class="form-group col-lg-4 ml-1">
<input type="text" class="form-control" placeholder="Уробилиноген" name="Urobilinogen"
required>
</div>
</div>
<div class="d-md-flex">
<div class="form-group col-lg-4">
<input type="text" class="form-control" placeholder="Кетоны" name="Ketones" required>
</div>
<div class="form-group col-lg-4 ml-1">
<input type="text" class="form-control" placeholder="Нитриты" name="Nitrite" required>
</div>
<div class="form-group col-lg-4 ml-1">
<input type="text" class="form-control" placeholder="Лейкоциты в полоске"
name="White_blood_cells" required>
</div>
</div>
<div class="d-md-flex">
<div class="form-group col-lg-4 ml-1">
<input type="text" class="form-control" placeholder="Эритроциты в полоске"
name="Red_blood_cells" required>
</div>
<div class="form-group col-lg-4 ">
<input type="submit" class="form-control" value="Отправить анализы"
name="submit_analysys_B_A_K" required>
</div>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
<div class="tab-pane fade" id="v-pills-6" role="tabpanel" aria-labelledby="v-pills-day-6-
tab">
<div class="row departments">
<div class="col-lg-12">
<h2>Результаты скринингов</h2>
<form action="analisis/general_blood" class="" method="POST" enctype="multipart/form-
data">
<div class="d-md-flex">
<div class="form-group col-lg-4">
<label for="freeStyleLibre">FreeStyleLibre</label>
<input type="file" class="form-control" placeholder="FreeStyleLibre" name="freeStyleLibre"
required>

```


//В данном файле хранится html код для авторизации врачей. Данные отправляются на сервер, где проверяются на корректность, после пароль кодируется и проверяется с паролем который находится в Mongoose.

Doctor_log.hbs

```
{{> header}}
<section class="hero-wrap hero-wrap-2" style="background-image: url('/images/bg_1.jpg');"
data-stellar-background-ratio="0.5">
<div class="overlay"></div>
<div class="container">
<div class="row no-gutters slider-text align-items-center justify-content-center">
<div class="col-md-9 ftco-animate text-center">
<h1 class="mb-2 bread">Авторизация врачей</h1>
<p class="breadcrumbs"><span class="mr-2"><a href="/">Главная <i class="ion-ios-
arrow-forward"></i></a></span><span>Авторизация <i class="ion-ios-arrow-
forward"></i></span></p>
</div>
</div>
</div>
</section>
<section class="ftco-section ftco-no-pt ftco-no-pb ftco-counter img" id="section-counter"
style="background-image: url('/images/bg_3.jpg');" data-stellar-background-ratio="0.5" >
<div class="container">
<div class="row">
<div class="col-md-6 py-5 pr-md-5">
<div class="heading-section heading-section-white ftco-animate mb-5">
<span class="subheading">Вход в систему</span>
<br />
<span class="subheading">{{message}}</span>
</div>
<form action="login" class="appointment-form ftco-animate" method="POST">
<div class="d-md-flex">
<div class="form-group">
<input type="text" class="form-control" name="email" placeholder="Логин" >
</div>
<div class="form-group ml-md-4">
<input type="password" class="form-control" placeholder="Пароль" name="password">
</div>
</div>
<div class="d-md-flex">
<div class="form-group">
<input type="submit" class="form-control" name="registration" value="Войти">
</div>
</div>
</form>
</div>
<div class="col-lg-6 bg-counter p-0">

</div>
</div>
</div>
</section>
{{> footer}}
```

// В данном файле производится регистрация врачей. Доктор вводит свои данные и затем отправляет методом пост на сервер. На сервере данные проверяются на корректность, а также отсутствия данной почты в Mongoose, после пароль кодируется и загружается в Mongoose.

Doctor_reg.hbs

```
{{> header}}
<section class="hero-wrap hero-wrap-2" style="background-image: url(/images/bg_1.jpg);"
data-stellar-background-ratio="0.5">
<div class="overlay"></div>
<div class="container">
<div class="row no-gutters slider-text align-items-center justify-content-center">
<div class="col-md-9 ftco-animate text-center">
<h1 class="mb-2 bread">Регистрация пользователей</h1>
<p class="breadcrumbs"><span class="mr-2"><a href="/">Главная <i class="ion-ios-
arrow-forward"></i></a></span><span>Регистрация <i class="ion-ios-arrow-
forward"></i></span></p>
</div>
</div>
</div>
</section>
<section class="ftco-section ftco-no-pt ftco-no-pb ftco-counter img" id="section-counter"
style="background-image: url(/images/bg_3.jpg);" data-stellar-background-ratio="0.5" >
<div class="container">
<div class="row">
<div class="col-md-6 py-5 pr-md-5">
<div class="heading-section heading-section-white ftco-animate mb-5">
<span class="subheading">Регистрация</span>
<h5 class="mb-4">{{message}}</h5>
</div>
<form action="register" class="appointment-form ftco-animate" method="POST">
<div class="d-md-flex">
<div class="form-group">
<input type="text" class="form-control" placeholder="Имя" name="fname">
</div>
<div class="form-group ml-md-4">
<input type="text" class="form-control" placeholder="Фамилия" name="sname">
</div>
</div>
<div class="d-md-flex">
<div class="form-group">
<input type="text" class="form-control" placeholder="Отчество" name="tname">
</div>
<div class="form-group ml-md-4">
<div class="input-wrap">
<div class="icon"><span class="ion-md-calendar"></span></div>
<input type="text" class="form-control appointment_date" placeholder="Дата рождения"
name="date">
</div>
</div>
</div>
</div>
</div>
<div class="d-md-flex">
<div class="form-group">
```

```

<input type="text" class="form-control" placeholder="Email" name="email">
</div>
<div class="form-group ml-md-4">
<input type="password" class="form-control" placeholder="Пароль" name="password">
</div>
</div>
<div class="d-md-flex">
<div class="form-group">
<input type="submit" class="form-control" name="registration" value="Пройти
регистрацию">
</div>
</div>
</form>
</div>
<div class="col-lg-6 bg-counter aside-stretch">

</div>
</div>
</div>
</section>
{{> footer}}

```

// В данном файле реализован процесс сдачи анализов. Пользователь может зайти на эту страницу только после авторизации.

General_blood_five.hbs

```

{{> header}}
<section class="ftco-section" id="services">
  <div class="container">
    <ul class="nav">
      <li class="nav-item">
        <a class="nav-link active" href="/">Главная</a>
      </li>
      <li class="nav-item">
        <a class="nav-link " href="/analysis/{{analysis.userId}}" > Анализ крови</a>
      </li>
      <li class="nav-item">
        <a class="nav-link " href="/blood_analysis/{{analysis.userId}}">Анализ крови
(беременные)</a>
      </li>
      <li class="nav-item">
        <a class="nav-link " href="/blood_analysis_g/{{analysis.userId}}">Анализ крови (гормоны.
медиаторы)</a>
      </li>
      <li class="nav-item">
        <a class="nav-link " href="/b_blood_analysis/{{analysis.userId}}" >Биохимический анализ
крови</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" tabindex="-1" aria-disabled="true">Анализ мочи</a>
      </li>
      <li class="nav-item">

```

```

<a class="nav-link " href="/analysis_scrining/{{analysis.userId}}" >Результаты
скринингов</a>
</li>
</ul>
    <div class="fico-departments">
        <div class="row">
            <div class="col-md-12 tab-wrap">
                <div class="tab-content bg-light p-4 p-md-5 fico-animate" id="v-pills-tabContent">
                    <div class="tab-pane py-2 fade show active" id="v-pills-1" role="tabpanel" aria-
labelledby="day-1-tab">
                        <div class="row departments">
                            <div class="col-lg-4 order-lg-last d-flex align-items-stretch">
                                <div class="img d-flex align-self-stretch" style="background-
image: url(/images/dept-1.jpg);"></div>
                            </div>
                            <div class="col-lg-8">
                                <div class="row mt-5 pt-2">
                                    <div class="col-lg-12">
                                        <form action="analysis/general_blood" class="" method="POST">
                                            <div class="d-md-flex">
                                                <div class="form-group col-lg-4">
                                                    <p>Количество:<strong>{{analysis.Quantity}}</strong></p>
                                                </div>
                                                <div class="form-group col-lg-4 ml-1">
                                                    <p>Цвет:<strong>{{analysis.Color}}</strong></p>
                                                </div>
                                            <div class="form-group col-lg-4 ml-1">
                                                <p>Прозрачность:<strong>{{analysis.Transparency}}</strong></p>
                                            </div>
                                        </div>
                                        <div class="d-md-flex">
                                            <div class="form-group col-lg-4">
                                                <p>Относительная плотность:<strong>{{analysis.Relative_density}}</strong></p>
                                            </div>
                                            <div class="form-group col-lg-4 ml-1">
                                                <p>Реакция:<strong>{{analysis.Reaction}}</strong></p>
                                            </div>
                                        <div class="form-group col-lg-4 ml-1">
                                            <p>Белок:<strong>{{analysis.Protein}}</strong></p>
                                        </div>
                                        <div class="d-md-flex">
                                            <div class="form-group col-lg-4">
                                                <p>Глюкоза:<strong>{{analysis.Glucose}}</strong></p>
                                            </div>
                                            <div class="form-group col-lg-4 ml-1">
                                                <p>Билирубин:<strong>{{analysis.Bilirubin}}</strong></p>
                                            </div>
                                        <div class="form-group col-lg-4 ml-1">

```



```

<a class="nav-link " href="/blood_analysis_g/{{analysys.userId}}">Анализ крови (гармоны.
медиаторы)</a>
</li>
<li class="nav-item">
<a class="nav-link disabled" tabindex="-1" aria-disabled="true" >Биохимический анализ
крови</a>
</li>
<li class="nav-item">
<a class="nav-link" href="/Analysis_of_urine/{{analysys.userId}}">Анализ мочи</a>
</li>
<li class="nav-item">
<a class="nav-link " href="/analysys_scrining/{{analysys.userId}}" >Результаты
скринингов</a>
</li>
</ul>
    <div class="ftco-departments">
        <div class="row">
            <div class="col-md-12 tab-wrap">
                <div class="tab-content bg-light p-4 p-md-5 ftco-animate" id="v-pills-tabContent">
                    <div class="tab-pane py-2 fade show active" id="v-pills-1" role="tabpanel" aria-
labelledby="day-1-tab">
                        <div class="row departments">
                            <div class="col-lg-4 order-lg-last d-flex align-items-stretch">
                                <div class="img d-flex align-self-stretch" style="background-
image: url(/images/dept-1.jpg);"></div>
                                </div>
                                <div class="col-lg-8">
                                    <div class="row mt-5 pt-2">
                                        <div
class="col-lg-12">
                                            <form action="analysys/general_blood" class="" method="POST">
                                                <div class="d-md-flex">
                                                    <div class="form-group col-lg-4">
<p>Общий белок:<strong>{{analysys.Total_protein}}</strong></p>
                                                    </div>
                                                    <div class="form-group col-lg-4 ml-1">
<p>Мочевина:<strong>{{analysys.Urea}}</strong></p>
                                                    </div>
                                                    <div class="form-group col-lg-4 ml-1">
<p>Глюкоза:<strong>{{analysys.Glucose}}</strong></p>
                                                    </div>
                                                </div>
                                                <div class="d-md-flex">
                                                    <div class="form-group col-lg-4">
<p>АЛТ:<strong>{{analysys.ALT}}</strong></p>
                                                    </div>
                                                    <div class="form-group col-lg-4 ml-1">
<p>АСТ:<strong>{{analysys.ACT}}</strong></p>
                                                    </div>
                                                    <div class="form-group col-lg-4 ml-1">
<p>Билирубин (общий):<strong>{{analysys.Bilirubin}}</strong></p>

```



```
{{> footer}}
```

```
// В данном файле реализован процесс сдачи анализов. Пользователь может зайти на эту  
страницу только после авторизации.
```

```
General_blood_three.hbs
```

```
{{> header}}
```

```
<section class="ftco-section" id="services">
```

```
  <div class="container">
```

```
    <ul class="nav">
```

```
      <li class="nav-item">
```

```
        <a class="nav-link active" href="/">Главная</a>
```

```
      </li>
```

```
      <li class="nav-item">
```

```
        <a class="nav-link " href="/analysis/{{analysis.userId}}" > Анализ крови</a>
```

```
      </li>
```

```
      <li class="nav-item">
```

```
        <a class="nav-link " href="/blood_analysis/{{analysis.userId}}">Анализ крови  
(беременные)</a>
```

```
      </li>
```

```
      <li class="nav-item">
```

```
        <a class="nav-link disabled" tabindex="-1" aria-disabled="true">Анализ крови (гармоны.  
медиаторы)</a>
```

```
      </li>
```

```
      <li class="nav-item">
```

```
        <a class="nav-link" href="/b_blood_analysis/{{analysis.userId}}">Биохимический анализ  
крови</a>
```

```
      </li>
```

```
      <li class="nav-item">
```

```
        <a class="nav-link" href="/Analysis_of_urine/{{analysis.userId}}">Анализ мочи</a>
```

```
      </li>
```

```
      <li class="nav-item">
```

```
        <a class="nav-link " href="/analysis_scrining/{{analysis.userId}}" >Результаты  
скринингов</a>
```

```
      </li>
```

```
    </ul>
```

```
    <div class="ftco-departments">
```

```
      <div class="row">
```

```
        <div class="col-md-12 tab-wrap">
```

```
          <div class="tab-content bg-light p-4 p-md-5 ftco-animate" id="v-pills-tabContent">
```

```
            <div class="tab-pane py-2 fade show active" id="v-pills-1" role="tabpanel" aria-  
labelledby="day-1-tab">
```

```
              <div class="row departments">
```

```
                <div class="col-lg-4 order-lg-last d-flex align-items-stretch">
```

```
                  <div class="img d-flex align-self-stretch" style="background-  
image: url(/images/dept-1.jpg);"></div>
```

```
                </div>
```

```
                <div class="col-lg-8">
```

```
                  <div class="row mt-5">
```

```
                    <div class="col-lg-12">
```

```
                      <form action="analysis/general_blood" class=""
```

```
method="POST">
```

```
                    <div class="d-md-flex">
```



```

<section class="ftco-section" id="services">
  <div class="container">
    <ul class="nav">
      <li class="nav-item">
        <a class="nav-link active" href="/">Главная</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Общий анализ
        крови</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="/blood_analysis/{{analysys.userId}}">Анализ крови
        (беременные)</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="/blood_analysis_g/{{analysys.userId}}">Анализ крови (гормоны.
        медиаторы)</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="/b_blood_analysis/{{analysys.userId}}">Биохимический анализ
        крови</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="/Analysis_of_urine/{{analysys.userId}}">Анализ мочи</a>
      </li>
      <li class="nav-item">
        <a class="nav-link " href="/analysys_scrining/{{analysys.userId}}">Результаты
        скринингов</a>
      </li>
    </ul>
    <div class="ftco-departments">
      <div class="row">
        <div class="col-md-12 tab-wrap">
          <div class="tab-content bg-light p-4 p-md-5 ftco-animate" id="v-pills-tabContent">
            <div class="tab-pane py-2 fade show active" id="v-pills-1" role="tabpanel" aria-
            labelledby="day-1-tab">
              <div class="row departments">
                <div class="col-lg-4 order-lg-last d-flex align-items-stretch">
                  <div class="img d-flex align-self-stretch" style="background-
                  image: url(/images/dept-1.jpg);"></div>
                </div>
                <div class="col-lg-8">
                  <h2>Общий анализ крови</h2>
                  <p> Врачебный анализ, позволяющий оценить содержание
                  гемоглобина в системе красной крови, количество эритроцитов, цветовой показатель,
                  количество лейкоцитов, тромбоцитов. </p>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="row mt-5
    pt-2">
      <div
      class="col-lg-12">
        <form action="analysys/general_blood" class="" method="POST">
          <div class="d-md-flex">

```



```

        <div class="form-group col-lg-4">
<p>Гемоглобин:<strong>{{analysys.HGB}}</strong></p>
        </div>
        <div class="form-group col-lg-4 ml-1">
<p>Эритроциты:<strong>{{analysys.RBC}}</strong></p>
        </div>
        <div class="form-group col-lg-4 ml-1">
<p>Цветной показатель:<strong>{{analysys.CP}}</strong></p>
        </div>
        </div>
        <div class="d-md-flex">
        <div class="form-group col-lg-4">
<p>Средний объем эритроцита:<strong>{{analysys.MCV}}</strong></p>
        </div>
        <div class="form-group col-lg-4 ml-md-1">
<p>Среднее содержание Hb в эритроците:<strong>{{analysys.MCH}}</strong></p>
        </div>
        <div class="form-group col-lg-4 ml-md-1">
<p>Средняя концентрация Hb в эритроците:<strong>{{analysys.MCHC}}</strong></p>
        </div>
        </div>
        <div class="d-md-flex">
        <div class="form-group col-lg-4">
<p>Распределение эритроцитов по объему:<strong>{{analysys.RDW}}</strong></p>
        </div>
        <div class="form-group col-lg-4 ml-md-1">
<p>Тромбоциты:<strong>{{analysys.PLT}}</strong></p>
        </div>
        <div class="form-group col-lg-4 ml-md-1">
<p>Тромбокрит:<strong>{{analysys.PCT}}</strong></p>
        </div>
        </div>
        <div class="d-md-flex">
        <div class="form-group col-lg-4">
<p>Средний объем тромбоцита:<strong>{{analysys.MPV}}</strong></p>
        </div>
        <div class="form-group col-lg-4 ml-md-1">
<p>Лейкоциты:<strong>{{analysys.WBC}}</strong></p>
        </div>
        <div class="form-group col-lg-4 ml-md-1">
<p>Нейтрофилы:<strong>{{analysys.NEU1}}</strong></p>
        </div>
        </div>
        <div class="d-md-flex">
        <div class="form-group col-lg-4">
<p>Нейтрофилы (абс. кол-во):<strong>{{analysys.NEU2}}</strong></p>
        </div>
        <div class="form-group col-lg-4 ml-md-1">
<p>Эозинофилы:<strong>{{analysys.EOS1}}</strong></p>
        </div>
        <div class="form-group col-lg-4 ml-md-1">
<p>Эозинофилы (абс. кол-во):<strong>{{analysys.EOS2}}</strong></p>
        </div>

```



```

        <p class="button-custom order-lg-last mb-0"><a href="doctor/login"
class="btn btn-secondary py-2 px-5">Врач</a></p>
        <div class="collapse navbar-collapse" id="ftco-nav">
            <ul class="navbar-nav mr-auto header_menu">
                <li class="nav-item active"><a href="index.html" class="nav-link
pl-0">Главная</a></li>
                <li class="nav-item"><a href="#about" class="nav-link">О
нас</a></li>
                <li class="nav-item"><a href="/login" class="nav-
link">Анализы</a></li>
                <li class="nav-item"><a href="#contact" class="nav-
link">Контакты</a></li>
            </ul>
        </div>
    </div>
</nav>
<!-- END nav -->

<section class="home-slider owl-carousel">
<div class="slider-item" style="background-image:url(/images/bg_1.jpg);" data-stellar-
background-ratio="0.5">
    <div class="overlay"></div>
<div class="container">
<div class="row no-gutters slider-text align-items-center justify-content-start" data-scrollax-
parent="true">
<div class="col-md-6 text ftco-animate">
<h1 class="mb-4">Лечение диабета</h1>
<h3 class="subheading">Мы поможем Вам выявить диабет</h3>
<p><a href="#" class="btn btn-secondary px-4 py-3 mt-3">Подробнее </a></p>
</div>
</div>
</div>
</div>
</section>
<section class="ftco-services ftco-no-pb">
    <div class="container">
        <div class="row no-gutters">
            <div class="col-md-3 d-flex services align-self-stretch p-4 ftco-animate">
            <div class="media block-6 d-block text-center">
            <div class="icon d-flex justify-content-center align-items-center">
                <span class="flaticon-doctor"></span>
            </div>
            <div class="media-body p-2 mt-3">
            <h3 class="heading">Квалифицированные доктора</h3>
            <p>На нашем сайте зарегистрированы доктора со всего Казахстана</p>
            </div>
            </div>
            </div>
            <div class="col-md-3 d-flex services align-self-stretch p-4 ftco-animate">
            <div class="media block-6 d-block text-center">
            <div class="icon d-flex justify-content-center align-items-center">
                <span class="flaticon-ambulance"></span>
            </div>

```


<p>
Сахарный диабет – это хроническое заболевание, основным признаком которого является повышение уровня глюкозы (сахара) в крови
</p>

<div class="row mt-5 pt-2">
<div class="col-lg-6">
<div class="services-2 d-
flex">
<div class="icon mt-2
mr-3 d-flex justify-content-center align-items-center"><span class="flaticon-first-aid-
kit"></div>
<div class="text">

<h3>Диабет 1 типа</h3>

<p>В краткой
и познавательной форме вы узнаете, что такое инсулин и как он действует, можно ли
заниматься спортом и принимать алкоголь.</p>

</div>
</div>
</div>
<div class="col-lg-6">
<div class="services-2 d-
flex">

<div class="icon mt-2 mr-3 d-flex justify-content-center align-items-center"><span
class="flaticon-heart-rate"></div>

<h3>Диабет 2 типа</h3>

<p>Правда ли, что диабетом 2 типа
болеют только пожилые люди? Как поддерживать состояние своего здоровья и нужно
ли заниматься спортом?</p>

</div>
</div>
</div>
<div class="col-lg-6">
<div class="services-2 d-
flex">

<div class="icon mt-2
mr-3 d-flex justify-content-center align-items-center"><span class="flaticon-heart-
rate"></div>

<div class="text">
<h3>Heart

Rate</h3>

<p>Far far
away, behind the word mountains, far from the countries Vokalia.</p>

</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

```

</section>
<section class="ftco-intro" style="background-image: url(/images/bg_3.jpg);"
data-stellar-background-ratio="0.5">
  <div class="overlay"></div>
  <div class="container">
    <div class="row">
      <div class="col-md-9">
        <h2>Мы поможем Вам определить точный
диагноз</h2>
        <!-- <p class="mb-0">Your Health is Our Top
Priority with Comprehensive, Affordable medical.</p> -->
        <p></p>
      </div>
      <div class="col-md-3 d-flex align-items-center">
        <p class="mb-0"><a href="#section-counter"
class="btn btn-secondary px-4 py-3">Получить диагноз</a></p>
      </div>
    </div>
  </div>
</section>
<section class="ftco-section ftco-no-pt ftco-no-pb ftco-counter img" id="section-counter"
style="background-image: url(/images/bg_3.jpg);" data-stellar-background-ratio="0.5" >
  <div class="container">
    <div class="row">
      <div class="col-md-6 py-5 pr-md-5">
        <div class="heading-section heading-section-white ftco-animate mb-5">
          <span class="subheading">Консультация</span>
          <h2 class="mb-4">Регистрация</h2>
        </div>
        <form action="register" class="appointment-form ftco-animate" method="POST">
          <div class="d-md-flex">
            <div class="form-group">
              <input type="text" class="form-control"
placeholder="Фамилия" name="fname" required>
            </div>
            <div class="form-group ml-md-4">
              <input type="text" class="form-control"
placeholder="Имя" name="sname" required>
            </div>
          </div>
          <div class="d-md-flex">
            <div class="form-group">
              <input type="text" class="form-control"
placeholder="Отчество" name="tname">
            </div>
            <div class="form-group ml-md-4">
              <div class="input-wrap">
                <div class="icon"><span class="ion-md-
calendar"></span></div>
                <input type="text" class="form-control appointment_date"
placeholder="Дата рождения" name="date" required>
              </div>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>

```

```

</div>
</div>
<div class="d-md-flex">
<div class="form-group">
<input type="text" class="form-control"
placeholder="Email" name="email" required>
</div>
<div class="form-group ml-md-4">
<input type="password" class="form-control"
placeholder="Пароль" name="password" required>
</div>
</div>
<div class="d-md-flex">
<div class="form-group">
<input type="text" class="form-control"
placeholder="Лечащий доктор" name="doctor" required>
</div>
<div class="form-group ml-md-4">
<input type="text" class="form-control" placeholder="Адрес
проживания" name="address" required>
</div>
</div>
<div class="d-md-flex">
<div class="form-group">
<input type="submit" class="form-control"
name="registration" value="Пройти регистрацию">
</div>
<div class="form-group ml-
md-4 mt-3">
<a href="login"
style="color: #fff;"> У Вас уже есть аккаунт? </a>
</div>
</div>
</form>
</div>
<div class="col-lg-6 bg-counter aside-stretch">

</div>
</div>
</div>
</section>

```

```

// В данном файле описан «footer» сайта. Находится в самом конце html страницы.
<footer class="ftco-footer ftco-bg-dark ftco-section" id="contact">
<div class="container">
<div class="row mb-2">
<div class="col-md">
<div class="ftco-footer-widget mb-5">
<h2 class="ftco-heading-2 logo"><span>Diabetes</span></h2>
<p>Одно из условия выздоровления - желание выздороветь</p>
</div>
<div class="ftco-footer-widget mb-5">

```

```

    <h2 class="ftco-heading-2">Есть ли у Вас вопросы?</h2>
    <div class="block-23 mb-3">
      <ul>
        <li><a href="mailto:info@diabetes.kz"><span class="icon icon-envelope"></span><span class="text">info@diabetes.kz</span></a></li>
      </ul>
    </div>
  </div>
</div>
<div class="col-md">
  <div class="ftco-footer-widjet mb-2 ml-md-4">
    <h2 class="ftco-heading-2">Ссылки</h2>
    <ul class="list-unstyled">
      <li><a href="/"><span class="ion-ios-arrow-round-forward mr-2"></span>Главная</a></li>
      <li><a href="#about"><span class="ion-ios-arrow-round-forward mr-2"></span>О нас</a></li>
      <li><a href="/login"><span class="ion-ios-arrow-round-forward mr-2"></span>Анализы</a></li>
      <li><a href="#contact"><span class="ion-ios-arrow-round-forward mr-2"></span>Контакты</a></li>
    </ul>
  </div>
</div>
<div class="col-md">
  <div class="ftco-footer-widjet mb-2">
    <h2 class="ftco-heading-2">Анализы</h2>
    <ul class="list-unstyled">
      <li><a href="/login"><span class="ion-ios-arrow-round-forward mr-2"></span>ОБЩИЙ АНАЛИЗ КРОВИ</a></li>
      <li><a href="/login"><span class="ion-ios-arrow-round-forward mr-2"></span>БИОХИМИЧЕСКИЙ АНАЛИЗ КРОВИ (для беременных) </a></li>
      <li><a href="/login"><span class="ion-ios-arrow-round-forward mr-2"></span>АНАЛИЗ КРОВИ (СОДЕРЖАНИЕ ГОРМОНОВ И МЕДИАТОРОВ)</a></li>
      <li><a href="/login"><span class="ion-ios-arrow-round-forward mr-2"></span>АНАЛИЗ МОЧИ</a></li>
      <li><a href="/login"><span class="ion-ios-arrow-round-forward mr-2"></span>БИОХИМИЧЕСКИЙ АНАЛИЗ КРОВИ </a></li>
    </ul>
  </div>
</div>
</div>
</div>
<!-- <div class="row">
  <div class="col-md-12 text-center">
    <p>
      Copyright &copy; <script>document.write(new Date().getFullYear());</script> Все права защищены | Сделано <i class="icon-heart" aria-hidden="true"></i> в лаборатории <a href="#" target="_blank">КазНИТУ им.К.И.Сатпаева</a>
    </p>
  </div>
</div> -->
</div>

```



```

</footer>
<!-- loader -->
<div id="ftco-loader" class="show fullscreen"><svg class="circular" width="48px"
height="48px"><circle class="path-bg" cx="24" cy="24" r="22" fill="none" stroke-
width="4" stroke="#eeeeee"/><circle class="path" cx="24" cy="24" r="22" fill="none"
stroke-width="4" stroke-miterlimit="10" stroke="#F96D00"/></svg></div>
{{> footer}}
Login.hbs
{{> header}}
<section class="hero-wrap hero-wrap-2" style="background-image: url('/images/bg_1.jpg');"
data-stellar-background-ratio="0.5">
<div class="overlay"></div>
<div class="container">
<div class="row no-gutters slider-text align-items-center justify-content-center">
<div class="col-md-9 ftco-animate text-center">
<h1 class="mb-2 bread">Авторизация пользователей</h1>
<p class="breadcrumbs"><span class="mr-2"><a href="/">Главная <i class="ion-ios-
arrow-forward"></i></a></span><span> Авторизация <i class="ion-ios-arrow-
forward"></i></span></p>
</div>
</div>
</div>
</div>
</section>
<section class="ftco-section ftco-no-pt ftco-no-pb ftco-counter img" id="section-counter"
style="background-image: url(/images/bg_3.jpg);" data-stellar-background-ratio="0.5" >
<div class="container">
<div class="row">
<div class="col-md-6 py-5 pr-md-5">
<div class="heading-section heading-section-white ftco-animate mb-5">
<span class="subheading">Вход в систему</span>
<br />
<span class="subheading">{{message}}</span>
</div>
<form action="login" class="appointment-form ftco-animate" method="POST">
<div class="d-md-flex">
<div class="form-group">
<input type="text" class="form-control" name="email" placeholder="Логин" >
</div>
<div class="form-group ml-md-4">
<input type="password" class="form-control" placeholder="Пароль" name="password">
</div>
</div>
<div class="d-md-flex">
<div class="form-group">
<input type="submit" class="form-control" name="registration" value="Войти">
</div>
</div>
</form>
</div>
<div class="col-lg-6 bg-counter p-0">

</div>

```

```
</div>
</div>
</section>
{{> footer}}
```

// В данном файле производится регистрация пациентов. Пациент вводит свои данные после отправляет методом пост на сервер. На сервере данные проверяются на корректность, а также отсутствия данной почты в Mongoose. Затем пароль кодируется и загружается в Mongoose. Пациенту становятся доступны загруженные анализы на сервере, которые отправлены для проверки врачу.

Register.hbs

```
{{> header}}
<section class="hero-wrap hero-wrap-2" style="background-image: url('/images/bg_1.jpg');"
data-stellar-background-ratio="0.5">
<div class="overlay"></div>
<div class="container">
<div class="row no-gutters slider-text align-items-center justify-content-center">
<div class="col-md-9 ftco-animate text-center">
<h1 class="mb-2 bread">Регистрация пользователей</h1>
<p class="breadcrumbs"><span class="mr-2"><a href="/">Главная <i class="ion-ios-
arrow-forward"></i></a></span><span>Регистрация <i class="ion-ios-arrow-
forward"></i></span></p>
</div>
</div>
</div>
</section>
<section class="ftco-section ftco-no-pt ftco-no-pb ftco-counter img" id="section-counter"
style="background-image: url('/images/bg_3.jpg');" data-stellar-background-ratio="0.5" >
<div class="container">
<div class="row">
<div class="col-md-6 py-5 pr-md-5">
<div class="heading-section heading-section-white ftco-animate mb-5">
<span class="subheading">Регистрация</span>
<h5 class="mb-4">{{message}}</h5>
</div>
<form action="register" class="appointment-form ftco-animate" method="POST">
<div class="d-md-flex">
<div class="form-group">
<input type="text" class="form-control" placeholder="Имя" name="fname">
</div>
<div class="form-group ml-md-4">
<input type="text" class="form-control" placeholder="Фамилия" name="sname">
</div>
</div>
<div class="d-md-flex">
<div class="form-group">
<input type="text" class="form-control" placeholder="Отчество" name="tname">
</div>
<div class="form-group ml-md-4">
<div class="input-wrap">
<div class="icon"><span class="ion-md-calendar"></span></div>
```

```

<input type="text" class="form-control appointment_date" placeholder="Дата рождения"
name="date">
</div>
</div>
</div>
<div class="d-md-flex">
<div class="form-group">
<input type="text" class="form-control" placeholder="Email" name="email">
</div>
<div class="form-group ml-md-4">
<input type="password" class="form-control" placeholder="Пароль" name="password">
</div>
</div>
<div class="d-md-flex">
<div class="form-group">
<input type="text" class="form-control" placeholder="Лечащий доктор" name="doctor">
</div>
<div class="form-group ml-md-4">
<input type="text" class="form-control" placeholder="Адрес проживания" name="address">
</div>
</div>
<div class="d-md-flex">
<div class="form-group">
<input type="submit" class="form-control" name="registration" value="Пройти
регистрацию">
</div>
</div>
</form>
</div>
<div class="col-lg-6 bg-counter aside-stretch">

</div>
</div>
</div>
</section>
{{> footer}}

```

// Основной файл app.js. Данный файл является основным. Выполняемые операции:

1. Подключение всех необходимых библиотек.

```

const express = require('express')
const mongoose = require('mongoose')
const bodyParser = require('body-parser')
const passport = require('passport')
const keys = require('./config/keys')
const authRoutes = require('./routes/auth')
const path = require('path')
const app = express()
const hbs = require("hbs")
const publicDirectoryPath = path.join(__dirname, './public')
const partialDirectoryPath = path.join(__dirname, './templates/partials')
const viewsDirectoryPath = path.join(__dirname, './templates/views')

```

2. Подключение к БД.

```
mongoose.connect(keys.mongoURI, {useUnifiedTopology: true, useNewUrlParser: true,
useCreateIndex: true,})
```

```
.then(() => console.log('MongoDB connected.'))
```

```
.catch(error => console.log(error))
```

3. Основные методы express библиотеки.

```
app.use(passport.initialize())
```

```
require('./middleware/passport')(passport)
```

```
app.use(require('morgan')('dev'))
```

```
app.use(require('cors')())
```

```
app.set('view engine', 'hbs')
```

```
app.set('views', viewsDirectoryPath)
```

```
app.use(express.static(publicDirectoryPath))
```

4. Использование Hbs для разделение на блоки.

```
hbs.registerPartials(partialDirectoryPath)
```

```
//uploads/02112019-220835_518-Chrysanthemum.jpg
```

5. Загрузка данных в папку uploads.

```
app.use('/uploads', express.static('/public/uploads'))
```

6. Для парсинга данных. Без нее данные не будут загружаться на сервер.

```
//for parsing
```

```
app.use(bodyParser.urlencoded({extended: true}))
```

```
app.use(bodyParser.json())
```

7. Подключение маршрутизатора.

```
app.use('/', authRoutes)
```

8. Запуск сервера на порту 3000.

```
const port = process.env.PORT || 3000
```

```
app.listen(port, () => console.log(`Server has been started ${port}`))
```

//Файл в котором хранится все установленные библиотеки. В данном файле указывается как запускать сервер в тестовом режиме, какой файл является главным, версия сайта и устанавливаемые библиотеки.

```
package.json
```

```
{
  "name": "diabetes",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "node index",
    "server": "nodemon index",
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "Assel Mukasheva",
  "license": "ISC",
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "body-parser": "^1.19.0",
    "bootstrap": "^4.3.1",
    "cors": "^2.8.5",
    "express": "^4.17.1",
    "hbs": "^4.0.6",
    "jquery": "^3.4.1",
    "jsonwebtoken": "^8.5.1",
```

```

    "moment": "^2.24.0",
    "mongodb": "^3.3.3",
    "mongoose": "^5.7.7",
    "morgan": "^1.9.1",
    "multer": "^1.4.2",
    "passport": "^0.4.0",
    "passport-jwt": "^4.0.0"
  }
}
Package-lock.json
{
  "name": "diabetes",
  "version": "1.0.0",
  "lockfileVersion": 1,
  "requires": true,
  "dependencies": {
    "accepts": {
      "version": "1.3.7",
      "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.7.tgz",
      "integrity": "sha512-Iil88Z8nUWJjzWu/3XRyF7AZicwTq3zayV/uhVniVxpTqDz108g1Eh/Khg5VZj93S2PbUk50Jji2FaDAkA==",
      "requires": {
        "mime-types": "~2.1.24",
        "negotiator": "0.6.2"
      }
    },
    "append-field": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/append-field/-/append-field-1.0.0.tgz",
      "integrity": "sha1-HjRA6RXwsSA9I3SOeO3XubW0PIY="
    },
    "array-flatten": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-1.1.1.tgz",
      "integrity": "sha1-l9Jv83389198735923wA9Z3bk="
    },
    "basic-auth": {
      "version": "2.0.1",
      "resolved": "https://registry.npmjs.org/basic-auth/-/basic-auth-2.0.1.tgz",
      "integrity": "sha512-Ixut7LstCjYheZa7z0M0SfDkVVY+zK6yG2g3Oxv6QRuXUHqST+bHJgj2m7fPP6X4QJfF3vz6y6+KO5zaKA==",
      "requires": {
        "safe-buffer": "5.1.2"
      }
    },
    "bcryptjs": {
      "version": "2.4.3",
      "resolved": "https://registry.npmjs.org/bcryptjs/-/bcryptjs-2.4.3.tgz",
      "integrity": "sha1-lmrVe5PmBiH/fNrF2pczAn3x0Ms="
    },
    "bluebird": {
      "version": "3.5.1",
      "resolved": "https://registry.npmjs.org/bluebird/-/bluebird-3.5.1.tgz",

```

```

    "integrity": "sha512-
MKiLiV+IIAA596t9w1sQJ8jkiSr5+ZKi0WKrYGUn6dIFx+Ij4tIj+m2WMQSGczs5jZVxV339chE
8iwk6F64wjA=="
  },
  "body-parser": {
    "version": "1.19.0",
    "resolved": "https://registry.npmjs.org/body-parser/-/body-parser-1.19.0.tgz",
    "integrity": "sha512-
dhEPs72UPbDnAQJ9ZKMNTp6ptJaionhP5cBb541nXPlW60Jepo9RV/a4fX4XWW9CuFNK22kr
hrj1+rgzifNCsw=="
  },
  "requires": {
    "bytes": "3.1.0",
    "content-type": "~1.0.4",
    "debug": "2.6.9",
    "depd": "~1.1.2",
    "http-errors": "1.7.2",
    "iconv-lite": "0.4.24",
    "on-finished": "~2.3.0",
    "qs": "6.7.0",
    "raw-body": "2.4.0",
    "type-is": "~1.6.17"
  }
},
  "bootstrap": {
    "version": "4.3.1",
    "resolved": "https://registry.npmjs.org/bootstrap/-/bootstrap-4.3.1.tgz",
    "integrity": "sha512-
rXqOmH1ViAt2DyPzluTi2blhk17bO7ef+zLLPlWvG494pDxcM234pJ8wTc/6R40UWizAIIMgxjv
xZg5kmsbag=="
  },
  "bson": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/bson/-/bson-1.1.1.tgz",
    "integrity": "sha512-
jCGVYLoYMHDkOsbwJZBCqwMHYH4c+wzgi9hG7Z6SZJRXR+x58pdIbm2i9a/jFGCkRjQrU
r8eoI7lDwa0hTkxg=="
  },
  "buffer-equal-constant-time": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/buffer-equal-constant-time/-/buffer-equal-constant-
time-1.0.1.tgz",
    "integrity": "sha1-+OcRMvf/5uAaXJaXpMbz5I1cyBk="
  },
  "buffer-from": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/buffer-from/-/buffer-from-1.1.1.tgz",
    "integrity": "sha512-
MQcXEUBcKtEo7bhqEs6560Hyd4XaovZlO/k9V3hjVUF/zwW7KBVdSK4gIt/bzwS9MbR5qob+F
5jusZsb0YQK2A=="
  },
  "busboy": {
    "version": "0.2.14",
    "resolved": "https://registry.npmjs.org/busboy/-/busboy-0.2.14.tgz",

```

```

    "integrity": "sha1-bCpiLvz0fFe7vh4qnDetNseSVFM=",
    "requires": {
      "dicer": "0.2.5",
      "readable-stream": "1.1.x"
    }
  },
  "bytes": {
    "version": "3.1.0",
    "resolved": "https://registry.npmjs.org/bytes/-/bytes-3.1.0.tgz",
    "integrity": "sha512-zaULjrfCG+xvoyaQLoV8bLVXXNGC4JqlxFCutSDWA6fJrTo2ZuvLYTqZ7aHBLZSMOopbzwv8f+wZcVzfVTI2Dg==",
  },
  "commander": {
    "version": "2.20.3",
    "resolved": "https://registry.npmjs.org/commander/-/commander-2.20.3.tgz",
    "integrity": "sha512-GpVkmM8vF2vQUkj2LvZmD35JxeJOLCwJ9cUkugyk2nuhbv3+mJvpLYYt+0+USMxE+oj+ey/lJEnhZw75x/OMcQ==",
    "optional": true
  },
  "concat-stream": {
    "version": "1.6.2",
    "resolved": "https://registry.npmjs.org/concat-stream/-/concat-stream-1.6.2.tgz",
    "integrity": "sha512-22o1XU5Y066eNTp2VUN78V75GGL/0WpLXIWsX41nU4D56+qo1y4WkspIb+u6t6oq+qY1728Yz0n/bo0=",
    "requires": {
      "buffer-from": "^1.0.0",
      "inherits": "^2.0.3",
      "readable-stream": "^2.2.2",
      "typedarray": "^0.0.6"
    }
  },
  "dependencies": {
    "isarray": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/isarray/-/isarray-1.0.0.tgz",
      "integrity": "sha1-u5NdsFgsuhaMBoNJV6VKPgcSTxE="
    },
    "readable-stream": {
      "version": "2.3.6",
      "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.6.tgz",
      "integrity": "sha512-tQtKA9WIAhBF3+VLaseyMqZeBjW0AHJoxOtyqSUZnJxauErmLbVm2FWIy+J/YA9dUrAC39I TejlZWhVIwawkKw==",
      "requires": {
        "core-util-is": "~1.0.0",
        "inherits": "~2.0.3",
        "isarray": "~1.0.0",
        "process-nextick-args": "~2.0.0",
        "safe-buffer": "~5.1.1",
        "string_decoder": "~1.1.1",
        "util-deprecate": "~1.0.1"
      }
    }
  }
}

```

```

    }
  },
  "string_decoder": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.1.1.tgz",
    "integrity": "sha512-n/ShnvDi6FHbbVfviro+WojiFzv+s8MPMHBczVePfUpDJLwoLT0ht114YwBCbi8pJAveEEdnkHy
    PyTP/mzRfwg==",
    "requires": {
      "safe-buffer": "~5.1.0"
    }
  },
  "content-disposition": {
    "version": "0.5.3",
    "resolved": "https://registry.npmjs.org/content-disposition/-/content-disposition-0.5.3.tgz",
    "integrity": "sha512-ExO0774ikEOBIAEV9kDo50o+79VCUdEB6n6lzKgGwupcVeRlhrj3qGAfwq8G6uBJjkQLrhT0qE
    YFcWng8z1z0g==",
    "requires": {
      "safe-buffer": "5.1.2"
    }
  },
  "content-type": {
    "version": "1.0.4",
    "resolved": "https://registry.npmjs.org/content-type/-/content-type-1.0.4.tgz",
    "integrity": "sha512-hIP3EEPs8tB9ATIL+NUqtwOAPS4mk2Zob89MXXMHjHWg9milF/j4osnnQLXBCBFBk/tvIG/t
    Uc9mOUJiPBhPXA==",
    "requires": {}
  },
  "cookie": {
    "version": "0.4.0",
    "resolved": "https://registry.npmjs.org/cookie/-/cookie-0.4.0.tgz",
    "integrity": "sha512-Hp8fLp57wnUS0tY0tHEXh4voZRDnoIrZPqlo3DPiI4y9lwg/jqx+1Om94/W6ZaPDOUbnjOt/99
    w66zk+1IXg==",
    "requires": {}
  },
  "cookie-signature": {
    "version": "1.0.6",
    "resolved": "https://registry.npmjs.org/cookie-signature/-/cookie-signature-1.0.6.tgz",
    "integrity": "sha1-4wOogrNCzD7oylE6eZmXNNqzriw="
  },
  "core-util-is": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/core-util-is/-/core-util-is-1.0.2.tgz",
    "integrity": "sha1-tf1UIgqivFqleqtxQMLAdUUDwac="
  },
  "cors": {
    "version": "2.8.5",
    "resolved": "https://registry.npmjs.org/cors/-/cors-2.8.5.tgz",
    "integrity": "sha512-KIHbLJqu73RGr/hnbrO9uBeixNGuvSQjul/jdFvS/KFSIH1hWVdIng7zOHx+YrEjInLG7q4n6GH
    Q9cDtxv/P6g==",
    "requires": {
      "object-assign": "^4",
      "vary": "^1"
    }
  }
}

```



```

    } },
    "debug": {
      "version": "2.6.9",
      "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",
      "integrity": "sha512-
bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDjm/E7AdgFBVeAP
VMNcKGsHMA==",
      "requires": {
        "ms": "2.0.0"
      } },
    "depd": {
      "version": "1.1.2",
      "resolved": "https://registry.npmjs.org/depd/-/depd-1.1.2.tgz",
      "integrity": "sha1-m81S4UwJd2PnSbJ0xDRu0uVgtak="
    },
    "destroy": {
      "version": "1.0.4",
      "resolved": "https://registry.npmjs.org/destroy/-/destroy-1.0.4.tgz",
      "integrity": "sha1-l4hXRCxEdJ5CBmE+N5RiBYJqvYA="
    },
    "dicer": {
      "version": "0.2.5",
      "resolved": "https://registry.npmjs.org/dicer/-/dicer-0.2.5.tgz",
      "integrity": "sha1-WZbAhrsziYyBLAkL3cCc0S+stw8=",
      "requires": {
        "readable-stream": "1.1.x",
        "streamsearch": "0.1.2"
      } },
    "ecdsa-sig-formatter": {
      "version": "1.0.11",
      "resolved": "https://registry.npmjs.org/ecdsa-sig-formatter/-/ecdsa-sig-formatter-1.0.11.tgz",
      "integrity": "sha512-
nagl3RYrbNv6kQkeJIpt6NJZy8twLB/2vtz6yN9Z4vRKHN4/QZJIEbqohALSgwKdnksuY3k5Addp5
lg8sVoVcQ==",
      "requires": {
        "safe-buffer": "^5.0.1"
      } },
    "ee-first": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/ee-first/-/ee-first-1.1.1.tgz",
      "integrity": "sha1-WQxhFWsK4vTwJVcyoViyZrxWsh0="
    },
    "encodeurl": {
      "version": "1.0.2",
      "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-1.0.2.tgz",
      "integrity": "sha1-rT0yG7C0CkyL1oCw6mmBslbP1k="
    },
    "escape-html": {
      "version": "1.0.3",
      "resolved": "https://registry.npmjs.org/escape-html/-/escape-html-1.0.3.tgz",
      "integrity": "sha1-Aljq5NPQwJdN4cFpGI7wBR0dGYg="
    }
  },

```

```

"etag": {
  "version": "1.8.1",
  "resolved": "https://registry.npmjs.org/etag/-/etag-1.8.1.tgz",
  "integrity": "sha1-Qa4u62XvpiJorr/qg6x9eSmbClc="
},
"express": {
  "version": "4.17.1",
  "resolved": "https://registry.npmjs.org/express/-/express-4.17.1.tgz",
  "integrity": "sha512-mHJ9O79RqluphRrcw2X/GTh3k9tVv8YcoyY4Kkh4WDMUYKRZUq0h1o0w2rrrxBqM7VoeUVq
gb27xlEMXTnYt4g==",
  "requires": {
    "accepts": "~1.3.7",
    "array-flatten": "1.1.1",
    "body-parser": "1.19.0",
    "content-disposition": "0.5.3",
    "content-type": "~1.0.4",
    "cookie": "0.4.0",
    "cookie-signature": "1.0.6",
    "debug": "2.6.9",
    "depd": "~1.1.2",
    "encodeurl": "~1.0.2",
    "escape-html": "~1.0.3",
    "etag": "~1.8.1",
    "finalhandler": "~1.1.2",
    "fresh": "0.5.2",
    "merge-descriptors": "1.0.1",
    "methods": "~1.1.2",
    "on-finished": "~2.3.0",
    "parseurl": "~1.3.3",
    "path-to-regexp": "0.1.7",
    "proxy-addr": "~2.0.5",
    "qs": "6.7.0",
    "range-parser": "~1.2.1",
    "safe-buffer": "5.1.2",
    "send": "0.17.1",
    "serve-static": "1.14.1",
    "setprototypeof": "1.1.1",
    "statuses": "~1.5.0",
    "type-is": "~1.6.18",
    "utils-merge": "1.0.1",
    "vary": "~1.1.2"
  }
},
"finalhandler": {
  "version": "1.1.2",
  "resolved": "https://registry.npmjs.org/finalhandler/-/finalhandler-1.1.2.tgz",
  "integrity": "sha512-aAWcW57uxVNrQZqFXjITpW3sIUQmHGG3qSb9mUah9MgMC4NeWhNOINjXEYq3HjRAvL6a
rUviZGGJsBg6z0zsWA==",
  "requires": {
    "debug": "2.6.9",
    "encodeurl": "~1.0.2",

```

```

    "escape-html": "~1.0.3",
    "on-finished": "~2.3.0",
    "parseurl": "~1.3.3",
    "statuses": "~1.5.0",
    "unpipe": "~1.0.0"
  } },
  "foreachasync": {
    "version": "3.0.0",
    "resolved": "https://registry.npmjs.org/foreachasync/-/foreachasync-3.0.0.tgz",
    "integrity": "sha1-VQKYfchxS+M5IJzLgBxyd7gfPY="
  },
  "forwarded": {
    "version": "0.1.2",
    "resolved": "https://registry.npmjs.org/forwarded/-/forwarded-0.1.2.tgz",
    "integrity": "sha1-mM19qxFlZXuMBXPozszZGw/xjIQ="
  },
  "fresh": {
    "version": "0.5.2",
    "resolved": "https://registry.npmjs.org/fresh/-/fresh-0.5.2.tgz",
    "integrity": "sha1-PYyt2Q2XZWn6glqx+OSyOhBWBac="
  },
  "handlebars": {
    "version": "4.3.5",
    "resolved": "https://registry.npmjs.org/handlebars/-/handlebars-4.3.5.tgz",
    "integrity": "sha512-1I6T/18X9DV3sEkY9sK9lsPRgDsj82ayBY/4pAZyP2BcX5WeRM3O06bw9kIs2GLrHvFB/DNzW
    WJyFvof8wQGqw==",
    "requires": {
      "neo-async": "^2.6.0",
      "optimist": "^0.6.1",
      "source-map": "^0.6.1",
      "uglify-js": "^3.1.4"
    } },
  "hbs": {
    "version": "4.0.6",
    "resolved": "https://registry.npmjs.org/hbs/-/hbs-4.0.6.tgz",
    "integrity": "sha512-
    KFt3Y4zOvVQOp84TmqVaFTpBTYO1sVenBoBY712MI3vPkKxVoO6AsuEyDayIRPRAHRYZHH
    Wnmc4spFa8fhQpLw==",
    "requires": {
      "handlebars": "4.3.5",
      "walk": "2.3.14"
    } },
  "http-errors": {
    "version": "1.7.2",
    "resolved": "https://registry.npmjs.org/http-errors/-/http-errors-1.7.2.tgz",
    "integrity": "sha512-
    uUQBt3H/cSIVfch6i1EuPNy/YsRSOUBXTVfZ+yR7Zjez3qjBz6i9+i4zjNaoqcoFVI4lQJ5plg63Tv
    GfRSDCRg==",
    "requires": {
      "depd": "~1.1.2",
      "inherits": "2.0.3",

```

```

    "setprototypeof": "1.1.1",
    "statuses": ">= 1.5.0 < 2",
    "toidentifier": "1.0.0"
  } },
  "iconv-lite": {
    "version": "0.4.24",
    "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.4.24.tgz",
    "integrity": "sha512-v3MXnZAcvnywkTUEZomIActle7RXXeedOR3Iwwl7VlyoXO4Qi9arvSenNQWne1TcRwhCL1Hw
LI21bEqdpj8/rA==",
    "requires": {
      "safer-buffer": ">= 2.1.2 < 3"
    } },
  "inherits": {
    "version": "2.0.3",
    "resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.3.tgz",
    "integrity": "sha1-YzwsG+PaQqUC9SRmAiSA9CCCYd4="
  },
  "ipaddr.js": {
    "version": "1.9.0",
    "resolved": "https://registry.npmjs.org/ipaddr.js/-/ipaddr.js-1.9.0.tgz",
    "integrity": "sha512-36+AdBzCL+y6qjw5Tx7HgzeGCzC81MDDgaUP8ld2zhx58HdqXGoBd+tHdrBMiyjGQs0Hxs/M
LZTu/eHNJJuWPw==",
  },
  "isarray": {
    "version": "0.0.1",
    "resolved": "https://registry.npmjs.org/isarray/-/isarray-0.0.1.tgz",
    "integrity": "sha1-ihis/Kmo9Bd+Cav8YDiTmwXR7t8="
  },
  "jquery": {
    "version": "3.4.1",
    "resolved": "https://registry.npmjs.org/jquery/-/jquery-3.4.1.tgz",
    "integrity": "sha512-36+AdBzCL+y6qjw5Tx7HgzeGCzC81MDDgaUP8ld2zhx58HdqXGoBd+tHdrBMiyjGQs0Hxs/M
LZTu/eHNJJuWPw==",
  },
  "jsonwebtoken": {
    "version": "8.5.1",
    "resolved": "https://registry.npmjs.org/jsonwebtoken/-/jsonwebtoken-8.5.1.tgz",
    "integrity": "sha512-XjwVfRS6jTMsqYs0EsuJ4LGxXV14zQybNd4L2r0UvbVnSF9Af8x7p5MzbJ90Ioz/9TI41/hTCvznF
/loiSzn8w==",
    "requires": {
      "jws": "^3.2.2",
      "lodash.includes": "^4.3.0",
      "lodash.isboolean": "^3.0.3",
      "lodash.isinteger": "^4.0.4",
      "lodash.isnumber": "^3.0.3",
      "lodash.isplainobject": "^4.0.6",
      "lodash.isstring": "^4.0.1",
      "lodash.once": "^4.0.0",
    }
  }

```

```

    "ms": "^2.1.1",
    "semver": "^5.6.0"
  },
  "dependencies": {
    "ms": {
      "version": "2.1.2",
      "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.2.tgz",
      "integrity": "sha512-sGkPx+VjMtmA6MX27oA4FBFELFCZZ4S4XqeGOXCv68tT+jb3vk/RyaKWP0PTKyWtmLSM0b+adUTEvbs1PEaH2w=="
    }
  },
  "jwa": {
    "version": "1.4.1",
    "resolved": "https://registry.npmjs.org/jwa/-/jwa-1.4.1.tgz",
    "integrity": "sha512-sqLW1D1S0aQXQ2f0oJwYbTUAMf1e3gmdE8QAGb3r1cDk1dusx0Zbkp18Tz2ZD41d1WlyhEHYiKN39VtV3aA=="
  },
  "requires": {
    "buffer-equal-constant-time": "1.0.1",
    "ecdsa-sig-formatter": "1.0.11",
    "safe-buffer": "^5.0.1"
  }
},
  "jws": {
    "version": "3.2.2",
    "resolved": "https://registry.npmjs.org/jws/-/jws-3.2.2.tgz",
    "integrity": "sha512-12015863890j1413a4i051187905d4c304a4c10a697291491991571c81241e2b"
  },
  "requires": {
    "jwa": "^1.4.1",
    "safe-buffer": "^5.0.1"
  }
},
  "kareem": {
    "version": "2.3.1",
    "resolved": "https://registry.npmjs.org/kareem/-/kareem-2.3.1.tgz",
    "integrity": "sha512-lh1oR3814l26xjrsZ0NKHt1j1PqZd121ngaxSg+Iz1f0S7diUdNv3WNIIVS4s9p6tMQiJ6soQm1nUXvke0A=="
  },
  "lodash.includes": {
    "version": "4.3.0",
    "resolved": "https://registry.npmjs.org/lodash.includes/-/lodash.includes-4.3.0.tgz",
    "integrity": "sha1-YLuYqHy5I8aMoeUTJUgzFISfVT8="
  },
  "lodash.isboolean": {
    "version": "3.0.3",
    "resolved": "https://registry.npmjs.org/lodash.isboolean/-/lodash.isboolean-3.0.3.tgz",
    "integrity": "sha1-bC4XHbKiV82WgC/UOwGyDV9YcPY="
  },
  "lodash.isinteger": {
    "version": "4.0.4",
    "resolved": "https://registry.npmjs.org/lodash.isinteger/-/lodash.isinteger-4.0.4.tgz",

```

```

    "integrity": "sha1-YZwK89A/iwTDH1iChAt3sRzWg0M="
  },
  "lodash.isnumber": {
    "version": "3.0.3",
    "resolved": "https://registry.npmjs.org/lodash.isnumber/-/lodash.isnumber-3.0.3.tgz",
    "integrity": "sha1-POdoEMWSjQM1IwGsKHMx8RwLH/w="
  },
  "lodash.isplainobject": {
    "version": "4.0.6",
    "resolved": "https://registry.npmjs.org/lodash.isplainobject/-/lodash.isplainobject-4.0.6.tgz",
    "integrity": "sha1-fFJqUtibRcRcxpC4gWO+BJfIUMs="
  },
  "lodash.isstring": {
    "version": "4.0.1",
    "resolved": "https://registry.npmjs.org/lodash.isstring/-/lodash.isstring-4.0.1.tgz",
    "integrity": "sha1-ISfftUVuynzJu5XV2ur4i6VKVFE="
  },
  "lodash.once": {
    "version": "4.1.1",
    "resolved": "https://registry.npmjs.org/lodash.once/-/lodash.once-4.1.1.tgz",
    "integrity": "sha1-DdOXEhPHxW34gJd9UEyI+0cal6w="
  },
  "media-typer": {
    "version": "0.3.0",
    "resolved": "https://registry.npmjs.org/media-typer/-/media-typer-0.3.0.tgz",
    "integrity": "sha1-hxDXrwqmJvj/+hgzAWhUUmMlV0g="
  },
  "memory-pager": {
    "version": "1.5.0",
    "resolved": "https://registry.npmjs.org/memory-pager/-/memory-pager-1.5.0.tgz",
    "integrity": "sha512-ZS4Bp4r/Zoeq6+NLJpP+0Zzm0pR8whtGPf1XExKLJBaczGMnSi3It14OiNCStjQjM6NU1okjQG  
SxgEZN8eBYKg==",
    "optional": true
  },
  "merge-descriptors": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/merge-descriptors/-/merge-descriptors-1.0.1.tgz",
    "integrity": "sha1-sAqqVW3YtEVoFQ7J0blT8/kMu2E="
  },
  "methods": {
    "version": "1.1.2",
    "resolved": "https://registry.npmjs.org/methods/-/methods-1.1.2.tgz",
    "integrity": "sha1-VSmkInZUE07cxSZmVoNbD4Ua/O4="
  },
  "mime": {
    "version": "1.6.0",
    "resolved": "https://registry.npmjs.org/mime/-/mime-1.6.0.tgz",
    "integrity": "sha512-x0Vn8spI+wuJ1O6S7gnbaQg8Pxx4NNHb7KSINmEWKiPE4RKOpIvijn+NkmYmmRgP68mc70j  
2EbeTFRsrswaQeg=="
  },

```

```

"mime-db": {
  "version": "1.40.0",
  "resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.40.0.tgz",
  "integrity": "sha512-
jYdeOMPy9vnxEqFRRo6ZvTZ8d9oPb+k18PKoYNYUe2stVEBPPwsln/qWzdbmaIvnhZ9v2P+Cu
ecK+fpUfsV2mA=="
},
"mime-types": {
  "version": "2.1.24",
  "resolved": "https://registry.npmjs.org/mime-types/-/mime-types-2.1.24.tgz",
  "integrity": "sha512-
WaFHS3MC15fapm3oLxU4eYDw77IQM2ACcxQ9RIxfaC3ooc6PFuBMGZZsYpvoXS5D5QTWPI
eol1jjLdAm3TBP3cQ==",
  "requires": {
    "mime-db": "1.40.0"
  }
},
"minimist": {
  "version": "0.0.8",
  "resolved": "https://registry.npmjs.org/minimist/-/minimist-0.0.8.tgz",
  "integrity": "sha1-hX/Kv8M5fSYluCKCYuhqp6ARsF0="
},
"mkdirp": {
  "version": "0.5.1",
  "resolved": "https://registry.npmjs.org/mkdirp/-/mkdirp-0.5.1.tgz",
  "integrity": "sha1-MAV0OOrGz3+MR2fzhkjWaX11yQM=",
  "requires": {
    "minimist": "0.0.8"
  }
},
"moment": {
  "version": "2.24.0",
  "resolved": "https://registry.npmjs.org/moment/-/moment-2.24.0.tgz",
  "integrity": "sha512-
bV7f+6l2QigeBBZSM/6yTNq4P2fNpSWj/0e7jQcy87A8e7o2nAfP/34/2ky5Vw4B9S446EtlhodAzk
FCcR4dQg=="
},
"mongodb": {
  "version": "3.3.3",
  "resolved": "https://registry.npmjs.org/mongodb/-/mongodb-3.3.3.tgz",
  "integrity": "sha512-
MdRnoOjstmnrKJsK8PY0PjP6fyF/SBS4R8coxmhsfEU7tQ46/J6j+aSHF2n4c2/H8B+Hc/Klbf8v
ggZfI0mA=="
  "requires": {
    "bson": "^1.1.1",
    "require_optional": "^1.0.1",
    "safe-buffer": "^5.1.2",
    "sasprep": "^1.0.0"
  }
},
"mongoose": {
  "version": "5.7.7",
  "resolved": "https://registry.npmjs.org/mongoose/-/mongoose-5.7.7.tgz",

```

```

    "integrity": "sha512-
FU59waB4LKBa9KOnqBUcCcMIVRc09TFo1F8nMxrzSiIWATaJpjjxSSH5FBVUDxQfNdJLfg9
uFHxaTxhhwjsZOQ==",
    "requires": {
      "bson": "~1.1.1",
      "kareem": "2.3.1",
      "mongodb": "3.3.3",
      "mongoose-legacy-pluralize": "1.0.2",
      "mpath": "0.6.0",
      "mquery": "3.2.2",
      "ms": "2.1.2",
      "regexp-clone": "1.0.0",
      "safe-buffer": "5.1.2",
      "sift": "7.0.1",
      "sliced": "1.0.1"
    },
    "dependencies": {
      "ms": {
        "version": "2.1.2",
        "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.2.tgz",
        "integrity": "sha512-
sGkPx+VjMtmA6MX27oA44FBFELFCZZ4S4XqeGOXCv68tT+jb3vk/RyaKWP0PTKyWtmLSM0b
+adUTEvbs1PEaH2w=="
      }
    },
    "mongoose-legacy-pluralize": {
      "version": "1.0.2",
      "resolved": "https://registry.npmjs.org/mongoose-legacy-pluralize/-/mongoose-legacy-
pluralize-1.0.2.tgz",
      "integrity": "sha512-
Yo/7qQU4/EyIS8YDFSeenIvXxZN+ld7YdV9LqFVQJzTLye8unujAWPZ4NWKjFA+RNjh+wwTW
KY9Z3E5XM6ZZiQ=="
    },
    "morgan": {
      "version": "1.9.1",
      "resolved": "https://registry.npmjs.org/morgan/-/morgan-1.9.1.tgz",
      "integrity": "sha512-
HQStPIV4y3afTiCYVxirakhlCfGkI161c76kKFca7Fk1JusM//Qeo1ej2XaMniiNeaZklMVRh3vTtIzP
zwbpmA=="
    },
    "requires": {
      "basic-auth": "~2.0.0",
      "debug": "2.6.9",
      "depd": "~1.1.2",
      "on-finished": "~2.3.0",
      "on-headers": "~1.0.1"
    }
  },
  "mpath": {
    "version": "0.6.0",
    "resolved": "https://registry.npmjs.org/mpath/-/mpath-0.6.0.tgz",
    "integrity": "sha512-
i75qh79MJ5Xo/sbhxrDrPSEG0H/mr1kcZXJ8dH6URU5jD/knFxCVqVC/gVSW7GIXL/9hHWIT9
haLbCXWOLL3qw=="
  },
}

```



```

    "mquery": {
      "version": "3.2.2",
      "resolved": "https://registry.npmjs.org/mquery/-/mquery-3.2.2.tgz",
      "integrity": "sha512-
XB52992COp0KP230I3qloVUbKLUxJlu328HBP2t2EsxSFtf4W1HPSOBWOXf1bqxK4Xbb66lfM
J+Bpfd9/yZE1Q==",
      "requires": {
        "bluebird": "3.5.1",
        "debug": "3.1.0",
        "regexp-clone": "^1.0.0",
        "safe-buffer": "5.1.2",
        "sliced": "1.0.1"
      }
    },
    "dependencies": {
      "debug": {
        "version": "3.1.0",
        "resolved": "https://registry.npmjs.org/debug/-/debug-3.1.0.tgz",
        "integrity": "sha512-
OX8XqP7/1a9cqkxYw2yXss15f26NKWBpDXQd0/uK/KPqdQhxbPa994hnzjcE2VqQpDslf55723c
KPUOGSmMY3g==",
        "requires": {
          "ms": "2.0.0"
        }
      }
    },
    "ms": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",
      "integrity": "sha1-VgiurfwAvmwpAd9fmGF4jeDVI8g="
    },
    "multer": {
      "version": "1.4.2",
      "resolved": "https://registry.npmjs.org/multer/-/multer-1.4.2.tgz",
      "integrity": "sha512-
xY8pX7V+byUpbYMxtjM9KAiD9ixtg5/JkeKUTD6xilfDv0vzzOFcCp4Ljb1UU3tSOM3VTZtKo6
3OmzOrGi3Cg==",
      "requires": {
        "append-field": "^1.0.0",
        "busboy": "^0.2.11",
        "concat-stream": "^1.5.2",
        "mkdirp": "^0.5.1",
        "object-assign": "^4.1.1",
        "on-finished": "^2.3.0",
        "type-is": "^1.6.4",
        "xtend": "^4.0.0"
      }
    },
    "negotiator": {
      "version": "0.6.2",
      "resolved": "https://registry.npmjs.org/negotiator/-/negotiator-0.6.2.tgz",
      "integrity": "sha512-
hZXc7K2e+PgeI1eDBe/10Ard4ekbfrrqG8Ep+8Jmf4JID2bNg7NvCPOZN+kfF574pFQI7mum2A
UqDidoKqcTOw=="
    },
    "neo-async": {

```

```

    "version": "2.6.1",
    "resolved": "https://registry.npmjs.org/neo-async/-/neo-async-2.6.1.tgz",
    "integrity": "sha512-iyam8fBuCUpWeKPGpaNMetEocMt364qkCsfL9JuhjXX6dRnguRVOfk2GZaDpPjcOKiiXCPINZ
    C1GczQ7iTq3Zw=="
  },
  "object-assign": {
    "version": "4.1.1",
    "resolved": "https://registry.npmjs.org/object-assign/-/object-assign-4.1.1.tgz",
    "integrity": "sha1-IQmtx5ZYh8/AXLvUQsrIv7s2CGM="
  },
  "on-finished": {
    "version": "2.3.0",
    "resolved": "https://registry.npmjs.org/on-finished/-/on-finished-2.3.0.tgz",
    "integrity": "sha1-IPEzZIGwg811M3mSoWlxqi2QaUc=",
    "requires": {
      "ee-first": "1.1.1"
    }
  },
  "on-headers": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/on-headers/-/on-headers-1.0.2.tgz",
    "integrity": "sha512-iyam8fBuCUpWeKPGpaNMetEocMt364qkCsfL9JuhjXX6dRnguRVOfk2GZaDpPjcOKiiXCPINZ
    pZAE+FJLoyITytdqK0U5s+FIpjN0JP3OzFi/u8Rx+EV5/W+JTWGXG8xFzevE7AjBfDqHv/8vL8
    qQsIhHnqRkrA=="
  },
  "optimist": {
    "version": "0.6.1",
    "resolved": "https://registry.npmjs.org/optimist/-/optimist-0.6.1.tgz",
    "integrity": "sha1-2j6nRob6IaGaERwybpDrFaAZZoY=",
    "requires": {
      "minimist": "~0.0.1",
      "wordwrap": "~0.0.2"
    }
  },
  "parseurl": {
    "version": "1.3.3",
    "resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.3.tgz",
    "integrity": "sha512-CiyeOxFT/JZyN5m0z9PfXw4SCBJ6Sygz1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG1fcj3fHynXi9
    NYO4nWOL+qQ=="
  },
  "passport": {
    "version": "0.4.0",
    "resolved": "https://registry.npmjs.org/passport/-/passport-0.4.0.tgz",
    "integrity": "sha1-xQlWkTR71a07XhgCOMORTRbwWBE=",
    "requires": {
      "passport-strategy": "1.x.x",
      "pause": "0.0.1"
    }
  },
  "passport-jwt": {
    "version": "4.0.0",
    "resolved": "https://registry.npmjs.org/passport-jwt/-/passport-jwt-4.0.0.tgz",

```

```

    "integrity": "sha512-
BwC0n2GP/1hMVjR4QpnvqA61TxenUmlmfNjYNgK0ZAs0HK4SOQkHcSv4L328blNTLtHq7Db
mvyNjH+bn6C5Mg==",
    "requires": {
      "jsonwebtoken": "^8.2.0",
      "passport-strategy": "^1.0.0"
    },
    "passport-strategy": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/passport-strategy/-/passport-strategy-1.0.0.tgz",
      "integrity": "sha1-tVOaqPwiWj0a0XlHbd8ja0QPuUQ="
    },
    "path-to-regexp": {
      "version": "0.1.7",
      "resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-regexp-0.1.7.tgz",
      "integrity": "sha1-32BBEABfUi8V60SQ5yR6G/qmf4w="
    },
    "pause": {
      "version": "0.0.1",
      "resolved": "https://registry.npmjs.org/pause/-/pause-0.0.1.tgz",
      "integrity": "sha1-HUCLP9t2kjuVQ9lvtMnfITXZy10="
    },
    "process-nextick-args": {
      "version": "2.0.1",
      "resolved": "https://registry.npmjs.org/process-nextick-args/-/process-nextick-args-
2.0.1.tgz",
      "integrity": "sha512-
3ouUOpQhtgrbOa17J7+uxOTpITYWaGP7/AhoR3+A+/1e9skrzelGi/dXzEYyvxubEF6Wn2ypsc
TKiKJFFnlag=="
    },
    "proxy-addr": {
      "version": "2.0.5",
      "resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-2.0.5.tgz",
      "integrity": "sha512-
t/7RxHXP6cJtP0pRG6smSr9QJidhB+3kXu0KgXnbGYMgzEnUxRQ4/LDdfOwZEMyIh3/xHb8
PX3t+lfL9z+YVQ==",
      "requires": {
        "forwarded": "~0.1.2",
        "ipaddr.js": "1.9.0"
      }
    },
    "qs": {
      "version": "6.7.0",
      "resolved": "https://registry.npmjs.org/qs/-/qs-6.7.0.tgz",
      "integrity": "sha512-
VCdBRNFTX1fyE7Nb6FYoURo/SPe62QCaAyzJvUjwRaIsc+NePBEhiHlvxFmmX56+HZphIGtV
0XeCirBtpDrTyQ=="
    },
    "range-parser": {
      "version": "1.2.1",
      "resolved": "https://registry.npmjs.org/range-parser/-/range-parser-1.2.1.tgz",

```

```

    "integrity": "sha512-
Hrgsx+orqoygnmhFbKaHE6c296J+HTAQXoxEF6gNupROmmGJRozzfG3ccAveqCBwr/2yxQ5
BVd/GTl5agOwSg=="
  },
  "raw-body": {
    "version": "2.4.0",
    "resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.4.0.tgz",
    "integrity": "sha512-
4Oz8DUlwdvoa5qMJelxipi/iJii40O5cGV1wNYp5hvZP8ZN0T+jiNkL0QepXs+EsQ9XJ8ipEDoi
H70ySUJP3Q==",
    "requires": {
      "bytes": "3.1.0",
      "http-errors": "1.7.2",
      "iconv-lite": "0.4.24",
      "unpipe": "1.0.0"
    }
  },
  "readable-stream": {
    "version": "1.1.14",
    "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-1.1.14.tgz",
    "integrity": "sha1-fPTFvZI44EwhMY23SB54WbAgdk=",
    "requires": {
      "core-util-is": "~1.0.0",
      "inherits": "~2.0.1",
      "isarray": "0.0.1",
      "string_decoder": "~0.10.x"
    }
  },
  "regexp-clone": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/regexp-clone/-/regexp-clone-1.0.0.tgz",
    "integrity": "sha512-
TuAasHQNamyyJ2hb97IuBEif4qBHGjPHBS64sZwytpLEqtBQ1gPJTnOaQ6qmpET16cK14kkjba
zl6+p0RRv0yw=="
  },
  "require_optional": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/require_optional/-/require_optional-1.0.1.tgz",
    "integrity": "sha512-
qhM/y57enGWHAE3v/NcwML6a3/vfESLe/sGM2dII+gEO0BpKRUKWZow/tyloNqJyN6kXSl3Ryy
M8Ll5D/sJP8g==",
    "requires": {
      "resolve-from": "^2.0.0",
      "semver": "^5.1.0"
    }
  },
  "resolve-from": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/resolve-from/-/resolve-from-2.0.0.tgz",
    "integrity": "sha1-lICrIOLP+h2egKgeX+oUdhGwa1c="
  },
  "safe-buffer": {
    "version": "5.1.2",
    "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",

```

```

    "integrity": "sha512-
Gd2UZBJDkXlY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpScSmMJSO
F8LOIrt8u
d/wPtojys4G6+g=="
  },
  "safer-buffer": {
    "version": "2.1.2",
    "resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-2.1.2.tgz",
    "integrity": "sha512-
YZo3K82SD7Riyi0E1EQPojLz7kpepnSQI9IyPbHHg1XXXevb5dJI7tpyN2ADxGcQbHG7vcyRHk
0cbwqcQriUtg=="
  },
  "saslp̄rep": {
    "version": "1.0.3",
    "resolved": "https://registry.npmjs.org/saslp̄rep/-/saslp̄rep-1.0.3.tgz",
    "integrity": "sha512-
/MY/PEMbk2SuY5sScONwhUDsV2p77Znkb/q3nSVstq/yQzYJOH/Azh29p9oJLsl3LnQwSvZDKag
DGBsBwSooag==",
    "optional": true,
    "requires": {
      "sparse-bitfield": "^3.0.3"
    }
  },
  "semver": {
    "version": "5.7.1",
    "resolved": "https://registry.npmjs.org/semver/-/semver-5.7.1.tgz",
    "integrity": "sha512-
sauaDf/PZdVgrLTNYHRtpXa1iRiKcaebiKQ1BJdpQlWH2lCvexQdX55snPFyK7Qzpu
dqbcI0qX
FfOasHdyNDGQ=="
  },
  "send": {
    "version": "0.17.1",
    "resolved": "https://registry.npmjs.org/send/-/send-0.17.1.tgz",
    "integrity": "sha512-
BsVKsiGcQMFwT8UxypobUKyv7irCNRHk1T0G680vk88yf6LBBByGcZJOTJCrTP2xVN6yI+XjP
JcNuE3V4fT9sAg==",
    "requires": {
      "debug": "2.6.9",
      "depd": "~1.1.2",
      "destroy": "~1.0.4",
      "encodeurl": "~1.0.2",
      "escape-html": "~1.0.3",
      "etag": "~1.8.1",
      "fresh": "0.5.2",
      "http-errors": "~1.7.2",
      "mime": "1.6.0",
      "ms": "2.1.1",
      "on-finished": "~2.3.0",
      "range-parser": "~1.2.1",
      "statuses": "~1.5.0"
    }
  },
  "dependencies": {
    "ms": {
      "version": "2.1.1",

```

```

    "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.1.tgz",
    "integrity": "sha512-
tgp+dl5cGk28utYktBsrFqA7HKgrhgPsg6Z/EfhWI4gl1Hwq8B/GmY/0oXZ6nF8hDVesS/FpnYaD/
kOWhYQvyg=="
  } } },
  "serve-static": {
    "version": "1.14.1",
    "resolved": "https://registry.npmjs.org/serve-static/-/serve-static-1.14.1.tgz",
    "integrity": "sha512-
JMrvUwE54emCYWITI+hGrGv5I8dEwmco/00EvkzIIsR7MqrHonbD9pO2MOjFnpFnt17ecpZs+
3mW+XbQZu9QCg==",
    "requires": {
      "encodeurl": "~1.0.2",
      "escape-html": "~1.0.3",
      "parseurl": "~1.3.3",
      "send": "0.17.1"
    } },
  "setprototypeof": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.1.1.tgz",
    "integrity": "sha512-
JvdAWfbXeIGaZ9cILp38HntZSFSo3mWg6xGcJJsd+d4aRMOqauag1C63dJfDw7OaMYwEbHM
OxEZ1lqVRYP2OAw=="
  },
  "sift": {
    "version": "7.0.1",
    "resolved": "https://registry.npmjs.org/sift/-/sift-7.0.1.tgz",
    "integrity": "sha512-
oqD7PMJ+uO6jV9EQCl0LrRw1OwsiPsiFQR5AR30heR+4Dl7jBBbDLnNvWiak20tzZlSE1H7R
B30SX/1j/YYT7g=="
  },
  "sliced": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/sliced/-/sliced-1.0.1.tgz",
    "integrity": "sha1-CzpmK10Ewxd7GSa+qCsD+Dei70E="
  },
  "source-map": {
    "version": "0.6.1",
    "resolved": "https://registry.npmjs.org/source-map/-/source-map-0.6.1.tgz",
    "integrity": "sha512-
UjgapumWlbMhkBgzT7Ykc5YXUT46F0iKu8SGXq0bcwP5dz/h0Plj6enJqjz1Zbq2l5WaqYnrVbw
WOWMyF3F47g=="
  },
  "sparse-bitfield": {
    "version": "3.0.3",
    "resolved": "https://registry.npmjs.org/sparse-bitfield/-/sparse-bitfield-3.0.3.tgz",
    "integrity": "sha1-/0rm5oZWBWuks+eSgzM004JzyhE=",
    "optional": true,
    "requires": {
      "memory-pager": "^1.0.2"
    } },
  "statuses": {

```

```

    "version": "1.5.0",
    "resolved": "https://registry.npmjs.org/statuses/-/statuses-1.5.0.tgz",
    "integrity": "sha1-Fhx9rBd2Wf2YEfQ3cfqZOBR4Yow="
  },
  "streamsearch": {
    "version": "0.1.2",
    "resolved": "https://registry.npmjs.org/streamsearch/-/streamsearch-0.1.2.tgz",
    "integrity": "sha1-gludDlb8Jz2Am6VzOOkpkZoanxo="
  },
  "string_decoder": {
    "version": "0.10.31",
    "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-0.10.31.tgz",
    "integrity": "sha1-YuIDvEF2bGwoyfyEMB2rHFMQ+pQ="
  },
  "toidentifier": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.0.tgz",
    "integrity": "sha512-yaOH/Pk/VEhBWWTlhI+qXxDFXlejDGcQipMlyxda9nthulaxLZUNcUqFxoKp0vcYnvteJln5FNQDRrxj3YcbVw=="
  },
  "type-is": {
    "version": "1.6.18",
    "resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18.tgz",
    "integrity": "sha512-TkRKR9sUTxEH8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCnlFPW4ky9Q+iA+m a9BGm06XQBy8g==",
    "requires": {
      "media-typer": "0.3.0",
      "mime-types": "~2.1.24"
    }
  },
  "typedarray": {
    "version": "0.0.6",
    "resolved": "https://registry.npmjs.org/typedarray/-/typedarray-0.0.6.tgz",
    "integrity": "sha1-hnrHTjkhGHsdPUfZlqeOxciDB3c="
  },
  "uglify-js": {
    "version": "3.6.7",
    "resolved": "https://registry.npmjs.org/uglify-js/-/uglify-js-3.6.7.tgz",
    "integrity": "sha512-4sXQDzmdnoXiO+xvmTzQsfIiwrijUCSA95rSP4SEd8tDb51W2TiDOIL76Hl+Kw0Ie42PSItCW8/t6pBNCF2R48A==",
    "optional": true,
    "requires": {
      "commander": "~2.20.3",
      "source-map": "~0.6.1"
    }
  },
  "unpipe": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",
    "integrity": "sha1-sr9O6FFKrmFltIF4KdIbLvSZBOw="
  }

```

```

    },
    "util-deprecate": {
      "version": "1.0.2",
      "resolved": "https://registry.npmjs.org/util-deprecate/-/util-deprecate-1.0.2.tgz",
      "integrity": "sha1-RQ1Nyfpw3nMnYvvS1KKJgUGaDM8="
    },
    "utils-merge": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/utils-merge/-/utils-merge-1.0.1.tgz",
      "integrity": "sha1-n5VxDiCiZ5R7LMwSR0HBAoQn5xM="
    },
    "vary": {
      "version": "1.1.2",
      "resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",
      "integrity": "sha1-IpnwLG3tMNSlIhsLn3RSShj2NPw="
    },
    "walk": {
      "version": "2.3.14",
      "resolved": "https://registry.npmjs.org/walk/-/walk-2.3.14.tgz",
      "integrity": "sha512-5skcWAUmySj6hkBdH6B6+3ddMjVQYH5Qy9QGbPmN8kVmLteXk+yVXg+yfkInbX30EYakahLrr8iPcCxJQSCBeg==",
      "requires": {
        "foreachasync": "^3.0.0"
      }
    },
    "wordwrap": {
      "version": "0.0.3",
      "resolved": "https://registry.npmjs.org/wordwrap/-/wordwrap-0.0.3.tgz",
      "integrity": "sha1-o9XabNXAvAAI03I0u68b7WMFkQc="
    },
    "xtend": {
      "version": "4.0.2",
      "resolved": "https://registry.npmjs.org/xtend/-/xtend-4.0.2.tgz",
      "integrity": "sha512-LKYU1iAXJXUGAXn9URjiu+MWhyUXHsvfp7mcuYm9dSUKK0/CjtrUwFAXD82/mCWbtLsGjFIad0wIsod4zrTAEQ=="
    }
  }
}

```


ПРИЛОЖЕНИЕ Д

Акт внедрения

"ҚОЖА АХМЕТ ЯСАУИ атындағы
ХАЛЫҚАРАЛЫҚ ҚАЗАҚ-ТҮРІК
УНИВЕРСИТЕТІНІҢ
КЛИНИКА-ДИАГНОСТИКАЛЫҚ ОРТАЛЫҒЫ"
МЕКЕМЕСІ

THE CLINIC-DIAGNOSTIC CENTRE
OF THE A.YASAWI
INTERNATIONAL
KAZAKH-TURKISH UNIVERSITY



HOCA AHMET YESEVI
ULUSLARARASI TÜRK-KAZAK
ÜNİVERSİTESİ
TIP FAKÜLTESİ HASTANESİ

УЧРЕЖДЕНИЕ
"КЛИНИКО-ДИАГНОСТИЧЕСКИЙ ЦЕНТР
МЕЖДУНАРОДНОГО
КАЗАХСКО-ТУРЕЦКОГО
УНИВЕРСИТЕТА имени Х.А.ЯСАВИ"

Мекен-жайы: Қазақстан, Түркістан облысы, Түркістан қаласы, көшесі Б.Сатпаұханов, үй 29, пошталық индекс 161200
Тел: 8(72533) 6-39-07,6-38-35 Факс: 8(72533) 6-35-63. www.ayclinic.kz e-mail:ayclinic@mail.ru

№ 473

16 . 10 20 19

АКТ

внедрения научных результатов диссертационной работы PhD докторанта Казахского национального исследовательского технического университета имени К. И. Сатпаева Мукашевой Асель Коптлеувны на тему «Исследование и разработка информационной системы диагностики сахарного диабета на базе инструментов BigData технологий», представленную на соискание ученой степени доктора PhD по специальности 6D070300 –«Информационные системы»

Мы, нижеподписавшиеся, комиссия в составе: Главного врача Клиники С. Рустемовой; заместителя главного врача по методическому обеспечению и организационным делам Л.Кенжетовой; врача общей практики А. Бабажанова; врача-эндокринолога С.Каримовой составили настоящий акт о том, что результаты научно-исследовательской работы диссертационной работы PhD докторанта НАО «Казахского национального исследовательского технического университета имени К.И.Сатпаева» Мукашевой А.К. по теме «Исследование и разработка информационной системы диагностики сахарного диабета на базе инструментов BigData технологий» на соискание ученой степени доктора PhD по специальности 6D070300 - «Информационные системы» используются в работе клиники в виде информационной системы поддержки врача-эндокринолога по диагностике сахарного диабета.

Разработанная информационная система позволила на основе алгоритмического и программного обеспечения по диагностике сахарного диабета повысить качество диагностирования болезни. Использование в системе методов регрессионного анализа для прогнозирования роста пациентов позволяют спрогнозировать количество возможных пациентов, которые дают возможность заранее запланировать количество закупаемого инсулина и других средств контроля диабета в Туркестанской области.

Разработанная система по диагностике сахарного диабета испытана в условиях реального времени.

Научные результаты диссертационной работы являются актуальными и представляют практический интерес. Разработанные и исследованные алгоритмы и полученные результаты диссертационной работы применяются для автоматизации процесса диагностирования болезни и прогнозирования роста пациентов.

Испытания информационной системы показали правильность выбранного принципа работы системы и его работоспособности в реальных условиях при правильном выборе платформы - как услуги информационной системы, а также размещением в среде системы инструментов BigData технологий.

Председатель комиссии:
Главный врач КДЦ
МКТУ имени Х.А. Ясави
Туркестан, Казахстан



С. Рустимова

Члены комиссии:

Кенжетаева Л.
Бабажанов А.
Каримова С.

ПРИЛОЖЕНИЕ Е
Авторское свидетельство

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ  РЕСПУБЛИКА КАЗАХСТАН

СВИДЕТЕЛЬСТВО
О ВНЕСЕНИИ СВЕДЕНИЙ В ГОСУДАРСТВЕННЫЙ РЕЕСТР
ПРАВ НА ОБЪЕКТЫ, ОХРАНЯЕМЫЕ АВТОРСКИМ ПРАВОМ
№ 9730 от «11» мая 2020 года

Фамилия, имя, отчество, (если оно указано в документе, удостоверяющем личность) автора (ов):
МУКАШЕВА АСЕЛЬ КОПТЛЕУВНА, САПАРХОДЖАЕВ НУРБЕК ПАЖАРБЕКОВИЧ

Вид объекта авторского права: программа для ЭВМ

Название объекта: Информационная система поддержки врача эндокринолога «Diabetes.kz»

Дата создания объекта: 01.05.2019





Кредитная ссылка: <http://www.kazpatent.kz/m/saitiny-n>
"Авторские ссылки" разделе: <https://copyright.kazpatent.kz>

Подлинность документа возможно проверить на сайте [kazpatent.kz](http://www.kazpatent.kz)
в разделе «Авторское право» <https://copyright.kazpatent.kz>

Подписано ЭЦП Куантыров Е.С.