

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

Құрманғалиұлы Санжар

Разработка WEB-приложения для врачей-эндокринологов
«Оценка скрининга больных диабетом»

ДИПЛОМНАЯ РАБОТА

Специальность 5В070300 – Информационные системы

Алматы 2021

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт кибернетики и информационных технологий Кафедра

кибербезопасность, обработка и хранение информации

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой КОиХИ

канд. техн. наук



Н. А. Сейлова

«31» 05 2021г.

ДИПЛОМНАЯ РАБОТА

На тему: “ Разработка WEB-приложения для врачей-эндокринологов

«Оценка скрининга больных диабетом»”

Специальность 5В070300 – Информационные системы

Выполнил: Құрманғалиұлы Санжар

Научный руководитель

канд. техн. наук, ассоц. проф.



Жумағалиев Б. И.

«27» мая 2021г.

Алматы 2021

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт кибернетики и информационных технологий Кафедра

кибербезопасность, обработка и хранение информации

5B070300 – Информационные системы

УТВЕРЖДАЮ

Заведующий кафедрой КОиХИ
канд. техн. наук

 Н. А. Сейлова

«_31_» __05__ 2021_г.

ЗАДАНИЕ

на выполнение дипломной работы

Обучающемуся: Құрманғалиұлы Санжар

Тема: Разработка WEB-приложения для врачей-эндокринологов «Оценка скрининга больных диабетом»»

Утверждена приказом Ректора Университета № 762-б от 24.11.2020г.

Срок сдачи законченной работы «_27_» _мая_ 2021 г.

Исходные данные к дипломной работе: результаты преддипломной практики, обзор предметной области, сбор теоретического материала.

Краткое содержание дипломной работы:

- а) Обзор технологий скрининг больных диабетом;
- б) Выбор средств разработки системы;
- в) Проектирование и разработка приложения.

Рекомендуемая основная литература: 12 наименований

ГРАФИК

подготовки дипломной работы (проекта)

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Обзор решений и технологий электронного обучения	18.01.2021 – 12.02.2021	
Выбор средств проектирования и разработки системы	15.02.2021 – 01.03.2021	
Проектирование и разработка системы	02.03.2021 – 29.03.2021	

Подписи

консультантов и нормоконтролера на законченную дипломную работу (проект) с указанием относящихся к ним разделов работы (проекта)

Наименование разделов	Консультанты, Ф.И.О. (уч. степень, звание)	Дата подписания	Подпись
“ Разработка WEB-приложения для врачей-эндокринологов «Оценка скрининга больных диабетом»”	Жумагалиев Б. И., кандидат технических наук, ассоциированный профессор	28.05.2021	
Нормоконтролер	Кабдулин А., тьютор		
Программная часть	Кабдуллин М. А., ассистент		

Научный руководитель:  Жумагалиев Б. И.

Задание принял к исполнению обучающийся  Күрманғалиұлы Санжар.

Дата " 24 " 11 2020 г.

Аннотация

В данной дипломной работе рассматривается разработка мобильного веб приложения для информационной поддержки больных сахарным диабетом, включая оценку текущего состояния пациента. Мобильное веб-приложение относится к приложениям поддержки пациента и врача. Приложение позволяет оказывать помощь в фиксации самоконтроля уровня сахара в крови с использованием смартфона, который имеет выход в интернет и, поэтому обеспечивает оперативную связь с врачом. В работе представлены результаты разработки мобильного веб приложения, осуществляющего автоматизированный сбор, анализ и обработку данных о течении болезни-диабета, представление данных лечащему врачу и медицинскому учреждению. Представлена структурная схема, интерфейс и структура электронных записей разработанной информационной системы. Система предназначена для практического использования.

Annotation

This thesis discusses the development of a mobile web application for information support for patients with diabetes, including the assessment of the current state of the patient. Mobile web application refers to the patient and doctor support applications. The application allows you to assist in fixing self-monitoring of blood sugar levels using a smartphone that has Internet access and, therefore, provides prompt communication with the doctor. The paper presents the results of the development of a mobile web application that performs automated collection, analysis and processing of data on the course of the disease-diabetes, presenting data to the attending physician and medical institution. The block diagram, interface and structure of electronic records of the developed information system are presented. The system is designed for practical use.

Аннотация

Бұл тезисте қант диабетімен ауыратын науқастарды ақпараттық қолдауға арналған мобильді веб-қосымшаның дамуын, оның ішінде науқастың қазіргі жағдайын бағалау қарастырылған. Мобильді веб-қосымша пациентті және дәрігерді қолдау қосымшаларына жатады. Қосымша интернетке шығуға мүмкіндігі бар смартфонды пайдалана отырып, қандағы қант деңгейін өзін-өзі бақылауды бекітуге көмек көрсетуге мүмкіндік береді, сондықтан дәрігермен жедел байланысты қамтамасыз етеді. Жұмыста аурудың-диабеттің барысы туралы деректерді автоматтандырылған жинауды, талдауды және өңдеуді жүзеге асыратын мобильді веб-қосымшаны әзірлеу нәтижелерін, деректерді емдеуші дәрігерге және медициналық мекемеге ұсыну ұсынылған. Әзірленген ақпараттық жүйенің электрондық жазбаларының құрылымдық схемасы, интерфейсі және құрылымы ұсынылған. Жүйе практикалық қолдануға арналған.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	8
1 ОБЗОР РЕШЕНИЙ И ТЕХНОЛОГИЙ БОЛЬНЫХ ДИАБЕТОМ.....	9
1.1 Анализ существующих приложений для скрининга.....	9
1.2 Преимущества и недостатки электронного скрининга.....	10
1.3 Постановка задач дипломной работы.....	11
2 ВЫБОР СРЕДСТВ РАЗРАБОТКИ СИСТЕМЫ.....	12
2.1 Архитектура веб-приложения.....	12
2.2 Выбор системы управлениями базой данных.....	16
2.3 Интерфейс пользователя работы с информационной системой.....	20
2.4 Средства разработки веб-приложения.....	22
3 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРИЛОЖЕНИИ.....	26
3.1 Моделирование базы данных.....	24
3.2 Безопасность веб-приложения.....	26
3.3 Разработка компонентов серверной части.....	29
3.4 Разработка компонентов клиентской части.....	32
3.5 Серверное программное обеспечение.....	34
3.6 Тестирование веб-приложения.....	35
ЗАКЛЮЧЕНИЕ.....	41
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	42
ПРИЛОЖЕНИЕ А	
ПРИЛОЖЕНИЕ Б	
ПРИЛОЖЕНИЕ В	

ВВЕДЕНИЕ

Целью данной дипломной работы является разработать веб-приложения для врачей-эндокринологов «Оценка скрининга больных диабетом», которое обеспечивает связь между врачом и пациентом для сбора информации сахара в крови. Разработка приложений разделилась на серверную часть и клиентскую. Серверная часть была реализовано с помощью языка программирование SWIFT, так же облачное вычисление Firebase backend, база данных была использована технология noSQL на облачном сервере от гугл firebase и была использована технология realtime. Клиентская часть было реализовано с средой разработкой xCode, а так же классы main.storyboard, AppDelegate.swift, SceneDelegate.swift, AutchViewControl.swift, , ViewController.swift.

Проектирование информационной системы было сделано с применением ER и UML диаграмм.

Одной из сфер, на которую повлияло информационная технология, является медицина. С учетом специфики в медицинском учреждение вводятся учетные записи, амбулаторная книжка, данные о пациенте, адрес пациента и т.д. ИТ технология успешно интегрирована в медицинскую деятельность так как упрощает и избавляется от старого формата с бумагами, и тасканием амбулаторной карты. Данное время достаточно прийти в прописанный вами поликлинику и показать ваше удостоверение личности со штрих кодом.

Развивать медицине помогает направление в компьютерные вычисления, но в нашем приложении используется облачное вычисления так же и хранилище которые считаются актуальными и по сей день, разработка веб - приложений. Веб-приложение diaMed - это прикладная программа, хранящаяся на удаленном сервере.

1. ОБЗОР РЕШЕНИЙ И ТЕХНОЛОГИЙ БОЛЬНЫХ ДИАБЕТОМ

1.1 Анализ существующих приложений для скрининга

Скрининг-эти исследования являются одним из основных принципов раннего выявления заболеваний сахарного диабета. Поскольку это исследование является методом, используемым в ряде стран в качестве одного из основных способов оценки ранней стадии диабета. Одной из целей скринингового исследования является выявление факторов риска возникновения социально значимых заболеваний с последующим динамическим наблюдением, улучшением и снижением первичной заболеваемости населения.

В Казахстане этот метод практикуется с 2008 года-для выявления заболеваний сердечно-сосудистой системы.

Анализ существующих приложений в Казахстане имеется. В настоящее время имеется аналог веб-приложений по названию *damumed*. Что представляет нам это приложение. В данном веб приложении имеется где вводятся данные с глюкометра. Но только данные для мед карточки, то есть оно будет храниться в локальном базе для вас, вы можете изменять а так же смотреть как запись, последний раз когда было и сколько было уровень глюкозы в крови, эти данные не обрабатывается врачом.

Разработка веб-приложений для медицины является большим приоритетом, так как в медицине накануне идет жизнь человека. Приложение всегда было и будет в спросе так как, 90% население Казахстана используют мобильные смартфоны с выходом в интернет. Веб приложение автоматизирует, а также упрощает работу медицинским врачам для определения пред сахарным диабетом, то есть заранее понять, что у человека уровень глюкозы повышена с расчетом массы тело и роста человека.

В рамках исследование рынка проведен текущего состояния в предоставленной области и создано всеохватывающее заключение для амбулаторного исследования болезненных сахарным диабетом.

В реальное время в государствах Европы и USA развиваются мобильные приложения системы для контроля ЭКГ, артериального давления, режима питания и телесных нагрузок. Свежим считается назначение в области разработки мобильных приложений систем для исследования людей больных сахарным диабетом. Первыми шагами в разработке мобильных веб-приложений стали показавшиеся в связи с развитием мобильных приборов приложения для самоконтроля диабета. Они предполагают собой независимые мобильные приложения для сбора и анализа данных о диабете для контроля и анализа именно прикованным к своим кроватям сладким диабетом. Главная задача данных приложений - гарантировать подмену классическому картонному дневнику исследований и привнести составляющие классификации в процесс самоконтроля диабета. Итоги ряда изучений зарекомендовали, собственно что использование аналогичных приложений разрешает добиться снижения значения гликированного гемоглобина в длительной возможности.

В настоящее время есть 2 более известные системы для телефонов: iOS и Android. Были рассмотрены надлежащие приложения, дешевые сквозь GooglePlay: Ontrack; Diabetesbox; Glucol Lite; Glucose Buddy; Sugar Log; Diabetes Records; dbees; Diabetes Log Book; Diabetes Tracker; Diabetes Health; SiDiary; DiabControl; Diabetes risk. Грядущим рубежом в направленности разработки мобильных веб-приложения систем стала разработка приложений, встроенных с централизованными серверами для сбора информации.

1.2 Преимущества и недостатки электронного скрининга

Преимущества нашего приложения в том что сейчас каждый 3 имеет смартфоны на базе IOS. По статистике операционная система ios стоит на 2 позиций после андройда. И тем самым для скрининга больных, передачи данных между пациентом и врачом будет осуществлена с помощью нашего приложения diaMed. Приложение будет установлена на мобильные устройства пациента, и будет сбор информации о глюкозы в крови каждый день. Приложение удобно с тем, что не придется идти в больницу стоять в очередях для сдачи крови, для анализа. То есть приложение буквально занимает 5 минут вашего времени, данные с глюкометра пишется в приложение с периодом до еды. Вам придется 1 раз зарегистрироваться в diaMed, и сессия будет сохранена, и не нужно будет вводить пароль и логин при каждой сдаче анализа. Врачи будут через веб-сайт смотреть ваши данные и рассчитывать для норм глюкозы в крови. Если при повышении глюкозы и показатели начинают превышать, то с помощью нашего приложения придется уведомление о записи к врачу с датой и временем.

Недостатки нашего электронного скрининга сахарного диабета в том, что, не полный анализ, то есть вам все равно при повышении ехать в мед учреждение вашего врача для полного анализа, и назначение лечение.

1.3 Постановка задач дипломной работы

Цель дипломной работы – разработка веб-приложение, которая обеспечит дальнейшее развитие электронного скрининга больных диабетом человека и связь лечащим врачом.

Задачи, которые необходимо решить в данной работе:

- анализ современных архитектурных решений в веб-сфере;
- выбор системы управления базами данных (СУБД);
- современные инструменты разработки и тестирования программного обеспечения для исследований.
- разработка веб-приложения в соответствии с общепринятыми стандартами и методиками.
- проектирование информационной системы (ИС), в которой будет работать веб-приложение;

Разрабатываемые требования к программному обеспечению:

- Наличие клиентской части, которая отправляет запросы на сервер;
- Наличие серверной части, позволяющей передавать данные из базы данных, создавать и изменять их по запросу клиента;
- Должно быть кроссбраузерное веб-приложение и адаптивная верстка для корректного вывода страниц на устройство для врачей, и приложение на IOS;
- База данных должна быть создана с использованием сопоставления не реляционных объектов для облегчения связи между таблицами;
- Возможность частого хранения необходимых данных, что позволит повысить производительность информационной системы.

Веб-приложение должно предлагать такие услуги, как:

- Аутентификация и авторизация, которые являются наиболее важными функциями сервера, обеспечивающими полную безопасность приложения путем защиты данных и процессов от несанкционированного доступа злоумышленников;
- просмотр профиль пациента больного диабетом;
- календарь, ввод данных с глюкометра;
- связь между пациентом и врачом;
- рекомендаций при повышенных показаниях от 6,5 % и больше.
- выбор тип диабета: 1 тип до 30 лет, 2 тип от 30 лет

2 ВЫБОР СРЕДСТВ РАЗРАБОТКИ СИСТЕМЫ

2.1 Архитектура веб-приложения

Архитектура нашего IOS приложение строится на MVVM-R и используем MVVM-R (MVVM + Router). Так как наше веб-приложение сделано на среде разработки xCode для операционных систем IOS(Iphone).

Это обычный MVVM, в котором навигация между экранами размещается в отдельном уровне - Router, а логика получения данных размещается в сервисах. Далее рассмотрим наши лучшие практики реализации каждого уровня.

MVVM - это паттерн разработки, позволяющий разделить приложение на три функциональные части:

Model - основная логика программы (работа с данными, вычисления, запросы и так далее).

View - вид или представление (пользовательский интерфейс).

ViewModel - модель представления, которая служит прослойкой между View и Model.

Такое разделение позволяет ускорить разработку и поддерживаемость программы - можно менять один компонент, не затрагивая код другого.

Архитектура IOS

Платформа - это каталог, содержащий динамическую общую библиотеку и связанные с ней ресурсы: заголовочные файлы, изображения и т. д. Платформы - это точка доступа к различным системным интерфейсам, таким как адресная книга iOS, фотоленка устройства и медиатека.

Слой CocoaTouch

Cocoa Touch - это фреймворк для создания приложений под iPhone, iPod touch, и iPad.

CocoaTouch имеет большие функциональные перспективы, в частности, в том, как приложения полагаются на слой CocoaTouch и какова их роль в архитектуре iOS.

Слой CocoaTouch является верхним слоем в архитектуре IOS. У него есть несколько основных платформ, на которых основаны собственные приложения IOS, и одной из них является платформа UIKit. Уровень CocoaTouch определяет инфраструктуру для базовых приложений и предоставляет ряд жизненно важных технологий, таких как многозадачность или ввод датчиков.

Как мы уже отмечали, приложения iOS тесно основаны на платформе UIKit.

Платформа UIKit предоставляет необходимую инфраструктуру для ваших приложений IOS или tvOS. Он предоставляет архитектуру окон и представлений для реализации вашего интерфейса, инфраструктуру обработки событий для доставки Multi-Touch и других типов ввода в ваше приложение, а также основной цикл выполнения, необходимый для управления взаимодействиями между пользователем, системой и вашим приложением. Другие функции, предлагаемые платформой, включают поддержку анимации, поддержку документов, поддержку рисования и печати, информацию о текущем устройстве, управление текстом и отображением, поддержку поиска, поддержку специальных возможностей, поддержку расширений приложений и управление ресурсами.

UIKit это библиотека для подключение к ViewController.swift. Нельзя запускать собственные приложения iOS, если они не связаны с UIKit и Foundation.

Платформа UIKit тесно связана с iOS. Существует среда разработки, эквивалентная OS X, которая позволяет использовать набор приложений или среду разработки AppKit. Среда разработки UIKit предоставляет инфраструктуру для графических приложений iOS на основе событий. Однако он также работает с другими важными аспектами платформы iOS (такими как многозадачность, оповещения и инструменты доступа).

Слой CocoaTouch предоставляет разработчикам большое количество функций высокого уровня, таких как автоматический режим шаблонов, печать, функции распознавания жестов и поддержка документов. В дополнение к платформе UIKit, она также имеет платформы MARKIT, EventKit и интерфейс сообщений.

Носитель слоя

Графика, аудио и видео управляются медиауровнем. Этот уровень состоит из нескольких ключевых технологий, включая CoreGraphics, OpenGL ES, OpenAL, Av Foundation и CoreMedia. Медиауровень состоит из множества платформ, включая библиотеку ресурсов для доступа к фотографиям и видео для конкретных устройств, базовое изображение для обработки изображений с использованием фильтров и базовую графику для создания двумерных графических элементов. Если вам нужен полный список платформ на уровне мультимедиа, мы рекомендуем вам обратиться к руководству по обзору технологий iOS.

Слой CoreServices

Уровень CoreServices отвечает за управление основными системными службами, использующими собственные приложения iOS. Слой CocoaTouch тесно связан со слоем базовых служб в некоторых аспектах функциональности. Уровень CoreServices также предоставляет множество невосполнимых свойств, таких как объекты блоков, Grand Central Dispatch, In-AppPurchase и iCloudStorage.

Одно из основных дополнений в слое CoreServices называется ARC или автоматический подсчет ссылок. Что такое ARC? Это функция уровня компилятора, которая упрощает процесс выделения памяти в Objective-C.

Платформа Foundation-еще одна важная среда разработки для приложений iOS и OS X. Платформа foundation - это больше, чем просто набор полезных глав, таких как NSArray, NSDictionary и NSDate. Он предлагает корневой класс NSObject, который обеспечивает основной интерфейс для Objective-C, а также предлагает различные модели, такие как политики объектов свойств. В CoreFoundation корпоративная среда разработки позволяет многим библиотекам и средам разработки обмениваться данными и кодом.

Другой средой разработки базового уровня, которая также тесно связана с базовым уровнем, является среда разработки базового уровня на основе C. Как и на предприятии, он позволяет различным библиотекам и средам разработки разделять код и данные. Базовый фундамент имеет свойство, часто называемое свободным мостом, которое позволяет взаимно заменять объекты cocoa базовыми объектами фундамента и наоборот.

Слой Core OS

На рисунке 2.1 изображена схема - MVVM нашего разрабатываемой веб приложения diaMed, в приложении мы использовали Фреймворк Cocoa Touch с используемой библиотекой UIKit

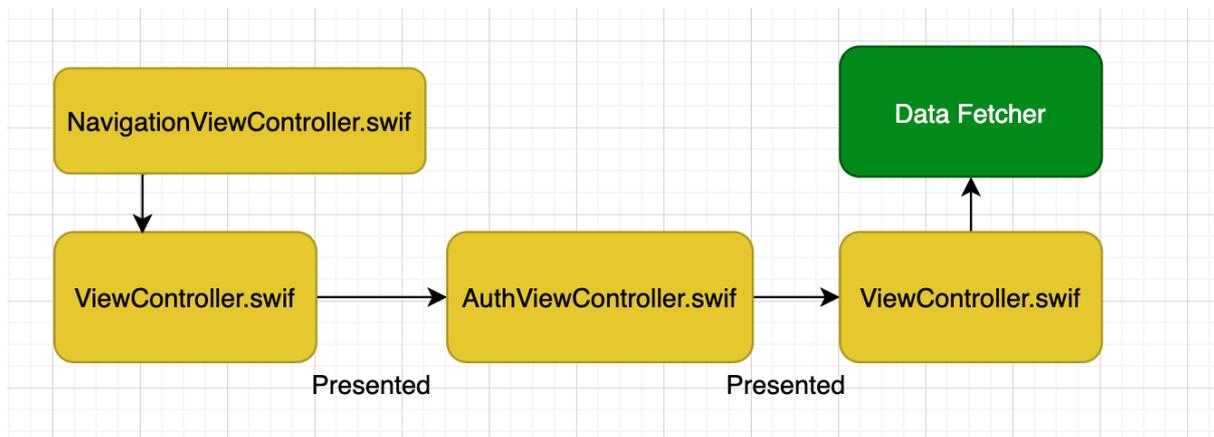


Рисунок 2.1 – MVVM - схема

На рисунке 2.2 изображена архитектура нашего приложение diaMed. Визуализация нашего приложение.

В проект создается файл main.storyboard и загрузка происходит из него весь интерфейс. Описана сцена внутри этого файла.



Рисунок 2.2

На рис. 2.2 эта стрелка указывает на первый экран, который будет виден при запуске программы. Он автоматически добавляется после добавления первой консоли на рабочую панель, а затем вы можете перетащить его на любой другой контроллер, сделав его первым.



Рисунок 2.3

На рис. 2.3 эта стрелка описывает отношения между контроллерами, в моем нашем случае навигационный контроллер AuthViewController соответствует первой консоли ViewController. Это соединение может быть настроено таким же образом, как и соединения и действия сокетов.



Рисунок 2.4

На рис. 2.4 эта стрелка показывает переход к другому контроллеру. Такой переход можно легко создать таким же образом, и, как и в предыдущих случаях, начало соединения может идти непосредственно от нужного элемента интерфейса - кнопок, ячеек таблицы и т.д. Он имеет настройки-идентификатор и тип перехода. В нашем случае кнопка профиля превращается в ProfileViewController. swift.

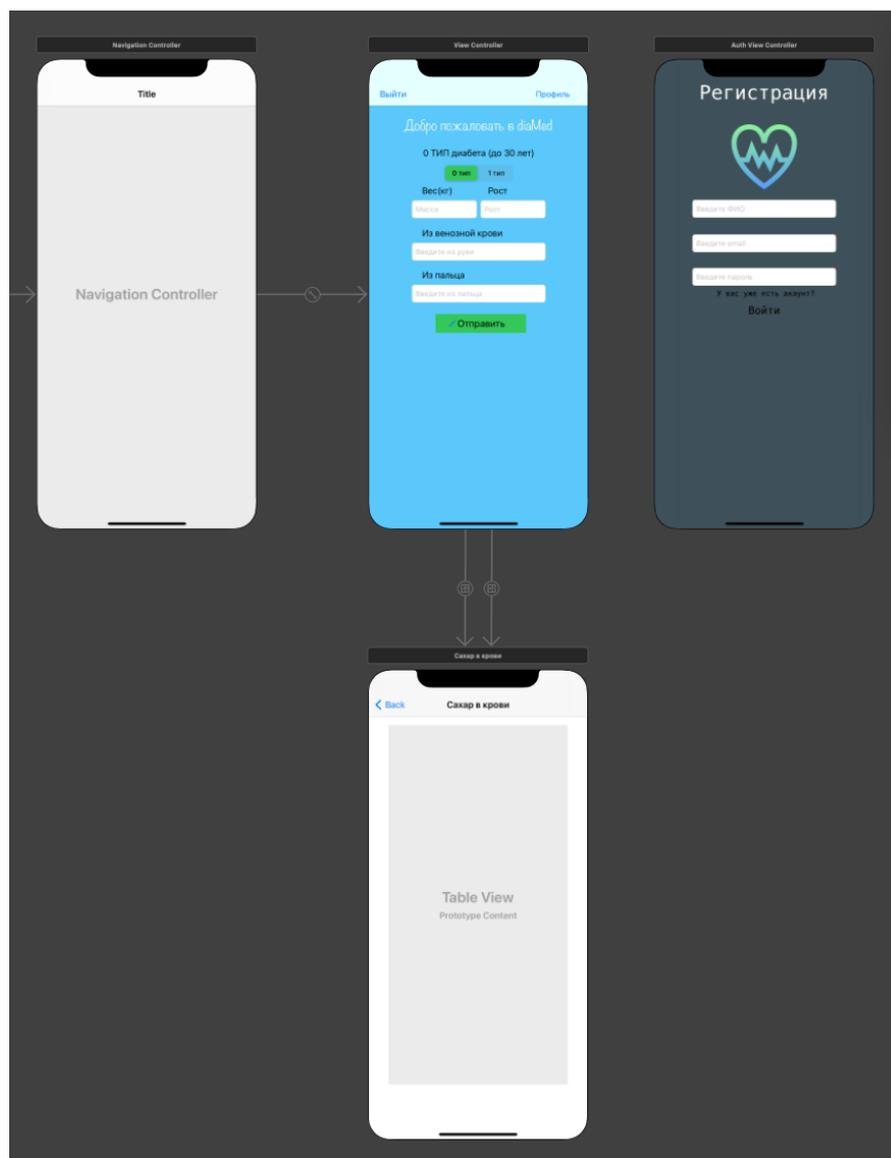


Рисунок 2.5 – main.Storyboard.

На рис. 2.5 внешний вид Storyboard(раскадровки) представляет собой удобный механизм разработки приложений. Этот механизм позволяет значительно сократить объем кода, связанного с переходами между экранами.

2.1 Выбор системы управлениями базой данных

Выбор системы для нашего приложения впало на сервис от гугл Firebase.

Преимущества и недостатки сервиса Firebase.

Для начала определим, что наша база данных имеет структуру NoSQL, хотя это имеет преимущества с точки зрения масштабирования, оно имеет недостатки, если вы управляете большими объемами высокоструктурированных данных. С обычной реляционной базой данных вы можете писать мощные запросы на SQL, чтобы быстро получить данные, которые вы ищете, а в Firebase нет эквивалента. Кроме того, вы можете использовать ORM[2.1] для дальнейшего упрощения обработки данных.

Firebase, безусловно, отлично подходит для выполнения, выполнения и эксплуатации. Firebase действует как база данных, которая изменяется в режиме реального времени и хранит данные в JSON. Любые изменения в базе данных в одну и ту же секунду синхронизируются между всеми клиентами или устройствами, использующими одну и ту же базу данных. Другими словами, обновления в Firebase происходят мгновенно. Единая база, которая будет передавать гаджетам на операционных системах, как и IOS так и для Android в реальном времени. То есть пациент вводит свои данные и отправляет в Realtime Database в ту же секунду лечащий врач может увидеть, не обновляя страницу.

Наряду с хранилищем Firebase также обеспечивает аутентификацию пользователей, поэтому все данные передаются по защищенному SSL-соединению[2.2]. Мы можем выбрать любую комбинацию электронной почты и пароля для аутентификации, будь то Facebook, Twitter, GitHub, Google или что-то еще. Но в нашем случае мы выбрали регистрацию пациента с использованием электронной почты и пароля.

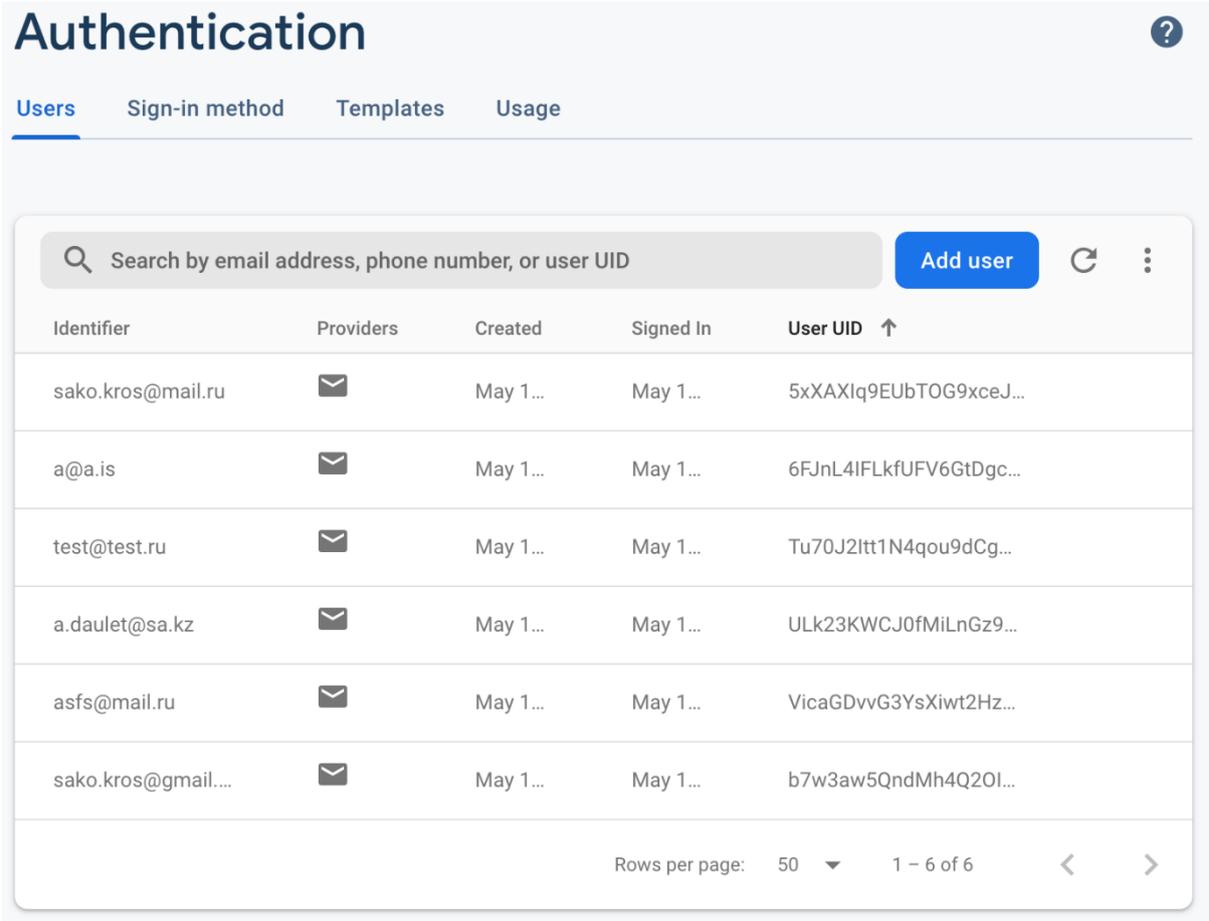


Рисунок 2.6 – Учетные записи

На рисунке 2.6 изображена панель управления учетными записями нашего diaMed приложения. Имеется таблица с идентификатором Email, почта для отправки пароля к пользователю, дата создание аккаунта, последний вход в приложений, UID идентификационный номер.

В добавку к IOS SDK, у Firebase есть SDK для Android и JavaScript. Все платформы могут использовать одну базу данных. В бесплатном пакете Firebase может обрабатывать до 100 одновременных соединений. Этого вполне достаточно для нашего веб-приложения diaMed. Это говорит о том что мы можем масштабировать.

В дополнение к IOS SDK, содержит Firebase SDK для Android и JavaScript. Все платформы используют одну и ту же базу данных. В бесплатном пакете Firebase может обрабатывать до 100 одновременных подключений. Этого вполне достаточно для нашего веб-приложения diaMed. Сервис предлагает что мы можем расширяться.



Рисунок 2.7 – База данных в реальном времени

Как видно на данном рисунке 2.7 изображена экран нашего diaMed веб приложений. В разделе реального времени имеется такие функций как Data, Rules, Backups, Usage.

Data – это хранилище наших пользовательский данных с уникальным uid кодом, внутри идет логин и пароль а так же информация нашего пациента.

Rules – это правила для базы данных.

Backups – делать бэкап нашего БД на случае не предвиденной ситуаций

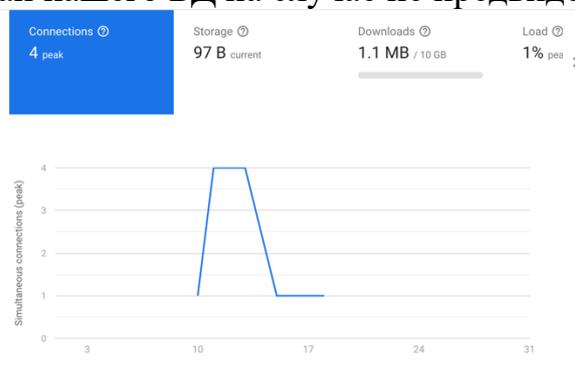


Рисунок 2.8 – Поток данных

Usage - это обращений к базе данных скачивание данных, подключение и размер.

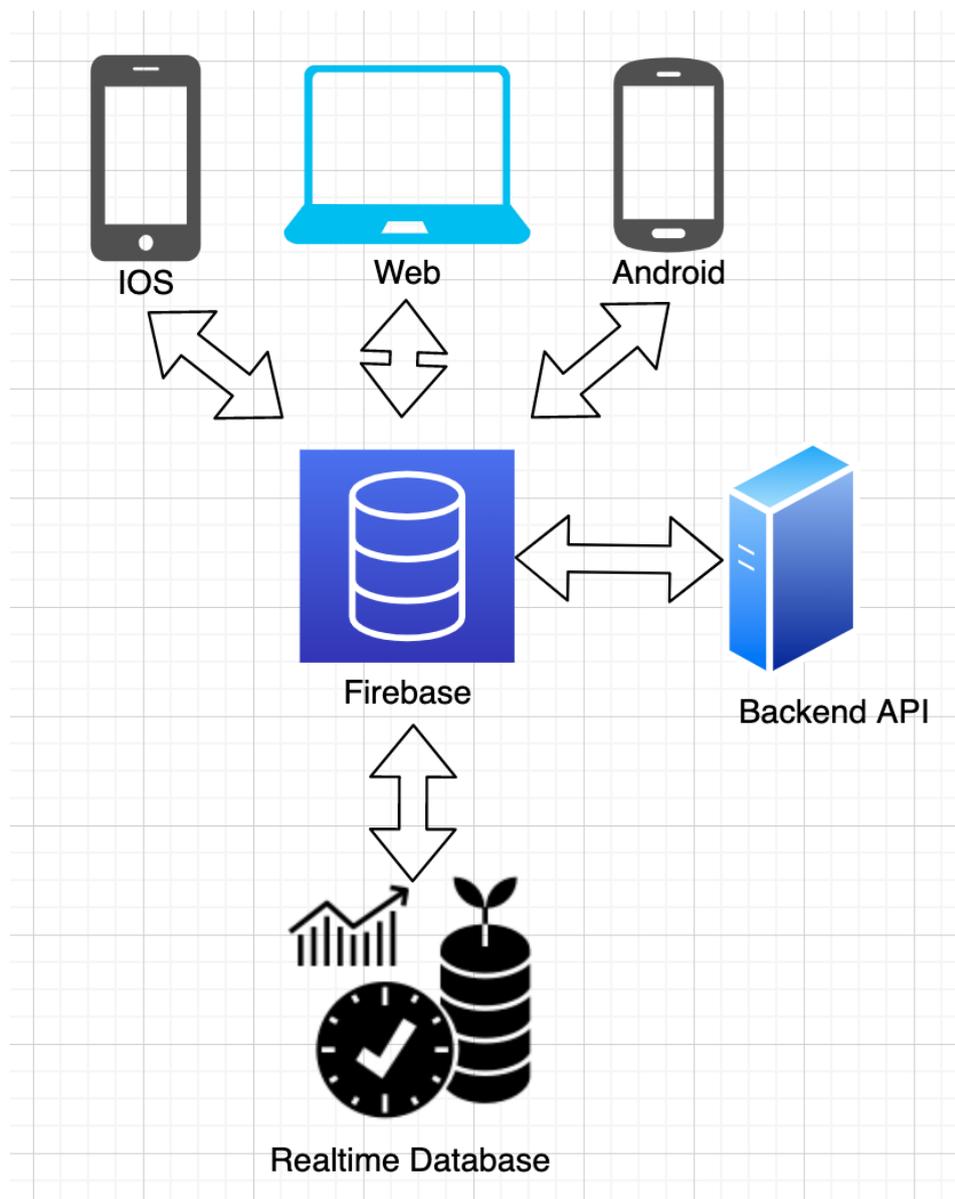


Рисунок 2.9 - Схема архитектуры базы данных

На рисунке 2.9 изображена схема архитектуры базы данных а так же их связи. Имеется гаджеты с операционной системой IOS, Android и windows(Браузер) подключенные к сервису Firebase. Firebase использует облачное вычисление и имеет связи с API, так же облачное хранилище в реальном времени.

2.1 ORM- (объектно-реляционное отображение)

2.2 SSL- (англ. *Secure Sockets Layer* - уровень защищённых сокетов)

2.2 Интерфейс пользователя работы с информационной системой

Важную роль в области информационных технологий играют базы данных, которые можно охарактеризовать как организованный сбор, хранение, обновление и обработку информации. Существуют следующие типы баз данных, которые классифицируются в соответствии с моделью данных:

- объектно-ориентированные и объектно-ориентированные;

Разработанное веб-приложение не может существовать без базы данных из-за необходимости хранить большой объем данных, которые должны быть активно отправлены клиенту. Взаимодействие приложений и баз данных создает данные о пациентах в режиме реального времени по запросу пользователя.

Для заполнения, изменения и представления информации используется набор специальных программных средств, называемых системой управления базами данных или системой управления базами данных.

Операции, предоставляемые СУБД:

- управление данными во внешней памяти;
- управление данными в оперативной памяти;
- резервное копирование и восстановление базы данных;

Базы данных-это хранилища, предназначенные специально для различных типов данных. Любая база данных, содержащая определенные модификации (реляционные, ориентированные документы), которая обеспечивает легкий доступ к данным. Системы управления базами данных (СУБД) - это специальные приложения (или библиотеки) для управления базами данных различных размеров и форм.

В настоящее время наиболее популярными СУБД в веб-домене являются MySQL, PostgreSQL и MongoDB. Но мы выбрали для нашего веб приложения noSQL. noSQL - первоначально со ссылкой на «не - SQL » или «не-реляционный» база данных обеспечивает механизм для хранения и извлечения данных , которые смоделированы в других , чем табличные отношениях, используемых в средствах реляционных баз данных. Базы данных NoSQL все чаще используются в больших данных и веб- приложениях реального времени.

Он часто используется в качестве сервера, доступного клиентам при сборке приложения. Одним из главных преимуществ СУБД является то, что с увеличением объема информации ее способность к хранению значительно снижается.

Однако после покупки услуги на Firebase NoSQL был менее обновлен, благодаря тарифу услуги, можно хранения данных масштабировать и увеличить обработке данных в режиме реального времени, бесплатном пакете содержит обработку до 100 клиентов. Я хочу отметить, что все данные в базе данных Firebase сохраняются в формате JSON. В JSON здесь нет таблиц или записей. Когда мы добавляем данные в базу данных Firebase, она становится ключом архитектуры JSON. Например, только что созданные данные выглядят следующим образом:

```
{
```

```
"users" : {
  "Tu70J2Itt1N4qou9dCgWz3WuKet2" : {
    "email" : "test@test.ru",
    "info" : {
      "Iz palca" : "2323",
      "Iz ruki" : "123",
      "tip" : 0
    },
    "name" : "Add adds Sdad",
    "password" : "sanjar955"
  },
}
```

Формат JSON не требует отображения схемы таблицы. Изменение данных системных документов позволяет разработчикам иметь большую гибкость, в то время как распределенная архитектура обеспечивает большую масштабируемость.

В результате СУБД выбирается для проектов, требующих управления большим объемом информации и не вписывающихся в реляционную модель данных. Разработанная программа невелика и не требует гибкости, обеспечиваемой MongoDB, а MySQL не получил должного развития из-за сообщества, поэтому целесообразно использовать NoSQL в качестве СУБД для нашего веб-приложения diaMed.

2.3 Средства разработки веб-приложения

Xcode-это программное обеспечение среды разработки, интегрированное для платформ macOS, iOS, watchOS и tvOS, разработанное компанией Apple. Первая версия была выпущена в 2003 году. Стабильные версии распространяются бесплатно через Mac App Store. Зарегистрированные разработчики также могут получить доступ к бета-версиям через веб-сайт разработчика Apple.

main.storyboard-это визуальное представление пользовательского интерфейса приложения IOS, отображающее связи между экранами контента и этими экранами. Storyboard состоит из последовательности представлений, каждая из которых представляет контроллер представления и его типы; представления связаны с объектами segue, представляющими переход между двумя контроллерами представления.

Xcode предлагает визуальный редактор для storyboard, в котором вы можете создавать и создавать пользовательский интерфейс приложения, добавляя такие представления, как кнопки, представления таблиц и текстовые представления. Кроме того, storyboard позволяет подключать представление к объекту контроллера и управлять передачей данных между контроллером представления.

Использование storyboard-это рекомендуемый способ создания пользовательского интерфейса приложения, поскольку они позволяют визуализировать представление и поток пользовательского интерфейса на одном полотне.

Платформа UIKit обеспечивает инфраструктуру, необходимую для приложений IOS. Архитектура окна и представления для реализации нашего интерфейса обеспечивает инфраструктуру обработки событий для передачи нескольких сенсорных и других типов ввода в приложение и базовый цикл запуска, необходимый для управления взаимодействием между пользователем, системой и приложением. Декомпозиция включает в себя интерфейс. Animate, другие функции, предлагаемые платформой, включают поддержку анимации, поддержку документов, поддержку изображений и печати, доступную информацию об устройстве, управление текстом и экраном, поддержку поиска, поддержку доступа, поддержку расширений приложений и управление ресурсами.

SWIFT - это новый язык программирования для разработки приложения IOS и OS X, который сочетает в себе лучшие из C и Objective-C, но ему не хватает ограничений, применяемых для обеспечения совместимости с C.декомпозиции в мире Swift использует безопасные шаблоны программирования и добавляет современные функции, которые делают создание приложения простым, более гибким.

CocoaPods-очень мощный инструмент со статусом, который должен иметь любой IOS разработчик. Тем не менее, мы можем легко, быстро и легко подключать различные полезные инструменты, утилиты и библиотеки, которые значительно облегчают нашу жизнь при разработке приложений. Таким образом мы подключили сервис Firebase. Но мы должны использовать весь инструмент, и

CocoaPods не является исключением. Под слоем знаний, который позволяет вам начать изучать этот инструмент.

Firebase - GoogleService-Info.plist, чтобы получить файл конфигурации Firebase iOS (GoogleService-Info.plist).

Средства разработки для нашего веб-приложение мы выбрали xCode 12.4 на языке SWIFT 5 с использованием Фреймворком.

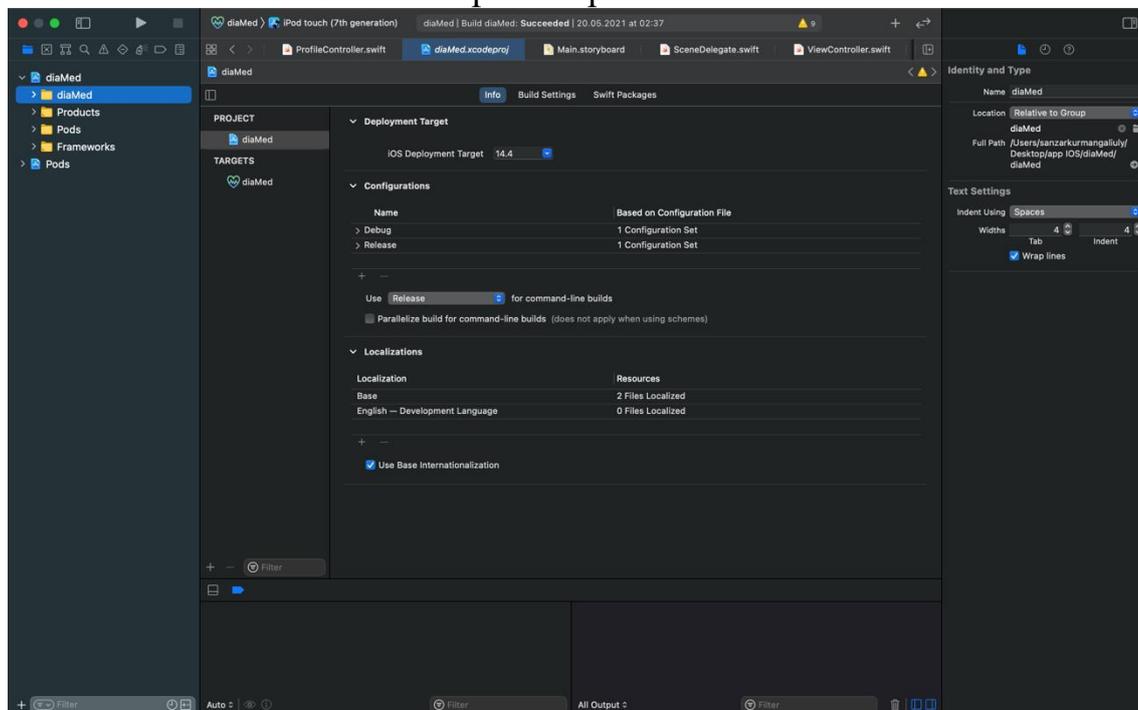


Рисунок 3.0 xCode 12.4

На рисунке 3.0 изображена сред разработки xCode с версией 12.4 для Mac OS Big Sur. Очень удобный простой при использования среда разработки, имеет внутренний терминал для подключения библиотек и инструментов.

3 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРИЛОЖЕНИИ

3.1 Моделирование базы данных

Основываясь на услугах, которые должен предоставлять проект, база данных требует сущностей и их отношений, которые определены в этой главе.

Пользователь-лицо, которое выполняет определенные действия в приложении.

Действие – действие, авторизация, выход, вход пользователя.

Запись данных – Viewcontroller для записи данных сахара в крови, так же масса тела и рост человека.

Информация – информация и анализ запись данных

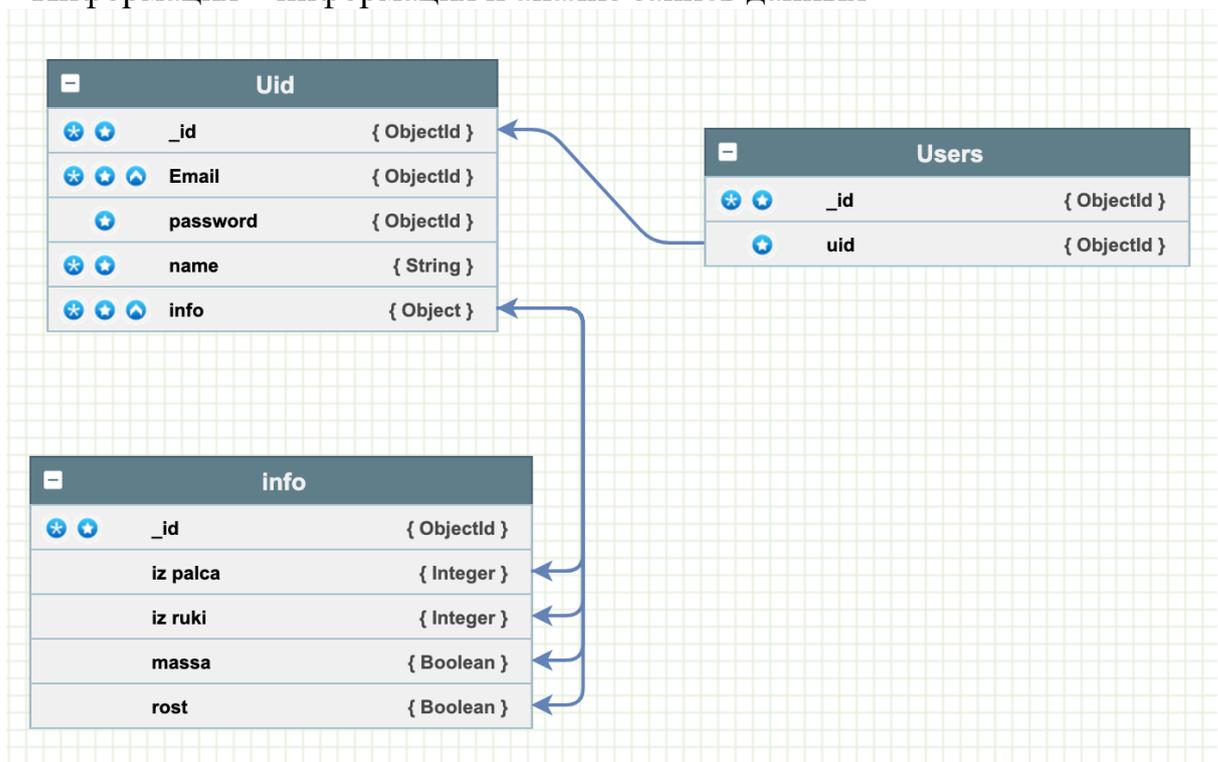


Рисунок 3.1 - ER-диаграмма JSON.

Как показано на рисунке 3.1, структура данных NoSQL-СУБД не использует реляционную модель. существует множество приложений, которые самостоятельно перестраивают текущий вопрос, часто в необычном виде. Эти бесконечные решения обеспечивают неограниченное формирование записи и сохранение данных о значении ключа.

В отличие от традиционных СУБД, некоторые базы данных NoSQL, такие как MongoDB, позволяют группировать наборы данных с другими базами данных. Эта СУБД хранит данные в виде единиц. Эти данные могут представлять один объект, такой как json, а также точно соответствовать запросам поля. Базы данных NoSQL (например, SQL в реляционной базе данных) не используют общий формат запроса. Каждое решение использует свою систему запросов.

3.2 Безопасность веб-приложения

Защита нашего приложения diaMed стоит в приоритете на первом месте, так как приложение имеет конфиденциальные сведения нашего пользователя. Для ограничения доступа пользователей использовались такие операции, как авторизация и аутентификация. Ограничение доступа и индивидуальность пользователя.

Регистрация – регистрация пользователя приводит к созданию учетной записи, для записи данных об уровне сахара в крови, регистрация имеет ФИО, email и пароль. Firebase предлагает нам дополнительные регистрационные методы с помощью API с другими сервисами как Google, Facebook, по номеру телефона с sms паролем, twitter, Microsoft и тд.

Вход – вход производится после регистраций, ввод email и пароля, с сохранением сессии о пользователя.

Аутентификация – это процесс проверки подлинности email и пароля, введенного пользователем, который сохранен в базе данных сервера Google Firebase.

Токены аутентификации – это сгенерированный код предоставляют собой средство авторизации для каждого запроса от клиента к серверу. Токен требует, чтобы пользователи получили сгенерированный компьютером код(UID).

Когда вы выполняете аутентификацию с помощью Firebase, вы можете встретить три типа токенов аутентификации:

	Создается Firebase, когда пользователь входит в приложение.
Токены Firebase ID	Эти токены представляют собой подписанные JWT, которые надежно идентифицируют пользователя в проекте Firebase. Эти токены содержат основную информацию профиля пользователя, включая строку идентификатора пользователя, которая уникальна для проекта Firebase. Поскольку целостность токенов ID может быть проверена, вы можете отправить их на внутренний сервер для идентификации текущего пользователя, вошедшего в систему.
Токены поставщика удостоверений	Создано федеративными поставщиками удостоверений, такими как Google и Facebook. Эти токены могут иметь разные форматы, но часто являются токенами доступа OAuth 2.0. Приложения используют эти токены для проверки того, что пользователи успешно прошли аутентификацию с помощью поставщика удостоверений, а затем конвертируют их в учетные данные, используемые службами Firebase.
Пользовательские токены Firebase	Создано вашей пользовательской системой аутентификации, чтобы пользователи могли входить в приложение, используя вашу систему аутентификации. Пользовательские токены – это JWT, подписанные с использованием закрытого ключа учетной записи службы. Приложения используют эти токены так же, как они используют токены, возвращенные от

федеративных поставщиков удостоверений. [
<https://firebase.google.com/docs/auth/users>]

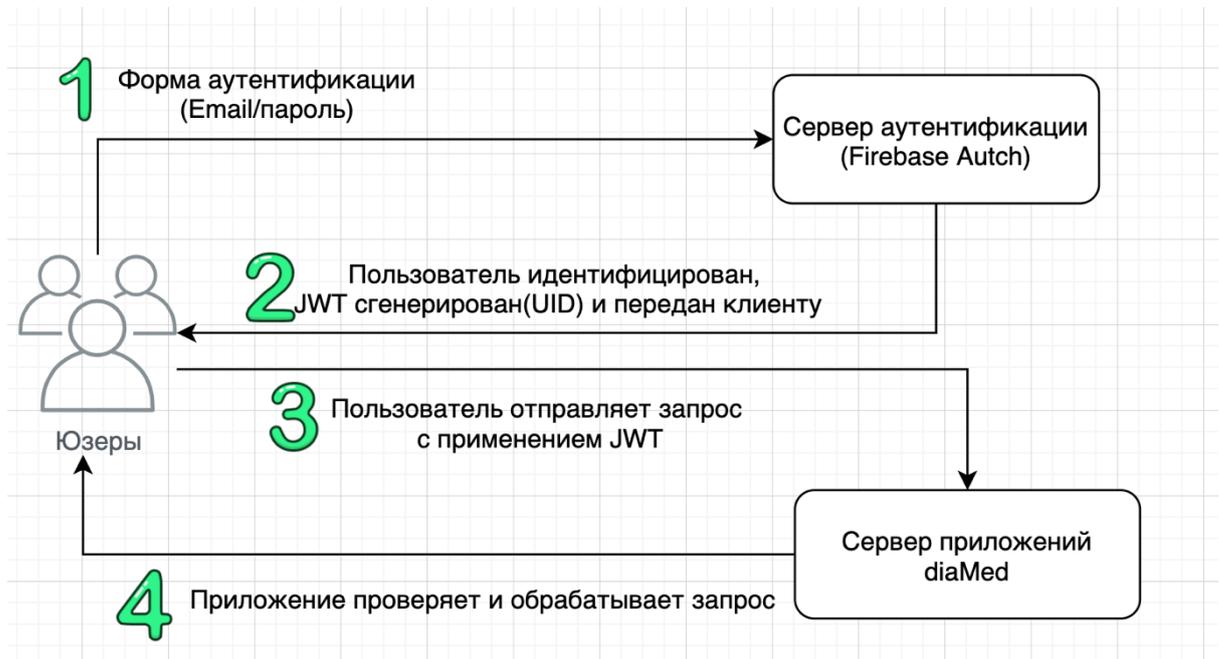


Рисунок 3.2 Алгоритм JWT

Надо отметить что доступ к приложению имеется 2 роли:

- врачи
- пациенты
- Администратор

У администратор имеется доступ к пользовательским данным, в случае отключение или заблокированные пользователя, производится с помощью функций Disable Account. Так же читать изменять данные сахара крови, имени, и тд.

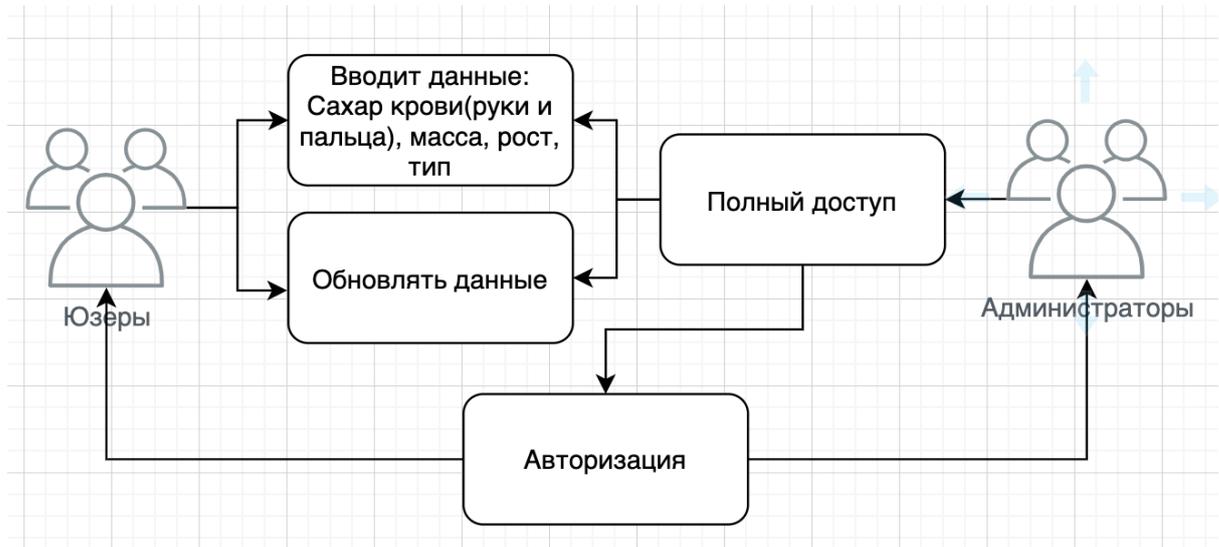


Рисунок 3.3 Диаграмма использование вариантов

Каждый раз, когда пользователь(Юзер) обращается к API, маркер должен быть добавлен в заголовок авторизации. Сервер проверяет отправленный токен и определяет, представлен ли пользователь тем человеком, за которого он себя выдает.

JWT предотвращает несанкционированный вход в нашу веб приложение делая ее безопасной. Firebase может до 100 юзеров и админов в реальном времени обслуживать, это позволит ускорит работу нашего приложение. И повышая безопасность.

3.3 Разработка компонентов серверной части

Разработка серверной части (backend). Запросы к серверной части для обращения к базе данных и обработки данных осуществляется по адресу(<https://websan-fa50d-default-rtdb.europe-west1.firebaseio.com/>)

Ссылка указывает название нашего проекта, и локация сервера. В нашем случае сервер находится в Европе в городе Бельгий. Но прежде чем начнем создавать контроллеры, у нас должны быть другие элементы системы, которые описаны в приложении В.

Сервер Firebase ожидают сообщение с клиентскими запросами с помощью ответного API сообщений. При просмотре на стороне пользователя swift-код, выполняемый в веб-приложениях пользователя, отвечает за запрос данных с сервера и, при необходимости, за управление веб-приложением. Например, если пользователь вводит недопустимое значение в форму, клиентский код открывает окно сообщения об ошибке без доступа к серверу или веб-приложению.

При просмотре на стороне сервера пользователь запрашивает новую веб-приложения (при клике профиль) с сервера для каждого обновления и отображает данные о себе. Например, если пользователь вводит недопустимое значение в объект, попросите клиентский код сохранить данные правильно, так как это приведет к сообщению об ошибке.

В последнее время все чаще используется разработка приложений, предлагаемых клиентом, а обмен данными с сервером осуществляется через запросы API.

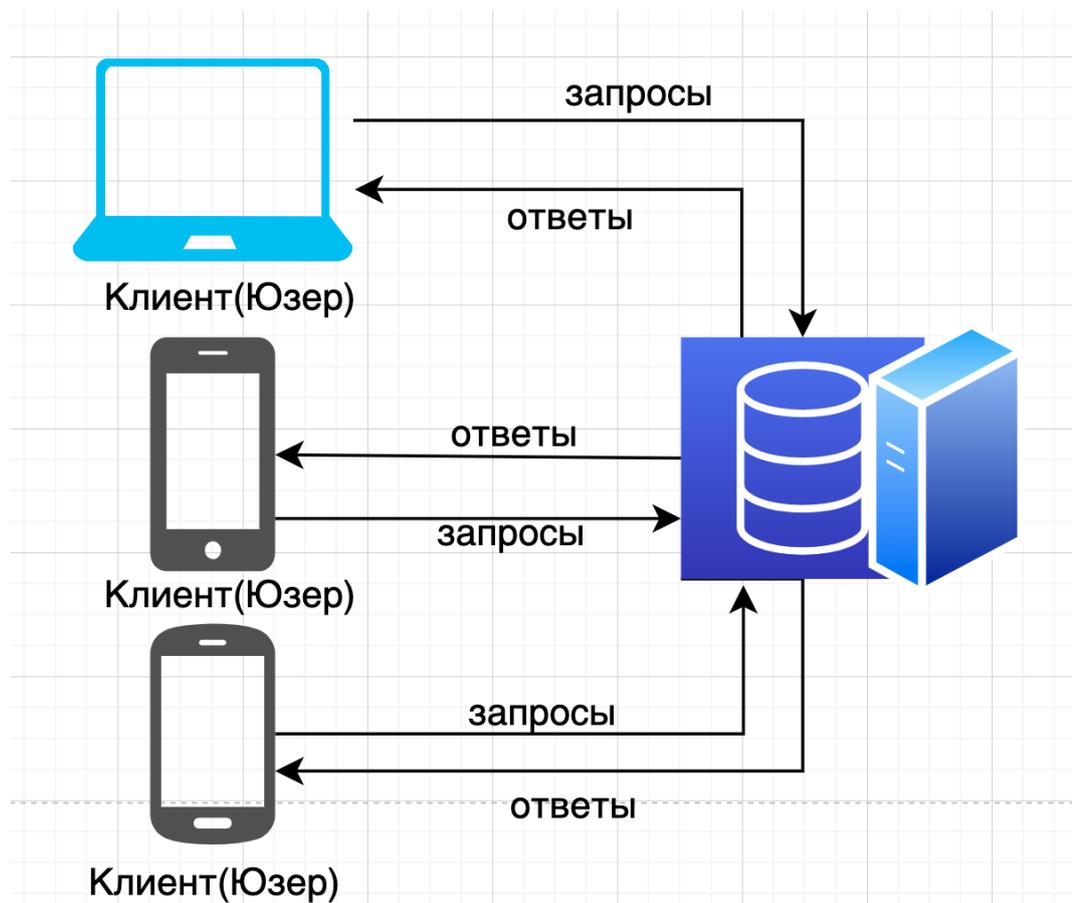


Рисунок 3.4 - Взаимодействие клиента и веб-сервера

Например, если пользователь вводит недопустимое значение в форму, код клиента запрашивает правильное хранение данных, так как это приведет к сообщению об ошибке. Наиболее распространенным действием этих методов является использование представления на стороне сервера исходной веб-страницы, а затем использование представления клиента для обновления по мере необходимости. В последнее время многие разработчики приложений используют клиентский рендеринг и запросы API для обмена данными с серверами.

Веб-сервер-это компьютер, на котором уже размещено наше приложение и который доступен пользователям по всему миру онлайн. Как разработчик, мы можете выбрать операционную систему (Windows, Linux и т.д.), программное обеспечение веб-сервера (Apache, Nginx, IIS) и аппаратное обеспечение, соответствующее вашим потребностям. Но в нашем случае мы не имеем права выбирать операционную систему, а также веб-сервер. Почему не можем? Так как мы можем отправить на сервер AppStore где можно будет скачать наше приложение с помощью apple id.

Управление базами данных Google Cloud storage предоставляет нам простой в использовании, высокоскоростной веб-сервер СУБД+, что означает, что он бесплатный и имеет доступ к хостингу машинного обучения и многое другое. Тем не менее, мы не можем жестко и глубоко работать с веб-сервером, потому что Google сделал это за нас. По этой причине мы выбрали службу Firebase. На базовом уровне веб-сервер обрабатывает запрос веб-контента и передает его

клиенту. Это достигается путем выбора материалов, необходимых пользователю, на основе интерфейса прикладного программирования, а затем выбора файла на swift и главной веб-странице. Таблица стилей раскадровки передается клиенту и отображается в приложении IOS.

База данных-это место, где хранятся данные приложения и пользователей. Базы данных хранят, организуют и предоставляют доступ к данным приложений. Существует несколько типов баз данных системы управления базами данных, которые могут быть использованы для разработки веб-сайтов на стороне сервера.

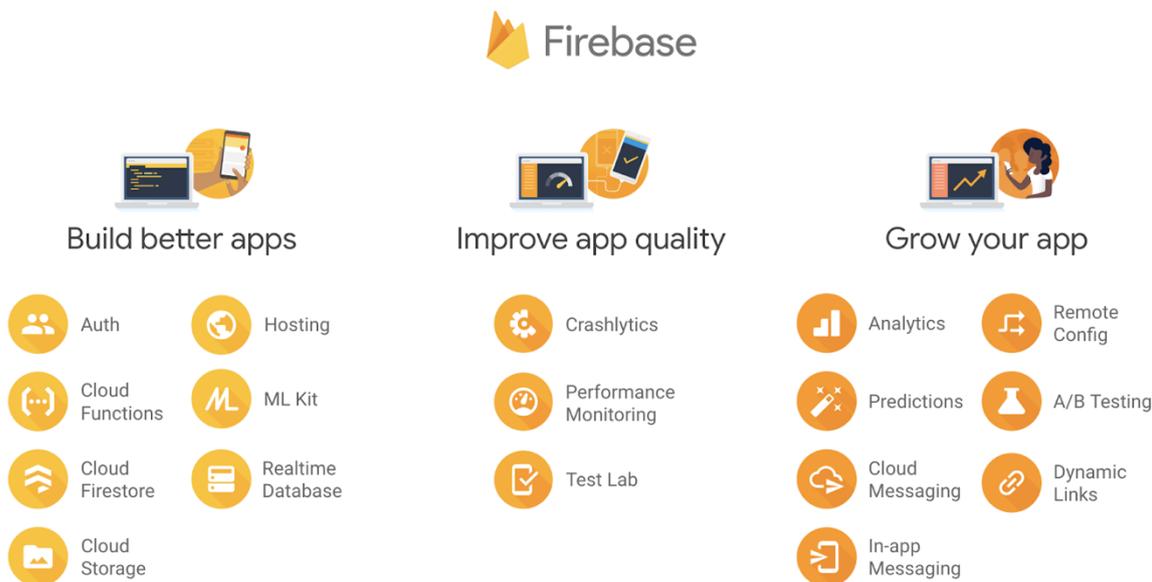


Рисунок 3.5 - Услуги Firebase

Firebase позаботится обо всем, что нам понадобится в нашем мобильном приложении. Сервис сочетает в себе полные функции бэкенда, такие как хранение данных, синхронизация, аутентификация, облачные функции (реализация бэкенд-кода), и на данный момент новыми технологиями являются машинное обучение, которые реализуют различные функции, основанные на машинном обучении в приложении (распознавание текста, объектов в изображениях и многое другое). В ИИ найдется применение всегда.

3.4 Разработка компонентов клиентской части

Приложение начинает работать на операционной системе iOS, которая начинается с запуска объекта класса делегата. Его цель - принимать и передавать системные вызовы программе, например, для выполнения первоначальной настройки графического интерфейса приложения. Все методы и механизмы, не связанные с запуском приложения или получением сообщения от системы, должны быть переданы отдельным классам. Сразу после завершения начальной настройки управление должно быть передано классу, выполняющему другие операции опции приложения: авторизацию, перенастройку интерфейса в зависимости от критериев, загрузку исходных данных, получение важных меток и другие. Все коды, размещенный в AppDelegate, [Приложений Б] - частая ошибка создателей. В недавнем обновлении Delegate разделили на две части AppDelegate и SceneDelegate делегат класса UIWindowScene, используется для управления сценами в приложении.

Чтобы это было понятно, здесь для всех примеров внешних фреймов есть отдельный код. Среда очень обычна для тех, кто использует интегрированную модель сборки CoreData, но мы подключили к нашему AppDelegate облачный сервис Firebase путем подключения импорта import Firebase затем подключили конфигурацию FirebaseApp.configure() для общего приложения.

Часто можно увидеть выполняемые соответствующие функции:

Установка и обслуживание сессий Authenticator

Если программа поддерживает UITabBarController, настройте диспетчер вкладок.

Тестирование и инициализация обновлений

Отчет на внешние серверы статистики

Синхронизировать модель данных

Создание класса AuthViewController и передача данных в SceneDelegate, а также начало наращивания данных о процессах, запущенных на Navigationbar, было бы гораздо более сложным выводом firebase.

При открытии содержимое файла заголовка AuthViewController.swift в отдельном окне, дважды щелкнув по нему. Этот файл содержит наклейку для расширения класса UIViewController, [Приложений Б] отмеченную основным textinterface. Эта возможность была добавлена во второй версии swift. Теперь проделали аналогичный действие с наведением текста Вход и Регистрация мы проделали нить называемые Outlet, путем подключения графического интерфейса к коду. Наводим курсор на окно со значком (файл AuthViewController.Swift), за курсором появится синий след. При отпускании мыши и кнопки из описания интерфейса AuthViewController. Это свойство объекта, которое ссылается на другой объект (в данном случае на визуальный компонент).

Для нас нужно задать имя выходного объекта, под которым мы будем указывать видимый компонент, titleLabel, он станет текстом вход и регистрация. Затем выбирается изображение объекта, и оно выбрано правильно: UILabel, затем выбирается изображение ссылки на объект: слабое или сильное в списке историй.

Если вы выберете strong, объект, указанный свойством, будет существовать до тех пор, пока свойство ссылается на него, и в этом случае его нельзя оставить механически после завершения приложения.

С другой стороны, при работе гиперссылки слабой (слабой), объект обладает способностью к самоуничтожению. Наконец, выберите изображение слабого звена и нажмите кнопку Подключиться. В результате эта строка будет добавлена в код:[Приложение Б]

3.5 Серверное программное обеспечение

Серверная часть по поводу увеличение памяти RAM, ЦП, память это все лежит на облачном сервисе Firebase. Мы упираемся только на создание приложение. А вся серверная аппаратная часть все лежит на плечо облачного сервиса Firebase. Языки программирования сами по себе очень гибки и популярные, и благодаря этому мы, разработчики, получаем мощный инструмент для создания веб-приложения. Вначале меня интересовала гибкость Firebase и скорость развертывания в нашем проекте.

И нам было достаточно бесплатного пакета.

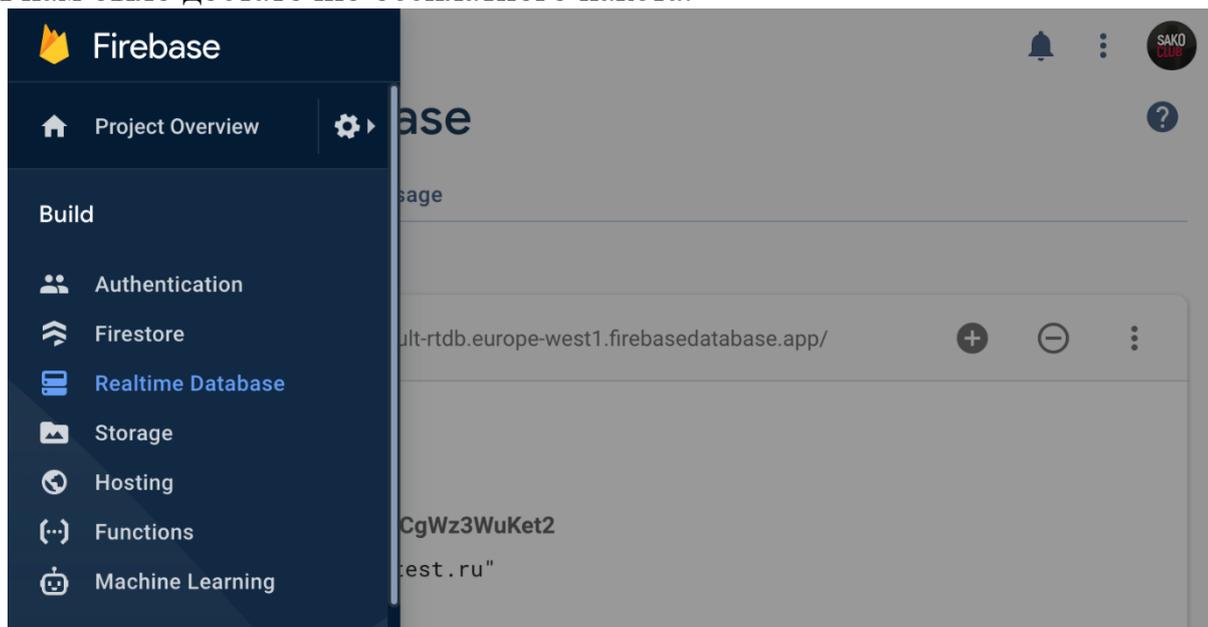


Рисунок 3.6 - Интерфейс нашего серверного пакета:

Облачное вычисление это модель для обеспечения надлежащего доступа к сети на основе претензии к определенному настраиваемому фонду вычислительных ресурсов (например, сети передачи данных, серверы, устройства хранения данных, приложения и услуги - вместе и отдельно), которые имеют каждую возможность минимальные эксплуатационные расходы или получают и выдают почти мгновенно, написав в тех. поддержку.

Одним словом, можно много что говорить о облачном сервисе Firebase так как это лучшее решение для нашего приложение.

3.6 Тестирование веб-приложения

Давайте начнем с того как выглядит наше приложение внешне:



Рисунок 3.7 – Вид с рабочего стола

Как видим на рисунке 3.7 приложение имеет красивый вид логотипа, то есть это фирменный лого приложение в виде сердца и пульса на черном фоне (Почему черный? Так как у IOS телефона имеется темный режим и светлый, в данном случае стоит темный режим).

При клике по нашему приложению diaMed открывается окно регистрации. Сверху над кнопкой имеется «Войти» имеется надпись того что при клике «Войти» откроется окно со входом.

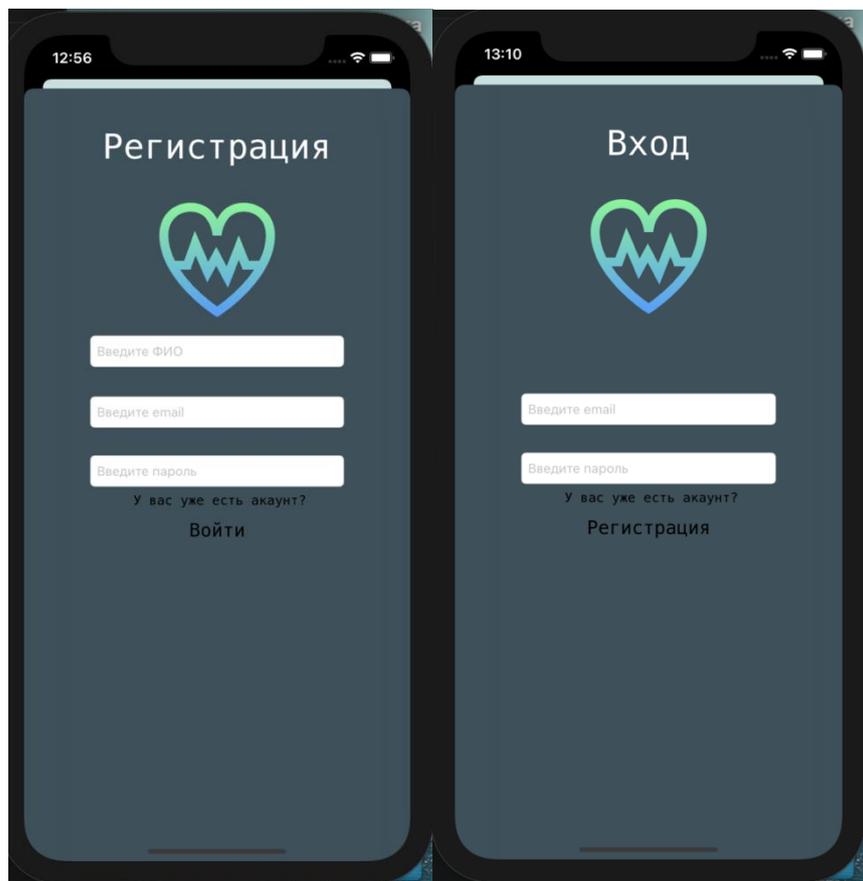


Рисунок 3.8 – Окно с регистрацией и входом

На рисунке 3.8 было изображена окно со входом и регистрацией. Правильным решением было убрать кнопку Вход, Регистрация для входа в приложение, так как мы сделали вход и регистрацию приложение с помощью клавиши «Done(Готова)» после ввод пароля.

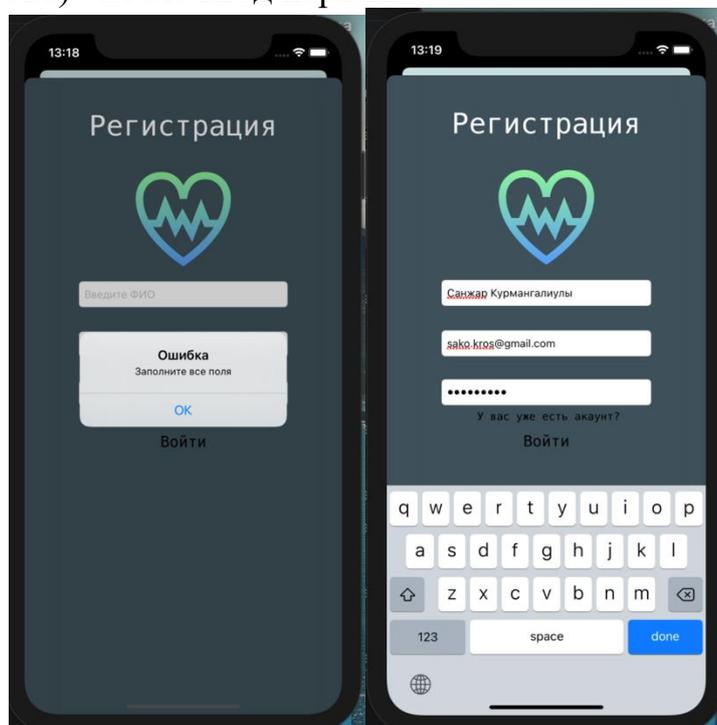


Рисунок 3.9 – Заполненный вход и регистрация

При не правильном заполнений вызывается ошибка входа и регистраций «Заполните все поля».

После нажатие готова, у нас создается учетная запись, как мы видим на рисунке 4.0 с идентификатором (Email) UID «5xXAXIq9EUbTOG9xceJ4mE4TD493»

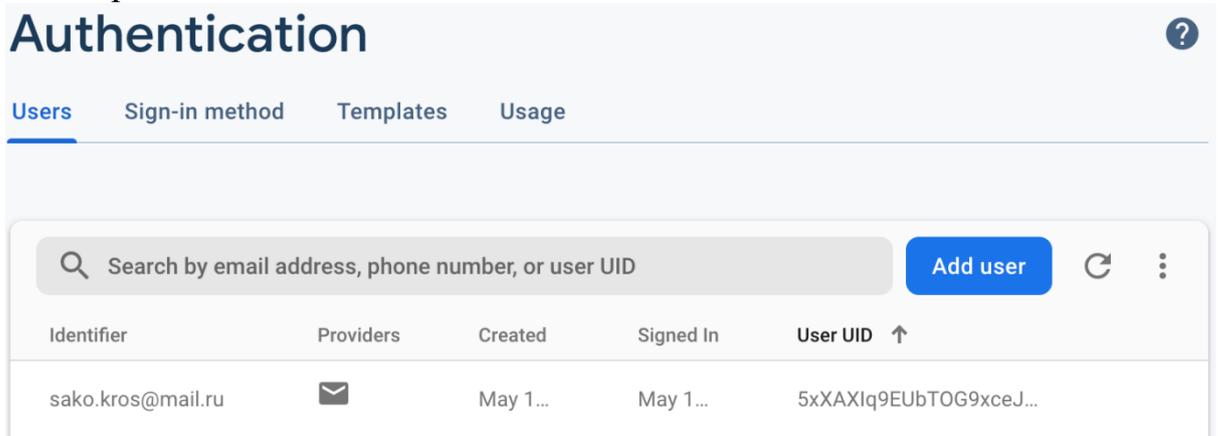


Рисунок 4.0

Таким образом на рисунке 4.1 создается в ветке noSQL в виде JSON users/uid/{name, email, password} Пароль можно было бы не выводить, но для изменение и точки восстановления было принято записать пароль в базу для администратора, нужно было выводить.

websan-fa50d-default-rtdb

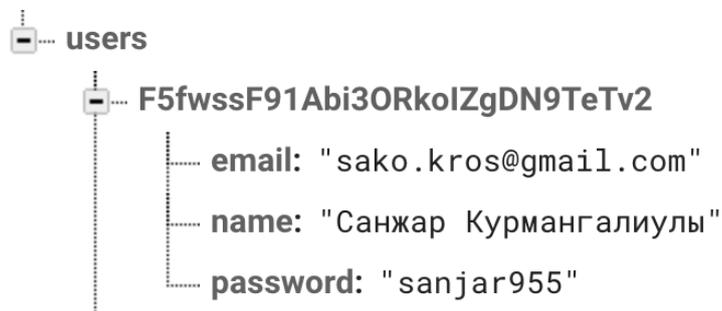


Рисунок 4.1 – Данные о пользователе

У нас после регистраций открывается ViewConrtoller то есть для ввода наших данных.

В правом верхнем углу имеется выход (Logout) для выхода с аккаунта. Но при каждом заходе в приложение не требуется каждый раз вводить пароль и email так как сохраняется текущая сессия. Один раз зашли, всё.

В левом углу имеется профиль при клике открывается следующая страница в виде таблицы TableViewUI для вывода запися по дням, то есть в какое время было сданы данные с глюкометра.

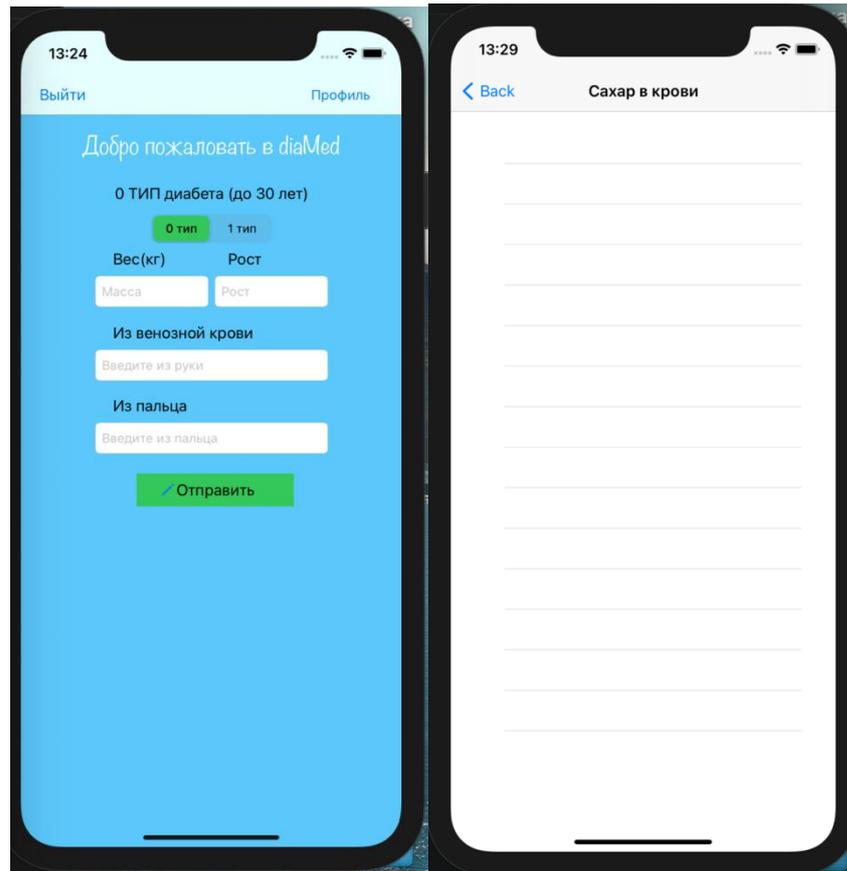


Рисунок 4.2 – Главная окошка для ввода данных и профиль

Вводим данные выбирая тип сегмента 0 и 1 тип. Меняется название текста при выборе сегмента.

Давайте взглянем на рисунок 4.3, дальше отправим и посмотрим как будет выглядеть на нашем базе в реальном времени.

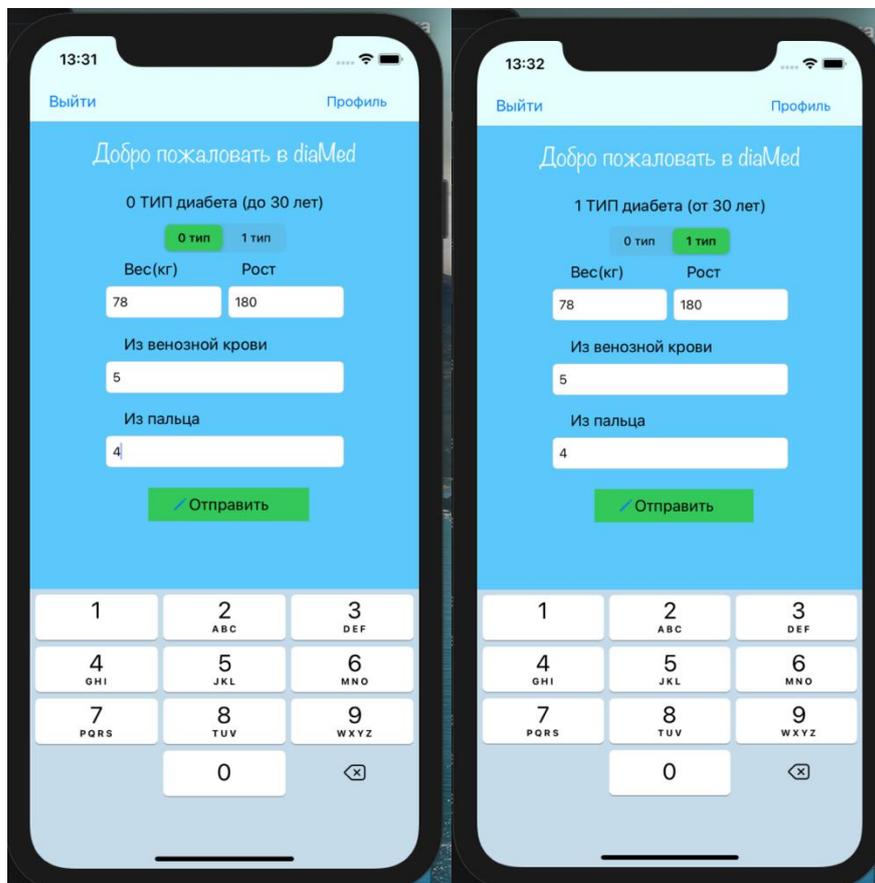


Рисунок 4.3 – Запись данных
websan-fa50d-default-rtdb

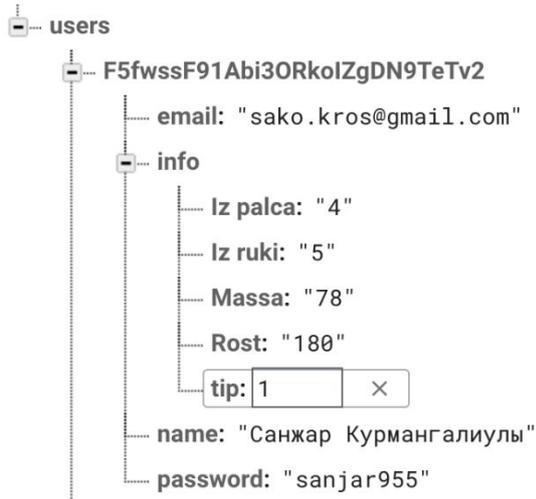


Рисунок 4.4 – Результат отправки в БД

Как видим все хорошо работает. При входе врача, у нее открывается в веб-сайт, с выводом данных о пациенте в виде таблицы

Интерфейс врача аналогичный только они могут входить в свой личный кабинет с помощью любого браузера на компьютере.

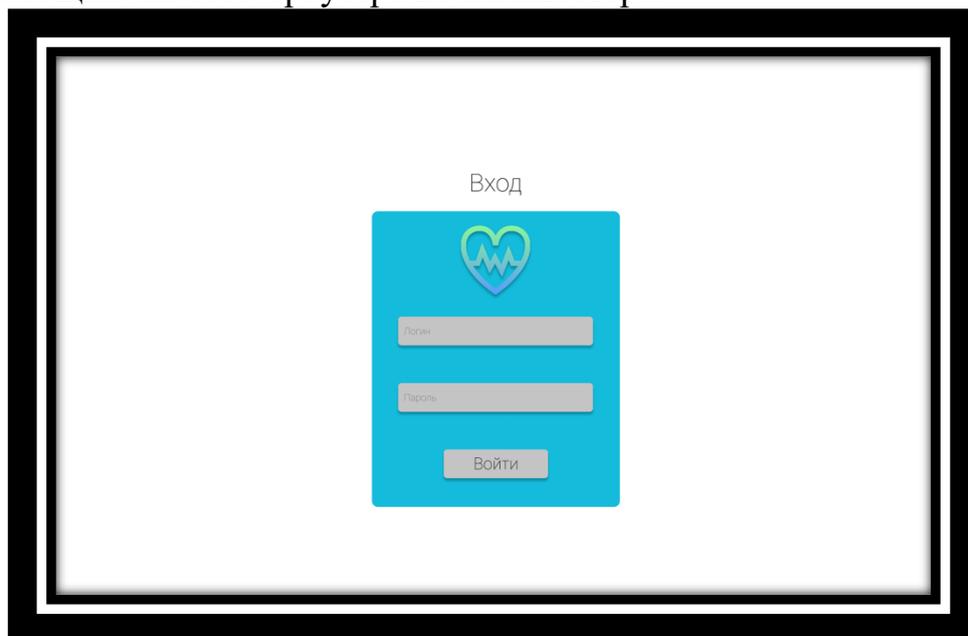
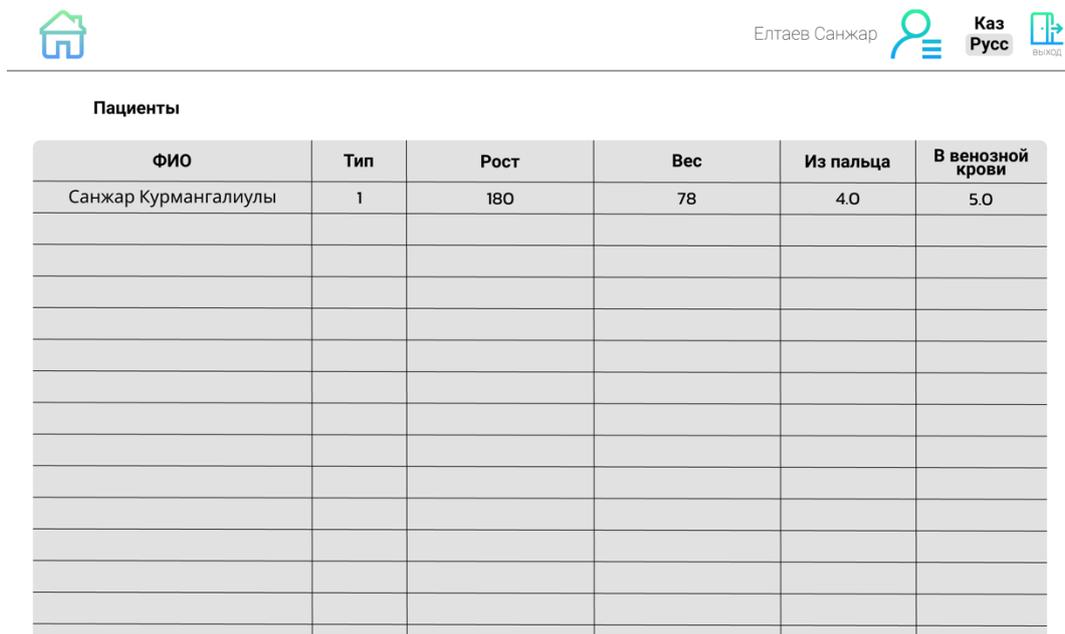


Рисунок 4.5 – Вход для врача

На рисунке 4.6 изображена открытая страница личного кабинета, как видим выводится список пациентов и выводятся их данные о сахаре.



ФИО	Тип	Рост	Вес	Из пальца	В венозной крови
Санжар Курмангалиулы	1	180	78	4.0	5.0

Рисунок 4.6 – Таблица с пациентами

Затем врач делает анализы расчеты, и связь с пациентом. Так как для определения пред сахарный диабет, врач в течение нескольких дней должен понаблюдать за пациентом виртуально.

ЗАКЛЮЧЕНИЕ

В данном дипломном проекте мы разработали приложение под названием «diaMed» для мобильной операционной системы IOS, используя среду разработки Xcode с языком Swift.

При разработке веб-приложения были учтены все современные технологий с применением облачного хранилища и вычислений.

Были проведены следующие действия:

- анализ существующих архитектур веб-приложений diamed;
- выбор инструментов проектирования и разработки программного обеспечения;
- проектирование ИС с использованием диаграмм ER и UML;
- Создание веб-приложение.

Для реализаций серверной части нашего веб-приложения были использованы сервис Firebase от компаний google. Тем самым все обработки запросов и записи данных в базу было использована удобный сервис Firebase. Для клиентской части Swift с использованием StoryBoard для внешнего интерфейса.

Мы протестировали финальное приложение на симуляторе iPhone 11. Мы узнали, как создать пользовательский интерфейс с помощью рабочей панели раскадровки (Storyboard), как связать графические элементы с кодом приложения и как создавать обработчики событий. Мы знаем основы Swift.

Таким образом дипломный проект получился очень интересным. Приложение имеет все функции, необходимые для обследования пациентов с диабетом, в будущем мы можем улучшить приложение с помощью push-уведомления для оценки теста на глюкозу в крови, и добавить мессенджер.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. <https://webmed.irkutsk.ru/doc/pdf/algosd.pdf>
2. <https://developer.apple.com/documentation/uikit>
3. <https://firebase.google.com/docs/auth?hl=uk>
4. https://www.ibase.ru/files/firebird/Firebird_Hardware_Guide_2015_rus.pdf
5. https://developer.apple.com/documentation/swift/swift_standard_library
6. <https://docs.swift.org/swift-book/index.html>
7. <https://firebase-wordpress-docs.readthedocs.io/en/latest/database/save-data-realtime-firestore.html>
- 8.

ПРИЛОЖЕНИЕ А

Для подключение к бд

Подключение к активному окну

```
//  
// SceneDelegate.swift  
// diaMed  
//  
// Created by Санжар Курмангалиулы on 02.04.2021.  
//
```

```
import UIKit
```

```
import Firebase
```

```
class SceneDelegate: UIResponder, UIWindowSceneDelegate {
```

```
    var window: UIWindow?
```

```
    /*func application(_ application: UIApplication, didFinishLaunchingWithOptions  
launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
```

```
        FirebaseApp.configure()
```

```
        Auth.auth().addStateDidChangeListener { (auth, user) in
```

```
            if user == nil{
```

```
                self.showModalAuth()
```

```
            }
```

```
        }
```

```
        return true
```

```
    }
```

```
    func showModalAuth(){
```

```
        let storyboard = UIStoryboard(name: "Main", bundle: nil)
```

```
        let newvc = storyboard.instantiateViewController(withIdentifier:  
"AuthViewController") as! AuthViewController
```

```
        self.window?.rootViewController?.present(newvc, animated: true, completion: nil)
```

```
    }*/
```

```
    func scene(_ scene: UIScene, willConnectTo session: UISceneSession, options  
connectionOptions: UIScene.ConnectionOptions) {
```

```
        guard let _ = (scene as? UIWindowScene) else { return }
```

Продолжение приложения А

```
}  
  
func sceneDidDisconnect(_ scene: UIScene) {  
  
`application:didDiscardSceneSessions` instead).  
}  
  
func sceneDidBecomeActive(_ scene: UIScene) {  
    Auth.auth().addStateDidChangeListener { (auth, user) in  
        if user == nil{  
            self.showModalAuth()  
        }  
    }  
}  
  
func showModalAuth(){  
    let storyboard = UIStoryboard(name: "Main", bundle: nil)  
    let newvc = storyboard.instantiateViewController(withIdentifier:  
"AuthViewController") as! AuthViewController  
    self.window?.rootViewController?.present(newvc, animated: true, completion: nil)  
}  
  
func sceneWillResignActive(_ scene: UIScene) {  
  
}  
  
func sceneWillEnterForeground(_ scene: UIScene) {  
  
}  
  
func sceneDidEnterBackground(_ scene: UIScene) {  
  
}  
  
}
```

ПРИЛОЖЕНИЕ Б

```
// Регистрация и вход

// AuthViewController.swift
// diaMed
//
// Created by Санжар Курмангалиулы on 07.04.2021.
//

import UIKit
import Firebase

class AuthViewController: UIViewController {
    var signup: Bool = true{
        didSet{
            if newValue{
                titleLabel.text = "Регистрация"
                nameField.isHidden = false
                enterButton.setTitle("Войти", for: .normal)
            }else{
                titleLabel.text = "Вход"
                nameField.isHidden = true
                enterButton.setTitle("Регистрация", for: .normal)
            }
        }
    }
    @IBOutlet weak var enterButton: UIButton!
    @IBOutlet weak var emailField: UITextField!
    @IBOutlet weak var passwordField: UITextField!
    @IBOutlet weak var nameField: UITextField!
    @IBOutlet weak var titleLabel: UILabel!
    @IBOutlet weak var imageLogo: UIImageView!
    override func viewDidLoad() {
        super.viewDidLoad()
        nameField.delegate = self
        emailField.delegate = self
        passwordField.delegate = self
    }
    @IBAction func switchLogin(_ sender: UIButton) {
        signup = !signup
    }

    func showAlert(){
```

Продолжение приложения Б

```
let alert = UIAlertController(title: "Ошибка", message: "Заполните все поля",
preferredStyle: .alert)
    alert.addAction(UIAlertAction(title: "OK", style: .default, handler: nil))
    present(alert, animated: true, completion: nil)
}
}
extension AuthViewController: UITextFieldDelegate{
    func textFieldShouldReturn(_ textField: UITextField) -> Bool {
        let name = nameField.text!
        let email = emailField.text!
        let password = passwordField.text!

        if (signup){
            if (!name.isEmpty && !email.isEmpty && !password.isEmpty){
                Auth.auth().createUser(withEmail: email, password: password) { (result,
error) in
                    if error == nil{
                        if let result = result{
                            print(result.user.uid)
                            let ref = Database.database().reference().child("users")
                            ref.child(result.user.uid).updateChildValues(["name" : name, "email"
: email, "password": password])
                            self.dismiss(animated: true, completion: nil)
                        }
                    }
                }else{
                    showAlert()
                }
            }else{
                if (!email.isEmpty && !password.isEmpty){
                    Auth.auth().signIn(withEmail: email, password: password) { (result, error) in
                        if error == nil{
                            self.dismiss(animated: true, completion: nil)
                        }
                    }
                }else{
                    showAlert()
                }
            }
        }
        return true
    }
}
```

ПРИЛОЖЕНИЕ В

```
// Ввод данных с окна
// ViewController.swift
// diaMed
//
// Created by Санжар Курмангалиулы on 02.04.2021.
//
```

```
import UIKit
import Firebase
```

```
class ViewController: UIViewController, UITextFieldDelegate{
    var ref: DatabaseReference!
```

```
    //Outlet текста
    override func viewDidLoad() {
        super.viewDidLoad()
    }
    //Logout
    @IBAction func logoutActoin(_ sender: Any) {
        do{
            try Auth.auth().signOut()
        }
        catch{
            print(error)
        }
    }
}
```

```
    //отправить данные кнопка
    @IBOutlet weak var rostField: UITextField!
    @IBOutlet weak var massaField: UITextField!
    @IBOutlet weak var izpalcaField: UITextField!
    @IBOutlet weak var izrukiField: UITextField!
    @IBOutlet weak var tipField: UISegmentedControl!
    @IBOutlet weak var titleLabel: UILabel!
    //Выбор типа
    @IBAction func segmentIp(_ sender: Any) {
        titleLabel.isHidden = false
```

```
        switch tipField.selectedSegmentIndex {
        case 0:
```

```

        titleLabel.text = "0 ТИП диабета (до 30 лет)"

    case 1:
        titleLabel.text = "1 ТИП диабета (от 30 лет)"

    default:
        print("Что то пошло не так (choiseSegment)")
    }
}
//кнопка отправить
@IBAction func buttonPalec(_ sender: UIButton) {
    let userID = Auth.auth().currentUser?.uid
    if let actualUserID = userID {
        ref = Database.database().reference()
        ref.child("users^(actualUserID)/info").setValue(["Iz
self.izpalcaField!.text,      "Iz      ruki":      self.izrukiField!.text,      palca":
self.tipField!.selectedIndex,  "Massa":      self.massaField!.text,      "tip":
self.rostField!.text])      "Rost":
        //self.tipField!.selectedIndex
        //ref.child("users").child(actualUserID).setValue(["Palec":
self.izpalcaField!.text])
        //self.ref.child("users").child(actualUserID).setValue(["Efternavn":
Efternavn.text!])

```

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Отзыв научного руководителя

Дипломная работа

Құрманғалиұлы Санжар

5B070300 – Информационные системы

Тема: Разработка WEB-приложения для
врачей-эндокринологов
«Оценка скрининга больных диабетом»

Дипломная работа представляет собой выпускную квалификационную работу по специальности 5B070300 – «Информационные системы». Пояснительная записка состоит из введения, 3 глав, заключения, списка использованных источников и приложения. Автор дипломной работы поставленные задачи полностью выполнил и показал владение современными технологиями в предметной области. Дипломная работа выполнена на достаточном профессиональном уровне и содержит все необходимые сведения для такого рода работ. К замечаниям следует отнести незначительные стилистические ошибки. Считаю, что дипломная соответствует требованиям предъявляемым к выпускным квалификационным работам по специальности 5B070300 – «Информационные системы». Автор работы Құрманғалиұлы Санжар заслуживает присвоения академической степени бакалавра.

Научный руководитель
Ассоц. проф., канд. техн. наук



Жумағалиев Б. И.

« 31 _ » Мая 2020 г.

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Құрманғалиұлы Санжар

Название: Разработка WEB-приложения для врачей эндокринологов «Оценка скрининга больных диабетом»

Координатор: Жумагалиев Биржан

Коэффициент подобия 1: 11.55

Коэффициент подобия 2: 4.02

Замена букв: 34

Интервалы: 0

Микропробелы: 13

Белые знаки: 0

После анализа Отчета подобия констатирую следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата.
В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

Замены букв было связано с применением общих принятых сокращений в виде аббревиатур на английском языке.....

...31 мая 2021.....

Дата


.....
Подпись Научного руководителя

**Протокол анализа Отчета подобия
заведующего кафедрой / начальника структурного подразделения**

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Құрманғалиұлы Санжар

Название: Разработка WEB-приложения для врачей эндокринологов «Оценка скрининга больных диабетом»

Координатор: Жумагалиев Биржан

Коэффициент подобия 1: 11.55

Коэффициент подобия 2: 4.02

Замена букв: 34

Интервалы: 0

Микропробелы: 13

Белые знаки: 0

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата.

В связи с чем, признаю работу самостоятельной и допускаю ее к защите;

обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора.

В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....
..... Заимствования являются добросовестными.....
.....
.....

Дата 31.05.2021  Заведующий кафедрой КОУХИ Сейлова Н.А.

Окончательное решение в отношении допуска к защите, включая обоснование:

.....
..... допускается к защите.....

Дата 31.05.2021  Заведующий кафедрой КОУХИ Сейлова Н.А.



Metadane

Tytuł

Разработка WEB-приложения для врачей-эндокринологов «Оценка скрининга больных диабетом»

Autorzy

Құрмағалиұлы Санжар

Promotor

Биржан Жумағалиев

Jednostka organizacyjna

ИКИИТ

Alerty

W tej sekcji znajdują się statystyki występowania w tekście zabiegów edytorskich, które mogą mieć na celu zaburzenie wyników analizy. Niewidoczne dla osoby zapoznającej się z treścią pracy na wydruku lub w pliku, wpływają na frazy porównywane podczas analizy tekstu (poprzez celowe błędy pisowni) w celu ukrycia zapożyczeń lub obniżenia wyników w Raporcie podobieństwa. Należy ocenić, czy zaznaczone wystąpienia wynikają z uzasadnionego formatowania tekstu (nadmrażliwość systemu), czy są celową manipulacją.

Znaki z innego alfabetu		32
Rozstrzelenia		0
Mikrospacje		13
Białe znaki		0
Parafrazy		74

Metryka podobieństw

Należy pamiętać, że wysokie wartości Współczynników nie oznaczają automatycznie plagiatu. Raport powinien zostać przeanalizowany przez kompetentną / upoważnioną osobę. Wyniki są uważane za wymagające szczegółowej analizy, jeśli WP 1 wynosi ponad 50%, a WP 2 ponad 5%.



WP1

25

Długość frazy dla WP 2



WP2

5576

Liczba słów



CYT

43201

Liczba znaków

Aktywne listy podobieństw

Uwagi wymagają szczególnie fragmenty, które zostały włączone do WP 2 (zaznaczone pogrubieniem). Użyj linku "Pokaż w tekście" i zobacz, czy są to krótkie frazy rozproszone w dokumencie (przypadkowe podobieństwa), skupione wokół siebie (parafraza) lub obszerne fragmenty bez wskazania źródła (tzw. "kryptocytaty").

10 najdłuższych fragmentów

Kolor w tekście

LP	TYTUŁ LUB ADRES URL ŹRÓDŁA (NAZWA BAZY)	IDENTYCZNYCH SŁÓW (FRAGMENTÓW)	
1	http://www.ohandroid.com/google-access.html	79	1.42 %
2	https://skillbox.ru/media/code/mvym_proektirovanie_prilozheniy_dlya_windows/	55	0.99 %
3	https://igorosa.com/what-is-backend-development/	32	0.57 %
4	https://studbooks.net/2249813/informatika/arhitektura	31	0.56 %
5	СИСТЕМЫ УДАЛЕННОЙ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ HELPDESK Ахмедов Бекмурат Мамадиярулы 6/4/2019 Kazakh University of Economics, Finance and International Trade (KUEFIT) (Кафедра «Информационные системы и технологии»)	27	0.48 %

6	https://igorosa.com/what-is-backend-development/	24	0.43 %
7	http://www.ohandroid.com/google-access.html	24	0.43 %
8	https://studbooks.net/2249813/informatika/arhitektura	24	0.43 %
9	https://studbooks.net/2249813/informatika/arhitektura	23	0.41 %
10	https://studbooks.net/2249813/informatika/arhitektura	23	0.41 %

z bazy RefBooks (0.00 %) 

LP	TYTUŁ	IDENTYCZNYCH SŁÓW (FRAGMENTÓW)	
----	-------	--------------------------------	--

z bazy macierzystej (0.00 %) 

LP	TYTUŁ	IDENTYCZNYCH SŁÓW (FRAGMENTÓW)	
----	-------	--------------------------------	--

z Programu Wymiany Baz (1.51 %) 

LP	TYTUŁ	IDENTYCZNYCH SŁÓW (FRAGMENTÓW)	
1	СИСТЕМЫ УДАЛЕННОЙ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ HELPDESK Ахмедов Бекмурат Мамадиярулы 6/4/2019 Kazakh University of Economics, Finance and International Trade (KUEFIT) (Кафедра «Информационные системы и технологии»)	84 (6)	1.51 %

z Internetu (10.04 %) 

LP	ADRES URL ŹRÓDŁA	IDENTYCZNYCH SŁÓW (FRAGMENTÓW)	
1	https://studbooks.net/2249813/informatika/arhitektura	248 (16)	4.45 %
2	http://www.ohandroid.com/google-access.html	115 (3)	2.06 %
3	https://igorosa.com/what-is-backend-development/	109 (8)	1.95 %
4	https://skillbox.ru/media/code/mvvm_proektirovanie_prilozheniy_dlya_windows/	55 (1)	0.99 %
5	https://pastebin.com/jEx2CHi0	17 (2)	0.30 %
6	https://kolmogorov.pro/what-is-firebase-chto-takoe	16 (1)	0.29 %

Lista zaakceptowanych fragmentów (brak zaakceptowanych fragmentów)

LP	TREŚĆ	IDENTYCZNYCH SŁÓW (FRAGMENTÓW)	
----	-------	--------------------------------	--