

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

Боранбаев Даулет Құрманалыұлы

Разработка и внедрение систем взаимодействия с клиентами банка с использованием современных информационных технологий

ДИПЛОМНАЯ РАБОТА

Специальность 5В070300 – Информационные системы

Алматы, 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой _____

(наименование кафедры)

(ученая степень, звание)

подпись _____ Ф.И.О.

« _____ » _____ 20__ г.

ДИПЛОМНАЯ РАБОТА

На тему: «Разработка и внедрение систем взаимодействия с клиентами банка с использованием современных информационных технологий»

Специальность 5В070300 – Информационные системы

Выполнил:

Боранбаев Д.К.

Рецензент

Научный руководитель

к.т.н., доцент

_____ Косников В.А.

« _____ » _____ 20__ г.

« _____ » _____ 20__ г.

Алматы, 2020

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 _____

(наименование кафедры)

(ученая степень, звание)

подпись

Ф.И.О.

« _____ » _____ 20__ г.

ЗАДАНИЕ

на выполнение дипломной работы

Обучающемуся: Боранбаев Даулет Курманалыұлы

Тема: Разработка и внедрение систем взаимодействия с клиентами банка с использованием современных информационных технологий

Утверждена приказом Ректора Университета № 762–б от 27.01.2020г.

Срок сдачи законченной работы - 27.05.2020г.

Исходные данные к дипломной работе: результаты преддипломной практики, результат обзора современного состояния по данной теме, сбор теоретического материала.

Краткое содержание дипломной работы:

- а) Современное состояние использования банковских технологий;
- б) Структура и алгоритмы функционирования системы;
- в) Информационное и программное обеспечение системы;
- г) Внедрение системы;

Рекомендуемая основная литература: *из 11 наименований*

ГРАФИК

подготовки дипломной работы (проекта)

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Современное состояние использования банковских технологий	17.02.2020г.	
Структура и алгоритмы функционирования системы	09.03.2020г.	
Информационное и программное обеспечение системы	30.03.2020г.	
Внедрение системы	13.04.2020г.	

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 и внедрение систем взаимодействия с клиентами банка с использованием современных информационных технологий»	Косников В.А., кандидат технических наук, доцент		
Нормоконтролер	Бауыржан М.Б., тьютор		
Программная часть	Кабдуллин М.А., ассистент		

Научный руководитель: _____

Косников В.А.

Задание принял к исполнению обучающийся _____

Боранбаев Д.К.

Дата

" ____ " _____ 20__ 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: Боранбаев Д.К.

Название: Разработка и внедрение систем взаимодействия с клиентами банка с использованием современных информационных технологий

Координатор: Вячеслав Косников

Коэффициент подобия 1: 14,1

Коэффициент подобия 2: 9,3

Замена букв: 1

Интервалы: 0

Микропробелы: 0

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:

.....

.....

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: Боранбаев Д.К.

Название: Разработка и внедрение систем взаимодействия с клиентами банка с использованием современных информационных технологий

Координатор: Вячеслав Косников

Коэффициент подобия 1:14,1

Коэффициент подобия 2:9,3

Замена букв:1

Интервалы:0

Микропробелы:0

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, работа признается самостоятельной и допускается 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, работа не допускается к защите.

Обоснование:

.....

.....

Дата

.....

*Подпись заведующего кафедрой/
начальника структурного подразделения*

Окончательное решение в отношении допуска к защите, включая обоснование:

.....
.....
.....
.....
.....

.....
Дата

.....
*Подпись заведующего кафедрой/
начальника структурного подразделения*

НАЗВАНИЕ:
Разработка и внедрение систем взаимодействия с клиентами банка с использованием современных информационных технологий

АВТОР:
Боранбаев Д.К.

ПОДРАЗДЕЛЕНИЕ:
ИКиИТ

НАУЧНЫЙ РУКОВОДИТЕЛЬ:
Вячеслав Косников

ДАТА ЗАГРУЗКИ ДОКУМЕНТА:
2020-05-19 07:42:53

ЧИСЛО ПРОВЕРЕК ДОКУМЕНТА: 1

ПРОПУЩЕННЫЕ ВЕБ-СТРАНИЦЫ: 1

Уровень заимствований

Обратите внимание! Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.



Предупреждение и сигналы тревоги

В этом разделе вы найдете информацию, касающуюся манипуляций в тексте, с целью изменить результаты проверки. Для того, кто оценивает работу на бумажном носителе или в электронном формате, манипуляции могут быть невидимы (может быть также целенаправленное вписывание ошибок). Следует оценить, являются ли изменения преднамеренными или нет.

Замена букв Использование символов из другого алфавита - может указывать на способ обойти систему, поэтому следует установить их использование.	1	показать в тексте
Интервалы Количество увеличенного расстояния между буквами (просим определить является ли расстояние имитацией пробела, так как исходя слова могут быть написаны слитно).	0	показать в тексте
Микропробелы Количество пробелов с нулевым размером - необходимо проверить влияют ли они на неправильное разделение слов в тексте.	0	показать в тексте
Белые знаки Количество символов, выделенных белым цветом, пожалуйста, проверьте не используются ли белые символы вместо пробела, соединя слова (в отчете подобия система изменяет автоматически цвет букв в черный, чтобы их сделать видимыми).	0	показать в тексте

Заимствования по списку источников

Просмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и посмотрите, являются ли выделенные фрагменты повторяющимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

10 самых длинных фраз (8,54 %)

Десять самых длинных фрагментов найденных во всех доступных ресурсах.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	АВТОР	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ
1	https://novainfo.ru/article/6500		129 1,73 %
2	https://novainfo.ru/article/6500		98 1,32 %
3	http://www.brownbear.ru/goods/31384.html		93 1,25 %
4	https://novainfo.ru/article/6500		70 0,94 %
5	https://stackoverflow.com/questions/55602994/spring-boot-and-angular-json-parse-error-can-not-deserialize-in-stance-of-java		53 0,71 %
6	https://megalektsii.ru/s18663t8.html		50 0,67 %
7	https://ppt-online.org/50299		42 0,56 %
8	https://novainfo.ru/article/6500		37 0,50 %
9	https://novainfo.ru/article/6500		33 0,44 %

10	https://stackoverflow.com/questions/55602994/spring-boot-and-angular-json-parse-error-can-not-deserialize-in-stance-of-java	31	0,42 %
----	---	----	--------

из базы данных RefBooks (0,00 %)

Все фрагменты найдены в базе данных RefBooks, которая содержит более 3 миллионов текстов от редакторов и авторов.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	АВТОР	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (КОЛИЧЕСТВО ФРАГМЕНТОВ)
---------------------	----------	-------	---

ЗАИМСТВОВАНИЙ НЕ НАЙДЕНО

из домашней базы данных (0,00 %)

Все фрагменты найдены в базе данных вашего университета.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	АВТОР	ДАТА ИНДЕКСАЦИИ	ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
---------------------	----------	-------	--------------------	---------------------------------

ЗАИМСТВОВАНИЙ НЕ НАЙДЕНО

из программы обмена базами данных (0,51 %)

Все фрагменты найдены в базе данных других университетов.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ НАЗВАНИЕ БАЗЫ ДАННЫХ	АВТОР	ДАТА ИНДЕКСАЦИИ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (КОЛИЧЕСТВО ФРАГМЕНТОВ)	
1	Документ из базы НЭУ 527c97a2-4fe8-427c-93f4-071dc0a80d1d.docx NARXOZ (NEU) (Information Technology Center)	na	2015-11-05	38 (2)	0,51 %

из интернета (13,63 %)

Все фрагменты найдены в глобальных интернет-ресурсах открытого доступа.

ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	https://novainfo.ru/article/6500	392 (7)	5,26 %
2	https://stackoverflow.com/questions/55602994/spring-boot-and-angular-json-parse-error-can-not-deserialize-instance-of-java	175 (7)	2,35 %
3	http://www.brownbear.ru/goods/31384.html	93 (1)	1,25 %
4	https://megalektsii.ru/s18663t8.html	50 (1)	0,67 %
5	https://ppt-online.org/50299	42 (1)	0,56 %
6	http://expo-itsecurity.ru/upload/iblock/a74/nekirivp%20ivyflhsbrf%20-%20kznisy.%20twathq%20b%20vmdmvzbe.pdf	39 (2)	0,52 %
7	https://grokez.com/spring-framework/spring-boot/spring-security-jwt-authentication-postgresql-restapis-springboot-spring-mvc-spring-jpa	34 (2)	0,46 %
8	https://otherreferats.allbest.ru/bank/00509957_0.html	33 (2)	0,44 %
9	https://habr.com/ru/post/488054/	27 (2)	0,36 %
10	http://litra.studentochka.ru/s?l=%C8%ED%F4%EE%F0%EC%E0%F6%E8%EE%ED%ED%E0%FF+%F1%E5%E7%EE%EF%E0%F1%ED%EE%F1%F2%FC+	22 (2)	0,30 %
11	https://cyberpedia.su/4x6855.html	19 (2)	0,26 %
12	https://stackoverflow.com/questions/59173155/spring-security-401-unauthorized-looks-like-problem-with-filter-or-permitall	18 (2)	0,24 %
13	http://docplayer.ru/69307891-Upravlencheskie-nauki-v-sovremennom-mire-management-sciences-in-the-modern-world.html	18 (1)	0,24 %
14	https://gitlab.com/morawiec/mgr-backend/commit/250a2664fdaa6f32d70a339ea624fb2ef4e41896	16 (2)	0,21 %
15	https://referatbank.ru/market/referat/i/203435/referat-analiz-biznes-processa.html	14 (1)	0,19 %
16	https://habr.com/ru/post/38730/	12 (1)	0,16 %
17	https://studopedia.su/6_57772_primeri-ogranicheniy.html	11 (1)	0,15 %

АНДАТПА

Дипломдық жұмыстың мақсаты қазіргі заманғы ақпараттық технологияларды қолдана отырып, банк клиенттерімен өзара әрекеттесу жүйесін құру болып табылады. Дипломдық жұмысты бастамас бұрын ұқсас заманауи жүйелер зерттелді. Жүйені жасау құрылғысы ретінде Angular және Spring фреймворк (шенбер), бағдарламалау тілдері – Java және JavaScript, онымен қоса RestAPI технологиясы және де басқада құрылғылар таңдалды. Сондай-ақ, жүйенің функционалдық құрылымы құрылды, функционалдық есептерді шешудің алгоритмдері құрастырылып, еңгізілді, ақпараттық мәліметтер базасы сипатталды. Берілген жұмыстың нәтижелері Екінші деңгейдегі Банктерде клиенттермен өзара әрекеттесудің тиімді жүйесін еңгізу үшін пайдалануға болады.

АННОТАЦИЯ

Целью дипломной работы является разработка собственной системы взаимодействия с клиентами банка с использованием современных информационных технологий. Перед началом работы были изучены и проанализированы уже существующие аналогичные системы. В качестве инструмента разработки системы были использованы фреймворки Angular и Spring, языки программирования Java и JavaScript, также технологии RestAPI и др. Кроме того, была создана функциональная структура системы, разработаны и реализованы алгоритмы решения функциональных задач, описана Информационная База Данных. Результаты этой работы могут быть использованы для внедрения эффективной системы взаимодействия с клиентами в Банках Второго Уровня.

SUMMARY

The purpose of the diploma work is the development system of interaction with bank customers using modern information technologies. Before starting work, already existing similar systems were studied and analyzed. The Angular and Spring frameworks, Java and JavaScript programming languages, RestAPI technologies, etc. were used as a system development tool. In addition, a functional structure of the system was created, algorithms for solving functional problems were developed and implemented, and the Information Database was described. The results of this work can be used to implement an effective system of interaction with customers in Second Level Banks.

СОДЕРЖЕНИЕ

	ВВЕДЕНИЕ.....	14
1	Современное состояние использования банковских технологий.....	15
1.1	Обзор и анализ информационных технологий, используемых в банковской сфере.....	15
1.2	Анализ инфраструктурных решений, используемых в банковской сфере. Выявление проблем.....	16
1.3	Постановка задачи дипломирования.....	19
2	Структура и алгоритмы функционирования системы.....	20
2.1	Функциональная структура системы.....	20
2.2	Обобщенный алгоритм функционирования.....	22
2.3	Алгоритм авторизации и аутентификации клиента.....	25
2.4	Алгоритм восстановления пароля.....	26
2.5	Алгоритм работы с Базой Данных.....	27
3	Информационное и программное обеспечение системы.....	28
3.1	Описание информационного обеспечения.....	28
3.2	Системное программное обеспечение.....	31
3.3	Инструментальное программное обеспечение.....	32
3.4	Прикладное программное обеспечение.....	33
3.5	Описание контрольного примера.....	36
4	Внедрение системы.....	39
4.1	Описание инфраструктуры системы.....	39
4.2	Перспективы развития.....	41
	ЗАКЛЮЧЕНИЕ.....	43
	СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	44
	ПРИЛОЖЕНИЕ А.....	45
	ПРИЛОЖЕНИЕ Б.....	48
	ПРИЛОЖЕНИЕ В.....	50
	ПРИЛОЖЕНИЕ Г.....	52

ВВЕДЕНИЕ

В век инноваций использование информационных технологий стало объективной необходимостью. Одна из сфер, где их значение очень велико – финансовая сфера. И даже можно с уверенностью сказать, что процесс информатизации банковской деятельности продолжится и в дальнейшем. В банковском секторе в ближайшем будущем будут преобладать тенденции к повышению качества и надежности предлагаемых продуктов и услуг, увеличению скорости проведения расчетных операций, организации электронного доступа клиентов к банковским продуктам. Это обусловлено, прежде всего, стремлением банков к достижению конкурентных преимуществ на финансовых рынках.

В данной дипломной работе была разработана и полностью описана система взаимодействия с клиентами Банка. Система была разработана при помощи объектно-ориентированного языка программирования java и мультипарадигменного языка Javascript, а также с помощью различных технологий и архитектур. Работа состоит из 4-х глав.

В первой главе был проведен обзор и анализ информационных технологий, используемых в банковской сфере. Были рассмотрены существующие системы в Банках, для чего и где они используются, и, конечно же, какими функциями обладают. Также были проанализированы инфраструктурные решения в банковской сфере, выявлены проблемы, с которыми все чаще сталкиваются банки Второго Уровня РК. В результате анализа проблем была определена актуальность данной темы и были определены задачи, которые нужно решить в дипломной работе.

Во второй главе была описана структура функционирования системы, также разработаны и реализованы алгоритмы функционирования системы, описаны решения функциональных задач, реализация работы с БД.

В третьей главе идет описание структуры информационного и программного обеспечения. Была спроектирована информационная база данных и организована структура программного обеспечения.

В четвертой главе приведена и описана вся инфраструктура системы, организация локальной сети и т.д. Также были оценены перспективы развития системы в ближайшем будущем.

1 СОВРЕМЕННОЕ СОСТОЯНИЕ ИСПОЛЬЗОВАНИЯ БАНКОВСКИХ ТЕХНОЛОГИЙ

1.1 Обзор и анализ информационных технологий, используемых в банковской сфере

Информационная банковская технология - процесс преобразования банковской информации на основе методов сбора, регистрации, передачи, хранения и обработки данных в целях обеспечения подготовки, принятия и реализации управленческого решения с использованием средств персональной и вычислительной техники.

Банковские технологии неразрывно связаны с информационными технологиями, которые обеспечивают комплексную автоматизацию бизнеса.[Тютюнник А.В., Шевелев А.С. Информационные технологии в банке, 2003]. Современные банковские технологии как инструмент поддержки и развития банковского бизнеса создаются на базе ряда основополагающих принципов:

- модульный принцип построения, позволяющий легко конфигурировать системы под конкретный заказ с последующим наращиванием;
- открытость технологий, способных взаимодействовать с различными внешними системами, обеспечивать выбор программно-технической платформы и переносимость ее на другие аппаратные средства;
- гибкость настройки модулей банковской системы и адаптация их к потребностям и условиям конкретного банка;
- масштабируемость, предусматривающая расширение и усложнение функциональных модулей системы по мере развития бизнес-процессов.
- многопользовательский доступ к данным в реальном времени и реализация функций в едином информационном пространстве;
- моделирование банка и его бизнес-процессов, возможность алгоритмических настроек бизнес-процессов;
- непрерывное развитие и совершенствование системы на основе ее реинжиниринга бизнес-процессов.

В настоящее время можно выделить три основных направления развития: система «Клиент-банк», интернет-банкинг и мобильный банкинг. Разберем каждое направление по отдельности:

С помощью системы **«Клиент-банк»** клиенты банка могут совершать различные операции из дома или из офиса: управление счетом, получение информации о состоянии счетов и другой банковской информации, проведение платежей и оплата услуг с расчетных и других счетов и с пластиковых карт, а также проведение других операций.

Мобильный банкинг – получение банковских услуг непосредственно с помощью мобильного телефона или ноутбука при использовании технологии

беспроводного доступа. Такая технология позволяет передавать информацию интернет-сайтов на мобильные телефоны с функцией выхода в Интернет. Эта система предоставляет еще большую свободу доступа. Среди потребителей банковских услуг при помощи мобильного телефона первое место занимают Скандинавские страны.

Наиболее перспективным направлением развития банковских информационных технологий является **интернет-банкинг**. Развитие систем дистанционного обслуживания привело к созданию различных по объему и формам предоставления банковских услуг систем: «Интернет—Банк», «Интернет—Клиент», домашний банк, телебанк, мобильный банк или WAP-сервис(WAP- Wireless Application Protocol – протокол беспроводной передачи данных). С помощью этих систем выполняются практически любые, кроме кассового обслуживания, требования клиентов банка.

1.2 Анализ инфраструктурных решений используемых в банковской сфере. Выявление проблем

Инфраструктура информационных технологий (ИТ-инфраструктура) - это организационно-техническое объединение программных, вычислительных и телекоммуникационных средств, связей между ними и эксплуатационного персонала, обеспечивающее предоставление информационных, вычислительных и телекоммуникационных ресурсов, возможностей и услуг работникам (подразделениям) предприятия (организации), необходимых для осуществления профессиональной деятельности и решения соответствующих бизнес - задач.

В основном, банковская система подразделяется на несколько блоков в зависимости от направлений и выполняемых функций [1, Тютюнник А.В., Шевелев А.С. ,Информационные технологии в банке, 2003]. Далее подробно рассмотрим **Блок Информационных технологий**(в дальнейшем БИТ).

БИТ(Блок Информационных Технологий) включает в себя несколько департаментов и управлений, а именно(рис 1.1.):

- Департамент карточных технологий.
- Департамент развития ИТ.
- Управление гибкой методологии.
- Управление ИТ- архитектуры.
- Управление поддержки пользователей и сопровождение банковских систем.
- Управление сопровождения инфраструктуры.
- Управление технологического развития.

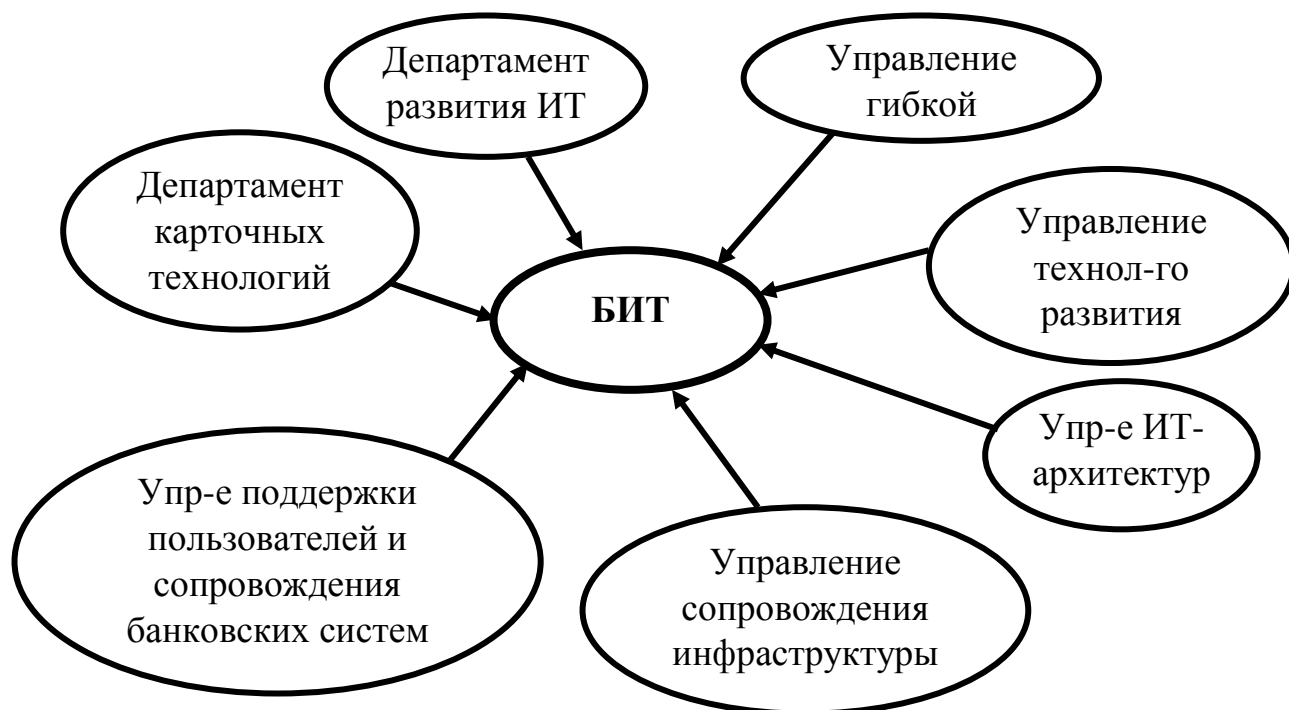


Рисунок 1.1 – Блок Информационных Технологий

В свою же очередь, каждый департамент имеет свои внутренние подразделения. Департамент карточных технологий включает в себя 3 управления, это:

- Управление администрирования и разработки карточной платформы.
- Управление мониторинга и администрирования терминальной сети.
- Управление эмиссии и развития карточной платформы.

Департамент развития Информационных технологий содержит в себе 4 управления, а именно:

- Управление развития бэк-офисных систем.
- Управление развития фронт-офисных систем.
- Управление системного анализа.
- Управление тестирования.

Также Управление поддержки пользователей и сопровождения банковских систем, в свою очередь, содержит 3 отдела, это:

- Отдел SD (Service Desk).
- Отдел сопровождения банковских систем.
- Отдел технической поддержки(Hard).

И управление сопровождения инфраструктуры содержит в себе:

- Отдел аппаратного обеспечения и баз данных.
- Отдел сети и телефонии.
- Отдел системного администрирования.

В настоящее время банки Республики Казахстан сталкиваются с рядом проблем, связанных с информационными технологиями. Разберем каждый из них.

Одной из проблем банков является отсутствие CRM системы, так как покупка готовой системы стоит не малых денег, а для того, чтобы разработать сырую систему взаимодействия с клиентами нужны квалифицированные специалисты (разработчики, тестировщики и т.д.) и огромные ресурсы. Множество казахстанских фирм или банков предпочитают сразу покупать готовую CRM – систему, но лучше всего разработать свою систему. Потому что лучше создать систему под свой бизнес, которая будет решать нужные задачи, чем подстраивать бизнес под существующий шаблон. Но разработка CRM системы с нуля стоит дороже, чем готовая CRM. Хотя чем каждый раз дорабатывать готовую систему, лучше разработать свою программу в соответствии с задачами бизнеса.

Также можно отметить отсутствие систем взаимодействия с клиентами (мобильное приложение, Интернет-банкинг) с дружелюбным интерфейсом и, где могут быть предоставлены все услуги.

Немаловажной проблемой является аппаратное обеспечение IT – инфраструктуры. А именно установление различных операционных систем усложняет работу системы и замедляет обработку данных. А решением этой проблемы является – Облачные технологии (cloud technology). Аппаратное обеспечение может быть сильно упрощено при обработке данных и хранении информации в удаленных центрах данных. Облачные технологии облегчают обеспечение доступа к данным Банка как для клиентов, так и для собственных сотрудников, находящихся вне офиса, но имеющих возможность подключиться через Интернет.

В последние годы в банковской деятельности обострилась проблема обеспечения безопасности данных. Ценность банковской информации предъявляет особые требования к защите данных от несанкционированного доступа, в том числе к контролю управления процессами, изменяющими состояние данных. Банки, чтобы не потерять своих клиентов, в основном, умалчивают насчет кибер-атак, но в малых процентах кибер-атаки присутствуют в каждом банке РК. Но в любом случае для обеспечения безопасности хранения данных системы должны соответствовать требованиям государственных стандартов.

Ну и наконец, можно отметить проблемы хранения данных. В крупных компаниях и банках, где клиентская база является очень большой, возникают ряд проблем с хранением данных.

1.3 Постановка задачи дипломирования

За последние 5 лет большая часть банков РК, предоставляющие свои услуги клиентам, сталкиваются с вышеперечисленными проблемами. И для

решения этих проблем требуются разного рода ресурсы. В настоящее время основным ресурсом для решения многих проблем являются деньги, но банки не любят огромных расходов и поэтому нанимают квалифицированных специалистов для разработки и внедрения систем, которые бы помогли решить эти проблемы. В результате анализа существующих систем взаимодействия с клиентами банков и нерационального использования в данной дипломной работе прилагается : разработка собственной системы взаимодействия с клиентами Банка и интегрирование его в информационную систему Банка.

В данной дипломной работе будет разработана и внедрена система взаимодействия с клиентами Банка, а именно веб- приложение с удобным пользовательским интерфейсом и включающим в себя услуги, которые предоставляются в отделениях или филиалах банка. Также будут предложены идеи в архитектуре Информационных технологий в соответствии с государственными стандартами и современными технологиями.

Задачи, которые должны быть решены в результате дипломной работы:

- Разработка и описание функциональной структуры системы с использованием UML – диаграммы.
- Разработка обобщенного алгоритма функционирования и ее описание с помощью UML-диаграмм.
- Разработка алгоритмов решения функциональных задач.
- Разработка описание структуры Информационной Базы Данных с помощью ER –диаграммы.
- Разработка структуры программного обеспечения и реализация всех алгоритмов.
- Внедрение результатов дипломной работы в информационную систему Банка.

Полная разработка самого веб-приложения будет состоять из следующих этапов:

- определение целей и задач приложения (т.е. какие задачи будет решать и какими функциями обладать);
- разработка структуры страниц;
- разработка дизайн- макетов;
- html-вёрстка;
- программирование и контроль качества;
- тестирование, запуск и сопровождение.

2 СТРУКТУРА и АЛГОРИТМЫ ФУНКЦИОНИРОВАНИЯ СИСТЕМЫ

2.1 Функциональная структура системы

Функциональная структура ИС — совокупность функциональных подсистем, комплексов задач и процедур обработки информации, реализующих функции системы управления. В хозяйственной практике производственных и коммерческих объектов типовыми видами деятельности, которые определяют функциональный признак классификации информационных систем, являются: производственная, маркетинговая, финансовая, кадровая.

Указанные направления деятельности определили типовой набор информационных систем: производственные системы; системы маркетинга; финансовые и учетные системы; системы кадров (человеческих ресурсов); другие ИС, которые выполняют вспомогательные функции в зависимости от деятельности фирмы.

Основные преимущества функциональной структуры:

- высокая компетентность специалистов, отвечающих за осуществление конкретных функций;
- освобождение линейных менеджеров от решения многих специальных вопросов и расширение их возможностей по оперативному управлению производством;
- использование в работе консультаций опытных специалистов, уменьшение потребности в специалистах широкого профиля;
- снижение риска ошибочных решений;
- исключение дублирования в выполнении управленческих функций.

В данной системе имеются 2 роли пользователей, это администратор (сотрудники Банка, Service Desk) и клиент. У каждой роли имеются свои возможности, т.е. функции, которые доступны тому или иному пользователю. Например, клиент может зарегистрироваться, авторизоваться, восстановить пароль, осуществление платежей, осуществление переводов, а также и связь со Службой Поддержки (Service Desk).

Так на следующем рисунке 2.1 описана функциональная структура системы (все функции, которые доступны и связь между клиентами с сотрудниками Банка, службой Service Desk и т.д.):

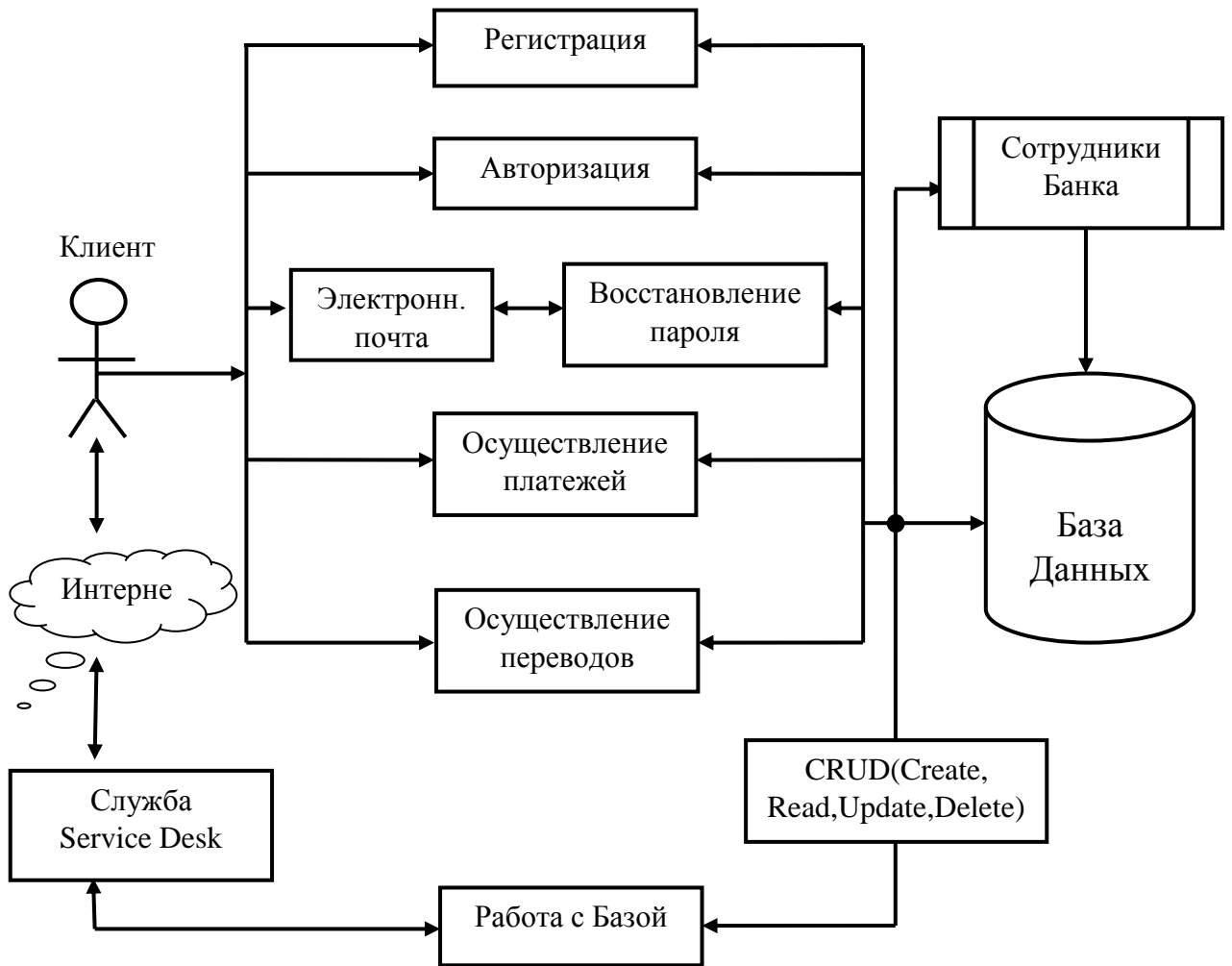


Рисунок 2.1 – Функциональная структура системы

При помощи UML – диаграммы на рисунке 2.2 представлена общая схема работы системы, а именно при помощи диаграммы активности. В настоящий момент все больше обороты набирает описание объектного моделирования при помощи UML (Unified Modeling Language). Это унифицированный язык, который предназначен для того, чтобы можно было сделать графическое описание разработки программного обеспечения, для системного проектирования, для моделирования бизнес- процессов. На рис. 2.2 представлена диаграмма активности для описания общей работы системы:

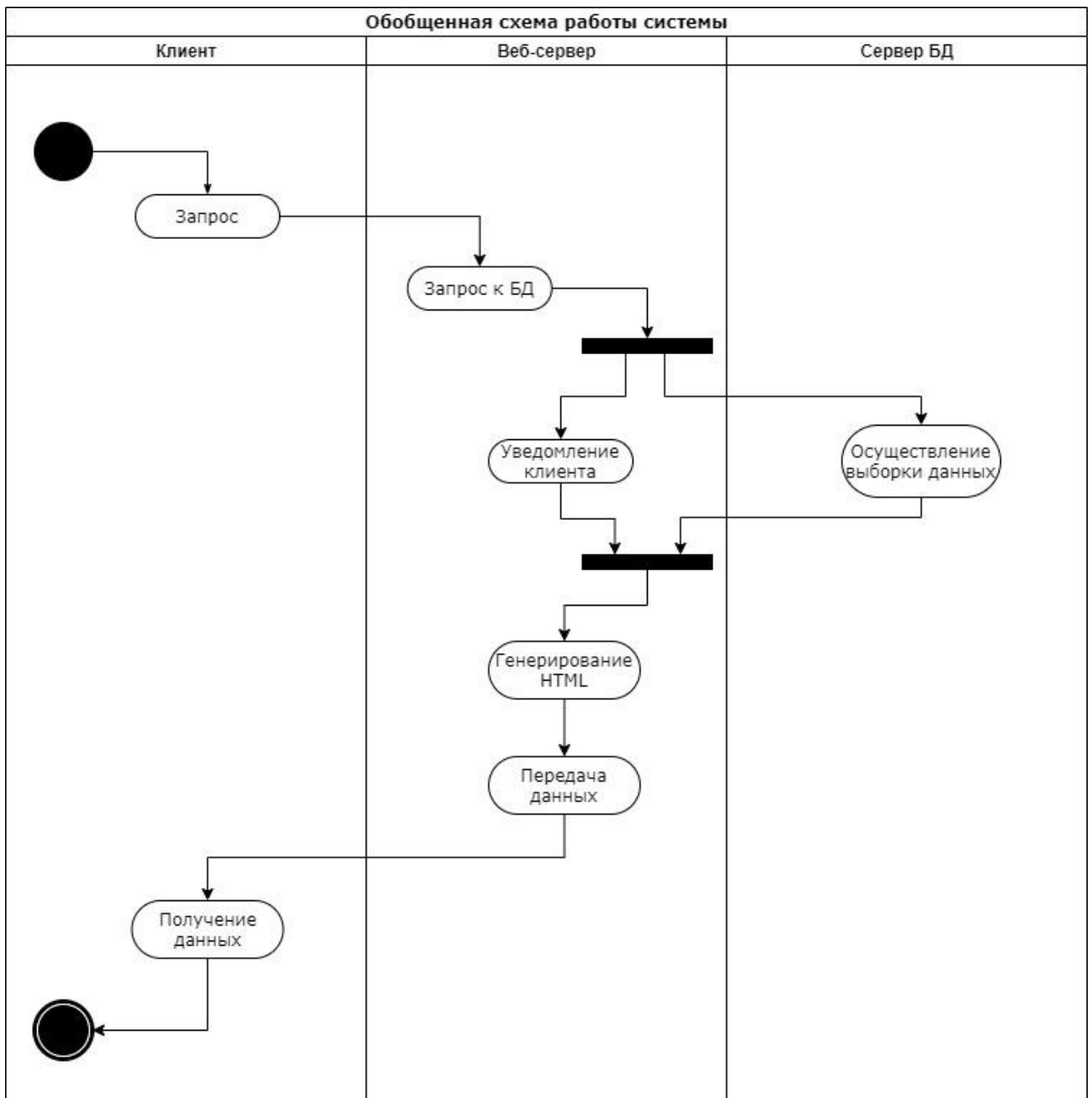


Рисунок 2.2 – Диаграмма активности обработки запросов пользователей

2.2 Обобщенный алгоритм функционирования

Любое веб-приложение состоит из 2 частей это: Back-end – серверная часть и Front-end – клиентская часть[2, Яков Файн и Антон Моисеев, Angular и TypeScript. Сайтостроение для профессионалов, 2018]. Т.е. используется клиент-серверная технология. «Клиент-сервер» - это архитектура, где для снижения затрат, избегания дублированных кодов все выполняемые задачи и нагрузки поделены между поставщиками услуг, т.е. back-end - часть, называемыми серверами, и заказчиками услуг(front-end-часть) называемыми клиентами. Зачастую 2 части (back и front) разрабатываются на отдельных

вычислительных машинах и взаимодействуют между собой через специальную сеть с помощью сетевых протоколов. Но в век новейших технологий эти 2 части уже разрабатываются одновременно на одной вычислительной машине для снижения эксплуатационных расходов, бесперебойной работы, надежности и т.д.

Но поскольку сейчас очень много активных клиентов, соответственно, и запросов, поступающих от клиентов, очень много, Back-end должен был размещен на специальной вычислительной машине, где производительность очень высокая и соответствует современным Государственным стандартам.

На следующем рисунке 2.3 показана UML - диаграмма деятельности, т.е. последовательность действий, события, инициирующие действия или являющиеся конечным результатом, условия расширения сценария:

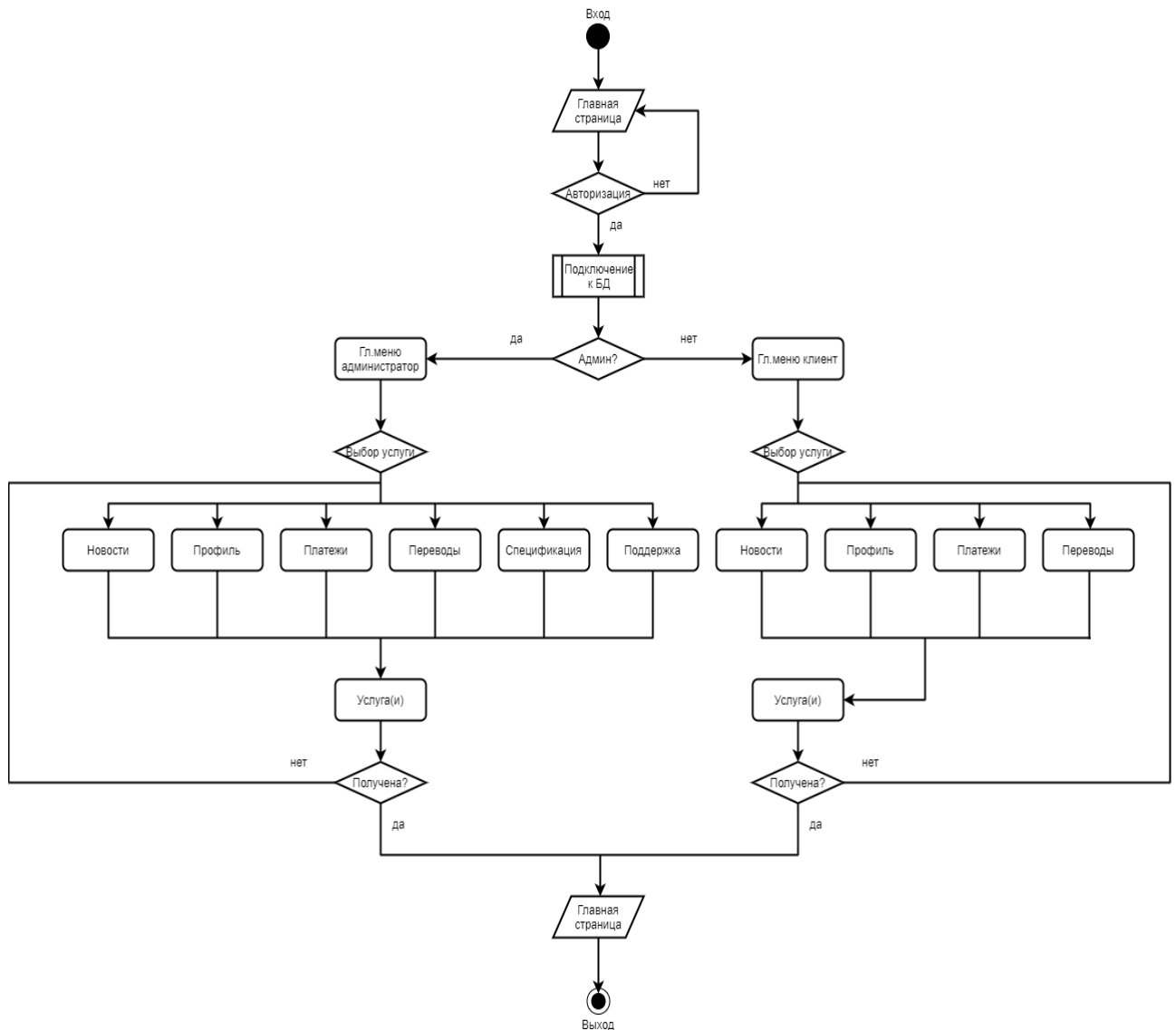


Рисунок 2.3 –Диаграмма деятельности пользователей

Общий алгоритм функционирования системы состоит из 2 этапов:
Первый этап (серверная часть):

- Шаг 1. Начало;
- Шаг 2. Создание моделей и описание;
- Шаг 3. Создание контроллеров и репозиторий, которые будут использованы в контроллерах для упрощения многих действий и во избежание дублирования кодов;
- Шаг 4. Конфигурация почты;
- Шаг 5. Разработка системы безопасности (авторизация и аутентификация);
- Шаг 6. Конец.

Второй этап (клиентская часть):

- Шаг 1. Начало;
- Шаг 2. Создание главной страницы;
- Шаг 3. Генерирование основных и второстепенных компонентов;
- Шаг 4. Генерирование сервисов;
- Шаг 5. Соединение 2 приложений (back-end и front-end);
- Шаг 6. Конец.

В данной дипломной работе разработка была начата с написания серверной части(back-end). Сам back-end состоит из следующих частей:

- Модели – это классы, которые хранятся в таблицах.
- Репозитории – это интерфейсы, которые используют для взаимодействия с данными(удалить, добавить, изменить и т.д.).
- Контроллеры – это специальные классы, которые предназначены для обработки запросов от клиента и возвращения результатов.
- Mail – конфигурация почты.
- Security – безопасность и надежность системы посредством авторизации и аутентификации.

Далее в клиентской части основной страницей является – App.component.html, где при изменении URL – страницы header и hooter остаются неизменными, а остальные части страницы меняются[Приложение В]. А за изменение router - outlet по URL отвечает App-routing.module.ts формата TypeScript. После написания основной страницы идет генерирование всех компонентов и, соответственно, генерирование сервисов, отвечающих за функционирование каждой вкладки. Т.е. например news.service.ts, password.service.ts, payment.service.ts, specification.service.ts и т.д. Следует заметить, что все сервисы на Front-end были разработаны в формате ts-TypeScript. И конечно же, каждый компонент разрабатывается на 3 языках : ts-TypeScript – язык программирования, представленный компанией Microsoft, используется для расширения возможностей языка JavaScript, html – HyperText Markup Language – язык «гипертекстовой разметки» и css- Cascading Style Sheets – язык таблиц стилей. Back-end и Front-end располагаются на разных портах. Т.е. front-end на порте localhost:4200 и back-end на порте localhost:8080. Для того, чтобы соединить эти 2 приложения, front-end localhost:4200 обращается к back-end localhost:8080 через http module.

2.3 Алгоритм авторизации и аутентификации клиента

Для безопасности системы любая компания или любая организация использует авторизацию и аутентификация клиента.[3,Ю.А.Родичев, Нормативная база и стандарты в области информационной безопасности, 2017]. Под авторизацией клиента понимается предоставление определенному клиенту права на выполнение каких-либо действий и ,конечно же, проверка данных прав при попытке совершения каких-либо действий. А аутентификация – это одно из важнейших составляющих основы безопасности каждой системы, посредством проверки и контроля подлинности данных о клиенте сервером. Существуют различные виды аутентификации, такие как комбинированные, биометрические, парольные и т.д. В зависимости от вида веб-приложения и предоставляемых услуг каждая система безопасности устанавливает соответствующий вид аутентификации. В настоящее время все большую популярность набирает метод аутентификации – биометрический, который можно использовать в любое время (TouchID, FaceID и др.).Но в любом случае при использовании биометрического метода аутентификации в обязательном порядке используется и парольный вид, т.к. разработчики учитывают любые факторы, такие как операции, различные гематомы, которые привели к изменению биометрических данных клиента, вследствие чего, надо будет использовать проверенный метод аутентификации - парольный.

В данной системе будет разработан метод аутентификации – парольный. Т.е. аутентификация будет проходить посредством постоянного пароля при регистрации пользователя. Также, в случае когда клиент забывает пароль, можно будет восстановить пароль с помощью электронной почты, которая будет указана при регистрации.

Алгоритм функционирования авторизации и аутентификации:

Шаг 1.Начало.

Шаг 2. Создание модели User.

Шаг 3. Добавление репозитория к модели User.

Шаг 4. Добавление конфигурации в WebSecurityConfig.

Шаг 5. Создание и добавление контроллера auth.

Шаг 6. Создание сервиса token storage.

Шаг 7. Создание сервиса auth.service.

Шаг 8. Реализация auth.guard.

Шаг 9. Конец.

Для функционирования авторизации и аутентификации в серверной части будет создана модель User и к нему добавлен репозиторий для взаимодействия с данными. Далее будет добавлена конфигурация через папку security – WebSecurityConfig с соответствующими классами и методами для обеспечения безопасности системы. После этого следует добавление контроллера auth , у которого 2 метода: SignIn и SignUp, т.е. зайти и зарегистрироваться[Приложение А].

В клиентской части для функционирования авторизации и аутентификации клиента будут созданы 2 сервиса. Первый сервис это - token storage – сервис, который хранит информацию в браузере о пользователе(клиента). Хранит следующую информацию – строку токена, строку username и authority(роль клиента). Существует 2 роля – это роль клиент(обычный пользователь с ограниченными возможностями) – и роль admin (администратор с расширенными возможностями). Второй сервис это auth.service. Обращение клиентов идет через auth.service, т.е. данный сервис обращается к контроллеру auth в серверной части. И при каждом обращении в клиентской части срабатывает auth-interceptor и идет сравнение токена в браузере и токена на сервере. Для того, чтобы не допускать проникновение незарегистрированных пользователей к дальнейшим возможностям, срабатывает auth-guard. Он определен на странице app-routing.model для того, чтобы запрещать переходить по некоторым ссылкам.

2.4 Алгоритм восстановления пароля

В 21 веке из-за того, что каждый клиент зарегистрирован на разных системах зачастую забывают свой пароль. В зависимости от лояльности к клиентам каждая Служба Безопасности разрабатывают специальную возможность- восстановление пароля. Восстановление пароля можно осуществить с помощью: одноразового кода в сообщении, который поступает на электронную почту, которая была указана при регистрации пользователя; одноразового кода, который высылается на номер мобильного телефона, который был указан и активирован при регистрации клиента. Т.е. при способе восстановления через номер телефона, придет смс-оповещение на основной номер телефона, зачастую он состоит из цифр. Клиент вводит код из цифр, который указан в сообщении и восстанавливает доступ. При восстановлении через электронную почту приходит смс-оповещение на почту в виде кода, состоящий из цифр, или в виде ссылки, для того, чтобы пройти по нему и восстановить доступ, т.е пароль. В данной дипломной работе было разработано восстановление пароля с помощью электронной почты. То есть, в случае, когда клиент забыл пароль, он может восстановить пароль посредством электронной почты. Поступит сообщение на электронную почту со специальным токеном для восстановления. Клиент его копирует и вставляет в URL –адрес и автоматически идет перевод на страницу восстановления пароля.

Алгоритм разработки функции - восстановление пароля:

Шаг 1. Начало.

Шаг 2. Создание контроллера password.

Шаг 3. Добавление методов в контроллер password – forgot и reset.

Шаг 4. Создание компонентов – forgot – password и reset – password.

Шаг 5. Создание сервиса password.service.ts.

Шаг 6. Конец.

Разработано будет восстановление пароля следующим образом: на серверной части будет создан контроллер password, который будет содержать в себе 2 метода – forgot и reset.[Приложение Б]. При нажатии клиентом специальной кнопки: «Забыл пароль», срабатывает метод forgot и отправляет ссылку, по которой можно будет восстановить пароль и доступ в веб-приложение. В сообщении хранится токен. Изначально значения токена равны нулю. Но при нажатии «Забыл пароль», генерируется токен с значениями и отправляется на электронную почту.

После того, как клиент перешел по ссылке и указал новый пароль на странице, срабатывает метод reset. При срабатывании метода reset, соответственно, автоматически меняются данные пользователя (клиента) в таблице. На клиентской части (front-end) для реализации функции восстановления пароля, т.е. взаимодействия с серверной частью(back-end) были созданы 2 компонента. А именно forgot- password и reset- password. И также один сервис – password.service.ts в формате TypeScript.

2.5 Алгоритм работы с Базой Данных

База данных и работа с ней является неотъемлемой частью системы, т.к. система не может функционировать без Базы Данных. База Данных – это организованная структура, которая используется для того, чтобы данные : хранить, изменять, обрабатывать данные больших размеров .А для того, чтобы управлять имеющимися данными используются СУБД – Системы Управления Базами Данных.[4, Тарасов С.В., СУБД для программиста. База данных изнутри, 2015]. Это комплекс программных средств, предназначенные для манипулирования данными (create,select,update,delete). Существуют множество видов СУБД от различных компаний. Наиболее популярные это – MySQL, PostgreSQL, Oracle, Microsoft SQL Server. . В работе была использована СУБД – PostgreSQL. Это свободная объектно- реляционная СУБД.

Алгоритм работы с Базой Данных:

Шаг 1. Начало.

Шаг 2. В application.properties указание пути к БД и конфигурациям ORM.

Шаг 3. Добавление репозиторий в контроллеры, где нужно обращение к Базе Данных.

Шаг 4. Добавление определенных действий в контроллерах для управления данными в БД.

Шаг 5. Конец.

Работа с Базой Данных была разработана следующим образом: на серверной части (back-end) в application.properties указываем путь к Базе Данных и к конфигурациям ORM – Object- Relational Mapping. Это технология, которая предназначена для связи БД с концепциями языков программирования. При запуске веб-приложения создаются таблицы в Базе Данных, на серверной части с помощью репозиторий обращаемся в Базу Данных и управляем ими.

3 ИНФОРМАЦИОННОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

3.1 Описание информационного обеспечения

Информационное обеспечение – это одно из важных элементов всей информационной системы веб- приложения. Для того, чтобы система качественно функционировала важно правильно описать Базу Данных.[4, Тарасов С.В.,СУБД для программиста. База данных изнутри. 2015 год]. Также важную роль играет взаимосвязь имеющихся данных в БД, т.к. изменение одной строчки может привести к значительным и глобальным изменениям других строк в БД. А для того, чтобы правильно определить взаимосвязи в БД используются Системы Управления Базами Данных (СУБД). Главная функция СУБД – это управление данными, которые хранятся в Базе Данных. Система управления базами данных должен поддерживать языки баз данных, и ,конечно же, отвечает за копирование и восстановление имеющихся данных после каких – либо сбоев. [5, Пол Уилтон, Джон Колби,SQL для начинающих,2006г.]. В данной работе в качестве СУБД была выбрана – PostgreSQL , т.к. она обладает рядом преимуществ, такие как : надежность транзакций, возможность наследования и легкая расширяемость при необходимости .[6, Дж.Уорсли, Дж.Дрейк, PostgreSQL Для профессионалов, 2003 г.]. На данный момент PostgreSQL уже может конкурировать с такой мощной Системой Управления Базами Данных как Oracle.

База данных состоит из нескольких таблиц. А именно:

- News – «Новости». Таблица содержит в себе следующие поля: id, caption, date,text. Соответственно, id – это номер новости. В каждой таблице под id понимается нумерация какого либо объекта. Поле caption – это заголовок новости, date- дата опубликования новости и text – сама новость.

- Notes – «Обращения клиентов». Данная таблица содержит в себе поля : id, text, user_id. Т.е., text- суть обращения клиента, user_id – пользователь, который оставил обращение.

- Payments- «Платежи». В данной таблице описываются платежи, которые были произведены клиентом и сохранились истории платежей. Таблица содержит в следующие поля: id, date – дата произведения платежа, payment_amount – сумма платежа, payment_description – назначение платежа и user_id – пользователь, который совершил платеж.

- Roles- Таблица «Роли». Содержит в себе только 2 поля. Id и name. То есть в определены только 2 роля – роль Admin (администратор) и роль User (пользователь).

- Specifications – данная таблица предназначена для добавления новых клиентов с помощью администратора. То есть, таблица состоит из полей: id, balance – остаток на счете клиента, credit_card – 16-значный номер карты, individual – Индивидуальный Идентификационный Номер клиента, name – имя клиента и surname, соответственно, - фамилия клиента.

- Transfers – Таблица «Переводы». В этой таблице описываются переводы, которые были осуществлены клиентом и хранятся в Базе Данных. Она содержит в себе следующие поля: id, date – дата совершения перевода, transfer_amount – сумма перевода, transfer_card – номер карты получателя, user_id – пользователь, которым был совершен данный перевод.

- User_roles – данная таблица предназначена для определения роли клиента и администратора. Содержит в себе 2 поля: user_id – номер пользователя и role_id – номер роли, соответствующий пользователю.

- Users – Таблица «Пользователи», которая хранит всю информацию обо всех пользователях (клиенты, администраторы). Данная таблица состоит из нескольких важных полей, а именно: id, balance – остаток на карте пользователя, credit_card – номер кредитной карты пользователя, email – электронный адрес пользователя, individual – ИИН пользователя, name – имя пользователя, password – пароль пользователя для входа в систему, reset_token – поле, предназначенная для восстановления пароля, surname – фамилия пользователя, username – логин для входа в веб-приложение.

Также как и в реляционной любой Базе Данных таблицы связаны между собой.[Тарасов С.В., СУБД для программиста. База данных изнутри, 2015]. Связи могут быть : один – к – одному; один – к – многим; многие – ко – многим. А сами связи создаются с помощью внешних ключей (foreign key). Внешний ключ (foreign key) – это набор атрибутов, которые ссылаются на первичный ключ(primary key) или unique другой таблицы. Так как основными потребителями данного веб-приложения являются - клиенты, практически все таблицы связаны с таблицей Users через поле user_id.

На рис.3.1. на примере таблицы «Платежи» показана, как она выглядит в Базе Данных:

	id [PK] bigint	date timestamp without time zone	payment_amount bigint	payment_description character varying (255)	user_id bigint
1	1	2020-03-15 21:50:42.276	4000	ONAY	1

Рисунок 3.1 – Таблица «Платежи» в Базе Данных

На следующем рисунке 3.2 описана структура информационной Базы Данных(описание таблиц и полей, взаимосвязи между таблицами) в виде ER - диаграммы:

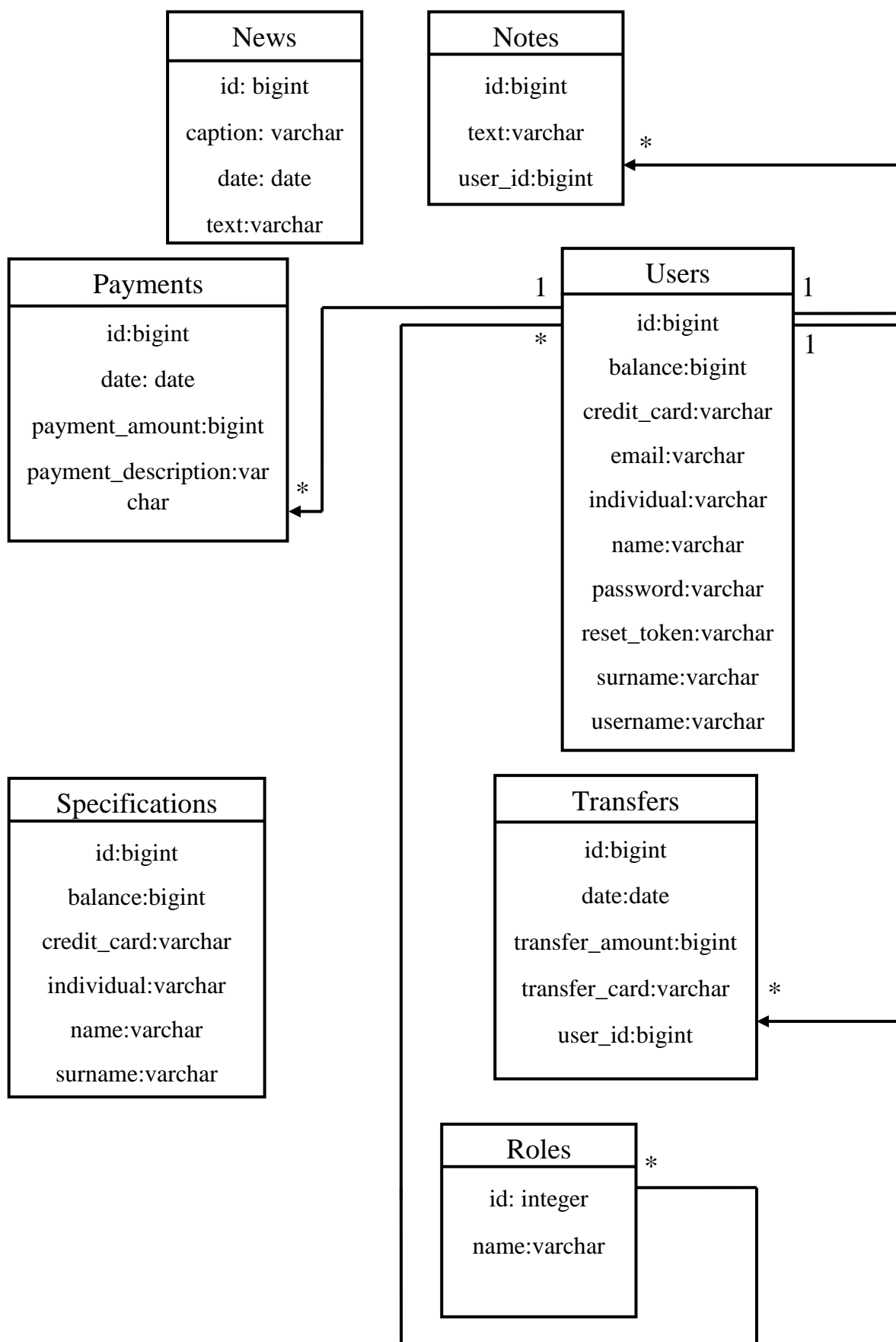


Рисунок 3.2 – Структура Информационной Базы Данных

3.2 Системное программное обеспечение

Системное программное обеспечение, инструментальное программное обеспечение и прикладное программное обеспечение составляют основу (структуру) любого программного обеспечения (рис.3.3).

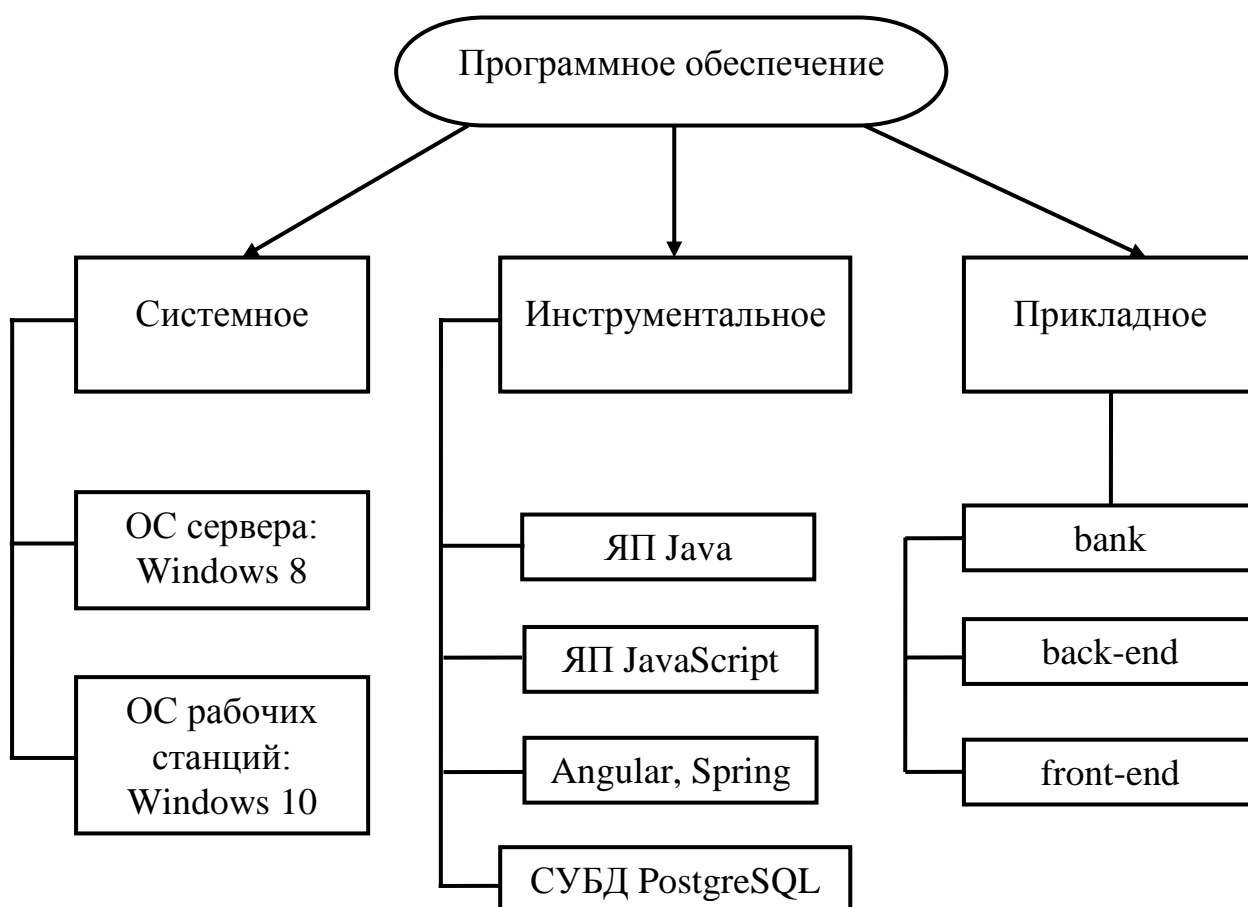


Рисунок 3.3 – Структура программного обеспечения

Системное программное обеспечение – это комплекс программ, которые предназначены для обеспечения управления компонентами компьютерной системы. Системное ПО не занимается решением каких-либо практических задач, а лишь обеспечивает бесперебойную работу программ, оказывая сервисные функции, а также управляет аппаратной частью системы (hardware). Оно состоит из операционной системы и служебных программ (утилит). Для реализации системного программного обеспечения веб-приложения стоял вопрос выбора платформы Windows 8 или же Windows 10. Но все-таки выбор пал на Windows версии 8. Подтолкнула к выбору ОС Windows 8 несколько положительных аспектов, а именно: во-первых, низкая стоимость лицензии, по сравнению с другими версиями Windows; во-вторых простота и скорость установки (в сравнении с другими версиями, данная ОС устанавливается очень быстро); в третьих, стабильность. Тут можно было бы отметить Windows 10, но в связи с тем, что это одна из новейших версий, разработчики все еще дорабатывают, изменяя некоторые возможности и функции. Возможно, в

будущем, и Windows 10 выйдет в чистые лидеры, но пока эта версия не может похвастаться своей стабильностью, нежели Windows 8.

3.3 Инструментальное программное обеспечение

В качестве языков программирования были выбраны **Java** и **JavaScript**. Так как имеется уже знания и опыт работы с этими языками программирования выбор пал именно на эти языки. Java является объектно-ориентированным языком программирования, обладающая рядом преимуществ. Самым главным и основным преимуществом языка программирования Java является полная независимость байт-кода от операционной системы и оборудования, т.е. приложение, разработанное на языке Java может функционировать на любом устройстве. Также технология Java обладает гибкой системой безопасности.[7, Хорстманн Кей С. Java. Библиотека профессионала. Том 1. Основы. 2018 г.].

Для Front-End разработки была использована платформа **Angular**. Это свободная платформа для разработки веб-приложений (фреймворк). Для загрузки пакетов (библиотек) JavaScript был использован **NPM** (Node Package Manager) – стандартный менеджер пакетов. [8, Крокфорд Д., Javascript. Сильные стороны. 2012г.]. Также для Front – разработки был использован редактор исходного кода – **Visual Studio Code** от компании Microsoft.

Для серверной разработки (Back-End) была использована платформа **Spring**. Это простой и универсальный фреймворк(платформа) с открытым исходным кодом для Java-платформы.[9, Уоллс Крейг, Spring в действии. 2015г.]. Для загрузки пакетов Java был использован **Maven**.

Также в ходе разработки веб-приложения была использована технология **REST**. Representational state transfer – это стиль архитектуры программного обеспечения для распределенных систем. Он является простым интерфейсом управления информацией.

Для реализации восстановления пароля был использован **Mailtrap.io**. Это сервис безопасного тестирования писем. Он является доступным, бесплатным для личных проектов и простым в понимании для разработчиков. Принцип работы заключается в следующем: мною был создан специальный тестовый ящик и настроен почтовый сервер для пробных писем (в данном случае, письма для восстановления пароля). Сообщения попадают в тестовый электронный ящик.

В качестве СУБД (системы управления базами данных) был использован **PostgreSQL**. Это свободная объектно- реляционная система управления базами данных. На рынке она сейчас очень популярна, т.к. она базируется на языке SQL. Даже если сейчас набирают большие обороты работы с неструктурированными данными, большая половина компаний в мире все еще не отошла от SQL. PostgreSQL обладает рядом преимуществ, а именно – поддержка БД неограниченного размера, наследование, расширяемость и др.

Для оформления и разработки стилей был использован язык таблиц стилей – **CSS** (Cascading Style Sheets).[10, Д.Макфарланд, Новая большая книга CSS. 2016г.]. А также и был использован **Bootstrap** – набор инструментов для

создания сайтов и веб-приложений. Он содержит в себе шаблоны оформления для веб-форм, кнопок, меток и другие компоненты веб-интерфейса.[11, Роббинс Дженнифер Нидерст. Web-дизайн. Справочник.. 2008г.].

3.4 Прикладное программное обеспечение

Под прикладным программным обеспечением (ПО) понимается специальное ПО, которое предназначено для выполнения определенных задач.

В качестве прикладного программного обеспечения была использована специальная среда разработки **IntelliJ IDEA**. IntelliJ IDEA – специальная интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JS(JavaScript), Python, разработанная компанией JetBrains.

Своими характеристиками и возможностями IntelliJ IDEA напоминает операционную систему MacOS от компании Apple. IntelliJ IDEA, также как и MacOS, закрытая, консервативная, не вызывает какие-либо сложности в работе и, конечно же, пользуется популярностью среди начинающих разработчиков. Разработчику, который использует среду IntelliJ IDEA не нужно находить и устанавливать плагины для интеграции с системами контроля версий и поддержки популярных языков и фреймворков. В соответствии с имеющимися инструментами IntelliJ IDEA наперед предлагает и автоматически вызывает нужный инструмент для разработки.

И, конечно же, основным прикладным программным обеспечением является собственная разработанная система взаимодействия с клиентами Банка – bank, которая состоит из двух приложений это backend(серверная часть) и front-end (клиентская часть). Для функционирования всей системы необходимо соединить 2 приложения : back-end и front-end и подключить к Базе Данных. На рисунке 3.4 представлен фрагмент кода обращения front-end к back-end:

```
return this.http
    .get<News>(this.apiUrl + `/${id}`)
    .pipe(
        retry(2),
        catchError(this.handleError)
    );
```

Рисунок 3.4 - Фрагмент кода обращения клиентской части к серверной части

Как было описано в алгоритме авторизации и аутентификации клиента, для функционирования системы в серверной части был создан контроллер auth, у которого описываются 2 метода : SignIn и SignUp, т.е. зайти и

зарегистрироваться. Ниже представлен листинг программы SignIn (зайти) на рис.3.5:

```
...
public ResponseEntity<?> authenticateUser(@Valid @RequestBody
SignInForm signInRequest) {
    Authentication authentication =
authenticationManager.authenticate(
        new
UsernamePasswordAuthenticationToken(signInRequest.getUsername(),
signInRequest.getPassword()));

SecurityContextHolder.getContext().setAuthentication(authentication);

    String jwt = jwtProvider.generateJwtToken(authentication);
    UserDetails userDetails = (UserDetails)
authentication.getPrincipal();

    return ResponseEntity.ok(new JwtResponse(jwt,
userDetails.getUsername(), userDetails.getAuthorities()));
}
...
```

Рисунок 3.5 - Листинг программы метода SignIn в back-end

После разработки авторизации и аутентификации клиента следует реализация функции восстановления пароля. Для этого на серверной части был создан контроллер password, который содержит в себе 2 метода – forgot и reset. На следующем рисунке 3.6 представлен листинг программы метода forgot:

```
public ResponseEntity<?> forgotPassword(@Valid @RequestBody PasswordForgot
passwordForgotRequest) {
    Optional<User> optional =
userRepository.findByEmail(passwordForgotRequest.getEmail());
    if (!optional.isPresent()) {
        return new ResponseEntity<>(new ResponseReport("User with email does
not exist!"), HttpStatus.BAD_REQUEST);}
    User user = optional.get();
    user.setResetToken(UUID.randomUUID().toString());
    userRepository.save(user);
    // Sending message
    String message = String.format(
        "Пройдите по ссылке для восстановления пароля:
http://localhost:4200/password-reset/"
        + user.getResetToken());
    mailSender.send(user.getEmail(), "Password Reset", message);
    return new ResponseEntity<>(new ResponseReport("Check your email,
please"), HttpStatus.OK);}
}
```

Рисунок 3.6 - Листинг программы метода forgot на back-end

После того, как были созданы 2 приложения, реализованы алгоритмы функционирования, следует подключение веб-приложения к Базе данных. Для этого на серверной части в application.properties был указан путь к Базе Данных и к конфигурациям ORM :

```
# Database
spring.datasource.url=jdbc:postgresql://localhost/bank
spring.datasource.username=postgres
spring.datasource.password=root
spring.jpa.generate-ddl=true

# JWT
jwtSecret=secret
jwtExpiration=86400

# Mail
spring.mail.host=smtp.mailtrap.io
spring.mail.username=5c3d92d1578dde
spring.mail.password=44f4d232e384c0
spring.mail.port=465
spring.mail.protocol=smtp
mail.debug=true
```

Рисунок 3.7 - Листинг программы связи Базы Данных с веб-приложением

Обращение и управление Базой Данных идет с помощью определенных репозиториях в контроллерах. На рисунке - 3.8 в качестве примера показано, как идет обращение к Базе Данных через контроллер NewsController:

```
...
public class NewsController {
    private NewsRepository newsRepository;

    public NewsController(NewsRepository newsRepository) {
        this.newsRepository = newsRepository;
    }
}
...
```

Рисунок 3.8 - Фрагмент листинга программы обращения к БД

Как показано на рисунке 3.8, обращение идет через репозиторий Private Repository. Для того, чтобы при изменении данных автоматически изменения были произведены и сохранены в Базе Данных используется следующий метод(рис.3.9):

```
...
return
ResponseEntity.ok(newsRepository.save(news));
```

Рисунок 3.9 - Листинг программы сохранения данных в БД

3.5 Описание контрольного примера

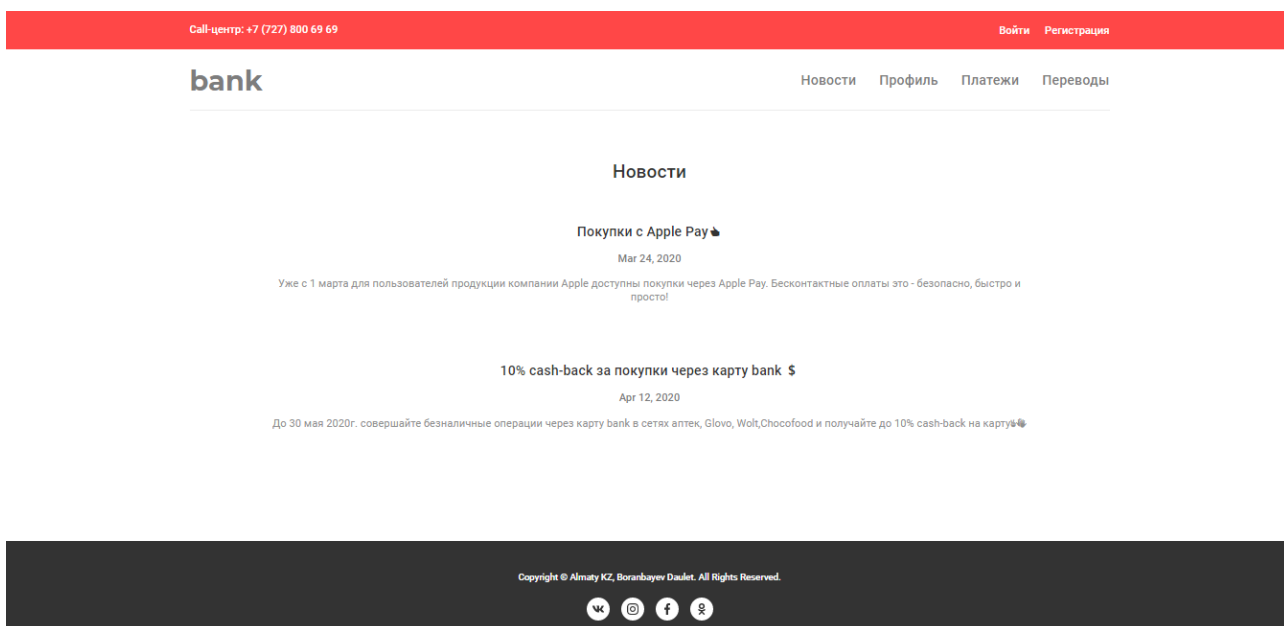


Рисунок 3.10 – Главное меню пользователей и гостей веб-приложения

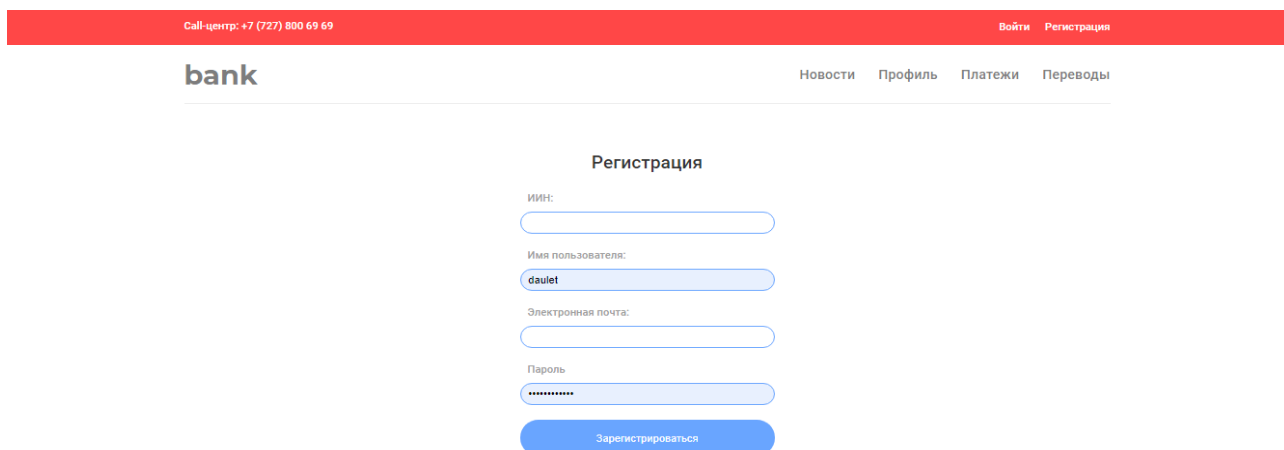


Рисунок 3.11 – Меню регистрации пользователей

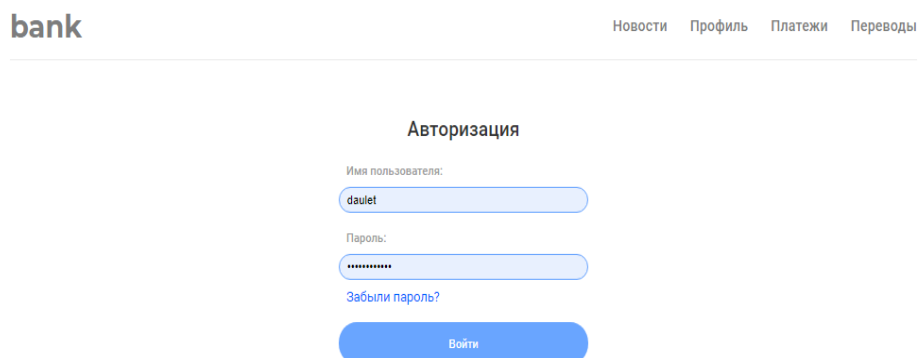


Рисунок 3.12 – Меню авторизации пользователей

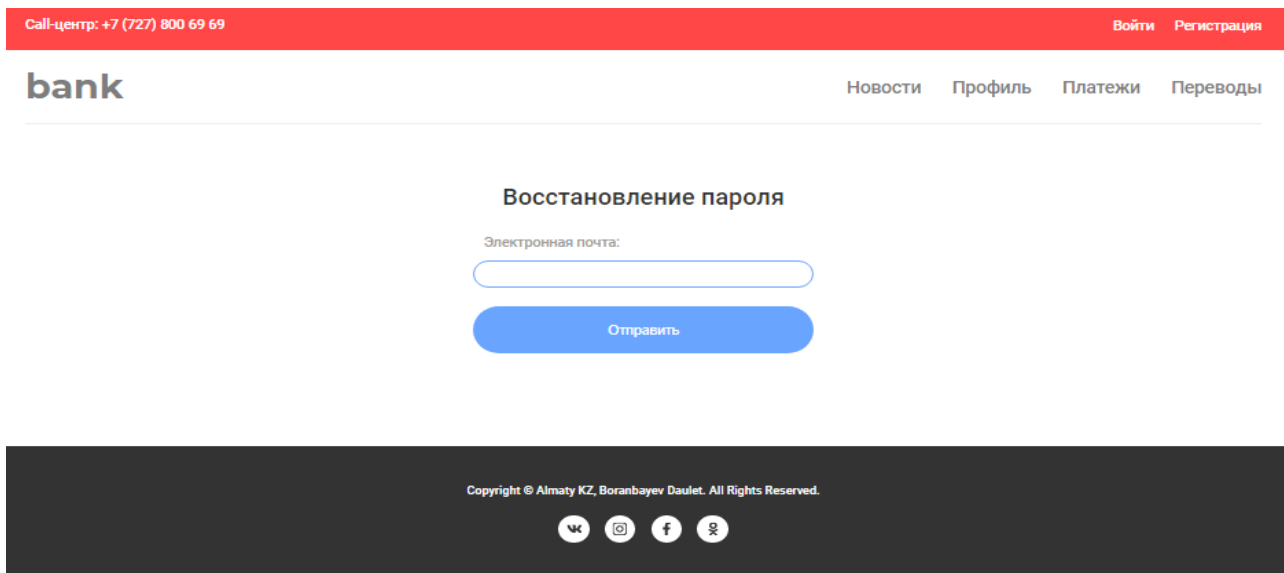


Рисунок 3.13 – Восстановление пароля через электронную почту

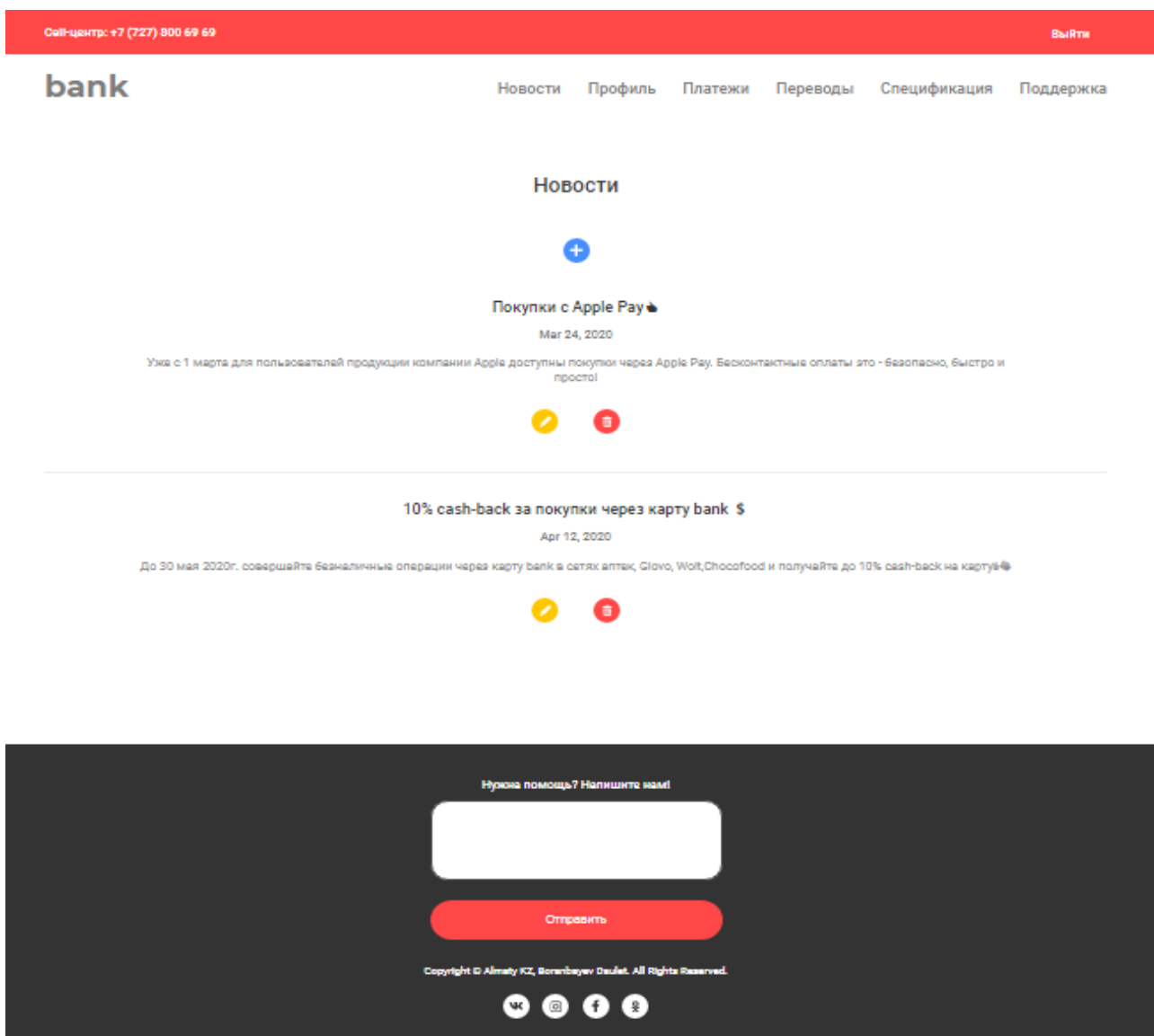


Рисунок 3.14 – Главное меню администратора

Новости

Покупки с Apple Pay 🍏

Мар 24, 2020

Уже с 1 марта для пользователей продукции компании Apple доступны покупки через Apple Pay. Бесконтактные оплаты это - безопасно, быстро и просто!

10% cash-back за покупки через карту bank \$

Апр 12, 2020

До 30 мая 2020г. совершайте безналичные операции через карту bank в сетях аптек, Glovo, Wolt, Chocofood и получайте до 10% cash-back на карту👉

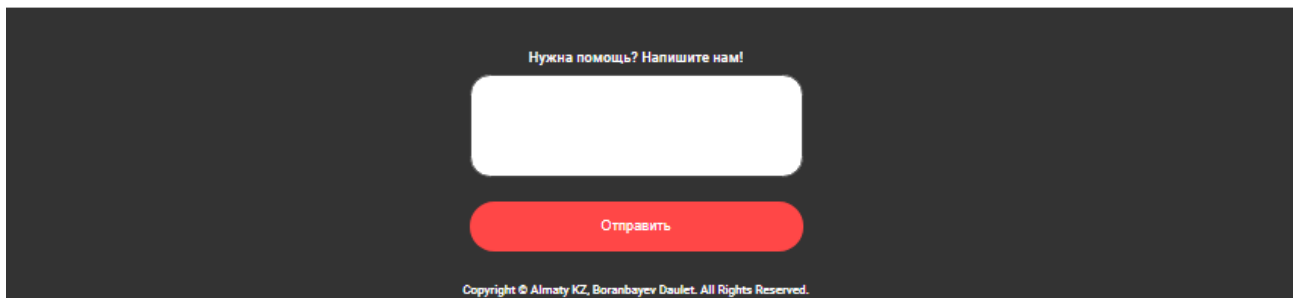


Рисунок 3.15 – Главное меню клиента

4 ВНЕДРЕНИЕ СИСТЕМЫ

4.1 Описание инфраструктуры системы

IT – инфраструктура – это комплекс взаимосвязанных объектов, объединяющая все информационные технологии и ресурсы, используемые компанией. IT-инфраструктура включает в себя аппаратную (hardware) и программную часть (software), а также и системы связи и коммуникации, информационные центры, сети и базы данных(рис.4.1).

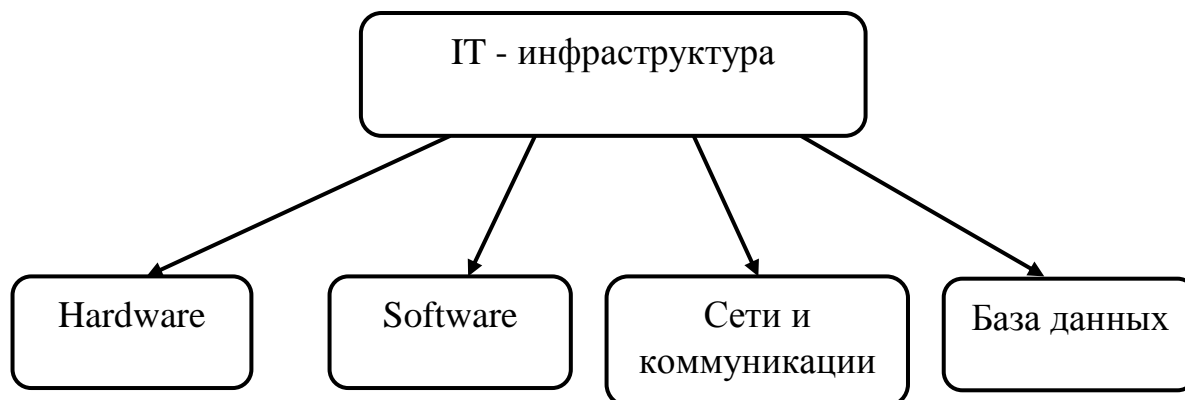


Рисунок 4.1 – IT –инфраструктура

В зависимости от исполняемых задач для организаций выбираются несколько различных конфигураций рабочих станций.

В отделах, где не требуются выполнение сложных и многогранных задач будут соответствовать рабочие станции с доступной средней ценой и производительностью. А именно, рабочие станции Neo Office (PDC G4400 3,30 Ghz/4GbB/500GB/DVDRW)- за среднюю цену на рынке – 80 000 тг., со следующими характеристиками:

- процессор Intel Pentium с тактовой частотой 3.3 ГГц;
- оперативная память DDR 4 Гб;
- количество ядер – 2;
- видеоадаптер Intel HD Graphics 510;
- объем HDD накопителя – 500 Гб.

А в отделы IT, которым требуется большая вычислительная мощность и производительность, соответствует рабочая станция HP EliteDesk 800 G4 (4KW72EA) – за среднюю цену на рынке – 398 564 тг с характеристиками:

- процессор Intel Core i7-8700 с тактовой частотой 3.2 ГГц;
- оперативная память DDR 8 Гб;
- видеоадаптер NVIDIA Graphics 630 2Гб;
- HDD 1 Тб.

В качестве монитора для всех станций - Acer HA220Q Awi с IPS-матрицей, максимальным разрешением 1920x1080 и соотношением сторон 16:9. со средней ценой на рынке - 38100тг.

Операционная система для всех станций – лицензионная версия ОС Windows 8 от компании Microsoft. Также каждому отделу были куплены лицензионные ПО, например отделу бухгалтерии – 1С – Wise – Advice.

Так как все отделы связаны между собой, для функционирования компьютерных сетей была выбрана сетевая технология – Gigabit Ethernet. В компьютерных сетях – термин, описывающий различные технологии передачи Ethernet- кадров со скоростью 1 гигабит в секунду, определяемые рядом стандартов группы IEEE 802.3.

Основные преимущества технологии Gigabit Ethernet:

- пропускная способность сегментов до 1 Гб/с;
- метод случайного доступа Ethernet;
- топология сети «звезда» и поддержка среды передачи данных – витая пара или оптоволоконный кабель.

Выбранный стандарт – 1000Base – LX, сегмент на многомодовом оптоволоконном кабеле с максимальной длиной сегмента 550 м. Скорость передачи 1 Гбит/с.

Для функционирования компьютерной сети была выбрана физическая топология «звезда», где посредством центрального коммутатора будет осуществляться обмен информацией. Тут центральный коммутатор выступает в роли многопортового повторителя. В качестве логической топологии была выбрана логическое кольцо, которое определяется методом доступа Token Passing.

В качестве Интернет-провайдера выступает – Казахстанский провайдер ТОО «Intelsoft Kazakhstan LTD», обладающий рядом преимуществ: скорость 1Гб/с, нет абонентской платы, нет ограничений времени работы, нет ограничений количества подключаемых компьютеров.

В качестве коммуникационного оборудования сети был выбран - PLANET GS-4210-16UP4C - Гигабитный Ultra PoE управляемый коммутатор оснащенный Интеллектуальными функциями PoE от PLANET, созданный для обеспечения потребностей в бизнес приложениях. GS-4210-16UP4C - IPv6/IPv4 dual stack управление и L2/L4 коммутационная архитектура вместе с 16-ю 10/100/1000BASE-T портами с поддержкой 60-Ватт Ultra PoE и 4 Гигабитных TP/SFP комбинированных порта. С максимальной нагрузкой PoE до **400 Ватт** для различных PoE приложений, GS-4210-16UP4C предоставляет быстрое, безопасное и выгодное Ultra PoE решение.

4.2 Перспективы развития

Для того, чтобы обеспечить эффективное функционирование системы, создание сильных конкурентных позиций будут вводиться новые изменения в соответствии с современными Государственными Стандартами. В первую очередь разработчики должны с каждым разом улучшать и обновлять пользовательский интерфейс. В XXI веке все пользователи (user-ы) требуют наиболее дружелюбный интерфейс веб-приложений, а также упрощения многих действий и функций, не теряя при этом никаких возможностей. Если

клиент (пользователь) недоволен чем-то или хочет предложить свои идеи в качестве рекомендаций, то он может связаться со службой Service Desk через обратную связь на главной странице веб-приложения.

В первую же очередь, одним из новшеств, которые будут введены в веб-приложение в ближайшем будущем это упрощенная форма аутентификации. В данной дипломной работе в качестве формы аутентификации была использована наиболее старая, но проверенная форма – парольная. Но в соответствии с требованиями клиентов и быстро растущими информационными технологиями будет реализована и внедрена новая форма аутентификации через биометрические данные пользователей. Эта технология сейчас уже используется на рынке, поэтому внедрение данной формы аутентификации является одним из наиболее важных и востребованных изменений в систему. Уклон будет больше на FaceID, т.е. распознавание лица через камеру ПК или ноутбука, т.к. для распознавания пальца - TouchID требуются аппаратные решения. То есть на данный момент не все компании, занимающиеся выпуском персональных компьютеров и ноутбуков, внедряют специальные возможности распознавания отпечатки пальца – сканер в аппаратную часть. Также рассматриваются варианты распознавания пользователя с помощью сетчатки глаз.

Далее будут разработаны и внедрены технологии анализа больших данных. То есть на главной странице веб-приложения будут проводиться опросы и тестирования пользователей на определение уровня обслуживания клиентов. Исходя из результатов опроса пользователей с помощью метода анализа Больших Данных будет изучена поведенческая модель пользователей, а именно, что хотят, что не нравится, что в основном требуют пользователи. Также с помощью методов анализа Big Data (Больших данных) можно будет выяснить какие из изменений улучшат целевые показатели (KPI).

Ну и ,конечно же, для решения аппаратного обеспечения информационной системы Банка будут изучены и внедрены технологии облачных вычислений. На данный момент облачные вычисления - это один из наиболее быстро развивающихся сегментов рынка информационных технологий. Они обладают рядом преимуществ, а именно: гибкость, мобильность, экономичность, надежность. Технологии облачных вычислений базируются на технологии виртуализации. В первую же очередь будет внедрена виртуализация приложений, т.е. для того, чтобы программа запускалась на одном изолированном сервере, а доступ к ней получали большое количество удаленных пользователей независимо от месторасположения. А также будет внедрена виртуализация на уровне операционных систем для бесперебойного функционирования определенных приложений на различных ОС.

ЗАКЛЮЧЕНИЕ

На данный момент многие Банки Второго Уровня Республики Казахстан сталкиваются с рядом проблем, связанных с информационными технологиями. Это отсутствие CRM, систем взаимодействия с клиентами, проблемы с аппаратной реализацией IT – инфраструктуры, безопасность данных и проблемы хранения данных. Исходя из общего положения и проанализировав все ранее вышперечисленные и упомянутые проблемы, в данной дипломной работе была разработана собственная система взаимодействия с клиентами и предложена руководству «АО ДБ Альфа Банк Казахстан» [Приложение Г].

В ходе работы в первую очередь была разработана функциональная структура системы, описывающая возможности, функции и связи между каждым пользователем системы. Далее был описан обобщенный алгоритм функционирования системы. Разработаны и реализованы алгоритмы решения функциональных задач, а именно: регистрация пользователей, авторизация, восстановление пароли, обратная связь со Службой Поддержки и работа с Базой Данных. Была описана информационная База Данных.

В данном веб-приложении могут зарегистрироваться, авторизоваться и получать услуги (читать новости Банка, совершать платежи, переводы, просматривать личные данные и т.д.) – все клиенты Банка, имеющие выход в Интернет.

Также были оценены перспективы развития системы в соответствии с современными Государственными и Международными стандартами. А именно улучшение разработанной системы: добавление новых и удобных методов аутентификации пользователей, реализация технологий больших данных для анализа и принятия глобальных решений на основе анализа поведенческой модели пользователей и, конечно же, решения аппаратного обеспечения IT-инфраструктуры Банка.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Информационные технологии в банке. Тютюнник А.В., Шевелев А.С. 2003 г.
- 2 Angular и TypeScript. Сайтостроение для профессионалов. Яков Файн и Антон Моисеев. 2018 г.
- 3 Нормативная база и стандарты в области информационной безопасности. Ю.А.Родичев. 2017г.
- 4 СУБД для программиста. База данных изнутри. Тарасов С.В., 2015 г.
- 5 SQL для начинающих. Пол Уилтон, Джон Колби, 2006 г.
- 6 PostgreSQL.Для профессионалов. Дж.Уорсли, Дж.Дрейк, 2003 г.
- 7 Java. Библиотека профессионала. Том 1. Основы. Хорстманн Кей С. 2018 г.
- 8 Javascript. Сильные стороны. Крокфорд Д., 2012 г.
- 9 Spring в действии. Уоллс Крейг, 2015 г.
- 10 Новая большая книга CSS. Д.Макфарланд, 2016 г.
- 11 Web-дизайн. Справочник. Роббинс Дженнифер Нидерст. 2008 г.

ПРИЛОЖЕНИЕ А

```
public class AuthController {
    private UserRepository userRepository;
    private RoleRepository roleRepository;
    private SpecificationRepository specificationRepository;
    private AuthenticationManager authenticationManager;
    private PasswordEncoder passwordEncoder;
    private JwtProvider jwtProvider;

    public AuthController(UserRepository userRepository, RoleRepository
roleRepository, SpecificationRepository specificationRepository,
AuthenticationManager authenticationManager, PasswordEncoder passwordEncoder,
JwtProvider jwtProvider) {
        this.userRepository = userRepository;
        this.roleRepository = roleRepository;
        this.specificationRepository = specificationRepository;
        this.authenticationManager = authenticationManager;
        this.passwordEncoder = passwordEncoder;
        this.jwtProvider = jwtProvider;
    }

    @PostMapping("/signin")
    public ResponseEntity<?> authenticateUser(@Valid @RequestBody SignInForm
signInRequest) {
        Authentication authentication = authenticationManager.authenticate(
            new
UsernamePasswordAuthenticationToken(signInRequest.getUsername(),
signInRequest.getPassword()));

        SecurityContextHolder.getContext().setAuthentication(authentication);

        String jwt = jwtProvider.generateJwtToken(authentication);
        UserDetails userDetails = (UserDetails) authentication.getPrincipal();

        return ResponseEntity.ok(new JwtResponse(jwt, userDetails.getUsername(),
userDetails.getAuthorities()));
    }

    @PostMapping("/signup")
    public ResponseEntity<?> registerUser(@Valid @RequestBody SignUpForm
signUpRequest) {
        if (userRepository.existsByIndividual(signUpRequest.getIndividual())) {
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

```
    return new ResponseEntity<>(new ResponseReport("Fail -> Individual is
already taken!"), HttpStatus.BAD_REQUEST);
}

    if (!specificationRepository.existsByIndividual(signUpRequest.getIndividual()))
{
    return new ResponseEntity<>(new ResponseReport("Fail -> Individual does
not exist!"), HttpStatus.BAD_REQUEST);
}

    if (userRepository.existsByUsername(signUpRequest.getUsername())) {
    return new ResponseEntity<>(new ResponseReport("Fail -> Username is
already taken!"), HttpStatus.BAD_REQUEST);
}

    if (userRepository.existsByEmail(signUpRequest.getEmail())) {
    return new ResponseEntity<>(new ResponseReport("Fail -> Email is already
in use!"), HttpStatus.BAD_REQUEST);
}

    // Looking for specification in DB
    Specification specification =
specificationRepository.findByIndividual(signUpRequest.getIndividual()).get();

    // Creating user
    User user = new User(signUpRequest.getIndividual(),
signUpRequest.getUsername(),
    specification.getName(), specification.getSurname(),
specification.getCreditCard(), specification.getBalance(),
    signUpRequest.getEmail(),
passwordEncoder.encode(signUpRequest.getPassword()));

    Set<String> strRoles = signUpRequest.getRoles();
    Set<Role> roles = new HashSet<>();

    strRoles.forEach(role -> {
    switch (role) {
    case "admin":
    Role adminRole =
roleRepository.findByName(RoleName.ROLE_ADMIN)
    .orElseThrow(() -> new RuntimeException("Fail! -> Cause: User
Role not found"));
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

```
roles.add(adminRole);

break;
default:
    Role userRole = roleRepository.findByName(RoleName.ROLE_USER)
        .orElseThrow(() -> new RuntimeException("Fail! -> Cause: User
Role not found"));
    roles.add(userRole);
}
});

user.setRoles(roles);
userRepository.save(user);

specificationRepository.deleteById(specification.getId());

return new ResponseEntity<>(new ResponseReport("User registered
successfully! Check your mail to activate"), HttpStatus.OK);
}
}
```

ПРИЛОЖЕНИЕ Б

```
public class PasswordController {
    private UserRepository userRepository;
    private MailSender mailSender;
    private PasswordEncoder passwordEncoder;

    public PasswordController(UserRepository userRepository, MailSender
mailSender, PasswordEncoder passwordEncoder) {
        this.userRepository = userRepository;
        this.mailSender = mailSender;
        this.passwordEncoder = passwordEncoder;
    }

    @GetMapping("/{username}")
    public ResponseEntity<?> getResetToken(@PathVariable String username) {
        Optional<User> optional = userRepository.findByUsername(username);

        if (!optional.isPresent()) {
            return new ResponseEntity<>(new ResponseReport("User with username
does not exist!"), HttpStatus.BAD_REQUEST);
        }

        User user = optional.get();

        return new ResponseEntity<>(user.getResetToken(), HttpStatus.OK);
    }

    @PostMapping("/forgot")
    public ResponseEntity<?> forgotPassword(@Valid @RequestBody
PasswordForgot passwordForgotRequest) {
        Optional<User> optional =
userRepository.findByEmail(passwordForgotRequest.getEmail());

        if (!optional.isPresent()) {
            return new ResponseEntity<>(new ResponseReport("User with email does
not exist!"), HttpStatus.BAD_REQUEST);
        }

        User user = optional.get();

        user.setResetToken(UUID.randomUUID().toString());
    }
}
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```
userRepository.save(user);

// Sending message
String message = String.format(
    "Пройдите по ссылке для восстановления пароля:
http://localhost:4200/password-reset/"
    + user.getResetToken()
);

mailSender.send(user.getEmail(), "Password Reset", message);

return new ResponseEntity<>(new ResponseReport("Check your email,
please"), HttpStatus.OK);
}

@PutMapping("/reset/{token}")
public ResponseEntity<?> resetPassword(@PathVariable String token, @Valid
@RequestBody PasswordReset passwordResetRequest) {
    Optional<User> user = userRepository.findByResetToken(token);

    if (user.isPresent()) {
        User resetUser = user.get();

        if
(passwordResetRequest.getPassword().equals(passwordResetRequest.getConfirmPas
sword())) {

resetUser.setPassword(passwordEncoder.encode(passwordResetRequest.getPassword
()));
        resetUser.setResetToken(null);
        userRepository.save(resetUser);
        return ResponseEntity.ok().body(resetUser);
        }

        return new ResponseEntity<>(new ResponseReport("Invalid confirmation of
password!"), HttpStatus.BAD_REQUEST);
    }

    return ResponseEntity.notFound().build();
}
}
```


ПРИЛОЖЕНИЕ В

```
<header class="header">
  <!-- Contacts -->
  <div class="contacts">
    <div class="container">
      <div class="contacts__wrapper">
        <a class="contacts__phone" href="tel:#">Call-центр: +7 (727) 800 69 69</a>
        <div class="contacts__authorization">
          <a class="contacts__link" *ngIf="!authority" (click)="login()">Войти</a>
          <a
            class="contacts__link"
            routerLink="signup">Регистрация</a>
            *ngIf="!authority"
            <a class="contacts__link" *ngIf="authority" (click)="logout()">Выйти</a>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

<!-- Nav -->
<div class="container">
  <div class="header__wrapper">
    <a class="header__logo" href="#">bank</a>
    <nav class="nav">
      <a class="nav__link" routerLink="news">Новости</a>
      <a class="nav__link" routerLink="profile">Профиль</a>
      <a class="nav__link" routerLink="payment">Платежи</a>
      <a class="nav__link" routerLink="transfer">Переводы</a>
      <a class="nav__link" routerLink="specification" *ngIf="authority
'admin'">Спецификация</a>
      <a class="nav__link" routerLink="help" *ngIf="authority
'admin'">Поддержка</a>
    </nav>
  </div>
</div>
</header>

<div class="container">
  <router-outlet></router-outlet>
</div>

<footer class="footer">
  <!-- Help -->
  <form *ngIf="authority" class="help" name="help" (ngSubmit)="h.form.valid &&
onSubmit()" #h="ngForm" novalidate>
```

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ В

```
<div class="help__element">
  <label for="helptext" class="help__label">Нужна помощь? Напишите
нам!</label>
  <textarea type="text" class="help__textarea" rows="6" maxlength="255"
name="helptext" [(ngModel)]="help.text"
  #helptext="ngModel" required></textarea>
  <div *ngIf="h.submitted && helptext.invalid" class="help__danger">
    <div *ngIf="helptext.errors.required">Заполните форму!</div>
  </div>
  <button type="submit" class="help__submit">Отправить</button>
</form>
```

```
<p class="footer__text">
  Copyright © Almaty KZ, Boranbayev Daulet. All Rights Reserved.
</p>
<div class="footer__contacts">
  <button class="vk-link" onclick="window.location.href='https://vk.com'" aria-
label="ВКонтакте"></button>
  <button class="inst-link" onclick="window.location.href='https://instagram.com'"
aria-label="Instagram"></button>
  <button class="facebook-link"
onclick="window.location.href='https://facebook.com'"
aria-
label="Facebook"></button>
  <button class="ok-link" onclick="window.location.href='https://ok.ru'" aria-
label="Одноклассники"></button>
</div>
</footer>
```

ПРИЛОЖЕНИЕ Г

«Альфа-Банк»
Еншілес Банк
Акцияерлік қоғамы

050059, Алматы қаласы, Назарбаев даңғылы, 226
2051, 8-8000-8000-75
infokz@alfabank.kz; www.alfabank.kz



Акционерное общество
дочерний банк
«Альфа-Банк»

050059, город Алматы, проспект Назарбаева, 226
2051, 8-8000-8000-75
infokz@alfabank.kz; www.alfabank.kz



Рекомендательное письмо

Боранбаев Даулет Құрманалыұлы работает в АО ДБ Альфа Банк, Департамент клиентского сервиса, Управление Контакт-центр, Отдел Call Center с 06 июня 2019 г. – в должности специалиста группы по обслуживанию физических лиц. В функциональные обязанности Боранбаева Даулета входило:

- Предоставление информационно-справочного обслуживания по телефону действующим и потенциальным клиентам Банка с целью консультирования и сохранения клиентов Банка в соответствии с установленными нормативами;
- Сбор данных по устным обращениям абонентов и передача в смежные подразделения для проведения анализа с целью, повышения качества обслуживания дальнейшего решения вопроса;
- Предложение и продажа услуг продуктов Компании на входящих и исходящих звонках;
- Техническая поддержка по Интернет Банкингу для Юридических лиц.

Помимо своих должностных обязанностей, Боранбаев Даулет занимался системным администрированием и участвовал в команде поддержки функционирования информационной системы Банка. На данный момент разрабатывается система взаимодействия с клиентами Банка, где активное участие принимает Боранбаев Даулет. Результаты дипломной работы будут учтены в перспективах внедрения данного продукта в Информационную систему Банка.

Айгерим Усенова
Руководитель группы Отдела Call Center
Управление Контакт-центр
Департамент клиентского сервиса
АО ДБ "Альфа-Банк Казахстан" г. Алматы
15 мая 2020 г.

029001