

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

Санкт-Петербургский колледж телекоммуникаций им. Э.Т. Кренкеля

Допустить к защите
Заместитель директора
по учебной работе
_____ Н.В. Калинина
6 июня 2024 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Разработка корпоративной системы коммуникаций

(тема ВКР)

Вид выпускной квалификационной работы Дипломный проект
(дипломный проект, дипломная работа)

Специальность 09.02.07 Информационные системы и программирование
(код и наименование специальности)

Квалификация Программист
(наименование квалификации в соответствии с ФГОС СПО)

Студент _____ А. А.
(подпись) Малакаускайте
05 июня 2024 г.

Руководитель _____ О.В. Колбанева
(подпись) 05 июня 2024 г.

Санкт-Петербург
2024

Работа написана мною самостоятельно и не содержит неправомерных заимствований.
Работа может быть размещена в электронно-библиотечной системе университета)

05.06.2024

(дата)

(подпись студента)

(ИОФ студента)

Текст ВКР размещен в электронно-библиотечной системе университета

Руководитель отдела комплектования библиотеки

05.06.2024

(дата)

(подпись)

(ИОФ)

Коэффициент оригинальности ВКР _____ %

Проверил:

(Должность, ИОФ)

05.06.2024

(дата)

(подпись)

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

Санкт-Петербургский колледж телекоммуникаций им. Э.Т. Кренкеля

УТВЕРЖДАЮ

Заместитель директора
по учебной работе

_____ Н.В. Калинина

13 марта 2024 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы (ВКР)

- | | | | |
|-----------------|---|----------|------------------|
| 1. Студент | <u>Малакаускайте Анна Альгирдовна</u>
(фамилия, имя, отчество) | № группа | <u>К 512/517</u> |
| 2. Руководитель | <u>Преподаватель СПбКТ, Колбанева Ольга Владимировна</u>
(фамилия, имя, отчество, должность, уч. степень и звание) | | |
| 3. Квалификация | <u>Программист</u>
(наименование квалификации в соответствии с ФГОС СПО) | | |
| 4. Вид ВКР | <u>Дипломный проект</u>
(дипломный проект, дипломная работа) | | |
| 5. Тема ВКР | <u>Разработка корпоративной системы коммуникаций</u> | | |

утверждена приказом ректора университета от 11.04.2024 г. № 437/КС

6. Исходные данные (технические требования):

Технологический стек: язык программирования, СУБД, защита персональных данных
на основании ФЗ-152, авторизация на основе JWT, клиент-серверная архитектура,
взаимодействие с базой данных через API, установочный файл, реализация защиты
ввода данных

7. Содержание работы (анализ состояния проблемы, проведение исследований, разработка, расчеты параметров, экономическое обоснование и др.)

Постановка задачи, проектирование программных модулей, разработка программного
продукта, тестирование программного обеспечения, организация защиты данных,
оформление технической документации на программный продукт, обзор мероприятий
по технике безопасности и охране труда, экономическое обоснование проекта

8. Вид отчетных материалов, представляемых в ГЭК_(пояснительная записка, перечень, графического материала, отчет о НИР, технический проект, образцы и др.)

Пояснительная записка, презентация для защиты проекта, исходные коды проекта

9. Консультанты по ВКР с указанием относящихся к ним разделов

Раздел	Консультант (фамилия, имя, отчество)	Подпись, дата	
		Задание выдал	Задание принял
Введение	<i>О.В. Колбанева</i>	<i>05.04.2024</i>	<i>10.04.2024</i>
1. Постановка задачи	<i>О.В. Колбанева</i>	<i>11.04.2024</i>	<i>15.04.2024</i>
2. Проектирование программных модулей	<i>О.В. Колбанева</i>	<i>16.04.2024</i>	<i>20.04.2024</i>
3. Разработка программного продукта	<i>О.В. Колбанева</i>	<i>20.04.2024</i>	<i>30.04.2024</i>
4. Тестирование программного обеспечения	<i>О.В. Колбанева</i>	<i>10.05.2024</i>	<i>15.05.2024</i>
5. Информационная безопасность	<i>О.В. Колбанева</i>	<i>16.05.2024</i>	<i>20.05.2024</i>
6. Техническая документация на программный продукт	<i>О.В. Колбанева</i>	<i>21.05.2024</i>	<i>25.05.2024</i>
7. Мероприятия по технике безопасности и охране труда	<i>Н.А. Ожерельева</i>	<i>26.05.2024</i>	<i>28.05.2024</i>
8. Экономическое обоснование проекта	<i>Т.А. Барулина</i>	<i>29.05.2024</i>	<i>02.06.2024</i>
Заключение	<i>О.В. Колбанева</i>	<i>03.06.2024</i>	<i>05.06.2024</i>

Дата выдачи задания

18 апреля 2024 г.

Дата представления ВКР к защите

05 июня 2024 г.

Председатель предметной (цикловой) комиссии
в компьютерных системах

Информатики и программирования

Н.В. Кривоносова

(подпись)

(И.О. Фамилия)

Руководитель ВКР

О.В. Колбанева

(подпись)

(И.О. Фамилия)

Студент

А.А. Малакаускайте

(подпись)

(И.О. Фамилия)

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

Санкт-Петербургский колледж телекоммуникаций им. Э.Т. Кренкеля

Календарный план выполнения ВКР

№ п/п	Наименование этапов выпускной квалификационной работы (ВКР)	Срок выполнения этапов ВКР	Примечание
1.	Постановка цели выполнения ВКР и задач	19.04.2024-16.05.2024	
2.	Работа с теоретическим материалом	19.04.2024-16.05.2024	
3.	Сбор информации, необходимой для написания работы		
4.	Систематизация и обработка материалов ВКР		
5.	Анализ полученных в работе результатов, обобщение		
6.	Подготовка отчетных материалов, представляемых в государственную экзаменационную комиссию, доклада к защите и презентации	19.04.2024-16.05.2024	
7.	Консультации с руководителем ВКР		
8.	Представление выполненной ВКР руководителю для подготовки отзыва		
9.	Подготовка к защите ВКР, включая подготовку к процедуре защиты и процедуру защиты		

Председатель предметной (цикловой) комиссии
в компьютерных системах

Информатики и программирования

Н.В. Кривоносова

(подпись)

(И.О. Фамилия)

Руководитель ВКР

О.В. Колбанева

(подпись)

(И.О. Фамилия)

Студент

А.А. Малакаускайте

(подпись)

(И.О. Фамилия)

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ	8
ВВЕДЕНИЕ	9
1. ПОСТАНОВКА ЗАДАЧИ	11
1.1. Обзор аналогов	11
1.2. SWOT-анализ	18
1.3. PEST-анализ	18
1.4. Формирование требований к программному продукту	19
1.4.1. Бизнес требования	19
1.4.2. Пользовательские требования	20
1.4.3. Функциональные требования	20
1.4.4. Нефункциональные требования	21
1.4.5. Ограничения	21
1.4.6. Требования к интерфейсам	21
1.4.7. Требования к данным	22
1.5. Программные средства разработки	22
1.6. Аппаратные средства разработки	22
2. ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ	24
2.1. Архитектура системы	24
2.2. Моделирование основных сценариев системы	27
2.3. Проектирование графического интерфейса пользователя	34
2.4. Проектирование и разработка модели данных	39
3. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА	42
3.1. Реализация регистрации и авторизации пользователя	42
3.2. Реализация клиентской части	43
3.2.1. Функция отправки сообщения	43
3.2.2. Метод загрузки файла в облачное хранилище	46
3.2.3. Метод сохранения данных о пользователе	48
3.3. Реализация серверной части	48
4. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	50
4.1. История изменений	50
4.2. Терминология тестирования программного обеспечения	50
4.3. Стратегия тестирования	51
4.4. Определение объектов тестирования	51
4.5. Архитектура тестируемой системы	52
4.6. Описание процесса тестирования	53
5. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ	55
5.1. Определение защищаемых информационных активов	55
5.2. Вид и содержание персональных данных	55
5.3. Модель угроз ПНД	56
5.4. Возможные сценарии угроз	59
5.5. Организация защиты данных в проекте	63
5.6. Защита программного продукта	63
6. ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ НА ПРОГРАММНЫЙ ПРОДУКТ	65
6.1. Руководство пользователя	65
6.2. Руководство администратора	65
6.3. Определение качества продукта по метрикам	65
7. МЕРОПРИЯТИЯ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ И ОХРАНЕ ТРУДА	67
7.1. Характеристика объекта проектирования и условий его эксплуатации	67
7.1.1. Технические данные проектного устройства	67

7.1.2. Условия эксплуатации проектного объекта.....	67
7.1.3. Требования к обслуживающему персоналу.....	68
7.2. Мероприятия по технике безопасности	69
7.2.1. Организационно-технические мероприятия по охране труда.....	69
7.2.2. Требования электробезопасности при работе с ПК	69
7.2.3. Требования защиты от излучения.....	70
7.2.4. Требования защиты от шума	70
7.3. Противопожарная безопасность.....	71
7.3.1. Противопожарная безопасность при работе с персональным компьютером	71
7.4. Общие требования охраны труда.....	72
8. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА	74
8.1. Расчёт затрат на разработку ПО.....	74
8.1.1. Затраты на оплату (ЗОТ).....	74
8.2. Эксплуатационные затраты на оборудование	75
8.2.1. Суммарная годовая стоимость эксплуатационных затрат.....	75
8.2.2. Затраты на техническое обслуживание и ремонт.....	75
8.2.3. Затраты на материалы и комплектующие	75
8.2.4. Амортизационные отчисления.....	75
8.2.5. Стоимость электроэнергии.....	76
8.2.6. Эффективный фонд рабочего времени.....	76
8.2.7. Цена эксплуатационных затрат.....	76
8.3. Затраты на материалы	76
8.3.1. Накладные расходы.....	76
8.4. Себестоимость разработки программного обеспечения.....	77
8.5. Расчет экономического эффекта разработчика и пользователя (заказчика) ПО	77
8.5.1. Экономический эффект у разработчика программного обеспечения	77
ЗАКЛЮЧЕНИЕ.....	79
СПИСОК ИСТОЧНИКОВ ИНФОРМАЦИИ	80
ПРИЛОЖЕНИЕ 1.	82
ПРИЛОЖЕНИЕ 2	99
ПРИЛОЖЕНИЕ 3	103
ПРИЛОЖЕНИЕ 4	119
ПРИЛОЖЕНИЕ 5	135
ПРИЛОЖЕНИЕ 6	136

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ПО – программное обеспечение;

БД – база данных;

СУБД – система управления базами данных;

ОС – операционная система;

SQL – Structured Query Language, язык структурированных запросов;

SSD – Solid-State Drive, твердотельный накопитель;

HDD – Hard Disk Drive, жесткий диск;

CRM-система — это программное обеспечение для бизнеса, которое помогает работать с клиентской базой, собирать лиды, отслеживать действия клиентов и сотрудников и автоматизировать рутинные операции;

XAML (eXtensible Application Markup Language) - язык разметки, используемый для инициализации объектов в технологиях на платформе .NET.

Wcf_chat – сервис, который отвечает за то, как будет работать сервер (библиотека классов .net framework).

ВВЕДЕНИЕ

В современном прогрессивном мире компаниям и предприятиям важно постоянно улучшать показатели своей деятельности быть надежным и устойчивым ко всем внешним и внутренним изменениям. Для успешной работы руководитель должен постоянно следить за изменениями в законодательстве и новыми тенденциями, регулярно обращать внимание на мощную поддержку, обновление и улучшение эффективности производительности системы. Одним из важных аспектов продуктивной деятельности является грамотно выстроенная корпоративная коммуникация.

На сегодняшний день наблюдается ряд тенденций рынка информационных технологий, таких как мобильность сотрудников, возрастающее количество удаленно работающего персонала. В связи с этим от компаний требуется гибкость в организации бизнес-процессов. Для реализации системы передачи информации могут использоваться разные средства: интерактивные доски, например, Migo, планировщики или CRM системы, такие как Jira или Trello. Телефонная связь и электронная почта также являются распространенными средствами коммуникации. Однако наиболее эффективным и удобным способом связи является мессенджер. Мессенджер – это программа для мгновенного обмена текстовыми сообщениями, аудиозаписями, фотографиями и другими видами файлов. Корпоративные социальные сети помогают компаниям решить эту задачу и, при грамотном использовании, повышают продуктивность работы сотрудников.

Корпоративная коммуникация — это совокупность действий, связанных с управлением и организацией всех внутренних и внешних коммуникаций, направленных на формирование благоприятной точки зрения среди заинтересованных сторон, от которых зависит компания. Это включает в себя сообщения, которые распространяются корпоративной организацией, органом или учреждением для своей аудитории, такой как сотрудники, средства массовой информации, партнеры по каналам и широкая общественность. Организации стремятся донести одно и то же послание до всех заинтересованных сторон, обеспечить согласованность действий, доверие и этичность.

Коммуникации внутри организации выполняют множество функций, которые взаимосвязаны между собой, отвечают за создание эффективной рабочей обстановки на предприятии.

Корпоративный мессенджер – это современный инструмент коммуникации, который обеспечивает эффективное взаимодействие внутри компании. Он позволяет сотрудникам легко обмениваться информацией, делиться файлами и координировать свою работу без привязки к офису или рабочему месту.

Преимущества использования корпоративного мессенджера невероятно широки. Во-первых, он значительно упрощает коммуникацию между коллегами, особенно если они находятся в разных городах или даже странах. Команда может быстро обмениваться идеями, задавать вопросы, получать обратную связь и сотрудничать удаленно.

Во-вторых, корпоративный мессенджер способствует более эффективной организации работы. Простое создание групповых чатов позволяет сформировать команды для конкретных задач и проектов, где люди могут обсуждать и координировать свои действия. Это сокращает время, затрачиваемое на необходимое общение и согласование, и повышает производительность работы.

Также стоит отметить возможность управления процессами через корпоративный мессенджер. Многие платформы предлагают интеграцию с другими инструментами, такими как системы управления проектами, CRM или электронный документооборот. Это позволяет сотрудникам иметь доступ к необходимым ресурсам прямо из мессенджера и упрощает взаимодействие с другими системами.

Несмотря на наличие огромного количества общедоступных мессенджеров, крупный бизнес постепенно отказывается от публичных сервисов и переходит в защищенные приложения для обмена сообщениями.

Российские корпоративные мессенджеры за последний год значительно продвинулись в качестве предоставляемых возможностей, поэтому важно учесть все аспекты, которые должны присутствовать, чтобы быть конкурентоспособным на рынке. Быстрая, кроссплатформенная и простая в работе система коммуникация выполнит основные задачи, которые могут понадобиться предприятию. Из дополнительных требований могут быть наличие чатов, каналов, видео звонков, общих хранилищ, а также преимуществом станут интеграции со сторонними бизнес-приложениями и сервисами, календари. Администрирование учетных записей, наличие шифрования передаваемых сообщений, механизмов защиты данных, соблюдение требований законодательства тоже не мало важные аспекты, которые стоит учитывать.

В разрабатываемом продукте, корпоративной системе коммуникации, важно обратить внимание не реализацию трех взаимодействующих между собой слоев, отвечающим за отображение данных и обработкой событий пользовательского интерфейса, слоем, реализующим бизнес-логику приложения и слоем, который регулирует обмен сообщениями и обращения к базе данных.

Таким образом, целью написания данной дипломной работы является создание продукта для обеспечения корпоративной коммуникации, а именно корпоративного мессенджера, выполняющего базовые функции, которые необходимы для качественной и эффективной работы предприятия.

Объект исследования: корпоративные коммуникации.

Предмет исследования: корпоративный мессенджер.

Для достижения намеченной цели были поставлены следующие задачи:

1. Произвести анализ существующих корпоративных коммуникаций и мессенджеров. Изучить и сравнить функциональность, удобство использования, безопасность и интеграционные возможности различных мессенджеров, чтобы определить требования к разрабатываемому продукту.

2. Исследовать деятельность сотрудников, чтобы определить требования к корпоративной системе коммуникаций, которые будут наиболее полезными и востребованными.

3. Разработать план и архитектуру приложения. Выделить основные модули и функциональность мессенджера, а также способы интеграции с другими бизнес-приложениями и сервисами.

4. Реализовать программное приложение мессенджера для корпоративных коммуникаций, используя выбранные технологии и языки программирования, уделив особое внимание безопасности передаваемых сообщений и соблюдению требований законодательства.

5. Выполнить комплексную оценку реализованной информационной системы. Оценить функциональность, производительность, удобство использования и эффективность мессенджера на основе тестирования.

В процессе написания данной дипломной работы были использованы различные методы исследования, такие как анализ литературы, сравнение, синтез информации, системный анализ, математическое моделирование СУБД, информационное моделирование данных. В разработке приложения были применены стандарты объектно-ориентированного подхода к проектированию и программированию, сетевые технологии, desktop-приложение полностью реализовано на языке C# с использованием SQL-запросов для работы с базой данных.

1. ПОСТАНОВКА ЗАДАЧИ

1.1. Обзор аналогов

Перед тем, как приступить к реализации программного продукта, необходимо провести анализ конкурентов на рынке. С его помощью можно сделать выводы о том, как должен создаваться проект на основе достоинств и недостатков уже существующих ресурсов.

В качестве первого аналога хотелось бы рассмотреть корпоративный мессенджер VK Teams. Данный сервис входит в корпоративный набор инструментов VK Workspace, который является частью социальной сети VK. В рассматриваемой системе представлены такие функции, как синхронизация контактов, создание групп или каналов, переписка, видео- и аудио- звонки.

Внешний вид VK Teams представлен на рис. 1.1.1.

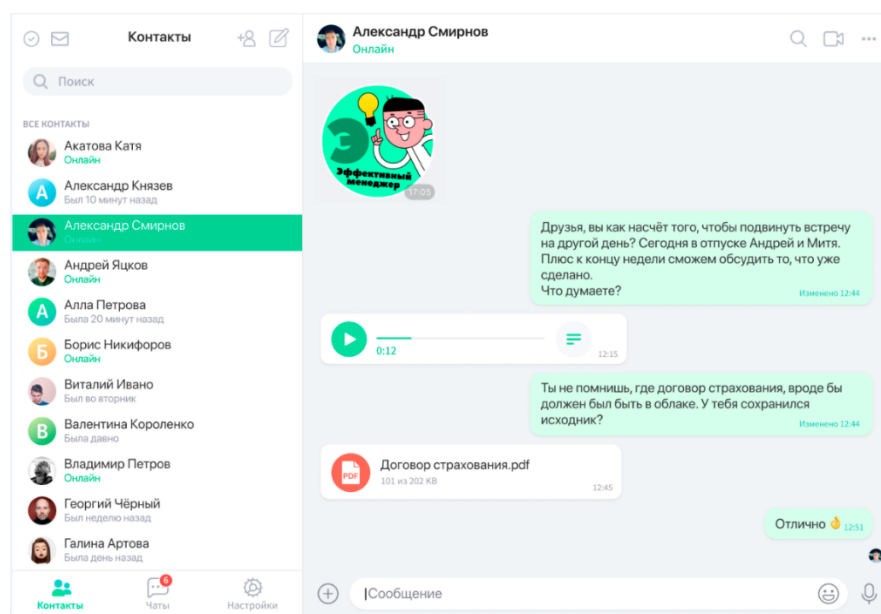


Рис. 1.1.1. Внешний вид VK Teams

В качестве достоинства данной системы можно отметить возможность создания задачи в таск-менеджер прямо из переписки (рис. 1.1.2). Также в VK Teams можно добавить исполнителя, указать дедлайн и обозначить приоритет задания и управлять статусом задачи (рис. 1.1.3).

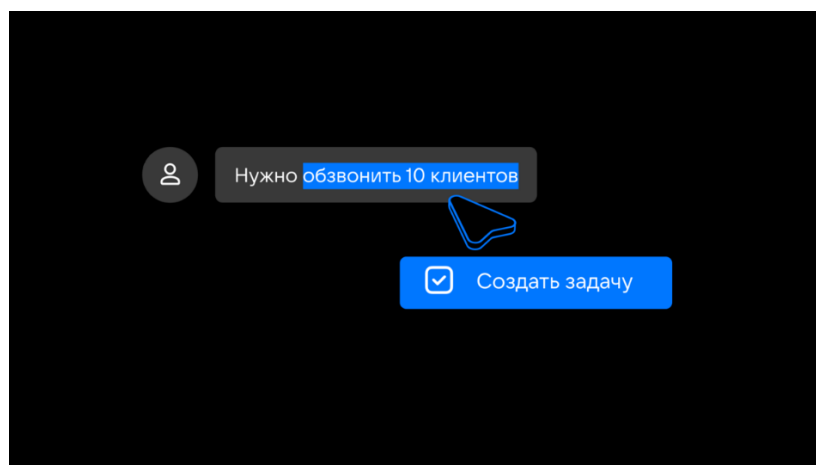


Рис. 1.1.2. Добавление задачи в таск-менеджер из переписки

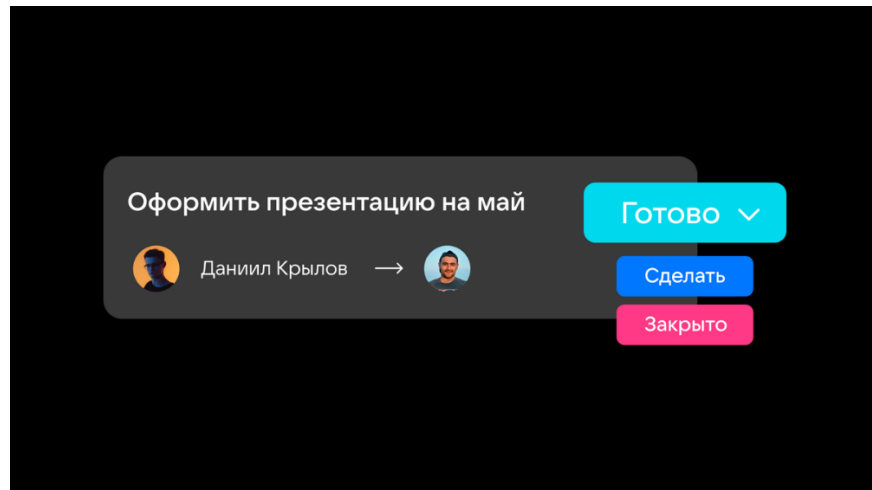


Рис. 1.1.2. Управление статусом задачи

Также достоинством можно считать планирование совместных видео- и аудиозвонков во встроенном календаре. Календарь предлагает интервалы, когда все участники свободны, а созвон сам добавляется в расписание каждого коллеги (рис. 1.1.4).

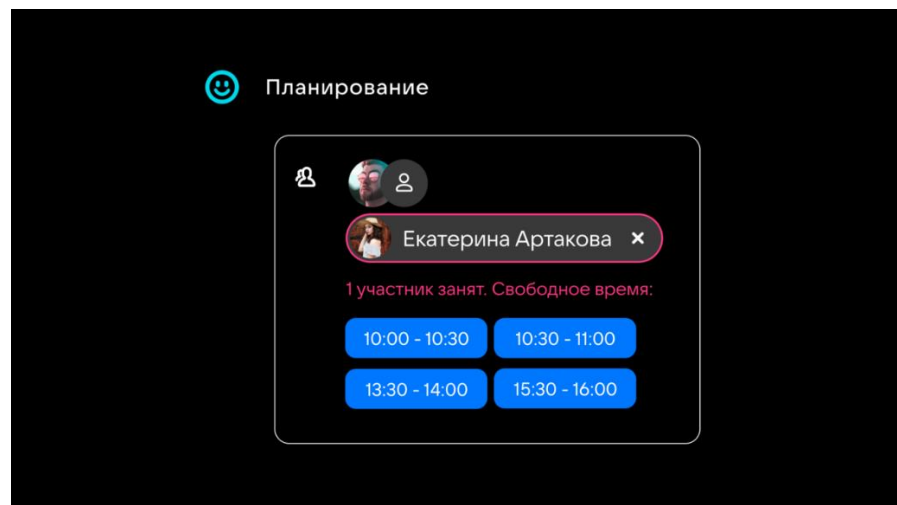


Рис. 1.1.3. Индивидуальное расписание

Однако среди недостатков этого сервиса пользователи выделяют то, что приложение не присылает уведомления о новых сообщениях, что вынуждает пользователей каждый раз открывать приложение.

Стоимость подписки на сервис варьируется от 1690 руб. (подписка для 1-30 пользователей на 1 месяц) до 83 900 руб. (101-300 пользователей на 1 год).

Вторым аналогом можно назвать Microsoft Teams. В данном сервисе реализованы следующие функции: создание групп или каналов, переписка, видео- и аудиозвонки, демонстрация экранов.

Внешний вид сервиса представлен на рис. 1.1.5.

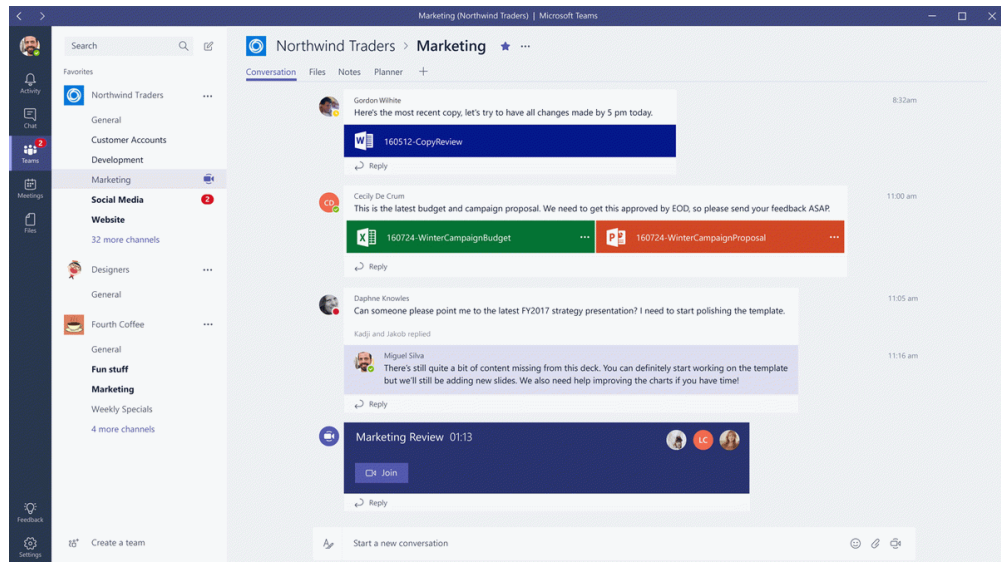


Рис. 1.1.4. Внешний вид Microsoft Teams

Одним из самых главных достоинств Microsoft Teams можно назвать совместное редактирование документов Word, Excel, Power Point в режиме реального времени. Эта функция позволяет просматривать изменения, вносить правки и оставлять комментарии, не покидая платформу (рис. 1.1.6).

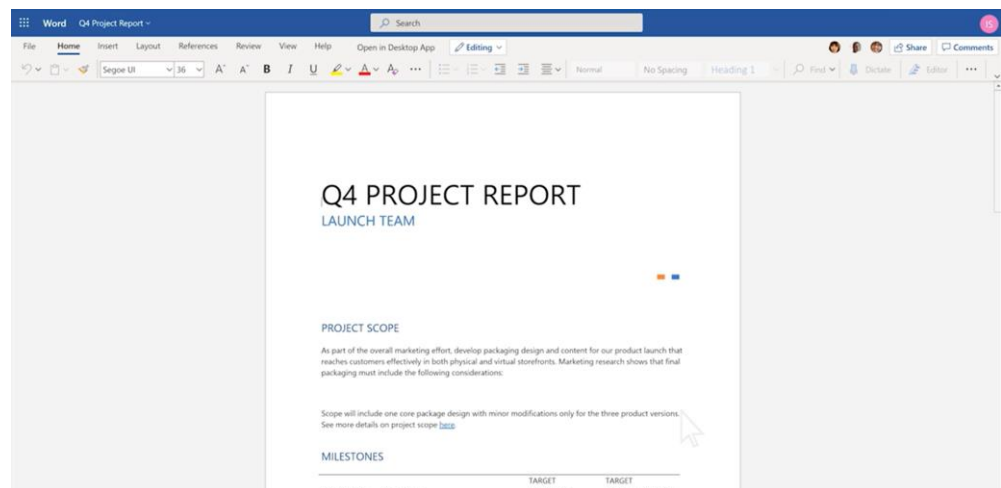


Рис. 1.1.5. Совместное редактирование документов

Среди достоинств также нужно выделить интеграцию с другими инструментами. Microsoft Teams создает единое место для работы над проектами и обмена информацией, интегрируясь с другими сервисами и инструментами Microsoft (SharePoint, OneDrive, Outlook и др.).

В качестве третьего аналога хотелось бы назвать TrueConf. В данном корпоративном мессенджере реализованы следующие функции: создание и планирование онлайн-конференций, обмен сообщениями и файлами, демонстрация экрана, трансляция онлайн-мероприятий.

Внешний вид сервиса представлен на рис. 1.1.7.

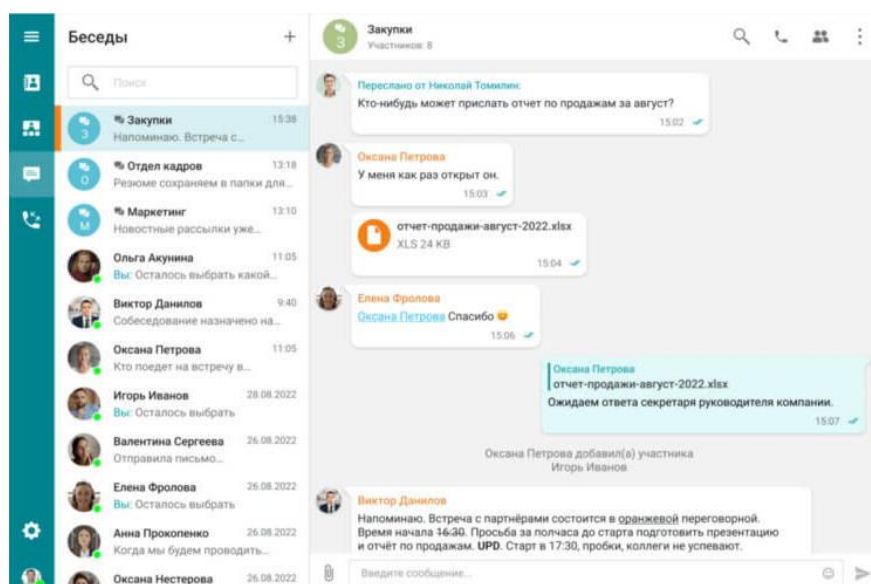


Рис. 1.1.6. Окно с чатами в TrueConf

В качестве главного преимущества данного сервиса можно назвать встроенный ИИ в видеоконференции, который используется для адаптивного улучшения видео и звука. Сервер TrueConf Server научили распознавать русскую речь всех участников конференции, а также поиск по произнесенным фразам с проигрыванием нужных фрагментов в записи встречи.

Главным недостатком данной системы можно считать отсутствие собственного сервера и необходимость установки системы на сервер компании.

Четвертый аналог корпоративного мессенджера, который будет рассмотрен – Slack. Это корпоративный мессенджер с функциями коммуникации и обмена информацией. Он позволяет командам быстро обмениваться сообщениями, файлами, назначать задачи и управлять проектами. Slack также имеет интеграцию с множеством других приложений, что делает его универсальным инструментом для командной работы. Мессенджер обеспечивает высокий уровень безопасности данных используя шифрование и другие меры защиты.

Из минусов продукта можно назвать стоимость – бесплатный тарифный план с ограниченными возможностями – и ограничение количества участников.

Последним, пятым рассмотренным аналогом, будет корпоративный мессенджер Пачка. Этот мессенджер также обеспечивает высокий уровень безопасности данных, легко интегрируется с другими бизнес-приложениями, дает возможности создавать групповые чаты, управлять правами доступа. Из минусов: ограниченная бесплатная версия, сложная настройка, требуется обучение сотрудников.

Сравним данные приложения по соответствию требованиям FURPS+.

FURPS+ - это метод классификации требований к программному обеспечению, который включает в себя следующие категории:

- Features (Функции) - основные возможности и услуги, которые должен предоставлять продукт.
- User Stories (Пользовательские истории) - описание того, как пользователи будут использовать продукт, включая их цели и задачи.
- Risks (Риски) - возможные проблемы и препятствия, которые могут возникнуть в процессе разработки и эксплуатации продукта.
- Priority (Приоритет) - определение важности и срочности выполнения задач и требований.
- Scope (Объем) - границы проекта, определение того, какие функции и возможности будут включены в продукт.
- Stability (Стабильность) - оценка надежности и устойчивости продукта к изменениям и ошибкам.

- Testability (Тестируемость) - возможность проверить и подтвердить соответствие продукта требованиям и спецификациям.
- Support (Поддержка) - процесс предоставления помощи и поддержки пользователям продукта.

Ниже представлен анализ аналогов рассмотренных ранее корпоративных мессенджеров по некоторым из категории FURPS+ (табл. 1.1.1).

Таблица 1.1.1

Анализ аналогов

Требования	VK Teams	Microsoft Teams	TrueConf	Slack	Пачка
Features (Функции)	Приложение для обмена сообщениями и проведения видеоконференций, разработанное для пользователей социальной сети “ВКонтакте”. Оно позволяет общаться в группах, обмениваться файлами, проводить видеоконференции и звонить другим пользователям.	Корпоративное приложение, которое предоставляет набор инструментов для коммуникации, совместной работы и доступа к информации.	Платформа для видеоконференцсвязи и обмена сообщениями между сотрудниками компании, которая позволяет проводить онлайн-встречи, вебинары и звонки.	Корпоративный мессенджер для обмена сообщениями. Обладает понятным интерфейсом, высокой производительностью и безопасностью. Поддерживает аудио- и видеозвонки, интеграцию с различными приложениями и возможность управления проектами.	Корпоративное приложение для обмена сообщениями и совместной работы, которое позволяет командам общаться, обмениваться файлами и работать над проектами вместе. Он имеет широкий функционал, включая поддержку аудио- и видеозвонков, интеграцию с различными приложениями и возможность управления проектами.

Требования	VK Teams	Microsoft Teams	TrueConf	Slack	Пачка
User Stories (Пользовательские истории)	Основной пользовательской аудиторией Vk Teams являются пользователи социальной сети “ВКонтакте”, которые хотят участвовать в групповых чатах, видеоконференциях и звонках. Однако, приложение также может быть полезно для любых пользователей, которые ищут удобный способ общения и видеоконференцсвязи.	Пользователями Microsoft Teams являются сотрудники компаний, которым необходимо общаться, работать вместе и получать доступ к корпоративной информации.	Пользователи TrueConf - это люди, которые хотят общаться и сотрудничать с другими людьми через видеоконференции и звонки. Это могут быть как частные лица, так и компании.	Пользователями являются люди, работающие в компаниях. Корпоративный мессенджер повышает эффективность работы сотрудников.	Сотрудники различных компаний, которые используют его для общения и совместной работы.

Требования	VK Teams	Microsoft Teams	TrueConf	Slack	Пачка
Priority (Приоритет)	Целью приложения является предоставление пользователю социальной сети “ВКонтакте” удобного способа общения и проведения видеоконференций.	Целью приложения является улучшение коммуникации и совместной работы внутри компании, упрощение доступа к корпоративной информации и повышение эффективности работы сотрудников.	Цель TrueConf - предоставить пользователям удобное и надежное средство для видеоконференций и звонков. (бесплатно только для 50 онлайн-пользователей)	Целью корпоративного мессенджера Slack является обеспечение эффективного общения и сотрудничества между сотрудниками компании. Это позволяет ускорить процесс принятия решений, улучшить коммуникацию и повысить продуктивность работы команды.	Цель корпоративного мессенджера Пачка заключается в обеспечении эффективной коммуникации и сотрудничества между сотрудниками компании, что позволяет ускорить процесс принятия решений и повысить продуктивность работы команды.
Risks (Риски)	Приложение не присылает уведомления о новых сообщениях, что вынуждает пользователей каждый раз открывать приложение. Высокая цена подписки.	Возможность скорой блокировки на территории Российской Федерации. Высокая цена подписки.	Необходимость установки системы на сервер компании. Высокая цена подписки.	Возможность скорой блокировки на территории Российской Федерации. Возможная утечка информации из-за большой нагрузки на приложение. Высокая цена подписки.	Возможность утечки данных. Высокая цена подписки.

Сравнение разрабатываемого корпоративного мессенджера с существующими аналогами позволило выявить его отличительные особенности и уникальность. В ходе анализа были изучены различные представители рынка корпоративных мессенджеров, такие как Microsoft Teams, VK Teams и TrueConf. Каждый из представленных аналогов обладает качественным функционалом. Несмотря на это были выявлены следующие преимущества разрабатываемого продукта – PigeonGram - перед конкурентами:

- доступная цена приобретения продукта;
- отсутствие популярности, поэтому ниже вероятность утечки важной информации;
- удобный и понятный интерфейс, не уступающий аналогам.

1.2. SWOT-анализ

В ходе выполнения выпускной квалификационной работы для определения функциональных требований был проведен SWOT-анализ (табл. 1.2.1):

Таблица 1.2.1

SWOT-анализ

	Возможности 1. Растущие рекламные возможности 2. Огромная база клиентов позволяет открыть новые источники дохода 3. Цифровая трансформация	Угрозы 1. Меняющиеся тенденции в экономике 2. Низкие барьеры для входа конкурентов
Сильные стороны 1. Удобный и понятный интерфейс 2. Современные технологии приложения 3. Бесплатное пользование 4. Хранилище для файлов 5. Обмен сообщениями 6. Регистрация и аутентификация	За счет удобного и понятного интерфейса пользователь интуитивно пользуется функционалом. Программа позволяет регистрироваться, а затем, при каждом входе авторизоваться. Пользователи могут обмениваться сообщениями между друг другом. Также она поддерживает хранилище различных объемов информации, с доступом к облачному хранилищу, при этом все происходит бесплатно.	Из-за нестабильной ситуации в мире не всегда известно актуальность продукта, поэтому нужно быть вовлеченным в процессы, чтобы оставаться конкурентоспособным на фоне других участников, выходящих на рынок.
Слабые стороны 1. Проблема с защитой данных пользователей 2. Не работает без интернета 3. Нет групповых звонков	Важно сократить ситуации, при которых может быть затронута безопасность пользователя. Также необходимо сделать все возможное, чтобы пользователь хотел пользоваться приложением, независимо от того, что оно работает только с интернетом. Проработать функционал обмена сообщениями, которые будут заменять групповые звонки.	Необходимо следить за конкурентами, а именно за тем, как они справляются с аналогичными проблемами, чтобы не пропасть с поля зрения пользователей. Также важно придумать особенности, благодаря которым приложение будет выигрывать на рынке.

Рассмотренный SWOT-анализ помог в формировании функциональных требований, которые будут представлены в пункте 1.4.3.

1.3. PEST-анализ

В ходе выполнения выпускной квалификационной работы для определения функциональных требований был проведен PEST-анализ (табл. 1.3.1):

Таблица 1.3.1

PEST-анализ

Фактор	Возможность	Угроза
Политические факторы	1.Внешнеполитические санкции	1.Иностранные сервисы запретили российским пользователям пользоваться своими приложениями, появилось необходимость искать аналоги
	2.Изменения в законодательстве, появление запретов на распространение той или иной информации	2.Появилась необходимость отслеживать за каждым словом, используемым в ПО как разработчиками, так и пользователями, чтобы не попасть под нарушение законодательства РФ
Экономические факторы	1.Рост инфляции	1.Повышение возможностей ПО может быть экономически не выгодно
	2.Нестабильный курс валют	2.От курса валют может зависеть тенденции в мире технологий, важно следить за этим
Социокультурные факторы	1.Нестабильные тренды в обществе	1.Важно следить за изменениями, чтобы быть конкурентоспособным
	2.Повышение миграционных показателей	2.Может произойти резкий отток или приток клиентов, поэтому важно следить за этим, чтобы не допустить поломку ПО
	3.Требования пользователей к удобству и функциональности ПО	3.Представленные возможности ПО могут разочаровать аудиторию, из-за чего продукт станет не актуальным
Технологические факторы	1.Активное использование ИИ	1.Необходимо пользоваться новыми технологиями, чтобы приобретать те же возможности для улучшения возможностей ПО
	2.Повышение спроса технических специалистов	2.Необходимо создать минимальные угрозы для поломки ПО, чтобы избежать того случая, что починить будет некому
	3.Рост числа гаджетов	3.Необходимо делать ПО, которое можно использовать на разных типах ОС

Представленный PEST-анализ указывает на факторы, требующие внимания во время реализации программного продукта.

1.4. Формирование требований к программному продукту

На основании вывода из обзора аналогов, SWOT-анализа и PEST-анализа были сформированы требования к разрабатываемому приложению.

1.4.1. Бизнес требования

Бизнес требования к корпоративному мессенджеру:

1. **Безопасность данных:** мессенджер должен обеспечивать высокий уровень защиты конфиденциальных и чувствительных информационных данных компании.

2. **Функциональность:** мессенджер должен предоставлять широкий набор функций, таких как обмен сообщениями, возможность сохранения файлов и документов в общее хранилище, наличие доступа к облачному хранилищу, функция поиска чатов, обзор статистики пользователя, редактирование профиля.

3. **Удобный пользовательский интерфейс:** мессенджер должен иметь простой и интуитивно понятный интерфейс, приятный дизайн, чтобы каждый сотрудник мог без затруднений начать им пользоваться.

4. **Аналитика и отчетность:** мессенджер должен иметь функционал аналитики и отчетности, чтобы предоставлять данные о количестве сообщений, активности пользователей, эффективности коммуникаций и т.д.

5. **Разграничение доступа:** мессенджер должен позволять установку прав доступа на уровне отдельных пользователей или групп, чтобы обеспечить безопасность и контроль доступа к информации.

6. **Масштабируемость:** мессенджер должен быть готов к масштабированию и способен адаптироваться к растущим потребностям компании, поддерживая большое количество пользователей и высокую производительность.

1.4.2. Пользовательские требования

Пользовательские требования к корпоративному мессенджеру:

1. **Простота использования:** мессенджер должен иметь интуитивно понятный интерфейс и простые настройки, чтобы пользователи могли легко освоить и начать пользоваться системой без необходимости обучения.

2. **Обмен сообщениями:** мессенджер должен позволять отправлять текстовые сообщения, чтобы пользователи могли взаимодействовать друг с другом и передавать информацию.

3. **Сохранение файлов и документов:** мессенджер должен позволять пользователям обмениваться файлами и документами различных форматов, чтобы упростить совместное редактирование и обмен материалами.

4. **Поиск чатов:** мессенджер должен позволять находить нужный чат, чтобы пользователи могли легко находить нужную информацию в случае необходимости.

5. **Редактирование профиля:** мессенджер должен позволять пользователю изменить информацию о своем аккаунте.

1.4.3. Функциональные требования

Функциональные требования к корпоративному мессенджеру:

1. **Регистрация и аутентификация:** мессенджер должен предоставить возможность регистрации новых пользователей и аутентификации существующих пользователей с использованием безопасных методов (например, паролей).

2. **Обмен сообщениями:** мессенджер должен позволять пользователям обмениваться текстовыми сообщениями в режиме реального времени.

3. **Сохранение файлов и документов:** мессенджер должен позволять пользователям обмениваться файлами и документами различных форматов, чтобы упростить совместное редактирование и обмен материалами.

4. **Поиск чатов:** мессенджер должен позволять находить нужный чат, чтобы пользователи могли легко находить нужную информацию в случае необходимости.

5. **Редактирование профиля:** мессенджер должен позволять пользователю изменить информацию о своем аккаунте.

6. Управление доступом: мессенджер должен позволять устанавливать права доступа к чатам и файлам для различных пользователей, администраторов и групп, обеспечивая безопасность и конфиденциальность информации.

7. Шифрование и безопасность: мессенджер должен обеспечивать шифрование передаваемых данных и сохраненных сообщений, а также обеспечивать защиту от несанкционированного доступа к чатам и файлам.

1.4.4. Нефункциональные требования

Нефункциональные требования к корпоративному мессенджеру:

1. Безопасность: мессенджер должен обеспечивать высокий уровень защиты данных, включая защиту от несанкционированного доступа, за счет шифрование паролей.

2. Производительность: мессенджер должен обеспечивать высокую производительность при отправке, получении сообщений и при выполнении других операций для обеспечения плавной и эффективной коммуникации.

3. Пользовательский опыт: мессенджер должен предоставлять удобный и интуитивно понятный интерфейс, приятный дизайн и гибкие настройки, чтобы пользователи могли легко освоить систему и настроить ее под свои потребности.

4. Поддержка: мессенджер должен предоставлять эффективную техническую поддержку, и регулярные обновления для устранения возможных проблем, и обеспечения бесперебойной работы системы.

1.4.5. Ограничения

Основные ограничения, которые могут повлиять на проектирование и разработку корпоративного мессенджера:

1. Бюджет: наличие ограниченного бюджета может влиять на выбор технологий, функциональности и возможностей, которые можно реализовать в мессенджере.

2. Ресурсы: доступные ресурсы, включая команду разработчиков, аппаратное и программное обеспечение, могут оказать влияние на сложность и сроки разработки мессенджера.

3. Совместимость и интеграция: наличие существующих систем и инструментов в организации, с которыми необходимо интегрировать мессенджер, может ограничивать выбор технологий и возможности интеграции.

4. Технические ограничения: имеющиеся технические ограничения, такие как доступные технологии, совместимость устройств и операционных систем, могут оказывать влияние на выбор и реализацию функциональности мессенджера.

5. Отсутствие опыта: небольшой опыт команды разработчиков может влиять на качество программного продукта.

1.4.6. Требования к интерфейсам

Требования к интерфейсу у корпоративного мессенджера:

1. Простота использования: интерфейс должен быть интуитивно понятным и легким в освоении для новых пользователей, а также предоставлять удобные и понятные элементы управления для выполнения основных операций.

2. Четкая структура и навигация: интерфейс должен иметь ясную структуру, упрощающую поиск и доступ к основным функциям, а также эффективные механизмы навигации для перемещения между различными разделами и чатами.

3. Понятные и информативные элементы: элементы интерфейса, такие как кнопки,

иконки, метки полей и т.д., должны быть понятными и информативными для пользователей, предоставлять достаточную информацию и подсказки для правильного использования функций.

4. Адаптивность и отзывчивость: интерфейс должен быть адаптивным к различным размерам экранов и высоким разрешениям, а также обеспечивать отзывчивость в реальном времени для быстрого и плавного взаимодействия.

5. Удобство работы с чатами: интерфейс должен обеспечивать удобство работы с чатами, включая возможность быстрого переключения между чатами, отображения прочитанных и непрочитанных сообщений, а также поддержки ответов и цитирования сообщений.

6. Визуальная привлекательность: интерфейс должен быть эстетически привлекательным и приятным для глаза, с использованием соответствующих цветов, шрифтов и графических элементов.

1.4.7. Требования к данным

Требования к данным у корпоративного мессенджера:

1. Конфиденциальность: данные, передаваемые и хранящиеся в мессенджере, должны быть конфиденциальными и защищенными от несанкционированного доступа.

2. Шифрование: мессенджер должен обеспечивать шифрование данных во время их передачи и хранения, чтобы предотвратить возможность перехвата и чтения информации третьими лицами.

3. Управление доступом: мессенджер должен предоставлять возможность управления доступом к данным на уровне пользователей и групп, чтобы ограничить доступ только необходимым пользователям и защитить информацию от несанкционированного распространения.

5. Сохранение данных: мессенджер должен обеспечивать сохранность и целостность данных в течение заданного срока, чтобы пользователи могли хранить и получать доступ к истории сообщений.

6. Сохранение файлов и документов: мессенджер должен позволять пользователям обмениваться файлами и документами различных форматов, чтобы упростить совместное редактирование и обмен материалами.

7. Поиск чатов: мессенджер должен позволять находить нужный чат, чтобы пользователи могли легко находить нужную информацию в случае необходимости.

1.5. Программные средства разработки

Корпоративный мессенджер «PigeonGram» был разработан в среде разработки Visual Studio с на языке программирования C#. Для гибкой работы с графическими стилями xml-код был написан вручную.

Для реализации сервера был использован язык программирования C#.

Для хранения данных о пользователях, сообщений и файлов был выбран SQL Server.

1.6. Аппаратные средства разработки

К основным аппаратным средствам разработки корпоративного мессенджера относятся: Сервер. Является мощным вычислительным узлом с достаточным объемом оперативной памяти, процессором высокой производительности и большим объемом хранилища данных.

В ходе реализации были использованы два ноутбука на операционной системе Windows:

1. Сервер был реализован на ПК с процессором Intel Core i7, оперативной памятью 8 МБ и SSD на 512 ГБ;

2. Клиент был написан на ПК с процессором Intel Core i5, оперативной памятью 8 МБ и HDD на 256 Гб и SSD на 128 Гб.

Все аппаратные средства взаимодействуют между собой, чтобы обеспечить работу корпоративного мессенджера на высоком уровне производительности и безопасности.

В представленном параграфе было произведено формирование основных требований к программному продукту.

Выводы первой главы:

Проведенные исследования помогло сделать следующее:

- рассмотрены пять аналогов программного продукта и проведено сравнение;
- сформулированы основные требования к системе;
- представлен SWOT и PEST анализ продукта.

2. ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ

2.1. Архитектура системы

В ходе проектирования вариантов использования был выбран инструмент Visio 2016 позволяющий пользователям создавать диаграммы из обычного текстового языка. Разработана диаграмма вариантов использования.

Use case – это часть требований к ПО. Он улучшает разработку продукта, благодаря чему пользователь получает ожидаемый результат (см. рис. 2.1.1).

Актер – любая внешняя по отношению к программной системе сущность.

Прецедент – вариант использования, функционал программного продукта.

Виды ассоциаций между компонентами UseCase:

- прямая ассоциация;
- наследование – возможно только между Актёрами, указывает на общий функционал;
- ассоциация включения – возможна только между вариантами использования, предполагает выделения родительского и дочернего варианта использования, обязательное выполнение;

- расширение – возможно только между вариантами использования, позволяет показать дополнительный функционал с родительскими и дочерними вариантами использования.

От актера «Пользователь» идут следующие прямые ассоциации к прецедентам вариантам использования таким, как:

- регистрация;
- вход в систему;
- редактирование профиля;
- чаты;
- отправка сообщения;
- сохранить в архив;
- найти сообщение.

Прецедент варианта использования «Регистрация» имеет следующие ассоциации включения такие, как:

- ввод фамилии;
- ввод электронной почты;
- ввод имени;
- ввод пароля.

Прецедент варианта использования «Регистрация» имеет следующие альтернативные потоки такие, как:

- ввод пустого поля, вместо фамилии;
- ввод электронной почты неверного формата;
- ввод пустого поля, вместо почты;
- ввод пустого поля, вместо имени;
- ввод пароля неверного формата.

Прецедент варианта использования «Вход в систему» имеет следующие расширения такие, как:

- выход из системы.

Прецедент варианта использования «Редактирование профиля» имеет следующие ассоциации включения такие, как:

- изменения почты;
- изменение имени пользователя;
- изменение фамилии пользователя;
- изменение логина пользователя;
- установка фото профиля.

Прецедент варианта использования «Редактирование профиля» имеет следующие

альтернативные потоки такие, как:

- ввод почты неверного формата;
- ввод логина неверного формата;
- ввод фамилии неверного формата;
- оставление полей пустыми;
- установка фото неверного формата;
- имя пользователя неверного формата.

Прецедент варианта использования «Отправка сообщения» имеет следующие ассоциации включения такие, как:

- отправить текст;

Прецедент варианта использования «Отправка сообщения» имеет следующие расширения такие, как:

- получение сообщения;
- ответить.

Прецедент варианта использования «Отправка сообщения» имеет следующие альтернативные потоки такие, как:

- отправить не существующему контакту;
- отправить слишком большой текст.

Прецедент варианта использования «Сохранить в архив» имеют следующие расширения такие, как:

- сохранить документ в формате pdf;
- сохранить документ в формате word;
- удалить документ.

Прецедент варианта использования «Редактирование профиля» имеет следующие альтернативные потоки такие, как:

- сохранить документ не подходящего формата;
- сохранить документ слишком большого размера;
- удалить документ без прав доступа.

Прецедент варианта использования «Чаты» имеет следующие расширения такие, как:

- создать чат.

Прецедент варианта использования «Чаты» имеет следующие альтернативные потоки такие, как:

- создать чат с несуществующим пользователем.

От актера «Администратор» идут следующие прямые ассоциации к прецедентам вариантам использования таким, как:

- создание профиля пользователя;
- просмотр активности пользователя.

Также актер «Администратор» связан обобщением с актером «Пользователя».

Прецедент «Создание профиля пользователя» включает:

- удаление профиля пользователя.

Прецедент «Создание профиля пользователя» имеет следующие альтернативные потоки такие, как:

- создание профиля пользователя, не соответствующего требованиям;
- удаление не существующего профиля.

Прецедент «Просмотр активности пользователя» включает:

- выбор даты для просмотра активности пользователя;
- выбор пользователя для просмотра активности.

Прецедент «Просмотр активности пользователя» имеет следующие альтернативные потоки такие, как:

- просмотр активности не существующего пользователя;
- ошибочный выбор даты просмотра.

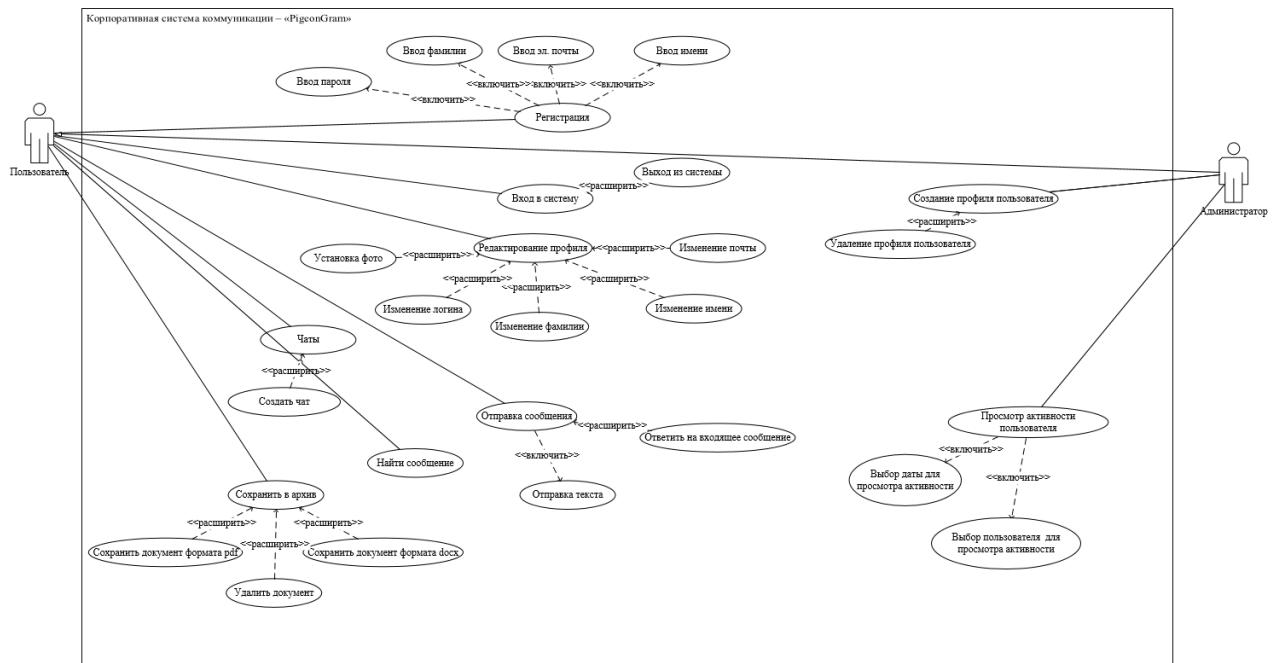


Рис. 2.1.1. Диаграмма вариантов использования корпоративного мессенджера

Диаграмма вариантов использования программного обеспечения наглядно демонстрирует отношения между актерами и вариантами использования для лучшего понимания механики работы и автоматизирования системы.

Для того чтобы сформировать представление работы продукта – то, как компоненты корпоративного мессенджера взаимодействуют друг с другом, благодаря чему пользователь получает желаемое, рассмотрим диаграмму компонентов клиент-серверного приложения (рис. 2.1.2).

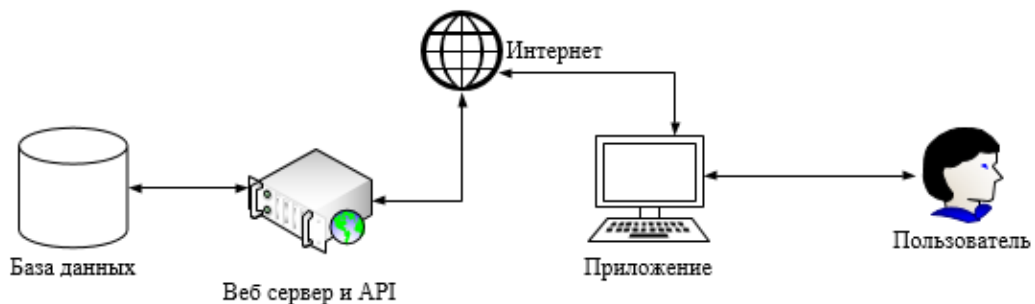


Рис. 2.1.2. Диаграмма компонентов корпоративного мессенджера

На основе предложенной выше схемы составим подробно архитектуру системы корпоративной коммуникации «PigeonGram». Архитектура разрабатываемого приложения представлена на рис. 2.1.3.

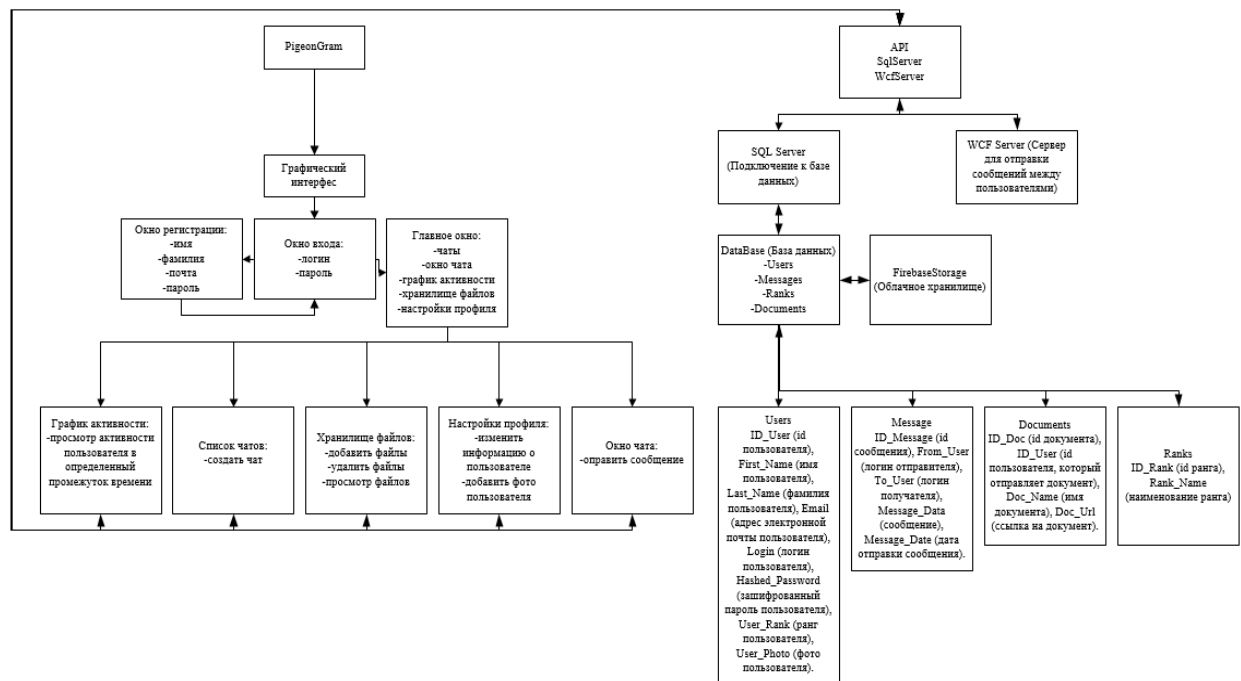


Рис. 2.1.3. Архитектура корпоративного мессенджера

Проект состоит из следующих компонентов:

- пользовательский интерфейс;
- API;
- SQL Server;
- локальная база данных;
- облачное хранилище;
- WCF Server.

Пользователь взаимодействует с мессенджером через пользовательский интерфейс, который взаимодействует с API, в последствии соединяясь с общим хранилищем, SQL Server (а затем локальной базой данных) и WCF Server.

В общем хранилище пользователь может сохранить файлы, которые будут доступны другим пользователям. Файлы сохраняются в локальной базе данных и облачном хранилище.

Через SQL Server программа получает доступ к базе данных и управляет работой в этой части архитектуры.

Благодаря взаимодействию с WCF Server происходит передача сообщений между двумя клиентами, то есть двумя пользователями.

В данном параграфе были представлены различные варианты изображения архитектуры корпоративного мессенджера, проделанный анализ будет использован для лучшего понимания механики работы и автоматизирования системы.

2.2. Моделирование основных сценариев системы

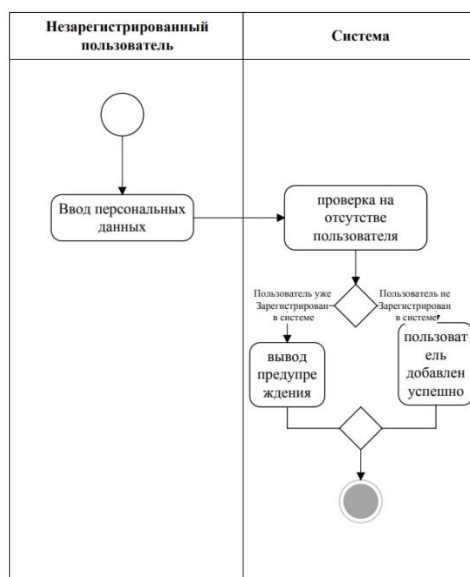


Рис. 2.2.1. Диаграмма деятельности. Регистрация пользователя

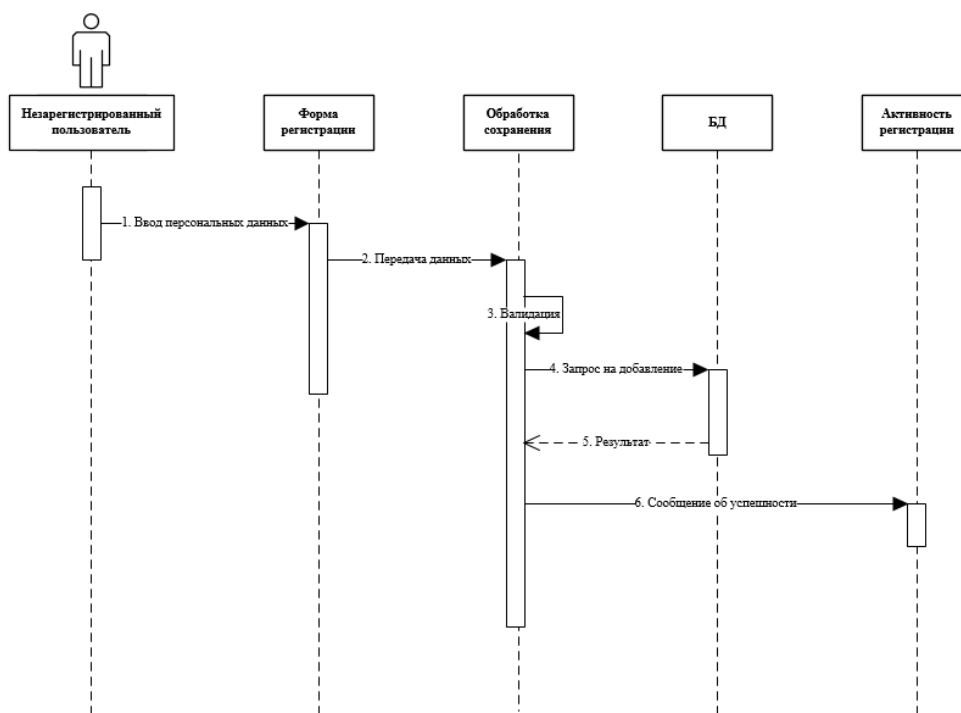


Рис. 2.2.2. Диаграмма последовательности. Регистрация пользователя

На рис. 2.2.1-2.2.2 представлены диаграммы последовательности и деятельности на прецедент «Регистрация пользователя». Регистрация проходит следующим образом: незарегистрированный пользователь вводит персональные данные в форму регистрации. Данные передаются на сервер, где происходит валидация. После проверки правильности введенных данных делается запрос на добавление пользователя в БД. БД сообщает результат, после

выводится сообщение об успешности.

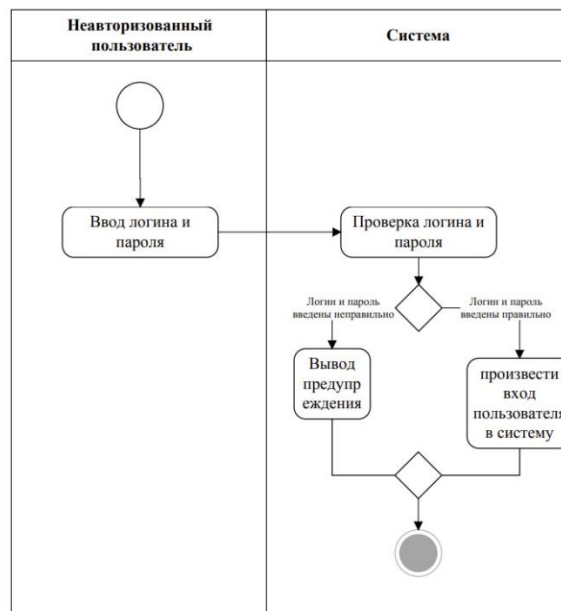


Рис. 2.2.3. Диаграмма деятельности. Авторизация пользователя

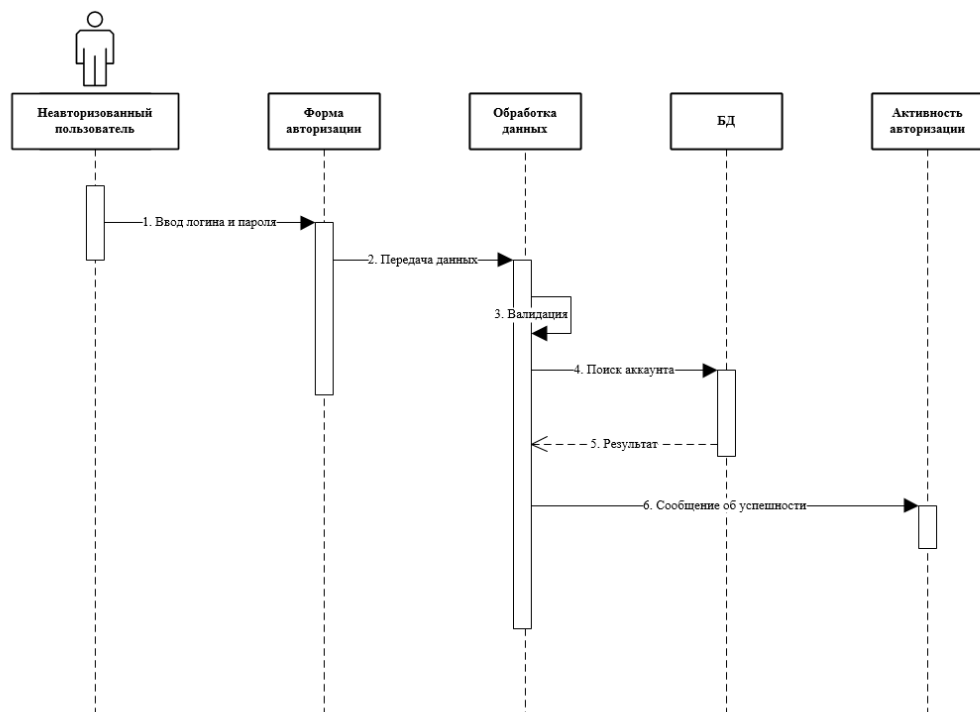


Рис. 2.2.4. Диаграмма последовательности. Авторизация пользователя

На рис. 2.2.3-2.2.4 представлены диаграммы последовательности и деятельности на прецедент «Авторизация пользователя». Авторизация проходит следующим образом: неавторизованный пользователь вводит логин и пароль в форму авторизации. Данные передаются на сервер, где происходит валидация. После проверки правильности введенных данных происходит поиск аккаунта в БД. БД сообщает результат, после выводится сообщение об

успешности.

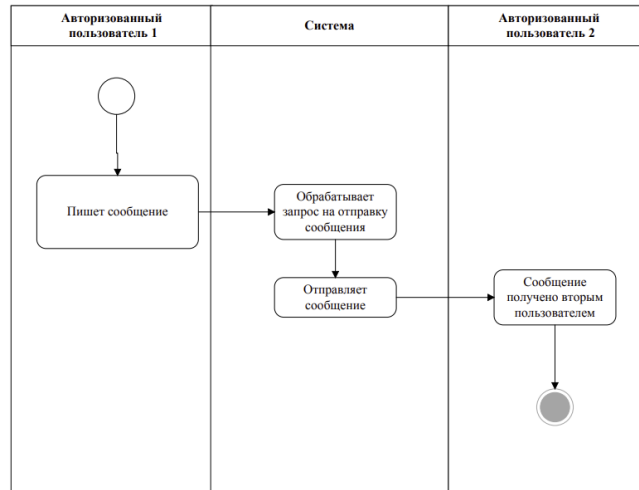


Рис. 2.2.5. Диаграмма деятельности. Отправка сообщений

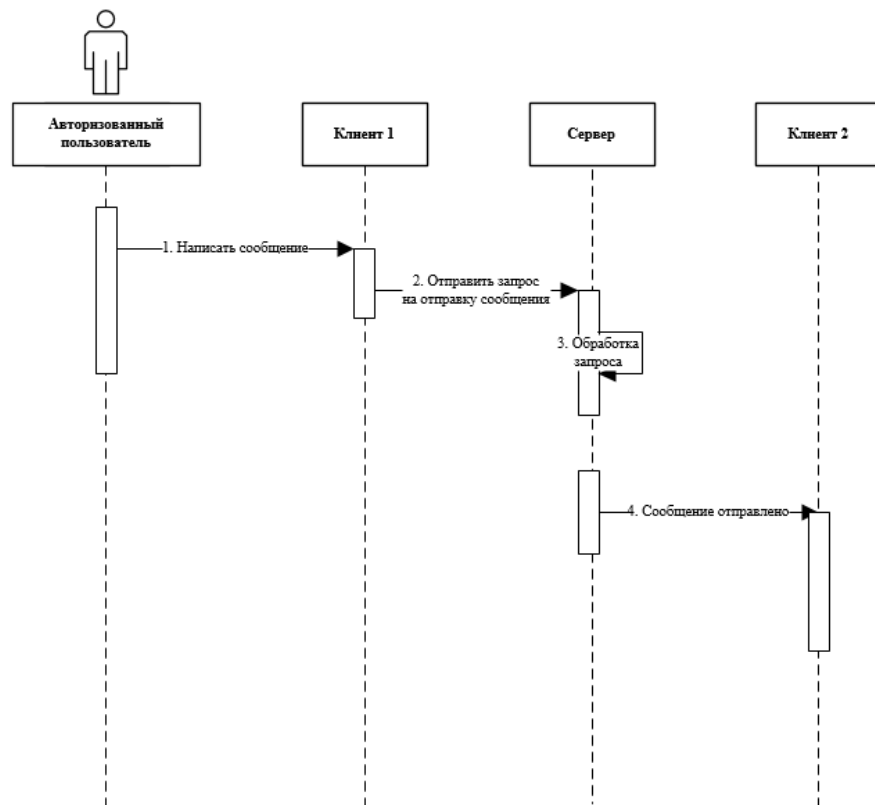


Рис. 2.2.6. Диаграмма последовательности. Отправка сообщений

На рис. 2.2.5-2.2.6 представлены диаграммы деятельности и последовательности на прецедент «Отправка сообщений». Отправка происходит по следующему алгоритму: авторизованный пользователь пишет сообщение в интерфейсе клиента (под вопросом как назвать). Затем клиент делает запрос на отправку сообщения серверу. После обработки сервер отправляет сообщение другому клиенту.

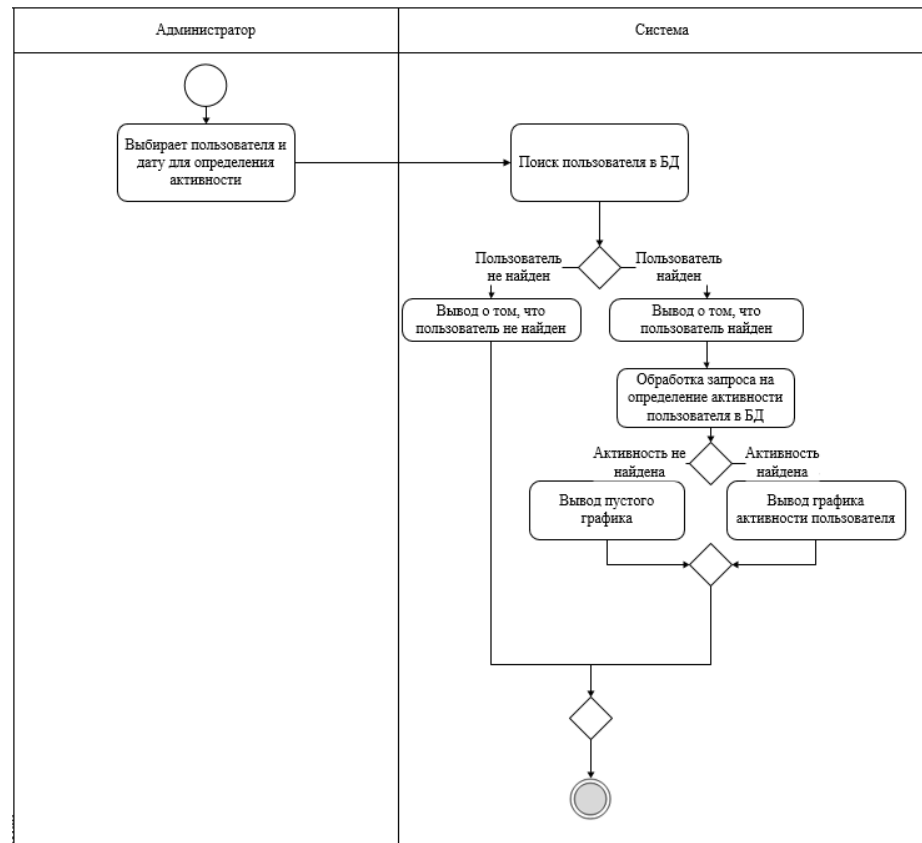


Рис. 2.2.7. Диаграмма деятельности. Вывод графика активности пользователя

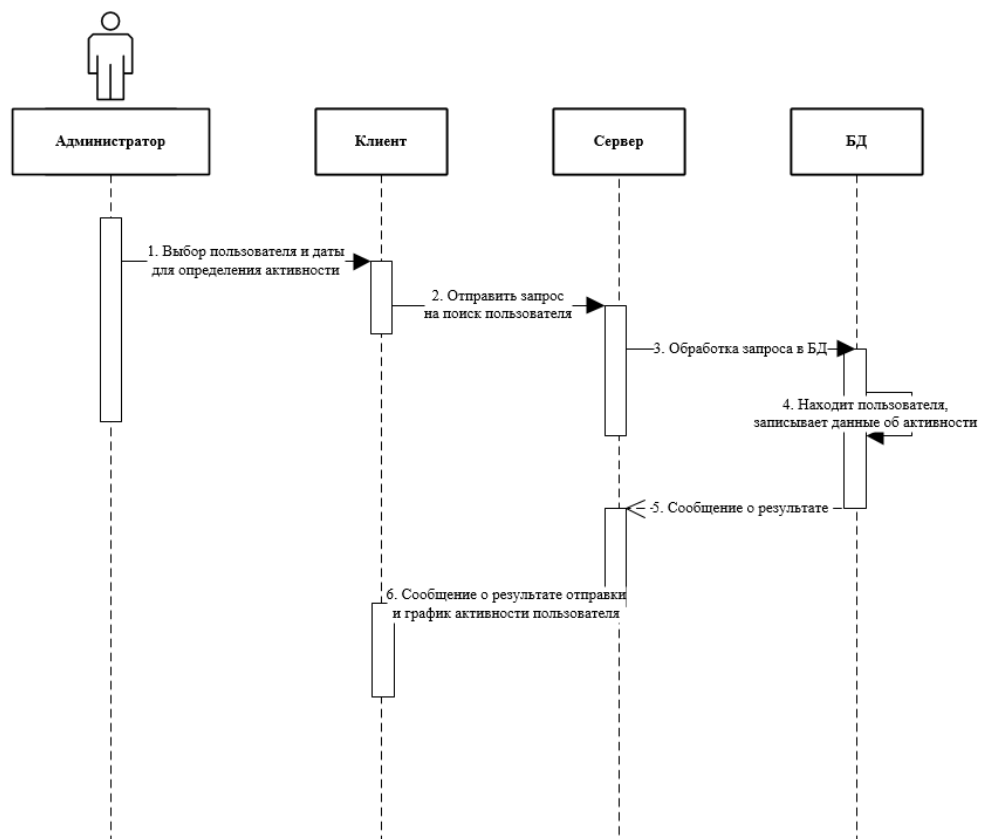


Рис. 2.2.8. Диаграмма последовательности. Вывод графика активности пользователя

На рис. 2.2.7-2.2.8 представлены диаграммы деятельности и последовательности на прецедент «Просмотр активности пользователя». Администратор выбирает через интерфейс

клиента пользователя, активность которого интересуют, и даты. Клиент делает серверу запрос на поиск пользователя, сервер перенаправляет запрос к БД. БД ищет пользователя и сохраняет данные об его активности, сообщает о результате серверу и отображает график активности пользователя.

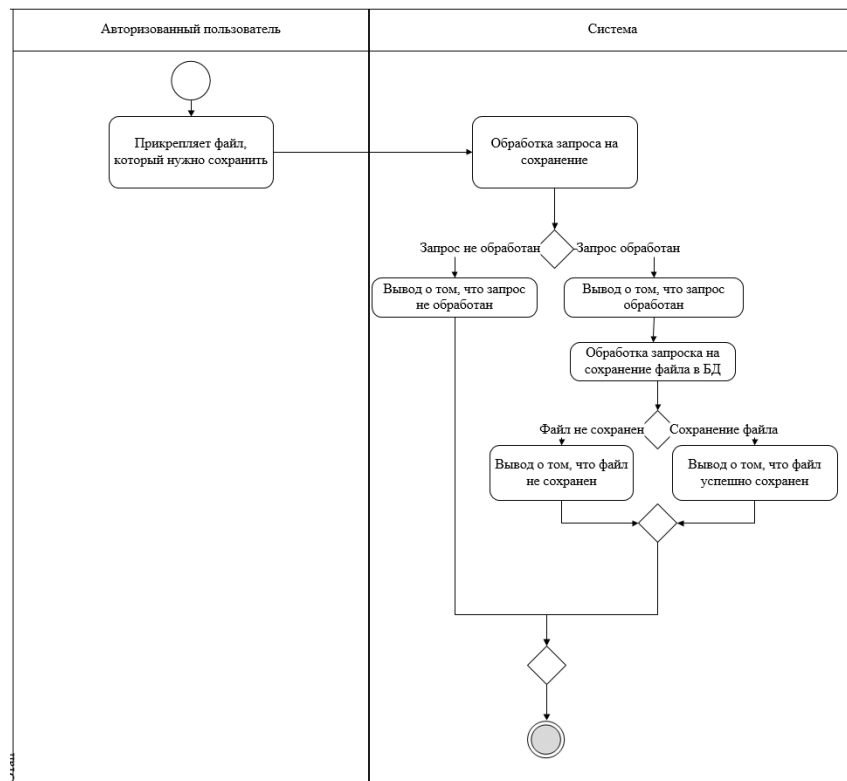


Рис. 2.2.9. Диаграмма деятельности. Сохранение файлов в общем хранилище

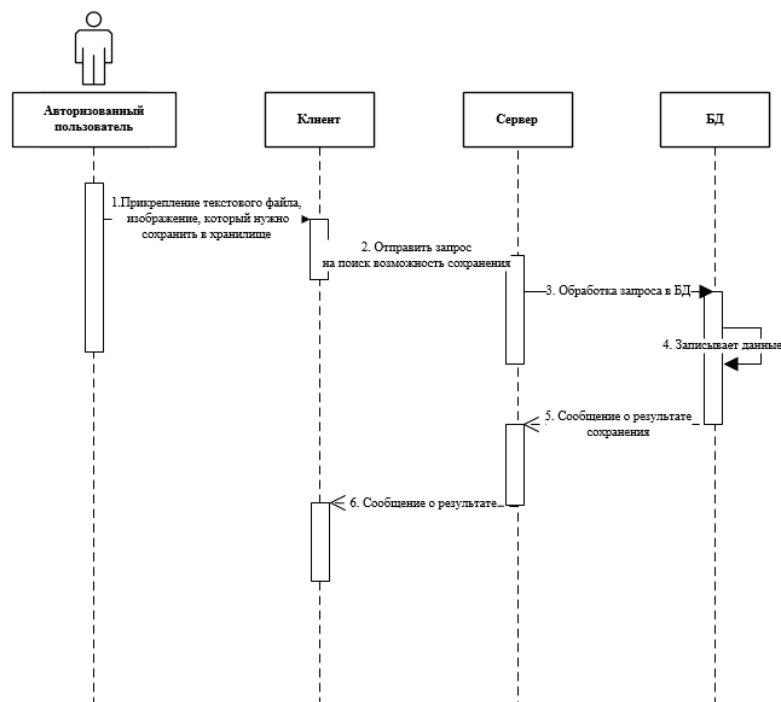


Рис. 2.2.10. Диаграмма последовательности. Сохранение файлов в общем хранилище

На рис. 2.2.9-2.2.10 представлены диаграммы деятельности и последовательности на

прецедент «Сохранение файлов в общем хранилище». Авторизованный пользователь прикрепляет через интерфейс клиента сообщение, которое ему сохранить. Клиент делает серверу запрос на возможность операции, сервер перенаправляет запрос к БД. БД сохраняет данные, сообщает о результате работы серверу и клиенту.

Также для моделирования основных сценариев системы корпоративного мессенджера «PigeonGram» использовалась методология моделирования бизнес-процессов IDEF, чтобы грамотно описать общую концептуальную модель системы и ее взаимодействие с внешней средой. Диаграммы IDEF представлены далее на рис. 2.2.11-2.2.13.

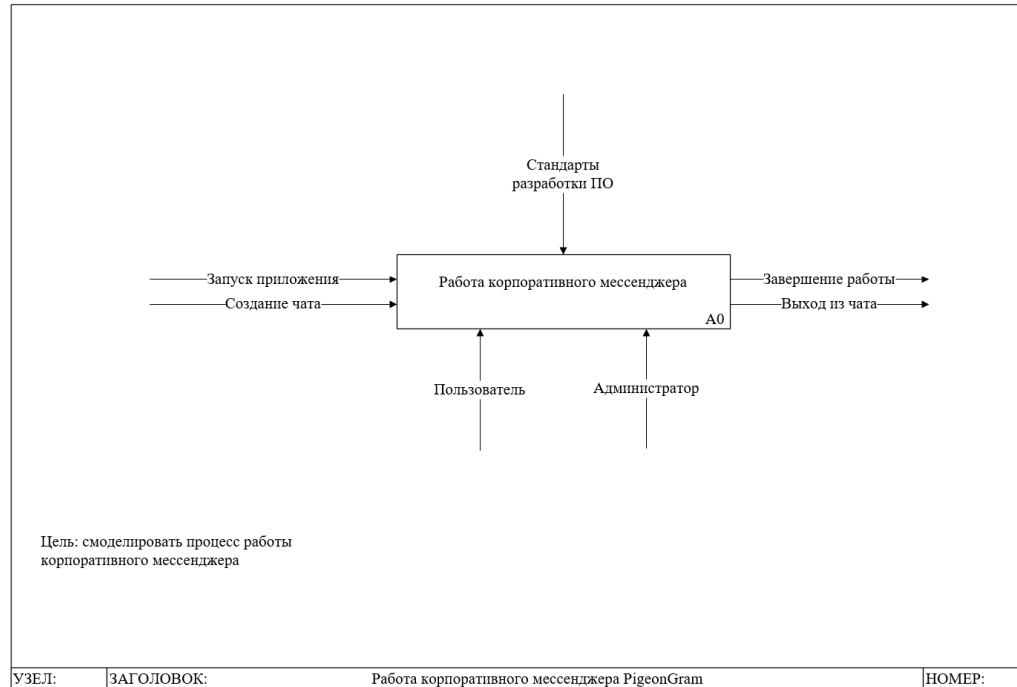


Рис. 2.2.11. Диаграмма процесса «Работа корпоративного мессенджера» первого уровня IDEF0

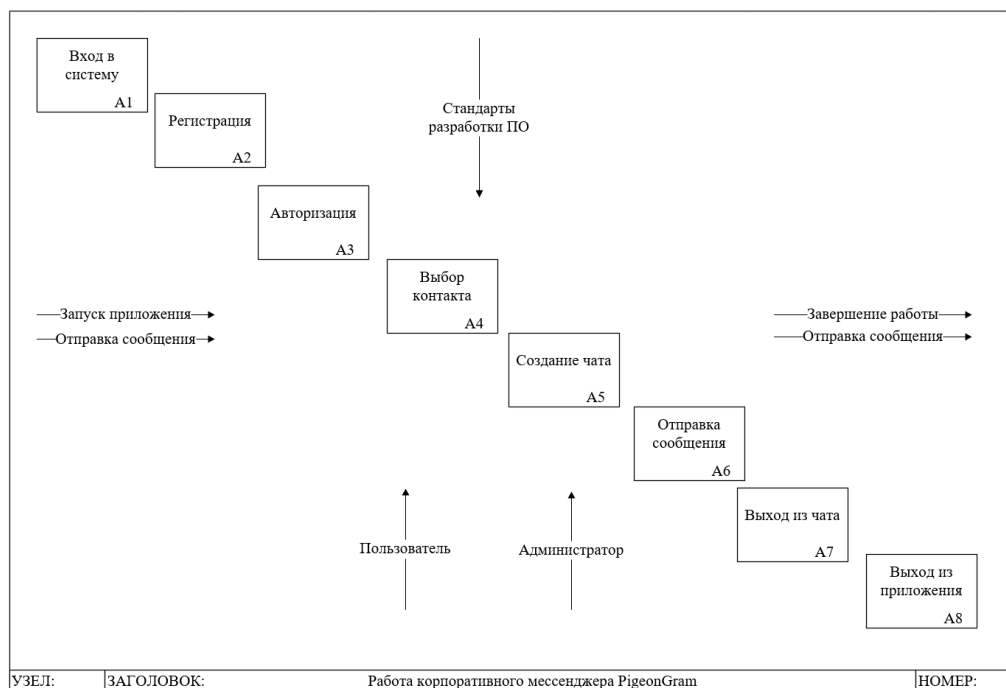


Рис. 2.2.12. Диаграмма процесса «Работа корпоративного мессенджера» второго уровня IDEF0

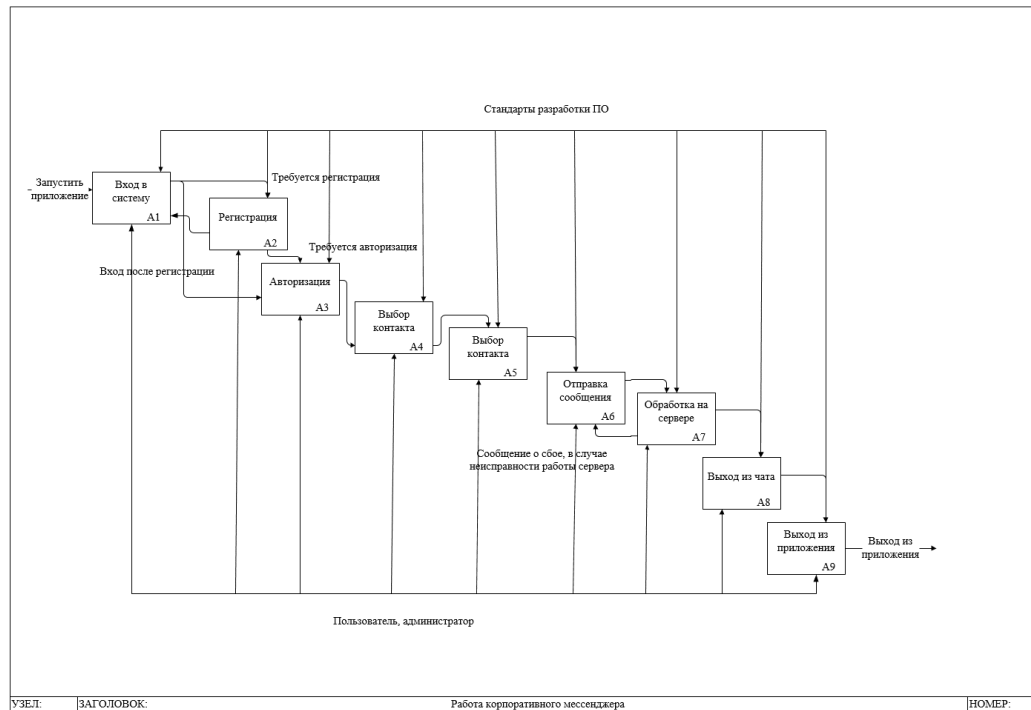


Рис. 2.2.13. Диаграмма процесса «Работа корпоративного мессенджера» третьего уровня IDEF0

Выводом моделирования основных сценариев системы можно выделить, что каждый элемент программного продукта выполняет полезную и логичную функцию для успешной работы всего приложения.

2.3. Проектирование графического интерфейса пользователя

В ходе разработки графического интерфейса пользователя был использован язык XAML. Применительно к WPF (а также к Silverlight) данный язык используется прежде всего для создания пользовательского интерфейса декларативным путем. XAML - не является обязательной частью приложения, мы вообще можем обходиться без него, создавая все элементы в файле связанного с ним кода на языке C#. Однако использование XAML все-таки несет некоторые преимущества:

- возможность отделить графический интерфейс от логики приложения, благодаря чему над разными частями приложения могут относительно автономно работать разные специалисты: над интерфейсом - дизайнеры, над кодом логики – программисты;
- компактность, понятность, код на XAML относительно легко поддерживать.

При компиляции приложения в Visual Studio код в xaml-файлах также компилируется в бинарное представление кода xaml, которое называется BAML (Binary Application Markup Language). И затем код baml встраивается в финальную сборку приложения - exe или dll-файл.

При создании кода на языке C#, чтобы были доступны определенные классы, подключаются пространства имен с помощью директивы using, например, using System.Windows;.

Чтобы задействовать элементы в XAML, также подключаются пространства имен.

На рис. 2.3.1 представлены интерфейс и основные функции корпоративного мессенджера PigeonGram.

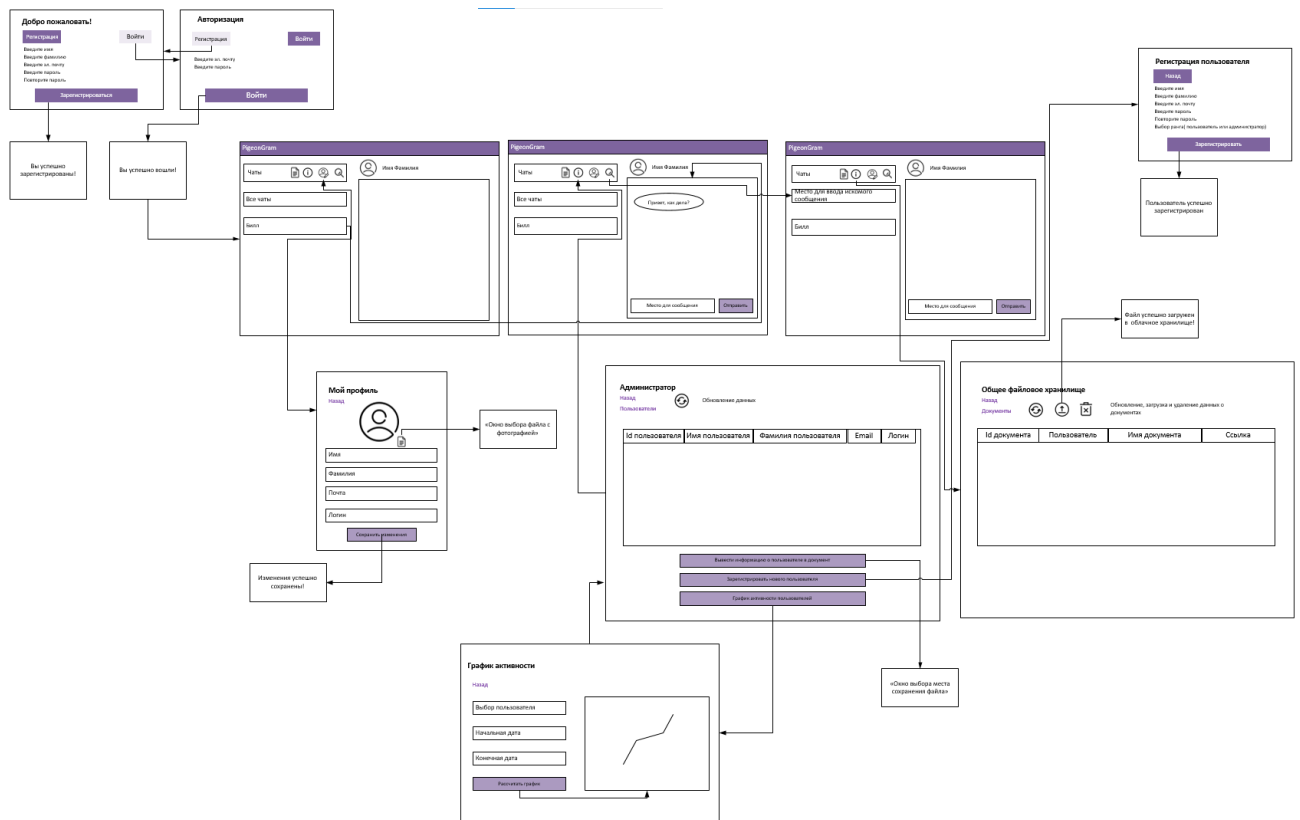


Рис. 2.3.1. Интерфейс приложения

На рис. 2.3.2 представлен непосредственно интерфейс главного окна приложения. Окно содержит список чатов, на которые может нажать пользователь. Посередине представлен открытый чат. Под окном чата представлено место для ввода сообщения и кнопка «Отправить». Над списком чатов присутствуют кнопки, благодаря которым можно открыть облачное хранилище, узнать информацию о активности пользователя, изменить информацию о пользователе, найти сообщение. Также присутствуют значки закрытия, сворачивания и расширения окна. Все приложение выполнено в оттенках синего, фиолетового и голубого цветов.

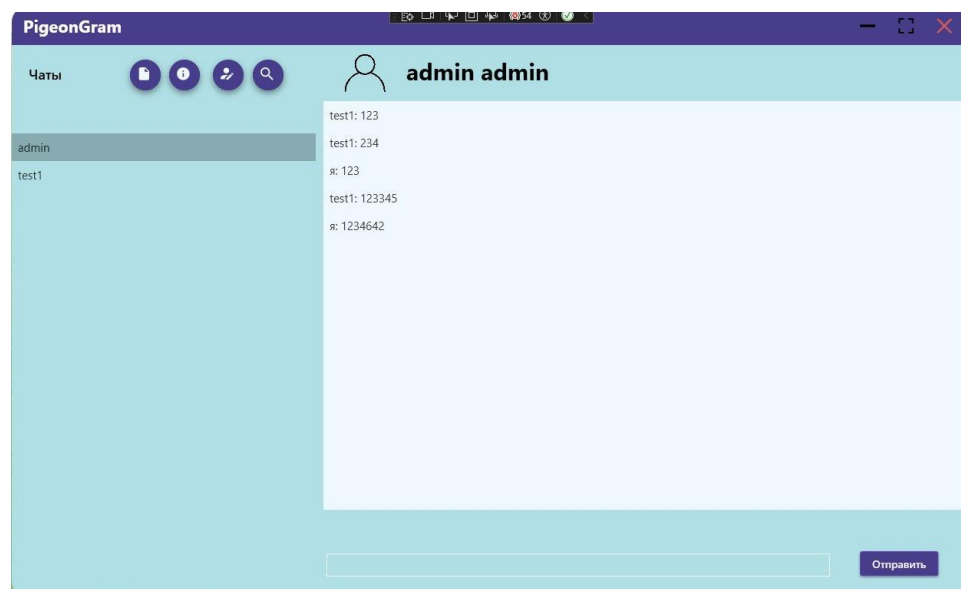


Рис. 2.3.2. Вид главного окна

На рис. 2.3.3 показано окно, в котором можно работать с облачным хранилищем.

Пользователь может выбрать загружаемый документ формата .docx или .pdf и поместить его в общее файловое хранилище. После нажатия на кнопку «Обновить» появится информация о идентификационном номере документа, пользователе, кто загрузил документ, наименование и ссылка.

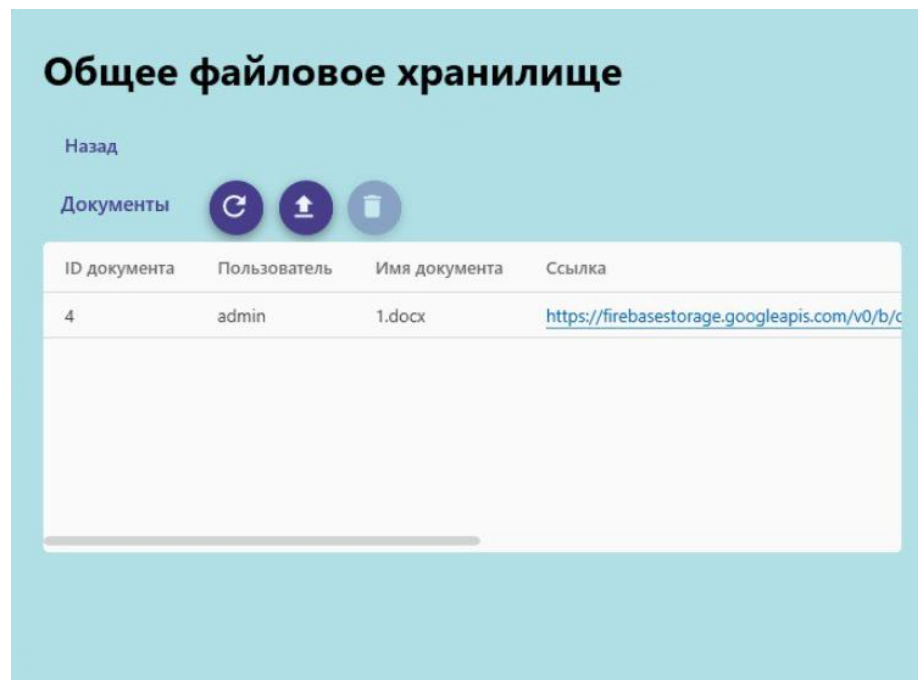


Рис. 2.3.3. Общее файловое хранилище

На рис. 2.3.4 показана реализация окна администратора. Здесь администратор может видеть информацию о зарегистрированных пользователях, может перейти на страницу регистрации нового пользователя или страницу, демонстрирующую график активности пользователей.

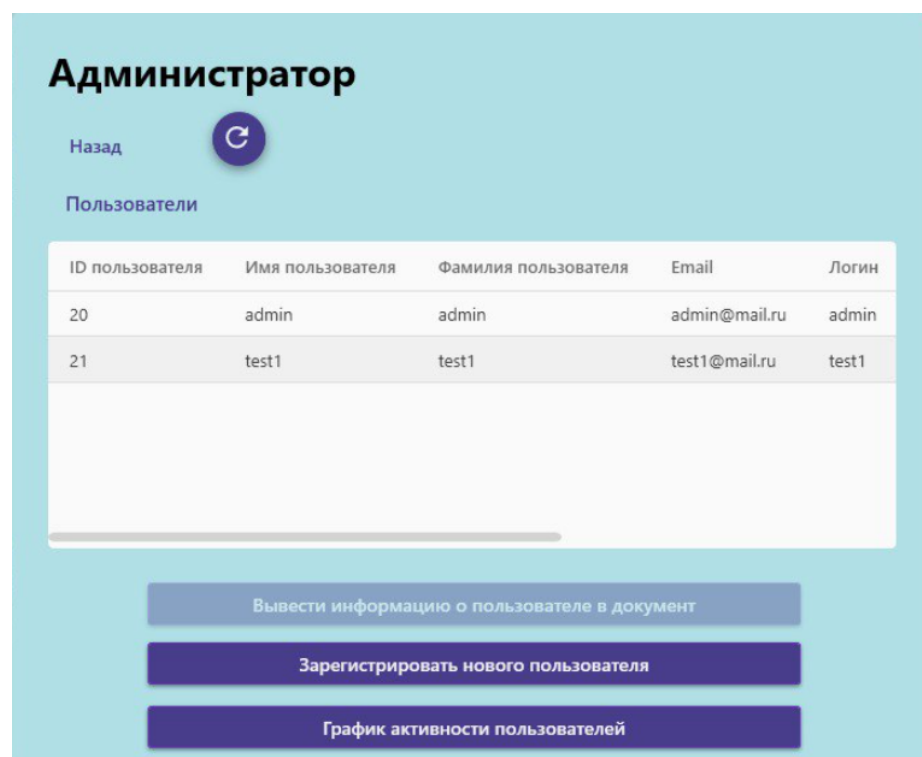


Рис. 2.3.4. Окно администратора

На рис. 2.3.5 показана реализация функции демонстрации активности пользователя за определенный промежуток времени.

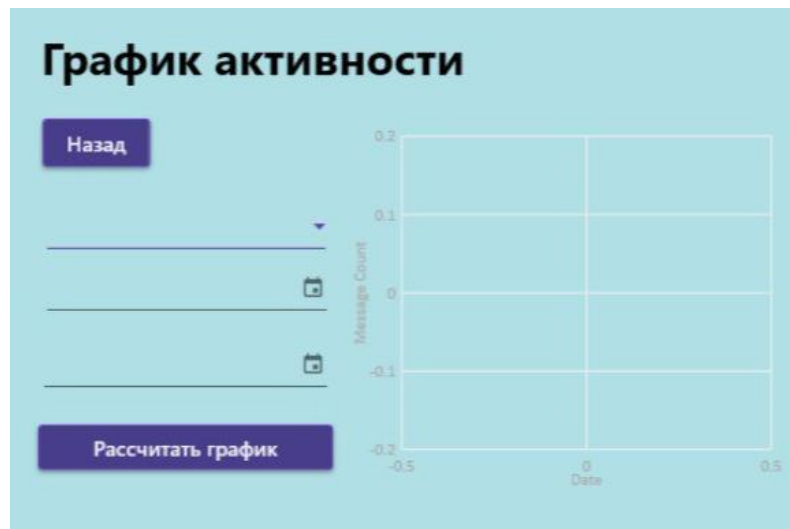


Рис. 2.3.5. График активности пользователя

На рис. 2.3.6. показана реализация окна регистрации. Пользователь вводит свои имя и фамилию, также электронную почту и пароль. После этого он может нажать на кнопку «Зарегистрироваться» и перейти на главный экран.

Рис. 2.3.6. Окно регистрации

На рис. 2.3.7 показана реализация окна входа. Пользователь вводит свою почту, с которой ранее был зарегистрирован и пароль, нажимает на кнопку «Войти» и, после проверки корректности введенных данных, переходит на следующий экран.

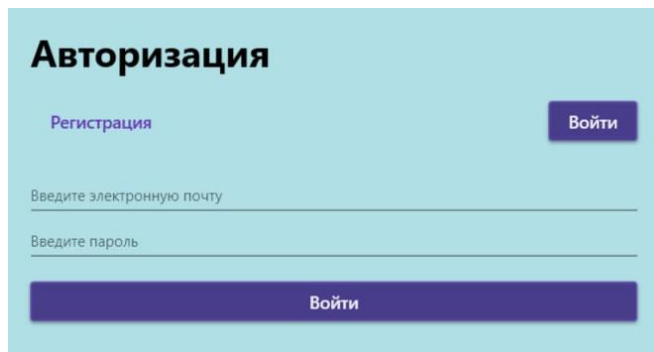


Рис. 2.3.7. Окно входа

На рис. 2.3.8 показана реализация изменения данных профиля. Пользователь может добавить фото для своего аккаунта и изменить имя, фамилию, почту или логин.

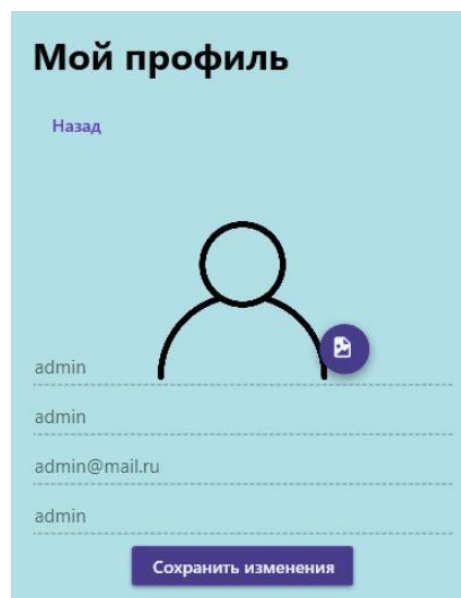


Рис. 2.3.8. Изменение данных профиля

На рис. 2.3.9 показано окно администратора, с помощью которого он может зарегистрировать нового пользователя. Администратор заполняет имя, фамилию, электронную почту, логин нового пользователя. Дважды вписывает пароль и выбирает ранг, администратор или пользователь.

Рис. 2.3.9. Регистрация пользователя

Выводами по проектированию графического интерфейса являются то, что разрабатываемый продукт обладает удобным и понятным интерфейсом, отвечает требованиям пользователей и бизнес-процессов. Вся структура и навигация продумана с учетом поставленных требований.

2.4. Проектирование и разработка модели данных

В предметной области можно выделить следующие сущности:

- сущность «Пользователи»;
- сущность «Сообщения»;
- сущность «Ранги»;
- сущность «Документы».

В ходе разработки было выделено 4 сущности и 4 связи типа «один ко многим», а именно:

- столбец User_Rank в таблице Users будет ссылаться на ID_Rank из таблицы Ranks;
- столбец Login в таблице Users будет ссылаться на From_User из таблицы Messages;
- столбец Login в таблице Users будет ссылаться на To_User из таблицы Messages;
- столбец ID_User в таблице Users будет ссылаться на ID_User в таблице Documents.

Далее представлена диаграмма сущность-связь для базы данных реализованного программного продукта.

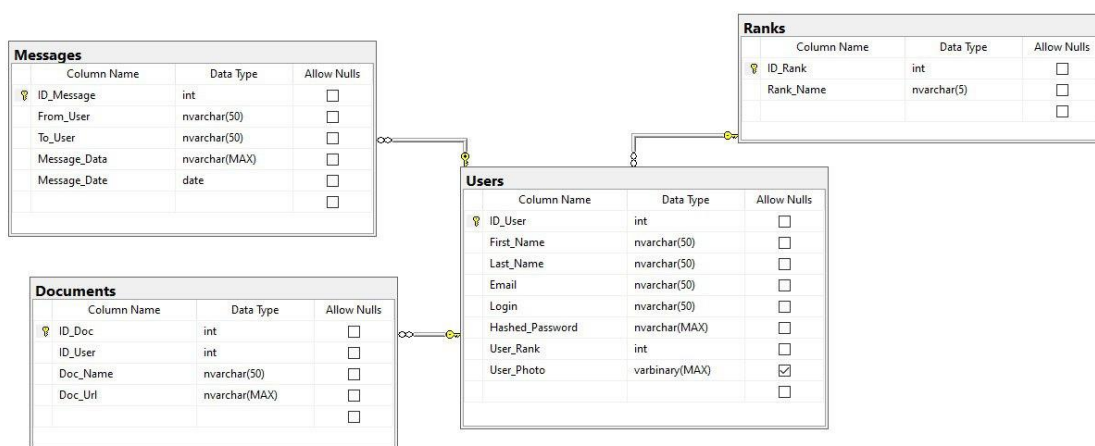


Рис. 2.4.1. ER-диаграмма

Разрабатываемая БД предназначена для хранения всей долговременной информации системы, является реляционной и реализована при помощи СУБД MS SQL Server.

Ниже приведено описание назначений и свойств каждого поля таблиц в БД:

LAPTOP-S3L918JB\S...abase - dbo.Users*			
Column Name	Data Type	Allow Nulls	
ID_User	int	<input type="checkbox"/>	
First_Name	varchar(50)	<input type="checkbox"/>	
Last_Name	varchar(50)	<input type="checkbox"/>	
Email	varchar(50)	<input type="checkbox"/>	
Login	varchar(50)	<input type="checkbox"/>	
Hashed_Password	varchar(50)	<input type="checkbox"/>	
User_Rank	int	<input type="checkbox"/>	
User_Photo	image	<input checked="" type="checkbox"/>	

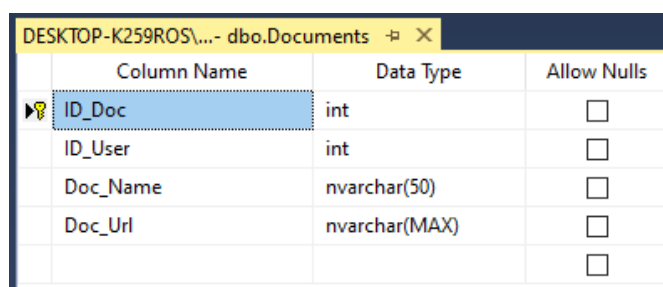
Рис. 2.4.2. Таблица «Users» – Пользователи

В данной таблице хранятся данные о пользователях – ID_User (id пользователя), First_Name (имя пользователя), Last_Name (фамилия пользователя), Email (адрес электронной почты пользователя), Login (логин пользователя), Hashed_Password (зашифрованный пароль пользователя), User_Rank (ранг пользователя), User_Photo (фото пользователя). Ни одно поле не может иметь нулевого значения. Id пользователя является первичным ключом.

DESKTOP-K259ROS\...se - dbo.Messages			
Column Name	Data Type	Allow Nulls	
ID_Message	int	<input type="checkbox"/>	
From_User	nvarchar(50)	<input type="checkbox"/>	
To_User	nvarchar(50)	<input type="checkbox"/>	
Message_Data	nvarchar(MAX)	<input type="checkbox"/>	
Message_Date	date	<input type="checkbox"/>	

Рис. 7.3. Таблица «Messages» – Сообщения

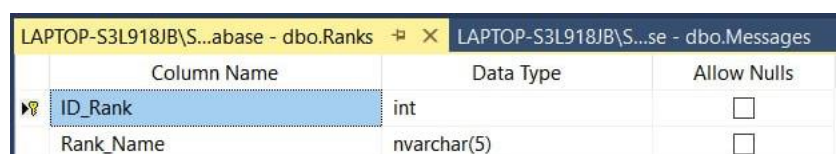
В данной таблице хранятся данные о сообщениях – ID_Message (id сообщения), From_User (логин отправителя), To_User (логин получателя), Message_Data (сообщение), Message_Date (дата отправки сообщения). Id является первичным ключом. Все поля не могут иметь нулевого значения.



Column Name	Data Type	Allow Nulls
ID_Doc	int	<input type="checkbox"/>
ID_User	int	<input type="checkbox"/>
Doc_Name	nvarchar(50)	<input type="checkbox"/>
Doc_Url	nvarchar(MAX)	<input type="checkbox"/>

Рис. 8.4. Таблица «Chats» – Чаты

В данной таблице хранятся данные о документах – ID_Doc (id документа), ID_User (id пользователя, который загрузил этот документ), Doc_Name (наименование документа), Doc_Url (ссылка на документ). Id является первичным ключом. Все поля не могут иметь нулевого значения.



Column Name	Data Type	Allow Nulls
ID_Rank	int	<input type="checkbox"/>
Rank_Name	nvarchar(5)	<input type="checkbox"/>

Рис. 9.5. Таблица «Ranks» – Ранги

В данной таблице хранятся данные о чатах – ID_Rank (id ранга), Rank_Name (наименование ранга). id ранга является первичным ключом. Все поля не могут иметь нулевого значения.

Разработана высококачественная база данных с целью оптимизации регистрации, авторизации пользователей, хранения данных и опправки сообщений.

Выводы второй главы:

Подводя промежуточные итоги, разработано следующее:

- представлен вариант использования системы;
- спроектирована архитектура системы;
- смоделирован графический интерфейс системы;
- реализована база данных программного продукта.

3. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

Поскольку программный продукт является клиент-серверным приложением, он состоит из двух частей – сервера и клиента, взаимодействие которых осуществляется после успешной регистрации и авторизации пользователя. Ниже представлены описания и листинги кода реализации функций и методов.

3.1. Реализация регистрации и авторизации пользователя

В методе Authorisation происходит попытка найти пользователя с заданным логином Login в списке пользователей users. Если пользователь не найден (переменная user равна null), то вызывается метод IsUserInDatabase, который осуществляет проверку наличия пользователя в базе данных по его логину и хешированному паролю. Если пользователь не найден в списке, но присутствует в базе данных, то в методе добавляется новый пользователь с указанным логином и хешированным паролем, и возвращается значение true, что означает успешную аутентификацию. Если пользователь найден в списке (переменная user не равна null), то сразу возвращается значение true, так как пользователь уже существует и считается успешно аутентифицированным.

```
public bool Authorisation(string Login, string hashedPass)
{
    ServerUser user = users.FirstOrDefault(u => u.Login == Login);

    if (user == null)
    {
        if (!IsUserInDatabase(Login, hashedPass))
        {
            return false;
        }

        // Пользователь не найден в списке, добавляем его
        user = new ServerUser { Login = Login, Password = hashedPass, operationContext =
OperationContext.Current };
        users.Add(user);

        return true; // Возвращаем true после добавления пользователя
    }

    return true; // Возвращаем true, так как пользователь был найден в списке
}
//проверяет наличие юзера в бд
```

В методе Registration создается объект ServerUser с заполненными данными: firstName, lastName, email, login, password, roleRank. Эти данные используются для создания нового пользователя. Загружается изображение по указанному пути и читается в байтовый массив imageBytes. Формируется SQL-запрос для вставки новой записи о пользователе в базу данных. Запрос содержит значения полей для вставки. Устанавливается соединение с базой данных через SqlConnection с помощью строки подключения connectionString. Открывается соединение, и создается команда SqlCommand для выполнения SQL-запроса с параметром UserPhoto для изображения пользователя. Значение параметра UserPhoto задается как байтовый массив imageBytes. Выполняется команда ExecuteNonQuery(), которая вставляет новую запись в таблицу Users базы данных. Проверяется количество затронутых строк. Если одна строка вставлена

успешно, то:

- новый пользователь добавляется в список users;
- метод возвращает true, указывая на успешную регистрацию пользователя;
- если не удалось вставить запись (количество затронутых строк не равно 1), то метод возвращает false, указывая на неудачную регистрацию.

```
public bool Registration(string firstName, string lastName, string email, string login, string
password)
{
    ServerUser user = new ServerUser()
    {
        First_Name = firstName,
        Last_Name = lastName,
        Email = email,
        Login = login,
        Password = password,
        operationContext = OperationContext.Current
    };
    //дефолт картинка для всех юзеров(смени путь обязательно)
    string path = "/Assets/user with no photo.png"; //"C:\\Users\\samsung\\Pictures\\user with no
photo.png";
    byte[] imageBytes = File.ReadAllBytes(path);

    string query = $"INSERT INTO Users (First_Name, Last_Name, Email, Login,
Hashed_Password, User_Rank, User_Photo) " +
        $"VALUES ('{firstName}', '{lastName}', '{email}', '{login}', '{password}', '2',
@UserPhoto)";
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@UserPhoto", imageBytes);
            int rowsAffected = command.ExecuteNonQuery();
            if (rowsAffected == 1)
            {
                //добавление в серверный список
                users.Add(user);
                return true;
            }
            else return false;
        }
    }
}
```

Таким образом, данные методы выполняют добавление нового пользователя в базу данных и в список пользователей, используя предоставленные данные и изображение по умолчанию, благодаря чему, в последующем, пользователь может заходить в приложение под своим логином и паролем.

3.2. Реализация клиентской части

3.2.1. Функция отправки сообщения

По заданным именам пользователей UserFrom и UserTo находятся объекты ServerUser отправителя и получателя соответственно в списке пользователей users. Проверяется, что и отправитель, и получатель существуют в списке пользователей. Если у получателя есть активный контекст операции (operationContext) (он подключен), то:

- создается соединение с базой данных через SqlConnection с использованием строки подключения connectionString;
- формируется запрос SQL для добавления сообщения в таблицу Messages, указывающий от кого (From_User), кому (To_User), само сообщение (Message_Data) и дату сообщения (Message_Date);
- значения параметров запроса устанавливаются из данных отправителя, получателя, сообщения и текущей даты;
- выполняется команда ExecuteNonQuery(), которая вставляет новую запись в таблицу сообщений;
- если сообщение было успешно добавлено в базу данных, то сообщение передается получателю через его callback-канал с информацией о времени, отправителе и содержании сообщения.
- если получатель не найден, выводится сообщение об ошибке.
- если у отправителя есть активный контекст операции, то отправителю также передается сообщение через его callback-канал с информацией о времени и содержании сообщения.
- если отправитель и/или получатель не найдены, выводится сообщение об ошибке.

GetCallbackChannel – вызывает канал экземпляру клиента, вызвавшему текущую операцию (вызов канала экземпляру клиента свидетельствует о взаимодействии между сервером и конкретным клиентским приложением для передачи информации или управления операциями.)

```
public void SendMsg(string UserFrom, string UserTo, string MessageData)
{
    ServerUser sender = users.FirstOrDefault(u => u.Login == UserFrom);
    ServerUser receiver = users.FirstOrDefault(u => u.Login == UserTo);

    if (sender != null && receiver != null)
    {
        if (receiver.operationContext != null)
        {
            receiver.operationContext.GetCallbackChannel<IServerChatCallback>()
                .MsgCallback($"{ DateTime.Now.ToShortTimeString() } | { sender.Login } : { MessageData }");

            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                connection.Open();

                string query = "INSERT INTO Messages (From_User, To_User, Message_Data, Message_Date) " +
                    "VALUES (@From_User, @To_User, @Message_Data, @Message_Date)";

                SqlCommand command = new SqlCommand(query, connection);

                command.Parameters.AddWithValue("@From_User", sender.Login);
                command.Parameters.AddWithValue("@To_User", receiver.Login);
```

```

        command.Parameters.AddWithValue("@Message_Data", MessageData);
        command.Parameters.AddWithValue("@Message_Date",    DateTime.Now);    //
Добавляем текущую дату как дату сообщения

        int rowsAffected = command.ExecuteNonQuery();

        if (!(rowsAffected > 0))
        {
            Console.WriteLine("error");
        }
    }
}
else
{
    Console.WriteLine($"Пользователь {UserTo} не был найден. Сообщение не было
отправлено.");
}

if (sender.operationContext != null)
{
    sender.operationContext.GetCallbackChannel<IServerChatCallback>()
        .MsgCallback($"{DateTime.Now.ToShortTimeString()} | я: {MessageData}");
}
}
else
{
    Console.WriteLine("error");
}
}

```

На рис. 3.2.1.1 представлен результат работы функции отправки сообщений.

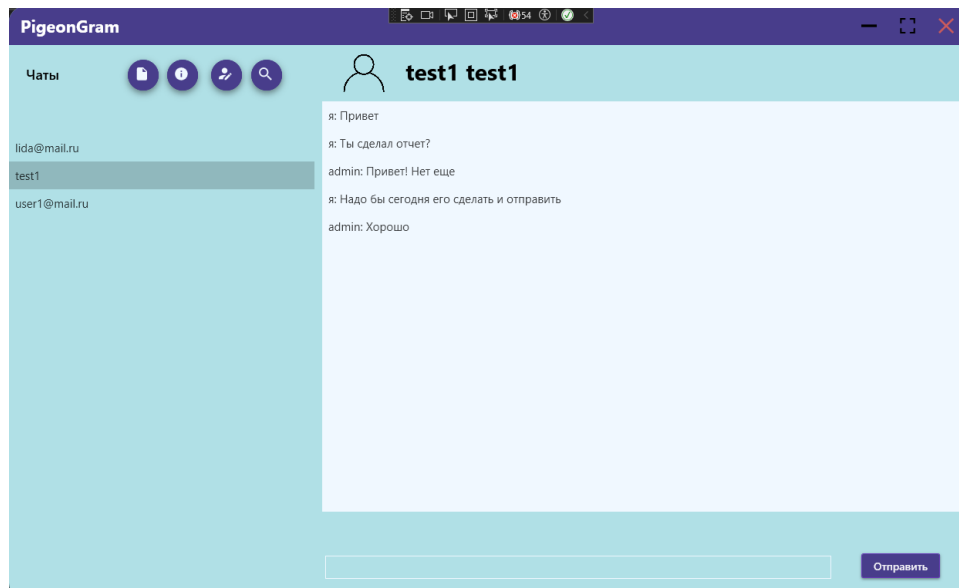


Рис. 3.2.1.1. Результат работы функции отправки сообщения

Таким образом, данный метод обеспечивает отправку сообщений между пользователями в чате, обновление данных в базе данных и отправку сообщений через callback-каналы пользователей.

3.2.2. Метод загрузки файла в облачное хранилище

Сначала определяются константы `FirestoreStorageBucket` и `FirestoreAuthToken`, которые содержат адрес `Bucket` и токен аутентификации `Firestore` соответственно. Метод `UploadFileToFirestoreStorage` объявлен как асинхронный и принимает путь к файлу `filePath` в качестве параметра. В блоке `try` начинается попытка выполнить операцию загрузки файла в `Firestore Storage`. Файл, указанный в `filePath`, открывается на чтение с помощью `File.OpenRead(filePath)`. Создается задача `task`, которая запускает асинхронную операцию. Внутри этой задачи происходит следующее: создается экземпляр объекта `storage` класса `FirestoreStorage`, используя константы `FirestoreStorageBucket` и `FirestoreAuthToken`. Далее вызывается метод `PutAsync` для загрузки файла в `Firestore Storage` по указанному пути `files/ + имя файла из filePath`. После успешной загрузки файла в `Firestore Storage`, выводится сообщение об успешной операции. Затем программа ждет завершения выполнения задачи, используя `await task`. Если в блоке `try` возникает исключение (ошибка), перехватывается и программа переходит в блок `catch`. В блоке `catch` выводится сообщение об ошибке при загрузке файла в `Firestore Storage` с указанием деталей исключения.

```
public async Task UploadFileToFirestoreStorage(string filePath)
{
    try
    {
        var stream = File.OpenRead(filePath);

        var task = Task.Run(async () =>
        {
            var storage = new FirestoreStorage(FirestoreStorageBucket, new FirestoreStorageOptions
            { AuthTokenAsyncFactory = () => Task.FromResult(FirestoreAuthToken) });
```

// Загрузка файла

```

        await storage
            .Child("files/" + Path.GetFileName(filePath))
            .PutAsync(stream,
                cancellationToken: default
            );

        MessageBox.Show("Файл успешно загружен в Firebase Storage!");
    });

    await task;
}
catch (Exception ex)
{
    MessageBox.Show($"Произошла ошибка при загрузке файла в Firebase Storage:
{ex.Message}");
}
}

```

Аналогично работает метод удаления файла из облака, только вместо PutAsync вызывается метод DeleteAsync.

```

public async Task DeleteFileFromFirebaseStorage(string fileName)
{
    try
    {
        var storage = new FirebaseStorage(FirebaseStorageBucket, new FirebaseStorageOptions {
            AuthTokenAsyncFactory = () => Task.FromResult(FirebaseAuthToken) });

        // Удаление файла
        await storage
            .Child("files/" + fileName)
            .DeleteAsync();

        MessageBox.Show("Файл успешно удален из Firebase Storage!");
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Произошла ошибка при удалении файла из Firebase Storage:
{ex.Message}");
    }
}

```

На рис. 3.2.2.1 представлен результат работы метод загрузки файла в облачное хранилище.

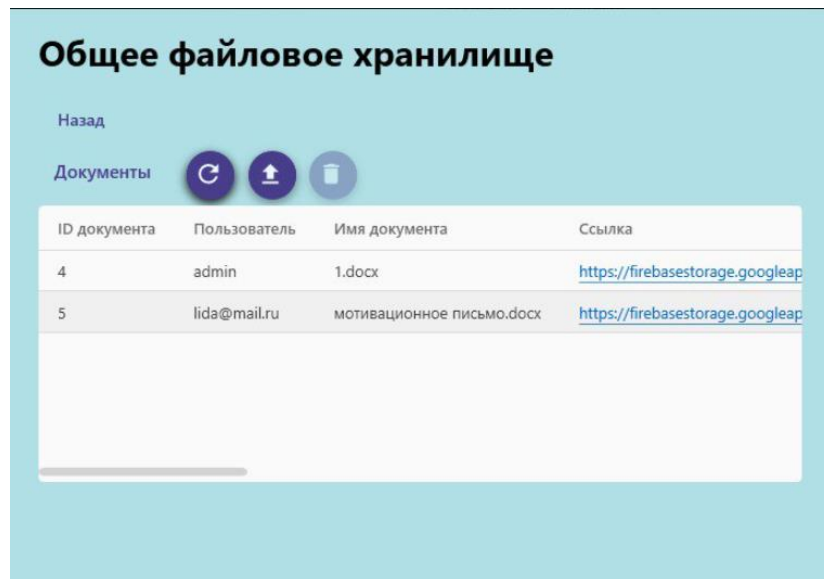


Рис. 3.2.2.1. Результат работы общего файлового хранилища

Таким образом, с помощью метода получения ссылок, пользователь может получить доступ к файлу в облачном хранилище.

3.2.3. Метод сохранения данных о пользователе

Получение выбранной строки данных из таблицы:

- извлекается первый выбранный элемент из таблицы `UserInfoDataGrid` и из него извлекается значение `ID_User`.
- формируется SQL-запрос для получения данных выбранного пользователя:
- строится запрос на выборку данных о пользователе на основе `ID_User`.

Создается команда SQL `command` для выполнения запроса. Считываются данные с помощью `SqlDataReader reader`. Для каждого поля результирующего набора данных формируется строка `message`, содержащая название поля и его значение. Открывается диалоговое окно для сохранения файла с выбором форматов `.docx` или `.pdf`. Если пользователь выбирает сохранение в формате `.docx`, создается новый документ Word с текстом `message`, который затем сохраняется в выбранный файл. Если пользователь выбирает сохранение в формате `.pdf`, создается PDF-документ с текстом `message`, который также сохраняется в выбранный файл. После завершения операций соединение с базой данных закрывается. Для реализации метода использовались библиотеки `Microsoft.Office.Interop.Word` (для формата `.docx`) и `iText` (для формата `.pdf`). На рис. 2.3.8 представлено окно для изменения данных о пользователе.

Таким образом, с помощью метода сохранения данных, администратор может отправить общие данные о пользователе в форматах `.docx` или `.pdf`.

3.3. Реализация серверной части

Рассмотрим серверную часть приложения, благодаря которой осуществляется обмен сообщениями между пользователями. В начале кода происходит подключение необходимых пространств имен. В основной части кода (метод `Main`) происходит следующее:

- создается новый экземпляр класса `ServiceHost`, который представляет хост WCF-службы. Этот экземпляр инициализируется с типом сервиса, который известен как `WCF_Service`;

- затем вызывается метод `Open()` для открытия хоста и начала прослушивания входящих запросов;
- после успешного запуска хоста выводится сообщение "Server has been started..." в консоль;
- далее программа ожидает ввода пользователем, нажатия любой клавиши через `Console.ReadKey()`;
- в случае возникновения исключения в блоке `catch`, ошибка выводится в консоль.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.ServiceModel;
using System.Text;
using System.Threading.Tasks;
using WCF_Library_Server;

namespace ConsoleServer
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                using (var host = new ServiceHost(typeof(WCF_Service)))
                {
                    host.Open();
                    Console.WriteLine("Server has been started...");
                    Console.ReadKey();
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }
        }
    }
}
```

Таким образом, с помощью серверной части, обеспечивается отправка сообщений.

Выводы третьей главы:

Подводя итоги вышесказанному необходимо отметить следующее:

- описан порядок разработки программных модулей;
- реализована регистрация и авторизация пользователя;
- реализована клиентская часть;
- реализована серверная часть;
- предоставлена организация взаимодействия с базой данных системы;
- представлена работа функций клиентской части через интерфейс и код программы.

4. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

4.1. История изменений

Во время разработки корпоративного мессенджера были грамотно распределены силы и время, так как это важно для получения качественного программного продукта. Ниже представлена табл. 4.1.1, в которой описаны этапы разработки.

Таблица 4.1.1

История изменений

Дата	Версия	Автор	Описание изменений
15.02.2024	1.0	Малакаускайте Анна	Создание сервера
25.02.2024	1.1	Преснухина Лидия	Добавление главного окна
22.03.2024	1.3	Малакаускайте Анна, Преснухина Лидия	Консультация с научным руководителем по поводу разработки дипломного проекта
10.04.2024	2.0	Преснухина Лидия	Разработка текстового описания продукта
10.04.2024	2.1	Малакаускайте Анна	Разработка окон регистрации и авторизации
13.04.2024	2.2	Малакаускайте Анна	Реализация связи между сервером и клиентом
15.04.2024	2.3	Преснухина Лидия	Техническое проектирование
23.04.2024	2.4	Малакаускайте Анна	Реализация шифрования продукта
24.04.2024	2.5	Малакаускайте Анна, Преснухина Лидия	Модификация отправки сообщений, изменение функционала программы
07.05.2024	2.6	Малакаускайте Анна, Преснухина Лидия	Добавление функции сохранения файлов в облачное хранилище
11.05.2024	2.7	Преснухина Лидия, Малакаускайте Анна	Техническое проектирование
11.05.2024	2.7	Малакаускайте Анна, Преснухина Лидия	Консультация с научным руководителем по поводу разработки дипломного проекта
15.05.2024	2.8	Малакаускайте Анна	Добавления графика активности пользователей

4.2. Терминология тестирования программного обеспечения

Список терминов и определений, используемых далее в документе и тестировании в целом:

- «Клиент — сервер» — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиент и сервер — это программное обеспечение.
- Стратегия тестирования — это план проведения работ по тестированию системы или ее модуля, учитывающий специфику функциональности и зависимости с другими компонентами системы и платформы.
- Позитивное тестирование — это тестирование с применением сценариев, которые соответствуют нормальному (штатному, ожидаемому) поведению системы.
- Негативное или отрицательное тестирование — это тип тестирования программного

обеспечения (ПО), направленный на проверку того, что система или приложение ведут себя должным образом в негативных ситуациях, то есть, когда они получают недопустимые или неожиданные входные данные.

- Тестирование методом «белого ящика – метод тестирования программного обеспечения, который предполагает, что внутренняя структура/устройство/реализация системы известны тестирующему.

4.3. Стратегия тестирования

Стратегия тестирования ПО заключается в использовании метода тестирования «белого ящика» для анализа и оценки работоспособности программы.

4.4. Определение объектов тестирования

Таблица 4.4.1

Определение объектов тестирования

Тесты	Объекты тестирования
Виды тестирования	
Функциональное тестирование (functional testing)	<ul style="list-style-type: none"> • работа системы регистрации пользователя; • работа системы авторизации пользователей
Тестирование производительности (performance testing)	<ul style="list-style-type: none"> • скорость сохранения данных в БД; • скорость отображения данных из БД
Нагрузочное тестирование (load testing)	<ul style="list-style-type: none"> • нахождение 15 авторизованных пользователей в системе
Тестирование совместимости (Compatibility testing)	<ul style="list-style-type: none"> • корректное отображение программы в разных версиях ОС Windows (8, 10); • корректное отображение программы на различных соотношениях сторон экрана
Различные виды тестов	
Позитивные тесты	<ul style="list-style-type: none"> • регистрация пользователей, ввод данных в форме регистрации корректных значений; • авторизация пользователей, ввод данных в форме авторизации корректных значений
Негативные тесты	<ul style="list-style-type: none"> • регистрация пользователей, ввод данных в форме регистрации некорректных значений; • авторизация пользователей, ввод данных в форме авторизации некорректных значений
Уровни тестирования	
Модульное тестирование	<ul style="list-style-type: none"> • тестирование функций пользователя: регистрация, авторизация, отправка сообщений
Интеграционное тестирование	<ul style="list-style-type: none"> • переход с стартового модуля на модуль

Тесты	Объекты тестирования
	авторизации; <ul style="list-style-type: none"> • переход с модуля авторизации на модуль регистрации.
Системное тестирование	<ul style="list-style-type: none"> • попытка обойти средства защиты программы; • скорость отклика программы; • способность системы эффективно использовать оперативную память.

4.5. Архитектура тестируемой системы

Правило функционирования архитектуры клиент-серверного приложения заключается в обработке запросов, происходящих на другом компьютере. Архитектура системы представлена на рисунке 2.1.3.

Достоинством реализации данной модели является то, что программный код клиентской и серверной части отделен. Также к достоинству можно отнести то, что к компьютерам, на которых установлен клиент, низкие требования, поскольку все основные вычисления происходят на сервере.

К недостаткам можно отнести высокую стоимость серверного оборудования, поскольку на сервере происходят основные вычисления.

Наконец выделим то, что архитектура позволяет равномерно разбить нагрузку и поделить функционал между клиентской и серверной частью системы.

Схема промышленного стенда будет реализована в офисной обстановке.

Схема тестового стенда будет реализована на личном компьютере.

Таблица 4.5.1

Конфигурация промышленного стенда

Промышленное оборудование	
Ресурсы	Детали
Сервер 1	
CPU	Intel core i7
RAM	8 ГБ
HDD	512 ГБ
ОС	Windows 10
Дополнительно	Разрешение 1920x1080
Клиент 1	
CPU	Intel core i7
RAM	8 ГБ
HDD	512 ГБ
ОС	Windows 10
Дополнительно	Разрешение 1920x1080

Таблица 4.5.2

Конфигурация тестового стенда

Промышленное оборудование	
Ресурсы	Детали
Сервер 1	
CPU	Intel core i7

Промышленное оборудование	
Ресурсы	Детали
RAM	8 ГБ
HDD	512 ГБ
ОС	Windows 10
Дополнительно	Разрешение 1920x1080
Клиент 1	
CPU	Intel core i7
RAM	8 ГБ
HDD	512 ГБ
ОС	Windows 10
Дополнительно	Разрешение 1920x1080

Требования к тестируемому окружению:

- оборудование среднего сегмента рынка;
- версия ОС Windows 10;
- используемая БД MS SQL Server.

4.6. Описание процесса тестирования

Таблица 4.6.1

Тестовые сценарии с указанием конкретных тестовых наборов

Название тестового сценария	Тестовый набор	Ожидаемый результат	Актуальный результат	Результат тестирования
Проверка отправки сообщения	Графический интерфейс ПО	Сообщение отправлено	Функция ПО работает корректно	Успешно
Проверка регистрации пользователя	Админ, Админ, admin@mail.ru 12345,12345	Пользователь зарегистрирован	Пользователь зарегистрирован	Успешно
Проверка входа пользователя	admin@mail.ru, 12345	Пользователь успешно вошел в систему	Пользователь успешно вошел в систему	Успешно
Проверка входа пользователя	admin@mail.ru, 123	Пользователь не смог войти в систему (пароль неверный)	Пользователь не смог войти в систему (пароль неверный)	Успешно
Проверка регистрации пользователя	Пустое поле	Пользователь не зарегистрирован (пустое поле)	Пользователь не зарегистрирован	Успешно
Проверка регистрации пользователя	Админ, Админ, admin, 12345,12345	Пользователь не зарегистрирован (неверный адрес почты)	Пользователь не зарегистрирован	Успешно

Название тестового сценария	Тестовый набор	Ожидаемый результат	Актуальный результат	Результат тестирования
Проверка удаления файлов из общего хранилища	Пользовательский интерфейс	Пользователь не смог удалить файл, сообщение об ограничении доступа	Пользователь не смог удалить файл, сообщение об ограничении доступа	Успешно
Проверка удаления файлов из общего хранилища	Пользовательский интерфейс с правами администратора	Пользователь смог удалить файл	Пользователь смог удалить файл	Успешно
Проверка сохранения данных о пользователе в БД	admin@mail.ru, 12345	Данные о пользователе сохранены	Данные о пользователе сохранены	Успешно
Проверка скорости сохранения изменений	admin@mail.ru, 12345	Данные о пользователе сохранены за допустимое время	Данные о пользователе сохранены за 13 мс	Успешно
Запуск программы в разных версиях ОС	Запуск программы в Windows 8	Программа запущена и работает корректно	Программа запущена и работает корректно	Успешно
Запуск программы в разных версиях ОС	Запуск программы в Windows 10	Программа запущена и работает корректно	Программа запущена и работает корректно	Успешно
Изменение данных о пользователе	Добавление фото профиля	Фото добавлено. Сообщение об успешном сохранении изменений	Фото добавлено. Сообщение об успешном сохранении изменений	Успешно
Нагрузочное тестирование	Использование программы одновременно 15 пользователями	Программа работает медленнее, но корректно	Программа работает медленнее, но корректно	Успешно

Выводы четвертой главы:

Все вышесказанное доказывает, что:

- описана история изменения системы;
- разработана стратегия тестирования программного обеспечения;
- определены объекты тестирования;
- приведена архитектура тестируемой системы и описание процесса тестирования.

5. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

5.1. Определение защищаемых информационных активов

Информация уже давно стала одним из основных ресурсов, обеспечивающих деятельность компании. Более того, информация, зачастую, сама является исходным сырьем или результатом производства – товаром, предлагаемым конечному потребителю. Информация стала активом компании, который нуждается, в каком-то измерении, учете и выражении в общепринятых количественных показателях.

Для обеспечения качественной функциональности разрабатываемой корпоративной системы коммуникации требуется идентифицировать защищаемые информационные активы, оценить угрозы, уязвимости и последствия, а также рассчитать их стоимость.

Исходный код программы – самый важный актив, для которого необходима особая защита. От состояния исходного кода зависит вся работа программного продукта. Незащищенный код – это огромные риски для разработчика и высокая вероятность потерять созданное программное обеспечение. Защита кода должна вестись как на стадии его создания, так и тестирования, эксплуатации. Она необходима для предотвращения атак со стороны киберпреступников, которые могут использовать уязвимости и ошибки кода для получения доступа к информационной системе и кражи личной информации, такой как: логин, пароль, личная информация пользователя и т.д., которые тоже являются активами программного продукта. Угрозами являются: внедрение вредоносного кода; недостаточная обработка ошибок; недостаточное обновление. В случае утечки информации произойдет снижение репутации программного продукта на рынке. Уязвимыми местами являются используемые функции и методы, через которые осуществляется доступ к другим активам. Недостаточная аутентификация и авторизация может позволить злоумышленникам получить несанкционированный доступ к системе или выполнить действия от имени другого пользователя. Недостаточно сильные пароли, отсутствие двухфакторной аутентификации, неправильная настройка прав доступа – все это может стать причиной угрозы для программных кодов. Неправильное управление исключениями, отсутствие проверки на наличие ошибок, некорректная обработка исключительных ситуаций – все это может стать причиной угрозы для программных кодов.

База данных – второй по значимости информационный актив. Средства обеспечения безопасности баз данных защищают данные и сетевую инфраструктуру. Защита нужна для предотвращения утечки информации и сбоев работы продукта. Основными угрозами являются: SQL-инъекции, то есть внедрение вредоносного кода на языке SQL в формы, поля ввода, другие сервисы; взлом паролей; несанкционированный доступ к данным.

5.2. Вид и содержание персональных данных

Сбор, хранение и обработка информации, а именно персональных данных, позволяет работать программному продукту исправно, выполняя необходимый функционал. Определим, какие персональные данные обрабатываются при этом.

При создании аккаунта указывается имя, фамилия, адрес электронной почты. Эти данные обрабатываются для последующего пользования личным аккаунтом, входом и выходом из него, отправлением сообщений от своего аккаунта другим пользователям. Эти данные являются общедоступными, хранятся в базе данных в течение года, как обнаруживается отсутствие использования аккаунта, только после этого удаляются. Каждые 6 месяцев пользователю приходит запрос на подтверждение персональных данных.

Во время использования средства коммуникации отправляются различные виды данных: сообщения, фотографии, документы. Этот вид персональных данных хранится на устройствах пользователей. На сервере он сохраняется в зашифрованном виде временно, в течение двух дней, если происходит сбой в отправке или сообщение пересылается.

5.3. Модель угроз ПНД

Модель угроз персональных данных – это структурный анализ расчетов рисков и последствий для информации, обрабатываемой в рассматриваемом программном продукте.

Ниже представлены цели реализации угроз для программного продукта (табл. 5.3.1), обзор которых необходим для составления таблицы, расположенной в приложении 6.

Таблица 5.3.1

Цели реализации угроз

№	Виды нарушителей	Возможные цели реализации угроз безопасности информации			Соответствие целей видам риска (ущерба) и возможным негативным последствиям
		Нанесение ущерба физическому лицу	Нанесение ущерба юридическому лицу, индивидуальному предпринимателю	Нанесение ущерба государству в области обеспечения обороны страны, безопасности государства и правопорядка, а также в социальной, экономической, политической, экологической сферах деятельности	
ВН 1	Отдельные физические лица (хакеры)	+ (желание самореализоваться)	+ (получение финансовой выгоды за счет кражи и коммерческой тайны)	-	У1 (нарушение личной, семейной тайны, утрата чести и доброго имени) У2 (утечка коммерческой тайны; потеря клиентов)
ВН 2	Конкурирующие организации	+ (месть за ранее совершенные действия)	+ (получение финансовой выгоды за счет кражи и коммерческой тайны)	-	-

№	Виды нарушителей	Возможные цели реализации угроз безопасности информации			Соответствие целей видам риска (ущерба) и возможным негативным последствиям
		Нанесение ущерба физическому лицу	Нанесение ущерба юридическому лицу, индивидуальном у предпринимател ю	Нанесение ущерба государству в области обеспечения обороны страны, безопасности государства и правопорядк а, а также в социальной, экономической, политическо й, экологическ ой сферах деятельность и	
ВН 3	Разработчики программных , программно-аппаратных средств	+ (желание самореализоваться)	+ (передача информации о юридическом лице третьим лицам)	+ (внедрение дополнительных функциональ ных возможносте й в программные или программно-аппаратные средства на этапе разработки при вступлении в сговор со специальным и службами иностранных государств)	У2 (недополучено ожидаемой прибыли) У3 (нарушение функциониров ания государственно го органа, дискредитация деятельности органа государственно й власти)

№	Виды нарушителей	Возможные цели реализации угроз безопасности информации			Соответствие целей видам риска (ущерба) и возможным негативным последствиям
		Нанесение ущерба физическому лицу	Нанесение ущерба юридическому лицу, индивидуальном у предпринимател ю	Нанесение ущерба государству в области обеспечения обороны страны, безопасности государства и правопорядк а, а также в социальной, экономическ ой, политическо й, экологическ ой сферах деятельност и	
ВН 4	Поставщики вычислитель ных услуг, услуг связи	-	+(передача информации о юридическом лице третьим лицам)	-	-
ВН 5	Авторизован ные пользователи систем и сетей	+(непреднамеренн ые, неосторожные или неквалифицирова нные действия)	-	-	У1 (финансовый, иной материальный ущерб физическим лицам)

№	Виды нарушителей	Возможные цели реализации угроз безопасности информации			Соответствие целей видам риска (ущерба) и возможным негативным последствиям
		Нанесение ущерба физическому лицу	Нанесение ущерба юридическому лицу, индивидуальным предпринимателям	Нанесение ущерба государству в области обеспечения обороны страны, безопасности государства и правопорядка, а также в социальной, экономической, политической, экологической сферах деятельности	
ВН 6	Системные администраторы и администраторы безопасности	+ (месть за ранее совершенные действия)	+ (любопытство или желание самореализации)	+ (получение финансовой или иной материальной выгоды при вступлении в сговор с преступной группой)	У1 (финансовый, иной материальный ущерб физическим лицам) У2 (невозможность заключения договоров, соглашений) У3 (утечка информации ограниченного доступа)
ВН 7	Бывшие работники (пользователи)	+ (месть за ранее совершенные действия)	+ (получение финансовой выгоды за счет кражи и коммерческой тайны)	-	-

5.4. Возможные сценарии угроз

Таблица 5.4.1

Возможные сценарии угроз

УБИ	Наименование угрозы	Актуальность угрозы	Потенциал нарушителя	Тактика	Основные техники
6	Угроза внедрения кода или данных	Актуальна	Внешний нарушитель с низким потенциалом	T1	T1.1-T1.4
15	Угроза доступа к защищаемым файлам с использованием обходного пути	Актуальна	Внешний нарушитель с низким потенциалом Внутренний нарушитель с низким потенциалом	T3	T3.1-T3.12
27	Угроза искажения вводимой и выводимой на периферийные устройства информации	Актуальна	Внешний нарушитель с высоким потенциалом Внутренний нарушитель с низким потенциалом	T2	T2.1-T2.3
33	Угроза использования слабостей кодирования входных данных	Актуальна	Внешний нарушитель со средним потенциалом Внутренний нарушитель со средним потенциалом	T3	T3.1-T3.12
88	Угроза несанкционированного копирования защищаемой информации	Актуальна	Внешний нарушитель с низким потенциалом Внутренний нарушитель с низким потенциалом	T3	T3.1-T3.12
90	Угроза несанкционированного создания учётной записи пользователя	Актуальна	Внешний нарушитель с низким потенциалом Внутренний нарушитель с низким потенциалом	T4	T4.1-T4.8
115	Угроза перехвата вводимой и выводимой на периферийные устройства информации	Актуальна	Внешний нарушитель с низким потенциалом Внутренний нарушитель с низким потенциалом	T3	T3.1-T3.12

Таблица 5.4.2

Описание тактик

№	Тактика	Основные техники
T1	Внедрение и исполнение вредоносного программного обеспечения в системах и сетях Тактическая задача: получив доступ к узлу сети или системы, нарушитель	T1.1 Автоматический запуск скриптов и исполняемых файлов в системе с использованием пользовательских или системных учетных данных, в том числе с использованием методов социальной инженерии

№	Тактика	Основные техники
	стремится внедрить в его программную среду инструментальные средства, необходимые ему для дальнейших действий	<p>T1.2 Активация и выполнение вредоносного кода, внедренного в виде закладок в легитимное программное и программное-аппаратное обеспечение систем и сетей</p> <p>T1.3 Автоматическая загрузка вредоносного кода с удаленного сайта или ресурса с последующим запуском на выполнение</p> <p>T1.4 Копирование и запуск скриптов и исполняемых файлов через средства удаленного управления операционной системой и сервисами</p>
T2	<p>Управление вредоносным программным обеспечением и (или) компонентами, к которым ранее был получен доступ</p> <p>Тактическая задача: внедрив вредоносное программное обеспечение или обеспечив постоянное присутствие на узле сети, нарушитель стремится автоматизировать управление внедренными инструментальными средствами, организовав взаимодействия скомпрометированным узлом и сервером управления, который может быть размещен в сети Интернет или в инфраструктуре организации</p>	<p>T2.1. Удаленное управление через стандартные протоколы (например, RDP, SSH), а также использование инфраструктуры провайдеров средств удаленного администрирования. Пример: использование средств удаленного управления RMS/teamviewer для создания канала связи и управления скомпрометированной системой со стороны злоумышленников</p> <p>T2.2. Использование штатных средств удаленного доступа и управления операционной системы</p> <p>T2.3. Коммуникация с внешними серверами управления через нестандартные порты на этих серверах, что в некоторых случаях позволяет эксплуатировать уязвимости средств сетевой фильтрации для обхода этих средств</p>
T3	<p>Сбор и вывод из системы или сети информации, необходимой для дальнейших действий при реализации угроз безопасности информации или реализации новых угроз</p> <p>Тактическая задача: в ходе реализации угроз безопасности информации, нарушителю может потребоваться получить и вывести за пределы инфраструктуры большие объемы информации, избежав при этом обнаружения или противодействия</p>	<p>T3.1. Доступ к системе для сбора информации и вывод информации через стандартные протоколы управления (например, RDP, SSH), а также использование инфраструктуры провайдеров средств удаленного администрирования. Пример: использование средств удаленного управления RMS / teamviewer для создания канала связи и управления скомпрометированной системой со стороны злоумышленников</p> <p>T3.2. Доступ к системе для сбора информации и вывод информации через использование штатных средств удаленного доступа и управления операционной системы</p> <p>T3.3. Вывод информации на хорошо известные порты на внешних серверах, разрешенные на межсетевом экране (SMTP/25, HTTP/80, HTTPS/443 и др.)</p> <p>T3.4. Вывод информации на нестандартные порты на внешних серверах, что в некоторых случаях позволяет эксплуатировать уязвимости средств сетевой фильтрации для обхода этих средств</p> <p>T3.5. Отправка данных по известным протоколам управления и передачи данных</p>

№	Тактика	Основные техники
		Т3.6. Отправка данных по собственным протоколам
		Т3.7. Проксирование трафика передачи данных для маскировки подозрительной сетевой активности, обхода правил на межсетевом экране и сокрытия адресов инфраструктуры нарушителей, дублирование каналов связи, обфускация и разделение трафика передачи данных во избежание обнаружения.
		Т3.8. Туннелирование трафика передачи данных через VPN
		Т3.9. Туннелирование трафика управления в поля заполнения и данных служебных протоколов, к примеру, туннелирование трафика управления в поля данных и заполнения протоколов DNS, ICMP или другие
		Т3.10. Вывод информации через съемные носители, в частности, передача данных между скомпрометированной изолированной системой и подключенной к Интернет системой через носители информации, используемые на обеих системах
		Т3.11. Вывод информации через предоставление доступа к файловым хранилищам и базам данных в инфраструктуре скомпрометированной системы или сети, в том числе путем создания новых учетных записей или передачи данных для аутентификации и авторизации имеющихся учетных записей
		Т3.12. Вывод информации путем размещения сообщений или файлов на публичных ресурсах, доступных для анонимного нарушителя (форумы, файлообменные сервисы, фотобанки, облачные сервисы, социальные сети)
Т4	Несанкционированный доступ и (или) воздействие на информационные ресурсы или компоненты систем и сетей, приводящие к негативным последствиям Тактическая задача: достижение нарушителем конечной цели, приводящее к реализации моделируемой угрозы и причинению недопустимых негативных последствий	<p>Т4.1. Несанкционированный доступ к информации в памяти системы, файловой системе, базах данных, репозиториях, в программных модулях и прошивках</p> <p>Т4.2. Несанкционированное воздействие на системное программное обеспечение, его конфигурацию и параметры доступа</p> <p>Т4.3. Несанкционированное воздействие на программные модули прикладного программного обеспечения</p> <p>Т4.4. Несанкционированное воздействие на программный код, конфигурацию и параметры доступа прикладного программного обеспечения</p> <p>Т4.5. Несанкционированное воздействие на программный код, конфигурацию и параметры доступа системного программного обеспечения</p>

№	Тактика	Основные техники
		Т4.6. Несанкционированное воздействие на программный код, конфигурацию и параметры доступа прошивки устройства
		Т4.7. Подмена информации (например, платежных реквизитов) в памяти или информации, хранимой в виде файлов, информации в базах данных и репозиториях, информации на неразмеченных областях дисков и сменных носителей
		Т4.8. Уничтожение информации, включая информацию, хранимую в виде файлов, информацию в базах данных и репозиториях, информацию на неразмеченных областях дисков и сменных носителей

5.5. Организация защиты данных в проекте

Согласно методическим рекомендациям по обеспечению информационной безопасности при создании и эксплуатации открытых репозиториях программного обеспечения составлены принципы, на которых основывается организация защита данных в проекте по разработке корпоративной системы коммуникации.

Основные принципы:

1. Конфиденциальность: защита конфиденциальности персональных данных, а также других важных информационных ресурсов проекта.
2. Целостность: защита от несанкционированного изменения, уничтожения или потери данных.
3. Доступность: обеспечение доступа к данным только для тех пользователей, которым это необходимо, и при этом гарантирование отсутствия несанкционированного доступа.
4. Управляемость: обеспечение возможности контроля и управления доступом к данным, а также контроля за использованием информационных ресурсов проекта.
5. Авторизация и аутентификация: обеспечение идентификации пользователей и проверки их прав доступа к информационным ресурсам проекта.
6. Непрерывность: обеспечение непрерывности функционирования информационных ресурсов проекта в случае возникновения аварийных ситуаций или катастрофических событий.
7. Соблюдение законодательства: соблюдение законодательства о защите персональных данных и других правил и норм, регулирующих обработку и использование информации.

5.6. Защита программного продукта

Безопасность и защита данных в программном продукте являются одними из важных аспектов при разработке. Система корпоративной коммуникации строится на постоянном обмене информацией, которая может быть перехвачена и использована в разных целях, не благоприятных для работы предприятия.

В п. 5.4 приведен анализ актуальных угроз для разрабатываемого программного продукта, поэтому были предприняты следующие меры для оптимизации работы приложения:

1. Аутентификация и авторизация. Создано для предотвращения несанкционированного доступа к функционалу программы и личной информации пользователя.
2. Шифрование данных. Используется для защиты файлов от перехвата и прослушивания третьими лицами. Данные шифруются при передаче на сервер. И в самом коде, за счет того, что он трудно читается.

3. Аудит и отчетность. Это позволяет отслеживать и анализировать действия пользователя, помогает предотвратить угрозу.

4. Управление доступом. В разрабатываемом корпоративном мессенджере есть два уровня доступа: администратор и пользователь. Администратор может управлять аккаунтом пользователя, поэтому в случае подозрительных действий второго, произойдет блокировка или удаление аккаунта.

Вывод пятой главы:

Подводя итоги вышесказанному необходимо отметить следующее:

- определены защищаемые информационные активы;
- разработаны модели угроз для персональных данных;
- определены возможные сценарии угроз;
- описана организация защиты данных в проекте;
- проанализирована защита программного продукта.

6. ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ НА ПРОГРАММНЫЙ ПРОДУКТ

6.1. Руководство пользователя

Документ «Руководство пользователя» относится к пакету эксплуатационной документации. Его основная цель заключается в предоставлении пользователю необходимой информации для самостоятельной работы с программой или автоматизированной системой. Руководство должно ответить на такие вопросы: описание программы, ее возможности, требования для корректного функционирования и действия при отказе системы. Соответственно основными пунктами этого документа являются:

1. Введение.
2. Назначение и условия применения.
3. Подготовка к работе.
4. Описание операций.
5. Аварийные ситуации.
6. Рекомендации по освоению.

С описанным документом можно ознакомиться в приложении 3.

6.2. Руководство администратора

Документ «Руководство администратора» также относится к пакету эксплуатационной документации. В первую очередь, руководство администратора предоставляет инструкции по установке и настройке корпоративного мессенджера. Оно включает в себя выбор нужной платформы для установки, создание аккаунтов для пользователей и определение прав доступа к функциям и информации. Руководство администратора корпоративного мессенджера не только освещает вопросы установки и настройки, но и описывает методы эффективного использования мессенджера для повышения коммуникации и сотрудничества в организации. Благодаря такому руководству, администратор может обеспечить максимальное использование потенциала корпоративного мессенджера и повысить эффективность работы всей организации. Основными пунктами этого документа являются:

1. Назначение программы.
2. Функции программы.
3. Условия выполнения программы.
4. Выполнение программы.
5. Сообщения администратору.

С описанным документом можно ознакомиться в приложении 4.

6.3. Определение качества продукта по метрикам

Таблица 6.3.1

Расчет метрик стилистики и понятности программы

Количество строк программы	9623
Строк в каждом сегменте	70

Кол-во сегментов программы	9	
	Количество комментариев	$F_i = \text{sign}(\frac{N_{ком}}{N_{смп}} - 0.1)$
1 сегмент	3	-1
2 сегмент	2	-1
3 сегмент	12	1
4 сегмент	19	1
5 сегмент	29	1
6 сегмент	3	-1
7 сегмент	3	-1
8 сегмент	72	1
9 сегмент	36	1
$F = \sum_{i=1}^n F_i$		1

Расчет:

Сегмент 1. $F_1 = \text{sgn}(3/70 - 0.1) = -1$

Сегмент 2. $F_2 = \text{sgn}(2/70 - 0.1) = -1$

Сегмент 3. $F_3 = \text{sgn}(12/70 - 0.1) = 1$

Сегмент 4. $F_4 = \text{sgn}(19/70 - 0.1) = 1$

Сегмент 5. $F_5 = \text{sgn}(29/70 - 0.1) = 1$

Сегмент 6. $F_6 = \text{sgn}(3/70 - 0.1) = -1$

Сегмент 7. $F_7 = \text{sgn}(3/70 - 0.1) = -1$

Сегмент 8. $F_8 = \text{sgn}(72/70 - 0.1) = 1$

Сегмент 9. $F_9 = \text{sgn}(36/70 - 0.1) = 1$

Вывод: так как уровень комментированности программы $F=1$, а число сегментов $n=9$, то условие нормального уровня комментированности $F=n$ не выполняется. При $F < n$ уровень комментированности программы считается низким.

Выводы шестой главы:

Подводя итоги вышесказанному необходимо отметить следующее:

- рассчитана метрика стилистики и понятности программы;
- разработано руководство пользователя и администратора.

7. МЕРОПРИЯТИЯ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ И ОХРАНЕ ТРУДА

Одним из важных элементов успешной работы любой организации является соблюдение техники безопасности и охраны труда. Ведь только здоровые и безопасные условия работы сотрудников могут обеспечить высокую производительность и минимизировать возможные риски. Организация мероприятий по технике безопасности и охране труда становится неотъемлемой частью корпоративной культуры, помогая повысить осведомленность сотрудников о правилах охраны труда, опасных ситуациях и методах превентивных мер.

7.1. Характеристика объекта проектирования и условий его эксплуатации

7.1.1. Технические данные проектного устройства

Целью проекта является разработка корпоративной системы коммуникации, представляющая собой приложение, написанное для работы в операционной системе Windows, и сервер, для реализации передачи сообщений от клиента к клиенту.

Согласно ГОСТ 12.1.005-88 ССБТ “Общие санитарно-гигиенические требования к воздуху рабочей зоны” необходимо обеспечить комфортные климатические условия для сотрудников, работающих за компьютером.

В помещении технического персонала в холодное время года должны быть поддерживаемые оптимальные показатели температуры воздуха, равные 22-24°C (допустимые в пределах 18-26°C), относительной влажности 40-60% и скорости движения воздуха не более 0,1 м/с.

В теплое время года оптимальные показатели температуры воздуха составляют 23-25°C (допустимые в пределах 20-30°C), относительной влажности 40-60% и скорости движения воздуха 0,1 м/с (допустимая скорость 0,1-0,2 м/с).

Так как технический персонал проводит большую часть рабочего времени за компьютером, им необходимо иметь специально спроектированное рабочее место. Для этого требуется стул, который можно регулировать по высоте для комфортной работы, подставка для ног с возможностью регулировки высоты, а также рекомендуется расположение монитора на расстоянии 60-100 см от сотрудника. Технический персонал также должен соблюдать правильную осанку при работе за компьютером, чтобы избежать проблем со спиной.

При проектировании рабочего места следует учесть, что фиксированная рабочая поза является физиологически неоправданной, поскольку может привести к профессиональным заболеваниям. Поэтому целесообразно предусмотреть возможность работы как в сидячем, так и в стоячем положении.

7.1.2. Условия эксплуатации проектного объекта

Помещения, где используются ПК, должны иметь естественное и искусственное освещение. Расположение рабочих мест за мониторами для взрослых пользователей в подвальных помещениях не допускается. Естественное освещение обеспечивается через оконные проемы с коэффициентом естественного освещения КЕО не ниже 1,2% в зонах с устойчивым снежным покровом и не ниже 1,5% на остальной территории.

Площадь на одно рабочее место с компьютером для взрослых пользователей должна составлять не менее 4,5 м², если монитор жидкокристаллический, и не менее 6 м² для ЭЛТ-монитора. Объем на одно рабочее место - не менее 20 м³.

Помещения с компьютерами должны оборудоваться системами отопления, кондиционирования воздуха или эффективной приточно-вытяжной вентиляцией.

Для внутренней отделки интерьера помещений с компьютерами должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка — 0,7-0,8; для стен — 0,5-0,6; для пола — 0,3-0,5.

Поверхность пола в помещениях эксплуатации компьютеров должна быть ровной, без выбоин, нескользкой, удобной для очистки и влажной уборки, обладать антистатическими свойствами.

В помещении должны находиться аптечка первой медицинской помощи, углекислотный огнетушитель для тушения пожара.

Работая с ПК, оператор подвергается воздействию следующих психофизических факторов:

- умственное перенапряжение;
- перенапряжение зрительных и слуховых анализаторов;
- монотонность труда;
- эмоциональные перегрузки.

Кроме того, работа с ПК связана с воздействием таких вредных и опасных факторов, как:

- повышенный уровень шума;
- повышенная температура внешней среды;
- недостаточная освещенность рабочей зоны;
- электрический ток;
- статическое электричество;
- электромагнитное излучение и др.

Влияние выше приведенных факторов приводит к утомлению и снижению работоспособности. Появление и развитие утомления вызывает изменения в центральной нервной системе человека. В результате длительного нахождения человека в зоне комбинированного воздействия вредных различных и опасных факторов может привести к профессиональному заболеванию.

7.1.3. Требования к обслуживающему персоналу

Технический персонал будет состоять из шести сотрудников:

- системного администратора, который будет отвечать за работу сервера системы, а именно за корректную обработку запросов и контролировать нагрузку на сервер.
- администратора базы данных, который отвечает за хранение и организацию данных системы.
- архитектора программного обеспечения, который отвечает за программную часть разрабатываемого приложения и сервера.
- UX/UI-дизайнера, который отвечает за интерфейс приложения, его удобство и эстетичность.
- аналитика программного обеспечения, который проверяет эффективность и правильность работы программы.
- специалиста-тестировщика, который проводит все виды тестирования приложения, выявляет ошибки и недочеты, которые нужно будет исправить для оптимизации работы приложения.

Образование у сотрудников должно быть среднее профессиональное или выше.

Особые требования к персоналу:

- умение работать с ПК;
- умение работать в СУБД Microsoft SQL Management studio;
- необходим опыт работы с базами данных;
- умение работать с профессиональными программами, например, Adobe FrameMaker, MS Word;

- знание языков программирования, в частности, C#;
- владение профессиональными программами, например, Adobe Photoshop, Adobe Illustrator, GIMP, Figma;
- знание IDE и средств коллективной разработки (Git и/или других);
- обладание навыком работы с документацией;
- умение анализировать данные;
- необходим опыт работы в команде;
- обладание навыком эффективного распределения времени.

7.2. Мероприятия по технике безопасности

7.2.1. Организационно-технические мероприятия по охране труда

При работе за ПК требуется соблюдать правила:

- правильно сидеть для избегания переутомления и проблем со спиной;
- держать глаза на расстоянии 60-100 см. от монитора;
- не хранить воду рядом с электрическими приборами;
- не допускать большое скопление пыли на приборах и в помещении.

При работе с ПК защитные средства не требуются, кроме случаев, если у сотрудника плохое зрение и ему нужно носить очки.

В помещении или рядом с помещением должна быть схема здания, если понадобится пожарная эвакуация.

7.2.2. Требования электробезопасности при работе с ПК

Необходимо постоянно следить на рабочем месте за исправным состоянием электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть. При обнаружении неисправности немедленно обесточить оборудование, оповестить администрацию. Продолжение работы возможно только после устранения неисправности.

Во избежание повреждения изоляции проводов и возникновения коротких замыканий не разрешается:

- вешать что-либо на провода;
- закрашивать и белить шнуры и провода;
- закладывать провода и шнуры за газовые и водопроводные трубы, за батареи отопительной системы;
- выдергивать штепсельную вилку из розетки за шнур, усилие должно быть приложено к корпусу вилки.

Для исключения поражения электрическим током запрещается:

- часто включать и выключать компьютер без необходимости;
- прикасаться к экрану и к тыльной стороне блоков компьютера;
- работать на средствах вычислительной техники и периферийном оборудовании мокрыми руками;
- работать на средствах вычислительной техники и периферийном оборудовании, имеющих нарушения целостности корпуса, нарушения изоляции проводов, неисправную индикацию включения питания, с признаками электрического напряжения на корпусе
- класть на средства вычислительной техники и периферийном оборудовании посторонние предметы.

Запрещается:

- под напряжением очищать от пыли и загрязнения электрооборудование;
- проверять работоспособность электрооборудования в непригодных для эксплуатации помещениях с токопроводящими полами, сырых, не позволяющих заземлить доступные металлические части.

Ремонт электроаппаратуры производится только специалистами-техниками с соблюдением необходимых технических требований.

Во избежание поражения электрическим током, при пользовании электроприборами нельзя касаться одновременно каких-либо трубопроводов, батарей отопления, металлических конструкций, соединенных с землей.

При пользовании электроэнергией в сырых помещениях соблюдать особую осторожность. При обнаружении оборванного провода необходимо немедленно сообщить об этом администрации, принять меры по исключению контакта с ним людей.

Во всех случаях поражения человека электрическим током немедленно вызывают врача. До прибытия врача нужно, не теряя времени, приступить к оказанию первой помощи пострадавшему. Искусственное дыхание пораженному электрическим током производится вплоть до прибытия врача.

7.2.3. Требования защиты от излучения

Источниками электромагнитного излучения, при работе пользователя за ПК, являются компоненты внутри корпуса и дисплея. Электромагнитное излучение нормируется согласно ГОСТ 12.1.006-84 «ССБТ. Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля». Согласно СанПиН 2.2.2./2.5.1340-03 «Гигиенические требования к видеодисплейным терминалам, персональным ЭВМ и организации работы» конструкция ПЭВМ должна обеспечивать мощность экспозиционной дозы рентгеновского излучения в любой точке на расстоянии 0,05 м от экрана при любых положениях регулировочных устройств не должна превышать 1 мкЗв/час (100 мкР/час).

Допустимые значения неионизирующих электромагнитных излучений:

- напряженность электромагнитного поля по электрической составляющей на расстоянии 50 см от поверхности видеомонитора не более 10 В/м или 0,3 А/м;
- напряженность электростатического поля 20 кВ/м.

Согласно СанПиН 2.2.4/2.1.8.055-96 «Электромагнитные излучения радиочастотного диапазона», для рабочего места оператора ЭВМ или программиста при восьмичасовой рабочей неделе существуют следующие ограничения на уровень электромагнитного излучения: – напряженность магнитного поля не должна превышать 500 В/м; – напряженность электрического поля не должна превышать 50 А/м.

7.2.4. Требования защиты от шума

Уровни шума на рабочих местах пользователей персональных компьютеров не должны превышать значений, установленных СанПиН 2.2.4/2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки» и составляют не более 50 дБА.

На рабочих местах в помещениях для размещения шумных агрегатов уровень шума не должен превышать 75 дБА.

Снизить уровень шума в помещениях можно использованием звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот 63-8000 Гц для отделки стен и потолка помещений. Дополнительный звукопоглощающий эффект создают

однотонные занавески из плотной ткани, повешенные в складку на расстоянии 15-20 см от ограждения. Ширина занавески должна быть в 2 раза больше ширины окна.

7.3. Противопожарная безопасность

Причинами пожара могут быть:

- возгорание компьютера;
- несоблюдение режимов эксплуатации оборудования;
- неправильные действия персонала.

Помещения, в которых располагаются ЭВМ, по пожарной опасности относятся к категории пожароопасности «В», также нужно учитывать высокую стоимость электронного оборудования. Поэтому для изготовления строительных конструкций используется кирпич, железобетон, стекло, металлы и другие негорючие материалы. Применение дерева ограничено, но в случае его использования оно пропитывается огнезащитным веществом (антипиреном).

Воздух в помещении должен очищаться от пыли, так как она может воспламениться при попадании в воздух искр. В соответствии с правилами пожарной безопасности на каждые 100 м² должен находиться углекислотные огнетушители ОУ-5, с помощью которых можно тушить возгорание различных материалов и установок напряжением до 1000В.

При возникновении пожара, в здании должна быть предусмотрена эвакуация людей, при которой люди смогут в кратчайшие сроки покинуть горящее здание. Помещение должно быть оборудовано противопожарной сигнализацией.

В помещениях применяются дымовые пожарные сигнализации. В местах, практически недоступных для тушения пожаров первичными средствами или передвижной техникой пожарной охраны, применяют установки автоматического пожаротушения. Для хранения носителей информации используются негорючие металлические шкафы, двери в хранилище также должны быть негорючими. Для противодействия пожарам применяются специальные мероприятия, направленные на предотвращение и эффективную борьбу с пожарами, такие как установка противопожарных сигнализаций, установка схем для эвакуации и установка огнетушителей.

Для устранения локальных возгораний на рабочих местах в легкодоступном месте находятся ручные огнетушители марки ОУ-25, а также применяется специализированная система автоматического пожаротушения на основе газообразного фреона.

Для помещения, в котором есть вычислительная техника, устанавливается категория пожарной безопасности «Д» по НПБ 105 05.

ЭВМ нельзя устанавливать вблизи источников тепла или термоизлучателей. Персоналу нельзя хранить воду рядом с техникой или проводами.

Чтобы предотвратить пожары в помещении, необходимо максимально сократить количество предметов из горючих материалов или разместить их на безопасном расстоянии.

В качестве первичного средства тушения пожара используется ОУ-2 являющийся углекислотным огнетушителем. Ручные огнетушители устанавливаются из расчета один огнетушитель на 40-50 м², но не меньше двух в помещении. Также в здании должны быть инструкции по пожарной безопасности и планы эвакуации, которые будут показывать кратчайшие выходы из здания в случае пожара.

7.3.1. Противопожарная безопасность при работе с персональным компьютером

На рабочем месте запрещается иметь огнеопасные вещества. В помещениях запрещается:

- зажигать огонь;
- включать электрооборудование, если в помещении пахнет газом;

- курить;
- сушить что-либо на отопительных приборах;
- закрывать вентиляционные отверстия в электроаппаратуре.

Источниками воспламенения являются:

- искра при разряде статического электричества;
- искры от электрооборудования;
- искры от удара и трения;
- открытое пламя.

При возникновении пожароопасной ситуации или пожара персонал должен немедленно принять необходимые меры для его ликвидации, одновременно оповестить о пожаре администрацию.

Помещения с электрооборудованием должны быть оснащены огнетушителями типа ОУ-2 или ОУБ-3.

7.4. Общие требования охраны труда

- К работе программистом допускаются лица не моложе 18 лет, имеющие соответствующую выполняемой работе квалификацию, прошедшие вводный и первичный на рабочем месте инструктажи по охране труда, обученные безопасности труда при работе с персональным компьютером.
- Женщины со времени установления беременности и в период кормления ребенка грудью к выполнению всех видов работ, связанных с использованием видео дисплейных терминалов и персональных компьютеров, не допускаются.
- Для выполнения работ на персональном компьютере программист должен изучить Инструкцию по эксплуатации персонального компьютера, на котором работник выполняет работы, пройти инструктаж по электробезопасности и получить I группу.
- Программист, выполняющий работу на персональном компьютере, независимо от квалификации и стажа работы, не реже одного раза в шесть месяцев должен проходить повторный инструктаж по безопасности труда; в случае нарушения требований безопасности труда, при перерыве в работе более чем на 60 календарных дней программист должен пройти внеплановый инструктаж.
- Программист, не прошедший инструктажи по охране труда и не имеющий I группы по электробезопасности, к самостоятельной работе не допускается.
- Программист, показавший неудовлетворительные навыки и знания требований безопасности при работе на персональном компьютере, к самостоятельной работе не допускается.
- Программист, допущенный к постоянной работе на персональном компьютере, перед поступлением на работу и в дальнейшем периодически (не реже 1 раза в год) должен проходить медицинские осмотры.
- Программист, допущенный к самостоятельной работе, должен знать: правила эксплуатации и требования безопасности при работе с персональным компьютером, способы рациональной организации рабочего места, санитарно-гигиенические требования к условиям труда, опасные и вредные производственные факторы, которые могут оказывать неблагоприятное воздействие на программиста.
- Программист, направленный для участия в несвойственных его профессии работах, должен пройти целевой инструктаж по безопасному выполнению предстоящих работ.
- Программисту запрещается пользоваться инструментом, приспособлениями и оборудованием, безопасному обращению с которыми он не обучен.
- Во время работы на программиста могут оказывать неблагоприятное воздействие в основном следующие опасные и вредные производственные факторы:

- перенапряжение зрительного анализатора при работе за экраном дисплея;
 - длительное статическое напряжение мышц спины, шеи, рук и ног, что может привести к статическим перегрузкам программиста;
 - повышенный уровень шума;
 - ионизирующие и неионизирующие излучения, источниками которых являются видео дисплейные терминалы;
 - статическое электричество;
 - электрический ток, путь которого в случае замыкания на корпус может пройти через тело человека.
- Программист, работающий на персональном компьютере, должен соблюдать установленные для него режимы труда и отдыха.
 - Для предупреждения возможности возникновения пожара программист должен соблюдать требования пожарной безопасности сам и не допускать нарушений со стороны других работников.
 - Для предупреждения заболеваний программисту следует знать и соблюдать правила личной гигиены.
 - В случае заболевания, плохого самочувствия, недостаточного отдыха программисту следует сообщить о своем состоянии непосредственному руководителю и обратиться за медицинской помощью.
 - Если программист оказался очевидцем несчастного случая, он должен оказать пострадавшему первую помощь и сообщить о случившемся руководителю.
 - Программист должен уметь оказать первую помощь, в том числе при поражении электрическим током, пользоваться медицинской аптечкой.
 - Программист, допустивший нарушение или невыполнение требований инструкции по охране труда, рассматривается как нарушитель производственной дисциплины и может быть привлечен к дисциплинарной ответственности, а в зависимости от последствий — и к уголовной; если нарушение связано с причинением предприятию материального ущерба, то виновный может привлекаться к материальной ответственности в установленном порядке.

Вывод седьмой главы:

В данной части выпускной квалификационной работы определены технические данные проектного устройства, условия эксплуатации и требования к обслуживающему персоналу. Были рассмотрены мероприятия по технике безопасности и выделены требования охраны труда.

Так же были рассмотрены мероприятия противопожарной защиты, были определены категория и степень пожарной безопасности объекта и приняты решения для уменьшения вероятности пожара и его предотвращения.

8. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА

8.1. Расчёт затрат на разработку ПО

8.1.1. Затраты на оплату (ЗОТ)

Затраты на оплату (ЗОТ) труда разработчика ПО включают затраты на оплату труда и отчисления от фонда заработной платы.

Затраты на оплату труда разработчика ПО складывается из двух составляющих: основной заработной платы и дополнительной заработной платы. Расчет текущих тарифных ставок приведен в табл. 8.1.1.1.

Таблица 8.1.1.1

Расчёт текущих тарифных ставок оплаты труда персонала

Категории работников-исполнителей работ	Тарифная ставка руб./чел.-ч	Количество рабочих часов	Всего
Архитектор ПО	750	524	393000
UX/UI-дизайнер	1125	60	67500
Аналитик ПО	937	160	149920
Специалист-тестировщик	668	584	390112
Администратор БД	262	160	41920
IT-администратор	325	160	52000
Всего	4067	1648	1094452

Затраты на оплату (ЗОТ) труда разработчика ПО.

Дополнительная заработная плата составляет 20% от основной ЗП, рассчитывается по формуле (1):

$$\begin{aligned} Z_{\text{доп}} &= (20/100)Z_{\text{осн}}, \\ Z_{\text{доп}} &= 0,2 * 1094452 = 218890,4 \text{ руб.} \end{aligned} \quad (1)$$

Затраты на оплату труда разработчика ПО в рублях (ФОТ – фонд оплаты труда) – это сумма основной и дополнительной ЗП, рассчитываемой по формуле (2):

$$\begin{aligned} \text{ФОТ} &= Z_{\text{осн}} + Z_{\text{доп}}, \\ \text{ФОТ} &= 1094452 + 218890,4 = 1313342,4 \text{ руб.} \end{aligned} \quad (2)$$

Отчисления от фонда отплаты труда составляют 30% от фонда оплаты труда и рассчитывается по формуле (3):

$$\begin{aligned} \text{Офот} &= (30/100) * \text{ФОТ}, \\ \text{Офот} &= 0,3 * 1313342,4 = 394002,72 \text{ руб.} \end{aligned} \quad (3)$$

Затраты на оплату труда разработчика ПО рассчитывается по формуле (4):

$$\begin{aligned} 30T &= \Phi OT + O\Phi OT, \\ 30T &= 1313342,4 + 394002,72 = 1707345,12 \text{ руб.} \end{aligned} \quad (4)$$

8.2. Эксплуатационные затраты на оборудование

8.2.1. Суммарная годовая стоимость эксплуатационных затрат

Суммарная стоимость эксплуатационных затрат (Сэз) рассчитывается по формуле (5):

$$\begin{aligned} \text{Сэз} &= \text{СТО} + \text{СМК} + \text{СЭЭ} + \text{Агод}, \\ \text{Сэз} &= 4500 + 3000 + 10272 + 30000 = 47772 \text{ руб.} \end{aligned} \quad (5)$$

где СТО – затраты на техническое обслуживание и ремонт оборудования, руб.,

СМК – затраты на материалы и комплектующие, руб.,

СЭЭ – годовая стоимость электроэнергии, руб.,

Агод – годовые амортизационные отчисления, руб.

8.2.2. Затраты на техническое обслуживание и ремонт

Затраты на техническое обслуживание и ремонт (СТО) составляет 3% от стоимости оборудования (Собор) и рассчитывается по формуле (6):

$$\begin{aligned} \text{СТО} &= 0,03 * \text{Собор}, \\ \text{СТО} &= 0,03 * 150000 = 4500 \text{ руб.} \end{aligned} \quad (6)$$

8.2.3. Затраты на материалы и комплектующие

Затраты на материалы и комплектующие (СМК) принимаются в размере 2% от стоимости оборудования и рассчитывается по формуле (7):

$$\begin{aligned} \text{СМК} &= 0,02 * \text{Собор}, \\ \text{СМК} &= 0,02 * 150000 = 3000 \text{ руб.} \end{aligned} \quad (7)$$

8.2.4. Амортизационные отчисления

Амортизационные отчисления (Агод) производятся по установленным нормам амортизации, выражаются в процентах к стоимости оборудования и рассчитываются по формуле (8):

$$\begin{aligned} \text{Агод} &= \text{Собор} * (\text{НА} / 100), \\ \text{Агод} &= 150000 * (20/100) = 30000 \text{ руб.} \end{aligned} \quad (8)$$

где Собор – стоимость компьютеров;

НА – норма амортизации, рассчитываемая по формуле (9):

$$\text{НА} = 100 / \text{Тнорм}, \quad (9)$$

$$HA = 100/5 = 20 \text{ лет},$$

где $T_{\text{норм}}$ – нормативный срок службы (для ПК $T_{\text{норм}} = 5$ лет).

8.2.5. Стоимость электроэнергии

Стоимость электроэнергии (СЭЭ) вычисляется по формуле (10):

$$\begin{aligned} \text{СЭЭ} &= M * k_z * F_{\text{эф}} * \text{Сквт.ч} * \text{КС}, \\ \text{СЭЭ1} &= 0,5 * 0,8 * 1975,6048 * 6,19 * 1,05 \approx 5136 \text{ руб.}, \\ \text{СЭЭ2} &= 0,5 * 0,8 * 1975,6048 * 6,19 * 1,05 \approx 5136 \text{ руб.}, \\ \text{СЭЭ} &= \text{СЭЭ1} + \text{СЭЭ2}; \\ \text{СЭЭ} &= 5136 + 5136 = 10272, \end{aligned} \quad (10)$$

где M – мощность компьютера, кВт;

k_z – коэффициент загрузки, учитывающий использование оборудования по времени (0,8),

$F_{\text{эф}}$ – эффективный фонд рабочего времени;

Сквт.ч – стоимость 1 кВт/ч электроэнергии (6,19 руб./квт по состоянию на 2024 год);

КС – коэффициент, учитывающий потери в сети (КС = 1,05).

8.2.6. Эффективный фонд рабочего времени

Эффективный фонд рабочего времени ($F_{\text{эф}}$) рассчитывается по формуле (11):

$$\begin{aligned} F_{\text{эф}} &= D_{\text{ном}} * d * (1 - (f/100)), \\ F_{\text{эф}} &= 247 * 8 * (1 - (0,02/100)) = 1975,6048 \approx 1976 \text{ ч}, \end{aligned} \quad (11)$$

где $D_{\text{ном}}$ – номинальное число рабочих дней в году при пятидневной рабочей неделе (247 дн.);

d – продолжительность рабочего дня (8 ч.);

f – планируемый процент времени на ремонт оборудования.

8.2.7. Цена эксплуатационных затрат

Тем не менее, рассчитанная цена эксплуатационных затрат (Эз) – это значения годовых расходов. Следует подкорректировать в соответствии с временным коэффициентом, который формируется исходя из суммарных годовых эксплуатационных затрат, которые рассчитываются по формуле (12):

$$\begin{aligned} \text{Эз} &= \sum t * (\text{Сэз}/F_{\text{эф}}), \\ \text{Эз} &= 1648 * (47772/1976) = 39842 \text{ руб.}, \end{aligned} \quad (12)$$

где $\sum t$ – общее время использования оборудования для решения задачи (из таблицы 1);

Сэз – суммарная годовая стоимость эксплуатационных затрат, вычисленный по формуле (5);

$F_{\text{эф}}$ – эффективный фонд рабочего времени, вычисленный по формуле (11).

8.3. Затраты на материалы

8.3.1. Накладные расходы

Накладные расходы, связанные с управлением, организационными расходами и прочими дополнительными затратами, составляют 40% от фонда оплаты труда (ФОТ), вычисленный по формуле (2), вычисляются по формуле (13):

$$\text{Снакл} = 0,4 * \text{ФОТ}, \quad (13)$$

$$\text{Снакл} = 0,4 * 1313342,4 = 525336,96 \text{ руб.}$$

8.4. Себестоимость разработки программного обеспечения

Таблица 8.4.1

Расчёт затрат на разработку ПО

№ п/п	Наименование статей затрат	Обозначение	Сумма, руб.
1	Эксплуатационные затраты	Эз	47772
2	Фонд оплаты труда	ФОТ	1313342,4
3	Отчисления от фонда оплаты труда	Офот	394002,72
4	Накладные расходы	Снакл	525336,96
5	Себестоимость ПО без учета расходов на сопровождение и адаптацию	п.1+п.2+п.3+п.4	2280454,08
6	Расходы на сопровождение и адаптацию	0,1*п.5	228045,408
7	Полная (плановая) себестоимость ПО	Сполн=п.5+п.6	2508499,488

8.5. Расчет экономического эффекта разработчика и пользователя (заказчика) ПО

8.5.1. Экономический эффект у разработчика программного обеспечения

Прогнозируемая отпускная цена ПО (ЦПО) с учетом НДС рассчитывается по формуле (14):

$$\text{ЦПО} = (\text{Сполн} + \text{П}) * (100 + \text{СТНДС})/100, \quad (14)$$

$$\text{ЦПО} = (2508499,488 + 501699,9) * (100 + 20)/100 = 3612239 \text{ руб.},$$

где Сполн – полная (плановая) себестоимость ПО, руб., (из таблицы 2);

П – прибыль разработчика ПО, руб.;

СТНДС – ставка налога на добавленную стоимость (20%) в %.

Прибыль (П) закладывается в цену исходя из уровня рентабельности (устанавливается студентом самостоятельно), расчет производится по формуле (15):

$$\text{П} = \text{R} * \text{Сполн}/100, \quad (15)$$

$$\text{П} = 20 * 2508499,488/100 = 501699,9 \text{ руб.},$$

где R – уровень рентабельности, % (в рамках ВКР рекомендуемый уровень рентабельности ≈ 20%);

Сполн – полная (плановая) себестоимость ПО, руб. (из таблицы 2).

Чистая прибыль (ЧП) рассчитывается по формуле (16):

$$\begin{aligned} \text{ЧП} &= \text{П}(100 - \text{СТП})/100, \\ \text{ЧП} &= 501699,9 * (100 - 20)/100 = 401359,92 \text{ руб.}, \end{aligned} \quad (16)$$

где П – прибыль, рассчитанная по формуле (15);
СТП – ставка налогообложения прибыли составляет 20%.

Срок окупаемости капитальных вложений (Ток) определяется по формуле (17):

$$\begin{aligned} \text{Ток} &= \text{Сполн}/\text{П}, \\ \text{Ток} &= 2508499,488/501699,9 = 5 \text{ лет}, \end{aligned} \quad (17)$$

где Сполн – полная (плановая) себестоимость продукта;
П – прибыль.

Вывод восьмой главы:

Таким образом, проведен расчёт трудоемкости создания ПО, затрат на создание ПО, экономического эффекта разработчика и пользователя (заказчика) ПО.

Разработчик ПО может продать заказчику программное обеспечение по рассчитанной цене (3612239 руб.), что покроет затраты на разработку ПО (2508499,49 руб.) и обеспечит получение чистой прибыли при его реализации заказчику (401359,92 руб.)

ЗАКЛЮЧЕНИЕ

В заключении выпускной квалификационной работы по теме “Разработка корпоративной системы коммуникации” были представлены результаты исследования о возможности применения разработанного программного продукта в реальных бизнес-процессах и поэтапная реализация корпоративного мессенджера.

В ходе работы был проведен анализ существующих корпоративных мессенджеров и их возможностей, выявлены основные требования к системе и определены основные функциональные возможности будущего мессенджера. На основе полученных данных была разработана архитектура продукта, а также создана его базовая версия с использованием современных технологий и языков программирования. Основные выводы о проделанной работе:

- В первой главе были произведены обзор аналогов, сформирован SWOT и PEST анализы, сформулированы основные требования продуктов.
- Во второй главе была рассмотрена архитектура программного продукта, были смоделированы диаграммы деятельности и последовательности.
- В третьей главе были рассмотрены и описаны основные функции программного продукта.
- В четвертой главе было проведено подробное тестирование системы.
- В пятой главе было отведено внимание информационной безопасности, данным, которые нуждаются в защите.
- В шестой главе была представлена техническая документация корпоративного мессенджера, а именно руководство пользователя, руководство администратора и определение качества продукта по метрикам.
- В седьмой главе приведены мероприятия по технике безопасности и охране труда.
- В восьмой главе произведено экономическое обоснование проекта.

Разработанный корпоративный мессенджер позволяет существенно повысить эффективность коммуникации между сотрудниками компании, упростить процесс обмена информацией и снизить затраты на связь. В процессе тестирования были выявлены преимущества продукта перед существующими аналогами, среди которых: удобный и понятный интерфейс, защита данных, бесплатный доступ и наличие общего хранилища.

На основе полученных результатов можно сделать вывод, что разработанный корпоративный мессенджер отвечает требованиям современных компаний и может быть успешно использован в различных отраслях бизнеса. Однако, для достижения максимальной эффективности необходимо провести дополнительные исследования и усовершенствования, связанные с улучшением интеграции с другими корпоративными системами, увеличении скорости передачи данных, расширением функциональных возможностей и оптимизацией производительности системы.

Таким образом, в ходе выполнения дипломной работы были успешно решены все поставленные задачи и указаны возможности для дальнейшего развития разработанного корпоративного мессенджера, который может быть использован для оптимизации бизнес-процессов и повышения эффективности работы компаний. Нельзя не отметить, что в процессе написания работы были улучшены навыки взаимодействия в команде, распределения обязанностей и работы с программным продуктом. Полученные знания, умения, навыки помогут в избежание ошибок в дальнейшей профессиональной карьере

СПИСОК ИСТОЧНИКОВ ИНФОРМАЦИИ

1. С. С. Качаева. Внутренние коммуникации в период пандемии. 2021. – 7 с.
2. Л.Г. Гагарина. Разработка и эксплуатация автоматизированных информационных систем: Учебное пособие. 2021. -384 с.;
3. В.В. Лисяк. Разработка информационных систем: Учебное пособие. 2019. – 96 с.;
4. SignalR Chat: Сервер и его клиенты. // youtube.com: Sergei Calabonga. Канал для программистов и разработчиков. 2021. URL: <https://youtu.be/TIG2oMgkbcg?si=qzmFIQYKHUB4vTTp> (дата обращения: 24.11.2023)
5. Е. Кучерявый. Пишем мессенджер на C#. Часть 1. Верстка. // skillbox.ru. Онлайн платформа обучения. 2020. URL: https://skillbox.ru/media/code/pishem_messendzher_na_c_chast_1_vyerstka/ (дата обращения: 24.11.2023)
6. Сборник курсов по языку программирования C#. // codecademy.com. Онлайн школа для программистов. 2023. URL: <https://www.codecademy.com/catalog/language/c-sharp> (дата обращения: 24.11.2023)
7. Как создать свой мессенджер. // mdm.ooo. Портал для мобильной разработки. 2022. URL: <https://mdm.ooo/ru/how-to-create-a-messaging-app/amp/> (дата обращения: 24.11.2023)
8. Разработка интерфейса мессенджера на C#. youtube.com: JD's Code Lab. Обучение UI Design. 2020. URL: Chat App UI Design in Wpf C# - YouTube
9. Создание клиент-серверного приложения C# // youtube.com: Simple Code. Подробные уроки программирования. 2018. URL: https://www.youtube.com/watch?v=QohqDyTjclw&t=1645s&ab_channel=%23SimpleCode (дата обращения: 24.11.2023)
10. Miao Yu, Huan Yang, The impact of effective communication on corporate performance, 2023. – 3 с. (дата обращения: 24.11.2023)
11. Субботина Т. Н. Анализ корпоративных мессенджеров разработчиков // Экономика и бизнес: теория и практика. - Калуга: 2023. - С. 107-111. (дата обращения: 24.11.2023)
12. Российский рынок корпоративных мессенджеров в 2023 году // МТС ЛИНК URL: Рынок корпоративных мессенджеров в России в 2023 году (mts-link.ru) (дата обращения: 26.01.2024).
13. Субботина Т.Н. Трансформация банковской деятельности в условиях пандемии COVID-19 / Т.Н. Субботина, К.В. Исакичева // Экономика и бизнес: теория и практика. 2021. - № 1-2 (71). - С. 144-147. (дата обращения: 15.12.2023)
14. Митина Е.В. Трансформация бизнес-коммуникаций в организации под влиянием цифровизации / Е. В. Митина, Т. Н. Субботина // Экономика и бизнес: теория и практика. - 2022. - № 9 (91). - С. 156-161(дата обращения: 15.12.2023)
15. Сколько зарабатывает UX/UI-дизайнер на пике своей формы // UxJournal URL: <https://ux-journal.ru/skolko-zarabatyvaet-ux-ui-dizajner-na-pike-svoej-formy.html> (дата обращения: 26.01.2024).
16. Сколько зарабатывает архитектор ПО в России и за границей в 2024 году // Учись Онлайн.ру URL: <https://uchis-online.ru/blog/professii/skolko-zarabativaet-arhitektor-po> (дата обращения: 30.01.2024).
17. Кто такой системный аналитик? Профессия, требования, зарплата // Учись Онлайн.ру URL: https://systems.education/systems_analyst (дата обращения: 30.01.2024).
18. Сколько зарабатывают тестировщики и QA-инженеры в России: исследование Skillbox Media // Skillbox URL: <https://skillbox.ru/media/code/skolko-zarabatyvayut-testirovshchiki-i-qainzhenery-v-rossii-issledovanie-skillbox-media/> (дата обращения: 30.01.2024).
19. Разработчики и администраторы баз данных // Mojazarplata.ru URL: 1. <https://mojazarplata.ru/karqera/rabota-i-zarplata-v-rossii/rossija-razrabotchiki-i->

- administratory-baz-dannyh (дата обращения: 30.01.2024).
20. Как создать свой мессенджер: все, что нужно знать // Лайв Тайпинг URL: <https://livetyping.com/ru/kak-sozdat-prilozhenie-messenger> (дата обращения: 15.02.2024).
 21. Обеспечение безопасности исходного кода // Solar URL: https://rt-solar.ru/products/solar_dozor/blog/3423/ (дата обращения: 18.02.2024).
 22. Модель угроз безопасности персональных данных: что такое и как составлять // Академия Selectel URL: <https://selectel.ru/blog/personal-data-security-threat-model/> (дата обращения: 20.02.2024).
 23. Защита персональных данных и не только - книга рецептов // Blogger URL: <https://shydrova.blogspot.com/2022/07/21.html> (дата обращения: 03.03.2024).
 24. Фленов М. Е. Библия С#. - 4-е изд. - СПб: БХВ-Петербург, 2019. - 512 с. (дата обращения: 20.03.2024).
 25. Тюкачев Н. А. С#. Алгоритмы и структуры данных: учебное пособие для СПО. - СПб: Лань, 2021. - 232 с. (дата обращения: 28.03.2024).
 26. Фаулер Мартин Шаблоны корпоративных приложений. - СПб: Вильямс, 2016. - 544 с. (дата обращения: 01.03.2024).

ЛИСТИНГ ПРОГРАММНЫХ МОДУЛЕЙ

```

1. MainWindow.xaml.cs
using login_registration;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.ListView;
using System.ServiceModel;
using System.Windows.Interop;
using Microsoft.Win32;
using System.IO;
using System.Xml.Linq;
using Microsoft.Office.Interop.Word;
using FirebaseStorageExample;
using app_interface.ServiceReference1;
using Firebase.Storage;
using Google.Cloud.Firestore;
using System.Net.Sockets;
using System.Windows.Forms;
using System.Diagnostics;
using Firebase.Auth;
using iTextSharp.text;
using iTextSharp.text.pdf;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.Header;
using LiteDB;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
using LiveCharts;
using LiveCharts.Defaults;
using LiveCharts.Wpf;

namespace app_interface
{

    public partial class MainWindow : System.Windows.Window , IServiceChatCallback
    {
        public LiveCharts.SeriesCollection MessagesSeries { get; set; }

        ServiceReference1.ServiceChatClient client;

        bool check = false;

        string connectionString = "Data Source=DESKTOP-K259ROS\\SQLEXPRESS;Initial Catalog=Database;Integrated Security=True";

        public void MsgCallback(string msg)
        {
            messageTest.Items.Add(msg);
        }

        DataBase database = new DataBase();
        public MainWindow()
        {
            InitializeComponent();

            client = new ServiceChatClient(new InstanceContext(this));
            fill_table();
            fill_doc_table();
            fill_role();
            fill_users();
        }

        void fill_tbs()
        {

```

```

string login = emailLogBox.Text;
SqlConnection connection = new SqlConnection(connectionString);
connection.Open();
string sqlQuery = $"select First_Name, Last_Name, Email, Login from Users where Login = '{login}'";
SqlCommand command = new SqlCommand(sqlQuery, connection);
SqlDataReader reader = command.ExecuteReader();
if (reader.Read())
{
    string name = reader["First_Name"].ToString();
    string lname = reader["Last_Name"].ToString();
    string log = reader["Email"].ToString();
    string email = reader["Login"].ToString();
    fnInfTb.Text = name;
    lnInfTb.Text = lname;
    logInfTb.Text = log;
    emailInfTb.Text = email;
}

reader.Close();
connection.Close();
}
private List<Message> LoadMessagesForUser(string user)
{
    List<Message> messages = new List<Message>();
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        string query = $"SELECT From_User, To_User, Message_Data, Message_Date FROM Messages WHERE From_User = '{user}' OR To_User = '{user}'";
        SqlCommand command = new SqlCommand(query, connection);
        SqlDataReader reader = command.ExecuteReader();

        while (reader.Read())
        {
            string fromUser = reader["From_User"].ToString();
            string toUser = reader["To_User"].ToString();
            string text = reader["Message_Data"].ToString();
            DateTime date = Convert.ToDateTime(reader["Message_Date"]);

            if (fromUser == user)
            {
                messages.Add(new Message { FromUser = "я", ToUser = toUser, Text = text, Date = date });
            }
            else
            {
                messages.Add(new Message { FromUser = fromUser, ToUser = toUser, Text = text, Date = date });
            }
        }

        reader.Close();
    }

    return messages;
}
void fill_table()
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        string query = "SELECT Users.ID_User, Users.First_Name, Users.Last_Name, Users.Email, Users.Login, Users.Hashed_Password, Ranks.Rank_Name AS User_Rank " +
            "FROM Users " +
            "INNER JOIN Ranks ON Users.User_Rank = Ranks.ID_Rank";
        SqlCommand command = new SqlCommand(query, connection);
        SqlDataReader reader = command.ExecuteReader();
        System.Data.DataTable dataTable = new System.Data.DataTable();
        dataTable.Load(reader);
        UserInfoDataGrid.ItemsSource = dataTable.DefaultView;
    }
}
void fill_doc_table()
{
    string sqlQuery = "SELECT Documents.ID_Doc, Users.Login AS ID_User, Documents.Doc_Name, Documents.Doc_Url " +
        "FROM Documents " +
        "INNER JOIN Users ON Documents.ID_User = Users.ID_User";
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
    }
}

```

```

        SqlDataAdapter dataAdapter = new SqlDataAdapter(sqlQuery, connection);
        DataSet dataSet = new DataSet();
        dataAdapter.Fill(dataSet, "Documents");
        documentsDataGrid.ItemsSource = dataSet.Tables["Documents"].DefaultView;
    }
}

void fill_role()
{
    SqlConnection connection = new SqlConnection(connectionString);
    string query = "select Rank_Name from Ranks";
    SqlDataAdapter dataAdapter = new SqlDataAdapter(query, connection);
    DataSet dataSet = new DataSet();
    dataAdapter.Fill(dataSet, "Ranks");
    System.Data.DataTable dt = dataSet.Tables["Ranks"];
    roleCBox.ItemsSource = dt.DefaultView;
    roleCBox.DisplayMemberPath = "Rank_Name";
}

void fill_users()
{
    SqlConnection connection = new SqlConnection(connectionString);
    string query = "select Login from Users";
    SqlDataAdapter dataAdapter = new SqlDataAdapter(query, connection);
    DataSet dataSet = new DataSet();
    dataAdapter.Fill(dataSet, "Users");
    System.Data.DataTable dt = dataSet.Tables["Users"];
    usersCb.ItemsSource = dt.DefaultView;
    usersCb.DisplayMemberPath = "Login";
}

private string GetSqlResultByLogin(string login)
{
    string result = "";
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        string query = $"SELECT First_Name, Last_Name FROM Users WHERE Login = @login";
        SqlCommand command = new SqlCommand(query, connection);
        command.Parameters.AddWithValue("@login", login);

        SqlDataReader reader = command.ExecuteReader();

        while (reader.Read())
        {
            result += $"{reader["First_Name"]} {reader["Last_Name"]} " + Environment.NewLine;
        }
    }

    return result;
}

private byte[] GetImageByteArrayById(string login)
{
    byte[] imageByteArray = null;

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        string query = $"SELECT User_Photo FROM Users WHERE Login = @login";
        SqlCommand command = new SqlCommand(query, connection);
        command.Parameters.AddWithValue("@login", login);

        SqlDataReader reader = command.ExecuteReader();

        if (reader.Read())
        {
            imageByteArray = (byte[])reader["User_Photo"];
        }
    }

    return imageByteArray;
}

private ImageSource GetImageFromByteArray(byte[] byteArray)
{
    BitmapImage image = new BitmapImage();

    using (MemoryStream memoryStream = new MemoryStream(byteArray))
    {

```

```

        image.BeginInit();
        image.StreamSource = memoryStream;
        image.CacheOption = BitmapCacheOption.OnLoad;
        image.EndInit();
    }

    return image;
}

private void Border_MouseDown(object sender, MouseButtonEventArgs e)
{
    if (e.ChangedButton == MouseButton.Left)
        DragMove();
}

private void btnMinimize_Click_1(object sender, RoutedEventArgs e)
//to minimize the window
{
    WindowState = WindowState.Minimized;
}

private void btnMaximize_Click_1(object sender, RoutedEventArgs e)
//to maximize the window
{
    if (WindowState == WindowState.Normal)
        WindowState = WindowState.Maximized;
    else
        WindowState = WindowState.Normal;
}

private void btnClose_Click_1(object sender, RoutedEventArgs e)
//to close the window
{
    System.Windows.Application.Current.Shutdown();
}

private void Conversation_Loaded(object sender, RoutedEventArgs e)
{
}

private void Button_Back_Reg_Click(object sender, RoutedEventArgs e)
{
    logWinGrid.Visibility = Visibility.Hidden;
    regWinGrid.Visibility = Visibility.Visible;
}

private void Button_Back_Log_Click(object sender, RoutedEventArgs e)
{
    regWinGrid.Visibility = Visibility.Hidden;
    logWinGrid.Visibility = Visibility.Visible;
}

// авторизация
private void Button_LogIn_Click(object sender, RoutedEventArgs e)
{
    string login = emailLogBox.Text.Trim().ToLower();
    string password = passwordLogBox.Password.Trim();
    string hashedPass = MyHash.HashPassword(password);
    check = client.Authorisation(login, hashedPass);
    if (check)
    {
        System.Windows.MessageBox.Show("Вы успешно вошли!");
        logWinGrid.Visibility = Visibility.Hidden;
        mainWinBorder.Visibility = Visibility.Visible;
        mainWinGrid.Visibility = Visibility.Visible;
        string query = $"SELECT Login FROM Users where Login <> '{login}'";
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();

            using (SqlCommand command2 = new SqlCommand(query, connection))
            {
                using (SqlDataReader reader = command2.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        string value = reader["Login"].ToString();
                        usersListBox.Items.Add(value);
                    }
                }
            }
        }
    }
}

```

```

    }
}
else System.Windows.MessageBox.Show("такого аккаунта не существует");
}
// регистрация
private void Button_Registration_Click(object sender, RoutedEventArgs e)
{
    string firstName = firstNameRegBox.Text.Trim();
    string lastName = lastNameRegBox.Text.Trim();
    string email = emailRegBox.Text.Trim().ToLower();
    string login = logRegBox.Text.Trim();
    string password = passwordRegBox.Password.Trim();
    string confirmPassword = passwordRegBox.Password.Trim();
    bool loginExists = CheckLoginExistence(login);

    if (loginExists)
    {
        System.Windows.MessageBox.Show("Логин уже существует!");
    }
    else
    {
        if (firstName.Length < 1)
        {
            firstNameRegBox.ToolTip = "Это поле введено некорректно!";
            firstNameRegBox.Background = Brushes.DarkRed;
        }
        else if (lastName.Length < 1)
        {
            lastNameRegBox.ToolTip = "Это поле введено некорректно!";
            lastNameRegBox.Background = Brushes.DarkRed;
        }
        else if (email.Length < 5 || !email.Contains("@") || !email.Contains("."))
        {
            emailRegBox.ToolTip = "Это поле введено некорректно!";
            emailRegBox.Background = Brushes.DarkRed;
        }
        else if (password.Length < 5)
        {
            System.Windows.MessageBox.Show("Слишком короткий пароль!");
            passwordRegBox.Password = "Это поле введено некорректно!";
            passwordRegBox.Background = Brushes.DarkRed;
        }
        else if (confirmPassword != password)
        {
            passwordRegBox.Password = "Пароли не совпадают!";
            confirmPasswordRegBox.Background = Brushes.DarkRed;
        }
        else
        {
            firstNameRegBox.ToolTip = "";
            firstNameRegBox.Background = Brushes.Transparent;

            lastNameRegBox.ToolTip = "";
            lastNameRegBox.Background = Brushes.Transparent;

            emailRegBox.ToolTip = "";
            emailRegBox.Background = Brushes.Transparent;

            passwordRegBox.ToolTip = "";
            passwordRegBox.Background = Brushes.Transparent;

            confirmPasswordRegBox.ToolTip = "";
            confirmPasswordRegBox.Background = Brushes.Transparent;

            string hashedPass;
            hashedPass = MyHash.HashPassword(password);
            check = client.Registration(firstName, lastName, email, login, hashedPass);
            if (check)
            {
                System.Windows.MessageBox.Show("Регистрация успешна. Пожалуйста, войдите в аккаунт для продолжения работы.");
                logWinGrid.Visibility = Visibility.Visible;
                regWinGrid.Visibility = Visibility.Hidden;
            }
            else System.Windows.MessageBox.Show("Аккаунт не создан");
        }
    }
}
}
public bool CheckLoginExistence(string login)

```

```

{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();

        string query = "SELECT COUNT(*) FROM Users WHERE Login = @Login";

        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@Login", login);
            int count = (int)command.ExecuteScalar();

            return count > 0;
        }
    }
}

private void backToMainWinBtn_Click(object sender, RoutedEventArgs e)
{
    myRprofileInfoGrid.Visibility = Visibility.Hidden;
    mainWinGrid.Visibility = Visibility.Visible;
    mainWinBorder.Visibility = Visibility.Visible;
}

private void usersListBox_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    beautyPanel.Visibility = Visibility.Hidden;
    messageTest.Visibility = Visibility.Visible;
    messageTest.Items.Clear();

    MessageTB.Visibility = Visibility.Visible;
    sendMsgBtn.Visibility = Visibility.Visible;

    string selectedUser = (string)usersListBox.SelectedItem;
    var messages = LoadMessagesForUser(selectedUser);

    string sqlResult = GetSqlResultByLogin(selectedUser);
    userName.Text = sqlResult;

    byte[] imageArray = GetImageByteArrayById(selectedUser);
    userPhoto.Source = GetImageFromByteArray(imageArray);

    messageTest.Items.Clear();
    foreach (var message in messages)
    {
        if (message.FromUser == "me")
        {
            messageTest.Items.Add($"me: {message.Text}");
        }
        else
        {
            messageTest.Items.Add($" {message.FromUser}: {message.Text}");
        }
    }
}

private void sendMsgBtn_Click(object sender, RoutedEventArgs e)
{
    if (usersListBox.SelectedItem != null)
    {
        client.SendMsg(emailLogBox.Text, usersListBox.SelectedItem.ToString(), MessageTB.Text.ToString());

        MessageTB.Text = "";
    }
}

private void myProfileInfo_Click(object sender, RoutedEventArgs e)
{
    fill_tbs();
    myRprofileInfoGrid.Visibility = Visibility.Visible;
    mainWinGrid.Visibility = Visibility.Hidden;
    mainWinBorder.Visibility = Visibility.Hidden;
    string login = emailLogBox.Text;

    try
    {

```

```

using (SqlConnection connection = new SqlConnection(connectionString))
{
    connection.Open();

    string querystring = $"SELECT User_Photo FROM Users where Login = '{login}'";

    using (SqlCommand command = new SqlCommand(querystring, connection))
    {
        command.Parameters.AddWithValue("@Login", login);

        byte[] imageBytes = (byte[])command.ExecuteScalar();

        if (imageBytes != null)
        {
            BitmapImage bitmapImage = new BitmapImage();
            using (MemoryStream memoryStream = new MemoryStream(imageBytes))
            {
                memoryStream.Position = 0;
                bitmapImage.BeginInit();
                bitmapImage.CreateOptions = BitmapCreateOptions.PreservePixelFormat;
                bitmapImage.CacheOption = BitmapCacheOption.OnLoad;
                bitmapImage.UriSource = null;
                bitmapImage.StreamSource = memoryStream;
                bitmapImage.EndInit();
            }

            Dispatcher.Invoke(() => {
                pictureSet.Source = bitmapImage;
            });
        }
        //else
        //{
        //    System.Windows.MessageBox.Show("Изображение не найдено.");
        //}
    }
}
catch (Exception ex)
{
    System.Windows.MessageBox.Show("Произошла ошибка при получении изображения: " + ex.Message);
}

private void UserInfoDataGrid_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    toDocBtn.IsEnabled = true;
}

//открыть диалоговое окно и выбрать картинку
string picAddress = "";
private void newPhoto_Click(object sender, RoutedEventArgs e)
{
    System.Windows.Forms.OpenFileDialog openFile = new System.Windows.Forms.OpenFileDialog();
    if (openFile.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        picAddress = openFile.FileName;
        pictureSet.Source =
            new BitmapImage(new Uri(openFile.FileName, UriKind.Absolute)) { CreateOptions = BitmapCreateOptions.IgnoreImageCache };
    }
}

//сохраняет картинку в бд в виде массива байтов
private void changeProfileBtn_Click(object sender, RoutedEventArgs e)
{
    string login = emailLogBox.Text;
    BitmapImage bitmapImage = (BitmapImage)pictureSet.Source;

    byte[] imageBytes;
    using (MemoryStream memoryStream = new MemoryStream())
    {
        JpegBitmapEncoder encoder = new JpegBitmapEncoder();
        encoder.Frames.Add(BitmapFrame.Create(bitmapImage));
        encoder.Save(memoryStream);
        imageBytes = memoryStream.ToArray();
    }
    string query = "UPDATE Users SET User_Photo = @User_Photo WHERE Login = @Login";

    using (SqlConnection connection = new SqlConnection(connectionString))

```



```

{
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.Parameters.Add("@User_Photo", SqlDbType.VarBinary, -1).Value = imageBytes;
        command.Parameters.Add("@Login", SqlDbType.VarChar).Value = login;

        try
        {
            connection.Open();
            command.ExecuteNonQuery();
            System.Windows.MessageBox.Show("Изменения успешно сохранены");
        }
        catch (Exception ex)
        {
            System.Windows.MessageBox.Show("Произошла ошибка при сохранении данных, попробуйте позже.");
        }
    }
}

private void adminWinInfoBtn_Click(object sender, RoutedEventArgs e)
{
    string login = emailLogBox.Text;
    string query = $"select User_Rank from Users where Login = '{login}'";

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();

        using (SqlCommand command = new SqlCommand(query, connection))
        {
            int idrank = (int)command.ExecuteScalar();
            if (idrank == 1)
            {
                adminInfoGrid.Visibility = Visibility.Visible;
                mainWinBorder.Visibility = Visibility.Hidden;
                mainWinGrid.Visibility = Visibility.Hidden;
            }
            else System.Windows.MessageBox.Show("У вас нет прав доступа");
        }
    }
}

private void toDocBtn_Click(object sender, RoutedEventArgs e)
{
    DataRowView row = (DataRowView)UserInfoDataGrid.SelectedItems[0];
    int userId = (int)row.Row.ItemArray[0];
    string query = $"select ID_User, First_Name, Last_Name, Email, Login, Hashed_Password, User_Rank from Users where ID_User = {userId}";

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        SqlCommand command = new SqlCommand(query, connection);
        SqlDataReader reader = command.ExecuteReader();

        if (reader.Read())
        {
            string message = "";

            for (int i = 0; i < reader.FieldCount; i++)
            {
                message += reader.GetName(i) + ": " + reader.GetValue(i).ToString() + "\n"; // Добавляем название поля и значение
            }

            //MessageBox.Show(message, "Данные пользователя"); // Выводим результат в MessageBox с заголовком

            System.Windows.Forms.SaveFileDialog saveFile = new System.Windows.Forms.SaveFileDialog();
            saveFile.Filter = "Документ Word (*.docx)|*.docx|Документ PDF (*.pdf)|*.pdf";

            if (saveFile.ShowDialog() == System.Windows.Forms.DialogResult.OK && saveFile.FileName != "")
            {
                if (saveFile.FilterIndex == 1)
                {
                    string fileName = saveFile.FileName;

                    // Создаем приложение Word и документ
                    Microsoft.Office.Interop.Word.Application wordApp = new Microsoft.Office.Interop.Word.Application();

```

```

Microsoft.Office.Interop.Word.Document document = wordApp.Documents.Add();

// Записываем сообщение в документ
document.Content.Text = message;

// Сохраняем документ в формате .docx
object filePath = fileName;
document.SaveAs2(filePath, WdSaveFormat.wdFormatXMLDocument);

// Закрываем документ и приложение Word
document.Close();
wordApp.Quit();

System.Windows.MessageBox.Show("Сообщение сохранено в файле .docx", "Успешно");
}

if (saveFile.FilterIndex == 2)
{
    using (var pdfWriter = new iText.Kernel.Pdf.PdfWriter(saveFile.FileName))
    {
        var pdfDoc = new iText.Kernel.Pdf.PdfDocument(pdfWriter);

        var pageSize = new iText.Kernel.Geom.PageSize(iText.Kernel.Geom.PageSize.A4);
        pdfDoc.SetDefaultPageSize(pageSize);

        var doc = new iText.Layout.Document(pdfDoc);
        doc.Add(new iText.Layout.Element.Paragraph(message));

        doc.Close();
        pdfDoc.Close();

        System.Windows.MessageBox.Show("Сообщение сохранено в файле .pdf", "Успешно");
    }
}
}
}
}

private void fileStorageBtn_Click(object sender, RoutedEventArgs e)
{
    docGrid.Visibility = Visibility.Visible;
    mainWinBorder.Visibility = Visibility.Hidden;
    mainWinGrid.Visibility = Visibility.Hidden;
}

private void searchBtn_Click(object sender, RoutedEventArgs e)
{
    searchTB.Visibility = Visibility.Visible;
}

private void searchTB_KeyDown(object sender, System.Windows.Input.KeyEventArgs e)
{
    if (e.Key == Key.Enter)
    {
        string username = searchTB.Text;
        if (string.IsNullOrEmpty(username))
        {
            System.Windows.MessageBox.Show("Введите имя пользователя");
        }
        else if (usersListBox.Items.Contains(username))
        {
            usersListBox.SelectedItem = username;
        }
        else
        {
            // Пользователь не найден, выводим сообщение
            System.Windows.MessageBox.Show("Такого пользователя нет", "Поиск пользователя", MessageBoxButton.OK,
            MessageBoxImage.Information);
        }
    }
}

private void Hyperlink_RequestNavigate(object sender, RequestNavigateEventArgs e)
{
    System.Diagnostics.Process.Start(new ProcessStartInfo(e.Uri.AbsoluteUri));
    e.Handled = true;
}

private void refreshDocBtn_Click(object sender, RoutedEventArgs e)

```

```

{
    fill_doc_table();
}

private async void uploadDocBtn_Click(object sender, RoutedEventArgs e)
{
    try
    {
        string login = emailLogBox.Text;
        System.Windows.Forms.OpenFileDialog openFileDialog = new System.Windows.Forms.OpenFileDialog();

        if (openFileDialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)
        {
            string selectedFilePath = openFileDialog.FileName;

            FirebaseStorageHelper firebaseStorageHelper = new FirebaseStorageHelper();
            await firebaseStorageHelper.UploadFileToFirebaseStorage(selectedFilePath);

            string fileName = System.IO.Path.GetFileName(selectedFilePath);
            string downloadUrl = await firebaseStorageHelper.GetFileDownloadUrl(fileName);
            string query = $"insert into Documents (ID_User, Doc_Name, Doc_Url) values ((select ID_User from Users where Login = '{login}'), '{fileName}', '{downloadUrl}')";
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                connection.Open();
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    command.ExecuteNonQuery();
                }
            }
        }
    }
    catch (Exception ex)
    {
        System.Windows.MessageBox.Show("Произошла ошибка при загрузке файла: " + ex.Message, "Ошибка", MessageBoxButton.OK,
        MessageBoxImage.Error);
    }
}

private void roleCBox_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
}

private void RegAdminButton_Click(object sender, RoutedEventArgs e)
{
    DataRowView selectedDataRowViewRole = roleCBox.SelectedItem as DataRowView;
    string firstName = firsrNameRegAdminBox.Text.Trim();
    string lastName = lastNameRegAdminBox.Text.Trim();
    string email = emailRegAdminBox.Text.Trim().ToLower();
    string login = logRegAdminBox.Text.Trim();
    string password = passwordRegAdminBox.Password.Trim();
    string confirmPassword = passwordRegAdminBox.Password.Trim();
    string role = selectedDataRowViewRole["Rank_Name"].ToString();
    string query = $"select ID_Rank from Ranks where Rank_Name = '{role}'";
    bool loginExists = CheckLoginExistence(login);

    if (loginExists)
    {
        System.Windows.MessageBox.Show("Логин уже существует!");
    }
    else
    {
        if (firstName.Length < 1)
        {
            firsrNameRegAdminBox.ToolTip = "Это поле введено некорректно!";
            firsrNameRegAdminBox.Background = Brushes.DarkRed;
        }
        else if (lastName.Length < 1)
        {
            lastNameRegBox.ToolTip = "Это поле введено некорректно!";
            lastNameRegAdminBox.Background = Brushes.DarkRed;
        }
        else if (email.Length < 5 || !email.Contains("@") || !email.Contains("."))
        {
            emailRegAdminBox.ToolTip = "Это поле введено некорректно!";
            emailRegAdminBox.Background = Brushes.DarkRed;
        }
        else if (password.Length < 5)
        {

```

```

        System.Windows.MessageBox.Show("Слишком короткий пароль!");
        passwordRegAdminBox.Password = "Это поле введено некорректно!";
        passwordRegAdminBox.Background = Brushes.DarkRed;
    }
    else if (confirmPassword != password)
    {
        passwordRegAdminBox.Password = "Пароли не совпадают!";
        confirmPasswordRegAdminBox.Background = Brushes.DarkRed;
    }
    else if (roleCBox.SelectedItem == null)
    {
        System.Windows.MessageBox.Show("Выберите роль пользователя");
    }
    else
    {
        firstNameRegAdminBox.Text = "";
        firstNameRegAdminBox.Background = Brushes.Transparent;
        lastNameRegAdminBox.Text = "";
        lastNameRegAdminBox.Background = Brushes.Transparent;
        emailRegAdminBox.Text = "";
        emailRegAdminBox.Background = Brushes.Transparent;
        passwordRegAdminBox.Password = "";
        passwordRegAdminBox.Background = Brushes.Transparent;
        confirmPasswordRegAdminBox.Password = "";
        confirmPasswordRegAdminBox.Background = Brushes.Transparent;
        string hashedPass;
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                int idrank = (int)command.ExecuteScalar();
                hashedPass = MyHash.HashPassword(password);
                check = client.RegistrationForAdmin(firstName, lastName, email, login, hashedPass, idrank);
                if (check)
                {
                    System.Windows.MessageBox.Show("Регистрация успешна.");
                }
                else System.Windows.MessageBox.Show("Аккаунт не создан");
            }
        }
    }
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    adminInfoGrid.Visibility = Visibility.Visible;
    regAdminWinGrid.Visibility = Visibility.Hidden;
}

private void registNewUserBtn_Click(object sender, RoutedEventArgs e)
{
    adminInfoGrid.Visibility = Visibility.Hidden;
    regAdminWinGrid.Visibility = Visibility.Visible;
}

private void backToMainFAdminWinBtn_Click(object sender, RoutedEventArgs e)
{
    adminInfoGrid.Visibility = Visibility.Hidden;
    mainWinBorder.Visibility = Visibility.Visible;
    mainWinGrid.Visibility = Visibility.Visible;
}

private void refreshUserBtn_Click(object sender, RoutedEventArgs e)
{
    fill_table();
}

private void backToMainFDocWinBtn_Click(object sender, RoutedEventArgs e)
{
    docGrid.Visibility = Visibility.Hidden;
    mainWinBorder.Visibility = Visibility.Visible;
    mainWinGrid.Visibility = Visibility.Visible;
}

private void documentsDataGrid_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    delDocBtn.IsEnabled = true;
}

private async void deleteDocBtn_Click(object sender, RoutedEventArgs e)
{
    string login = emailLogBox.Text;
    DataRowView row = (DataRowView)documentsDataGrid.SelectedItems[0];

```

```

int ID_Doc = (int)row.Row.ItemArray[0];
string query = $"DELETE FROM Documents WHERE ID_Doc = '{ID_Doc}' AND ID_User = (SELECT ID_User FROM Users WHERE Login = '{login}')";

string selectQuery = $"SELECT Doc_Name, ID_User FROM Documents WHERE ID_Doc = '{ID_Doc}'";

using (SqlConnection connection = new SqlConnection(connectionString))
{
    FirebaseStorageHelper firebaseStorageHelper = new FirebaseStorageHelper();
    connection.Open();

    SqlCommand command = new SqlCommand(selectQuery, connection);
    SqlDataReader reader = command.ExecuteReader();

    if (reader.Read())
    {
        string docName = (string)reader["Doc_Name"];
        int creatorID = (int)reader["ID_User"];

        int currentID = GetCurrentUserID(login);

        if (creatorID == currentID)
        {
            MessageBoxResult result = (MessageBoxResult)System.Windows.Forms.MessageBox.Show("Вы уверены, что хотите удалить файл?",
"Подтверждение удаления", (MessageBoxButtons)MessageBoxButton.YesNo, (MessageBoxIcon)MessageBoxImage.Question);

            if (result == MessageBoxResult.Yes)
            {
                SqlCommand deleteCommand = new SqlCommand(query, connection);
                deleteCommand.ExecuteNonQuery();

                await firebaseStorageHelper.DeleteFileFromFirebaseStorage(docName);
            }
            else
            {
                System.Windows.Forms.MessageBox.Show("У вас нет прав для удаления этого файла.", "Ошибка удаления",
(MessageBoxButtons)MessageBoxButton.OK, (MessageBoxIcon)MessageBoxImage.Error);
            }
        }
    }
}

private int GetCurrentUserID(string login)
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        string query = $"SELECT ID_User FROM Users WHERE Login = '{login}'";
        SqlCommand command = new SqlCommand(query, connection);
        int userID = (int)command.ExecuteScalar();
        return userID;
    }
}

private void countBtn_Click(object sender, RoutedEventArgs e)
{
    if (usersCb.SelectedItem == null || startDatePicker.SelectedDate == null || endDatePicker.SelectedDate == null)
    {
        System.Windows.Forms.MessageBox.Show("Пожалуйста, выберите пользователя и укажите даты начала и конца периода для подсчета сообщений.");
        return;
    }

    //запомнить юзера
    DataRowView selectedDataRowViewRole = usersCb.SelectedItem as DataRowView;
    string login = selectedDataRowViewRole["Login"].ToString();
    //запомнить интервал в datepicker
    DateTime startDate = startDatePicker.SelectedDate ?? DateTime.MinValue;
    DateTime endDate = endDatePicker.SelectedDate ?? DateTime.MaxValue;

    //вставить знач в запрос
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();

        // Создание и выполнение SQL-запроса
        string sqlQuery = $"SELECT CAST(Message_Date AS DATE) AS date, COUNT(*) AS message_count FROM [Database].[dbo].Messages";
    }
}

```

```
WHERE From_User = '{login}' AND Message_Date BETWEEN '{startDate}' AND '{endDate}' GROUP BY CAST(Message_Date AS DATE) ORDER BY date;";
```

```
SqlCommand command = new SqlCommand(sqlQuery, connection);
SqlDataReader reader = command.ExecuteReader();

// Загрузка данных из результата запроса
MessagesSeries = new LiveCharts.SeriesCollection();
ChartValues<ObservablePoint> chartValues = new ChartValues<ObservablePoint>();

if (!reader.HasRows)
{
    System.Windows.Forms.MessageBox.Show("У выбранного пользователя нет сообщений в указанном периоде.");
}
else
{
    while (reader.Read())
    {
        DateTime date = reader.GetDateTime(0);
        int messageCount = reader.GetInt32(1);

        chartValues.Add(new ObservablePoint(date.ToOADate(), messageCount));
    }

    MessagesSeries.Add(new LineSeries
    {
        Title = "Message Count",
        Values = chartValues
    });

    DataContext = this;
    messagesChart.AxisX.Add(new LiveCharts.Wpf.Axis
    {
        LabelFormatter = value => new DateTime((long)value).ToString("dd.MM.yyyy") // Форматирование даты в нужный вид
    });
}
}

private void graphBtn_Click(object sender, RoutedEventArgs e)
{
    graphGrid.Visibility = Visibility.Visible;
    adminInfoGrid.Visibility = Visibility.Hidden;
}

private void backFGraphBtn_Click(object sender, RoutedEventArgs e)
{
    graphGrid.Visibility = Visibility.Hidden;
    adminInfoGrid.Visibility = Visibility.Visible;
}
}

2. FirebaseStorage.cs
using Firebase.Storage;
using System;
using System.IO;
using System.Threading.Tasks;
using System.Windows;

namespace FirebaseStorageExample
{
    public class FirebaseStorageHelper
    {
        private const string FirebaseStorageBucket = "degree-eaeef.appspot.com";
        private const string FirebaseAuthToken = "AIzaSyCSY4_IcFMcIsOu7048DDIHyoF8xFQoBPk";

        public async Task UploadFileToFirebaseStorage(string filePath)
        {
            try
            {
                var stream = File.OpenRead(filePath);

                var task = Task.Run(async () =>
                {
                    var storage = new FirebaseStorage(FirebaseStorageBucket, new FirebaseStorageOptions { AuthTokenAsyncFactory = () =>
                        Task.FromResult(FirebaseAuthToken) });
                });
            }
            catch { }
        }

        // Загрузка файла
    }
}
```

```

        await storage
            .Child("files/" + Path.GetFileName(filePath))
            .PutAsync(stream,
                cancellationTokens: default
            );

        MessageBox.Show("Файл успешно загружен в Firebase Storage!");
    });

    await task;
}
catch (Exception ex)
{
    MessageBox.Show($"Произошла ошибка при загрузке файла в Firebase Storage: {ex.Message}");
}
}

public async Task<string> GetFileDownloadUrl(string fileName)
{
    var task = new FirebaseStorage("degree-eaeef.appspot.com")
        .Child("files/" + fileName)
        .GetDownloadUrlAsync();

    string downloadUrl = await task;

    return downloadUrl;
}

public async Task DeleteFileFromFirebaseStorage(string fileName)
{
    try
    {
        var storage = new FirebaseStorage(FirebaseStorageBucket, new FirebaseStorageOptions { AuthTokenAsyncFactory = () =>
Task.FromResult(FirebaseAuthToken) });

        // Удаление файла
        await storage
            .Child("files/" + fileName)
            .DeleteAsync();

        MessageBox.Show("Файл успешно удален из Firebase Storage!");
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Произошла ошибка при удалении файла из Firebase Storage: {ex.Message}");
    }
}
}
}

3. IServiceChat.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
namespace wcf_chat
{
    // ПРИМЕЧАНИЕ. Команду "Переименовать" в меню "Рефакторинг" можно использовать для одновременного изменения имени интерфейса
    "IServiceChat" в коде и файле конфигурации.
    [ServiceContract(CallbackContract = typeof(IServerChatCallback))]
    public interface IServiceChat
    {
        [OperationContract]
        int Connect(string name);
        [OperationContract]
        void Disconnect(int id);
        [OperationContract(IsOneWay = true)]
        void SendMsg(string msg, int id);
    }
    public interface IServerChatCallback
    {
        [OperationContract(IsOneWay = true)]
        void MsgCallback(string msg);
    }
}

4. ServerUser.cs
using System.ServiceModel;

```

```

namespace wcf_chat
{
    public class ServerUser
    {
        public int ID { get; set; }

        public string Name { get; set; }

        public OperationContext operationContext { get; set; }
    }
}

5. ServiceChat.cs
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
using System.IO;
using System.Windows.Input;
using System.Xml.Linq;

namespace wcf_chat
{
    // [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
    [ServiceBehavior(IncludeExceptionDetailInFaults = true)]
    public class ServiceChat : IServiceChat
    {
        public string connectionString = "Data Source=DESKTOP-K259ROS\\SQLEXPRESS;Initial Catalog=Database;Integrated Security=True";
        List<ServerUser> users = new List<ServerUser>();
        public bool Authorisation(string Login, string hashedPass)
        {
            ServerUser user = users.FirstOrDefault(u => u.Login == Login);

            if (user == null)
            {
                if (!IsUserInDatabase(Login, hashedPass))
                {
                    return false;
                }

                // Пользователь не найден в списке, добавляем его
                user = new ServerUser { Login = Login, Password = hashedPass, operationContext = OperationContext.Current };
                users.Add(user);

                return true; // Возвращаем true после добавления пользователя
            }

            return true; // Возвращаем true, так как пользователь был найден в списке
        }
        //проверяет наличие юзера в бд
        private bool IsUserInDatabase(string login, string hashedPass)
        {
            string query = "SELECT COUNT(*) FROM Users WHERE Login = @Login AND Hashed_Password = @HashedPass";

            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    command.Parameters.AddWithValue("@Login", login);
                    command.Parameters.AddWithValue("@HashedPass", hashedPass);

                    int count = Convert.ToInt32(command.ExecuteScalar());

                    return count == 1;
                }
            }
        }

        public bool Registration(string firstName, string lastName, string email, string login, string password)
        {
            ServerUser user = new ServerUser()
            {
                First_Name = firstName,
                Last_Name = lastName,
            }
        }
    }
}

```



```

        Email = email,
        Login = login,
        Password = password,
        operationContext = OperationContext.Current
    };
    //дефолт картинка для всех юзеров(смени путь обязательно)
    string path = "/Assets/user with no photo.png"; //"C:\\Users\\samsung\\Pictures\\user with no photo.png";
    byte[] imageBytes = File.ReadAllBytes(path);

    string query = $"INSERT INTO Users (First_Name, Last_Name, Email, Login, Hashed_Password, User_Rank, User_Photo) " +
        $"VALUES ('{firstName}', '{lastName}', '{email}', '{login}', '{password}', '2', @UserPhoto)";
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@UserPhoto", imageBytes);
            int rowsAffected = command.ExecuteNonQuery();
            if (rowsAffected == 1)
            {
                //добавление в серверный список
                users.Add(user);
                return true;
            }
            else return false;
        }
    }
}

public bool RegistrationForAdmin(string firstName, string lastName, string email, string login, string password, int roleRank)
{
    ServerUser user = new ServerUser()
    {
        First_Name = firstName,
        Last_Name = lastName,
        Email = email,
        Login = login,
        Password = password,
        ID_Rank = roleRank,
        operationContext = OperationContext.Current
    };
    //дефолт картинка для всех юзеров(смени путь обязательно)
    string path = "C:\\Users\\samsung\\Pictures\\user with no photo.png";
    byte[] imageBytes = File.ReadAllBytes(path);

    string query = $"INSERT INTO Users (First_Name, Last_Name, Email, Login, Hashed_Password, User_Rank, User_Photo) " +
        $"VALUES ('{firstName}', '{lastName}', '{email}', '{login}', '{password}', '{roleRank}', @UserPhoto)";
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@UserPhoto", imageBytes);
            int rowsAffected = command.ExecuteNonQuery();
            if (rowsAffected == 1)
            {
                //добавление в серверный список
                users.Add(user);
                return true;
            }
            else return false;
        }
    }
}

public void SendMsg(string UserFrom, string UserTo, string MessageData)
{
    ServerUser sender = users.FirstOrDefault(u => u.Login == UserFrom);
    ServerUser receiver = users.FirstOrDefault(u => u.Login == UserTo);

    if (sender != null && receiver != null)
    {
        if (receiver.operationContext != null)
        {
            receiver.operationContext.GetCallbackChannel<IServerChatCallback>()
                .MsgCallback($"{DateTime.Now.ToShortTimeString()} | {sender.Login}: {MessageData}");
        }

        using (SqlConnection connection = new SqlConnection(connectionString))
    
```

```

{
    connection.Open();

    string query = "INSERT INTO Messages (From_User, To_User, Message_Data, Message_Date) " +
        "VALUES (@From_User, @To_User, @Message_Data, @Message_Date)";

    SqlCommand command = new SqlCommand(query, connection);

    command.Parameters.AddWithValue("@From_User", sender.Login);
    command.Parameters.AddWithValue("@To_User", receiver.Login);
    command.Parameters.AddWithValue("@Message_Data", MessageData);
    command.Parameters.AddWithValue("@Message_Date", DateTime.Now); // Добавляем текущую дату как дату сообщения

    int rowsAffected = command.ExecuteNonQuery();

    if (!(rowsAffected > 0))
    {
        Console.WriteLine("error");
    }
}
else
{
    Console.WriteLine($"Пользователь {UserTo} не был найден. Сообщение не было отправлено.");
}

if (sender.operationContext != null)
{
    sender.operationContext.GetCallbackChannel<IServerChatCallback>()
        .MsgCallback($"{DateTime.Now.ToShortTimeString()} | я: {MessageData}");
}
else
{
    Console.WriteLine("error");
}
}
}
}

```

ПОЛИТИКА БЕЗОПАСНОСТИ

Политика безопасности корпоративного мессенджера PigeonGram

1. Введение

Целью данной политики безопасности корпоративного мессенджера является обеспечение конфиденциальности, целостности и доступности информации, передаваемой через мессенджер внутри организации. Мессенджер является важной коммуникационной платформой, используемой сотрудниками компании для обмена сообщениями и файлами, поэтому необходимо принять соответствующие меры по обеспечению его безопасности. Данная Политика конфиденциальности определяет порядок использования и защиты разработчиками корпоративного мессенджера PigeonGram (далее — «PigeonGram»), персональных данных, предоставленных пользователями или иным образом полученных или сгенерированных в связи с использованием клиентом услуг облачного обмена сообщениями (далее — «Услуги»).

Регистрируясь в PigeonGram, пользователь принимает Политику конфиденциальности приложения и соглашается:

- Не использовать мессенджер для рассылки спама или обмана пользователей;
- Не призывать к насилию в публичных каналах, ботах и других общедоступных чатах в PigeonGram;
- Не размещать незаконный порнографический контент в каналах, ботах и других общедоступных чатах в PigeonGram.
- Разработчики и модераторы оставляют за собой право в дальнейшем вносить изменения и дополнения в данное Пользовательское соглашение.

1.1. Принципы конфиденциальности

В вопросах сбора и обработки персональных данных PigeonGram придерживается двух основополагающих принципов:

1. Персональные данные не используются для показа рекламы.
2. Сохраняются только те данные, которые необходимы для корректной работы PigeonGram в качестве надёжного и многофункционального сервиса для обмена сообщениями.

1.2. Пользовательское соглашение

Данная Политика конфиденциальности является частью Пользовательского соглашения PigeonGram, определяющего условия использования Услуг. Таким образом, данную Политику конфиденциальности следует рассматривать в совокупности с Пользовательским соглашением, представленным в приложении 5.

1.3. Содержание

Данная Политика конфиденциальности определяет:

1. на каких правовых основаниях мы обрабатываем персональные данные;
2. какие из Ваших персональных данных мы можем собирать;
3. как мы обеспечиваем безопасность персональных данных;

4. для каких целей мы можем использовать персональные данные;
5. кому могут быть переданы персональные данные; и
6. какие у Вас есть права в отношении Ваших персональных данных.

2. Правовые основания для обработки персональных данных

Основанием для обработки персональных данных пользователя является её необходимость для реализации законных интересов корпоративного мессенджера (включая (1) предоставление эффективных и современных Услуг пользователям и (2) выявление, пресечение и другие меры по предотвращению мошенничества или угроз безопасности в связи с предоставлением Услуг), за исключением случаев, когда над интересами приложения преобладают интересы пользователя или фундаментальные права и свободы, требующие защиты персональных данных.

3. Виды используемых персональных данных

3.1. Основные сведения об учетной записи

PigeonGram – корпоративный мессенджер. При создании учетной записи в PigeonGram пользователь указывает основные сведения об учетной записи (имя, фамилию, адрес электронной почты, фотографию)

Выбранные имя и фотография пользователя всегда доступны другим пользователям корпоративного мессенджера PigeonGram, чтобы сотрудникам компании и другим людям было проще связаться друг с другом.

Адрес электронной почты указывается для возможности восстановления пароля.

3.2. Сообщения пользователей

3.2.1. Облачные чаты

PigeonGram – облачный сервис. Сообщения и другие файлы хранятся на устройствах пользователей и передаются через сервер. При передаче данных используются методы шифрования, поэтому данные надежно защищены.

4. Безопасность персональных данных

4.1. Хранение данных

При регистрации данные пользователя сохраняются в базе данных администратора. Все данные надежно зашифрованы и сторонние лица не могут получить к ним доступ.

4.2. Срок хранения данных

Персональные данные хранятся до тех пор, пока они необходимы для выполнения обязательств по предоставлению Услуг.

5. Обработка персональных данных

5.1. Предоставляемые услуги

PigeonGram – корпоративный мессенджер, который является важной коммуникационной платформой, используемой сотрудниками компании для обмена сообщениями и файлами, поэтому необходимо принять соответствующие меры по обеспечению его безопасности.

5.2. Безопасность и защита

Основными пользователями корпоративного мессенджера являются предприятия, которые нуждаются в защите своих данных от посторонних лиц. В качестве защиты данных используются надежные алгоритмы шифрования, предотвращающие несанкционированный доступ.

5.3. Спам и злоупотребления

В целях борьбы с мошенничеством, рассылкой спама и другими видами нарушений Пользовательского соглашения PigeonGram модераторы программного обеспечения могут проверять сообщения, на которые поступают жалобы от получателей. Если после проверки модераторы подтвердят, что сообщение является спамом, учётная запись может быть ограничена — временно или бессрочно. В период действия ограничений Вы не сможете связаться с пользователями не из списка контактов. Оспорить решение модераторов можно через техническую поддержку. В случае более серьёзных нарушений учётная запись пользователя может быть полностью заблокирована. В целях борьбы со спамом и фишингом также модераторы могут применять автоматические алгоритмы для анализа сообщений в облачных чатах.

6. Лица, которым могут быть переданы персональные данные

6.1. Другие пользователи PigeonGram

Под другими пользователями PigeonGram понимаются пользователи, с которыми клиент по своему желанию общается и делится той или иной информацией. Принимая Пользовательское соглашение и общаясь с другими пользователями PigeonGram, дается согласие на передачу личных персональных данных другим пользователям согласно данной Политике конфиденциальности. Разработчики корпоративного мессенджера PigeonGram принимают все соответствующие технические и организационные меры (включая шифрование персональных данных), чтобы обеспечить такой уровень безопасности данных, который будет соответствовать возможным рискам.

7. Права пользователя в отношении предоставленных персональных данных

7.1. Права пользователя

Согласно действующему законодательству о защите данных, при определённых обстоятельствах пользователь обладает правами в отношении своих персональных данных. Пользователь имеет право: (1) запросить копию всех своих персональных данных, хранящихся в PigeonGram, и передать эту копию другому сервису; (2) удалить (см. пункт 8 ниже) или изменить свои персональные данные; (3) ограничивать обработку своих персональных данных или

возражать против неё; (4) исправлять любые неточные или неполные персональные данные, хранящиеся в PigeonGram; и (5) обратиться в национальные органы по защите данных с жалобой относительно обработки мессенджером его персональных данных.

7.3. Управление данными

Пользователь может управлять использованием своих данных в разделе Настройки > Конфиденциальность.

Если пользователь не готов предоставить PigeonGram минимальные необходимые данные для работы приложения, к сожалению, предоставить Услуги будет невозможно. В этом случае пользователь может удалить свою учётную запись на странице деактивации.

8. Удаление данных

8.1. Учётные записи

Удалить свою учётную запись в PigeonGram можно на странице деактивации. При удалении учётной записи все пользовательские данные в облачном хранилище PigeonGram будут удалены. После этого восстановить учётную запись будет невозможно.

8.2. Автоматическое удаление учётной записи

По умолчанию, если пользователь не входит в свою учётную запись и не использует PigeonGram в течение как минимум 6 месяцев, профиль удаляется вместе со всеми данными, которые хранятся в облачном хранилище PigeonGram. Период, после которого неактивная учётная запись автоматически удаляется, можно изменить в меню «Настройки».

9. Изменения в Политике конфиденциальности

Разработчики и модераторы корпоративного мессенджера оставляют за собой право пересматривать данную Политику конфиденциальности и по мере необходимости вносить в неё изменения. Любые изменения в данной Политике конфиденциальности вступают в силу с момента публикации новой версии приложения.

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

КОРПОРАТИВНЫЙ МЕССЕНДЖЕР «PIGEONGRAM»

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	105
1.1. Область применения.....	105
1.2. Краткое описание возможностей.....	105
1.3. Уровень подготовки пользователя.....	105
1.4. Перечень эксплуатационной документации	105
2. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ	106
3. ПОДГОТОВКА К РАБОТЕ	107
3.1. Состав и содержание дистрибутивного носителя данных	107
3.2. Порядок загрузки данных и программ	107
3.3. Порядок проверки работоспособности.....	107
4. ОПИСАНИЕ ОПЕРАЦИЙ.....	109
4.1. Выполняемые функции и задачи	109
4.2. Описание операций технологического процесса обработки данных, необходимых для выполнения задач	110
5. АВАРИЙНЫЕ СИТУАЦИИ.....	116
6. РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ	118

1. ВВЕДЕНИЕ

1.1. Область применения

Требования настоящего документа применяются при осуществлении коммуникации между сотрудниками внутри организации с помощью корпоративного мессенджера «PigeonGram».

1.2. Краткое описание возможностей

Корпоративный мессенджер «PigeonGram» (КМ PG) позволяет оптимизировать процесс передачи информации внутри компании, способствуют благоприятному рабочему процессу. Описываемый продукт предоставляет пользователю следующие возможности:

1. Создание своей учетной записи, то есть регистрация и авторизация.
2. Отправление сообщений другому пользователю и получение сообщений в ответ.
3. Изменение учетной записи.
4. Использование общего облачного хранилища.

1.3. Уровень подготовки пользователя

Пользователь КМ PG должен иметь опыт работы с ОС MS Windows (95/98/NT/2000/XP/7/10/11), навык работы с ПО Internet Explorer, сторонними мессенджерами, а также обладать следующими знаниями:

1. Знать как регистрировать учетную запись;
2. Знать каким образом работают классические мессенджеры;
3. Понимать какие форматы файлов существуют.

1.4. Перечень эксплуатационной документации

Перечень эксплуатационной документации, с которыми необходимо ознакомиться пользователю:

1. Корпоративный мессенджер «PigeonGram». Паспорт;
2. Корпоративный мессенджер «PigeonGram». Общее описание системы.

2. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

КМ PG предназначен для автоматизации и улучшения эффективности передачи сообщений между сотрудниками внутри организации. Он позволяет сотрудникам легко обмениваться информацией, делиться файлами и координировать свою работу без привязки к офису или рабочему месту.

Работа с корпоративным мессенджером «PigeonGram» возможна всегда, когда есть стабильное соединение с сетью, работающее электронное вычислительное устройство, на котором установлена данная программа и квалификация пользователя соответствует требованиям, перечисленным в п. 1.3. настоящего документа.

3. ПОДГОТОВКА К РАБОТЕ

3.1. Состав и содержание дистрибутивного носителя данных

Для работы с КМ PG необходимо следующее программное обеспечение: Internet Explorer (входит в состав операционной системы Windows).

3.2. Порядок загрузки данных и программ

Перед началом работы с КМ PG на рабочем месте пользователя необходимо выполнить следующие действия:

1. Необходимо зайти на сайт <https://github.com/lidi-kitti/PigeonGram> и выбрать файл Installer.
2. Во время загрузки в появившемся окне "Предупреждение о безопасности", которое будет содержать следующее: 'Хотите установить и выполнить "PigeonGram" ...'
3. Нажимаем на кнопку "Да".
4. После чего запустится установка PigeonGram на Ваш компьютер.
5. Выбираем кнопку Next и затем ОК.

3.3. Порядок проверки работоспособности

Для проверки доступности КМ PG с рабочего места пользователя необходимо выполнить следующие действия:

1. Открыть приложение PigeonGram, для этого необходимо кликнуть по ярлыку «PigeonGram» на рабочем столе или вызвать из меню «Пуск».

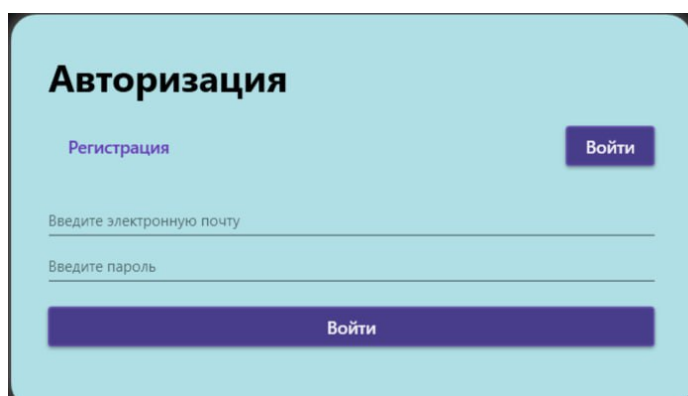


Рисунок 3.3.1. Авторизация

2. В форме аутентификации ввести пользовательский логин и пароль. Нажать кнопку «Далее» (см. рис. 3.3.1).
3. Убедиться, что в окне открылось приложение PigeonGram. (см. рис. 3.3.2)

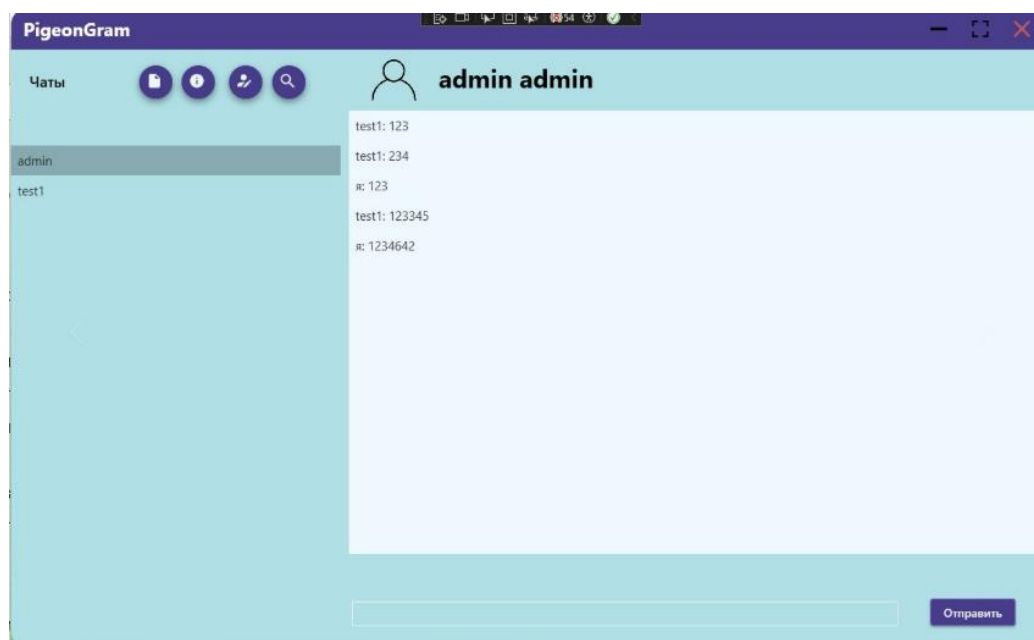


Рисунок 3.3.2. Главное окно приложения

4. В случае если приложение PigeonGram не запускается, следует обратиться в службу поддержки.

4. ОПИСАНИЕ ОПЕРАЦИЙ

4.1. Выполняемые функции и задачи

PigeonGram выполняет функции и задачи, которые описаны в табл. 4.1.1.

Таблица 4.1.1

Выполняемые функции и задачи

Функции	Задачи	Описание
Обеспечивает создание и использование личной учетной записи.	Регистрация пользователя	В ходе выполнения данной задачи пользователю приложения предоставляется возможность ввести свои личные данные для последующего использования во время входа в приложения.
	Вход с помощью ранее созданной учетной записи	В ходе выполнения этой задачи пользователь, используя свое имя и пароль, входит в свою учетную запись и видит свои сообщения.
Управление сообщениями.	Отправка сообщений	В ходе выполнения этой задачи пользователь создает и отправляет сообщение другому пользователю.
	Получение сообщений	В ходе выполнения этой задачи пользователь получает сообщение от другого пользователя.
Работа с файлами.	Сохранение файлов в общее хранилище.	В ходе выполнения этой задачи пользователь сохраняет необходимые файлы в общее хранилище, к которому у него есть доступ.
Управление учетной записью	Редактирование учетной записи	В ходе выполнения данной задачи пользователь изменяет данные аккаунта и имеет возможность добавить фотографию.
Работа с чатами	Поиск чата	В ходе выполнения этой задачи пользователь с помощью поисковой строки находит нужный ему чат.

4.2. Описание операций технологического процесса обработки данных, необходимых для выполнения задач

Ниже приведено описание пользовательских операций для выполнения каждой из задач.

Задача: «Регистрация пользователя»

Операция 1: Регистрация в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

В открывшемся окне (см. рис. 4.2.1) в поле «Введите имя» ввести имя пользователя, в поле «Введите фамилию» ввести фамилию пользователя, в поле «Введите электронную почту» ввести email пользователя, в поле «Введите пароль» ввести пароль пользователя и в поле «Повторите пароль» ввести пароль пользователя повторно. Нажать кнопку «Зарегистрироваться».

Рисунок 4.2.1. Регистрация пользователя

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд.

Задача: «Вход с помощью ранее созданной учетной записи»

Операция 1: Вход в приложение PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.

4. В приложении PigeonGram ранее была создана учетная запись.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

В открывшемся окне (см. рис. 3.3.1) в поле «Введите электронную почту» ввести email пользователя, в поле «Введите пароль» ввести пароль пользователя. Нажать кнопку «Войти».

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд.

Задача: «Отправка сообщения»

Операция 1: Отправка сообщения в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.
5. Ранее был создан чат

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

На главном экране (см. рис. 3.3.2), где представлены ранее созданные чаты, нажимаем левой кнопкой мыши на чат. Появляется окно чата, а внизу присутствует место для набора сообщения и кнопка «Отправить». Набираем необходимый текст и нажимаем на кнопку. Сообщение отправиться другому пользователю.

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

Задача: «Сохранение файла в общем файловом хранилище»

Операция 1: Сохранение файла в облачное хранилище в приложении PigeonGram, использую облачное хранилище.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

На главном экране (см. рис. 3.3.2) нажимаем на иконку, изображающую документы (см. рис. 4.2.2)

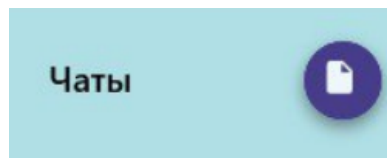


Рис. 4.2.2. Иконка, направляющая в общее файловое хранилище

Появляется следующее окно, представленное на рис. 4.2.3. Здесь можно увидеть информацию о пользователях и о документах, которые ранее были загружены. Чтобы загрузить документ необходимо нажать на иконку загрузки (вторая от слова «Документы») и выбрать необходимый документ в системном окне.

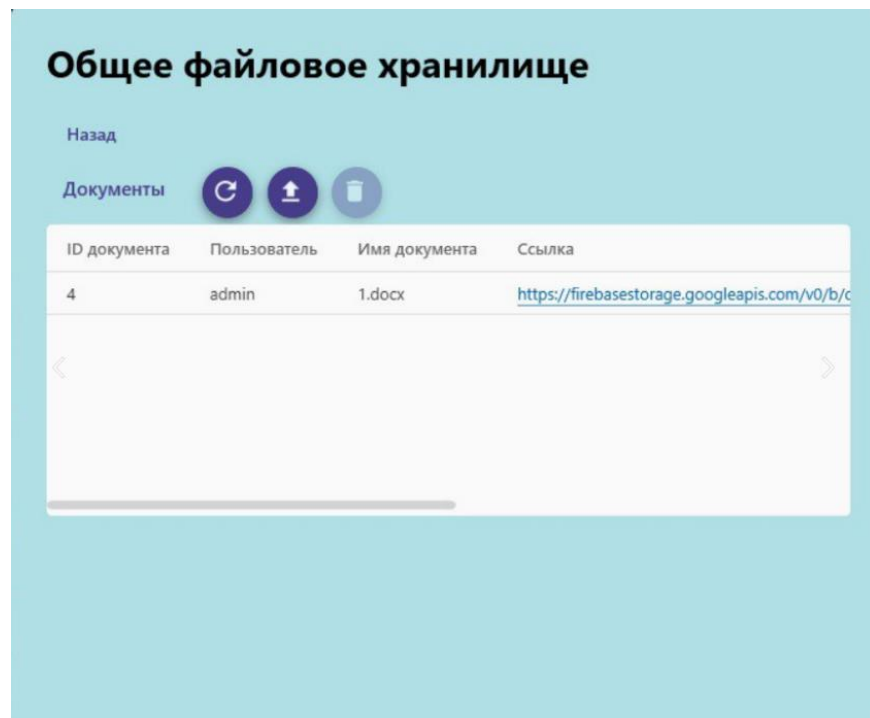


Рис. 4.2.3. Окно общего файлового хранилища

После успешной загрузки появится сообщение (см. рис. 4.2.4).

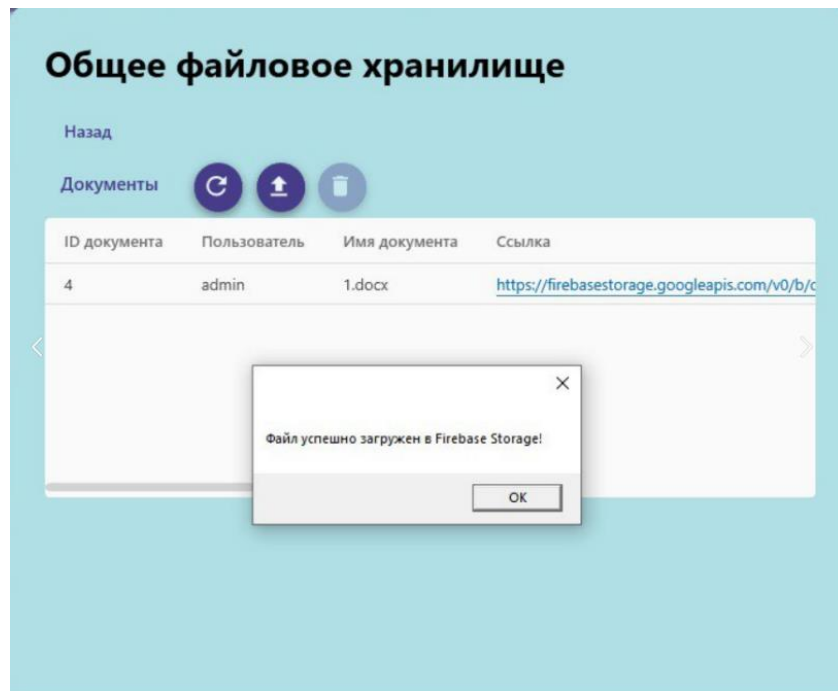


Рис. 4.2.4. Сообщение об успешном сохранении файла

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

Задача: «Редактирование учетной записи»

Операция 1: Редактирование учетной записи в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.
5. Ранее был создан чат

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

На главном экране нажимаем на иконку (см. рис. 4.2.5) и переходим в окно редактирования (см. рис. 4.2.6). Далее появляется возможность изменить имя, фамилию, электронную почту и логин, а также добавить фотографию. После изменений необходимо нажать на кнопку «Сохранить изменения» и дождаться сообщения, представленного на рис. 4.2.7.



Рис. 4.2.5. Иконка «Редактирование профиля»

Мой профиль

[Назад](#)

Лида Преснухина

lida@mail.ru

lida@mail.ru

Сохранить изменения

Рис. 4.2.6. Окно редактирование профиля

Мой профиль

[Назад](#)

Лида Преснухина

lida@mail.ru

lida@mail.ru

Сохранить изменения

Изменения успешно сохранены

ОК

Рис. 4.2.7. Успешное сохранение изменений

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

Задача: «Поиск чата»

Операция 1: Поиск необходимого чата в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.
5. Ранее был создан чат

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

На главном экране, где представлены ранее созданные чаты. Нажимаем левой кнопкой мышки на иконку, похожую на лупу (см. рис. 4.2.8) и в появившуюся строку вводим названия искомого чата.



Рисунок 4.2.8. Функциональные иконки

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

Задача: «Получение сообщений»

Операция 1: Получение сообщений в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

На главном экране, где представлены ранее созданные чаты. Нажимаем левой кнопкой мышки на чат. Появляется окно, где будут изображаться ранее отправленные сообщения, видим новые сообщения (см. рис. 3.3.2)

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

5. АВАРИЙНЫЕ СИТУАЦИИ

В случае возникновения ошибок при работе с корпоративным мессенджером «PigeonGram», не описанных ниже в табл. 5.1 данного раздела, необходимо обращаться к сотруднику подразделения технической поддержки либо к ответственному Администратору КМ РГ.

Таблица 5.1

Описание возможных ошибок и их устранения в ходе работы программы

Класс ошибки	Ошибка	Описание ошибки	Требуемые действия пользователя при возникновении ошибки
Установка	Сбой в установке.	Сбой в установке приложения.	Проверить подключение к сети и наличие свободного места на диске. В противном случае обратиться к сотруднику технической поддержки.
Портал КМ РГ	Сервер не найден. Невозможно отобразить сообщение.	Возможны проблемы с сетью или с доступом к portalу КМ РГ.	Для устранения проблем с сетью обратиться к сотруднику подразделения технической поддержки (HelpDesk). В других случаях к администратору КМ РГ.
	Ошибка: требуется ввести действительное имя пользователя	При регистрации на портале КМ РГ не введено имя пользователя.	Ввести имя пользователя.
	Ошибка: требуется ввести пароль для регистрации	При регистрации на портале КМ РГ не введен пароль.	Ввести пароль.
	Ошибка: сбой аутентификации. Повторите попытку	Неверно введено имя пользователя или пароль, либо такая учетная запись не зарегистрирована.	Нужно повторить ввод имени пользователя и пароля. Если учетная запись заблокирована, нужно обратиться к администратору КМ РГ.

Сбой в отправке, доставке, пересылке сообщений	Сообщение не доставлено.	Сообщение не отправляется.	Проверить подключение к интернету, а также подключение к серверу. Проверить содержание сообщения, оно не должно быть пустым. В случае стабильного подключения — перезагрузить программу и попробовать еще раз.
	Сообщение доставлено.	Сообщение доставлено.	Проверить подключение к интернету, а также подключение к серверу. В случае стабильного подключения — перезагрузить программу.
	Невозможно переслать сообщение.	Сообщение не пересылается.	Проверить подключение к интернету, а также подключение к серверу. Проверить содержание сообщения, оно не должно быть пустым. В случае стабильного подключения — перезагрузить программу и попробовать еще раз.
Исходный код	Ошибка ключей	Не отправляются сообщения. Появляется сообщение о том, что контрольные суммы не совпадают.	Переустановите приложения и запустите повторно, если ошибка не исчезла, обратитесь в техническую поддержку.

6. РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ

В наше время, где коммуникация играет ключевую роль в бизнесе, корпоративные мессенджеры становятся неотъемлемой частью рабочей среды. С их помощью можно легко и быстро обмениваться информацией с коллегами, решать рабочие вопросы и координировать свою команду. Однако, многие сотрудники часто испытывают трудности в освоении новых инструментов. Далее представлены полезные рекомендации по освоению корпоративного мессенджера:

1. Изучите функционал. Прежде всего, ознакомьтесь с функциями, которые представляет корпоративный мессенджер «PigeonGram». Инвестируйте время в изучение всех возможностей, чтобы максимально глубоко понять, какие инструменты помогут в работе.

2. Создайте профиль. Настройте свой профиль в мессенджере. Укажите полезную информацию о себе - имя, должность, контактные данные. Это поможет вашим коллегам легко найти и связаться с вами, а также создаст положительное впечатление о вашей профессиональности.

3. Используйте чаты и группы. Создайте или присоединитесь к соответствующим чатам и группам, которые касаются ваших рабочих задач. Постепенно освоитесь с функциями мессенджера, такими как общие дискуссии, комментарии, чат-боты и прочее. Это позволит вам быть в курсе всех текущих процессов и событий в вашей команде.

4. Не забывайте о безопасности. Соблюдайте правила безопасности при работе с корпоративным мессенджером. Старайтесь отправлять конфиденциальную информацию только в закрытых чатах, используйте двухфакторную аутентификацию для повышения защиты вашего аккаунта и следите за обновлениями безопасности, чтобы исключить возможность взлома.

5. Обучайтесь. Если у вас возникают затруднения или вопросы при работе с мессенджером, обратитесь за помощью к IT-специалистам, которые могут предоставить вам дополнительную информацию и настроить мессенджер, чтобы сделать его максимально удобным для работы.

В заключение, освоение корпоративного мессенджера может показаться сложным, однако, следуя этим рекомендациям, можно улучшить коммуникацию внутри команды. Не нужно бояться идти в ногу со временем и использовать новые технологии, ведь именно это способствует продуктивности и развитию конкурентоспособными.

РУКОВОДСТВО АДМИНИСТРАТОРА

КОРПОРАТИВНЫЙ МЕССЕНДЖЕР «PIGEONGRAM»

РУКОВОДСТВО АДМИНИСТРАТОРА

СОДЕРЖАНИЕ

1. АННОТАЦИЯ	121
2. НАЗНАЧЕНИЕ ПРОГРАММЫ	122
3. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ	123
3.1. Серверная часть	123
3.1.1. Комплекс технических средств.....	123
3.1.2. Программное обеспечение.....	123
3.2. Клиентская часть	124
3.2.1. Комплекс технических средств.....	124
3.2.2. Программное обеспечение.....	124
4. ОПИСАНИЕ ОПЕРАЦИЙ	125
4.1. Выполняемые функции и задачи	125
4.2. Описание операций технологического процесса обработки данных, необходимых для выполнения задач	126

1. АННОТАЦИЯ

Корпоративный мессенджер — это современный инструмент коммуникации, который обеспечивает эффективное взаимодействие внутри компании. Он позволяет сотрудникам легко обмениваться информацией, делиться файлами и координировать свою работу без привязки к офису или рабочему месту. Представленный документ демонстрирует полное описание функций и возможностей мессенджера, которые могут быть важными при быстром реагировании на различные аварийные ситуации.

Руководство администратора адресовано лицу, задача которого — обеспечить определенный порядок функционирования системы. Обычно администратор считается пользователем системы, однако, при этом он наделен как особыми обязанностями, так и необходимыми для их выполнения привилегиями.

2. НАЗНАЧЕНИЕ ПРОГРАММЫ

КМ PG предназначен для автоматизации и улучшения эффективности передачи сообщений между сотрудниками внутри организации. Он позволяет сотрудникам легко обмениваться информацией, делиться файлами и координировать свою работу без привязки к офису или рабочему месту.

Корпоративный мессенджер «PigeonGram» (КМ PG) позволяет оптимизировать процесс передачи информации внутри компании, способствуют благоприятному рабочему процессу. Описываемый продукт предоставляет пользователю следующие возможности:

1. Создание своей учетной записи, то есть регистрация и авторизация.
2. Отправление сообщений другому пользователю и их получение.
3. Сохранение файлов в общем файловом хранилище.
4. Поиск чатов.
5. Редактирование профиля.

Во время работы корпоративного мессенджера «PigeonGram» администратор следит за процессом обработки информации, которая передается между клиентской и серверной частью приложения. Отвечает за блокировку пользователей, за восстановление данных учетных записей пользователя. Обеспечивает актуализацию и безопасность мессенджера через регулярные обновления и устранение ошибок. Анализирует активность пользователей, следит за работоспособностью общего файлового хранилища.

3. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Администраторы КМ РГ играют важную роль в обеспечении безопасности, функциональности и удобства использования приложения. В связи с этим, предъявляются определенные требования к квалификации и навыкам специалистов.

1. Технические навыки. Администратор должен обладать хорошими знаниями и опытом в области информационных технологий. Важно, чтобы администратор умел разбираться в сетевой архитектуре, протоколах обмена информацией и системах безопасности. Уверенное владение основными операционными системами (Windows, macOS, Linux) и мессенджерами также является важным требованием.

2. Знания о безопасности. Администраторы мессенджера должны иметь хорошие навыки в области защиты данных и принятия мер по обеспечению конфиденциальности. Понимание методов шифрования, аутентификации и авторизации, а также способность эффективно реагировать на уязвимости и угрозы безопасности – необходимые компетенции для администраторов.

3. Коммуникативные навыки. Администратор мессенджера должен быть коммуникабельным и обладать отличными навыками общения. Взаимодействие с пользователями и разработчиками требует умения ясно и четко выражать свои мысли, решать проблемы и отвечать на вопросы. Умение работать в команде также является важным качеством.

4. Аналитические способности. Администраторы мессенджера должны быть внимательными к деталям и уметь анализировать большие объемы данных для выявления проблем и повышения эффективности работы приложения. Умение принимать взвешенные решения на основе анализа информации – важный навык для администратора.

3.1. Серверная часть

3.1.1. Комплекс технических средств

Требования к техническим средствам для размещения серверной части приведены в таблице 3.1.1.1.

Таблица 3.1.1.1

Требования к техническим средствам для размещения серверной части

Компонент	Минимальная конфигурация
Процессор	Не менее 4 ядер CPU
Объем оперативной памяти	Не менее 8 Гбайт
Свободное дисковое пространство	Не менее 40 Гбайт

3.1.2. Программное обеспечение

Для обеспечения работы серверной части ПО должно быть установлено программное обеспечение, перечень которого приведен в таблице 3.1.2.1.

Таблица 3.1.2.1

Требования к составу программного обеспечения серверной части КМ РГ

Компонент	Конфигурация
Операционная система	Windows или допустимый аналог
СУБД	MySQL

Компонент	Конфигурация
Общесистемное ПО	Docker 20.10 PHP 7.1 Composer v1

3.2. Клиентская часть

3.2.1. Комплекс технических средств

Для работы с ПО рабочие станции пользователей должны удовлетворять требованиям к техническому обеспечению согласно таблице 3.2.1.1

Таблица 3.2.1.1

Требования к техническим характеристикам рабочих станций пользователей

Компонент	Минимальная конфигурации	Рекомендуемая конфигурация
Частота процессора	1,0 ГГц	2,0 ГГц и более
Объем оперативной памяти	2 Гбайт	3 Гбайт и более
Свободное дисковое пространство	3 Гбайт	5 Гбайт и более

3.2.2. Программное обеспечение

На рабочих местах должно быть установлено программное обеспечение, перечень которого приведен в таблице 3.2.2.1.

Таблица 3.2.2.1

Требования к составу программного обеспечения рабочих мест пользователей КМ РГ

Компонент	Конфигурация
Операционная система	Windows или допустимый аналог
Общесистемное ПО	Последние версии следующих Интернет-браузеров (Google Chrome, Microsoft Edge и др.)

4. ОПИСАНИЕ ОПЕРАЦИЙ

4.1. Выполняемые функции и задачи

PigeonGram выполняет функции и задачи, которые описаны в табл. 4.1.1.

Таблица 4.1.1

Выполняемые функции и задачи

Функции	Задачи	Описание
Обеспечивает создание и использование учетной записи.	Регистрация новых пользователей.	В ходе выполнения данной задачи администратору приложения предоставляется возможность регистрации нового пользователя.
	Вход в учетную запись администратора	В ходе выполнения этой задачи администратор, используя свое имя и пароль, входит в свою учетную запись и видит все функциональные возможности, которые есть у администратора.
Управление сообщениями.	Отправка сообщений	В ходе выполнения этой задачи администратор создает и отправляет сообщение другому пользователю.
	Получение сообщений	В ходе выполнения этой задачи администратор получает сообщение от другого пользователя.
Работа с файлами.	Сохранение файлов в общее хранилище	В ходе выполнения этой задачи администратор сохраняет необходимые файлы в общее хранилище, к которому у него есть доступ.
	Удаление файлов из общего хранилища	В ходе выполнения этой задачи администратор может выбрать файл, который нужно удалить.
Управление учетной записью	Редактирование учетной записи	В ходе выполнения данной задачи пользователь изменяет данные аккаунта и имеет возможность добавить фотографию.

Функции	Задачи	Описание
Работа с чатами	Поиск чата	В ходе выполнения этой задачи пользователь с помощью поисковой строки находит нужный ему чат.
Просмотр статистики	Просмотр данных о зарегистрированных пользователях	В ходе выполнения этой задачи администратор с помощью таблицы может увидеть всю информацию о пользователях, кроме паролей от их аккаунтов
	Просмотр активности пользователей в выбранный промежуток времени	В ходе выполнения этой задачи администратор, с помощью графика, может увидеть статистику активности определенного пользователя.

4.2. Описание операций технологического процесса обработки данных, необходимых для выполнения задач

Ниже приведено описание администраторских операций для выполнения каждой из задач.
Задача: «Регистрация новых пользователей»

Операция 1: Регистрация новых пользователей в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер администратора подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

В открывшемся окне (см. рис. 4.2.1) в поле «Введите имя» ввести имя пользователя, в поле «Введите фамилию» ввести фамилию пользователя, в поле «Введите электронную почту» ввести email пользователя, в поле «Введите пароль» ввести пароль пользователя и в поле «Повторите пароль» ввести пароль пользователя повторно, также необходимо выбрать статус пользователя, то есть кем он является: пользователь или администратор. Нажать кнопку «Зарегистрировать пользователя».

Рисунок 4.2.1. Регистрация пользователя

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд.

Задача: «Вход с помощью ранее созданной учетной записи»

Операция 1: Вход в приложение PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. В приложении PigeonGram ранее была создана учетная запись.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

В открывшемся окне (см. рис. 3.3.1) в поле «Введите электронную почту» ввести email пользователя, в поле «Введите пароль» ввести пароль пользователя. Нажать кнопку «Войти».

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд.

Задача: «Отправка сообщения»

Операция 1: Отправка сообщения в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.

5. Ранее был создан чат

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

На главном экране (см. рис. 3.3.2), где представлены ранее созданные чаты, нажимаем левой кнопкой мыши на чат. Появляется окно чата, а внизу присутствует место для набора сообщения и кнопка «Отправить». Набираем необходимый текст и нажимаем на кнопку. Сообщение отправиться другому пользователю.

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

Задача: «Сохранение файла в общем файловом хранилище»

Операция 1: Сохранение файла в облачное хранилище в приложении PigeonGram, использую облачное хранилище.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

На главном экране (см. рис. 3.3.2) нажимаем на иконку, изображающую документы (см. рис. 4.2.2)



Рис. 4.2.2. Иконка, направляющая в общее файловое хранилище

Появляется следующее окно, представленное на рис. 4.2.3. Здесь можно увидеть информацию о пользователях и о документах, которые ранее были загружены. Чтобы загрузить документ необходимо нажать на иконку загрузки (вторая от слова «Документы») и выбрать необходимый документ в системном окне.

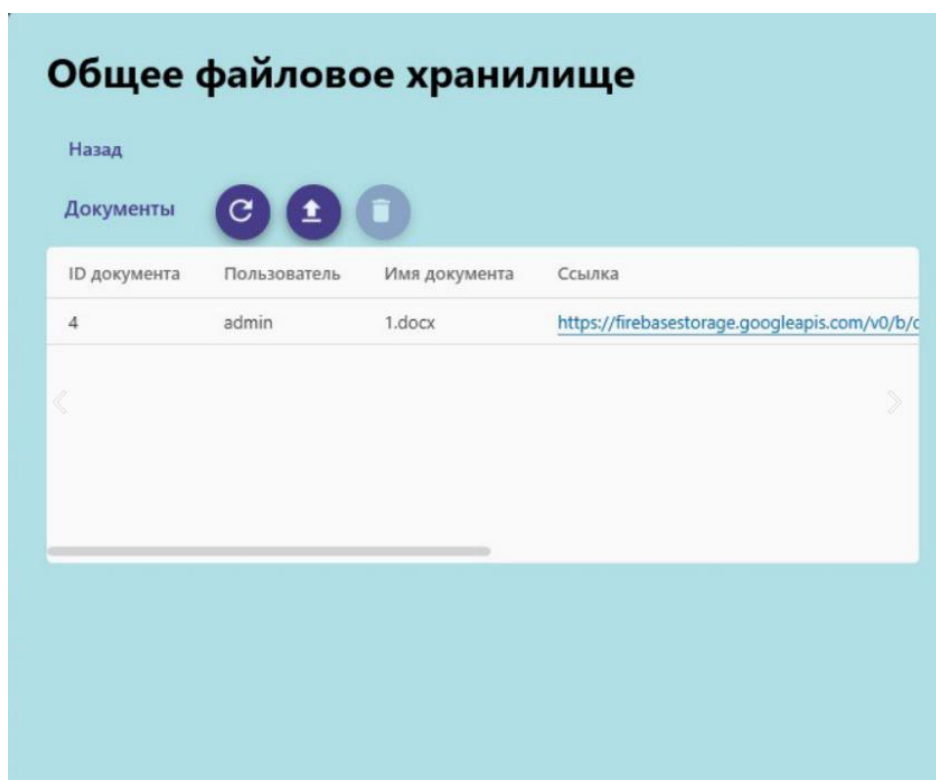


Рис. 4.2.3. Окно общего файлового хранилища

После успешной загрузки появится сообщение (см. рис. 4.2.4).

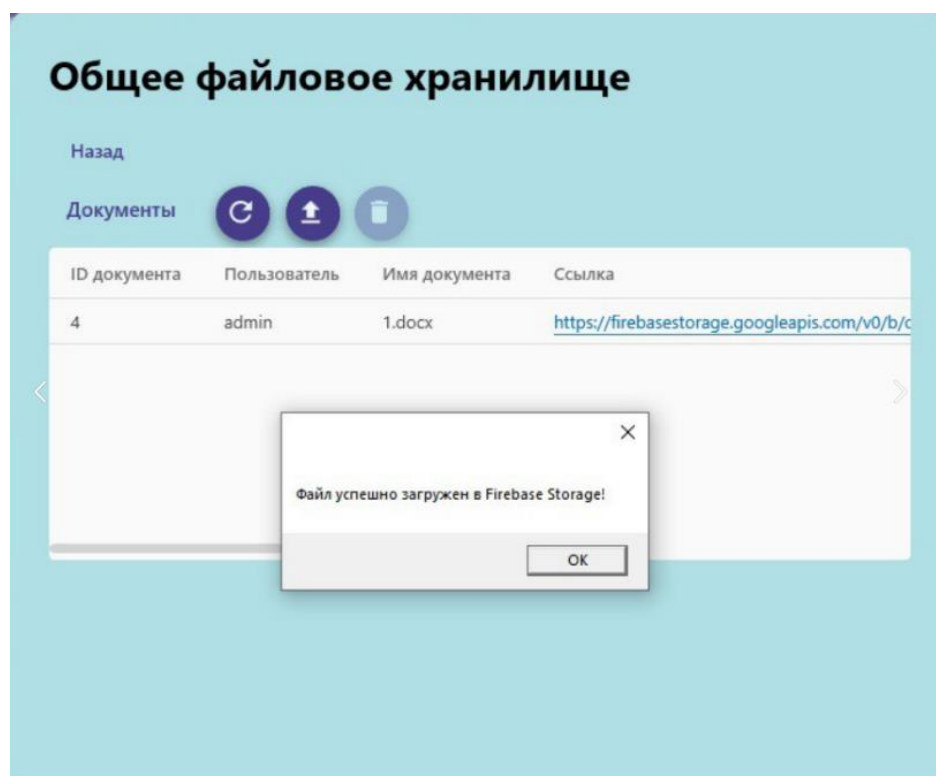


Рис. 4.2.4. Сообщение об успешном сохранении файла

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

Задача: «Редактирование учетной записи»

Операция 1: Редактирование учетной записи в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.
5. Ранее был создан чат

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

На главном экране нажимаем на иконку (см. рис. 4.2.5) и переходим в окно редактирования (см. рис. 4.2.6). Далее появляется возможность изменить имя, фамилию, электронную почту и логин, а также добавить фотографию. После изменений необходимо нажать на кнопку «Сохранить изменения» и дождаться сообщения, представленного на рис. 4.2.7.



Рис. 4.2.5. Иконка «Редактирование профиля»

The screen has a light blue background. At the top, the text 'Мой профиль' is displayed in a large, bold, black font. Below it, the word 'Назад' is in a smaller, purple font. In the center is a large white square containing a blue circular profile picture placeholder with a white person silhouette. A small purple circle with a white camera icon is at the bottom right of the profile picture. Below the picture, the name 'Лида Преснухина' is shown in a grey font, followed by the email 'lida@mail.ru' twice. At the bottom is a dark blue button with the white text 'Сохранить изменения'.

Рис. 4.2.6. Окно редактирование профиля

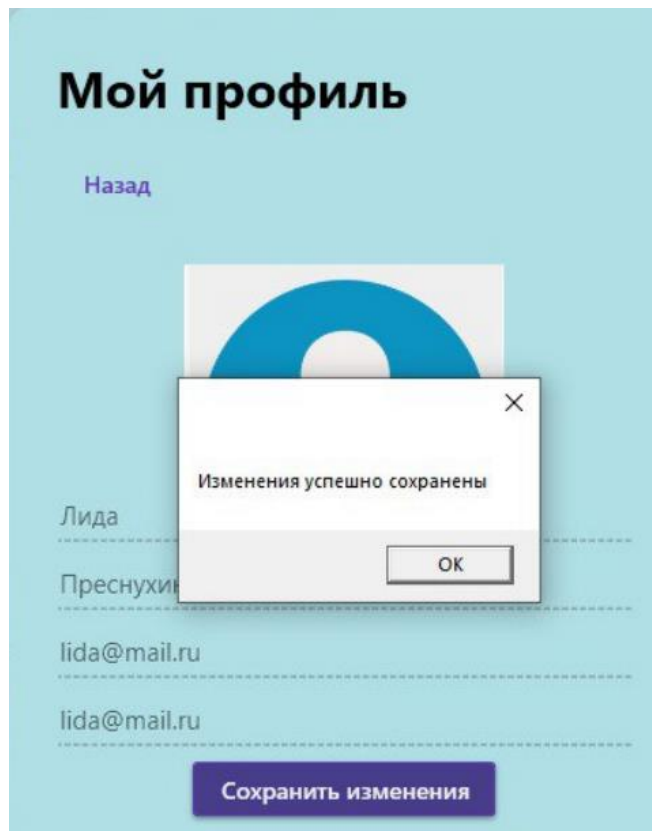


Рис. 4.2.7. Успешное сохранение изменений

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

Задача: «Поиск чата»

Операция 1: Поиск необходимого чата в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.
5. Ранее был создан чат

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

На главном экране, где представлены ранее созданные чаты. Нажимаем левой кнопкой мыши на иконку, похожую на лупу (см. рис. 4.2.8) и в появившуюся строку вводим названия искомого чата.



Рисунок 4.2.8. Функциональные иконки

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

Задача: «Получение сообщений»

Операция 1: Получение сообщений в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

На главном экране, где представлены ранее созданные чаты. Нажимаем левой кнопкой мышки на чат. Появляется окно, где будут изображаться ранее отправленные сообщения, видим новые сообщения (см. рис. 3.3.2)

Заклучительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

Задача: «Удаление файлов из общего файлового хранилища»

Операция 1: Удаление файлов из общего файлового хранилища в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

В окне общего файлового хранилища (см. рис. 4.2.3) нажать на строку с файлом, который необходимо удалить и нажать на иконку с изображением «Корзина». Файл будет удален.

Заклучительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

Задача: «Просмотр статистики активности пользователя»

Операция 1: Просмотр статистики активности пользователя в приложении PigeonGram.

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к стабильной сети.
2. Приложение PigeonGram доступно.
3. PigeonGram функционирует в штатном режиме.
4. Произведен вход в приложение под личной учетной записью.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

На иконке «PigeonGram» рабочего стола произвести двойной щелчок левой кнопкой мышки.

На главном окне выбрать вторую слева иконку (см. рис. 4.2.8) и перейти в окно администратора (см. рис. 4.2.9)

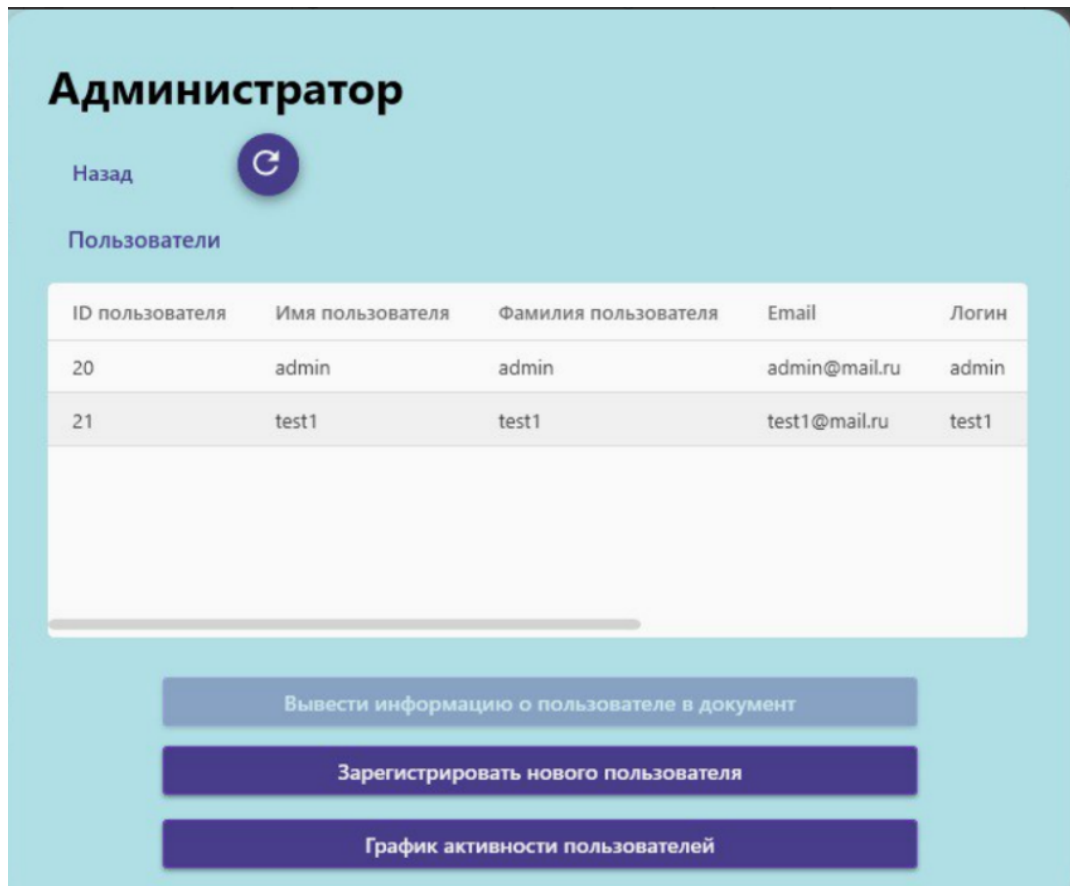


Рис. 4.2.9. Окно администратора

Далее необходимо нажать на кнопку «График активности пользователей» и перейти в окно, представленное на рис. 4.2.10. Выбрать нужного пользователя и даты, после чего нажать на кнопку «Рассчитать график» и результат будет выглядеть аналогичным образом (см. рис. 4.2.11).

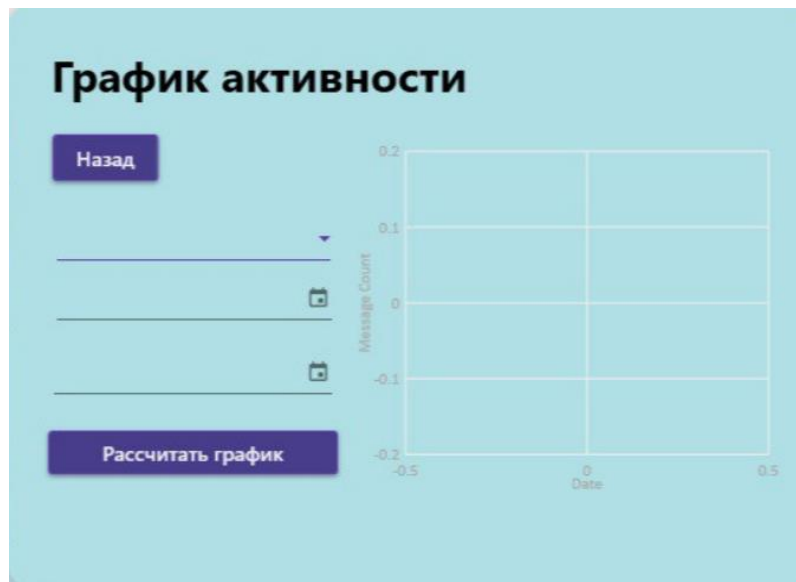


Рис. 4.2.10. График активности пользователей

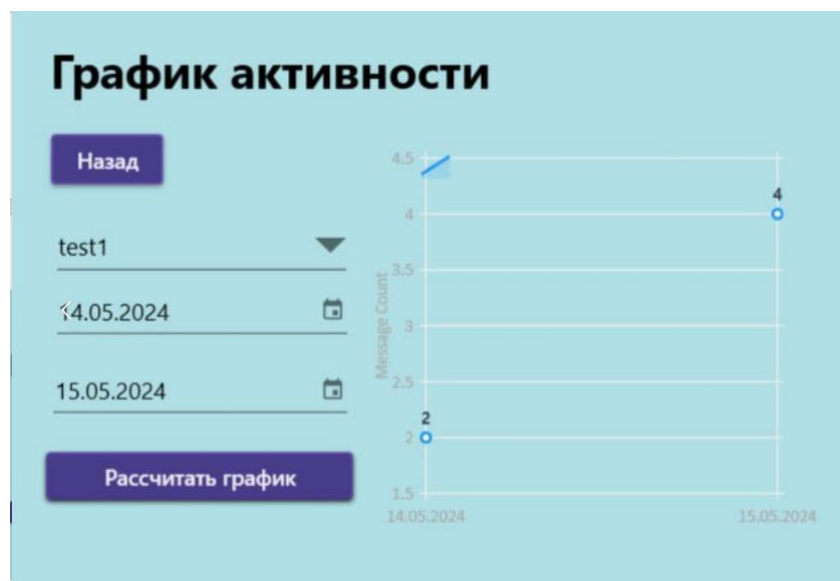


Рис. 4.2.11. Результат работы функции

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд

ПОЛЬЗОВАТЕЛЬСКОЕ СОГЛАШЕНИЕ**Пользовательское соглашение PigeonGram**

Регистрируясь в PigeonGram, пользователь принимает Политику конфиденциальности приложения и соглашается:

- Не использовать мессенджер для рассылки спама или обмана пользователей;
- Не призывать к насилию в публичных каналах, ботах и других общедоступных чатах в PigeonGram;
- Не размещать незаконный порнографический контент в каналах, ботах и других общедоступных чатах в PigeonGram.
- Разработчики и модераторы оставляют за собой право в дальнейшем вносить изменения и дополнения в данное Пользовательское соглашение.

ПРИЛОЖЕНИЕ 6

Модель угроз персональных данных

Угроза безопасности информации		Источники угрозы	Нарушитель	Актуальный нарушитель							Объекты воздействия				Способы реализации	Возможные угрозы
Угроза	Описание			ВН1	ВН2	ВН3	ВН4	ВН5	ВН6	ВН7	АРМ	Сервер	БД	ПО		Нарушение конфиденциальности и целостности. Нарушение доступности. Возможна .
УБИ.006: Угроза внедрения кода или данных	Угроза заключается в возможности внедрения и запуска вредоносного кода от имени доверенного процесса на любом из ресурсных центров грид-системы и его автоматического распространения	Внешний нарушитель с низким потенциалом		+	+	+	-	+	+	+	+	-	+	+	в случае работы дискредитируемого пользователя с файлами, поступающим и из недоверенных источников; при наличии у него привилегий установки программного обеспечения; в случае неизмененных владельцем учетных данных IoT-устройства (заводских пароля и логина)	Нарушение конфиденциальности и целостности. Нарушение доступности. Возможна .

[illegible]

	ля приви легий легал ьного пользо вателе ля грид- систе мы														
УБИ .015: Угро за дост упа к защ ища емы м фай лам с испо льзо вани ем обхо дног о пути	Угроз а заклю чае тс я в возмо жнос ти получ ения нару шите лем досту па к скрыт ым/за щища емым катал огам или файла м посре дство м разли чных возде йстви й на файл овую систе му (доба влени е допол нител	Внешний нарушител ь с низким потенциал ом Внутренни й нарушител ь с низким потенциал ом	+	-	+	+	+	+	+	+	+	+	+	Реализация данной угрозы возможна при условиях: наличие у нарушителя прав доступа к некоторым объектам файловой системы; отсутствие проверки вводимых пользователе м данных; наличие у дискредитиру емой программы слишком высоких привилегий доступа к файлам, обработка которых не предполагаетс я с её помощью	Нарушен ие конфиден циальност и. Возможна .

	БНЫХ СИМВО ЛОВ В УКАЗА НИИ ПУТИ К ФАЙЛ У; ОБРА ЩЕНИ Е К ФАЙЛА М, КОТО РЫЕ ЯВНО НЕ УКАЗА НЫ В ОКНЕ ПРИЛО ЖЕНИ Я).														
УБИ .033: Угро за испо льзо вани я слаб осте й коди рова ния вход ных данн ых	Угроз а заклю чае тся в возмо жнос ти осу ществ ления нару шите лем дестр уктив ного инфо рмац ионно го возде йстви я на дискр едити руем ую	Внешний нарушитель со средним потенциал ом Внутренний нарушитель со средним потенциал ом	+	+	-	-	+	+	+	+	-	-	+	Данная угроза обусловлена слабостями механизма контроля входных данных. Реализация данной угрозы возможна при условиях: дискредитируемая система принимает входные данные от нарушителя; нарушитель обладает возможностью управления одним или несколькими параметрами входных данных	Нарушен ие целостнос ти Нарушен ие доступно сти. Возможна .

	системе путём манипулирования значениями и входных данных и формой их предоставления (альтернативные кодировки, некорректное расширение файлов и т.п.).														
УБИ.088: Угроза несанкционированного копирования защищаемой информации	Угроза заключается в возможности неправомерного получения нарушителем копии защищаем	Внешний нарушитель с низким потенциалом Внутренний нарушитель с низким потенциалом	+	+	+	-	+	+	+	+	+	+	-	Данная угроза обусловлена слабостями механизмов разграничения доступа к защищаемой информации и контроля доступа лиц в контролируемой зоне. Реализация данной угрозы возможна в случае отсутствия криптографич	Нарушение конфиденциальности. Возможна.

аци и	ой инфо рмац ии путём прове дения после доват ельно сти непра вомер ных дейст вий, вклю чающ их: несан кцион ирова нный досту п к защи щаем ой инфо рмац ии, копир овани е нажде нной инфо рмац ии на съёмн ый носит ель (или в друго е место , досту пное нару												еских мер защиты или снятия копии в момент обработки защищаемой информации в нешифрованн ом виде	
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--

	шите лю вне систе мы).														
УБИ .090: Угро за неса нкц ион иров анно го созд ания учёт ной запи си поль зова теля	Угроз а заключаетс я в возмо жнос ти созда ния нару шите лем в систе ме допол нител ьной учётн ой запис и пользо вателе и её дальн ейшег о испол ьзова ния в собст венн ых непра вомер ных целях (вход а в систе му с права ми этой учётн	Внешний нарушител ь с низким потенциал ом Внутренни й нарушител ь с низким потенциал ом	+	+	+	-	+	+	+	+	-	-	-	Данная угроза обусловлена слабостями механизмов разграничени я доступа к защищаемой информации. Реализация данной угрозы возможна в случае наличия и прав на запуск специализиро ванных программ для редактирован ия файлов, содержащих сведения о пользователях системы (при удалённом доступе) или штатных средств управления доступом из состава операционной системы (при локальном доступе)	Нарушен ие конфиден циальност и Нарушен ие целостнос ти Нарушен ие доступно сти. Возможна .

	ой запис и и осуш ествл ения дестр уктив ных дейст вий по отно шени ю к дискр едити рован ной систе ме или из дискр едити рован ной систе мы по отно шени ю к други м систе мам).														
УБИ .115: Угро за пере хват а ввод имо й и выв оди мой на пери фер	Угроз а заключаетс я в возмо жнос ти осуш ествл ения нару шите лем несан кцион	Внешний нарушител ь с низким потенциал ом Внутренни й нарушител ь с низким потенциал ом	+	+	-	+	-	-	-	-	+	+	-	Данная угроза обусловлена недостаточностью мер защиты информации от утечки и контроля потоков данных, а также невозможностью осуществления защиты вводимой и	Нарушен ие конфиден циальност и. Возможна .

ийн ые устр ойст ва инф орм аци и	ирова нного досту па к инфо рмац ии, вводи мой и вывод имой на пери фери йные устро йства, путём перех вата данн ых, обраб атыва емых контр оллер ами пери фери йных устро йств.													выводимой на периферийны е устройства информации с помощью криптографич еских средств (т.к. представлени е пользователя м системы информации должно осуществлять ся в доступном для понимания виде). Реализация данной угрозы возможна при условии наличия у нарушителя привилегий на установку и запуск специализиро ванных вредоносных программ, реализующих функции «клавиатурны х шпионов» (для получения нарушителем паролей пользователей), виртуальных драйверов принтеров (перехват документов, содержащих защищаемую	
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

														информацию) и др.	
--	--	--	--	--	--	--	--	--	--	--	--	--	--	----------------------	--