**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**

**Предметная (цикловая) комиссия** *Информатики и программирования в компьютерных системах*

Допустить к защите

Заместитель директора

по учебной работе

\_\_\_\_\_\_\_\_\_\_ Н.В. Калинина

14 июня 2023 г.

|  |  |  |  |
| --- | --- | --- | --- |
| **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА** | | | |
| Тема | **Разработка корпоративной системы коммуникации** | | |
|  |  | | |
| Вид выпускной квалификационной работы | | | *Дипломный проект* |
|  | | | *(дипломный проект, дипломная работа)* |
|  | | | |
| Специальность | | *09.02.07 Информационные системы и программирование* | |
| Квалификация | | *Программист* | |

|  |  |  |  |
| --- | --- | --- | --- |
| Дипломник | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Преснухина Л. И. |
|  |  | | 10 июня 2024 г. |
| Руководитель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Кривоносова Н. В. |
|  |  | | 10 июня 2024 г. |
| Консультант по экономической части | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Афанасьева Н.И. |
|  |  | | 10 июня 2024 г. |
| Консультант по охране труда | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Ожерельева Н.А. |
|  |  | | 10 июня 2024 г. |

Санкт-Петербург

2024

|  |  |  |
| --- | --- | --- |
| Работа написана мною и не содержит неправомерных заимствований | | |
| 10.06.2024 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| *(дата)* | | *(подпись студента)* |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
| Текст ВКР размещен в электронно-библиотечной системе университета | | |
| Руководитель отдела комплектования библиотеки | |  |
|  | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(ФИО)* |
| 10.06.2024 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| *(дата)* | | *(подпись)* |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
| Коэффициент оригинальности ВКР | | \_\_\_\_\_\_\_ % |
|  | |  |
| Проверил: | Председатель П(Ц)К № 5 Кривоносова Н.В. | |
|  | | *(Должность, ФИО)* |
|  | |  |
| 10.06.2024 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| *(дата)* | | *(подпись)* |

ОГЛАВЛЕНИЕ

[ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ 6](#_Toc160209356)

[ВВЕДЕНИЕ 7](#_Toc160209357)

[1. ПОСТАНОВКА ЗАДАЧИ 9](#_Toc160209358)

[1.1. Обзор аналогов 9](#_Toc160209359)

[1.2. SWOT-анализ 13](#_Toc160209360)

[1.3. PEST-анализ 14](#_Toc160209361)

[1.4. Формирование требований к программному продукту 15](#_Toc160209362)

[1.4.1. Бизнес требования 15](#_Toc160209363)

[1.4.2. Пользовательские требования 16](#_Toc160209364)

[1.4.3. Функциональные требования 16](#_Toc160209365)

[1.4.4. Нефункциональные требования 17](#_Toc160209366)

[1.4.5. Ограничения 17](#_Toc160209367)

[1.4.6. Требования к интерфейсам 18](#_Toc160209368)

[1.4.7. Требования к данным 18](#_Toc160209369)

[1.5. Программные средства разработки 18](#_Toc160209370)

[1.6. Аппаратные средства разработки 19](#_Toc160209371)

[2. ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ 20](#_Toc160209372)

[2.1. Архитектура системы 20](#_Toc160209373)

[2.2. Моделирование основных сценариев системы 20](#_Toc160209374)

[2.3. Проектирование графического интерфейса пользователя 29](#_Toc160209375)

[2.4. Проектирование и разработка модели данных 31](#_Toc160209376)

[3. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА 34](#_Toc160209377)

[3.1. Реализация серверной части 34](#_Toc160209378)

[3.2. Реализация клиентской части 35](#_Toc160209379)

[4. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 39](#_Toc160209380)

[4.1. История изменений 39](#_Toc160209381)

[4.2. Терминология тестирования программного обеспечения 39](#_Toc160209382)

[4.3. Стратегия тестирования 39](#_Toc160209383)

[4.4. Определение объектов тестирования 39](#_Toc160209384)

[4.5. Архитектура тестируемой системы 40](#_Toc160209385)

[4.6. Описание процесса тестирования 41](#_Toc160209386)

[5. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ 43](#_Toc160209387)

[5.1. Определение защищаемых информационных активов 43](#_Toc160209388)

[5.2. Вид и содержание персональных данных 43](#_Toc160209389)

[5.3. Модель угроз ПНД 44](#_Toc160209390)

[5.4. Возможные сценарии угроз 62](#_Toc160209391)

[5.5. Организация защиты данных в проекте 67](#_Toc160209392)

[6. ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ НА ПРОГРАММНЫЙ ПРОДУКТ 69](#_Toc160209393)

[6.1. Руководство пользователя 69](#_Toc160209394)

[6.2. Руководство администратора 69](#_Toc160209395)

[6.3. Определение качества продукта по метрикам 69](#_Toc160209396)

[7. МЕРОПРИЯТИЯ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ И ОХРАНЕ ТРУДА 70](#_Toc160209397)

[7.1. Характеристика объекта проектирования и условий его эксплуатации 70](#_Toc160209398)

[7.1.1. Технические данные проектного устройства 70](#_Toc160209399)

[7.1.2. Условия эксплуатации проектного объекта 70](#_Toc160209400)

[7.1.3. Требования к обслуживающему персоналу 71](#_Toc160209401)

[7.2. Мероприятия по технике безопасности 72](#_Toc160209402)

[7.2.1. Организационно-технические мероприятия по охране труда 72](#_Toc160209403)

[7.2.2. Требования электробезопасности при работе с ПК 72](#_Toc160209404)

[7.2.3. Требования защиты от излучения 73](#_Toc160209405)

[7.2.4. Требования защиты от шума 73](#_Toc160209406)

[7.3. Противопожарная безопасность 74](#_Toc160209407)

[7.3.1. Противопожарная безопасность при работе с персональным компьютером 74](#_Toc160209408)

[7.4. Общие требования охраны труда 75](#_Toc160209409)

[7.5. Вывод 76](#_Toc160209410)

[8. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА 77](#_Toc160209411)

[8.1. Расчёт затрат на разработку ПО 77](#_Toc160209412)

[8.1.1. Затраты на оплату (ЗОТ) 77](#_Toc160209413)

[8.2. Эксплуатационные затраты на оборудование 78](#_Toc160209414)

[8.2.1. Суммарная годовая стоимость эксплуатационных затрат 78](#_Toc160209415)

[8.2.2. Затраты на техническое обслуживание и ремонт 78](#_Toc160209416)

[8.2.3. Затраты на материалы и комплектующие 78](#_Toc160209417)

[8.2.4. Амортизационные отчисления 78](#_Toc160209418)

[8.2.5. Стоимость электроэнергии 79](#_Toc160209419)

[8.2.6. Эффективный фонд рабочего времени 79](#_Toc160209420)

[8.2.7. Цена эксплуатационных затрат 79](#_Toc160209421)

[8.3. Затраты на материалы 79](#_Toc160209422)

[8.3.1. Накладные расходы 79](#_Toc160209423)

[8.4. Себестоимость разработки программного обеспечения 80](#_Toc160209424)

[8.5. Расчет экономического эффекта разработчика и пользователя (заказчика) ПО 80](#_Toc160209425)

[8.5.1. Экономический эффект у разработчика программного обеспечения 80](#_Toc160209426)

[8.6. Вывод 81](#_Toc160209427)

[ЗАКЛЮЧЕНИЕ 82](#_Toc160209428)

[СПИСОК ИСТОЧНИКОВ ИНФОРМАЦИИ 83](#_Toc160209429)

[ПРИЛОЖЕНИЕ 1. 85](#_Toc160209430)

[ПРИЛОЖЕНИЕ 2 99](#_Toc160209431)

[ПРИЛОЖЕНИЕ 3 100](#_Toc160209432)

[ПРИЛОЖЕНИЕ 4 101](#_Toc160209433)

**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.

**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, Miro, 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 электронные документообороты. 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 конкурентноспособным на рынке. Быстрая, кроссплатформенная и простая в работе система коммуникация выполнит основные задачи, которые могут понадобится предприятию. Из дополнительных требований могут быть наличие чатов, каналов, видеозвонков, 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.

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 С# 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 аудиозвонки.

Внешний вид VK Teams представлен на рис. 1.1.1.

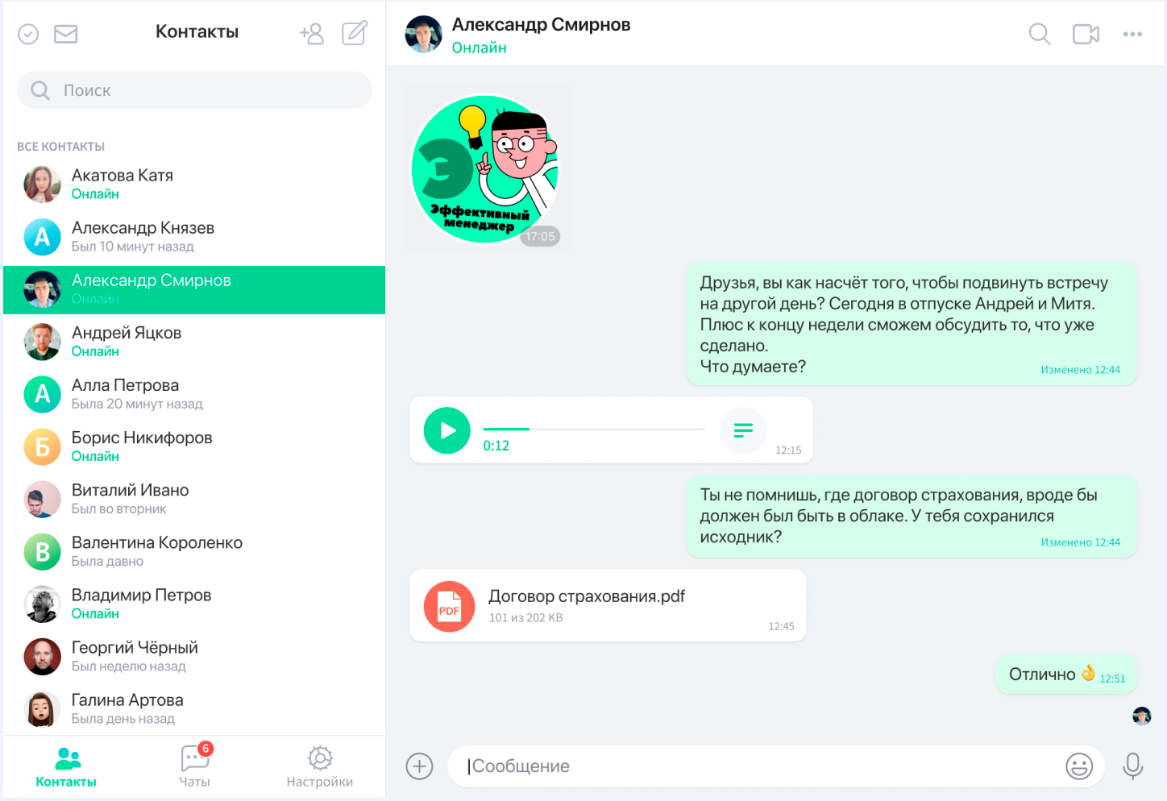


Рисунок 1.1.1. Внешний вид VK Teams

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).

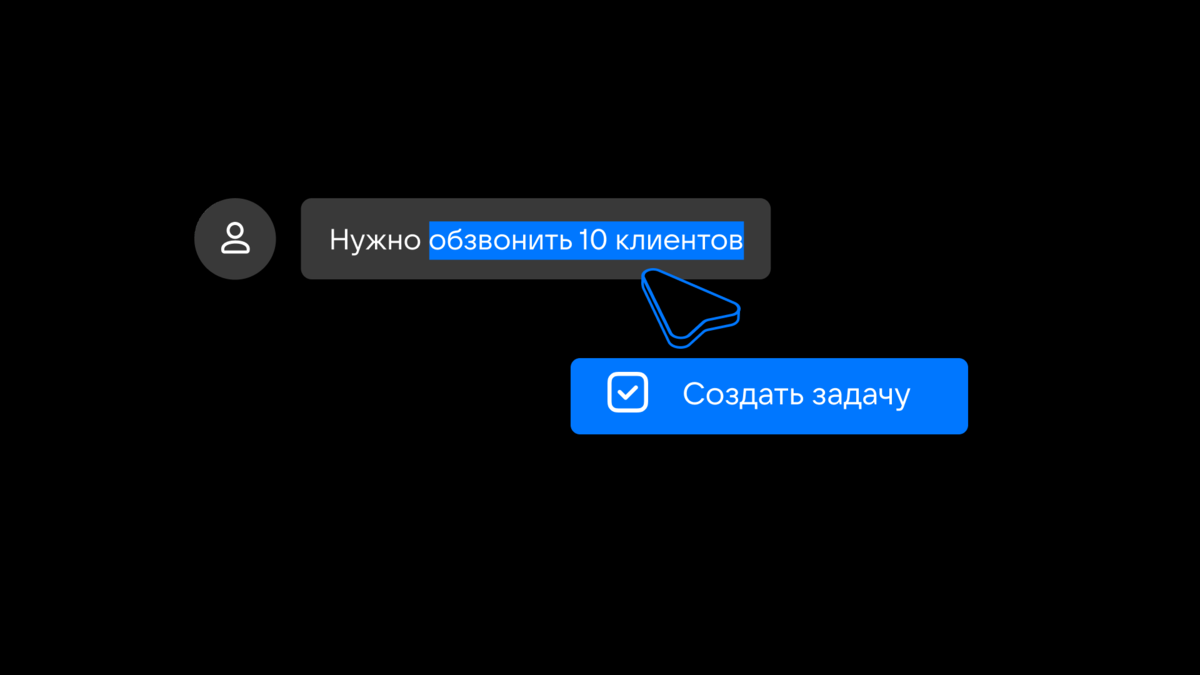


Рисунок 1.1.2. Добавление задачи в таск-менеджер из переписки

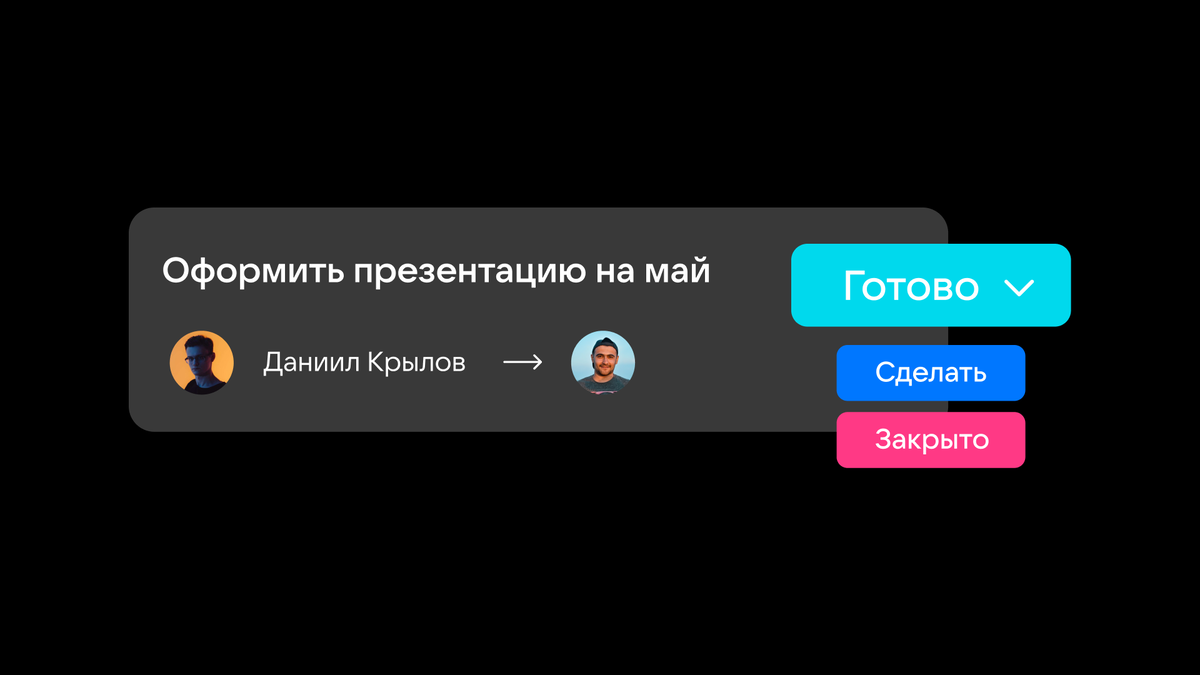


Рисунок 1.1.3

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).

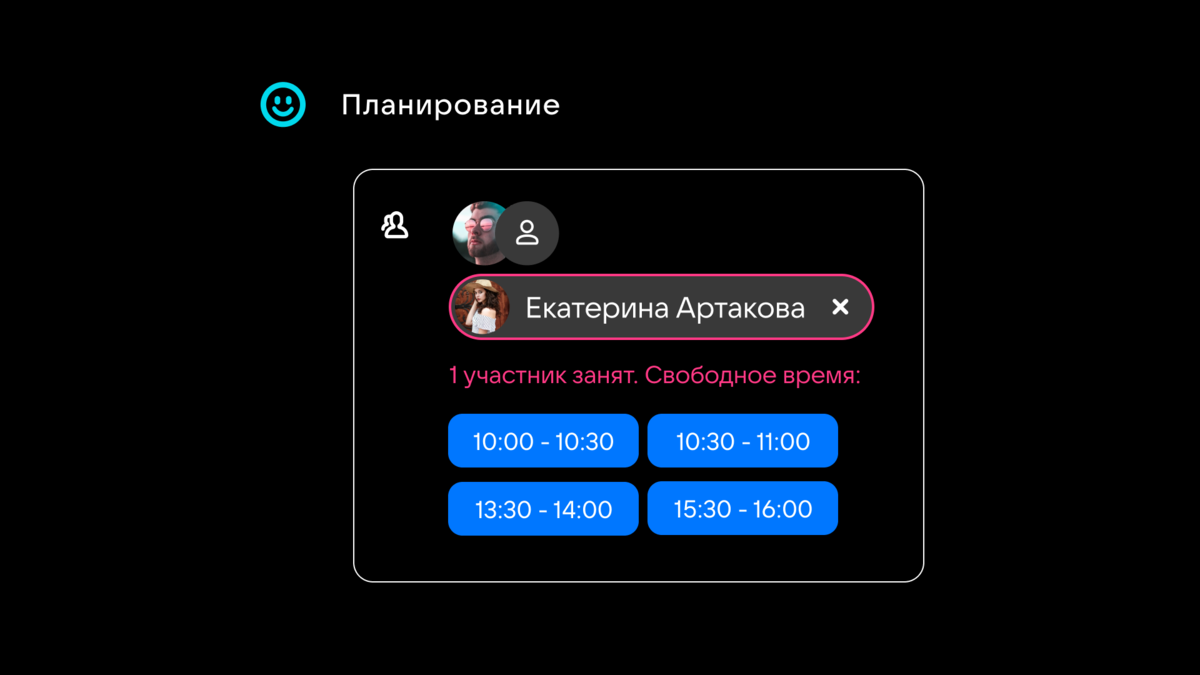


Рисунок 1.1.4

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 рисунке 1.1.5.

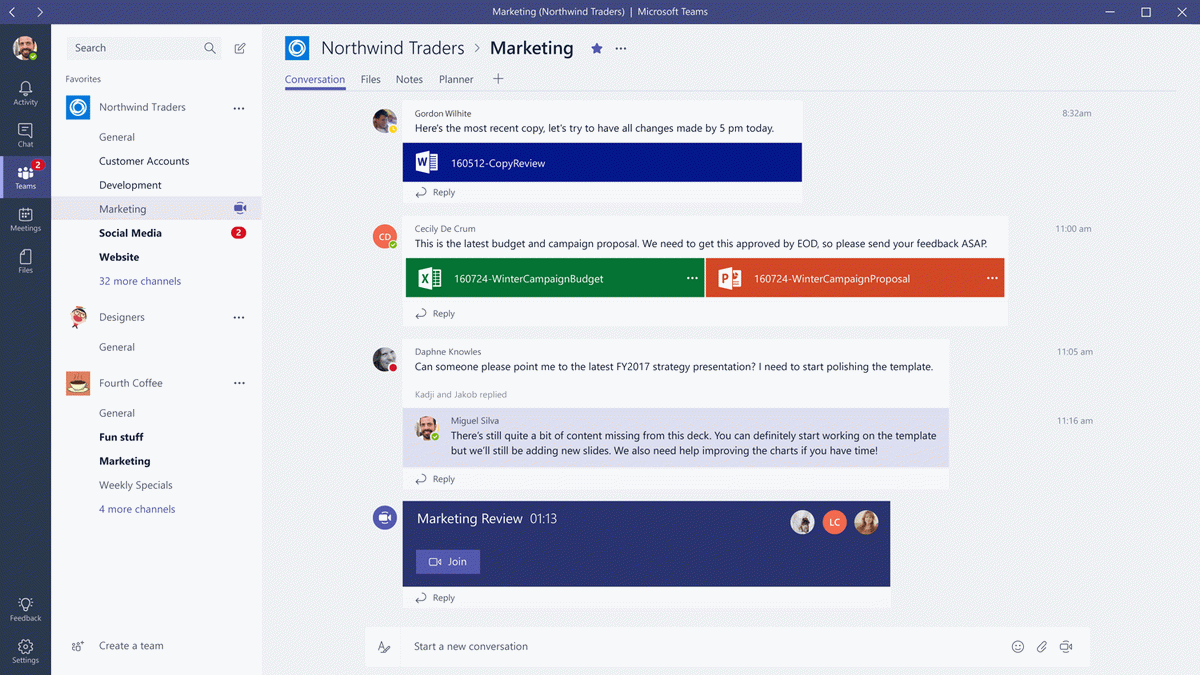


Рисунок 1.1.5. Внешний вид Microsoft Teams

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).

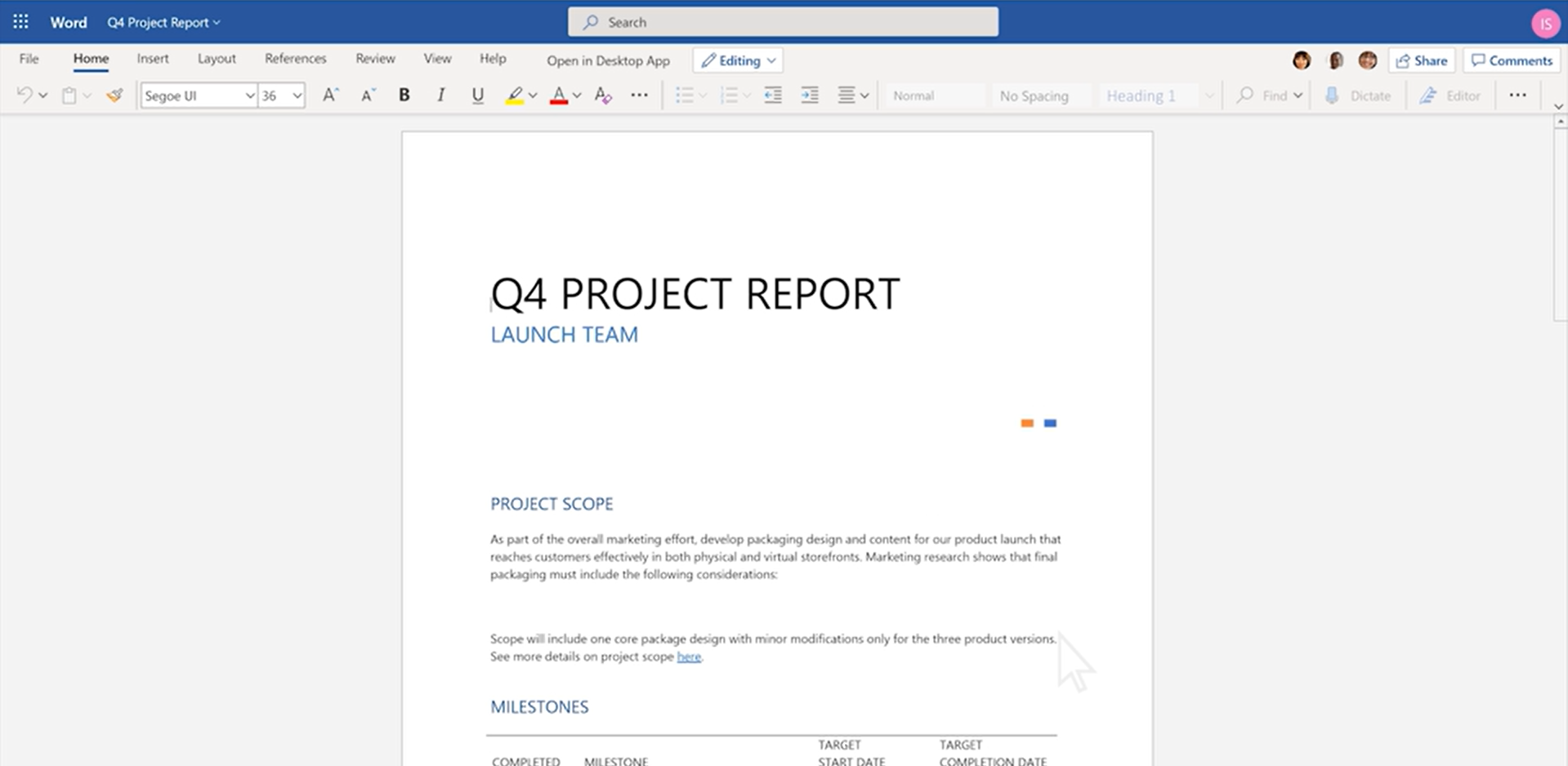


Рисунок 1.1.6

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.

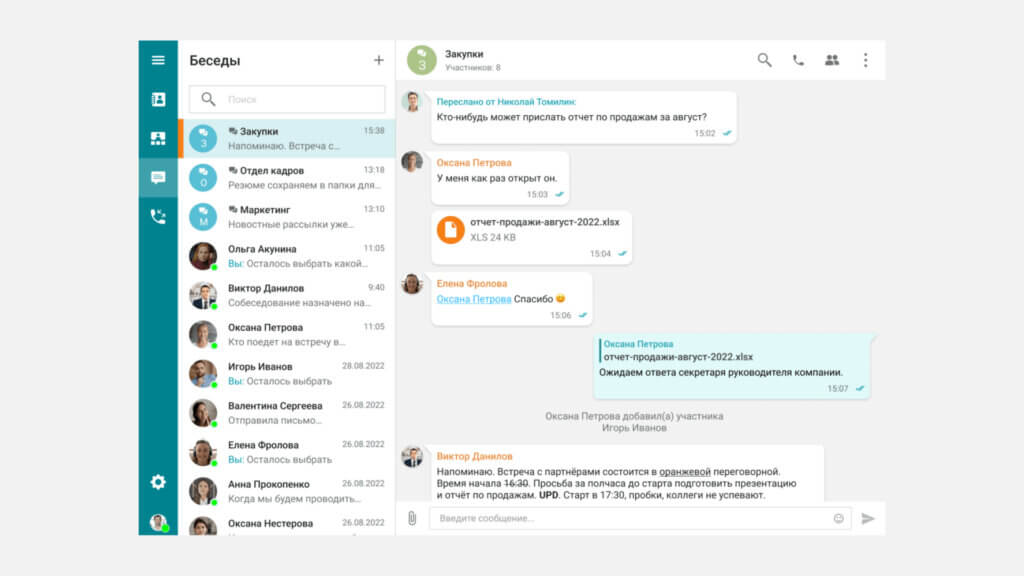


Рисунок 1.1.7. Окно с чатами в TrueConf

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.

**1.2. SWOT-анализ**

В ходе выполнения курсового проекта для определения функциональных требований был проведен SWOT-анализ (таблица 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.Многопользовательский режим  2.Современные технологии приложения  3.Бесплатное пользование  4.Доступный на разных платформах  5.Групповые чаты  6.Хранилище для файлов | За счет использования многопользовательского режима программа эффективно справляется с потоком новых пользователей, успешно проходя цифровую трансформацию. Также она поддерживает хранилище различный объемов информации, введение групповых чатов, 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 групповых чатов, 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. |

**1.3. PEST-анализ**

В ходе выполнения курсового проекта для определения функциональных требований был проведен PEST-анализ (таблица 2):

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 выгодна. |
| 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. |

**1.4. Формирование требований к программному продукту**

**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. Интеграция с другими инструментами: мессенджер должен позволять интеграцию с другими корпоративными инструментами, такими как электронная почта, календари, проектные системы и CRM-системы. Это поможет улучшить производительность, сократить время на обмен информацией и избежать дублирования данных.

3. Функциональность: мессенджер должен предоставлять широкий набор функций, таких как обмен сообщениями, голосовые и видео вызовы, возможность создания групповых чатов и каналов, возможность обмена файлами и документами, функция поиска и архивирования сообщений.

4. Географическая доступность: мессенджер должен быть доступным для использования в любой точке мира и на любых устройствах - компьютерах, смартфонах и планшетах.

5. Удобный пользовательский интерфейс: мессенджер должен иметь простой и интуитивно понятный интерфейс, приятный дизайн и настраиваемые настройки, чтобы каждый сотрудник мог настроить его под свои потребности.

6. Функциональная мобильная версия: мессенджер должен иметь полноценную мобильную версию, чтобы сотрудники могли общаться и работать вне офиса.

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

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

9. 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 с возможностью форматирования и вложениями, чтобы пользователи могли коммуницировать и передавать информацию в удобной форме.

3. Голосовые и видео вызовы: мессенджер должен обеспечивать возможность проводить голосовые и видео вызовы для более продуктивного взаимодействия и обсуждения вопросов с коллегами.

4. Групповые чаты и каналы: мессенджер должен поддерживать возможность создания групповых чатов и каналов для организации работы в команде, обмена информацией и совместной работы над проектами.

5. Обмен файлами и документами: мессенджер должен позволять пользователям обмениваться файлами и документами различных форматов, чтобы упростить совместное редактирование и обмен материалами.

6. Уведомления: мессенджер должен предоставлять возможность получать уведомления о новых сообщениях и других активностях, чтобы пользователи могли быть в курсе событий и отреагировать своевременно.

8. Архивирование и поиск сообщений: мессенджер должен хранить и архивировать сообщения, чтобы пользователи могли легко находить нужную информацию в случае необходимости.

**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, двухфакторной аутентификации и др.).

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

3. Голосовые и видео вызовы: мессенджер должен обеспечивать возможность проводить голосовые и видео вызовы между пользователями с использованием оптимального вызова (прямого или через сервер).

4. Групповые чаты и каналы: мессенджер должен поддерживать создание и управление групповыми чатами и каналами, где пользователи могут общаться и обмениваться информацией с несколькими участниками сразу.

5. Обмен файлами и документами: мессенджер должен позволять пользователям отправлять и получать файлы и документы различных форматов, поддерживать возможность просмотра и скачивания файлов.

6. Уведомления: мессенджер должен предоставлять уведомления о новых сообщениях, голосовых и видео вызовах, изменениях в групповых чатах и других активностях.

7. Архивирование и поиск сообщений: мессенджер должен сохранять и архивировать сообщения, обеспечивать возможность поиска по ключевым словам, датам и отправителям для облегчения поиска и доступа к информации.

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

9. 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, шифрование информации, защиту от вредоносного программного обеспечения и анонимности пользователей.

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

3. Интеграция: мессенджер должен обеспечивать легкую интеграцию с другими существующими системами и инструментами, такими как почта, календари, CRM и др., для удобства и эффективности коммуникации.

4. 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.

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

6. Совместимость: мессенджер должен быть совместим с широким спектром устройств и операционных систем, чтобы пользователи могли использовать его на своих предпочтительных платформах и устройствах.

7. Локализация: мессенджер должен поддерживать возможность локализации для адаптации интерфейса и коммуникации на разных языках, соответствуя потребностям пользователей из разных стран и регионов.

**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.

**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. Резервное копирование: мессенджер должен иметь механизм резервного копирования данных, чтобы защитить информацию от потери или повреждения и обеспечить возможность восстановления.

4. 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. Архивирование и поиск данных: мессенджер должен предоставлять возможность архивирования и поиска сообщений и файлов, чтобы пользователи могли быстро находить нужные данные по ключевым словам, датам и участникам.

**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.

**2. ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ**

**2.1. Архитектура системы**

В ходе проектирования вариантов использования был выбран инструмент Visual Paradigm 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 отношении к программной системе сущность.

Прецедент – вариант использования, функционал программного продукта.

Виды ассоциаций между компонентами UseCase:

- прямая ассоциация;

- наследование – возможно только между Актёр, 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.

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

- ввод фамилии;

- ввод электронной почты;

- ввод имени;

- ввод пароля.

Прецедент «Ввод электронной почты» включает:

- ввод кода подтверждения.

Прецедент «Ввод кода подтверждения» включает:

- подтверждение кода.

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;

- установка фото профиля.

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

- отправить контакта;

- отправить аудио;

- отправить текст;

- отправить фото/видео;

- отправить локацию.

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

- удалить;

- получение сообщения;

- переслать;

- ответить.

Расширение варианта использования «Сохранить в архив» имеют следующие прецеденты такие, как:

- отправить аудио;

- отправить текст;

- отправить фото/видео;

- отправить локацию.

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

- выключить уведомления чата;

- закрепить чат;

- создать чат;

- архивировать чат;

- удалить чат.

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

- отправка кода подтверждения;

- создание профиля пользователя.

Прецедент «Создание профиля пользователя» включает:

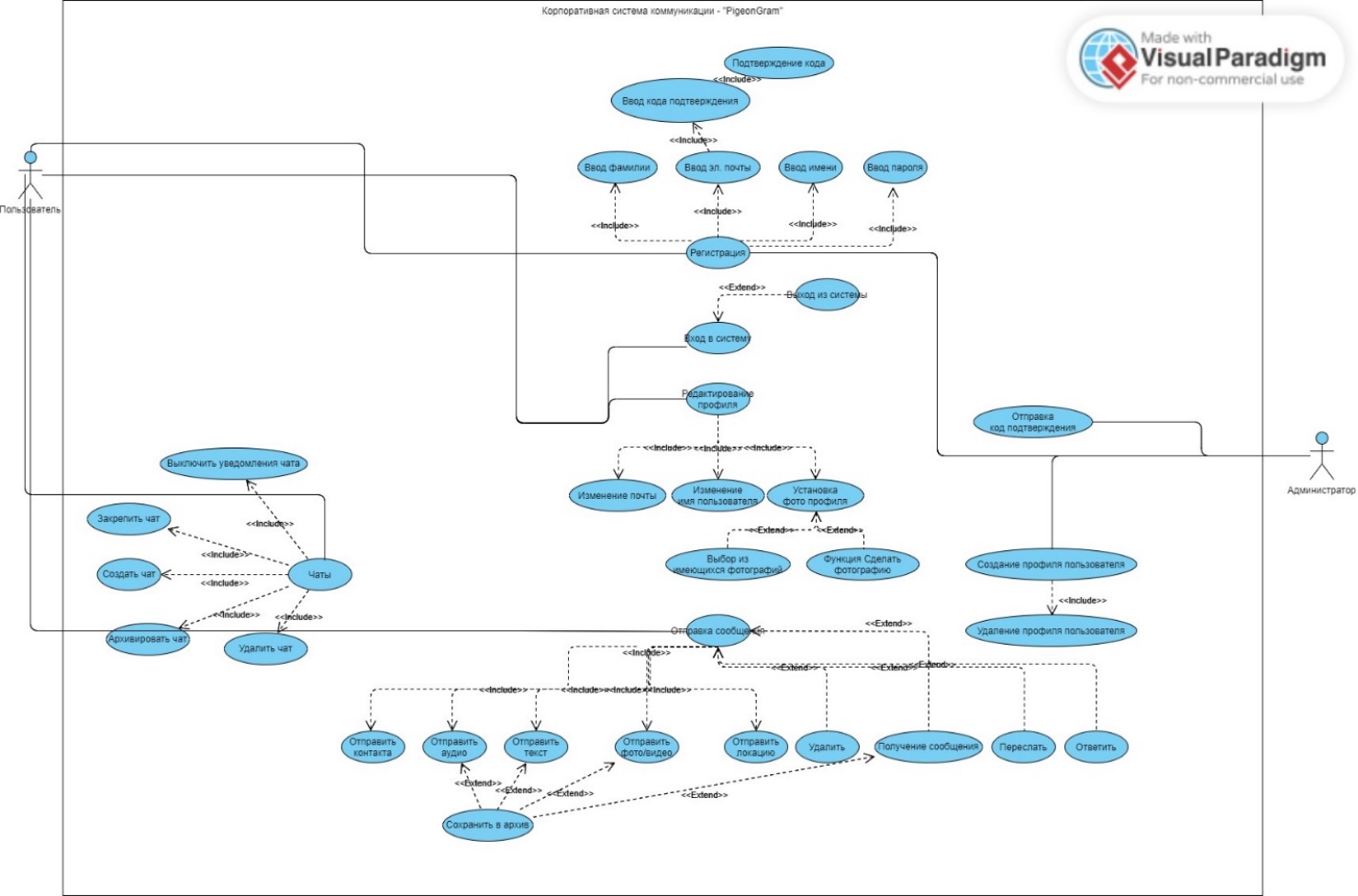
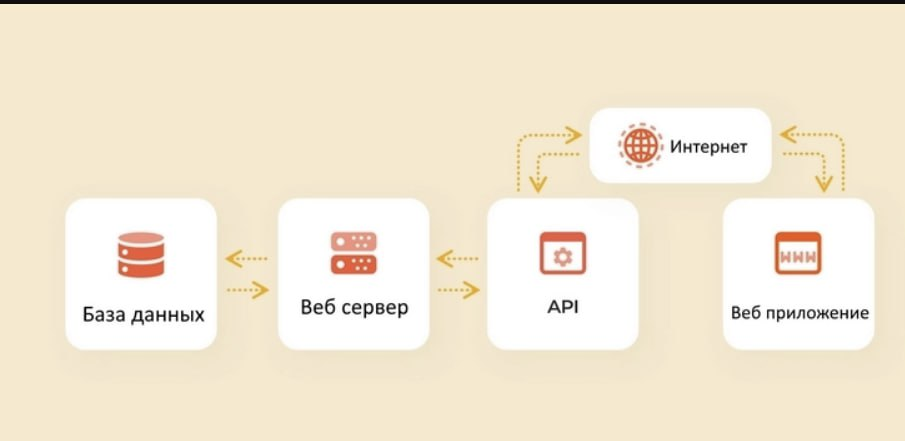
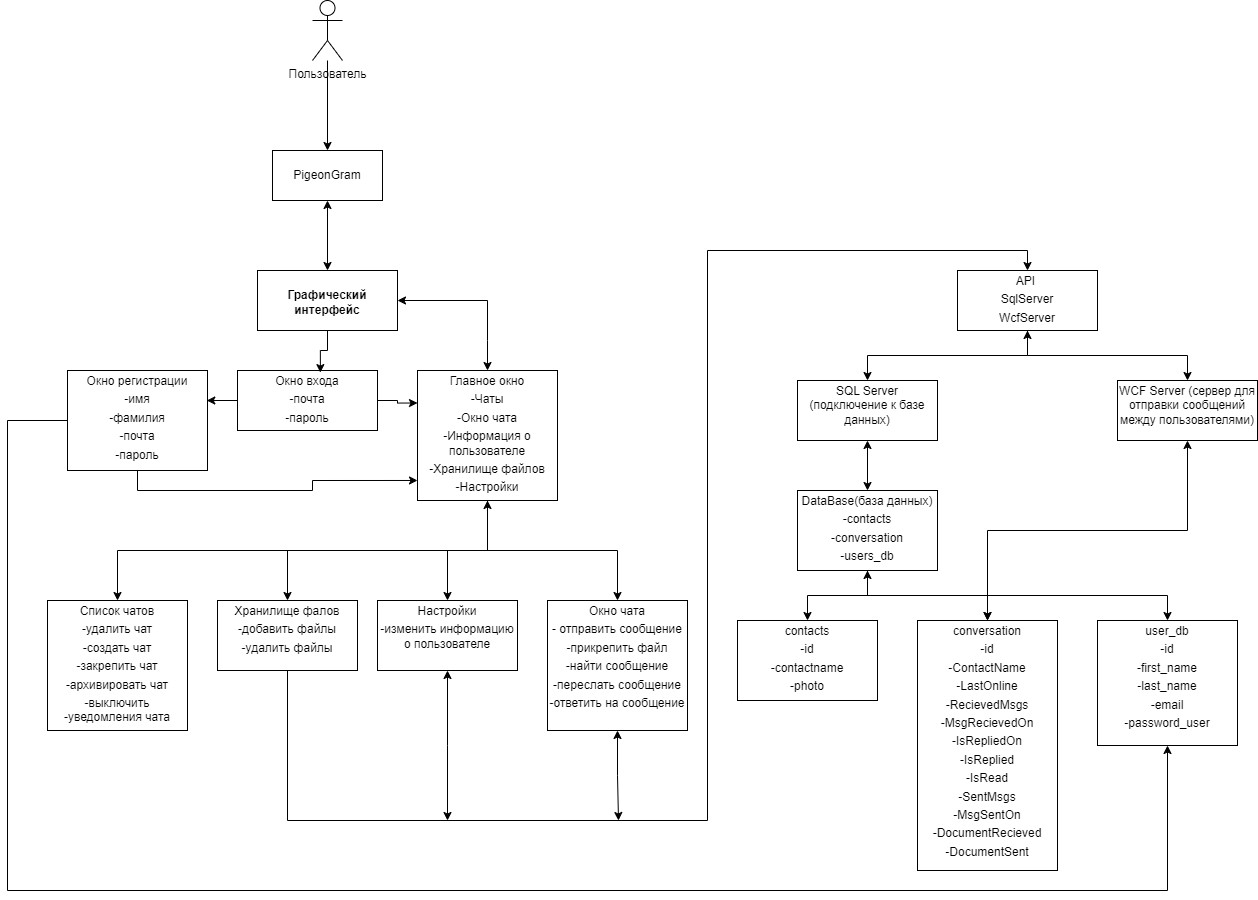
- удаление профиля пользователя.

Рисунок 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 базовую архитектуру клиент-серверного приложения (рис. 2.1.2).

*Рисунок 2.1.2. Архитектура корпоративного мессенджера*

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

*Рисунок 2.1.3. Архитектура корпоративного мессенджера*

Проект состоит из следующих компонентов:

* Пользовательский интерфейс;
* Локальная база данных;
* Общее хранилище;
* Менеджер профиля;
* Хранилище ключей шифрования;
* Сетевой клиент;
* Децентрализованная распределительная система.

Пользователь взаимодействует с мессенджером через пользовательский интерфейс, который взаимодействует с общим хранилищем, локальной базой данных, менеджером профиля и децентрализованной распределительной системой.

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

Через децентрализированную распределительную систему программа идентифицирует пользователя, посылая адрес узла сетевому клиенту.

Сетевой клиент посылает зашифрованные сообщения менеджеру профиля, который использует хранилище ключей шифрования для шифровки сообщений.

**2.2. Моделирование основных сценариев системы**

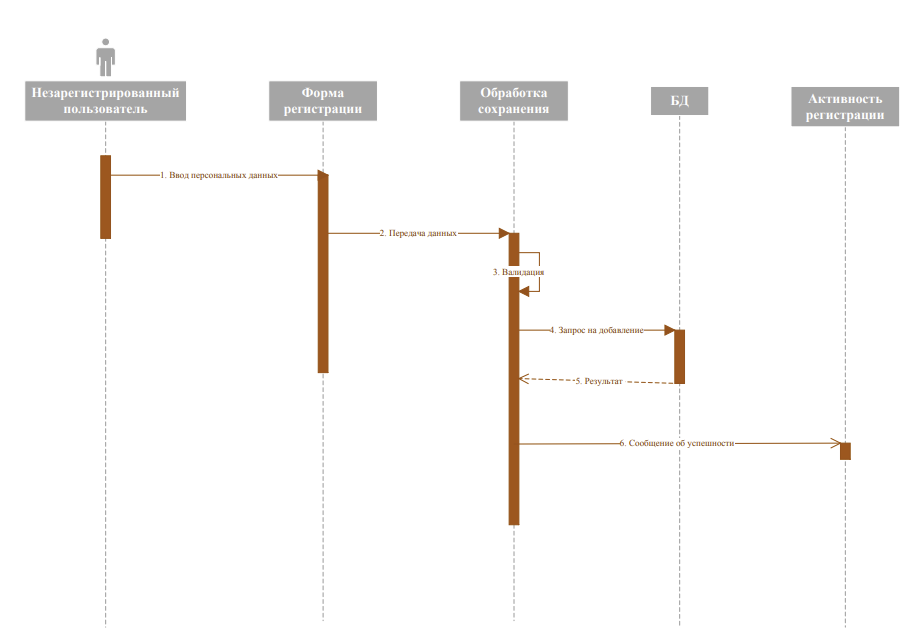
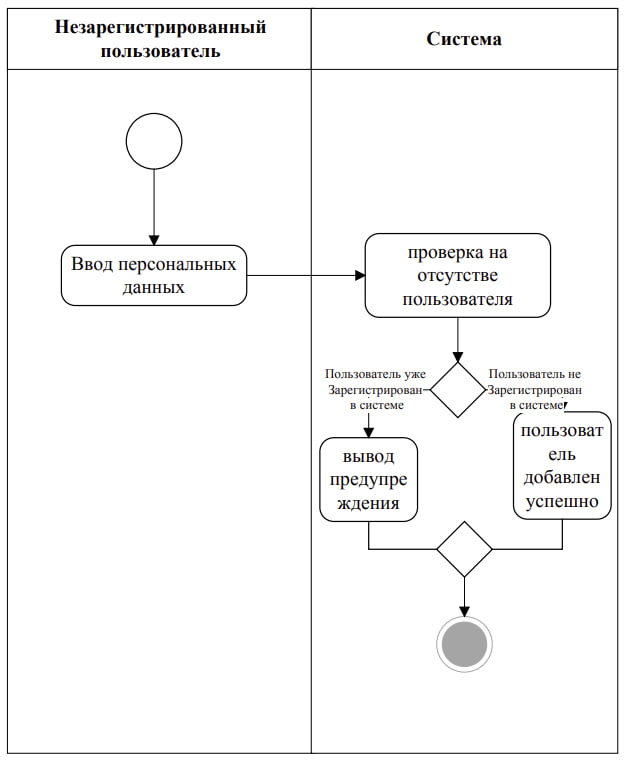
*Рисунок 2.2.1. Диаграмма деятельности. Регистрация пользователя*

Рисунок 2.2.2. Диаграмма последовательности. Регистрация пользователя

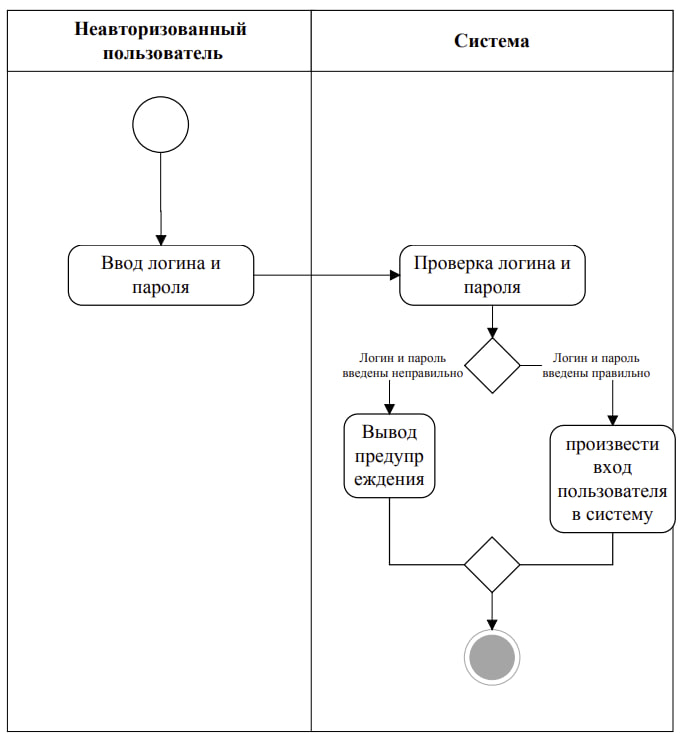
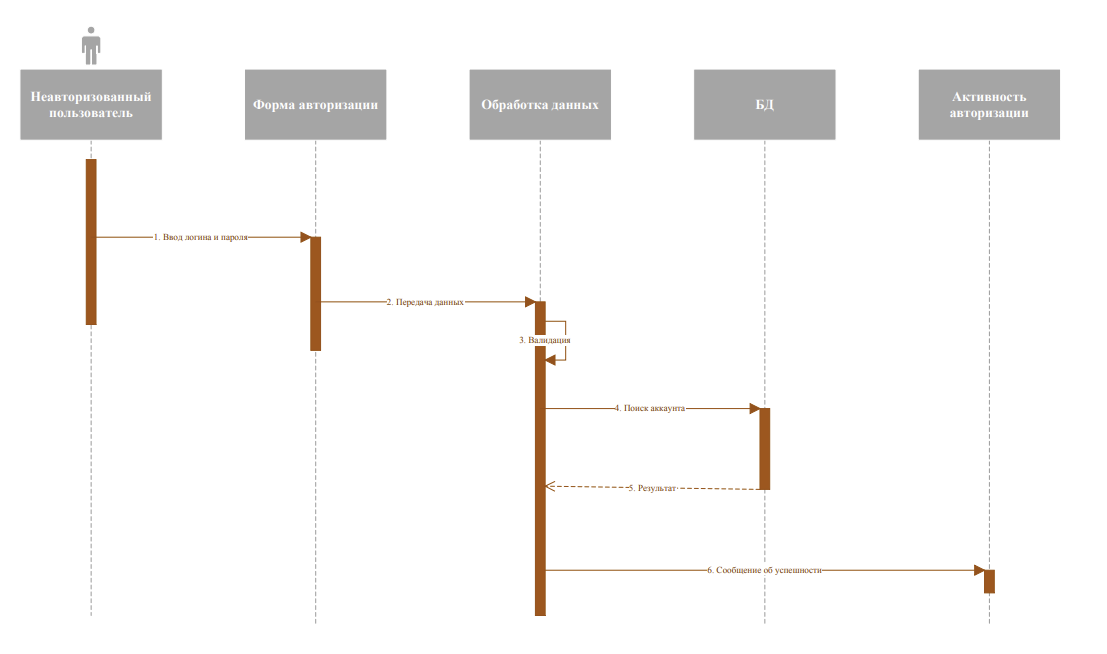
На рис. 2.2.1-2.2.2 представлены диаграммы последовательности и деятельности на прецедент «Регистрация пользователя». Регистрация проходит следующих образом: незарегистрированный пользователь вводит персональные данные в форму регистрации. Данные передаются на сервер, где происходит валидация. После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

Рисунок 2.2.3. Диаграмма деятельности. Авторизация пользователя

*Рисунок 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 следующих 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.

Рисунок 2.2.5. Диаграмма деятельности. Отправка сообщений

Рисунок 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.

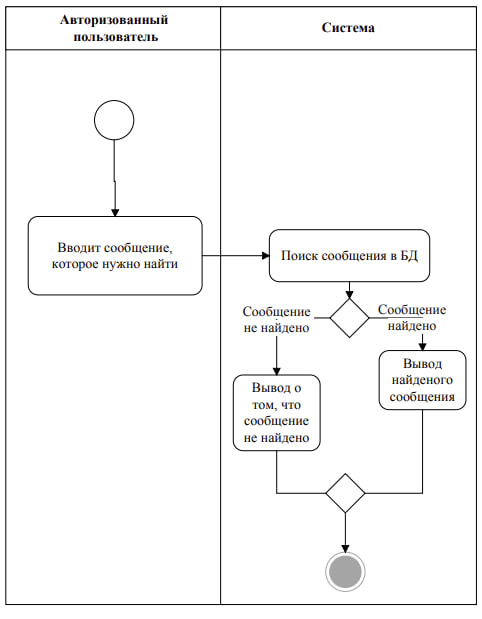


Рисунок 2.2.7. Диаграмма деятельности. Поиск сообщения

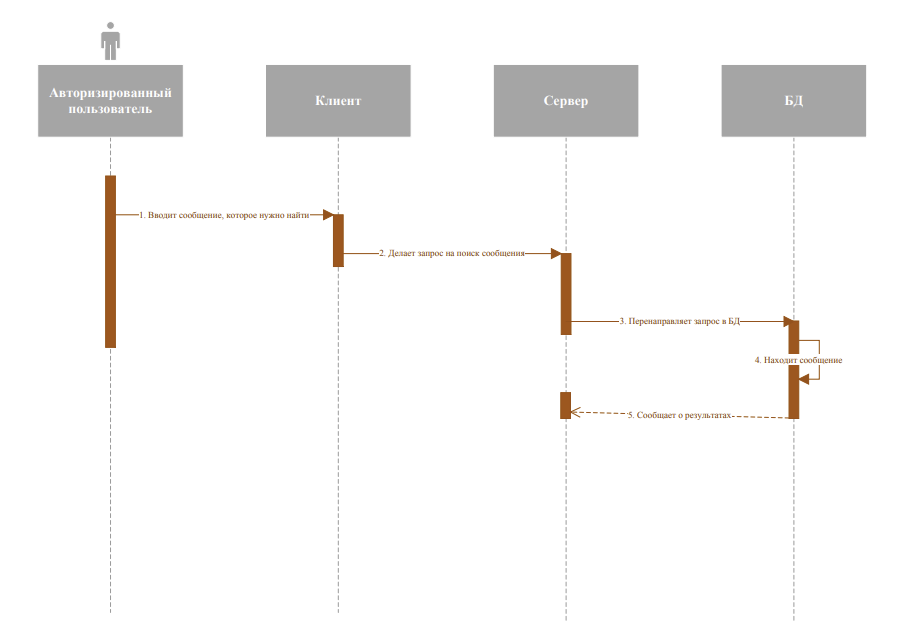


Рисунок 2.2.8. Диаграмма последовательности. Поиск сообщения

На рис. 2.2.7-2.2.8 представлены диаграммы деятельности и последовательности на прецедент «Поиск сообщения». Авторизованный пользователь вводит через интерфейс клиента сообщение, которое ему необходимо найти. Клиент делает серверу запрос на поиск сообщения, сервер перенаправляет запрос к БД. БД ищет сообщение и сообщает о результате серверу.

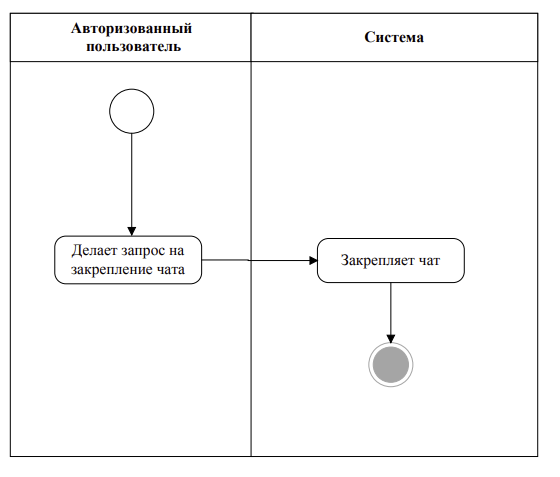


Рисунок 2.2.9. Диаграмма деятельности. Закрепление чата

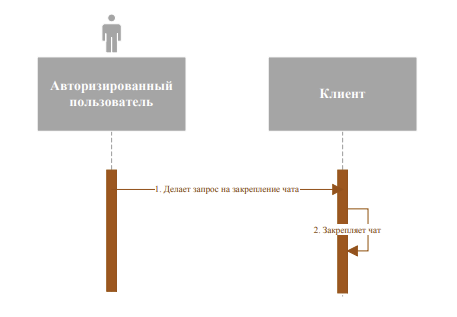


Рисунок 2.2.10. Диаграмма последовательности. Закрепление чата

На рис. 2.2.9-2.2.10 представлены диаграммы деятельности и последовательности на прецедент «закрепление чата». Пользователь через интерфейс клиента делает запрос на закрепление чата, клиент закрепляет чат.

**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 вобще 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 рисунках ниже представлены интерфейс и основные функции корпоративного мессенджера PigeonGram.

На рис. 2.3.1 представлен непосредственно сам интерфейс приложения. Окно содержит список чатов, на которые может нажать пользователь. Посередине изображена картинка «Свяжись с коллегой». Также присутствуют значки закрытия, сворачивания и расширения окна. Все приложение выполнено в оттенках синего и голубого цветов.

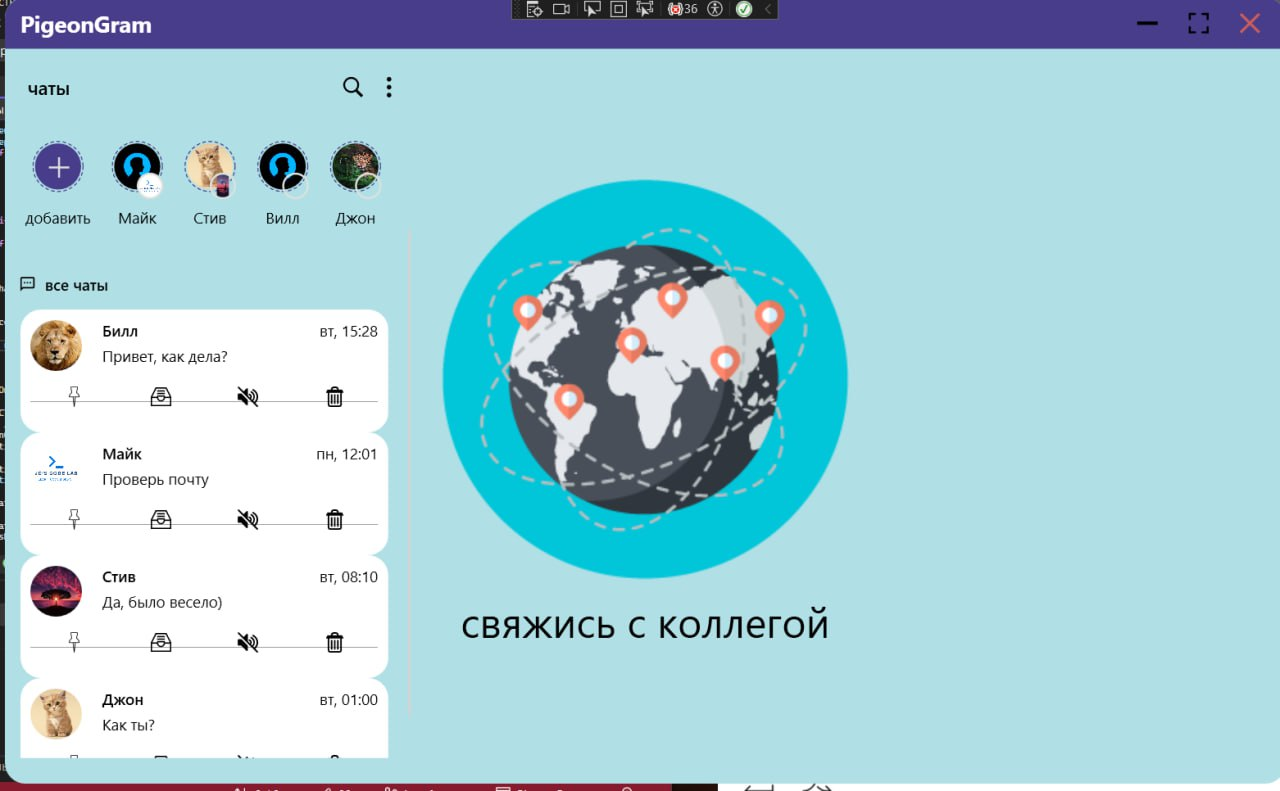


Рисунок 2.3.1. Интерфейс приложения

На рис. 2.3.2 показаны основные функции приложения, а именно: создание новой группы, начать видеозвонок, избранные сообщения и настройки.

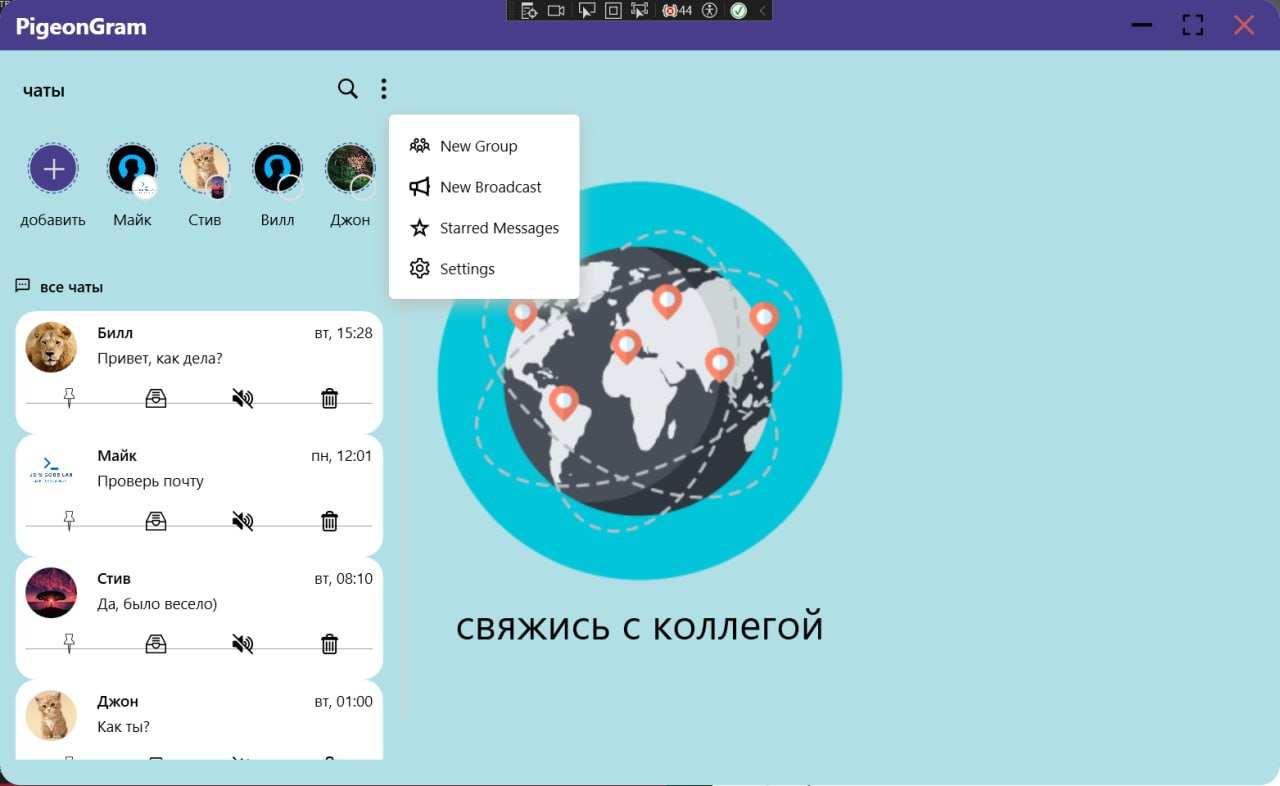


Рисунок 2.3.2. Дополнительные функции

На рис. 2.3.3 показана реализация функции закрепления сообщения.

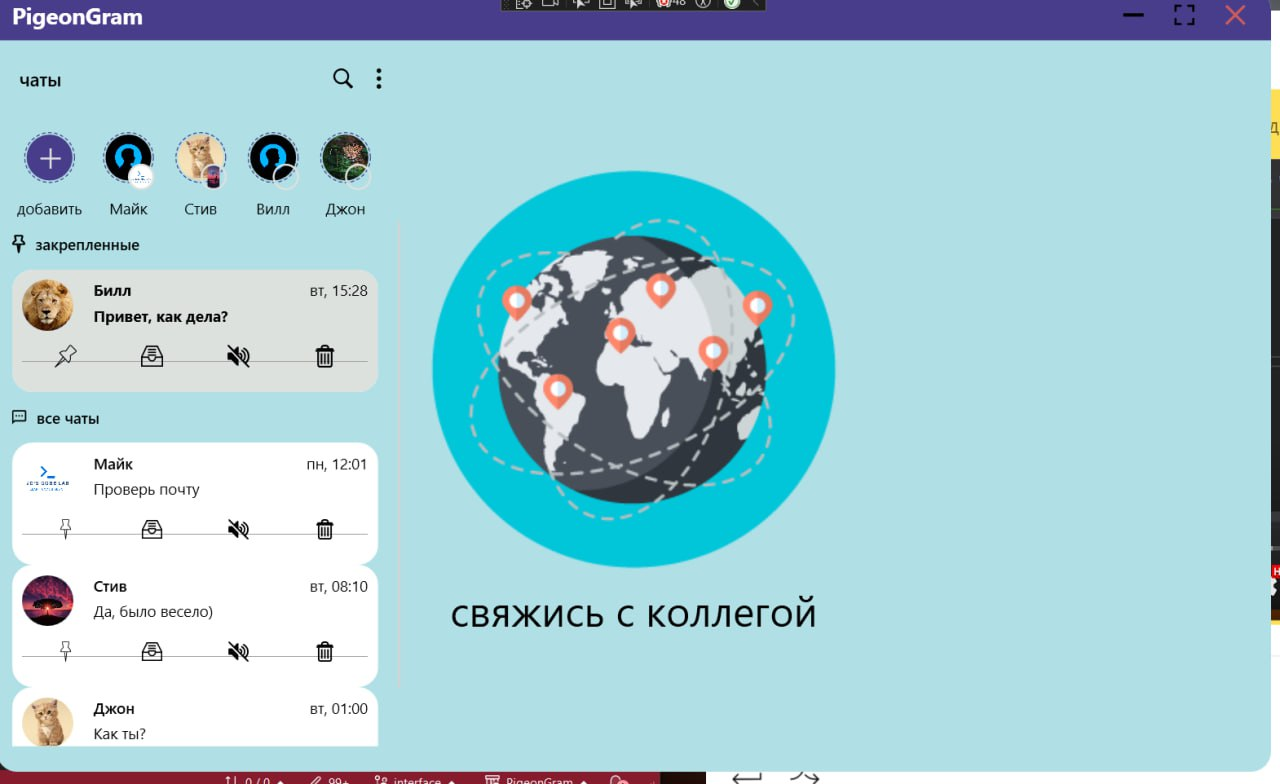


Рисунок 2.3.3. Закрепленные сообщения

На рис. 2.3.4 показана реализация функции архивирования сообщения.

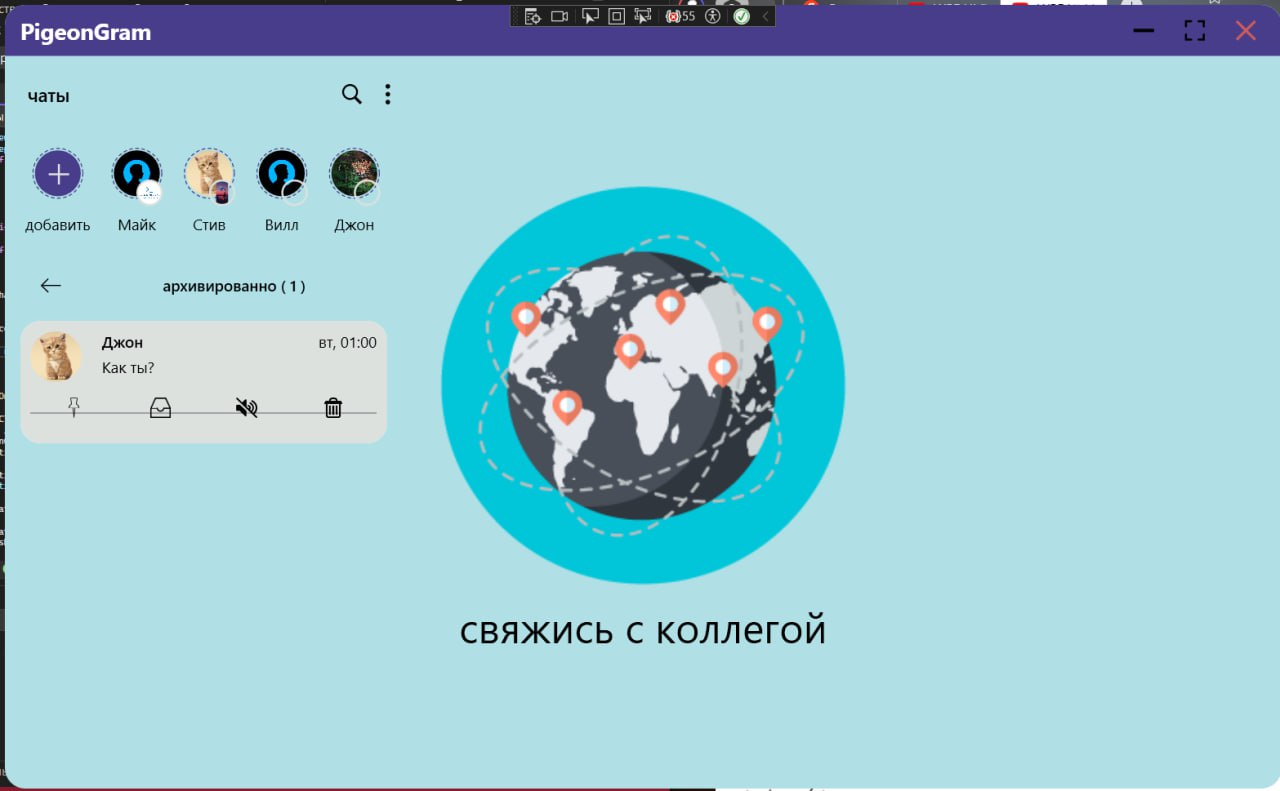


Рисунок 2.3.4. Архивированные сообщения

На рис. 2.3.5. 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.

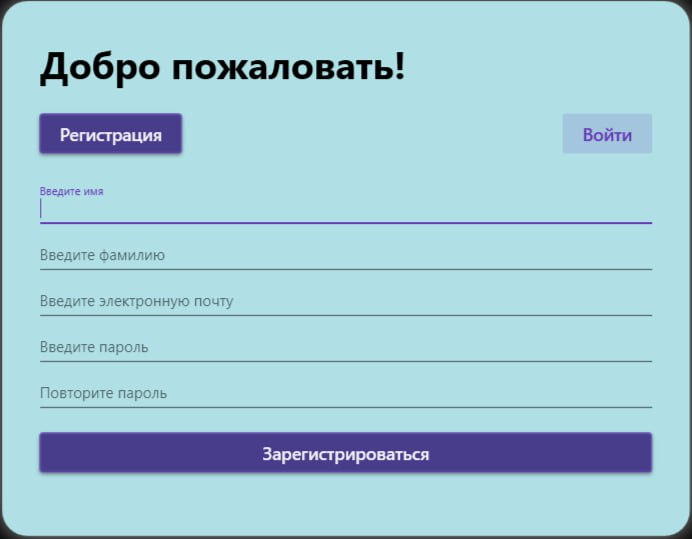


Рисунок 2.3.5. Окно регистрации

На рис. 2.3.6 показана реализация окно 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.

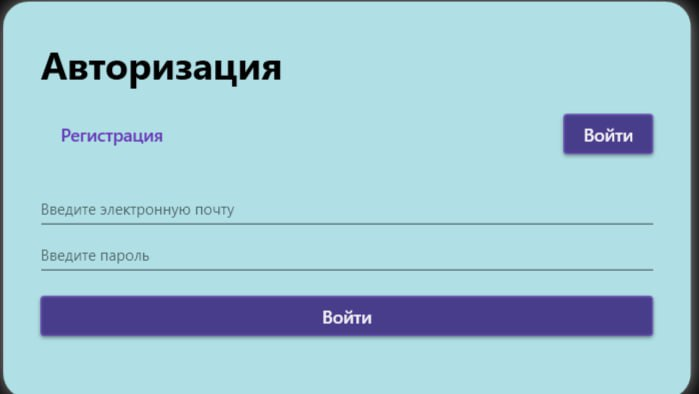


Рисунок 2.3.6. Окно входа

**2.4. Проектирование и разработка модели данных**

В предметной области можно выделить следующие сущности:

* Сущность «Контакты»;
* Сущность «Чаты».

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:

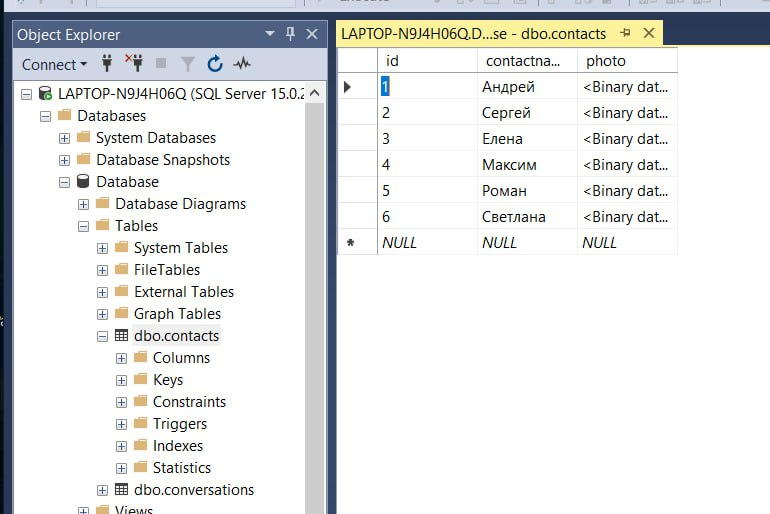


Рисунок 2.4.1. Таблица «Contacts» – Контакты

В данной таблице хранятся данные о контактах – Id контакта (id), имя контакта (ContactName) и фото контакта (Photo). Ни одно поле не может иметь нулевого значения. Id контакта является первичным ключом.

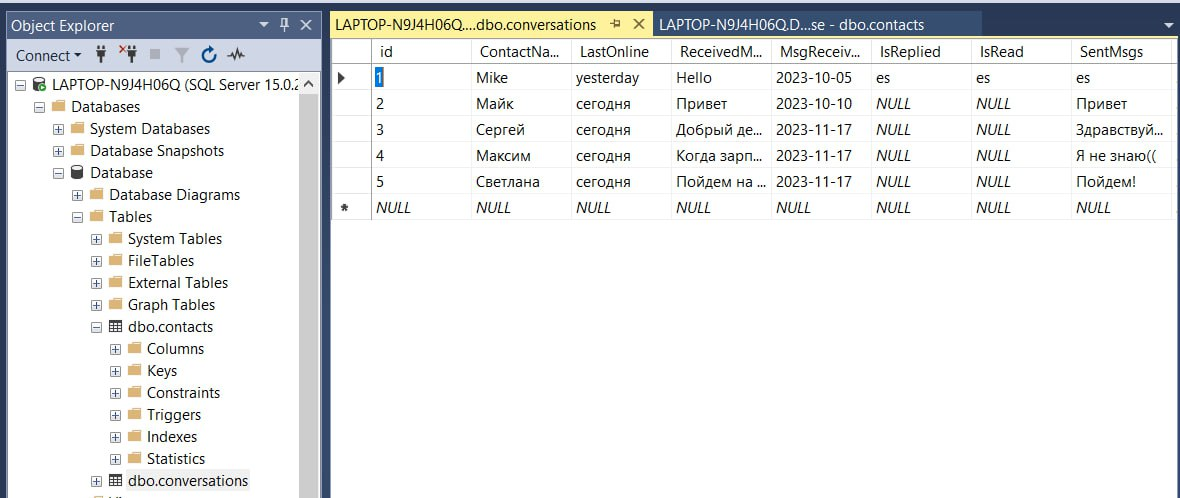


Рисунок 2.4.2. Таблица «Conversation» – Чаты

В данной таблице хранятся данные о чатах – id, имя контакта (ContactName), последнее посещение (LastOnline), текст полученного сообщения (ReceivedMsg), время, когда сообщение было получено (MsgReceivedTime), информация об ответе на сообщение (IsReplied), информация о прочтении сообщения (IsRead). Id является первичным ключом. Все поля, кроме информации об ответе на сообщение и информации о прочтении сообщения не могут иметь нулевого значения.

**3.** **РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА**

Поскольку программный продукт является клиент-серверным приложением, он состоит из двух частей – сервера (который, в свою очередь, состоит из хоста и сервиса) и клиента. Ниже представлены описания и листинги кода реализации функций.

**3.1. Реализация серверной части**

Функция «Connect» добавляет нового пользователя и принимает следующие значения: ID, которое присваивает новое значение id каждому пользователю, Name, которое хранит поступившее имя пользователя, и operationContext, значение для которого поступают вместе с соединением пользователя. После соединения, все пользователи получают уведомление о том, что был добавлен новый юзер. Позже будет реализовано добавление пользователя в БД.

public int Connect(string name)

{

ServerUser user = new ServerUser()

{

ID = nextId,

Name = name,

operationContext = OperationContext.Current

};

nextId++;

SendMsg(": " + user.Name + " was added", 0);

users.Add(user);

return user.ID;

}

Функция «SendMsg» отправляет сообщения пользователю. Функция принимает значения msg и id; перебирает всех пользователей из списка и формирует сообщение из параметра msg и текущего времени. После функция перебирает всех пользователей и проверяет, являются ли они ненулевыми. После отправляет сообщение пользователям.

public void SendMsg(string msg, int id)

{

foreach (var item in users)

{

string answer = DateTime.Now.ToShortTimeString();

var user = users.FirstOrDefault(i => i.ID == id);

if (user != null)

{

answer += ": " + user.Name + " ";

}

answer += msg;

item.operationContext.GetCallbackChannel<IServerChatCallback>().MsgCallback(answer);

}

}

Также в серверной части присутствует хост, задача которого заключается в том, чтобы распределять данные сервиса.

static void Main(string[] args)

{

using (var host = new ServiceHost(typeof(wcf\_chat.ServiceChat)))

{

host.Open();

Console.WriteLine("host is started!");

Console.ReadLine();

}

}

**3.2. Реализация клиентской части**

Функция отправки сообщения.

В представленной функции элементы конвертируются и передаются в момент, когда сообщение отправляется, если текстбокс не пустой. Благодаря методу происходит доваление сообщения в лист диалога. И текстбокс становится пустым, после отправки сообщения.

public void SendMessage()

{

//send message only when the textbox is not empty

if(!string.IsNullOrEmpty(MessageText))

{

var conversation = new ChatConversation()

{

ReceivedMessage = MessageToReplyText,

SentMessage = MessageText,

MsgSentOn = DateTime.Now.ToString("MMM dd, hh:mm tt"),

MessageContainsReply = IsThisAReplyMessage

};

//add msg to conversation list

FilteredConversations.Add(conversation);

Conversations.Add(conversation);

//clear message properties and textbox when message is sent

MessageText=string.Empty;

IsThisAReplyMessage = false;

MessageToReplyText= string.Empty;

UpdateChatAndMoveUp(Chats, conversation);

UpdateChatAndMoveUp(PinnedChats, conversation);

UpdateChatAndMoveUp(FilteredChats, conversation);

UpdateChatAndMoveUp(FilteredPinnedChats, conversation);

UpdateChatAndMoveUp(ArchiveChats, conversation);

//update

OnPropertyChanged("FilteredConversations");

OnPropertyChanged("Conversations");

OnPropertyChanged("MessageText");

OnPropertyChanged("IsThisAReplyMessage");

OnPropertyChanged("MessageToReplyText");

}

}

На рис. 3.2.1 представлен результат работы функции отправки сообщений.

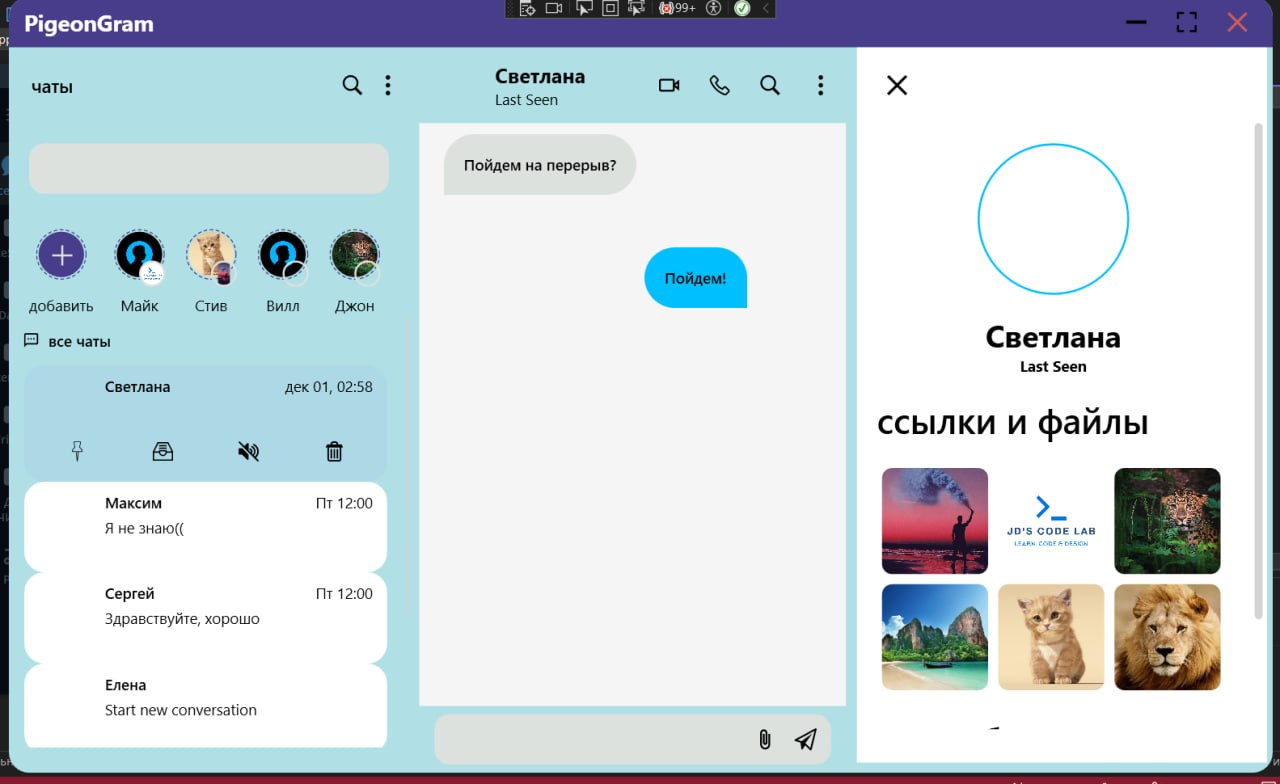


Рисунок 3.2.1. Результат работы функции отправки сообщения

Во время функции закрепления чата считывается значение mPinnedChats, после чего реализуется функционал. От значения будет зависеть закрепится чат или - нет.

public ObservableCollection<ChatListData> PinnedChats

{

get => mPinnedChats;

set

{

if (mPinnedChats == value) return;

mPinnedChats = value;

FilteredPinnedChats = new ObservableCollection<ChatListData>(mPinnedChats);

OnPropertyChanged("PinnedChats");

OnPropertyChanged("FilteredPinnedChats");

}

}

На рис. 3.2.2 представлен результат работы функции закрепления чата.



Рисунок 3.2.2. Результат работы функции закрепления чата

Во время функции архивирования чата проверяется нулевое значение, после чего выполняется команда по архивации чата. При выборе архивации, чат архивируется.

public ICommand ArchiveChatCommand => \_archiveChatCommand ?? new RelayCommand(parameter =>

{

if (parameter is ChatListData v)

{

if (!ArchiveChats.Contains(v))

{

//add chat in archive list

ArchiveChats.Add(v);

v.ChatIsArchive = true;

v.ChatIsPinned= false;

//remove chat from pinned and unpinned chat list

Chats.Remove(v);

FilteredChats.Remove(v);

PinnedChats.Remove(v);

FilteredPinnedChats.Remove(v);

//update list

OnPropertyChanged("Chats");

OnPropertyChanged("FilteredChats");

OnPropertyChanged("PinnedChats");

OnPropertyChanged("FilteredPinnedChats");

OnPropertyChanged("ArchiveChats");

}

}

});

На рис. 3.2.3 представлен результат работы функции архивирования чата.



Рисунок 3.2.3. Результат работы функции архивирования чата

**4. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**4.1. История изменений**

Таблица 4.1.1

История изменений

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Автор | Описание изменений |
| 15.10.2023 | 1.0 | Малакаускайте Анна | Создание сервера |
| 25.10.2023 | 1.1 | Преснухина Лидия | Добавление главного окна |
| 4.11.2023 | 1.2 | Преснухина Лидия | Добавление функции отправки сообщений, закрепления, архивирования и поиска чата |
| 17.11.2023 | 1.3 | Преснухина Лидия | Привязка БД к проекту, добавление файлов в БД |

**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 тестировщику.

**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) | * Работа системы регистрации пользователе; * Работа системы авторизации пользователей. |
| Тестирование производительности (performance testing) | * Скорость сохранения данных в БД; * Скорость отображения данных из БД. |
| Нагрузочное тестирование (load testing) | * Нахождение 100 авторизованных пользователей в системе. |
| Тестирование совместимости (Compatibility testing) | * корректное отображение программы в разных версиях ОС Windows (8, 10); * корректное отображение программы на различных соотношениях сторон экрана; * корректное отображение программы на различных разрешениях экрана; |
| **Различные виды тестов** | |
| Позитивные тесты | * регистрация пользователей, ввод данных в форме регистрации корректных значений; * авторизация пользователей, ввод данных в форме авторизации корректных значений. |
| Негативные тесты | * регистрация пользователей, ввод данных в форме регистрации некорректных значений; * авторизация пользователей, ввод данных в форме авторизации некорректных значений. |
| Исследовательские тесты |  |
| **Уровни тестирования** | |
| Модульное тестирование | * тестирование функций пользователя: регистрация, авторизация, отправка сообщений |
| Интеграционное тестирование | * переход с стартового модуля на модуль авторизации; * переход с модуля авторизации на модуль регистрации. |
| Системное тестирование | * 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.

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 |
| 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 | Тестирование проведено успешно |
| Проверка функции закрепления чата | Графический интерфейс ПО | Чат закреплен | Функция ПО работает корректно | Тестирование проведено успешно |
| Проверка функции архивирования чата | Графический интерфейс ПО | Чат архивирован | Функция ПО работает корректно | Тестирование проведено успешно |
| Проверка регистрации пользователя | Админ, Админ, [admin@mail.ru](mailto:admin@mail.ru) 12345,12345 | Пользователь зарегистрирован | Пользователь зарегистрирован | Пользователь зарегистрирован |
| Проверка входа пользователя | [admin@mail.ru](mailto:admin@mail.ru), 12345 | Пользователь успешно вошел в систему | Пользователь успешно вошел в систему | Пользователь успешно вошел в систему |
| Проверка входа пользователя | [admin@mail.ru](mailto:admin@mail.ru), 123 | Пользователь не смог войти в систему (пароль неверный) | Пользователь не смог войти в систему (пароль неверный) | Пользователь не смог войти в систему (пароль неверный) |
| Проверка регистрации пользователя | Пустое поле | Пользователь не зарегистрирован (пустое поле) | Пользователь не зарегистрирован | Пользователь не зарегистрирован |
| Проверка регистрации пользователя | Админ, Админ, admin, 12345,12345 | Пользователь не зарегистрирован (неверный адрес почты) | Пользователь не зарегистрирован | Пользователь не зарегистрирован |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**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, полей ввода, другие сервисы; взлом паролей; несанкционированный доступ к данным.

(КАК ЭТО СДЕЛАТЬ???) 5. Расчет стоимости информационного актива - определение стоимости каждого информационного актива на основе его значимости для компании и возможных последствий от утраты или повреждения. Формула для расчета стоимости информационного актива может быть следующей:

Стоимость информационного актива = Значимость информационного актива x Вероятность реализации угрозы x Величина возможных последствий.

Важно учитывать, что стоимость информационного актива может изменяться в зависимости от изменения уровня угроз и уязвимостей, а также от изменения значимости актива для компании. Поэтому периодически необходимо проводить повторную оценку и корректировку стоимости информационных активов.

**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.

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 |
| ВН1 | 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) |
| ВН2 | Конкурирующие  организации | +  (месть за ранее  совершенные  действия) | +  (получение финансовой  выгоды за счет кражи и  коммерческой тайны) | - | - |
| ВН3 | Разработчики  программных,  программно-аппаратных средств | + (желание самореализоваться) | +  (передача информации о  юридическом лице третьим  лицам) | +  (внедрение дополнительных  функциональных возможностей  в программные или программно-аппаратные средства на этапе  разработки при вступлении в  сговор со специальными  службами иностранных  государств) | У2  (недополучение ожидаемой  прибыли)  У3  (нарушение функционирования  государственного органа,  дискредитация деятельности  органа государственной власти) |
| ВН4 | Поставщики  вычислительных услуг,  услуг связи | - | +  (передача информации о  юридическом лице третьим  лицам) | - | - |
| ВН5 | Авторизованные пользователи систем и  сетей | +  (непреднамеренные,  неосторожные или  неквалифицированные  действия) | - | - | У1  (финансовый, иной материальный ущерб  физическим лицам) |
| ВН6 | 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) |
| ВН7 | Бывшие работники  (пользователи) | +  (месть за ранее  совершенные  действия) | +  (получение финансовой  выгоды за счет кражи и  коммерческой тайны) | - | - |

Таблица 5.3.2

Возможности нарушителей

|  |  |  |  |
| --- | --- | --- | --- |
| Уровень возможностей нарушителя | | Возможности нарушителя | Виды нарушителя |
| Н1 | Нарушитель,  обладающий  базовыми  возможностями | Имеет возможность при реализации угроз безопасности информации использовать только известные уязвимости, скрипты и инструменты. | Физическое лицо (хакер).  Авторизованные пользователи систем и  Сетей.  Бывшие работники (пользователи). |
| Использовать только известные уязвимости, скрипты и инструменты. Имеет возможность использовать средства реализации угроз (инструменты), свободно распространяемые в сети «Интернет» и разработанные другими лицами, имеет минимальные знания механизмов их функционирования, доставки и выполнения вредоносного программного обеспечения, эксплойтов. |
| Обладает базовыми компьютерными знаниями и навыками на уровне пользователя. |
| Имеет возможность реализации угроз за счет физических воздействий на технические средства обработки и хранения информации, линий связи и обеспечивающие системы систем и сетей при наличии физического доступа к ним. |
| Таким образом, нарушители с базовыми возможностями имеют возможность реализовывать только известные угрозы, направленные на известные (документированные) уязвимости, с использованием общедоступных инструментов |
| Н2 | Нарушитель,  обладающий  базовыми  повышенными  возможностями | Обладает всеми возможностями нарушителей с базовыми возможностями. | Конкурирующие организации. Системные администраторы и администраторы безопасности |
| Имеет возможность использовать средства реализации угроз (инструменты), свободно распространяемые в сети «Интернет» и разработанные другими лицами, однако хорошо владеет этими средствами и инструментами, понимает, как они работают и может вносить изменения в их функционирование для повышения эффективности реализации угроз. |
| Оснащен и владеет фреймворками и наборами средств, инструментов  для реализации угроз безопасности информации и использования  уязвимостей. |
| Имеет навыки самостоятельного планирования и реализации  сценариев угроз безопасности информации |
| Обладает практическими знаниями о функционировании систем и сетей, операционных систем, а также имеет знания защитных механизмов, применяемых в программном обеспечении, программно-аппаратных средствах. |
| Таким образом, нарушители с базовыми повышенными возможностями имеют возможность реализовывать угрозы, в том числе направленные на неизвестные (недокументированные) уязвимости, с использованием специально созданных для этого инструментов, свободно распространяемых в сети «Интернет». Не имеют возможностей реализации угроз на физически изолированные сегменты систем и сетей |
| Н3 | Нарушитель,  обладающий  средними  возможностями | Обладает всеми возможностями нарушителей с базовыми повышенными возможностями | Разработчики программных, программно-аппаратных средств |
| Имеет возможность приобретать информацию об уязвимостях, размещаемую на специализированных платных ресурсах (биржах уязвимостей). |
| Имеет возможность приобретать дорогостоящие средства и инструменты для реализации угроз, размещаемые на специализированных платных ресурсах (биржах уязвимостей). |
| Имеет возможность самостоятельно разрабатывать средства (инструменты), необходимые для реализации угроз (атак), реализовывать угрозы с использованием данных средств. |
| Имеет возможность получения доступа к встраиваемому программному обеспечению аппаратных платформ, системному и прикладному программному обеспечению, телекоммуникационному оборудованию и другим программно-аппаратным средствам для проведения их анализа |
| Обладает знаниями и практическими навыками проведения анализа программного кода для получения информации об уязвимостях |
| Обладает высокими знаниями и практическими навыками о  функционировании систем и сетей, операционных систем, а также имеет глубокое понимание защитных механизмов, применяемых в программном обеспечении, программно-аппаратных средствах. |
| Имеет возможность реализовывать угрозы безопасности информации  в составе группы лиц. |
| Таким образом, нарушители со средними возможностями имеют возможность реализовывать угрозы, в том числе на выявленные ими неизвестные уязвимости, с использованием самостоятельно разработанных для этого инструментов. Не имеют возможностей реализации угроз на физически изолированные сегменты систем и сетей |

Таблица 5.3.3

Актуальные нарушители

|  |  |  |  |
| --- | --- | --- | --- |
| Виды риска (ущерба) и возможные негативные последствия | Виды актуального нарушителя | Категория нарушителя | Уровень возможностей нарушителя |
| У1:  нарушение  конфиденциальности  персональных данных  граждан;  нарушение личной,  семейной тайны, утрата  чести и доброго имени;  финансовый, иной  материальный ущерб  физических лиц | Отдельные  физические лица  (хакеры) | Внешний | Н2 |
| Разработчики  программных,  программно-аппаратных средств | Внутренний | Н3 |
| Системные  администраторы и  администраторы  безопасности | Внутренний | Н2 |
| Авторизованные  пользователи систем  и сетей | Внутренний | Н2 |
| Бывшие (уволенные) работники (пользователи) | Внешний | Н1 |
| У2:  невозможность  заключения договоров,  соглашений;  утечка коммерческой  тайны;  потеря клиентов;  нарушение деловой  репутации;  недополучение  ожидаемой прибыли | Отдельные  физические лица  (хакеры) | Внутренний | Н2 |
| Разработчики  программных,  программно-аппаратных средств | Внутренний | Н3 |
| Системные  администраторы и  администраторы  безопасности | Внутренний | Н2 |
| У3:  нарушение  функционирования  государственного органа,  дискредитация  деятельности органа государственной власти;  доступ к системам и сетям  с целью незаконного  использования  вычислительных  мощностей;  утечка информации  ограниченного доступа;  организация митингов,  забастовок из-за  публикаций  недостоверной  информации;  отсутствие доступа к  социально значимым  государственным услугам | Разработчики  программных,  программно-аппаратных средств | Внутренний | Н3 |
| Системные  администраторы и  администраторы  безопасности | Внутренний | Н3 |
| Авторизованные  пользователи систем  и сетей | Внутренний | Н1 |
| Бывшие  (уволенные)  работники  (пользователи) | Внутренний | Н1 |

Таблица 5.3.4

Способы реализации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Виды нарушителей | Категории нарушителей | Объект воздействия | Доступный интерфейс | Способы реализации |
| Отдельные физические лица (хакеры) (Н2) | Внешний | Удаленное автоматизированное  рабочее место (АРМ)  пользователя:  несанкционированный доступ к  операционной системе АРМ  пользователя;  нарушение  конфиденциальности  информации, содержащейся на  АРМ пользователя | Доступ через  локальную  вычислительную сеть  организации | Внедрение вредоносного  программного обеспечения |
| Съемные машинные  носители информации,  подключаемые к АРМ  пользователя | Использование  уязвимостей конфигурации  системы управления  доступом к АРМ  пользователя |
| Разработчики программных, программно-аппаратных средств (Н3) | Внешний | Удаленное автоматизированное  рабочее место (АРМ) | Доступ через  локальную  вычислительную сеть  организации | Внедрение вредоносного  программного обеспечения |
| Внутренний | Исходный код | Доступ через  исходный код программного продукта |
| Поставщики  вычислительных услуг,  услуг связи (Н2) | Внешний | Линия связи между сервером  основного центра обработки  данных и сервером резервного  центра обработки данных:  перехват (нарушение  конфиденциальности)  защищаемой информации, системных, конфигурационных,  иных служебных данных | Канал передачи данных  между сервером  основного центра  обработки данных и  сервером резервного  центра обработки  данных | Установка программных  закладок в  телекоммуникационное  оборудование |
| Авторизованные  пользователи систем и  сетей (Н1) | Внутренний | Удаленное автоматизированное  рабочее место (АРМ) пользователя | Интерфейс пользователя корпоративной системы коммуникаций | Сбой работы программы из-за неверно использования приложения |
| Внедрение вредоносного  программного обеспечения |
| Системные  администраторы и  администраторы  безопасности (Н3) | Внутренний | АРМ системного администратора и администратора безопасности  организации: | Интерфейс пользователя корпоративной системы коммуникаций | Сбой работы программы из-за неверно использования приложения |
|  |  |  |
| Исходный код | Доступ через  исходный код программного продукта | Внедрение вредоносного  программного обеспечения |
| Бывшие  (уволенные)  работники  (пользователи) (Н1) | Внутренний | Удаленное автоматизированное  рабочее место (АРМ) пользователя | Интерфейс пользователя корпоративной системы коммуникаций | Внедрение вредоносного  программного обеспечения |
| Сбой работы программы из-за неверно использования приложения |

Таблица 5.3.5

Возможные угрозы

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Угроза безопасности информации | | Источник 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  Нарушение 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 на все узлы грид-системы.  Данная угроза обусловлена слабостями технологии грид-вычислений – высоким уровнем автоматизации при малой администрируемости грид-системы.  Реализация данной угрозы возможна при условии наличия у нарушителя привилегий легального пользователя грид-системы | Внешний нарушитель с низким потенциалом | | + | + | + | - | + | + | + | + | - | + | + | в случае работы дискредитируемого пользователя с файлами, поступающими из недоверенных источников;  при наличии у него привилегий установки программного обеспечения;  в случае неизмененных владельцем учетных данных IoT-устройства (заводских пароля и логина) |
| УБИ.015: Угроза доступа к защищаемым файлам с использованием обходного пути | Угроза заключается в возможности получения нарушителем доступа к скрытым/защищаемым каталогам или файлам посредством различных воздействий на файловую систему (добавление дополнительных символов в указании пути к файлу; обращение к файлам, которые явно не указаны в окне приложения). | Внешний нарушитель с низким потенциалом  Внутренний нарушитель с низким потенциалом | | + | - | + | + | + | + | + | + | + | + | + | Реализация данной угрозы возможна при условиях:  наличие у нарушителя прав доступа к некоторым объектам файловой системы;  отсутствие проверки вводимых пользователем данных;  наличие у дискредитируемой программы слишком высоких привилегий доступа к файлам, обработка которых не предполагается с её помощью | Нарушение конфиденциальности. Возможна. |
| УБИ.033: Угроза использования слабостей кодирования входных данных | Угроза заключается в возможности осуществления нарушителем деструктивного информационного воздействия на дискредитируемую систему путём манипулирования значениями входных данных и формой их предоставления (альтернативные кодировки, некорректное расширение файлов и т.п.). | Внешний нарушитель со средним потенциалом  Внутренний нарушитель со средним потенциалом | | + | + | - | - | + | + | + | + | - | - | + | 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 | Нарушение целостности  Нарушение доступности. Возможна. |
| УБИ.088: Угроза несанкционированного копирования защищаемой информации | Угроза заключается в возможности неправомерного получения нарушителем копии защищаемой информации путём проведения последовательности неправомерных действий, включающих: несанкционированный доступ к защищаемой информации, копирование найденной информации на съёмный носитель (или в другое место, доступное нарушителю вне системы). | 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 криптографических мер защиты или снятия копии в момент обработки защищаемой информации в нешифрованном виде | Нарушение конфиденциальности. Возможна. |
| УБИ.090: Угроза несанкционированного создания учётной записи пользователя | Угроза заключается в возможности создания нарушителем в системе дополнительной учётной записи пользователя и её дальнейшего использования в собственных неправомерных целях (входа в систему с правами этой учётной записи и осуществления деструктивных действий по отношению к дискредитированной системе или из дискредитированной системы по отношению к другим системам). | Внешний нарушитель с низким потенциалом  Внутренний нарушитель с низким потенциалом | | + | + | + | - | + | + | + | + | - | - | - | Данная угроза обусловлена слабостями механизмов разграничения доступа к защищаемой информации.  Реализация данной угрозы возможна в случае наличия и прав на запуск специализированных программ для редактирования файлов, содержащих сведения о пользователях системы (при удалённом доступе) или штатных средств управления доступом из состава операционной системы (при локальном доступе) | Нарушение конфиденциальности  Нарушение целостности  Нарушение доступности. Возможна. |
| УБИ.115: Угроза перехвата вводимой и выводимой на периферийные устройства информации | Угроза заключается в возможности осуществления нарушителем несанкционированного доступа к информации, вводимой и выводимой на периферийные устройства, путём перехвата данных, обрабатываемых контроллерами периферийных устройств. | Внешний нарушитель с низким потенциалом  Внутренний нарушитель с низким потенциалом | | + | + | - | + | - | - | - | - | + | + | - | Данная угроза обусловлена недостаточностью мер защиты информации от утечки и контроля потоков данных, а также невозможностью осуществления защиты вводимой и выводимой на периферийные устройства информации с помощью криптографических средств (т.к. представление пользователям системы информации должно осуществляться в доступном для понимания виде).  Реализация данной угрозы возможна при условии наличия у нарушителя привилегий на установку и запуск специализированных вредоносных программ, реализующих функции «клавиатурных шпионов» (для получения нарушителем паролей пользователей), виртуальных драйверов принтеров (перехват документов, содержащих защищаемую информацию) и др. | Нарушение конфиденциальности. Возможна. |

**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 | Т1 | Т1.1-Т1.4 |
| 15 | Угроза доступа к защищаемым файлам с использованием обходного пути | Актуальна | Внешний нарушитель с низким потенциалом  Внутренний нарушитель с низким потенциалом | Т3 | Т3.1-Т3.12 |
| 27 | Угроза искажения вводимой и выводимой на периферийные устройства информации | Актуальна | Внешний нарушитель с высоким потенциалом  Внутренний нарушитель с низким потенциалом | Т2 | Т2.1-Т2.3 |
| 33 | Угроза использования слабостей кодирования входных данных | Актуальна | Внешний нарушитель со средним потенциалом  Внутренний нарушитель со средним потенциалом | Т3 | Т3.1-Т3.12 |
| 88 | Угроза несанкционированного копирования защищаемой информации | Актуальна | Внешний нарушитель с низким потенциалом  Внутренний нарушитель с низким потенциалом | Т3 | Т3.1-Т3.12 |
| 90 | Угроза несанкционированного создания учётной записи пользователя | Актуальна | Внешний нарушитель с низким потенциалом  Внутренний нарушитель с низким потенциалом | Т4 | Т4.1-Т4.8 |
| 115 | Угроза перехвата вводимой и выводимой на периферийные устройства информации | Актуальна | Внешний нарушитель с низким потенциалом  Внутренний нарушитель с низким потенциалом | Т3 | Т3.1-Т3.12 |

Таблица 5.4.2

Описание тактик

|  |  |  |
| --- | --- | --- |
| № | Тактика | Основные техники |
| Т1 | Внедрение и исполнение  вредоносного программного  обеспечения в системах и сетях  Тактическая задача: получив доступ к  узлу сети или системы, нарушитель  стремится внедрить в его программную среду  инструментальные средства,  необходимые ему для дальнейших  действий | Т1.1 Автоматический запуск скриптов и исполняемых файлов в системе с использованием  пользовательских или системных учетных данных, в том числе с использованием методов  социальной инженерии |
| Т1.2 Активация и выполнение вредоносного кода, внедренного в виде закладок в  легитимное программное и программное-аппаратное обеспечение систем и сетей |
| Т1.3 Автоматическая загрузка вредоносного кода с удаленного сайта или ресурса с  последующим запуском на выполнение |
| Т1.4 Копирование и запуск скриптов и исполняемых файлов через средства удаленного  управления операционной системой и сервисами |
| Т2 | Управление вредоносным  программным обеспечением и (или)  компонентами, к которым ранее был  получен доступ  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 | Т2.1. 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 |
| Т2.2. Использование штатных средств удаленного доступа и управления операционной  системы |
| Т2.3. 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 |
| Т3 | Сбор и вывод из системы или сети  информации, необходимой для  дальнейших действий при реализации  угроз безопасности информации или  реализации новых угроз  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 | Т3.1. 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 |
| Т3.2. Доступ к системе для сбора информации и вывод информации через использование  штатных средств удаленного доступа и управления операционной системы |
| Т3.3. Вывод информации на хорошо известные порты на внешних серверах, разрешенные на  межсетевом экране (SMTP/25, HTTP/80, HTTPS/443 и др.) |
| Т3.4. Вывод информации на нестандартные порты на внешних серверах, что в некоторых случаях  позволяет эксплуатировать уязвимости средств сетевой фильтрации для обхода этих средств |
| Т3.5. Отправка данных по известным протоколам управления и передачи данных |
| 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 |
| Т9.9. 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  скомпрометированными 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 |
| Т9.12. 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 | Т4.1. Несанкционированный доступ к информации в памяти системы, файловой системе,  базах данных, репозиториях, в программных модулях и прошивках |
|  |  | Т4.2. Несанкционированное воздействие на системное программное обеспечение, его  конфигурацию и параметры доступа |
|  |  | Т4.3. Несанкционированное воздействие на программные модули прикладного  программного обеспечения |
|  |  | Т4.4. Несанкционированное воздействие на программный код, конфигурацию и параметры  доступа прикладного программного обеспечения |
|  |  | Т4.5. Несанкционированное воздействие на программный код, конфигурацию и параметры доступа системного программного обеспечения |
|  |  | 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 репозиториев 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 информации, 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.

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

**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 ознакомится 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 ознакомится в приложении 4.

**6.3. Определение качества продукта по метрикам**

Таблица 6.3.1

Расчет метрик стилистики и понятности программы

|  |  |  |
| --- | --- | --- |
| Количество строк программы | 78 | |
| Строк в каждом сегменте | 13 | |
| Кол-во сегментов программы | 6 | |
|  | Количество  комментариев |  |
| 1 сегмент | 3 | 1 |
| 2 сегмент | 2 | 1 |
| 3 сегмент | 2 | 1 |
| 4 сегмент | 0 | -1 |
| 5 сегмент | 3 | 1 |
| 6 сегмент | 1 | -1 |
|  |  | 2 |

Расчет:

Сегмент 1. F1=sgn(3/13-0.1) = 1

Сегмент 2. F2=sgn(2/13-0.1) = 1

Сегмент 3. F3=sgn(2/13-0.1) = 1

Сегмент 4. F4=sgn(0/13-0.1) = -1

Сегмент 5. F5=sgn(3/13-0.1) = 1

Сегмент 6. F6=sgn(1/13-0.1) = -1

Вывод: так как уровень комментированности программы F=2, а число сегментов n=6, то условие нормального уровня комментированности F=n не выполняется. При F<n уровень комментированности программы считается низким.

**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°С (допустимые в пределах 18-26°С), 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°С (допустимые в пределах 20-30°С), 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 избежания 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 неприспособленных 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 оборвавшегося 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 несгораемые металлические шкафы, двери в хранилище также должны быть несгораемыми. 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:

o перенапряжение зрительного анализатора при работе за экраном дисплея;

o длительное статическое напряжение мышц спины, шеи, рук и ног, что может привести к статическим перегрузкам программиста;

o повышенный уровень шума;

o ионизирующие и неионизирующие излучения, источниками которых являются видео дисплейные терминалы;

o статическое электричество;

o 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.

**7.5. Вывод**

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 предотвращение.

**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.

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):

, (1)

руб.

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

, (2)

руб.

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

, (3)

руб.

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

, (4)

руб.

**8.2. Эксплуатационные затраты на оборудование**

**8.2.1. Суммарная годовая стоимость эксплуатационных затрат**

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

, (5)

руб,

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):

, (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):

, (7)

руб.

**8.2.4. Амортизационные отчисления**

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

, (8)

руб,

где Собор – стоимость компьютеров;

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

, (9)

лет,

где Тнорм – нормативный срок службы (для ПК Тнорм – 5 лет).

**8.2.5. Стоимость электроэнергии**

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

, (10)

,

,

где М – мощность компьютера, КВт;

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

Fэф – 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 (Fэф) рассчитывается по формуле (11):

, (11)

ч,

где Дном – номинальное число рабочих дней в году при пятидневной рабочей неделе (247 дн.);

d – продолжительность рабочего дня (8 ч.);

f – планируемой 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 временных коэффициентом, который формируется исходя из суммарных годовых эксплуатационных затрат, которые рассчитываются по формуле (12):

(12)

руб.,

где ∑*t* – общее время использования оборудования для решения задачи (из таблицы 1);

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

Fэф – 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):

, (13)

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 четом НДС рассчитывается по формуле (14):

, (14)

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):

, (15)

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):

, (16)

руб.,

где П – прибыль, рассчитанная по формуле (15);

СТП – ставка налогообложения прибыли составляет 20%.

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

, (17)

лет,

где Сполн – полная (плановая) себестоимость продукта;

П – прибыль.

**8.6. Вывод**

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 ()

**ЗАКЛЮЧЕНИЕ**

В ходе исследовательской работы был разработан корпоративный мессенджер.

В первой главе были сформулированы бизнес требования, пользовательские требования, функциональные и нефункциональные требования, требования к интерфейсу и требования к данным.

Во второй главе была рассмотрена архитектура программного продукта, были смоделированы диаграммы деятельности и последовательности.

В третьей главе были рассмотрены и описаны функции продукта.

В четвертой главе было проведено тестирование системы.

В дальнейшем программный продукт будет доработан в рамках дипломного проекта; будут добавлены общая папка, шифрование данных, а также будут соединены серверная и клиентская части.

В процессе написания работы научились работать в команде, распределять обязанности и разрабатывать программный продукт. Полученные нами знания, умения, навыки позволят нам избежать ошибок при 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 С#. 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 С#. // codeacademy.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. <https://mdm.ooo/ru/how-to-create-a-messaging-app/amp/> (дата обращения: 24.11.2023)
8. Разработка интерфейса мессенджера на С#. youtube.com: JD’s Code Lab. Обучение UI Design. 2020. URL: [Chat App UI Design in Wpf C# - YouTube](https://www.youtube.com/playlist?list=PLe_Naf3_gfQl8ZDxkIQpB3-aZJ7aEPFUC)
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.
11. [Рынок корпоративных мессенджеров в России в 2023 году (mts-link.ru)](https://mts-link.ru/blog/rossijskij-rynok-korporativnyh-messendzherov/?ysclid=lt33vzkmrm928283298)
12. <https://ux-journal.ru/skolko-zarabatyvaet-ux-ui-dizajner-na-pike-svoej-formy.html>
13. <https://uchis-online.ru/blog/professii/skolko-zarabativaet-arhitektor-po>
14. <https://systems.education/systems_analyst>
15. <https://skillbox.ru/media/code/skolko-zarabatyvayut-testirovshchiki-i-qainzhenery-v-rossii-issledovanie-skillbox-media/>
16. <https://mojazarplata.ru/karqera/rabota-i-zarplata-v-rossii/rossija-razrabotchiki-i-administratory-baz-dannyh#:~:text=%D0%91%D0%BE%D0%BB%D1%8C%D1%88%D0%B8%D0%BD%D1%81%D1%82%D0%B2%D0%BE%20%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%87%D0%B8%D0%BA%D0%B8%20%D0%B8%20%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80%D1%8B%20%D0%B1%D0%B0%D0%B7,61%20224%20%E2%82%BD%20%D0%B2%20%D0%BC%D0%B5%D1%81%D1%8F%D1%86>.
17. <https://www.rabota.ru/career/catalogue/it/system-administrator/>
18. https://livetyping.com/ru/kak-sozdat-prilozhenie-messenger
19. <https://rt-solar.ru/products/solar_dozor/blog/3423/>
20. <https://selectel.ru/blog/personal-data-security-threat-model/>
21. <https://digital.gov.ru/ru/documents/8930/?utm_referrer=https%3a%2f%2fwww.google.com%2f>
22. <http://shudrova.blogspot.com/2022/07/21.html>
24. <https://www.securitylab.ru/blog/personal/shudrova/352346.php>

**ПРИЛОЖЕНИЕ 1.**

**ЛИСТИНГ ПРОГРАММНЫХ МОДУЛЕЙ**

1. ViewModel.cs

using app\_interface.Commands;

using app\_interface.CustomControls;

using app\_interface.Models;

using System;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.ComponentModel;

using System.Data.SqlClient;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Input;

using System.Windows.Media;

namespace app\_interface.ViewModels

{

public class ViewModel : INotifyPropertyChanged

{

//initializing resource dictionary file

// private readonly ICommand \_command;

private readonly ResourceDictionary dictionary =

Application.LoadComponent(new Uri("app\_interface;component/Assets/icons.xaml",

UriKind.RelativeOrAbsolute)) as ResourceDictionary;

#region MainWindow

#region Properties

public string ContactName { get; set; }

public Uri ContactPhoto { get; set; }

public string LastSeen { get; set; }

#region Search Chats

protected bool \_isSearchBoxOpen { get; set; }

public bool IsSearchBoxOpen

{

get => \_isSearchBoxOpen;

set

{

if (\_isSearchBoxOpen == value) return;

\_isSearchBoxOpen = value;

if (\_isSearchBoxOpen != false) SearchText = string.Empty;

OnPropertyChanged("IsSearchBoxOpen");

OnPropertyChanged("SearchText");

}

}

protected string LastSearchText { get; set; }

protected string mSearchText { get; set;}

public string SearchText

{

get => mSearchText;

set

{

//checked if value is different

if (mSearchText == value) { return; }

//update view

mSearchText = value;

//if search text is empty restore message

if (string.IsNullOrEmpty(SearchText))

Search();

}

}

//this is our list containing the window options...

private ObservableCollection<MoreOptionsMenu> \_windowMoreOptionsMenuList;

public ObservableCollection<MoreOptionsMenu> WindowMoreOptionsMenuList

{

get {

return \_windowMoreOptionsMenuList;

}

set

{

\_windowMoreOptionsMenuList = value;

}

}

private ObservableCollection<MoreOptionsMenu> \_attachmentOptionsMenuList;

public ObservableCollection<MoreOptionsMenu>AttachmentOptionsMenuList

{

get

{

return \_attachmentOptionsMenuList;

}

set

{

\_attachmentOptionsMenuList = value;

}

}

#endregion

#endregion

#region Logics

#region Window: More Options Popup

void WindowMoreOptionsMenu()

{

WindowMoreOptionsMenuList = new ObservableCollection<MoreOptionsMenu>()

{

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["newgroup"],

MenuText = "New Group"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["newbroadcast"],

MenuText = "New Broadcast"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["starredmessages"],

MenuText = "Starred Messages"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["settings"],

MenuText = "Settings"

},

};

OnPropertyChanged("WindowMoreOptionsMenuList");

}

void ConversationScreenMoreOptionsMenu()

{

//to populate menu items for conversation screen options list..

WindowMoreOptionsMenuList = new ObservableCollection<MoreOptionsMenu>()

{

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["all media"],

MenuText = "All Media"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["wallpaper"],

MenuText = "Change Wallpaper"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["report"],

MenuText = "Report"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["block"],

MenuText = "Block"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["clearchat"],

MenuText = "Clear Chat"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["exportchat"],

MenuText = "Export Chat"

},

};

OnPropertyChanged("WindowMoreOptionsMenuList");

}

void AttachmentOptionsMenu()

{

//to populate menu items for attachment menu options list..

AttachmentOptionsMenuList = new ObservableCollection<MoreOptionsMenu>()

{

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["docs"],

MenuText = "Docs",

BorderStroke="#3F3990",

Fill="#CFCEEC"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["gallery"],

MenuText = "Gallery",

BorderStroke="#3F3990",

Fill="#CFCEEC"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["audio"],

MenuText = "Audio",

BorderStroke="#3F3990",

Fill="#CFCEEC"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["location"],

MenuText = "Location",

BorderStroke="#3F3990",

Fill="#CFCEEC"

},

new MoreOptionsMenu()

{

Icon = (PathGeometry)dictionary["contact"],

MenuText = "Contact",

BorderStroke="#3F3990",

Fill="#CFCEEC"

},

};

OnPropertyChanged("AttachmentOptionsMenuList");

}

#endregion

public void OpenSearchBox()

{

IsSearchBoxOpen = true;

}

public void ClearSearchBox()

{

if (!string.IsNullOrEmpty(SearchText)) SearchText = string.Empty;

else CloseSearchBox();

}

public void CloseSearchBox() => IsSearchBoxOpen = false;

public void Search()

{

//to avoid searching same text again

if((string.IsNullOrEmpty(LastSearchText) && string.IsNullOrEmpty(SearchText)) || string.Equals(LastSearchText,SearchText))

{ return; }

//if searchbox is empty or chat in null pr char cound less then 0

if (string.IsNullOrEmpty(SearchText) || Chats == null || Chats.Count <= 0)

{

FilteredChats = new ObservableCollection<ChatListData>(Chats ?? Enumerable.Empty<ChatListData>());

OnPropertyChanged("FilteredChats");

FilteredPinnedChats = new ObservableCollection<ChatListData>(PinnedChats ?? Enumerable.Empty<ChatListData>());

OnPropertyChanged("FilteredPinnedChats");

//update last search text

LastSearchText= SearchText;

return;

}

//now, to find all chats that contain the text in our search box

FilteredChats = new ObservableCollection<ChatListData>(

Chats.Where(

chat => chat.ContactName.ToLower().Contains(SearchText)

||

chat.Message!= null && chat.Message.ToLower().Contains(SearchText)

));

OnPropertyChanged("FilteredChats");

FilteredPinnedChats = new ObservableCollection<ChatListData>(

PinnedChats.Where(

pinnedchat => pinnedchat.ContactName.ToLower().Contains(SearchText)

||

pinnedchat.Message != null && pinnedchat.Message.ToLower().Contains(SearchText)

));

OnPropertyChanged("FilteredPinnedChats");

//update last search text

LastSearchText = SearchText;

}

#endregion

#region Commands

protected ICommand \_windowsMoreOptionsCommand;

public ICommand WindowsMoreOptionsCommand

{

get

{

if (\_windowsMoreOptionsCommand == null)

\_windowsMoreOptionsCommand = new CommandViewModel(WindowMoreOptionsMenu);

return \_windowsMoreOptionsCommand;

}

set

{

\_windowsMoreOptionsCommand = value;

}

}

protected ICommand \_conversationScreenMoreOptionsCommand;

public ICommand ConversationScreenMoreOptionsCommand

{

get

{

if (\_conversationScreenMoreOptionsCommand == null)

\_conversationScreenMoreOptionsCommand = new CommandViewModel(ConversationScreenMoreOptionsMenu);

return \_conversationScreenMoreOptionsCommand;

}

set

{

\_conversationScreenMoreOptionsCommand = value;

}

}

protected ICommand \_attachmentOptionsCommand;

public ICommand AttachmentOptionsCommand

{

get

{

if (\_attachmentOptionsCommand == null)

\_attachmentOptionsCommand = new CommandViewModel(AttachmentOptionsMenu);

return \_attachmentOptionsCommand;

}

set

{

\_attachmentOptionsCommand = value;

}

}

protected ICommand \_openSearchCommand;

public ICommand OpenSearchCommand

{

get

{

if (\_openSearchCommand == null)

\_openSearchCommand = new CommandViewModel(OpenSearchBox);

return \_openSearchCommand;

}

set

{

\_openSearchCommand = value;

}

}

protected ICommand \_closeSearchCommand;

public ICommand CloseSearchCommand

{

get

{

if (\_closeSearchCommand == null)

\_closeSearchCommand = new CommandViewModel(CloseSearchBox);

return \_closeSearchCommand;

}

set

{

\_closeSearchCommand = value;

}

}

protected ICommand \_searchCommand;

public ICommand SearchCommand

{

get

{

if (\_searchCommand == null)

\_searchCommand = new CommandViewModel(Search);

return \_searchCommand;

}

set

{

\_searchCommand = value;

}

}

protected ICommand \_clearSearchCommand;

public ICommand ClearSearchCommand

{

get

{

if (\_clearSearchCommand == null)

\_clearSearchCommand = new CommandViewModel(ClearSearchBox);

return \_clearSearchCommand;

}

set

{

\_clearSearchCommand = value;

}

}

#endregion

#endregion

#region Status Thumbs

#region Properties

#region Logics

public ObservableCollection<StatusDataModel> StatusThumbsCollection { get; set; }

#endregion

void LoadStatusThumbs()

{

StatusThumbsCollection = new ObservableCollection<StatusDataModel>()

{

new StatusDataModel{

IsMeAddStatus = true

},

new StatusDataModel{

ContactName="Майк",

ContactPhoto=new Uri("/Assets/1.png",UriKind.RelativeOrAbsolute),

StatusImage=new Uri("/Assets/5.jpg",UriKind.RelativeOrAbsolute),

IsMeAddStatus=false

},

new StatusDataModel{

ContactName="Стив",

ContactPhoto=new Uri("/Assets/2.jpg",UriKind.RelativeOrAbsolute),

StatusImage=new Uri("/Assets/8.jpg",UriKind.RelativeOrAbsolute),

IsMeAddStatus=false

},

new StatusDataModel{

ContactName="Вилл",

ContactPhoto=new Uri("/Assets/3.png",UriKind.RelativeOrAbsolute),

StatusImage=new Uri("/Assets/5.jpg",UriKind.RelativeOrAbsolute),

IsMeAddStatus=false

},

new StatusDataModel{

ContactName="Джон",

ContactPhoto=new Uri("/Assets/4.png",UriKind.RelativeOrAbsolute),

StatusImage=new Uri("/Assets/3.jpg",UriKind.RelativeOrAbsolute),

IsMeAddStatus=false

},

};

OnPropertyChanged("StatusThumbsCollection");

}

#endregion

#endregion

#region Chats List

#region Properties

public ObservableCollection<ChatListData> mChats;

public ObservableCollection<ChatListData> mPinnedChats;

public ObservableCollection<ChatListData> Chats

{

get => mChats;

set {

//to change the list

if (mChats == value) return;

//to update the list

mChats = value;

//updating filtered chats to match

FilteredChats = new ObservableCollection<ChatListData>(mChats);

OnPropertyChanged("Chats");

OnPropertyChanged("FilteredChats");

}

}

public ObservableCollection<ChatListData> PinnedChats

{

get => mPinnedChats;

set

{

//to change the list

if (mPinnedChats == value) return;

//to update the list

mPinnedChats = value;

//updating filtered chats to match

FilteredPinnedChats = new ObservableCollection<ChatListData>(mPinnedChats);

OnPropertyChanged("PinnedChats");

OnPropertyChanged("FilteredPinnedChats");

}

}

protected ObservableCollection<ChatListData> \_ArchiveChats;

public ObservableCollection<ChatListData> ArchiveChats

{ get => \_ArchiveChats;

set {

\_ArchiveChats = value;

OnPropertyChanged();

} }

//Filtering Chats, Pinned chats

public ObservableCollection<ChatListData> FilteredChats { get; set; }

public ObservableCollection<ChatListData> FilteredPinnedChats { get; set; }

protected int ChatPosition { get; set; }

#endregion

#region Logics

void LoadChats()

{

Chats = new ObservableCollection<ChatListData>()

{

new ChatListData

{

ContactName="Билл",

ContactPhoto=new Uri("/Assets/6.jpg",UriKind.RelativeOrAbsolute),

Message="Привет, как дела?",

LastMessageTime="вт, 15:28",

ChatIsSelected=true

},

new ChatListData

{

ContactName="Майк",

ContactPhoto=new Uri("/Assets/1.png",UriKind.RelativeOrAbsolute),

Message="Проверь почту",

LastMessageTime="пн, 12:01"

},

new ChatListData

{

ContactName="Стив",

ContactPhoto=new Uri("/Assets/7.png",UriKind.RelativeOrAbsolute),

Message="Да, было весело)",

LastMessageTime="вт, 08:10"

},

new ChatListData

{

ContactName="Джон",

ContactPhoto=new Uri("/Assets/8.jpg",UriKind.RelativeOrAbsolute),

Message="Как ты?",

LastMessageTime="вт, 01:00"

}

};

OnPropertyChanged();

}

#endregion

#region Commands

//to get the contactnaem of selected chat

protected ICommand \_getSelectedChatCommand;

public ICommand GetSelectedChatCommand => \_getSelectedChatCommand ?? new RelayCommand(parameter =>

{

if (parameter is ChatListData v)

{

//getting contactname from selected chat

ContactName = v.ContactName;

OnPropertyChanged("ContactName");

//getting contactphoto from selected chat

ContactPhoto = v.ContactPhoto;

OnPropertyChanged("ContactPhoto");

LoadChatConversation(v);

}

});

//to pin chat on pin button click

protected ICommand \_pinChatCommand;

public ICommand PinChatCommand => \_pinChatCommand ?? new RelayCommand(parameter =>

{

if (parameter is ChatListData v)

{

if (!FilteredPinnedChats.Contains(v))

{

//add selected chat to pin chat

PinnedChats.Add(v);

FilteredPinnedChats.Add(v);

OnPropertyChanged("PinnedChats");

OnPropertyChanged("FilteredPinnedChats");

v.ChatIsPinned = true;

//remove selected chat from all chats / unpinned chats

//store position of chat befor pinning sp that when we unoin or unarchive we get it on original position...

ChatPosition = Chats.IndexOf(v);

Chats.Remove(v);

FilteredChats.Remove(v);

OnPropertyChanged("Chats");

OnPropertyChanged("FilteredChats");

if(ArchiveChats!=null)

{

if (ArchiveChats.Contains(v))

{

ArchiveChats.Remove(v);

v.ChatIsArchive = false;

}

}

}

}

});

//to pin chat on pin button click

protected ICommand \_unPinChatCommand;

public ICommand UnPinChatCommand => \_unPinChatCommand ?? new RelayCommand(parameter =>

{

if (parameter is ChatListData v)

{

if (!FilteredPinnedChats.Contains(v))

{

//restore position of chat befor pinning sp that when we unoin or unarchive we get it on original position...

//remove selected chat from all chats / unpinned chats

Chats.Add(v);

FilteredChats.Add(v);

//restore position of chat befor pinning sp that when we unoin or unarchive we get it on original position...

Chats.Move(Chats.Count-1, ChatPosition);

FilteredChats.Move(Chats.Count-1, ChatPosition);

//update

OnPropertyChanged("Chats");

OnPropertyChanged("FilteredChats");

//add selected chat to pin chat

PinnedChats.Remove(v);

FilteredPinnedChats.Remove(v);

OnPropertyChanged("PinnedChats");

OnPropertyChanged("FilteredPinnedChats");

v.ChatIsPinned = false;

}

}

});

protected ICommand \_archiveChatCommand;

public ICommand ArchiveChatCommand => \_archiveChatCommand ?? new RelayCommand(parameter =>

{

if (parameter is ChatListData v)

{

if (!ArchiveChats.Contains(v))

{

//add chat in archive list

ArchiveChats.Add(v);

v.ChatIsArchive = true;

v.ChatIsPinned= false;

//remove chat from pinned and unpinned chat list

Chats.Remove(v);

FilteredChats.Remove(v);

PinnedChats.Remove(v);

FilteredPinnedChats.Remove(v);

//update list

OnPropertyChanged("Chats");

OnPropertyChanged("FilteredChats");

OnPropertyChanged("PinnedChats");

OnPropertyChanged("FilteredPinnedChats");

OnPropertyChanged("ArchiveChats");

}

}

});

protected ICommand \_unArchiveChatCommand;

public ICommand UnArchiveChatCommand => \_unArchiveChatCommand ?? new RelayCommand(parameter =>

{

if (parameter is ChatListData v)

{

if (!FilteredChats.Contains(v) && !Chats.Contains(v))

{

//remove chat from pinned and unpinned chat list

Chats.Add(v);

FilteredChats.Add(v);

}

//add chat in archive list

ArchiveChats.Remove(v);

v.ChatIsArchive = false;

v.ChatIsPinned = false;

OnPropertyChanged("Chats");

OnPropertyChanged("FilteredChats");

OnPropertyChanged("ArchiveChats");

}

});

#endregion

#endregion

#region Conversations

#region Properties

protected ObservableCollection<ChatConversation> mConversations;

public ObservableCollection<ChatConversation> Conversations

{

get => mConversations;

set

{

//to change the list

if (mConversations == value) return;

//to update the list

mConversations = value;

//updating filtered chats to match

FilteredConversations = new ObservableCollection<ChatConversation>(mConversations);

OnPropertyChanged("Conversations");

OnPropertyChanged("FilteredConversations");

}

}

public ObservableCollection<ChatConversation> FilteredConversations { get; set; }

protected string messageText;

public string MessageText

{

get => messageText;

set

{

messageText = value;

OnPropertyChanged("MessageText");

}

}

protected string LastSearchConversationText;

protected string mSearchConversationText;

public string SearchConversationText

{

get => mSearchConversationText;

set

{

//checked if value is different

if (mSearchConversationText == value) { return; }

//update view

mSearchConversationText = value;

//if search text is empty restore message

if (string.IsNullOrEmpty(SearchConversationText)) SearchInConversation();

}

}

public bool FocusMessageBox { get; set; }

public bool IsThisAReplyMessage { get; set; }

public string MessageToReplyText { get; set; }

#endregion

#region Logics

protected bool \_isConversationSearchBoxOpen;

public bool IsConversationSearchBoxOpen

{

get => \_isConversationSearchBoxOpen;

set

{

if (\_isConversationSearchBoxOpen == value) return;

\_isConversationSearchBoxOpen = value;

if (\_isConversationSearchBoxOpen == false) SearchConversationText = string.Empty;

OnPropertyChanged("IsConversationSearchBoxOpen");

OnPropertyChanged("SearchConversationText");

}

}

public void OpenConversationSearchBox()

{

IsConversationSearchBoxOpen = true;

}

public void ClearConversationSearchBox()

{

if (!string.IsNullOrEmpty(SearchConversationText))

SearchConversationText = string.Empty;

else CloseConversationSearchBox();

}

public void CloseConversationSearchBox() => IsConversationSearchBoxOpen = false;

private void LoadChatConversation( ChatListData chat)

{

if (connection.State == System.Data.ConnectionState.Closed)

connection.Open();

if(Conversations == null)

Conversations = new ObservableCollection<ChatConversation>();

Conversations.Clear();

FilteredConversations.Clear();

using (SqlCommand com = new SqlCommand("select \* from conversations where ContactName=@ContactName", connection))

{

com.Parameters.AddWithValue("@ContactName",chat.ContactName);

using (SqlDataReader reader = com.ExecuteReader())

{

while (reader.Read())

{

string MsgReceivedOn = !string.IsNullOrEmpty(reader["MsgReceivedOn"].ToString()) ?

Convert.ToDateTime(reader["MsgReceivedOn"].ToString()).ToString("MMM dd, hh:mm tt") : "";

string MsgSentOn = !string.IsNullOrEmpty(reader["MsgSentOn"].ToString()) ?

Convert.ToDateTime(reader["MsgSentOn"].ToString()).ToString("MMM dd, hh:mm tt") : "";

var conversation = new ChatConversation()

{

ContactName = reader["ContactName"].ToString(),

ReceivedMessage = reader["ReceivedMsgs"].ToString(),

MsgReceivedOn = MsgReceivedOn,

MsgSentOn = MsgSentOn,

SentMessage = reader["SentMsgs"].ToString(),

IsMessageReceived = !string.IsNullOrEmpty(reader["ReceivedMsgs"].ToString())

};

Conversations.Add(conversation);

OnPropertyChanged("Conversations");

FilteredConversations.Add(conversation);

OnPropertyChanged("FilteredConversations");

chat.Message = !string.IsNullOrEmpty(reader["ReceivedMsgs"].ToString())? reader["ReceivedMsgs"].ToString():

reader["SentMsgs"].ToString();

}

}

}

//reset reply message text when the new chat is fetched

MessageToReplyText=string.Empty;

OnPropertyChanged("MessageToReplyText");

}

void SearchInConversation()

{

//to avoid searching same text again

if ((string.IsNullOrEmpty(LastSearchConversationText) && string.IsNullOrEmpty(SearchConversationText)) || string.Equals(LastSearchConversationText, SearchConversationText))

{ return; }

//if searchbox is empty or chat in null pr char cound less then 0

if (string.IsNullOrEmpty(SearchConversationText) || Conversations == null || Conversations.Count <= 0)

{

FilteredConversations = new ObservableCollection<ChatConversation>(Conversations ?? Enumerable.Empty<ChatConversation>());

OnPropertyChanged("FilteredConversations");

//update last search text

LastSearchConversationText = SearchConversationText;

return;

}

//now? to find all chats that contain the text in our search box

FilteredConversations = new ObservableCollection<ChatConversation>(

Conversations.Where(

chat => chat.ReceivedMessage.ToLower().Contains(SearchConversationText)

||

chat.SentMessage.ToLower().Contains(SearchConversationText)

));

OnPropertyChanged("FilteredConversations");

//update last search text

LastSearchConversationText = SearchConversationText;

}

public void CancelReply()

{

IsThisAReplyMessage = false;

//reset reply msg text

MessageToReplyText = string.Empty;

OnPropertyChanged("MessageToReplyText");

}

public void SendMessage()

{

//send message only when the textbox is not empty

if(!string.IsNullOrEmpty(MessageText))

{

var conversation = new ChatConversation()

{

ReceivedMessage = MessageToReplyText,

SentMessage = MessageText,

MsgSentOn = DateTime.Now.ToString("MMM dd, hh:mm tt"),

MessageContainsReply = IsThisAReplyMessage

};

//add msg to conversation list

FilteredConversations.Add(conversation);

Conversations.Add(conversation);

//clear message properties and textbox when message is sent

MessageText=string.Empty;

IsThisAReplyMessage = false;

MessageToReplyText= string.Empty;

UpdateChatAndMoveUp(Chats, conversation);

UpdateChatAndMoveUp(PinnedChats, conversation);

UpdateChatAndMoveUp(FilteredChats, conversation);

UpdateChatAndMoveUp(FilteredPinnedChats, conversation);

UpdateChatAndMoveUp(ArchiveChats, conversation);

//update

OnPropertyChanged("FilteredConversations");

OnPropertyChanged("Conversations");

OnPropertyChanged("MessageText");

OnPropertyChanged("IsThisAReplyMessage");

OnPropertyChanged("MessageToReplyText");

}

}

//move the chat contact on top when new message is sent or recieved

protected void UpdateChatAndMoveUp(ObservableCollection<ChatListData> chatList, ChatConversation conversation)

{

//check if the message sent is to selected contact or not

var chat = chatList.FirstOrDefault(x=>x.ContactName == ContactName);

//if found.. then

if(chat != null)

{

//Update Contact Chat Last Message and Messahe Time..

chat.Message = MessageText;

chat.LastMessageTime = conversation.MsgSentOn;

//move chat on top when new msg is recieved/sent...

chatList.Move(chatList.IndexOf(chat), 0);

//update collection

OnPropertyChanged("Chats");

OnPropertyChanged("PinnedChats");

OnPropertyChanged("FilteredChats");

OnPropertyChanged("FilteredPinnedChats");

OnPropertyChanged("ArchiveChats");

}

}

#endregion

#region Commands

protected ICommand \_openConversationSearchCommand;

public ICommand OpenConversationSearchCommand

{

get

{

if (\_openConversationSearchCommand == null)

\_openConversationSearchCommand = new CommandViewModel(OpenConversationSearchBox);

return \_openConversationSearchCommand;

}

set

{

\_openConversationSearchCommand = value;

}

}

protected ICommand \_clearConversationSearchCommand;

public ICommand ClearConversationSearchCommand

{

get

{

if (\_clearConversationSearchCommand == null)

\_clearConversationSearchCommand = new CommandViewModel(ClearConversationSearchBox);

return \_clearConversationSearchCommand;

}

set

{

\_searchConversationCommand = value;

}

}

protected ICommand \_closeConversationSearchCommand;

public ICommand CloseConversationSearchCommand

{

get

{

if (\_closeConversationSearchCommand == null)

\_closeConversationSearchCommand = new CommandViewModel(CloseConversationSearchBox);

return \_closeConversationSearchCommand;

}

set

{

\_searchConversationCommand = value;

}

}

protected ICommand \_searchConversationCommand;

public ICommand SearchConversationCommand

{

get

{

if (\_searchConversationCommand == null)

\_searchConversationCommand = new CommandViewModel(SearchInConversation);

return \_searchConversationCommand;

}

set

{

\_searchConversationCommand = value;

}

}

protected ICommand \_replyCommand;

public ICommand ReplyCommand => \_replyCommand ?? new RelayCommand(parameter =>

{

if (parameter is ChatConversation v)

{

//if replying sender's message

if (v.IsMessageReceived)

MessageToReplyText = v.ReceivedMessage;

//if replying own message

else

MessageToReplyText = v.SentMessage;

//update

OnPropertyChanged("MessageToReplyText");

//set focus on message box when user clicks reply button

FocusMessageBox = true;

OnPropertyChanged("FocusMessageBox");

//flag this msg as reply msg

IsThisAReplyMessage = true;

OnPropertyChanged("IsThisAReplyMessage");

}

});

protected ICommand \_cancelReplyCommand;

public ICommand CancelReplyCommand

{

get

{

if (\_cancelReplyCommand == null)

\_cancelReplyCommand = new CommandViewModel(CancelReply);

return \_cancelReplyCommand;

}

set

{

\_cancelReplyCommand = value;

}

}

protected ICommand \_sendMessageCommand;

public ICommand SendMessageCommand

{

get

{

if (\_sendMessageCommand == null)

\_sendMessageCommand = new CommandViewModel(SendMessage);

return \_sendMessageCommand;

}

set

{

\_sendMessageCommand = value;

}

}

#endregion

#endregion

#region ContactInfo

#region Properties

protected bool \_IsContactInfoOpen;

public bool IsContactInfoOpen

{

get=> \_IsContactInfoOpen;

set { \_IsContactInfoOpen = value;

OnPropertyChanged("IsContactInfoOpen");

}

}

#endregion

#region Logics

public void OpenContactInfo()=>IsContactInfoOpen=true;

public void CloseContactInfo() => IsContactInfoOpen = false;

#endregion

#region Commands

protected ICommand \_openContactInfoCommand;

public ICommand OpenContactInfoCommand

{

get

{

if (\_openContactInfoCommand == null)

\_openContactInfoCommand = new CommandViewModel(OpenContactInfo);

return \_openContactInfoCommand;

}

set

{

\_openContactInfoCommand = value;

}

}

protected ICommand \_closeContactInfoCommand;

public ICommand CloseContactInfoCommand

{

get

{

if (\_closeContactInfoCommand == null)

\_closeContactInfoCommand = new CommandViewModel(CloseContactInfo);

return \_closeContactInfoCommand;

}

set

{

\_closeContactInfoCommand = value;

}

}

#endregion

#endregion

SqlConnection connection = new SqlConnection(@"Data Source=LAPTOP-S3L918JB;Initial Catalog=Database;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

public ViewModel()

{

LoadStatusThumbs();

LoadChats();

PinnedChats = new ObservableCollection<ChatListData>();

ArchiveChats = new ObservableCollection<ChatListData>();

}

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void OnPropertyChanged([CallerMemberName] string propertyName = null)

{

PropertyChanged?.Invoke(this,new PropertyChangedEventArgs(propertyName));

}

}

}

2. Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.ServiceModel;

namespace ChatHost

{

class Program

{

static void Main(string[] args)

{

using (var host = new ServiceHost(typeof(wcf\_chat.ServiceChat)))

{

host.Open();

Console.WriteLine("host is started!");

Console.ReadLine();

}

}

}

}

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.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.Text;

namespace wcf\_chat

{

[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

public class ServiceChat : IServiceChat

{

List<ServerUser> users = new List<ServerUser>();

int nextId = 1;

public int Connect(string name)

{

ServerUser user = new ServerUser()

{

ID = nextId,

Name = name,

operationContext = OperationContext.Current

};

nextId++;

SendMsg(": " + user.Name + " was added", 0);

users.Add(user);

return user.ID;

}

public void Disconnect(int id)

{

var user = users.FirstOrDefault(i => i.ID == id);

if (user != null)

{

users.Remove(user);

SendMsg(": " + user.Name + " left", 0);

}

}

public void SendMsg(string msg, int id)

{

foreach (var item in users)

{

string answer = DateTime.Now.ToShortTimeString();

var user = users.FirstOrDefault(i => i.ID == id);

if (user != null)

{

answer += ": " + user.Name + " ";

}

answer += msg;

item.operationContext.GetCallbackChannel<IServerChatCallback>().MsgCallback(answer);

}

}

}

}

**ПРИЛОЖЕНИЕ 2**

**ПОЛИТИКА БЕЗОПАСНОСТИ**

**Политика безопасности корпоративного мессенджера PigeonGram**

**1. Введение**

Целью данной политики безо **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.

пасности 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»).

**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 проверят 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 персональные 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 пользуетесь 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.

**ПРИЛОЖЕНИЕ 3**

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**ПРИЛОЖЕНИЕ 4**

**РУКОВОДСТВО АДМИНИСТРАТОРА**

**ПРИЛОЖЕНИЕ 5**

**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.