

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

Институт Информационных Технологий

Кафедра Корпоративных Информационных Систем

**КУРСОВАЯ РАБОТА**

по дисциплине

Разработка клиент-серверных приложений

**Тема курсовой работы:** Система корпоративных сообщений с групповыми и индивидуальными чатами

|  |  |  |
| --- | --- | --- |
| Студент группы | ИКБО-08-18  Валяев Данила Андреевич | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |
|  |  |  |
| Руководитель курсовой работы | Ст. преп.  Мирзоян Дмитрий Ильич | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись руководителя) |
|  |  |  |
| Работа представлена к защите | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2020 г. |  |
|  |  |  |
| Допущен к защите | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2020 г. |  |

Москва 2020



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

Институт Информационных Технологий

Кафедра Корпоративных Информационных Систем

Утверждаю

Заведующий кафедрой КИС

\_\_\_\_\_\_\_\_\_\_\_ Андрианова Е.Г.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ**

**на выполнение курсовой работы по дисциплине**

Разработка клиент-серверных приложений

**Студент:** Валяев Данила Андреевич **Группа:** ИКБО-08-18

**Тема** Система корпоративных сообщений с групповыми и индивидуальными чатами

**Исходные данные:** Описание языка и среды разработки на языке Python, стандарты оформления программного кода, Инструкция по организации и проведению курсового проектирования СМКО МИРЭА 7.5.1/04.И.05-18, Рекомендации по оформлению письменных работ СМКО МИРЭА 7.5.1/03.П.69

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:**

1. Анализ предметной области
2. Проектирование приложения
3. Формат долговременного хранения данных в БД
4. Пользовательский интерфейс приложения
5. Набор автоматизируемых функций

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Срок представления к защите курсовой работы:** | | | до «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г. | |
|  | | |  | |
| **Задание на курсовую работу выдал** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | (Мирзоян Д.И.) |
|  | | | до «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г. | |
| **Задание на курсовую работу получил** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | (Валеяв Д.А.) |
|  | | до «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г. | | |

# Техническое задание:

# *Наименование работы*

# Система корпоративных сообщений с групповыми и индивидуальными чатами.

# *Исполнитель*

Валяев Данила Андреевич Шифр: 18И1192

*Основание для разработки:*

Учебный план бакалавриата по направлению 09.03.04 «Программная инженерия».

# *Цель и назначение работы*

Данный продукт разработан с целью повышения удобства обмена сообщениями между работниками предприятия.

# *Технические требования*

# Система должна обеспечивать передачу сообщений между пользователями, как в индивидуальных (пользователь-пользователь, приват), так и в настраиваемых групповых каналах. Необходимо предусмотреть аутентификацию, управление учетными записями, включающее группы и списки контроля доступа, создание и настройку групп (обратите внимание, группа чата и группа безопасности — понятия, не связанные между собой).

# Система должна хранить и позволять просматривать историю сообщений за длительный период (не менее года). Также необходимо предусмотреть механизм обмена сообщениями с прикрепленными вложениями (файлами)*Содержание работы*

В результате выполнения работы решаются следующие задачи:

1. Составление теоретического материала
2. Разработка структуры системы
3. Реализация приложения
4. Тестирование приложения
5. Написание инструкции
6. *Порядок сдачи работы*

Выполнения работы (РПЗ и ИС в соответствии со стандартами кафедры КИС)

**Руководитель**  **Исполнитель**

(подпись, дата) (подпись, дата)

**Аннотация**

В данной курсовой работе рассмотрено создание системы корпоративных сообщений с групповыми и индивидуальными чатами, приложение типа Desktop application для целевой операционной системы Windows. Рассмотрены процессы: анализа предметной области, проектирования приложения, реализации приложения и тестирования приложения, а также написания инструкции для конечного пользователя.

Была написана инструкция для пользователя, в которой подробно и пошагово расписаны действия пользователя для осуществления типовых сценариев работы с системой корпоративных сообщений с групповыми и индивидуальными чатами.

**Графические материалы**

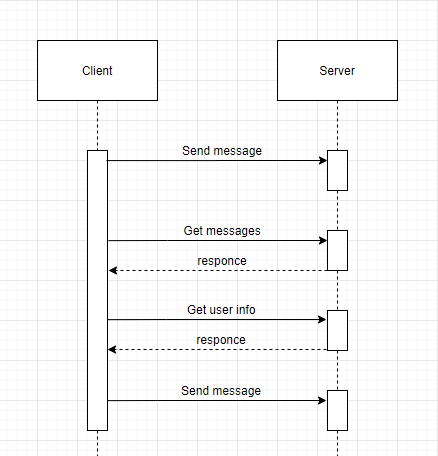


Рис. 1 – Диаграмма сетевого обмена

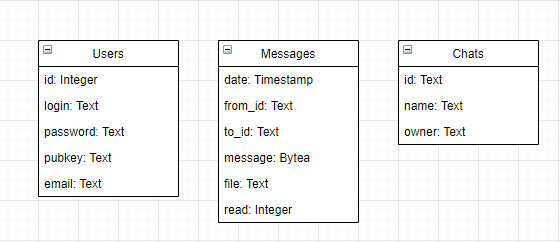


Рис. 2 – ER диаграмма приложения

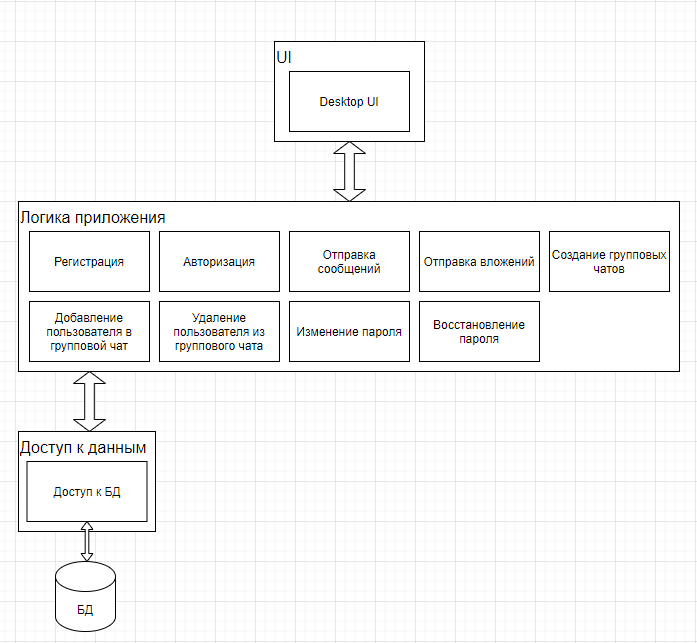


Рис. 3 – Архитектурная модель приложения

**Содержание**

[**Введение 1**](#_Toc59737763)

[**1. Анализ предметной области 3**](#_Toc59737764)

[**1.1. Исследование предметной области 3**](#_Toc59737765)

[**1.2. Обзор существующих в предметной области систем 5**](#_Toc59737766)

[**1.3. Обзор существующих в предметной области технологий 8**](#_Toc59737767)

[**1.4. Вывод к первой главе 12**](#_Toc59737768)

[**2. Проектирование приложения 12**](#_Toc59737769)

[**2.1. Разработка пользовательского интерфейса 12**](#_Toc59737770)

[**2.2. Описание структуры программы 13**](#_Toc59737771)

[**2.3. Алгоритмы, используемые вы программе 13**](#_Toc59737772)

[**2.4. Проектирование системы хранения 14**](#_Toc59737773)

[**2.5. Проектирование программного приложения 14**](#_Toc59737774)

[**3. Реализация приложения 16**](#_Toc59737775)

[**3.1. Реализация алгоритмов 16**](#_Toc59737776)

[**3.2. Реализация программного приложения 20**](#_Toc59737777)

[**3.3. Вывод к третьей главе 24**](#_Toc59737778)

[**4. Тестирование приложения 24**](#_Toc59737779)

[**4.1. Тестирование программного приложения 24**](#_Toc59737780)

[**5. Инструкция 27**](#_Toc59737781)

[**5.1. Инструкция для пользователя 27**](#_Toc59737782)

[**6. Заключение 32**](#_Toc59737783)

[**7. Список используемых источников 33**](#_Toc59737784)

[**8. Приложение 34**](#_Toc59737785)

# **Введение**

В данной курсовой работе представлена реализация системы корпоративных сообщений с групповыми и индивидуальными чатами. Данная система реализует функции отправки сообщений или вложений в индивидуальные и групповые чаты; создания группового чата; регистрации; авторизации; а также восстановления или изменения пароля.

Программная часть приложения реализована на языке Python, с использованием библиотеки Tkinter для создания графического интерфейса в среде разработки PyCharm, библиотеки psycopg2 для работы с базой данных, а также библиотеки bcrypt для безопасного хранения паролей пользователей.

Организация коммуникаций внутри компании влияет на общую эффективность. До сих пор многие команды используют звонки, электронную почту, переписки через социальные сети и приложения (WhatsApp, Telegram, Viber, Skype и другие). Однако их функционал ограничен и не может закрыть все потребности рабочих процессов.

Помимо ограниченного функционала, обычные мессенджеры могут мешать нормальным коммуникациям внутри компании. В личном общении каждый предпочитает использовать определённое приложение, поэтому бывает сложно «достучаться» до кого-то с помощью другого чата. Многие не следят за уведомлениями на смартфоне в течение рабочего дня, поэтому взаимодействие усложняется.

Вторая проблема – привычное использование приложений. Подписки на какие-либо каналы или группы, чаты с друзьями, обсуждение семейных вопросов – всё это отвлекает. Кроме того, всегда есть риск отправить какой-то личный файл в рабочую переписку.

Корпоративный мессенджер позволяет решать гораздо больше оперативных задач, нежели обычные чаты. Высокий уровень защищённости, удобный обмен файлами – всё это помогает лучше организовать взаимодействие между коллегами, упростить процесс согласования или принятия решений и повысить продуктивность, поэтому отличным решением является разработка специального приложения и набором функций, которые необходимы для конкретного предприятия.

# **Анализ предметной области**

## **Исследование предметной области**

Система корпоративных сообщений с групповыми и индивидуальными чатами разработана как настольное приложение для компьютеров, с графическим пользовательским интерфейсом. Информация о книгах хранится в базе данных.

Языком реализации проекта является Python. Также при разработке использовалась стандартная библиотека Tkinter для создания графического интерфейса, библиотека psycopg2 для работы с базой данных, библиотека bcrypt для защищенного хранения паролей.

Целевой операционной системой для использования программы является Windows.

1. **Зарегистрированные пользователи**

Все зарегистрированные пользователи, которые занесены в базу данных имеют определенные характеристики.

К этим характеристикам относятся:

* Id пользователя;
* Имя пользователя;
* Хэш пароля;
* Электронная почта;
* Публичный ключ.

1. **Групповые чаты**

Информация групповых чатах также заносится в базу данных и имеет определенные характеристики.

К этим характеристикам относятся:

* Id чата;
* Название;
* Создатель чата.

1. **Отправленные сообщения**

Все отправленные сообщения, которые занесены в базу данных имеют определенные характеристики.

К этим характеристикам относятся:

* Дата и время отправки в формате UTC;
* Отправитель сообщения;
* Получатель сообщения;
* Текст сообщения или вложение;
* Флаг;
* Метка прочитано ли сообщение.

1. **Функционал программного приложения**

Также при анализе предметной области был выявлен основной функционал программного приложения.

К нему относится:

* Возможность регистрации;
* Возможность авторизации с использование своего логина и пароля;
* Создание группового чата;
* Добавление пользователя в групповой чат;
* Удаление пользователя из группового чата;
* Отправка сообщения в групповой чат;
* Отправка сообщений в индивидуальный чат;
* Возможность изменить или восстановить пароль;
* Возможность просматривать историю сообщений;
* Отправка сообщений с вложениями в групповых и индивидуальных чатах.

## **Обзор существующих в предметной области систем**

Так как информационная система «Сервисный центр» напрямую относиться к CRM (Customer Relationship Management) системам, следут так же учитывать уже существующие стандартные системы и технологии, задействованные в целевом направлении нашей информационной системы.

* + 1. **Microsoft Teams**

В групповом мессенджере можно создавать группы под проекты, делиться файлами и общаться с сотрудниками. Программа предполагает использование ботов, которые оповещают и отвечают на запросы.

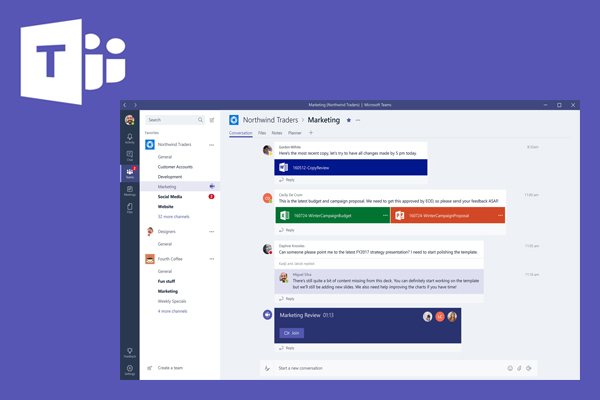


Рис. 4 – Microsoft Teams

Преимущества: тесно интегрирован с приложениями от Майкрософт такими, как SharePoint, Skype, Word, Power BI, Planner и OneNote.

Недостатки:

* скудный бесплатный функционал по сравнению с конкурентами;
* не предусмотрен интегрированный поиск в чатах и файлах.

Минимальная цена за одного пользователя — $5 в месяц. Доступен на всех популярных платформах.

* + 1. **Telegram**

На данный момент Telegram завоевал популярность не только как корпоративный мессенджер, но и медийная площадка. Почти каждый пользователь Telegram подписывается на каналы новостей, анекдотов или развлечений.



Рис. 5 – Telegram

Преимущества:

* Поддержка на всех платформах — наличие мобильной, десктопной и версии для браузера.
* Пересылка больших файлов без «съедания» качества.
* Высокая скорость работы мессенджера.

Недостатки:

* Большое количество контактов.
* Тяжело ориентироваться в обилии чатов.

Как мы упоминали ранее, Telegram функционирует с iOS, Android и ПК. Telegram как корпоративный мессенджер не требует вложений, работая бесплатно.

* + 1. **Slack**

Программа стала популярной благодаря высокому качеству взаимосвязи между клиентами, сотрудниками, партнерами и подрядчиками. В функции Slack входит комментирование отдельных сообщений, показ экрана собеседнику, настройка уведомлений и много других полезностей.

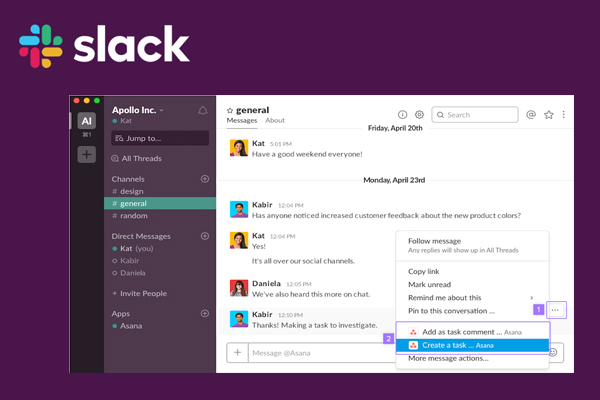


Рис. 6 – Slack

Преимущества:

* быстрая обратная связь;
* наличие тематических каналов внутри группы;
* синхронизация с популярными сервисами (Google Docs, Trello, Google Drive, DropBox и т.д.);
* интеграция с BITbucket Server и JIRA;
* легкий интерфейс.

Недостатки: для полноценной работы нужно покупать лицензию. Минимальная стоимость начинается с $6,67 в месяц за пользователя. Slack предлагает очень ограниченную бесплатную версию.

Оценив реализацию данных проектов и сравнив их возможности для себя выбрали основной функционал, который будет реализован и в нашей информационной системе.

## **Обзор существующих в предметной области технологий**

1. **Tkinter**

Tkinter – это пакет для Python, предназначенный для работы с библиотекой Tk. Библиотека Tk содержит компоненты графического интерфейса пользователя (graphical user interface – GUI), написанные на языке программирования Tcl.

В настоящее время почти все приложения, которые создаются для конечного пользователя, имеют GUI. Редкие программы, подразумевающие взаимодействие с человеком, остаются консольными. В предыдущих двух курсах мы писали только консольные программы.

Существует множество библиотек GUI. Tk далеко не самая популярная, хотя с ее использованием написано не мало проектов. Однако по ряду причин она была выбрана для Python по-умолчанию. Установочный файл Питона обычно уже включает пакет tkinter в составе стандартной библиотеки наряду с другими модулями.

Не вдаваясь в подробности, Tkinter можно охарактеризовать как переводчик с языка Python на язык Tcl. Вы пишете программу на Python, а код модуля tkinter у вас за спиной переводит ваши инструкции на язык Tcl, который понимает библиотека Tk.

Библиотека Tkinter была выбрана мной из-за того, что в отличии от остальных библиотек для работы с графическим интерфейсом, она не содержит в себе ничего лишнего, благодаря чему проста в использовании.

1. **SQL**

Для хранения информации о всех устройствах, поступивших в сервисный центр будет исспльзован SQL. SQL —декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных. При всех своих изменениях SQL остаётся самым распространённым лингвистическим средством для взаимодействия прикладного программного обеспечения с базами данных. В то же время современные СУБД, а также информационные системы, использующие СУБД, предоставляют пользователю развитые средства визуального построения запросов. Несмотря на наличие диалектов и различий в синтаксисе, в большинстве своём тексты SQL-запросов, содержащие DDL и DML, могут быть достаточно легко перенесены из одной СУБД в другую. Существуют системы, разработчики которых изначально ориентировались на применение по меньшей мере нескольких СУБД. Естественно, что при применении некоторых специфичных для реализации возможностей такой переносимости добиться уже очень трудно. С помощью SQL программист описывает только то, какие данные нужно извлечь или модифицировать. То, каким образом это сделать, решает СУБД непосредственно при обработке SQL-запроса. Однако не стоит думать, что это полностью универсальный принцип — программист описывает набор данных для выборки или модификации, однако ему при этом полезно представлять, как СУБД будет разбирать текст его запроса. Чем сложнее сконструирован запрос, тем больше он допускает вариантов написания, различных по скорости выполнения, но одинаковых по итоговому набору данных.

1. **Python**

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты

1. **Bcrypt**

**Bcrypt** — адаптивная криптографическая хеш-функция формирования ключа, используемая для защищенного хранения паролей. Для защиты от атак с помощью радужных таблиц bcrypt использует соль (salt); кроме того, функция является адаптивной, время её работы легко настраивается и её можно замедлить, чтобы усложнить атаку перебором.

1. **Windows Credentials**

Диспетчер учетных данных (Credential Manager) — предназначен для хранения и управления учетных данных пользователей. Данный механизм позволяет автоматически выполнять вход на компьютер, сайт, почту и другие приложения. Данные хранятся в так называемых хранилищах Windows локальная папка на компьютере в зашифрованном виде.

Все данные в диспетчере учетных данных сгруппированы по трем категориям:

• Учетные данные Windows (Windows Credentials) — это имена и пароли, которые используются для доступа к общим сетевым папкам, веб-сайтам, применяющим интегрированную аутентификацию Windows (Windows Integrated Authentication), а также при подключении к удаленному рабочему столу;

• Учетные данные на основе сертификатов (Certificate-Based Credentials) - предназначены для аутентификации с помощью смарт-карт;

• Общие учетные данные (Generic Credentials) - используются сторонними приложениями, для которых требуется отдельная авторизация с учетными данными, отличными от тех, что применяются для входа в систему. В этом разделе могут храниться практически любые учетные данные, соответствующие стандартам Microsoft.

Также можно заводить новые учетные записи в самом диспетчере. И при новом подключении их не потребуется вводить. В диспетчере учетных данных (Credential Manager), есть возможность архивирования хранилища и его восстановление, например для переноса на другой компьютер или повреждения хранилища.

1. **PostgreSQL**

**PostgreSQL** — свободная объектно-реляционная система управления базами данных (СУБД). PostgreSQL базируется на языке SQL и поддерживает многие из возможностей стандарта SQL:2011.

Сильными сторонами PostgreSQL считаются:

– высокопроизводительные и надёжные механизмы транзакций и репликации;

– расширяемая система встроенных языков программирования: в стандартной поставке поддерживаются PL/pgSQL, PL/Perl, PL/Python и PL/Tcl; дополнительно можно использовать PL/Java, PL/PHP, PL/Py, PL/R, PL/Ruby, PL/Scheme, PL/sh и PL/V8, а также имеется поддержка загрузки модулей расширения на языке C;

– наследование;

– возможность индексирования геометрических объектов и наличие базирующегося на ней расширения PostGIS;

– встроенная поддержка слабоструктурированных данных в формате JSON с возможностью их индексации;

– расширяемость (возможность создавать новые типы данных, типы индексов, языки программирования, модули расширения, подключать любые внешние источники данных).

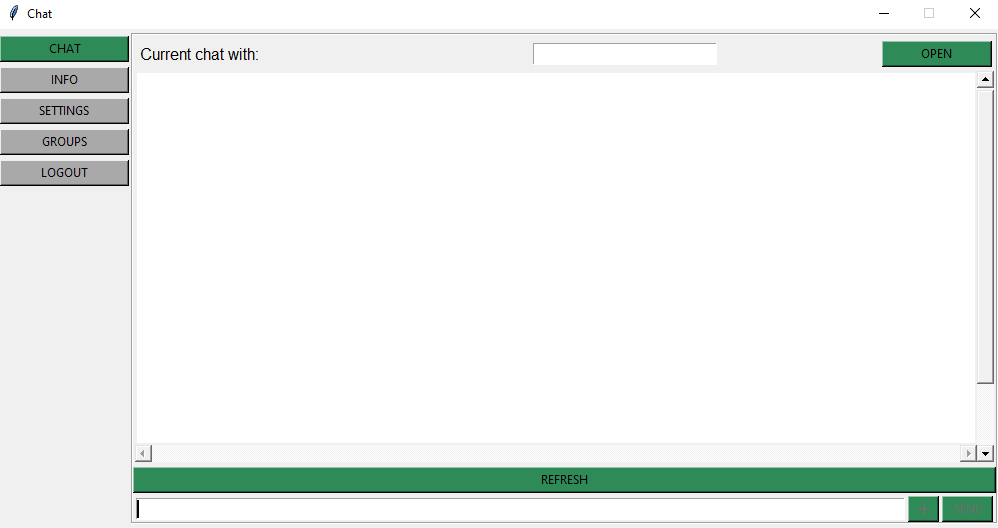
## **1.4. Вывод к первой главе**

В результате анализа предметной области и структур данных были выбраны программные библиотеки, с помощью которых происходит дальнейшая разработка приложения, а также была построена структура всего проекта.

# **Проектирование приложения**

## **Разработка пользовательского интерфейса**

Всего приложение содержит одно окно, в котором присутствует несколько вкладок. Графический интерфейс проектировался с целью удобства и понятности использования. Пользовательский интерфейс состоит из четырех вкладок для переключения между чатами, а также меню настроек, из двух элементов canvas для отображения индивидуальных и групповых сообщений.

Рис. 7 – Пользовательский интерфейс приложения

## **Описание структуры программы**

Логическая структура программы подразумевает использование одной из парадигм программирования – процедурного программирования.

Процедурный стиль программирования предназначен для различного рода дополнительных функций или операций, где производятся лишь видоизменения и преобразования над уже созданными объектами классов и переменными. ПП позволяет сохранить простоту читаемости кода, когда необходимо создать функции, внесение которых в конкретный класс неуместно, а создание нового усложняет читаемость.

## **Алгоритмы, используемые вы программе**

Основными алгоритмами данного приложения являются операции регистрации, авторизации, отправки и принятия сообщений, а также вложений, создания групповых чатов, приглашение участников в групповые чаты. Также к основным алгоритмам относятся:

* Возможность изменения пароля;
* Возможность восстановления пароля;
* Изменение частоты получения обновлений.

## **Проектирование системы хранения**

Во многих проектах возникает потребность в хранении немалого объема данных, но в то же время использование СУБД является слишком накладным из-за сложности развертывания приложения. И тут на помощь приходит такая вещь как PostgreSQL – высокопроизводительная база данных с которой работает и наше приложение.

Данные о зарегистрированных пользователях (Id пользователя, имя пользователя, хэш пароля, электронная почта, публичный ключ) и обо всех отправленных сообщениях (дата и время, отправитель, получатель, текст сообщения, флаг) хранятся в базе данных, также там содержится информация о существующих групповых чатах (Id чата, название, создатель).

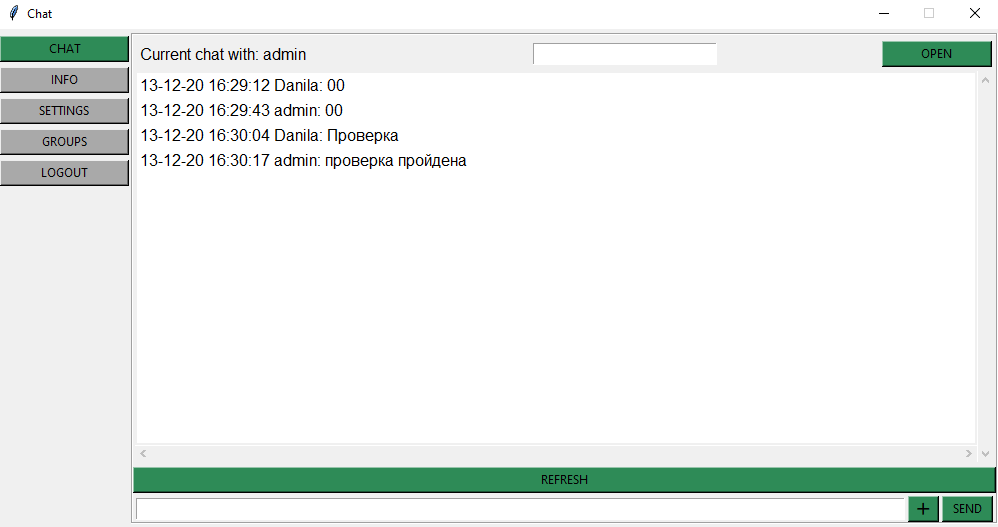


Рис. 8 – Пример вывода сообщений из базы данных на

пользовательский интерфейс.

## **Проектирование программного приложения**

Во время разработки следует разбить программное обеспечение на несколько областей по смыслу, для удобства пользования.

Пользовательский ввод должно проверяться на основе допустимых значений и выводить ошибку если что-то пошло не так.

Для того чтобы точнее проинформировать пользователя о некорректном вводе, следует разбить одну большую проверку, на несколько меньших, что позволит увеличить точность отображаемых сообщений с предупреждением об ошибке ввода.

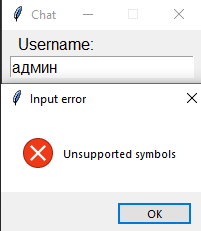


Рис. 9 – Окно ошибки при вводе неподдерживаемых символов в поле логина

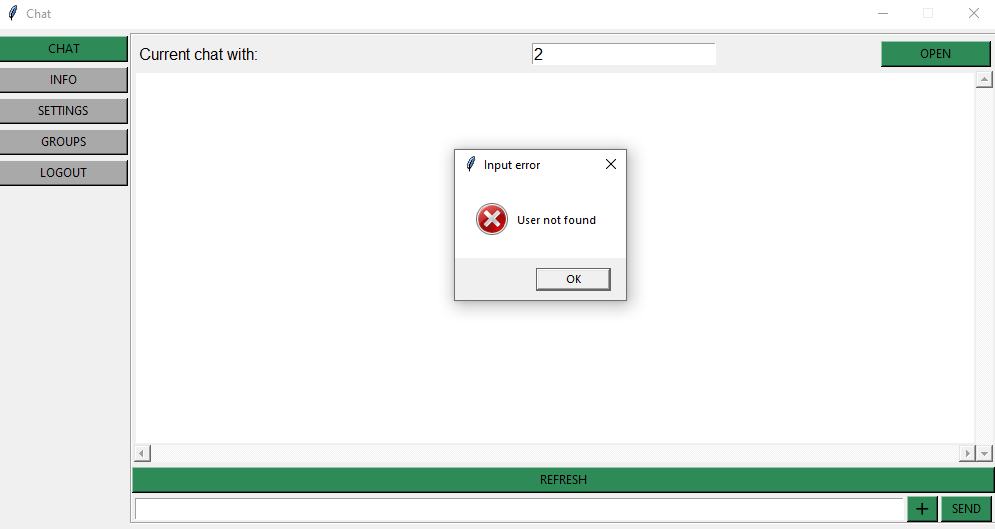


Рис. 10 – Окно ошибки при некорректном id пользователя

# **Реализация приложения**

## **Реализация алгоритмов**

**Регистрация пользователя**

Функция регистрации пользователя получает на вход логин, пароль и тип пользователя, далее программа проверяет зарегистрирован ли пользователь с введенным логином, если пользователь отсутствует, программа заносит информацию о новом пользователе в базу данных.

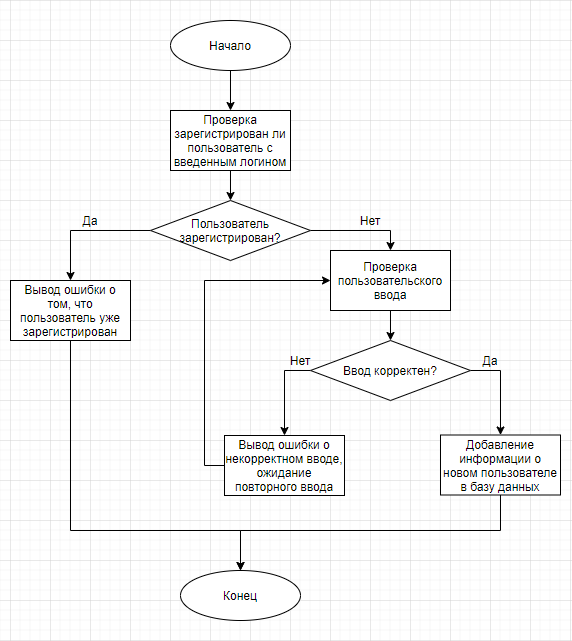


Рис. 11 – Графическое изображение процесса регистрации пользователя

**Авторизация пользователя**

Функция авторизации пользователя получает на вход логин и пароль, далее программа проверяет зарегистрирован ли пользователь с введенным логином, если пользователь существует, программа проверяет правильность ввода пароля, если пароль введен верно, программа разрешает доступ к библиотеке.

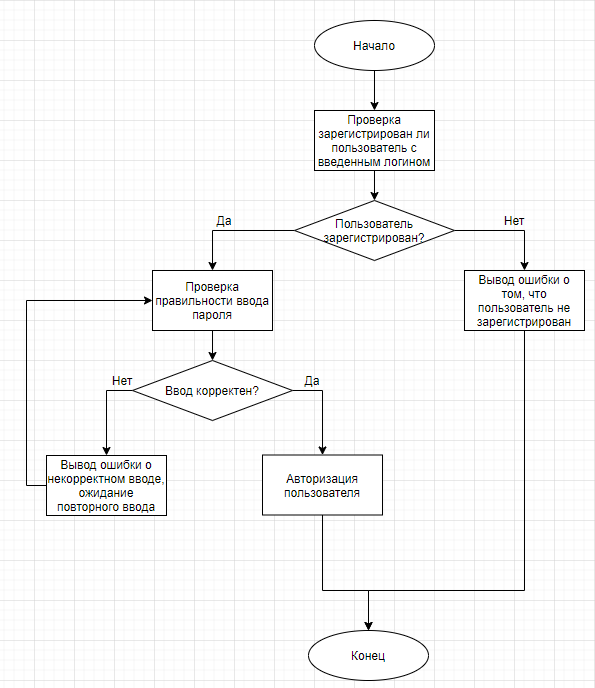
****

Рис. 12 – Графическое изображение процесса авторизации пользователя

**Создание группового чата**

Функция создания группового чата получает на вход название, далее программа проверяет свободно ли название, если группа с таким названием существует, программа выводит ошибку, о том, что необходимо выбрать другое название, если название свободно, программа создает групповой чат с введенным названием.

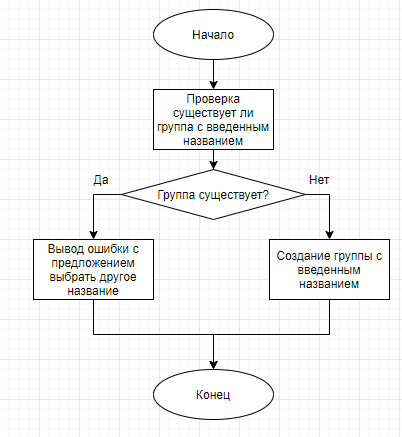


Рис. 13 – Графическое изображение процесса создания группы

**Отправка сообщения**

Функция отправки сообщения в групповой или индивидуальный чат получает на вход текст сообщения или вложение и уникальный идентификатор чата, далее программа получает публичный ключ получателя, с его помощью шифрует сообщение, когда процесс завершается, происходит процесс занесения даты и времени, id получателя, id отправителя, а также теста сообщения в базу данных.

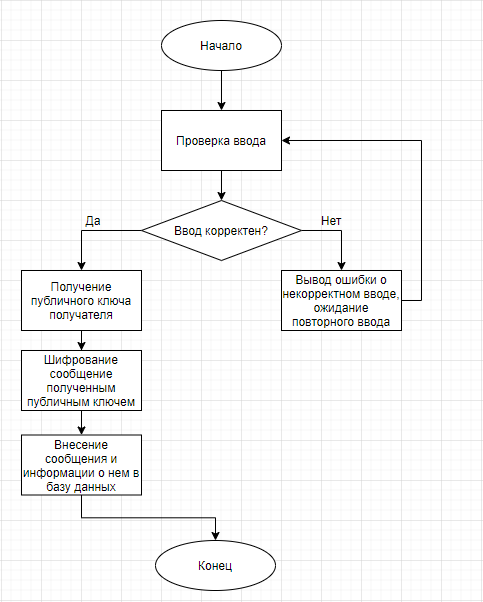


Рис. 14 – Графическое изображение процесса отправки сообщения

**Проверка ввода**

Функция проверки ввода является одной из основных функций программы, она обеспечивает проверку ввода пользователя, на вход функции поступает строка (логин, пароль, сообщение), далее программа получает номер в таблице юникода для каждого символа строки, если этот номер не входит в диапазон допустимых символов, выдается ошибка о некорректном вводе.

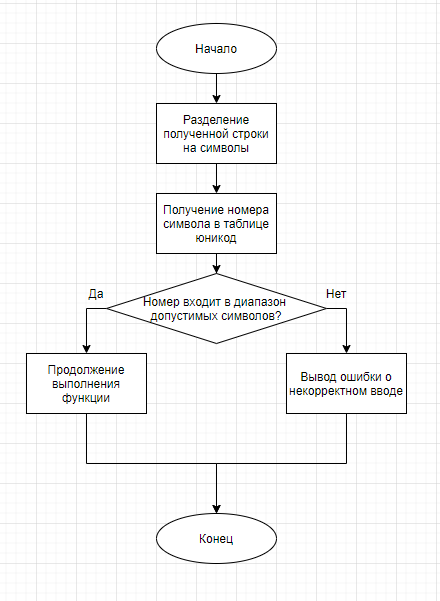


Рис. 15 – Графическое изображение процесса проверки ввода

## **Реализация программного приложения**

У каждого поля свой тип данных, и что бы программа не прекращала работу при некорректном пользовательском вводе, будет обработана и выведена ошибка, которая сообщит, что введенные данные некорректны, и вы сможете повторить ввод.

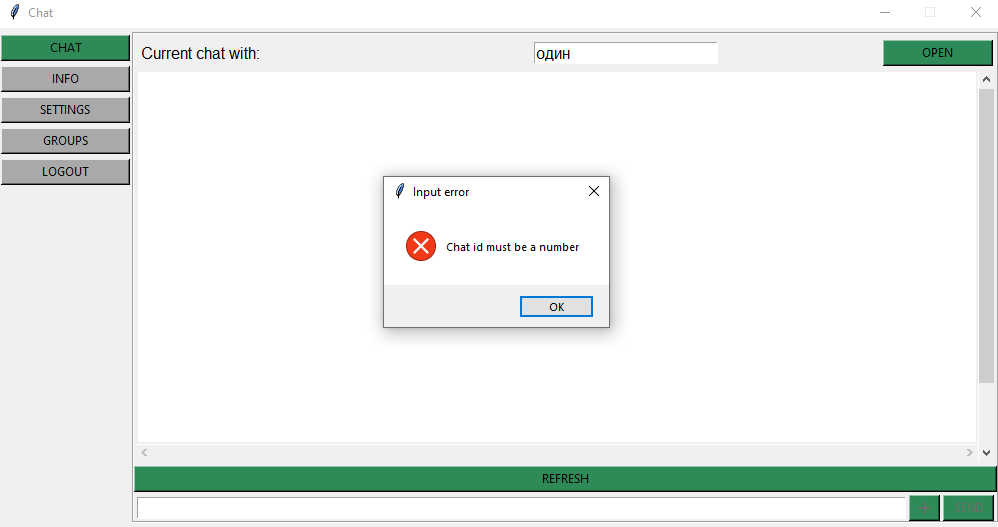


Рис. 16 – Окно ошибки при некорректном вводе id чата

При отправке сообщения необходимо выбрать чат и заполнить поле для текста сообщения, в случае если эти условия не были выполнены будет выведена соответствующая ошибка.

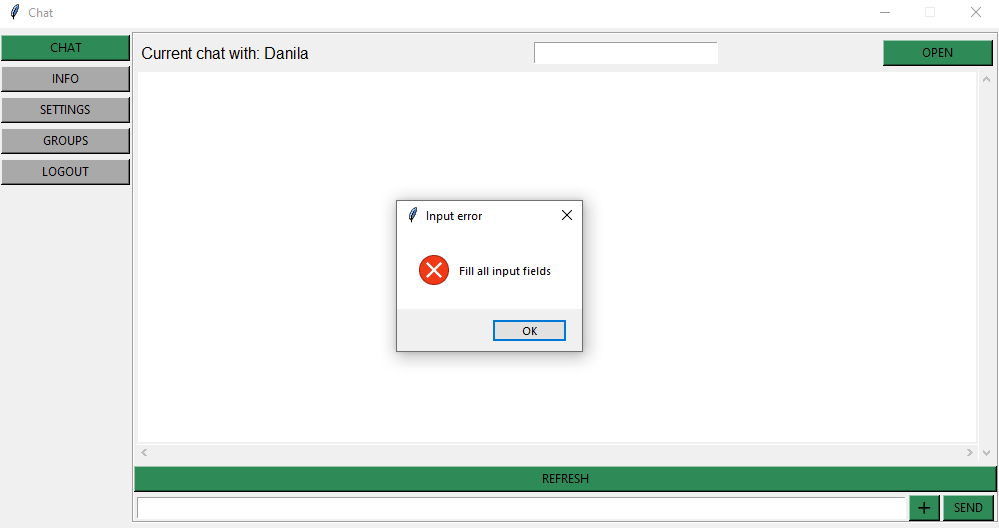


Рис. 17 – Окно ошибки при неполном заполнении

Если чат выбран и поле для ввода текста сообщения заполнено корректно, сообщение будет отправлено.

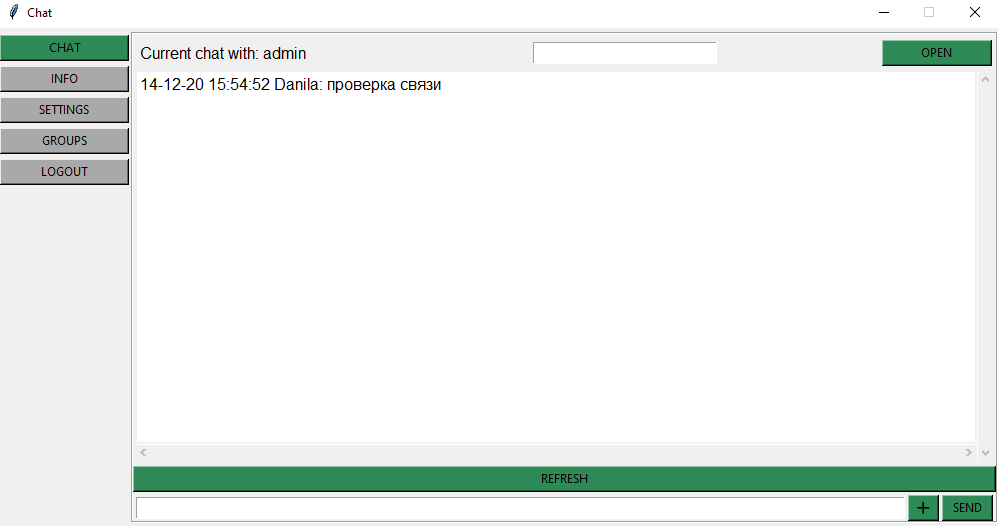


Рис. 18 – Корректная отправка сообщения

При попытке открытия чата с незарегистрированным пользователем, будет выведена ошибка.

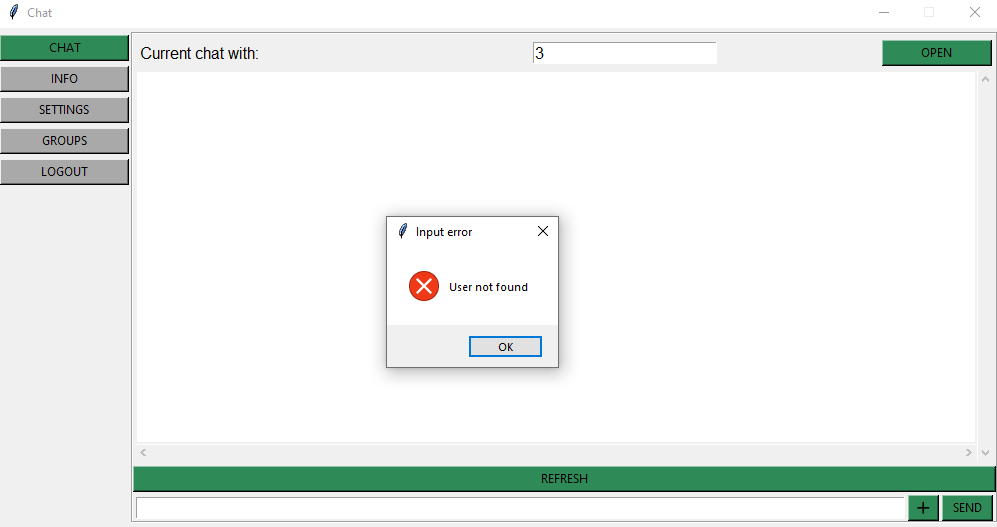


Рис. 19 – Окно ошибки при вводе несуществующего id

Для создания чата необходимо перейти во вкладку «SETTINGS» и ввести желаемое название для группового чата, если группы с таким название не существует она будет создана и ее можно будет увидеть во вкладке «GROUPS».

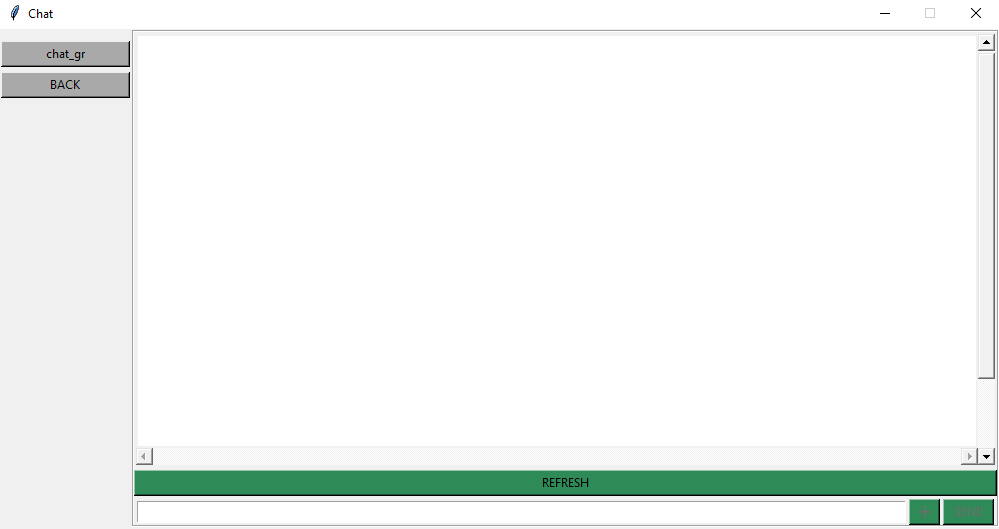


Рис. 20 – Вкладка «GROUPS» с созданной группой

В случае если пользователю необходимо сменить пароль, он должен перейти во вкладку «SETTINGS» и ввести пароль, который установлен на данный момент, а также новый пароль, если старый пароль некорректен, а новый не удовлетворяет требованиям, будет выведена ошибка.



Рис. 21 – Попытка изменения пароля

Для отправки сообщения в групповой чат, пользователю необходимо перейти на вкладку «GROUPS», на ней будет отображен список всех групп, в которых состоит пользователь, далее требуется выбрать группу (выбранная группа будет подсвечена зеленым), которую будет отправлено сообщение и ввести текст.

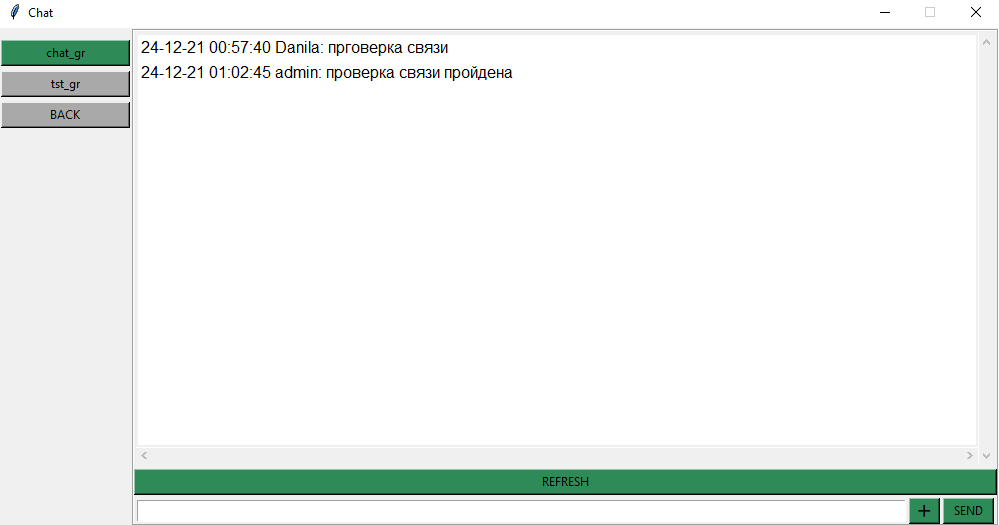


Рис. 22 – Отправка сообщения в групповой чат

## **Вывод к третьей главе**

В результате работы над проектной частью был спроектирован пользовательский интерфейс проекта.

# **Тестирование приложения**

## **Тестирование программного приложения**

Целью проведения испытаний является:

* проверка работоспособности функций программного продукта;
* проверка надежности функционирования программного продукта;
* проверка соответствия заявленным характеристикам и требованиям, изложенным в техническом задании;
* проверка соответствия фактического результата работы программы с теоретическим.

|  |  |  |  |
| --- | --- | --- | --- |
| Входные  данные | Ожидаемый результат | Фактический результат | Статус теста |
| Пустое поле ввода теста сообщения | Сообщение о том, что все поля должны быть заполнены  (рис 17) | Сообщение о том, что все поля должны быть заполнены  (рис 17) | Пройден |
| Тестовое сообщение | Отправка сообщения | Отправка сообщения | Пройден |
| Ввод свободного названия для группы | Создание группы | Создание группы | Пройден |
| Попытка открыть чат с незарегистрированным пользователем | Сообщение о том, что пользователь не найден | Сообщение о том, что пользователь не найден | Пройден |
| Попытка замены пароля с введением неверного пароля | Сообщение о том, что текущий пароль введен неверно | Сообщение о том, что текущий пароль введен неверно | Пройден |
| Попытка авторизации с введением корректных данных | Авторизация пользователя | Авторизация пользователя | Пройден |
| Создание группы с введением уже существующего названия | Сообщение о том, что группа с таким названием уже существует  (рис 18) | Сообщение о том, что группа с таким названием уже существует  (рис 18) | Пройден |
| Регистрация пользователя с введением корректных данных | Успешная регистрация пользователя | Успешная регистрация пользователя | Пройден |
| Выбор группы и введение корректного текста сообщения | Отправка группового сообщения | Отправка группового сообщения | Пройден |
| Попытка открыть чат с пользователем, который зарегистрирован | Успешное открытие чата | Успешное открытие чата | Пройден |
| Приглашение зарегистрированного пользователя в существующую группу | Успешное добавление пользователя в группу | Успешное добавление пользователя в группу | Пройден |
| Попытка регистрации с существующим именем пользователя | Сообщение о том, что пользователь с таким именем уже зарегистрирован | Сообщение о том, что пользователь с таким именем уже зарегистрирован | Пройден |
| Авторизация с введение некорректного пароля | Сообщение о том, что пароль введен неверно, и предложение восстановить его | Сообщение о том, что пароль введен неверно, и предложение восстановить его | Пройден |

Демонстрация некоторых ошибок

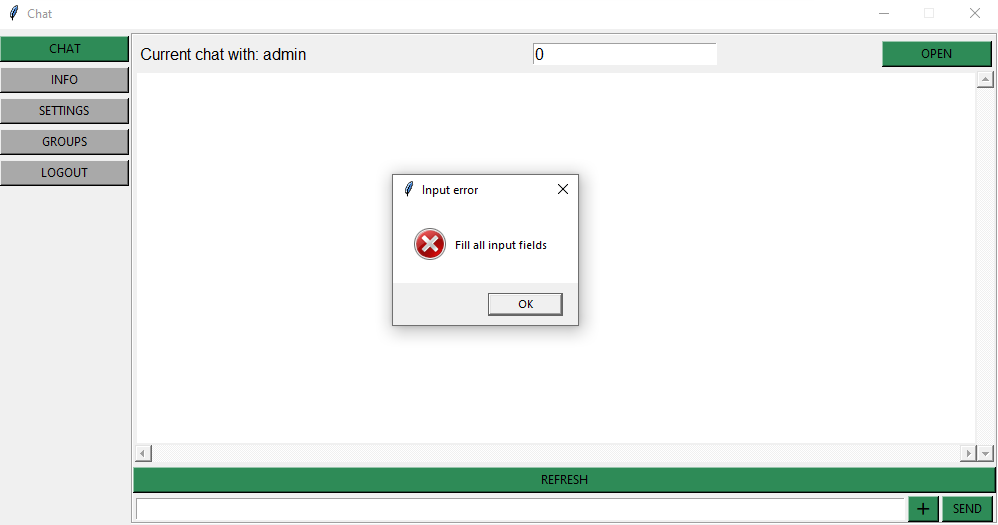


Рис. 23 – Ошибка при не заполнении требуемых полей

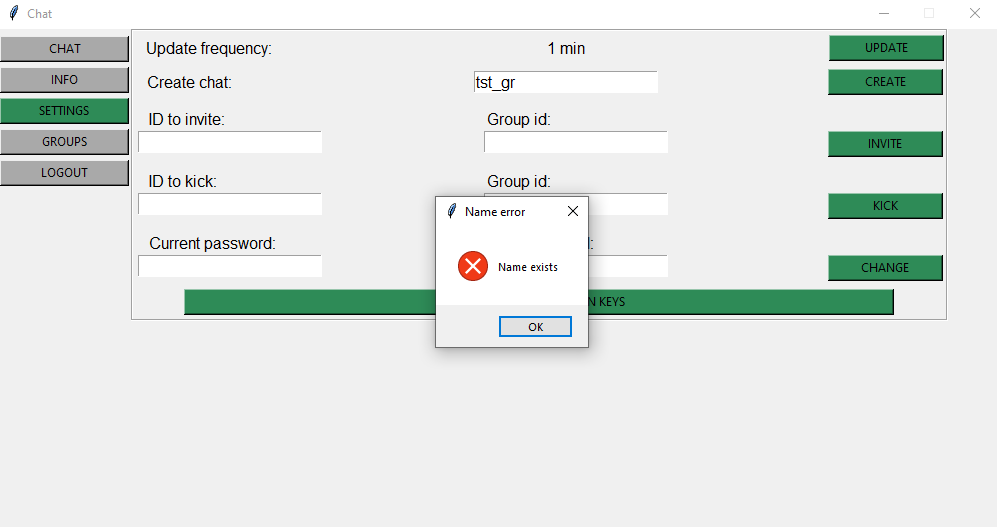


Рис. 24 – Ошибка при создании группового чата с существующим названием

Тестовые запуски показали, что программное приложение:

* Работоспособно;
* Удобно в использовании;
* Устойчиво к некорректному вводу пользователя;
* Интуитивно понятно, благодаря простому интерфейсу программы;
* Приложение выполнено полностью в соответствии с техническим заданием.

# **Инструкция**

## **Инструкция для пользователя**

**Регистрация:**

1. Заполнить поля необходимые для регистрации. Поля которые требуется заполнять показаны на рисунке 25.

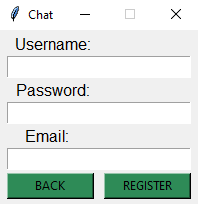


Рис. 25 — Поля заполнения для регистрации

1. Нажать на кнопку «REGISTER». После регистрации у пользователя появится возможность авторизоваться под своими учетными данными.

**Авторизация:**

1. Заполнить поля, необходимые для авторизации, как показано на рисунке 26.

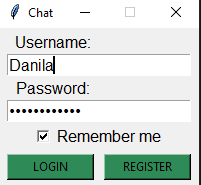


Рис. 26 — Поля заполнения для авторизации

1. Нажать на кнопку «LOGIN», которая изображена на рисунке 27.



Рис. 27 — Кнопка авторизации

**Отправка сообщения:**

1. Ввести Id пользователя, как показано на рисунке 28.



Рис. 28 — Кнопки сортировки списка книг

1. Нажать кнопку «OPEN».
2. Ввести текст в поле для ввода сообщения, которое изображено на рисунке 29 и нажать кнопку «SEND».



Рис. 29 — Кнопка авторизации

**Создание группового чата:**

1. Перейти на вкладку «SETTINGS», которая показана на рисунке 30, после чего в правом окне программы ввести название группы и нажать кнопку «CREATE», как на рисунке 31.

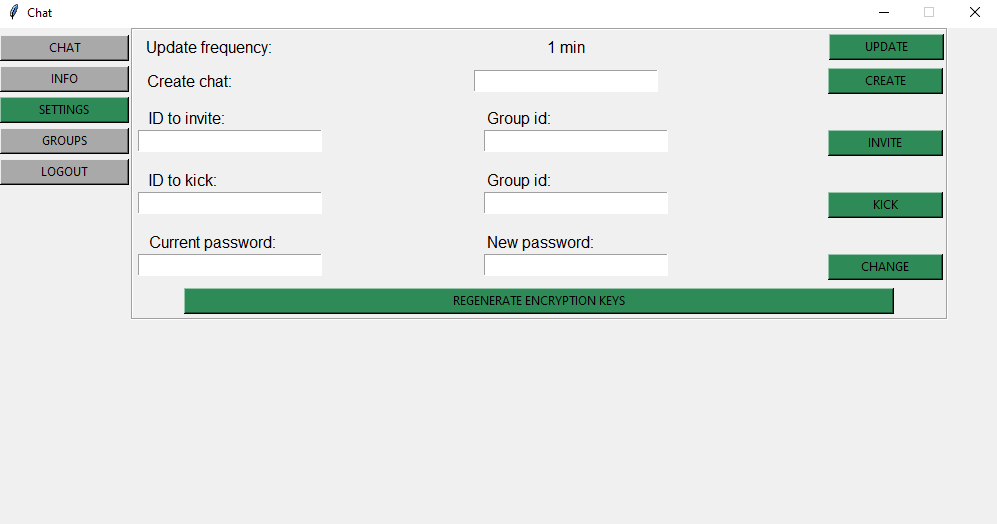


Рис. 30— Вкладка «SETTINGS»



Рис. 31— Поле заполнения для создания группы

**Приглашение пользователя в групповой чат:**

1. Перейти на вкладку «SETTINGS», которая показана на рисунке 30, после чего в правом окне программы ввести id группы, id пользователя и нажать кнопку «INVITE», как на рисунке 32.

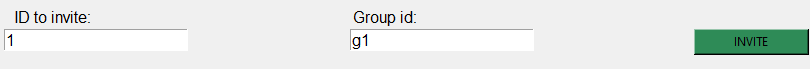


Рис. 32 — Поля заполнения для приглашения пользователя

1. После приглашения пользователя, он сможет видеть группу во вкладке «GROUPS», как показано на рисунке 33.

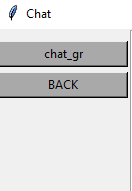


Рис. 33 — Вкладка «GROUPS»

**Отправка сообщения в групповой чат:**

1. Перейти на вкладку «GROUPS», которая показана на рисунке 33, после чего в левом окне программы будет выведен список всех групп в которых состоит пользователь, необходимо выбрать одну из них. Выбранная группа будет подсвечена зеленым и подгружена история сообщений данного чата, как на рисунке 34.

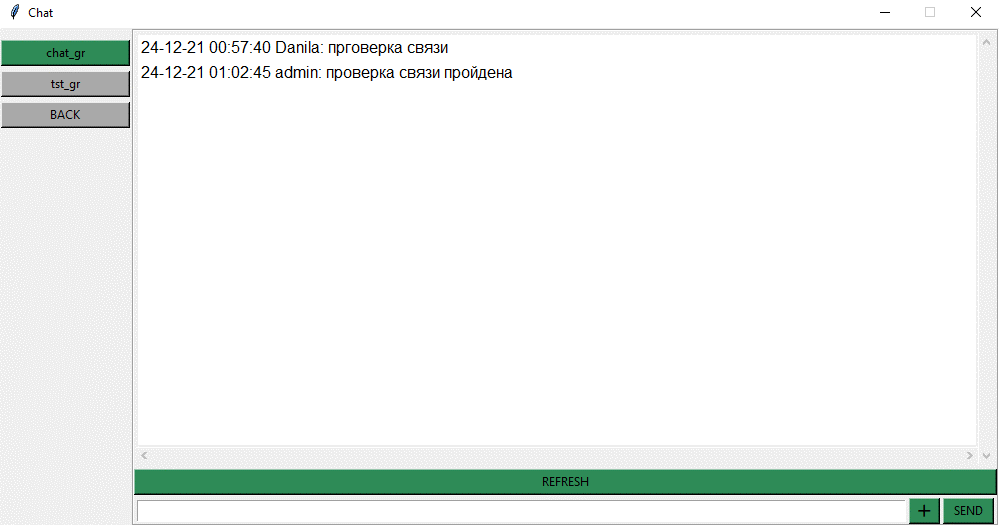


Рис. 34 — Выбор чата

1. Далее необходимо ввести текст сообщения в поле для ввода и нажать на кнопку «SEND», что показано на рисунке 35.



Рис. 35 — Отправка сообщения в групповой чат

**Отправка вложения в групповой или индивидуальный чат:**

1. Находясь в индивидуальном или групповом чате, необходимо нажать на кнопку, изображенную на рисунке 36.



Рис. 36 — Кнопка отправки вложения

1. Далее в открывшемся окне выбрать файл для отправки, как показано на рисунке 37.

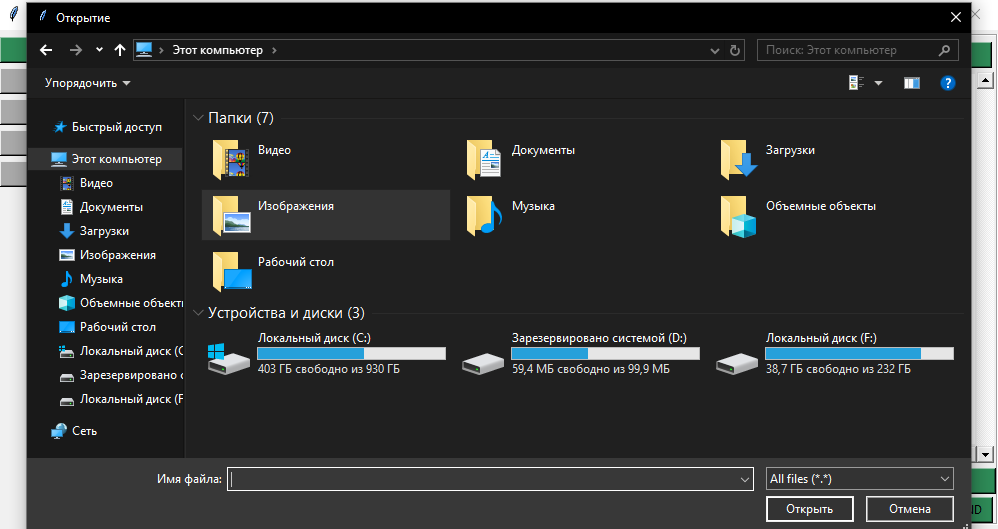


Рис. 37 — Выбор файла для отправки

# **Заключение**

В результате выполнения курсовой работы мы имеем протестированную и исправно работающую систему корпоративных сообщений с групповыми и индивидуальными чатами.

Реализованное приложение имеет следующие функциональные возможности:

* Возможность регистрации;
* Возможность авторизации с использование своего логина и пароля;
* Создание рабочей группы;
* Добавление/удаление пользователя из рабочей группы;
* Отправка сообщения в рабочую группу;
* Отправка сообщений в индивидуальный чат;
* Возможность изменить или восстановить пароль;
* Возможность просматривать историю сообщений;
* Отправка сообщений с вложениями в рабочих группах и индивидуальных чатах.

В процессе работы над курсовой работой были закреплены навыки:

* Анализа предметной области;
* Работы с базой данных PostgreSQL;
* Разработки на языке Python;
* Проектирования графического интерфейса пользователя c помощью стандартной библиотеки Tkinter;
* Защиты пользовательских паролей;
* Тестирования программного продукта;
* Подготовки документации.

# **Список используемых источников**

Нормативные документы:

1. О введении в действие Инструкции по организации и проведению

курсового проектирования. – М.: РТУ МИРЭА, Приказ №1325 от 05.10.2018. – 17 с.;

Бумажные источники:

1. Mishra R.K., Raman S.R. Introduction to PySpark SQL. - 978-1-4842-4334-3 изд. - California: Apress, 2019.
2. Practical Statistics for Data Scientists Peter Bruce, Andrew Bruce, Peter

Gedeck изд. – 2020

1. The Quick Python Book, Second Edition Manning Publications Co.3

Lewis Street Greenwich, CT -978-1-935182-20-7 изд. - United States 2020

Электронные ресурсы:

1. Руководство по языку программирования Python [Электронный

ресурс]. – URL: <https://metanit.com/python/tutorial/>

1. Руководство по SQL (полная версия) [Электронный ресурс]. – URL:

<https://proselyte.net/tutorials/sql/>

1. PostgreSQL: Документация: 9.4: Справочное руководство [Электронный ресурс]. – URL:

<https://postgrespro.ru/docs/postgresql/9.4/reference>

1. Руководство по Tkinter [Электронный ресурс]. – URL:

<https://docs.python.org/3/library/tkinter.html>

1. Построение блок-схем [Электронный ресурс]. – URL:

<https://www.draw.io>

# **Приложение**

import os

import rsa

import time

import shutil

import qrcode

import bcrypt

import random

import yadisk

import keyring

import smtplib

import psycopg2

import threading

import tkinter as tk

from tkinter import \*

from PIL import Image

from datetime import datetime, timezone

from tkinter import messagebox

from tkinter import filedialog

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

from keyring.backends.Windows import WinVaultKeyring

# from keyring.backends.OS\_X import Keyring

keyring.set\_keyring(WinVaultKeyring())

y = yadisk.YaDisk(token="AgAAAABITC7sAAbGEG8sF3E00UCxjTQXUS5Vu28")

code = None

chats = {}

current\_chat = "-1"

root = tk.Tk()

spacing, spacing\_2 = 0, 0

w = root.winfo\_screenwidth() // 2 - 140

h = root.winfo\_screenheight() // 2 - 100

user\_id, email, user\_login = '', '', ''

var = IntVar()

private\_key = rsa.PrivateKey(1, 2, 3, 4, 5)

files\_dir = 'files'

private\_key\_file = files\_dir + '/priv\_key.PEM'

time\_to\_check = 60.0

db\_log = "register"

db\_pass = "reg"

dt = datetime.now(timezone.utc).astimezone()

utc\_diff = dt.utcoffset()

try:

os.mkdir(files\_dir)

except FileExistsError:

pass

def exception\_handler(e, connect, cursor):

try:

cursor.close()

connect.close()

print(e)

except Exception as e:

print(e)

def pg\_connect():

global db\_log, db\_pass

try:

con = psycopg2.connect(

host="localhost",

database="postgres",

user=db\_log,

port="5432",

password=db\_pass)

cur = con.cursor()

return con, cur

except Exception as e:

print(e)

def auto\_check\_message():

try:

get\_message()

except Exception as e:

print(e)

def debug(cursor):

cursor.execute("SELECT \* FROM chats")

print(cursor.fetchall())

cursor.execute("SELECT table\_name FROM information\_schema.tables WHERE table\_schema NOT IN ("

"'information\_schema', 'pg\_catalog') AND table\_schema IN('public', 'myschema');")

print(cursor.fetchall())

def create\_tables():

connect, cursor = pg\_connect()

try:

# cursor.execute("DROP TABLE messages")

# cursor.execute("DROP TABLE users")

# cursor.execute("DROP TABLE chats")

# debug(cursor)

cursor.execute('CREATE TABLE IF NOT EXISTS users(id INTEGER,'

'login TEXT,'

'password TEXT,'

'pubkey TEXT,'

'email TEXT)')

cursor.execute('CREATE TABLE IF NOT EXISTS chats(id TEXT,'

'name TEXT,'

'owner INTEGER)')

cursor.execute('CREATE TABLE IF NOT EXISTS messages(date TIMESTAMP,'

'from\_id TEXT,'

'to\_id TEXT,'

'message BYTEA,'

'message1 BYTEA,'

'file TEXT,'

'read INTEGER)')

connect.commit()

cursor.close()

connect.close()

except Exception as e:

exception\_handler(e, connect, cursor)

def check\_input(password: str, log: str):

if len(log) < 5:

messagebox.showerror('Input error', 'Login length must be more than 5 characters')

return False

if len(password) < 8:

messagebox.showerror('Input error', 'Password does not meet the requirements')

return False

for i in password:

if ord(i) < 45 or ord(i) > 122:

messagebox.showerror('Input error', 'Unsupported symbols')

return False

for i in log:

if ord(i) < 45 or ord(i) > 122:

messagebox.showerror('Input error', 'Unsupported symbols')

return False

return True