



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра корпоративных информационных систем (КИС)

ОТЧЁТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ
Преддипломная практика

приказ Университета о направлении на практику от «20» апреля 2022 г. № 2410-С

Отчет представлен к
рассмотрению:
Студент группы ИКБО-08-18

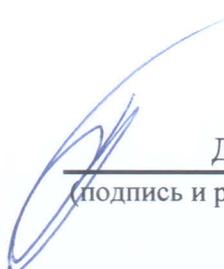
 « » мая 2022

 Валяев Д.А.
(подпись и расшифровка подписи)

Отчет утвержден.
Допущен к защите:

Руководитель практики
от кафедры

 « » мая 2022


Демидова Л.А.
(подпись и расшифровка подписи)

Москва 2022 г.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра корпоративных информационных систем (КИС)

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРОИЗВОДСТВЕННУЮ ПРАКТИКУ
Преддипломная практика

Студенту 4 курса учебной группы ИКБО-08-18
Валяеву Даниле Андреевичу

Место и время практики: РТУ МИРЭА кафедра КИС, с 20 апреля 2022 г. по 17 мая 2022 г.

Должность на практике: студент

1. СОДЕРЖАНИЕ ПРАКТИКИ:

1.1. Изучить: техническую литературу, электронные информационно-образовательные ресурсы, применяемые для профессиональной деятельности по теме дипломной работы

1.2. Практически выполнить: в соответствии с темой дипломной работы:

- описать функционал разрабатываемого программного продукта;
- обосновать выбор средств разработки;
- описать реализацию программного продукта, реализацию БД;
- описать процесс тестирования;
- описать работу пользователя с программным продуктом.

1.3. Ознакомиться: с актуальными нормативно-правовыми документами, международными и отечественными стандартами, с СУБД и инструментальными средствами разработки

2. ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ: нет

3. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ: в процессе практики рекомендуется использовать периодические издания и отраслевую литературу годом издания не старше 5 лет от даты начала прохождения практики

Руководитель практики от
кафедры

Подпись

(Демидова Л.А.)

«20» апреля 2022 г.

Задание получил

Подпись

(Валяев Д.А.)

«20» апреля 2022 г.

СОГЛАСОВАНО:

Заведующий кафедрой:

Подпись

(Андрианова Е.Г.)

«20» апреля 2022 г.

Проведенные инструктажи:

Охрана труда:

Инструктирующий


Подпись

«20» апреля 2022 г.

Карусевич Т.Е., старший преподаватель кафедры КИС

Инструктируемый


Подпись

Валяев Д.А.

Техника безопасности:

Инструктирующий


Подпись

«20» апреля 2022 г.

Карусевич Т.Е., старший преподаватель кафедры КИС

Инструктируемый


Подпись

Валяев Д.А.

Пожарная безопасность:

Инструктирующий


Подпись

«20» апреля 2022 г.

Карусевич Т.Е., старший преподаватель кафедры КИС

Инструктируемый


Подпись

Валяев Д.А.

С правилами внутреннего распорядка ознакомлен:


Подпись

«20» апреля 2022 г.

Валяев Д.А.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

**РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

студента Валяева Д.А. 4 курса группы ИКБО-08-18 очной формы обучения, обучающегося по направлению подготовки 09.03.04 Программная инженерия, профиль «Корпоративные информационные системы».

Неделя	Сроки выполнения	Этап	Отметка о выполнении
1	20.04.2022	Подготовительный этап, включающий в себя организационное собрание (Вводная лекция о порядке организации и прохождения производственной практики, инструктаж по технике безопасности, получение задания на практику)	
1	21.04.2022- 26.04.2022	1) Описание функционала разрабатываемого программного продукта в соответствии с темой дипломной работы; 2) Обоснование выбора средств разработки.	
2	27.04.2022- 03.05.2022	3) Описание этапов реализации программного продукта, реализации базы данных в соответствии с темой дипломной работы.	
3	04.05.2022- 10.05.2022	4) Описание процесса тестирования программного продукта в соответствии с темой дипломной работы; 5) Описание работы пользователя с программным продуктом.	
4	11.05.2022- 14.05.2022	Представление руководителю предварительной версии отчета с обеспечением согласованности материала по всем его частям, полученных на предыдущих этапах	
4	17.05.2022	Подготовка окончательной версии отчета по практике (Оформление материалов отчета в полном соответствии с требованиями на оформление письменных учебных работ студентов)	

Руководитель практики от кафедры

/Демидова Л.А., д.т.н., профессор/

Обучающийся

/Валяев Д.А./

Согласовано:

Заведующий кафедрой

/Андрианова Е.Г., к.т.н., доцент/

Оглавление

Введение.....	1
1. Функционал программного продукта.....	2
2. Выбор средств разработки.....	4
3. Реализация программного продукта, базы данных.....	6
3.1. Разработка пользовательского интерфейса.....	6
3.2. Алгоритмы реализации программного продукта.....	6
3.3. Логическая модель базы данных.....	13
4. Тестирование.....	15
5. Инструкция пользователя.....	18
Заключение.....	21
Список используемых источников.....	22

Введение

В данной производственной практике представлена реализация системы корпоративной коммуникации поддержки реализации проектов группы разработчиков. Разрабатываемая система реализует функции отправки сообщений или вложений в индивидуальные и групповые чаты; создания группового чата; регистрации; авторизации; а также восстановления или изменения пароля.

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.

1. Функционал программного продукта

Назначение системы

Система корпоративной коммуникации поддержки реализации проектов группы разработчиков предназначена для увеличения общей производительности работы и упрощения коммуникации между сотрудниками компании, во время разработки по водопадной модели жизненного цикла.

Система корпоративной коммуникации, включает в себя:

- приложение, устанавливаемое на стороне клиента;
- службу получения уведомлений;
- серверную часть (веб-сервер, файловый сервер, СУБД, БД).

Функционал системы включает в себя:

– использование ролевой модели, где у всех участников рабочей группы равные права, но организатор имеет расширенный набор возможностей (добавление/удаление участников, отправка специальных уведомлений);

– разрабатываемая система должна поддерживать клиент-серверное взаимодействие;

– возможность регистрации/авторизации пользователей (контроль имен пользователя должен осуществляться со стороны организации разработчиков проекта);

– возможность создания рабочей группы;

– возможность управлять (удалять/добавлять) участниками в рабочих группах;

– возможность отправки личных и групповых сообщений (максимальная длина сообщений устанавливается системным администратором), а также файлов (максимальный объем файлов зависит от конфигурации сервера и устанавливается системным администратором);

– возможность создавать/получать специальные уведомления;

- возможность редактировать/удалять данные аккаунта;
- система должна сохранять получаемый контент на стороне клиента с возможностью повторного получения утраченных данных с сервера;
- асинхронное шифрование (на стороне приложения приватный ключ, на серверной стороне публичный ключ);
- приложение должно иметь физическую привязку к устройству пользователя, где был зарегистрирован аккаунт.

Основные задачи системы

- Создание рабочих групп;
- Добавление/удаление участников рабочей группы;
- Получение уведомлений о сообщениях;
- Организация коммуникации в рабочих группах;

Функциональные требования

1. Создание рабочих групп

Предоставление пользователю возможности создания чатов с возможностью использования специальных уведомлений и последующего добавления/удаления участников.

2. Создание аккаунта

Предоставление пользователю возможности создания аккаунта, для последующего использования функций системы.

3. Отправка сообщений

Возможность пользователя отправлять как личные сообщения, так и в рабочих группах.

4. Управление участниками рабочей группы

Возможность организатора рабочей группы добавлять или исключать пользователей.

2. Выбор средств разработки

В качестве языка программирования для реализации как клиентской, так и серверной частей системы был выбран Python версии 3.7 из-за простоты и скорости разработки, а также из-за большого количество поддерживаемых библиотек, используемых при разработке системы [1].

Для разработки клиентской части системы были использованы библиотеки:

- Tkinter – стандартная библиотека Python, предназначенная для создания приложений с графическим пользовательским интерфейсом [4];

- bcrypt – библиотека, позволяющая создать хэш пароля. Поскольку хэш-функция является односторонней, в случае несанкционированного доступа к базе данных злоумышленники не смогут получить пароли пользователей;

- Keyring – библиотека для работы с диспетчером учетных данных (Credential Manager), который предназначен для хранения и управления учетными данными пользователей в операционной системе;

- Rsa – библиотека для реализации одноименного асимметричного алгоритма шифрования, который используется в разрабатываемой системе для шифрования сообщений пользователей. Асимметричный алгоритм подразумевает использование двух ключей: публичного для шифрования (хранится в базе данных и запрашивается пользователями, которые собираются отправить сообщение) и приватного для расшифровки (хранится только у пользователя).

Для разработки серверной части системы были использованы библиотеки:

- FastAPI – фреймворк для создания достаточно быстрых HTTP API-серверов со встроенными валидацией, сериализацией и асинхронностью, существенным плюсом использования данного фреймворка является его простота использования и скорость разработки.

Хранение данных на сервере реализовано с использованием СУБД, в качестве которой выбрана PostgreSQL, распространяемая по лицензии PostgreSQL License, которая предоставляет права на бесплатное использование. Для работы с PostgreSQL из Python была использована библиотека `psycopg2` [3].

3. Реализация программного продукта, базы данных

3.1. Разработка пользовательского интерфейса

Пользовательское приложение содержит одно окно, на котором присутствует несколько вкладок (список всех чатов, настройки, поиск пользователей). Графический интерфейс проектировался с целью удобства и интуитивности использования и изображен на рисунке 3.1.

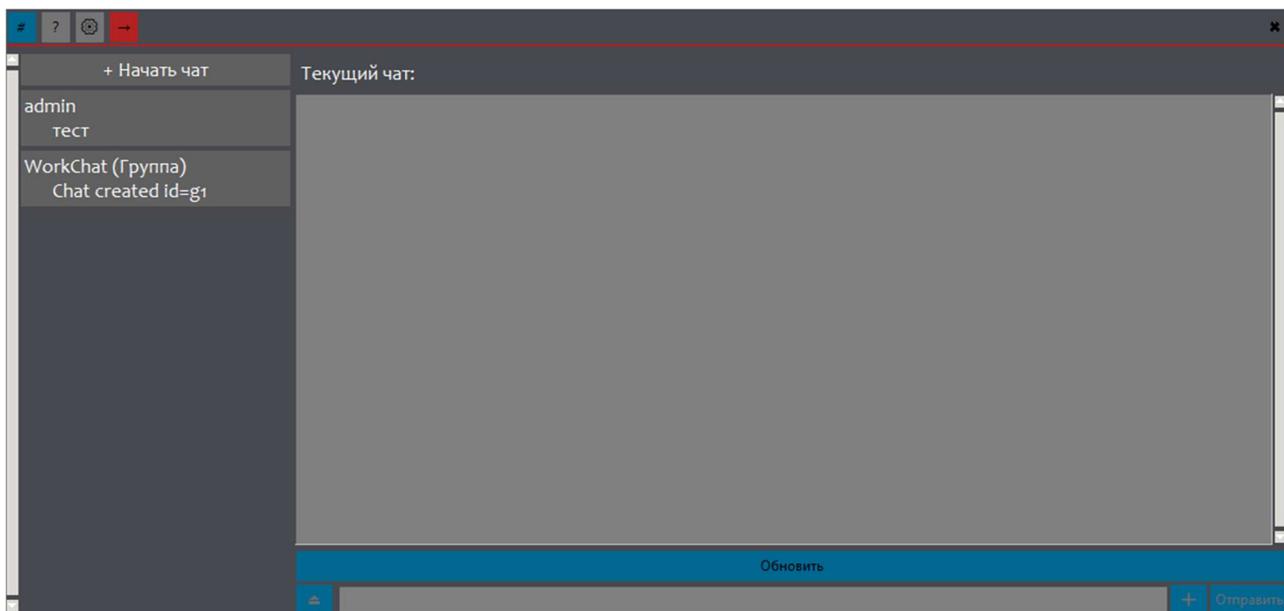


Рисунок 3.1 – Графический интерфейс пользовательского приложения

3.2. Алгоритмы реализации программного продукта

Основными алгоритмами разрабатываемой системы являются операции регистрации, авторизации, отправки и принятия сообщений, создания рабочих групп, приглашение и удаление участников из рабочих групп. Также к основным алгоритмам относятся:

- изменение/восстановление пароля;
- получение специальных уведомлений от организатора рабочей группы [5].

Регистрация пользователя

Функция регистрации пользователя получает на вход логин, пароль и почту пользователя, далее проверяет корректность ввода и производится

отправка запроса на веб-сервер, где проверяется зарегистрирован ли пользователь с введенным логином, если пользователь отсутствует, веб-сервер заносит информацию о новом пользователе в базу данных, графическое изображение процесса регистрации пользователя представлено на рисунке 3.2.

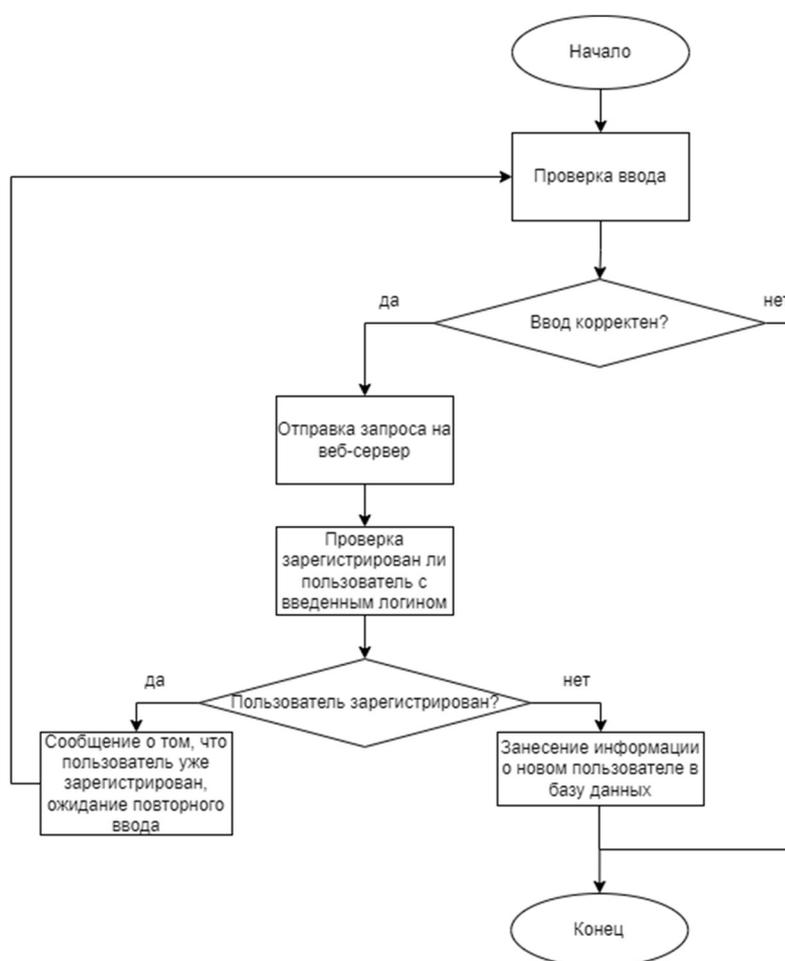


Рисунок 3.2 – Графическое изображение процесса регистрации пользователя

Авторизация пользователя

Функция авторизации пользователя получает на вход логин и пароль пользователя, далее приложение отправляет запрос на веб-сервер, который проверяет зарегистрирован ли пользователь с введенным логином, если пользователь существует, веб-сервер проверяет правильность ввода пароля посредством сравнения полученного пароля с хэшем, записанным в базу

данных, если пароль введен верно, веб-сервер генерирует токен и отправляет обратно в приложение, графическое изображение процесса авторизации пользователя представлено на рисунке 3.3.

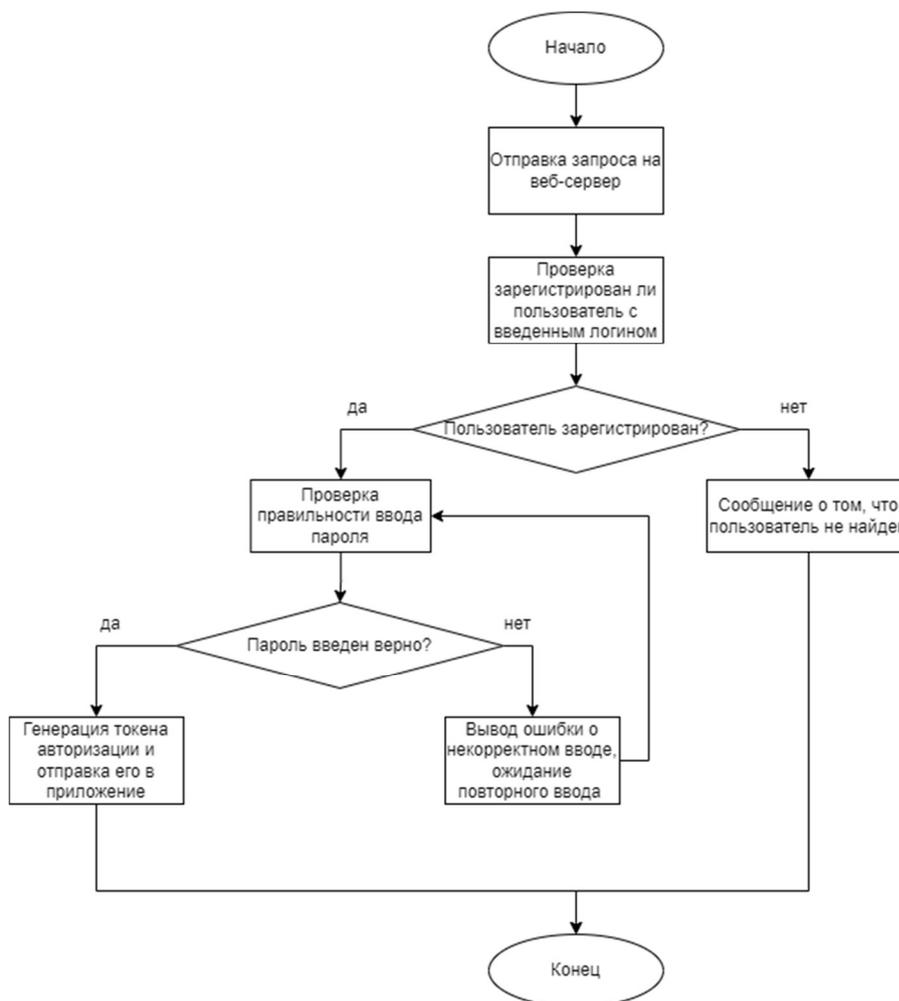


Рисунок 3.3 – Графическое изображение процесса авторизации пользователя

Создание группового чата

Функция создания группового чата получает на вход название, далее приложение проверяет корректность ввода и отправляет запрос на веб-сервер, для проверки существования рабочей группы, если группа с таким названием существует, программа выводит ошибку, о необходимости выбрать другое название, если новое название свободно, веб-сервер записывает информацию о созданной рабочей группе в БД, графическое

изображение процесса создания рабочей группы представлено на рисунке 3.4.

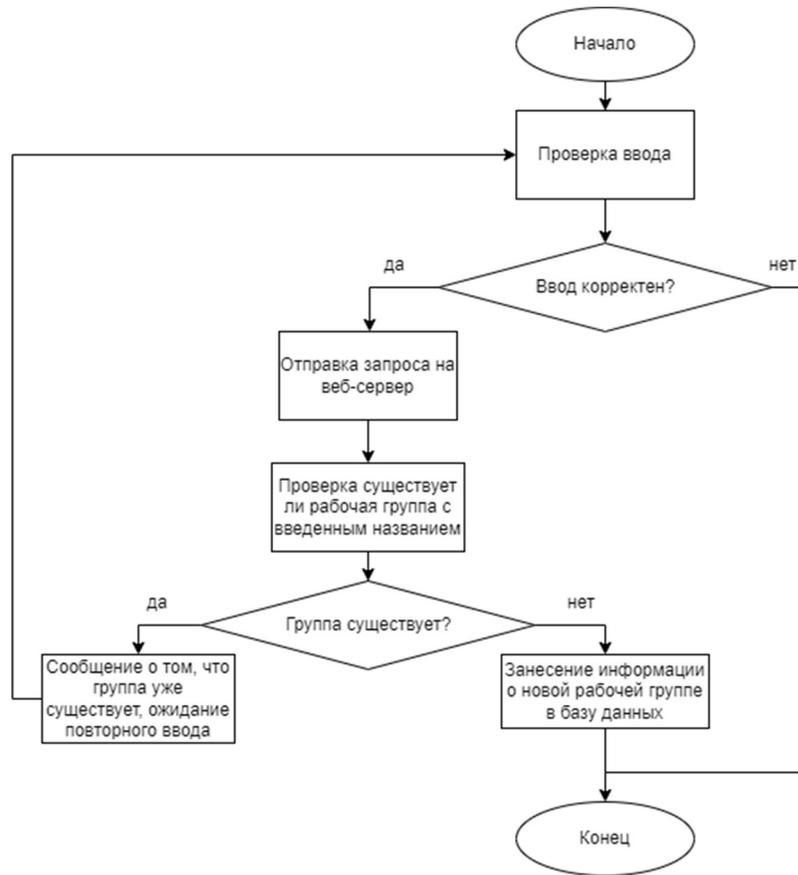


Рисунок 3.4 – Графическое изображение процесса создания рабочей группы

Приглашение участников в рабочую группу

Функция приглашения участников в рабочую группу принимает на вход id пользователя, далее отправляет запрос на веб-сервер, где проверяется зарегистрирован ли пользователь, если пользователь зарегистрирован происходит проверка нахождения пользователя в рабочей группе, если пользователь еще не был добавлен, происходит его добавление, графическое изображение процесса приглашения участника в рабочую группу представлено на рисунке 3.5.

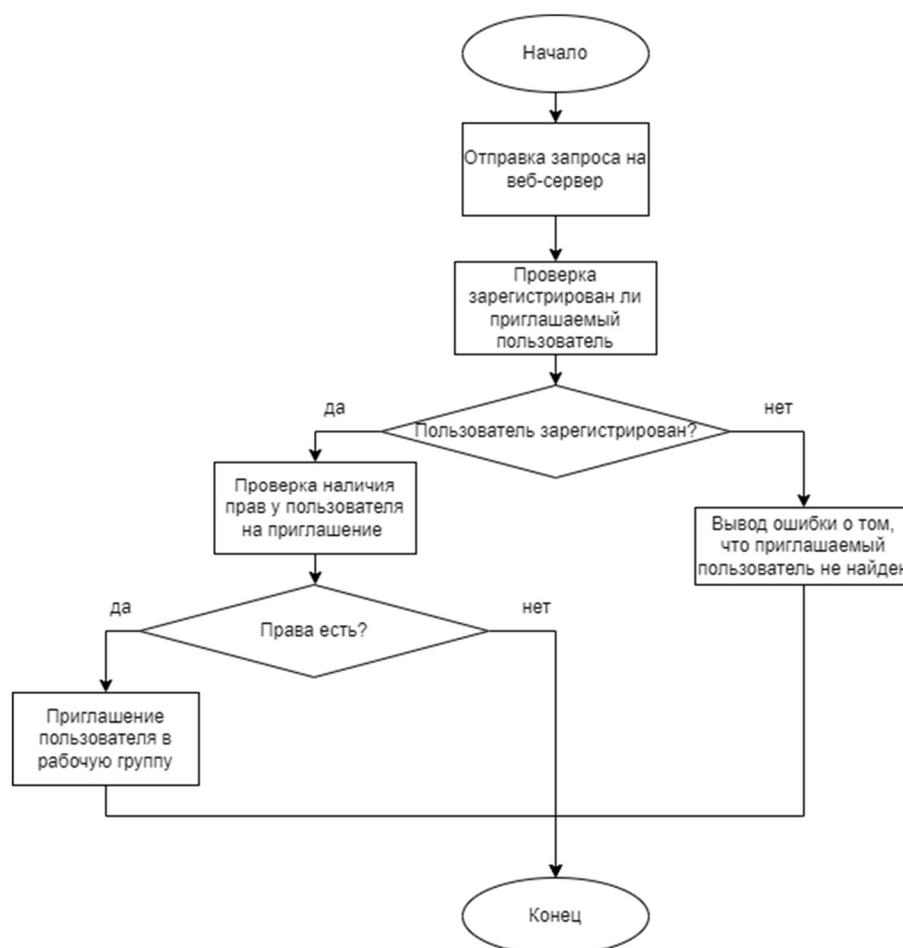


Рисунок 3.5 – Графическое изображение процесса приглашения участника в рабочую группу

Отправка сообщения

Функция отправки сообщения в рабочую группу или личную беседу получает на вход текст сообщения или вложение, далее приложение запрашивает с веб-сервера публичный ключ получателя/получателей, с его помощью шифрует сообщение, когда процесс шифрования завершается, происходит отправка зашифрованного сообщения на сервер, где данные о сообщении (id сообщения, id получателя, id отправителя, дата и время в формате utc, сообщение в зашифрованном виде) записываются в базу данных, графическое изображение процесса отправки сообщения представлено на рисунке 3.6.



Рисунок 3.6 – Графическое изображение процесса отправки сообщения

Изменение пароля

Функция изменения пароля получает на вход старый и новый пароли, далее клиентское приложение проверяет корректность нового пароля, если пароль соответствует требованиям, происходит отправка запроса на веб-сервер, где старый пароль сверяется с хэшем, записанным в базу данных, если пароль подошел данные пользователя обновляются.

Получение специальных уведомлений от организатора рабочей группы

Для получения специальных уведомлений разработана служба Windows, устанавливаемая при первом запуске клиентского приложения. При запуске операционной системы запускается и служба оповещений, которая проверяет включена ли функция получения уведомлений у

пользователя, если включена, служба считывает необходимую для ее работы информацию (адрес веб-сервера и список рабочих групп, от которых необходимо получать уведомления) из папки «AppData», которая была записана клиентским приложением при первом запуске, графическое изображение процесса получения специальных уведомлений представлено на рисунке 3.7.

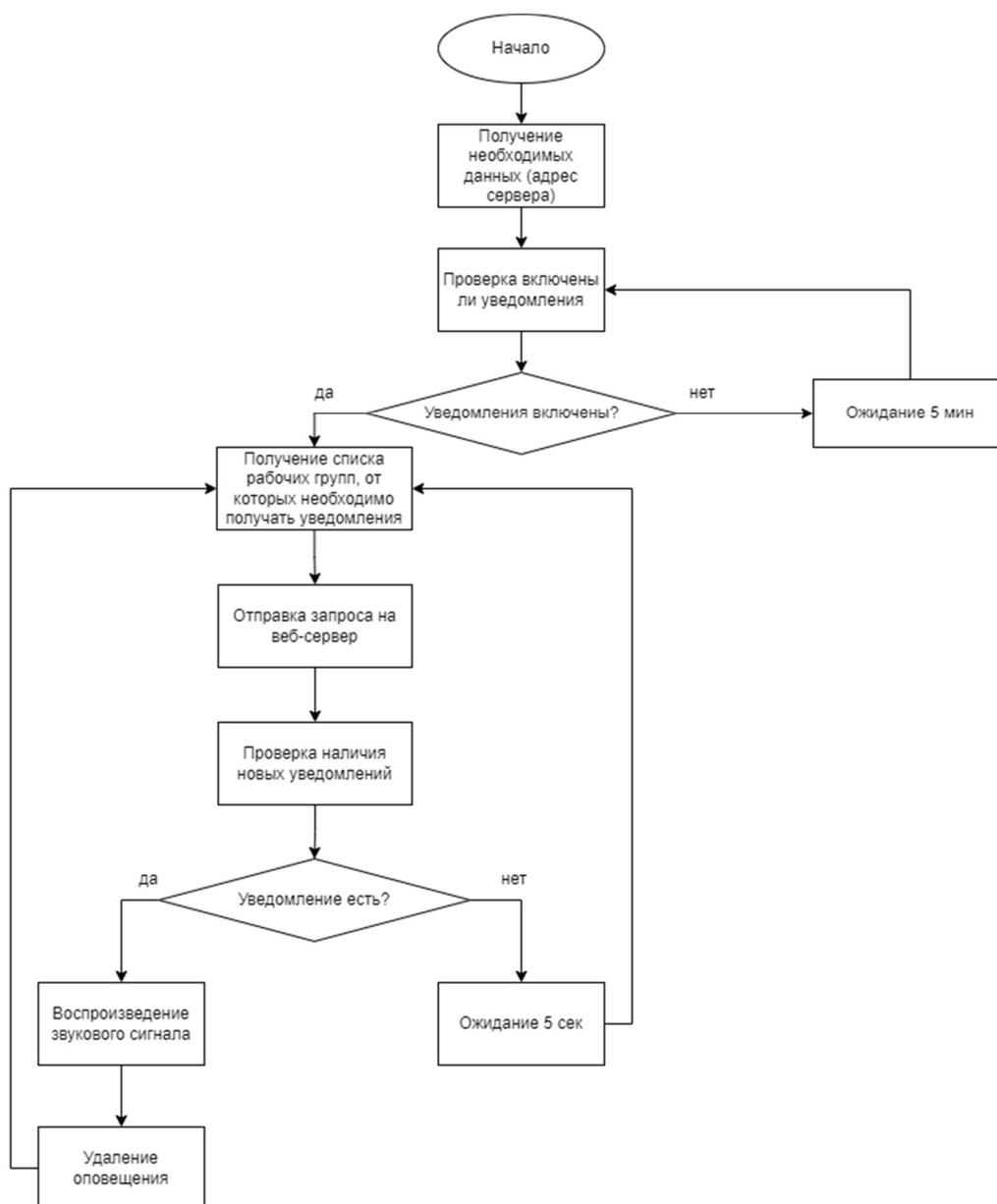


Рисунок 3.7 – Графическое изображение процесса получения специальных уведомлений

3.3. Логическая модель базы данных

Основной сущностью БД является пользователь, данные о котором хранятся в таблице с названием users.

Таблица chats служит для хранения информации о рабочей группе (идентификатор, название группы и создатель группы), поле создатель группы имеет внешний ключ на поле идентификатора в таблице users.

В таблице members хранится информация об участниках рабочих групп, такая как: идентификатор группы, который имеет внешний ключ на поле идентификатора в таблице chats и идентификатор пользователя, который имеет внешний ключ на поле идентификатора в таблице users.

Таблица links предназначена для хранения сокращенных ссылок на отправляемые файлы, содержит поля идентификатор и первоначальной длинной ссылки.

Для хранения информации о специальных уведомлениях предусмотрена таблица alerts с полями идентификатора и идентификатора группы, для которой предназначено уведомление. Поле идентификатора группы имеет внешний ключ на поле идентификатора в таблице chats

В таблице messages содержится информация об отправляемых сообщениях, такая как:

- идентификатор сообщения;
- текст сообщения;
- дата отправки;
- идентификатор отправителя;
- идентификатор получателя;
- флаг состояния сообщения (прочитано или нет).

Поля идентификатора отправителя и получателя имеют внешний ключ на поле идентификатора в таблице users

В результате проектирования базы данных была получена логическая модель, представленная на рисунке 3.8.

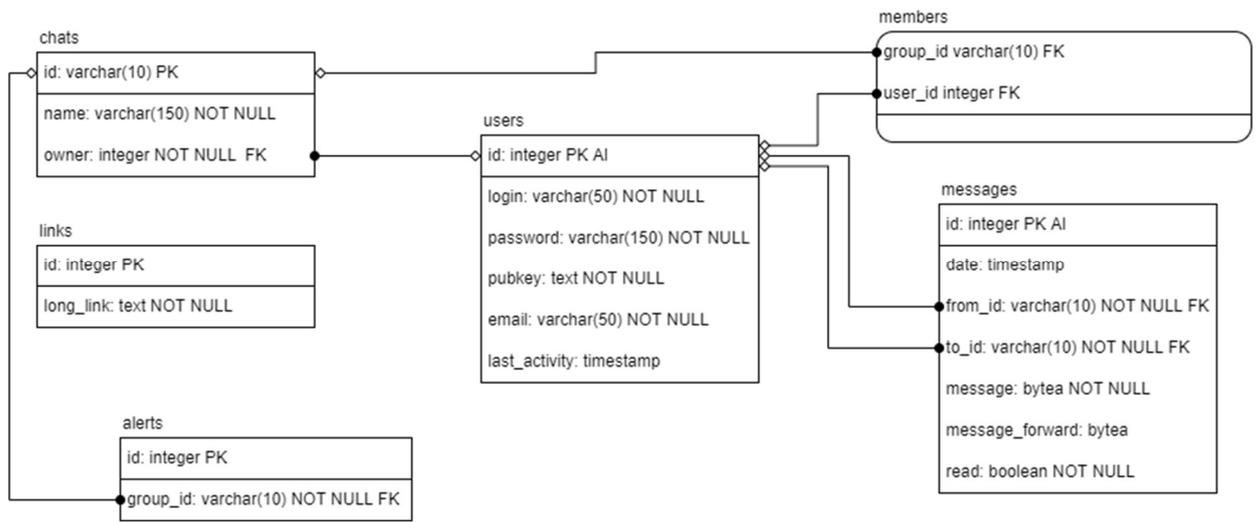


Рисунок 3.8 – Логическая модель базы данных

4. Тестирование

Тестирование серверной части системы реализуется с использованием модульного тестирования, которое выполняется с помощью средств тестирования, встроенных в фреймворк и библиотеки `pytest`. Для выполнения тестовых сценариев, все названия должны начинаться со слова «test». Каждый тестовый сценарий содержит в себе:

- тестируемый адрес и метод веб-сервера;
- параметры, необходимые для выполнения запроса;
- сравнение полученного кода запроса с ожидаемым;
- сравнение полученного результата с ожидаемым.

Если полученный код запроса совпадает с ожидаемым и полученный результат также совпадает с ожидаемым, тест считается пройденным. Пример тестовых сценариев представлен на рисунке 4.1.

```
273 def test_get_chat_name():
274     response = client.get(f"/chat/?chat_name=true&id={chat_id}")
275     assert response.status_code == 200
276     assert response.json() == "test_chat_gr"
277
278
279 def test_get_all_chats_messages():
280     response = client.put("/chat/?all=true", headers={"Authorization": f"Bearer {access_token}"})
281     assert response.status_code == 200
282     message = decrypt(int2bytes(response.json()[0]['message']), privkey)
283     assert message.decode('utf-8') == f"Chat created id={chat_id}"
284     json = {"user_id": chat_id, "username": "test_chat_gr", "message": response.json()[0]['message'], "read": 0}
285     assert response.json()[0] == json
```

Рисунок 4.1 – Пример тестовых сценариев

Используемая библиотека для тестирования способна отобразить процент покрытия функций тестами, это необходимо для того, чтобы все возможные исходы выполнения функции (положительный, отрицательный, ошибка) были протестированы. Результат тестирования веб-сервера и процент покрытия тестами представлены на рисунке 4.2.

```

C:\Users\dakfa\Desktop\st\Python\Chat\chat_api>pytest --cov-report term-missing --cov=Routers
===== test session starts =====
platform win32 -- Python 3.7.4, pytest-6.2.4, py-1.10.0, pluggy-0.13.1
rootdir: C:\Users\dakfa\Desktop\st\Python\Chat\chat_api
plugins: anyio-3.3.0, cov-3.0.0
collected 65 items

testing\test_main.py ..... [100%]

----- coverage: platform win32, python 3.7.4-final-0 -----
Name                               Stmts  Miss  Cover  Missing
-----
Routers\Alerts.py                   51     0   100%
Routers\Authorization.py            48     0   100%
Routers\Chats.py                    89     0   100%
Routers\Files.py                    72     0   100%
Routers\Messages.py                 71     0   100%
Routers\Recovery.py                 39     0   100%
Routers\Services.py                 17     0   100%
Routers\Users.py                    53     0   100%
Routers\__init__.py                  0     0   100%
-----
TOTAL                               440     0   100%

===== 65 passed in 55.46s =====

```

Рисунок 4.2 – Результат тестирования веб-сервера

Взаимодействие пользователя с файловым сервером происходит через веб-сервер, а не напрямую, следовательно, возможность работы пользователя с файлами определяется на стороне веб-сервера. Сам файловый сервер реализует функции загрузки и получения файлов без дополнительных проверок.

Файловый сервер также тестировался с применением модульного тестирования, результаты и процент покрытия тестами представлены на рисунке 4.3.

```

(venv) C:\Users\dakfa\Desktop\st\Python\ftp_vkr>pytest --cov-report term-missing --cov=.
===== test session starts =====
platform win32 -- Python 3.7.4, pytest-6.2.4, py-1.10.0, pluggy-0.13.1
rootdir: C:\Users\dakfa\Desktop\st\Python\ftp_vkr
plugins: anyio-3.3.0, cov-3.0.0
collected 3 items

test_main.py ... [100%]

----- coverage: platform win32, python 3.7.4-final-0 -----
Name                               Stmts  Miss  Cover  Missing
-----
config.py                           5     0   100%
dependencies.py                      4     0   100%
main.py                              22     0   100%
test_main.py                         15     0   100%
-----
TOTAL                               46     0   100%

===== 3 passed in 0.64s =====

```

Рисунок 4.3 – Результат тестирования файлового сервера

Тестирование системы корпоративной коммуникации поддержки реализации проектов группы разработчиков

Тестирование разработанной системы представлено в таблице 4.1.

Таблица 4.1. – Тестирование разработанной системы

Входные данные	Ожидаемый результат	Фактический результат	Статус теста
Регистрация пользователя с введением корректных данных	Успешная регистрация пользователя	Успешная регистрация пользователя	Пройден
Регистрация пользователя с использованием неподдерживаемых символов	Сообщение о том, что используются неподдерживаемые символы	Сообщение о том, что используются неподдерживаемые символы	Пройден
Отключение файлового сервера и последующая отправка файла	Сообщение об ошибке	Сообщение об ошибке	Пройден
Авторизация с введением 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
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
Попытка авторизации с введением корректных данных	Успешная авторизация пользователя	Успешная авторизация пользователя	Пройден
Создание рабочей группы с введением уже существующего названия	Сообщение о том, что группа с таким названием уже существует	Сообщение о том, что группа с таким названием уже существует	Пройден
Создание рабочей группы с использованием корректных данных	Сообщение о том, что рабочая группа создана успешно	Сообщение о том, что рабочая группа создана успешно	Пройден
Попытка отправки пустого сообщения	Сообщение о том, что поле ввода должно быть заполнено	Сообщение о том, что поле ввода должно быть заполнено	Пройден
Попытка отправки сообщения	Успешная отправка сообщения	Успешная отправка сообщения	Пройден
Попытка приглашения пользователя в рабочую группу	Успешное добавление участника в рабочую группу	Успешное добавление участника в рабочую группу	Пройден

5. Инструкция пользователя

Регистрация:

1. Заполнить поля необходимые для регистрации. Поля, которые требуется заполнять, показаны на рисунке 5.1.

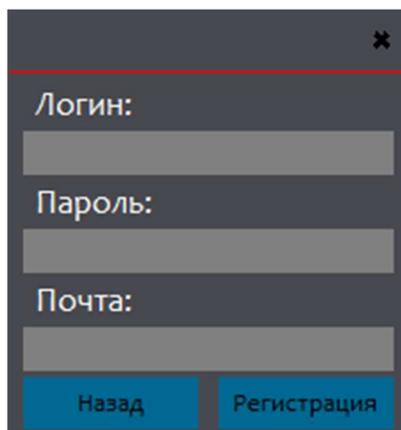
A screenshot of a registration form. It features three input fields labeled "Логин:", "Пароль:", and "Почта:". Below the fields are two buttons: "Назад" (Back) and "Регистрация" (Registration). The form is displayed in a dark-themed window with a close button in the top right corner.

Рисунок 5.1 – Поля заполнения для регистрации

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

Авторизация:

1. Заполнить поля, необходимые для авторизации, как показано на рисунке 5.2, и нажать на кнопку «Вход».

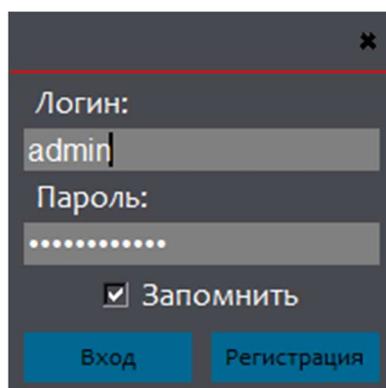
A screenshot of a login form. It features two input fields: "Логин:" with the text "admin" entered, and "Пароль:" with masked characters. Below the fields is a checked checkbox labeled "Запомнить" (Remember me). At the bottom are two buttons: "Вход" (Login) and "Регистрация" (Registration). The form is displayed in a dark-themed window with a close button in the top right corner.

Рисунок 5.2 – Поля заполнения для авторизации

Создание группового чата:

1. Перейти на вкладку настроек, которая расположена в верхней части интерфейса и обозначена знаком шестерни, после чего ввести название группы и нажать кнопку «Создать», как на рисунке 5.3.

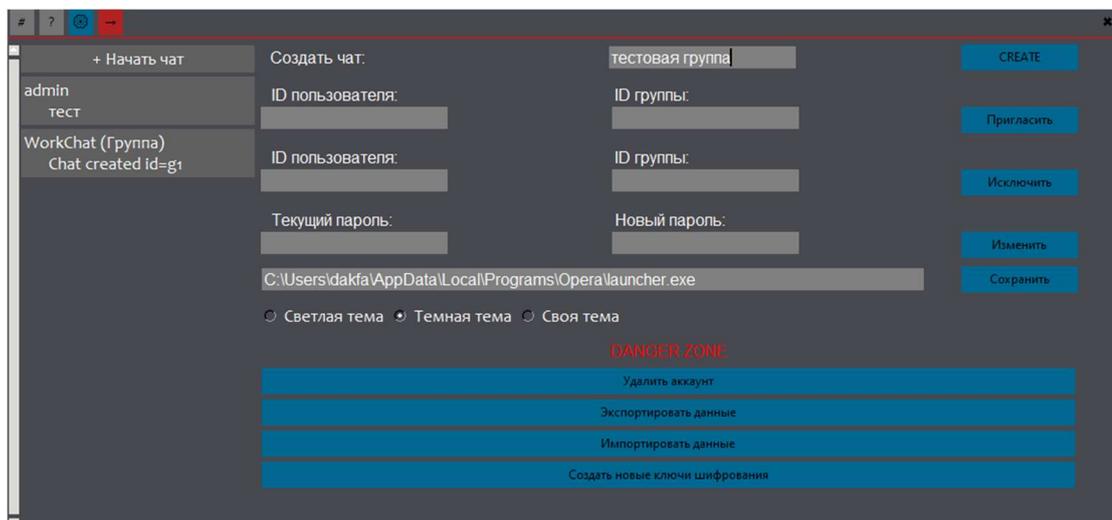


Рисунок 5.3 – Вкладка настроек

Отправка сообщения:

Для отправки сообщения необходимо нажать в левой части окна на необходимый чат, далее написать текст в поле ввода и нажать кнопку «Отправить», как показано на рисунке 5.4.

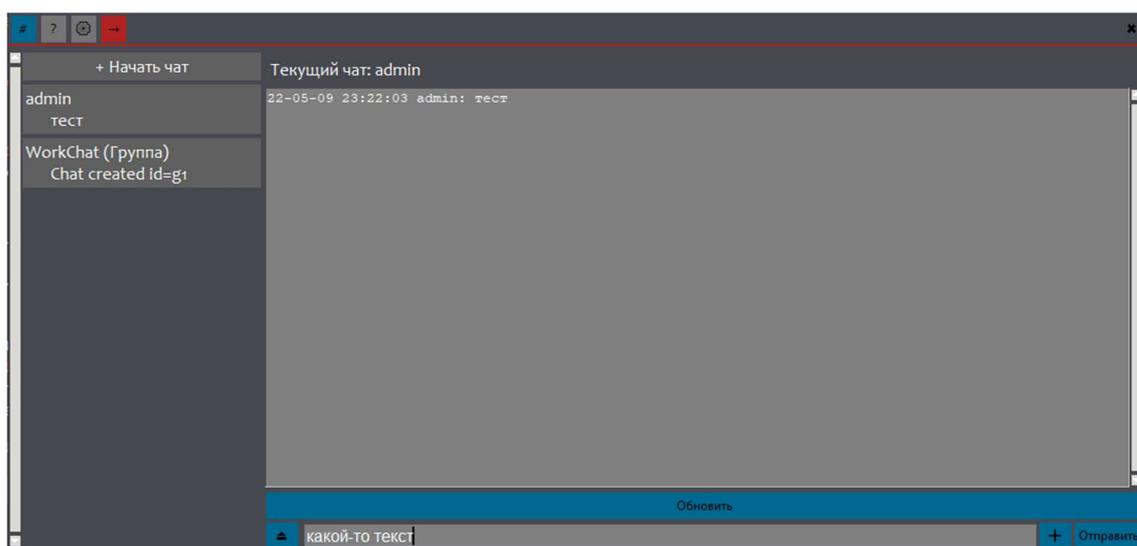
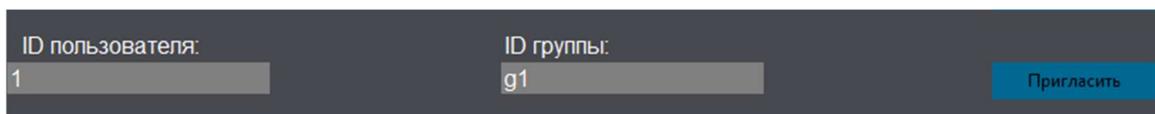


Рисунок 5.4 – Отправка сообщения

Приглашение пользователя в групповой чат:

1. Перейти на вкладку настроек, после чего заполнить id группы, id пользователя и нажать кнопку «Пригласить», как изображено на рисунке 5.5.



The image shows a dark-themed user interface for inviting a user to a group chat. It features two input fields: 'ID пользователя:' with the value '1' and 'ID группы:' with the value 'g1'. To the right of these fields is a blue button labeled 'Пригласить'.

Рисунок 5.5 – Поля заполнения для приглашения пользователя

Заключение

В результате выполнения производственной практики был определен перечень используемых инструментальных средств разработки, функционал разрабатываемой системы, спроектирована логическая модель базы данных, разработана и протестирована система корпоративной коммуникации поддержки групп разработчиков.

Разработанная система имеет следующие функциональные особенности:

- возможность регистрации;
- возможность авторизации с использованием 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;
- возможность получать специальные уведомления от организатора рабочей группы;
- отправка сообщений с вложениями в рабочих группах и индивидуальных чатах.

Список используемых источников

1. Руководство по языку программирования Python [Электронный ресурс] – URL: <https://metanit.com/python/tutorial/> (дата обращения 15.03.2022)
2. Руководство по SQL (полная версия) [Электронный ресурс] – URL: <https://proselyte.net/tutorials/sql/> (дата обращения 17.03.2022)
3. PostgreSQL: Документация: 9.4: Справочное руководство [Электронный ресурс]. – URL: <https://postgrespro.ru/docs/postgresql/9.4/reference> (дата обращения 20.03.2022)
4. Руководство по Tkinter [Электронный ресурс]. – URL: <https://docs.python.org/3/library/tkinter.html> (дата обращения 05.04.2022)
5. Построение блок-схем [Электронный ресурс]. – URL: <https://www.draw.io> (дата обращения 07.04.2022)