**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Измерительно-вычислительные комплексы»

Курсовая работа

По дисциплине «Алгоритмы и структуры данных»

Тема «Информационная система для обмена оповещаниями в корпоративной сети предприятия»

Руководство программиста

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Р.02069337.22/314-19 РП-01

Листов 15

**Исполнитель**:

студент гр. ИСТбд-21

*Жуклин Никита Дмитриевич*

« » 2023 г.

**2023**

**1. Назначение и условия применения программы**

**1.1 Назначение и функции, выполняемые приложением**

Приложение «Мессенджер» создано для обмена оповещениями внутри сети предприятия.

**Назначение и функции:**

* Распространение информации. Обеспечивает механизм быстрого и надежного распространения важных сообщений и оповещений среди сотрудников предприятия.
* Своевременное уведомление. Предоставляет средство мгновенного информирования о критических событиях, изменениях в расписании, экстренных уведомлениях и других ситуациях, требующих немедленного внимания.
* Удобство использования. Обеспечивает удобный интерфейс для отправки и приема сообщений, что делает процесс коммуникации интуитивно понятным и доступным для всех пользователей

**Пользователи**. Пользователями приложения являются работники предприятия.

**Требования к устройству.**

* Операционная система: Windows
* Разрешение экрана: 1000x750 и выше
* Интернет-соединение: подключение к сети предприятия

**1.2 Условия, необходимые для использования приложения**

Рекомендуется к использованию на Windows 7.

При создании программы использовались встроенные библиотеки “tkinter”, “json”, “socket“, “threading“, “sqlite3“ и сторонняя библиотека “notifypy“

Разработка велась в “PyCharm community edition 2022.3” на версии языка программирования Python 3.9.

**2. Характеристики программы**

**2.1 Характеристики приложения**

Значимых строк кода 281(client,py), 151(server.py), 432(client.py + server.py)

Использованные библиотеки:

“tkinter“ — это библиотека для разработки графического интерфейса на языке Python. Tkinter позволяет создавать простые GUI-приложения, используя различные виджеты и менеджеры компоновки для управления их расположением на экране.

“threading“ — это библиотека, позволяющая создавать и управлять потоками выполнения в программе. Потоки позволяют выполнять несколько задач одновременно. Модуль «threading» позволяет создавать, запускать и управлять потоками выполнения, что полезно для параллельного выполнения задач и управления ресурсами.

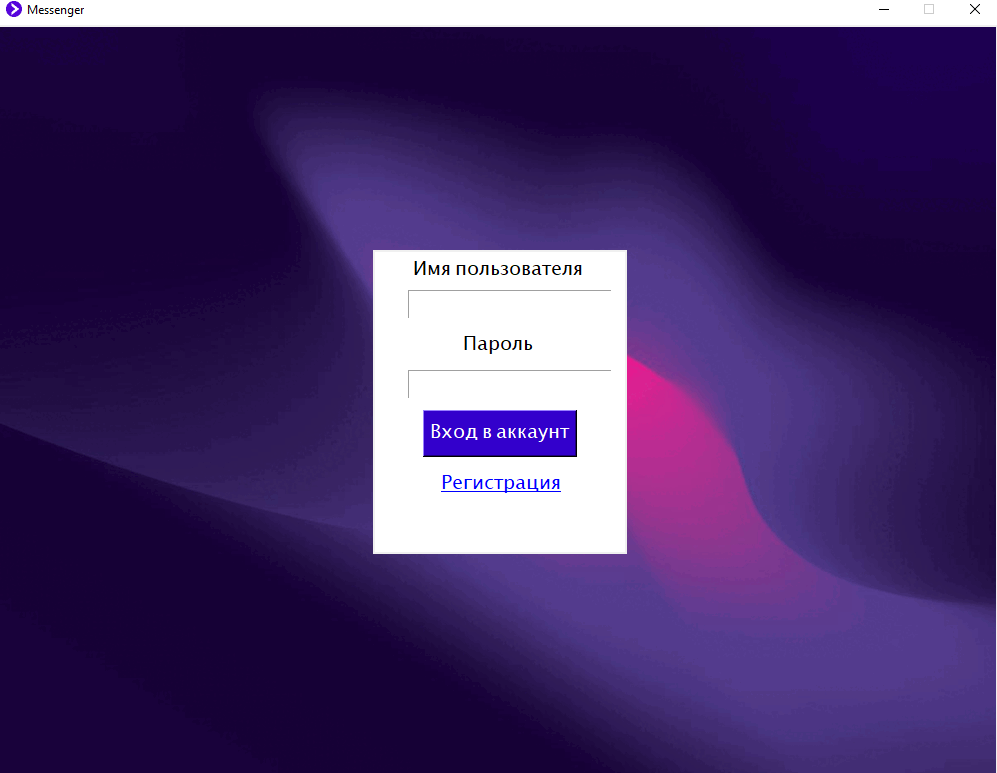
“socket“ — эта библиотека предоставляет функции для создания сетевых приложений, включая создание серверов и клиентов, обмен сообщениями и управление соединениями.

“json“ — эта библиотека позволяет кодировать и декодировать данные в/из формата JSON. Она включает методы для работы с JSON-строками, преобразования Python-объектов в JSON и обратно. JSON - это *lightweight* формат обмена данными, который удобен для чтения и записи как людьми, так и компьютерами. Он используется для передачи структурированных данных между сервером и клиентом.

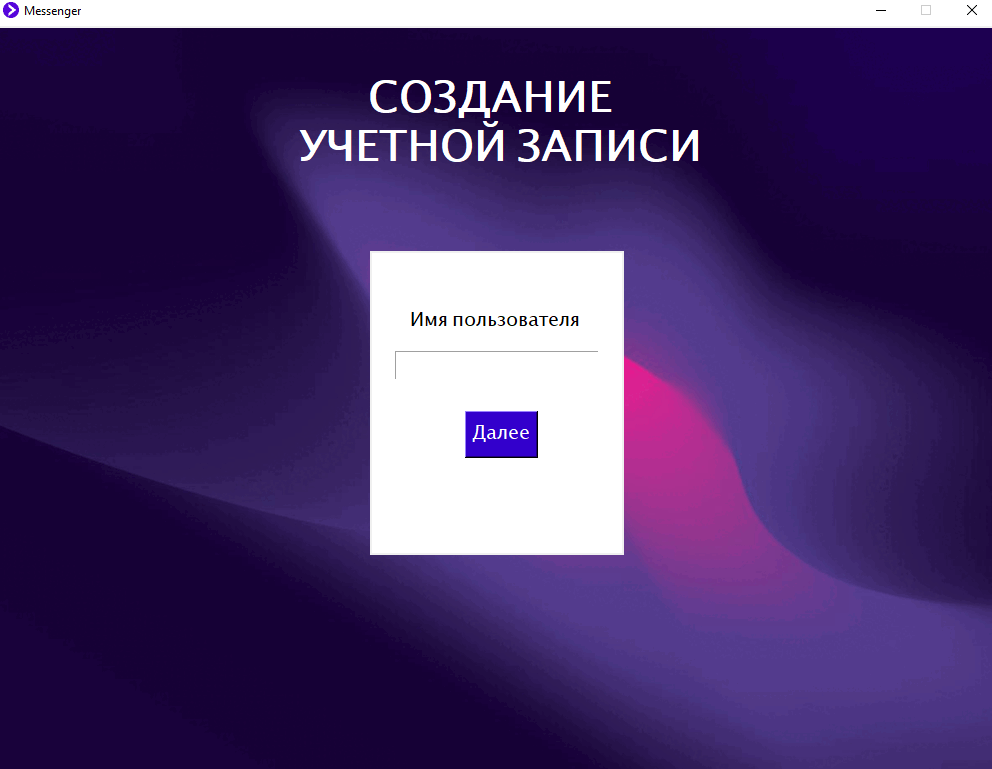
“sqlite3“ — эта библиотека предоставляет возможности работы с базами данных SQLite, включая создание, чтение, запись и выполнение запросов к базе данных. SQLite3 — это легковесная реляционная база данных, которая хранит базы данных в одном файле без необходимости сервера. Она хорошо подходит для небольших приложений или прототипов.

“notifypy“ — это библиотека для уведомлений на рабочем столе. Она позволяет отправлять уведомления на различные платформы. Удобна для создания уведомлений, информирующих пользователя о событиях в приложениях, например, об успешном завершении задачи или получении новых сообщений.

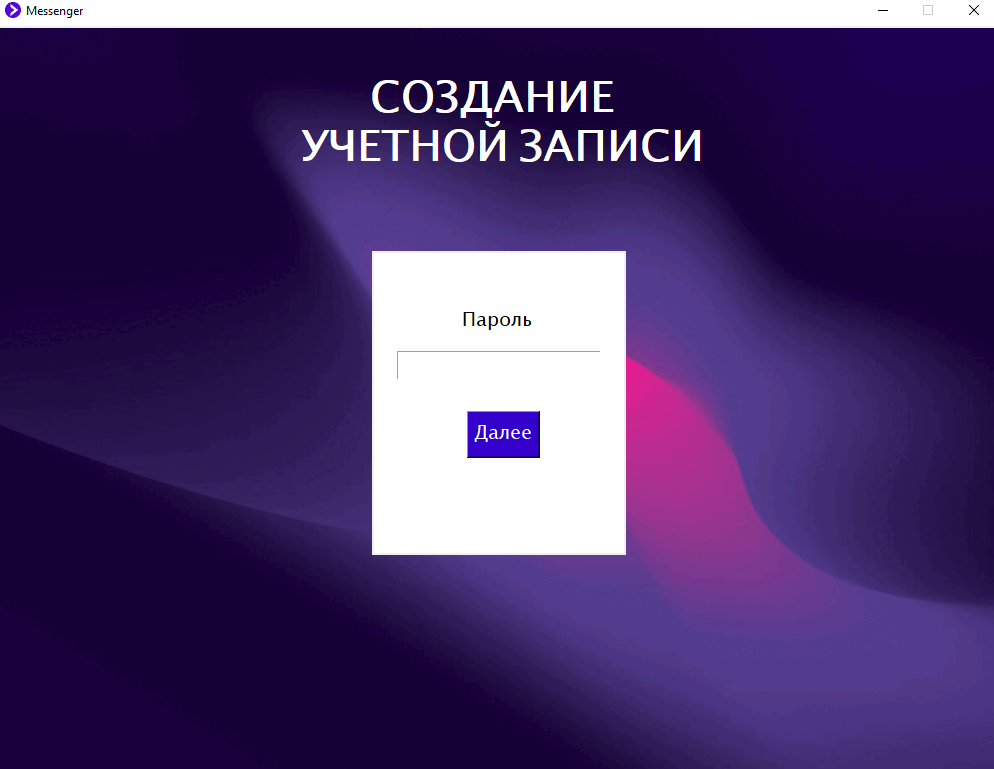
Первым окном, которое видит пользователь является окно авторизации c возможностью перейти к регистрации



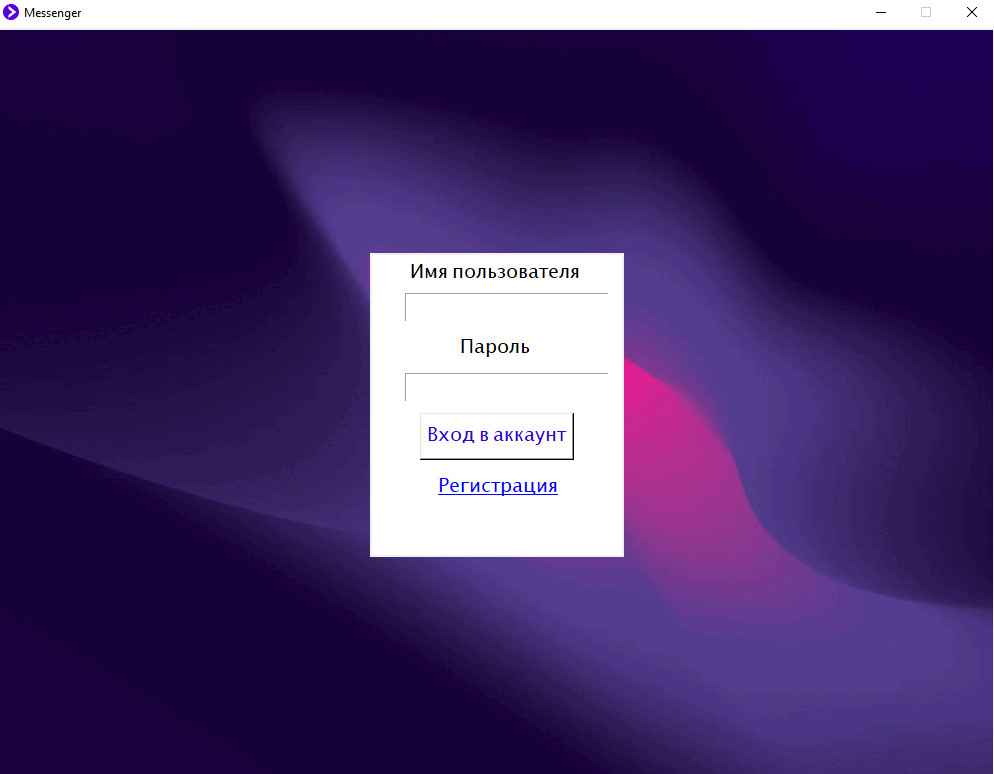
Окно регистрации, ввод имени пользователя

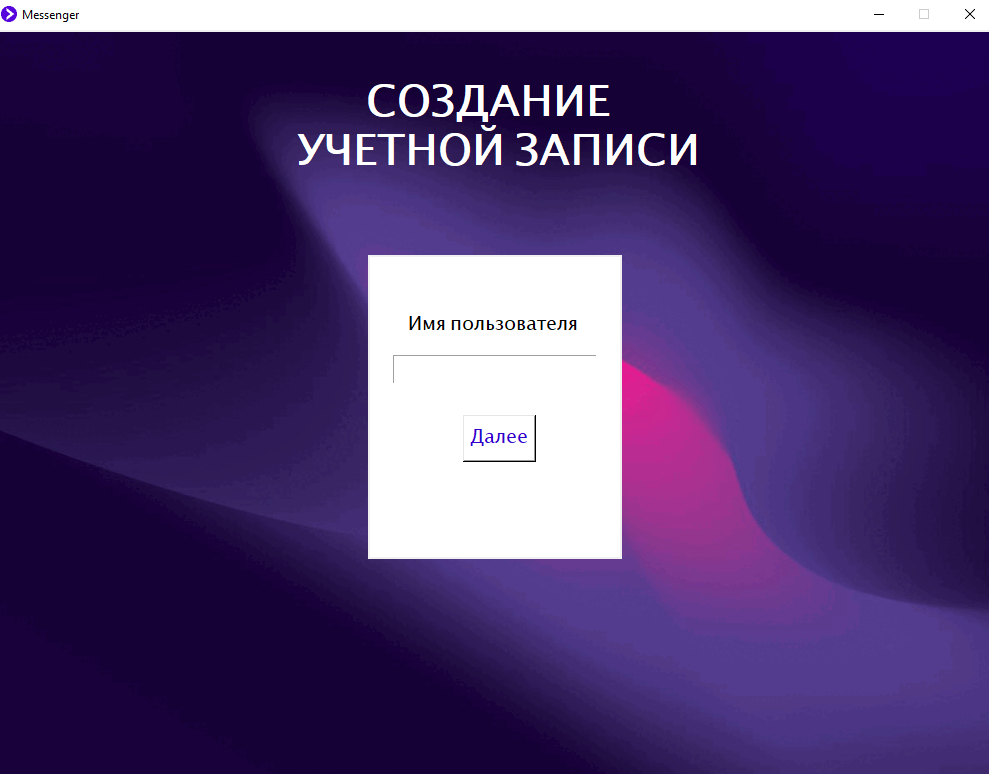


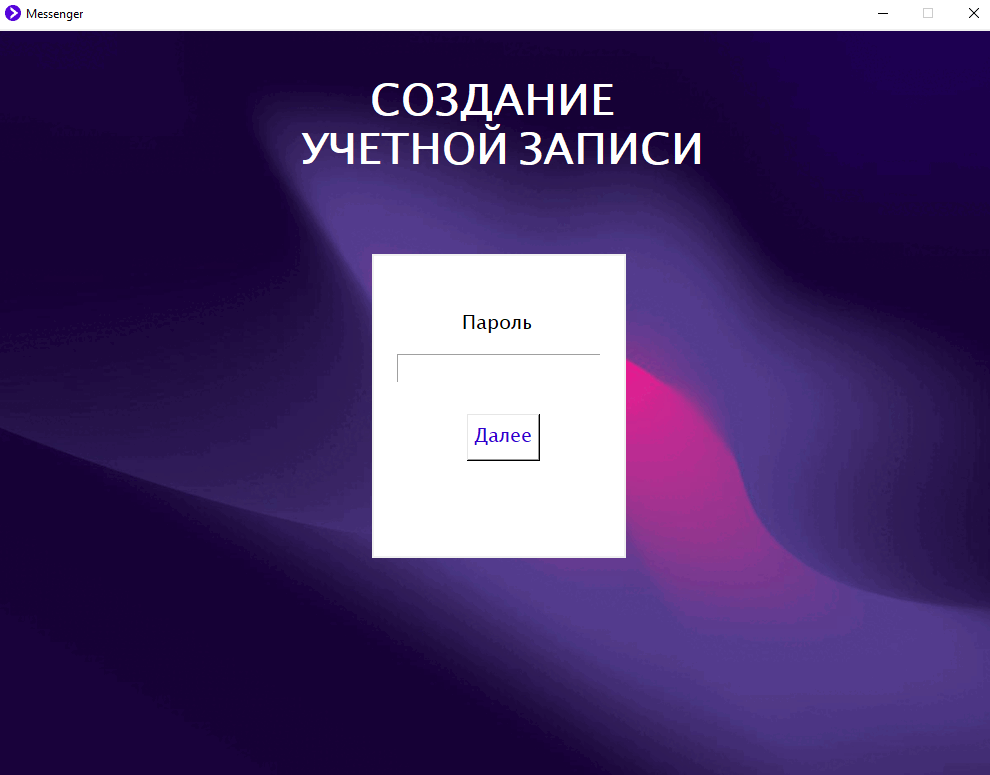
Окно регистрации, ввод пароля



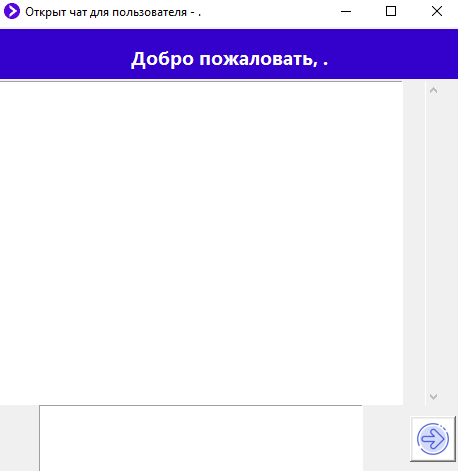
Наведение на кнопки



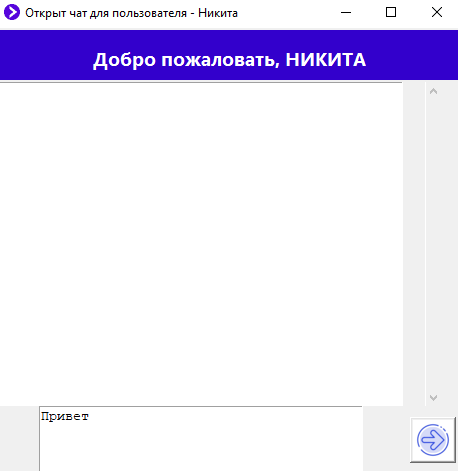


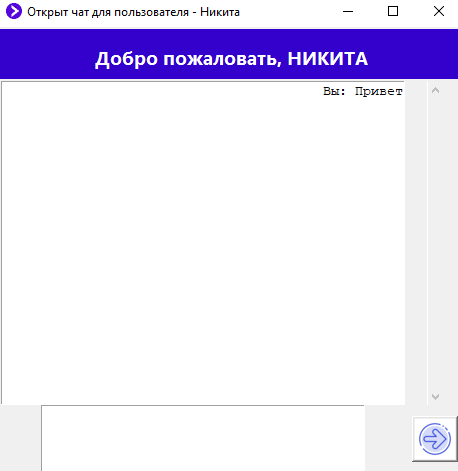


Отрисовка чата пользователей

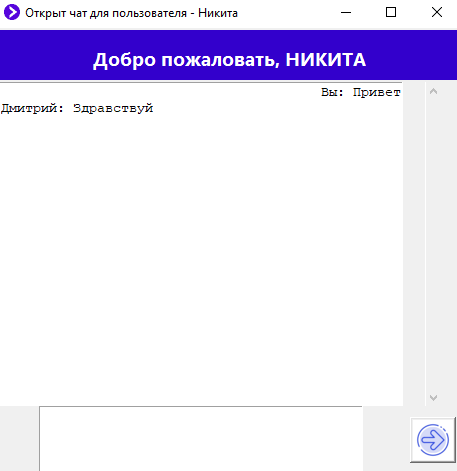


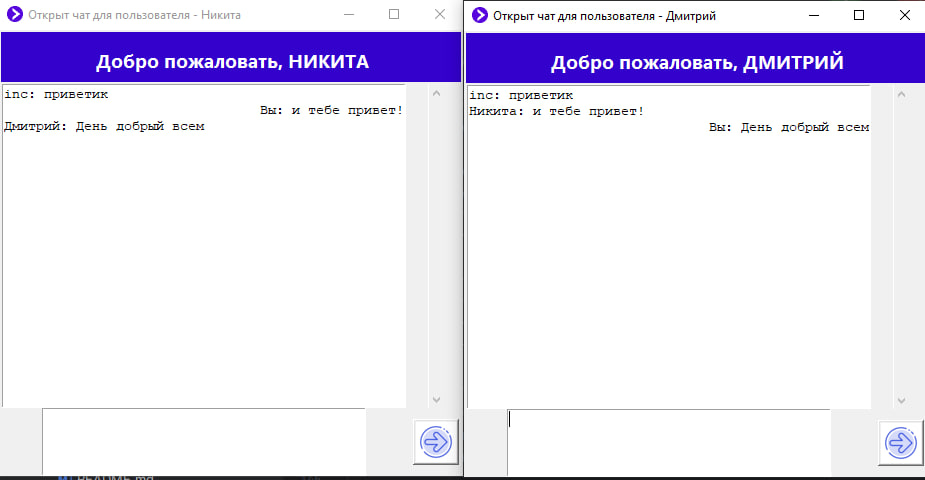
Отправка сообщений в чат





Получение сообщений от пользователей и их отображение





**2.2 Особенности реализации приложения**

В программе используется серверная часть с подключением БД для хранения истории чата, а также паролей и юзернеймов пользователей. Также используется json для надежного обмена данными между сервером и клиентов.

**3. Обращение к программе**

Код клиента содержит в себе следующие алгоритмы:

1. log\_window

Выполняет подключение к серверу с помощью библиотеки socket. Отрисовывает окно авторизации login\_window с возможностями войти в аккаунт или же перейти к окну регистрации.

1.1) Метод focus\_in

При наведении на кнопку для авторизации изменяет ее цвет текста и цвет фона

1.2) Метод focus\_off

При изменении фокуса с кнопки для авторизации на другие элементы, кнопка изменяет свой цвет текста и цвет фона

1. login

Клиент отправляет на сервер информацию о том, что пользователь хочет войти в аккаунт. После пакует username и password в список, преобразуя в json, а затем отправляет эти данные на сервер. Затем клиент ожидает ответа от сервера с разрешением на подключение или же отказом. В случае разрешения подключения осуществляется переход в окно чата. В ином случае выскакивает соответствующая ошибка.

1. reg\_window\_username

Перерисовывает окно login\_window под создание учетной записи. Создает поле для ввода имени пользователя и кнопку для продолжения со следующими методами:

3.1) Метод focus\_in

При наведении на кнопку для авторизации изменяет ее цвет текста и цвет фона

3.2)Метод focus\_off

При изменении фокуса с кнопки для авторизации на другие элементы, кнопка изменяет свой цвет текста и цвет фона

3.3) Метод check\_data

При нажатии на кнопку происходит проверка поля ввода на корректность данных по длине. Если условие выполнено, то осуществляется вызов reg\_window\_password с передачей имени пользователи в этот алгоритм

1. reg\_window\_password

Перерисовывает окно login\_window после алгоритма reg\_window\_username, заменяя поле ввода имени пользователя на поле для ввода пароля. Кнопка для продолжения также перерисовывается и имеет следующие методы:

4.1) Метод focus\_in

При наведении на кнопку для авторизации изменяет ее цвет текста и цвет фона

4.2) Метод focus\_off

При изменении фокуса с кнопки для авторизации на другие элементы, кнопка изменяет свой цвет текста и цвет фона

4.3) Метод check\_data

При нажатии на кнопку происходит проверка поля ввода на корректность данных по длине. Если условие выполнено, то осуществляется вызов алгоритма register с передачей имени пользователя и введённого пароля.

1. register

Клиент отправляет на сервер информацию о том, что пользователь хочет зарегистрироваться. После чего упаковывает username и password в список, преобразуя в json, а затем отправляет эти данные на сервер. Затем клиент ожидает ответа от сервера с информацией об успехе регистрации. В случае успешной регистрации пользователя происходит отключения от сервера, для прохождения повторного выбора опции входа или регистрации, осуществляется вызов log\_window.

1. open\_chat

Отрисовывает окно чата chat\_window, в котором можно прочитать старые сообщения и новые сообщения от пользователей, для этого в окне имеется средство прокрутки. Имеется окно ввода для своего сообщения, а также кнопка для отправки сообщений, которая вызывает алгоритм send\_message. Также отправка сообщений вызывается с помощью нажатия кнопки Enter в окне ввода. И напоследок создается новый поток с помощью threading с запуском алгоритма обработки сообщений handle\_message

1. handle\_message

При подключении пользователя к чату прогружает историю сообщения с базы данных, находящейся на сервере. После загрузки истории переходит в режим принятия сообщений в реальном времени от остальных пользователей. Если окно чата окажется свернутым, то в центр уведомления будут приходить оповещения о новых сообщениях от пользователей с помощью библиотеки notifypy.

1. send\_message

При вызове обрабатывает поле ввода из окна chat\_window, удаляя пустые символы в начале и конце сообщения. После этого отправляет обработанное сообщение на сервер, очищает поле ввода и вызывает алгоритм display\_message для отображения сообщения у пользователя, который отправил данное сообщение.

1. display\_message

Проверяет чтобы сообщение, полученное после обработки в send\_message не оказалось пустым. Если сообщение не пустое выводит сообщение пользователя справа, визуально отделяя его от чужих сообщений. Также имя пользователя сменяется на «Вы»

Код сервера содержит следующие алгоритмы:

1. accept\_clients

Принимает на вход подключение от клиента. После чего ожидает выбора операции от пользователя. Например, вход или регистрация.

* Если выбрана операция входа сервер ожидает на вход данные в формате json, при их получении сервер отправляет данные пользователя на проверку(check\_credentials). Если данные прошли проверку успешно, то сервер отправляет информацию о разрешении подключения к чату на клиент. После запускает поток для прослушивания сообщений от пользователя и отправления ему сообщений от других пользователей(handle\_client). В случае если данные не прошли проверку отправляет информацию об ошибке.
* Если выбрана операция регистрации сервер ожидает на вход данные в формате json, при их получении сервер обращается к алгоритм register\_user для проверки имени пользователя на существование в БД. Если регистрация успешно, то отправляется соответствующее сообщение на клиенте, если неуспешно, то отправляет информацию о неудаче.

1. check\_credentials

Алгоритм подключается к БД users\_credentials и проверяет данные пользователя на соответствие. В случае успеха передает True, в случае неудачи False

1. register\_user

Алгоритм подключается к БД users\_credentials и проверяет имя пользователя на существование в БД. Если такое имя пользователя еще не числится в БД, то возвращает True и записывает данные нового аккаунта такие как username и password. В случае если такое имя пользователя уже существует, то возвращает False

1. handle\_client

При подключении клиента вызывает алгоритм для получения истории сообщений(retrieve\_chat\_history). Начинает прослушивать сообщения от пользователей. При получении сообщения запускает алгоритм save\_message\_to\_db, чтобы сохранить сообщение в БД chat\_history.db. После запускает алгоритм broadcast, чтобы отправить полученное сообщение всем пользователям чата.

1. save\_message\_to\_db

Получает на вход имя пользователя и его сообщения, а затем сохраняет эту информацию в базу данных c историей чата(chat\_history.db)

1. broadcast

Принимает на вход строку вида «username:message». Алгоритм проходит по списку всех клиентов и отправляет каждому полученное на вход сообщение. Если с каким то клиентом возникает проблема, то подключение закрывается, а клиент удаляется из списка.

Алгоритм подключается к БД users\_credentials и проверяет имя пользователя на существование в БД. Если такое имя пользователя еще не числится в БД, то возвращает True и записывает данные нового аккаунта такие как username и password. В случае если такое имя пользователя уже существует, то возвращает False

Библиотеки, используемые во время разработки приложения:

Библиотека “json“ – предоставляет возможность для работы с файлами json. Позволяет кодировать и декодировать данные в/из формата JSON.

Библиотека “socket“ – предоставляет инструменты для работы с сокетами. Позволяет создавать клиент-серверные приложения, обмениваться данными между компьютерами через сеть, взаимодействовать по протоколам TCP/IP и др.

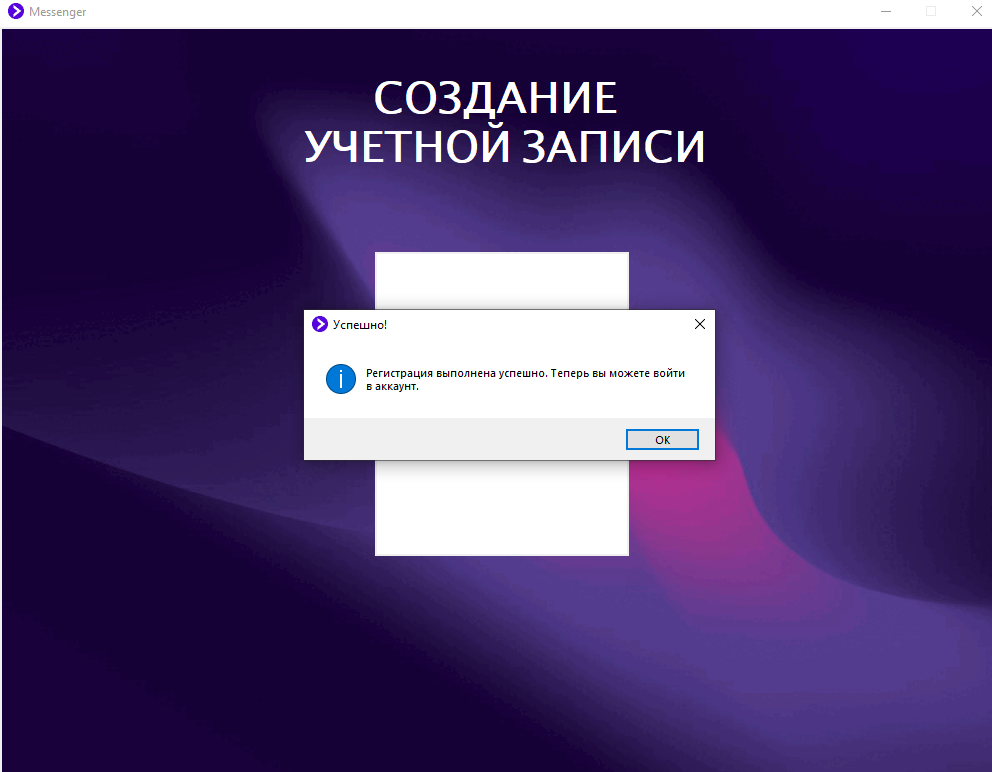
Библиотека “notifypy“ – позволяет создавать различные уведомления на рабочем столе.

Библиотека “threading” – дает возможность создавать и управлять потоками выполнения. Потоки позволяют выполнять параллельные задачи, ускоряя работу программы и позволяя работать с асинхронными задачами.

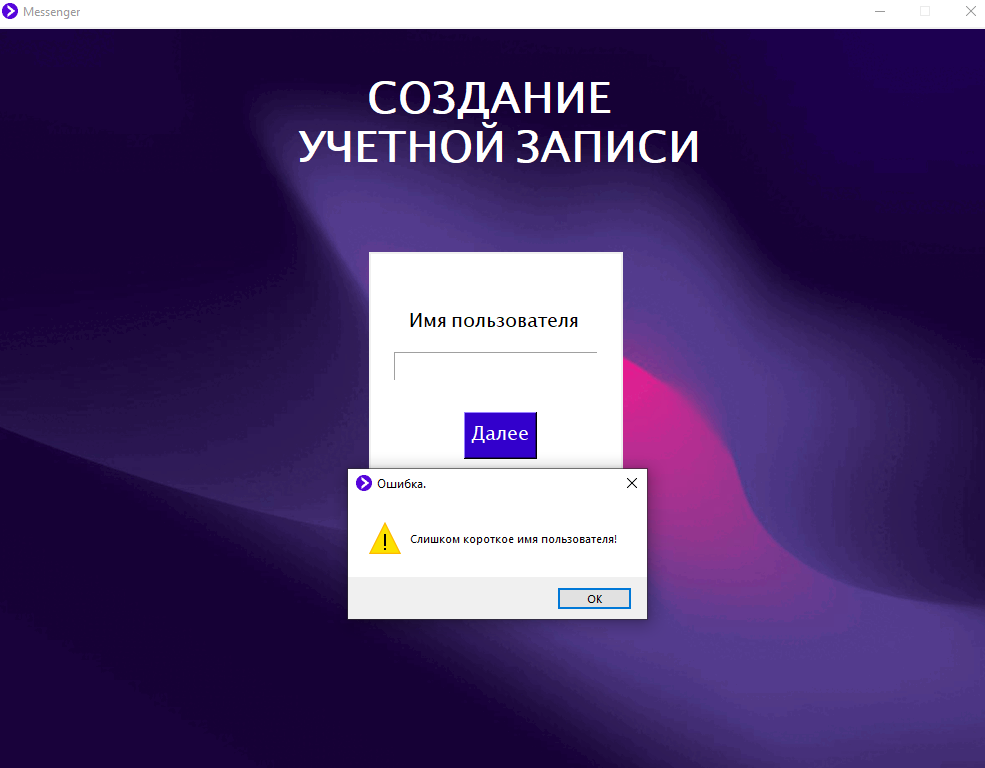
Библиотека “tkinter” – библиотека для создания графического интерфейса. Предоставляет инструменты для создания окон, кнопок, полей ввода и др.

**4. Сообщения**

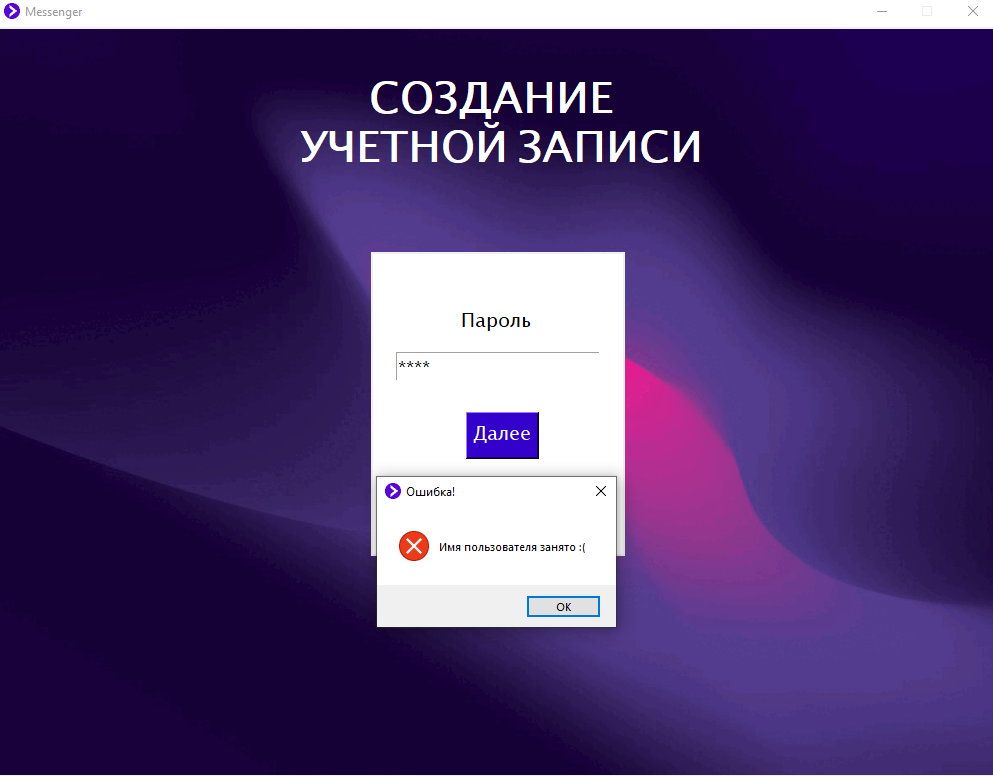
Успешная регистрация



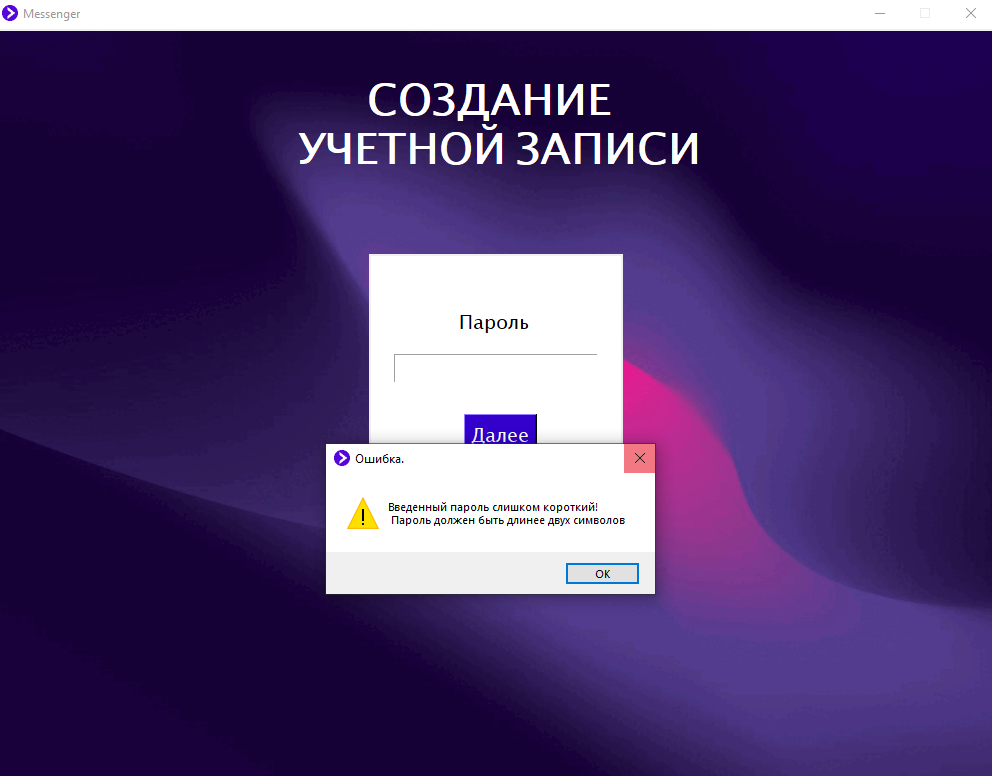
Введенное имя пользователя слишком короткое



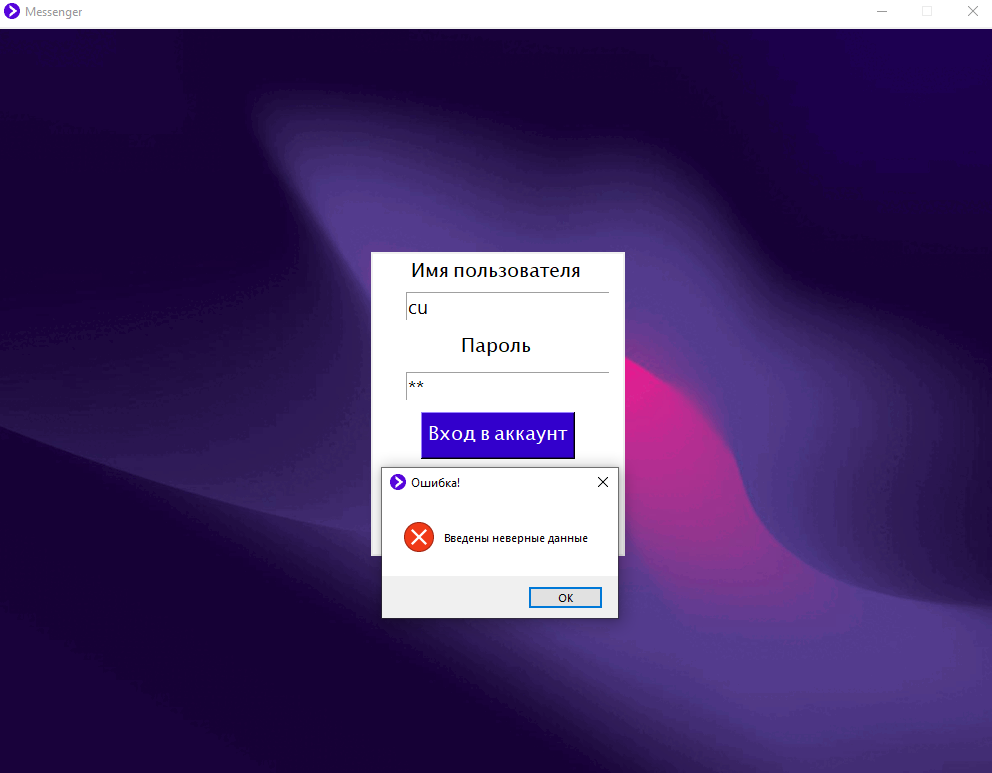
Аккаунт с таким именем пользователя уже существует



Слишком короткий пароль



Введены неверный пароль или имя пользователя



Звуковое оповещение в Windows

