Содержание

**Введение**

В современном мире сетевые мессенджеры стали неотъемлемой частью повседневной жизни, обеспечивая мгновенную связь между людьми независимо от их географического положения. Развитие технологий и широкое распространение интернета способствовали появлению множества приложений для обмена сообщениями, таких как WhatsApp, Telegram, Viber и другие. Однако, несмотря на обилие существующих решений, разработка собственного мессенджера остается актуальной задачей, позволяющей глубже понять принципы работы сетевых технологий и протоколов обмена данными.

Целью данной курсовой работы является разработка простейшего сетевого мессенджера, который будет обеспечивать базовые функции обмена сообщениями между пользователями.

Для достижения поставленной цели необходимо решить следующие задачи:

1) Изучить современные технологии и протоколы, используемые в разработке мессенджеров.

2) Спроектировать архитектуру простейшего мессенджера, включающую серверную и клиентскую части.

3) Реализовать серверную часть мессенджера, обеспечивающую прием и передачу сообщений между клиентами.

4) Разработать клиентское приложение, позволяющее пользователям обмениваться сообщениями в реальном времени.

5) Провести тестирование и отладку разработанного мессенджера для обеспечения его корректной работы.

Методы и средства решения поставленных задач включают использование языка программирования Python, а также библиотек и фреймворков, таких как Flask для серверной части и Tkinter для клиентской. Для обмена данными между клиентами и сервером будет использоваться протокол HTTP и WebSocket обеспечивающий двустороннюю связь в реальном времени.

Актуальность данной работы обусловлена потребностью в создании новых и улучшении существующих инструментов для обмена сообщениями, что способствует развитию коммуникационных технологий и повышению удобства взаимодействия между пользователями. Новизна работы заключается в создании мессенджера, который не только демонстрирует базовые принципы работы подобных приложений, но и предоставляет возможность для дальнейшего расширения и модификации.

Таким образом, данная курсовая работа направлена на разработку простейшего сетевого мессенджера, который будет служить эффективным инструментом для обмена сообщениями и позволит углубить понимание принципов работы сетевых приложений.

**1 Обоснование выбора методов реализации**

Я выбрал Python в качестве основного языка программирования для разработки данного программного продукта по нескольким причинам. Во-первых, у меня есть значительный опыт работы с Python, что позволяет мне эффективно и быстро решать задачи, возникающие в процессе разработки. Python известен своей простотой и читаемостью кода, что ускоряет процесс написания и отладки программ. Кроме того, богатая экосистема библиотек и фреймворков Python, таких как Tkinter и другие, предоставляет мощные инструменты для создания функциональных и удобных пользовательских интерфейсов.

Выбор формата настольного приложения был обусловлен рядом факторов, среди которых ключевым является наличие значительного опыта в разработке именно настольных приложений, что существенно превосходит мой опыт в создании web-приложений. Настольные приложения предоставляют более глубокий контроль над системными ресурсами и позволяют реализовать более сложные и производительные функции. Этот формат также обеспечивает более тесное взаимодействие с пользователем, что способствует улучшению пользовательского опыта.

Для реализации серверной части моего приложения я остановили свой выбор на библиотеке Flask. Этот выбор был сделан по нескольким причинам. Во-первых, Flask отличается простотой и удобством в использовании, что позволяет значительно ускорить процесс разработки и сосредоточиться на функциональности приложения. Во-вторых, Flask обладает обширным и активным сообществом разработчиков, что обеспечивает доступ к большому количеству ресурсов, таких как документация, учебные материалы и форумы, где можно найти ответы на возникающие вопросы и решения для различных проблем. Это сообщество также активно поддерживает и развивает библиотеку, что гарантирует её актуальность и надёжность.

Для создания интерфейса пользователя клиентской части приложения был выбран Tkinter, так как он изначально включен в установочный пакет Python. Tkinter достаточно удобен для создания простых графических интерфейсов. Он позволяет создавать все стандартные элементы интерфейсов, такие как надписи, кнопки, поля ввода текста и т.д. Поскольку Tkinter существует достаточно давно, в сети имеется большое количество различных обучающих материалов и возможностей решения тех или иных проблем, возникающих в ходе разработки приложения.

**3 Проектирование**

В качестве основных функций приложения были выбраны базовые функции простейшего мессенджера:

1. Регистрация новых пользователей в системе
2. Авторизация существующих пользователей в систему
3. Поиск пользователей другими пользователями и создание диалогов
4. Отправка текстовых сообщений в приватном диалоге
5. Обновление окна диалога в реальном времени