**Министерство промышленности и торговли**

**Тверской области**

**ГБПОУ «Тверской колледж им. А.Н. Коняева»**

##### **КУРСОВОЙ ПРОЕКТ**

На тему *«Разработка веб-приложения “Messanger“»*

Студента(ки) группы 4-ИС1

Ф.И.О.

Специальность Информационные системы и программирование

Руководитель Ишкова Лариса Георгиевна

(фамилия, имя, отчество)

**Тверь** **2022**

**МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ И ТОРГОВЛИ ТВЕРСКОЙ ОБЛАСТИ ГБПОУ «Тверской колледж им. А.Н.Коняева»**

Отзыв на курсовую работу (проект)

По учебной дисциплине/ МДК:

Студентки

Группа

Тема

Вид

1.Работа соответствует требованиям, предъявленным к работам данного вида:

1 2 3 4 5 6 7 8 9 10

2.Отображение в плане основных вопросов темы, логичность и последовательность:

1 2 3 4 5 6 7 8 9 10

3. Содержание работы соответствует представленному плану:

1 2 3 4 5 6 7 8 9 10

4.Качество выполнения и оформления работы:

1 2 3 4 5 6 7 8 9 10

5.Использование новейших источников (2018-2023г. издания):

1 2 3 4 5 6 7 8 9 10

6.Наличие самостоятельных выводов и их обоснование:

1 2 3 4 5 6 7 8 9 10

7.Проявление самостоятельности и творчества в разработке темы:

1 2 3 4 5 6 7 8 9 10

Всего балов:

Примечание: 70-57 б.- «5», 56-45 б. – «4», 44-30 б. – «3», ниже 30 б. – «2»

Особое мнение по оценке работы:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

В целом работа заслуживает оценки\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_)

Дата: \_\_\_\_\_\_\_\_\_\_ Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ И ТОРГОВЛИ ТВЕРСКОЙ ОБЛАСТИ**

**ГБПОУ «Тверской колледж им. А.Н.Коняева»**

ЗАДАНИЕ ДЛЯ ВЫПОЛНЕНИЯ КУРСОВОГО ПРОЕКТА

Студентке

Тема:

Содержание расчетно-пояснительной записки

Срок окончания выполнения работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г.

Руководитель курсового проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Заведующий отделением\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г.

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись,

**Содержание**

[Введение 7](#_Toc153153468)

[1 Теоретическая часть(технологический раздел) 8](#_Toc153153469)

[1.1 Создание десктоп-мессенджера 8](#_Toc153153470)

[1.2 Понятие Front-end разработки 9](#_Toc153153471)

[1.3 Понятие Back-end разработки 10](#_Toc153153472)

[2 Практическая часть 15](#_Toc153153475)

[2.1 Анализ предметной области 15](#_Toc153153476)

[2.2 Техническое задание 15](#_Toc153153477)

[2.3 Разработка функционала и базы данных 19](#_Toc153153491)

[2.4 Создание интерфейса 22](#_Toc153153494)

[2.5 Тестирование 26](#_Toc153153499)

[Заключение 40](#_Toc153153503)

[Список литературы и источников 41](#_Toc153153504)

# Введение

Тема курсового проекта – Разработка десктоп-приложения “Messanger”

В современном мире очень важна скорость передачи данных, поэтому практически мгновенная отправка сообщений облегчает жизни миллионам людей каждый час. Но современные мессенджеры и социальные сети перегружены обильным количество рекламы и спама, а также занимают большой объём памяти на устройствах пользователя.

Данное приложение создано для облегчения общения между людьми и заведением новых знакомств. Все что нужно сделать так это скачать и зарегистрироваться в нем.

Для использования этого приложения достаточно основных знаний об устройстве.

В рамках курсового проекта можно определить следующие направления развития:

Можно выделить следующие задачи развития курсового проекта:

* Создание пользовательского интерфейса;
* Проектирование и разработка основных функций приложения;
* Интеграция всех модулей в рабочую версию приложения.

# Теоретическая часть(технологический раздел)

## Создание десктоп-мессенджера

Создание десктоп мессенджера на основе C# и WPF, подобно созданию веб-сайта, включает несколько этапов и компонентов:

* Анализ и проектирование: Определение требований к мессенджеру, функциональности, интерфейса пользователя и основных характеристик приложения.
* Бекенд: это серверная часть, обрабатывающая логику бизнес-приложения и взаимодействующая с базой данных. На C# это может быть реализовано с использованием технологий, таких как ASP.NET, .NET Core, или другие фреймворки для создания веб-серверов и API.
* Фронтенд (WPF): WPF (Windows Presentation Foundation) - это технология для создания графического интерфейса пользователя в приложениях под Windows. Ваш десктопный мессенджер будет использовать WPF для создания пользовательского интерфейса. В этом случае, можно будет создавать окна, элементы управления, обработчики событий и визуальные элементы приложения, используя XAML (язык разметки) и C#.
* Коммуникация через сеть: для обмена сообщениями между пользователями через сеть понадобится реализация сетевой логики. Можете использовать TCP/IP или более высокоуровневые протоколы, такие как HTTP/WebSockets для обеспечения связи между клиентами и сервером.
* Безопасность: Реализация механизмов безопасности и аутентификации важна для защиты данных пользователей. Это может включать в себя шифрование сообщений, аутентификацию пользователей и управление доступом.
* Тестирование и отладка: после создания мессенджера необходимо тестировать его на различных устройствах и условиях сети, чтобы убедиться в его надежности и эффективности.

Все это неотъемлемые части создания любого десктоп приложения. Без выполнения этих пунктов невозможно представить корректно работающую программу.

## Понятие Front-end разработки

Frontend — это часть веб-разработки, которая отвечает за создание пользовательского интерфейса веб-сайтов или веб-приложений. Она занимается тем, что пользователь видит и с чем взаимодействует в браузере или приложении.

Frontend-разработчики заботятся о том, чтобы веб-сайт был удобен в использовании, отзывчивым (адаптивным под разные устройства и разрешения экранов) и обладал хорошей пользовательской интеракцией.

Основная цель frontend-разработки - обеспечить пользователей более интуитивным, эстетически приятным и удобным пользовательским опытом. Это включает в себя не только визуальную составляющую, но и обеспечение быстрой загрузки страниц, адаптивности под разные устройства, а также доступность для пользователей с ограниченными возможностями.

Frontend-разработчики тесно взаимодействуют с другими специалистами, такими как дизайнеры, backend-разработчики и специалисты по UX/UI, чтобы создать полноценный и функциональный веб-продукт. Их работа оказывает значительное влияние на восприятие и успешность сайта или приложения, так как пользовательский интерфейс является ключевым элементом взаимодействия пользователя с продуктом в онлайн-среде.

Фронтенд-разработчики используют различные средства для создания пользовательского интерфейса веб-приложений или сайтов. Вот некоторые из них:

* Языки разметки: HTML используется для структурирования контента на веб-страницах. CSS применяется для стилей и внешнего оформления элементов, делая страницу более привлекательной и удобной для использования.
* Языки программирования: C# является основным языком для программирования веб-интерфейса. Он добавляет интерактивность, обработку событий и анимации на веб-страницах.
* Фреймворки и библиотеки: Фронтенд-разработчики часто используют фреймворки и библиотеки, такие как React.js, Angular, Vue.js, чтобы упростить процесс создания сложных пользовательских интерфейсов и повысить эффективность разработки.

## Понятие Back-end разработки

Backend (или серверная часть) отвечает за работу веб-сайта или веб-приложения на стороне сервера. Это то, что происходит за кулисами и не отображается непосредственно перед пользователем, в отличие от фронтенда, который представляет интерфейс для пользователей.

Backend включает в себя серверы, базы данных, приложения и всю логику, необходимую для работы веб-сайта или приложения. Здесь обрабатываются запросы, хранится информация, выполняются операции с данными и формируется ответ, который отправляется обратно пользователю через фронтенд.

### **Схема работы**

Работа Backend и основные принципы функционирования веб-сайта можно сжать до трех ключевых этапов:

* Получение и обработка данных от пользователя, например, ввод поискового запроса.
* Обработка этих данных на сервере, включая поиск в базе данных и другие запросы.
* Представление полученного ответа от сервера в удобочитаемом виде для пользователя, т.е. вывод результатов.

Хотя взаимодействие с бэкендом кажется простым, его реализация может быть сложной. Это связано с тем, что различные веб-серверы используют разные языки программирования и могут работать с разными базами данных, интерфейсами и архитектурами. Необходимо тщательно объединить все эти компоненты и тестировать каждый шаг, иначе результат может быть нечитаемым.

### Используемые языки программирования в бэкенде

В бекенде (серверной части) веб-приложений и сайтов используются различные языки программирования, каждый из которых имеет свои особенности и подходит для определенных задач:

* Java: это универсальный язык программирования общего назначения. Он используется для создания масштабируемых, надежных и высокопроизводительных веб-приложений. Java подходит для корпоративных систем, распределенных приложений и работы с большими объемами данных. Он также используется в разработке Android-приложений.
* PHP: Это один из наиболее распространенных языков программирования для веб-разработки. PHP подходит для создания динамических веб-сайтов, обработки форм, управления базами данных и разработки веб-приложений. Он часто используется в сочетании с веб-фреймворками, такими как Laravel или Symfony.
* Python: Python хорошо подходит для разработки веб-приложений, имеет широкий спектр применения от простых скриптов до сложных веб-сервисов. Он удобен в использовании, имеет чистый синтаксис и обширные библиотеки, что делает его отличным выбором для быстрой разработки прототипов и сложных систем.
* Ruby: Язык Ruby известен своей простотой и удобством разработки. Он используется для построения гибких и легко поддерживаемых веб-приложений. Фреймворк Ruby on Rails является популярным выбором для быстрой разработки веб-приложений.
* JavaScript (Node.js): Node.js позволяет использовать JavaScript для серверной разработки. Он подходит для создания масштабируемых и высокопроизводительных веб-приложений. Node.js также хорошо совмещается с JavaScript на фронтенде, что позволяет разработчикам использовать один язык для всей системы.

Выбор языка программирования в бекенде зависит от требований проекта, сценариев использования, экспертизы разработчиков и специфики задачи. Кроме того, разработчики также могут использовать различные фреймворки, библиотеки и инструменты, чтобы облегчить процесс разработки и расширить функциональность приложения.

Также не стоит забывать о различных системах управления базами данных. Хотя MySQL является чаще всего используемым вариантом, на рынке существуют альтернативы, такие как PostgreSQL, SQLite, MongoDB и другие. Каждая из этих систем имеет свои особенности и специфику работы. Это предоставляет множество инструментов для выбора, но при этом их набор достаточно структурирован, что помогает сосредотачиваться на конкретных решениях, не рассеивая усилия. Обязанности и задачи backend-разработчика

Исходя из вышесказанного, backend-разработчику нужно выполнять следующие задачи:

* Разработка серверной логики: Создание и поддержка алгоритмов и процессов для корректной обработки запросов на сервере.
* Управление базами данных: Создание, настройка и оптимизация баз данных, включая выполнение запросов SQL и обеспечение безопасности данных.
* API и взаимодействие: Создание и поддержка API для обеспечения взаимодействия между серверной и клиентской частями приложения или различными компонентами системы.
* Безопасность приложения: Разработка мер безопасности, таких как аутентификация, авторизация и защита от взломов и утечек данных.
* Оптимизация производительности: Улучшение скорости и эффективности работы приложения, включая оптимизацию запросов к базам данных и использование кэширования данных.

Таким образом, backend-разработчику приходится поддерживать весь набор веб-систем, исключая разве что функционирование самого веб-сервера и каналов связи (это уже задача системного администратора).

# Практическая часть

## Анализ предметной области

Для того чтобы определиться с наилучшим подходом к разработке, а также необходимостью внедрения тех или иных функций необходимо определиться с целевой аудиторией. Это поможет нам понять, что именно хочет получить пользователь и на что стоит обращать внимание.

Мессенджеры — это не изобретение 21 века и уже давно широко используются людьми всех возрастов. На это повлияло несколько факторов:

1. Мессенджеры легки в эксплуатации, очень легко научить звонить или отправлять сообщения как пожилых людей, так и маленьких детей.
2. Они полностью бесплатные. В отличие от смс отправка сообщений в мессенджере полностью бесплатная, пользователю нужно беспокоиться о наличии соединения с интернетом.
3. Быстрота отправки и доставки сообщений. Мессенджеры могут предложить практически мгновенную отправку сообщений, что существенно уменьшает время ответа собеседника.

Охват целевой аудитории нашего проекта необычайно высок, ведь большинство людей на земле уже пользуются различными мессенджерами. Однако, я считаю, что наиболее разумным решением будет сосредоточиться на пользователях младше 20-25 лет. Молодое поколение всегда наиболее открыто к инновациям и нашему приложению будет проще всего привлечь именно эту аудиторию.

## Техническое задание[12]

### Введение

Работа выполняется в рамках курсового проекта по МДК 01.01 «Разработка программных модулей». В результате выполнения курсового проекта, должна быть разработан десктопный мессенджер.

### Основания для разработки

Основанием для разработки является учебный план по специальности 09.02.07. «Информационные системы и программирование», приказ на курсовой проект №

Организация, утвердившая договор: ГБПОУ «Тверской колледж имени А. Н. Коняева».

Наименование работы: Десктопный Мессенджер».

### Назначение разработки

Целью курсовой работы является разработка десктопного мессенджера.

Пользователями продукта выступают, преимущественно молодые пользователи сети интернет.

### Требования к программе или программному изделию

Функциональные требования представляют собой подробное описание того, что приложение должно делать и как оно должно взаимодействовать с внешними элементами. Они формируются исходя из потребностей пользователей и бизнес-процессов, включая в себя операции, обработку данных, а также установку ограничений и правил, которые приложение должно соблюдать. Этот этап является краеугольным в разработке, так как определяет, как приложение удовлетворит запросы пользователей и достигнет поставленных целей. Ниже представлен подробный список функциональных требований, которые должно выполнять данное приложение:

* Регистрация нового пользователя
* Авторизация
* Сохранение сессии
* Вывод в самом приложении имя авторизовавшегося пользователя
* Отображение списка друзей
* Отображение списка диалогов
* Поиск зарегистрированных пользователей
* Добавление в друзья
* Удаление из друзей
* Создание диалога
* Удаление диалога
* Отправка сообщений
* Отображение сообщений любого диалога

### Бизнес требования

Бизнес-требования представляют собой спецификацию функций и возможностей продукта или системы, необходимых для достижения бизнес-целей и удовлетворения потребностей пользователей. Они выражаются в виде конкретных функций, характеристик или условий, которые должны быть реализованы в разрабатываемом продукте. Эти требования определяют, как продукт будет поддерживать бизнес-процессы, удовлетворять потребности пользователей и достигать поставленных целей компании.

В контексте разработки десктопного мессенджера, бизнес-требования могут охватывать вопросы безопасности данных, удобства использования, и другие аспекты, необходимые для создания успешного продукта, соответствующего бизнес-потребностям и ожиданиям пользователей. Ниже приведен список функциональных требований к приложению:

* Безопасность данных: Обеспечение конфиденциальности сообщений, защита личных данных пользователей и шифрование передаваемой информации.
* Удобство использования: Интуитивный интерфейс, простота навигации и удобство взаимодействия с приложением.
* Функциональные возможности: Разнообразные функции, такие как отправка сообщений, добавление и удаление друзей и т.д.
* Поддержка и обновления: Гарантия регулярных обновлений для исправления ошибок, улучшения безопасности и добавления новых функций, если требуется.

Эти бизнес-требования помогут определить основные цели и ожидания от разрабатываемого десктопного мессенджера.

### Требования к программной документации

В ходе разработки веб-сервиса должны быть подготовлены: текст программы, её описание, программа и методика испытаний, руководство пользователя и технико-экономическое обоснование

### Технико-экономическое обоснование

Наше приложение позволит получить лучший опыт в использовании мессенджеров.

### Требования к надёжности

Приложение должно функционировать без ошибок, работать стабильно и выполнять все заложенные функции.

### Требования к составу и параметрам технических средств

* Процессор с тактовой частотой 800 MHz или более мощный
* Оперативная память 8 гб или больше
* Свободное место на жёстком диске от 500 Мб
* Архитектура с разрядностью 32 бит или 64 бит (x86 или x64)
* Операционная система Windows 10 и выше

### Требования к программной документации

В процессе разработки десктопного приложения необходимо подготовить: текст программы, её описание, само приложение и методика тестирования, руководство пользователя и технико-экономическое обоснование.

### Потребность в продукте

Не смотря на обильное количество аналогов на рынке не выделилось единого монополиста, что позволяет даже не очень крупным проектам собирать собственную аудиторию. Совмещая приятный удобный дизайн, стабильность работы и интересные функции приложение может снискать популярность.

### Порядок контроля и приёмки

Продукт проходит проверку работоспособности и функций на основе тестовых данных. Интерфейс проверяется на соответствие техническому заданию. Продукт принимается, если он соответствует установленным требованиям.

## Разработка функционала и базы данных [6]

Невозможно представить работоспособный мессенджер, в котором не использовались бы базы данных. Для ее создания необходимо проанализировать предметную область и определиться с сущностями и их атрибутами. Я выделила следующие сущности:

* Пользователи
* Друзья
* Сообщения
* Диалоги
* Участники диалогов

На основе этих данных был создан словарь БД Мессенджера(Таблица 1)

Таблица 1 – Словарь данных для БД «Мессенджер»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table | Field | Format | Key | FK Reference |
| Users | UserID | int |  |  |
|  | Name | varchar(255) |  |  |
|  | Email | varchar(255) |  |  |
|  | ProfilePhoto | blob |  |  |
|  | Password | varchar(255) |  |  |
| Friends | FriendshipID | int |  |  |
|  | UserID1 | int |  |  |
|  | UserID2 | int |  |  |
| Messages | MessageID | int |  |  |
|  | DialogID | int |  |  |
|  | SenderID | int |  |  |
|  | ReceiverID | int |  |  |
|  | Content | text |  |  |
|  | Image | blob |  |  |
|  | ImageMimeType | varchar(50) |  |  |
|  | Timestamp | timestamp |  |  |
| Dialogs | DialogID | int |  |  |
|  | Name | varchar(255) |  |  |
| DialogParticipants | ParticipantID | int |  |  |
|  | DialogID | int |  |  |
|  | UserID | int |  |  |

### Логическая схема БД

В результате назначения внешних ключей была построена логическая модель данных, представленная на рисунке 1.

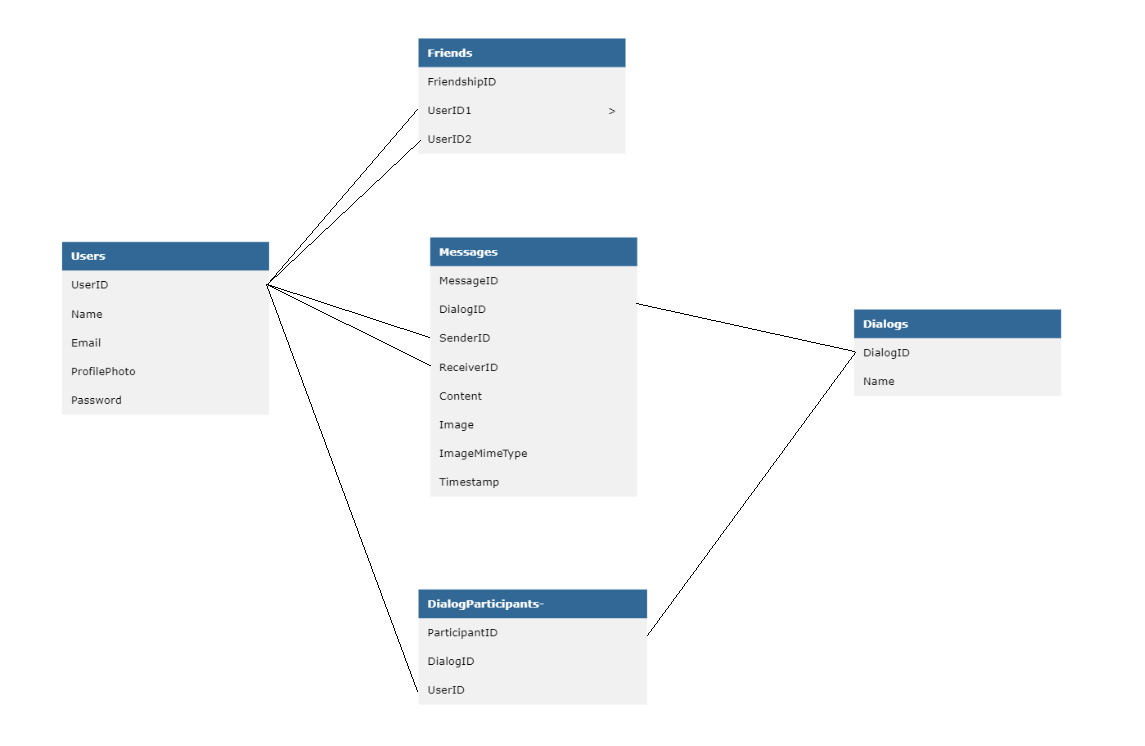


Рисунок 1 - Логическая схема БД

### Физическая схема БД

Физическая модель данных, обычно производная от логической, отображает каждый элемент логической модели в конкретные структуры. В этой модели каждой сущности из логической модели соответствует таблица, каждому экземпляру сущности — строка в таблице, а каждому атрибуту — колонка в таблице. Таким образом, физическая модель представляет конкретные детали хранения данных на уровне таблиц, строк и колонок (Рисунок 2).

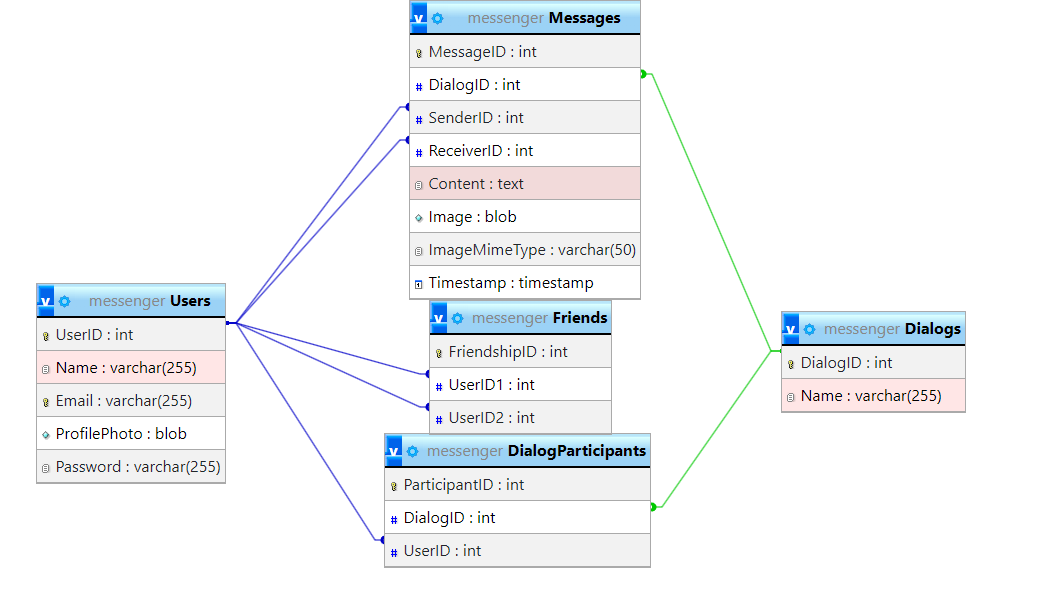


Рисунок 2 – Логическая схема БД

## Создание интерфейса [7]

### Блок “Начните общаться”

Это меню появляется перед глазами каждого пользователя, запускающего приложение (Рисунок 3) (Рисунок 4).

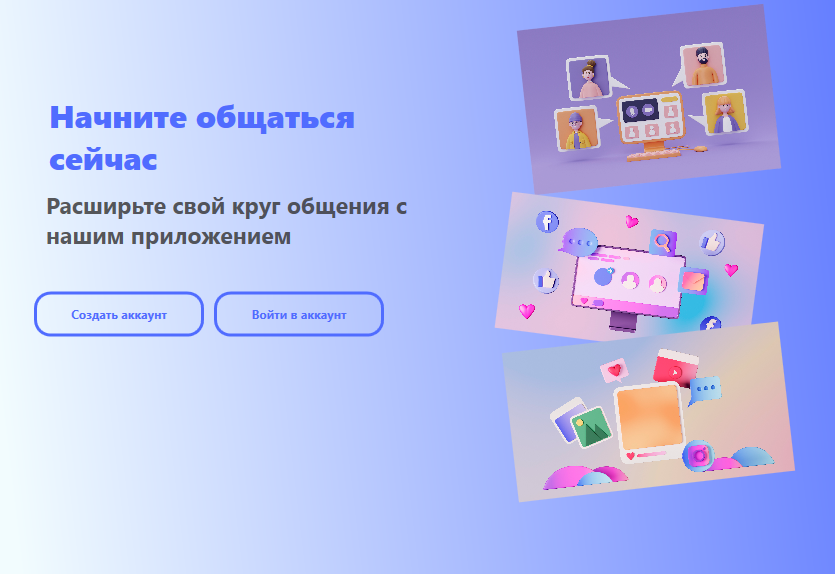


Рисунок 3 – Блок “Начните общаться”

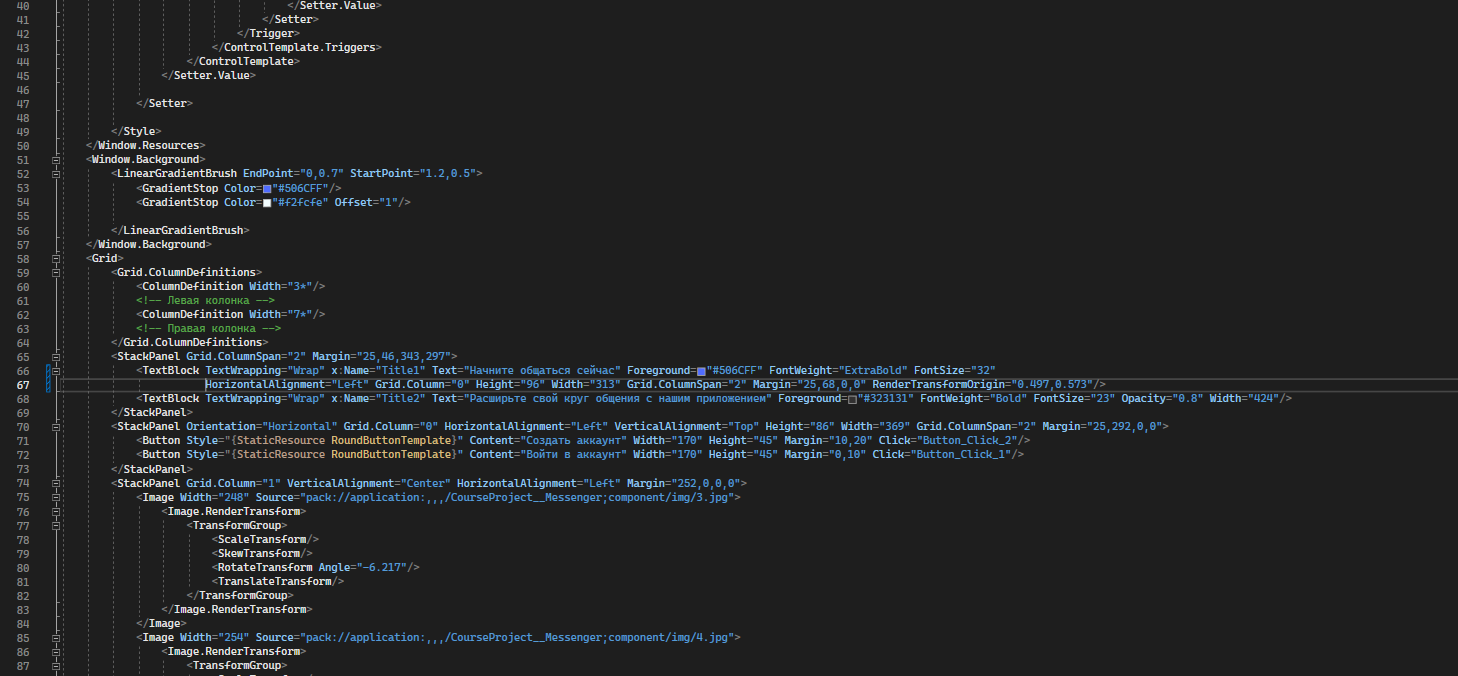


Рисунок 4 – Разметка “Блок “Начните общаться””

### Регистрация

Этот блок позволит новым пользователям зарегистрироваться в приложении(Рисунок 5) (Рисунок 6).

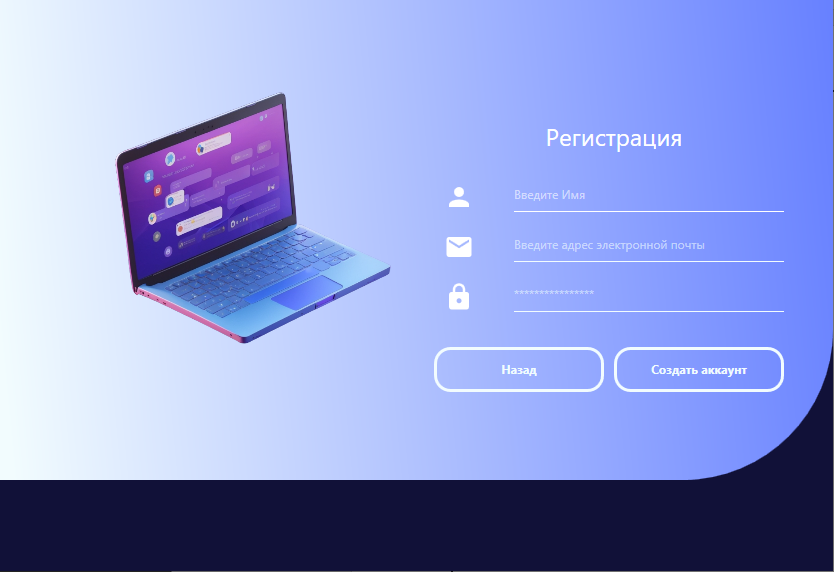


Рисунок 5 – Регистрация

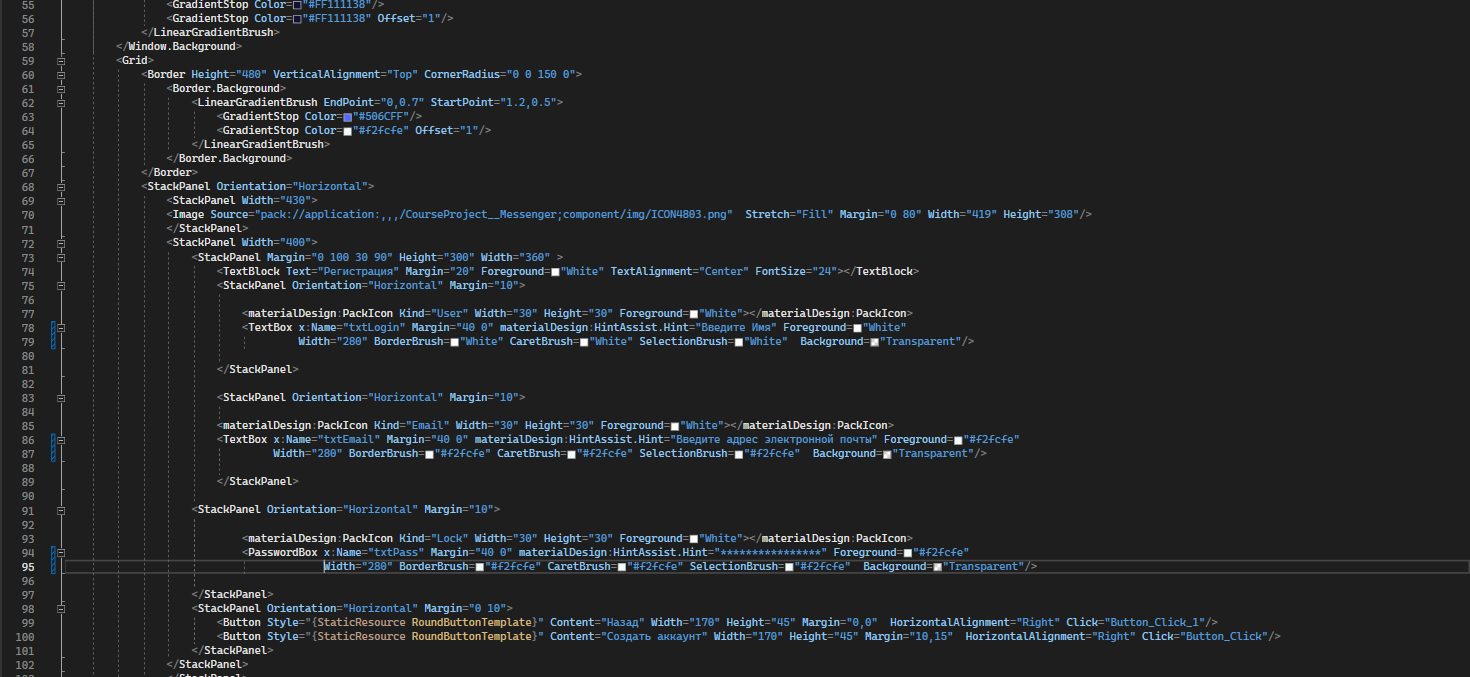


Рисунок 6 – Разметка” Регистрация”

### Авторизация

В этом блоке ранее зарегистрированные пользователи могут войти в свои аккаунты (Рисунок 7) (Рисунок 8).

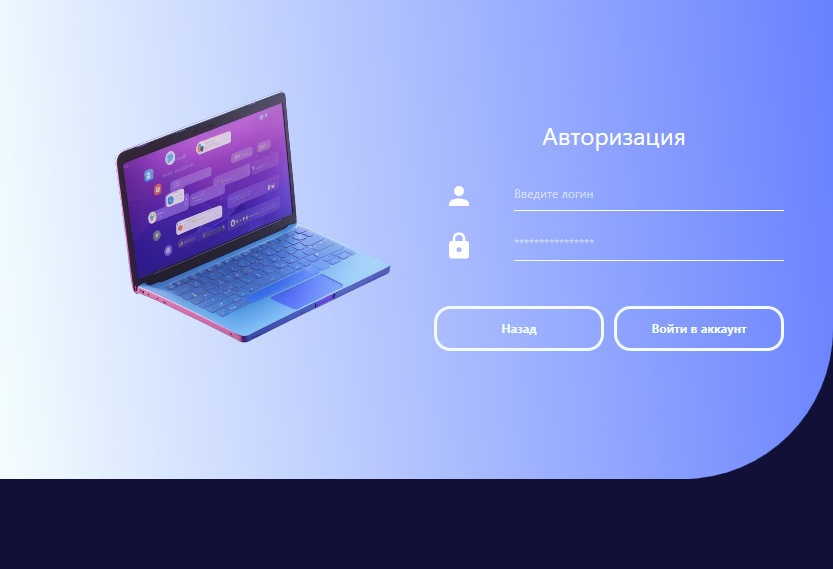


Рисунок 7 – Авторизация

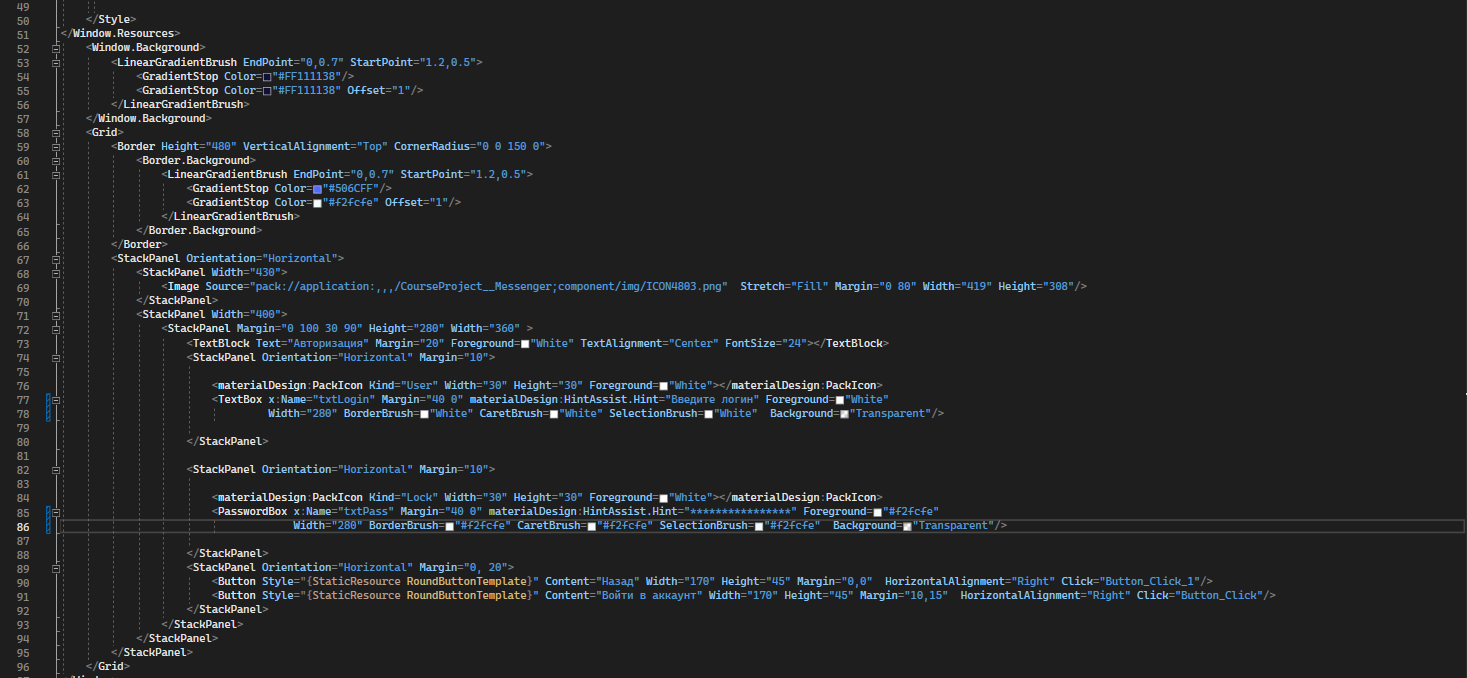


Рисунок 8 – Разметка “Авторизация”

### Блок Чат

Основной блок приложения. Здесь располагается список друзей и окно чата для общения с другими пользователями (Рисунок 9) (Рисунок 10).

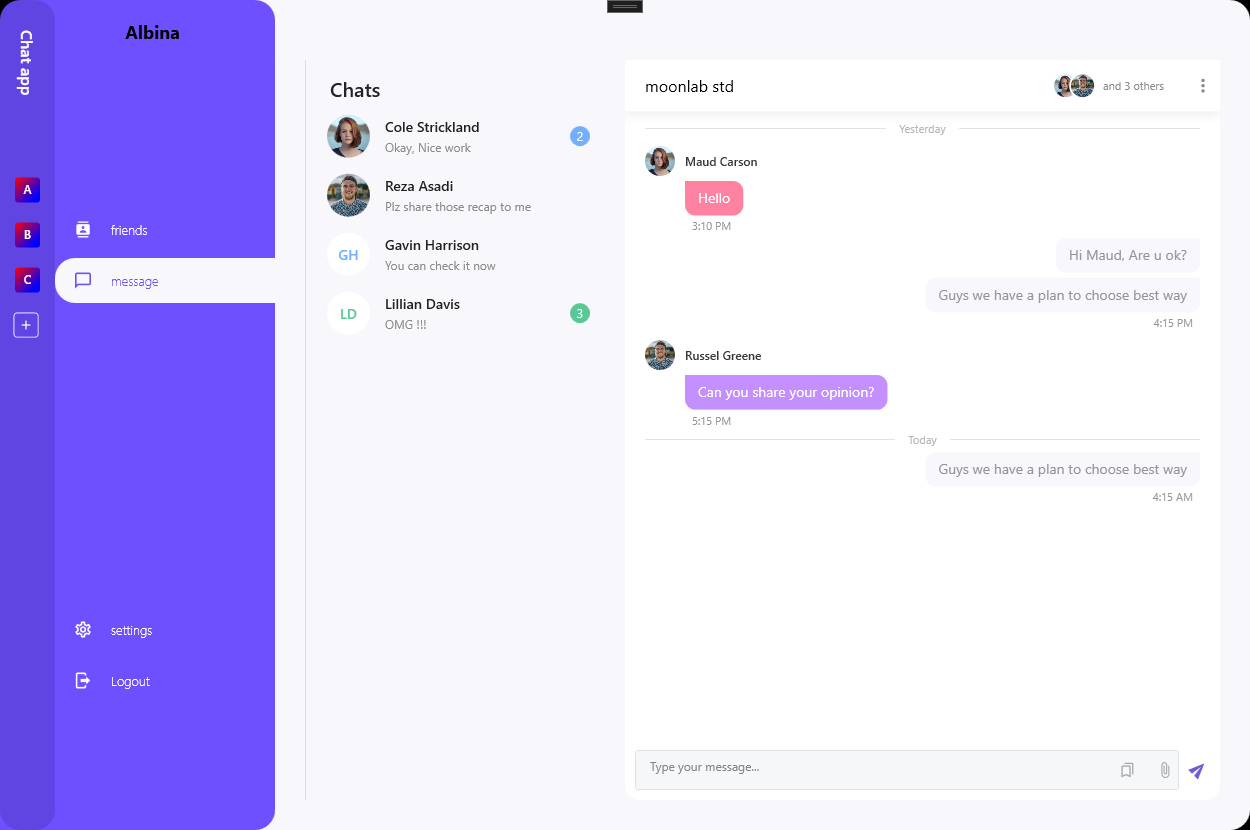


Рисунок 9 – Чат

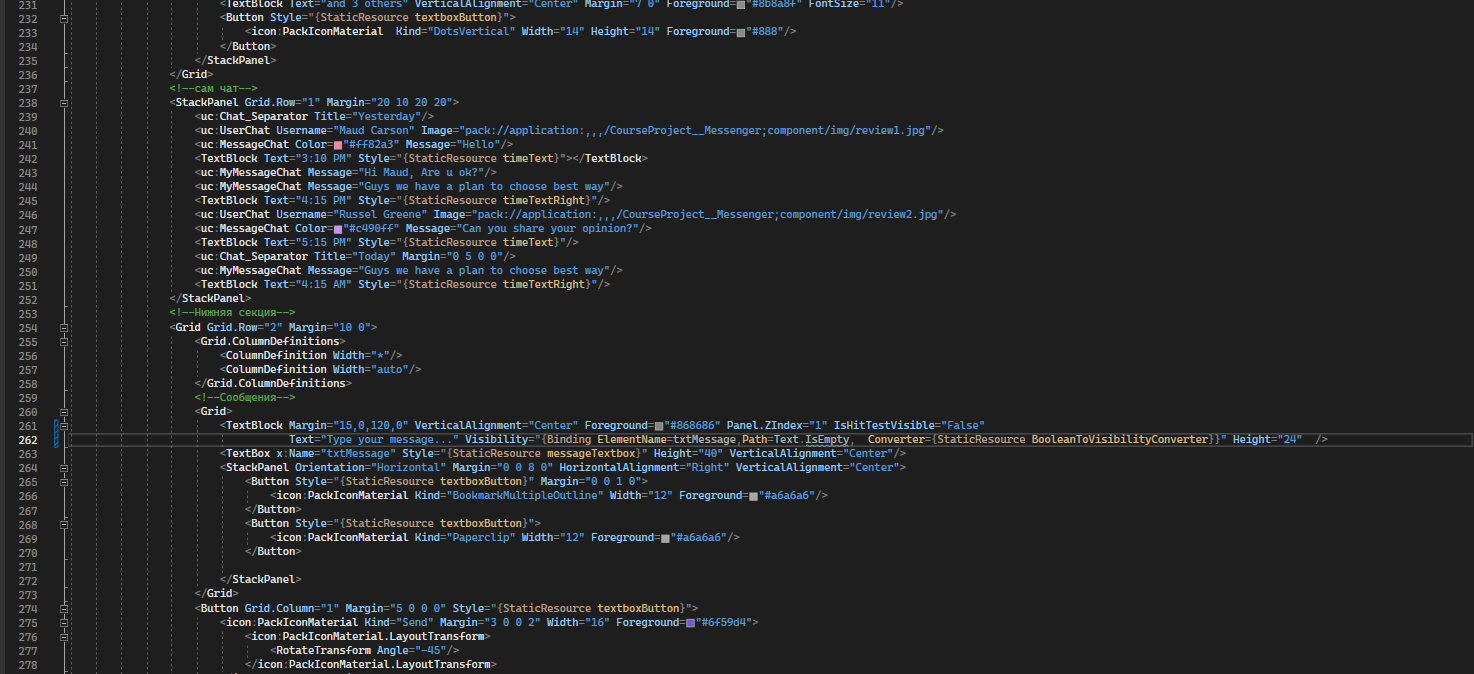


Рисунок 10 – Разметка “Чат”

## Тестирование [5]

Тестирование программного обеспечения представляет собой процесс проверки соответствия фактического поведения программы ожидаемому результату на основе определенного набора тестов. Это не только метод контроля качества, но и комплексная техника, охватывающая планирование, дизайн тестов, их выполнение и анализ результатов. Разнообразные виды тестирования, такие как модульное, интеграционное, системное, и другие, способствуют обеспечению надежности и эффективности программного продукта. Тестирование становится ключевым этапом в жизненном цикле разработки, гарантируя соответствие программы требованиям и высокий стандарт качества. Существует несколько видов тестирования программного обеспечения, каждый из которых направлен на проверку определенных аспектов приложения. Некоторые из основных видов тестирования:

* Функциональное тестирование: это проверка того, как программное обеспечение выполняет заданные функции. Проверяется соответствие функций программы ее спецификациям. Примеры включают тестирование пользовательского интерфейса, работы функций и обработки данных.
* Нагрузочное тестирование: Оценка производительности приложения под различными нагрузками. Цель - узнать, как приложение ведет себя при большом количестве пользователей или высокой нагрузке данных, чтобы предотвратить сбои или замедления работы.
* Интеграционное тестирование: это проверка взаимодействия между различными частями приложения. Цель - убедиться, что все компоненты работают вместе корректно. Примеры включают проверку взаимодействия модулей или API.
* Модульное тестирование: Проверка отдельных модулей или компонентов приложения на корректность их работы. Это тестирование проводится независимо от других компонентов для проверки их функциональности и корректности работы.

Так же глобально можно разделить типы тестирования на ручное и автоматическое. В рамках данного проекта было принято решение использовать ручное тестирование. Оно обладает некоторыми несомненными плюсами по сравнению с автомотическим:

* Гибкость: Ручное тестирование позволяет адаптироваться к изменениям в приложении без необходимости переписывания тестов.
* Интуитивность: Тестировщик может использовать свой опыт и интуицию для выявления неочевидных проблем.
* Эффективность в начале проекта: В начале разработки, когда приложение еще нестабильно, ручное тестирование может быть более эффективным.

### Тест-план

#### ID

Тестирование десктопного мессенджера.

#### Введение

Этот документ представляет собой план тестирования для десктопного мессенджера. Он содержит информацию о стратегии и методиках, которые будут использоваться для проверки качества приложения. Этот план необходим для обеспечения высокого уровня качества и подтверждения функциональности мессенджера.

#### Объекты тестирования

1. Блок “Начните общаться”
   1. Кнопка “Cоздать аккаунт”
   2. Кнопка “Войти в аккаунт”
2. Регистрация
   1. Поле “Логин”
   2. Поле “Email”
   3. Поле “Пароль”
   4. Кнопка “Назад”
   5. Кнопка “Создать аккаунт”
   6. Создание аккаунта
3. Авторизация
   1. Поле “Логин”
   2. Поле “Пароль”
   3. Кнопка “Назад”
   4. Кнопка “Войти аккаунт”
4. Блок “Сообщения”
   1. Кнопка “Friends”
   2. Поле выбора диалога
   3. Кнопка удаления диалога
   4. Поле ввода Сообщений
   5. Поле вывода Сообщений
   6. Кнопка “Отправить”
5. Блок “Друзья”
   1. Кнопка “Message”
   2. Список контактов
   3. Кнопка удаления друга
   4. Поле поиска контактов
   5. Кнопка Добавить друга

#### Что будет тестироваться

* Стабильность работы
* Выполнение заложенных функций

#### Способ тестирования

Будет проводиться:

* Функциональное тестирование: Проверка основных функций мессенджера, таких как отправка и получение сообщений, управление контактами, создание групповых чатов и т.д.
* Тестирование пользовательского интерфейса: Убедиться, что интерфейс мессенджера интуитивно понятен и легок в использовании для пользователя.

#### Условие прохождения тестирования

Все функции работают корректно: Все основные и вспомогательные функции приложения должны быть проверены и функционировать без ошибок.

#### Результатом проведения тестирования будет являться получение чек листа и тест кейса.

### Чек лист

Чек-лист представляет собой набор шагов, задач или критериев, которые тестировщики используют для проверки функциональности, производительности, безопасности или других аспектов приложения. Чек-листы в тестировании мессенджеров помогают стандартизировать процесс проверки различных аспектов приложения, обеспечивая полноту и систематичность в оценке его функциональности и качества(Таблица 2).

Таблица 2 – Чек-Лист

|  |  |  |
| --- | --- | --- |
| Проверка | Результат | Комментарии |
| Запуск приложения | + |  |
| Регистрация | + |  |
| Авторизация | + |  |
| Отображение имени пользователя | + |  |

Таблица 2 продолжение

|  |  |  |
| --- | --- | --- |
| Проверка | Результат | Комментарии |
| Переход в раздел “Friends“ | + |  |
| Поиск контактов | + |  |
| Добавление в друзья | + |  |
| Начало нового диалога | + |  |
| Переход в раздел “Message” | + |  |
| Отправка сообщения | + |  |
| Удаление диалога | + |  |
| Удаление контакта | + |  |

### Тест-кейс

Тест-кейс — это способ документирования проверки, которую осуществляет тестировщик. Это набор инструкций или последовательность шагов, по которым планируется проверять функциональность программы. В тест-кейсе детально описываются этапы подготовки к тестированию, сам процесс проверки и ожидаемый результат. Это своего рода план действий, по которому осуществляется тестирование уже созданного программного продукта.

#### Тест-кейс Регистрация

Этим тест-кейсом нам необходимо проверить корректность работы регистрации в приложении

Таблица 4 – Тест-кейс 1

|  |  |
| --- | --- |
| Наименование проекта | Десктопный мессенджер |
| Номер версии | 1.0 |
| Имя тестера | Джорубова Альбина |
| Даты тестирования | 10.12.2023 |
| Test Case # | 1 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка авторизации |
| Резюме испытания | Тестирование регистрации на наличие ошибок |
| Шаги тестирования | 1. Открыть приложение 2. Нажать на кнопку “Создать аккаунт” 3. Ввести email, логин и пароль 4. Нажать кнопку “Создать аккаунт” |
| Данные тестирования | Логин – User1111  Email – [q11@bk.ru](mailto:q11@bk.ru)  Пароль – Qwerty100500 |
| Ожидаемый результат | Регистрация выполнена успешна |
| Фактический результат | Регистрация выполнена успешна |
| Статус | Пройдено |

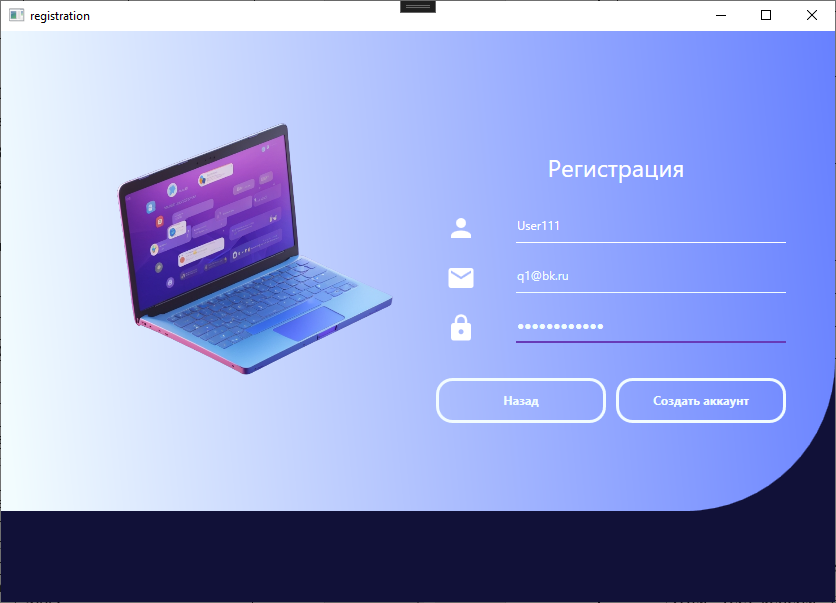


Рисунок 11 – Заполненная страница Регистрации

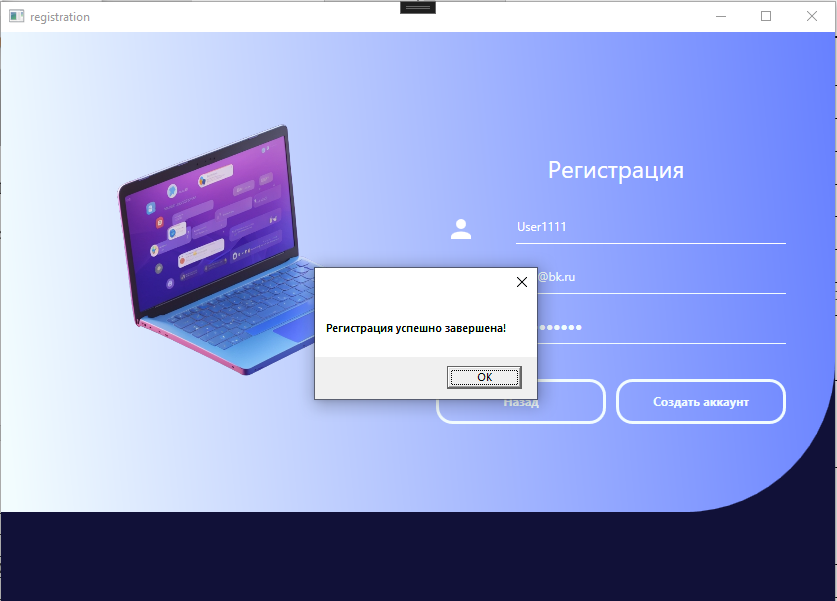


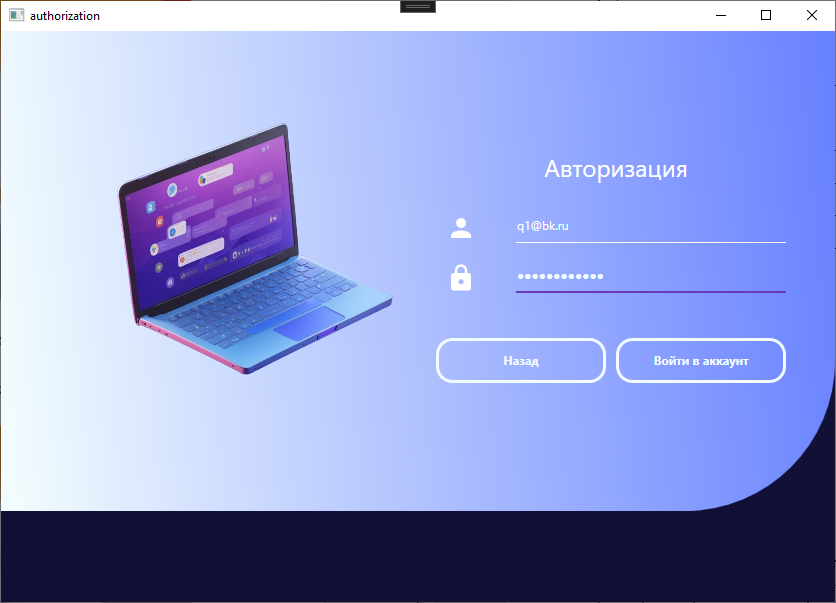
Рисунок 12 – Успешная регистрация

#### Тест-кейс Авторизация

Этим тест-кейсом нам необходимо проверить корректность работы авторизации в приложении

Таблица 3 – Тест-кейс 2

|  |  |
| --- | --- |
| Наименование проекта | Десктопный мессенджер |
| Номер версии | 1.0 |
| Имя тестера | Джорубова Альбина |
| Даты тестирования | 10.12.2023 |
| Test Case # | 2 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка авторизации |
| Резюме испытания | Тестирование авторизации на наличие ошибок |
| Шаги тестирования | 1. Открыть приложение 2. Нажать на кнопку “Войти в аккаунт” 3. Ввести логин и пароль 4. Нажать кнопку “Войти в аккаунт” |
| Данные тестирования | Логин – q1@bk.ru  Пароль – Qwerty100500 |
| Ожидаемый результат | Авторизация выполнена успешна |
| Фактический результат | Авторизация выполнена успешна |
| Статус | Пройдено |

Рисунок 13 – Заполненная страница авторизации

#### Тест-кейс добавление контактов

Этим тест-кейсом нам необходимо проверить корректность работы добавления контактов в приложении.

Таблица 4 – Тест-кейс 3

|  |  |
| --- | --- |
| Наименование проекта | Десктопный мессенджер |
| Номер версии | 1.0 |
| Имя тестера | Джорубова Альбина |
| Даты тестирования | 10.12.2023 |
| Test Case # | 3 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка добавления контактов |
| Резюме испытания | Тестирование добавления контактов на наличие ошибок |
| Шаги тестирования | 1. Открыть приложение 2. Нажать на кнопку “Войти в аккаунт” 3. Ввести логин и пароль 4. Нажать кнопку “Войти в аккаунт” 5. Перейти во вкладку Friends 6. Выбрать из списка пользователя и нажать “Добавить в друзья” |
| Данные тестирования | Логин – q1@bk.ru  Пароль – Qwerty100500 |
| Ожидаемый результат | Контакт успешно добавлен |
| Фактический результат | Контакт успешно добавлен |
| Статус | Пройдено |

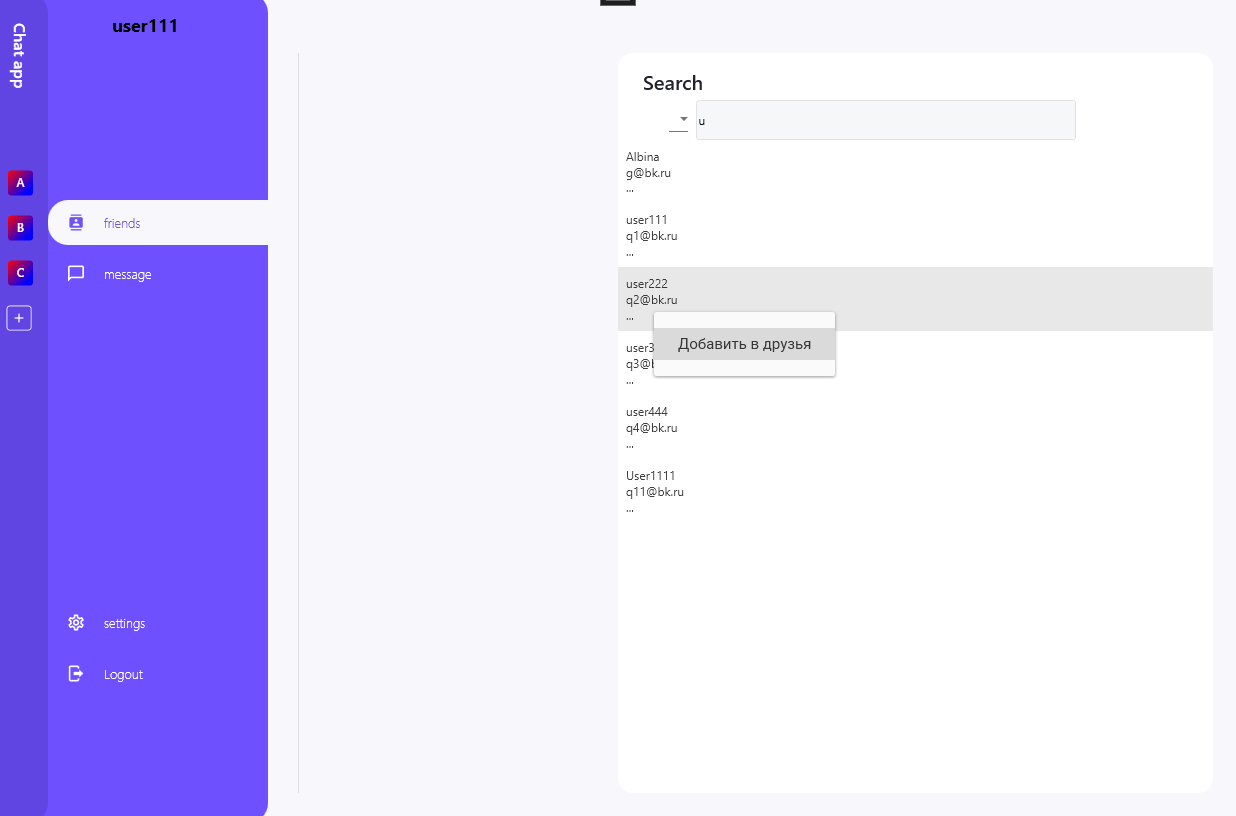


Рисунок 14 – Добавление в друзья

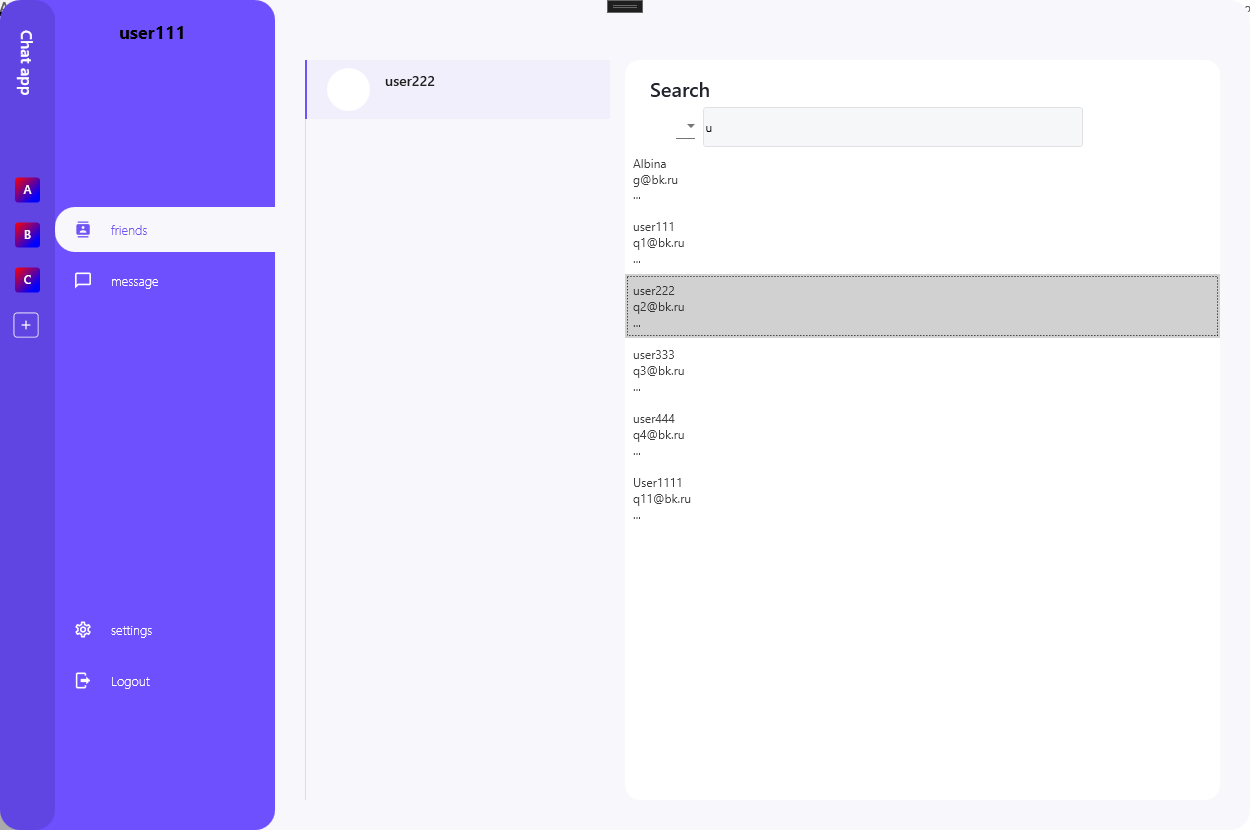


Рисунок 15 – Успешное добавление в друзья

#### Тест-кейс создание диалога

Этим тест-кейсом нам необходимо проверить корректность работы создания диалога в приложении.

Таблица 5 – Тест-кейс 4

|  |  |
| --- | --- |
| Наименование проекта | Десктопный мессенджер |
| Номер версии | 1.0 |
| Имя тестера | Джорубова Альбина |
| Даты тестирования | 10.12.2023 |
| Test Case # | 3 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка добавления контактов |
| Резюме испытания | Тестирование авторизации на наличие ошибок |
| Шаги тестирования | 1. Открыть приложение 2. Нажать на кнопку “Войти в аккаунт” 3. Ввести логин и пароль 4. Нажать кнопку “Войти в аккаунт” 5. Перейти во вкладку Friends 6. Выбрать пользователя в списке друзей и нажать “Начать диалог” |
| Данные тестирования | Логин – q1@bk.ru  Пароль – Qwerty100500 |
| Ожидаемый результат | Диалог успешно создан |
| Фактический результат | Диалог успешно создан |
| Статус | Пройдено |

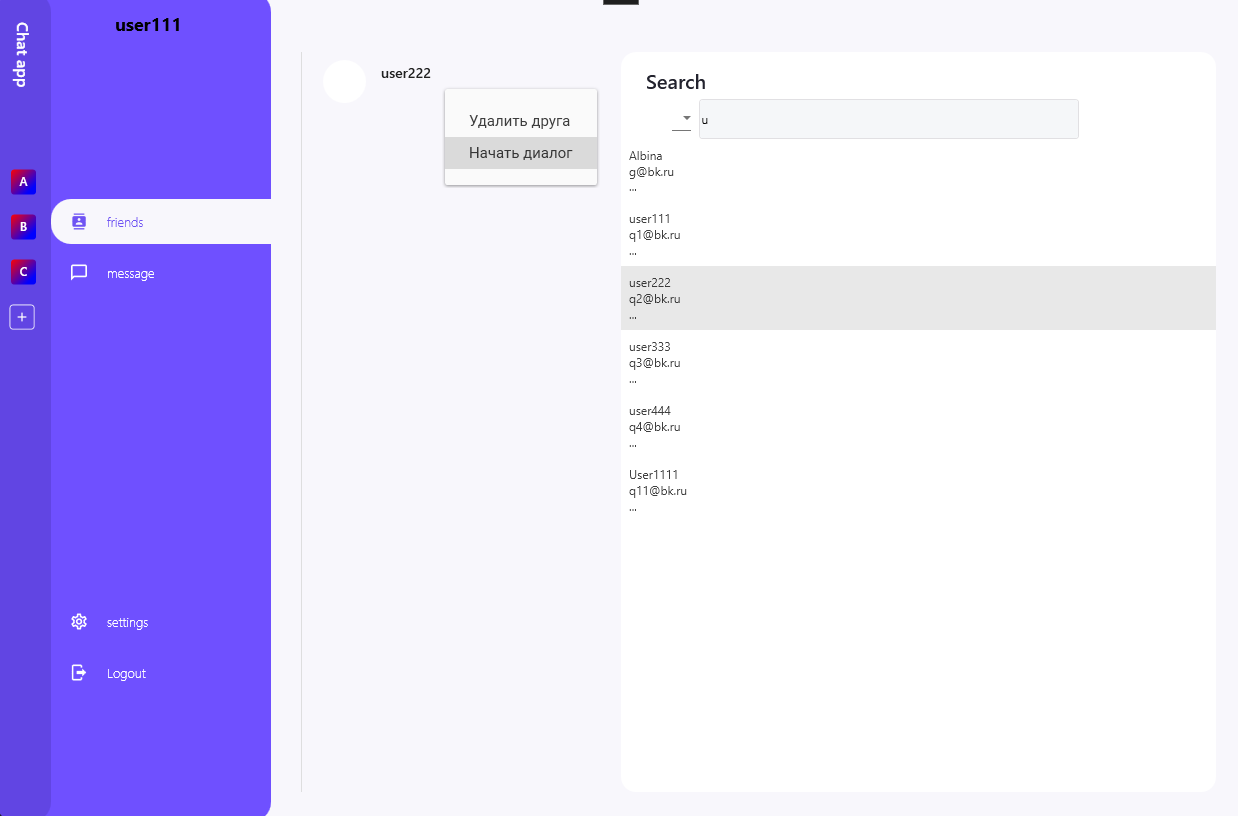


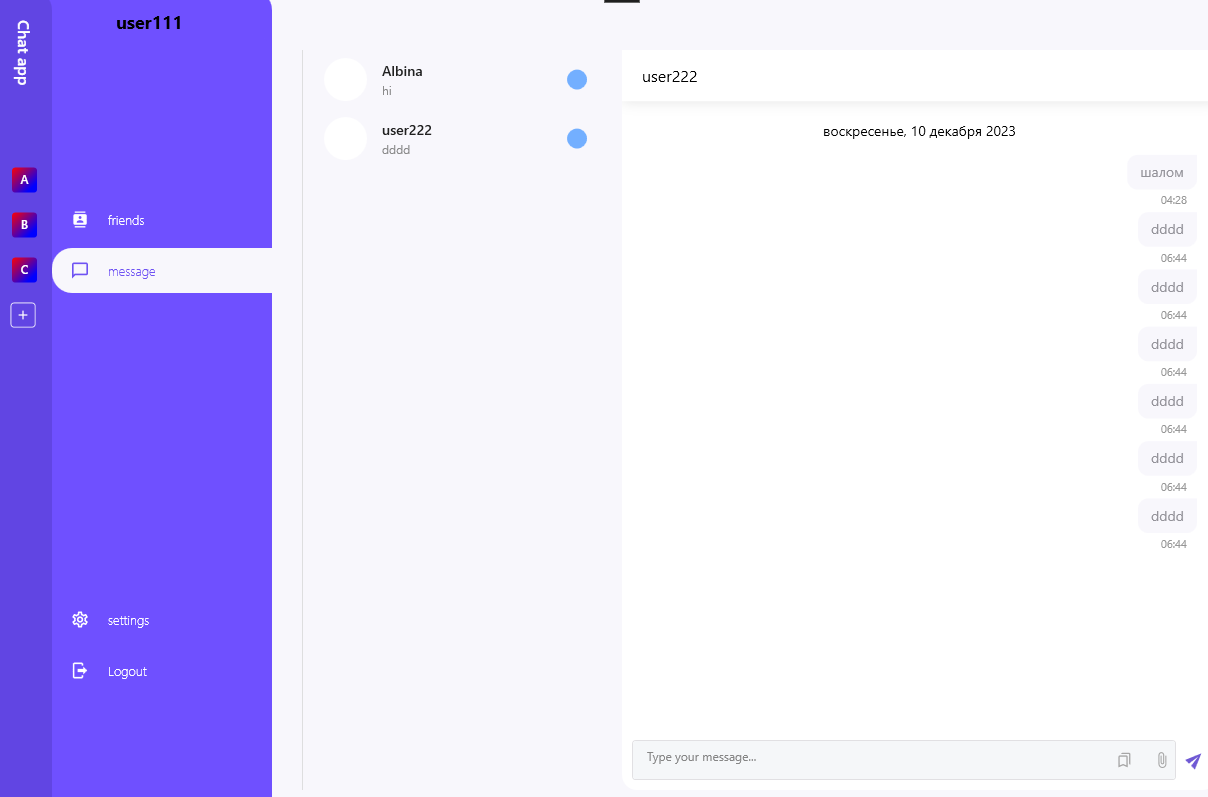
Рисунок 16 – Создание диалога

Рисунок 17 – Диалог успешно создан

# Заключение

В процессе работы над десктопным мессенджером был основательно изучен материал, связанный с принципами функционирования подобных приложений. После этого был проведен анализ предметной области и сформулировано техническое задание. Мы разработали структуру мессенджера, обеспечивающего быстрый обмен сообщениями, файлами и медиаконтентом.

Основной акцент был сделан на создание интуитивно понятного и удобного интерфейса, который делает использование мессенджера максимально комфортным для пользователей. В результате был создан десктопный мессенджер, который обеспечивает надежную и безопасную коммуникацию, предоставляет широкий функционал и удобство в общении.

В ходе работы все поставленные задачи и цели были выполнены.

# Список литературы и источников

1. ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам.
2. ГОСТ 19.201-78. ЕСПД. Техническое задание. Требования к содержанию и оформлению.
3. ГОСТ 19.402-78 Единая система программной документации. Описание программы.
4. ГОСТ 19.102-77 Единая система программной документации. Стадии разработки
5. ГОСТ 19.301-79 Единая система программной документации. Программа и методика испытаний. Требования к содержанию и оформлению
6. Ильин И. В. Базы данных : учебное пособие / И. В. Ильин, О. Ю. Ильяшенко. – Санкт-Петербург : СПбГПУ, 2020. – 96 с. – ISBN 978-5-7422-7101-7
7. Документация по C# [Электронный ресурс]. 2023. URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/