**Мы работаем над клиент-серверным приложением мессенджера.**

Моя подзадача в работе над созданием мессенджера - работа над механизмом обмена файлами между клиентами.

**Составлена архитектура мессенджера:**

***Серверная часть:***

Описание: Консольное приложение, отвечающее за обработку подключений клиентов, обмен сообщениями и файлами.

Хранение данных:

Информация о пользователях (логины и хэшированные пароли) хранится в базе данных MSSQL Server.

Сообщения хранятся в файле в директории сервера.

Файлы хранятся в корневом каталоге сервера в зашифрованном виде.

Функции:

Обработка подключений клиентов.

Прием и отправка сообщений между клиентами.

Прием и отправка файлов между клиентами.

Аутентификация пользователей с проверкой хэшей паролей в базе данных.

Шифрование и расшифровка файлов перед отправкой и после получения.

***База данных MSSQL Server:***

Описание: Хранит информацию о пользователях.

Структура данных:

Таблица пользователей с полями: id, username, password\_hash.

***Клиентская часть на WPF:***

Авторизация и Регистрация:

При запуске приложения открывается окно авторизации.

Пользователь вводит свой логин и пароль.

Если пользователь нажимает "Login", клиент отправляет запрос на сервер для авторизации (LOGIN).

Если пользователь нажимает "Sign In", открывается окно регистрации.

При регистрации нового пользователя клиент отправляет запрос на сервер с данными для создания нового аккаунта (SIGN IN).

После успешной авторизации или регистрации окно авторизации закрывается, и открывается окно со списком подключенных клиентов.

Список подключенных клиентов:

В окне со списком подключенных клиентов отображается список всех активных пользователей.

Пользователь может выбрать клиента из списка для начала общения.

Для подключения к выбранному клиенту пользователь нажимает кнопку "Connect".

После подключения к выбранному клиенту открывается диалоговое окно для обмена сообщениями и файлами.

Диалоговое окно для обмена сообщениями и файлами:

В диалоговом окне отображается история переписки с выбранным клиентом.

Пользователь может вводить текст сообщения в соответствующее поле.

При нажатии кнопки "Отправить сообщение" введенное сообщение отправляется выбранному клиенту.

При нажатии кнопки "Отправить файл" открывается стандартное окно выбора файла.

После выбора файла пользователь подтверждает отправку файла выбранному клиенту.

После отправки файла серверу, он отображается в истории переписки с информацией о времени отправки, отправителе и размере файла.

При нажатии кнопки "Получить файл" пользователь вводит имя файла, который хочет получить.

После отправки запроса серверу, пользователь выбирает место сохранения файла через стандартное окно сохранения.

Взаимодействие с сервером:

Клиент общается с сервером посредством сетевого соединения.

Он отправляет запросы на сервер для авторизации, регистрации, отправки сообщений и файлов, а также для получения истории переписки и списка подключенных клиентов.

Вся обработка данных происходит на стороне сервера, а клиент лишь отображает информацию и отправляет запросы на сервер.

Обмен данными:

Описание: Механизм обмена сообщениями и файлами между клиентами через сервер.

Технологии:

Шифрование и расшифровка файлов перед отправкой и после получения.

Функции:

Обеспечение безопасности и конфиденциальности данных.

Проверка наличия файла у отправителя перед отправкой.

Подтверждение операций отправки и получения файлов.

Общие данные о приложении:

Язык программирования: c#

Шаблон приложения: .NET Framework

Объектно-реляционный связующий модуль для .NET: EntityFramework

Модель ADO.NET EDM: db Context

Архитектурный паттерн: MVVM (Model-View-ViewModel)

Количество портов: 1 порт для обмена файлами, 1 порт для обмена сообщениями

Количество потоков: 1 поток для отправления сообщений, 1 поток для получения сообщений, 1 поток для отправления файлов, 1 поток для получения файлов

Интерфейс клиентской части приложения: WPF (cs и xaml)