

Министерство образования Иркутской области
Государственное бюджетное профессиональное
образовательное учреждение Иркутской области
«Иркутский авиационный техникум»
(ГБПОУИО «ИАТ»)

КП.09.01.07-3.23.201.09 ПЗ

ИНФОРМАЦИОННАЯ СИСТЕМА
«СОЦИАЛЬНАЯ СЕТЬ»

Председатель ВЦК:	_____	(А.С. Александрова)
	(подпись, дата)	
Руководитель:	_____	(А.С. Александрова)
	(подпись, дата)	
Студент:	_____	(З.З. Исомадинов)
	(подпись, дата)	

Иркутск 2023

СОДЕРЖАНИЕ

Введение	3
1 Описание предметной области ИС.....	5
2 Анализ инструментальных средств разработки	6
3 Техническое задание	14
4 Проектирование информационной системы.....	15
4.1 Структурная схема информационной системы	15
4.2 Функциональная схема веб-приложения	17
4.3 Проектирование базы данных	21
4.4 Проектирование пользовательского интерфейса	26
5. Разработка информационной системы.....	28
5.1 Разработка интерфейса информационной системы.....	28
5.2 Разработка базы данных информационной системы.....	29
5.3 Разработка информационной системы	31
5.4 Тестирование информационной системы	32
6 Документирование программного продукта.....	36
6.1 Руководство пользователя ИС	36
Заключение	40
Список используемых источников.....	41
Приложение А – Техническое задание	42
Приложение Б – Листинг	47

					КП.09.01.07-3.23.201.09 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Исомадинов З.З.			ИНФОРМАЦИОННАЯ СИСТЕМА «СОЦИАЛЬНАЯ СЕТЬ» пояснительная записка	Лит.	Лист	Листов
Провер.		Александрова А.С					2	47
						ГБПОУИО «ИАТ» ВЕБ-20-1		

Введение

Социальная сеть – это онлайн-платформа, которая позволяет пользователям создавать профили, обмениваться информацией, коммуницировать и устанавливать контакты с другими пользователями. Социальные сети стали популярными вместе с развитием Интернета и широкополосного доступа к нему.

Большинство социальных сетей позволяют пользователям создавать профили, где они могут размещать информацию о себе, своих интересах, фотографии и видео. Пользователи могут добавлять других пользователей в свой список друзей.

Социальные сети стали неотъемлемой частью нашей жизни, позволяя нам оставаться на связи с друзьями и близкими, находить новых друзей и знакомых, делиться своими мыслями и идеями, а также получать информацию о мире вокруг нас. Однако, с ростом популярности социальных сетей возникают и новые проблемы, связанные с безопасностью и конфиденциальностью данных пользователей.

В связи с этим, разработка безопасной и надежной социальной сети становится все более актуальной задачей. Такая социальная сеть должна обеспечивать защиту личной информации пользователей, предотвращать несанкционированный доступ к данным и гарантировать безопасность при обмене информацией между пользователями.

Для создания такой социальной сети необходимо провести тщательное предпроектное исследование, изучить инструментальные средства разработки, спроектировать структурную и функциональную схемы веб-приложения, разработать базу данных и обеспечить безопасность и конфиденциальность данных пользователей.

В результате успешной реализации данного проекта, пользователи смогут наслаждаться общением и обменом информацией в безопасной и защищенной

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

среде, что является важным условием для создания комфортной и удобной социальной сети. Целью данного курсового проекта является разработка социальной сети, которая будет обеспечивать безопасность и конфиденциальность данных пользователей, а также удовлетворять их потребности в общении и обмене информацией.

Основными задачами данной работы являются:

- Провести предпроектное исследование.
- Провести обзор инструментальных средств.
- Разработать техническое задание.
- Спроектировать информационную систему.
- Спроектировать пользовательский интерфейс.
- Обеспечить безопасность и конфиденциальность данных пользователей.
- Разработать информационную систему.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

1 Описание предметной области ИС

Основным видом деятельности информационной системы «Социальная сеть» является создание социальных отношений между пользователями, которые имеют схожие интересы или офлайн-связи.

Информационная система – система обработки информации совместно с соответствующими организационными ресурсами, которая обеспечивает и распространяет информацию.

Информационная система предназначена для своевременного обеспечения нуждающихся людей надлежащей информацией, то есть для удовлетворения конкретных информационных потребностей в рамках определённой предметной области, при этом результатом функционирования информационных систем является информационная продукция – документы, информационные массивы, базы данных и информационные услуги.

Предметная область социальной сети охватывает широкий спектр интересов и функций, направленных на обеспечение межличностного взаимодействия и обмена информацией между пользователями.

Пользователи: социальная сеть предоставляет платформу для регистрации и создания профилей пользователям. Каждый пользователь может добавлять друзей, подписываться на интересных людей и организации, а также делиться своими мыслями, медиа контентом и жизненными событиями.

Профили и ленты: каждый пользователь имеет свой профиль, где можно указать информацию о себе.

Чаты: социальная сеть обеспечивает возможность отправки сообщений между пользователями в рамках платформы. Пользователи могут обмениваться текстовыми сообщениями.

Лайки и комментарии: пользователи могут выражать свою поддержку и интерес к контенту, оставляя лайки и комментарии к постам других пользователей.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

2 Анализ инструментальных средств разработки

Проектировать структуру информационной системы удобно через MySQL Workbench и Draw.io, а его дизайн – через онлайн-сервис Figma. Информационная система будет состоять из двух частей – клиентская и серверная. Для реализации клиентской части отлично подойдут следующие инструменты: HTML5, CSS3, NextJs. Серверная часть будет действовать на JS-фреймворке Nestjs со стандартной базой данных MongoDB.

MySQL Workbench – инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL. В проекте используется для создания ER-диаграммы БД.

Draw.io – это удобное бесплатное онлайн-приложение для создания диаграмм для рабочих процессов, BPM, организационных, сетевых диаграмм, блок-схем (флоучарты), UML и принципиальных электросхем. В проекте используется для создания прототипа страниц.

Figma – бесплатный удобный онлайн-сервис для дизайнеров, веб-разработчиков и маркетологов. Он предназначен для создания прототипов сайтов или приложений, иллюстраций и векторной графики. В редакторе можно настроить совместную работу, вносить и обсуждать правки, причём как в браузере, так и через приложение на компьютере. Популярен, в частности, для разработки прототипа и дизайна сайта или приложения. В проекте используется для создания наглядного ожидаемого дизайна проекта, а также уникальных SVG-элементов в едином стиле.

HTML – язык разметки гипертекста. Язык разметки дает браузеру необходимые инструкции о том, как отображать тексты и другие элементы страницы на мониторе. Язык HTML интерпретируется браузерами и отображается в виде документа, в удобной для человека форме.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

CSS – каскадные таблицы стилей, которые используются для определения стилей (правил) оформления документов – включая дизайн, вёрстку и вариации макета для различных устройств и размеров экрана.

Next.js - Современный фреймворк для разработки веб-приложений на основе React. Он предоставляет возможности серверного рендеринга и статической генерации, а также обладает интегрированным маршрутизатором. Next.js упрощает создание мощных, быстрых и масштабируемых веб-приложений.

Visual Studio Code (часто называемый VS Code) – это бесплатный и открытый редактор кода, разработанный компанией Microsoft. Он предоставляет множество инструментов для редактирования, отладки и управления кодом, а также имеет возможность расширения функционала с помощью плагинов. Visual Studio Code поддерживает множество языков программирования, включая JavaScript, Python, C++, Java и многие другие. Благодаря своей гибкости и открытому исходному коду, VS Code стал одним из наиболее популярных редакторов кода среди разработчиков программного обеспечения.

Веб-приложение будет содержать в себе информацию – её необходимо хранить, изменять, структурировать и использовать. Это реализуется благодаря базе данных. Были рассмотрены следующие варианты реализации СУБД:

- 1.MySQL.
- 2.SQLite.
- 3.PostgreSQL.
- 4.MongoDB.

MySQL – популярная СУБД с моделью «клиент-сервер». Он создан для обеспечения доступа к данным для других сервисов и приложений. Реляционная система хранения и управления базами данных. Почему именно она? - Все благодаря ее популярности и признанности среди крупных корпораций. Ее используют в Facebook, YouTube, Google и тысячах других IT-компаний за счёт своей простоты, скорости работы и внушительного функционала.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

SQLite – автономная база данных без сервера SQL компактная встраиваемая СУБД с исходным кодом. SQLite поддерживает динамическое типизирование данных. Другими словами, движок Sqlite становится как бы частью нашего веб-приложения. Напрямую хранит информацию в одном файле, что облегчает его копирование.

PostgreSQL – это свободно распространяемая объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows. Производительность PostgreSQL основывается на использовании индексов, интеллектуальном планировщике запросов, тонкой системы блокировок, системе управления буферами памяти и кэширования, превосходной масштабируемости при конкурентной работе.

MongoDB – это NoSQL база данных, которая хранит данные в формате BSON(бинарное представление JSON). MongoDB хорошо подходит для приложений, работающих с большим количеством неструктурированных данных, таких как логи, социальные сети и IoT-приложения. Она предлагает гибкую модель данных и масштабируемость, что позволяет обрабатывать большие объемы данных и управлять ими эффективно.

Таким образом, в качестве базы данных для будущего продукта была выбрана MongoDB, так как она предоставляет весь необходимый функционал для разработки продукта, и при этом она проста в изучении и овладении, достаточно быстрая, предоставляет возможность гибкой настройки, легко переносимая.

Для разработки программного продукта рассмотрены следующие инструментальные средства разработки программных продуктов:

1. IntelliJ IDEA.
2. Visual Studio Code.
3. Eclipse.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

IntelliJ IDEA – это интегрированная среда разработки (IDE) для языков программирования Java, Kotlin, Groovy, Scala и других. Она была разработана компанией JetBrains и выпущена в 2001 году.

IntelliJ IDEA предоставляет широкий набор инструментов для разработки приложений на Java и других языках программирования. Она обладает мощным редактором кода, который поддерживает автодополнение, рефакторинг, анализ кода и другие функции. Кроме того, IntelliJ IDEA имеет интегрированный отладчик, который позволяет отслеживать выполнение кода и находить ошибки.

Одной из главных особенностей IntelliJ IDEA является поддержка языка Kotlin, который был разработан компанией JetBrains и стал официальным языком программирования для разработки приложений на платформе Android. IntelliJ IDEA также поддерживает различные фреймворки и технологии, такие как Spring, Hibernate, Maven, Gradle и другие.

IntelliJ IDEA доступна в двух версиях: Community Edition и Ultimate Edition. Community Edition является бесплатной и содержит базовый набор функций для разработки приложений на Java и Kotlin. Ultimate Edition является платной и содержит дополнительные инструменты и функции для разработки приложений на других языках программирования, таких как Python, PHP, JavaScript и другие.

IntelliJ IDEA является одной из самых популярных IDE для разработки приложений на Java и Kotlin. Она широко используется в индустрии и среди разработчиков open-source проектов.

Visual Studio Code – это легкий и популярный редактор кода, предоставляющий множество возможностей разработки для PHP и других языков программирования.

Особенности Visual Studio Code включают:

- IntelliSense: Технология автодополнения от Microsoft, которая предлагает подсказки и автозаполнение кода на основе контекста.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

- Настраиваемая панель и закрепляемые окна: Вы можете настроить интерфейс редактора, добавлять и закреплять окна инструментов по своему усмотрению.
- Простой рабочий процесс и файловая иерархия: Удобная навигация по проектам и файлам, а также инструменты для быстрой навигации и поиска в коде.
- Мониторинг производительности в режиме реального времени: Встроенные инструменты для отслеживания производительности вашего приложения и выявления узких мест.
- Инструменты автоматизации: Возможность настройки рабочего процесса с помощью задач и сценариев автоматизации, таких как Gulp или Grunt.
- Поддержка разделенного экрана: Возможность разделить редактор на несколько панелей для одновременной работы с разными файлами.
- Список ошибок: интуитивно понятный список ошибок и предупреждений, который упрощает процесс отладки и исправления проблем в коде.
- Visual Studio Code поддерживается на различных операционных системах, включая Windows, macOS и Linux, что делает его доступным для широкого круга разработчиков.

Таким образом, Visual Studio Code является мощным и гибким редактором кода, который обладает богатым набором функций и предоставляет удобную среду разработки для JavaScript и других языков программирования.

Eclipse – это интегрированная среда разработки (IDE) для различных языков программирования, включая Java, C++, Python, PHP и другие. Она была разработана компанией Eclipse Foundation и выпущена в 2001 году.

Eclipse предоставляет широкий набор инструментов для разработки приложений, включая мощный редактор кода, отладчик, систему контроля версий, инструменты для автоматической генерации кода и многое другое. Она также поддерживает различные фреймворки и технологии, такие как Spring, Hibernate, Maven, Gradle и другие.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

Одной из главных особенностей Eclipse является ее расширяемость. Она позволяет разработчикам создавать свои собственные плагины и интегрировать их в среду разработки. Это делает Eclipse очень гибкой и позволяет ей быть использованной для разработки приложений в различных областях.

Eclipse доступна в нескольких версиях, включая Eclipse IDE for Java Developers, Eclipse IDE for C/C++ Developers, Eclipse IDE for PHP Developers и другие. Каждая версия содержит набор инструментов и функций, оптимизированных для разработки приложений на конкретном языке программирования.

Eclipse является одной из самых популярных IDE для разработки приложений на Java и других языках программирования. Она широко используется в индустрии и среди разработчиков open-source проектов.

Для создания программного обеспечения было решено использовать следующие технологии Visual Studio Code, MongoDB, Workbench, draw.io.

Для выбора инструментов разработки были произведены сравнения, результат которых представлены в таблицах 1, 2.

Таблица 1 – Сравнение IDE для разработки программного продукта

IDE	Языки программирования	Операционная система	Бесплатная лицензия	Поддержка GiT	Поддержка отладки
Visual Studio Code	JavaScript, TypeScript, Python, Java, C++, и другие	Windows, macOS, Linux	Да	Да	Да
IntelliJ IDEA	Java, Kotlin, Groovy, Scala, и другие	Windows, macOS, Linux	Да, так же есть платная	Да	Да
Eclipse	Java, C/C++, PHP, и другие	Windows, macOS, Linux	Да	Да	Да

После тщательного анализа и сравнения различных средств разработки, мы приняли решение использовать Visual Studio Code. Это был важный выбор,

который был сделан на основе многих факторов, таких как функциональность, удобство использования, наличие необходимых инструментов и технологий, а также поддержка сообщества разработчиков. Мы уверены, что это решение поможет нам значительно повысить эффективность и качество нашей работы.

Таблица 2 – Сравнение средств реализации базы данных

Название СУБД	MySQL	SQLite	PostgreSQL	MongoDB
Тип базы данных	Реляционная	Встраиваемая	Объектно-реляционная	NoSQL
Большое кол-во типов данных	+	-	+	+
Отказоустойчивость	-	-	+	+
Требуется удаленный сервер	+	-	+	-
Простота использования	-	+	-	+
Портативность	-	+	-	+
Гибкая модель данных	-	-	-	+
Масштабируемость	-	-	+	+

Таким образом, после анализа представленных средств проектирования, было принято решение использовать MongoDB, так как оно обладает большим количеством функций и удобно в работе.

После рассмотрения всех инструментов для создания программного продукта было решено использовать следующие средства:

1) Для создания структурных схем, контекстной и диаграмм декомпозиции использовались CASE-средства – Draw.io.

2) Для наглядного составления структуры базы данных использовался инструмент для визуального проектирования баз данных – MySQL Workbench.

3) Для разработки дизайна web-приложения использовался онлайн-сервис для разработки дизайна и прототипа сайта или приложения – Figma.

4) На этапе разработки программного продукта использовались инструменты и среды для разработки: HTML5, CSS3, JS, NextJs, NestJs а также редактор кода Visual Studio Code.

5) Для структурирования, чтения, изменения и удаления информации использовалась база данных MongoDB.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

3 Техническое задание

В начале разработки создавалось техническое задание, в котором указывались основные требования.

Для создания технического задания использовался стандарт ГОСТ 34.602-2020.

Согласно ГОСТ 34.602-2020 техническое задание должно включать следующие разделы:

1. общие сведения;
2. назначение и цели создания системы;
3. требования к системе в целом;
 - 3.1. требования к структуре и функционированию системы;
 - 3.2. требования к надежности;
 - 3.3. требования к безопасности;
 - 3.4. требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы;
4. требования к документированию;
5. состав и содержание работ по созданию системы.

Техническое задание на разработку информационной системы представлено в приложении А.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

4 Проектирование информационной системы

4.1 Структурная схема информационной системы

В процессе разработки информационной системы очень важно создавать диаграммы, которые помогут лучше понять её структуру и работоспособность. Одна из таких диаграмм – это диаграмма вариантов использования, которая показывает, какие действия могут выполнять пользователи системы и как они взаимодействуют с ней. Прецеденты – это конкретные возможности, которые предоставляет система пользователям для достижения нужного им результата. На рисунке 1 показана диаграмма вариантов использования для информационной системы "Социальная сеть", которая отображает действия, которые могут выполнять администраторы и пользователи.



Рисунок 1 – Диаграмма прецедентов

Диаграмма деятельности – UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний. Под деятельностью

понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений. На рисунке 2 изображена диаграмма деятельности.

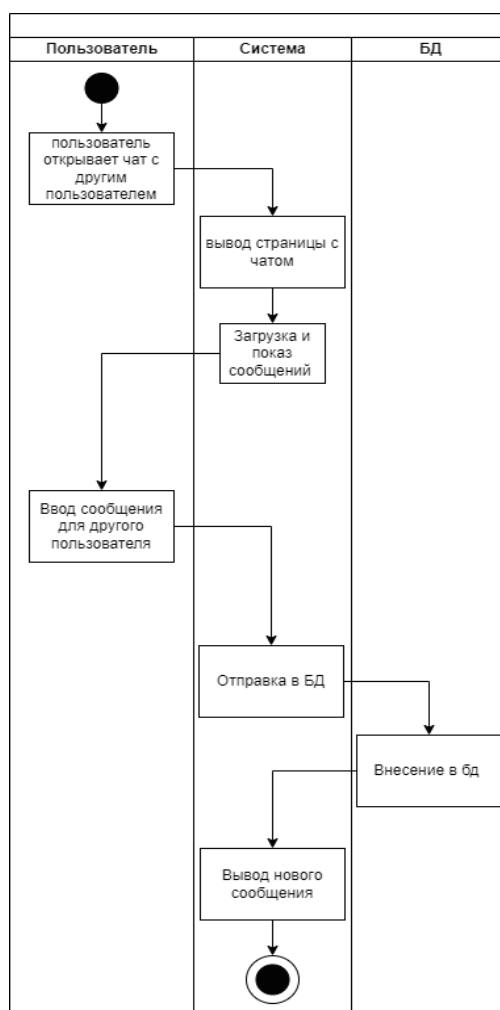


Рисунок 2 – Диаграмма деятельности

В итоге проектирования диаграммы деятельности были выделены основные возможные действия пользователя с программным продуктом.

Диаграмма компонентов предназначена для распределения классов и объектов по компонентам при физическом проектировании системы. Компоненты на диаграмме компонентов представляют собой физические модули программного кода. Обычно они в точности соответствуют пакетам на диаграмме пакетов; таким образом, диаграмма компонентов отражает выполнение каждого пакета в системе. На рисунке 3 изображена диаграмма компонентов.

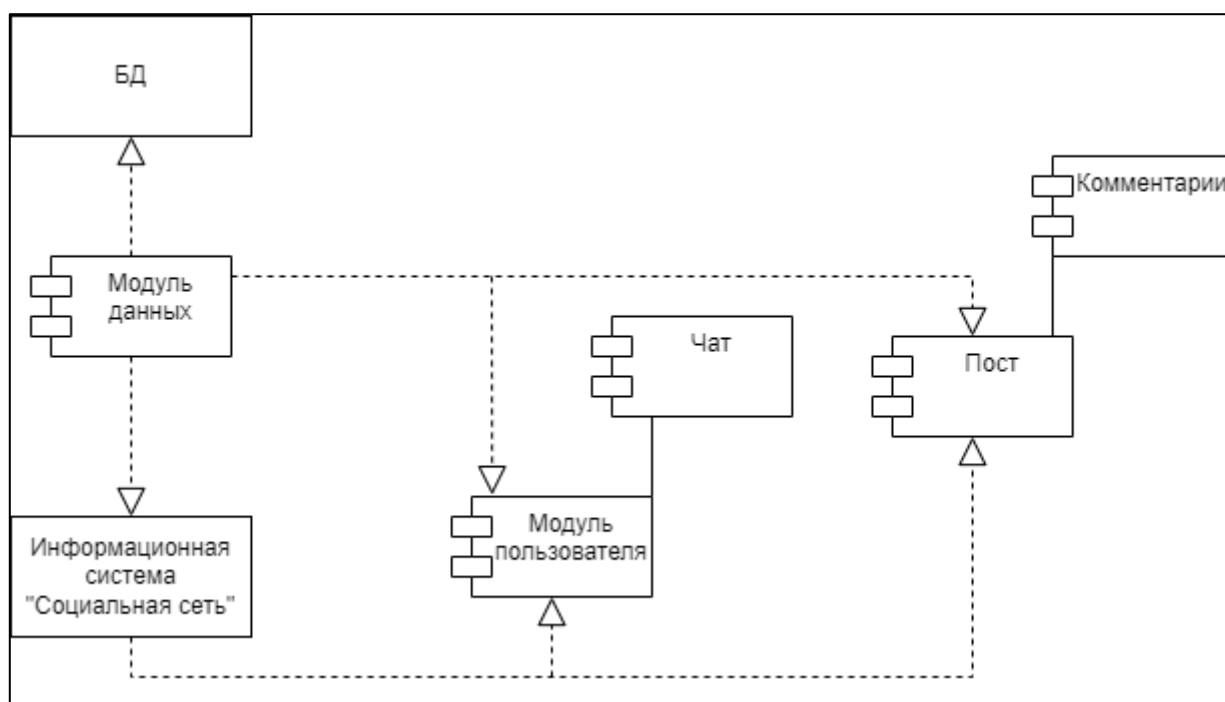


Рисунок 3 – Диаграмма компонентов

4.2 Функциональная схема веб-приложения

Структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними. На рисунке 4 представлена диаграмма классов.

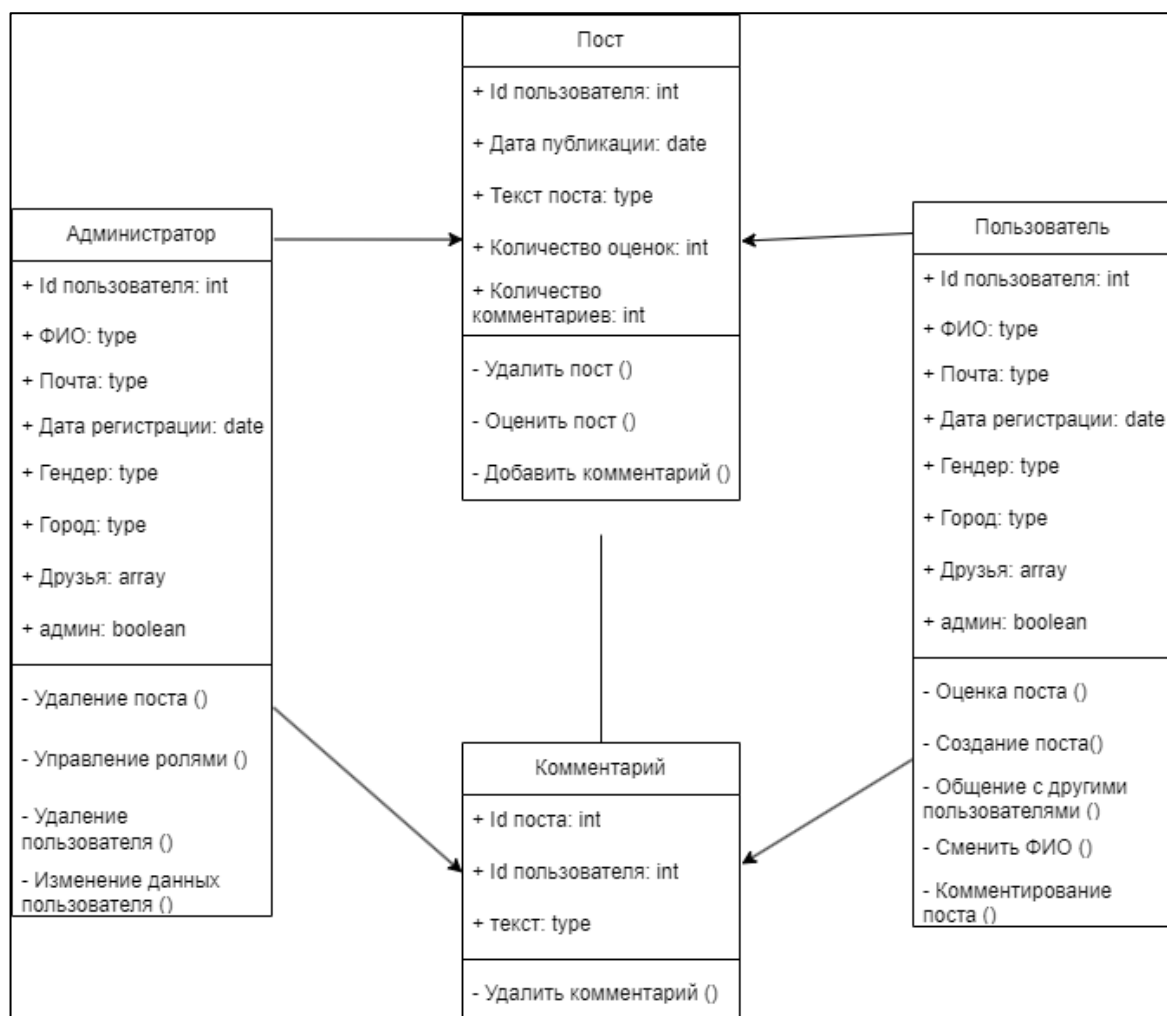


Рисунок 4 – Диаграмма классов

Диаграмма потоков данных (data flow diagram, DFD) – один из основных инструментов структурного анализа и проектирования информационных систем, существовавших до широкого распространения UML.

И так DFD – это нотация, предназначенная для моделирования информационный систем с точки зрения хранения, обработки и передачи данных, то есть используется разработчиками ИС для разработчиков ИС. На рисунке 5 представлена диаграмма потоков данных.

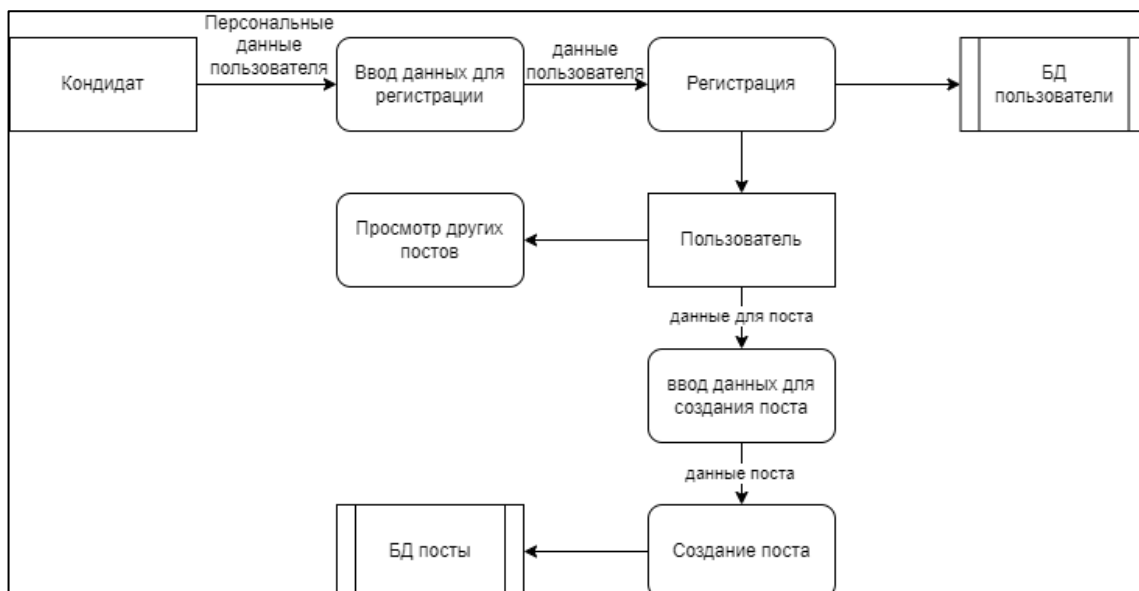


Рисунок 5 – Диаграмма потоков данных

Контекстная диаграмма в нотации IDEF0 (A0) – это графическое представление системы, которое показывает ее взаимодействие с внешней средой. Она состоит из одного блока, который представляет собой систему, и внешних блоков, которые представляют собой внешнюю среду. На рисунке 6 представлена контекстная диаграмма IDEF0 (A0).

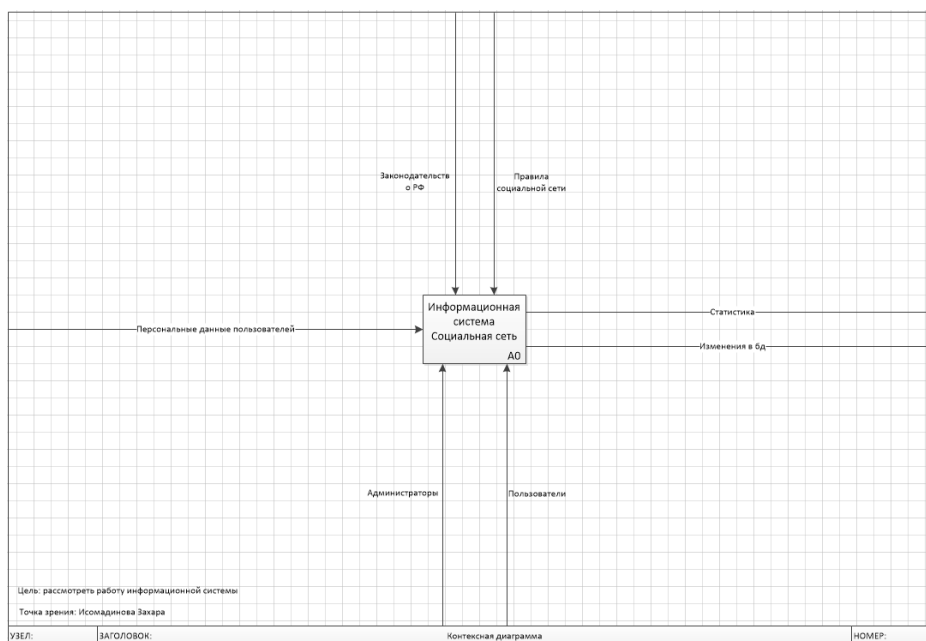


Рисунок 6 – Контекстная диаграмма IDEF0 (A0)

Контекстная диаграмма в нотации IDEF0 (A0) используется для понимания общей структуры системы и ее взаимодействия с внешней средой. Она не детализирует внутреннюю структуру системы и не показывает ее внутренние процессы. Ее цель - помочь понять контекст, в котором функционирует система, и определить ее границы.

Диаграмма декомпозиций (A1) в нотации IDEF0 представляет собой расширение контекстной диаграммы (A0) и используется для детализации внутренних процессов системы. Она позволяет разбить систему на более мелкие функциональные блоки и показать их взаимосвязь. На рисунке 7 представлена диаграмма декомпозиций (A1).

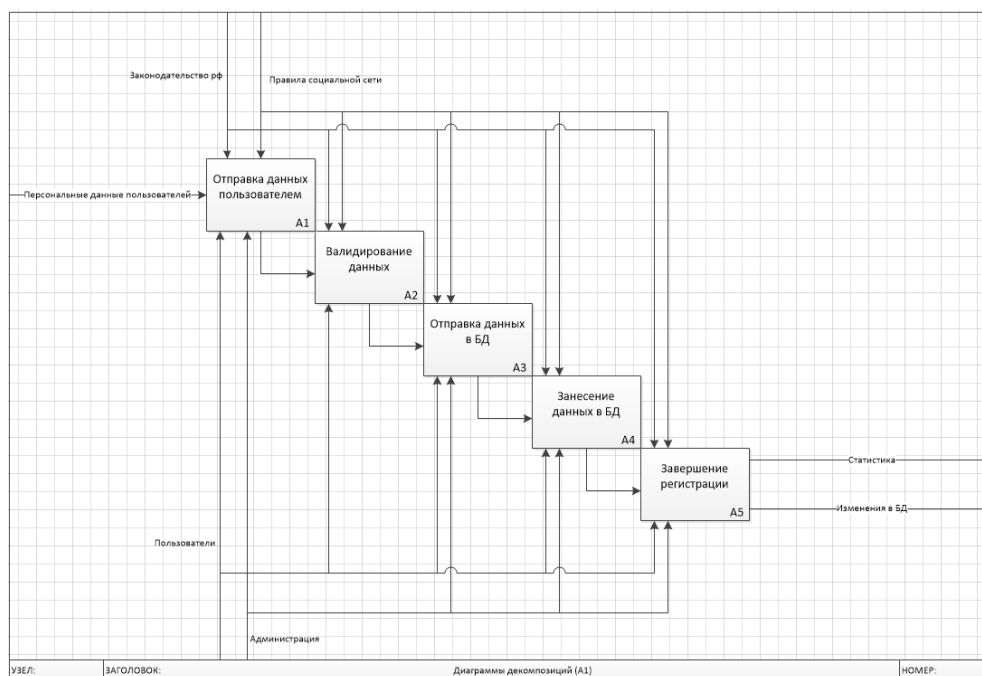


Рисунок 7 – Диаграммы декомпозиций (A1)

Диаграмма декомпозиций (A1) позволяет более подробно описать внутренние процессы системы и показать их взаимосвязь. Она помогает лучше понять, как система функционирует и какие процессы в ней происходят. Детализация на этой диаграмме может быть продолжена дальше с помощью диаграмм декомпозиций более низкого уровня, чтобы достичь более глубокого понимания системы и ее процессов.

Диаграмма декомпозиций (A2) в нотации IDEF0 является продолжением диаграммы декомпозиций (A1) и используется для дальнейшей детализации функциональных блоков системы. На рисунке 8 представлена диаграммы декомпозиций (A1).

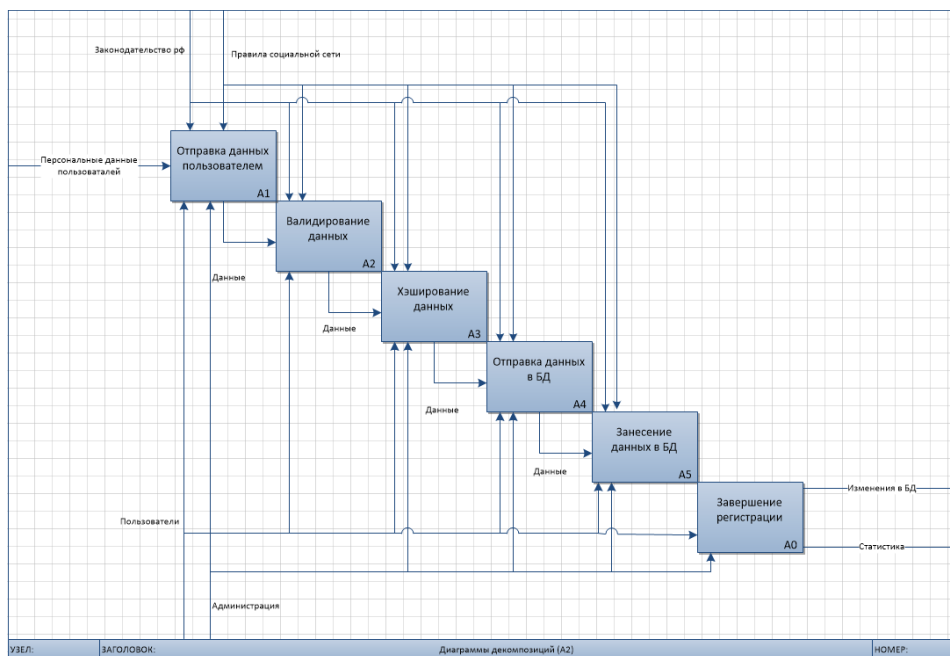


Рисунок 8 - Диаграммы декомпозиций (A2)

Диаграмма декомпозиций (A2) позволяет еще более детально описать внутренние процессы системы и показать их взаимосвязь на более низком уровне. Она помогает разбить функциональные блоки на более мелкие подпроцессы и показать, как они взаимодействуют друг с другом. Детализация на этой диаграмме может быть продолжена дальше с помощью диаграмм декомпозиций еще более низкого уровня, чтобы достичь максимальной детализации и полного понимания системы и ее процессов.

4.3 Проектирование базы данных

Прежде чем приступить разработке программного обеспечения необходимо спроектировать базу данных, а именно, определить с какими данными будут

работать участники системы, и чем данные связаны между собой. В этом заключается процесс проектирования.

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных рисунок 5.

Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

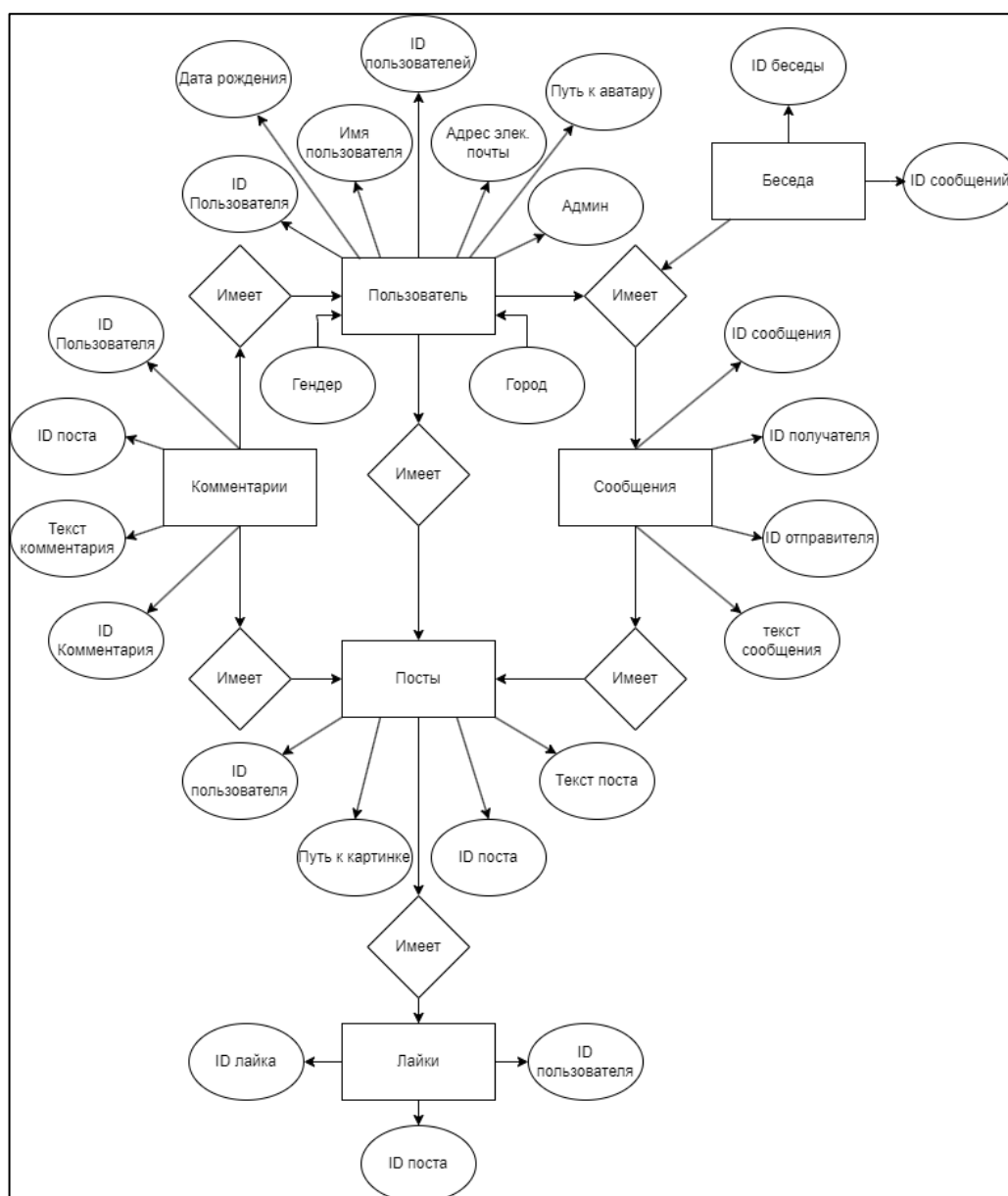


Рисунок 5 – Инфологическая модель

На модели видно четыре сущности, которые имеют свои уникальные атрибуты. Также сущности связаны между собой и в сумме получается структура под названием – инфологическая модель.

Проектирование базы данных не ограничивается только этим этапом разработки, а проходит на протяжении всей разработки, до того момента пока в системе не появятся данные, которые нельзя потерять.

На рисунке 6 представлена даталогическая модель логического уровня системы, представляющая собой отображение логических связей между элементами базы данных в среде конкретной СУБД.

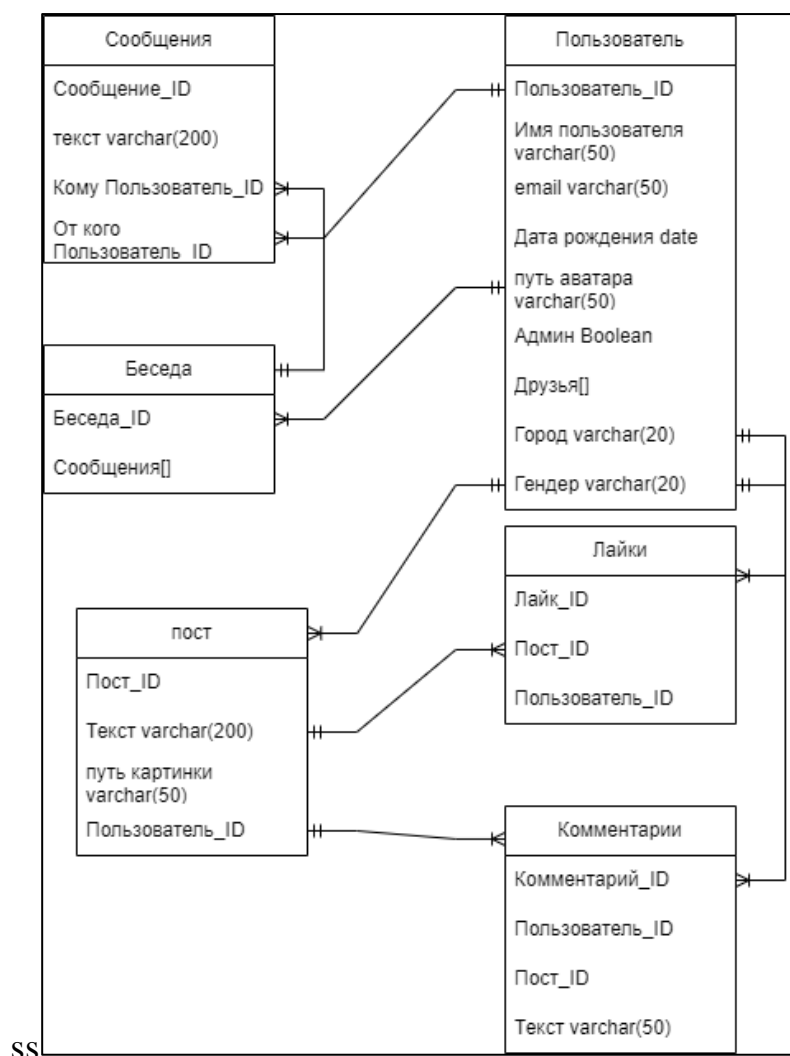


Рисунок 6 – Даталогическая модель

На рисунке 7 представлена ER-модель. Всего она содержит 6 таблиц:

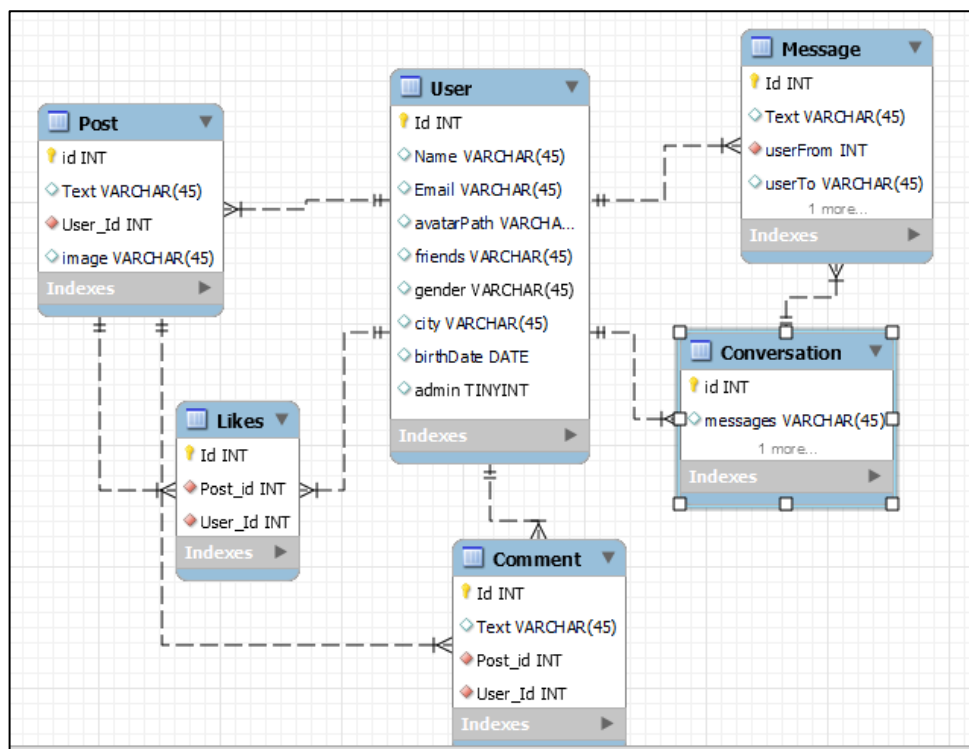


Рисунок 7 – ER-модель

1. Таблица «User», которая предназначена для хранения таких данных, как:
 - id (идентификатор);
 - Email (электронная почта пользователя);
 - Name (ФИО пользователя);
 - avatarPath (путь к аватару пользователя);
 - friends (Массив друзей пользователя);
 - gender (Гендер пользоваля);
 - city (Город пользователя);
 - birthdate (Дата рождения);
 - admin (пользователь является администратором или нет)
2. Таблица «likes», предназначена для хранения таких данных, как:
 - id (идентификатор);
 - user_id (идентификатор пользователя);
 - post_id (идентификатор поста, под которым оставили комментарий)

3. Таблица «Post», хранит такие данные, как:
 - id (идентификатор);
 - text (описание поста);
 - User_id (идентификатор пользователя);
 - image (путь к картинке).
4. Таблица «Comment», содержит в себе информацию о комментариях, такую как:
 - id (идентификатор);
 - Text (текст комментария);
 - Post_id (идентификатор поста);
 - User_id (идентификатор пользователя).
5. Таблица «Conversation», содержит в себе информацию о группах, такую как:
 - id (идентификатор права);
 - messages (массив сообщений);
6. Таблица «Message», содержит информацию о сообщениях, отправленных пользователя:
 - id (идентификатор сообщения);
 - text (текст сообщения);
 - userFrom (идентификатор пользователя отправителя);
 - userTo (идентификатор пользователя получателя).

Для разработки даталогической и ER-модели был использован инструмент MySQL Workbench.

После завершения разработки прототипа базы данных информационной системы получится готовая схема, по которой будет разрабатываться база данных.

4.4 Проектирование пользовательского интерфейса

Пользовательский интерфейс – это средство взаимодействия между человеком и компьютером. Говоря простыми словами, интерфейс – внешняя часть программы или устройства, с которыми работает пользователь.

Интерфейсы являются основой взаимодействия со всеми современными веб-приложениями. Если интерфейс какого-либо объекта не изменяется (стабилен, стандартизирован), это даёт возможность модифицировать сам объект, не перестраивая принципы его взаимодействия с другими объектами. Данный прототип интерфейса был построен в онлайн конструкторе Figma. На рисунке 8 представлена страница загрузки видео.

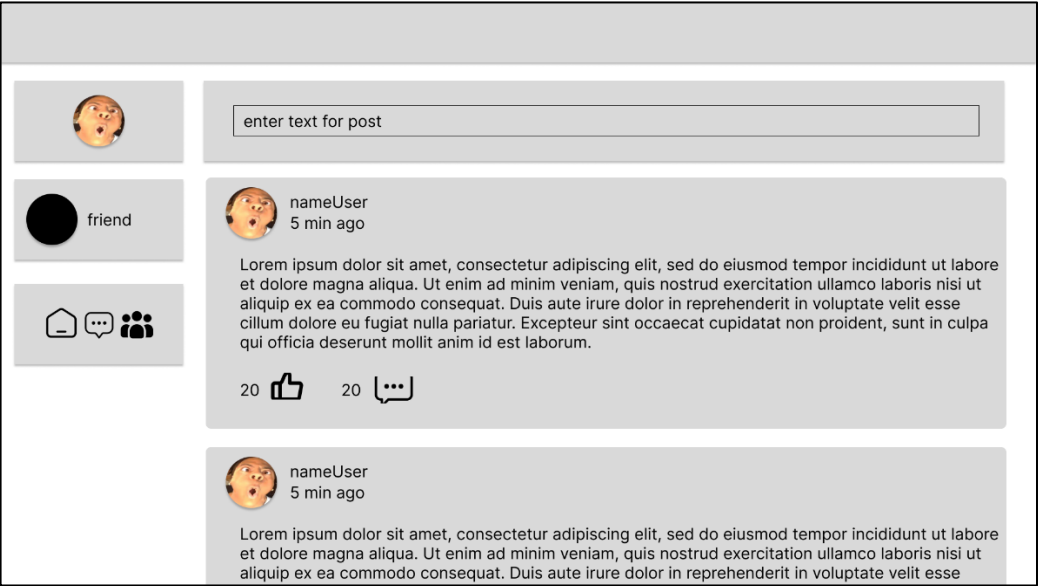


Рисунок 8 – Главная страница

Рисунок 9 содержит в себе страницу регистрации аккаунта пользователя.

Sign up

Фамилия

Электронная почта

Пароль

Имя

Логин

Повтор пароля

Отчество

Дата рождения

Submit

Already have a account?

Рисунок 9 – Страница регистрации

Рисунок 10 содержит в себе страницу авторизации пользователя, также можно перейти к регистрации в случае отсутствия аккаунта.

Sign in

Электронная почта

Пароль

Submit

Dont have a account?

Рисунок 10 – Страница авторизации

Во время разработки прототипа интерфейса был учтен проведенный анализ аналогов, что помогло реализовать представленный вид прототипа информационной системы «Социальная сеть».

5. Разработка информационной системы

5.1 Разработка интерфейса информационной системы

Разработка удобного пользовательского интерфейса – это один из важнейших этапов в процессе создания информационной системы. Для разработки интерфейса информационной системы использовался формат TS(JavaScript). Это позволяет создавать динамические элементы, обрабатывать события пользователя и взаимодействовать с сервером.

На рисунке 11 представлена панель навигации.

```
import React, { FC } from 'react'

import Menu from './Menu'
import User from './User'
import UserItems from './UserItems/UserItems'

const Sidebar: FC = () => {
  return (
    <div>
      <User />
      <UserItems />
      <Menu />
    </div>
  )
}

export default Sidebar
```

Рисунок 11 – Панель навигации

На рисунке 12 представлен код главной страницы.

```

import { FC } from 'react'
import { useQuery } from 'react-query'

import Layout from '@components/layout/Layout'
import AddPost from '@components/ui/posts/AddPost'
import Posts from '@components/ui/posts/Posts'

import { PostService } from '@services/post.service'

const Home: FC = () => {
  const { data, isLoading, refetch } = useQuery(
    'get all posts',
    () => PostService.getAll(),
    {
      select: ({ data }) => data
    }
  )

  return (
    <Layout title='Главная'>
      <div>
        <AddPost refetch={refetch} />
        <Posts
          posts={data || []}
          isLoading={isLoading}
          refetchPosts={refetch}
        />
      </div>
    </Layout>
  )
}

export default Home

```

Рисунок 12 – Главная страница

5.2 Разработка базы данных информационной системы

Разработка базы данных выполнена с использованием СУБД MongoDB. База данных состоит из семи таблиц, которые отражают структуру. Каждая таблица представляет определенные сущности и их атрибуты, чтобы эффективно хранить и организовывать данные.

Рисунок 13 отображает таблицу пользователей, которая содержит информацию о зарегистрированных пользователях, их идентификаторах, связанных с профилем пользователя.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

```

_id: ObjectId('655a13eeb86edae63038b111')
email: "zaharisomadinov7@gmail.com"
name: "Maddox"
isVerified: false
avatarPath: "https://lh3.googleusercontent.com/a/ACg8ocITm0nGuEMznDPANYW7V3aQaBpSSL..."
isAdmin: true
friends: Array (1)
createdAt: 2023-11-19T13:55:58.782+00:00
updatedAt: 2023-11-20T15:54:33.768+00:00
__v: 0

```

Рисунок 13 – Таблица «users»

Рисунок 14 представляет таблицу поста, которая содержит информацию о созданных постах пользователями.

```

_id: ObjectId('655ee4fd1b5ed5f573f06b0b')
content: "asd"
user: ObjectId('655cb75be4c011e8efad7dbf')
createdAt: 2023-11-23T05:37:01.764+00:00
updatedAt: 2023-11-23T05:37:01.764+00:00
__v: 0

```

Рисунок 14 – Таблица «post»

Рисунок 15 отображает таблицу сообщения, которая содержит информацию о отправленных пользователями сообщениях, их идентификаторах, а также кому и от кого отправлено сообщение.

```

_id: ObjectId('655ee5471b5ed5f573f06b57')
text: "asdfsdf"
userFrom: ObjectId('655cb75be4c011e8efad7dbf')
userTo: ObjectId('655a147cb86edae63038b120')
createdAt: 2023-11-23T05:38:15.650+00:00
updatedAt: 2023-11-23T05:38:15.650+00:00
__v: 0

```

Рисунок 15 – Таблица «message»

Рисунок 16 показывает таблицу лайка, которая содержит информацию о поставленных лайках пользователями.

```
_id: ObjectId('655ee5081b5ed5f573f06b1f')
post: ObjectId('655ee5061b5ed5f573f06b15')
user: ObjectId('655cb75be4c011e8efad7dbf')
createdAt: 2023-11-23T05:37:12.821+00:00
updatedAt: 2023-11-23T05:37:12.821+00:00
__v: 0
```

Рисунок 16 – Таблица «likes»

Рисунок 17 представляет таблицу беседы, которая содержит информацию о сообщениях в беседе пользователей.

```
_id: ObjectId('655a14b0b86edae63038b12d')
messages: Array (empty)
createdAt: 2023-11-19T13:59:12.508+00:00
updatedAt: 2023-11-19T13:59:12.508+00:00
__v: 0
```

Рисунок 17 – Таблица «conversation»

Рисунок 18 описывает таблицу комментария, которая содержит информацию о оставленных пользователями комментариями.

```
_id: ObjectId('655ee50d1b5ed5f573f06b2c')
user: ObjectId('655cb75be4c011e8efad7dbf')
post: ObjectId('655ee5061b5ed5f573f06b15')
message: "Cool!"
createdAt: 2023-11-23T05:37:17.704+00:00
updatedAt: 2023-11-23T05:37:17.704+00:00
__v: 0
```

Рисунок 18 – Таблица «comment»

5.3 Разработка информационной системы

В соответствии с заданием, в данной информационной системе необходимо

реализовать следующие функции:

1. Распределение уровней доступа.
2. Чтение данных.
3. Хранение и добавление данных.
4. Удаление и изменение данных.

Разработка функционала информационной системы было реализовано на TS(typeScript). Сначала идёт подключение к базе данных, показано на рисунке 19.

```
TypegooseModule.forRootAsync({  
  imports: [ConfigModule],  
  inject: [ConfigService],  
  useFactory: getMongoConfig  
}),
```

Рисунок 19 – Подключение к базе данных

5.4 Тестирование информационной системы

Одним из этапов разработки любого программного продукта является тестирование и отладка.

Тестирование – это процесс, который используется для проверки правильности работы программы и обнаружения ошибок в ней. Основная цель тестирования программного продукта заключается в проверке его работоспособности, надежности и безопасности.

Отладка – это процесс обнаружения, локализации и устранения ошибок в программе.

Отладка проводилась в процессе написания программного кода, путем запуска небольших фрагментов кода и проверки их работоспособности. В случае обнаружения ошибок, корректировался код и снова проводилось тестирование.

Этот процесс повторялся до тех пор, пока весь программный код не стал работать без ошибок.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

Данный программный продукт тестировался методом “белого ящика”. Это означает, что был доступ к исходному коду программы и могли проверить каждую строку кода, чтобы убедиться в его правильности.

Модульное тестирование – это метод тестирования, при котором тестируется отдельный модуль или класс программы. Цель модульного тестирования – убедиться, что каждый модуль работает правильно и взаимодействует с другими модулями без ошибок.

На рисунке 20 представлен код файла media.service.spec.ts, в котором тестируется проверка нахождения файла по указанному пути. Данный тест выполнен безошибок.

```
describe('MediaService', () => {
  let mediaService: MediaService;

  beforeEach(async () => {
    const module: TestingModule = await Test.createTestingModule({
      providers: [MediaService],
    }).compile();

    mediaService = module.get<MediaService>(MediaService);
  });

  afterEach(async () => {
    // Удаление файла после каждого теста
    const uploadFolder = './uploads/default';
    const filePath = `${uploadFolder}/image.jpg`;
    if (await fs.pathExists(filePath)) {
      await fs.unlink(filePath);
    }
  });

  it('should save media file and verify its existence', async () => {
    const mediaFile: Express.Multer.File = {
      fieldname: 'file',
      originalname: 'image.jpg',
      encoding: '7bit',
      mimetype: 'image/jpeg',
      buffer: Buffer.from('file-content'),
      size: 123,
      destination: '',
      filename: '',
      path: '',
      stream: null,
    };

    const result = await mediaService.saveMedia(mediaFile);

    // Проверка возвращаемых данных
    expect(result).toEqual({
      url: '/uploads/default/image.jpg',
      name: 'image.jpg',
    });

    // Проверка наличия файла в указанном пути
    const fileExists = await fs.pathExists(`${uploadFolder}/image.jpg`);
    expect(fileExists).toBeTruthy();
  });
});
```

Рисунок 20 – Тестирование нахождения файла

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						33
Изм.	Лист	№ докум.	Подпись	Дата		

На рисунке 21 представлен результат тестирования.

```
PASS src/file/media.service.spec.ts
MediaService
  ✓ should save media file and verify its existence (11 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.39 s, estimated 2 s
```

Рисунок 21 – Результат тестирования

Для тестирования ИС был разработан сценарий тестирования. В таблице 4 представлен сценарий для роли пользователь.

Таблица 4 – Сценарий тестирования

Поле	Описание
Дата теста	20.11.2023
Приоритет тестирования	Средний
Название теста	Отправка сообщения другому пользователю
Этапы теста	1. Авторизация 2. Выбор пользователя 3. Отправка сообщения
Тестовые данные	Данные пользователя для авторизации
Ожидаемый результат	Система валидирует, заносит в базу данных и показывает на сайте сообщение отправленное другому пользователю.
Фактический результат	Система показала отправленное сообщение.

Кроме того, был создан чек-лист для роли пользователь, представленный в таблице 5

Таблица 5 – Чек-лист

Тест	Входные данные	Ожидаемый результат	Фактический результат	Результат тестирования	комментарий
Отправка сообщения	Данные пользователя: текст сообщения	Пользователь успешно отправил сообщение	Пользователь успешно отправил сообщение	Успешно	-
Создание поста	Данные пользователя: текст для поста	Пользователь успешно создал пост	Пользователь успешно создал пост	успешно	-
Авторизация	Данные пользователя: пароль и адрес электронной почты от Google	Пользователь успешно авторизовался	Пользователь успешно авторизовался	Успешно	-

6 Документирование программного продукта

6.1 Руководство пользователя ИС

Для запуска информационной системы необходимо установить последнюю версию npm и node. Затем открыть папку с проектом в любой доступной среде разработки.

Далее открыть два терминала и в каждом написать «npm install» для установки всех зависимостей. В один терминал ввести команду «cd .\server\» и «npm start», а во второй «cd .\client \» и «npm run dev». После запуска перейти в браузере по ссылке «http://localhost:3000/auth».

После запуска информационной системы откроется страница авторизации через сервис google. На рисунке 22 представлена страница авторизации.

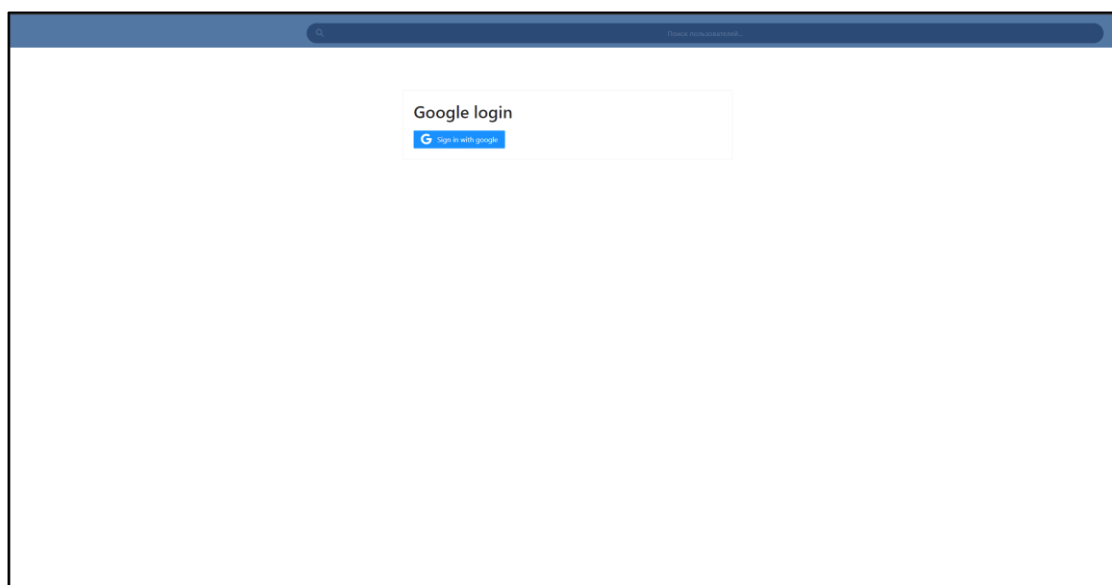


Рисунок 22 – Страница авторизации

После авторизации откроется главная страница с имеющимися постами от пользователей, список друзей, возможностью создать, оценить или комментировать пост.

Чтобы создать пост нужно на главной странице нажать на поле, где написано «Расскажи, что у тебя нового», ввести текст, выбрать картинку при

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						36
Изм.	Лист	№ докум.	Подпись	Дата		

необходимости и нажать enter на клавиатуре.

Чтобы посмотреть список друзей нужно на навигационной панели слева нажать на «Друзья». На рисунке 23 представлена страница друзей.

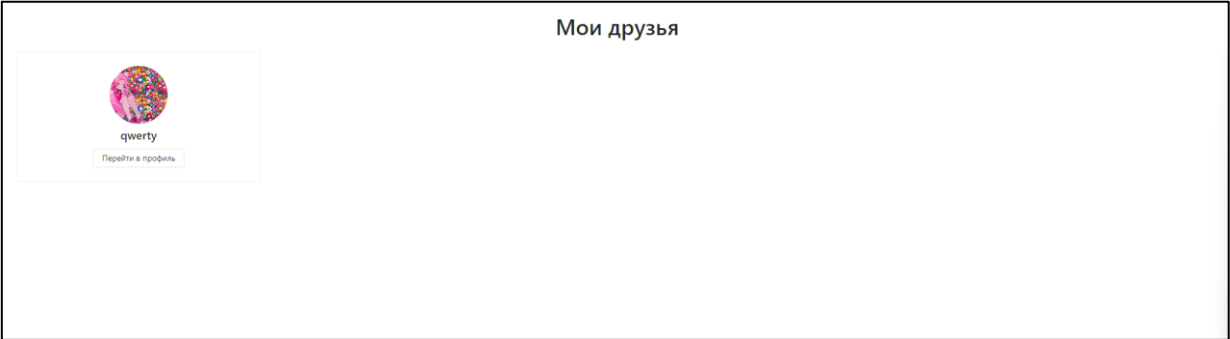


Рисунок 23 – Страница друзей

Чтобы посмотреть профиль друга нужно нажать на кнопку «Перейти в профиль». Откроется профиль друга, представленный на рисунке 24.

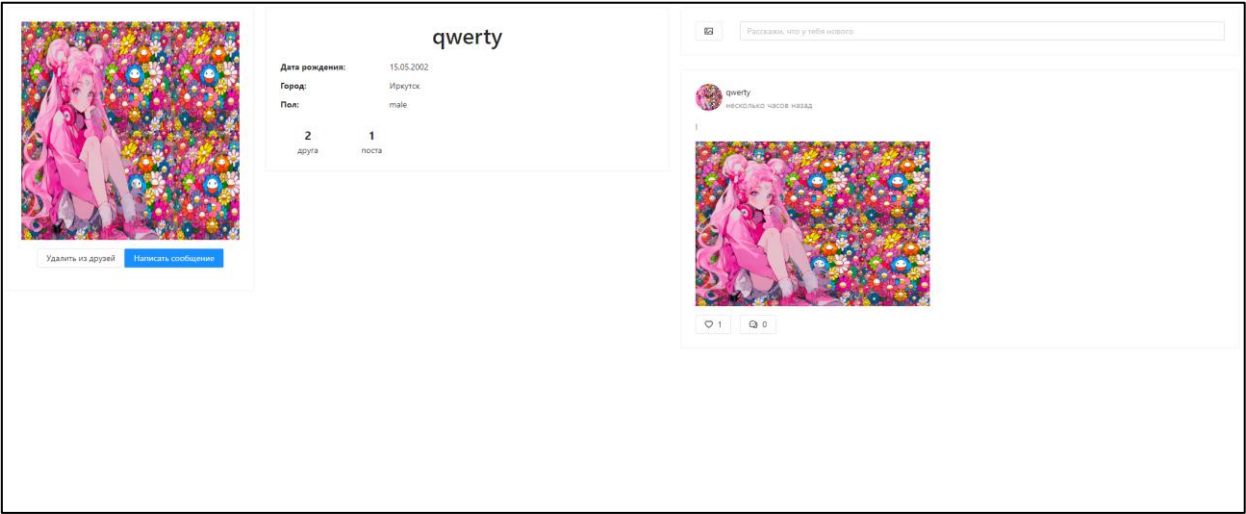


Рисунок 24 – Страница друга

Если у вас отсутствуют друзья, то можно на главной странице нажать на поле «Поиск пользователей» и найти себе друга по имени. На рисунке 25 представлена главная страница с результатами поиска пользователей.



Рисунок 25 – Главная страница с результатами поиска

Для добавления или удаления пользователя из списка друзей нужно после нажатия на кнопку «Перейти в профиль» нажать соответствующую кнопку. На рисунках 26 и 27 показана кнопка для добавления и удаления пользователя из списка друзей.

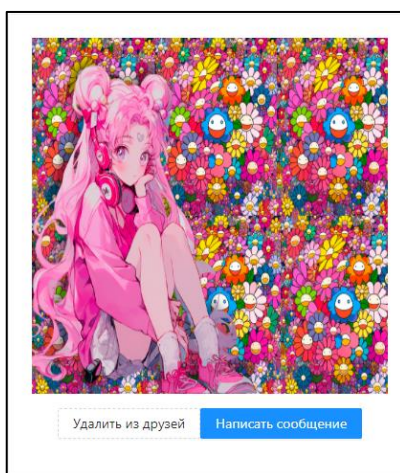


Рисунок 26 – кнопка для удаления пользователя из списка друзей

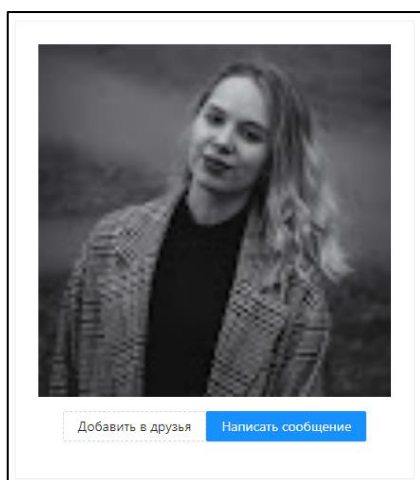


Рисунок 27 – кнопка добавления пользователя в список друзей

Для того чтобы написать сообщение пользователю нужно перейти в профиль пользователя, которому нужно написать сообщение и нажать на кнопку «Написать сообщение». Кнопки и страницы чата с другим пользователем представлена на рисунках 28 и 29 соответственно.

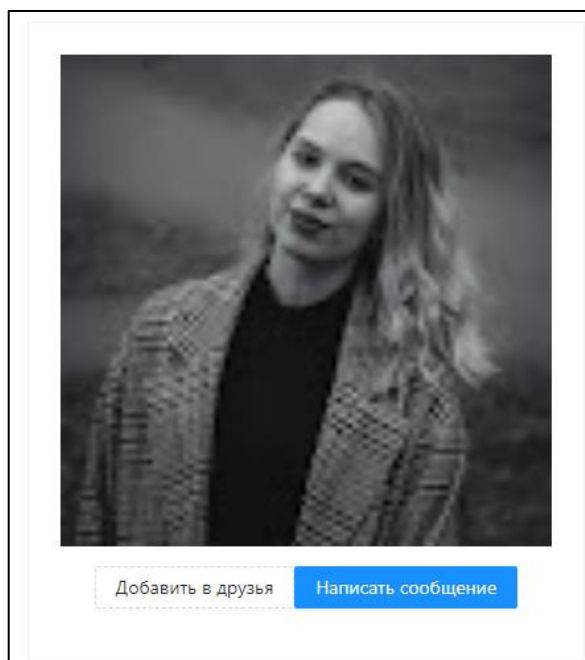


Рисунок 28 – кнопка «написать сообщение»

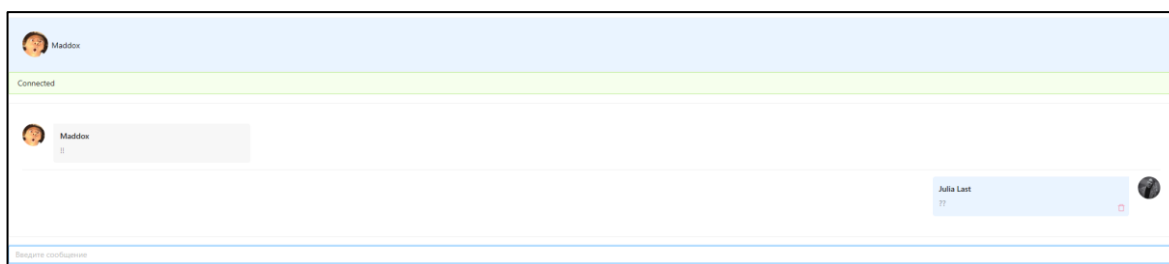


Рисунок 29 – страница чата с другим пользователем

Заключение

В ходе разработки проекта была поставлена цель разработка информационной системы «Социальная сеть». Для данной цели были определены ряд задач.

Для разработки информационной системы были определены следующие технологии: HTML5, CSS3, JS, NextJs, NestJs.

Цель и задачи выполнены, и программный продукт был создан в соответствии с техническим заданием. В процессе создания курсового проекта, были получены и применены новые знания о технологиях и инструментах разработки. Был создан программный продукт, который позволяет общаться с пользователями и делиться впечатлениями через посты.

В результате курсового проектирования была выполнена главная цель разработка информационной системы «Социальная сеть». Были проанализированы инструменты и выполнены поставленные задачи для достижения цели:

- Провести предпроектное исследование.
- Провести обзор инструментальных средств.
- Разработать техническое задание.
- Спроектировать информационную систему.
- Спроектировать пользовательский интерфейс.
- Обеспечить безопасность и конфиденциальность данных пользователей.
- Разработать информационную систему.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

Список используемых источников

1 bobbyhadz.com – Making HTTP requests with Axios in TypeScript – URL: <https://bobbyhadz.com/blog/typescript-http-request-axios> (Дата обращения: 20.10.2023). – Текст электронный.

2 Dev.to – How to Implement Login with Google in Nest JS – URL: <https://dev.to/imichaelowolabi/how-to-implement-login-with-google-in-nest-js-2aoa> (Дата обращения: 17.09.2023). – Текст электронный.

3 Nestjs.ru – Документация к фреймворку NestJj. – URL: <https://nestjs.ru/> (дата обращения: 10.09.2023). – Текст: электронный.

4 Nestjs.ru – Документация о тестировании на nestjs с использованием jest– URL: <https://nestjs.ru/fundamentals/testing> (Дата обращения: 18.11.2023). – Текст электронный.

5 Nestjs.ru – Использование шлюзов в Nestjs – URL: <https://docs.nestjs.com/websockets/gateways> (Дата обращения: 15.09.2023). – Текст электронный.

6 nextjs.jrg – Документация к фреймворку NextJs – URL: <https://nextjs.org/> (дата обращения: 08.10.2023). – Текст: электронный.

7 Npmjs.com – A simple, lightweight JavaScript API for handling cookies – URL: <https://bobbyhadz.com/blog/typescript-http-request-axios> (Дата обращения: 20.10.2023). – Текст электронный.

8 Npmjs.com – React-query is often described as the missing data-fetching library for web applications– URL: <https://www.npmjs.com/package/react-query> (Дата обращения: 20.10.2023). – Текст электронный.

9 Passportjs.org – A Passport strategy for authenticating with a JSON Web Token – URL: <https://www.passportjs.org/packages/passport-jwt/>. (дата обращения: 17.09.2023). – Текст электронный.

10 typegoose.github.io – Define Mongoose models – URL: <https://typegoose.github.io/> (Дата обращения: 20.09.2023). – Текст электронный.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						41
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А – Техническое задание

Министерство образования Иркутской области

Государственное бюджетное профессиональное образовательное учреждение
Иркутской области
«Иркутский авиационный техникум»
(ГБПОУИО «ИАТ»)

ТЕХНИЧЕСКОЕ ЗАДАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «СОЦИАЛЬНАЯ СЕТЬ»

Руководитель: _____ (А.С. Александрова)
(подпись, дата)

Студент: _____ (З.З. Исомадинов)
(подпись, дата)

Иркутск 2023

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						42
Изм.	Лист	№ докум.	Подпись	Дата		

1 Общие сведения

Наименование работы: информационная система «Социальная сеть».

Исполнитель: студент иркутского авиационного техникума, группы ВЕБ-20-1, 3.3. Исомадинов

Разработка информационной системы проходит в рамках курсового проекта по МДК.05.02 «Разработка кода информационных систем», на основании приказа №39-у от 14 сентября 2023 года.

Сроки разработки Информационной системы с 17.09.2023 по 17.11.2023 года.

2 Назначение и цели создания системы

Назначение информационной системы «Социальная сеть» заключается в создании комфортных условий для общения пользователей. Пользователь сможет оценивать и комментировать посты, делиться своими.

3 Требования к системе в целом

3.1 Требования к структуре и функционированию системы

Функции Информационной системы:

1. раздел «Главная»:

1.1. Просмотр всех постов от других пользователей.

1.2. Оценивание постов.

1.3. Комментирование.

1.4. Создание поста

2. раздел «Профиль другого пользователя»:

2.1. Просмотр постов, опубликованных пользователем.

2.2. Комментирование постов.

2.3. Оценивание постов.

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

3. Раздел «Моя страница»:

3.1. Изменить информацию профиля.

3.2. Опубликовать пост.

4. Раздел «Авторизация»:

4.1. Авторизация через сервис Google.

3.2 Требования к надежности

Для обеспечения надежности необходимо проверять корректность получаемых данных и реализовать валидность полей. Входные данные поступают в виде значений с клавиатуры. Эти значения отображаются в отдельных полях таблицы.

3.3 Требования к безопасности

Для обеспечения безопасности в информационной системе, необходимо реализовать разграничение прав доступа.

3.4 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Минимальные системные требования для сервера:

1. Операционная система: Windows 7/ 8/ 10.
2. Процессор: 2 ГГц.
3. Видеоадаптер (GPU): nVidia GF 6800, ATI HD2400XT.
4. Видеопамять (VRAM): 256 МБ.
5. Винчестер (SSD): 1000 МБ.
6. Оперативная память (RAM): 4 ГБ.
7. Версия MySQL 5.0 и выше.

Минимальные системные требования для рабочей станции:

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						44
Изм.	Лист	№ докум.	Подпись	Дата		

1. Операционная система: Windows XP/ 7/ 8/ 10.
2. Процессор: 1,8 ГГц.
3. Видеоадаптер (GPU): nVidia GF 6800, ATI HD2400XT.
4. Видеопамять (VRAM): 64 Mb.
5. Винчестер (SSD): 800 МБ.
6. Оперативная память (RAM): 2 ГБ.

4 Требования к документированию

Основным документам, регламентирующими использование информационной системы является руководство пользователя.

Основным документам, регламентирующими разработку информационной системы является техническое задание.

5 Состав и содержание работ по созданию системы

В таблице 1 представлены плановые сроки начала и окончания работы по разработке информационной системы.

Таблица 1 – Плановые сроки по Информационной системе

№ п\п	Этапы разработки	Сроки выполнения	
		Начало	Окончание
1	Предпроектное исследование предметной области (выбор темы, постановка цели, задач, описание области применения, исследование предметной области)	15.09.23	21.09.23
2	Разработка технического задания (выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор языка и среды программирования)	21.09.23	26.09.23
3	Проектирование программного обеспечения. (разработка структурной и функциональной схемы ПО, проектирование базы данных (инфологическое, ER-модель, физическая модель)	26.09.23	10.10.23
4	Разработка (программирование) программного продукта	10.10.23	10.11.23

Продолжение таблицы 1.

5	Тестирование и отладка программного продукта (функциональное тестирование, составление тест-планов)	10.11.23	15.11.23
6	Составление программной документации (оформление ПЗ, написание руководства пользователя, составление презентации и речи)	15.11.23	17.11.23

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						46
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение Б – Листинг

```
import { FC } from 'react'
import { useQuery } from 'react-query'

import Layout from '@components/layout/Layout'
import AddPost from '@components/ui/posts/AddPost'
import Posts from '@components/ui/posts/Posts'

import { PostService } from '@services/post.service'

const Home: FC = () => {
  const { data, isLoading, refetch } = useQuery(
    'get all posts',
    () => PostService.getAll(),
    {
      select: ({ data }) => data
    }
  )

  return (
    <Layout title='Главная'>
      <div>
        <AddPost refetch={refetch} />
        <Posts
          posts={data || []}
          isLoading={isLoading}
          refetchPosts={refetch}
        />
      </div>
    </Layout>
  )
}

export default Home
```

					КП.09.01.07-3.23.201.09 ПЗ	Лист
						47
Изм.	Лист	№ докум.	Подпись	Дата		