МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное   
учреждение высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

**Разработка веб-приложения «NetMate»**

КУРСОВАЯ РАБОТА

по дисциплине «Программная инженерия»

ЮУрГУ – 02.03.02.2024.308-007.КР

|  |  |
| --- | --- |
| Нормоконтролер,ст. преподаватель кафедры СП  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.С. Силкина  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г. | Научный руководитель:  ст. преподаватель кафедры СП  \_\_\_\_\_\_\_\_\_\_ Н.С. Силкина  Автор работы:  студент группы КЭ-301  \_\_\_\_\_\_\_\_\_\_ Д.А. Иванов  Работа защищена  с оценкой: \_\_\_\_\_\_\_\_\_\_\_  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

Челябинск 2024МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_\_Л.Б. Соколинский

10.02.2024

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине «Программная инженерия»

студенту группы КЭ-301 Иванову Дмитрию Александровичу,

обучающемуся по направлению 02.03.02 «Фундаментальная информатика и информационные технологии»

1. **Тема работы**Разработка веб-приложения «NetMate»
2. **Срок сдачи студентом законченной работы:** 31.05.2024 г.
3. **Исходные данные к работе**
4. React – A JavaScript library for building user interfaces. [Электронный ресурс] URL: <https://legacy.reactjs.org> (дата обращения 19.02.2024 г.)
5. React Documentations. [Электронный ресурс] URL: <https://react.dev> (дата обращения 19.02.2024 г.)
6. Руководство по CSS. [Электронный ресурс] URL: https://developer.mozilla.org/ru/docs/Web/CSS/Reference (дата обращения: 19.02.2024 г.).
7. Руководство JavaScript. [Электронный ресурс] URL: https://developer.mozilla.org/ru/docs/Web/JavaScript (дата обращения: 19.02.2024 г.).
8. **Перечень подлежащих разработке вопросов**
9. Провести анализ аналогичных приложений;
10. Спроектировать веб-приложение;
11. Реализовать прототип веб-приложения;
12. Провести тестирование реализованного прототипа.
13. **Дата выдачи задания:** 9 февраля 2024 г.

Научный руководитель

ст. преподаватель кафедры СП, Н.С. Силкина

Задание принял к исполнению Д.А. Иванов

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 5](#_Toc168931312)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc168931313)

[1.1. Обзор аналогичных проектов 6](#_Toc168931314)

[1.2. Анализ существующих решений для реализации проекта 9](#_Toc168931315)

[2. ТРЕБОВАНИЯ К СИСТЕМЕ 12](#_Toc168931316)

[2.1. Функциональные требования 12](#_Toc168931317)

[2.2. Нефункциональные требования 12](#_Toc168931318)

[3. ПРОЕКТИРОВАНИЕ 14](#_Toc168931319)

[3.1. Диаграмма вариантов использования 14](#_Toc168931320)

[3.2. Проектирование интерфейса 15](#_Toc168931321)

[3.3. Диаграмма компонентов 15](#_Toc168931322)

[4. РЕАЛИЗАЦИЯ 16](#_Toc168931323)

[5. ТЕСИРОВАНИЕ 17](#_Toc168931324)

[ЗАКЛЮЧЕНИЕ 18](#_Toc168931325)

[ЛИТЕРАТУРА 19](#_Toc168931326)

ВВЕДЕНИЕ

**Актуальность**

В современном мире интернет играет ключевую роль в повседневной жизни людей, обеспечивая доступ к информации, развлечениям и социальным взаимодействиям. Одной из наиболее быстрорастущих сфер является индустрия видеотрансляций. Подобные платформы, например Twitch, привлекают миллионы пользователей, предоставляя им возможность как смотреть, так и создавать собственный контент.

Актуальность разработки нового веб-приложения для трансляции видео и просмотра трансляций обусловлена ростом популярности стриминга, широкими возможностями для монетизации контента, большим охватом аудитории за счёт разнообразия контента. [25]

**Цель и задачи работы**

Целью проекта является разработка веб-приложения для потоковой передачи видео и общения пользователей.

Для реализации поставленной цели были определены следующие задачи:

1. Провести анализ предметной области;
2. Разработать требования к веб-положению;
3. Спроектировать веб-приложение;
4. Реализовать приложение;
5. Провести тестирование веб-приложения.

**Структура и содержание работы**

Содержание

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ
   1. Обзор аналогичных проектов

В течение последнего десятилетия всё популярнее становятся сервисы для публикации и просмотра медиаконтента. В том числе и для потокового вещания или стриминга. Наиболее часто стриминг предполагает прямое общение, трансляции видеоигр и спортивных состязаний. Именно его стали называть «медиа будущего». [12]

**Twitch**

Twitch – видеостриминговый сервис, специализирующийся на тематике компьютерных игр. Но, помимо этого, Twitch проводит трансляции и видео на другие тематик, например, музыкальные выступления, искусство, турниры по различным дисциплинам. Видео на платформе Twitch можно просматривать как в реальном времени, так и по запросу. Twitch принадлежит Twitch Interactive – дочерней компании Amazon. Сервис был создан в 2011 году. [8]

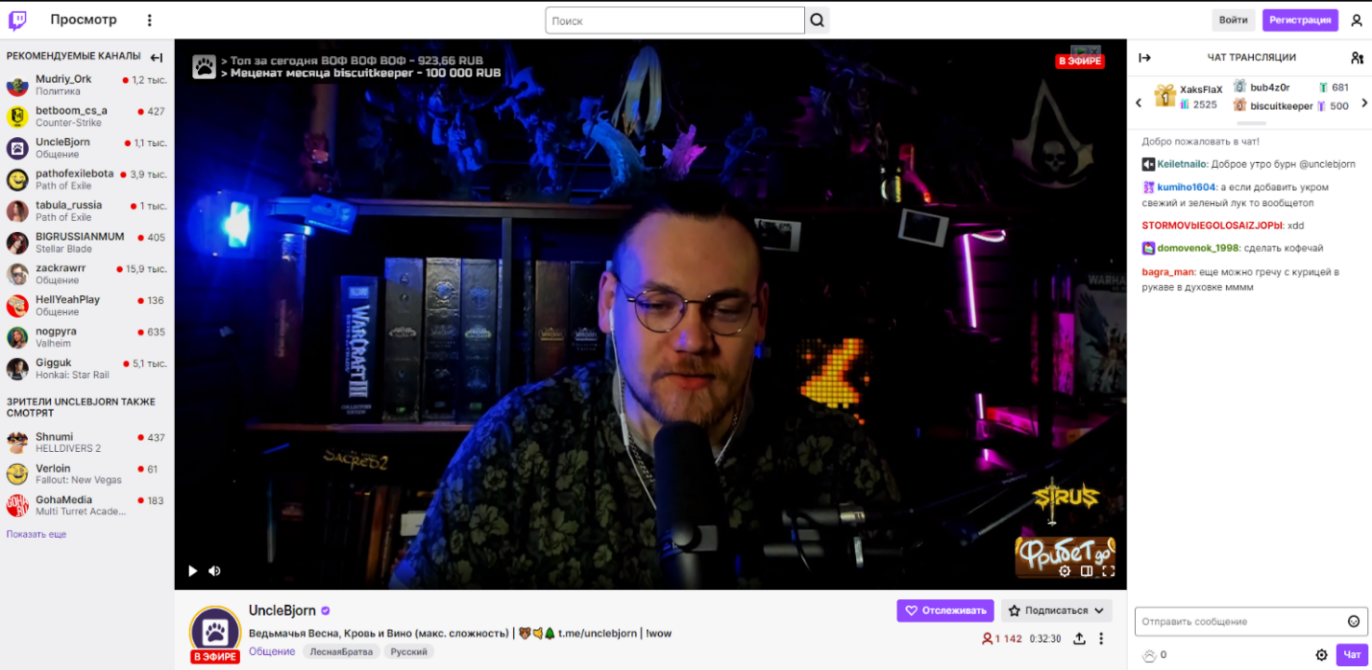
****

Рисунок 1 – скриншот Twitch

Twitch предлагает множество инструментов для взаимодействия между стримером и аудиторией. Это групповой чат, денежные поощрения, подписки, опросы, рейды и многое другое. Его легко начать использовать как стримеру, так и зрителю. Twitch первым предложил систему подписок, которая позволяет зрителям материально поддерживать своих любимых стримеров, при этом получая уникальные возможности для общения.

Twitch ведёт модерацию контента, благодаря чему на платформе нет вредного и опасного для общества контента. Однако в последнее время люди начали находить лазейки для трансляции откровенного контента, при этом модерация платформы ничего с этим не делает. [9]

Также стоит отметить, что Twitch даёт возможность составить расписание трансляций и отправляет уведомления о начале стрима. Ещё Twtich позволяет зрителям делать клипы – короткие вырезки интересных или смешных моментов со стрима.

Из основных минусов можно отметить большую конкуренцию и несправедливую, по мнению большинства зрителей и стримеров модерацию контента на платформе.

**VK Play Live**

VK Play Live – стриминговая платформа, принадлежащая российская инвестиционная технологическая корпорация VK. Была создана в 2022 году, входит в состав крупного российского игрового сервиса VK Play. Создавалась как аналог Twitch.

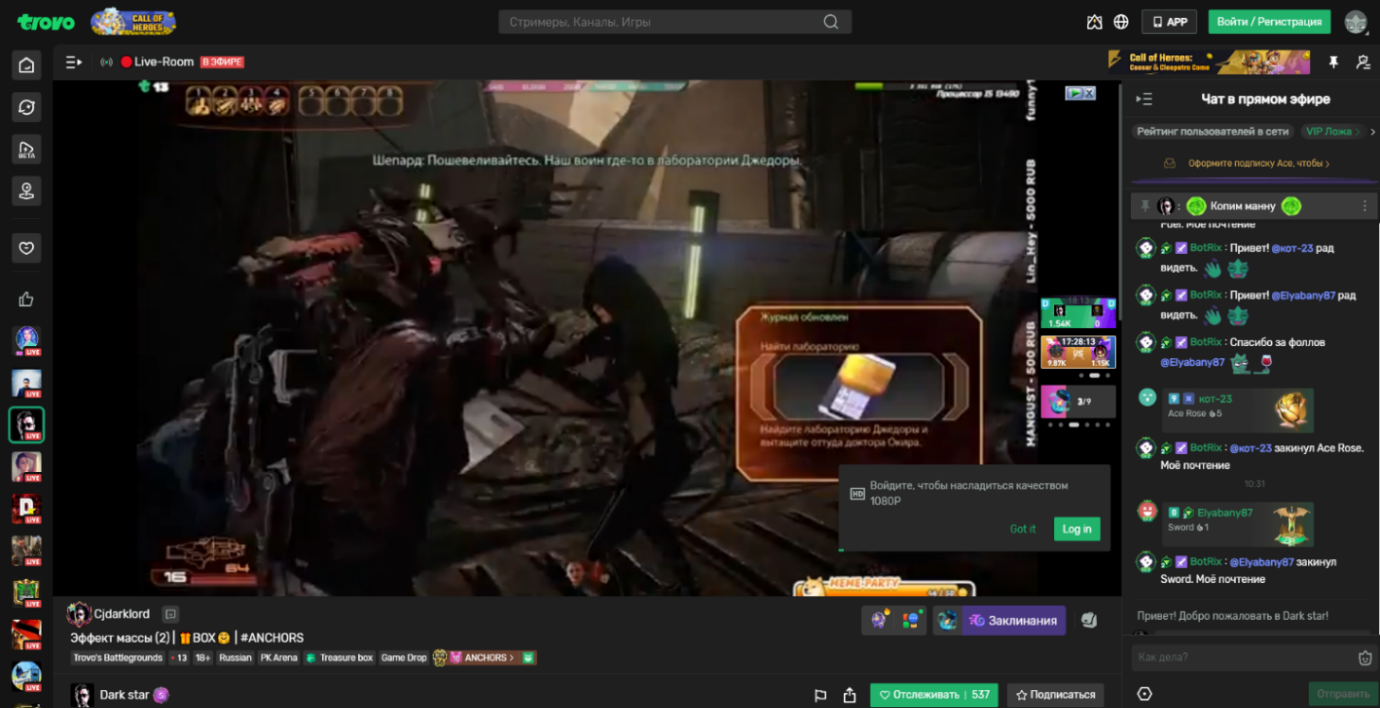
****

Рисунок 2 – Trovo

VK Play Live, как и Twitch, в первую очередь ориентирован на видеоигры и игроков. Однако имеет значительно меньшую аудиторию, чем Twitch. В основном имеет те же возможности что и конкурент, но при этом имеет пониженную комиссию. Основной отличительной чертой можно назвать то, что он ориентирован на Россию и внедрён в игровой сервис VK Play. [10]

**Kick**

Kick – видеостриминговый сервис, основанный одноимённой компанией в 2022 году при поддержке сайта Stake.com как конкурент Twitch, с акцентом на более свободную модерацию и более высокую долю доходов для стримеров.

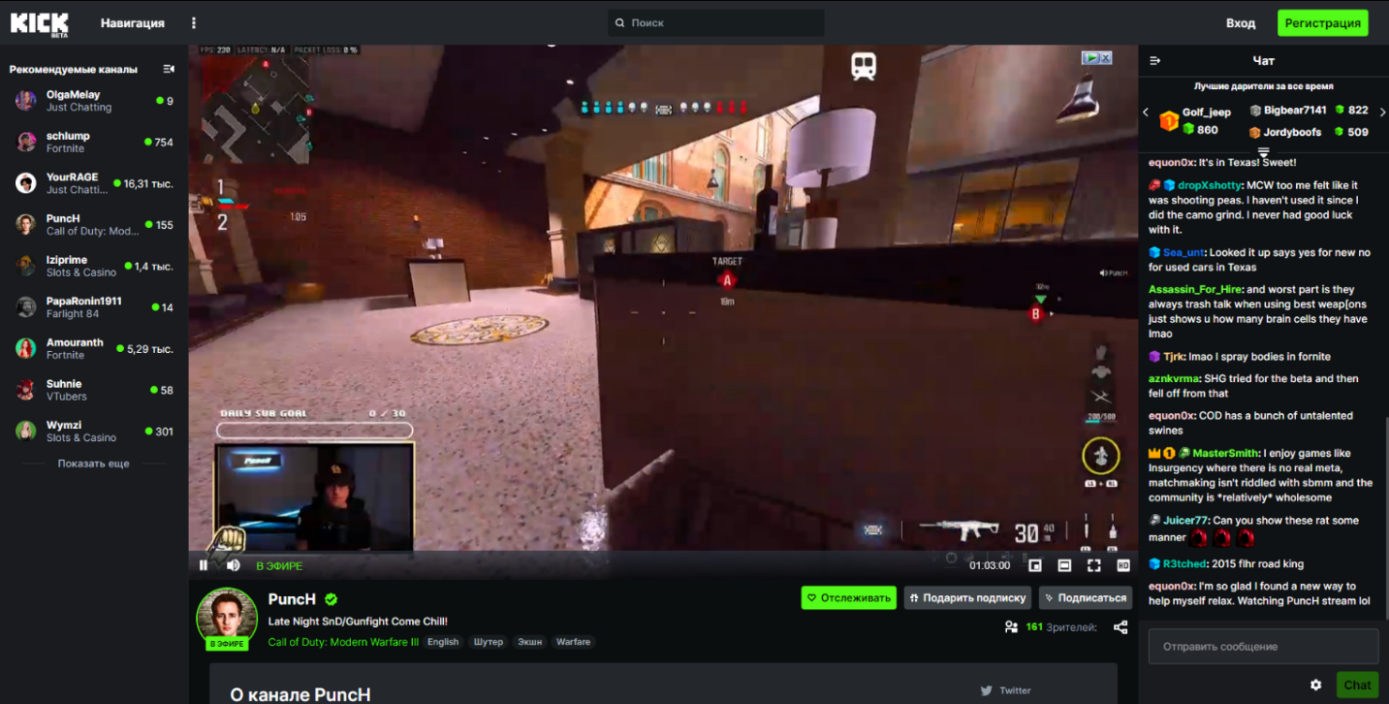
****

Рисунок 3 – Kick

Предлагает тот же функционал что и Twitch, но имеет и некоторые преимущества. К примеру, Kick не показывает рекламу во время трансляций. Kick предоставляет большую свободу создателям контента, а также самую выгодную для стримеров систему монетизации (95%/5%). [11]

Таким образом, можно выделить набор обязательных функций, которые следует реализовать при разработке подобного веб-приложения:

* Предоставлять возможность авторизации на сервисе;
* Предоставлять возможность проведения прямых трансляций на сервисе;
* Предоставлять возможность поиска каналов и трансляций;
* Предоставлять возможность переписке в общем чате трансляции.
  1. Анализ существующих решений для реализации проекта

На сегодняшний день существует немало систем для создания веб-приложений. Рассмотрим наиболее популярные технологии.

При создании интерфейса (frontend), разработчики обычно используют, такие языки программирования как HTML, CSS и JavaScript, которые позволяют создавать различные элементы интерфейса, определять их внешний вид и поведение. [7]

При разработке также используются наборы уже написанного кода, и специальных инструментов, предоставляющих готовые функции и структуры, которые разработчики могут использовать для упрощения определенных задач разработки. Такие наборы инструментов называют библиотеками и фреймворками. Вот некоторые популярные из них [5]:

* React.js: Библиотека от Facebook для создания пользовательских интерфейсов. Очень популярна за счет своей гибкости и мощного экосистема. [2]
* Vue.js: Прогрессивный фреймворк для построения пользовательских интерфейсов. Легкий в изучении и интеграции в проекты, что делает его привлекательным для стартапов и малых проектов. [13]
* Angular: Платформа и фреймворк для создания SPA (одностраничных приложений), разработанный Google. Предоставляет мощные инструменты для разработки. [14]
* Tailwind: CSS-фреймворк, ориентированный на полезность, содержит множество классов, которые можно комбинировать для создания любого дизайна прямо в вашей разметке. [4]
* Bootstrap: бибилиотека готовых компонентов для создания веб-страниц. [15]

При выполнении работы я выбрал библиотеку React для JS, потому что она достаточно популярна и уже прошла проверку временем, над её поддержкой работает крупная компания. Также эту библиотеку мы проходили на курсе веб-программирования. Помимо этого, я выбрал Tailwind для изменения внешнего вида веб-страницы.

Для бэкенда, то есть для той части, которая отвечает за логику приложения на сервере, также существуют свои языки программирования и фреймворки. На сегодняшний день самыми популярными являются следующие решения [5]:

* Python: универсальный язык программирования, хорошо подходит для веб-разработки. [21]
* JavaScript: с помощью Node.js JavaScript можно использовать и на сервере. [16]
* PHP: язык программирования, изначально созданный для веб-разработки. [17]
* Java: популярный язык программирования, подходит для создания сложных веб-приложений. [18]
* Django: фреймворк Python, известный своей простотой и удобством использования. [19]
* Next.js: полнофункциональный фреймворк React для создания веб-приложений. [3]
* Spring: фреймворк Java, хорошо подходит для создания масштабируемых веб-приложений. [20]

При выполнении работы я выбрал программную платформу Node.js и фреймворк Next.js. Node.js предоставляет хорошую производительность, широкий выбор модулей и библиотек. Next.js в свою очередь предоставляет удобную систему папок, простую интеграцию с API и неплохую производительность.

Ещё важным элементом являются системы управления базами данных. СУБД – это программное обеспечение, которое позволяет организовывать и хранить данные, а также управлять ими. [6]

* MySQL: свободная реляционная система управления базами данных. Одна из популярнейших СУБД. [22]
* PostgreSQL: это продвинутая объектно-реляционная система баз данных с открытым исходным кодом. [23]
* MongoDB: NoSQL база данных, хорошо подходит для работы с JSON-документами. [24]

В качестве систему управления базами данных я выбрал MySQL, так как она является одной из популярнейших СУБД, распространяется бесплатно с открытым исходным кодом, и при этом предоставляет оптимальную производительность и стабильность.

Для реализации потоковой трансляции видео будет использоваться LiveKit – набор инструментов с открытым исходным кодом, обеспечивающий масштабируемую многопользовательскую конференц-связь на основе WebRTC. Выбор пришёлся на LiveKit поскольку это проект с открытым исходным кодом, а также он предоставляет бесплатный пробный период использования.

1. ТРЕБОВАНИЯ К СИСТЕМЕ
   1. Функциональные требования

В ходе работы мной были сформулированы следующие функциональные требования, которые обеспечат удобство использования и широкие возможности для пользователей нашего веб-приложения.

* + Пользователи должны иметь возможность создавать учетные записи и входить в систему с использованием уникального логина и пароля.
  + Приложение должно предоставлять возможность поиска каналов и трансляций по названию.
  + Пользователи должны иметь возможность просматривать живые трансляции.
  + Пользователи должны иметь возможность начинать и завершать собственные трансляции с помощью специальных приложений.
  + Приложение должно поддерживать возможность подписки на каналы стримеров.
  + Пользователи должны иметь возможность писать сообщения во встроенный групповой чат для общения друг с другом и со стримером во время трансляции.
  + Пользователи, ведущие трансляцию, должны иметь возможность блокировать других пользователей.
  1. Нефункциональные требования

Таким же образом были сформулированы следующие нефункциональные требования.

* + Веб-приложение должно быть разработано с использованием HTML, CSS, TypeScript;
  + Веб-приложение должно быть разработано с помощью библиотеки React и фреймворков Next.js, Tailwind CSS;
  + Для реализации потокового вещания будет использоваться инструментарий LiveKit.
  + База данных должна быть реализована с помощью MySQL.

1. ПРОЕКТИРОВАНИЕ
   1. Диаграмма вариантов использования

В процессе определения требований к веб-приложению была сформирована диаграмма вариантов использования. В процессе определения требований к веб-приложению была сформирована диаграмма вариантов использования.

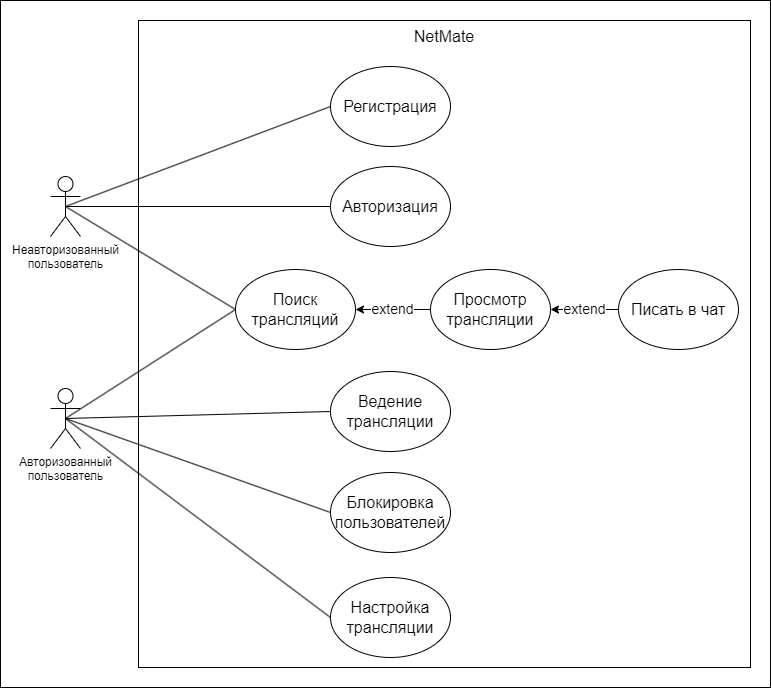


Рисунок 4 – Диаграмма вариантов использования

**Основные актёры, взаимодействующие с системой**

В данной диаграмме представлены два основных актера: авторизованный пользователь и неавторизованный пользователь.

Авторизованный пользователь – это зарегистрированный пользователь, авторизовавшийся в системе, который может просматривать трансляции и взаимодействовать с контентом.

Неавторизованный пользователь – это любой человек, который посещает платформу без регистрации и входа в систему.

**Краткое описание вариантов использования**

У актёров есть разные варианты использования. Так неавторизованный пользователь имеет возможность для регистрации и авторизации, также он может производить поиск трансляций и просматривать их, но не может общаться в чате.

Авторизованный пользователь в свою очередь не может зарегистрировать новый аккаунт или авторизоваться под другими данными, для этого ему потребуется выйти из своей учётной записи. Но при этом ему доступны все остальные функции платформы. Авторизованный пользователь может искать и смотреть трансляции, а также самостоятельно проводить трансляции. Можно настраивать трансляцию изменяя её название и обложку. Также при настройке трансляции пользователю необходимо сгенерировать и сохранить уникальный ключ трансляции, для того чтобы стримить на платформе.

Спецификация вариантов использования можно найти в приложении к работе.

* 1. Проектирование интерфейса

У рассмотренных аналогов интерфейсы очень похожи. Поэтому за основу я взял интерфейс платформы Twitch, для того чтобы пользователям, которые использовали такие платформы ранее было легче ориентироваться в моём приложении.

* 1. Диаграмма компонентов

В процессе определения требований к веб-приложению была сформирована диаграмма компонентов.

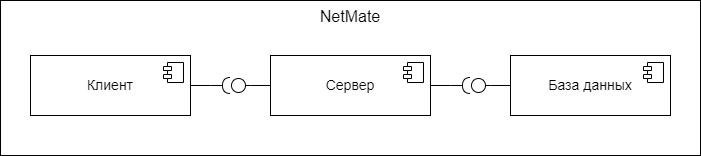


Рисунок 4 – Диаграмма вариантов использования

Для разработки за основу взята трёхуровневая архитектура. Архитектурная модель, предполагающая наличие в веб-приложении трёх типов компонентов: клиентское приложение, сервер приложения и сервер базы данных.

* 1. Проектирование базы данных

База данных предназначена для хранения, обработки и изменения информации, выстраивания взаимосвязей. В этой работе она необходима для работы с данными пользователей.

Для работы с базой данных выбрана СУБД MySQL. Это реляционная система управления базами данных с открытым исходным кодом. Реляционная модель довольно удобная и производительная, также она позволяет легко масштабировать данные без нарушения структуры базы.

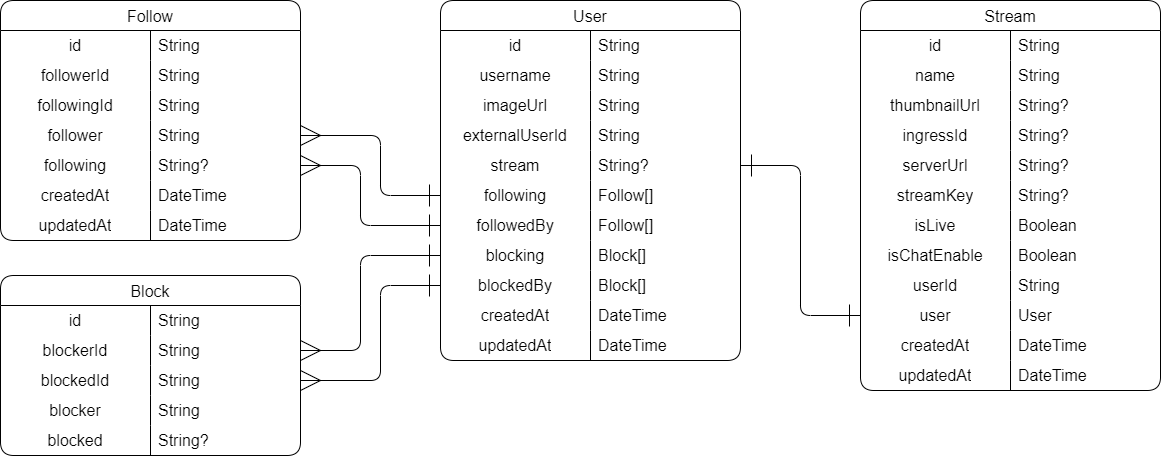


Рисунок 5 – Схема базы данных

1. РЕАЛИЗАЦИЯ
2. ТЕСИРОВАНИЕ

ЗАКЛЮЧЕНИЕ

ЛИТЕРАТУРА

1. SimilarWeb – сервиса для анализа сайтов. [Электронный ресурс] URL: https://www.similarweb.com (дата обращения: 3.03.2024).
2. React – библиотека для JavaScript [Электронный ресурс] URL: https://react.dev (дата обращения: 01.03.2024).
3. Next.js – фреймворк для React. [Электронный ресурс] URL: https://nextjs.org (дата обращения: 01.04.2024).
4. Tailwind CSS – CSS-фреймворк. [Электронный ресурс] URL: https://tailwindcss.com (дата обращения: 01.04.2024).
5. Statista – Портал статистики. [Электронный ресурс] URL: <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/> (дата обращения: 20.04.2024).
6. DB-Engines – База знаний о СУБД. [Электронный ресурс] URL: <https://db-engines.com/en/ranking> (дата обращения: 1.05.2024).
7. Айтилогия – Онлайн-школа. [Электронный ресурс] URL: <https://itlogia.ru/article/html_css_javascript_i_php_chto_takoe_i_dlya_chego> (дата обращения: 1.05.2024).
8. Twitch – Видеостриминговый сервис. [Электронный ресурс] URL: https://www.twitch.tv (дата обращения: 5.04.2024).
9. 3DNews – российское онлайн-издание, посвящённое цифровым технологиям. [Электронный ресурс] URL: https://3dnews.ru/1097451/o-vremena-o-nravy-twitch-smyagchil-pravila-pokaza-seksualnogo-kontenta (дата обращения: 5.04.2024).
10. RBC – медиахолдинг. [Электронный ресурс] URL: <https://www.rbc.ru-/life/news/6308acd09a7947f6f1995ea4> (дата обращения: 5.04.2024).
11. Internet Matters – медиа-портал. [Электронный ресурс] URL: https://www.internetmatters.org/ru/hub/news-blogs/what-is-kick-streaming/ (дата обращения: 5.04.2024).
12. Onlypult – платформа для работы с социальными медиа. [Электронный ресурс] URL: https://onlypult.com/ru/blog/media-buduschego-7-sposobov-ispolzovat-striming (дата обращения: 5.04.2024).
13. Vue.js – JavaScript-фреймворк. [Электронный ресурс] URL: https://vuejs.org (дата обращения: 5.04.2024).
14. Angular – JavaScript-фреймворк. [Электронный ресурс] URL: https://angular.io (дата обращения: 5.04.2024).
15. Bootstrap – библиотека для работы c HTML, CSS и JS. [Электронный ресурс] URL: https://getbootstrap.com (дата обращения: 5.04.2024).
16. Node.js – среда выполнения JavaScript. [Электронный ресурс] URL: https://nodejs.org/en (дата обращения: 5.04.2024).
17. Php – язык программирования. [Электронный ресурс] URL: https://www.php.net (дата обращения: 5.04.2024).
18. Java – язык программирования. [Электронный ресурс] URL: https://vuejs.org (дата обращения: 5.04.2024).
19. Django – Python-фреймворк. [Электронный ресурс] URL: https://www.djangoproject.com (дата обращения: 5.04.2024).
20. Spring – Java-фреймворк. [Электронный ресурс] URL: https://spring.io (дата обращения: 5.04.2024).
21. Python – язык программирования. [Электронный ресурс] URL: https://www.python.org (дата обращения: 5.04.2024).
22. MySQL – система управления базами данных. [Электронный ресурс] URL: https://www.mysql.com (дата обращения: 5.04.2024).
23. PostgreSQL – система управления базами данных. [Электронный ресурс] URL: https://www.postgresql.org (дата обращения: 5.04.2024).
24. MongoDB – система управления базами данных. [Электронный ресурс] URL: https://www.mongodb.com (дата обращения: 5.04.2024).
25. Business of Apps. [Электронный ресурс] URL: https://www.mongo-db.com (дата обращения: 16.05.2024).
26. LiveKit. [Электронный ресурс] URL: https://livekit.io (дата обращения: 18.05.2024).

ПРИЛОЖЕНИЕ

Таблица 1 – Спецификация варианта использования Авторизация

|  |
| --- |
| ***UseCase:* Авторизация** |
| *ID:* 1 |
| *Аннотация:* Авторизация пользователя |
| *Главные актеры:* Неавторизованный пользователь |
| *Второстепенные актеры:* нет |
| *Предусловия:* Пользователь не авторизован |
| *Основной поток:*   1. Пользователь вводит данные для авторизации |
| *Постусловия:* Пользователь авторизован |
| *Альтернативные потоки:* 2 |

Таблица 2 – Спецификация варианта использования Регистрация

|  |
| --- |
| ***UseCase:* Регистрация** |
| *ID:* 2 |
| *Аннотация:* Регистрация пользователя |
| *Главные актеры:* Неавторизованный пользователь |
| *Второстепенные актеры:* нет |
| *Предусловия:* Незарегистрированный пользователь пытается авторизоваться в системе |
| *Основной поток:*   1. Пользователь вводит данные для регистрации |
| *Постусловия:* Пользователь зарегистрирован |
| *Альтернативные потоки:* нет |

Таблица 3 – Спецификация варианта использования Ведение трансляции

|  |
| --- |
| ***UseCase:* Ведение трансляции** |
| *ID:* 3 |
| *Аннотация:* Пользователь запускает трансляцию |
| *Главные актеры:* Стример |
| *Второстепенные актеры:* нет |
| *Предусловия:* Стример запускает трансляцию |
| *Основной поток:*   1. Пользователь ведет прямую трансляцию контента на платформу |
| *Постусловия:* Пользователь ведет трансляцию |
| *Альтернативные потоки:* нет |

Таблица 4 – Спецификация варианта использования Настройка трансляции

|  |
| --- |
| ***UseCase:* Настройка трансляции** |
| *ID:* 4 |
| *Аннотация:* Стример производит настройку трансляции |
| *Главные актеры:* Стример |
| *Второстепенные актеры:* нет |
| *Предусловия:* Пользователь заходит в меню настройки трансляции |
| *Основной поток:*   1. Пользователь изменяет параметры трансляции |
| *Постусловия:* Произведена настройка трансляции |
| *Альтернативные потоки:* нет |

Таблица 5 – Спецификация варианта использования Настройка трансляции

|  |
| --- |
| ***UseCase:* Поиск трансляции** |
| *ID:* 5 |
| *Аннотация:* Зритель производит поиск трансляции по названию |
| *Главные актеры:* Зритель |
| *Второстепенные актеры:* нет |
| *Предусловия:* Пользователь хочет найти трансляцию по названию |
| *Основной поток:*   1. Пользователь вводит поисковый запрос в системе 2. Система выдаёт перечень подходящих трансляций |
| *Постусловия:* Пользователь нашёл перечень подходящих трансляций |
| *Альтернативные потоки:* нет |

Таблица 6 – Спецификация варианта использования Писать в чат

|  |
| --- |
| ***UseCase:* Писать в чат** |
| *ID:* 6 |
| *Аннотация:* Пользователь пишет сообщение в групповой чат |
| *Главные актеры:* Зритель |
| *Второстепенные актеры:* нет |
| *Предусловия:* Пользователь хочет написать сообщение в чат трансляции |
| *Основной поток:*   1. Пользователь пишет сообщение в форму и отправляет 2. Отправленное сообщение становится видно всем зрителям трансляции |
| *Постусловия:* Пользователь отправил |
| *Альтернативные потоки:* нет |

Таблица 7 – Спецификация варианта использования Настройка трансляции

|  |
| --- |
| ***UseCase:* Настройка трансляции** |
| *ID:* 7 |
| *Аннотация:* Настройки трансляции |
| *Главные актеры:* Стример |
| *Второстепенные актеры:* нет |
| *Предусловия:* Пользователь заходит в меню настройки трансляции |
| *Основной поток:*   1. Пользователь изменяет параметры трансляции |
| *Постусловия:* Произведена настройка трансляции |
| *Альтернативные потоки:* нет |

Таблица 8 – Спецификация варианта использования Настройка трансляции

|  |
| --- |
| ***UseCase:* Настройка трансляции** |
| *ID:* 8 |
| *Аннотация:* Настройки трансляции |
| *Главные актеры:* Стример |
| *Второстепенные актеры:* нет |
| *Предусловия:* Пользователь заходит в меню настройки трансляции |
| *Основной поток:*   1. Пользователь изменяет параметры трансляции |
| *Постусловия:* Произведена настройка трансляции |
| *Альтернативные потоки:* нет |