Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Северо-Восточный федеральный университет имени М.К. Аммосова»

Колледж инфраструктурных технологий

Кафедра эксплуатации и обслуживания информационных систем

Спецификация веб-приложения

Twitch

|  |
| --- |
| Выполнил работу Винокуров Ян Русланович ИСИП-21-5.  Проверил: Саввин Ф.Г.  Оценка: \_\_\_\_\_\_\_\_\_ |

Г. Якутск – 2024г

Оглавление

[1. **Общее описание** 3](#_Toc157587578)

[1.1. Цель и функции 3](#_Toc157587579)

[1.2. Предполагаемая аудитория и пользователи приложения 3](#_Toc157587580)

[1.3. Технические требования и ограничения 3](#_Toc157587581)

[1.4. Внешние системы и сервисы, с которыми приложение должно взаимодействовать 4](#_Toc157587582)

[2. **Функциональные требования** 4](#_Toc157587583)

[2.1. Перечень основных функций и возможности Twitch 4](#_Toc157587584)

[2.2. Описание пользовательских сценариев и шагов для выполнения операций 4](#_Toc157587585)

[2.3. Права доступа и роли пользователей 5](#_Toc157587586)

[2.4. Параметры и ограничения для вводимых данных 5](#_Toc157587587)

[2.5. Обработка ошибок и исключительных ситуаций 5](#_Toc157587588)

[2.6. Интеграция с внешними системами и API 5](#_Toc157587589)

[3. **Технические требования и архитектура** 5](#_Toc157587590)

[3.1. Версии и требования по использованию программного обеспечения, языков программирования, баз данных и серверов 5](#_Toc157587591)

[3.2. Структура базы данных и модель данных 6](#_Toc157587592)

[3.3. Архитектура приложения 6](#_Toc157587593)

[3.4. Масштабируемость, надежность и безопасность 6](#_Toc157587594)

[4. **Управление проектом** 6](#_Toc157587595)

[4.1. Определение этапов разработки и сроков выполнения 6](#_Toc157587596)

[4.2. Основная команда и их роли 7](#_Toc157587597)

[4.3. Оценка затрат 8](#_Toc157587598)

[4.4. Расписание выполнения работ 8](#_Toc157587599)

1. Общее описание
   1. Цель и функции

Цель Twitch – это предоставление платформы для стриминга и просмотра веб-телевидения в режиме реального времени. Основанная в 2011 году, Twitch стала популярной среди игроков видеоигр и стримеров, позволяя им вещать свою игровую сессию или другое веб-телевидение в прямом эфире. Однако в последние годы платформа расширила свою аудиторию, основывая свою состоятельность на различных категориях контента, таких как кулинария, музыка, обучение и многое другое.

Функции Twitch включают:

1. Стриминг: Twitch предоставляет возможность пользователям отправлять свой контент в прямом эфире, включая видеоигры, полезные советы, кулинарные приготовления и другие виды контента.

2. Чат: Пользователям доступен интерактивный чат, где зрители могут общаться с стримерами и другими зрителями.

3. Подписки: Пользователи могут подписываться на стримеров, чтобы получать определенные преимущества, такие как эмоции, специальные значки и доступ к эксклюзивному контенту.

4. Комментарии и обратная связь: Зрители могут оставлять комментарии и оценивать контент, что позволяет стримерам получать обратную связь и улучшать свою трансляцию.

5. Игровой контент: Twitch активно поддерживает сообщество игровых стримеров, предлагая инструменты и функции, такие как интеграция с популярными видеоиграми, панель инструментов для стримеров и другие функции, сделанные специально для геймеров.

6. Сотрудничество и спонсорство: Twitch позволяет стримерам сотрудничать с различными компаниями и брендами, например, через партнерскую программу или спонсорированные контент и передачи.

7. Кураторство и открытость: Twitch активно работает над тем, чтобы предоставлять возможности для развития кураторства и поддержки разнообразного контента, поднимая на передний план подпредставленные голоса и создавая открытую среду для всех пользователей.

* 1. Предполагаемая аудитория и пользователи приложения

Предполагаемая аудитория Twitch - это люди, интересующиеся видеоиграми, стримингом, электронной спортом и различными развлекательными контентами. Пользователи Twitch могут быть как игроками, стримерами, так и просто зрителями, которые просто наблюдают за игровым процессом или общаются с другими пользователями через чат. Также многие профессиональные команды и игроки используют Twitch для трансляции своих матчей и тренировок. Среди аудитории Twitch можно найти как подростков, так и взрослых, так как интерес к видеоиграм и стримингу не ограничивается возрастом.

* 1. Технические требования и ограничения

Технические требования и ограничения для Twitch могут включать следующее:

1. Интерфейсы и API: Twitch предоставляет различные API и SDK для разработчиков, позволяющие интегрировать функции Twitch во внешние приложения. Они включают Twitch API, Twitch Embed Video Player API, Twitch Chat API и множество других.

2. Авторизация и безопасность: Приложение должно соответствовать стандартам безопасности Twitch. Это включает использование OAuth и Twitch Authentication для авторизации пользователей и получения доступа к конкретным функциям и данным.

3. Потоковая передача: Приложение должно обеспечивать возможность просмотра и потоковой передачи контента на Twitch. Это может включать использование Twitch Video Ingest Servers для передачи видео.

4. Чат и уведомления: Приложение должно поддерживать взаимодействие с Twitch Chat для отправки и получения сообщений, а также с системой уведомлений для отображения информации о событиях и обновлениях.

5. Ограничения и правила: Twitch устанавливает различные ограничения и правила для использования и интеграции своей платформы. Например, есть ограничения на использование контента, массовую рассылку сообщений и другие вещи, которые могут повлиять на функциональность приложения.

* 1. Внешние системы и сервисы, с которыми приложение должно взаимодействовать

Внешние системы и сервисы, с которыми приложение должно взаимодействовать на Twitch, они могут быть различными в зависимости от конкретного приложения или функциональности. Некоторые из них могут включать:

1. Операционные системы: Приложение должно быть совместимо с платформой или операционной системой пользователя, на которой оно будет работать. Например, Windows, macOS, Android или iOS.

2. Видео-кодеки и медиа-проигрыватели: Приложение может взаимодействовать с различными видео-кодеками и медиа-проигрывателями для воспроизведения видео-контента на Twitch.

3. Системы передачи данных и сетевые протоколы: Приложение должно использовать сетевые протоколы и технологии передачи данных, такие как TCP/IP, HTTP(S), WebSocket и др., для связи с Twitch.

4. Серверы и хостинг: Приложение может использовать серверы и хостинг-провайдеры для развертывания и хранения данных, веб-страниц, изображений и других ресурсов.

1. Функциональные требования
   1. Перечень основных функций и возможности Twitch

Стриминг: позволяет пользователям транслировать свою игру или другой контент в режиме реального времени.

Чат: пользователи могут общаться друг с другом и смотреть стрим в аналогичные время.

Премиум-подписка: пользователи могут подписаться на особые каналы для получения дополнительных возможностей и преимуществ.

Мобильное приложение: Twitch доступен также на мобильных устройствах, что позволяет смотреть стримы в любом месте.

* 1. Описание пользовательских сценариев и шагов для выполнения операций

Чтение стримов: пользователь открывает Twitch и просматривает список стримов, выбирает интересующий его контент и начинает просмотр.

Вещание: пользователь создает канал, настраивает программу для стриминга, подключает свою игру или другой контент и начинает трансляцию.

Участие в чате: пользователь открывает чат к выбранному стриму и присоединяется к общению с другими зрителями и стримером.

Подписка и донаты: пользователь может подписаться на партнерский канал или сделать донат, чтобы поддержать стримера финансово.

Настройки аккаунта: пользователь может настроить свой профиль, установить изображение аватара, изменить никнейм.

* 1. Права доступа и роли пользователей

Зритель: обычный пользователь, может просматривать стримы, участвовать в чате.

Стример: пользователь, транслирующий свои стримы и управляющий своим каналом.

Партнер: стример, участвующий в партнерской программе, получающий дополнительные возможности и заработок через Twitch.

Администратор: сотрудник Twitch, обладающий расширенными правами по управлению и модерации платформы.

* 1. Параметры и ограничения для вводимых данных

Никнейм: должен быть уникальным и соответствовать правилам платформы (нет оскорбительных или запрещенных слов).

Изображение аватара: ограниченное разрешение и размер файла.

Стримы: могут быть ограничены по разрешению и битрейту в зависимости от качества подключения пользователя.

* 1. Обработка ошибок и исключительных ситуаций

Сообщения об ошибках: при возникновении ошибок (например, проблемы с соединением) пользователь получает соответствующее уведомление.

Поддержка пользователей: Twitch предлагает поддержку пользователям для решения проблем и ответов на вопросы через специальные контактные формы или форумы поддержки.

* 1. Интеграция с внешними системами и API

API Twitch: позволяет разработчикам интегрировать свои приложения и сервисы с Twitch, получать доступ к данным о стримах, пользователях, чате и т. д.

Внешние системы: Twitch имеет возможность интеграции с другими платформами и социальными сетями для расширения функциональности и обмена данными.

1. Технические требования и архитектура
   1. Версии и требования по использованию программного обеспечения, языков программирования, баз данных и серверов

- Для разработки и поддержки Twitch используются различные языки программирования, включая Python, Ruby, Java, PHP и Go.

- Основной язык программирования для веб-приложений Twitch - JavaScript с использованием фреймворка React.

- Для работы с базами данных Twitch используются различные технологии, включая MySQL, PostgreSQL и Redis.

- Для хостинга серверов Twitch использует различные провайдеры облаков, включая Amazon Web Services (AWS).

* 1. Структура базы данных и модель данных

- База данных Twitch состоит из нескольких компонентов, включая информацию о пользователях, стримах, видео, чатах и других.

- Twitch использует реляционные и нереляционные базы данных для хранения и обработки данных.

- Модель данных в Twitch строится в соответствии с требованиями к хранению информации о пользователях, стримах, видео, чатах и другой функциональности платформы.

* 1. Архитектура приложения

- Твич использует клиент-серверную архитектуру для обеспечения взаимодействия между клиентскими устройствами (пользовательскими приложениями или веб-браузерами) и серверами.

- В Twitch также используется микросервисная архитектура, где каждая функциональность платформы предоставляется отдельным сервисом.

- Микросервисы в Twitch могут взаимодействовать друг с другом через API или сообщения.

* 1. Масштабируемость, надежность и безопасность

- Масштабируемость Twitch обеспечивается при помощи горизонтального масштабирования и использования облачных сервисов.

- Надежность Twitch обеспечивается с помощью балансировки нагрузки, резервирования системных ресурсов, резервного копирования данных и других мер безопасности.

- Безопасность в Twitch обеспечивается с помощью шифрования данных, многоуровневой аутентификации, системы контроля доступа и других мер безопасности. Также Twitch использует инструменты для обнаружения и предотвращения атак, такие как веб-брандмауэры и системы мониторинга.

1. Управление проектом
   1. Определение этапов разработки и сроков выполнения

|  |  |
| --- | --- |
| Этапы разработки | Сроки выполнения |
| Анализ требований и планирование проекта | 2 недели |
| Проектирование архитектуры и UI/UX пользовательского интерфейса | 3 недели |
| Развитие серверной инфраструктуры и базы данных | 4 недели |
| Разработка основных функциональных возможностей | 8 недель |
| Тестирование, откладка и устранение ошибок | 2 недели |
| Внедрение и релиз платформы | 1 неделя |
| Поддержка и обновление функционала | Постоянная |

* 1. Основная команда и их роли

|  |  |
| --- | --- |
| Роль | Смысл этой роли |
| Проект менеджер | Ответственен за планирование, контроль сроков и ресурсов, а также координацию работы команды. |
| Аналитик | Проводит исследование рынка и организует сбор требований от пользователей, осуществляет анализ конкурентов. |
| Архитектор | Разрабатывает общую архитектуру приложения и спецификации для разработчиков. |
| Дизайнер UI/UX | Создает интерфейсы пользователя, обеспечивает эффективность и удобство использования платформы |
| Разработчики (Frontend и Backend) | Занимаются программированием клиентской и серверной частей платформы. |
| Тестировщики | Проводят функциональное тестирование, обнаруживают и исправляют ошибки. |
| Дерлойеры | Устанавливают и настраивают платформу на производственном сервере. |
| Техническая поддержка | Оказывает помощь пользователям, обнаруживает и устраняет проблемы. |

* 1. Оценка затрат

Зарплата команды: определяется исходя из рыночных ставок и уровня опыта членов команды

Техническое оборудование и программное обеспечение: провести анализ стоимости покупки или аренды необходимого оборудования и лицензий на программные системы.

Инфраструктура серверов и облачных услуг: определить затраты на настройку и поддержку серверов и облачных услуг.

Маркетинг и реклама: изучить и оценить стоимость проведения рекламной кампании для привлечения пользователей.

* 1. Расписание выполнения работ
* Месяц 1:

- Анализ требований и планирование проекта

- Архитектура и UI/UX дизайн

- Набор команды разработчиков

* Месяц 2-3:

- Расширение команды разработчиков

- Развитие серверной инфраструктуры и базы данных

* Месяц 4-11:

- Основная разработка функциональных возможностей

- Параллельное тестирование разработанных модулей

* Месяц 12:

- Итоговое тестирование и отладка

- Подготовка к релизу

* Месяц 13:

- Релиз платформы

- Поддержка и обновление функционала