

Архитектура веб-сервиса Twitch.tv

Требования к качеству

Тип приложения

Веб-приложение для выполнения преимущественно на сервере в сценариях с постоянным подключением.

Ключевые характеристики качества

Таблица 1

Внешние характеристики ПО: Оценка	Внутренние характеристики ПО: Оценка
Корректность 6	Удобство сопровождения 10
Практичность 10	Гибкость 8
Эффективность 9	Портируемость 8
Надежность 6	Возможность повторного использования 5
Целостность 6	Удобочитаемость 6
Адаптируемость 8	Тестируемость 6
Правильность 6	Понятность 6
Живучесть 8	

На основе таблицы выделим три наиболее значимых характеристики и сконцентрируемся на них. Таковыми являются: практичность, эффективность, удобство сопровождения. Практичность важна, потому что сервис нацелен на широкую аудиторию и тут необходима простота взаимодействия с сервисом. Эффективность важна, так же из-за обширности целевой аудитории, потому, что скорость работы сервиса очевидно прямо влияет на его успех у конечного пользователя. Удобство сопровождение важно потому, что широкая аудитория предполагает быструю масштабируемость и её не достичь без легкого изменения программной системы. Как следствие важности удобства сопровождения следует важность гибкости системы. Для данного приложения важно также адаптируемость, потому что предполагается использование сервиса в том числе и на мобильных устройствах и как следствие этого во внутренних характеристиках системы становится важной портируемость системы. И наконец, из-за потенциально широкой аудитории и большого траффика (напомню, что Twitch.tv - сервис видеотрансляций) необходимо обеспечить высокую живучесть системы.

Координационная модель

Стиль архитектуры

Трехуровневая архитектура (клиент - сервер - сервер БД)

Ключевые требования к функциональности системы

Основная функциональность системы в предоставлении возможности запустить свою видеотрансляцию через стороннее клиентское ПО и поместить ссылку на данную трансляцию в каталог сервиса. Нужно учитывать большую аудиторию и потому отдавать видео поток не напрямую с машины клиента, а кэшируя на сервере и отдавая конечному пользователю с минимальной задержкой.

Функциональные требования к системе в формате User Story

А - авторизованный пользователь

Н - неавторизованный пользователь

Просмотр каталога трансляций сгруппированные по играм	N	Нажатие на кнопку "Каталог"	Появление страницы со списком трансляций и информации о них
Просмотр трансляции	N	Нажатие на элемент каталога трансляций	Открытие страницы с трансляцией
Комментирование трансляции	A	Ввод сообщения и нажатие на кнопку "Отправить"	Появление сообщения пользователя в блоке комментариев
Создание собственной трансляции	A	Заполнение формы создания трансляции и нажатие на кнопку "Создать"	Появление трансляции в каталоге сервиса

USE-CASE диаграмма

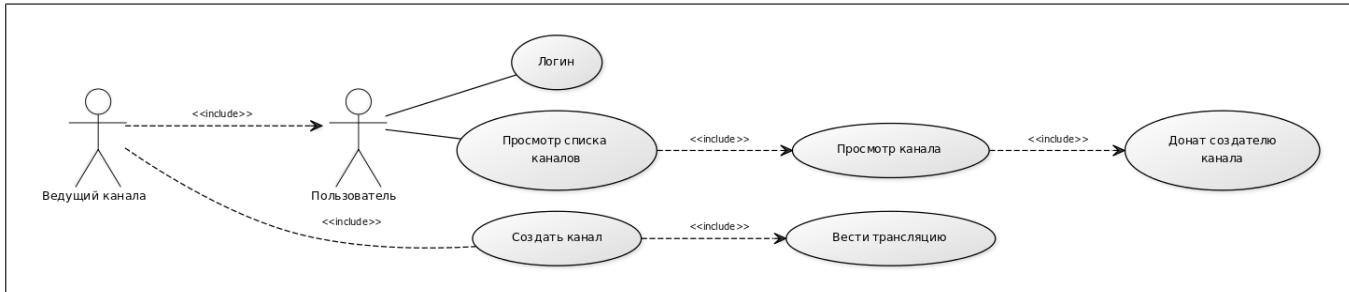
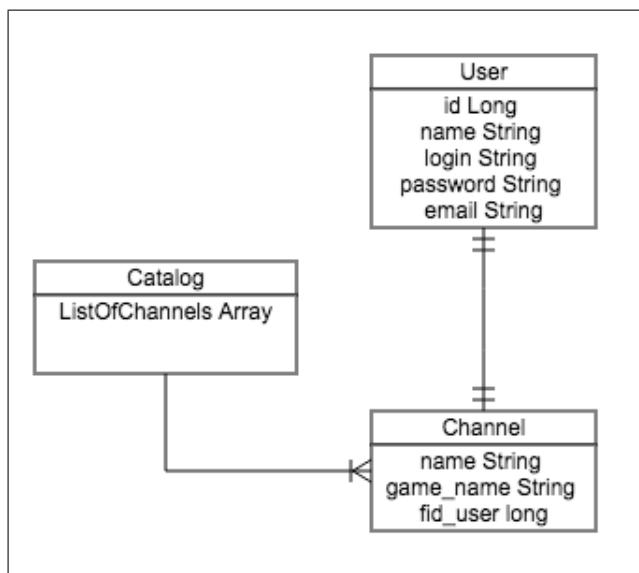


Рис. 1

Основные программные модули системы

1. Модуль регистрации/аутентификации/авторизации (сквозная функциональность)
2. Модуль создания трансляции
3. Модуль просмотра трансляции



Управление ресурсами

Тип хранения данных

СУБД

Реляционная СУБД

Реляционная СУБД: Postgresql. Уровень изоляции: read committed. SQL.

Выбор Postgresql был основан на сравнении с MySQL. У СУБД MySQL есть очень хорошее преимущество для высоконагруженных проектов, это механизм репликации. Репликация — одна из техник масштабирования баз данных. Состоит в том, что данные с одного сервера базы данных постоянно копируются

(реплицируются) на один или несколько других (называемые репликами). Для приложения появляется возможность использовать не один сервер для обработки всех запросов, а несколько. Таким образом можно распределить нагрузку с одного сервера на несколько. Но MySQL использует механизм подключаемых search engines (например InnoDB и MyISAM) и из-за этого репликация реализуется неэффективным способом.

Связи между элементами архитектуры

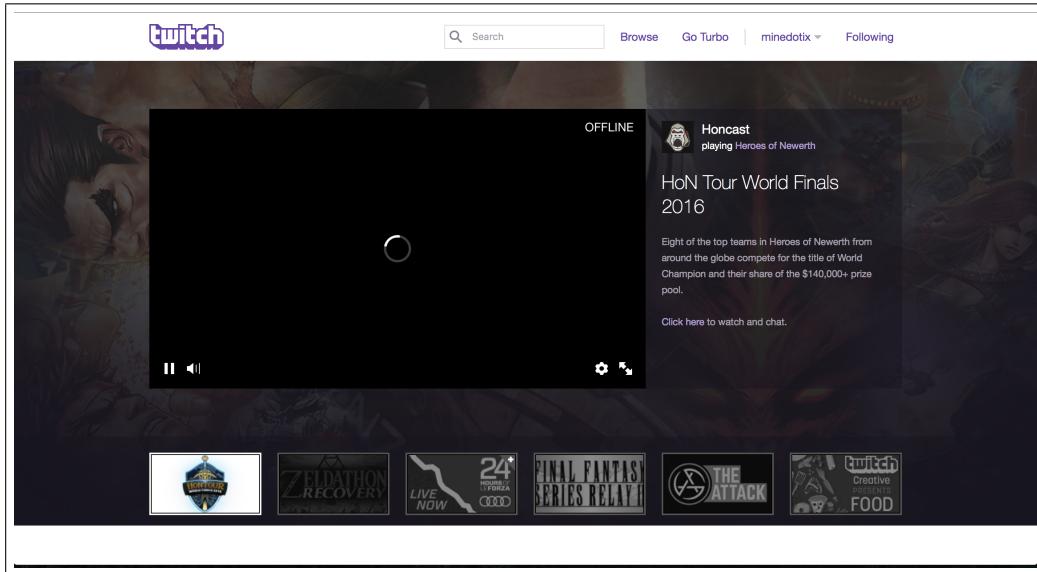


Рис. 2. Главная страница

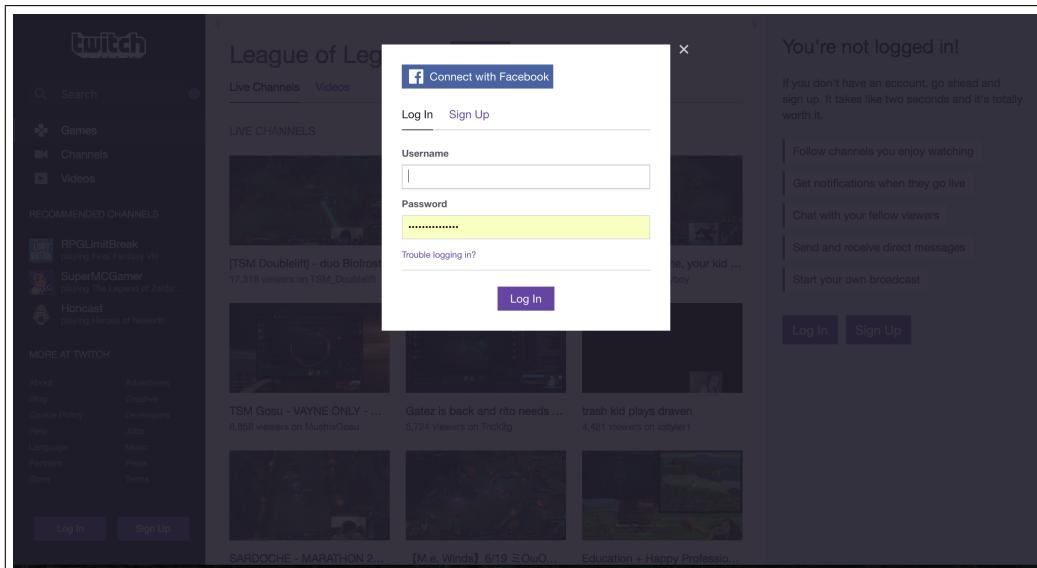


Рис. 3. Страница авторизации

0.1 Выбор технологии

0.1.1 Сервер

Play Framework + Scala + Akka

0.1.2 Клиент

Backbone.js + React.js

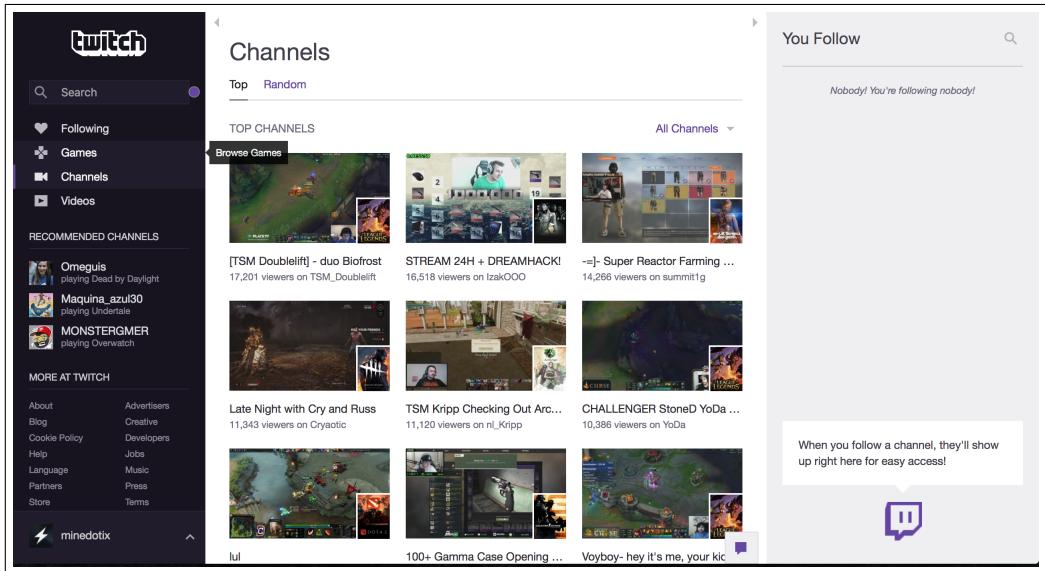


Рис. 4. Каталог каналов

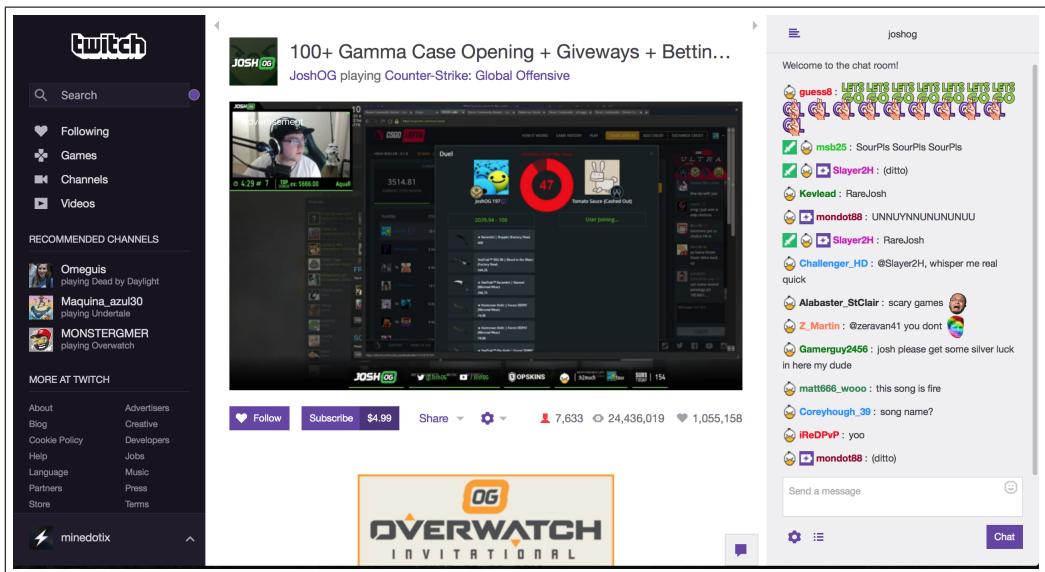


Рис. 5. Просмотр канала