

Федеральное государственное образовательное учреждение высшего
образования

Российский Университет Дружбы Народов

Факультет «Физико-математический факультет»

ДОМАШНЕЕ ЗАДАНИЕ

по дисциплине «Программная инженерия»

Студент: Петрова Мария Евгеньевна

Ст.билет: 1032216450

Группа: НФИбд-02-21

Москва 2023

Видеосервис IVI

IVI - это видеосервис, который предоставляет доступ к широкому выбору фильмов, сериалов, мультфильмов и других видео контента.

Техническое задание (ТЗ) для видеосервиса IVI:

1. Общие слова:

- Цель проекта: Разработка видеосервиса IVI, который предоставляет доступ к широкому ассортименту фильмов и сериалов.
- Основная задача ТЗ: Сведение к минимуму разницы между представлениями заказчика и исполнителя, чтобы обеспечить ясное и подробное описание проекта.

2. Эксплуатационное назначение:

- Видеосервис IVI предназначен для предоставления пользователям возможности просмотра фильмов и сериалов на различных устройствах с подключением к Интернету.

3. Функциональное назначение:

- Разработать веб-платформу, которая обеспечит следующий функционал:
 - Поиск и фильтрация контента по жанру, актерам, режиссерам и другим критериям.
 - Просмотр фильмов и сериалов в потоковом режиме с возможностью переключения языка и настроек отображения.
 - Создание профилей пользователей со списками избранного, историей просмотров и рекомендациями.
 - Обработка платежей и предоставление доступа к премиум-контенту для платных подписчиков.
 - Возможность оценки и добавления отзывов к фильмам и сериалам.

4. Термины и определения:

- Видеосервис IVI: Имя проекта, который разрабатывается.
- Потоковый режим: Функция воспроизведения контента без необходимости его загрузки сначала.
- Платные подписчики: Пользователи, осуществляющие платную подписку на сервис.

5. Данные и списки:

- Список фильмов и сериалов с информацией о них, включая название, описание, жанр, актеров, режиссеров и т.д.
- Список пользователей с профилями, списками избранного, историей просмотров и рекомендациями.

- Данные о платежах подписчиков и статусе их подписки.
6. Страницы с описанием:
- Главная страница с функцией поиска и рекомендаций.
 - Страница описания фильма или сериала, содержащая информацию о нем, трейлер и отзывы.
 - Страница профиля пользователя с его списками избранного, историей просмотров и настройками.
7. Требования к надежности:
- Видеосервис должен быть доступен для пользователей в течение 99% времени без плановых или неплановых перерывов.
 - Должны применяться меры безопасности для защиты данных пользователей и предотвращения несанкционированного доступа.
8. Требования к хостингу:
- Хостинг должен обеспечивать высокую скорость загрузки и потоковое воспроизведение видео контента.
 - Должны быть предусмотрены регулярные резервные копии данных и мониторинг доступности сервиса.
9. Наполнение контентом:
- Платформа должна быть снабжена инструментами для управления и добавления новых фильмов и сериалов в базу данных.
 - Необходимо предусмотреть систему проверки и модерации контента, чтобы предотвратить публикацию незаконного или неприемлемого материала.
10. Сдача и приемка:
- По окончании разработки, заказчик проведет тестирование сервиса для оценки его функционала и соответствия требованиям, описанным в ТЗ.
 - После успешного тестирования и устранения всех замечаний, сервис будет считаться сданным и готовым к выпуску.

Архитектура для видеосервиса IVI:

1. Клиентский интерфейс:
- Веб-приложение: разработано на основе фреймворка React с использованием HTML, CSS и JavaScript. Обеспечивает удобный и респонсивный интерфейс для просмотра видеоконтента на различных устройствах.
 - Мобильное приложение: разработано для платформ iOS и Android с использованием языков программирования Swift и Kotlin. Предоставляет аналогичный функционал веб-приложению.

2. Управление пользователями и аккаунтами:

- Система аутентификации и авторизации: использует протоколы OAuth 2.0 или OpenID Connect для безопасной аутентификации пользователей и выдачи токенов доступа.
- Хранение и управление информацией о пользователях, аккаунтах и предпочтениях просмотра: база данных, такая как PostgreSQL или MongoDB, используется для хранения профилей пользователей и связанных данных.
- Управление подписками и платежами: интеграция с платежными системами, такими как Stripe или PayPal, для обработки подписок и платежей пользователей.

3. Каталог видеоконтента:

- Хранение метаданных о фильмах, сериалах и других видеофайлах: база данных, такая как MySQL или Elasticsearch, используется для хранения информации о фильмах, сериалах, эпизодах, актерах, режиссерах и других связанных данных.
- Категоризация, индексация и тегирование контента: применение алгоритмов обработки естественного языка (Natural Language Processing) для автоматического тегирования и классификации контента по жанрам, рейтингам, актерам и другим атрибутам.
- Хранение видеофайлов: децентрализованные файловые системы, такие как Amazon S3 или Google Cloud Storage, используются для хранения и обеспечения высокой доступности видеофайлов.

4. Механизмы поиска и фильтрации:

- Сервис поиска контента: поиск различных атрибутов контента, таких как название, жанр, актеры и режиссеры, может быть реализован с использованием поисковых индексов, таких как Elasticsearch или Apache Solr, для обеспечения высокой производительности и точности результатов.
- Рекомендации контента: с использованием алгоритмов коллаборативной фильтрации или глубокого обучения (deep learning) для анализа и рекомендаций контента на основе предпочтений и истории просмотра пользователей.

5. Управление воспроизведением:

- Видео-стримеры: использование технологий потоковой передачи, таких как HTTP Live Streaming (HLS) или Dynamic Adaptive Streaming over HTTP (DASH), для обеспечения потокового воспроизведения видео с автоматическим выбором оптимального качества видео в зависимости от доступной пропускной способности.
- Клиентский плеер: разработка специального плеера или использование сторонних плееров, таких как Plyr или Video.js, для управления воспроизведением видео, включая паузу, перемотку, субтитры, аудиодорожки и другие функции.

6. Рекламная поддержка:

- Сервис управления рекламными кампаниями: разработка собственного сервиса для управления рекламными кампаниями, установки ограничений и определения целевых аудиторий, либо интеграция с рекламными платформами, такими как Google Ad Manager или AdMob.

- Вставка рекламных роликов: интеграция с видеорекламными сетями, такими как Google AdSense или VAST/VPaid-плеерами, для вставки рекламных роликов перед, во время или после просмотра видеоконтента.

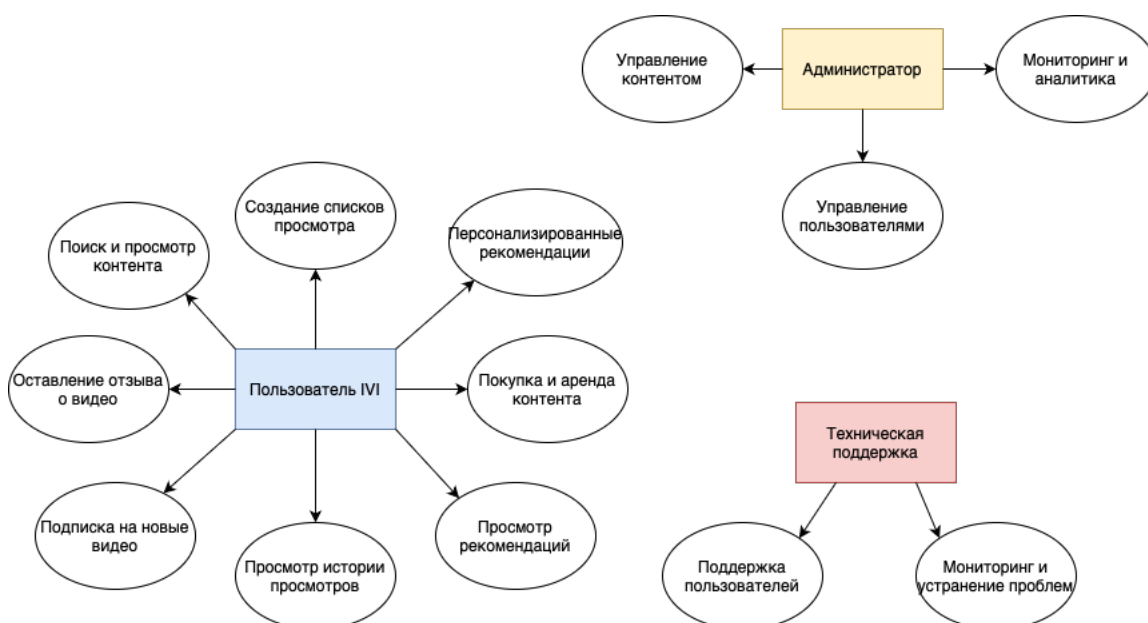
7. Аналитика и мониторинг:

- Сбор и агрегация данных: использование инструментов сбора данных, таких как Google Analytics или самостоятельно разработанные механизмы, для сбора и агрегации информации о поведении пользователей, просмотрах, предпочтениях и других метриках.

- Мониторинг производительности и доступности: применение систем мониторинга, таких как Prometheus или New Relic, для отслеживания производительности, доступности и обработки ошибок сервиса.

Это лишь подробный обзор основных компонентов архитектуры видеосервиса IVI, и конкретные технологии и инструменты могут варьироваться в зависимости от особенностей проекта и требований системы.

Диаграмма Use Case для видеосервиса IVI:



1. Администратор:

- Управление контентом: добавление, редактирование и удаление контента;
- Управление пользователями: добавление, редактирование и удаление пользователей, управление правами доступа;
- Мониторинг и аналитика: просмотр статистики использования сервиса, мониторинг работы системы.

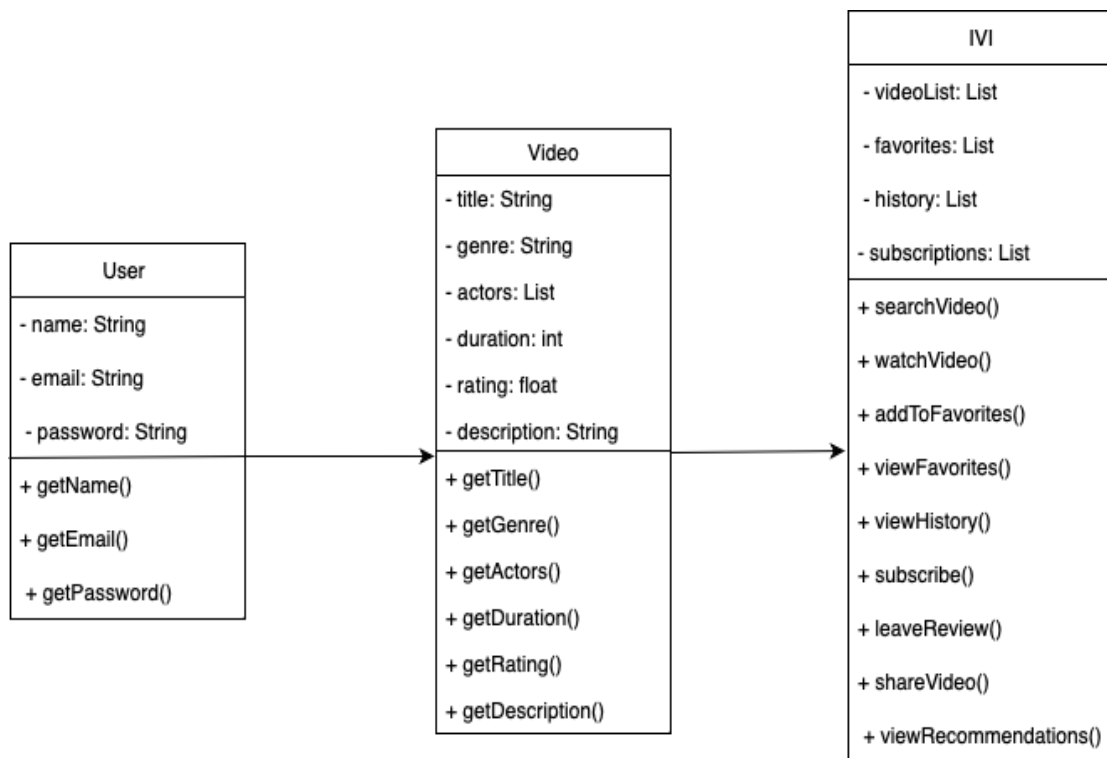
2. Пользователь:

- Поиск и просмотр контента: поиск фильмов, сериалов и другого контента, просмотр контента в онлайн-режиме;
- Создание списков просмотра: добавление фильмов, сериалов и другого контента в список для последующего просмотра;
- Персонализированные рекомендации: получение рекомендаций на основе просмотренного контента;
- Покупка и аренда контента: возможность купить или арендовать фильмы и сериалы, которые не входят в подписку.

3. Техническая поддержка:

- Поддержка пользователей: ответы на вопросы пользователей, помощь в решении проблем с сервисом;
- Мониторинг и устранение проблем: мониторинг работы системы, выявление и устранение технических проблем.

Диаграмма классов для видеосервиса IVI:



Данная диаграмма классов описывает систему видеохостинга IVI. Класс IVI представляет собой главный класс системы, который содержит список видео (videoList), список избранных видео (favorites), список просмотренных видео (history) и список подписок (subscriptions). Он также содержит методы для поиска видео (searchVideo), просмотра видео (watchVideo), добавления видео в избранное (addToFavorites), просмотра списка избранных видео (viewFavorites), просмотра списка просмотренных видео (viewHistory), подписки на каналы (subscribe), оставления отзывов (leaveReview), публикации видео в социальных сетях (shareVideo) и просмотра рекомендаций (viewRecommendations). Класс Video представляет собой видеофайл и содержит информацию о названии (title), жанре (genre), актерах (actors), продолжительности (duration), рейтинге (rating) и описании (description). Он содержит методы для получения информации о названии (getTitle), жанре (getGenre), актерах (getActors), продолжительности (getDuration), рейтинге (getRating) и описании (getDescription). Класс User представляет собой пользователя системы и содержит информацию о его имени (name), электронной почте (email) и пароле (password). Он содержит методы для получения информации об имени (getName), электронной почте (getEmail) и пароле (getPassword).

Класс IVI имеет четыре приватных полей. Он также предоставляет публичные методы для взаимодействия с видеосервисом. Стрелки указывают

на направление зависимости. Класс `iVI` зависит от классов `Video` и `User`, поскольку использует их для своих функций.

Диаграмма последовательности действий:

Первый объект - "`User`" (Пользователь) - представляет собой каждого отдельного пользователя `iVI`. Он имеет возможность выполнить вход (`login`) в систему и выполнять различные действия на платформе.

На диаграмме показаны несколько основных действий, доступных пользователю:

- ``searchVideo()`` (Поиск видео) - пользователь может выполнять поиск видео на платформе, используя различные критерии.
- ``viewVideoDetails()`` (Просмотр деталей видео) - пользователь может просмотреть подробную информацию о выбранном видео.
- ``watchVideo()`` (Просмотр видео) - пользователь может просматривать выбранное видео на платформе.
- ``addToFavorites()`` (Добавить в избранное) - пользователь может добавить видео в свой список избранных для быстрого доступа в будущем.
- ``viewFavorites()`` (Просмотр избранного) - пользователь может просмотреть список видео, которые он добавил в избранное.
- ``viewHistory()`` (Просмотр истории) - пользователь может просмотреть историю просмотренных им видео.
- ``subscribe()`` (Подписка) - пользователь может подписаться на определенный канал или автора видео, чтобы получать обновления о новых видео от них.
- ``leaveReview()`` (Оставить отзыв) - пользователь может оставить отзыв о просмотренном им видео.
- ``shareVideo()`` (Поделиться видео) - пользователь может поделиться ссылкой на видео с другими пользователями.

Дополнительно, на диаграмме показано взаимодействие с объектами класса `Video`, которые представляют видеоматериалы на платформе. Взаимодействия с видео включают поиск, просмотр деталей, просмотр и добавление в избранное. Объекты пользователя и видео взаимодействуют между собой, обмениваясь информацией и вызывая соответствующие методы для выполнения действий на платформе `IVI`.

