

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Мобильное приложение по фильмам

Курсовой проект

по дисциплине

Технологии программирования

09.03.02 Информационные системы и технологии

Информационные системы и технологии управления предприятием

6 семестр 2022/2023 учебного года

Зав. кафедрой	_____	С.Д. Махортов, д.ф.-м. н., профессор
Обучающийся	_____	А.А. Петров, ст. 3 курса оч. отд.
Обучающийся	_____	П.В. Волченко, ст. 3 курса оч. отд.
Руководитель	_____	И.В. Клейменов, ассистент
Руководитель	_____	В.С. Тарасов, ст. преподаватель __.__.20__

Воронеж 2023

Содержание

Содержание	2
Введение	4
1 Постановка задач	5
1.1 Требования к разрабатываемой системе	5
1.1.1 Функциональные требования.....	5
1.1.2 Технические требования.....	6
1.2 Требования к интерфейсу	6
1.3 Задачи, решаемые в процессе разработки.....	6
2 Анализ предметной области	8
2.1 Терминология (гlossарий) предметной области.....	8
2.2 Анализ целевой аудитории	10
2.3 Обзор аналогов	11
2.3.1 IMDb	11
2.3.2 Rotten Tomatoes.....	12
2.3.3 Kinopoisk	13
2.4 Моделирование системы	16
2.4.1 Диаграмма в стиле методологии IDEF0.....	16
2.4.2 Диаграмма активностей	16
2.4.3 Диаграмма прецедентов.....	17
2.4.4 Диаграмма состояний	19
2.4.5 Диаграмма последовательности	19
2.4.6 ER-диаграмма	22
2.4.7 Диаграмма сотрудничества	22
2.5 Аналитика веб-приложения	24
2.6 Границы проекта	25
3 Реализация.....	26
3.1 Средства реализации	26
3.2 Реализация серверной части приложения.....	27

3.2.1 Приложение users	28
3.2.2 Приложение categories	28
3.2.3 Приложение films	29
3.3 Реализация клиентской части приложения.....	30
3.3.1 Форма для просмотра категорий фильмов	31
3.3.2 Форма для просмотра списка фильмов в категории	32
3.3.3 Форма для просмотра статьи о фильме.....	32
3.3.4 Форма для просмотра списка избранного.....	33
3.3.5 Форма для просмотра поиска.....	34
3.3.6 Форма для просмотра профиля.....	35
3.3.7 Форма добавления отзыва к фильму	36
Заключение.....	38
Список использованных источников.....	39

Введение

Мир кинематографа насчитывает огромное количество фильмов различных жанров и направлений, и иногда сложно определиться с выбором того, что посмотреть. Для решения этой проблемы можно воспользоваться мобильным приложением по фильмам, которое поможет ориентироваться в мире кино и выбирать фильмы на основе личных предпочтений и рекомендаций других пользователей. Это приложение предоставляет удобную возможность разбивать фильмы на категории, создавать список желаемого и читать рецензии на фильмы, чтобы сформировать мнение о том, стоит ли смотреть данный фильм или нет.

1 Постановка задач

Целью данного проекта является создание мобильного приложения для платформы Android для поиска информации о фильмах и сериалах и их оценивания.

Основными задачами проекта является реализация представления информации о фильмах и сериалах, а также создание площадки для обсуждения фильмов и сериалов.

Для достижения поставленных целей необходимо иметь представление о разрабатываемой системе, представленное необходимыми UML–диаграммами и разработанным дизайном приложения, как в целом, так и в отдельных сценариях, а также реализовать базы данных, которые будут хранить информацию о фильмах и сериалах, отзывы к ним.

Также, необходимо провести тестирование системы и ее аналитику.

1.1 Требования к разрабатываемой системе

1.1.1 Функциональные требования

К разрабатываемому приложению выдвинуты следующие требования:

- разделение пользователей на неавторизованных и авторизованных;
- обеспечение возможности регистрации для незарегистрированных пользователей;
- обеспечение возможности авторизации для неавторизованных пользователей;
- обеспечение возможности деавторизации для авторизованных пользователей;
- обеспечение возможности просмотра пользователями каталога фильмов и сериалов с информацией о них и отзывами к ним;
- обеспечение возможности поиска фильмов из каталога по части названия пользователями;

- обеспечение возможности отправки авторизованными пользователями отзывов к фильмам и сериалам;
- обеспечение возможности добавления пользователями фильмов и сериалов в список избранного и удаления их оттуда.

1.1.2 Технические требования

Программный продукт должен обеспечить:

- регистрацию и авторизацию пользователей с использованием адреса электронной почты и пароля;
- шифрование пароля при записи в БД;
- хранение данных в БД;
- использование механизмов защиты от SQL-инъекций;
- использование протокола передачи данных HTTP;
- трехуровневую архитектуру.

1.2 Требования к интерфейсу

Интерфейс должен быть выполнен в единой для всех экранов цветовой гамме, едином стиле. Все надписи должны быть легко читаемы, все элементы управления должны быть выполнены в едином стиле, должны выделяться на фоне содержимого экранов и хорошо заметны для пользователя.

1.3 Задачи, решаемые в процессе разработки

Перед проектом были поставлены следующие задачи:

- анализ предметной области;
- анализ аналогов;
- написание технического задания;
- описание разрабатываемой системы UML диаграммами;
- разработка БД;
- реализация ролей;
- реализация функциональных возможностей ролей;
- разработка функциональных возможностей приложения;

- создание макета дизайна и его реализация;
- реализация интерфейса;
- проведение тестирования;
- описание процесса разработки и результата.

2 Анализ предметной области

2.1 Терминология (гlossарий) предметной области

Мобильное приложение — программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для конкретной платформы (iOS, Android, Windows Phone и т. д.).

Android — это операционная система с открытым исходным кодом, созданная для мобильных устройств на основе модифицированного ядра Linux.

Android-приложение — программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для платформы Android.

Клиент — это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу.

Сервер — выделенный или специализированный компьютер для выполнения сервисного программного обеспечения.

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД).

SQL-запросы — это наборы команд для работы с реляционными базами данных.

Аутентификация — процедура проверки подлинности, например, проверка подлинности пользователя путем сравнения введенного им пароля с паролем, сохраненным в базе данных.

Авторизация — предоставление определенному лицу или группе лиц прав на выполнение определенных действий.

Фреймворк — программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

SQL-инъекция — внедрении в запрос произвольного SQL-кода, который может повредить данные, хранящиеся в БД или предоставить доступ к ним.

Пользователь – человек, который использует приложение.

Аккаунт или учетная запись — это персональная страница пользователя или личный кабинет, который создается после регистрации.

Frontend — клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

Backend — программно-аппаратная часть сервиса, отвечающая за функционирование его внутренней части.

REST — архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

API — описание взаимодействия одной компьютерной программы с другой.

Сериализация — это процесс преобразования объекта в поток байтов для сохранения или передачи в память, базу данных или файл.

JSON (JavaScript Object Notation) — это открытый стандарт формата файла и обмена данными формат, который использует удобочитаемый текст, чтобы сохранить и передать данные объекты, состоящие из пар атрибут–значение и массивов (или других сериализуемых значений).

Firebase Realtime Database – бесплатная удалённая СУБД, разрабатываемая Google. Имеется интерфейс REST API. Основными плюсами данной удалённой СУБД является отсутствие необходимости поиска собственного локального хостинга.

Firebase Auth – сервис авторизации пользователей от компании Google, предоставляющий различные способы входа в аккаунт, в частности через почту и пароль, и в дальнейшем использовать для аутентификации полученный персональный токен (далее idToken), а также предоставляющий персональные пользовательские аутентификаторы (далее uid).

Django – фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC.

Docker – программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений.

Kotlin – статически типизированный, объектно-ориентированный язык программирования, работающий поверх Java Virtual Machine и полностью совместимый с языком Java, что позволит интегрировать библиотеки, написанные на Java, в код на Kotlin.

Android SDK – универсальное средство разработки мобильных приложений для операционной системы Android.

Retrofit – это библиотека для разработки клиентской части приложений на языке Java или Kotlin, которая позволяет удобно взаимодействовать с веб-сервером через HTTP-запросы. Она предоставляет простой и эффективный способ выполнять сетевые запросы и обрабатывать полученные ответы.

SQLite – встраиваемая СУБД, подходящая для хранения локальных данных приложения.

2.2 Анализ целевой аудитории

Потенциальные пользователи: целевой аудиторией мобильного приложения по фильмам являются люди, которые увлечены кинематографом, интересуются новинками и классикой, хотят ознакомиться с рецензиями и отзывами других зрителей перед просмотром. Потенциальные пользователи могут быть различных возрастов и полов, но скорее всего, это будут люди в возрасте от 18 до 45 лет, так как именно они чаще всего смотрят фильмы.

Интересы: целевая аудитория приложения по фильмам имеет общий интерес к киноиндустрии, но в то же время может быть различных групп, например, фанаты конкретных жанров (боевики, драмы, комедии), любители киноклассики или новинок, а также те, кто следит за премиями и наградами в киноиндустрии.

Потребности: целевая аудитория ищет удобный и быстрый способ выбрать фильм для просмотра, их потребности включают в себя быстрый

доступ к информации о фильме, рейтингу и отзывам, возможность сохранять фильмы в список желаемого и получать персонализированные рекомендации на основе своих предпочтений.

Поведение: целевая аудитория часто использует мобильные приложения для поиска и ознакомления с информацией, поэтому приложение по фильмам должно иметь простой и понятный интерфейс, легкую навигацию и быстрый доступ к необходимой информации.

Опыт использования: опыт использования приложения по фильмам должен быть удобным и приятным для пользователя. Это может быть достигнуто путем создания интересного дизайна, обширной базы данных фильмов и рецензий, быстрой загрузки и персонализированных рекомендаций.

Распределение по регионам: целевая аудитория приложения по фильмам может быть распределена по различным регионам, таким как США, Европа, Азия и т.д.

2.3 Обзор аналогов

2.3.1 IMDb

IMDb – это один из наиболее популярных сайтов, посвященных фильмам и телевизионным шоу, который предоставляет информацию о фильмах, актерах, режиссерах, критиках, рейтингах, отзывах и других связанных с киноиндустрией темах. IMDb имеет большую базу данных, которая постоянно обновляется, а также приложение для мобильных устройств. Интерфейс показан на рисунке 1.

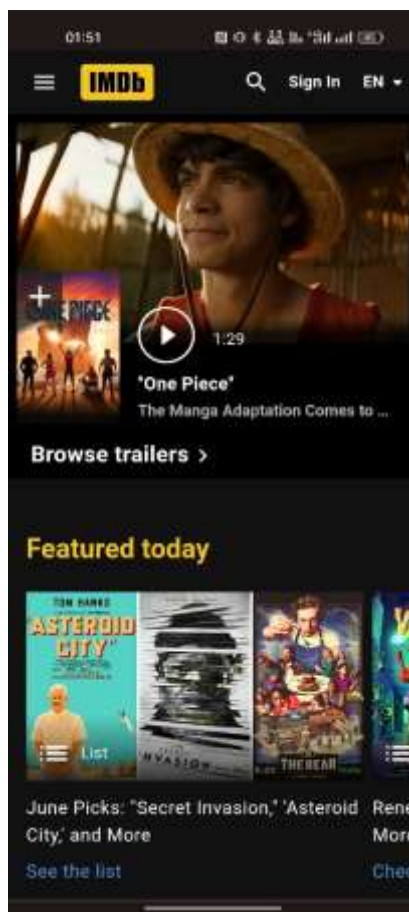


Рисунок 1 - Интерфейс IMDb

Плюсы:

- большая база данных, содержащая информацию о большом количестве фильмов и телесериалов;
- большое количество пользовательских отзывов и оценок.

Минусы:

- не всегда точная информация о фильмах;
- некоторые пользовательские отзывы и оценки могут быть необъективными;
- интерфейс может казаться перегруженным и не очень удобным для использования.
- Приложение больше не доступно на территории РФ.

2.3.2 Rotten Tomatoes

Rotten Tomatoes – это сайт, на котором представлены рецензии на фильмы и телевизионные шоу, а также их рейтинги. Рецензии и рейтинги на Rotten Tomatoes основаны на мнениях критиков и зрителей. Интерфейс показан на рисунке 2.

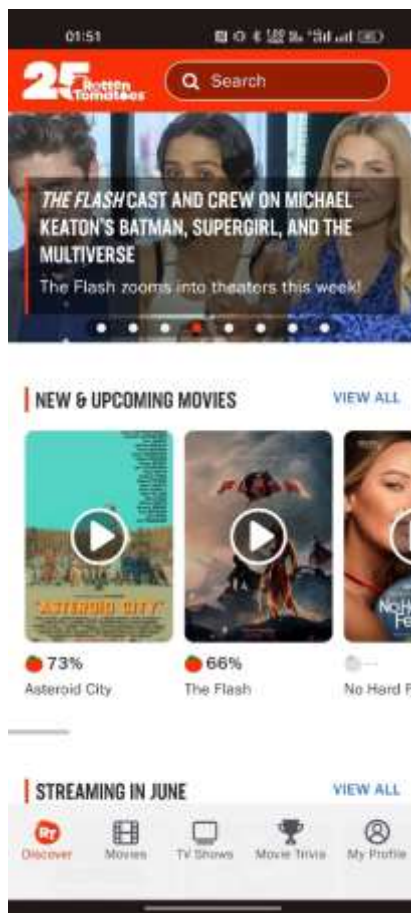


Рисунок 2 - Интерфейс Rotten Tomatoes

Плюсы:

— большое количество критиков и пользователей.

Минусы:

- рейтинги и отзывы могут быть не совсем объективными, так как могут быть влияния со стороны студий или крупных кинокомпаний;
- не все критики и зрители могут быть знакомы с вашими вкусами, так что их отзывы и оценки могут не совпадать с вашими;
- не всегда хорошо адаптирован под региональную аудиторию.

2.3.3 Kinopoisk

Kinopoisk – это крупнейший российский сайт о кино, который предоставляет информацию о фильмах, актерах, режиссерах, критиках, рейтингах, отзывах и других связанных с киноиндустрией темах. Kinopoisk также имеет свое мобильное приложение. Интерфейс показан на рисунке 3.

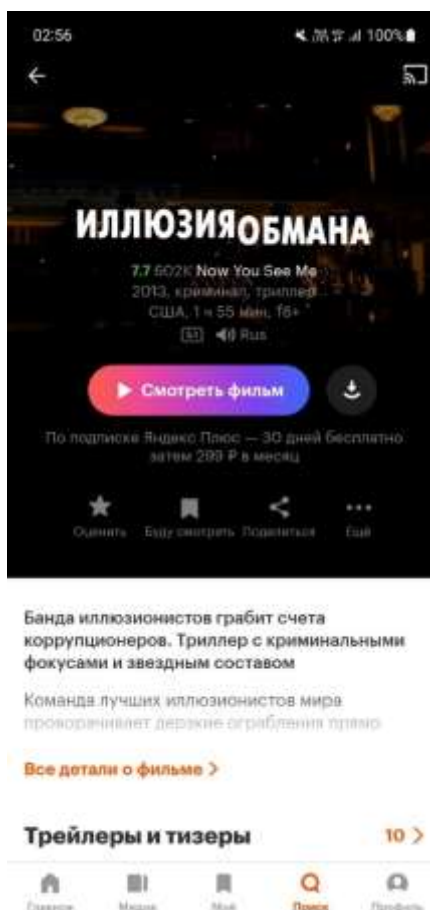


Рисунок 3 - Интерфейс приложения Kinopoisk

Минусы:

- некоторые оценки пользователей могут быть субъективными или заниженными/завышенными;
- отсутствие объективной критической оценки фильмов;
- реклама на платформе, что может быть раздражающим для некоторых пользователей;
- не всегда актуальная информация о датах выхода фильмов и сериалов.

2.4 Моделирование системы

2.4.1 Диаграмма в стиле методологии IDEF0

IDEF0 диаграмма является иерархической структурой функций, которая отображает бизнес-процесс и его взаимодействие с различными функциональностями для достижения целей организации. Ее главная задача заключается в оценке стоимости выполнения операций и задач, улучшения понимания процессов и оптимизации их для повышения эффективности организации. IDEF0 диаграмма приведена на рисунке 4.

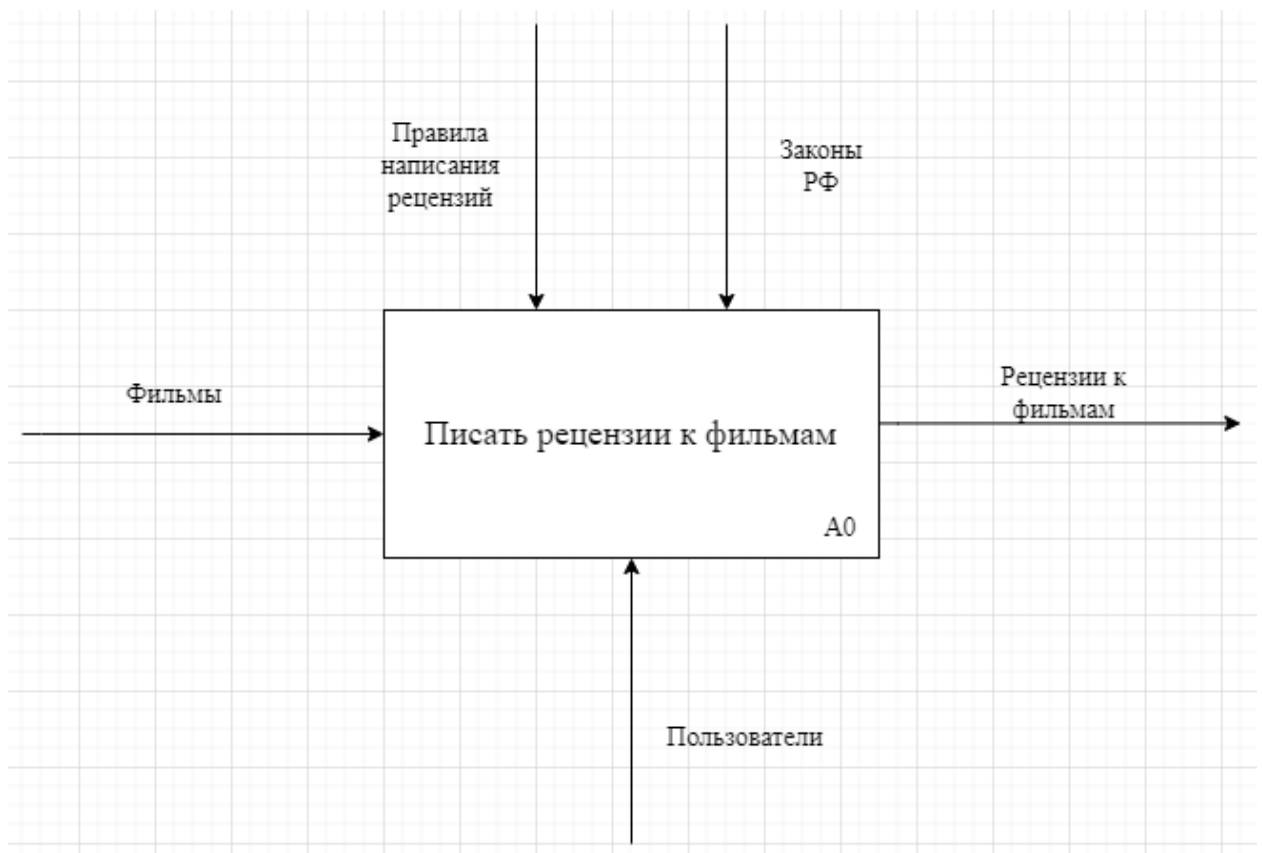


Рисунок 4 - Диаграмма IDEF0

2.4.2 Диаграмма активностей

Диаграмма активностей позволяет более детально визуализировать конкретный случай использования. Это поведенческая диаграмма, которая иллюстрирует поток деятельности через систему. Диаграмма активностей для авторизованного пользователя представлена на рисунке 5.

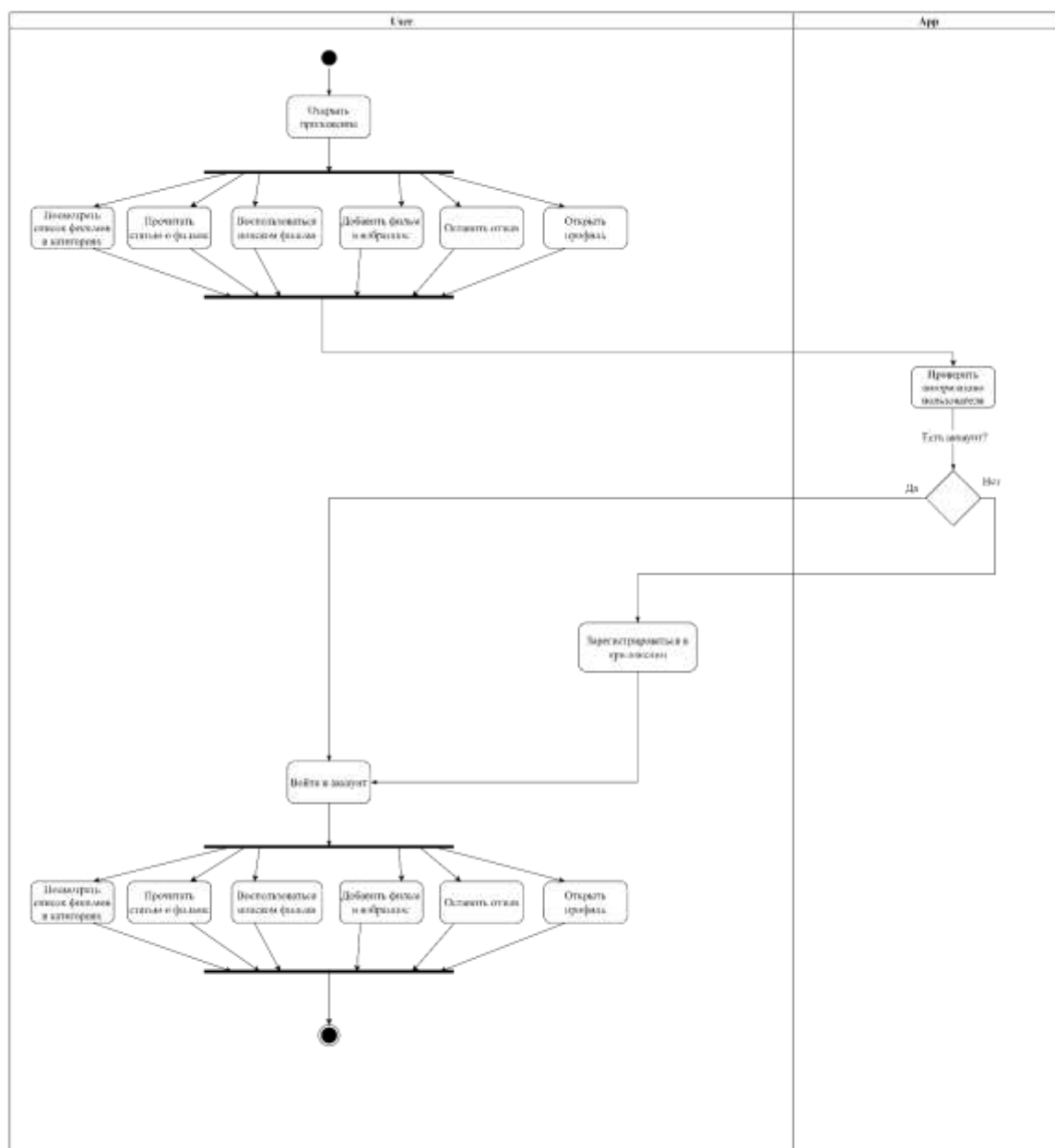


Рисунок 5 - Диаграмма активностей для пользователя

2.4.3 Диаграмма прецедентов

Диаграмма прецедентов – это диаграмма, которую используют для анализа различных систем. Она позволяет визуализировать различные типы ролей в системе и то, как эти роли взаимодействуют с системой. Диаграмма прецедентов представлена на рисунке 6.



Рисунок 6 - Диаграмма прецедентов

Каждый пользователь способен взаимодействовать с приложением. В зависимости от того, является пользователь неавторизованным или авторизованным, его возможности будут отличаться.

Неавторизованный пользователь способен:

- авторизоваться;
- просмотреть список категорий фильмов;
- просмотреть список фильмов;
- воспользоваться поиском по фильмам;
- прочитать статью о фильме;
- прочитать отзывы к фильму.

Авторизованный пользователь способен делать все то же самое, что и неавторизованный, кроме авторизации, плюс:

- оставить отзыв о фильме;

- добавить фильм в избранное;
- удалить фильм из избранного.

2.4.4 Диаграмма состояний

Диаграмма состояний позволяет описывать поведение системы и проектировать способы реакции на различные события. Диаграмма сотрудничества представлена на рисунке 7

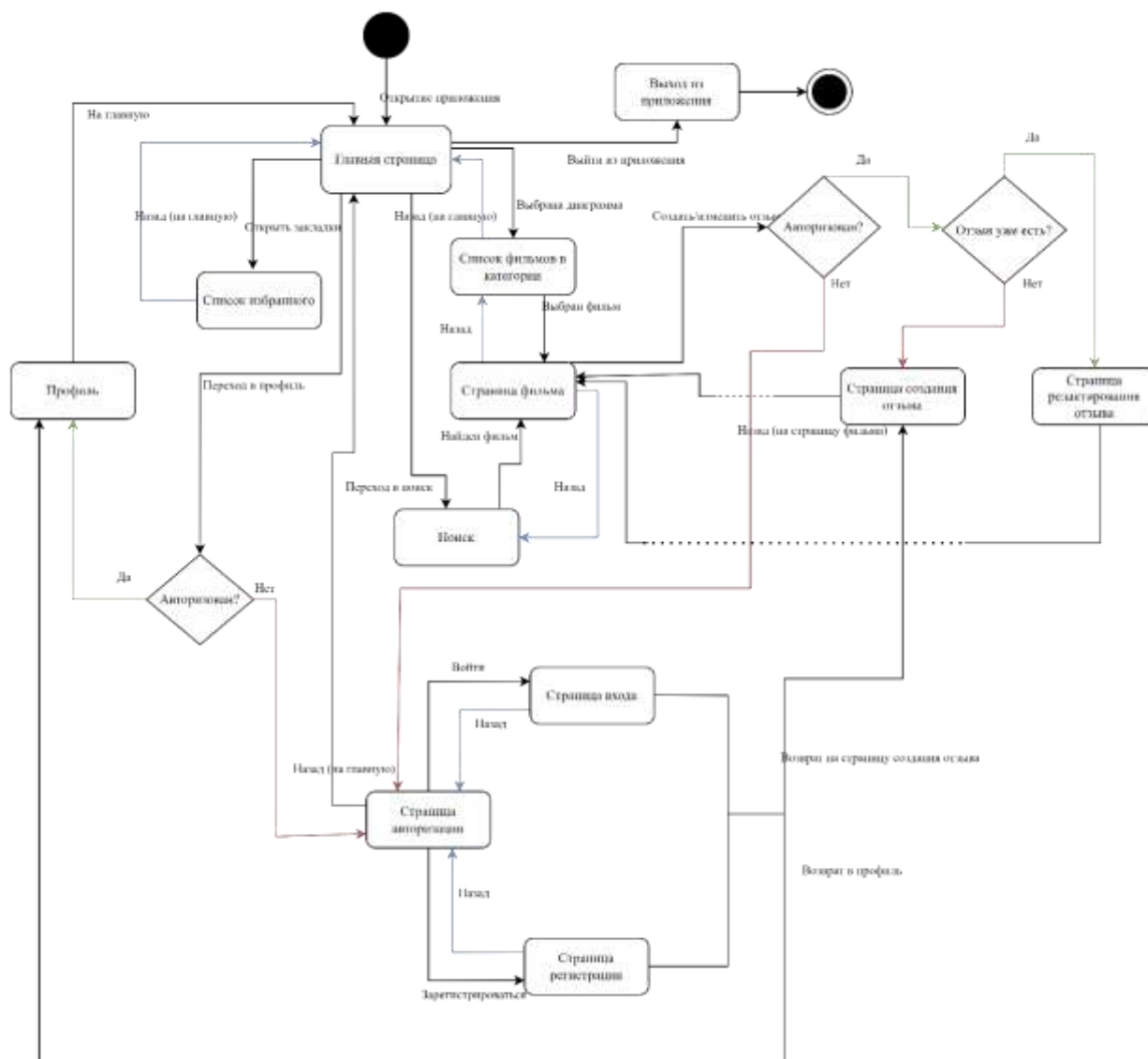


Рисунок 7 - Диаграмма состояний

2.4.5 Диаграмма последовательности

Диаграммы последовательностей используются для более детального описания логики сценариев использования. Диаграммы последовательностей

содержат объекты, которые взаимодействуют в рамках сценария, сообщения, которыми они обмениваются, и возвращаемые результаты, связанные с сообщениями. Диаграмма последовательности представлена на рисунках 8-9.

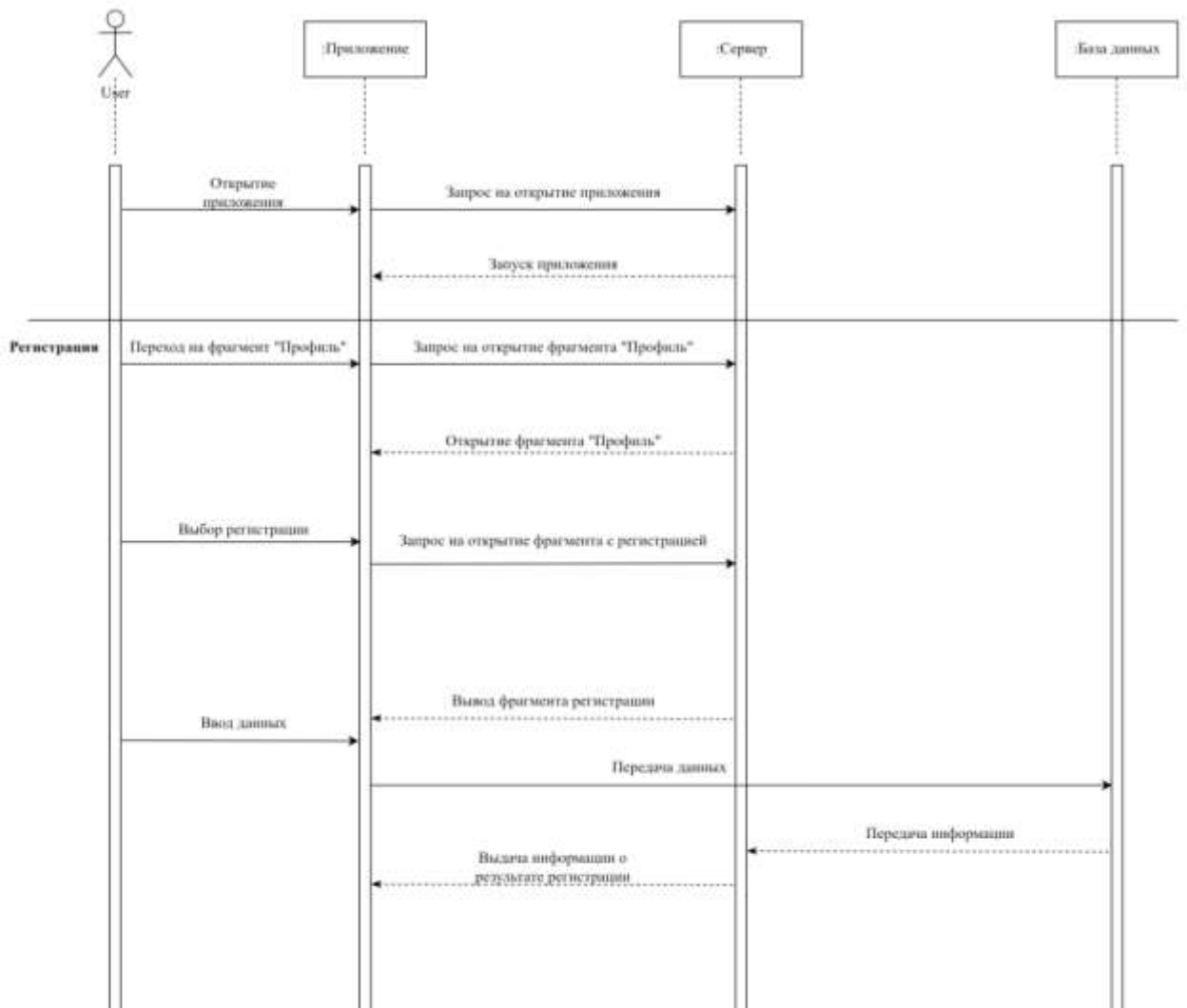


Рисунок 8 - Диаграмма последовательности

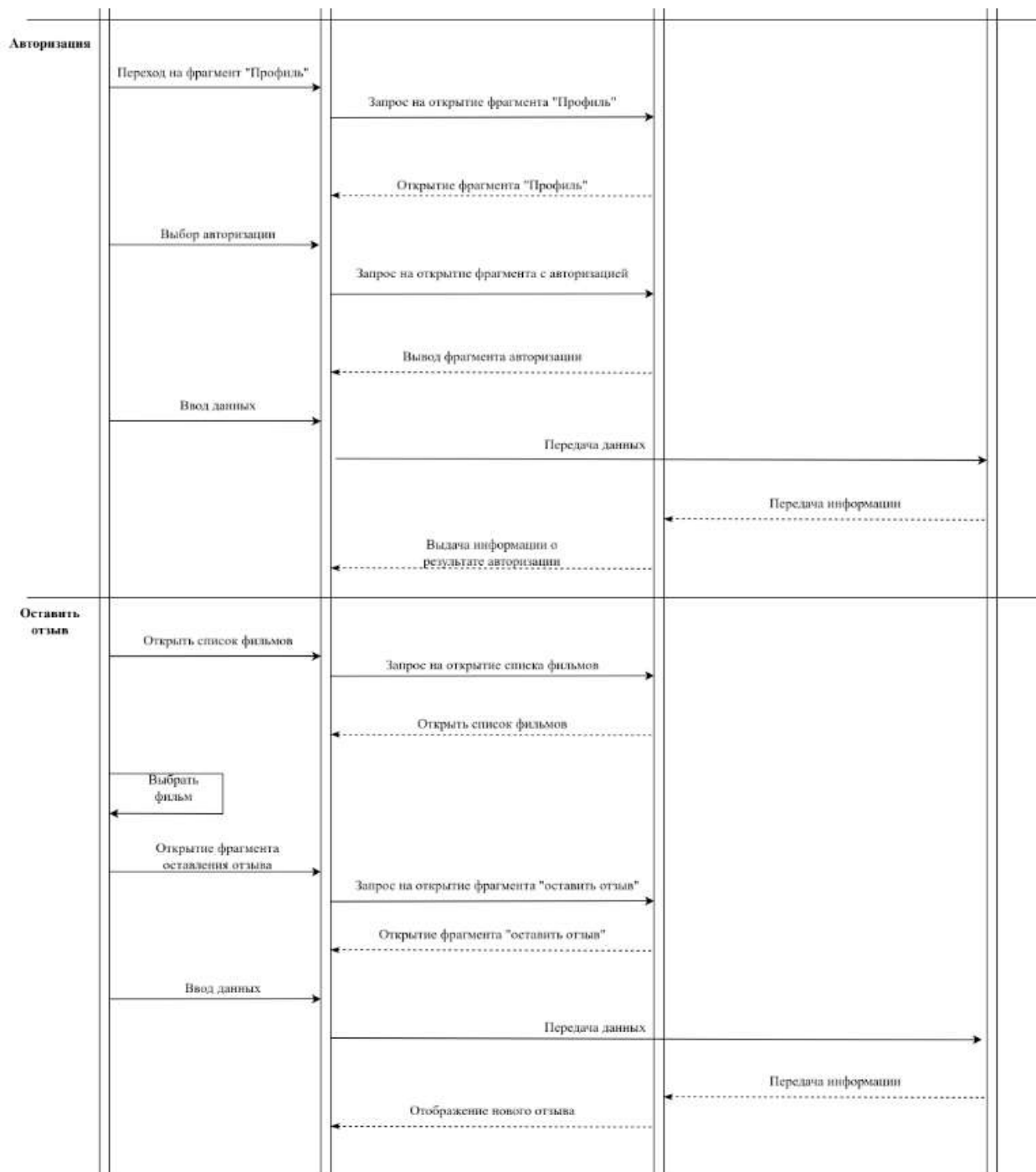


Рисунок 9 - Диаграмма последовательности

2.4.6 ER-диаграмма

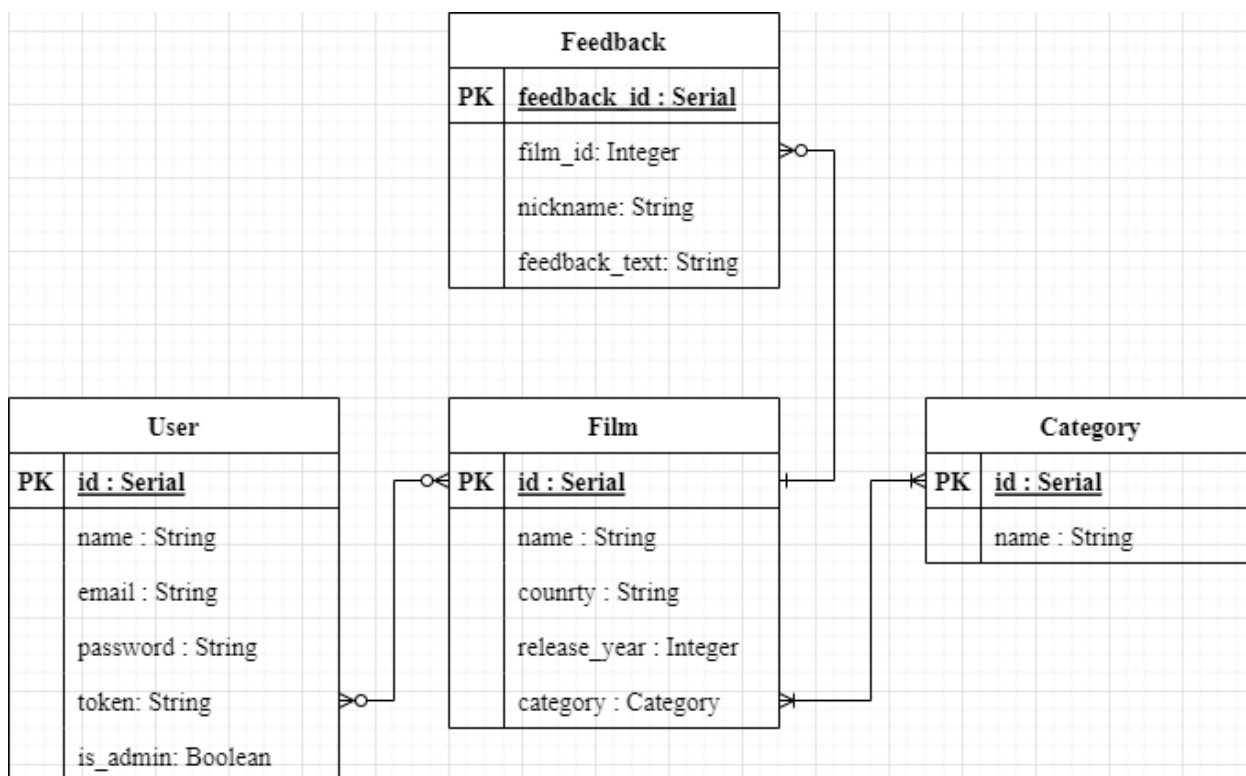


Рисунок 10 - ER-диаграмма

2.4.7 Диаграмма сотрудничества

Диаграмма сотрудничества изображает отношения и взаимодействия между программными объектами. Они используются для понимания объектной архитектуры внутри системы. Диаграммы сотрудничества представлены на рисунках 11-17



Рисунок 11 - Диаграмма сотрудничества для регистрации пользователя



Рисунок 12 - Диаграмма сотрудничества для авторизации пользователя



Рисунок 13 - Диаграмма сотрудничества для поиска фильмов

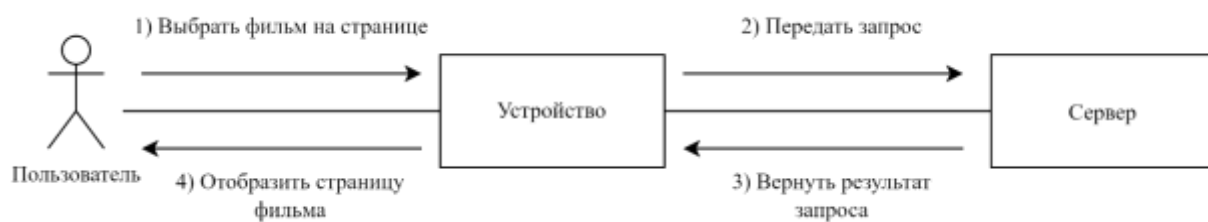


Рисунок 14 - Диаграмма сотрудничества для просмотра страницы фильма

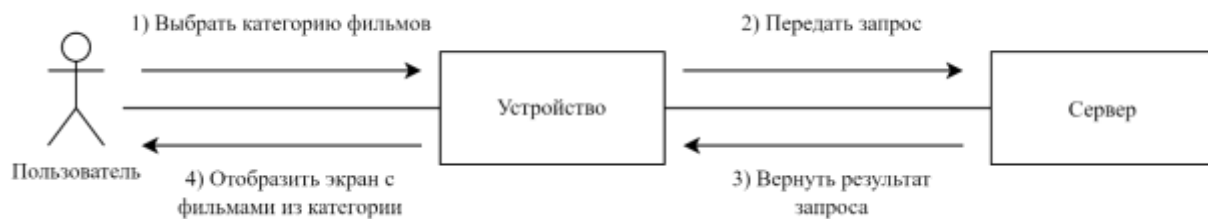


Рисунок 15 - Диаграмма сотрудничества для просмотра фильмов в категории

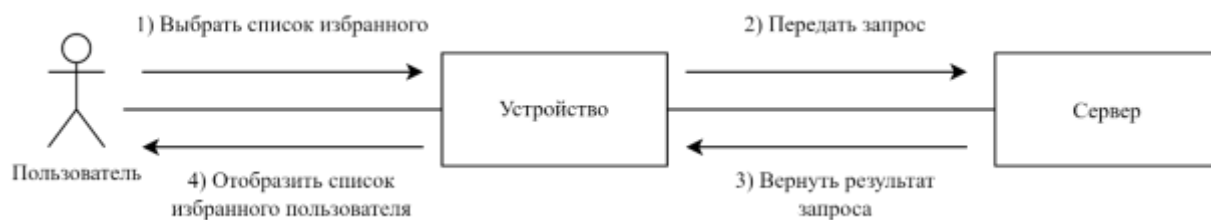


Рисунок 16 - Диаграмма сотрудничества для просмотра списка избранного пользователя



Рисунок 17 - Диаграмма сотрудничества для отправки отзыва к фильму

2.5 Аналитика мобильного приложения

В качестве системы для сбора данных об использовании приложения пользователями выступает сервис «Яндекс.Метрика», так как позволяет наиболее быстро и легко настроить метрики для мобильных приложений, обладает понятным интерфейсом и доступным руководством по использованию.

На рисунке 18 показаны данные, собранные с сервиса «Яндекс.Метрика».



Рисунок 18 - Данные сервиса «Яндекс.Метрика»

2.6 Границы проекта

Основными границами проекта являются:

- добавление в список избранного фильмов из каталога;
- оставление отзывов к фильмам, которые находятся в каталоге;
- ограниченный бюджет на разработку и продвижение проекта;
- ограниченное количество фильмов в каталоге на начальной стадии проекта;
- ограниченные сроки разработки и запуска проекта.

На фоне основных сценариев и функциональных требований, можно сформулировать то, что точно выходит за основные границы проекта:

- съемки фильмов и сериалов;
- проведение рекламных компаний к фильмам и сериалам;
- прокат фильмов и сериалов;
- продажа сопутствующей фильмам и сериалам атрибутики;
- разработка и продажа видеоигр по мотивам фильмов и сериалов.

3 Реализация

3.1 Средства реализации

Мобильное приложение имеет архитектуру, соответствующую шаблону клиент-серверного приложения и разделяется на части посредством REST API.

Данная архитектура мобильного приложения соотносится с основными требованиями к проекту, а именно:

- разделение пользователей на неавторизованных и авторизованных;
- обеспечение возможности регистрации для незарегистрированных пользователей;
- обеспечение возможности авторизации для неавторизованных пользователей;
- обеспечение возможности деавторизации для авторизованных пользователей;
- обеспечение возможности просмотра пользователями каталога фильмов и сериалов с информацией о них и отзывами к ним;
- обеспечение возможности поиска фильмов из каталога по части названия пользователями;
- обеспечение возможности отправки авторизованными пользователями отзывов к фильмам и сериалам;
- обеспечение возможности добавления пользователями фильмов и сериалов в список избранного и удаления их оттуда.

Для реализации серверной части приложения были выбраны следующие технологии:

- Firebase Realtime Database;
- Firebase Auth;
- Django;
- Docker.

Для реализации клиентской части приложения были выбраны следующие технологии:

- Kotlin;
- Android SDK;
- Retrofit;
- SQLite.

3.2 Реализация серверной части приложения

В качестве серверной части приложения используются фреймворки Django и Django REST Framework (далее – DRF). Использование Django в сочетании с DRF для создания API для мобильного приложения на Android предлагает ряд преимуществ. DRF предоставляет удобные инструменты для создания RESTful API, обеспечивая автоматическую сериализацию и десериализацию данных, обработку запросов и другие функции, необходимые для разработки API. Django и DRF также обеспечивают гибкость и масштабируемость, позволяя легко добавлять новые конечные точки API и настраивать функциональность. Безопасность обеспечивается встроенными механизмами Django и механизмами аутентификации и авторизации DRF.

Архитектура данного решения основана на шаблоне проектирования MTV (Model-Template-View) в Django. Модели описывают структуру данных и взаимодействие с базой данных. Шаблоны определяют внешний вид страниц, а представления обрабатывают запросы и управляют передачей данных между моделями и шаблонами. DRF дополняет эту архитектуру, предоставляя инструменты для создания API-представлений и сериализации данных. Такая комбинация Django и DRF обеспечивает быстрое написание API для мобильного приложения.

Также применяется программная платформа Docker для быстрой разработки, тестирования и развертывания приложений.

Вся логика разбита на отдельные пакеты, выполняющие различные функции и представляющие собой отдельные приложения MVC:

- users – приложение, отвечающее за функциональность пользователей;
- categories – приложение, отвечающее за функциональность категорий.

films – приложение, отвечающее за функциональность фильмов.

3.2.1 Приложение users

В данном приложении имеется следующая структура:

- models.py: в этом файле определяются классы модели профиля пользователя приложения. А также представлены классы, на основе которых происходит сериализация/десериализация моделей в JSON формат;
- views.py: в этом файле имеются методы, привязанные к различным функциям API;
- apps.py: данный файл хранит в себе конфигурацию для моделей текущего модуля.

В данном модуле имеется модель User, хранящая данные о пользователе.

User имеет следующие поля:

- user_id: идентификатор;
- email: адрес электронной почты;
- password: пароль;
- nickname: никнейм;
- favorite_films: список фильмов в избранном.

Данный модуль реализует для данной модели возможность регистрации и авторизации пользователя, а также возможность получения информации о нём.

3.2.2 Приложение categories

В данном приложении имеется следующая структура:

- models.py: в этом файле определяются классы модели категории фильмов в приложении. А также представлены классы, на основе которых происходит сериализация/десериализация моделей в JSON формат;

- `views.py`: в этом файле имеются методы, привязанные к различным функциям API;
- `apps.py`: данный файл хранит в себе конфигурацию для моделей текущего модуля.

В данном модуле имеется модель `Category`, хранящая данные о категории. Она имеет следующие поля:

- `category_id`: идентификатор;
- `category_name`: название категории.

Данный модуль реализует для данной модели возможность добавление и удаление категории, получения списка категорий или получение конкретной категории по `id`.

3.2.3 Приложение `films`

В данном приложении имеется следующая структура:

- `models.py`: в этом файле определяются классы модели фильма в приложении. А также представлены классы, на основе которых происходит сериализация/десериализация моделей в JSON формат;
- `views.py`: в этом файле имеются методы, привязанные к различным функциям API;
- `apps.py`: данный файл хранит в себе конфигурацию для моделей текущего модуля.

В данном модуле имеется модель `Film`, хранящая данные о категории. Она имеет следующие поля:

- `film_id`: идентификатор;
- `name`: название фильма;
- `country`: страна производства;
- `description`: описание фильма;
- `releaseDate`: дата релиза фильма;
- `categories`: категории фильма.

Данный модуль реализует для данной модели возможность добавление и удаление фильма, получения списка фильмов или получение конкретного фильма по id.

3.3 Реализация клиентской части приложения

Для реализации клиентской части приложения было принято решение воспользоваться Android SDK, Retrofit и языком программирования Kotlin. Этот выбор сочетает в себе множество преимуществ и предоставляет разработчикам широкий набор инструментов для создания современных мобильных приложений на платформе Android.

Android SDK, или набор разработчика для Android, является ключевым компонентом при разработке приложений под Android. Он обеспечивает разработчикам доступ к различным API и сервисам, а также предоставляет богатый набор инструментов для разработки пользовательского интерфейса, управления жизненным циклом приложения и обработки событий.

Kotlin – это современный язык программирования, полностью совместимый с Java и специально разработанный для платформы Android. Он обладает богатым набором функций и усовершенствований по сравнению с Java, что делает процесс разработки более эффективным и удобным.

Основным преимуществом Kotlin является его поддержка функционального программирования и расширений языка. Он позволяет разработчикам писать более компактный и читаемый код, сокращая количество шаблонного кода, необходимого в Java. Благодаря таким возможностям как null-безопасность и инференция типов, Kotlin помогает предотвратить множество ошибок на этапе компиляции и повысить общую надежность приложения.

Retrofit облегчает создание клиентской части приложения, которая взаимодействует с сервером, путем предоставления абстракции над HTTP-протоколом. Он позволяет определить интерфейс, описывающий конечные точки (endpoints) и операции (методы) для взаимодействия с сервером. Затем

Retrofit генерирует реализацию этого интерфейса, которая выполняет необходимые HTTP-запросы и обрабатывает полученные ответы.

Retrofit поддерживает различные форматы сериализации данных, такие как JSON, XML и другие. Вы можете указать формат, используя соответствующий конвертер, например, Gson для работы с JSON. Он также предоставляет возможность добавления заголовков, обработки ошибок и других функций, которые облегчают работу с сетевыми запросами.

В целом, Retrofit является мощным и удобным инструментом для работы с сетевыми запросами в приложениях на Java или Kotlin, который позволяет сократить объем кода и упростить взаимодействие с сервером.

3.3.1 Форма для просмотра категорий фильмов

Пользователь (авторизованный и неавторизованный) имеет возможность просматривать категории фильмов. Для этого пользователь должен нажать на кнопку «Главное» в панели навигации. Форма данного экрана представлена на рисунке 19.



Рисунок 19 - Список категорий в приложении

3.3.2 Форма для просмотра списка фильмов в категории

Пользователь (авторизованный и неавторизованный) имеет возможность просматривать список фильмов в категории.

Для этого пользователь должен выбрать категорию из списка, после чего на экране отобразится список всех фильмов этой категории. Нажав на конкретный фильм, пользователь сможет просмотреть подробную информацию о нем и при желании добавить в избранное или оставить отзыв. Форма данного экрана представлена на рисунке 20.

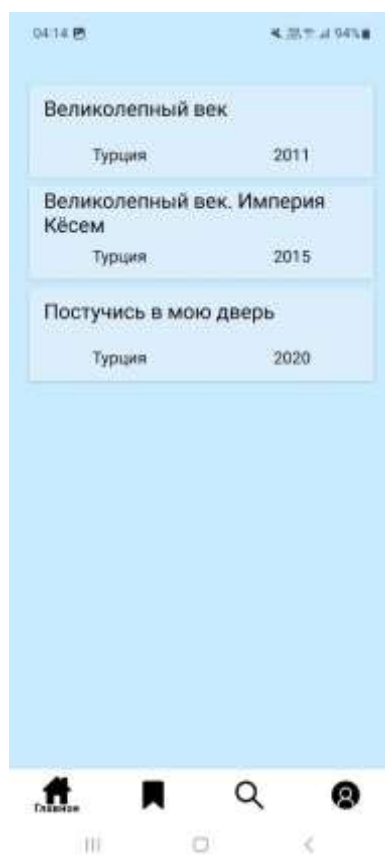


Рисунок 20 - Список фильмов в категории

3.3.3 Форма для просмотра статьи о фильме

Пользователь (авторизованный и неавторизованный) имеет возможность просматривать статью о фильме.

Для этого пользователь должен нажать на кнопку фильма из списка категорий или поиска, после чего на экране отобразится статья и отзывы на этот фильм. На экране также будет кнопка, где пользователь может оставить

свою оценку и отзыв на данное кино. Если пользователь не авторизован, то система попросит его войти в аккаунт или зарегистрироваться. Также пользователь имеет возможность добавить фильм в избранное, если он авторизован. Иначе система попросит пользователя войти в аккаунт. Форма данного экрана представлена на рисунке 21.



Рисунок 21 - Страница фильма из категории

3.3.4 Форма для просмотра списка избранного

Пользователь (авторизованный) имеет возможность добавить к себе в избранное фильм, который он желает просмотреть. Для того, чтобы посмотреть список фильма, желаемых к просмотру, пользователь должен нажать на кнопку «Избранное» в панели навигации. Форма данного экрана представлена на рисунке 22.

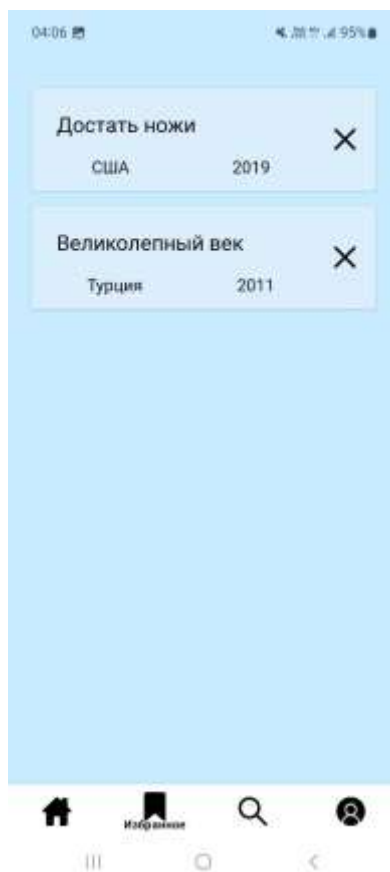


Рисунок 22 - Список избранных фильмов

3.3.5 Форма для просмотра поиска

Пользователь (авторизованный и неавторизованный) имеет возможность воспользоваться поиском фильма по его названию. Для этого нужно нажать на кнопку «Поиск» в панели навигации. На экране появится строка поиска, где пользователь может ввести название фильма и из поиска перейти на нужную ему статью. Форма данного экрана представлена на рисунке 23.

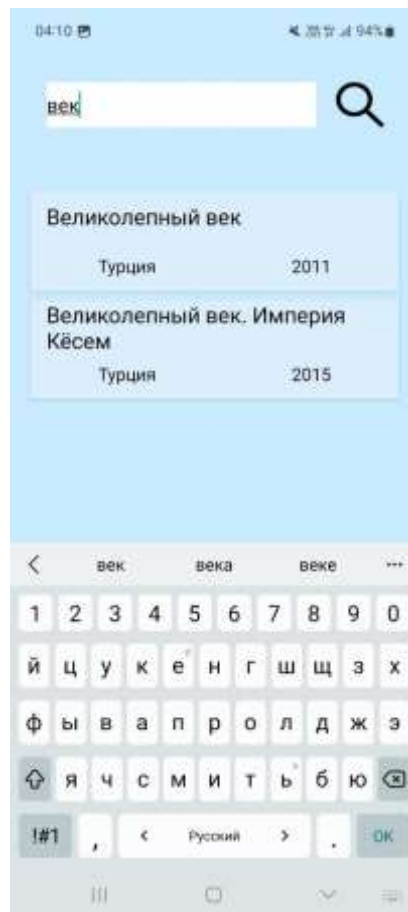


Рисунок 23 - Страница поиска

3.3.6 Форма для просмотра профиля

Пользователь (авторизованный) имеет возможность просматривать и информацию о себе, а также выйти из своего аккаунта. Для этого нужно перейти в личный кабинет с помощью кнопки «Профиль» в панели навигации и нажать на кнопку. Форма данного экрана представлена на рисунке 24.



Рисунок 24 - Просмотр профиля в приложении

3.3.7 Форма добавления отзыва к фильму

Пользователь (авторизованный) имеет возможность добавить отзыв к фильму нажав на соответствующую кнопку на странице фильма, после чего текстовый отзыв. Форма данного экрана представлена на рисунке 25.

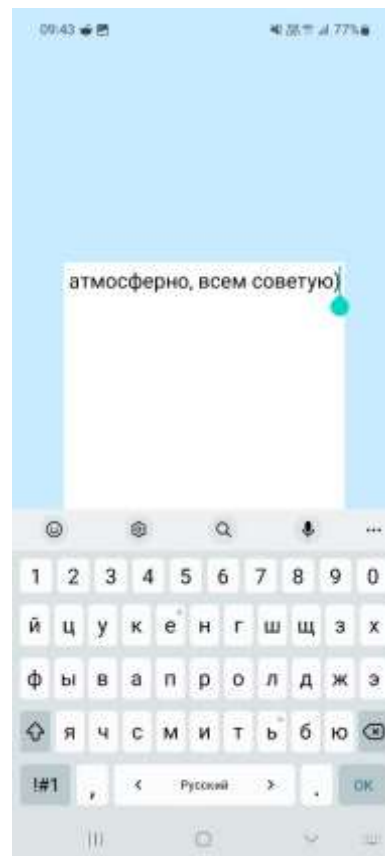


Рисунок 25 - Добавление отзыва

Заключение

При выполнении этой работы был проведён обзор существующих аналогичных приложений и сайтов, позволяющих взаимодействовать с фильмами, после чего были выявлены их основные недостатки. На их основе был создан перечень задач и требований к разрабатываемому приложению, позволяющий сделать приложение, лишённое данных минусов.

Для разработки приложения были разработаны макеты интерфейса, выбрана платформа приложения, построены UML диаграммы. Для контроля версий был создан репозиторий GitHub.

При разработке приложения были выполнены следующие задачи:

- просмотр статей о фильмах;
- добавление статей фильмов в избранное;
- добавление новых отзывов;
- поиск фильмов в приложении.

Backend часть приложения и база данных были размещены в контейнере Docker compose на хостинге. Разработанное приложение удовлетворяет поставленным требованиям. Все поставленные задачи были выполнены.

Список использованных источников

1. Android Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides) / K. Marsicano, B. Gardner, B. Phillips, C Stewart. – New York: Big Nerd Ranch Guides, 2019. – 1036 с.
2. Android Application Development Cookbook: 93 Recipes for Building Winning Apps / L. Wei-Meng. – New York: Wrox, 2013. – 408 с.
3. Android Documentation. – URL: <https://developer.android.com/docs/> (Дата обращения 01.06.2023). – Текст: электронный.
4. Django Documentation. – URL: <https://docs.djangoproject.com/en/4.2/> (Дата обращения 01.06.2023). – Текст: электронный.
5. Firebase Documentation. – URL: <https://firebase.google.com/docs> (Дата обращения 01.06.2023). – Текст: электронный.