

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Мобильное приложение по фильмам

Курсовая работа

Направление 09.03.02 «Информационные системы и технологии»

Зав. Кафедрой _____ С. Д. Махортов, д. ф.-м. н., профессор

Обучающийся _____ А.А. Петров, 3 курс, д/о

Обучающийся _____ П.В. Волченко, 3 курс, д/о

Руководитель _____ В.С. Тарасов, ст. преподаватель

Руководитель _____ И.В. Клейменов, ассистент

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Постановка задачи.....	5
1.1 Цели создания приложения	5
1.2 Требования к разрабатываемой системе	5
2 Анализ предметной области	6
2.1 Терминология	6
2.2 Анализ целевой аудитории	8
2.3 Обзор аналогов	9
2.3.1 IMDb.....	9
2.3.2 Rotten Tomatoes	10
2.3.3 Kinopoisk.....	11
2.4 Моделирование системы.....	11
2.4.1 Диаграмма IDEF0	11
2.4.2 Диаграмма активности	12
2.4.3 Диаграмма прецедентов	14
2.4.4 Диаграмма развёртывания	16
2.4.5 Диаграмма состояний.....	17
2.4.6 Диаграмма сотрудничества.....	18
2.4.7 Диаграмма последовательности	21
3 Реализация	23
3.1 Средства реализации	23
3.1.1 Технологический стек backend-разработки	23
3.1.2 Технологический стек frontend-разработки	23
3.1.3 Дополнительный инструментарий	24
3.2 Реализация базы данных	25

3.3 Реализация клиентской части	25
3.3.1 Форма для просмотра категорий фильмов.....	26
3.3.2 Форма для просмотра списка фильмов в категории.....	27
3.3.3 Форма для просмотра статьи о фильме.....	28
3.3.4 Форма для просмотра списка избранного	30
3.3.5 Форма для просмотра поиска	31
3.3.6 Форма для просмотра профиля	32
3.3.7 Форма добавления отзыва к фильму	36
3.3.8 Удаление отзыва	37
3.3.9 Удаление фильмов из категории	38
3.3.10 Добавление фильма в категорию.....	39
3.3.11 Редактирование фильма	40
3.4 Серверная часть	42
3.4.1 Приложение users	43
3.4.2 Приложение categories	45
3.4.3 Приложение films	45
ЗАКЛЮЧЕНИЕ	47
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	48

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 задачи

1.1 Цели создания приложения

Разработать мобильное приложение по фильмам, которое позволяет пользователям находить и просматривать информацию о фильмах, создавать список желаемого, оставлять отзывы и рейтинги.

1.2 Требования к разрабатываемой системе

- Обеспечение авторизации и аутентификации пользователей;
- использование механизмов защиты от SQL-инъекций;
- использование протокола передачи данных HTTP;
- приложение должно быть построено на трехуровневой архитектуре.

2 Анализ предметной области

2.1 Терминология

- 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).

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.

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.

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. Rotten Tomatoes также имеет свое мобильное приложение.

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.

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 IDEF0

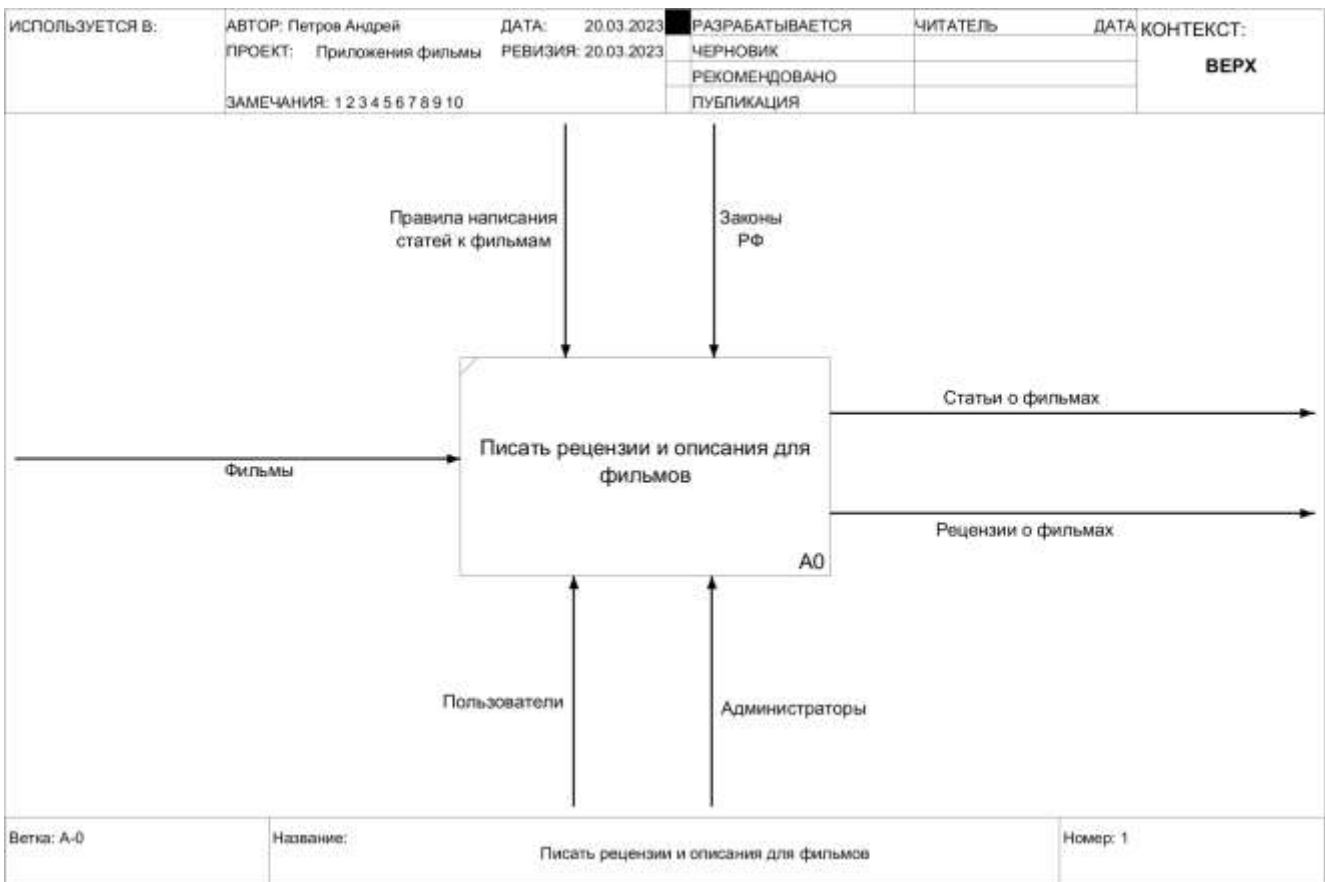


Рисунок 1. - Диаграмма IDEF0

2.4.2 Диаграмма активности

Диаграмма активности для обычного пользователя и для администратора выглядят следующим образом:

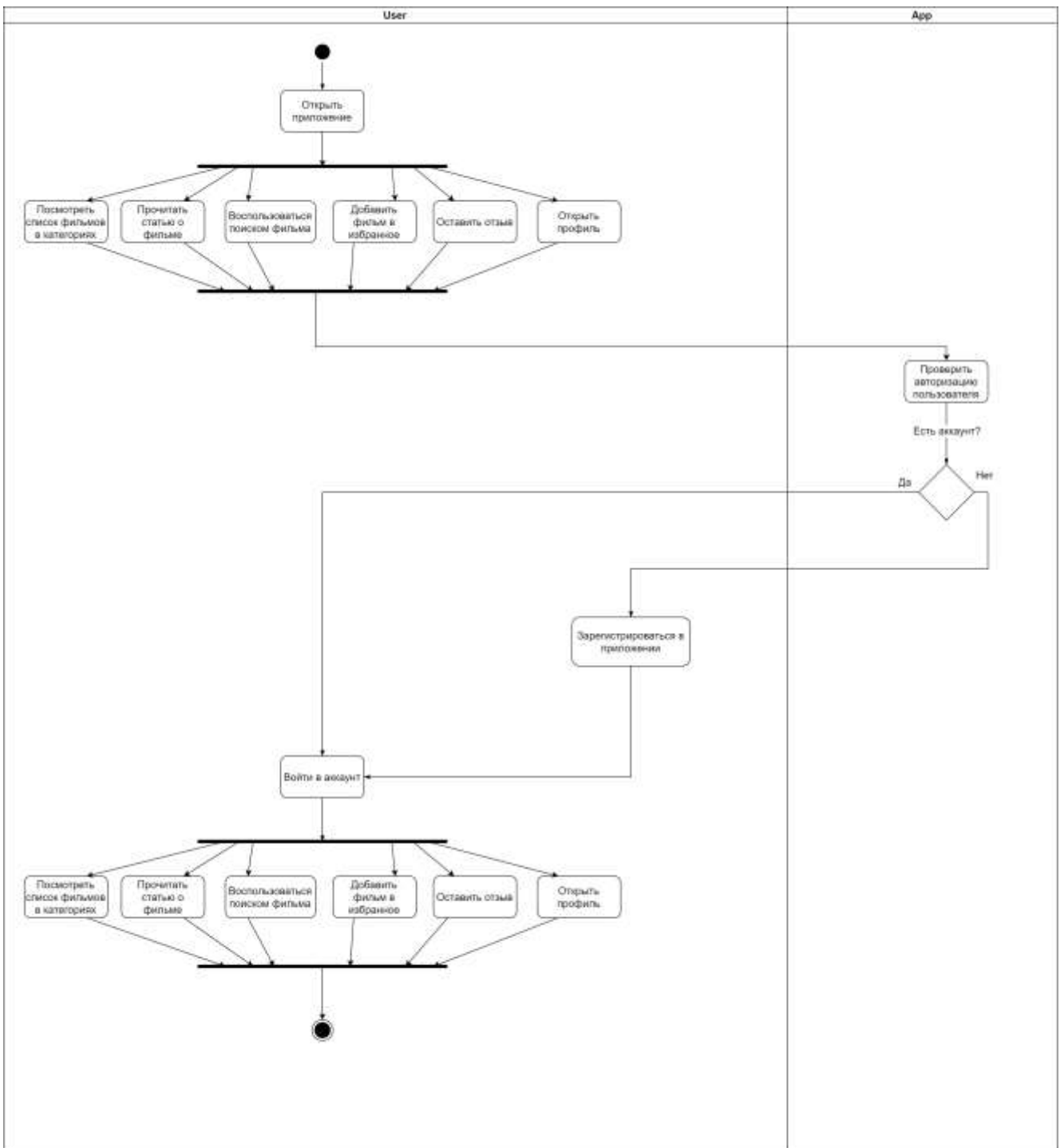


Рисунок 2. - Диаграмма активности для пользователя

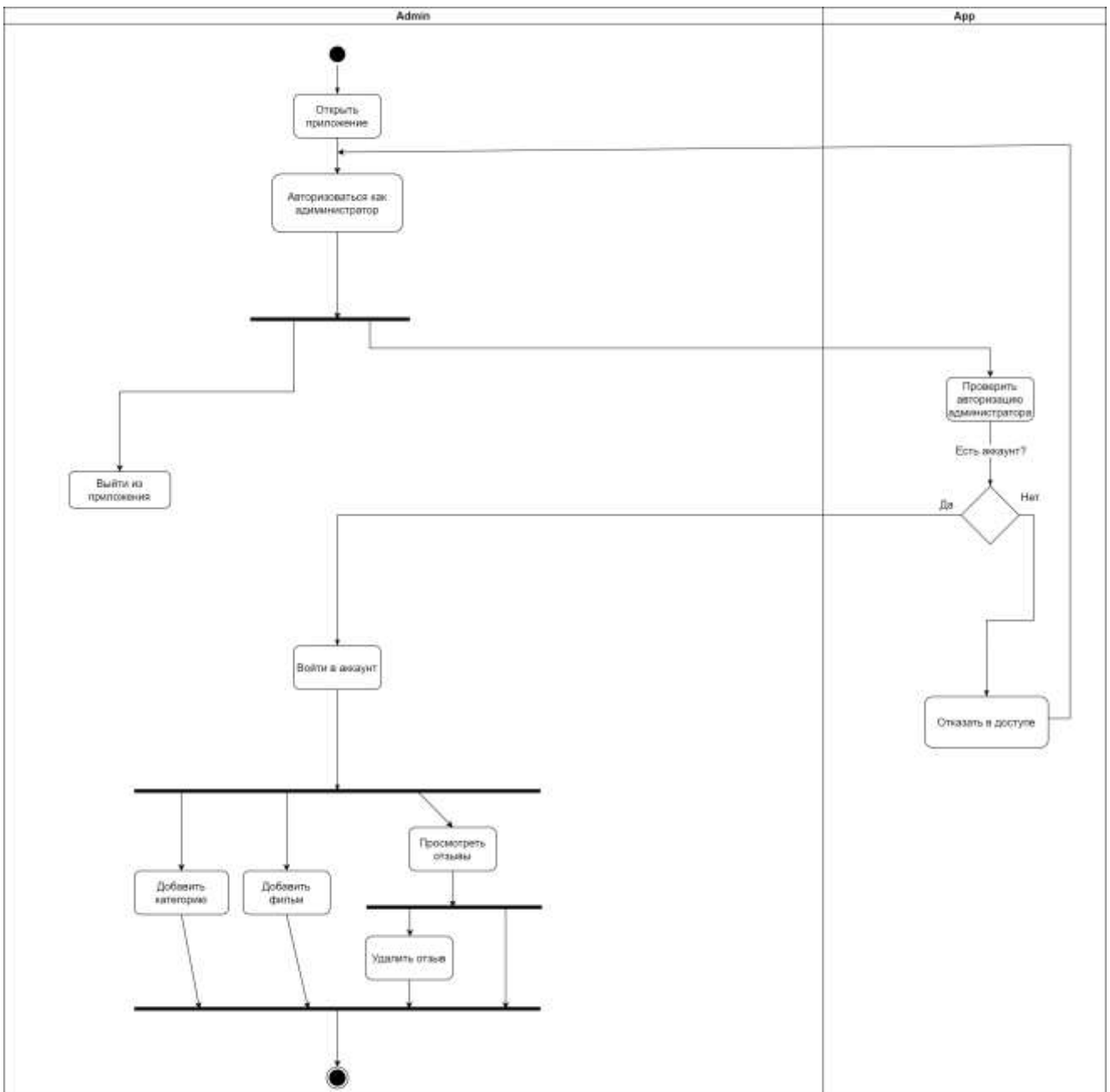


Рисунок 3. - Диаграмма активности для администратора

2.4.3 Диаграмма прецедентов

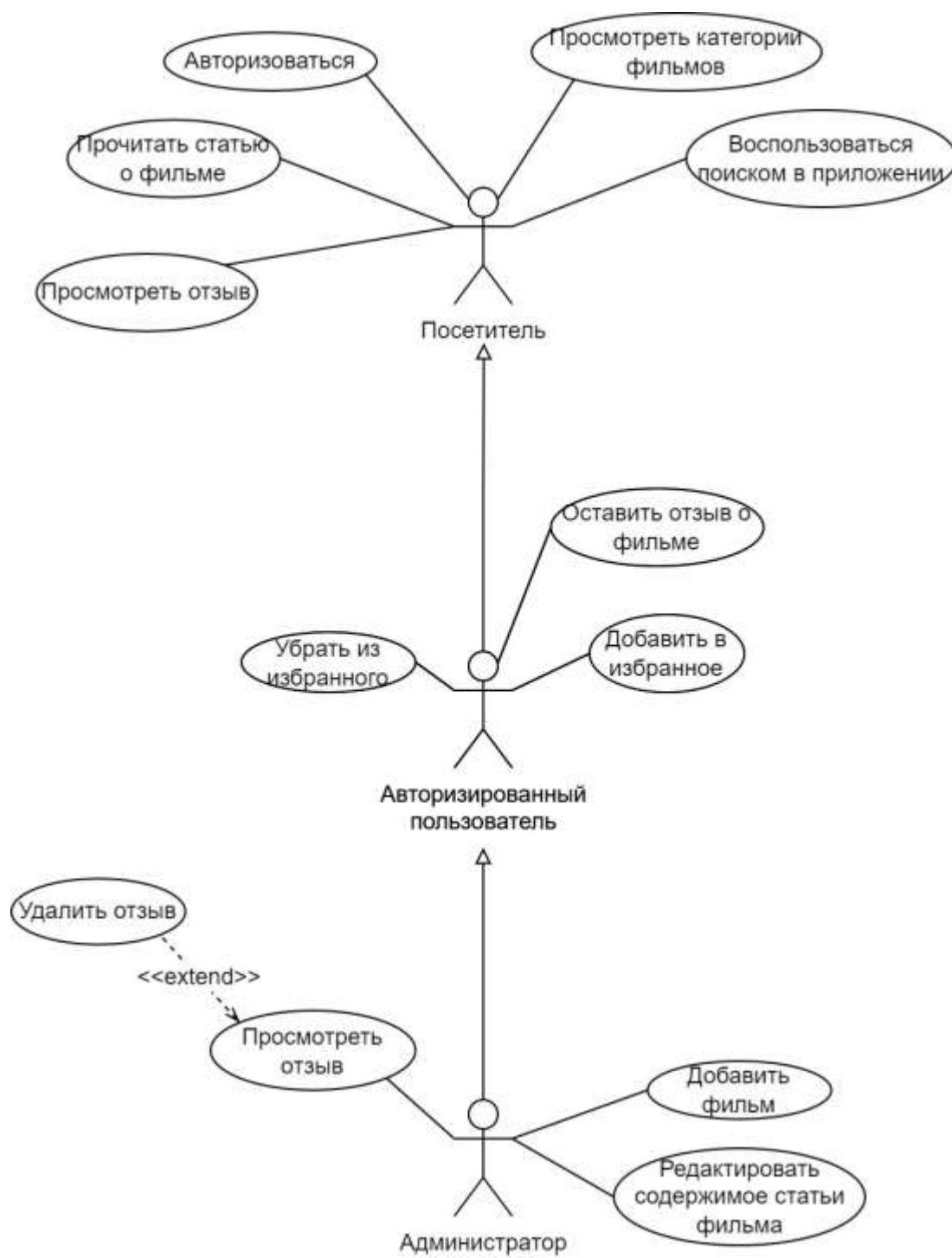


Рисунок 4. - Диаграмма прецедентов

Каждый пользователь способен взаимодействовать с приложением. В зависимости от того, является пользователь неавторизованным, авторизованным, или администратором, 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 всё то же самое, что и неавторизованный, кроме авторизации, плюс:

- Оставить отзыв о фильме
- Добавить фильм в избранное
- Удалить фильм из избранного

Администратор способен совершать все вышеперечисленные действия, также:

- Добавить фильм
- Редактировать содержимое статьи фильма
- Удалить отзыв

2.4.4 Диаграмма развёртывания

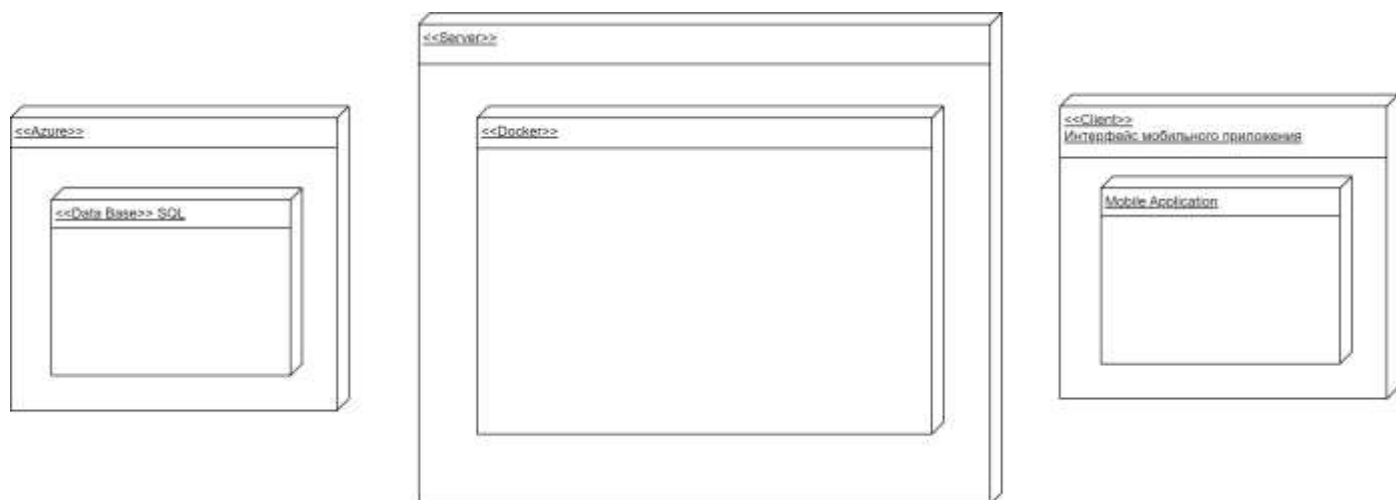


Рисунок 5. - Диаграмма развертывания

2.4.5 Диаграмма состояний

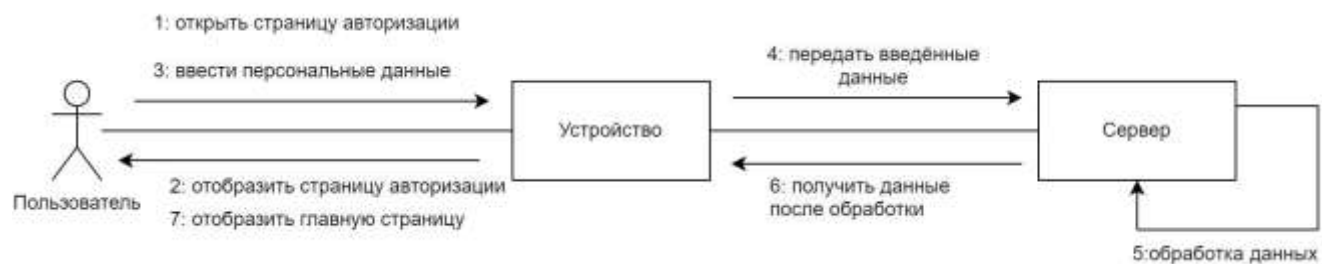


Рисунок 7. - Авторизация пользователя

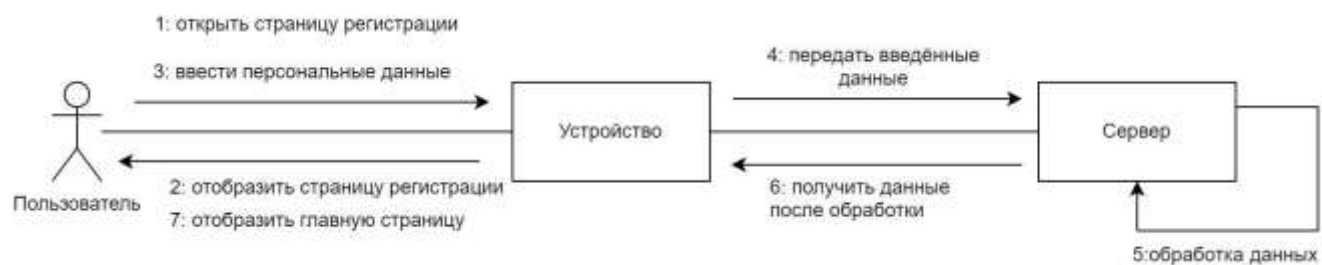


Рисунок 8. - Регистрация пользователя

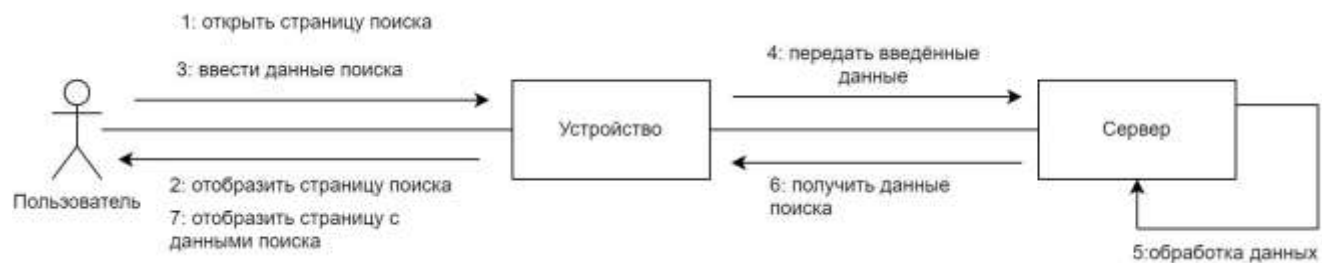


Рисунок 9. - Поиск фильмов



Рисунок 10. - Просмотр страницы фильма



Рисунок 11. - Просмотр фильмов, находящихся в категории



Рисунок 12. - Просмотр избранного

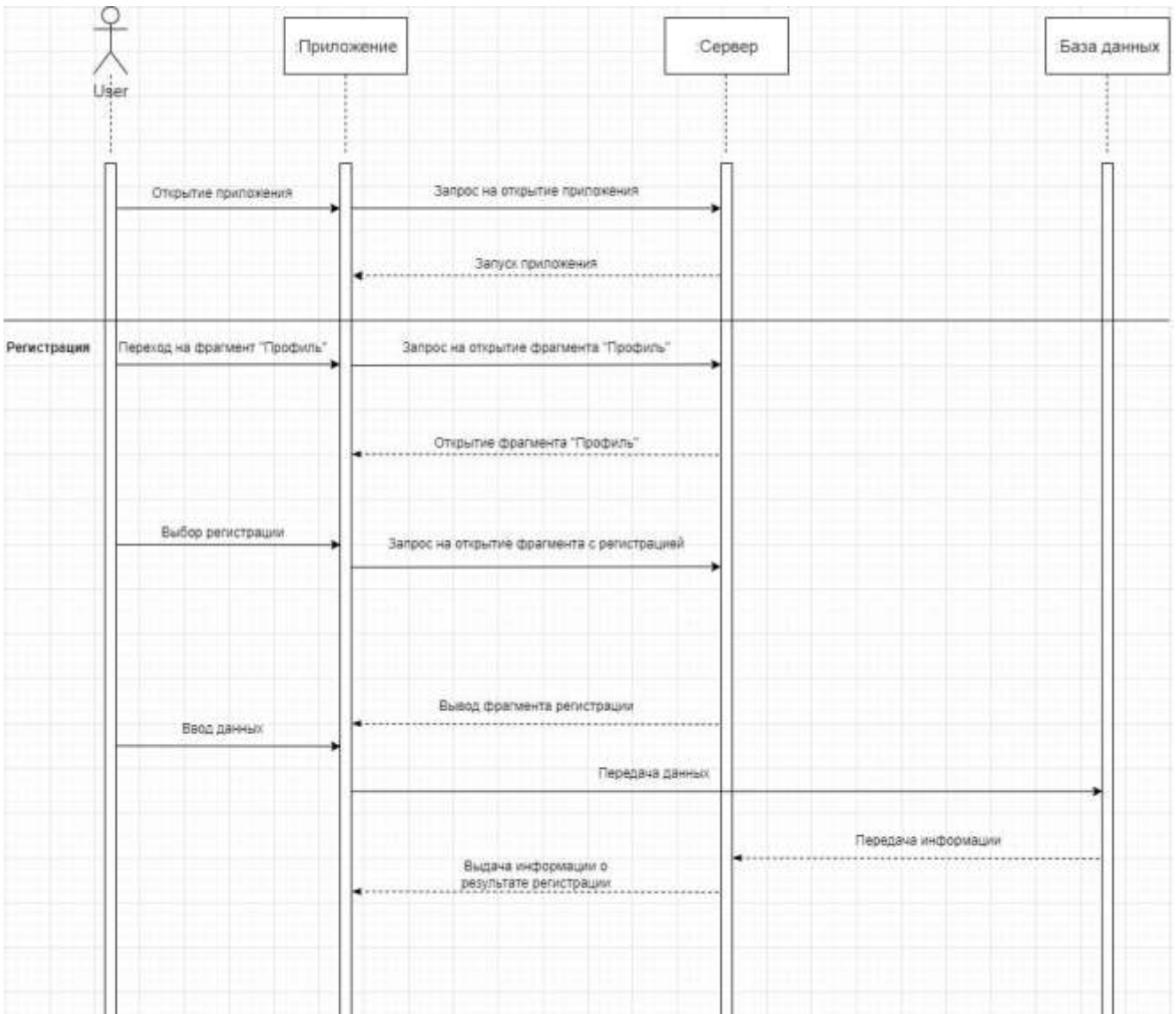


Рисунок 13. - Создание отзыва



Рисунок 14. - Настройки профиля

2.4.7 Диаграмма последовательности



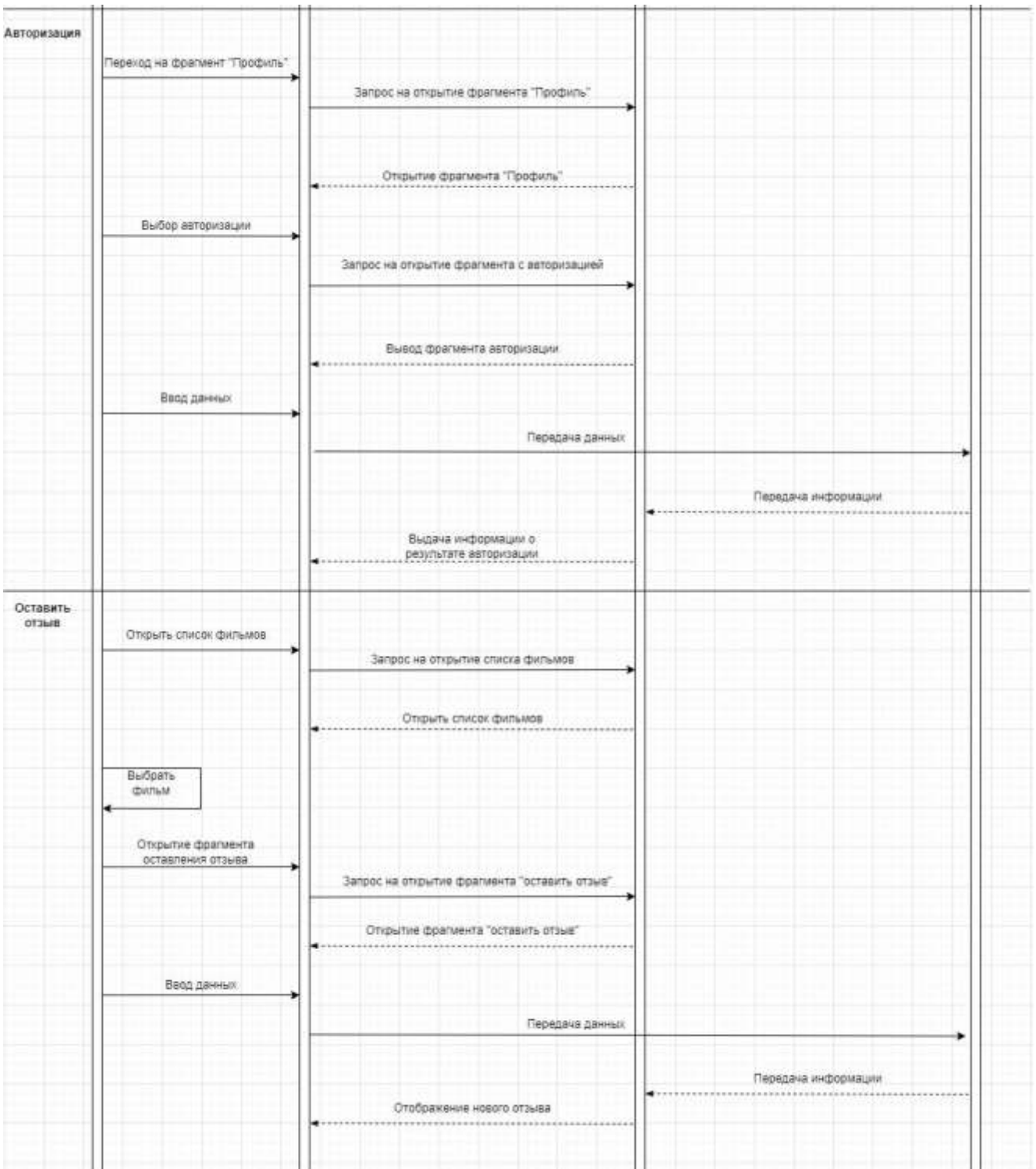


Рисунок 15. - Диаграмма последовательности

3 Реализация

3.1 Средства реализации

3.1.1 Технологический стек backend-разработки

- 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.

3.1.2 Технологический стек frontend-разработки

- 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.

3.1.3 Дополнительный инструментарий

- Trello – визуальный инструмент, обеспечивающий эффективность командной работы на любом проекте;
- Miro – платформа для совместной работы команды, виртуальная доска;
- Github – веб-сервис для хостинга IT-проектов и их совместной разработки, основанный на Git;
- Git – распределённая система управления версиями;
- Swagger – фреймворк для спецификации REST API, автоматически описывающий API на основе его кода.

3.2 Реализация базы данных

Для хранения данных была выбрана СУБД Firebase Realtime Database, хранящая данные в формате JSON. Далее приведена ER-диаграмма используемой базы данных:

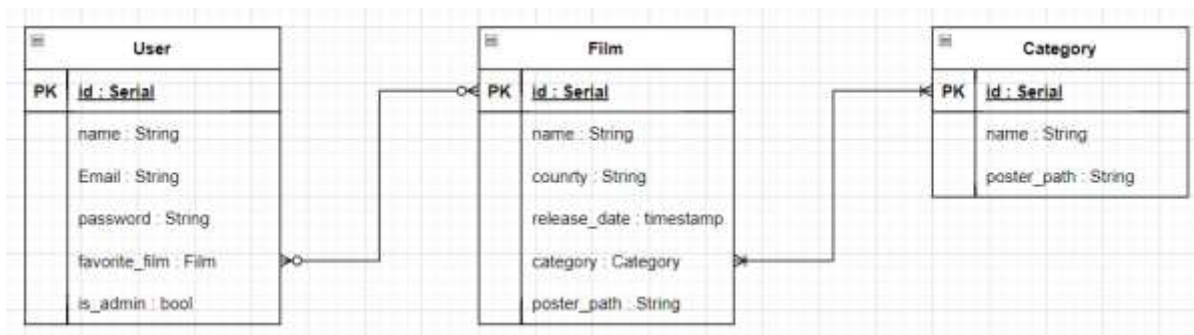


Рисунок 16. - ER-диаграмма базы данных

3.3 Реализация клиентской части

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.

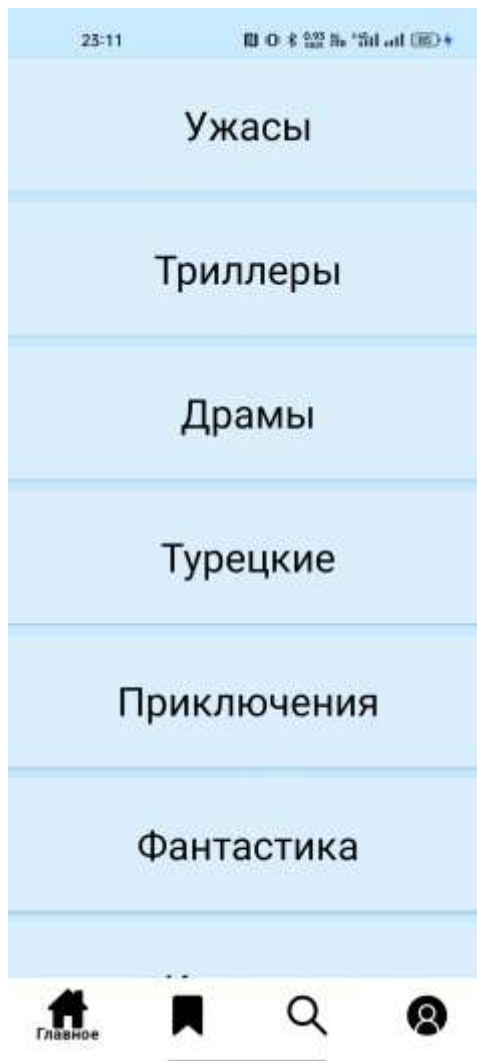


Рисунок 17. - Список категорий в приложении

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.



Рисунок 18. - 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.



Рисунок 19. - 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.

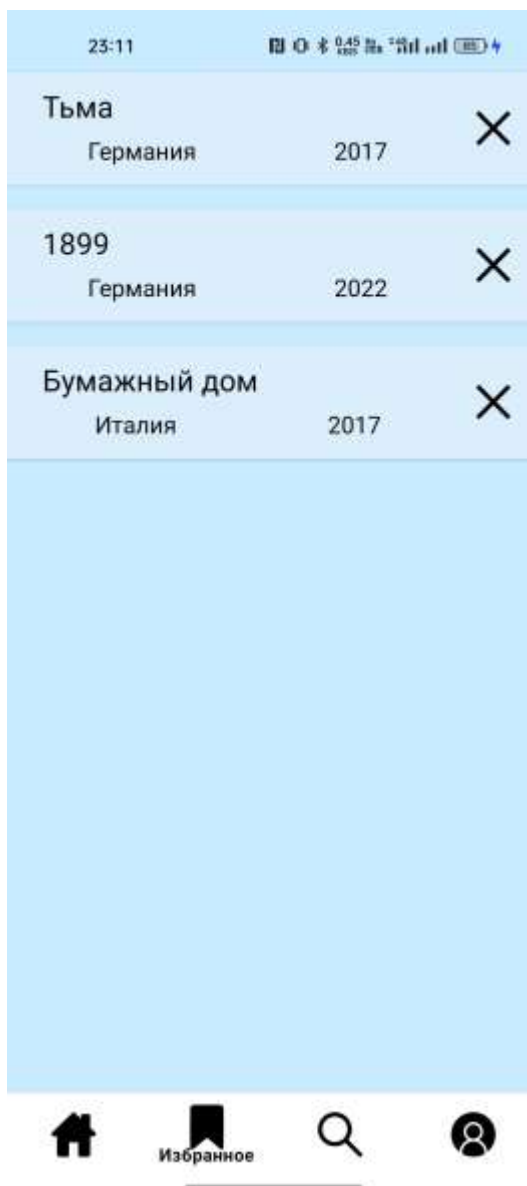


Рисунок 20. - Список избранных фильмов

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.



Рисунок 21. - Страница поиска

3.3.6 Форма для просмотра профиля

Пользователь (авторизованный) имеет возможность просматривать и редактировать информацию о себе или сменить пароль, а также выйти из своего

аккаунта. Для этого нужно перейти в личный кабинет с помощью кнопки «Профиль» в панели навигации и нажать на кнопку. Чтобы изменить данные профиля необходимо нажать кнопку «Настройки» и выбрать, что конкретно хочет отредактировать пользователь.



Рисунок 22. - Просмотр профиля в приложении

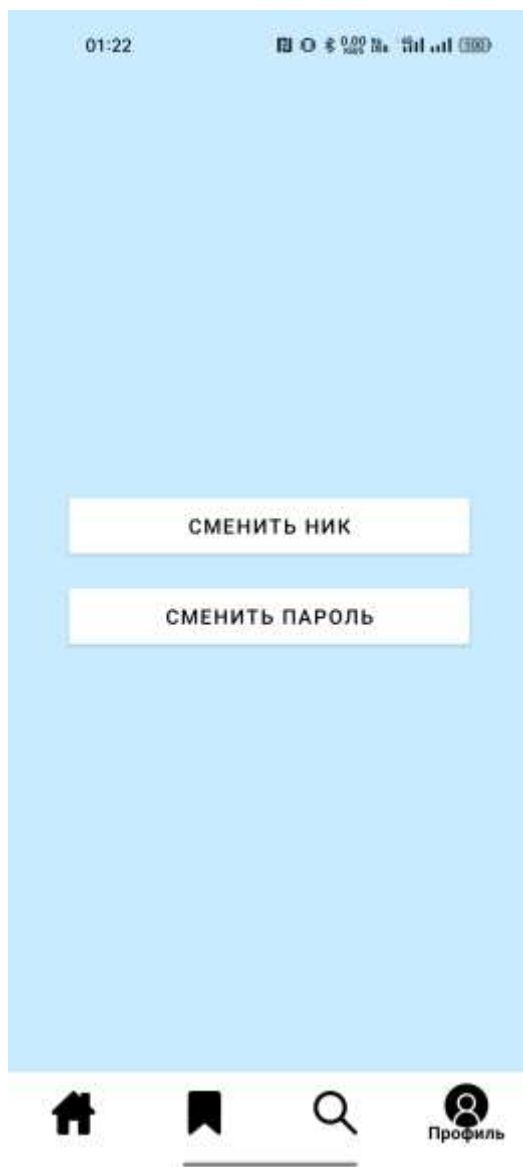


Рисунок 23. - Настройки профиля



Рисунок 24. - Смена ника

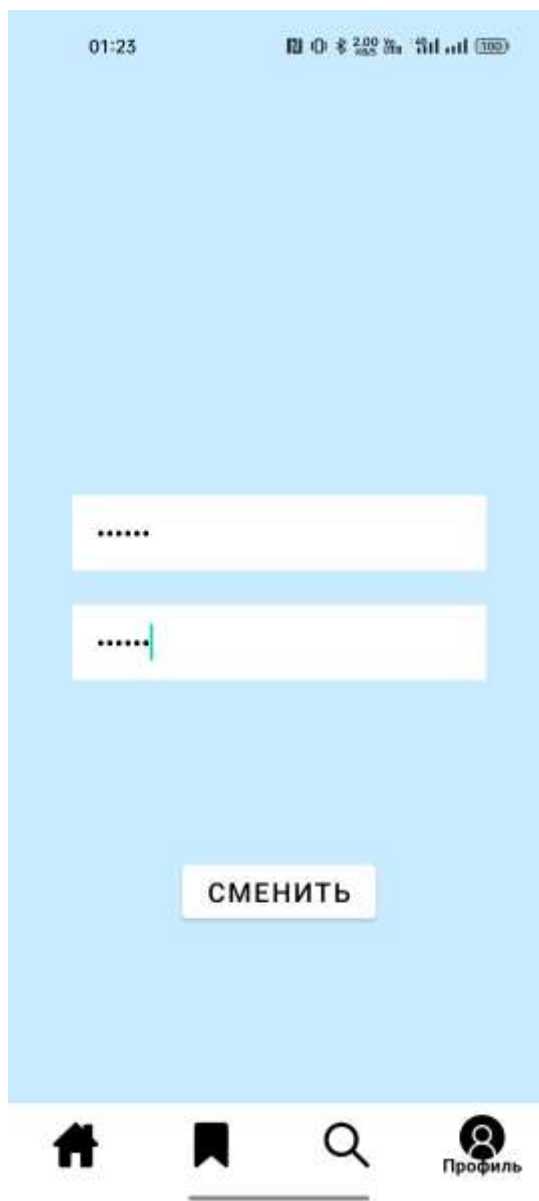


Рисунок 25. - Смена пароля

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.

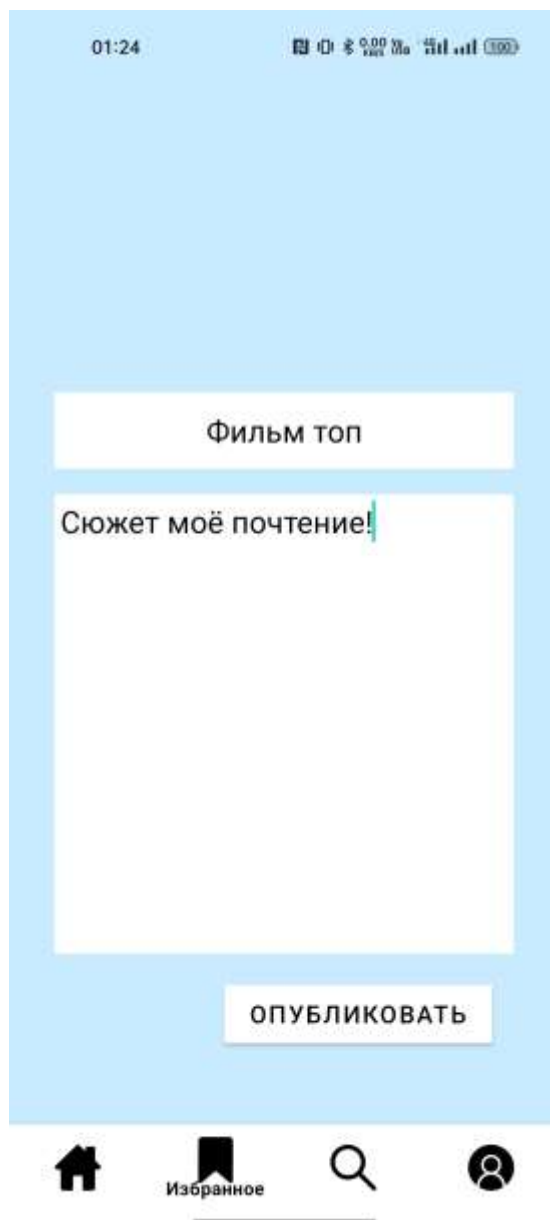


Рисунок 26. - Добавление отзыва

3.3.8 Удаление отзыва

Пользователь (администратор) имеет возможность удалить отзыв пользователя к фильму, нажав на соответствующую кнопку рядом.



Рисунок 27. - Удаление отзыва к фильму

3.3.9 Удаление фильмов из категории

Пользователь (администратор) имеет возможность удалить фильм из категории, нажав на соответствующую кнопку в странице категории.

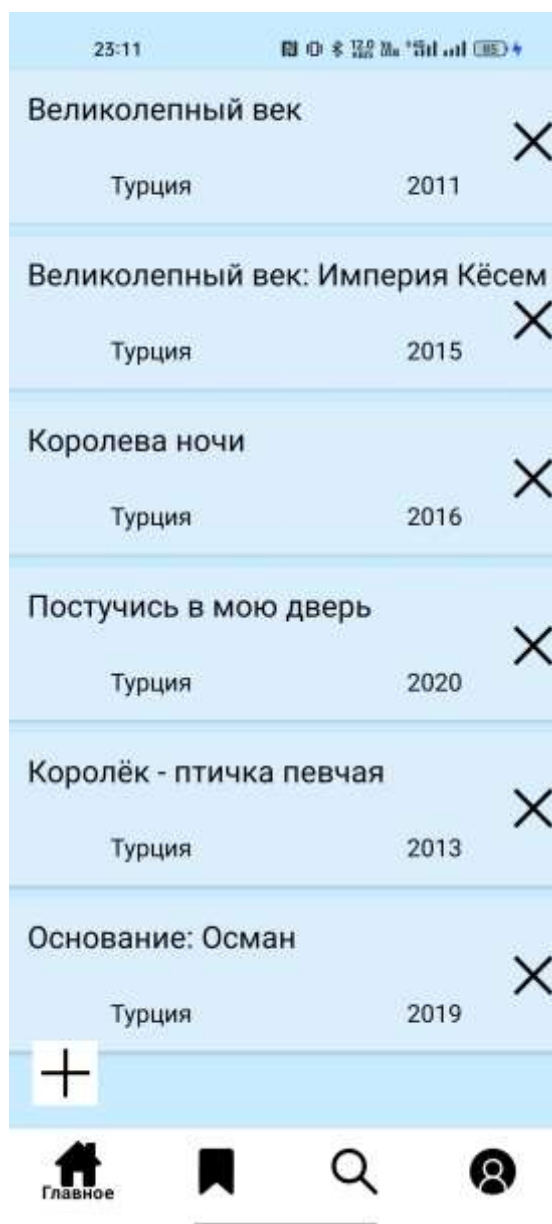


Рисунок 28. - Удаление фильмов из категории

3.3.10 Добавление фильма в категорию

Пользователь (администратор) имеет возможность добавить фильм в категорию, нажав на соответствующую кнопку в странице категории.

01:23

Название фильма

Заголовок фильма

Описание фильма

Загрузить обложку фильма

Добавить

Избранное

Рисунок 29. - Добавление фильма

3.3.11 Редактирование фильма

Пользователь (администратор) имеет возможность редактировать информацию о фильме, нажав на соответствующую кнопку на странице фильма.



Рисунок 30. - Страница фильма от лица администратора



Рисунок 31. - Редактирование фильма

3.4 Серверная часть

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.

3.4.1 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;
- `is_admin`: статус администратора.

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

3.4.2 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`.

3.4.3 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`.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данного курсового проекта был выполнен анализ предметной области и аналогов разрабатываемого приложения.

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

При разработке приложения были реализованы следующие задачи:

- Реализован список категорий фильмов;
- Реализованы списки фильмов в категории;
- Реализована возможность сохранить в избранное;
- Реализована возможность написать отзыв;
- Реализован поиск фильмов;
- Реализованы возможности добавления, редактирования и удаления фильмов администратором приложения;
- Реализовано удаление отзывов администратором приложения.

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.