

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

Факультет компьютерных наук

Кафедра обработки изображений и машинного обучения

Система учета просмотра сериалов «Максимально Сериально»

Курсовой проект

09.03.02 Информационные системы и технологии

Обработка изображений и машинное обучение

Допущен к защите

Обучающийся _____ *А.С. Зверева, 3 курс, д/о*

Обучающийся _____ *В.С. Попков, 3 курс, д/о*

Воронеж 2020

Оглавление

Оглавление	2
Введение.....	3
1. Постановка задачи	4
2. Глоссарий	4
4. Анализ предметной области	6
4.1. Анализ существующих решений	6
5. Анализ задачи.....	8
5.1. Варианты использования системы	8
5.2. Общая структура системы	11
5.3. Конфигурация системы.....	12
5.4. Взаимодействие компонентов системы.....	12
5.6. Взаимодействие в системе	14
5.7. Варианты состояния системы	15
5.8. Действия с системой	17
5.9. Развертывание системы	18
5.10. IDEF0	19
5.11. ER-диаграмма	19
6. Анализ средств реализации	20
7. Метрика	20
8. Реализация.....	21
9. Тестирование.....	21
Заключение.....	21

Введение

В настоящее время киноиндустрия выпускает огромное количество различных сериалов ежегодно. Держать в голове все просматриваемые сериалы и просмотренные серии затруднительно, учитывая количество иной важной информации, которую необходимо помнить. Можно вести записи на бумаге, но это не всегда удобно, так как блокнот или тетрадь не всегда доступны. Доступность информации в различные моменты времени с различных устройств упрощает процесс слежения за сериалами становится преимуществом перед контролем на бумаге.

Сайт должен облегчать ведение учета просмотра сериалов. Основная функциональность сайта должна включать в себя:

- добавление сериала в личный список сериалов
- отметка о просмотре серии
- комментирование и оценивание сериала
- просмотр даты выхода серии

Данный групповой проект посвящен разработке такой системы: простой для восприятия человеком, но в то же время выполняющей все необходимые функции.

1. Постановка задачи

Данный проект является информационным интернет-сервисом о сериалах.

Данная система автоматизирует процесс слежения за выходом серий сериалов и потраченных часов за просмотром сериалов.

Система предназначена для:

- упрощения процесса слежения за выходом серий сериалов
- за мониторингом статистики просмотренных сериалов
- получения информации о сериалах, включающую название, год выпуска, краткое описание, и их рейтингах

Для достижения данной цели были выделены следующие подзадачи:

1. Разработка Front-end части сервиса;
2. Разработка Back-end части сервиса;
3. Создание связи между Front-end и Back-end частями приложения;
4. Разработка базы данных.

2. Глоссарий

Личный кабинет - это раздел сервиса, в котором Пользователь может получить доступ к своим данным.

MVC (Model-View-Controller) - схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

Model (база данных) - этот компонент отвечает за данные в проекте, а также реагирует на команды контроллера, изменяя свое состояние.

Представление данных пользователю, Графический интерфейс (View) - это компонент, отвечающий за взаимодействие с пользователем. То есть данный компонент определяет внешний вид приложения и способы его использования.

Контроллер (Controller) - это компонент, который управляет запросами пользователя. Его основная функция — вызывать и координировать действие необходимых ресурсов и объектов, нужных для выполнения действий, задаваемых пользователем. Обычно контроллер вызывает соответствующую модель для задачи и выбирает подходящий вид.

front-end - клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

back-end - программно-аппаратная часть сервиса.

REST API - это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов.

GitHub - крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

Веб-сервис, интернет-сервис, система, веб-приложение, проект - идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя.

Гость - неавторизованный на веб-сервисе человек, пользующийся ограниченным функционалом веб-сервиса.

Пользователь - авторизованный на портале человек, пользующийся функционалом веб-сервиса.

Администратор - человек, имеющий доступ к расширенному функционалу веб-сервиса.

Сериял – ознакомительная информация, содержащая название, дату выхода, страну производства и жанр.

Карточка сериала – веб-страница, содержащая ознакомительную информацию.

4. Анализ предметной области

4.1. Анализ существующих решений

1. Кинопоиск

Является самым популярным интернет-сервисом о кино и сериалах в России (по версии <https://ru.ratrating.com>). Содержит большую базу фильмов и сериалов с подробным описанием. Но не имеется возможность отметки каждой просмотренной серии у сериалов. Существует также онлайн-кинотеатр Кинопоиск HD, но и в нем есть трудности с отметкой просмотра серии. Сервис доступен только по платной подписке, серии можно отметить, только просмотрев ее на сервисе.

При посещении главной страницы сайта сразу заметен крупный рекламный баннер, отвлекающий от просмотра контента (рис.1).

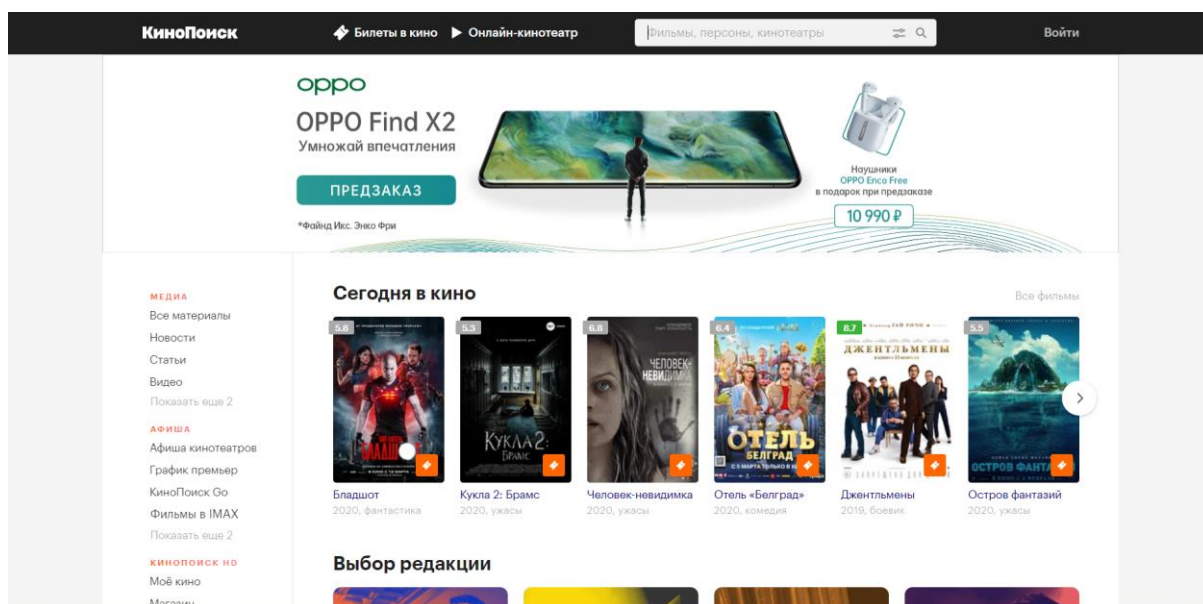


Рисунок 1. Домашняя страница сервиса Кинопоиск

2. Netflix

Американская компания, владеющая не только интернет-сервисом по просмотру сериалов, но и выпускающая их. В базу данных входят фильмы и сериалы не только собственного производства, но и других компаний-производителей фильмов.

Доступ к просмотру информации о фильмах и сериалах также предоставляется только по платной подписке, отметка о просмотре серии только после просмотра ее на сервисе.

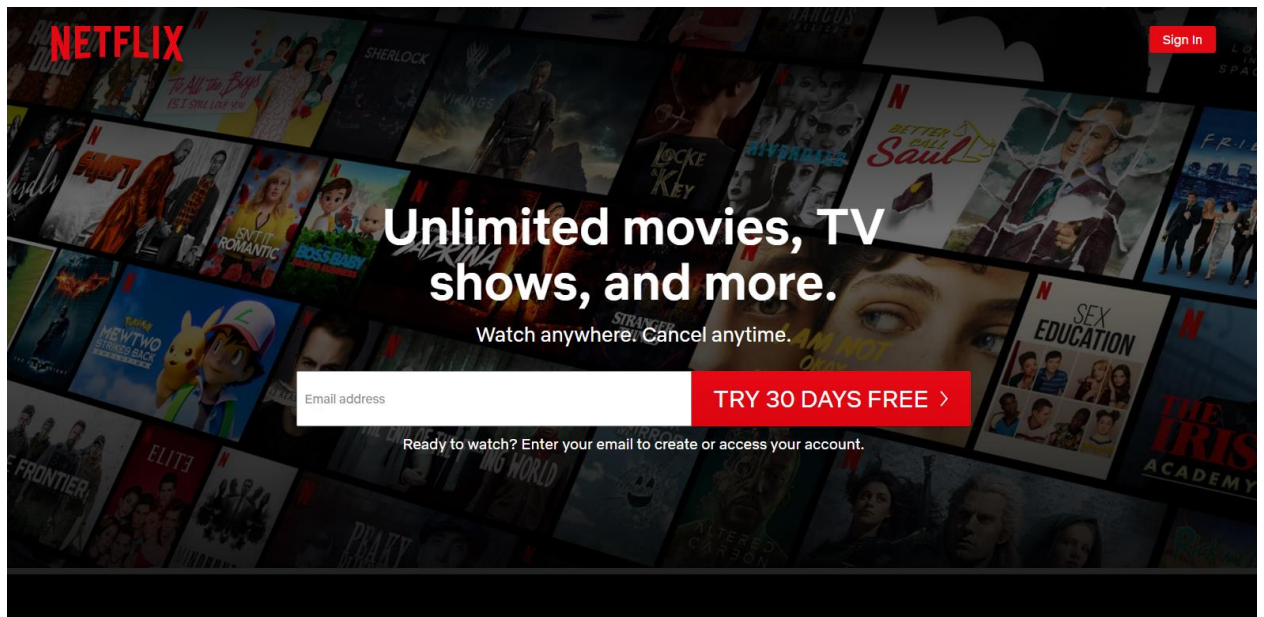


Рисунок 2. Домашняя страница сервиса Netflix

Для доступа к базе данных многочисленных интернет-сервисов, необходима платная подписка, что могут позволить себе не все желающие ознакомиться с информацией о сериале. Так же просмотр сериалов возможен и по телевидению, что несет за собой необходимость запоминания просмотренных серий, за неимением возможности отметки на существующих сайтах или записи в блокнот.

5. Анализ задачи

5.1. Варианты использования системы

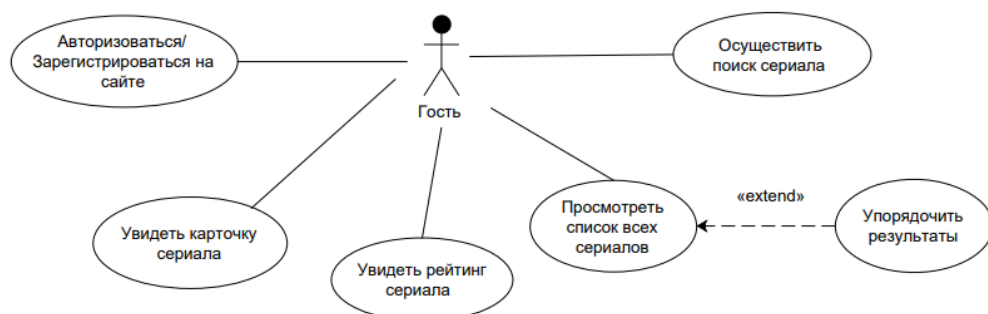


Рисунок 3. Часть диаграммы прецедентов с актором Гость.

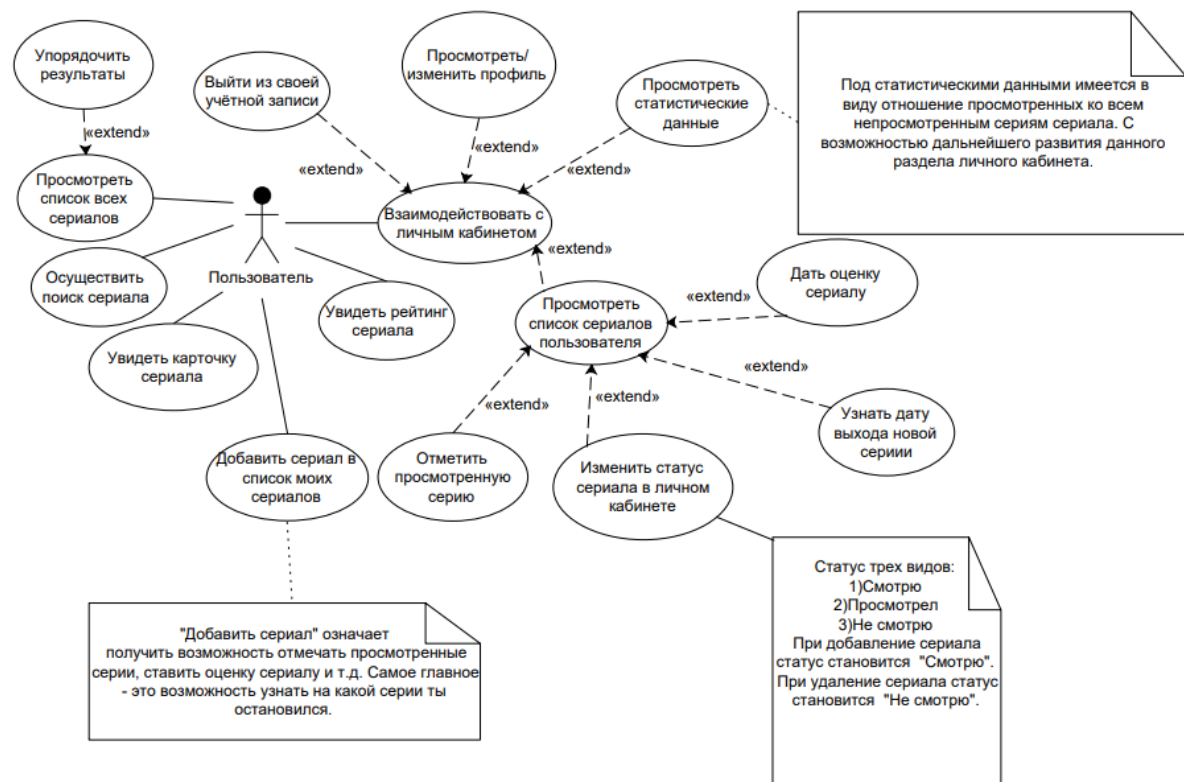


Рисунок 4. Часть диаграммы развертывания с актором Пользователь.

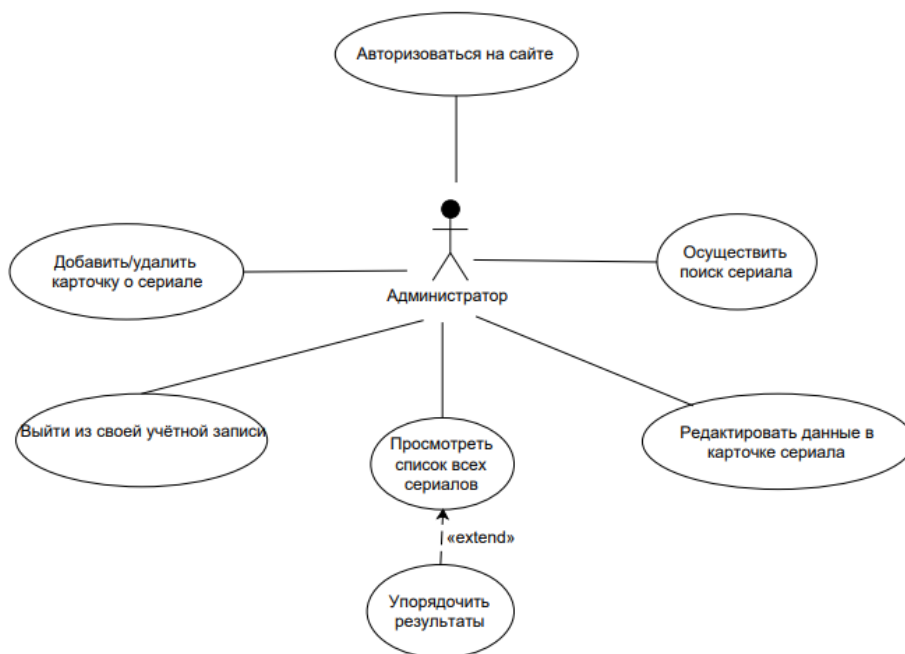


Рисунок 5. Часть диаграммы прецедентов с актором Администратор.

При взаимодействии с приложением выделяются три роли: гость, пользователь, администратор. Их роли наглядно изображены на рисунке.

Гость:

- Авторизоваться/Зарегистрироваться на сайте
- Увидеть карточку сериала
- Увидеть рейтинг сериала
- Посмотреть список всех сериалов
 - Упорядочить результаты
- Осуществить поиск сериала

Пользователь:

- Взаимодействовать с личным кабинетом
 - Выйти из учетной записи
 - Посмотреть/Изменить профиль
 - Просмотреть статистические данные
 - Посмотреть список сериалов пользователя
 - Отметить просмотренную серию
 - Изменить статус сериала в личном кабинете
 - Узнать дату выхода новой серии
 - Дать оценку сериалу/серии

Администратор:

- Авторизоваться на сайте
- Осуществить поиск сериала
- Добавить/удалить карточку о сериале
- Редактировать данные в карточке сериала
- Посмотреть список всех сериалов
 - Упорядочить результаты

- Выйти из своей учетной записи

5.2. Общая структура системы

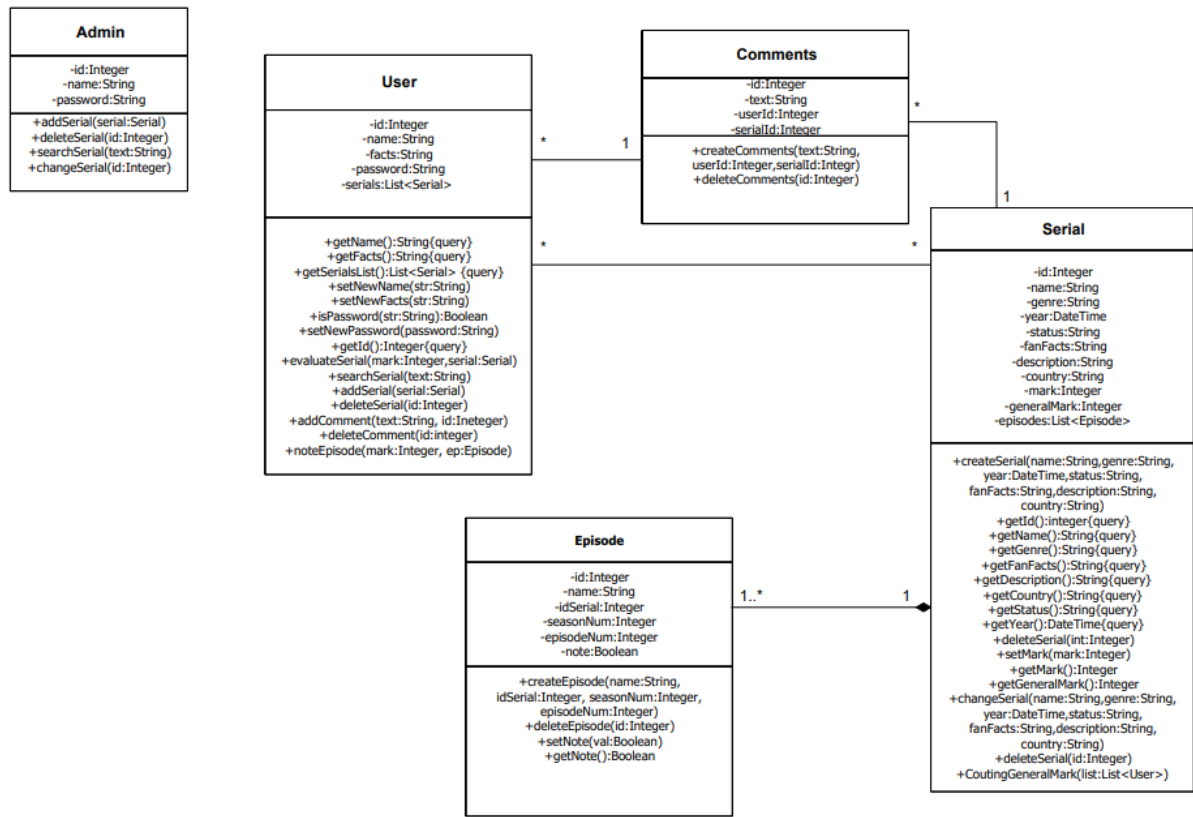


Рисунок 6. Диаграмма классов.

На рисунке 6 изображена диаграмма классов. Она показывает набор классов и интерфейсов, а также их связи. Между классами User, Comments, Serial установлены взаимосвязи ассоциации. Между классами Episode и Serial установлена взаимосвязь композиции.

5.3. Конфигурация системы

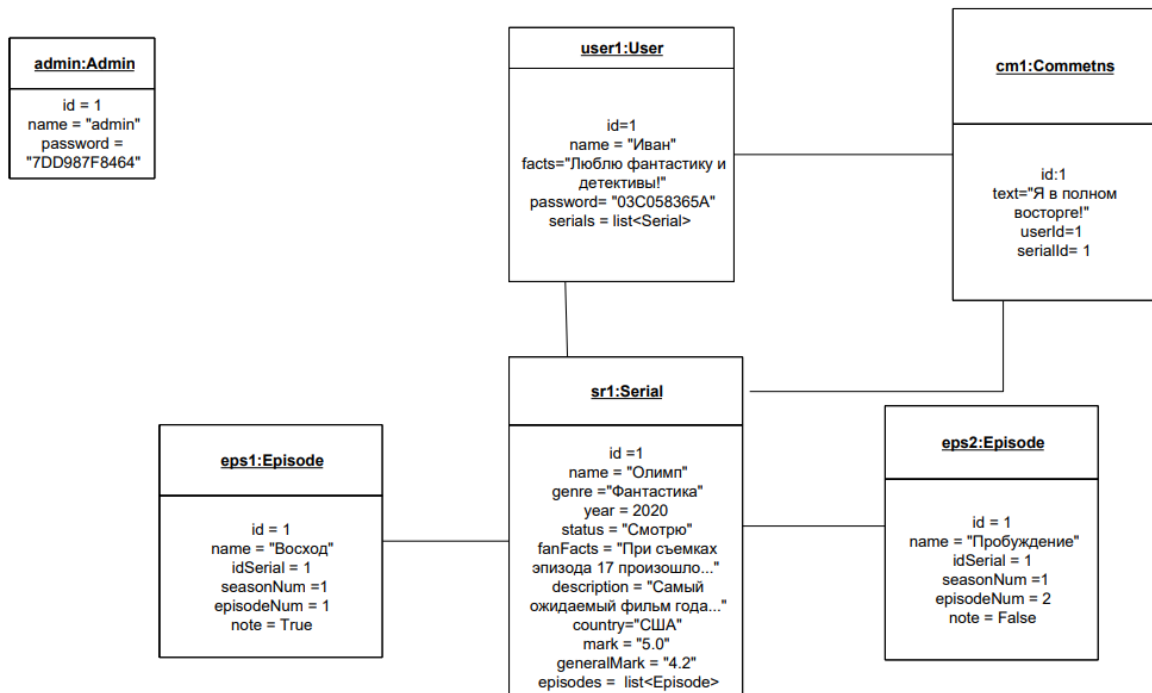


Рисунок 7. Диаграмма объектов.

5.4. Взаимодействие компонентов системы

На следующих рисунках изображены диаграммы последовательности для ролей пользователей системы, на которых изображены упорядоченные во времени взаимодействия объектов.

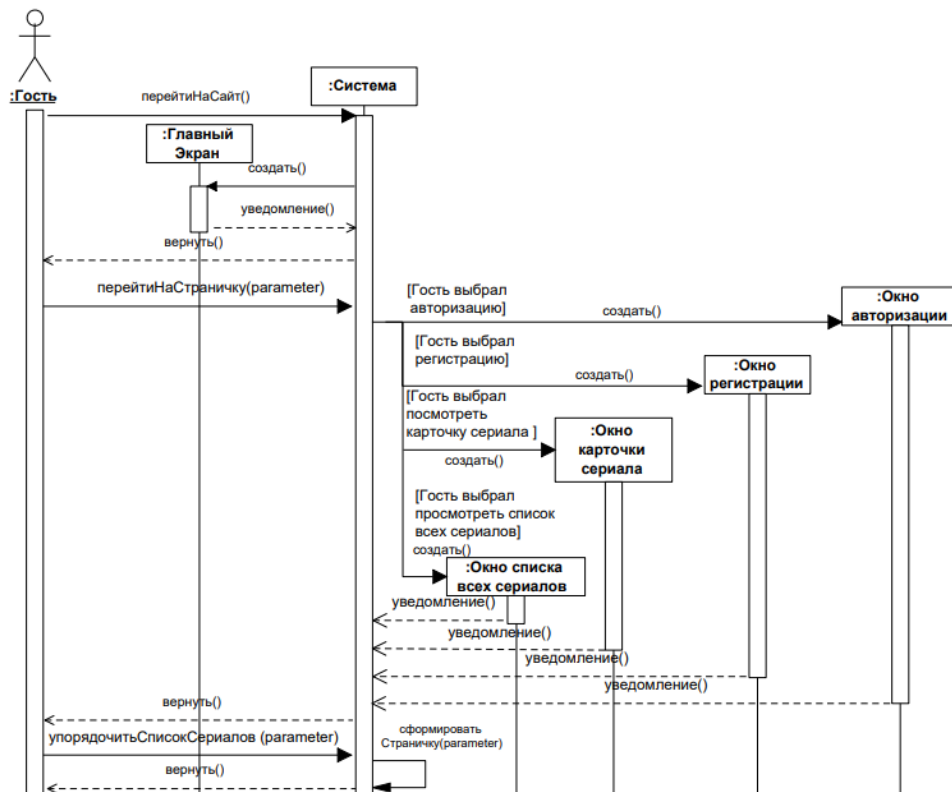
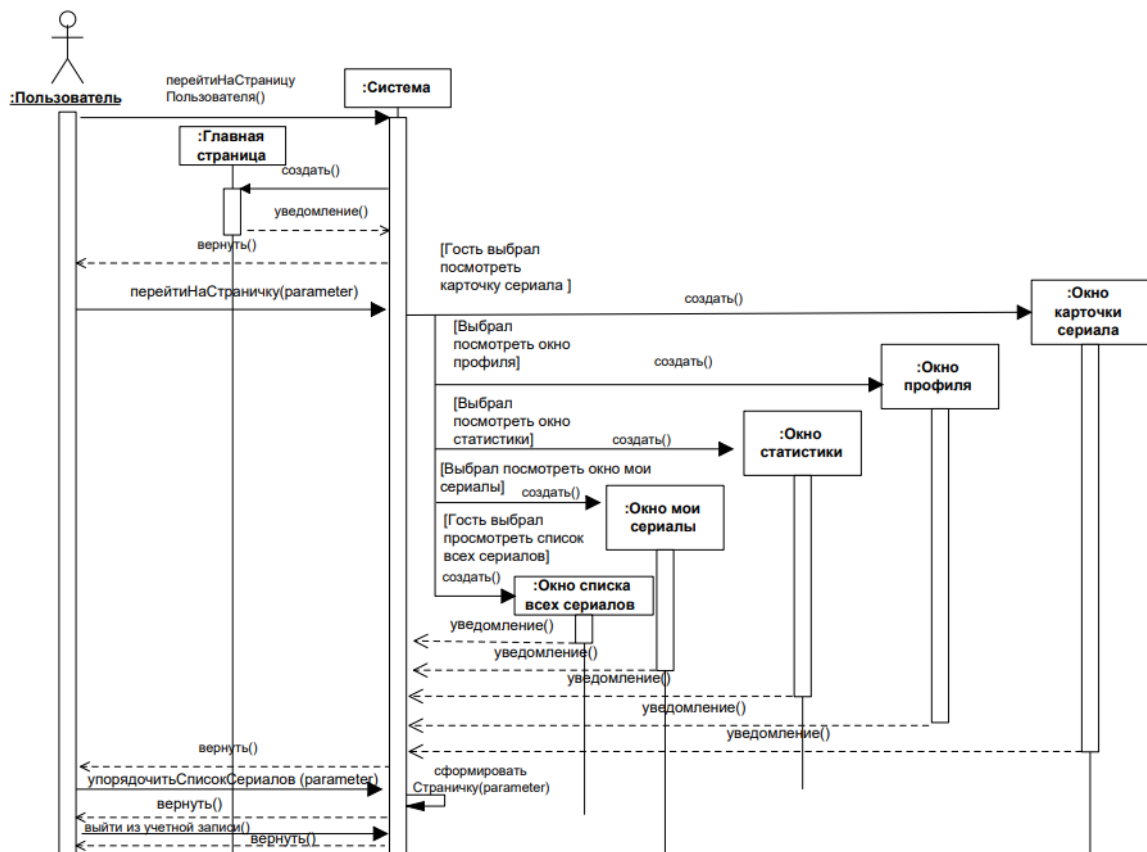


Рисунок 8. Диаграмма последовательностей гостя.



```
sequenceDiagram
    actor Admin as :Администратор
    participant System as :Система
    participant Main as :Главная страница
    participant AddSerial as :Окно добавления сериала
    participant EditSerial as :Окно редактирования

    Admin->>System: перейтиНаСтраницу Администратора()
    activate System
    System->>Main: создать()
    activate Main
    Main->>System: уведомление()
    deactivate Main
    System-->>Admin: вернуть()
    Admin->>System: перейтиНаСтраничку(parameter)
    activate System
    System->>AddSerial: создать()
    activate AddSerial
    AddSerial->>EditSerial: создать()
    activate EditSerial
    EditSerial-->>System: уведомление()
    deactivate EditSerial
    AddSerial-->>System: уведомление()
    deactivate AddSerial
    System-->>Admin: вернуть()
    Admin->>System: упорядочитьСписокСериалов(parameter)
    activate System
    System->>System: сформировать Страничку(parameter)
    activate System
    System-->>Admin: вернуть()
    deactivate System
    Admin->>System: удалитьСериал(parameter)
    activate System
    System-->>Admin: вернуть()
    deactivate System
    Admin->>System: выйти из учетной записи()
    activate System
    System-->>Admin: вернуть()
    deactivate System
    deactivate System
```

5.6. Взаимодействие в системе

Рисунок 11. Диаграмма взаимодействия.

На рисунке 11 изображена диаграмма взаимодействия, которая представляет взаимодействие между объектами системы и сообщения, которыми они обмениваются.

5.7. Варианты состояния системы

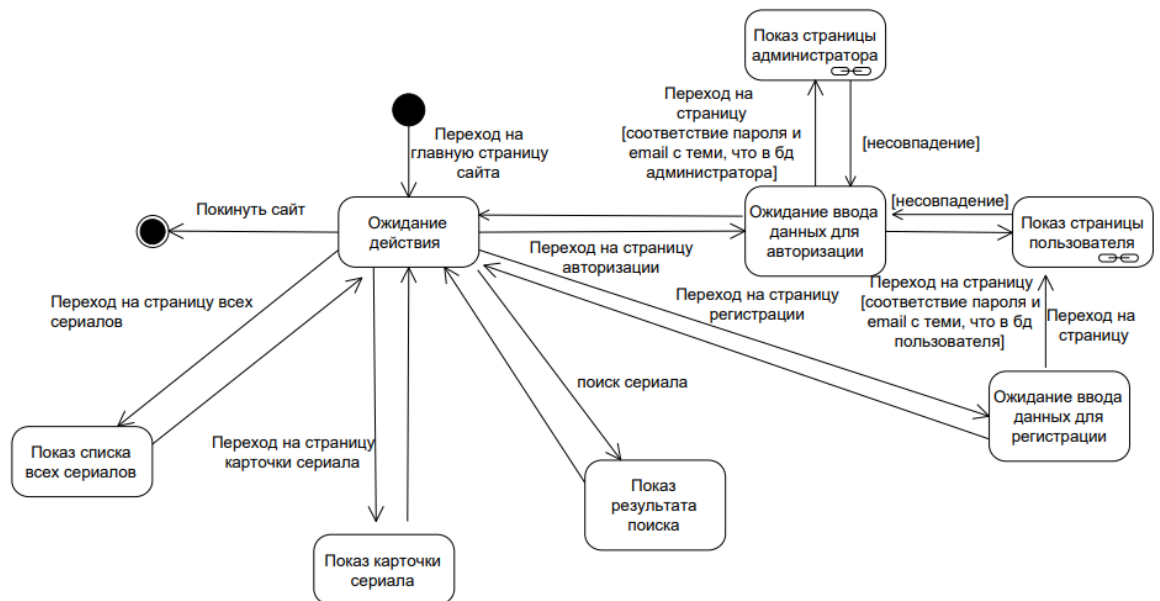


Рисунок 12. Диаграмма состояний.

Диаграмма состояний, изображенная на рисунке 12, отражает возможные состояния системы. При переходе на главную страницу сайта, система находится в ожидании выбора действия. В зависимости от выбора гостя возможны следующие возможные цепочки событий:

- Переход на страницу всех сериалов
- Переход на страницу карточки сериала
- Поиск сериала
- Переход на страницу регистрации
- Переход на страницу авторизации

Если пользователь системы выбирает переход на страницу всех сериалов, система переходит в состояние показа данных всех сериалов из базы данных системы. Далее возвращается в состояние ожидания действия.

Если пользователь системы выбирает переход на страницу карточки сериала, система переходит в состояние показа данных сериала из базы данных системы. Далее возвращается в состояние ожидания действия.

Если пользователь системы выбирает поиск сериала, система переходит в состояние показа результата поиска сериала в базе данных. Далее возвращается в состояние ожидания действия.

Если пользователь системы выбирает переход на страницу регистрации, система переходит в состояние ожидания ввода данных для регистрации нового пользователя. Далее система переходит к показу страницы пользователя

Если пользователь системы выбирает переход на страницу авторизации, система переходит в состояние авторизации. Если данных пользователя нет в базе данных, то система возвращается в состояние ожидания авторизации. Если данные пользователя были опознаны, как данные администратора, система переходит к показу страницы администратора.

5.8. Действия с системой

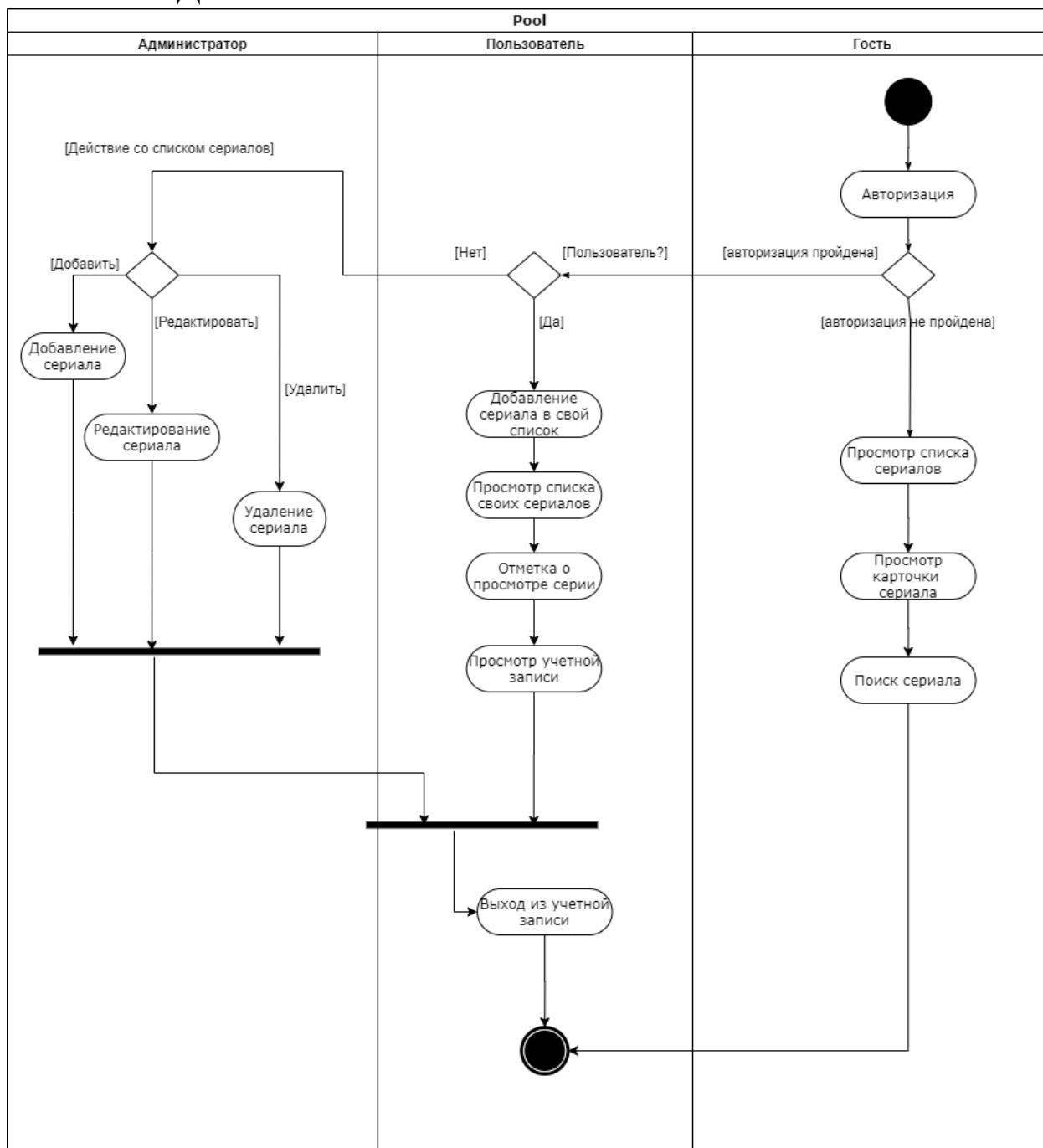


Рисунок 13. Диаграмма активности.

На рисунке 13 изображена диаграмма состояний, которая показывает поток последовательности действий для перехода от одной деятельности к другой.

5.9. Развертывание системы

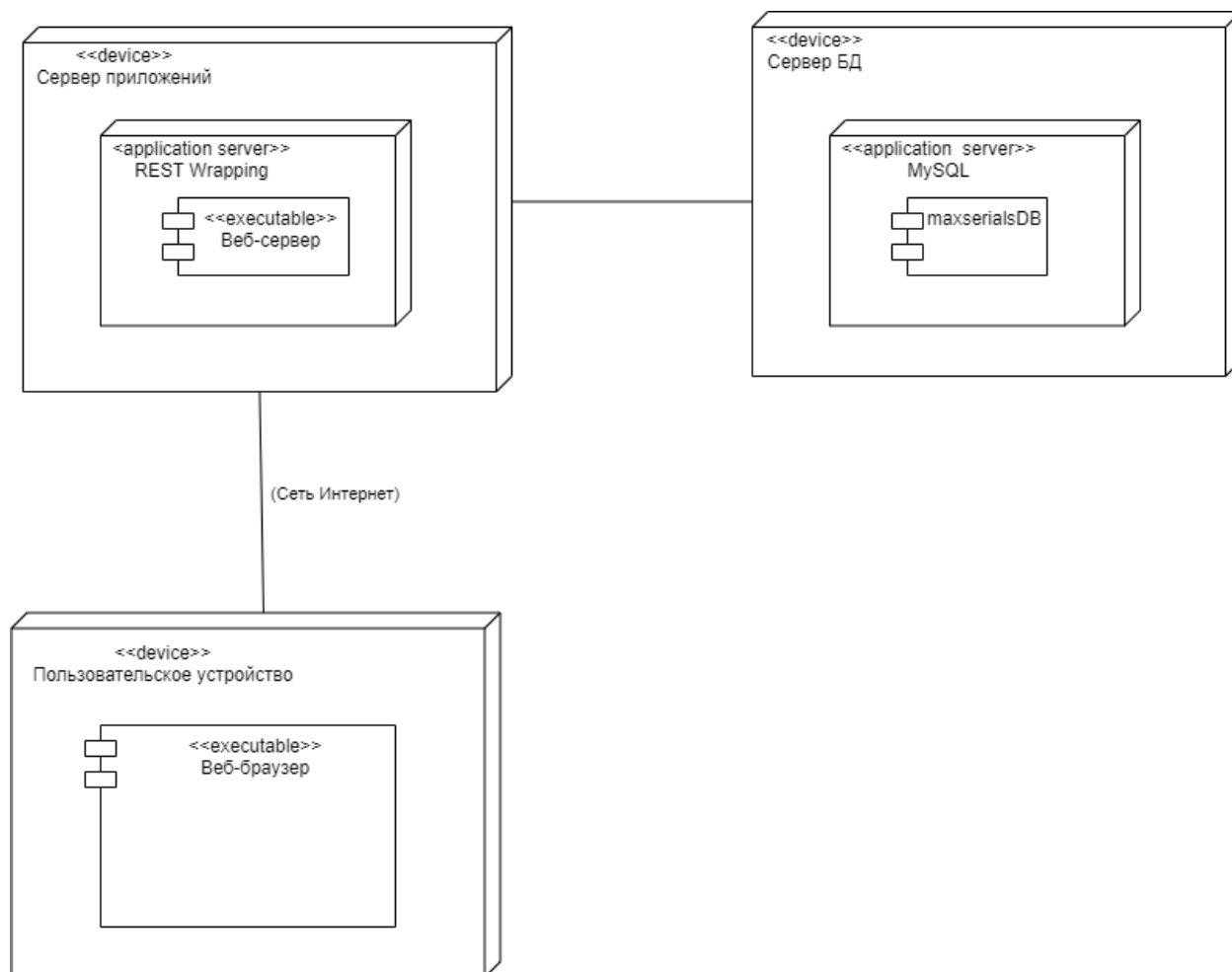


Рисунок 14. Диаграмма развертывания.

На рисунке 14 изображена диаграмма развертывания, показывающая топологию системы и распределение компонентов по ее узлам, а так же соединения – маршруты передачи информации.

5.10. IDEF0

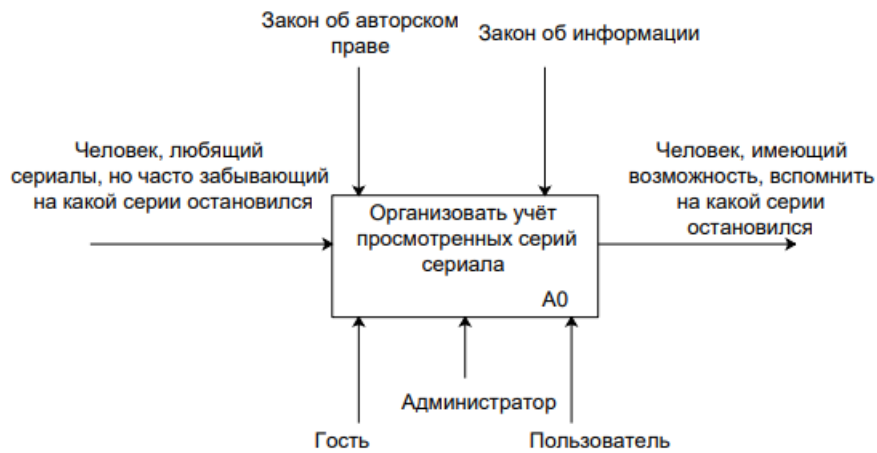


Рисунок 15. IDEF0

На рисунке 15 изображена IDEF0 диаграмма.

5.11. ER-диаграмма

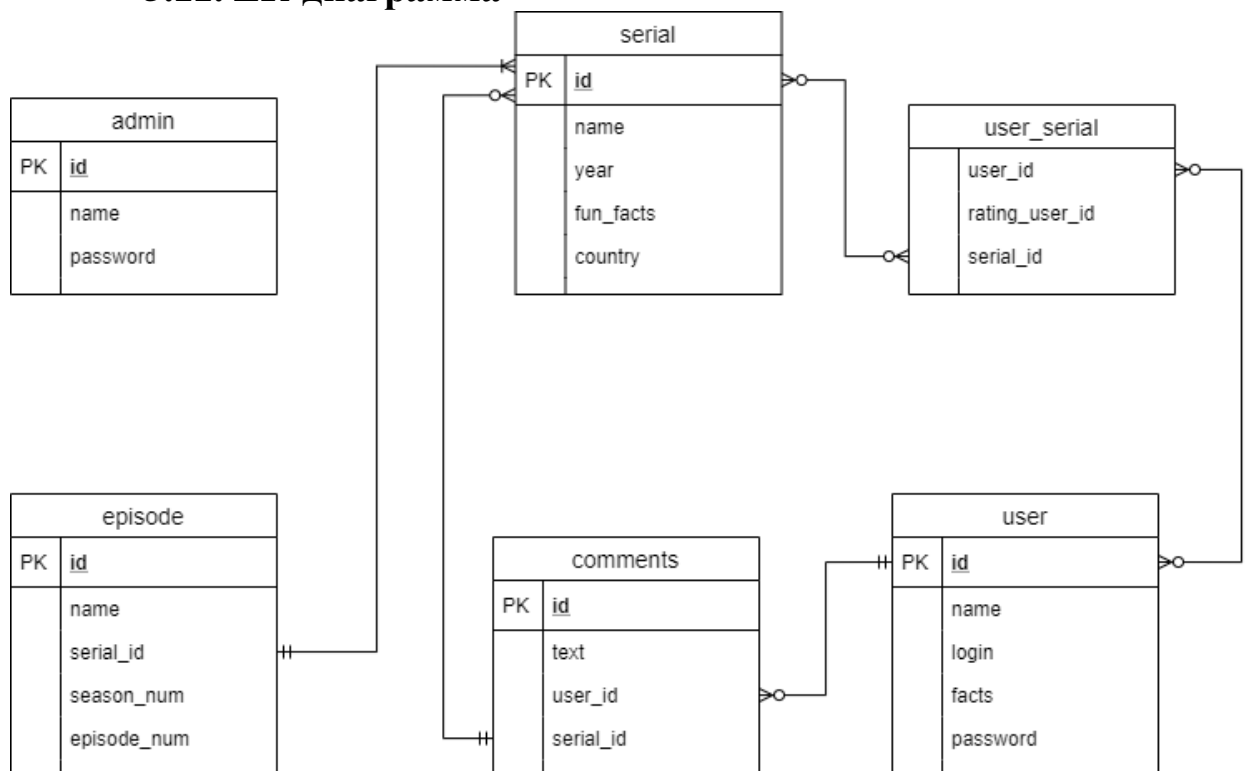


Рисунок 15. ER-диаграмма

На рисунке 15 изображена ER-диаграмма, схема «сущность-связь», показывающая, как связаны между собой «сущности» внутри системы.

6. Анализ средств реализации

В качестве средств реализации приложения были выбраны следующие технологии:

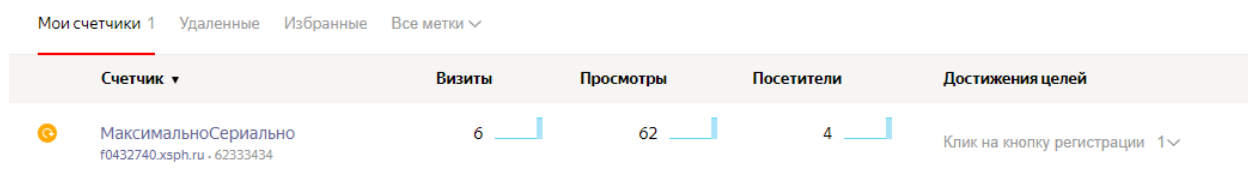
- Brackets – текстовый редактор для веб-разработки. Ориентирован на работу с HTML, CSS, JavaScript. Имеет множество расширений, которые позволяют сделать приложение более гибким. Включает такие функции, как завершение кода, переход к определению и многое другое.
- MySQL – свободная реляционная система управления базами данных. Поддерживает архитектуру клиент-сервер. Преимуществами являются быстрота, надежность и легкость использования.
- В качестве языка разработки использовался язык PHP. Этот язык специально сконструирован для веб-разработки и может интегрироваться в HTML. Также, сценарии выполняются на стороне сервера, следовательно, не зависит от аппаратного обеспечения клиента.
- Шаблон проектирования MVC выбран по причине разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компоненты: модель, представление и контроллер.

7. Метрика

Метрика – инструмент веб-аналитики, позволяющий получать наглядные отчеты о действиях пользователей системы, отслеживания источников трафика и других параметров.

Веб-аналитика – система сбора и анализа информации о посетителях сайта для улучшения и оптимизации системы работы ресурса. Главным назначением является мониторинг посещений веб-страниц.

При помощи сервиса Яндекс.Метрика подключены счетчики для анализа посещений и посетителей ресурса.







Мои счетчики 1 Удаленные Избранные Все метки ▾				
Счетчик ▾	Визиты	Просмотры	Посетители	Достижения целей
 МаксимальноСериально f0432740.xsph.ru - 62333434	6 	62 	4 	Клик на кнопку регистрации 1 ▾

Рисунок 16. Яндекс.Метрика

Причины построения воронок:

- Изучение поведения гостя и пользователя сервиса
- Получение статистических данных и сравнение их с ожидаемыми

Окончательный анализ воронок будет представлен в пункте Реализация.

8. Реализация

9. Тестирование

Заключение