**Министерство образования и науки Российской Федерации**

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

**ДАЛЬНЕВОСТОЧНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**

**(ДВФУ)**

|  |
| --- |
| **ШКОЛА ЕСТЕСТВЕННЫХ НАУК**  **Кафедра информатики, математического и компьютерного моделирования** |

|  |
| --- |
| **ПРОЕКТНАЯ ДОКУМЕНТАЦИЯ**  По дисциплине «Языки и методы программирования»  Специальность 01.03.02 «Прикладная математика и информатика»  Курсовой проект «**Система управления кинотеатром**» |

|  |  |
| --- | --- |
|  | Выполнили студенты группы Б8203б  \_\_\_\_Михайленко Антон Денисович\_\_\_\_\_  \_Абдурашидов Абдувохоб Акмал Угли\_\_  \_\_\_\_Вотинцева Дарья Игоревна\_\_\_\_\_  Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Сущенко  (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (зачтено/не зачтено)  «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |

г. Владивосток

2018

**Оглавление**

[Введение 4](#_Toc295959746)

[Глава 1. Анализ предметной области 6](#_Toc295959747)

[1.1 Введение 6](#_Toc295959761)

[1.2 КИНО VL.RU 7](#_Toc295959761)

[1.3 МИР КИНО 7](#_Toc295959761)

[1.4 SAKH.COM 7](#_Toc295959761)

[1.5 Классификация веб-сайтов для взаимодействия с кинотеатрами. 8](#_Toc295959761)

[1.6 Заключение 9](#_Toc295959761)

[Глава 2. Анализ требований к системе 10](#_Toc295959760)

[2.1 Введение 10](#_Toc295959761)

[2.2 Глоссарий 10](#_Toc295959762)

[2.3 Требования к веб-сайту 11](#_Toc295959762)

[2.4 Требования к реализации 13](#_Toc295959763)

[2.5 Заключение 13](#_Toc295959767)

[Глава 3. Проектирование программного средства 14](#_Toc295959768)

[3.1 Введение 14](#_Toc295959769)

[3.2 Диаграмма прецедентов 15](#_Toc295959770)

[3.3 Диаграмма классов 16](#_Toc295959771)

[3.4 Диаграма структуры сайта 18](#_Toc295959772)

[3.5 User Flow диаграмма 19](#_Toc295959773)

[3.6 Проект пользовательского интерфейса 21](#_Toc295959774)

[3.7 Заключение 23](#_Toc295959776)

[Глава 4. Тестирование 24](#_Toc295959777)

[3.1 Введение 24](#_Toc295959778)

[4.2. Описание пользовательских функций 24](#_Toc295959779)

[4.3 Функциональное тестирование 29](#_Toc295959780)

[Глава 5. Проектирование БД 30](#_Toc295959785)

[5.1 Введение 30](#_Toc295959786)

[5.2 ER-диаграмма, демонстрирующая структуру основной БД. 30](#_Toc295959787)

[5.5 Выбор СУБД 33](#_Toc295959790)

[5.6 Заключение 35](#_Toc295959791)

[Заключение. 36](#_Toc295959785)

**Введение**

Несмотря на то, что интернет давно и прочно вошел в нашу жизнь, многие предприниматели и даже крупные фирмы не понимают, что им даст создание собственного сайта, ведь есть другие хорошо зарекомендовавшие себя проверенные способы саморекламы: телевидение, радио, СМИ, баннеры, флайеры и тому подобное.

Актуальность создания сайта состоит также в том, что если вы хотите донести информацию максимально быстро до огромного количества людей, то лучше, чем с помощью собственного сайта сделать это не получится никак.

Веб-сайт позволяет представить информацию о компании и ее услугах или услугах сжато и одновременно полноценно. Также сайт может сообщать о новостях фирмы, об изменениях в прайсе или режиме работы, содержать отзывы благодарных клиентов. Для кинотеатра выгода в том, что сайт приносит новых посетителей, позволяет удобно управлять всеми ресурсами, получать статистические данные.

При работе с сайтом возникают проблемы для разных групп пользователей: бронирование билетов, редактирования фильмов, добавление фильмов, добавление отзывов. Эти проблемы необходимо решить: разграничить права для пользователей, сделать удобный пользовательский интерфейс, реализовать общение с базой данных

Таким образом, необходимо разработать веб-сайт для взаимодействия между потенциальными посетителями и самим кинотеатром.

Система должна удовлетворять среднему пользователю, обладать гибким интерфейсом, быть доступной с любой точки земного шара.

Цель курсового проекта: спроектировать веб-сайт для взаимодействия между потенциальными зрителями и кинотеатром. Для этого необходимо:

1. Изучить особенности наиболее популярных сайтов кинотеатров.
2. Сформулировать требования к программному средству.
3. Разработать проект программного средства, пригодный для дальнейшей реализации.
4. Создать программный продукт – веб-сайт для взаимодействия с кинотеатрами.

**Глава 1. Анализ предметной области.**

**1.1 Введение**

***Веб-сайт*** - совокупность файлов, документов, отраженных при помощи языка программирования таким образом, чтобы их видели пользователи сети Интернет. Другими словами, сайты включают в себя любую текстовую, графическую, аудио- или видеовизуальную информацию, собранную на странице или нескольких страницах.

Веб-сайты для взаимодействия с кинотеатрами включают в себя регистрацию пользователей с различными привилегиями, возможность смотреть расписание фильмов, бронировать билеты. Некоторые имеют партнерские программы.

Зарабатывают двумя основными способами:

1. Показ рекламы. Могут рекламировать как различные кинофильмы, так и посторонние сервисы, связанные со сферой досуга.
2. Получение процента от продажи билета. При продаже(бронировании) билетов через сайт получает заранее установленную сумму(процент от суммы)

Наиболее известные сайты:

* https://kino.vl.ru/films/
* http://nsk.mirkino.pro/
* https://bilet.sakh.com/cinema

**1.2 КИНО VL.RU** (https://kino.vl.ru/films/)

Веб-сайт, предназначенный для поиска/покупки/бронирования билетов в кино.

1. Регистрация, управление профилем – авторизация через соц. сети, данные берутся из профиля соц. сети
2. Получение информации о фильмах, работе кинотеатров города.
3. Возможность оставлять отзывы о фильмах.
4. Администрирование – администраторы могут добавлять фильмы, сеансы, кинотеатры. Также могут редактировать данные пользователей.

**1.3 МИР КИНО** (http://nsk.mirkino.pro/)

Веб сайт для нескольких городов, предназначенный для взаимодействия между потенциальным зрителем и кинотеатром.

1. Регистрация, управление профилем – для покупки и бронирования билетов.
2. Предоставляет обширные возможности для получения информации о фильмах/сеансах.
3. Есть вкладка новости, где показываются новинки кино.
4. Администрирование – позволяет редактировать данные о фильмах, кинотеатрах и т.д.

**1.4. SAKH.COM** (https://bilet.sakh.com/cinema)

Веб сайт с плоским дизайном, адаптирующийся под каждую платформу, для удобной покупки билетов в кино.

1. Личный кабинет, просмотр своих сеансов.
2. Предоставляет достоверную информацию о кино и билетах.
3. Понятный интерфейс взаимодействия с пользователем.
4. Возможность администрирования – добавления/изменения фильмов

**1.5 Классификация веб-сайтов для взаимодействия с кинотеатрами.**

На основе анализа предметной области веб-сайтов для взаимодействия с кинотеатрами можно составить классификацию по следующим характеристикам:

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | **КИНО VL.RU** | **МИР КИНО** | **SAKH.COM** |
| Регистрация, управление профилем | Есть, через соц. сети | Есть, через почту | Есть через телефон |
| Информативность о деятельности кинотеатров | Есть, по клику на кинотеатр, выдает о нем сводку | Вкладка «кинотеатры», общая информация о кинотеатрах | Адрес кинотеатра и его сеансы |
| Актуальность информации о фильмах | Есть, вкладка «премьеры недели» | Есть, вкладка «новости» | Вкладка кино, с новинками |
| Администрирование | Есть, администратор может изменять фильмы, кинотеатры | Есть, администратор может добавлять/изменять фильмы | Есть, администратор может изменять данные пользователей, фильмы, сеансы |

**1.6. Заключение**

Проанализировав эти сайты, можно сформулировать пользовательские Предпочтения к будущему программному продукту.

Пользователям важно:

* Наличие меню с афишами

Имеется на всех сайтах, позволяет пользователям увидеть текущие фильмы.

* Сортировка по фильмам и датам

Позволяет подобрать подходящее время и нужный фильм.

* Отображение свободных мест на сеансе и бронирование

Для выбора удобного места и экономии времени.

* Информация о фильме

Позволяет узнать, о чем фильм.

* Отзывы других зрителей

Позволяют решить стоит ли идти на данный фильм.

* Описание различных кинотеатров

Для поиска удобного кинотеатра в ближайшей доступности.

**Глава 2. Анализ требований к системе.**

**2.1 Введение**

Основываясь на анализе предметной области и задачах, необходимо разработать веб-сайт для взаимодействия между зрителями и кинотеатрами. Внедрение такого программного средства (ПС) позволит сократить время обработки каждого потенциального зрителя кинотеатра. Также увеличит их количество.

**2.2 Глоссарий**

1) ***ПС*** – программное средство

2) ***Потенциальный зритель*** – пользователь, перешедший на сайт кинотеатра, интересующийся новыми фильмами.

3) ***Личный кабинет*** – данные о пользователе:

Обязательные:

Телефон или email

Необязательные

Имя

4) ***Регистрация*** — это создание своей учетной записи или, иными словами, заведение своего персонального имени (логина), которое будет отличать вас от всех остальных и от лица, которого вы сможете взаимодействовать с другими людьми на сайте.

5) ***Авторизация*** — это ввод своего уникального логина и пароля, которые вы указали при регистрации

**2.3 Требования к веб-сайту**

**1) Регистрация пользователей**

Веб-сайт должен позволять посетителям регистрироваться с помощью email или номера мобильного телефона. Данные о регистрации должны сохраняться в базу данных на сервере. Должны соблюдаться все права пользователя N 152-ФЗ «О персональных данных».

**2) Авторизация пользователей**

Веб-сайт должен позволять входить на сайт уже зарегистрированным пользователем, используя в качестве логина свой email или номер мобильного телефона.

**3) Бронирование билетов**

Пользователь, авторизовавшийся на сайте, может бронировать билет на конкретный сеанс в конкретном кинотеатре. Пользователь может выбирать место, фильм, кинотеатр, время. Каждому бронированию билета присваивается свой уникальный идентификатор, позволяющий пользователю получить распечатанный билет в кинотеатре.

**4) Личный кабинет**

Отдельная страница на сайте, которая показывает текущую информацию о пользователе (email, телефон, имя и т.д.). А также историю посещенных сеансов.

Для отдельных групп пользователей предусмотрен переход в усовершенствованный личный кабинет с возможностью администрирования.

**5) Добавление отзывов**

Под каждым фильмом расположено место для отзывов, позволяющее получить обратную связь о фильме. Авторизованный пользователь может добавлять отзывы через специальную форму, используя свой логин. Также пользователь может оставлять свою оценку о фильме.

**6) Редактирование информации о фильмах**

Предусмотрено для отдельных групп пользователей:

• администратор

• кинотеатр

Позволяет редактировать информацию о фильмах, добавлять изображения и менять описание.

**7) Редактирование информации о кинотеатрах**

Доступно пользователю с правами администратора через личный кабинет. Возможно добавление нового кинотеатра, редактирование описания, местоположения, доступных сеансов и т.д.

**8) Веб-интерфейс**

Веб-сайт должен иметь удобный для пользователя интерфейс, обеспечивающий удобный просмотр информации с различных девайсов.

**9) Данные (База данных) на сервере**

Данные на сервере должны быть структурированы. Имеется несколько таблиц, содержащие в себе данные о:

* Пользователях
* Кинотеатрах
* Сеансах
* Фильмах
* Билетах

**2.4 Требования к реализации**

Необходимо разработать систему, включающую в себя web-интерфейс, созданный при помощи:

* HTML, CSS, JavaScript, jQuery – FRONTEND
* JAVA, spring – BACKEND

Система должна поддерживать базу данных, хранимую на сервере. База данных создается при помощи СУБД MySQL.

**2.5 Заключение**

В ходе требований к будущему веб-сайту был сформирован ряд четких условий, функций необходимых для ее реализации. На основании проведенного анализа можно приступать к проектированию ПС.

**Глава 3. Проектирование программного средства**

**3.1 Введение**

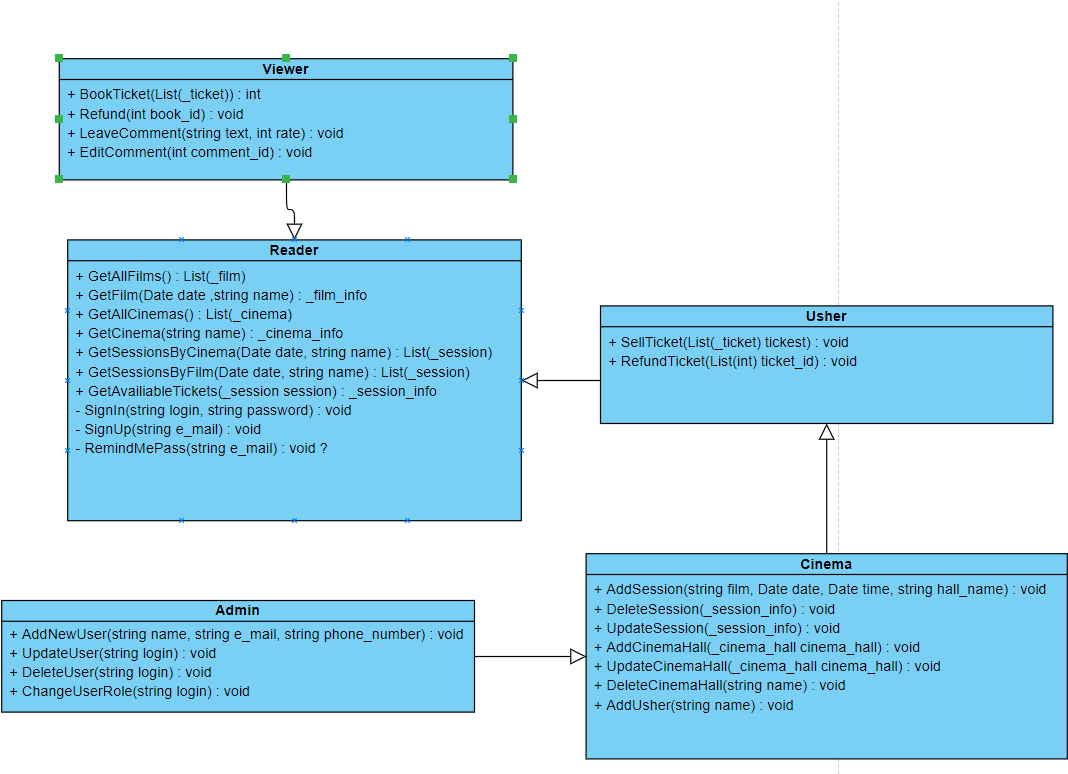
Проектирование программного средства – это последний этап разработки ПО, который контролируется заказчиком. В нем утверждаются все подсистемы системы, потоки данных между подсистемами, основные функции и подфункции пользователей, их тестирование. Также прогнозируется результата ответа системы в любой момент работы. Описывается связь объектов системы. Процесс проектирования сопровождается построением разных диаграмм, отражающих пользовательское восприятие системы, и поясняющих разработчику поставленные перед ним задачи.

На основании анализа предметной области и требований, необходимо:

1. Построить диаграмму прецедентов.
2. Построить диаграмму классов.
3. Построить диаграмму структуры сайта.
4. Построить User Flow диаграмму.
5. Разработать проект пользовательского интерфейса.

**3.2 Диаграмма прецедентов.**

**3.3 Диаграмма классов.**

****

Есть 5 классов, отвечающие за функционал пользователя, принадлежащего конкретной группе.

Все классы являются потомками класса «Reader»

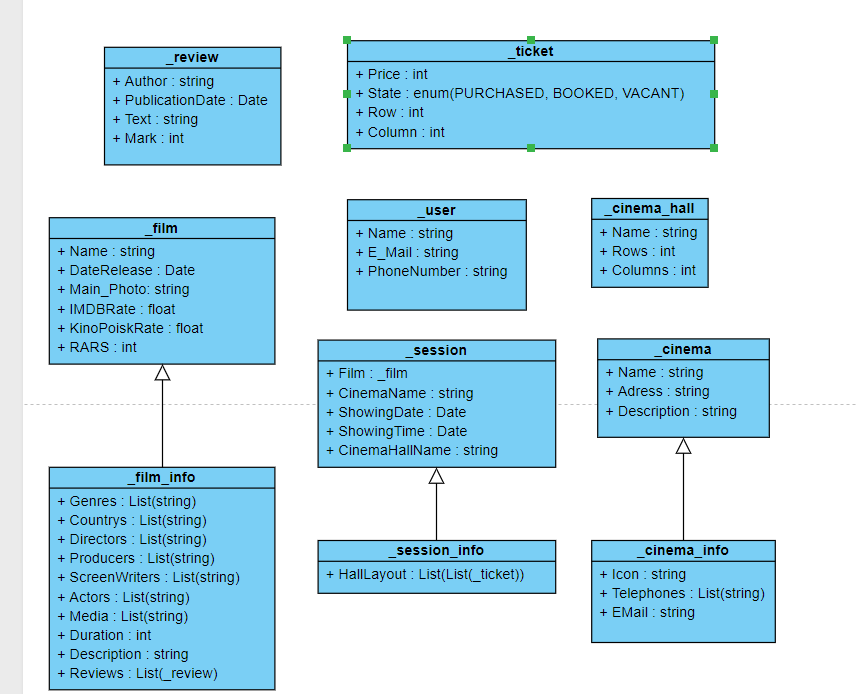
Класс Admin – «Админ» выполняет все функции редактирования, управления, обновления

Класс Cinema – «Кинотеатр» выполняет функции обновления информации о сеансах

Класс Usher – «Билетер» возможность продавать и возвращать билеты

Класс Viewer – «Зритель» может бронировать билеты, оставлять комментарии

Класс Reader – «Читатель» может смотреть информацию по фильмах, возможность авторизации/регистрации

****На этой диаграмме видно описание функций для работы с базой данных, и взаимодействия с веб интерфейсом.

Основные классы, получающие данные из соответствующих таблиц в БД.

\_review – получает отзывы

**\_ticket - получает информацию о билетах**

**\_film – получается информацию о фильмах**

**\_film\_info – подробная информация о фильмах (наследуется от \_film)**

**\_user – информация о пользователях**

**\_session – информация о сеансе**

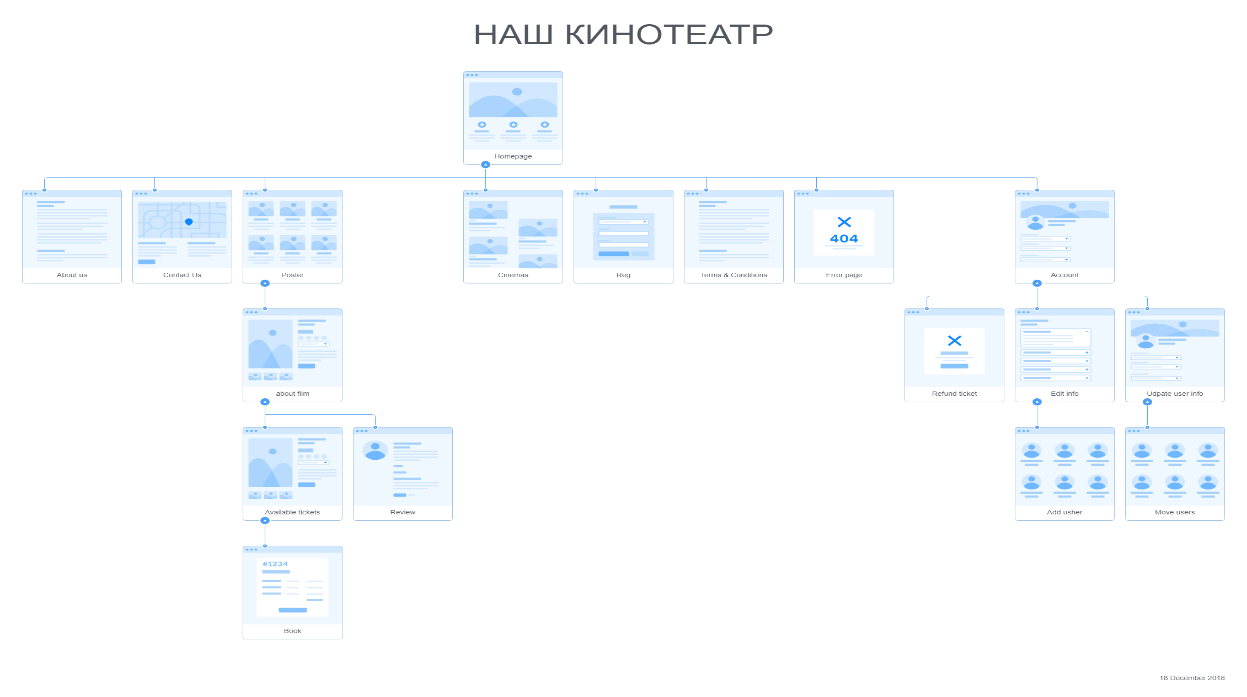
**\_session\_info – подробная информация о сеансе(количество билетов и тд) (наследуется от \_session)**

**\_cinema\_hall – информация о залах**

**\_cinema – информация о кинотеатрах**

**\_cinema\_info – подробная информация о кинотеатрах (наследуется от \_cinema)**

**3.4 Диаграмма структуры сайта.**

****

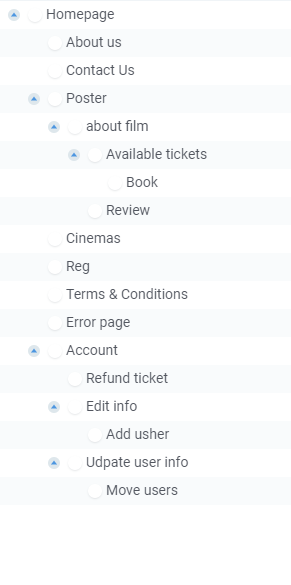


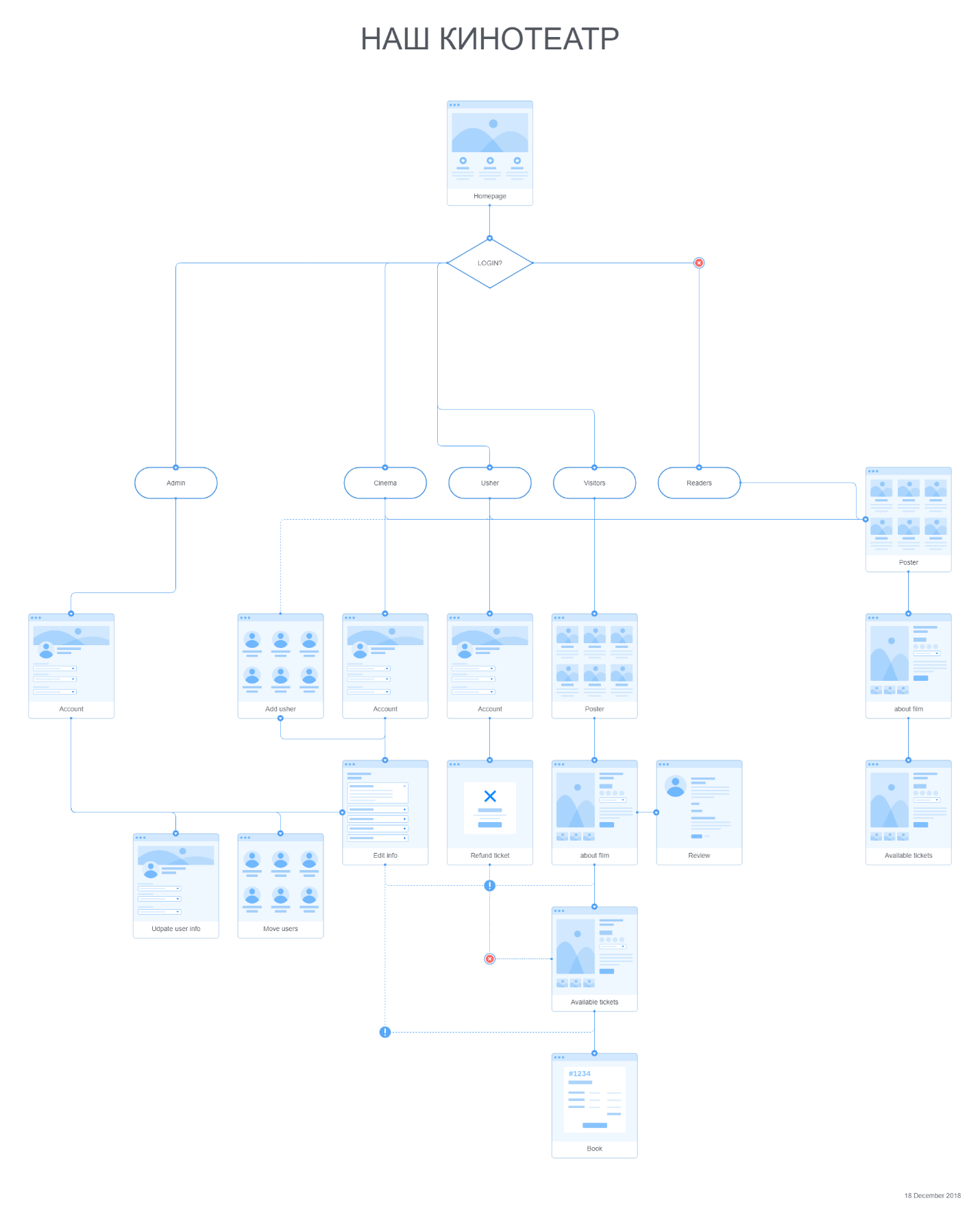
Диаграмма представляет из себя иерархическую структуру сайта.

Во главе стоит «Homepage», означающий страницу с АФИШЕЙ.

Остальные страницы раскрывают весь функционал данного сайта:

* Регистрация
* Личный кабинет
* Просмотр информации о фильмах
* Просмотр информации о кинотеатрах

**3.5 User Flow диаграмма.**

User Flow представляет из себя диаграмму, показывающая как посетитель может взаимодействовать с сайтом.

При регистрации и дальнейшей авторизации пользователь попадает в конкретную группу пользователей:

* Зритель
* Библиотекарь
* Кинотеатр
* Администратор

Зрители, которые не вошли на сайт – попадают в группу Читатели

В зависимости от группы у пользователя есть различные возможности.

У администратора и кинотеатра есть возможность редактировать различные параметры.

Билетер может выкупать билеты и управлять бронированием. Для этого предусмотрен дополнительный интерфейс.

Каждый пользователь может менять информацию о себе.

Редактирование параметров происходит через личный кабинет.

**3.6 Проект пользовательского интерфейса.**

Прототип пользовательского интерфейса разрабатывался с помощью платформы Figma. Также есть возможность посмотреть [ПРОТОТИП](https://www.figma.com/proto/gJGap6G8nruBzBjdZPC4eAzS/web?node-id=2%3A17&scaling=min-zoom).

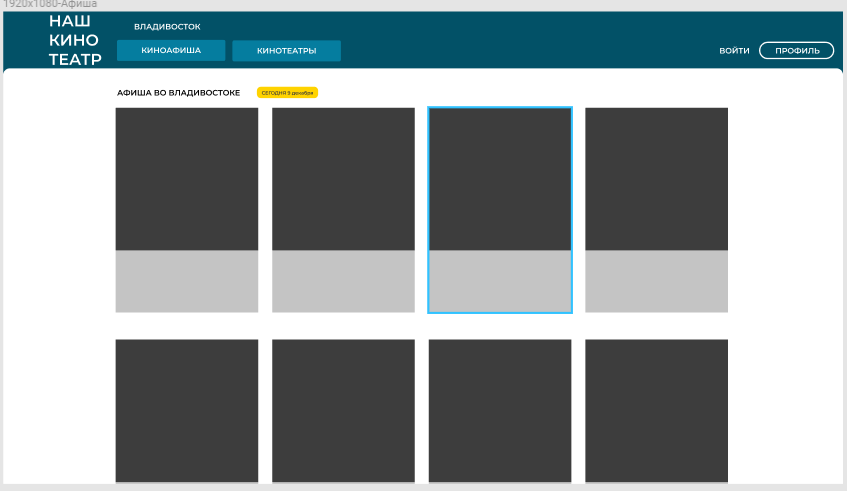
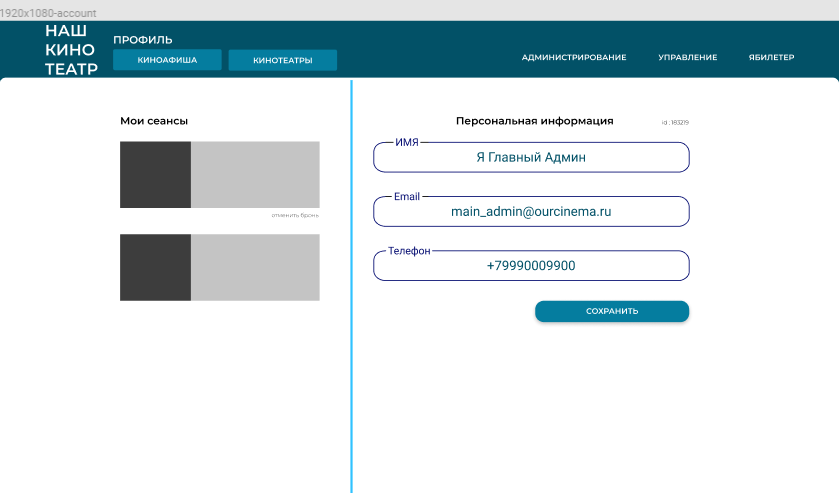
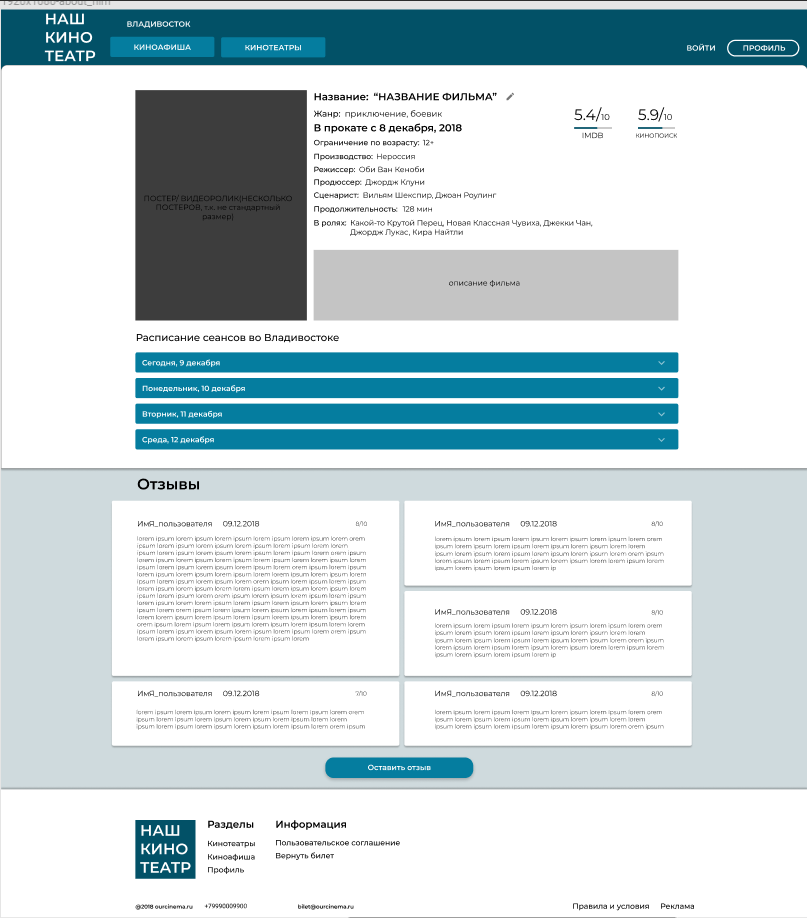


рис. 1

рис. 2

 рис. 3

На рис. 1 Представлен прототип главной страницы «КИНОАФИША»

Содержит меню, с кнопками перехода на страницу с описанием кинотеатров, а также регистрацию/авторизацию и переход в личный кабинет.

Рис. 2 представляет из себя страницу «ЛИЧНЫЙ КАБИНЕТ»

На странице есть информация о пользователе, а также история его сеансов. Также в зависимости от группы появляются кнопки для администрирования, управления кинотеатром и режим билетера.

Рис. 3 Представляет собой описание фильма, содержащее постер, краткую аннотацию и другую информацию о фильме.

Дальше идет место для отзывов, показываемые по новизне.

В футере(низ страницы) содержатся основные ссылки, а также контактная информация о создателях.

**3.7 Заключение**

На основе выявленных требований в предыдущей главе был сформирован проект ПС, позволяющий пользователю и разработчику окончательно договориться об однозначном понимании системы. Проект позволяет разработчику приступать к дальнейшему переходу к реализации. Но на данном этапе необходимо провести тестирование всех пользовательских функций, что и будет сделано в следующей главе.

**Глава 4. Тестирование**

**4.1 Введение**

Тестирование – это процесс проверки ПС на предмет его бесперебойной работы в независимости от действий пользователя. Существует несколько видов тестирования. Самыми распространенными являются функциональное тестирование (метод «черного ящика») и структурное тестирование (метод «белого ящика»)

Функциональное тестирование строится на основе анализа предметной области. Из анализа разработчик определяет пользовательские функции, для которых и будут спроектированы тестовые ситуации. На данном этапе необходимо уточнить пользовательские функции: описать входные данные, выходные данные, ограничения на процессы, контролируемые пользователем. Все это позволит, не зная устройства системы (“черного ящика”), построить ряд тестовых ситуаций, которые защитят программное средство от некорректных действий пользователя.

На основе модели поведения пользователя, спроектированной в предыдущей главе, можно сформировать тестовые ситуации, позволяющие ПС отвечать на любые действия пользователя.

**4.2 Описание пользовательских функций**

Пользовательские функции были описаны ранее в диаграмме прецедентов. Для успешного тестирования необходимо их уточнить.

Пользовательские функции:

1. Регистрация пользователей

Функция, позволяющая добавлять новых зрителей, т.е. создавать новый личный кабинет и заполнять данные пользователя, а именно:

Имя, email, телефон, id

Id – уникальный идентификационный код, состоящий 7 цифр (номер каждого зарегистрированного зрителя). Например: 0000001, 3232321.

Ограничение для входных данных:

1. Имя – строковая константа. Не превышает 45 символов.
2. Email – не превышает 45 символов.
3. Телефон в формате +7 (\*\*\*) \*\*\* \*\* \*\*, где \* числа от 0 до 9
4. Авторизация пользователей

Функция, позволяющая пользователю войти на сайт. Пользователь должен ввести свой логин – email/телефон и пароль, которые будут проверяться по базе данных. Должны выводиться соответствующие сообщения, если вход на сайт не увенчался успехом.

После успешного входа на сайт в меню должен отразиться логин пользователя, равный названию почты без @mail.

**Вход**: new\_user132@mail.ru

**Выход**: new\_user132

Если почты нету, то логином становится номер телефона (либо “user” + id).

1. Изменение пароля

Функция, позволяющая пользователю изменить пароль для доступа к личному кабинету

1. Оставление отзывов

Функция позволяет пользователю оставлять отзывы к фильмам.

Функция доступна после авторизации. Пользователь может оставить отзыв, длинной 250 символов. У каждого отзыва подпись – логин пользователя. Отзыв отображается под данным фильмом в порядке новизны.

1. Бронирование билетов

При выборе фильма и конкретного времени пользователь может выбрать ряд и место. При нажатии кнопки принять пользователь получает номер брони – уникальный идентификатор, состоящий из 8 цифр. Информация о данной брони появляется в личном кабинете пользователя.

1. Добавление данных о фильме

Это функция доступна администраторам и кинотеатрам.

Данная функция позволяет добавить:

* Название фильма (45 символов)
* Описание фильма (250 символов)
* Режиссеры/Сценаристы и т.д. (100 символов)
* Актёрский состав (100 символов)
* Рейтинг фильма
* Постер
* Дата выхода фильма

Эти данные заносятся в базу данных и фильм появляется в начале «киноафиши».

1. Добавление данных кинотеатра

Функция для администратора.

Позволяет добавить новый кинотеатр.

* Название кинотеатра
* Email
* Телефон
* Адрес
* Фото
* Какие залы
* Описание

После добавления зал появится в списке «Кинотеатры»

1. Добавление зала.

Функция доступна для администраторов и кинотеатров из вкладки «Управление»

Добавляет новый зал

* Кинотеатр
* Название
* Количество мест
* Схема зала

После добавления зал появляется в списках доступных залов данного кинотеатра.

1. Добавление сеанса

Функция доступна для администраторов и кинотеатров.

Добавляет новый сеанс.

* Название фильма
* Зал
* Дата, время

1. Поиск пользователей и изменение их данных

Функция, доступная для администратора.

Осуществляет поиск пользователей по логину,

Если пользователь найден, то возможно редактирование его данных или перемещение в другую группу пользователей.

1. Выбор фильма по времени и кинотеатру

Во вкладке «Киноафиша» возможен поиск по дате, можно выбрать интересующую дату для конкретного фильма, либо фильмы в конкретном кинотеатре.

1. Возврат билета

Пользователь может вернуть билет/отменить бронь.

При нажатии на кнопку бронь аннулируется, а место становится свободным.

Если билет возвращают в кинотеатр, то есть «возврат билета» для билетера. Билетер может регулировать состояние мест в зале, поэтому при возврате билета, он ставит место как «свободное»

* 1. **Функциональное тестирование**

Функция №1: «Регистрация»

1. Длина почты должна быть больше 8 символов.

Правильный класс эквивалентности: 4 < |e-mail| < 45

**Вход:** admin@mail.ru

**Выход:** admin@mail.ru

Неправильный класс эквивалентности: |e-mail| < 8|

**Вход:** p@nda.ru

**Выход:** неверные данные, электронная почта должна включать больше 8 символов.

1. Тип номера телефона

**Вход**: +7 923 123 21

**Выход:** не соответствует формату

Аналогичные тестовые ситуации проектируются для остальных пользовательских функций.

**Глава 5. Проектирование БД**

**5.1 Введение**

Веб-сайт для организации работы с кинотеатрами – это многопользовательский веб-интерфейс, работающий с большими объемами данных. Для успешной работы требуется разработать СУБД, основанную на анализе предметной области и проектировании ПС. Система должна включать в себя все объекты предметной области, учитывать взаимосвязи между ними. В связи с большим потоком пользователей и ограниченных аппаратных средствах, СУБД должна быть быстрой и безопасной в использовании.

**5.2 ER-диаграмма, демонстрирующая структуру основной БД**

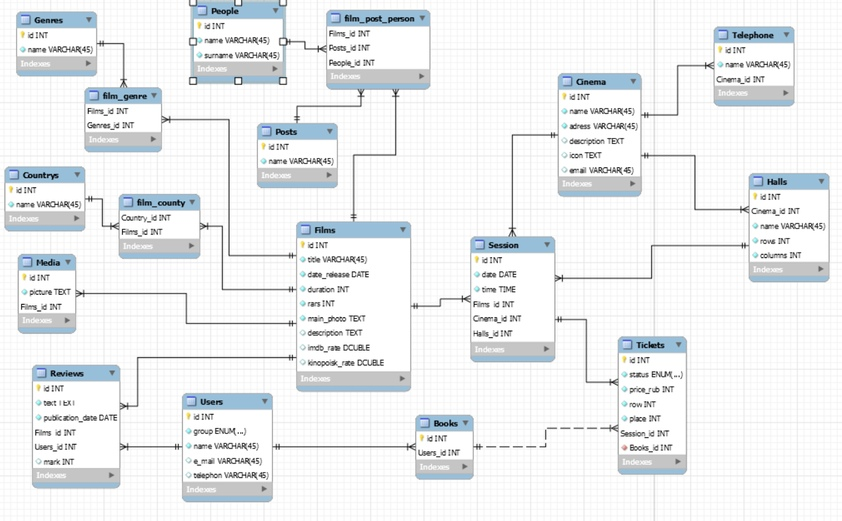
****

Диаграмма состоит из:

1. Таблица «Films»

Содержит в себе основную информацию о фильме

* Название
* Дата
* Длительность
* Описание
* Ссылку на постер
* Рейтинг фильма

1. Таблица «Media»

Содержит изображения для фильма (название изображения)

Содержит в себе Foreign Key, который ссылается на табличку «Films»

1. Таблица «Film\_genre»

У фильма может быть насколько жанров.

Содержит информацию о жанре фильма

Содержит в себе Foreign Key, который ссылается на табличку «Films» и id жанра

1. Таблица «Genres»

Содержит название жанра и id, которому этот жанр принадлежит

1. Таблица «Reviews»

Содержит информацию об отзывах:

* На каком фильме
* Когда был сделан
* Кем сделан
* Текст отзыва

1. Таблица «Users»

Содержит в себе:

* id и имя пользователя
* email
* телефон
* к какой группе принадлежит

1. Таблица «Session»

Содержит информацию о:

* Фильме
* Кинотеатре
* Времени сеанса
* Количестве билетов
* Информация о зале

1. Таблица «Tickets»

Содержит информацию о:

* Статусе билета (свободен, забронирован, куплен)
* Цена
* Ряд
* Место
* Id сеанса, которому он принадлежит
* Забронирован билет или нет

1. Таблица «Cinema»

Содержит информацию о метаположении кинотеатра

1. Таблица «Halls»

Информация о залах, привязанная к определенному кинотеатру

1. Таблица «Telephone»

Номер телефона конкретного кинотеатра

1. Таблица «Books»

Информация о забронированном билете,

Id билета и id пользователя

1. Таблица «Film\_post\_person»

Содержит в себе id людей, занимающих особое место при создании фильма (режиссеры, сценаристы и т.д.)

1. Таблица «Post»

Информация о названии роли, занимающей при создании фильма (сценарист, режиссер и т.д.)

1. Таблица «People»

Информация о персонале (режиссерах, сценаристах и т.д.) – ФИО

1. Таблица «Film\_country»

Содержит id фильма и id страны, где снимался фильм

1. Таблица «Counties»

Информация о странах по id

**5.3 Выбор СУБД**

При решении данной задачи СУБД должна удовлетворять требованиям:

* Поддержка многопользовательского использования.
* Возможность управления распределенными БД.
* Возможность реализации хранимых процедур.
* Возможность добавления новых объектов (таблиц).
* Возможность добавления триггеров
* Кроссплатформенность
* Возможность добавлять транзакции

**Список претендентов:**

MYSQL

1. MySQL может использоваться в качестве сервера.
2. Возможность реализации хранимых процедур.
3. Удовлетворяет поддержке многопользовательского использования.
4. Возможность добавления триггеров.
5. Взаимодействие с моделями и ER диаграммами.
6. Кроссплатформенность.

PostgreSQL

1. Кроссплатформенность.
2. Может использоваться в качестве сервера.
3. Удовлетворяет поддержке многопользовательского использования.
4. Возможность реализации хранимых процедур.

Проведя сравнительный анализ, можно сделать вывод, что наиболее подходящей СУБД является MySQL. Данная СУБД способна обеспечивать эффективную работу с БД при необходимом количестве записей, СУБД кроссплатформенная, присутствует возможность расширения и реализации хранимых процедур и триггеров. Трудовые затраты и интегрирование БД с языком программирования Java более низкие, чем с PostgreSQL.

**5.4 Заключение**

На основе данных, полученных при анализе предметной области, было сформировано четкое представление БД, необходимой для создания веб-сайта, взаимодействующего с кинотеатрами. Было составлено описание объектов предметной области. Были установлены взаимосвязи между объектами. В качестве СУБД была выбрана бесплатная СУБД MySQL, успешно интегрируемая с языком реализации веб-сайта Java. Таким образом, был создан проект БД для веб-сайта кино.

**Заключение**

Таким образом, в рамках курсового проекта были выполнены все поставленные задачи. Был проведен обзор существующих веб-сайтов, взаимодействующих с кинотеатрами, на основе которого была сформирована абстрактная модель разрабатываемого ПС.

На основе анализа предметной области были спроектированы взаимосвязи системы с внешними и внутренними факторами. Взаимосвязи были отражены при помощи диаграмм, каждая из которых показывала определенную точку зрения пользователя и разработчика на проект.

Диаграмма прецедентов выявила все пользовательские функции, User Flow диаграмма показала, как пользователь будет взаимодействовать с веб-сайтом.

На их основе было построено функциональное тестирование программы.

Теперь перед нами стоит задача реализовать веб-сайт на основе созданной документации, используя язык Java с использованием фреймворка Spring, библиотекой Hibernate и средствами вёрстки (HTML, CS, JS)