МИНИСТЕРСТВО ОБРАЗОВАНИЯ и науки РЕСПУБЛИКИ БАШКОРТОСТАН

Государственное автономное ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

УФИМСКИЙ КОЛЛЕДЖ СТАТИСТИКИ, ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЧЕТ ПО ИНДИВИДУАЛЬНОМУ ПРОЕКТУ

На тему «Сервисный центр по ремонту компьютерной техники»

По УП 09.01 Учебная практика

Выполнил студент группы 20ВЕБ1

Рахимов Альфир Фаризович

Проверил(а) преподаватель информатики

и программирования

Дмитриева Елизавета Константиновн

2023

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
|  | лист |
| 1 Описание предметной области | 3 |
| 2 Описание входной информации | 4 |
| 3 Описание выходной информации  4 UML диаграммы | 4  5 |
| 5 Логическое моделирование | 5 |
| 6 Описание структуры базы данных | 6 |
| 7 Контрольный пример | 11 |
| 8 Общие требования к программному продукту | 13 |
| 9 Карта сайта | 16 |
| 10 Физическая схема | 17 |
| 11 Описание модулей | 18 |
| 12 Протоколы тестирования  13 Руководство пользователя | 19  31 |
| Приложение А. UML диаграммы | 36 |
| Приложение Б. Даталогическая модель БД  Приложение В. Физическая схема сайта | 37  38 |
| Список литературы | 41 |

1. Описание предметной области

Требуется разработать веб-сервис «Афиша».

Сервис «Экспресс Афиша» это веб-платформа, сервис, предназначенный для организации, управления и распространения разнообразных событий в цифровом формате.

В системе предусмотрены роли:

* пользователь;
* администратор.

При входе на сайт клиент увидит главную страницу, на которой находятся популярные события и жанры, о которых известно:

* название;
* дата проведения;
* время проведения;
* место проведения;
* имя артиста;

Для того чтобы добавить событие в избранное необходимо пройти регистрацию, в которой требуется указать:

* имя;
* фамилия;
* отчество;
* e-mail;
* номер телефона;
* пароль;
* дата рождения;
* фото;
* пол;

Пользователь имеет следующие возможности:

* купить билет на событие;
* добавить событие в избранное;
* редактировать свой аккаунт;
* сделать возврат билета.

Администратор имеет следующие функции:

* добавлять, редактировать, удалять события;
* добавлять, редактировать, удалять жанры;
* добавлять, редактировать, удалять поджанры;
* добавлять, редактировать, удалять исполнителей;
* добавлять, редактировать, удалять отношения между событием и исполнителем;
* добавлять, редактировать, удалять места проведения;
* добавлять, редактировать, удалять города.

Ограничения системы:

- Незарегистрированный пользователь, не может добавить в избранное событие;

1. Описание входной информации

Входным документом сервис «Экспресс Афиша» является описание мероприятий:

* название;
* дата проведения;
* время проведения;
* место проведения;
* описание исполнителя.

1. Описание выходной информации

Описание выходных документов представлено на таблице 1.3.1.

Таблица 1.3.1 – Описание выходных документов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование документа (шифр) | Периодичность выдачи документа | Кол-во экз. | Куда передаются | Итоги |
| Электронный билет | При покупке билета на событие | Не ограничено | Пользователь | Электронный билет |

Форма выходного документа «Электронный билет» представлена ниже:

* название события;
* дата и время проведения;
* место проведения.

1. Use Case диаграмма (диаграмма прецедентов)

Диаграмма прецедентов – это диаграмма, которая позволяет визуализировать различные типы ролей и их взаимодействия с системой.

В данной диаграмме существует 4 объекта:

* актер;
* случай использования;
* система;
* пакет.

Актер в диаграмме прецедентов – это любая сущность, которая взаимодействует с системой. Это может быть человек, организация или внешняя система.

Случай использования представляет собой действие внутри системы.

Система используется для определения сферы применения.

Пакет является дополнительным параметром, который используется в сложных диаграммах. Пакеты используются для группировки большого количества случаев использования.

Каждая группа пользователей на диаграмме вариантов использования изображается человечком.

Диаграмма прецедентов представлена в приложении А, рисунок А.1

1. Логическое (даталогическое) проектирование

Под даталогической понимается модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации.

Даталогическая модель представлена в приложении Б, рисунок Б.1

1. Описание структуры базы данных

Описание структуры базы данных представлена в таблицах 6.1 – 6.14

Таблица 6.1 – «Age\_limits» (ограничение возраста)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| limit | Ограничение возраста | INT | 2 |  |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.2 – «Cities» (города)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| name | Название города | VARCHAR | 255 |  |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.3 –«Events» (события)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| name | Название события | VARCHAR | 255 |  |
| description | Описание события | TEXT | 1000 |  |
| image | Картинка события | VARCHAR | 255 |  |
| date | Дата проведения | DATE | 20 |  |
| time | Время проведения | TIME | 20 |  |
| age\_limit\_id | Вторичный ключ возраста | INT | 4 | FK |
| subgenre\_id | Вторичный ключ поджанра | INT | 4 | FK |
| place\_id | Вторичный ключ места | INT | 4 | FK |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.4 –«Events-Performers» (отношение между событием и исполнителем)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| event\_id | Вторичный ключ события | INT | 4 | FK |
| performer\_id | Вторичный ключ исполнителя | INT | 4 | FK |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.5 –«Favorites» (избранные события)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| event\_id | Вторичный ключ события | INT | 4 | FK |
| user\_id | Вторичный ключ пользователя | INT | 4 | FK |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.6 –«Genders» (Полы)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| name | Название пола | VARCHAR | 255 |  |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.7 –«Genres» (Жанры)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| name | Название жанра | VARCHAR | 255 |  |
| image | Картинка жанра | VARCHAR | 255 |  |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.8 –«Performers» (исполнители)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| name | Имя исполнителя | VARCHAR | 255 |  |
| description | Описание исполнителя | TEXT | 1000 |  |
| photo | Картинка исполнителя | VARCHAR | 255 |  |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.9 –«Places» (места проведения)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| name | Название места | VARCHAR | 255 |  |
| address | Адрес | VARCHAR | 255 |  |
| image | Картинка места | VARCHAR | 255 |  |
| scheme | Картинка схемы | VARCHAR | 255 |  |
| description | Описание места | TEXT | 1000 |  |
| city\_id | Вторичный ключ города | INT | 4 | FK |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.10 –«Roles» (роли)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| name | Название роли | VARCHAR | 255 |  |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.11 –«Subgenres» (Поджанры)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| name | Название поджанра | VARCHAR | 255 |  |
| genre\_id | Вторичный ключ жанра | INT | 4 | FK |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.12 –«Tickets» (Билеты)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| user\_id | Вторичный ключ пользователя | INT | 4 | FK |
| event\_id | Вторичный ключ события | INT | 4 | FK |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.13 –«Tickets» (Билеты)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| user\_id | Вторичный ключ пользователя | INT | 4 | FK |
| event\_id | Вторичный ключ события | INT | 4 | FK |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.13 –«Users» (пользователи)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| name | Имя | VARCHAR | 255 |  |
| surname | Фамилия | VARCHAR | 255 |  |
| patronymic | Отчество | VARCHAR | 255 |  |
| email | Электронная почта | VARCHAR | 255 |  |
| phone | Номер телефона | VARCHAR | 255 |  |
| birthday | Дата рождения | DATE | 18 |  |
| photo | Фото пользователя | VARCHAR | 255 |  |
| gender\_id | Вторичный ключ пола | INT | 4 | FK |
| role\_id | Вторичный ключ роли | INT | 4 | FK |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

Таблица 6.14 –«Watched» (просмотренные)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| 1 | 2 | 3 | 4 | 5 |
| id | Первичный ключ | INT | 4 | PK |
| event\_id | Вторичный ключ события | INT | 4 | FK |
| user\_id | Вторичный ключ пользователя | INT | 4 | FK |
| created\_at | Время создания | TIMESTAMP | 20 |  |
| updated\_at | Время обновления | TIMESTAMP | 20 |  |

1. Контрольный пример

Контрольный пример представлен в таблицах 8.1 –8.14

Таблица 8.1 – Таблица «age\_limits» (возрастное ограничение)

|  |
| --- |
| Лимит |
| 0 |
| 6 |
| 12 |

Таблица 8.2 – Таблица «Cities» (города)

|  |
| --- |
| Название города |
| Уфа |
| Москва |
| Казань |

Таблица 8.3 – Таблица «Events» (события)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название события | Описание события | Картинка события | Дата проведения | Время проведения |
| HammAli & Navai | Хаммали лучший | hamm.png | 23.04.2023 | 16:31 |
| Aidar | Лучший Айдар | aidar.png | 13.05.2023 | 16:33 |
| Уфа - Адмирал | Адмирал проиграют | ufadm.png | 05.12.2023 | 16:32 |

Таблица 8.4 – Таблица «Genders» (полы)

|  |
| --- |
| Название пола |
| Мужской |
| Женский |

Таблица 8.5 – Таблица «Genres» (жанры)

|  |  |
| --- | --- |
| Название жанра | Картинка жанра |
| Концерты | concert.png |
| Театр | teatr.png |
| Кино | kino.png |

Таблица 8.6 – Таблица «Performers» (исполнители)

|  |  |  |
| --- | --- | --- |
| Имя | Описание | Фото |
| Полина Гагарина | Лучшая в мире | jenchina.png |
| Aidar | Айдарус нотариус | gulyash.png |
| ARSLANUS | Асралнус лучший из худших | heisenberg.png |

Таблица 8.7 – Таблица «Places» (места проведения)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Адрес | Картинка | Схема | Описание |
| Тинькофф Холл | ул. 50 лет октября | jenchina.png | gulyash.png | Тинькофф построил сам |
| Korston | ул. 51 лет октября | gulyash.png | heisenberg.png | В казани |
| Уфа-Арена | ул. 52 лет октября | heisenberg.png | jenchina.png | Арена гладиаторская |

Таблица 8.9 – Таблица «Roles» (роли)

|  |
| --- |
| Название |
| Администратор |
| Гость |

Таблица 8.10 – Таблица «Subgenres» (поджанры)

|  |
| --- |
| Название |
| ХИП-ХОП РЭП |
| Опера |
| Комедия |

Таблица 8.11 – Таблица «Users» (пользователь)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя | Фамилия | Пароль | Фото | Электронная почта | Дата рождения |
| Альфир | Рахимов | parol | foto.png | anton@gmail.com | 23.12.12 |
| Кирилл | Фамилиевич | kirill | fotograf.png | ru@mail.ru | 22.04.14 |
| Руслан | Русланов | ruskiy | jizn.png | com@com.com | 13.10.11 |

1. Общие требования к программному продукту
   1. Введение
2. Краткое описание проекта

«Экспресс Афиша» предназначена для оформления билета на развлекательное событие.

Основным назначением «Экспресс Афиша» является организация и управление событиями.

1. Цели и задачи разработки

«Экспресс Афиша» создается с целью:

- предоставления услуг оформления билетов на события;

- повышения качества (полноты, точности, достоверности, своевременности, согласованности) информации;

1. Ссылки на связанные документы или проекты

https://github.com/Alfir6719/afisha

* 1. Функциональные требования

1. Подробное описание функциональности, которую должен предоставлять программный продукт.
2. Входы и выходы системы.

Описание мероприятия, электронный билет.

1. Взаимодействие с другими системами.

Отсутствует

* 1. Требования к интерфейсу

1. Описание пользовательского интерфейса.

- Главная страница:

- Новые события (слайдер);

- События в ближайшие дни;

- Топ событий;

- Ожидаемые события;

- Жанры;

- Просмотренные события.

- Страница отдельного жанра:

- новые события по жанру;

- события по жанру.

- Страница события:

- Баннер события;

- Схема события;

- Описание события;

- Исполнители;

- Адрес;

- Просмотренные события.

- Страница с моими билетами:

- Купленные билеты под определенные события.

- Страница с избранными событиями:

- Избранные события.

- Страница с просмотренными событиями:

- Просмотренные события.

- Страница с данными о моем аккаунте:

- поля для редактирования аккаунта;

- кнопка для удаления аккаунта.

1. Требования к навигации и взаимодействию пользователя с системой.

- Навигационное меню:

- Фиксированное меню для быстрого доступа ко всем разделам (топ-10, концерты, театр, кино, стендап, спорт, мои билеты, мой аккаунт, регистрация, авторизация).

* 1. Требования к производительности

1. Ожидаемые показатели производительности.

Пропускная способность: 100;

Время ответа сервера: 2мс;

Время отображения: 2мс.

1. Требования к загрузке системы при различных нагрузках.

Запросов в секунду: не более 1000 в секунду;

Пользователей в системе:

Активных – 4 человек.

Зарегистрированных – 100 человек.

Одновременно работающих – 4 человек.

* 1. Требования к безопасности

1. Требования к управлению доступом и аутентификации.

Доступ к учетным записям администратора, мастера, менеджера и клиента разграничивается c помощью механизма Laravel – Middleware.

1. Меры безопасности для защиты данных.

Пароли пользователей автоматически хешируются при регистрации.

1. Требования к обеспечению конфиденциальности.

Требования к обеспечению конфиденциальности основываются на: Закон РФ от 27.12.1991 N 2124-1(ред. от 13.06.2023)»О средствах массовой информации»

1. Карта сайта

HTML карта – это статическая веб-страница, которая включает все основные и подстраницы вашего сайта в соответствии с их иерархией. Проще говоря, это что-то вроде оглавления книги, которое разработано для сайта, чтобы помочь посетителям легко находить страницы на вашем ресурсе.

Карта сайта представлена на рисунке 1.10.1.

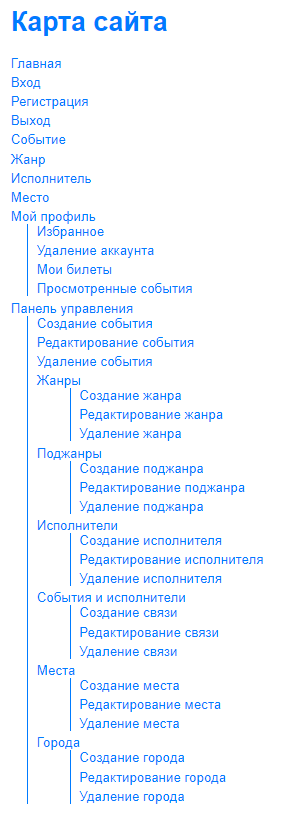


Рисунок 10.1 – Карта сайта

1. Физическая схема

Схема базы данных — это структура и организация базы данных, которая определяет ее таблицы, поля, связи, ограничения и типы данных. Он служит образцом для организации данных и доступа к ним, предоставляя план действий для разработчиков, администраторов и пользователей при работе с базой данных. Схема необходима для эффективного и действенного управления данными, что приводит к повышению производительности базы данных и упрощению обслуживания.

Физическая схема базы данных представлена в приложении В.

1. Описание модулей

При запуске программного продукта выполняется проект «AMechanic», после которого выполняются следующие процедуры, которые представлены в таблице 12.1.

Таблица 12.1 – Описание модулей

|  |  |  |
| --- | --- | --- |
| 1. Модуль | Метод | Назначение |
| AdminController.php | index | Открытие главной страницы администратора |
| create (event, genre, place…) | Добавление чего либо |
| delete (event, genre, place…) | Удаление чего либо |
| update (event, genre, place…) | Редактирование чего либо |

Продолжение таблицы 2.1.1– Описание файлов

|  |  |  |
| --- | --- | --- |
| All controllers | Index and show | Вывод чего либо на странице |
| Favorite Controller | likeEvent | Лайкнуть событие |
| dislikeEvent | Убрать лайк с события |
| UserController | registration | Регистрация пользователя |
| login | Авторизация пользователя |
| logout | Выход из аккаунта |
| updateUser | Редактирование информации пользователя |

1. Протоколы тестирования

Тестирование программного продукта является неотъемлемой частью разработки системы. Его проводят для достижения наилучшей работоспособности и исправления ряда ошибок, которые могут возникнуть при взаимодействии с системой.

Обычно тестирование ПО проводится на всех этапах жизненного цикла: с момента разработки ПО до его запуска в эксплуатацию.

Цель тестирования - проверка соответствия ПО предъявляемым требованиям, обеспечение уверенности в качестве ПО, поиск очевидных ошибок в программном обеспечении.

Для выявления ошибок в работе системы были проведены следующие тесты.

Общая информация о тестировании приведена в таблице 13.1.

Таблица 13.1 – Общая информация о тестировании

|  |  |
| --- | --- |
| Общая информация о тестировании | |
| Название теста | Tests\_Service |
| Номер версии | 1.0 |
| Имя тестера | Муллагалеев Булат |
| Дата тестирования | 12.12.2023 |

Тестирование авторизации при вводе некорректных данных приведено в таблице 13.2.

Таблица 13.2 - Тестирование авторизации при вводе некорректных данных

|  |  |  |
| --- | --- | --- |
| Модуль | Метод | Назначение |
| AdminController.php | index | Открытие главной страницы администратора |
| create (event, genre, place…) | Добавление чего либо |
| delete (event, genre, place…) | Удаление чего либо |
| update (event, genre, place…) | Редактирование чего либо |

Продолжение таблицы 2.1.1– Описание файлов

|  |  |  |
| --- | --- | --- |
| All controllers | Index and show | Вывод чего либо на странице |
| Favorite Controller | likeEvent | Лайкнуть событие |
| dislikeEvent | Убрать лайк с события |
| UserController | registration | Регистрация пользователя |
| login | Авторизация пользователя |
| logout | Выход из аккаунта |
| updateUser | Редактирование информации пользователя |

* 1. Тестирование программного продукта

Цель тестирования – проверка соответствия программного обеспечения к предъявляемым требованиям, обеспечение уверенности в качестве программного обеспечения, поиск ошибок.

Для выявления ошибок были проведены тесты.

Таблица 6.1.1 – Общая информация о тестировании

|  |  |
| --- | --- |
| Общая информация о тестировании | |
| Название теста | Tests\_Afisha |
| Номер версии | 1 |
| Имя тестера | Рахимов Альфир |
| Дата тестирования | 19.12.2023 |

Таблица 6.1.2 - Тестирование регистрации при вводе верных данных

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | T\_REGISTRATION\_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка регистрации с верными пользовательскими данными |
| Резюме испытания | Необходимо добиться корректного поведения системы, при верных данных при регистрации пользователя |
| Шаги тестирования | 1. Открыть модульное окно регистрации пользователя; 2. Ввести в поля «Имя», «Фамилия», «Номер телефона», «Электронная почта», «Пароль», «Подтверждение пароля» данные; 3. Нажать на кнопку «Создать аккаунт». |
| Данные тестирования | Имя: Кирилл;  Фамилия: Тихонов;  Номер телефона: 8 800 555 35 35;  Электронная почта: example@mail.ru;  Пароль: password;  Подтверждение пароля: password. |

Продолжение таблицы 6.1.2 – тестирование регистрации при вводе верных данных

|  |  |
| --- | --- |
| 1 | 2 |
| Ожидаемый результат | Сайт должен вывести сообщение о верном вводе данных |
| Фактический результат | Вывелось сообщение «Успешная регистрация»  Результат работы показан на рисунке 6.1.1 |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

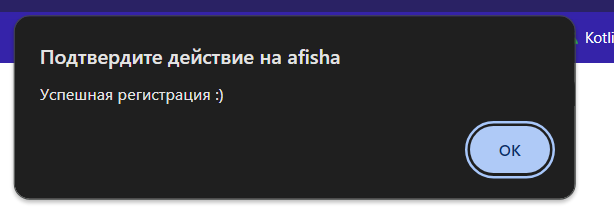


Рисунок 6.1.1 – Результат работы регистрации пользователя при вводе неверных данных

Таблица 6.1.3 - тестирование регистрации при вводе неверных данных

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | T\_ REGISTRATION \_ERROR\_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка регистрации с неверными пользовательскими данными |
| Резюме испытания | Необходимо добиться корректного поведения системы, при неверных данных при регистрации пользователя |
| Шаги тестирования | 1. Открыть модульное окно регистрации пользователя; 2. Ввести в поля необходимы данные 3. Нажать на кнопку «Создать аккаунт». |

Продолжение таблицы 6.1.3 тестирование регистрации при вводе неверных данных

|  |  |
| --- | --- |
| 1 | 2 |
| Данные тестирования | Имя: NULL;  Фамилия: NULL;  Номер телефона: NULL;  Электронная почта: NULL;  Пароль: NULL;  Подтверждение пароля: NULL. |
| Ожидаемый результат | Сайт должен вывести сообщение о незаполненных полях |
| Фактический результат | Вывелось сообщение «Заполните поле»  Результат работы показан на рисунке 6.1.2 |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

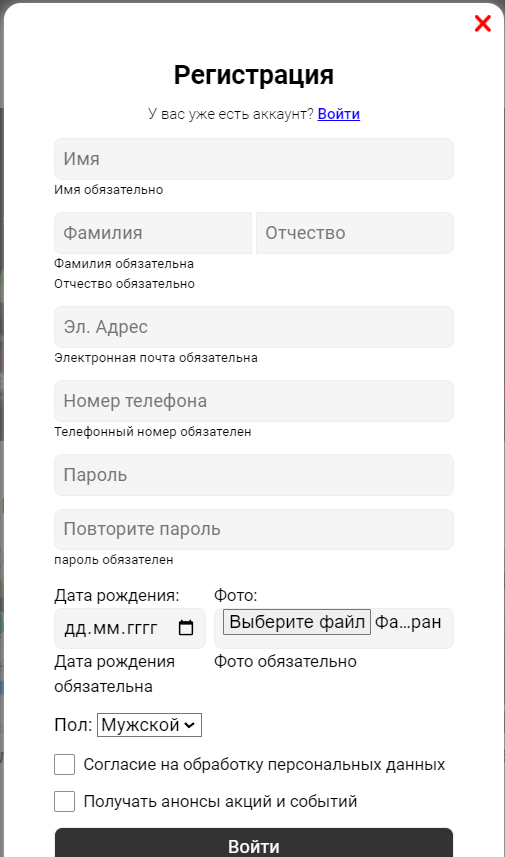


Рисунок 6.1.2 – Результат работы регистрации пользователя при вводе неверных данных

Таблица 6.1.4 - Тестирование авторизации пользователя

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | T\_AUTHORIZATION \_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка авторизации пользователя при вводе верных данных |
| Резюме испытания | Необходимо добиться корректного поведения системы, при авторизации пользователя |
| Шаги тестирования | 1. Нажать на кнопку «Вход»; 2. Ввести в поля данные; 3. Нажать на кнопку «Войти». |
| Данные тестирования | Телефонный номер: 89659244147;  Пароль: password; |
| Ожидаемый результат | Авторизация |
| Фактический результат | Авторизация |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

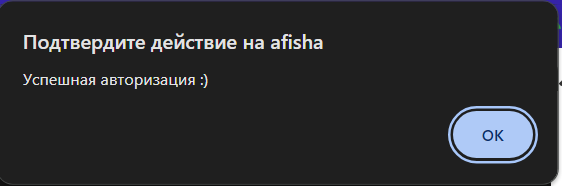


Рисунок 6.1.3 – Результат работы авторизации пользователя при вводе верных данных

Таблица 6.1.5 - тестирование авторизации при вводе неверных данных.

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | T\_ AUTHORIZATION \_ERROR\_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка авторизации пользователя при вводе неверных данных |

Продолжение таблицы 6.1.5 тестирование авторизации при вводе неверных данных

|  |  |
| --- | --- |
| 1 | 2 |
| Резюме испытания | Необходимо добиться корректного поведения системы, при авторизации пользователя |
| Шаги тестирования | 1. Нажать на кнопку «Вход»; 2. Ввести в поля данные; 3. Нажать на кнопку «Войти». |
| Данные тестирования | Телефонный: Null;  Пароль: Null; |
| Ожидаемый результат | Ошибка авторизация |
| Фактический результат | Ошибка авторизация |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

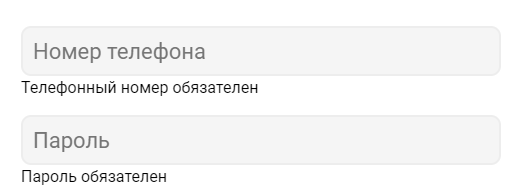


Рисунок 6.1.4 – Результат работы авторизации пользователя при вводе верных данных

Таблица 6.1.6 – Тестирование добавление поджанра.

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | T\_IZBRAN\_FORM \_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Добавление события в избранное |
| Резюме испытания | Необходимо добиться корректного поведения системы, при добавлении в избранное |
| Шаги тестирования | 1. Нажать на кнопку «Поджанры» 2. Заполнить форму |
| Данные тестирования | Название: Теннис;  Жанр:Спорт. |

Продолжение таблицы 6.1.6 тестирование отправления заявки на артиста

|  |  |
| --- | --- |
| 1 | 2 |
| Ожидаемый результат | Поджанр создан |
| Фактический результат | Поджанр создан |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

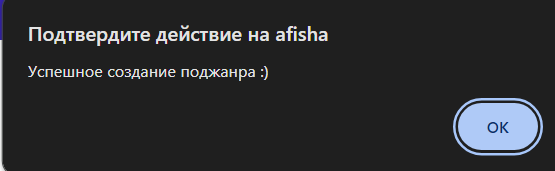


Рисунок 6.1.5 – Результат работы создания поджанра

Таблица 6.1.7 - Тестирование создания жанра при неправильном вводе данных.

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | T\_ PODJANR \_FORM\_ERROR \_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка отправления заявки на Артиста |
| Резюме испытания | Необходимо добиться корректного поведения системы, при отправлении неправильной заявки |
| Шаги тестирования | 1. Нажать на кнопку «Поджанры» 2. Заполнить форму |
| Данные тестирования | Название: Null;  Жанр: Null. |
| Ожидаемый результат | Ошибка отправления |
| Фактический результат | Ошибка отправления |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

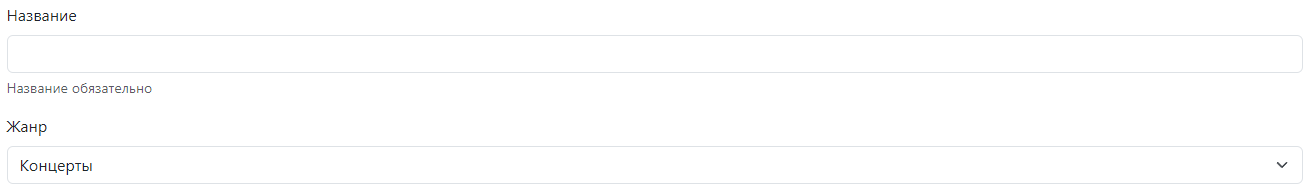


Рисунок 6.1.6 – Результат работы создания поджанра

Таблица 6.1.8 - Тестирование возврата билета

|  |  |
| --- | --- |
| 1 | 2 |
| Наименование | Описание |
| Test Case # | T\_DENY \_APPLICATION \_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка отклонения билета пользователя |
| Резюме испытания | Необходимо добиться корректного отклонении заявки пользователя админом |
| Шаги тестирования | 1. Зайти в аккаунт 2. На странице «Мои билеты» нажать «Возврат» |
| Данные тестирования | Номер телефона: 89659244147;  Пароль: pass |
| Ожидаемый результат | Билет возвращен |
| Фактический результат | Билет возвращен |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

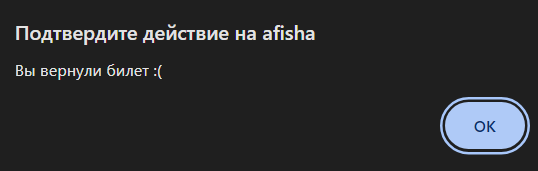


Рисунок 6.1.7 – Результат работы возврата билета

Таблица 6.1.9 – Тестирование удаления аккаунта

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | T\_ACCEPT \_ APPLICATION \_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка удаления аккаунта пользователем |

Продолжение таблицы 6.1.9 - тестирование удаления аккаунта

|  |  |
| --- | --- |
| 1 | 2 |
| Резюме испытания | Необходимо добиться корректного принятии заявки пользователя админом |
| Шаги тестирования | 1. Зайти в аккаунт 2. На странице «Мой аккаунт» нажать «Удалить аккаунт» |
| Данные тестирования | Номер телефона: 89659244147;  Пароль: pass |
| Ожидаемый результат | Аккаунт удален |
| Фактический результат | Аккаунт удален |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

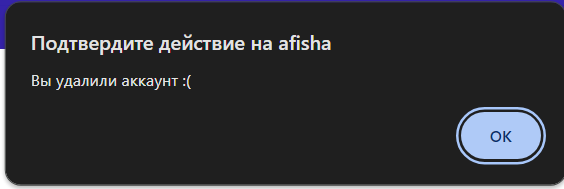


Рисунок 6.1.8 – Результат удаления аккаунта

Таблица 6.1.10 – Тестирование удаление артиста

|  |  |
| --- | --- |
| Наименование | Описание |
| 1 | 2 |
| Test Case # | T\_ARTIST \_DELETE \_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Удаление профиля артиста админом |
| Резюме испытания | Необходимо добиться корректного удаления артиста админом |
| Шаги тестирования | 1. Зайти в аккаунт админа 2. На странице «Исполнители» нажать «Удалить» |
| Данные тестирования | Номер телефона: 800000000;  Пароль: admin |

Продолжение таблицы 6.1.10 – тестирование удаления артиста

|  |  |
| --- | --- |
| Ожидаемый результат | Артист удалён |
| Фактический результат | Артист удалён |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

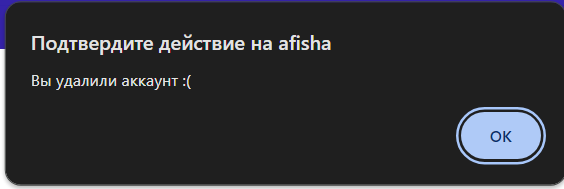


Рисунок 6.1.8 – Результат удаления артиста

1. Руководство пользователя

Главной целью руководства является обучение работы с системой.

Данный продукт предназначен для хранения, управления и распространения аудио-контента в цифровом формате.

Минимальные требования к аппаратному обеспечению:

* oc: Windows 8/8.1/10/11 (64 bit);
* процессор: Intel Core I3 10100 или AMD Ryzen 3 3200 (2.7 GHz);
* озу: 8 ГБ;
* видеокарта: UHD 630 или Vega 8 и выше;
* жёсткий диск: 30 ГБ свободного места и выше.

Для запуска и использования программного продукта, пользователю необходимы практические знания в области операционной системы Windows и базовые знания о работе с браузером.

Подготовка системы к работе:

* перейти во вкладку Мои проекты и выбрать необходимый проект;
* появится главная страница сайта, но на ней доступен функционал только для просмотра;
* необходимо зайти в систему (для администратора/пользователь) или зарегистрироваться (клиент).
* ввести логин и пароль в поля для авторизации (рисунок 15.1);

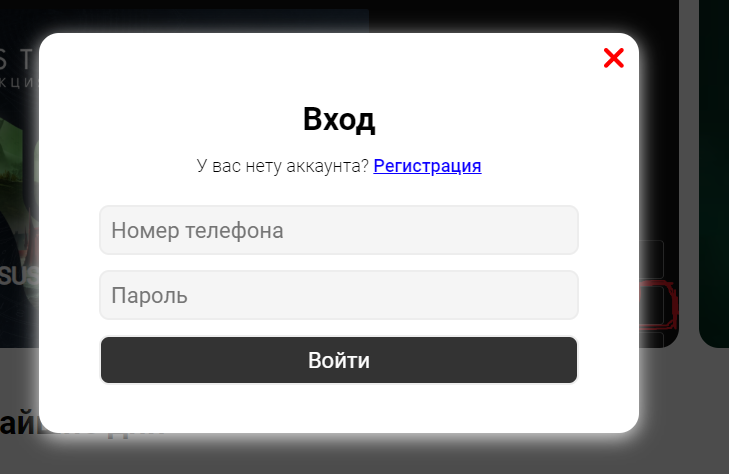


Рисунок 15.1 – Страница авторизации

* если же пользователя нет в системе, зарегистрироваться (рисунок 15.2);

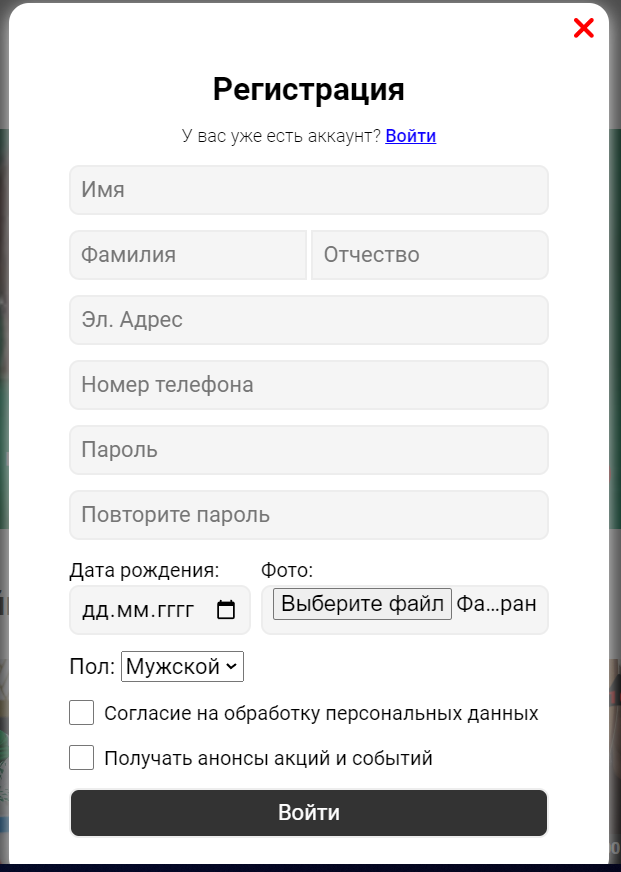


Рисунок 15.2 – Страница регистрации

Описание функций, которые доступны пользователю.

Добавление события в избранное:

Для добавления события в избранное, зарегистрированному пользователю на главной странице необходимо, выбрать понравившееся событие, нажав на кнопку «Лайк» (рисунок 15.3);

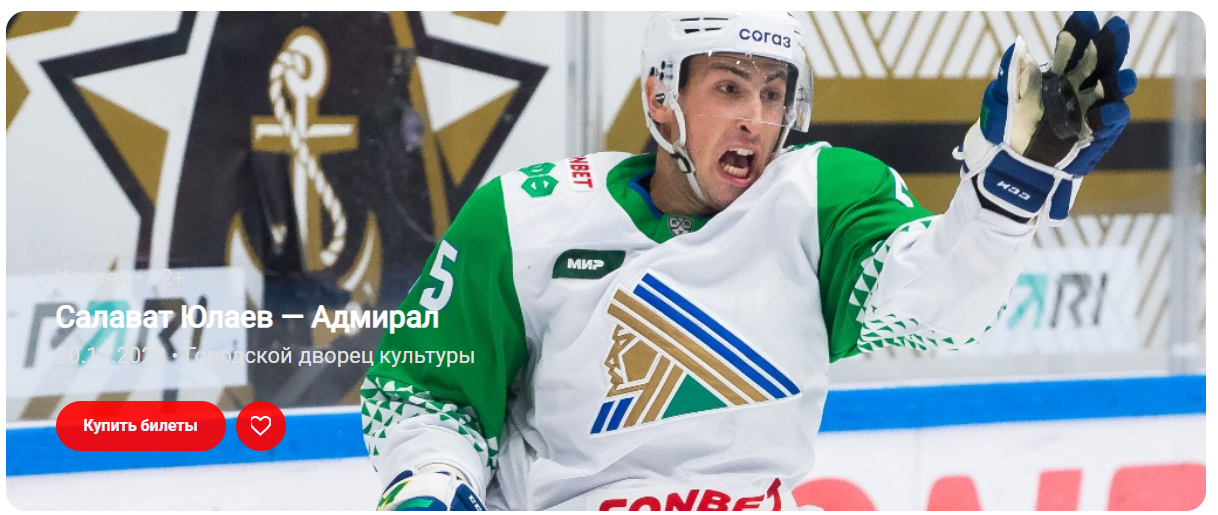


Рисунок 15.3 – Понравившееся событие

Чтобы перейти в «Избранное» необходимо перейти в шапку сайта и перейти в «Мой аккаунт» (рисунок 15.4);

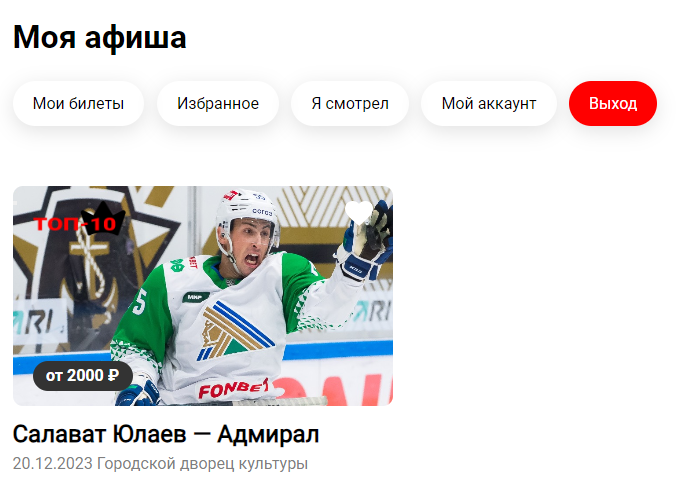


Рисунок 15.4 – Избранные

Для покупки билета, нужно перейти на событие и нажать на кнопку «Купить билет» (рисунок 15.5);

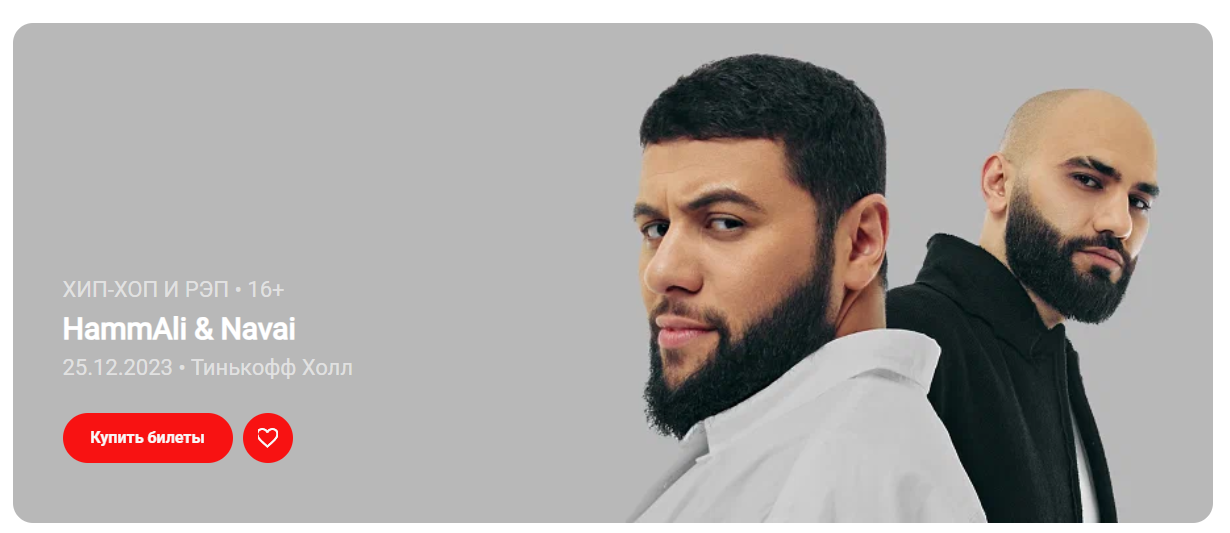


Рисунок 15.5 – Вкладка событие

Редактирование аккаунта:

Для управления аккаунтом нужно нажать на аватарку аккаунта (рисунок 15.6).

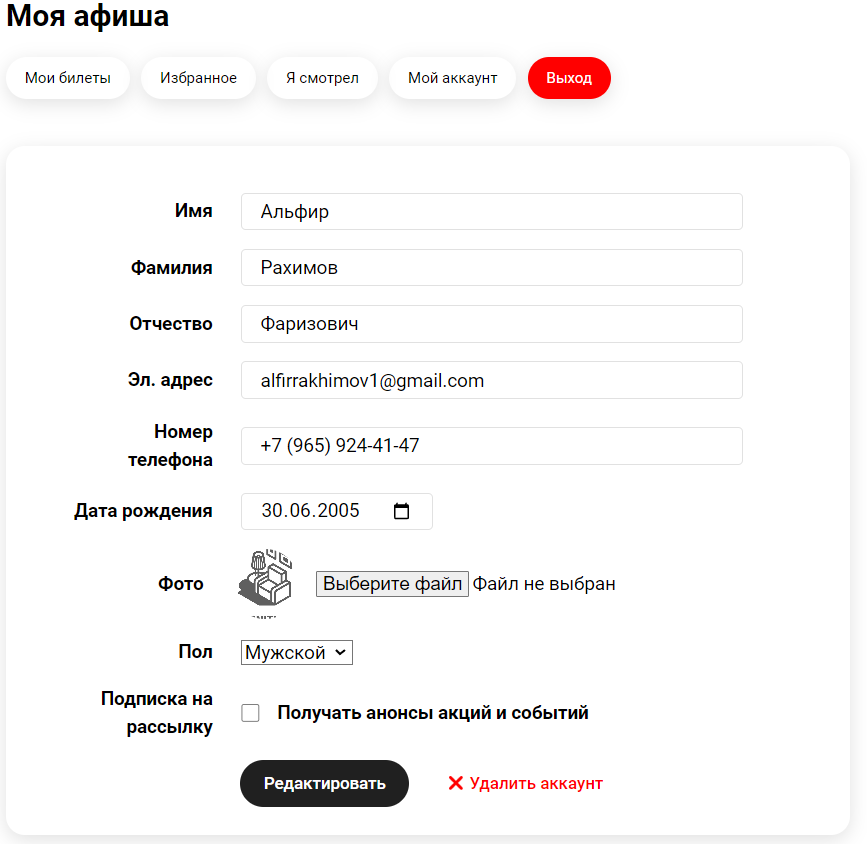


Рисунок 15.8 – Мой аккаунт

Описание функций, которые доступны администратору.

В личном кабинете администратор может добавлять, редактировать и удалять события, исполнителей, места, жанры, поджанры, связи между исполнителем и событием, города (рисунок 15.9).



Рисунок 15.9 – Личный кабинет администратора

Описание функций, которые доступны артисту.

Для редактирования данных нужно выбрать нужный пункт в навигации и далее выбирать редактирование, добавление или удаление в соответствии с усмотрением админа (рисунок 15.10);

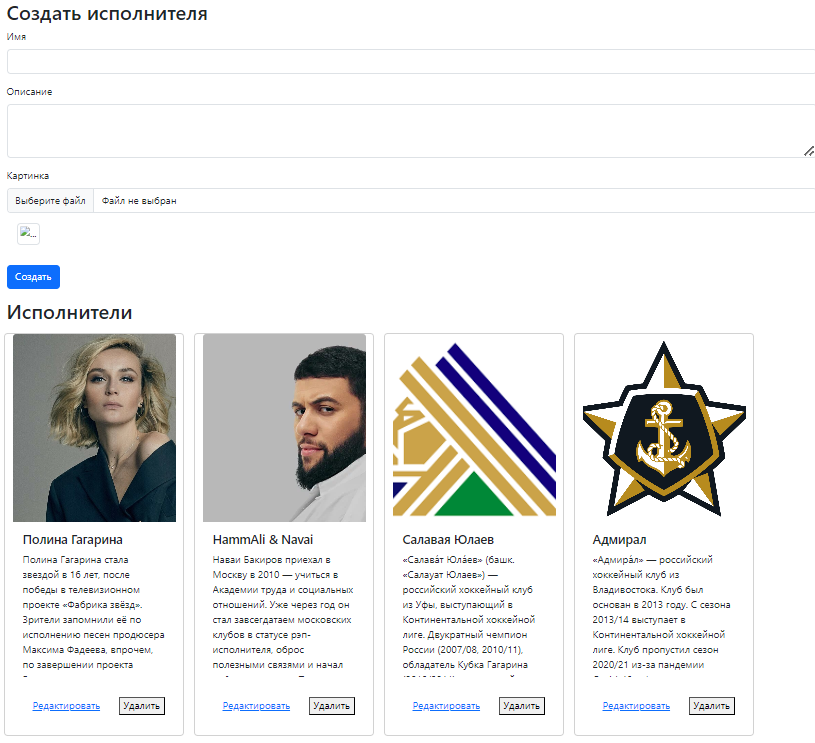


Рисунок 15.10 – Личный кабинет админа

Приложение А

UML диаграммы

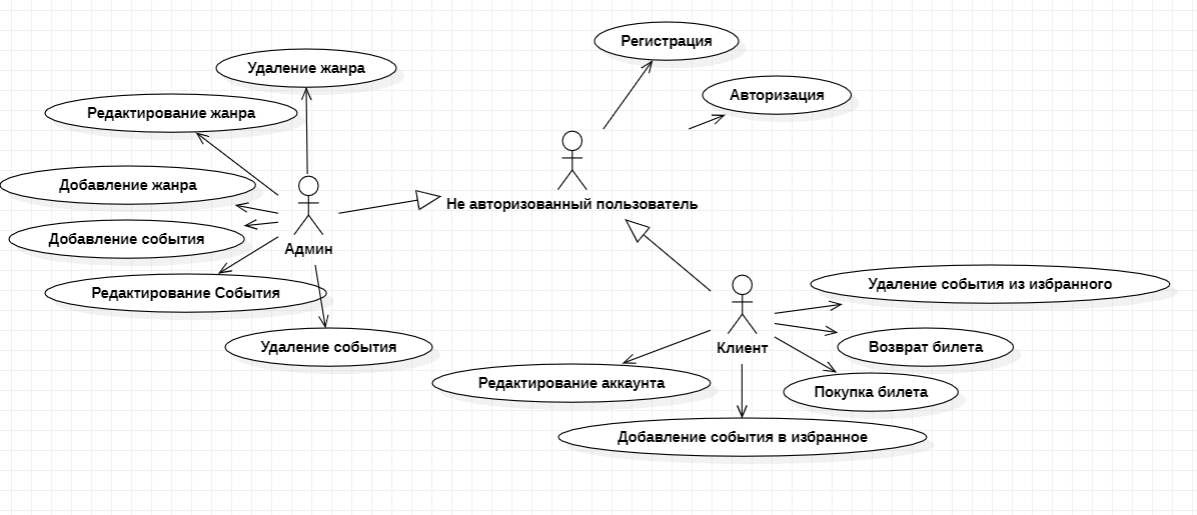


Рисунок А.1 – диаграмма прецедентов

Приложение Б

Даталогическая модель базы данных

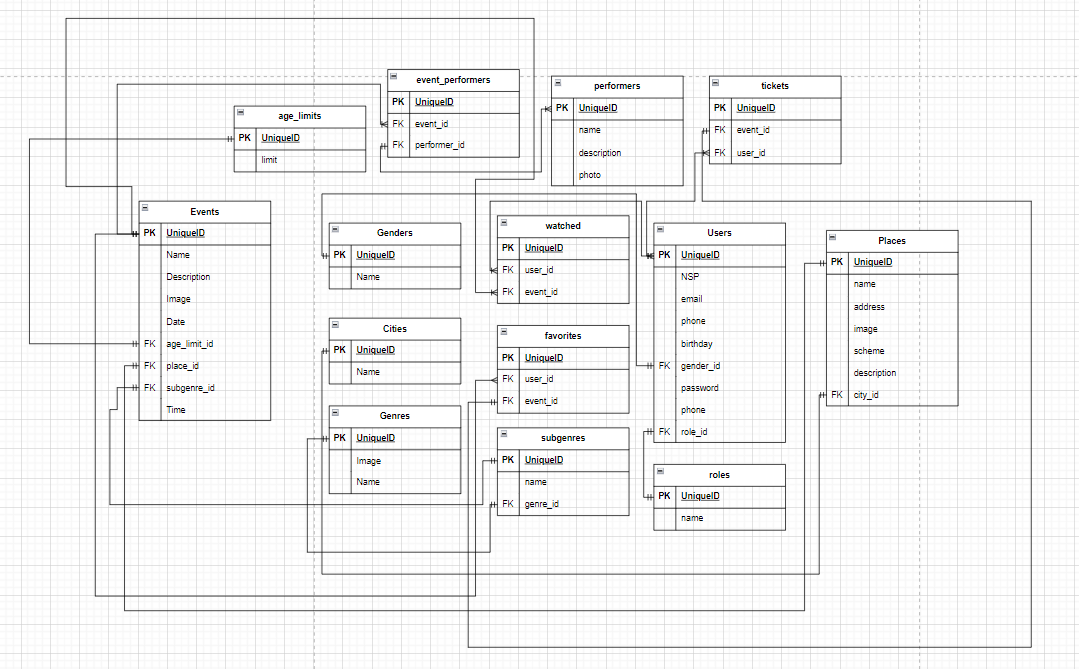


Рисунок Б.1 – даталогическая модель базы данных

Продолжение приложения А

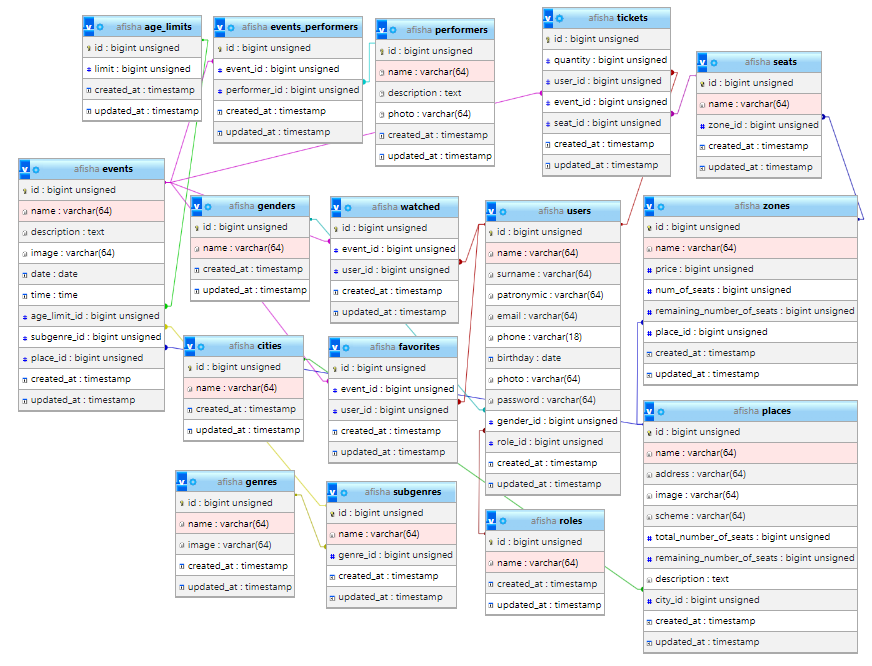


Рисунок Б.2 – даталогическая модель базы данных

Приложение В

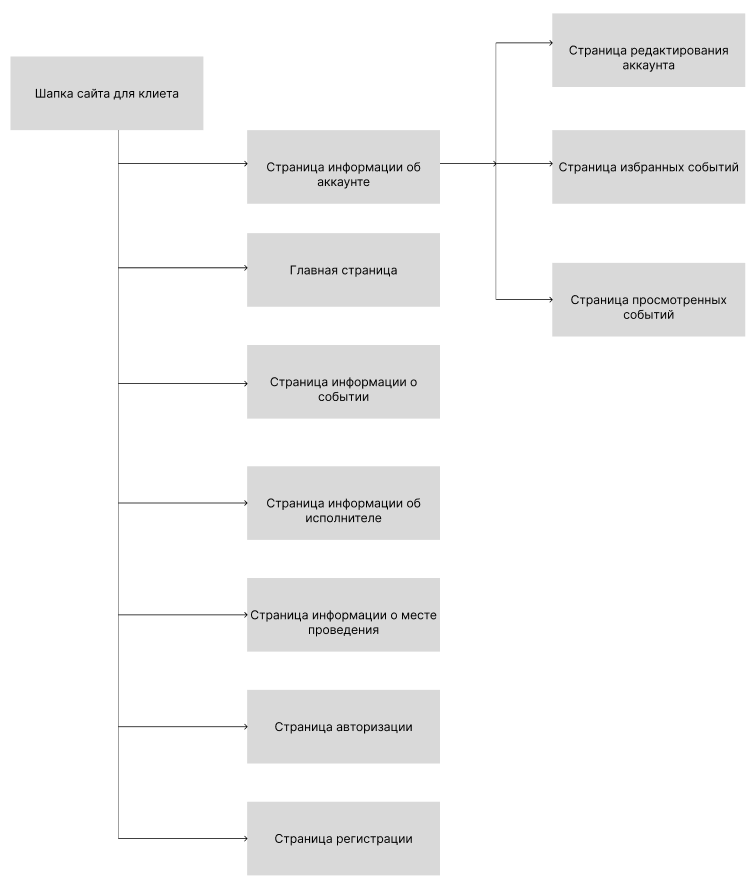


Рисунок В.1 – физическая модель БД для клиента

Продолжение приложения В



Рисунок В.2 – физическая модель БД для админа

СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ 19.001–77. ЕСПД. Общие положения.
2. ГОСТ 19.003–80. ЕСПД. Схемы алгоритмов и программ. Обозначения условные графические. – Заменен на ГОСТ 19.701-90.
3. Википедия: офиц. сайт. URL: http://ru.wikipedia.org/wiki/ (дата обращения: 3.09.2023).
4. Язык запросов SQL: офиц. сайт. URL: https://sql-language.ru (дата обращения: 15.09.2023).
5. Хабр: офиц. сайт. URL: https://habr.com/ru/all/ (дата обращения: 24.09.2023).
6. Htmlbook: офиц. сайт. URL: http://htmlbook.ru/ (дата обращения: 5.09.2023).
7. Html5 book: офиц. сайт. URL: http://html5book.ru/ (дата обращения: 5.09.2023).
8. Metanit: офиц. сайт. URL: https://metanit.com/ (дата обращения: 15.10.2023).
9. W3School: офиц. сайт. URL: https://www.w3schools.com/ (дата обращения: 28.10.2023).
10. itProger: офиц. сайт. URL: https://itproger.com/ (дата обращения: 03.11.2023).
11. Хекслет: офиц. сайт. URL: https://ru.hexlet.io/ (дата обращения: 04.11.2023).
12. Денвик: офиц. сайт. URL: https://denvic.ru/ (дата обращения: 21.11.2023).
13. JavaScript: офиц. сайт. URL: https://learn.javascript.ru/first-steps/ (дата обращения: 07.09.2023).
14. Bootstrap: офиц. сайт. URL: https://getbootstrap.com/ (дата обращения: 5.09.2023).
15. Laravel: офиц. сайт. URL: https://laravel.com/ (дата обращения: 5.09.2023).