

Федеральное государственное автономное образовательное учреждение
высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 25

ПРЕПОДАВАТЕЛЬ

доцент, к.т.н		Е.М.Линский
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ ПО КУРСОВОЙ РАБОТЕ

ОНЛАЙН АФИША МЕРОПРИЯТИЙ

по курсу: ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГРУППЫ №	2052		С.А. Баландюк
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2022

Содержание

1. Функциональная спецификация.....	3
2. Руководство пользователя (Покупатель).....	3-7
3. Руководство пользователя (Менеджер).....	7-10
4. Архитектура программы.....	11
5. Описание наиболее важных классов.....	12-17

1. Функциональная спецификация

Данный проект представляет собой онлайн афишу мероприятий. Всего существует два вида пользователей: покупатели и менеджеры (admin). Для обоих требуется регистрация на сайте.

Менеджер может редактировать список предстоящих мероприятий, то есть добавлять новые мероприятия, удалять мероприятия, изменять информацию о мероприятии (название, цена, количество билетов, обложка). Менеджер может отслеживать какое количество билетов сейчас в наличии. Также менеджер имеет статистику заказов в виде таблицы.

Клиент в свою очередь может добавлять билеты на различные мероприятия в корзину, а также удалять их оттуда. Клиент может выбрать одну из четырех категорий событий - 'КИНО', 'КОНЦЕРТЫ', 'ТЕАТР', 'СПОРТ'. Также покупателю предлагается выбрать дату, чтобы просмотреть все мероприятия назначенные на эту дату. Менеджер и покупатель магазина могут использовать функцию поиска по каталогу. Одновременно пользоваться сайтом могут несколько покупателей, соответственно у каждого из них будет своя корзина покупок.

2. Руководство пользователя (Покупатель)

При переходе на сайт пользователя встречает главная страница (Рис.1). В правом верхнем углу две ссылки для входа пользователя и его регистрации на сайте. В левом верхнем углу пользователь может отсортировать мероприятия по категориям. То есть нажимая на кнопку 'Концерты' будут показаны только мероприятия с категорией 'Концерты'. Помимо этого на главной странице есть возможность поиска по товарам. Также у пользователя есть возможность выбрать все мероприятия, назначенные на определенную дату.

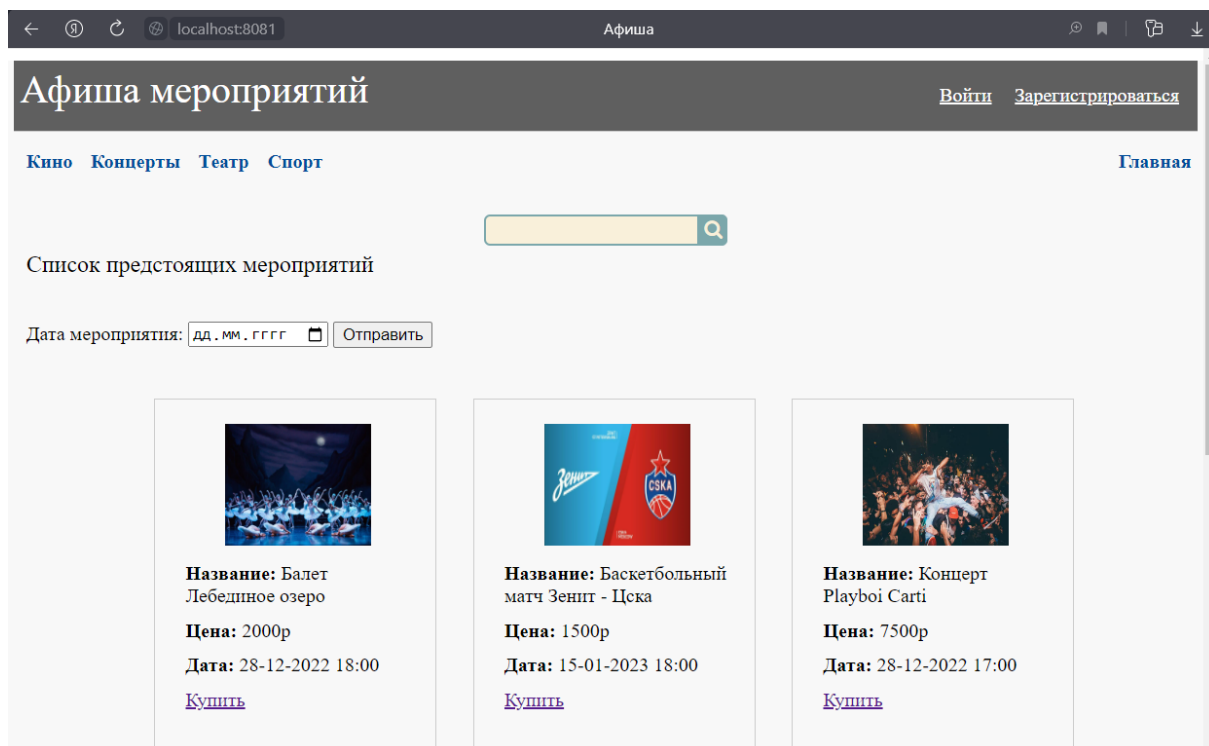


Рис. 1 : Главная страница. URL: '/'

При нажатии на кнопку 'Войти' пользователь попадает на страницу с формой для входа, которая представлена на (Рис. 2). Если пользователь, у которого нет учетной записи попытается войти в систему, то будет ошибка. Сначала ему нужно пройти регистрацию. При вводе неправильного пароля или логина также будет ошибка и просьба ввести данные повторно.

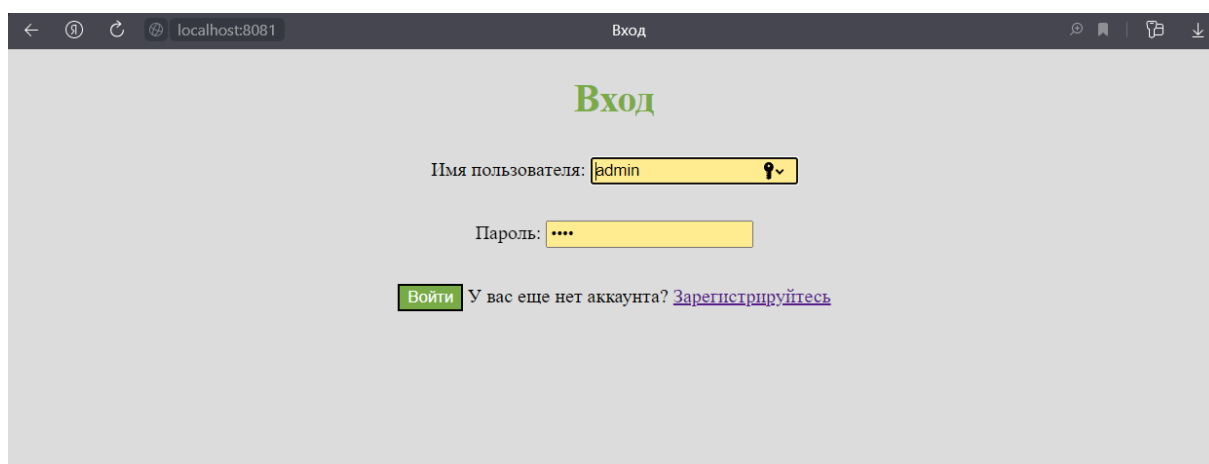


Рис. 2 : Вход пользователя. URL: '/login'

При нажатии на кнопку 'Зарегистрироваться' пользователь попадает на страницу с формой для регистрации, которая представлена на (Рис. 3). При вводе почты в некорректном формате, при несовпадении паролей и слишком коротком логине будут выдаваться ошибки.

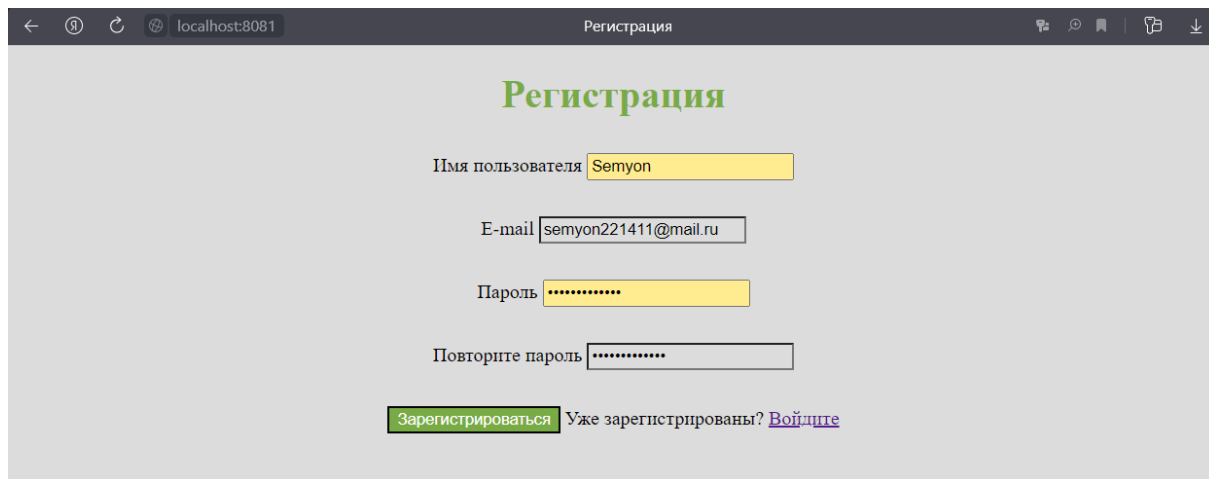


Рис. 3 : Регистрация пользователя. URL: '/registration'

После входа пользователя в систему у него появляется доступ к корзине. Чтобы добавить товар в корзину нужно нажать на кнопку 'Купить'. При нажатии на кнопку 'Корзина' пользователь переходит в корзину (Рис. 4)

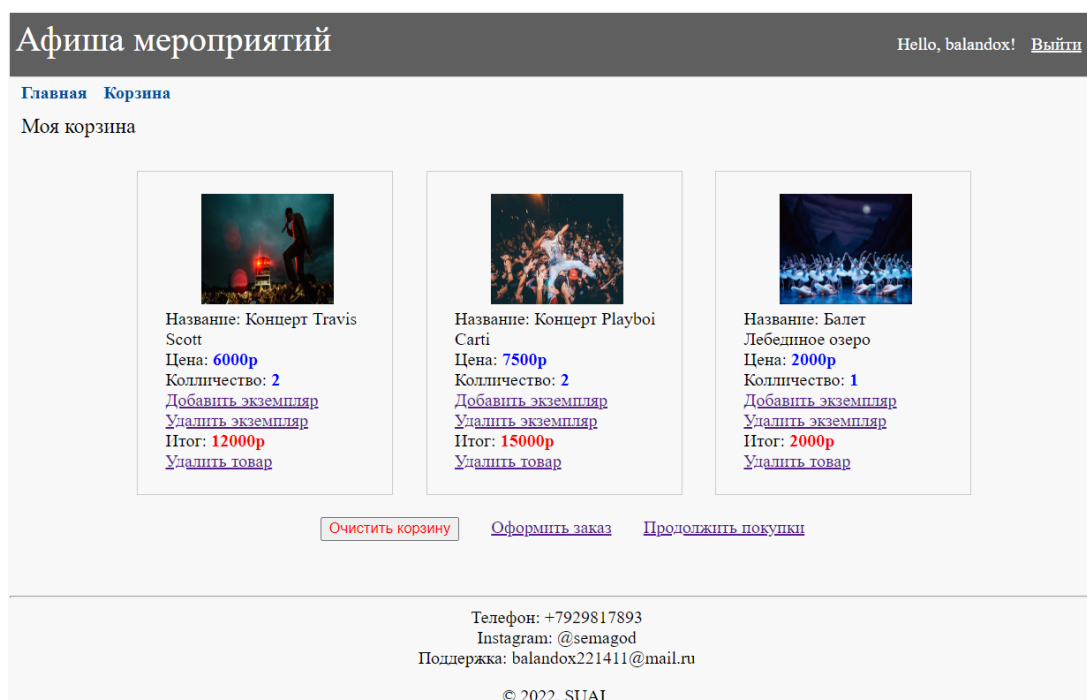


Рис. 4 : Корзина пользователя. URL: '/user/bucket'

Пользователь может редактировать корзину при помощи следующих кнопок:

- 'Добавить экземпляр' - добавляет в корзину один экземпляр билета. При это происходит автоматический пересчет стоимости.
- 'Удалить экземпляр' - удаляет из корзины один экземпляр билета. При это происходит автоматический пересчет стоимости.
- 'Удалить товар' - удаляет все экземпляры данного товара. При это происходит автоматический пересчет стоимости.

При нажатии на кнопку 'Оформить заказ' пользователь переходит на страницу подтверждения заказа (Рис. 5).

Афиша мероприятий

Hello, balandox! [Выйти](#)


[Главная](#) [Корзина](#)

Подтверждение


Информация о покупателе:
Имя: balandox
Email: balandox221411@mail.ru

Итоговая корзина:
Количество товаров: 5 Билетов
Итоговая сумма: **29000р**

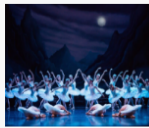
[Редактировать корзину](#) [Подтвердить заказ](#)



Название: Концерт Travis Scott
Цена: **6000р**
Количество: **2**
Итог: **12000р**



Название: Концерт Playboi Carti
Цена: **7500р**
Количество: **2**
Итог: **15000р**



Название: Балет Лебединое озеро
Цена: **2000р**
Количество: **1**
Итог: **2000р**

Телефон: +7929817893
Instagram: @semagod
Поддержка: balandox221411@mail.ru

© 2022, SUAI

Рис. 5 : Подтверждение заказа. URL: '/user/bucketConfirmation'

При нажатии на кнопку 'Подтвердить заказ' пользователь завершает оформление заказа и перенаправляется на страницу успешного оформления заказа (Рис. 6).

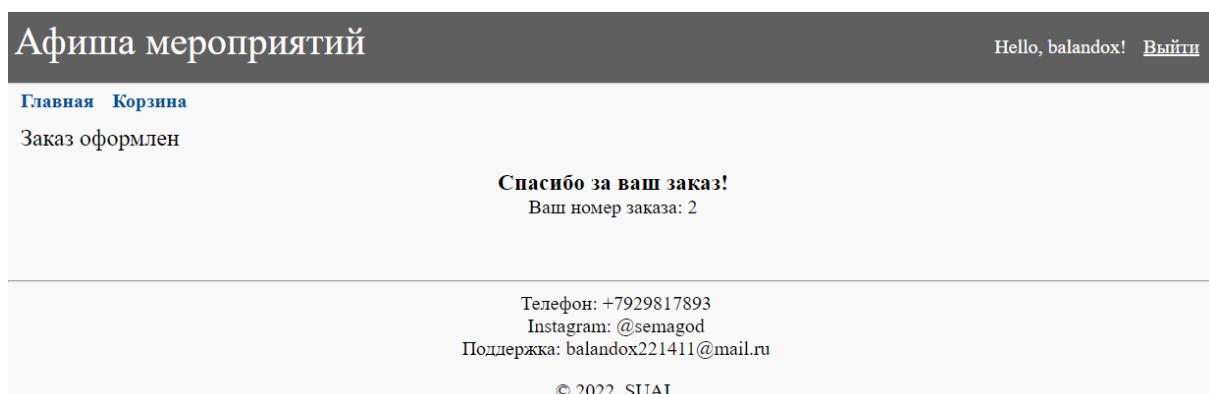


Рис. 6 : Успешное оформление заказа. URL: '/user/bucketFinalize'

3. Руководство пользователя (Менеджер)

Главная страница сайта при входе от имени менеджера показана на (Рис. 7).

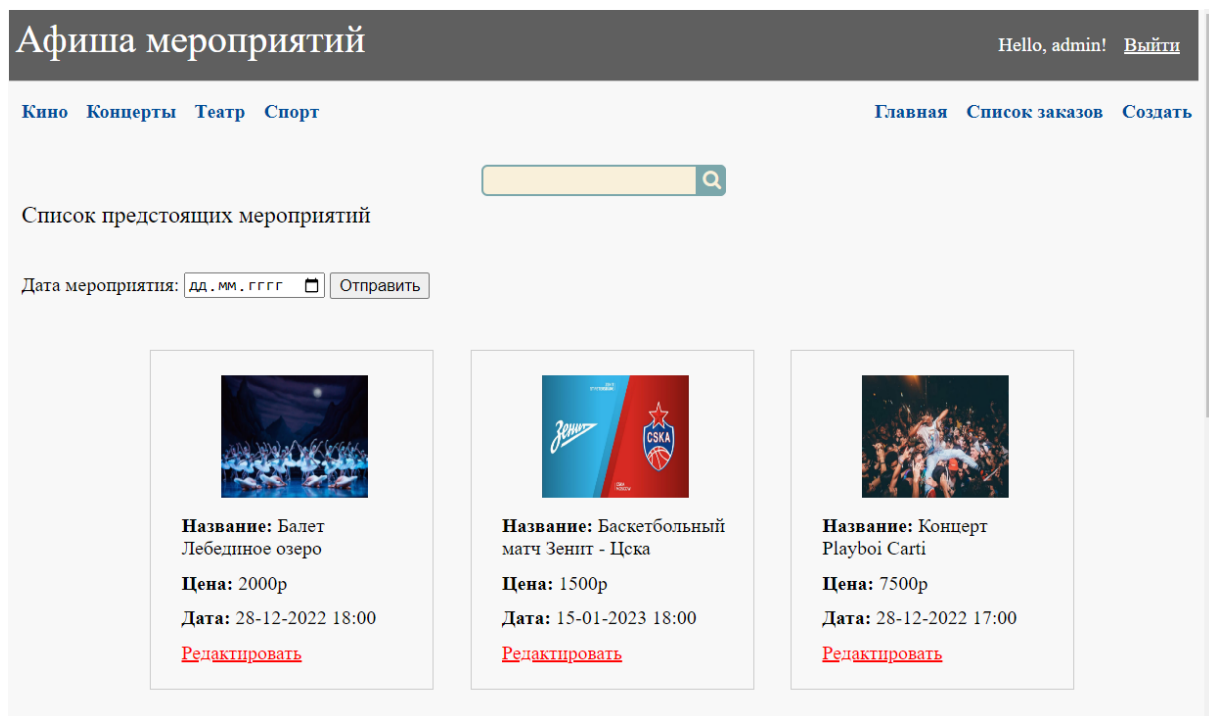


Рис. 7 : Главная страница (Менеджер). URL: '/'

При нажатии на кнопку 'Редактировать' открывается страница редактирования товара (Рис. 8). На данной странице менеджер может изменить имя товара, цену товара, количество оставшихся билетов, дату и время проведения мероприятия, категорию мероприятия и также обложку мероприятия.

The screenshot shows a web interface for editing an event. At the top, there's a dark header with the title 'Афиша мероприятий' and a user greeting 'Hello, admin!' with a 'Выйти' (Logout) link. Below the header, there are navigation links: 'Главная', 'Список заказов', and 'Создать'. The main section is titled 'Редактировать товар'. It contains several form fields: 'Название' (Name) with the value 'Концерт Playboi Carti', 'Цена' (Price) with '7500', 'Кол-во' (Quantity) with '1198', 'Дата и время мероприятия' (Event date and time) with '28.12.2022 17:00' and a calendar icon, 'Категория' (Category) with a dropdown menu showing 'CONCERT', and 'Картинка' (Image) with a preview of a concert scene. Below the image preview is a text field for the image URL containing 'https://i.postimg.cc/8kHvhVI'. At the bottom of the form is a 'Подтвердить' (Confirm) button. The footer of the page contains contact information: 'Телефон: +7929817893', 'Instagram: @semagod', 'Поддержка: balandox221411@mail.ru', and a copyright notice '© 2022, SUAI'.

Рис. 8 : Редактирование товара. URL: '/admin/editProduct'

При нажатии на кнопку 'Список заказов' открывается страница, на которой отображены все заказы (Рис. 9). На данной странице в виде таблицы отображаются все заказы. Каждая строка - это один заказ. На этой странице менеджер может увидеть номер заказа, дату, когда заказ был оформлен, логин покупателя, email покупателя, общее количество товаров в заказе и итоговую сумму заказа.

Афиша мероприятий
Hello, admin! [Выйти](#)

[Главная](#)
[Список заказов](#)
[Создать](#)

Список заказов

Общее количество заказов: 2

Номер Заказа	Дата заказа	Логин покупателя	Email покупателя	Кол-во товаров	Итоговая сумма	Смотреть
1	21-12-2022 02:12	balandox	balandox221411@mail.ru	5	29000р	Смотреть
2	21-12-2022 02:14	balandox	balandox221411@mail.ru	3	3950р	Смотреть

Телефон: +7929817893
Instagram: @semagod
Поддержка: balandox221411@mail.ru
© 2022, SUAI

Рис. 9 : Список заказов. URL: '/admin/orderList'

При нажатии на кнопку смотреть открывается подробная информация о заказе (Рис. 10). На данной странице отображаются данные покупателя, итоговая сумма заказа, общее количество билетов в заказе, а также в виде таблицы представлены товары, которые купил пользователь, цена товара, их количество и сумма всех экземпляров одного товара.

Афиша мероприятий
Hello, admin! [Выйти](#)

[Главная](#)
[Список заказов](#)
[Создать](#)

Информация о заказе

Информация о покупателе:
Имя: balandox
Email: balandox221411@mail.ru
Итоговая корзина:
Количество товаров: 5 Билетов
Итоговая сумма: **29000р**

Название	Цена	Количество	Итог
Концерт Playboi Carti	7500	2	15000
Концерт Travis Scott	6000	2	12000
Балет Лебединое озеро	2000	1	2000

Телефон: +7929817893
Instagram: @semagod
Поддержка: balandox221411@mail.ru
© 2022, SUAI

Рис. 10 : Подробная информация о заказе. URL: '/admin/orderderInfo?orderId=id'

При нажатии на кнопку 'Создать' открывается страница добавления товара (Рис. 11). Для создания товара менеджеру необходимо указать название мероприятия, цену билета, общее количество билетов, дату и время проведения мероприятия, категорию мероприятия и его обложку. При вводе отрицательной цены, отрицательного количества товаров или оставление пустых полей будет ошибка.

Афиша мероприятий

Hello, admin! [Выйти](#)

[Главная](#) [Список заказов](#) [Создать](#)

Редактировать товар

Название

Цена


Кол-во

Дата и время мероприятия

Категория

CONCERT ▾

Картинка



Имя картинк

Подтвердить

Телефон: +7929817893

Instagram: @semagod

Поддержка: balandox221411@mail.ru

© 2022, SUAI

Рис. 11 : Создать товар. URL: '/admin/createProduct'

4. Архитектура программы

Графическая схема архитектуры программы показана на (Рис. 12). Архитектура данного приложения основана на принципе Model-View-Controller (MVC). View посылает GET и POST запросы, далее их принимает Controller (1). Controller обеспечивает взаимодействие между пользователем и системой, контролирует и направляет данные от пользователя к системе и наоборот. После получение запросов из View, Controller создает определенную логику в Model (2). То есть, в Model мы обрабатываем запросы клиента. Как правило, данные в современных приложениях хранятся в базах данных. За получение данных из БД, их обработку и передачу этих самых данных обратно в Model, а затем во View отвечают Service(3) и Repository(4). В Repository мы напрямую обращаемся к БД (5) и получаем из нее необходимые данные (делаем запросы). А затем эти полученные данные направляются в Service (6), где они определенным образом обрабатываются и создается основная бизнес-логика проекта.

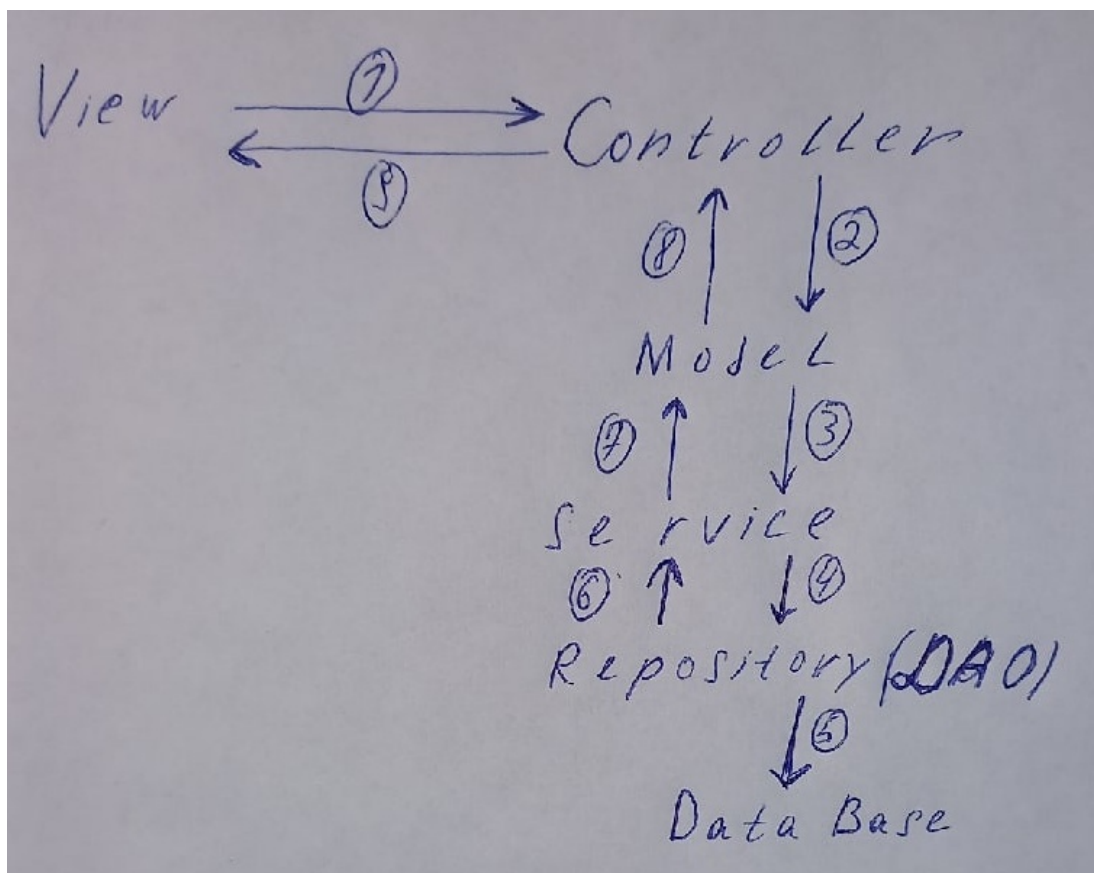


Рис. 12 : Архитектура программы

5. Описание наиболее важных классов

1. Пакет config.

Классы:

- SecurityConfig - отвечает за конфигурирование авторизации, аутентификации и безопасности приложения.

2. Пакет controllers.

Классы:

- MainController - в данном классе мы обрабатываем следующие запросы:
 - GET-запрос '/' - возвращает главную страницу со всеми мероприятиями.
 - GET-запрос '/registration' - возвращает страницу с формой для регистрации.
 - POST-запрос '/registration' - добавляет зарегистрировавшегося пользователя.
 - GET-запрос '/login' - возвращает страницу с формой для входа пользователя.
- UserController - в данном классе мы обрабатываем следующие запросы:
 - GET-запрос '/user/buyProduct' - добавляет товар в корзину.
 - GET-запрос '/user/bucket' - возвращает страницу с корзиной покупателя.
 - POST-запрос '/user/bucket' - очищает корзину.
 - GET-запрос '/user/bucket/addProduct' - добавляет экземпляр товара в корзину.
 - GET-запрос '/user/bucket/reduceProduct' - удаляет экземпляр товара из корзины.

- GET-запрос `’/user/bucket/removeProduct’` - удаляет все экземпляры товара из корзины.
 - GET-запрос `’/user/bucket/bucketConfirmation’` - возвращает страницу подтверждения заказа.
 - POST-запрос `’/user/bucket/bucketConfirmation’` - возвращает страницу успешного оформления заказа.
- AdminController - в данном классе мы обрабатываем следующие запросы:
 - GET-запрос `’/admin/orderList’` - возвращает страницу со списком заказов.
 - GET-запрос `’/admin/orderInfo’` - возвращает страницу с подробной информацией о заказе.
 - GET-запрос `’/admin/productEdit’` - возвращает страницу с редактированием товара.
 - POST-запрос `’/admin/productEdit’` - сохраняет измененный товар.
 - GET-запрос `’/admin/createProduct’` - возвращает страницу с созданием товара.
 - POST-запрос `’/admin/createProduct’` - сохраняет новый товар.

3. Пакет models

Все классы, лежащие в данном пакете являются сущностями, то есть каждому классу соответствует одна таблица в БД.

Классы:

- User - данный класс характеризует пользователя, имеет следующие поля: id, name, password, email, role, bucket, orderList.
- Product - данный класс характеризует мероприятие, имеет следующие поля: id, title, price, amount, img, dateOfEvent, category, bucketList, orderDetails.

- OrderDetails - данный класс характеризует товар в заказе, имеет следующие поля: id, quantity, price, totalSum, order, product.
- Order - данный класс характеризует заказ, имеет следующие поля: id, orderNum, creationTime, totalSum, user, orderDetails.
- Bucket - данный класс характеризует корзину, имеет следующие поля: id, user, productList.

4. Пакет repositories

Все интерфейсы в данном пакете наследуются от интерфейса JpaRepository, который уже реализует за нас основные методы CRUD приложения (save, delete, update, get).

Интерфейсы:

- UserRepository - данный интерфейс делает запросы к сущности User в БД.

Основные методы:

- User findFirstByName(String name) - находит пользователя по его имени.
- User findUserByEmail(String email); - находит пользователя по его email.

- ProductRepository - данный интерфейс делает запросы к сущности Product в БД.

Основные методы:

- List<Product> findByTitleContainingIgnoreCase(String title); - используется для поиска по каталогу.
- List<Product> findByCategory(Category category); - возвращает мероприятия, соответствующие определенной категории.
- List<Product> findByDateOfEvent(Date date); - возвращает мероприятия, запланированные на определенную дату.

- OrderRepository - данный интерфейс делает запросы к сущности Order в БД.

Основные методы:

- List<Order> findByOrderByCreationTime(); - возвращает список всех заказов и сортирует их по дате создания.

- OrderDetailsRepository - данный интерфейс делает запросы к сущности OrderDetails в БД.

- BucketRepository - данный интерфейс делает запросы к сущности Bucket в БД.

Основные методы:

- void deleteBucketById(int id); - удаляет корзину по id.

5. Пакет services

Классы:

- UserService - данный класс отвечает за бизнес-логику, связанную с сущностью User.

Основные методы:

- public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException - возвращает класс UserDetails, соответствующий определенному пользователю, необходимый для Spring Security.
- public User findByEmail(String email) - находит пользователя по email.
- public User findFirstByName(String name) - находит пользователя по имени.
- public List<User> allUsers() - возвращает список всех пользователей.
- public User saveUser(User user) - сохраняет нового пользователя.
- public boolean deleteUser(int userId) - удаляет пользователя.

- ProductService - данный класс отвечает за бизнес-логику, связанную с сущностью Product.

Основные методы:

- public Product getProductById(int id) - возвращает продукт по id.
 - public List<Product> getAll() - возвращает список всех продуктов.
 - public void saveEditedProduct(ProductForm productForm, int productId, Date dateOfEvent) - сохраняет измененный продукт.
 - public List<Product> searchProductByTitle(String search) - поиск в каталоге мероприятий.
 - public List<Product> getAllByDate(String dateString) - возвращает все мероприятия, назначенные на определенную дату.
 - public List<Product> getAllByCategory(Category category) - возвращает все мероприятия, соответствующие определенной категории.
- OrderService - данный класс отвечает за бизнес-логику, связанную с сущностью Order.

Основные методы:

- public Order saveOrder(String name) - сохраняет заказ.
- public OrderDTO findOrderById(int id) - находит заказ по id.

- BucketService - данный класс отвечает за бизнес-логику, связанную с сущностью Bucket.

Основные методы:

- public void addProductToBucket(Product product, String name) - добавляет товар в корзину.
- public void reduceQuantityOfProduct(Product product, String name) - удаляет экземпляр товара из корзины.

- `public void deleteProductFromBucket(Product product, String name)`
 - удаляет все экземпляры определенного товара из корзины.
- `public void clearBucket(String name)` - очищает корзину.