**Министерство науки и высшего образования Российской Федерации** **Федеральное государственное бюджетное образовательное учреждение высшего образования** **«Московский государственный технический университет** **имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Отчёт по курсу

«Разработка интернет приложений»

По теме «Интернет-магазин игр»

Выполнил:

студент группы ИУ5-53Б

Ефременко Д.С.

Преподаватель:

Канев А.И.

2022 г.

Содержание

[Введение 3](#_Toc121153259)

[1.Бизнес-процесс 4](#_Toc121153260)

[1.1. Диаграмма прецедентов 4](#_Toc121153261)

[1.2. Диаграмма состояний 5](#_Toc121153262)

[2.Архитектура 7](#_Toc121153263)

[2.1 Диаграммы развертывания 7](#_Toc121153264)

[2.2. ER с назначением таблиц 8](#_Toc121153265)

[2.3. Диаграмма компонентов с детализацией бэкенда 9](#_Toc121153266)

[3.Алгоритмы. 9](#_Toc121153267)

[3.1. Диаграмма последовательности и деятельности 9](#_Toc121153268)

[4. Описание интерфейса. Перечень окон, и их назначение 9](#_Toc121153269)

[4.1 Авторизация 10](#_Toc121153273)

[4.2. Вход 10](#_Toc121153274)

[4.3. Регистрация 10](#_Toc121153275)

[4.4. Пользователь 10](#_Toc121153276)

[4.5. Главное окно пользователя 10](#_Toc121153277)

[4.6. Корзина пользователя 10](#_Toc121153278)

[4.7. Поиск пользователя 10](#_Toc121153279)

# **Техническое задание**

**Цель:** изучение основы различных фреймворков, инструментов и технологий для создания полноценного интернет приложения.

**Задачи:**

1. Базовая шаблонизация в Django для словаря.
2. Создание базового интерфейса для просмотра списка с ссылками и частью полей
3. Создание базы данных, подключение к шаблонизатору.
4. Создание админки.
5. Создание веб-сервиса для получения/редактирования данных из БД.
6. Базовая работа по фронтенду.
7. Подключение интерфейса фронтенда к веб-сервису.
8. Возможность добавления услуги в корзину.
9. Добавление фильтрации и поиска на странице услуг
10. Переключение между интерфейсом гостя и интерфейсом пользователя
11. Аутентификация и регистрация.
12. Авторизация через хранение сессий
13. Создание мобильного нативного приложения
14. Добавление роли пользователя-менеджера контента

**Требования к функциональным характеристикам:**

Существует три типа пользователей: администратор, менеджер и обычный пользователь. Администратор и менеджер, в дополнение ко всем правам обычного пользователя, также могут редактировать данные о товарах и пользователях. Пользователь может зарегистрироваться и затем входить под своей учетной записью, указав логин и пароль. Только авторизованный пользователь может добавлять/удалять товары из корзины.

**Введение**

**Описание предметной области:**

Это приложение по продаже компьютерных игр. Пользователю предоставляется широкий ассортимент игр, каждая из которых обладает следующим характеристиками: название, жанр, разработчик, издатель, описание, цена. Также данное приложение предоставляет широкий функционал. Пользователь может просматривать список игр, переходить на страницу с подробной информацией, отфильтровывать ассортимент по цене и названию, производить поиск, а также добавлять/удалять игры из корзины. Также имеется возможность авторизации и аутентификации. Пользователь может зарегистрироваться и затем входить под своей учетной записью, указав логин и пароль. Только авторизованный пользователь может добавлять/удалять товары из корзины. Существует три типа пользователей: администратор, менеджер и обычный пользователь. Администратор и менеджер, в дополнение ко всем правам обычного пользователя, также могут редактировать данные о товарах и пользователях. Пользователи могут использовать приложение не только с ПК, но и с мобильных устройств

При разработке использовалась трехуровневая модель MVC (Model, View, Control), а также Django Rest Framework (DRF) для написания сервера с API.

Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.

Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.

Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

В качестве базы данных используется MySQL

Django ORM служит связью между БД и Django, облегчает и ускоряет разработку. Данная технология позволяет создавать модели и управлять ими из кода, создавать БД по описанию моделей и изменять структуру БД при изменении моделей.

Для преобразования модели из базы данных в JSON и наоборот используются сериализаторы.

Проверка правильности работы API и запросов проводилась при помощи Insomnia.

При проектировании проекта использовались различные UML диаграммы: ER, классов, компонентов, deployment, sequence, activity.

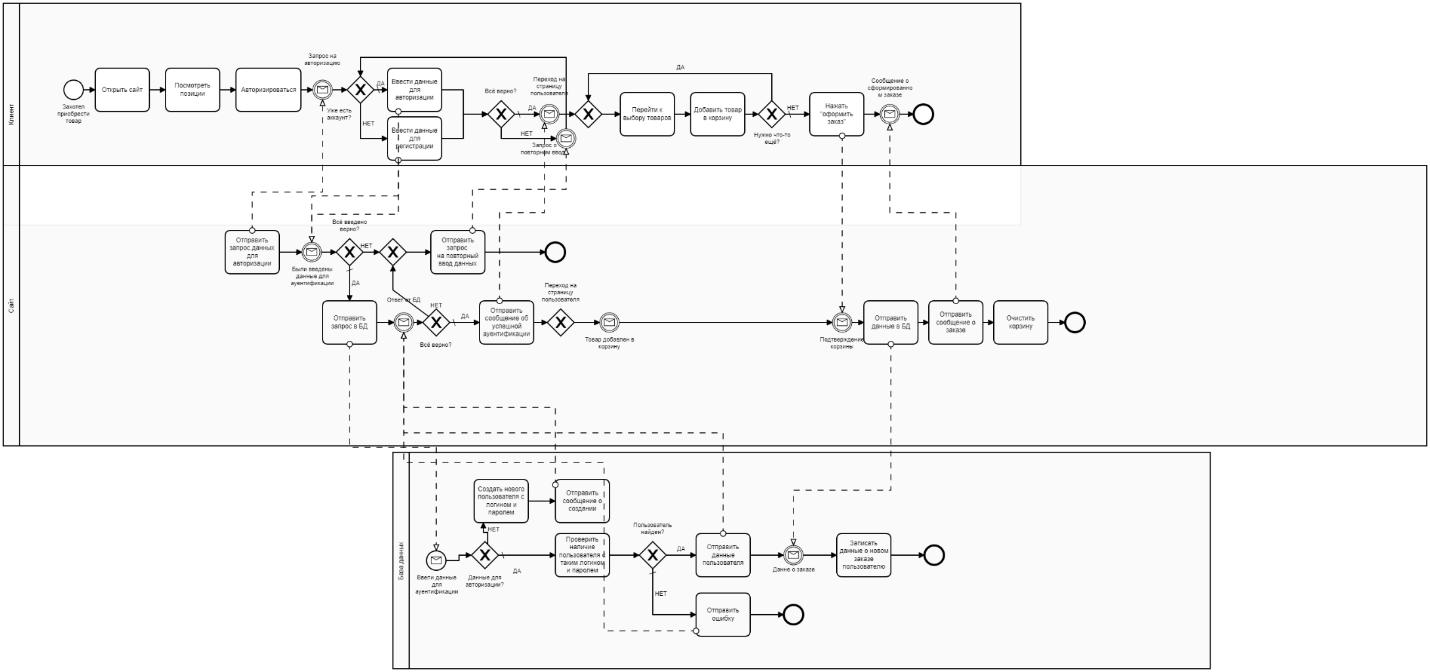
Для написания фронтенда используется фреймворк React c применением функциональных компонентов, хуков, библиотеки для использования состояний Redux.

Авторизация основана на применении JWT, и куки.

**Актуальность:** Данное приложение актуально для магазина компьютерных игр, так как является универсальным хранилищем данных, с которым легко и быстро работать. Удобство работы заключается в возможности просмотра товаров, создания корзины, в которую пользователь может добавлять нужные ему игры и фильтрации покупок.

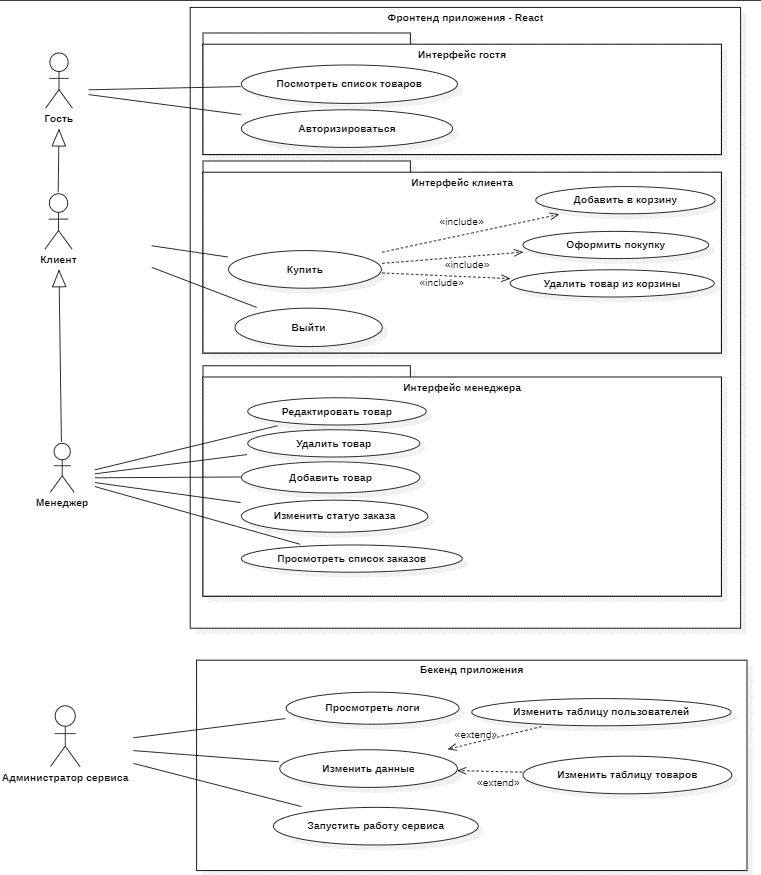
**Бизнес-процесс**

1. **Схема бизнес-процесса**



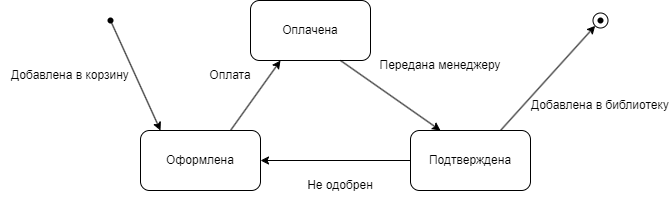
На данной схеме описана суть и схема бизнес-процесса регистрации и авторизации пользователя. Пользователь открывает сайт. Ему предоставляется возможность зарегистрироваться или войти в аккаунт, если он уже ранее был зарегистрирован. При регистрации пользователь вводит логин и пароль, запрос отправляется в БД. При успешной регистрации создается новый пользователь в базе данных, а в случае ошибки отправляется соответствующее сообщение (пользователь с таким логином уже существует или некорректно введены данные). При входе пользователя в систему, он также вводит свои данные. При успешном входе пользователя перенаправляет на начальную страницу приложения, где он может просмотреть список товаров, отфильтровать их и положить в корзину.

1. **Диаграмма прецедентов**



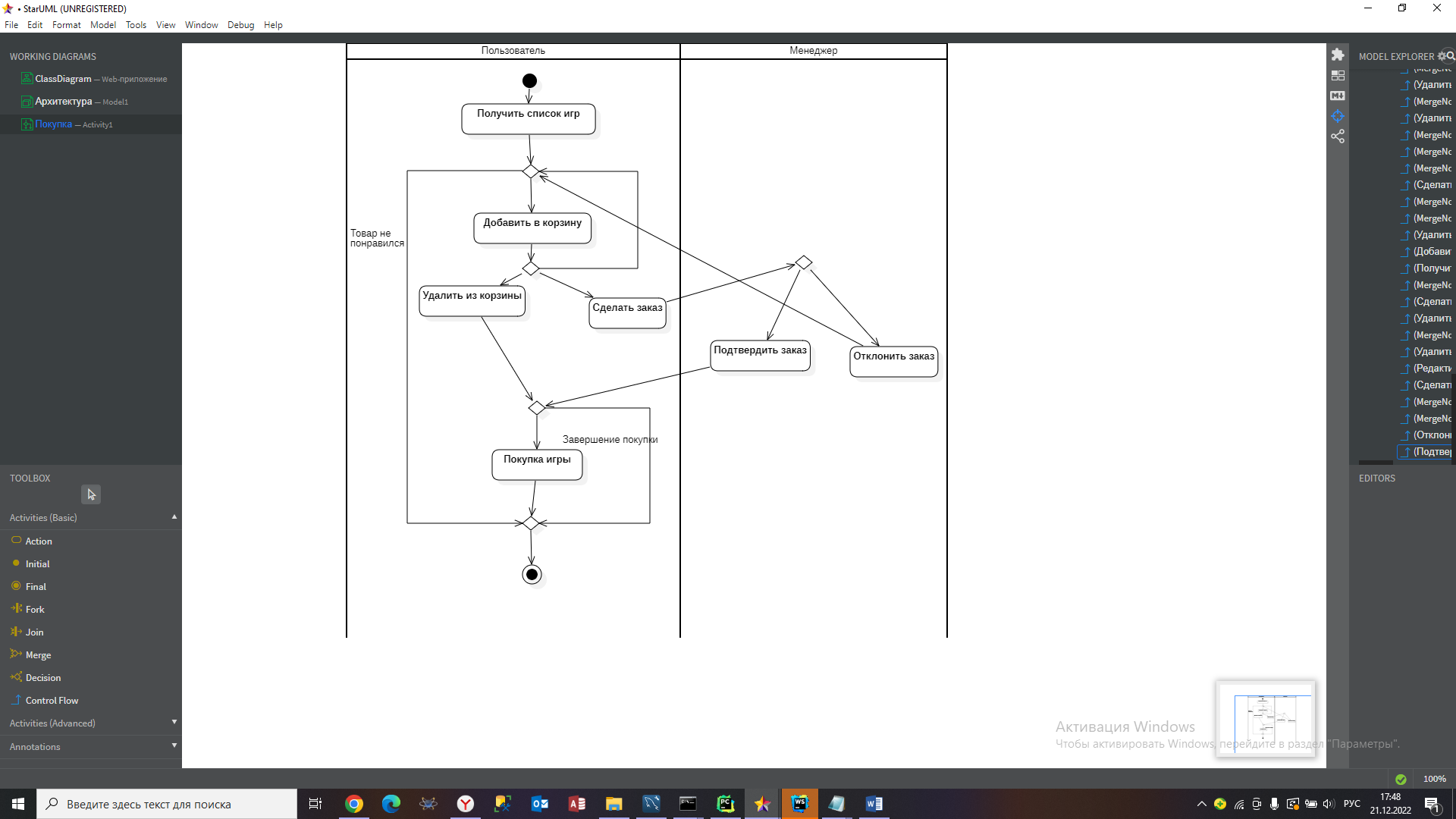
На данной диаграмме описано поведение пользователей приложения. Описаны следующие пользователи:

1. Гость – может просматривать список товаров и авторизоваться
2. Клиент – умеет добавлять игры в корзину, оформлять покупку и удалять товары из корзины
3. Менеджер редактирует товары, удаляет товары, добавляет товары и просматривает список заказов
4. Администратор может менять данные, а также запускать работу сервиса и просматривать логи
5. **Диаграмма состояний**

****

На диаграмме описан жизненный цикл заказа и его возможные состояния:

1. Оформлен
2. Оплачен
3. Подтвержден
4. Игра добавлена в библиотеку
5. **Диаграмма деятельности**



На данной диаграмме описан процесс добавления пользователем товара в корзину и последующая оплата товаров в корзине.

**Архитектура**

1. **Диаграммы развертывания**

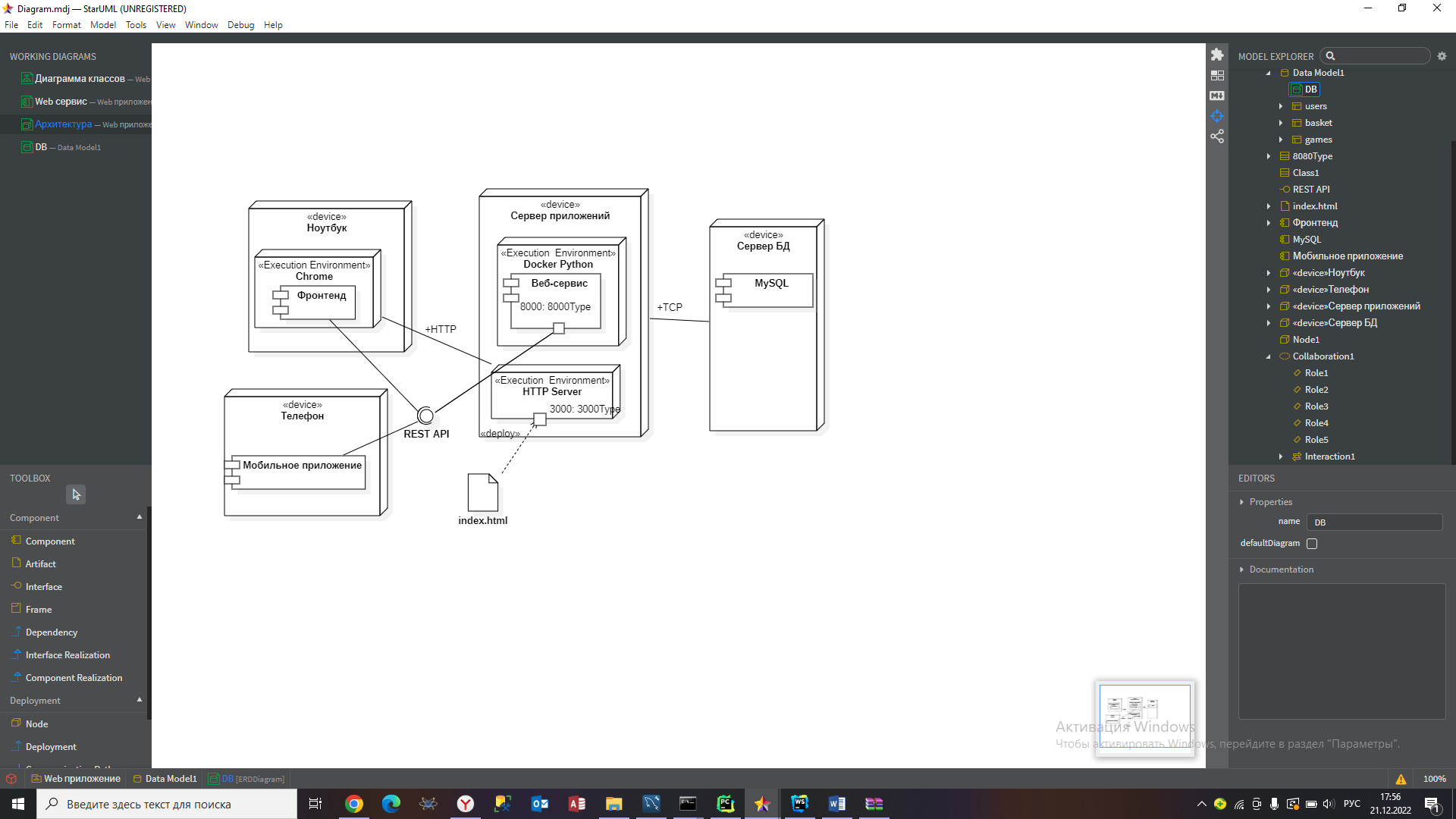
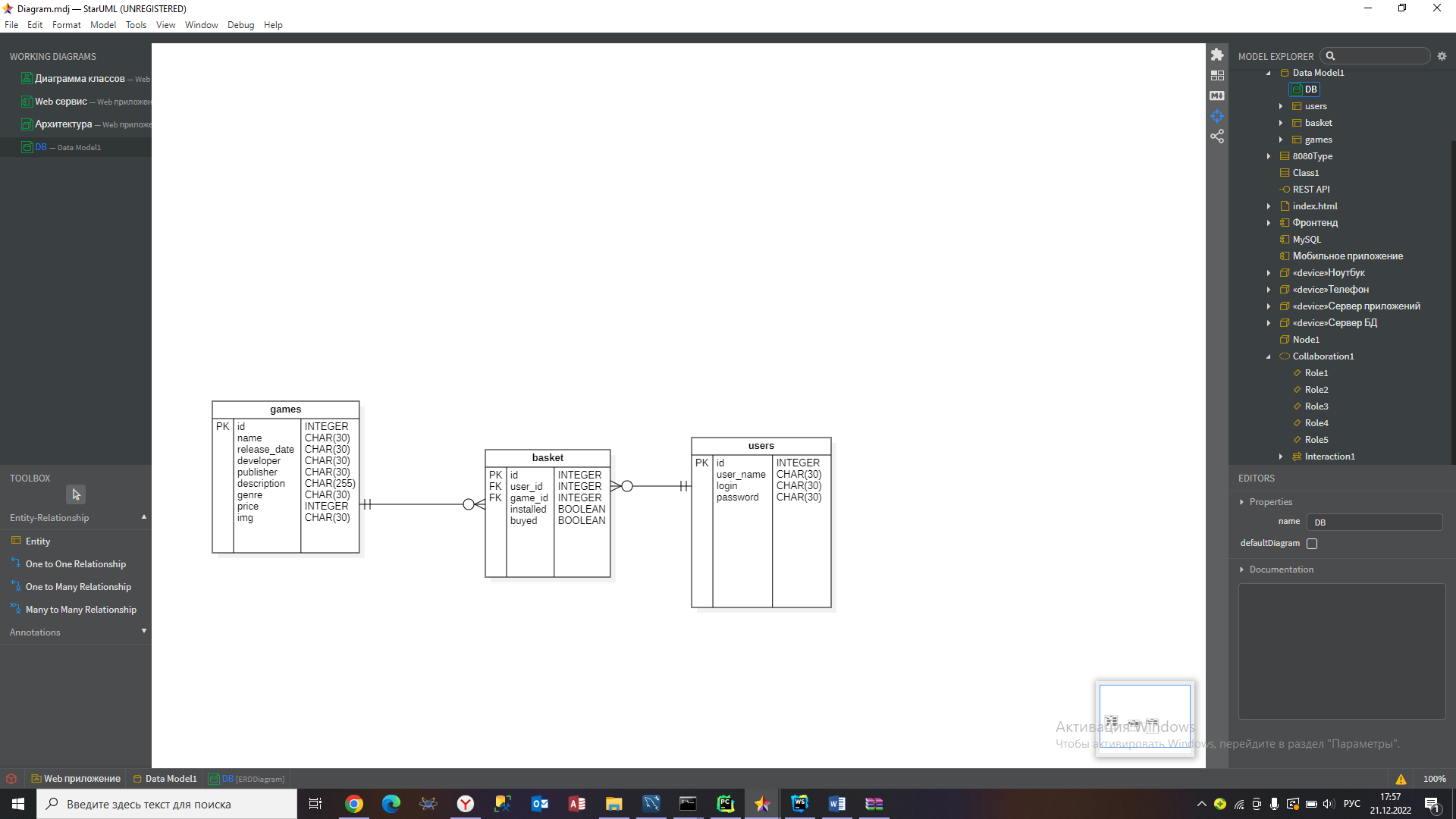
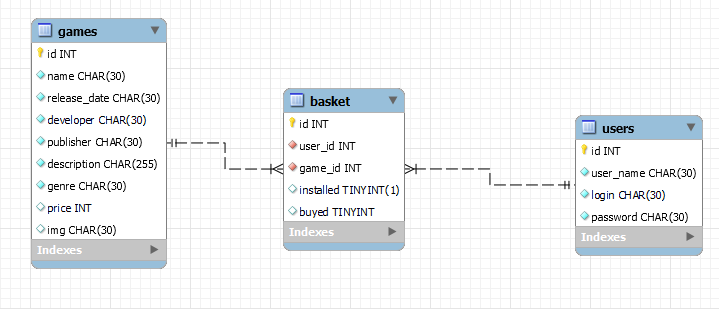


Диаграмма развертывания описывает общую структуру приложения, связь между сервером приложения и базой данных, и устройствами.

1. **ER с назначением таблиц**

На данной и следующей схемах описаны сущности базы данных и связи между ними.



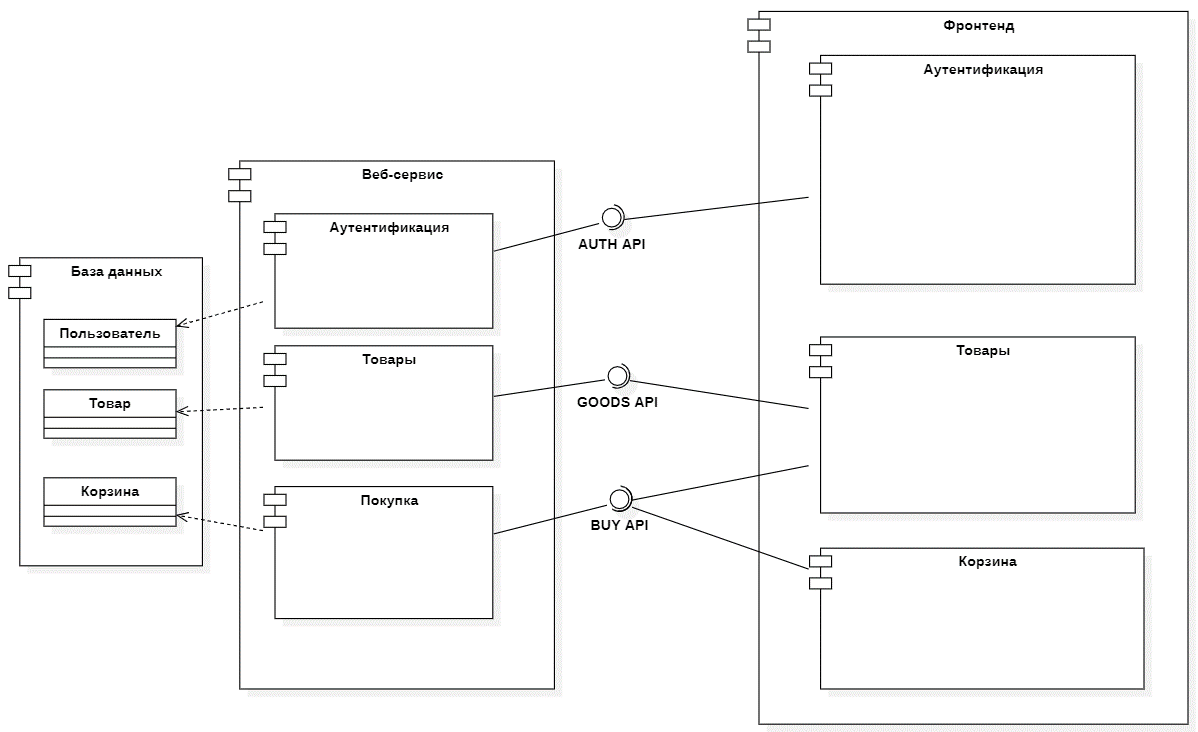
****

В сущности games хранятся данные об играх, из название, жанр, цена и т.д.

Users предназначена для хранения данных о зарегистрированных пользователях.

В basket хранятся связи между играми и пользователями (какие игры приобрел каждый пользователь)

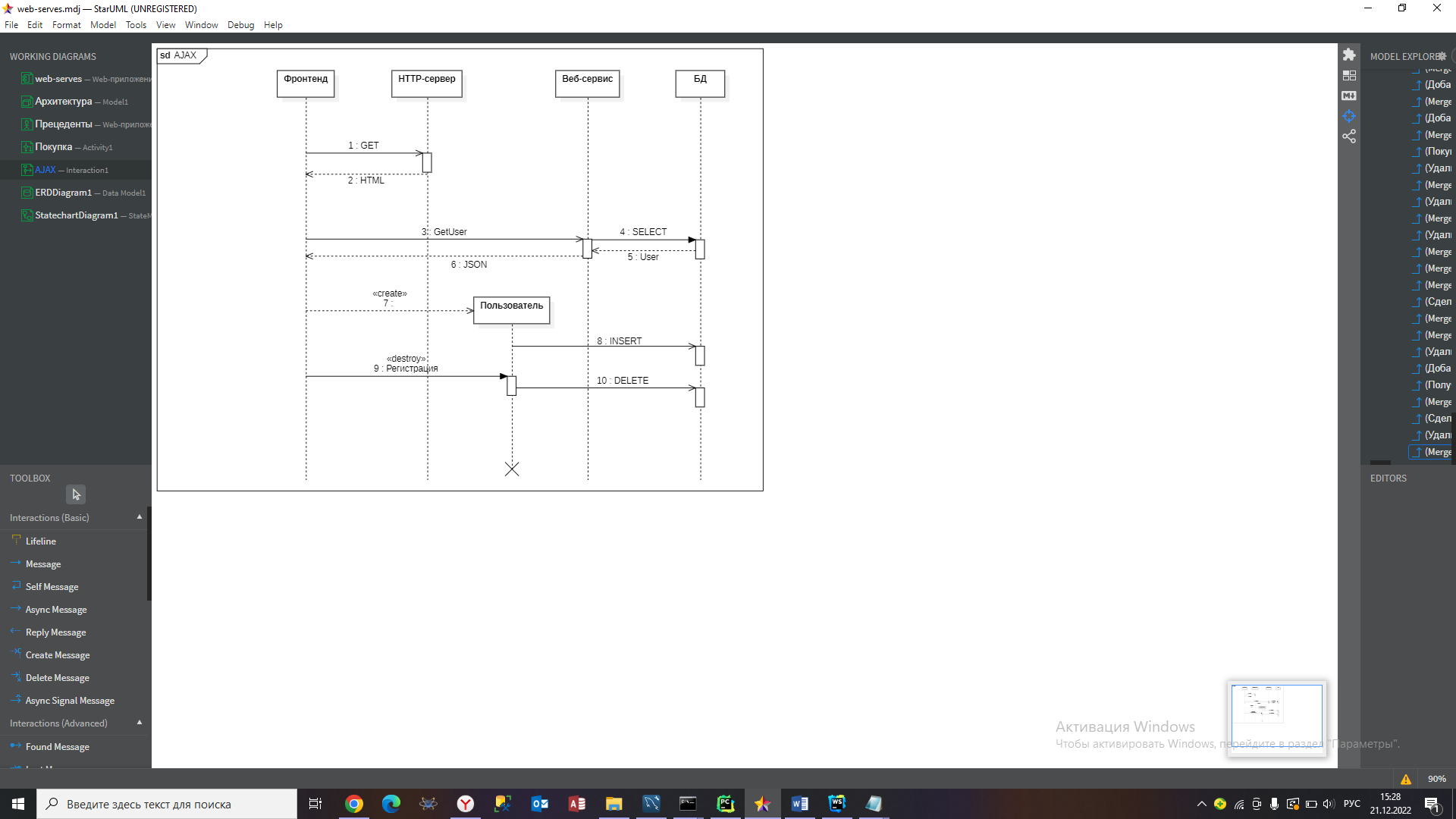
1. **Диаграмма компонентов**



На диаграмме компонентов отражено взаимодействие между идентичными объектами веб-сервиса и фронтенда, а также связь объектов веб-сервиса с базой данных.

**Алгоритмы**

1. **Диаграмма последовательности**

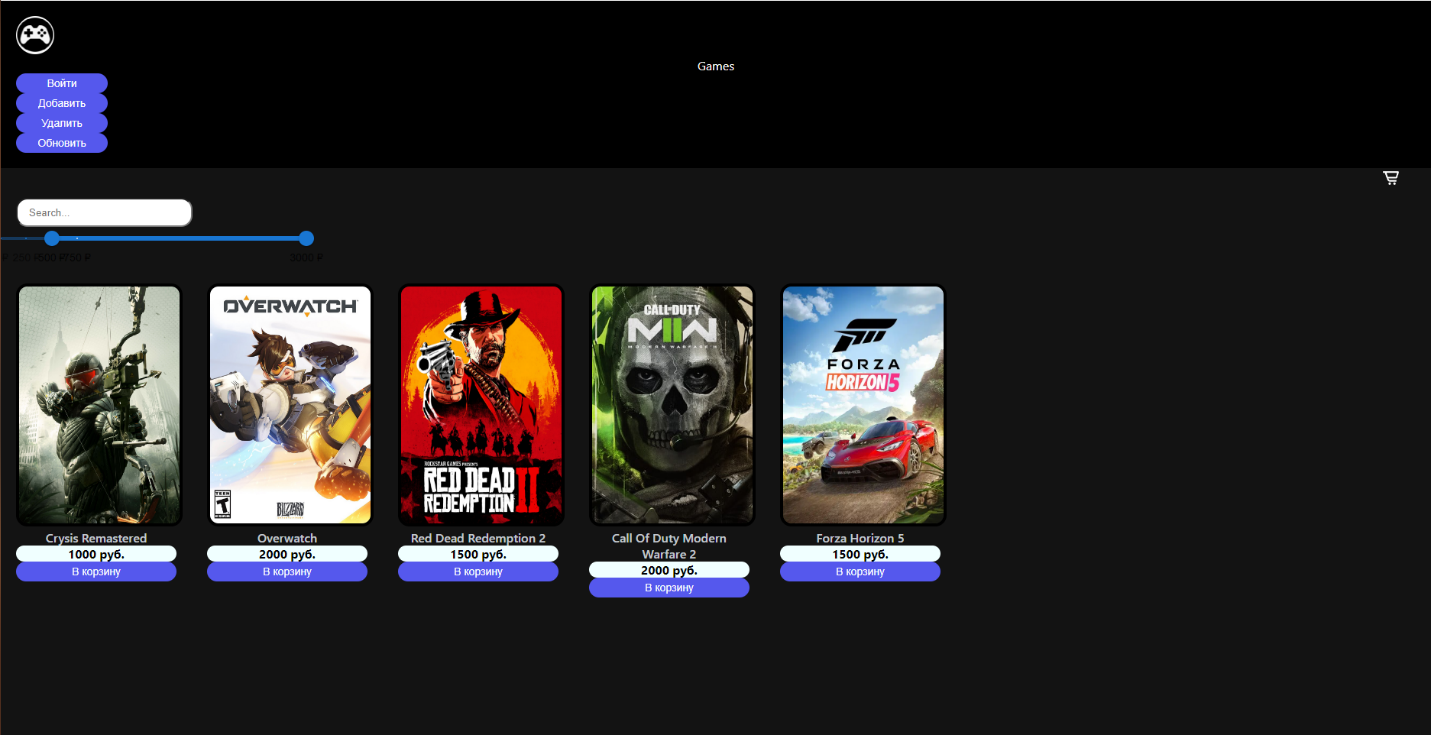


На диаграмме последовательности описано взаимодействие объектов приложения и их жизненный цикл на единой временной оси.

**Описание интерфейса**

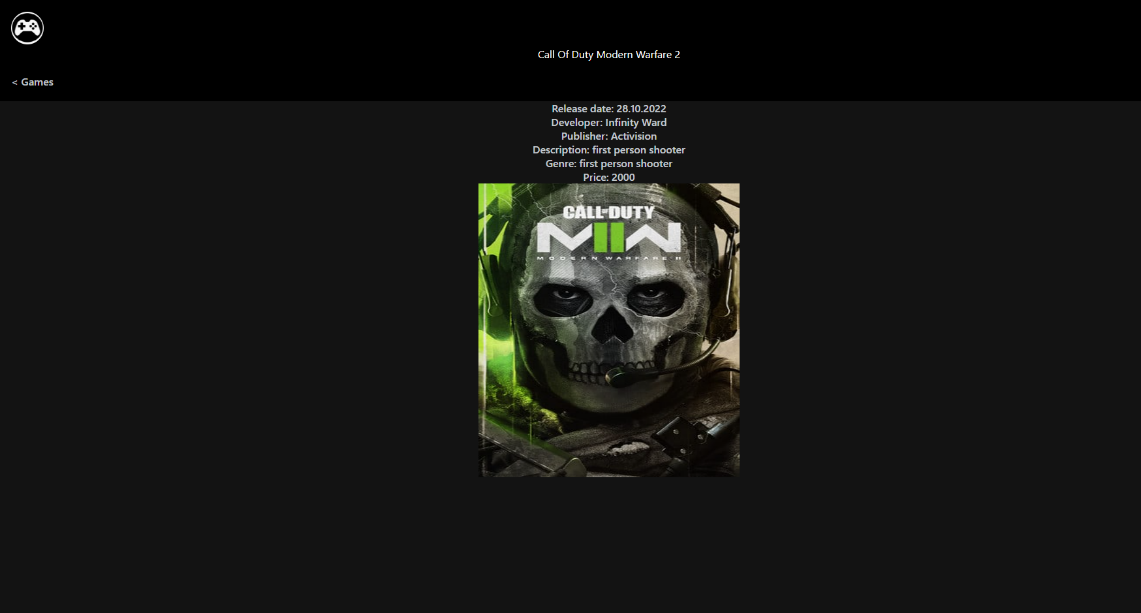
**Начальное окно**

Начальная страница позволяет просматривать игры и их цену, добавлять в корзину, а также производить поиск по названию игр и фильтровать по цене.



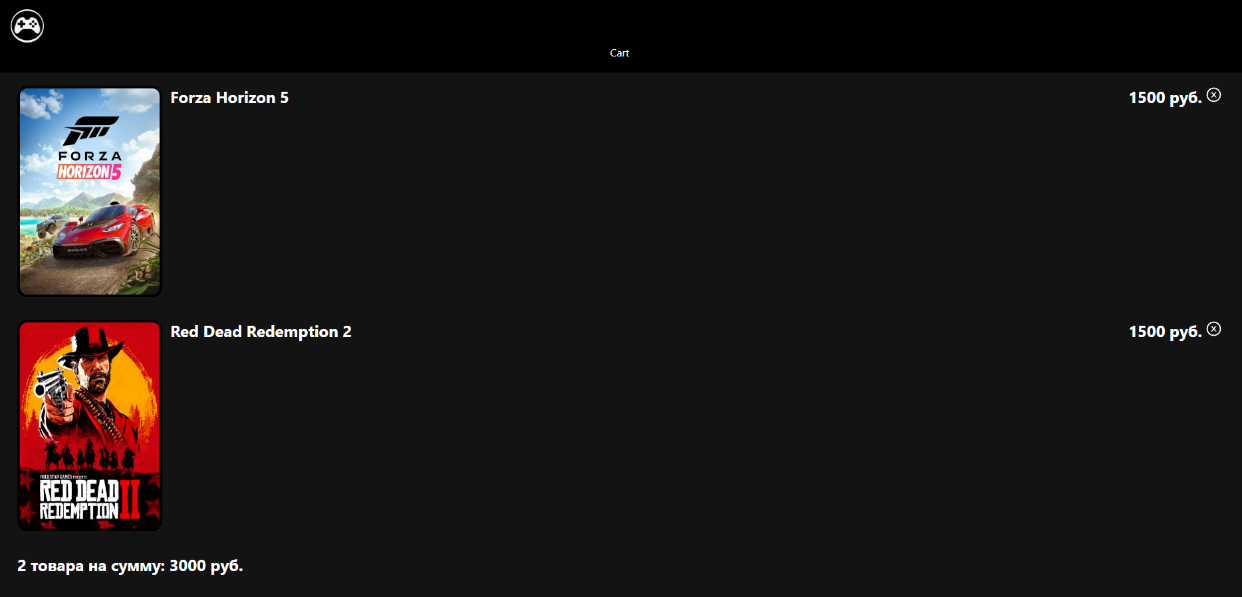
**Страница с подробной информацией**

Подробную информацию о каждой игре пользователь может получить по нажатию на название игры.



**Корзина**

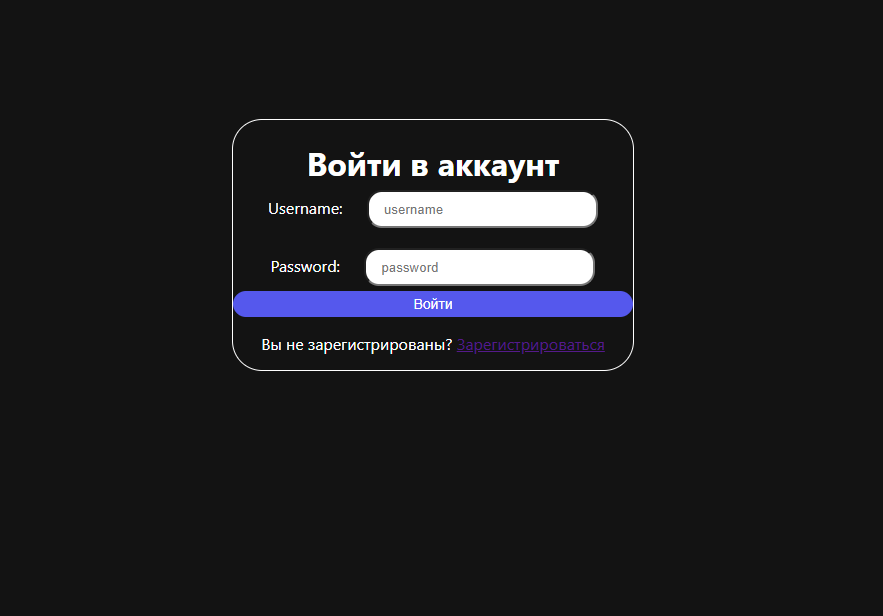
В **корзине** можно посмотреть добавленные игры и общую цену покупки



**Авторизация**

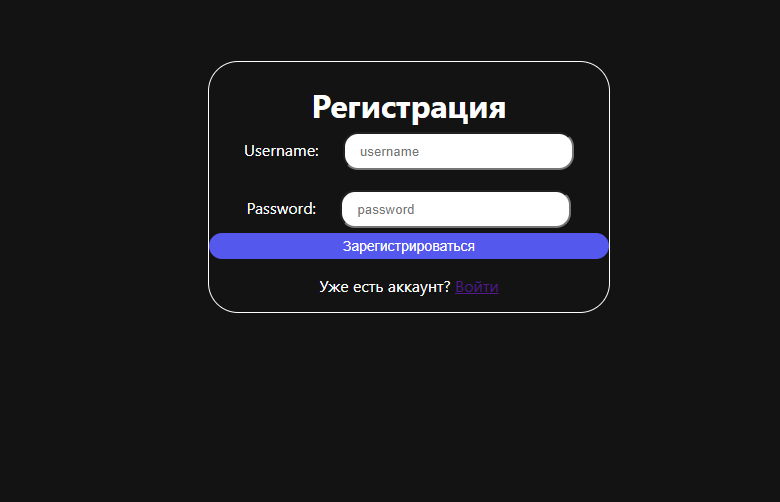
**Вход**

Просит гостя с учётной записью ввести логин и пароль. Если такой пользователь найден в БД переходит на окно пользователя. В противном случае выдаёт ошибку

****

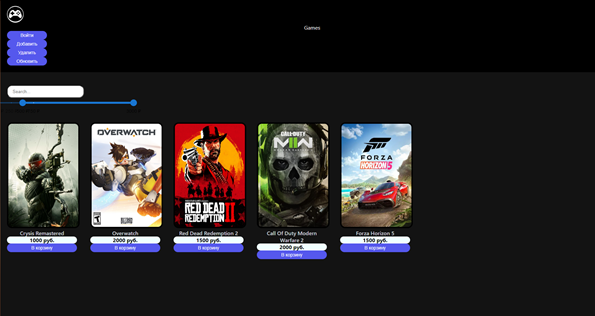
**Регистрация**

У гостя запрашиваются личные данные, необходимые для регистрации: почта, имя, фамилия, пароль. Идёт проверка корректности введённых данных. Если регистрация прошла успешно создаётся запись в БД о новом пользователе с его данными, происходит переход на окно пользователя.



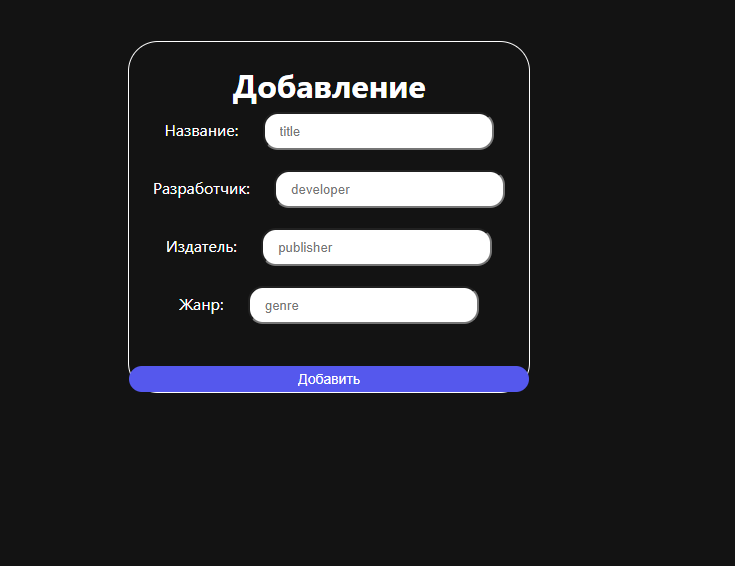
**Главное окно менеджера**

Позволяет просматривать игры и их цену производить поиск по названию игр и фильтровать по цене, а также редактировать, добавлять и удалять игры.

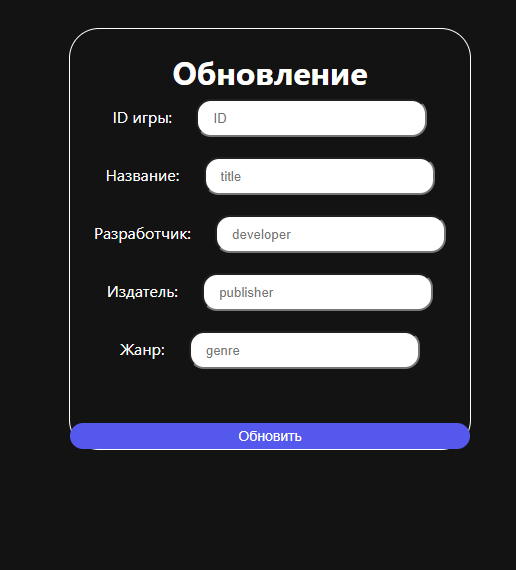
. 

**Добавление, изменение, удаление игр**

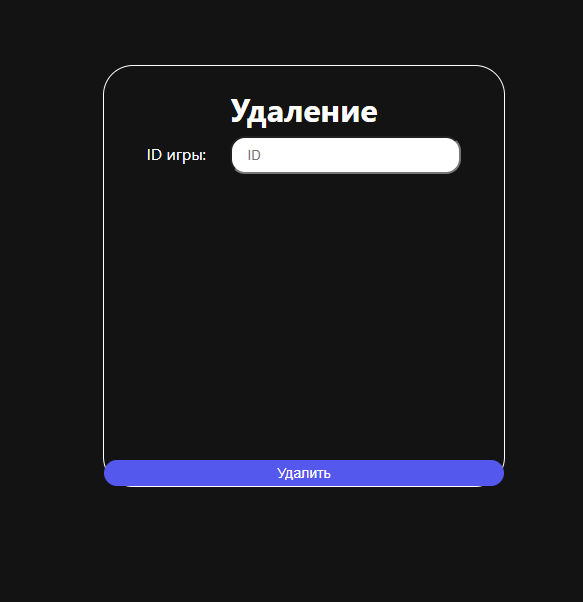
**Добавление**

****

**Изменение**

****

**Удаление**

****