**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»**

Домашнее задание

по дисциплине «Разработка интернет приложений»

Выполнил:

студент группы ИУ5-53Б

Румак Д.П.

Проверил:

Канев А.И.

2022 г.

**Содержание**

[1. Техническое задание 3](#_Toc122447531)

[Цель 3](#_Toc122447532)

[Задачи 3](#_Toc122447533)

[2. Введение 4](#_Toc122447534)

[Описание предметной области 4](#_Toc122447535)

[Актуальность 6](#_Toc122447536)

[3. Бизнес-процесс 6](#_Toc122447537)

[Диаграмма прецедентов 8](#_Toc122447538)

[Диаграмма состояний 10](#_Toc122447539)

[Диаграмма деятельности 11](#_Toc122447540)

[4. Архитектура 11](#_Toc122447541)

[Диаграмма развертывания 11](#_Toc122447542)

[ER-диаграмма с назначением таблиц 12](#_Toc122447543)

[Диаграмма компонентов 13](#_Toc122447544)

[5. Алгоритмы 14](#_Toc122447545)

[Диаграмма последовательности 14](#_Toc122447546)

[6. Описание интерфейса. Перечень окон и их назначение 15](#_Toc122447547)

# Техническое задание

# Цель

Изучение основ различных фреймворков, создание веб-приложения с удобной навигацией и интуитивным интерфейсом.

# 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 товара в корзину.
9. Добавление фильтрации и поиска на странице товаров.
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**

Система должна обеспечивать работу для трех видов пользователей: гость, авторизованный пользователь и менеджер контента.

При работе с приложением авторизованный пользователь должен иметь возможность решать следующие задачи:

1. Просматривать весь каталог игрушек.
2. Осуществлять поиск игрушек.
3. Фильтровать товары по диапазону цены.
4. Добавлять игрушки в корзину и удалять из нее.
5. Изменять количество одного товара в корзине для покупки.
6. Делать заказ на покупку игрушек.

При работе с приложением неавторизованный пользователь должен иметь возможность решать следующие задачи:

1. Просматривать каталог игрушек.
2. Осуществлять поиск по странице с товарами.
3. Фильтровать товары по цене.

При работе с приложением менеджер контента должен иметь возможность решать следующие задачи:

1. Изменять статус заказа.
2. Просмотр списка заказов.
3. Добавлять новые товары, редактировать существующие и удалять товары по причине их отсутствия или по еще какой-либо другой.

**Требования к программному обеспечению**

# Введение

# Описание предметной области

Исследуемая предметная область – веб-приложение «Интернет-магазин игрушек», с помощью которого пользователь имеет возможность просмотреть все товары каталога, выбрать и купить мягкую игрушку.

Пользователю предоставляется каталог игрушек, каждая из которых обладает следующим характеристиками: название, цена и подробное описание. Пользователь может просматривать список товаров, переходить на страницу с подробной информацией, отфильтровывать ассортимент по цене и названию, производить поиск, а также добавлять/удалять товары из корзины. Также имеется возможность авторизации и регистрации. Только авторизованный пользователь может добавлять/удалять товары из корзины.

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

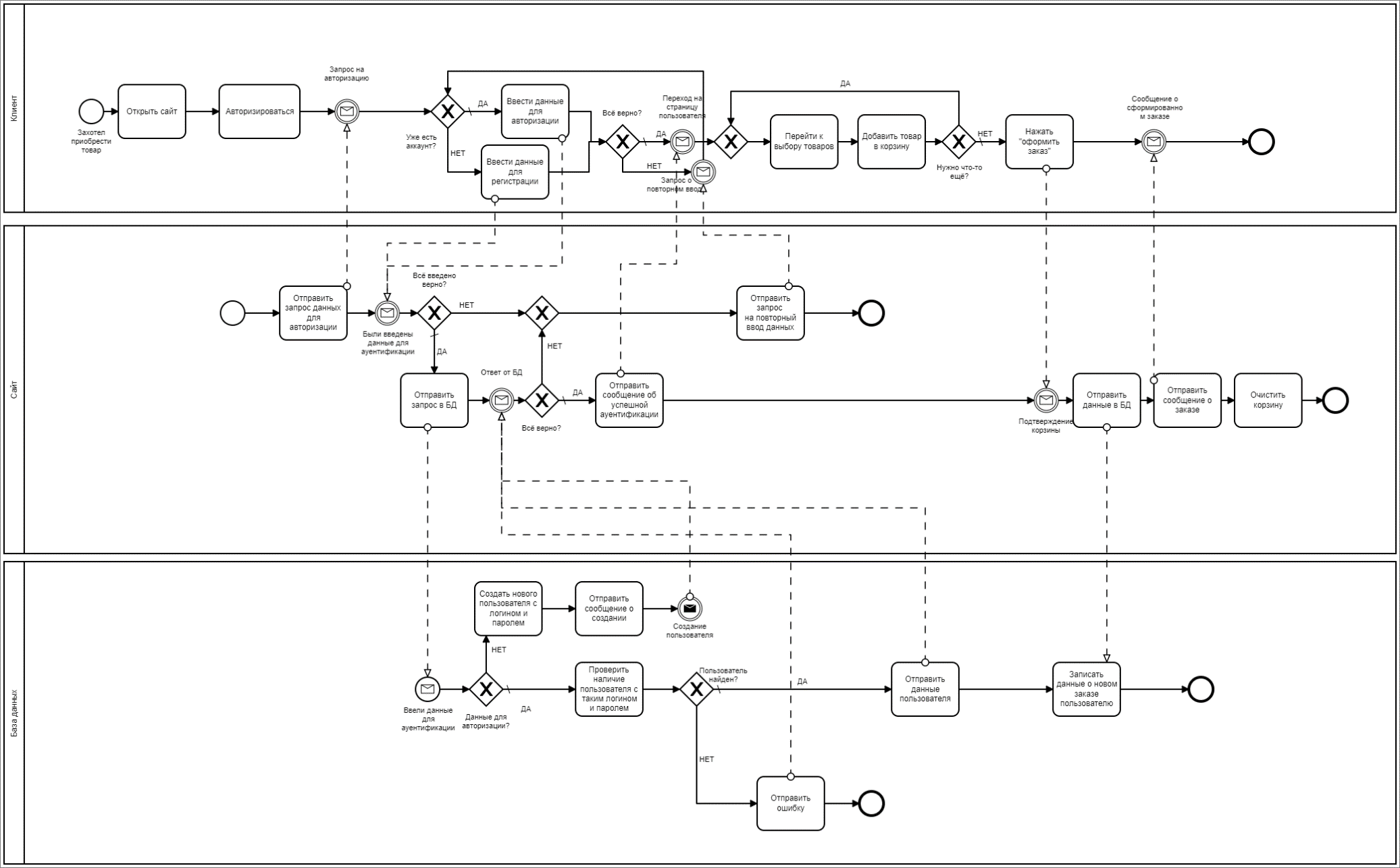
В качестве базы данных используется MySQL. Django ORM служит связью между БД и Django.

# Актуальность

Актуальность темы работы обусловлена активным внедрением и использованием интернет-магазинов. Разработанное приложение актуально для пользователей интернет-магазинов мягких игрушек, так как не забито лишней информацией и упор делается на правтичность. Удобство работы заключается в возможности просмотра, поиска, фильтрации товаров, добавления их в корзину и просмотра заказов.

# Бизнес-процесс

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, найти нужный товар, отфильтровать их и положить в корзину.

# Диаграмма прецедентов

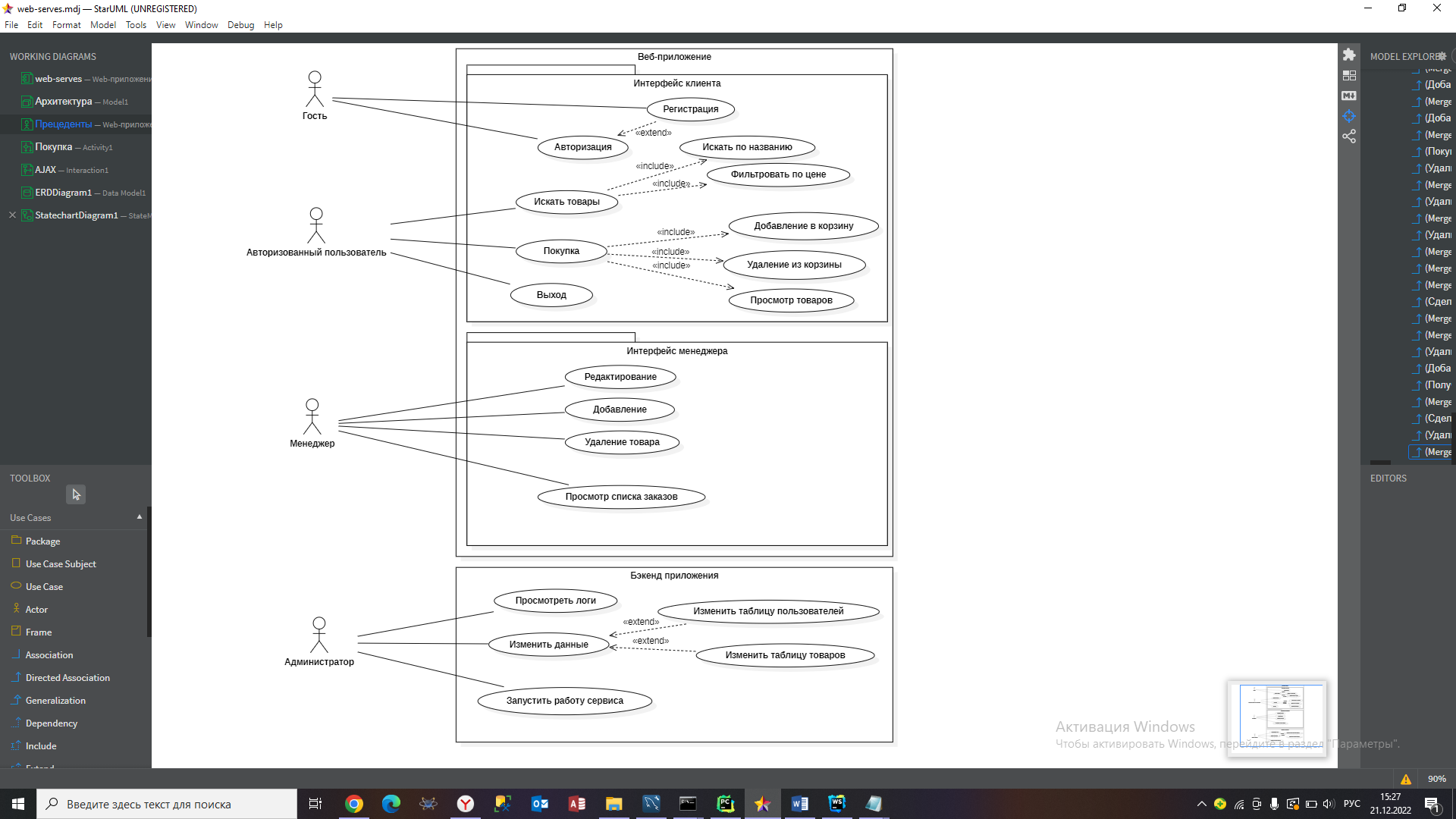


Рисунок 1 – Диаграмма прецедентов

На диаграмме прецендентов мы можем наглядно рассмотреть возможности разных типов пользователя, таких как:

1. Гость
2. Авторизованный пользователь
3. Менеджер
4. Администратор

# Диаграмма состояний

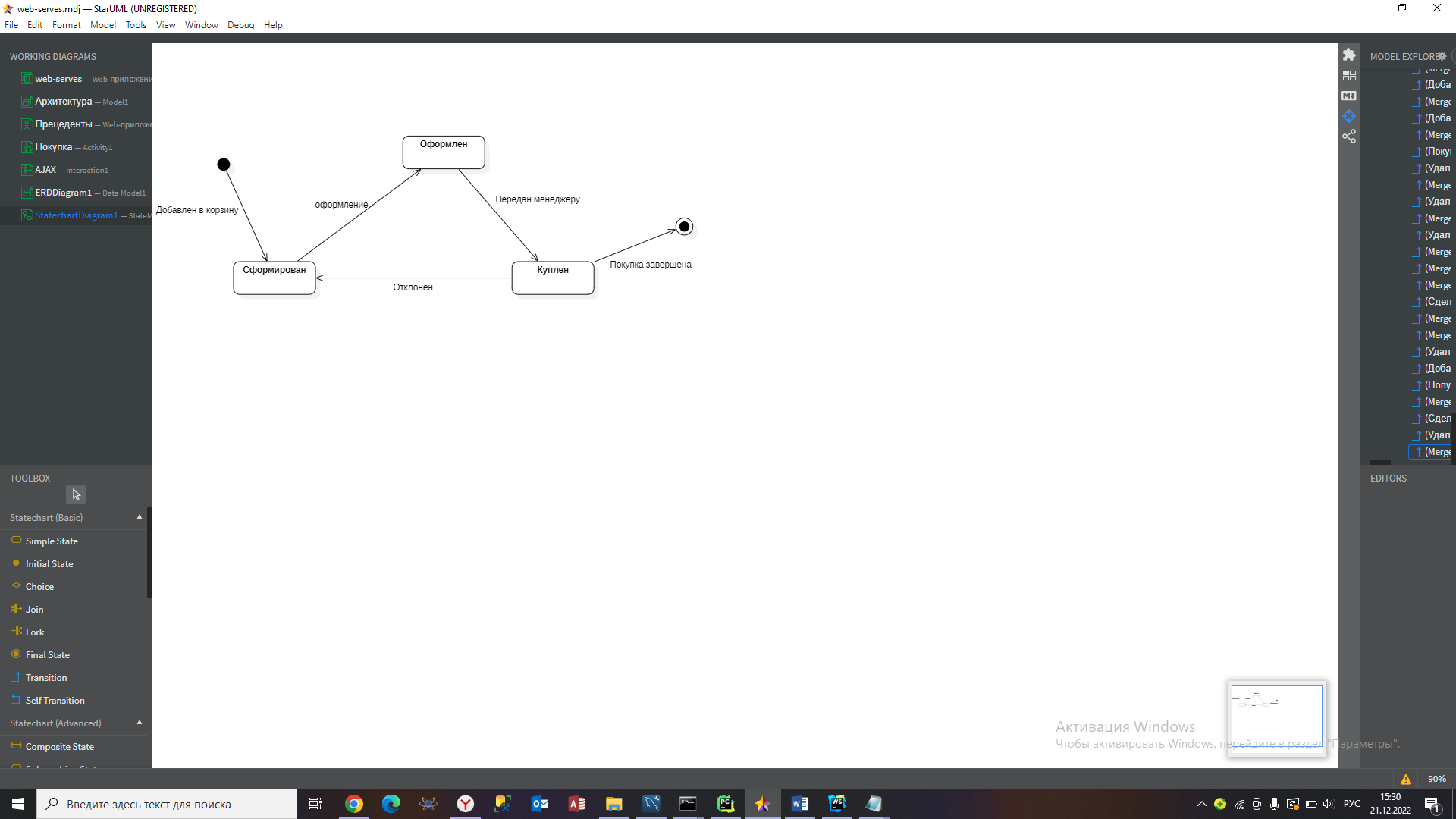


Рисунок 4 – Диаграмма состояний

На представленной диаграмме состояний видим, через какие этапы проходит процесс покупки выбранного/выбранных товаров:

1. Сформирован
2. Оформлен
3. Подтвержден

В конечном итоге покупка либо будет совершена, либо удален, либо заказ будет отклонен и покупателю придется сформировывать заказ по новой.

# Диаграмма деятельности

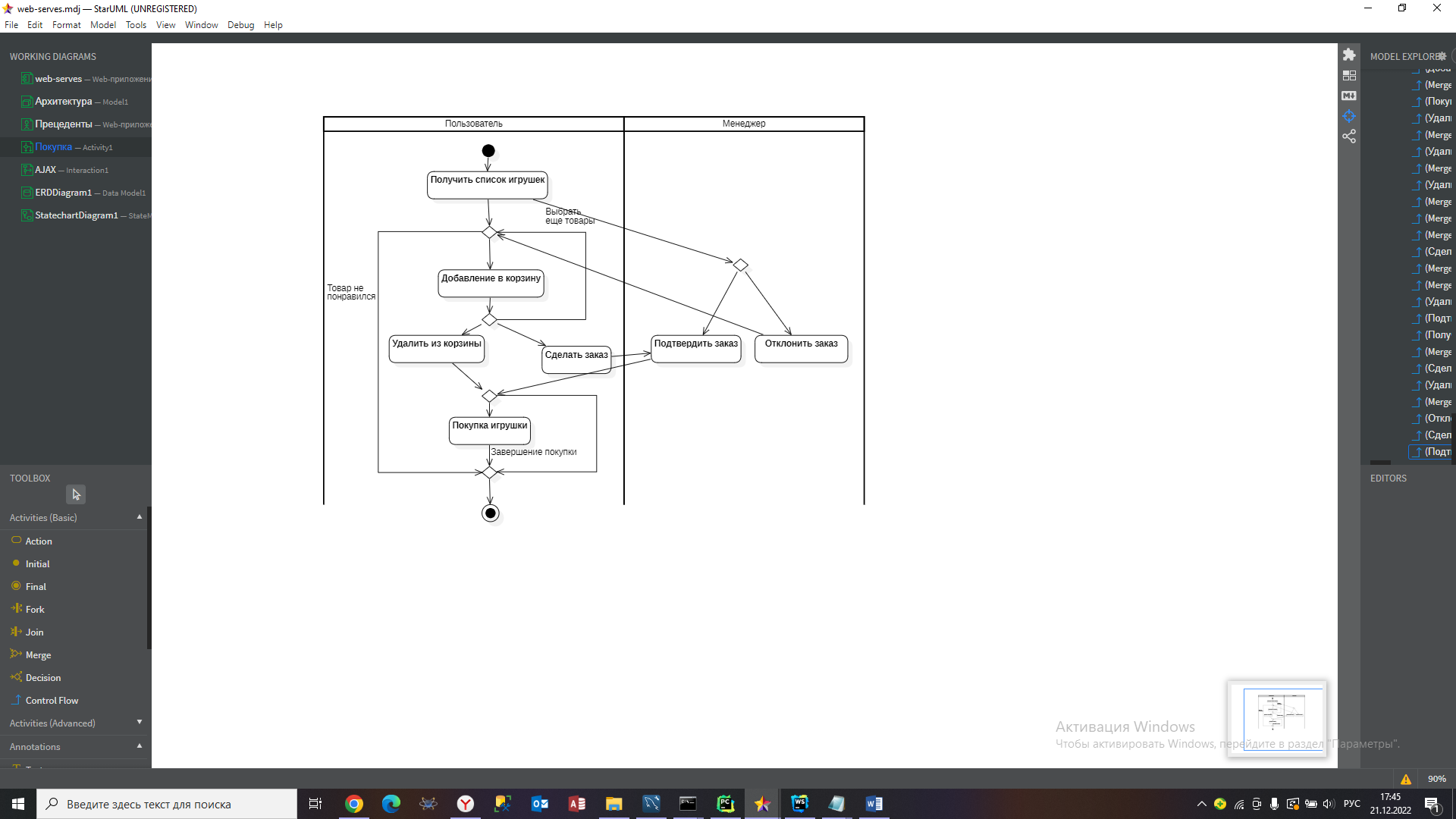


Рисунок 4 – Диаграмма деятельности

На данной диаграмме описан процесс добавления пользователем товара в корзину и его покупка. Описана работа менеджера, который имеет возможность менять статус заказа. Результатом может быть отклонение или одобрение.

# Архитектура

# Диаграмма развертывания

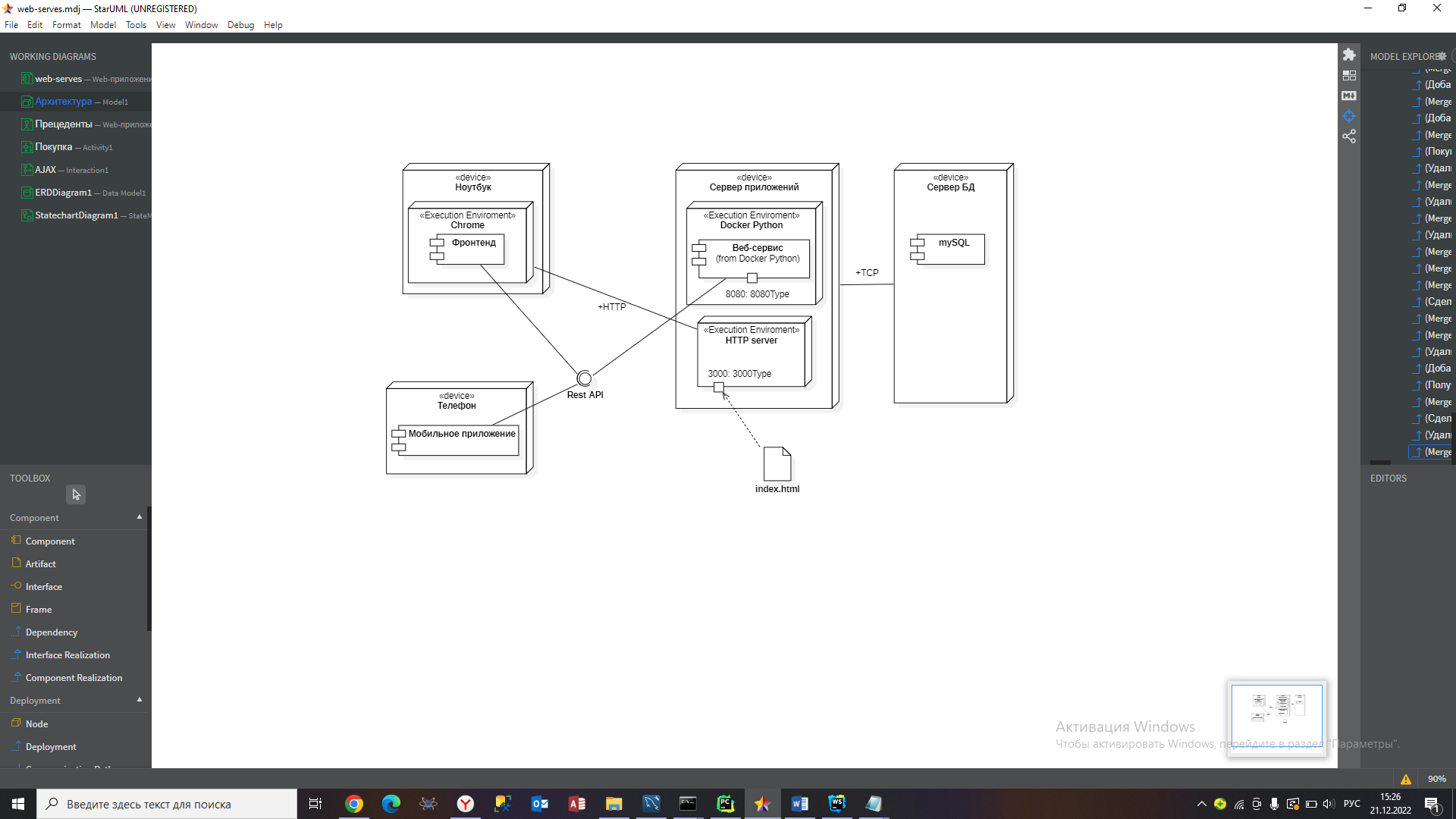


Рисунок 4 – Диаграмма развертывания

Диаграмма развертывания описывает общую структуру приложения, связь между сервером приложения и базой данных, и устройствами. Здесь мы наблюдаем архитектуру исполнения системы. В отличии от большинства других диаграмм на диаграмме развертывания мы наблюдаем аппаратную топологию системы. В нашей системе мы имеем сервер БД выполненный в MySQL, также бэкенд выполненный в фреймворке Django на Python, который с помощью 3000-го порта соединен с фронтендом.

# ER-диаграмма с назначением таблиц

ER-диаграмма показывает архитектуру БД. Были созданы три таблицы: Toys (Игрушки), Users (Пользователи), Basket (Корзина). Таблица «Пользователи» содержит в себе информацию о пользователях веб-приложения. Таблица «Игрушки» содержит в себе информацию о каждом товаре (игрушке). Таблица «Корзина» содержит в себе информацию о пользователях и добавленных ими игрушках, также количество одного товара в корзине.

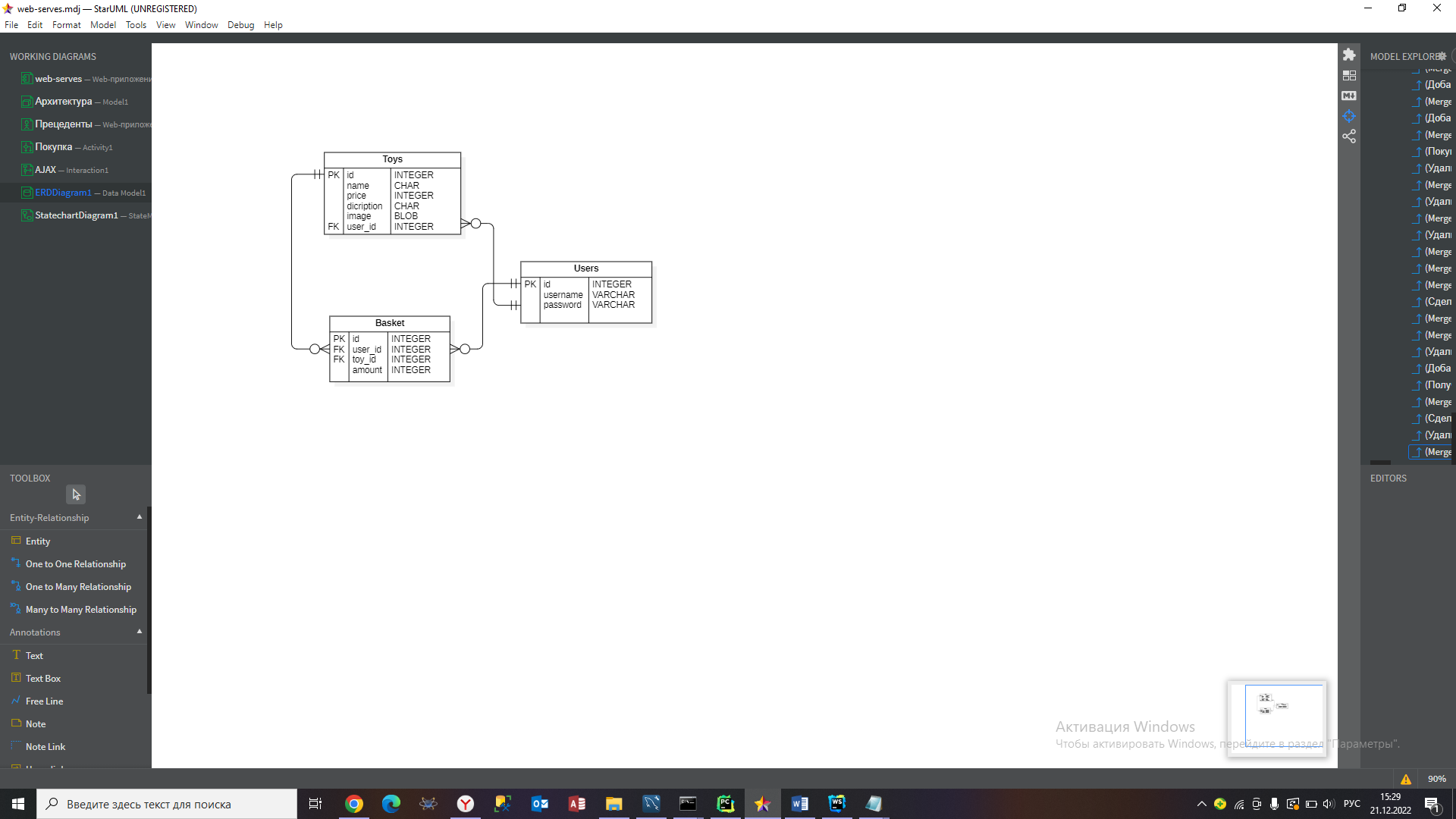


Рисунок 4 – ER-диаграмма

# Диаграмма компонентов

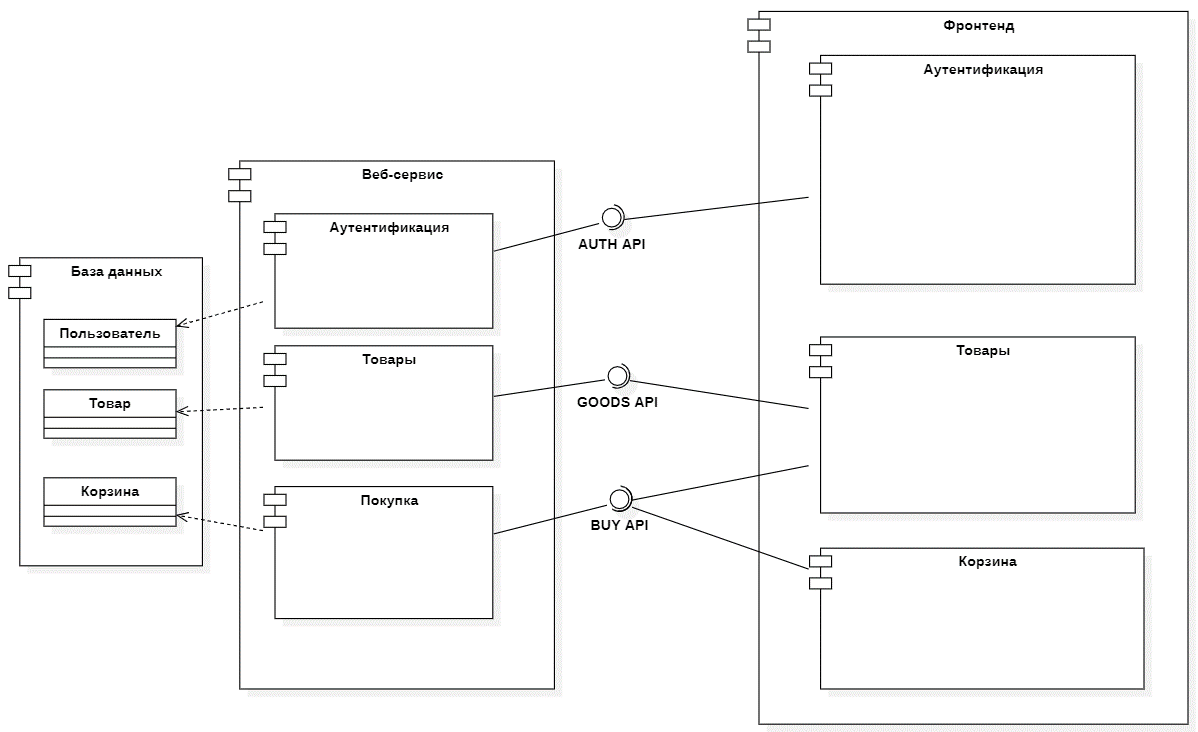


Рисунок 4 – Диаграмма компонентов

На диаграмме компонентов отражено взаимодействие между идентичными объектами веб-сервиса и фронтенда, а также связь объектов веб-сервиса с базой данных. Диаграмма компонентов используются для визуализации организации компонентов системы и зависимостей между ними.

# Алгоритмы

# Диаграмма последовательности

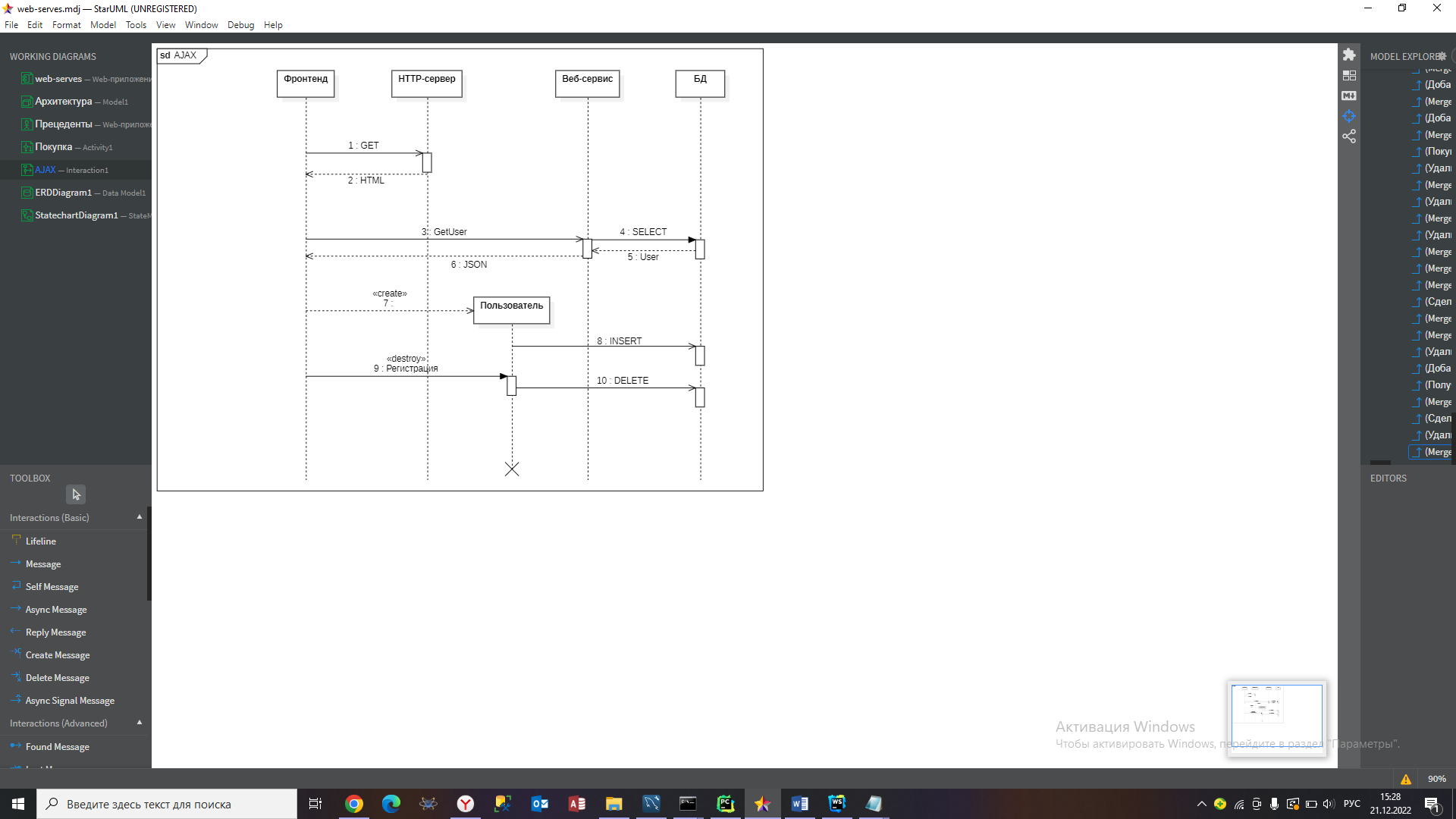


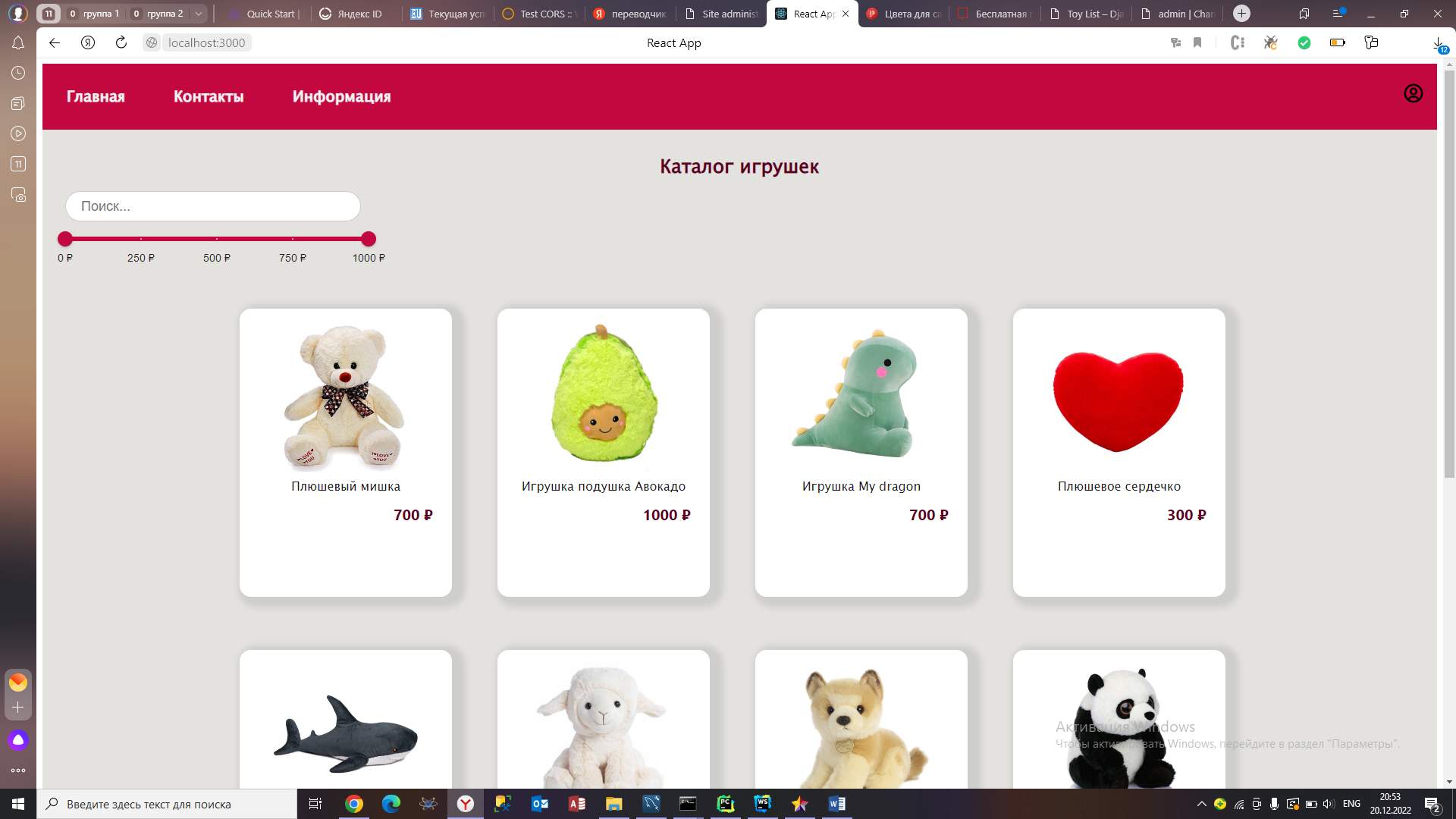
Рисунок 4 – Диаграмма последовательности

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

# Описание интерфейса. Перечень окон и их назначение

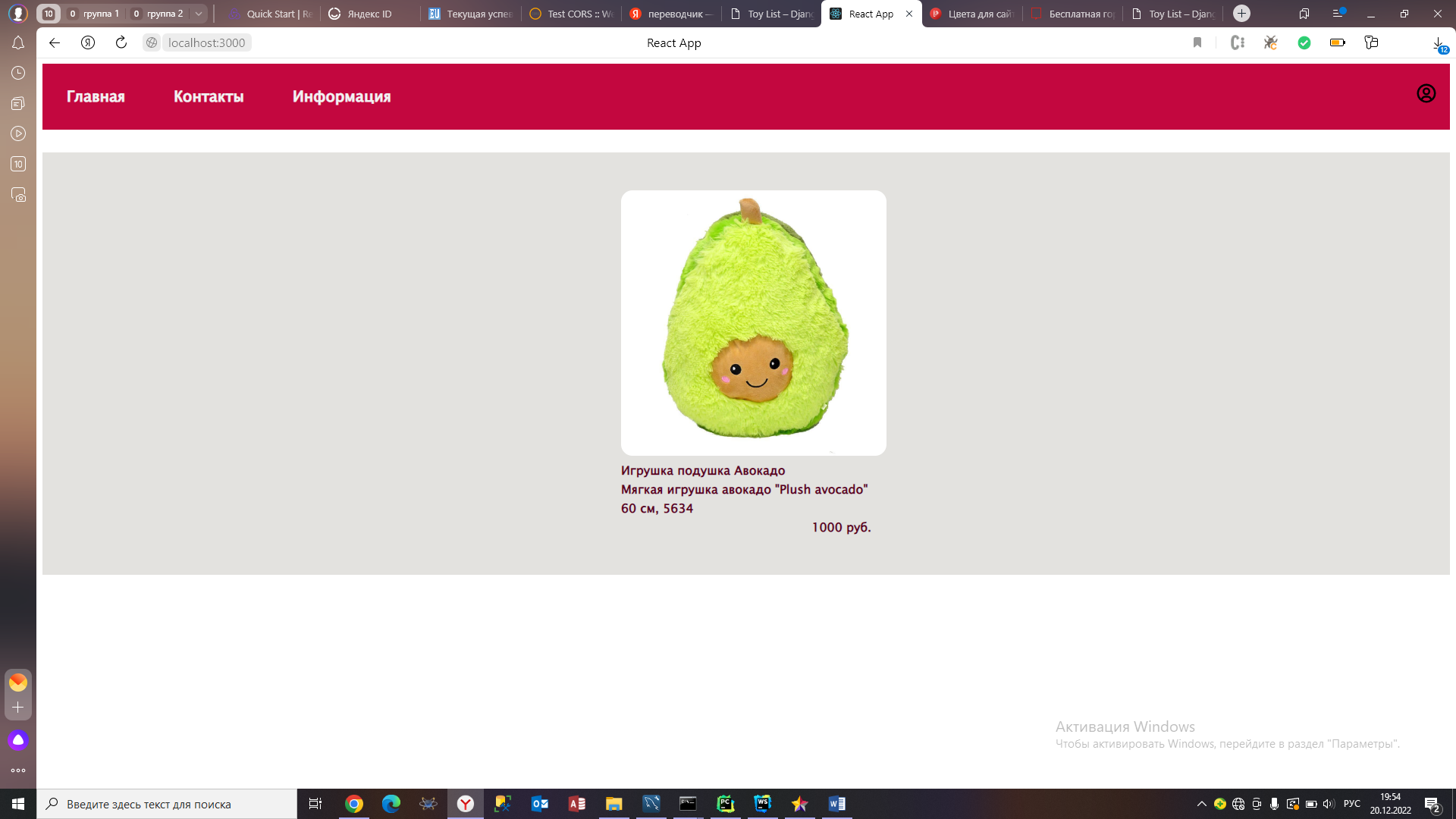
**Начальное окно**

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



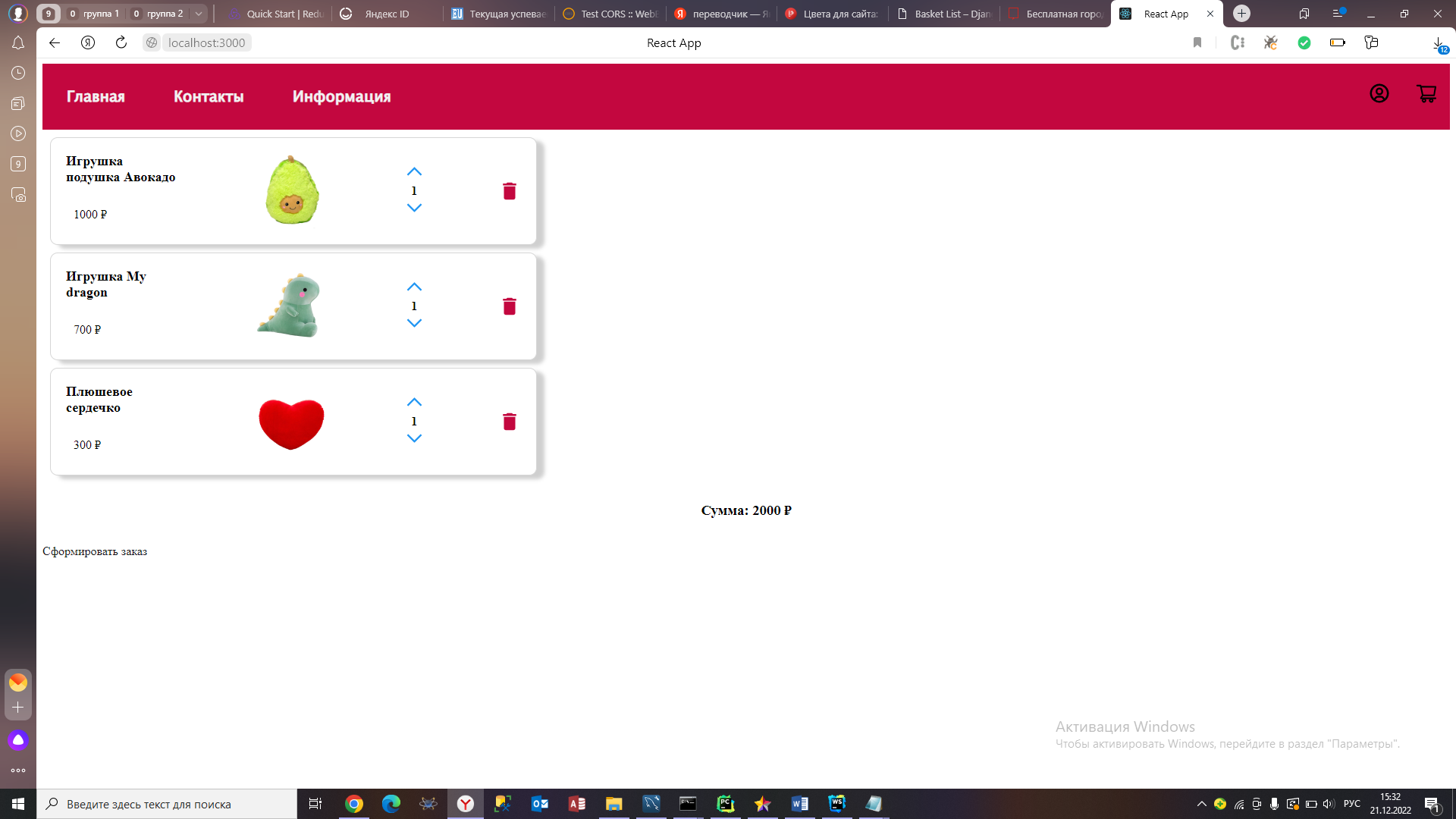
**Страница с подробной информацией**

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



**Корзина**

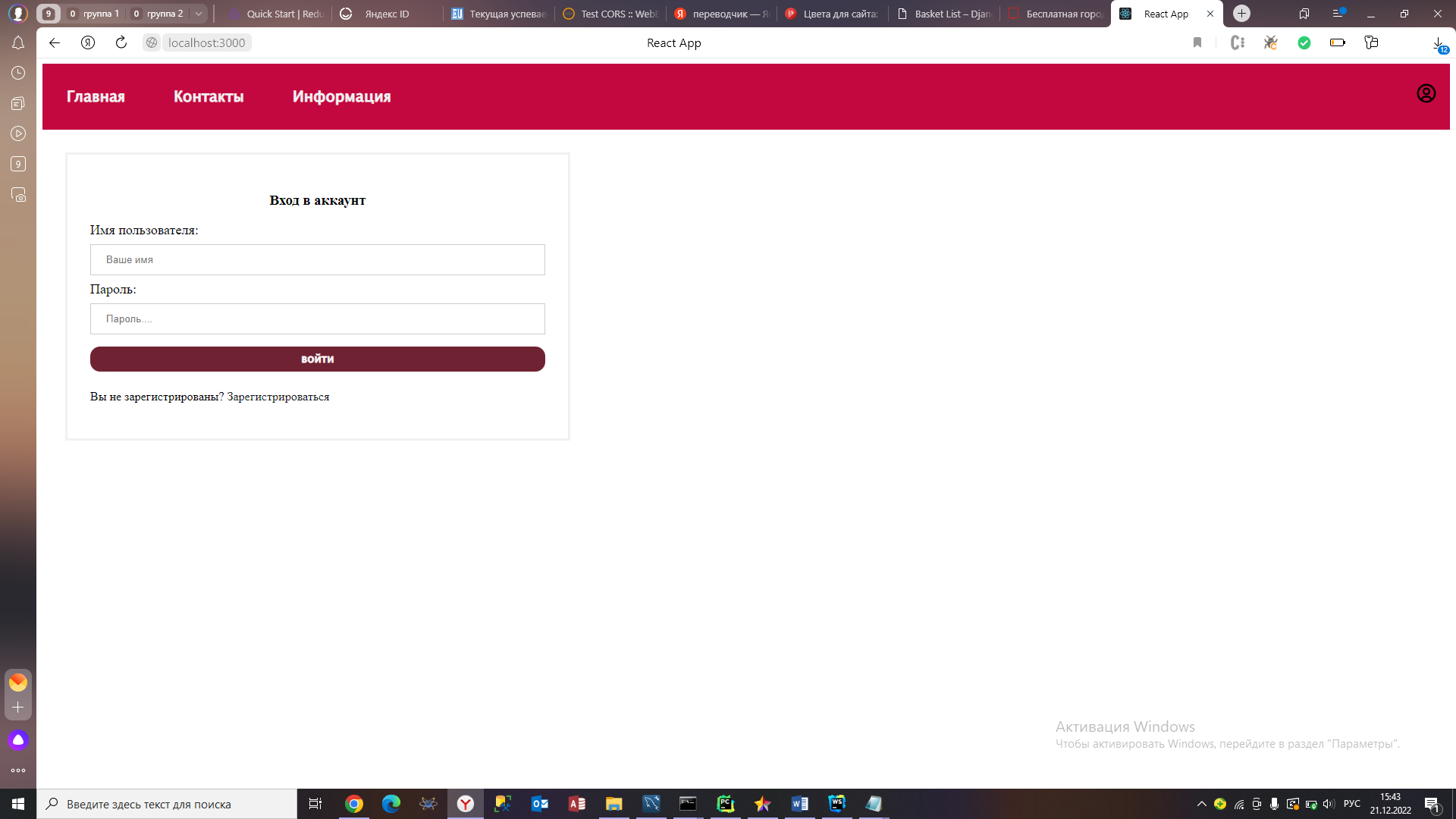
В корзине можно посмотреть добавленные игры и общую цену покупки



**Авторизация**

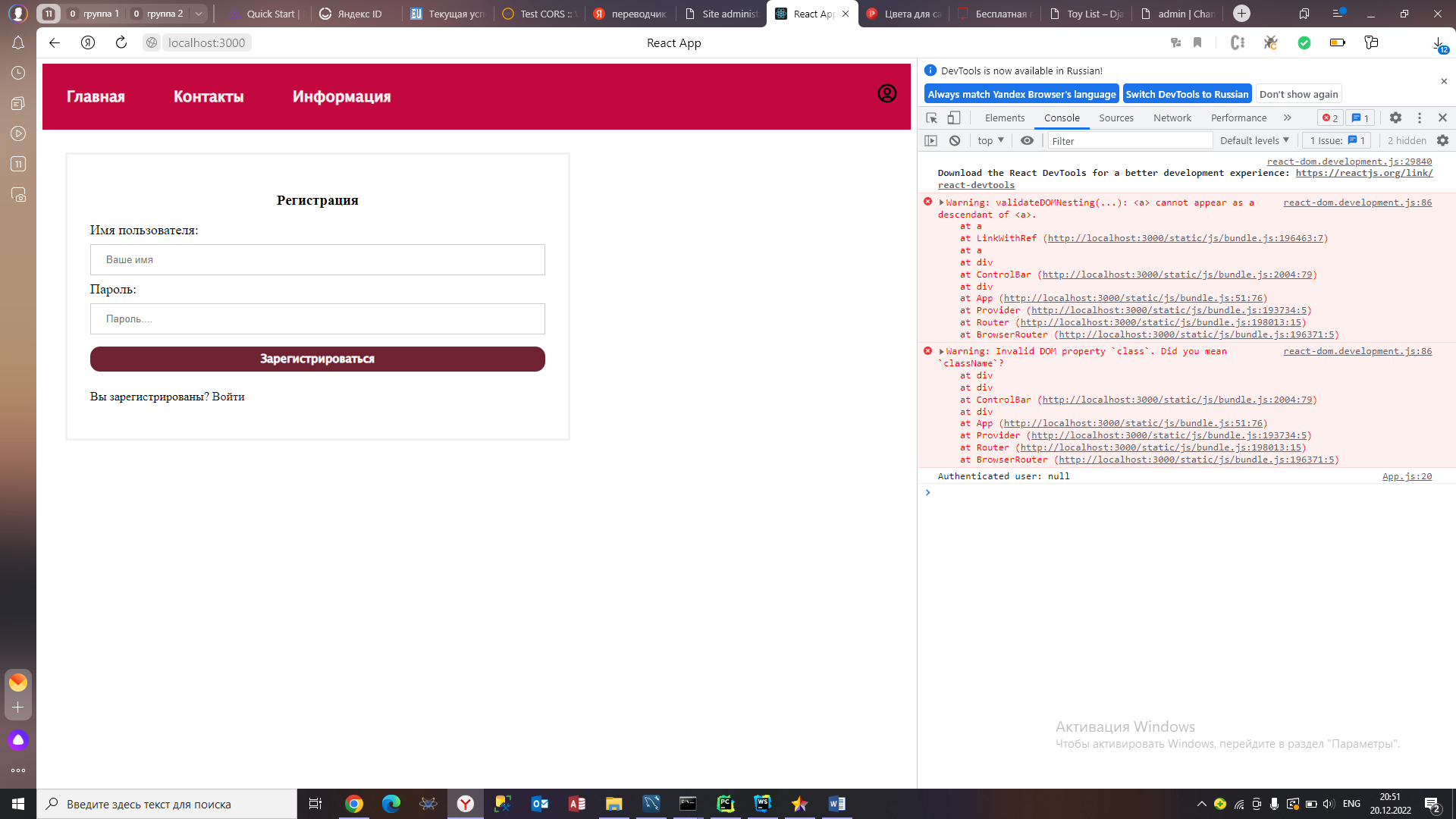
**Вход**

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



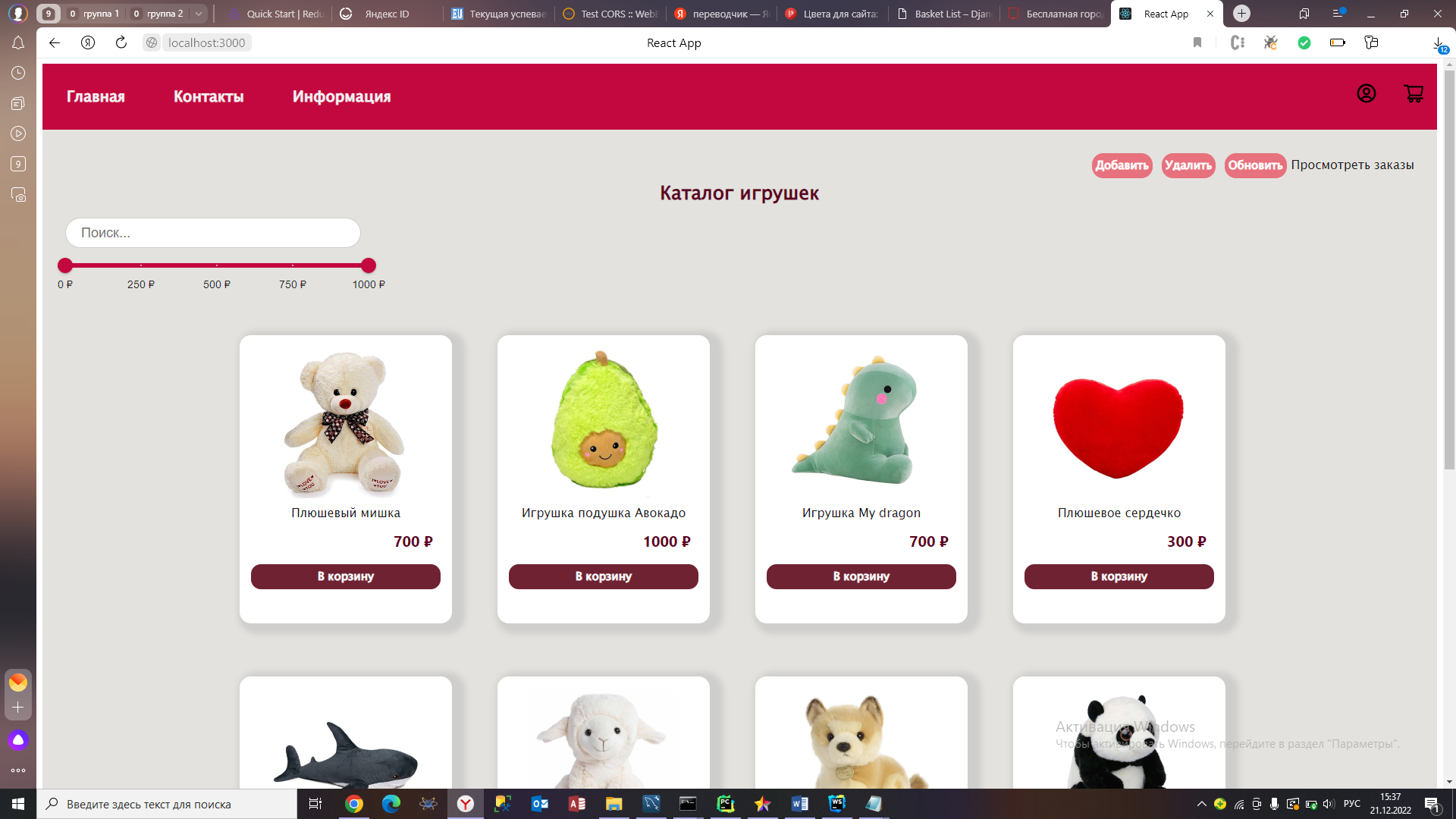
**Регистрация**

У гостя запрашиваются личные данные, необходимые для регистрации: 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.



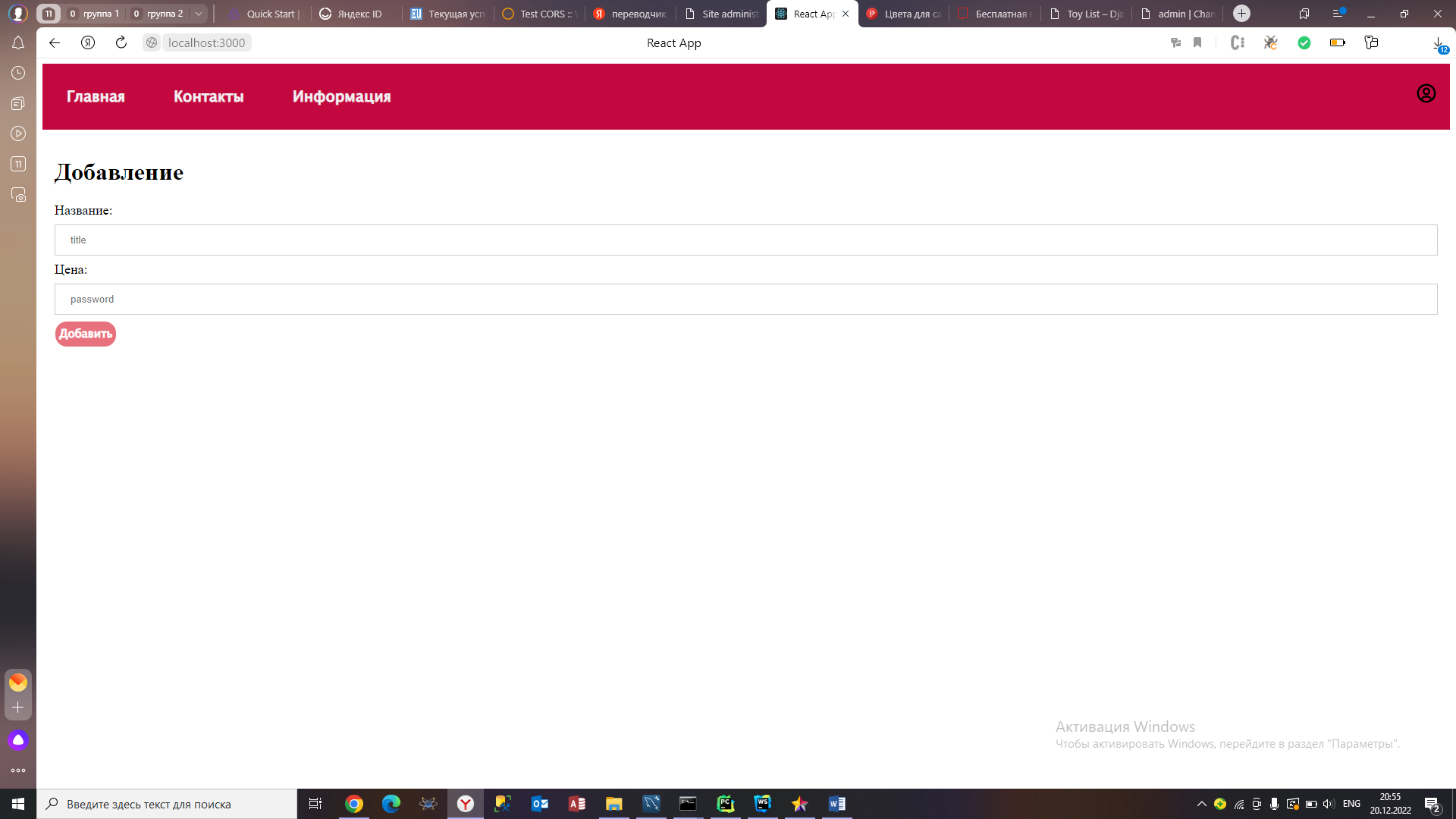
**Главное окно менеджера и администратора**

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

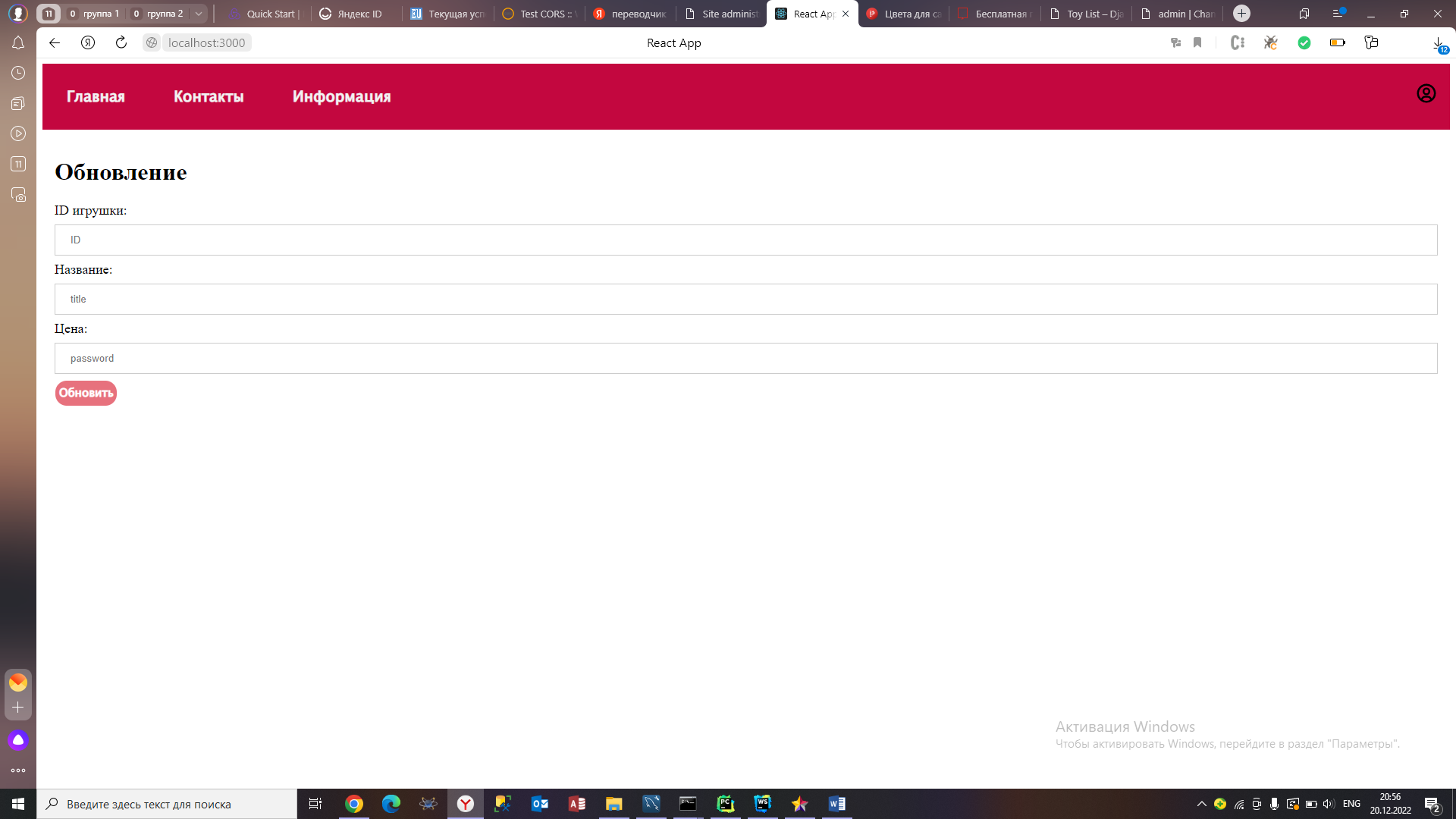


**Добавление, изменение, удаление товаров, просмотр заказов**

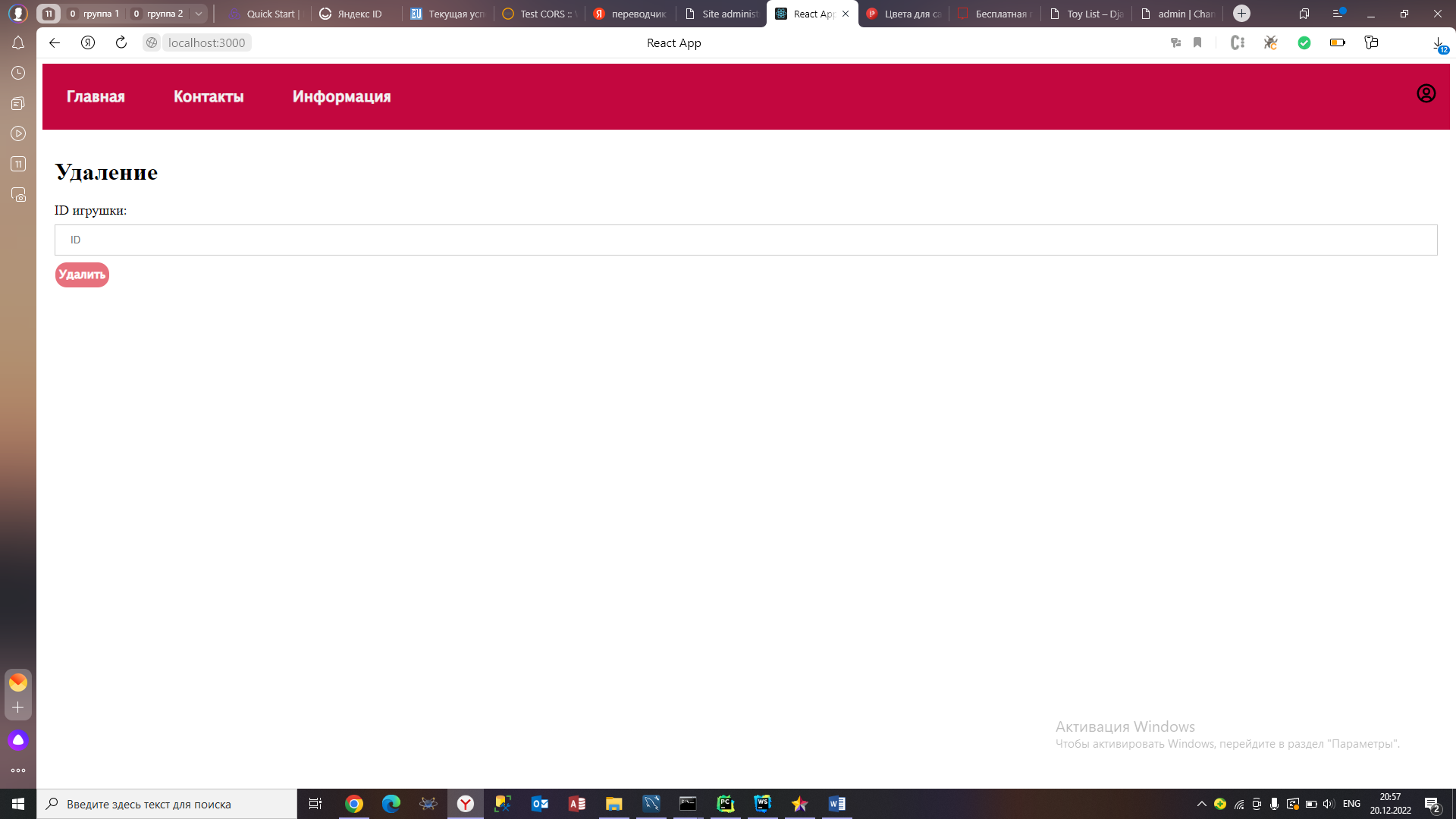
**Добавление**



**Изменение**



**Удаление**



**Просмотр заказов**

