

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук
Кафедра информационных систем
Интернет магазин футболок «MarketVSU»

Курсовой проект

09.03.02 Информационные системы и технологии
Информационные системы и сетевые технологии

Зав. кафедрой _____ д. ф.-м.н., профессор Э.К. Алгазинов

Обучающийся _____ *И.Е. Лесных*

Обучающийся _____ *Г.О. Латынин*

Обучающийся _____ *А.А. Товмасын*

Руководитель _____ *В.С. Тарасов.*

Воронеж 2020

Содержание

Содержание	2
Введение	3
1. Постановка задачи	4
2. Анализ предметной области	6
2.1 Анализ существующих решений	6
2.1.1 СпортМастер	6
2.1.2 Lamoda	6
2.1.3 Asos	7
2.2 Анализ задачи	8
2.2.1 Варианты использования приложения	8
2.2.2 Взаимодействие компонентов системы	10
2.2.3 Взаимодействие компонентов системы	14
2.2.4 Взаимодействие действия в системе	15
2.2.5 Модель базы данных	17
2.2.6 Развертывание приложения	18
4. Интерфейс	20
4.1 Интерфейс неавторизованного пользователя	20
4.3 Интерфейс пользователя, авторизованного под ролью «Администратор»	25
Заключение	28

Введение

В современном мире мы наблюдаем постоянный рост рынка одежды. Компаниям и предпринимателям необходимо улучшать свой продукт и улучшать методы его сбыта, чтобы выдерживать высокую конкуренцию на рынке.

С приходом интернета в повседневную жизнь человека все больше компаний привлекают его себе в помощь, в том числе и магазины одежды. Он позволил расширить территорию сбыта продукции вплоть до всего мира и не удивительно, что почти все магазины создают свои веб-сайты для реализации своей продукции. Эти интернет магазины имеют очевидные преимущества по сравнению и их аналогами в реальном мире. Главным преимуществом онлайн-магазинов для покупателей является отсутствие нужды в непосредственном присутствии в магазине или торговом центре. А для владельцев магазина одним из ключевых факторов является то, что снижаются издержки на содержание магазина, так как отсутствует арендная плата за место для магазина и заработная плата продавцов, так как большую часть их обязанностей берет на себя веб-приложение.

При разработке проекта используется каскадная модель, которая включает в себя следующие этапы:

1. Определение требований.
2. Проектирование.
3. Конструирование (также «реализация» либо «кодирование»).
4. Воплощение.
5. Тестирование и отладка (также «верификация»).
6. Инсталляция.
7. Поддержка.

Также в процессе разработки будет использоваться парадигма объектно-ориентированного программирования.

1. Постановка задачи

Целью курсового проекта является создание веб-приложения, выполняющего функции интернет-магазина по продаже футболок.

В системе предусмотрены три уровня доступа:

1. Неавторизованный пользователь;
2. Клиент;
3. Администратор.

Неавторизованный пользователь обладает следующими возможностями:

- Регистрация;
- Авторизация;
- Просмотр товаров из каталога;
- Добавление товаров в корзину;
- Удаление товаров из корзины.

Клиент обладает следующими возможностями:

- Просмотр товаров из каталога;
- Добавление товаров в корзину;
- Удаление товаров из корзины;
- Оформление заказа;
- Изменение личных данных;
- Выход из системы.

Администратор обладает следующими возможностями:

- Добавление товаров в каталог;
- Изменение товаров в каталоге;
- Удаление товаров из каталога;
- Обработка заказа:

1. Прием/Отклонение заказа;
 2. Изменение статуса заказа.
- Изменение личных данных;
 - Выход из системы.

2. Анализ предметной области

2.1 Анализ существующих решений

2.1.1 СпортМастер

СпортМастер является одним из ключевых игроков на российском рынке спортивных товаров.

Достоинства:

1. Наличие личного кабинета;
2. С помощью различных фильтров легко можно найти интересующий товар;
3. Наличие раздела «Отзывы»;
4. Наличие приложения для мобильных устройств;
5. Огромный ассортимент товаров;
6. Наличие скидочной системы;
7. Наличие доставки;
8. Возможность возврата товара.

Недостатки:

1. Довольно сложный интерфейс;
2. Отсутствие собственного бренда одежды.

2.1.2 Lamoda

Lamoda является одним из ключевых игроков на российском рынке онлайн магазинов одежды.

Достоинства:

1. Наличие личного кабинета;
2. Наличие отличного фильтра для поиска товара;
3. Огромный ассортимент товаров;
4. Наличие скидочной системы;
5. Наличие раздела «Отзывы»;

6. Наличие приложения для мобильных устройств;
7. Наличие доставки с примеркой перед покупкой.
8. Возможность возврата товара.

Недостатки:

1. Отсутствие собственного бренда одежды.

2.1.3 Asos

Asos является одним из ключевых игроков на российском рынке онлайн магазинов одежды, который имеет собственный бренд одежды.

Достоинства:

1. Наличие личного кабинета;
2. Наличие отличного фильтра для поиска товара;
3. Наличие приложения для мобильных устройств;
4. Наличие скидочной системы;
5. Наличие доставки;
6. Наличие собственного бренда одежды;
7. Возможность возврата товара.

Недостатки:

1. Отсутствие раздела «Отзывы».

2.2 Анализ задачи

2.2.1 Варианты использования приложения

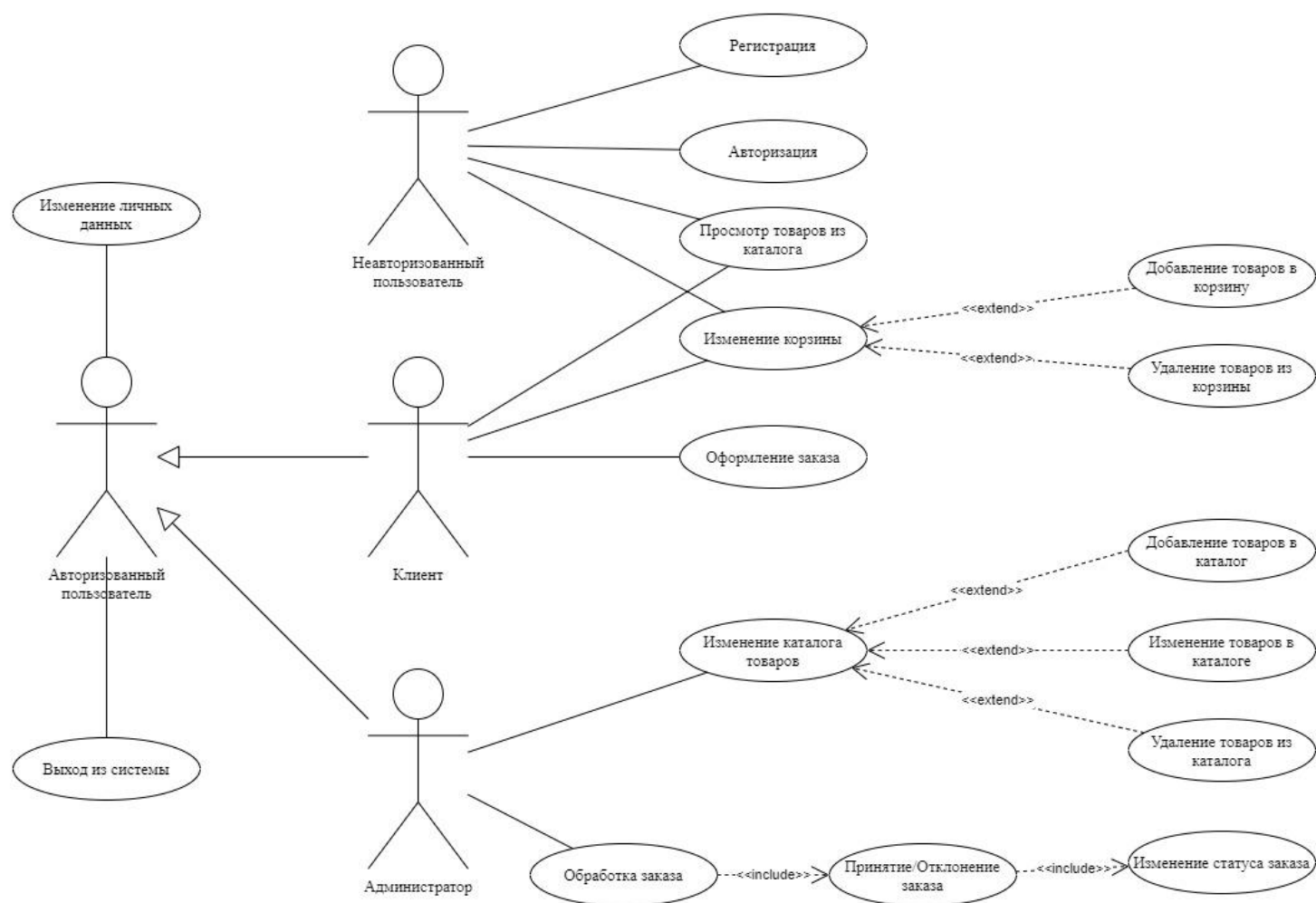


Рисунок 1. Диаграмма прецедентов

Пользователь имеет возможность взаимодействовать с приложением без авторизации. При таком сценарии ему будут доступны следующие действия:

1. Регистрация;
2. Авторизация;
3. Просмотр товаров из каталога;
4. Добавление товаров в корзину;
5. Удаление товаров из корзины.

Также пользователь имеет возможность авторизоваться под ролью «Администратор» или ролью «Клиент». Если пользователь авторизуется под ролью «Администратор», то ему будут доступны следующие действия:

1. Изменение личных данных;
2. Добавление товаров в каталог;
3. Изменение товаров в каталоге;
4. Удаление товаров из каталога;
5. Обработка заказа:
 - Прием/Отклонение заказа;
 - Изменение статуса заказа.
6. Выход из системы.

Если же пользователь авторизуется под ролью «Клиент», то ему будут доступны следующие действия:

1. Изменение личных данных;
2. Добавление товаров в корзину;
3. Удаление товаров из корзины;
4. Оформление заказа;
5. Выход из системы.

2.2.2 Взаимодействие компонентов системы

Взаимодействие неавторизованного пользователя с системой:

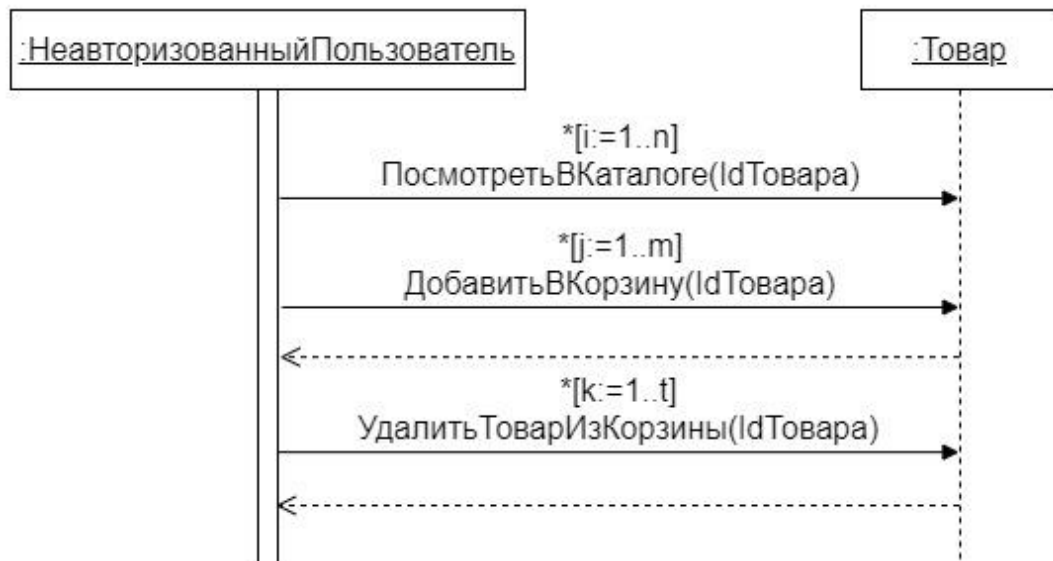


Рисунок 2. Диаграмма последовательностей для неавторизованного пользователя

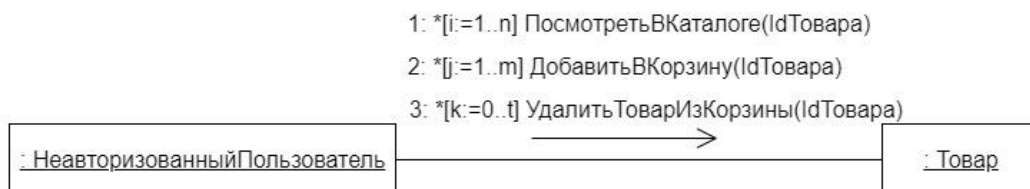


Рисунок 3. Диаграмма взаимодействия для неавторизованного пользователя

На рисунке 2 показана диаграмма последовательностей, на которой изображено упорядоченное во времени взаимодействие объектов.

На рисунке 3 показана диаграмма взаимодействия, на которой явно указываются отношения между объектами.

Взаимодействие авторизованного под ролью «Клиент» пользователя:

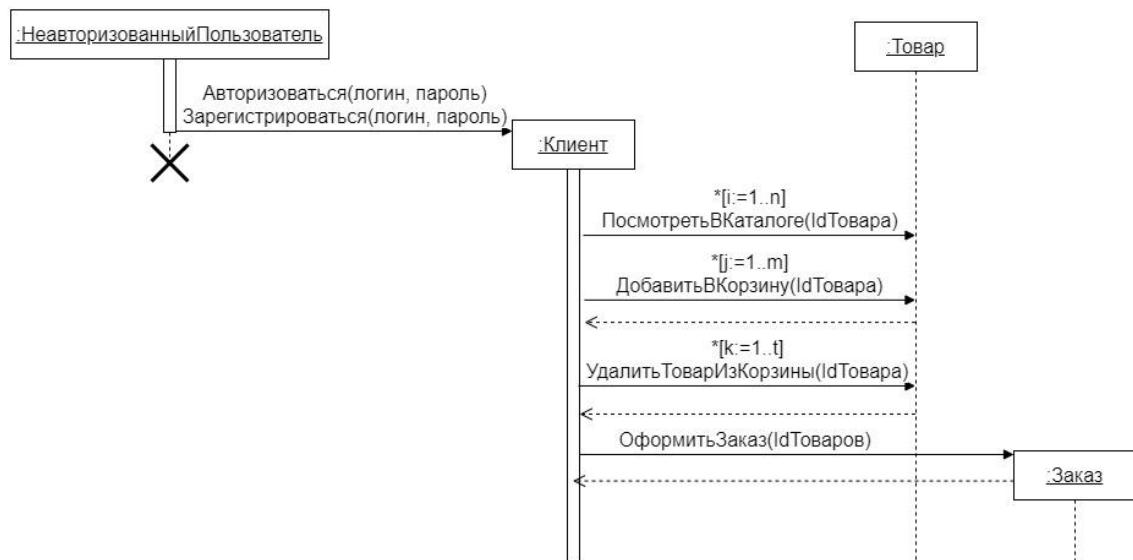


Рисунок 4. Диаграмма последовательностей для авторизованного под ролью «Клиент» пользователя

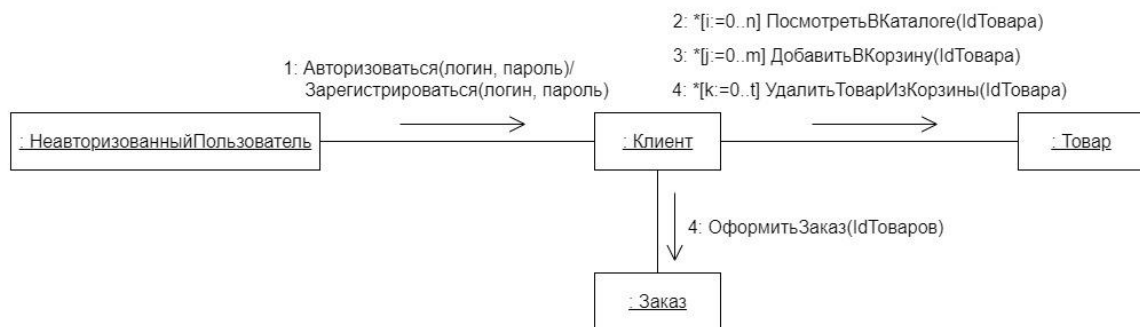


Рисунок 5. Диаграмма взаимодействия для авторизованного под ролью «Клиент» пользователя

На рисунке 4 показана диаграмма последовательностей, на которой изображено упорядоченное во времени взаимодействие объектов.

На рисунке 5 показана диаграмма взаимодействия, на которой явно указываются отношения между объектами.

Взаимодействие авторизованного под ролью «Администратор»

пользователя:





Рисунок 6. Диаграмма последовательностей для авторизованного под ролью «Администратор» пользователя

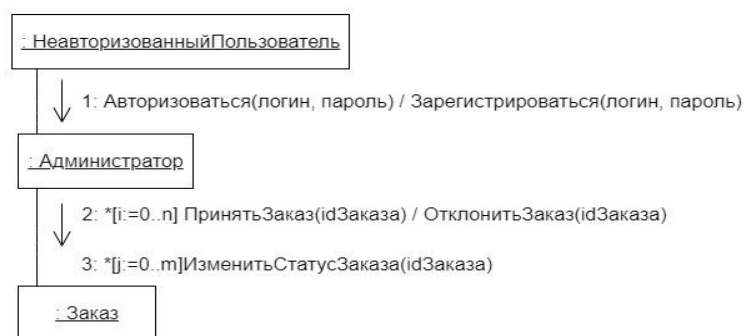
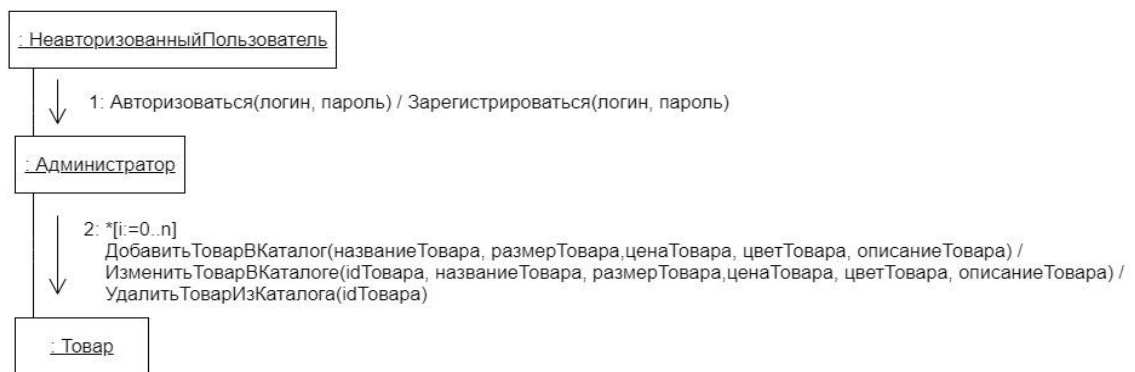
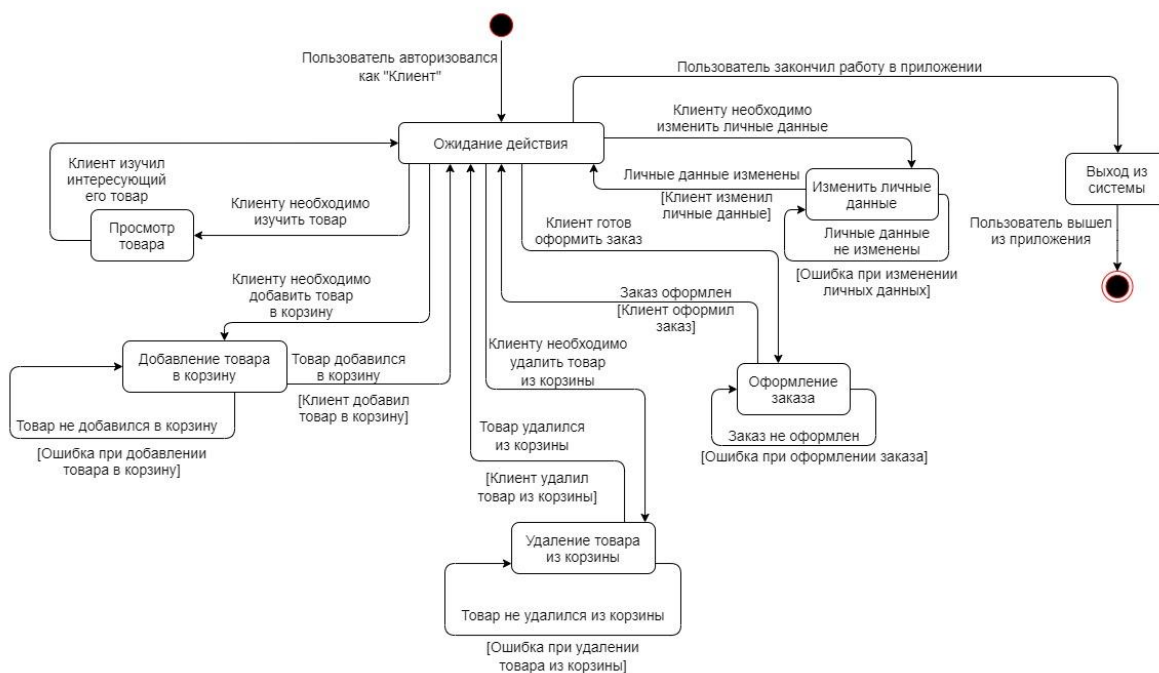
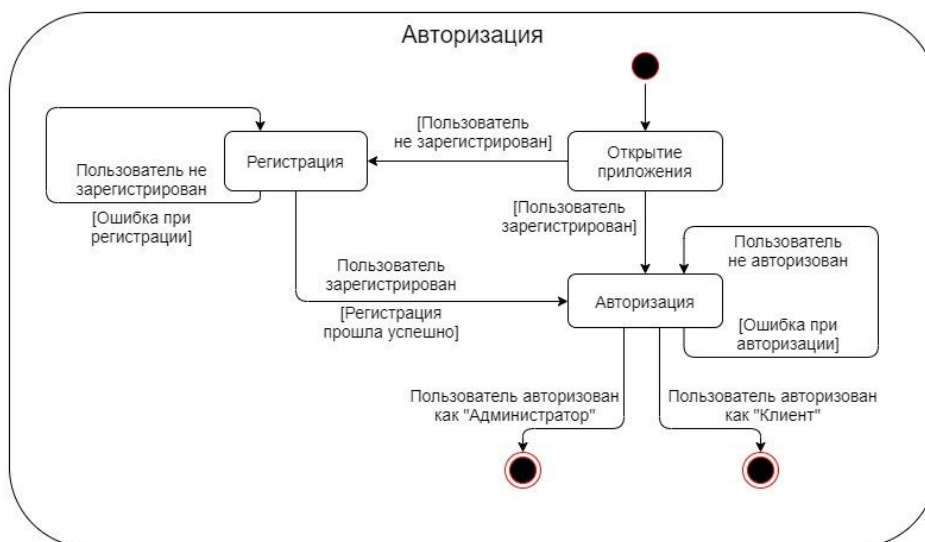


Рисунок 7. Диаграмма взаимодействия для авторизованного под ролью «Администратор» пользователя

На рисунке 6 показана диаграмма последовательностей, на которой изображено упорядоченное во времени взаимодействие объектов.

На рисунке 7 показана диаграмма взаимодействия, на которой явно указываются отношения между объектами.

2.2.3 Взаимодействие компонентов системы



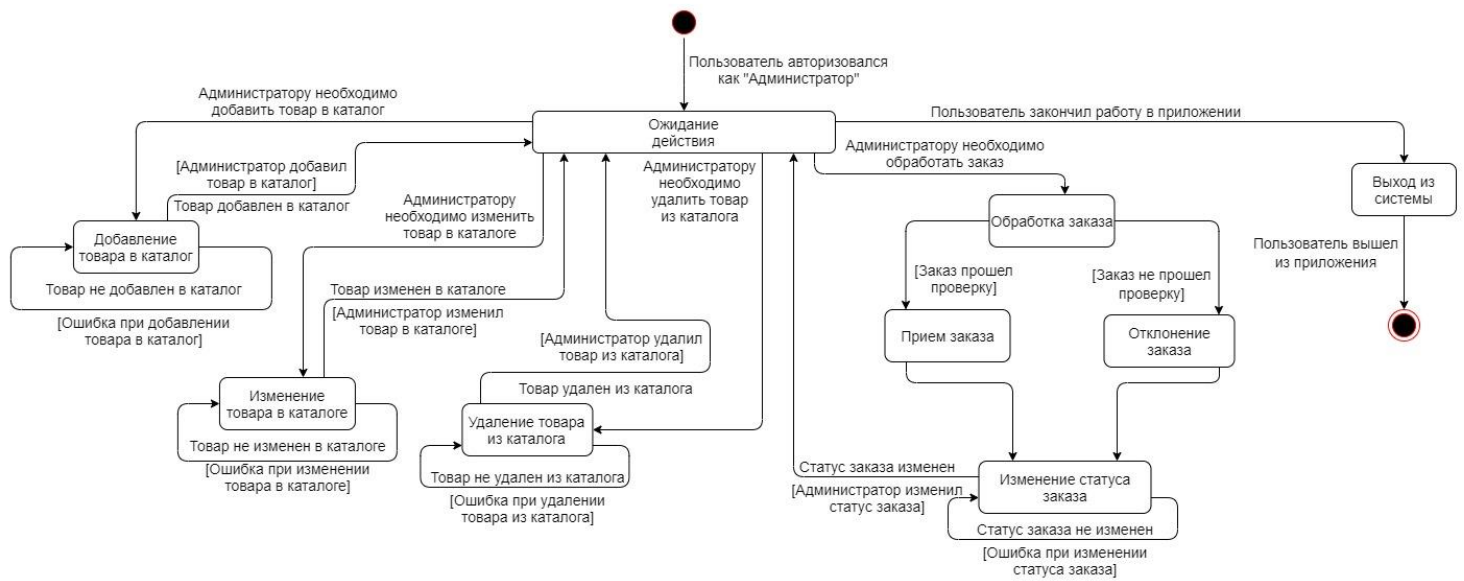
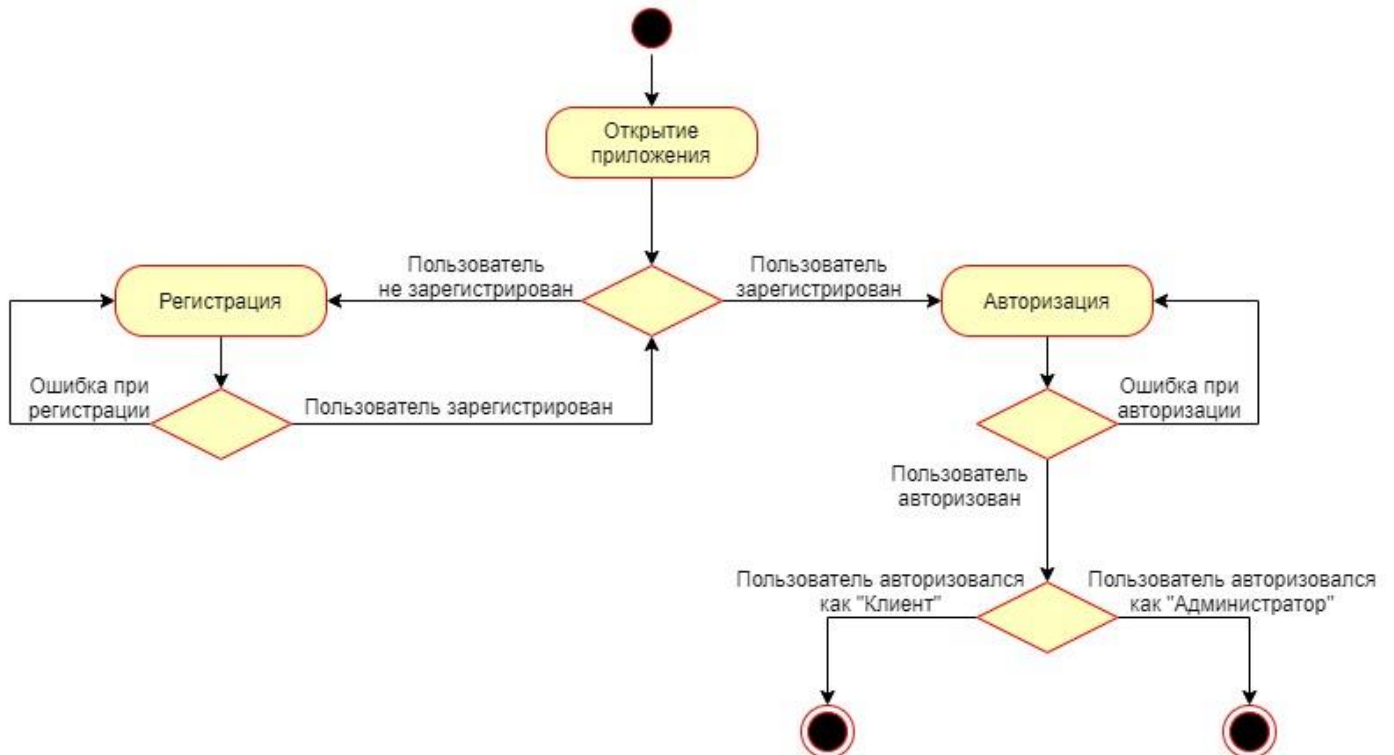


Рисунок 8. Диаграмма состояния

Диаграмма состояний, изображенная на Рисунке 8, отражает возможные состояния системы.

2.2.4 Взаимодействие действия в системе



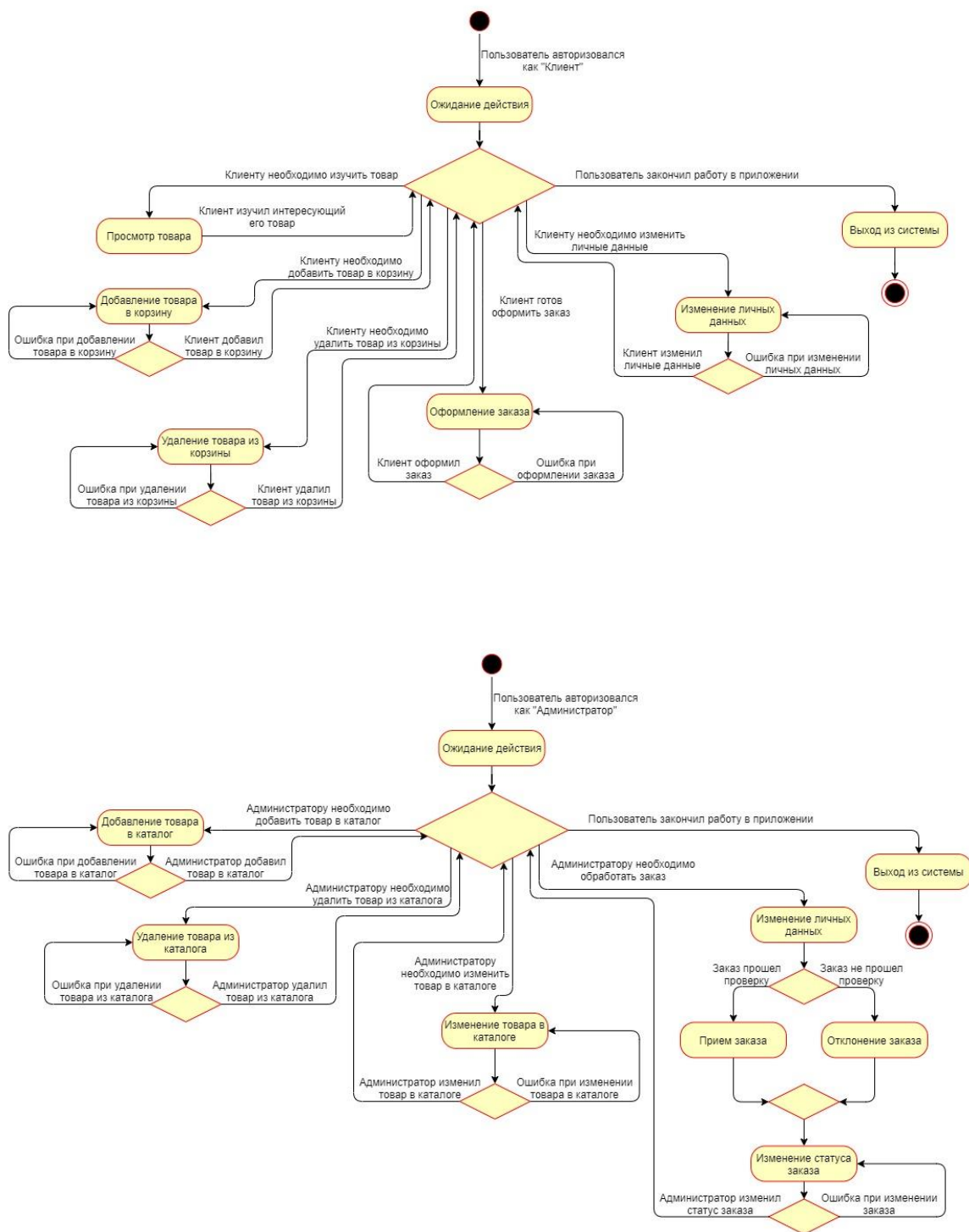


Рисунок 9. Диаграмма активности.

Диаграмма активности, изображенная на рисунке 9, отражает возможные действия, состояния которых описаны на диаграмме состояния (Рисунок 8).

2.2.5 Модель базы данных

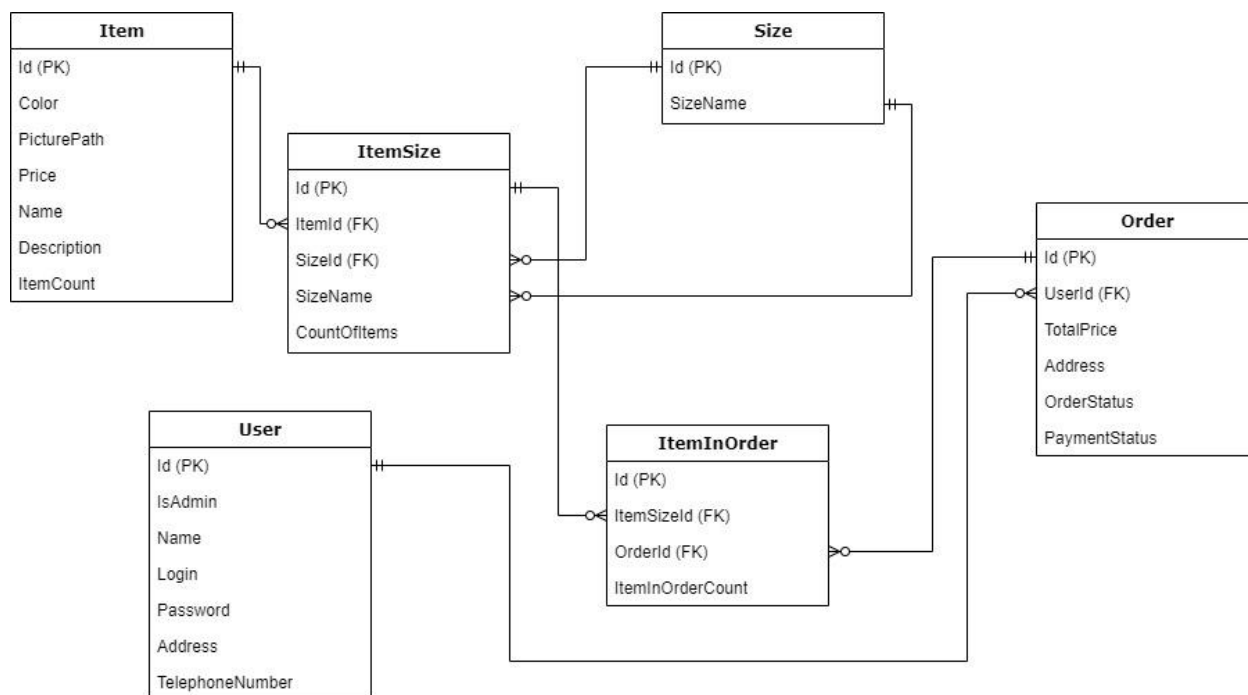


Рисунок 10. Схема базы данных.

На рисунке 10 представлена схема базы данных для разрабатываемого приложения.

2.2.6 Развертывание приложения

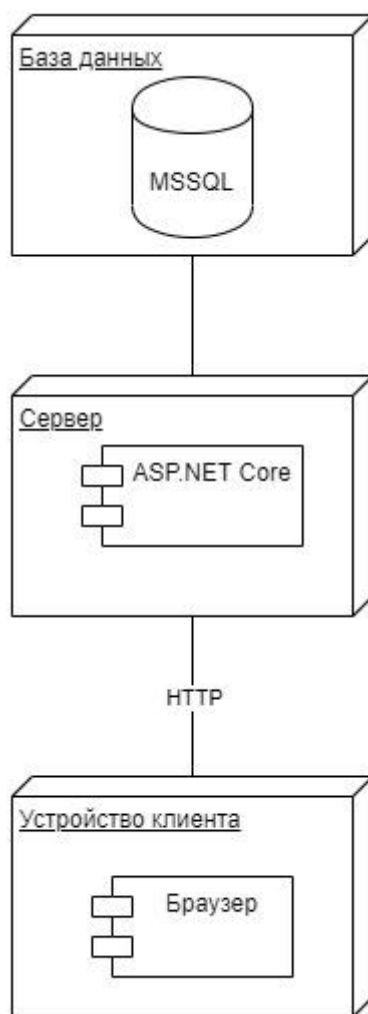


Рисунок 11. Диаграмма развертывания.

На Рисунке 11 представлена диаграмма развертывания, чтобы определить какие аппаратные компоненты (узлы) существуют, какие программные компоненты (артефакты) работают на каждом узле и как различные части этого комплекса соединяются друг с другом.

3. Реализация

3.1 Frontend

При создании клиентской части использовались фреймворк Angular и библиотека Bootstrap.

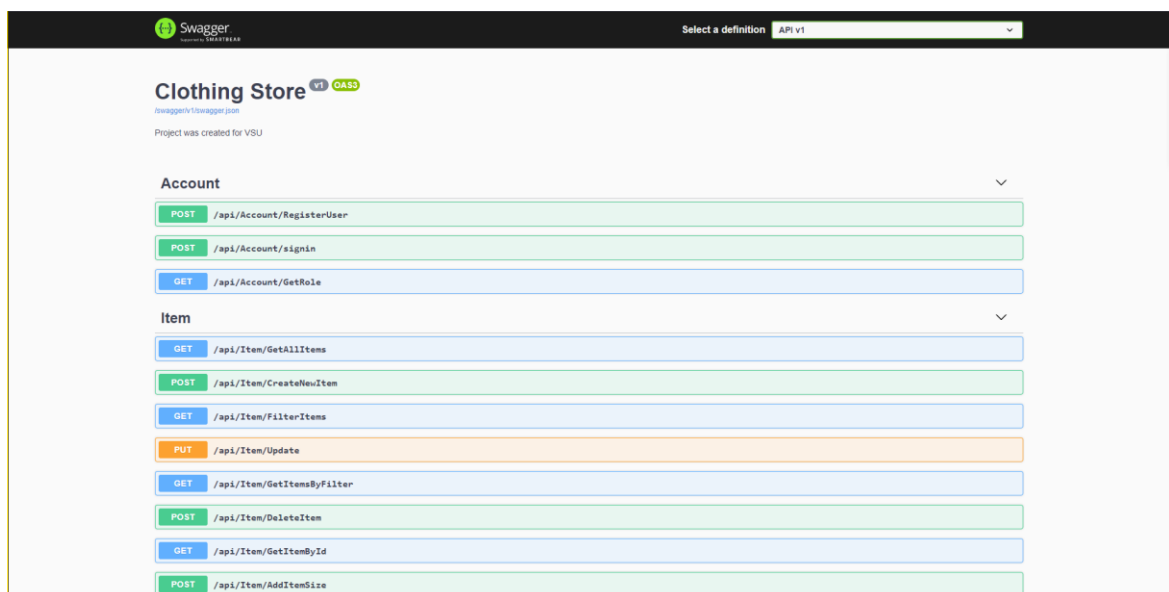
3.2 Backend

При создании серверной части использовался фреймворк ASP.NET Core 3.0. Для работы программы была выбрана ORM Entity framework core.

Документирование бэкенда приложения ведётся с помощью ПО с открытым исходным кодом Swagger. Этот инструмент позволяет разработчикам вест и предоставлять документацию REST веб-сервисов с возможностью тестирования запросов к методам API.

Структуру API Swagger понимает благодаря специальным аннотациям перед методом, например, [SwaggerResponse(HttpStatusCode.BadRequest)] или с помощью разбора xml комментариев, вот комментарий-аналог аннотации выше: `/// <response code="400">Некорректные значения</response>`.

Также ПО Swagger позволяет вызывать методы, требующие авторизации пользователя. Для этого предусмотрена специальная настройка. Достаточно в специальное поле внести значение действительного токена.



4. Интерфейс

4.1 Интерфейс неавторизованного пользователя

При открытии приложения пользователь попадает на главную страницу сайта (Рисунки 12.1 и 12.2). На ней пользователь имеет возможность перейти на страницу авторизации или регистрации. Также он может воспользоваться поисковой строкой и формой с фильтрами для поиска интересующего его товара.

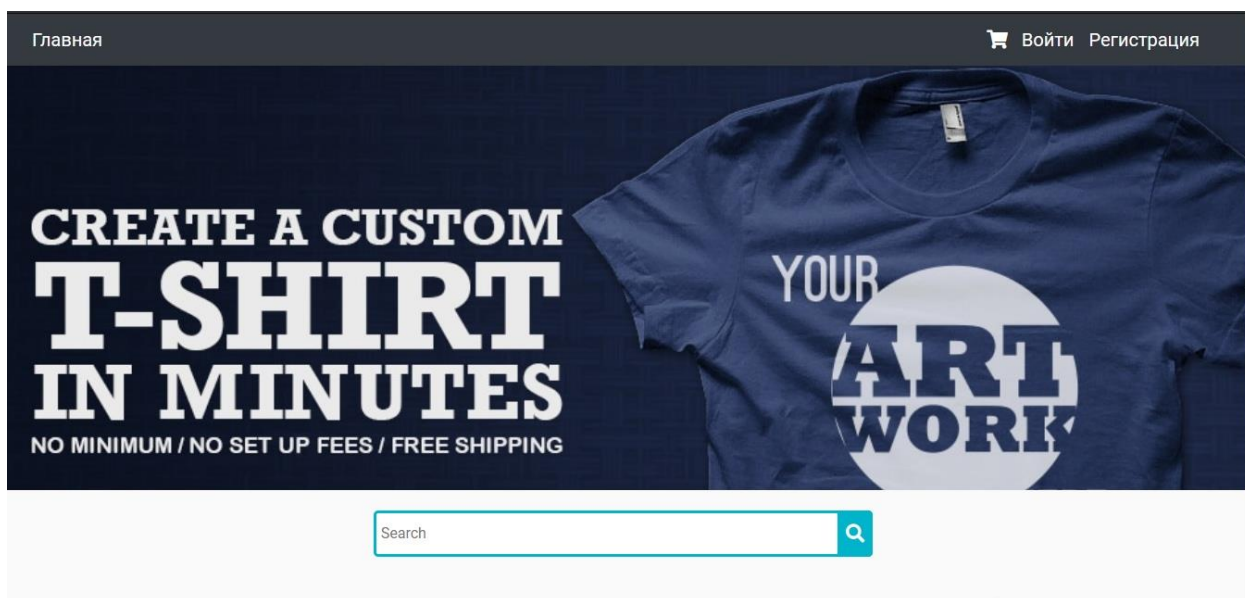


Рисунок 12.1.

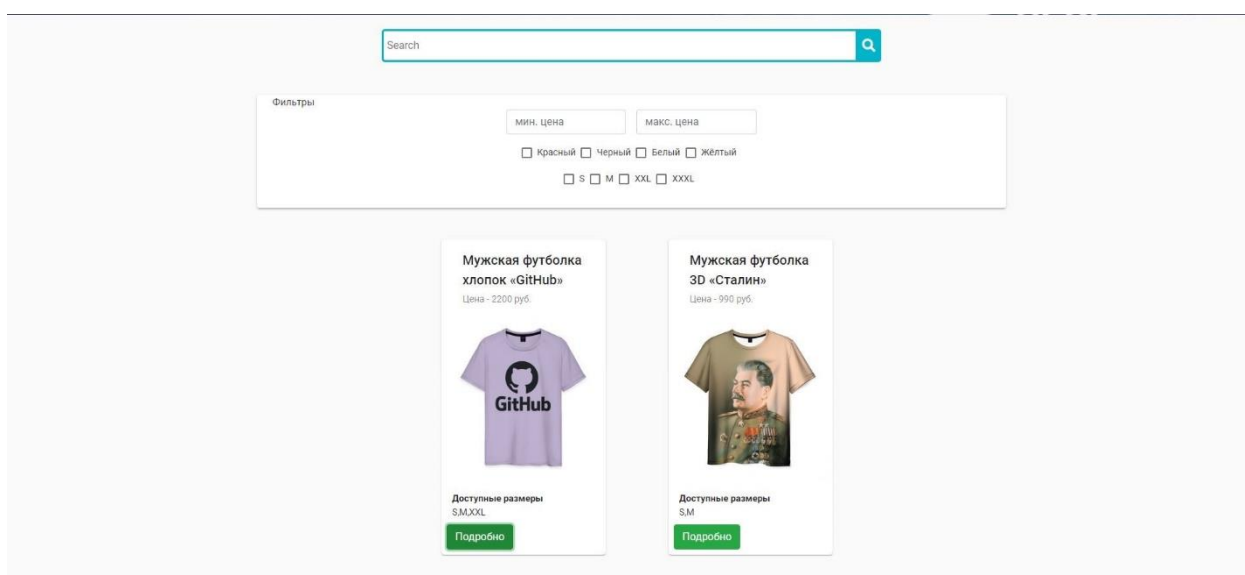


Рисунок 12.2.

Если ему пользователю понравился какой-то товар, то он может изучить его подробнее. Для этого ему нужно нажать на кнопку «Подробнее» под соответствующим товаром. При нажатии этой кнопки откроется страница с выбранным товаром (Рисунок 13).

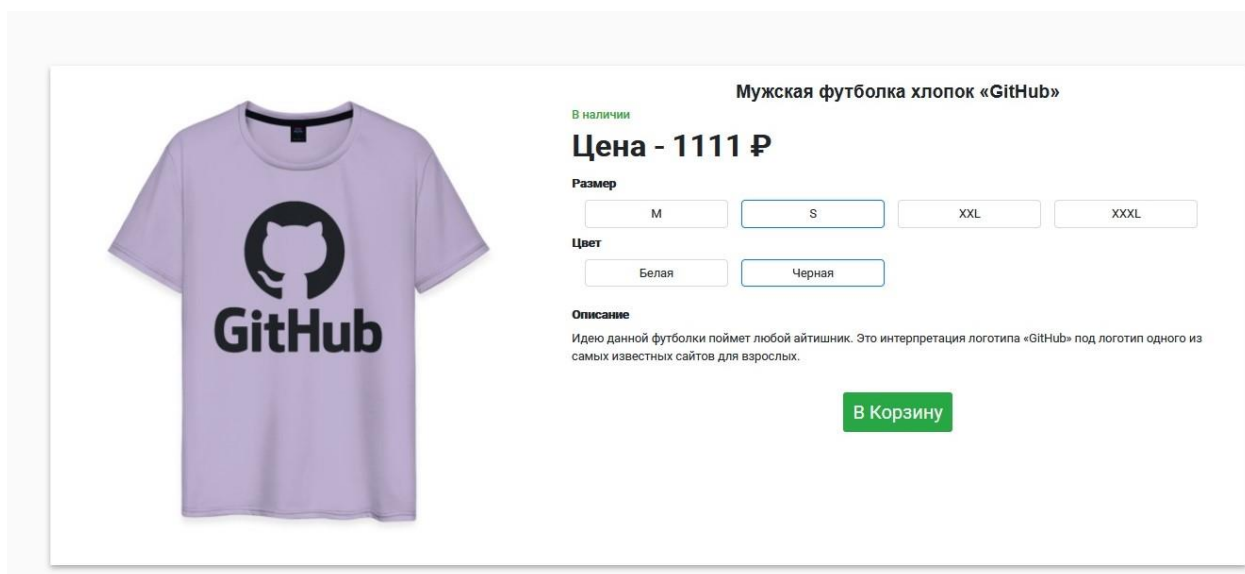


Рисунок 13.

Далее если пользователь захочет купить товар, то он должен добавить товар в корзину, нажав кнопку «В Корзину» и перейти в корзину нажав на значок корзины в шапке сайта. Перейдя на страницу корзины (Рисунок 14) пользователь увидит все добавленные товары. Для оформления заказа пользователю необходимо авторизоваться.

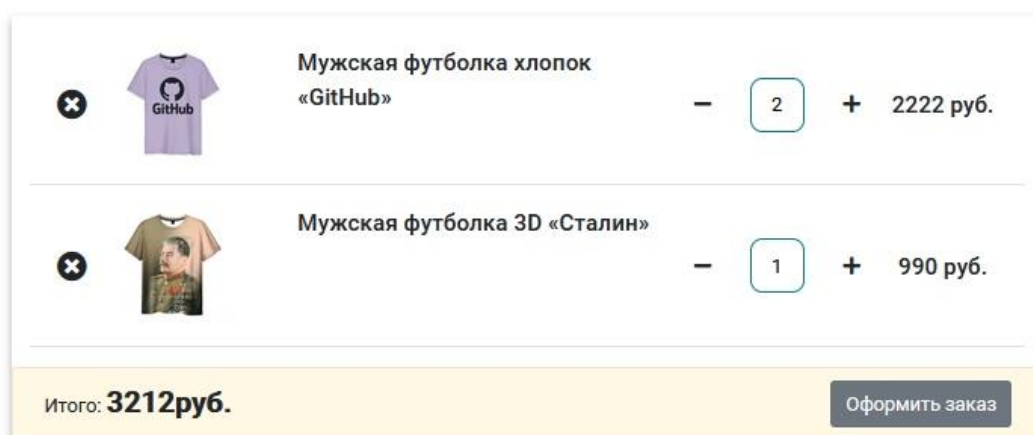


Рисунок 14.

Если пользователь решил авторизоваться и нажал на кнопку «Войти», то он попадает на страницу авторизации (Рисунок 15). Для того чтобы авторизоваться ему необходимо ввести логин, пароль в соответствующие поля и нажать кнопку «Войти».

Главная [Войти](#) [Регистрация](#)

вход

Email

Email

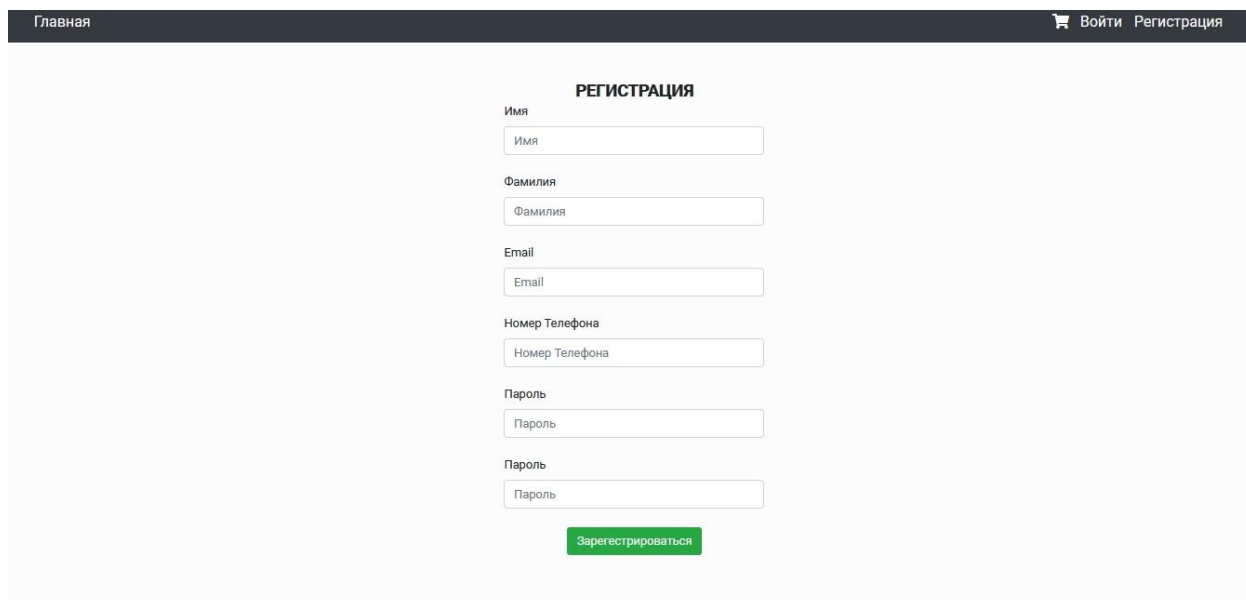
Пароль

Пароль

Войти

Рисунок 15.

Если по какой-то причине пользователь еще не зарегистрирован, то он имеет возможность нажать на кнопку «Регистрация» для перехода на страницу регистрации (Рисунок 16) и дальнейшей регистрации. Для того, чтобы зарегистрироваться пользователю необходимо заполнить поля «Имя», «Фамилия», «Email», «Номер Телефона», «Пароль» и нажать кнопку «Зарегистрироваться».



The image shows a registration form titled "РЕГИСТРАЦИЯ" (Registration). At the top, there is a dark navigation bar with the text "Главная" (Home) on the left and "Войти" (Login) and "Регистрация" (Registration) on the right. The form itself is centered and contains the following fields: "Имя" (Name), "Фамилия" (Surname), "Email", "Номер Телефона" (Phone Number), and two "Пароль" (Password) fields. Each field is represented by a text input box with its label above it. Below the password fields is a green button labeled "Зарегистрироваться" (Register).

Рисунок 16.

При успешной регистрации выводится сообщение «Регистрация прошла успешно!» (Рисунок 17).

Регистрация прошла успешно ! ✓

Войти ➔

Рисунок 17.

4.2 Интерфейс пользователя, авторизованного под ролью «Клиент»

Если пользователь авторизовался под ролью «Клиент», то он попадает на главную страницу (Рисунок 18).

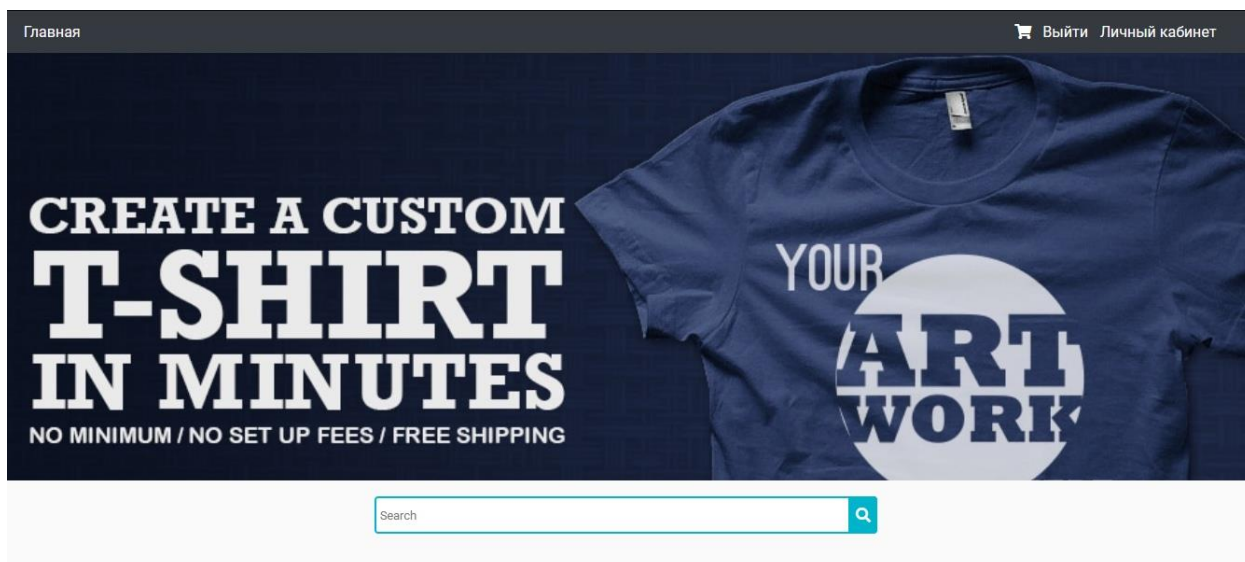


Рисунок 18.

На этой странице клиент может перейти на страницу с его личными данными, нажав кнопку «Личный кабинет» в шапке страницы. После нажатия на эту кнопку пользователь переходит на страницу личного кабинета (Рисунок 19). На этой странице клиент может просмотреть свои личные данные и при необходимости их отредактировать, нажав на кнопку «Редактировать».

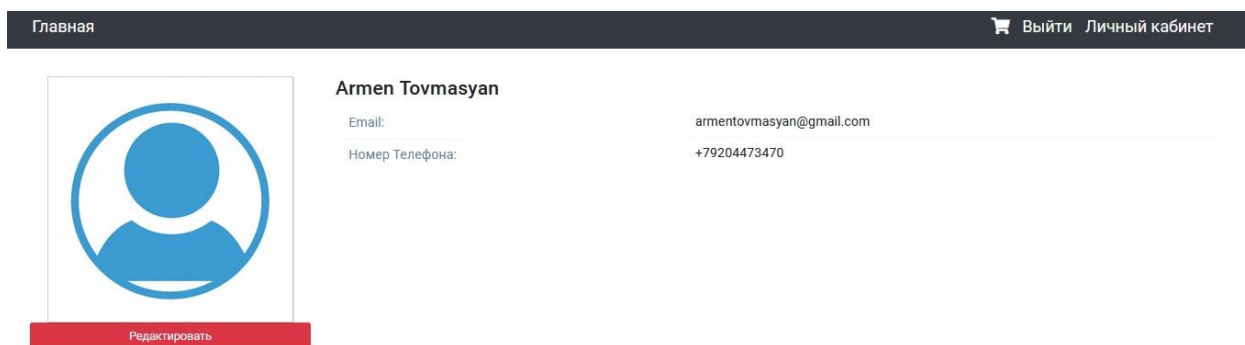


Рисунок 19.

Также на главной странице клиент может найти, ознакомиться и добавить в корзину интересующие его товары выше указанным способом. Перейдя в корзину и убедившись в правильности выбранных товаров, клиент

может перейти на страницу оформления заказа нажав на кнопку «Оформить заказ». После этого пользователь попадает на страницу оформления заказа (Рисунок 20).

The screenshot displays a web form for order completion. At the top, a dark navigation bar contains the link 'Главная' (Home) on the left and 'Выйти' (Logout) and 'Личный кабинет' (My Account) on the right. The main form area is divided into several sections. The 'ДОСТАВКА' (DELIVERY) section has two tabs: 'Почта' (Post) and 'Курьер' (Courier). Below these are input fields for 'Город*' (City) and '№ отделения' (Post office number), each with a placeholder text. A note below these fields states: 'Укажите город и адрес доставки и курьер доставит Ваш заказ прямо в руки. Курьер, по телефону предварительно согласует удобное для Вас время доставки.' To the right of the delivery section is a box titled 'Ваш заказ' (Your order) containing a t-shirt icon, the text 'Мужская футболка хлопок «GitHub»', 'Размер - XXL', 'Количество - 2', 'Общая Цена - 2222', a 'Редактировать заказ' (Edit order) link, and a bold 'Итого 2222 руб.' (Total 2222 rub.). Below the delivery section is the 'ОПЛАТА' (PAYMENT) section with tabs for 'Наличными' (Cash) and 'Онлайн' (Online). At the bottom is the 'ПРЕМЕЧАНИЕ К ЗАКАЗУ' (REMARKS TO ORDER) section with a large text area and a placeholder 'Укажите здесь Ваши пожелания'. A red 'Отправить' (Send) button is located at the bottom center of the form.

Рисунок 20.

Чтобы оформить и отправить заказ необходимо выбрать способ доставки, указать город и номер почтового отделения в соответствующих полях, выбрать способ оплаты, при желании написать примечание соответствующем поле и нажать кнопку «Отправить».

После окончания своей деятельности на сайте клиент может выйти из аккаунта, нажав на кнопку «Выйти» в шапке страницы.

4.3 Интерфейс пользователя, авторизованного под ролью «Администратор»

Если пользователь авторизовался под ролью «Администратор», то он попадает на главную страницу (Рисунок 21).

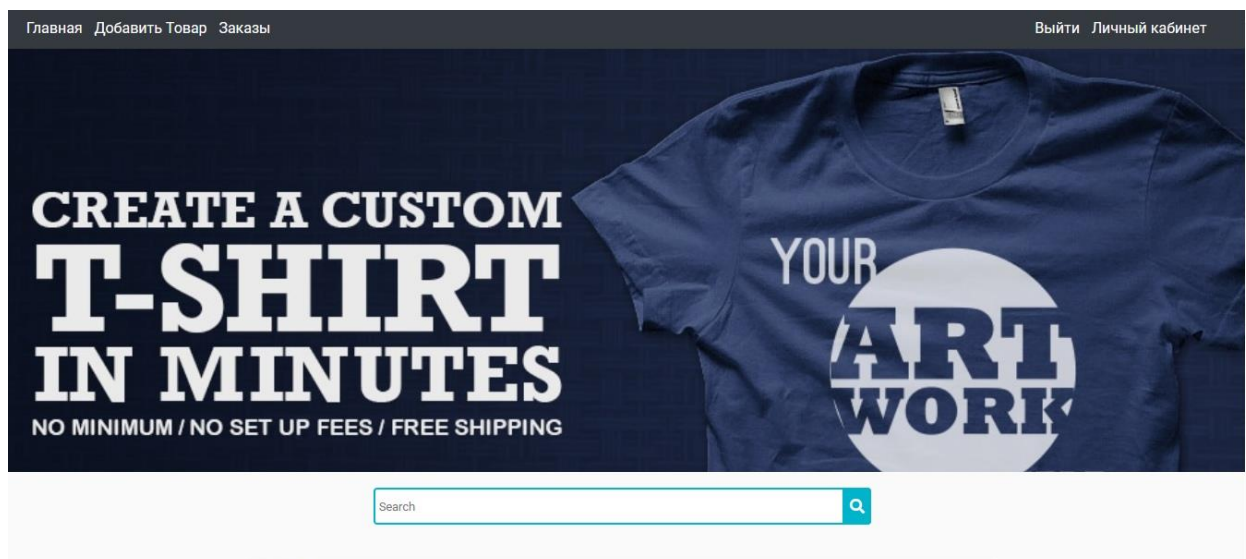


Рисунок 21.

Администратор может посмотреть свои личные данные также, как и пользователь авторизованный под ролью «Клиент».

Администратор имеет возможность добавлять товары в каталог. Для этого ему необходимо перейти на страницу добавления товара (Рисунок 22), нажав кнопку «Добавить товар» в шапке страницы.

Рисунок 22.

Для того, чтобы добавить товар в каталог необходимо заполнить поля «Название», «Image url», «Цена(РУБ.)», «Описание», выбрать доступные цвета, выбрать доступные размеры и нажать на кнопку «Создать».

Также администратор может редактировать и удалять товары в каталоге. Для этого ему необходимо на главной странице найти интересующий его товар, перейти на страницу этого товара. Для того, чтобы удалить товар необходимо нажать на значок мусорной урны, а для того чтобы перейти на страницу изменения товара (Рисунок 23) необходимо нажать на значок в форме пера с листком.

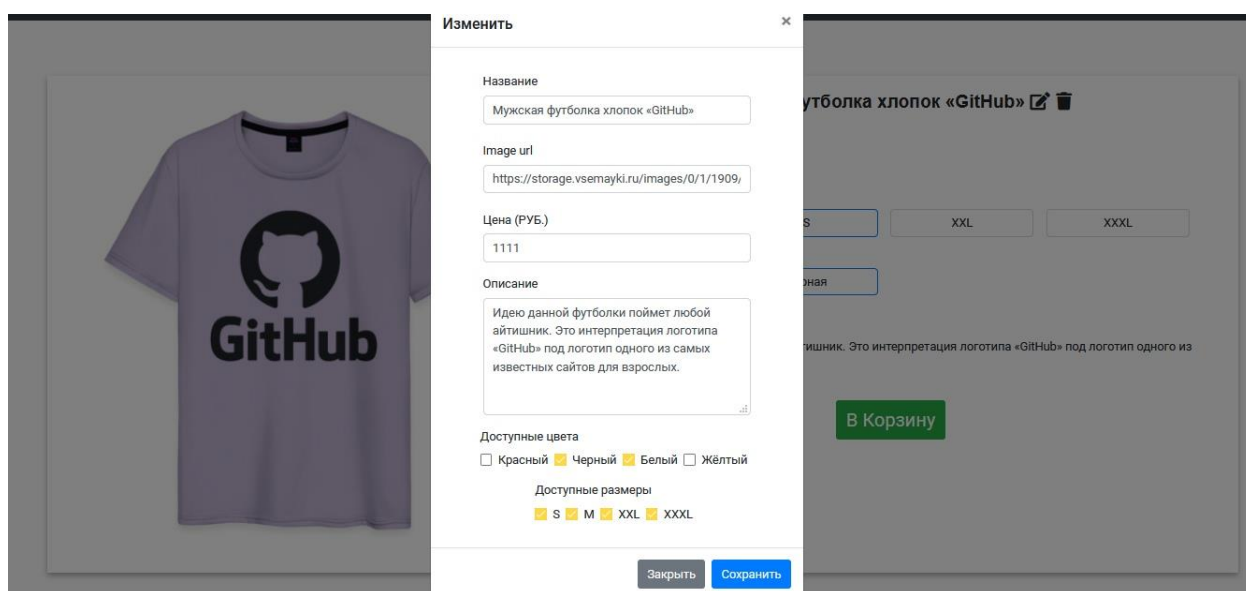


Рисунок 23.

Для того чтобы изменить товар необходимо изменить нужные поля и нажать кнопку «Сохранить».

После окончания своей деятельности на сайте администратор может выйти из аккаунта, нажав на кнопку «Выйти» в шапке страницы.

5. Тестирование

Для тестирования приложения на стороне backend были использованы юнит-тесты.

Фреймворк для тестирования – NUnit.net. Это открытая среда юнит-тестирования приложений для .NET.

В том же решении, где располагается проект backend, был создан дополнительный проект для тестов – Integration.Tests.

В этом проекте были реализованы классы тестирования для каждого контроллера, которые существуют в проекте backend.

Для каждого метода, которые используются со стороны frontend, были реализованы методы тестирования. Им на вход подавались различные данные, при этом ожидался определенный ответ.

Юнит-тесты позволили нам сократить время для проверки работоспособности приложения. Если мы что-то меняли в уже созданном методе, мы просто могли запустить юнит-тест этого метода, и увидеть при каких данных появляется ошибка, если она вообще появляется.

Заключение

В результате работы было реализовано веб-приложение, выполняющее функции интернет-магазина по продаже футболок.

В системе предусмотрены три уровня доступа:

1. Неавторизованный пользователь;
2. Клиент;
3. Администратор.

Неавторизованный пользователь обладает следующими возможностями:

- Регистрация;
- Авторизация;
- Просмотр товаров из каталога;
- Добавление товаров в корзину;
- Удаление товаров из корзины.

Клиент обладает следующими возможностями:

- Просмотр товаров из каталога;
- Добавление товаров в корзину;
- Удаление товаров из корзины;
- Оформление заказа;
- Изменение личных данных;
- Выход из системы.

Администратор обладает следующими возможностями:

- Добавление товаров в каталог;
- Изменение товаров в каталоге;
- Удаление товаров из каталога;
- Обработка заказа:

1. Прием/Отклонение заказа;
 2. Изменение статуса заказа.
- Изменение личных данных;
 - Выход из системы.

Отчет по ролям

И.Е. Лесных:

1. Написать главу «Введение» в курсовом проекте;
2. Создание диаграммы классов;
3. Создание диаграммы последовательностей;
4. Создание диаграммы состояний;
5. Написать главу «Интерфейс» в курсовом проекте;
6. Написать главу «Заключение» в курсовом проекте;
7. Создание главной страницы для клиента;
8. Создание главной страницы для администратора;
9. Создание страницы корзины;
10. Создание контроллеров для работы с пользователем;
11. Создание контроллеров для работы с заказами;
12. Создание отчета по обязанностям;
13. Создание презентации.

Г.О. Латынин:

1. Написать ТЗ;
2. Написать главу «Анализ существующих решений»;
3. Создание диаграммы объектов;
4. Создание диаграммы активности;
5. Написать главу «Тестирование» в курсовом проекте;
6. Создание страницы личного кабинета;
7. Создание страницы для добавления товара в каталог;
8. Создание страницы для изменения товара в каталоге;
9. Создание страницы для оформления заказа;
10. Создание контроллеров для работы с товарами;
11. Создание интеграционных тестов для всех методов контроллера;
12. Подключение SWAGGER к серверной части;
13. Deploy серверной части на хостинг Reg.ru;

14. Подключение Яндекс.Метрика.

А.А. Товмасын:

1. Написать главу «Постановка задачи» в курсовом проекте;
2. Создание диаграммы прецедентов;
3. Создание диаграммы взаимодействий;
4. Создание диаграммы развертывания;
5. Написать главу «Реализация» в курсовом проекте;
6. Создание страницы регистрации;
7. Создание страницы авторизации;
8. Создание страницы товара для клиента;
9. Создание страницы товара для администратора;
10. Создание страницы для управления заказами;
11. Создание контроллеров для работы с размерами;
12. Настройка CI/CD процесса;
13. Deploy клиентской части на хостинг Heroku;
14. Запись видео.