

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

Факультет компьютерных наук

Кафедра информационных технологий управления

Курсовая работа по дисциплине «Технологии программирования»
Продовольственный онлайн-супермаркет с доставкой на дом «Diego»

Направление 09.03.02 Информационные системы и технологии

Преподаватель _____ В.С. Тарасов, ст. преподаватель _____. 20__

Обучающийся _____ Д.Ю. Бесхмельницына, 3 курс, д/о

Обучающийся _____ Е.А. Киринов, 3 курс, д/о

Воронеж 2023

Содержание

Содержание	2
Введение	5
1 Постановка задачи	6
1.1 Требования к разрабатываемой системе	6
1.1.1 Функциональные требования	6
1.1.2 Нефункциональные требования	7
2 Анализ предметной области	8
2.1 Терминология (гlossарий) предметной области	8
2.2 Обзор аналогов	8
2.2.1 METRO	9
2.2.2 Перекрёсток	11
2.2.3 Лента	13
3 Диаграммы	17
3.1 Диаграмма прецедентов (Use-case diagram)	17
3.2 Диаграмма классов (Class diagram)	18
3.3 Диаграмма деятельности (Activity diagram)	19
3.4 Диаграмма последовательности (Sequence diagram)	21
3.5 Диаграмма развёртывания (Deployment diagram)	22
3.6 Диаграмма сотрудничества (Collaboration diagram)	23
3.7 Диаграмма объектов (Object diagram)	28
3.8 Диаграмма состояний (Statechart diagram)	29
3.9 ER–диаграмма	31
3.10 IDEF0	31

4 Реализация.....	34
4.1 Средства реализации.....	34
4.2 Разработка frontend	35
4.3 Разработка backend.....	37
4.4 Основная функциональность сайта.....	39
4.5 Для пользователя.....	39
4.5.1 Оформление заказа	39
4.5.2 Калькулятор калорий.....	45
4.5.3 Возможности сайта для вегетарианцев	46
4.5.4 Аккаунт пользователя	46
4.6 Для администратора.....	48
4.6.1 Добавление товаров	48
4.6.2 Редактирование и удаление товаров	49
4.6.3 Изменение ролей пользователей и их удаление	50
4.6.4 Управление заказами	51
4.7 Для курьера.....	52
4.7.1 Доставка заказов	52
5 Тестирование	54
5.1 UI-тестирование	54
5.2 Дымовое тестирование	57
5.3 Usability-тестирование.....	60
5.4 Unit-тесты.....	64
6 Анализ веб-приложения	66
Заключение	69

Список использованной литературы.....	70
---------------------------------------	----

Введение

Все большее влияние информационные технологии оказывают на жизнь людей. Например, популярность набирает доставка продуктов питания на дом. Если раньше для приобретения продуктов человек должен был пойти в специально отведенное для этого место, то сейчас не выходя из дома можно сделать заказ в онлайн-супермаркете и в ближайшее время получить его с доставкой на указанный адрес.

Сегодняшний быстрый темп жизни, ограниченное количество времени и изменение покупательских привычек привели к росту популярности онлайн-супермаркетов с доставкой на дом. Онлайн-покупки продовольственных товаров стали эффективным способом удовлетворения потребностей, минуя необходимость физического посещения традиционных магазинов.

Для совершения такого рода покупок нужна лишь цифровая техника. Кроме этого у пользователя появится информация о продукте, его составе, пищевой ценности. Пользователь сможет просмотреть ассортимент онлайн-супермаркета в короткие сроки, а с помощью строки поиска найдет интересующий товар по названию. А после оформления заказа пользователь сможет всегда знать, на каком этапе находится заказ, отслеживая его статус.

Данный проект «Продовольственный онлайн-супермаркет с доставкой на дом» повышает качество жизни общества и позволяет людям совершать покупки.

Целью проекта является создание сайта, который позволит пользователям совершать заказы товаров с помощью нескольких кликов. Поэтому продовольственный онлайн-супермаркет с доставкой на дом является важным и актуальным направлением.

1 Постановка задачи

Целью данного курсового проекта является создание сайта, предназначенного для оформления продовольственных заказов с их доставкой пользователям по указанным адресам.

Сайт решает следующий ряд задач:

- Оформление заказов с доставкой по указанному адресу;
- Расчет числа упаковок товара на основании введенного количества калорий;
- Реализация продажи товаров для конкретной группы людей с системой питания, исключающей определенные типы пищи.

1.1 Требования к разрабатываемой системе

1.1.1 Функциональные требования

К разрабатываемому сайту выдвигаются следующие функциональные требования:

- Оформление заказов;
- Расчет числа упаковок товара на основании введенного количества калорий;
- Просмотр товаров питания, пригодных для вегетарианцев;
- Просмотр товаров питания посредством расширенной фильтрации;
- Получение информации о наличии определенных пищевых ценностей в различных товарах;
- Просмотр категорий и товаров категорий;
- Поиск товаров по названию в поисковой строке;
- Регистрация, авторизация в системе;
- Добавление товаров в корзину;
- Очистка корзины с товарами;
- Осуществление редактирования данных своего аккаунта после входа в авторизованную зону;
- Выход из аккаунта и его удаление;

- Просмотр списка сделанных заказов и товаров, содержащихся в каждом заказе;
- Добавление, редактирование товаров и их удаление администратором;
- Удаление пользователей администратором;
- Изменение ролей пользователей администратором;
- Выполнение доставки заказа по адресу курьером.

1.1.2 Нефункциональные требования

К разрабатываемому сайту выдвигаются следующие технические требования:

- Регистрация и авторизация должна реализовываться с использованием данных пользователя;
- Данные пользователей после регистрации должны храниться и идентифицироваться системой авторизации;
- Удаление пользователей происходит с удалением его учетной записи.

2 Анализ предметной области

2.1 Терминология (гlossарий) предметной области

Пользователь — неавторизованный пользователь сайта, не имеющий полного доступа к функциям сайта.

Курьер — авторизованный пользователь сайта, занимающийся доставкой заказов пользователям.

Авторизованный пользователь (клиент) – авторизованный пользователь сайта, имеющий полный доступ к функциям сайта.

Администратор — авторизованный пользователь сайта, занимающийся управлением сайта, добавлением, удалением, редактированием товаров, изменением ролей пользователей и их удалением, записью заказов курьерам.

Профиль (аккаунт) — веб-страница пользователя сайта с персональными данными о нем.

Сайт (веб-приложение) — интернет-ресурс, состоящий из одной, нескольких или множества виртуальных страниц. Клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера [1].

Backend (серверная часть) – программно-аппаратная часть сервиса, которая хранится на сервере, обрабатывает полученные данные и отправляет ответ обратно.

Frontend (клиентская сторона) – интерфейс с набором функций, с которым взаимодействует пользователь.

2.2 Обзор аналогов

Прежде, чем начинать разработку сайта онлайн-супермаркета с доставкой на дом, необходимо проанализировать уже имеющиеся платформы для понимания их плюсов и минусов. После сбора информации можно переходить к этапу разработки с учетом сделанных выводов по изученным аналогам.

2.2.1 METRO

METRO – интернет-магазин продуктов с доставкой на дом. Сайт магазина позволяет пользователям приобрести товары, добавив их в корзину и оформив заказ.

После входа на сайт пользователь видит главный экран с блоками товаров, акций и предложений. Категории располагаются в отдельном раскрывающемся табло при нажатии на его оглавление «Каталог». Пользователь может просматривать информацию о товарах, добавлять их в корзину, переходить в корзину, но при условии, что корзина не пустая. В противном случае страница корзины не откроется. Корзина открывается не сразу после нажатия кнопки корзины. Появляется лишь всплывающее окно, на котором можно нажать кнопку «Войти и оформить». Только после авторизации или регистрации корзина и последующее оформление заказа при желании будут доступны.

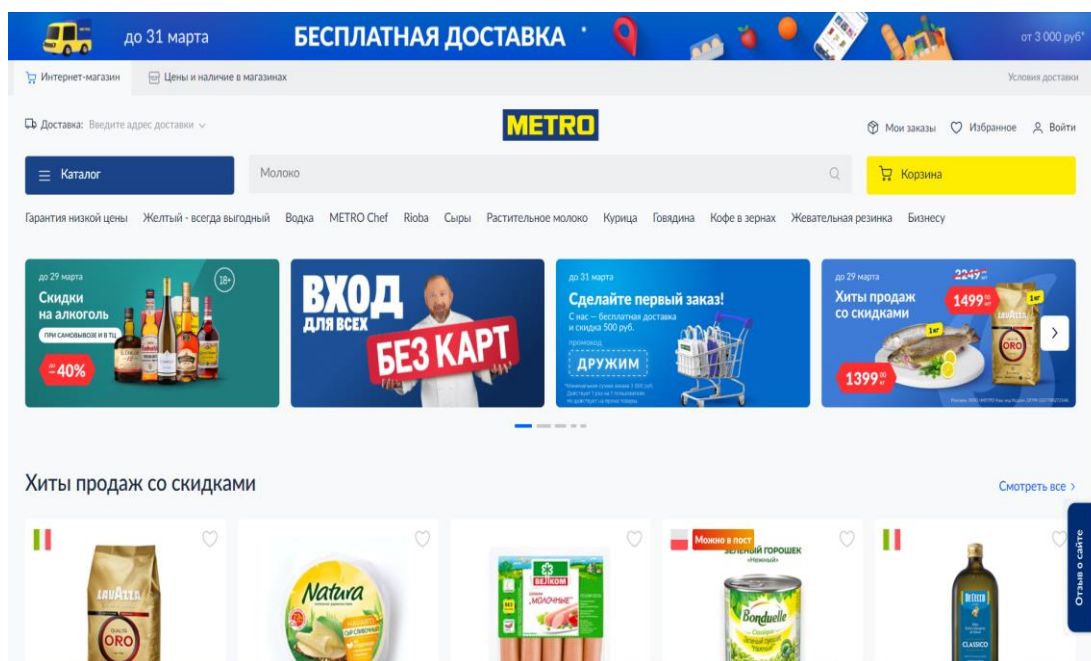


Рисунок 1 - Главная страница сайта «METRO»

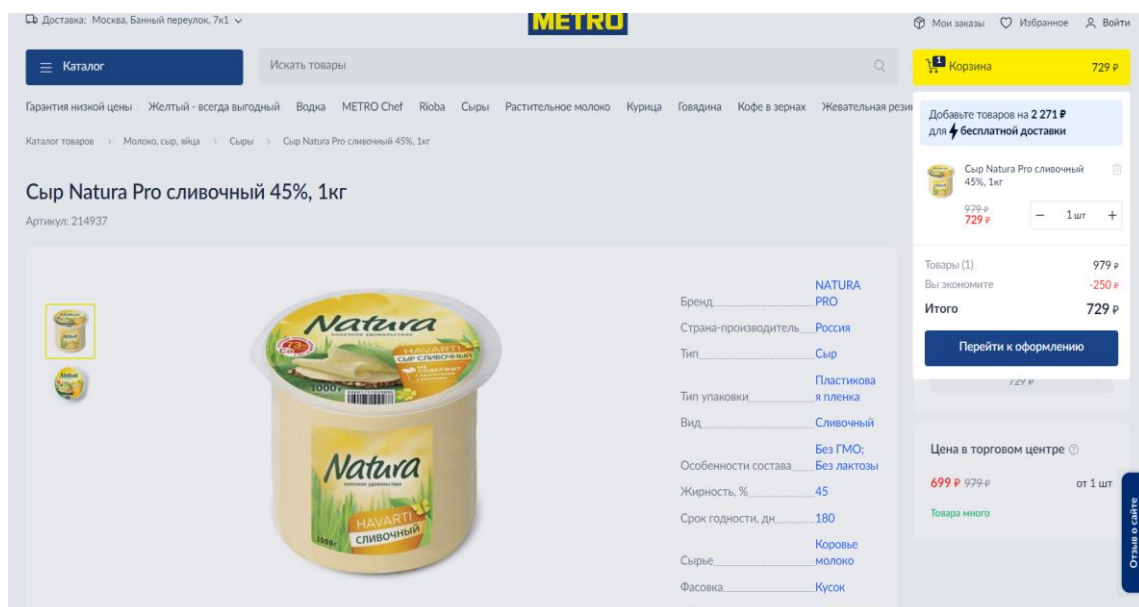


Рисунок 2 - Страница товара при нажатии кнопки «В корзину» сайта «METRO»

После входа в авторизованную зону пользователь может продолжить оформление заказа и ввести необходимые данные для доставки и оплаты. После этого заказ будет оформлен и доставлен.

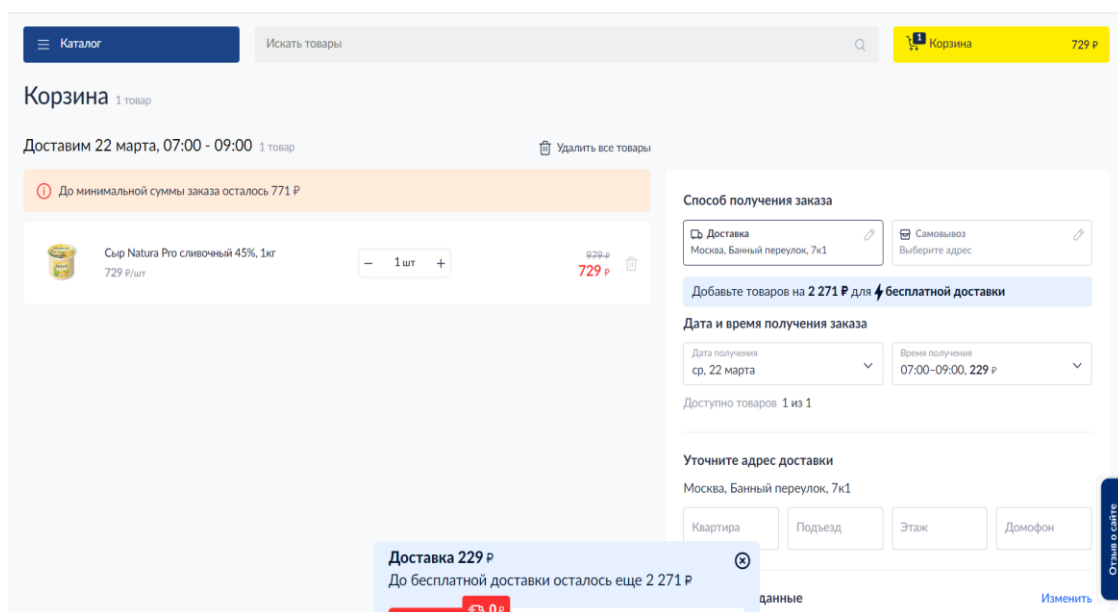


Рисунок 3 - Страница корзины сайта «METRO»

Для просмотра аккаунта можно нажать кнопку в шапке сайта «Аккаунт». Отобразится страница с данными аккаунта, с возможностью редактировать профиль [2].

Преимущества сайта:

- Сайт позволяет просматривать и сравнивать цены интернет-магазина и сети магазинов, расположенных на территории города;
- Доступность и полнота информации о товарах, акциях, предложениях;
- Сайт позволяет оставить отзыв о сайте в отдельном окне;
- Наличие бесплатной доставки при достижении определенной суммы.

Недостатки сайта:

- Сложный процесс оформления заказа с множеством промежуточных действий;
- Заказ продуктов с доставкой возможно оформить лишь при достижении определенной суммы;
- Некоторые товары, отображающиеся на сайте, невозможно заказать.

2.2.2 Перекрёсток

Перекрёсток – интернет-магазин продуктов с доставкой на дом. Сайт магазина позволяет пользователям приобрести товары, добавив их в корзину и оформив заказ. Кроме этого сайт магазина имеет вкладку «Акции», где отображаются товары со скидкой.

После входа на сайт пользователь видит главный экран с категориями и блоками товаров, блоки акций и предложений. Также пользователь может просматривать информацию о товарах, добавлять их в корзину, переходить в корзину, но при желании оформить заказ необходимо зарегистрироваться или авторизоваться.

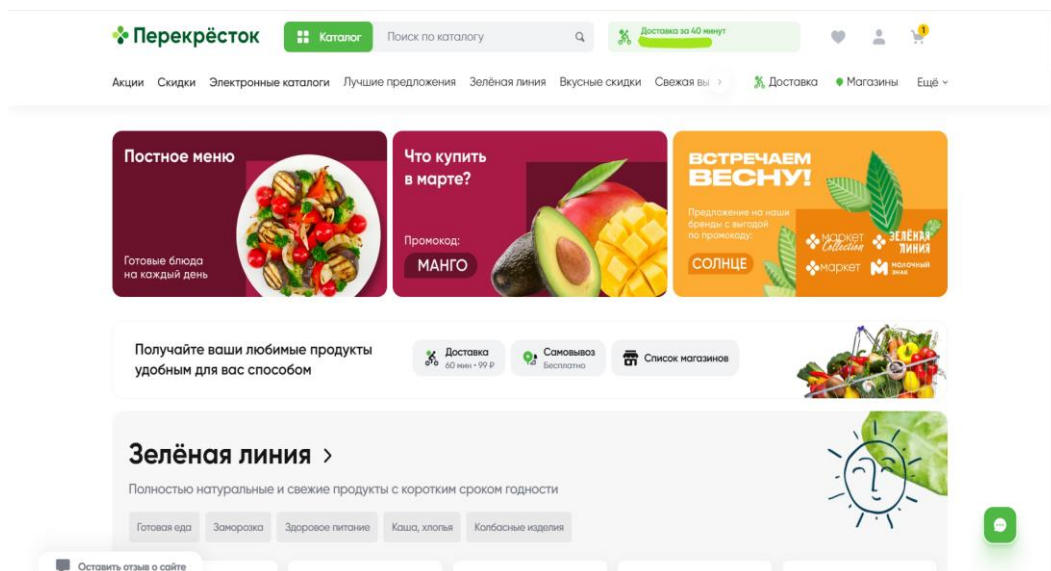


Рисунок 4 - Главная страница сайта «Перекрёсток»

После входа в авторизованную зону пользователь может продолжить оформление заказа и ввести адрес и время доставки, детали доставки и способ оплаты. Предоставив все необходимые данные, заказ будет оформлен и доставлен.

Для просмотра корзины можно нажать кнопку в шапке сайта с изображением корзины. Отобразится страница с наличием товаров или их отсутствием.

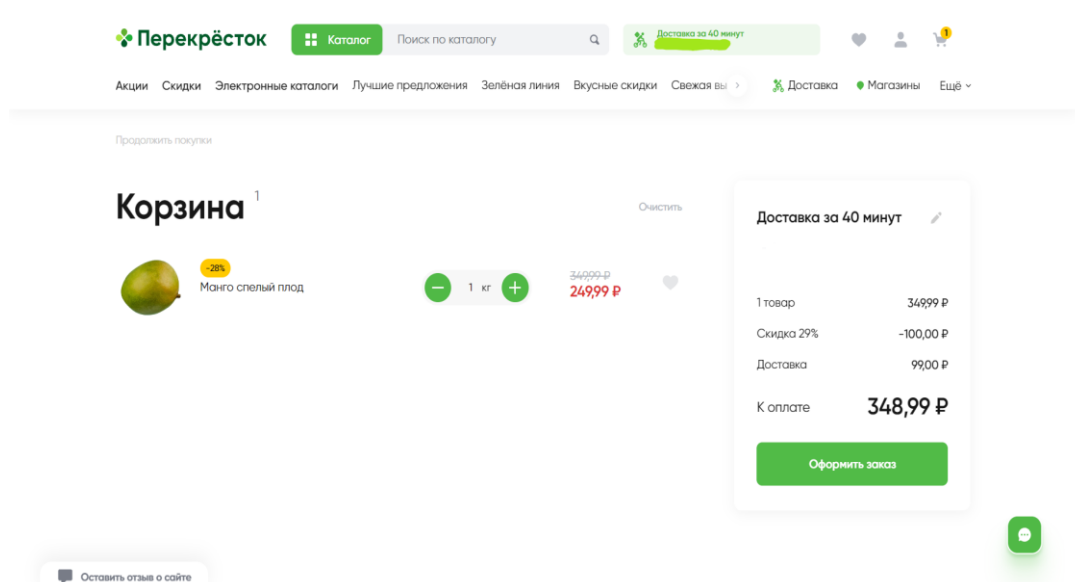


Рисунок 5 - Страница корзины сайта «Перекрёсток»

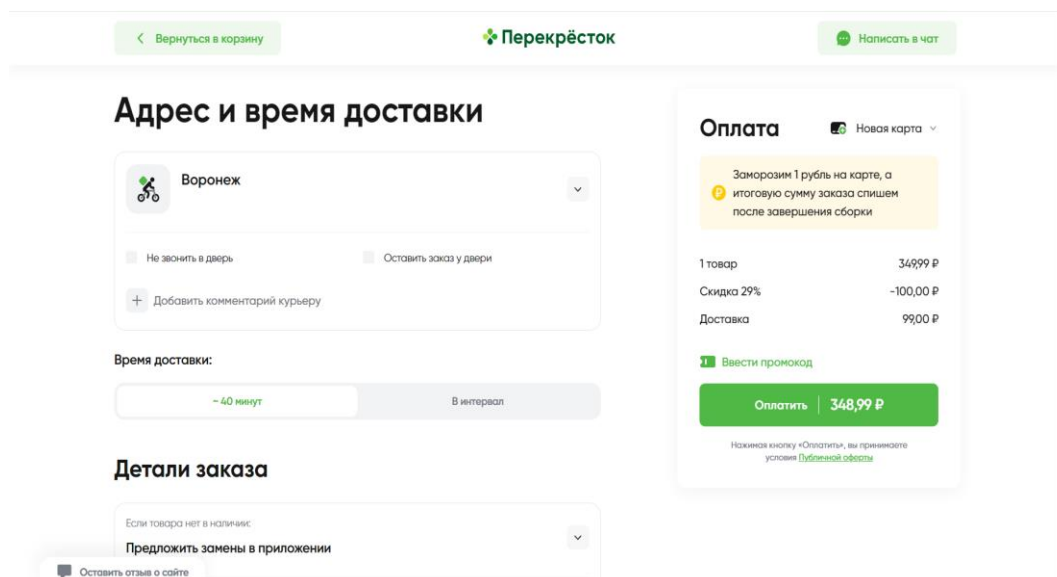


Рисунок 6 - Страница оформления заказа сайта «Перекрёсток»

Для просмотра аккаунта можно нажать кнопку в шапке сайта «Аккаунт». Отобразится страница со специализированной картой магазина и размером баллов на ней, а также отображаются оплаченные или находящиеся в процессе оформления заказы, специальные предложения, промокоды.

Преимущества сайта:

- Сайт содержит множество различных категорий товаров;
- Сайт имеет раздел акций на некоторые товары;
- Сайт предлагает промокоды со скидкой на некоторые товары;
- Сайт обладает чатом поддержки для помощи пользователям.

Недостатки сайта:

- Многочисленные предложения, категории, акции сайта сбивают пользователя от первоначальных потребностей;
- При наличии большого функционала потерявшимся пользователям необходимы сообщения с указаниями по навигации для осуществления основных функций сайта.

2.2.3 Лента

Лента – интернет-магазин продуктов с доставкой на дом. Сайт магазина позволяет пользователям приобрести товары, добавив их в корзину и оформив

заказ. Кроме этого сайт магазина имеет вкладку «Акции», где отображаются товары со скидкой.

После входа на сайт пользователь видит главный экран с категориями и блоками товаров, блоки акций и предложений. Также пользователь может выбирать категорию товаров, просматривать информацию о товарах, добавлять их в корзину, переходить в корзину, но при желании оформить заказ необходимо зарегистрироваться или авторизоваться.

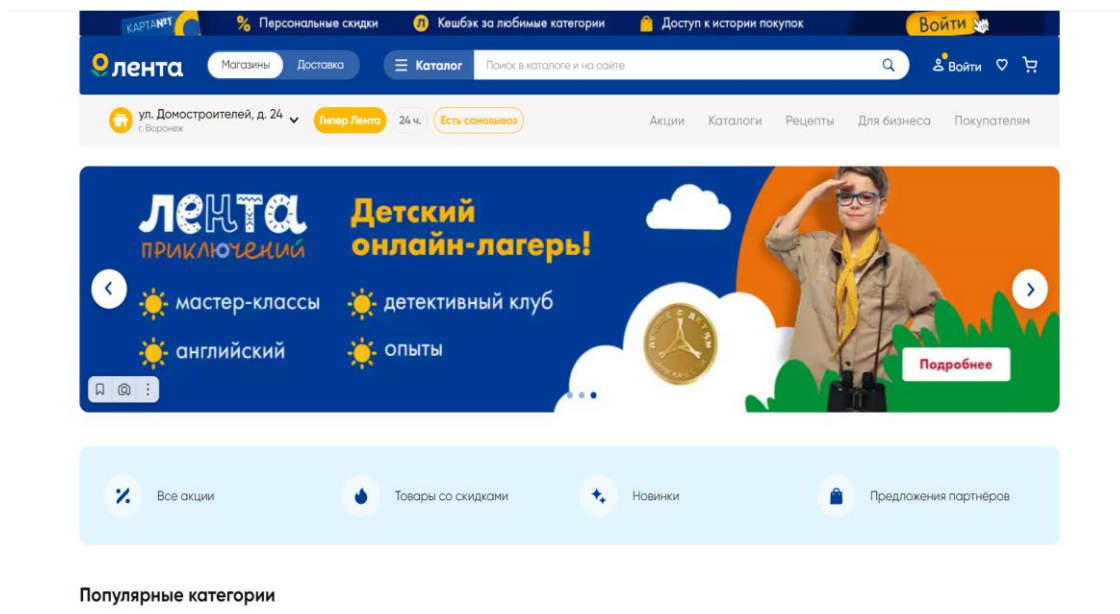


Рисунок 7 - Главная страница сайта «Лента»

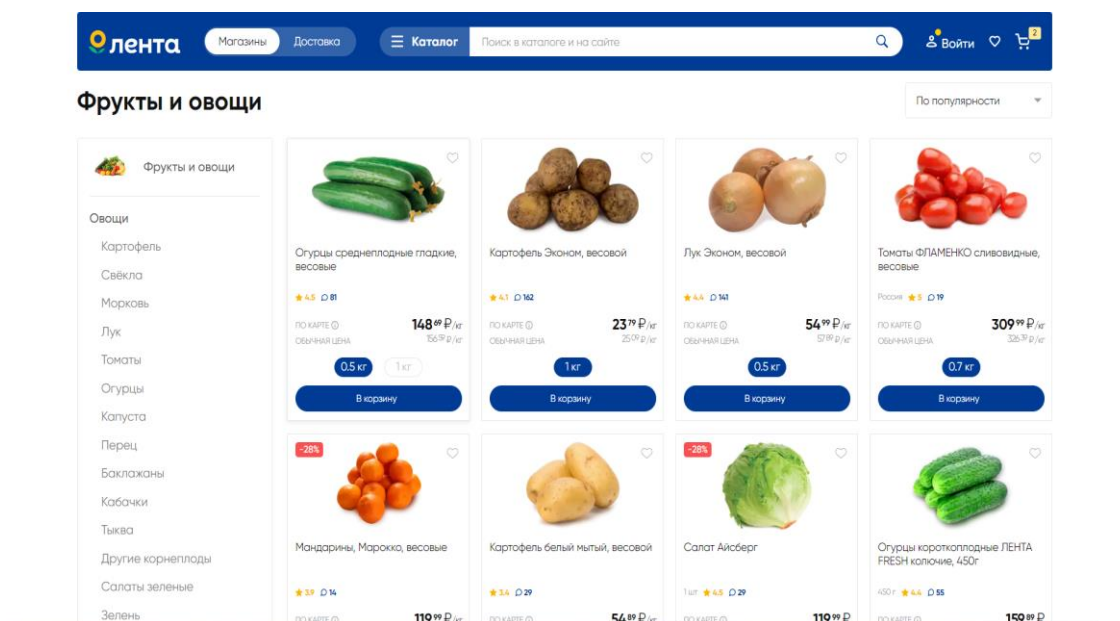


Рисунок 8 - Страница товаров сайта «Лента»

Для просмотра корзины можно нажать кнопку в шапке сайта с изображением корзины. Отобразится страница с наличием товаров или их отсутствием.

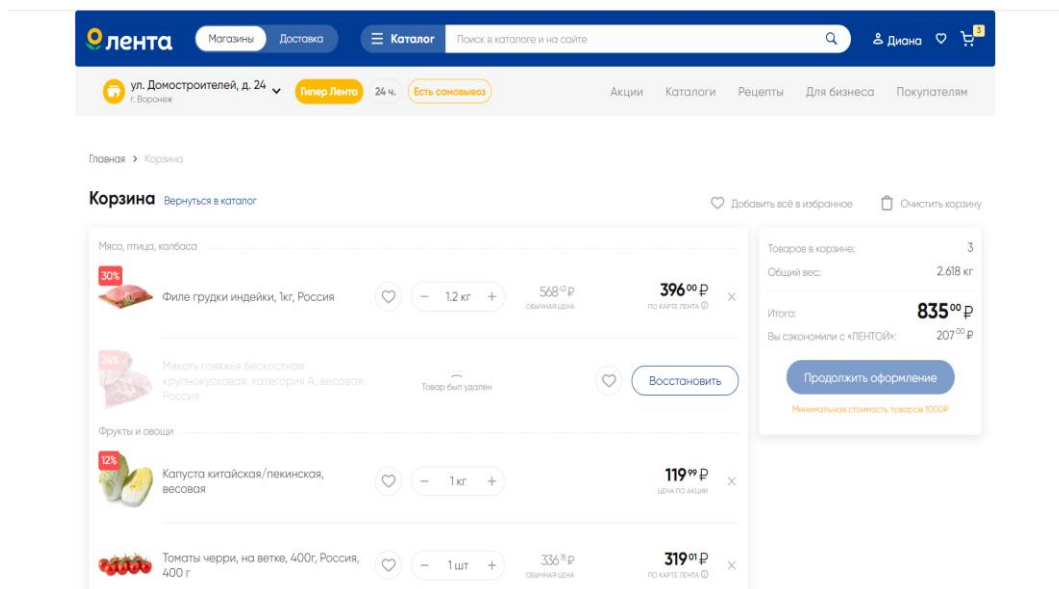


Рисунок 9 - Страница корзины сайта «Лента»

После входа в авторизованную зону пользователь может продолжить оформление заказа и ввести необходимые данные для завершения оформления. Предоставив все необходимые данные, заказ будет оформлен и доставлен.

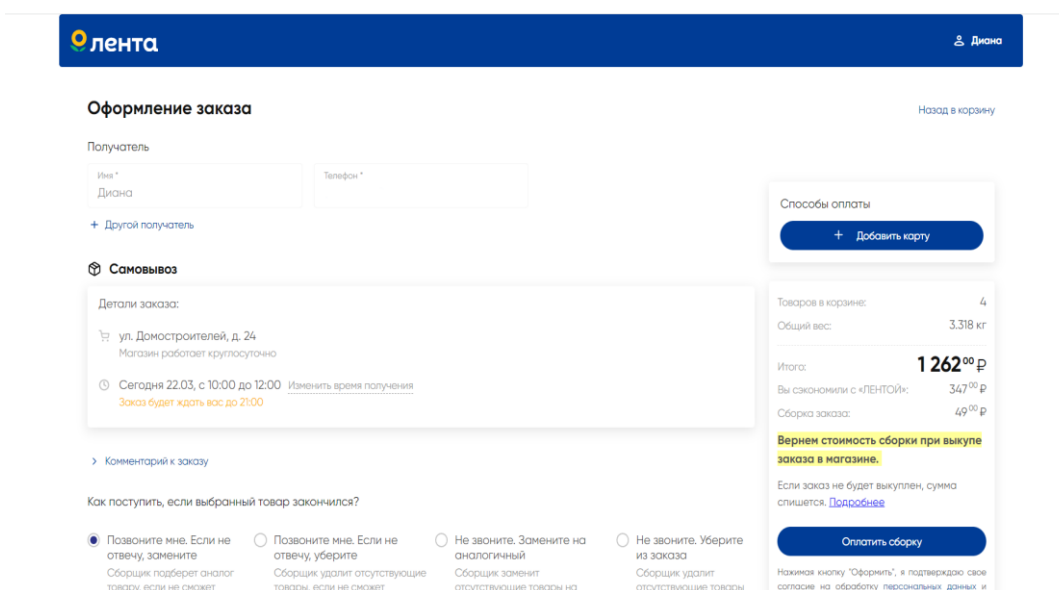


Рисунок 10 - Страница оформления заказа на сайте «Лента»

Для просмотра аккаунта нужно нажать кнопку в шапке сайта «Аккаунт». Отобразится всплывающее окно, на котором нужно выбрать ссылку на личный кабинет, и только после этого пользователь сможет попасть на страницу аккаунта [3].

Преимущества сайта:

— Сайт имеет простой и понятный интерфейс.

Недостатки сайта:

- Отсутствие некоторых товаров в магазине в зависимости от выбора адреса магазина и отсутствие возможности заказать этот товар;
- Ограниченные возможности сайта;
- Заказ невозможно оформить, не достигнув определенной суммы.

Проанализировав несколько онлайн-магазинов, можно сделать выводы над недостатками и в реализуемом проекте избежать столкновения с ними путем упрощения функциональной части пользователя и уменьшения промежуточных действий от входа пользователя на сайт до оформления заказа.

3 Диаграммы

3.1 Диаграмма прецедентов (Use-case diagram)

Диаграмма прецедентов представляет собой диаграмму, которая моделирует функциональность системы, показывая ее взаимодействие с актерами, внешними сущностями, которые взаимодействуют с системой. Диаграмма прецедентов фокусируется на функциональных возможностях системы [4].

На рисунке 11 изображена диаграмма прецедентов системы продовольственного онлайн-супермаркета.

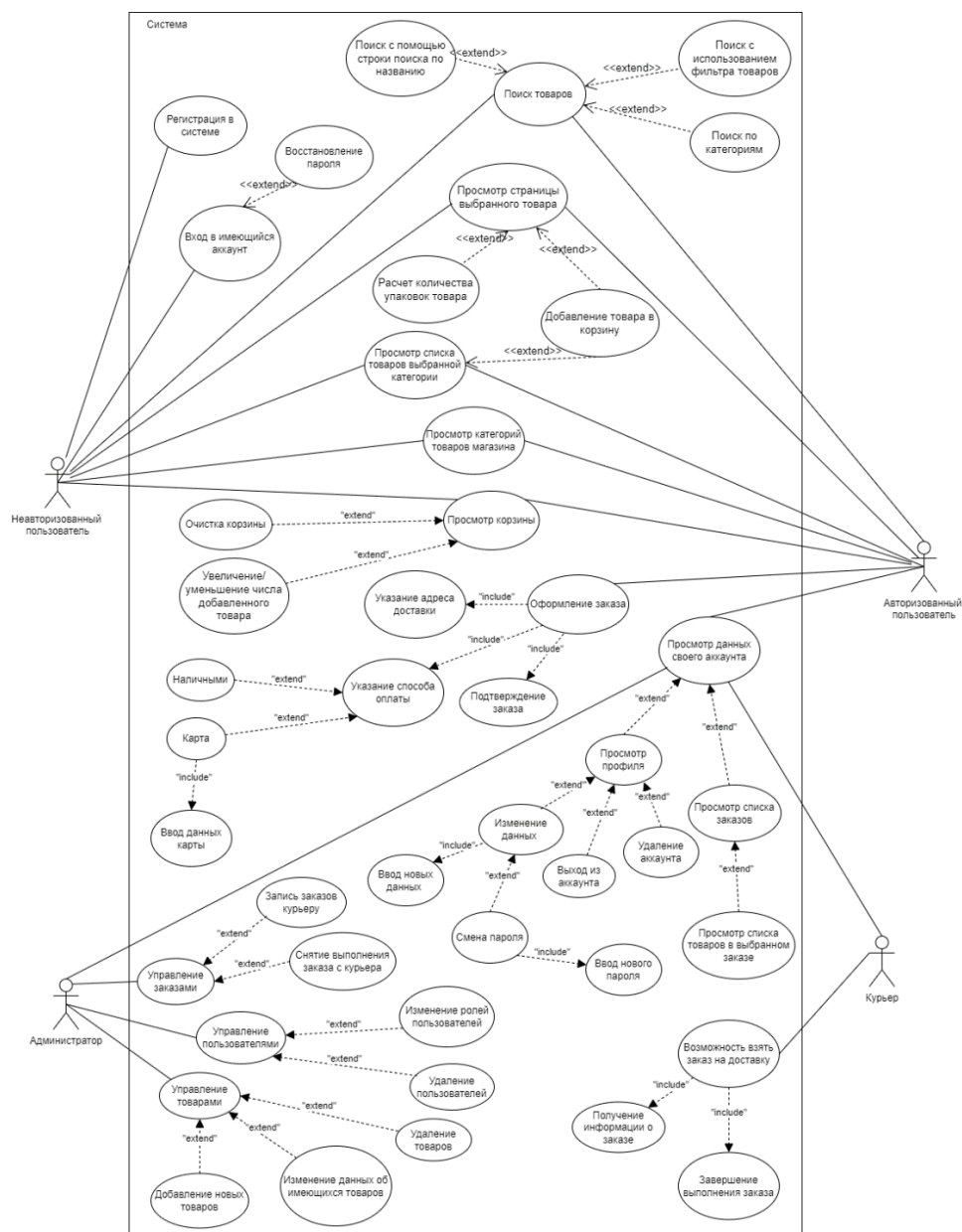


Рисунок 11 - Use-case diagram

3.2 Диаграмма классов (Class diagram)

Диаграмма классов представляет собой графическую модель, которая отображает структуру системы, описывая классы, их атрибуты, методы и взаимоотношения между классами.

На рисунке 12 изображена диаграмма классов системы продовольственного онлайн-супермаркета.

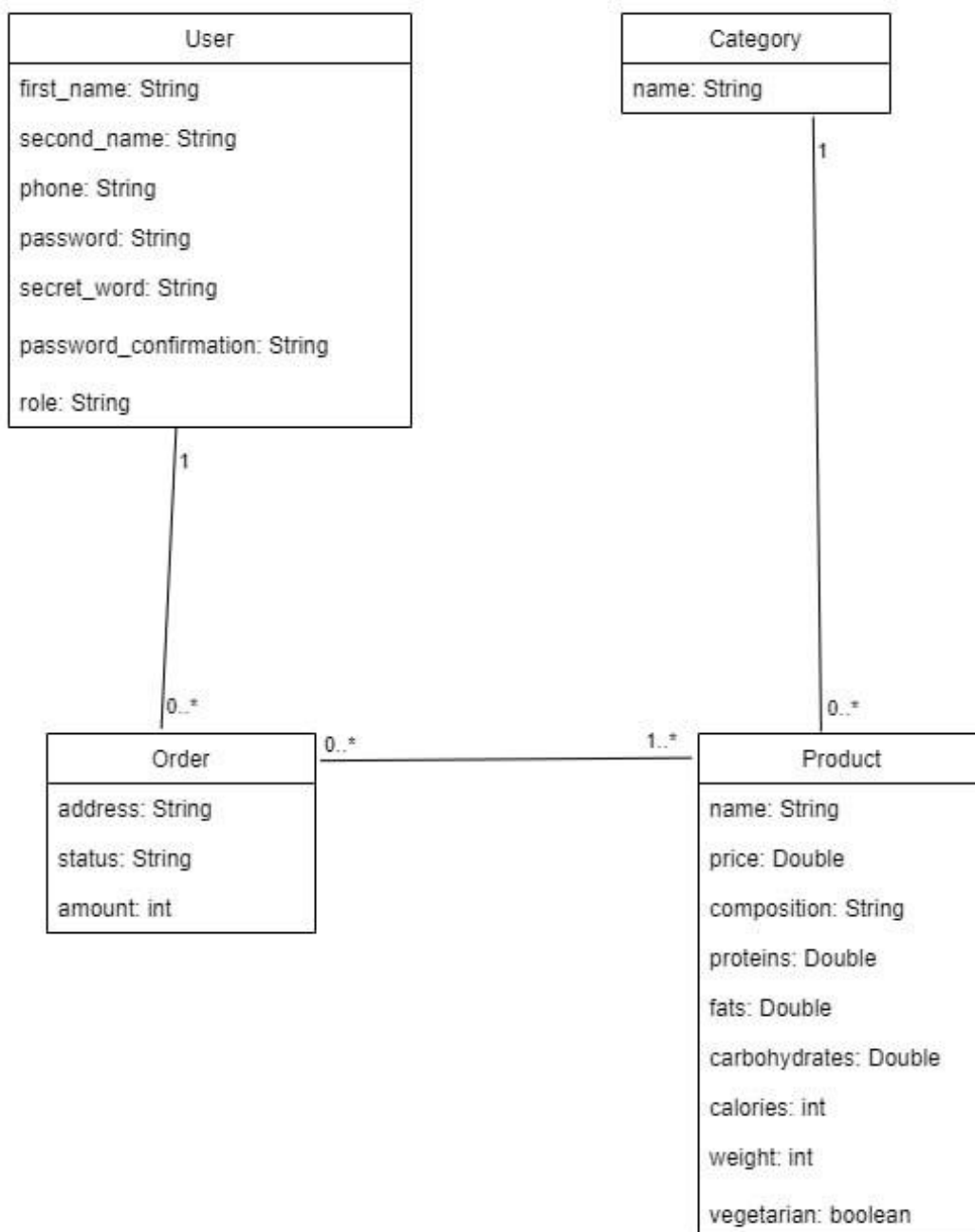


Рисунок 12 - Class diagram

3.3 Диаграмма деятельности (Activity diagram)

Диаграмма деятельности представляет собой графическую модель, которая позволяет описывать последовательность действий, процессы и поведение системы. Эта диаграмма используется для моделирования бизнес-процессов, алгоритмов и составных частей системы [5].

На рисунке 13 изображена диаграмма деятельности системы продовольственного онлайн-супермаркета.

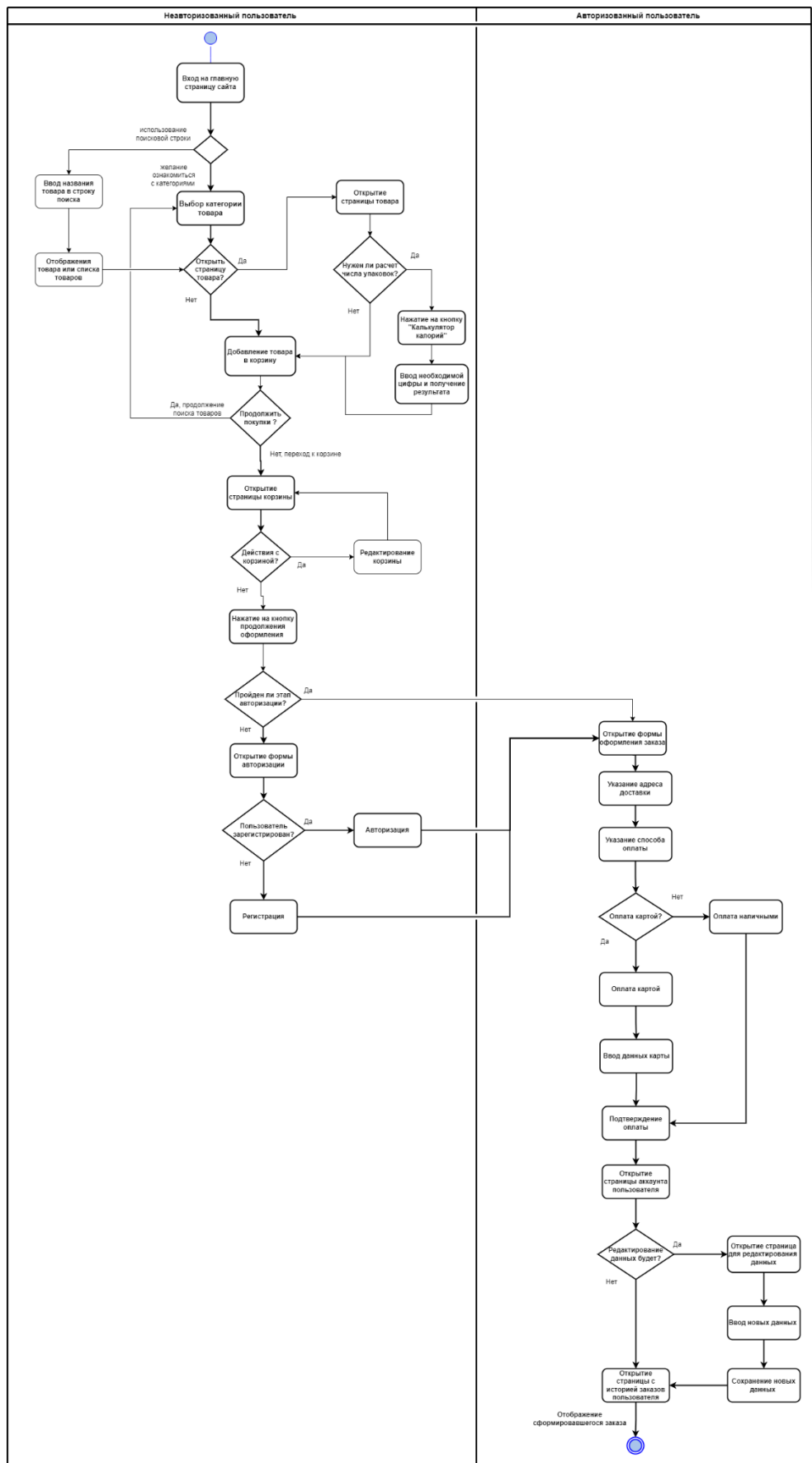


Рисунок 13 - Activity diagram

3.4 Диаграмма последовательности (Sequence diagram)

Диаграмма последовательности представляет собой диаграмму, которая моделирует взаимодействие между объектами в определенной последовательности времени. Она позволяет показать, как объекты обмениваются сообщениями и взаимодействуют друг с другом для выполнения определенной функциональности.

На рисунке 14 изображена диаграмма последовательности системы продовольственного онлайн-супермаркета.

На рисунке 15 изображена диаграмма развертывания системы продовольственного онлайн-супермаркета.

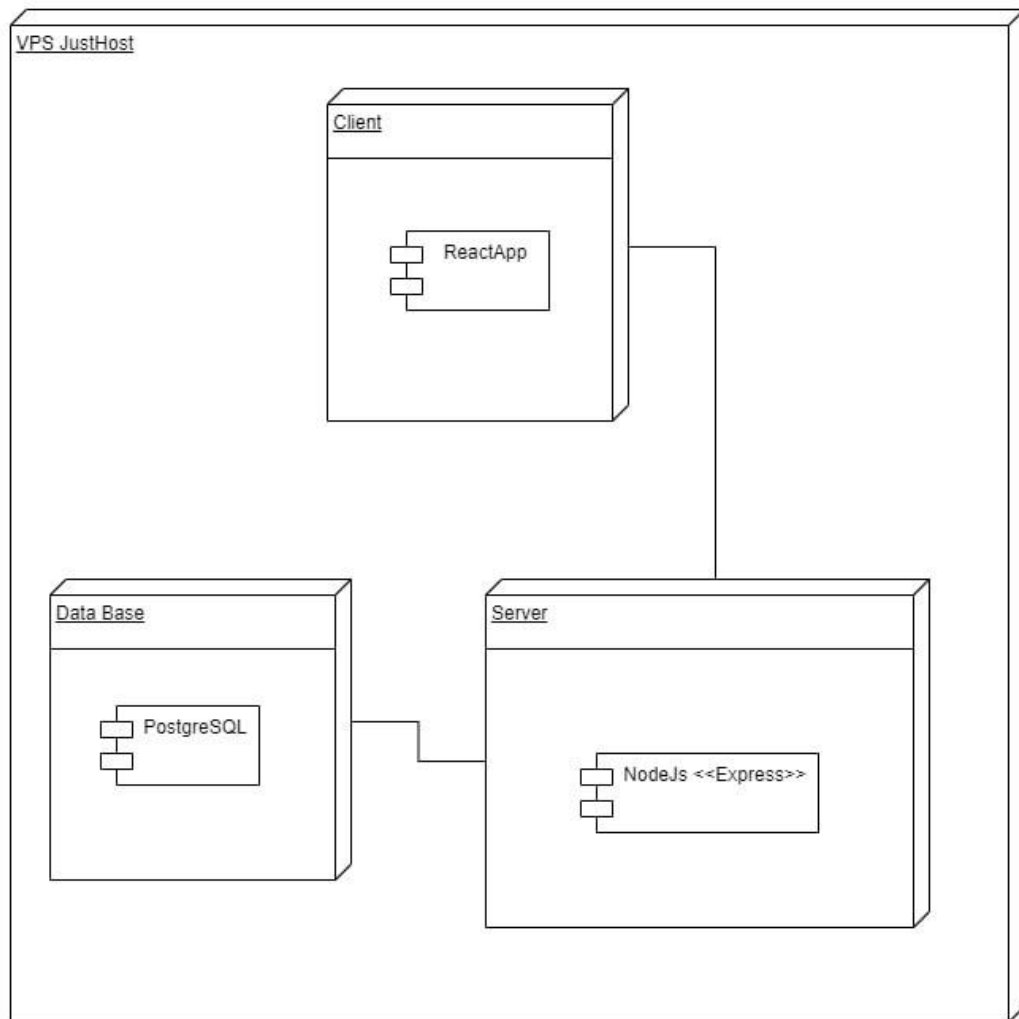


Рисунок 15 - Deployment diagram

3.6 Диаграмма сотрудничества (Collaboration diagram)

Диаграмма сотрудничества представляет собой диаграмму, которая моделирует взаимодействие между объектами или ролями в системе. Эта диаграмма позволяет показать, как объекты сотрудничают друг с другом для достижения определенной функциональности.

На рисунках 16-24 изображены диаграммы сотрудничества системы продовольственного онлайн-супермаркета с различными сценариями действий.



Рисунок 16 - Регистрация

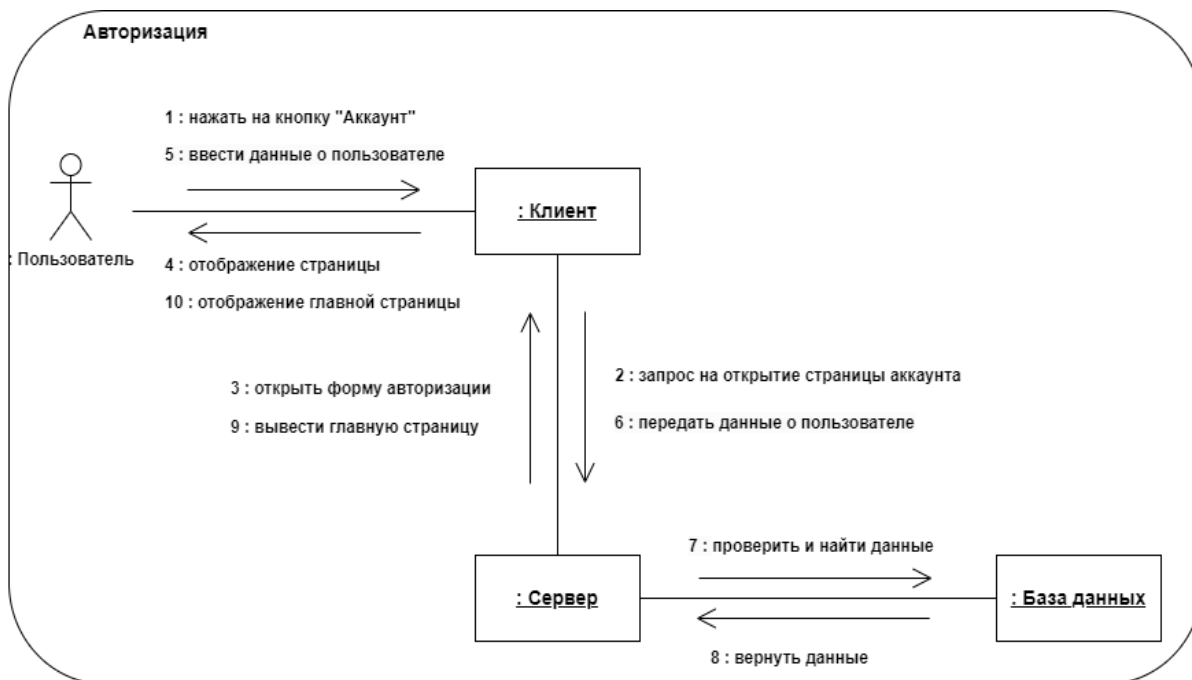


Рисунок 17 - Авторизация

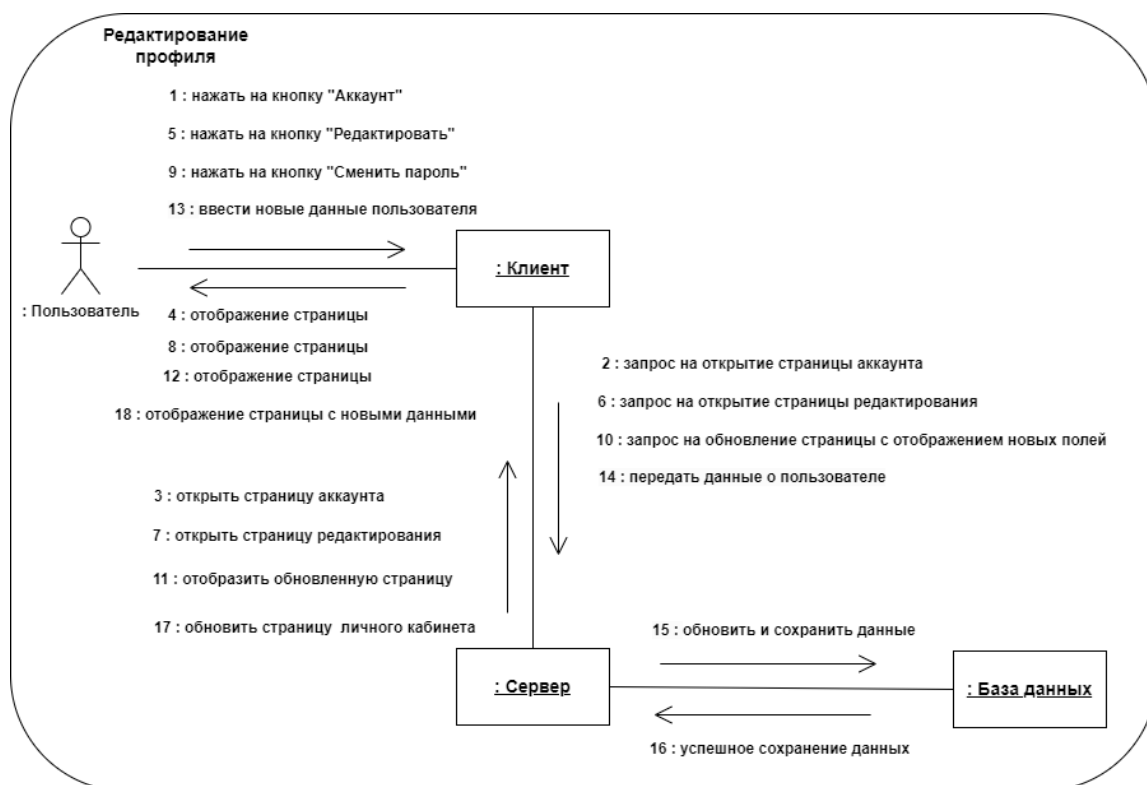


Рисунок 18 - Редактирование профиля

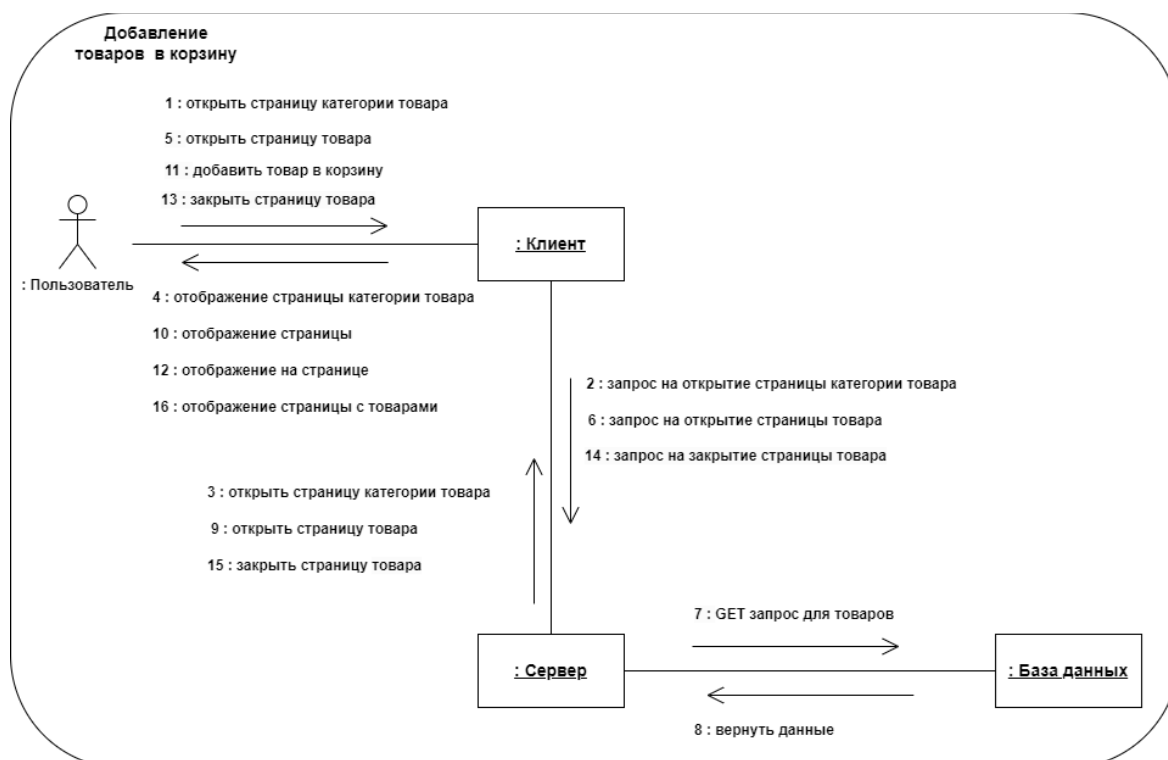


Рисунок 19 - Добавление товаров в корзину



Рисунок 20 - Взаимодействие пользователя с корзиной

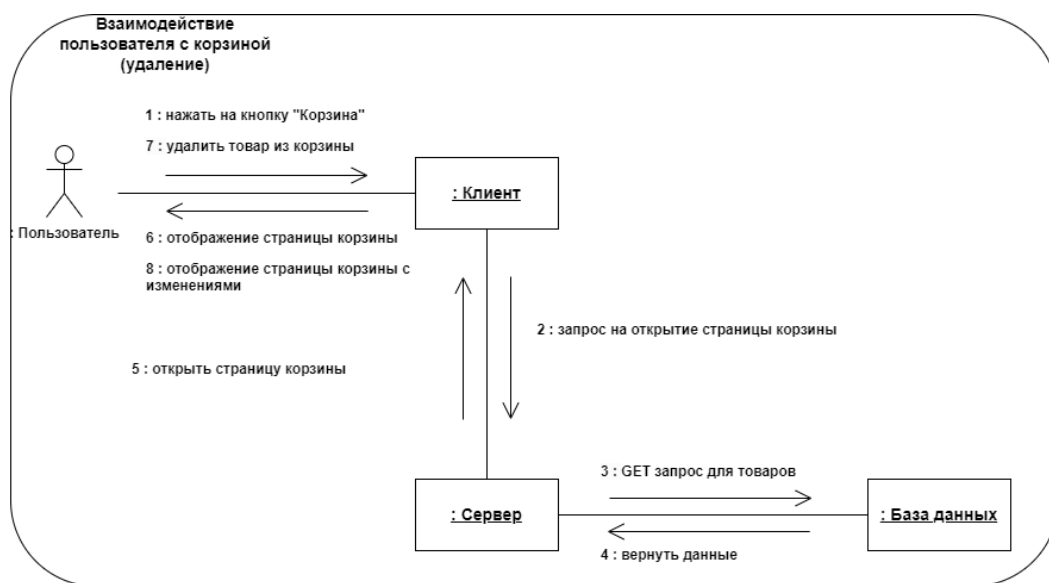


Рисунок 21 - Взаимодействие пользователя с корзиной (удаление)

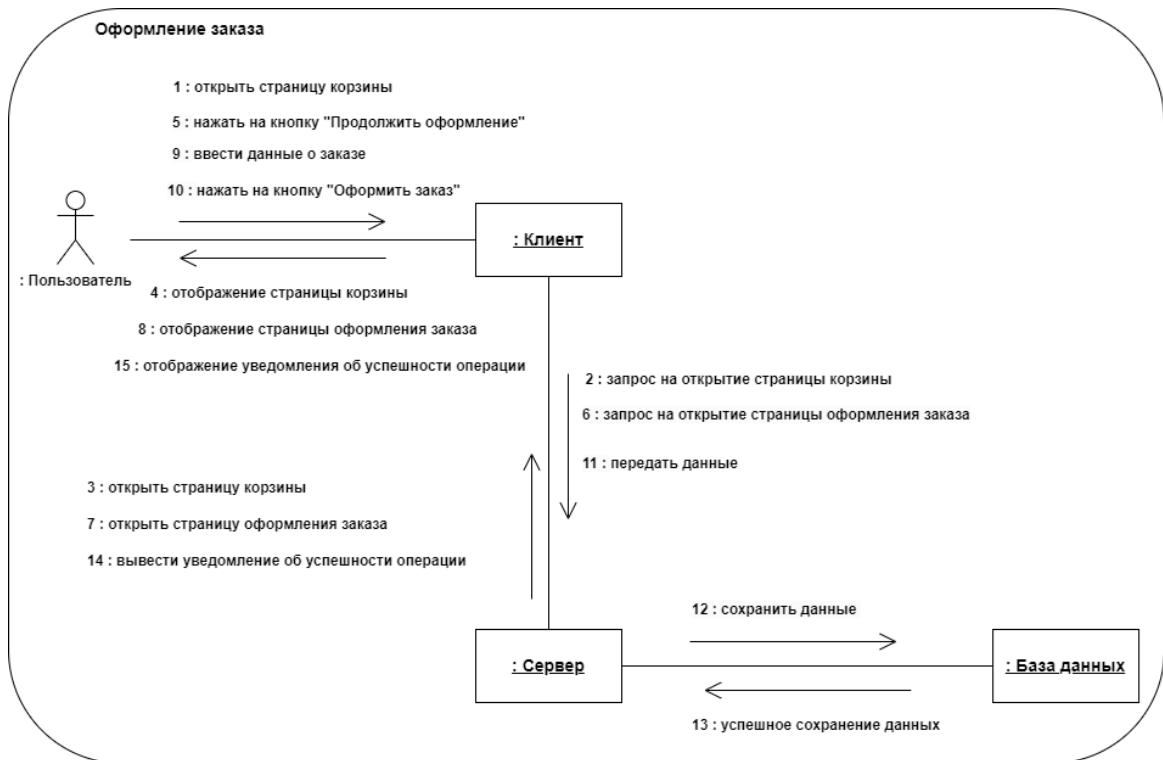


Рисунок 22 - Оформление заказа



Рисунок 23 - Взаимодействие пользователя со строкой поиска товаров

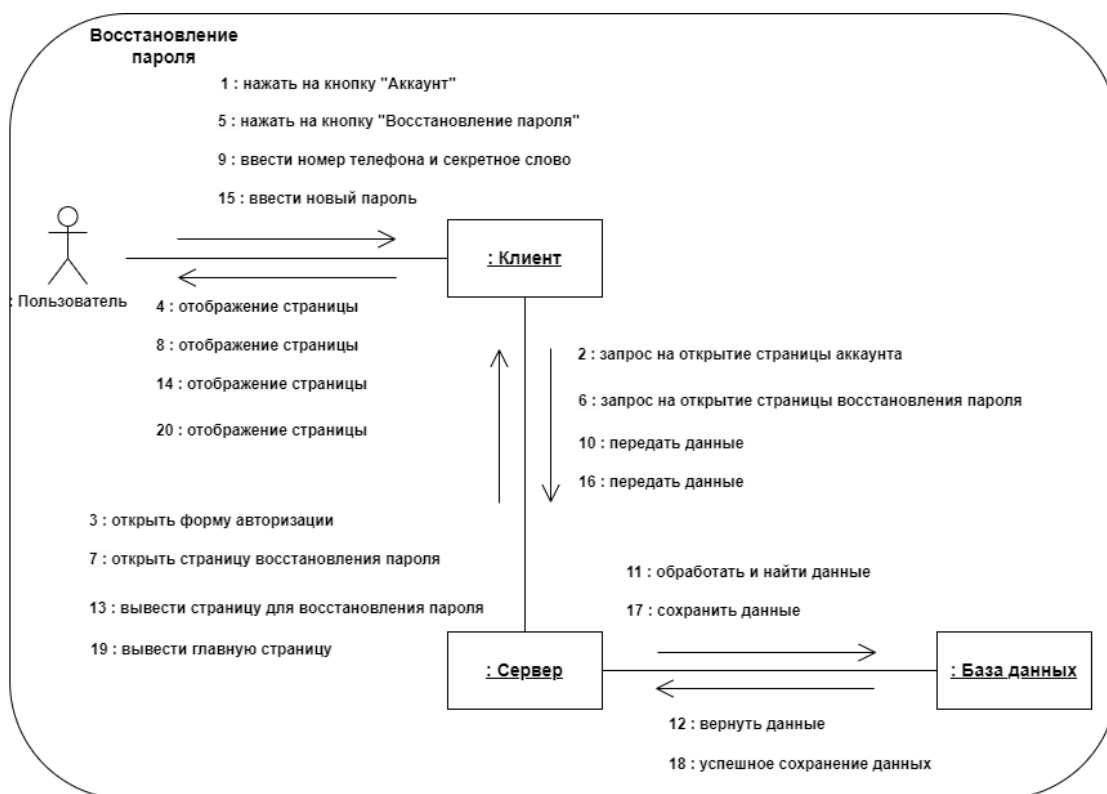


Рисунок 24 - Восстановление пароля

3.7 Диаграмма объектов (Object diagram)

Диаграмма объектов представляет собой диаграмму, которая моделирует конкретные экземпляры объектов и их взаимоотношения в определенный момент времени. Эта диаграмма позволяет показать структуру системы, отображая объекты и связи между ними.

На рисунке 25 изображена диаграмма объектов системы продовольственного онлайн-супермаркета.

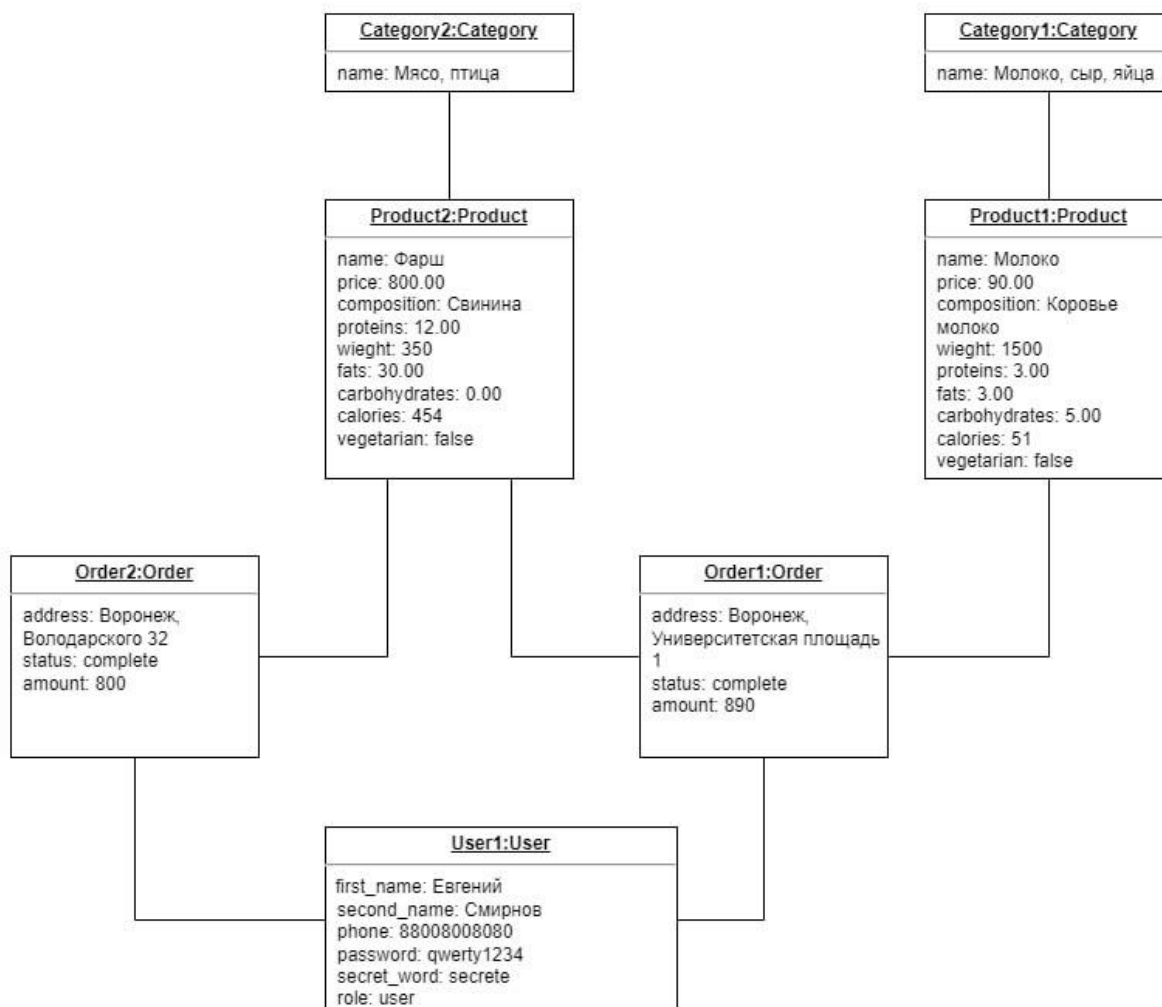


Рисунок 25 - Object diagram

3.8 Диаграмма состояний (Statechart diagram)

Диаграмма состояний представляет собой диаграмму, которая моделирует поведение объекта или системы в различных состояниях и переходах между этими состояниями. Она описывает, как объект или система реагирует на различные события, изменяя свое состояние.

На рисунке 26 изображена диаграмма состояния системы продовольственного онлайн-супермаркета.

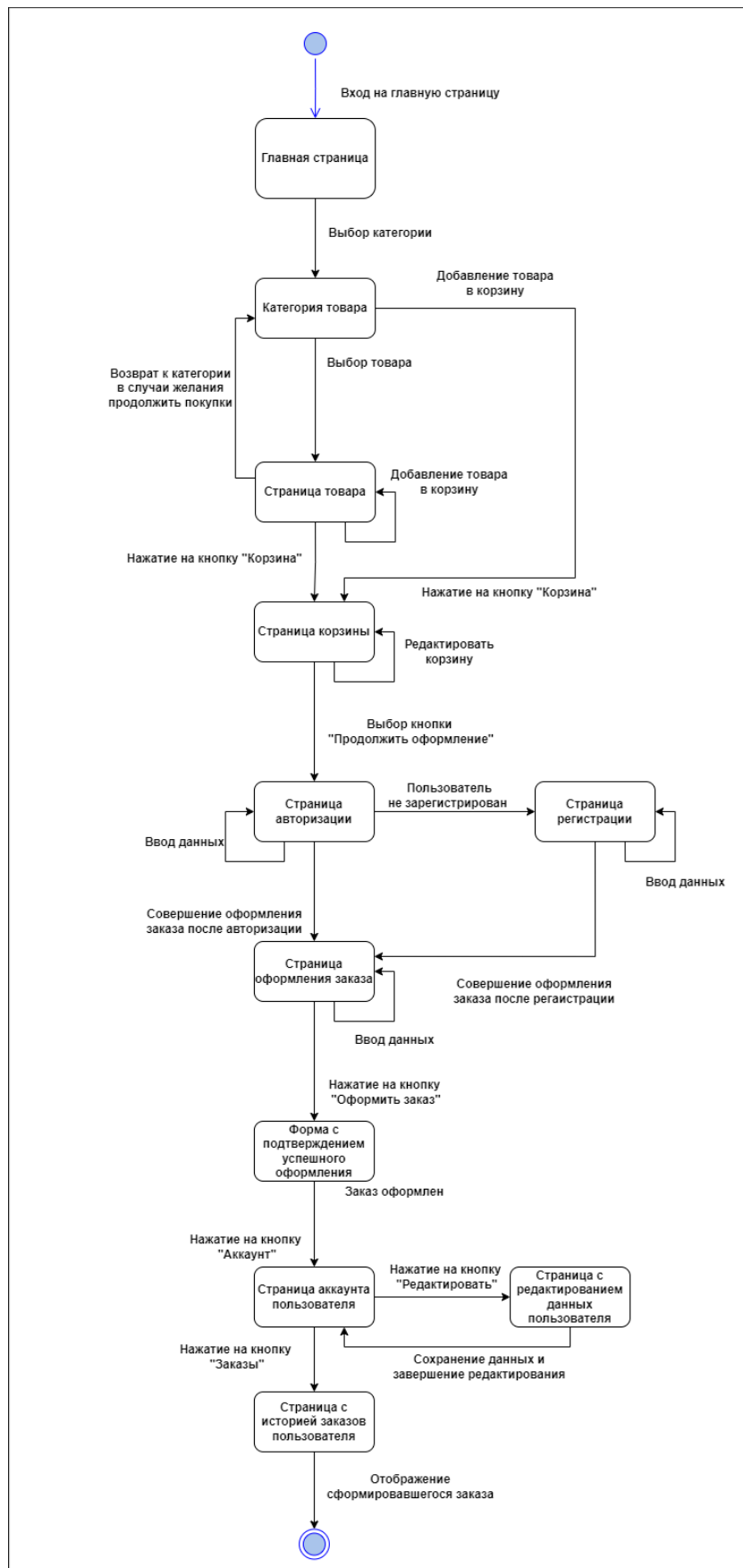


Рисунок 26 - Statechart diagram

3.9 ER–диаграмма

ER-диаграмма - это графическая модель, используемая в базах данных для описания сущностей и их взаимосвязей.

На рисунке 27 изображена ER-диаграмма системы продовольственного онлайн-супермаркета.

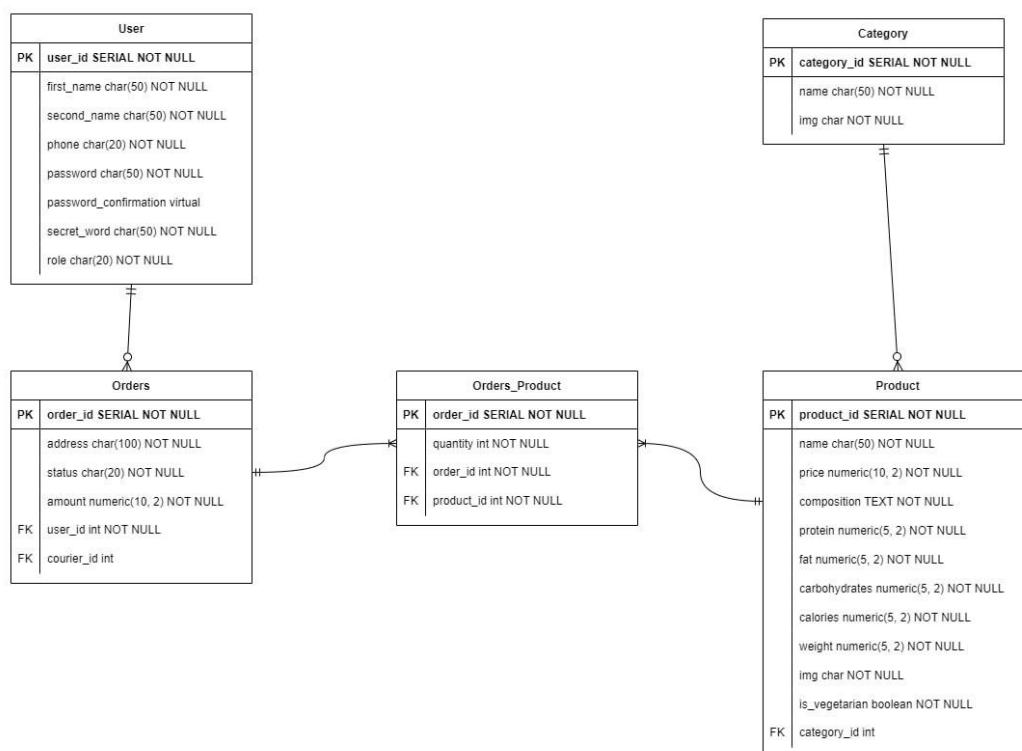


Рисунок 27 - ER-диаграмма

3.10 IDEF0

IDEF0 представляет собой графическую модель, используемую для анализа и описания функциональности системы или процесса.

На рисунке 28 изображена IDEF0 0 уровня декомпозиции. На вход системы поступает пользователь, изначально который не авторизован, но затем в процессе использования системы он может быть авторизован. Ресурсами являются администратор, курьер, сайт и товары, которые обеспечивают работу системы. Работу системы регулируют законодательство РФ и свод правил по продаже товаров. На выходе системы имеется удовлетворенный пользователь.

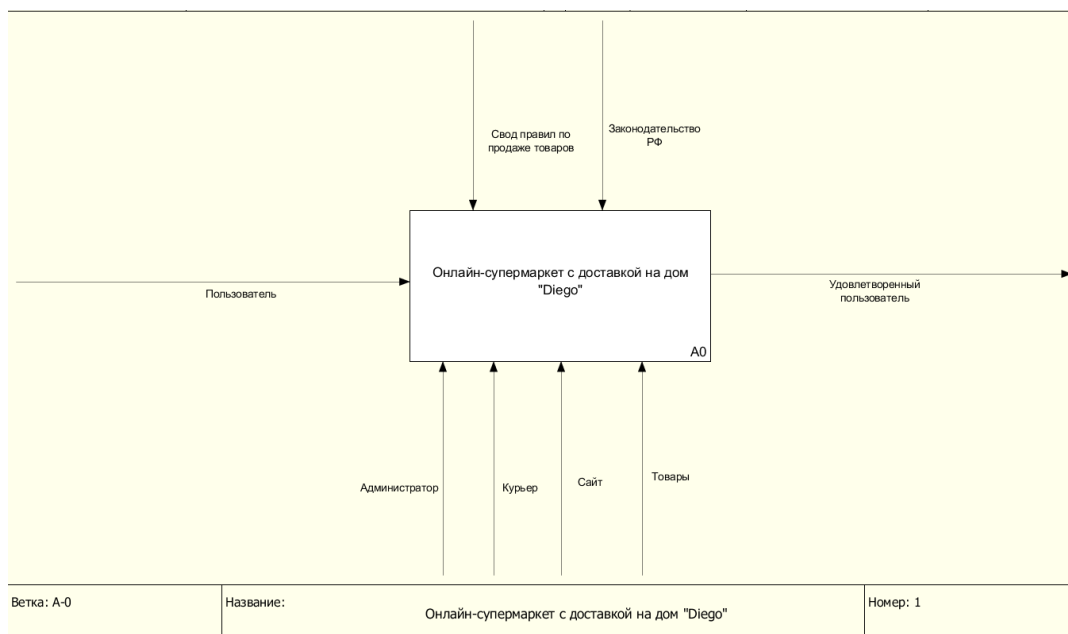


Рисунок 28 - Система онлайн-супермаркета (0 уровень)

На рисунке 29 изображен 1 уровень декомпозиции IDEF0, где работа системы разбита на несколько блоков, которые отображают подробную работу системы.

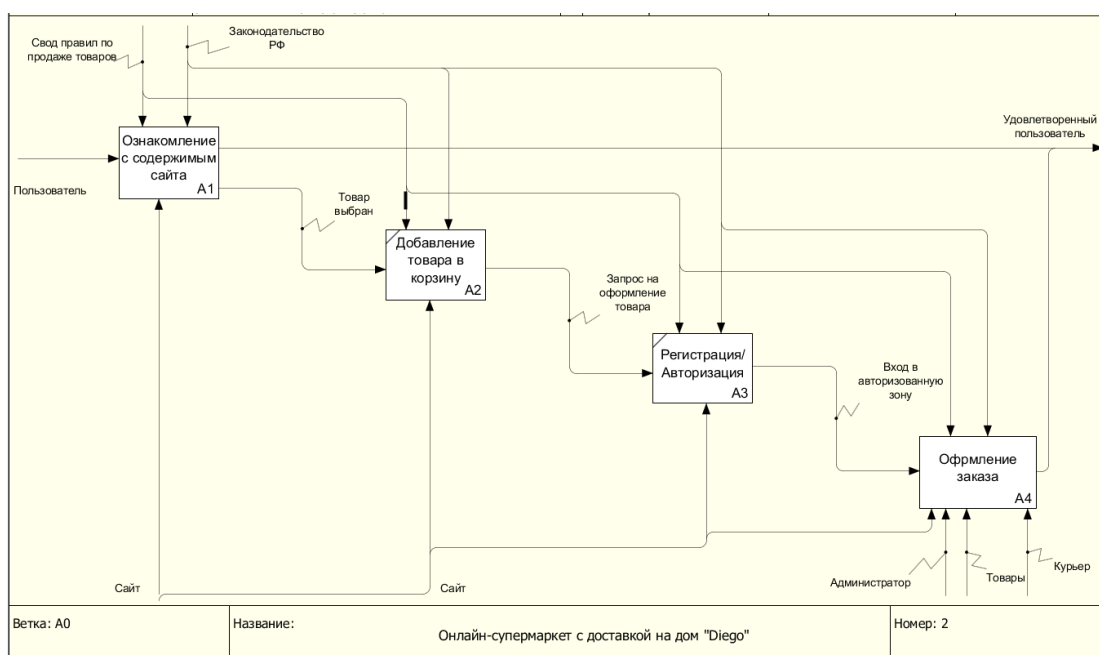


Рисунок 29 - Система онлайн-супермаркета (1 уровень)

На рисунке 30 изображен 2 уровень декомпозиции IDEF0, где процесс ознакомления с содержимым сайта разбит на несколько блоков, которые отображают подробный путь пользователя по сайту.

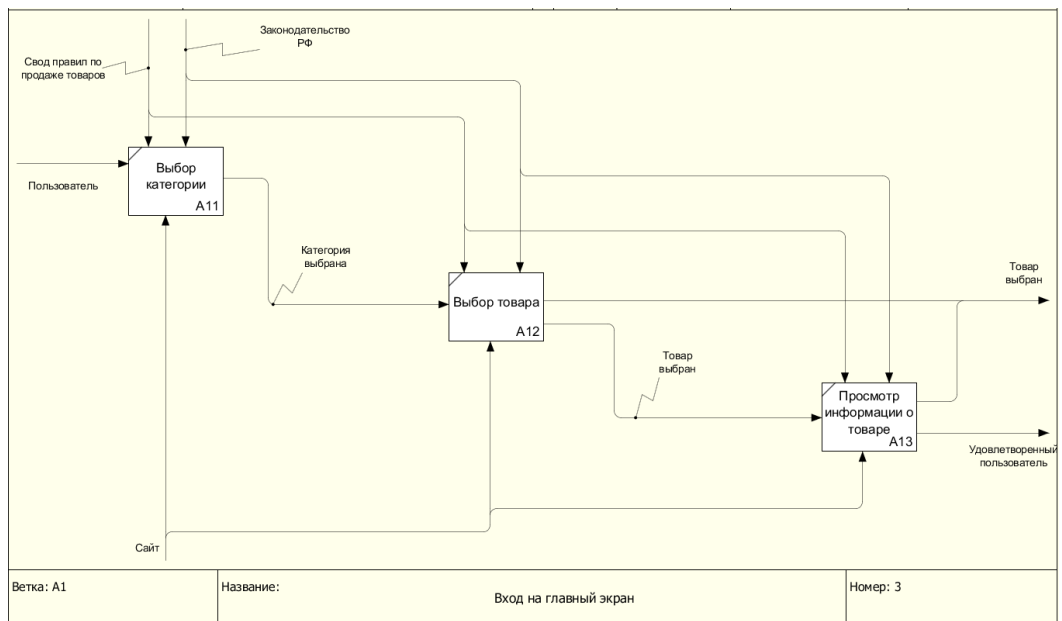


Рисунок 30 - Система онлайн-супермаркета (2 уровень)

На рисунке 31 изображен 2 уровень декомпозиции IDEF0, где процесс оформления заказа разбит на несколько блоков, которые отображают подробный путь пользователя до успешного получения заказа.

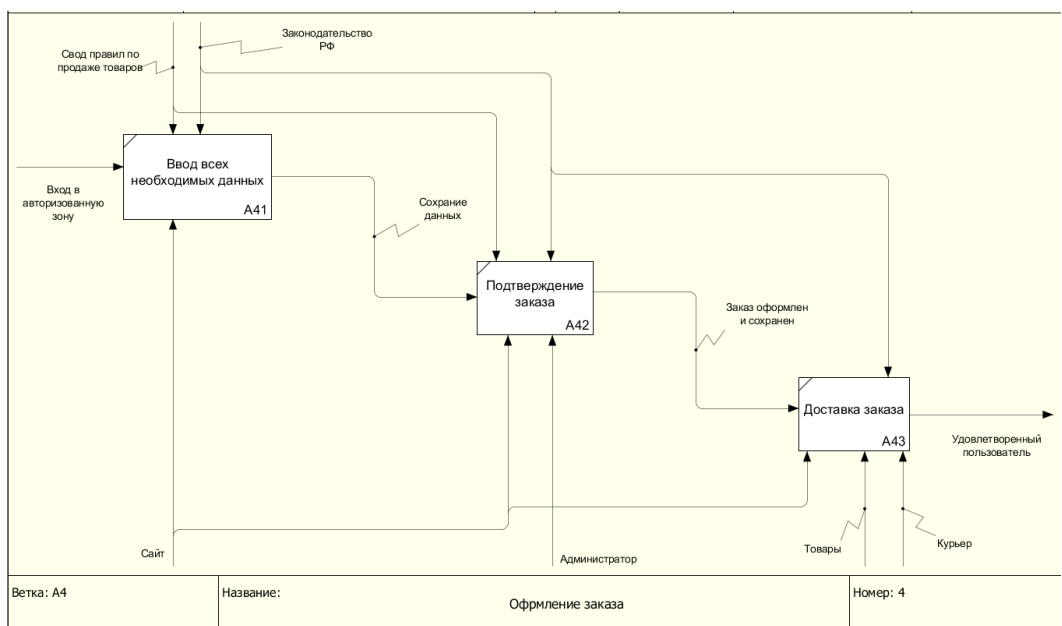


Рисунок 31 - Система онлайн-супермаркета (2 уровень)

4 Реализация

4.1 Средства реализации

Веб-приложение использует клиент-серверную архитектуру и разделяется на две основные части: backend и frontend. Обмен данными между ними осуществляется посредством REST API.

В процессе разработки был использован Git-flow для организации работы. В нашем проекте эта модель ветвления включает две основные ветки: main и develop, а также временные ветки для добавления новых функций.

Для разработки серверной части сайта был выбран следующий ряд технологий:

- JavaScript — высокоуровневый язык программирования, который широко используется для разработки интерактивных веб-страниц и веб-приложений. JavaScript позволяет добавлять динамическое поведение к веб-страницам, взаимодействовать с пользователем, обрабатывать и валидировать данные на стороне клиента, создавать анимацию, выполнять асинхронные запросы к серверу [6];
- Node.js — среда выполнения JavaScript, позволяет запускать JavaScript на сервере и выполнять различные серверные операции. Node.js предоставляет множество встроенных модулей, которые облегчают разработку серверных приложений, таких как работа с файловой системой, сетевыми запросами, создание веб-серверов;
- Express — веб-фреймворк для Node.js, который облегчает разработку веб-приложений и API. Особенности фреймворка являются маршрутизация, управление сессиями и состояниями, использование промежуточного программного обеспечения;
- Swagger — набор инструментов и спецификаций для разработки, проектирования и документирования REST API. Он предоставляет возможность описывать структуру, операции, параметры и ответы API в удобном для чтения и понимания формате [7].

Для разработки клиентской части сайта был выбран следующий ряд технологий:

- HTML — стандартизированный язык разметки, используемый для создания веб-страниц. Он определяет структуру и содержимое веб-страницы с помощью различных тегов и атрибутов. HTML состоит из элементов, которые обрамляют содержимое веб-страницы и определяют его семантику;
- CSS — это язык стилей, используемый для определения внешнего вида и форматирования веб-страниц, созданных с помощью HTML. CSS позволяет управлять цветами, шрифтами, размерами, расположением элементов и другими аспектами визуального представления веб-страницы;
- JavaScript — высокоуровневый язык программирования, который широко используется для разработки интерактивных веб-страниц и веб-приложений;
- React — JavaScript-библиотека для создания пользовательских интерфейсов. Она позволяет разработчикам создавать масштабируемые и переиспользуемые компоненты, которые обновляются автоматически при изменении данных. React основан на концепции компонентного подхода, что делает код более организованным, понятным и легко поддерживаемым [8].

4.2 Разработка frontend

Клиентская часть приложения была написана на языке JavaScript с использованием фреймворка React. Для отправки запросов с клиентской части приложения на серверную часть используется библиотека `axios`. Она предоставляет интерфейс для выполнения HTTP-запросов прямо из браузера.

Для создания модульной и переиспользуемой архитектуры, приложение было разбито на компоненты. Каждый компонент отвечает за конкретную

функциональность или отображение определенной части пользовательского интерфейса.

Проект имеет следующую структуру (Рисунок 32):

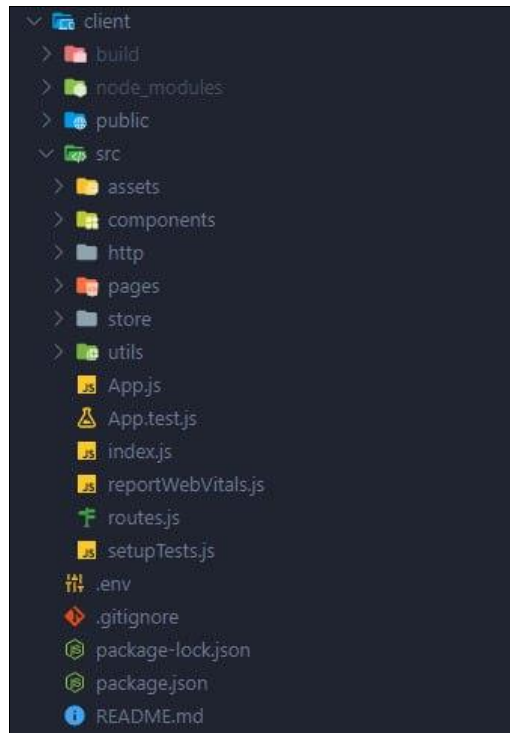


Рисунок 32 - Структура клиентской части приложения

- Модуль «assets». Этот модуль содержит все статические ресурсы, такие как изображения (img) и стили (styles);
- Модуль «components». Этот модуль содержит компоненты, которые используются в приложении;
- Модуль «http». Этот модуль содержит файлы, связанные с сетевыми запросами и обработкой данных;
- Модуль «pages». Этот модуль содержит компоненты, представляющие отдельные страницы приложения;
- Модуль «store». Этот модуль содержит файлы, связанные с управлением состоянием приложения;
- Модуль «utils». Этот модуль содержит вспомогательные функции и константы маршрутов;

- Файл «index.js». Этот файл является точкой входа в приложение. Он служит главным файлом, с которого начинается выполнение приложения.

4.3 Разработка backend

Серверная часть приложения была написана на языке JavaScript с использованием Node.js и фреймворка Express. Проект имеет следующую структуру и модули (Рисунок 33):



Рисунок 33 - Структура серверной части приложения

- Файл «index.js». Этот файл является точкой входа в приложение. Он содержит код, который запускает сервер Express, подключает необходимые модули, маршруты, middleware;
- Модуль «controller». Этот модуль содержит функции и методы, ответственные за обработку запросов от клиентской части приложения. Внутри данного модуля находятся контроллеры, каждый из которых отвечает за определенные маршруты и бизнес-логику. Контроллеры принимают входные данные, вызывают соответствующие методы модуля "models" для выполнения операций с данными и возвращают результаты обратно клиенту;

- Модуль «error». Этот модуль содержит класс `ApiError`, который играет роль пользовательского класса ошибок. Он предназначен для упрощения создания и обработки ошибок;
- Модуль «middleware». Этот модуль содержит промежуточное программное обеспечение (middleware), которое выполняет определенные операции перед или после обработки запросов контроллерами. Это функции для валидации полей регистрации, валидации адреса в заказе и проверки на аутентификацию пользователя;
- Модуль «models». Этот модуль содержит модели данных, которые отображают сущности приложения и взаимодействуют с базой данных. Внутри данного модуля определены схемы данных, методы для выполнения операций чтения, записи, обновления или удаления данных;
- Модуль «routes». Этот модуль определяет маршруты приложения, которые соответствуют определенным URL-адресам. Внутри данного модуля содержатся обработчики маршрутов, которые связывают URL-адреса с соответствующими контроллерами. Маршруты определяются с использованием методов Express, таких как `get`, `post`, `put`, `delete`, и указываются пути и обработчики для каждого маршрута;
- Модуль «static». Этот модуль содержит изображения, которые должны быть доступны напрямую из браузера;
- Модуль «swagger». Этот модуль содержит файлы, связанные с документацией API приложения. Внутри данного модуля определяются спецификации маршрутов, параметры запросов, схемы данных.

Для развертывания приложения был использован виртуальный сервер (VPS) от провайдера JustHost. В качестве веб-сервера был выбран Nginx. Приложение было развернуто на этом сервере, а Nginx использован для обслуживания статического контента и проксирования запросов к приложению.

Для описания спецификации API использовался Swagger. Для интеграции его в проект потребовалось добавить библиотеки «swagger-jsdoc», «swagger-ui-express».

4.4 Основная функциональность сайта

4.5 Для пользователя

4.5.1 Оформление заказа

При входе на сайт пользователю открывается главная страница сайта с категориями продуктов (Рисунок 34).

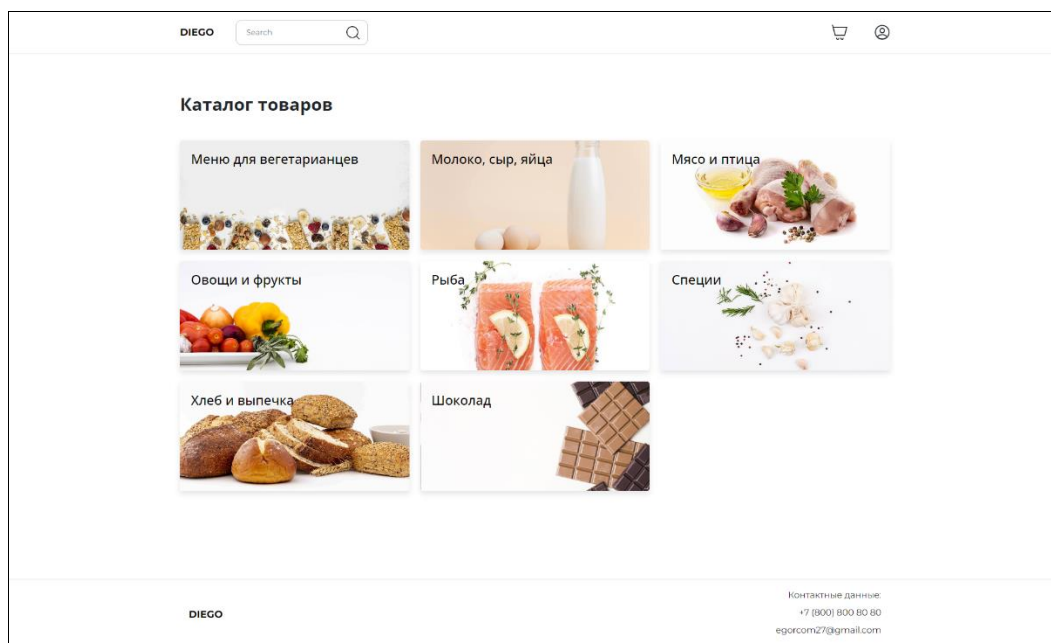


Рисунок 34 - Главная страница

На этой странице пользователь может выбрать интересующую категорию или найти необходимый товар, используя строку поиска. Кроме этого пользователь может зарегистрироваться или авторизоваться, нажав на кнопку аккаунта в шапке сайта. А также с этой страницы возможен переход к корзине.

После выбора нужной категории пользователь переходит к странице с товарами (Рисунок 35), где он может выбрать товар, нажать на него и перейти к странице товара (Рисунок 36) со всей необходимой информацией о товаре. Если пользователь захочет приобрести выбранный продукт, он может

добавить товар в корзину на странице товара, нажав на кнопку «В корзину», и перейти на страницу корзины (Рисунок 37) для дальнейшего оформления заказа.

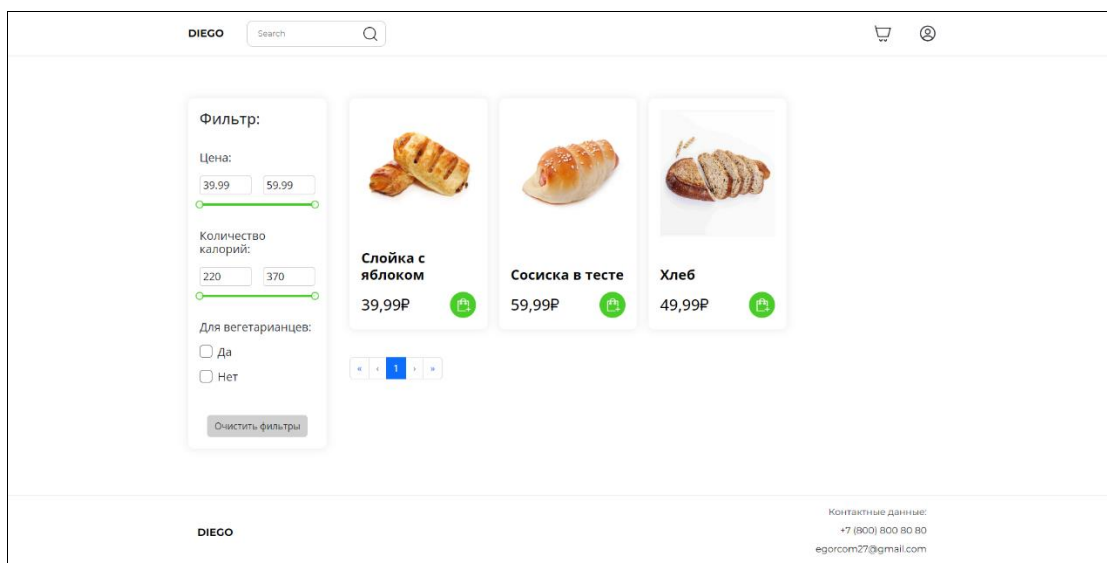


Рисунок 35 - Страница товаров

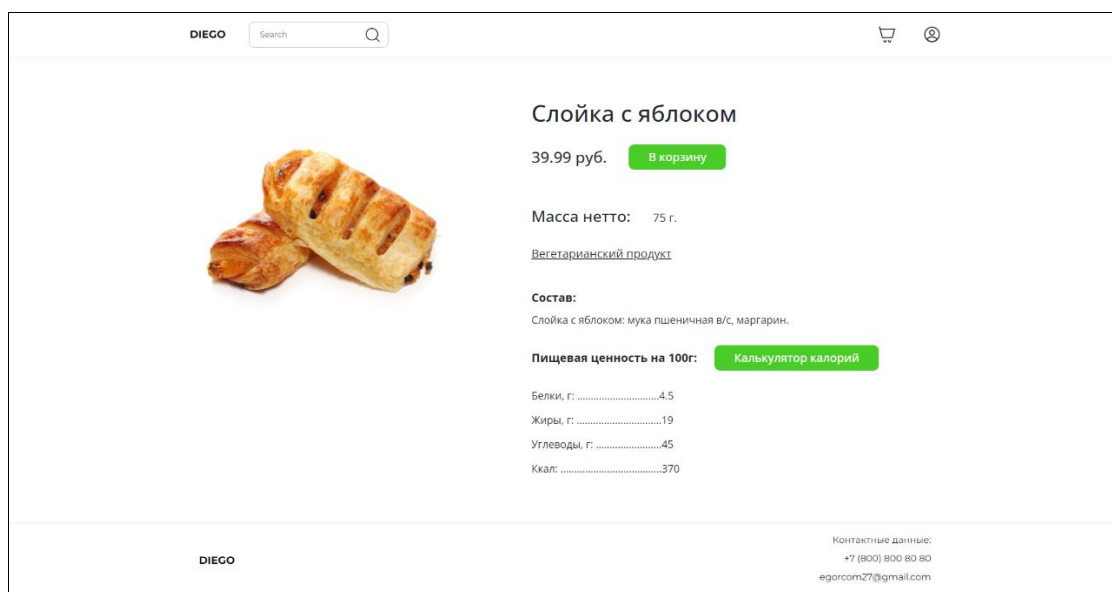


Рисунок 36 - Страница товара

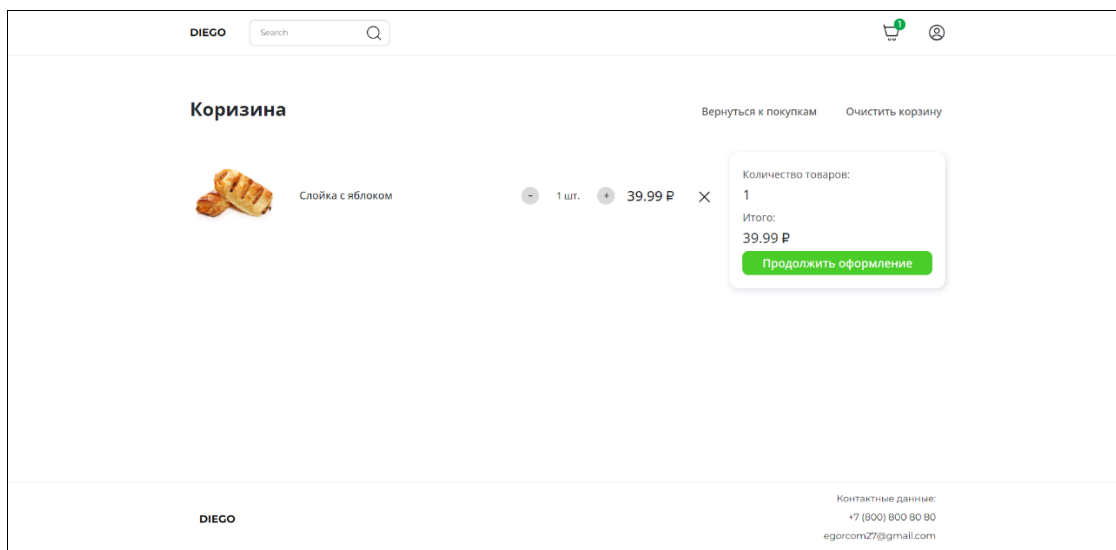


Рисунок 37 - Страница корзины

На странице корзины пользователь может ознакомиться с добавленным товаром и для осуществления оформления должен нажать на кнопку «Продолжить оформление».

Если пользователь не авторизован в системе, после нажатия на кнопку перед ним появится окно с выбором дальнейших действий (Рисунок 38), где пользователь может выбрать вариант регистрации или авторизации.

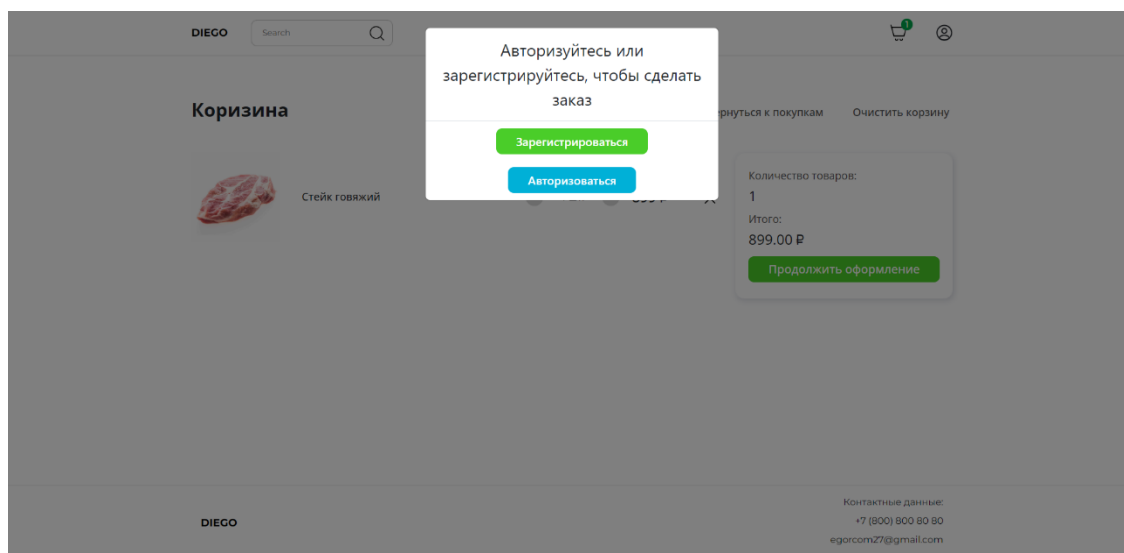
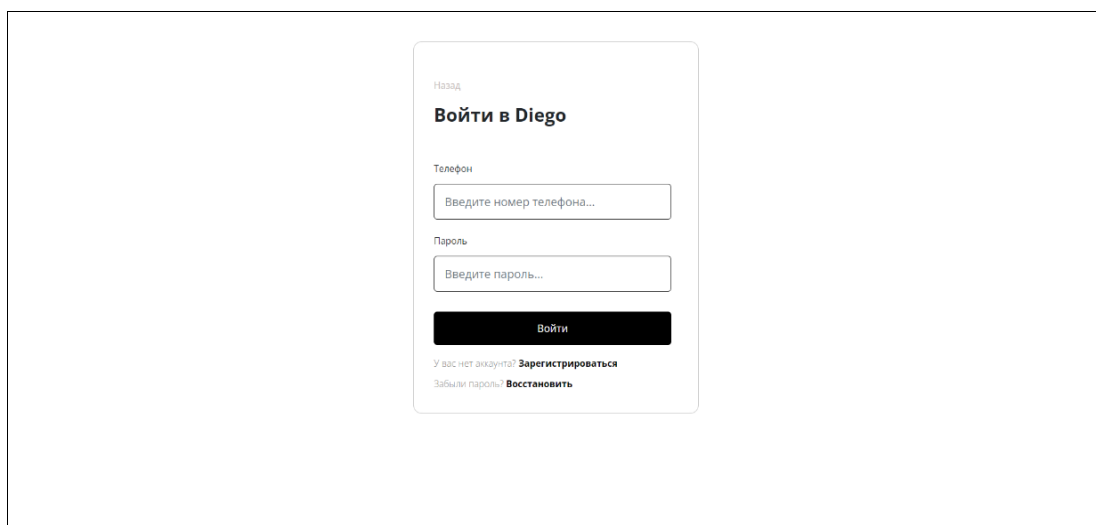


Рисунок 38 - Вход в систему для продолжения оформления заказа

В случае выбора авторизации, пользователь нажимает на кнопку «Авторизоваться», и перед ним открывается страница авторизации (Рисунок

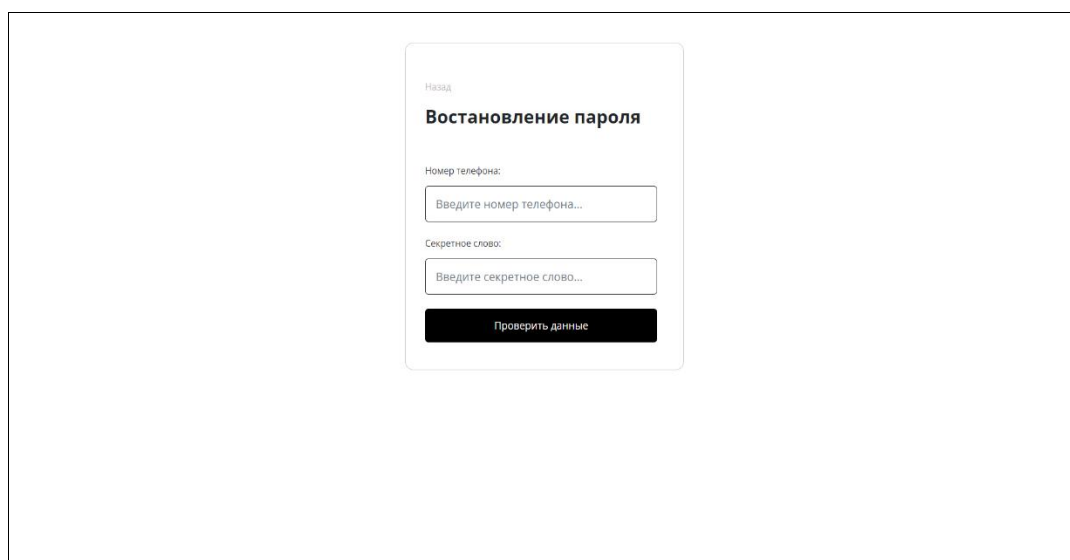
39). Если пользователь забыл пароль от аккаунта, есть возможность его восстановить при нажатии на кнопку «Восстановить пароль» на странице авторизации (Рисунок 39).



The screenshot shows a login form titled "Войти в Diego". At the top left is a "Назад" (Back) link. The form contains two input fields: "Телефон" (Phone) with the placeholder "Введите номер телефона..." and "Пароль" (Password) with the placeholder "Введите пароль...". Below these is a black button labeled "Войти" (Login). At the bottom, there are two links: "У вас нет аккаунта? Зарегистрироваться" (Don't have an account? Register) and "Забыли пароль? Восстановить" (Forgot password? Recover).

Рисунок 39 - Страница авторизации

Восстановление пароля проходит в двух этапах. Сначала пользователь вводит свои данные для проверки личности системой (Рисунок 40), после проверки уже возможна смена пароля (Рисунок 41).



The screenshot shows a verification form titled "Восстановление пароля" (Password Recovery). At the top left is a "Назад" (Back) link. The form contains two input fields: "Номер телефона:" (Phone number) with the placeholder "Введите номер телефона..." and "Секретное слово:" (Secret word) with the placeholder "Введите секретное слово...". Below these is a black button labeled "Проверить данные" (Check data).

Рисунок 40 - Страница восстановления пароля с проверкой данных

Назад

Восстановление пароля

Новый пароль:

Введите новый пароль...

Повторите пароль:

Повторите пароль...

Изменить пароль

Рисунок 41 - Страница восстановления пароля

Если пользователь не зарегистрирован в системе, он нажимает на кнопку «Зарегистрироваться» на странице авторизации (Рисунок 39), и перед ним открывается страница регистрации (Рисунок 42).

Назад

Регистрация на Diego

Имя

Введите свое имя...

Фамилия

Введите свою фамилию...

Телефон

Введите номер телефона...

Пароль

Введите пароль...

Повторите пароль:

Повторите пароль...

Секретное слово

Введите секретное слово...

Зарегистрироваться

У вас уже есть аккаунт? [Войти](#)

Рисунок 42 - Страница регистрации

После входа в систему пользователь сможет продолжить оформление заказа. Перед ним откроется главная страница, откуда он может перейти в корзину и продолжить оформление заказа (Рисунок 43), где ему необходимо будет указать все запрашиваемые данные для успешного оформления.

The screenshot shows the 'Оформить заказ' (Place Order) page of the DIEGO website. At the top, there is a header with the 'DIEGO' logo, a search bar, and icons for a shopping cart and user profile. The main content area contains two form fields: 'Адрес доставки' (Delivery Address) with a placeholder 'Введите адрес доставки' (Enter delivery address) and 'Способ оплаты' (Payment Method) with a placeholder 'Выберите способ оплаты' (Select payment method). Below these fields is a blue button labeled 'Оформить заказ' (Place Order). At the bottom of the page, there is a footer with the 'DIEGO' logo on the left and contact information on the right: 'Контактные данные: +7 (800) 800 80 80, egorcom27@gmail.com'.

Рисунок 43 - Страница оформления заказа

Пользователь должен указать адрес и способ оплаты. Если он выбирает оплату картой, то при нажатии на кнопку «Оформить заказ» появится форма для заполнения данных о карте (Рисунок 44). После заполнения пользователь должен нажать на кнопку «Сохранить» и на странице оформления заказа повторно нажать на кнопку «Оформить заказ», и тогда появится сообщение об успешной операции (Рисунок 45).

The screenshot shows the 'Оплата картой' (Card Payment) modal form overlaid on the 'Оформить заказ' (Place Order) page. The modal form has a title bar with a close button (X). It contains three input fields: 'Номер карты' (Card Number) with a placeholder 'Введите номер карты' (Enter card number), 'Срок действия' (Expiration Date) with two sub-inputs for 'ММ' (Month) and 'ГГ' (Year), and 'CVV' with a placeholder 'Введите CVV' (Enter CVV). At the bottom of the modal, there are two buttons: 'Закрыть' (Close) and 'Оплатить' (Pay). The background page is dimmed, showing the 'Оформить заказ' button and the footer with contact information.

Рисунок 44 - Форма для данных карты

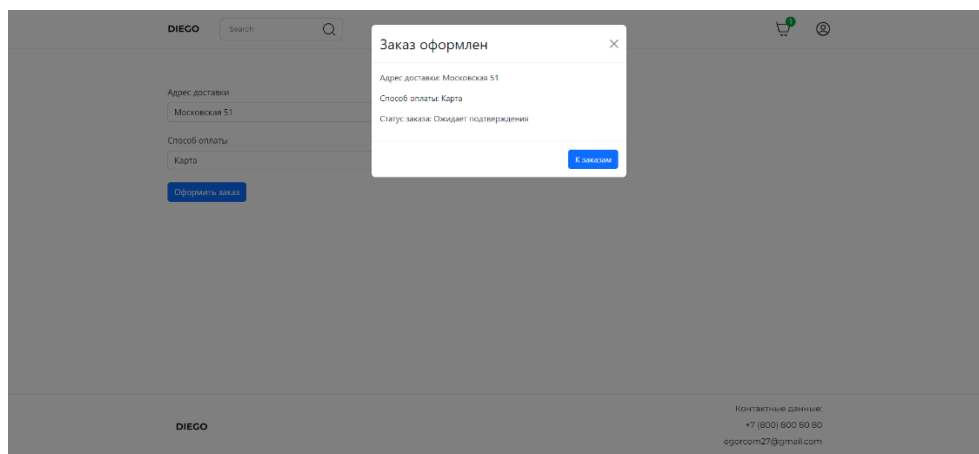


Рисунок 45 - Сообщение об успешности операции

4.5.2 Калькулятор калорий

Кроме возможности оформления заказа сайт позволяет пользователям посчитать количество упаковок товара при вводе необходимого количества калорий. Это позволит пользователям эффективно совершать покупки без лишних затрат, а также поможет вести верный подсчет съедаемых калорий в течение дня.

Для использования этой функциональности сайта пользователю нужно перейти на страницу товара (Рисунок 36) и нажать на кнопку «Калькулятор калорий». После этого появится поле для ввода количества калорий (Рисунок 46), и будет произведен автоматический подсчет количества товара, содержащего заданное число калорий.

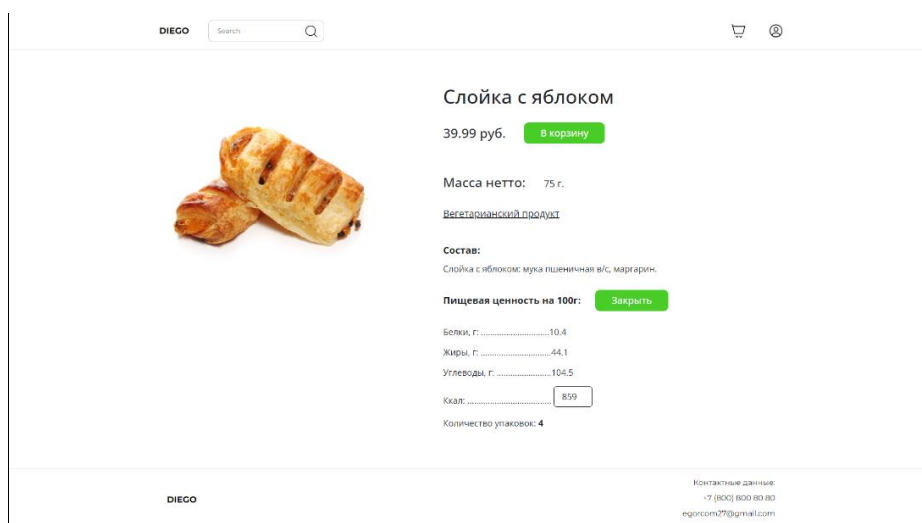


Рисунок 46 - Калькулятор калорий

4.5.3 Возможности сайта для вегетарианцев

Сайт оснащен функцией фильтрации продуктов для конкретной группы людей с системой питания, исключающей определенные типы пищи.

Это возможность для пользователей настраивать фильтр продуктов так, чтобы они видели лишь только те продукты, которые соответствуют их потребностям, в том числе диетическим.

Для этого на странице товаров (Рисунок 35) пользователь может выбрать поле «Да» в разделе «Для вегетарианцев». В результате будет произведена фильтрация товаров и на странице будут только те продукты, которые входят в рацион вегетарианцев (Рисунок 47).

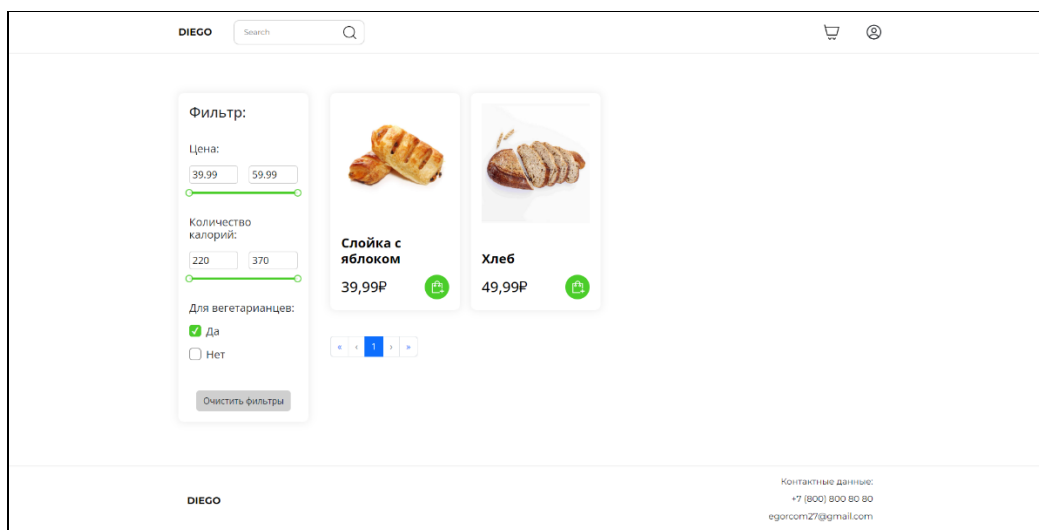


Рисунок 47 - Фильтр для вегетарианцев

4.5.4 Аккаунт пользователя

В возможности авторизованного пользователя входит просмотр аккаунта, где пользователь в разделе «Профиль» может ознакомиться со своими данными, удалить аккаунт или выйти из аккаунта (Рисунок 48), а также отредактировать данные о себе (Рисунок 49) и сменить пароль (Рисунок 50).

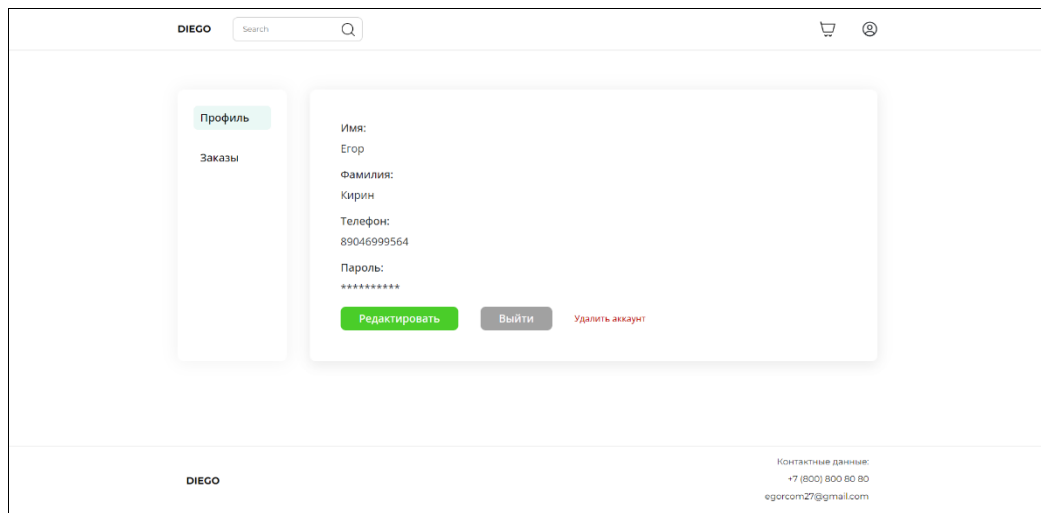


Рисунок 48 - Страница аккаунта

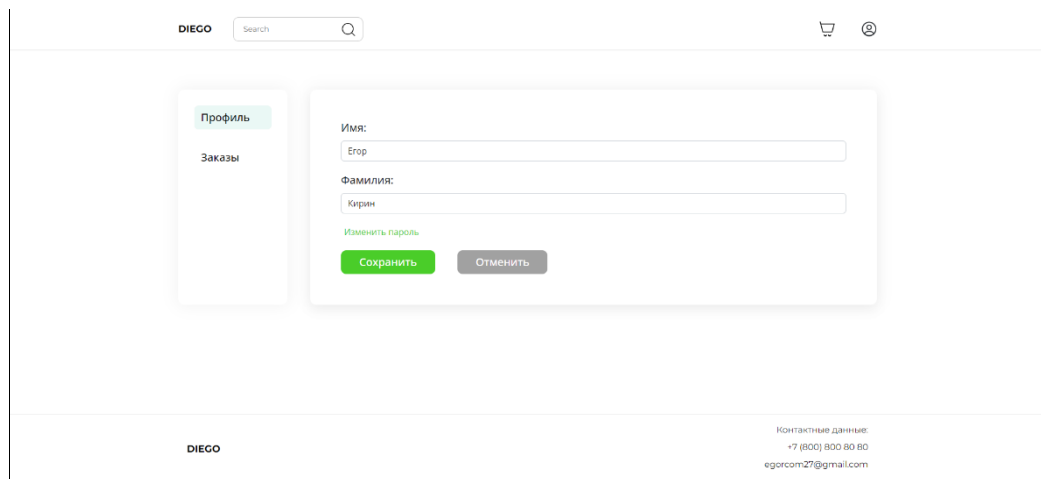


Рисунок 49 - Страница редактирования данных профиля

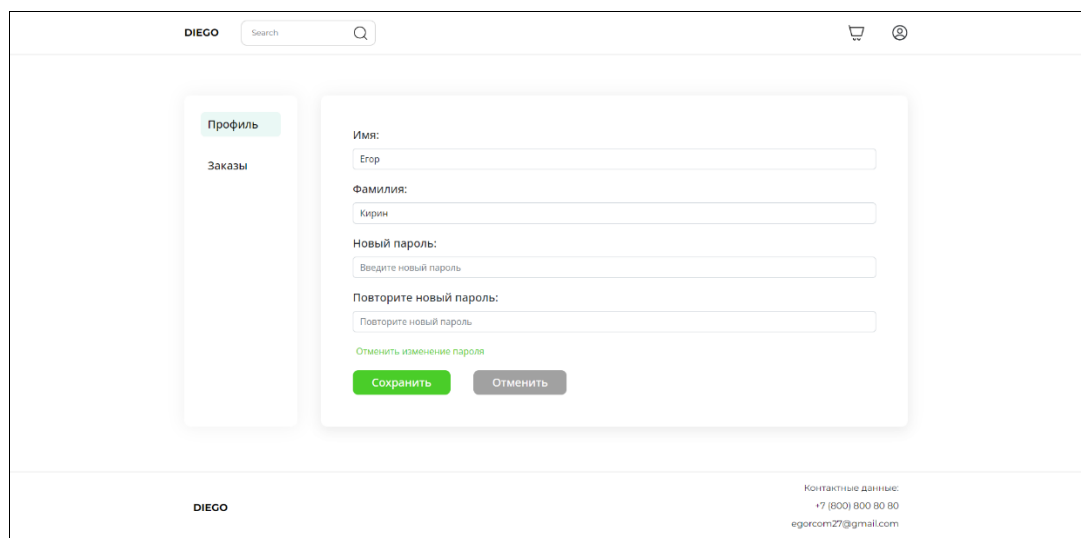


Рисунок 50 - Страница редактирования данных пароля и пароля

В разделе «Заказы» пользователь может ознакомиться с историей своих заказов (Рисунок 51), посмотреть статус заказа, товары конкретного заказа (Рисунок 52) и прочую информацию.

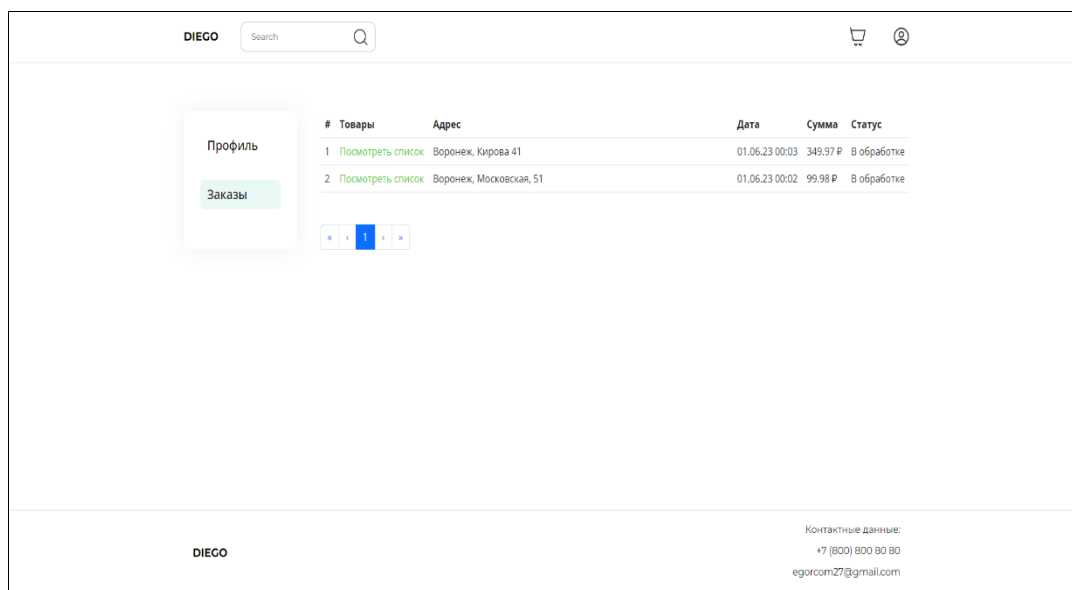


Рисунок 51 - Страница с историей заказов

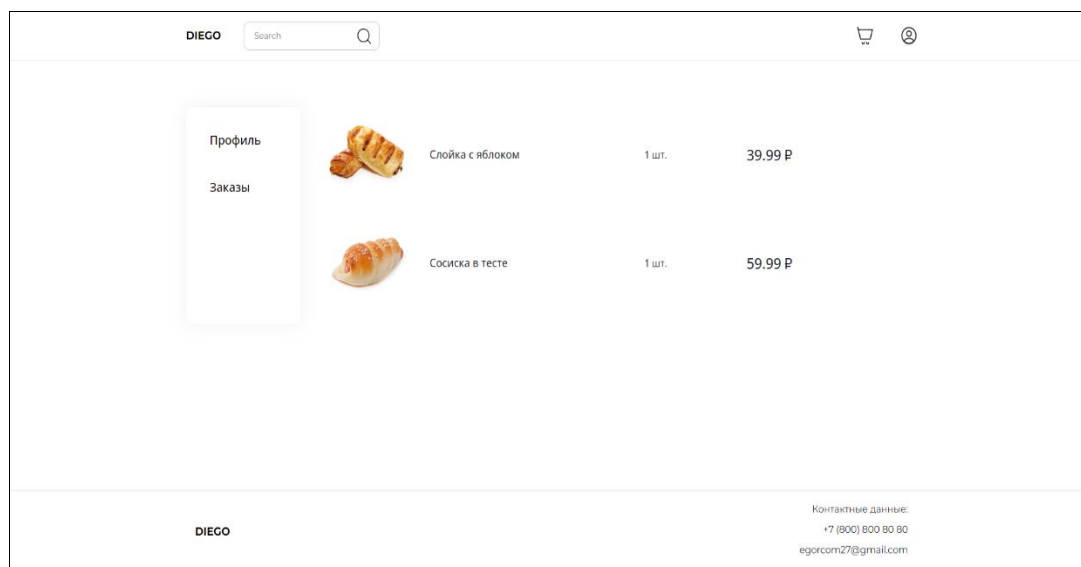


Рисунок 52 - Страница с товарами заказа

4.6 Для администратора

4.6.1 Добавление товаров

В возможности администратора входит функция добавления товаров. Это можно сделать на странице товаров при нажатии на кнопку «Добавить товар» (Рисунок 53). После нажатия открывается страница с полями для ввода

данных о новом товаре (Рисунок 54). По окончании ввода данных администратору нужно нажать на кнопку «Добавить», и товар добавится в список имеющихся.

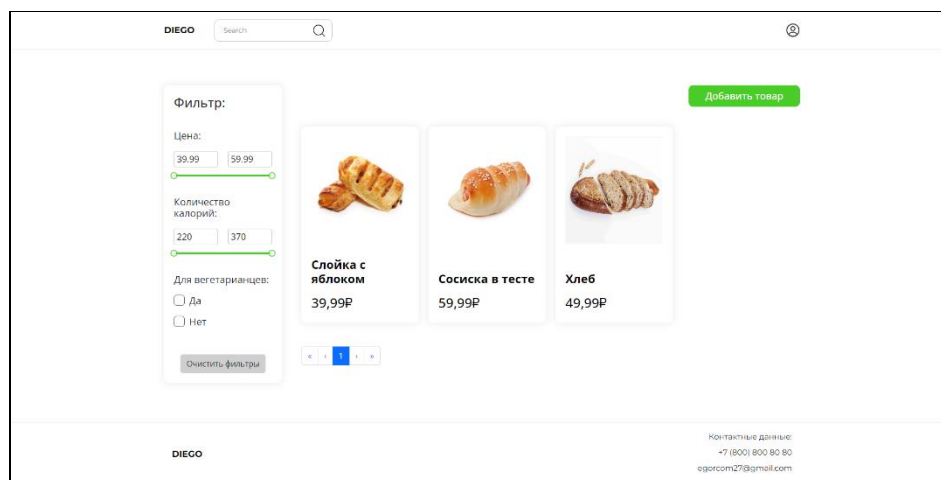


Рисунок 53 - Страница товаров для администратора

Рисунок 54 - Страница добавления товара

4.6.2 Редактирование и удаление товаров

К функциям, доступным администратору, относятся редактирование и удаление товаров.

На странице товара администратор при нажатии на кнопку «Удалить» может удалить товар из имеющихся, а при нажатии на кнопку «Редактировать»

товар» (Рисунок 55) открывается страница для редактирования товара, на которой администратор может изменить данные о товаре (Рисунок 56).

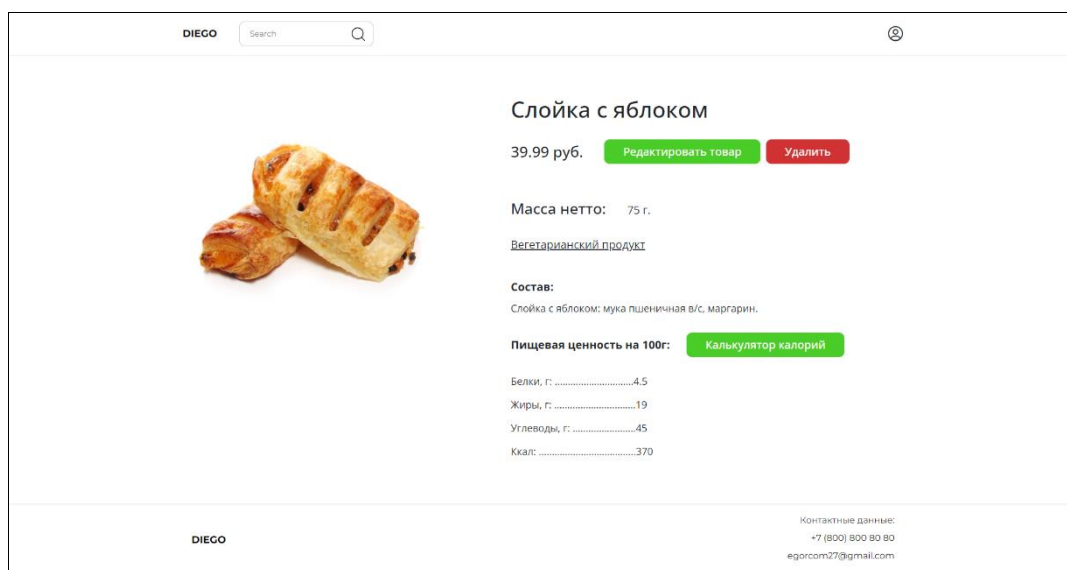


Рисунок 55 - Страница товара для администратора

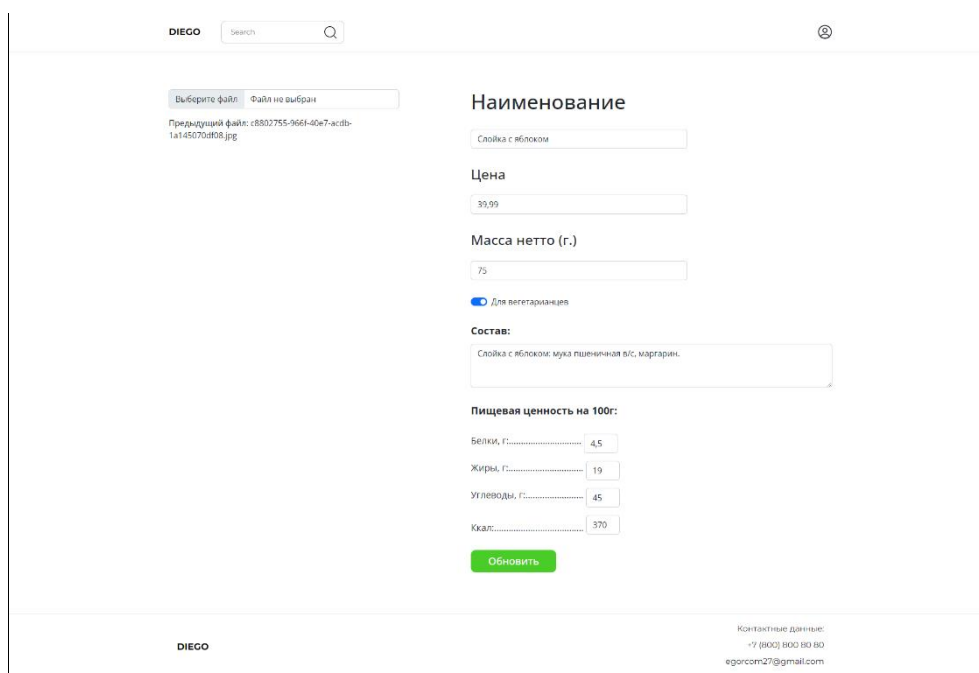


Рисунок 56 - Страница редактирования товара

4.6.3 Изменение ролей пользователей и их удаление

В возможности администратора входит функция изменения ролей пользователей и их удаление.

На странице со списком пользователей (Рисунок 57) администратор может изменять роли пользователей системы, а также удалять их.

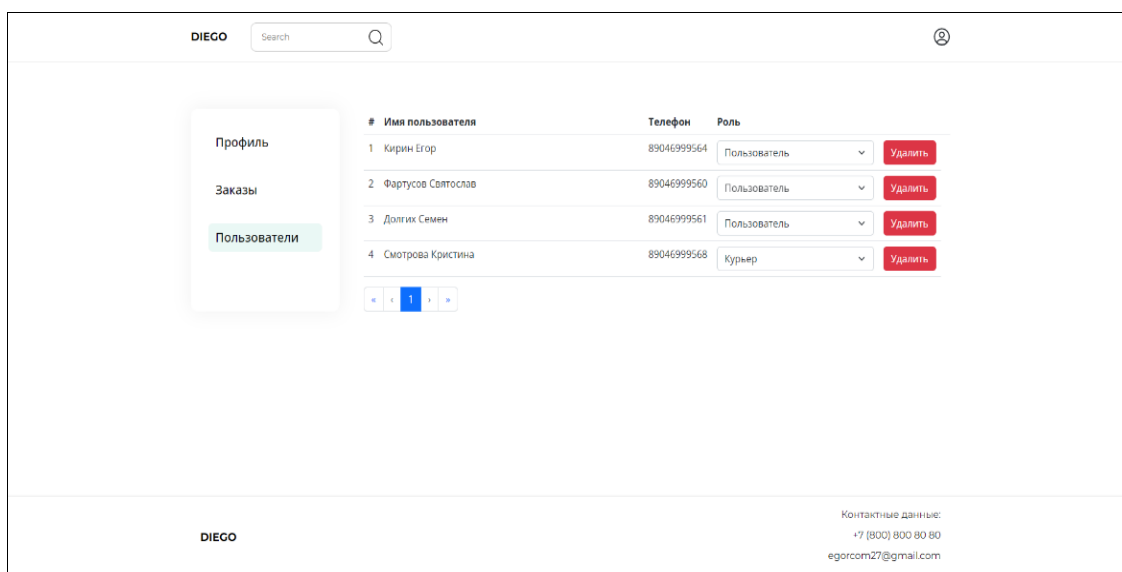


Рисунок 57 - Страница со списком пользователей

4.6.4 Управление заказами

Администратор может управлять заказами на странице аккаунта в разделе «Заказы». На этой странице администратор может просмотреть список товаров в конкретном заказе (Рисунок 52), распределить заказы между курьерами, а также отменить выполнение заказа, определенного конкретному курьеру (Рисунок 58).

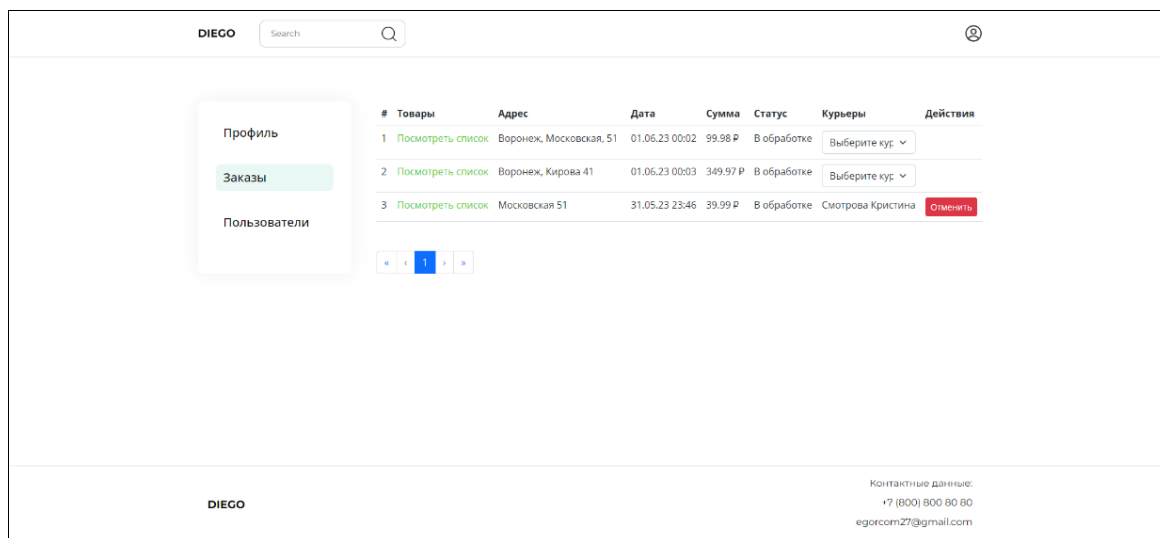


Рисунок 58 - Страница управления заказами

4.7 Для курьера

4.7.1 Доставка заказов

В возможности курьера входит функция выбора заказа для доставки. Для ее осуществления курьер заходит на страницу аккаунта в раздел заказы (Рисунок 59), где выбирает заказ для доставки, нажав на кнопку «Взять». На этой же странице курьер может ознакомиться со списком товаров в конкретном заказе (Рисунок 52).

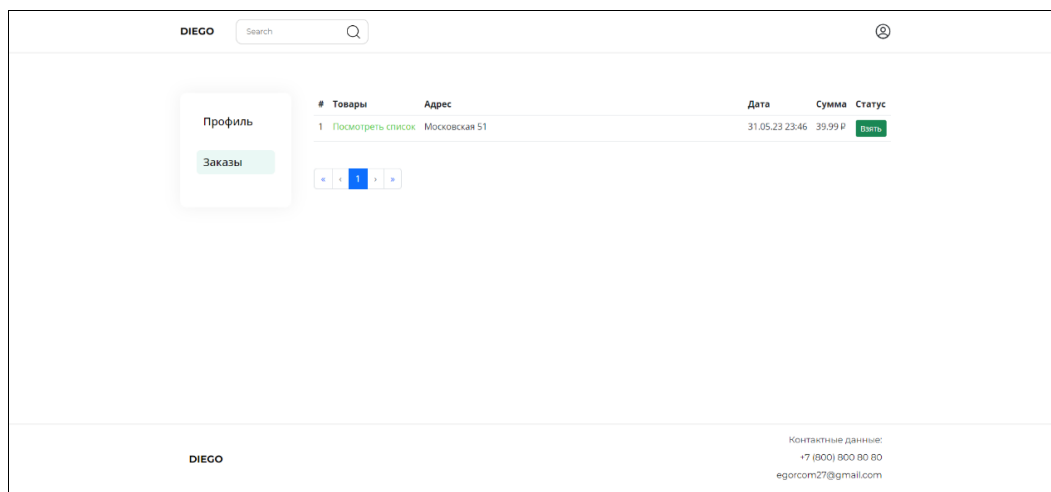


Рисунок 59 - Страница списка заказов

После нажатия на кнопку «Взять» курьеру открывается страница с информацией о заказе (Рисунок 60). После выполнения доставки заказа на этой же странице курьеру необходимо нажать на кнопку «Завершить заказ», тогда заказу присваивается статус «Завершен» (Рисунок 61).

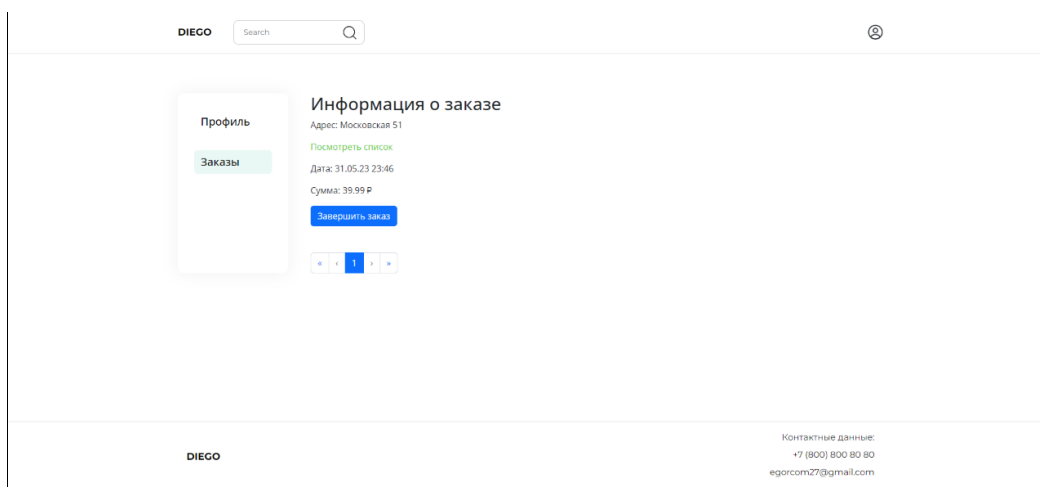


Рисунок 60 - Страница с информацией о заказе

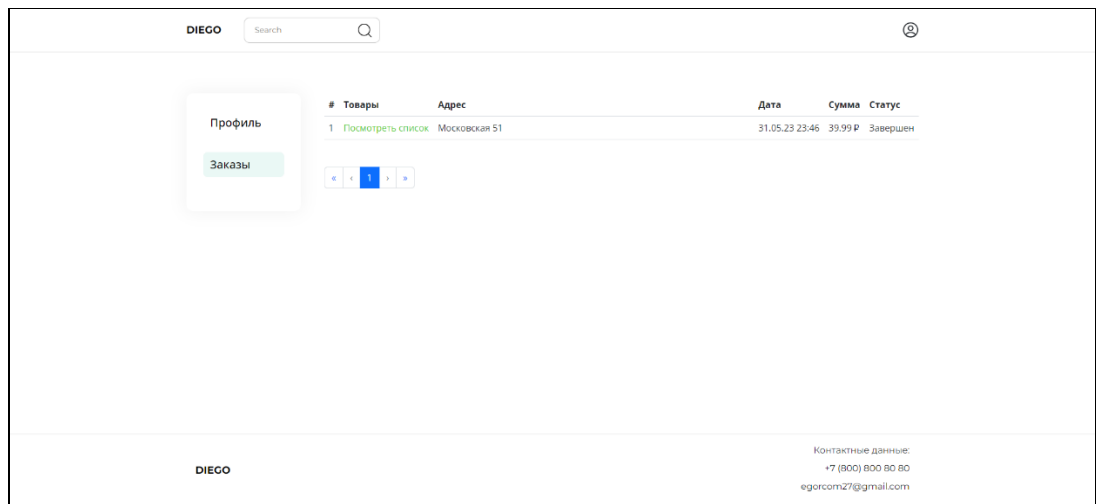


Рисунок 61 - Страница с информацией о завершении заказа

5 Тестирование

Для проверки работоспособности разработанного приложения было проведено тестирование системы с использованием следующего типа тестов:

- UI-тестирование (тестирование пользовательского интерфейса);
- Smoke testing (дымовое тестирование);
- Usability-тестирование;
- Unit-тесты (модульное тестирование).

5.1 UI-тестирование

Тестирование пользовательского интерфейса (UI-тестирование) - это процесс проверки графического интерфейса приложения или веб-сайта на соответствие требованиям, удобство использования, визуальное оформление и правильное функционирование элементов пользовательского интерфейса.

Основная цель UI-тестирования заключается в убеждении, что пользовательский интерфейс выглядит и работает так, как ожидается, и что пользователи могут легко взаимодействовать с приложением или веб-сайтом [9].

В таблице 1 продемонстрированы результаты тестирования пользовательского интерфейса для неавторизованного пользователя.

Таблица 1 - Результаты UI-тестирования для неавторизованного пользователя

Тестовый сценарий	Ожидаемый результат	Статус теста
Нажатие на одну из категорий на главной странице	Переход на страницу товаров выбранной категории	Пройден
Нажатие на логотип	Переход на главную страницу	Пройден
Нажатие на кнопку «Корзина» в шапке сайта	Переход на страницу корзины	Пройден

Нажатие на кнопку «Аккаунт» в шапке сайта	Переход на страницу авторизации	Пройден
Нажатие на кнопку «Войти» на странице авторизации с незаполненными полями	Вывод предупреждения о необходимости заполнения полей	Пройден
Нажатие на ссылку «Зарегистрироваться» на странице авторизации	Переход на страницу регистрации	Пройден
Нажатие на кнопку «Зарегистрироваться» на странице регистрации с незаполненными полями	Вывод предупреждения о необходимости заполнения полей	Пройден
Нажатие на ссылку «Войти» на странице регистрации	Переход на страницу авторизации	Пройден
Нажатие на ссылку «Восстановить» на странице авторизации	Переход на страницу восстановления пароля	Пройден
Нажатие на кнопку «Проверить» на странице восстановления пароля для проверки данных с незаполненными полями	Вывод предупреждения о необходимости заполнения полей	Пройден
Нажатие на кнопку «Сохранить» на странице восстановления пароля с незаполненными полями	Вывод предупреждения о необходимости заполнения полей	Пройден
Ввод названия товара в строку поиска	Переход на страницу с найденными товарами	Пройден

Нажатие на кнопку добавления товара в корзину на странице товаров	Добавление товара в корзину с отображением числа товаров на корзине в шапке сайта	Пройден
Указать в фильтре товаров желаемый размер цен на странице товаров	Вывод товаров, подходящих под заданный фильтр	Пройден
Указать в фильтре товаров желаемое количество калорий на странице товаров	Вывод товаров, подходящих под заданный фильтр	Пройден
Выбрать один из вариантов «Для вегетарианцев» в фильтре товаров на странице товаров	Вывод товаров, подходящих под заданный фильтр	Пройден
Нажатие на кнопку «Очистить фильтры» на странице товаров	Вывод всех товаров выбранной категории	Пройден
Нажатие на кнопку «В корзину» на странице товара	Добавление товара в корзину с отображением числа товаров на корзине в шапке сайта	Пройден
Нажатие на кнопку «Калькулятор калорий» на странице товара	Отображение на страницы товара поля для ввода калорий	Пройден
Ввод количества калорий в поле на странице товара, появившееся после нажатия на кнопку «Калькулятор калорий»	Отображение числа упаковок, содержащих заданное количество калорий	Пройден
Нажатие на кнопку «Закрыть» на странице товара после	Отображение исходной страницы товара	Пройден

использования калькулятора калорий		
Нажатие на ссылку «Вернуться к покупкам» на странице корзины	Переход на главную страницу	Пройден
Нажатие на ссылку «Очистить корзину» на странице корзины	Страница корзины пустая	Пройден
Нажатие на кнопки «+» или «-» на странице корзины у добавленного товара	Увеличение или уменьшение количества добавленного товара	Пройден
Нажатие на кнопку «⊗» на странице корзины у добавленного товара	Удаление выбранного товара	Пройден
Нажатие на кнопку «Продолжить оформление» на странице корзины	Появление формы с выбором кнопок «Зарегистрироваться» и «Авторизоваться»	Пройден
Попытка входа в аккаунт с одним из незаполненных полей	Вывод предупреждения о необходимости заполнения полей	Пройден
Попытка регистрации с одним из незаполненных полей	Вывод предупреждения о необходимости заполнения полей	Пройден

5.2 Дымовое тестирование

Дымовое тестирование (smoke testing) - это одна из методик функционального тестирования программного обеспечения, направленная на быструю проверку основных функций или основных компонентов приложения с целью выявления критических проблем.

Целью дымового тестирования является убедиться, что основные функциональности приложения работоспособны, а другие компоненты системы могут быть интегрированы и протестированы более подробно в дальнейшем [10].

В таблицах 2, 3, 4 продемонстрированы результаты дымового тестирования для сценариев пользователя, администратора, курьера.

Таблица 2 - Результаты дымового тестирования для пользователя

Тестовый сценарий	Результат теста
Регистрация	Пройден
Авторизация	Пройден
Восстановление пароля	Пройден
Просмотр категорий товаров	Пройден
Просмотр списка товаров выбранной категории	Пройден
Фильтрация товаров	Пройден
Использование калькулятора калорий	Пройден
Использование строки поиска товаров	Пройден
Добавление товара в корзину	Пройден
Просмотр корзины	Пройден
Очистка корзины	Пройден
Увеличение/уменьшение числа товара в корзине	Пройден
Удаление товара из корзины	Пройден
Оформление заказа	Пройден
Просмотр профиля	Пройден
Редактирование данных профиля	Пройден
Удаление профиля	Пройден
Выход из аккаунта	Пройден
Получение информации о списке заказов в профиле	Пройден

Получение информации в профиле о списке товаров в заказе	Пройден
--	---------

Таблица 3 - Результаты дымового тестирования для администратора

Тестовый сценарий	Результат теста
Авторизация	Пройден
Просмотр списка товаров	Пройден
Фильтрация товаров	Пройден
Использование калькулятора калорий	Пройден
Добавление товара	Пройден
Удаление товара	Пройден
Редактирование данных товара	Пройден
Просмотр списка категорий	Пройден
Просмотр списка пользователей	Пройден
Изменение ролей пользователей	Пройден
Удаление пользователя	Пройден
Просмотр списка заказов	Пройден
Просмотр списка товаров в заказе	Пройден
Распределение заказов между курьерами	Пройден
Отмена выполнения доставки заказа курьером	Пройден
Просмотр профиля	Пройден
Редактирование данных профиля	Пройден
Удаление профиля	Пройден
Выход из аккаунта	Пройден

Таблица 4 - Результаты дымового тестирования для курьера

Тестовый сценарий	Результат теста
Авторизация	Пройден
Просмотр списка заказов	Пройден

Просмотр списка товаров в заказе	Пройден
Взятие заказа на выполнение доставки	Пройден
Завершение доставки заказа	Пройден
Просмотр профиля	Пройден
Редактирование данных профиля	Пройден
Удаление профиля	Пройден
Выход из аккаунта	Пройден

5.3 Usability-тестирование

Usability-тестирование — это процесс оценки и измерения, насколько легко и удобно пользователи могут взаимодействовать с приложением, веб-сайтом или продуктом, а также насколько эффективно они могут достигать своих целей при использовании данного продукта [11].

В таблице 5 продемонстрированы результаты usability-тестирования для 4 респондентов, перед которыми был поставлен ряд задач.

Оценка успешности выполнения задания будет оцениваться из ряда трех показателей:

- В случае успешного выполнения задачи — хорошо;
- В случае если у респондента возникли проблемы, но он смог выполнить задачу — удовлетворительно;
- В случае не выполнения задачи — плохо.

Таблица 5 - Результаты usability-тестирования

Задание	Респондент 1	Респондент 2	Респондент 3	Респондент 4
Регистрация	хорошо	хорошо	хорошо	хорошо
Авторизация	хорошо	хорошо	хорошо	хорошо
Восстановление пароля	хорошо	хорошо	хорошо	хорошо

Просмотр категорий товаров	хорошо	хорошо	хорошо	хорошо
Просмотр списка товаров выбранной категории	хорошо	хорошо	хорошо	хорошо
Фильтрация товаров	хорошо	хорошо	хорошо	хорошо
Использование калькулятора калорий	хорошо	хорошо	хорошо	хорошо
Использование строки поиска товаров	хорошо	удовлетворительно	удовлетворительно	хорошо
Добавление товара в корзину	хорошо	хорошо	хорошо	удовлетворительно
Просмотр корзины	хорошо	хорошо	хорошо	хорошо
Очистка корзины	хорошо	хорошо	хорошо	хорошо
Увеличение/уменьшение числа товара в корзине	хорошо	хорошо	хорошо	хорошо
Удаление товара из корзины	хорошо	хорошо	хорошо	хорошо
Оформление заказа	хорошо	хорошо	хорошо	хорошо
Просмотр профиля	хорошо	хорошо	хорошо	хорошо
Редактирование данных профиля	хорошо	хорошо	хорошо	хорошо
Удаление профиля	хорошо	хорошо	хорошо	хорошо
Выход из аккаунта	хорошо	хорошо	хорошо	хорошо

Получение информации о списке заказов в профиле	удовлетворительно	хорошо	хорошо	хорошо
Получение информации в профиле о списке товаров в заказе	хорошо	хорошо	хорошо	хорошо

После проведенного опроса мнения по некоторым функциональностям сайта разделились. Большинство из них получили оценку «хорошо» от респондентов, например, возможность регистрации была успешно пройдена всеми без проблем (Рисунок 62).

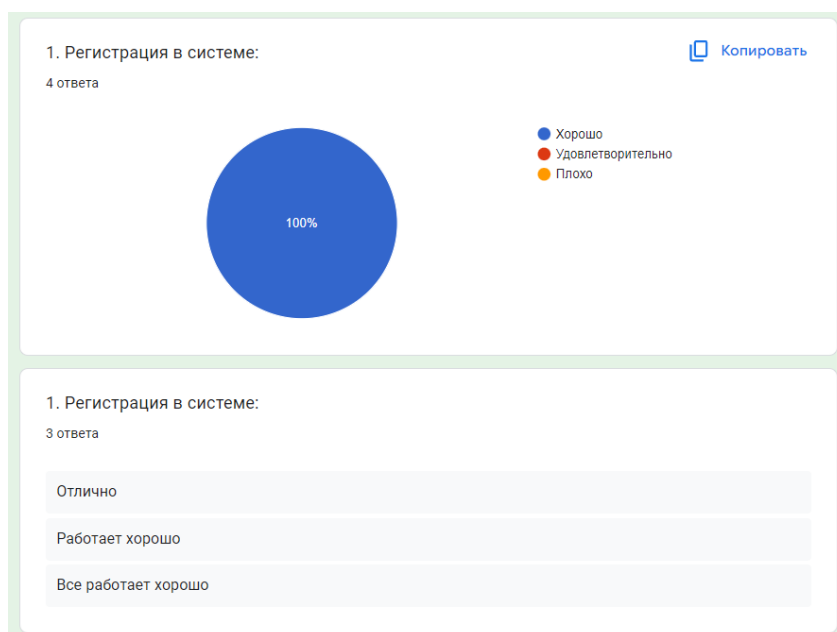


Рисунок 62 - Отзыв по возможности регистрации

Но было несколько функциональностей, мнения на которых разделились. Одной из них является использование строки поиска товаров. Неудобство заключается в том, что поиск сделан по точному совпадению слова с отсутствием других его форм (Рисунок 63).



Рисунок 63 - Отзыв по возможности использования строки поиска товаров

Еще одной функциональностью, где мнения разделились, является просмотр списка заказов на странице аккаунта. Одному из респондентов не понравилось отсутствие информации о курьере, выполняющем доставку его заказа (Рисунок 64).

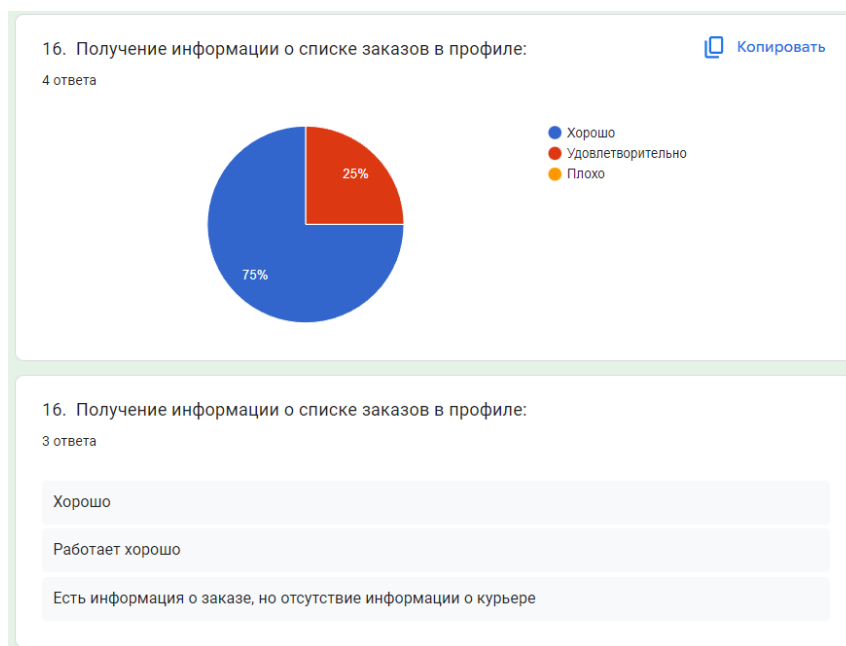


Рисунок 64 - Отзыв по возможности просмотра списка заказов на странице аккаунта

Кроме этого было высказано мнение по поводу функциональности добавления товаров в корзину (Рисунок 65), а также по поводу возможности оформления заказа (Рисунок 66).

8. Добавление товара в корзину:

Можно сделать анимацию перемещения (или что-то похожее), чтобы было нагляднее

Рисунок 65 - Отзыв по возможности добавления товаров в корзину

13. Оформление заказа:

Желательно добавить форму добавления адреса по нескольким полям (улица дом подъезд квартира)

Рисунок 66 - Отзыв по возможности оформления заказа

В остальных функциональностях вопросов и проблем не возникло. Респонденты справились со всеми поставленными задачами. Полученные результаты являются возможными путям дальнейшего развития и доработки проекта.

5.4 Unit-тесты

Модульное тестирование, также известное как unit-тестирование, представляет собой метод тестирования программного обеспечения, в котором отдельные модули или компоненты программы проверяются независимо от остальной системы. Модуль в данном контексте обычно является наименьшей тестируемой единицей, например, функцией, методом или классом.

Основная цель модульного тестирования — убедиться, что каждый модуль работает правильно в изоляции от других модулей и в соответствии с ожидаемым поведением. Это позволяет выявить и исправить ошибки на ранних стадиях разработки, обеспечивая более надежную и стабильную систему в целом.

На рисунках 67, 68 отображены результаты unit-тестов серверной части приложения. Все тесты были успешно пройдены системой. Никаких неполадок не было выявлено.


```

Category Controller
Server started on port 8080
  getAll
    ✓ should return all categories
  getOne
    ✓ should return a specific category by id

Order Controller
  create
    ✓ should create a new order with order products
  getAll
    ✓ should return paginated list of orders with default limit and page
    ✓ should return paginated list of orders with custom limit and page
  getOne
    ✓ should return orders for the authenticated user
    ✓ should return 404 if user is not found
    ✓ should return 401 if user is not authorized to access the orders
  getOrdersByCourierId
    ✓ should return orders for the authenticated courier
    ✓ should return 401 if user is not authorized as a courier
  updateCourier
    ✓ should update the courier for the order
    ✓ should return 404 if the order is not found
    ✓ should return 404 if the courier is not found
  delete
    ✓ should delete the order and associated products
    ✓ should return 404 if the order is not found
  getProductsByOrderId
    ✓ should return the products for the authorized user
    ✓ should return 404 if the order is not found
    ✓ should return 401 if the user is not authorized to access the order

Product Controller
  getProductsByCategory, search
    ✓ should retrieve products by category without filters
    ✓ should search products with filters
  getOne
    ✓ should retrieve a single product by name
    ✓ should return 404 if product is not found
  delete
    ✓ should delete a product by name
  create
    ✓ should create a new product
    ✓ should return 400 if required fields are missing

```

Рисунок 67 - Результаты unit-тестов

```

User Controller
  Registration Controller
    ✓ should register a new user
    ✓ should return an error if password confirmation does not match
    ✓ should return an error if phone number is already registered
  Login Controller
    ✓ should log in an existing user with correct credentials
    ✓ should return 404 if user with given phone number does not exist
    ✓ should return 400 if password is incorrect
  Verify Account Controller
    ✓ should verify account with correct phone number and secret word
    ✓ should return 404 if user with given phone number does not exist
    ✓ should return 400 if secret word is empty
    ✓ should return 400 if secret word is incorrect
    ✓ should reset password with valid token and matching passwords
    ✓ should return error if token is missing
    ✓ should return error if passwords do not match
  getAll
    ✓ should get all users with default limit and page
    ✓ should get all users with custom limit and page
  getAllCourier
    ✓ should get all courier
  getOne
    ✓ should get one user
    ✓ should return 404 if user with given phone number does not exist
  deleteUserForAdmin
    ✓ should delete user with valid token and matching user ID
    ✓ should return 404 if user with the given phone number doesn't exist

45 passing (1s)

```

Рисунок 68 - Результаты unit-тестов

6 Анализ веб-приложения

Главной функциональностью приложения является оформление заказов. Поэтому с использованием сервиса Яндекс.Метрика [12] была добавлена продуктовая воронка по оформлению заказа (Рисунок 69). Она демонстрирует заинтересованность пользователей в возможности оформления заказа и отражает статистику переходов по страницам.

Данная воронка отражает количество пользователей, зашедших на страницу корзины, а именно 18, а среди продолживших оформление насчитывается 12, что составляет 67% от общего числа. К шагу завершения оформления перешли 11 человек, то есть 92% от тех, кто перешел на страницу оформления.

Такой результат позволяет сделать вывод, что большинство посетителей сайта доходит до завершения оформления заказа и проявляет интерес к приобретению товаров, но есть и небольшое число тех, кто покидал сайт, не завершив оформление. Поэтому даже небольшие проценты стимулируют к доработке и усовершенствованию возможностей сайта.



Рисунок 69 - Продуктовая воронка для возможности оформления заказа

Кроме продуктовой воронки на рисунке 70 отражена статистика взаимодействия пользователей с различной функциональностью сайта. Например, калькулятор калорий, как и фильтрация для вегетарианцев являются востребованными у большинства посетителей сайта.

«Клик на корзину» и «Клик на профиль» отображают положительную динамику в использовании этих возможностей пользователями. Можно сделать вывод, что посетители сайта стремятся к оформлению заказов и после их осуществления просматривают информацию о заказе и проверяют его статус.

А «Клик по почте в контактах» и «Клик по номеру телефона в контактах» не являются популярными среди посетителей. Можно сделать вывод, что они не нуждаются в связи с компанией, и доступность сайта не побуждает их задавать вопросы о работе сайта.

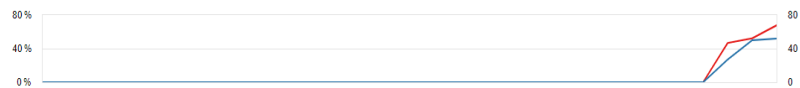
1. Клик на ссылку логотипа ⓘ

Конверсия 51,6 %
Достижения цели 163
Целевые визиты 49



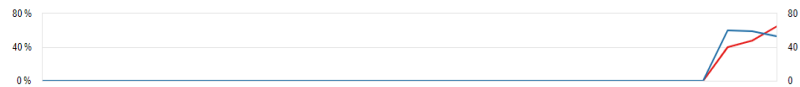
2. Клик на корзину ⓘ

Конверсия 53,7 %
Достижения цели 129
Целевые визиты 51



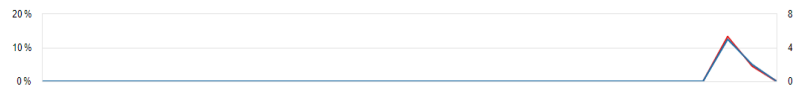
3. Клик на профиль ⓘ

Конверсия 49,5 %
Достижения цели 172
Целевые визиты 47



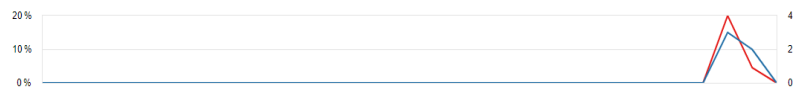
4. Клик по почте в контактах ⓘ

Конверсия 4,21 %
Достижения цели 7
Целевые визиты 4



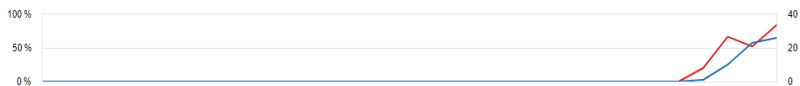
5. Клик по номеру телефона в контактах ⓘ

Конверсия 5,26 %
Достижения цели 5
Целевые визиты 5



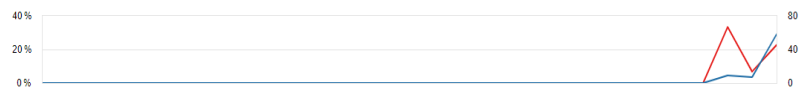
6. Просмотр за один визит ⓘ

Конверсия 63,2 %
Достижения цели 60
Целевые визиты 60



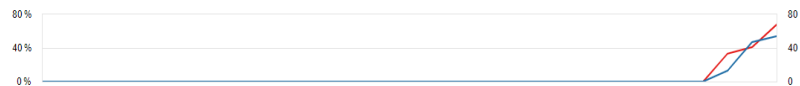
7. Клик по калькулятору калорий ⓘ

Конверсия 15,8 %
Достижения цели 74
Целевые визиты 15



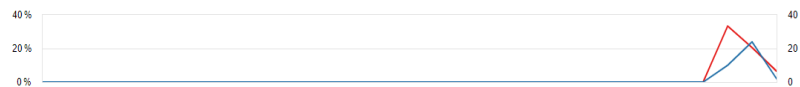
8. Клик по кнопке в Корзину на странице всех товаров ⓘ

Конверсия 46,3 %
Достижения цели 114
Целевые визиты 44



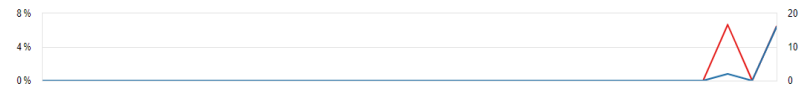
9. Клик по кнопке в Корзину на странице товара ⓘ

Конверсия 16,8 %
Достижения цели 36
Целевые визиты 16



10. Клик по кнопке Да для вегетарианцев в фильтре ⓘ

Конверсия 3,16 %
Достижения цели 18
Целевые визиты 3



11. Клик по кнопке Нет для вегетарианцев в фильтре ⓘ

Конверсия 2,11 %
Достижения цели 4
Целевые визиты 2

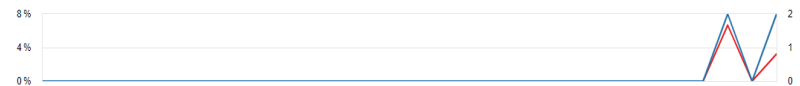


Рисунок 70 - Конверсии на сервисе Яндекс.Метрика

Заключение

В ходе данной курсовой работы были выполнены все поставленные задачи. Была реализована функциональность оформления заказов с доставкой по указанному адресу, расчета числа упаковок товара на основании введенного количества калорий, а также реализована продажа товаров для конкретной группы людей с системой питания, исключающей определенные типы пищи.

В работе использовался ряд технологий, который позволил реализовать сайт продовольственного онлайн-супермаркета с доставкой на дом. Эти действия привели к выполнению цели данного курсового проекта.

В заключение следует отметить, что разработанный функциональный прототип продовольственного онлайн-супермаркета с доставкой на дом предоставляет пользователям возможность осуществлять покупки продуктов онлайн и получать их по указанному адресу. В дальнейшем, на основе разработанного прототипа, можно продолжить развитие и расширение функциональностей, учитывая отзывы пользователей и растущие требования рынка.

Список использованной литературы

1. Что такое сайт [Электронный ресурс]. — Режим доступа: https://www.alkosto.ru/blog/chto-takoe-sajt/?utm_referrer=https%3A%2F%2Fwww.google.com%2F. — Заглавие с экрана. — (Дата обращения: 11.03.2023).
2. METRO [Электронный ресурс]. — Режим доступа: <https://www.metro-ss.ru/>. — Заглавие с экрана. — (Дата обращения: 11.03.2023).
3. Лента [Электронный ресурс]. — Режим доступа: https://lenta.com/?utm_referrer=https%3A%2F%2Fwww.google.com%2F. — Заглавие с экрана. — (Дата обращения: 11.03.2023).
4. Использование диаграммы вариантов использования UML при проектировании программного обеспечения [Электронный ресурс]. — Режим доступа: <https://habr.com/ru/articles/566218/>. — Заглавие с экрана. — (Дата обращения: 15.03.2023).
5. Простое руководство по диаграммам активности UML [Электронный ресурс]. — Режим доступа: <https://creatly.com/blog/ru/uncategorized/64435/>. — Заглавие с экрана. — (Дата обращения: 15.03.2023).
6. Что такое JavaScript? [Электронный ресурс]. — Режим доступа: https://developer.mozilla.org/ru/docs/Learn/JavaScript/First_steps/What_is_JavaScript. — Заглавие с экрана. — (Дата обращения: 16.03.2023).
7. Документация по веб-API ASP.NET Core с использованием Swagger (OpenAPI) [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/aspnet/core/tutorials/web-api-help-pages-using-swagger?view=aspnetcore-5.0>. — Заглавие с экрана. — (Дата обращения: 18.05.2023).
8. Общие сведения о React [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/windows/dev-environment/javascript/react-overview>. — Заглавие с экрана. — (Дата обращения: 20.05.2023).

9. Как проводить UI-тестирование мобильных и верстки + сравнений инструментов автоматизации [Электронный ресурс]. — Режим доступа: <https://ux-journal.ru/kak-provodit-ui-testirovanie-sravnenie-instrumentov.html#testcase-example>. — Заглавие с экрана. — (Дата обращения: 25.05.2023).
10. Smoke-тестирование [Электронный ресурс]. — Режим доступа: <https://blog.skillfactory.ru/glossary/smoke-test/>. — Заглавие с экрана. — (Дата обращения: 25.05.2023).
11. Юзабилити-тестирование: пошаговая инструкция на примере Yelp [Электронный ресурс]. — Режим доступа: <https://lpgenerator.ru/blog/2016/07/28/yuzabiliti-testirovanie-poshagovaya-instrukciya-na-primere-yelp/>. — Заглавие с экрана. — (Дата обращения: 25.05.2023).
12. Счетчики – Яндекс.Метрика [Электронный ресурс]. — Режим доступа: <https://metrika.yandex.ru/list?>. — Заглавие с экрана. — (Дата обращения: 04.06.2023).