

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВА-  
ТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук  
Кафедра программирования и информационных технологий

Отчёт по курсовому проекту  
продуктовый онлайн-магазин «GoodFood»

Исполнители \_\_\_\_\_ *Я.В. Лазарев,*  
*М.А. Мотенко,*  
*Н. А. Трофимов*  
Заказчик \_\_\_\_\_ *В.С. Тарасов*

Воронеж 2021

## Содержание

Введение.....	3
1 Постановка задачи.....	4
1.1 Цель .....	4
1.2 Сфера использования.....	4
1.3 Требования.....	4
1.4 Задачи .....	5
2 Анализ предметной области .....	5
2.1 Целевая аудитория .....	5
2.2 Пользователи системы.....	6
2.3 Анализ существующих решений.....	7
2.4 Анализ задачи.....	11
2.4.1 Варианты использования приложения .....	11
2.4.2 Взаимодействие компонентов системы.....	13
2.4.3 Варианты состояния системы .....	16
2.4.4 Варианты действий в системе.....	17
2.4.5 Развертывание приложения .....	19
2.4.6 Диаграмма классов.....	20
2.4.7 Диаграмма объектов .....	20
2.4.8 IDEF0.....	21
3 Стек используемых технологий.....	22

## **Введение**

На данный момент всё больше людей предпочитают пользоваться услугами online супермаркетов. Их главным преимуществом является возможность выбирать, оплачивать и получать товары, не выходя из дома. Именно для реализации этих целей и создаётся сайт «GoodFood».

Также большой популярностью пользуется система правильного питания, позволяющая людям худеть и следить за своим здоровьем, подсчитывая количество калорий в съеденной пищи. Поэтому, помимо экономии времени на поход в магазин, наш сервис должен помогать людям, придерживающимся правил здорового питания, контролировать количество калорий в покупаемых продуктах.

## **1 Постановка задачи**

Проект «GoodFood» является продуктовым онлайн-магазином, предназначенным для пользователей, желающих приобрести продукты, не выходя из дома.

### **1.1 Цель**

Создать сервис, который предоставит всем желающим:

- Возможность заказать продукты онлайн;
- Простую навигацию по сайту;
- Помощь в выборе продуктов с помощью рекомендательной системы;
- Возможность подсчета суточной нормы калорий каждого зарегистрированного пользователя;
- Возможность регулирования количества калорий в заказе каждого зарегистрированного пользователя.

### **1.2 Сфера использования**

Повседневная жизнь.

### **1.3 Требования**

Разрабатываемый сайт должен предоставлять пользователям следующие возможности:

1. Регистрация и авторизация;
2. Редактирование профиля пользователя;
3. Введения дополнительных личных данных, позволяющих сайту рассчитать суточную норму калорий;
4. Просмотр товаров и добавления в корзину с регулированием их количества;
5. Оформления заказа с оплатой на сайте;
6. Ознакомление со статьёй о правильном питании и перехода в режим регулирования калорий.

## **1.4 Задачи**

1. Провести анализ рынка с целью выявления достоинств и недостатков схожих по функционалу систем;
2. Спроектировать систему с учетом информации, полученной ранее в ходе анализа;
3. Разработать базу данных;
4. Разработать Back-end часть приложения;
5. Разработать Front-end часть приложения;
6. Создать связь между Front-end и Back-end частями приложения;
7. Создать профиль пользователя;
8. Создать профиль администратора;
9. Описать процесс разработки и результат.

## **2 Анализ предметной области**

### **2.1 Целевая аудитория**

Сайт продуктового магазина нацелен на всех людей. Человек любого возраста может просматривать страницы супермаркета, т.к. ни одна из них не будет содержать информацию с возрастными ограничениями. Разумеется, человек должен уметь работать с браузером и просматривать сайты. Согласно проведенному опросу, в котором участвовало 100 человек, 30% из них предпочитают заказывать продукты онлайн, ещё 25% хотели бы иметь такую возможность, но из-за недоступности сервисов в их городах воспользоваться такими услугами невозможно. Остальные 45% людей предпочитают посещать магазины лично.

Мы решили узнать, по какой причине 45% опрошенных не заказывают продукты онлайн, смотреть Рис.1

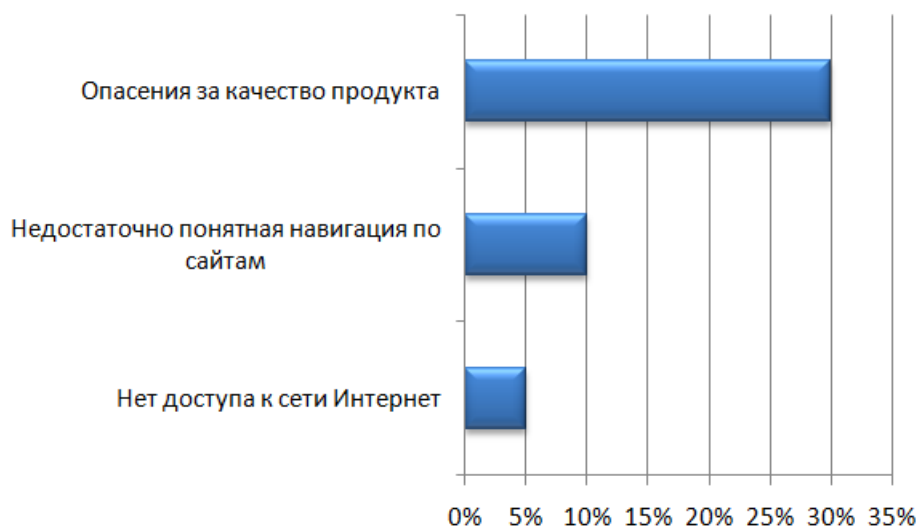


Рисунок 1 - Основные причины отказа от онлайн супермаркетов

Также стоит заметить, что 15% опрошенных придерживаются правил здорового питания. Они отметили, что после похода в магазин им приходится вводить в свои приложения для подсчёта калорий данные о купленных продуктах, чтобы просмотреть калории в них и рассчитать, какое количество допустимо для употребления.

Таким образом, наше веб-приложение ориентировано на большинство пользователей, предпочитающих заказывать продукты, не выходя из дома.

При этом мы ориентируем наш сайт на правильное питание, предоставляя пользователям возможность при желании воспользоваться функцией подсчёта калорий прямо на сайте, не прибегая к использованию дополнительных приложений.

## 2.2 Пользователи системы

Для взаимодействия с разрабатываемой системой выделяют следующие типы пользователей: гость, пользователь, администратор.

Гость не предоставляет системе никаких личных данных.

Пользователь, авторизованный в системе, сообщает ей при регистрации следующие параметры:

- Имя и фамилия пользователя;
- Электронная почта;
- Телефон;
- Пароль.

После перехода в систему подсчёта калорий будут добавлены следующие параметры:

- Вес;
- Пол;
- Возраст.

Администратор (авторизованный в системе работник магазина) обладает следующими параметрами:

- Логин;
- Пароль.

### **2.3 Анализ существующих решений**

Проведя анализ нескольких супермаркетов, мы выявили основные проблемы подобных сервисов. Основные недостатки:

- Нет возможности в любой момент перейти к другой категории товаров;
- Необходимо подниматься на верх страницы, чтобы выбрать другие продукты, смотреть Рис.2 (Сайт магазина <https://av.ru/>).

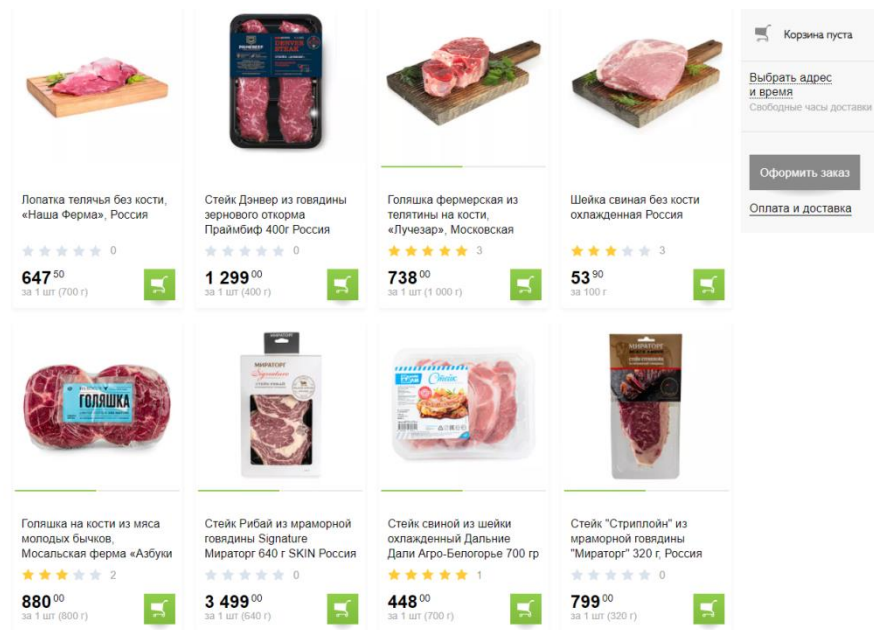


Рисунок 2 - Страница с товарами магазина «Азбука Вкуса»

— Нет возможности в фильтре задать диапазон максимальной и минимальной цены товаров, смотреть Рис.3 и Рис.4 (Сайты магазинов <https://www.auchan.ru/>, <https://vkusvill.ru/>).

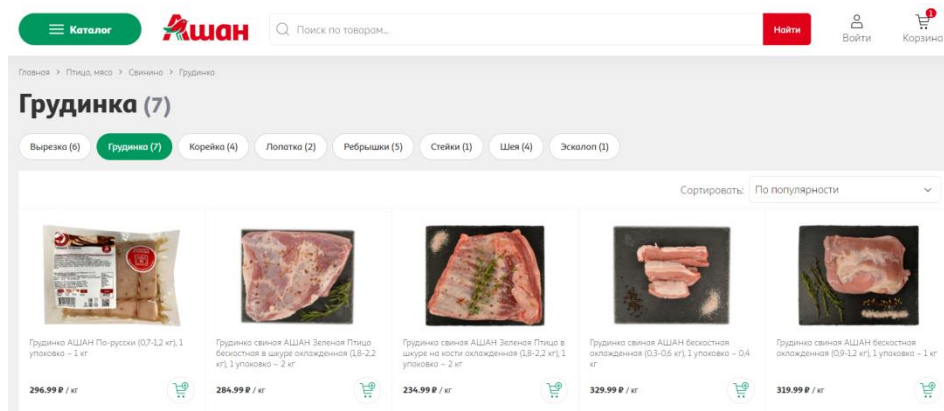


Рисунок 3 - Страница с товарами магазина «Ашан»



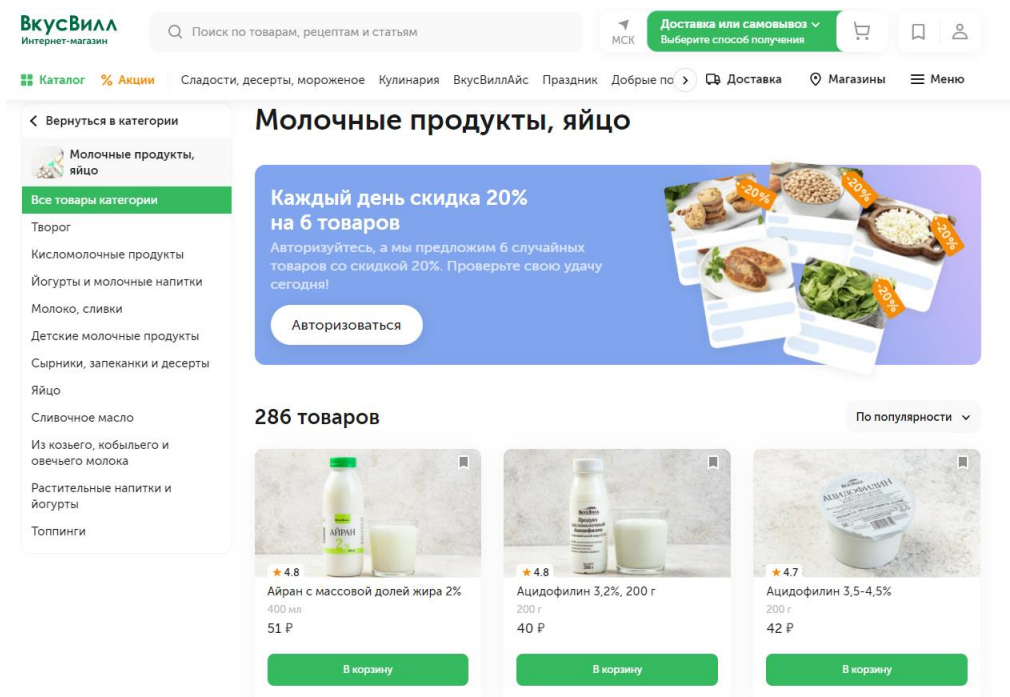


Рисунок 4 - Страница с товарами магазина «ВкусВилл»

- Есть каталоги с продуктами, информация для покупателей о магазине, но нет товаров. Невозможно определить цену продуктов по категориям. (Например, <https://www.okmarket.ru/>) После поиска в Интернете, выяснилось, что для товаров существует отдельный сайт (<https://www.okeydostavka.ru/msk/catalog>). Явных ссылок или кнопок на этот сервис на главном сайте не обнаружилось;
- Брошюры располагаются посреди каталога товаров, что может некоторым пользователям мешать во время выбора продуктов, смотреть Рис.5(Сайт магазина <https://lenta.com/>).

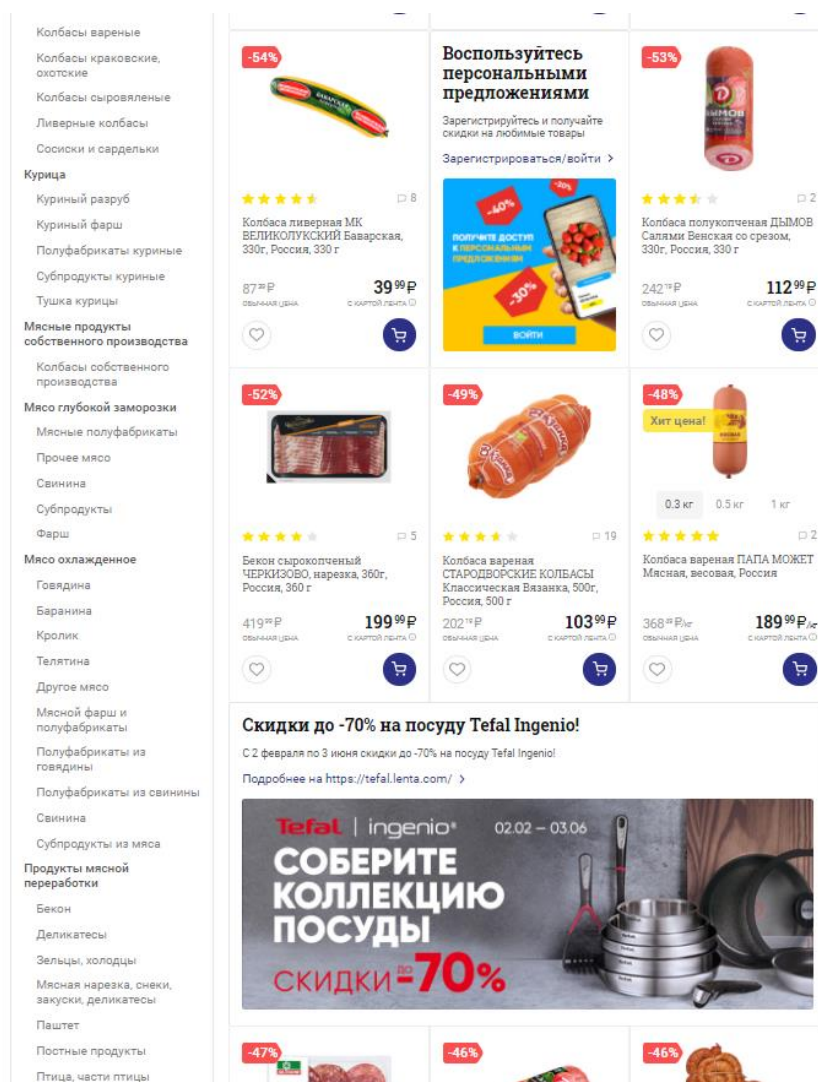


Рисунок 5 - Страница товаров магазина «Лента»

— В настоящее время популярные онлайн-супермаркеты явно не предоставляют возможность подсчитывать общее количество калорий в заказе. Существуют только отдельные сервисы, в которые пользователи могут вводить данные о еде и узнавать количество калорий в ней. (Например, <https://supercalorizator.ru/> или <https://www.calc.ru/kalkulyator-kalorii.html>).

В результате анализа существующих решений был сформирован перечень основных недостатков аналогичных веб-приложений, а также продуманы варианты их устранения:

Недостаток	Решение
Нет возможности перейти к другим товарам, не поднимаясь на верх страницы	Добавить «Товары» в шапку сайта, которая не должна пропадать при пролистывании страницы
Большие брошюры при просмотре каталога товаров	Не использовать большие брошюры, которые нагружают основные элементы страницы каталога товаров или отвлекают пользователя от основного контента
Нет возможности установить диапазон цен	Реализовать возможность регулирования максимальной и минимальной цены

Таблица 1 - Недостатки аналогов и варианты их устранения

## 2.4 Анализ задачи

### 2.4.1 Варианты использования приложения

Диаграмма прецедентов представлена на Рис. 6:

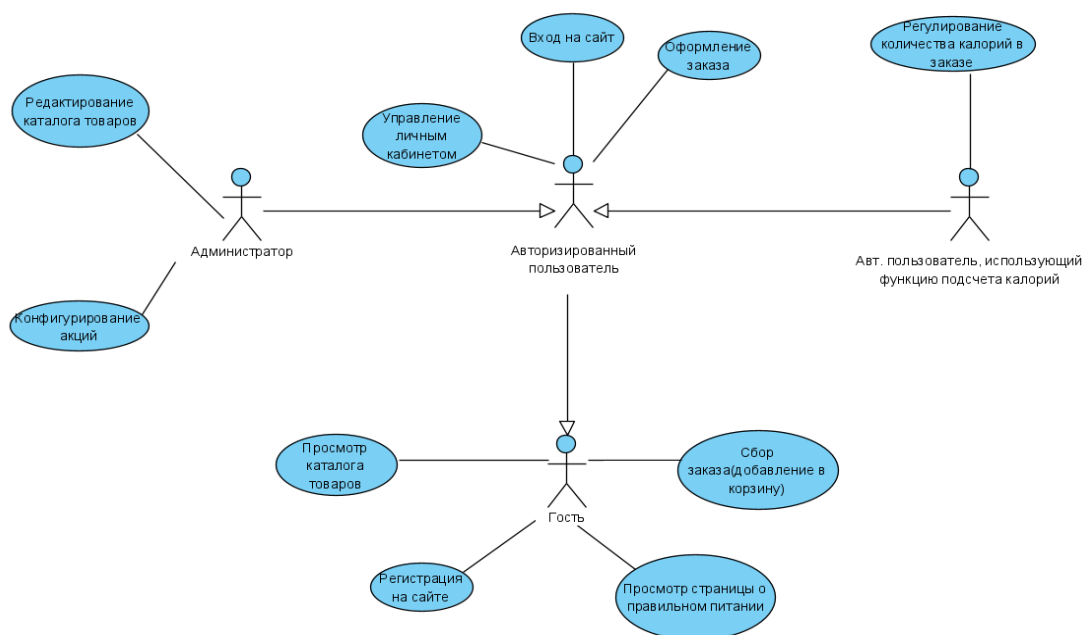


Рисунок 6 - Диаграмма прецедентов

Гость (неавторизованный в системе пользователь) имеет следующие возможности:

- Возможность просмотра всех страниц сайта, кроме Личного кабинета;

- Возможность добавления товаров в корзину;
- Возможность регистрации на сайте.

В процессе взаимодействия с системой пользователь имеет следующие возможности:

- Возможность просмотра всех страниц;
- Возможность редактирования личных данных;
- Возможность добавления товаров в корзину;
- Возможность оформления заказа;
- Возможность оплаты заказа на сайте;
- Возможность видеть количество калорий каждого продукта, суммарное количество калорий в корзине и каждого продукта в отдельности.

Пользователь, подключенный к системе подсчета калорий, обладает теми же возможностями, что и обычный пользователь, и дополнительной возможностью видеть предупреждающее сообщение в корзине, в случае превышения своей суточной нормы калорий.

В процессе взаимодействия с системой администратор имеет те же возможности, что и пользователь, и несколько дополнительных:

- Возможность добавлять, изменять и удалять товары;
- Возможность конфигурирования акций (добавление, удаление, изменение).

Отметим, что администратор, хоть и может, просматривая каталог, добавлять товары в корзину, функция оформления заказа для него отключена.

Опишем конкретно 3 сценария использования приложения.

Первый сценарий – сценарий незарегистрированного пользователя. Гость, попадая на сайт, может просматривать каталог товаров и добавлять товары в корзину. Затем он, вероятнее всего, захочет оформить заказ, однако система потребует от гостя, чтобы он зарегистрировался. Это необходимо для того, чтобы полностью оформить заказ (внести имя клиента, адрес доставки и

способ оплаты). Здесь часть пользователей может покинуть сайт из-за того, что не желает проходить регистрацию

Второй сценарий – сценарий зарегистрированного пользователя. Ему доступно оформление заказов. Некоторая группа пользователей может прекратить оформление заказов по той причине, что не желает производить оплату на сайте (магазин поддерживает только безналичную оплату).

Третий сценарий – пользователь, который включил функцию подсчета калорий. Он может при оформлении заказа увидеть сообщение о том, что его суточная норма потребления калорий может быть превышена, и перед оплатой изменить состав заказа, тем самым регулируя количество калорий в нём. Однако, конверсия по этому сценарию уменьшится, так как не всем пользователям требуется такая функция.

#### 2.4.2 Взаимодействие компонентов системы

Рассмотрим диаграммы последовательностей и взаимодействий с системой.

Ниже представлена диаграмма (Рис.7), отражающая процесс регистрации гостя на сайте:

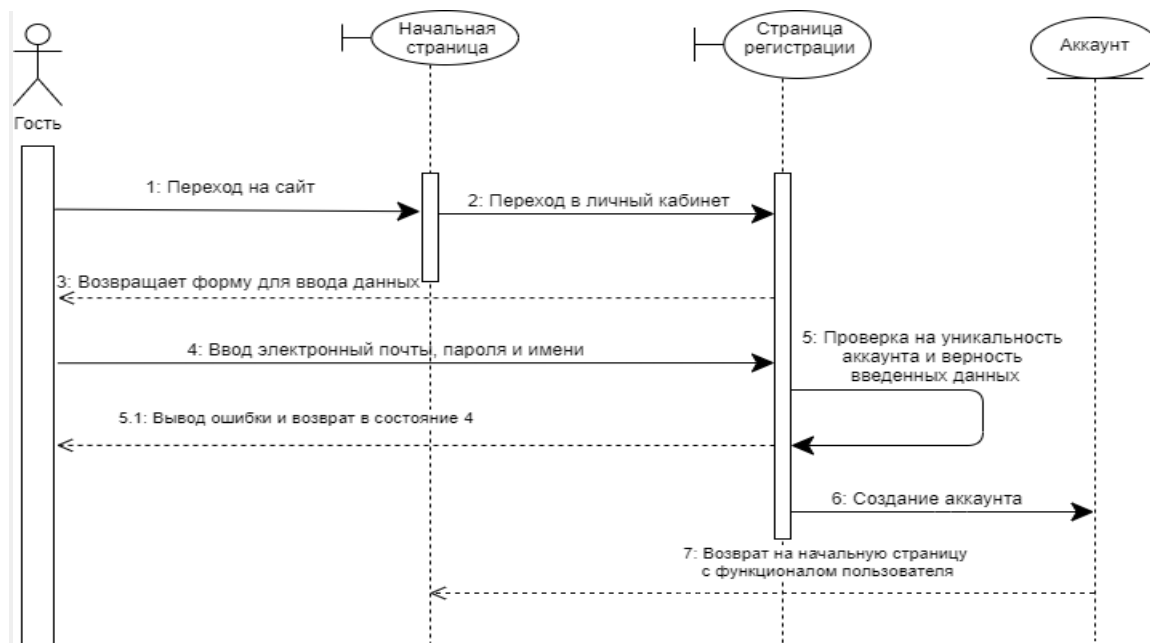


Рисунок 7 - Диаграмма взаимодействия для прецедента «Регистрация на сайте»

Гостю необходимо посетить главную страницу приложения, заполнить форму со своими личными данными, дождаться проверки введенных данных. После проведенных действий гость становится пользователем и имеет доступ ко всем возможностям, предоставленным пользователю.

Рассмотрим процесс взаимодействия пользователя для прецедента «Просмотр каталога товаров» (Рис.8)

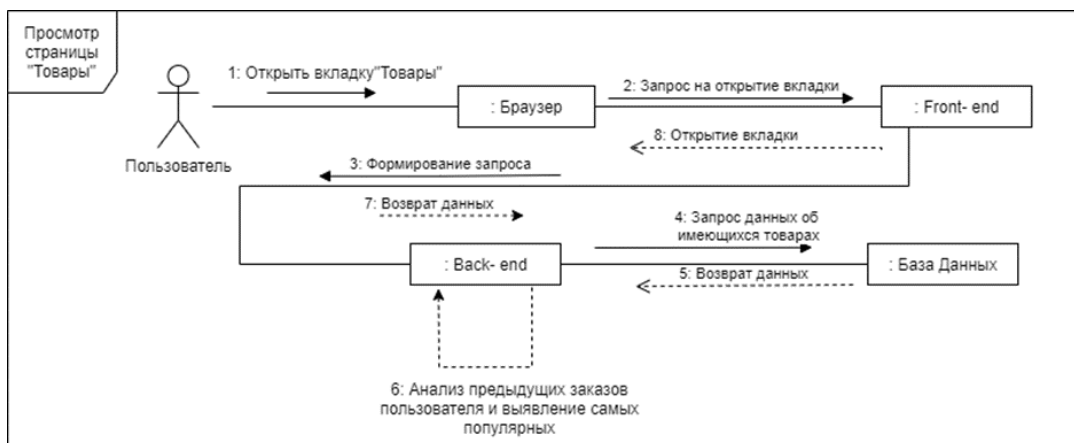


Рисунок 8 - Диаграмма взаимодействия для прецедента «Просмотр каталога товаров»

Важно отметить, что Пользователь отличается от Гостя тем, что каталог товаров отображается на основе покупок Пользователя, совершенных им ранее: в первую очередь будут показаны самые часто покупаемые Пользователем товары.

Рассмотрим варианты взаимодействия пользователя и системы (Рис. 9) и гостя с системой (Рис.10)

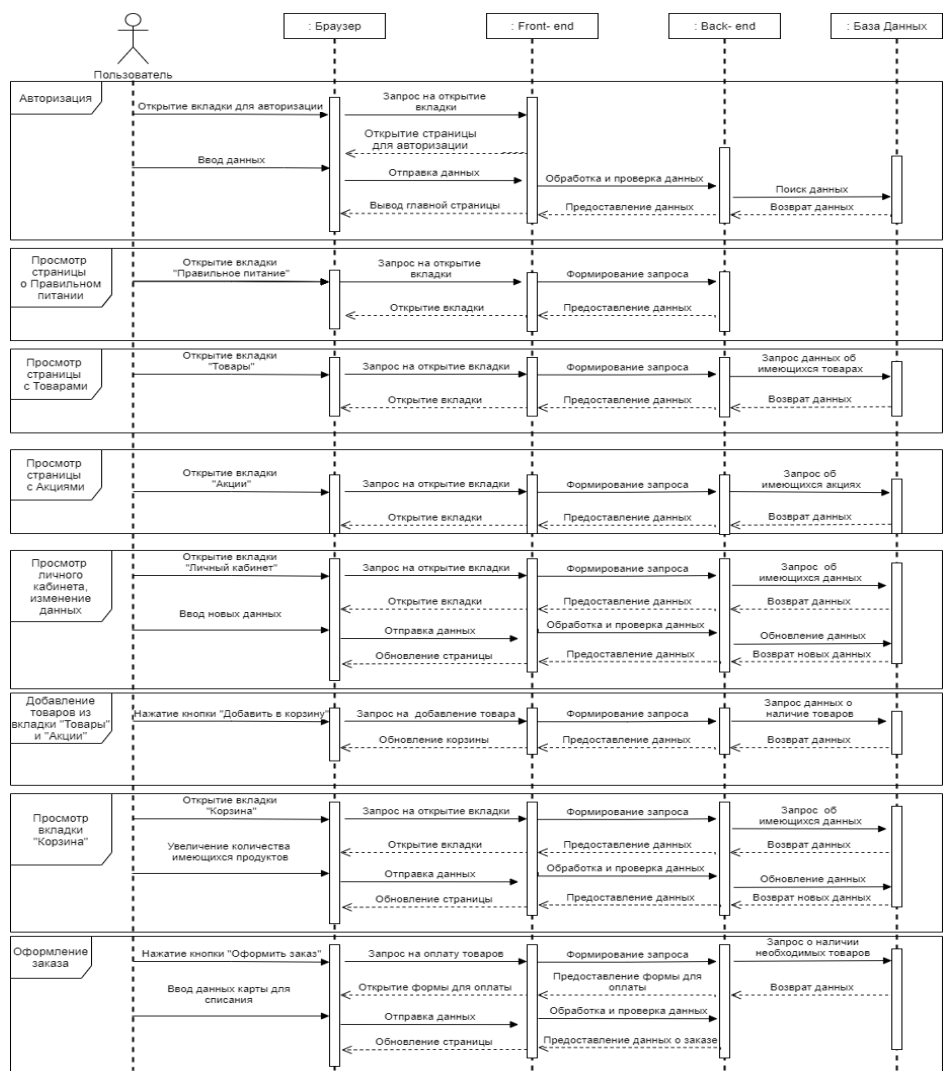


Рисунок 9 - Диаграмма последовательности пользователя с системой

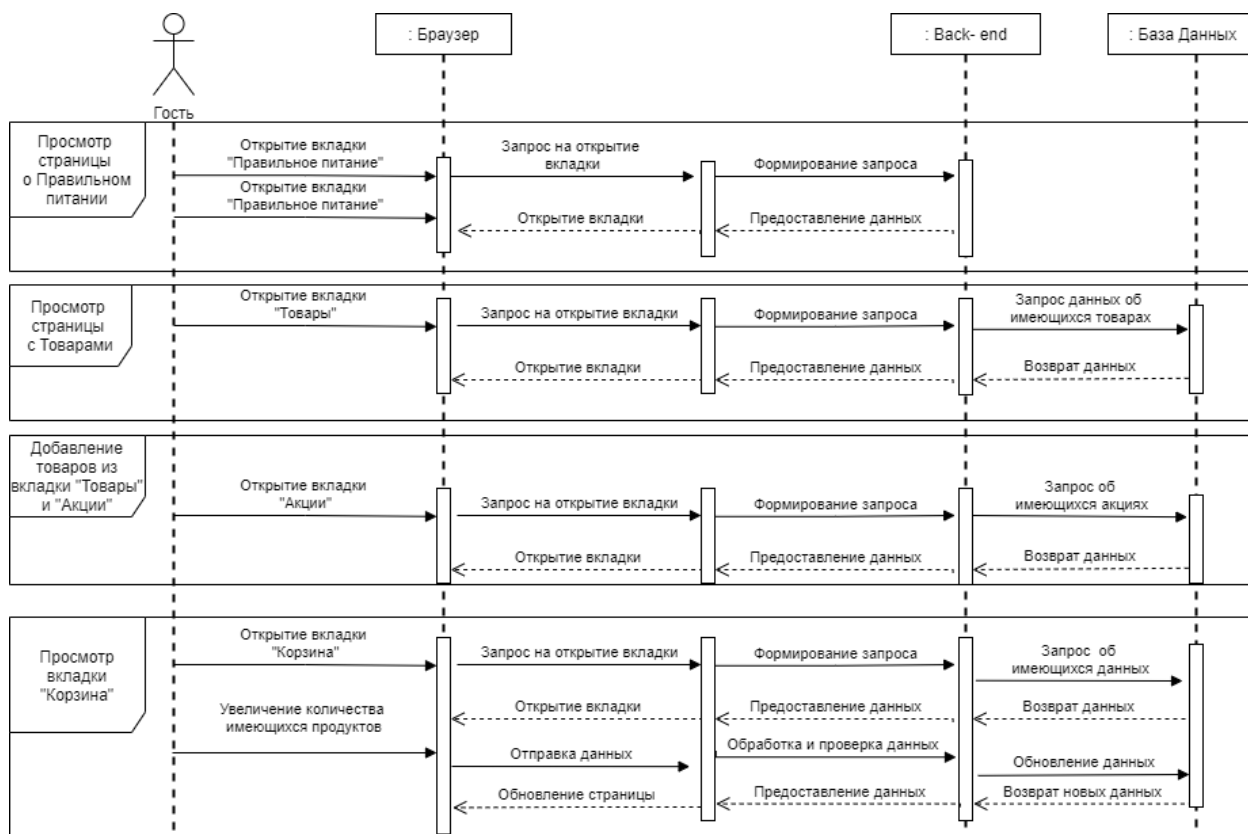


Рисунок 10 - Диаграмма последовательности гостя с системой

### 2.4.3 Варианты состояния системы

Данная диаграмма (Рис.11) отражает состояния системы, в которых она может находиться во время просмотра пользователями всех страниц сайта и оформлении заказа.

Система начинает работу с пользователями, когда они заходят на главную страницу сайта. Всем посетителям сайта доступны возможности гостя. Если у посетителя есть аккаунт, то необходима авторизация для возможности оформления заказа; если же аккаунта нет, то необходима сначала регистрация.

После авторизации в системе пользователь может перейти на все страницы сайта, добавлять товары, формировать заказ и оплачивать его. После перехода в «Корзину» пользователь может проверить выбранные продукты и перейти к оплате.



Авторизованный пользователь также может перейти на страницу личного кабинета, изменить личные данные и выйти из системы.

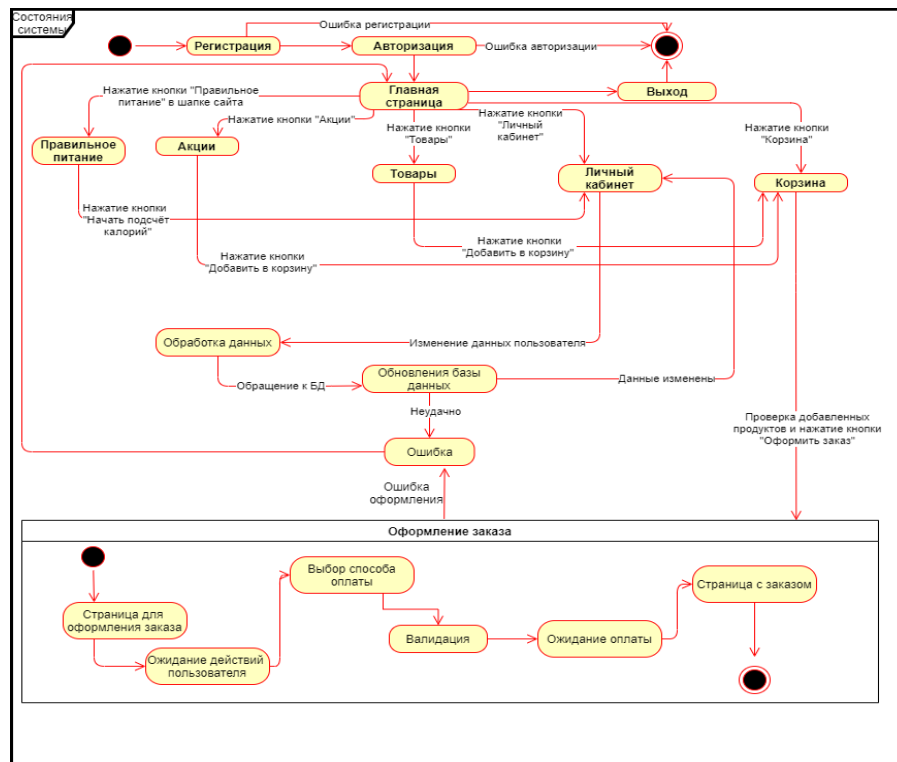
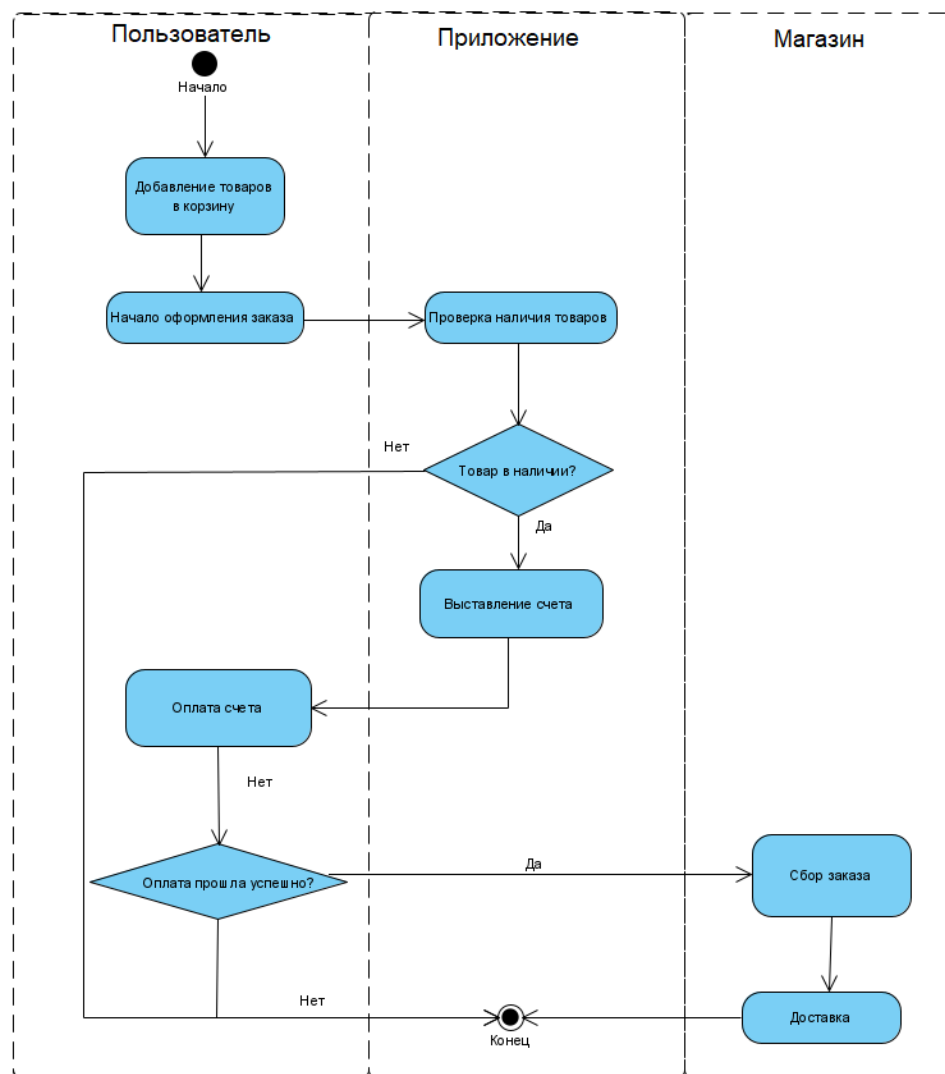


Рисунок 11 - Диаграмма состояний системы

#### 2.4.4 Варианты действий в системе

Ниже представленная диаграмма (Рис.12) дает возможность увидеть варианты действий пользователя при оформлении заказа в системе.



**Рисунок 12 -** Диаграмма активностей оформления заказа

## 2.4.5 Развертывание приложения

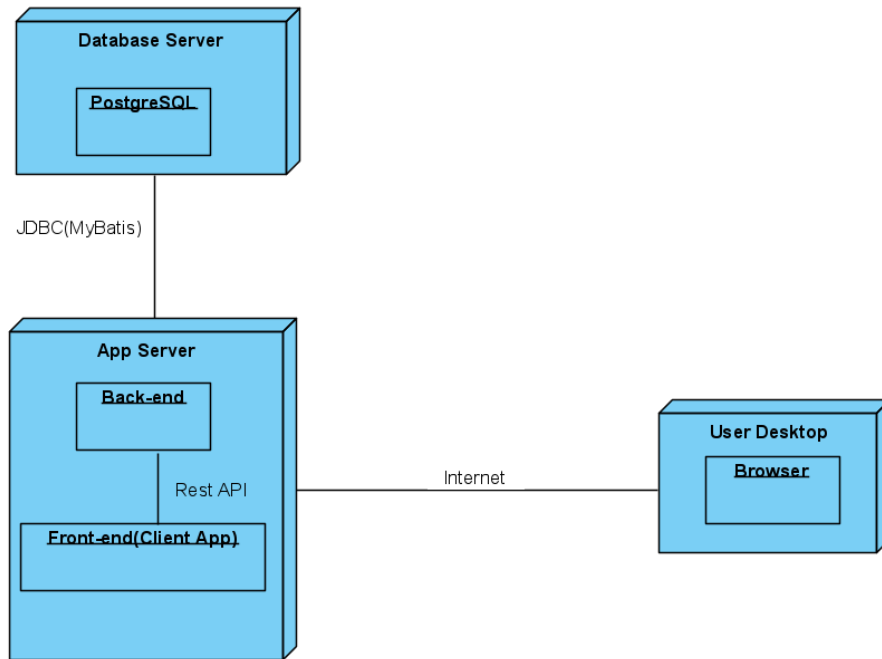


Рисунок 13 - Диаграмма развёртывания

Данная диаграмма (Рис.13) позволяет определить какие аппаратные компоненты («узлы») и программные компоненты («артефакты») существуют, как работают артефакты на каждом узле и как различные части системы соединяются друг с другом. Для разрабатываемого веб-приложения узлом устройства является компьютер, сервер и база данных, а в качестве узла среды выполнения выступает браузер на компьютере клиента. На серверной части развернуты front-end и back-end приложения и отдельно база данных.

## 2.4.6 Диаграмма классов

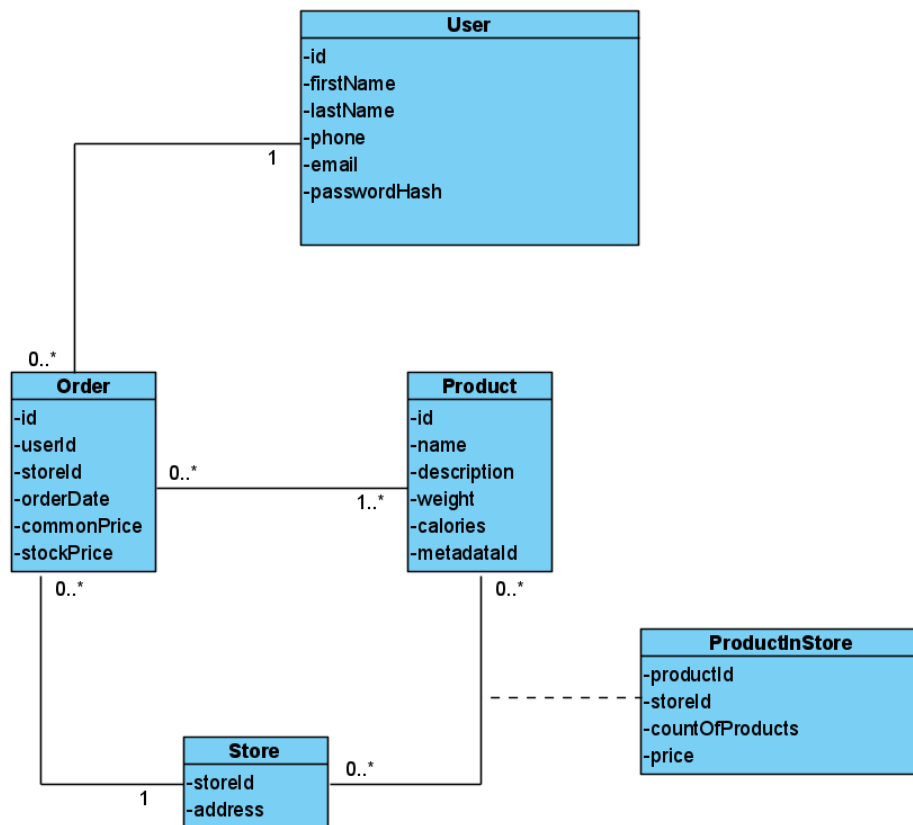


Рисунок 14 - Диаграмма классов

Диаграмма классов (Рис.14) отображает основные сущности системы и различного рода связи, которые существуют между ними.

## 2.4.7 Диаграмма объектов

Диаграмма объектов предназначена для демонстрации совокупности моделируемых объектов и связей между ними в фиксированный момент времени. Рассмотрим момент (Рис.15), когда в системе есть два пользователя, которые выбирают товары и формируют заказы. Причём товар «Яблоко» попадает в оба заказа.

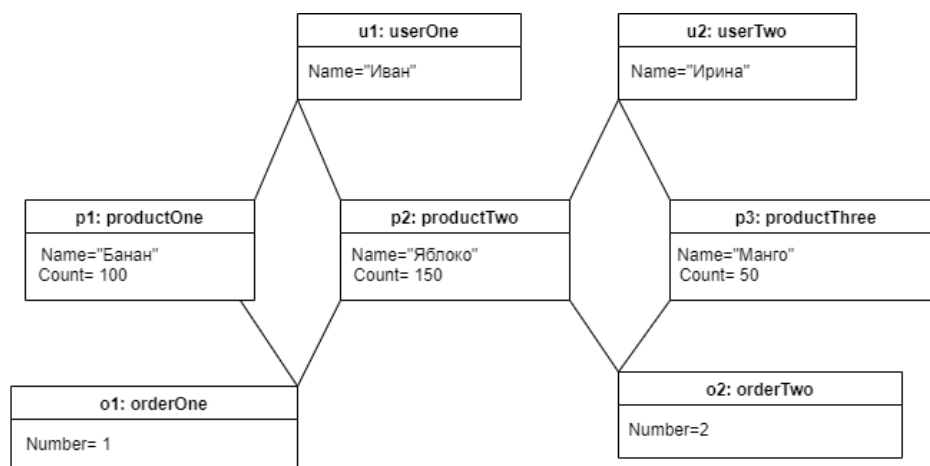
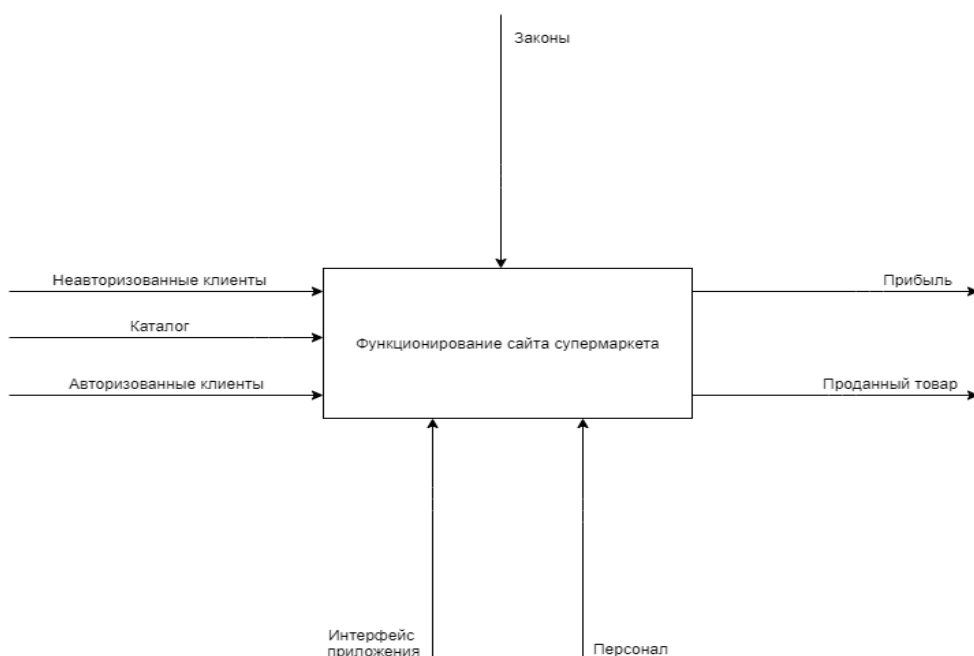


Рисунок 15 - Диаграмма объектов

#### 2.4.8 IDEF0

Контекстная диаграмма системы функционирования сайта супермаркета (или диаграмма верхнего уровня) является вершиной древовидной структуры диаграмм и показывает назначение системы (или основную функцию) и ее взаимодействие с внешней средой. Для супермаркета «GoodFood» контекстная диаграмма представлена на Рисунке 16.



A-0

Рисунок 16 - Контекстная диаграмма

Диаграмма декомпозиции, получающиеся при разбиении контекстной диаграммы на крупные подсистемы, предназначена для детализации функций и описывают каждую подсистему и взаимодействие с другими подсистемами. Рассмотрим диаграмму декомпозиции первого уровня (Рис. 17), демонстрирующую оформление заказа пользователем, который изначально был гостем.

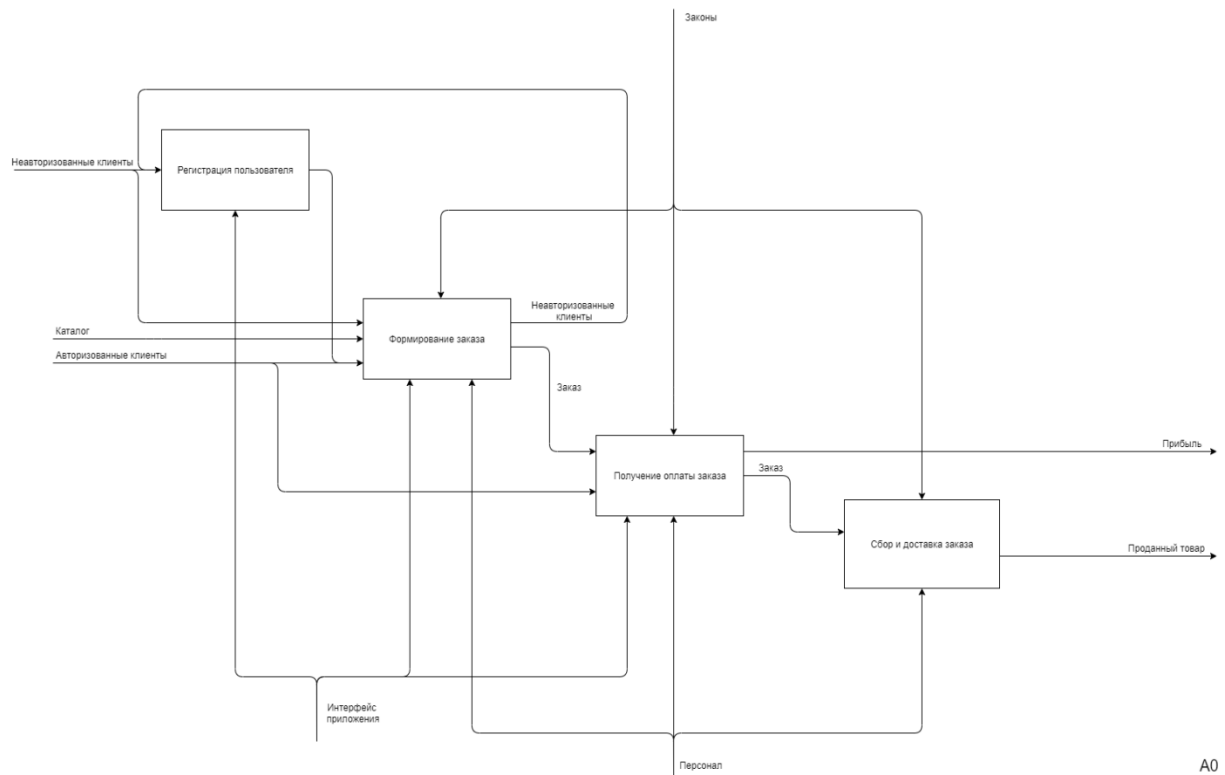


Рисунок 17 - Диаграмма декомпозиции первого уровня

### 3 Стек используемых технологий

В качестве средств реализации приложения были выбраны следующие технологии:

- Язык программирования Java;
- Spring Framework(Spring Boot);
- MyBatis Framework;
- PostgreSQL Database;
- Angular Framework;
- Bootstrap;

- HTML 5;
- CSS 3;
- TypeScript;
- JavaScript.

Spring обеспечивает решения многих задач, с которыми сталкиваются Java-разработчики и организации, которые хотят создать информационную систему, основанную на платформе Java. Из-за широкой функциональности трудно определить наиболее значимые структурные элементы, из которых он состоит. Spring не всецело связан с платформой Java Enterprise, несмотря на его масштабную интеграцию с ней, что является важной причиной его популярности.

Spring Boot позволяет вам легко создавать полноценные, производственного класса Spring-приложения, про которые можно сказать - "просто запустить". Большинству Spring Boot приложениям требуется совсем маленькая Spring-конфигурация. Возможности:

- Создание полноценных Spring приложений;
- Встроенный Tomcat или Jetty (не требуется установки WAR файлов);
- Обеспечивает 'начальные' POMs для упрощения Maven конфигурации;
- Автоматическая конфигурирование Spring когда это возможно;
- Абсолютно без генерации кода и без написания XML конфигурации.

Angular — открытая и свободная платформа для разработки веб-приложений, написанная на языке TypeScript, разрабатываемая командой из компании Google, а также сообществом разработчиков из различных компаний. Он позволяет использовать привычные редакторы и расширения для браузеров. Версия Angular CLI (Command Line Interface) стандартизирует структуру, позволяет создать сущности внутри приложения, а также автоматизировать его сборку.

На фреймворках разрабатываются довольно большие и сложные сайты с уникальным функционалом. Это значительно быстрее и дешевле, чем на чистом языке, но при этом такое решение позволяет разрабатывать действительно сложные вещи и оптимизировать все это под нагрузки. Кроме того, это почти всегда более безопасно, чем любая коробочная CMS.

Существует множество языков программирования, шаблонов и фреймворков. Но, благодаря многим функциям, упомянутым выше, Spring Framework и Angular является отличным выбором для нашего проекта.

В качестве СУБД была выбрана PostgreSQL в силу открытого доступа и высокой производительности, а также за счет поддержки данной СУБД schema-less данных, такие как JSON. PostgreSQL поддерживает интеграцию на различные платформы, взаимодействие с большинством языков программирования. PostgreSQL имеет преимущество перед другими DBMS, когда необходимо перенести базу данных из одной ОС в другую или в случае, когда реализация приложения производится на различных ОС.

Для связи back-end с СУБД используется MyBatis. MyBatis - это постоянная среда Java, которая связывает объекты с хранимыми процедурами или операторами SQL с использованием XML-дескриптора или аннотаций.

Специфика клиентской стороны веб-приложения требует отладки кода во всех популярных браузерах. Наше приложение разрабатывалось под браузер Google Chrome. Этот браузер, основан на WebKit, обладает встроенным инструментом разработки Web Inspector, который очень хорошо развит и позволяет выполнять отладку JavaScript кода.

Общение между Front-end и Back-end происходит по средствам REST API, а обмен информации происходит с помощью передачи данных в формате JSON.