

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный автомобильно-дорожный университет
(СибАДИ)»

Факультет	Информационные системы в управлении
Кафедра	Автоматизированные системы и цифровые технологии
Направление	Информатика и вычислительная техника
Профиль	Автоматизированные системы обработки информации и управления

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к выпускной квалификационной работе

Обозначение ВКР	ВКР-02068982-09.03.01-12-20
Тема ВКР	Разработка online-сервиса по выбору и покупке электронных и цифровых товаров для торгового предприятия
Студент	АСб-16И1 Крылов Валерий Сергеевич

Выпускная квалификационная работа допущена к защите в ГЭК

Заведующий кафедрой АСи ЦТ, канд. техн. наук, доц.	_____ С.А.Зырянова Дата _____
---	----------------------------------

Руководитель ВКР проф. каф. АСи ЦТ, д-р техн. наук, проф.	_____ С.Н.Чуканов Дата _____
--	---------------------------------

Нормоконтроль старший преподаватель каф. АСи ЦТ	_____ О.А.Филимонова Дата _____
--	------------------------------------

Омск 2020

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный автомобильно-дорожный университет
(СибАДИ)»

Кафедра Автоматизированные системы и цифровые технологии
УТВЕРЖДАЮ

Зав. кафедрой Зырянова С.А.

«___» _____ 2020 г.

ЗАДАНИЕ

к выпускной квалификационной работе (ВКР) студента Крылова Валерия Сергеевича

1. Тема ВКР Разработка online сервиса по выбору и покупке электронных и цифровых товаров

2. Исходные данные ВКР

- результаты преддипломной практики;
- результаты анализа литературы и интернет-источников.

3. Содержание пояснительной записки (конкретный перечень подлежащих разработке по разделам)

- 1) Введение
- 2) Исследование объекта автоматизации
- 3) Исследование аналогов
- 4) Проектирование системы
- 5) Разработка системы
- 6) Документационное обеспечение системы
- 7) Заключение
- 8) Список использованных источников

5. Перечень демонстрационного материала для сопровождения докладов в ГЭК

- 1) Презентация к выпускной квалификационной работе

Задание выдано «___» _____ 2020 г.

Руководитель ВКР _____ / Чуканов С.Н.

подпись

Задание к исполнению принял «___» _____ 2020 г.

Студент _____ / Крылов В.С.

подпись

РЕФЕРАТ

Пояснительная записка 61 с., 32 рис., 10 табл., 31 источник.

WEB-ПРИЛОЖЕНИЕ, ТОРГОВЛЯ ЭЛЕКТРОННЫМИ И ЦИФРОВЫМИ ТОВАРАМИ, СИСТЕМА, БАЗА ДАННЫХ, СЕРВЕР

Предметом проектирования являются процессы оформления заказов, регистрация новых пользователей, авторизация новых пользователей, смена статусов существующих заказов.

Целью работы является создание web-приложения (клиентской и серверной части), позволяющего регистрировать заказы пользователей через сеть интернет и контролировать текущее состояние всех заказов пользователя.

В процессе работы производилось проектирование моделей программной системы, осуществлялась разработка базы данных и серверной части приложения, работающего с базой, а также клиентской части, работающей в интернет – браузере на стороне пользователя.

Результатом выполнения работы является приложение, имеющее веб – интерфейс и представляющее средства для выбора и заказа необходимых товаров.

Созданная система имеет простой пользовательский интерфейс, для освоения которого не понадобится много времени, также отличительной особенностью системы является то, что она построена по концепции одностраничного приложения со множеством компонентов (SPA).

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

API	Application Programming Interface (рус. Программный интерфейс приложения)
URL	Universal Resource Locator (рус. Универсальный локатор ресурса)
JSON	JavaScript Object Notation (рус. Нотация объекта JavaScript)
SPA	Single Page Application (рус. Одностраничное приложение)
B2B	Business-to-business (рус. Бизнес для бизнеса)
B2C	Business-to-customer (рус. Бизнес для клиента)
DOM	Document Object Model (рус. Объектная модель документа)
АС	Автоматизированная Система
ВУЗ	Высшее Учебное Заведение
ОС	Операционная Система
ПО	Программное Обеспечение
СУБД	Система управления базами данных

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1 ИССЛЕДОВАНИЕ ОБЪЕКТА АВТОМАТИЗАЦИИ	10
1.1 Функции системы	10
1.2 Входная и выходная информация.....	10
1.3 Объектная модель программной системы	12
1.3.1 Построение диаграммы прецедентов	13
1.3.2 Построение диаграммы деятельности.....	15
1.3.3 Построение диаграммы компонентов	17
1.3.4 Построение диаграммы развертывания	18
1.4 Функциональная модель системы	21
2 ИССЛЕДОВАНИЕ АНАЛОГОВ	23
2.1 Интернет-магазин РС-1	23
2.2 Интернет-магазин электроники Zakazpc.ru	25
2.3 Интернет-магазин по продаже компьютерной техники и программного обеспечения «МАКСИМУМ техноцентр»	27
2.4 Сравнение систем аналогов.....	28
3 ПРОЕКТИРОВАНИЕ СИСТЕМЫ	30
3.1 Назначение и цели создания системы	30
3.2 Характеристика объекта автоматизации	30
3.3 Требования к системе.....	31
4 РАЗРАБОТКА СИСТЕМЫ	33
4.1 Общие сведения.....	33
4.2 Выбор и обоснование программного обеспечения	34
4.2.1 Система управления базами данных	34
4.2.2 Программирование серверной части	36
4.2.3 Программирование клиентской части	39
5 ДОКУМЕНТАЦИОННОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ	42
5.1 Технические требования для выполнения АС	42

5.2 Руководство пользователя.....	43
5.2.1 Регистрация и авторизация	43
5.2.2 Взаимодействие с каталогом товаров и корзиной	45
5.2.3 Управление заказами	48
5.2.4 Модификация информации в профиле	51
5.2.5 Инструменты поиска и фильтрации товаров	52
5.3.1 Просмотр пользовательских профилей и заказов.....	54
5.3.2 Операции добавления, модификации и удаления товаров	55
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	59
ПРИЛОЖЕНИЕ А	62

ВВЕДЕНИЕ

Сфера торговли в современном цифровизированном мире является одним из главных двигателей мировой экономики. Для успешного ведения бизнеса, в котором, необходимо оперативно и незамедлительно действовать, и принимать быстрые решения. Большую роль играют автоматизированные системы и инструменты, увеличивающие надежность, скорость и эффективность выполнения основных бизнес-процессов, выполняемых организациями. Так же данные системы предоставляют возможность формировать различные статистические данные, помогающие бизнесу успешно развиваться и принимать верные решение в области планирования своей коммерческой деятельности. Необходимость использования автоматизированных торговых систем и систем электронной коммерции в современном мире становится уже необходимостью для эффективной работы различных компаний.

Объектом выполненной работы может являться предприятие торгового профиля, которое ведет продажу своих товаров или услуг через сеть интернет.

Предметом проектирования является - online сервис по выбору и покупке электронных и цифровых товаров для торгового предприятия.

Актуальность создания веб-приложения состоит прежде всего в том, что если вы хотите донести информацию максимально быстро до огромного количества людей, то лучше, чем с помощью собственного сайта сделать это не получится никак [9]. Веб-ресурс позволяет представить информацию о компании и ее товарах или услугах сжато и одновременно полноценно. Также веб-приложение может сообщать о новостях фирмы, об изменениях в прайсе или режиме работы, содержать отзывы благодарных клиентов. Главное преимущество интернет торговли – интерактивность составляющих её элементов. Интернет-торговля на собственной платформе связана с существенно меньшим количеством издержек, чем торговля в обычном

магазине. Интернет-магазин в виде веб-приложения позволяют снизить траты на оплату труда сотрудникам, а также уменьшить затраты на аренду помещения и его обслуживание.

При проектировании системы использовались следующие методы: метод морфологического анализа, метод декомпозиции.

Метод декомпозиции заключается в том, что мы можем разделить автоматизированную систему на несколько составных частей функциональность которых будет составлять общую функциональность системы в целом.

Метод морфологического анализа позволяет проработать концепцию решения отдельных разработческих задач и последующую их систематизацию в решение общей задачи по функционированию системы.

Новизна и отличительные особенности автоматизированной системы состоят в использовании современной серверной технологии *Node.js* которая основана на событийно-ориентированном подходе по обработке HTTP запросов с клиентского устройства. Этот подход серьезно снижает накладные расходы по сравнению с идеологией «один поток на каждое соединение» которая используется в традиционной модели параллелизма на основе потоков. Node.js легко обрабатывает большое количество запросов одновременно и обеспечивает быстрое действие приложения. Поэтому серверный JavaScript часто используют для создания SPA — одностраничных веб-приложений, в которых рендеринг выполняется на стороне клиента. Node.js на бэкенде используют Netflix, Uber, eBay, Groupon, Yahoo и другие известные организации, и проекты.

Практическая значимость проекта заключается в том, что он позволяет апробировать и внедрить современные подходы и методологии по созданию web-приложений и на практике оценить эффективность этих подходов. Позволит оценить удобство и практическую применимость использованных технологий в условиях, приближенных к реальному коммерческому проекту.

А также сравнить используемые подходы и технологии с аналогичными или конкурирующими на рынке разработки коммерческого ПО.

Структура пояснительной записки приведена ниже.

Первая глава описывает цели и задачи разрабатываемой системы, а так формулирует требования, обозначает входную и выходную информацию.

Во второй главе описывается объектная модель разрабатываемой системы посредством диаграмм деятельности, прецедентов, компонентов и развертывания, служащих основанием для моделирования процессов.

Третья глава описывает общие сведения о разработке, выборке базы данных и ее модели, методику проектирования основные алгоритмы функционирования системы, выбор и обоснование языка и среды программирования.

Четвертая глава является руководством пользователя. Обозначает требования к программному и техническому обеспечению, описывает основные элементы интерфейса АС, позволяет понять принципы работы с системой и выполнения всех доступных операций.

На данном этапе система находится в стадии тестирования.

Областью возможного применения автоматизированной системы является сфера торговли, предприятия по продаже товаров и услуг.

Целью работы является повышение удобства обслуживания клиентов заказчика, повышение экономической эффективности работы путем автоматизации процессов сбора и хранения информации о текущих заказах, завершенных заказов, и мониторинг текущего состояния хранилища доступных товаров.

1 ИССЛЕДОВАНИЕ ОБЪЕКТА АВТОМАТИЗАЦИИ

1.1 Функции системы

Система автоматизации торговли должна осуществлять выполнение следующих функций:

- 1) хранение большого массива данных:
 - о клиентах организации;
 - регистрационные данные клиентов;
 - авторизационные данные клиентов;
 - о товаре и его описании;
 - о заказах;
 - о количестве товара на складе.
- 2) корректное и оперативное выполнение запросов пользователей:
 - внесение данных в систему;
 - изменение данных в системе;
 - удаление данных из системы.
- 3) контроль работы пользователей:
 - разграничение прав доступа к определенной информации;
 - защита от некорректных действий пользователя.

1.2 Входная и выходная информация

Основной информацией, которая должна находиться в АС, являются данные о товаре (описание), количестве товара, клиентах и заказах.

В информации о клиентах должно содержаться следующее: ID в базе данных разрабатываемой системы, контактное лицо, телефон, почта. Выделены следующие сущности предметной области. Описание сущностей и атрибутов представлено в таблицах ниже.

Условные обозначения: PK (primary key) – первичный ключ; FK (foreign key) – внешний ключ.

Таблица 1 – Сущность «Заказы»

Имя атрибута	Описание атрибута	Тип данных
Код заказа (PK)	Идентифицирующий атрибут	INTEGER
Код покупателя (FK)	Идентификатор покупателя	INTEGER
Общая сумма	Общая стоимость всех позиций заказа	INTEGER
Статус заказа	Текущий статус заказа	STRING

Таблица 2 – Сущность «Товары в заказе»

Имя атрибута	Описание атрибута	Тип данных
ИД-записи (PK)	Идентифицирующий атрибут	INTEGER
Количество	Количество данного товара в заказе	INTEGER
Код заказа (FK)	Идентификатор заказа	INTEGER
Код товара(FK)	Идентификатор товара	INTEGER

Таблица 3 – Сущность «Товары»

Имя атрибута	Описание атрибута	Тип данных
Код товара (PK)	Идентификатор товара	INTEGER
Название	Название товара	STRING
Количество	Количество товара на складе	INTEGER
Описание	Краткое текстовое описание товара	STRING
Цена	Цена товара	INTEGER (250)
Изображение	Ссылка на фотографию товара	STRING

Таблица 4 – Сущность «Авторизационные данные пользователей»

Имя атрибута	Описание атрибута	Тип данных
Код записи (PK)	Идентифицирующий атрибут	INTEGER
E-mail	Адрес электронной почты пользователя (она же является логином)	STRING
Пароль	Пароль от аккаунта пользователя	STRING
Код пользователя (FK)	Идентификатор пользователя	INTEGER

Таблица 5 – Сущность «Пользователи»

Имя атрибута	Описание атрибута	Тип данных
Код пользователя (PK)	Идентифицирующий атрибут	INTEGER
Имя	Имя пользователя	STRING (30)
Баланс	Текущий денежный баланс пользователя	INTEGER
Телефон	Номер телефона сотрудника	STRING (12)
E-mail	Электронный адрес	STRING (30)

Таким образом, к сущностям входных данных относятся: Товары, Описание, Количество товара, Пользователи. Сущностями исходящей информации являются: Данные о заказах пользователей.

1.3 Объектная модель программной системы

Для разработки автоматизированной системы необходимо понимание всех процессов, которые происходят в рамках функционирования данной системы, анализ потоков данных, который может существовать у организации занимающейся торговлей. Для этого необходимо построить ряд диаграмм, которые представлены ниже.

1.3.1 Построение диаграммы прецедентов

На диаграмме прецедентов (рисунок 1.3.1.1), которую также называют диаграммой вариантов использования [4], показано то, какие существуют варианты использования данной системы и какие взаимодействия могут между ними происходить. Варианты использования представляют собой функции системы, эти функции взаимодействуют с действующими лицами, которые представляют людей или системы. Действующие лица могут получать информацию из системы или передавать информацию в систему. Диаграммы вариантов использования предоставляют достаточно обширную информацию о том, как работает система и как она устроена. Диаграмма прецедентов описывает функционирование системы в обобщенном виде, без представления деталей реализации конкретных процессов.

Для нашей предметной области мы выделили актеров, представленных в таблице 6. Описание прецедентов находится в таблице 7.

Таблица 6 – Описание актеров

Актер	Краткое описание
Клиент	Физическое лицо, потенциальный покупатель интернет-магазина, выбирает товар по каталогу на сайте, делает заказ, оплачивает безналично через банк или наличными при доставке
Администратор	Управляет каталогом товаров, отслеживает заказы клиентов, следит за наличием товара

Таблица 7 – Описание прецедентов

Прецедент	Краткое описание
Работа с заказами	Запускается модератором сайта. Позволяет вносить, изменять, удалять или просматривать заказ.
Ведение каталога клиентов	Запускается модератором. Позволяет добавлять, редактировать, удалять данные о клиенте в БД
Получение информации о наличии на складе	Запускается модератором. Позволяет добавлять, редактировать, удалять данные о товарах в БД при поступлении на склад или изъятии со склада
Оплата товаров (наличный/безналичный расчет)	Выполняет клиент при покупке книжной продукции
Регистрация пользователя	Производится ввод регистрационных данных клиента, затем они отправляются на сервер для дальнейшей обработки.
Авторизация пользователя	Клиентом производится ввод своих авторизационных данных, которыми являются Е-mail и Пароль пользователя указанные при регистрации, затем данные отправляются на сервер для дальнейшей обработки и проверки
Выход из системы	Авторизованный пользователь покидает систему и переходит в разряд неавторизованных пользователей
Изменение личных данных	Пользователь имеет возможность изменять данные своего профайла

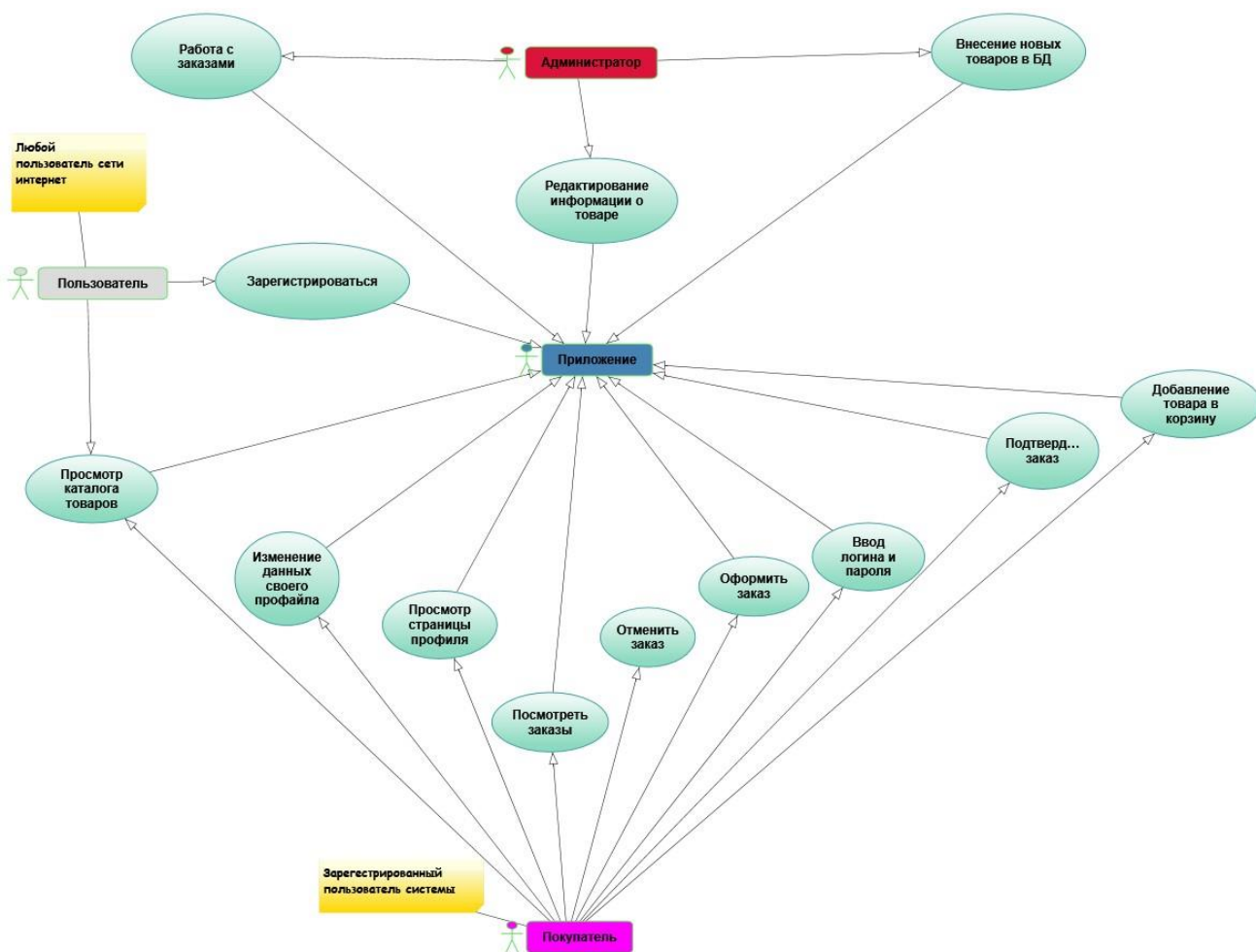


Рисунок 1.3.1.1 – Диаграмма прецедентов

1.3.2 Построение диаграммы деятельности

Диаграмма деятельности (англ. *activity diagram*) — это тип UML-диаграммы, которая отображает некоторые виды активностей, которые представляют действия, производимые в системе [10]. Деятельность определяет отличительные особенности выполняемого в системе действия в виде согласованного последовательного или координированного параллельного исполнения подчиненных элементов – внутренних видов активностей и отдельных действий, соединённых между собой стрелками, направленными от корневого узла к низлежащим подчинённым узлам.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных

вычислений. В разработанной диаграмме используются ветвления, связанные с наличием товара в базе и изменением статуса заказа (рисунок 1.3.2.1).



Рисунок 1.3.2.1 – Диаграмма деятельности

1.3.3 Построение диаграммы компонентов

Диаграмма компонентов (англ. *Component diagram*) — элемент языка моделирования UML, статическая структурная диаграмма, которая показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п. Диаграмма представлена на рисунке 1.3.3.3.

Функционирование системы основано на следующих компонентах (рисунок 1.3.3.3):

- Web (Node 12) - основной модуль, которые в свою очередь делится на серверную и клиентскую часть, и содержит весь код приложения включая конфигурационные файлы. Серверная часть базируется на движке Node - программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Клиентская часть основана на библиотеке React - JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов;

- DB Client(adminer) – клиент сервера базы данных, который имеет простой web-интерфейс с широкими возможностями по управлению БД. Adminer (бывший phpMinAdmin) — это легковесный инструмент администрирования MySQL, PostgreSQL, SQLite, MS SQL и Oracle;

- DataBase (postgres 11) - свободная объектно-реляционная система управления базами данных (СУБД). Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows. PostgreSQL базируется на языке SQL и поддерживает многие из возможностей стандарта SQL:2011.

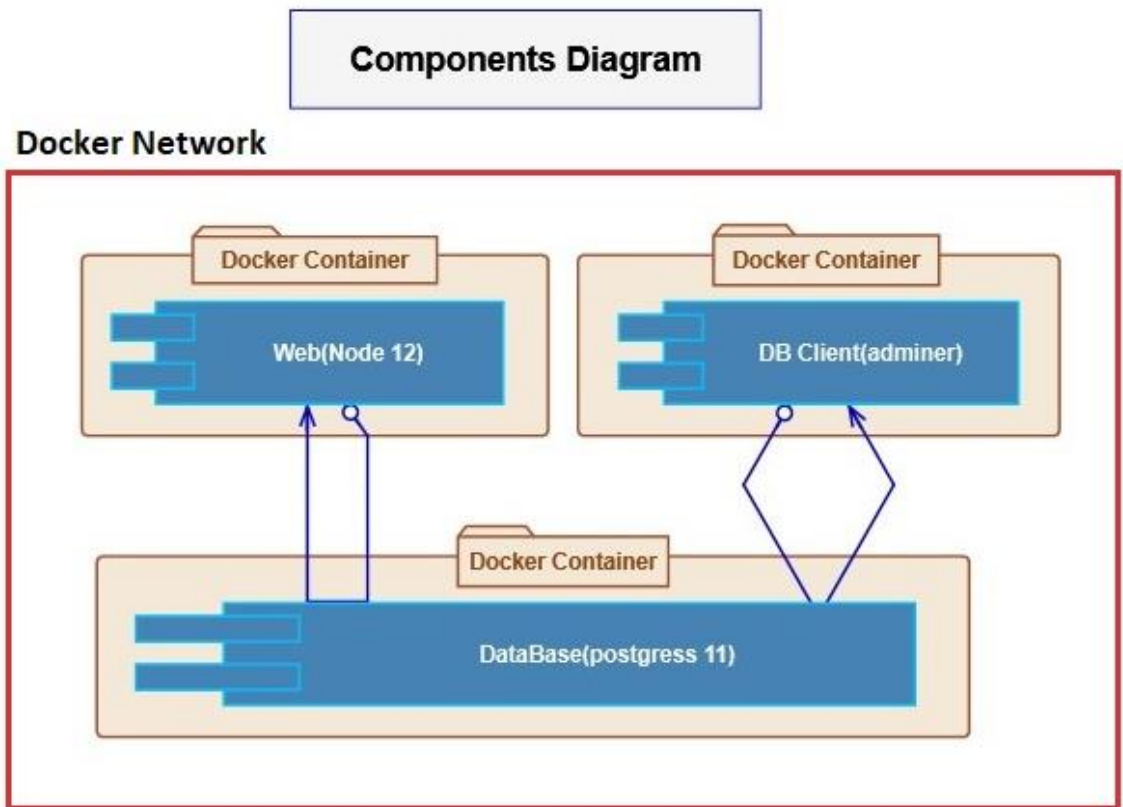


Рисунок 1.3.3.1 – Диаграмма компонентов

1.3.4 Построение диаграммы развертывания

Диаграмма развёртывания (англ. Deployment diagram) в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать веб-сайт, диаграмма развертывания должна показывать, какие аппаратные компоненты («узлы») существуют (например, веб-сервер, сервер базы данных, сервер приложения), какие программные компоненты («артефакты») работают на каждом узле (например, веб-приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, REST, RMI).

В диаграмме развертывания используются следующие узлы (рисунок 1.3.4.1):

- Web Host – отображает хостинг – провайдера которые размещает на своих серверах систему, также это может быть собственный выделенный сервер компании [5];

- Production Server –отображает физический сервер, на котором развернуты сервисы системы, в данном примере все три сервиса расположены на одном физическом сервере, но потенциально могут располагаться на разных [17];

- Web Service (node v12) – отображает сервис, содержащий слой бизнес-логики приложения, а также, клиентскую часть приложения, отдающуюся в браузер пользователя в виде java-script файла;

- DB Postgress 11 – отображает сервис, содержащий слой данных-приложения, представленных в виде таблиц реляционной базы данных;

- Client for DB (adminer) – отображает сервис, позволяющий администратору базы данных удобным способом вносить изменения в таблицы и получать новую информацию из них;

- Web developer – отображает разработчика (или команду разработчиков), который занимается техническим обслуживанием и поддержкой системы. По требованию заказчика разработчик выпускает обновления системы, устраняет уязвимости, занимается версионированием и т.д.;

- Administrator – отображает администратора, которые должен следить за работоспособность сервера, его доступностью, работать с обратной связью с пользователями, передавать необходимую информацию менеджерам;

- Sales manager – отображает менеджера, который получает информацию о новых заказах от администратора и занимается предпродажной подготовкой и планированием доставки, а также, обработкой отменённых заказов;

- ISP (Internet Provider) – отображает Интернет-провайдера, который предоставляет статический (белый) ip-адрес, используемый для

маршрутизации сервера из сети интернет, а также обеспечивающий доступ к глобальной сети с сервера;

- DNS Server Provider – отображает либо кэширующий dns-сервер провайдера, либо dns-сервер доменной зоны, в которой принадлежит домен сервера и служит для разрешения ip-адреса сервера;

- Client – отображает пользователя, подключенного к серверу.

На диаграмме отображены устройства необходимые для функционирования АС. Для понимания взаимодействия приведена диаграмма подключения некоторого количества пользователей к серверу, фактически максимальное количество пользовательских сессий ограничено только возможностями сервера.

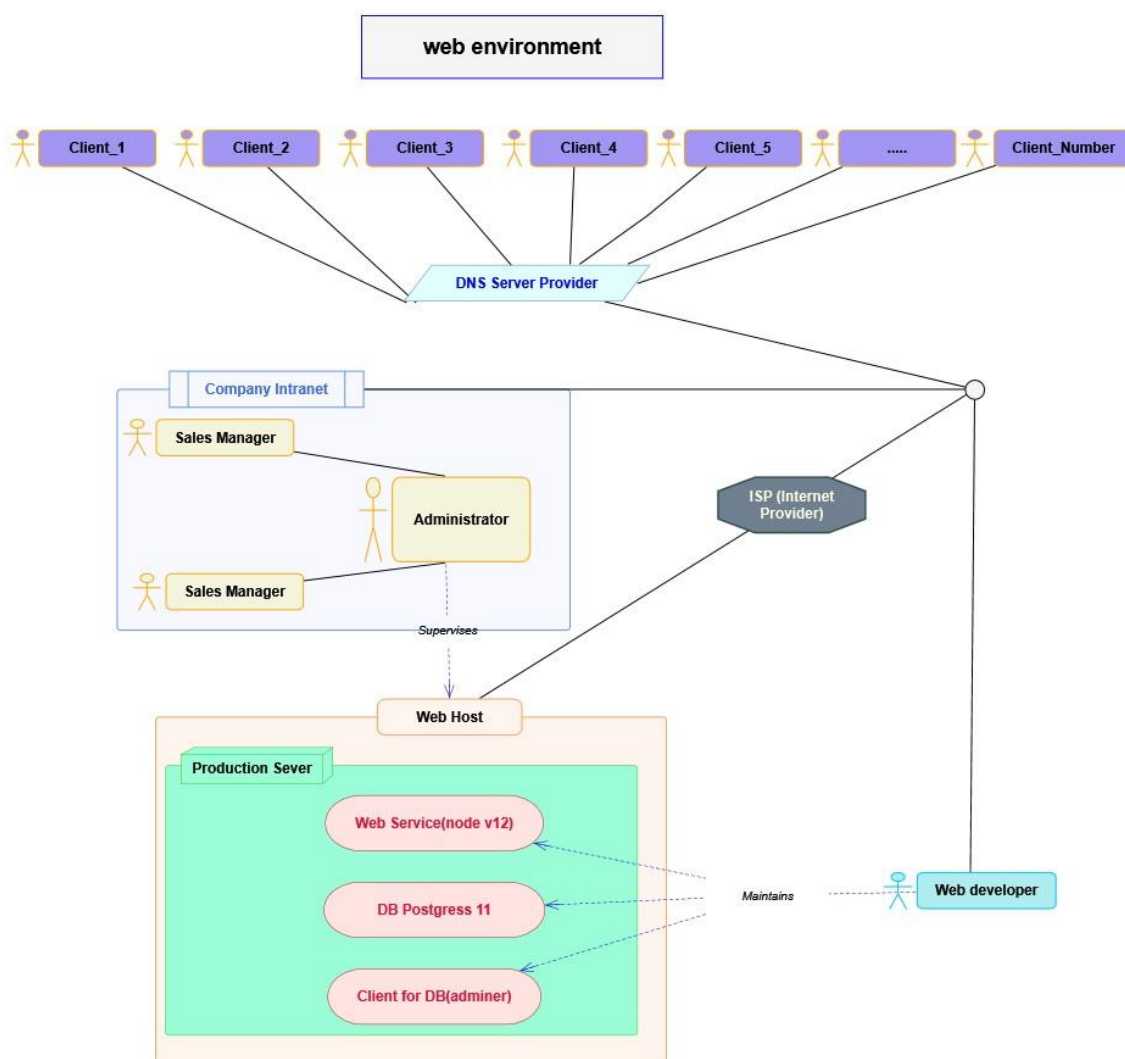


Рисунок 1.3.4.1 – Диаграмма развертывания

1.4 Функциональная модель системы

В данном разделе представлена общая функциональная схема работы системы, представленная двумя диаграммами – общей контекстной диаграммой и диаграммой декомпозиции. Контекстная диаграмма и диаграмма декомпозиции составлены по методологии IDEF0.

IDEF0 – это методология функционального моделирования в которой рассматриваются логические отношения между функциями а не их временная последовательность. В центре контекстной диаграммы указывается главная функция, которую выполняет исполнитель, который может являться как физическим, так и юридическим лицом. Стрелка входящая в верхнюю кромку главного функционального блока называется стрелкой *управления*. Стрелки управления обозначают управляющие механизмы, которые оказывают регулятивное воздействие на главную функцию. Стрелка, приходящая в нижнюю кромку, называется стрелкой *механизмом*. Стрелка механизм обозначает то, что используется, чтобы произвести необходимую работу. Стрелки, приходящие в левую кромку, называют стрелками *входа*. Стрелки входа обозначают данные подающиеся на вход главной функции. Стрелки, исходящие от правой кромки, называют *исходящими* стрелками. Исходящие стрелки обозначают результат выполнения главной функции, исходящие стрелки выводят результаты деятельности.

Главный блок контекстной диаграмма представляет собой чёрный ящик, сложность реализации которого скрыта от наблюдателя. У главного блока может быть множество входов и выходов, а также несколько управляющих механизмов и регуляторов. Все входы и выходы могут быть декомпозированы до необходимого уровня на диаграмме декомпозиции.

Контекстная диаграмма отображена на рисунке 1.4.4.1.

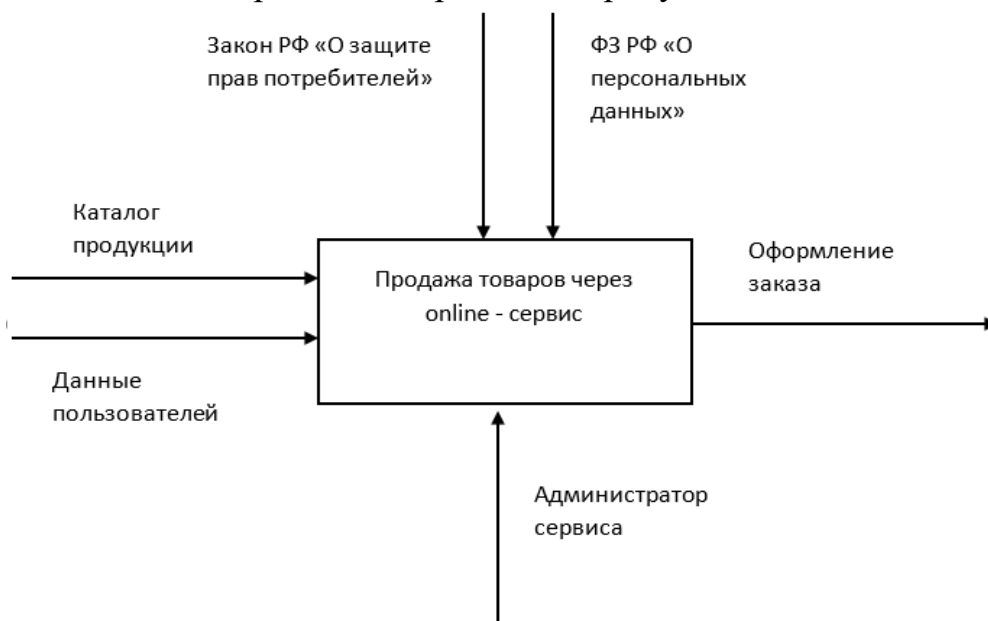


Рисунок 1.4.4.1 – Контекстная диаграмма

Диаграмма декомпозиции отображена на рисунке 1.4.4.2.

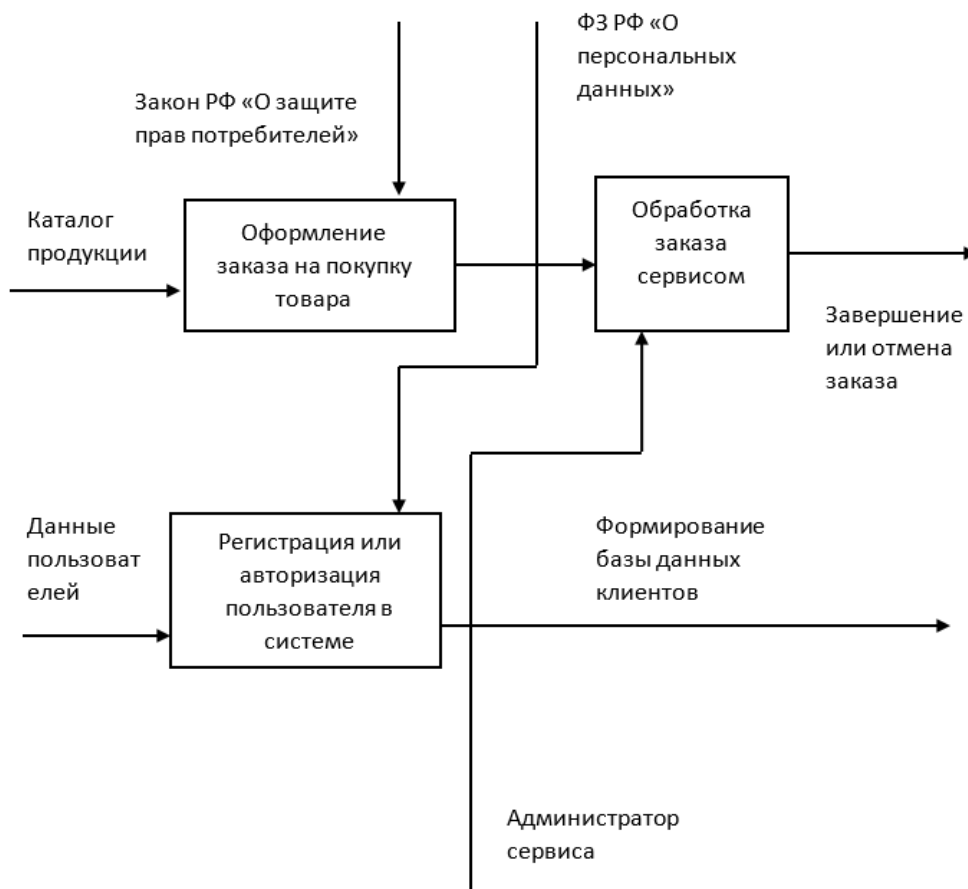


Рисунок 1.4.4.2 – Диаграмма декомпозиции

2 ИССЛЕДОВАНИЕ АНАЛОГОВ

В качестве аналогов для исследования были выбраны следующие торговые интернет платформы: интернет магазин PC-1, интернет-магазин электроники Zakazrsc.ru, интернет-магазин по продаже компьютерной техники и программного обеспечения «МАКСИМУМ техноцентр».

2.1 Интернет-магазин PC-1

Интернет-магазин PC-1 осуществляет торговлю в основном б/у - ассортиментом товаров. На главной странице магазина указано, что товары периодически обновляются и добавляются новые позиции в каталог. Как утверждает магазин техника проходит предпродажную подготовку и полностью исправна. Внешний вид магазина показан ниже на скриншотах (рисунок 2.1.1, рисунок 2.1.2, рисунок 2.1.3).

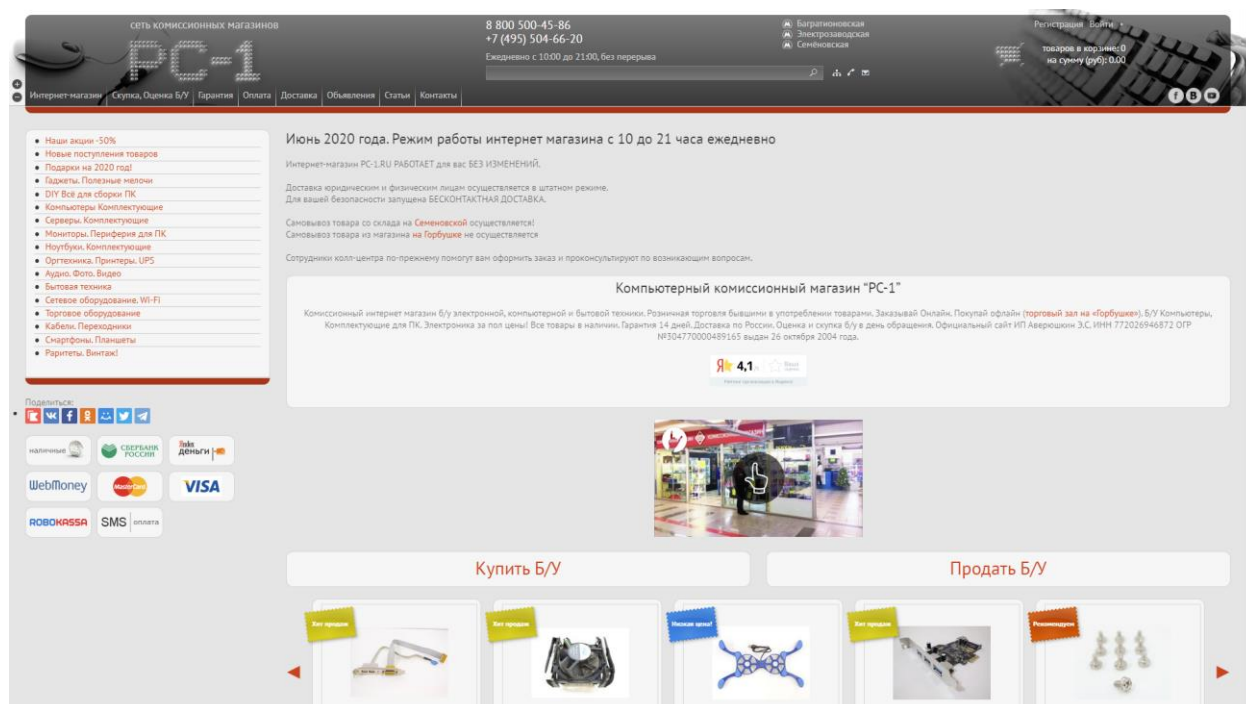


Рисунок 2.1.1 – Главная страница магазина PC-1

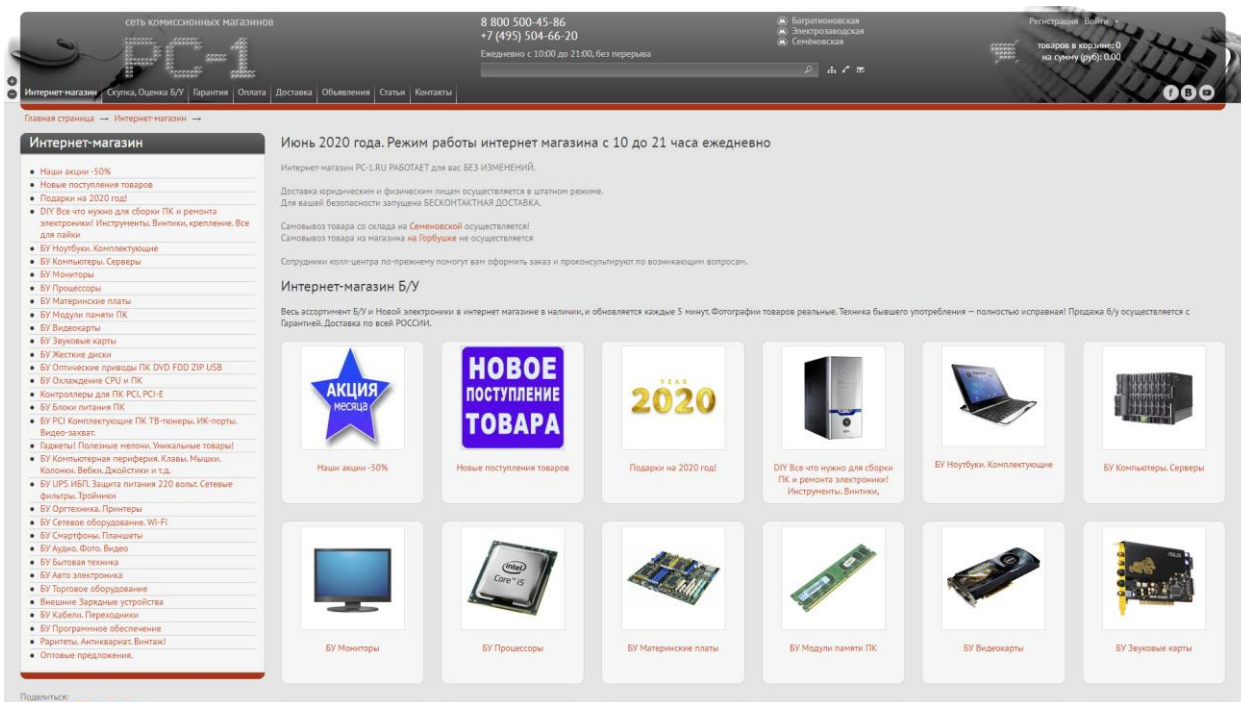


Рисунок 2.1.2 – Каталог категорий магазина PC-1

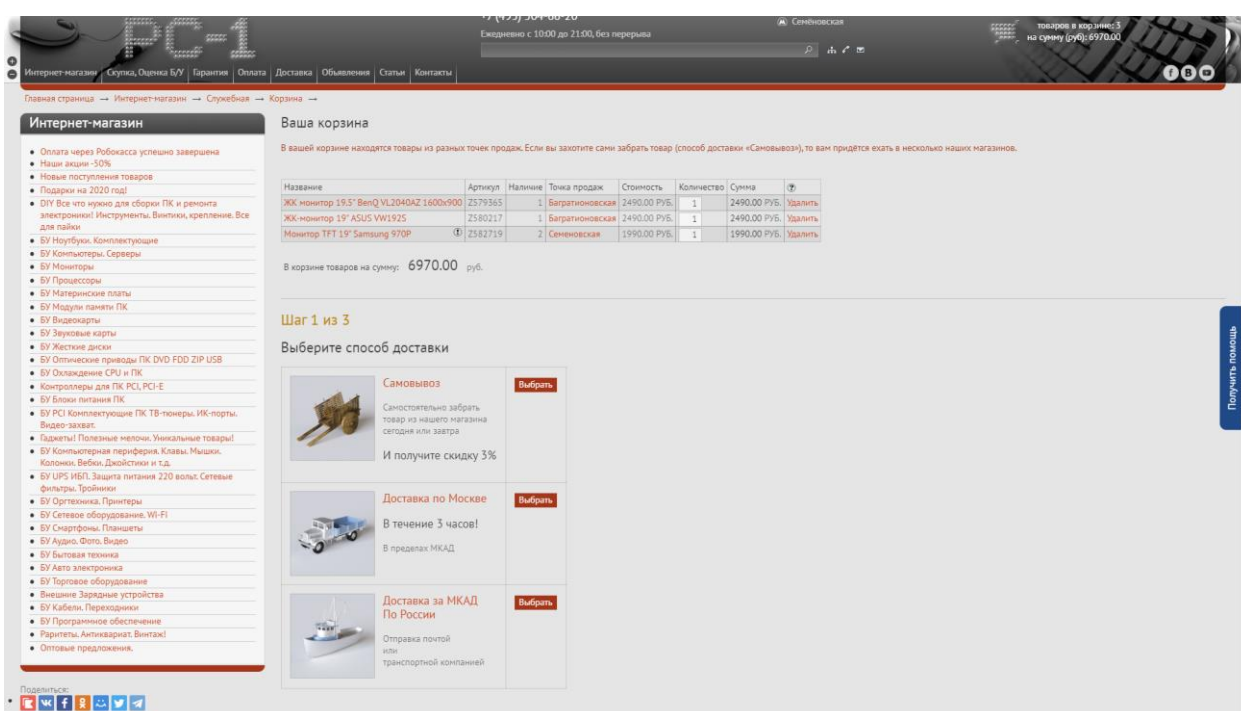


Рисунок 2.1.3 – Страница корзины магазина PC-1

2.2 Интернет-магазин электроники Zakazpc.ru

Интернет-магазин электроники Zakazpc.ru осуществляет торговлю различного рода комплектующими и расходниками для компьютерной техники, а также кабелями, переходниками и сетевым оборудованием.

Магазин уверяет, что клиенту оказывается послепродажное обслуживание, ремонт и настройка оборудования. Есть своя бригада выездных мастеров, которые по договоренности с клиентом могут выполнить различные виды пусконаладочных работ. Внешний вид магазина показан ниже на скриншотах (рисунок 2.2.1, рисунок 2.2.2, рисунок 2.2.3).



Рисунок 2.2.1 – Главная страница магазина Zakazpc.ru

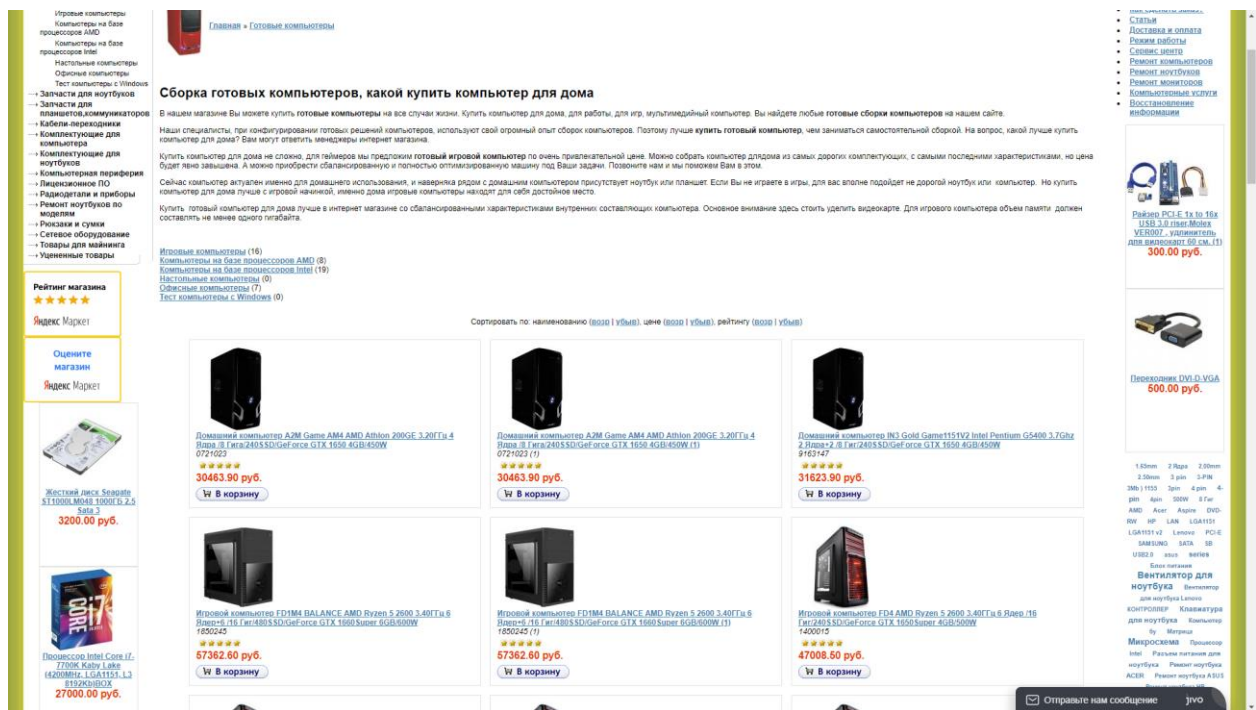


Рисунок 2.2.2 –Каталог компьютеров магазина Zakazrsc.ru

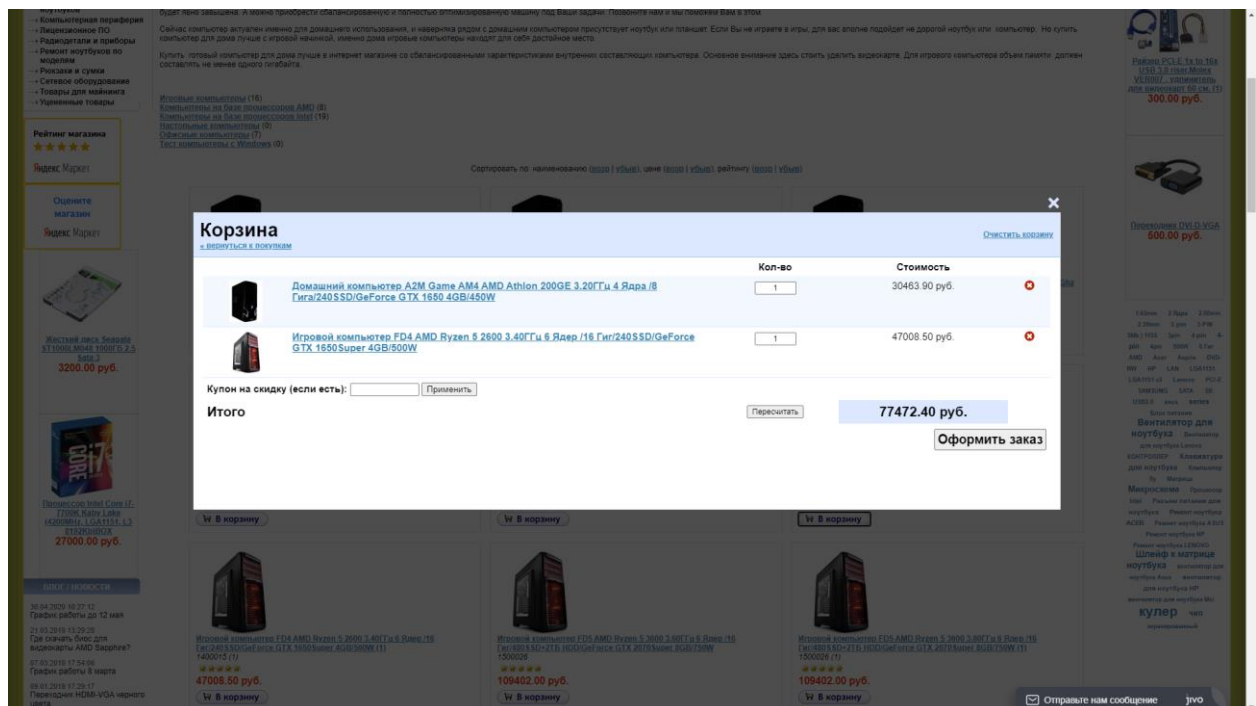


Рисунок 2.2.3 – Корзина магазина Zakazrsc.ru

2.3 Интернет-магазин по продаже компьютерной техники и программного обеспечения «МАКСИМУМ техноцентр»

Интернет-магазин «МАКСИМУМ техноцентр» осуществляет продажу различных видов и категории товаров, в их числе: автоэлектроника, бытовая техника, компьютеры, приемное оборудование, программное обеспечение, радиодетали и запчасти. Магазин осуществляет розничную продажу физическим лицам, юридическим лицам и индивидуальным предпринимателям. Внешний вид магазина показан ниже на скриншотах (рисунок 2.3.1, рисунок 2.3.2).



Рисунок 2.3.1 – Главная страница магазина «МАКСИМУМ техноцентр»

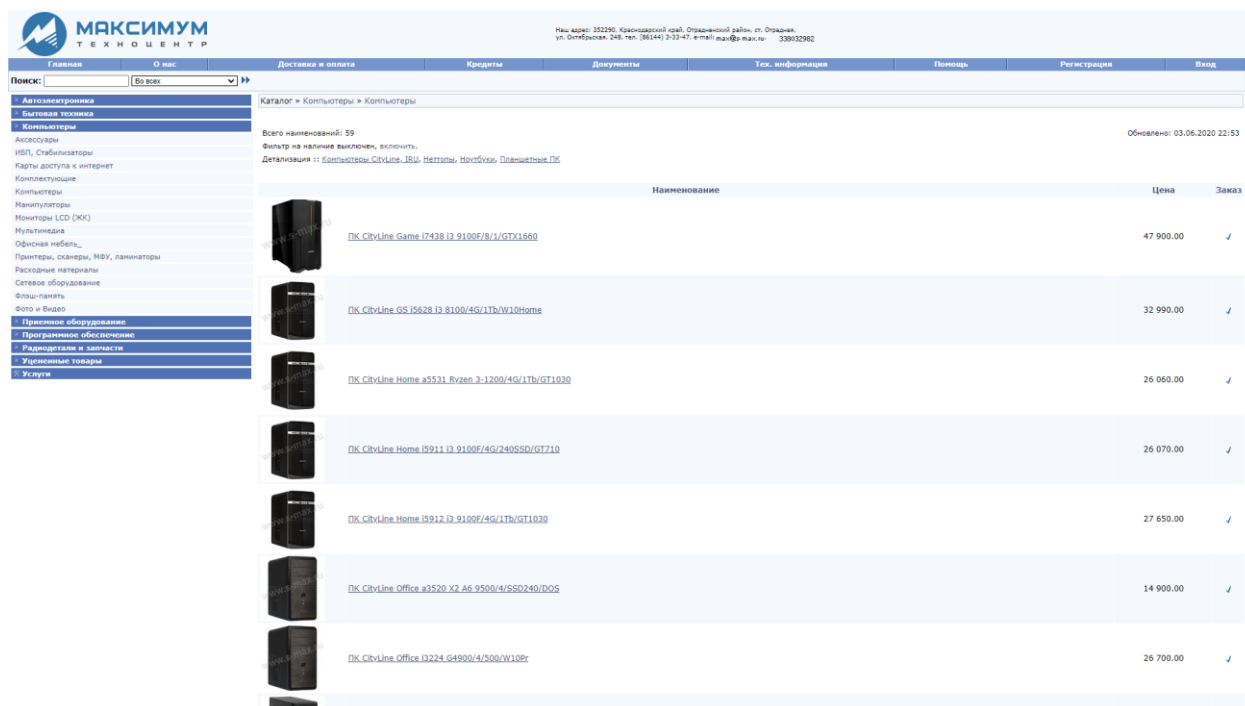


Рисунок 2.3.2 – Каталог компьютеров магазина «МАКСИМУМ техноцентр»

2.4 Сравнение систем аналогов

Сравнение систем аналогов представлено ниже на таблице 8.

Таблица 8 – Сравнительная таблица систем-аналогов

Выполняемые функции и критерии	Системы-аналоги		
	РС-1	Zakazpc.ru	«МАКСИМУМ техноцентр»
Возможность добавления товаров в пользовательскую корзину	Есть	Есть	Ссылка добавления товара не работает
Регистрация пользователей	Есть	Нет	Есть
Скорость загрузки страниц	Средняя	Средняя	Низкая

Продолжение таблицы 8

Удобство расположения элементов графического интерфейса	Каталог товаров представлен в виде плитки, это популярное и удобное решение	Товары представлены в виде плитки с ссылками на подробное описание	Каталог представлен в виде списка, такое решение является менее удобным чем плитка
Поддержка добавления множества товаров в корзину	Есть	Есть	Неизвестно, так как функционал корзины не был доступен
Удобство навигации по сайту	Левый side-бар и навигационная панель сверху, достаточно стандартное решение	Достаточно маленький левый side-бар, и навигации прямо в шапке гл. страницы, что не слишком удобно	Левый side-бар и навигационная панель сверху, но шрифт выбран достаточно мелкий, что не очень удобно
Нотификации и подсказки пользователю	Всплывающих нотификаций нет	Всплывающих нотификаций нет	Всплывающих нотификаций нет
Удобство использования ресурса в целом	Ресурс является достаточно удобным в плане использования	Ресурс является средним по удобству использования	Ресурс является не очень удобным в плане использования

В результате проведенного анализа систем аналогов можно прийти к выводу, что все системы имеют какие – либо выраженные недостатки, отраженные в таблице 9, но при этом предоставляют необходимый функционал для их использования конечным потребителем.

3 ПРОЕКТИРОВАНИЕ СИСТЕМЫ

3.1 Назначение и цели создания системы

Данная система предназначена для реализации возможности автоматизации бизнес-процессов торгового предприятия. Среди которых: возможность регистрации клиента в системе и сохранения данных его профиля в базу данных, возможность оформления клиентом заказов и отслеживание заказанных товаров с помощью клиентского функционала web-приложения, возможность автоматизированного администрирования системы (добавление товаров, удаление товаров, удаление пользовательских профилей, изменение цен товаров и т.д.).

Цели создания системы: повышение удобства обслуживания клиентов заказчика, повышение экономической эффективности работы путем автоматизации процессов сбора и хранения информации о текущих заказах, завершенных заказах, и мониторинге текущего состояния хранилища доступных товаров [27].

3.2 Характеристика объекта автоматизации

Характеристика объекта автоматизации представлена в таблице 9.

Таблица 9 – Характеристика объектов автоматизации

Структурное подразделение	Наименование процесса	Возможность автоматизации	Решение об автоматизации в ходе проекта
Отдел торговли	Мониторинг текущих заказов товаров	Возможна	Будет автоматизирован

Окончание таблицы 9

Отдел торговли	Мониторинг текущего состояния базы доступных для заказа товаров	Возможна	Будет автоматизирован
Отдел торговли	Своевременное добавление новых товаров в базу данных доступных товаров	Возможна	Будет автоматизирован
Отдел доставки	Мониторинг статусов текущих заявок на доставку товаров	Возможна	Будет автоматизирован
Отдел доставки	Оценка количества текущих открытых заявок на доставку	Возможна	Будет автоматизирован

3.3 Требования к системе

Система должна быть централизованной, т.е. все данные должны располагаться в центральном хранилище. Система должна иметь трехуровневую архитектуру: уровень пользовательского интерфейса, уровень серверной бизнес-логики, уровень базы данных [14]. В системе предлагается выделить следующие функциональные подсистемы:

- подсистема сбора, обработки и загрузки данных, которая предназначена для реализации процессов сбора данных из систем источников, приведения указанных данных к виду, необходимому для наполнения подсистемы хранения данных [31];

- подсистема хранения данных, которая предназначена для хранения данных в структурах в виде реляционных моделей;
- подсистема формирования и визуализации данных и текущего состояния системы в web-интерфейсе [26].

Реализация обмена данными в системе должна базироваться на применении транспортного протокола прикладного уровня HTTP.

4 РАЗРАБОТКА СИСТЕМЫ

Данный раздел описывает разработку клиентской части приложения с использованием библиотек React и Redux, серверной части приложения с использованием фреймворков NodeJS и Sequelize, библиотек Koa2, Koa2-Router и Passport, системой управления базами данных PostgreSQL 11, алгоритмы работы приложения, а также программное и техническое обеспечение.

4.1 Общие сведения

Покупки в интернете в настоящий момент являются формой электронной коммерции, которая позволяет напрямую взаимодействовать покупателям и продавцам, осуществляя покупки и продажи в сети с помощью веб-браузера [3]. Потребители, осуществляя поиск нужных им товаров в сети интернет, имеют большой выбор поставщиков, которые готовы продавать свои товары и услуги через Интернет напрямую или через посредников. Также осуществляя поиск нужных товаров и услуг в сети, мы имеем возможность эффективно сравнивать цены и условия у различных поставщиков этих услуг, где бы они не находились [20]. В настоящее время совершать покупки в сети можно практически с любого типа устройств, которые имеют возможность подключения к сети Интернет.

Покупка товаров в интернет-магазине похожа на покупку в обычном магазине, т.е. когда напрямую взаимодействуют продавец и покупатель. Этот процесс называется - «бизнес-потребитель» (B2C) [2]. Когда компания, занимающаяся онлайн торговлей, настроена на оптовые продажи или специфика ее продаж такова что невозможно продать товар конечному потребителю, то этот процесс называется «бизнес-бизнес» (B2B), т.е. когда одни компании продают свои товары или услуги другим компаниям. Стандартный интернет-магазин предоставляет покупателю возможность

поиска товаров или услуг на сайте, поиск по разным критериям, наличию товара на складе, цене, популярности среди других покупателей и т.д. [11].

Наиболее крупными и известными онлайн-магазинами являются Alibaba, Amazon.com и eBay [23].

4.2 Выбор и обоснование программного обеспечения

Раздел описывает набор программных средств и технологический стек, на котором будет строиться данная автоматизированная система.

4.2.1 Система управления базами данных

Для функционирования автоматизированной системы требуется система управления базами данных - СУБД. В качестве целевой платформы для хранения и управления данными была выбрана свободная объектно-реляционная система управления базами данных – PostgreSQL. PostgreSQL базируется на языке SQL и поддерживает многие из возможностей стандарта SQL:2011. В PostgreSQL версии 11 есть ограничения, представленные в таблице 10.

Таблица 10 – Технические ограничения базы данных

Название ограничение	Значение ограничения
Максимальный размер базы данных	Нет ограничений
Максимальный размер таблицы	32 Тбайт
Максимальный размер поля	1 Гбайт
Максимум записей в таблице	Ограничено размерами таблицы
Максимум полей в записи	250-1600, в зависимости от типов полей
Максимум индексов в таблице	Нет ограничений

Общая визуальная схема таблиц базы данных представлена на рисунке 4.2.1.1.

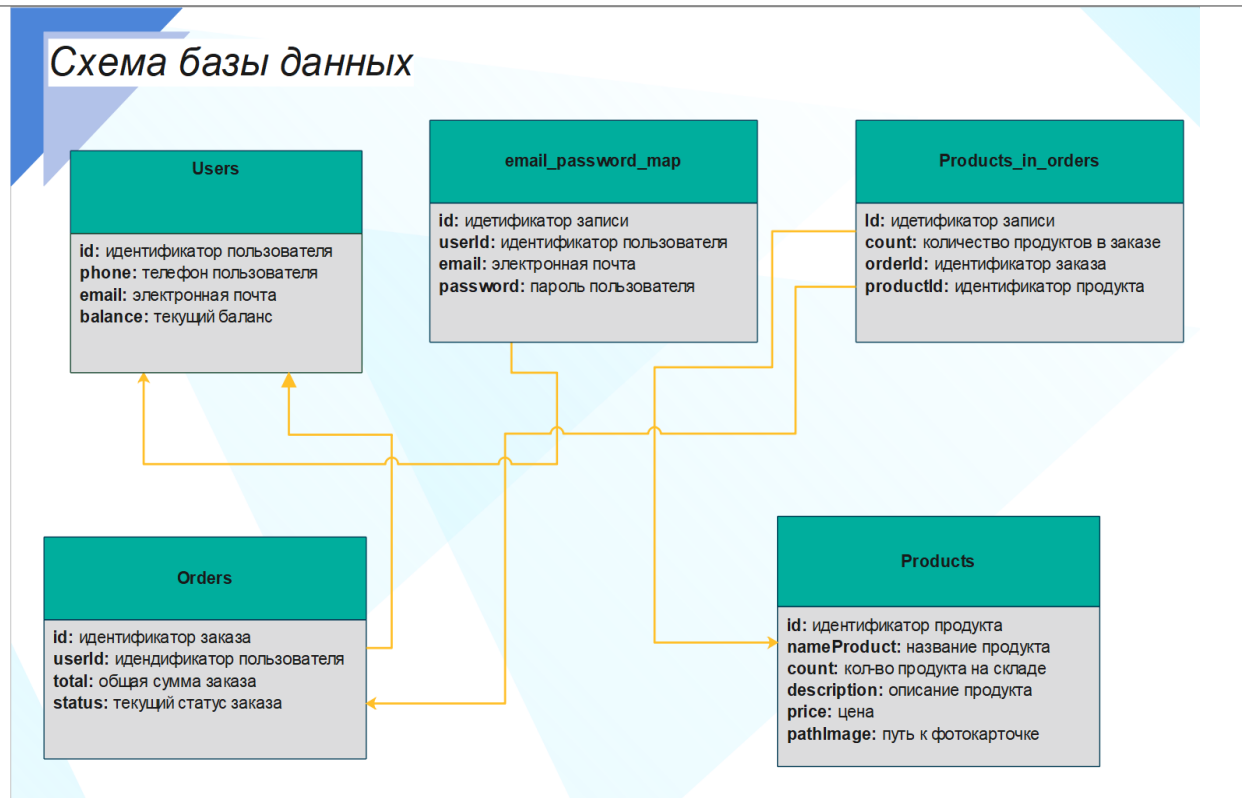


Рисунок 4.2.1.1 – Схема построения базы данных

Сильными сторонами PostgreSQL считаются:

- высокопроизводительные и надёжные механизмы транзакций и репликации;
- расширяемая система встроенных языков программирования: в стандартной поставке поддерживаются PL/pgSQL, PL/Perl, PL/Python и PL/Tcl; дополнительно можно использовать PL/Java, PL/PHP, PL/Py, PL/R, PL/Ruby, PL/Scheme, PL/sh и PL/V8, а также имеется поддержка загрузки модулей расширения на языке C [13];
- наследование;
- возможность индексирования геометрических объектов и наличие базирующегося на ней расширения PostGIS;
- встроенная поддержка слабоструктурированных данных в формате JSON с возможностью их индексации;

- расширяемость (возможность создавать новые типы данных, типы индексов, языки программирования, модули расширения, подключать любые внешние источники данных).

Согласно результатам автоматизированного исследования, различного ПО на предмет ошибок, проведенном в 2005 году, в исходном коде PostgreSQL было найдено 20 проблемных мест на 775000 строк исходного кода (в среднем, одна ошибка на 39000 строк кода) [25]. Для сравнения: MySQL – 97 проблем, одна ошибка на 8000 строк кода; FreeBSD (целиком) – 306 проблем, одна ошибка на 2500 строк кода; Linux (только ядро) – 950 проблем, одна ошибка на 800 строк кода.

4.2.2 Программирование серверной части

Для разработки серверной части приложения была выбрана платформа *Node* - программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения [15]. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода [1, 24].

Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения (при помощи NW.js, AppJS или Electron для Linux, Windows и macOS) и даже программировать микроконтроллеры (например, *tessel*, *low.js* и *espruino*) [28].

В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом [19]. Общая схема работы веб-сервера показана ниже (рис. 4.2.2.1)

Для маршрутизации запросов внутри веб-сервера был применен фреймворк *Koa2*. На сайте разработчиков этого фреймворка представлено следующее описание данной технологии: *Koa* - это новая веб-инфраструктура, разработанная командой разработчиков *Express*, целью которой является создание более компактной, выразительной и более надежной основы для веб-приложений и API. Используя асинхронные функции, *Koa* позволяет отказаться от обратных вызовов и значительно улучшить обработку ошибок [16]. *Koa* не связывает какое-либо промежуточное программное обеспечение в своем ядре и предоставляет элегантный набор методов, которые делают запись серверов быстрой и приятной. Основная суть работы этого фреймворка заключается в использовании функции промежуточной обработки запросов так называемый – *Middleware* (рус. промежуточный слой) [29].

Middleware – обрабатывает запрос на разных его этапах в зависимости от контекста запроса и его типа [18]. Каждый вышестоящий *middleware* оборачивает нижестоящий, для вызова, следующего *middleware* используется функция *next* [30]. Схема работы *middlewares* показана ниже (рис. 4.2.2.2).

Общая схема работы серверной части приложения.

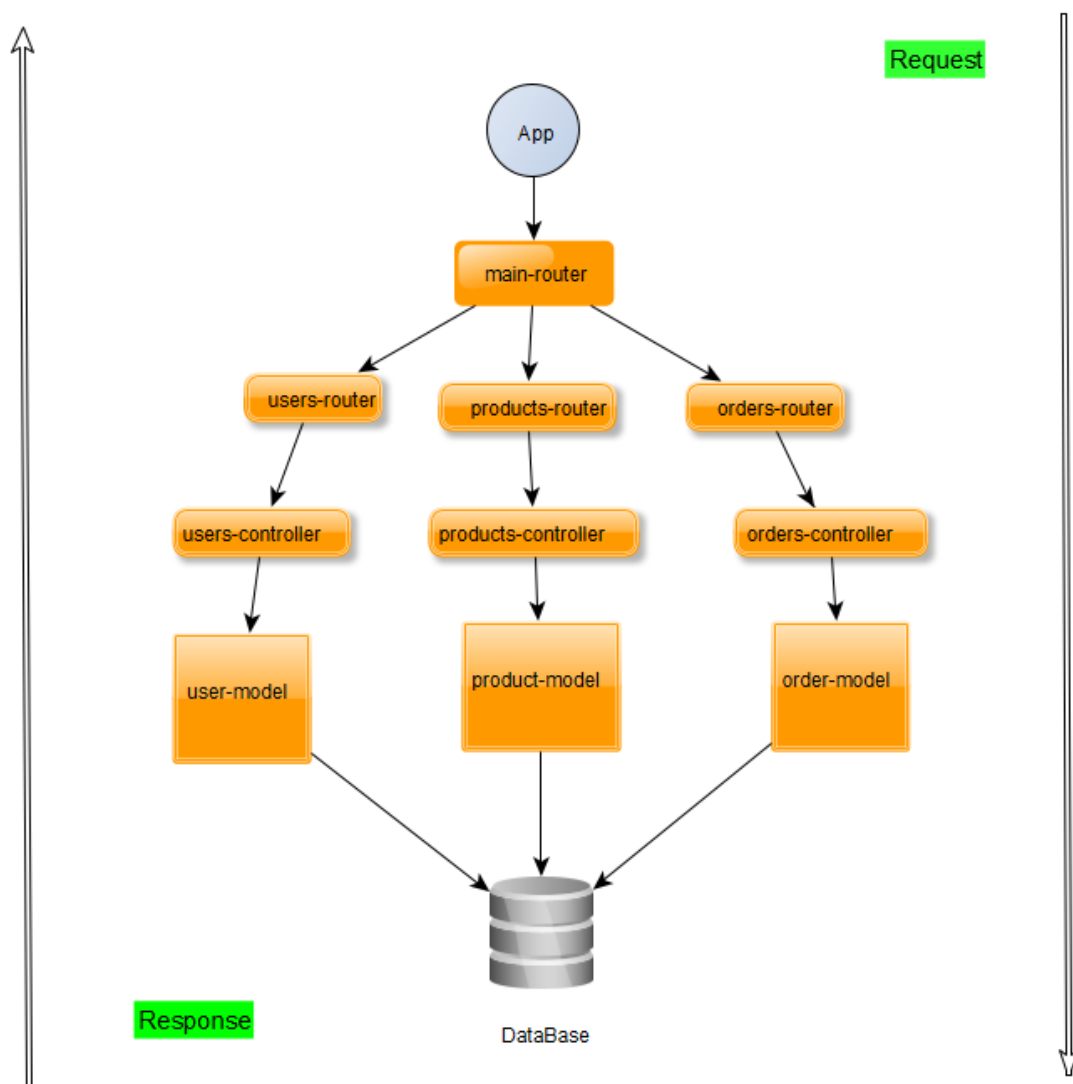


Рисунок 4.2.2.1 – Схема внутренней организации web-сервера

Общая схема работы промежуточных обработчиков серверной части приложения.

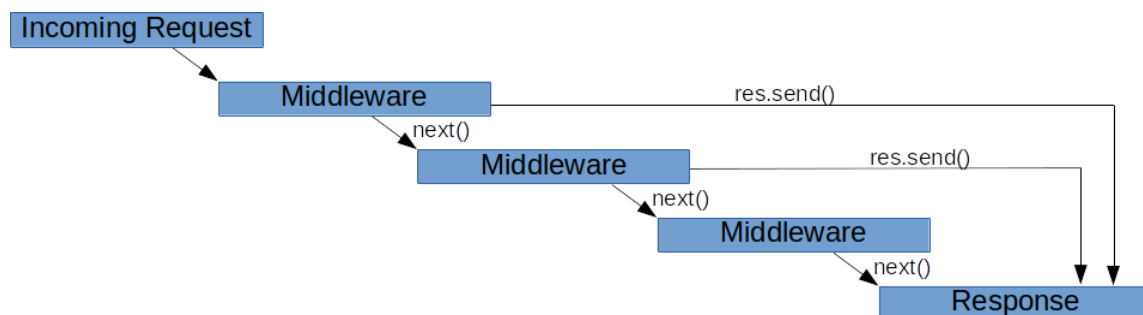


Рисунок 4.2.2.2 – Визуализация работы middlewares

4.2.3 Программирование клиентской части

Для разработки клиентской части приложения была выбрана библиотека *React* в связке с библиотекой *Redux*. А также ряд других библиотек, реализующих отдельные функциональные компоненты приложения. *React* (иногда *React.js* или *ReactJS*) — JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов [8].

React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций.

React может использоваться для разработки одностраничных и мобильных приложений. Его цель – предоставить высокую скорость, простоту и масштабируемость. В качестве библиотеки для разработки пользовательских интерфейсов *React* часто используется с другими библиотеками, такими как *Redux* и *GraphQL*. *React* использует виртуальный DOM (англ. *virtual DOM*). *React* создаёт кэш-структуру в памяти, что позволяет вычислять разницу между предыдущим и текущим состояниями интерфейса для оптимального обновления DOM браузера [22]. Таким образом, программист может работать со страницей, считая, что она обновляется вся, но библиотека самостоятельно решает, какие компоненты страницы необходимо обновить [21].

Redux – библиотека для JavaScript с открытым исходным кодом, предназначенная для управления состоянием приложения [7]. Чаще всего используется в связке с *React* или *Angular* для разработки клиентской части. Содержит ряд инструментов, позволяющих значительно упростить передачу данных хранилища через контекст. Внешний вид и стилизация приложения были оформлены с помощью формального языка описания внешнего вида документа CSS версии 3 и языка разметки гипертекста HTML версии 5[6].

Ключевой особенностью библиотеки *React* является жизненный цикл компонента приложения. Жизненный цикл компонента представляет собой

набор конечных состояний некоторой части виртуального DOM-дерева используемого внутри библиотеки, которые напрямую влияют на то что пользователь видит в текущий момент на странице в браузере.

Виртуальный DOM (VDOM) – это концепция программирования, в которой идеальное или «виртуальное» представление пользовательского интерфейса хранится в памяти и синхронизируется с «настоящим» DOM при помощи библиотеки, такой как ReactDOM. Этот процесс называется согласованием.

Такой подход и делает API React декларативным: вы указываете, в каком состоянии должен находиться пользовательский интерфейс, а React добивается, чтобы DOM соответствовал этому состоянию. Это абстрагирует манипуляции с атрибутами, обработку событий и ручное обновление DOM, которые в противном случае пришлось бы использовать при разработке приложения.

Поскольку «виртуальный DOM» – это скорее паттерн, чем конкретная технология, этим термином иногда обозначают разные понятия [12]. В мире React «виртуальный DOM» обычно ассоциируется с React-элементами, поскольку они являются объектами, представляющими пользовательский интерфейс. Тем не менее, React также использует внутренние объекты, называемые «волоконнами» (fibers), чтобы хранить дополнительную информацию о дереве компонентов. Их также можно считать частью реализации «виртуального DOM» в React.

Диаграмма жизненного цикла компонента в React и общая схема работы библиотеки представлены на рисунках 4.2.3.1 и 4.2.3.2.

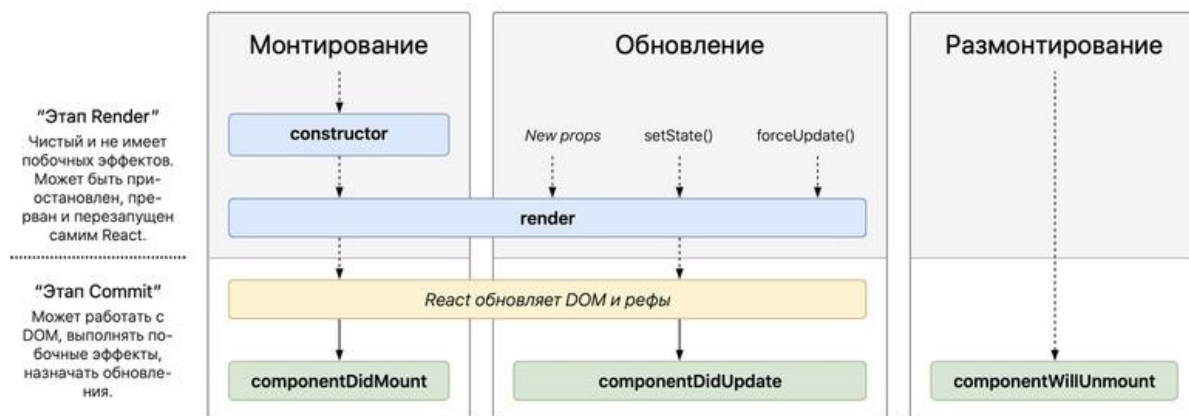


Рисунок 4.2.3.1 – Жизненный цикл компонента

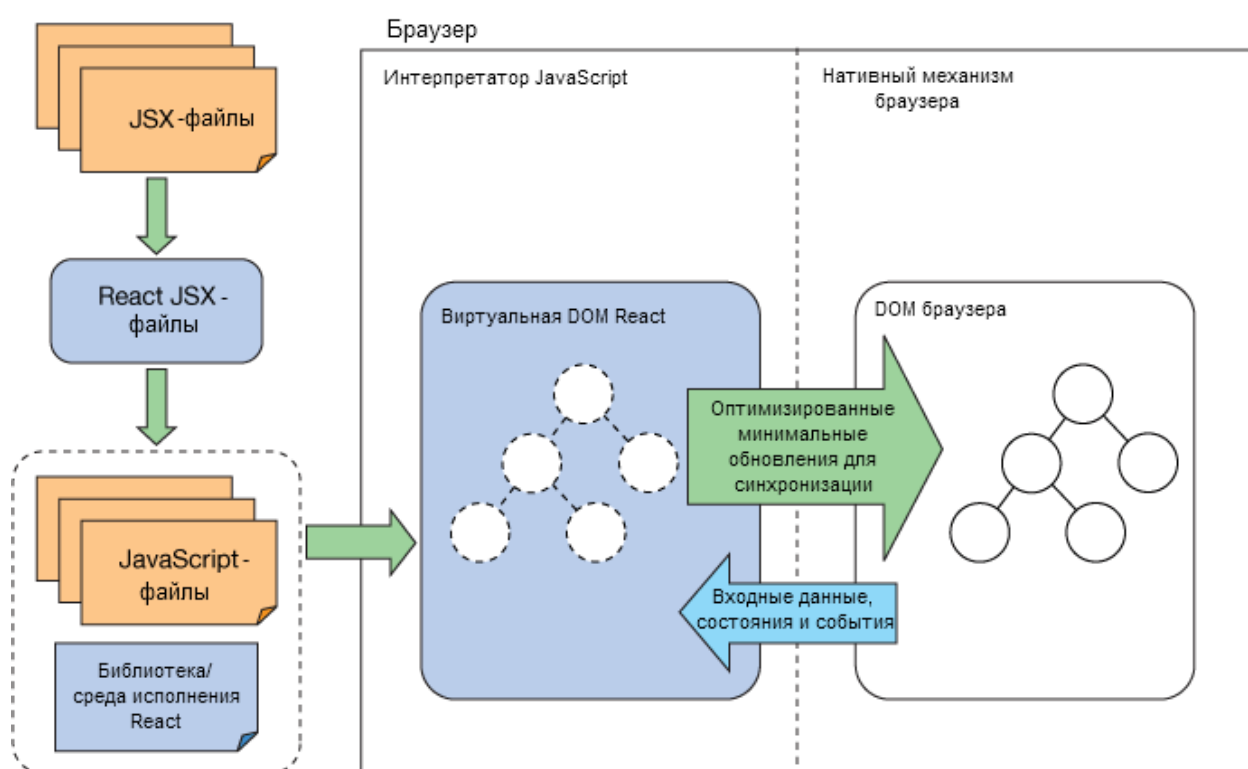


Рисунок 4.2.3.2 – Работа библиотеки React

5 ДОКУМЕНТАЦИОННОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

5.1 Технические требования для выполнения АС

Данный раздел содержит руководство пользователя и администратора системы, а также описание возможностей АС. Система предназначена для применения в сфере торговли и электронной коммерции. Предусмотрена регистрация и авторизация пользователей, добавление и удаление товаров, формирование пользовательской корзины, оформление заказов клиента, получение уведомлений о статусе заказа, и т.д.

Система может работать со всеми со следующими операционными системами: Windows 10 Professional, любой современный дистрибутив ОС Linux, Mac OS. В качестве рабочих станций должны применяться IBM совместимые персональные компьютеры.

Минимальная конфигурация для запуска серверной части приложения:

- 64-разрядный (x64) процессор с тактовой частотой 2 ГГц или выше, с поддержкой программной или аппаратной виртуализации и трансляции виртуальных адресов оперативной памяти;
- 8 Гб или более ОЗУ;
- HDD или SSD объемом 20 Гб и выше;
- Сетевая карта, подключенная к сети Internet;
- Сетевой адаптер должен иметь пропускную способность не ниже, чем 100Мбит\с.

Клиентская часть приложения запускается в браузере пользователя и не предъявляет серьезных требований по аппаратной конфигурации ПК.

Для успешного запуска клиентской части требуется современный веб-браузер, желательно разработанный на браузерном движке Chromium.

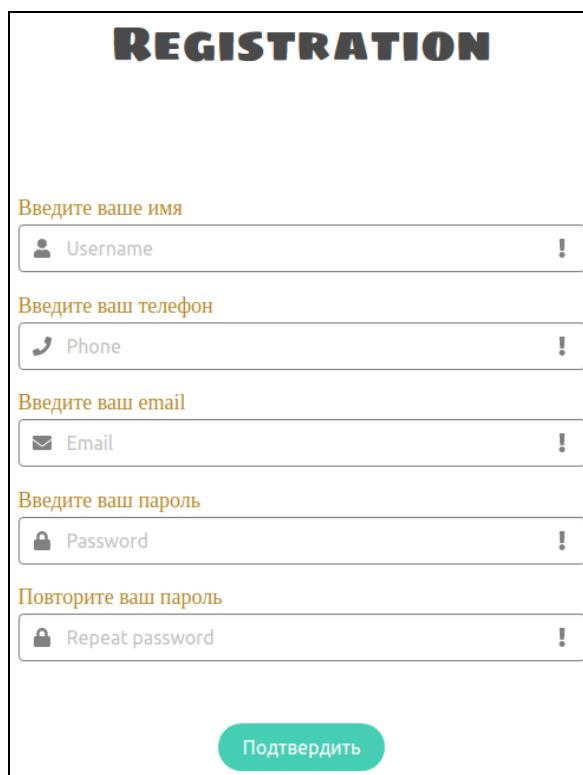
Клиентское приложение протестировано и полностью работоспособно в следующих веб-браузерах: Google Chrome версия 81, Microsoft Edge версия 83, Mozilla Firefox версия 76.

5.2 Руководство пользователя

В этом разделе описывается пользовательский интерфейс приложения и сценарии работы с интерфейсом.

5.2.1 Регистрация и авторизация

Форма регистрации в системе представлена на рисунке 5.2.1.



The image shows a registration form titled "REGISTRATION" in bold black letters. Below the title, there are five input fields, each with a label in orange text above it and a placeholder text in gray inside the field. The fields are: 1. "Введите ваше имя" (Enter your name) with a person icon and placeholder "Username". 2. "Введите ваш телефон" (Enter your phone) with a phone icon and placeholder "Phone". 3. "Введите ваш email" (Enter your email) with an envelope icon and placeholder "Email". 4. "Введите ваш пароль" (Enter your password) with a lock icon and placeholder "Password". 5. "Повторите ваш пароль" (Repeat your password) with a lock icon and placeholder "Repeat password". Each field has a small exclamation mark icon on the right side. At the bottom of the form, there is a green button with the text "Подтвердить" (Confirm) in white.

Рисунок 5.2.1 – Форма регистрации

Если пользователь уже зарегистрирован, он может перейти по ссылке на страницу авторизации, нажав на кнопку *Log In* в правом верхнем углу навигационной панели. Чтобы авторизоваться на ресурсе, пользователь должен ввести в форму своей email, который одновременно является уникальным логином пользователя на ресурсе и был указан при регистрации. Затем ниже ввести свой пароль, который также был указан при регистрации пользователя.

Форма для авторизации пользователя, представлена на рисунке 5.2.2.



The image shows a 'SIGN IN' form. At the top, the text 'SIGN IN' is displayed in a bold, dark blue font. Below this, there are two input fields. The first field is labeled 'Введите ваш email' (Enter your email) in a light blue font. It contains the text 'joo@google.com' and has a small blue checkmark icon on the right. The second field is labeled 'Введите ваш пароль' (Enter your password) in a light blue font. It contains a series of dots and has a small blue checkmark icon on the right. Below these fields is a green button with the text 'Подтвердить' (Confirm) in white.

Рисунок 5.2.2 – Форма авторизации пользователя

В формах для авторизации и регистрации при вводе данных происходит проверка корректности вводимых пользователем данных, например, пароль не должен быть короче 6 символов, или поле с именем не должно быть пустым, или поле email должно содержать символ - @ (коммерческое at) и символ точки.

Если пользователь ввел некорректный пароль или email в интерфейсе текущей страницы, в правом верхнем углу появится нотификация красного цвета, сообщающая, что были введены некорректные авторизационные данные.

Эта нотификация может появиться также в том случае, когда пользователь с таким логином не был найден в базе зарегистрированных пользователей. В этом случае необходимо сначала пройти процесс регистрации и затем заново повторить авторизацию.

Пользователь не будет переадресован на другие страницы, недоступные неавторизованным пользователям, пока успешно не завершит процесс авторизации.

Если авторизация прошла успешно пользователь получит сообщющую об этом нотификацию, которая появится в правом верхнем углу страницы. Пользовательский интерфейс изменится.

В правой части навигационной панели появится имя текущего пользователя, являющееся навигационным компонентом, содержащем ссылку на страницу профиля текущего пользователя. Слева от имени текущего пользователя располагается мини-аватар пользователя. Справа от имени пользователя располагается кнопка, отвечающая за выход пользователя из текущего аккаунта (Log out). Справа от кнопки Log out располагается навигационный компонент в виде корзины, переадресующий пользователя на страницу с корзиной товаров (пользовательская корзина сохраняет состояние в пределах текущей сессии, поэтому сразу после авторизации корзина будет пустая).

5.2.2 Взаимодействие с каталогом товаров и корзиной

Вне зависимости, от того авторизован пользователь или нет, остается доступной страница просмотра каталога товаров. Но неавторизованный пользователь не может добавлять товары в корзину и совершать заказ, ему доступен только просмотр списка товаров.

Пример страницы с каталогом товаров представлен на рисунке 5.2.6.

Авторизованный пользователь может добавлять необходимые ему товары нажатием кнопки «Добавить в корзину» на карточке товара. После добавления товара в корзину пользователь получает соответствующее уведомление с названием добавленного к пользовательской корзине товара.

Затем пользователь может сделать заказ на странице с пользовательской корзиной.

Пример добавления товара в пользовательскую корзину (рисунок 5.2.7)

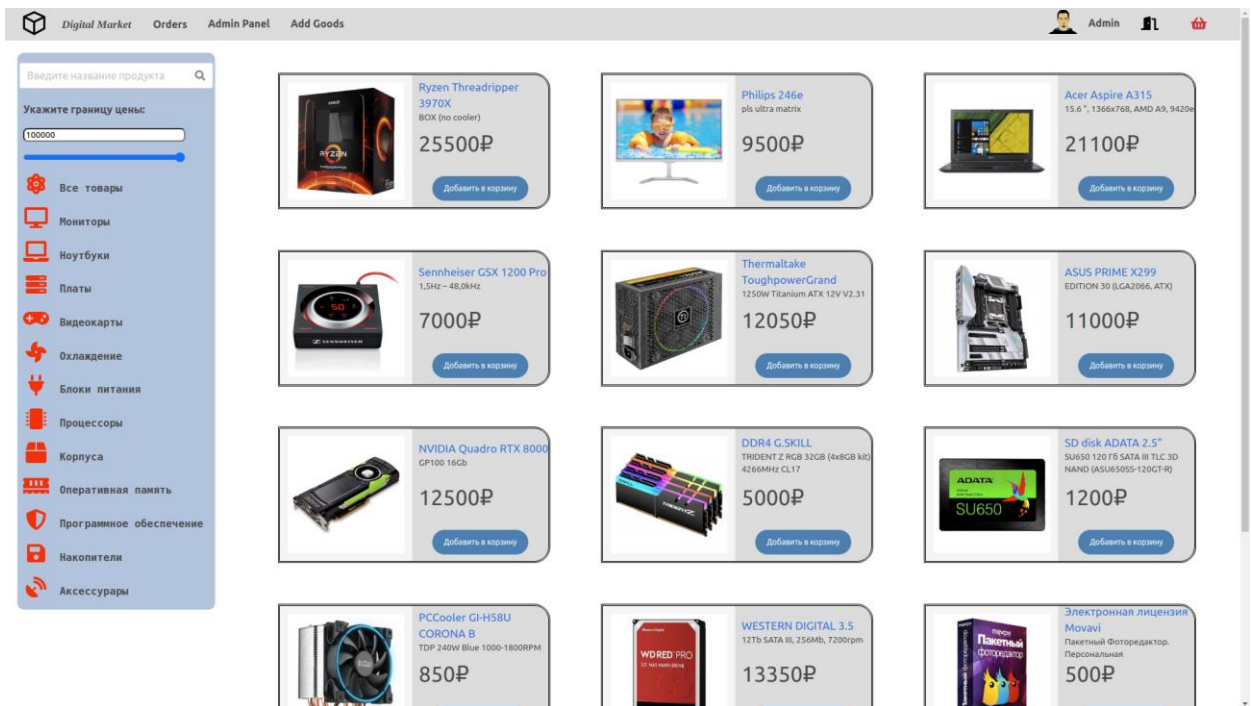


Рисунок 5.2.6 – Каталог товаров



Рисунок 5.2.7 – Добавление товара в корзину

Если кликнуть на ссылку с названием товара на карточке, то откроется отдельная страница с подробным описанием этого товара, количеством товара, которое доступно для заказа и увеличенным изображением. Также на странице будет продублирована кнопка для добавления товара в корзину.

Примеры страницы отдельного товара представлены на рисунках 5.2.8 и 5.2.9.

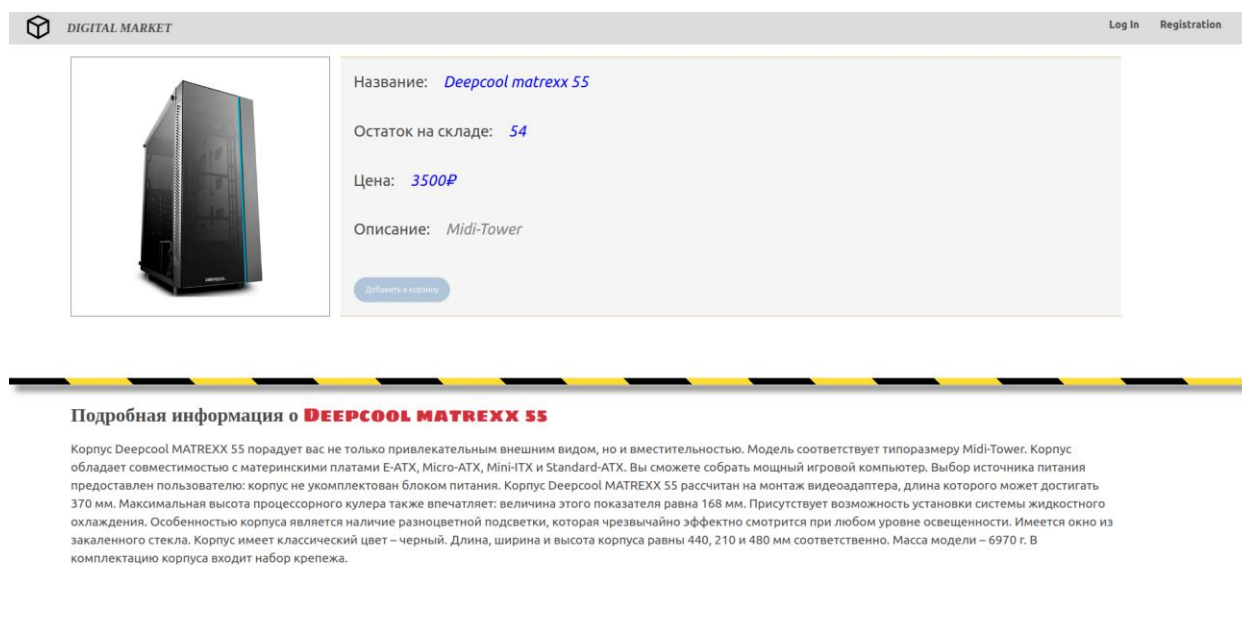


Рисунок 5.2.8 – Подробная информация о товаре

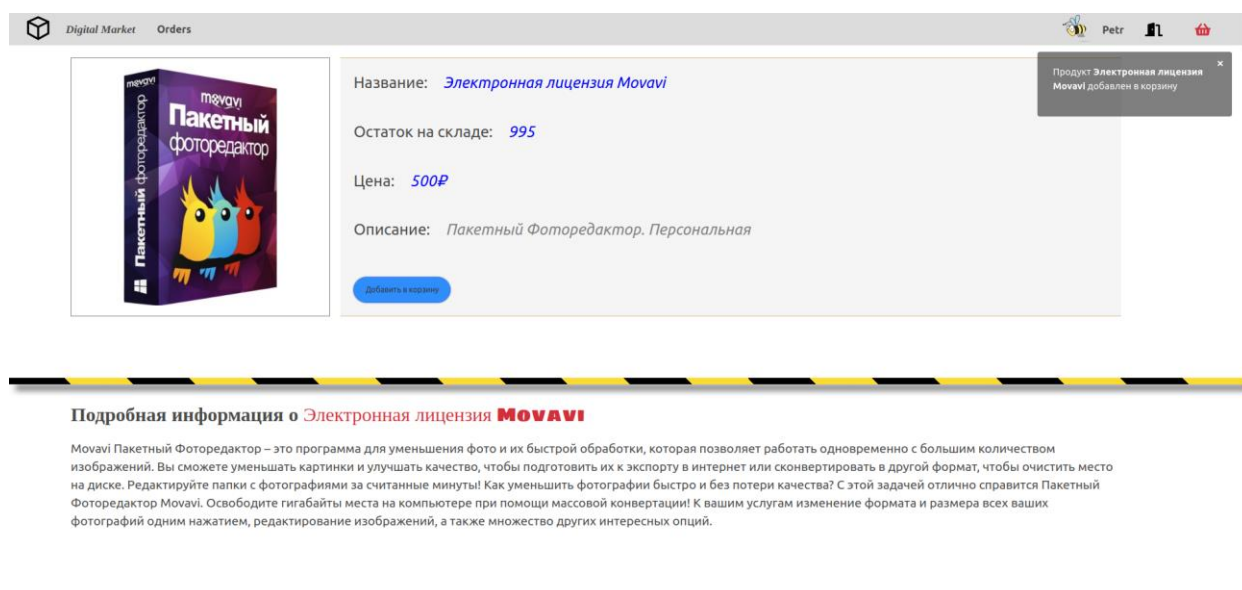


Рисунок 5.2.9 – Добавление товара в корзину

На странице с корзиной товаров, добавленных пользователем, выводится список добавленных товаров, количество экземпляров конкретного товара (если пользователь несколько раз совершил добавление товара в корзину – это считается за одну позицию в заказе с отображением

количества штук в данной позиции), общая сумма по конкретной позиции в корзине и суммарная стоимость всех позиций, входящих в корзину.

Чтобы подтвердить заказ пользователь должен нажать кнопку «ЗАКАЗАТЬ», появится соответствующая нотификация, сообщающая что заказ, был подтвержден.

Пример формы пользовательской корзины представлен на рисунке 5.2.10.





Количество	Название	Стоимость
<div>1</div>	<div>Электронная лицензия Movavi</div>	500₽
Сумма заказа 500₽		ЗАКАЗАТЬ

Рисунок 5.2.10 – Пример формы пользовательской корзины

Когда пользователь нажимает кнопку «ЗАКАЗАТЬ» то с его баланса списываются средства в размере суммы заказа, интерфейс пользовательской корзины очищается, а заказ переходит в стадию обработки.

5.2.3 Управление заказами

После подтверждения заказа пользователем, заказ добавляется в обработку. Данные заказа сохраняются в базу данных. По умолчанию новым заказам присваивается статус «В обработке». Увидеть свои текущие заказы и историю предыдущих заказов, пользователь может на странице списка заказов.

Если пользователь захочет отменить свой заказ, ему необходимо нажать на кнопку «ДЕЙСТВИЯ» и выбрать действие «Отменить». После

этого заказ приобретает статус «Отменен» и перемещается в секцию истории заказов.


Если заказ был успешно доставлен, пользователь может выбрать действие «Подтвердить». После этого заказ меняет свой статус на «Выполнен» и также попадает в историю предыдущих заказов.

Примеры заказов пользователя и действия с заказами представлены на рисунках 5.2.12, 5.2.13, 5.2.14.

Текущие заказы

ИНФОРМАЦИЯ О ЗАКАЗЕ

Код заказа: 1957861207



Acer Aspire A315

Количество1

Стоимость21100₽

Статус заказаВ обработке

Общая стоимость21100₽


ДЕЙСТВИЯ

Отменить заказ

Подтвердить получение

ИНФОРМАЦИЯ О ЗАКАЗЕ

Код заказа: 3306342244



Deepcool matrexx 55

Количество1

Стоимость3500₽

Статус заказаВ обработке

Общая стоимость3500₽

ДЕЙСТВИЯ


Рисунок 5.2.12 – Заказы, находящиеся в обработке

Если пользователь отменил свой заказ, находящийся в обработке, то на его баланс возвращаются средства в размере полной суммы отмененного заказа.

История заказов

ИНФОРМАЦИЯ О ЗАКАЗЕ

Код заказа: 3326123608



Arctic MX-2

Количество1


Стоимость80₽

Статус заказаОтменен

Общая стоимость80₽

ИНФОРМАЦИЯ О ЗАКАЗЕ

Код заказа: 3306342244



Deepcool matrexx 55

Количество1

Стоимость3500₽


Статус заказаОтменен

Общая стоимость3500₽

Рисунок 5.2.13 – Заказ отменены пользователем

ИНФОРМАЦИЯ О ЗАКАЗЕ

Код заказа: 1957861207



Acer Aspire A315

Количество1

Стоимость21100₽

Статус заказаВыполнен

Общая стоимость21100₽

Рисунок 5.2.14 – Заказ успешно завершен

5.2.4 Модификация информации в профиле

Если пользователь желает изменить свои персональные данные (имя, номер телефона, или email) или совершить пополнение баланса своего аккаунта, ему необходимо с помощью ссылки в навигационной панели перейти на страницу с профилем текущего пользователя.

На странице профиля пользователя отображаются его имя, адрес электронной почты, являющийся одновременно его логином, телефон и текущий баланс для совершения заказов.

Справа от каждого из этих полей имеется ссылка, которая открывает модальное окно для изменения текущего значения поля. При нажатии на кнопку редактирования с помощью клика левой кнопкой мыши будет выведено окно, имеющее поле для ввода нового значения.

В нижней части модального окна расположены две кнопки «Отменить» и «Подтвердить». Кнопка «Отменить» закрывает текущее модальное окно, и прежнее значение поля остается неизменённым. Кнопка «Подтвердить» применяет новое значение к текущему редактируемому полю профиля. После этого отправляется запрос на сервер для обновления данных в БД.

Пример страницы профиля пользователя приведен на рисунке 5.2.15.

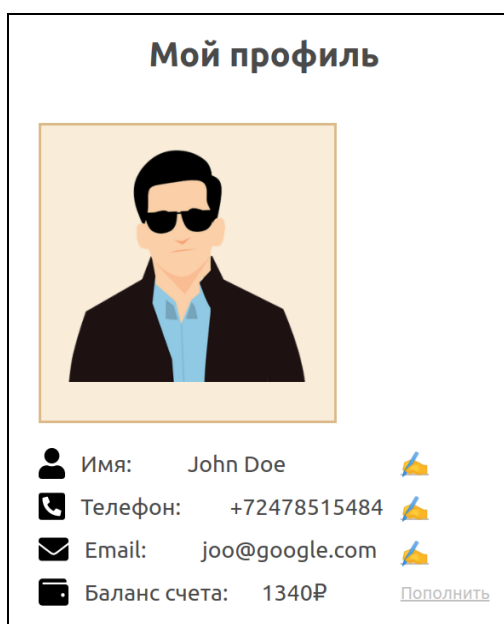


Рисунок 5.2.15 – Страница профиля пользователя

Пример модального окна пополнения баланса представлен на рисунке 5.2.16.

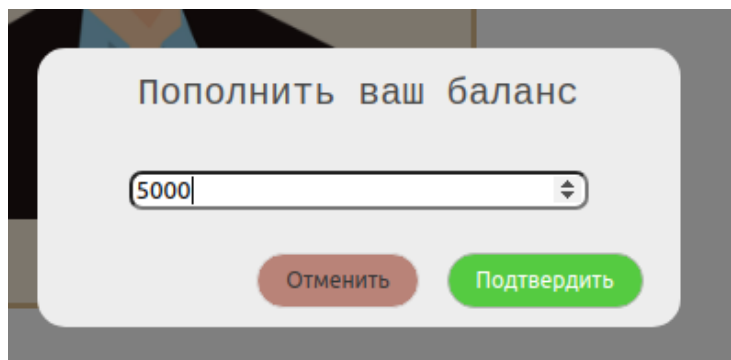


Рисунок 5.2.16 – Пополнение баланса

После отправки запроса на пополнение баланса страница обновилась, и были подгружены новые данные о счете пользователя (рисунок 5.2.17).

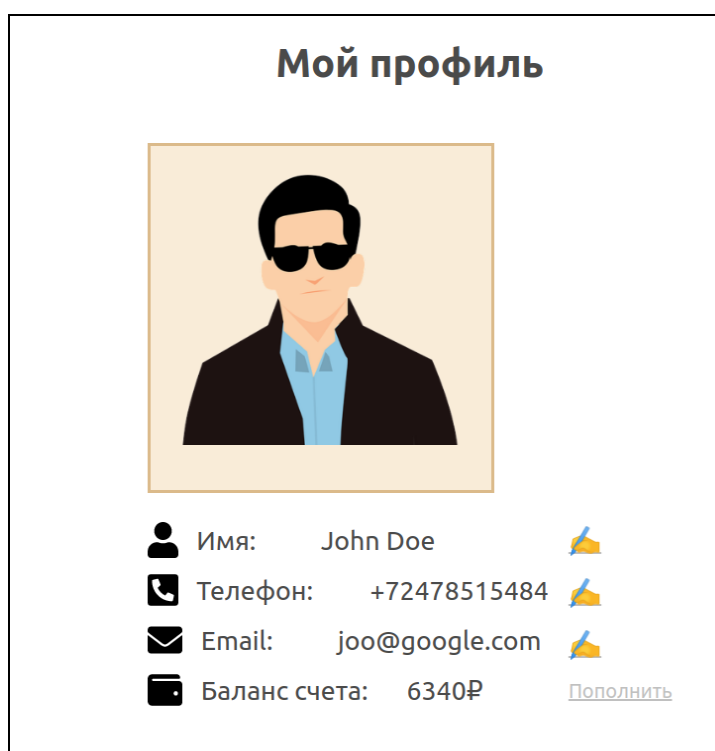


Рисунок 5.2.17 – Обновленные данные о балансе пользователя

5.2.5 Инструменты поиска и фильтрации товаров

На странице с каталогом товаров пользователь имеет возможности фильтрации списка представленных товаров. Например, на рисунке 5.2.18

используется фильтрация по категории «Охлаждение», а на рисунке 5.2.19 используется фильтрация списка по введенной в поле поиска строке. Если введенная пользователем строка или часть названия содержится в наименовании товара, то только эти товары будут отображены на странице.

Чтобы посмотреть товары, цена которые ниже чем установленная фильтром, пользователь может воспользоваться ползунком для регуляции цены или ввести точное значение напрямую в поле ввода

Чтобы сбросить поисковые фильтры необходимо очистить строку поиска и в меню категории выбрать пункт «Все категории».

Пример фильтрации списка товаров по категории приведен на рисунке 5.2.18.

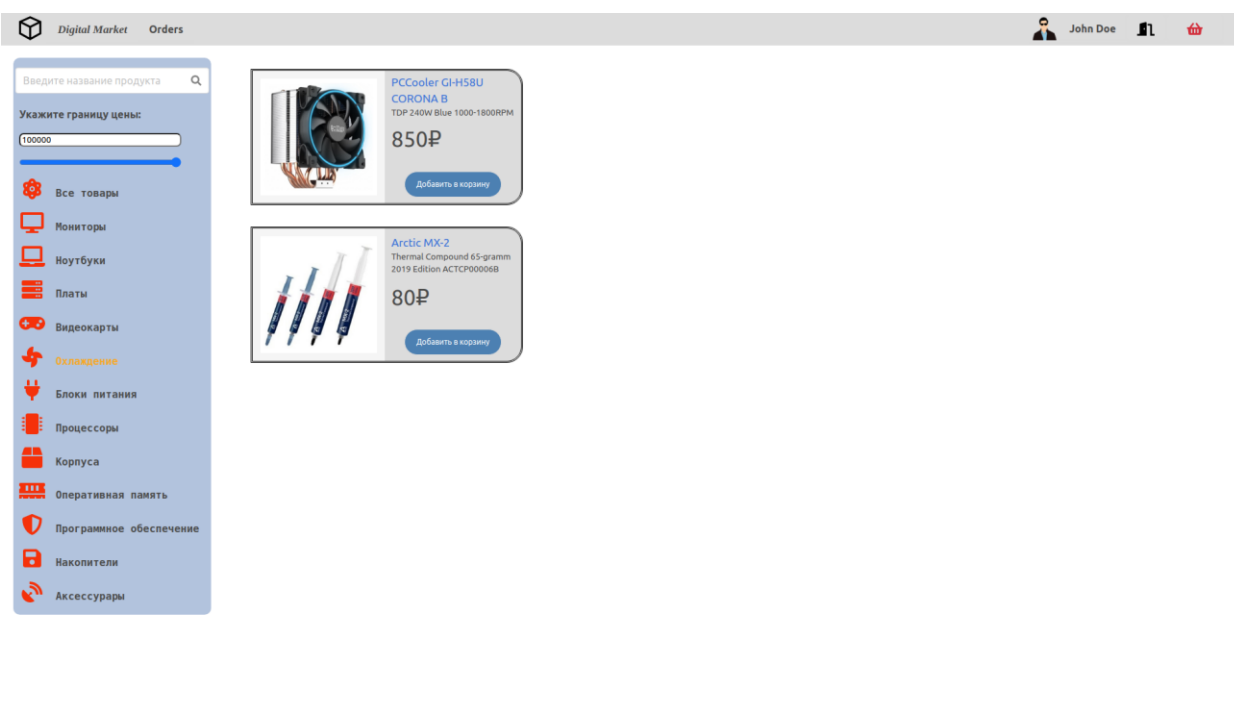


Рисунок 5.2.18 – Фильтрация по категории

Фильтрация списка товаров по названию и по цене работает аналогично.

5.3 Руководство администратора

5.3.1 Просмотр пользовательских профилей и заказов

Администратор ресурса имеет возможность просматривать профили пользователей, просматривать все текущие заказы пользователей, а также их историю, искать нужного пользователя по имени, видеть статистику заказов любого пользователя, их контактные данные и текущий баланс. Администратор может вручную отменить или подтвердить заказ любого пользователя. Это необходимо, если пользователь сам по каким-либо причинам не может этого сделать

Административная панель, отображающая информацию о пользователях и их заказах, представлена на рисунке 5.3.1.

The screenshot displays the administrative interface. On the left is a sidebar with a list of users: Admin, John Doe (selected), Michail, Ivan, Petr, Dima, Anton, Joseph, Micle, Luisa, Sebastian, Vika, Polina, and Achmed. The main area shows the profile of John Doe, including a placeholder avatar, email (joo@google.com), phone (+72478515484), balance (7000), and order statistics (0 created, 1 confirmed, 2 canceled). Below the profile is a section titled 'Список заказов пользователя John Doe' containing three order cards. Each card lists the order number, item name, quantity, price, status, total amount, and actions (Cancel/Confirm).

Номер	Название	Количество	Цена	Статус	Сумма заказа	Действия
3326123608	Arctic MX-2	1	80Р	Отменен	80Р	Отменить / Подтвердить
3306342244	Deepcool matrexx 55	1	3500Р	Отменен	3500Р	Отменить / Подтвердить
1957861207	Acer Aspire A315	1	21100Р	Выполнен	21100Р	Отменить / Подтвердить

Рисунок 5.3.1 – Информация о заказах пользователя

Слева расположена панель, отображающая список пользователей, можно выбирать пользователя из списка левой клавишей мыши (рисунок 5.3.2)

Информация о пользователях

The screenshot shows a web interface titled "Информация о пользователях". On the left, there is a search bar labeled "Enter user name" and a list of user names: Admin, John Doe, Michail, Ivan, Petr, Dima, Anton (highlighted), Joseph, Micle, Luisa, Sebastian, Vika, Polina, and Achmed. In the center, there is a large orange square containing a cartoon illustration of a man with brown hair and a blue shirt. On the right, there is a summary of user data for Anton: "Имя пользователя: Anton", "Email пользователя: Anton@mail.ru", "Телефон пользователя: +79823457156", "Баланс пользователя: 1200", "Созданные заказы: 0", "Подтвержденные заказы: 0", and "Отмененные заказы: 0".

Список заказов пользователя Anton

Рисунок 5.3.2 – Выбор пользователя из списка

При вводе имени пользователя в поисковую строку список будет автоматически фильтроваться, если пользователя с таким именем нет, то будет выведена соответствующая надпись.

В списке будут оставаться те пользователи, в имени которых содержится вводимая поисковая строка (рисунок 5.3.4).

Информация о пользователях

The screenshot shows the same web interface as Figure 5.3.2, but with the search bar containing "mic" and the list of user names filtered to show only "Michail" and "Micle". The central illustration now shows a cartoon man with grey hair wearing a green hat and a green shirt. The summary of user data on the right is for Michail: "Имя пользователя: Michail", "Email пользователя: micha@mail.ru", "Телефон пользователя: +75123694596", "Баланс пользователя: 70000", "Созданные заказы: 0", "Подтвержденные заказы: 0", and "Отмененные заказы: 0".

Список заказов пользователя Michail

Рисунок 5.3.4 – Фильтрация пользователей по имени

5.3.2 Операции добавления, модификации и удаления товаров

Администратор имеет возможности удаления, модификации и добавления товаров в базу данных. Для этих целей в интерфейсе приложения

предусмотрены специальные возможности, а также закрытые страницы, которые доступны только тем, кто имеет статус администратора (роль указывается в таблице «Пользователи» в базе данных).

Страница добавления товара состоит из двух частей: левая (обязательная для заполнения) и правая (заполнение предусмотрено опционально). В левой части интерфейса добавления товара необходимо заполнить поля – название товара, краткое описание, количество товара на складе, цена, и выбрать категорию товара. Перед отправкой формы на сервер поля левой части проходят валидацию. Например, поля – название товара и цена никогда не могут быть пустыми. В правой части интерфейса мы можем загрузить изображение товара (ограничения на форматы фотографий не предусмотрено), и дать подробную характеристику добавляемого товара. Если мы не загрузили изображение товара и отправили форму, то в интерфейсе пользователя будет отображаться изображение по умолчанию, устанавливаемое для товаров, не имеющих собственного изображения. При нажатии на кнопку «Сохранить товар» информация будет сохранена в базу данных приложения.

В интерфейсе страницы отдельного товара для администратора дополнительно появляются две кнопки «Модифицировать» и «Удалить товар», в интерфейсе обычного пользователя этих кнопок нет.

При нажатии на кнопку «Удалить товар» будет сделан запрос на сервер которая переадресует запрос к базе данных, а база данных в свою очередь, удалит товар из таблицы продуктов. После чего в интерфейсе страницы должно появиться всплывающее уведомление, указывающее что товар, был успешно удален, в ином случае будет показано уведомление об ошибке.

Пример страницы товара в интерфейсе администратора представлен на рисунке 5.3.6.



Рисунок 5.3.6 – Пример страницы товара в интерфейсе администратора

При нажатии на кнопку «Модифицировать» произойдет переход по якорной ссылке, расположенной ниже на этой же странице. При этом администратору будет доступен интерфейс для модификации товара. Интерфейс модификации имеет те же функциональные возможности, что и интерфейс добавления товара. Особенности интерфейса модификации является то, что поля в левой части по умолчанию уже заполнены текущими данными относящимися к модифицируемому товару. Таким образом, нет необходимости заново заполнять все поля и добавлять изображение товара, нужно лишь изменить те поля, которые необходимо, например, поля цены.

ЗАКЛЮЧЕНИЕ

В ходе проделанной работы была разработана универсальная автоматизированная система для управления торговыми процессами коммерческого предприятия.

Результатом работы стало выявление и обозначение проблем выполнения основных процессов, происходящих при выполнении торговой деятельности предприятия. Перед выполнением разработки автоматизированной системы были сформулированы цель и назначение. Обозначены необходимые функции, указана входная и выходная информация и требования к системе.

Исходя из собранных данных, была разработана объектная модель разрабатываемой системы, включающая диаграмму деятельности, диаграмму прецедентов, диаграмму компонентов диаграмму развертывания.

На следующем этапе были разработаны алгоритмы функционирования системы, приведена логическая модель структуры данных разрабатываемой БД, а также выбраны и обоснованы средства и методы разработки.

В итоге проделанной работы достигнута поставленная цель. Данная автоматизированная система позволит автоматизировать процессы торговли, что позволит сотрудникам торгового предприятия получать, добавлять и редактировать поступающую информацию, оперативнее и эффективнее обрабатывать заявки на клиентские заказы, увеличить надежность и доступность информации. Так оформление заказов ускорится до 5 раз, поиск информации о необходимых товарах до 10 раз, редактирование информации до 4 раз. Также применение автоматизированной системы сократит количество ошибок при обработке информации о пользователе. Что позволит эффективнее использовать трудовые и временные ресурсы предприятия.

Автоматизированная система может использоваться в сфере торговли, на предприятиях по продаже товаров и услуг в коммерческой сфере.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Вайк Аллен. JavaScript. Энциклопедия пользователя: Пер.англ. К.: ТОВ "ТИД" ДС", 2001. – 480 с.
- 2 Lenta Ru. Бизнес. – [Электронный ресурс] – Режим доступа: <https://lenta.ru/articles/2015/02/10/online/> (дата обращения: 20.06.2020).
- 3 Ашманов И. С., Иванов А. А. Продвижение сайта в поисковых системах. – М.: Вильямс, 2007. - 304 с.
- 4 Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. М.: Финансы и статистика, 1998. – 176 с.
- 5 Гагин А. Технология работы в глобальных в общедоступных сетях. М: Jet Infosystems, 2006. – 235 с.
- 6 Вильямсон Х. Универсальный Dynamic HTML. Библиотека программиста. СПб.: Питер, 2001. – 304 с.
- 7 Грызлов В. Java Script. Изд. 3-е.М.: ДМК Пресс, 2005. – 416 с.
- 8 Гудман Д. JavaScript. Библия пользователя, 4-е изд.: Пер. с англ. М.: Изд. Дом "Вильямс", 2003. – 960 с.
- 9 Гусев В.С. Google: эффективный поиск. Краткое руководство. М., 2006. – 296 с.
- 10 Калянов Г.Н. CASE. Структурный системный анализ (автоматизация и применение). М.: Лори, 1996. – 457 с.
- 11 Коггзолл Джон. PHP 5. Полное руководство.: Пер. с англ. М.: Издательский дом "Вильямс", 2006. – 752 с.: рис. – Парал. тит. англ.
- 12 Тереза Нейл, Билл Скотт. Проектирование веб-интерфейсов – Designing Web Interfaces. М.: Символ-Плюс, 2010. – 352 с.
- 13 Томсон Лаура, Веллинг Люк. Разработка Web-приложений на PHP и MySQL: Пер. с англ. - 2-е изд., испр. СПб: ООО ДиаСофтЮП, 2003. – 672 с.
- 14 Петров В.И. Информационные системы. СПб.:Питер, 2002. – 688 с.
- 15 Ратбон Э. JavaScript для чайников. К.: Диалектика, 1995. – 236 с.

- 16 Фленов М. Е. Программирование глазами хакера. СПб.: БХВПетербург, 2003. – 368 с.
- 17 Колисниченко Д. Н. Поисковые системы и продвижение сайтов в Интернете. М.: Диалектика, 2007. – 272 с.
- 18 Гудман Д. JavaScript и DHTML. Сборник рецептов. Для профессионалов, Питер, 2004 г.
- 19 Дэвис С. Языки JavaScript и VBScript.–К.: Диалектика,1996.– 654 с.
- 20 Лещев,Д. Создание интерактивного web-сайта: учебный курс. СПб.: Питер, 2003. – 544 с.: ил. 74
- 21 Матросов А., Сергеев А., Чаунин М. HTML 4.0. СПб: Издательство "БНВ", 1999. – 670 с.
- 22 Мэрдок, Келли, Л. JavaScript: наглядный курс создания динамических Web-страниц.: Пер. с англ.: Уч. пос. М.: Издательский дом "Вильямс", 2001. – 288 с.: ил. – Парал. тит. англ.
- 23 Шапошников И. Web-сайт своими руками. БХВ – Санкт-Петербург, 2000.
- 24 Microsoft Visual Studio 2015 – ПО от Microsoft [Электронный ресурс] – Режим доступа: <https://www.microsoft.com/ru-ru/SoftMicrosoft/vs2015Community.aspx> (дата обращения: 20.05.2020).
- 25 И.В. Ананченко, И.В. Козлов Администрирование Microsoft SQL Server 2012R2: учеб. пособие / СПб.: НИУ ИТМО, 2014. – 71 с.
- 26 Kors software – программы для торговли [Электронный ресурс] – Режим доступа: <https://www.kors-soft.net/differences.php?id=40> (дата обращения: 21.05.2020).
- 27 ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы (Взамен ГОСТ 24.201-85) [Электронный ресурс]. – Режим доступа: <http://www.rugost.com> (дата обращения: 25.05.2020).

28 Современный учебник JavaScript [Электронный ресурс] – Режим доступа: <https://learn.javascript.ru/> (дата обращения: 15.06.2020).

29 Mark Myers - A Smarter Way to Learn JavaScript: The new tech-assisted approach that requires half the effort: учеб. пособие, 2014. – 256 с.

30 Эрик Фримен. Паттерны проектирования. СПб.: Элизабет Фримен, 2011. – 645 с.

31 Павловская, Т. А. Объектно-ориентированное программирование. – Т. А. Павловская, Ю. А. Щупак – СПб: Питер, 2006, 256 с.

ПРИЛОЖЕНИЕ А

Примеры отображения веб-ресурса на мобильном устройстве.

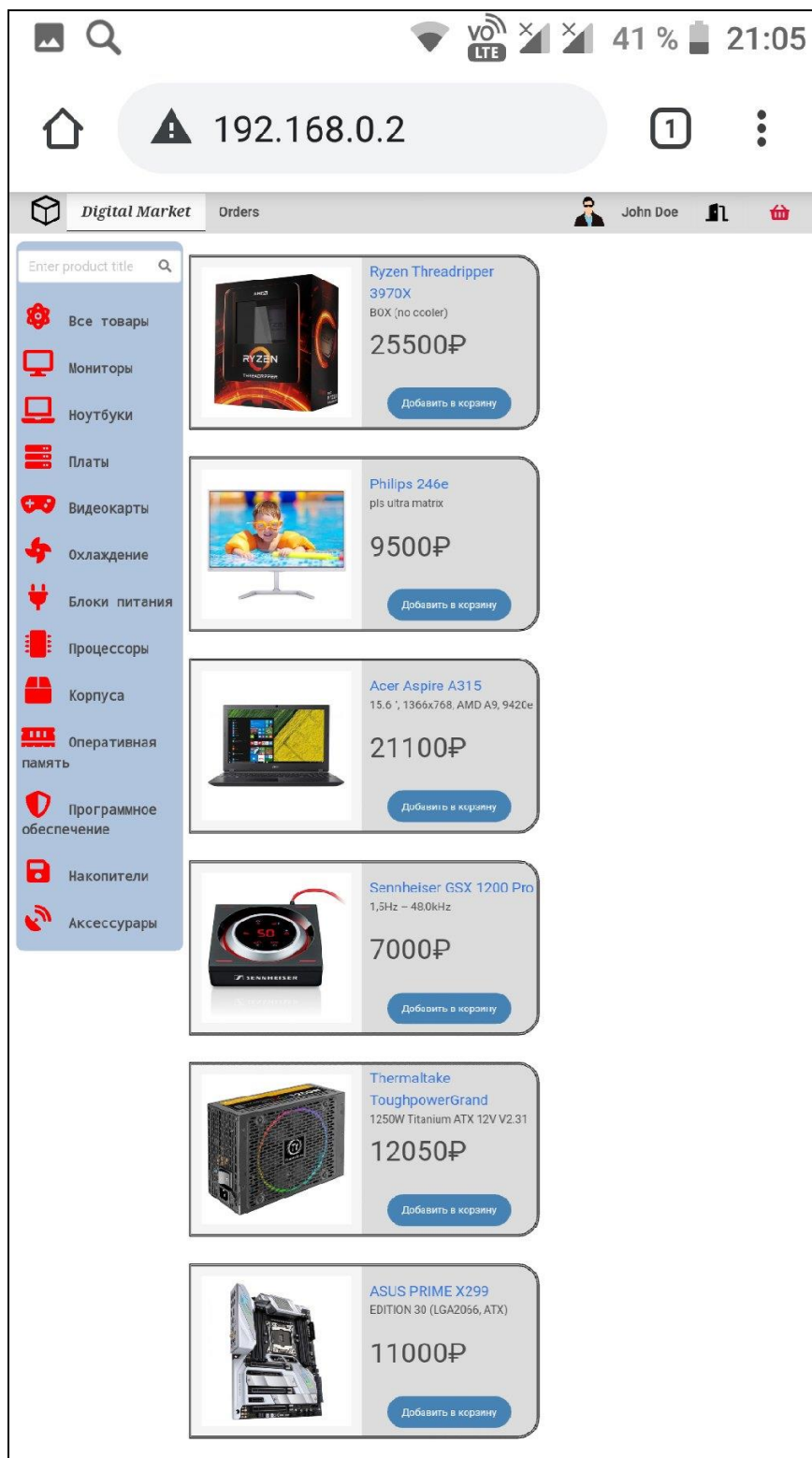


Рисунок А.1 – Каталог товаров на мобильном устройстве

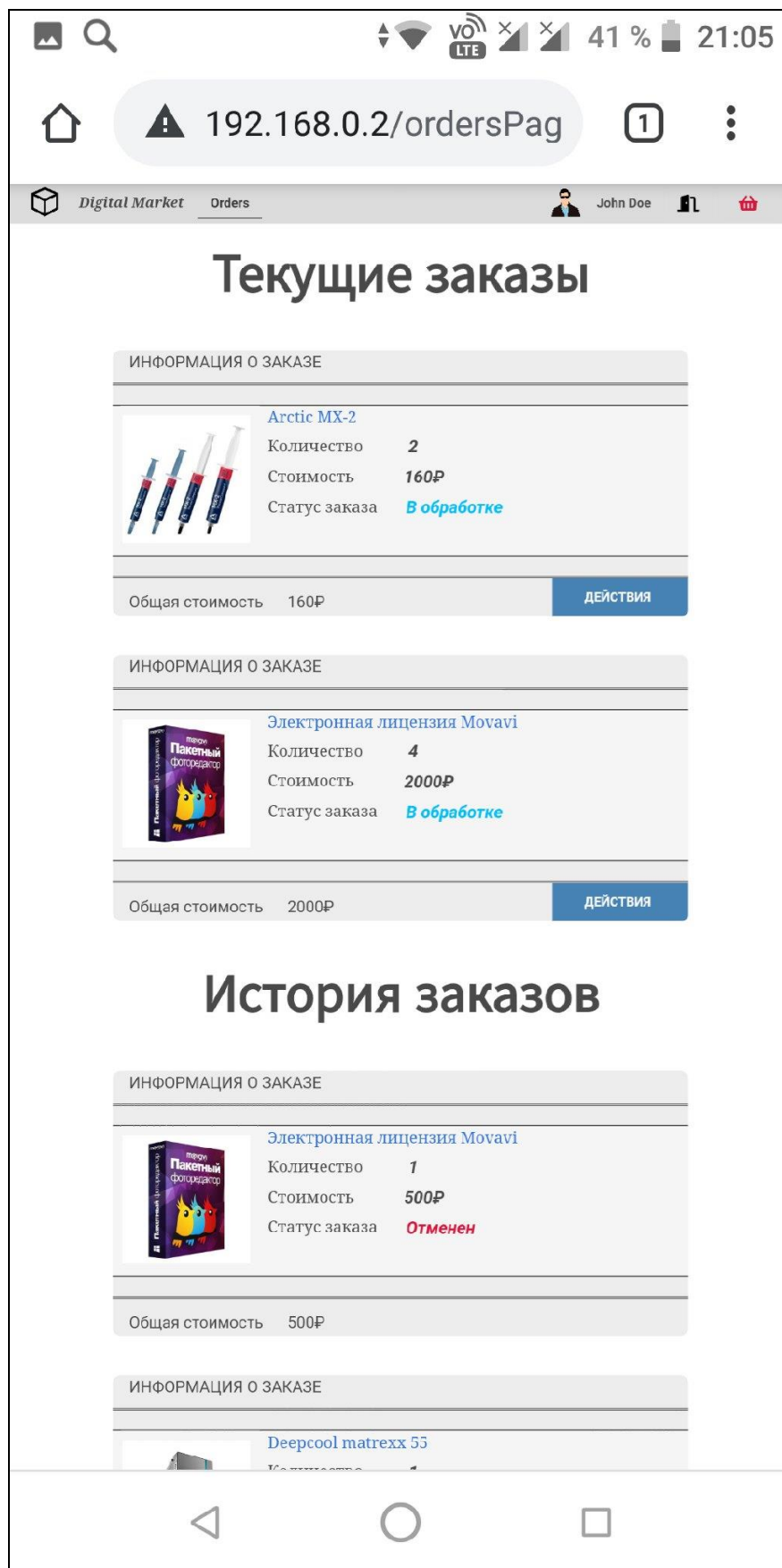


Рисунок А.2 – Страница заказов на мобильном устройстве

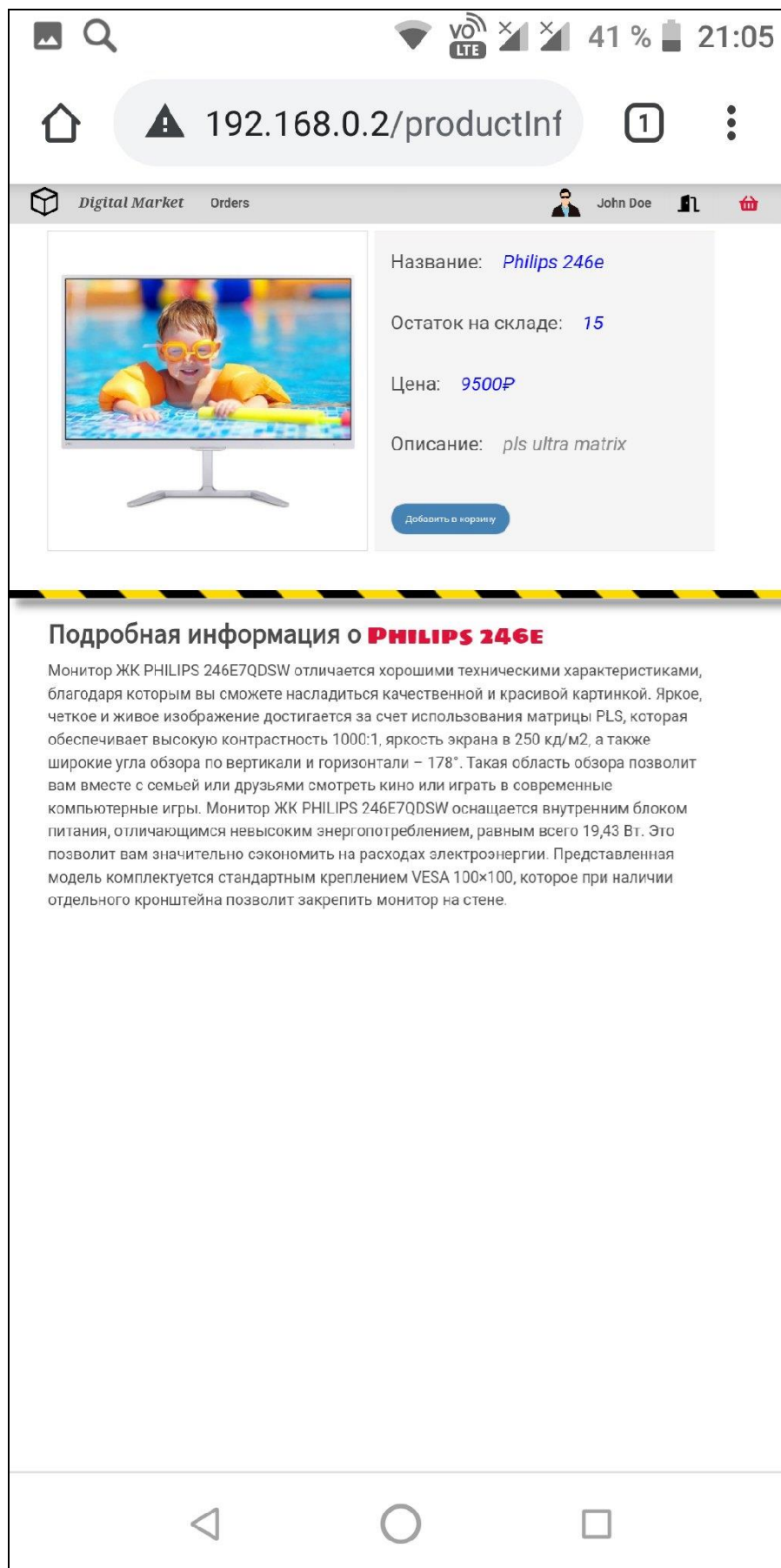


Рисунок А.3 –Страница товара на мобильном устройстве

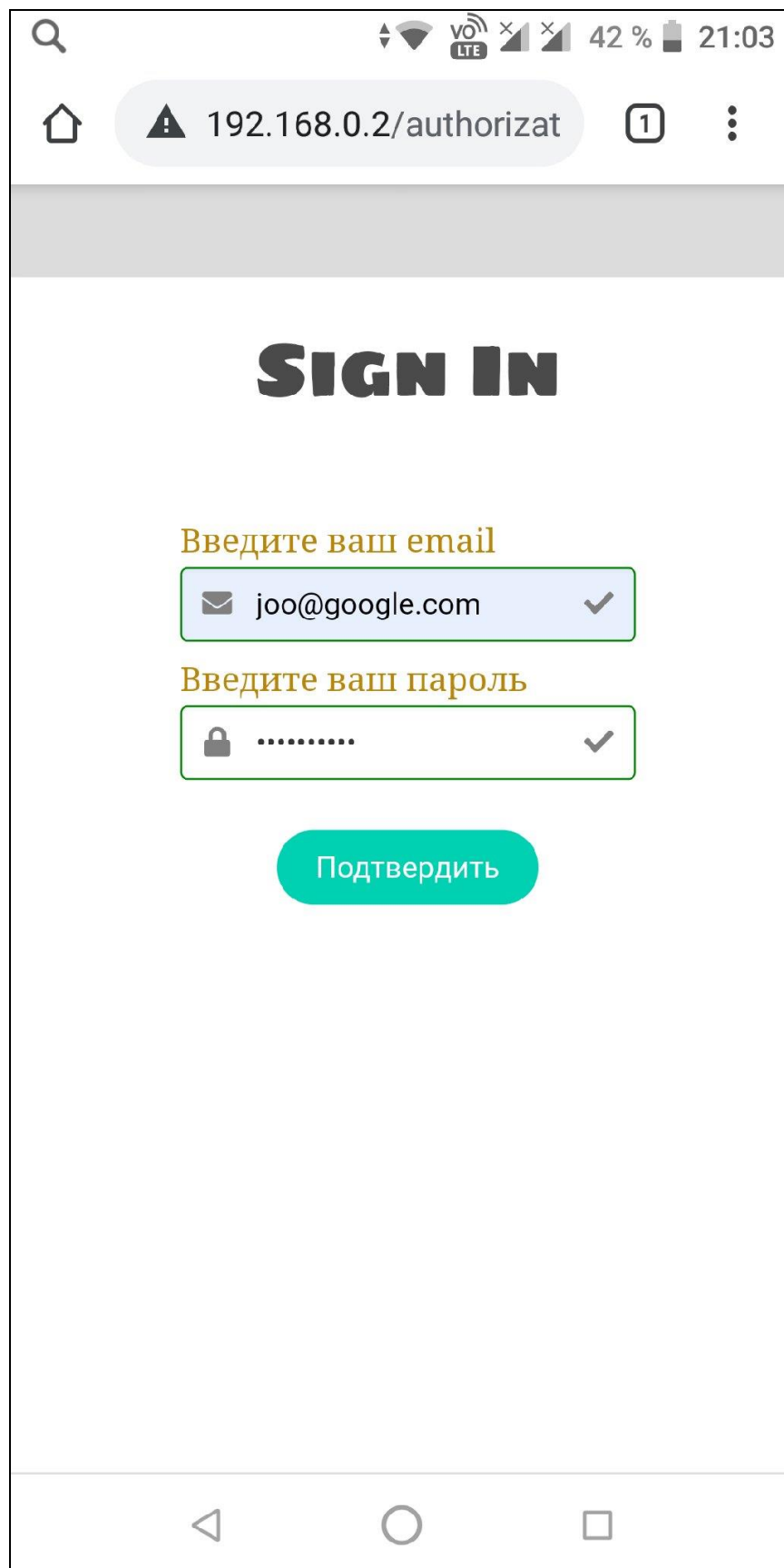


Рисунок А.4 – Страница авторизации на мобильном устройстве

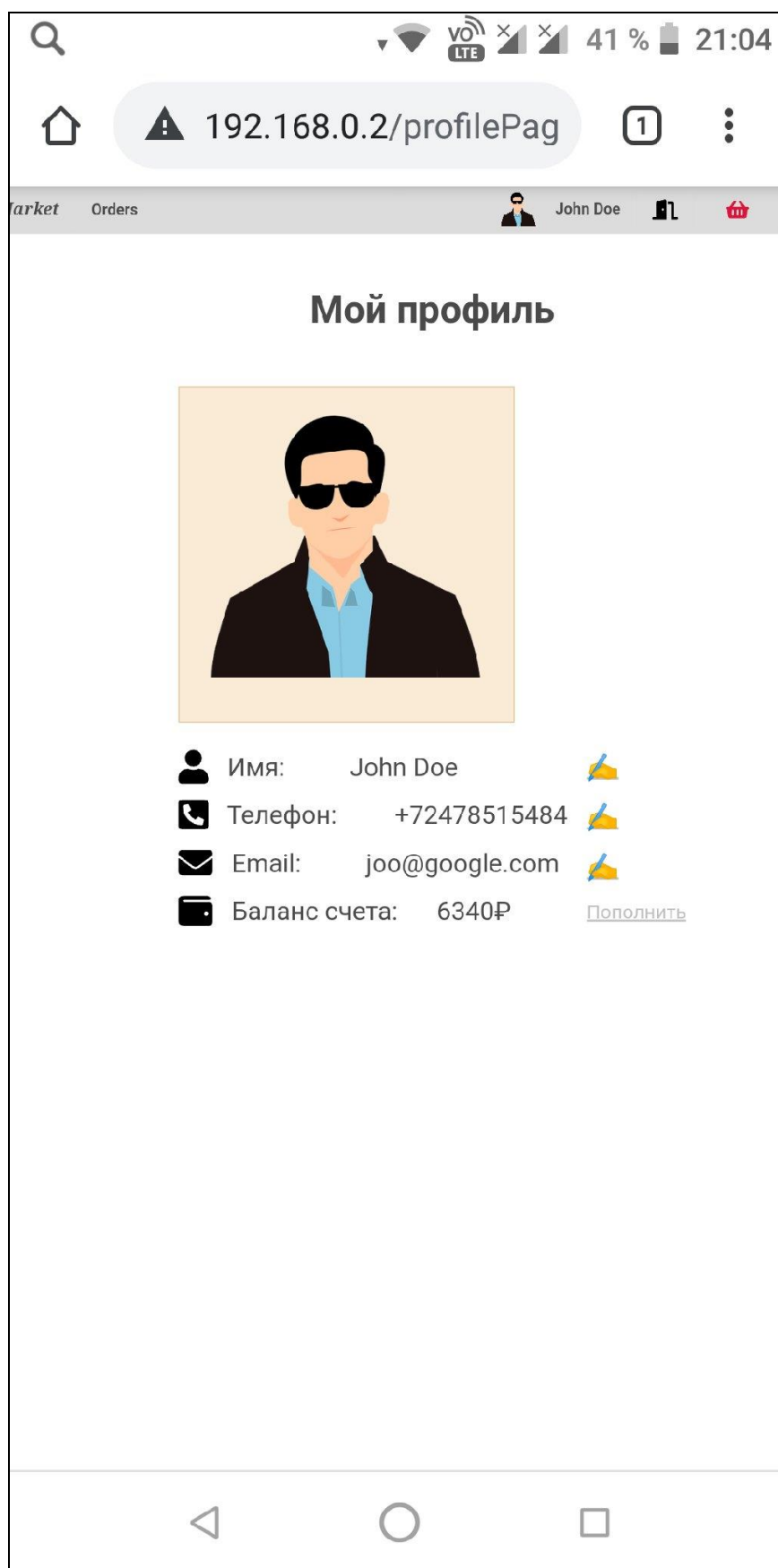


Рисунок А.5 – Профиль пользователя на мобильном устройстве

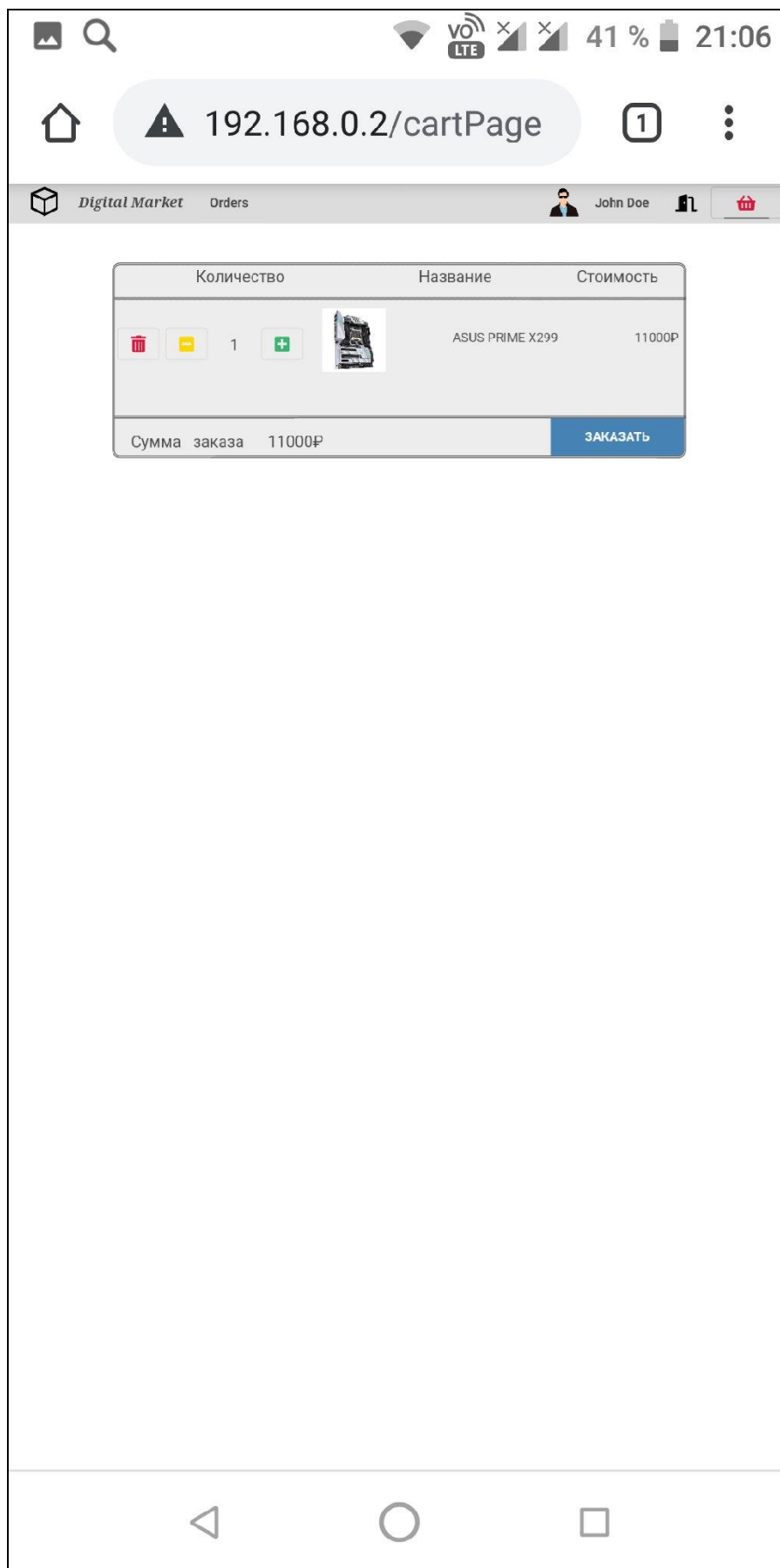


Рисунок А.6 – Страница корзины на мобильном устройстве