Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Инженерно-экономический факультет

Кафедра экономической информатики

Дисциплина: Распределенные системы обработки информатики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту на тему:

**Разработка автоматизированной системы управления контентом магазина электроники**

Выполнил: студент гр. 074002

Шабанов Тимофей Маратович

Проверил: ассистент кафедры ЭИ

Лыщик Артём Петрович

Минск 2023

**СОДЕРЖАНИЕ**

[Введение](#_Toc71623245)  3

[1 Описание работы интернет-маркетинга и системы управления контенотом магазина электроники](#_Toc71623246)  6

[1.1 Основные понятия об интернет-маркетинге](#_Toc71623247)  6

[1.2 Отличия и преимущества интернет-маркетинга от офлайн-маркетинга](#_Toc71623248) 7

[1.3 Интернет-маркетинг в Республике Беларусь](#_Toc71623249)  9

[2 Описание применения паттернов проектирования 1](#_Toc71623250)1

[2.1 Постановка задачи 1](#_Toc71623251)1

[2.2 Обзор вариантов решения поставленной задачи 1](#_Toc71623252)1

[3 Функциональное моделирование на основе стандарта IDEF0, IDEF1.X 1](#_Toc71623253)4

[4 Информационная модель системы и ее описание 1](#_Toc71623255)9

[5 Модели представления и их описание 2](#_Toc71623256)3

[5.1 Диаграмма вариантов использования системы управления контентом магазина электроники 2](#_Toc71623257)3

[5.2 Диаграмма состояний системы управления контентом магазина электроники 2](#_Toc71623258)3

[5.3 Диаграмма последовательностей системы управления контентом магазина электроники 2](#_Toc71623259)4

[5.4 Диаграмма классов системы управления контентом магазина электроники 2](#_Toc71623260)5

[5.5 Диаграмма компонентов системы управления контентом магазина электроники 2](#_Toc71623261)5

[5.6 Диаграмма развертывания системы управления контентом магазина электроники 2](#_Toc71623262)6

[6 Описание алгоритмов, реализующих бизнес-логику серверной части проектируемой системы 2](#_Toc71623263)7

[7 Руководство пользователя 2](#_Toc71623264)8

[8 Результаты тестирования разработанной ситемы 3](#_Toc71623265)6

[Заключение](#_Toc71623266)  39

Список используемых источников40

[Приложение А 4](#_Toc71623267)1

[Приложение Б](#_Toc71623267) 42

[Приложение В](#_Toc71623268) 46

[Приложение Г](#_Toc71623269) 48

[Приложение Д](#_Toc71623270)  55

[Приложение Е 6](#_Toc101697314)1

# **ВВЕДЕНИЕ**

Главная особенность нынешнего этапа развития общества заключается в его информатизации, то есть во внедрении компьютеров и различных средств связи во все сферы деятельности человека. Необходимость, важность и незаменимость использования компьютерных технологий для учета различных отраслей предприятий и слаженной организации их деятельности в целом, уже давно принято во всеобщее внимание и потребность в технологиях невозможно переоценить.

Конъюнктура рынка постоянно меняется, надобность и необходимость высоких скоростей в принятии решений, многозадачность в управлении активами и потребность снижения рисков требуют современных подходов к организации хозяйственной деятельности. В связи со всеми вышеперечисленными факторами бизнес-процессы нуждаются в автоматизации. Она позволяет высвободить драгоценные ресурсы для стратегического планирования и концентрации менеджмента на ключевых направлениях работы компании.

Информационные технологии постоянно развиваются и с развитием сети Интернет создаются новые способы ведения современного бизнеса. Растет рынок интернет-торговли и в свою очередь меняется потребитель. Люди все больше прибегают к Интернету и все чаще покупают в Сети. Кроме того, интернет-маркетинг стал существенно влиять на традиционные продажи. В сознании потребителей онлайн и офлайн-торговля постепенно смешиваются, перестают конкурировать и становятся взаимодополняемыми элементами процесса покупок.

Самая распространенная модель электронной коммерции – это интернет-магазины. Практически все сервисы в интернете, это магазины, которые продают, либо товар, либо услугу, либо информацию. Такой вид бизнеса не требует таких ресурсов, как аренда или покупка торгового помещения, приобретение торгово-технологического оборудования, не требует затрат на огромное количество торгового и административного персонала. Такая торговля предоставляет возможность быстрого реагирования на спрос.

Создание эффективного менеджмента возможно через продуманное управление документооборотом, корпоративным контентом и бизнес-процессами путем внедрения системы для автоматизации бизнеса, что в свою очередь ведет к значительной результативности и точности процессов, а также увеличению прибыли и экономии времени.

Основными задачами автоматизации бизнеса являются следующие:

− эффективная поддержка оперативной деятельности предприятия, организация учета и контроля;

− подготовка различных документов для партнеров, включая накладные, акты сверки, деловые предложения и т. д.;

− быстрое получение отчетов о состоянии дел в компании за любой временной период;

− оптимизация затрат на персонал, увеличение эффективности использования рабочего времени путем освобождения сотрудников от рутинной и долгой работы;

− сведение к минимуму негативного влияния «человеческого фактора» на важнейшие бизнес-процессы;

− безопасное хранение информации;

− повышение качества обслуживания клиентов.

Автоматизация бизнес-процессов может значительно повысить качество управления в компании и качество ее продукта, дает ряд существенных преимуществ, таких как:

− увеличение скорости обработки информации и решения повторяющихся задач;

− рост согласованности действий персонала и качества его работы;

− возможность контроля больших объемов информации;

− автоматизация ручного труда;

− уменьшение количества разного рода ошибок и повышение точности управления;

− параллельное решение нескольких задач.

Перед данной системой разработки автоматизированной системы управления контентом электронного магазина стоят следующие задачи:

− надежность работы системы;

− простота, доступность и удобство для пользователей (как для покупателей, так и для работников);

− достоверность имеющейся информации.

Целью курсового проекта является повышение эффективности бизнес-процессов предприятия при помощи автоматизации некоторых расчетов, удобство работы с товаром для покупателей и работников, составление отчетов о состоянии бюджета предприятия.

Для достижения поставленной цели необходимо:

− изучит особенности интернет-маркетинга;

− изучить особенности учёта такого производства, как электронный магазин;

− сформировать диаграмму классов;

− спроектировать объектную модель;

− разработать базу данных;

− разработать удобную и понятную программу.

**1 ОПИСАНИЕ РАБОТЫ ИНТЕРНЕТ-МАРКЕТИНГА И СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ МАГАЗИНА ЭЛЕКТРОНИКИ**

**1.1 Основные понятия об интернет-маркетинге**

Интернет-маркетинг появился в начале 1990-х годов, когда текстовые сайты начали размещать информацию о товарах. Интернет-маркетинг – деятельность, направленная на создание условий прямого приобретения товаров потребителями при помощи инструментов Всемирной паутины. Иначе говоря, это – торговля, которая осуществляется при помощи сети Интернет. Следует отметить, что сетевые инструменты довольно специфичны и универсальны, что позволяет при их помощи продавать не только товары, но еще и услуги.

В настоящее время благодаря высокому уровню развития логистики и наличию высококачественных транспортных сервисов интернет-маркетинг уже не имеет существенных отличий от обычной торговли и во многом уже обошел ее. Электронный магазин, как и любой иной, позволяет человеку купить товар и получить его, при этом имея существенное отличие, являющееся положительным фактором, – для приобретения товара искать время и силы на преодоление пути до него не нужно, стоит лишь зайти на специально предназначенную платформу и выбрать необходимый продукт.

У интернет-маркетинга есть три важнейших преимущества, каковыми не обладает любой другой маркетинг:

1. Интерактивность. Благодаря интернету можно постоянно находиться во взаимодействии с потенциальными покупателями, поддерживать с ними связь и контролировать их действия.

2.Таргетирование. Механизм, позволяющий выбрать определенную целевую аудиторию и показывать рекламу именно ей.

3. Веб-аналитика. Позволяет выявить наиболее эффективные действия, благодаря которым на сайте появились новые посетители, в дальнейшем ставшие покупателями [1].

Если рассматривать пирамидальную структуру онлайн-маркетинга, можно понять, что она является многокомпонентной и включает в себя:

− уникальное торговое предложение;

− определенные бизнес-цели компании, имеющей сайт в сети;

− принципы работы этой компании;

− ее основные приоритеты и ценности.

Основу интернет-маркетинга составляют следующие его виды:

− поисковой и мобильный маркетинг;

− социальный маркетинг;

− интернет-реклама;

− контент-маркетинг;

− e-mail-маркетинг.

Цели интернет-маркетинга:

1. Продвижение компании:

− работа с брендингом – укрепление всевозможных компонентов, которые позволят повысить узнаваемость бренда, сформировать у потенциального потребителя определенное мнение о нем, создать нужный имидж, положительно влияющий на развитие компании;

− узнаваемость – одна из составляющих брендинга, при этом представляет с собой наиболее важную его часть. Именно узнаваемость позволяет понять, насколько представители целевой аудитории осведомлены о наличии бренда;

− повышение охвата и распространения информации о бренде – отражает такие показатели интернет-маркетинга как воспроизведение информации по разным медиаканалам, вирусный эффект и пр.

1. Работы над продажами:

− оффлайн-стимулирование – повышение показателей продаж вне сети через Интернет;

− повышение продаж через Интернет – цель, ориентированная больше всего на показатели конверсии веб-ресурса;

− Вывод на рынок новой продукции, продукта или услуги.

1. Промежуточные цели:

− снижение уровня затрат, которые необходимы для привлечения новых клиентов, их удержание, обеспечение качественного сервиса;

− повышение уровня эффективности рекламных кампаний, вне зависимости от того, проводятся они в сети или за ее пределами;

− формирование определенного мнения о продукте (товаре или услуге), которая выводилась на рынок инструментами интернет-маркетинга;

− проведение анализа и различных исследований, которые позволяют понять потребности целевой аудитории, ее потребительские предпочтения и т.д.

Если подвезти итог о цели интернет-маркетинга, то все настроено на одно: превращение посетителей сайта в покупателей и увеличение прибыли.

**1.2 Отличия и преимущества интернет-маркетинга от офлайн-маркетинга**

В [основе интернет-маркетинга](https://webcom.academy/kursyi_internet_marketing/) лежит практика использования традиционных маркетинговых действий, но адаптированных к потребностям интернета и пользователя сети. Если говорить о характерных элементах как интернет-, так и оффлайн-маркетинга, то следует перечислить такие как: наличие товара или услуги, которую следует продать, цена, место продажи и комплекс мероприятий по продвижению.

Отличия интернет-маркетинга от офлайн-маркетинга:

1.Временная и пространственная континуальность – непрерывное функционирование интернета в любой точке мира.

2.Универсальность – отсутствие пространственных ограничений при реализации маркетинговой кампании, сетевые протоколы выполняются одинаково в любой точке планеты.

3. Многоканальность – распространение информации возможно через несколько сервисов и механизмов: веб-сайты, обмен файлами, электронная почта, видео- и аудиосервисы, соцсети, мобильные приложения и многое другое.

4. Управляемость маркетинговыми действиями – возможность, благодаря которой отслеживается активность потребителей и изменение стратегии при отсутствии полезных действий пользователей.

Особенности комплекса интернет-маркетинга:

− товар (product) – то, что продается в интернете. Очень часто интерактивность интернет-маркетинга позволяет продвигать, позиционировать и осуществлять продажу технически-сложного, венчурного продукта;

− цена (price) — цена на товар в интернете зачастую ниже цены на аналогичный товар в ритейле, в силу отсутствия затрат на дистрибьюцию и снижение издержек on-line продаж;

− продвижение (promotion) — комплекс мер по продвижению в интернете. Интернет-маркетинг предполагает продвижение как самого товара (услуги), так и информационной площадки (интернет-ресурса);

− место продаж (place) — в роли точки продаж выступает интернет-ресурс – сайт торговой марки, продукта, компании. Необходимо обратить внимание на юзабилити – удобство пользования интернет-сайтом, удобство совершения покупки: интерактивность взаимодействия с покупателями до, во время и после продажи, информационный сервис, сервис по доставке, разнообразие методов оплаты.

Таким образом, интернет-маркетинг опережает своего конкурента. Он предоставляет больше возможностей, позволяя практически мгновенно вносить изменения, подстраиваясь под настроения и потребности покупателей. С помощью виртуальных инструментов намного проще подобрать площадки для размещения объявлений, более разумно расходовать бюджет и получать нужный результат быстрее, позволяет мгновенно анализировать целевую аудиторию и ее настроения, исправлять недочеты и прогнозировать дальнейшие новшества.

Но не стоит преуменьшать эффективность офлайн-маркетинга для развития интернет-бизнеса. Это необычайно актуально для компаний, желающих пополнить свою аудиторию представителями поколения Х, привыкших к традиционной рекламе в журналах, на радио и ТВ, но имеющих доступ в сеть.

Не стоит сомневаться в эффективности совместной работы онлайн и офлайн-маркетинга. Необходимо лишь разработать стратегию, учитывающую тренды обоих направлений и социальной действительности.

**1.3 Интернет-маркетинг в Республике Беларусь**

В Республике Беларуси компании, которые планируют свой выход в сеть Интернет, ставят перед собой следующие цели и задачи, выбирают Интернет, чтобы получить следующие преимущества и получить возможность:

− избежать давления на физическом рынке;

− найти новый сегмент потребителей;

−обслуживать имеющихся клиентов через новый канал, поддерживать их лояльность;

− обеспечить конкурентоспособность;

− сократить издержки;

− обеспечить большую информативность и динамичность.

Также переход компаний в сторону использования сети Интернет обусловлен и тем, что зачастую гораздо проще определить общий результат кампании в Интернете и эффективность каждого задействованного в ней направления. Используя всевозможные метрики и счетчики, можно подсчитать результат конверсии между количеством людей, увидевших объявления, пришедших на сайт, позвонивших в компанию и совершивших покупку и т.д.

По данным пресс-релиза Международного союза электросвязи от 22.07.2016, количество пользователей Интернета в мире составляет 3,5 миллиарда человек. В период с 2000 по 2015 год удельный вес пользователей интернета увеличился почти в 7 раз − с 6,5 до 43 % мирового населения.

Если говорить о Республике Беларусь, то стоит отметить, что использование Интернета жителями Республики Беларусь постоянно растет. Так, в 2011 году доля интернет-пользователей составила приблизительно 47%, в 2013 – уже 58%, а в 2015 – перешла отметку в 63%. По данным медиа исследования gemiusAudience за август 2016 года в Республике Беларусь общая аудитория интернета в возрасте 15-74 года достигла 5 802 000 человек. Этот показатель вырос на 18% по сравнению с мартом 2014 года – размер интернет-аудитории в марте 2014 г. составил 4 910 571 пользователей.

Набирает обороты и использование Интернета белорусскими компаниями. В настоящее время на территории республики зарегистрированы 5817 интернет-магазинов – это в 1,5 раза больше по сравнению с 2010 годом.

Динамика роста объемов продаж товаров через интернет-магазины по г. Минску за последние 3 года, позволяют спрогнозировать дальнейший рост продаж не менее чем в 1,5 раза и увеличении доли в товарообороте до 5%. Самое большое влияние интернет-маркетинг оказал в Беларуси на торговлю. В 2017 году в стране было зарегистрировано 13,8 тысяч интернет-магазинов – это в 4 раза больше, чем в 2011 году.

Но не стоит преуменьшать эффективность офлайн-маркетинга для развития интернет-бизнеса. Это необычайно актуально для компаний, желающих пополнить свою аудиторию представителями поколения Х, привыкших к традиционной рекламе в журналах, на радио и ТВ, но имеющих доступ в сеть.

Не стоит сомневаться в эффективности совместной работы онлайн и офлайн-маркетинга. Необходимо лишь разработать стратегию, учитывающую тренды обоих направлений и социальной действительности [2].

В процессе продвижения в Интернете компании сталкиваются с определенными трудностями и сложностями. Так, специфика интернет-аудитории накладывается на портреты потенциальных клиентов. У них свои привычки и ожидания, свое видение удобства интерфейса сайта и приобретения покупки на нем, на сайте и в магазине люди будут вести себя по-разному и выбор их также будет разниться. Портреты целевой аудитории, по сути, задают вектор всего продвижения: нужно продвигаться там, где преобладает нужная аудитория, и использовать те маркетинговые зацепки, которые будут интересны именно этим людям.

**2 ОПИСАНИЕ ПРИМЕНЕНИЯ ПАТТЕРНОВ ПРОЕКТИРОВАНИЯ**

**2.1 Постановка задачи**

Ссылаясь на задание, необходимо выполнить следующую задачу: разработать систему управления контентом электронного магазина. Программа должна представлять собой графическое приложение в архитектуре клиент-сервер с многопоточным сервером для работы сразу с несколькими пользователями на языке Java, которое даст возможность пользователю просматривать информацию, вносить изменения, выводить конечный результат, должно быть организовано взаимодействие с базой данных.

Главная цель приложения – эффективная работа по управлению контентом электронного магазина, удобное пользование интерфейсом.

Основные задачи, которые должен выполнять проект:

− регистрация и вход в учетную запись пользователя (в роли администратора или покупателя);

− добавление товара;

− поиск товара;

− редактирование информации о товаре;

− создание заказа;

− удаление товара;

− добавление товара в корзину;

− статистика покупок.

Возможность добавления, редактирования и удаления данных о товаре существует только у администратора.

**2.2 Обзор вариантов решения поставленной задачи**

Приложение должно быть написано на объектно-ориентированном языке программирования JAVA. Взаимодействие между серверной и клиентскими частями будет осуществляться с использованием сокетов и протокола TCP/IP.

В разработанной системе будет использоваться СУБД MySQL. MySQL – это система управления реляционными (данные реализованы в виде таблиц) базами данных с открытым исходным кодом (СУРБД) с моделью клиент-сервер. [СУРБД](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) — это программное обеспечение или служба, используемая для создания и управления базами данных на основе реляционной модели.

База данных — это просто набор структурированных данных, место, в котором хранятся данные. MySQL создаёт базу данных для хранения и управления данными, определяющими отношения каждой таблицы. Клиенты могут делать запросы, вводя определённые команды SQL на MySQL.

Для создания графического интерфейса будет использоваться библиотека JavaFX, которая представляет с собой инструментарий для создания кроссплатформенных графических приложений на платформе Java, позволяет создавать приложения с богатой и насыщенной графикой благодаря использованию аппаратного ускорения графики и возможностей GPU. JavaFX предоставляет большие возможности по сравнению с рядом других подобных платформ для создания интерфейса. Он обладает большим набором элементов управления, и возможности по работе с мультимедиа, двухмерной, а также трехмерной графикой, декларативный способ описания интерфейса с помощью языка разметки FXML, возможность стилизации интерфейса с помощью CSS, интеграция со Swing и многое другое [3].

Также были использованы паттерны программирования.

Паттерн Model-view-controller (MVC) позволяет отделить графический интерфейс от бизнес-логики, а бизнес логику от данных. Таким образом, MVC состоит из трех частей, которые и дали ему название, модель, представление, контроллер. Моделью, это та часть, которая содержит в себе функциональную бизнес-логику приложения. Модель должна быть полностью независима от остальных частей продукта, она ничего не должна знать об элементах дизайна, и каким образом он будет отображаться. Представление отображает данные, полученные от модели. Однако, представление не может напрямую влиять на модель, то есть обладает доступом «только на чтение» к данным. Контроллер перехватывает событие извне и в соответствии с заложенной в него логикой, реагирует на это событие изменяя модель, посредством вызова соответствующего метода. Контроллер определяет, какое представление должно быть отображено в данный момент. События представления могут повлиять только на контроллер, контроллер, в свою очередь, может повлиять на модель и определить другое представление [4].

ПаттернSingleton– порождающий шаблон проектирования, гарантирующий, что в приложении будет единственный экземпляр некоторого класса, и предоставляющий глобальную точку доступа к этому экземпляру. При попытке создания данного объекта он создаётся только в том случае, если ещё не существует, в противном случае возвращается ссылка на уже существующий экземпляр и нового выделения памяти не происходит [5].

Паттерн Repository *–* позволяет абстрагироваться от конкретных подключений к источникам данных, с которыми работает программа, и является промежуточным звеном между классами, непосредственно взаимодействующими с данными, и остальной программой,репозиторий добавляет программе гибкость при работе с разными типами подключений.

Паттерн Unit of Work – позволяет упростить работу с различными репозиториями и дает уверенность в том, что все репозитории будут использовать один и тот же контекст данных. Класс Unit of Work предоставляет доступ к репозиториям через отдельные свойства и определяет общий контекст данных для всех репозиториев.

Паттерн стратегия – поведенческий паттерн проектирования, который определяет семейство схожих алгоритмов и помещает каждый из них в собственный класс, после чего алгоритмы можно взаимозаменять прямо во время исполнения программы [6].

**3 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ НА ОСНОВЕ СТАНДАРТА IDEF0, IDEF1.X**

IDEF0 – это методология графического описания систем и процессов деятельности организации как множества взаимосвязанных функций. Она позволяет исследовать функции организации, не связывая их с объектами, обеспечивающими их реализацию.

Вероятность происхождения какой-либо ошибки в одной из части программ не исключена. Поэтому помимо знания о том, какая и в каком блоке ошибка, важно понимать, каким образом этот блок взаимодействует с остальными. В ином случае, выявление проблем потребует огромного количества времени и сил или найти ошибку вовсе не удастся. С этим поможет справиться методология графического описания системы.

Для начала следует обратиться к контекстной диаграмме верхнего уровня, изображенной на рис. 3.1, на ней представлена функциональная модель «Продать товар» с четырьмя видами потоков: входные, ставящие определенную задачу, с помощью этих данных задача будет решаться; выходные, определяющие результат деятельности, то, во что преобразовались входные данные; управляющие, некоторые положения или инструкции, законы; механизмы управления данными, необходимые для того, чтобы произвести работу, какие-либо технические средства или же работники.

Разработка данной модели иллюстрирует процессы реализации продажи товаров в электронном магазине, анализ взаимодействия покупателя с товаром, выявление наиболее ходовых продаж и наиболее часто посещающих приложение и регистрировавшихся покупателей.

****

Рисунок 3.1 – Контекстная диаграмма верхнего уровня

На диаграмме представлены четыре входных потока: покупатель, данные о покупателе, товар, данные о товаре. Очень важно иметь сведения о покупателях и знать его предпочтения по товарам, это является частью маркетинга, определение целевой аудитории определенного товара, анализ продаж – позволят в дальнейшем оптимизироваться приложение интернет-магазина, а также окажет влияние на рекламу, например, в социальных сетях. На выходе мы получаем: проданный товар, отчет о заказах, прибыль от продажи. При этом необходимы такие инструкции, как накладные, государственные законы и внутренний регламент. Для реализации всех действий и управления работой: персонал магазина, курьер, база данных.

На рис. 3.2 изображена декомпозиция контекстной диаграммы, состоящая из четырех блоков: «Принимать товар от поставщиков», «Регистрировать заказ», «Подготавливать документацию», «Доставить заказ».



Рисунок 3.2 – Декомпозиция контекстной диаграммы верхнего уровня

Первый блок декомпозиции («Принимать товар от поставщиков») состоит из проверки товара на количество и качество, после этого документы по приему подписываются и заносятся в отчет о заказах и данные вносятся в базу, это выполняет персонал магазина. Декомпозиция данного блока представлена на рис. 3.3.



Рисунок 3.3 – Декомпозиция блока «Принимать товар от поставщиков»

Декомпозиция «Проверить товар на количество» включает в себя четыре компонента: проверка документации, обработка товара, создания плана подсчета товара и подсчет товара. Декомпозиция данного блока представлена на рис. 3.4.

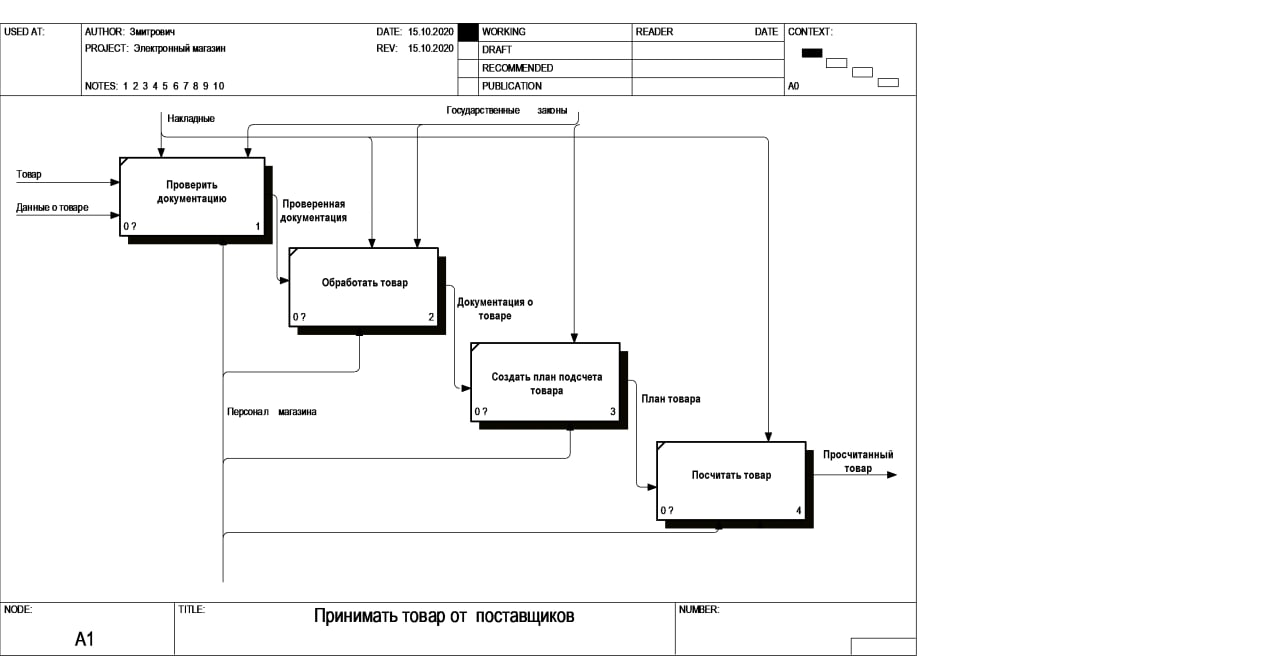


Рисунок 3.4 – Декомпозиция блока «Проверить товар на количество»

Следующий блок декомпозиции («Регистрировать заказ») включает в себя три компонента: получение данных об учетной записи пользователя, сбор заказа, а также добавление данных в документацию магазина. Декомпозиция данного блока представлена на рис. 3.5.



Рисунок 3.5 – Декомпозиция блока «Регистрировать заказ»

Декомпозиция «Подготавливать документацию» включает в себя два компонента: по чеку сверяется и вносится информация в базу о продаже и сохраняется в архивы для дальнейшего анализа сведений о клиентах и товарах, затем выписывается гарантия на товар. Декомпозиция данного блока представлена на рис. 3.6.



Рисунок 3.6 – Декомпозиция блока ««Подготавливать документацию»

И последний блок декомпозиции («Доставить заказ»), в нем осуществляется доставка заказа клиенту, перед этим персонал ищет свободного курьера, готового осуществить доставку и происходит передача ему всего необходимого. Декомпозиция данного блока представлена на рисунке 3.7.



Рисунок 3.7 – Декомпозиция блока ««Доставить заказ»

Модель обладает достойным визуализирующим потенциалом, позволяющим понять суть процесса реализации заказа в определенной предметной области. Заложенные в методологию правила и ограничения позволяют выработать системное и строгое отношение к моделям, что благоприятно сказывается на качестве конечного результата.

**4 ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ И ЕЕ ОПИСАНИЕ**

Информационная модель представляет собой информацию, описывающую необходимые для некоторого процесса (системы) параметры, свойства и характеристики, связи между ними.

Для работы над разработкой системы важно и необходимо понимать модели базового взаимодействия между объектами программы.

Информационная модель системы электронного магазина представлена на рисунке 4.1. Модель проектировалась с помощью базы данных, она приведена к третьей нормальной форме. Это значит, что все атрибуты сущностей являются простыми, все используемые домены содержат только скалярные значения. Каждый не ключевой атрибут неприводимо зависит от первичного ключа. Каждый не ключевой атрибут нетранзитивно зависит от первичного ключа.

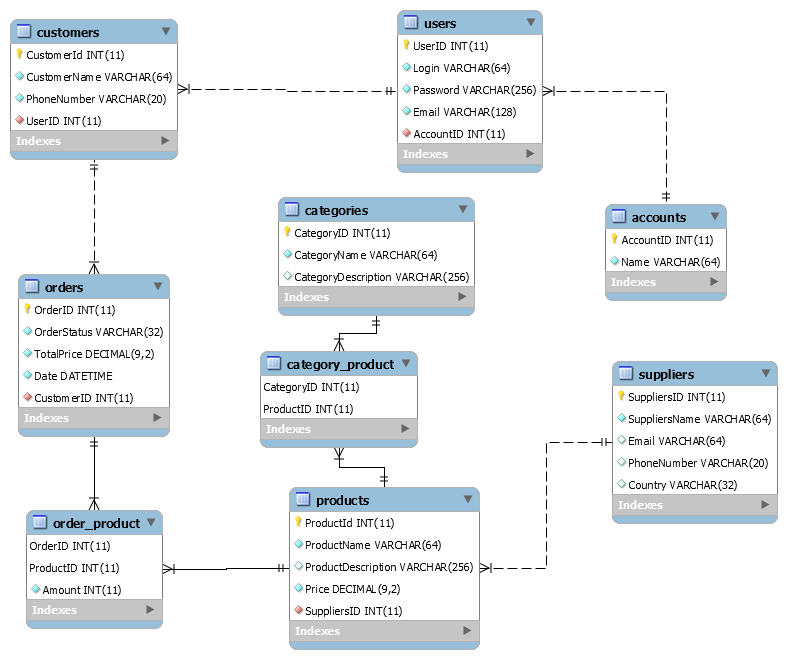


Рисунок 4.1 – Информационная модель системы электронного магазина

Представленная база данных состоит из семи сущностей:

– Accounts (аккаунты). Аккаунт сотрудника магазина отличается от аккаунта покупателя, так как имеет возможности управления товарами. Таким образом, имеется две роли: admin и user.

– Users (пользователи). Хранит данные о зарегистрированных пользователях для их входа в систему.

– Customers (покупатели). Содержит личную информацию о покупателях, такую как имя и номер мобильного телефона.

– Suppliers (поставщики). Хранит информацию обо всех имеющихся поставщиках, а именно: имя, контактный номер, адрес почтового ящика и страну.

– Categories (категории). Хранит информацию о различных категориях, к которым может относиться товар.

– Orders (заказы). Хранит в себе всю необходимую информацию о заказах зарегистрированных пользователей.

– Products (продукция). Товары, которые предлагает для просмотра и покупки магазин электроники.

В результате определения связей между сущностями, было спроектировано 9 таблиц:

* + 1. «Аккаунты». Эта таблица необходима для разделения возможностей аккаунтов. Аккаунт «Admin» позволяет пользователю управлять продажами и иметь доступ к аккаунтам покупателей, также дает возможность просмотра и изучения статистики интернет-магазина. С аккаунта «User» пользователь имеет возможность приобретать товары, то есть совершать покупки, просматривать их. Поля данной таблицы:
* идентификационный номер аккаунта, являющийся уникальным (первичный ключ).
* наименование, характеризующее привилегии аккаунта.
  + 1. «Пользователи». Данная таблица содержит информацию, которая необходима для входа в систему: уникальный логин и пароль, в шифрованном виде и адрес электронной почты. В ней хранятся данные как покупателей, так и сотрудников. Данная таблица связана с таблицей «Аккаунты» связью один к одному, так как лишь одному аккаунту могут соответствовать определенны данные и наоборот. Таблица содержит следующие поля:
* идентификационный номер пользователя, являющийся уникальным (первичный ключ).
* логин, уникальный набор символов.
* пароль, уникальный набор символов, необходимый для входа.
* электронная почта пользователя, необходимая для связи с клиентом и новостных рассылок для него.
* идентификационный номер аккаунта, к которому привязан пользователь.
  + 1. «Покупатели». Таблица содержит личную информацию о пользователях, которая заполняется при регистрации аккаунта. Данная таблица связана с таблицей «Пользователи» связью один к одному, так как личные данные одного человека закрепляются за одним аккаунтом с данными для входа. Содержит поля:
* идентификационный номер покупателя, являющийся уникальным (первичный ключ).
* имя покупателя.
* номер телефона покупателя.
* идентификационный номер пользователя, к которому привязан покупатель.
  + 1. «Поставщики». Таблица содержит информацию обо всех поставщиках, которые когда-либо были у магазина. Данные о них важны для статистики и анализа. Таблица содержит следующие поля:

− идентификационный номер поставщика, являющийся уникальным (первичный ключ).

* имя поставщика или название компании.
* электронная почта.
* контактный номер телефон.
* страна-производитель/поставщик
  + 1. «Категории». Таблица содержит различные категории товаров. Данная таблица связана с таблицей «Продукты» связью многие ко многим, так как в одной категории может быть много продуктов, один продукт может находиться в нескольких категориях. Таблица содержит такие поля, как:
* идентификационный номер, являющийся уникальным (первичный ключ).
* название категории.
* описание товара, находящегося в определенной категории.
  + 1. «Продукты». Таблица содержит информацию обо всех имеющихся в наличии товарах. Данная таблица связана с таблицей «Категории» связью многие ко многим, что было объяснено в описании предыдущей таблицы. Также с таблицей «Поставщики» связью многие ко многим, так как один поставщик может продавать сразу несколько продуктов, в то же время у одного продукта может быть несколько поставщиков. Таблица содержит поля:
* идентификационный номер продукта, являющийся уникальным (первичный ключ).
* название продукта.
* описание продукта, его основные характеристики.
* стоимость продукта в рублях.
* идентификационный номер поставщика.
  + 1. «Заказы». Таблица хранит в себе множество заказов, проданных потребителю. Это необходимо для ведения анализа и статистики интернет-маркетинга. Данная таблица связана с таблицей «Продукты» связью многие ко многим, в одном заказе может находиться много продуктов, один продукт может попасть в разные заказы. Таблица содержит следующие поля:
* идентификационный номер, являющийся уникальным (первичный ключ).
* статус заказа.
* дата сборки заказа.
* идентификационный номер покупателя.
  + 1. «Категория-продукт», таблица, создание которой необходимо для связи многие ко многим между таблицами «Продукты» и «Категории». Таблица содержит следующие поля:
* идентификационный номер продукта.
* идентификационный номер категории.
  + 1. «Заказ-продукт» таблица, создание которой необходимо для связи многие ко многим между таблицами «Заказы» и «Продукты». Таблица содержит следующие поля:
* идентификационный номер заказа.
* идентификационный номер продукта
* количество продуктов в заказе.

**5 МОДЕЛИ ПРЕДСТАВЛЕНИЯ И ИХ ОПИСАНИЕ**

Диаграммы позволяют быстро разобраться в сложных системах, а также служить справочными картами, позволяющими в них ориентироваться, поэтому не стоит забывать про важность их наличия.

**5.1 Диаграмма вариантов использования системы управления контентом магазина электроники**

Диаграмма описывает функциональные требования к системе. В конкретной диаграмме, представленной в приложении В, описывается взаимодействие двух участников, имеющих конкретную цель: покупка товара в магазине (Администратор-Пользователь). Осуществить вход в систему или зарегистрироваться, найти необходимый товар – эти функции являются общими для обоих участников. Только администратор может управлять заказами, просматривать статистику и диаграммы, управлять пользователями. В свою очередь, пользователь может приобрести товар, работать с корзиной.

**5.2 Диаграмма состояний системы управления контентом магазина электроники**

Диаграмма, представленная на рисунке 5.1 демонстрирует состояния относительно пользователя системы.

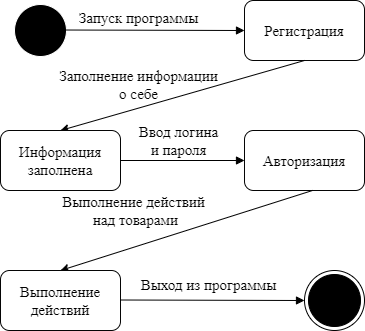


Рисунок 5.1 – Диаграмма состояния относительно пользования

Диаграмма состояний помогает описать поведение отдельно взятого объекта. Она демонстрирует последовательности перехода объекта из одного состояния в другое. Диаграмма состояний показывает все возможные состояния, в которых может находиться объект, а также процесс смены состояний в результате внешнего влияния [7].

**5.3 Диаграмма последовательностей системы управления контентом магазина электроники**

Диаграмма последовательностей, представленная на рис. 5.3 отображает процесс оформления заказа пользователем.

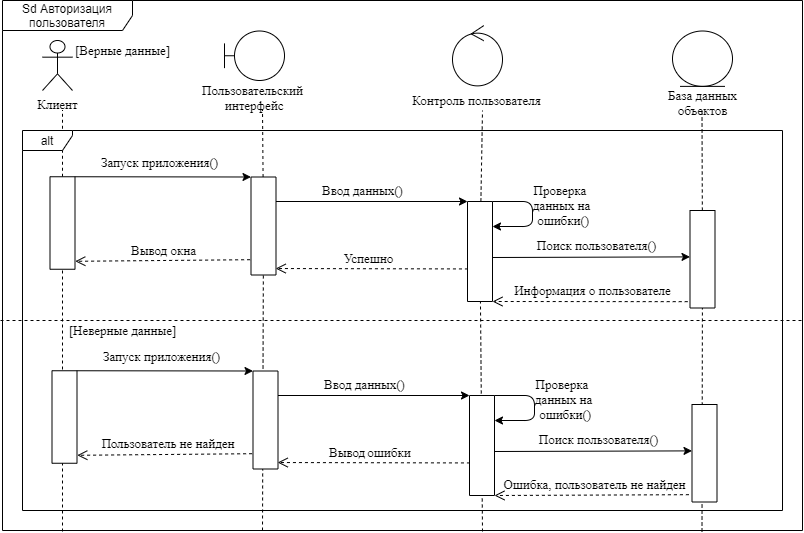


Рисунок 5.2 – Диаграмма последовательности авторизации

Диаграмма последовательностей предназначена для моделирования взаимодействия объектов системы во времени, а также обмена сообщениями между ними.

Одним из основных принципов ООП является способ информационного обмена между элементами системы, выражающийся в отправке и получении сообщений друг от друга. То есть, основные понятия диаграммы последовательности связаны с понятием объект и сообщение.

**5.4 Диаграмма классов системы управления контентом магазина электроники**

Диаграмма классов, позволяет видеть список классов, связи между ними, методы и аргументы.

Диаграмма классов представлена в приложении Г на рисунках Г.1-Г.5.

**5.5 Диаграмма компонентов системы управления контентом магазина электроники**

Данная диаграмма представлена на рисунке 5.3.

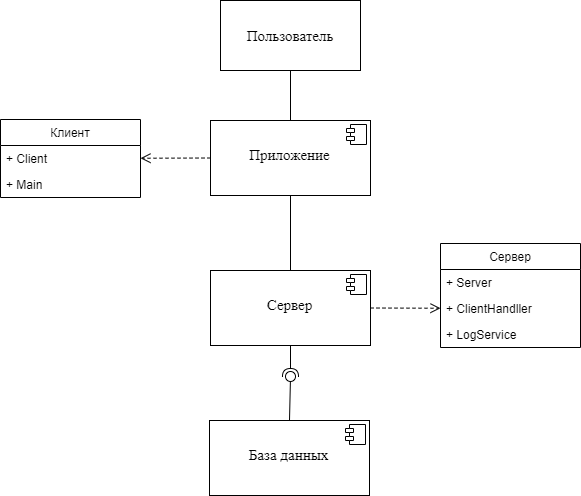


Рисунок 5.3 Диаграмма компонентов

Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами.

**5.6 Диаграмма развертывания системы управления контентом магазина электроники**

Диаграмма развертывания представлена на рис. 5.4



Рисунок 5.4 – Диаграмма развертывания

Диаграмма развёртывания показывает топологию системы и распределение компонентов системы по ее узлам, а также соединения - маршруты передачи информации между аппаратными узлами.

**6 ОПИСАНИЕ АЛГОРИТМОВ, РЕАЛИЗУЮЩИХ БИЗНЕС-ЛОГИКУ СЕРВЕРНОЙ ЧАСТИ ПРОЕКТИРУЕМОЙ СИСТЕМЫ**

Описание алгоритмов представлено с помощью блок-схем. Схемы позволяют графически, доступно и понятно, описывать алгоритмы и процессы, происходящие в программе, которая может решать достаточно сложные задачи, состоящие из множества пунктов. Каждый шаг алгоритма отображается в виде блоков, соединенных между собой линиями, указывающими направление последовательности операций.

Для приобретения товара пользователю требуется найти и выбрать необходимый ему товар, после он добавится в его корзину, затем он может удалить этот товар из корзины, так и не совершив покупку или же приобрести его.

Алгоритм приобретения товара представлен на схеме Д.1 в приложении Д.

После входа программа определяет роль пользователя, является он администратором, покупателем или же его аккаунт заблокирован. Если данный пользователь – покупатель, выводится главное меню с выбором одной из трех вкладок. Если данный пользователь – администратор, в меню пользователя будет четыре вкладки, последняя из них будет отвечать за управление магазином. Если пользователь является заблокированным, выводится сообщение об ошибке, это значит, что он не сможет пользоваться системой.

Алгоритм определения роли представлен на схеме Д.2 в приложении Д.

**7 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

При запуске программы появляется окно, позволяющее пользователю войти или зарегистрироваться (рис.7.1).

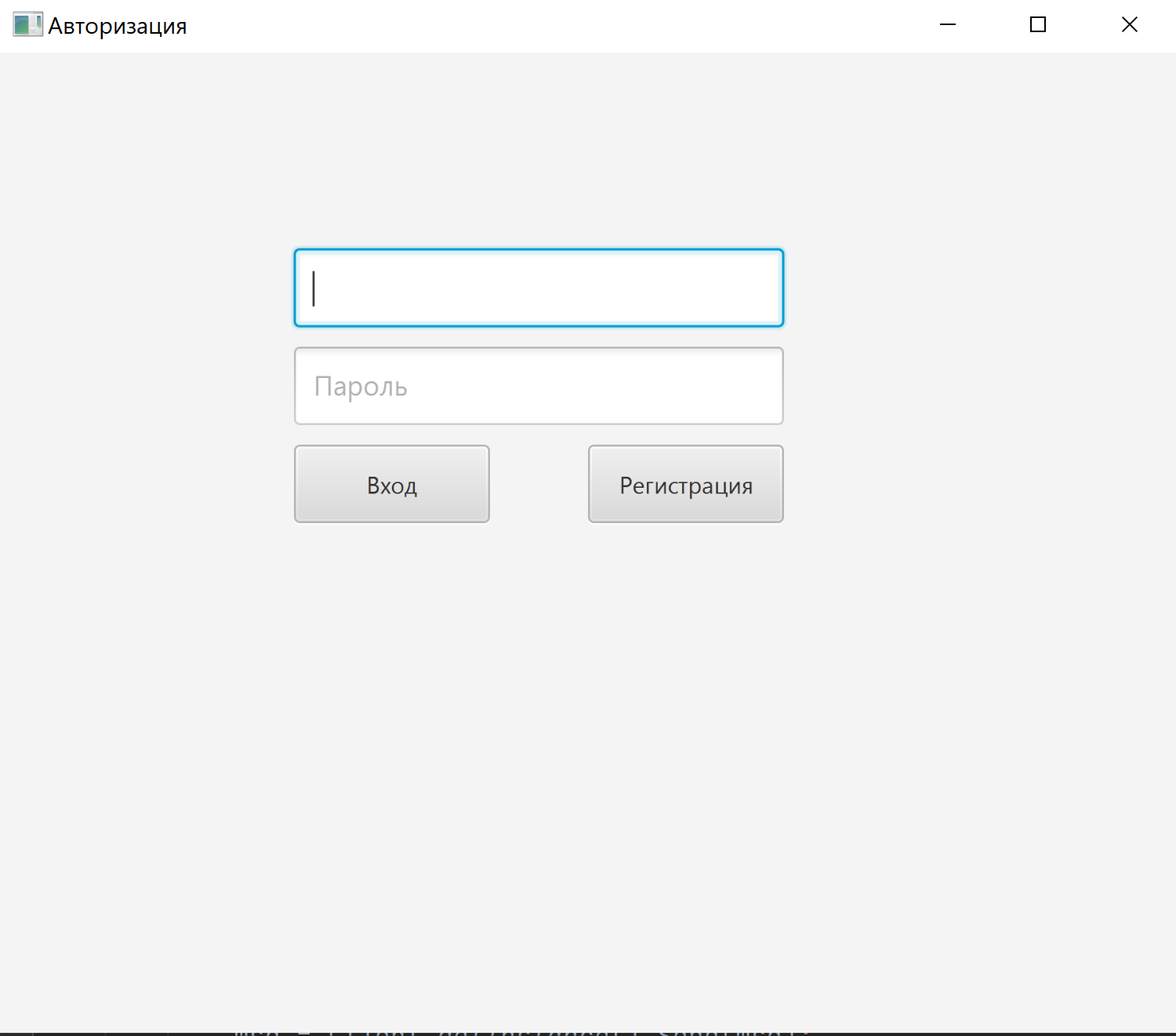


Рисунок 7.1 – Меню входа

Если пользователь выбирает регистрацию, ему необходимо будет придумать логин и пароль, а также заполнить всю необходимую информацию о себе (рис.7.2).

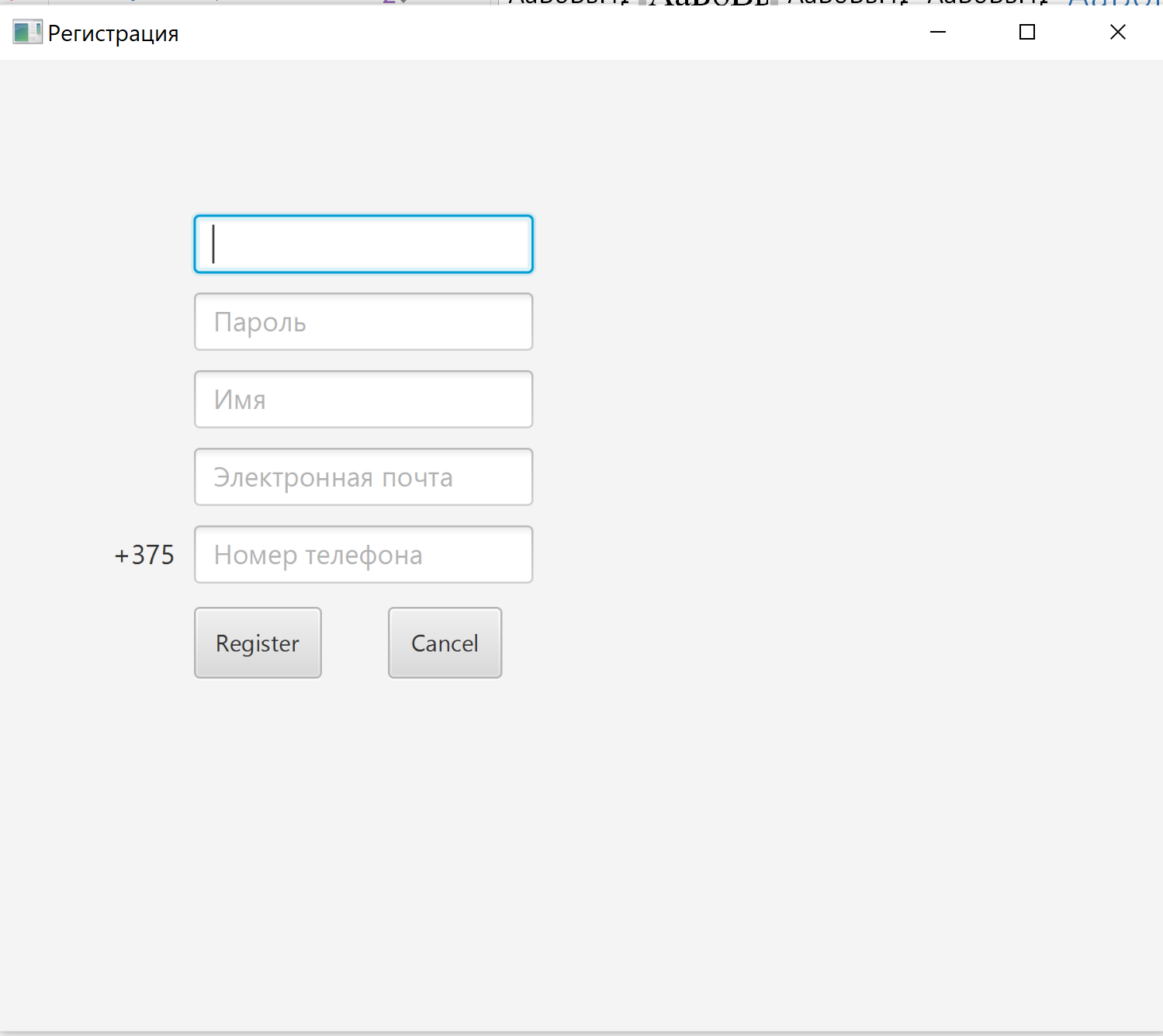


Рисунок 7.2 – Окно для регистрации

После успешной регистрации пользователь может войти в свой аккаунт, после чего появится главное меню программы, состоящее из трех вкладок (рис.7.3). После регистрации каждого пользователя создается файл, содержащий информацию о его действиях в системе ().

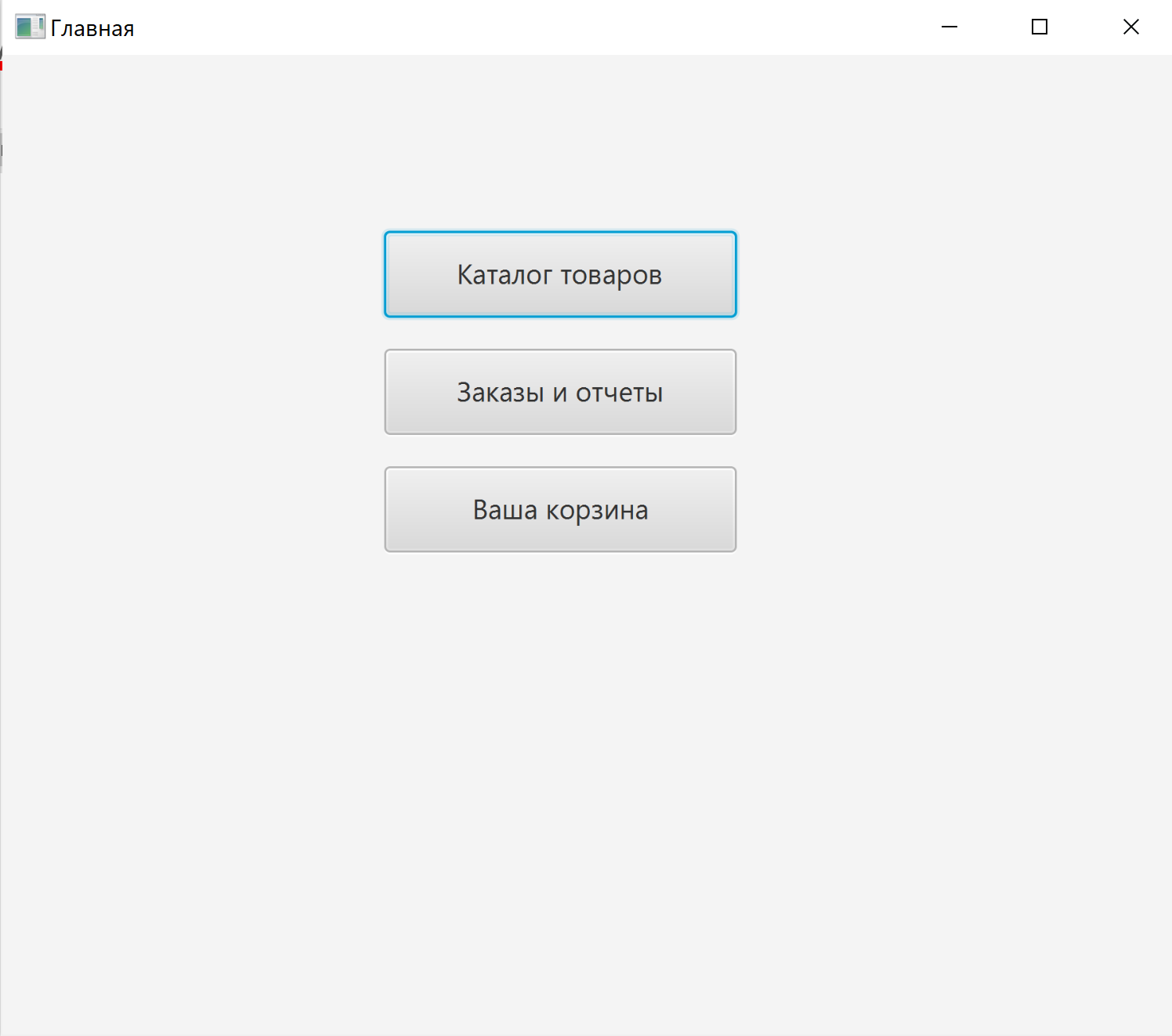


Рисунок 7.3 – Главное меню пользователя

После регистрации каждого пользователя создается файл, содержащий информацию о его действиях в системе (рис.7.4).

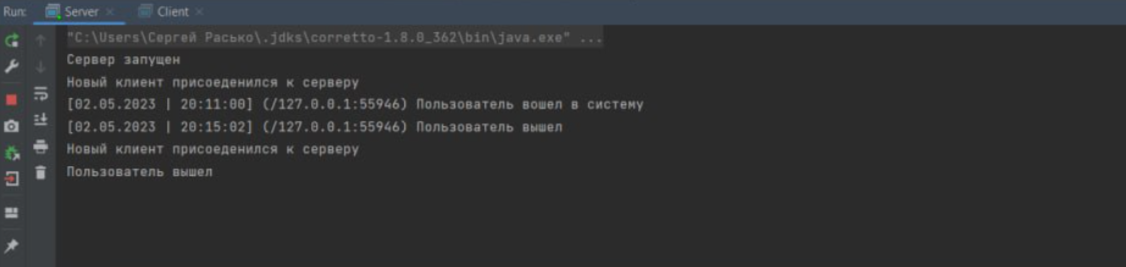


Рисунок 7.4 – Файл с действиями пользователя в системе

При выборе первого пункта меню пользователь может увидеть список товаров, имеющихся в продаже, что представлено на рисунке 7.5.

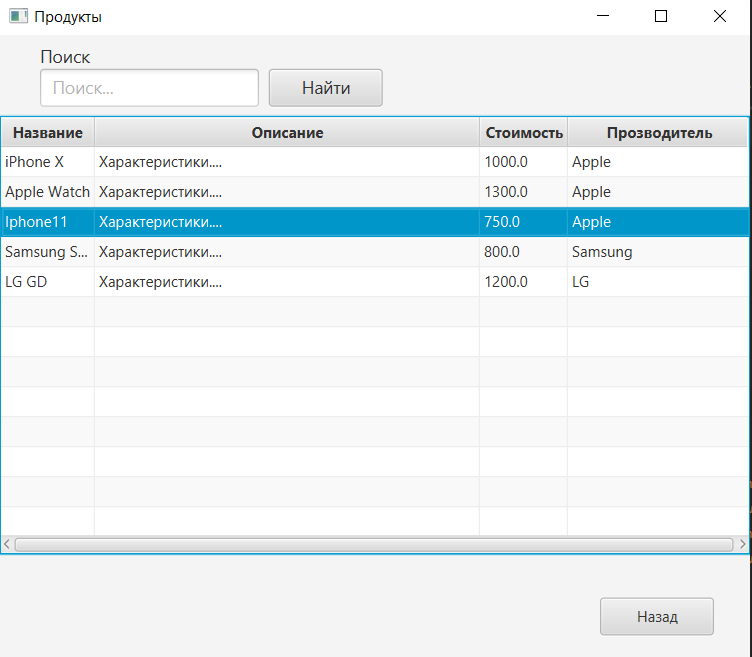


Рисунок 7.5 – Список товаров

С помощью функции поиска по названию продукта его можно найти (рис. 7.6).

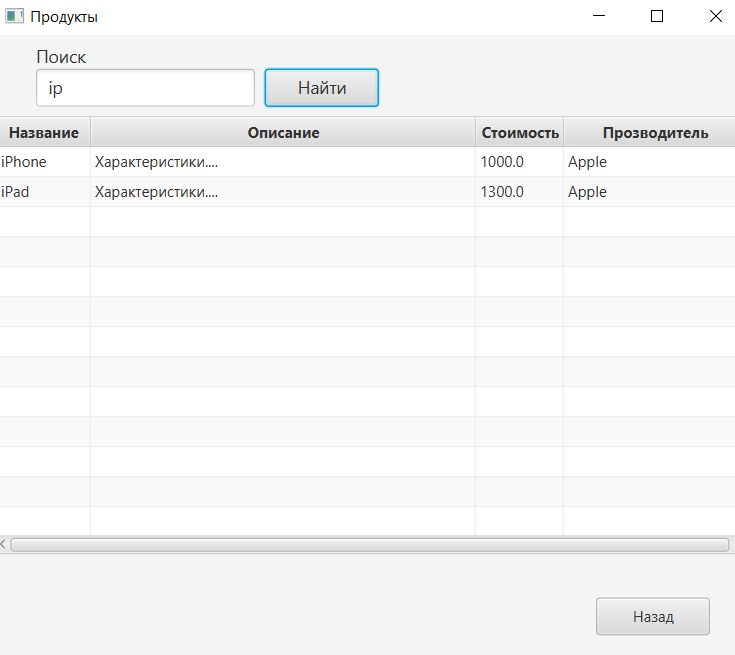


Рисунок 7.6 – Поиск товаров

При двойном нажатии мышью на определенный товар, появится окно, позволяющее выбрать количество продукции, которая добавится в корзину. Данные действия представлены на рисунке 7.7.

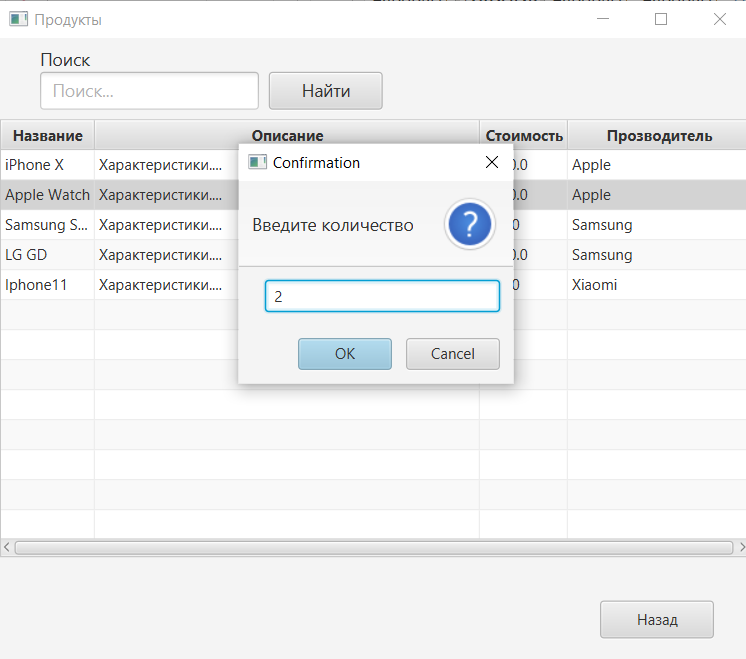


Рисунок 7.7 – Добавление товара в корзину

С этого момента во вкладке меню «Ваша корзина» появится только что добавленный товар (рис.7.8).

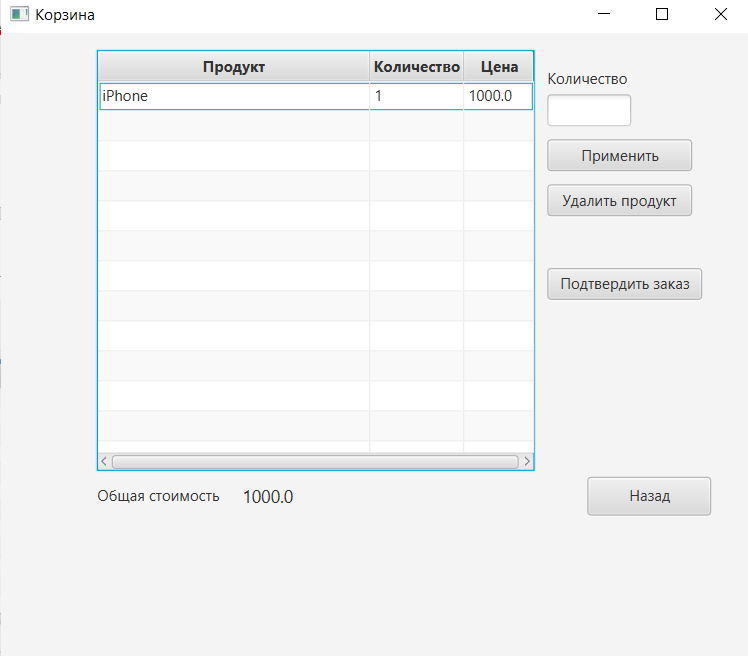


Рисунок 7.8 – Корзина пользователя

Количество товара можно изменить, нажав после этого кнопку выбора. После подтверждения данный товар перестанет находиться в корзине и будет являться заказом, информацию о котором можно просмотреть во втором пункте меню (рис. 7.9). Так же товар из корзины можно удалить.

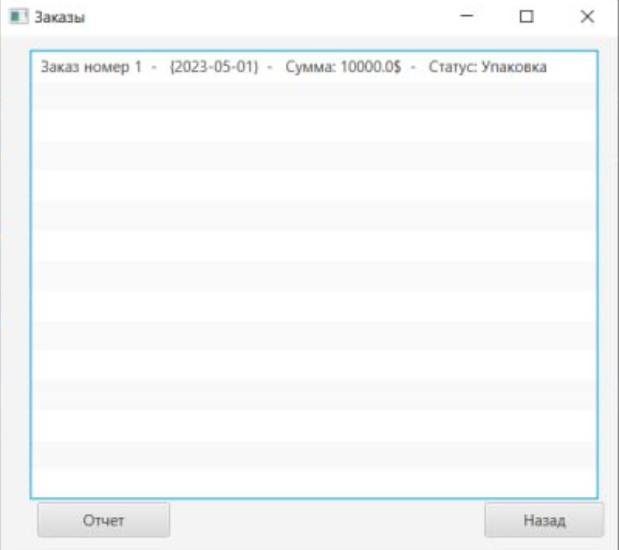


Рисунок 7.9 – Информация о заказах

При нажатии на кнопку отчет, появится окно с информацией о заказе и покупателе (рис. 7.10). Этот отчет можно сохранить на компьютер в отдельный файл (рис. 7.11).

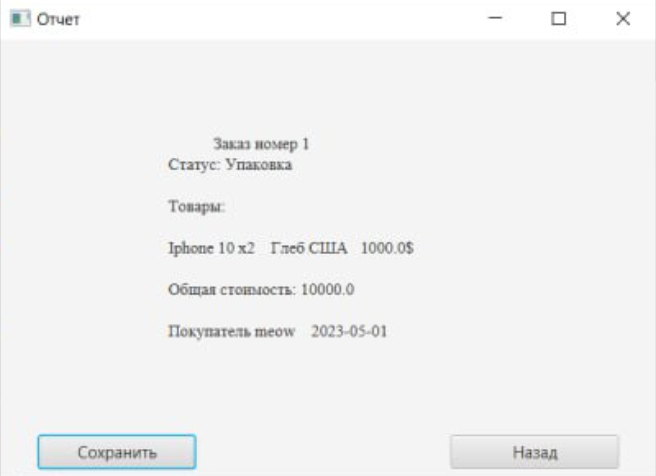


Рисунок 7.10 – Отчет по заказу

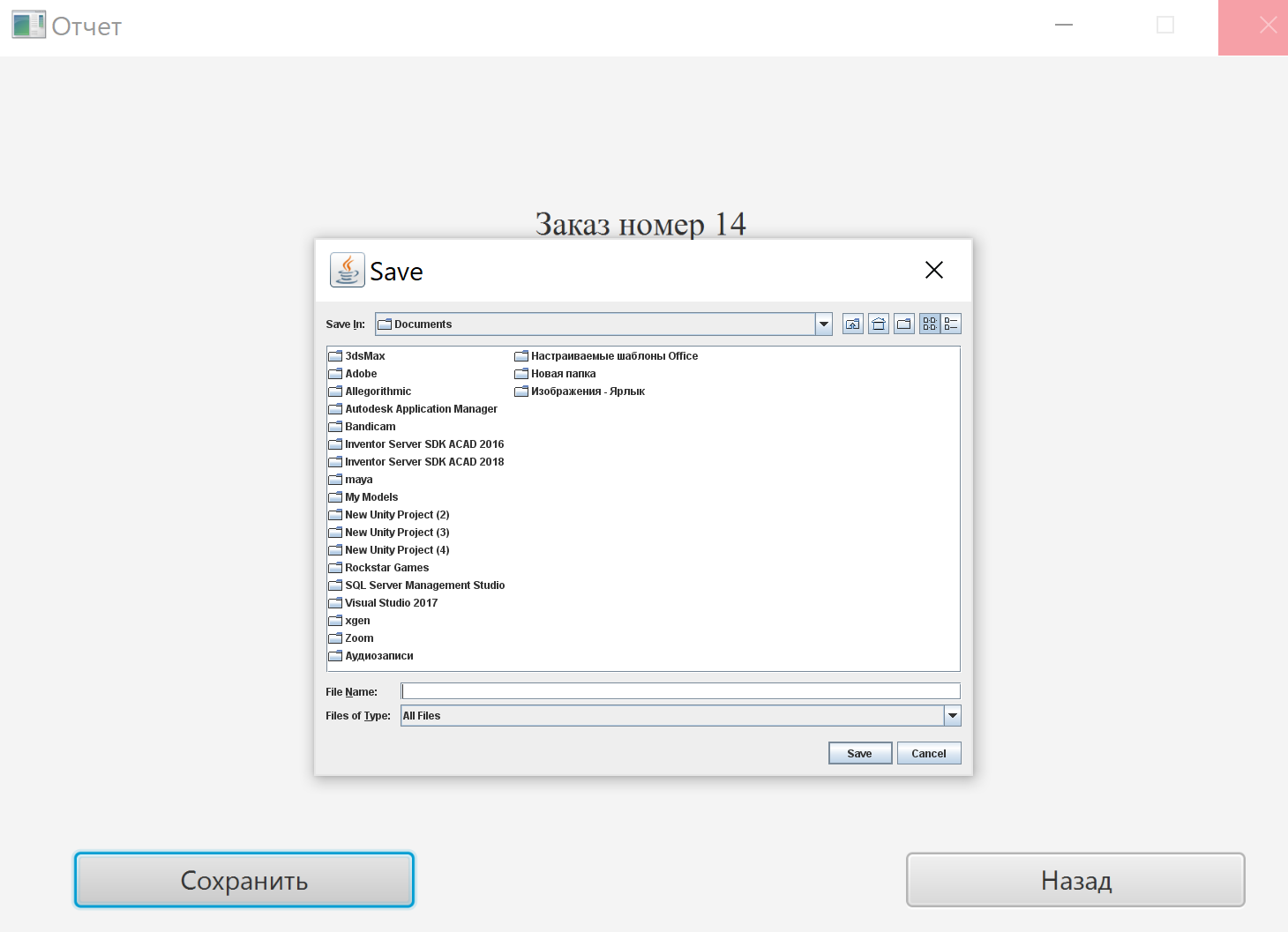


Рисунок 7.11 – Сохранение отчета

На этом функции покупателя заканчиваются, но, если зайти в систему в качестве администратора будет доступен четвертый пункт из меню – управление (рис. 7.12).

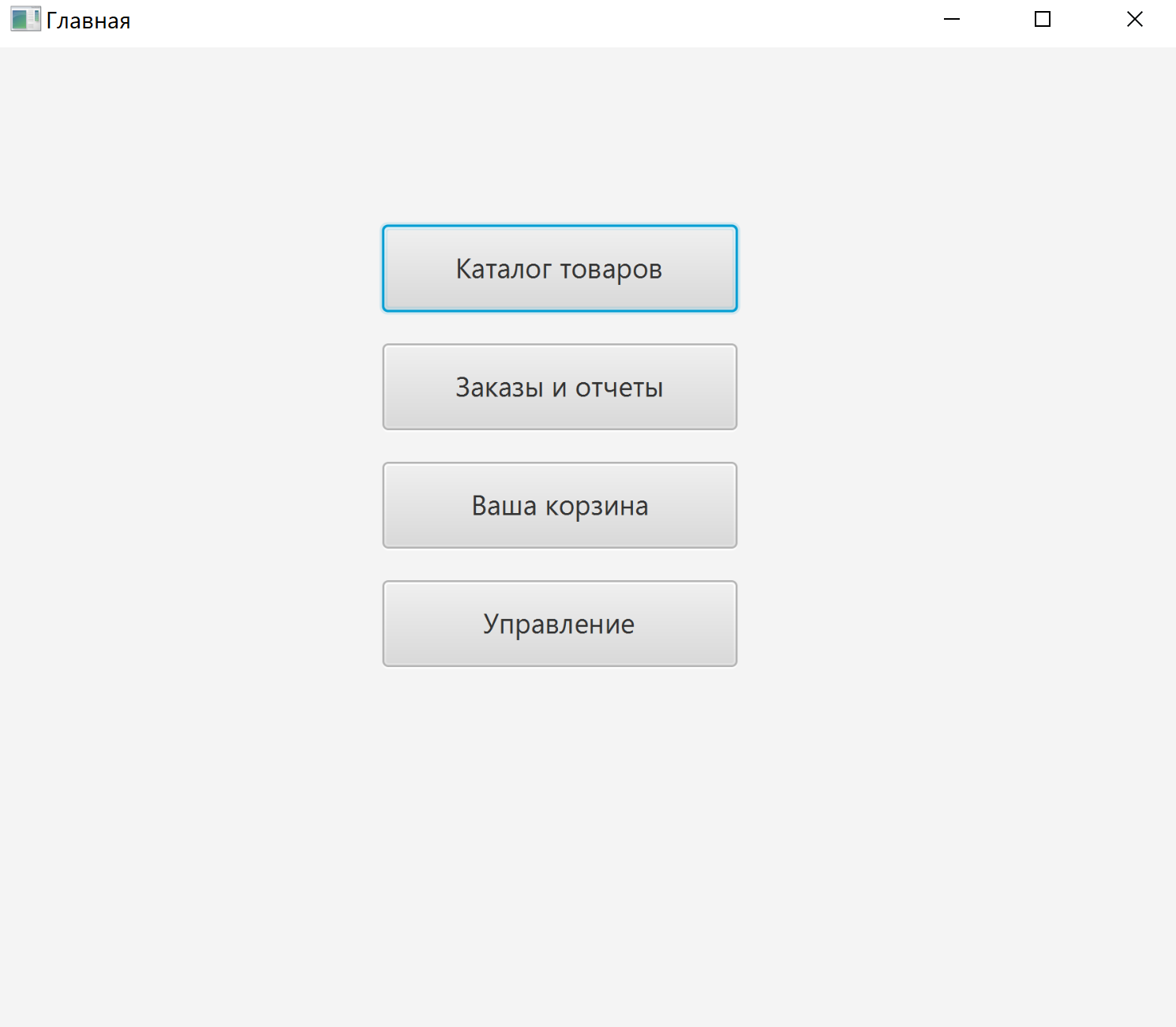


Рисунок 7.12 – Главное меню администратора

На первой вкладке администратору видны все зарегистрированные в системе пользователи и нажатием на правую кнопку мыши, вызывается контекстное меню, с помощью которого можно изменить роль пользователя, а также заблокировать его или разблокировать (рис. 7.13).

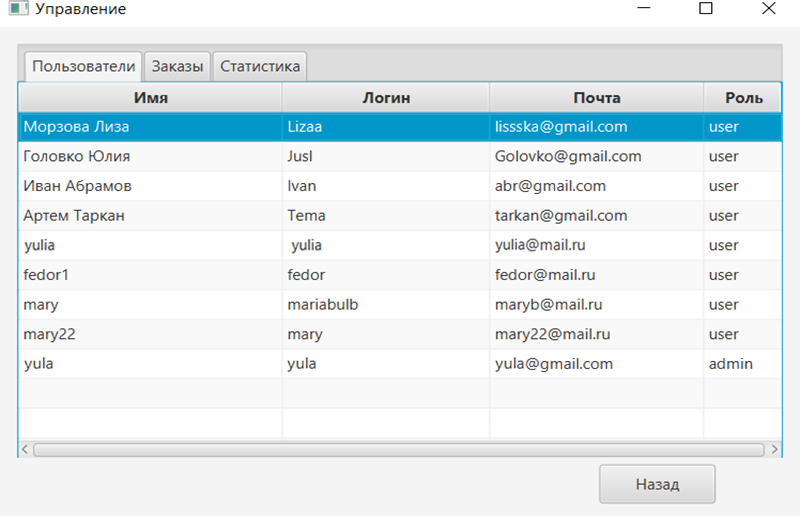


Рисунок 7.13 – Информация о пользователях

Во вкладке со статистикой администратор может посмотреть диаграмму, показывающую, популярность товаров магазина. Она строится относительно единиц товара, когда-либо купленных в магазине. Данная диаграмма представлена на рис. 7.14.

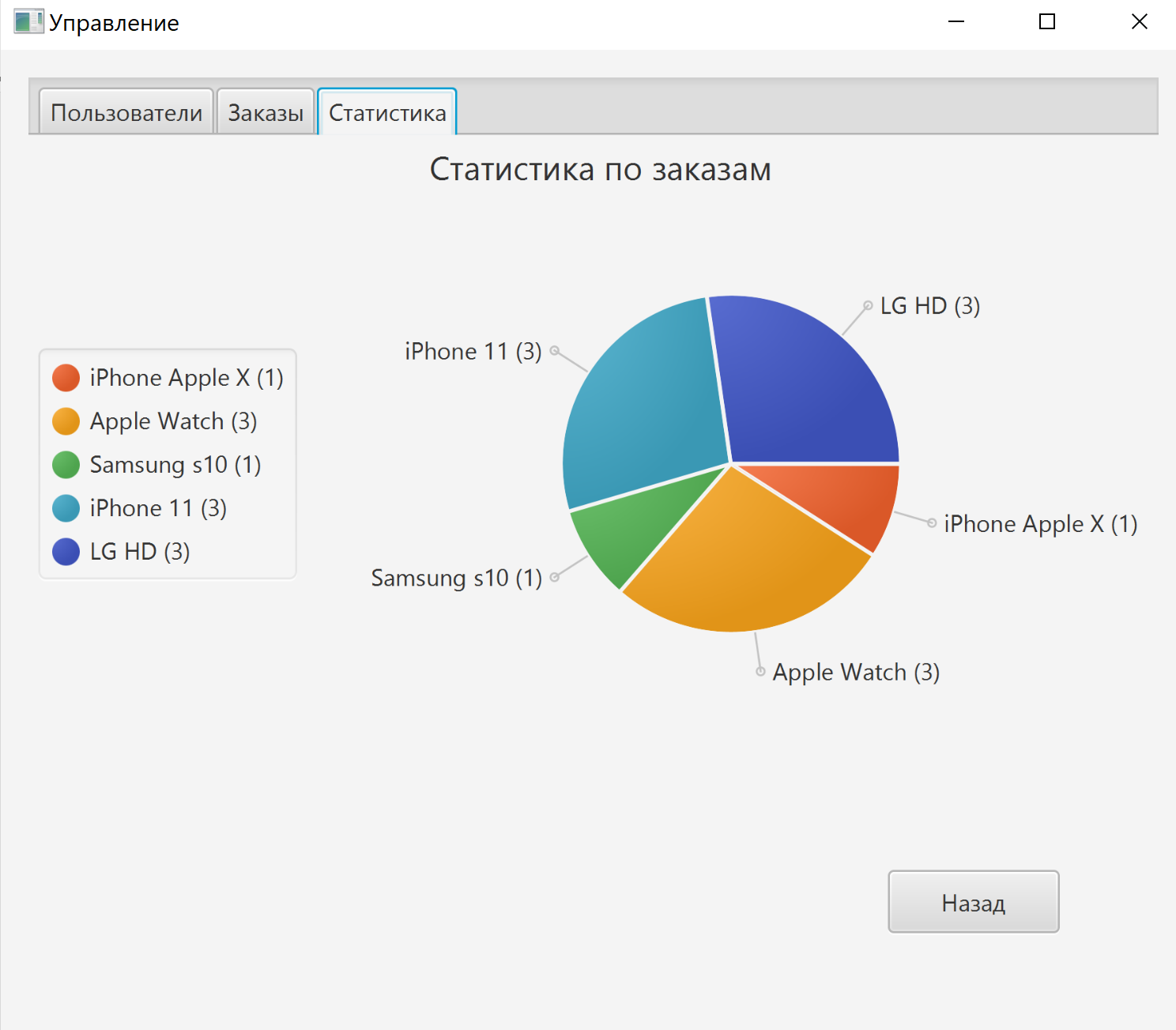


Рисунок 7.14 – Статистика по заказам

Так же на каждом экране имеется клавиша выхода, возвращающая пользователя в предыдущее окно.

**8 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ РАЗРАБОТАННОЙ СИСТЕМЫ**

В данной главе будут рассмотрены случаи ошибочного ввода данных или ошибочных действий и реагирование системы на них. Система должна сообщить о некорректных действиях и оповестить пользователя сообщением об ошибке, чтобы после правильного действия можно было продолжить работу правильно.

Первая ошибка, с которой можно столкнуться на этапе регистрации – недостаточная длина логина, что при этом происходит изображено на рис. 8.1. А также длина пароля (рис. 8.2), формат почтового ящика (рис. 8.3) и формат мобильного телефона (рис. 8.4).

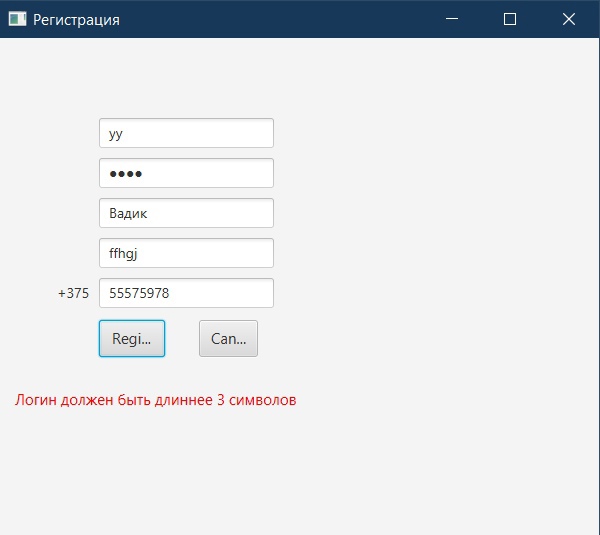


Рисунок 8.1 – Неправильный ввод логина

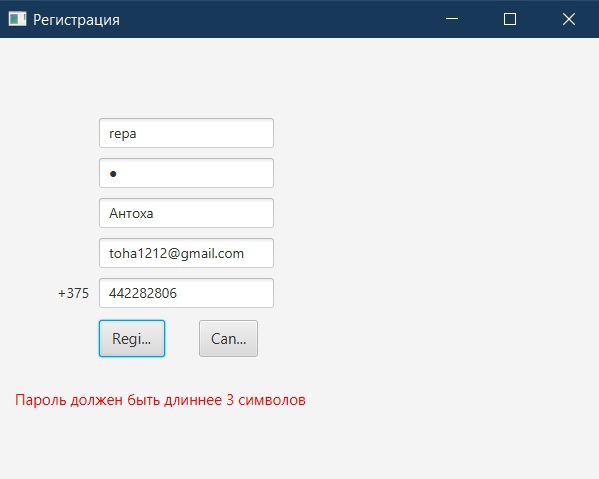


Рисунок 8.2 – Неправильный ввод пароля

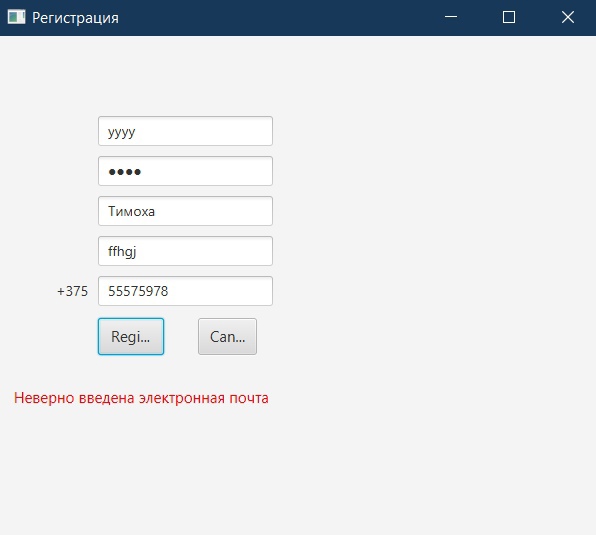


Рисунок 8.3 – Ввод неправильного формата электронной почты

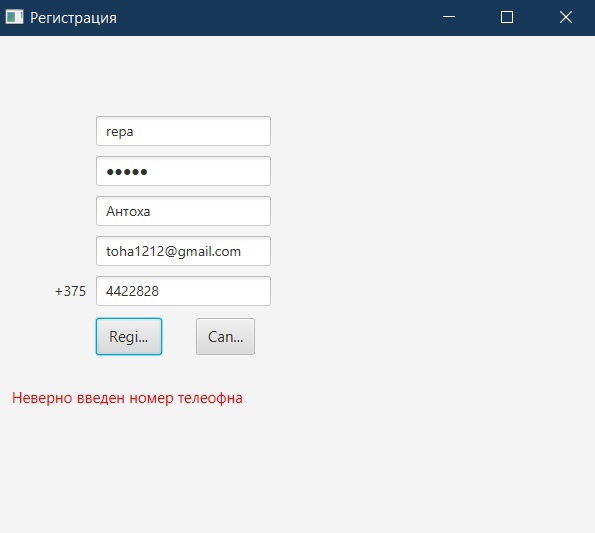


Рисунок 8.4 – Ввод неправильного формата мобильного телефона

При входе пользователь может ввести неверные данные о логине или пароле (рис. 8.5), после любой из этих ошибок он может повторить свои действия.

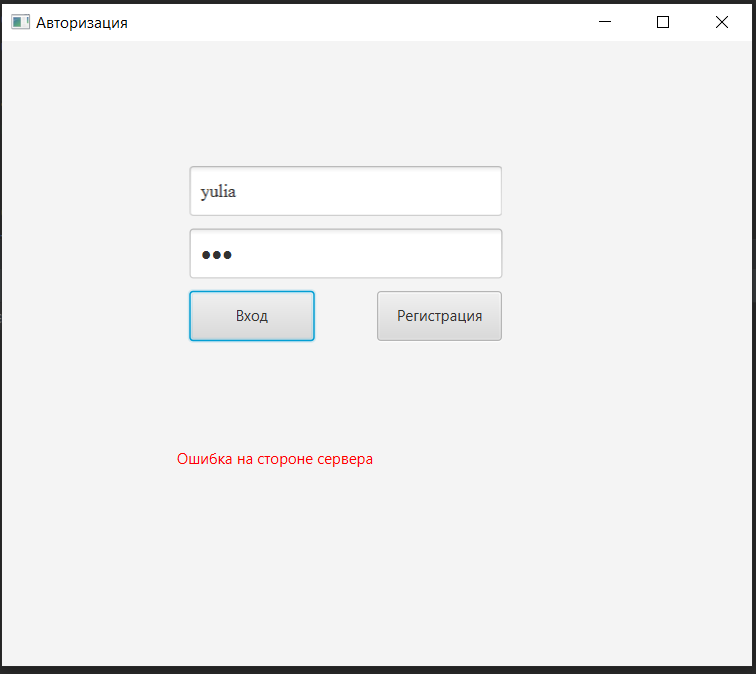


Рисунок 8.5 – Ввод несуществующего логина или пароля

Также окно с ошибкой появится, если пользователь попробует собрать заказ, если корзина пуста. Данное действие изображено на рисунке 8.6.

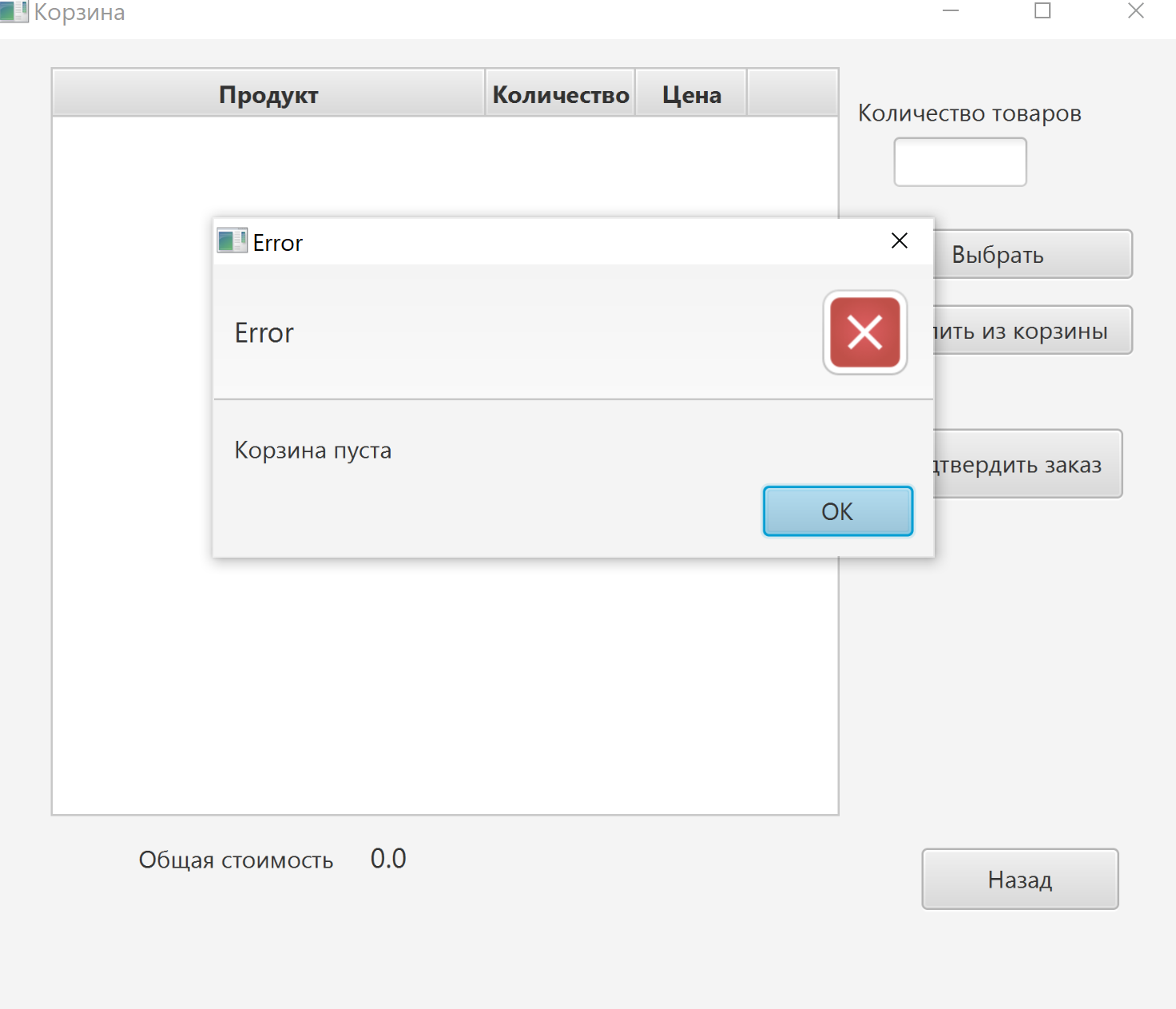


Рисунок 8.6 – Недоступная операция с корзиной

**ЗАКЛЮЧЕНИЕ**

В результате выполнения данной курсовой работы было разработано графическое приложение на языке Java, позволяющее управлять деятельностью электронного магазина. Программа, созданная в рамках курсового проекта, является актуальным средством автоматизации и оптимизации, который позволяет грамотно и эффективно распределять должностные полномочия, человеческий ресурс и занятость сотрудников, работающих в магазине.

В результате написания программы был упрощен процесс контроля товаров. Программный продукт представляет собой удобный и эффективный способ оформления заказов, хранения имеющихся товаров, их статистики и управление ими. Информационная система работает вместе с файловой системой как хранилищем данных, записывая информацию в базу данных, что делает данный метод наименее требовательным к ресурсам компьютера и системы в целом, а также для взаимодействия между группами реализованных в проекте объектов. Разработанное шифрование обеспечивает надежную защиту данных в случае попытки злоумышленников их заполучить.

В программе были эффективно использованы и закреплены знания в области объектно-ориентированного программирования и языка Java, полученные в ходе изучения университетского курса дисциплины «Программирование сетевых приложений». Широко использовались основные средства языка Java, позволяющее результативно реализовывать данный проект, и возможности интегрированной среды разработки программного обеспечения, что обеспечило удобство в написании исходного кода программы.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://beseller.by/](http://usu.kz/index.php)
2. [Электронный ресурс]. – Электронные данные. – Режим доступа: https://rusability.ru/internet-marketing/
3. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://metanit.com/java/javafx/>
4. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://habr.com/ru>
5. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://ru.wikipedia.org/>
6. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://refactoring.guru/ru/design-patterns/strategy>
7. [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://it-gost.ru/>
8. Шилдт, Герберт Java 8. Руководство для начинающих / Пер. с англ. – М.: ООО «И. Д. Вильямс», 2015. – 720 с.

**ПРИЛОЖЕНИЕ A**

**(обязательное)**

**Диаграмма вариантов использования**

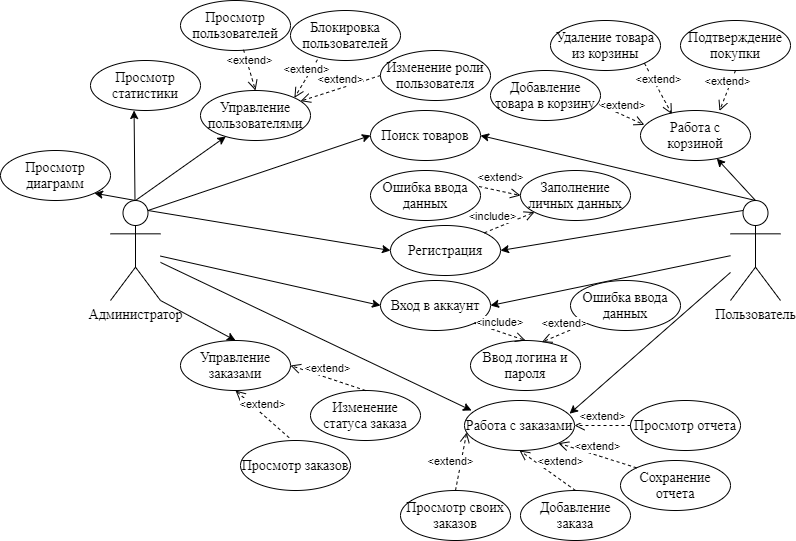
****

Рисунок А.1 – Диаграмма вариантов использования

**ПРИЛОЖЕНИЕ Б**

**(обязательное)**

**Диаграммы классов**

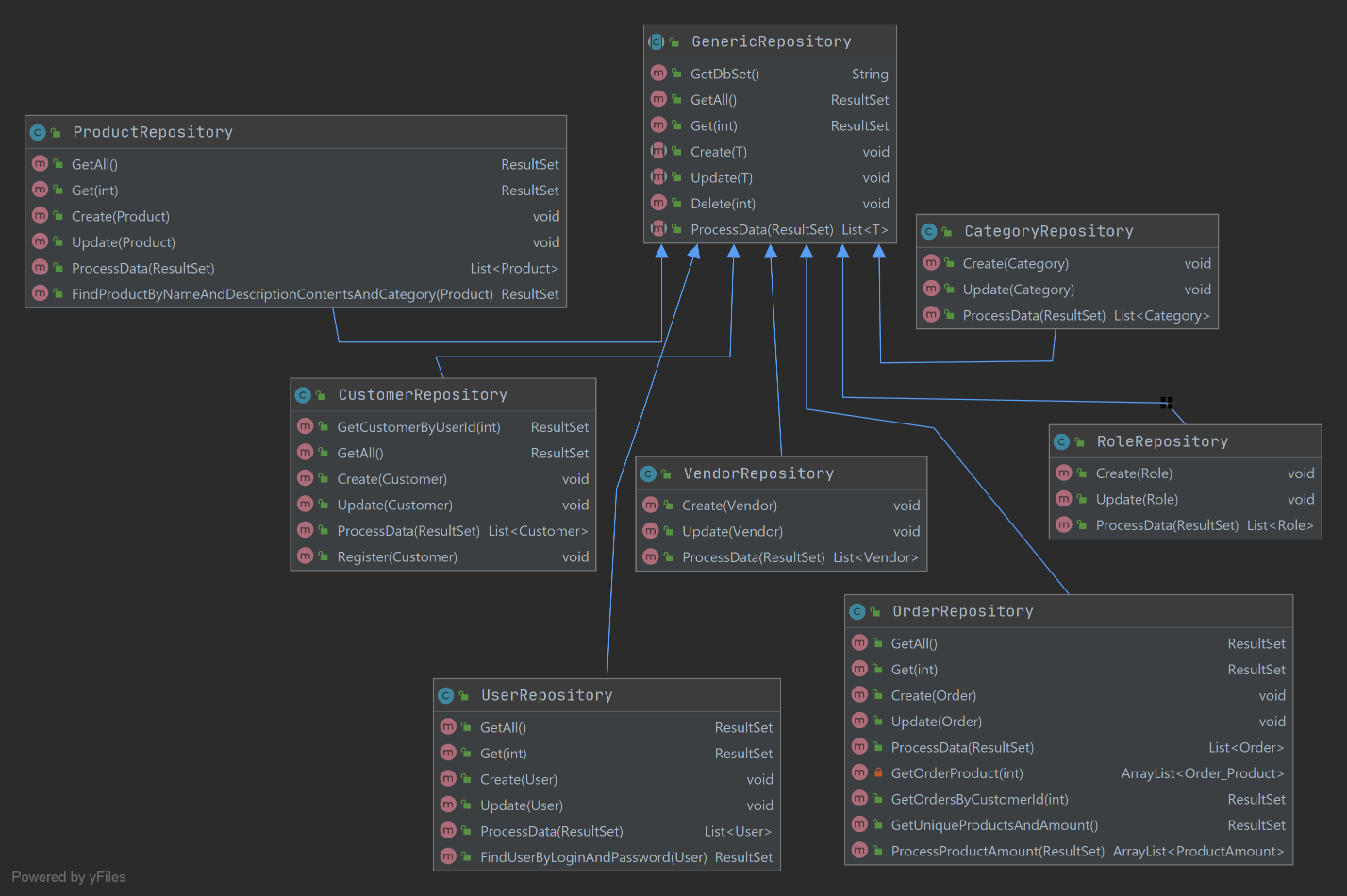
****

Рисунок Б.1 – Диаграмма классов пакета Repositories

****

Рисунок Б.2 – Диаграмма классов пакета Models

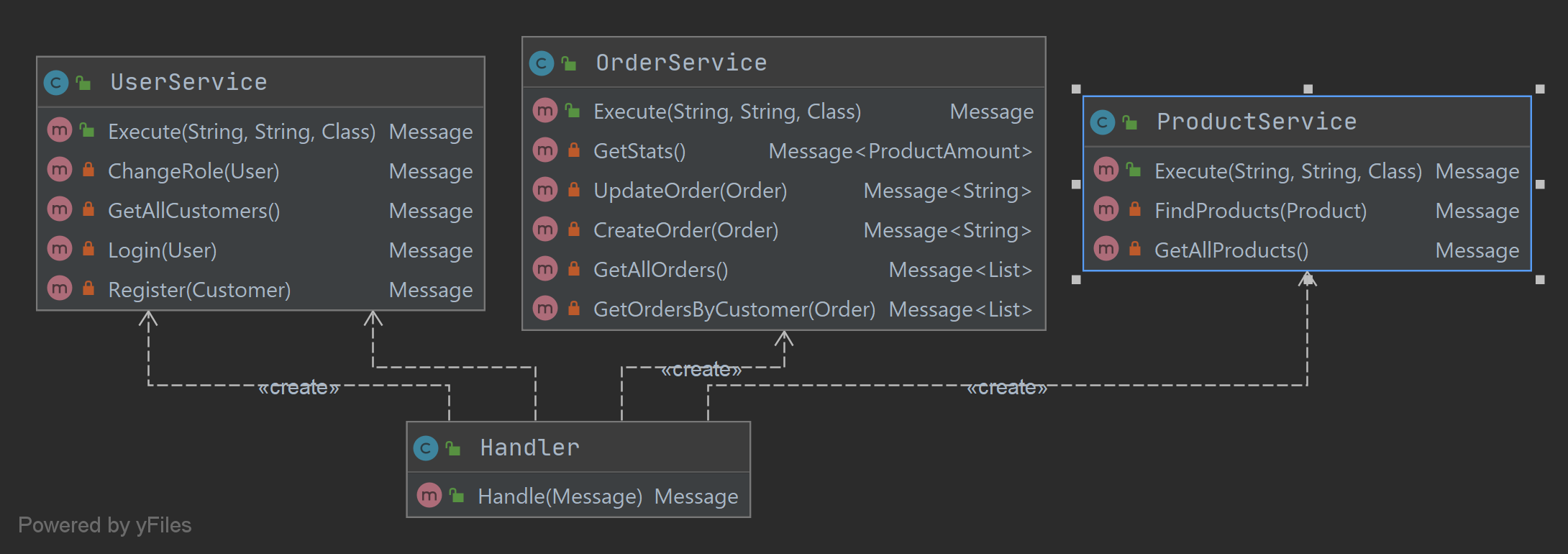
****

Рисунок Б.3 – Диаграмма классов пакета Services

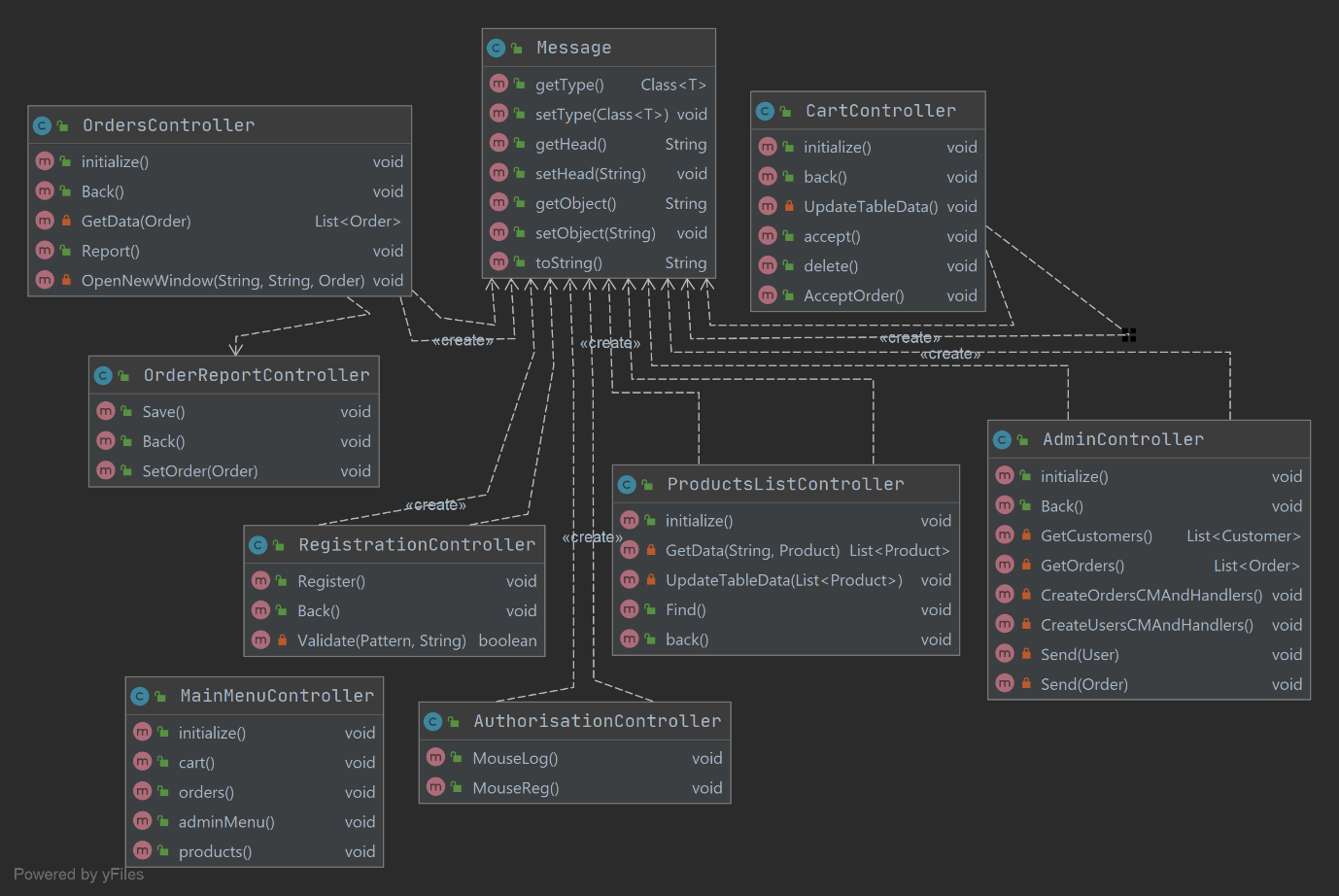
****

Рисунок Б.4 – Диаграмма классов пакета Controllers

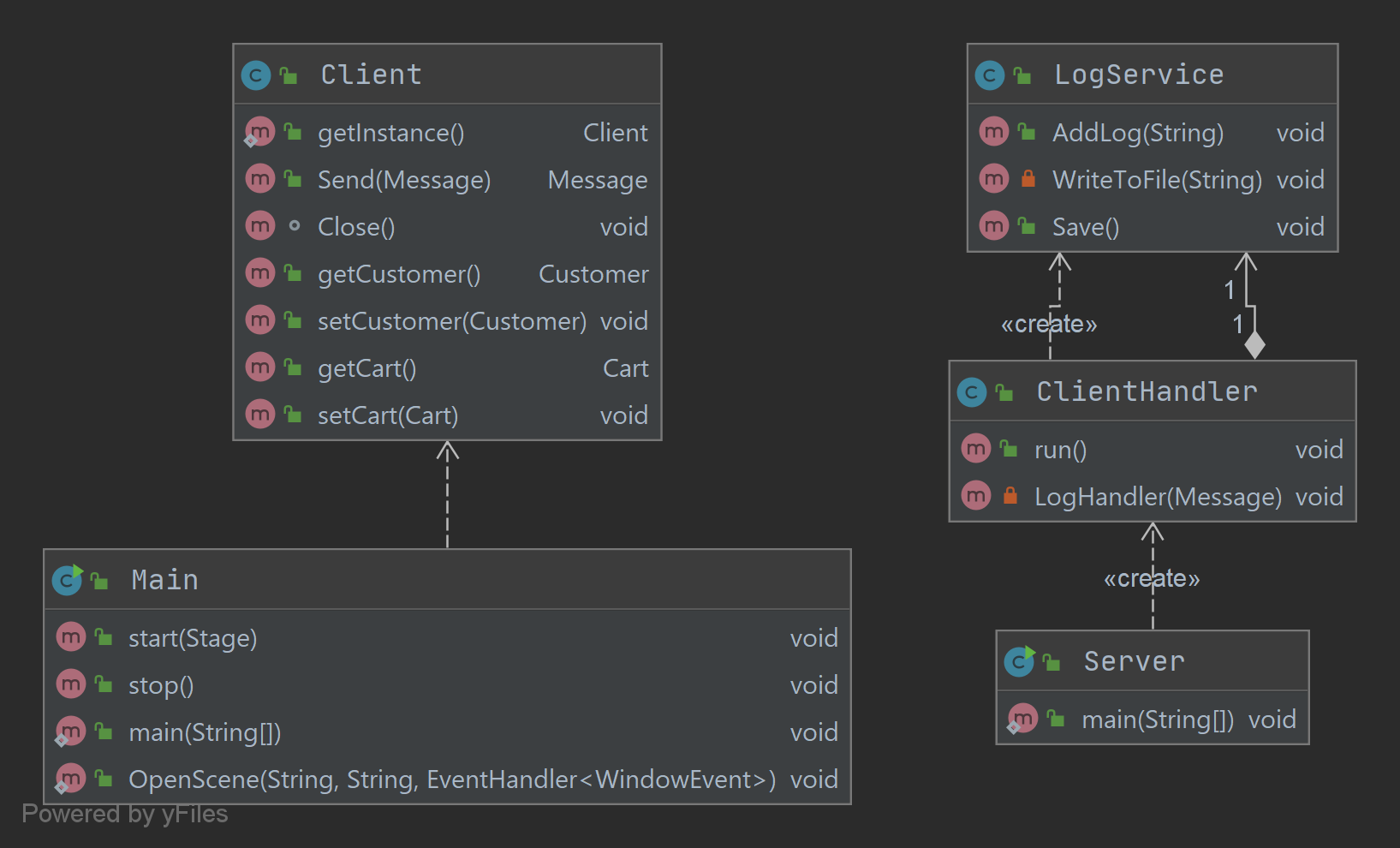


Рисунок Б.5 – Диаграмма классов пакета Клиент и Сервер

**ПРИЛОЖЕНИЕ В**

**(обязательное)**

**Схемы алгоритмов**

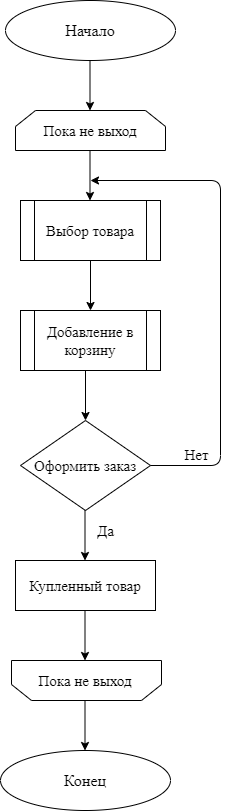


Схема В.1 – Блок-схема алгоритма приобретения товара

Продолжение приложения В

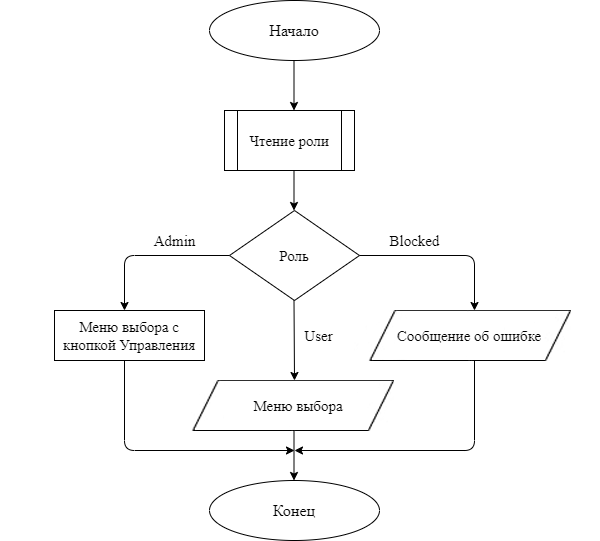


Схема В.2 – Блок - схема алгоритма определения роли

**ПРИЛЖЕНИЕ Г**

**(обязательное)**

**Листинг кода**

package Shop.Infrastructure.Server;  
import java.io.IOException;  
import java.net.ServerSocket;  
  
public class Server  
{  
 public static void main(String[] args)  
 {  
 try  
 {  
 ServerSocket ss = new ServerSocket(7777);  
 System.out.println("Сервер запущен");  
 String connectionString = "jdbc:mysql://localhost:3306/Shop?verifyServerCertificate=false" +  
 "&useSSL=false&requireSSL=false&useLegacyDateTimeCode=false&amp&serverTimezone=UTC"+"&allowPublicKeyRetrieval=true";  
 //( "root", "1234");// сделать подключение к базе данных  
 //...  
  
 }  
 catch (IOException e)  
 {  
 e.printStackTrace();  
 }  
 }  
}

package Shop.Infrastructure.Client;  
  
import Shop.Models.Customer;  
import Shop.Infrastructure.Models.Message;  
import com.fasterxml.jackson.databind.ObjectMapper;  
  
import java.io.DataInputStream;  
import java.io.DataOutputStream;  
import java.io.IOException;  
import java.net.InetAddress;  
import java.net.Socket;  
  
public class Client

Продолжение приложения Г

{  
 private static Client inst;  
  
 private Socket socket;  
 private DataInputStream input;  
 private DataOutputStream output;  
  
 private Customer customer = null;  
  
  
 public static Client getInstance() throws IOException  
 {  
 if(inst == null)  
 {  
 inst = new Client();  
 inst.socket = new Socket(InetAddress.getByName("localhost"), 7777);  
 inst.input = new DataInputStream(inst.socket.getInputStream());  
 inst.output = new DataOutputStream(inst.socket.getOutputStream());  
 }  
 return inst;  
 }  
  
 public Message Send(Message message) throws IOException  
 {  
 String messageStr = message.toString();  
 output.writeUTF(messageStr);  
 if(message.getHead().equals("EXIT"))  
 return null;  
 String response = input.readUTF();  
 ObjectMapper mapper = new ObjectMapper();//jackson  
 return mapper.readValue(response, Message.class);  
 }  
  
 void Close() throws IOException  
 {  
 Send(new Message<>("EXIT", String.class, "EXIT"));  
 socket.close();  
 input.close();  
 output.close();  
 }  
  
 public void setCustomer(Customer customer)

Продолжение приложения Г

{  
 this.customer = customer;

public Customer getCustomer()  
 {  
 return customer;  
 }  
}

package Shop.Infrastructure.Client;  
  
import javafx.application.Application;  
import javafx.event.EventHandler;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.stage.Stage;  
import javafx.stage.WindowEvent;  
  
import java.io.IOException;  
  
public class Main extends Application  
{  
 private static Stage stage;  
 @Override  
 public void start(Stage stage)  
 {  
 Main.stage = stage;  
 OpenScene("../../GUI/AuthorisationGUI.fxml", "Авторизация", null);  
 }  
 @Override  
 public void stop()  
 {  
 try  
 {  
 Client.getInstance().Close();  
 } catch (IOException e)  
 {  
 e.printStackTrace();  
 }  
 }  
  
 public static void main(String[] args)  
 {  
 launch(args);

Продолжение приложения Г

}  
 public static void OpenScene(String path, String title, EventHandler<WindowEvent> onShowing)

try  
 {  
 Parent root = FXMLLoader.load(Main.class.getResource(path));  
 stage.setTitle(title);  
 assert root != null;  
 stage.setScene(new Scene(root, 600, 500));  
 stage.show();  
 if(onShowing != null)  
 stage.setOnShowing(onShowing);  
 }  
 catch (IOException e)  
 {  
 e.printStackTrace();  
 }  
 }  
}

package Shop.Models;  
  
import com.fasterxml.jackson.annotation.JsonIgnore;  
  
public class Customer  
{  
 private int id;  
 private String name;  
 private String phoneNumber;  
  
 private User user;  
  
 public Customer() { }  
 public Customer(int id, String name, String phoneNumber, User user)  
 {  
 this.id = id;  
 this.name = name;  
 this.phoneNumber = phoneNumber;  
 this.user = user;  
 }  
  
 public void setId(int id)

Продолжение приложения Г

{  
 this.id = id;

public int getId()  
 {  
 return id;  
 }  
  
 public void setName(String name)  
 {  
 this.name = name;  
 }  
 public String getName()  
 {  
 return name;  
 }

public void setPhoneNumber(String phoneNumber)  
 {  
 this.phoneNumber = phoneNumber;  
 }  
  
 public String getPhoneNumber()  
 {  
 return phoneNumber;  
 }  
  
 public User getUser()  
 {  
 return user;  
 }  
 public void setUser(User user)  
 {  
 this.user = user;  
 }  
  
 @JsonIgnore  
 public String getLogin()  
 {  
 return user.getLogin();  
 }  
 @JsonIgnore  
 public String getRole()

Продолжение приложения Г

{  
 return user.getRole().getName();  
 }

@JsonIgnore  
 public String getEmail()

return user.getEmail();  
 }

package Shop.Models;  
  
public class User  
{  
 private int id;  
 private String login;  
 private String password;  
 private String email;  
 private Role role;  
  
 public User() { }  
   
 public User(int id, String login, String password, String email, Role role)  
 {  
 this.id = id;  
 this.login = login;  
 this.password = password;  
 this.email = email;  
 this.role = role;  
 }  
   
 public void setId(int id)  
 {  
 this.id = id;  
 }  
  
 public int getId()  
 {  
 return id;  
 }  
  
 public void setLogin(String login)  
 {  
 this.login = login;  
 }

Продолжение приложения Г

public String getLogin()  
 {  
 return login;  
 }

public void setPassword(String password)  
 {  
 this.password = password;  
 }  
 public String getPassword()  
 {  
 return password;  
 }  
   
 public void setEmail(String email)  
 {  
 this.email = email;  
 }  
  
 public String getEmail()  
 {  
 return email;  
 }  
  
 public void setRole(Role role)  
 {  
 this.role = role;  
 }  
 public Role getRole()  
 {  
 return role;  
 }  
  
}

**ПРИЛОЖЕНИЕ Д**

**(обязательное)**

**Листинг скрипта генерации базы данных**

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema shop

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema shop

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `shop` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci ;

USE `shop` ;

-- -----------------------------------------------------

-- Table `shop`.`categories`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `shop`.`categories` (

`CategoryId` INT(11) NOT NULL AUTO\_INCREMENT,

`CategoryName` VARCHAR(64) NOT NULL,

`CategoryDescription` VARCHAR(256) NULL DEFAULT NULL,

PRIMARY KEY (`CategoryId`))

ENGINE = InnoDB

Продолжение приложения Д

AUTO\_INCREMENT = 4

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `shop`.`vendors`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `shop`.`vendors` (

`VendorId` INT(11) NOT NULL AUTO\_INCREMENT,

`VendorName` VARCHAR(64) NOT NULL,

`Email` VARCHAR(64) NULL DEFAULT NULL,

`PhoneNumber` VARCHAR(20) NULL DEFAULT NULL,

`Country` VARCHAR(32) NULL DEFAULT NULL,

PRIMARY KEY (`VendorId`))

ENGINE = InnoDB

AUTO\_INCREMENT = 2

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `shop`.`products`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `shop`.`products` (

`ProductId` INT(11) NOT NULL AUTO\_INCREMENT,

`ProductName` VARCHAR(64) NOT NULL,

`ProductDescription` VARCHAR(256) NULL DEFAULT NULL,

`Price` DECIMAL(9,2) NOT NULL,

`VendorId` INT(11) NOT NULL,

PRIMARY KEY (`ProductId`),

INDEX `VendorId` (`VendorId` ASC),

CONSTRAINT `products\_ibfk\_1`

Продолжение приложения Д

FOREIGN KEY (`VendorId`)

REFERENCES `shop`.`vendors` (`VendorId`))

ENGINE = InnoDB

AUTO\_INCREMENT = 3

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `shop`.`category\_product`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `shop`.`category\_product` (

`CategoryId` INT(11) NOT NULL,

`ProductId` INT(11) NOT NULL,

PRIMARY KEY (`CategoryId`, `ProductId`),

INDEX `ProductId` (`ProductId` ASC),

CONSTRAINT `category\_product\_ibfk\_1`

FOREIGN KEY (`CategoryId`)

REFERENCES `shop`.`categories` (`CategoryId`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `category\_product\_ibfk\_2`

FOREIGN KEY (`ProductId`)

REFERENCES `shop`.`products` (`ProductId`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

Продолжение приложения Д

-- Table `shop`.`roles`

CREATE TABLE IF NOT EXISTS `shop`.`roles` (

`RoleId` INT(11) NOT NULL AUTO\_INCREMENT,

`Name` VARCHAR(64) NOT NULL,

PRIMARY KEY (`RoleId`))

ENGINE = InnoDB

AUTO\_INCREMENT = 5

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `shop`.`users`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `shop`.`users` (

`UserId` INT(11) NOT NULL AUTO\_INCREMENT,

`Login` VARCHAR(64) NOT NULL,

`Password` VARCHAR(256) NOT NULL,

`Email` VARCHAR(128) NOT NULL,

`RoleId` INT(11) NOT NULL DEFAULT '2',

PRIMARY KEY (`UserId`),

INDEX `fk\_users\_roles1\_idx` (`RoleId` ASC),

CONSTRAINT `fk\_users\_roles1`

FOREIGN KEY (`RoleId`)

REFERENCES `shop`.`roles` (`RoleId`))

ENGINE = InnoDB

AUTO\_INCREMENT = 9

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `shop`.`customers`

Продолжение приложения Д

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `shop`.`customers` (

`CustomerId` INT(11) NOT NULL AUTO\_INCREMENT,

`CustomerName` VARCHAR(64) NOT NULL,

`PhoneNumber` VARCHAR(20) NOT NULL,

`UserId` INT(11) NOT NULL,

PRIMARY KEY (`CustomerId`),

INDEX `fk\_customers\_users1\_idx` (`UserId` ASC) VISIBLE,

CONSTRAINT `fk\_customers\_users1`

FOREIGN KEY (`UserId`)

REFERENCES `shop`.`users` (`UserId`))

ENGINE = InnoDB

AUTO\_INCREMENT = 8

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `shop`.`orders`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `shop`.`orders` (

`OrderId` INT(11) NOT NULL AUTO\_INCREMENT,

`OrderStatus` VARCHAR(32) NOT NULL DEFAULT 'Pending',

`TotalPrice` DECIMAL(9,2) NOT NULL,

`Date` DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`CustomerId` INT(11) NOT NULL,

PRIMARY KEY (`OrderId`),

INDEX `CustomerId` (`CustomerId` ASC) VISIBLE,

CONSTRAINT `orders\_ibfk\_1`

FOREIGN KEY (`CustomerId`)

REFERENCES `shop`.`customers` (`CustomerId`))

Продолжение приложения Д

ENGINE = InnoDB

AUTO\_INCREMENT = 1

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `shop`.`order\_product`

CREATE TABLE IF NOT EXISTS `shop`.`order\_product` (

`OrderId` INT(11) NOT NULL,

`ProductId` INT(11) NOT NULL,

`Amount` INT(11) NOT NULL,

PRIMARY KEY (`OrderId`, `ProductId`),

INDEX `ProductId` (`ProductId` ASC) VISIBLE,

CONSTRAINT `order\_product\_ibfk\_1`

FOREIGN KEY (`OrderId`)

REFERENCES `shop`.`orders` (`OrderId`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `order\_product\_ibfk\_2`

FOREIGN KEY (`ProductId`)

REFERENCES `shop`.`products` (`ProductId`)

ON DELETE CASCADE

ON UPDATE CASCADE)

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

INSERT INTO Roles VALUES (1, "admin");

INSERT INTO Roles VALUES (2, "user");

INSERT INTO Roles VALUES (3, "blocked");

**ПРИЛОЖЕНИЕ Е**

**(обязательное)**

Проверка на антиплагиат:

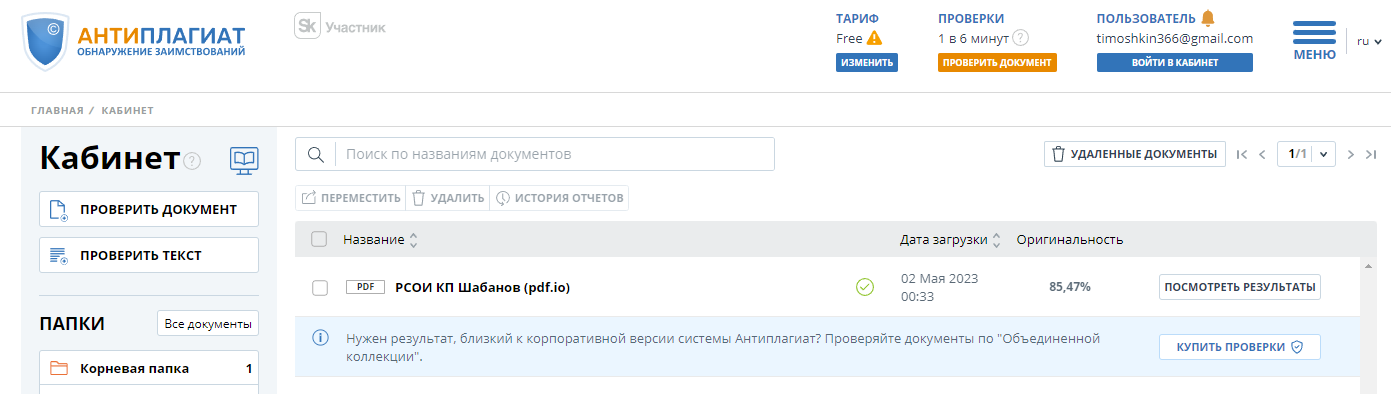


Рисунок Е.1 – Проверка на антиплагиат