**САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ**

**БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

МДК 01.01. Разработка программных модулей

Специальность 09.02.07

«Информационные системы и программирование»

Квалификация «Программист»

Руководитель учебной практики:

Матысик И.А.

Выполнил студент группы 493:

Баланин Э.Н.

Санкт-Петербург, 2023

ОГЛАВЛЕНИЕ

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ 3](#_Toc129201047)

[ВВЕДЕНИЕ 4](#_Toc129201048)

[1. ОБЩАЯ ЧАСТЬ 6](#_Toc129201049)

[1.1 Цель разработки 6](#_Toc129201050)

[1.2 Описание предметной области 7](#_Toc129201051)

[1.3 Анализ средств и технологий 8](#_Toc129201052)

[2. ПРОЕКТНАЯ ЧАСТЬ 10](#_Toc129201053)

[2.1. Постановка задачи 10](#_Toc129201054)

[2.2. Функционально-логическая структура проекта 12](#_Toc129201055)

[2.2.1. Диаграмма вариантов использования 12](#_Toc129201056)

[2.2.2. Диаграмма деятельности 12](#_Toc129201057)

[2.3. Логическая модель данных 14](#_Toc129201058)

[2.4. Разработка программных модулей 17](#_Toc129201059)

[2.4.1. Пользовательский интерфейс 17](#_Toc129201060)

[2.5. Руководство пользователя 18](#_Toc129201061)

[ПРИЛОЖЕНИЕ 1. Исходный программный код модуля «Авторизация» 29](#_Toc129201062)

[ПРИЛОЖЕНИЕ 2. Исходный программный код модуля «Список заказов» 31](#_Toc129201063)

# **ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ**

Требуется разработать информационную систему (ИС) «Компьютерный Мир» – ПО разрабатываемое как для клиентов (с целью демонстрирования ассортимента компании, помощи в выборе товара и оформлении заказа), так и для автоматизации работы компании (добавление новых товаров, проведение заказов). Для этого требуется разработать основные программные модули ИС для следующих пользователей:

* + неавторизованный клиент и авторизованный клиент может просматривать товары и формировать заказ;
  + менеджер может просматривать товары, формировать и редактировать заказы;
  + системный администратор может добавлять/редактировать/удалять товары, просматривать и редактировать заказы.

Выполнить анализ и проектирование бизнес-процессов разрабатываемой информационной системы путем составления диаграммы прецедентов (Use–Case) для пользователей системы.

Выполнить проектирование реляционного хранилища данных путем составления ER-диаграммы разрабатываемой информационной системы и ведение словаря данных.

Выполнить разработку базы данных информационной системы «Компьютерный мир» на сервере баз данных, создать таблицы основных сущностей, атрибуты, отношения и необходимые ограничения. Выполнить импорт подготовленных данных в разработанную базу данных информационной системы.

Выполнить разработку desktop-приложения для автоматизации работы сотрудников компании предоставляющих продажу компьютерных комплектующих. Система должна функционировать в многопользовательском режиме, поэтому каждый пользователь должен иметь свой пароль доступа в систему, т.е. приложение должно обеспечивать процесс авторизации пользователя в системе.

Клиент может просматривать информацию о существующих товарах. Так же клиенту доступна возможность формирования заказа.

Менеджер обеспечивает создание заказа и также обладает возможностью просмотра номенклатуры компании.

Системный администратор обеспечивает безопасность системы от внешнего доступа к данным о сотрудниках и клиентах компании. В его функции входит контроль за входом в систему (ведение истории входов), выдача уникальных учетных данных для авторизации в системе.

В целом разрабатываемая система направлена на автоматизацию бизнес-процессов, протекающих в компании занимающейся продажей компьютерных комплектующих.

# **ВВЕДЕНИЕ**

Учебная практика по МДК 01.01 «Разработка программных модулей» является обязательным учебным элементом для формирования общих и профессиональных компетенцией по модулю ПМ 01 «Разработка модулей программного обеспечения для компьютерных систем».

Практика проходит в колледже информационных технологий, в компьютерной лаборатории №213 «Прикладное программирование». Период прохождения практики с 23.02.2023 по 15.03.2023. Объем учебной практики 30 часов. Тема учебной практики: «Разработка информационной системы магазина по продаже компьютерных комплектующих».

В отчете к учебной практике представлена краткая необходимая теоретическая справка, представлена четкая постановка задачи, представлены диаграммы, описывающие функциональную структуру программы и логическую модель данных, подробная инструкция пользователя и выводы по проделанной работе. Представлен программный код в приложении. В результате прохождения учебной практики по МДК 01.01 были подтверждены общие и профессиональные компетенции.

Перечень общих компетенций:

1. ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
2. ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
3. ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.
4. ОК 4. Планировать и реализовывать собственное профессиональное и личностное развитие.
5. ОК 5. Планировать и реализовывать собственное профессиональное и личностное развитие.
6. ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения.
7. ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.
8. ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.
9. ОК 9. Использовать информационные технологии в профессиональной деятельности.
10. ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.

Перечень профессиональных компетенций:

1. ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием;
2. ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

В результате прохождения учебной практики по МДК 01.01 «Разработка программных модулей», был получен практический опыт в следующих пунктах:

1. В разработке кода программного продукта на основе готовой спецификации на уровне модуля;
2. В использовании инструментальных средств на этапе отладки программного продукта.

В результате прохождения учебной практики, были получены умения:

1. Осуществлять разработку кода программного модуля на языке высокого уровня;
2. Создавать программу по разработанному алгоритму как отдельный модуль;
3. Осуществлять разработку кода программного модуля на современных языках программирования;
4. Выполнять оптимизацию и рефакторинг программного кода;
5. Оформлять документацию на программные средства;
6. Разрабатывать код программного модуля с применением технологии объектно-ориентированного программирования;
7. Использовать современные технологии обработки данных.
8. В результате прохождения учебной практики по модулю, были получены знания по следующим понятиям:
9. Основные этапы разработки программного обеспечения;
10. Основные принципы технологии структурного и объектно-ориентированного программирования;
11. Основные возможности современных технологий.

# **ОБЩАЯ ЧАСТЬ**

В данной части отчета по учебной практике представлена краткая теория по теме учебной практике, а также представлены результаты исследования проведенного анализа программных средств и технологий для разработки проекта учебной практики.

## **Цель разработки**

Целью проекта является проектирование и разработка информационной системы для клиентов и специалистов компании по продаже компьютерных комплектующих, которая поможет автоматизировать добавления новых товаров, п, что сократит время обслуживания клиента компании. В итоге разрабатываемая система должна привлечь больше потенциальных клиентов, обеспечив стабильный рост доходов и рейтингов компании, для которого разрабатывается данная информационная система.

## **Описание предметной области**

Информационная система "Компьютерный мир" – прикладное программное обеспечение, использующееся для предоставления клиентам сведений об товаре выставляемом на продажу компанией, а также ряда следующих функциональных особенностей:

1. Добавление нового товара в базу данных и предоставление сведений о нем клиентам;
2. Редактирование или удаление уже существующего товара, оформление заказа для клиента.

Основными целями внедрения системы являются:

1. Упрощение предоставления информации об ассортименте компании;
2. Автоматизация процесса оформления заказа;
3. Минимизация человеческого фактора при ведении документов;
4. Снижение рутиной работы при работе с документами.

Рассмотрим определение прецедентов (вариантов использования). Система требуется, прежде всего, следующим заинтересованным лицам:

1. Менеджер компании;
2. Клиент;
3. Системный администратор.

Функционал каждой роли:

Системный администратор и менеджер обладают возможностью просмотра, редактирования, добавления, удаления данных. Ко всему прочему системный администратор имеет возможность создавать пользователей для системы, удалять их или же изменять их учётные данные.

Клиент может только просматривать сведения о товаре и формировать заказы.

Система должна функционировать в многопользовательском режиме, поэтому каждый пользователь должен иметь свой пароль доступа в систему.

Проект предназначен для коммерческого компании, занимающегося продажей компьютерных комплектующих физическим и юридическим лицам.

Внедрение данного проекта поможет автоматизировать процесс добавления новых клиентов, способствуя снижению рутиной работы с ведением документооборота и минимизации случаев человеческого фактора.

## **Анализ средств и технологий**

В данном разделе описываются возможные пути и программные средства, с помощью которых можно реализовать приложение.

В данном подразделе пояснительной записки представлен материал проведенного анализа программных средств, которыми можно разработать приложение. В анализе присутствуют сравнительные характеристики программных технологий и методов, языков программирования и интегрированных сред.

Инструментальные средства:

1. Операционная система Windows 10;
2. Язык программирования C#;
3. Среда программирования Visual Studio 2022;
4. СУБД SQL Server Management Studio.

Для создания приложения будет использована интегрированная среда программирования Microsoft Visual Studio, язык программирования высокого уровня С#, систему построения клиентских приложений Windows Presentation Foundation (WPF), а также SQL Server Management для создания базы данных.

В качестве среды разработки была выбрана инструментальная среда разработки Microsoft Visual Sutdio, включающая в себя интегрированную среду разработки, редактор исходного кода, встроенный отладчик. Многие другие инструменты возможно получить благодаря подключению плагинов – сторонних расширений. Был выбран по причине поддержки языка программирования С#. Также это наиболее изученная разработчиком платформа, что позволяет быстрее разрабатывать приложения не тратя время на изучения нового материала.

В качестве языка разработки был выбран язык программирования C# – объектно-ориентированный язык программирования. Разработан в 1998 — 2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework.

Достоинства С# по книге Биллига:

1. Создавался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности - как FCL, так и CLR;
2. Является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;
3. Является мощным объектным языком с возможностями наследования и универсализации;
4. Является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход от С++ к C#;
5. Сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства С++ как указатели, адресация, разыменование, адресная арифметика;
6. Благодаря каркасу.Net, ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java;
7. Мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;
8. Реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#.

Выделение и объединение лучших идей современных языков программирования делает язык C# не просто суммой их достоинств, а языком программирования нового поколения.

Windows Presentation Foundation (WPF) — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

SQL Server Management Studio – утилита из Microsoft SQL Server 2019 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

# **ПРОЕКТНАЯ ЧАСТЬ**

В данной части отчета по учебной практике представлено подробное описание разработанного приложения, соответствующего поставленной задаче в техническом задании к учебной практике.

## **Постановка задачи**

Информационная система «Компьютерный мир» – прикладное программное обеспечение, использующееся для предоставления информации о товарах предоставляемых на продажу компанией, упрощения формирования заказа пользователями, а также предназначена для решения следующих задач:

1. Контроль заказов;
2. Формирование необходимой информации по итогам оформленных заказов на физическое или юридическое лицо (дата оформления заказа, количество товаров в заказе, необходимая для выплаты сумма).

Основными целями внедрения системы являются:

1. Упрощение предоставления информации об ассортименте компании;
2. Автоматизация процесса оформления заказа;
3. Минимизация человеческого фактора при ведении документов.

Исходная информация в систему поступает из документов о номенклатуре компании. Входными данными являются следующие:

1. Название товара;
2. Изображение товара;
3. Системные характеристики товара.

Рассмотрим определение прецедентов (вариантов использования). Система требуется прежде всего следующим заинтересованным лицам:

1. Менеджер;
2. Клиент;
3. Системный администратор.

Определяется перечень функциональных подсистем, их назначение и основные характеристики, требования к числу уровней иерархии и степени централизации системы.

Смежными системами для «Клиенты» являются:

1. информационные системы ввода, хранения, поиска и оперативной обработки информации о клиентах;
2. информационная система формирование отчетов, необходимых менеджеру, содержащих следующие данные о клиенте: ФИО, номер телефона.

Источниками данных для системы должны быть:

1. СУБД MS SQL;
2. таблицы Excel.

Входными данными являются следующий набор данных:

1. ФИО клиента;
2. Номер телефона клиента;
3. Логин и пароль.

Система должна функционировать в многопользовательском режиме и давать возможность:

1. Просматривать записи базы данных, в том числе и с помощью различных фильтров;
2. Добавлять новые записи и изменять существующие;
3. Удалять записи.

## **Функционально-логическая структура проекта**

В данном разделе отчета по учебной практике представлены различные диаграммы на языке UML, описывающие функциональную и логическую структуру разрабатываемой программы учебной практики.

### **Диаграмма вариантов использования**

В этом разделе приведена UML-диаграмма типа вариантов использования или диаграмма прецедентов, представленная на рисунке 1. Диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Прецедент - возможность моделируемой системы, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат.

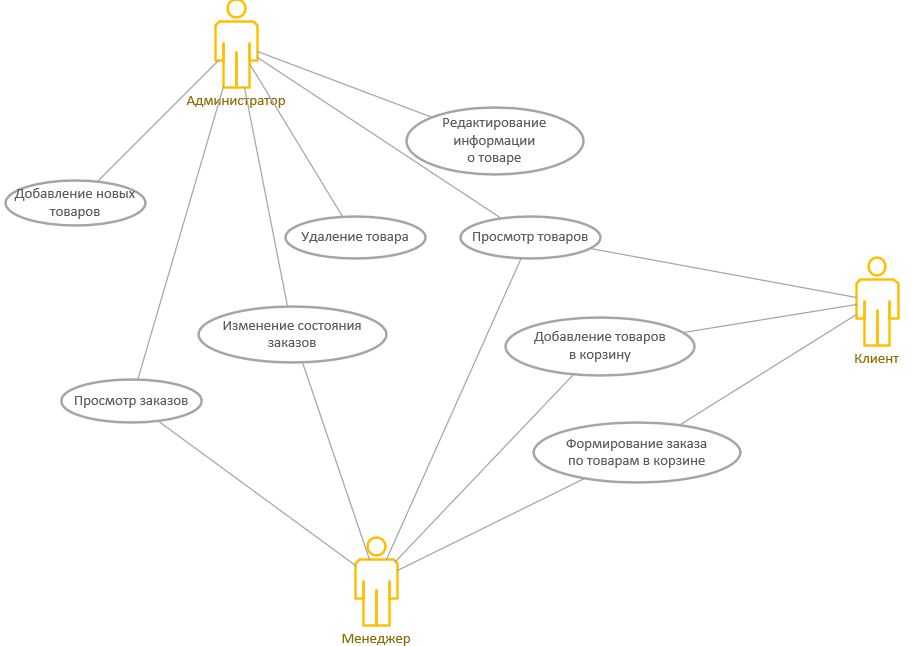
****

Рисунок 1 – Диаграмма вариантов использования

### **Диаграмма деятельности**

Диаграмма деятельности (англ. activity diagram) – UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов – вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

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

Была разработана диаграмма деятельности для прецедента «Добавление нового товара», которая осуществляется ролью «системный администратор». Данная диаграмма представлена на рисунке 2.

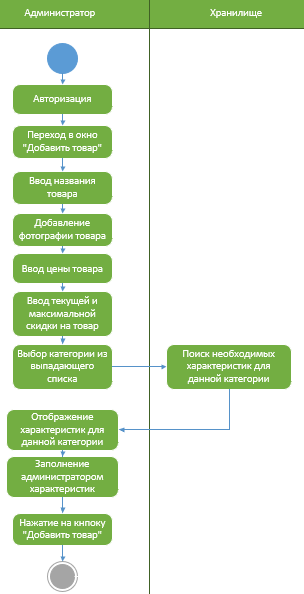


Рисунок 2 – Диаграмма деятельности

## **Логическая модель данных**

В основе разрабатываемого приложения находится база данных, как структурированное хранилище информации разрабатываемой системы. Удобно логическую модель данных описывать с применением UML-языка на основе ED-диаграммой. На рисунке 3 представлена ER-диаграмма разработанной базы данных.

В системе есть две сущности: «Пользователи» и «Товары» и «Характеристики».

Таблица «User» состоит из полей Фамилия, Имя, Отчество, email-адрес, логин, пароль и ID роли (связанная таблица, описана ниже).

Таблица «Role» состоит из полей идентификатора записи и названия роли.

Таблица «Product» состоит из полей идентификатора, названия товара, идентификатора категории данного товара (внешний ключ, связывающий данную таблицу с таблицей «Category», описана ниже), цены товара, текущей и максимальной скидки на товар.

Таблица «Category» состоит из следующих полей: поле-идентификатор записи, название категории в единственном числе.

Таблица «ProductPicture» содержит в себе поле-идентификатор записи, идентификатор продукта (при помощи этого поля таблица «ProductPicture» связана с таблицей «Product») и поле двоичного представления изображения данного товара.

Таблица «Characteristic» содержит в себе поле идентификатор записи и название характеристики (у разных товаров разные характеристики, при помощи данной таблицы получается гибко указывать к товарам характеристики).

Таблица «CategoryCharacteristic» обладает полем-идентификатором категории и полем идентификатором характеристики товара. Необходимость в данной таблице возникает во время добавления нового товара, нужно вывести определенный список характеристик для их дальнейшего заполнения администратором.

Таблица «ProductCharacteristic» содержит в себе поле-идентификатор записи, поле-идентификатор товара к которому относится данная запись, поле-идентификатор характеристики, значение этой самой характеристики и поле для записи единиц измерения.

Таблица «Outpost» состоит из поля-идентификатора и поля содержащего в себе адрес пункта выдачи товара.

Таблица «OrderStatus» содержит в себе поле-идентификатор записи и поле содержащее в себе возможное состояние заказа (новый, завершен и другие возможные состояния).

Таблица «Order» содержит в себе информацию о заказе: дата создания заказа, приблизительная дата получения заказа, поле содержащее идентификатор пункта выдачи (связь с таблицей «Outpost»), поле содержащее в себе идентификатор состояния заказа (связь с таблицей «OrderStatus»), код для получения заказа в пункте выдачи, ФИО клиента оформившего заказ (поле может быть пустым в случае если заказ сформировал гость).

Таблица «OrderContent» состоит из поля-идентификатора продукта, поля-идентификатора заказа и поля отвечающего за кол-во заказанного товара конкретной позиции.

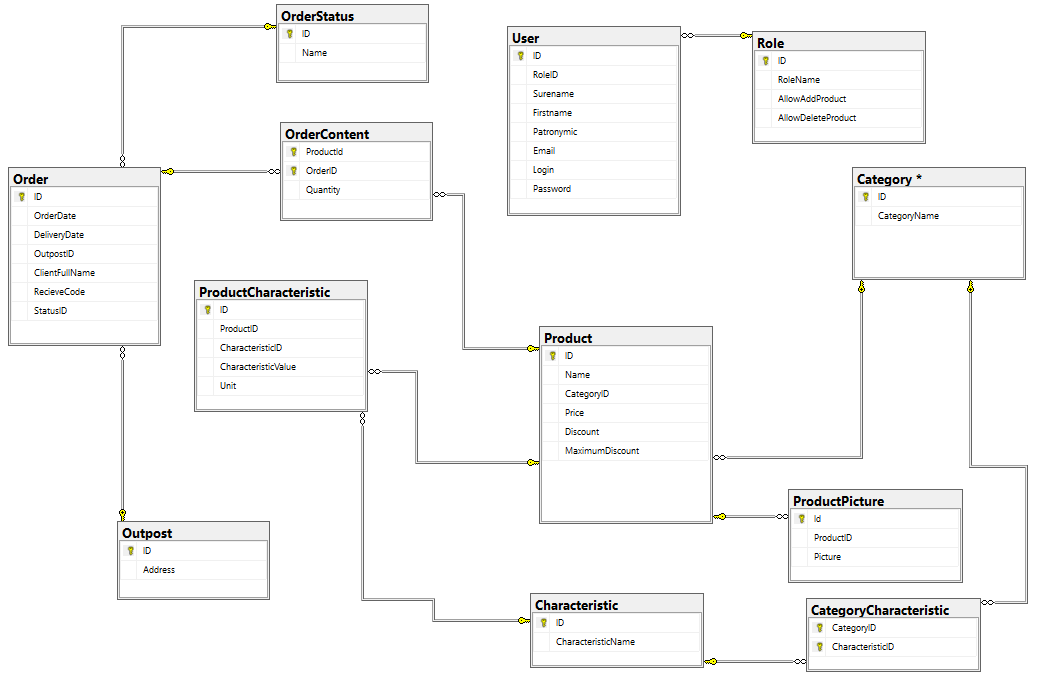


Рисунок 3 – ER-диаграмма базы данных

На рисунке 4 представлен заполненный словарь данных системы – набор информации, описывающий, какой тип данных хранится в БД, их формат, структуру и способы использования данных. Дополнительно в словаре данных могут содержаться описания к отдельным полям таблицы базы данных.



Рисунок 4 – Словарь данных системы

На рисунке 5 изображен словарь данных для таблиц «ProductCharacteristic», «Outpost», «OrderStatus», «Order», «OrderContent».

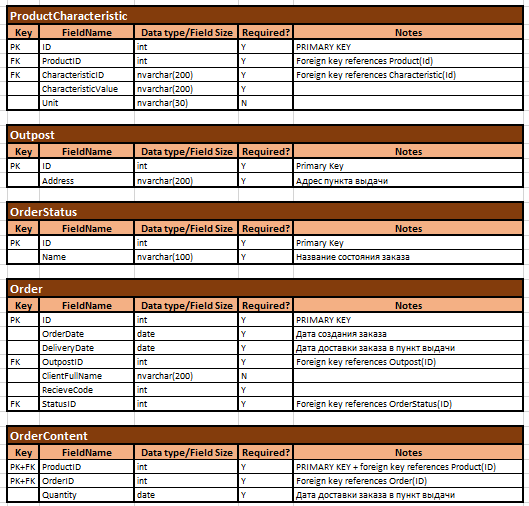


Рисунок 5 – Словарь данных для оставшихся таблиц

## **Разработка программных модулей**

В данном разделе отчета представлены действия, необходимые для программной реализации разрабатываемого приложения, а так же требования к разработке пользовательского интерфейса.

### **Пользовательский интерфейс**

Не допускаются орфографические и грамматические ошибки.

Все экранные формы пользовательского интерфейса должны иметь заголовок с логотипом logo.png (расположен в ресурсах). Нельзя искажать логотип (не изменяйте изображение, его пропорции, цвет).

Также для приложений должна быть установлена иконка.

В случае возникновения ошибок система оповещает пользователя всплывающими окнами с информацией об ошибке.

## **Руководство пользователя**

Для начала работы необходимо запустить exe файл – PcComponents.exe

**Форма ввода данных для входа**

После запуска программы открывается окно аутентификации (рисунок 6), где нужно ввести логин и пароль для входа в систему, на рисунке 6 продемонстрировано данное окно.

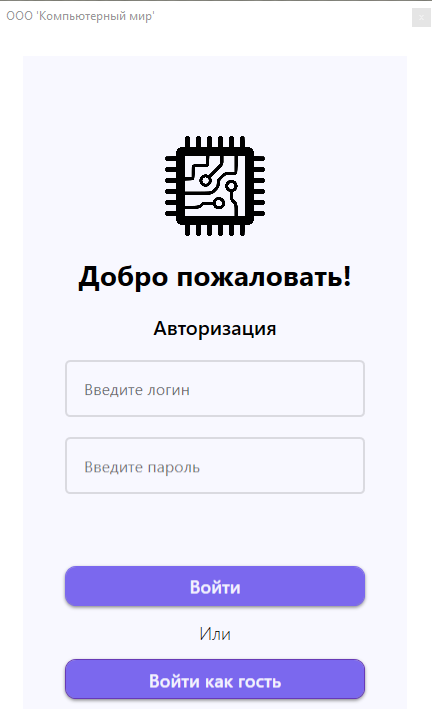


Рисунок 6 – Форма для ввода аутентификационных данных

При ошибочном вводе пароля или логина вам будет показано сообщение о том, что вы должны проверить правильность ввода, как показано на рисунке 7. Сообщение было показано, отслеживание ошибок ввода пароля работает.

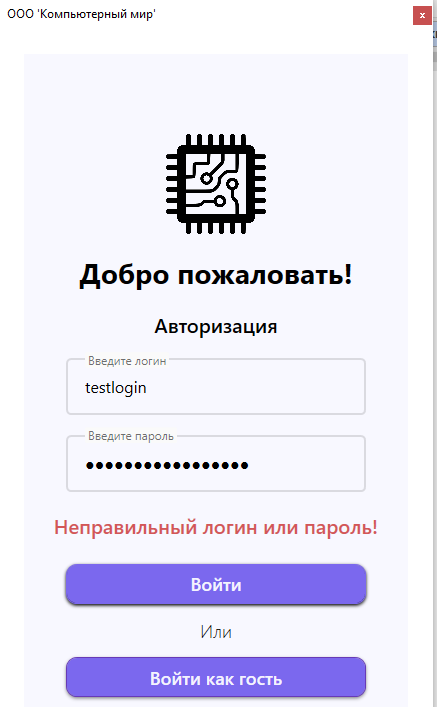


Рисунок 7 – Результат неправильного ввода пароля

В случае корректного ввода данных, программа сразу же откроет окно каталога для пользователя, под которым вы зашли. В данном случае это администратор, как показано на рисунке 8.

На рисунке 8 продемонстрирован результат успешной авторизации под учетную запись администратора

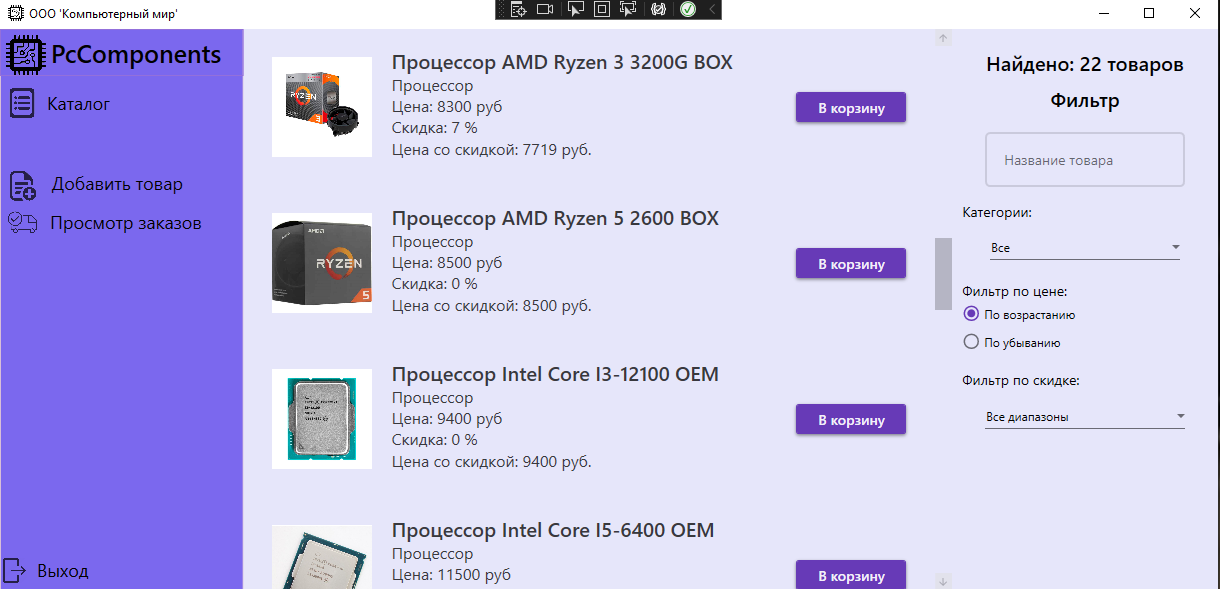


Рисунок 8 – Успешная авторизация

**Каталог товаров**

Форма каталога товаров доступна каждому пользователю, но все роли кроме администратора могут формировать заказа по нажатию кнопки «В корзину». У администратора кнопка не функционирует.

В левой части экрана расположено навигационное меню при помощи которого можно перемещается по приложению. Здесь будут доступные следующее возможности: открытие каталога, открытие корзины, открытие окна добавление нового товара, просмотр существующих а также выход обратно к окну авторизации. Все эти переходы разграничены для каждой роли пользователя в соответствии обозначенными вариантами использования в диаграмме вариантов использования (рисунок 1) (кнопка для перехода в корзину отсутствует, т.к. авторизация была выполнена на роль системного администратора, у которого нет возможности формирования заказов).

Каталог товаров можно отсортировать по цене товара: по возрастанию или убыванию. Также доступна сортировка товара по диапазонам скидки: все диапазоны, от 0 до 3%, от 3 до 6%, 7% и более. Также доступен поиск товара по его имени и категории. В верхней правой части экрана отображается итоговое кол-во найденных товаров по всех описанным выше категориям. Если в поле для поиска товара по названию введена пустая строка или ничего не введено, то поиск по имени товара не осуществляется. По нажатию на кнопку «В корзину» товар добавится в корзину (только у гостя, авторизированного клиента и менеджера). В левой части экрана около слова «Корзина» появится число означающее кол-во разных товаров помещенных в корзину.

На рисунке 9 продемонстрирован поиск товаров по категории «Процессор», содержащие в названии «Intel» отсортированные по убыванию цены.

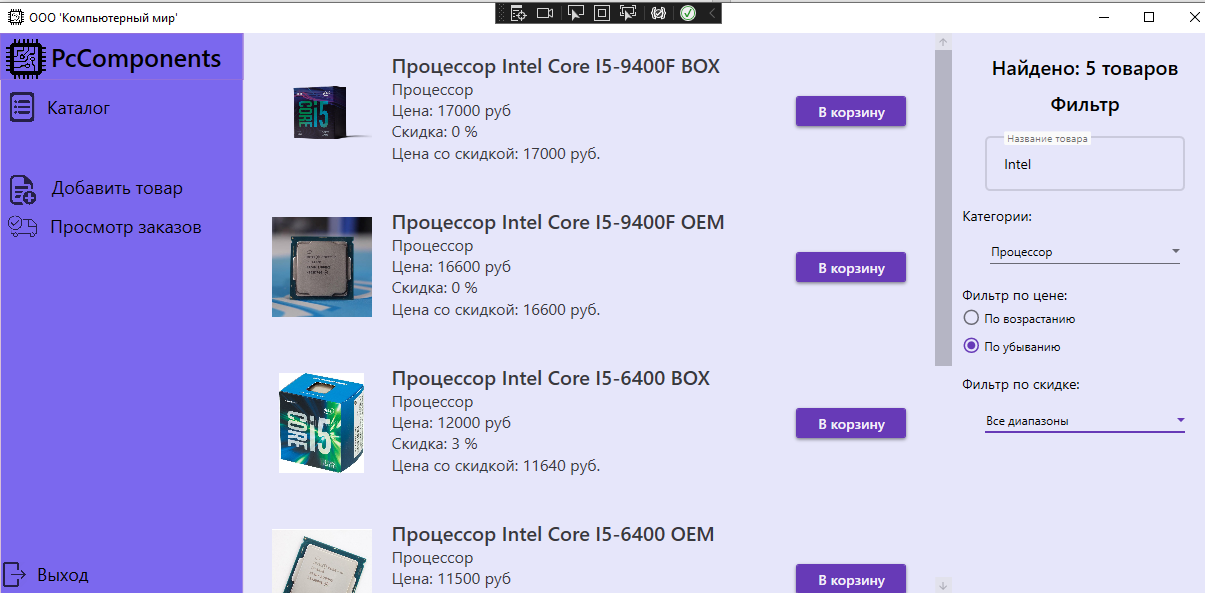


Рисунок 9 – Форма системного администратора

**Окно корзины**

Для добавления товара в корзину необходимо выбрать в списке товаров интересующий вас товар и нажать на кнопку «В корзину». Далее в левом меню около слова «Каталог» появится цифра символизирующая кол-во различных товаров добавленных в вашу корзину. По нажатию на эту кнопку «Каталог» вы перейдете в окно каталога, где будут отображены все выбранные вами товары, ваше ФИО или слово гость в случае авторизации в качестве гостя, общая цена за все товары с учетом кол-ва каждого вида товара.

Если вам необходимо несколько товаров одного типа, то увеличить кол-во выбранного в корзине товара можно будет при помощи кнопок «->» или уменьшить при помощи кнопки «<-». Также между двух ранее указанных кнопок находится поле для ввода кол-ва товара. В случае если в этом поле число товара становится 0, то товар исчезает из корзины. Общая цена обновляется при любом изменении корзины, добавление нового товара, удалении товара или изменении кол-ва товаров одного вида. На рисунке 10 изображено меню корзины пользователя роли «Клиент».

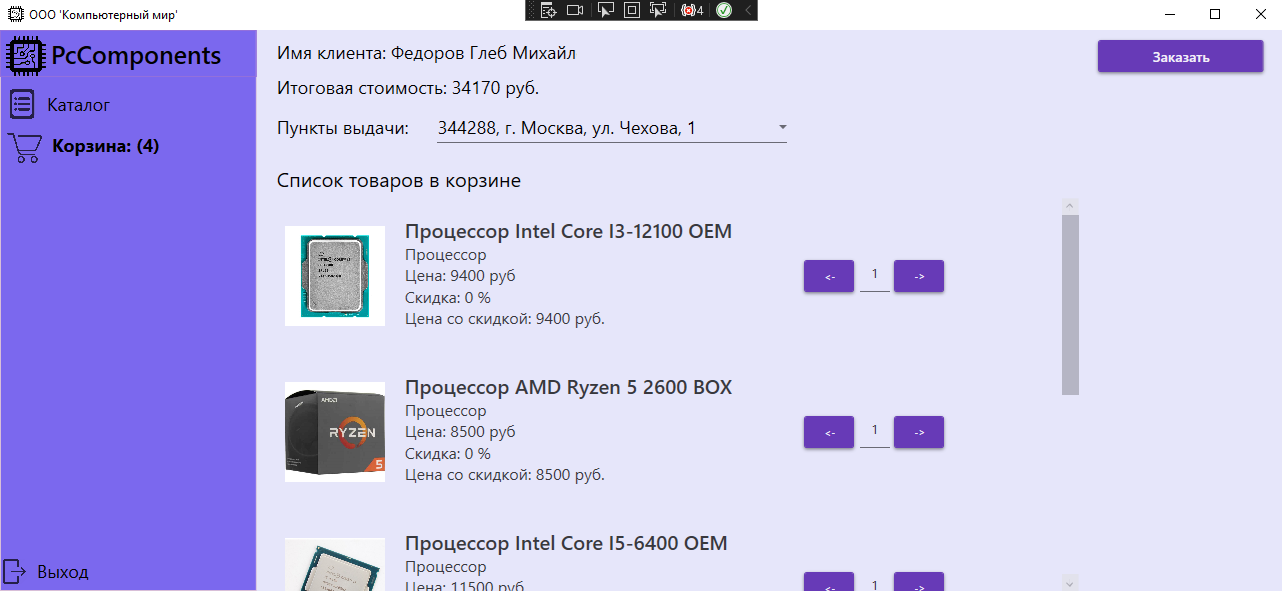


Рисунок 10 – Окно корзины

При нажатии на кнопку «Заказать» в верхнем правом углу приложения, сформируется заказ в выбранный пункт выдачи и на экране появится сообщение показанное на рисунке 11.

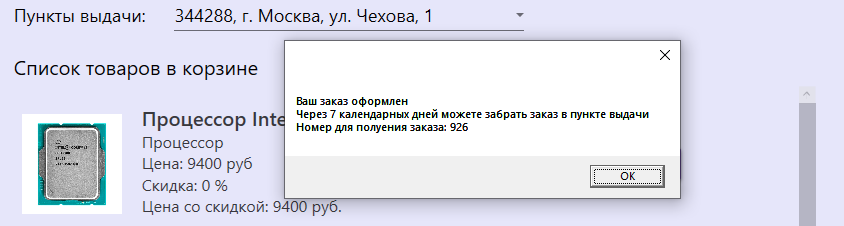


Рисунок 1 – Оформление заказа

Номер для получения заказа уникальный для каждого заказа, его необходимо запомнить, при помощи этого номер придя в пункт выдачи вы заберете свой заказ (также можно идентифицировать заказ по своему ФИО в случае оформления заказа под ролью «Клиент»).

**Просмотр полной информации о товаре**

Находясь в каталоге можно посмотреть подробную информацию о товаре дважды нажав на него левой кнопкой мыши. На рисунке 12 изображено окно просмотра полной информации о товаре.



Рисунок 12 – Полная информация о товаре

**Просмотр заказов**

Для пользователей с ролями «Менеджер» и «Администратор» доступен просмотр оформленных заказов. В данном окне доступна для просмотра большая часть информации о заказе: Номер заказ, ФИО клиента оформившего заказ (или пустая строка в случае оформления заказа гостем), адрес пункта выдачи, дата формирования заказа и дата доставки заказа в пункт выдачи. Также доступна возможность для поиска заказа по его номеру в правом верхнем углу окна просмотра заказов. При нажатии на определенный заказ будет выполнен переход на форму содержащую подробную информацию о данном заказе (в том числе товары данного заказа).

На рисунке 13 продемонстрировано окно для просмотра заказов. Отображены только товары которые удовлетворяют значению в поле поиска товара по его номеру (в поле записано значение 1). Для завершенных заказов кнопка «Изменить» недоступна, т.к. подразумевается что заказ уже получен клиентом.

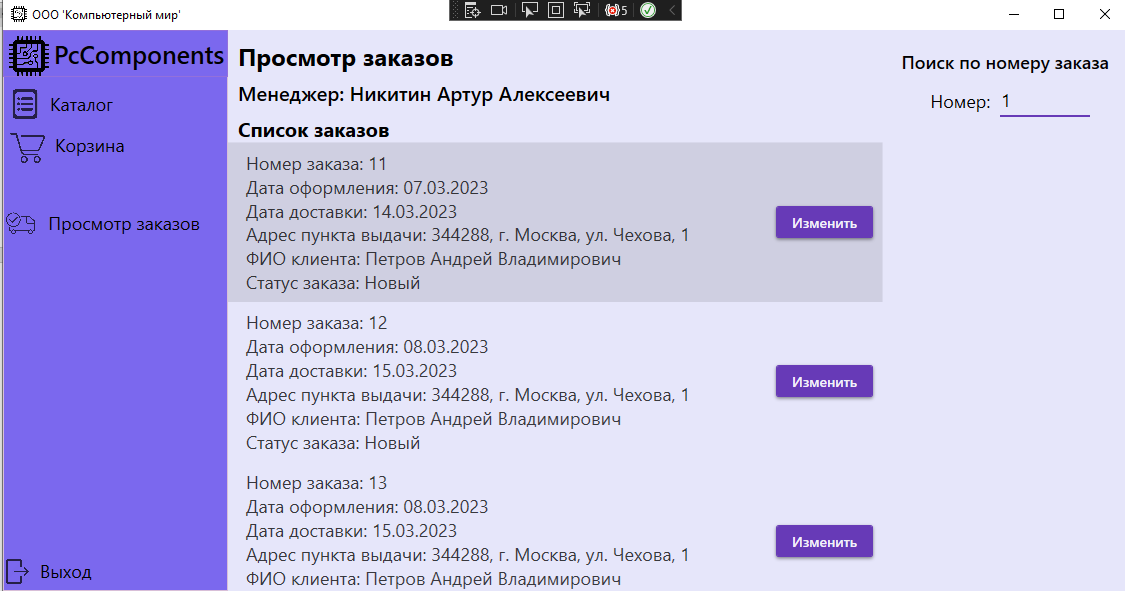


Рисунок 13 – Окно просмотра заказов

При двойном нажатии на конкретный заказ произойдет переход на окно с подробным его описанием. На рисунке 14 продемонстрировано данное окно. При двойном нажатии на товар будет открыто окно с его подробным описанием. Данное окно идентично окну, продемонстрированному на рисунке 12.

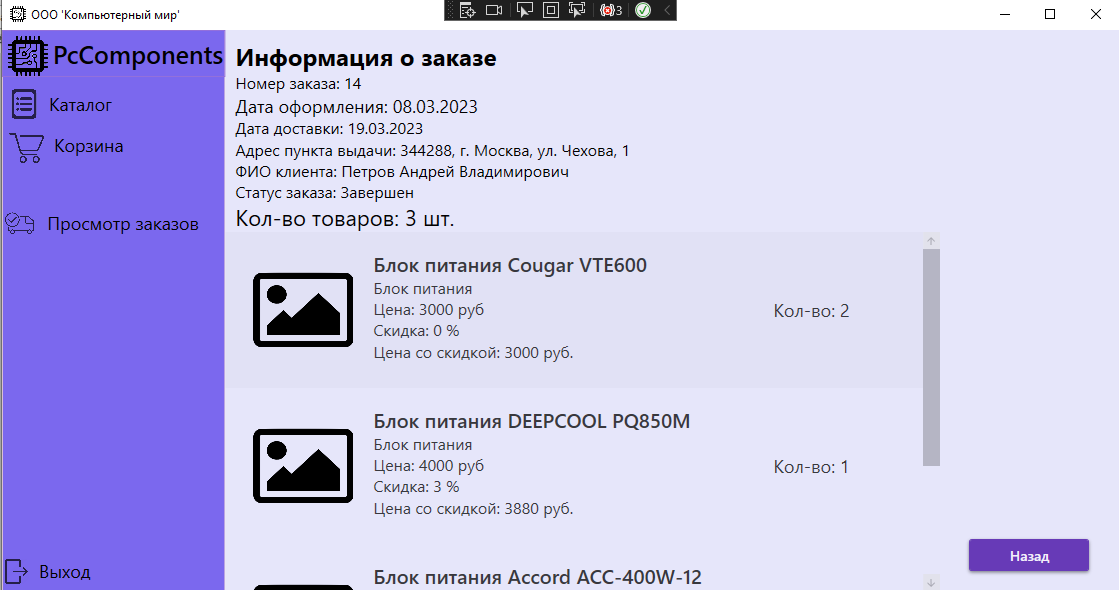


Рисунок 14 – Подробное описание заказа

При нажатии на кнопку «Изменить» открывается окно подробной информации о заказе, где становятся доступные для редактирования поля даты формирования заказ, дата доставки заказа, ФИО клиента и состояние заказа. Все изменения по нажатию на кнопку «Сохранить» применятся и отразятся в базе данных. На рисунке 15 изображено данное окно редактирования заказа.

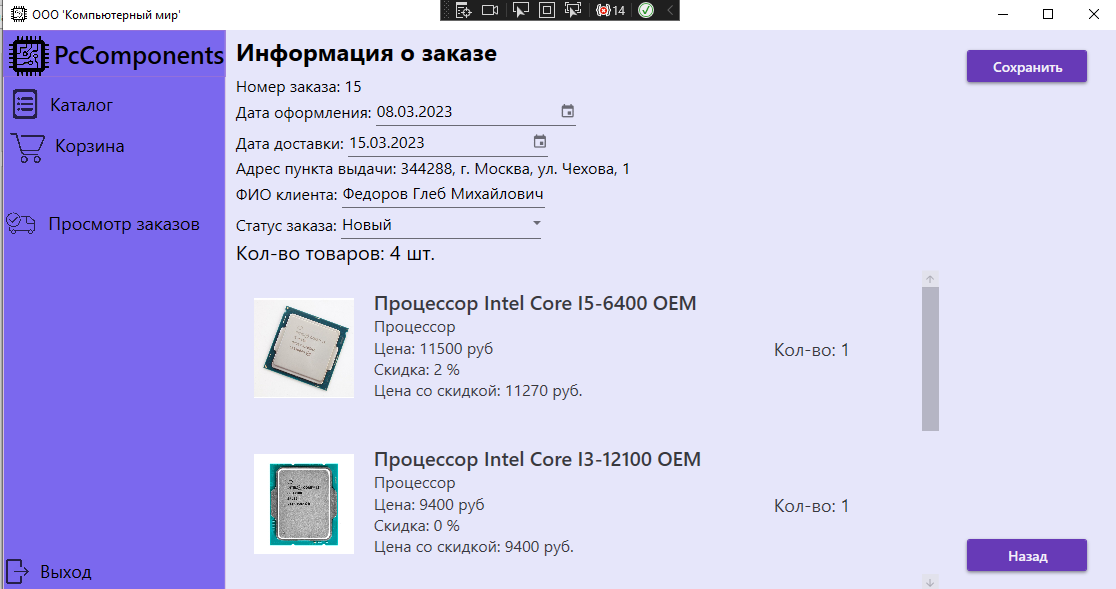


Рисунок 15 – Окно редактирования заказа

**Добавление нового товара и редактирование информации о товаре**

Администратору доступна функция добавления нового товара. По нажатию на кнопку «Добавить товар» в левом навигационном меню приложения произойдет переход на новое окно добавления нового товара. Становятся доступны поля «Название товара», выпадающий список для выбора категории нового товара, поле для указания цены нового товара, текущей скидки и поле для максимальной скидки товара. Также есть возможность добавления картинки к товару по нажатию на кнопку «Изменить картинку». По умолчанию картинки у нового товара нет, выводится заглушка для данного товара которая будет отображаться в случае если картинка не будет добавлена. Нажатие на кнопку «Изменить картинку» открывает диалоговое окно для выбора картинки с компьютера на котором находится приложение. При выборе картинки она автоматически отобразится на соответствующем компоненте отображения и запишется к новому пока еще не созданному товару.

При выборе категории товара автоматически загрузятся общие характеристики для данной категории товаров. Не обязательно заполнять все характеристики, в случае если некоторые из них были пропущены они не запишутся в базу данных и товар не будет владеть информацией о них.

На рисунке 16 изображено окно для создания нового товара категории «Процессоры» где уже заполнены некоторые характеристики данного товара.

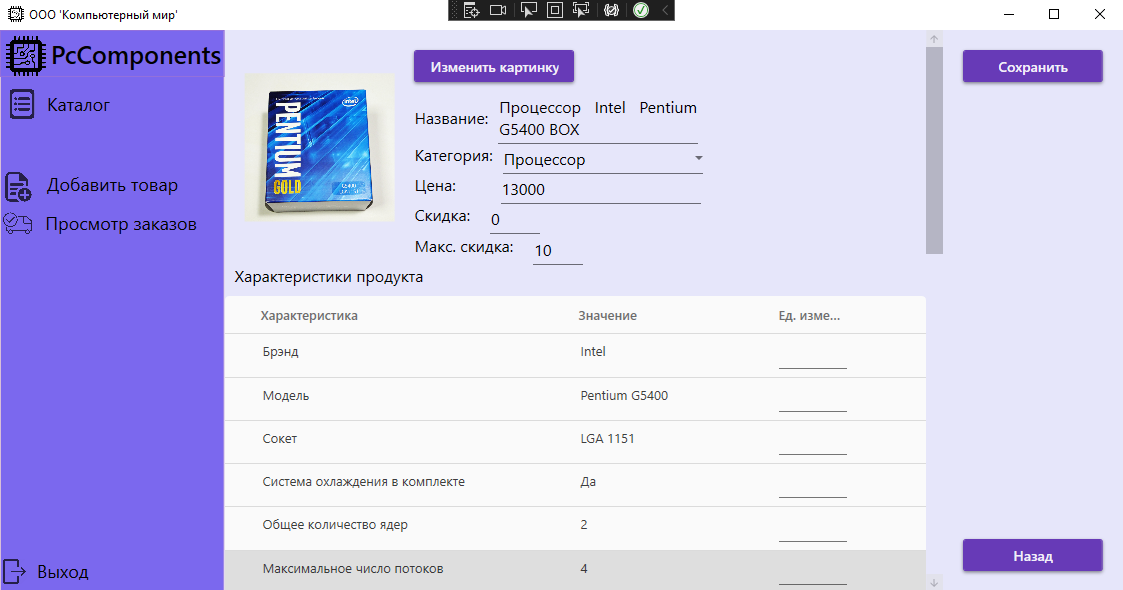


Рисунок 16 – Добавление нового товара

При нажатии на кнопку «Сохранить» отобразится сообщение «Товар успешно добавлен» и данные о новом товаре занесутся в БД и он станет доступен для отображения в каталоге.

Также для администратора доступна возможность редактирования информации о существующем товаре. Чтобы попасть в соответствующее окно администратору необходимо найти интересующий его товар в каталоге и дважды нажать на него. После этого, если авторизированный пользователь обладает ролью «Администратор», то откроется окно редактирования информации о данном товаре, где будут доступны для редактирования все поля.

На рисунке 17 продемонстрировано окно редактирования информации о товаре.

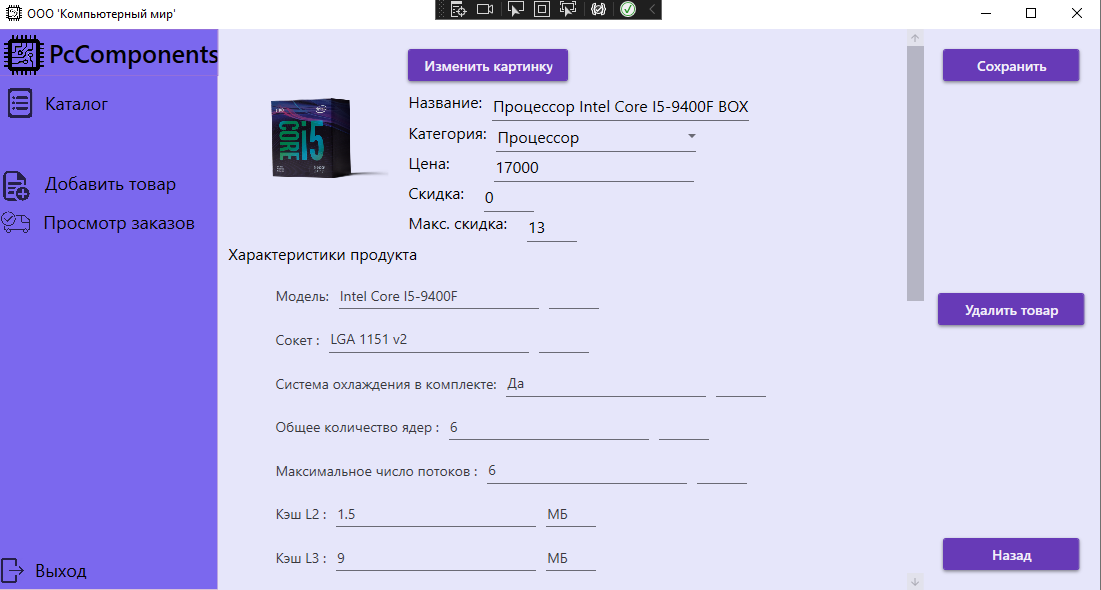


Рисунок 17 – Редактирование информации о товаре

При изменении какой-либо характеристики продукта или его основной информации или картинки и нажатии на кнопку «Сохранить» изменения автоматически отобразятся в базе данных и каталоге соответственно.

Также администратору доступна возможность удаления существующего товара. Для этого администратору необходимо найти интересующий его товар в каталоге, нажать на него дважды левой кнопкой мыши. После это откроется окно редактирования заказа, где доступна кнопка «Удалить товар». При нажатии на кнопку появится окно с подтверждением действия. В случае согласия подтвердить свои действия товар удалится из базы данных и каталога соответственно.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Г. Н. Федорова Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие / Г. Н. Федорова. – Москва : КУРС : ИНФРА-М, 2020 – 336 с. (Среднее профессиональное образование).
2. Гагарина Л. Г. Технология разработки программного обеспечения : учебное пособие / Л. Г. Гагарина, Е. В. Кокорина, Б. Д. Сидорова-Виснадул : под ред. Л. Г. Гагариной. – Москва : ИД «ФОРУМ» : ИНФРА-М, 2021. – 400 с. – (Среднее профессиональное образование).
3. Федорова Г. Н. Осуществление интеграции программных модулей: учебник для студ. учреждений сред. проф. образования / Г. Н. Федорова. 2-е изд., стер. – М. : Издательский центр «Академия», 2018. – 288 с.
4. Перлова О. Н. Проектирование и разработка информационных систем : учебник для студ. учреждений сред. проф. образования / О. П. Ляпина, А. В. Гусева. - 3-е изд., испр. – М. : Издательский центр «Академия», 2020. – 256 с.
5. Зверева В.П. Сопровождение и обслуживание программного обеспечения компьютерных систем : учебник для студ. учреждений сред. проф. образования / В. П. Зверева, А. В. Назаров. – 2-е изд., испр. – М. : Издательский центр «Академия», 2020. -256 с.

# **ПРИЛОЖЕНИЕ 1. Исходный программный код модуля «Авторизация»**

using PcComponents.Models.Entities;

using PcComponents.Views;

using System;

using System.Data.Entity;

using System.Drawing;

using System.Linq;

using System.Windows;

using System.Windows.Interop;

using System.Windows.Media;

using System.Windows.Media.Imaging;

namespace PcComponents

{

public partial class MainWindow : Window

{

int attempts = 0;

public MainWindow()

{

Bitmap img = Properties.Resources.noproduct;

ImageConverter converter = new ImageConverter();

Helper.NoImage = (byte[])converter.ConvertTo(img, typeof(byte[]))!;

InitializeComponent();

Bitmap logo = Properties.Resources.logo;

var handle = logo.GetHbitmap();

ImageSource isc = Imaging.CreateBitmapSourceFromHBitmap(handle, IntPtr.Zero, Int32Rect.Empty, BitmapSizeOptions.FromEmptyOptions());

imgIcon.Source = isc;

Helper.context = new PcComponentsDbWpfContext();

textBlockUserName.Text = "19dn@outlook.com";

passwordBoxUserPassword.Password = "RSbvHv";

}

private void btnLogIn\_Click(object sender, RoutedEventArgs e)

{

string login = textBlockUserName.Text;

string pass = passwordBoxUserPassword.Password;

if (attempts > 1)

{

CaptchaWindow cv = new CaptchaWindow();

this.Hide();

cv.ShowDialog();

this.Show();

}

User? usr = Helper.context.Users.Include(u => u.Role).Where(u => u.Login == login && u.Password == pass).FirstOrDefault();

if (usr == null)

{

tbFail.Text = "Неправильный логин или пароль!";

attempts++;

return;

}

tbFail.Text = "";

MessageBox.Show($"Добро пожаловать, {usr.Firstname} {usr.Patronymic}\n Роль: {usr.Role.RoleName}", "Приветствие");

Helper.UserName = $"{usr.Surename} {usr.Firstname} {usr.Patronymic}";

Helper.UserRole = usr.Role.RoleName;

MainView mv = new MainView();

this.Hide();

mv.ShowDialog();

this.Show();

}

private void btnGuest\_Click(object sender, RoutedEventArgs e)

{

Helper.UserRole = "Гость";

Helper.UserName = "";

MainView mv = new MainView();

this.Hide();

mv.ShowDialog();

this.Show();

}

}

}

# **ПРИЛОЖЕНИЕ 2. Исходный программный код модуля «Список заказов»**

using PcComponents.Models.Entities;

using PcComponents.ViewModels;

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace PcComponents.Views.Pages

{

public partial class OrderPage : Page

{

MainView parent;

public OrderPage(MainView mv)

{

this.parent = mv;

InitializeComponent();

InvalidateList();

cbOutposts.ItemsSource = (Helper.context.Outposts.ToList());

cbOutposts.DisplayMemberPath = "Address";

cbOutposts.SelectedValuePath = "Id";

cbOutposts.SelectedIndex = 0;

string fio = Helper.UserName == "" ? "Гость" : Helper.UserName;

tbFIO.Text = $"Имя клиента: {fio}";

}

private void orderProductList\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

int productId;

try

{

ListView? lv = sender as ListView;

if (lv?.SelectedItem == null) return;

productId = ((OrderedProductViewModel)((sender as ListView)!).SelectedItem).OrderedProduct.Product.Id;

}

catch (Exception ex) { return; }

this.NavigationService.Navigate(new ProductInformationPage(productId));

}

public void RecalculateCost()

{

double totalPrice = 0;

foreach (OrderedProduct op in Order.Products)

{

double priceWithoutDiscount = (double)op.Product.Price;

double discountAmount = (double)op.Product.Price \* (op.Product.Discount / 100.0);

double priceWithDiscount = priceWithoutDiscount - discountAmount;

totalPrice += priceWithDiscount \* op.Quantity;

}

tbCost.Text = "Итоговая стоимость: " + Math.Round(totalPrice,2).ToString() + " руб.";

}

public void InvalidateList()

{

orderProductList.ItemsSource = null;

List<OrderedProductViewModel> listOrderedProducts = new List<OrderedProductViewModel>();

foreach (OrderedProduct op in Order.Products)

{

OrderedProductViewModel opvm = new OrderedProductViewModel(op);

listOrderedProducts.Add(opvm);

}

if (listOrderedProducts.Count <= 0)

{

textBlockProductsHeader.Text = "Корзина пуста. Нет товаров";

}

else

{

textBlockProductsHeader.Text = "Список товаров в корзине";

}

orderProductList.ItemsSource = listOrderedProducts;

RecalculateCost();

}

private void plusProduct\_Click(object sender, RoutedEventArgs e)

{

int id = (int)(sender as Button)!.Tag;

OrderedProduct op = Order.Products.Where(oProduct => oProduct.Product.Id == id).FirstOrDefault()!;

op.Quantity++;

InvalidateList();

}

private void minusProduct\_Click(object sender, RoutedEventArgs e)

{

int id = (int)(sender as Button)!.Tag;

OrderedProduct op = Order.Products.Find(p => p.Product.Id == id)!;

op.Quantity--;

if (op.Quantity <= 0)

Order.RemoveProduct(id);

string cartButtonText = "Корзина";

if (Order.Products.Count > 0)

{

cartButtonText += $": ({Order.Products.Count})";

}

parent.textBlockCart.Text = cartButtonText;

InvalidateList();

}

private void tbProductQuantity\_TextChanged(object sender, TextChangedEventArgs e)

{

TextBox TextBox = (sender as TextBox)!;

int id = (int)TextBox.Tag;

OrderedProduct op = Order.Products.Where(op => op.Product.Id == id).FirstOrDefault()!;

int quantity = 0;

if (int.TryParse(TextBox.Text, out quantity))

{

op.Quantity = quantity;

RecalculateCost();

}

return;

}

private void Page\_Loaded(object sender, RoutedEventArgs e)

{

InvalidateList();

}

private void ButtonSaveOrder\_Click(object sender, RoutedEventArgs e)

{

if (orderProductList.Items.Count <= 0 )

{

MessageBox.Show("Корзина пуста!");

return;

}

PcComponents.Models.Entities.Order order = new PcComponents.Models.Entities.Order();

order.StatusId = 1;

order.OutpostId = (int) cbOutposts.SelectedValue;

order.OrderDate = DateTime.Now;

order.DeliveryDate = DateTime.Now.AddDays(7);

order.RecieveCode = Helper.context.Orders.Max(o => o.RecieveCode) + 1;

order.ClientFullName = Helper.UserName;

order.Id = Helper.context.Orders.Max(o => o.Id) +1;

Helper.context.Orders.Add(order);

Helper.context.SaveChanges();

foreach (OrderedProduct op in Order.Products)

{

OrderContent oc = new OrderContent();

oc.ProductId = op.Product.Id;

oc.OrderId = order.Id;

oc.Quantity = op.Quantity;

Helper.context.OrderContents.Add(oc);

}

Helper.context.SaveChanges();

MessageBox.Show("Ваш заказ оформлен\nЧерез 7 календарных дней можете забрать заказ в пункте выдачи\nНомер для полуения заказа: "+ order.RecieveCode);

Order.Products = new List<OrderedProduct>();

NavigationService.GoBack();

}

private void tbProductQuantity\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

try {

Convert.ToInt32(e.Text);

}

catch {

e.Handled = true;

}

}

}

}