### Частное учреждение образования

### «Колледж бизнеса и права»

### СОЗДАНИЕ ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ ОПТОВОЙ ФИРМЫ «ОПТИК»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### к курсовому проектированию по дисциплине

### «Конструирование программ и языки программирования»

КП Т.196008.401 ПЗ

Руководитель проекта (А.В.Кривошеина)

Учащийся (Е.В. Дронченко )

Минск, 2024

**СОДЕРЖАНИЕ**

Изм.

Изм.

Лист

Лист

№ докум.

№ докум.

Подпись

Подпись

Дата

Дата

Лист

Лист

3

3

КП Т.196008.401 ПЗ

ПЗ

ДП Т.517074.401 ПЗ

Разраб.

Разраб.

Дронченко Е.В.

Мовчан П.И.

Провер.

Провер.

Кривошеина А.В.

Багласова Е.В.

Т. контр.

Т. контр.

*.*

*Якимович К.О.*

Н. контр.

Н. контр.

Багласова Е.В.

Утверд.

Утверд.

*Багласова Т.Г.*

### *СОЗДАНИЕ ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ ОПТОВОЙ ФИРМЫ «ОПТИК»*

Лит.

Лит.

Листов

Листов

41

54

КБП

КБП

[Введение 3](#_Toc169941869)

[1. Описание задачи 4](#_Toc169941870)

[1.1. Анализ предметной области 4](#_Toc169941871)

[1.2. Постановка задачи 5](#_Toc169941872)

[2. Проектирование системы 7](#_Toc169941873)

[2.1. Требования к приложению 7](#_Toc169941874)

[2.2. Проектирование модели 7](#_Toc169941875)

[2.3. Организация данных 10](#_Toc169941876)

[2.4. Концептуальный прототип 12](#_Toc169941877)

[3. Описание реализации программного средства 13](#_Toc169941878)

[3.1. Инструменты разработки и применения технологии 13](#_Toc169941879)

[3.2. Порядок авторизации пользователей 14](#_Toc169941880)

[3.3. Организация данных 15](#_Toc169941881)

[3.4. Функции: логическая и физическая организация 17](#_Toc169941882)

[3.5. Входные и выходные данные 21](#_Toc169941883)

[3.6. Функциональное тестирование 21](#_Toc169941884)

[3.7. Описание справочной системы 25](#_Toc169941885)

[4. Применение 26](#_Toc169941886)

[4.1. Назначение программного средства 26](#_Toc169941887)

[4.2. Условия применения 26](#_Toc169941888)

[Заключение 28](#_Toc169941889)

[Список использованных источников 29](#_Toc169941890)

[Приложение А 31](#_Toc169941891)

[Приложение Б 34](#_Toc169941892)

[Приложение В 35](#_Toc169941893)

# **Введение**

В условиях современного рынка, где конкуренция растет, а требования клиентов к скорости и качеству обслуживания становятся все более высокими, автоматизация бизнес-процессов становится необходимой. Это особенно актуально для отрасли оптовой торговли, где эффективное управление запасами, заказами и взаимодействием с клиентами играет ключевую роль в успехе компании. Внедрение автоматизированных систем управления позволяет минимизировать временные и трудозатраты, повысить точность операций и улучшить качество обслуживания клиентов.

Целью курсового проекта «Автоматизированное рабочее место менеджера по работе с клиентами и сотрудника отдела продаж в оптово-торговой компании» является создание программного средства, которое позволит автоматизировать работу менеджера и сотрудника отдела продаж в оптово-торговой компании.

Пояснительная записка к курсовому проекту содержит пять разделов:

**Описание задачи**: В этом разделе раскрывается организационная сущность задачи, описывается предметная область и круг задач, которые должны быть решены. Описываются существующие аналоги.

**Проектирование системы**: В этом разделе перечисляются требования к аппаратному обеспечению и конфигурации компьютера, проводится характеристика операционной системы, обоснование выбранной среды для разработки приложения. Описываются требования к приложению, строится концептуальный прототип, описывается логическая и физическая организация данных, проектируется справочная система.

**Описание реализации программного средства**: В этом разделе представлены общие сведения о программном средстве и его функциональном назначении, входные и выходные данные.

**Применение**: Этот раздел предназначен для описания сведений о назначении программного средства и области его применения. В этом разделе приводится структура справочной системы.

Заключение будет проанализировано созданное программное средство, определена степень соответствия поставленной задачи и выполненной работы.

Приложение А будет содержать текст программы.

Графическая часть содержит в себе все необходимые диаграммы для проектирования данного программного средства, такие как:

* диаграмма вариантов использования;
* диаграмма деятельности;
* диаграмма классов;
* диаграмма последовательности.

# **Описание задачи**

# **Анализ предметной области**

Оптовая торговля – это процесс покупки и последующей перепродажи товаров в больших количествах. Она играет ключевую роль в цепочке поставок, обеспечивая движение товаров от производителей к розничным торговцам или другим покупателям, что способствует достижению эффективности и экономии за счет масштаба.

Программные средства для автоматизации оптовой торговли включают в себя комплекс инструментов и технологий, направленных на улучшение процессов управления закупками, продажами, складскими запасами, логистикой и финансовыми операциями. Цель таких решений – оптимизация бизнес-процессов, снижение операционных затрат и повышение конкурентоспособности компании.

Важными элементами оптовой торговли являются:

* управление закупками: это включает в себя автоматизация процессов выбора поставщиков, интеграция с системами управления запасами для оптимизации объема закупок, анализ и прогнозирование потребностей на основе данных о продажах и запасах;
* управление продажами: это включает в себя автоматизация обработки заказов от клиентов, включая создание и отправку счетов-фактур, поддержка различных каналов продаж (онлайн, офлайн);
* **управление складом**: это включает в себя управление процессами приема, хранения и отпуска товаров на складе;
* **управление информацией**: это включает в себя сбор, обработку и анализ данных для поддержки принятия решений в области торговли;
* финансовое управление: это включает в себя ведение учета финансовых операций, связанных с закупками и продажами;
* **управление складскими операциями**: это включает в себя управление процессами приема, хранения и отпуска товаров на складе;
* **управление транспортными операциями**: это включает в себя планирование и контроль над движением товаров от склада к потребителю. Это может включать в себя выбор транспортных средств, планирование маршрутов и управление процессами погрузки и разгрузки;
* **управление информацией**: это включает в себя сбор, обработку и распространение информации, необходимой для эффективного управления торговыми операциями. Это может включать в себя управление данными о товарах, поставщиках, заказах, доставках и другой информацией;
* **управление рисками**: это включает в себя идентификацию, оценку и управление рисками, связанными с торговыми операциями. Это может включать в себя риски, связанные с поставками, транспортировкой, складированием, информацией и другими аспектами торговли.

Промежуточные показатели, такие как статус доставки, пункт назначения, время в пути, могут быть получены через систему отслеживания в приложении. Итоговые показатели, такие как общее время доставки, стоимость транспортировки, эффективность использования транспортных средств, могут быть рассчитаны на основе собранных данных.

Автоматизация процессов в оптовой торговле с помощью современных программных решений позволяет значительно повысить эффективность работы, минимизировать риски и затраты, а также улучшить качество обслуживания клиентов, что в конечном итоге способствует успешному развитию бизнеса.

Целевая аудитория программных средств для автоматизации оптовой торговли: оптовые компании и дистрибьюторы, стремящиеся улучшить свои операционные процессы и увеличить конкурентоспособность, менеджеры по закупкам и продажам, которым необходимо оперативное и точное управление своими процессами, финансовые специалисты, ответственные за учет и анализ финансовых операций компании, сотрудники складов и логистических подразделений, которым важно оптимизировать свои ежедневные операции и сократить время на выполнение задач

# **Постановка задачи**

Исходя из анализа предметной области, можно выделить следующие задачи, подлежащие автоматизации в программном средстве для автоматизации оптовой торговли:

* автоматизация обработки заказов;
* автоматизировать управление запасами;
* автоматизация отслеживания поставок;
* автоматизация ведения отчетности;
* автоматизация управления клиентскими данными;

В сравнении с другими приложениями для автоматизации оптовой торговли, такими как «1С:Предприятие», «SAP», «TradeGecko», «NetSuite», «Microsoft Dynamics» и другими, данное программное средство имеет следующие преимущества:

* интеграция с различными системами, данное программное средство легко интегрируется с другими системами управления предприятиями (ERP), системами управления складами (WMS) и бухгалтерским ПО, что обеспечивает бесшовный обмен данными и улучшает общую эффективность бизнеса;
* гибкость и масштабируемость: Оптик спроектирован так, чтобы быть гибким и масштабируемым, позволяя легко добавлять новые функции и поддерживать большое количество пользователей. Это делает его подходящим для различных типов и размеров бизнеса;
* аналитика и прогнозирование: встроенные инструменты аналитики и прогнозирования помогают пользователям принимать более обоснованные решения, основываясь на данных о продажах, запасах и поставках.

Таким образом, программное средство Оптик предлагает уникальный набор функций, которые помогут улучшить эффективность и производительность в области оптовой торговли, делая его важным инструментом для любого бизнеса, стремящегося оставаться конкурентоспособным в современном быстро меняющемся бизнес-окружении.

# **Проектирование системы**

# **Требования к приложению**

В приложении будет реализована система аутентификации для обеспечения безопасности данных пользователей, доступ к функционалу будет возможен только после ввода уникальных учетных данных (имя пользователя и пароль). Кроме того, функции будут ограничены в зависимости от роли пользователя, например, администратор, менеджер, оператор и другие.

Обработка заказов будет автоматизирована, включая автоматическую обработку входящих заказов от клиентов и возможность отслеживания их статуса в реальном времени. Также будет реализована функция уведомлений о новых заказах и изменении их статуса.

Производительность приложения будет обеспечена быстрой обработкой запросов и минимальным временем отклика системы, а также оптимизацией работы с большими объемами данных. Приложение будет совместимо с различными операционными системами, включая Windows, macOS и Linux, и сможет работать на различных устройствах, таких как настольные ПК, ноутбуки и планшеты.

Основными минимальными требованиями, выдвигаемыми к аппаратному обеспечению персонального компьютера, являются:

* процессор 800 МГц и выше;
* оперативная память 128 Мбайт и более;
* свободное место на диске 100 Мбайт;
* интегрированная видеокарта на 512 Мбайт и более;
* монитор;
* мышь, клавиатура;
* принтер.

# **Проектирование модели**

Главной целью проектирования моделей является отображение функциональной структуры объекта, то есть производимые ими действия и связи между этими действиями.

Наиболее распространенные средства моделирования являются: диаграмма вариантов использования, диаграмма классов, диаграмма последовательности, диаграмма деятельности.

Диаграмма вариантов использования – это тип диаграммы, используемой в области разработки программного обеспечения для визуализации системы с точки зрения её акторов и вариантов использования.

Основные элементы диаграммы вариантов использования:

* актеры: ­ актеры представляют собой людей или системы, которые взаимодействуют с системой. Они находятся вне системы и могут быть как людьми (например, пользователи или администраторы), так и другими системами или устройствами;
* варианты использования: варианты использования – это функции или действия, которые система может выполнять в ответ на запросы от актеров. Они представляют собой цели, которые ­ актеры хотят достичь с помощью системы;
* отношения: отношения между актёрами и вариантами использования показывают, как ­ актеры и варианты использования взаимодействуют друг с другом. Они могут быть обозначены стрелками или линиями.

Диаграмма вариантов использования помогает разработчикам и стейкхолдерам лучше понять, как система будет работать и какие функции она будет выполнять. Она также помогает определить требования к системе и спланировать её разработку. Это важный инструмент в процессе проектирования системы.

Диаграмма вариантов использования представлена на листе 1 графической части.

На этой диаграмме можно выделить двух пользователей программным продуктом. Клиент, который может обратиться к менеджеру с просьбой о оформлении заказа. Менеджер, который добавляет заказ в систему и может его редактировать, а также создавать заявки на перевозку посылок.

Диаграмма классов – это структурная диаграмма языка моделирования UML, которая демонстрирует общую структуру иерархии классов системы, их атрибуты (полей), методы, Макеты и взаимосвязи (отношений) между ними. Она используется для визуализации структуры классов в системе и их взаимосвязей.

Вот основные элементы диаграммы классов:

* Классы: Классы представляют собой ключевые элементы в объектно-ориентированном моделировании. На диаграмме классы представлены в рамках, содержащих три компонента: имя класса, поля (атрибуты) класса и методы класса.
* Отношения: Отношения между классами на диаграмме классов могут представлять отношения, такие как ассоциация, наследование и реализация.

Диаграмма классов соответствует принципам объектно-ориентированного программирования (ООП) и является одним из базовых инструментов проектирования ООП-систем. Она помогает лучше понимать структуру системы и ее компоненты.

В данном приложении присутствуют следующие классы:

* пользователь;
* заказ;
* товар;
* клиент.

Диаграмма классов представлена на листе 2 графической части.

Диаграмма деятельности – это графическое представление процессов и задач, выполняемых в рамках определенного проекта или деятельности. Эта диаграмма используется для анализа и оптимизации бизнес-процессов, планирования работы, принятия решений и управления проектами.

Основные элементы диаграммы деятельности:

* начальный узел: Начальный узел активности является первым элементом диаграммы деятельности и представляет начальную точку процесса;
* конечный узел активности: Конечный узел активности является последним элементом диаграммы деятельности и обозначает завершение процесса;
* действие: Действия представляют собой отдельные шаги или операции, которые выполняются в процессе;
* поток управления: Эти потоки показывают последовательность действий и передачу данных между действиями;
* узел принятия решений: Эти узлы используются для представления ветвлений и слияний в потоке управления.

Диаграмма деятельности позволяет визуализировать последовательность задач и операций, которые нужно выполнить, чтобы достичь желаемого результата. Она является мощным инструментом для анализа, планирования и управления проектами и бизнес-процессами.

В рамках проекта рассмотрим функцию добавления заказа на перевозку.

Диаграмма деятельности для создания посылки представлена на листе 3 графической части.

Диаграмма последовательности – это UML-диаграмма, которая показывает взаимодействие между объектами в системе в виде последовательности сообщений, действий и операций. Она отображает порядок выполнения действий и обмена информацией между объектами во времени.

Вот основные элементы диаграммы последовательности:

* объекты: представлены горизонтальной осью диаграммы. Каждый объект имеет свою колонку;
* сообщения: передаются горизонтально между объектами и располагаются вертикально в хронологическом порядке. Они представлены стрелками;
* время: представлено вертикальной осью диаграммы. Жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актёров (действующих лиц) информационной системы в рамках прецедента показаны на единой временной оси.

Диаграмма последовательности помогает анализировать и понимать взаимодействие между объектами в системе, а также порядок, в котором происходит взаимодействие. Она является важным инструментом для анализа и проектирования систем.

Диаграмма последовательности для создания заказа представлена на листе 4 графической части.

# **Организация данных**

Наиболее распространенным средством моделирования таких типов документации являются схемы-сети (ERD), которые применяются для графического представления многих элементов разработанных программ систем и представляют некоторые габаритные размеры обозначений для определения данных и отношений между ними. С помощью такой диаграммы можно описать отдельные компоненты концептуальной модели или выразить свою точку зрения взаимосвязей между ними, иногда какое значение или реализацию системы. Основными потенциалом данной нотации является понятная сущность в каждом случае каждый рассматриваемый объект может явиться экземпляром одного и только одного сущности, должен иметь уникальное имя или идентификатор, а также отличается от других экземпляров данной сущности. Следовательно: как фотонные линии короткого ассортимента между отделами сущности.

Графическая модель данных строится таким образом, чтобы связь между отделениями сущности органа не только естественный характер соответствующего отношения, но и дополнительные аспекты обязательности связей.

Информационная модель представлена на диаграмме «Сущность-связь» на рисунки 2.2

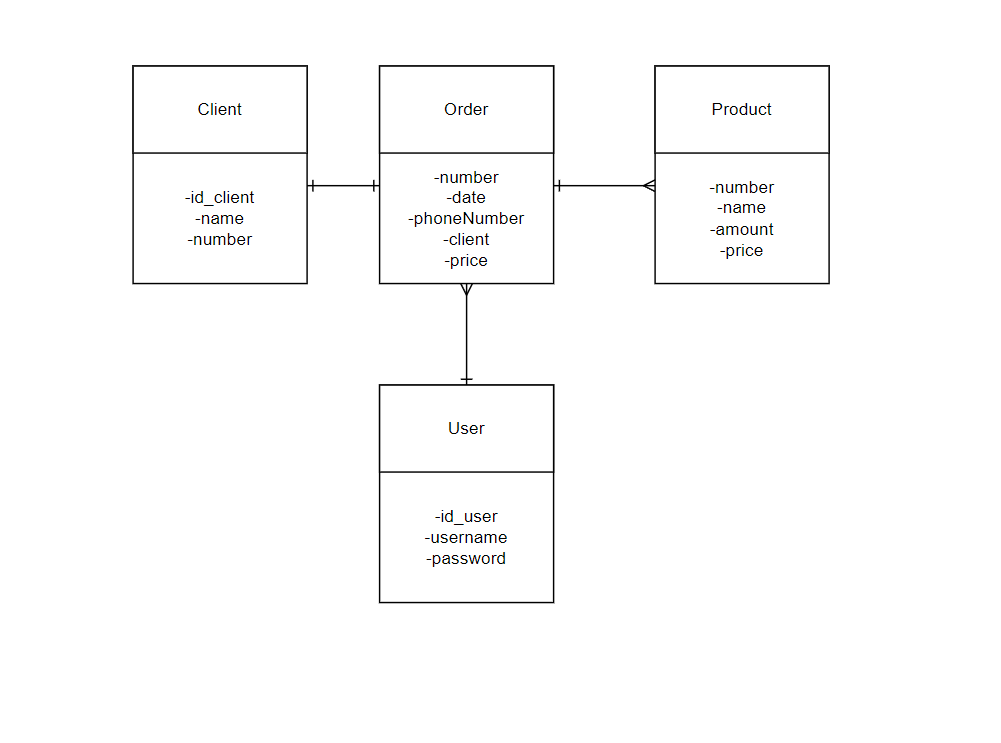


Рисунок 2.2 – Диаграмма «Сущность-связь»

Исследовав предметную область, можно выделить следующие сущности: «Заказ», «Товар», «Клиент», «Пользователь»

Для сущности «Заказ» атрибутами будут являться:

* номер
* дата;
* номер телефона;
* клиент;
* общая стоимость;

Для сущности «Товар» атрибутами будут являться:

* номер;
* наименование;
* количество;
* стоимость;

Для сущности «Клиент» атрибутами будут являться:

* номер;
* ФИО;
* Номер телефона;

Для сущности «Пользователь» атрибутами будут являться:

* номер;
* логин;
* пароль;

# **Концептуальный прототип**

Приложение "Оптик" включает несколько интерфейсов для различных задач автоматизации оптовой торговли.

Интерфейс "Авторизация" включает два текстовых поля, подписанных как "Фамилия" и "Пароль". Под этими полями находятся две кнопки: "Войти" и "Отмена". Пользователи вводят свои учетные данные в соответствующие поля и нажимают кнопку "Войти" для доступа к функционалу приложения. Кнопка "Отмена" позволяет пользователям выйти из приложения.

Главное окно "Выбор" состоит из заголовка "Выберите действие" и трех кнопок: "Управление заказами", "Управление складом" и "Выход". Кнопка "Управление заказами" переводит пользователя в окно "Заказы" для работы с заказами, кнопка "Управление складом" переводит в окно "Товар" для работы с товарами, а кнопка "Выход" предназначена для выхода из приложения.

Интерфейс "Заказы" включает таблицу (Datagrid) для отображения данных из базы данных, кнопки "Добавить заказ", "Удалить заказ", "На главную" и "Выход". Также имеется текстовое поле для ввода номера заказа и кнопка "Найти", позволяющая искать заказы по номеру.

Интерфейс "Добавление заказа" состоит из трех текстовых полей для ввода данных. После заполнения всех полей пользователи могут нажать кнопку "Добавить товар" для добавления заказа. Также есть кнопка "Калькулятор" для расчета стоимости товара и кнопка "Отмена" для возврата к предыдущему окну.

Интерфейс "Калькулятор суммы" включает три надписи (label), два текстовых поля для ввода данных и две кнопки: "Расчитать" и "Отмена". Надпись "Здесь будет ответ" выводит результат после расчета. Кнопка "Расчитать" становится активной после заполнения всех полей, а кнопка "Отмена" возвращает пользователя на предыдущую форму.

Интерфейс "Товары" включает таблицу (Datagrid) для отображения данных из базы данных, кнопки "Добавить товар", "Удалить товар", "На главную" и "Выход". Также имеется текстовое поле для ввода номера товара и кнопка "Найти", позволяющая искать товары по номеру.

Интерфейс "Добавление товара" состоит из трех текстовых полей для ввода данных и двух кнопок: "Добавить товар" и "Отмена". Кнопка "Добавить товар" активируется после заполнения всех полей, а кнопка "Отмена" возвращает пользователя к предыдущей форме.

Макеты всех вышеописанных форм будут предоставлены в приложении В с подробными рисунками каждого интерфейса, отображающими расположение и внешний вид элементов управления.

# **Описание реализации программного средства**

# **Инструменты разработки и применения технологии**

Инструментами разработки для будущего программного приложения будут являться:

* операционная система MS Widows 10;
* программная среда разработки Microsoft Visual Studio 2022;
* язык программирования C#;
* система управления реляционными базами данных (СУБД) Microsoft SQL Server 2024;
* программа для построения блок-схем Microsoft Visio
* язык определения данных DDL;
* справочная система Dr.Explain;
* офисный пакет приложений Microsoft Office.

MS Windows 10 – операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства MS Windows. Доступна единая платформа разработки и единый магазин универсальных приложений, совместимых со всеми поддерживаемыми устройствами [19].

Microsoft Visual Studio 2022 – это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского Макета, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию [14].

MS Visual Studio 2022 включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня [14].

Microsoft SQL Server – система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка [15].

DDL состоит из набора SQL-команд, используемых для создания, изменения и удаления структур баз данных, но не данных. Он просто имеет дело с описаниями схемы базы данных и используется для создания и изменения структуры объектов, присутствующих в базе данных. Все DDL-команды автоматически фиксируются, что означает, что все изменения навсегда сохраняются в базе данных. DDL подразделяется на пять команд, которые широко используются в SQL-запросах[15].

Microsoft Visio – векторный графический редактор, редактор диаграмм и блок-схем для Windows. Выпускается в трёх редакциях: Standard, Professional и Pro for Office 365.Аналогично с Adobe Reader, в стандартный набор программ Microsoft Office входит только средство для просмотра и печати диаграмм Microsoft Visio Viewer. Полнофункциональная версия Microsoft Visio Professional для создания и редактирования монограмм и диаграмм. Первоначально Visio разрабатывался и выпускался компанией Shapeware, затем переименованной в Visio Corporation. Microsoft приобрела компанию в 2000 году, тогда продукт назывался Visio 2000 [18].

C# – объектно-ориентированный язык программирования общего назначения. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота[6] как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML [16].

Dr.Explain – это профессиональная программа для создания электронных справочных руководств, пользовательских инструкций и баз знаний. Она автоматизирует создание скриншотов, аннотацию изображений, индексацию контента и форматирование документации.

Microsoft Office – это офисный пакет, разработанный американской корпорацией Microsoft. В его состав входят приложения для работы с текстовыми документами, электронными таблицами, базами данных и другими типами файлов. Он предоставляет инструменты, такие как Word, Excel, PowerPoint, Outlook, Teams и многое другое [17].

# **Порядок авторизации пользователей**

В приложении можно выделить следующие роли и функции:

Менеджер по заказам:

* Аутентификация: Менеджер регистрируется в системе, предоставляя необходимые учетные данные (например, имя пользователя и пароль);
* Авторизация: после успешной аутентификации менеджер получает доступ к функциям, связанным с его ролью;
* Функции: Создание заказов, их редактирование и удаление. Это может включать ввод информации о заказе.

Управляющий складом:

* Аутентификация: сотрудник регистрируется в системе, предоставляя необходимые учетные данные;
* Авторизация: после успешной аутентификации сотрудник получает доступ к функциям, связанным с его ролью;
* Функции: Добавление товаров, редактирование и удаление. Это может включать ввод информации о товаре.

# **Организация данных**

Организация данных подразумевает создание модели данных, главными элементами, которые являются сущности и их связи.

Реляционная модель основана на математическом понятии отношения, представлением которого является таблица. В реляционной модели отношения используются для хранения информации об объектах, представленных в базе данных. Отношение имеет вид двухмерной таблицы, в которой строки соответствуют записям, а столбцы – атрибутам. Каждая запись должна однозначно характеризоваться в таблице. Для этого используют первичные и вторичные ключи. Достоинством реляционной модели является простота и удобство физической реализации.

Реляционная модель базы данных подразумевает нормализацию всех таблиц данных. Нормализация – это формальный метод анализа отношений на основе их первичного ключа и функциональных зависимостей, существующих между их атрибутами.

Структура базы данных разрабатываемого программного средства включает четыре таблицы.

Таблица «Clients» содержит информацию о клиентах, структура таблицы представлена в таблице 3.1

Таблица 3.1 – Clients

|  |  |  |  |
| --- | --- | --- | --- |
| Название строки | Тип данных | Размер в байтах | Описание |
| Id\_client | int | 4 | Номер клиента |
| Name | nvarchar(50) | 100 | ФИО клиента |
| number | nvarchar(50) | 100 | Номер телефона |

Таблица «Order» содержит информацию о заказах, структура таблицы представлена в таблице 3.2

Таблица 3.2 – Order

|  |  |  |  |
| --- | --- | --- | --- |
| Название строки | Тип данных | Размер в байтах | Описание |
| id | int | 4 | Номер заказа |
| Client | nvarchar(50) | 100 | Наименование Юр. Лица |
| Price | float | 4 | Стоимость |
| Number | int | 4 | Артикул заказа |
| PhoneNumber | nvarchar(50) | 100 | Номер телефона |
| Date | nvarchar(MAX) | 100 | Время |

Таблица «Product» содержит информацию о товарах, структура таблицы представлена в таблице 3.3

Таблица 3.3 – Product

|  |  |  |  |
| --- | --- | --- | --- |
| Название строки | Тип данных | Размер в байтах | Описание |
| id | int | 4 | Номер товара |
| Name | nvarchar(50) | 100 | Наименование товара |
| Number | int | 4 | Артикул товара |
| Amount | int | 4 | Кол-во |
| Price | float | 4 | Цена |

Таблица «Register» содержит информацию о пользователях, структура таблицы представлена в таблице 3.4

Таблица 3.4 – Register

|  |  |  |  |
| --- | --- | --- | --- |
| Название строки | Тип данных | Размер в байтах | Описание |
| id\_user | int | 4 | Номер пользователя |
| login\_user | varchar(50) | 250 | Логин пользователя |
| password\_user | varchar(50) | 250 | Пароль пользователя |

Все таблицы представлены на рисунке 3.1

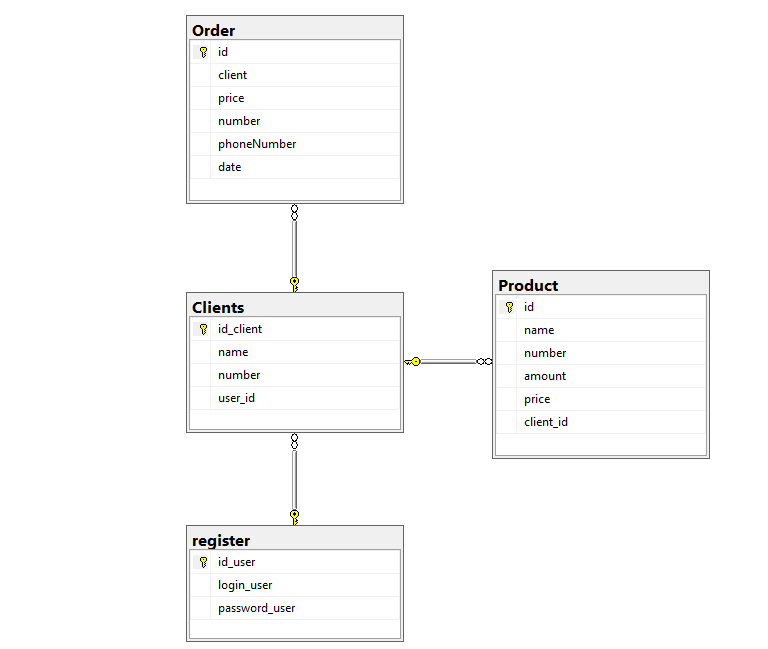


Рисунок 3.1 – Схема данных

# **Функции: логическая и физическая организация**

В данном проекте реализованы функции добавления, редактирования и удаления информации о заказе, товаре; формирование счет-фактуры; формирование поиска.

Функция добавления информации о заказе находиться на форме «Добавление заказа». После заполнения всех полей и нажатия на кнопку «Добавить» на данной форме происходит обработка события Button\_Click\_AddOrder. Все данные, вводимые в поля формы, будут сохранены в таблице «Order». Код функции добавления заказа:

private void \_\_\_button1\_\_Click(object sender, RoutedEventArgs e)

{

Random rand = new Random();

try

{

OrderClass order = new OrderClass

{

Number = rand.Next(),

Client = \_\_\_textbox1\_.Text,

Price = Convert.ToInt32(\_\_\_textbox2\_.Text),

PhoneNumber = \_\_\_textbox3\_.Text,

Date = DateTime.Now

};

Main2 main2 = (Main2)Application.Current.MainWindow;

main2.AddOrder(order);

CreateOrderDocument(order);

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

public void AddOrder(OrderClass order)

{

try

{

DataRow newRow = dt.NewRow();

newRow["Number"] = order.Number;

newRow["Client"] = order.Client;

newRow["Price"] = order.Price;

newRow["PhoneNumber"] = order.PhoneNumber;

newRow["Date"] = order.Date;

dt.Rows.Add(newRow);

SaveOrder(order);

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

Функция редактирования информации о заказе находится на форме «Добавление заказа». После заполнения всех полей и нажатия на кнопку «Сохранить» на данной форме происходит обработка событий Button\_Click. Все данные, вводимые в поля формы, будут сохранены в таблице «Order».

Функция удаления информации о заказе находится на форме «Главная». После нажатия на кнопку «Удалить» на данной форме происходит обработка событий deleteOrder\_Click. Выбранные данные будут удалены из таблицы «Order».

Функция добавления информации о товаре находиться на форме «Товары». После заполнения всех полей и нажатия на кнопку «Добавить» на данной форме происходит обработка события Button\_Click. Все данные, вводимые в поля формы, будут сохранены в таблице «Product».

Функция редактирования информации о товаре находится на форме «Добавление товара». После заполнения всех полей и нажатия на кнопку «Сохранить» на данной форме происходит обработка событий Button\_Click. Все данные, вводимые в поля формы, будут сохранены в таблице «Product».

Функция удаления информации о товаре находится на форме «Главная». После нажатия на кнопку «Удалить» на данной форме происходит обработка событий deleteProduct\_Click. Выбранные данные будут удалены из таблицы «Product».

Функция формирования созданной счет-фактуры происходит при нажатии на кнопку «Сформировать» на форме «Главная». Происходит обработка события Button\_Click, в процессе которого будет создан документ Word со всеми данными счет-фактуры. Код функции создания документа word:

private void CreateOrderDocument(OrderClass order)

{

string fileName = $"Заказ\_{order.Number}.docx";

var doc = DocX.Create(fileName);

var title = doc.InsertParagraph("Счет-фактура")

.FontSize(24)

.Bold()

.SpacingAfter(10)

.Alignment = Alignment.center;

doc.InsertParagraph($"Номер заказа: {order.Number}")

.FontSize(16)

.SpacingAfter(5);

doc.InsertParagraph($"Клиент: {order.Client}")

.FontSize(16)

.SpacingAfter(5);

doc.InsertParagraph($"Цена: {order.Price} руб.")

.FontSize(16)

.SpacingAfter(5);

doc.InsertParagraph($"Номер телефона: {order.PhoneNumber}")

.FontSize(16)

.SpacingAfter(5);

doc.InsertParagraph($"Дата: {order.Date}")

.FontSize(16)

.SpacingAfter(10);

var table = doc.AddTable(6, 2);

table.Alignment = Alignment.center;

table.Design = TableDesign.LightShadingAccent1;

table.Rows[0].Cells[0].Paragraphs[0].Append("Поле").Bold().FontSize(16);

table.Rows[0].Cells[1].Paragraphs[0].Append("Значение").Bold().FontSize(16);

table.Rows[1].Cells[0].Paragraphs[0].Append("Номер заказа").FontSize(14);

table.Rows[1].Cells[1].Paragraphs[0].Append(order.Number.ToString()).FontSize(14);

table.Rows[2].Cells[0].Paragraphs[0].Append("Клиент").FontSize(14);

table.Rows[2].Cells[1].Paragraphs[0].Append(order.Client).FontSize(14);

table.Rows[3].Cells[0].Paragraphs[0].Append("Цена").FontSize(14);

table.Rows[3].Cells[1].Paragraphs[0].Append($"{order.Price} руб.").FontSize(14);

table.Rows[4].Cells[0].Paragraphs[0].Append("Номер телефона").FontSize(14);

table.Rows[4].Cells[1].Paragraphs[0].Append(order.PhoneNumber).FontSize(14);

table.Rows[5].Cells[0].Paragraphs[0].Append("Дата").FontSize(14);

table.Rows[5].Cells[1].Paragraphs[0].Append(order.Date.ToString()).FontSize(14);

doc.InsertParagraph().InsertTableAfterSelf(table);

doc.InsertParagraph("Дополнительная информация:")

.FontSize(18)

.Bold()

.SpacingAfter(8);

doc.InsertParagraph("Доставка:")

.FontSize(16)

.Bold()

.SpacingAfter(5);

doc.InsertParagraph("Адрес доставки: ул. Примерная, д. 1, кв. 1")

.FontSize(14)

.SpacingAfter(5);

doc.InsertParagraph("Дата доставки: " + order.Date.AddDays(3).ToString("dd.MM.yyyy"))

.FontSize(14)

.SpacingAfter(5);

doc.InsertParagraph("Комментарий к заказу: Пожалуйста, звоните за 30 минут до доставки.")

.FontSize(14)

.SpacingAfter(10);

doc.InsertParagraph("Оплата:")

.FontSize(16)

.Bold()

.SpacingAfter(5);

doc.InsertParagraph("Метод оплаты: Наличные")

.FontSize(14)

.SpacingAfter(5);

doc.InsertParagraph("Сдача с: 5000 руб.")

.FontSize(14)

.SpacingAfter(10);

doc.InsertParagraph("Дополнительные условия:")

.FontSize(16)

.Bold()

.SpacingAfter(5);

doc.InsertParagraph("При возникновении вопросов, пожалуйста, свяжитесь с нашим оператором по номеру +375 (29) 728-57-25.")

.FontSize(14)

.SpacingAfter(10);

doc.InsertParagraph()

.AppendLine(new string(' ', 3000))

.SpacingAfter(5);

doc.Save();

MessageBox.Show($"Документ заказа создан: {fileName}");

}

Функция формирования созданного документа информации об посылках заказчика происходит при нажатии на кнопку «Создать» на форме «Документы». Происходит обработка события Button\_Click, в процессе которого будет создан документ Word со всеми данными о заказах.

Полный код программных модулей представлен в приложении А.

# **Входные и выходные данные**

Входными данными являются данные, заносимые программой в таблицы: «Order», «Product», «Register». Также входными данными являются данные, вводимые в поля для поиска.

Для добавления данных в таблицу «Order» необходимо ввести следующие данные:

* клиент;
* конечная сумма;
* номер заказа;
* номер телефона;
* дата создания заказа.

Для добавления данных в таблицу «Product» необходимо ввести следующие данные:

* наименование товара;
* номер товара;
* количество;
* цена.

Для добавления данных в таблицу «Register» необходимо ввести следующие данные:

* логин пользователя;
* пароль.

Выходные данными являются данные, которые экспортируются в файлы Microsoft Office, а именно счет-фактура заказа.

Договор генерируется по нажатию на кнопку «Создать». Выходные документы представлены в приложении В.

# **Функциональное тестирование**

Функциональное тестирование – это тестирование функций приложения на соответствия требованиям. Оценка производится в соответствии с ожидаемым и фактическим результатом.

Тестирование программы будет производится последовательно, переходя из одной программы в другую. Во время теста будут проверятся все действия с программой, навигация пунктов меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

В таблицах 3.7-3.11 представлены тест-кейсы, подготовленные для проведения функционального тестирования.

Первый тест будет проведён в окне авторизации программы.

Тест-кейс для тестирования функции авторизации представлен в таблице 3.7

Таблица 3.7 – Тест-кейс функции авторизации

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | Функция | Шаги выполнения | Результат |
| 1 | Авторизация в приложении | 1. Заполнить поля данными:  - Фамилия: «admin»;  - Пароль: «123».  2. Нажать на кнопку «Войти» | Ожидаемый результат: переход на форму «Главное окно» |
| Фактический результат: успешный переход на форму «Главное окно» |

Второй тест будет произведен на форме «Заказы».

Тест-кейс для тестирования функции добавления заказа представлен в таблице 3.8

Таблица 3.8 – Тест-кейс функции добавления заказа

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | Функция | Шаги выполнения | Результат |
| 2 | Добавление заказа | 1. На главной форме нажать на кнопку «Заказы»;  2. Нажать на кнопку «Добавить заказ»; 3. Заполнить поля данными: - Наименование юр. лица: «ОАО БибасикПро»;  - Конечная сумма: «2500»; - Номер телефона: «+375 (29) 728-57-25»;  - Нажать кнопку «Добавить заказ» | Ожидаемый результат: Добавление заказа в таблицу и в базу данных. |
| Фактический результат: успешное добавление в таблице и в базе данных. |

Третий тест будет произведен на форме «Склад».

Тест-кейс для тестирования функции добавления товара представлен в таблице 3.9

Таблица 3.9 – Тест-кейс функции добавления товара

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | Функция | Шаги выполнения | Результат |
| 3 | Добавление товара | 1. На главной форме нажать на кнопку «Товары»;  2. Нажать на кнопку «Добавить товар»; 3. Заполнить поля данными: -Наименование товара: «Мышка Logitech G102»;  - Цена: «100»; - Количество: «500»; 4. Нажать на кнопку «Добавить товар». | Ожидаемый результат: Добавление товара в таблицу. |
| Фактический результат: успешное добавление в таблице. |

Четвертый тест будет произведен на форме «Склад».

Тест-кейс для тестирования функции редактирования данных о товаре представлен в таблице 3.9

Таблица 3.9 – Тест-кейс функции изменения данных о товаре

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | Функция | Шаги выполнения | Результат |
| 4 | Редактирование товара | 1. На главной форме нажать на кнопку «Товары»;  2. Нажать на товар «Мышка Logitech G102»; 3. Нажать кнопку «Редактировать товар»;  4. Изменить значения поля количество на 250; 5. Нажать на кнопку «Сохранить». | Ожидаемый результат: Данные изменились в таблице и в базе данных. |
| Фактический результат: Данные успешно изменились в таблице и в базе данных. |

Пятый тест будет произведен на форме «Заказы».

Тест-кейс для тестирования функции редактирования данных о заказе представлен в таблице 3.10

Таблица 3.10– Тест-кейс функции редактирования данных о заказе

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | Функция | Шаги выполнения | Результат |
| 5 | Изменение заказа | 1. На главной форме нажать на кнопку «Заказы»;  2. Нажать на заказ с юр. лицом «ОАО БибасикПро» 3. Нажать кнопку «Редактировать»;  4. Изменить значения поля цена на 1000; 5. Нажать на кнопку «Сохранить». | Ожидаемый результат: Значение поля цена изменились в таблице и в базе данных. |
| Фактический результат: Значение поля цена изменились в таблице и в базе данных успешно. |

Первый тест будет произведен на форме «Заказы».

Тест-кейс для тестирования функции поиска заказа представлен в таблице 3.11

Таблица 3.11 – Тест-кейс функции поиска данных о заказе

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | Функция | Шаги выполнения | Результат |
| 6 | Поиск заказа | 1. На главной форме нажать на кнопку «Заказы»  2. Заполнить поле поиска данными: «241199919»;  3. Нажать на кнопку «Поиск». | Ожидаемый результат: Показ результата поиска. |
| Фактический результат: успешный поиск и вывод. |

# **Описание справочной системы**

Справочная система была разработана с использованием программного обеспечения Dr.Explain.

Справочная система включает в себя несколько ключевых разделов, которые были определены в предоставленном изображении:

* начальная страница;
* авторизация пользователя;
* выбор рабочего места;
* заказы;
* товары.

Справочная система может быть вызвана из главной формы при нажатии на кнопку «Справка». Справка представлена на рисунке 3.3

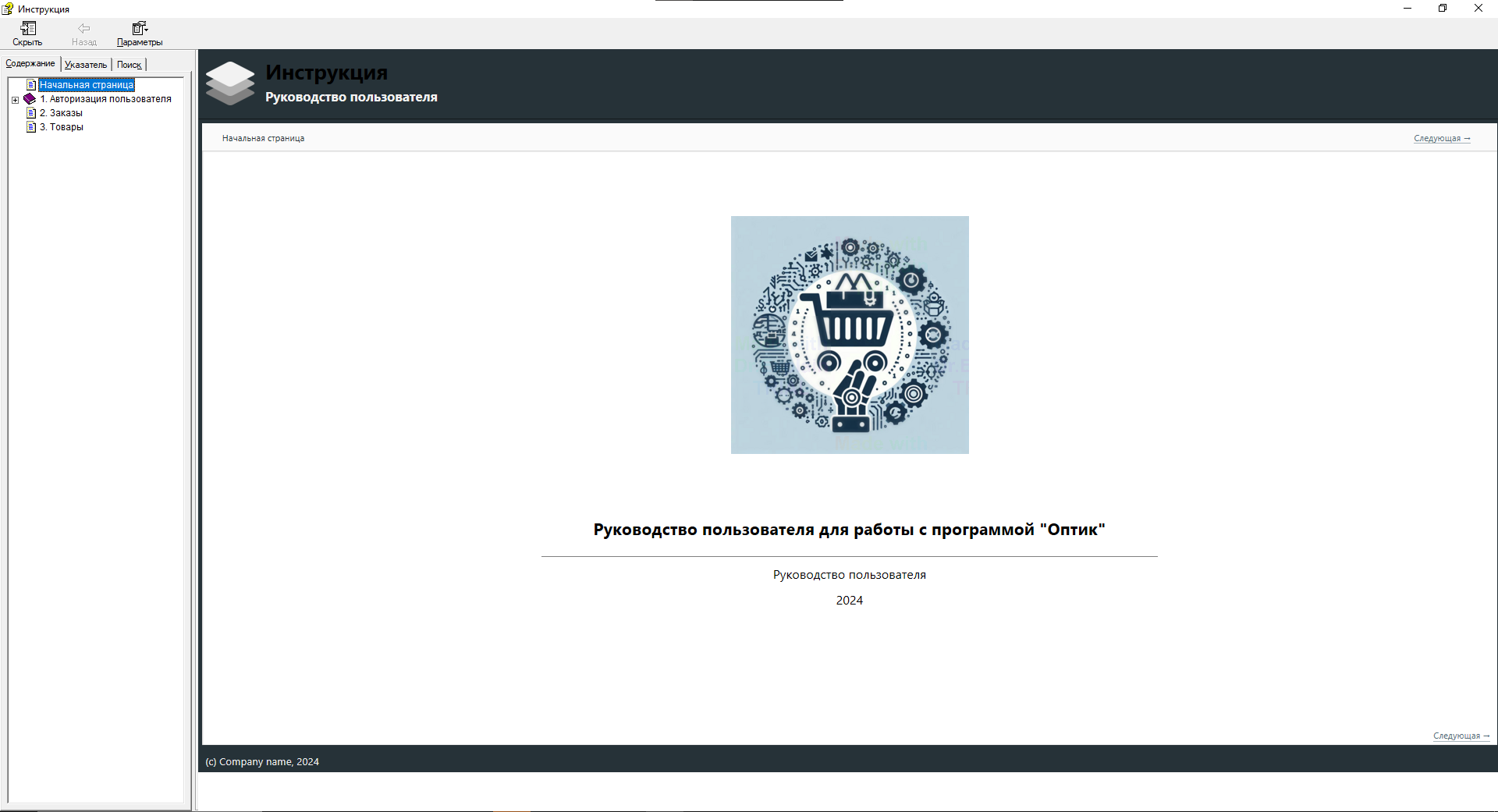


Рисунок 3.3 – Справка

# **Применение**

# **Назначение программного средства**

Приложение «Optik.exe» предназначено для автоматизации процессов оптовой торговли. Программное средство предоставляет пользователю возможности для поиска, добавления, отображения и редактирования объектов, таких как товары, заказы, клиенты, финансовые операции, что уменьшает затраты времени, используемого при управлении оптовыми продажами и складскими запасами. Оно ведет учет товаров и заказов клиентов, что способствует более эффективному управлению торговыми операциями. Под эту роль был создан специальный макет, который позволяет менеджерам быстро и удобно работать с заказами и запасами.

Основные разработанные функции:

* авторизация работников в системе;
* возможность добавлять и редактировать заказы;
* возможность добавлять и редактировать товары;
* возможность экспорта данных в документы Microsoft Office.

Приложение «Optik.exe» разработано специально для оптовой торговли и не предназначено для использования в других отраслях, таких как производство или розничная торговля. Оно оптимизировано для работы с данными, связанными с оптовыми продажами и складскими операциями, и может не подходить для управления данными, не относящимися к торговле. Кроме того, функциональность экспорта ограничена поддержкой документов Microsoft Office, что может не соответствовать потребностям компаний, использующих другие форматы для документооборота.

# **Условия применения**

Для применения данного программного средства необходимы следующие технические требования:

* процессор Intel Core i5 или выше;
* минимальный объём оперативной памяти 500 Мбайт;
* оперативная система Windows 11;
* платформа .Net Framework v4.8;
* видеокарта с объёмом памяти не менее 256 Мбайт;
* клавиатура, мышь, принтер.

Процесс установки программного средства «Optik.exe» включает следующие шаги:

* загрузка программы: скачайте установочный файл «Optik.exe» с github или носителя, предоставленного разработчиком;
* установка: запустите скачанный установочный файл и следуйте инструкциям мастера установки.

# **Заключение**

Проект “Optik.exe” представляет собой программное средство, разработанное для автоматизации рабочего места менеджера по оптовой торговле и сотрудника складского отдела в торговой компании. Организационная сущность задачи заключается в создании инструмента, который облегчит работу менеджеров, складских работников и финансовых аналитиков в области оптовой торговли. Предметная область проекта охватывает управление запасами, обработку заказов, учет финансовых операций, взаимодействие с клиентами и генерацию аналитических отчетов.

Технические требования для программы включают процессор Intel Core i5 или выше, минимальный объем оперативной памяти 500 Мбайт, операционную систему Windows 11, платформу .Net Framework v4.8 и видеокарту с объемом памяти не менее 256 Мбайт. Логическая организация данных включает описание структуры данных и связей между объектами (товары, заказы, клиенты, финансовые операции). Физическая организация данных включает выбор СУБД и оптимизацию запросов. Справочная система описывает функциональность и предоставляет помощь пользователям.

Программное средство Optik.exe предоставляет возможности поиска, добавления, отображения и редактирования объектов в базе данных. Входные данные включают информацию о товарах, заказах, клиентах и финансовых операциях. Выходные данные могут быть отчетами, аналитическими данными и другой информацией.

Среди преимуществ разработанного ПС можно выделить автоматическое управление запасами, гибкость и масштабируемость, а также интуитивный пользовательский макет. Однако следует учитывать ограниченность функциональности экспорта данных в документы Microsoft Office и то, что программа не подходит для управления данными, не связанными с оптовой торговлей.

Optik.exe представляет собой важный инструмент для торговых компаний, помогая им оставаться конкурентоспособными в современном бизнес-окружении.

# **Список использованных источников**

1. Багласова, Т.Г. Методические указания по оформлению курсовых проектов, дипломных проектов и отчетов для учащихся специальности 2-40 01 01 «Программное обеспечение технологий» / С.В. Банцевич, Т.Г. Багласова. – Минск: КБП, 2022. – 49 c.
2. Михалевич, В.Ю. Методические указания к курсовому проектированию для обучающихся специальности 2-40 01 01 «Программное обеспечение информационных технологий – Минск: КБП, 2023. – 3 c.
3. ГОСТ Р 54694-2011. Логистика. Термины и определения. – Введ. 2013-01-01. – М.: Стандартинформ, 2012.;
4. Текст программы. Требования к содержанию, оформлению и контролю качества: ГОСТ 19.401-2000. – Введ. 01.09.2001. – Минск : Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 16 с.
5. 9 лучших программ для автоматизации рабочих процессов [2024 г.] [Электронный ресурс] – Режим доступа: https://nauchniestati.ru/spravka;
6. Автоматизация рабочего места – RPA ROBIN [Электронный ресурс] – Режим доступа: https://www.atlassian.com/ru/agile;
7. Автоматизированное рабочее место (АРМ): что это? [Электронный ресурс] – Режим доступа: https://www.liga-commersantov.ru/articles;
8. Обязанности менеджера по работе с клиентами – чем занимается и как [Электронный ресурс] – Режим доступа: https://sales-generator.ru/blog;
9. Разработка автоматизированного рабочего места менеджера [Электронный ресурс] – Режим доступа: https://spravochnick.ru/informatika;
10. Топ лучших приложений для управления персоналом для автоматизации [Электронный ресурс] – Режим доступа: https://www.guru99.com/ru/best;
11. ТОП-26 программ мониторинга и контроля работы сотрудников за [Электронный ресурс] – Режим доступа: https://habr.com/ru/articles;
12. Удобство и эффективность: разработка автоматизированного рабочего места [Электронный ресурс] – Режим доступа: <https://appmaster.io/ru/blog>;
13. Что значит АРМ (Автоматизированное рабочее место)? [Электронный ресурс] – Режим доступа: https://alexrovich.ru/info;
14. [Visual Studio 2022 IDE - Programming Tool for Software Developers (microsoft.com)](https://visualstudio.microsoft.com/vs/) [Электронный ресурс] – Режим доступа: https://visualstudio.microsoft.com/vs/.
15. [Visual Studio 2022 IDE - Programming Tool for Software Developers (microsoft.com)](https://visualstudio.microsoft.com/vs/) [Электронный ресурс] – Режим доступа: https://visualstudio.microsoft.com/vs/.
16. C Sharp – Википедия [2024 г.] [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/C\_Sharp;
17. [Microsoft Office – Wikipedia](https://en.wikipedia.org/wiki/Microsoft_Office) [2024 г.] [Электронный ресурс] – Режим доступа: https://en.wikipedia.org/wiki/Microsoft\_Office;
18. [Microsoft Visio – Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Microsoft_Visio) 2024 г.] [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Microsoft_Visio>;
19. Windows 11 [Электронный ресурс]. – Microsoft, 2024. – Режим доступа: https://support.microsoft.com/ru-ru/meetwindows11. – Дата доступа: 16.03.2024;

# **Приложение А**

**(обязательное)**

**Текст программных модулей**

private void \_\_\_button1\_\_Click(object sender, RoutedEventArgs e)

{

Random rand = new Random();

try

{

OrderClass order = new OrderClass

{

Number = rand.Next(),

Client = \_\_\_textbox1\_.Text,

Price = Convert.ToInt32(\_\_\_textbox2\_.Text),

PhoneNumber = \_\_\_textbox3\_.Text,

Date = DateTime.Now

};

Main2 main2 = (Main2)Application.Current.MainWindow;

main2.AddOrder(order);

CreateOrderDocument(order);

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

private void CreateOrderDocument(OrderClass order)

{

string fileName = $"Заказ\_{order.Number}.docx";

var doc = DocX.Create(fileName);

var title = doc.InsertParagraph("Счет-фактура")

.FontSize(24)

.Bold()

.SpacingAfter(10)

.Alignment = Alignment.center;

doc.InsertParagraph($"Номер заказа: {order.Number}")

.FontSize(16)

.SpacingAfter(5);

doc.InsertParagraph($"Клиент: {order.Client}")

.FontSize(16)

.SpacingAfter(5);

doc.InsertParagraph($"Цена: {order.Price} руб.")

.FontSize(16)

.SpacingAfter(5);

doc.InsertParagraph($"Номер телефона: {order.PhoneNumber}")

.FontSize(16)

.SpacingAfter(5);

doc.InsertParagraph($"Дата: {order.Date}")

.FontSize(16)

.SpacingAfter(10);

var table = doc.AddTable(6, 2);

table.Alignment = Alignment.center;

table.Design = TableDesign.LightShadingAccent1;

table.Rows[0].Cells[0].Paragraphs[0].Append("Поле").Bold().FontSize(16);

table.Rows[0].Cells[1].Paragraphs[0].Append("Значение").Bold().FontSize(16);

table.Rows[1].Cells[0].Paragraphs[0].Append("Номер заказа").FontSize(14);

table.Rows[1].Cells[1].Paragraphs[0].Append(order.Number.ToString()).FontSize(14);

table.Rows[2].Cells[0].Paragraphs[0].Append("Клиент").FontSize(14);

table.Rows[2].Cells[1].Paragraphs[0].Append(order.Client).FontSize(14);

table.Rows[3].Cells[0].Paragraphs[0].Append("Цена").FontSize(14);

table.Rows[3].Cells[1].Paragraphs[0].Append($"{order.Price} руб.").FontSize(14);

table.Rows[4].Cells[0].Paragraphs[0].Append("Номер телефона").FontSize(14);

table.Rows[4].Cells[1].Paragraphs[0].Append(order.PhoneNumber).FontSize(14);

table.Rows[5].Cells[0].Paragraphs[0].Append("Дата").FontSize(14);

table.Rows[5].Cells[1].Paragraphs[0].Append(order.Date.ToString()).FontSize(14);

doc.InsertParagraph().InsertTableAfterSelf(table);

doc.InsertParagraph("Дополнительная информация:")

.FontSize(18)

.Bold()

.SpacingAfter(8);

doc.InsertParagraph("Доставка:")

.FontSize(16)

.Bold()

.SpacingAfter(5);

doc.InsertParagraph("Адрес доставки: ул. Примерная, д. 1, кв. 1")

.FontSize(14)

.SpacingAfter(5);

doc.InsertParagraph("Дата доставки: " + order.Date.AddDays(3).ToString("dd.MM.yyyy"))

.FontSize(14)

.SpacingAfter(5);

doc.InsertParagraph("Комментарий к заказу: Пожалуйста, звоните за 30 минут до доставки.")

.FontSize(14)

.SpacingAfter(10);

doc.InsertParagraph("Оплата:")

.FontSize(16)

.Bold()

.SpacingAfter(5);

doc.InsertParagraph("Метод оплаты: Наличные")

.FontSize(14)

.SpacingAfter(5);

doc.InsertParagraph("Сдача с: 5000 руб.")

.FontSize(14)

.SpacingAfter(10);

doc.InsertParagraph("Дополнительные условия:")

.FontSize(16)

.Bold()

.SpacingAfter(5);

doc.InsertParagraph("При возникновении вопросов, пожалуйста, свяжитесь с нашим оператором по номеру +375 (29) 728-57-25.")

.FontSize(14)

.SpacingAfter(10);

doc.InsertParagraph()

.AppendLine(new string(' ', 3000))

.SpacingAfter(5);

doc.Save();

MessageBox.Show($"Документ заказа создан: {fileName}");

}

public void SaveOrder(OrderClass order)

{

try

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string query = "INSERT INTO Table\_1 (Number, Client, Price, PhoneNumber, Date) VALUES (@Number, @Client, @Price, @PhoneNumber, @Date)";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@Number", order.Number);

command.Parameters.AddWithValue("@Client", order.Client);

command.Parameters.AddWithValue("@Price", order.Price);

command.Parameters.AddWithValue("@PhoneNumber", order.PhoneNumber);

command.Parameters.AddWithValue("@Date", order.Date);

command.ExecuteNonQuery();

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

# **Приложение Б**

**(справочное)**

**Формы выходных документов**

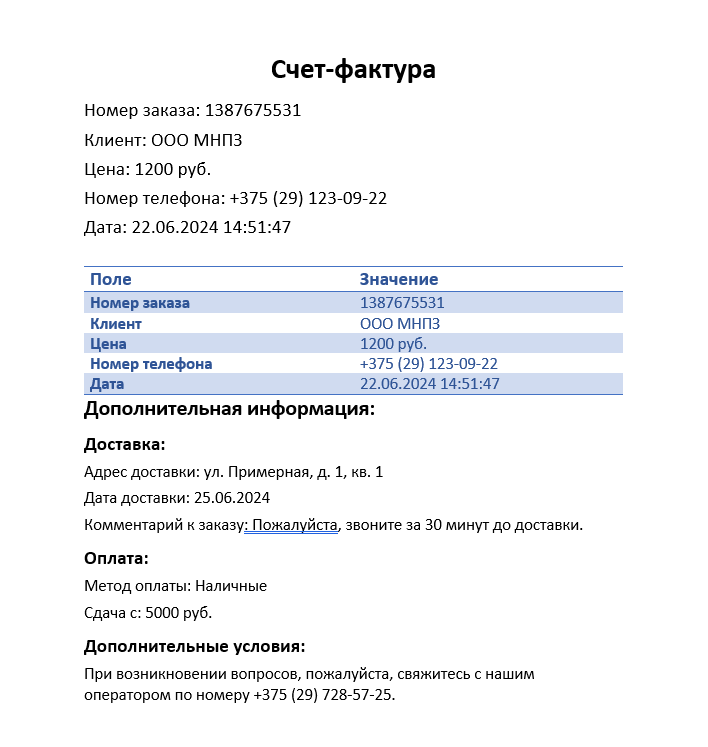


Рисунок Б.1 – Счет-фактура

# **Приложение В**

**(справочное)**

**Результаты работы программы**

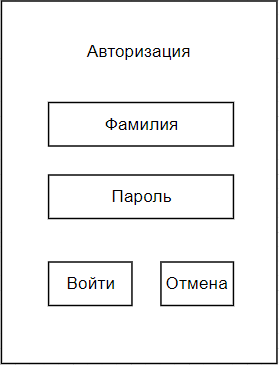


Рисунок В.1 – Форма входа

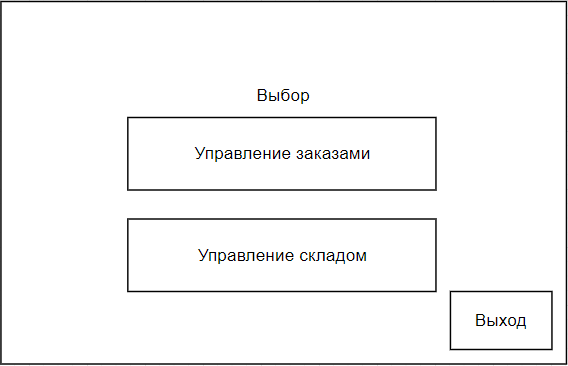


Рисунок В.2 – Форма выбора рабочего места

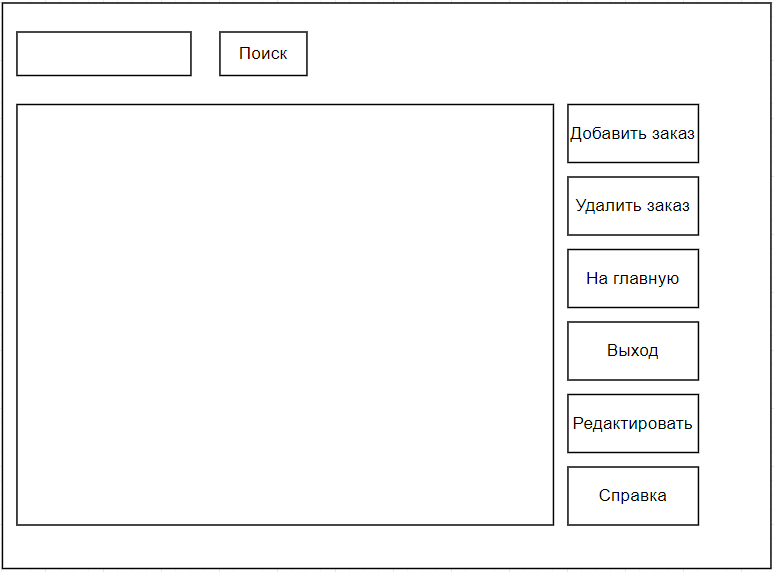


Рисунок В.3 – Форма главная для управления заказами

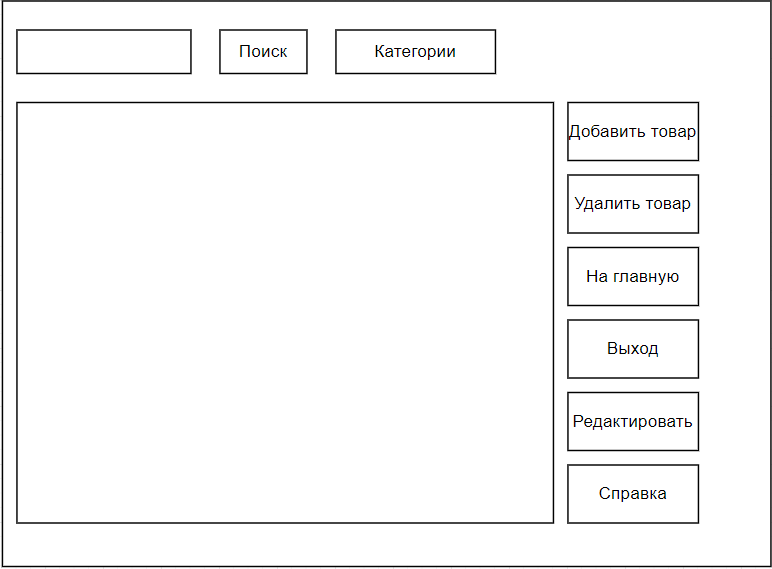


Рисунок В.4 – Форма главная для управления складом

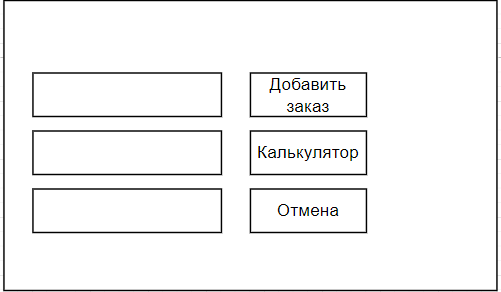


Рисунок В.5– Форма добавления заказа



Рисунок В.6– Форма добавления товара

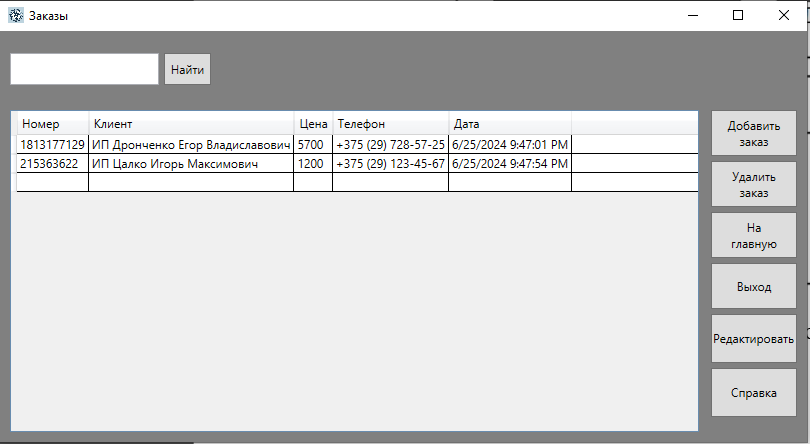


Рисунок В.7 – Результат входа менеджера управления заказами

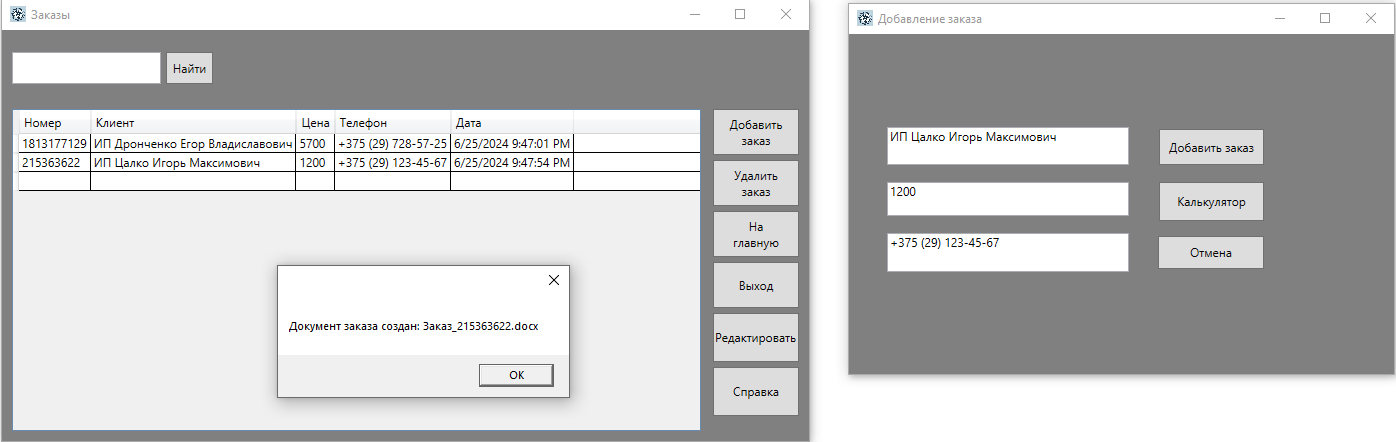


Рисунок В.8 – Результат добавления заказа

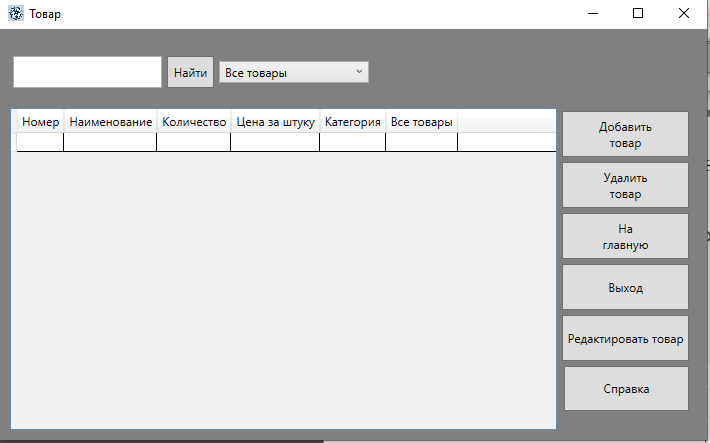


Рисунок В.9 – Результат входа работника управления складом

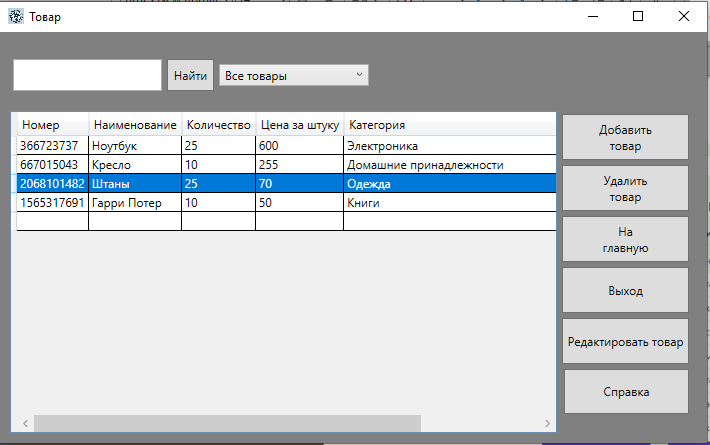


Рисунок В.10 – Результат добавления товара

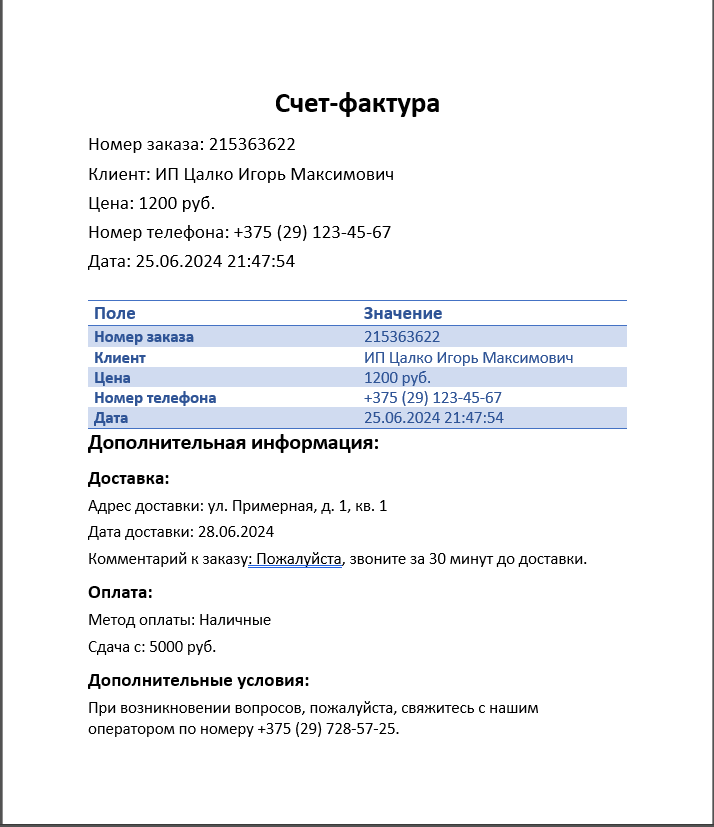


Рисунок В.11 – Результат создания счет-фактуры