



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ  
**«КОЛЛЕДЖ СВЯЗИ №54»**  
ИМЕНИ П.М. ВОСТРУХИНА

(наименование образовательного учреждения)

**ДОПУСКАЮ К ЗАЩИТЕ**

Заместитель директора по  
УПР *О.В.Корешков*

(дата)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Разработка автоматизированной информационной системы учета товаров  
складского помещения

(тема)

Выпускная квалификационная работа должна быть выполнена в виде:  
дипломной работы и демонстрационного экзамена

студентом группы 4ИСП9-4

(номер группы)

Дианой Илшатовой Абдулкадировой

(И. О. Фамилия)

(подпись, дата)

Основная профессиональная образовательная программа по специальности  
09.02.07 Информационные системы и программирование (квалификация –  
администратор баз данных)

(шифр и наименование специальности)

Форма обучения очная

Руководитель

преподаватель

Анастасия Николаевна Виеру

(ученая степень, должность, И. О.

Фамилия)

(подпись, дата)

Руководитель мастер п. о.

(ученая степень, должность, И. О. Фамилия)

Анастасия Николаевна Виеру

(подпись, дата)

Председатель предметной(междисциплинарной, модульной) комиссии  
Кирилл Михайлович Бастрыкин

(И. О. Фамилия)

(подпись, дата)

Москва  
2022



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ  
«КОЛЛЕДЖ СВЯЗИ №54»  
ИМЕНИ П.М. ВОСТРУХИНА

полное название образовательного учреждения

УТВЕРЖДАЮ

Заместитель директора по УПР О.В. Корешков  
13 апреля 2022 года  
(дата)

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

Студенту Абдулкадировой Диане Илшатовне  
(фамилия, имя, отчество полностью)

I. Тема выпускной квалификационной работы Разработка автоматизированной информационной системы учета товаров складского помещения

II. Срок сдачи студентом законченной работы 07 июня 2022 г.

III. Исходные данные Программные средства: Microsoft Office Visio 2019, СУБД MS SQL Server 2022, среда разработки Visual Studio 2022

IV. Перечень подлежащих разработке вопросов

- Анализ особенностей предметной области
- Сравнительный анализ программных средств  
Проектирование автоматизированной информационной системы учета товаров складского помещения (создание Use Case диаграммы, ER диаграммы, прототипа настольного приложения)
- Разработка автоматизированной информационной системы учета товаров складского помещения (создание базы данных в СУБД MS SQL Server, настольного приложения на языке программирования C# в Visual Studio)

V. Перечень графического/иллюстрационного материала

- Use Case диаграмма
- ER диаграмма базы данных, разработанная в Microsoft Office Visio
- Диаграмма базы данных, разработанная в СУБД MS SQL Server
- Скриншоты экрана разработанного настольного приложений
- Презентация

VI. Дата выдачи задания «13» апреля 2022 г.

Руководитель

Виеру А.Н.  
(подпись)

Задание принял к исполнению

Абдулкадирова Д.И.  
(подпись)

«13» апреля 2022 г.



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ  
**«КОЛЛЕДЖ СВЯЗИ №54»**  
ИМЕНИ П.М. ВОСТРУХИНА

КАЛЕНДАРНЫЙ ПЛАН ВКР  
(с указанием сроков выполнения отдельных этапов)

для специальности 09.02.07 Информационные системы и программирование  
(группа 4ИСП9-4)

<b>Утверждение темы и руководителя ВКР</b>		25.03.2022
<b>Выдача задания на ВКР</b>		с 26.03.2022 по 13.04.2022
<b>Сроки преддипломной практики</b>		4 недели с 13.04.22 по 10.05.22
	Выполнение задания по теме ВКР	с 13.04.22 по 05.05.22
	Предоставление отчета по практике руководителю	06.05.22
	Аттестация по практике	07.05.22
<b>Подготовка ВКР</b>		4 недели с 11.05.22 по 14.06.22
	Подбор и анализ исходной информации	1 неделя
	Подготовка и утверждение плана (оглавления) ВКР	с 11.05.22 по 17.05.22
	Работа над разделами (главами) и устранение замечаний руководителя ВКР	1 неделя с 18.05.22 по 24.05.22
	Согласование содержания ВКР, устранение замечаний	1 неделя с 25.05.22 по 31.05.22
	<b>Оформление и представление руководителю полного текста работы. Получение отзыва руководителя ВКР.</b>	1 неделя с 01.06.22 по 07.06.22
	<b>Предоставление студентом готовой ВКР рецензенту</b>	
	<b>Сдача ДЭ</b>	Согласно отдельному графику
	<b>Предзащита ВКР</b>	08.06.22-14.06.22

Руководитель \_\_\_\_\_ **А. Н. Виеру**

План принял к исполнению  
«13» апреля 2022 г.

\_\_\_\_\_ **Д. И. Абдулкадирова**

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	5
ГЛАВА I. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПРИМЕНЯЕМЫХ ПРОГРАММНЫХ СРЕДСТВ .....	7
1.1 Информационная система. Создание, применение, свойства, классификация .....	7
1.2 Анализ программных и технических средств реализации информационной системы.....	8
1.2.1 Анализ и выбор СУБД .....	8
1.2.2 Анализ и выбор языка программирования .....	10
Выводы по главе I .....	12
ГЛАВА II. РАЗРАБОТКА И ВНЕДРЕНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА ТОВАРОВ СКЛАДСКОГО ПОМЕЩЕНИЯ .....	13
2.1. Создание USE CASE диаграммы.....	13
2.2 Проектирование базы данных. Создание ER-диаграммы.....	14
2.3 Разработка базы данных в СУБД MSSQL Server.....	16
2.4 Разработка функционала настольного приложения .....	23
Выводы по главе II .....	27
ЗАКЛЮЧЕНИЕ .....	29
СПИСОК ЛИТЕРАТУРЫ.....	30
ПРИЛОЖЕНИЕ 1. USE-CASE ДИАГРАММА	
ПРИЛОЖЕНИЕ 2. ER-MODEL	
ПРИЛОЖЕНИЕ 3. ФИЗИЧЕСКАЯ МОДЕЛЬ	
ПРИЛОЖЕНИЕ 4. ЛИСТИНГ КОДА ОСНОВНЫХ ФОРМ РАЗРАБОТАННОГО ПОЛЬЗОВАТЕЛЬСКОГО ПРИЛОЖЕНИЯ	

					09.02.07 – 4 ИСП 9-4		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Д.И.Абдулкадирова			«Разработка автоматизированной информационной системы учета товаров складского помещения»	Лит.	Лист
Провер.		А.Н.Виеру					4
Рецензент						ГБПОУ КС №54 им. П.М. Вострухина	
Н. Контр.							
Утверд.							
						Листов	32

## ВВЕДЕНИЕ

В современных требованиях, предъявляемых к качеству работы торговых розничных предприятий, отмечается, что эффективная работа сотрудников таких предприятий полностью зависит от оснащения их информационными средствами и их успешного использования в системе автоматизированного учёта товаров.

Компьютерный учёт товара полностью отличается от классического, рукописного. Компьютерные программы упрощают учёт товаров, сокращают время, требуемое на оформление документов для анализа торговой деятельности, следовательно, при применении компьютерных программ, повышается эффективность работы персонала торгового предприятия, уменьшается время обучения персонала.

Результаты выполнения торговых операций заносятся в соответствующие журналы, что позволяет автоматически их сохранять и использовать в дальнейшем.

Основные преимущества автоматизации учёта: экономия времени, сжатие хранимых данных с экономией объёма памяти и снижение затрат на операции обновления данных. Всё это будет сопутствовать более качественному обслуживанию, повышению результативности работы предприятия, повышению точности учёта и снижению потерь товара.

Актуальность темы данной выпускной квалификационной работы связана с постоянным развитием информационных технологий и необходимостью автоматизирования процесса учёта товаров в организации. Применение интерфейса для базы данных позволит максимально упростить и оптимизировать труд уполномоченных сотрудников при работе с данными.

Целью выпускной квалификационной работы является систематизация и закрепление полученных теоретических знаний и практических умений, проектирование и разработка информационной системы учёта товаров, которая позволит повысить производительность труда персонала организации, в виде

программного приложения.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- изучение особенностей конкретной предметной области, относящихся к теме выпускной квалификационной работы;
- анализ программных средств с обоснованием выбора;
- проектирование и разработка базы данных;
- разработка интерфейса для созданной базы данных;
- анализ полученных результатов работы разработанного программного обеспечения.

Объектом исследования выступает автоматизированная информационная система учета товаров складского помещения.

Предметом исследования является разработка базы данных и удобного интерфейса для автоматизации учета товаров складского помещения.

По результатам выполненной работы готовое программное средство будет иметь данный функционал:

- авторизация потенциальных пользователей компании в системе;
- разграничение доступа;
- возможность редактировать, изменять, добавлять данные о клиентах и продуктах;
- автоматизированное ведение отчетности;
- возможность быстрого просмотра данных.

Информационной базой выпускной квалификационной работы является научная и техническая литература, электронные источники.

Данная выпускная квалификационная работа выполнена на 32 страницах машинописного текста, состоит из введения, двух глав, заключения, списка использованной литературы и приложений.

# ГЛАВА I. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПРИМЕНЯЕМЫХ ПРОГРАММНЫХ СРЕДСТВ

## 1.1 Информационная система. Создание, применение, свойства, классификация

В рамках данной выпускной квалификационной работы рассматривается предметная область учёта товаров складского помещения. В первую очередь, успешность данного проекта зависит от умелого распределения ресурсов. Следовательно, вопросы продуманного учёта товаров, должны занимать одно из ключевых мест в общей стратегии развития предприятия. Важная роль в оптимизации управления любого предприятия принадлежит автоматизированным системам. В своей деятельности компании складского помещения вынуждены следить за состоянием имеющихся товаров, тщательно проверять соблюдение компаниями условий предоставления аренды, вести учёт заключаемых сделок с компаниями. Автоматизированная система должна предоставлять всем заинтересованным лицам (администраторам, работникам) мгновенный доступ к необходимой информации.

Для авторизации в сервисе работнику достаточно ввести логин или пароль. В случае, если директор компании решит воспользоваться услугами сервиса, он должен будет предоставить сотрудникам предприятия достоверную информацию личную и компании, а именно: номер телефона, корректные ФИО и название компании, инн, огрн, регистрационный номер. В случае, если предоставленные данные соответствуют требованиям сервиса, пользователь получает возможность заключения договора об аренде места хранения продукции. После предоставления данных сервису клиент имеет право добавлять информацию о его работниках, о поставляемой продукции. Также он может отслеживать итоговый чек, его работников, поставляемой продукции. Далее сотрудники оформляют договор аренды - соглашение между сервисом и клиентом, в соответствии с которым клиент обязуется следовать условия сервиса при аренде, а сервис, в свою очередь, обязуется в полной мере предоставить

клиенту оплачиваемые услуги.

С другой стороны, помимо клиента, данным приложением может пользоваться более привилегированный пользователь - администратор.

Основными задачами администратора являются:

- добавление клиента компании;
- добавление тарифов;
- добавление данных о новых компаниях.

Администратор не может удалять продукцию из каталога, удалять учетные записи, а также не имеет права редактировать собственную информацию.

Последней категорией пользователей является директор склада. Директор склада обладает всеми возможными правами для добавления информации:

- добавлять руководите компании;
- добавлять информация о компании;
- просмотр тарифов.

## 1.2 Анализ программных и технических средств реализации информационной системы

### 1.2.1 Анализ и выбор СУБД

Для начала разработки необходимо выбрать СУБД для хранения и работы с данными и язык программирования.

Современные СУБД в основном являются приложениями Windows, так как данная среда позволяет более полно использовать возможности персональной ЭВМ, нежели среда DOS. Среди наиболее ярких представителей систем управления базами данных можно отметить PostgreSQL, Microsoft SQL Server и Oracle, используемые в приложениях, построенных по технологии «клиент-сервер». Фактически, у любой современной СУБД существует аналог, выпускаемый другой компанией, имеющий аналогичную область применения и возможности, любое приложение способно работать со многими форматами представления данных, осуществлять экспорт и импорт данных благодаря



наличию большого числа конвертеров. Общепринятыми, также, являются технологии, позволяющие использовать возможности других приложений, например, текстовых процессоров, пакетов построения графиков и т.п., и встроенные версии языков высокого уровня и средства визуального программирования интерфейсов разрабатываемых приложений. При выборе СУБД основными факторами являются её гибкость, совместимость с выбранной средой разработки и языком программирования.

Поскольку MS SQL Server и C# разработаны одной компанией и имеют высокую совместимость и простоту совместного использования, а также необходимую гибкость – выбор пал на СУБД Microsoft SQL Server.

Основные преимущества СУБД MS SQL Server:

1. оптимизированный многопользовательский режим работы с развитой системой транзакционной обработки, что обеспечивает многочисленным пользователям возможность работы с базой данных, не мешая друг другу. Это – одно из важнейших требований, выдвигаемых сегодня к информационной системе;
2. надежные механизмы защиты информации (учитывая стандартную трехзвенную архитектуру защиты: на уровне сети - на уровне сервера БД - на уровне клиентской ОС);
3. эффективные инструменты для разграничения доступа к БД;
4. поддержка широкого диапазона аппаратно-программных платформ;
5. реализация распределенной обработки данных;
6. возможность построения гетерогенных и распределенных сетей;
7. развитые средства управления, администрирования, контроля и мониторинга сервера БД;
8. поддержка таких эффективных механизмов, как: триггеры, хранимые процедуры и функции, пакеты, словари данных и т.п.

Все вышеперечисленное обусловило широкое распространение решений на базе данной СУБД.

### 1.2.2 Анализ и выбор языка программирования

Далее необходимо выбрать язык программирования для создания настольного приложения.

Существует достаточно широкий спектр языков программирования для создания информационных систем. Авторы С# стремились создать язык, сочетающий простоту и выразительность современных объектно-ориентированных языков с богатством возможностей и мощностью С++.

С# позаимствовал большинство своих синтаксических конструкций из С++. В частности, в нем присутствуют такие удобные типы данных, как структуры и перечисления.

Например, в С#, определение класса состоит из одного файла (\*.cs), в отличие от С++, где определение класса разбито на заголовок (\*.h) и реализацию (\*.cpp). Как С#, так и Java основаны на синтаксических конструкциях С++. Если Java во многих отношениях можно назвать «очищенной» версией С++, то С# можно охарактеризовать как «очищенную» версию Java.

Также, в С#, используются свойства классов. Как и в С++, С# позволяет производить перегрузку операторов для созданных вами типов). Отличием между Java и С# является то, что они созданы для разных типов вычислительных сред. С# - это фактически гибрид разных языков. При этом С# синтаксически не менее чист, чем Java, прост, как Visual Basic, и обладает практически той же мощностью и гибкостью, что и С++.

Основные преимущества С#:

1. полный и хорошо определенный набор основных типов;
  2. полная поддержка классов и объектно-ориентированного программирования;
  3. возможность перегружать операторы, унаследованные от С++.
- При этом значительная часть возникавших при этом сложностей ликвидирована;

4. встроенные синтаксические конструкции для работы с перечислениями, структурами и свойствами классов;

5. кроссплатформенность. Доступны компиляторы для большого количества платформ, на языке C++ разрабатывают программы для самых различных платформ и систем.

Visual Studio – полнофункциональная интегрированная среда разработки (IDE) с поддержкой популярных языков программирования, среди которых C, C++, VB.NET, C#, F#, JavaScript, Python. Функциональность Visual Studio охватывает все этапы разработки программного обеспечения, предоставляя современные инструменты для написания кода, проектирования графических интерфейсов, сборки, отладки и тестирования приложений. Возможности Visual Studio могут быть дополнены путем подключения необходимых расширений.

Редактор кода Visual Studio поддерживает подсветку синтаксиса, вставку фрагментов кода, отображение структуры и связанных функций. Существенно ускорить работу помогает технология IntelliSense – автозавершение кода по мере ввода.

Основные преимущества Visual Studio:

1. встроенный Web-сервер – для обслуживания Web-приложения ASP.NET необходим Web-сервер, который будет ожидать Web-запросы и обрабатывать соответствующие страницы;

2. поддержка множества языков при разработке – Visual Studio позволяет писать код на своем языке или любых других предпочитаемых языках, используя все время один и тот же интерфейс (IDE);

3. меньше кода для написания – для создания большинства приложений требуется приличное количество стандартного стереотипного кода, и Web-страницы ASP. NET тому не исключение;

4. интуитивный стиль кодирования – по умолчанию Visual Studio форматирует код по мере его ввода, автоматически вставляя необходимые отступы и применяя цветовое кодирование для выделения элементов типа комментариев;

5. более высокая скорость разработки – многие из функциональных возможностей Visual Studio направлены на то, чтобы помогать разработчику делать свою работу как можно быстрее;

6. возможности отладки – предлагаемые в Visual Studio инструменты отладки являются наилучшим средством для отслеживания загадочных ошибок и диагностирования странного поведения.

#### Выводы по главе I

В ходе изучения теоретического материала и анализа предметной области были рассмотрены аналогичные проекты, а также средства и технологии разработки. В результате было решено реализовывать базу данных в СУБД MS SQL Server, а для создания интерфейса пользователя использовать язык программирования C#, написанный специально для среды разработки Visual Studio.

Программная среда разработки интуитивно понятна и обладает технологиями, сокращающими время на написание программного кода и улучшающими его качество, использование технологии WPF обеспечит гибкость построения графического интерфейса и оптимизацию производительности при помощи аппаратного ускорения, а язык программирования C# позволит обеспечить наилучшую логику взаимодействия элементов приложения.

## ГЛАВА II. РАЗРАБОТКА И ВНЕДРЕНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА ТОВАРОВ СКЛАДСКОГО ПОМЕЩЕНИЯ

### 2.1. Создание USE CASE диаграммы

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур. UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода. Для достижения этих целей вначале строится модель в форме так называемой диаграммы вариантов использования (use case diagram), которая описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является концептуальной моделью системы в процессе ее проектирования и разработки.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей, взаимодействующих с системой с помощью так называемых вариантов использования. При этом действующим лицом называется любая сущность, взаимодействующая с системой. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет участнику диаграммы. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с

пользователем. В данной диаграмме 4 субъекта.

Субъект «Работник склада» должен иметь возможность авторизации, добавление информации по заполнению контейнеров определенной продукцией.

Субъект «Директор компании» должен иметь возможность авторизации, добавление работников компании, продукции, просмотр занимаемых контейнеров, чеков, работников компании.

Субъект «Начальник склада» имеет возможность добавлять новые компании, просматривать информацию по контейнерам и отчеты по использованию складского помещения.

Субъект «Администратор» должен иметь возможность добавлять новых пользователей, и редактировать, или создавать новые тарифы пользования.

Возможности всех четырех субъектов отображены на USE CASE диаграмме, которая была разработана в Microsoft Office Visio 2019, и представлена в Приложении 1.

## 2.2 Проектирование базы данных. Создание ER-диаграммы

Для анализа структуры данных и информационного моделирования существует множество программных продуктов. Один из них – Microsoft Office Visio 2019, поддерживающий методологию IDEF1X (Integration DEFinition for Information Modeling). Он позволяет построить логическую модель данных, представляющую собой совокупность информационных объектов и связей между ними, а также физическую модель, непосредственно связанную с конкретной СУБД.

Физическая модель БД представляется в виде набора таблиц специальной формы, в которых указываются наименование атрибута, идентификатор, тип, длина, формат, ограничения.

Прежде чем строить физическую модель, необходимо решить, какая информация будет храниться в базе данных.

Модель «сущность-связь» (ER модель) модель данных, которая позволяет описывать концептуальные схемы предметной области.

Сущность – реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области, информация о котором подлежит хранению.

Экземпляр сущности – конкретный представитель данной сущности, уникально идентифицируемый объект.

Атрибут сущности – именованная характеристика, являющаяся некоторым свойством сущности. Наименование атрибута должно быть выражено существительным в единственном числе.

Ключ сущности – атрибут или набор атрибутов, однозначно идентифицируют экземпляр сущности. Ключ сущности должен быть минимально достаточным: в нем не должно быть атрибутов, удаление которых из ключа не отразится на его уникальности.

Чтобы система смогла выполнять свои функции база данных должна содержать следующие данные:

- данные об учётных записях пользователей;
- данные о секторах;
- данные о продуктах;
- информация о расписании занятий;

Для проверки правильности построенной логической модели она приводится к 3 нормальным формам:

- отношение находится в 1НФ, если все его атрибуты являются простыми (имеют единственное значение). Исходное отношение строится таким образом, чтобы оно было в 1НФ;
- отношение находится в 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут функционально полно зависит от первичного ключа (составного);
- отношение находится в 3НФ в том и только в том случае, если все не ключевые атрибуты отношения взаимно независимы и полностью зависят от первичного ключа.

Полноразмерная схема базы данных, разработанной в Microsoft Office

Visio 2019, представлена в Приложении 2.

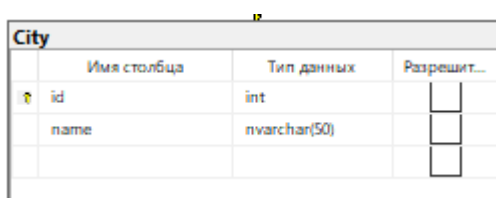
### 2.3 Разработка базы данных в СУБД MSSQL Server

Физическое проектирование – создание схемы базы данных для конкретной СУБД. Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д.

Результатом физического проектирования ранее созданной логической схемы является полноценная база данных, к которой в последующем будет разработан интерфейс.

Рассмотрим основные таблицы базы данных.

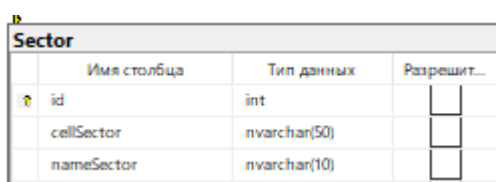
Необходимо создать таблицу «City» (см. рис. 2.1). Эта таблица необходима для хранения городов и последующего их использования в базе данных.



Имя столбца	Тип данных	Разрешит...
id	int	<input type="checkbox"/>
name	nvarchar(50)	<input type="checkbox"/>

Рисунок 2.1 Сущность City

Для использования таблицы «InfoContainer» (см. рис. 2.3), которая нам понадобится для хранения и использования информации по нахождению продуктов различных компаний, необходимо создать таблицу «Sector» (см. рис. 2.2), так как для хранения информации о контейнере, необходимо понимать, в каком секторе он находится.



Имя столбца	Тип данных	Разрешит...
id	int	<input type="checkbox"/>
cellSector	nvarchar(50)	<input type="checkbox"/>
nameSector	nvarchar(10)	<input type="checkbox"/>

Рисунок 2.2 Сущность Sector



InfoContainer			
	Имя столбца	Тип данных	Разрешит...
↑	id	int	<input type="checkbox"/>
	idContainer	int	<input type="checkbox"/>
	idProduct	int	<input type="checkbox"/>
	QuantityProduct	nvarchar(20)	<input checked="" type="checkbox"/>
	CargoWeight	nvarchar(20)	<input checked="" type="checkbox"/>
	dateStart	datetime	<input type="checkbox"/>
	available	bit	<input type="checkbox"/>
	title	nvarchar(50)	<input checked="" type="checkbox"/>

Рисунок 2.3 Сущность InfoContainer

Так же требуется таблица «Product» (см. рис. 2.4), для хранения информации о продукте.

Product			
	Имя столбца	Тип данных	Разрешит...
↑	id	int	<input type="checkbox"/>
	name	nvarchar(50)	<input type="checkbox"/>
	image	nvarchar(100)	<input checked="" type="checkbox"/>
	description	nvarchar(250)	<input checked="" type="checkbox"/>

Рисунок 2.4 Сущность Product

Для хранения информации по тарифам и их актуальности, необходимо создать таблицу «Tariff» (см. рис. 2.5)

Tariff			
	Имя столбца	Тип данных	Разрешит...
↑	id	int	<input type="checkbox"/>
	name	nvarchar(100)	<input type="checkbox"/>
	cost	decimal(18, 0)	<input type="checkbox"/>
	available	bit	<input type="checkbox"/>

Рисунок 2.5 Сущность Tariff

Таблица «Receipt» (см. рис. 2.6), создана для мониторинга информации по использовании услуг. В этой таблицы включена информация по тарифу, задолженности, по компании пользующей услугой склада и по контейнеру, который хранит в себе информацию по продукту и его нахождению в складе.

Receipt			
	Имя столбца	Тип данных	Разрешит...
↑	id	int	<input type="checkbox"/>
	infoContainer	int	<input type="checkbox"/>
	idCompany	int	<input type="checkbox"/>
	idTariff	int	<input type="checkbox"/>
	delay	bit	<input type="checkbox"/>

Рисунок 2.6 Сущность Receipt

Созданная таблица «Company» (см. рис. 2.7), используется для формирования отчетов и хранения информации по компаниям.

Company			
	Имя столбца	Тип данных	Разрешит...
🔑	id	int	<input type="checkbox"/>
	name	nvarchar(100)	<input type="checkbox"/>
	inn	nvarchar(20)	<input type="checkbox"/>
	ogrn	nvarchar(25)	<input type="checkbox"/>
	registrNum	nvarchar(25)	<input type="checkbox"/>
	discount	nvarchar(10)	<input checked="" type="checkbox"/>
	city	int	<input type="checkbox"/>

Рисунок 2.7 Сущность Company

Таблица «InfoUserCompany» (см. рис. 2.8), необходима для соотношения компаний и их сотрудников.

InfoUserCompany			
	Имя столбца	Тип данных	Разреш...
↑	id	int	<input type="checkbox"/>
	idCompany	int	<input type="checkbox"/>
	idUser	int	<input type="checkbox"/>

Рисунок 2.8 Сущность InfoUserCompany

Созданная таблица «User» (см. рис. 2.9), используется для формирования отчетов и хранения информации по работникам и клиентам системы.

User			
	Имя столбца	Тип данных	Разрешит...
↑	id	int	<input type="checkbox"/>
	login	nvarchar(50)	<input type="checkbox"/>
	password	nvarchar(50)	<input type="checkbox"/>
	fioContactFace	nvarchar(100)	<input type="checkbox"/>
	numberContactFace	nvarchar(28)	<input type="checkbox"/>
	emailContactFace	nvarchar(100)	<input type="checkbox"/>
	idRole	int	<input type="checkbox"/>

Рисунок 2.9 Сущность User

Созданная таблица «Role» (см. рис. 2.10), используется для хранения информации по привилегиям пользователей.

Role			
	Имя столбца	Тип данных	Разрешит...
id	id	int	<input type="checkbox"/>
	name	nvarchar(100)	<input checked="" type="checkbox"/>

Рисунок 2.10 Сущность Role

Таблица «Support» (см. рис. 2.11) используется для хранения обращений пользователей в компанию.

Support			
	Имя столбца	Тип данных	Разреш...
🔑	id	int	<input type="checkbox"/>
	email	nvarchar(50)	<input type="checkbox"/>
	question	nvarchar(150)	<input checked="" type="checkbox"/>
	appeal	nvarchar(500)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Рисунок 2.11 Сущность Support

Полноразмерная схема база данных, разработанной в СУБД MS SQL Server 2022, представлена в Приложении 3.

Также, для удобного отображения данных из базы, создано представление «Данные о сотрудниках», «Данные по компаниям», «Итоговый чек», «Данные по использованию контейнеров», «Данные по руководителям склада».

Данные по компаниям (см. рис. 2.12), содержат в себе полную информацию о всех зарегистрированных компании.

```

alter View Vm_DirComp
as
select top 100 m.fioContastFace as 'Директор компании',
c.[name] as 'Название компании',
m.numberContactFace as 'Номер телефона компании',
c.discount as 'Скидка компании',
t.[name] as 'Тариф компании'

from [User] m
left join InfoUserCompany i on m.id = i.idUser
left join Company c on c.id = i.idCompany
left join [Role] tm on m.idRole = tm.id
left join [Receipt] r on c.id = r.idCompany
left join [Receipt] on c.id = r.idTariff
left join Tariff t on t.id = r.idTariff

where (m.fioContastFace = m.fioContastFace and m.idRole = 2)

group by m.fioContastFace, c.[name], m.numberContactFace, c.discount, t.[name]

```

Рисунок 2.12 Код создания представления «Данные по компаниям» на T-SQL

Результат созданного представления «Данные по компаниям» приведен ниже (см. рис. 2.13)

	Директор компании	Название компании	Номер телефона компании	Скидка компании	Тариф компании
1	Гавриленкова Зинаида Тимофеевна	Тайм-1	+7 (985) 586-79-29	0	NULL
2	Добрыня Илья Никитич	NULL	+7 (943) 456-43-65	NULL	NULL
3	Ломоносов Евгений Николаевич	Радиовнимание	+7 (932) 651-44-97	0	NULL

Рисунок 2.13 Результат представления «Данные по компаниям»

Данные по работникам (см. рис. 2.14) содержат в себе полную информацию о всех зарегистрированных сотрудниках в определенных компаниях. Это

представление необходимо для верного вывода информации в приложении.

```

alter View Vm_User
as
select top 100 m.fioContastFace as 'ФИО Работника',
m.numberContactFace as 'Номер телефона работника',
m.emailContactFace as 'Почта работника',
c.[name] as 'Компания'

from [User] m
left join InfoUserCompany i on m.id = i.idUser
left join Company c on c.id = i.idCompany
left join [Role] tm on m.idRole = tm.id

where (m.fioContastFace = m.fioContastFace and m.idRole = 1)

group by m.fioContastFace, m.emailContactFace, m.numberContactFace, c.[name]

```

Рисунок 2.14 Код создания представления «Данные по работникам» на T-SQL

Результат созданного представления «Данные по работникам» приведен ниже (см. рис. 2.15)

	ФИО Работника	Номер телефона работника	Почта работника	Компания
34	Капылюшная Альбина Кирилловна	+7 (978) 897-31-19	albina19021996@gmail.com	NULL
35	Карапетян Василий Всеволодович	+7 (925) 643-33-53	vasiliy93@gmail.com	NULL
36	Карасева Оксана Адамовна	+7 (960) 367-25-14	oksana1974@yandex.ru	NULL
37	Качусова Елизавета Ильиновна	+7 (946) 612-23-12	elizaveta1990@gmail.com	NULL
38	Кидин Илья Данилович	+7 (913) 262-91-86	ilya1984@mail.ru	NULL
39	Ковшевников Кирилл Ефимович	+7 (957) 495-35-58	kirill1967@yandex.ru	NULL
40	Колганов Тарас Николаевич	+7 (957) 869-70-98	taras2635@mail.ru	NULL
41	Копорикова Милана Тимофеевна	+7 (910) 397-53-97	milana.koporikova@rambler.ru	NULL
42	Коржукова Жанна Прокловна	+7 (943) 519-39-67	janna.korjukova@ya.ru	NULL
43	Корзоватых Надежда Романовна	+7 (952) 100-21-73	nadejda22091996@rambler.ru	NULL
44	Косма Антонина Филипповна	+7 (987) 887-48-49	antonina1991@mail.ru	NULL
45	Краева Арина Тарасовна	+7 (973) 625-34-52	arina14041984@mail.ru	NULL
46	Краева Мария Максимовна	+7 (953) 582-50-26	mariya75@gmail.com	NULL
47	Крамник Инна Тимофеевна	+7 (923) 679-39-11	inna7837@hotmail.com	NULL
48	Куприянова Людмила Марковна	+7 (920) 571-45-26	lyudmila.kupriyanova@rambler.ru	NULL

Рисунок 2.15 Результат представления «Данные по работникам» на T-SQL

Созданное представление итоговый чек (см. рис. 2.16), служит для понятного, логически, вывода информации из таблиц. Оно создано для вывода информации по использованию услуг складского помещения определенных компаний, по их программе лояльности, и по хранению их продукции на складе.

```

alter View Vm_TotalReceipt
as
select top 100
c.[name] as 'Название контейнера',
comp.[name] as 'Название компании',
t.[name] as 'Тариф',
comp.discount as 'Скидка компании',
r.[delay] as 'Задолженность'

from InfoContainer ic
join Container c on ic.idContainer = c.id
join Receipt r on c.id = r.infoContainer
join Company comp on r.idCompany = comp.id
join Tariff t on t.id = r.idTariff

group by c.[name], comp.[name], t.[name], comp.discount, r.[delay]

```

Рисунок 2.16 Код создания представления «Итоговый чек» на T-SQL

Результат созданного представления «Итоговый чек» приведен ниже (см. рис. 2.17)

	Название контейнера	Название компании	Тариф	Скидка компании	Задолженность
1	Контейнер 1	Джонсон Контролс	Бытовая техника, зоотовары, спорттовары	0	1
2	Контейнер 1	ТД Вольт	стандартный	0	1
3	Контейнер 2	Спектр РС	Косметическая продукция, медтехника, спортпитани...	0	1
4	Контейнер 3	Баттерика	стандартный	0	1
5	Контейнер 4	ЭнерджиВинд	стандартный	0	1
6	Контейнер 5	Мировые смазочные материалы	Косметическая продукция, медтехника, спортпитани...	0	1
7	Контейнер 6	Суперфонарик	Продукты питания	0	1
8	Контейнер 7	Реал Таерс	Бытовая техника, зоотовары, спорттовары	0	1

Рисунок 2.17 Результат представления «Итоговый чек» на T-SQL

Созданное представление «Информация по контейнерам» (см. рис. 2.18) необходимо для мониторинга данных по использованию контейнеров. Также для соотношения продуктов и контейнеров, в которых они находятся и для просмотра доступности контейнеров, для последующего его использования.

```

alter View Vm_InfoContainer
as
select top 100
c.[name] as 'Название контейнера',
p.[name] as 'Название продукта',
ic.available as 'Доступность'

from InfoContainer ic
join Container c on ic.idContainer = c.id
join Product p on ic.idProduct = p.id

group by c.[name], p.[name], ic.available

```

Рисунок 2.18 Код создания представления «Информация по контейнерам» на T-SQL

Результат созданного представления «Информация по контейнерам» приведен ниже (см. рис. 2.19)

	Название контейнера	Название продукта	Доступность
1	Контейнер 1	Гранулы зеленый 0x3	1
2	Контейнер 1	Рулон зеленый 0x0	0
3	Контейнер 2	Рулон зеленый 0x0	1
4	Контейнер 3	Нарезка серый 1x1	1
5	Контейнер 4	Рулон серый 1x0	1
6	Контейнер 5	Нарезка белый 1x2	1
7	Контейнер 6	Нарезка зеленый 3x0	1
8	Контейнер 7	Спрессованный пак синий 0x2	1
9	Контейнер 8	Спрессованный пак зеленый 2x0	1

Рисунок 2.19 Результат представления «Информация по контейнерам» на T-SQL

Представление «Данные по руководителям» (см. рис. 2.20) создано для мониторинга определенных категорий в последующем этапе .

```
alter View Vm_DirGoods
as
select top 100 m.fioContastFace as 'Директор склада',
c.[name] as 'Название компании',
m.numberContactFace as 'Номер телефона компании',
m.emailContactFace as 'Почта директора'
from [User] m
left join InfoUserCompany i on m.id = i.idUser
left join Company c on c.id = i.idCompany
left join [Role] tm on m.idRole = tm.id
left join [Receipt] r on c.id = r.idCompany
left join [Receipt] on c.id = r.idTariff
left join Tariff t on t.id = r.idTariff

where (m.fioContastFace = m.fioContastFace and m.idRole = 3)

group by m.fioContastFace, c.[name], m.numberContactFace, c.discount, t.[name], m.emailContactFace
```

Рисунок 2.20 Результат представления «Данные по руководителям склада» на T-SQL

Результат созданного представления «Данные по руководителям склада» приведен ниже (см. рис. 2.21)

	Директор склада	Название компании	Номер телефона компании	Почта директора
1	Бореева Надежда Михайловна	ТД Вольт	+7 (990) 811-24-97	nadejda54@ya.ru
2	Пичушкина Римма Емельяновна	ТД Вольт	+7 (918) 592-37-27	rimma50@outlook.com

Рисунок 2.21 Код создания представления «Данные по руководителям склада» на T-SQL

Представление — это объект базы данных, являющийся результатом выполнения запроса к базе данных, определенного с помощью оператора SELECT, в момент обращения к представлению. Представления дают возможность гибкой настройки прав доступа к данным, за счет того, что права даются на представление. Также оно было нам необходимо для разделения логики хранения данных и программного обеспечения. Для этого были реализованы представления выше для приложения учета товаров складского помещения.

## 2.4 Разработка функционала настольного приложения

Пользовательский интерфейс для базы данных будет выполнен с использованием платформы Windows Presentation Foundation (WPF).

WPF - 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 декларативного описания интерфейса XAML.

Первое, что увидит пользователь при запуске приложения – окно авторизации (см. рис. 2.22). После определения цветовой палитры, шрифтов и стиля, верстаем разметку для данного окна.

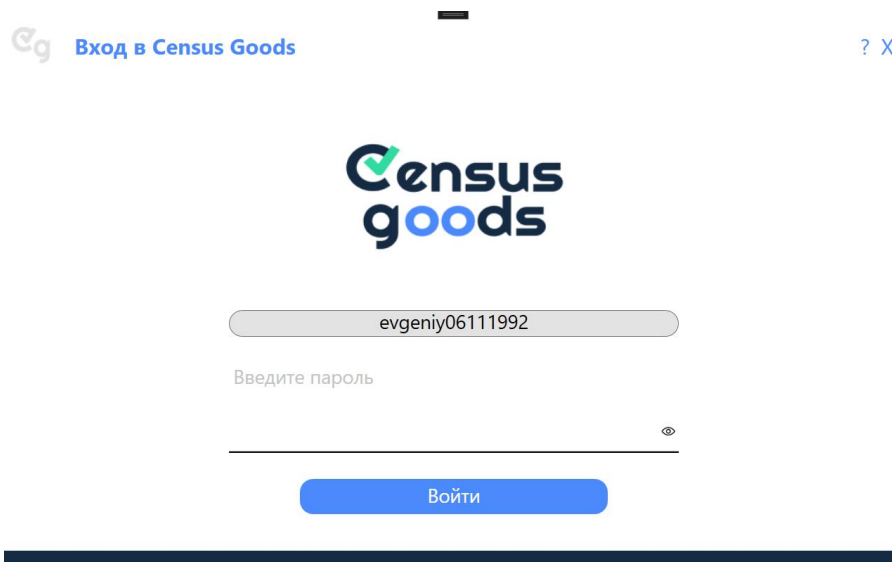


Рисунок 2.22 Визуализация окна «Авторизация»

В случае, если пользователь уже зарегистрирован в сервисе, он имеет возможность авторизоваться в приложении при помощи логина и пароля. Если у пользователя нет аккаунта, он должен получить у своего руководителя компании персональный логин и пароль от аккаунта.

Авторизация производится посредством поиска в таблице «Пользователь» совпадающего логина и соответствующего ему пароля. В случае отсутствия совпадения пользователю выводится информационный диалог (см. рис. 2.23):

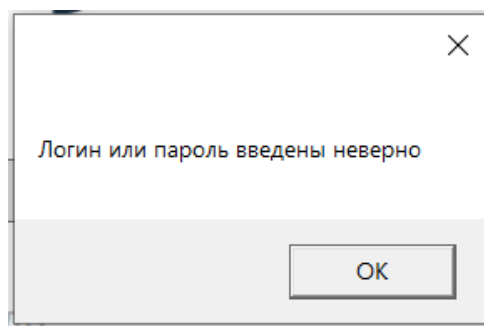


Рисунок 2.23 Оповещение о неверно введенных данных

В случае удачной авторизации, в зависимости от уровня привилегий пользователя открывается форма (см. рис. 2.24).

В случае авторизации в качестве руководителя склада, пользователь видит страницу приложения, на которой выводится список его работников.

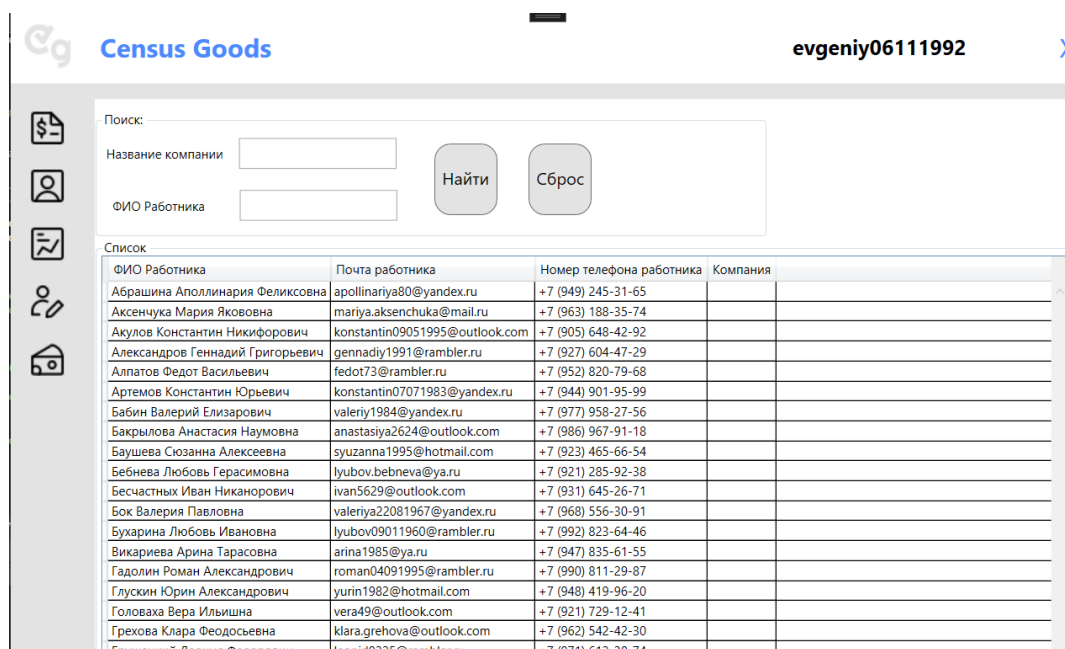


Рисунок 2.24 Пример главной страницы

На экране навигации приложения, пользователь может выбрать функцию добавления или просмотра при помощи кнопок.



Рисунок 2.25 Пример окна добавления

Если данные в каком-либо поле введены неверно, выведутся сообщение о проблеме (см. рис. 2.26).

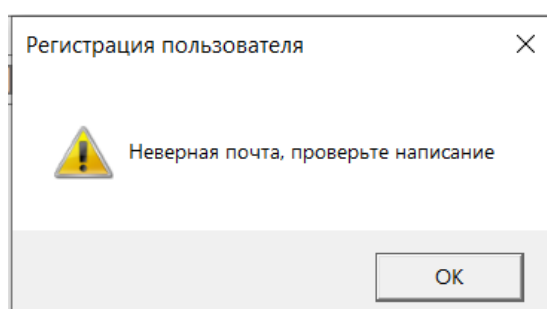


Рисунок 2.26 Ошибка ввода информации

К интерактивным функциям меню можно отнести всплывающие пояснения к каждой кнопке (см. рис 2.27).

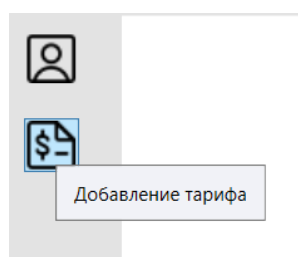


Рисунок 2.27 Меню навигации

К дополнительным функциям приложения можно отнести прямой работы компании и ее сотрудников. Для этого была создана форма обращения (см. рис. 2.28) для пользователей. Она была необходима для каких-либо вопросов по работе приложения, или при возникших вопросов при работе с приложением, или других вопросов.

Рисунок 2.28 Форма обращения пользователей

Логика работы всех вышеперечисленных окон приведена в Приложение 4.

Помимо директора компании, приложением так же будут пользоваться сотрудники компании, руководитель склада, администратор. В случае авторизации под учётной записью, встретит аналогичное пользовательскому окну.

Администратор может добавлять аккаунты определенной привилегией, также добавлять тарифы аренды, и компании клиентов. Администратор не имеет права редактировать собственную информацию, поэтому вкладка «профиль» для модератора отсутствует.

Помимо директора компании и администратора, приложением так же может пользоваться директор склада. От директора компании его отличают возможность изменения актуальности информации по клиентам, добавления аккаунты определенной привилегией. Мониторинг компаний и тарифов. Осуществление поиск компании по его названию. При необходимости, директора компании может изменять личную информацию пользователя.

Сотрудник компании имеет ограниченный доступ к приложению. Он

имеет право только добавлять и редактировать поставляемую продукцию в определенный сектор.

Основной функционал главного окна представлен в Приложение 4 (подпункт 1). В данном подпункте описан код работы основных кнопок: выход из приложения, выход из профиля, переход по кнопке на окно добавления.

В Приложение 4 (подпункт 2). Описан код работы страницы пользователя «Директор компании». В данном подпункте описан функционал работы поисковой строки, загрузки информации из подключенной базы в окно и функционал кнопки сброса введенных значений в кнопках навигации.

В Приложение 4 (подпункт 3). Описан класс для проверки введенных значений. На данном рисунке описана проверка поля «ФИО работника» и поля ввода пароля.

Также в приложении имеется страница, необходимая для вывода информации из базы данных по какому-либо пользователю. Разметка данной страницы и сама страница находится в Приложение 4 (подпункт 4).

В Приложение 4 (подпункт 5) имеется листинг кода добавления пользователя «Директор компании». В данном подпункте описывается функционал кнопки добавления нового пользователя и добавления его данных.

В Приложении 4 (подпункт 6) приведен листинг кода окна авторизации.

В нем обозначено, что пользователь уже имеется в системе, и по введенным значениям в поля авторизации проходит проверка на то, этот пользователь уже имеется в системе.

## Выводы по главе II

По итогу проведения анализа предметной области, во второй главе были разработаны ER и USE CASE диаграммы, на основе которых была создана база данных, которая позволяет быстро и эффективно распределять информацию по таблицам без избыточности данных, а также программа на языке C#.

Интуитивно-понятный интерфейс приложения позволит быстро получить или отредактировать нужную информацию из нескольких таблиц

как клиентам компании, так и её администраторам. Созданный программный продукт удобен и эффективен в использовании. База данных может быть усовершенствована и дополнена новыми данными, в нее могут быть введены дополнительные средства формирования данных.

Применение разработанной автоматизированной системы для управления умным домом позволит максимально упростить работу сотрудников компании на складе.

					09.02.07 – 4ИСП9-4	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

## ЗАКЛЮЧЕНИЕ

Целью данной выпускной квалификационной работы являлось закрепление полученных теоретических знаний и практических умений, проектирование и разработка информационной системы для учета клиентов компании. Для достижения поставленной цели использовались программные средства - система управления базами данных MS SQL Server и среда разработки Visual Studio 2022 с применением языка программирования C#. Проект выполнялся при помощи технологии WPF.

Разработанная база данных позволяет быстро и эффективно распределять, информацию о клиентах по таблицам без избыточности данных.

Созданный программный продукт удобен и эффективен в использовании. Его интуитивно-понятный интерфейс позволит пользователям без специальных навыков быстро просмотреть и отредактировать информацию из нескольких таблиц. Разработанная программа позволяет систематизировать и отображать информацию о клиентах компании, добавлять их и удалять при необходимости, а также делать выборку клиентов по фильтрам. Помимо этого, база данных предоставляет доступ к информации о сотрудниках компании, продуктах, свободных секторах и контейнерах, заключённых между клиентами и компанией для привилегированных пользователей.

Созданная база данных может быть усовершенствована и дополнена новыми данными. Применение разработанной автоматизированной системы позволит максимально упростить учёт товаров компании и других сущностей, относящихся к исследуемой сфере услуг.

## СПИСОК ЛИТЕРАТУРЫ

1. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2: учебник для вузов / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — М. Издательство Юрайт, 2019.
2. Грофф, Джеймс Р., Вайнберг, Пол Н., SQL: полное руководство, 3-е изд.: Пер. с англ. — М.: издательство «И.Д. Вильямс», 2018.
3. Гукин, Д. Программирование на С для чайников / Д. Гукин. - М.: Диалектика, 2019.
4. Гуриков, С.Р. Введение в программирование на языке Visual C#: Учебное пособие / С.Р. Гуриков. — М, 2019
5. Зыков, С. В. Программирование. Объектно-ориентированный подход учебник и практикум для академического бакалавриата / С. В. Зыков. — М. Издательство Юрайт, 2019.
6. Кубенский, А. А. Функциональное программирование учебник и практикум для академического бакалавриата / А. А. Кубенский. — М. Издательство Юрайт, 2019.
7. Кудрина, Е. В. Основы алгоритмизации и программирования на языке с# учеб. пособие для СПО / Е. В. Кудрина, М. В. Огнева. — М. Издательство Юрайт, 2019.
8. Кудрина, Е. В. Основы алгоритмизации и программирования на языке с# учеб. пособие для бакалавриата и специалитета / Е. В. Кудрина, М. В. Огнева. — М. Издательство Юрайт, 2019.
9. Малявко, А. А. Формальные языки и компиляторы учеб. пособие для вузов / А. А. Малявко. — М. Издательство Юрайт, 2018.
10. Мамонова, Т. Е. Информационные технологии. Лабораторный практикум учеб. пособие для СПО / Т. Е. Мамонова. — М. Издательство Юрайт, 2019.
11. Маркин, А. В. Программирование на sql в 2 ч. Часть 2 учебник и

практикум для бакалавриата и магистратуры / А. В. Маркин. — М. Издательство Юрайт, 2019.

12. Петкович Д. Microsoft SQL Server 2017. Руководство для начинающих: Пер. с англ. – СПб.: БХВ-Петербург, 2018.

13. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер. - СПб.: Питер, 2019.

14. Стасышин, В. М. Базы данных: технологии доступа учеб. пособие для СПО / В. М. Стасышин, Т. Л. Стасышина. — 2-е изд., испр. и доп. — М. Издательство Юрайт, 2018.

15. Трофимов, В. В. Основы алгоритмизации и программирования: учебник для СПО / В. В. Трофимов, Т. А. Павловская под ред. В. В. Трофимова. — М. Издательство Юрайт, 2019.

#### Электронные ресурсы

16. WPF 4.5 in C#: Windows Presentation Foundation in .NET 4.5, Matthew MacDonald (русский перевод: WPF: Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0 для профессионалов, Мэтью Макдональд). ([https://sd.blackball.lv/library/Pro\\_WPF\\_4.5\\_in\\_CSharp\\_4th\\_edition.pdf](https://sd.blackball.lv/library/Pro_WPF_4.5_in_CSharp_4th_edition.pdf))

17. Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers (2019) Автор: Bruce Johnson ([https://docs.yandex.ru/docs/view?tm=1653564852&tld=ru&lang=en&name=Visual-Studio-Code\\_RuLit\\_Me\\_650262.pdf](https://docs.yandex.ru/docs/view?tm=1653564852&tld=ru&lang=en&name=Visual-Studio-Code_RuLit_Me_650262.pdf))

18. Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programming (2019) Автор: Stephen Cleary ([https://docs.yandex.ru/docs/view?tm=1653564998&tld=ru&lang=ru&name=Konkurentnost\\_v\\_C\\_\\_Asinkhronnoe\\_parallelnoe\\_i\\_mnogopotchnoe\\_programmirovanie\\_2020\\_Stiven\\_Kliri.pdf](https://docs.yandex.ru/docs/view?tm=1653564998&tld=ru&lang=ru&name=Konkurentnost_v_C__Asinkhronnoe_parallelnoe_i_mnogopotchnoe_programmirovanie_2020_Stiven_Kliri.pdf))

19. Основы программирования на C# (<https://metanit.com/sharp/tutorial/1.5.php>)

20. SQL – язык реляционных баз данных. <https://docs.yandex.ru/docs/view?tm=1653565330&tld=ru&lang=ru&name=978-5-7996-1622->

9\_2016.pdf)

21. Учебник по языку SQL (DDL, DML) на примере диалекта MS SQL Server. (<https://habr.com/ru/post/255361>)

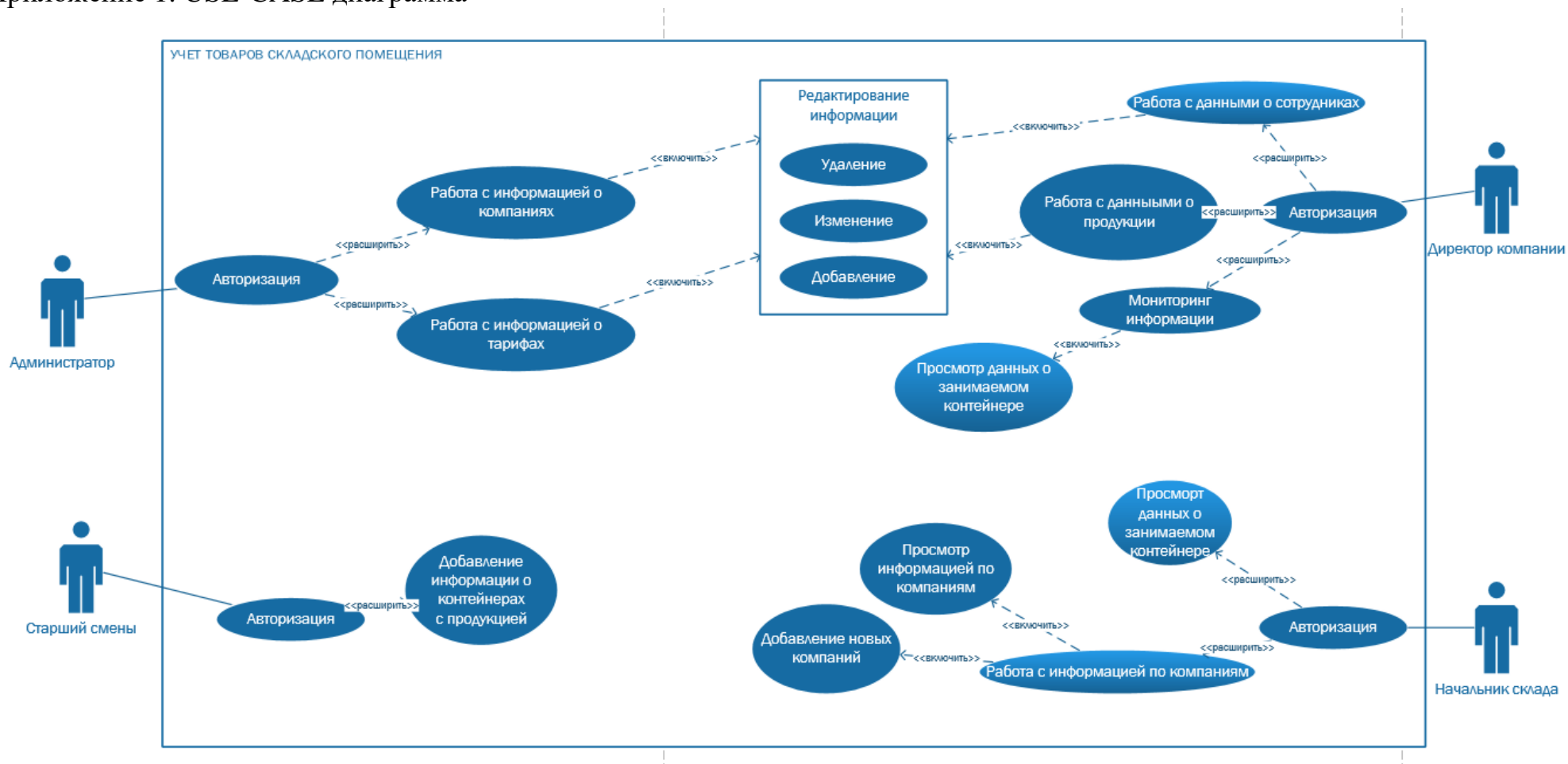
22. Ресурс по работе в WPF C# ([https://www.tutorialspoint.com/wpf/wpf\\_overview.htm](https://www.tutorialspoint.com/wpf/wpf_overview.htm))

23. Руководство по работе с Microsoft Visio (<https://coderlessons.com/tutorials/microsoft-technologies/izuchite-microsoft-visio/microsoft-visio-kratkoe-rukovodstvo>)

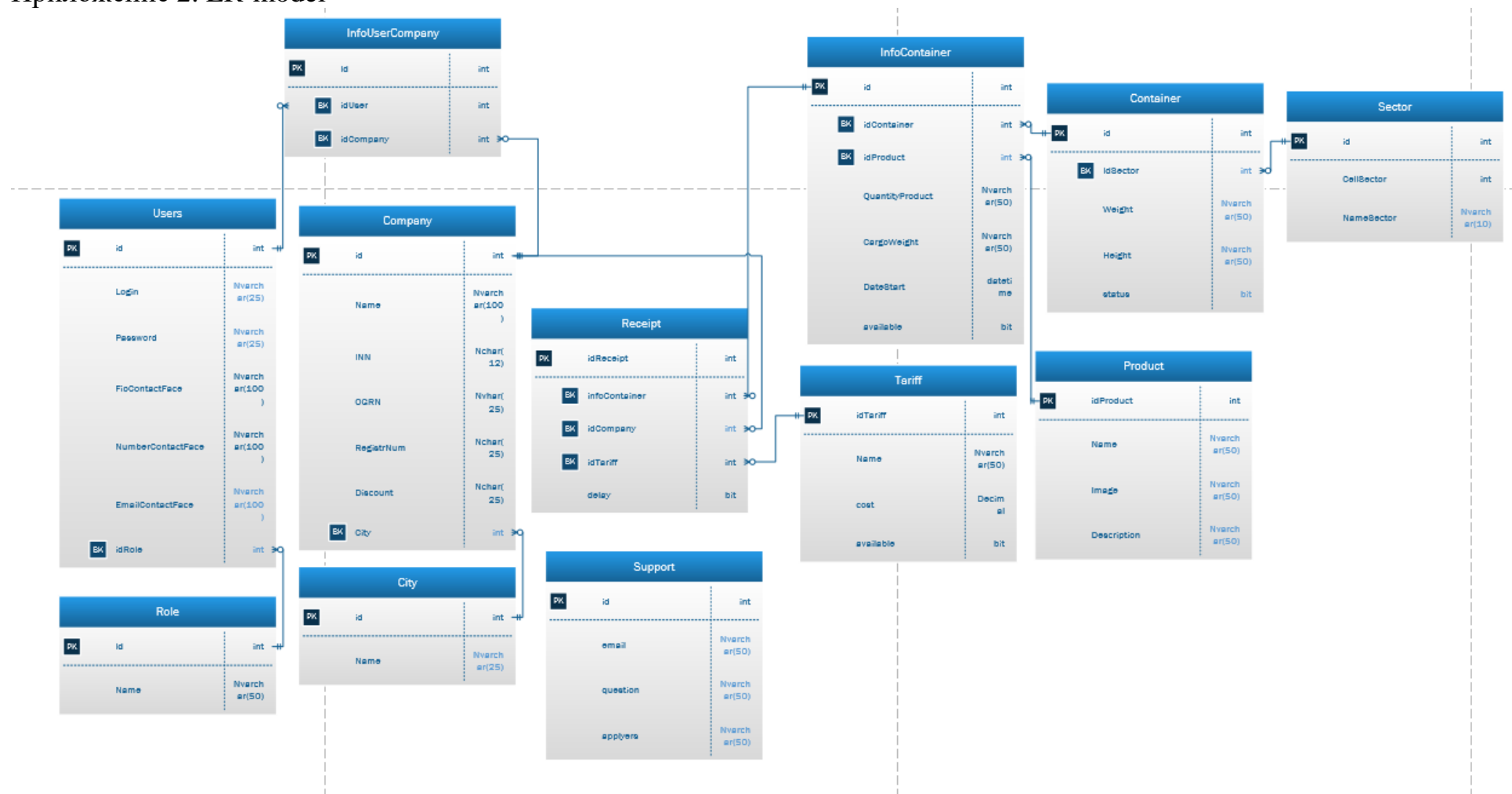
					09.02.07 – 4ИСП9-4	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32



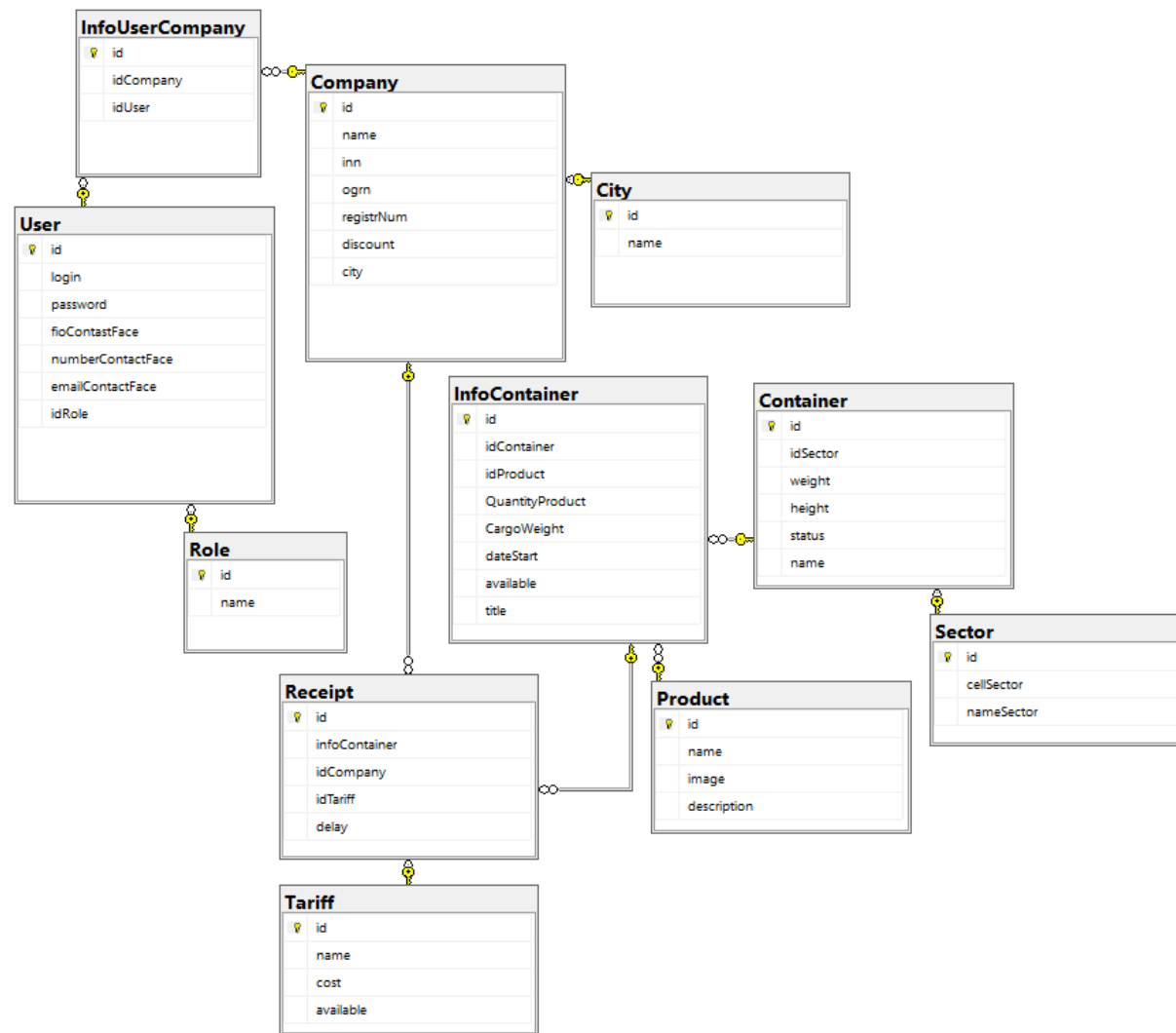
## Приложение 1. USE-CASE диаграмма



## Приложение 2. ER-model



### Приложение 3. Физическая модель



## Приложение 4. Листинг кода основных форм разработанного пользовательского приложения

Ссылка: 1

```
private void undo_Click(object sender, RoutedEventArgs e)
{
    if (MessageBox.Show("Вы действительно хотите выйти из приложения?", "Выход", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)
    {
        Application.Current.Shutdown();
    }
}
```

Ссылка: 1

```
private void ExitProdile_Click(object sender, RoutedEventArgs e)
{
    Autorization auth = new Autorization();
    this.Close();
    auth.ShowDialog();
}
```

Admin

#region DirectGoods

Ссылка: 1

```
private void companyadd_Click_1(object sender, RoutedEventArgs e)
{
    Menu menu = new Menu();
    menu.IsEnabled = false;
    Windows.AddInfo registrat = new Windows.AddInfo();
    registrat.addInfoCompany.Visibility = Visibility.Visible;
    registrat.ShowDialog();
}
```

Ссылка: 1

```
private void companyAdd_Click(object sender, RoutedEventArgs e)
{
    Menu menu = new Menu();
    menu.IsEnabled = false;
    Windows.AddInfo registrat = new Windows.AddInfo();
    registrat.addInfoCompany.Visibility = Visibility.Visible;
    registrat.ShowDialog();
}
```

### 4.1 Листинг кода навигаций главного окна

```

public PageDirectoryCompany()
{
    InitializeComponent();
    GridMain.ItemsSource = googsHelper.GetVm_DirUser();
}

Ссылка: 1
private void SearchButton_Click(object sender, RoutedEventArgs e)
{
    if (NameCompanyTBox.Text is null && FIOTBox.Text is null)
        return;
    if (NameCompanyTBox.Text is null)
    {
        if (FIOTBox.Text != null)
        {
            GridMain.ItemsSource = googsHelper.GetVm_DirUser().Where(i => i.ФИО_Работника == (string)FIOTBox.Text).ToList();
        }
    }
    else if (NameCompanyTBox.Text != null)
    {
        if (FIOTBox.Text is null)
        {
            GridMain.ItemsSource = googsHelper.GetVm_DirUser().Where(i => i.Компания == (string)NameCompanyTBox.Text).ToList();
        }
        else
        {
            GridMain.ItemsSource = googsHelper.GetVm_DirUser().Where(i => i.Компания == (string)NameCompanyTBox.Text && i.ФИО_Работника == (string)FIOTBox.Text).ToList();
        }
    }
}

Ссылка: 1
private void RebootButton_Click(object sender, RoutedEventArgs e)
{
    NameCompanyTBox.Text = null;
    NameCompanyTBox.Text = null;
    GridMain.ItemsSource = googsHelper.GetVm_DirUser();
}

```

4.2 Листинг кода страницы пользователя «Директор компании»

```

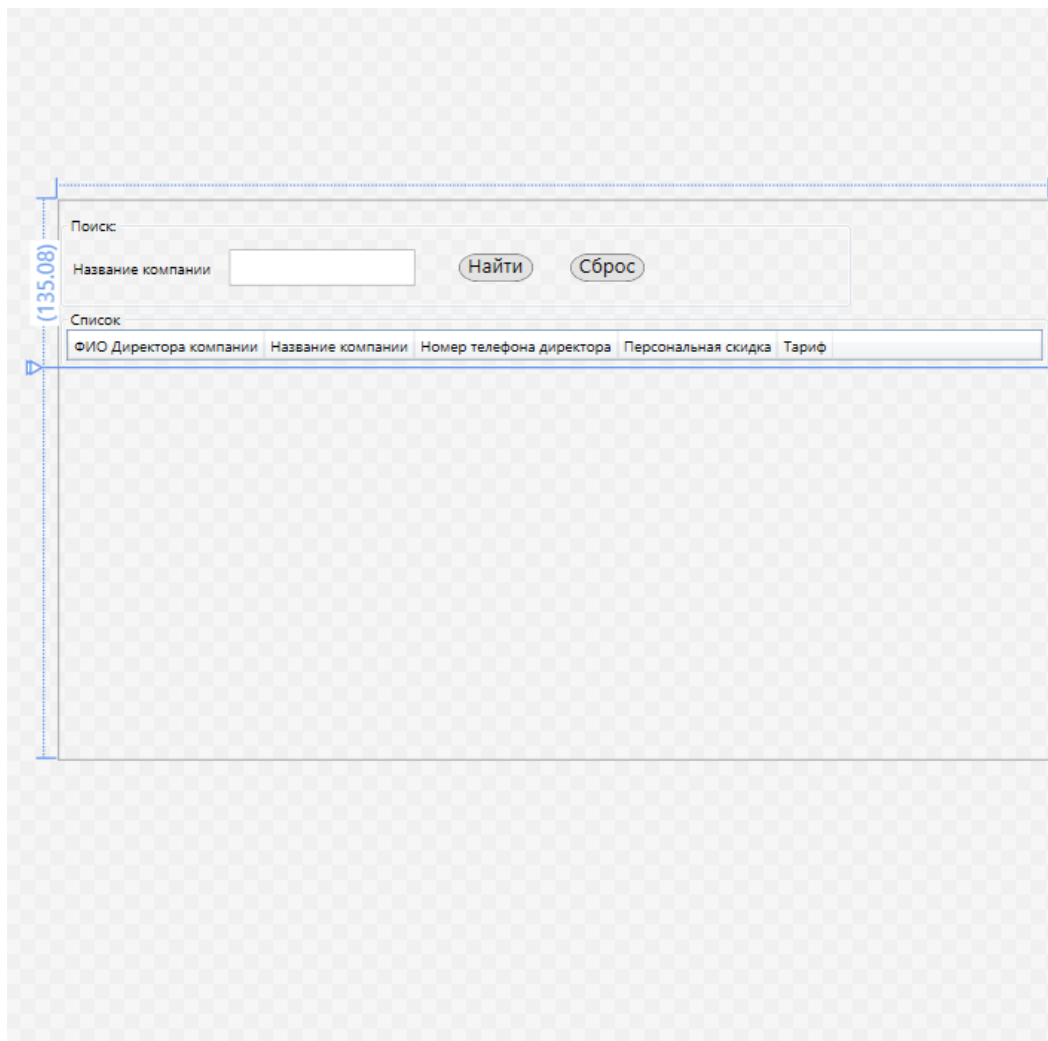
class ValidationClass
{
    См. прим. 4
    public static bool ValidateFIO(string name)
    {
        int s = 0;
        int t = 0;
        if (name != "ФИО Руководителя склада"
            && name != "ФИО Руководителя компании" && name != "ФИО Работника")
        {
            if (t < name.Length)
            {
                while (s < name.Length)
                {
                    if ((name[s] >= 'А' && name[s] <= 'Я' || name[s] >= 'а' && name[s] <= 'я') ||
                        (name[s] == '-' || name[s] == ' '))
                    {
                        s++;
                    }
                    else return false;
                }
            }
            else return false;
        }
        else return false;
        return true;
    }

    См. прим. 3
    public static bool ValidatePassw(string passw)
    {
        int t = 7;

        if (passw.Length >= t)
        {
            for (int i = 0; i < passw.Length; i++)
            {
                if ((passw[i] >= '0') && (passw[i] <= '9'))
                {
                    for (int d = 0; d < passw.Length; d++)
                    {
                        for (int j = 0; j < passw.Length; j++)
                        {
                            if ((passw[j] >= 'a') && (passw[j] <= 'z'))
                            {
                                for (int k = 0; k < passw.Length; k++)
                                {
                                    if ((passw[k] >= 'A') && (passw[k] <= 'Z'))
                                    {
                                        return true;
                                    }
                                }
                                return false;
                            }
                        }
                    }
                    return false;
                }
            }
            return true;
        }
        else
        {
            return false;
        }
    }
}
    См. прим. 4

```

4.3 Листинг кода с основной проверкой введенных пользователем значений



```

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
d:DesignHeight="450" d:DesignWidth="800"
Title="PageDirectoryGoods">
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition Height="auto"/>
    <RowDefinition Height="auto"/>
  </Grid.RowDefinitions>
  <StackPanel Orientation="Vertical"
    VerticalAlignment="Center">
    <GroupBox x:Name="GRBox"
      HorizontalAlignment="Left"
      Width="640"
      Margin="0 10 0 0">
      <StackPanel Orientation="Horizontal">
        <TextBlock Text="Поиск:"
          VerticalAlignment="Center"/>
        <StackPanel Orientation="Vertical">
          <StackPanel Orientation="Horizontal">
            <StackPanel x:Name="NameSearch"
              VerticalAlignment="Center">
                <StackPanel
                  Orientation="Horizontal"
                  HorizontalAlignment="Left">
                  <Label Content="Название компании"
                    VerticalContentAlignment="Center"
                    HorizontalAlignment="Left"/>
                  <xctk:WatermarkTextBox
                    Margin="10"
                    FontSize="16"
                    x:Name="NameCompanyTBox"
                    Width="150"
                    />
                </StackPanel>
              </StackPanel>
            <StackPanel
              Orientation="Horizontal"
              HorizontalAlignment="Right"
              Margin="10 0 0 0">
                <Button Content="Найти"
                  x:Name="SearchButton"
                  Click="SearchButton_Click"
                  Width="60"
                  Margin="15"
                  FontSize="16"
                  Style="{StaticResource ButtonAutor}"/>
                <Button Content="Сброс"
                  x:Name="RebootButton"
                  Click="RebootButton_Click"
                  Width="60"
                  Margin="15"
                  FontSize="16"
                  Style="{StaticResource ButtonAutor}"/>
              </StackPanel>
            </StackPanel>
          </StackPanel>
        </StackPanel>
      </GroupBox>
    </StackPanel>
  </Grid>

```

4.4 Листинг страницы вывода информации по пользователю «Директор компании»

```

Source: 1
private void SaveAddDirectory_Click(object sender, RoutedEventArgs e)
{
    string logBD = CensGoodsEnt.User.Where(i => i.emailContactFace ==
    loginTextBox.Text.ToString()).Select(j => j.emailContactFace)
    .FirstOrDefault();

    #region Директор склада
    if (loginTextBox.Text == "" || PasswordTextBox.Text == "" ||
    NameMainGoods.Text == "" || NumberMainGoods.Text == ""
    || EmailMainGoods.Text == "")
    {
        MessageBox.Show("Не все поля заполнены!");
    }
    else if (logBD != null)
    {
        MessageBox.Show("Почта уже используется");
    }
    else if (ValidatePassw(PasswordTextBox.Text) == false)
    {
        MessageBox.Show("Недопустимый пароль, необходимо ввести (от 8 символов, минимум одна: заглавная, строчная, число)",
        "Регистрация пользователя", MessageBoxButton.OK, MessageBoxImage.Exclamation);
    }
    else if (ValidatePhone(NumberMainGoods.Text) == false)
    {
        MessageBox.Show("Недопустимый номер телефона, проверьте написание",
        "Регистрация пользователя", MessageBoxButton.OK, MessageBoxImage.Exclamation);
    }

    else if (ValidateFIO(NameMainGoods.Text) == false)
    {
        MessageBox.Show("Недопустимое имя пользователя",
        "Регистрация пользователя", MessageBoxButton.OK, MessageBoxImage.Exclamation);
    }
    else if (ValidateEmail(EmailMainGoods.Text) == false)
    {
        MessageBox.Show("Неверная почта, проверьте написание",
        "Регистрация пользователя", MessageBoxButton.OK, MessageBoxImage.Exclamation);
    }
    else if (ValidateLogin(loginTextBox.Text) == false)
    {
        MessageBox.Show("Недопустимый логин, необходимо ввести (от 8 символов, минимум одна: заглавная, строчная)",
        "Регистрация пользователя", MessageBoxButton.OK, MessageBoxImage.Exclamation);
    }
    else
    {
        MessageBoxResult result = MessageBox.Show(
        NameMainGoods.Text + " " +
        "Логин: " + loginTextBox.Text + "\n" +
        "Пароль: " + PasswordTextBox.Text + "\n", "Создать пользователя:", MessageBoxButton.YesNo);
        if (result == MessageBoxResult.Yes)
        {
            CensGoodsEnt.User.Add(new EF.User()
            {
                login = loginTextBox.Text.ToString(),
                password = PasswordTextBox.Text.ToString(),
                fioContactFace = NameMainGoods.Text.ToString(),
                numberContactFace = NumberMainGoods.Text.ToString(),
                emailContactFace = EmailMainGoods.Text.ToString(),
                idRole = 3
            });
            CensGoodsEnt.SaveChanges();
            Menu menu = new Menu();
            menu.IsEnabled = true;
            this.Close();
        }
    }
}

```

4.5 Листинг кода добавления пользователя «Директор компании»



```

C:\Users\1
private void entry_Click(object sender, RoutedEventArgs e)
{
    try
    {
        User user = CensGoodsEnt.User
        .ToList().Where(i => i.login == LoginTxt.Text && i.password == PasswTxt.Password)
        .FirstOrDefault();
        if (user != null)
        {
            Class.DataUser.User1 = user;
            if (Class.DataUser.User1.idRole == 4)
            {
                Menu menu = new Menu();
                menu.FunctionAdmin.Visibility = Visibility.Visible;
                this.Hide();
                menu.ShowDialog();
            }
            else if (Class.DataUser.User1.idRole == 3)
            {
                Menu menu = new Menu();
                menu.FunctionMainGoods.Visibility = Visibility.Visible;
                this.Hide();
                menu.ShowDialog();
            }
            else if (Class.DataUser.User1.idRole == 2)
            {
                Menu menu = new Menu();
                menu.FunctionMainCompany.Visibility = Visibility.Visible;
                this.Hide();
                menu.ShowDialog();
            }
            else if (Class.DataUser.User1.idRole == 1)
            {
                Menu menu = new Menu();
                menu.FuntionEmployee.Visibility = Visibility.Visible;
                this.Hide();
                menu.ShowDialog();
            }
        }
        else
        {
            MessageBox.Show("Логин или пароль введены неверно");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
C:\Users\1

```

4.6 Листинг кода окна авторизации