МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РД

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РД «ТЕХНИЧЕСКИЙ КОЛЛЕДЖ им. Р.Н. Ашуралиева»

Отделение «Программирование»

П(Ц). К УГС 09.00.00 Информатика и вычислительная техника

Специальность 09.02.07 «Информационные системы и программирование»

**Курсовая работа (проект)**

по междисциплинарному курсу

**МДК.02.01 Технология разработки программного обеспечения**

на тему

«Разработать систему для автоматизации процесса учета товародвижения в магазине ювелирных изделий»

Выполнил студент группы 1-ИСП-9-3

Магомедов М. А.

Руководитель: Азизова Л.Н.

Махачкала 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РД

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РД «ТЕХНИЧЕСКИЙ КОЛЛЕДЖ им. Р.Н. Ашуралиева»

Отделение «Программирование»

П(Ц). К УГС 09.00.00 Информатика и вычислительная техника

Специальность 09.02.07 «Информационные системы и программирование»

**Задание на курсовую работу (проект)**

Студента: Магомедова М. А

Группы: 1-ИСП-9-3

по **МДК.02.01 Технология разработки программного обеспечения**

1. Тема: «Разработать систему для автоматизации процесса учета товародвижения в ювелирном магазине»
2. Дата выдачи задания\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Дата представления к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Исходные данные: тема КП, задание на КП, структура КП, литературные источники, требования к функциональной части системы, требования к пользовательскому интерфейсу системы

# Оглавление

[Введение 3](#_Toc52810032)

[1. Разработка системного проекта 5](#_Toc52810033)

[1.1 Назначение разработки 5](#_Toc52810034)

[1.2 Требования к функциональным характеристикам 5](#_Toc52810035)

[1.3 Требования к надежности и безопасности 6](#_Toc52810036)

[1.4 Требования к составу и параметрам технических средств 7](#_Toc52810037)

[1.5 Требования к информационной и программной совместимости 7](#_Toc52810038)

[2 Разработка технического проекта 8](#_Toc52810039)

[2.1 Анализ требований и определение спецификаций программного обеспечения 8](#_Toc52810040)

[2.2 Детальное проектирование программного обеспечения (конструирование прототипа) 9](#_Toc52810041)

[3 Реализация 11](#_Toc52810042)

[3.1 Обоснование выбора средств разработки 11](#_Toc52810043)

[3.2 Описание основных программных модулей 11](#_Toc52810044)

[4. Тестирование 23](#_Toc52810045)

[4.1 Объект испытаний 23](#_Toc52810046)

[4.2 Цель испытаний 23](#_Toc52810047)

[Заключение 26](#_Toc52810048)

[Список используемой литературы 27](#_Toc52810049)

[Приложение А 28](#_Toc52810049)

[Приложение Б 31](#_Toc52810049)

[Приложение В 39](#_Toc52810049)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

Задание принял(а) к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

# Введение

На сегодняшний день существует большое количество программных

средств автоматизации бизнес-процессов в той или иной организации, предприятии или какой-либо другой компании. Организации стали сводить к минимуму документы на бумажных носителях и начали использовать электронный документооборот.

**Цель:** разработать программное обеспечение, которое позволит автоматизировать и облегчить товародвижение в ювелирном магазине.

**Характеристика:** на современном рынке представлено достаточное количество программных продуктов для автоматизации различных сфер деятельности предприятий. Но следует учитывать, что каждое предприятие может иметь свои особенности, которые не позволяют ему работать уже с представленными на рынке программными продуктами, так как для организаций необходима только часть представленных возможностей программного обеспечения, а платить приходится за весь комплект и его сопровождения. Кроме того, стоимость программного обеспечения велика для малого бизнеса. Внедрение на предприятие подобного программного продукта, позволит автоматизировать те процессы, которые необходимы для организации за умеренную плату.

Автоматизация товародвижения в салоне ювелирных изделий позволяет в первую очередь сэкономить время работников обслуживающих клиентов. Такая автоматизированная система позволит компании быстро, оперативно и качественно обрабатывать запросы клиентов и приводит к минимуму усилия сотрудников, которые занимаются формированием договоров и помогают им качественно и безопасно выполнять свою работу.

Автоматизированная система нацелена снизить расходы и время на выполнение работы, так как большую часть рутинной работы возьмёт на себя разработанное программное обеспечение.

**Основные задачи программного обеспечения:**

1. Повышение качественного уровня обслуживания клиентов.
2. Повышение оперативности работы сотрудников.
3. Снижение трудозатрат на поиск информации и подготовку документов.
4. Сокращение персонала юверлиного магазина.
5. Сокращение расходов на материалы (бумага, картриджи, ручки).
6. Ускорение процесса ввода, вывода и обработки информации.
7. Формирование отчётов о деятельности кампании.
8. Детализированный расчёт стоимости заказа

# Разработка системного проекта

## Назначение разработки

Разработать систему автоматизации товародвижения в ювелирном магазине, которая позволит проходить авторизацию, а затем на основе привелегий работника позволит либо регистрировать клиентов, чеки по продажам, чтобы они хранились в Базе Данных, либо работу по администрированию.

## Требования к функциональным характеристикам

* Хранение в базе данных данные клиентов, сотрудников, изделий, производителей и договоров.
* Администратор системы, который разграничивает уровни доступа в системе, предоставляет доступ новым сотрудникам, имеет право удалять или изменять параметры параметры доступа сотрудников.
* Пароль нового сотрудника и данные нового сотрудника регистрирует администратор.
* Введённый адрес электронной почты для запроса доступа должен быть действительным и уникальным. Если такой email уже существует в базе данных сотрудников, то система предупредит об этом и поможет восстановить доступ, если сотрудник не помнит свой пароль.
* Система проверяет на корректность введённый логин сотрудника и на действительность доменного имени электронной почты.
* После авторизации, система на основе должностных прав предоставляет набор функционала.
* Добавлять новое изделие в базу данных, заполняя все необходимые поля.
* Отображать все существующие изделия в базе данных, поиск и фильтрация нужного изделия по всем критериям.
* Распечатать данные нужного изделия, или категорию изделий, удовлетворяющих фильтрации или поиску.
* Сохранять в PDF файл данные нужного изделия, или категорию изделий, удовлетворяющих фильтрации или поиску.
* Редактировать выбранные данные в базе.
* Удаление данных из базы.
* Регистрация клиентов, по предоставлению всех необходимых документов.
* Отображение всех зарегистрированных клиентов в базе данных.
* Фильтрация клиентов и нечеткий поиск по всем критериям.
* Распечатать данные нужного клиента, или категорию клиентов, удовлетворяющих фильтрации или поиску.
* Сохранять в PDF файл данные нужного клиента, или категорию клиентов, удовлетворяющих фильтрации или поиску.
* Формирование договора, выбор клиента и желаемого им изделия с указанием суммы.
* Создание чека и договора.
* Возможность распечатать чек, сохранить в PDF документ.

## Требования к надежности и безопасности

Единая база данных будет храниться в сервере, клиентские компьютеры сотрудников будут синхронно работать с базой данных. Все их запросы будет принимать и обрабатывать сервер. Все данные находятся в безопасности, так как Системой Управления Базы Данных при создании единой базы данных для автосалона служил Microsoft SQL Server, который включает все протоколы безопасности как от возможной утечки данных, так и от непредвиденных сбоев всей системы. Все возможные события нарушения целостности базы данных учтены системой. Программа не допустит аварийного завершения работы при критических ошибках, все возможные ошибки будут оперативно обработаны и сообщены сотруднику. При сбоях все данные будут сохранены, никакой угрозы целостности данных со стороны сбоев нет. Система грамотно реагирует на ввод недопустимого вида информации и попросит сотрудника исправить формат вводимых данных. Все ошибочные действия сотрудника система отловит и не даст нарушить целостность системы. При таких ситуациях система даст знать всплывающими окнами и уведомлениями сотрудника с текстом ошибки.

## Требования к составу и параметрам технических средств

Для комфортной эксплуатации данной системы, минимальными требованиями к техническим характеристикам является:

* Операционная система Windows 10 64x
* CPU intel Pentium Gold (2 ядра, 2 потока, 4 Мб кеш памяти), тактовая частота больше 1.2 ГГрц.
* GPU встроенное графическое ядро
* RAM 4 Гб. (DDR 3)
* Жесткий диск на 55 Гб.
* Блок питания 300 Ватт
* Монитор 18 дюймов, с разрешением 1366x720
* Клавиатура
* Мышь

## Требования к информационной и программной совместимости

Система разработана на языке программирования C# (си шарп), на основе платформы .NET Framework 7.2. Может работать в совместимости с операционной системой семейства Microsoft Windows 64x, а конкретно Windows 10 64x. Данная система разработана с помощью технологии Windows Presentation Foundation .NET Framework 7.2, класса Desktop Application.

# 2 Разработка технического проекта

## 2.1 Анализ требований и определение спецификаций программного обеспечения

Система требует вовлечение следующих лиц:

* Администратор
* Сотрудник
* Клиент.

На начальном этапе система требует заполнить данные для входа, далее на основе его рабочих обязанностей программа открывает либо страницу администратора, где он может работать с данными о рабочих, либо открывается страница для сотрудника, где он может проконсультировать клиента, показать ему товар, и если клиент предоставит необходимые данные, то может оформить ему покупку.

**Роли:**

Роль администратора системы предоставлять доступ в систему новым пользователям, удостовериться о подлинности предоставленных документов клиента остается на программу. Так же восстанавливать доступ и удалять.

Роль сотрудника состоит в обслуживании клиентов, который хотят проконсультироваться или же купить товар. Сотрудник авторизуется в систему и регистрирует клиента, после чего предоставляет на выбор, какие автомобили сейчас есть и свободны для проката. После чего сотрудник формирует договор, привязывая данные клиента и данные выбранной машины в одну цепочку.

Клиент, это обычный человек, который обращается в магазин, чтобы приобрести некое изделие. Придя в магазин его обслуживает сотрудник, после чего формируется договор, и получив чек на оплату, клиент может забрать приобретенный им товар.

Система работает по принципу клиент-сервер. Сотрудник авторизуется и попадает в свою рабочую площадку в программе, он может просматривать существующие товары и при необходимости редактировать, удалять существуюшие данные. Так же может добавлять новые изделия. Сотрудник отправляет запрос в систему, а система обрабатывает запрос сотрудника, если все данные верны и не вредят целостности системы, запрос сотрудника будет выполнен, иначе система сообщит об ошибке. Так же система предоставляет регистрацию нового клиента, который намеревается приобрести изделие, сотрудник вводит данные клиента в систему, после чего идет обработка запроса, если запрос удовлетворяет требованиям системы, то есть клиент предоставил все необходимые документы и все они действительны, то система добавит нового клиента и даст ему возможность совершить покупку. Сотрудник может редактировать и удалять данные клиента, и тут все действия сотрудника будут жестко контролироваться системой. После чего, новый клиент готов заключить договор и приобрести товар, заключение договора тоже контролируется системой. Если желаемый приобрести товар отсутствует, система не разрешит заключить договор о аренде этого автомобиля и выдаст сообщение о ошибки. Редактирование и удаление договора тоже контролируется системой, дабы избежать серьёзных сбоев и ошибок в системе.

## 2.2 Детальное проектирование программного обеспечения (конструирование прототипа)

Разбор диаграммы, как полную и подробную структурированную базу (диаграмма была создана при помощи программы Microsoft Visio Professional). В первую очередь сотрудник должен войти в систему введя свой Логин и Пароль. После подтверждения своей личности сотрудник имеет доступ в свою рабочую площадку и может полноценно обслуживать клиентов. Сотрудник должен зафиксировать все необходимые документы клиента в базе данных, чтобы клиент мог получить услуги магазина. Далее, после регистрации клиента, сотрудник заключает договор, какое изделие приобретается клиентом. Есть несколько категорий товаров, это: Кольца, Браслеты, Цепи, Серьги.

Так же сотрудник может регистрировать новые товары, которые поступили.

Для того чтобы Администратор предоставил доступ, сотруднику необходимо ввести Email, после чего система проверяет на корректность введённый Email и на наличие такого Email в БД, так как указанный электронный адрес будет служить логином при авторизации в систему у сотрудников, посему он должен быть уникальным. Если Email верного формата и его нет в БД, система одобряет доступ и позволяет создать пароль. Если email неверного формата или он уже есть в БД, система предупредит об этом, и попросит сбросить пароль, если такой email уже есть в БД или ввести корректный адрес электронной почты для получения доступа.

# 

# 3 Реализация

## 3.1 Обоснование выбора средств разработки

Учитывая то, что данная система нацелена работать на платформе операционных система семейства Windows, а именно Windows 10, наиболее подходящим средой разработки самого софта было признано среда разработки Microsoft Visual Studio 2019, и использование SDK .NET Framework 4.7.2. данная SDK считается наиболее устойчивой и наиболее новой для .NET. Технология разработки интерфейса программы служил Windows Presentation Foundation со встроенным языком разметки XAML, который позволяет разработать гибкий и очень приятный на глаз дизайн программы. При выборе языка программирования были учтены такие качества как: быстродействие, качество написанного кода, наименьший риск ошибок при работе с сервером, и возможность разработать необходимый функционал системы. Этим языком является C# (си шарп), именно C# включает все необходимые качества для реализации подобных автоматизированных систем. В качестве проектировки и создания единой базы данных для автосалона был принят Система Управления Базами Данных (СУБД) Microsoft SQL Server 2019, своим универсальным инструментом Management SQL Server Tools 18, который позволил максимально быстро разработать требуемую базу данных и максимально минимизировать риск утечки.

## 3.2 Описание основных программных модулей

Теперь можно представить автоматизированную систему для Ювелирного магазина. При запуске системы, она будет иметь следующий вид.

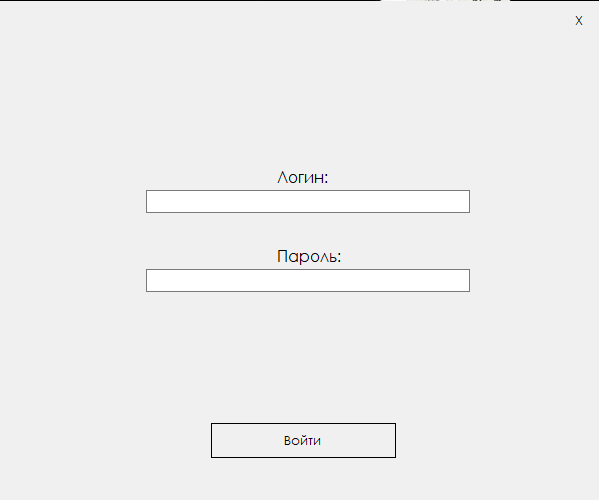


Рисунок 1. Окно авторизации

Для получения доступа в систему, сотрудник должен пройти обязательную авторизацию. Если введённые данные неверные, система даст об этом знать, как показано на рисунке 2.

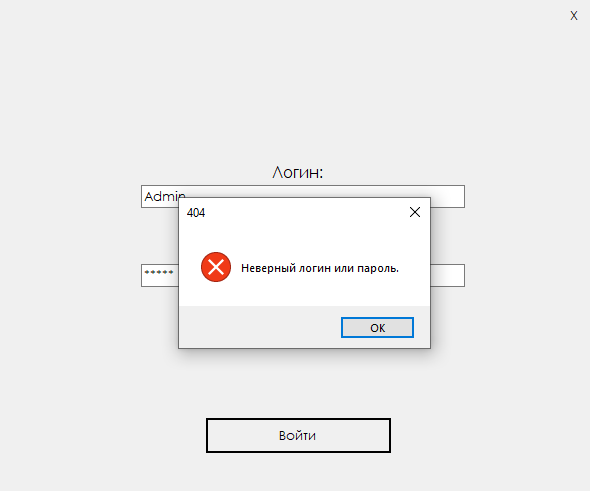


Рисунок 2. Неверные данные авторизации

После успешной авторизации, перед сотрудником откроется стартовое окно:

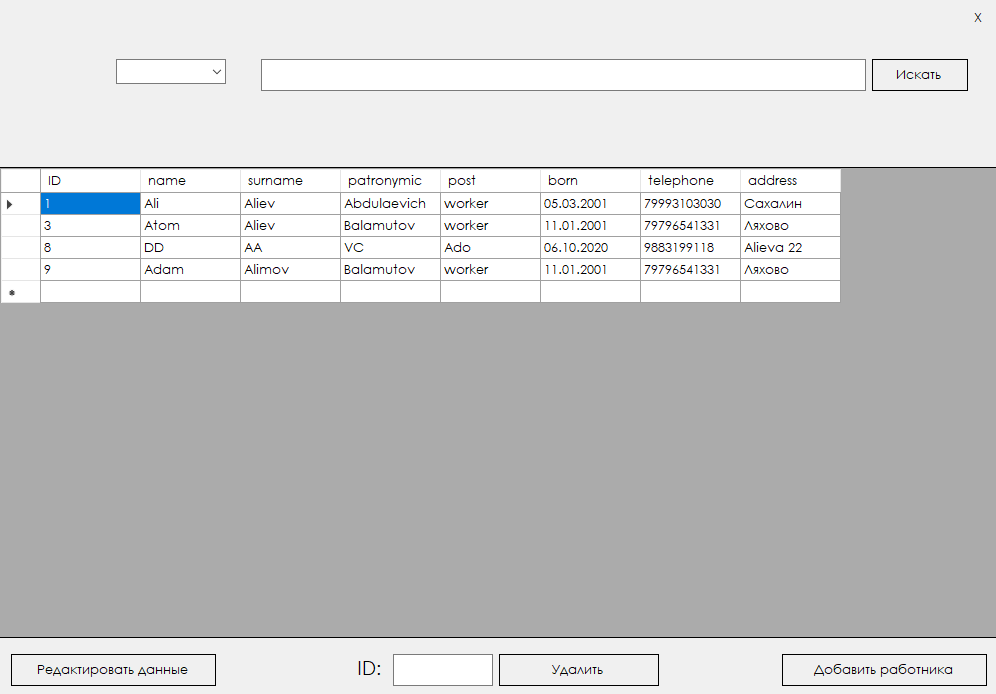


Рисунок 3. Стартовое окно Администратора.

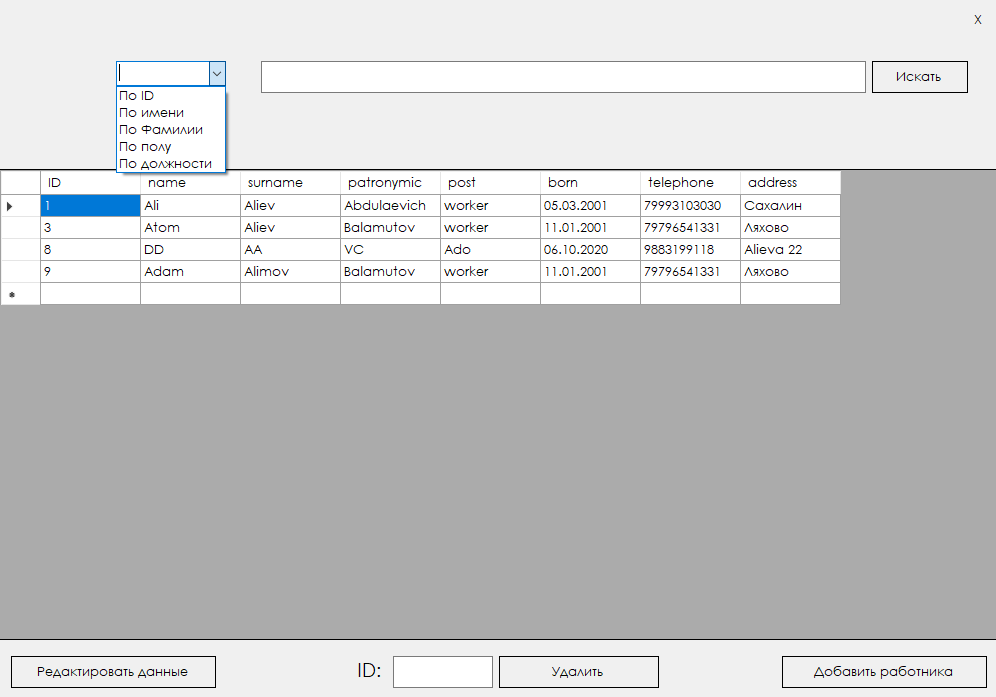
Главное окно имеет 4 кнопки:

* Кнопку поиска информации
* Кнопку изменения данных
* Кнопку удаления данных
* Кнопку добавления работника

Рассмотрим каждое из них подробнее.

**Поиск.**

Поиск осуществляется по нескольким критериям на выбор: ID, имя, фамилия, пол, должность. (рис 4)

Рис 4. Выбор параметров поиска.

**Редактирование данных.**

При нажатии на кнопку редактирования данных открывается отдельное окно со своим интерфейсом. (рис 5)

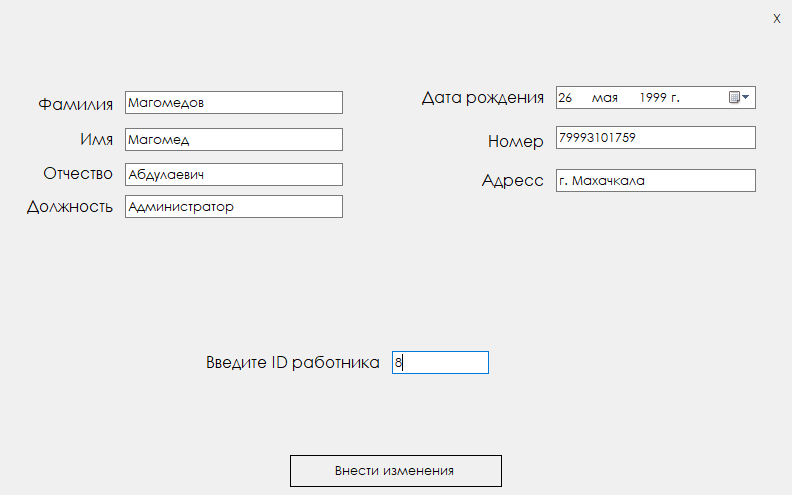


Рисунок 5. Окно для редактирования данных.

Введеные даные сохраняются в базе, результат на изображении 6.

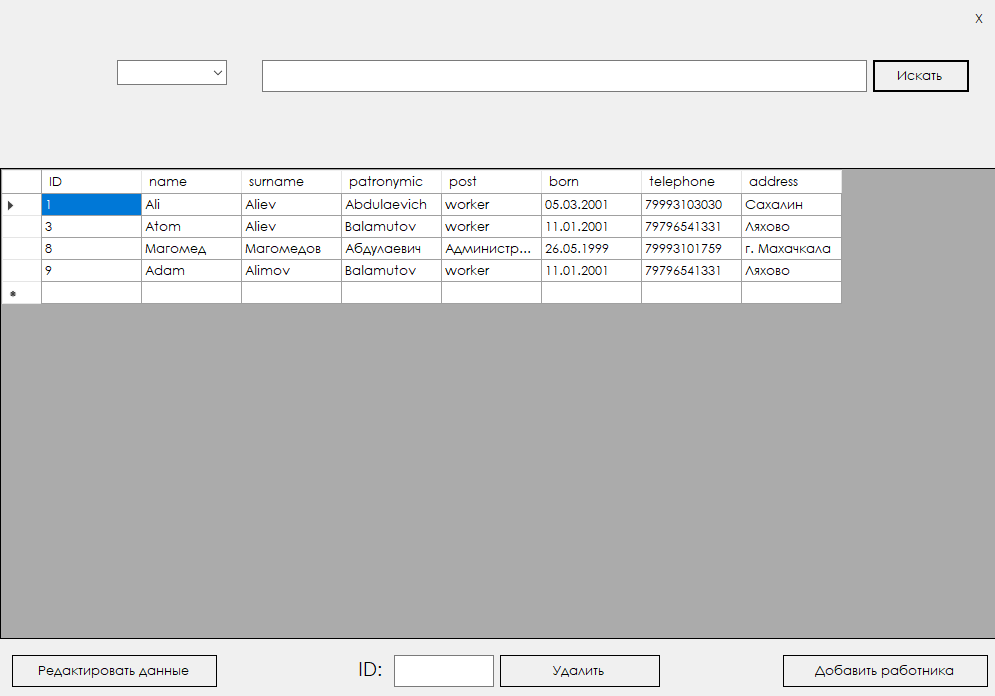


Рисунок 6. Сохраненные данные под ID 8.

**Добавление данных.**

В этом окне можно добавлять новых сотрудников в базу.

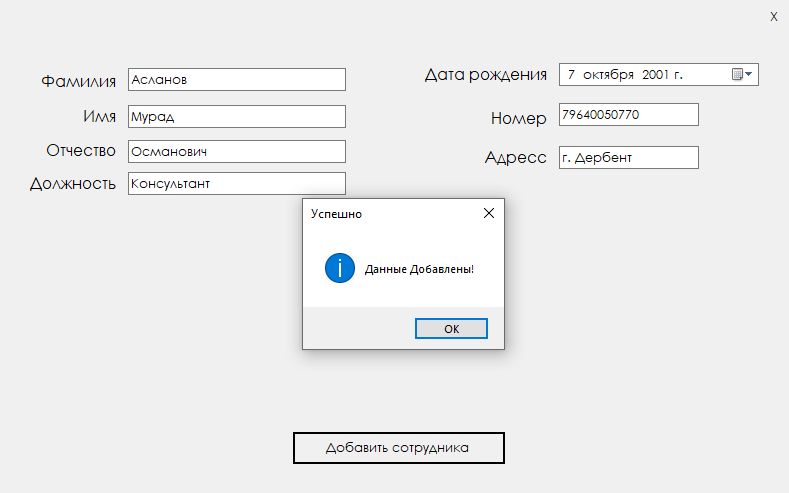


Рисунок 7. Регистрация нового сотрудника.

При регистрации мы случайно допустили ошибку и создали двух сотрудников с одинаковыми данными, это хорошая возможность показать следующую функцию.

**Удаление данных.**

В нижнем окне вводим ID сотрудника, которого нам необходимо удалить.

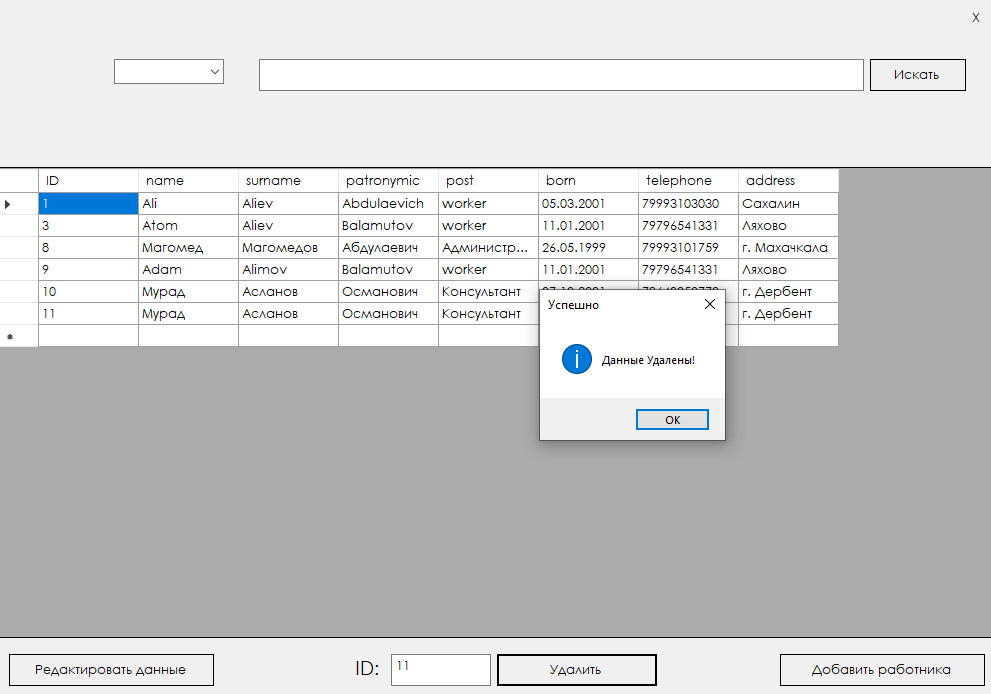
****

Рисунок 8. Удаление данных.

Как результат, после нажатия на кнопку удаляются выбранные данные.

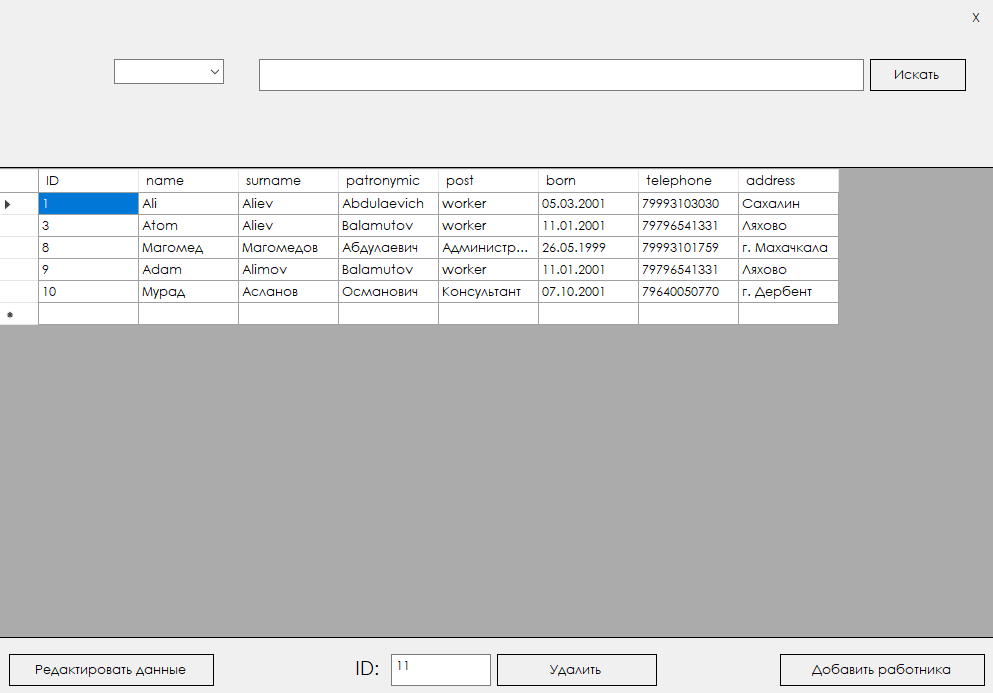


Рисунок 9. Таблица данных после обновления.

Так же в окне отображения данных мы можем выгружать данные в PDF файл, демонстрация в рисунке 10

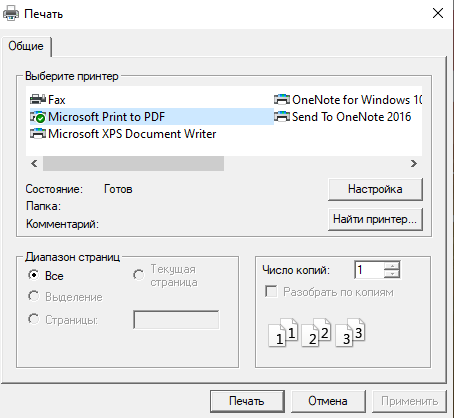


Рисунок 10. Распечатать

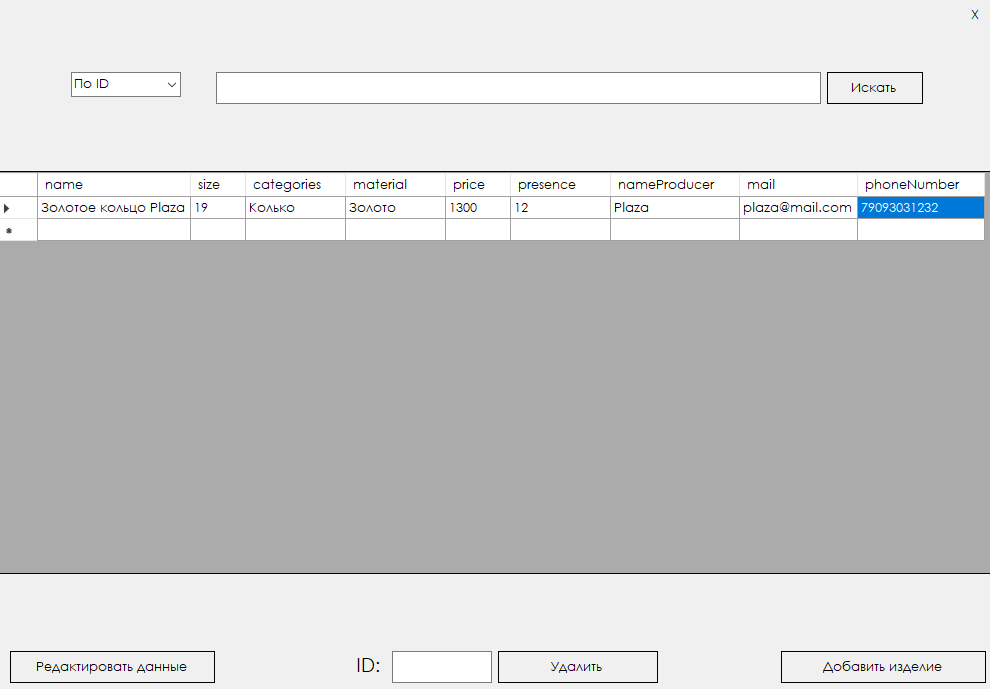


Рисунок 11. Отображение данных сотрудников

Панель сотрудника имеет много общего с панелью администратора, но работа с данными у них отличаются. Помимо основных задач по работе с информацией, тут существует дополнительная кнопка – **Договор.**

Данная кнопка открывает новое окно рис 12.

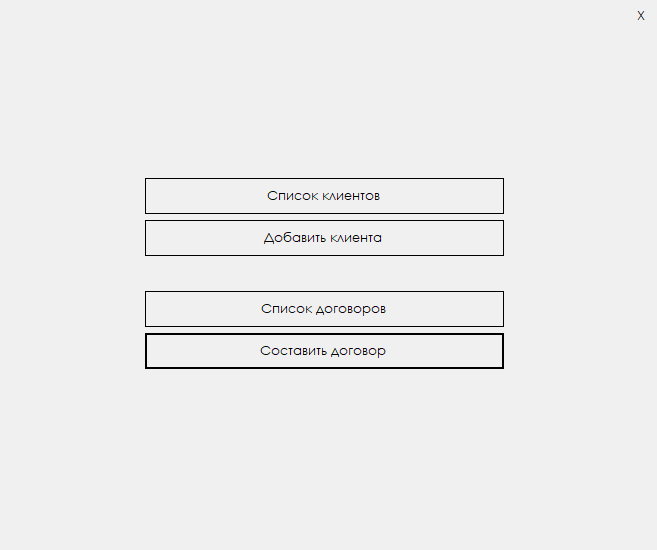


Рисунок 12. Окно Контрактов

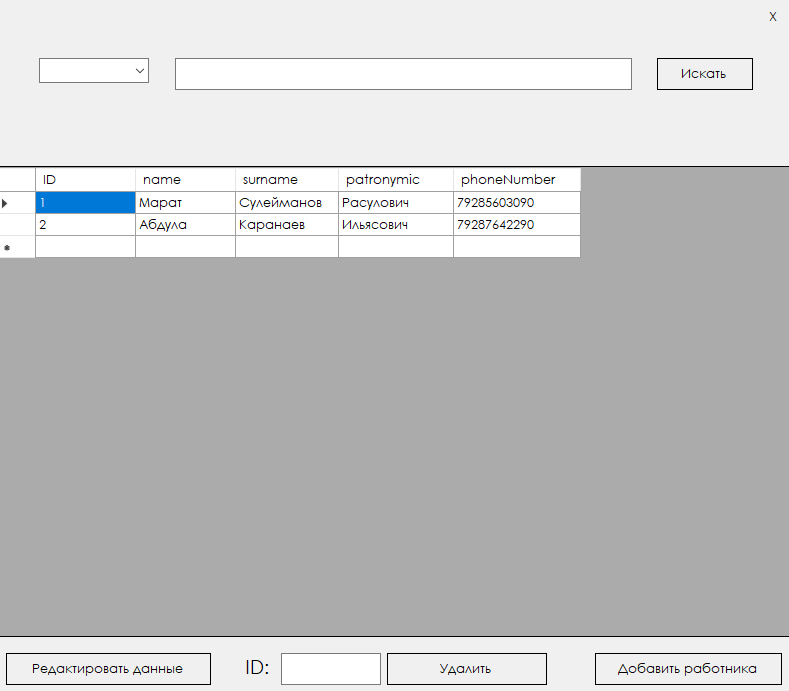
При нажатии на список клиентов мы получаем окно с отображением уже имеющихся клиентов. 

Рисунок 13. Список клиентов

На экране отображения данных Клиентов доступны те же функции, как печатать на принтере или сохранить в PDF и выгрузка данных в TXT.

Поиск и организован по следующим критериям: Имя, Фамилия, Номер и ID.

Окно добавления нового клиента выглядит максимально просто и минималистично.

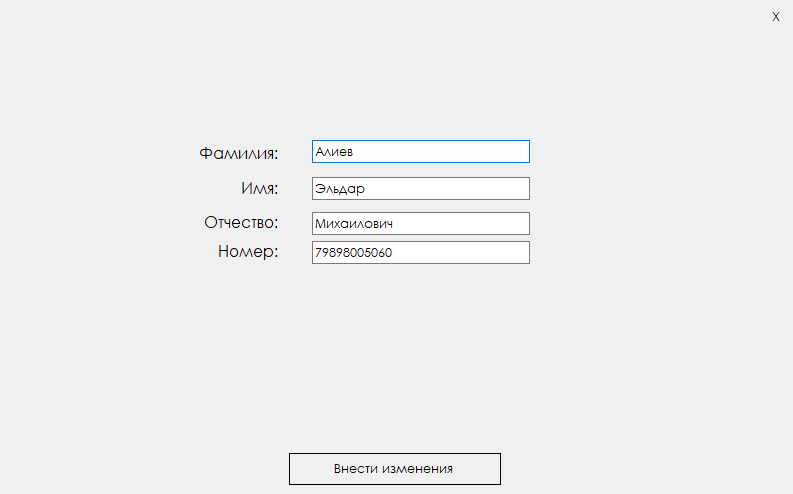


Рис 14. Окно добавления клиента

А вот так выглядит уже обновленная страница списка клиентов.

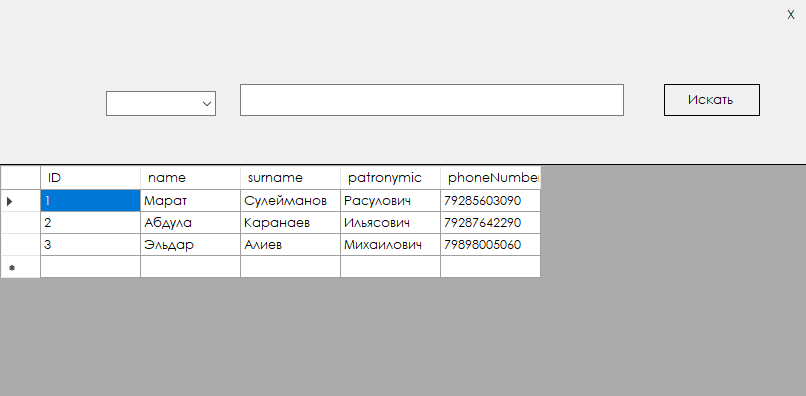


Рисунок 15. Обновленная страница списка клиентов.

Список договоров.

В данном окне хранится информации о всех договорах, продавцах, клиентах и прочем.

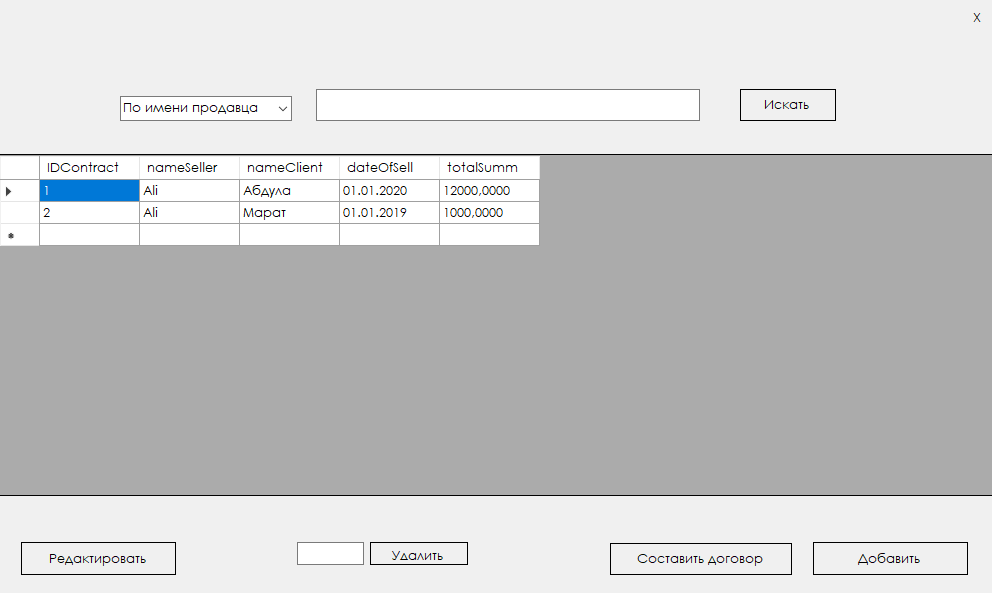


Рис 16. Окно с договорами.

# 

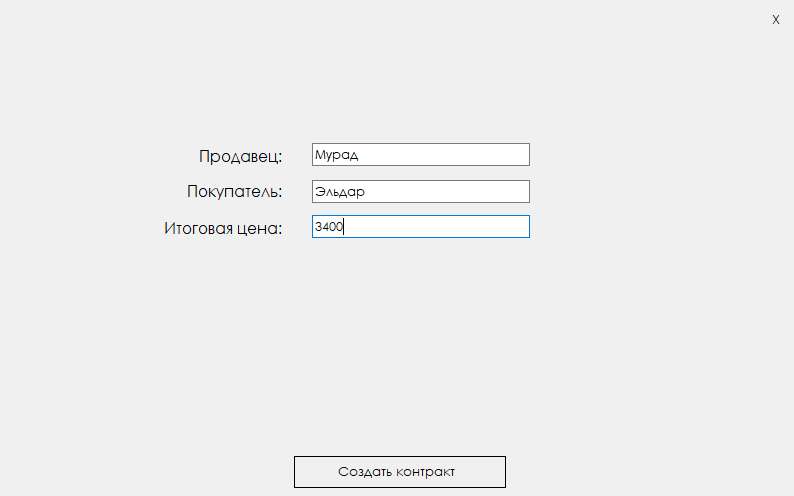
Все окошки, по добавлению и изменению данных выглядят аналогично прошлым, составление же договора выглядит следующим образом.

Рис 17. Внесение информации в таблицу.

Результат на рисунке 18.

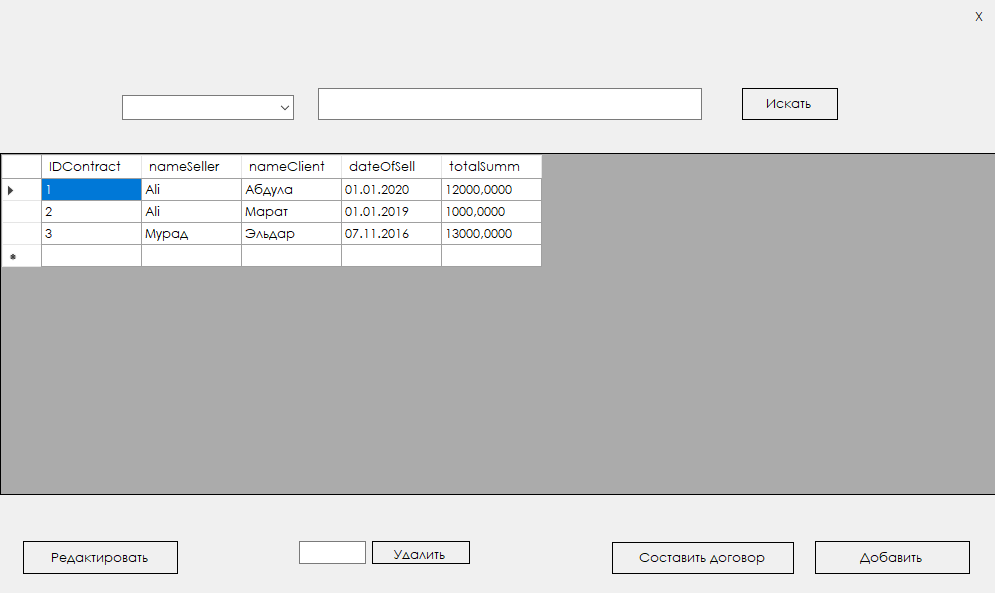


Рис 18. Обновленная страница с данными о контрактах.

# 4. Тестирование

## 4.1 Объект испытаний

Функционал системы, База данных и логистика.

## 4.2 Цель испытаний

Корректность работы системы, правильная обработка запросов от Базы Данных и к Базе Данных.

* 1. **Ход испытаний**

Попробуем ввести в поле авторизации несуществующие данные.

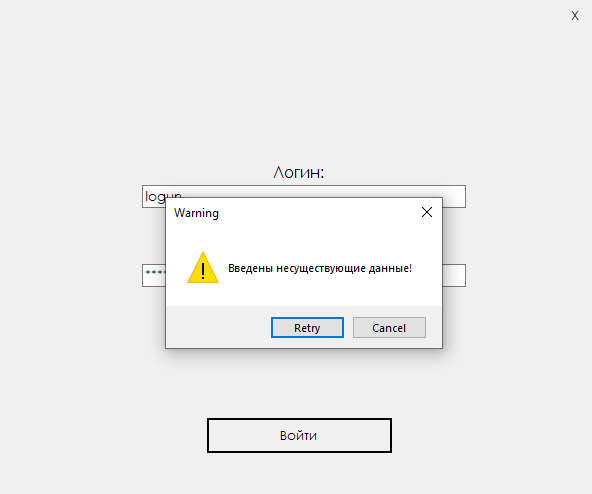


Рис 19. Ошибка авторизации.

При попытке задать новому сотруднику дату, которая является большей, чем сегодняшняя, система даст знать об этом.

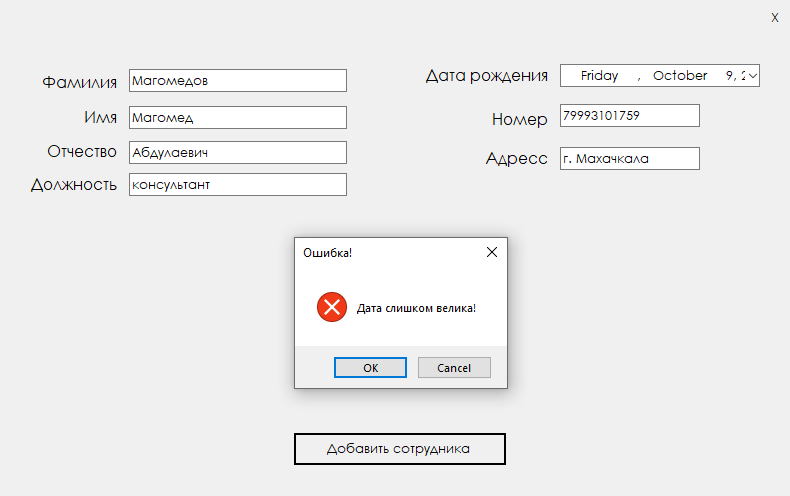


Рис 20. Ошибка даты.

Поле для заполнения номера имеет ограничение в 11 цифр, больше невозможно ввести.

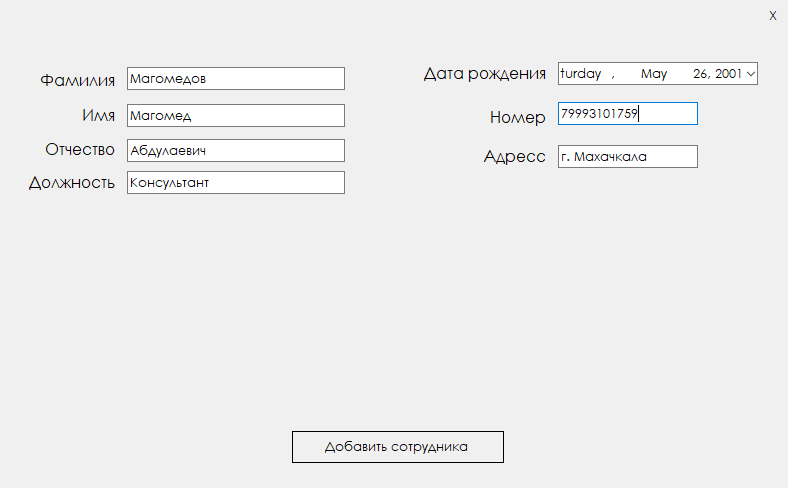


Рис 21. Ограничение номера

При попытке ввести несуществующего производителя, система выдаст ошибку.

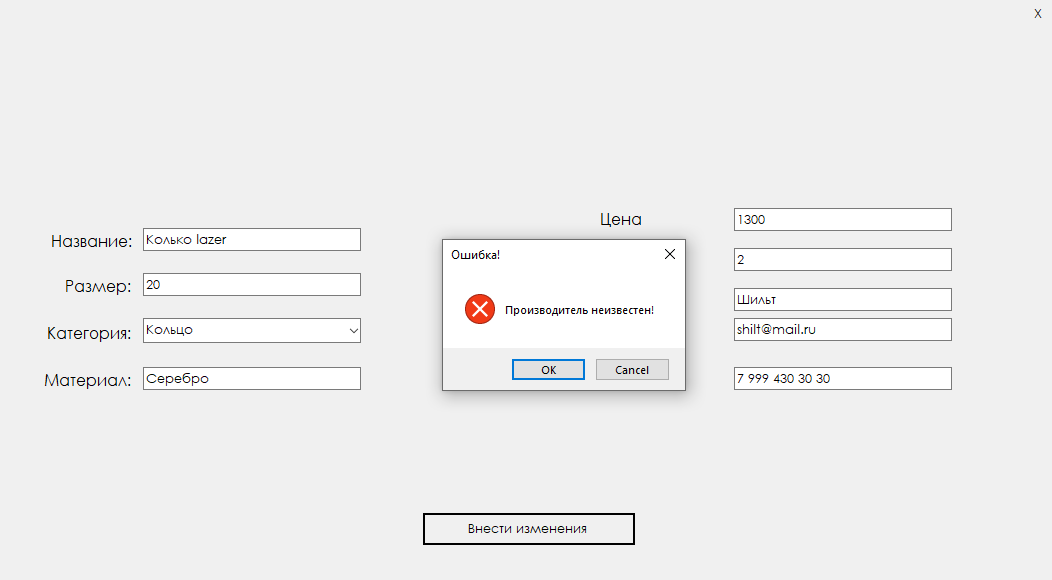


Рис 22. Несуществующий производитель.

# Заключение

Цель работы заключалась в создании программного обеспечения для ювелирного магазина, которое позволяет вести учёт поступления, продаж, регистрацию клиентов, консультирование клиентов и заключать договора.

Было проведено исследование предметной области и проанализирован процесс работы аренды. В процессе изучения документации, наблюдения и опроса сотрудников, были выделены особенности текущего состояния автоматизации в автосалоне, также описаны основные процессы разработки.

На основе этих данных были выработаны рекомендации по автоматизации процесса работы с клиентами, определены цели и задачи автоматизации, сформированы требования к программному продукту, которые учитывают особенности обработки информации

С помощью разработанного ПО автоматизированы:

• ведение базы данных клиентов;

• ведение базы изделий;

• формирование поступлений, наличия и продаж изделий;

• консультирование клиента;

• формирование договоров.

Опытная эксплуатация показала, что разработанный программный продукт удовлетворяет требованиям заказчика по функциональным требованиям и удобному интерфейсу, а также позволяет автоматизировать основные бизнес-процессы.

Таким образом, можно сделать вывод о том, что все задачи выполнены,

цель достигнута.

# Список используемой литературы

1. Джеффери Рихтер., Программирование на платформе Microsoft .NET Framework 4.5 на языке C#, учебник для студ. учреждений сред. проф. Образования/ 4-изд., стер. – М. стер., Издательский центр «Питер».
2. Адам Фримен Джозеф Раттц-мл., LINQ язык интегрированных запросов в C# 2020, для профессионалов, 2-изд., Издательский центр «Apress»
3. Адам Натан, WPF 4, пособное руководство, 1-изд., Издатель: ИМВО
4. Автоматизация и консалтинг: http://www.itone.ru/avtomatizatsiya
5. Автоматизация процессов: <http://www.itone.ru/avtomatizatsiya>
6. Полное руководство по WPF: <https://metanit.com/sharp/wpf/>
7. Паттерн проектирование: <https://metanit.com/sharp/general.php>
8. Базы данных, руководство по Entity Framework 3: <https://metanit.com/sharp/entityframeworkcore/>
9. Руководство по ADO.NET и работа с базами данных: <https://metanit.com/sharp/adonet/>
10. Руководство по Entity Framework: <https://metanit.com/sharp/entityframework/>
11. SQL Server: <https://metanit.com/sql/sqlserver/>
12. Проектирование: <https://metanit.com/sql/tutorial/>
13. Синтаксис по LINQ: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/linq/>

# Приложение A. Техническое задание

## 1. Общие сведения.

### 1.1. Полное наименование системы

Разработка приложения для автоматизации процесса учета товародвижения в ювелирном магазине.

1.2. Наименование предприятия - заказчика системы

ГБПОУ «Технический колледж» имени Р.Н. Ашуралиева

### 1.3. Перечень документов, на основании которых создается система

Основанием для разработки системы является задание на курсовую работу на тему: «Разработка приложения для автоматизации процесса учета товародвижения в ювелирном магазине»

### 1.4. Плановые сроки начала и окончания работы по созданию системы

Система должна быть создана к \_\_\_\_\_ Дата начала проектирования - \_\_\_\_

## 2. Назначение разработки

### 2.1. Назначение разработки

Данное информационная система предназначена для автоматизации товародвижения автосалона по аренде автомобилей, связанной с формированием информационной базы для регистрации автомобилей, клиентов и заелюченных договров и работников данного автосалона.

Информационная система должна автоматизировать работу сотрудника, который занимается регистрацией клиентов, автомобилей и заключением договоров по аренде автомобилей.

### 2.2. Цели создания системы

С использованием информационной системы по автоматизации товародвижения должна повыситься эффективность документооборота предприятия в данной сфере деятельности за счет:

* Уменьшения трудозатрат на прием и систематизацию информации.
* Уменьшения времени поиска по запросам.

## 3. Требования к программе.

### 3.1. Требования к функциональным характеристикам.

Информационная система по автоматизации документооборота отдела охраны труда ОАО «Авиаагрегат» должна обеспечивать выполнение следующих функций:

Ввод, хранение, поиск и обработку информации при проведении инструктажей по технике безопасности и квалификационных испытаний;

Ведение базы данных работников и должностей предприятия;

Своевременное получение информации по запросам.

Первичные данные для занесения в базу – инструкции по технике безопасности, номенклатурный список профессий предприятия, личные документы работников.

Выходными данными является информация из базы данных по запросам в виде создаваемых отчетов.

### 3.2. Требования к надежности:

* Возможность самовосстановления после сбоев (отключение питания, сбои ОС и т.д.);
* Парольную защиту при запуске программы;
* Ограничение несанкционированного доступа к данным;
* Разграничение пользовательских прав;
* Предусмотреть контроль вводимой информации и блокировку некорректных действий пользователя.

### 3.3. Требования к составу и параметрам технических средств.

Для комфортной эксплуатации данной системы, минимальными требованиями к техническим характеристикам является: CPU intel Pentium Gold (2 ядра, 2 потока, 4 Мб кеш памяти), тактовая частота больше 1.2 ГГрц.

* GPU встроенное графическое ядро
* RAM 4 Гб. (DDR 3)
* Жесткий диск на 55 Гб.
* Блок питания 300 Ватт
* Монитор 18 дюймов, с разрешением 1366x720
* Клавиатура
* Мышь

Программа может работать в ОС семейства Windows, например, Windows 2016 Server, Windows 2019 Server или более поздние версии; Windows 10

3.4. Требования к транспортированию и хранению.

Программа предоставляется на лазерном носителе. Программная документация поставляется в электронном и печатном виде.

### 3.5 Специальные требования.

Программное обеспечение должно иметь дружественный интерфейс, рассчитанный на пользователя средней квалификации. Модули программного обеспечения должны предполагать возможность наращивания и быть совместимыми друг с другом. Язык программирования определяется выбором исполнителя, при этом он должен обеспечивать возможность интеграции программного обеспечения с пакетом MS Office 2016.

## 4. Требования к программной документации.

В ходе разработки программы должны быть подготовлены: текст программы, описание программы, программа и методика испытаний, руководство пользователя.

# Приложение Б. Текст программы

Экран авторизации сотрудников, c# код.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CourseD

{

public partial class Form1 : Form

{

string constring = @"Data Source=DESKTOP-QEO06O0;Initial Catalog=DomovskiBase;Integrated Security=True";

public Form1()

{

InitializeComponent();

movePanel();

}

private void btnClose\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button2\_Click(object sender, EventArgs e)

{

autorization();

}

void movePanel()

{

int move = 0, moveX = 0, moveY = 0; ;

PnlUp.MouseDown += (s, e) => { move = 1; moveX = e.X; moveY = e.Y; };

PnlUp.MouseMove += (s, e) => { if (move == 1) SetDesktopLocation(MousePosition.X - moveX, MousePosition.Y - moveY); };

PnlUp.MouseUp += (s, e) => { move = 0; };

}

void autorization()

{

try

{

using (SqlConnection conect = new SqlConnection(constring))

{

string cmd = $"SELECT \* FROM Autorization WHERE [login] = '{textBox1.Text}' and password = '{textBox2.Text}'";

conect.Open();

SqlCommand comand = new SqlCommand(cmd, conect);

SqlDataReader reader = comand.ExecuteReader();

if (reader.HasRows)

{

reader.Read();

if (reader.GetValue(1).ToString() == $"{textBox1.Text}" && reader.GetValue(2).ToString() == $"{textBox2.Text}")

{

if (reader.GetValue(3).ToString() == "admin")

{

adminPage admPage = new adminPage();

Hide();

admPage.Show();

}

else if (reader.GetValue(3).ToString() == "worker")

{

workerPage workPage = new workerPage();

workPage.Show();

Hide();

}

else

MessageBox.Show("This invalid dates");

}

}

else

{

MessageBox.Show("Неверный логин или пароль.", "404", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Exception", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

Окно администратора, C# код

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CourseD

{

public partial class adminPage : Form

{

string constring = @"Data Source=DESKTOP-QEO06O0;Initial Catalog=DomovskiBase;Integrated Security=True";

public adminPage()

{

InitializeComponent();

movePanel();

dataFull();

}

private void btnClose\_Click(object sender, EventArgs e)

{

Application.Exit();

}

// Поиск

private void button4\_Click(object sender, EventArgs e)

{

string sel = $"SELECT \* FROM Workers WHERE [name] = '{textBox1.Text}'";

if (textBox1.Text == "")

{

MessageBox.Show("Поле является пустым, пожалуйста, введите данные для поиска!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

dataFull();

} else if (textBox1.Text != "")

sqlComando(sel);

MessageBox.Show("Данные отсутствуют! Проверьте на корректность веденные данные!", "Error!", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);

}

// Страница для добавления данных

private void button1\_Click(object sender, EventArgs e)

{

addWorker add = new addWorker();

add.Show();

}

// Страница для изменения данных

private void button2\_Click(object sender, EventArgs e)

{

updatePage uPage = new updatePage();

uPage.Show();

}

// Кнопка удаления данных

private void button3\_Click(object sender, EventArgs e)

{

string del = $"DELETE FROM Workers WHERE [ID] = '{textBox2.Text}'";

deleteDates(del);

}

#region

// Для заполнения таблицы данными.

void dataFull()

{

string sel = $"SELECT \* FROM Clients";

using (SqlConnection conet = new SqlConnection(constring))

{

conet.Open();

SqlCommand comand = new SqlCommand(sel, conet);

SqlDataAdapter adapter = new SqlDataAdapter(comand);

DataTable table = new DataTable();

adapter.Fill(table);

DGV.DataSource = table;

}

}

// Движение панели

void movePanel()

{

int move = 0, moveX = 0, moveY = 0; ;

PnlUp.MouseDown += (s, e) => { move = 1; moveX = e.X; moveY = e.Y; };

PnlUp.MouseMove += (s, e) => { if (move == 1) SetDesktopLocation(MousePosition.X - moveX, MousePosition.Y - moveY); };

PnlUp.MouseUp += (s, e) => { move = 0; };

}

// Команда для вывода данных поиска

void sqlComando (string sel)

{

try

{

using (SqlConnection conet = new SqlConnection(constring))

{

conet.Open();

SqlCommand comand = new SqlCommand(sel, conet);

SqlDataAdapter adapter = new SqlDataAdapter(comand);

DataTable table = new DataTable();

adapter.Fill(table);

DGV.DataSource = table;

MessageBox.Show("Данные выведены!", "Успешно", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Исключение!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

// Удаление данных

void deleteDates (string cmd)

{

try

{

using (SqlConnection conet = new SqlConnection(constring))

{

conet.Open();

SqlCommand comand = new SqlCommand(cmd, conet);

comand.ExecuteNonQuery();

MessageBox.Show("Данные Удалены!", "Успешно", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Исключение!", MessageBoxButtons.OK, MessageBoxIcon.Error);

} finally

{

dataFull();

}

}

#endregion

}

}

Окно сотрудника-консультанта, с# код

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CourseD

{

public partial class workerPage : Form

{

string constring = @"Data Source=DESKTOP-QEO06O0;Initial Catalog=DomovskiBase;Integrated Security=True";

public workerPage()

{

InitializeComponent();

dataFull();

movePanel();

}

private void btnClose\_Click(object sender, EventArgs e)

{

Application.Exit();

}

#region voids

void movePanel()

{

int move = 0, moveX = 0, moveY = 0; ;

PnlUp.MouseDown += (s, e) => { move = 1; moveX = e.X; moveY = e.Y; };

PnlUp.MouseMove += (s, e) => { if (move == 1) SetDesktopLocation(MousePosition.X - moveX, MousePosition.Y - moveY); };

PnlUp.MouseUp += (s, e) => { move = 0; };

}

void dataFull()

{

string sel = $"SELECT [name], size, categories, material, price, presence, nameProducer, mail, phoneNumber FROM Products, Producer";

using (SqlConnection conet = new SqlConnection(constring))

{

conet.Open();

SqlCommand comand = new SqlCommand(sel, conet);

SqlDataAdapter adapter = new SqlDataAdapter(comand);

DataTable table = new DataTable();

adapter.Fill(table);

DGV.DataSource = table;

}

}

#endregion

private void btnAdd\_Click(object sender, EventArgs e)

{

addJewerly adJew = new addJewerly();

adJew.Show();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Contract contract = new Contract();

contract.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

}

}

}

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace CourseD

{

public partial class addWorker : Form

{

string constring = @"Data Source=DESKTOP-QEO06O0;Initial Catalog=DomovskiBase;Integrated Security=True";

public addWorker()

{

InitializeComponent();

movePanel();

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

string cmd = $"INSERT INTO Workers ([name], [surname], patronymic, post, born, telephone, [address])VALUES('{txtName.Text}', '{txtSurname.Text}', '{txtPatronymic.Text}', '{txtPost.Text}', '{dateTimePicker1.Value.ToString()}', '{textBox1.Text}', '{txtAddress.Text}')";

MessageBox.Show("Дата слишком велика!", "Ошибка!", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);

//sqlComando(cmd);

}

void sqlComando(string cmd)

{

try

{

using (SqlConnection conet = new SqlConnection(constring))

{

conet.Open();

SqlCommand comand = new SqlCommand(cmd, conet);

comand.ExecuteNonQuery();

MessageBox.Show("Данные Добавлены!", "Успешно", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Исключение!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

void movePanel()

{

int move = 0, moveX = 0, moveY = 0; ;

PnlUp.MouseDown += (s, e) => { move = 1; moveX = e.X; moveY = e.Y; };

PnlUp.MouseMove += (s, e) => { if (move == 1) SetDesktopLocation(MousePosition.X - moveX, MousePosition.Y - moveY); };

PnlUp.MouseUp += (s, e) => { move = 0; };

}

}

}

Окно добалвения новых сотрдуников, XAML код

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace CourseD

{

public partial class addWorker : Form

{

string constring = @"Data Source=DESKTOP-QEO06O0;Initial Catalog=DomovskiBase;Integrated Security=True";

public addWorker()

{

InitializeComponent();

movePanel();

}

private void button1\_Click(object sender, EventArgs e)

{

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

string cmd = $"INSERT INTO Workers ([name], [surname], patronymic, post, born, telephone, [address])VALUES('{txtName.Text}', '{txtSurname.Text}', '{txtPatronymic.Text}', '{txtPost.Text}', '{dateTimePicker1.Value.ToString()}', '{textBox1.Text}', '{txtAddress.Text}')";

MessageBox.Show("Дата слишком велика!", "Ошибка!", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);

//sqlComando(cmd);

}

void sqlComando(string cmd)

{

try

{

using (SqlConnection conet = new SqlConnection(constring))

{

conet.Open();

SqlCommand comand = new SqlCommand(cmd, conet);

comand.ExecuteNonQuery();

MessageBox.Show("Данные Добавлены!", "Успешно", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Исключение!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

void movePanel()

{

int move = 0, moveX = 0, moveY = 0; ;

PnlUp.MouseDown += (s, e) => { move = 1; moveX = e.X; moveY = e.Y; };

PnlUp.MouseMove += (s, e) => { if (move == 1) SetDesktopLocation(MousePosition.X - moveX, MousePosition.Y - moveY); };

PnlUp.MouseUp += (s, e) => { move = 0; };

}

}

}

Логика экрана отображения данных Автомобилей, C# код

# Приложение В. Пользовательское руководство

Для работы в системе, вам потреубется запросить доступ у Администратора системы. Администратор зарегистрирует вас в системе и даст вам данные необходимые для авторизации.

После получения доступа и авторизации, вы имеете полный доступ всем функциям системы, первым экраном после авторизации будет стартовый экран вашей рабочей площадки, где вам надо будет выбрать действие.

Перед вами предстанет панель с данными о товарах. Вы можете выполнять некие администрационные манипуляции, такие как Удаление данных, поиск, добавление и редактировка. Выбрав «Добавление товара» вам предстанет окошко, где нужно будет ввести все необходимые данные о товаре. Выбрав «редактировать данные» перед вами откроется окошко, где вам нужно будет ввести ID товара, который вы хотите изменить, а затем ввести новые данные.

Это было работа с товаром, давайте рассмотрим, как можно будет работать с клиентами. При работе с клиентами, вам надо будет зарегистрировать нового клиента, после чего можно будет заключать с ним договор. Так же доступно редактирование данных клиента и их удаление.

«Контракты» - это окно, где вы можете заключить договор с зарегистрированным клиентом, просмотреть имеющиеся договора. Так же вы можете удалять и редактирвоать договоры.