Государственное бюджетное профессиональное образовательное учреждение

Самарской области

«Тольяттинский социально-экономический колледж»

**КУРСОВАЯ РАБОТА**

**РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

**«УЧЁТ СОТРУДНИКОВ ПРЕДПРИЯТИЯ»**

**ПМ.05 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ**

**МДК 05.02 Разработка кода информационных систем**

**09.02.07 Информационные системы и программирование**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  | **/** | М.В. Немов |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |
| Оценка выполнения и защиты курсовой работы | | |  |
|  | | |  |
| Руководитель |  | **/** | Е.В. Плюснина |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |

Тольятти, 2022

Государственное бюджетное профессиональное образовательное учреждение  
Самарской области  
«Тольяттинский социально-экономический колледж»

|  |  |  |
| --- | --- | --- |
|  |  | Утверждаю:  Заместитель директора по УР  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.С. Киронова  *« » 2022 г.* |

**ЗАДАНИЕ**

на курсовую работу

**по ПМ.05 Проектирование и разработка информационных системмодуля, выполняемой в рамках МДК.05.02 Разработка кода информационных систем**

студента группы ИСП-31

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Немова Макария Владимировича\_\_\_\_\_\_\_\_\_\_\_\_\_

*Фамилия Имя отчество студента*

Тема курсовой работы: «Разработка информационной системы **«**Учёт сотрудников предприятия»

1. Содержание задания:

1.1 Разработать физическую структуру информационной системы.

1.2 Разработать запросы к информационной системе.

1.3 Разработать приложение.

1. Исходные данные:

Исходные данные для практической реализации автоматизированной информационной системы (АИС) берутся из различных информационных источников (Интернет-ресурсы, печатные издания, периодика и др.).

1. Содержание курсовой работы

Введение

1 Разработка базы данных информационной системы (название)

1.1 Разработка физической структуры БД

1.2 Разработка запросов ИС

2 Разработка интерфейса информационной системы (название)

Заключение

Список использованных источников

Приложения

Приложение А Листинг программы

Дата выдачи задания: «13» января 2022 г.

Дата сдачи работы на отделение: «\_\_»\_\_\_\_\_\_\_\_2022 г.

Руководитель курсового(ой) проекта(работы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Е.В. Плюснина

подпись расшифровка подписи

**КАЛЕНДАРНЫЙ ПЛАН**

выполнения курсовой работы

Студентом 3 курса группы ИСП-31

По теме Разработка и проектирование ИС «Учёт сотрудников предприятия»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  этапа  работы | Содержание этапов работы | Плановый срок выполнения этапа | Планируемый объем выполнения  этапа, % | Отметка  о  выполнении  этапа |
| 1 | Выбор, обоснование темы и объекта исследования | Январь 2022 | 5% |  |
| 2 | Утверждение темы, согласование плана. Введение, библиография | Январь 2022 | 10% |  |
| 3 | Изучение и анализ информационных материалов по теме | Февраль 2022 | 15% |  |
| 4 | Обоснование актуальности выбранной темы применительно к профессиональной деятельности (введение) | Февраль 2022 | 20% |  |
| 5 | Изложение материала основной части по теме курсовой работы | Февраль 2022 | 20% |  |
| 6 | Подведение итогов проведенного анализа, формулировка выводов УИР применительно к профессиональной деятельности (заключение) | Март 2022 | 20% |  |
| 7 | Оформление работы и сдача на проверку | Март 2022 | 10% |  |
| 8 | Защита работы |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  | **/** | М.В. Немов |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |
| Руководитель |  | **/** | Е.В. Плюснина |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |

**СОДЕРЖАНИЕ**

[1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТ СОТРУДНИКОВ ПРЕДПРИЯТИЯ 6](#_Toc101479012)

[1.1 Разработка физической структуры БД 6](#_Toc101479013)

[1.2 Разработка запросов ИС 9](#_Toc101479014)

[2 РАЗРАБОТКА ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЁТ СОТРУДНИКОВ ПРЕДПРИЯТИЯ» 12](#_Toc101479015)

[2.1 Разработка интерфейса 12](#_Toc101479016)

[2.2 Разработка руководства пользователя 17](#_Toc101479017)

[ЗАКЛЮЧЕНИЕ 21](#_Toc101479018)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc101479019)

[ПРИЛОЖЕНИЕ А. ЛИСТИНГ ПРОГРАММЫ 24](#_Toc101479020)

# ВВЕДЕНИЕ

В настоящее время учёт сотрудников на предприятие и ведение кадрового делопроизводства является обязательным условием эффективного управления персоналом любой компании. Неукоснительного соблюдения требований к документам по учету личного состава требует действующее гражданское, налоговое и трудовое законодательство. Кроме этого, кадровое делопроизводство является немаловажной составляющей трудовых взаимоотношений, оно официально отражает отношения работодателя и наемного работника.

Кадры — это совокупность работников различных профессионально - квалификационных групп, занятых на предприятии и входящих в его списочный состав.

Сегодня множество людей открывают для себя необходимость и важность использования приложения для учета сотрудников предприятия в производстве и организации деятельности. Одним из основных элементов такого механизма является разработка приложения для учета сотрудников предприятия, а также бизнес-процессов, которые не только отражают функциональную и информационную деятельность, но и воздействуют на нее.

Целью курсовых работ является создание информационной системы «Учёт сотрудников предприятия». Для достижения поставленной цели были определены следующие задачи:

* описать предметную область;
* спроектировать программный комплекс;
* выполнить программную реализацию;
* проанализировать качество программного обеспечения.

В данной курсовой работе внимание будет уделено программной реализации информационной системы «Учёт сотрудников предприятия». Будет организовано подключение к БД и разработан интерфейс программного продукта.

# 1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТ СОТРУДНИКОВ ПРЕДПРИЯТИЯ

## 1.1 Разработка физической структуры БД

В начале необходимо создать новую базу данных, откроем SQL Server Management Studio. Нажмем на назначение сервера в окне Object Explorer и в появившемся меню выберем пункт Новый запрос. В среде SQL Server Management Studio создаем базу данных с помощью графического интерфейса.

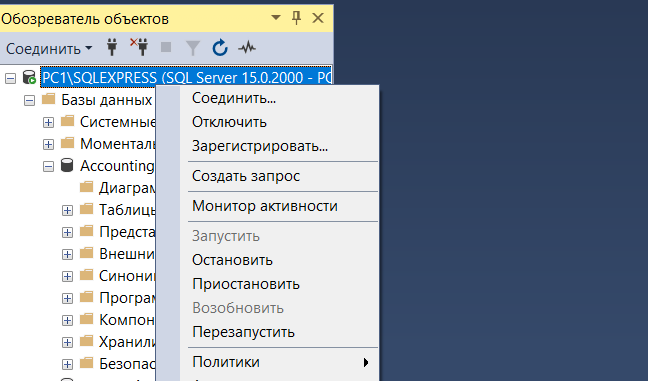
****

Рисунок 1 – Окно Object Explorer

В обозревателе объектов открываем контейнер «Базы данных», затем открываем нужную базу данных и щелкаем правой кнопкой мыши по пункту «Таблицы», и выбираем «Таблица».

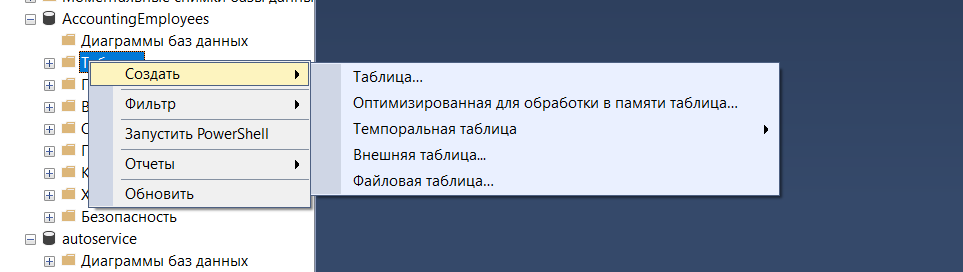


Рисунок 2 – Окно Object Explorer

Введите имена столбцов, выберите типы данных и определите для каждого столбца, могут ли в нем присутствовать значения NULL, как показано на следующей иллюстрации:

Таблицы создаются в рамках текущей базы данных.

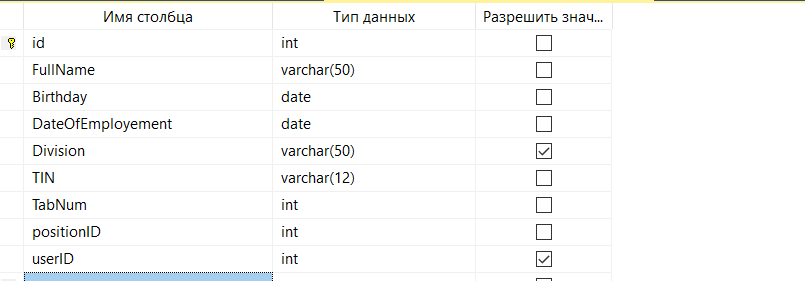
**

Рисунок 3 – Таблица «Сотрудники»

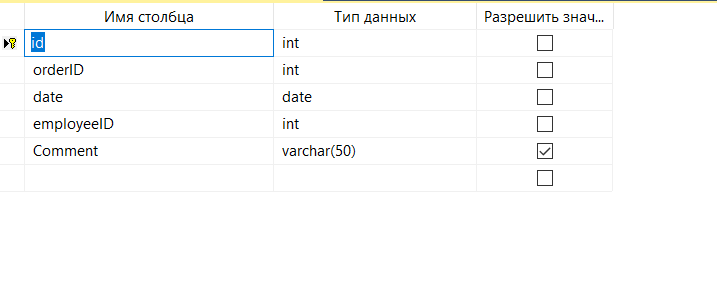


Рисунок 4 - Таблица «Приказы»

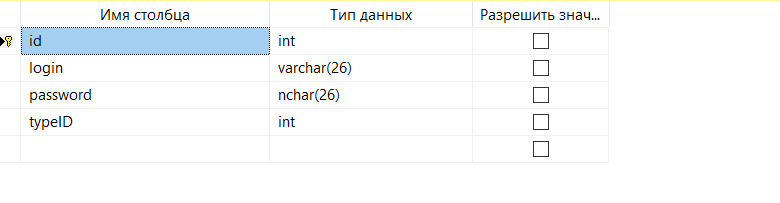
**

Рисунок 5 - Таблица «Пользователи»

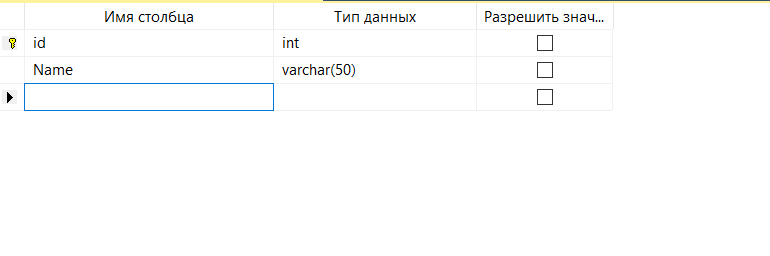


Рисунок 6 - Таблица «Тип приказа»

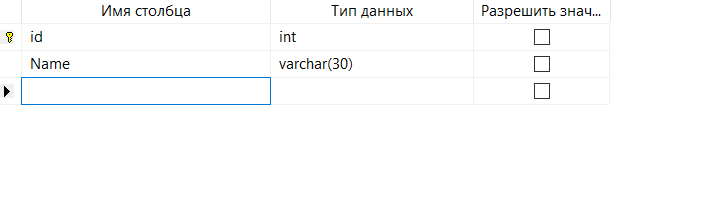
**

Рисунок 7 – Таблица «Тип пользователя»

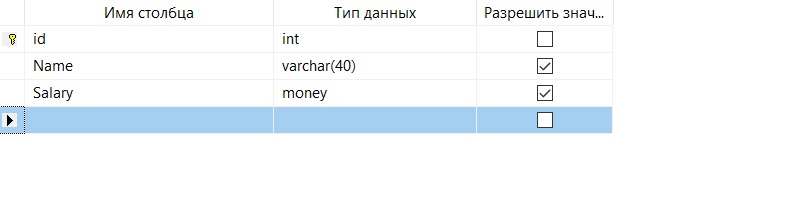


Рисунок 8 – Таблица «Должности»

Вы также можете задать другие свойства столбца, например, является ли этот столбец столбцом идентификаторов или вычисляемым столбцом. Для этого щелкните столбец на вкладке свойств столбцов.

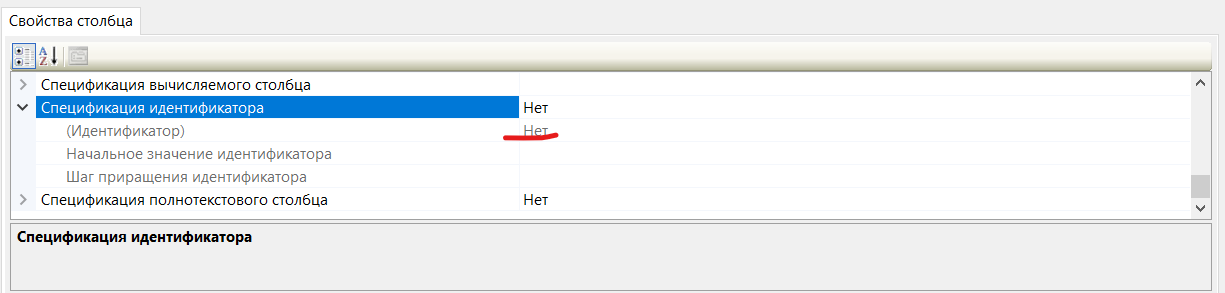


Рисунок 9 – Свойства столбцов

Чтобы указать, что столбец является столбцом первичного ключа, щелкните его правой кнопкой мыши и выберите З**адать первичный ключ**.

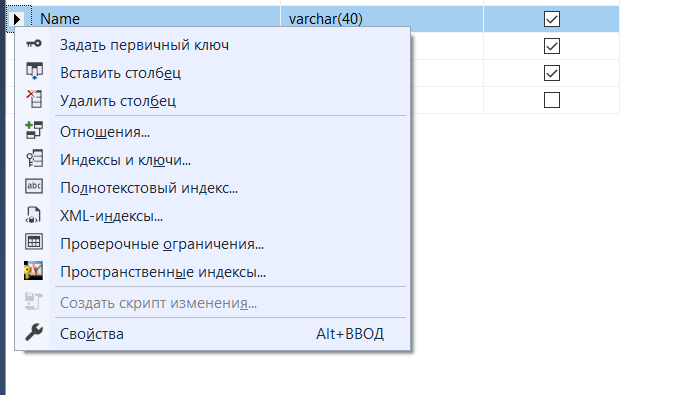


Рисунок 10 – Назначение первичного ключа

Чтобы создать связи по внешнему ключу, проверочные ограничения или индексы, щелкните правой кнопкой мыши панель конструктора таблиц и выберите в списке объект, как показано на следующей иллюстрации:

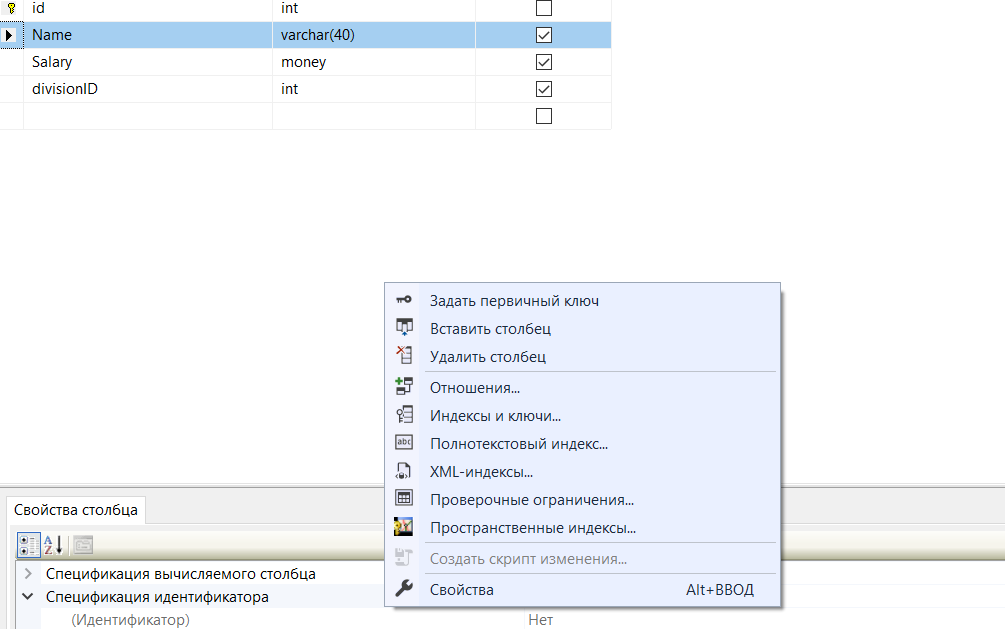


Рисунок 11 – Назначение первичного ключа

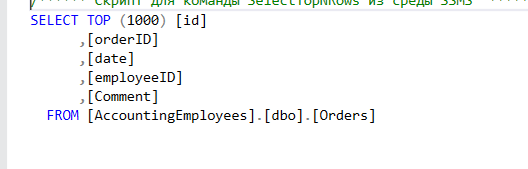
**

Рисунок 12 - Запрос приказы

## 1.2 Разработка запросов ИС

Все операции, которые вы можете выполнять с данными, следуют аббревиатуре CRUD.

CRUD обозначает 3 основные операции, которые мы выполняем при запросе базы данных: **C**reate (создание), Update (обновление) и **D**elete (удаление).

Для получения данных применяется команда SELECT. Рассмотрим на примере таблицы «Должности»:

GO

USE [AccountingEmployees]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Positions] Script Date: 21.04.2022 15:36:08 \*\*\*\*\*\*/

IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Positions]') AND type in (N'U'))

DROP TABLE [dbo].[Positions]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Positions] Script Date: 21.04.2022 15:36:08 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Positions](

[id] [int] IDENTITY(1,1) NOT NULL,

[Name] [varchar](40) NULL,

[Salary] [money] NULL,

[divisionID] [int] NULL,

CONSTRAINT [PK\_Positions] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

Для изменения уже имеющихся строк в таблице применяется команда UPDATE.

Также рассмотрим на примере таблицы «Должности»:

USE [AccountingEmployees]

GO

UPDATE [dbo].[Positions]

SET [Name] = <Name, varchar(40),>

,[Salary] = <Salary, money,>

,[divisionID] = <divisionID, int,>

WHERE <Условия поиска,,>

GO

USE [AccountingEmployees]

GO

Оператор DELETE удаляет строки из временных или постоянных базовых таблиц, представлений или курсоров, причем в двух последних случаях действие оператора распространяется на те базовые таблицы, из которых извлекались данные в эти представления или курсоры.

Рассмотрим на примере таблицы «Приказы»:

DELETE FROM [dbo].[Orders]

WHERE <Условия поиска,,>

GO

# 2 РАЗРАБОТКА ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЁТ СОТРУДНИКОВ ПРЕДПРИЯТИЯ»

## 2.1 Разработка интерфейса

Пользовательский интерфейс (user interface или сокращенно UI) – это интерфейс, с помощью которого человек может управлять программным обеспечением или аппаратным оснащением. UI должны быть удобными в использовании, чтобы взаимодействие с ними происходило на максимально интуитивном уровне. Интерфейсы программного обеспечения также называют графическими пользовательскими интерфейсами (graphical user interface или GUI).

Первым шагом к созданию пользовательского интерефейса является создание инфраструктуры приложения.

Чтобы создать новый проект приложения WPF в Visual Studio первым делом после запуска приложения в открывшимся окне выбрать кнопку «Создание проекта»:

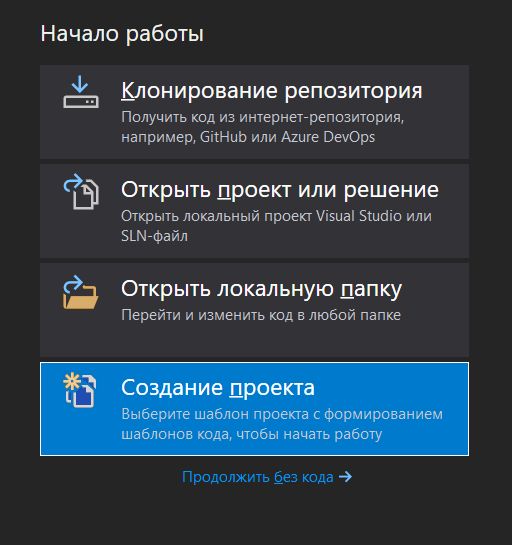


Рисунок 13 – Создание проекта

В выведенном окне вводим в поиске WPF и ищем приложение WPF (.Net.Framework), создаём проект и нажимаем далее

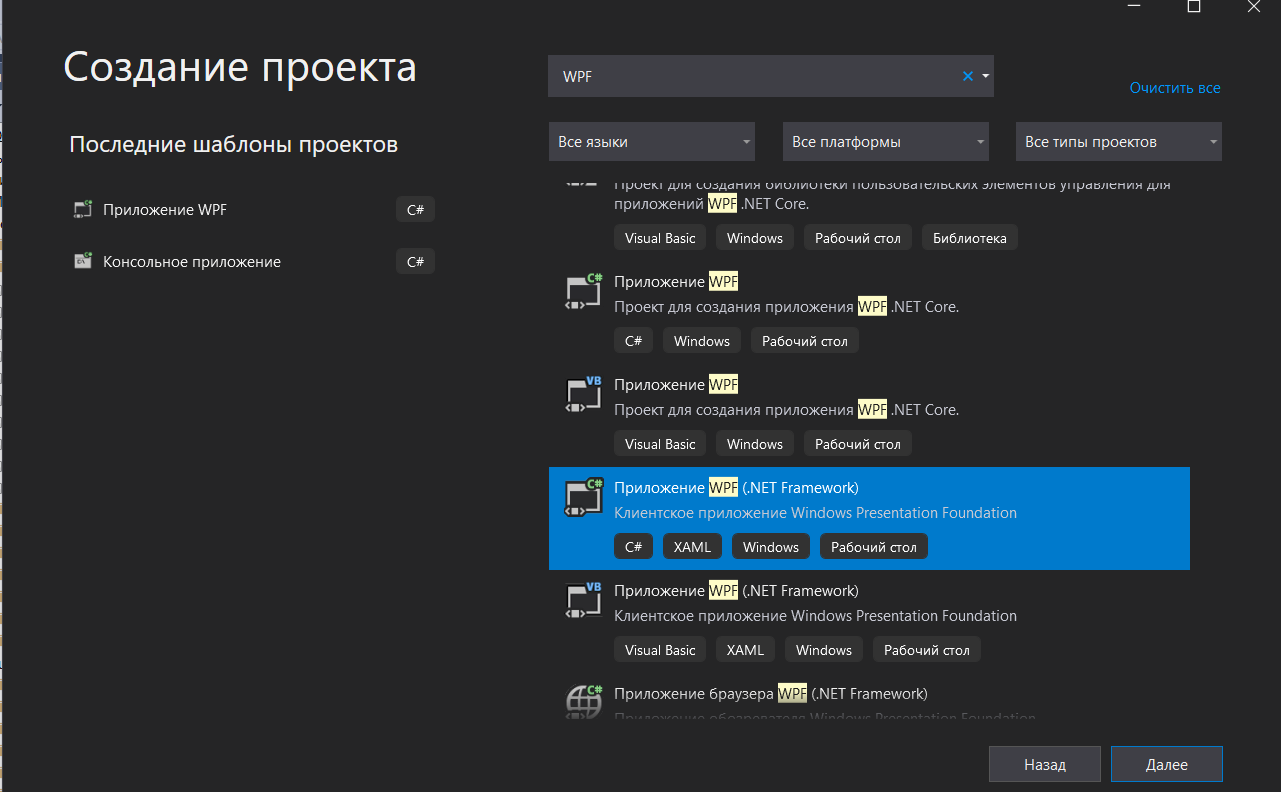


Рисунок 14 – Создание проекта

Visual Studio создает проект и открывает конструктор для окна приложения по умолчанию с именем MainWindow.xaml

Для вывода данных о сотрудниках нужен специализированный интерфейс и элементы, которые позволят работать с данными.

Прежде всего стоит определится с элементами, которые помогут сделать разметку изначального холста. Больший интерес представляют такие элементы как – Grid, Border, StackPanel – они позволят более органично и без неровностей устанавливать элементы управления окна.

Также не меньший интерес представляет элемент DataGrid – таблица с помощью которой будет выводиться информация с базы данных.

Код XAML для окна списка сотрудников выглядит следующим образом:

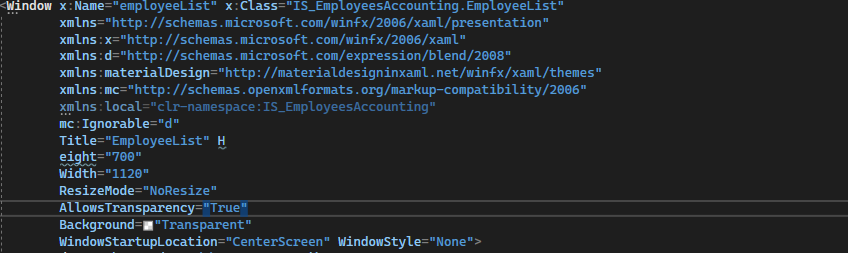


Рисунок 15– XAML - Код окна списка сотрудников

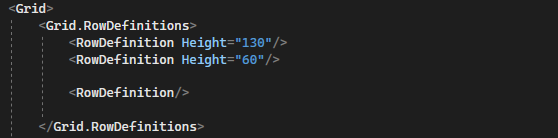


Рисунок 16 – XAML - Код Grid

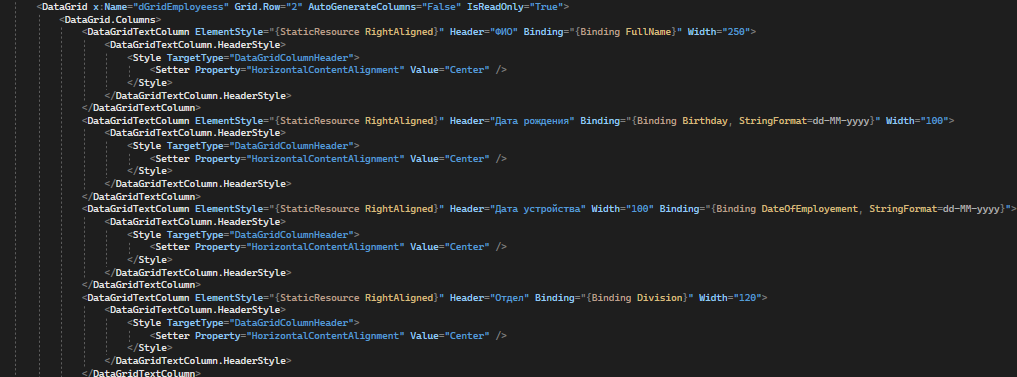


Рисунок 17 – XAML - Код DataGrid

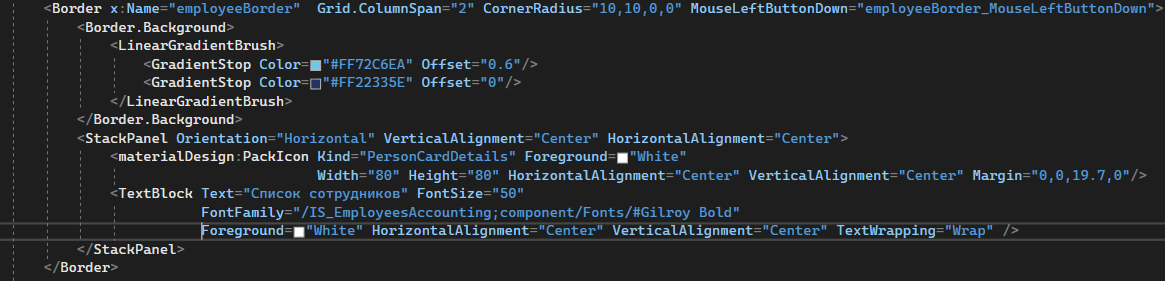


Рисунок 18 – XAML – Код Border с внутренним StackPanel

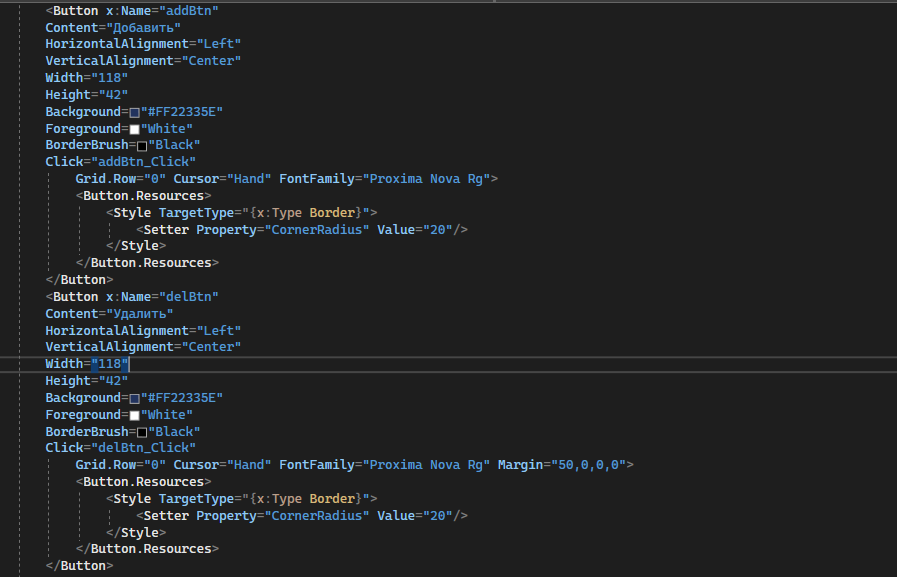


Рисунок 19 – XAML – Код кнопки «Добавить и удалить»

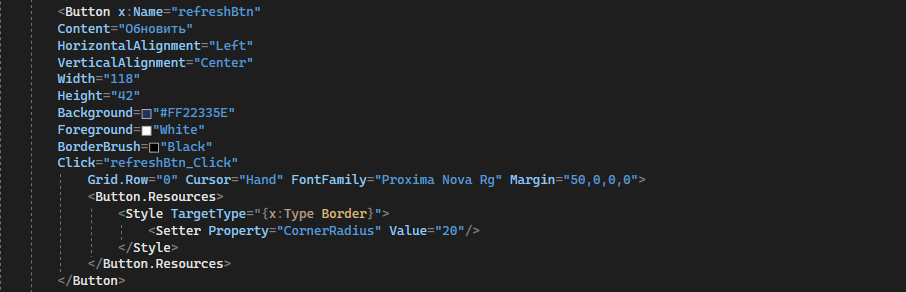


Рисунок 20 – XAML – Код кнопки «Обновить»

В конечном итоге окно должно иметь следующий вид:

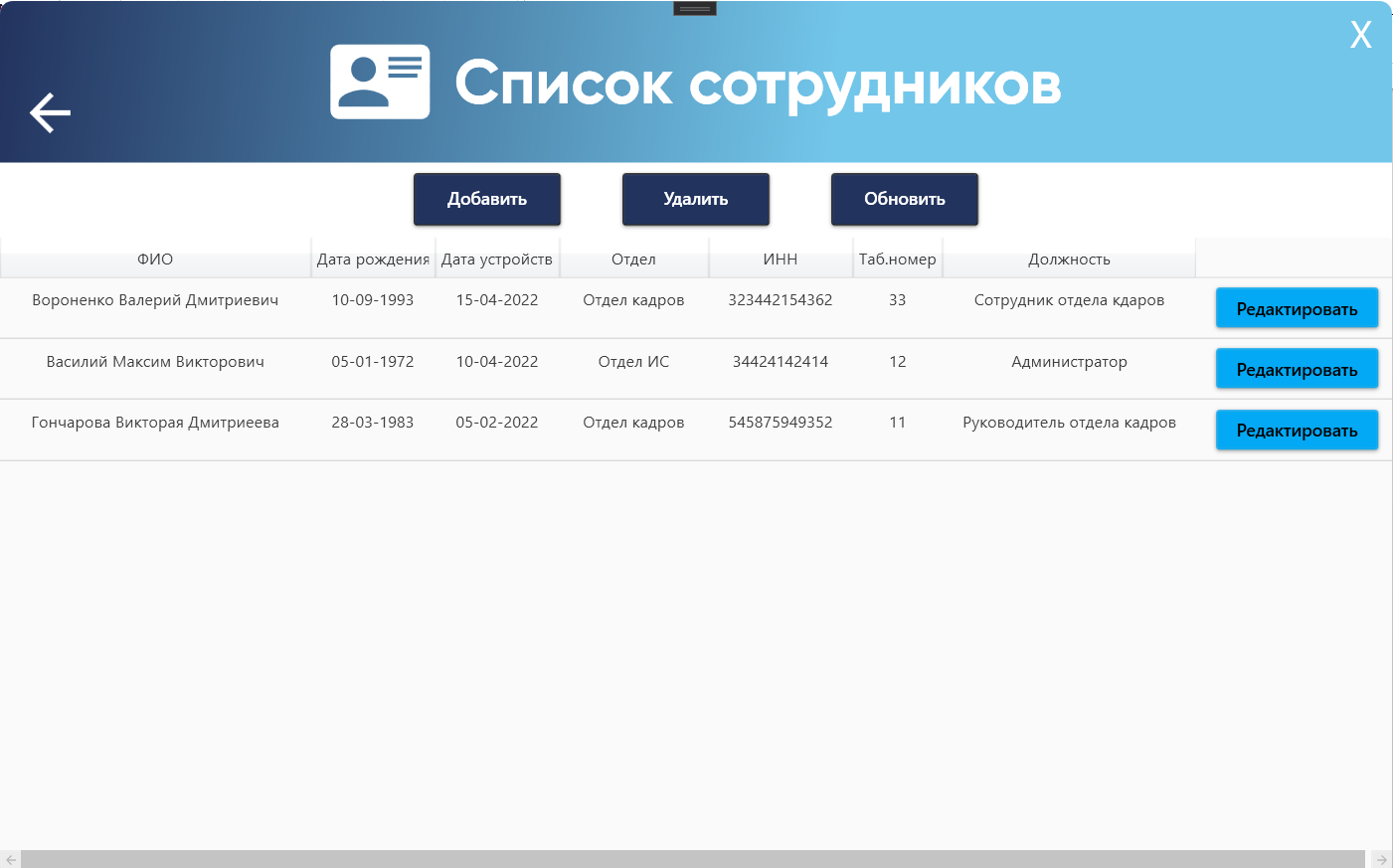


Рисунок 21– Окно списка сотрудников

Затем формируем код данного окна, в первую очередь понадобиться сделать вывод данных с готовой базы данных непосредственно в саму таблицу. Для данной цели используется параметр .ItemSource, который будет получать GetContext с именем нужной таблицы.

Выглядеть код будет следующим образом:

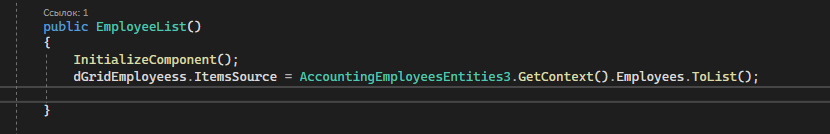


Рисунок 22– Код для вывода в DataGrid

Дальше пишется код к кнопкам окна, например в «Добавить» пишется вызов нового окна с помощью, которого будет происходить процесс заполнения данных таблицы:

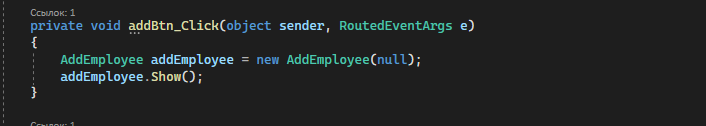


Рисунок 23– Код кнопки «Добавить»

Для кнопки «Удалить» пишется код позволяющий удалить одну или несколько выделенных строк, не только с элемента DataGrid, но и непосредственно с самой базы данных. Код выглядит следующим образом:

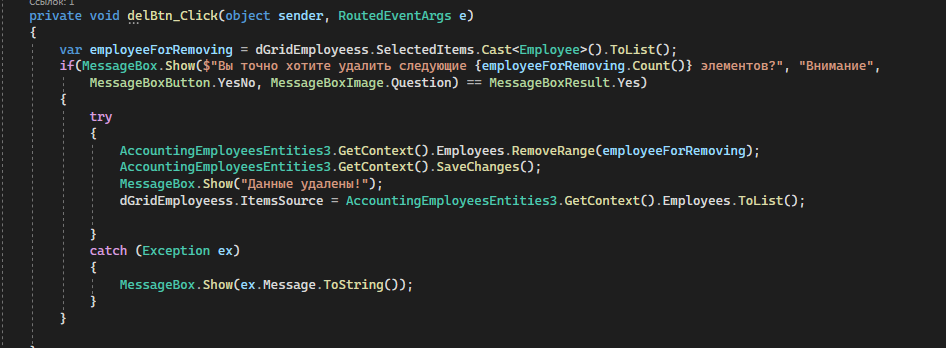


Рисунок 24 – Код кнопки «Удалить»

Также добавляем пару строчек в кнопку «Обновить»:

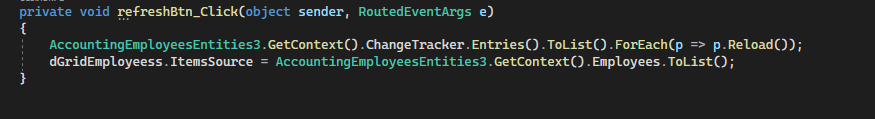


Рисунок 25– Код кнопки «Обновить»

Кнопка позволит обновлять и перезагружать данные с таблицы базы данных в таблицу DataGrid.

Путем подобных манипуляций с кодом и интерфейсом получается работоспособное окно позволяющее просматривать вносимую информацию о сотрудниках предприятия

Интерфейс всех остальных окон разрабатывается аналогичным образом.

## 2.2 Разработка руководства пользователя

Информационная система «Учёт сотрудников предприятия» представляет собой программный комплекс для ознакомления заказчиков и потенциальных клиентов с деятельностью и юридической информацией об организации ООО «Агентство недвижимости «Высота»».

Данное руководство предназначено для пользователей приложения, установленного на устройствах, работающих под управлением операционной системы Windows 10.

Документ «Руководство Пользователя» предназначен для конечных пользователей продукта, администраторов системы и сотрудников отдела кадров.

Для установки приложения компьютер конечного пользователя должен удовлетворять требованиям.

* Минимальная конфигурация сервера:
* тип процессора – Ryzen 7 2700 и выше;
* объем оперативного запоминающего устройства 8Гб и более;
* жесткий диск 80 Гб;
* модем, для выхода в Интернет;
* монитор, клавиатура, мышь.

Требования, предъявляемые к конфигурации клиентских станций:

* процессор – Ryzen 3 или i3 нового поколения и выше;
* 4Гб оперативной памяти;
* монитор – VGA, DVI-D, HDMI;
* клавиатура - 101/102 клавиши;
* манипулятор типа «мышь».

Для реализации данной системы требуется установка следующего ПО: Microsoft Visual Studio 2022 и СУБД Microsoft SQL Server 2018 Express Edition.

При открытии программного обеспечения авторизуйтесь в системе путем ввода логина и пароля в соответствующие поля, затем нажмите кнопку «Войти». Для проверки правильности введенных данных предусмотрена кнопка «Показать пароль» с правой стороны от поля «Пароль».

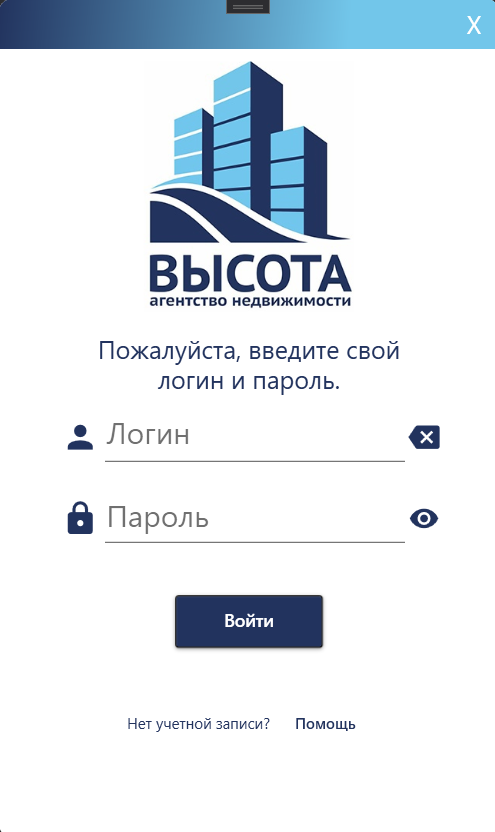


Рисунок 26 – Окно авторизации

Также при необходимость имеется возможность быстрого удаления введенного логина путем нажатия на кнопку «Очистить» с правой стороны от поля «Логин».

Для закрытия приложения на стадии авторизации в любой момент есть возможность нажатия на кнопку «Закрыть» представленная в виде крестика с правой верхней стороны.

При правильно введенном логине и пароле система выдаст сообщение об удачной авторизации и переведет вас в основное меню программы, в противном же случае выдаст ошибку, которая указывает на ошибку.

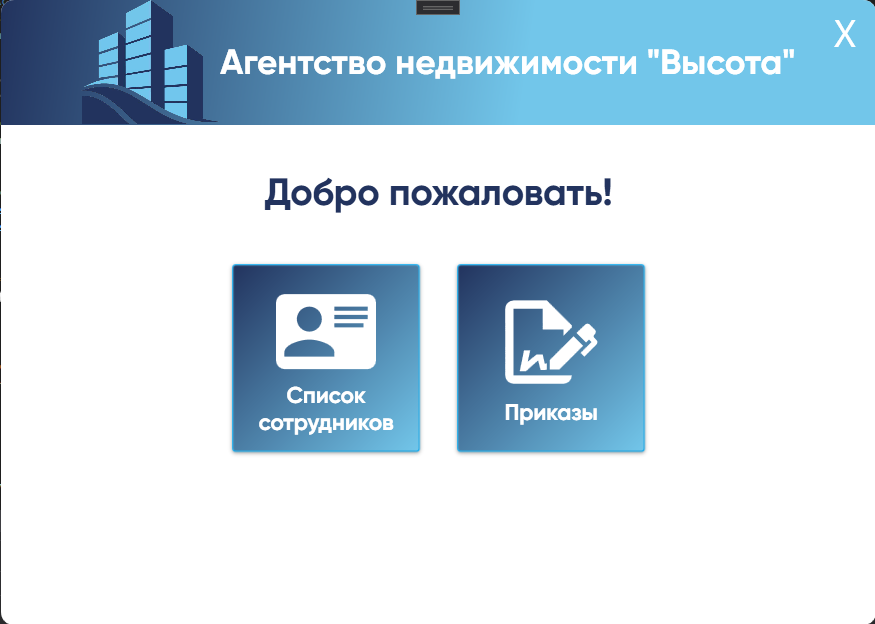


Рисунок 27 – Окно основного меню

В основном меню имеется два раздела на выбор пользователя программы:

* список сотрудников (показывает информацию о добавленных сотрудниках и дает возможность удалить или изменить информацию о них);
* приказы (показывает список подписанных приказов, а также позволяет удалять или добавлять их).

Выбрав в основном меню «Список сотрудников» выполняется переход в новое окно, при появлении показывается таблица, содержащая информацию о учтённых сотрудниках. В любой момент нажав на кнопку «Назад» представленную в виде стрелочки влево, можно вернуться в основное меню.

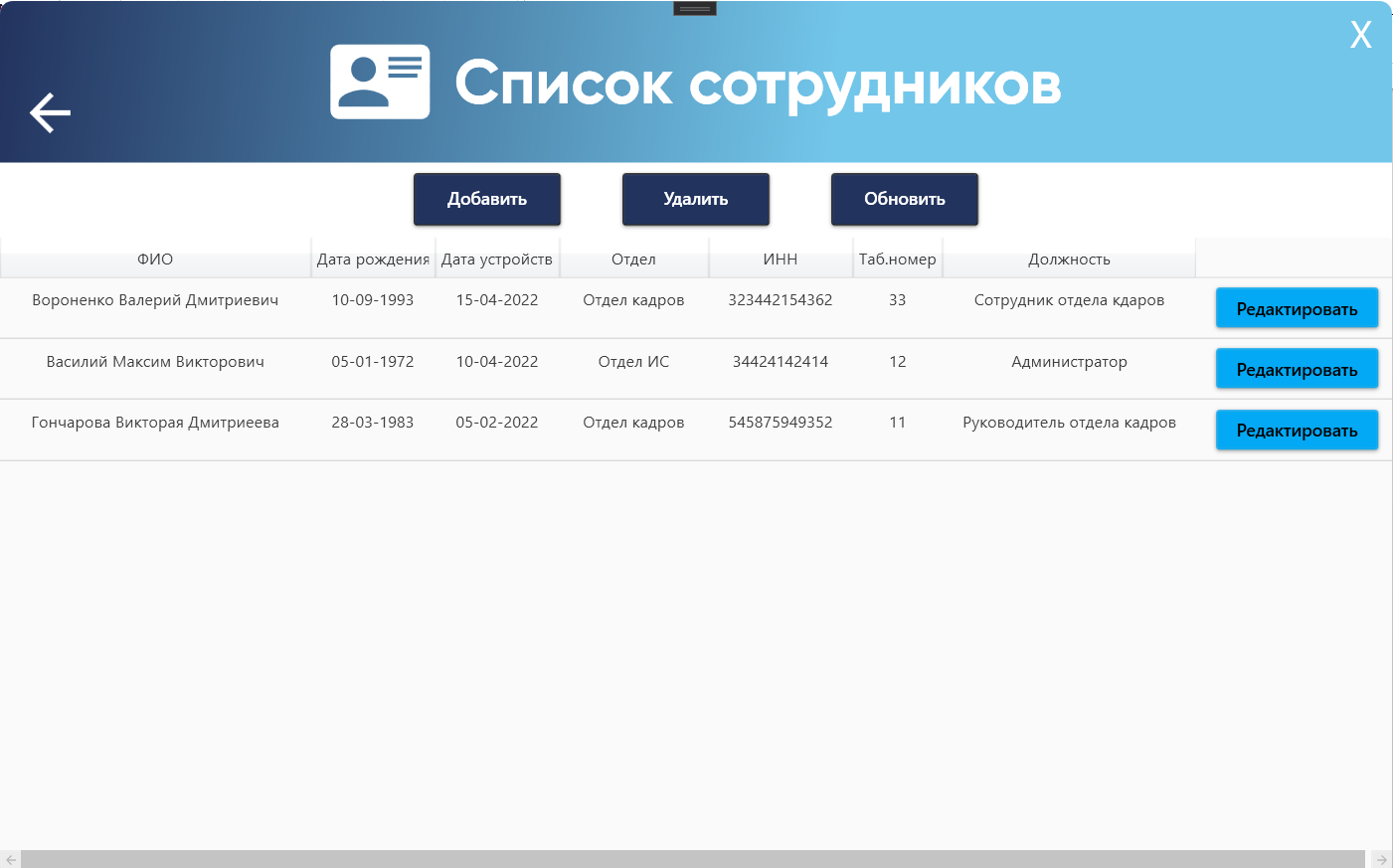


Рисунок 28 – Окно списка сотрудников

В окне представлены следующие функциональные кнопки:

* добавить (позволяет перейти в окно для добавления сотрудника в список, при заполнении всех полей и нажатии на кнопку сохранить добавляет информацию в таблицу);
* удалить (дает возможность удалить одну или несколько записей о сотрудниках с таблицы);
* обновить (обновляет содержимое таблицы);
* редактировать (предоставляет при надобности возможность изменить данные в поле таблицы).

Выбрав в основном меню «Приказы» также как и в случае с списком сотрудников происходит переход в окно.

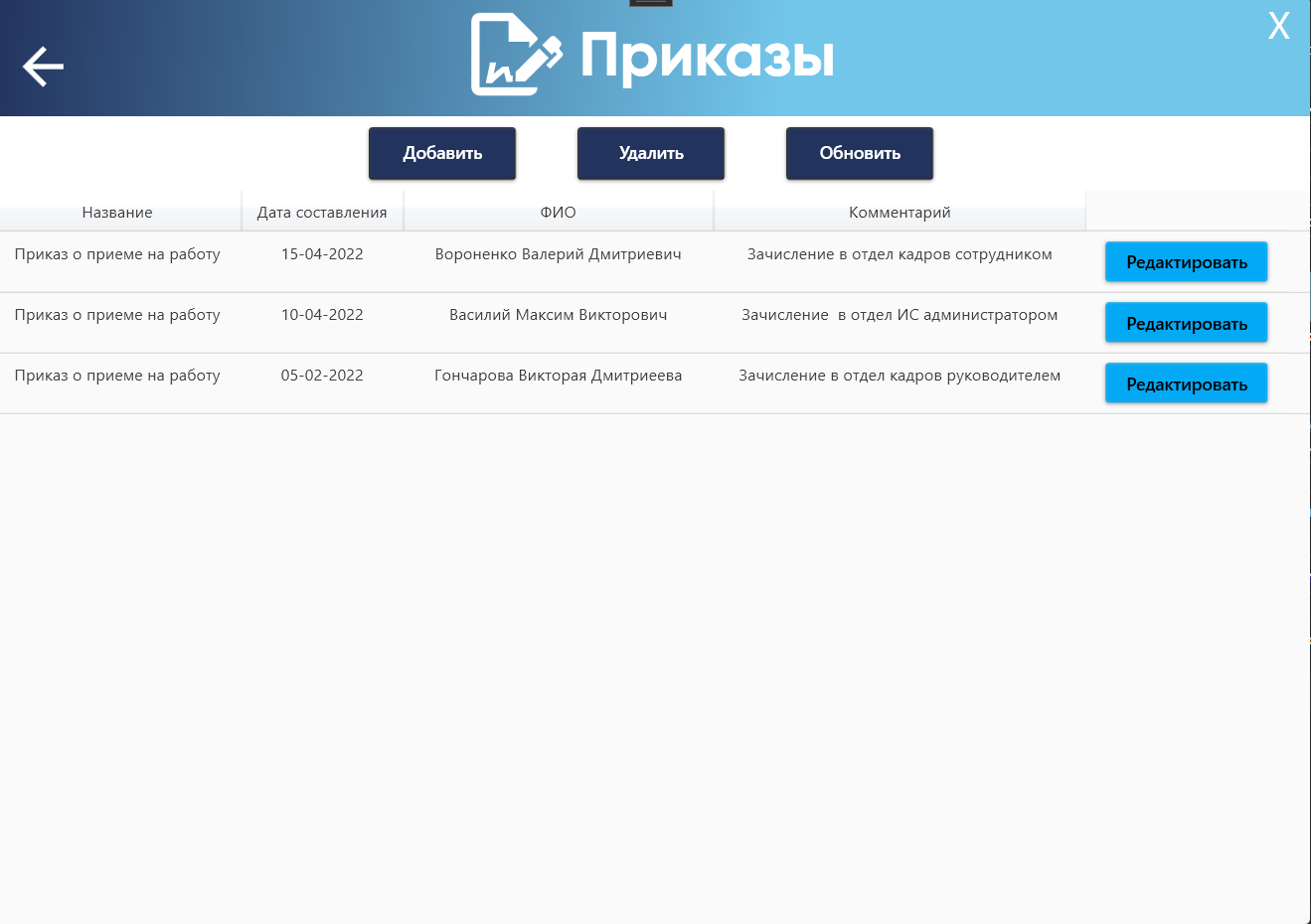


Рисунок 29 – Окно приказов

В окне представлены следующие функциональные кнопки:

* добавить (позволяет перейти в окно для добавления приказа в список, при заполнении всех полей и нажатии на кнопку сохранить добавляет информацию в таблицу);
* удалить (дает возможность удалить одну или несколько записей о приказах);
* обновить (обновляет содержимое таблицы);
* редактировать (предоставляет при надобности возможность изменить данные в поле таблицы).

# ЗАКЛЮЧЕНИЕ

В первой части курсовой работы была разработана база данных информационной системы «Учёт сотрудников предприятия». Также, был рассмотрен процесс разработки запросов ИС «Учёт сотрудников предприятия». В дальнейшем создана база данных системы в программной среде MS SQL Server Management Studio 18.

Во второй части курсовой работы на основе рассмотренной предметной области «Агентство недвижимости «Высота»» и анализа процесса учёта сотрудников данного предприятия был разработан и подробно описан интерфейс информационной системы в программе MS Visual Studio 22, а затем был разработан программный модуль автоматизированной информационной системы «Учёт сотрудников предприятия», конечным шагом стала разработка руководства пользователя.

В данной курсовой работ было уделено внимание программной реализации информационной системы «Учёт сотрудников предприятия». Было организовано подключение к БД и был разработан интерфейс программного продукта, исходя из чего цель данной курсовой работы можно назвать выполненными.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 34.601 – 90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
2. ГОСТ 34.602 - 2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
3. ГОСТ 19.201 - 78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.
4. ГОСТ 19.202 - 78 ЕСПД. Спецификация. Требования к содержанию и оформлению.
5. ГОСТ Р ИСО/МЭК 12207. Процессы жизненного цикла программных средств.
6. Золотов С.Ю. Проектирование информационных систем: Учебно-методическое пособие. Методические рекомендации для выполнения курсового проекта, лабораторных работ и практических занятий по дисциплине «Проектирование информационных систем» - Томск: ТУСУР, 2013. - 34 с.
7. Шнайдер, Роберт Microsoft SQL Server 6.5. Проектирование высокопроизводительных баз данных; М.: Лори, 2010. - 361 c
8. Петкович, Душан Microsoft SQL Server 2012. Руководство для начинающих / Душан Петкович. - М.: БХВ-Петербург, 2012. - 460 c.
9. Тейлор, Аллен SQL для чайников / Аллен Тейлор. - М.: Вильямс, 2014. - 416 c.
10. Браст, Э.Дж. Разработка приложений на основе Microsoft SQL Server 2008 / Э.Дж. Браст. - М.: Русская Редакция, 2010. - 751 c.
11. Хетагуров, Я. А. Проектирование автоматизированных систем обработки информации и управления (АСОИУ). Учебник / Я.А. Хетагуров. - М.: Бином. Лаборатория знаний, 2015. - 240 c.
12. Кристофер, Д. Маннинг Введение в информационный поиск / Кристофер Д. Маннинг, ПрабхакарРагхаван ,ХайнрихШютце. - М.: Вильямс, 2014. - 528 c.
13. Бишоп Дж. С# в кратком изложении; Бином. Лаборатория знаний - М., 2015. - 234 c.
14. ГриффитсИэн Программирование на C# 5.0; Эксмо - М., 2014. - 580 c.
15. Гуриков С. Р. Введение в программирование на языке Visual C#; ИЛ - Москва, 2013. - 448 c.
16. Шилдт Герберт C# 4.0. Полное руководство; Вильямс - М., 2015. - 291 c.
17. Эндрю Троелсен Язык программирования C# 5.0 и платформа .NET 4.5; Диалектика / Вильямс - М., 2015. - 126 c.

# ПРИЛОЖЕНИЕ А.

# ЛИСТИНГ ПРОГРАММЫ

namespace IS\_EmployeesAccounting

{

/// <summary>

/// Логика взаимодействия для MainMenuWindow.xaml

/// </summary>

public partial class MainMenuWindow : Window

{

EmployeeList employeeList = new EmployeeList();

public MainMenuWindow()

{

InitializeComponent();

}

private void mainMenuBorder\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

this.DragMove();

}

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

{

Close();

}

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

{

employeeList.Show();

this.mainMenuWindow.Close();

}

}

}

namespace IS\_EmployeesAccounting

{

/// <summary>

/// Interaction logic for EmployeeList.xaml

/// </summary>

public partial class EmployeeList : Window

{

public EmployeeList()

{

InitializeComponent();

dGridEmployeess.ItemsSource = AccountingEmployeesEntities3.GetContext().Employees.ToList();

}

private void employeeBorder\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

this.DragMove();

}

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

{

Close();

}

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

{

MainMenuWindow mainMenuWindow = new MainMenuWindow();

mainMenuWindow.Show();

employeeList.Hide();

}

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

{

AddEmployee addEmployee = new AddEmployee(null);

addEmployee.Show();

}

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

{

var employeeForRemoving = dGridEmployeess.SelectedItems.Cast<Employee>().ToList();

if(MessageBox.Show($"Вы точно хотите удалить следующие {employeeForRemoving.Count()} элементов?", "Внимание",

MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{ AccountingEmployeesEntities3.GetContext().Employees.RemoveRange(employeeForRemoving);

AccountingEmployeesEntities3.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

dGridEmployeess.ItemsSource = AccountingEmployeesEntities3.GetContext().Employees.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

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

{

AddEmployee addEmployee = new AddEmployee((sender as Button).DataContext as Employee);

addEmployee.Show();

}

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

{

AccountingEmployeesEntities3.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

dGridEmployeess.ItemsSource = AccountingEmployeesEntities3.GetContext().Employees.ToList();

}

}

}

namespace IS\_EmployeesAccounting

{

/// <summary>

/// Логика взаимодействия для AuthorizationWindow.xaml

/// </summary>

public partial class AuthorizationWindow : Window

{

public AuthorizationWindow()

{

InitializeComponent();

OpenPage(pages.loginPage);

}

private void AuthorizationBorder\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

this.DragMove();

}

public enum pages

{

loginPage

}

public void OpenPage(pages pages)

{

if (pages == pages.loginPage)

{

authorizationFrame.Navigate(new LoginPage (this));

}

}

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

{

Close();

}

}

}

namespace IS\_EmployeesAccounting

{

/// <summary>

/// Interaction logic for AddEmployee.xaml

/// </summary>

public partial class AddEmployee : Window

{

private Employee \_employee = new Employee();

public AddEmployee(Employee selectedEmployee)

{

InitializeComponent();

if (selectedEmployee != null)

\_employee = selectedEmployee;

DataContext = \_employee;

comboPositions.ItemsSource = AccountingEmployeesEntities3.GetContext().Positions.ToList();

}

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

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_employee.FullName))

errors.AppendLine("Укажите ФИО");

if (\_employee.TIN == null || \_employee.TIN.Length < 12)

errors.AppendLine("ИНН должен содержать 12 цифр");

if (string.IsNullOrWhiteSpace(\_employee.Division))

errors.AppendLine("Укажите отдел");

if (\_employee.TabNum == 0)

errors.AppendLine("Укажите табельный номер");

if (\_employee.Position == null)

errors.AppendLine("Выберете должность");

if(errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;

}

if(\_employee.id == 0)

{

AccountingEmployeesEntities3.GetContext().Employees.Add(\_employee);

}

try

{

AccountingEmployeesEntities3.GetContext().SaveChanges();

MessageBox.Show("Информация сохранена!");

addEmployeeWindow.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

private void TextBlock\_SelectedDateChanged(object sender, SelectionChangedEventArgs e)

{

}

private void addEmployeeBorder\_MouseLeftButtonDown\_1(object sender, MouseButtonEventArgs e)

{

this.DragMove();

}

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

{

addEmployeeWindow.Close();

}

}

}