Департамент образования и науки Курганской области

Государственное бюджетное профессиональное образовательное учреждение «Курганский технологический колледж

имени Героя Советского Союза Н. Я. Анфиногенова»

|  |  |
| --- | --- |
|  | Защищен с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. |

Отделение Автоматизация и вычислительная техника

КУРСОВОЙ ПРОЕКТ

По МДК.01.01 «Разработка программных модулей»

на тему: «Разработка приложения для автоматизации работы сотрудника отдела кадров»

КТК.09.02.07.2063.14874.КП

Студент: Ершов Илья Александрович

Преподаватель: Афанасьев Алексей Васильевич

Курган, 2022

Задание

на курсовой проект

по МДК.01.01 «Разработка программных модулей»

на тему: «Разработка приложения для автоматизации работы сотрудника отдела кадров»

Пояснительная записка

Введение

1 Аналитическая часть

­ дать характеристику предметной области;

­ дать характеристику программных средств для разработки приложения.

2 Проектная часть

­ описать структуру программного продукта;

­ определить роли и пользователей программного продукта.

- описать руководство пользователя программного продукта

Заключение

Литература

Получил студент гр. 2063 \_\_\_\_\_\_\_\_\_\_\_ И.А. Ершов

Преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Афанасьев

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022

Содержание

Введение 4

1. Аналитическая часть 5

1.1 Характеристика предметной области 5

1.2 Характеристика программных средств для разработки программы 8

2. Проектная часть 10

2.1 Структура программного продукта 10

2.2 Роли и пользователи программного продукта 10

2.3 Руководство пользователя программного продукта 11

Заключение 17

Литература 18

Введение

Суть проекта состоит в создании приложения, в котором сотрудники отдела кадров будут иметь возможность вносить и изменять данные работников какого-либо предприятия. Сотрудник отдела кадров имеет возможность редактировать данные работников, добавлять должности работников.

Данное приложение автоматизирует работу сотрудника отдела кадров, избавляет от большого количества распечатанных бумаг.

Цель курсового проекта: «Разработка приложения для автоматизации работы сотрудника отдела кадров»

Для удовлетворения поставленной цели необходимы выполнить ряд задач:

* Изучить область данных работы сотрудника отдела кадров
* Создать базу данных на основе изученной области данных;
* Создать приложение WPF на языке C#, взаимодействующее с областью данных.
* Протестировать созданное приложение.

1. Аналитическая часть.

1.1 Характеристика предметной области.

Предметной областью проекта является деятельность сотрудника отдела кадров.

Структуру предметной области отображает данная ER-диаграмма (Рисунок 1.1).

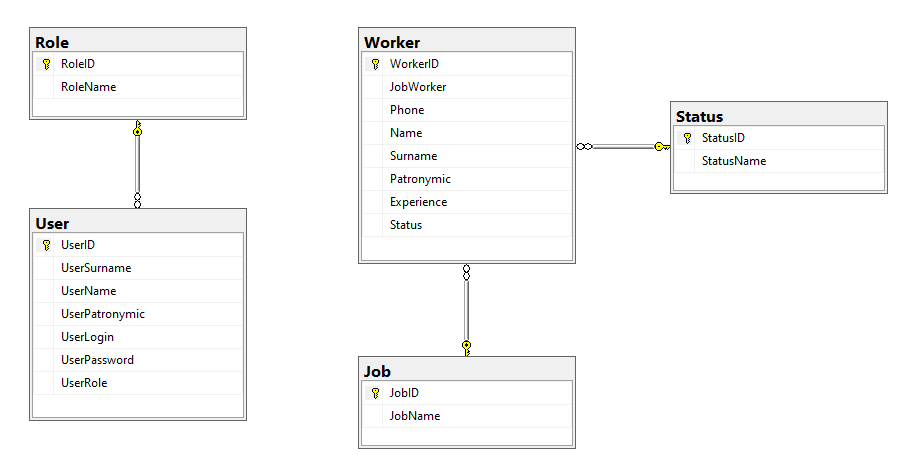


Рисунок 1.1 – ER-диаграмма базы данных.

База данных состоит из 5 таблиц:

* Worker – содержит информацию о работниках предприятия.
* Job – содержит информацию о наименование должностей
* Users – содержит информацию о работниках.
* Role – содержит о ролях сотрудников отдела кадров.
* Status– содержит информацию о статусе работников предприятия.

Создание базы данных выполнено путем написания SQL-скрипта с помощью Microsoft SQL Server Management Studio 18 (Рисунок 1.2).

create database HR

go

use HR

go

create table Job

(

JobID int primary key identity(1,1) not null,

JobName nvarchar(32) not null,

)

go

create table [Status]

(

StatusID int primary key not null,

StatusName nvarchar(20) not null,

)

go

create table Worker

(

WorkerID int primary key identity(1,1) NOT NULL,

JobWorker int foreign key references Job(JobID) not null,

Phone nvarchar(20) NOT NULL,

[Name] nvarchar(32) NOT NULL,

Surname nvarchar(32) NOT NULL,

Patronymic nvarchar(32) NOT NULL,

Experience int,

[Status] int foreign key references [Status](StatusID),

)

create table [Role]

(

RoleID int primary key identity,

RoleName nvarchar(100) not null

)

create table [User]

(

UserID int primary key identity(1,1),

UserSurname nvarchar(100) not null,

UserName nvarchar(100) not null,

UserPatronymic nvarchar(100) not null,

UserLogin nvarchar(max) not null,

UserPassword nvarchar(max) not null,

UserRole int foreign key references [Role](RoleID) not null)

1 Аналитическая часть

1.1 Характеристика программных средств для создания приложения

Для реализации цели были использованы следующие программы:

- Microsoft Visual Studio - это панель запуска для написания, отладки и создания кода для приложений, а затем их публикации. В дополнение к стандартному редактору и отладчику, которые есть в большинстве IDE, Visual Studio включает компиляторы, надстройки кода, графические дизайнеры и многие другие функции, расширяющие возможности разработки.

Visual Studio включает редактор исходного кода, который поддерживает технологию IntelliSense и позволяет легко перерабатывать код. Встроенный отладчик может действовать как отладчик уровня исходного кода или уровня машины. Другие встраиваемые инструменты включают редактор форм, упрощающий создание графического пользовательского интерфейса приложения, веб-редактор, конструктор классов и конструктор схемы базы данных.

Visual Studio позволяет создавать и подключать сторонние надстройки (расширения) для расширения функциональности практически на каждом уровне, включая поддержку систем контроля версий исходного кода (таких как Subversion и Visual SourceSafe), добавление новых наборов инструментов (таких как domain- определенные языки программирования), редактирование кода и визуальный дизайн) или другие аспекты процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server)

- Material Design In XAML - дополнение (плагин), разработанное компанией Microsoft, для удобного создания графического интерфейса в приложение WPF

- Database Designer - онлайн-инструмент проектирования и моделирования баз данных, графическое представление таблиц и связей. Помимо создания SQL-скрипта для всех основных движков баз данных, также можно экспортировать свою базу данных в формате image (PNG) или pdf. SQL скрипт создается автоматически после визуального построения модели и поддерживается следющие движки баз данных: MySQL, PostgreSQL, MS SQL, Oracle и SQLite.

- Microsoft SQL Server - система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, созданный совместно Microsoft и Sybase. Transact-SQL — это реализация стандартного языка структурированных запросов ANSI/ISO с расширениями. Он используется для работы с базами данных, начиная от персональных и заканчивая крупномасштабными базами данных

- Microsoft Excel - программа для работы с электронными таблицами, созданная корпорацией Microsoft для Microsoft Windows, Windows NT и Mac OS, а также Android, iOS и Windows Phone. Она предоставляет возможности экономико-статистических расчетов, графические инструменты и, за исключением Excel 2008 под Mac OS X, язык макропрограммирования VBA (Visual Basic for Application). Microsoft Excel входит в состав Microsoft Office.

1.2 Функции и параметры программных средств, используемых для решения задач

Для реализации кода в Microsoft Visual Studio был использован язык программирования С#. Для реализации дизайна, язык разметки XAML.

C# — это объектно - и компонентно-ориентированный язык программирования. C# предоставляет языковые конструкции прямая поддержка такой концепции работы. Это делает C# подходящим для создания и использования программных компонентов.

С момента своего появления C# был обогащен функциями, поддерживающими новые рабочие нагрузки и современные передовые методы разработки программного обеспечения. Программы C# выполняются в .NET, виртуальной системе выполнения, которая вызывает CLR и библиотеки классов. Common Language Runtime (CLR) — это международная стандартная реализация Microsoft Common Language Infrastructure (CLI). CLI является основой для создания сред выполнения и разработки, в которых языки и библиотеки прозрачно взаимодействуют друг с другом.

Исходный код, написанный на C#, переводится на промежуточный язык (IL), соответствующий спецификациям CLI. Код и ресурсы IL, включая растровые изображения и строки, хранятся в одной сборке, обычно с расширением .dll. Компиляция содержит манифест, содержащий информацию о типах, версии и культуре компиляции.

Когда программа C# выполняется, сборка загружается в среду CLR. CLR JIT преобразует код IL в инструкции машинного языка. Среда CLR также выполняет другие операции, такие как автоматическая сборка мусора, обработка исключений и управление ресурсами. Код, выполняемый в общеязыковой среде выполнения, иногда называют «управляемым кодом», чтобы отличить этот подход от «неуправляемого кода» который немедленно переводится на машинный язык для данной платформы.

Обеспечение взаимодействия между языками — ключевая особенность .NET. IL-код, сгенерированный компилятором C#, соответствует стандарту Common Type Specification (CTS). Код IL, сгенерированный из кода C#, может работать с кодом, сгенерированным из языковых версий .NET F#, Visual Basic, C++ и более чем 20 CTS-совместимых языков. Сборка может содержать несколько модулей, написанных на разных языках. NET, и все типы могут ссылаться друг на друга, как если бы они были написаны на одном языке.

В дополнение к службам среды выполнения .NET включает расширенные библиотеки. Эти библиотеки поддерживают широкий спектр рабочих нагрузок. Они организованы в пространства имен, которые предоставляют полезные опции от операций ввода и вывода файлов до манипуляций со строками и анализа XML, от фреймворков веб-приложений до элементов управления Windows Forms. Приложения C# обычно широко используют библиотеку классов .NET для решения общих задач.

Работа над C# началась в декабре 1998 года. Проект получил кодовое название COOL (объектно-ориентированный язык в стиле C).

Первая бета-версия C# 1.0 была выпущена летом 2000 года, а в феврале 2002 года это была финальная версия языка в дополнение к Microsoft Visual Studio. Поскольку C# сочетает в себе лучшие черты популярных языков программирования, которые предшествовали ему в C, Java и C++, программисты могут легко перейти на C# на основе своих знаний любого из этих языков.

Основным отличием от предшественников была возможность прописывать ингредиенты. C # представил такие функции, как события, методы и свойства. Финальный релиз C# 2.0 вышел в 2005 году. Это укрепило ее позиции на рынке.

Новые функции, такие как анонимные методы, частичные и параметризованные типы, значительно расширили возможности использования C#.

Во второй версии добавлены 64-битные вычисления, что открыло возможность увеличения адресного пространства. Триггеры, хранимые процедуры и типы данных также были созданы на языках .NET.

Версия 3.0, выпущенная в 2008 году, позволила C# занять лидирующие позиции в «гонке вооружений» языков. Новым в C# 3.0 является интегрированный язык запросов (LINQ); неявно введенные переменные и методы расширения; лямбда-выражения, которые назначают реализацию кода делегатам с использованием нового, более простого синтаксиса.

В частности, лямбда-выражения можно «отличить» вместе с языком интегрированных запросов.

Переменные анонимного типа позволили избежать неудобств и прямого неудобства описания переменных, позволяя объявлять новый тип непосредственно при создании. Также новым в C# 3.0 являются так называемые «ленивые вычисления», которые выполняют необходимые вычисления только в том случае, если запрашиваются соответствующие релевантные данные.

C# 4.0 был выпущен в 2010 году. Основное дополнение к предыдущим версиям с именем и необязательными аргументами. Первый позволяет связать аргумент и параметр по имени, а второй позволяет указать аргумент по умолчанию для каждого параметра. Не менее важным нововведением является динамический тип. Позволяет проверять соответствие типов объектов не на этапе компиляции, а непосредственно во время выполнения программы.

В то же время в .NET Framework 4.0 были внедрены нововведения — параллельная версия библиотеки распараллеливания задач (TPL) и интегрированный язык запросов (PLNQ). Их поддержка позволяет выполнять код параллельно на компьютерах с многоядерными или несколькими одноядерными процессорами.

XAML - основанный на XML язык разметки для декларативного программирования приложений, разработанный Microsoft. XAML широко используется в .NET Framework 3.0, особенно в Windows Presentation Foundation (WPF), Xamarin.Forms, Windows Workflow Foundation (WWF) и Silverlight. В WPF XAML — это язык разметки пользовательского интерфейса для определения элементов пользовательского интерфейса, привязки данных, поддержки событий и других свойств. В WWF XAML можно использовать для определения рабочих процессов.

Файлы XAML можно создавать и редактировать с помощью инструментов визуального дизайна, таких как Microsoft Expression Blend, Microsoft Visual Studio, WPF Visual Designer. Их также можно создавать с помощью стандартного текстового редактора, редактора кода, такого как XAMLPad, или графического редактора, такого как Vectropy.

Все, что создано или реализовано на XAML, может быть выражено на традиционных языках .NET, таких как C# или Visual Basic.NET. Однако ключевым аспектом технологии является снижение сложности инструментов, используемых для обработки XAML, поскольку XAML основан на XML. В результате существует множество продуктов, которые создают приложения на основе XAML. Поскольку XAML основан на XML, разработчики и дизайнеры могут работать с содержимым одновременно без необходимости перевода.

WPF (Windows Presentation Foundation) - это .NET платформа для создания клиентских приложений для настольных систем. Данный видео курс предназначен для тех, кто владеет языком C# на базовом уровне и хочет развиваться в сфере разработки настольного программного обеспечения2. Проектная часть.

2.1 Структура программного продукта.

Структура программного продукта курсового проекта состоит из следующих частей:

* Главное окно отвечает за авторизацию сотрудника отдела кадров
* Следующая страница после авторизации отвечает за выбор действия сотрудника отдела кадров
* Добавление работников и должностей
* Редактирование работников

2.2 Роли и пользователи.

Для предотвращения несанкционированного доступа к приложению была введена авторизация пользователей с помощью логина, пароля и капчи.

2.3 Руководство пользователя программного продукта

При запуске приложения вы попадаете на страницу авторизации сотрудника отдела кадров (Рисунок 2.1)

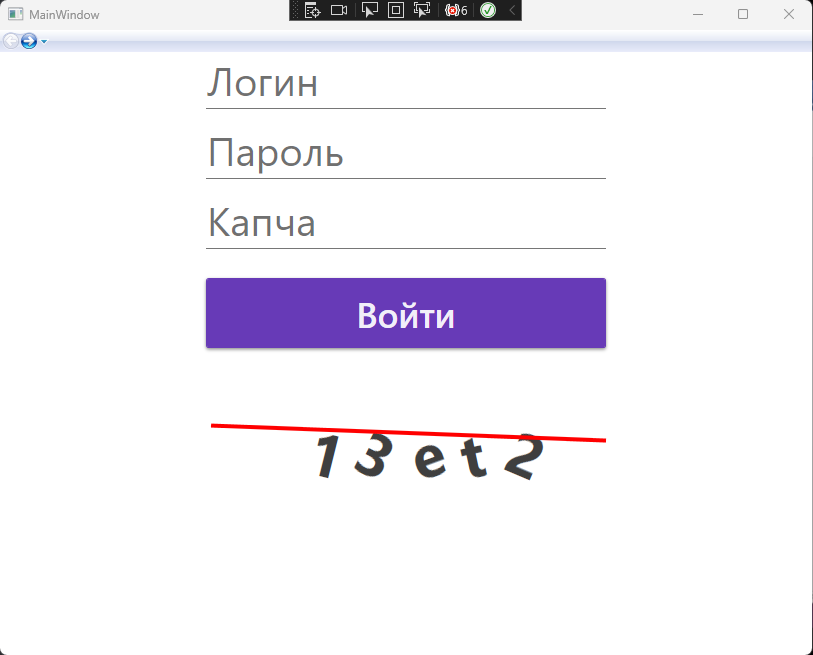


Рисунок 2.1 – Страница авторизации.

После успешной авторизации вы попадете в окно выбора действия.

(Рисунок 2.2)



Рисунок 2.2 – Страница выбора действия.

При выборе действия «Добавление работников и должностей» вы попадете на страницу, которая указана на (Рисунок 2.3)

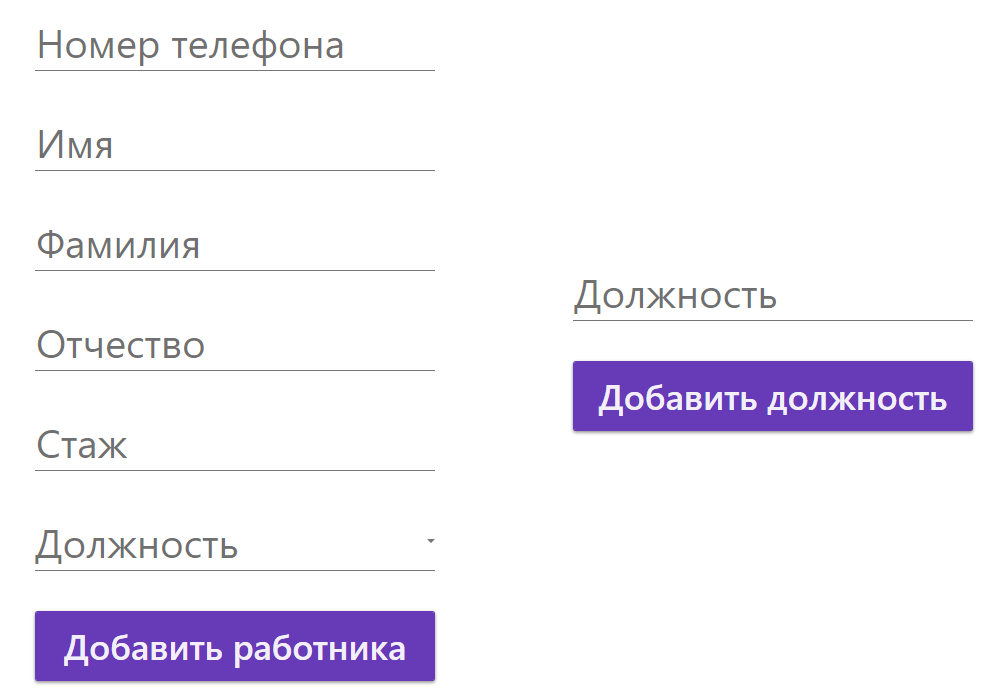


Рисунок 2.3 – Страница добавления работников и должностей.

На странице «Добавление работников и должностей» нужно указать номер телефона, имя, фамилию, отчество, стаж и должность нового работника.

В случае если необходимо отредактировать данные о уже добавленных работниках необходимо перейти на страницу «Редактирование работников» (Рисунок 2.4)

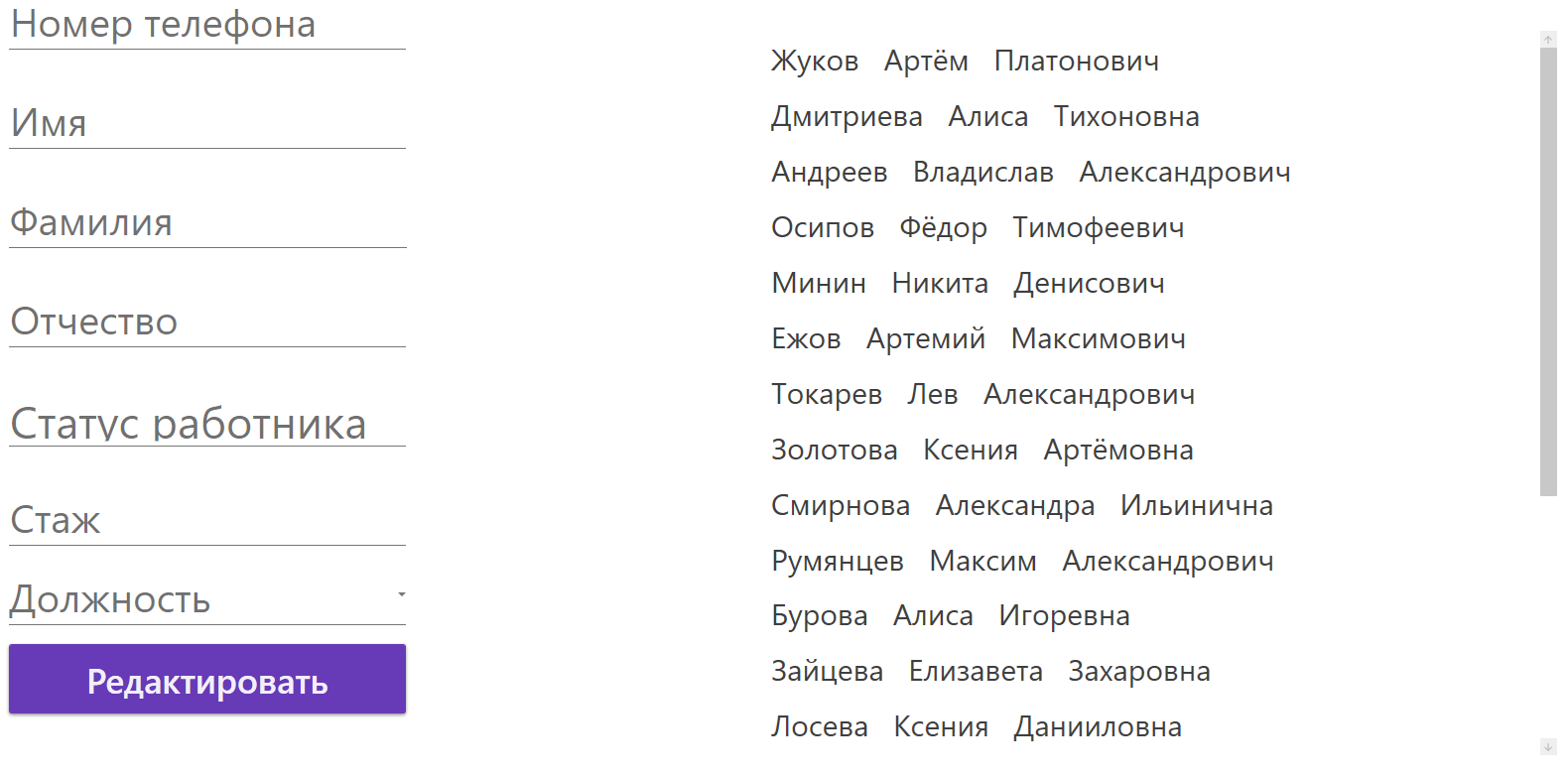


Рисунок 2.4 – Страница редактирования работников.

На данной странице нужно выбрать необходимого работника и нажать на его ФИО. (Рисунок 2.5)

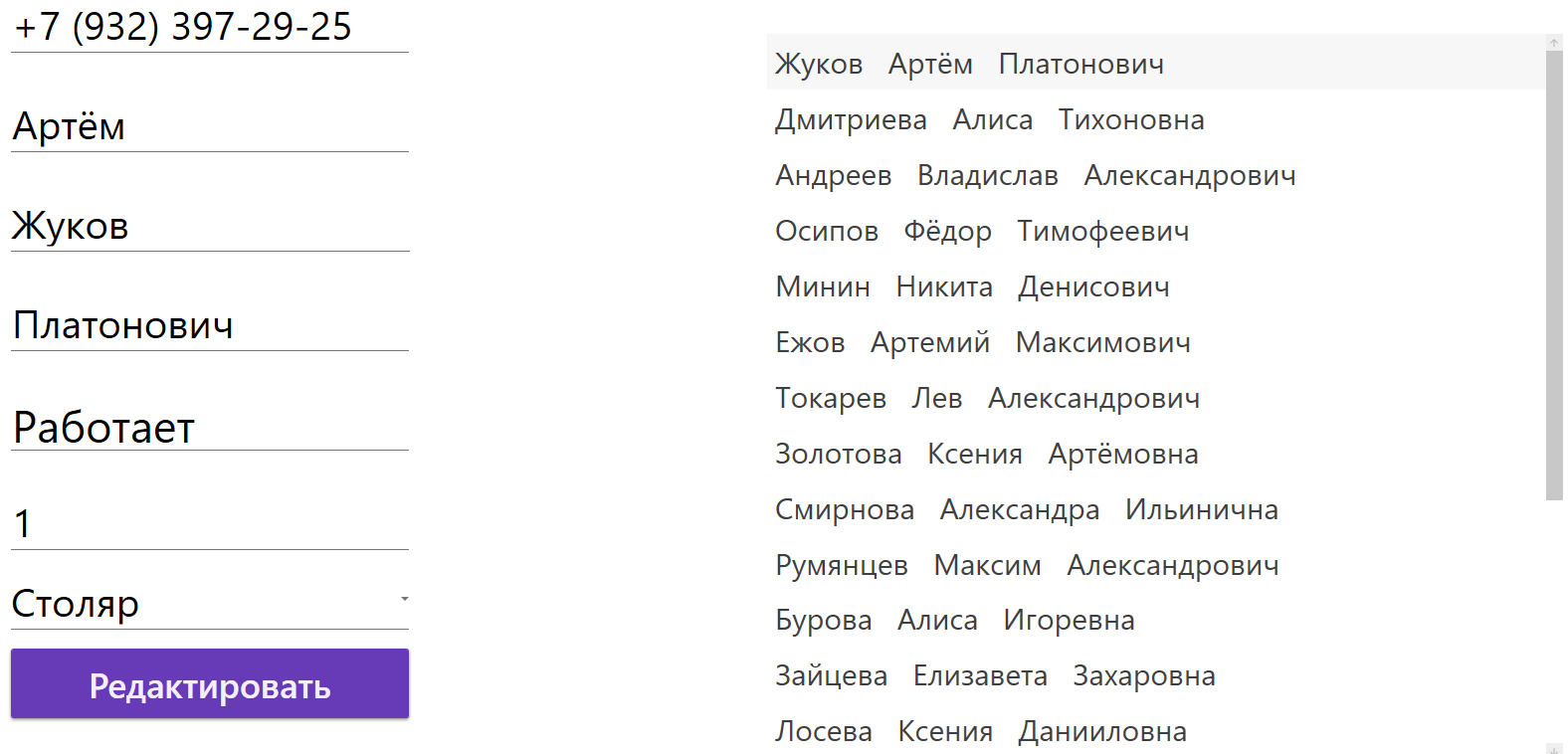
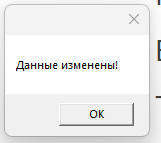
При нажатии на ФИО будут доступны текущие данные работника. 

Рисунок 2.5 – Пример выбранного работника.

Необходимо внести нужные изменения в данные работника и нажать кнопку «Редактировать» после чего вы получите уведомление о успешном изменение данных (Рисунок 2.6)

Заключение.

Автоматизация работы отдела кадров призвана дать сотрудникам удобный инструмент для хранения и редактирования данных о работниках, а также сократить большое количество бумажной волокиты.

Данный курсовой проект позволяет автоматизировать работу сотрудников отдела кадров, дав им инструмент, позволяющий им удобно вести учет о работниках предприятия.

Были выполнены все необходимые задачи:

• Изучена область данных работы работников отдела кадров;

• Создана база данных на основе изученной области данных;

• Создано приложение WPF на языке C#, взаимодействующее с областью данных.

Литература.

1. Рихтер CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C# / Рихтер, Джефри. - М.: Питер, 2020. - 656 c.
2. Нейгел, К. C# 2005 для профессионалов / К. Нейгел. - М.: Вильямс, 2020. – 927 c.
3. Милов, А. В. Основы программирования в задачах и примерах / А.В. Милов. - М.: Фолио, 2018. - 400 c.
4. Грин Дж. Изучаем C# / Дж. Грин, Э. Стиллмен. – СПб.: «Питер», 2012.
5. Основы языка программирования C#: [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/C\_Sharp
6. Microsoft Visual Studio [Электронный ресурс]. URL <https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio>
7. Microsoft SQL Server [Электронный ресурс] URL <https://ru.wikipedia.org/wiki/Microsoft_SQL_Server>
8. XAML [Электронный ресурс] URL https://ru.wikipedia.org/wiki/XAML
9. Основы работы с платформой WPF: [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf (дата 20.06.2022).
10. Windows Presentation Foundation: [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Windows\_Presentation\_Foundatio (дата 20.05.2022).