Министерство образования Новосибирской области

Государственное бюджетное профессиональное образовательное учреждение

Новосибирской области

«Новосибирский колледж электроники и вычислительной техники»

(ГБПОУ НСО «Новосибирский колледж электроники и вычислительной техники»)

Курсовой проект

по дисциплине: «Разработка программных модулей»

на тему: «Разработка информационной подсистемы системы по учёту персонала»

Выполнил студент:

Группы 9ИС-320

Крамар Игорь Евгеньевич

Проверил преподаватель:

Конюхова Анна Алексеевна

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

\_\_\_\_\_\_\_\_\_\_(оценка)

г. Новосибирск, 2024г.

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc186215796)

[ВВЕДЕНИЕ 3](#_Toc186215797)

[1. ОБЩАЯ ЧАСТЬ 4](#_Toc186215798)

[1.1. Общие сведения о системе по учёту персонала 4](#_Toc186215799)

[1.2. IDE Microsoft Visual Studio 5](#_Toc186215800)

[1.3. WPF и C# 7](#_Toc186215801)

[1.4. Понятие базы данных и SSMS 9](#_Toc186215802)

[2. ОСНОВНАЯ ЧАСТЬ 11](#_Toc186215803)

[2.1. Проектирование и разработка базы данных 11](#_Toc186215804)

[2.2. Разработка приложения «Учёт персонала» 13](#_Toc186215805)

[3. ЗАКЛЮЧЕНИЕ 19](#_Toc186215806)

[СПИСОК ЛИТЕРАТУРЫ 20](#_Toc186215807)

[ПРИЛОЖЕНИЕ 21](#_Toc186215808)

[Листинг приложения 21](#_Toc186215809)

[Листинг процедур 34](#_Toc186215810)

# ВВЕДЕНИЕ

В современных условиях динамичного развития бизнеса и стремительного внедрения информационных технологий, управление человеческими ресурсами становится одной из ключевых задач для организаций различного масштаба. Эффективный учёт и управление персоналом не только способствуют повышению производительности труда, но и обеспечивают конкурентные преимущества на рынке.

Создание информационной подсистемы для учёта персонала представляет собой важную задачу, способствующую улучшению управления человеческими ресурсами и повышению общей эффективности деятельности организации. Это обосновывает актуальность выбранной темы курсовой работы.

Целью данной курсовой работы является разработка информационной подсистемы для учёта персонала, которая позволит автоматизировать процессы управления человеческими ресурсами, повысить эффективность учёта и анализа данных о сотрудниках, а также улучшить взаимодействие между различными подразделениями организации.

Задачами данной курсовой работы являются следующие пункты:

* изучить необходимую теорию для разработки;
* разработать архитектуру информационной подсистемы, включая базы данных и пользовательский интерфейс;
* создать прототип информационной подсистемы с базовым функционалом;
* подготовить отчёт о проведённой работе и полученных результатах.

# ОБЩАЯ ЧАСТЬ

## 1.1. Общие сведения о системе по учёту персонала

Кадровый учет должен вести любой бизнес с сотрудниками. Иногда такую работу поручают бухгалтеру, если он не против. Для этого ему нужно знать Трудовой кодекс РФ и уметь работать с кадровыми документами.  
 Кадровый учет иногда называют «кадровое делопроизводство» — это документальное оформление всех этапов трудовых отношений с работником. Например, прием на работу, перевод и увольнение, командировки, отпуска, больничные, учет рабочего времени. Он нужен бизнесу, чтобы регламентировать отношения с работниками, сообщать им об условиях работы и их изменении так, как этого требует закон. Это поможет подстраховаться на случай конфликтных ситуаций.  
 В крупных компаниях кадровым документооборотом обычно занимается специальное подразделение — отдел кадров. Отделы кадров подбирают специалистов, ведут кадровый учет, иногда занимаются укреплением HR-бренда — создают привлекательный образ компании среди потенциальных сотрудников. По закону такой отдел компания создавать не обязана. Малый бизнес может поручить кадровую работу бухгалтеру — по согласованию с ним.  
 Система по учёту персонала — это комплекс мер, направленных на документальное оформление всех этапов трудовых отношений с работником. Например, приём на работу, перевод и увольнение, командировки, отпуска, больничные, учёт рабочего времени.

Основная задача — следить за правильностью и своевременностью отражения изменений, происходящих в подсистеме функционирования компании.

Каждый факт в системе трудовых ресурсов должен быть задокументирован. При приёме в штат работника (или увольнении) фирма формирует приказ, после чего на каждого специалиста заводится личное дело.

По каждому специалисту ведутся учётные документы: табель учёта рабочих часов, табель расчёта зарплаты, расчётные ведомости и т. д..

На основе данных учёта персонала фирма может составлять отчёты. Они показывают руководителям подразделений актуальную и полную картину функционирования подведомственных департаментов или отделов.

## 1.2. IDE Microsoft Visual Studio

Программисты используют разные инструменты и программы. Один из основных, с которыми они имеют дело чаще всего, — IDE, или интегрированная среда разработки.

IDE (от англ. Integrated Development Environment, «интегрированная среда разработки») — это программа, в которой разработчики пишут, проверяют, тестируют и запускают код, а также ведут большие проекты. Она включает в себя сразу несколько инструментов: редактор для написания кода, сервисы для его проверки и запуска, расширения для решения дополнительных задач разработки. Можно сказать, что это как Photoshop для дизайнера — общее пространство для большинства рабочих процессов.

Иногда IDE путают с редакторами кода, но на самом деле это не одно и то же. Редактор кода — это простой инструмент. Его задача — обеспечить программисту удобное пространство для написания кода, с подсветкой синтаксиса и автозаполнением отдельных функций. В них, как правило, довольно мало дополнительных инструментов, и часто их требуется подключать отдельно. Обычно запускать код в них можно только с помощью расширений. IDE — более комплексный инструмент. Он обязательно включает в себя редактор кода и массу дополнительных инструментов. В IDE всегда доступен запуск кода сразу, без дополнительных настроек.

IDE пользуются программисты — это их главный инструмент. Он нужен им, чтобы:

* писать код в удобной среде: с подсветкой синтаксиса, автозаполнением, быстрым подключением библиотек;
* переключаться между разными файлами проекта с помощью встроенного файлового менеджера;
* искать нужные строки кода во всём проекте;
* рефакторить, то есть улучшать код, с помощью встроенных инструментов;
* проводить простые автотесты программы;
* запускать и отлаживать код прямо там, где его пишешь;
* собирать уже написанный код в готовые проекты;
* работать с системой контроля версий;
* создавать визуальный интерфейс с помощью встроенных инструментов;
* запускать код в контейнерах;
* прорабатывать визуальный интерфейс программы;
* выполнять другие операции, необходимые для работы.

Существует несколько десятков IDE, созданных для разных целей. Некоторые распространены более широко и используются сотнями тысяч программистов во всём мире. В данной работе использована IDE Microsoft Visual Studio.

Visual Studio — это интегрированная среда разработки (IDE), разработанная компанией Microsoft. Она используется для разработки компьютерных программ, включая веб-сайты, веб-приложения, веб-сервисы и мобильные приложения. Visual Studio использует платформы разработки программного обеспечения Microsoft, включая Windows API, Windows Forms, Windows Presentation Foundation (WPF), Microsoft Store и Microsoft Silverlight. Она может создавать как собственный код, так и управляемый код.

Visual Studio включает редактор кода, поддерживающий IntelliSense (компонент автозаполнения кода), а также рефакторинг кода. Встроенный отладчик работает как на уровне исходного кода, так и на уровне машинного кода. Другие встроенные инструменты включают профилировщик кода, конструктор для создания приложений с графическим интерфейсом, веб-конструктор, конструктор классов и конструктор схем баз данных. Он поддерживает плагины, которые расширяют функциональность практически на всех уровнях, включая поддержку систем управления версиями (таких как Subversion и Git) и добавление новых наборов инструментов, таких как редакторы и визуальные дизайнеры для предметно-ориентированных языков или наборы инструментов для других аспектов жизненного цикла разработки программного обеспечения (например, клиент Azure DevOps: Team Explorer).

Visual Studio поддерживает 36 различных языков программирования и позволяет редактору кода и отладчику поддерживать (в той или иной степени) практически любой язык программирования при условии наличия службы для конкретного языка. Встроенные языки включают C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML и CSS. Поддержка других языков, таких как Python, Ruby, Node.js и M, доступна через плагины. Java (и J#) поддерживались в прошлом.

## 1.3. WPF и C#

Данная работа написана на С#. C# (произносится си шарп) — объектно-ориентированный язык программирования общего назначения. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

Название «Си шарп» (от англ. sharp — диез) происходит от буквенной музыкальной нотации, где латинской букве C соответствует нота До, а знак диез (англ. sharp) означает повышение соответствующего ноте звука на полутон, что аналогично названию языка C++, где «++» обозначает инкремент переменной. Название также является игрой с цепочкой C → C++ → C++++(C#), так как символ «#» можно представить состоящим из 4 знаков «+».

Из-за технических ограничений на отображение (стандартные шрифты, браузеры и т. д.), а также из-за того, что знак диеза ♯ не представлен на стандартной клавиатуре компьютера, при записи имени языка программирования используют знак решётки (#). Это соглашение отражено в Спецификации языка C# ECMA-334. Тем не менее, на практике (например, при размещении рекламы и коробочном дизайне), «Майкрософт» использует знак диеза.

Названия языков программирования не принято переводить, поэтому язык называют, используя транскрипцию, — «Си шарп».

Windows Presentation Foundation (WPF) — аналог WinForms, 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 разметки XAML.

WPF предустановлена в Windows Vista (.NET Framework 3.0), Windows 7 (.NET Framework 3.5 SP1), Windows 8 (.NET Framework 4.0 и 4.5), Windows 8.1 (.NET Framework 4.5.1) и Windows 10 (.NET Framework 4.7). С помощью WPF можно создавать широкий спектр как автономных, так и запускаемых в браузере приложений.

## 1.4. Понятие базы данных и SSMS

Базы данных — это совокупность информации, которая хранится в соответствии с ее схемой и назначением. Они используются как инструмент для хранения данных о товарах, заявках, заказах, поставках, покупателях, объектах, услугах и т.д.

Простыми словами, БД — это набор сведений, которые упорядочены и структурированы.

Люди придумали для того, чтобы в удобном формате работать с большими объемами информации. Раньше ее хранили на бумагах, сортировали в папки, а затем раскладывали на полках огромного помещения с названием «Архив». С появлением компьютеров ситуация изменилась — теперь стало возможным собирать и хранить информацию в электронном виде. Но возник вопрос — как именно это делать. Первоначально данные хранили в текстовых файлах (и даже сейчас активно используется такой способ.)

Область применения:

* сбор и хранение. Особенно важно, когда информации очень много — требуется ее правильно собирать, использовать. В современном мире только тенденция к росту, и потребность в базах данных увеличивается;
* доступ в реальном времени. Базы должны быть устроены так, чтобы к ним был доступ. Если вносятся изменения, то они должны быть доступны для всех пользователей. Суть в том, что в любой момент можно обратиться за нужной информацией и получить ее;
* сохранение конфиденциальности. Бизнес использует персональную информацию. Например, телефоны, адреса, имена. С помощью использования баз данных к ним ограничивают доступ и сохраняют тем самым конфиденциальность;
* доступ к сведениям от разных пользователей. Использование баз данных позволяет организовать доступ к информации на многопользовательском уровне. Кроме того можно ограничивать или, наоборот, расширять доступ для разных служб и подразделений.

Базы данных разделяют на два основных типа: реляционные и нереляционные. В данной работе использована реляционная.

SQL Server Management Studio (SSMS) – это графическая среда (бесплатная), в которую входит набор инструментов для разработки сценариев на T-SQL и управления инфраструктурой Microsoft SQL Server.

Среда SQL Server Management Studio – это основной, стандартный и полнофункциональный инструмент для работы с Microsoft SQL Server, разработанный компанией Microsoft, который предназначен как для разработчиков, так и для администраторов SQL Server. С помощью SSMS можно разрабатывать базы данных, выполнять инструкции T-SQL, а также администрировать Microsoft SQL Server.

# 2. ОСНОВНАЯ ЧАСТЬ

## 2.1. Проектирование и разработка базы данных

На этапе создания архитектуры базы данных были определены ключевые сущности и их взаимосвязи, которые обеспечивают полноценное функционирование системы.

Основными сущностями являются:

Logins1 (логины) – таблица, предназначенная для хранения данных о клиентах. Она включает следующие атрибуты:

* Login (уникальное имя клиента)
* Parol (пароль для доступа к учётной записи)
* Email (уникальная почта клиента, к которой привязывается учётная запись)

Personal (персонал) – таблица, содержащая информацию о персонале:

* Number (уникальный идентификатор работника)
* FIO (фамилия, имя, отчество работника)
* Podrazdelenie (подразделение организации)
* Dolznost (должность)
* INN (ИНН)
* Adres (адрес)
* Data\_priema (дата приёма работника на работу)
* Semia (информация о семье)
* Obrazovanie (информация об образовании)
* Nagradi (информация о награждениях)

Vacansii1 (вакансии) – таблица, содержащая информацию о вакансиях:

* Number (уникальный идентификатор вакансии)
* Dolznost (должность, указанная в вакансии)
* Uslovia (условия работодателя)

Rezume1 (резюме) – таблица, содержащая данные о резюме потенциальных работников:

* Number (уникальный номер резюме)
* FIO (фамилия, имя, отчество потенциального работника)
* Dolznost (должность, на которую он устраивается)
* Num\_Vac (номер вакансии) – является внешним ключём для таблицы. Связывается с таблицей Vacansii1 по полю Number
* Ssilka (ссылка на полное резюме работника)

Raspisanie1 (расписание) – описание штатного расписания каждого отдела и каждой должности:

* Number (уникальный номер каждой должности)
* Podrazdelenie (подразделение, в котором находится должность)
* Dolznost (должность)
* Kol\_vo (количество работников данной должности)
* Tarif (плата)

На этапе разработки был использован SQL Server Management Studio 19 для реализации схемы. В рамках SQL реализованы таблицы в базе данных Kursa4 (см. рис. 1).

Также для удобства обновления базы данных были созданы процедуры (см. прил. Листинг процедур).

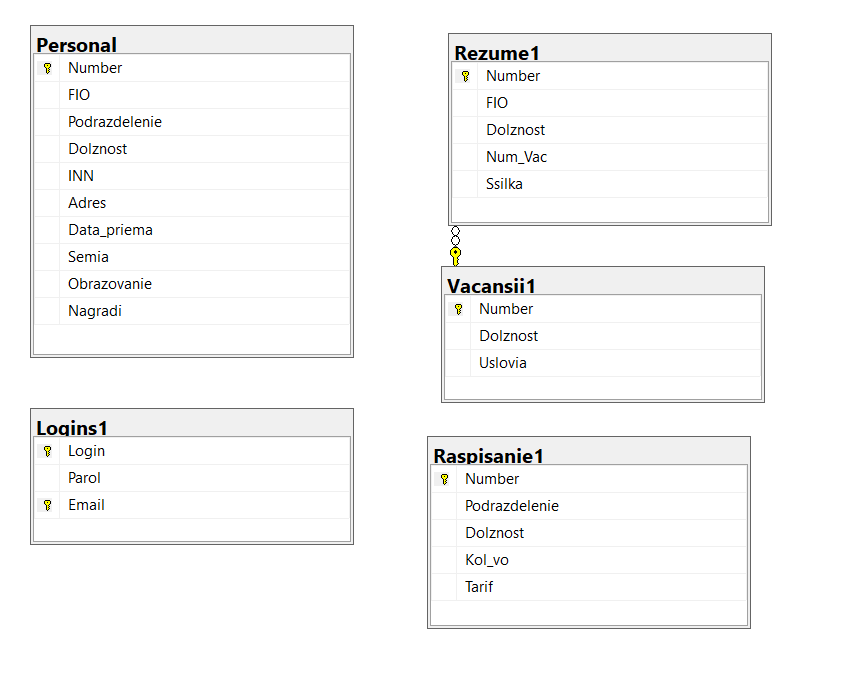


Рис. 1. Диаграмма базы данных Kursa4

## 2.2. Разработка приложения «Учёт персонала»

В проекте предусмотрены следующее окна:

MainWindow: окно для авторизации



Рис. 2. Окно авторизации MainWindow

На окне располагаются кнопки для входа, регистрации нового пользователя и выхода из программы. При нажатии на кнопку «Выйти из программы» программа выключится. При нажатии на кнопку «Регистрация» пользователя перенесёт на окно регистрации. При нажатии на кнопку «Вход» при условии, что введены неверные (незарегистрированные) логин или пароль появляется сообщение с выводом ошибки «Неверные логин или пароль» (см. рис. 3); в случае, если же пользователь вводит верные логин и пароль и нажимает на кнопку, его переносит на окно главного меню.

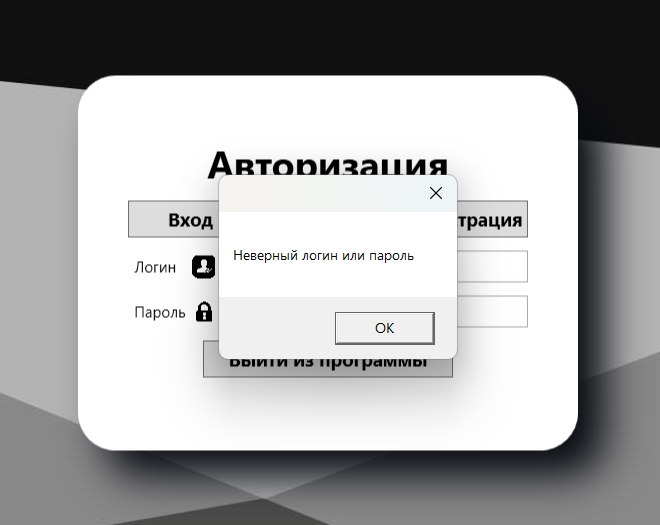


Рис. 3. Сообщение с выводом ошибки «Неверные логин или пароль»

Registracia: окно для регистрации нового пользователя (см. рис. 4).

На окне располагаются кнопки «Регистрация», «Отмена» и поля для ввода логина, пароля и почты. Пользователь заполняет поля для ввода и нажимает на кнопку «Регистрация», после чего его переносит на окно входа. Если пользователь вводит уже зарегистрированные данные, появляется сообщение «Данные логин или почта уже зарегистрированы». Кнопка «Отмена» нужна для перехода на окно входа без сохранения данных (о чём помечено в нижней части окна).

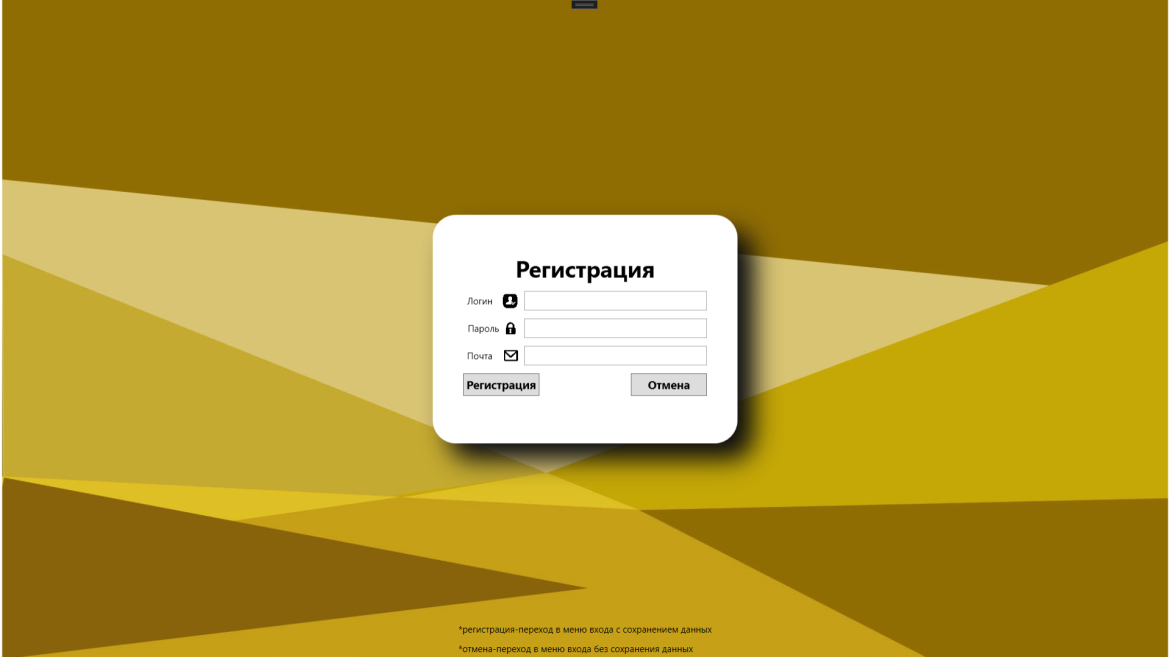


Рис. 4. Окно регистрации нового пользователя Registracia

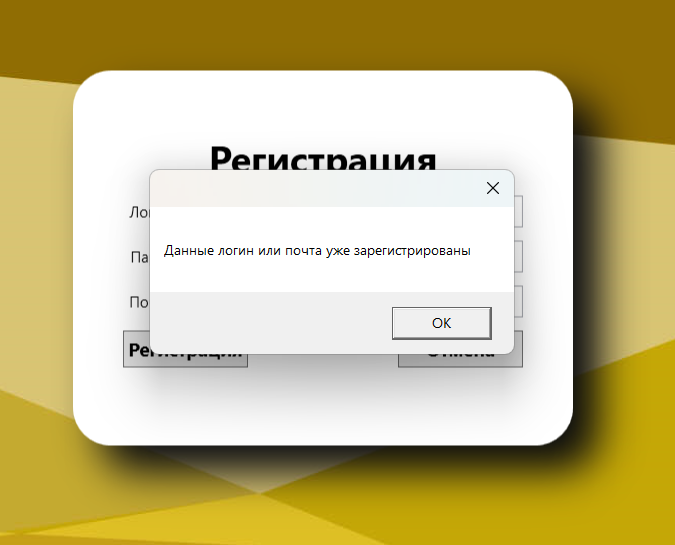


Рис.5. Сообщение «Данные логин или почта уже зарегистрированы»

WindowMenu: окно для вывода главного меню приложения (см. рис. 6)

На окне расположены кнопки «Работники», «Вакансии», «Расписание» и «Выйти из программы». Каждая из кнопок (за исключением последней) переносят пользователя в окно с выбранным функционалом. При нажатии на кнопку «Выйти из программы» программа выключается.

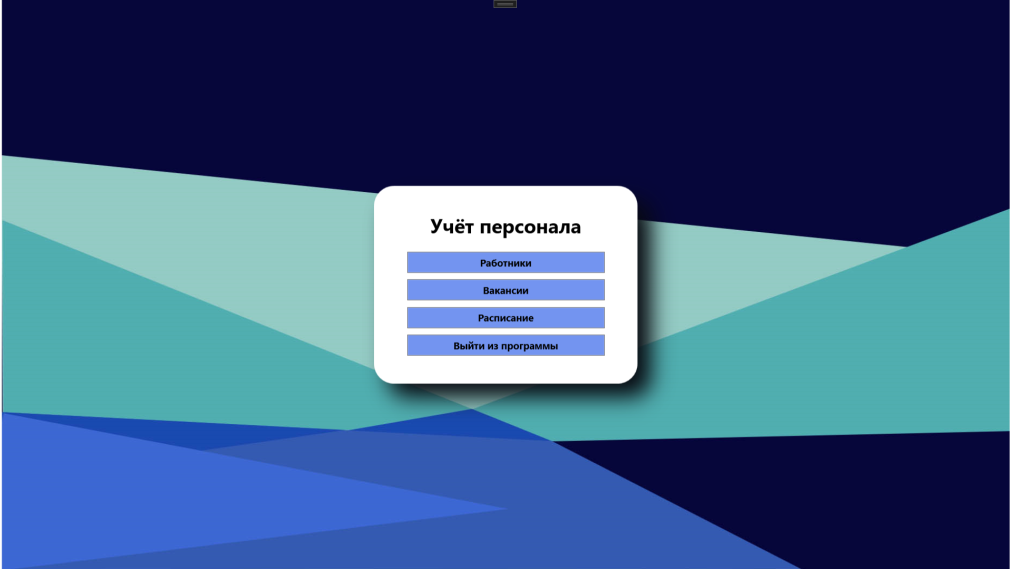


Рис.6. Окно главного меню WindowMenu

WindowWorkers: окно для вывода данных о работниках

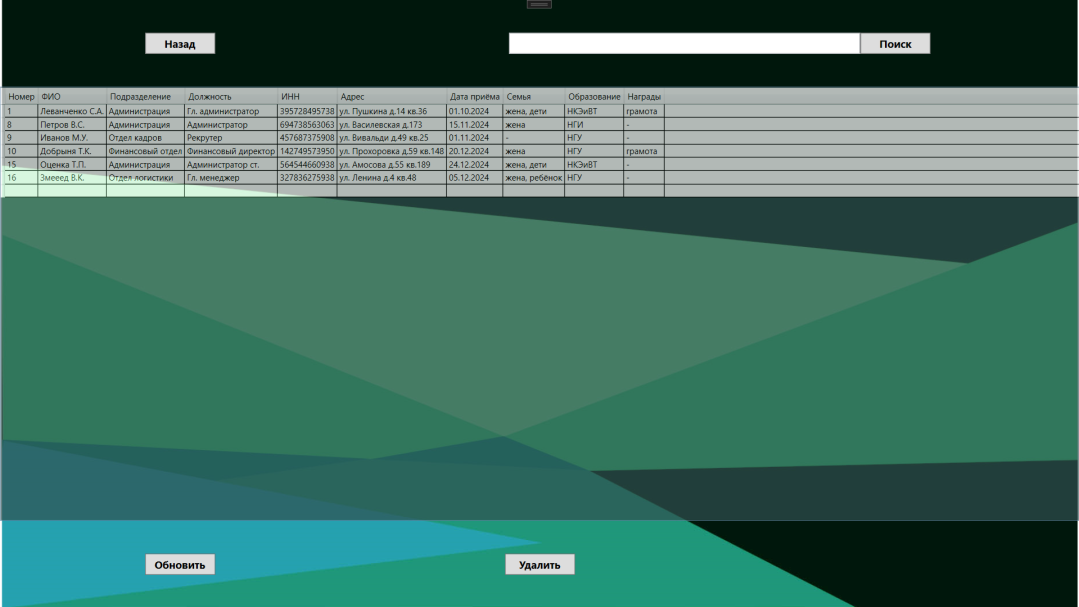


Рис.7. Окно вывода данных о работниках WindowWorkers

На данном окне располагается таблица с данными о работниках, кнопки «Назад», «Обновить», «Удалить» и «Поиск», текстовое поле для ввода искомой информации. Кнопка «Назад» отвечает за возвращение пользователя в главное меню приложения. Для заполнения таблицы новыми данными следует нажать на пустую строку и начать заполнять её. Кнопка «Обновить» отвечает за обновление таблицы. Кнопка «Удалить» отвечает за удаление выбранной строки из таблицы. После заполнения текстового поля и нажатия на кнопку «Поиск» организуется сортировка строк и отображаются только строки, соответствующие данному шаблону.

WindowVacansii: окно для вывода вакансий и резюме

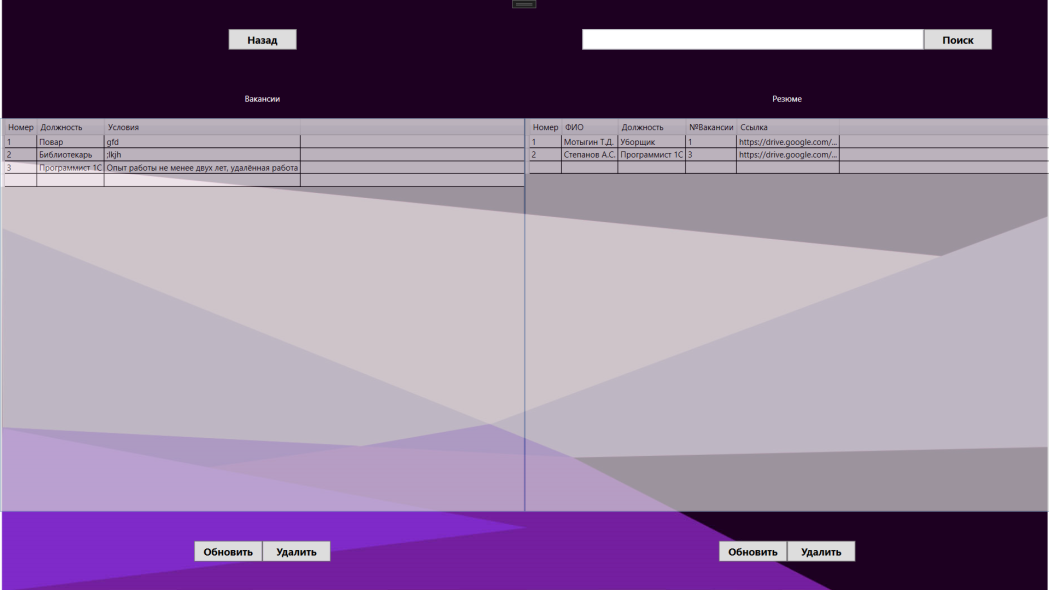


Рис.8. Окно для вывода вакансий и резюме WindowVacansii

На данном окне располагаются кнопки «Назад», «Поиск», «Обновить» (в количестве двух штук) и «Удалить» (также в количестве двух штук), а также текстовое поле для поиска и две таблицы. Для заполнения таблицы новыми данными следует нажать на пустую строку и начать заполнять её. Стоит заметить, что эти две таблицы связаны, и нельзя ввести номер вакансии, если такого номера нет в таблице «Вакансии». Кнопка «Обновить» отвечает за обновление таблицы. Кнопка «Удалить» отвечает за удаление выбранной строки из таблицы. После заполнения текстового поля и нажатия на кнопку «Поиск» организуется сортировка строк и отображаются только строки, соответствующие данному шаблону (поиск организуется в обоих таблицах одновременно).

WindowRaspisanie: окно с выводом данных о расписании (см. рис. 9)

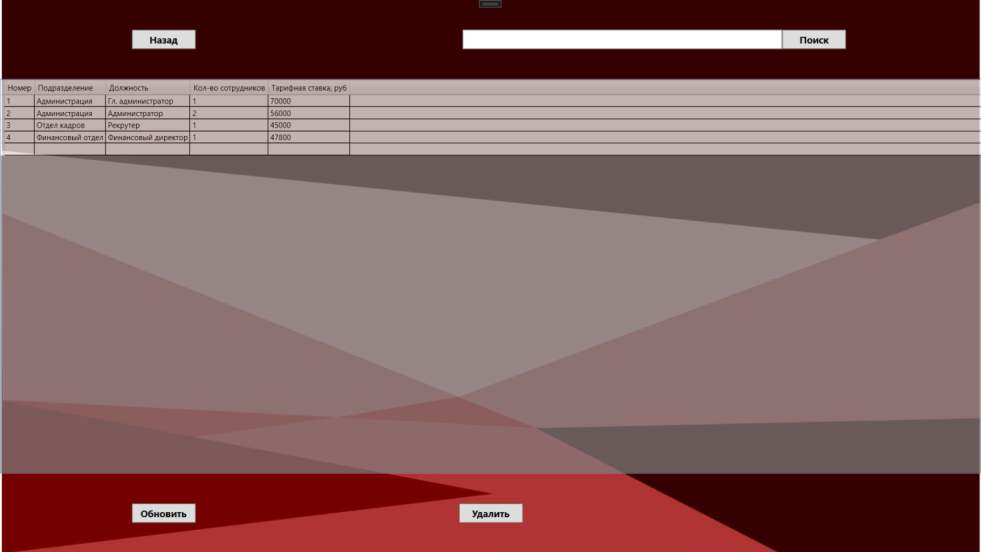


Рис.9. Окно с выводом данных о расписании WindowRaspisanie

На данном окне располагается таблица с данными о работниках, кнопки «Назад», «Обновить», «Удалить» и «Поиск», текстовое поле для ввода искомой информации. Кнопка «Назад» отвечает за возвращение пользователя в главное меню приложения. Для заполнения таблицы новыми данными следует нажать на пустую строку и начать заполнять её. Кнопка «Обновить» отвечает за обновление таблицы. Кнопка «Удалить» отвечает за удаление выбранной строки из таблицы. После заполнения текстового поля и нажатия на кнопку «Поиск» организуется сортировка строк и отображаются только строки, соответствующие данному шаблону.

# 3. ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта на тему «Разработка информационной подсистемы системы по учёту персонала» была создана эффективная и функциональная система, позволяющая автоматизировать процессы учета и управления персоналом в организации.

Разработанная информационная подсистема включает в себя ключевые модули, такие как регистрация сотрудников, управление должностями и подразделениями, управление вакансиями и резюме. Использование современных технологий и подходов к разработке обеспечило высокую производительность и удобство в использовании системы.

В процессе работы над проектом были решены следующие задачи:

* изучена необходимую теория для разработки;
* разработана архитектура информационной подсистемы, включая базы данных и пользовательский интерфейс;
* создан прототип информационной подсистемы с базовым функционалом;
* подготовлен отчёт о проведённой работе и полученных результатах.

В результате использования системы повысилась эффективность управления персоналом, что является важным аспектом для достижения стратегических целей организации.

В будущем планируется расширение функционала системы с учетом отзывов пользователей и внедрение новых технологий, таких как аналитика данных и интеграция с другими информационными системами. Это позволит еще больше улучшить качество учета и управления персоналом, а также обеспечить более глубокий анализ данных.

# СПИСОК ЛИТЕРАТУРЫ

1. Общие сведения о системе по учёту персонала | Как бизнесу вести кадровый учет

- URL: <https://secrets.tinkoff.ru/buhgalteriya/kadroviy-uchet-na-predpriyatii/>

1. Главный инструмент разработчика: что такое IDE, зачем она нужна и как её выбрать

- URL: <https://practicum.yandex.ru/blog/integrirovannaya-sreda-razrabotki-ide>[/](https://practicum.yandex.ru/blog/integrirovannaya-sreda-razrabotki-ide/)

1. Что такое Microsoft Visual Studio

- URL: <https://en.wikipedia.org/wiki/Visual_Studio>

1. Что такое WPF (Windows Presentation Foundation)

- URL: <https://ru.wikipedia.org/wiki/Windows_Presentation_Foundation>

1. Что такое язык программирования C Sharp

- URL: <https://ru.wikipedia.org/wiki/C_Sharp>

1. Что такое база данных (БД): виды, типы, примеры использования и систем управления базами данных

- URL: <https://kokoc.com/blog/baza-dannyh/>

1. Что такое SSMS ( SQL Server Management Studio

- URL: <https://falconspace.ru/list/sqlserver/ustanovka-sql-server-management-studio--->[488](https://falconspace.ru/list/sqlserver/ustanovka-sql-server-management-studio---488)

# ПРИЛОЖЕНИЕ

## Листинг приложения

Листинг MainWindow.xaml

<Window x:Class="учёт\_персонала.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:учёт\_персонала"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800"

WindowState="Maximized"

WindowStyle="None">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Image Source="C:\Users\User\Desktop\курсовая РПМ\визуал\фон вход.jpg" Height="Auto" Width="Auto" Grid.ColumnSpan="3" Grid.RowSpan="6"/>

<Border MinWidth="400" MaxHeight="300" Background="White" HorizontalAlignment="Center" Padding="40" CornerRadius="30" Grid.Column="0" Grid.Row="0" Grid.RowSpan="3" Grid.ColumnSpan="3">

<Border.Effect>

<DropShadowEffect BlurRadius="50" Color="#2E3446" ShadowDepth="30"/>

</Border.Effect>

<StackPanel>

<TextBlock Text="Авторизация" HorizontalAlignment="Center" FontSize="30" FontWeight="Bold" Margin="0 10 0 10"/>

<Grid>

<Button Name="But\_vxod" Content="Вход" HorizontalAlignment="Left" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="But\_vxod\_Click"/>

<Button Name="But\_reg" Content="Регистрация" HorizontalAlignment="Right" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="But\_reg\_Click"/>

</Grid>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="3\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal">

<Label Content="Логин" Margin="0 10 0 0"/>

<Image Source="C:\Users\User\Desktop\курсовая РПМ\визуал\login.png" Height="20" Width="20" Margin="7 10 0 0"/>

</StackPanel>

<TextBox x:Name="tb\_Avtor" Grid.Column="1" Margin="0 10 0 0"/>

</Grid>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="3\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal">

<Label Content="Пароль" Margin="0 10 0 0"/>

<Image Source="C:\Users\User\Desktop\курсовая РПМ\визуал\Lock\_font\_awesome.svg.png" Height="20" Width="20" Margin="0 10 0 0"/>

</StackPanel>

<PasswordBox x:Name="pb\_Avtor" Grid.Column="1" Margin="0 10 0 0"/>

</Grid>

<Button Name="But\_Close\_Prog" Content="Выйти из программы" FontSize="15" FontWeight="Bold" Width="200" Height="30" Margin="0 10 0 0" Click="But\_Close\_Prog\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

Листинг MainWindow.xaml.cs

using System.Windows;

using System.Configuration;

using System.Data.SqlClient;

using System.Data;

namespace учёт\_персонала

{

public partial class MainWindow : Window

{

string connectionString;

SqlDataAdapter adapter;

DataTable connectionTable;

public MainWindow()

{

InitializeComponent();

connectionString = ConfigurationManager.ConnectionStrings["CONNECTION"].ConnectionString;

}

private void But\_Close\_Prog\_Click(object sender, RoutedEventArgs e)

{

System.Windows.Application.Current.Shutdown();

}

private void But\_vxod\_Click(object sender, RoutedEventArgs e)

{

DataTable dts = Select("SELECT \* FROM [dbo].[Logins1] WHERE [Login] = '" + tb\_Avtor.Text + "' AND [Parol] = '" + pb\_Avtor.Password + "'");

if (dts.Rows.Count > 0)

{

WindowMenu winMenu = new WindowMenu();

this.Visibility = Visibility.Collapsed;

winMenu.ShowDialog();

}

else

{

MessageBox.Show("Неверный логин или пароль");

}

}

private void But\_reg\_Click(object sender, RoutedEventArgs e)

{

Registracia winRegist = new Registracia();

this.Visibility = Visibility.Collapsed;

winRegist.ShowDialog();

}

public DataTable Select(string zapros)

{

connectionTable = new DataTable();

SqlConnection connection = null;

connection = new SqlConnection(connectionString);

SqlCommand command = new SqlCommand(zapros, connection);

adapter = new SqlDataAdapter(command);

connection.Open();

adapter.Fill(connectionTable);

return connectionTable;

}

}

}

Листинг WindowMenu.xaml

<Window x:Class="учёт\_персонала.WindowMenu"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:учёт\_персонала"

mc:Ignorable="d"

Title="WindowMenu" Height="450" Width="800"

WindowState="Maximized"

WindowStyle="None">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Image Source="C:\Users\User\Desktop\курсовая РПМ\визуал\фон меню.jpg" Height="Auto" Width="Auto" Grid.ColumnSpan="3" Grid.RowSpan="6"/>

<Border MinWidth="400" MaxHeight="300" Background="White" HorizontalAlignment="Center" Padding="40" CornerRadius="30" Grid.Column="0" Grid.Row="1" Grid.RowSpan="4" Grid.ColumnSpan="3">

<Border.Effect>

<DropShadowEffect BlurRadius="50" Color="#2E3446" ShadowDepth="30"/>

</Border.Effect>

<StackPanel>

<TextBlock Text="Учёт персонала" HorizontalAlignment="Center" FontSize="30" FontWeight="Bold" Margin="0 0 0 20"/>

<Button Name="But\_Rab" Content="Работники" Click="But\_Rab\_Click" Background="#7394F0" FontSize="15" FontWeight="Bold" Padding="5" Margin="10 0 10 10"/>

<Button Name="But\_Vak" Content="Вакансии" Click="But\_Vak\_Click" Background="#7394F0" FontSize="15" FontWeight="Bold" Padding="5" Margin="10 0 10 10"/>

<Button Name="But\_Rasp" Content="Расписание" Click="But\_Rasp\_Click" Background="#7394F0" FontSize="15" FontWeight="Bold" Padding="5" Margin="10 0 10 10"/>

<Button Name="But\_Close\_Prog" Content="Выйти из программы" Click="But\_Close\_Prog\_Click" Background="#7394F0" FontSize="15" FontWeight="Bold" Padding="5" Margin="10 0 10 10"/>

</StackPanel>

</Border>

</Grid>

</Window>

Листинг WindowMenu.xaml.cs

using System.Windows;

namespace учёт\_персонала

{

public partial class WindowMenu : Window

{

public WindowMenu()

{

InitializeComponent();

}

private void But\_Rab\_Click(object sender, RoutedEventArgs e)

{

WindowWorkers winWor = new WindowWorkers();

this.Visibility = Visibility.Collapsed;

winWor.ShowDialog();

}

private void But\_Vak\_Click(object sender, RoutedEventArgs e)

{

WindowVacansii winVac = new WindowVacansii();

this.Visibility = Visibility.Collapsed;

winVac.ShowDialog();

}

private void But\_Rasp\_Click(object sender, RoutedEventArgs e)

{

WindowRaspisanie winRasp = new WindowRaspisanie();

this.Visibility = Visibility.Collapsed;

winRasp.ShowDialog();

}

private void But\_Close\_Prog\_Click(object sender, RoutedEventArgs e)

{

System.Windows.Application.Current.Shutdown();

}

}

}

Листинг WindowWorkers.xaml

<Window x:Class="учёт\_персонала.WindowWorkers"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:учёт\_персонала"

Title="WindowRab" Height="450" Width="800"

WindowState="Maximized"

WindowStyle="None" Loaded="Window\_Loaded">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="5\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Image Source="C:\Users\User\Desktop\курсовая РПМ\визуал\фон персонал.jpg" Height="Auto" Width="Auto" Grid.ColumnSpan="3" Grid.RowSpan="6"/>

<Border>

<Button Name="But\_Nazad1" Content="Назад" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="But\_Nazad1\_Click"/>

</Border>

<DataGrid AutoGenerateColumns="False" x:Name="TABLEPersonal" Grid.Column="0" Grid.Row="1" Grid.ColumnSpan="3" Opacity="0.7" Background="DarkSlateGray">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Number}" Header="Номер" Width="Auto" />

<DataGridTextColumn Binding="{Binding FIO}" Header="ФИО" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Podrazdelenie}" Header="Подразделение" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Dolznost}" Header="Должность" Width="Auto"/>

<DataGridTextColumn Binding="{Binding INN}" Header="ИНН" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Adres}" Header="Адрес" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Data\_priema}" Header="Дата приёма" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Semia}" Header="Семья" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Obrazovanie}" Header="Образование" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Nagradi}" Header="Награды" Width="Auto"/>

</DataGrid.Columns>

</DataGrid>

<Button x:Name="updateButton\_Per" Content="Обновить" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="updateButton\_Per\_Click" Grid.Row="2" Grid.Column="0"/>

<Button x:Name="deleteButton\_Per" Content="Удалить" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="deleteButton\_Per\_Click" Grid.Row="2" Grid.Column="1"/>

<StackPanel Grid.Column="1" Grid.ColumnSpan="2" Orientation="Horizontal" HorizontalAlignment="Center">

<TextBox x:Name="tb\_Poisk" Width="500" Height="30"/>

<Button x:Name="Poisk" Content="Поиск" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="Poisk\_Click"/>

</StackPanel>

</Grid>

</Window>

Листинг WindowWorkers.xaml.cs

using System;

using System.Data.SqlClient;

using System.Data;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Configuration;

namespace учёт\_персонала

{

public partial class WindowWorkers : Window

{

string connectionString;

SqlDataAdapter adapter;

DataTable Personal\_dt;

public WindowWorkers()

{

InitializeComponent();

connectionString = ConfigurationManager.ConnectionStrings["CONNECTION"].ConnectionString;

TABLEPersonal.RowEditEnding += TABLEPersonal\_RowEditEnding;

}

private void But\_Nazad1\_Click(object sender, RoutedEventArgs e)

{

WindowMenu winMenu = new WindowMenu();

this.Visibility = Visibility.Collapsed;

winMenu.ShowDialog();

}

private void UpdateDB()

{

SqlCommandBuilder comandbuilder = new SqlCommandBuilder(adapter);

adapter.Update(Personal\_dt);

}

private void updateButton\_Per\_Click(object sender, RoutedEventArgs e)

{

UpdateDB();

}

private void TABLEPersonal\_RowEditEnding(object sender, DataGridRowEditEndingEventArgs e)

{

UpdateDB();

}

private void deleteButton\_Per\_Click(object sender, RoutedEventArgs e)

{

if (TABLEPersonal.SelectedItems != null)

{

for (int i = 0; i < TABLEPersonal.SelectedItems.Count; i++)

{

DataRowView datarowView = TABLEPersonal.SelectedItems[i] as DataRowView;

if (datarowView != null)

{

DataRow dataRow = (DataRow)datarowView.Row;

dataRow.Delete();

}

}

}

UpdateDB();

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

string sql = "SELECT \* FROM Personal";

Personal\_dt = new DataTable();

SqlConnection connection = null;

try

{

connection = new SqlConnection(connectionString);

SqlCommand command = new SqlCommand(sql, connection);

adapter = new SqlDataAdapter(command);

adapter.InsertCommand = new SqlCommand("INSERTPersonal2", connection);

adapter.InsertCommand.CommandType = CommandType.StoredProcedure;

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Number", SqlDbType.Int, 0, "Number"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@FIO", SqlDbType.NVarChar, 50, "FIO"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Podrazdelenie", SqlDbType.NVarChar, 50, "Podrazdelenie"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Dolznost", SqlDbType.NVarChar, 50, "Dolznost"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@INN", SqlDbType.NVarChar, 50, "INN"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Adres", SqlDbType.NVarChar, 50, "Adres"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Data\_priema", SqlDbType.NVarChar, 50, "Data\_priema"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Semia", SqlDbType.NVarChar, 250, "Semia"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Obrazovanie", SqlDbType.NVarChar, 250, "Obrazovanie"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Nagradi", SqlDbType.NVarChar, 500, "Nagradi"));

SqlParameter parameter = adapter.InsertCommand.Parameters.Add("@id", SqlDbType.Int, 0, "Id");

parameter.Direction = ParameterDirection.Output;

connection.Open();

adapter.Fill(Personal\_dt);

TABLEPersonal.ItemsSource = Personal\_dt.DefaultView;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

if (connection != null)

connection.Close();

}

}

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

{

Personal\_dt.DefaultView.RowFilter = string.Format("FIO LIKE '%{0}%' OR Podrazdelenie LIKE '%{0}%' OR Dolznost LIKE '%{0}%' OR Adres LIKE '%{0}%' " +

"OR Data\_priema LIKE '%{0}%' OR Semia LIKE '%{0}%' OR Nagradi LIKE '%{0}%' OR INN LIKE '%{0}%'", tb\_Poisk.Text);

}

}

}

Листинг WindowVacansii.xaml

<Window x:Class="учёт\_персонала.WindowVacansii"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:учёт\_персонала"

mc:Ignorable="d"

Title="WindowVacansii" Height="450" Width="800"

WindowState="Maximized"

WindowStyle="None" Loaded="Window\_Loaded">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="0.5\*"></RowDefinition>

<RowDefinition Height="5\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Image Source="C:\Users\User\Desktop\курсовая РПМ\визуал\фон вакансии.jpg" Height="Auto" Width="Auto" Grid.ColumnSpan="3" Grid.RowSpan="6"/>

<Button Name="Nazad2" Content="Назад" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="Nazad2\_Click"/>

<StackPanel Grid.Row="3" Grid.Column="0" Orientation="Horizontal" HorizontalAlignment="Center">

<Button x:Name="updateButton\_Vac" Content="Обновить" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="updateButton\_Vac\_Click"/>

<Button x:Name="deleteButton\_Vac" Content="Удалить" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="deleteButton\_Vac\_Click"/>

</StackPanel>

<StackPanel Grid.Row="3" Grid.Column="1" VerticalAlignment="Center" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal">

<Button x:Name="updateButton\_Vac2" Content="Обновить" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="updateButton\_Rez\_Click"/>

<Button x:Name="deleteButton\_Vac2" Content="Удалить" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="deleteButton\_Rez\_Click"/>

</StackPanel>

</StackPanel>

<Label Name="labVac" Content="Вакансии" Grid.Row="1" Grid.Column="0" VerticalAlignment="Center" HorizontalAlignment="Center" Foreground="White">

</Label>

<Label Name="labVac2" Content="Резюме" Grid.Row="1" Grid.Column="1" VerticalAlignment="Center" HorizontalAlignment="Center" Foreground="White"/>

<DataGrid AutoGenerateColumns="False" x:Name="TABLEVacansii" Grid.Column="0" Grid.Row="2" Opacity="0.7" Background="LightGray">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Number}" Header="Номер" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Dolznost}" Header="Должность" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Uslovia}" Header="Условия" Width="Auto"/>

</DataGrid.Columns>

</DataGrid>

<DataGrid AutoGenerateColumns="False" x:Name="TABLERezume" Grid.Column="1" Grid.Row="2" Opacity="0.7" Background="LightGray">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Number}" Header="Номер" Width="Auto"/>

<DataGridTextColumn Binding="{Binding FIO}" Header="ФИО" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Dolznost}" Header="Должность" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Num\_Vac}" Header="№Вакансии" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Ssilka}" Header="Ссылка" Width="Auto"/>

</DataGrid.Columns>

</DataGrid>

<StackPanel Grid.Column="1" Grid.ColumnSpan="2" Orientation="Horizontal" HorizontalAlignment="Center">

<TextBox x:Name="tb\_Poisk" Width="500" Height="30"/>

<Button x:Name="Poisk" Content="Поиск" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="Poisk\_Click"/>

</StackPanel>

</Grid>

</Window>

Листинг WindowVacansii.xaml.cs

using System;

using System.Data.SqlClient;

using System.Data;

using System.Windows;

using System.Configuration;

namespace учёт\_персонала

{

public partial class WindowVacansii : Window

{

string connectionString;

SqlDataAdapter adapter;

SqlDataAdapter adapter1;

DataTable Vac\_tb;

DataTable Rez\_tb;

public WindowVacansii()

{

InitializeComponent();

connectionString = ConfigurationManager.ConnectionStrings["CONNECTION"].ConnectionString;

}

private void Nazad2\_Click(object sender, RoutedEventArgs e)

{

WindowMenu winMenu = new WindowMenu();

this.Visibility = Visibility.Collapsed;

winMenu.ShowDialog();

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

string sql = "SELECT \* FROM Vacansii1";

string sql2 = "SELECT \* FROM Rezume1";

Vac\_tb = new DataTable();

Rez\_tb = new DataTable();

SqlConnection connection = null;

try

{

connection = new SqlConnection(connectionString);

SqlCommand command = new SqlCommand(sql, connection);

adapter = new SqlDataAdapter(command);

adapter.InsertCommand = new SqlCommand("INSERTVac1", connection);

adapter.InsertCommand.CommandType = CommandType.StoredProcedure;

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Number", SqlDbType.Int, 0, "Number"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Dolznost", SqlDbType.NVarChar, 50, "Dolznost"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Uslovia", SqlDbType.NVarChar, 500, "Uslovia"));

SqlParameter parameter = adapter.InsertCommand.Parameters.Add("@Id", SqlDbType.Int, 0, "Id");

parameter.Direction = ParameterDirection.Output;

connection.Open();

adapter.Fill(Vac\_tb);

TABLEVacansii.ItemsSource = Vac\_tb.DefaultView;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

if (connection != null)

connection.Close();

}

try

{

connection = new SqlConnection(connectionString);

SqlCommand command = new SqlCommand(sql2, connection);

adapter1 = new SqlDataAdapter(command);

adapter1.InsertCommand = new SqlCommand("INSERTRez1", connection);

adapter1.InsertCommand.CommandType = CommandType.StoredProcedure;

adapter1.InsertCommand.Parameters.Add(new SqlParameter("@Number", SqlDbType.Int, 0, "Number"));

adapter1.InsertCommand.Parameters.Add(new SqlParameter("@FIO", SqlDbType.NVarChar, 50, "FIO"));

adapter1.InsertCommand.Parameters.Add(new SqlParameter("@Dolznost", SqlDbType.NVarChar, 50, "Dolznost"));

adapter1.InsertCommand.Parameters.Add(new SqlParameter("@Num\_Vac", SqlDbType.Int, 0, "Num\_Vac"));

adapter1.InsertCommand.Parameters.Add(new SqlParameter("@Ssilka", SqlDbType.NVarChar, 100, "Ssilka"));

SqlParameter parameter2 = adapter1.InsertCommand.Parameters.Add("@Id", SqlDbType.Int, 0, "Id");

parameter2.Direction = ParameterDirection.Output;

connection.Open();

adapter1.Fill(Rez\_tb);

TABLERezume.ItemsSource = Rez\_tb.DefaultView;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

if (connection != null)

connection.Close();

}

}

private void UpdateDB()

{

SqlCommandBuilder comandbuilder = new SqlCommandBuilder(adapter);

adapter.Update(Vac\_tb);

}

private void UpdateDB2()

{

SqlCommandBuilder comandbuilder = new SqlCommandBuilder(adapter1);

adapter1.Update(Rez\_tb);

}

private void updateButton\_Rez\_Click(object sender, RoutedEventArgs e)

{

UpdateDB2();

}

private void deleteButton\_Rez\_Click(object sender, RoutedEventArgs e)

{

if (TABLERezume.SelectedItems != null)

{

for (int i = 0; i < TABLERezume.SelectedItems.Count; i++)

{

DataRowView datarowView = TABLERezume.SelectedItems[i] as DataRowView;

if (datarowView != null)

{

DataRow dataRow = (DataRow)datarowView.Row;

dataRow.Delete();

}

}

}

UpdateDB2();

}

private void updateButton\_Vac\_Click(object sender, RoutedEventArgs e)

{

UpdateDB();

}

private void deleteButton\_Vac\_Click(object sender, RoutedEventArgs e)

{

if (TABLEVacansii.SelectedItems != null)

{

for (int i = 0; i < TABLEVacansii.SelectedItems.Count; i++)

{

DataRowView datarowView = TABLEVacansii.SelectedItems[i] as DataRowView;

if (datarowView != null)

{

DataRow dataRow = (DataRow)datarowView.Row;

dataRow.Delete();

}

}

}

UpdateDB();

}

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

{

Vac\_tb.DefaultView.RowFilter = string.Format("Dolznost LIKE '%{0}%' OR Uslovia LIKE '%{0}%' ", tb\_Poisk.Text);

Rez\_tb.DefaultView.RowFilter = string.Format("FIO LIKE '%{0}%' OR Dolznost LIKE '%{0}%' OR Ssilka LIKE '%{0}%' ", tb\_Poisk.Text);

}

}

}

Листинг WindowRaspisanie.xaml

<Window x:Class="учёт\_персонала.WindowRaspisanie"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:учёт\_персонала"

mc:Ignorable="d"

Title="WindowRaspisanie" Height="450" Width="800"

WindowState="Maximized"

WindowStyle="None" Loaded="Window\_Loaded">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="5\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Image Source="C:\Users\User\Desktop\курсовая РПМ\визуал\фон расписание.jpg" Height="Auto" Width="Auto" Grid.ColumnSpan="3" Grid.RowSpan="6"/>

<DataGrid AutoGenerateColumns="False" x:Name="TABLERaspisanie" Grid.Column="0" Grid.Row="1" Grid.ColumnSpan="3" Opacity="0.7" Background="Gray">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Number}" Header="Номер" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Podrazdelenie}" Header="Подразделение" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Dolznost}" Header="Должность" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Kol\_vo}" Header="Кол-во сотрудников" Width="Auto"/>

<DataGridTextColumn Binding="{Binding Tarif}" Header="Тарифная ставка, руб" Width="Auto"/>

</DataGrid.Columns>

</DataGrid>

<Button x:Name="Nazad3" Content="Назад" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="Nazad3\_Click"/>

<Button x:Name="updateButton\_Rasp" Content="Обновить" Grid.Row="2" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="updateButton\_Rasp\_Click"/>

<Button x:Name="deleteButton\_Rasp" Content="Удалить" Grid.Row="2" Grid.Column="1" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="deleteButton\_Rasp\_Click"/>

<StackPanel Grid.Column="1" Grid.ColumnSpan="2" Orientation="Horizontal" HorizontalAlignment="Center">

<TextBox x:Name="tb\_Poisk" Width="500" Height="30"/>

<Button x:Name="Poisk" Content="Поиск" FontSize="15" FontWeight="Bold" Width="100" Height="30" Click="Poisk\_Click"/>

</StackPanel>

</Grid>

</Window>

Листинг WindowRaspisanie.xaml.cs

using System;

using System.Windows;

using System.Configuration;

using System.Data.SqlClient;

using System.Data;

namespace учёт\_персонала

{

public partial class WindowRaspisanie : Window

{

string connectionString;

SqlDataAdapter adapter;

DataTable Rasp\_tb;

public WindowRaspisanie()

{

InitializeComponent();

connectionString = ConfigurationManager.ConnectionStrings["CONNECTION"].ConnectionString;

}

private void Nazad3\_Click(object sender, RoutedEventArgs e)

{

WindowMenu winMenu = new WindowMenu();

this.Visibility = Visibility.Collapsed;

winMenu.ShowDialog();

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

string sql = "SELECT \* FROM Raspisanie1";

Rasp\_tb = new DataTable();

SqlConnection connection = null;

try

{

connection = new SqlConnection(connectionString);

SqlCommand command = new SqlCommand(sql, connection);

adapter = new SqlDataAdapter(command);

adapter.InsertCommand = new SqlCommand("INSERTRas1", connection);

adapter.InsertCommand.CommandType = CommandType.StoredProcedure;

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Number", SqlDbType.Int, 0, "Number"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Podrazdelenie", SqlDbType.NVarChar, 100, "Podrazdelenie"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Dolznost", SqlDbType.NVarChar, 50, "Dolznost"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Kol\_vo", SqlDbType.Int, 0, "Kol\_vo"));

adapter.InsertCommand.Parameters.Add(new SqlParameter("@Tarif", SqlDbType.Float, 0, "Tarif"));

SqlParameter parameter = adapter.InsertCommand.Parameters.Add("@Id", SqlDbType.Int, 0, "Id");

parameter.Direction = ParameterDirection.Output;

connection.Open();

adapter.Fill(Rasp\_tb);

TABLERaspisanie.ItemsSource = Rasp\_tb.DefaultView;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

if (connection != null)

connection.Close();

}

}

private void UpdateDB()

{

SqlCommandBuilder comandbuilder = new SqlCommandBuilder(adapter);

adapter.Update(Rasp\_tb);

}

private void updateButton\_Rasp\_Click(object sender, RoutedEventArgs e)

{

UpdateDB();

}

private void deleteButton\_Rasp\_Click(object sender, RoutedEventArgs e)

{

if (TABLERaspisanie.SelectedItems != null)

{

for (int i = 0; i < TABLERaspisanie.SelectedItems.Count; i++)

{

DataRowView datarowView = TABLERaspisanie.SelectedItems[i] as DataRowView;

if (datarowView != null)

{

DataRow dataRow = (DataRow)datarowView.Row;

dataRow.Delete();

}

}

}

UpdateDB();

}

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

{

Rasp\_tb.DefaultView.RowFilter = string.Format("Podrazdelenie LIKE '%{0}%' OR Dolznost LIKE '%{0}%'", tb\_Poisk.Text);

}

}

}

Листинг Registracia.xaml

<Window x:Class="учёт\_персонала.Registracia"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:учёт\_персонала"

mc:Ignorable="d"

Title="Registracia" Height="450" Width="800"

WindowState="Maximized"

WindowStyle="None">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Image Source="C:\Users\User\Desktop\курсовая РПМ\визуал\фон регистрация.jpg" Height="Auto" Width="Auto" Grid.ColumnSpan="3" Grid.RowSpan="6"/>

<Border MinWidth="400" MaxHeight="300" Background="White" HorizontalAlignment="Center" Padding="40" CornerRadius="30" Grid.Column="0" Grid.Row="0" Grid.RowSpan="4" Grid.ColumnSpan="3">

<Border.Effect>

<DropShadowEffect BlurRadius="50" Color="#2E3446" ShadowDepth="30"/>

</Border.Effect>

<StackPanel>

<TextBlock Text="Регистрация" HorizontalAlignment="Center" FontSize="30" FontWeight="Bold" Margin="0 10 0 10"/>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="3\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal">

<Label Content="Логин"/>

<Image Source="C:\Users\User\Desktop\курсовая РПМ\визуал\login.png" Height="20" Width="20" Margin="8 0 0 0"/>

</StackPanel>

<TextBox x:Name="tb\_log" Grid.Column="1" />

</Grid>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="3\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal">

<Label Content="Пароль" Margin="1 10 0 0"/>

<Image Source="C:\Users\User\Desktop\курсовая РПМ\визуал\Lock\_font\_awesome.svg.png" Height="20" Width="20" Margin="0 10 0 0"/>

</StackPanel>

<TextBox x:Name="tb\_reg" Grid.Column="1" Margin="0 10 0 0"/>

</Grid>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="3\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal">

<Label Content="Почта" Margin="0 10 0 0"/>

<Image Source="C:\Users\User\Pictures\Screenshots\Снимок экрана 2024-12-24 172103.png" Height="20" Width="20" Margin="9 10 0 0"/>

</StackPanel>

<TextBox x:Name="tb\_poch" Grid.Column="1" Margin="0 10 0 0"/>

</Grid>

<Grid>

<Button Name="But\_Reg" Content="Регистрация" FontSize="15" FontWeight="Bold" Width="100" Height="30" HorizontalAlignment="Left" Margin="0 10 0 0" Click="But\_Reg\_Click"/>

<Button Name="But\_Otmena" Content="Отмена" FontSize="15" FontWeight="Bold" Width="100" Height="30" HorizontalAlignment="Right" Margin="0 10 0 0" Click="But\_Otmena\_Click"/>

</Grid>

</StackPanel>

</Border>

<StackPanel Grid.Row="3" Grid.ColumnSpan="3" VerticalAlignment="Bottom" HorizontalAlignment="Center">

<Label Content="\*регистрация-переход в меню входа с сохранением данных"/>

<Label Content="\*отмена-переход в меню входа без сохранения данных"/>

</StackPanel>

</Grid>

</Window>

Листинг Registracia.xaml.cs

using System;

using System.Windows;

using System.Configuration;

using System.Data.SqlClient;

using System.Data;

namespace учёт\_персонала

{

public partial class Registracia : Window

{

string connectionString;

SqlDataAdapter adapter;

DataTable connectionTable;

public Registracia()

{

InitializeComponent();

connectionString = ConfigurationManager.ConnectionStrings["CONNECTION"].ConnectionString;

}

public DataTable Select(string zapros)

{

connectionTable = new DataTable();

SqlConnection connection = null;

connection = new SqlConnection(connectionString);

SqlCommand command = new SqlCommand(zapros, connection);

adapter = new SqlDataAdapter(command);

connection.Open();

adapter.Fill(connectionTable);

return connectionTable;

}

private void But\_Otmena\_Click(object sender, RoutedEventArgs e)

{

MainWindow winMain = new MainWindow();

this.Visibility = Visibility.Collapsed;

winMain.ShowDialog();

}

private void But\_Reg\_Click(object sender, RoutedEventArgs e)

{

DataTable dts = Select("SELECT \* FROM [dbo].[Logins1] WHERE [Login] = '" + tb\_log.Text + "' OR [Email] = '" + tb\_poch.Text + "' ");

if (dts.Rows.Count > 0)

{

MessageBox.Show("Данные логин или почта уже зарегистрированы");

}

else

{

dts = Select("INSERT INTO Logins1 (Login,Parol,Email) VALUES ('" + tb\_log.Text + "', '" + tb\_reg.Text + "','" + tb\_poch.Text + "')");

MainWindow winMain = new MainWindow();

this.Visibility = Visibility.Collapsed;

winMain.ShowDialog();

}

}

}

}

## Листинг процедур

Процедура INSERTPersonal2

ALTER PROCEDURE [dbo].[INSERTPersonal2]

@Number int,

@FIO nvarchar(50),

@Podrazdelenie nvarchar(50),

@Dolznost nvarchar(50),

@INN int,

@Adres nvarchar(250),

@Data\_priema nvarchar(50),

@Semia nvarchar(250),

@Obrazovanie nvarchar(250),

@Nagradi nvarchar(500),

@id Int out

as

insert into Personal (FIO,Podrazdelenie,Dolznost,INN,Adres,Data\_priema,Semia,Obrazovanie,Nagradi)

values (@FIO,@Podrazdelenie,@Dolznost,@INN,@Adres,@Data\_priema,@Semia,@Obrazovanie,@Nagradi)

set @id=SCOPE\_IDENTITY()

Процедура INSERTSRas1

ALTER PROCEDURE [dbo].[INSERTRas1]

@Number int,

@Podrazdelenie nvarchar(100),

@Dolznost nvarchar(50),

@Kol\_vo int,

@Tarif float,

@Id int out

AS

INSERT INTO Raspisanie1(Number,Podrazdelenie, Dolznost, Kol\_vo, Tarif)

VALUES (@Number,@Podrazdelenie, @Dolznost, @Kol\_vo, @Tarif)

SET @Id=SCOPE\_IDENTITY()

Процедура INSERTRez1

ALTER PROCEDURE [dbo].[INSERTRez1]

@Number int,

@FIO nvarchar(50),

@Dolznost nvarchar(50),

@Num\_Vac int,

@Ssilka nvarchar(100),

@Id int out

AS

INSERT INTO Rezume1(Number,FIO,Dolznost,Num\_Vac,Ssilka)

VALUES (@Number, @FIO,@Dolznost,@Num\_Vac,@Ssilka)

SET @Id=SCOPE\_IDENTITY()

Процедура INSERTVac1

ALTER PROCEDURE [dbo].[INSERTVac1]

@Number int,

@Dolznost nvarchar(50),

@Uslovia nvarchar(500),

@Id int out

AS

INSERT INTO Vacansii1(Number,Dolznost,Uslovia)

VALUES (@Number, @Dolznost, @Uslovia)

SET @Id=SCOPE\_IDENTITY()