|  |
| --- |
| титул |

полное название образовательного учреждения

|  |
| --- |
| **ДОПУСКАЮ К ЗАЩИТЕ** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Заместитель директора по УПР О.В. Корешков  14 июня 2023 года |
| (дата) |
|  |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

|  |
| --- |
| Разработка информационной системы подбора, найма и сопровождения трудовых ресурсов |

(тема)

Выпускная квалификационная работа должна быть выполнена в виде:

дипломной работы и демонстрационного экзамена

|  |  |
| --- | --- |
| студентом группы | 3ИСП11-16 |
|  | (номер группы) |
| Никитой Валерьевичем Симоновым | |

(И.О.Фамилия) (подпись, дата)

Основная профессиональная образовательная программа по специальности 09.02.07 Информационные системы и программирование

(шифр и наименование специальности)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Форма обучения | | очная | | |
| Руководитель |  | преподаватель |  | Щур Даниил Сергеевич |
|  | (ученая степень, должность, И.О.Фамилия) | | |  |
|  |  | | | (подпись, дата) |

|  |  |  |
| --- | --- | --- |
| Председатель предметной (междисциплинарной, модульной) комиссии | | |
| Кирилл Михайлович Бастрыкин |  |  |
| (И. О. Фамилия) |  | (подпись, дата) |

Москва

2023

|  |
| --- |
| титул |

полное название образовательного учреждения

|  |
| --- |
| **УТВЕРЖДАЮ** |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| Заместитель директора по УПР О.В. Корешков  04 апреля 2023 года |
| (дата) |
| **ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Студенту | Никите Валерьевичу Симонову | | | | | (фамилия, имя, отчество полностью) | | | | | | I. Тема выпускной квалификационной работы | | | *Разработка информационной системы подбора, найма и сопровождения трудовых ресурсов* | | | II. Срок сдачи студентом законченной работы | | | | 07 июня 2023 г. | | III. Исходные данные | | *Программные средства: Microsoft Office Visio 2019,* | | | | *СУБД MS SQL Server 2022, среда разработки Visual Studio 2022* | | | | |   IV. Перечень подлежащих разработке вопросов   |  |  | | --- | --- | | 1. | *Анализ особенностей предметной области* | | 2. | *Сравнительный анализ программных средств* | | 3. | *Проектирование автоматизированной информационной системы учета клиентов компании (создание Use Case диаграммы, ER диаграммы, прототипа настольного приложения)* | | 4. | *Разработка автоматизированной информационной системы учета клиентов компании (создание базы данных в СУБД MS SQL Server, настольного приложения на языке программирования C# в Visual Studio)* |   V. Перечень графического/иллюстрационного материала   |  |  | | --- | --- | | 1. | *Use Case диаграмма* | | 2. | *ER диаграмма базы данных, разработанная в Microsoft Office Visio* | | 3. | *Диаграмма базы данных, разработанная в СУБД MS SQL Server* | | 4. | *Скриншоты экрана разработанного настольного приложения* | | 5. | *Презентация* |   VI. Дата выдачи задания «04» апреля 2023 г.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Руководитель | |  | | Щур Д.С. | |  | | | (подпись) |  | | Задание принял к исполнению |  | | | Симонов Н.В. | |  | | | (подпись) |  |   «04» апреля 2023 г. |



КАЛЕНДАРНЫЙ ПЛАН ВКР

для специальности 09.02.07 Информационные системы и программирование

(группа 3ИСП11-16)

|  |  |  |
| --- | --- | --- |
| **Утверждение темы и руководителя ВКР** | | 31.03.23 |
| **Выдача задания на ВКР** | | с 31.03.23 по 20.04.23 |
| **Сроки преддипломной практики** | | 3 недели  с 20.04.23 по 10.05.23 |
|  | Выполнение задания по теме ВКР | с 20.04.23 по 05.05.23 |
|  | Предоставление отчета по практике руководителю | 06.05.23 |
|  | Аттестация по практике | 10.05.23 |
| **Подготовка ВКР** | | 5 недель  с 11.05.23 по 14.06.23 |
|  | Подбор и анализ исходной информации | 1 неделя  с 11.05.23 по 17.05.23 |
|  | Подготовка и утверждение плана (оглавления) ВКР |
|  | Работа над разделами (главами) и устранение замечаний руководителя ВКР | 1 неделя  с 18.05.23 по 24.05.23 |
|  | Согласование содержания ВКР, устранение замечаний | 1 неделя  с 25.05.23 по 31.05.23 |
|  | **Оформление и представление руководителю полного текста ВКР на проверку** | 1 неделя  с 01.06.23 по 07.06.23 |
|  | **Сдача демонстрационного экзамена** | Согласно отдельному графику |
|  | **Подготовка к защите, получение допуска к защите ВКР** | 08.06.23-14.06.23 |

**Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.С. Щур**

План принял к исполнению

«04» апреля 2023 г. \_\_\_\_\_\_\_\_\_\_\_\_\_ Аралбаева К.Ш.

**СОДЕРЖАНИЯ**

[ВВЕДЕНИЕ 5](#_Toc135957072)

[ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПРИМЕНЯЕМЫХ ПРОГРАММНЫХ СРЕДСТВ 8](#_Toc135957073)

[1.1. Анализ предметной области и применяемых программных средств 8](#_Toc135957074)

[1.2 Сравнительный анализ используемых программных средств 9](#_Toc135957075)

[1.2.1 Анализ СУБД SQL Server Management Studio 2019 9](#_Toc135957076)

[1.2.2 Анализ объектно-ориентированного языка программирования С# 11](#_Toc135957077)

[1.2.3 Анализ среды разработки Visual Studio 2019 12](#_Toc135957078)

[ГЛАВА 2. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ 16](#_Toc135957079)

[2.1 Создание USE CASE диаграммы 16](#_Toc135957080)

[2.2 Проектирование базы данных. Создание ER-диаграммы 17](#_Toc135957081)

[2.3 Разработка базы данных в СУБД MS SQL SERVER 18](#_Toc135957082)

[2.4 Разработка функционала настольного приложения 20](#_Toc135957083)

[ЗАКЛЮЧЕНИЕ 31](#_Toc135957084)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 32](#_Toc135957085)

[Приложение №1. USE CASE диаграмма](#_Toc135957086)

[Приложение №2. Общая диаграмма базы данных VISIO](#_Toc135957087)

[Приложение №3. Общая диаграмма базы данных VISIO](#_Toc135957088)

[Приложение №4 общая диаграмма базы данных MS SQL SERVER](#_Toc135957089)

[Приложение №5 общая диаграмма базы данных MS SQL SERVER](#_Toc135957090)

# ВВЕДЕНИЕ

Для современных предприятий эффективное управление трудовыми ресурсами является важнейшим фактором успешной деятельности. Одной из задач в этой области является подбор, найм и сопровождение персонала, что также включает в себя анализ и оценку профессиональных навыков кандидатов, управление документами и отчетами, а также эффективный контроль за персоналом. Для решения такой задачи может быть использована информационная система, которая позволит автоматизировать процесс подбора, найма и сопровождения трудовых ресурсов, чтобы минимизировать временные и финансовые затраты, ускорить процесс принятия решений и повысить эффективность работы сотрудников отдела кадров. В данной работе будет рассмотрено проектирование и разработка такой информационной системы на примере конкретной компании, работающей в сфере информационных систем и программирования.

Можно с уверенностью утверждать, что решение широкого круга задач в любой сфере деятельности человека на данный момент практически невозможно без использования автоматизированных информационных систем.

Актуальность данной работы обусловлена необходимостью предприятий в разных сферах экономики иметь эффективную систему управления трудовыми ресурсами, повышая качество и эффективность работы сотрудников и обеспечивая конкурентоспособность на рынке труда. Разработка информационной системы для подбора, найма и сопровождения трудовых ресурсов является актуальной темой в сфере информационных технологий и управления персоналом.

Цель данной работы состоит в разработке и реализации информационной системы для подбора, найма и сопровождения трудовых ресурсов на конкретном предприятии, работающем в сфере информационных технологий и программирования. Другими словами, цель данной работы заключается в автоматизации процессов управления кадрами, минимизации временных затрат на подбор и сопровождение персонала, а также в повышении качества подбора и эффективности работы сотрудников отдела кадров.

Для достижения цели были поставлены следующие задачи:

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

Объектом исследования данной работы является информационная система для подбора, найма и сопровождения трудовых ресурсов на конкретном предприятии, работающем в сфере информационных технологий и программирования.

Предметом исследования является разработка базы данных и удобного интерфейса для автоматизации управления трудовыми ресурсами на конкретном предприятии, а также процессы, сопутствующие этой системе.

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

* подбор персонала;
* найм персонала;
* возможность редактировать, изменять, добавлять данные о персонале;
* сопровождение персонала;
* анализ данных о персонале.

В ходе исследования применялись следующие методы: анализ, моделирование, тестирование, изучение технической литературы и ее обобщение.

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

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

# ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПРИМЕНЯЕМЫХ ПРОГРАММНЫХ СРЕДСТВ

## 1.1. Анализ предметной области и применяемых программных средств

Предметная область управления трудовыми ресурсами (УТР) является одной из ключевых областей управления предприятием. Она включает в себя все аспекты управления персоналом, начиная от набора и найма новых сотрудников и заканчивая их увольнением. В условиях рынка, который постоянно меняется, предприятиям необходимо знать, как эффективно использовать трудовые ресурсы, чтобы остаться конкурентоспособными.

Автоматизированная система должна предоставлять всем заинтересованным лицам (администраторам, менеджерам) мгновенный доступ ко всей необходимой информации.

Разработку базы данных для автоматизации основных процессов и операций деятельности кадровой службы конкретного предприятия необходимо производить с учетом всех особенностей кадрового делопроизводства предприятия:

* масштаба предприятия;
* количества работающих;
* количества кадровых работников и уровня их компьютерной грамотности;
* объема обрабатываемых организационно-документационных массивов;
* оснащенности предприятия аппаратно-программными средствами.

Данная автоматизированная система предназначена для оптимизации работы таких ролей, как “Менеджер”, “Администратор”.

Рассмотрим возможности каждого из них, начнем с менеджера. Основными функциями для него будут:

* Просмотр работников;
* Утраивать и увольнять из должности;
* Отправлять в командировки;
* Оформлять отпуск.

Роль администратора будет связана с технической частью.

Основными функциями для него будут:

* Редактирование данных;
* Добавление и удаление данных;
* Добавление правок.

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

Основными задачами модератора являются:

* проверка документов пользователя;
* утверждение или отклонение заявок пользователей на получение услуг;
* добавление данных о новых автомобилях;
* обновление информации об автомобилях.

Модератор не может удалять автомобили из каталога или автопарка, удалять учётные записи, а также не имеет права редактировать собственную информацию.

Последней категорией пользователей является администратор. Администратор обладает всеми возможными правами для редактирования базы данных, кроме, само собой, редактирования заключённых договоров:

* добавление, изменение, удаление учётных записей пользователей, модераторов, администраторов;
* добавление, изменение, удаление данных об автомобилях.

## 1.2 Сравнительный анализ используемых программных средств

### 1.2.1 Анализ СУБД SQL Server Management Studio 2019

Среда «SQL Server Management Studio» это единая универсальная среда для доступа, настройки и администрирования всех компонентов MS SQL Server, а также для разработки компонентов системы, редактирования текстов запросов, создания скриптов и пр. Благодаря наличию большого количества визуальных средств управления, среда «SQL Server Management Studio*»* позволяет выполнять множество типовых операций по администрированию MS SQL Server администраторам с любым уровнем знаний SQL Server. Удобная среда разработки, встроенный веб–браузер для быстрого обращения к библиотеке MSDN или получения справки в сети, подробный учебник, облегчающий освоение многих новых возможностей, встроенная справка от сообществ в Интернете и многое другое позволяют максимально облегчить процесс разработки в среде SQL Server, а также дает богатые возможности для создания различных сценариев SQL Server.

* среде SQL Server Management Studio встроен обозреватель объектов, который позволяет просматривать все объекты сервера, и предоставляет графический интерфейс для управления этими объектами. Для запуска можно использовать меню «*Вид* *–>* *Обозреватель объектов*».

Основные преимущества СУБД MS SQL Server:

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

Все вышеперечисленное обусловило широкое распространение решений на базе данной СУБД.

### 1.2.2 Анализ объектно-ориентированного языка программирования С#

Язык программирования **C#** – популярный объектно–ориентированный язык программирования, который с выходом **.Net Core** в 2016 году, стал кроссплатформенным. С# представляет собой современный язык для разработки компонентов и модулей для платформ .Net. С# обеспечивает создание безопасного и верифицируемого кода, применение которого возможно в любом современном программном комплексе. Созданные на языке C# приложения могут успешно работать на любом компьютере под управлением операционной системы Windows, Linux и MacOS независимо от архитектуры железа.

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, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников – языков C++, Pascal, 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++ и некоторых других языков, не поддерживает множественное наследование классов (между тем допускается множественное наследование интерфейсов).

Достоинства:

1. высокая совместимость с языком С, позволяющая использовать весь существующий С–код (код С может быть с минимальными переделками скомпилирован компилятором С++; библиотеки, написанные на С, обычно могут быть вызваны из С++ непосредственно без каких–либо дополнительных затрат, в том числе и на уровне функций обратного вызова, позволяя библиотекам, написанным на С, вызывать код, написанный на С++);
2. поддерживаются различные стили и технологии программирования, включая традиционное директивное программирование, ООП, обобщенное программирование, метапрограммирование (шаблоны, макросы);
3. имеется возможность работы на низком уровне с памятью, адресами, портами;
4. возможность создания обобщённых контейнеров и алгоритмов для разных типов данных, их специализация и вычисления на этапе компиляции, используя шаблоны;

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

1. эффективность. Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы.

### 1.2.3 Анализ среды разработки Visual Studio 2019

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

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

Встроенный отладчик Visual Studio используется для поиска и исправления ошибок в исходном коде, в том числе на низком аппаратном уровне. Инструменты диагностики позволяют оценить качество кода с точки зрения производительности и использования памяти. Дизайнер форм Visual Studio незаменим при разработке программ с графическим интерфейсом, помогая спроектировать внешний вид будущего приложения и работу каждого элемента интерфейса. Наконец, Visual Studio предоставляет комплекс инструментов для автоматизации тестирования приложений в части проверки работы интерфейсов, модульного и нагрузочного тестирования. Для командных проектов Visual Studio предлагает поддержку групповой работы, позволяя выполнять совместное редактирование и отладку любой части кода в реальном времени, а в качестве системы управления версиями использовать Team Foundation или Git.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно–ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

Достоинства Visual Studio:

1. встроенный Web-сервер – для обслуживания Web-приложения ASP.NET необходим Web-сервер, который будет ожидать Web-запросы и обрабатывать соответствующие страницы. Наличие в Visual Studio интегрированного Web-сервера позволяет запускать Web-сайт прямо из среды проектирования, а также повышает безопасность, исключая вероятность получения доступа к тестовому Web-сайту с какого-нибудь внешнего компьютера, поскольку тестовый сервер может принимать соединения только с локального компьютера;
2. поддержка множества языков при разработке – Visual Studio позволяет писать код на своем языке или любых других предпочитаемых языках,

используя все время один и тот же интерфейс (IDE). Более того, Visual Studio также еще позволяет создавать Web-страницы на разных языках, но помещать их все в одно и то же Web-приложение. Единственным ограничением является то, что в каждой Web-странице можно использовать только какой-то один язык (очевидно, что в противном случае проблем при компиляции было бы просто не избежать).

1. меньше кода для написания – для создания большинства приложений требуется приличное количество стандартного стереотипного кода, и Web-страницы ASP. NET тому не исключение. Например, добавление Web-элемента управления, присоединение обработчиков событий и корректировка форматирования требует установки в разметке страницы ряда деталей. В Visual Studio такие детали устанавливаются автоматически;
2. интуитивный стиль кодирования – по умолчанию Visual Studio форматирует код по мере его ввода, автоматически вставляя необходимые отступы и применяя цветовое кодирование для выделения элементов типа комментариев. Такие незначительные отличия делают код более удобным для чтения и менее подверженным ошибкам. Применяемые Visual Studio автоматически параметры форматирования можно даже настраивать, что очень удобно в случаях, когда разработчик предпочитает другой стиль размещения скобок (например, стиль K&R, при котором открывающая скобка размещается на той же строке, что и объявление, которому она предшествует);
3. более высокая скорость разработки – многие из функциональных возможностей Visual Studio направлены на то, чтобы помогать разработчику делать свою работу как можно быстрее. Удобные функции, вроде функции IntelliSense (которая умеет перехватывать ошибки и предлагать правильные варианты), функции поиска и замены (которая позволяет отыскивать ключевые слова как в одном файле, так и во всем проекте) и функции автоматического добавления и удаления комментариев (которая может временно скрывать блоки кода), позволяют разработчику работать быстро и эффективно.
4. возможности отладки – педлагаемые в Visual Studio инструменты отладки являются наилучшим средством для отслеживания загадочных ошибок и диагностирования странного поведения. Разработчик может выполнять свой код по строке за раз, устанавливать интеллектуальные точки прерывания, при желании сохраняя их для использования в будущем, и в любое время просматривать текущую информацию из памяти.

Выводы

В результате изучения теоретического материала по свойствам перечисленных средств разработки программного обеспечения можно прийти к следующим выводам:

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

# ГЛАВА 2. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 2.1 Создание USE CASE диаграммы

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

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

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

Субъект «Менеджер» должен иметь возможность просматривать работников в компании, найма, отправление в отпуск, командировку, и увольнение.

Субъект «Администратор» должен владеть такие же возможности, как и субъект «Менеджер», но также редактирование данных, удаление и добавление правок.

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

## 2.2 Проектирование базы данных. Создание ER-диаграммы

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

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

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

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

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

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

Логическая модель по-прежнему не ориентирована, на какую-либо конкретную СУБД для дальнейшей физической реализации базы данных. Конечным результатом логического проектирования является описание логической структуры базы данных на языке описания данных. Процесс проектирования логической структуры базы данных подразумевает определение всех информационных единиц и связей между ними, задание их имен, определения их типов.

Прежде чем строить физическую модель, необходимо решить, какая информация будет храниться в базе данных. Чтобы система смогла выполнять свои функции база данных должна содержать следующие данные:

1) Данные о работниках;

2) На каких должностях они находятся;

3) Сведения об отпусках сотрудников;

4) Сведения об командировок сотрудников;

Полноразмерная схема базы данных, разработанной в Microsoft Office Visio 2019, представлена в Приложении 2-3.

## 2.3 Разработка базы данных в СУБД MS SQL Server

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

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

Рассмотрим основные сущности нашей базы данных.

Сущность “Worker”. В данной таблице находятся основные данные об работнике, включая стаж работы, образование, инициалы и т.д. Так же есть связующая сущность ”Gender”, которая определяет гендер работника.(Рисунок 2.1)

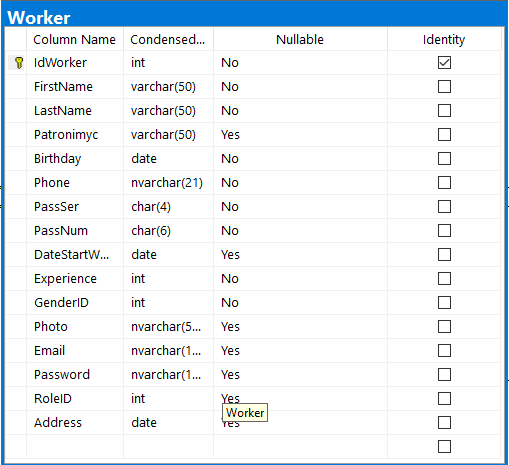


Рисунок 2.1 Таблица «Worker»

Таблицы “Post”. Тут можно узнать какая есть должность, оплата. Есть связующая сущность “Department”, где находятся отделы, в которых есть эти должности. (Рисунок 2.2)

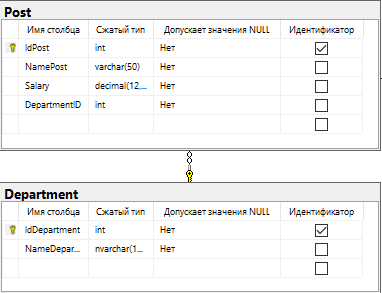


Рисунок 2.2 Таблицы «Post»

Таблица “WorkerPost”. С помощью этой сущности можно устраивать, так и увольнять работников с должности. (Рисунок 2.3)

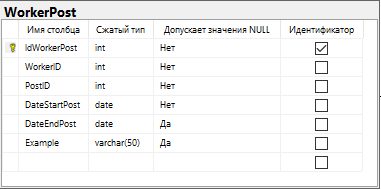


Рисунок 2.3 Таблица «WorkerPost»

Полноразмерная схема базы данных, разработанной в СУБД MS SQL Server 2019, представлена в Приложении 4-5.

## 2.4 Разработка функционала настольного приложения

Пользовательский интерфейс для базы данных будет выполнен с использованием платформы Windows Presentation Foundation (WPF). WPF - система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык декларативного описания интерфейса XAML.

Первое, что увидит пользователь при запуске приложения – окно авторизации. (Рисунок 2.4)

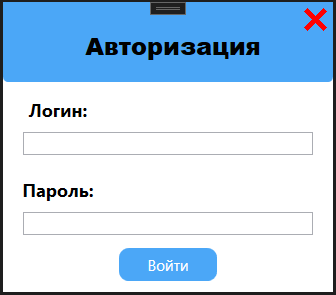


Рисунок 2.4 Визуализация окна «Авторизация»

Сотруднику выдается логин и пароль, который выдается для возможности авторизации в приложении. Авторизация производится посредством поиска в таблице “Worker” совпадающего логина и соответствующего ему пароля.

В случае отсутствия совпадения пользователю выводится информационный диалог (Рисунок 2.5).

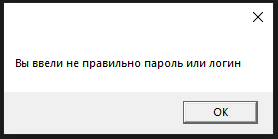


Рисунок 2.5 Пример неудачной попытки авторизации

В случае удачной авторизации, в зависимости от уровня привилегий пользователя открывается форма для сотрудников.

В случае авторизации в качестве “Менеджера”, пользователь видит страницу приложения, на которой предлагается просмотр работников, отправление работника в командировку или в отпуск (Рисунок 2.6).

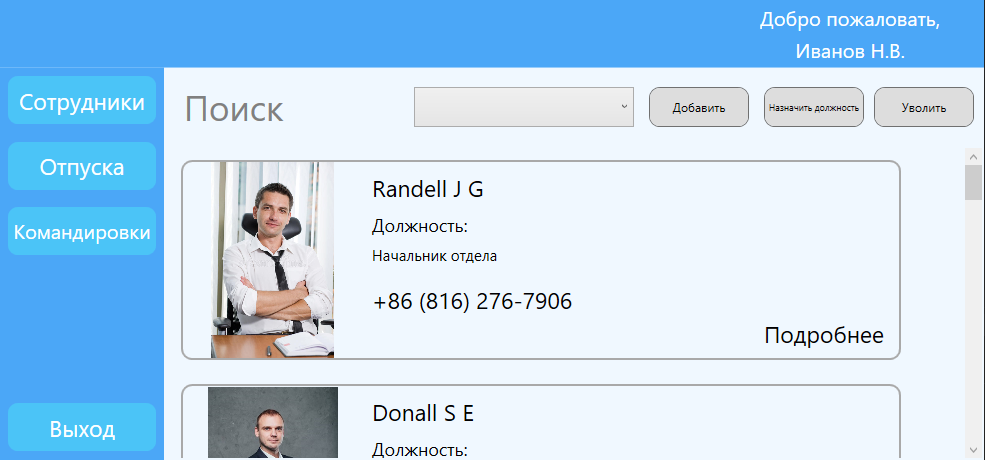


Рисунок 2.6 Пример главной страницы

Если хочется узнать информацию о работнике по подробнее, то можно нажать на кнопку “Подробнее” (Рисунок 2.7).



Рисунок 2.7 Окно с выводом информации о сотруднике.

Так же можно найти работника через поиск, отфильтровать список по фамилии, должности и по телефону.

Еще менеджер, может добавлять работника в компанию. Делается это через кнопку “Добавить”. При нажатии этой кнопки всплывает окно в котором появляются поля которые нужно заполнить, чтобы работник появился в базе (Рисунок 2.8).

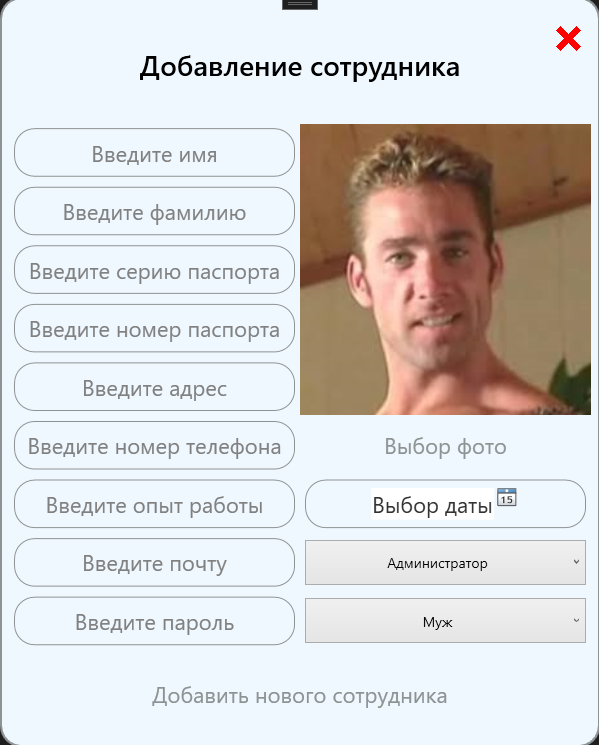
****

Рисунок (2.8) Окно с добавлением нового сотрудника

После того как мы добавили сотрудника в базу данных, ему нужно назначить должность, а также дата начало работы (Рисунок 2.9).

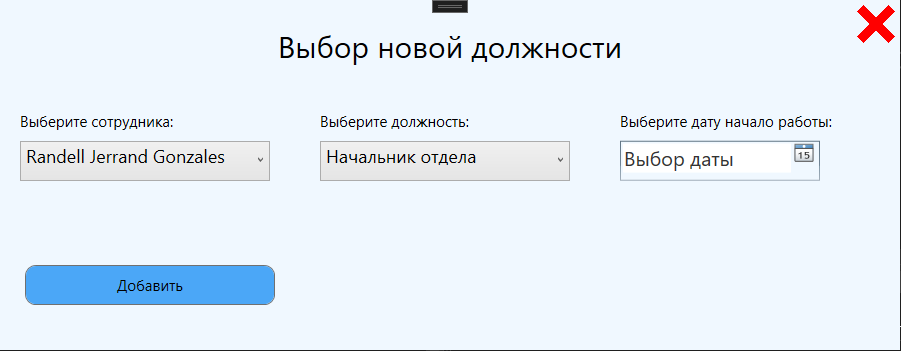


Рисунок 2.9 Окно на добавление на новую должность

Также сотрудника можно уволить из должности, на которой он устроен. (Рисунок 2.10)

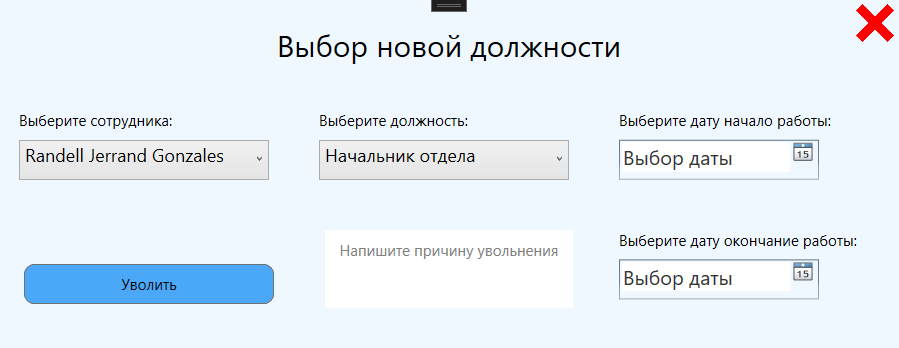


Рисунок 2.10 Окно увольнения

Менеджер может просматривать информацию об работниках, который были в отпуске. (Рисунок 2.11)

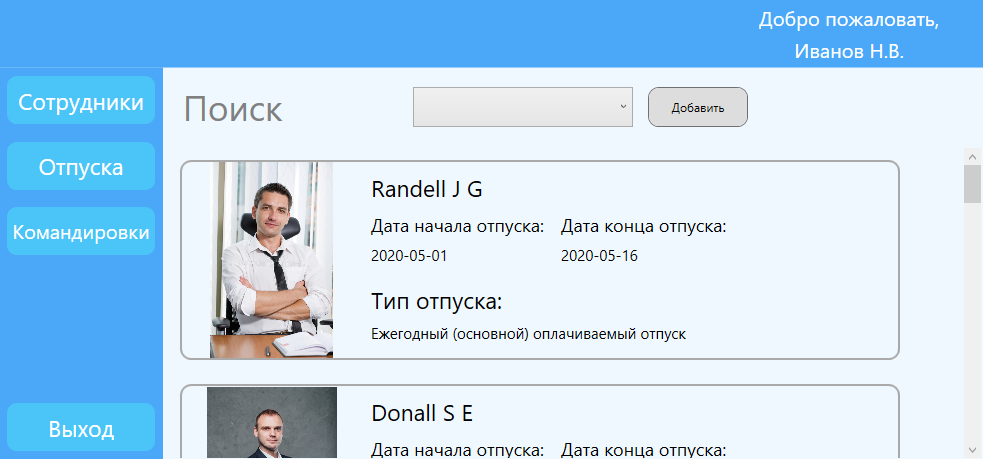


Рисунок 2.11 Окно с выводом информации об отпусках

В этом окне аналогично другому окну с сотрудниками, можно использовать поиск для удобства, фильтр, а также нажимая кнопку “Добавить” всплывает новое окно, в котором можно отправить сотрудника в отпуск. Чтобы отправить сотрудника в отпуск, нужно выбрать его из списка, указать дату начало и конца отпуска, а также тип отпуска. (Рисунок 2.12).

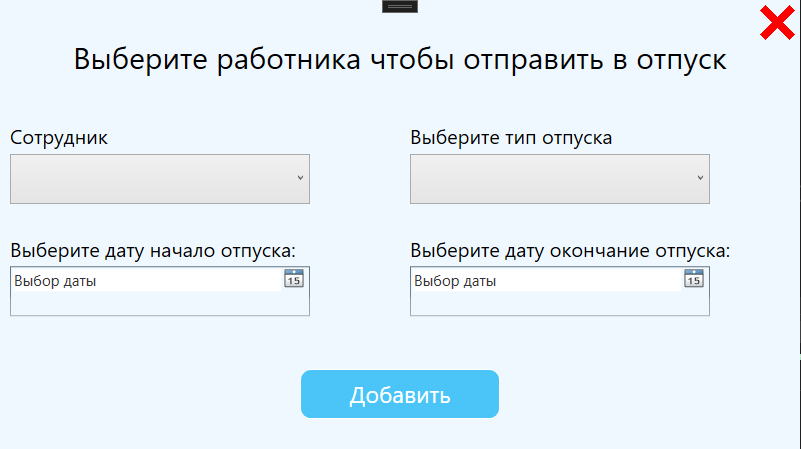


Рисунок 2.12 Окно для отправления сотрудника в отпуск

Менеджер может просматривать информацию об работниках, которые были в командировках (Рисунок 2.13).

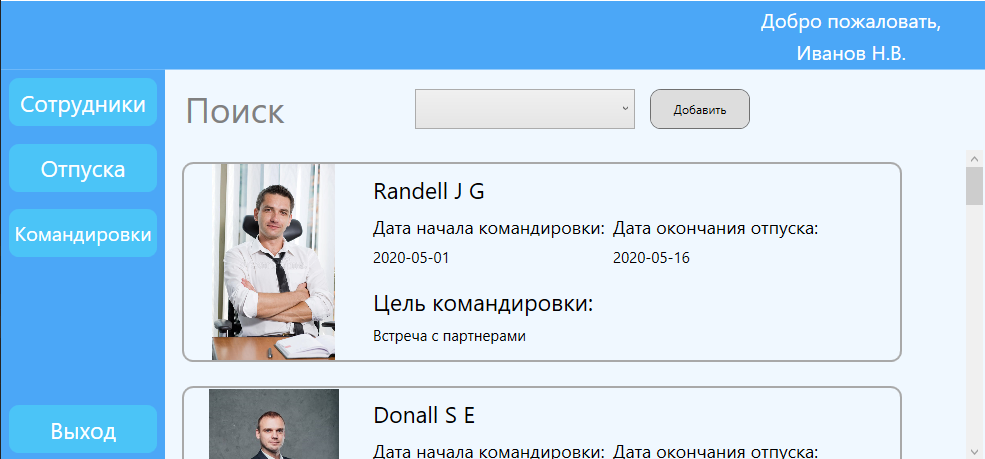


Рисунок 2.13 Окно с выводом информации об командировках

В этом окне так же аналогично с прошлыми окнами есть поиск с фильтром и кнопкой “Добавить”, который открывает новое окно, в котором можно отправить сотрудника в командировку. Чтобы отправить сотрудника в командировку нужно выбрать его из списка, далее указать место командировки, номер приказа, дату начало и конца командировки и вести цель поездки (Рисунок 2.14).

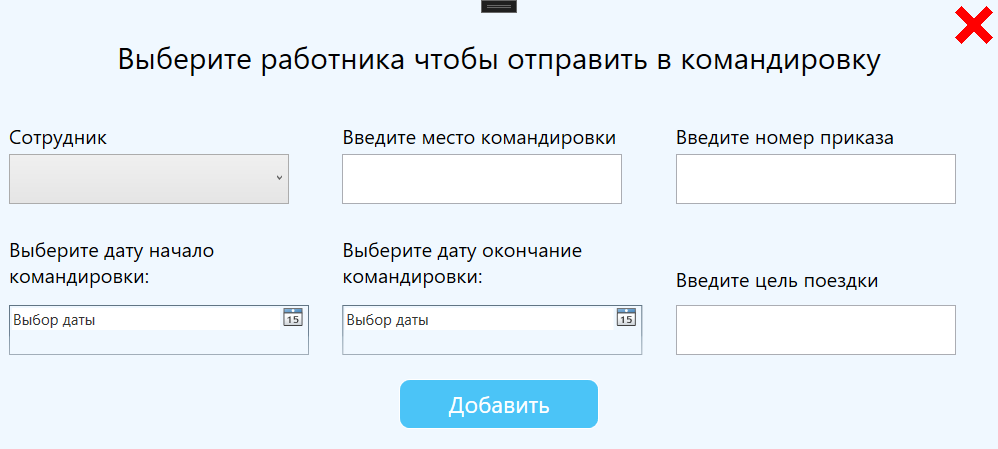


Рисунок 2.14 Окно для отправления сотрудника в командировку

Когда авторизуется юзер под ролью “Администратор”, то у него остаются те же возможности как у роли “Менеджер”, ток появляется возможность изменять, удалять данные (Рисунок 2.15).

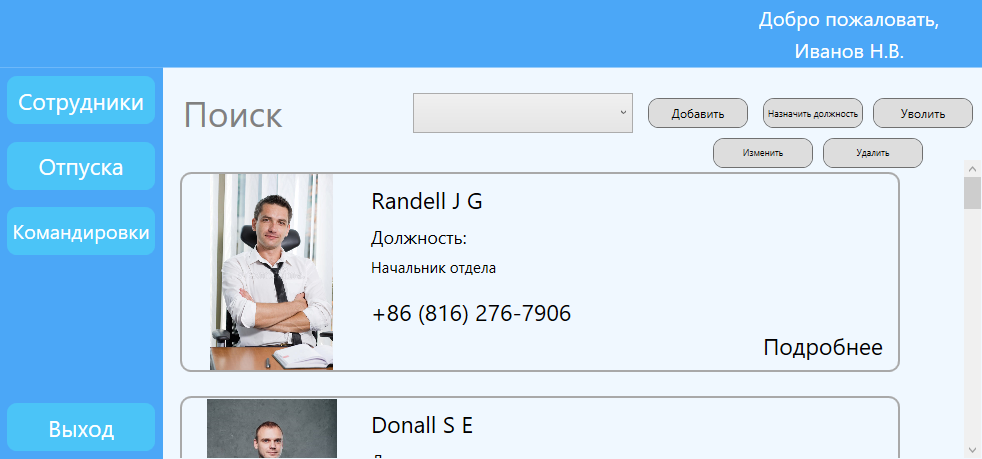


Рисунок 2.15 Окно Администратора

Нажимая кнопку “Изменить” открывается окно с редактированием данных сотрудника (Рисунок 2.16).

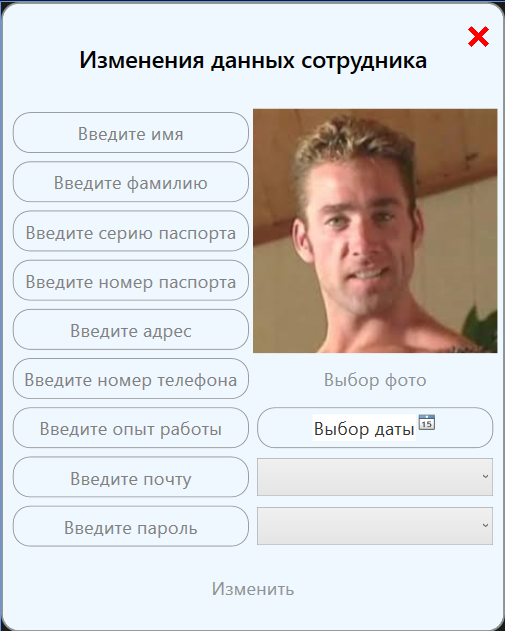


Рисунок 2.16 Окно с редактированием данных

Администратор может просматривать информацию об работниках, который были в отпуске. Также у него появились в окне кнопки “Изменить”, “Удалить”. Нажимая на кнопку “Удалить”, удаляется информация об отпуске сотрудника (Рисунок 2.17)

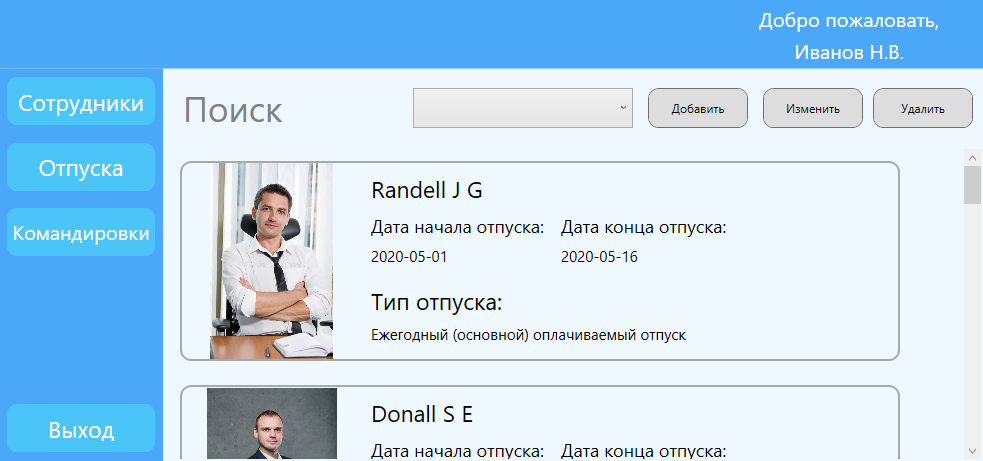


Рисунок 2.17 Окно с информацией об отпусках сотрудников

Нажимая на кнопку “Изменить” открывается окно которая редактирует данные, в данном случае мы редактируем данные об отпуске сотрудника (Рисунок 2.18)

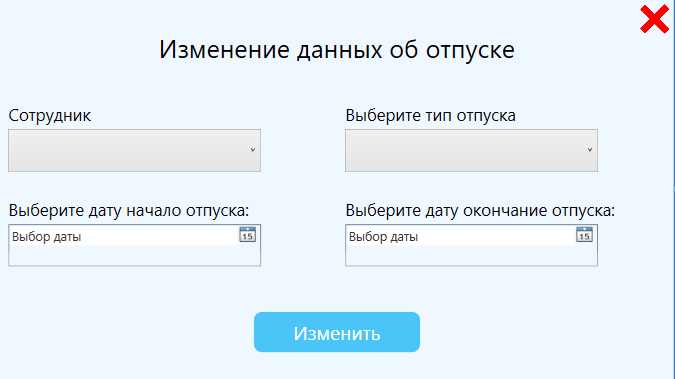


Рисунок 2.18 Окно изменение данных об отпуске

Администратор может просматривать информацию об работниках, которые были в командировках. Также у него появились в окне кнопки “Изменить”, “Удалить”. Нажимая на кнопку “Удалить”, удаляется информация о командировках сотрудников (Рисунок 2.19)

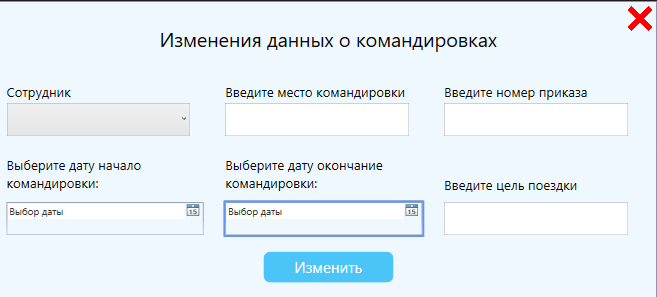


Рисунок 2.19 Окно изменение данных о командировке

# ЗАКЛЮЧЕНИЕ

Таким образом, во второй главе были разработаны ER и USE CASE диаграммы, с помощью которых была разработана база данных, а также программа на языке C#. С понятным интерфейсом пользователь сможет быстро и эффективно получить, отредактировать или добавить нужную информацию.

В ВКР были решены следующие задачи:

1. проведение анализа предметной области.
2. Создание USE CASE диаграммы
3. Создание ER-диаграммы
4. Разработка базы данных в СУБД
5. Разработка приложения

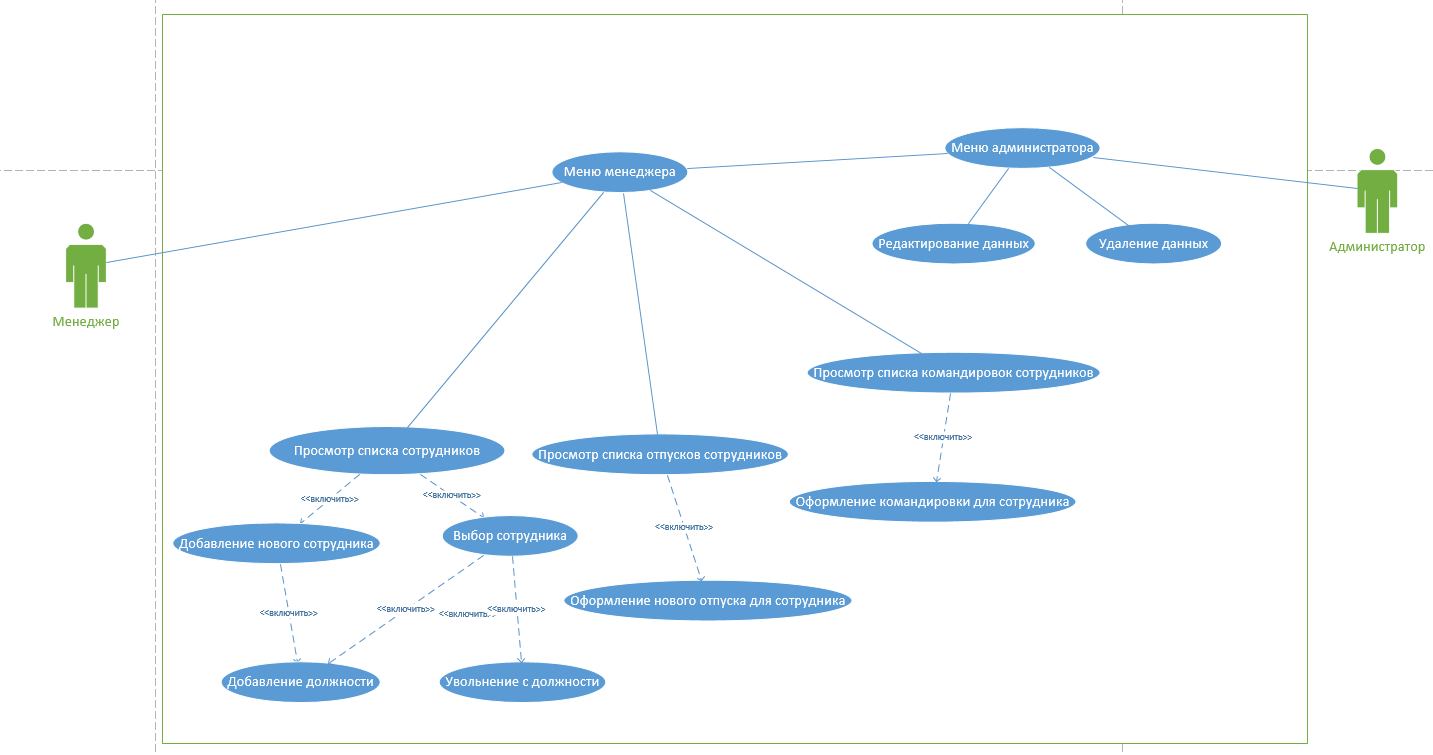
Цель работы была достигнута – разработка и оптимизирование работы автоматизированной системы.

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

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Тюкачев, Н. А. C#. Основы программирования : учебное пособие для вузов / Н. А. Тюкачев, В. Г. Хлебостроев. — 4-е изд., стер. — Санкт-Петербург : Лань, 2021. — 272 с.
2. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка C# / Л. А. Залогова. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 192 с.
3. Расторгуев, И. С. Справочник С#. Кратко, быстро, под рукой : справочник / И. С. Расторгуев, А. П. Никольский. — Санкт-Петербург : Наука и Техника, 2023. — 336 с.
4. Волк В. К / «Базы данных. Проектирование, программирование, управление и администрирование.» [Текст] / Москва: Издательство "Лань", 2022г. – 244 с.
5. Ревунков Г. И / «Базы данных: Учебно-методическое пособие.» [Текст] / Ковалева Н. А., Силантьева Е. Ю., Виноградова М. В., Маслеников К. Ю. Москва: Московский государственный технический университет имени Н.Э. Баумана, 2020г. – 28 с.
6. Смирнов М. В. / «Администрирование многопользовательских баз данных.» [Текст] / Москва: МИРЭА - Российский технологический университет, 2021г. – 75 с.
7. Курбанисмаилов, З. М. Основы языка программирования C# : учебно-методическое пособие / З. М. Курбанисмаилов, Е. В. Кашкин. — Москва : РТУ МИРЭА, 2019. — 93 с.
8. Кузниченко, М. А. Основы баз данных : учебно-методическое пособие / М. А. Кузниченко. — 2-е изд., стер. — Москва : ФЛИНТА, 2022. — 102 с.

# Приложение 1. USE CASE диаграмма



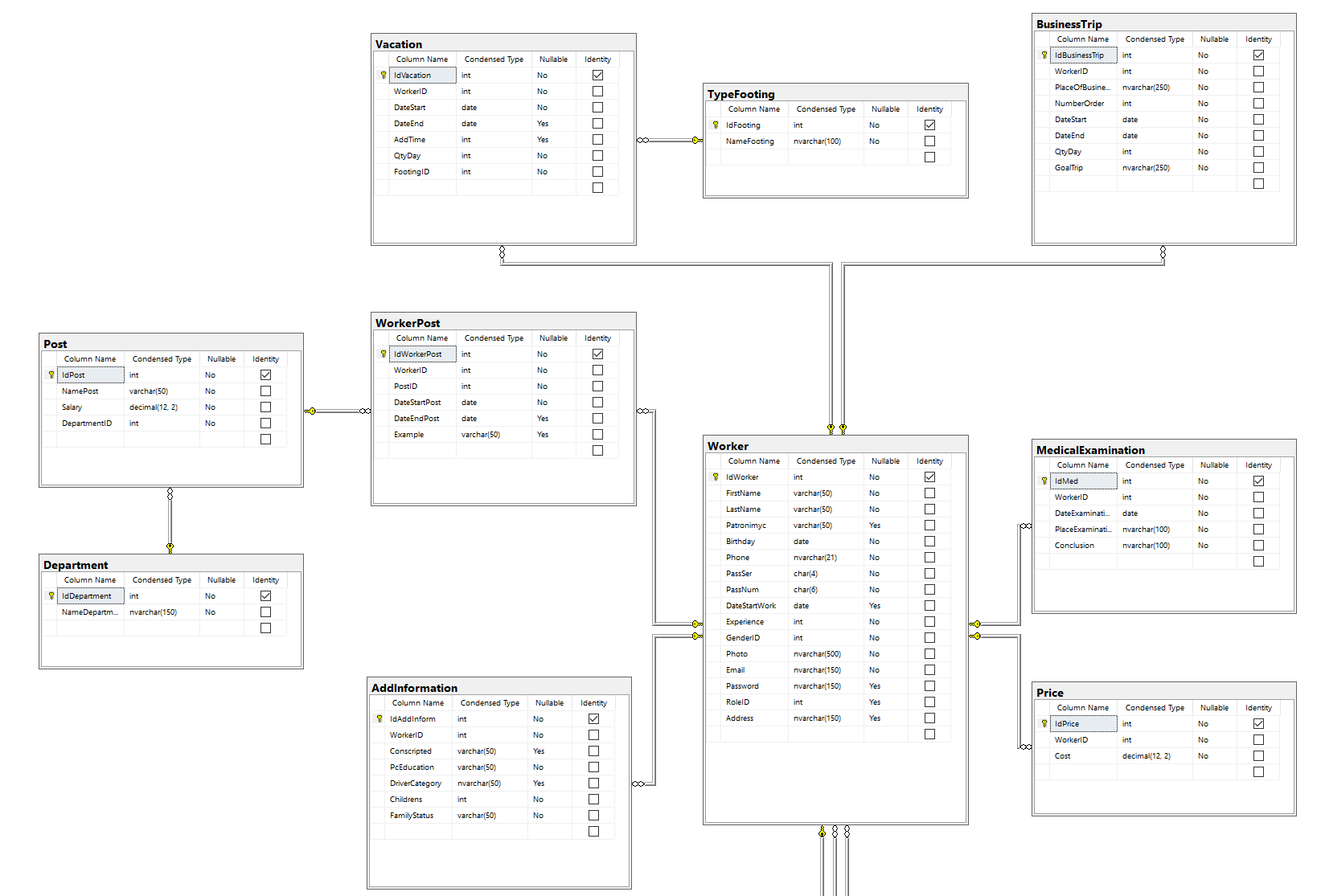
# Приложение 2. Общая диаграмма базы данных VISIO



# Приложение 3. Общая диаграмма базы данных VISIO



# Приложение №4 Общая диаграмма базы данных MS SQL Server



# Приложение №5 Общая диаграмма базы данных MS SQL Server

