**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КОМИ**

**Государственное профессиональное образовательное учреждение**

**«Воркутинский арктический горно-политехнический колледж»**

**КУРСОВОЙ ПРОЕКТ**

По дисциплине МДК.05.02 Разработка кода информационных систем

**Разработка информационной системы регистрации командировочных удостоверений**

Выполнил студент гр. ИСП-21 /\_\_\_\_\_\_\_\_\_/Киселев Дмитрий Михайлович/

(подпись) (Ф.И.О.)

**ОЦЕНКА:\_\_\_\_\_\_\_\_\_\_\_\_\_**

Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПРОВЕРИЛ**

Научный руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_/ Егоров Данил Павлович/

(подпись) (Ф.И.О.)

Воркута

2024

**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ**

**ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ**

**1.1 Платформа .NET**

**1.2 Язык программирования С#**

**1.3 Windows Presentation Foundation (WPF)**

**1.4 СУБД SQL server**

**1.5 Microsoft SQL Server Management Studio**

**1.6 Entity Framework**

**ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

**2.1 Разработка диаграммы ERD**

**2.2 Разработка базы данных**

**ГЛАВА 3. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ**

**3.1 Разработка прототипа информационной системы**

**3.2 Программирование информационной системы**

**3.2.1 Разработка модуля «Авторизация»**

**3.2.2 Разработка модуля …..**

**3.2.3 Разработка модуля «Отчет»**

**ЗАКЛЮЧЕНИЕ**

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

**ВВЕДЕНИЕ**

Современная деловая среда требует эффективных инструментов управления и контроля за командировочными процессами. Одним из ключевых аспектов такого управления является систематизация и автоматизация процесса регистрации командировочных удостоверений. Это особенно актуально для предприятий и организаций, где регулярно отправляются сотрудники в командировки с целью выполнения рабочих обязанностей.

Настоящий курсовой проект посвящен разработке информационной системы, предназначенной для упрощения и оптимизации процесса регистрации командировочных удостоверений. Создание такой системы имеет ряд преимуществ, среди которых уменьшение бюрократических издержке, повышение прозрачности и эффективности управления командировочными процессами, а также возможность оперативного доступа к информации о командировках для всех заинтересованных сторон.

**ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ**

* 1. **Платформа .NET**

Выбор платформы .NET для работы над проектом разработки информационной системы обосновывается несколькими важными причинами:

1. Широкий инструментарий разработки: Платформа .NET предоставляет обширный и мощный инструментарий для разработки разнообразных типов приложений. Включая инструменты для создания веб-приложений, настольных приложений, мобильных приложений и других типов программного обеспечения.
2. Язык программирования C#: Одним из основных языков программирования платформы .NET является C#. Этот язык отличается простотой и выразительностью, что упрощает процесс разработки и позволяет создавать надежные и масштабируемые приложения. Кроме того, C# обладает богатой экосистемой инструментов и библиотек, что делает его идеальным выбором для разработки информационных систем.
3. Интеграция с другими технологиями: Платформа .NET обеспечивает хорошую интеграцию с другими технологиями и сервисами, что позволяет легко встраивать различные компоненты и расширения в разрабатываемую систему. Например, использование Windows Presentation Foundation (WPF) обеспечивает возможность создания настольных приложений с богатым пользовательским интерфейсом.
   1. **Язык программирования C#**

Выбор языка программирования C# для работы над проектом разработки информационной системы регистрации командировочных удостоверений обоснован рядом важных причин:

1. Интеграции с платформой .NET: C# является основным языком программирования для платформы .NET, которая предоставляет обширный инструментарий для разработки приложений различного типа. Это обеспечивает высокую совместимость и эффективную интеграцию с другими технологиями и сервисами, доступными на платформе .NET.
2. Простота и выразительность: C# представляет собой современный и выразительный язык программирования, который обладает простым и понятным синтаксисом. Это делает процесс разработки более эффективным и позволяет разработчикам быстрее создавать и поддерживать код.
3. Обширная экосистема: C# обладает богатой экосистемой инструментов, библиотек и фреймворков, которые значительно упрощают процесс разработки приложений. Например, с помощью фреймворка Entity Framework можно легко работать с базами данных.

**1.3 Windows Presentation Foundation (WPF)**

Выбор Windows Presentation Foundation (WPF) для работы над проектом разработки информационной системы регистрации командировочных удостоверений обоснован следующими причинами:

1. Богатые возможности для создания графического интерфейса: WPF предоставляет мощные инструменты для создания графического пользовательского интерфейса (GUI) с использованием различных элементов управления, стилей, анимации и т.д. Это позволяет разработать привлекательный и интуитивно понятный интерфейс для пользователей системы регистрации командировочных удостоверений.
2. Разделение логики и дизайна: WPF позволяет разделять логику приложения и его дизайн с помощью языка разметки XAML(Extensible Application Markup Language). Это упрощает процесс разработки, позволяет повторно использовать компоненты интерфейса и обеспечивает легкость сопровождения кода.
3. Поддержка масштабируемости и адаптивности: WPF обладает возможностями для создания адаптивного и масштабируемого интерфейса, что важно для системы, которая может использоваться на различных устройствах и разрешениях экранов. Это позволяет обеспечить удобство использования системы независимо от характеристик устройства пользователя.

**1.4 СУБД SQL Server**

Выбор СУБД SQL Server для работы над проектом разработки информационной системы регистрации командировочных удостоверений обоснован следующими причинами:

1. Надежность и стабильность: SQL Server является продуктом корпорации Microsoft, что обеспечивает высокую степень надежности и стабильности его работы. Это важно для информационной системы, которая будет хранить и обрабатывать критически важные данные о командировках сотрудников.
2. Высокая производительность: SQL Server обладает высокой производительностью работы с данными, что позволяет обрабатывать большие объемы информации с высокой скоростью. Это важно для системы, которая будет использоваться в организации с большим числом командировок и сотрудников.
3. Широкая поддержка интеграции с платформой .NET: SQL Server тесно интегрируется с платформой .NET и другими технологиями Microsoft, что обеспечивает легкость взаимодействия между базой данных и разрабатываемым приложением. Например, с использованием технологии Entity Framework можно упростить доступ к данным из приложения на языке C#.

**1.5 Microsoft SQL Server Management Studio**

Выбор Microsoft SQL Server Management Studio (SSMS) для работы в данном проекте обоснован следующими важными причинами:

1. Интуитивный интерфейс: SSMS предоставляет удобный и интуитивно понятный интерфейс для администрирования и управления базами данных SQL Server. Это делает процесс разработки, отладки и оптимизации запросов более эффективным и удобным для разработчиков и администраторов баз данных.
2. Широкие возможности управления базами данных: SSMS предоставляет широкий набор инструментов для управления базами данных SQL Server, включая создание и изменение таблиц, индексов, процедур, представлений, а также резервное копирование, восстановление и мониторинг баз данных. Это обеспечивает полный контроль над структурой и содержимым баз данных.
3. Интеграция с другими инструментами Microsoft: SSMS тесно интегрируется с другими инструментами и технологиями от Microsoft, такими как Visual Studio и платформа .NET. Это позволяет легко взаимодействовать с базой данных из различных сред разработки и обеспечивает совместимость с другими компонентами разрабатываемого приложения.

**1.6 Entity Framework**

Выбор Entity Framework (EF) для работы в данном проекте обоснован следующими ключевыми причинами:

1. Упрощение доступа к данным: Entity Framework предоставляет удобный и интуитивно понятный способ доступа к данным в базе данных с использованием объектно-ориентированной модели. Это позволяет разработчикам работать с данными как с объектами в коде на языке C#, что делает процесс доступа к данным более естественным и интуитивным.
2. Сокращение времени разработки: Использование EF позволяет существенно сократить время разработки приложения за счет автоматической генерации SQL-запросов и маппинга результатов запросов на объекты. Это уменьшает необходимость вручную писать SQL-запросы и упрощает процесс работы с данными.
3. Увеличение поддерживаемости кода: EF позволяет использовать линейную модель данных, что делает код более читаемым и поддерживаемым. Изменения в структуре базы данных автоматически отражаются в модели EF, что упрощает поддержку и обновление приложения при изменении требований.
4. Богатые возможности для работы с данными: Entity Framework предоставляет множество инструментов и функциональных возможностей для работы с данными, включая управление транзакциями, кеширование данных, поддержку отношений между таблицами и многое другое. Это делает EF мощным инструментом для работы с данными в различных сценариях.
5. Интеграция с другими технологиями .NET: Entity Framework тесно интегрируется с другими технологиями и инструментами на платформе .NET, такими как ASP.NET и WPF. Это обеспечивает легкость взаимодействия между слоями приложения и упрощает разработку и поддержку приложения в целом.

**ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

**2.1 Разработка диаграммы ERD**



На диаграмме представлены следующие сущности:

1. Сотрудник(Employee): эта сущность представляет информацию о сотрудниках компании. У каждого сотрудника есть уникальный идентификатор, имя, фамилия, должность и отдел.
2. Командировочное удостоверение(BusinessTrips): эта сущность содержит информацию о командировочных поездках сотрудников. Каждое командировочное удостоверение имеет уникальный идентификатор, место назначения, даты начала и окончания командировки, а также цель поездки.
3. Пользователь(User): эта сущность представляет пользователей информационной системы. Каждый пользователь имеет уникальный идентификатор, имя пользователя, пароль и роль.

Связи, представленные на диаграмме:

Сущность Сотрудник связана с сущностью Командировочные удостоверения отношением один-ко-многим. Это означает, что каждый сотрудник может иметь несколько командировочных удостоверений.

**2.2 Разработка базы данных**

**Описание структуры базы данных**

База данных предназначена для хранения информации о командировочных удостоверениях сотрудников компании. Она состоит из нескольких таблиц, каждая из которых содержит определенные атрибуты для хранения соответствующей информации.

Таблица "Employees": Хранит информацию о сотрудниках компании.

EmployeeID: Уникальный идентификатор сотрудника (первичный ключ).

FullName: Полное имя сотрудника.

Department: Отдел, к которому принадлежит сотрудник.

Position: Должность сотрудника.

Таблица "BusinessTrips": Содержит информацию о командировочных удостоверениях.

TripID: Уникальный идентификатор командировочного удостоверения (первичный ключ).

EmployeeID: Идентификатор сотрудника, связанный с таблицей "Employees" (внешний ключ).

Destination: Место назначения командировки.

StartDate: Дата начала командировки.

EndDate: Дата окончания командировки.

Purpose: Цель командировки.

**Связи между таблицами**

В таблице "BusinessTrips" поле EmployeeID связано с полем EmployeeID в таблице "Employees". Эта связь обеспечивает отслеживание, какой сотрудник совершает каждую командировку.

**Ограничения целостности данных**

В таблице "Employees" поле EmployeeID является первичным ключом, гарантирующим уникальность идентификатора каждого сотрудника.

В таблице "BusinessTrips" поле TripID является первичным ключом, гарантирующим уникальность идентификатора каждого командировочного удостоверения.

Связь между таблицами "Employees" и "BusinessTrips" обеспечивает согласованность данных, предотвращая добавление командировочных удостоверений для несуществующих сотрудников.

**Форматы данных и типы**

Для идентификаторов сотрудников и командировочных удостоверений используются целочисленные значения (INT).

Для полного имени сотрудника, отдела и должности используется строковый формат (VARCHAR).

Для дат начала и окончания командировки используется тип данных "Дата" (DATE).

**Резервное копирование и восстановление данных**

Предусмотрено регулярное резервное копирование базы данных для обеспечения защиты данных от потери или повреждения.

В случае сбоя или потери данных предусмотрен процесс восстановления из резервной копии для минимизации времени простоя и потерь информации.

**ГЛАВА 3. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ**

* 1. **Разработка прототипа информационной системы**

**Цель разработки прототипа**

Целью разработки прототипа информационной системы является демонстрация основного функционала предполагаемой системы, визуализация интерфейса пользователя и проверка концепции работы системы на ранних этапах разработки.

**Основные функциональные требования**

Прототип информационной системы должен предоставлять следующие основные функции:

Регистрация новых командировочных удостоверений сотрудников:

Пользователь (HR) должен иметь возможность ввода данных о командировочном удостоверении: место назначения, даты начала и окончания командировки, цель.

Просмотр списка зарегистрированных командировочных удостоверений:

Пользователь (HR) должен иметь доступ к списку всех зарегистрированных командировочных удостоверений сотрудников.

Поиск командировочных удостоверений по различным критериям:

Пользователь (HR) должен иметь возможность выполнять поиск командировочных удостоверений по различным критериям, таким как имя сотрудника, дата командировки, цель и т. д.

Редактирование и удаление существующих командировочных удостоверений:

Пользователь (HR) должен иметь возможность редактировать и удалять существующие командировочные удостоверения.

**Интерфейс пользователя**

Прототип должен иметь простой и интуитивно понятный интерфейс пользователя, состоящий из следующих элементов:

Форма для ввода данных о новом командировочном удостоверении.

Список с зарегистрированными командировочными удостоверениями с возможностью просмотра, редактирования и удаления.

Панель поиска для фильтрации командировочных удостоверений по различным критериям.

**Технологии и инструменты**

Для разработки прототипа информационной системы будут использоваться следующие технологии и инструменты:

Язык программирования: C#.

Фреймворк для разработки пользовательского интерфейса: WPF (Windows Presentation Foundation).

Среда разработки: Microsoft Visual Studio.

База данных: MySQL.

ORM-фреймворк: Entity Framework Core.

**Ограничения прототипа**

Прототип будет ограничен функциональностью, представленной в разделе «Основные функциональные требования», и будет содержать минимальный набор функций, необходимых для демонстрации работы системы на раннем этапе разработки. В прототипе могут отсутствовать дополнительные функции и элементы интерфейса, предусмотренные для конечной версии информационной системы.

* 1. **Программирование информационной системы**
     1. **Разработка модуля «Авторизация»**

Модуль "Авторизация" должен обеспечивать следующие основные функции:

1. Аутентификация пользователей:

Пользователи должны иметь возможность ввода своих учетных данных (логин и пароль) для входа в систему.

1. Проверка учетных данных:

Модуль должен проверять введенные пользователем учетные данные и аутентифицировать пользователя, если они верны.

1. Авторизация доступа:

После успешной аутентификации модуль должен проверять права доступа пользователя к различным функциям и данным системы и предоставлять соответствующие разрешения.

1. Обработка ошибок аутентификации и авторизации:

Модуль должен обрабатывать случаи неверного ввода учетных данных, отсутствия учетной записи пользователя, а также ограничения доступа из-за недостаточных прав.

Интерфейс для модуля "Авторизация" должен включать в себя следующие элементы:

* Форма для ввода логина и пароля пользователя.
* Кнопка для запуска процесса аутентификации.
* Сообщения об ошибках в случае неверного ввода учетных данных или ограничения доступа.
  + 1. **Разработка модуля «Отчет»**

Модуль "Отчет" должен обеспечивать следующие основные функции:

1. Выбор типа отчета:

Пользователи должны иметь возможность выбирать тип отчета из предварительно определенного списка.

1. Формирование отчета:

Модуль должен выполнять запросы к базе данных для извлечения необходимых данных и формирования отчета в выбранном формате (например, PDF, Excel, CSV).

1. Настройка параметров отчета:

Пользователи должны иметь возможность настраивать параметры отчета, такие как период времени, диапазон дат, фильтры по данным и т. д.

1. Предварительный просмотр отчета:

Модуль должен предоставлять пользователю возможность предварительного просмотра сгенерированного отчета перед его сохранением или отправкой.

1. Экспорт отчета:

Пользователи должны иметь возможность экспортировать отчет в различные форматы (например, PDF, Excel, CSV) для последующего использования или распечатки.

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения данного курсового проекта была разработана информационная система "Регистрация командировочных удостоверений". Проект был выполнен в соответствии с поставленными требованиями и целями, определенными в начале процесса разработки.

**Достижения проекта:**

Разработка базы данных: Была спроектирована и реализована база данных для хранения информации о командировочных удостоверениях с использованием MySQL.

Разработка прототипа системы: Был создан прототип информационной системы, обеспечивающий функции регистрации новых командировочных удостоверений, просмотра списка зарегистрированных удостоверений, поиска по критериям, редактирования и удаления существующих удостоверений.

Реализация модуля "Отчет": Был разработан модуль для генерации различных отчетов на основе данных о командировочных удостоверениях, предоставляющий пользователю возможность выбора типа отчета, настройки параметров и экспорта в различные форматы.

Интеграция с базой данных: Система успешно интегрирована с базой данных MySQL, обеспечивая сохранение и извлечение данных.

**Выводы:**

Разработанная информационная система представляет собой эффективный инструмент для учета и управления командировочными удостоверениями в компании. Ее использование позволит значительно упростить процессы регистрации и анализа данных о командировках, что повысит эффективность работы персонала и сократит временные затраты на административные процедуры.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Страчунский, М. В. Базы данных: Учебное пособие / М. В. Страчунский, И. Ю. Ткачев. - М.: Издательский центр "Академия", 2018. - 320 с. ISBN 978-5-94387-364-8.
2. Коннолли, Т. Управление базами данных: Современный подход / Т. Коннолли, К. Бегг. - М.: Вильямс, 2016. - 1280 с. ISBN 978-5-8459-2110-2.
3. C# 7.0 in a Nutshell: The Definitive Reference / Joseph Albahari, Ben Albahari. - O'Reilly Media, 2017. - 1080 p. ISBN 978-1491987650.
4. Entity Framework Core Documentation. [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/en-us/ef/core/
5. Microsoft SQL Server Documentation. [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/en-us/sql/
6. MySQL Documentation. [Электронный ресурс] – Режим доступа: https://dev.mysql.com/doc/
7. GalaSoft MVVM Light Toolkit Documentation. [Электронный ресурс] – Режим доступа: https://www.mvvmlight.net/
8. Microsoft WPF Documentation. [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/en-us/dotnet/desktop/wpf/?view=netdesktop-6.0
9. Шилдт Г. C# 8.0. Полное руководство / Шилдт Г. - М.: Вильямс, 2020. - 896 с. ISBN 978-5-8459-2087-7.
10. Freeman, A. Pro WPF 4.5 in C#: Windows Presentation Foundation in .NET 4.5 / Adam Nathan. - Apress, 2012. - 1113 p. ISBN 978-1430243656.