**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 «Библиотека»

Выполнил студент гр. ИСП-22 /\_\_\_\_\_\_\_\_\_\_\_\_\_/Мороз Александр Дмитриевич/

(подпись) (Ф.И.О.)

**ОЦЕНКА:\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

**ПРОВЕРИЛ**

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

(подпись) (Ф.И.О.)

Воркута

2025

**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. ПРОЕТИРОВАНИЕ 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 …..**

**3.2.4 Разработка модуля …..**

**3.2.5 Разработка модуля «Отчет»**

**ЗАКЛЮЧЕНИЕ**

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

**ВВЕДЕНИЕ**

В современном мире автоматизация библиотечных процессов играет важную роль в повышении удобства использования библиотечных услуг. Традиционные методы аренды книг требуют значительных временных затрат как со стороны пользователей, так и сотрудников библиотеки. Кроме того, ручное оформление аренды и возврата книг может приводить к ошибкам в учете, потере данных и снижению общей эффективности работы.

В связи с этим возникает необходимость внедрения информационной системы, которая позволит автоматизировать процессы аренды и возврата книг, а также упростит управление библиотечным фондом. Внедрение такой системы сократит очереди, ускорит обработку операций и повысит точность учета. Кроме того, это снизит нагрузку на библиотекарей, перераспределив их задачи в пользу более значимых аспектов работы.

Данный курсовой проект посвящен разработке информационной системы для библиотеки, основанной на кассах самообслуживания. Система позволит пользователям арендовать и возвращать книги путем ввода их ID или сканирования QR-кода, что обеспечит удобство взаимодействия с системой и минимизирует вероятность ошибок.

Сотрудники библиотеки после авторизации смогут управлять базой данных, добавляя новые книги, редактируя информацию о них и контролируя учетные записи пользователей. Дополнительно система будет включать модули отчетности, что облегчит анализ данных о популярности книг, своевременности возвратов и общем состоянии фонда.

В ходе работы над проектом будут рассмотрены основные этапы разработки: выбор инструментов, проектирование базы данных, реализация функциональных модулей и тестирование системы. Итогом работы станет полнофункциональная информационная система, способная оптимизировать работу библиотеки и повысить качество предоставляемых услуг.

**ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ**

**1.1 Платформа .NET**

Платформа .NET представляет собой мощную среду разработки, разработанную корпорацией Microsoft. Она предоставляет широкий набор инструментов и библиотек для создания различных типов приложений, включая десктопные, веб- и мобильные решения. Одним из главных преимуществ .NET является его кроссплатформенность, благодаря чему разработчики могут создавать программное обеспечение для Windows, macOS и Linux.

.NET включает в себя Common Language Runtime (CLR) — среду выполнения, отвечающую за управление памятью, обработку исключений и многопоточность. Также в составе платформы присутствует набор стандартных библиотек (Base Class Library), включающий в себя функционал для работы с файлами, сетями, базами данных и другими необходимыми ресурсами.

Выбор .NET в качестве платформы для разработки информационной системы обусловлен его надежностью, удобством использования, высокой производительностью и активной поддержкой сообщества. Кроме того, .NET отлично интегрируется с другими технологиями Microsoft, такими как SQL Server и WPF.

**1.2 Язык программирования C#**

C# (C-Sharp) — это объектно-ориентированный язык программирования, разработанный Microsoft специально для платформы .NET. Он обладает выразительным синтаксисом, богатым функционалом и удобством для разработки современных приложений.

Ключевые особенности C#:

* Строгая типизация данных и мощная система типов;
* Поддержка парадигмы объектно-ориентированного программирования (ООП);
* Интеграция с .NET и использование его библиотек;
* Высокий уровень безопасности кода за счет механизма управления памятью (Garbage Collector);
* Простота работы с многопоточностью и асинхронными операциями.

Использование C# для разработки информационной системы библиотеки позволит добиться высокой производительности, удобства сопровождения кода и интеграции с базой данных SQL Server.

**1.3 Windows Presentation Foundation (WPF)**

Windows Presentation Foundation (WPF) — это графическая подсистема для создания пользовательских интерфейсов в Windows-приложениях. Она заменяет устаревшие технологии Windows Forms и предлагает разработчикам гибкие инструменты для создания современных, интерактивных интерфейсов.

Основные преимущества WPF:

* Разделение логики приложения и представления (поддержка архитектурной модели MVVM);
* Гибкость в разработке UI благодаря языку разметки XAML;
* Поддержка аппаратного ускорения графики с использованием DirectX;
* Встроенные механизмы анимации, привязки данных и масштабируемости интерфейса.

WPF будет использоваться в проекте для создания удобного и понятного интерфейса системы аренды книг.

**1.4 СУБД SQL Server**

Microsoft SQL Server — это реляционная система управления базами данных (РСУБД), разработанная корпорацией Microsoft. В основе работы SQL Server лежит язык запросов Transact-SQL (T-SQL), который представляет собой расширение стандарта ANSI SQL. Данная СУБД предназначена для управления базами данных различных масштабов — от персональных решений до корпоративных систем. SQL Server активно конкурирует с другими аналогичными СУБД на рынке.

**Основные особенности SQL Server:**

* Высокая производительность и надежность, обеспечиваемые клиент-серверной архитектурой;
* Простота управления благодаря удобному интерфейсу и мощным административным инструментам;
* Высокий уровень безопасности данных, включая механизмы шифрования и резервного копирования;
* Поддержка транзакционной обработки данных, обеспечивающая целостность информации.

Использование SQL Server позволяет автоматизировать бизнес-процессы, выполнять анализ данных в режиме реального времени, а также управлять ресурсами базы данных.

**1.5 Microsoft SQL Server Management Studio**

Microsoft SQL Server Management Studio (SSMS) — это графическая среда для управления, конфигурирования и администрирования всех компонентов SQL Server. В состав SSMS входит редактор SQL-запросов и удобный инструмент для работы с объектами базы данных.

**Основные возможности SSMS:**

* Управление серверами и базами данных через Object Explorer;
* Поддержка служб Analysis Services, Integration Services и Reporting Services;
* Настройка безопасности и администрирование пользователей;
* Возможность мониторинга работы сервера и анализа производительности запросов.

Кроме основной версии SSMS, существует облегченная версия SQL Server Management Studio Express, предназначенная для работы с бесплатными редакциями SQL Server. Однако она не поддерживает ряд корпоративных функций, таких как Analysis Services и Reporting Services.

Для пользователей, работающих на macOS и Linux, альтернативным инструментом является **Azure Data Studio**, обеспечивающий многоплатформенную поддержку и возможности администрирования баз данных в облаке.

**1.6 Entity Framework**

Entity Framework (EF) — это технология объектно-реляционного отображения (ORM), используемая для работы с базами данных в среде .NET. Она позволяет разработчикам взаимодействовать с базой данных на уровне объектов, упрощая процесс работы с данными и снижая объем кода SQL-запросов.

**Преимущества Entity Framework:**

* Автоматическое создание и обновление базы данных;
* Поддержка LINQ-запросов для удобной работы с данными;
* Возможность работы с несколькими моделями взаимодействия (Code First, Database First, Model First);
* Высокая производительность и удобство использования.

В данном проекте Entity Framework будет использоваться для удобного взаимодействия с SQL Server, сокращения количества ручного написания SQL-запросов и ускорения процесса разработки.

Таким образом, выбор перечисленных инструментов обусловлен их широкими возможностями, удобством разработки, поддержкой современных технологий и интеграцией друг с другом. Этот стек обеспечит стабильную, надежную и удобную в использовании систему аренды книг с кассами самообслуживания.

**ГЛАВА 2. ПРОЕТИРОВАНИЕ БАЗЫ ДАННЫХ**

**2.1 Разработка диаграммы ERD**

**2.1 Разработка диаграммы ERD**

ERD (Entity Relationship Diagram) — это диаграмма, отображающая сущности базы данных и связи между ними. Она позволяет визуализировать структуру базы данных, определить основные сущности, их атрибуты и взаимосвязи для эффективного проектирования системы. В данном проекте база данных разработана для автоматизации аренды книг в библиотеке через кассу самообслуживания.

**Сущности и их атрибуты:**

1. **Книга (Book)**
   * **PK**: ID книги (BookID)
   * Название (Title)
   * Автор (Author)
   * Жанр (Genre)
   * Год издания (YearPublished)
   * Статус доступности (AvailabilityStatus)
2. **Читатель (Reader)**
   * **PK**: ID читателя (ReaderID)
   * ФИО (FullName)
   * Дата рождения (DateOfBirth)
   * Адрес (Address)
   * Телефон (Phone)
   * Долг по штрафам (PenaltyAmount)
3. **Аренда (Rental)**
   * **PK**: ID аренды (RentalID)
   * **FK**: ID книги (BookID)
   * **FK**: ID читателя (ReaderID)
   * Дата аренды (RentalDate)
   * Дата возврата (ReturnDate)
   * Статус возврата (ReturnStatus)
4. **Штраф (Penalty)**
   * **PK**: ID штрафа (PenaltyID)
   * **FK**: ID читателя (ReaderID)
   * Сумма штрафа (Amount)
   * Причина (Reason)
   * Дата начисления (PenaltyDate)
   * Оплачен (IsPaid)
5. **Касса самообслуживания (SelfServiceKiosk)**
   * **PK**: ID кассы (KioskID)
   * Локация (Location)
   * Статус работы (OperationalStatus)
6. **Транзакция (Transaction)**
   * **PK**: ID транзакции (TransactionID)
   * **FK**: ID аренды (RentalID)
   * **FK**: ID штрафа (PenaltyID)
   * **FK**: ID кассы (KioskID)
   * Дата операции (TransactionDate)
   * Тип операции (OperationType
   * Сумма (Amount)
7. **Библиотекарь (Librarian)**
   * **PK**: ID библиотекаря (LibrarianID)
   * ФИО (FullName)
   * Логин (Login)
   * Пароль (Password)

**Связи между сущностями:**

* Книга (Book) связана с Арендой (Rental) отношением Один-ко-многим (One-to-Many), так как одна книга может быть арендована многократно.
* Читатель (Reader) связан с Арендой (Rental) отношением Один-ко-многим (One-to-Many), поскольку один читатель может арендовать множество книг.
* Читатель (Reader) связан с Кассой Cамообслуживания (SelfServiceKiosk) отношением Один-ко-одному (One-to-One), так как библиотекарь использует кассу для аренды, возрата книг и оплаты штрафов.
* Читатель (Reader) связан с Штрафами (Penalty) отношением Один-ко-многим (One-to-Many), так как один читатель может получать несколько штрафов.
* Аренда (Rental) связана с Транзакцией (Transaction) отношением Один-к-одному (One-to-One), потому что каждая аренда сопровождается одной транзакцией.
* Штраф (Penalty) связан с Транзакцией (Transaction) отношением Один-к-одному (One-to-One), так как каждая оплата штрафа фиксируется одной транзакцией.
* Касса самообслуживания (SelfServiceKiosk) связана с Транзакцией (Transaction) отношением Один-ко-многим (One-to-Many), так как через одну кассу проходит множество операций.
* Библиотекарь (Librarian) связан с Кассой Cамообслуживания (SelfServiceKiosk) отношением Один-ко-одному (One-to-One), так как библиотекарь использует кассу для редактирования базы данных при поступлении и списании книг.

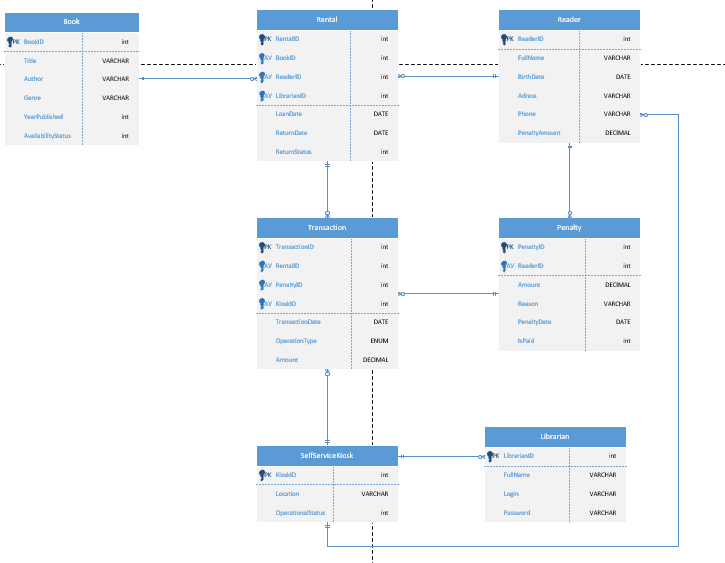


Рис. 2.1.1 ER-Диаграмма