

**Министерство образования и науки Республики  
Коми**

**Государственное профессиональное образовательное учреждение  
«Воркутинский арктический горно-политехнический колледж»**

**Допускается к защите**

**Первый заместитель  
директора**

\_\_\_\_\_ Романов А.В.

"\_\_" \_\_\_\_\_ 2024 г.

**ДИПЛОМНЫЙ ПРОЕКТ**

**Тема «Разработка автоматизированной информационной системы для  
библиотеки.»**

**ДП. 09.02.07.24.43 ПЗ**

**Выполнил студент  
Зверева Анастасия Ивановна**

\_\_\_\_\_ **подпись**

**Руководитель  
Чирков Иван Сергеевич**

\_\_\_\_\_ **подпись**

**Рецензент  
Кунский Ярослав Игоревич**

\_\_\_\_\_ **подпись**

Воркута, 2024 г.

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КОМИ**  
**Государственное профессиональное образовательное учреждение**  
**«Воркутинский арктический горно-политехнический колледж»**

УТВЕРЖДАЮ  
Первый заместитель  
директора

\_\_\_\_\_ Романов А.В.

"29" февраля 2024 г.

**ЗАДАНИЕ**

**для дипломного проекта**

Студенту: Зверевой Анастасии Ивановне

Группа: ИСП-20

Специальность: Информационные системы и программирование

Код: 09.02.07

Руководитель: Чирков Иван Сергеевич

**ТЕМА ДИПЛОМНОГО ПРОЕКТА:**

**«Разработка автоматизированной информационной системы для  
библиотеки»**

**СОДЕРЖАНИЕ**

- 1 Введение
- 2 Основная часть:
  - Глава 1. Техническое задание и выбор инструментария
  - Глава 2. Проектирование базы данных
  - Глава 3. Разработка информационной системы
  - Глава 4. Функциональность системы
- 3 Заключение
- 4 Список используемых источников и литературы
- 5 Рецензия

Дата выдачи задания "29" февраля 2024 г.

УТВЕРЖДАЮ  
Первый заместитель  
директора  
  
\_\_\_\_\_ Романов А.В.  
  
"29" февраля 2024 г.

Специальность: 09.02.07 Информационные системы и программирование

Индивидуальный план задание подготовки и выполнения  
дипломного проекта

Фамилия, имя, отчество студента: Зверева Анастасия Ивановна

Группа №: ИСП-20

Тема дипломного проекта: «Разработка автоматизированной информационной системы для библиотеки»

Фамилия, имя, отчество руководителя: Чирков Иван Сергеевич

Наименование действий	Исполнители	Сроки выполнения	Контроль	Отметка о выполнении
Выбор темы дипломной работы (проекта)	Студент руководитель	Не позднее 2-х недель после начала производственной практики	Руководитель	
Составление библиографического списка	Студент Консультация руководителя	29.03.2024г.	Руководитель	
Сбор материалов, подготовка плана дипломной работы (проекта)	Студент Консультация руководителя	12.04.2024г.	Руководитель	
Составление перечня заданий по работе на период прохождения практики	Руководитель Методист	Не позднее 2-х недель после начала производственной практики	Первый зам. директора	
Описания итогов выполнения задания, проведенного во время прохождения производственной практики (отчёт о прохождении практики)	Студент	Во время практики	Руководитель	
Консультирование по написанию теоретической части	Руководитель	По графику	Руководитель	
Написание теоретической части	Студент	До 03.06.2024г.	Руководитель	

Представление руководителю первого варианта теоретической части работы и обсуждение представленного материала	Студент	17.05.2024г.	Руководитель	
Составление окончательного варианта РАБОТЫ	Студент	10.06.2024г.	Руководитель	
Заключительное консультирование	Руководитель	03.06.2024г.		
Оформление работы в соответствии с требованиями, подготовка отзыва руководителя	Студент Руководитель	За 2 недели до защиты	Первый зам. директора	
Рецензирование дипломной работы (проекта)	Рецензент	10 дней до защиты	Первый зам. директора	
Ознакомление студента с рецензией на ДР/ДП	Руководитель	Не позднее 2-х дней до защиты	Руководитель	
Защита дипломной работы (проекта)	Студент	По графику	Комиссия	
Сдача дипломной работы (проекта) на хранение		После проведения защиты	Первый зам. директора	

Руководитель \_\_\_\_\_  
(подпись)

**«Ознакомлен»**

Зверева Анастасия Ивановна  
(Ф.И.О. студента)

"29" февраля 2024 г. \_\_\_\_\_  
(дата) (подпись)

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	6
ГЛАВА 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ И ВЫБОР ИНСТРУМЕНТАРИЯ .....	9
1.1 Техническое задание .....	9
1.2 Выбор инструментария .....	12
Язык программирования C# .....	12
СУБД SQL Server.....	14
Microsoft SQL Server Management Studio .....	17
Entity Framework .....	18
ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ.....	20
2.1 Разработка диаграммы ERD .....	20
2.2 Разработка диаграммы Use Case .....	22
2.3 Разработка базы данных .....	24
2.4 Словарь данных .....	29
ГЛАВА 3. РАЗРАБОКА ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	32
3.1 Создание дизайна пользовательского интерфейса.....	32
ГЛАВА 4. ФУНКЦИОНАЛЬНОСТЬ СИСТЕМЫ .....	36
Глава 4.1 Функционал .....	36
ЗАКЛЮЧЕНИЕ.....	45
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ. И ЛИТЕРАТУРЫ.....	47

## **ВВЕДЕНИЕ**

С развитием информационных технологий в современном обществе возникает необходимость в постоянном управлении и организации библиотечных процессов. Автоматизация библиотечной деятельности с использованием информационных систем становится неотъемлемой частью библиотечной практики. В данной работе рассматривается разработка автоматизированной информационной системы для библиотеки, которая направлена на оптимизацию управления библиотекой, учета читателей, организацию процесса выдачи и возврата литературы, а также обеспечение удобного доступа к ресурсам.

В современном мире библиотеки играют важную роль в распространении знаний и информации. Однако, с увеличением объема информации и изменением потребностей пользователей, библиотекам необходимо адаптироваться к новым вызовам. Автоматизация библиотечных процессов позволяет существенно улучшить качество обслуживания пользователей, оптимизировать управление библиотекой, повысить эффективность работы персонала и обеспечить более удобный доступ к ресурсам.

Целью данной работы является разработка автоматизированной информационной системы для библиотеки, которая будет способствовать повышению эффективности управления библиотечными ресурсами, обеспечивать удобство пользования услугами библиотеки как для посетителей, так и для персонала. Разработка данной информационной системы заключается в создании комплексной программы, охватывающей все основные аспекты работы библиотеки, начиная от учета литературы и заканчивая обслуживанием пользователей.

Данная работа состоит из нескольких разделов, каждый из которых посвящен определенному аспекту разработки автоматизированной

информационной системы для библиотеки. В ходе исследования будут рассмотрены основные этапы проектирования и разработки системы, ее функциональные возможности, а также практическая значимость подобных систем в библиотечных учреждениях.

Объектом дипломной работы является: база данных информационной системы библиотеки. Разработка и оптимизация базы данных для информационной системы библиотеки играет важную роль в эффективном функционировании системы, поскольку она хранит и обрабатывает информацию о читателях, операциях по выдаче и возврату книг, а также другие данные, необходимые для работы библиотеки.

Предмет дипломной работы: Автоматизация процесса работы библиотеки за счёт повышения эффективности работы, сокращения временных затрат на задачи, улучшение качества обслуживания посетителей и оптимизировать операциях по выдаче и возврату книг и процесс в целом.

Целью данной дипломной работы является: разработка и информационной системы, которая обеспечит эффективный учёт посетителей библиотеки, автоматизацию процессов оповещения посетителей, а также повышение качества и скорости обработки информации. Разработка информационной системы направлена на оптимизацию процессов и обеспечение более эффективного управления данными, что в конечном итоге способствует повышению качества работы сотрудников.

Для достижения цели продукта нужно выполнить следующие задачи.

Составить техническое задание к информационной системе:

- определение функциональных требований, таких как регистрация посетителей, ввод и обработка данных, генерация отчётов, отправка уведомлений на почту.

- описание пользовательских ролей и их прав доступа к функциональности системы.
- формирование документации по проекту, включая описание процессов и сценариев использования.

Определиться с инструментарием.

Спроектировать базу данных информационной системы:

- определение связей между сущностями и нормализация базы данных для предотвращения избыточности данных.
- разработка диаграммы базы данных, включая ER-диаграмму для визуализации структуры базы данных.
- учёт требований к производительности и безопасности при проектировании базы данных.

Разработать информационную систему:

- создание дизайна пользовательского интерфейса с учётом удобства использования и эффективности работы пользователей.
- написание кода для реализации функциональности системы, используя современные технологии разработки программного обеспечения.

Оптимизировать временные затраты сотрудников путём автоматизации:

- разработка процесса оповещения посетителей библиотеки и выявление этапов, подлежащих автоматизации.
- разработка решений для ускорения процессов, таких как автоматическое заполнение форм, генерация отчётов



# **ГЛАВА 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ И ВЫБОР ИНСТРУМЕНТАРИЯ**

## **1.1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**1.Наименование работы:** Разработка автоматизированной информационной системы для библиотеки.

### **2. Назначение разработки**

Разработка автоматизированной информационной системы для библиотеки представляет собой шаг в направлении автоматизации процессов библиотеки. Это позволит значительно упростить процедуры выдачи и приёма литературы, сократить временные затраты на обработку документов и повысить качество работы персонала.

### **3. Требования к программе:**

#### **3.1. Требования к функциональным характеристикам:**

##### **Авторизация:**

Разрабатываемая система будет иметь модульную структуру, доступ к модулям будет зависеть от роли пользователя. Доступ возможен только для авторизованных пользователей.

В связи с этим при запуске системы первым окном будет окно входа.

Для реализации регистрации и авторизации необходимо создать базу данных с необходимыми таблицами и заполнить их тестовыми данными.

Доступ к системе имеют только сотрудники. У каждого сотрудника есть логин, пароль по которому и разграничиваются права доступа.

##### **Алгоритм регистрации:**

- сотрудник вписывает свое полное имя, почту и вводит свой пароль;
- при вводе пароля сотрудником и нажатии кнопку «Зарегистрироваться» происходит проверка и вход в систему.

При введении почты, сотруднику приходит на электронную почту код, после которого нужно ввести его в поле подтверждения аккаунта на почте. В следствии чего пользователь является зарегистрированным в системе.

**Алгоритм авторизации:**

- сотрудник вписывает свою почту и вводит свой пароль;
- при вводе пароля сотрудником и нажатии кнопку войти происходит проверка и вход в систему.

При введении пароля появляется приветствующее окно, после чего пользователь успешно входит в систему.

**Разграничение прав доступа:**

Библиотекарь имеет возможность: добавлять, удалять, изменять посетителей, а также вести учёт книг, может добавлять, удалять, изменять списки литературы, создавать отчёты.

Администратор базы данных имеет возможность: добавлять, удалять, изменять пользователей таких как библиотекарь.

**Система:**

Информационная система выполняет учёт всех данных системы таких как:

- Ведется хранение информации по книгам, учёт всех данных по литературе.
- Ведется учёт информации по арендованным книгам, а также ведется контроль возврата взятой в аренду литературы.

Первичные данные для учёта информации о литературе библиотеки:

- Количество книг, жанры, название, год выпуска.

Первичные данные для учёта информации о посетителях библиотеки:

- ФИО, почта.

Отчёты показывают количество выданной литературы, в том числе дату возврата, и данные посетителя.

### **3.2. Специальные требования**

Программное обеспечение должно иметь дружелюбный интерфейс, рассчитанный на пользователя средней квалификации (с точки зрения компьютерной грамотности).

## **4. Требования к программной документации**

В ходе разработки программы должны быть подготовлены: описание программы.

## **5. Экономический эффект от информационной системы**

Разработка информационной системы направлена на оптимизацию процессов библиотеки и обеспечение более эффективного управления данными, что в конечном итоге способствует повышению качества выполняемой работы сотрудниками.

Экономический эффект от автоматизированной информационной системы для библиотеки ожидается за счет сокращения времени на выполняемые сотрудниками операции, исключения ошибок при формировании отчетов, увеличения времени на другие работы.

## 1.2 ВЫБОР ИНСТРУМЕНТАРИЯ

### ЯЗЫК ПРОГРАММИРОВАНИЯ C#

C# — объектно-ориентированный, ориентированный на компоненты язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО. В основном C# — объектно-ориентированный язык. Вы определяете типы и их поведение.

Вот лишь несколько функций языка C#, которые позволяют создавать надежные и устойчивые приложения. Сборка мусора автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами. Типы, допускающие значение null, обеспечивают защиту от переменных, которые не ссылаются на выделенные объекты. Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них. Лямбда-выражения поддерживают приемы функционального программирования.

Поддержка языков для асинхронных операций предоставляет синтаксис для создания распределенных систем. В C# имеется Единая система типов. Все типы C#, включая типы-примитивы, такие как int и double, наследуют от одного корневого типа object. Все типы используют общий набор операций, а значения любого типа можно хранить, передавать и обрабатывать схожим образом. Более того, C# поддерживает как определяемые пользователями ссылочные типы, так и типы значений. C# позволяет динамически выделять объекты и хранить упрощенные структуры в стеке. C# поддерживает универсальные методы и типы, обеспечивающие повышенную безопасность типов и производительность. C# предоставляет

итераторы, которые позволяют разработчикам классов коллекций определять пользовательские варианты поведения для клиентского кода.

Приложения C# получают преимущества автоматического управления памятью среды выполнения .NET. Приложения C# также используют обширные библиотеки среды выполнения, предоставляемые пакетом SDK для .NET. Некоторые компоненты являются независимыми от платформы, такими как библиотеки файловой системы, коллекции данных и математические библиотеки. Другие относятся к одной рабочей нагрузке, например к веб-библиотекам ASP.NET Core или библиотеке пользовательского интерфейса .NET MAUI. Многофункциональная экосистема Open Source в NuGet расширяет библиотеки, которые являются частью среды выполнения. Эти библиотеки предоставляют еще больше компонентов, которые можно использовать.

C# находится в семействе языков C. Синтаксис C# знаком, если вы использовали C, C++, JavaScript или Java. Как и все языки в семействе C, точки с запятой (;) определяют конец инструкций. Идентификаторы C# чувствительны к регистру. C# имеет то же самое использование фигурных скобок, {а также операторов управления, таких как, и, и switchциклов конструкций, таких как elseiffor, и. while} C# также имеет инструкцию foreach для любого типа коллекции.

Каждая объявленная переменная имеет тип, известный во время компиляции. Компилятор или средства редактирования сообщают вам, используется ли этот тип неправильно. Эти ошибки можно исправить, прежде чем вы когда-либо запускаете программу. Основные типы данных встроены в язык и среду выполнения: такие типы значений, как, doublecharссылочные типы, такие как intstringмассивы и другие коллекции. При написании программ вы создаете собственные типы. Эти типы могут быть struct типами значений или class типами, определяющими объектно-

ориентированное поведение. Модификатор можно добавить record в любой struct или class типы, чтобы компилятор синтезирует код для сравнения равенства. Можно также создавать interface определения, определяющие контракт или набор элементов, которые должны предоставлять тип, реализующий этот интерфейс. Вы также можете определить универсальные типы и методы. Универсальные шаблоны используют параметры типа для предоставления заполнителя фактического типа при использовании.

При написании кода вы определяете функции, также называемые методами struct, как элементы и class типы. Эти методы определяют поведение типов. Методы могут быть перегружены с разными числами или типами параметров. Методы могут при необходимости возвращать значение. Помимо методов, типы C# могут иметь свойства, которые являются элементами данных, поддерживаемыми функциями, называемыми методами доступа. Типы C# могут определять события, позволяющие типу уведомлять подписчиков о важных действиях. C# поддерживает объектно-ориентированные методы, такие как наследование и полиморфизм для class типов.

Приложения C# используют исключения для создания отчетов и обработки ошибок. Вы будете знакомы с этой практикой, если вы использовали C++ или Java. Код создает исключение, если оно не может сделать то, что было предназначено. Другой код, независимо от количества уровней стека вызовов, может при необходимости восстановиться с помощью try - catch блока.

## **СУБД SQL Server**

СУБД служит посредником между пользователем и БД. Сама структура БД хранится в виде набора файлов, и единственный способ получить доступ к данным в этих файлах – через СУБД. Она представляет конечному пользователю (или прикладной программе) единое интегрированное

представление данных в БД. СУБД получает запросы приложений и переводит их в сложные операции, необходимые для выполнения. Она скрывает большую часть внутренней сложности БД от прикладных программ и пользователей. Прикладная программа может быть написана программистом с использованием языка программирования высокого уровня, такого как Python, Java или C#, или она может быть создана с помощью инструментов СУБД. Наличие СУБД между приложениями пользователя и БД дает некоторые важные преимущества. Во-первых, СУБД позволяет совместно использовать данные в БД нескольким приложениям или пользователям. Во-вторых, СУБД объединяет представления данных разных пользователей в единый универсальный репозиторий данных.

Большинство систем баз данных содержат подкомпонент, называющийся оптимизатором, который рассматривает несколько возможных стратегий исполнения запроса данных и выбирает из них наиболее эффективную. Выбранная стратегия называется планом исполнения запроса. Оптимизатор принимает решение, принимая во внимание такие факторы, как размер таблиц, к которым направлен запрос, существующие индексы и логические операторы (AND, OR или NOT), используемые в предложении WHERE.

Одной из стоящих перед системой баз данных задач является идентифицировать логически противоречивые данные и не допустить их помещения в базу данных. Кроме этого, для большинства реальных задач, которые реализовываются с помощью систем баз данных, существуют ограничения для обеспечения целостности (integrity constraints), которые должны выполняться для данных. (В качестве примера ограничения для обеспечения целостности можно назвать требование, чтобы табельный номер сотрудника был пятизначным целым числом.)

Обеспечение целостности данных может осуществляться пользователем в прикладной программе или же системой управления базами данных. До максимально возможной степени эта задача должна осуществляться посредством СУБД.

Чтобы упростить работу с такими хранилищами данных и повысить эффективность их применения, создаются специализированные системы управления. Одной из наиболее популярных является разработка от Microsoft – SQL Server. Первый релиз платформы опубликован еще в 1989 году, а последняя версия выпущена в 2019 году (проект продолжает развиваться).

Преимущества решения:

- тесная интеграция с операционной системой Windows;
- высокая производительность, отказоустойчивость;
- поддержка многопользовательской среды;
- расширенные функции резервирования данных;
- работа с удаленным подключением.

Это снижает сложность внедрения и затраты на процесс разработки собственных решений, адаптированных для «узких» задач. При написании программного кода активно используется интеграция с продуктами Microsoft, например, с платформой Visual Studio.

Появление таких продуктов позволило объединить разное понимание БД (баз данных) со стороны пользователей и системных администраторов. Системный подход включает файлы с табличными данными, связанными друг с другом.



Функции базы данных:

- постоянное хранение информации;
- поиск по ключевым критериям;
- чтение и редактирование по запросу.

Клиентами БД являются прикладные программы, их интерфейсы, различные интерактивные модули сайтов вроде калькуляторов и онлайн-редакторов. Но есть еще один компонент системы – СУБД. Он предназначен для ручного доступа к информации и позволяет извлекать данные на диск, работать с ними в памяти сервера, в том числе с применением структурированного языка SQL.

Всего различают три типа БД – клиент-серверные, файл-серверные и встраиваемые. MS SQL Server относится к первой категории. Плюс система является реляционной, т.е. адаптированной для хранения данных без избыточности, с минимальными рисками появления аномалий и нарушения целостности внутренних таблиц.

### **MICROSOFT SQL SERVER MANAGEMENT STUDIO**

SQL Server Management Studio (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования и разработки всех компонентов SQL Server. Бесплатная графическая среда, включающая набор инструментов для разработки сценариев на T-SQL и управления инфраструктурой Microsoft SQL Server.

С помощью SSMS Вы можете разрабатывать базы данных, выполнять инструкции T-SQL, а также администрировать Microsoft SQL Server.

Если в задачи входит полное сопровождение Microsoft SQL Server, начиная от создания баз данных, написания SQL запросов, создания

хранимых процедур и функций, и заканчивая администрированием SQL Server, включая управление безопасностью, то основным инструментом будет как раз среда SQL Server Management Studio.

Среда SQL Server Management Studio реализована только под Windows, поэтому если Вам нужен инструмент для работы с Microsoft SQL Server, который будет работать на других платформах, например, на Linux или macOS, то Вам следует использовать инструмент Azure Data Studio, который также является официальным инструментом, разработанным компанией Microsoft.

### **Entity Framework**

Entity Framework представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работаем с объектами.

Первая версия Entity Framework - 1.0 вышла еще в 2008 году и представляла очень ограниченную функциональность, базовую поддержку ORM (object-relational mapping - отображения данных на реальные объекты) и один единственный подход к взаимодействию с базой данных - Database First. С выходом версии 4.0 в 2010 году многое изменилось - с этого времени Entity Framework стал рекомендуемой технологией для доступа к данным, а в сам фреймворк были введены новые возможности взаимодействия с бд - подходы Model First и Code First.

Набор технологий в ADO.NET, которые поддерживают разработку программных приложений, ориентированных на данные.

Архитекторам и разработчикам приложений, ориентированных на обработку данных, приходится учитывать необходимость достижения двух совершенно различных целей. Они должны моделировать сущности, связи и логику решаемых бизнес-задач, а также работать с ядрами СУБД, используемыми для сохранения и получения данных. Данные могут распределяться по нескольким системам хранения данных, в каждой из которых применяются свои протоколы, но даже в приложениях, работающих с одной системой хранения данных, необходимо поддерживать баланс между требованиями системы хранения данных и требованиями написания эффективного и удобного для обслуживания кода приложения.

Платформа Entity Framework позволяет работать с данными в форме специфических для домена объектов и свойств (например, с клиентами и их адресами) без необходимости учитывать формат базовых таблиц и столбцов базы данных, где хранятся эти данные.

Entity Framework дает разработчикам возможность работать с данными на более высоком уровне абстракции, создавать и сопровождать приложения, ориентированные на работу с данными, одновременно с этим сокращая объем кода по сравнению с традиционными приложениями.

## ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

### 2.1 Разработка диаграммы ERD

Диаграммы отношений сущностей (ER) — это визуальные представления баз данных, которые показывают, как элементы в базе данных связаны друг с другом. Примерами сущностей (с точки зрения логической модели предметной области) или объектов (с точки зрения внешнего мира по отношению к ИС).

ER-диаграммы — «родственники» схем структуры данных (DSD), где вместо связей между самими сущностями отображаются отношения между элементами внутри них. ER-диаграммы часто используются в сочетании с диаграммами DFD, которые схематично показывают движение потоков информации в рамках процесса или системы. С объектами связано две проблемы — идентификация и адекватное описание. Для идентификации используют имя. При этом предполагается, что происходит отказ от его смысла, который присущ естественному языку. Используется только указательная функция имени. Имя — это прямой способ идентификации объекта. К косвенным способам идентификации объекта относят его свойства в их понимании как характеристики или признака.

Объекты взаимодействуют между собой через свои свойства, что порождает ситуации. Ситуации — это взаимосвязи, выражающие взаимоотношения между объектами. Ситуации в предметной области описываются посредством высказываний о предметной области с использованием исчисления высказываний и исчисления предикатов, т.е. формальной, математической логики.

ER-диаграммы применяются для моделирования и проектирования реляционных баз данных, причем как в плане логических и бизнес-правил (логические модели данных), так и в плане внедрения конкретных технологий (физические модели данных). В сфере разработки программного обеспечения ER-диаграмма, как правило, служит первым шагом в

определении требований проекта по созданию информационных систем. На дальнейших этапах работы ER-диаграммы также применяются для моделирования конкретных баз данных. Реляционная база данных сопровождается соответствующей реляционной таблицей и при необходимости может быть представлена в этом формате.

Основными сущностями являются: клиенты, книги, арендованные книги, жанры и пользователи. Каждая сущность обделена следующими атрибутами. У сущности «Клиенты» имеется: ФИО и почта. У сущности «Книга» имеется: название, автор, год, количество, наличие. У сущности «Арендованные книги» имеется: название книги, имя, фамилия, жанр, страницы, стоимость, дата выдачи, дата приема, наличие книги. У сущности «Жанры» имеется: название. У сущности «Роли» имеется: название. У сущности «Пользователь» имеется: полное имя, логин и пароль.

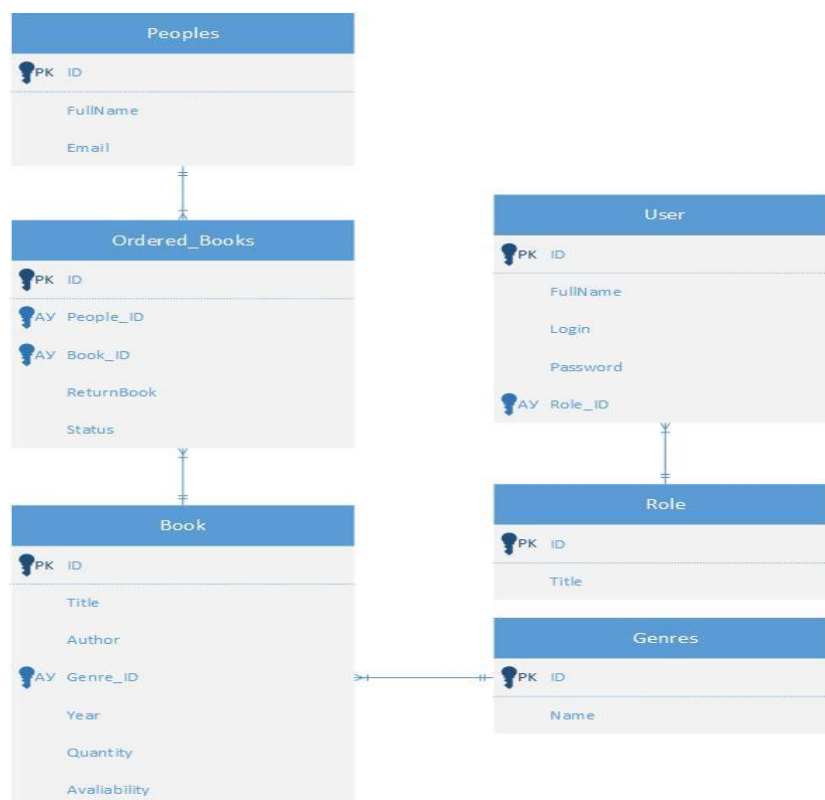


Рис. 1. Диаграмма ERD информационной системы библиотеки

## 2.2 Разработка диаграммы Use Case

Диаграммы прецедентов (Use Case Diagrams) являются важным инструментом в методологии UML (Unified Modeling Language) для визуализации функциональных требований системы. Они помогают идентифицировать различных актеров (пользователи или внешние системы), которые взаимодействуют с системой, и описывают функциональности, которые они могут выполнять. Они помогают командам разработки и заказчикам лучше понять функциональности, которые должны быть реализованы в системе.

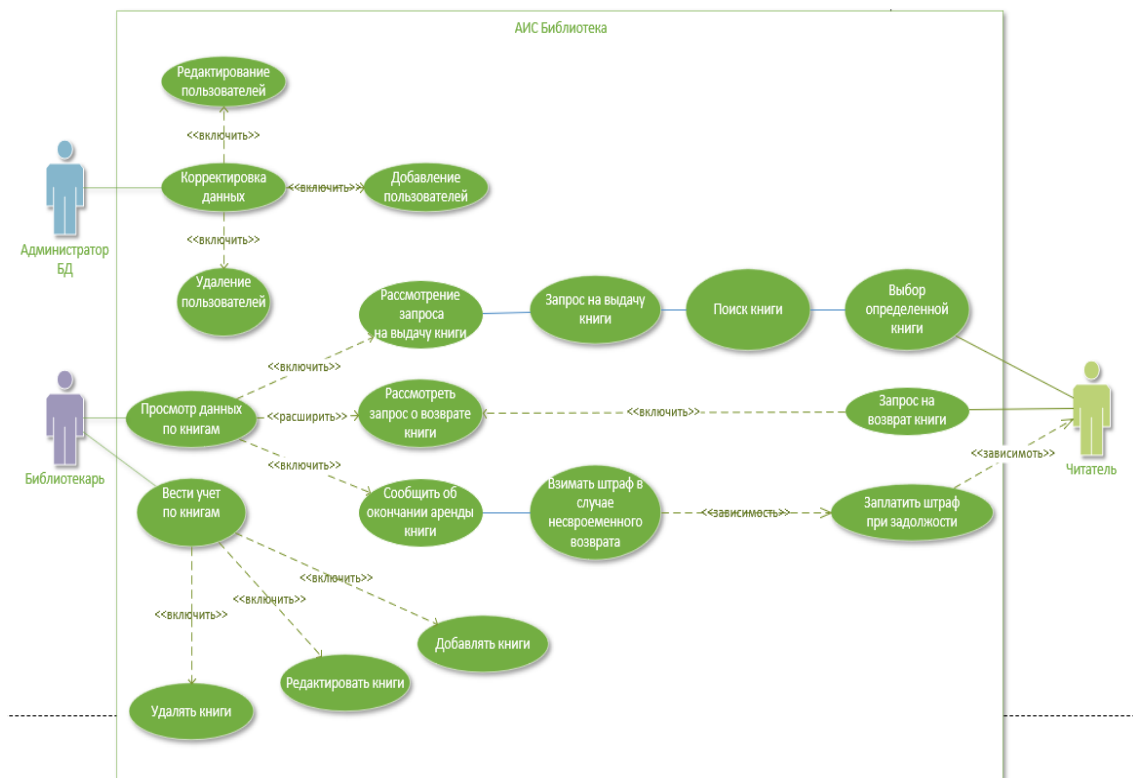


Рис. 2. Диаграмма Use Case информационной системы библиотеки

Основными актерами являются: администратор БД, библиотекарь и читатель. Каждый из актеров играют важную в информационной системе.

Администратор имеет полный доступ к данным о пользователях. Он может корректировать информацию о пользователях, добавлять новых

читателей, редактировать существующие записи и удалять учетные записи при необходимости.

Библиотекарь имеет возможность просматривать данные по книгам в библиотечном каталоге. Он может удалять книги из коллекции, редактировать информацию о книгах и добавлять новые позиции. Библиотекарь также отвечает за выдачу книг читателям, контроль сроков аренды и уведомление о необходимости возврата. В случае просрочки возврата книги библиотекарь может взимать штрафы с читателей.

Читатель может выбирать книгу из доступного ассортимента библиотеки для аренды.

## 2.3 РАЗРАБОТКА БАЗЫ ДАННЫХ

SQL Server Management Studio — это набор административных средств для управления компонентами, относящимися к SQL Server. Эта интегрированная среда позволяет пользователям выполнять разнообразные задачи, например резервное копирование данных, редактирование запросов и автоматизацию общих функций в одном интерфейсе.

Среда SQL Server Management Studio включает в себя следующие средства:

Редактор кода — богатый возможностями редактор скриптов для написания и редактирования скриптов. Среда SQL Server Management Studio предоставляет четыре версии редактора кода: редактор запросов ядра СУБД для скриптов Transact-SQL, редактор запросов многомерных выражений, редактор запросов расширения интеллектуального анализа данных и редактор запросов XML/A.

Обозреватель объектов для размещения, изменения, создания скрипта или выполнения объектов, принадлежащих экземплярам SQL Server.

Обозреватель шаблонов для размещения и написания сценариев шаблонов.

Обозреватель решений для организации и хранения связанных скриптов как части проекта.

Среда SQL Server Management Studio обеспечивает эффективность рабочих процессов, предоставляя:

Отключенный доступ. Можно писать и изменять скрипты, не соединяясь с экземпляром SQL Server.

Создание сценариев из любого диалогового окна. Можно создать скрипт из любого диалогового окна, а также читать, изменять, сохранять и многократно использовать скрипты после создания.



Затем следует процесс разработки базы данных на языке SQL, средствами для создания БД являются БД SQL Server Management Studio.

База данных АИС «Библиотека» была подготовлена способом создания диаграммы. Диаграммы базы данных графически изображают структуру баз данных. При помощи диаграмм баз данных можно создавать и изменять таблицы, столбцы, связи и ключи. Кроме того, можно изменять индексы и ограничения. Таблицы являются объектами, которые содержат все данные в базах данных. В таблицах данные логически организованы в виде строк и столбцов по аналогии с электронной таблицей. Каждая строка представляет собой уникальную запись, а каждый столбец — поле записи. Например, таблица, содержащая данные о сотрудниках компании, может иметь строку для каждого сотрудника и столбцы, представляющие сведения о сотрудниках (например, его идентификационный номер, имя, адрес и т.д.).

После запуска SQL Server Management Studio необходимо подключиться к серверу баз данных, на котором будет создаваться новая база данных. Для этого укажите имя сервера, тип аутентификации и учетные данные.

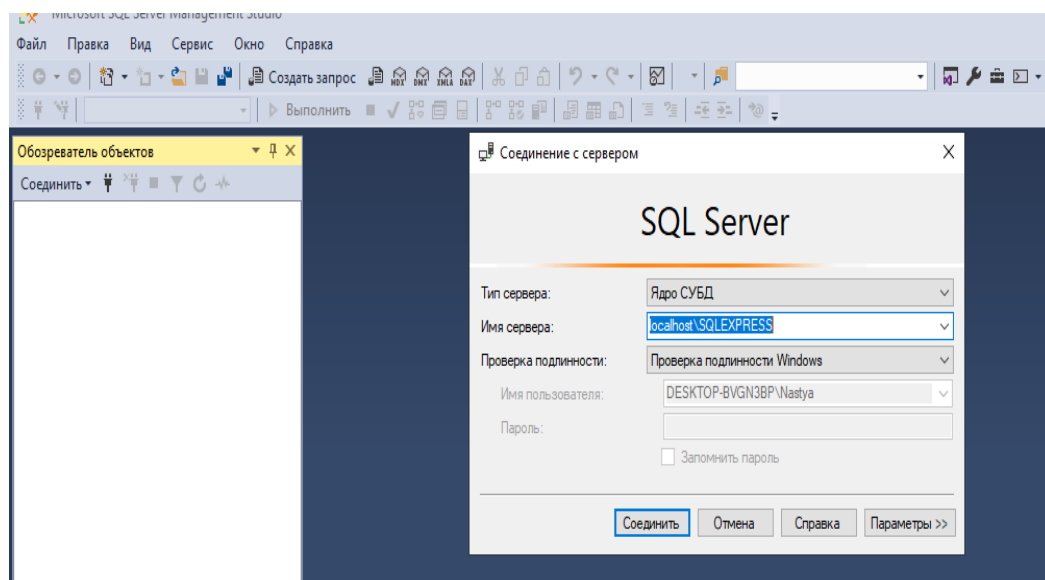


Рис. 3. Окно входа

### Создание новой базы данных:

Чтобы создать новую базу данных, щелкните правой кнопкой мыши на папке "Базы данных" в Object Explorer и выберите опцию "Новая база данных". В появившемся окне укажите имя базы данных, параметры хранения данных и логов, а также другие параметры, необходимые для создания базы данных.

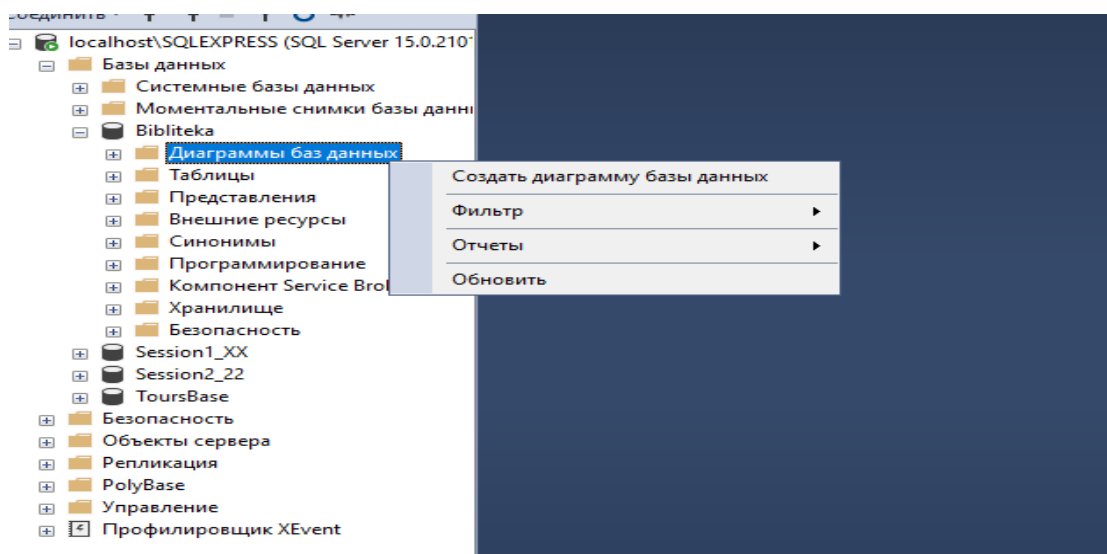


Рис. 4. Object Explorer

### Определение таблиц и структуры данных:

После создания базы данных определите необходимые таблицы и структуру данных. Создайте таблицы, определите поля и их типы данных, а также ограничения (например, первичные ключи, внешние ключи).

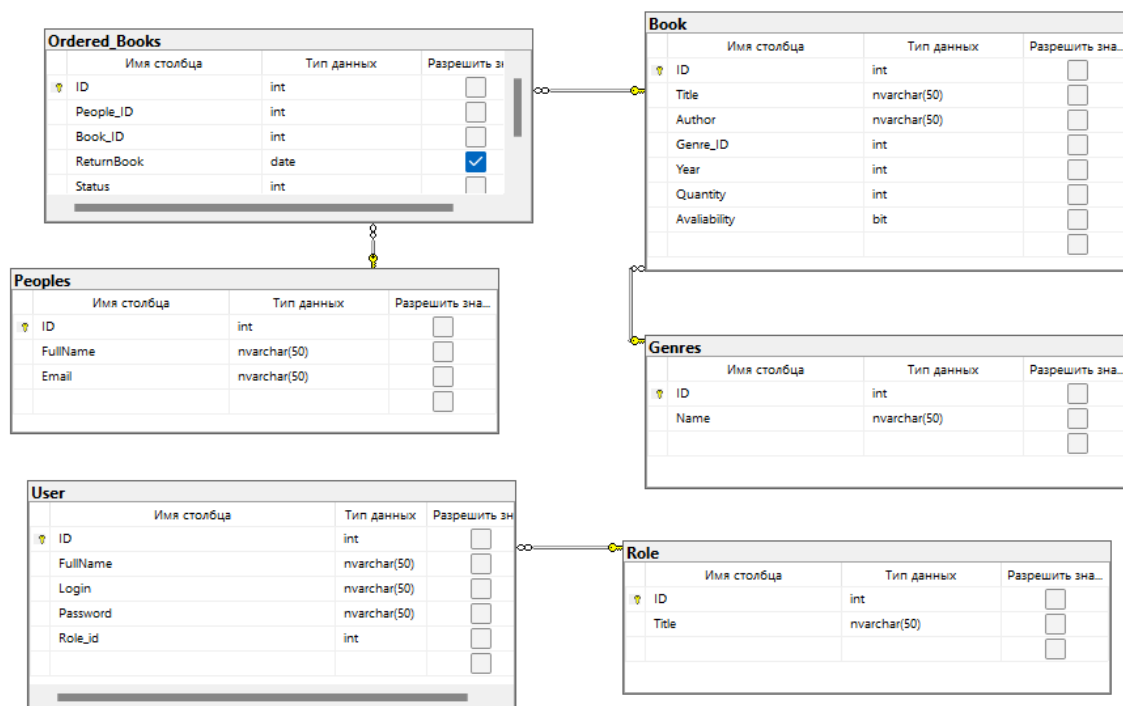


Рис. 5. Таблицы

В данной работе используется 6 таблиц базы данных:

User, Peoples, Order\_Books, Book, Genres, Role. Каждая таблица содержит уникальный ключ, атрибуты и связи.

В первую очередь, необходимо уделить внимание определению основных сущностей вашей информационной системы. Это могут быть, например, клиенты, продукты, заказы и т.д. Для каждой сущности определяются соответствующие таблицы в базе данных, а также их атрибуты, то есть характеристики или свойства, которые они имеют.

После этого следует продумать структуру каждой таблицы. Это включает в себя определение первичных и внешних ключей, которые обеспечивают связи между таблицами, а также выбор подходящих типов данных для каждого атрибута. Важно учитывать нормализацию данных для предотвращения избыточности и обеспечения целостности информации.

Этот этап создания базы данных для информационной системы в SQL Server Management Studio является ключевым для успешной реализации проекта и обеспечивает эффективную работу информационной системы.

## 2.4 СЛОВАРЬ ДАННЫХ

Словарь данных содержит информацию об источниках, взаимосвязях между данными, их описания, сведения о характере использования. Словарь данных можно рассматривать как вспомогательную базу данных, в которой хранится информация об основной базе данных.

Пользователи баз данных и разработчики приложений могут получить выгоду от единого стандартизированного документа словаря данных, который перечисляет организацию, содержимое, соглашения по одной или более баз данных. Это обычно включает в себя имена и описания различных таблиц и полей в каждой базе данных, дополнительные детали такие, как тип и длина каждого элемента данных. Не существует универсального стандарта, описывающего уровень детализации в подобном документе, но есть основное описание метаданных о структуре базы данных, а не о самих данных.

Одним из преимуществ хорошо спроектированного словаря данных является то, что он помогает упорядочить структуру базы данных или большого комплекса баз данных.

В следствии чего, ниже показан словарь данных для библиотеки в виде таблиц.

Таблица 1. Role

Data Dictionary				
Role				
KEY	FIELD NAME	DATA TYPE / FIELD SIZE	REQUIRED?	NOTES
Primary	Id	int	Y	Уникальный идентификатор
	Title	nvarchar (50)	Y	Название роли

Таблица 2. Peoples

Peoples				
KEY	FIELD NAME	DATA TYPE / FIELD SIZE	REQUIRED?	NOTES
Primary	Id	int	Y	Уникальный идентификатор
	FullName	nvarchar (50)	Y	Полное имя читателя
	Email	nvarchar (50)	Y	Электронная почта читателя

Таблица 3. User

User				
KEY	FIELD NAME	DATA TYPE / FIELD SIZE	REQUIRED?	NOTES
Primary	Id	int	Y	Уникальный идентификатор
Foregin	Role Id	int	Y	Вторичный идентификатор
	FullName	nvarchar (50)	Y	Полное имя пользователя для входа в систему
	Login	nvarchar (50)	Y	Логин пользователя для входа в систему

Таблица 4. Genres

Genres				
KEY	FIELD NAME	DATA TYPE / FIELD SIZE	REQUIRED?	NOTES
Primary	Id	int	Y	Уникальный идентификатор
	Name	nvarchar (50)	Y	Название жанра

Таблица 5. Book

Book				
KEY	FIELD NAME	DATA TYPE / FIELD SIZE	REQUIRED?	NOTES
Primary	Id	int	Y	Уникальный идентификатор
Foregin	Genre Id	int	Y	Вторичный идентификатор
	Title	nvarchar (50)	Y	Название роли
	Author	nvarchar (50)	Y	Автор книги, которую выбрал читатель
	Year	int	Y	Год выпуска книги, которую выбрал читатель
	Quantity	int	Y	Количество книг, которые выбрал читатель
	Avaliability	bit	Y	Наличие книги

Таблица 6. Ordered\_Books

Ordered_Books				
KEY	FIELD NAME	DATA TYPE / FIELD SIZE	REQUIRED?	NOTES
Primary	Id	int	Y	Уникальный идентификатор
Foregin	People Id	int	Y	Вторичный идентификатор
Foregin	Book Id	nvarchar (50)	Y	Вторичный идентификатор
	ReturnBook	date	N	Дата возврата книги
	Status	int	Y	Статус, обозначающий вернул или не вернул книгу читатель

## ГЛАВА 3. РАЗРАБОКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

### 3.1 Создание дизайна пользовательского интерфейса

Создание дизайна пользовательского интерфейса в Visual Studio включает в себя несколько шагов, которые позволяют создать удобный и привлекательный интерфейс для вашего приложения.

Первым шагом для создания дизайна пользовательского интерфейса в Visual Studio является создание нового проекта. Выберите тип проекта (например, Windows Forms Application или WPF Application) и укажите имя проекта и место его сохранения.

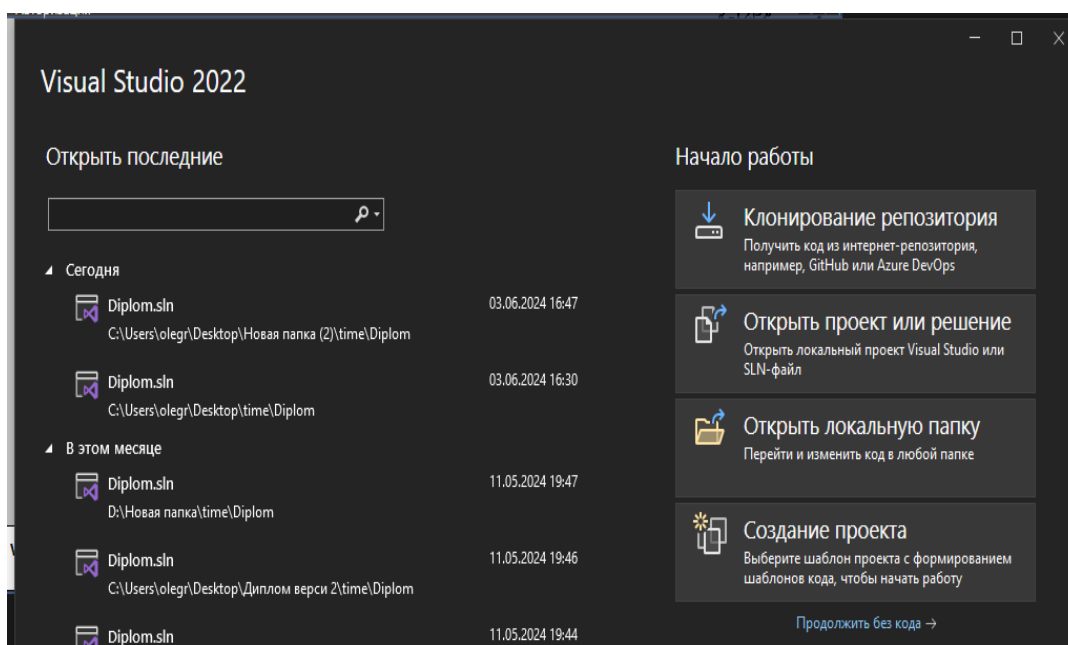


Рис. 6. создание нового проекта

Дизайн формы: Откройте файл формы в режиме дизайнера, где вы сможете создавать дизайн интерфейса визуально. Вы увидите пустую форму, на который можно размещать элементы управления.



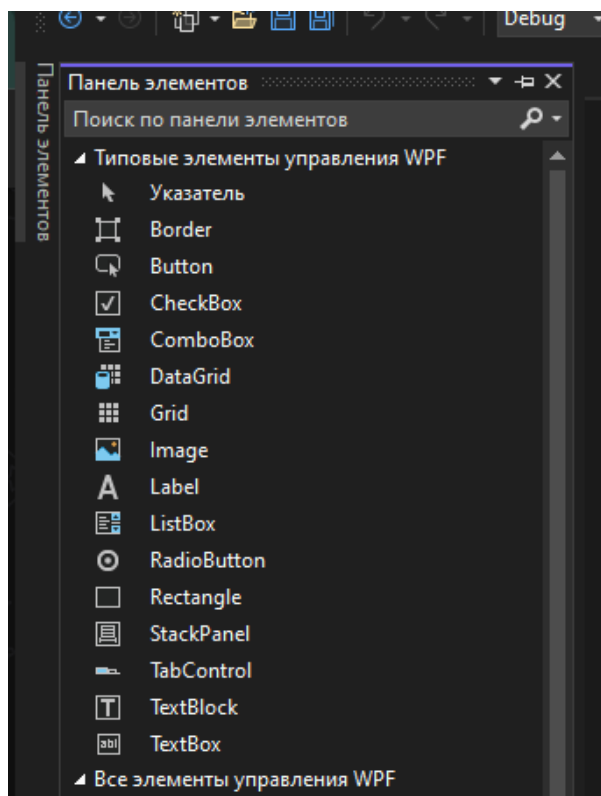


Рис. 7. Пример элементов

Добавление элементов управления: для добавления элементов управления на форму, нужно перейти к панели инструментов, на которой находятся все необходимые элементы: кнопки, текстовые поля, метки, списки и другие. Выберите элемент, перетащите его на форму и разместите его в нужном месте.

Настройка свойств элементов управления: После добавления элемента управления нужно открыть свойства этого элемента. На данном этапе можно настроить свойства элемента, такие как текст, цвет, размер, шрифт, выравнивание и другие параметры.

Дизайн макета: Расположите элементы управления на форме таким образом, чтобы дизайн был удобен для пользователя. Для корректного пользования можно использовать выравнивание, сетку и другие инструменты для создания привлекательного макета.

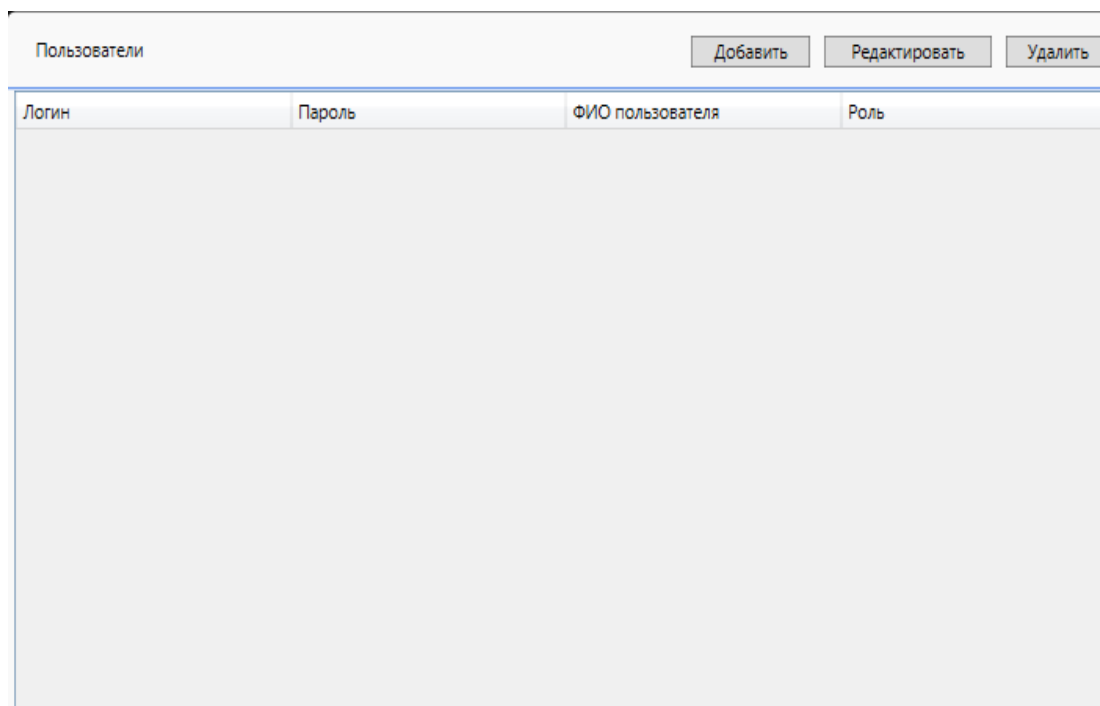


Рис. 8. Расположение элементов

Создание стилей и тем: В Visual Studio можно использовать стили и темы для создания единообразного дизайна вашего приложения. Вы можете создать собственные стили или воспользоваться готовыми темами.

```

19
20 <DataGrid x:Name="DGrid" Grid.Row="1" Margin="5,0,5,5" AutoGenerateColumns="False" IsReadOnly="True">
21     <DataGrid.Columns>
22         <DataGridTextColumn Binding="{Binding Login}" Header="Логин" Width="*" />
23         <DataGridTextColumn Binding="{Binding Password}" Header="Пароль" Width="*" />
24         <DataGridTextColumn Binding="{Binding FullName}" Header="ФИО пользователя" Width="*" />
25         <DataGridTextColumn Binding="{Binding Role.Title}" Header="Роль" Width="*" />
26     </DataGrid.Columns>
27 </DataGrid>

```

Рис. 9. Пример XAML разметки

Тестирование и отладка: После завершения дизайна интерфейса, запустите приложение для проверки работы элементов и визуального отображения. Проведите тестирование и отладку, чтобы устранить возможные ошибки.

Оптимизация для различных разрешений экрана: Учтите различные разрешения экрана и размеры устройств, чтобы ваш интерфейс выглядел хорошо на разных устройствах. Используйте адаптивный дизайн для того, чтобы ваш интерфейс корректно отображался на разных экранах.

Итак, процесс создания дизайна пользовательского интерфейса в Visual Studio требует внимания к деталям, креативности и тщательной работы с инструментами разработки.

## ГЛАВА 4. ФУНКЦИОНАЛЬНОСТЬ СИСТЕМЫ

### Глава 4.1 Функционал

Пользователь библиотеки может зарегистрироваться в приложение. Для этого есть 4 поля ввода данных, (Full\_Name) – поле ввода имени, (Email) – поле ввода почты, на которое отправляется четырёх значений код, (Password) – поле ввода пароля для даль отправленного входа в систему, (Password\_Again) поле повторного ввода пароля для проверки.

Рис. 10. Окно регистрации

Окно авторизации, для входа в приложении. Поле (Email) – используется как логин для входа в информационную систему. Поле (Password) – поле ввода пароля для проверки пользователя. Реализован также функционал показать пароль, это флажок, который можно переключить. По нажатию кнопки войти происходит проверка данных, если проверка прошла успешно будет открыто окно с основным функционалом информационной системы.

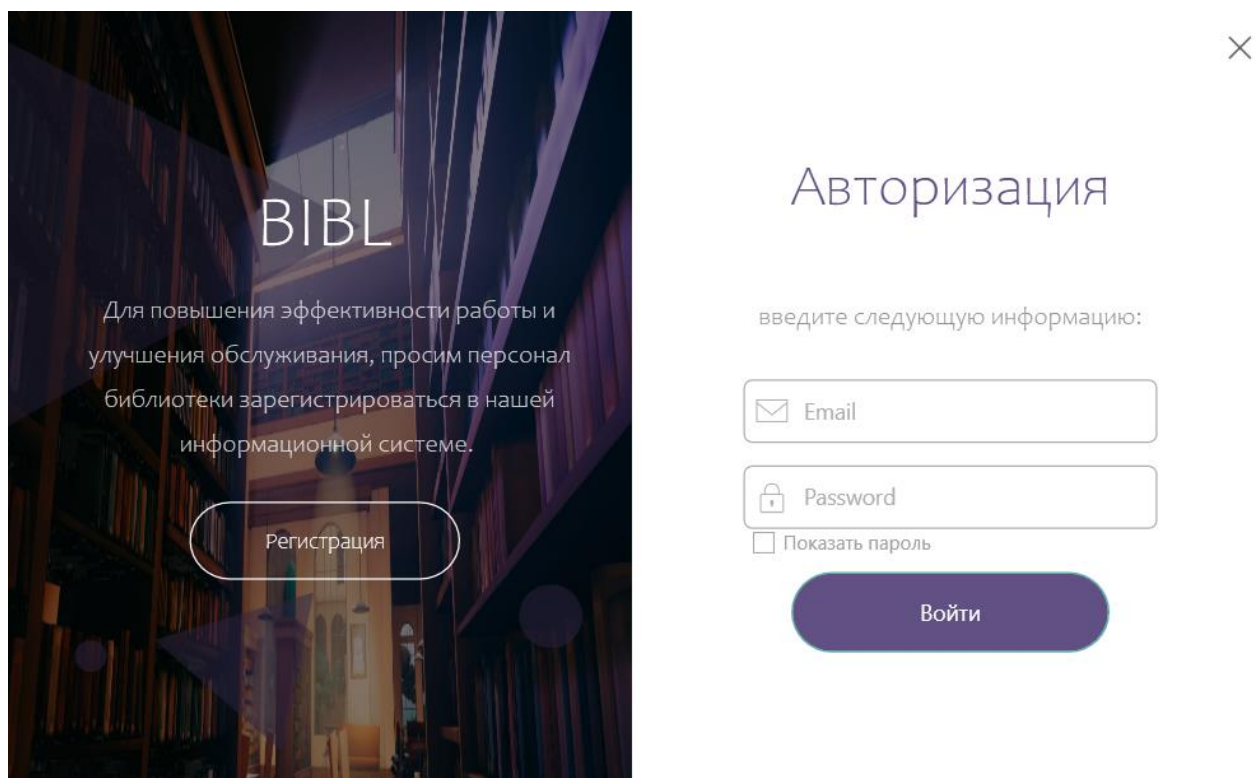


Рис. 11. Окно авторизации пользователя в ИС

В page (Клиенты) – находится таблица с читателями, в данной таблице выведены такие данные как ФИО читателя, а также их почта. В дальнейшем на почту клиента будут приходить уведомительные письма. Также в этом окне реализован функционал добавление, редактирование, удаление клиентов.

Меню слева является проводником между другими page, в меню есть переходы между page книги и аренда, также есть функция «Выход» возврата в окно авторизации.

Bibl	Клиенты		Добавить	Редактировать	Удалить
	ФИО	Email			
	Шуметов Максим Сергеевич	shumetov2004@gmail.com			
	Кузнецов Артём Павлович	weffaupruseulla-6944@yopmail.com			
	Родин Олег Альбертович	OlegSportKrutoy@mail.ru			
	Агафонов Иван Маркович	pihaquagrussoi-3346@yopmail.com			
	Дорохова Мария Марковна	tarin_orora9@inbox.ru			
	Дорохова Стефания Марковна	punopez_eso11@aol.com			
	Александрова Яна Львовна	Yana.666@gmail.com			
	Черчел Анна Алексеевна	cherchel_2019@mail.ru			
	Артамонова Полина Никитична	PolinA234@mail.ru			
	Скворцова Ульяна Павловна	fig-ulubigu44@gmail.com			
	Демедьева Лолита Петровна	Joos34@mail.ru			
	Кизиков Илья Вальеривич	bigpapail23@gmail.com			
	Бухаров Юрий Олегович	yuraboh554@gmail.com			
	Рудской Иван Владимирович	IvanGuy444@mail.ru			
	Савченко Дмитрий Максимович	Dimon666@mail.ru			
	Павленко Раиса Нурдиновна	RaisaNur224@gmail.com			
	Крощин Василий Петрович	smalfegrl@yandex.ru			
	Определнов Денис Фанильевич	deniska@gmail.com			
	Петров Михаил Петрович	petrov56@mail.ru			
	Воронин Константин Николаевич	lusa228@mail.ru			
	Воронина Вера Геннадьевна	VeraKruto@gmail.com			
	Баринов Виктор Петрович	SheffPovar@mail.ru			
	Лавров Максим Вальеревич	Oguzok@Mail.ru			
	Смирнова Елена Геннадьевна	elena.smirnova@mail.com			
	Козлова Наталья Сергеевна	NU natalya.kuznetsova@email.ru			

Рис. 12. Page клиенты

Bibl	Книги		Роман	Жанры	Выбрать	Добавить	Редактировать	Удалить
	Название	Автор	Жанр	Год	Количество	Доступность		
	"Лолита"	Владимир Набоков	Роман	2001	2	Доступно		
	"Улисс"	Джеймс Джойс	Роман	1992	1	Доступно		
	"Война и Мир"	Толстой Лев Николае	Роман	1812	2	Доступно		
	"Звук и ярость"	Уильям Фолкнер	Роман	1998	0	Не доступно		
	"Распад"	Чинуа Ачебе	Роман	1934	1	Доступно		
	"Уловка-22"	Джозеф Хеллер	Роман	1999	1	Доступно		
	"Возлюбленная"	Тони Моррисон	Роман	2005	3	Доступно		
	"Песнь Соломона"	Тони Моррисон	Роман	1993	2	Доступно		
	"Золотая тетрадь"	Дорис Лессинг	Роман	2006	2	Доступно		
	"Вечный сон"	Реймонд Чандлер	Роман	1992	0	Не доступно		
	"Когда я умирала"	Уильям Фолкнер	Роман	1949	2	Доступно		
	"Я, Клавдий"	Роберт Грейвс	Роман	1999	2	Доступно		
	"Ночь"	Эли Визель	Роман	2001	2	Доступно		
	"Один прекрасный веч	Потапова Лилия Влад	Роман	2009	2	Доступно		
	"Весна в пионерском	Малисова Елена Яков	Роман	2021	2	Доступно		
	"Моя прекрасная Жан	Джордж Вилстон	Роман	2016	2	Доступно		
	"Поющие в терновни	Колин Маккалоу	Роман	2003	2	Доступно		
	"Ангелы и демоны"	Дэн Браун	Роман	2000	1	Доступно		
	"Дневник Бриджит Д	Хелен Филдинг	Роман	1996	1	Доступно		
	"Любовь во время хо	Габриэль Гарсиа Мар	Роман	1985	2	Доступно		
	"Скотный двор"	Джордж Оруэлл	Роман	1945	2	Доступно		

Рис. 13. Page книги

В page (книги) реализован функционал добавления, редактирования и удаления литературы. Все данные выводятся в виде таблиц.

Таблицы данных — это структурированный способ представления информации в виде таблицы, состоящей из строк и столбцов. Каждая строка таблицы обычно представляет собой одну запись или элемент данных, а каждый столбец содержит определенные атрибуты или значения для этих записей.

Таблицы данных используются для организации и представления больших объемов информации в удобной форме.

Преимущества таблиц данных:

1. Структурированность: Таблицы позволяют структурировать данные для удобного анализа и просмотра.
2. Сравнение данных: благодаря таблицам легко сравнивать данные и выявлять паттерны или тренды.
3. Фильтрация и сортировка: Пользователи могут легко фильтровать и сортировать данные в таблицах для получения необходимой информации.

Есть возможность отсортировать данные по жанрам с помощью (combo\_box), также как и любые данные, в приложение реализовано добавление жанров, редактирование, удаление.

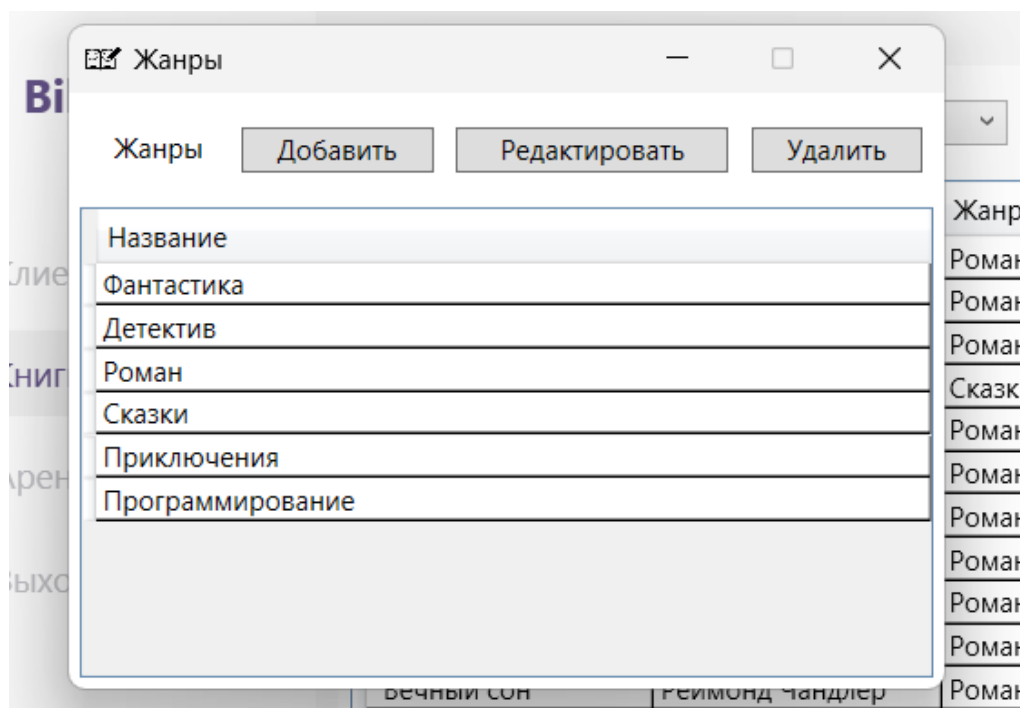


Рис. 14. Жанры

Кнопка выбрать позволяет выбрать конкретную книгу и добавить её уже зарегистрированному пользователю (читателю).



ФИО клиента	Название книги	Дата возврата	Статус
Шуметов Максим Сергеевич	"Улисс"	10.05.2024	Вернул
Шуметов Максим Сергеевич	"Война и Мир"	29.06.2024	Не вернул
Шуметов Максим Сергеевич	"Война и Мир"	01.06.2024	Вернул
Кузнецов Артём Павлович	"Лолита"	06.07.2024	Не вернул
Черчел Анна Алексеевна	"Уловка-22"	07.06.2024	Просрочил
Агафонов Иван Маркович	"Звук и ярость"	09.06.2024	Не вернул
Родин Олег Альбертович	"Улисс"	16.06.2024	Не вернул
Дорохова Стефания Марковна	"Вечный сон"	13.06.2024	Не вернул
Кизиков Илья Вальеревич	"Песнь Соломона"	08.06.2024	Не вернул
Савченко Дмитрий Максимович	"Песнь Соломона"	15.06.2024	Не вернул
Лавров Максим Вальеревич	"Когда я умирала"	12.06.2024	Не вернул
Демедьева Лолита Петровна	"Ночь"	09.06.2024	Не вернул
Дорохова Стефания Марковна	"Поющие в терновнике"	11.06.2024	Вернул
Павленко Раиса Нурдиновна	"Скотный двор"	03.06.2024	Просрочил
Павленко Раиса Нурдиновна	"Поющие в терновнике"	15.06.2024	Не вернул

Рис. 15. Page аренда

Page Аренда книг – это функционал, который позволяет вести учёт и оповещение пользователей (читателей) по взятым ими книгам, у каждого пользователя есть дата возврата книги, за 3 дня до этого момента приложение отправляет клиенту оповещение, на ранее им указанную почту. После возврата книги в библиотеку статус клиента меняется.

Аренда книг

Добавление новой аренды

ФИО клиента

Определнов Денис Фанильевич

Вернуть

15.06.2024

Добавить

Рис. 16. Добавление аренды

Отчёты играют важную роль в различных аспектах. Отчёты позволяют отслеживать ключевые показатели производства, анализировать данные и делать выводы на основе фактов.

Отчёт по аренде книг

Всего записей: 15   Вернули: 3   Не вернули: 12

Список аренд

ФИО клиента	Взятая книга	Дата возврата	Статус
Шуметов Максим Сергеевич	"Улисс"	10.05.2024	Вернул
Шуметов Максим Сергеевич	"Война и Мир"	29.06.2024	Не вернул
Шуметов Максим Сергеевич	"Война и Мир"	01.06.2024	Вернул
Кузнецов Артём Павлович	"Лолита"	06.07.2024	Не вернул
Черчел Анна Алексеевна	"Уловка-22"	07.06.2024	Не вернул
Агафонов Иван Маркович	"Звук и ярость"	09.06.2024	Не вернул
Родин Олег Альбертович	"Улисс"	16.06.2024	Не вернул
Дорохова Стефания Марковна	"Вечный сон"	13.06.2024	Не вернул
Кизиков Илья Валерьевич	"Песнь Соломона"	08.06.2024	Не вернул
Савченко Дмитрий Максимович	"Песнь Соломона"	15.06.2024	Не вернул
Лавров Максим Валерьевич	"Когда я умираю"	12.06.2024	Не вернул
Демидова Лолита Петровна	"Ночь"	09.06.2024	Не вернул
Дорохова Стефания Марковна	"Поющие в терновнике"	11.06.2024	Вернул
Павленко Раиса Нуридиновна	"Скотный двор"	03.06.2024	Не вернул
Павленко Раиса Нуридиновна	"Поющие в терновнике"	15.06.2024	Не вернул

Рис. 17. Отчёт

В целом, отчёты являются важным инструментом для оценки производительности, контроля и принятия решений в бизнесе. Они помогают

обеспечить эффективное управление компанией и достижение стратегических целей. Это помогает руководству и сотрудникам принимать важные решения.

Реализован функционал вывода отчёта, за весь период, для контроля и учёта книг в библиотеки.

Page администратора системы, в page реализован функционал добавление новых пользователей информационной системы, а также их редактирование и удаление. У каждого пользователя есть своя роль в системе.

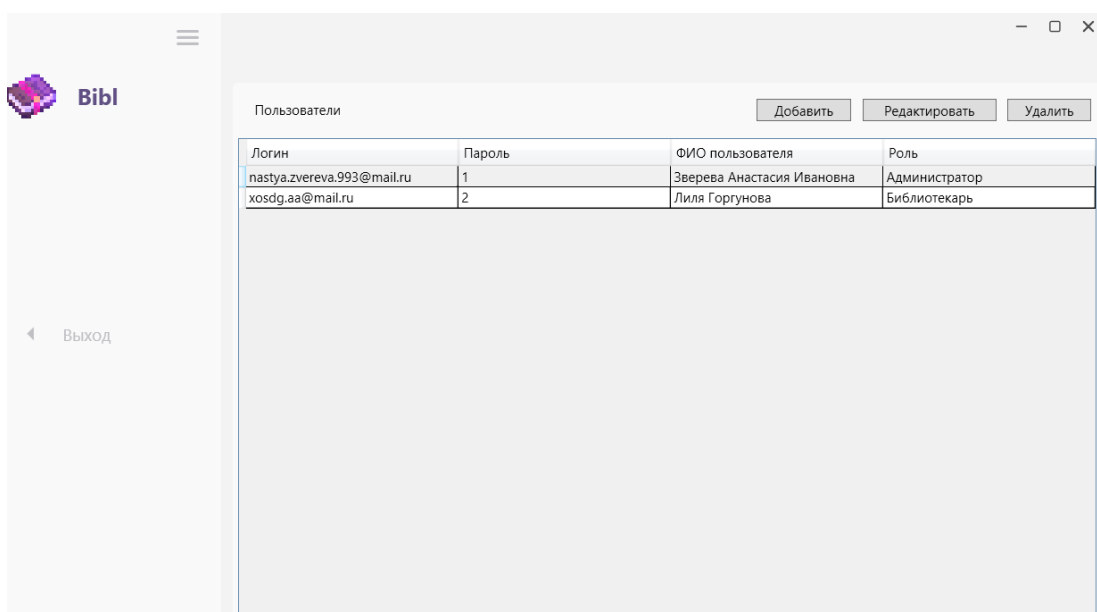


Рис. 18. Page Администратора

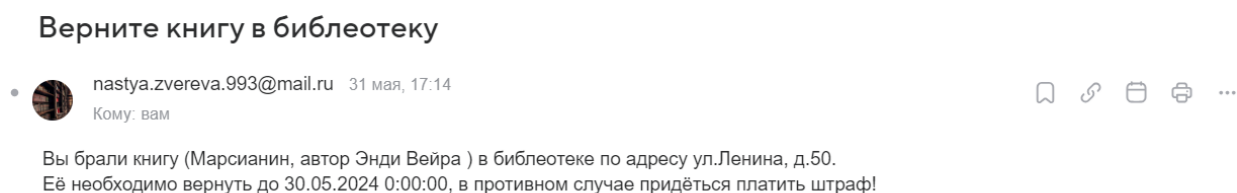


Рис. 19. Оповещение на почту

Оповещения по электронной почте позволяют оперативно информировать читателей о важных изменениях, новостях и других важных аспектах деятельности компании.

Оповещения на почту могут быть использованы для информирования клиентов о заказах, статусе обслуживания.

В информационной системе библиотека реализовано оповещение пользователей о скорой дате возврата книги, в оповещение входит:

Название книги, адрес библиотеки в который была взята книга, дата возврата до которой нужно вернуть книгу, и уведомление о штрафе за задержку возврата.

## ЗАКЛЮЧЕНИЕ

Информационные системы являются актуальными на сегодняшний день - одна из наиболее перспективных и быстроразвивающихся отраслей, по всему миру. С каждым годом растет количество информационных систем, как в нашей стране, так и за рубежом.

В ходе выполнения дипломной работы была разработана автоматизированная информационная системы для библиотеки. Были спроектированы диаграммы: Entity-Relationship Diagram и Use Case. Была разработана база данных, а также разработан интерфейс приложения и произведено программирование информационной системы. Данная информационная система была разработана для работников библиотеки.

Целью работы было создание системы, способной эффективно обрабатывать информацию о литературе, упрощать процедуры аренды и учёта книг, повышать качество работы библиотеки. В результате выполнения поставленных задач была разработана информационная система, способствующая оптимизации процесса библиотеки.

Актуальность данной работы заключается в необходимости совершенствования процессов работы библиотеки. Разработка информационной системы представляет собой важный шаг в направлении автоматизации процессов.

Поставленные задачи были выполнены успешно. Разработанная информационная система позволяет обрабатывать большие объемы информации о поступающей литературе, что способствует быстрому формированию посетителей и ведению учёта литературы. Система также упрощает оповещения пользователей сокращает временные затраты на обработку документов и повышает качество работы библиотеки.

Таким образом, разработка автоматизированной информационной системы для библиотеки имеет высокую актуальность, обладает значительным потенциалом, способствует современным подходам к автоматизации процессов в данной сфере.

Ссылка на проект в GitHub: <https://github.com/Anastasya33/Biblioteka-Diplom.git>

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Смит, Дж. Программирование на C#: основы и практика / Дж. Смит. - М.: Издательство Техносфера, 2020. – 320 с.
2. Meta nit Изучение C# // Язык C# и платформа .NET [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/tutorial/1.1.php>
3. Браун, К. Программирование на C# 7 и платформа .NET Core: современное руководство / К. Браун. - Киев: Издательский дом ДМК Пресс, 2021. – 400 с.
4. Уолш, М. SQL для профессионалов: полное руководство по языку запросов SQL / М. Уолш. - М.: Вильямс, 2020. – 480 с.б. Ошероув Р. Искусство автономного тестирования с примерами на C#: учебное пособие/ Рой Ошероув, 2022. – 362 с.
7. Рафалович В. Data mining, или Интеллектуальный анализ данных для занятых. Практический курс: учебное пособие/ Владимир Рафалович, 2020. – 110 с.
8. Рихтер Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#: учебное пособие/ Джеффри Рихтер, 2022. – 893 с.
9. Стиллман Э. Грин Д. Изучаем C#: учебное пособие/ Эндрю Стиллман, Дженнифер Грин, 2021. – 769 с.
10. Ульстайн С. Шепот питона: учебное пособие/ Силье Ульстайн, 2020. – 350 с.
11. Хортон, Э. Начальное программирование в C# 7.0 для чайников / Э. Хортон. - М.: Вильямс, 2022. – 816 с.
12. Уолтерс, Д. SQL для аналитиков: справочник по языку SQL в Microsoft Excel / Д. Уолтерс. - СПб: БХВ-Петербург, 2021. – 304 с.

13. Тепляков, А. Основы программирования на C# и платформе .NET Framework / А. Тепляков. - М.: ДМК Пресс, 2020. – 352 с.
14. Литвин, Ю. SQL и реляционные базы данных: учебный курс / Ю. Литвин. - Киев: КУБ, 2020. – 464 с.
15. Meta nit Изучение C# // Язык C# и платформа .NET [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/tutorial/1.1.php>
16. Макбрайд, Д. SQL и реляционные базы данных: практическое руководство / Д. Макбрайд. - СПб: Питер, 2021. – 352 с.
17. Geek Brains Изучение C# // ИТ-инженер. Передовые технологии [Электронный ресурс] – Режим доступа: [https://gb.ru/geek\\_university/engineer/main](https://gb.ru/geek_university/engineer/main)
18. Стэнли, Д. SQL: язык структурированных запросов / Д. Стэнли. - Киев: БХВ-Петербург, 2022. – 2023 с.
19. Албахари, Б. C# 7.0 in a Nutshell: The Definitive Reference / Б. Албахари. - Sebastopol: O'Reilly Media, 2022. – 1136 с.
20. Робинсон, Дж. SQL: Полное руководство / Дж. Робинсон. - М.: ДМК Пресс, 2021. – 704 с.