

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БАШКИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

БИРСКИЙ ФИЛИАЛ БАШГУ
ФАКУЛЬТЕТ ФИЗИКИ И МАТЕМАТИКИ
КАФЕДРА ИНФОРМАТИКИ И ЭКОНОМИКИ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ПО ПРОГРАММЕ БАКАЛАВРИАТА

НУРМУХАМЕТОВ ВАДИМ РАДИКОВИЧ

СЕРВИС ОНЛАЙН-БРОНИРОВАНИЯ КНИГ В БИБЛИОТЕКЕ БИРСКОГО
ФИЛИАЛА БАШКИРСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

Выполнил:
Студент 4 курса очной формы обучения
Направление подготовки
09.03.03 Прикладная информатика
Направленность (профиль)
Прикладная информатика в
информационной сфере

Руководитель:
к.ф.-м.н., доцент

А.А. Русинов

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	5
1.1. Структура организации работы библиотеки Бирского филиала Башкирского государственного университета.....	5
1.2. Исследование существующей системы работы библиотеки на диаграммах IDEF0 и IDEF3	5
1.3. Обзор программных средств реализации информационной системы и обоснование их выбора.....	9
Выводы по первой главе	12
ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	13
2.1. Разработка структурно-функциональной диаграммы работы библиотеки “Как должно быть”	13
2.2. Функционал системы на диаграмме прецедентов.....	16
2.3. Инфологическая модель данных.....	18
2.4. Даталогическая модель данных	18
Выводы по второй главе	22
ГЛАВА 3. РЕАЛИЗАЦИЯ КОМПОНЕНТОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	23
3.1. Программная реализация интерфейса.....	23
3.2. Реализация базы данных.....	27
3.3. Руководство пользователя.....	30
Выводы по третьей главе	42
ЗАКЛЮЧЕНИЕ.....	43
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ	44

ВВЕДЕНИЕ

В наше быстроизменяющееся время, когда всё чаще и чаще появляются новые и более продвинутые технологии, многие организации стараются их использовать. Работа с последними технологиями привлекает особенно молодое поколение, ведь они уже привыкли использовать Интернет во всём: заказ еды, покупка мебели, техники, одежды, поиск информации и, конечно же, общение. Каждый год по нарастающей идёт развитие услуг через Интернет. Люди всё чаще прибегают к приобретению онлайн услуг, включая предоставление доступа к книгам. Создание возможности удалённой работы библиотеки является также актуальным в современном мире. В таком случае она будет иметь возможность выдавать книги без необходимости в личном контакте с библиотекарем. Такая система также упростит работу библиотекарям, давая более удобные способы проверки наличия книги в библиотеке и поиска информации, кому выдана книга.

Объект исследования: информационный процесс выдачи книг в библиотеке Бирского филиала Башкирского государственного университета.

Предмет исследования: автоматизация информационного процесса выдачи книг посредством их бронирования на сайте библиотеки Бирского филиала Башкирского государственного университета.

Целью выпускной квалификационной работы является создание сервиса онлайн-бронирования книг в библиотеке Бирского филиала Башкирского государственного университета путём реализации информационной системы, автоматизирующей процесс выдачи книг.

В соответствии с целью поставлены следующие **задачи**:

1. Анализ предметной области.
2. Проектирование информационной системы библиотеки Бирского филиала Башкирского государственного университета.

3. Разработка информационной системы библиотеки Бирского филиала Башкирского государственного университета.

Практическая значимость: возможность применения данной системы для учёта и выдачи книг в библиотеке университета.

Выпускная квалификационная работа состоит из введения, трёх глав, заключения и списка используемых источников и литературы. В первой главе выполнен анализ предметной области и рассмотрены теоретические основы используемых технологий. Во второй главе спроектированы основные информационные-процессы разрабатываемой информационной системы. В третьей главе отображён процесс разработки и описание функциональных возможностей. Заключение содержит краткие положения по результатам выполнения работы и рекомендации по использованию полученных результатов.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Структура организации работы библиотеки Бирского филиала Башкирского государственного университета

Библиотека Бирского филиала Башкирского государственного университета создана на базе библиотеки Башкирского педагогического техникума в 1939 году. В настоящее время фонд библиотеки составляет 446 тысяч экземпляров изданий: книги, брошюры, периодические издания, издания на электронных носителях, иллюстрации. Библиотека ежегодно выписывает около 100 наименований журналов и газет. Состоит из трёх отделов: информационно-библиографического, комплектования и обработки литературы, обслуживания и книгохранения. Руководит библиотекой Лихачева Елена Александровна [33].

Интересом исследования является выдача книг и их учёт библиотекой. Положительным результатом будет считаться увеличение скорости работы библиотеки, увеличение количества посетителей и упрощение работы её сотрудников.

1.2. Исследование существующей системы работы библиотеки на диаграммах IDEF0 и IDEF3

Предназначение нотации IDEF0 заключается в формализации и описании информационных-процессов [4,18]. С использованием средств данной нотации построена контекстная диаграмма (Рисунок 1.1), направленная на отражение процесса деятельности библиотеки с точки зрения библиотекаря.

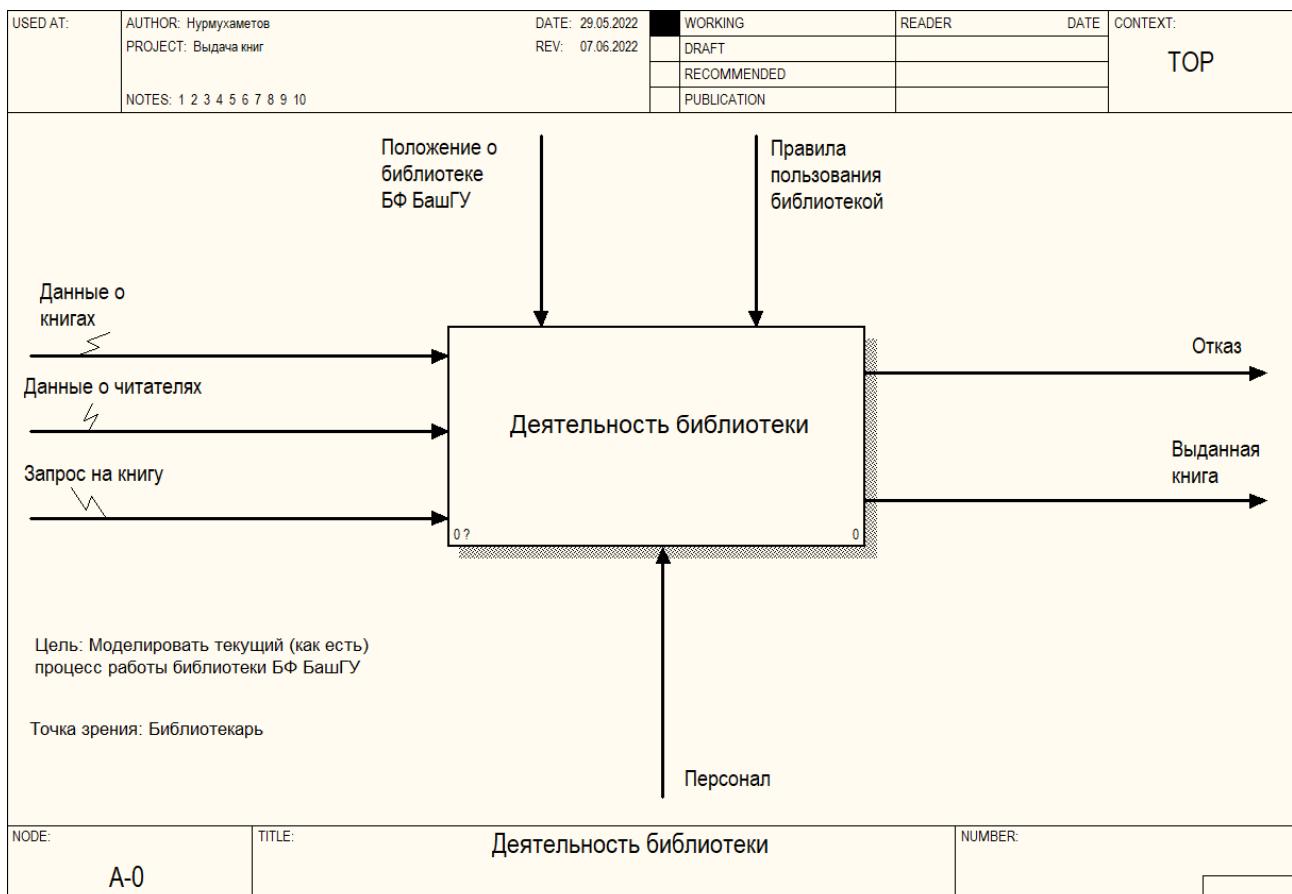


Рисунок 1.1 Контекстная диаграмма в нотации IDEF0 по типу “Как есть”

На показанной контекстной диаграмме можно выявить следующие виды стрелок:

- Вход – получаемые в библиотеку книги, данные о читателях, записанных в библиотеке и запросы на выдачу книги. Указанные данные представляют собой объекты, которые при работе процесса будут преобразованы или использованы для получения результатов (выход).
- Выход – выданная книга читателю на руки, с записанной библиотекарем информацией о том, кому и какая книга выдана, либо отказ, в случае отсутствия книги в библиотеке.
- Управление – положение о библиотеке БФ БашГУ, правила пользования библиотекой. Требования, необходимые для управления действиями происходящей работы.
- Механизм – персонал библиотеки. Ресурсы, которые будут выполнять работу.

В настоящее время выдача книг в библиотеке происходит следующим образом: новые поступающие книги расставляются на полки по тематике и алфавиту, заранее проводиться запись общей информации о книгах. Студенты с читательским билетом делают запрос на выдачу книги, библиотекарь находит нужную книгу, записывает кому она выдана и когда, либо сообщают о её отсутствии [15].

Расписанный процесс можно представить следующей декомпозицией контекстной диаграммы, в которой выявлены следующие функциональные блоки (Рисунок 1.2):

- Ведение каталога.
- Поиск.
- Оформление выдачи.

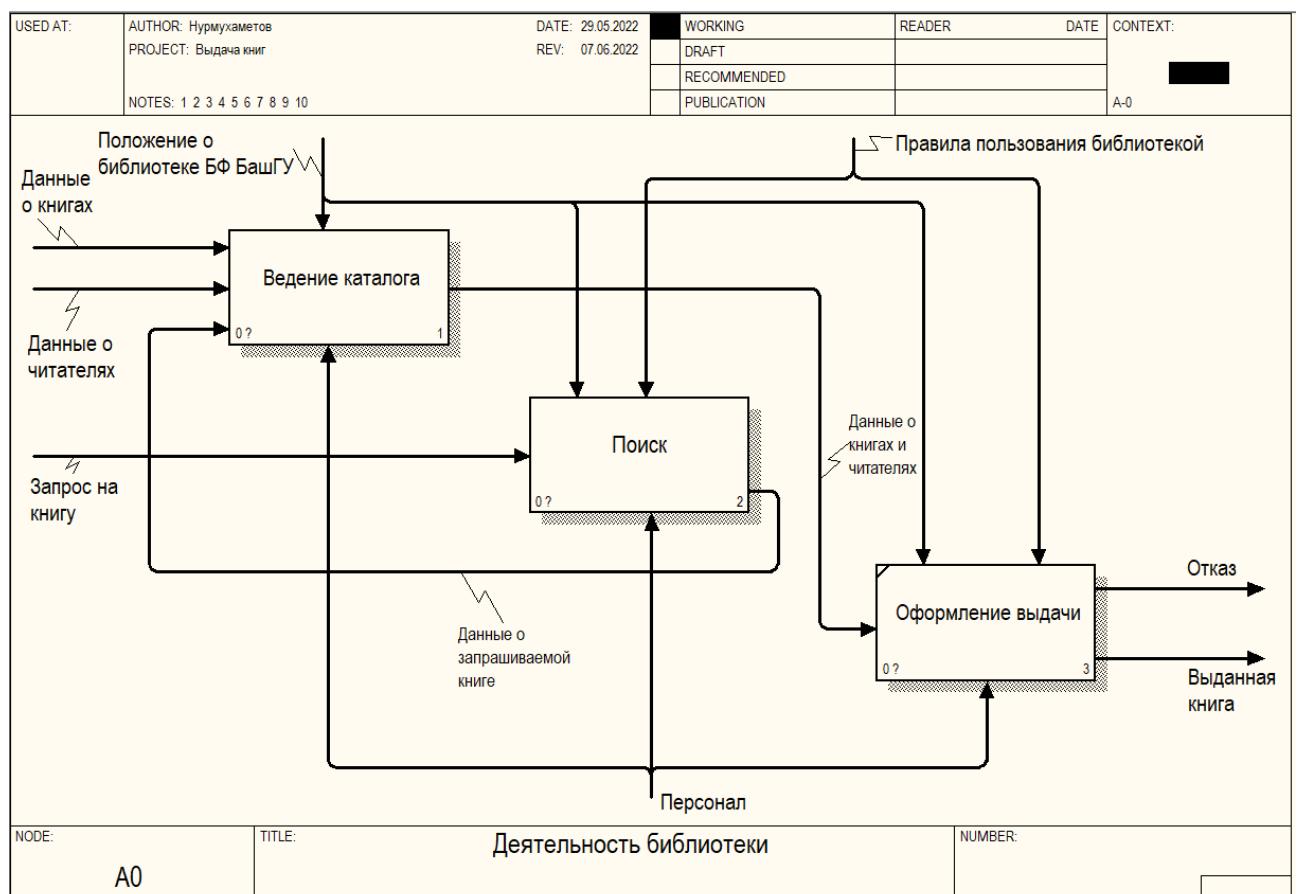


Рисунок 1.2 Результат декомпозиции контекстной диаграммы

Данная декомпозиция (Рисунок 1.3) представляет собой иллюстрацию ведения каталога и поиска необходимых книг в библиотеке и данных записанных читателей.

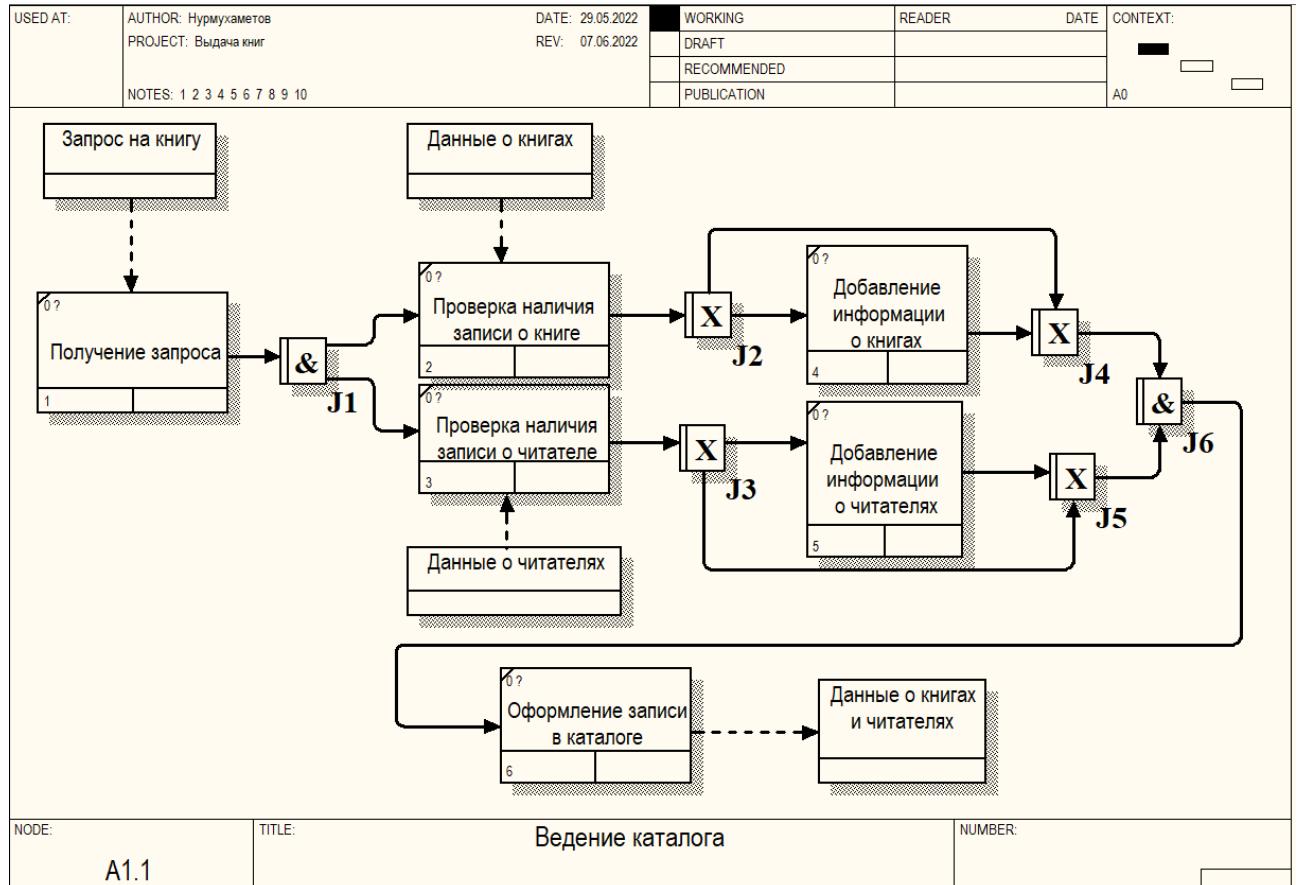


Рисунок 1.3 Декомпозиция блока “Ведение каталога” в нотации IDEF3

В ходе создания структурно-функциональной диаграммы по типу “Как есть” найдены следующие недостатки:

- Неудобство в работе с поиском запрашиваемых книг.
- Устаревание существующей системы работы библиотеки.
- Лишние затраты рабочего времени библиотекарей, как при получении новых книг, так и при выдаче имеющихся.

1.3. Обзор программных средств реализации информационной системы и обоснование их выбора

Среда разработки. В современное время рынок переполнен различными средами для разработки программного обеспечения, в том числе и веб-приложений. Популярной является интегрированная среда разработки программного обеспечения Microsoft Visual Studio с поддержкой удобного и многим известного языка программирования C#, на котором и будет написана большая часть будущей системы [2,5-14,20]. Данная среда включает в себя большой список полезных и нужных инструментов для создания планируемой системы. В частности, использованы следующие инструменты разработки:

– ASP.NET Core – кроссплатформенная, высокопроизводительная среда с открытым исходным кодом для создания современных облачных приложений, подключенных к Интернету [31,35]. В сравнении с предыдущей версией фреймворка ASP.NET, имеет большую производительность. Страницы системы написаны на языке разметки страниц HTML с использованием языка таблиц стилей CSS, которые также поддерживаются средой разработки. Также за счёт поддержки HTML даётся возможность писать некоторую логику прямо на страницах системы за счёт поддержки большинством браузеров скриптов, написанных на мультипарадигменном языке программирования JavaScript.

Выбор сделан также с учётом развития выбранной среды. Согласно исследованиям [40], выбранная платформа имеет наибольшую производительность и тенденцию развития, на том же оборудовании – как показано ниже на графике (Рисунок 1.4):

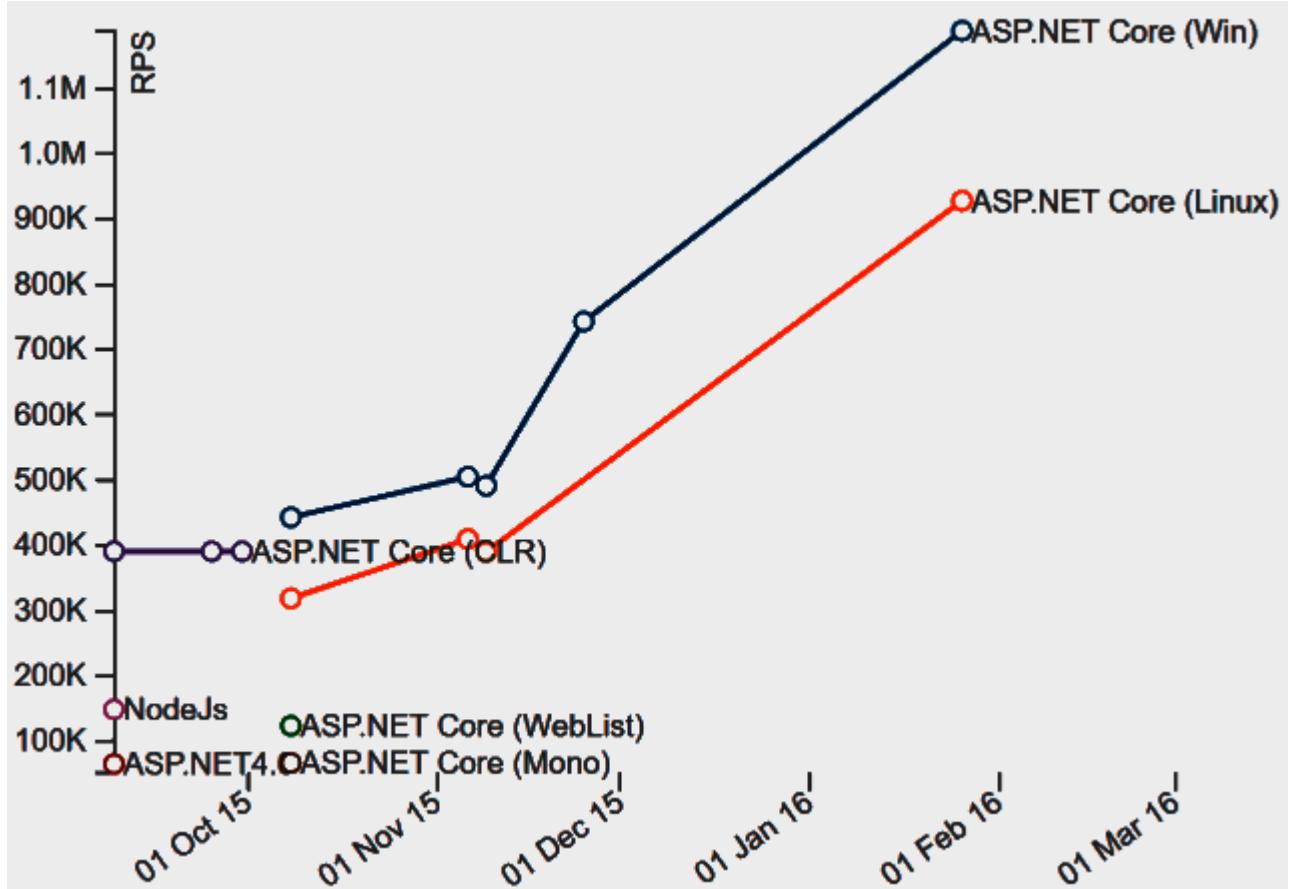


Рисунок 1.4 График сравнения количества запросов в секунду в различных средах

- Entity Framework Core – объектно-ориентированная, легковесная и расширяемая технология от компании Microsoft для доступа к данным [29]. Предоставляет возможность работы с данными при помощи удобных и понятных запросов на языке LINQ. Позволяет строить многоуровневые системы, реализуя шаблон проектирования MVC.

Технология Entity Framework Core обеспечивает более удобную связь между таблицами, создавая их автоматически [21]. Данное утверждение описано различными авторами [34], сравнивая с другими технологиями (Рисунок 1.5).

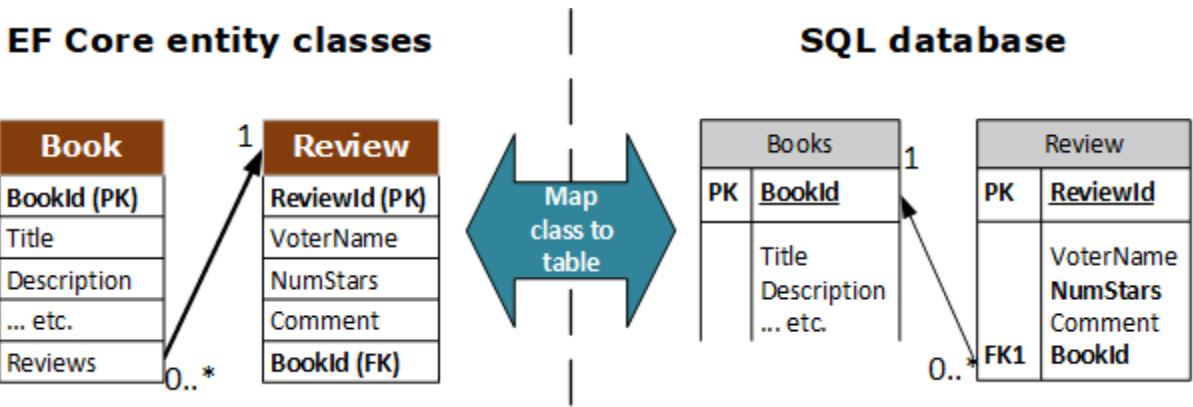


Рисунок 1.5 Сравнение связи между таблицами в отличных друг от друга

ТЕХНОЛОГИЯХ

– MySQL – одна из самых популярных СУБД для веб-приложений [37]. Является стандартом для веб-серверов. MySQL – это бесплатный пакет программ, однако новые версии выходят постоянно, расширяя функционал и улучшая безопасность [39].

Из сравнений (Рисунок 1.6), описанных в различных статьях, видно, что СУБД MySQL является наиболее подходящим для разработки Web-сайтов [32].

Name	Advantages	Disadvantages	When to use	When not to use
SQLite	<ul style="list-style-type: none"> • File based • Standards-aware • Great for developing and even testing 	<ul style="list-style-type: none"> • No user management • Lack of possibility to tinker with for additional performance 	<ul style="list-style-type: none"> • Embedded applications • Disk access replacement • Testing 	<ul style="list-style-type: none"> • Multi-user applications • Applications requiring high write volumes
MySQL	<ul style="list-style-type: none"> • Easy to work with • Feature rich • Secure • Scalable and powerful • Speedy 	<ul style="list-style-type: none"> • Known limitations • Reliability issues • Stagnated development 	<ul style="list-style-type: none"> • Distributed operations • High Security • Web-sites and Web-applications • Custom solutions 	<ul style="list-style-type: none"> • SQL compliance • Concurrency • Lack of features
PostgreSQL	<ul style="list-style-type: none"> • An open source SQL standard compliant RDBMS • Strong Community • Strong third-party support • Extensible • Objective 	<ul style="list-style-type: none"> • Performance • Popularity • Hosting 	<ul style="list-style-type: none"> • Data Integrity • Complex, custom procedures • Integration • Complex designs 	<ul style="list-style-type: none"> • Speed • Simple to set up • Replication

Рисунок 1.6 Сравнение современных СУБД

- LINQ (Language-Integrated Query) – простой и удобный язык запросов к источнику данных, разработанный корпорацией Microsoft, написанный на уже известном многим языке C# [36]. Предлагает программисту большой спектр запросов на получение и сортировку данных.

Выводы по первой главе

Произведён анализ предметной области работы библиотеки Бирского филиала Башкирского государственного университета. Выполнено моделирование текущего состояния (как есть) с использованием нотаций IDEF0 и IDEF3. Показано, что для улучшения работы библиотеки необходимо создание онлайн-сервиса.

Проведён обзор программных средств, предоставляющих возможность создания онлайн-сервиса. Для разработки выбрана интегрированная среда Microsoft Visual Studio (язык программирования C#).

ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1. Разработка структурно-функциональной диаграммы работы библиотеки “Как должно быть”

В ходе разработки информационной системы значительно переработана описанная ранее диаграмма “Как есть” и в результате получена структурно функциональная диаграмма “Как должно быть” (Рисунок 2.1) [27,28].

Входные данные и управление не изменены. К механизмам добавлена информационная система, участвующая в большинстве функций. А на выход добавлены выдача PDF версии книги и отказ.

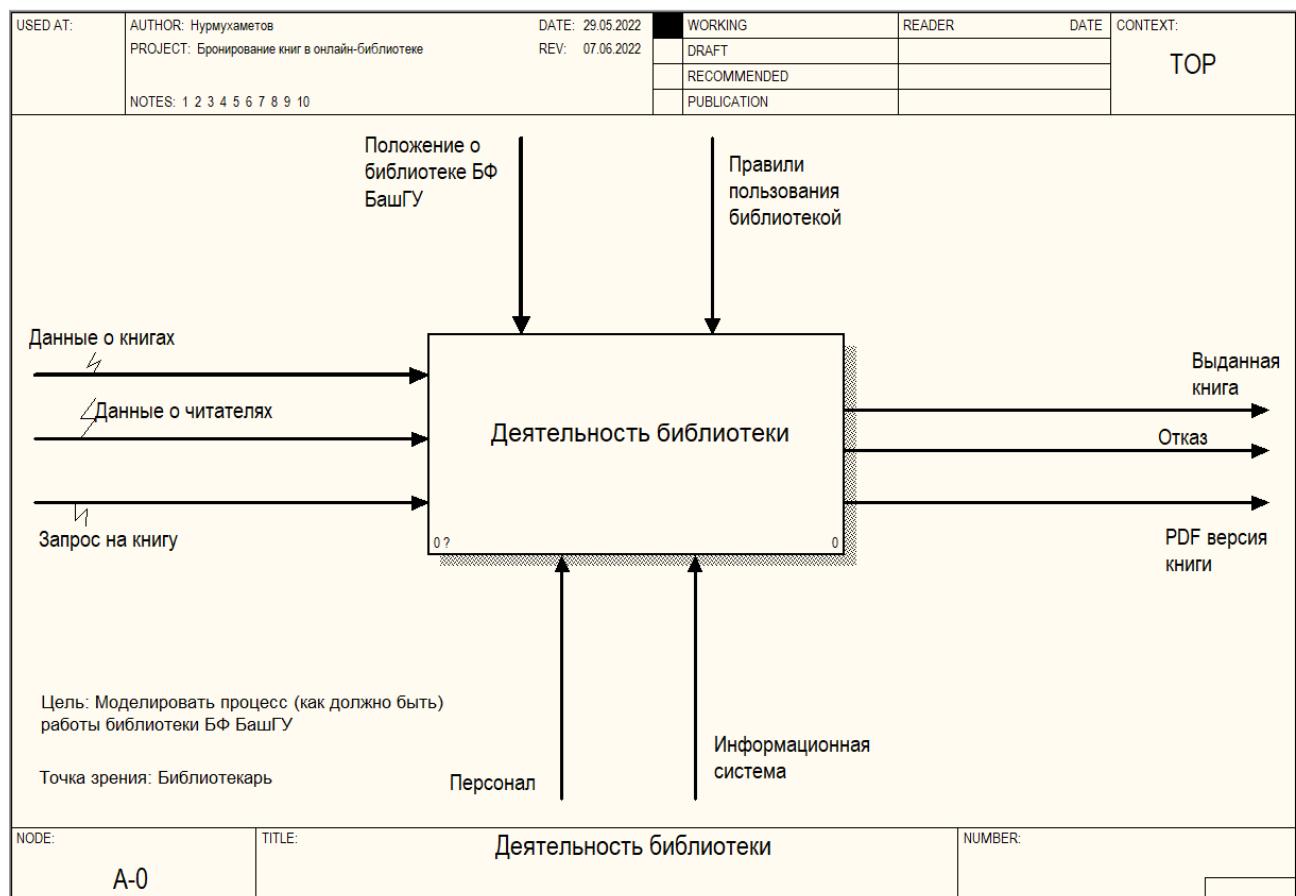


Рисунок 2.1 Контекстная диаграмма в нотации IDEF0 по типу “Как должно быть”

Декомпозиция контекстной диаграммы (Рисунок 2.2) включает в себя следующие функциональные блоки:

- Регистрация пользователей.
- Ведение каталога.
- Поиск.
- Оформление информации о выданной книге.

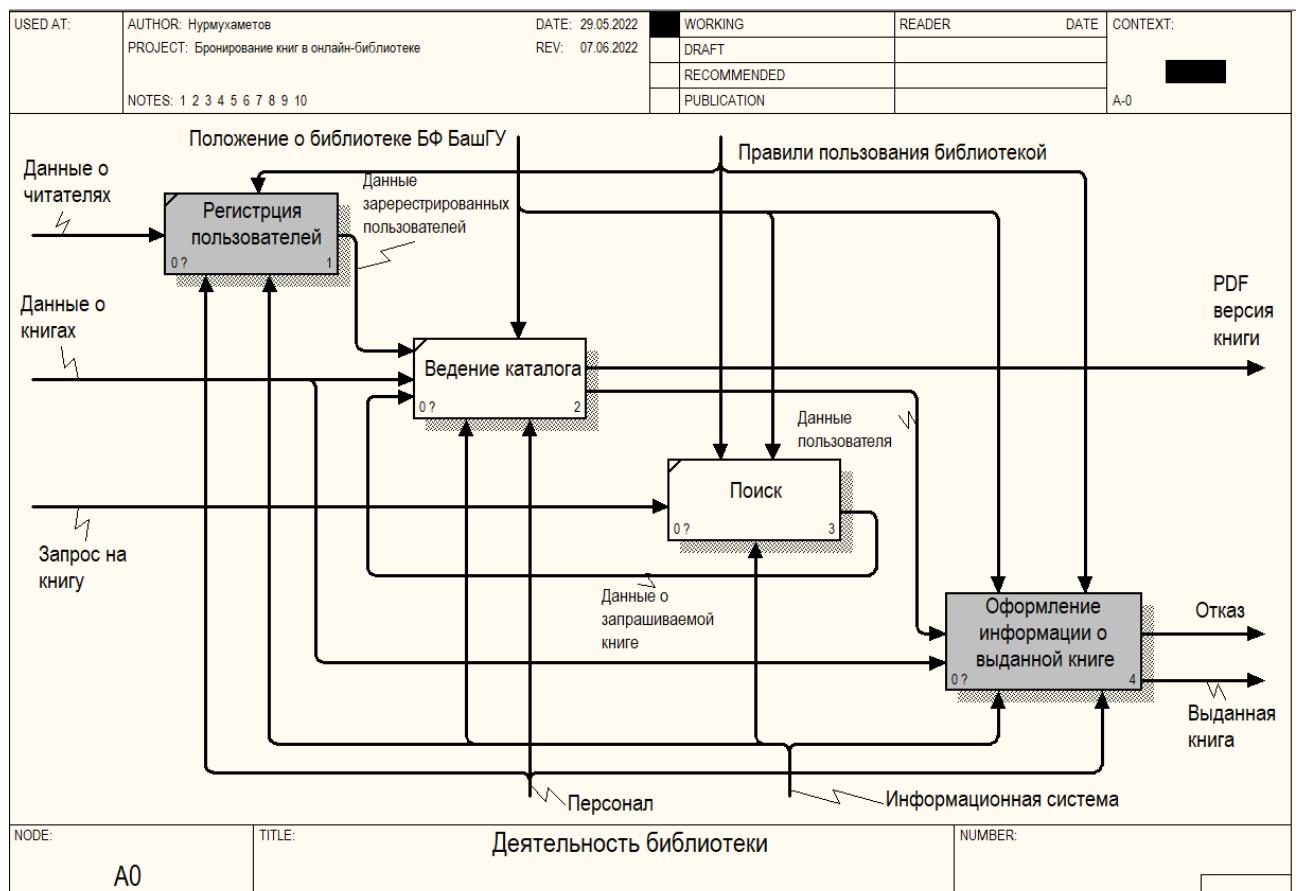


Рисунок 2.2 Результат декомпозиции контекстной диаграммы “Как должно быть”

Декомпозиция, показанная ниже (Рисунок 2.3), описывает то, как будет происходить процесс оформления информации о книге на выдачу. Данная диаграмма построена с использованием нотации IDEF3 [22].

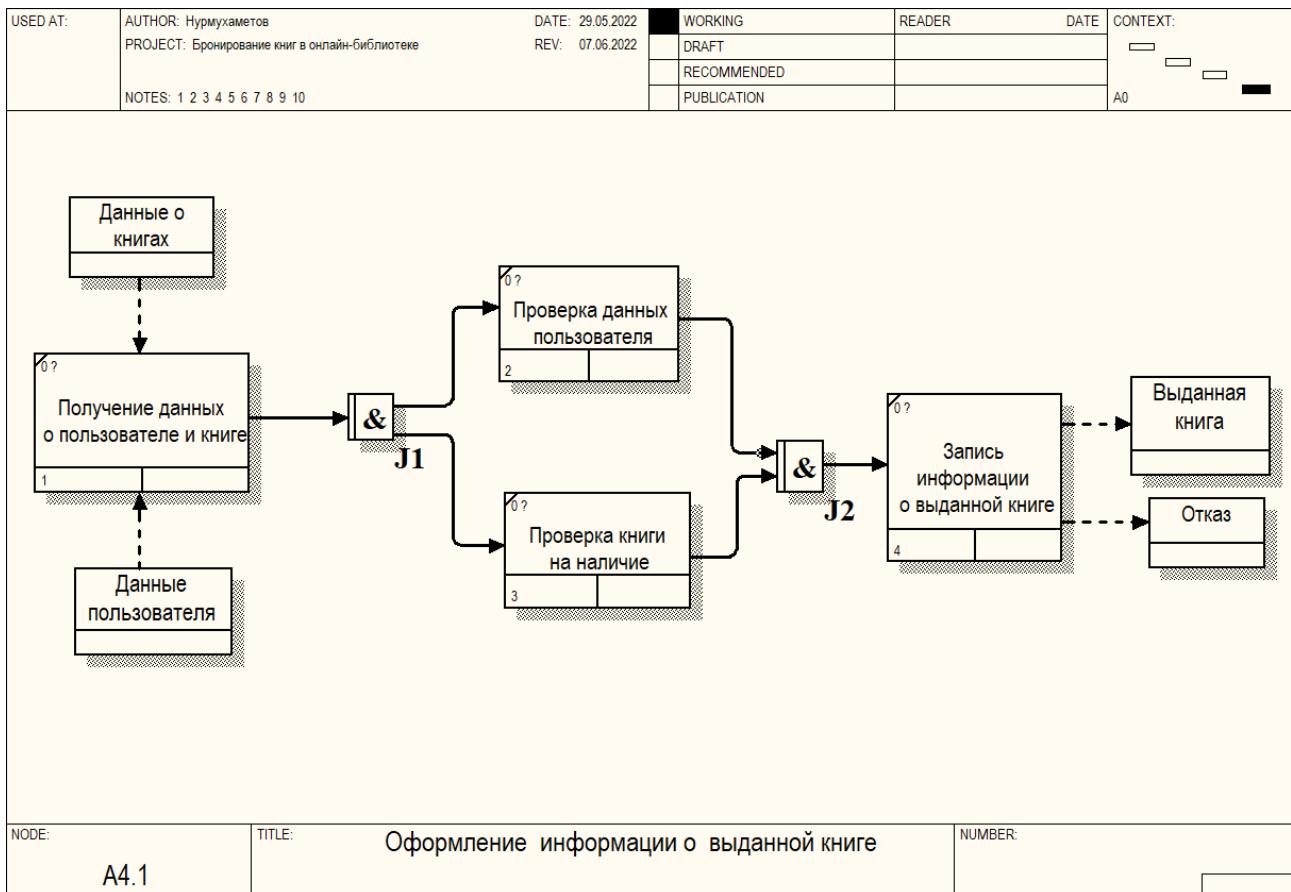


Рисунок 2.3 Декомпозиция блока “Оформление информации о выданной книге” в нотации IDEF3

С учётом спроектированного процесса выдачи книг, после процесса внедрения информационной системы, данная операция будет происходить в следующей последовательности:

- Пользователь регистрируется в системе.
- В случае, если он ввёл все свои данные, оператор подтверждает их и пользователю даётся доступ к бронированию книг и их скачиванию.
- Все новые книги добавляются в базу данных и оформляются оператором.
- При поступлении запросов на скачивание книг, система находит необходимый файл и выдаёт ей пользователю.
- При поступлении запроса на бронирование книги, информационная система ищет информацию о её физическом расположении и отправляет информацию на оформление выдачи.

- С учётом расположения книги и факта подтверждённости доступа пользователя происходит один из двух результатов: выдача книги или отказ, в случае недостаточного уровня доступа у пользователя или отсутствия книги в библиотеке.

2.2. Функционал системы на диаграмме прецедентов

Для понимания того, как будет работать будущая информационная система онлайн-бронирования книг в библиотеке, необходимо предварительно спроектировать способ взаимодействия с ней. Рассмотрим диаграмму прецедентов, где с системой взаимодействуют два актёра (Рисунок 2.4).

Оператор – работник системы, который будет редактировать данные в системе. В задачи оператора входят:

- Добавление или удаление книг.
- Изменение информации о книгах.
- Изменение информации о расположении книг.
- Добавление или удаление жанров.
- Подтверждение данных новых студентов.
- Сканирование возвращённых книг.
- Поиск книг.

Работа оператора системы связана с взаимодействием с информацией в базе данных и возвращением книг.

Читатель – пользователь системы, который будет запрашивать данные в системе. В возможности читателей входят:

- Скачивание книг.
- Регистрация в системе.
- Бронирование книг.
- Поиск книг.

Читатель в системе имеет возможность скачивать файлы из базы данных и бронировать книги на выдачу.



Рисунок 2.4 Диаграмма прецедентов

Диаграмма представляет взаимодействие оператора и читателя и их запросов к базе данных. Работа начинается с заполнения начальных данных, которые хранятся на физических носителях, далее происходит процесс запроса к базе данных, что включает в себя редактирование и удаление ранее записанных данных или добавление новых.

2.3. Инфологическая модель данных

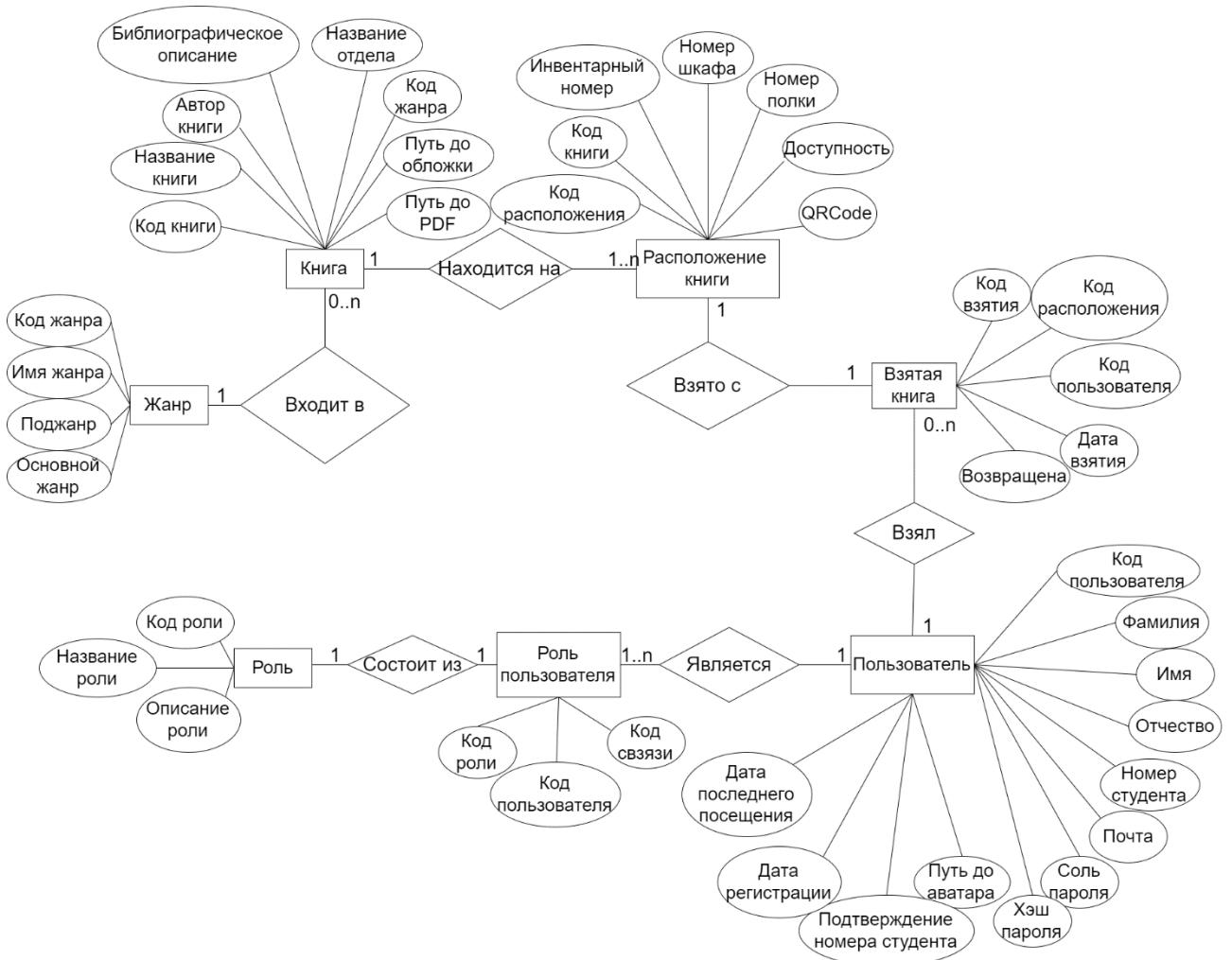


Рисунок 2.5 ER-модель в нотации Питера Чена

2.4. Даталогическая модель данных

Таблица 2.1

Сущность: жанр

Атрибут	Тип	Ключ	Описание
Код жанра	Числовой	Основной	Содержит виды жанров
Название	Текст		
Является поджанром	Логический		

Код основного жанра	Числовой	Вторичный	
Основной жанр	Объект сущности жанр		

Таблица 2.2

Сущность: книга

Атрибут	Тип	Ключ	Описание
Код книги	Числовой	Основной	Содержит книги
Название книги	Текст		
Автор книги	Текст		
Краткое название отдела	Текст		
Полное библиографическое описание	Текст		
Код жанра	Числовой	Вторичный	
Жанр	Объект сущности жанр		
Путь к обложке книги	Текст		
Путь к PDF книги	Текст		

Таблица 2.3

Сущность: книга на полке

Атрибут	Тип	Ключ	Описание
Код книги на полке	Числовой	Основной	
Код книги	Числовой	Вторичный	

Инвентарный номер	Числовой		Содержит расположение книги на полке
Номер шкафа	Числовой		
Номер полки	Числовой		
Книга доступна	Логический		
QRCode	Байты		

Таблица 2.4

Сущность: взятая книга

Атрибут	Тип	Ключ	Описание
Код взятой книги	Числовой	Основной	Содержит информацию о взятой книгу
Код книги на полке	Числовой	Вторичный	
Книга	Объект сущности книга		
Код пользователя	Числовой	Вторичный	
Дата взятия книги	Дата		
Возвращено	Логический		

Таблица 2.5

Сущность: пользователь

Атрибут	Тип	Ключ	Описание
Код пользователя	Числовой	Основной	Содержит данные пользователя
Фамилия	Текст		
Имя	Текст		
Отчество	Текст		

Номер студенческого билета	Текст		
Подтверждение студ билета	Логический		
Почта	Текст		
Соль пароля	Байты		
Хэш пароля	Текст		
Путь к аватару	Текст		
Дата регистрации	Дата		
Дата последнего доступа к сайту	Дата		

Таблица 2.6

Сущность: Роль

Атрибут	Тип	Ключ	Описание
Код роли	Числовой	Основной	Содержит роли пользователей
Название роли	Текст		
Описание роли	Текст		

Таблица 2.7

Сущность: Роли пользователя

Атрибут	Тип	Ключ	Описание
Код роли пользователя	Числовой	Основной	Содержит связь роли и пользователя
Код пользователя	Числовой	Вторичный	
Код роли	Числовой	Вторичный	

Роль	Объект сущности роль		

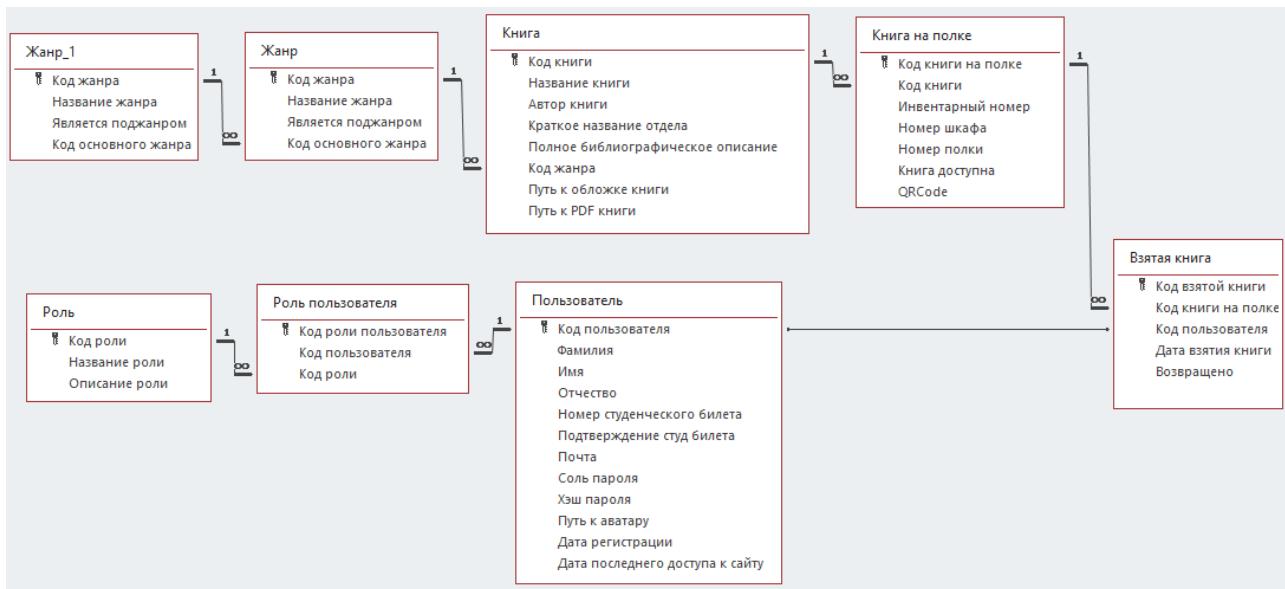


Рисунок 2.6 Даталогическая модель данных

Таким образом спроектированная база данных состоит из семи таблиц: Жанр, Книга, Книга на полке, Взятая книга, Роль пользователя и Роль [30,38].

Выводы по второй главе

Разработана структурно функциональная диаграмма по типу “Как должно быть” с использованием нотаций IDEF0 и IDEF3, спроектирована диаграмма вариантов использования информационной системы работы библиотеки Бирского филиала Башкирского государственного университета. Показано, что добавление онлайн-сервиса может привести к увеличению числа посетителей библиотеки, а также упрощению процесса выдачи запрашиваемой книги.

Разработана инфологическая модель данных с использованием ER-модели в нотации Питера Чена, после чего переработана в даталогическую, на основе которой спроектирована база данных при помощи программных средств MySQL.

ГЛАВА 3. РЕАЛИЗАЦИЯ КОМПОНЕНТОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1. Программная реализация интерфейса

Интерфейс программы представлен в виде страниц формата cshtml, написанных с использованием объединения двух языков: языка разметки HTML и языка программирования C# [3,17,23-26]. Для реализации интерфейса использован набор инструментов Bootstrap, предоставляющий доступ к большой библиотеке HTML- и CSS-шаблонов. Примером представлена одна из созданных страниц в коде ниже.

```
@model WebProject.Domain.Models.Books.BooksOnShelves

<form asp-action="ChangeBookOnShelves" asp-controller="Books" asp-anti-forgery="true" enctype="multipart/form-data">
    <div class="validation" asp-validation-summary="ModelOnly"></div>
    <div>
        @Html.HiddenFor(id => Model.BookOnShelfID)
        @Html.HiddenFor(id => Model.BookId)
        <div>
            <label asp-for="BookCaseNumber" class="col-form-label-lg">Номер шкафа</label><br />
            <input type="text" asp-for="BookCaseNumber" class="form-control-lg" value="@Model.BookCaseNumber" />
            <span asp-validation-for="BookCaseNumber" />
        </div>
        <div>
            <label asp-for="ShelfNumber" class="col-form-label-lg">Номер полки</label><br />
            <input type="text" asp-for="ShelfNumber" class="form-control-lg" value="@Model.ShelfNumber" />
            <span asp-validation-for="ShelfNumber" />
        </div>
        <label asp-for="IsAvailable" class="col-form-label-lg">Находится в библиотеке (не выдана на руки)</label>
        <input asp-for="IsAvailable" type="checkbox" />
        <span asp-validation-for="IsAvailable" />
        <hr color="black" />
        <br />
        <div>
            <input type="submit" value="Сохранить изменения" class="btn btn-lg btn-primary" />
        </div>
    </div>
</form>
```

Для того, чтобы отправлять необходимые данные от созданной страницу в контроллер обработки информации, используется инструмент, встроенный в выбранную среду разработки ASP.NET Core, тэг-хэлпер. На показанном ниже примере определён тег `<a>`, атрибутами которого объявлены три тэг-хэлпера: `asp-action`, определяющий, в какой метод будут отправлены данные, `asp-controller` - какой контроллер будет использован и `asp-route`, описывающий в какой параметр будет отправлена информация, в котором `userId` обозначает имя параметра.

```
<a asp-action="ChangeUser" asp-controller="Profile" class="btn" asp-route-userId="@u.UserId">
```

Для определения того, какие данные будут использоваться на странице, используется ключевой компонент выбранного паттерна работы с данными MVC (model-view-controller) – модели. Данный компонент описывает структуру и логику используемых данных, а также может хранить и передавать на страницу необходимые данные. На примере кода ниже показана модель `UserInfo`, передающая информацию о пользователе на страницу изменения его данных.

```
@model WebProject.Models.UserInfo
```

Также для передачи данных на страницу используется инструмент `ViewBag`, позволяющий передавать на представление любые заключённые в нём данные. Ниже показано создание инструмента `ViewBag.Genres`, содержащего список жанров, хранящихся в базе данных, для передачи его на страницу создания книги.

```
ViewBag.Genres = _context.Genres.ToList();
```

А на следующем примере показано использование данного инструмента для создания списка `select`, для выбора искомого жанра.

```
<select asp-for="GenreName">
    @foreach (var genre in ViewBag.Genres)
    {
        <option value="@genre.GenreName">@genre.GenreName</option>
    }
</select>
```

Третьим способом передачи данных является инструмент `ViewData`, который схож по функционалу с `ViewBag`. Использование `ViewData` в

реализованной системе показано на примерах ниже для передачи информации о названии страницы.

Передача данных с использованием ViewData:

```
ViewData["Title"] = "Новая книга";
```

Использование ViewData:

```
<title>@ViewData["Title"]</title>
```

Благодаря использованию среды ASP.NET Core общую часть всех страниц HTML нет необходимости прописывать каждый раз, так как они находятся в файле _Layout.cshtml. Первой общей частью является тэг <head>, в котором хранятся данные о кодировке текста, размере изображения для сайта, тэг <title>, задающий название страницы и ссылки на используемые библиотеки стилей.

```
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"]</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" />
</head>
```

Оставшаяся часть, убранная в _Layout.cshtml, является тэгом <body>, хранящим основную часть страницы. Но она также разделена на несколько тэгов. Первый из которых <header>, отвечающий за первый уровень навигации. Он хранит ссылки на страницы регистрации, авторизации, профиля и выхода из аккаунта пользователя.

```
<header class="blog-header py-3">
    <div class="container">
        <ul class="bg-transparent user-links wht-list">
            @if (!User.Identity.IsAuthenticated)
            {
                <li class="entry bg-transparent">
                    <a class="bg-transparent btn btn-sm btn-outline-secondary" asp-area="" asp-controller="Profile" asp-action="Login">Войти</a>
                </li>
                <li class="reg bg-transparent">
                    <a class="bg-transparent btn btn-sm btn-outline-secondary" asp-area="" asp-controller="Profile" asp-action="Register">Зарегистрироваться</a>
                </li>
            }
            else
            {
```

```

        <li class="profile bg-transparent">
            <a class="btn btn-sm btn-outline-secondary" asp-area=""
                asp-controller="Profile" asp-action="Profile"><img
                src "~/files/person.png" /> Профиль</a>
        </li>
        <li class="exit bg-transparent">
            <a class="btn btn-sm btn-outline-secondary" asp-area=""
                asp-controller="Profile" asp-action="Exit">Выйти</a>
        </li>
    }
</ul>
<div class="bg-transparent logo">
    <a class="background-color: #fff;" asp-area="" asp-
        controller="Home" asp-action="Index">
        <img src "~/files/logo.png" alt="Онлайн-библиотека"
            height="50" />
    </a>
</div>
</div>
</header>

```

На втором уровне авторизации расположены ссылки на основную работу с книгами, а также пользователями для оператора. Также в данном разделе находится метод, позволяющий вызывать и показывать всё оставшееся представление: RenderBody().

```

<div class="container">
    <div class="nav-scroller py-1 mb-2">
        <nav class="nav d-flex justify-content-between">
            @if (User.IsInRole("Администратор"))
            {
                <a class="p-2 link-secondary" asp-area="" asp-
                    controller="Profile" asp-action="ListOfUsers">Пользователи</a>
                <a class="p-2 link-secondary" asp-area="" asp-
                    controller="Books" asp-action="ListOfBooksForRent">Список книг
                    на выдачу</a>
                }
                <a class="p-2 link-secondary" asp-area="" asp-
                    controller="Books" asp-action="Genres">Жанры</a>
                <a class="p-2 link-secondary" asp-area="" asp-
                    controller="Books" asp-action="ListOfBooks" asp-route-
                    pageNumber="1">Книги</a>
                <a class="p-2 link-secondary" asp-area="" asp-
                    controller="Books" asp-action="BooksSearch">Поиск книг</a>
            </nav>
        </div>
        <main role="main" class="pb-3">
            @RenderBody()
        </main>
    </div>

```

Последними частями общего файла _Layout.cshtml является тэг <footer>, в котором хранится минимальная информация о сайте, а также ссылки на файлы JavaScript.

```
<footer class="border-top text-muted">
    <div class="container">
        © 2022 - Online-Biblio
    </div>
</footer>
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="~/js/site.js" asp-append-version="true"></script>
@await RenderSectionAsync("Scripts", required: false)
```

3.2. Реализация базы данных

Используемая технология Entity Framework Core позволяет применять подход CodeFirst, дающий возможность сначала написать код, определяющий модель и после этого будет создана база данных и её таблицы [16,19]. Пример ниже служит для создания таблицы данных пользователя:

```
public int UserId { get; set; }
public string SurName { get; set; } = string.Empty;
public string Name { get; set; } = string.Empty;
public string? MiddleName { get; set; }
public string? StudentIDNumber { get; set; }
public bool ConfirmedStudentId { get; set; }
public string Email { get; set; } = string.Empty;
public byte[] PasswordSalt { get; set; }
public string PasswordHash { get; set; } = string.Empty;
public string? AvatarPath { get; set; }
public DateTime RegistrationDate { get; set; }
public DateTime? LastSiteAccessDate { get; set; }
```

Для связи моделей и базы данных используется контекст данных, используя класс DbContext и наборы сущностей DbSet .

```
public DbSet<User> Users => Set<User>();
public DbSet<Role> Roles => Set<Role>();
public DbSet<UserRole> UserRoles => Set<UserRole>();
public DbSet<Book> Books => Set<Book>();
public DbSet<BooksOnShelves> BooksOnShelves => Set<BooksOnShelves>();
public DbSet<Genre> Genres => Set<Genre>();
public DbSet<TakenBooks> TakenBooks => Set<TakenBooks>();
```

Для подключения к физической базе данных необходимо определить параметры подключения в файле appsettings.json, прописывая путь к файлу и его имя.

```
{  
    "Backup.ConnectionStrings": {  
        "BackupConnection":  
            "Server=(localdb)\\mssqllocaldb;Database=bibliodb;Trusted_Connection=True  
            ;MultipleActiveResultSets=true"  
    },  
    "Logging": {  
        "LogLevel": {  
            "Default": "Information",  
            "Microsoft": "Warning",  
            "Microsoft.Hosting.Lifetime": "Information"  
        }  
    },  
    "AllowedHosts": "*",  
    "ConnectionStrings": {  
        "SqliteConnection": "Data Source=BiblioDb.db"  
    }  
}
```

Для получения доступа к базе данных в контроллере имеется два способа: с использованием контекста данных напрямую или же создание класса репозитория, использующего метод CRUD [1]. В первом случае необходимо определить в контроллере переменную класса BiblioDbContext.

```
public BooksController(BiblioDbContext context, IWebHostEnvironment  
appEnvironment)  
{
```

```
    _context = context;
```

После добавления в контроллер контекста появляется возможность обращаться к базе данных, используя методы языка запросов LINQ.

```
[HttpGet]  
public async Task<IActionResult> ChangeBook(int bookID)  
{  
    if (User.IsInRole("Администратор"))  
    {  
        Book oldBook = await _context.Books.FirstOrDefaultAsync(b =>  
            b.BookId == bookID);  
        ChangedBook model = new ChangedBook() { BookId = oldBook.BookId,  
            BookAuthor = oldBook.BookAuthor, BookTitle = oldBook.BookTitle};  
        if (oldBook.Genre != null)  
            model.Genre = oldBook.Genre.GenreName;  
        ViewBag.Genres = _context.Genres.ToList();  
        ViewData["Title"] = model.BookTitle;  
        return View(model);  
    }  
    return RedirectToAction("Index", "Home");  
}
```

Для второго способа обращения к базе данных необходимо использовать паттерн “Репозиторий”, позволяющий абстрагироваться от прямого обращения к контексту через контроллер и хранящий в себе заранее описанные методы обращения к данным. Для удобства и уникальности работы репозиториев они создаются для каждой из таблиц.

```
public class GenreRepository : BaseRepository, IRepository<Genre>
{
    public GenreRepository(BiblioDbContext context) : base(context){ }
    public async Task CreateAsync(Genre item)
    {
        await _context.Genres.AddAsync(item);
    }
    public Task DeleteAsync(int id)
    {
        var genre = new Genre() { GenreID = id };
        _context.Genres.Remove(genre);
        return Task.CompletedTask;
    }
}
```

После создания репозитория необходимо добавить в контроллер параметр класса репозитория, после чего, можно будет использовать его методы для работы с данными.

```
public async Task<IActionResult> ChangeBook(ChangedBook model)
{
    if (ModelState.IsValid)
    {
        Book changedBook = await _context.Books.FirstOrDefaultAsync(b =>
b.BookId == model.BookId);
        if (changedBook != null)
        {
            if (model.BookTitle != "")
                changedBook.BookTitle = model.BookTitle;
            if (model.BookAuthor != "")
                changedBook.BookAuthor = model.BookAuthor;
            if (model.BookPDFPath != null)
                changedBook.BookTitle = model.BookTitle;
            if (model.BookTitle != "")
                changedBook.BookTitle = model.BookTitle;
            Genre newGenre = _context.Genres.FirstOrDefault(g =>
g.GenreName == model.Genre);
            changedBook.Genre = newGenre;
            changedBook.GenreID = newGenre.GenreID;
            await bookRepository.UpdateAsync(changedBook);
            await bookRepository.SaveAsync();
            return RedirectToAction("Index", "Home");
        }
        else
            ModelState.AddModelError("", "Такой книги не существует");
    }
}
```

```
        return View(model);  
    }  
}
```

3.3. Руководство пользователя

Главная страница. Главная страница (Рисунок 3.1) предоставляет доступ к регистрации нового пользователя, авторизации, в том случае, если пользователь уже зарегистрирован в системе, просмотру списка последних шести новых книг, а также возможность перейти к пунктам “Жанры” – со списком всех жанров в системе, “Книги” – списку всех книг в системе, а также “Поиск книг” – поиск книг по необходимой фразе.

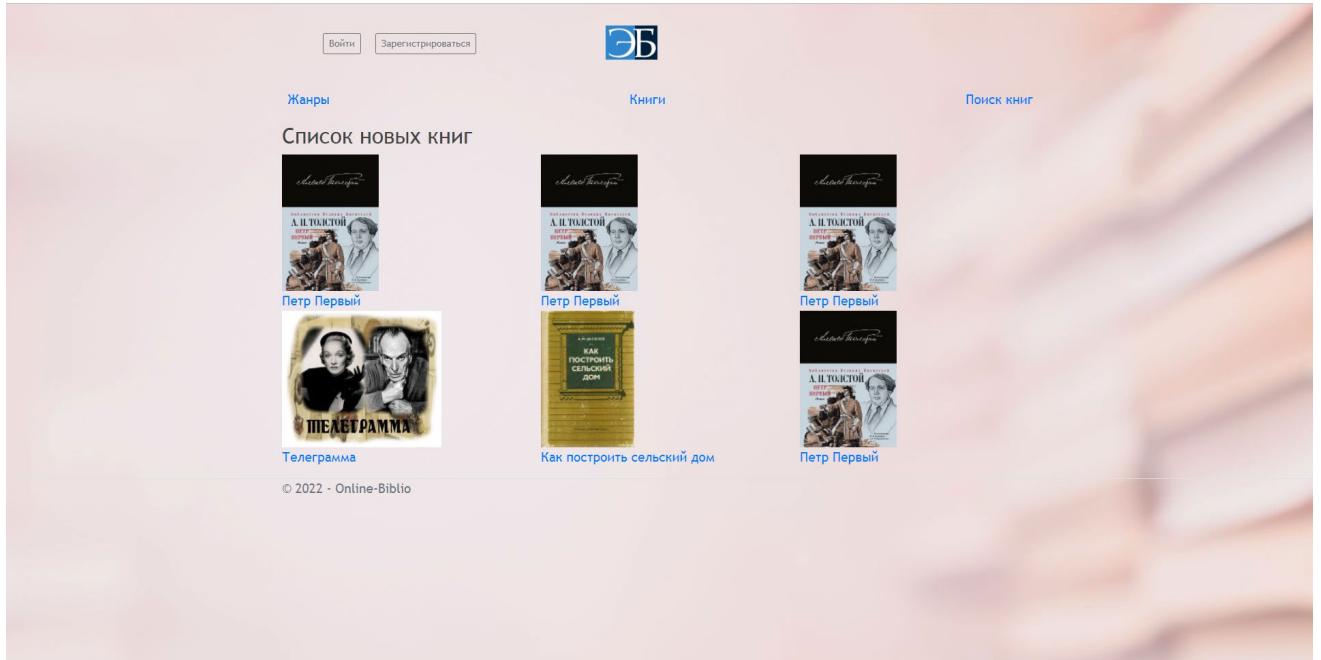


Рисунок 3.1 Главная страница сайта

При авторизации пользователя с ролью “Библиотекарь” (Рисунок 3.2) также появляется доступ к странице “Список книг на выдачу”, а при авторизации “Администратора”, добавляется пункт “Пользователи”.

Для всех пользователей после выполнения пунктов “Войти” и “Зарегистрироваться” появляются пункты “Профиль”, перенаправляющий пользователя на страницу его профиля и “Выход”.

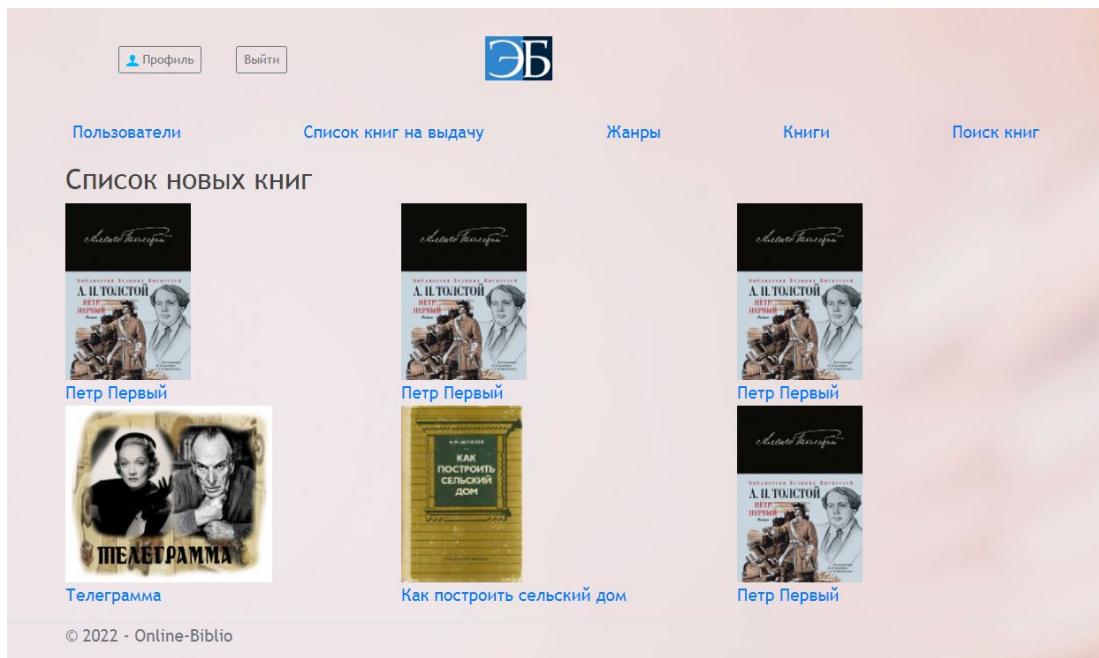


Рисунок 3.2 Главная страница сайта после авторизации

Страница авторизации. На данной странице (Рисунок 3.3) пользователи, зарегистрированные в системе, могут совершить авторизацию под своим аккаунтом, чтобы получить больший доступ, введя свою почту и пароль.

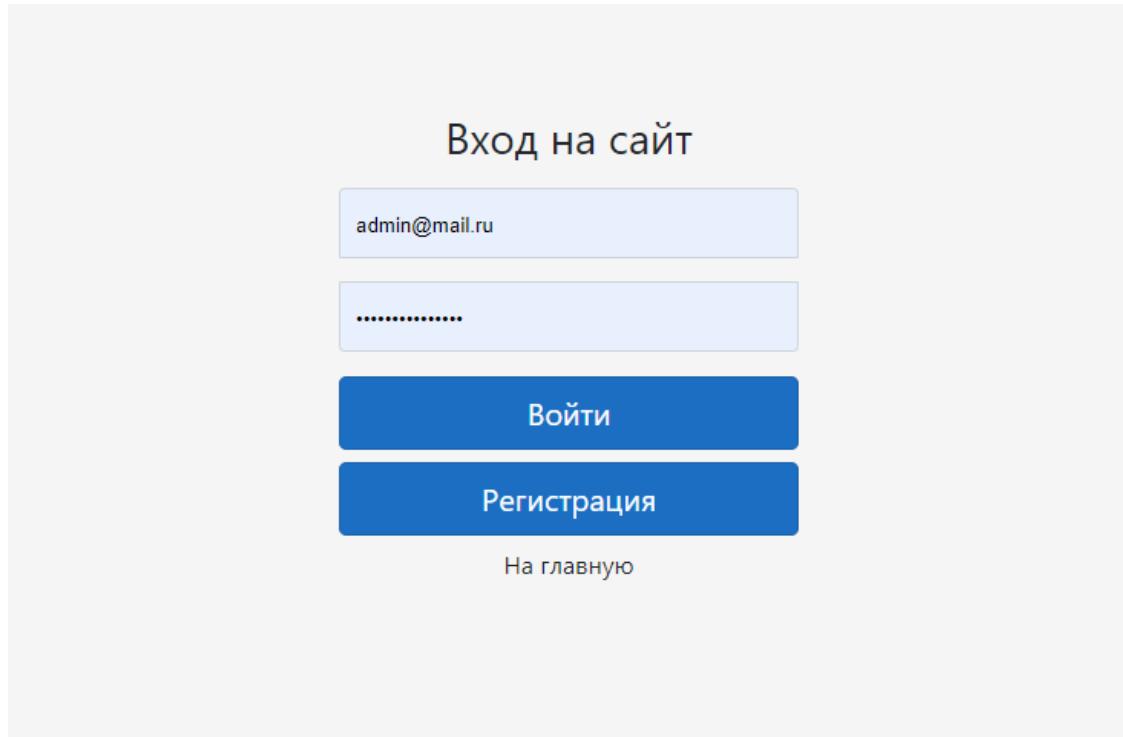


Рисунок 3.3 Страница авторизации

Страница регистрации. На странице регистрации (Рисунок 3.4) пользователь, ещё не зарегистрированный в системе, может создать новый аккаунт, введя необходимые данные. Обязательными данными являются: фамилия, имя, Email, пароль и подтверждение пароля. Номер студенческого билета, отчество и аватар могут быть добавлены позже.

Регистрация

Введите фамилию*

Введите имя*

Введите отчество

Введите номер студенческого билета

Введите Email*

Введите пароль*

Повторите пароль*

Выберите аватар

Выбор файла

Не выбран ни один файл

*

Регистрация

Рисунок 3.4 Страница регистрации

Страница списка пользователей. На странице списка пользователей (Рисунок 3.5) администратор может выбрать необходимого пользователя для изменения его данных. При помощи кнопки “Пользователи без подтверждения” в списке система проведёт сортировку для получения списка пользователей, которые уже ввели номер студенческого билета, но ещё не получили его подтверждение или вообще не ввели данные билета.

The screenshot shows a web application interface for managing users. At the top, there are navigation links: 'Профиль' (Profile), 'Выход' (Logout), and a logo 'ЭБ'. Below these are four more links: 'Пользователи' (Users), 'Список книг на выдачу' (List of books for issue), 'Жанры' (Genres), and 'Книги' (Books). The main title is 'Список пользователей' (List of users). A blue button labeled 'Пользователи без подтверждения' (Users without confirmation) is highlighted. Below this, five user profiles are listed in two rows:

Фамилия	Имя	Почта
Administrator	Main	admin@mail.ru
User	Test	test@mail.ru
One	Stundent	student1@mail.ru
Two	Student	student2@mail.ru
Librarian	Test	librarian@mail.ru

Рисунок 3.5 Страница списка пользователей

Страница изменения данных пользователя. На данной странице (Рисунок 3.6) администратор или модератор может изменять данные пользователя: менять его ФИО, номер студенческого билета, почту, аватар, узнать его роли, а также подтверждать его студенческий билет.

Фамилия

User

Имя

Test

Номер студенческого билета

Подтверждение студенческого билета

Email

test@mail.ru

Аватар

Выбор файла

Не выбран ни один файл

Роли

Пользователь

Сохранить изменения

Рисунок 3.6 Страница изменения данных пользователя

Страница профиля. На данной странице (Рисунок 3.7) пользователь может посмотреть свой профиль, в котором указана его личная информация, а также перейти к списку взятых им книг и перейти к редактированию своих данных.

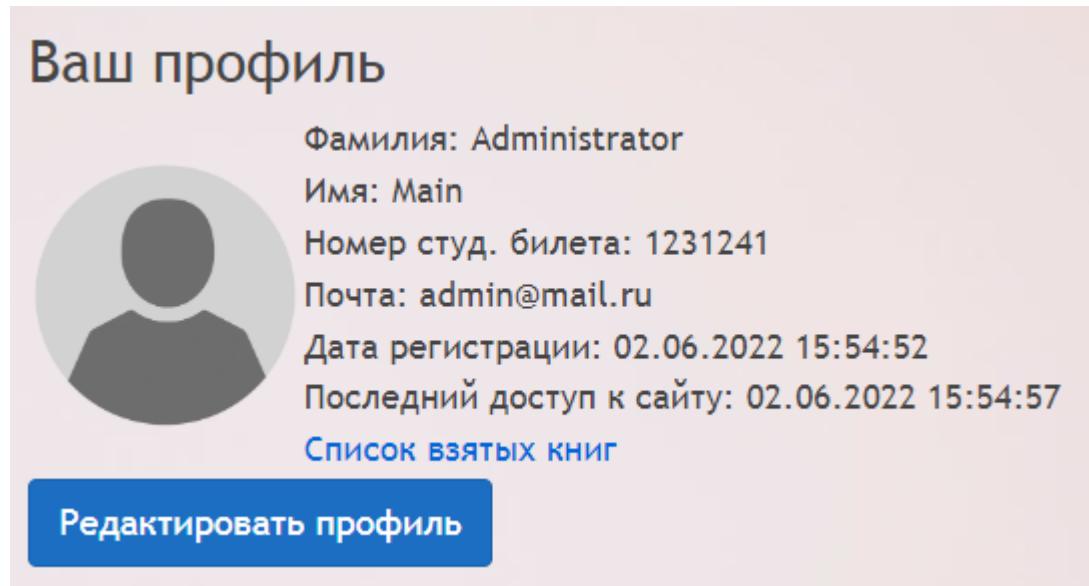


Рисунок 3.7 Страница профиля

Страница списка жанров. На странице списка жанров (Рисунок 3.8) пользователь может выбрать нужный ему жанр и перейти к отсортированному списку книг, в котором будут только книги с выбранным жанром. Оператор также имеет возможность добавить новый жанр и удалить существующий.

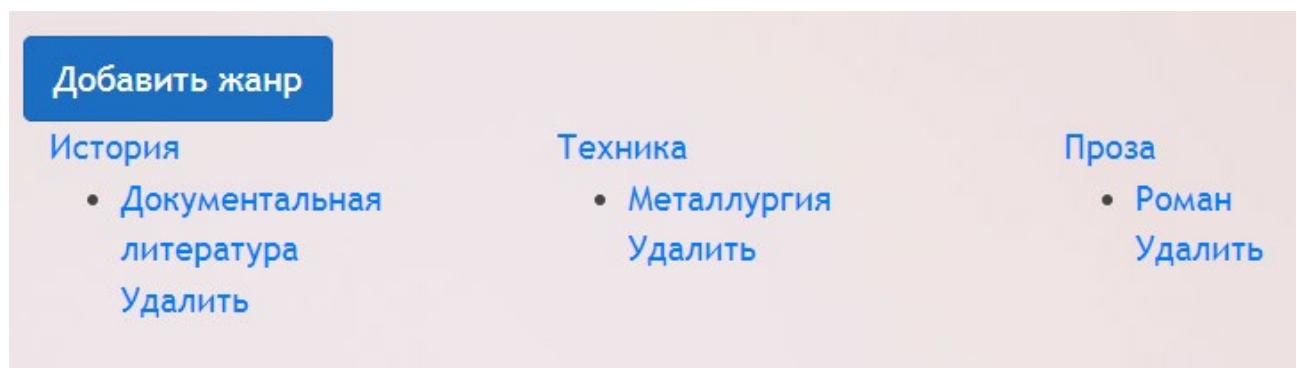


Рисунок 3.8 Страница списка жанров

Страница нового жанра. На странице добавления нового жанра (Рисунок 3.9) оператор вводит название нового жанра и при необходимости отмечает, что он является поджанром, после чего выбирает главный для него жанр.

The screenshot shows a form titled 'Создание нового жанра'. It has a field labeled 'Введите название жанра*' containing 'Информационная база'. Below it is a question 'Является ли поджанром*' with a checked checkbox. A section for 'Главный жанр' shows 'Программирование' in a dropdown menu. At the bottom is a note '* - обязательные поля' and a blue 'Добавить' button.

Создание нового жанра

Введите название жанра*

Информационная база

Является ли поджанром*

Главный жанр

Программирование ▾

* - обязательные поля

Добавить

Рисунок 3.9 Страница нового жанра

Страница списка книг. На странице списка литературы (Рисунок 3.10) система выводит по 4 книги за один раз, пользователь может увидеть остальные книги переходя по страницам списка, используя кнопки снизу. Пользователь может выбрать нужную ему книгу и перейти на её страницу, а у оператора есть также возможность добавить новые книги по одной или из Excel документа сразу несколько.

[Добавить книгу](#)[Добавить книги из Excel документа](#)

СПИСОК КНИГ



Название: Петр Первый

Автор: Толстой Алексей Николаевич

Жанр: История



Название: Петр Первый

Автор: Толстой Алексей Николаевич

Жанр: История



Название: Петр Первый

Автор: Толстой Алексей Николаевич

Жанр: История



Название: Телеграмма

Автор: Паустовский Константин Георгиевич

Жанр: Проза

[Предыдущая](#) [Следующая](#)

Рисунок 3.10 Страница список книг

Страница книги. На странице книги (Рисунок 3.11) не подтверждённый или не авторизированный пользователь может видеть только общую информацию о выбранной книге.

Телеграмма



Автор: Паустовский Константин Георгиевич
Жанр: Проза

Рисунок 3.11 Страница книги без подтверждения/авторизации

Авторизированный (Рисунок 3.12) же пользователь имеет возможность запросить на скачивание PDF файл выбранной книги, а также забронировать книгу, в том случае, если она доступна в данный момент. Также на странице будет виден QRCode, просканировав который, можно получить общую информацию о книге и её расположении и куда её вернуть в случае нахождения.

Телеграмма



Автор: Паустовский Константин Георгиевич
Жанр: Проза

PDF:[Скачать](#)

Забронировать книгу:[Забронировать](#)

QRCode:



Рисунок 3.12 Страница книги после авторизации и подтверждения

У оператора на странице книги (Рисунок 3.13) имеется возможность, в дополнение к существующим, перейти к странице изменения данных книги, странице добавления/изменения расположения книги и её удалению.

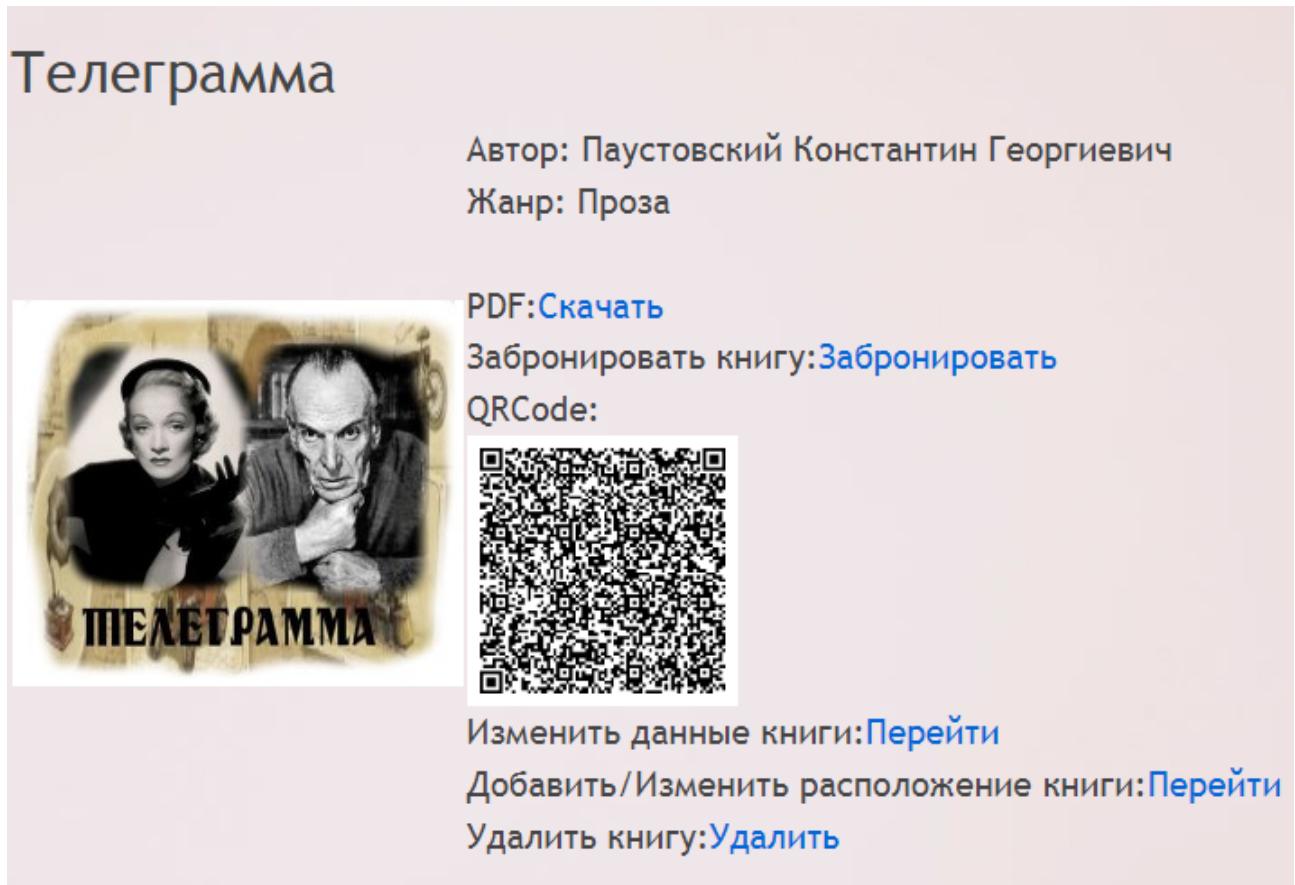


Рисунок 3.13 Страница книги у оператора

Страница изменения данных книги. На странице изменения данных книги (Рисунок 3.14) оператор может изменить название книги, её автора, выбрать нужный жанр, изменить обложку книги и загрузить PDF файл.

Название книги

Телеграмма

Автор книги

Паустовский Константи

Выберите жанр*

Проза

Выберите обложку книги

Выбор файлаНе выбран ни один файл

Выберите PDF-версию книги

Выбор файлаНе выбран ни один файл

Сохранить изменения

Рисунок 3.14 Страница изменения данных книги

Страница изменения расположения книг. На странице изменения расположения книг (Рисунок 3.15) оператор может изменить номер шкафа и номер полки, на которой находится книга.

Номер шкафа

2

Номер полки

3

Находится в библиотеке (не выдана на руки)

Сохранить изменения

Рисунок 3.15 Страница изменения расположения книг

Страница поиска книг. На странице поиска книг (Рисунок 3.16) посетитель сайта имеет возможность найти интересующую его книгу введя в окно поиска интересующую его информацию.

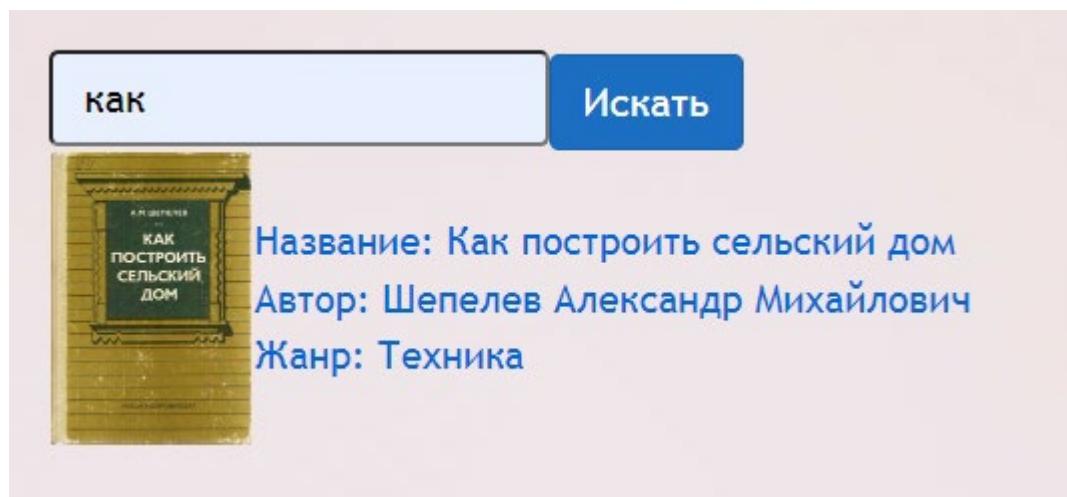


Рисунок 3.16 Страница поиска книг

Выходы по третьей главе

Представлена программная реализация компонентов онлайн-сервиса библиотеки Бирского филиала Башкирского государственного университета. Реализация интерфейса включает в себя объединение двух языков HTML и C#. Проведено описание набора инструментов для доступа к библиотекам и шаблонам. Разработана база данных пользователей с описанием её подключения к программе. Следует отметить, что разработка информационной системы выполнена с использованием современной платформы ASP.NET Core.

Предложено подробное руководство пользователя, включающее в себя возможности создания его профиля (авторизированный пользователь имеет больше возможностей), а также поиска, бронирования, скачивания книг, изменения данных о книге.

ЗАКЛЮЧЕНИЕ

На основании выполненного исследования информационных процессов библиотеки Бирского филиала Башкирского государственного университета, определены задачи автоматизации обработки и хранения информации о книгах и пользователях, а также выдачи книг. Детально изучен объект исследования с помощью спроектированных диаграмм в нотациях IDEF0 и IDEF3, диаграмм прецедентов и моделей представления баз данных. Решение задач работы определило направление разработки информационной системы библиотеки, а именно создание сервиса онлайн-бронирования книг и его функционал.

Разработка базы данных велась с использованием технологии Entity Framework Core на языке программирования C#. В ходе разработки созданы таблицы и связи между ними. Разработка сервиса онлайн-бронирования производилась на основе изученных платформ Visual Studio (C#), ASP.NET Core, интерфейс программы сделан интуитивно понятным для пользователя. К сайту написано руководство пользователя, необходимое в случае затруднений взаимодействия специалиста с программой.

Я подтверждаю, что настоящая работа написана лично мною, не нарушая интеллектуальные права третьих лиц и не содержит сведений, составляющих государственную тайну.

Илья Нуруллаев

09.06.2022

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Microsoft Corporation. Принципы проектирования и разработки программного обеспечения. Учеб. курс MCSD. М.: Изд.-торг. дом «Русская редакция», 2000.
2. Агуров П. В. C#. Разработка компонентов в MC Visual studio 2005/2008 / Агуров П. В. – СПб.:БХВ-Петербург, 2008. – 479 с.
3. Албахари Дж. C# 6.0. Справочник. Полное описание языка [Текст] / Дж. Албахари, Б. Албахари – 6-е изд. – Москва: Вильямс, 2016. – 1040 с.
4. Анисимова Г. Б., Романенко М. В. Выбор методологии проектирования информационных систем. Структурный подход // Научное обозрение. 2017. №22. С. 74–79.
5. Артамонов И.В. Разработка распределенных сервисно-ориентированных программных средств / И.В. Артамонов – Иркутск: Изд-во БГУЭП, 2012. – 130 с.
6. Биллиг В. А. Основы объектного программирования на C# (C# 3.0, Visual Studio 2008) [Текст]: учебное пособие / В. А. Биллиг. – Москва: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2016. – 584 с.
7. Биллиг В.А. Основы объектного программирования на C# (C# 3.0, Visual Studio 2008) [Электронный ресурс]: учебное пособие / В.А. Биллиг. – Электрон. текстовые данные. – Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. – 583 с.
8. Биллиг, В. А. Объектное программирование в классах на C# 3.0 [Электронный ресурс] / В.А. Биллиг. - 2-е изд., испр.. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 391 с.
9. Борисенко В.В. Основы программирования [Электронный ресурс] / В.В. Борисенко. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 323 с.

10. Введение в программирование на языке С#: курс лекций и лаб. практикум: учеб. пособие / А. В. Котельникова, М. Л. Вотинцева. - Киров: Изд-во ВятГГУ, 2011. - 299 с.: ил. - Библиогр.: с. 299
11. Гагарина Л.Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул. – М.: ИД «ФОРУМ»: ИНФРА-М, 2009.
12. Герман О. Программирование на Java и C# для студента [Текст]: учебное пособие / О. Герман, Ю. Герман. – Санкт-Петербург: БХВПетербург, 2014. – 512 с.
13. Грин Дж. Изучаем C# / Дж. Грин, Э. Стиллмен. – СПб.: «Питер», 2012.
14. Джон, Скит С#. Программирование для профессионалов / Скит Джон. - М.: Диалектика / Вильямс, 2017. – 2530 с.
15. Дригайло, В. Г., Башун Е. В., Волынец В. Н. Основы управления библиотекой высшего учебного заведения: Научно-практическое пособие.-2-е изд., перераб. и доп.- М.: ЛИБЕРИЯ, 2004 – 328с.
16. Илюшечкин, В. М. Основы использования и проектирования баз данных / В.М. Илюшечкин. - М.: Юрайт, 2015. – 214 с.
17. Ишкова Э. А. Самоучитель С#. Начала программирования [Текст]: учебное пособие / Э. А. Ишкова – 2-е изд. – Санкт-Петербург: Наука и Техника, 2013. – 496 с.
18. Каменнова М.С. Моделирование бизнес-процессов. Часть 1: учебник и практикум для академического бакалавриата. – М.: Издательство Юрайт, 2019. – 282 с.
19. Кузнецов, С. Д. Базы данных. Модели и языки / С.Д. Кузнецов. - М.: Бином-Пресс, 2013. – 720 с.
20. Культин, Н. С# в задачах и примерах / Н. Культин. - М.: БХВ-Петербург, 2020. – 1293 с.
21. Лотка, Р. С# и CSLA .NET Framework. Разработка бизнес-объектов / Р. Лотка. – М.: Диалектика / Вильямс, 2017. – 555 с.

22. Маклаков С.В. Моделирование бизнес-процессов. - М.: ДИАЛОГМИФИ, 2002. – 180 с.
23. Нейгел, Кристиан C# 4.0 и платформа .NET 4 для профессионалов (+ CD-ROM) / Кристиан Нейгел и др. - М.: Вильямс, 2020. - 1440 с.
24. Нейгел, Кристиан C# 5.0 и платформа .NET 4.5 для профессионалов / Кристиан Нейгел и др. - М.: Вильямс, 2020. - 1440 с.
25. Пахомов, Борис C# для начинающих / Борис Пахомов. - М.: БХВ-Петербург, 2017. – 2271 с.
26. Прайс, Джейсон Visual C# 2.0. Полное руководство / Джейсон Прайс, Майк Гандэрлой. – М.: Век +, Корона-Век, Энтроп, 2019. – 736 с.
27. Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов. – М.: Манн, Иванов и Фербер, 2013. – 544 с.
28. Силич В.А., Силич М.П. Моделирование и анализ бизнес-процессов: учебное пособие. – Томск: Издательство Томск. гос. ун.-та систем управления и радиоэлектроники, 2011. – 212 с.
29. Фримен Адам Entity Framework Core 2 для ASP.NET Core MVC для профессионалов. – М.: «Диалектика», 2019. – 626 с
30. Фуфаев, Э. В. Базы данных. Учебное пособие / Э.В. Фуфаев, Д.Э. Фуфаев. - М.: Академия, 2014. - 320 с.
31. Шилдт Г. Полное руководство C# 4.0 [Текст]: учебное пособие / Г. Шилдт – пер. с англ. Берштейн И. В. – Москва: Вильямс, 2012– 1051 с.
32. SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems [Электронный ресурс]. Режим доступа: <https://medium.com/@yangforbig/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems-afd5af6566>, свободный. – (Дата обращения: 15.11.2021).
33. Библиотека [Электронный ресурс]. Режим доступа: <https://www.birsk.ru/lib>, свободный. – (Дата обращения: 02.04.2022).
34. EF Core In depth – Tips and techniques for configuring EF Core [Электронный ресурс]. Режим доступа:

<https://www.thereformedprogrammer.net/ef-core-in-depth-tips-and-techniques-for-configuring-ef-core/>, свободный. – (Дата обращения: 04.04.2022).

35. Общие сведения об ASP.NET Core [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-3.1>, свободный. – (Дата обращения: 19.11.2021).

36. Основы LINQ [Электронный ресурс]. Режим доступа: <https://metanit.com/sharp/tutorial/15.1.php>, свободный. – (Дата обращения: 20.11.2021).

37. СУБД [Электронный ресурс]. Режим доступа: <https://blog.skillfactory.ru/glossary/subd/>, свободный. – (Дата обращения: 27.03.2022).

38. Что такое база данных? [Электронный ресурс]. Режим доступа: <https://www.oracle.com/cis/database/what-is-database/>, Свободный. – (Дата обращения: 26.03.2022).

39. Use Entity Framework Core 5.0 In .NET Core 3.1 With MySQL Database By Code-First Migration On Visual Studio 2019 For RESTful API Application [Электронный ресурс]. Режим доступа: <https://www.c-sharpcorner.com/article/tutorial-use-entity-framework-core-5-0-in-net-core-3-1-with-mysql-database-by2/>, Дата обращения: 20.11.2021, свободный. – (Дата обращения: 12.11.2021).

40. ASP.NET Core – Exceeds 1.15 Million request/s, 12.6 Gbps [Электронный ресурс]. Режим доступа: <https://www.ageofascent.com/2016/02/18/asp-net-core-exceeds-1-15-million-requests-12-6-gbps/>, свободный. – (Дата обращения: 25.02.2022).

Мурзалимов В.Р