



КУРСОВАЯ РАБОТА

**Тема: Разработка программного модуля системы
«Интернет магазин книг».**

**Специальность 09.02.07 Информационные системы и
программирование**

Выполнил студент(ка) группы _____ **Р.В. Капитонов**

Руководитель _____ **В.Ю. Назаров**

Москва 2023



УТВЕРЖДАЮ
Зам. директора КМПО
С.Ф. Гасанов
« _____ » _____ 2023 г.

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

по дисциплине: МДК.01.01 Разработка программных модулей
Специальность 09.02.07 Информационные системы и
программирование
Студент(ка) группы 31ИС-21 Капитонов Роман
ТЕМА: Разработка программного модуля системы
«Интернет магазин книг».

Дата выдачи задания « _____ » _____ 2023 г.
Срок сдачи работы « _____ » _____ 2023 г.

Москва 2023

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ
СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»
КОЛЛЕДЖ МНОГОУРОВНЕВОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

Задание на курсовой проект (работу)

Дисциплина: МДК.01.01 Разработка программных модулей

Тема: Разработка программного модуля системы

«Интернет магазин книг»

Специальность: 09.02.07 Информационные системы и программирование

Группа: 31ИС-21

ФИО студента Капитонов Р.В.

ФИО руководителя Назаров В.Ю.

1. Проанализировать предметную область
2. Проанализировать готовые решения
3. Подготовить техническое задание
4. Обосновать выбор инструментов и средств разработки
5. Описать реализацию технического задания
6. Выполнить тестирование

Задание выдано « ____ » _____ 2023 г.

Срок выполнения « ____ » _____ 2023 г.

Сроки защиты _____

Преподаватель: _____

Задание получил: _____

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
1.1 Особенности интернет-магазинов.....	7
1.2 Анализ существующих решений.....	8
1.3 Постановка задач.....	9
ГЛАВА 2. ПРОЕКТИРОВАНИЕ МОДУЛЯ СИСТЕМЫ	11
2.1 Выбор и описание программных инструментов.....	11
2.2 Требования к функциям, выполняемым системой	12
2.3 Модуль Classes	13
2.4 Модуль BookDesktop	14
2.5 Структура приложения.....	15
2.6 Структура интерфейса приложения.....	15
2.7 План тестирования	18
ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОЕКТА МОДУЛЯ СИСТЕМЫ	19
3.1 Описание кодом функциональных узлов модуля BookDesktop.....	19
3.2 Описание кодом функциональных узлов модуля Classes	25
3.3 Тестирование приложения	27
ЗАКЛЮЧЕНИЕ	30
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	32

ВВЕДЕНИЕ

Актуальность темы заключается в том, что интернет-магазины книг занимают свою нишу среди любителей литературы. Это связано с тем, что они предлагают широкий ассортимент книг, низкие цены и удобство покупки. Однако, для обеспечения эффективной работы интернет-магазина книг необходимо наличие развитой информационной системы, в том числе для сотрудников организации.

В такой системе должны выполняться такие функции как:

- регистрация и авторизация пользователей;
- поиск книг;
- добавление книг в корзину;
- оформление заказа;
- оплата заказа;
- доставка заказа;
- реализация подписок на категории или авторов;
- обработка и поддержание информации о книгах в базе данных;
- процедуры обработки заказов (отмена, перенос).

Разработка программного модуля системы «Интернет магазин книг» является актуальной задачей, поскольку позволит улучшить работу интернет-магазина и повысить его конкурентоспособность.

Разрабатываемый программный модуль системы «Интернет магазин книг» может быть использован в различных интернет-магазинах книг. Он позволит автоматизировать основные бизнес-процессы, связанные с работой интернет-магазина, и повысить эффективность его работы.

Целью курсовой работы является разработка программного модуля системы «Интернет магазин книг».

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области;

- провести анализ готовых решений;
- разработать логическую модель данных и проект модуля;
- реализовать программный модуль системы;
- провести тестирование программного модуля.

Объектом работы является процесс продажи книг.

Предметом работы является продажа книг через интернет-магазин.

В рамках курсовой работы будут рассмотрены авторизация пользователей, включая авторизацию пользователей с разным уровнем доступа, обработка заказов, поступающих в базу данных из внешнего сервиса, управление книгами в системе, включая их редактирование и добавление, а также просмотр действий других пользователей в системе при соответствующем уровне доступа.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Особенности интернет-магазинов

Традиционные магазины книг представляют собой особое пространство, обладающее своей уникальной атмосферой и характеристиками. Их расположение обычно выбирается в центре города или в крупных торговых центрах, обеспечивая удобный доступ для посетителей. Дизайн магазина направлен на создание приятного визуального восприятия и привлечение внимания.

Одним из плюсов традиционных книжных магазинов является возможность физически присутствовать и выбирать. Посетитель может физически прикоснуться к книге до покупки и прочесть то, что ему интересно. Но за это владельцам такого магазина приходится платить кассирам и за помещение, что является лишними тратами.

Также магазины книг в помещениях могут продавать не только книги, но и канцелярские предметы и другого рода вещи, необходимые в школе или творчестве.

В свою очередь интернет-магазины обеспечивают удобство онлайн-покупок с широким ассортиментом, доставкой на дом и возможностью ознакомления с отзывами других покупателей. Однако, в интернете, покупатель теряет возможность личного взаимодействия с продавцом и физического ощущения книг перед покупкой.

Основное различие заключается в способе взаимодействия с продуктом и окружающей обстановкой. Традиционные магазины книг стремятся создать уникальный опыт, в то время как интернет-магазины сосредотачиваются на удобстве и скорости онлайн-покупок.

Важными аспектами в сфере онлайн-торговли книгами являются несколько ключевых факторов. Прежде всего, это обширный каталог товаров, организованный в четкие категории, чтобы обеспечить удобство выбора и навигации для покупателей.

Особое внимание также уделяется созданию интуитивных интерфейсов, использованию продвинутых систем фильтрации и рекомендаций для обеспечения удобства процесса покупки и навигации по виртуальному магазину.

Интернет-магазины книг обязательно предлагают интеграцию методов доставки и обработки заказов для обоих форматов – электронных и физических книг. Это включает в себя системы, способные обеспечить безопасность онлайн-платежей, а также эффективную мобильную совместимость и аналитические инструменты для оценки и улучшения производительности интернет-магазина.

Этот обзор обеспечивает основополагающий взгляд на технические и функциональные аспекты, которые будут учтены при разработке программного модуля для системы "Интернет магазин книг".

1.2 Анализ существующих решений

Существует множество интернет-магазинов книг, каждый из которых предлагает уникальные решения и подходы.

Amazon Books — это крупнейший онлайн-ритейлер, предоставляющий огромный каталог книг в физическом и электронном форматах. Специализируется не только на продаже книг, но и на других товарах. Система рекомендаций Amazon позволяет предлагать пользователям товары, основанные на их предыдущих покупках и интересах.

Book Depository — онлайн-магазин, известный бесплатной международной доставкой. Специализируется на продаже книг, включая широкий выбор заграничных изданий. Привлекает внимание покупателей, ищущих разнообразие и доступность мировой литературы.

AbeBooks выделяется как платформа для поиска и покупки редких, антикварных и коллекционных книг. Этот магазин предоставляет возможность находить уникальные издания, которые могут быть трудно найти в других местах.

Google Play Books является частью экосистемы Google и предоставляет широкий выбор электронных книг и аудиокниг. Интегрируется с другими

сервисами Google, что обеспечивает удобство чтения на различных устройствах, включая смартфоны и планшеты.

ЛитРес — российский интернет-магазин, специализирующийся на продаже электронных книг. Предоставляет возможность скачивания и чтения книг в электронном формате. Также предлагает подписку для доступа к библиотеке книг за фиксированную плату. Фокусируется на российской аудитории, предоставляя широкий выбор российских и зарубежных произведений.

Каждый из рассмотренных интернет-магазинов книг имеет свои преимущества и недостатки. Amazon Books и Book Depository предлагают широкий ассортимент книг по низким ценам, но доставка в некоторые страны может быть платной. AbeBooks предлагает широкий ассортимент книг, включая редкие и коллекционные экземпляры, но время доставки может быть длительным. Google Play Books предлагает широкий ассортимент электронных книг и возможность слушать аудиокниги, но не всегда доступны книги в печатном формате. ЛитРес предлагает широкий ассортимент электронных книг и возможность слушать аудиокниги, а также доступ к книгам на иностранных языках, но стоимость подписки может быть высокой. (см. Приложение 1)

1.3 Постановка задач

Программный модуль системы "Интернет магазин книг" должен включать:

1.3.1 Функции управления каталогом

Управление каталогом должно включать в себя возможность добавления новых книг в систему. Это предполагает введение основной информации о книге, такой как название, автор, жанр, цена и наличие на складе. Дополнительно, должен предоставляться функционал редактирования информации о существующих книгах, а также фильтрации списка книг для удобства пользователей, позволяя им выбирать книги в соответствии с интересующими жанрами.

1.3.2 Функции обработки заказов

Обработка заказов должна включать в себя сохранение информации о заказах и управление ими. Это подразумевает хранение деталей заказов, таких как перечень книг, количество экземпляров, стоимость и статус. Администратору предоставляется функционал изменения статуса заказа, что упрощает отслеживание и обработку заказов в системе.

1.3.3 Функции просмотра действий пользователей

Просмотр действий пользователей должен предоставляться администратору. А также отслеживание действий, таких как добавление, редактирование книг или изменение статуса заказов. Эта функциональность обеспечивает контроль за происходящим в системе и может быть полезной для обнаружения проблем или необычных событий.

1.3.4 Функции безопасности

Функции безопасности включают в себя механизмы аутентификации и авторизации. Они обеспечивают безопасный доступ администраторов к системе, предотвращая несанкционированный доступ. Это также включает в себя защиту конфиденциальных данных, таких как информация о пользователях, заказах и других важных аспектах системы. Аутентификация подтверждает легитимность пользователя, а авторизация определяет его права доступа в системе. Эти меры гарантируют, что только уполномоченные пользователи могут взаимодействовать с критическими частями приложения.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ МОДУЛЯ СИСТЕМЫ

2.1 Выбор и описание программных инструментов

Для написания десктопных приложений существует множество популярных языков программирования, каждый из которых имеет свои преимущества, особенности и недостатки, которые не очевидны при беглом просмотре.

Для написания приложения подходит C# (C Sharp). Этот язык, разработанный Microsoft, часто используется для создания десктопных приложений с использованием технологии .NET. Он предоставляет удобный синтаксис, богатые библиотеки и интеграцию с различными инструментами разработки. Лёгок в освоении, а также имеет большое сообщество программистов, которое привлекает в том числе обилие фреймворков. Ко всему этому следует добавить, что платформа .NET поддерживает кроссплатформенность приложений, а также включает ASP.NET, позволяющий работать с веб-приложениями.

Из множества фреймворков подходит .NET MAUI с Blazor. Он предоставляет инструментарий для создания кроссплатформенных мобильных, настольных и веб-приложений с использованием языка программирования C# и платформы .NET. Сочетание .NET MAUI с Blazor позволяет использовать привычные технологии для создания веб-приложений и переносить их на мобильные и настольные платформы.

Синтаксис представляет собой HTML разметку с использованием «@», когда необходимо обратиться к переменным кода или использовать условные и циклические структуры.

Также поддерживает «модели», что позволяет удобным образом создавать формы заполнения, устанавливая условия для полей и сообщения об ошибках, которые увидит пользователь при несоблюдении условий.

Для работы с базой данных используется EntityFramework. Он используется для работы с данными в приложениях на платформе .NET и

предоставляет набор инструментов для работы с базами данных, абстрагируя слой доступа к данным и позволяя разработчикам взаимодействовать с базой данных с использованием объектно-ориентированной модели данных.

В качестве сред разработки были выбраны Visual Studio 2022 и JetBrains Rider. Каждая из сред имеет свои преимущества и недостатки. Visual Studio помогает писать простые и однотипные участки кода при помощи IntelliSense. Но несмотря на то, что Rider не имеет подобного, он намного удобнее при написании кода, а также отладке и изучении.

2.2 Требования к функциям, выполняемым системой

Функции управления каталогом включают в себя возможность добавления новых книг с указанием основной информации, такой как название, автор, жанр, цена и количество экземпляров. Также предусмотрена опция редактирования информации о уже имеющихся в каталоге книгах. Для облегчения поиска и выбора пользователю предоставляется функция фильтрации списка книг по жанрам.

Функции обработки заказов включают в себя хранение информации о заказах и возможность управления ими, включая изменение статуса заказов. Это обеспечивает систематизацию и эффективное ведение учета заказов в системе.

Функции просмотра действий пользователей предоставляют администратору системы возможность отслеживать действия других пользователей в системе. Эта функция обеспечивает контроль и мониторинг активности, что может быть полезным для анализа и обеспечения безопасности системы.

Функции безопасности включают в себя механизмы аутентификации и авторизации, обеспечивая безопасный доступ администраторов к системе. Эти меры гарантируют, что только уполномоченные пользователи могут вносить изменения в каталог, обрабатывать заказы и просматривать действия других пользователей, что важно для поддержания целостности и конфиденциальности информации.

В приложении должны быть 2 строго разделённых модуля:

- **Classes.** Содержит классы, с которыми ведётся информационная работа.
- **BookDesktop.** Приложение с классами, которые его описывают.

2.3 Модуль Classes

Рассмотрим диаграмму классов (см. рис.1), которые помогают вести информационную работу.

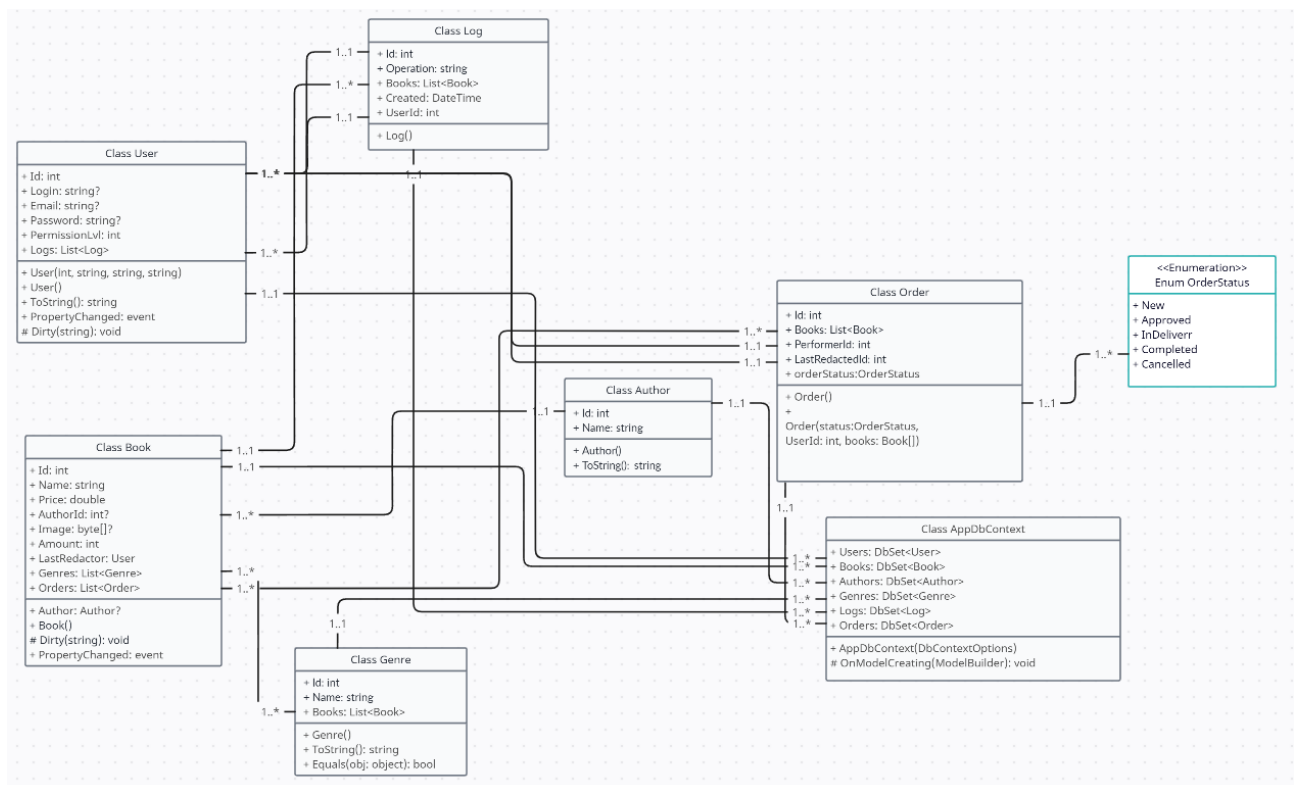


Рисунок1 - Диаграмма классов

AppDbContext, класс реализующий работу с БД, хранится в этом же модуле для удобства работы с ним. Имеет ссылки на все остальные классы. Через EntityFramework создаёт базу данных с таблицами и связями, которые соответствуют таковым в коде модуля.

Почти все классы в модуле являются описанием таблиц, но для некоторых прописаны и переопределены методы для удобства работы с ними в приложении.

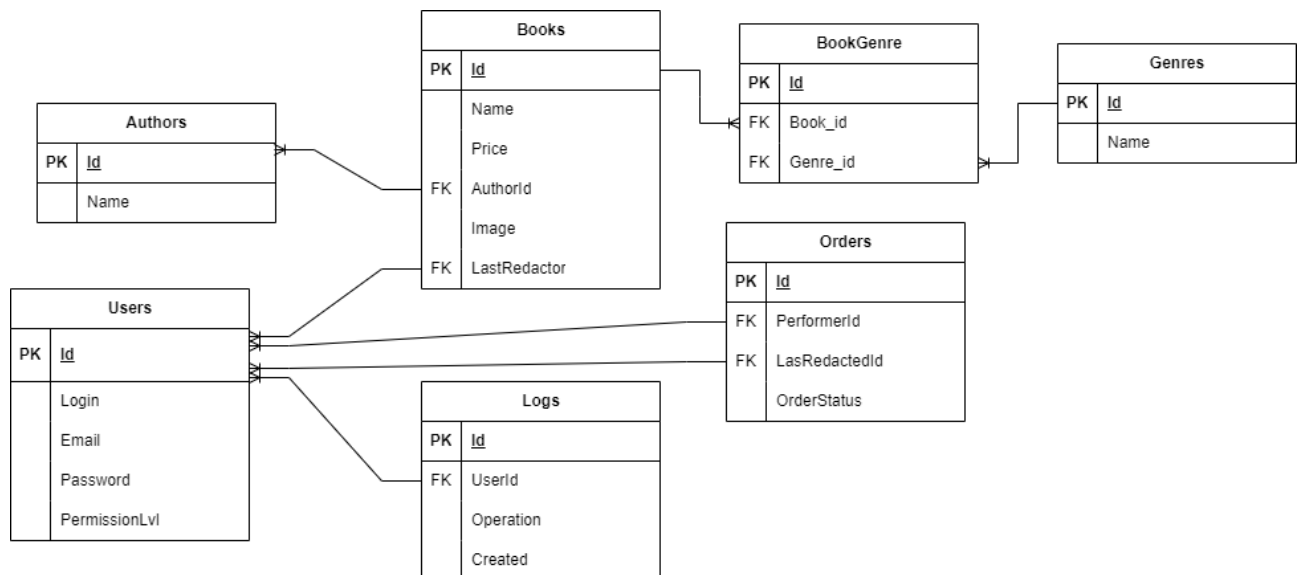


Рисунок 2 – ER-диаграмма базы данных

На рисунке 2 изображена диаграмма базы данных, которую можно использовать при разработке модуля.

2.4 Модуль BookDesktop

Этот модуль написан на основе .NET MAUI фреймворка. При запуске открывается веб-приложение, которое является отдельным экземпляром браузера, который установлен в системе по умолчанию.

- Благодаря этому приложение само по себе является кроссплатформенным с проблемой исключительно в вёрстке.
- Также это позволит легко перейти на браузерную работу приложения при необходимости.

Связь с модулем Classes реализована через ссылку на сборку в модуле и директивы using там, где необходима прямая работа с классами первого модуля.

Класс AppDbContext здесь устанавливается как сервис. Для удобства проектирования приложения использовалась база данных SQLite, которая является файловой БД, т. е. хранится в одном файле вместе с описанием и самими данными, что удобно с небольшими базами или при отладке. Благодаря использованию EntityFramework, приложение можно быстро перенести на PostgreSQL.

2.5 Структура приложения

Функционал, который предоставляет интерфейс, ограничивается в зависимости от уровня прав пользователя (см. рис. 3).

Предусмотрено 3 уровня доступа:

- 0 Root - полный доступ. Предназначен для администратора системы.
- 1 Администратор. Предназначен для администратора занимающимся исключительно аналитикой. Не обладает правами на редактирование.
- 2 Сотрудник. Предназначен для сотрудника занимающимся каталогизацией книг и обработкой заказов.

Все действия логируются и доступны пользователям с доступом к аналитике.

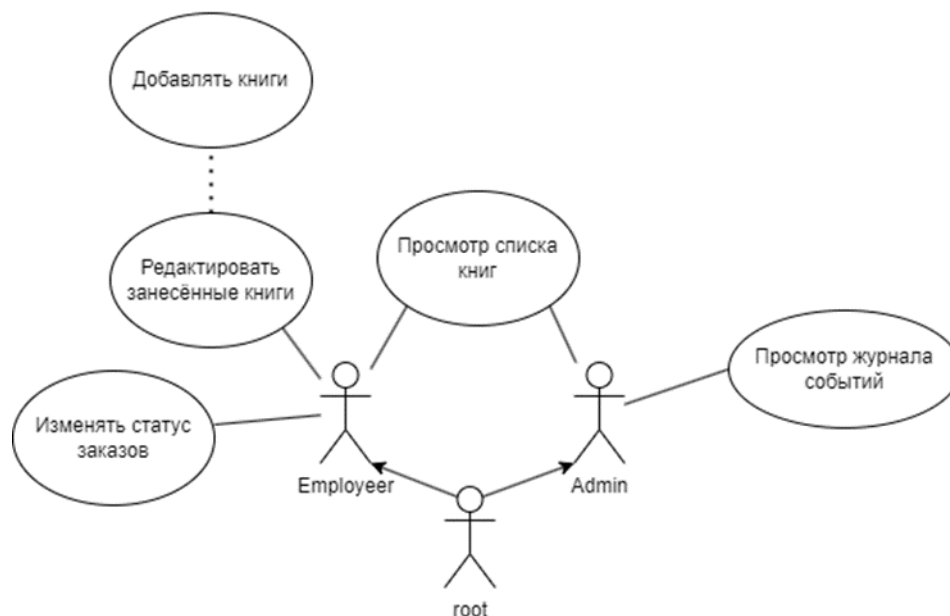


Рисунок 3 - Диаграмма прецедентов

2.6 Структура интерфейса приложения

В первую очередь при запуске приложения пользователь попадает на страницу авторизации. После ввода и подтверждения программа запоминает авторизованного пользователя, как объект класса User из модуля Classes.

Пользователю с соответствующими правами будут доступны:

- Главная страница
- Каталог

- Логи
- Заказы

2.6.1 Главная страница

На этой странице можно посмотреть основную информацию о пользователе, от имени которого производятся действия в программе. На этой же странице кнопка выхода, если есть необходимость ей воспользоваться.

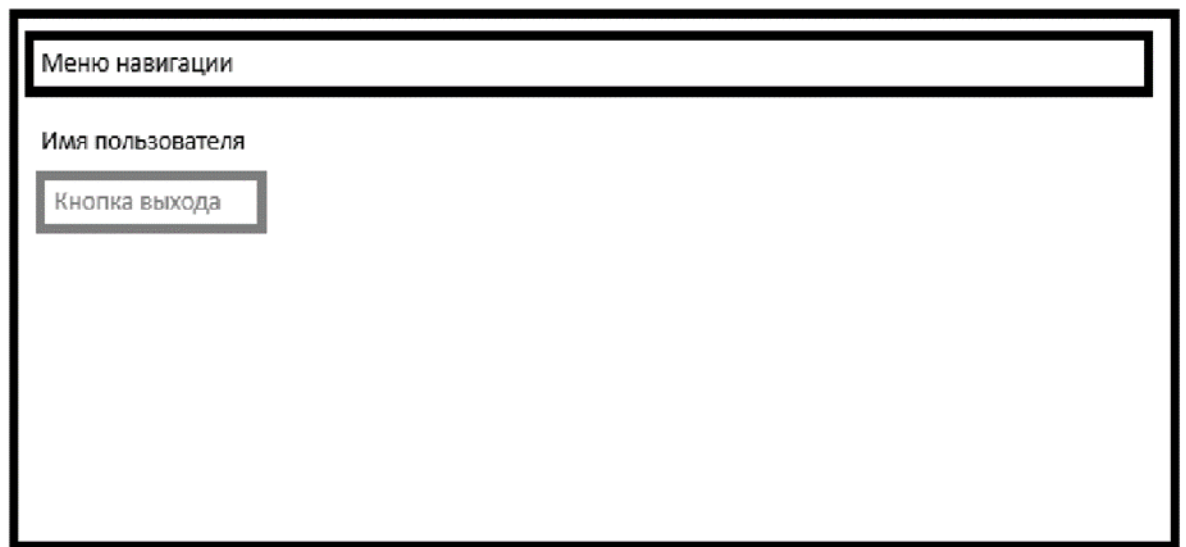


Рисунок 4 – Набросок интерфейса главной страницы

2.6.2 Каталог

Одна из основных страниц. Отображает все книги, которые есть в базе данных и основную информацию о них в виде списка. Позволяет изменить определённую книгу, добавить новую или отфильтровать по жанрам.

Функции редактирования и добавления не доступны для администратора(аналитика).

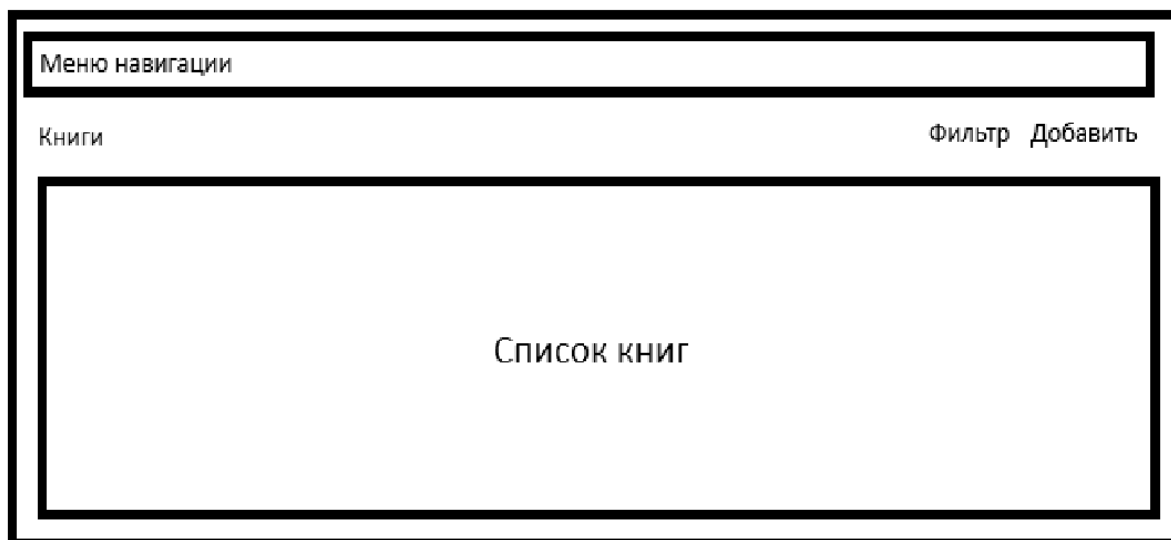


Рисунок 5 – Набросок окна каталога

2.6.3 Логи

Страница аналитики. Отображает все действия, которые записаны системой в виде списка. Записываются только изменения, которые повлияли на данные в базе данных.

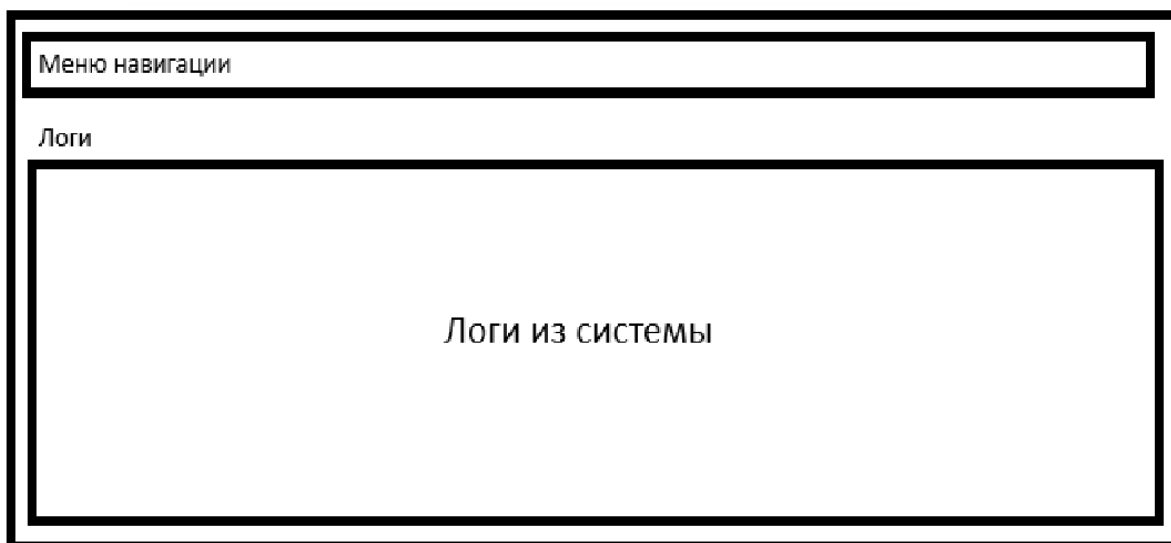


Рисунок 6 – Набросок окна логов

2.6.4 Заказы

Вторая по важности страница. Представляет все заказы, которые есть в БД в виде таблицы из карт. Позволяет изменять статус заказа.

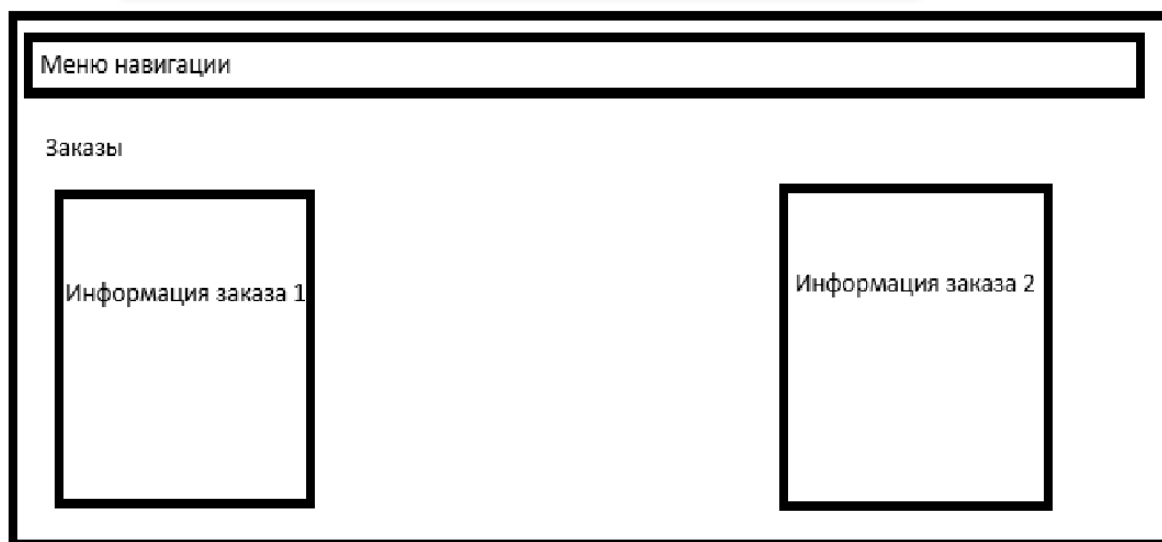


Рисунок 7 – набросок окна заказов

2.7 План тестирования

Разработан план тестирования, покрывающий некоторые сценарии программы (см. Приложение 2).

ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОЕКТА МОДУЛЯ СИСТЕМЫ

3.1 Описание кодом функциональных узлов модуля BookDesktop

3.1.1 Старт программы и подготовка

Программа начинает своё выполнение с класса MauiProgram, где приложения собирается. Производится добавление сервисов и их подготовка (см. Рисунок 8).

```
builder.Services.AddSingleton<AuthService>();
builder.Services.AddScoped<BookChangeService>();
builder.Services.AddDbContext<AppDbContext>(optionsAction: (options) =>
{
    options.UseSqlite(connectionString: $"Data Source={Environment.CurrentDirectory + @"\MyDatabase.db"}");
});
builder.Services.AddSingleton<BinService>();
var app = builder.Build();
var dbContext = app.Services.GetRequiredService<AppDbContext>();
dbContext.Database.EnsureCreated();
```

Рисунок 8 – MauiProgram, инсталляция сервисов

В представленном участке кода явно видно, что используется база данных SQLite, но при необходимости можно подключаться к PostgreSQL, которая является более профессиональным и производительным решением.

Также в этом файле база данных заполняется тестовыми данными при запуске приложения в отладочном режиме и отсутствии базы (см. Рисунок 9).

Главная страница Каталог Логин Заказы				
Книги		Фильтр		Добавить
Наименование	Автор	Стоимость	Экземпляры	Действия
Как на самом деле работают компьютеры. Практическое руководство по внутреннему устройству машины	Адам Фримен	0	130	✓
Параллельное программирование на современном C++	Мэттью Джастис	200	20	✓
Эффективный C. Профессиональное программирование	Райнер Гримм	3000	30	✓
Алгоритмы и программы. Язык C++ : учебное пособие для СПО	Роберт С. Сикорд	400	40	✓
Изучаем C# через разработку игр на Unity	Елена Александровна Конова	500	50	✓
Изучаем C#	Дженнифер Грин	600	60	✓
Паттерны проектирования для C# и платформы .NET Core	Гаурав Арора	700	70	✓
Экстремальный Си. Параллелизм, ООП и продвинутые возможности	Камран Амини	800	80	✓
Обратные вызовы в C++	Виталий Евгеньевич Ткаченко	900	90	✓

Рисунок 9 - Содержимое таблицы Books после удаления базы данных

После завершения подготовительных работ инициализируется интерфейс и появляется окно программы со страницей авторизации.

3.1.2 Окно авторизации

Представляет собой форму с двумя полями. Проверка введённых пользователем значений производится благодаря использованию модели `Person`, в которой через атрибуты указаны условия и выводимые ошибки.

При корректности данных программа обращается к сервису для авторизации. Сервис возвращает булево значение в зависимости от того, авторизован пользователь или нет. Если нет, то поля ввода очищаются.

```
private void Submit()
{
    if (Person.Login != "" && Person.Password != "")
    {
        if (!authService.Login(navigation, Person.Login!, Person.Password!))
        {
            Person.Login = String.Empty;
            Person.Password = String.Empty;
            IsInvalid = true;
            InvokeAsync(StateHasChanged);
        }
    }
}
```

Рисунок 10 – Код формы для авторизации

Также при неправильных данных авторизации пользователь получает уведомление о неправильных данных авторизации (см. Рисунок 11), что соответствует реакции программы при неправильных данных (см. Приложение 2).

```
@if (IsInvalid)
{
    <div class="d-flex border border-warning">
        <div class="p-2 w-100">
            <p>Некорректные данные авторизации</p>
        </div>
        <div class="p-2 flex-shrink-1">
            <button type="button" class="btn btn-warning" onClick="@(() => { IsInvalid = false; InvokeAsync(StateHasChanged); })">
                Закрыть
            </button>
        </div>
    </div>
}
```

Рисунок 11 – Уведомление о некорректном вводе данных авторизации

Авторизация

Логин

Пароль

➔ Авторизоваться

Рисунок 12 – Окно авторизации

3.1.3 Каталог

Страница с перечислением всех книг в базе данных. Рядом на строке с каждой книгой есть кнопка для изменения. Ниже часть кода, отвечающая за обработку действий пользователя.

Также, начиная с этой страницы, используется компонент для навигации, чтобы автоматически отслеживать активную страницу и корректно отображать доступные страницы.

Для реализации фильтра используется функция, которая запрашивает данные из базы, используя выбранные пользователем жанры.

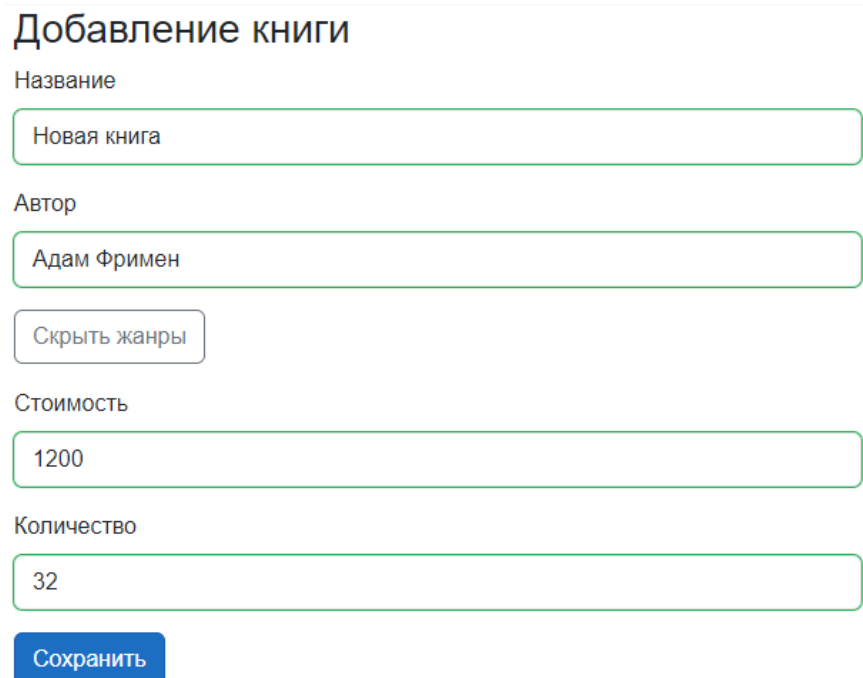
```
async Task<List<Book>> GetBooksWithFilter()
{
    if (selectedGenres.Count == 0)
        return books;

    var filteredBooks :List<Book> = await dbcontext.Books // DbSet<Book>
        .Include( navigationPropertyPath: x :Book => x.Author) // IQueryable<Book, Author?>
        .Where(book => book.Genres.Any(bg :Genre => selectedGenres.Contains(bg))) // IQueryable<Book>
        .ToListAsync(); // Task<List<...>>

    return filteredBooks;
}
```

Рисунок 13 – Функция выборки книг по выбранным жанрам

Для изменения книги или добавления используются разные страницы, но отличаются они не сильно по своему функционалу. Их задача отобразить форму (см. Рисунок 14) для пользователя и проконтролировать введенные значения, что выполняется при попытке пользователем сохранить книгу.



Добавление книги

Название

Новая книга

Автор

Адам Фримен

Скрыть жанры

Стоимость

1200

Количество

32

Сохранить

Рисунок 14 – Окно изменения/добавления книги

Рассмотрим подробнее работу этой функции. Сначала функция убирает лишние пробелы при помощи регулярного выражения, а после проверяет введенные значения на наличие пустой строки или отрицательных значений (см. Рисунок 15). Если были ошибки при написании, то выполнение идёт по ветке true и пользователь получает уведомление о некорректном вводе согласно плану тестирования (см .Приложение 2).

```
bookLocal.Name = Regex.Replace(input: bookLocal.Name, pattern: @"\s+", replacement: " ");
if (bookLocal.Name == "" || bookLocal.Price < 0 || bookLocal.Amount < 0)
{
    MessageError = "Ошибка при сохранении";
    HasError = true;
    CanEscape = false;
    InvokeAsync(StateHasChanged);
}
else
```

Рисунок 15 - Функция сохранения изменённой книги ветка true

Если ошибок не было, то программа идёт по ветке false (см. Рисунок 16) и обрабатывает добавление новой книги, также заносит запись в базу об изменении/добавлении книги.

```
else
{
    HasError = false;
    CanEscape = true;

    bookLocal.Genres.Clear();
    foreach (var id in checkedGenres.Where(x => x.Value).Select(x => x.Key))
    {
        bookLocal.Genres.Add(item: genres.First(x => x.Id == id));
    }

    bookLocal.LastRedactor = authService.CurrentUser!;

    dbContext.Update(bookLocal);
    dbContext.Logs.Add(entity: new Log
    {
        Operation = $"Внесены изменения в книгу с Id = {bookLocal.Id}",
        UserId = authService.CurrentUser!.Id,
    });
    await dbContext.SaveChangesAsync();
    navigation.NavigateTo(uri: "books");
}
```

Рисунок 16 – Функция сохранения изменённой книги ветка else

Также в ходе тестирования добавления и удаления пользователей была обнаружена ошибка, из-за которой программа аварийно завершала работу. Это происходило из-за попытки EntityFramework обработать автора книги как новый объект, а не уже существующий.

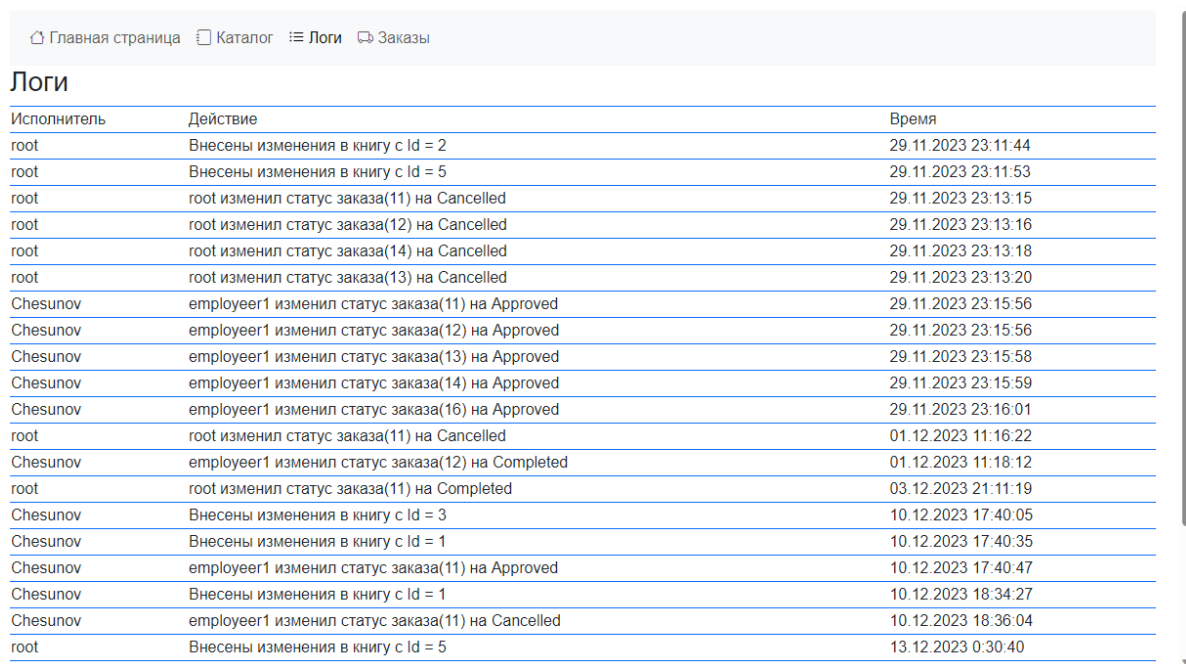
Если нужно будет занести книгу, автора которого нет в базе, то придётся делать запрос напрямую в базу данных. Аналогичная ситуация с жанрами. Сделано это так потому что такие процессы как добавление автора, жанра или издательства задача другого рода и не подходит под определённые при проектировании задачи, а также выходит за рамки разработки модуля.

3.1.4 Страница просмотра логов

На этой странице создаётся список из всех записей таблицы Logs, которые есть в базе данных. Записи в эту таблицу вносятся при любом значимом изменении (см. Рисунок 17).

При изменении вносятся записи следующих форматов:

- Заказы: (пользователь) изменил статус заказа(id заказа) на (статус заказа)
- Книги(изменение): Внесены изменения в книгу с Id = (id книги)
- Книги(добавление): Добавлена книга с именем = (название книги)



Исполнитель	Действие	Время
root	Внесены изменения в книгу с Id = 2	29.11.2023 23:11:44
root	Внесены изменения в книгу с Id = 5	29.11.2023 23:11:53
root	root изменил статус заказа(11) на Cancelled	29.11.2023 23:13:15
root	root изменил статус заказа(12) на Cancelled	29.11.2023 23:13:16
root	root изменил статус заказа(14) на Cancelled	29.11.2023 23:13:18
root	root изменил статус заказа(13) на Cancelled	29.11.2023 23:13:20
Chesunov	employee1 изменил статус заказа(11) на Approved	29.11.2023 23:15:56
Chesunov	employee1 изменил статус заказа(12) на Approved	29.11.2023 23:15:56
Chesunov	employee1 изменил статус заказа(13) на Approved	29.11.2023 23:15:58
Chesunov	employee1 изменил статус заказа(14) на Approved	29.11.2023 23:15:59
Chesunov	employee1 изменил статус заказа(16) на Approved	29.11.2023 23:16:01
root	root изменил статус заказа(11) на Cancelled	01.12.2023 11:16:22
Chesunov	employee1 изменил статус заказа(12) на Completed	01.12.2023 11:18:12
root	root изменил статус заказа(11) на Completed	03.12.2023 21:11:19
Chesunov	Внесены изменения в книгу с Id = 3	10.12.2023 17:40:05
Chesunov	Внесены изменения в книгу с Id = 1	10.12.2023 17:40:35
Chesunov	employee1 изменил статус заказа(11) на Approved	10.12.2023 17:40:47
Chesunov	Внесены изменения в книгу с Id = 1	10.12.2023 18:34:27
Chesunov	employee1 изменил статус заказа(11) на Cancelled	10.12.2023 18:36:04
root	Внесены изменения в книгу с Id = 5	13.12.2023 0:30:40

Рисунок 17 – Страница логов

3.1.5 Страница просмотра заказов

На этой странице отображаются все заказы, которые занесены в систему (см. Рисунок 18). Создать заказ нельзя, так как подразумевается, что это происходит во внешнем сервисе или приложении. На этой стороне происходит лишь обработка статуса. Также заказ не имеет возможности «транзакций». То есть при статусе «Выполнен» в системе не будет изменяться количество оставшихся экземпляров книг. При полноценной структуре системы у сотрудников не должна быть возможность напрямую изменить статус заказа на «Выполнен» или «Отменить». Это должно проводиться через запрос клиента в специальной форме. Здесь есть такая функция, чтобы показать рабочий функционал, а также факт учёта такого статуса в системе (см. Рисунок 19).

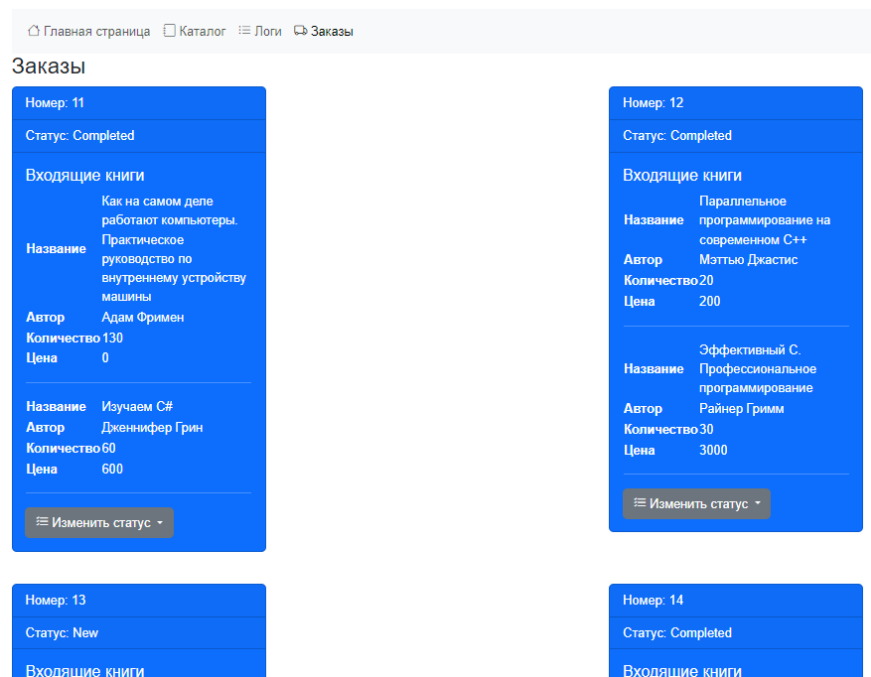


Рисунок 18 – Окно просмотра заказов

```
public void ChangeOrderStatus(Order order, OrderStatus status)
{
    order.orderStatus = status;
    dbcontext.Logs.Add(entity: new Log
    {
        Operation = $"{AuthService.CurrentUser.Login} изменил статус заказа({order.Id}) на {status}",
        UserId = AuthService.CurrentUser.Id,
    });
    dbcontext.SaveChanges();
    InvokeAsync(StateHasChanged);
}
```

Рисунок 19 – Обработка изменения статуса заказа

3.2 Описание кодом функциональных узлов модуля Classes

Состоит из 7 классов (6 функциональных, 1 служебного).

3.2.1 Работа с БД в контексте класса AppDbContext

Этот класс служебный и выполняет роль посредника между кодом и СУБД, т. е. является интерфейсом. Через свойства DbSet указаны классы и названия таблиц, которые должен создать EntityFramework, если их не будет в базе данных.

В ходе разработки потребовалось создание таблиц многие ко многим, поэтому через отношения это было прописано отдельно (см. Рисунок 20).

EntityFramework сам регулирует их и прямых операций со сводными таблицами не требуется, поэтому они не указаны как одно из свойств типа DbSet.

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Book>() // EntityTypeBuilder<Book>
        .HasMany(navigationExpression: b:Book => b.Genres) // CollectionNavigationBuilder<Book,Genre>
        .WithMany(navigationExpression: bg:Genre => bg.Books);

    modelBuilder.Entity<Order>() // EntityTypeBuilder<Order>
        .HasMany(navigationExpression: o:Order => o.Books) // CollectionNavigationBuilder<Order,Book>
        .WithMany(navigationExpression: b:Book => b.Orders);

    modelBuilder.Entity<User>() // EntityTypeBuilder<User>
        .HasMany(navigationExpression: x:User => x.Logs) // CollectionNavigationBuilder<User,Log>
        .WithOne(navigationExpression: b:Log => b.User);
}
```

Рисунок 20 – Отношения многие-ко-многим

3.2.2 Функциональные классы

Класс Author. Простой, но нужный класс для соответствия базы данных третьей нормальной форме, а также упрощает операции добавления и изменения книг.

Класс Book. Хранит в себе основную информацию о книгах такую как название, цену, автора, изображение в виде бинарного массива, количество экземпляров, список жанров, который контролируется через EntityFramework, как список заказов, а также пользователя, который последним редактировал конкретную книгу.

Класс Genre. Вспомогательный для Book, так как представляет собой жанры. Также хранит список книг, которые можно получить при необходимости через EntityFramework. Это позволяет узнать сколько книг есть у каждого жанра простым запросом.

Класс Log. Представляет записи о действиях пользователей. Хранит текстовое описание операции, дату создания записи и пользователя.

Класс Order. Для этого класса было создано перечисление, чтобы удобно работать со статусами заказов. Хранит список книг, пользователя, которому назначен заказ (если такое необходимо), пользователя, который последний раз редактировал этот заказ, а также конкретный элемент из перечисления.

Класс User. Самый важный класс, который заставляет работать половину приложения (не считая служебный AppDbContext). Содержит логин, почту, пароль и уровень доступа пользователя в качестве целочисленного значения.

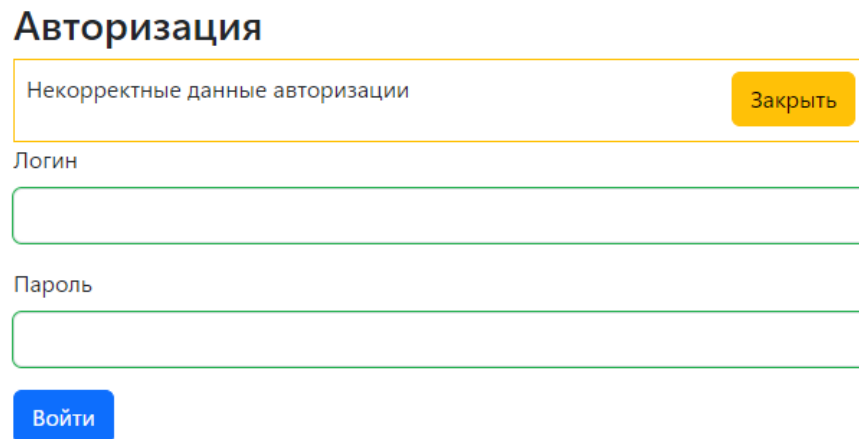
Также была переопределена стандартная функция ToString для удобства работы с доступами.

3.3 Тестирование приложения

Протестируем программу по плану тестирования (см. Приложение 2).

3.3.1 Тестирование окна авторизации

При вводе неверных данных пользователь должен увидеть уведомление о некорректном вводе (см. Рисунок 21).

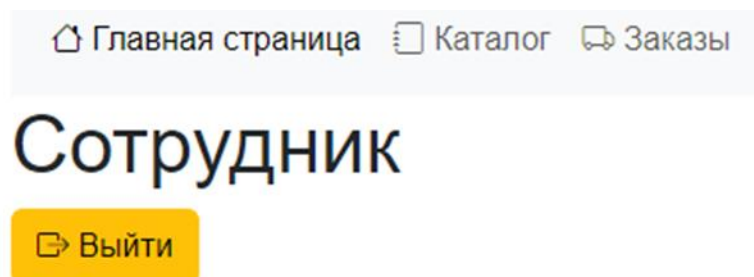


The screenshot shows a login form titled "Авторизация". At the top, there is a yellow error message box that says "Некорректные данные авторизации" with a "Закрыть" button. Below this, there are two input fields: "Логин" and "Пароль". At the bottom, there is a blue "Войти" button.

Рисунок 21 – Тестирование. Авторизация: некорректный ввод

Результат: отображение сообщения и очистка полей авторизации.

При вводе верных данных авторизации (существующего пользователя) пользователя должно перенаправить на главную страницу. На главной странице должна быть написана группа пользователя (см. Рисунок 22).



The screenshot shows the main page of the application. At the top, there is a navigation bar with links: "Главная страница", "Каталог", and "Заказы". Below this, the word "Сотрудник" is displayed in a large font. At the bottom, there is a yellow button with a right arrow icon and the text "Выйти".

Рисунок 22 – Успешная авторизация от пользователя Kalidsky

3.3.2 Тестирование окна Каталога

Рассмотрим 2 варианта. Выбор нескольких жанров (см. Рисунок 24) и пустой список (см. Рисунок 23).

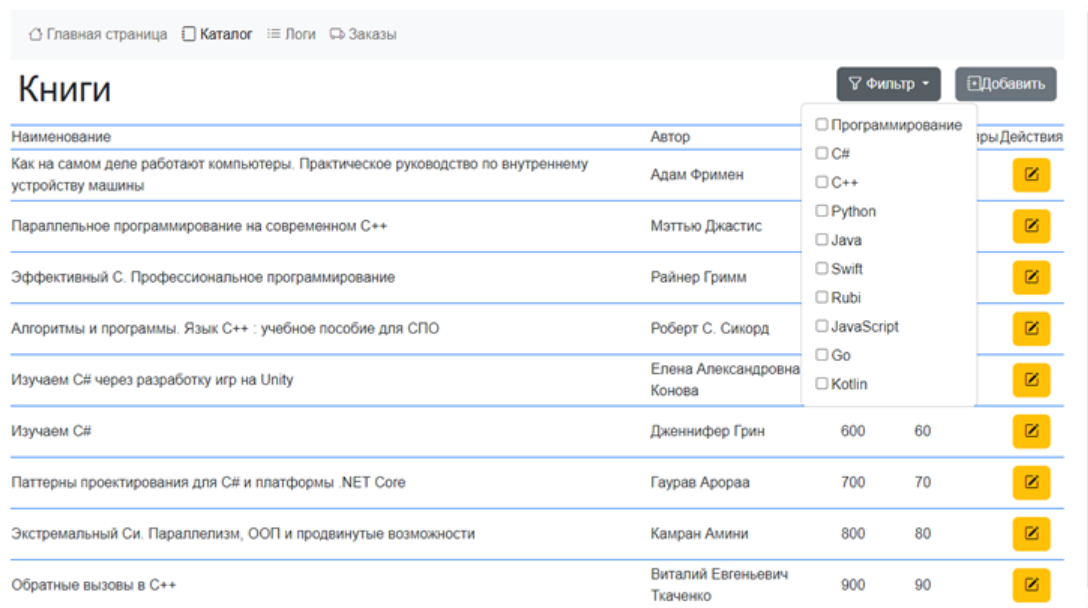


Рисунок 23 – Тестирование. Каталог: несколько жанров

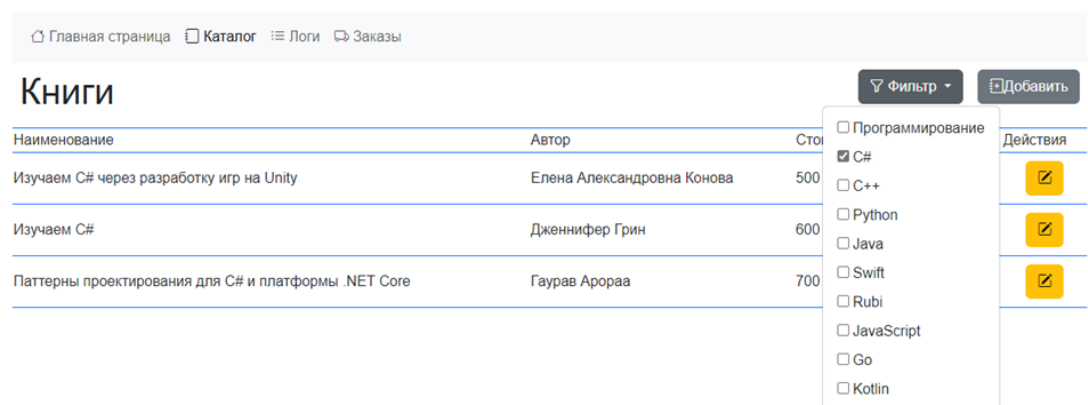


Рисунок 24 – Тестирование. Каталог: без жанров

Стоит отметить, что выбираются книги по принципу ANY, то есть наличие хотя бы 1 из выбранных жанров.

3.3.3 Тестирование изменения статуса заказов

Изменим статус заказа под номером 12 (см. Рисунок 25) на «Выполнен» (см. Рисунок 26).

Номер: 12

Статус: InDeliver

Рисунок 25 – Заказ 12 до выполнения изменения статуса

Главная страницаКаталогЛогиЗаказы

Заказы

Номер: 11

Статус: Completed

Входящие книги

Как на самом деле работают компьютеры. Практическое руководство по внутреннему устройству машины

Название

Адам Фримен

Автор

130

Количество

0

Цена

Изучаем C#

Название

Дженнифер Грин

Автор

60

Количество

600

Цена

Изменить статус

Номер: 12

Статус: Completed

Входящие книги

Параллельное программирование на современном C++

Название

Мэттью Джастис

Автор

20

Количество

200

Цена

Эффективный C. Профессиональное программирование

Название

Райнер Гримм

Автор

30

Количество

3000

Цена

Изменить статус

Рисунок 26 – Изменение статуса заказа

3.3.4 Просмотр окна логов и тестирование логирования системы

При тестировании заказов система записала действие изменения статуса заказа (см. Рисунок 27).

root	Внесены изменения в книгу с Id = 3	16.12.2023 19:08:32
root	Внесены изменения в книгу с Id = 4	16.12.2023 19:08:41
root	Внесены изменения в книгу с Id = 5	16.12.2023 19:08:48
root	Внесены изменения в книгу с Id = 6	16.12.2023 19:08:54
root	Внесены изменения в книгу с Id = 7	16.12.2023 19:09:03
root	Внесены изменения в книгу с Id = 8	16.12.2023 19:09:11
root	Внесены изменения в книгу с Id = 9	16.12.2023 19:09:19
root	root изменил статус заказа(12) на Completed	16.12.2023 19:13:57

Рисунок 27 – Просмотр логов приложения

ЗАКЛЮЧЕНИЕ

Разработанный программный модуль системы «Интернет магазин книг» является решением для автоматизации внутренних бизнес-процессов интернет-магазина книг. Он позволяет обеспечить удобную и эффективную работу интернет-магазина, повысить его конкурентоспособность и привлечь новых клиентов.

Реализация этого модуля позволяет интернет-магазину книг повысить эффективность работы сотрудников и обработки заказов. Это поможет сократить расходы и время сотрудников в таких процессах как каталогизация книг, их изменение и добавление, обработка заказов. За действиями сотрудников смогут наблюдать менеджеры и администраторы магазинов, что поможет при анализе работы сотрудников и потребностях клиентов.

Разработка такого программного модуля системы «Интернет магазин книг» является актуальной задачей ввиду множества площадок и магазинов, а также перспективной ввиду их расширения.

Исходя из вышесказанного модуль является эффективным решением для автоматизации внутренних бизнес-процессов интернет-магазина книг, так как экономит время и ресурсы организации/площадки.

Также модуль был протестирован и в ходе тестирования были исправлены критические ошибки, которые могли нарушить работу модуля.

На основе полученных результатов работы сделаны следующие выводы:

Есть необходимость в разработке и интеграции дополнительных функций для программного модуля. В частности возможность регистрации новых сотрудников, так как сейчас это возможно только посредством прямого запроса в базу данных.

Улучшение интерфейса работы с заказами, а именно создать и организовать процедуры работы с заказами, а также добавить возможность поиска и фильтрации заказов.

Упрощение работы с книгами: добавить поиск по названию и автору книг в каталоге, в том числе возможность фильтровать по количеству и цене.

Следует также рассмотреть возможность интеграции модуля с другими системами интернет-магазина.

Большим дополнением будет создание и организация клиентского интерфейса с оформлением заказов и соответствующими процедурами.

Для исправления невыявленных ошибок и оптимизации приложения необходимо провести анализ эффективности работы разработанного программного модуля в реальных условиях.

Реализация этих вопросов позволит улучшить функциональность программного модуля и повысить его эффективность.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Нормативно-правовые источники

1. ГОСТ 7.32–2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления
2. ГОСТ 7.1-2003. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления
3. ГОСТ 7.9—95 Система стандартов по информации, библиотечному и издательскому делу. Реферат и аннотация. Общие требования
4. ГОСТ 7.11—2004 (ИСО 832:1994) Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Сокращение слов и словосочетаний на иностранных европейских языках
5. ГОСТ 7.12—93 Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила
6. ГОСТ 7.80—2000 Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Заголовок. Общие требования и правила составления
7. ГОСТ 7.82—2001 Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления
8. ГОСТ Р 7.0.97-2016. Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу. Организационно-распорядительная документация. Требования к оформлению документов

Учебники, учебные пособия, статьи

9. Эндрю Троелсен, Филипп Джепикс Язык программирования C# 7 и платформы .NET и .NET Core. Санкт-Петербург. Издательство «Диалектика», 2019.

10. Стивен Клири Конкурентность в C#. Асинхронное, параллельное и многопоточное программирование. Санкт-Петербург. Издательство «Питер», 2020.

11. Эндрю Стилмен, Дженнифер Грин Изучаем C#. Санкт-Петербург. Издательство «Питер», 2022.

12. Сусликов А.А. Программирование на C#. Москва. Издательство "БХВ-Петербург", 2020.

Интернет-источники

13. BillWagner Отношения между типами в операциях запросов LINQ (C#). URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/linq/get-started/type-relationships-in-linq-query-operations>

14. Дениэл Рот, Дефф Фриц, Тейлор Саусвик, Скотт Эдди, Стив Смит Blazor для разработчиков ASP.NET Web Forms. URL: <https://learn.microsoft.com/ru-ru/dotnet/architecture/blazor-for-web-forms-developers/>

15. Rick-Anderson, ErikEJ, gewarren, roji, smitpatel Getting Started with EF Core. URL: <https://learn.microsoft.com/en-us/ef/core/get-started/overview/first-app?tabs=netcore-cli>

16. Bootstrap team Документация по использованию компонентов Bootstrap. URL: <https://getbootstrap.com/docs/5.3/getting-tarted/introduction/>

Таблица 1 Анализ существующих решений

Характеристика	Amazon Books	Book Depository	AbeBooks	Google Play Books	ЛитРес
Ассортимент книг	Широкий	Широкий	Широкий	Широкий	Широкий
Цены	Низкие	Низкие	Средние	Средние	Средние
Доставка	Быстрая	Бесплатная по всему миру	Длительная	Быстрая	Быстрая
Электронные книги	Да	Да	Да	Да	Да
Аудиокниги	Да	Нет	Нет	Да	Нет

Приложение 2
План тестирования

No	Наименование функциональности	Наименование поля	Тестовый набор	Результат (должно получиться)	Результат тестирования
1	Редактирование, добавление книги	Название	Введены текстовые данные (кириллица и латинские символы)	Согласен с разработчиком	Сообщение: «Ошибка сохранения книги»
		Цена	Введены цифры меньше 0	Согласен с разработчиком	Сообщение: «Ошибка сохранения книги»
		Количество	Введены цифры меньше 0	Согласен с разработчиком	Сообщение: «Ошибка сохранения книги»
		Список жанров	Ничего не выбрано	Согласен с разработчиком	Создание книги без жанров
		Название	Проверка на ограничение введенных символов (>=64, >=255)	Согласен с разработчиком введенных данных	Создание книги с длинным названием

		Название	Все пробелы	Согласен с разработчиком	Сообщение: «Ошибка сохранения книги»
		Название	Введены текстовые данные с пробелами в начале и середине слова	Согласен с разработчиком	Книга добавлена, пробелы нормализованы
		-	Добавление книг с одинаковыми данными	Ошибка: «Книга уже добавлена»	Добавлена новая книга с существующими значениями.
2	Авторизация	Логин и пароль	Ввод любых не подходящих значений	Согласен с разработчиком	Ошибка: «Неверные данные авторизации»
		Логин и пароль	Правильные данные авторизации	Согласен с разработчиком	Пользователь авторизован
3	Фильтр книг	Список жанров	Пустой список. Ничего не выбрано	Согласен с разработчиком	Отображается весь список.

		Список жанров	Выбрано несколько жанров	Согласен с разработчиком	Отобраны книги, в которых есть выбранные жанры
--	--	---------------	--------------------------	--------------------------	--