МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Специализация Программирование интернет-приложений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Тема Программное средство «WorldOfBook»

Исполнитель

Студент 2 курса группы 6 Алисевич Даниил Сергеевич

(Ф.И.О.)

Руководитель ассистент Радиванович Д.А.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Пацей Н. В.

(подпись)

Минск 2021

**Содержание**

[**Введение** 3](#_Toc9857374)

[**1. Аналитический обзор литературы** 4](#_Toc9857375)

[1.1. Анализ прототипов 4](#_Toc9857376)

[1.2. Актуальность решаемой задачи 8](#_Toc9857379)

[**2.** **Моделирование предметной области и разработка функциональных требований** 9](#_Toc9857381)

[2.1 Описание инструментов разработки программного средства 9](#_Toc9857382)

[2.2 Описание функций, реализуемых программой 10](#_Toc9857383)

[**3. Проектирование программного средства** 14](#_Toc9857384)

[3.1 Проектирование классов 14](#_Toc9857388)

[3.2 Размещение данных в программном проекте 16](#_Toc9857389)

[3.3 Проектирование базы данных 17](#_Toc9857390)

[**4.** **Создание программного средства** 19](#_Toc9857392)

[**5.** **Тестирование программного средства** 21](#_Toc9857393)

[**Список использованных источников** 24](#_Toc9857394)

[**Приложение А** 25](#_Toc9857395)

**Введение**

Книга – это один из важнейших способов передачи знаний. На протяжении многих веков своего существования она играла важнейшую роль в жизни людей. И если когда-то книги были роскошью, доступной лишь немногим (отчасти из-за своей дороговизны, а отчасти из-за недоступности образования), то сегодня книги доступны каждому.

В последнее время люди всё чаще предпочитают бумажным книгам электронные. Такая тенденция, на мой взгляд, продиктована экономической целесообразностью и удобством. Одна электронная книга значительно дешевле сотен бумажных, но может вместить в себя тысячи томов. Сегодня, когда большинство людей уже не мыслят своей жизни без различных гаджетов (смартфонов, планшетов, ридеров и ноутбуков), читать электронные книги удобнее, чем носить с собой книги традиционные.

Программное средство «WorldOfBook» сможет сделать процесс чтения книг более простым и удобным. Ведь, как было сказано ранее, – намного проще читать книги в электронном виде, нежели носить с собой множество тяжёлых бумажных изданий.

Программное средство «WorldOfBook» реализует работу с книгами. «WorldOfBook» имеет несколько режимов доступа: пользователь и администратор. Функциями приложения являются: ведение базы данных пользователей и, соответственно, книг, просматривание информации о книгах, добавление их в раздел «мои книги», блокировка пользователей, изменение личных данных пользователями, есть возможность добавлять новые книги (добавлять может только администратор.), чтение книг формата pdf.

Целью курсового проекта является разработка программного обеспечения для удобства чтения книг, связанная с ведением базы данных.

Базой данных называется набор логически связанных данных, который предназначен для удовлетворения информационных потребностей организации/предприятия. Для выполнения задания, поставленного в курсовой работе, необходимо освоить работу по проектированию баз данных и работу с базами данных в выбранной среде программирования.

В качестве интерфейса прикладного программирования был выбран обширный API-интерфейс — Windows Presentation Foundation (WPF), предназначенный для создания настольных программ с графически насыщенным пользовательским интерфейсом.

Для работы с WPF использовался объектно-ориентированный язык программирования с С-подобным синтаксисом — С#, разработанный для создания приложений на платформе Microsoft .NET Framework.

Кроме того, используемая в WPF модель разделения кода и дизайна предполагала работу с декларативным языком описания интерфейса — eXtensible Application Markup Language (XAML).

**1. Аналитический обзор литературы**

* 1. Анализ прототипов

В настоящий момент существует множество приложений-«читалок», как называют их в повседневной жизни пользователи. Сейчас бы хотелось рассмотреть некоторые из них. Однако будет проводиться анализ только десктоп-приложений, как наиболее близких к теме моего курсового проекта.

* «Cool Reader»

«Cool Reader» на данный момент является одной из наиболее популярных и распространённых программ для чтения. Поддерживаются такие форматы, как .fb2, .txt и .doc, .epub и .rtf, а также веб-страницы. Также к плюсам этого приложения можно отнести возможность регулирования яркости и чтение книг прямо из архива, не распаковывая его. К минусам я же хочу отнести функцию автоперелистывания страниц, которая не всегда работает корректно (иногда приложение может перевернуть одну страницу, иногда – все тридцать или же вообще «перебросить» пользователя на последнюю, заключительную страницу).

Хотелось бы отметить, что данное приложение постоянно обновляется и с каждым обновлением становиться всё более пригодным для комфортного чтения.

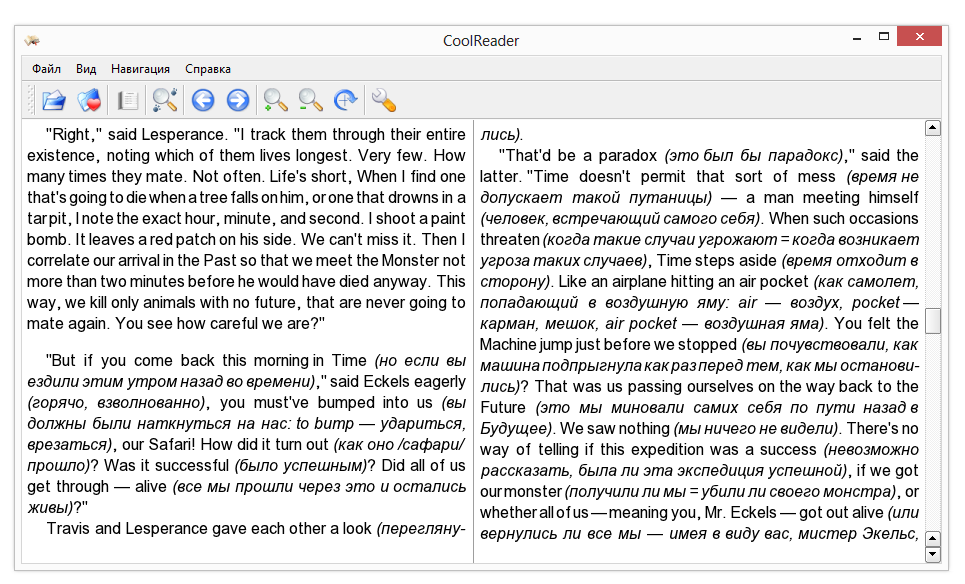


Рисунок 1.1 – Интерфейс приложения «Cool Reader» в 2013 году

Я считаю, что, несмотря на множество достоинств данного приложения, большим минусом является его дизайн, ведь новый пользователь не знает о всех «фичах» приложения, а оценивает его исключительно с точки зрения презентабельности. На данный момент приложение выглядит следующим образом:

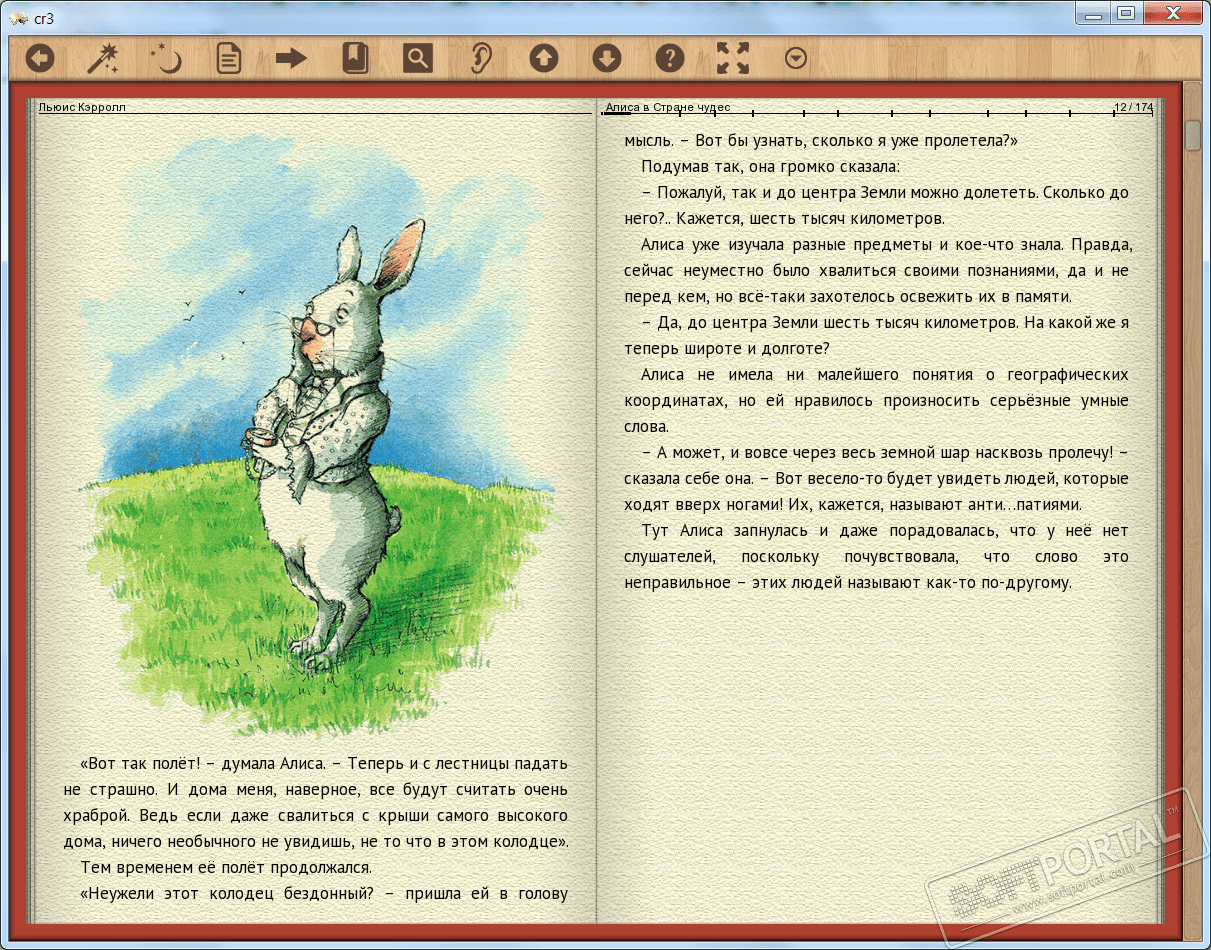


Рисунок 1.1 – Интерфейс приложения «Cool Reader» в 2021 году

* «ALReader»

Также среди популярных приложений можно найти и «ALReader».

Данное приложение подходит для чтения большинства книг, так как поддерживает такие форматы, как fb2, fbz, txt, epub, html, doc, docx, odt, rtf, mobi, tcr, prc, а также огромное количество различных функций: настройка стилей отображения (заголовки, цитаты, аннотации и т.д.), маркирование текста, наличие закладок, возможность отправки выделенного текста в сторонние программные, навигация, создание ярлыков для книги на домашнем экране, редактирование fb2 и txt файлов, одно- и двухстраничный режим просмотра текста, поддерживание сносок на странице в форматах fb2 и doc/docx. Возможность настройки приложения индивидуально под каждого пользователя не может не стать огромным плюсом «ALReader».

По своему внешнему виду программа напоминает книгу, напечатанную на газетной бумаге, что не может не порадовать пользователя, ведь при так чтение становиться более комфортным.

В качестве ещё одного плюса данного приложения хочется отметить возможность его работы даже без установки.

Однако и у этого приложения есть один большой минус: при перелистывании страниц периодически происходит непроизвольное затемнение страницы, что очень мешает чтению.

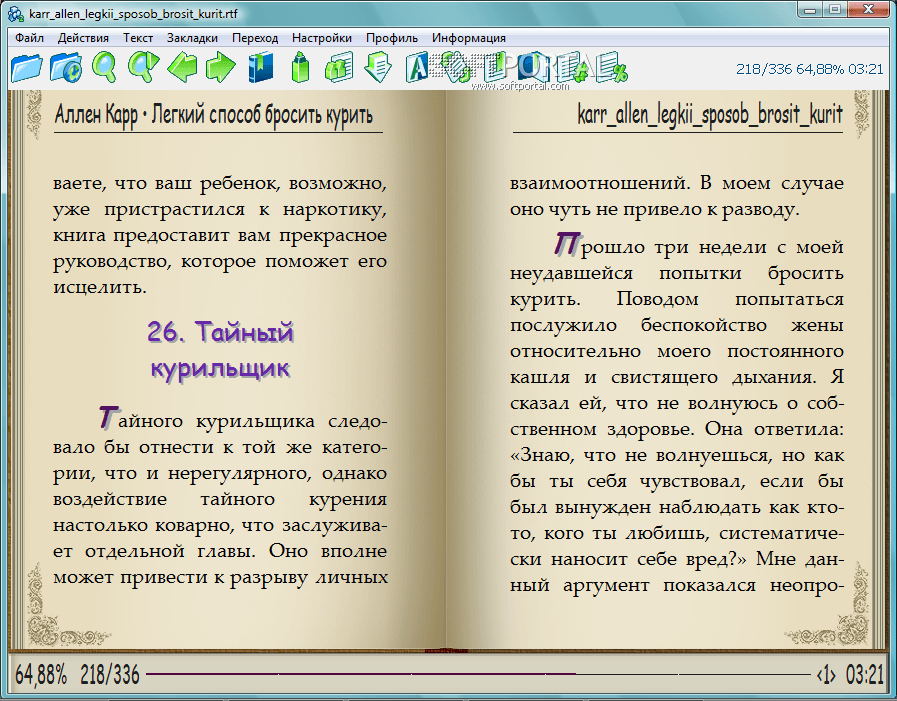


Рисунок 1.3 – Интерфейс приложения «ALReader»

* «FBReader»

«FBReader» – это кроссплатформенная программа для чтения электронных книг, благодаря которой можно ознакомиться с литературой, записанной в большинстве популярных форматов, а также легко настроить процесс чтение индивидуально под свои требования.

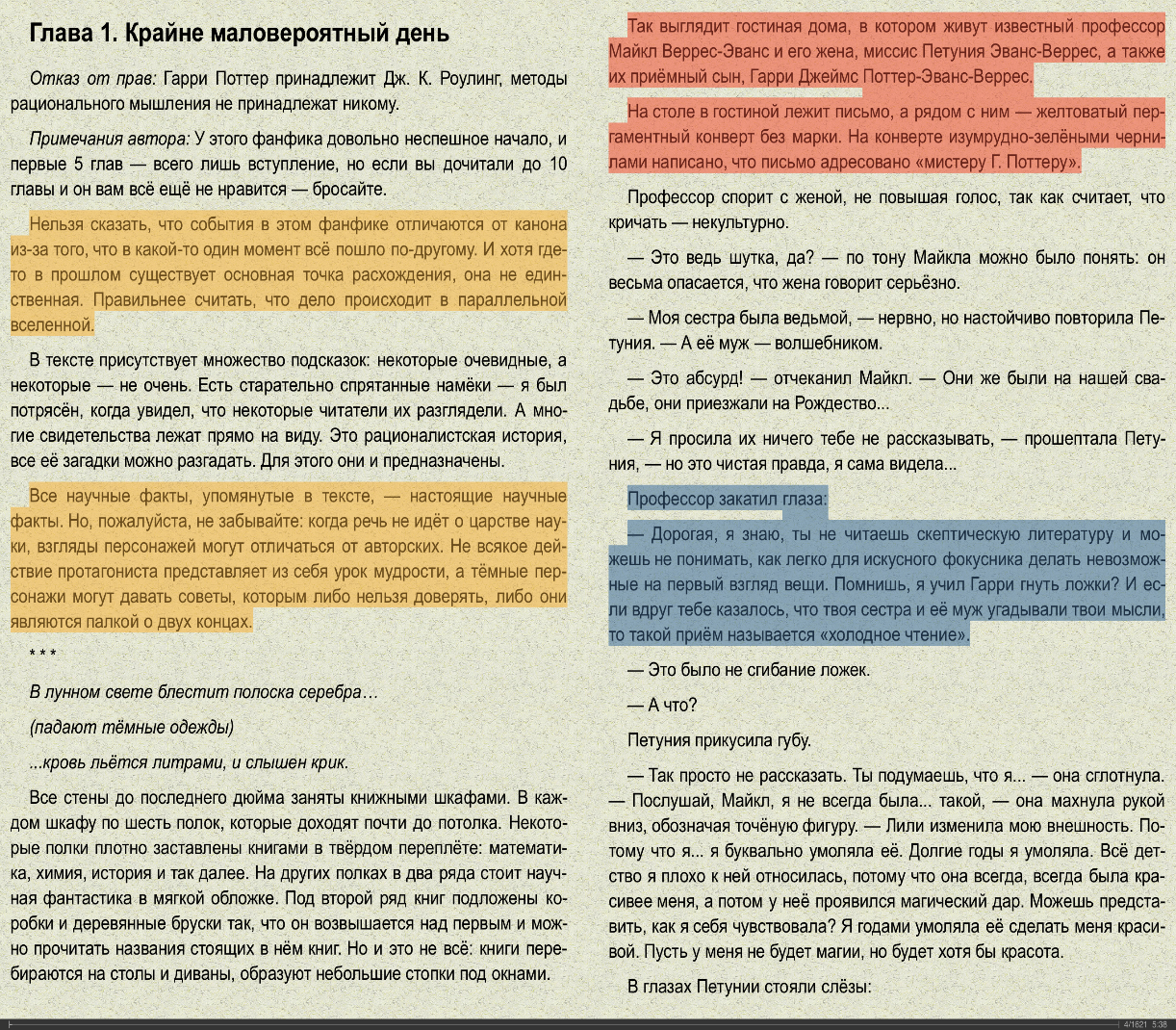


Рисунок 1.4 – Интерфейс приложения «FBReader»

Процесс настройки достаточно простой, интуитивно понятный, что является неотъемлемым плюсом данного приложения. Книги здесь сортируются по своим характеристикам: названию, жанру или же автору.

Также стоит отметить, что «FBReader» самостоятельно создаёт ссылки на местоположение добавленных книг в компьютере, т.е. пользователь нет необходимости самостоятельно перемещать все книги в общую папку.

Неизменным минусом является дизайн данного приложения.

* Веб-сервис «Google Play Книги»

На сайте популярного сервиса «Google Play Книги» можно добавлять и читать тексты в режиме онлайн. Вместе с тем у проекта есть приложение для ChromeOS, позволяющее читать загруженные книги на компьютере даже без подключения к сети. Интерфейс приложения почти полностью повторяет дизайн веб-версии. Есть возможность открывать книги из своей библиотеки, просматривать их содержание, выполнять поиск по тексту, настраивать шрифты и разметку. Чтобы читать офлайн, нужно предварительно загрузить нужные книги в память компьютера.

Проанализировав представленные выше программные средства, можно сформулировать базовые функциональные требования к моему курсовому проекту:

* программное средство должно представлять библиотеку, т.е. набор книг, из которых пользователь может выбрать желаемые;
* программное средство должно предоставлять пользователю о его собственных книгах, т.е. тех книгах, которые он выбрал из библиотеки;
* программное средство предоставляет возможность чтения данных книг, т.е. книг пользователя;

1. 2. Актуальность решаемой задачи

В наше время, век информационных технологий, появилась необходимость информатизировать бумажные носители на электронные. Книги же являются наиболее ярким примером необходимости перевода информации с таких [бумажных] носителей на цифровые.

Приложения, подобные программному средству «WorldOfBook», никогда не перестанут быть актуальными, так как люди не перестанут читать. Уверен, что ничто не способно заменить книги. Любой человек не сможет обходиться без процесса чтения как такового. В книгах сконцентрирован опыт поколений. Многие знания, заключённые в них, записанные много веков назад, и сегодня не утратили своей актуальности.



# **Моделирование предметной области и разработка функциональных требований**

## 2.1. Описание инструментов разработки программного средства

При разработке приложения были использованы:

* интегрированная среда разработки Microsoft Visual Studio 2019;
* программная платформа .NET Framework 4.7.2;
* язык программирования C#;
* расширяемый язык разметки XAML;
* технология WPF;
* Entity Framework;
* Material Design Library;
* iText Library;

*C#* — объектно-ориентированный язык программирования. Разработан в 1998 — 2001 под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework.

*XAML* — это декларативный язык разметки. С точки зрения модели программирования .NET Framework язык XAML упрощает создание пользовательского интерфейса для приложения .NET Framework.

WPF (Windows Presentation Foundation) — система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

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

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

## 2.2. Описание функций, реализуемых программой

Варианты использования программного средства показаны на диаграмме на рисунке 2.1.

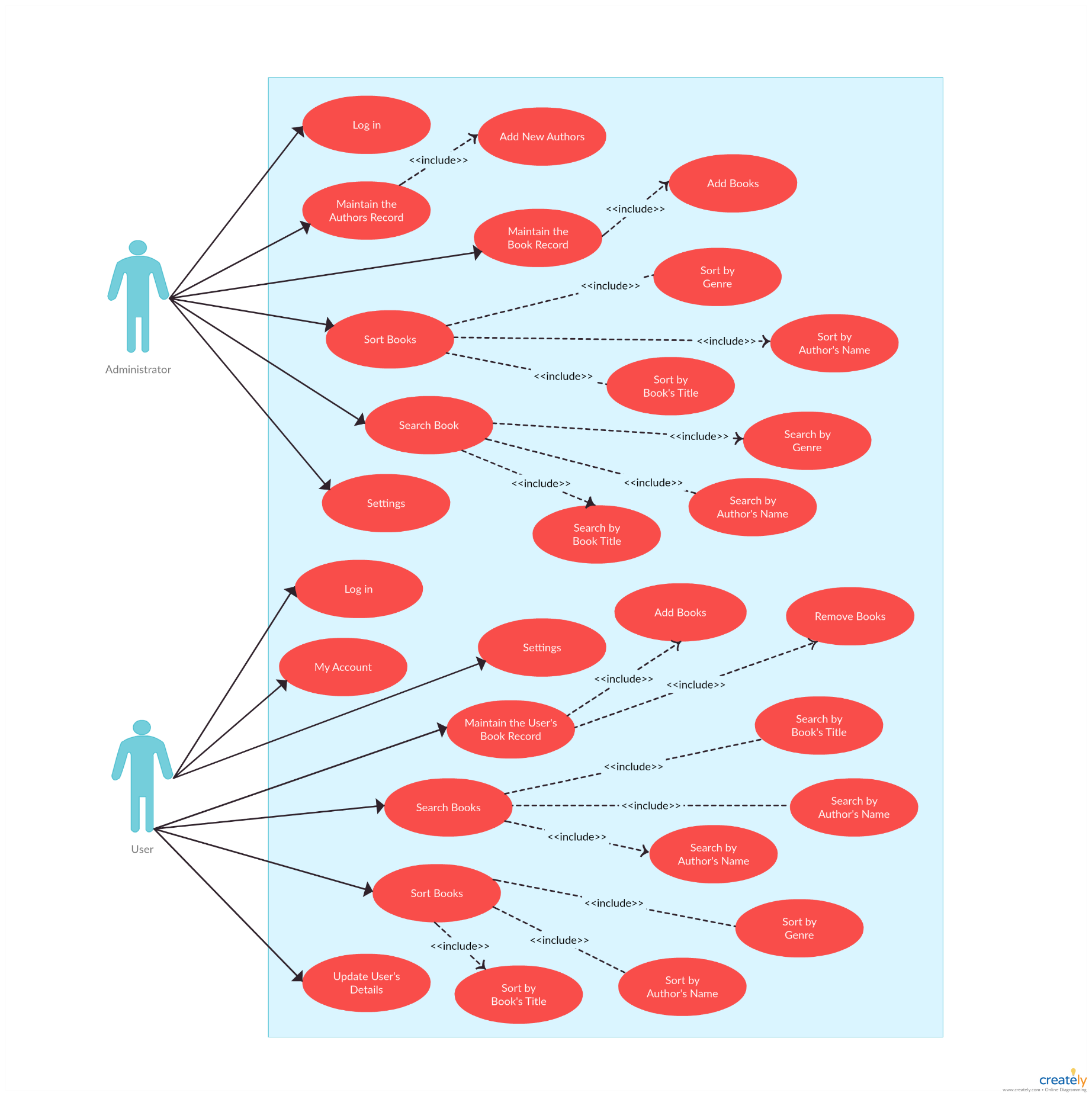
****

Рисунок 2.1 – Диаграмма вариантов использования (UML Case)

Сейчас же хотелось бы рассмотреть более подробно каждый из прецендентов.

Прецедент «Settings» заключается в просмотре информации о программе.

Прецедент «Log in» заключается в авторизации (т.е. выбора роли администратора/пользователя) в случае успешного авторизации в учетную запись.

Если мы зашли как «Администратор»:

Прецедент «Search Books» заключается в просмотре всех книг, находящихся в системе, их поиску по названию, имени автора, а также жанру. Также есть возможность осуществлять поиск книг по году публикации и стране издания.

Прецедент «Add New Book» заключается в добавлении в систему новой книги.

Прецедент «Add New Author» заключается в добавлении в систему нового автора.

Прецедент «Sort Books» заключается в просмотре всех книг, находящихся в системе, их сортировке по названию, имени автора, а также жанру. Также есть возможность осуществлять сортировку книг по году публикации и стране издания.

Прецедент «Settings» заключается в изменении тем.

Если мы зашли как «Пользователь»:

Прецедент «My Account» заключается в открытии окна, в котором находится вся информация о пользователе.

Прецедент «Update User’s Details» заключается в открытии окна, в котором находится вся информация о пользователе c возможностью последующего редактирования.

Прецедент «Search Books» заключается в просмотре всех книг, находящихся в системе, их поиску по названию, имени автора, а также жанру. Также есть возможность осуществлять поиск книг по году публикации и стране издания.

Прецедент «Maintain the User’s Book Record» заключается в просмотре книг пользователя.

Прецедент «Add Books» заключается в добавлении книги в раздел «My Books».

Прецедент «Remove Books» заключается в удалении книги из раздела «My Books».

Прецедент «Sort Books» заключается в просмотре всех книг, находящихся в системе, их сортировке по названию, имени автора, а также жанру. Также есть возможность осуществлять сортировку книг по году публикации и стране издания.

Прецедент «Settings» заключается в изменении тем.

Из всего вышесказанного следует, что программное средство работает следующим образом: изначально запускается главное окно MainWindow, в котором будут диактивированы кнопки списка (кнопка перехода в окно «Мой аккаунт», кнопка перехода в окно «Мои книги», кнопка перехода в окно «Поиск книг»), а кнопка, позволяющая переходить в окно администратора вовсе будет скрыта. Далее необходимо авторизоваться. После авторизации, в зависимости от роли пользователя, ему станет доступен основной функционал, предоставляемый программным средством. Итак, рассмотрим более подробно каждый из сценариев: администратора и пользователя.

* Администратор

Если пользователь вошёл в систему в качестве администратора, то на главном окне MainWindow появляется кнопка администратора, открывающая окно добавления новых книг, становиться доступной кнопка поиска, открывающая окно «Поиск», неактивными остаются кнопки, открывающие окна «Мой аккаунт» и «Мои Книги», так как предполагается, что администратор будет работать, а не читать.

В окне добавления новых книг администратор может добавить новую книгу, её название, год и страну издания, автора, аннотацию, а также выбрать изображение обложки. В этом же окне есть кнопка «Authors», которая позволяет добавить нового автора, его имя, фамилию, год рождения и смерти, небольшое описание, а также фотографию. Нюанс: если администратор добавляет новую книгу и в поле «автор» вписывает ещё не существующего в базе данных автора, то сначала он должен добавить информацию об этом авторе и только после добавить новую книгу с этим, новым автором. В том же окне есть кнопка «User page» которая позволяет просмотреть всех пользователей, при нажатии на пользователя администратор может посмотреть подробную информацию о пользователе, а также блокировать или разблокировать пользователя.

В окне поиска книг мы можем нажать кнопку «Show All», которая отобразит в текущем окне все имеющиеся книги, имеется функция поиска. Искать книги мы можем по названию, жанру, стране и году издания, автору. По этим же критериям мы можем и сортировать имеющиеся книги. Для этого существует кнопка «Sort by». В этом же окне есть кнопка «Authors», которая открывает окно с информацией о всех существующих авторах.

* Пользователь

Зайдя в систему в качестве стандартного пользователя, на базовом окне MainWindow становятся активными три кнопки: «MyAccount», «MyBooks» и «SearchBooks». Работа кнопки «Search» и все последующие действия у пользователя аналогичны действиям с этой же кнопкой у администратора. Нажав на какую-либо из книг в окне «SearchBooks» откроется окно «SelectedBook». В этом окне находится информация о книге. Здесь же пользователь может добавить книгу в «мои книги».

Если же пользователь нажмёт кнопки «MyAccount», расположенную в главном окне MainWindow, то откроется окно, содержащее информацию о самом пользователе, пользователь может изменить имя и фамилию, а также добавить информацию «о себе». В этом же окне есть кнопка «Picture», с помощью которой пользователь сможет выбрать себе «аватарку».

Также пользователю доступна кнопка «MyBooks», при нажатии на которую открывается окно, содержащее все книги данного пользователя. Можно нажать на одну из них. В этом случае откроется окно «ReadyToRead» с более подробной информацией об этой книге. Здесь же пользователь может удалить эту книгу из «моих книг». Также доступна кнопка «Read now». В случае нажатия на эту кнопку открывается окно «PDFReader», в котором можно прочитать выбранную книгу.

# **3. Проектирование программного средства**



## Проектирование классов

Диаграмма классов приложения представлена в **приложении А**.

*Контейнер сущностей­ –*  представляет собой логическое группирование [наборов сущностей](https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/entity-set), [наборов ассоциаций](https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/association-set), и [функции imports](https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/model-declared-function).

Для контейнера сущностей, определенного в концептуальной модели, должны выполняться следующие условия:

* В каждой концептуальной модели должен быть определен по крайней мере один контейнер сущностей.
* Контейнер сущностей должен иметь уникальное имя внутри концептуальной модели.

Контейнер сущностей может определять наборы сущностей или наборы ассоциаций, которые используют типы сущностей или ассоциации, определенные в одном или нескольких пространствах имен.

В связи с этим были созданы такие классы как: MyBooks, SelectedBook, Registration, BookElem, Settings и многие другие, которые хранят методы логического группирования сущностей как получения, добавления, обновления, удаления данных.

Book\_Context – предназначен для установления связи с базой данных.

Author\_Context – предназначен для установления связи с базой данных.

Genre\_Context – предназначен для установления связи с базой данных.

User\_Context – предназначен для установления связи с базой данных.

UserBooks\_Context – предназначен для установления связи с базой данных.

Book – предназначен для создания книги.

Author – предназначен для создания автора.

Genre – предназначен для создания жанра.

User – предназначен для создания пользователей.

UserBooks – предназначен для создания книги.

Также в проекте есть 17 классов окон. Классы окон представлены на рисунке 2.1.

В данных классах представлен весь основной функционал программного средства, включающее в себя добавление новых книг в базу данных, добавление новых авторов, возможность настройки тем, добавление книги пользователем в раздел «мои книги», удаление книг пользователем из раздела «мои книги», возможность чтения представленных книг в формате pdf. Данная функция [чтение в формате pdf] реализована в классе PDFReader с помощью дополнительной библиотеки iText, которая, непосредственно, предоставляет инструменты для работы с данным форматом.

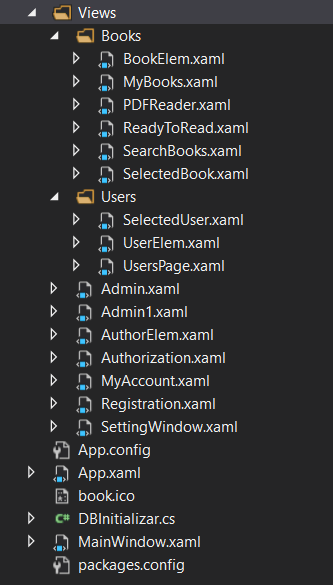


Рисунок. 2.1 – Классы окон

Окно MainWindow незначительно, но меняется в зависимости от роли пользователя. При начальной, первой загрузке, пока ни один пользователь ещё не авторизовался, кнопки перехода в разделы «Мой аккуант», «Мои книги» и «Поиск книг» являются неактивными, а кнопка разработчика вовсе скрыта. Далее, в зависимости от введённых данных в окне «Авторизация», MainWindow активирует кнопки, необходимые для работы с приложением. Если в систему вошёл Администратор, то становиться активной кнопка «Поиск» и видимой кнопка разработчика, кликнув на которую откроется окно добавления новой книги. На том же окне добавления новой книги есть кнопка «авторы», при нажатии на которую Администратору откроется окно добавления нового автора. Так же на том же окне добавления новой книги есть кнопка «UsersPage», при нажатии на которую откроется окно со всеми пользователями, при нажатии на пользователя откроется окно «SelectedUser», где можно увидеть подробную информацию о пользователе, а также блокировать или разблокировать его. Если же в систему вошёл Пользователь, то кнопка разработчика остаётся скрытой, но активируются кнопки «Мой аккаунт», «Мои книги» и «Поиск».

Окна SelectedBook и OpenReadBook служат для отображения информации о выбранной книге. В окне SelectedBook пользователь может добавить книги в раздел «мои книги». В окне OpenReadBook пользователь может открыть книгу из раздела «мои книги», может нажать кнопку «Read Now» для её чтения, а также удалить книгу из раздела «мои книги».

Окно BookElem служит для предварительного просмотра книги, для графического отображения книги в окнах MyBooks и SearchBooks.

Окно UserElem служит для предварительного просмотра книги пользователя, для графического отображения пользователя в окнах SelectedUser.

Окно Authorization служит для авторизации пользователя.

Окно Registration служит для регистрации пользователя.

В целом, названия каждого окна в полной мере отражают их функциональное назначение.

## Размещение данных в программном проекте

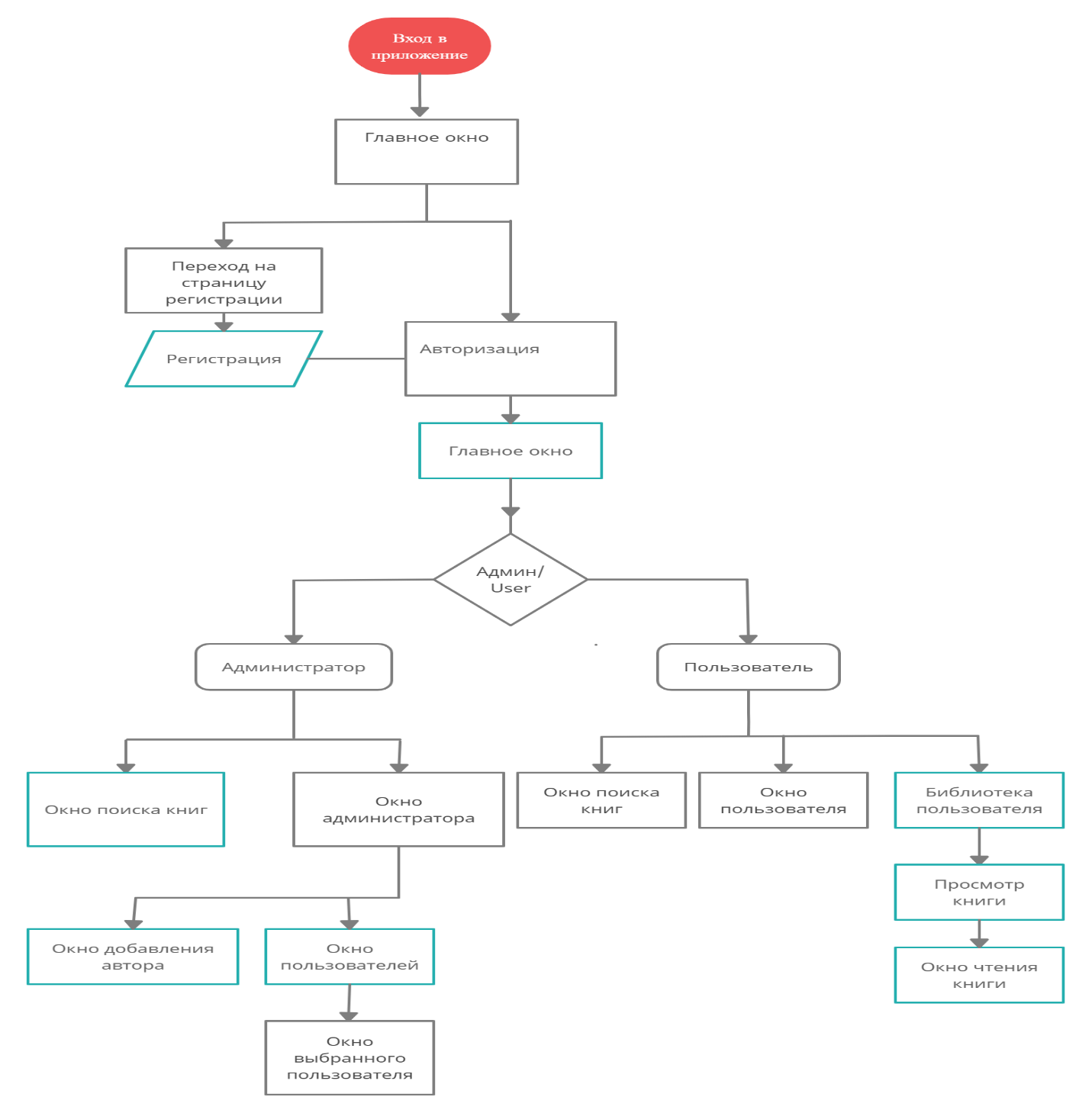


Рисунок. 3.2 – Диаграмма размещений окон

## Проектирование базы данных

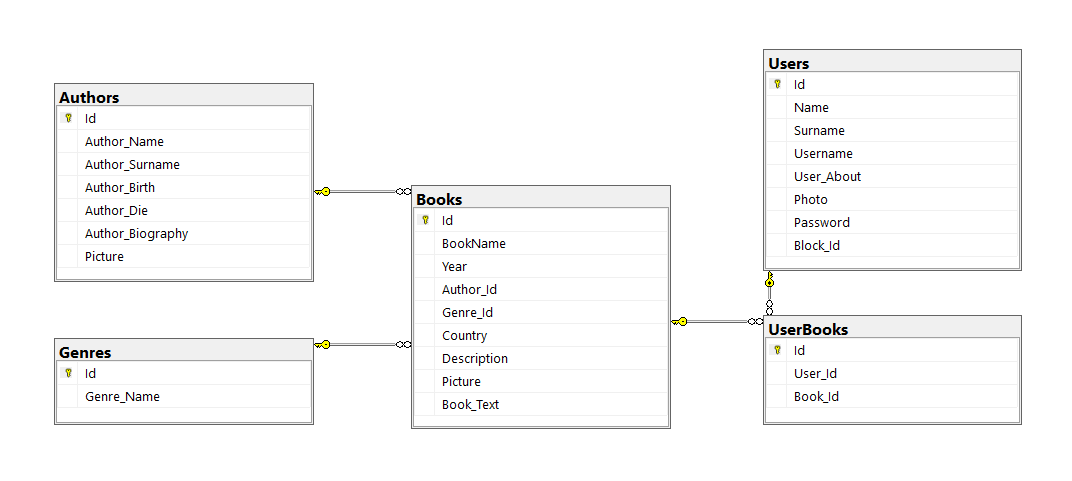


Рис. 3.3 – Диаграмма базы данных

Всего в базе данных 5 таблиц. В таблице Users хранятся все пользователи, зарегистрированные в программе, Books – информация о книгах, Authors – информация о авторах, Genres – информация о жанре, а также таблица UserBooks, с помощью которой реализована связь «многие ко многим» между таблицами Books и Users.

Таблица Users состоит из семи столбцов, иллюстрирующих полную информацию о пользователе:

* Id– хранит идентификатор пользователя;
* Name – хранит имя пользователя;
* Surname – хранит фамилию пользователя;
* Username – хранит псевдоним пользователя;
* User\_About – хранит информацию о пользователе;
* Password – хранит пароль пользователя;
* Photo – хранит фото пользователя (строку-путь, где располагается фото);
* Block\_Id – хранит число блокировки пользователя

Таблица Books состоит из девяти столбцов, иллюстрирующих полную информацию о книге:

* Id - хранит идентификатор книги;
* BookName – хранит имя книги;
* Year – хранит год публикации книги;
* Author\_Id – хранит идентификатор автора;
* Genre\_Id – хранит идентификатор жанра;
* Country – хранит страну публикации книги;
* Description – хранит описание книги;
* Picture – хранит изображение книги (строку-путь на расположение картинки на диске);
* BookText – хранит непосредственно текст книги (строку-путь на расположение pdf файла на диске);

Таблица Authors состоит из семи столбцов, иллюстрирующих полную информацию об авторе:

* Id - хранит идентификатор доктора;
* Author\_Name– хранит имя автора;
* Author\_Surname– хранит фамилию автора;
* Author\_Birth – хранит год рождения автора;
* Author\_Die – хранит год смерти автора;
* Biography – хранит информацию об авторе;
* Picture - хранит фото автора (строку-путь на расположение фото на диске);

Таблица Genres состоит из двух столбцов, содержащих информацию о жанрах:

* Id – хранит идентификатор жанра;
* Genre\_Name – хранит название жанра;

Таблица UserBooks состоит из семи столбцов:

* Id - хранит идентификатор данной таблицы, реализуемой связь «многие ко многим»;
* Book\_Id – хранит идентификатор книги, принадлежащей пользователю;
* User\_Id– хранит идентификатор пользователя;

# **Создание программного средства**

В данной главе будет рассмотрен базовый функционал классов. Для более полного понимания требуется ознакомиться с диаграммами классов (**Приложение А**), а также примером кода некоторых из них [классов] (**Приложение Б**).

В данном программном средстве все классы можно условно разделить на три группы: классы Data\_Context, Models и Views (Windows). Детально рассмотрим все классы из данных групп.

Model классы:

* Book – предназначен для создания книги. Данный класс содержит поля-свойства, хранимые в базе данных: Id – уникальный идентификатор книги, BookName – наименование книги, Author\_Id – идентификатор автора, Year – год издания книги, Country – страну издания книги, Picture – путь к изображению книги, Book\_Text – путь к файлу pdf, Description – аннотацию к книге, Genre\_Id – идентификатор жанра.
* Author – предназначен для создания автора. Класс содержит Id – уникальный идентификатор автора, Author\_Name – имя автора, Author\_Surname – фамилия автора, Author\_Birth – год рождения автора , Author\_Die – год смерти, Country – страну издания книги, Author\_Biography – информация об авторе, Picture – путь к изображению автора.
* Genre – предназначен для создания жанра. Класс содержит Id – уникальный идентификатор жанра, Genre\_Name – название жанра.
* User – предназначен для создания пользователей. Данный класс содержит поля-свойства, хранимые в базе данных: Id – уникальный идентификатор пользователя, Name – имя пользователя, Surname – фамилия пользователя, Username – ник пользователя, User\_About – информация о пользователе, Picture – путь к изображению аватарки, Password – пароль пользователя, Block\_Id – число блокировки (1/0).
* UserBooks – предназначен для создания книги. Класс содержит Id – уникальный идентификатор, Book\_Id – идентификатор книги, User\_Id – идентификатор пользователя.

Data\_Context классы:

* Book\_Context – предназначен для установления связи с базой данных, доступ к таблице Books;
* Author\_Context – предназначен для установления связи с базой данных, доступ к таблице Authors;
* Genre\_Context – предназначен для установления связи с базой данных, доступ к таблице Genres;
* User\_Context – предназначен для установления связи с базой данных, доступ к таблице Users;
* UserBooks\_Context – предназначен для установления связи с базой данных, доступ к таблице UserBooks;

Views (Windows) классы:

* Admin – предоставляет окно администратора для добавления новой книги.
* Admin1– предоставляет окно администратора для добавления нового автора.
* Authorization – окно авторизации.
* BookElem – предоставляет одну книгу из тех, что располагаются в окнах MyBooks и SearchBooks.
* UserElem – предоставляет одного пользователя, что располагается в окне UsersPage.
* MyAccount – предоставляет информацию о пользователе с возможностью смены «аватарки» пользователя.
* MyBooks – окно, предоставляющее список книг пользователя.
* ReadyToRead – окно выбранной книги пользователя.
* Registration – окно регистрации.
* SearchBooks – окно, предоставляющее список всех книг, их поиск и сортировку.
* SelectedBook – окно выбранной книги из всех книг.
* SelectedUser – окно выбранного пользователя.
* SettingsWindow – окно настроек.
* AuthorElem – предоставляет одного автора из тех, что располагаются в окне SearchBooks после нажатия кнопки «Authors».
* PDFReader – предоставляет возможность чтения выбранной книги в pdf формате.
* MainWindow – данный класс отвечает за окно, которое запускается при запуске приложения.

Класс DBInitializar выполняет инициализацию данных в таблице Genres.

# **Тестирование программного средства**

При тестировании данного приложения были проведены следующие тесты на ситуации, которые могли бы привести к ошибке:

* тесты авторизации и регистрации;
* тест некорректно введённых данных при добавлении новой книги;
* тест некорректно введённых данных при добавлении нового автора;
* Попытка введения неверного пароля:

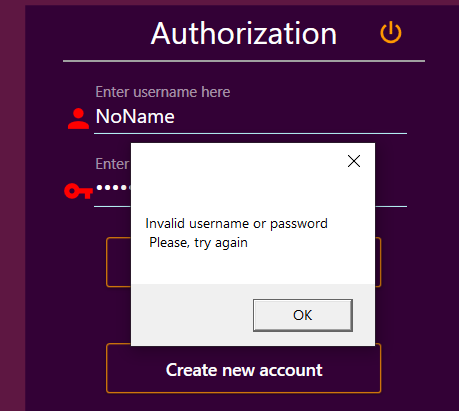
****

Рисунок 5.1 – Пользователь ввёл неверный пароль

– Попытка ввода некорректных данных при добавлении новой книги:

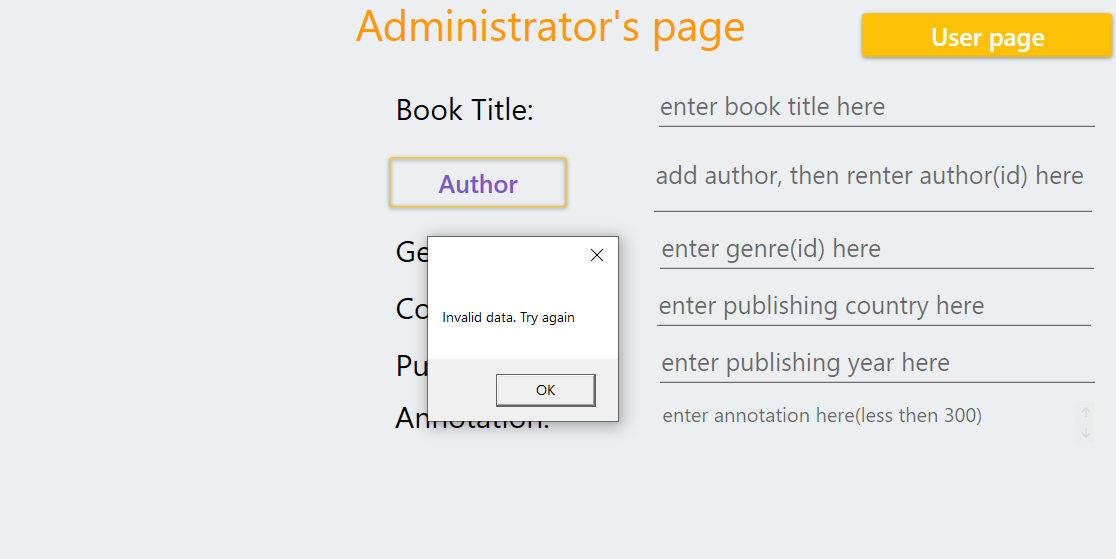
****

Рисунок 5.2 – Администратор ввёл некорректные данные

– Попытка ввода некорректных данных при добавлении нового автора:

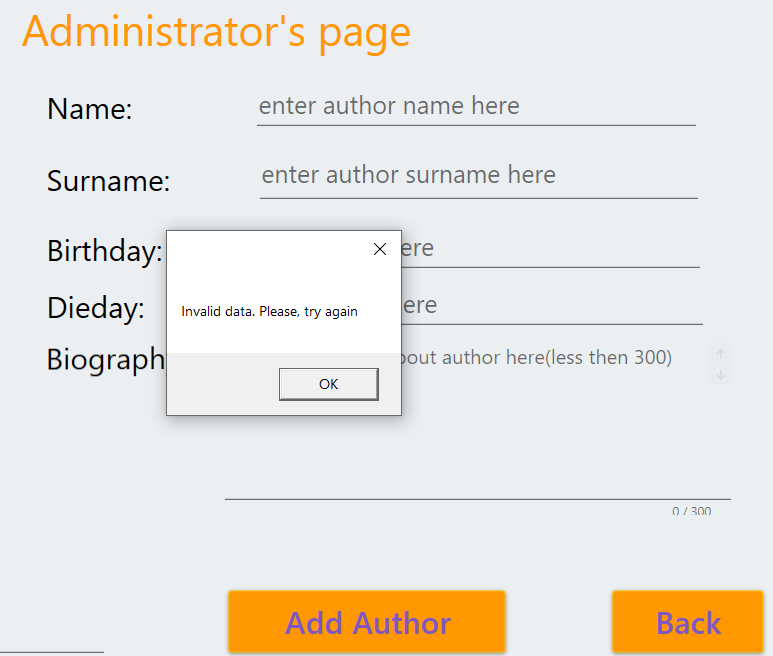


Рисунок 5.3 – Администратор ввёл некорректные данные

**Заключение**

В процессе решения поставленной задачи была достигнута поставленная цель по созданию программного средства «WorldOfBook». При разработке были выполнены все пункты из указанного списка предполагаемого основного функционала приложения.

В программном средстве были реализованы следующие функции: Администратор:

* + Добавить в систему новые книги;
  + Производить поиск книг;
  + Просматривать список всех пользователей;
  + Возможность блокировать/разблокировать пользователя

Пользователь:

* Добавлять книги в раздел «мои книги»;
* Читать книги из раздела «мои книги» в формате pdf;
* Удалять выбранные книги из раздела «мои книги»;
* Редактировать информацию о пользователе;
* Производить поиск книг;

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме.

**Список использованных источников**

1. MSDN сеть разработчиков в Microsoft [Электронный ресурс] / Режим доступа: <http://msdn.microsoft.com/library/rus/>. Дата доступа: 25.04.2021
2. METANIT.COM Сайт о программировании [Электронный ресурс] / Режим доступа: <https://metanit.com>. Дата доступа: 20.05.2021
3. ProfessorWeb .NET & Web Programming [Электронный ресурс] / Режим доступа: <https://professorweb.ru> Дата доступа: 13.04.2021
4. Youtube.com C# WPF Material Design UI Menu [Электронный ресурс]/ Режим доступа: <https://www.youtube.com/watch?v=YQ1EJJZBHyE&list=LLJYz_lF4d1mOUKPGhpU53AQ&index=2&t=584s>

**Приложение Б**