федеральное государственное бюджетное образовательное учреждение высшего образования

«**Вологодский государственный университет**»

университетский колледж

Информационные системы и программирование

(наименование специальности)

**КУРСОВОЙ ПРОЕКТ**

Дисциплина: «Проектно-исследовательская деятельность»

Наименование темы: «Разработка модуля чтения книг»

Код работы КП 09.02.07.07.00.005.2025

Код специальности, код УК, регистрационный номер по журналу, год

Руководитель Дегтярев Михаил Евгеньевич \_\_\_\_\_\_\_\_\_\_\_\_\_

(уч. степень, звание, должность. Ф.И.О)

Выполнил(а) студент(ка) Домаков Василий Николаевич \_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О)

Группа, курс ИСП 33-22, 3 курс \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата сдачи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка по защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись преподавателя)

Вологда

2025 г.

федеральное государственное бюджетное образовательное учреждение высшего образования

«**Вологодский государственный университет**»

университетский колледж

**Задание на курсовой проект**

По проектно-исследовательской деятельности

Обучающемуся Домаков Василий Николаевич

Специальности 09.02.07 Информационные системы и программирование

Тема: Разработка модуля чтения книг

Исходные данные к курсовому проекту:

1. .NET 7.0
2. ASP.NET
3. Microsoft SQL Server.

Содержание и объём курсового проекта:

1. Введение.
2. Аналитический обзор.
3. Разработка программного обеспечения.
4. Тестирование.
5. Руководство пользователя.
6. Заключение.
7. Список использованных источников.
8. Приложения.

Председатель П(Ц)К Дегтярёв Михаил Евгеньевич/ /

Руководитель курсового проекта Дегтярёв Михаил Евгеньевич/ /

Обучающийся Домаков Василий Николаевич/ /

Дата выдачи задания«\_\_»\_\_\_20\_\_г.

Срок выполнения«\_\_»\_\_\_20\_\_г.

Курсовой проект защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(Подпись руководителя)

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc197448573)

[1 АНАЛИТИЧЕСКИЙ ОБЗОР 5](#_Toc197448574)

[1.1 Описание требований к программному обеспечению 5](#_Toc197448575)

[1.2 Выбор средств разработки 6](#_Toc197448576)

[1.3 Описание базы данных 8](#_Toc197448577)

[2 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 14](#_Toc197448578)

[2.1 Разработка графического интерфейса 14](#_Toc197448579)

[2.2 Разработка программного функционала 92](#_Toc197448580)

[2.3 Разработка сервера 106](#_Toc197448581)

[3 ТЕСТИРОВАНИЕ 118](#_Toc197448582)

[3.1 Тестирование логики программы 118](#_Toc197448583)

[3.2 Тестирование интерфейса пользователя 118](#_Toc197448584)

[4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 119](#_Toc197448585)

[4.1 Введение 119](#_Toc197448586)

[4.2 Назначение и условия применения программного продукта 119](#_Toc197448587)

[4.3 Подготовка к работе 119](#_Toc197448588)

[4.4 Описание операций 120](#_Toc197448589)

[ЗАКЛЮЧЕНИЕ 128](#_Toc197448590)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 129](#_Toc197448591)

[ПРИЛОЖЕНИЕ 1 130](#_Toc197448592)

[ПРИЛОЖЕНИЕ 2 135](#_Toc197448593)

[ПРИЛОЖЕНИЕ 3 136](#_Toc197448594)

ВВЕДЕНИЕ

Автоматизация является важным аспектом в современном мире. Применение автоматизации позволяет сократить численность обслуживающего персонала, увеличить объем выпускаемой продукции, повысить эффективность производственного процесса, улучшить качество продукции, снизить расходы сырья, повысить безопасность, экологичность и экономичность производства.

В условиях роста популярности электронных книг и аудиокниг ключевыми задачами становятся обеспечение удобного доступа к контенту и повышение вовлеченности читателей. Современные пользователи ожидают от платформ не только широкого ассортимента, но и удобного интерфейса, возможности синхронизации прогресса чтения между устройствами и интеграции социальных функций. Разрабатываемый модуль призван решить эти задачи, сократив время на поиск и навигацию по материалам и упрощая управление персональной библиотекой.

Целью курсового проекта является разработка модуля информационной системы для сервиса чтения книг.

Задачи курсового проекта:

* разработка требований;
* выбор средств разработки;
* разработка графического интерфейса;
* разработка программного функционала;
* тестирование;
* разработка руководства пользователя.

1. АНАЛИТИЧЕСКИЙ ОБЗОР
2. Описание требований к программному обеспечению

Сервис чтения книг – это платформа, предоставляющая пользователям доступ к электронным книгам различных жанров и авторов. Пользователи могут читать книги онлайн или скачивать их для чтения в автономном режиме. Чем шире и разнообразнее библиотека, тем больше привлекается читателей.

Модуль информационной системы сервиса чтения книг позволит добавлять, редактировать и хранить информацию о книгах и пользователях, используя удобный интерфейс. Также хранить информацию о прочитанных книгах, отзывах и рейтингах. Система должна поддерживать управление правами доступа для различных ролей (администратор и пользователь).

С помощью модуля информационной системы пользователь сможет управлять библиотекой, управлять аккаунтов, контактировать с другими пользовталями, добавлять новые книги и читать их.

Use-case диаграмма представлена в соответствии с рисунком 1.

Изображение выглядит как текст, человек, Человеческое лицо, мультфильм

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок – Use-case диаграмма

Подробные требования к разрабатываемому программному обеспечению представлены в приложении 1.

1. Выбор средств разработки

Перед тем, как начать разработку информационной системы, необходимо произвести выбор средств, которые будут использоваться в ходе ее создания. В связи с этим было принято решение начать с определения оптимального языка программирования и фреймворка.

Существует несколько языков программирования, которые могут использоваться для разработки десктопных приложений. Один из них – C++. Он обладает высокой производительностью и позволяет использовать низкоуровневые возможности, что может быть полезно для решения задач, связанных с операционной системой [7]. Однако, C++ имеет сложный и запутанный синтаксис, и для создания десктопных приложений на C++ необходимо использование сторонних фреймворков, таких как Qt, MFC или wxWidgets.

Pascal – язык программирования, который имеет простой синтаксис и удобен для обучения программированию. Одним из фреймворков для создания десктопных приложений является Lazarus [8]. Он является кроссплатформенным средством разработки и использует графический интерфейс пользователя. Однако, разработка на Pascal может быть менее популярной и менее поддерживаемой, чем на других языках, и он может не иметь такого широкого сообщества разработчиков и библиотек, как C++ и C#.

В свою очередь, C# имеет более простой и удобный синтаксис, поддерживает объектно-ориентированное программирование и имеет широкие возможности для работы с графическим интерфейсом [9]. WPF предоставляет удобные инструменты для создания пользовательского интерфейса и может значительно ускорить разработку приложения.

Таким образом, было решено использовать C# и фреймворк WPF, потому что они являются современными и удобными в использовании, позволяют быстро разрабатывать десктопные приложения с хорошей производительностью, а также имеют широкий набор инструментов.

Далее необходимо определиться с интеграционной средой разработки, в которой будет разрабатываться будущее приложение. Были рассмотрены такие среды, как Visual Studio, JetBrains Rider и Eclipse

Visual Studio – это интегрированная среда разработки (IDE), разработанная Microsoft. Она поддерживает большой набор языков программирования, таких как C++, C#, F#, Visual Basic и другие [1]. Visual Studio имеет множество инструментов для упрощения процесса разработки, таких как отладчик, автодополнение, интеграцию с Git и другими системами контроля версий и многие другие. Одним из основных преимуществ Visual Studio является его удобный и интуитивно понятный интерфейс, что делает его очень простым в использовании.

JetBrains Rider – это кроссплатформенная IDE, которая поддерживает различные языки программирования, включая C#, F#, VB.NET, ASP.NET, JavaScript, TypeScript, HTML, CSS и другие [3]. Rider предлагает широкий набор инструментов для отладки, профилирования, тестирования и анализа кода. Он также интегрируется со многими системами контроля версий, такими как Git, Subversion и Mercurial.

Eclipse – это кроссплатформенная IDE, которая в основном используется для разработки на языках Java и PHP [4]. Он также поддерживает другие языки, такие как C++, C#, Python и другие. Eclipse имеет множество инструментов для упрощения процесса разработки, таких как отладчик, автодополнение и интеграцию с Git и другими системами контроля версий.

В итоге, было решено использовать Visual Studio в качестве интегрированной среды разработки, поскольку она предоставляет множество преимуществ по сравнению с другими рассмотренными средами. Она имеет широкий выбор инструментов и функций для разработки различных приложений, включая встроенный отладчик, удобный интерфейс и интеграцию с многими платформами и технологиями.

Далее необходимо определиться с системой управления баз данных, в которой будут храниться данные приложения. Наиболее популярными СУБД являются MySQL, Microsoft SQL Server и Oracle Database.

MySQL – это СУБД с открытым исходным кодом, созданная на базе языка программирования Си [5]. Она часто используется веб-разработчиками для создания сайтов и приложений. MySQL может работать на разных операционных системах и поддерживает большинство языков программирования. Однако, у нее могут быть проблемы с производительностью при большом объеме данных.

Microsoft SQL Server – это СУБД, созданная компанией Microsoft, предназначенная для работы в Windows-среде [2]. Она может работать с различными языками программирования и может использоваться для создания веб-приложений и десктопных приложений. Microsoft SQL Server имеет большой функционал и может обеспечить быстрое выполнение запросов и масштабирование приложения.

Oracle Database – это СУБД, созданная компанией Oracle. Она используется для создания больших корпоративных приложений и обладает высокой производительностью и масштабируемостью [6]. Oracle Database имеет множество функций и возможностей для управления данными, но она также имеет высокую стоимость и может быть сложной в использовании для малых и средних проектов.

Исходя из требований к проекту, было решено использовать MS SQL в качестве СУБД, так как она является надежной и масштабируемой СУБД, имеет высокую производительность и хорошо интегрируется с Visual Studio. Кроме того, MS SQL поддерживает большое количество языков программирования, что позволяет разработчикам использовать их наиболее удобный язык.

1. Описание базы данных
2. Таблица Users:

* Id – ключевое поле, тип данных int, уникальный идентификатор пользователя (автоинкремент);
* Nickname – тип данных nvarchar(50), содержит псевдоним пользователя;
* Description – тип данных nvarchar(150), содержит описание профиля пользователя (может быть NULL);
* Avatar\_image\_path – тип данных nvarchar(255), содержит путь к изображению профиля пользователя (по умолчанию «default-avatar.png»);
* Name – тип данных nvarchar(100), содержит имя пользователя;
* Email – тип данных nvarchar(150), содержит адрес электронной почты (уникальное значение);
* Password\_hash – тип данных nvarchar(255), содержит хэш пароля;
* Id\_role – тип данных int, ссылается на таблицу Roles (идентификатор роли пользователя по умолчанию 0);
* Created\_at – тип данных datetime, заполняется датой и временем создания записи.

1. Таблица Roles:

* Id – ключевое поле, тип данных int, уникальный идентификатор роли (автоинкремент);
* Name – тип данных nvarchar(20), содержит название роли («admin» или «user»).

1. Таблица Books:

* Id – ключевое поле, тип данных int, уникальный идентификатор книги (автоинкремент);
* Name – тип данных nvarchar(200), содержит название книги;
* Description – тип данных nvarchar(250), содержит краткое описание книги (может быть NULL);
* Id\_author – тип данных int, ссылается на таблицу Users (идентификатор автора книги);
* Cover\_image\_path – тип данных nvarchar(250), содержит путь к обложке книги;
* File\_book\_path – тип данных nvarchar(200), содержит путь к файлу книги;
* Id\_genre – тип данных int, ссылается на таблицу Genres (идентификатор категории книги);
* Is\_private – тип данных bit, содержит флаг приватности книги (0 — публичная, 1 — приватная);
* Created\_at – тип данных datetime, заполняется датой и временем создания записи (может быть NULL).

1. Таблица Genres:

* Id – ключевое поле, тип данных int, уникальный идентификатор категории (автоинкремент);
* Name – тип данных nvarchar(100), содержит название жанра.

1. Таблица Libraries:

* Id – ключевое поле, тип данных int, уникальный идентификатор записи (автоинкремент);
* Id\_user – тип данных int, ссылается на таблицу Users (идентификатор пользователя, владеющего библиотекой);
* Id\_book – тип данных int, ссылается на таблицу Books (идентификатор книги, добавленной в библиотеку);
* Created\_at – тип данных datetime, заполняется датой и временем добавления книги в библиотеку;
* Progress\_page – тип данных int, содержит номер текущей страницы, на которой находится пользователь (по умолчанию 0).

1. Таблица BookReviews:

* Id – ключевое поле, тип данных int, уникальный идентификатор отзыва (автоинкремент);
* Id\_author – тип данных int, ссылается на таблицу Users (идентификатор автора отзыва);
* Id\_book – тип данных int, ссылается на таблицу Books (идентификатор книги, к которой относится отзыв);
* Comment – тип данных nvarchar(500), содержит текст отзыва;
* Rating – тип данных tinyint, содержит рейтинг книги (число от 0 до 10);
* Created\_at – тип данных datetime, заполняется датой и временем создания отзыва.

1. Таблица Logs:

* Id – ключевое поле, тип данных int, уникальный идентификатор записи (автоинкремент);
* Id\_user – тип данных int, ссылается на таблицу Users (идентификатор пользователя, совершившего действие);
* Action – тип данных nvarchar(200), содержит описание действия;
* Created\_at – тип данных datetime, заполняется датой и временем совершения действия.

1. Таблица UserSubscribers:

* Id – ключевое поле, тип данных int, уникальный идентификатор записи (автоинкремент);
* Id\_author – тип данных int, ссылается на таблицу Users (идентификатор пользователя, на которого подписан другой пользователь);
* Id\_subscriber – тип данных int, ссылается на таблицу Users (идентификатор пользователя, который подписан на другого пользователя);
* Created\_at – тип данных datetime, заполняется датой и временем создания подписки.

1. Таблица UserSessions:

* Id – ключевое поле, тип данных int, уникальный идентификатор записи (автоинкремент);
* Id\_user – тип данных int, ссылается на таблицу Users (идентификатор пользователя, которому принадлежит сессия);
* Refresh\_token\_hash – тип данных nvarchar(255), содержит refresh токен;
* Device\_type – тип данных nvarchar(50), содержит тип устройства.
* Expires\_in – тип данных datetime, содержит дату и время истечения refresh токена.
* Created\_at – тип данныз datetime, содержит дату создания refresh токена.

1. Таблица ConfirmationCodes:

* Id – тип данных int (идентификатор кода подтверждения);
* Code – тип данных nvarchar(6), содержит код подтверждения;
* Email – тип данных nvarchar(150), содержит почту, на которую пришел код подтверждения;
* Expires\_id – тип данных datetime, содержит время, когда код подтверждения истекает.

ER-диаграмма базы данных представлена в соответствии с рисунком 2.

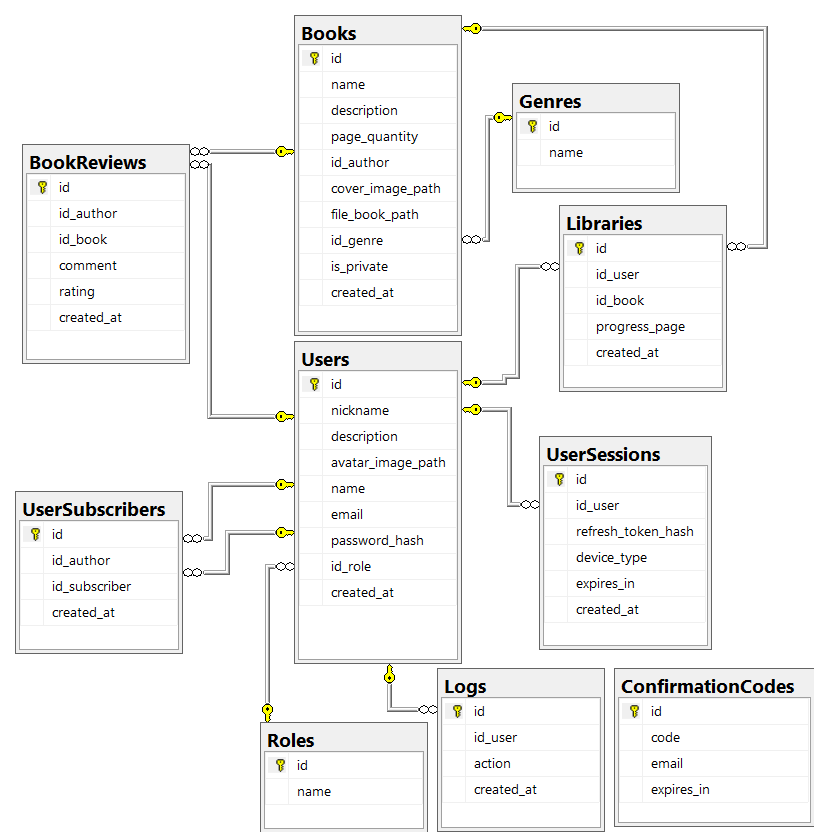


Рисунок 2 – ER-диаграмма базы данных.

1. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
2. Разработка графического интерфейса

Для начала был разработан набор индивидуальных стилей и шаблонов для элементов пользовательского интерфейса. Это было сделано с целью создания более привлекательного и современного дизайна, а также для облегчения использования приложения для конечных пользователей. С помощью этих стилей и шаблонов удалось создать единый и последовательный интерфейс на всех формах и элементах приложения. Пример одного из стилей представлен ниже:

XAML-код файла «Borders.xaml» представлен ниже:

*<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"*

*xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">*

*<Style x:Key="GreenLoginButton" TargetType="{x:Type Button}">*

*<Setter Property="Height" Value="39"/>*

*<Setter Property="Background" Value="#30C352"/>*

*<Setter Property="BorderThickness" Value="0"/>*

*<Setter Property="Foreground" Value="#FFFFFF"/>*

*<Setter Property="FontFamily" Value="{StaticResource Inter}"/>*

*<Setter Property="FontWeight" Value="SemiBold"/>*

*<Setter Property="FontSize" Value="17"/>*

*<Setter Property="Cursor" Value="Hand"/>*

*<Setter Property="Template">*

*<Setter.Value>*

*<ControlTemplate TargetType="{x:Type Button}">*

*<Border x:Name="border" CornerRadius="4" Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}">*

*<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/>*

*</Border>*

*<ControlTemplate.Triggers>*

*<Trigger Property="IsMouseOver" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#28A845"/>*

*</Trigger>*

*<Trigger Property="IsPressed" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#1F8C3A"/>*

*</Trigger>*

*</ControlTemplate.Triggers>*

*</ControlTemplate>*

*</Setter.Value>*

*</Setter>*

*</Style>*

*<Style x:Key="RegistrationButton" TargetType="Button">*

*<Setter Property="FontSize" Value="18"/>*

*<Setter Property="FontFamily" Value="{StaticResource Inter}"/>*

*<Setter Property="FontWeight" Value="Bold"/>*

*<Setter Property="Background" Value="#25292E"/>*

*<Setter Property="Foreground" Value="White"/>*

*<Setter Property="Height" Value="50"/>*

*<Setter Property="Margin" Value="0 10 0 0"/>*

*<Setter Property="Cursor" Value="Hand"/>*

*<Setter Property="Template">*

*<Setter.Value>*

*<ControlTemplate TargetType="Button">*

*<Border x:Name="border" CornerRadius="4" Background="{TemplateBinding Background}" BorderThickness="0">*

*<ContentPresenter x:Name="content" Content="{TemplateBinding Content}" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="10,0"/>*

*</Border>*

*<ControlTemplate.Triggers>*

*<Trigger Property="IsMouseOver" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#1A1D21"/>*

*</Trigger>*

*<Trigger Property="IsPressed" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#0E1012"/>*

*</Trigger>*

*</ControlTemplate.Triggers>*

*</ControlTemplate>*

*</Setter.Value>*

*</Setter>*

*</Style>*

*<Style x:Key="ProfileUserButton" TargetType="Button">*

*<Setter Property="FontSize" Value="15"/>*

*<Setter Property="FontFamily" Value="{StaticResource Inter}"/>*

*<Setter Property="FontWeight" Value="Bold"/>*

*<Setter Property="Background" Value="#25292E"/>*

*<Setter Property="Foreground" Value="White"/>*

*<Setter Property="Height" Value="45"/>*

*<Setter Property="Margin" Value="0 0 0 0"/>*

*<Setter Property="Cursor" Value="Hand"/>*

*<Setter Property="Template">*

*<Setter.Value>*

*<ControlTemplate TargetType="Button">*

*<Border x:Name="border" Height="40" CornerRadius="8" Background="{TemplateBinding Background}" BorderThickness="0">*

*<ContentPresenter x:Name="content" Content="{TemplateBinding Content}" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="10,0"/>*

*</Border>*

*<ControlTemplate.Triggers>*

*<Trigger Property="IsMouseOver" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#1A1D21"/>*

*</Trigger>*

*<Trigger Property="IsPressed" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#0E1012"/>*

*</Trigger>*

*</ControlTemplate.Triggers>*

*</ControlTemplate>*

*</Setter.Value>*

*</Setter>*

*</Style>*

*<Style x:Key="UploadBookButton" TargetType="Button">*

*<Setter Property="FontSize" Value="15"/>*

*<Setter Property="FontFamily" Value="{StaticResource Inter}"/>*

*<Setter Property="FontWeight" Value="Bold"/>*

*<Setter Property="Background" Value="#EFEFEF"/>*

*<Setter Property="Foreground" Value="Black"/>*

*<Setter Property="Height" Value="45"/>*

*<Setter Property="Margin" Value="0 0 0 0"/>*

*<Setter Property="Cursor" Value="Hand"/>*

*<Setter Property="Template">*

*<Setter.Value>*

*<ControlTemplate TargetType="Button">*

*<Border x:Name="border" Height="40" CornerRadius="8" Background="{TemplateBinding Background}" BorderThickness="0">*

*<ContentPresenter x:Name="content" Content="{TemplateBinding Content}" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="10,0"/>*

*</Border>*

*<ControlTemplate.Triggers>*

*<Trigger Property="IsMouseOver" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#E4E4E4"/>*

*</Trigger>*

*<Trigger Property="IsPressed" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#BABABA"/>*

*</Trigger>*

*</ControlTemplate.Triggers>*

*</ControlTemplate>*

*</Setter.Value>*

*</Setter>*

*</Style>*

*<Style x:Key="SubscriptionButtonStyle" TargetType="Button">*

*<Setter Property="Width" Value="120"/>*

*<Setter Property="Height" Value="35"/>*

*<Setter Property="FontSize" Value="16"/>*

*<Setter Property="HorizontalAlignment" Value="Right"/>*

*<Setter Property="VerticalAlignment" Value="Top"/>*

*<Setter Property="Margin" Value="0 12 0 0"/>*

*<Setter Property="Background" Value="#25292E"/>*

*<Setter Property="Foreground" Value="#FFFFFF"/>*

*<Setter Property="FontFamily" Value="{StaticResource Inter}"/>*

*<Setter Property="FontWeight" Value="SemiBold"/>*

*<Setter Property="BorderThickness" Value="0"/>*

*<Setter Property="Cursor" Value="Hand"/>*

*<Setter Property="Template">*

*<Setter.Value>*

*<ControlTemplate TargetType="Button">*

*<Border x:Name="border" CornerRadius="5" Background="{TemplateBinding Background}" BorderThickness="{TemplateBinding BorderThickness}">*

*<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/>*

*</Border>*

*<ControlTemplate.Triggers>*

*<Trigger Property="IsMouseOver" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#3A3F45"/>*

*</Trigger>*

*<Trigger Property="IsPressed" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#1D1F22"/>*

*</Trigger>*

*</ControlTemplate.Triggers>*

*</ControlTemplate>*

*</Setter.Value>*

*</Setter>*

*</Style>*

*<Style x:Key="BookReadButton" TargetType="Button">*

*<Setter Property="FontSize" Value="15"/>*

*<Setter Property="FontFamily" Value="{StaticResource Inter}"/>*

*<Setter Property="FontWeight" Value="Bold"/>*

*<Setter Property="Background" Value="#25292E"/>*

*<Setter Property="Foreground" Value="White"/>*

*<Setter Property="Height" Value="40"/>*

*<Setter Property="Margin" Value="0 5"/>*

*<Setter Property="Cursor" Value="Hand"/>*

*<Setter Property="Template">*

*<Setter.Value>*

*<ControlTemplate TargetType="Button">*

*<Border x:Name="border" Height="40" CornerRadius="8" Background="{TemplateBinding Background}" BorderThickness="0">*

*<ContentPresenter x:Name="content" Content="{TemplateBinding Content}" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="10,0"/>*

*</Border>*

*<ControlTemplate.Triggers>*

*<Trigger Property="IsMouseOver" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#1A1D21"/>*

*</Trigger>*

*<Trigger Property="IsPressed" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#0E1012"/>*

*</Trigger>*

*</ControlTemplate.Triggers>*

*</ControlTemplate>*

*</Setter.Value>*

*</Setter>*

*</Style>*

*<Style x:Key="BookButton" TargetType="Button">*

*<Setter Property="FontSize" Value="15"/>*

*<Setter Property="FontFamily" Value="{StaticResource Inter}"/>*

*<Setter Property="FontWeight" Value="Bold"/>*

*<Setter Property="Background" Value="#EFEFEF"/>*

*<Setter Property="Foreground" Value="Black"/>*

*<Setter Property="Height" Value="40"/>*

*<Setter Property="Margin" Value="0 5"/>*

*<Setter Property="Cursor" Value="Hand"/>*

*<Setter Property="Template">*

*<Setter.Value>*

*<ControlTemplate TargetType="Button">*

*<Border x:Name="border" Height="40" CornerRadius="8" Background="{TemplateBinding Background}" BorderThickness="0">*

*<ContentPresenter x:Name="content" Content="{TemplateBinding Content}" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="10,0"/>*

*</Border>*

*<ControlTemplate.Triggers>*

*<Trigger Property="IsMouseOver" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#E4E4E4"/>*

*</Trigger>*

*<Trigger Property="IsPressed" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#BABABA"/>*

*</Trigger>*

*</ControlTemplate.Triggers>*

*</ControlTemplate>*

*</Setter.Value>*

*</Setter>*

*</Style>*

*<Style x:Key="DeleteBookButton" TargetType="Button">*

*<Setter Property="FontSize" Value="15"/>*

*<Setter Property="FontFamily" Value="{StaticResource Inter}"/>*

*<Setter Property="FontWeight" Value="Bold"/>*

*<Setter Property="Background" Value="#D1242F"/>*

*<Setter Property="Foreground" Value="White"/>*

*<Setter Property="Height" Value="40"/>*

*<Setter Property="Margin" Value="0 5"/>*

*<Setter Property="Cursor" Value="Hand"/>*

*<Setter Property="Template">*

*<Setter.Value>*

*<ControlTemplate TargetType="Button">*

*<Border x:Name="border" Height="40" CornerRadius="8" Background="{TemplateBinding Background}" BorderThickness="0">*

*<ContentPresenter x:Name="content" Content="{TemplateBinding Content}" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="10,0"/>*

*</Border>*

*<ControlTemplate.Triggers>*

*<Trigger Property="IsMouseOver" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#CD1722"/>*

*</Trigger>*

*<Trigger Property="IsPressed" Value="True">*

*<Setter TargetName="border" Property="Background" Value="#C7000D"/>*

*</Trigger>*

*</ControlTemplate.Triggers>*

*</ControlTemplate>*

*</Setter.Value>*

*</Setter>*

*</Style>*

*</ResourceDictionary>*

Все стили подключаются в файле «App.xaml», потому что используются в большинстве окон приложения.

XAML-код файла «App.xaml» представлен ниже:

*<Application x:Class="Readify.App"*

*xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"*

*xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"*

*xmlns:local="clr-namespace:Readify">*

*<Application.Resources>*

*<ResourceDictionary>*

*<ResourceDictionary.MergedDictionaries>*

*<ResourceDictionary Source="Styles/Fonts.xaml" />*

*<ResourceDictionary Source="Styles/TextBoxes.xaml" />*

*<ResourceDictionary Source="Styles/Texts.xaml" />*

*<ResourceDictionary Source="Styles/Buttons.xaml" />*

*<ResourceDictionary Source="Styles/Windows.xaml" />*

*<ResourceDictionary Source="Styles/Borders.xaml" />*

*<ResourceDictionary Source="Styles/ScrollViewers.xaml" />*

*</ResourceDictionary.MergedDictionaries>*

*</ResourceDictionary>*

*</Application.Resources>*

*</Application>*

Окно входа в систему содержит поля ввода логина и пароля, а также кнопки «Войти» и «Загрегистрироваться». Окно входа в систему представлено в соответствии с рисунком 3.

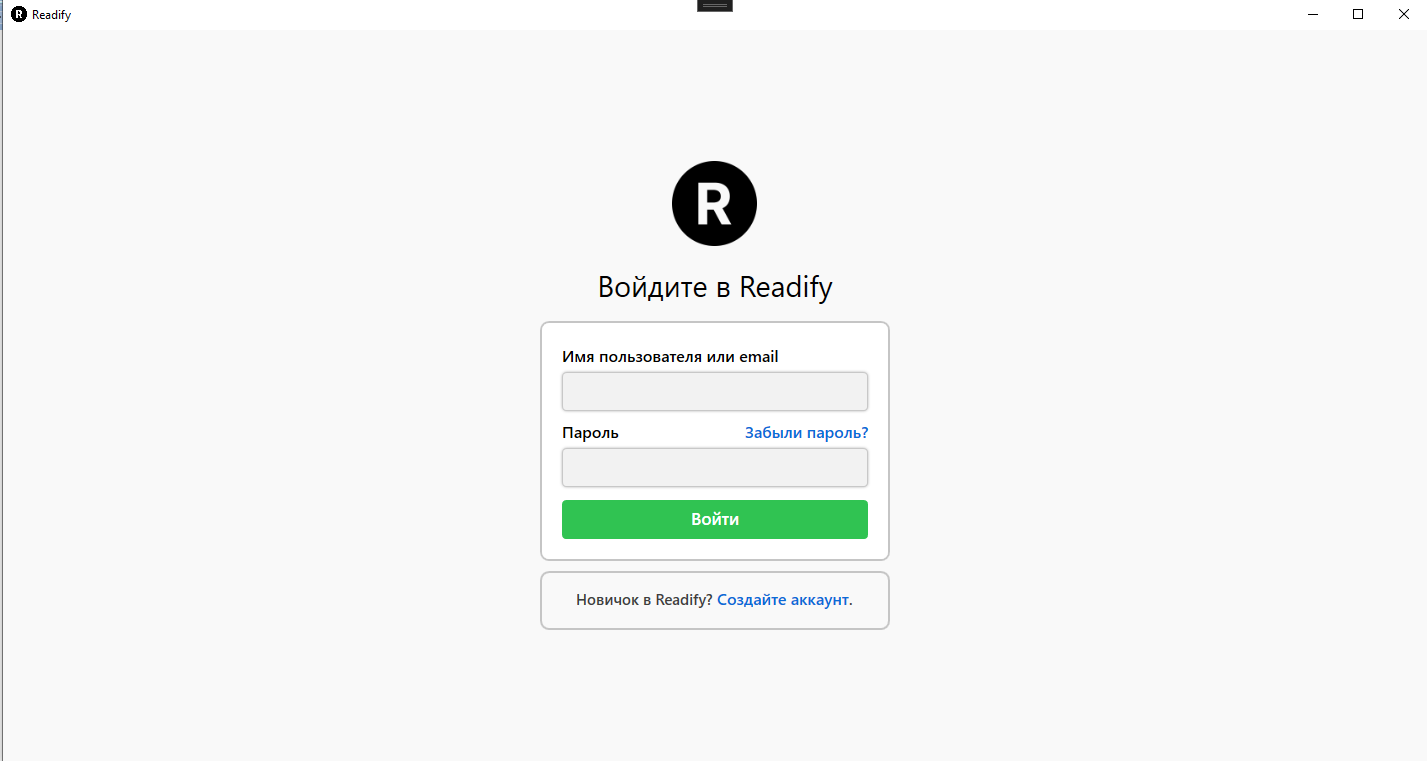


Рисунок – Окно входа

XAML-код файла «LoginPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.LoginPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Readify.Pages"

xmlns:registration="clr-namespace:Readify.ViewModels"

d:DataContext="{d:DesignInstance Type=registration:LoginViewModel}"

mc:Ignorable="d"

d:DesignHeight="700" d:DesignWidth="1300">

<Grid Background="#F9F9F9">

<ScrollViewer Style="{StaticResource MainScrollViewerStyle}">

<Border Padding="50 0" Width="450" VerticalAlignment="Center">

<StackPanel Orientation="Vertical">

<Image Source="pack://application:,,,/Images/Logo/logo.png" Width="85" RenderOptions.BitmapScalingMode="HighQuality" Margin="0 0 0 10"/>

<TextBlock Style="{StaticResource CenteredHeaderTextStyle}" Text="Войдите в Readify"/>

<Border CornerRadius="8" Background="#FFEBE9" BorderBrush="#FFC1C0" BorderThickness="2" Margin="0 0 0 10" Padding="20 20 20 20" Visibility="{Binding IsErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}">

<DockPanel LastChildFill="True">

<TextBlock DockPanel.Dock="Left" Style="{StaticResource ErrorTextBlockStyle}" Text="{Binding ErrorMessage}" MaxWidth="290" x:Name="ErrorText"/>

<Image DockPanel.Dock="Right" Source="pack://application:,,,/Images/close.png" Width="20" HorizontalAlignment="Right" VerticalAlignment="Center" Cursor="Hand">

<Image.InputBindings>

<MouseBinding Command="{Binding HiddenErrorCommand}" MouseAction="LeftClick"/>

</Image.InputBindings>

</Image>

</DockPanel>

</Border>

<Border CornerRadius="8" Background="#FFFFFF" BorderBrush="#C5C5C5" BorderThickness="2" Padding="20 25 20 20" Margin="0 0 0 10">

<StackPanel>

<TextBlock Text="Имя пользователя или email" Style="{StaticResource FormLabelLoginTextBlockStyle}"/>

<TextBox x:Name="Login" Text="{Binding Username}" Style="{StaticResource RoundedShadowTextBox}" TabIndex="0" PreviewKeyDown="Enter\_PreviewKeyDown"/>

<DockPanel LastChildFill="False">

<TextBlock DockPanel.Dock="Left" Text="Пароль" Style="{StaticResource FormLabelLoginTextBlockStyle}"/>

</DockPanel>

<PasswordBox x:Name="Password" PasswordChanged="PasswordBox\_PasswordChanged" Style="{StaticResource RoundedShadowPasswordBox}" TabIndex="1" PreviewKeyDown="Enter\_PreviewKeyDown"/>

<Button x:Name="LoginButton" Command="{Binding LoginCommand}" Style="{StaticResource GreenLoginButton}" Content="Войти" Focusable="True" TabIndex="2"/>

</StackPanel>

</Border>

<Border Padding="20 18 20 18" CornerRadius="8" BorderBrush="#C5C5C5" BorderThickness="2">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Новичок в Readify? " Style="{StaticResource RegularLoginTextStyle}"/>

<TextBlock Text="Создайте аккаунт" Style="{StaticResource HyperlinkLoginTextStyle}">

<TextBlock.InputBindings>

<MouseBinding Command="{Binding GoToRegisterCommand}" MouseAction="LeftClick"/>

</TextBlock.InputBindings>

</TextBlock>

<TextBlock Text="." Style="{StaticResource RegularLoginTextStyle}"/>

</StackPanel>

</Border>

</StackPanel>

</Border>

</ScrollViewer>

</Grid>

</UserControl>

В моем приложении регистрация состоит из 4 этапов:

1. Ввод почты, пароля и логина.
2. Ввод капчи.
3. Подтверждение почты.
4. Заполнение профиля.

Страница первого этапа регистрации содержит такие поля, как почта, пароль, повторения пароля, логин, а также кнопку «Продолжить» и кнопку «Уже есть аккаунт? Войти».

Страница первого этапа регистрации представлена в соответствии с рисунком 4.

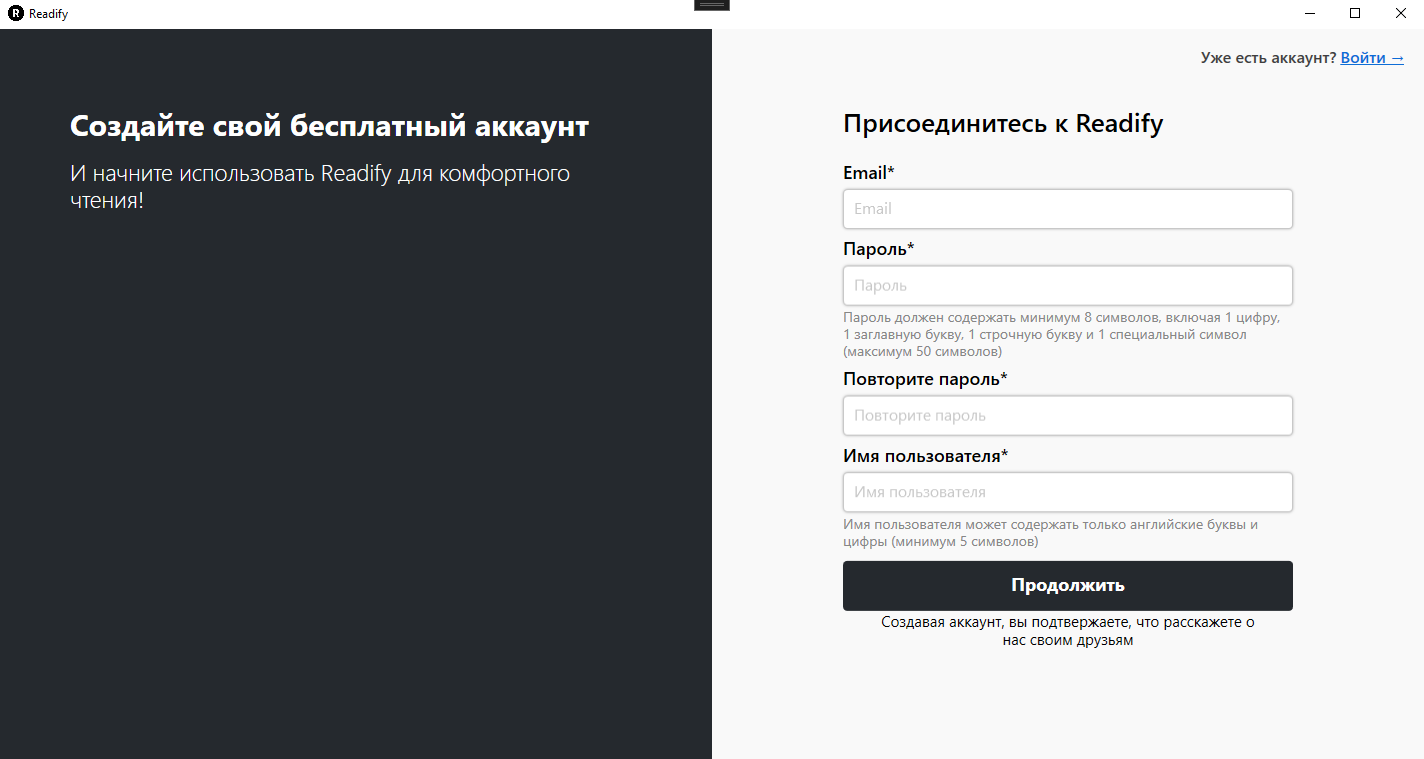


Рисунок – Первый этап регистрации

XAML-код файла «RegistrationMainPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.Registartion.RegistrationMainPage" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:local="clr-namespace:Readify.Pages.Registartion" xmlns:i="http://schemas.microsoft.com/xaml/behaviors" xmlns:registration="clr-namespace:Readify.ViewModels.Registration" d:DataContext="{d:DesignInstance Type=registration:RegistrationMainViewModel}" mc:Ignorable="d" d:DesignHeight="800" d:DesignWidth="1300">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Border Grid.Column="0" Style="{StaticResource LeftBorderRegistartion}">

<StackPanel>

<TextBlock Style="{StaticResource RegistartionLeftHeaderText}" Text="Создайте свой бесплатный аккаунт" />

<TextBlock Style="{StaticResource RegistartionLeftText}" Text="И начните использовать Readify для комфортного чтения!" />

</StackPanel>

</Border>

<Border Grid.Column="1" Style="{StaticResource RightBorderRegistartion}">

<StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right">

<TextBlock Style="{StaticResource RegistrationHelpLinkText}" Text="Уже есть аккаунт?" />

<TextBlock Style="{StaticResource RegistrationLinkText}" Text="Войти →" >

<TextBlock.InputBindings>

<MouseBinding Command="{Binding GoToLoginCommand}" MouseAction="LeftClick" />

</TextBlock.InputBindings>

</TextBlock>

</StackPanel>

<Border VerticalAlignment="Center" Width="450" Margin="0 40 0 0">

<StackPanel>

<TextBlock Style="{StaticResource RegistrationHeaderText}" Text="Присоединитесь к Readify" />

<TextBlock Style="{StaticResource RegistrationInputLabel}" Text="Email\*" />

<TextBox Text="{Binding Email}" Name="Email" Tag="Email" TabIndex="0" Style="{StaticResource RoundedShadowLoginTextBox}" PreviewKeyDown="Enter\_PreviewKeyDown">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding CloseEmailErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsEmailErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding EmailErrorMessage}" />

<TextBlock Style="{StaticResource RegistrationInputLabel}" Text="Пароль\*" />

<TextBox Style="{StaticResource RoundedShadowLoginTextBox}" Text="{Binding Password}" Name="Password" Tag="Пароль" TabIndex="1" PreviewKeyDown="Enter\_PreviewKeyDown">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding ClosePasswordErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsPasswordErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding PasswordErrorMessage}" />

<TextBlock Style="{StaticResource RegistrationInstructionText}" Text="Пароль должен содержать минимум 8 символов, включая 1 цифру, 1 заглавную букву, 1 строчную букву и 1 специальный символ (максимум 50 символов)"/>

<TextBlock Style="{StaticResource RegistrationInputLabel}" Text="Повторите пароль\*" />

<TextBox Style="{StaticResource RoundedShadowLoginTextBox}" Text="{Binding RetryPassword}" Tag="Повторите пароль" TabIndex="2" Name="RetryPassword" PreviewKeyDown="Enter\_PreviewKeyDown">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding CloseRetryPasswordErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsRetryPasswordErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding RetryPasswordErrorMessage}" />

<TextBlock Style="{StaticResource RegistrationInputLabel}" Text="Имя пользователя\*" />

<TextBox Style="{StaticResource RoundedShadowLoginTextBox}" Text="{Binding Username}" Name="Username" Tag="Имя пользователя" TabIndex="3" PreviewKeyDown="Enter\_PreviewKeyDown">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding CloseUsernameErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsUsernameErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding UsernameErrorMessage}" />

<TextBlock Style="{StaticResource RegistrationInstructionText}" Text="Имя пользователя может содержать только английские буквы и цифры (минимум 5 символов)"/>

<Button Style="{StaticResource RegistrationButton}" Command="{Binding ContinueCommand}" Content="Продолжить" TabIndex="4"/>

<TextBlock Style="{StaticResource RegistrationUnderButtonText}" Text="Создавая аккаунт, вы подтвержаете, что расскажете о нас своим друзьям" />

</StackPanel>

</Border>

</StackPanel>

</Border>

</Grid>

</UserControl>

Страница второго этапа регистрации содержит такое поле как ввод капчи, а также кнопку «Продолжить» и кнопку «Уже есть аккаунт? Войти».

Страница второго этапа регистрации представлена в соответствии с рисунком 5.

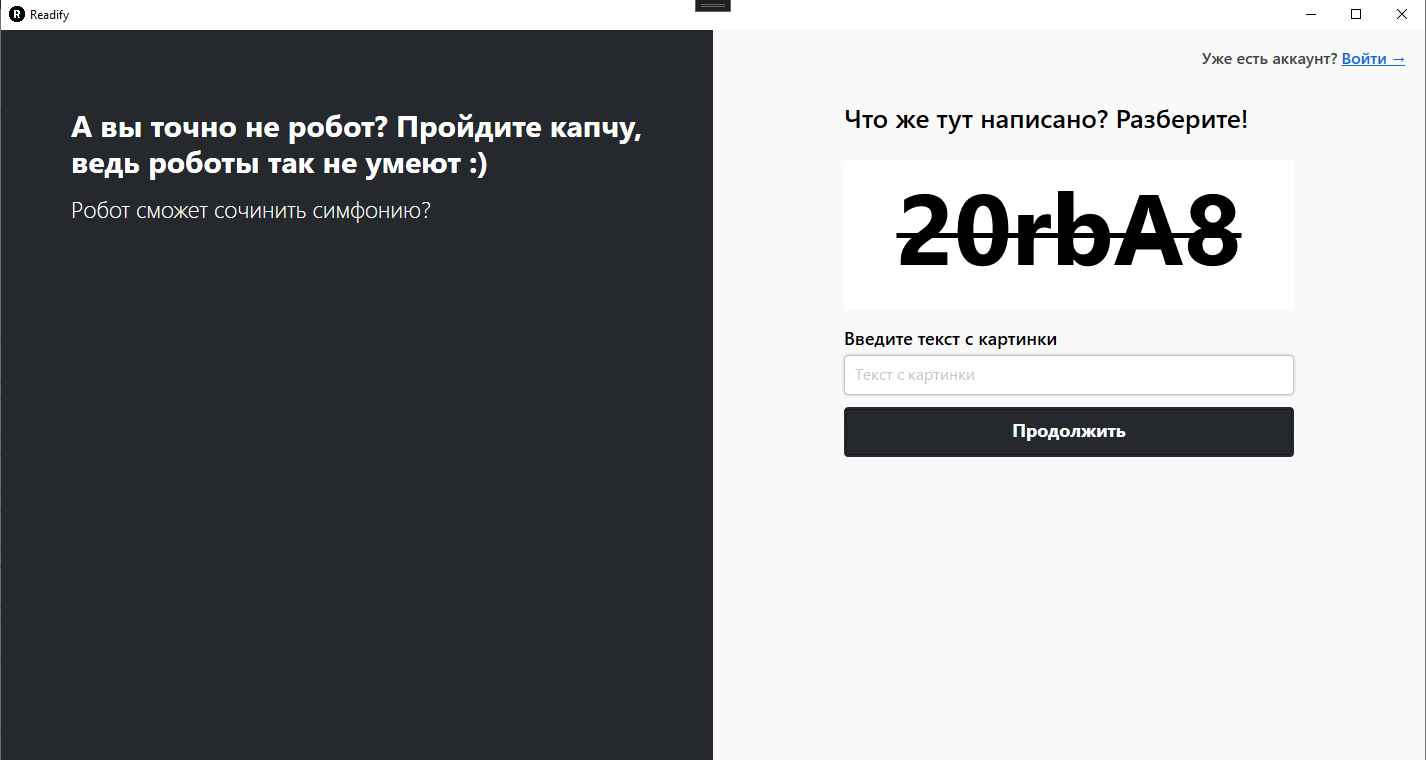


Рисунок – Второй этап регистрации

XAML-код файла «RegistrationCapthaPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.Registartion.RegistrationCapthaPage" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:local="clr-namespace:Readify.Pages.Registartion" xmlns:i="http://schemas.microsoft.com/xaml/behaviors" xmlns:registration="clr-namespace:Readify.ViewModels.Registration" d:DataContext="{d:DesignInstance Type=registration:RegistrationCaptchaViewModel}" mc:Ignorable="d" d:DesignHeight="700" d:DesignWidth="1300">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Border Grid.Column="0" Style="{StaticResource LeftBorderRegistartion}">

<StackPanel>

<TextBlock Style="{StaticResource RegistartionLeftHeaderText}" Text="А вы точно не робот? Пройдите капчу, ведь роботы так не умеют :)"/>

<TextBlock Style="{StaticResource RegistartionLeftText}" Text="Робот сможет сочинить симфонию?"/>

</StackPanel>

</Border>

<Border Grid.Column="1" Style="{StaticResource RightBorderRegistartion}">

<StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right">

<TextBlock Style="{StaticResource RegistrationHelpLinkText}" Text="Уже есть аккаунт?"/>

<TextBlock Style="{StaticResource RegistrationLinkText}" Text="Войти →">

<TextBlock.InputBindings>

<MouseBinding Command="{Binding GoToLoginCommand}" MouseAction="LeftClick"/>

</TextBlock.InputBindings>

</TextBlock>

</StackPanel>

<Border VerticalAlignment="Center" Width="450" Margin="0 35 0 0">

<StackPanel>

<TextBlock Style="{StaticResource RegistrationHeaderText}" Text="Что же тут написано? Разберите!"/>

<Border Height="150" Margin="0 10 0 10" Background="#FFFFFF">

<TextBlock FontFamily="{StaticResource Inter}" FontWeight="Bold" FontSize="100" Text="{Binding CaptchaText}" TextDecorations="Strikethrough" VerticalAlignment="Center" HorizontalAlignment="Center"/>

</Border>

<TextBlock Style="{StaticResource RegistrationInputLabel}" Text="Введите текст с картинки"/>

<TextBox Text="{Binding Captcha}" Tag="Текст с картинки" TabIndex="0" Style="{StaticResource RoundedShadowLoginTextBox}">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding CloseCaptchaErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsCaptchaErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding CaptchaErrorMessage}"/>

<Button Style="{StaticResource RegistrationButton}" Command="{Binding ContinueCommand}" Content="Продолжить" TabIndex="1"/>

</StackPanel>

</Border>

</StackPanel>

</Border>

</Grid>

</UserControl>

Страница третьего этапа регистрации содержит поле ввода кода подтверждения, а также кнопки «Продолжить», «Уже есть аккаунт? Войти», «Отправьте код повторно» и «Измените свой email».

Страница третьего этапа регистрации представлена в соответствии с рисунком 6.

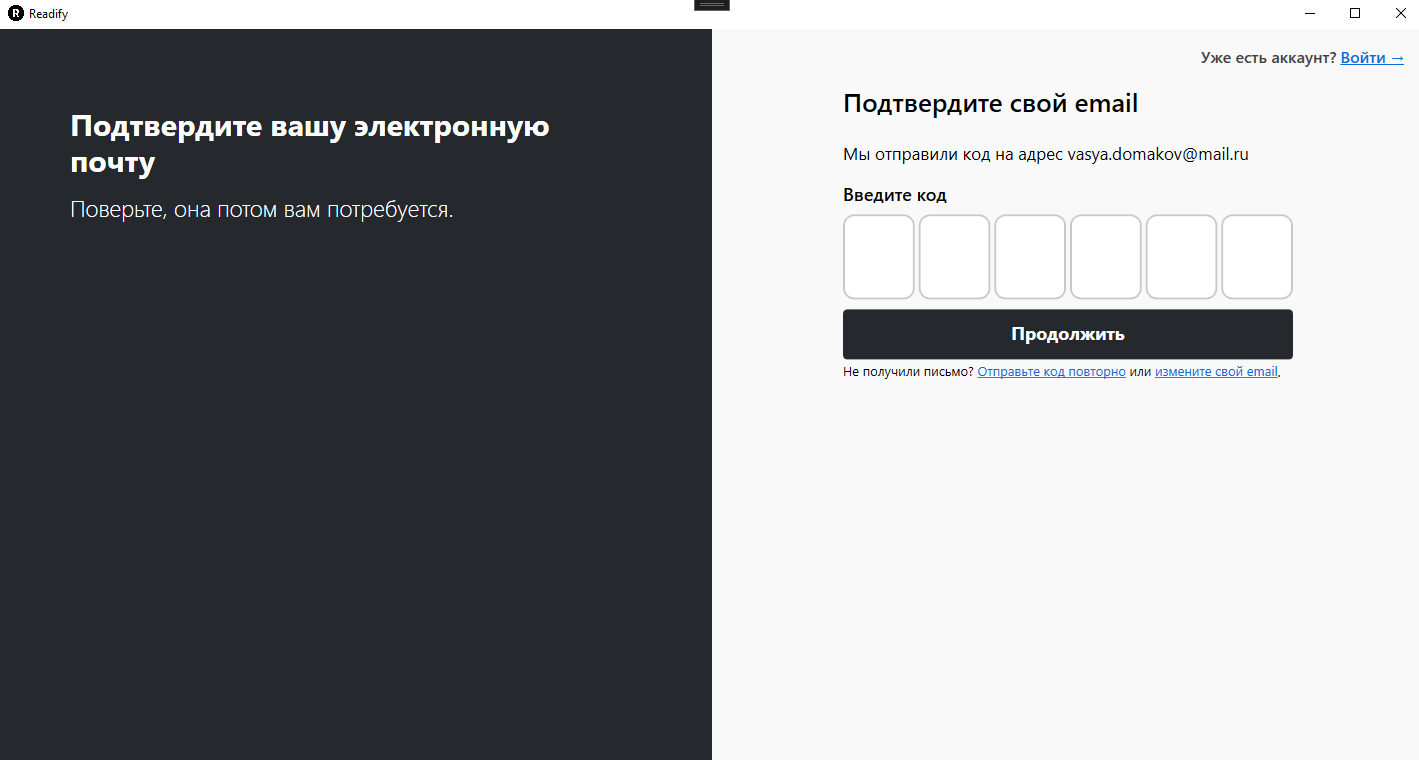


Рисунок – Третий этап регистрации

XAML-код файла «RegistartionEmailPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.Registartion.RegistrationEmailPage" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:local="clr-namespace:Readify.Pages.Registartion" xmlns:i="http://schemas.microsoft.com/xaml/behaviors" xmlns:registration="clr-namespace:Readify.ViewModels.Registration" d:DataContext="{d:DesignInstance Type=registration:RegistrationEmailViewModel}" mc:Ignorable="d" d:DesignHeight="700" d:DesignWidth="1200">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Border Grid.Column="0" Style="{StaticResource LeftBorderRegistartion}">

<StackPanel>

<TextBlock Style="{StaticResource RegistartionLeftHeaderText}" Text="Подтвердите вашу электронную почту"/>

<TextBlock Style="{StaticResource RegistartionLeftText}" Text="Поверьте, она потом вам потребуется."/>

</StackPanel>

</Border>

<Border Grid.Column="1" Style="{StaticResource RightBorderRegistartion}">

<StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right">

<TextBlock Style="{StaticResource RegistrationHelpLinkText}" Text="Уже есть аккаунт?"/>

<TextBlock Style="{StaticResource RegistrationLinkText}" Text="Войти →">

<TextBlock.InputBindings>

<MouseBinding Command="{Binding GoToLoginCommand}" MouseAction="LeftClick"/>

</TextBlock.InputBindings>

</TextBlock>

</StackPanel>

<Border VerticalAlignment="Center" Width="450" Margin="0 20 0 0">

<StackPanel>

<TextBlock Style="{StaticResource RegistrationHeaderText}" Text="Подтвердите свой email"/>

<TextBlock FontFamily="{StaticResource Inter}" FontWeight="Regular" Text="{Binding EmailText}" FontSize="17" TextWrapping="Wrap" Margin="0 10 0 20"/>

<TextBlock FontFamily="{StaticResource Inter}" FontWeight="SemiBold" Text="Введите код" FontSize="18" TextWrapping="Wrap" Margin="0 0 0 8"/>

<UniformGrid Columns="6" Margin="-2 0">

<TextBox x:Name="Digit1" Text="{Binding Code1, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" Style="{StaticResource CodeInputBox}" MaxLength="1" PreviewTextInput="Digit\_PreviewTextInput" PreviewKeyDown="Digit\_PreviewKeyDown"/>

<TextBox x:Name="Digit2" Text="{Binding Code2, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" Style="{StaticResource CodeInputBox}" MaxLength="1" PreviewTextInput="Digit\_PreviewTextInput" PreviewKeyDown="Digit\_PreviewKeyDown"/>

<TextBox x:Name="Digit3" Text="{Binding Code3, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" Style="{StaticResource CodeInputBox}" MaxLength="1" PreviewTextInput="Digit\_PreviewTextInput" PreviewKeyDown="Digit\_PreviewKeyDown"/>

<TextBox x:Name="Digit4" Text="{Binding Code4, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" Style="{StaticResource CodeInputBox}" MaxLength="1" PreviewTextInput="Digit\_PreviewTextInput" PreviewKeyDown="Digit\_PreviewKeyDown"/>

<TextBox x:Name="Digit5" Text="{Binding Code5, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" Style="{StaticResource CodeInputBox}" MaxLength="1" PreviewTextInput="Digit\_PreviewTextInput" PreviewKeyDown="Digit\_PreviewKeyDown"/>

<TextBox x:Name="Digit6" Text="{Binding Code6, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" Style="{StaticResource CodeInputBox}" MaxLength="1" PreviewTextInput="Digit\_PreviewTextInput" PreviewKeyDown="Digit\_PreviewKeyDown"/>

</UniformGrid>

<TextBlock FontFamily="{StaticResource Inter}" Foreground="#7D7D7D" TextWrapping="Wrap" FontSize="18" Margin="0 10 0 5" Text="{Binding ErrorMessage}" Visibility="{Binding IsErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" FontWeight="SemiBold"/>

<Button Style="{StaticResource RegistrationButton}" Command="{Binding ContinueCommand}" Content="Продолжить" TabIndex="1"/>

<WrapPanel Orientation="Horizontal">

<TextBlock FontFamily="{StaticResource Inter}" Margin="0 5 0 0" FontWeight="Regular" FontSize="13" Text="Не получили письмо? " TextWrapping="Wrap"/>

<TextBlock FontFamily="{StaticResource Inter}" Margin="0 5 0 0" FontWeight="Regular" Cursor="Hand" TextWrapping="Wrap" TextDecorations="Underline" FontSize="13" Text="Отправьте код повторно">

<TextBlock.InputBindings>

<MouseBinding Command="{Binding SendMailCommand}" MouseAction="LeftClick"/>

</TextBlock.InputBindings>

<TextBlock.Style>

<Style TargetType="TextBlock">

<Setter Property="Foreground" Value="#0B6BD7"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Foreground" Value="Red"/>

</Trigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

</TextBlock>

<TextBlock FontFamily="{StaticResource Inter}" Margin="0 5 0 0" FontWeight="Regular" FontSize="13" Text=" или " TextWrapping="Wrap"/>

<TextBlock FontFamily="{StaticResource Inter}" Margin="0 5 0 0" FontWeight="Regular" Cursor="Hand" TextWrapping="Wrap" TextDecorations="Underline" FontSize="13" Text="измените свой email">

<TextBlock.InputBindings>

<MouseBinding Command="{Binding GoToRegisterCommand}" MouseAction="LeftClick"/>

</TextBlock.InputBindings>

<TextBlock.Style>

<Style TargetType="TextBlock">

<Setter Property="Foreground" Value="#0B6BD7"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Foreground" Value="Red"/>

</Trigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

</TextBlock>

<TextBlock FontFamily="{StaticResource Inter}" Margin="0 5 0 0" FontWeight="Regular" FontSize="14" Text="." TextWrapping="Wrap"/>

</WrapPanel>

</StackPanel>

</Border>

</StackPanel>

</Border>

</Grid>

</UserControl>

Страница четвертого этапа регистрации содержит такие поля как место для выбора аватара, имя пользователя, описание пользователя а также кнопку «Завершить регистрацию» и кнопку «Пропустить». Страница четвертого этапа регистрации представлена в соответствии с рисунком 7.

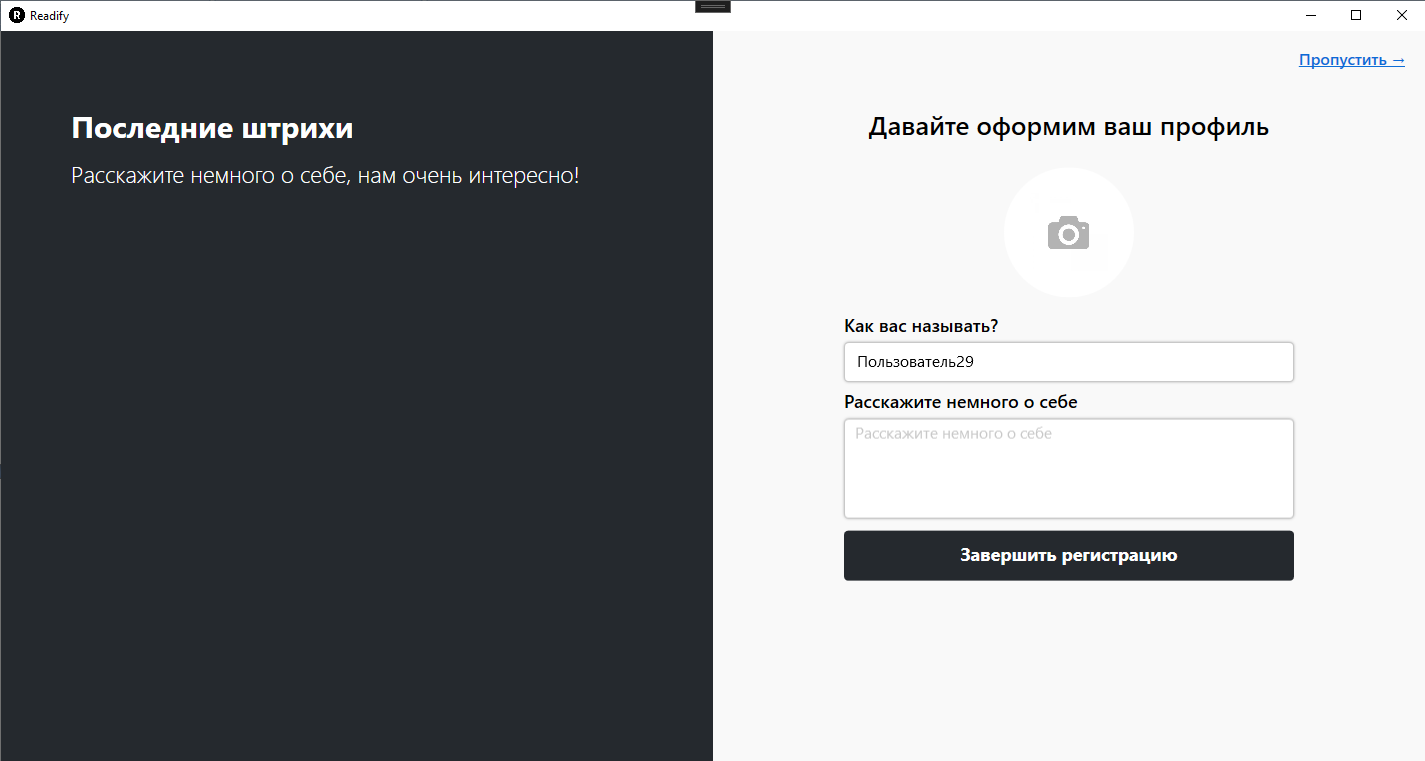


Рисунок – Четвертый этап регистрации

XAML-код файла «RegistrationUpdatePage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.Registartion.RegistrationUpdatePage" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:local="clr-namespace:Readify.Pages.Registartion" xmlns:i="http://schemas.microsoft.com/xaml/behaviors" xmlns:registration="clr-namespace:Readify.ViewModels.Registration" d:DataContext="{d:DesignInstance Type=registration:RegistrationUpdateViewModel}" mc:Ignorable="d" d:DesignHeight="700" d:DesignWidth="1200">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Border Grid.Column="0" Style="{StaticResource LeftBorderRegistartion}">

<StackPanel>

<TextBlock Style="{StaticResource RegistartionLeftHeaderText}" Text="Последние штрихи" />

<TextBlock Style="{StaticResource RegistartionLeftText}" Text="Расскажите немного о себе, нам очень интересно!" />

</StackPanel>

</Border>

<Border Grid.Column="1" Style="{StaticResource RightBorderRegistartion}">

<StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right">

<TextBlock Style="{StaticResource RegistrationLinkText}" Text="Пропустить →">

<TextBlock.InputBindings>

<MouseBinding Command="{Binding GoToProfilePageCommand}" MouseAction="LeftClick" />

</TextBlock.InputBindings>

</TextBlock>

</StackPanel>

<Border VerticalAlignment="Center" Width="450" Margin="0 41 0 0">

<StackPanel>

<TextBlock Style="{StaticResource RegistrationHeaderText}" Text="Давайте оформим ваш профиль" TextAlignment="Center"/>

<Border Width="130" Height="130" CornerRadius="100" Margin="0 10 0 10" HorizontalAlignment="Center" VerticalAlignment="Center" Cursor="Hand">

<Border.Clip>

<EllipseGeometry Center="65,65" RadiusX="65" RadiusY="65"/>

</Border.Clip>

<Grid>

<Image Source="{Binding AvatarBytes, Converter={StaticResource ImageConverter}}" Stretch="UniformToFill" HorizontalAlignment="Center" VerticalAlignment="Center">

<Image.RenderTransform>

<TransformGroup>

<TranslateTransform X="0" Y="0"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

<Border x:Name="HoverOverlay" Background="#40000000" CornerRadius="100" Visibility="Collapsed">

<Border.Clip>

<EllipseGeometry Center="65,65" RadiusX="65" RadiusY="65"/>

</Border.Clip>

</Border>

</Grid>

<Border.Triggers>

<EventTrigger RoutedEvent="Mouse.MouseEnter">

<BeginStoryboard>

<Storyboard>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="HoverOverlay" Storyboard.TargetProperty="Visibility">

<DiscreteObjectKeyFrame KeyTime="0:0:0" Value="{x:Static Visibility.Visible}"/>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

<EventTrigger RoutedEvent="Mouse.MouseLeave">

<BeginStoryboard>

<Storyboard>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="HoverOverlay" Storyboard.TargetProperty="Visibility">

<DiscreteObjectKeyFrame KeyTime="0:0:0" Value="{x:Static Visibility.Collapsed}"/>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

</Border.Triggers>

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding SelectAvatarCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</Border>

<TextBlock Style="{StaticResource RegistrationInputLabel}" Text="Как вас называть?" />

<TextBox Style="{StaticResource RoundedShadowLoginTextBox}" Text="{Binding Name}" Name="Name" MaxLength="100" Tag="Как вас называть?" TabIndex="0" PreviewKeyDown="Enter\_PreviewKeyDown">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding CloseNameErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsNameErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding NameErrorMessage}" />

<TextBlock Style="{StaticResource RegistrationInputLabel}" Text="Расскажите немного о себе" />

<TextBox Style="{StaticResource RoundedShadowAboutTextBox}" Text="{Binding About}" Name="About" Tag="Расскажите немного о себе" TabIndex="1" MaxLength="250" PreviewKeyDown="Enter\_PreviewKeyDown">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding CloseNameErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsAboutErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding AboutErrorMessage}" />

<Button Style="{StaticResource RegistrationButton}" Command="{Binding FinishRegistrationCommand}" Content="Завершить регистрацию" TabIndex="2"/>

</StackPanel>

</Border>

</StackPanel>

</Border>

</Grid>

</UserControl>

Главное окно содержит меню, в котором находится список доступных вошедшему пользователю разделов. При выборе раздела, в главном окне должна появиться соответствующая ему страница.

Главное окно представлено в соответствии с рисунком 8.



Рисунок – Главное окно

XAML-код файла «MainMenuPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.MainMenuPage" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:local="clr-namespace:Readify.Pages" xmlns:i="http://schemas.microsoft.com/xaml/behaviors" xmlns:menu="clr-namespace:Readify.ViewModels" d:DataContext="{d:DesignInstance Type=menu:MainMenuViewModel}" mc:Ignorable="d" d:DesignHeight="700" d:DesignWidth="1200">

<Grid>

<Grid Background="#FFFFFF">

<Grid.RowDefinitions>

<RowDefinition Height="70"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="1150"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<!--Верхняя панель навигации-->

<Border Grid.Row="0" Grid.Column="1" Style="{StaticResource TopBarStyle}">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="200"/>

</Grid.ColumnDefinitions>

<!--Логотип-->

<TextBlock Style="{StaticResource HeaderTextStyle}" FontSize="25" Grid.Column="0" HorizontalAlignment="Left" Text="Readify" Visibility="{Binding IsLogoVisibility, Converter={StaticResource BoolToVisibilityConverter}}">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding GoToProfilePageCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBlock>

<!--Кнопка навигации "Назад"-->

<Border Visibility="{Binding IsBackVisibility, Converter={StaticResource BoolToVisibilityConverter}}" HorizontalAlignment="Left">

<Path Data="M7.78 12.53a.75.75 0 0 1-1.06 0L2.47 8.28a.75.75 0 0 1 0-1.06l4.25-4.25a.751.751 0 0 1 1.042.018.751.751 0 0 1 .018 1.042L4.81 7h7.44a.75.75 0 0 1 0 1.5H4.81l2.97 2.97a.75.75 0 0 1 0 1.06Z" Fill="#25292E" VerticalAlignment="Center" HorizontalAlignment="Center" Height="20" Stretch="Fill" Width="20"/>

<Border.Style>

<Style TargetType="Border" BasedOn="{StaticResource RoundAvatarBorderStyle}">

<Setter Property="Background" Value="#F2F2F2"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#E2E2E2"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding BackFramePageCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</Border>

<!--Элементы меню-->

<StackPanel Grid.Column="1" Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Style="{StaticResource MenuTextStyle}" Text="Поиск" x:Name="SearchHoverRegion" MouseEnter="SearchBorder\_MouseEnter" Cursor="Hand"/>

<TextBlock Style="{StaticResource MenuTextStyle}" Text="Библиотека">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding GoToLibararyPageCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBlock>

<TextBlock Style="{StaticResource MenuTextStyle}" Text="Профиль">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding GoToProfilePageCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBlock>

</StackPanel>

<!--Меню, показывающееся при наведении на область с аватаром-->

<Popup x:Name="UserPopup" Style="{StaticResource UserPopupStyle}" PlacementTarget="{Binding ElementName=UserAvatarBorder}" MouseEnter="UserAvatarBorder\_MouseEnter" MouseLeave="UserAvatarBorder\_MouseLeave">

<Border Style="{StaticResource PopupContentStyle}" MouseEnter="Popup\_MouseEnter" MouseLeave="Popup\_MouseLeave">

<StackPanel Height="113">

<DockPanel Height="60">

<TextBlock Text="{Binding ApplicationUserUsername}" FontFamily="{StaticResource Inter}" FontWeight="Medium" VerticalAlignment="Center" FontSize="20" Margin="15 0 0 0" Width="80" />

<Border Style="{StaticResource RoundAvatarBorderStyle}" HorizontalAlignment="Right" Cursor="Hand" Margin="0 0 40 0">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding GoToProfilePageCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

<Grid>

<Image Source="{Binding ApplicationUserAvatarBytes, Converter={StaticResource ImageConverter}}" Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality" HorizontalAlignment="Center" VerticalAlignment="Center">

<Image.RenderTransform>

<TransformGroup>

<TranslateTransform X="0" Y="0"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

<Border x:Name="HoverOverlay" Background="#90000000" CornerRadius="100" Visibility="Collapsed">

<Border.Clip>

<EllipseGeometry Center="25,25" RadiusX="25" RadiusY="25"/>

</Border.Clip>

</Border>

</Grid>

</Border>

</DockPanel>

<TextBlock Style="{StaticResource LogoutTextStyle}" Text="Выход">

<TextBlock.InputBindings>

<MouseBinding Command="{Binding LogoutCommand}" MouseAction="LeftClick"/>

</TextBlock.InputBindings>

</TextBlock>

</StackPanel>

</Border>

</Popup>

<!--Меню, показывающееся при наведении на поиск-->

<Popup x:Name="SearchPopup" Style="{StaticResource SearchPopupStyle}" PlacementTarget="{Binding ElementName=SearchHoverRegion}" MouseEnter="SearchPopup\_MouseEnter" MouseLeave="SearchPopup\_MouseLeave">

<Border Style="{StaticResource SearchPopupContentStyle}" MouseEnter="SearchPopup\_MouseEnter" MouseLeave="SearchPopup\_MouseLeave" VerticalAlignment="Center">

<StackPanel Height="60" VerticalAlignment="Center" Orientation="Horizontal">

<Path Data="M10.68 11.74a6 6 0 0 1-7.922-8.982 6 6 0 0 1 8.982 7.922l3.04 3.04a.749.749 0 0 1-.326 1.275.749.749 0 0 1-.734-.215ZM11.5 7a4.499 4.499 0 1 0-8.997 0A4.499 4.499 0 0 0 11.5 7Z" Fill="Black" Stretch="Uniform" VerticalAlignment="Center" Margin="15 0 0 0" Width="20" Height="20"/>

<TextBox x:Name="SearchBox" Tag="Поиск" Text="{Binding SearchText, UpdateSourceTrigger=PropertyChanged}" Style="{StaticResource SearchTextBox}">

<TextBox.InputBindings>

<KeyBinding Key="Enter" Command="{Binding SearchCommand}"/>

</TextBox.InputBindings>

</TextBox>

</StackPanel>

</Border>

</Popup>

<!--Область с аватаром-->

<Border Grid.Column="2" x:Name="HoverRegion" Padding="0 0 40 0" Width="200" Height="60" CornerRadius="0 30 30 0" Background="Transparent" MouseEnter="UserAvatarBorder\_MouseEnter" Cursor="Hand" MouseLeave="UserAvatarBorder\_MouseLeave">

<Border x:Name="UserAvatarBorder" Style="{StaticResource RoundAvatarBorderStyle}" HorizontalAlignment="Right">

<Grid>

<Image Source="{Binding ApplicationUserAvatarBytes, Converter={StaticResource ImageConverter}}" Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality" HorizontalAlignment="Center" VerticalAlignment="Center">

<Image.RenderTransform>

<TransformGroup>

<TranslateTransform X="0" Y="0"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

<Border x:Name="HoverOverlay" Background="#90000000" CornerRadius="100" Visibility="Collapsed">

<Border.Clip>

<EllipseGeometry Center="25,25" RadiusX="25" RadiusY="25"/>

</Border.Clip>

</Border>

</Grid>

</Border>

</Border>

</Grid>

</Border>

<!--Дополнительный скроллер для прокрутки в левом гриду-->

<ScrollViewer Style="{StaticResource MainScrollViewerStyle}" x:Name="LeftMainScrollViewer" PreviewMouseWheel="MainScrollViewer\_PreviewMouseWheel" Grid.Column="0" Grid.Row="0" Grid.RowSpan="2"/>

<!--Скроллер основного контента-->

<ScrollViewer Style="{StaticResource MainScrollViewerStyle}" x:Name="MainScrollViewer" PreviewMouseWheel="MainScrollViewer\_PreviewMouseWheel" Grid.Row="1" Grid.Column="1">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="1150"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Frame Grid.Row="1" Grid.Column="1" Name="MainMenuFrame" JournalOwnership="OwnsJournal" Margin="30 20 30 0" NavigationUIVisibility="Hidden" PreviewKeyDown="MainMenuFrame\_PreviewKeyDown"/>

</Grid>

</ScrollViewer>

<!--Дополнительный скроллер для прокрутки в правом гриду-->

<ScrollViewer Style="{StaticResource MainScrollViewerStyle}" x:Name="RightMainScrollViewer" PreviewMouseWheel="MainScrollViewer\_PreviewMouseWheel" Grid.Column="2" Grid.Row="0" Grid.RowSpan="2"/>

</Grid>

</Grid>

</UserControl>

Первое, что видит пользователя после того, как вошел в аккаунт – свой профиль. Поэтому, после того, как главное окно создано, была разработана первая страница –«Профиль». В левой части виден аватар пользователя, имя, логин, количество подписчиков и подписок, и описание пользователя, также есть 2 кнопки «Обновить профиль» и «Опубликовать книгу».

Страница «Профиль» представлена в соответствии с рисунком 9.

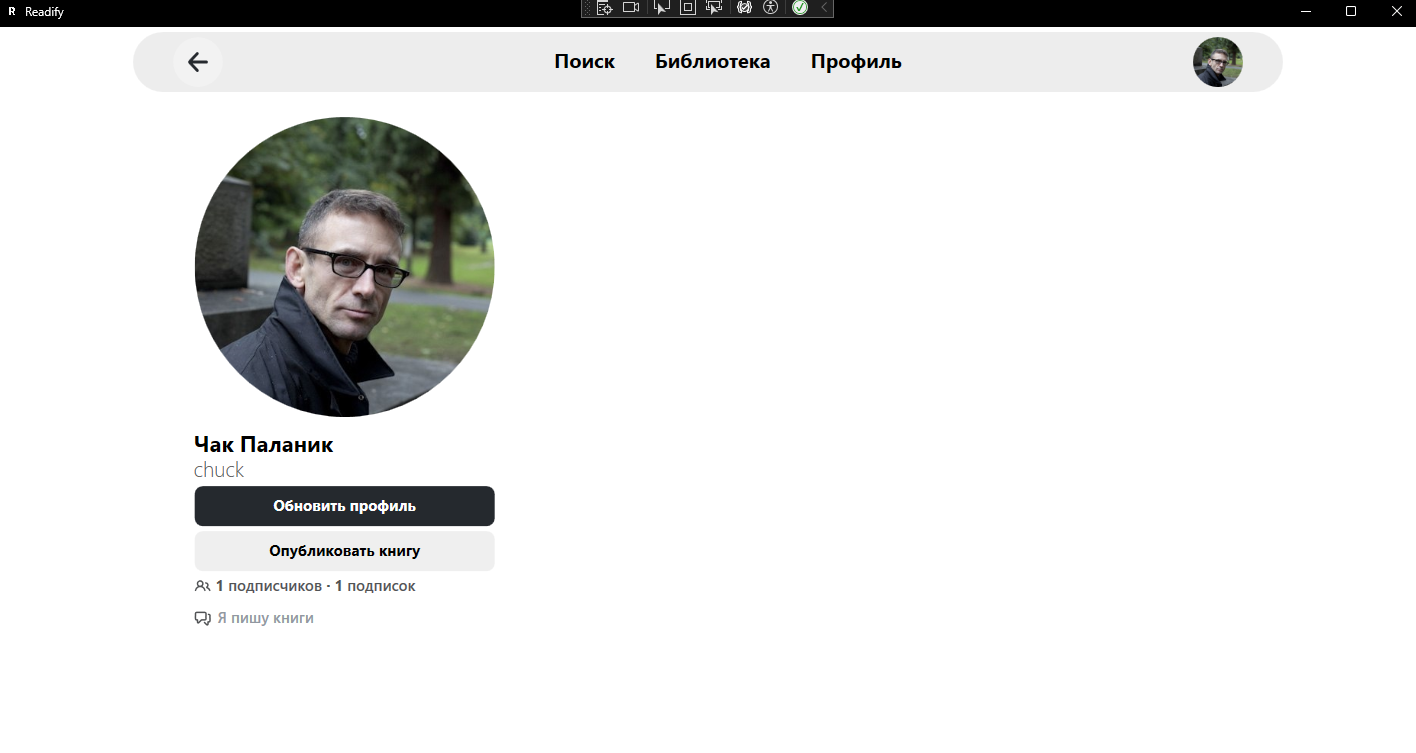


Рисунок – Страница «Профиль»

XAML-код файла «ProfilePage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.MainMenu.ProfilePage" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:local="clr-namespace:Readify.Pages.MainMenu" xmlns:i="http://schemas.microsoft.com/xaml/behaviors" xmlns:main="clr-namespace:Readify.ViewModels.MainMenu" d:DataContext="{d:DesignInstance Type=main:ProfileViewModel}" mc:Ignorable="d" d:DesignHeight="800" d:DesignWidth="1100">

<Grid>

<Grid Grid.Row="0">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<!--Левая часть страницы (информация о пользователе)-->

<Border Style="{StaticResource ProfilePageMainBorderStyle}">

<StackPanel>

<!--Аватар текущего пользователя-->

<Border Style="{StaticResource ProfileAvatarStyle}">

<Grid>

<Image Source="{Binding CurrentUserAvatarBytes, Converter={StaticResource ImageConverter}}" Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality" HorizontalAlignment="Center" VerticalAlignment="Center">

<Image.RenderTransform>

<TransformGroup>

<TranslateTransform X="0" Y="0"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

</Grid>

</Border>

<!--Имя пользователя (Name)-->

<TextBox Margin="-3 15 0 0" Text="{Binding CurrentName}" FontWeight="Bold" FontFamily="{StaticResource Inter}" FontSize="22" IsReadOnly="True" BorderThickness="0" Background="Transparent" Cursor="IBeam"/>

<!--Имя пользователя (Username)-->

<TextBox Text="{Binding CurrentUsername}" FontWeight="Light" FontFamily="{StaticResource Inter}" FontSize="21" Foreground="#595959" IsReadOnly="True" Margin="-3 0 0 0" BorderThickness="0" Background="Transparent" Cursor="IBeam"/>

<!--Кнопка "Обновить профиль"-->

<Button Content="Обновить профиль" Style="{StaticResource ProfileUserButton}" Command="{Binding EditUserCommand}" Visibility="{Binding IsEditButtonVisible, Converter={StaticResource BoolToVisibilityConverter}}"/>

<!--Кнопка "Опубликовать книгу"-->

<Button Content="Опубликовать книгу" Style="{StaticResource UploadBookButton}" Command="{Binding DeployBookCommand}" Visibility="{Binding IsEditButtonVisible, Converter={StaticResource BoolToVisibilityConverter}}"/>

<!--Кнопка "Подписаться"-->

<Button Content="{Binding TextFollowButton}" Style="{StaticResource ProfileUserButton}" Command="{Binding FollowUserCommand}" Visibility="{Binding IsFollowButtonVisible, Converter={StaticResource BoolToVisibilityConverter}}"/>

<!--Кнопка "Отписаться"-->

<Button Content="Отписаться" Style="{StaticResource ProfileUserButton}" Command="{Binding UnfollowUserCommand}" Visibility="{Binding IsUnfollowButtonVisible, Converter={StaticResource BoolToVisibilityConverter}}"/>

<!--Блок с подписчиками и подписками-->

<StackPanel Orientation="Horizontal">

<!--Подписчики-->

<StackPanel Orientation="Horizontal" Cursor="Hand" MouseEnter="Followers\_MouseEnter" MouseLeave="Followers\_MouseLeave">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding GoToFollowersPage}" CommandParameter="1"/>

</i:EventTrigger>

</i:Interaction.Triggers>

<StackPanel.Style>

<Style TargetType="StackPanel">

<Setter Property="TextBlock.Foreground" Value="#515355"/>

<Setter Property="Path.Fill" Value="#9198A1"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="TextBlock.Foreground" Value="Red"/>

<Setter Property="Path.Fill" Value="Red"/>

</Trigger>

</Style.Triggers>

</Style>

</StackPanel.Style>

<!--Иконка (SVG)-->

<Path VerticalAlignment="Center" Fill="#515355" x:Name="FollowersSVG" Data="M2 5.5a3.5 3.5 0 1 1 5.898 2.549 5.508 5.508 0 0 1 3.034 4.084.75.75 0 1 1-1.482.235 4 4 0 0 0-7.9 0 .75.75 0 0 1-1.482-.236A5.507 5.507 0 0 1 3.102 8.05 3.493 3.493 0 0 1 2 5.5ZM11 4a3.001 3.001 0 0 1 2.22 5.018 5.01 5.01 0 0 1 2.56 3.012.749.749 0 0 1-.885.954.752.752 0 0 1-.549-.514 3.507 3.507 0 0 0-2.522-2.372.75.75 0 0 1-.574-.73v-.352a.75.75 0 0 1 .416-.672A1.5 1.5 0 0 0 11 5.5.75.75 0 0 1 11 4Zm-5.5-.5a2 2 0 1 0-.001 3.999A2 2 0 0 0 5.5 3.5Z"/>

<!--Количество подписчиков-->

<TextBlock Margin="5 4 0 0" Text="{Binding CurrentSubscribers}" FontWeight="Bold" FontFamily="{StaticResource Inter}" FontSize="15" VerticalAlignment="Center"/>

<!--Текст "Тодписчиков"-->

<TextBlock Margin="0 4 0 0" Text=" подписчиков" FontWeight="DemiBold" FontFamily="{StaticResource Inter}" FontSize="15" VerticalAlignment="Center" Foreground="#515355"/>

</StackPanel>

<!--Разделяющая точка-->

<TextBlock Margin="0 3 0 0" Text=" · " FontWeight="Bold" FontFamily="{StaticResource Inter}" FontSize="15" VerticalAlignment="Center" Foreground="#515355"/>

<!--Подписки-->

<StackPanel Orientation="Horizontal" Cursor="Hand">

<StackPanel.Style>

<Style TargetType="StackPanel">

<Setter Property="TextBlock.Foreground" Value="#515355"/>

<Setter Property="Path.Fill" Value="#515355"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="TextBlock.Foreground" Value="Red"/>

<Setter Property="Path.Fill" Value="Red"/>

</Trigger>

</Style.Triggers>

</Style>

</StackPanel.Style>

<!--Количество подписчиков-->

<TextBlock Margin="0 4 0 0" Text="{Binding CurrentSubscribtions}" FontWeight="Bold" FontFamily="{StaticResource Inter}" FontSize="15" VerticalAlignment="Center"/>

<!--Текст подписок-->

<TextBlock Margin="0 4 0 0" Text=" подписок" FontWeight="DemiBold" FontFamily="{StaticResource Inter}" FontSize="15" VerticalAlignment="Center"/>

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding GoToFollowingPage}" CommandParameter="1"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</StackPanel>

</StackPanel>

<!--Описание пользователя-->

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<StackPanel Orientation="Horizontal" Margin="0 10 0 0">

<!--SVG-иконка-->

<Path Data="M1.75 1h8.5c.966 0 1.75.784 1.75 1.75v5.5A1.75 1.75 0 0 1 10.25 10H7.061l-2.574 2.573A1.458 1.458 0 0 1 2 11.543V10h-.25A1.75 1.75 0 0 1 0 8.25v-5.5C0 1.784.784 1 1.75 1ZM1.5 2.75v5.5c0 .138.112.25.25.25h1a.75.75 0 0 1 .75.75v2.19l2.72-2.72a.749.749 0 0 1 .53-.22h3.5a.25.25 0 0 0 .25-.25v-5.5a.25.25 0 0 0-.25-.25h-8.5a.25.25 0 0 0-.25.25Zm13 2a.25.25 0 0 0-.25-.25h-.5a.75.75 0 0 1 0-1.5h.5c.966 0 1.75.784 1.75 1.75v5.5A1.75 1.75 0 0 1 14.25 12H14v1.543a1.458 1.458 0 0 1-2.487 1.03L9.22 12.28a.749.749 0 0 1 .326-1.275.749.749 0 0 1 .734.215l2.22 2.22v-2.19a.75.75 0 0 1 .75-.75h1a.25.25 0 0 0 .25-.25Z" Fill="#515355" VerticalAlignment="Top" Margin="0 5 0 0">

</Path>

<!--Текст описания пользователя-->

<TextBox Margin="5 4 0 0" Text="{Binding CurrentDescriptrion}" FontWeight="SemiBold" FontFamily="{StaticResource Inter}" FontSize="15" Foreground="#9198A1" TextWrapping="Wrap" MaxWidth="283" IsReadOnly="True" HorizontalContentAlignment="Left" TextAlignment="Left" BorderThickness="0" Background="Transparent" Cursor="IBeam" Padding="0">

</TextBox>

</StackPanel>

</StackPanel>

</StackPanel>

</Border>

<!--Правая часть страницы (Книги и отзывы пользователя)-->

<Border Grid.Column="1" Padding="30 0" VerticalAlignment="Stretch" HorizontalAlignment="Stretch">

<Frame x:Name="ProfileFrame" NavigationUIVisibility="Hidden" VerticalAlignment="Top" HorizontalAlignment="Stretch"/>

</Border>

</Grid>

</Grid>

</UserControl>

Страница профиля по умолчанию представлена в соответствии с рисунком 10.

Изображение выглядит как текст, снимок экрана, Человеческое лицо, человек

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок – Страница профиля по умолчанию

XAML-код файла «ProfileMainPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.MainMenu.Profile.ProfileMainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Readify.Pages.MainMenu.Profile"

xmlns:i="http://schemas.microsoft.com/xaml/behaviors"

mc:Ignorable="d" d:DesignWidth="800" Height="Auto"

xmlns:main="clr-namespace:Readify.ViewModels.MainMenu.Profile"

d:DataContext="{d:DesignInstance Type=main:ProfileMainViewModel}">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="315"/>

<RowDefinition Height="315"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0">

<DockPanel LastChildFill="True">

<TextBlock Text="Добавленные книги" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="20"/>

</DockPanel>

<ScrollViewer x:Name="BooksHorizontalScroll" Margin="-5 0"

VerticalScrollBarVisibility="Disabled"

HorizontalScrollBarVisibility="Hidden"

PreviewMouseWheel="HorizontalScroll\_PreviewMouseWheel">

<ScrollViewer.Resources>

<Style TargetType="ScrollViewer">

<Setter Property="PanningMode" Value="HorizontalOnly"/>

<Setter Property="UseLayoutRounding" Value="True"/>

<Setter Property="ScrollViewer.IsDeferredScrollingEnabled" Value="True"/>

</Style>

</ScrollViewer.Resources>

<ItemsControl ItemsSource="{Binding CurrentUser.Books}"

Margin="0 10 0 0">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<StackPanel Orientation="Horizontal"/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ItemsControl.ItemTemplate>

<DataTemplate>

<Border Margin="5 0" Width="216" Height="270" Cursor="Hand"

CornerRadius="8" HorizontalAlignment="Center">

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#FFFFFF"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<StackPanel HorizontalAlignment="Center" Width="200">

<Border Height="215" Width="145">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding DataContext.ShowBookCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Book.Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

<Image Source="{Binding Book.CoverImage, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality"

Margin="0 20 0 0" VerticalAlignment="Center" HorizontalAlignment="Center"/>

</Border>

<TextBlock Text="{Binding Book.Name}"

FontFamily="{StaticResource Inter}"

TextAlignment="Center" FontSize="18"

Margin="0 5 0 2" FontWeight="SemiBold"/>

<TextBlock Text="{Binding Book.Author.Nickname}" MaxWidth="145" HorizontalAlignment="Center"

FontFamily="{StaticResource Inter}" FontWeight="Regular"

TextAlignment="Center" FontSize="16">

<TextBlock.Style>

<Style TargetType="TextBlock" BasedOn="{StaticResource {x:Type TextBlock}}">

<Setter Property="TextDecorations" Value="{x:Null}"/>

<Setter Property="Foreground" Value="#7A7A7A"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="TextDecorations" Value="Underline"/>

<Setter Property="Foreground" Value="#FF0000"/>

</Trigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding DataContext.ShowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Book.Author.Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBlock>

</StackPanel>

</Border>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</ScrollViewer>

</StackPanel>

<Border Visibility="{Binding IsNullBooksVisible, Converter={StaticResource BoolToVisibilityConverter}}"

Grid.Row="0" Background="#F8F8F8" CornerRadius="10" Margin="0 35 0 10">

<StackPanel VerticalAlignment="Center">

<Path VerticalAlignment="Center" Fill="#515355" HorizontalAlignment="Center"

Data="M2 2.5A2.5 2.5 0 0 1 4.5 0h8.75a.75.75 0 0 1 .75.75v12.5a.75.75 0 0 1-.75.75h-2.5a.75.75 0 0 1 0-1.5h1.75v-2h-8a1 1 0 0 0-.714 1.7.75.75 0 1 1-1.072 1.05A2.495 2.495 0 0 1 2 11.5Zm10.5-1h-8a1 1 0 0 0-1 1v6.708A2.486 2.486 0 0 1 4.5 9h8ZM5 12.25a.25.25 0 0 1 .25-.25h3.5a.25.25 0 0 1 .25.25v3.25a.25.25 0 0 1-.4.2l-1.45-1.087a.249.249 0 0 0-.3 0L5.4 15.7a.25.25 0 0 1-.4-.2Z"

Stretch="Uniform" Width="50" Height="30"/>

<TextBlock Text="{Binding NullBooksText}" Margin="0 15 0 0" TextWrapping="Wrap"

VerticalAlignment="Center" HorizontalAlignment="Center"

FontSize="20" FontFamily="{StaticResource Inter}"

FontWeight="Medium" Foreground="#535353"/>

</StackPanel>

</Border>

<StackPanel Grid.Row="1">

<DockPanel LastChildFill="True">

<TextBlock Text="Отзывы о книгах" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="20"/>

</DockPanel>

<ScrollViewer x:Name="ReviewsHorizontalScroll" Margin="-5 0"

VerticalScrollBarVisibility="Disabled"

HorizontalScrollBarVisibility="Hidden"

PreviewMouseWheel="HorizontalScroll\_PreviewMouseWheel">

<ScrollViewer.Resources>

<Style TargetType="ScrollViewer">

<Setter Property="PanningMode" Value="HorizontalOnly"/>

<Setter Property="UseLayoutRounding" Value="True"/>

<Setter Property="ScrollViewer.IsDeferredScrollingEnabled" Value="True"/>

</Style>

</ScrollViewer.Resources>

<ItemsControl ItemsSource="{Binding CurrentUser.Reviews}"

Margin="0 10 0 0">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<StackPanel Orientation="Horizontal"/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ItemsControl.ItemTemplate>

<DataTemplate>

<Border Margin="5 0" Width="216" Height="270" Cursor="Hand"

CornerRadius="8" HorizontalAlignment="Center">

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#FFFFFF"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<StackPanel HorizontalAlignment="Center">

<Border Height="215" Width="140">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding DataContext.ShowBookCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Book.Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

<Image Source="{Binding Book.CoverImage, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill"

Margin="0 20 0 0" VerticalAlignment="Center" HorizontalAlignment="Center"/>

</Border>

<TextBlock Text="{Binding Book.Name}"

FontFamily="{StaticResource Inter}"

TextAlignment="Center" FontSize="18"

Margin="0 5 0 0" FontWeight="SemiBold"

TextWrapping="Wrap"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="{Binding Rating}" VerticalAlignment="Center"

FontFamily="{StaticResource Inter}" FontWeight="Bold"

Foreground="#000000" FontSize="17"/>

<Path VerticalAlignment="Center" Fill="#FEC810" HorizontalAlignment="Center"

Data="M8 .25a.75.75 0 0 1 .673.418l1.882 3.815 4.21.612a.75.75 0 0 1 .416 1.279l-3.046 2.97.719 4.192a.751.751 0 0 1-1.088.791L8 12.347l-3.766 1.98a.75.75 0 0 1-1.088-.79l.72-4.194L.818 6.374a.75.75 0 0 1 .416-1.28l4.21-.611L7.327.668A.75.75 0 0 1 8 .25Zm0 2.445L6.615 5.5a.75.75 0 0 1-.564.41l-3.097.45 2.24 2.184a.75.75 0 0 1 .216.664l-.528 3.084 2.769-1.456a.75.75 0 0 1 .698 0l2.77 1.456-.53-3.084a.75.75 0 0 1 .216-.664l2.24-2.183-3.096-.45a.75.75 0 0 1-.564-.41L8 2.694Z"

Stretch="Uniform" Margin="4 0 0 3"/>

</StackPanel>

</StackPanel>

</Border>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</ScrollViewer>

</StackPanel>

<Border Visibility="{Binding IsNullReviewsVisible, Converter={StaticResource BoolToVisibilityConverter}}"

Grid.Row="1" Background="#F8F8F8" CornerRadius="10" Margin="0 35 0 10">

<StackPanel VerticalAlignment="Center">

<Path VerticalAlignment="Center" Fill="#515355" HorizontalAlignment="Center"

Data="M8 .25a.75.75 0 0 1 .673.418l1.882 3.815 4.21.612a.75.75 0 0 1 .416 1.279l-3.046 2.97.719 4.192a.751.751 0 0 1-1.088.791L8 12.347l-3.766 1.98a.75.75 0 0 1-1.088-.79l.72-4.194L.818 6.374a.75.75 0 0 1 .416-1.28l4.21-.611L7.327.668A.75.75 0 0 1 8 .25Zm0 2.445L6.615 5.5a.75.75 0 0 1-.564.41l-3.097.45 2.24 2.184a.75.75 0 0 1 .216.664l-.528 3.084 2.769-1.456a.75.75 0 0 1 .698 0l2.77 1.456-.53-3.084a.75.75 0 0 1 .216-.664l2.24-2.183-3.096-.45a.75.75 0 0 1-.564-.41L8 2.694Z"

Stretch="Uniform" Width="50" Height="30"/>

<TextBlock Text="{Binding NullReviewsText}" Margin="0 15 0 0"

VerticalAlignment="Center" HorizontalAlignment="Center"

TextWrapping="Wrap"

FontSize="20" FontFamily="{StaticResource Inter}"

FontWeight="Medium" Foreground="#535353"/>

</StackPanel>

</Border>

</Grid>

</UserControl>

Пользователь может посмотреть людей, на которых он подписан.

Страница «Подписки» представлена в соответствии с рисунком 11.

Изображение выглядит как текст, снимок экрана, Человеческое лицо, человек

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок – Страница «Подписки»

XAML-код файла «ProfileFollowingPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.MainMenu.Profile.ProfileFollowingPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Readify.Pages.MainMenu.Profile" xmlns:i="http://schemas.microsoft.com/xaml/behaviors"

mc:Ignorable="d"

xmlns:main="clr-namespace:Readify.ViewModels.MainMenu.Profile"

d:DataContext="{d:DesignInstance Type=main:ProfileFollowingViewModel}"

d:DesignHeight="589.437" d:DesignWidth="800">

<StackPanel>

<DockPanel LastChildFill="True">

<TextBlock Text="Подписки пользователя" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="20"/>

<TextBlock Text="Вернуться в профиль →" TextDecorations="Underline" FontFamily="{StaticResource Inter}"

FontWeight="Regular" FontSize="20" HorizontalAlignment="Right"

Cursor="Hand">

<TextBlock.Style>

<Style TargetType="TextBlock">

<Setter Property="Foreground" Value="#0B6BD7"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Foreground" Value="Red"/>

</Trigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding GoToProfileCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBlock>

</DockPanel>

<ItemsControl ItemsSource="{Binding CurrentUser.Subscriptions}" x:Name="Followers">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<StackPanel Orientation="Vertical"/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ItemsControl.ItemTemplate>

<DataTemplate>

<StackPanel>

<Border Height="120" Background="Transparent" Padding="0 15">

<DockPanel>

<StackPanel Orientation="Horizontal">

<Border x:Name="UserAvatarBorder"

Style="{StaticResource RoundAvatarFollowBorderStyle}"

HorizontalAlignment="Right">

<Grid>

<Image Source="{Binding AvatarImage, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill"

RenderOptions.BitmapScalingMode="HighQuality"

HorizontalAlignment="Center"

VerticalAlignment="Center">

<Image.RenderTransform>

<TransformGroup>

<TranslateTransform X="0" Y="0"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

</Grid>

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding DataContext.ShowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</Border>

<StackPanel Orientation="Vertical" Margin="10 15 0 0">

<StackPanel Orientation="Horizontal"

VerticalAlignment="Bottom"

Height="22">

<TextBlock Style="{StaticResource FollowersHeaderTextBlockStyle}"

Text="{Binding Name}">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding DataContext.ShowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBlock>

<TextBlock Style="{StaticResource FollowersSubheaderTextBlockStyle}"

Text="{Binding Nickname}">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding DataContext.ShowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBlock>

</StackPanel>

<TextBox Style="{StaticResource FollowersDescriptionTextBlockStyle}"

Text="{Binding Description}"/>

</StackPanel>

</StackPanel>

<StackPanel>

<TextBlock Text="Это вы" HorizontalAlignment="Right" Margin="0 17 0 0" Visibility="{Binding IsYouTexVisible, Converter={StaticResource BoolToVisibilityConverter}}"

FontFamily="{StaticResource Inter}" Foreground="#515355" FontWeight="SemiBold" FontSize="15"></TextBlock>

<Button Style="{StaticResource SubscriptionButtonStyle}" Content="Подписаться"

Visibility="{Binding IsFollowButtonVisible, Converter={StaticResource BoolToVisibilityConverter}}"

Command="{Binding DataContext.FollowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Id}"/>

<Button Style="{StaticResource SubscriptionButtonStyle}" Content="Отписаться"

Visibility="{Binding IsUnfollowButtonVisible, Converter={StaticResource BoolToVisibilityConverter}}"

Command="{Binding DataContext.UnfollowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Id}"/>

</StackPanel>

</DockPanel>

</Border>

<Border Height="0.2" Background="#1D1F22"/>

</StackPanel>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

<Border Visibility="{Binding IsNullFollowVisible, Converter={StaticResource BoolToVisibilityConverter}}"

Grid.Row="0" Background="#F8F8F8" CornerRadius="10" Height="400">

<StackPanel VerticalAlignment="Center">

<Path VerticalAlignment="Center" Fill="#515355" x:Name="FollowersSVG" HorizontalAlignment="Center"

Data="M2 5.5a3.5 3.5 0 1 1 5.898 2.549 5.508 5.508 0 0 1 3.034 4.084.75.75 0 1 1-1.482.235 4 4 0 0 0-7.9 0 .75.75 0 0 1-1.482-.236A5.507 5.507 0 0 1 3.102 8.05 3.493 3.493 0 0 1 2 5.5ZM11 4a3.001 3.001 0 0 1 2.22 5.018 5.01 5.01 0 0 1 2.56 3.012.749.749 0 0 1-.885.954.752.752 0 0 1-.549-.514 3.507 3.507 0 0 0-2.522-2.372.75.75 0 0 1-.574-.73v-.352a.75.75 0 0 1 .416-.672A1.5 1.5 0 0 0 11 5.5.75.75 0 0 1 11 4Zm-5.5-.5a2 2 0 1 0-.001 3.999A2 2 0 0 0 5.5 3.5Z"

Stretch="Uniform" Width="50" Height="30"/>

<TextBlock Text="{Binding NullText}" MaxWidth="500" TextWrapping="Wrap" TextAlignment="Center"

VerticalAlignment="Center" HorizontalAlignment="Center" Visibility="{Binding IsNullFollowVisible, Converter={StaticResource BoolToVisibilityConverter}}"

FontSize="25" FontFamily="{StaticResource Inter}"

FontWeight="Medium" Foreground="#535353" Margin="0 15 0 0"/>

</StackPanel>

</Border>

</StackPanel>

</UserControl>

Пользователь может посмотреть людей, которые на него подписаны.

Страница «Подписчики» представлена в соответствии с рисунком 12.

Изображение выглядит как текст, снимок экрана, Человеческое лицо, человек

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок – Страница «Подписчики»

XAML-код файла «ProfileFollowersPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.MainMenu.Profile.ProfileFollowersPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Readify.Pages.MainMenu.Profile"

xmlns:i="http://schemas.microsoft.com/xaml/behaviors"

xmlns:main="clr-namespace:Readify.ViewModels.MainMenu.Profile"

d:DataContext="{d:DesignInstance Type=main:ProfileFollowersViewModel}"

mc:Ignorable="d" d:DesignWidth="800" Height="Auto">

<StackPanel>

<DockPanel LastChildFill="True">

<TextBlock Text="Подписчики пользователя" FontFamily="{StaticResource Inter}" FontWeight="Bold" FontSize="20"/>

<TextBlock Text="Вернуться в профиль →" TextDecorations="Underline" FontFamily="{StaticResource Inter}" FontWeight="Regular" FontSize="20" HorizontalAlignment="Right" Cursor="Hand">

<TextBlock.Style>

<Style TargetType="TextBlock">

<Setter Property="Foreground" Value="#0B6BD7"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Foreground" Value="Red"/>

</Trigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding GoToProfileCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBlock>

</DockPanel>

<ItemsControl ItemsSource="{Binding CurrentUser.Subscribers}" x:Name="Followers">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<StackPanel Orientation="Vertical"/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ItemsControl.ItemTemplate>

<DataTemplate>

<StackPanel>

<Border Height="120" Background="Transparent" Padding="0 15">

<DockPanel>

<StackPanel Orientation="Horizontal">

<Border x:Name="UserAvatarBorder" Style="{StaticResource RoundAvatarFollowBorderStyle}" HorizontalAlignment="Right">

<Grid>

<Image Source="{Binding AvatarImage, Converter={StaticResource ImageConverter}}" Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality" HorizontalAlignment="Center" VerticalAlignment="Center">

<Image.RenderTransform>

<TransformGroup>

<TranslateTransform X="0" Y="0"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

</Grid>

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding DataContext.ShowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}" CommandParameter="{Binding Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</Border>

<StackPanel Orientation="Vertical" Margin="10 15 0 0">

<StackPanel Orientation="Horizontal" VerticalAlignment="Bottom" Height="22">

<TextBlock Style="{StaticResource FollowersHeaderTextBlockStyle}" Text="{Binding Name}">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding DataContext.ShowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}" CommandParameter="{Binding Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBlock>

<TextBlock Style="{StaticResource FollowersSubheaderTextBlockStyle}" Text="{Binding Nickname}">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding DataContext.ShowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}" CommandParameter="{Binding Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBlock>

</StackPanel>

<TextBox Style="{StaticResource FollowersDescriptionTextBlockStyle}" Text="{Binding Description}"/>

</StackPanel>

</StackPanel>

<StackPanel>

<TextBlock Text="Это вы" HorizontalAlignment="Right" Margin="0 17 0 0" Visibility="{Binding IsYouTexVisible, Converter={StaticResource BoolToVisibilityConverter}}" FontFamily="{StaticResource Inter}" Foreground="#515355" FontWeight="SemiBold" FontSize="15"/>

<Button Style="{StaticResource SubscriptionButtonStyle}" Content="Подписаться" Visibility="{Binding IsFollowButtonVisible, Converter={StaticResource BoolToVisibilityConverter}}" Command="{Binding DataContext.FollowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}" CommandParameter="{Binding Id}"/>

<Button Style="{StaticResource SubscriptionButtonStyle}" Content="Отписаться" Visibility="{Binding IsUnfollowButtonVisible, Converter={StaticResource BoolToVisibilityConverter}}" Command="{Binding DataContext.UnfollowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}" CommandParameter="{Binding Id}"/>

</StackPanel>

</DockPanel>

</Border>

<Border Height="0.2" Background="#1D1F22"/>

</StackPanel>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

<Border Visibility="{Binding IsNullFollowVisible, Converter={StaticResource BoolToVisibilityConverter}}" Grid.Row="0" Background="#F8F8F8" CornerRadius="10" Height="400">

<StackPanel VerticalAlignment="Center">

<Path VerticalAlignment="Center" Fill="#515355" x:Name="FollowersSVG" HorizontalAlignment="Center" Data="M2 5.5a3.5 3.5 0 1 1 5.898 2.549 5.508 5.508 0 0 1 3.034 4.084.75.75 0 1 1-1.482.235 4 4 0 0 0-7.9 0 .75.75 0 0 1-1.482-.236A5.507 5.507 0 0 1 3.102 8.05 3.493 3.493 0 0 1 2 5.5ZM11 4a3.001 3.001 0 0 1 2.22 5.018 5.01 5.01 0 0 1 2.56 3.012.749.749 0 0 1-.885.954.752.752 0 0 1-.549-.514 3.507 3.507 0 0 0-2.522-2.372.75.75 0 0 1-.574-.73v-.352a.75.75 0 0 1 .416-.672A1.5 1.5 0 0 0 11 5.5.75.75 0 0 1 11 4Zm-5.5-.5a2 2 0 1 0-.001 3.999A2 2 0 0 0 5.5 3.5Z" Stretch="Uniform" Width="50" Height="30"/>

<TextBlock Text="{Binding NullText}" MaxWidth="500" TextWrapping="Wrap" TextAlignment="Center" VerticalAlignment="Center" HorizontalAlignment="Center" Visibility="{Binding IsNullFollowVisible, Converter={StaticResource BoolToVisibilityConverter}}" FontSize="25" FontFamily="{StaticResource Inter}" FontWeight="Medium" Foreground="#535353" Margin="0 15 0 0"/>

</StackPanel>

</Border>

</StackPanel>

</UserControl>

Пользователь может изменять свой профиль.

Страница изменения профиля представлена в соответствии с рисунком 13.

Изображение выглядит как текст, Человеческое лицо, снимок экрана, человек

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок – Страница изменения профиля

XAML-код файла «UpdateUserPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.MainMenu.UpdateUserPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Readify.Pages.MainMenu"

xmlns:i="http://schemas.microsoft.com/xaml/behaviors"

xmlns:main="clr-namespace:Readify.ViewModels.MainMenu"

d:DataContext="{d:DesignInstance Type=main:UpdateUserViewModel}"

mc:Ignorable="d"

d:DesignHeight="583.662" d:DesignWidth="800">

<Grid>

<Border Padding="0 10 0 0">

<StackPanel Orientation="Vertical" HorizontalAlignment="Center">

<DockPanel Margin="0 0 0 50">

<TextBlock Text="Обновление профиля" HorizontalAlignment="Left" FontFamily="{StaticResource Inter}" FontSize="25" FontWeight="Medium" DockPanel.Dock="Left"/>

</DockPanel>

<DockPanel>

<Border Width="400" Height="400" VerticalAlignment="Top" CornerRadius="100" Margin="0 0 50 0" Background="#F5F5F5" HorizontalAlignment="Center" Cursor="Hand">

<Border.Clip>

<EllipseGeometry Center="200,200" RadiusX="200" RadiusY="200"/>

</Border.Clip>

<Grid>

<Image Source="{Binding UserAvatarBytes, Converter={StaticResource ImageConverter}}" Stretch="UniformToFill" HorizontalAlignment="Center" VerticalAlignment="Center">

<Image.RenderTransform>

<TransformGroup>

<TranslateTransform X="0" Y="0"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

<Border x:Name="HoverOverlay" Background="#40000000" CornerRadius="100" Visibility="Collapsed">

<Border.Clip>

<EllipseGeometry Center="200,200" RadiusX="200" RadiusY="200"/>

</Border.Clip>

</Border>

</Grid>

<Border.Triggers>

<EventTrigger RoutedEvent="Mouse.MouseEnter">

<BeginStoryboard>

<Storyboard>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="HoverOverlay" Storyboard.TargetProperty="Visibility">

<DiscreteObjectKeyFrame KeyTime="0:0:0" Value="{x:Static Visibility.Visible}"/>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

<EventTrigger RoutedEvent="Mouse.MouseLeave">

<BeginStoryboard>

<Storyboard>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="HoverOverlay" Storyboard.TargetProperty="Visibility">

<DiscreteObjectKeyFrame KeyTime="0:0:0" Value="{x:Static Visibility.Collapsed}"/>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

</Border.Triggers>

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding SelectAvatarCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</Border>

<Border HorizontalAlignment="Center" VerticalAlignment="Center">

<StackPanel>

<TextBlock Text="Имя пользователя" Margin="2 0 0 5" FontFamily="{StaticResource Inter}" FontWeight="SemiBold" FontSize="20"/>

<TextBox Style="{StaticResource UpdateRoundedShadowTextBox}" Text="{Binding UserNickname}" Name="Username" MaxLength="50" Tag="Имя пользователя" TabIndex="0">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding CloseNicknameErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsUserNicknameErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding UserNicknameErrorText}" FontWeight="SemiBold" Margin="2 0 0 2" FontSize="14"/>

<TextBlock Text="Имя" Margin="2 5 0 5" FontFamily="{StaticResource Inter}" FontWeight="SemiBold" FontSize="20"/>

<TextBox Style="{StaticResource UpdateRoundedShadowTextBox}" Text="{Binding UserName}" Name="Name" MaxLength="100" Tag="Имя" TabIndex="0">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding CloseNameErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsUserNameErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding UserNameErrorText}" FontWeight="SemiBold" Margin="2 0 0 2" FontSize="14"/>

<TextBlock Text="Описание" Margin="2 5 0 5" FontFamily="{StaticResource Inter}" FontWeight="SemiBold" FontSize="20"/>

<TextBox Style="{StaticResource UpdateRoundedShadowAboutTextBox}" Text="{Binding UserDescription}" Name="About" Tag="Описание" TabIndex="1" MaxLength="250">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding CloseNameErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<Button Style="{StaticResource RegistrationButton}" Content="Обновить профиль" Command="{Binding UpdateProfileCommand}"/>

</StackPanel>

</Border>

</DockPanel>

</StackPanel>

</Border>

</Grid>

</UserControl>

Пользователь может публиковать книги.

Страница публикации книги представлена в соответствии с рисунком 14.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок – Страница публикации книги

XAML-код файла «DeployBookPage.xaml» представлен ниже:

<Page x:Class="Readify.Pages.MainMenu.DeployBookPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Readify.Pages.MainMenu" xmlns:i="http://schemas.microsoft.com/xaml/behaviors"

mc:Ignorable="d"

xmlns:main="clr-namespace:Readify.ViewModels.MainMenu"

d:DataContext="{d:DesignInstance Type=main:DeployBookViewModel}"

d:DesignHeight="732.203" d:DesignWidth="1129.48"

Title="DeployBookPage">

<Grid>

<Border Padding="0 10 0 0">

<StackPanel Orientation="Vertical" HorizontalAlignment="Center">

<DockPanel Margin="0 0 0 50">

<TextBlock Text="Публикация книги" HorizontalAlignment="Left" FontFamily="{StaticResource Inter}" FontSize="25" FontWeight="Medium" DockPanel.Dock="Left"/>

</DockPanel>

<DockPanel>

<StackPanel>

<Border Width="300" Height="390" VerticalAlignment="Top" CornerRadius="0" Margin="0 0 50 0" Background="#F5F5F5" HorizontalAlignment="Center" Cursor="Hand" BorderBrush="Black" BorderThickness="0.5">

<Grid>

<Image Source="{Binding BookCoverBytes, Converter={StaticResource ImageConverter}}" Stretch="UniformToFill" HorizontalAlignment="Center" VerticalAlignment="Center"/>

<Border x:Name="HoverOverlay" Background="#40000000" CornerRadius="0"/>

</Grid>

<Border.Triggers>

<EventTrigger RoutedEvent="Mouse.MouseEnter">

<BeginStoryboard>

<Storyboard>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="HoverOverlay" Storyboard.TargetProperty="Visibility">

<DiscreteObjectKeyFrame KeyTime="0:0:0" Value="{x:Static Visibility.Visible}"/>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

<EventTrigger RoutedEvent="Mouse.MouseLeave">

<BeginStoryboard>

<Storyboard>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="HoverOverlay" Storyboard.TargetProperty="Visibility">

<DiscreteObjectKeyFrame KeyTime="0:0:0" Value="{x:Static Visibility.Collapsed}"/>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

</Border.Triggers>

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding SelectCoverCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</Border>

<Border Width="300" Height="50" VerticalAlignment="Top" CornerRadius="5" Margin="0 15 50 0" Background="#F5F5F5" HorizontalAlignment="Center" Cursor="Hand">

<Grid>

<Border x:Name="HoverFileOverlay" Background="#40000000" CornerRadius="0" Visibility="Collapsed"/>

<TextBlock FontFamily="{StaticResource Inter}" FontWeight="SemiBold" Text="{Binding SelectFileText}" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" Foreground="Black"/>

</Grid>

<Border.Triggers>

<EventTrigger RoutedEvent="Mouse.MouseEnter">

<BeginStoryboard>

<Storyboard>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="HoverFileOverlay" Storyboard.TargetProperty="Visibility">

<DiscreteObjectKeyFrame KeyTime="0:0:0" Value="{x:Static Visibility.Visible}"/>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

<EventTrigger RoutedEvent="Mouse.MouseLeave">

<BeginStoryboard>

<Storyboard>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="HoverFileOverlay" Storyboard.TargetProperty="Visibility">

<DiscreteObjectKeyFrame KeyTime="0:0:0" Value="{x:Static Visibility.Collapsed}"/>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

</Border.Triggers>

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction Command="{Binding SelectFileCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</Border>

</StackPanel>

<Border HorizontalAlignment="Center" VerticalAlignment="Center" Width="500">

<StackPanel>

<TextBlock Text="Название книги" Margin="2 0 0 5" FontFamily="{StaticResource Inter}" FontWeight="SemiBold" FontSize="20"/>

<TextBox Style="{StaticResource UpdateRoundedShadowTextBox}" Text="{Binding BookName}" Name="Username" MaxLength="200" Tag="Название книги" TabIndex="0" Width="500">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding CloseNameErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsBookNameErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding BookNameErrorText}" FontWeight="SemiBold" Margin="2 0 0 2" FontSize="14"/>

<TextBlock Text="Жанр книги" Margin="2 5 0 5" FontFamily="{StaticResource Inter}" FontWeight="SemiBold" FontSize="20"/>

<ComboBox Style="{StaticResource UpdateRoundedShadowComboBox}" ItemsSource="{Binding Genres}" SelectedItem="{Binding SelectedGenre}" DisplayMemberPath="Name" Name="GenreSelector" Width="500" Tag="Выберите жанр" TabIndex="0"/>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsGenreErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding GenreErrorText}" FontWeight="SemiBold" Margin="2 0 0 2" FontSize="14"/>

<TextBlock Text="Описание" Margin="2 5 0 5" FontFamily="{StaticResource Inter}" FontWeight="SemiBold" FontSize="20"/>

<TextBox Style="{StaticResource UpdateRoundedShadowAboutTextBox}" Text="{Binding BookDescription}" Name="About" Tag="Описание" TabIndex="1" MaxLength="1000" Width="500">

<i:Interaction.Triggers>

<i:EventTrigger EventName="GotFocus">

<i:InvokeCommandAction Command="{Binding CloseDescriptionErrorCommand}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</TextBox>

<TextBlock Style="{StaticResource RegistrationErrorText}" Visibility="{Binding IsBookDescriptionErrorVisible, Converter={StaticResource BoolToVisibilityConverter}}" Text="{Binding BookDescriptionErrorText}" FontWeight="SemiBold" Margin="2 0 0 2" FontSize="14"/>

<Button Style="{StaticResource RegistrationButton}" Content="Опубликовать книгу" Command="{Binding DeployBookCommand}"/>

</StackPanel>

</Border>

</DockPanel>

</StackPanel>

</Border>

</Grid>

</Page>

Пользователь может просматривать свою библиотеку.

Страница библиотеки представлена в соответствии с рисунком 15.

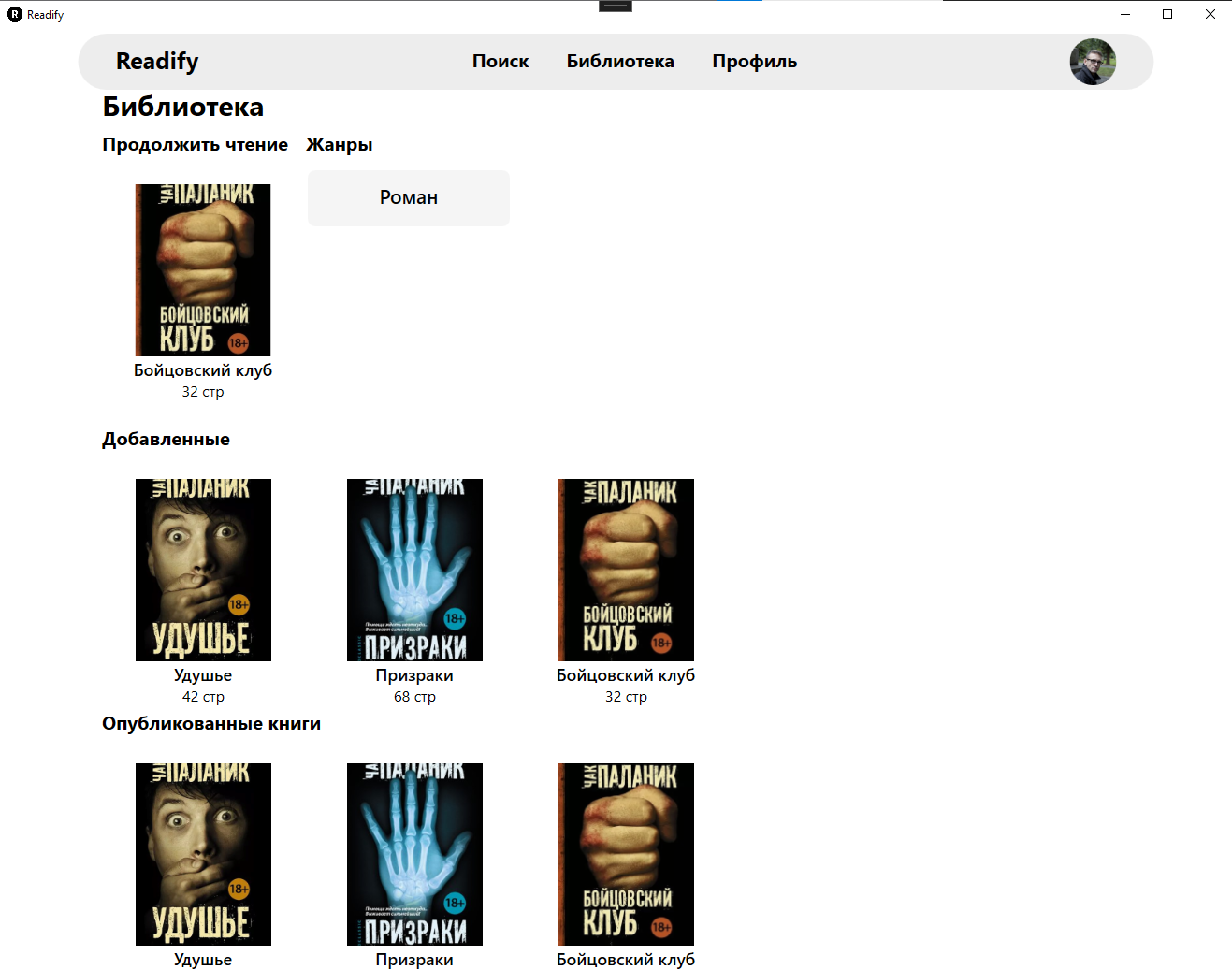


Рисунок – Страница библиотеки

XAML-код файла «LibraryPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.MainMenu.LibraryPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Readify.Pages.MainMenu" xmlns:i="http://schemas.microsoft.com/xaml/behaviors"

mc:Ignorable="d"

xmlns:main="clr-namespace:Readify.ViewModels.MainMenu"

d:DataContext="{d:DesignInstance Type=main:LibraryViewModel}"

d:DesignHeight="1318" d:DesignWidth="1090">

<Grid>

<Border Margin="-5 0">

<StackPanel>

<TextBlock Text="Библиотека" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="30" Margin="0 0 0 10"/>

<DockPanel Height="315">

<StackPanel Orientation="Vertical" VerticalAlignment="Top">

<TextBlock Text="Продолжить чтение" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="20" Margin="0 0 0 10"/>

<Border Margin="0 0 0 0" Width="216" Height="270" Cursor="Hand"

CornerRadius="8" HorizontalAlignment="Center"

Visibility="{Binding IsLastBookVisible, Converter={StaticResource BoolToVisibilityConverter}}">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding ShowBookCommand}"

CommandParameter="{Binding LastBook.Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#FFFFFF"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<StackPanel HorizontalAlignment="Center" Width="200">

<Border Height="204" Width="145">

<Image Margin="0 20 0 0" Source="{Binding LastBook.ImageBytes, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality"

VerticalAlignment="Center" HorizontalAlignment="Center"/>

</Border>

<TextBlock Text="{Binding LastBook.Name}"

FontFamily="{StaticResource Inter}"

TextAlignment="Center" FontSize="18"

Margin="0 5 0 2" FontWeight="SemiBold"/>

<TextBlock Text="{Binding LastBookPageText}" MaxWidth="145" HorizontalAlignment="Center"

FontFamily="{StaticResource Inter}" FontWeight="Regular"

TextAlignment="Center" FontSize="16">

</TextBlock>

</StackPanel>

</Border>

<Border Margin="0 0 0 0" Width="216" Height="270" Cursor="Hand"

CornerRadius="8" HorizontalAlignment="Center"

Visibility="{Binding IsBlockLastBookVisible, Converter={StaticResource BoolToVisibilityConverter}}">

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#F5F5F5"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<StackPanel HorizontalAlignment="Center" Width="200" VerticalAlignment="Center">

<Path VerticalAlignment="Center" Fill="#515355" HorizontalAlignment="Center"

Data="M2 2.5A2.5 2.5 0 0 1 4.5 0h8.75a.75.75 0 0 1 .75.75v12.5a.75.75 0 0 1-.75.75h-2.5a.75.75 0 0 1 0-1.5h1.75v-2h-8a1 1 0 0 0-.714 1.7.75.75 0 1 1-1.072 1.05A2.495 2.495 0 0 1 2 11.5Zm10.5-1h-8a1 1 0 0 0-1 1v6.708A2.486 2.486 0 0 1 4.5 9h8ZM5 12.25a.25.25 0 0 1 .25-.25h3.5a.25.25 0 0 1 .25.25v3.25a.25.25 0 0 1-.4.2l-1.45-1.087a.249.249 0 0 0-.3 0L5.4 15.7a.25.25 0 0 1-.4-.2Z"

Stretch="Uniform" Width="50" Height="30" Margin="0 0 0 10"/>

<TextBlock FontSize="18" Text="Вы не начинали читать ни одну книгу" TextWrapping="Wrap"

FontFamily="{StaticResource Inter}" FontWeight="Medium" TextAlignment="Center"/>

</StackPanel>

</Border>

</StackPanel>

<StackPanel Margin="2 0 0 0">

<TextBlock Text="Жанры" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="20" Margin="0 0 0 10"/>

<ItemsControl ItemsSource="{Binding Genres}" Margin="0">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<WrapPanel Orientation="Horizontal"

ItemWidth="220"

ItemHeight="70"

HorizontalAlignment="Left"/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ItemsControl.ItemTemplate>

<DataTemplate>

<Border Width="216" Height="60" Margin="0 0"

CornerRadius="8" Padding="5 0"

Cursor="Hand">

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#F5F5F5"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<TextBlock Text="{Binding Name}"

FontSize="21"

FontFamily="{StaticResource Inter}"

FontWeight="Medium"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>

</Border>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</StackPanel>

</DockPanel>

<StackPanel>

<StackPanel>

<DockPanel LastChildFill="True">

<TextBlock Text="Не прочитанные книги" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="20"/>

</DockPanel>

<ScrollViewer x:Name="NotFullyReadBookHorizontalScroll" Margin="-5 0"

VerticalScrollBarVisibility="Disabled" Visibility="{Binding IsNotReadVisible, Converter={StaticResource BoolToVisibilityConverter}}"

HorizontalScrollBarVisibility="Hidden"

PreviewMouseWheel="HorizontalScroll\_PreviewMouseWheel">

<ScrollViewer.Resources>

<Style TargetType="ScrollViewer">

<Setter Property="PanningMode" Value="HorizontalOnly"/>

<Setter Property="UseLayoutRounding" Value="True"/>

<Setter Property="ScrollViewer.IsDeferredScrollingEnabled" Value="True"/>

</Style>

</ScrollViewer.Resources>

<ItemsControl ItemsSource="{Binding NotFullyReadBooks}"

Margin="0 10 0 0">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<StackPanel Orientation="Horizontal"/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ItemsControl.ItemTemplate>

<DataTemplate>

<Border Margin="5 0" Width="216" Height="270" Cursor="Hand"

CornerRadius="8" HorizontalAlignment="Center">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding DataContext.ShowBookCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#FFFFFF"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<StackPanel HorizontalAlignment="Center" Width="200">

<Border Height="215" Width="145">

<Image Source="{Binding ImageBytes, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality"

Margin="0 20 0 0" VerticalAlignment="Center" HorizontalAlignment="Center"/>

</Border>

<TextBlock Text="{Binding Name}"

FontFamily="{StaticResource Inter}"

TextAlignment="Center" FontSize="18"

Margin="0 5 0 2" FontWeight="SemiBold"/>

<TextBlock Text="{Binding PageProgressText}" MaxWidth="145" HorizontalAlignment="Center"

FontFamily="{StaticResource Inter}" FontWeight="Regular"

TextAlignment="Center" FontSize="16" Foreground="Black">

<TextBlock.Style>

<Style TargetType="TextBlock" BasedOn="{StaticResource {x:Type TextBlock}}">

<Setter Property="TextDecorations" Value="{x:Null}"/>

<Setter Property="Foreground" Value="#7A7A7A"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="TextDecorations" Value="Underline"/>

<Setter Property="Foreground" Value="#FF0000"/>

</Trigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

</TextBlock>

</StackPanel>

</Border>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</ScrollViewer>

</StackPanel>

<Border Height="250" Background="#F5F5F5" CornerRadius="10" Visibility="{Binding IsBlockNotReadVisible, Converter={StaticResource BoolToVisibilityConverter}}">

<StackPanel VerticalAlignment="Center">

<Path VerticalAlignment="Center" Fill="#515355" HorizontalAlignment="Center"

Data="M2 2.5A2.5 2.5 0 0 1 4.5 0h8.75a.75.75 0 0 1 .75.75v12.5a.75.75 0 0 1-.75.75h-2.5a.75.75 0 0 1 0-1.5h1.75v-2h-8a1 1 0 0 0-.714 1.7.75.75 0 1 1-1.072 1.05A2.495 2.495 0 0 1 2 11.5Zm10.5-1h-8a1 1 0 0 0-1 1v6.708A2.486 2.486 0 0 1 4.5 9h8ZM5 12.25a.25.25 0 0 1 .25-.25h3.5a.25.25 0 0 1 .25.25v3.25a.25.25 0 0 1-.4.2l-1.45-1.087a.249.249 0 0 0-.3 0L5.4 15.7a.25.25 0 0 1-.4-.2Z"

Stretch="Uniform" Width="45" Height="45" Margin="0 0 0 10"/>

<TextBlock FontSize="25" Text="Здесь пусто..." TextWrapping="Wrap"

FontFamily="{StaticResource Inter}" FontWeight="Medium" TextAlignment="Center"/>

</StackPanel>

</Border>

<StackPanel>

<DockPanel LastChildFill="True">

<TextBlock Text="Опубликованные книги" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="20"/>

</DockPanel>

<ScrollViewer x:Name="DeployedBookHorizontalScroll" Margin="-5 0"

VerticalScrollBarVisibility="Disabled"

Visibility="{Binding IsDeployedVisible, Converter={StaticResource BoolToVisibilityConverter}}"

HorizontalScrollBarVisibility="Hidden"

PreviewMouseWheel="HorizontalScroll\_PreviewMouseWheel">

<ScrollViewer.Resources>

<Style TargetType="ScrollViewer">

<Setter Property="PanningMode" Value="HorizontalOnly"/>

<Setter Property="UseLayoutRounding" Value="True"/>

<Setter Property="ScrollViewer.IsDeferredScrollingEnabled" Value="True"/>

</Style>

</ScrollViewer.Resources>

<ItemsControl ItemsSource="{Binding DeployedBooks}"

Margin="0 10 0 0">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<StackPanel Orientation="Horizontal"/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ItemsControl.ItemTemplate>

<DataTemplate>

<Border Margin="5 0" Width="216" Height="270" Cursor="Hand"

CornerRadius="8" HorizontalAlignment="Center">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding DataContext.ShowBookCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#FFFFFF"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<StackPanel HorizontalAlignment="Center" Width="200">

<Border Height="215" Width="145">

<Image Source="{Binding ImageBytes, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality"

Margin="0 20 0 0" VerticalAlignment="Center" HorizontalAlignment="Center"/>

</Border>

<TextBlock Text="{Binding Name}" Width="145"

FontFamily="{StaticResource Inter}"

TextAlignment="Center" FontSize="20"

Margin="0 5 0 2" FontWeight="SemiBold"/>

<TextBlock Text="{Binding PageProgressText}" MaxWidth="145" HorizontalAlignment="Center"

FontFamily="{StaticResource Inter}" FontWeight="Regular"

TextAlignment="Center" FontSize="16" Foreground="Black">

<TextBlock.Style>

<Style TargetType="TextBlock" BasedOn="{StaticResource {x:Type TextBlock}}">

<Setter Property="TextDecorations" Value="{x:Null}"/>

<Setter Property="Foreground" Value="#7A7A7A"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="TextDecorations" Value="Underline"/>

<Setter Property="Foreground" Value="#FF0000"/>

</Trigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

</TextBlock>

</StackPanel>

</Border>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</ScrollViewer>

</StackPanel>

<Border Height="250" Background="#F5F5F5" CornerRadius="10" Visibility="{Binding IsBlockDeployedVisible, Converter={StaticResource BoolToVisibilityConverter}}">

<StackPanel VerticalAlignment="Center">

<Path VerticalAlignment="Center" Fill="#515355" HorizontalAlignment="Center"

Data="M2 2.5A2.5 2.5 0 0 1 4.5 0h8.75a.75.75 0 0 1 .75.75v12.5a.75.75 0 0 1-.75.75h-2.5a.75.75 0 0 1 0-1.5h1.75v-2h-8a1 1 0 0 0-.714 1.7.75.75 0 1 1-1.072 1.05A2.495 2.495 0 0 1 2 11.5Zm10.5-1h-8a1 1 0 0 0-1 1v6.708A2.486 2.486 0 0 1 4.5 9h8ZM5 12.25a.25.25 0 0 1 .25-.25h3.5a.25.25 0 0 1 .25.25v3.25a.25.25 0 0 1-.4.2l-1.45-1.087a.249.249 0 0 0-.3 0L5.4 15.7a.25.25 0 0 1-.4-.2Z"

Stretch="Uniform" Width="45" Height="45" Margin="0 0 0 10"/>

<TextBlock FontSize="25" Text="Здесь пусто..." TextWrapping="Wrap"

FontFamily="{StaticResource Inter}" FontWeight="Medium" TextAlignment="Center"/>

</StackPanel>

</Border>

<StackPanel>

<DockPanel LastChildFill="True">

<TextBlock Text="Прочитано" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="20"/>

</DockPanel>

<ScrollViewer x:Name="FullyReadBookHorizontalScroll" Margin="-5 0"

VerticalScrollBarVisibility="Disabled"

HorizontalScrollBarVisibility="Hidden"

Visibility="{Binding IsReadVisible, Converter={StaticResource BoolToVisibilityConverter}}"

PreviewMouseWheel="HorizontalScroll\_PreviewMouseWheel">

<ScrollViewer.Resources>

<Style TargetType="ScrollViewer">

<Setter Property="PanningMode" Value="HorizontalOnly"/>

<Setter Property="UseLayoutRounding" Value="True"/>

<Setter Property="ScrollViewer.IsDeferredScrollingEnabled" Value="True"/>

</Style>

</ScrollViewer.Resources>

<ItemsControl ItemsSource="{Binding FullyReadBooks}"

Margin="0 10 0 0">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<StackPanel Orientation="Horizontal"/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ItemsControl.ItemTemplate>

<DataTemplate>

<Border Margin="5 0" Width="216" Height="270" Cursor="Hand"

CornerRadius="8" HorizontalAlignment="Center">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding DataContext.ShowBookCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#FFFFFF"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<StackPanel HorizontalAlignment="Center" Width="200">

<Border Height="215" Width="145">

<Image Source="{Binding ImageBytes, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality"

Margin="0 20 0 0" VerticalAlignment="Center" HorizontalAlignment="Center"/>

</Border>

<TextBlock Text="{Binding Name}" Width="145"

FontFamily="{StaticResource Inter}"

TextAlignment="Center" FontSize="20"

Margin="0 5 0 2" FontWeight="SemiBold"/>

<TextBlock Text="{Binding PageProgressText}" MaxWidth="145" HorizontalAlignment="Center"

FontFamily="{StaticResource Inter}" FontWeight="Regular"

TextAlignment="Center" FontSize="16" Foreground="Black">

<TextBlock.Style>

<Style TargetType="TextBlock" BasedOn="{StaticResource {x:Type TextBlock}}">

<Setter Property="TextDecorations" Value="{x:Null}"/>

<Setter Property="Foreground" Value="#7A7A7A"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="TextDecorations" Value="Underline"/>

<Setter Property="Foreground" Value="#FF0000"/>

</Trigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

</TextBlock>

</StackPanel>

</Border>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</ScrollViewer>

</StackPanel>

<Border Height="250" Background="#F5F5F5" CornerRadius="10" Visibility="{Binding IsBlockReadVisible, Converter={StaticResource BoolToVisibilityConverter}}">

<StackPanel VerticalAlignment="Center">

<Path VerticalAlignment="Center" Fill="#515355" HorizontalAlignment="Center"

Data="M2 2.5A2.5 2.5 0 0 1 4.5 0h8.75a.75.75 0 0 1 .75.75v12.5a.75.75 0 0 1-.75.75h-2.5a.75.75 0 0 1 0-1.5h1.75v-2h-8a1 1 0 0 0-.714 1.7.75.75 0 1 1-1.072 1.05A2.495 2.495 0 0 1 2 11.5Zm10.5-1h-8a1 1 0 0 0-1 1v6.708A2.486 2.486 0 0 1 4.5 9h8ZM5 12.25a.25.25 0 0 1 .25-.25h3.5a.25.25 0 0 1 .25.25v3.25a.25.25 0 0 1-.4.2l-1.45-1.087a.249.249 0 0 0-.3 0L5.4 15.7a.25.25 0 0 1-.4-.2Z"

Stretch="Uniform" Width="45" Height="45" Margin="0 0 0 10"/>

<TextBlock FontSize="25" Text="Здесь пусто..." TextWrapping="Wrap"

FontFamily="{StaticResource Inter}" FontWeight="Medium" TextAlignment="Center"/>

</StackPanel>

</Border>

</StackPanel>

</StackPanel>

</Border>

</Grid>

</UserControl>

Пользователь может осуществлять поиск по сервису.

Страница поиска представлена в соответствии с рисунком 16.

Изображение выглядит как текст, снимок экрана, кот

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок – Страница поиска

XAML-код файла «SearchPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.MainMenu.SearchPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Readify.Pages.MainMenu" xmlns:i="http://schemas.microsoft.com/xaml/behaviors"

mc:Ignorable="d"

xmlns:main="clr-namespace:Readify.ViewModels.MainMenu"

d:DataContext="{d:DesignInstance Type=main:SearchViewModel}"

d:DesignHeight="800" d:DesignWidth="800">

<Grid>

<Border>

<StackPanel>

<TextBlock Text="Поиск" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="30" Margin="0 0 0 10"/>

<StackPanel Orientation="Horizontal" VerticalAlignment="Center" Margin="0 0 0 10">

<Path Data="M10.68 11.74a6 6 0 0 1-7.922-8.982 6 6 0 0 1 8.982 7.922l3.04 3.04a.749.749 0 0 1-.326 1.275.749.749 0 0 1-.734-.215ZM11.5 7a4.499 4.499 0 1 0-8.997 0A4.499 4.499 0 0 0 11.5 7Z"

Fill="Black" Stretch="Uniform" VerticalAlignment="Center"

Margin="0 0 10 0" Width="25" Height="25"/>

<TextBlock Text="{Binding SearchText}" FontFamily="{StaticResource Inter}"

FontWeight="Medium" FontSize="25"/>

</StackPanel>

<StackPanel>

<StackPanel>

<DockPanel LastChildFill="True" Width="1950">

<TextBlock Text="Книги" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="20"/>

<ComboBox ItemsSource="{Binding Genres}" BorderThickness="0" Background="#FFEDEDED"

SelectedItem="{Binding SelectedGenre}" Visibility="{Binding IsSearchBookVisible, Converter={StaticResource BoolToVisibilityConverter}}"

Width="170" Margin="10 0 0 0" FontFamily="{StaticResource Inter}" FontSize="14" FontWeight="Medium"

Tag="Выберите жанр"/>

</DockPanel>

<ScrollViewer x:Name="SearchBookHorizontalScroll" Margin="-5 0"

VerticalScrollBarVisibility="Disabled" Visibility="{Binding IsSearchBookVisible, Converter={StaticResource BoolToVisibilityConverter}}"

HorizontalScrollBarVisibility="Hidden"

PreviewMouseWheel="HorizontalScroll\_PreviewMouseWheel">

<ScrollViewer.Resources>

<Style TargetType="ScrollViewer">

<Setter Property="PanningMode" Value="HorizontalOnly"/>

<Setter Property="UseLayoutRounding" Value="True"/>

<Setter Property="ScrollViewer.IsDeferredScrollingEnabled" Value="True"/>

</Style>

</ScrollViewer.Resources>

<ItemsControl ItemsSource="{Binding FoundBooks}"

Margin="0 10 0 0">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<StackPanel Orientation="Horizontal"/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ItemsControl.ItemTemplate>

<DataTemplate>

<Border Margin="5 0" Width="216" Height="270" Cursor="Hand"

CornerRadius="8" HorizontalAlignment="Center">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding DataContext.ShowBookCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#FFFFFF"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<StackPanel HorizontalAlignment="Center" Width="200">

<Border Height="215" Width="145">

<Image Source="{Binding CoverImage, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality"

Margin="0 20 0 0" VerticalAlignment="Center" HorizontalAlignment="Center"/>

</Border>

<TextBlock Text="{Binding Name}"

FontFamily="{StaticResource Inter}"

TextAlignment="Center" FontSize="18"

Margin="0 5 0 2" FontWeight="SemiBold"/>

</StackPanel>

</Border>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</ScrollViewer>

</StackPanel>

<Border Height="250" Background="#F5F5F5" CornerRadius="10" Visibility="{Binding IsBlockNotBookVisible, Converter={StaticResource BoolToVisibilityConverter}}">

<StackPanel VerticalAlignment="Center">

<Path VerticalAlignment="Center" Fill="#515355" HorizontalAlignment="Center"

Data="M10.68 11.74a6 6 0 0 1-7.922-8.982 6 6 0 0 1 8.982 7.922l3.04 3.04a.749.749 0 0 1-.326 1.275.749.749 0 0 1-.734-.215ZM11.5 7a4.499 4.499 0 1 0-8.997 0A4.499 4.499 0 0 0 11.5 7Z"

Stretch="Uniform" Width="45" Height="45" Margin="0 0 0 10"/>

<TextBlock FontSize="25" Text="Ничего не найдено..." TextWrapping="Wrap"

FontFamily="{StaticResource Inter}" FontWeight="Medium" TextAlignment="Center"/>

</StackPanel>

</Border>

<StackPanel Margin="0 10 0 0">

<DockPanel LastChildFill="True">

<TextBlock Text="Пользователи" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="20"/>

</DockPanel>

<ScrollViewer x:Name="UsersHorizontalScroll" Margin="-5 0"

VerticalScrollBarVisibility="Disabled" Visibility="{Binding IsSearchUserVisible, Converter={StaticResource BoolToVisibilityConverter}}"

HorizontalScrollBarVisibility="Hidden"

PreviewMouseWheel="HorizontalScroll\_PreviewMouseWheel">

<ScrollViewer.Resources>

<Style TargetType="ScrollViewer">

<Setter Property="PanningMode" Value="HorizontalOnly"/>

<Setter Property="UseLayoutRounding" Value="True"/>

<Setter Property="ScrollViewer.IsDeferredScrollingEnabled" Value="True"/>

</Style>

</ScrollViewer.Resources>

<ItemsControl ItemsSource="{Binding FoundUsers}"

Margin="0 10 0 0">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<StackPanel Orientation="Horizontal"/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ItemsControl.ItemTemplate>

<DataTemplate>

<Border Margin="5 0" Width="216" Height="270" Cursor="Hand"

CornerRadius="8" HorizontalAlignment="Center">

<i:Interaction.Triggers>

<i:EventTrigger EventName="MouseLeftButtonDown">

<i:InvokeCommandAction

Command="{Binding DataContext.ShowUserCommand, RelativeSource={RelativeSource AncestorType=UserControl}}"

CommandParameter="{Binding Id}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#FFFFFF"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<StackPanel HorizontalAlignment="Center" Width="200">

<Border Grid.Row="0" Grid.RowSpan="2" Height="200" Width="200" Background="#F5F5F5" CornerRadius="100" VerticalAlignment="Top">

<Border.Clip>

<EllipseGeometry Center="100,100" RadiusX="100" RadiusY="100"/>

</Border.Clip>

<Image Source="{Binding AvatarImage, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality"

HorizontalAlignment="Center"

VerticalAlignment="Center">

<Image.RenderTransform>

<TransformGroup>

<TranslateTransform X="0" Y="0"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

</Border>

<TextBlock Text="{Binding Name}" TextAlignment="Center" Margin="0 5 0 0"

FontFamily="{StaticResource Inter}" FontWeight="Medium" FontSize="20"/>

<TextBlock Text="{Binding Nickname}" TextAlignment="Center" Foreground="Gray"

FontFamily="{StaticResource Inter}" FontWeight="SemiBold" FontSize="16"/>

</StackPanel>

</Border>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</ScrollViewer>

</StackPanel>

<Border Height="250" Background="#F5F5F5" CornerRadius="10" Visibility="{Binding IsBlockNotUserVisible, Converter={StaticResource BoolToVisibilityConverter}}">

<StackPanel VerticalAlignment="Center">

<Path VerticalAlignment="Center" Fill="#515355" HorizontalAlignment="Center"

Data="M10.68 11.74a6 6 0 0 1-7.922-8.982 6 6 0 0 1 8.982 7.922l3.04 3.04a.749.749 0 0 1-.326 1.275.749.749 0 0 1-.734-.215ZM11.5 7a4.499 4.499 0 1 0-8.997 0A4.499 4.499 0 0 0 11.5 7Z"

Stretch="Uniform" Width="45" Height="45" Margin="0 0 0 10"/>

<TextBlock FontSize="25" Text="Ничего не найдено..." TextWrapping="Wrap"

FontFamily="{StaticResource Inter}" FontWeight="Medium" TextAlignment="Center"/>

</StackPanel>

</Border>

</StackPanel>

</StackPanel>

</Border>

</Grid>

</UserControl>

Пользователь может смотреть информацию о книгах.

Страница «Книга» представлена в соответствии с рисунком 17.

Изображение выглядит как текст, снимок экрана, Веб-сайт, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок – Страница «Книга»

XAML-код файла «BookPage.xaml» представлен ниже:

<UserControl x:Class="Readify.Pages.MainMenu.BookPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Readify.Pages.MainMenu"

mc:Ignorable="d"

xmlns:main="clr-namespace:Readify.ViewModels"

d:DataContext="{d:DesignInstance Type=main:BookViewModel}"

d:DesignHeight="1318" d:DesignWidth="1090">

<Grid>

<Border Margin="0 0">

<StackPanel>

<TextBlock Text="{Binding Book.Name}" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="30" Margin="0 0 0 10"/>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2.5\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Border Padding="0 0 10 0" Grid.Column="0">

<StackPanel>

<Border Height="350" Background="#F5F5F5">

<Image Margin="0 0 0 0" Source="{Binding Book.CoverImage, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality"

VerticalAlignment="Center" HorizontalAlignment="Center"/>

</Border>

<Button Content="Читать" Style="{StaticResource BookReadButton}"

Command="{Binding ReadBookCommand}"/>

<Button Content="Удалить из библиотеки" Style="{StaticResource BookButton}"

Command="{Binding DeleteFromLibraryCommand}"

Visibility="{Binding IsBtnDeleteFromLibraryVisible, Converter={StaticResource BoolToVisibilityConverter}}"/>

<Button Content="Добавить в библиотеку" Style="{StaticResource BookButton}"

Command="{Binding AddToLibraryCommand}"

Visibility="{Binding IsBtnAddToLibraryVisible, Converter={StaticResource BoolToVisibilityConverter}}"/>

<Button Content="Удалить с сервиса" Style="{StaticResource DeleteBookButton}"

Command="{Binding DeleteFromServiceCommand}"

Visibility="{Binding IsBtnDeleteFromServiceVisible, Converter={StaticResource BoolToVisibilityConverter}}"/>

</StackPanel>

</Border>

<Border Grid.Column="1" Padding="5 0">

<StackPanel>

<TextBlock Text="{Binding DeployerText}" FontFamily="{StaticResource Inter}" FontSize="23"

FontWeight="Bold"/>

<TextBlock Text="{Binding GenreText}" FontFamily="{StaticResource Inter}" FontSize="21"

FontWeight="Bold" Margin="0 5" TextWrapping="Wrap"/>

<TextBlock Text="{Binding Book.Description}"

FontSize="16" FontFamily="{StaticResource Inter}" FontWeight="Regular"

Foreground="#5A5A5A" TextWrapping="Wrap"/>

<Border Background="#5A5A5A" Height="0.1" Margin="0 20 0 10"/>

<StackPanel Visibility="{Binding IsBtnWriteReviewVisible, Converter={StaticResource BoolToVisibilityConverter}}">

<TextBox Text="{Binding AddRatingText}"

Style="{StaticResource RoundedShadowAboutTextBox}"

Tag="Напишите отзыв"/>

<StackPanel HorizontalAlignment="Left" Orientation="Horizontal">

<Path Fill="#FEC810"

Data="M8 .25a.75.75 0 0 1 .673.418l1.882 3.815 4.21.612a.75.75 0 0 1 .416 1.279l-3.046 2.97.719 4.192a.751.751 0 0 1-1.088.791L8 12.347l-3.766 1.98a.75.75 0 0 1-1.088-.79l.72-4.194L.818 6.374a.75.75 0 0 1 .416-1.28l4.21-.611L7.327.668A.75.75 0 0 1 8 .25Zm0 2.445L6.615 5.5a.75.75 0 0 1-.564.41l-3.097.45 2.24 2.184a.75.75 0 0 1 .216.664l-.528 3.084 2.769-1.456a.75.75 0 0 1 .698 0l2.77 1.456-.53-3.084a.75.75 0 0 1 .216-.664l2.24-2.183-3.096-.45a.75.75 0 0 1-.564-.41L8 2.694Z"

Stretch="Uniform" Margin="0 5 0 5" Height="25" Width="26"/>

<ComboBox ItemsSource="{Binding Ratings}" BorderThickness="0" Background="#FFEDEDED"

SelectedItem="{Binding SelectedRating}"

VerticalAlignment="Center" Width="60" Margin="10 0 0 0" FontFamily="{StaticResource Inter}" FontSize="14" FontWeight="Medium"

Tag="Выберите жанр"/>

</StackPanel>

<Button Content="Опубликовать" Style="{StaticResource BookReadButton}"

Command="{Binding AddReviewCommand}"/>

</StackPanel>

<TextBlock Text="Отзывы:" FontFamily="{StaticResource Inter}" FontSize="23"

FontWeight="Bold"/>

<ItemsControl ItemsSource="{Binding Book.Reviews}" x:Name="Followers">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<StackPanel Orientation="Vertical"/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ItemsControl.ItemTemplate>

<DataTemplate>

<StackPanel>

<StackPanel>

<Grid Margin="0 10">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="6\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="20"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Border Grid.Row="0" Grid.RowSpan="2" Height="60" Width="60" Background="#F5F5F5" CornerRadius="100" VerticalAlignment="Top">

<Border.Clip>

<EllipseGeometry Center="30,30" RadiusX="30" RadiusY="30"/>

</Border.Clip>

<Image Source="{Binding Author.AvatarImage, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality"

HorizontalAlignment="Center"

VerticalAlignment="Center">

<Image.RenderTransform>

<TransformGroup>

<TranslateTransform X="0" Y="0"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

</Border>

<DockPanel Grid.Column="1" Grid.RowSpan="2" Margin="0 12 0 0">

<TextBlock Text="{Binding Author.Name}" MaxWidth="450" FontSize="14"

FontFamily="{StaticResource Inter}" FontWeight="Bold"/>

<DockPanel HorizontalAlignment="Right">

<TextBlock Text="{Binding RatingText}" FontFamily="{StaticResource Inter}" FontWeight="Bold" FontSize="14"/>

<Path Fill="#FEC810"

Data="M8 .25a.75.75 0 0 1 .673.418l1.882 3.815 4.21.612a.75.75 0 0 1 .416 1.279l-3.046 2.97.719 4.192a.751.751 0 0 1-1.088.791L8 12.347l-3.766 1.98a.75.75 0 0 1-1.088-.79l.72-4.194L.818 6.374a.75.75 0 0 1 .416-1.28l4.21-.611L7.327.668A.75.75 0 0 1 8 .25Zm0 2.445L6.615 5.5a.75.75 0 0 1-.564.41l-3.097.45 2.24 2.184a.75.75 0 0 1 .216.664l-.528 3.084 2.769-1.456a.75.75 0 0 1 .698 0l2.77 1.456-.53-3.084a.75.75 0 0 1 .216-.664l2.24-2.183-3.096-.45a.75.75 0 0 1-.564-.41L8 2.694Z"

Stretch="Uniform" Margin="3 0 5 0" Height="22" Width="17" VerticalAlignment="Top"/>

</DockPanel>

</DockPanel>

<Border Padding="80 15 8 15" Grid.Row="1" Grid.ColumnSpan="2" CornerRadius="10" Background="#EFEFEF" Panel.ZIndex="-1" MinHeight="50">

<TextBox Style="{StaticResource ReviewTextBoxStyle}" TextWrapping="Wrap" FontFamily="{StaticResource Inter}" FontSize="13" FontWeight="Medium"

Text="{Binding Comment}"/>

</Border>

</Grid>

</StackPanel>

</StackPanel>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</StackPanel>

</Border>

<Border Grid.Column="2" Padding="10 0 0 0">

<StackPanel>

<DockPanel VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="{Binding RatingText}" FontFamily="{StaticResource Inter}"

FontSize="23" FontWeight="SemiBold" VerticalAlignment="Center" Margin="0 0 5 0"/>

<Path Fill="#FEC810"

Data="M8 .25a.75.75 0 0 1 .673.418l1.882 3.815 4.21.612a.75.75 0 0 1 .416 1.279l-3.046 2.97.719 4.192a.751.751 0 0 1-1.088.791L8 12.347l-3.766 1.98a.75.75 0 0 1-1.088-.79l.72-4.194L.818 6.374a.75.75 0 0 1 .416-1.28l4.21-.611L7.327.668A.75.75 0 0 1 8 .25Zm0 2.445L6.615 5.5a.75.75 0 0 1-.564.41l-3.097.45 2.24 2.184a.75.75 0 0 1 .216.664l-.528 3.084 2.769-1.456a.75.75 0 0 1 .698 0l2.77 1.456-.53-3.084a.75.75 0 0 1 .216-.664l2.24-2.183-3.096-.45a.75.75 0 0 1-.564-.41L8 2.694Z"

Stretch="Uniform" Margin="0 0 0 1" Height="25" Width="26"/>

</DockPanel>

<TextBlock Text="Жанр" Width="265" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="16" Margin="0 8 0 8"/>

<Border Height="50" CornerRadius="8" Padding="5 0"

Cursor="Hand">

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#F5F5F5"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<TextBlock Text="{Binding Book.Genre.Name}"

FontSize="21"

FontFamily="{StaticResource Inter}"

FontWeight="Medium"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>

</Border>

<StackPanel Visibility="{Binding IsLastBookVisible, Converter={StaticResource BoolToVisibilityConverter}}">

<TextBlock Text="Продолжить чтение" Width="265" FontFamily="{StaticResource Inter}"

FontWeight="Bold" FontSize="16" Margin="0 8 0 8"/>

<Border Margin="0 0 0 0" Height="270" Cursor="Hand"

CornerRadius="8">

<Border.Style>

<Style TargetType="Border">

<Setter Property="Background" Value="#FFFFFF"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#EFEFEF"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<StackPanel HorizontalAlignment="Center" Width="200">

<Border Height="204" Width="145">

<Image Margin="0 20 0 0" Source="{Binding LastBook.ImageBytes, Converter={StaticResource ImageConverter}}"

Stretch="UniformToFill" RenderOptions.BitmapScalingMode="HighQuality"

VerticalAlignment="Center" HorizontalAlignment="Center"/>

</Border>

<TextBlock Text="{Binding LastBook.Name}"

FontFamily="{StaticResource Inter}"

TextAlignment="Center" FontSize="18"

Margin="0 5 0 2" FontWeight="SemiBold"/>

</StackPanel>

</Border>

</StackPanel>

</StackPanel>

</Border>

</Grid>

</StackPanel>

</Border>

</Grid>

</UserControl>

1. Разработка программного функционала

Поскольку был выбран «MVVM» в качестве программного архитектурного шаблона, для каждого пользовательского элемента управления нужна соответствующая «ViewModel». Однако, чтобы избежать написания повторного кода, было решено создать базовый класс «BaseViewModel».

Этот класс содержит основную логику для обновления элементов пользовательского интерфейса, используя интерфейс «INotifyPropertyChanged» из пространства имён System.ComponentModel. Кроме того, класс содержит словарь всех значений свойств, который удобно используется для вызова метода «OnPropertyChanged» и непосредственного обновления элементов пользовательского интерфейса.

CS-код класса «BaseViewModel» представлен ниже:

using System.ComponentModel;

using System.Runtime.CompilerServices;

namespace Readify.ViewModels.Base

{

/// <summary>

/// Базовая View Model

/// </summary>

public abstract class BaseViewModel : INotifyPropertyChanged

{

/// <summary>

/// Вызывается при каждом изменении пользовательского элемента

/// </summary>

public event PropertyChangedEventHandler? PropertyChanged;

/// <summary>

/// Уведомляет об изменении свойства

/// </summary>

/// <param name="propertyName">Имя измененного свойства</param>

protected virtual void OnPropertyChanged([CallerMemberName] string? propertyName = null)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));

}

/// <summary>

/// Устанавливает значение поля и уведомляет об изменении

/// </summary>

/// <typeparam name="T">Тип поля</typeparam>

/// <param name="field">Ссылка на поле</param>

/// <param name="value">Новое значение</param>

/// <param name="propertyName">Имя свойства</param>

/// <returns>True если значение изменилось</returns>

protected bool SetField<T>(ref T field, T value, [CallerMemberName] string? propertyName = null)

{

if(EqualityComparer<T>.Default.Equals(field,value)) return false;

field=value;

OnPropertyChanged(propertyName);

return true;

}

}

}

Класс «LoginViewModel» является одной из реализаций базового класса «BaseViewModel».

CS-код класса «LoginViewModel» представлен ниже:

using CommunityToolkit.Mvvm.Input;

using Readify.DTO.Users;

using Readify.Pages;

using Readify.Pages.Registartion;

using Readify.Services.Base;

using Readify.ViewModels.Base;

using System.Net.Http;

using System.Windows;

using System.Windows.Input;

namespace Readify.ViewModels

{

/// <summary>

/// ViewModel для LoginPage

/// </summary>

public class LoginViewModel : BaseViewModel

{

private string \_username = string.Empty;

private string \_password = string.Empty;

private string \_errorMessage = string.Empty;

private bool \_isErrorVisible;

/// <summary>

/// Сервис авторизации

/// </summary>

private readonly IAuthService \_authService;

/// <summary>

/// Имя пользователя

/// </summary>

public string Username

{

get => \_username;

set => SetField(ref \_username, value);

}

/// <summary>

/// Пароль

/// </summary>

public string Password

{

get => \_password;

set => SetField(ref \_password, value);

}

/// <summary>

/// Сообщение об ошибке

/// </summary>

public string ErrorMessage

{

get => \_errorMessage;

set => SetField(ref \_errorMessage, value);

}

/// <summary>

/// Видимость сообщения об ошибке

/// </summary>

public bool IsErrorVisible

{

get => \_isErrorVisible;

set => SetField(ref \_isErrorVisible, value);

}

/// <summary>

/// Команда обработки нажатия на кнопку "Войти"

/// </summary>

public ICommand LoginCommand { get; }

/// <summary>

/// Команда обработки нажатия на кнопку "Создайте аккаунт"

/// </summary>

public ICommand GoToRegisterCommand { get; }

/// <summary>

/// Команда обработки нажатия на крестик у блока ошибки

/// </summary>

public ICommand HiddenErrorCommand { get; }

/// <summary>

/// Конструктор

/// </summary>

public LoginViewModel(IAuthService authService)

{

\_authService = authService;

LoginCommand = new RelayCommand(ExecuteLoginAsync);

GoToRegisterCommand = new RelayCommand(ExecuteGoToRegister);

HiddenErrorCommand = new RelayCommand(() => IsErrorVisible = false);

}

/// <summary>

/// Конструктор с выводом ошибки

/// </summary>

public LoginViewModel(IAuthService authService, string message)

{

ShowError(message);

\_authService = authService;

LoginCommand = new RelayCommand(ExecuteLoginAsync);

GoToRegisterCommand = new RelayCommand(ExecuteGoToRegister);

HiddenErrorCommand = new RelayCommand(() => IsErrorVisible = false);

}

/// <summary>

/// Асинхронный метод входа в аккаунт

/// </summary>

private async void ExecuteLoginAsync()

{

if (string.IsNullOrWhiteSpace(Username) || string.IsNullOrEmpty(Password))

{

ShowError("Заполните все поля");

return;

}

else

{

CloseError();

try

{

if (await \_authService.LoginAsync(new LoginDTO

{

EmailOrNickname = Username,

Password = Password,

Device = Settings.Default.Device

}))

{

MainMenuPage mainMenuPage = new MainMenuPage();

App.MainMenuPage = mainMenuPage;

var mainWindow = Application.Current.MainWindow as MainWindow;

mainWindow?.MainFrame.Navigate(mainMenuPage);

}

else

{

ShowError("Неверное имя пользователя или пароль");

}

}

catch (HttpRequestException)

{

ShowError($"Отсутствует подключение к серверу");

}

catch (Exception ex)

{

ShowError($"Error: {ex}");

}

}

}

/// <summary>

/// Метод перехода на страницу <see cref="RegistrationMainPage"/>

/// </summary>

private void ExecuteGoToRegister()

{

var mainWindow = Application.Current.MainWindow as MainWindow;

mainWindow?.MainFrame.Navigate(new RegistrationMainPage());

}

/// <summary>

/// Метод для вывода блока ошибки с определенным текстом

/// </summary>

/// <param name="message">Текст ошибки</param>

private void ShowError(string message)

{

ErrorMessage = message;

IsErrorVisible = true;

}

/// <summary>

/// Метод закрытия блока ошибки

/// </summary>

private void CloseError()

{

IsErrorVisible = false;

}

}

}

Поскольку приложению необходимо постоянно обращаться к серверу, было принято решение использовать паттерн «Service», в котором бы содержалась бизнес-логика.

Для того, чтобы управлять зависимостями между компонентами приложения было принято решение использовать паттерн «Dependency Injection». Для его реализации данного паттерна требовалось создать интерфейсы сервисов.

Например, интерфейс «IUserService» описывает методы класса «UserService».

CS-код интерфейса «IUserService» представлен ниже:

using Readify.DTO;

using Readify.DTO.Subscribe;

using Readify.DTO.Users;

namespace Readify.Services.Base

{

/// <summary>

/// Интерфейс для работы с пользователями

/// </summary>

public interface IUserService

{

/// <summary>

/// Метод получения информации о пользователе по его Id

/// </summary>

/// <param name="id">Id пользователя</param>

/// <returns>Информация о пользователе</returns>

Task<UserDTO> GetUserByIdAsync(int id);

/// <summary>

/// Метод подписки на пользователя

/// </summary>

/// <param name="subscribeDTO">Данные подписки</param>

/// <returns>Информация об авторе</returns>

Task<UserDTO> FollowToUserAsync(SubscribeDTO subscribeDTO);

/// <summary>

/// Метод отписки от пользователя

/// </summary>

/// <param name="subscribeDTO">Данные подписки</param>

/// <returns>Информация об авторе</returns>

Task<UserDTO> UnfollowForUserAsync(SubscribeDTO subscribeDTO);

/// <summary>

/// Метод поиска

/// </summary>

/// <param name="searchText">Поисковые данные</param>

/// <returns>Найденные данные</returns>

Task<SearchDTO> SearchAsync(string searchText);

}

}

Для выполнения запросов на сервер используются HTTP-запросы. Таким образом, благодаря использованию паттерна «Service», приложение получает доступ к необходимой функциональности и становится более гибким и расширяемым.

Класс «UserService» является одним из сервисов.

CS-код класса «UserService»

using Readify.DTO;

using Readify.DTO.Subscribe;

using Readify.DTO.Users;

using Readify.Services.Base;

using Readify.Services.Connections;

using Readify.Services.Exceptions;

using System.Net;

using System.Net.Http;

using System.Net.Http.Json;

namespace Readify.Services

{

/// <summary>

/// Сервис для работы с пользователем

/// </summary>

public class UserService : IUserService

{

private const string GET\_USER\_ENDPOINT = $"/api/user/";

private const string FOLLOW\_USER\_ENDPOINT = $"/api/user/subscribe";

private const string UNFOLLOW\_USER\_ENDPOINT = $"/api/user/unsubscribe";

private const string SEARCH\_ENDPOINT = $"/api/user/search";

/// <summary>

/// ApiClient для работы с API

/// </summary>

private readonly ApiClient \_apiClient;

/// <summary>

/// Конструктор

/// </summary>

public UserService()

{

\_apiClient = new ApiClient();

}

/// <summary>

/// Метод получения информации о пользователе по его Id

/// </summary>

/// <param name="id">Id пользователя</param>

/// <returns>Информация о пользователе</returns>

public async Task<UserDTO> GetUserByIdAsync(int id)

{

var response = await \_apiClient.SendRequestAsync(HttpMethod.Get, GET\_USER\_ENDPOINT + id, new

{

Id = id

}, true);

if (response.StatusCode == HttpStatusCode.Unauthorized)

{

throw new UnauthoizeException("Сессия закончена");

}

if (response.IsSuccessStatusCode)

{

var content = await response.Content.ReadFromJsonAsync<UserDTO>();

return content!;

}

throw new Exception("Ответ сервера: " + response.StatusCode.ToString());

}

/// <summary>

/// Метод подписки на пользователя

/// </summary>

/// <param name="subscribeDTO">Данные подписки</param>

/// <returns>Информация об авторе (на кого подписываются)</returns>

public async Task<UserDTO> FollowToUserAsync(SubscribeDTO subscribeDTO)

{

var response = await \_apiClient.SendRequestAsync(

HttpMethod.Post,

FOLLOW\_USER\_ENDPOINT,

subscribeDTO, true);

if (response.StatusCode == HttpStatusCode.Unauthorized)

{

throw new UnauthoizeException("Сессия закончена");

}

if (response.IsSuccessStatusCode)

{

var content = await response.Content.ReadFromJsonAsync<UserDTO>();

return content!;

}

throw new Exception("Ответ сервера: " + response.StatusCode.ToString());

}

/// <summary>

/// Метод отписки от пользователя

/// </summary>

/// <param name="subscribeDTO">Данные подписки</param>

/// <returns>Информация об авторе (на кого подписываются)</returns>

public async Task<UserDTO> UnfollowForUserAsync(SubscribeDTO subscribeDTO)

{

var response = await \_apiClient.SendRequestAsync(

HttpMethod.Post,

UNFOLLOW\_USER\_ENDPOINT,

subscribeDTO, true);

if (response.StatusCode == HttpStatusCode.Unauthorized)

{

throw new UnauthoizeException("Сессия закончена");

}

if (response.IsSuccessStatusCode)

{

var content = await response.Content.ReadFromJsonAsync<UserDTO>();

return content!;

}

throw new Exception("Ответ сервера: " + response.StatusCode.ToString());

}

/// <summary>

/// Метод поиска

/// </summary>

/// <param name="searchText">Поисковые данные</param>

/// <returns>Найденные данные</returns>

public async Task<SearchDTO> SearchAsync(string searchText)

{

var response = await \_apiClient.SendRequestAsync(

HttpMethod.Get,

SEARCH\_ENDPOINT,

new

{

SearchText = searchText

}, true);

if (response.StatusCode == HttpStatusCode.Unauthorized)

{

throw new UnauthoizeException("Сессия закончена");

}

if (response.IsSuccessStatusCode)

{

var content = await response.Content.ReadFromJsonAsync<SearchDTO>();

return content!;

}

throw new Exception("Ответ сервера: " + response.StatusCode.ToString());

}

}

}

Чтобы не указывать адрес сервера напрямую в коде был создан файл «Settings.settings» с соответствующим полем.

Файл «Settings.settings» представлен в соответствии с рисунком 18.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок – Файл «Settings.settings»

Для того, чтобы при каждом входе не вводить пароль в классе «App.xaml.cs» была реализована логика проверки access-токена, так же в этом классе реализована логика паттерна «Dependency Injection».

CS-код класса «App.xaml.cs» представлена ниже:

using Microsoft.Extensions.DependencyInjection;

using Readify.DTO.Library;

using Readify.DTO.Users;

using Readify.Pages;

using Readify.Pages.MainMenu;

using Readify.Services;

using Readify.Services.Base;

using Readify.ViewModels;

using System.Net.Http;

using System.Windows;

namespace Readify

{

/// <summary>

/// Логика для App.xaml

/// </summary>

public partial class App : Application

{

/// <summary>

/// Сервис аутентификации

/// </summary>

private AuthService \_authService = new AuthService();

/// <summary>

/// Сервайс провайдер для реализации DI

/// </summary>

public static IServiceProvider ServiceProvider { get; private set; } = null!;

/// <summary>

/// Текущий авторизованный пользователь

/// </summary>

public static UserDTO CurrentUser { get; set; } = null!;

/// <summary>

/// Библиотека пользователя

/// </summary>

public static SeeLibrariesDTO CurrentUserLibrary { get; set; } = null!;

/// <summary>

/// Главная страница

/// </summary>

public static MainMenuPage MainMenuPage { get; set; } = null!;

public static MainMenuViewModel DataContextMainMenu

{

get => MainMenuPage.DataContext as MainMenuViewModel;

}

/// <summary>

/// Текущая страница

/// </summary>

public static object InitProfilePage { get; set; } = null!;

public static bool IsMainOnProfile

{

get

{

return MainMenuPage.MainMenuFrame.Content is ProfilePage;

}

}

/// <summary>

/// Текущая страница профиля

/// </summary>

public static ProfilePage ProfilePage

{

get

{

return MainMenuPage.MainMenuFrame.Content as ProfilePage;

}

}

/// <summary>

/// Метод, который вызывавется при запуске приложения (проверка refresh-токена)

/// </summary>

/// <param name="e"></param>

protected override async void OnStartup(StartupEventArgs e)

{

var serviceCollection = new ServiceCollection();

ConfigureServices(serviceCollection);

ServiceProvider = serviceCollection.BuildServiceProvider();

base.OnStartup(e);

var mainWindow = new MainWindow();

if (!string.IsNullOrEmpty(Settings.Default.RefreshToken))

{

try

{

if (await \_authService.RefreshTokenAsync())

{

mainWindow.NavigateToMenu();

}

else

{

mainWindow.NavigateToLogin();

}

}

catch (HttpRequestException)

{

mainWindow.NavigateToLoginError("Отсутствует соединение с сервером");

}

}

else

{

mainWindow.NavigateToLogin();

}

mainWindow.Show();

}

private void ConfigureServices(IServiceCollection services)

{

services.AddSingleton<IAuthService, AuthService>();

services.AddSingleton<IUserService, UserService>();

services.AddSingleton<IRegistrationService, RegistrationService>();

services.AddSingleton<ILibraryService, LibraryService>();

services.AddSingleton<IBookService, BookService>();

}

}

}

Для того, чтобы передавать только необходимые данные между сервером и приложением было принято решение использовать паттерн «Data Transfer Object».

Примером реализации данного паттерна является класс «UserDTO».

CS-код класса «UserDTO» представлен ниже:

using Readify.DTO.Library;

using Readify.DTO.Reviews;

namespace Readify.DTO.Users

{

public class UserDTO

{

/// <summary>Id пользователя</summary>

public int Id { get; set; }

/// <summary>Логин пользователя</summary>

public string? Nickname { get; set; }

/// <summary>Описание пользователя</summary>

public string? Description { get; set; }

/// <summary>Фотография профиля пользователя</summary>

public byte[]? AvatarImage { get; set; } = null!;

/// <summary>Имя пользователя</summary>

public string? Name { get; set; } = null!;

/// <summary>Email пользователя</summary>

public string? Email { get; set; } = null!;

/// <summary>Роль пользователя</summary>

public int? IdRole { get; set; }

/// <summary>Дата создания аккаунта</summary>

public DateTime? CreatedAt { get; set; }

/// <summary>Список книг пользователя</summary>

public List<LibraryDTO>? Books { get; set; }

/// <summary>Список подписчиков пользователя</summary>

public List<AuthorDTO>? Subscribers { get; set; }

/// <summary>Список подписок пользователя</summary>

public List<AuthorDTO>? Subscriptions { get; set; }

/// <summary>Список отзывов пользователя</summary>

public List<SeeReviewDTO>? Reviews { get; set; }

}

}

С помощью хорошо разработанной архитектуры не составило труда написать логику для остальных слоев системы.

1. Разработка сервера

Для разработки информационной системы было решено создать сервер, который будет взаимодействовать с базой данных и обрабатывать запросы от клиентов.

Для удобства работы с базой данных и избежание дублирования кода было принято решение использовать ORM (Object-Relational Mapping) «Entity Framework».

Данный фреймворк создает сущности и контекст базы данных для работы с ней без необходимости писать SQL-запросы.

Для удобства работы с БД было решено использовать паттерн «Controller-Service-Repository» который разделяет приложение на три ключевых слоя с определёнными обязанностями:

1. Слой контроллера. Обрабатывает входящие HTTP-запросы, проверяет входные данные и возвращает соответствующие ответы. Этот слой выступает точкой входа в приложение и направляет трафик к соответствующему сервису.
2. Слой сервиса. Содержит бизнес-логику приложения. Сервисы обрабатывают данные от контроллеров, выполняют необходимые операции и координируют работу с репозиториями для получения или сохранения данных.
3. Слой репозитория. Управляет доступом к данным и их сохранением. Репозитории абстрагируют взаимодействие с базами данных или другими источниками данных, инкапсулируя логику для запроса, сохранения и обновления данных. [10]

Для управления зависимостями был так же использован паттерн «Dependency Injection».

Примером репозитория является класс «UserRepository»

CS-код класса «UserRepositroy» представлен ниже:

using HelpLibrary.DTOs.Users;

using HelpLibrary.Entities;

using Microsoft.AspNetCore.Identity;

using Microsoft.EntityFrameworkCore;

using ServerLibrary.Data;

using ServerLibrary.Repositories.Interfaces.IUser;

namespace ServerLibrary.Repositories.Implementations.UserI

{

public class UserRepository : IUserRepository

{

private readonly ReadifyContext \_context;

public UserRepository(ReadifyContext context)

{

\_context = context;

}

public async Task<User> AddToDatabaseAsync(User user)

{

var result = await \_context.AddAsync(user!);

await \_context.SaveChangesAsync();

return result.Entity;

}

public async Task<User?> FindByEmailAsync(string email) =>

await \_context.Users

.Include(u => u.Books)

.ThenInclude(b => b.IdGenreNavigation)

.Include(u => u.Libraries)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(b => b.IdGenreNavigation)

.Include(u => u.Libraries)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(u => u.IdAuthorNavigation)

.Include(u => u.UserSubscriberIdAuthorNavigations)

.ThenInclude(us => us.IdSubscriberNavigation)

.Include(u => u.UserSubscriberIdSubscriberNavigations)

.ThenInclude(us => us.IdAuthorNavigation)

.Include(u => u.BookReviews)

.ThenInclude(r => r.IdBookNavigation)

.ThenInclude(b => b.IdAuthorNavigation)

.Include(u => u.BookReviews)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(b => b.IdGenreNavigation)

.AsSplitQuery()

.FirstOrDefaultAsync(u => u.Email == email);

public async Task<User?> FindByIdAsync(int id) =>

await \_context.Users

.Include(u => u.Books)

.ThenInclude(b => b.IdGenreNavigation)

.Include(u => u.Libraries)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(b => b.IdGenreNavigation)

.Include(u => u.Libraries)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(u => u.IdAuthorNavigation)

.Include(u => u.UserSubscriberIdAuthorNavigations)

.ThenInclude(us => us.IdSubscriberNavigation)

.Include(u => u.UserSubscriberIdSubscriberNavigations)

.ThenInclude(us => us.IdAuthorNavigation)

.Include(u => u.BookReviews)

.ThenInclude(r => r.IdBookNavigation)

.ThenInclude(b => b.IdAuthorNavigation)

.Include(u => u.BookReviews)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(b => b.IdGenreNavigation)

.AsSplitQuery()

.FirstOrDefaultAsync(u => u.Id == id);

public async Task<User?> FindByNicknameAsync(string nickname) =>

await \_context.Users

.Include(u => u.Books)

.ThenInclude(b => b.IdGenreNavigation)

.Include(u => u.Libraries)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(b => b.IdGenreNavigation)

.Include(u => u.Libraries)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(u => u.IdAuthorNavigation)

.Include(u => u.UserSubscriberIdAuthorNavigations)

.ThenInclude(us => us.IdSubscriberNavigation)

.Include(u => u.UserSubscriberIdSubscriberNavigations)

.ThenInclude(us => us.IdAuthorNavigation)

.Include(u => u.BookReviews)

.ThenInclude(r => r.IdBookNavigation)

.ThenInclude(b => b.IdAuthorNavigation)

.Include(u => u.BookReviews)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(b => b.IdGenreNavigation)

.AsSplitQuery()

.FirstOrDefaultAsync(u => u.Nickname == nickname);

public async Task RemoveFromDatabaseAsync(User user)

{

var subs = await GetAllSub(user.Id);

\_context.UserSubscribers.RemoveRange(subs);

var book = await GetAllBooks(user.Id);

\_context.Books.RemoveRange(book);

\_context.Users.Remove(user);

await \_context.SaveChangesAsync();

}

private async Task<List<Book>> GetAllBooks(int idUser) =>

await \_context.Books.Where(b => b.IdAuthor == idUser).ToListAsync();

private async Task<List<UserSubscriber>> GetAllSub(int idUser) =>

await \_context.UserSubscribers.Where(us => us.IdSubscriber == idUser || us.IdAuthor == idUser).ToListAsync();

public async Task<User> UpdateAsync(UpdateUserDTO updateUser)

{

var findUser = await FindByIdAsync(updateUser.UserId);

if (updateUser.Nickname is not null) findUser!.Nickname = updateUser.Nickname!;

findUser!.Description = updateUser.Description;

if (updateUser.Name is not null) findUser!.Name = updateUser.Name!;

await \_context.SaveChangesAsync();

return findUser!;

}

public async Task<List<User>>? FindAllByNicknameAsync(string name) =>

await \_context.Users

.Include(u => u.Books)

.ThenInclude(b => b.IdGenreNavigation)

.Include(u => u.Libraries)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(b => b.IdGenreNavigation)

.Include(u => u.Libraries)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(u => u.IdAuthorNavigation)

.Include(u => u.UserSubscriberIdAuthorNavigations)

.ThenInclude(us => us.IdSubscriberNavigation)

.Include(u => u.UserSubscriberIdSubscriberNavigations)

.ThenInclude(us => us.IdAuthorNavigation)

.Include(u => u.BookReviews)

.ThenInclude(r => r.IdBookNavigation)

.ThenInclude(b => b.IdAuthorNavigation)

.Include(u => u.BookReviews)

.ThenInclude(l => l.IdBookNavigation)

.ThenInclude(b => b.IdGenreNavigation)

.AsSplitQuery()

.Where(u => EF.Functions.Like(u.Name, $"%{name.ToLower()}%")).ToListAsync();

public async Task SaveChangesAsync()

{

await \_context.SaveChangesAsync();

}

}

}

Примером сервиса является класс «UserService».

CS-код класса «UserService» представлен ниже:

using HelpLibrary.DTOs.Users;

using HelpLibrary.DTOs;

using HelpLibrary.Entities;

using HelpLibrary.Responces;

using Microsoft.AspNetCore.Http;

using ServerLibrary.Helpers;

using ServerLibrary.Repositories.Interfaces;

using ServerLibrary.Services.Interfaces;

using ServerLibrary.Helpers.Exceptions.User;

using ServerLibrary.Helpers.Exceptions;

using Microsoft.EntityFrameworkCore;

using HelpLibrary.DTOs.Subscribe;

using ServerLibrary.Repositories.Interfaces.IUser;

using ServerLibrary.Helpers.Converters;

using ServerLibrary.Repositories.Interfaces.Books;

namespace ServerLibrary.Services.Implementations

{

public class UserService : IUserService

{

private readonly IUserRepository \_userRepository;

private readonly ILogRepository \_logRepository;

private readonly ISubscribeRepository \_subscribeRepository;

private readonly IBookRepository \_bookRepository;

public UserService(IUserRepository userRepository, ILogRepository logRepository,

ISubscribeRepository subscribeRepository, IBookRepository bookRepository)

{

\_userRepository = userRepository;

\_logRepository = logRepository;

\_subscribeRepository = subscribeRepository;

\_bookRepository = bookRepository;

}

public async Task<List<UserInfoDTO>> GetAllSubscribersAsync(int idAuthor) =>

await \_subscribeRepository.GetSubscribersByIdAsync(idAuthor);

public async Task<List<UserInfoDTO>> GetAllSubscriptionsAsync(int idUser) =>

await \_subscribeRepository.GetSubscriptionsByIdAsync(idUser);

public async Task<UserDTO> GetUserDTO(int id)

{

var user = await \_userRepository.FindByIdAsync(id);

return await ConvertToUserDTO.Convert(user);

}

public async Task<UserDTO> SubscribeAsync(SubscribeDTO subscribe)

{

if(subscribe is null) throw new NullReferenceException("Model is empty");

if(subscribe.SubscriberId == subscribe.AuthorId) throw new Exception();

var sub = await \_subscribeRepository.GetSubByIdAsync(subscribe);

if(sub is not null) throw new AlreadySubscribedExceprion("You have already subscribed to this user");

await \_subscribeRepository.SubscribeAsync(subscribe);

await \_logRepository.WriteLogsAsync(new LogsDTO{IdUser=subscribe.SubscriberId,Action=Constants.Subscribe+subscribe.AuthorId});

return await GetUserDTO(subscribe.AuthorId);

}

public async Task<UserDTO> UnsubscribeAsync(SubscribeDTO unsubscribe)

{

if(unsubscribe is null) throw new NullReferenceException("Model is empty");

var sub = await \_subscribeRepository.GetSubByIdAsync(unsubscribe);

if(sub is null) throw new NotFoundException("Don't found");

Console.WriteLine(sub.Id);

await \_subscribeRepository.UnsubscribeAsync(unsubscribe);

await \_logRepository.WriteLogsAsync(new LogsDTO{IdUser=unsubscribe.SubscriberId,Action=Constants.UnSubscribe+unsubscribe.AuthorId});

return await GetUserDTO(unsubscribe.AuthorId);

}

public async Task<UserDTO> UpdateUserAsync(UpdateUserDTO user)

{

var updateUser = await \_userRepository.FindByIdAsync(user.UserId);

if(updateUser is null) throw new NotFoundUserException("Not found user");

if(user.AvatarImage is not null)

{

if(File.Exists(updateUser.AvatarImagePath)&&

(updateUser.AvatarImagePath!=Constants.DefaultAvatar))

File.Delete(updateUser.AvatarImagePath);

await DownloadFile(user.AvatarImage,updateUser);

}

await \_userRepository.UpdateAsync(user);

await \_logRepository.WriteLogsAsync(new LogsDTO{IdUser=user.UserId,Action=Constants.Update});

return await ConvertToUserDTO.Convert(updateUser);

}

public async Task<GeneralResponce> RemoveUserAsync(int id)

{

var user = await \_userRepository.FindByIdAsync(id);

if(user is null) throw new NotFoundUserException("User not found");

await \_userRepository.RemoveFromDatabaseAsync(user);

return new GeneralResponce("Success");

}

private async Task<string> DownloadFile(IFormFile file,User user)

{

try

{

string fileName=$"user-id{user.Id}.png";

var filePath=Path.Combine(Constants.PathToUserAvatar!,fileName);

using(var stream=new FileStream(filePath,FileMode.Create))

{

await file.CopyToAsync(stream);

user.AvatarImagePath=Constants.PathUserImages+fileName;

return filePath;

}

}

catch(Exception ex)

{

throw new Exception($"Error while downloading file: {ex.Message}",ex);

}

}

public async Task<SearchDTO> SearchAsync(string searchText)

{

if(searchText is null) throw new Exception("Model is empty");

var foundUsers=await \_userRepository.FindAllByNicknameAsync(searchText)!;

var foundBooks=await \_bookRepository.FindAllBooksByNameAsync(searchText);

return new SearchDTO

{

SearchText=searchText,

FoundBooks=(await Task.WhenAll(foundBooks.Select(ConvertToSeeBookDTO.Convert))).ToList(),

FoundUsers=(await Task.WhenAll(foundUsers.Select(ConvertToAuthorDTO.Convert))).ToList()

};

}

}

}

Примером контроллера является класс «UserController».

CS-код класса «UserController» представлен ниже:

using HelpLibrary.DTOs.Subscribe;

using HelpLibrary.DTOs.Users;

using HelpLibrary.Entities;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.SignalR;

using ServerLibrary.Services.Interfaces;

namespace Server.Controllers

{

[Route("api/[controller]")]

[Authorize]

[ApiController]

public class UserController : Controller

{

private readonly IUserService \_userService;

public UserController(IUserService userService)

{

\_userService = userService;

}

[HttpPut("/api/user/update")]

public async Task<IActionResult> UpdateAsync([FromForm] UpdateUserDTO user)

{

if(user==null) return BadRequest("Model is empty");

var userIdClaim=User.Claims.FirstOrDefault(c=>c.Type=="id");

if(userIdClaim==null||!int.TryParse(userIdClaim.Value,out int userId))

{

return Unauthorized("Не удалось определить ID пользователя.");

}

if(user.UserId!=userId)

return Forbid("Ошибка");

try

{

var result=await \_userService.UpdateUserAsync(user);

return Ok(result);

}

catch(Exception ex)

{

return BadRequest(ex.Message);

}

}

[HttpGet("/api/user/{id}")]

public async Task<IActionResult> GetUser(int id)

{

var userDTO=await \_userService.GetUserDTO(id);

return Ok(userDTO);

}

[HttpGet("/api/user/search")]

public async Task<IActionResult> Search([FromForm] string searchText)

{

var searchDTO=await \_userService.SearchAsync(searchText);

return Ok(searchDTO);

}

[HttpPost("/api/user/subscribe")]

public async Task<IActionResult> Subscribe([FromForm] SubscribeDTO sub)

{

if(sub==null)

return BadRequest("Model is empty");

var userIdClaim=User.Claims.FirstOrDefault(c=>c.Type=="id");

if(userIdClaim==null||!int.TryParse(userIdClaim.Value,out int userId))

{

return Unauthorized("Не удалось определить ID пользователя.");

}

if(sub.SubscriberId!=userId)

return Forbid("Ошибка");

if(sub.SubscriberId==sub.AuthorId)

return BadRequest("Ошибка! Нельзя подписываться на себя.");

try

{

var result=await \_userService.SubscribeAsync(sub);

return Ok(result);

}

catch(Exception ex)

{

return BadRequest(ex.Message);

}

}

[HttpPost("/api/user/unsubscribe")]

public async Task<IActionResult> UnSubscribe([FromForm] SubscribeDTO unsub)

{

if(unsub==null)

return BadRequest("Model is empty");

var userIdClaim=User.Claims.FirstOrDefault(c=>c.Type=="id");

if(userIdClaim==null||!int.TryParse(userIdClaim.Value,out int userId))

{

return Unauthorized("Не удалось определить ID пользователя.");

}

if(unsub.SubscriberId!=userId)

return Forbid("Ошибка");

try

{

var result=await \_userService.UnsubscribeAsync(unsub);

return Ok(result);

}

catch(Exception ex)

{

return BadRequest(ex.Message);

}

}

[HttpGet("/api/user/subscribers")]

public async Task<IActionResult> GetSubscribers(int id)

{

try

{

var result=await \_userService.GetAllSubscribersAsync(id);

return Ok(result);

}

catch(Exception ex)

{

return BadRequest(ex.Message);

}

}

[HttpGet("/api/user/subscriptions")]

public async Task<IActionResult> GetSubscriptions(int id)

{

try

{

var result=await \_userService.GetAllSubscriptionsAsync(id);

return Ok(result);

}

catch(Exception ex)

{

return BadRequest(ex.Message);

}

}

}

}

С помощью хорошо разработанной архитектуры не составило труда написать логику для остальных слоев сервера.

1. ТЕСТИРОВАНИЕ

После создания приложения необходимо его протестировать. Тестирование проведено на устройстве с операционной системой Windows 11.

* 1. Тестирование логики программы

Для проверки корректности работы алгоритмов программы было проведено тестирование, результаты которого представлены в приложении 2. В процессе тестирования было испытано множество вариантов использования программы. В итоге было установлено, что поведение программы является стабильным и надежным. Критические ошибки, приводящие к аварийному завершению работы программы, не выявлены.

* 1. Тестирование интерфейса пользователя

Тестирование интерфейса программы представлено в приложении 3. В ходе тестирования пользовательского интерфейса не было выявлено некорректного поведения приложения. Внешний вид графических элементов приложения не противоречит требованиям системы. Ошибок в логике поведения программы также нет, что говорит об успешной разработке приложения и готовности к её использованию.

1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ
2. Введение

Модуль информационной системы был разработан с целью упрощения поиска и чтения книг. Для работы с системой пользователи могут зарегистрироваться и авторизироваться.

Пользователи данного модуля информационной системы должны обладать базовыми навыками работы с компьютером.

1. Назначение и условия применения программного продукта

Данный модуль информационной системы предназначен для коммуникации между сотрудниками компании.

Для работы с программой требуется ЭВМ с установленной операционной системой Windows 10 или 11, а также с доступом к серверу базы данных информационной системы.

1. Подготовка к работе

Для обеспечения работоспособности информационной системы необходимо выполнить следующие шаги. Необходимо выполнить скрипт базы данных, который находится в файле «script.sql», для создания базы данных информационной системы.

Также нужно изменить IP-адрес в файле по пути Readify\Settings.settings для подключения к серверу, если это необходимо.

После того, как была проделана вся необходимая настройка программного обеспечени, необходимо запустить сервер с помощью файла «StartServer.bat», потом нужно дважды кликнуть по исполняемому файлу «Readify.exe» для непосредственной работы с информационной системой.

Обратите внимание, что по умолчанию сервер и клиент работают с портом 6969.

1. Описание операций

В первую очередь нужно зарегистрироваться в информационной системе. Регистрация проходит в 4 этапа.

В первом этапе необходимо ввести почту, пароль, повторить пароль, логии и нажать на кнопку «Продолжить». Окно первого этапа регистрации представлено в соответствии с рисунком 19.

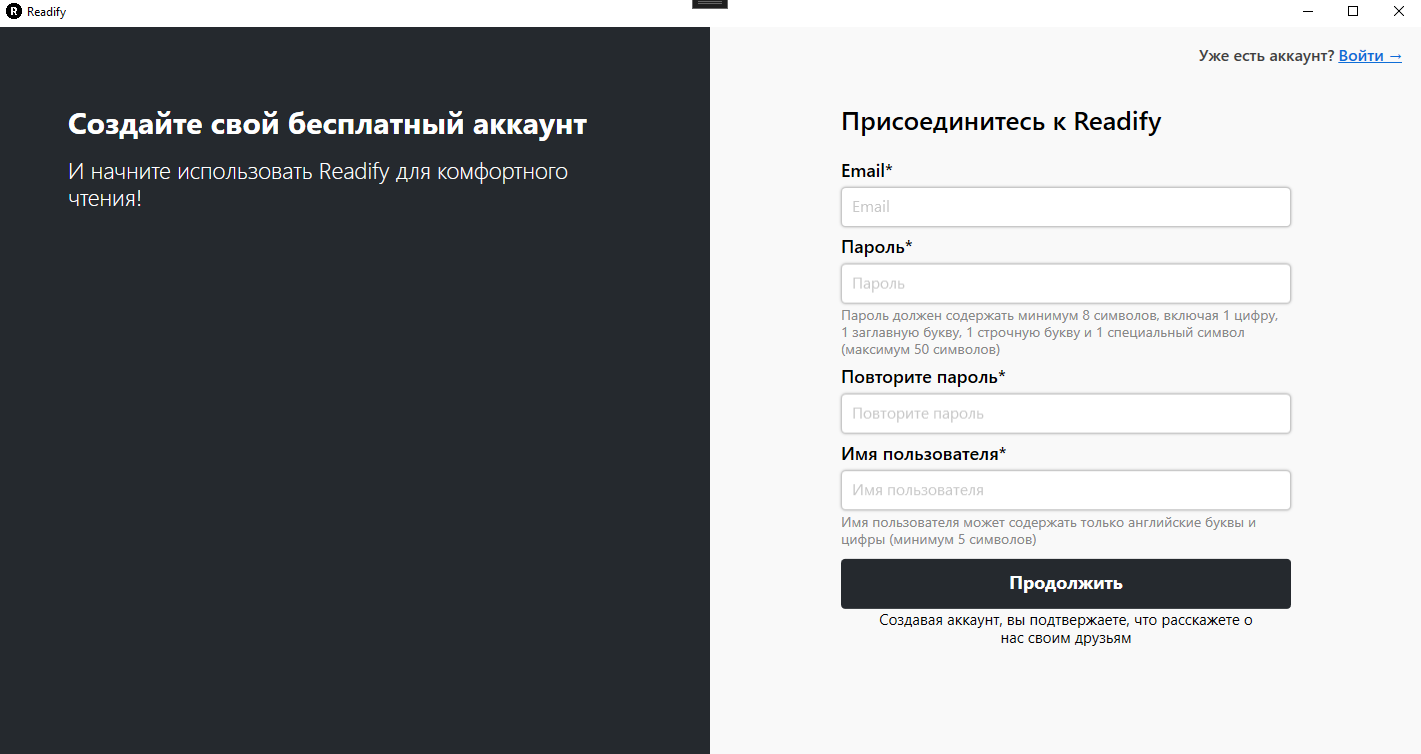


Рисунок – Окно перого этапа регистрации

В следующем окне необходимо ввести капчу. Окно второго этапа регистрации представлено в соответствии с рисунком 20.

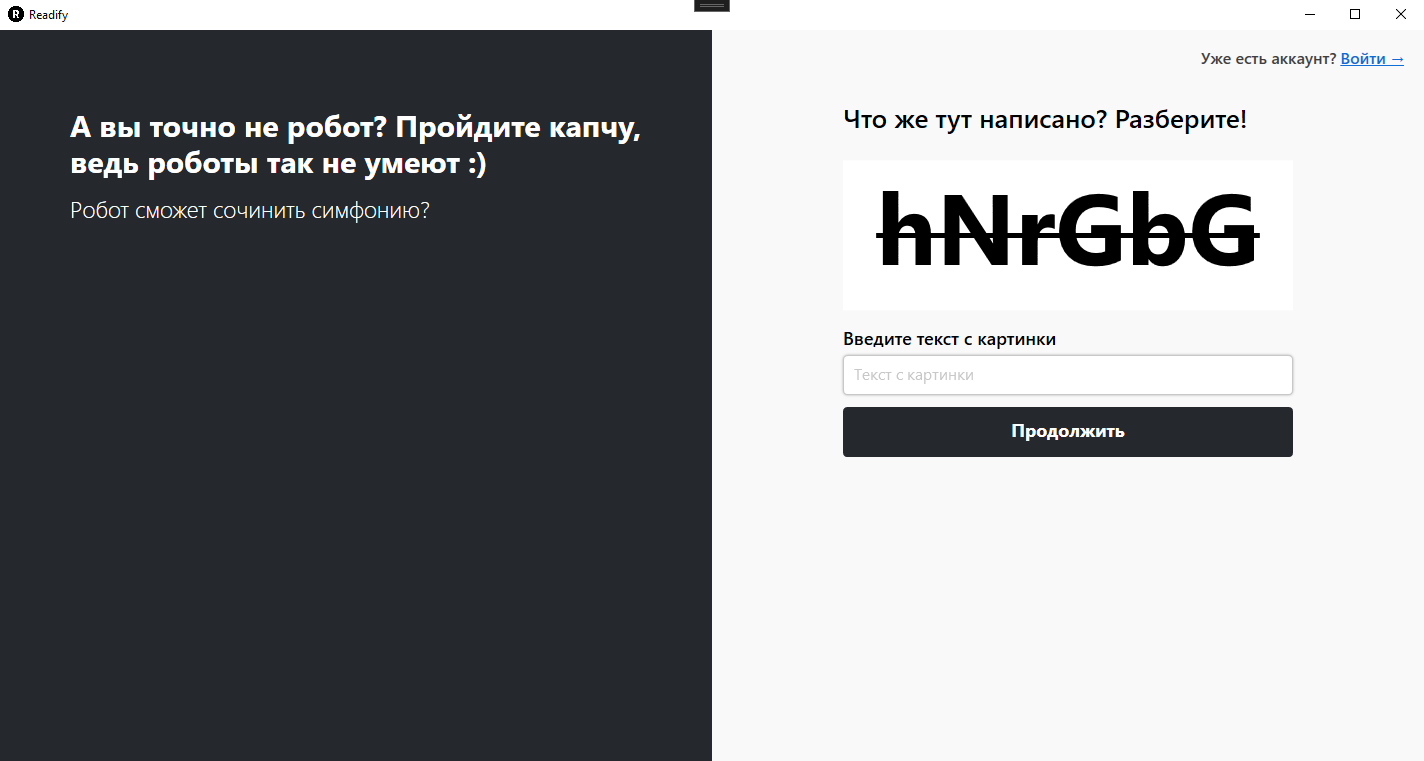


Рисунок – Окно второго этапа регистрации

Далее необходимо подтвердить почту, введя код, который пришел в письме на указанный в первом этапе адрес почты. Окно третьего этапа регистрации представлено в соответствии с рисунком 21.

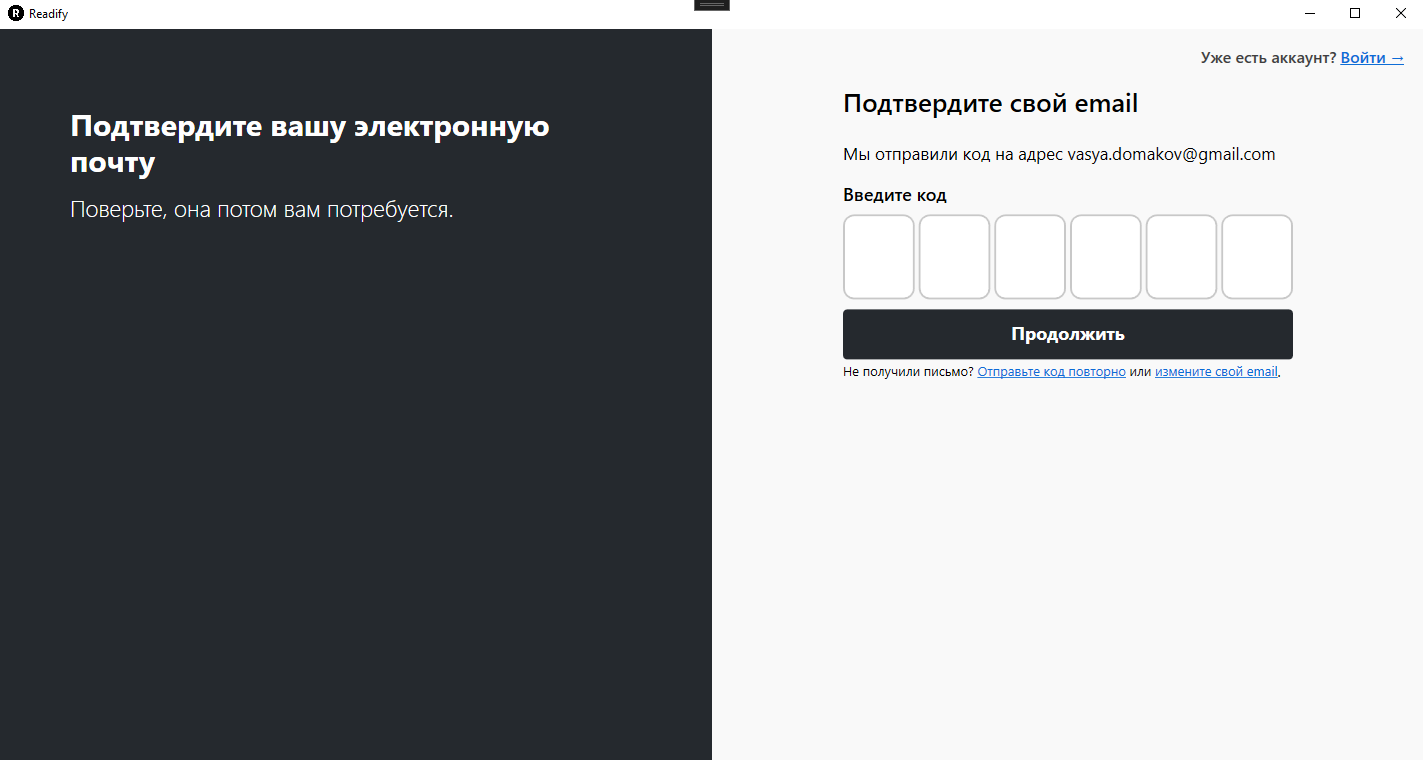


Рисунок – Окно третьего этапа регистрации

Теперь можно заполнить данные профиля (опционально). Чтобы пропустить этот этап, необходимо нажать на «Войти» в правом верхнем углу экрана. Окно четвертого этапа регистрации представлено в соответствии с рисукном 22.

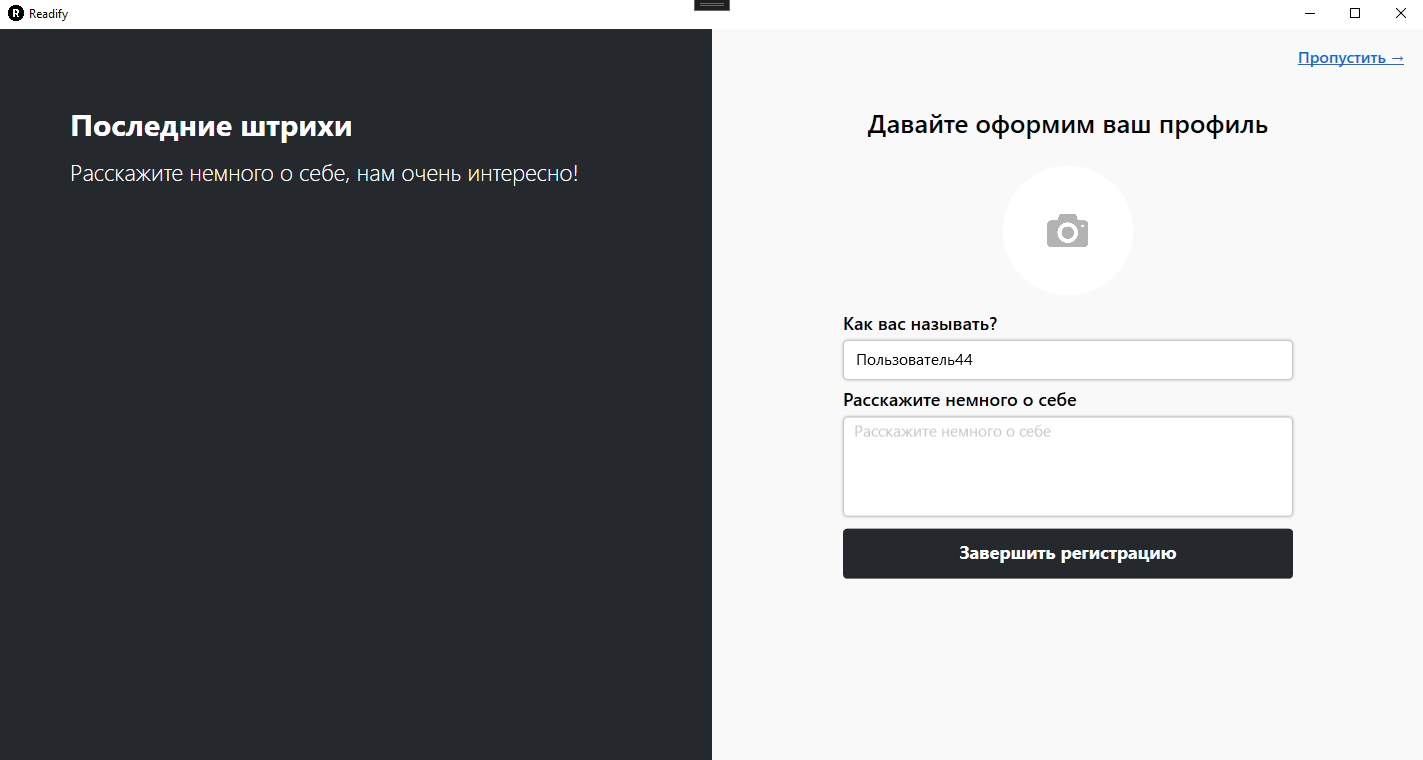


Рисунок – Окно четвертого этапа регистрации

После регистрации создаться аккаунт. Чтобы в него войти необходимо ввести свой логин или почту и пароль в окне авторизации. Окно авторизации представлено в соответствии с рисунком 23.



Рисунок – Окно авторизации

После авторизации покажется окно профиля. Окно профиля представлено в соответствии с рисунком 24.

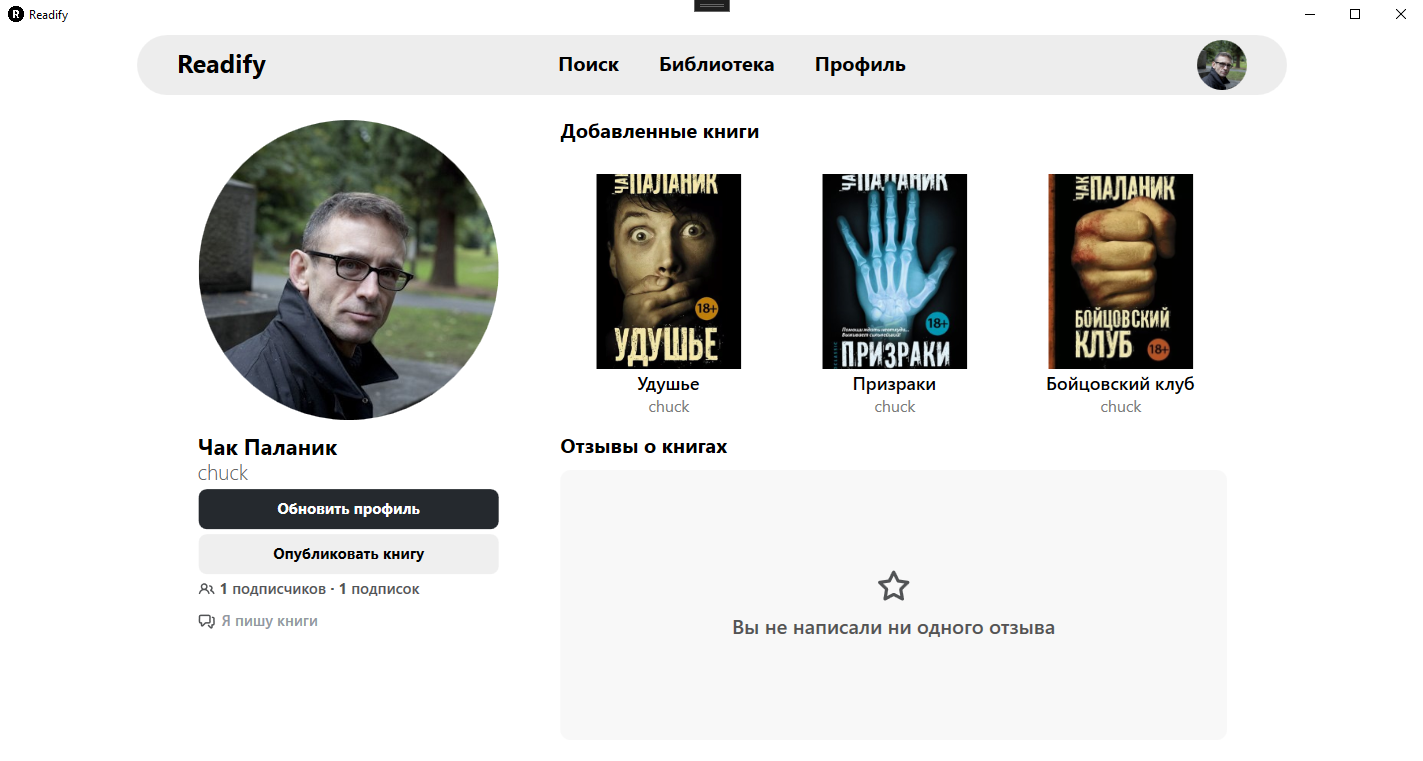


Рисунок – Окно профиля

Сверху присутствует панель навигации, с помощью нее можно попасть в меню поиска библиотеки и профиля. В меню поиска можно искать книги и пользователей. Окно поиска представлено в соответствии с рисунком 25.

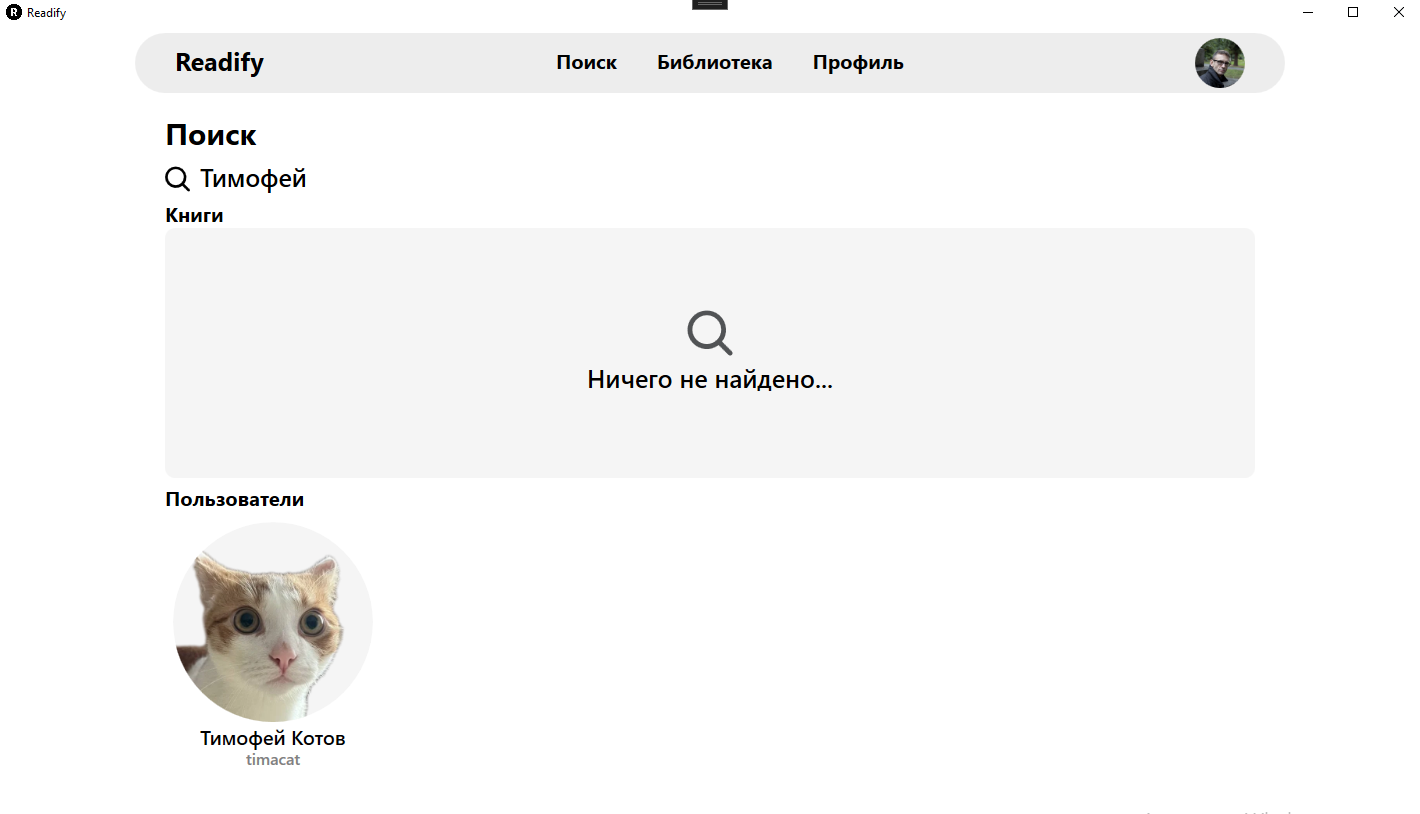


Рисунок – Окно поиска

В блоке с найденными пользователями можно нажать на одного из них и попасть в его профиль. Окно профиля пользователя представлено в соответствии с рисунком 26.

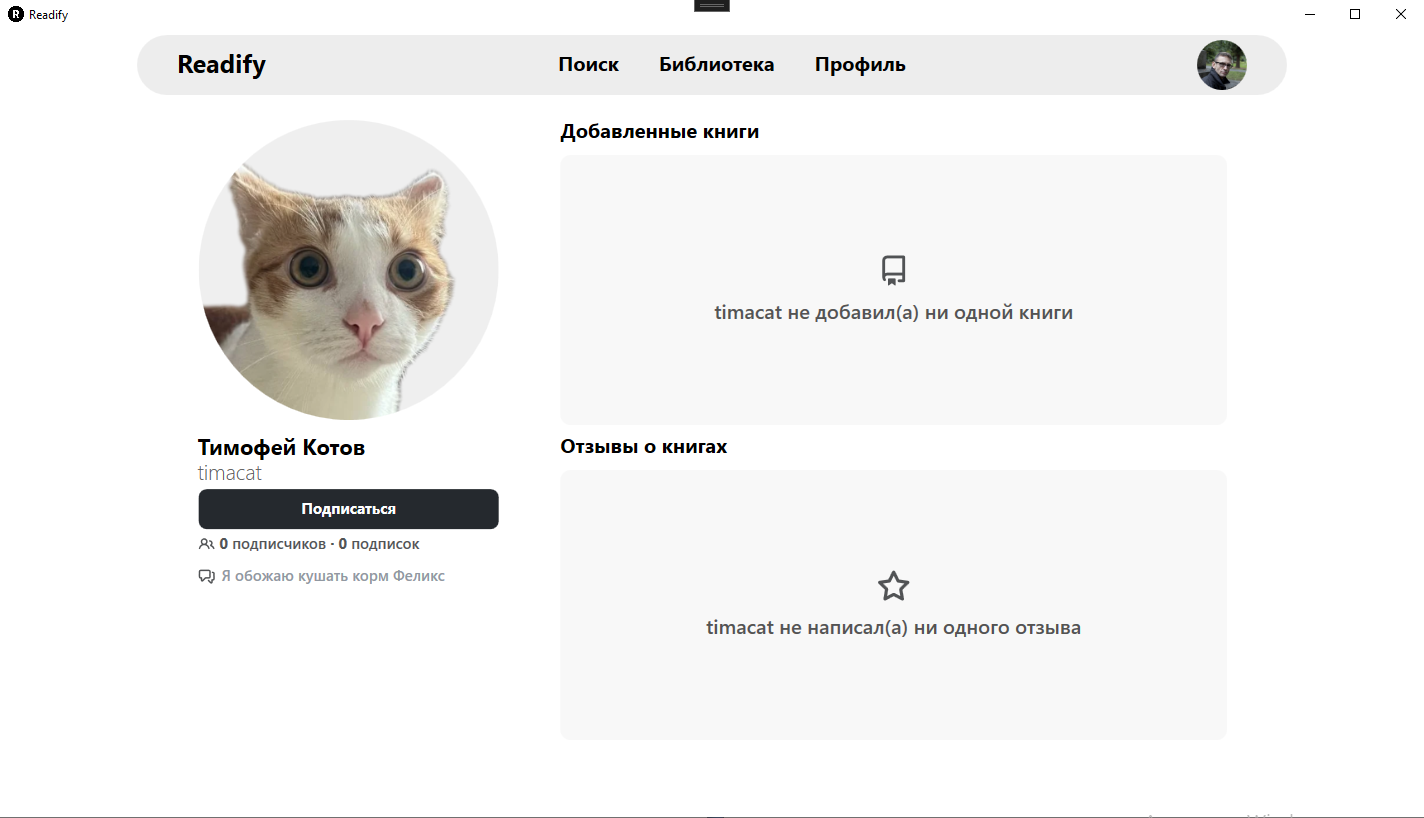


Рисунок – Окно профиля пользователя

В профиле другого пользователя можно увидеть какие книги он добавлял в свою библиотеку, какие отзывы написал, также можно подписаться и отписаться от него. Всех подписчиков и подписок пользователя можно посмотретьнажав на надпись N подписчиков и N подписок соответственно. Окно подписчиков представлено в соответствии с рисунком 27.

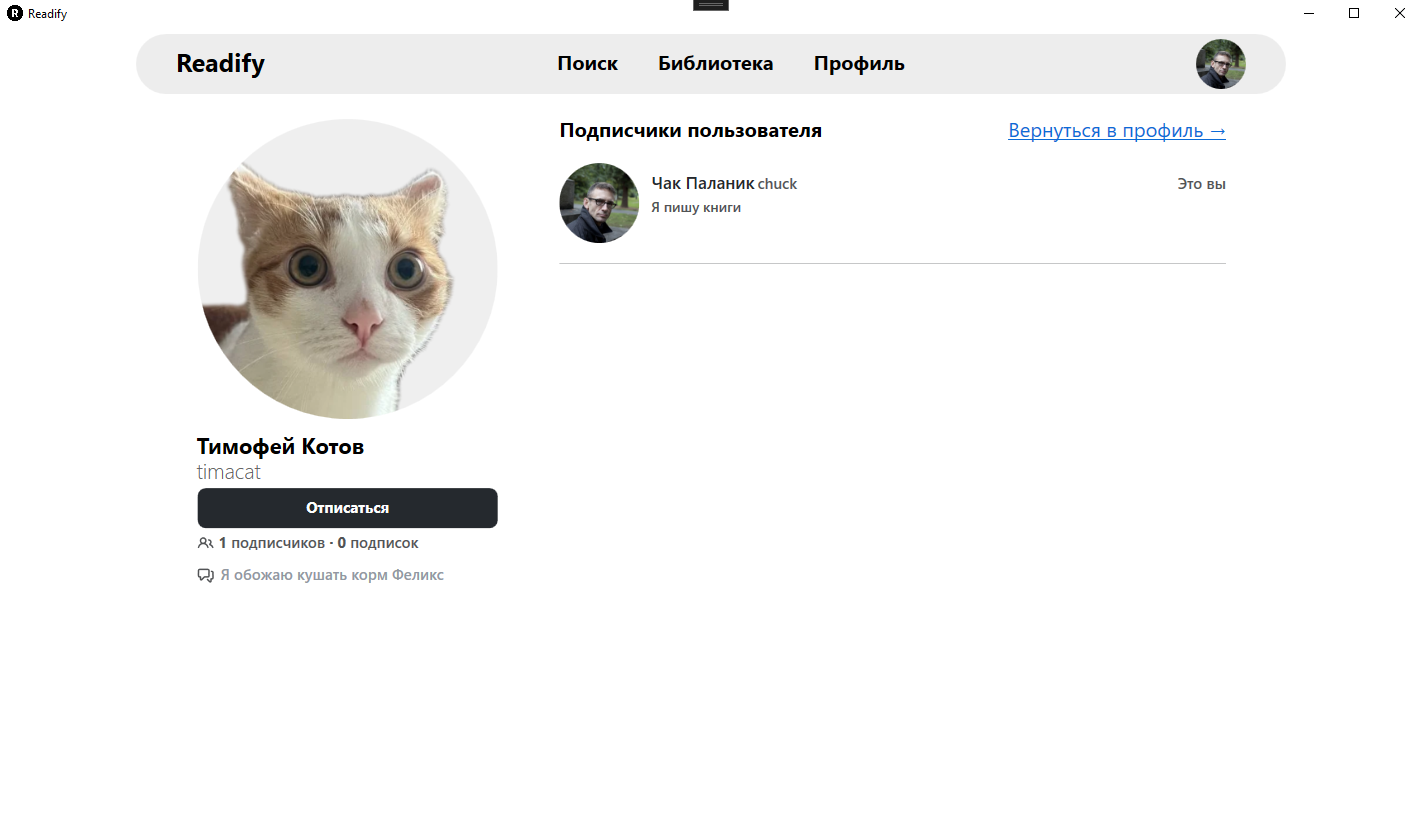


Рисунок – Окно подписчиков пользователя

Окно подписок представлено в соответствии с рисунком 28.

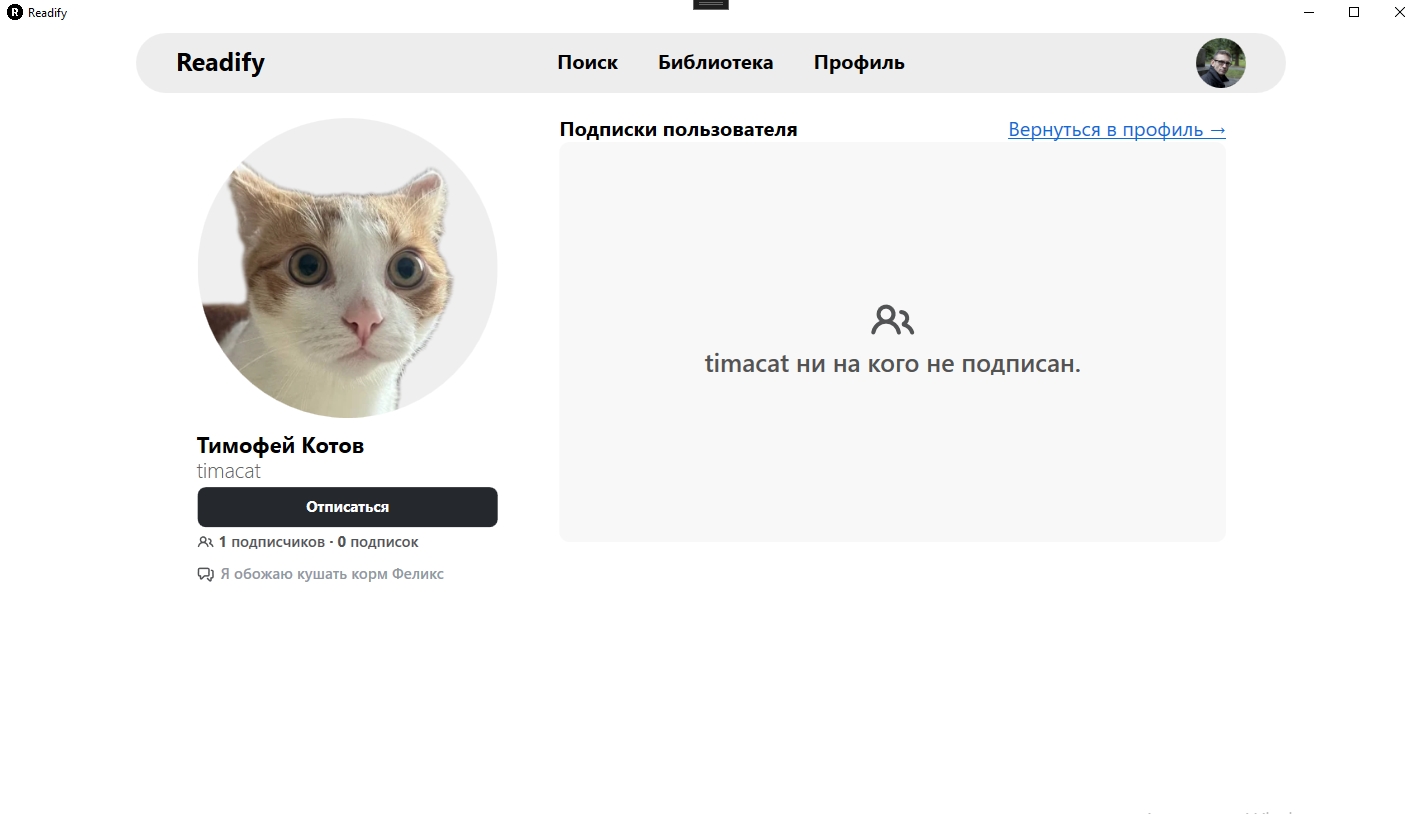


Рисунок – Окно подписок пользователя

В меню библиотеки можно посмотреть какие книги были добавлены и опубликованы. Окно библиотеки представлено в соответствии с рисунком 29.

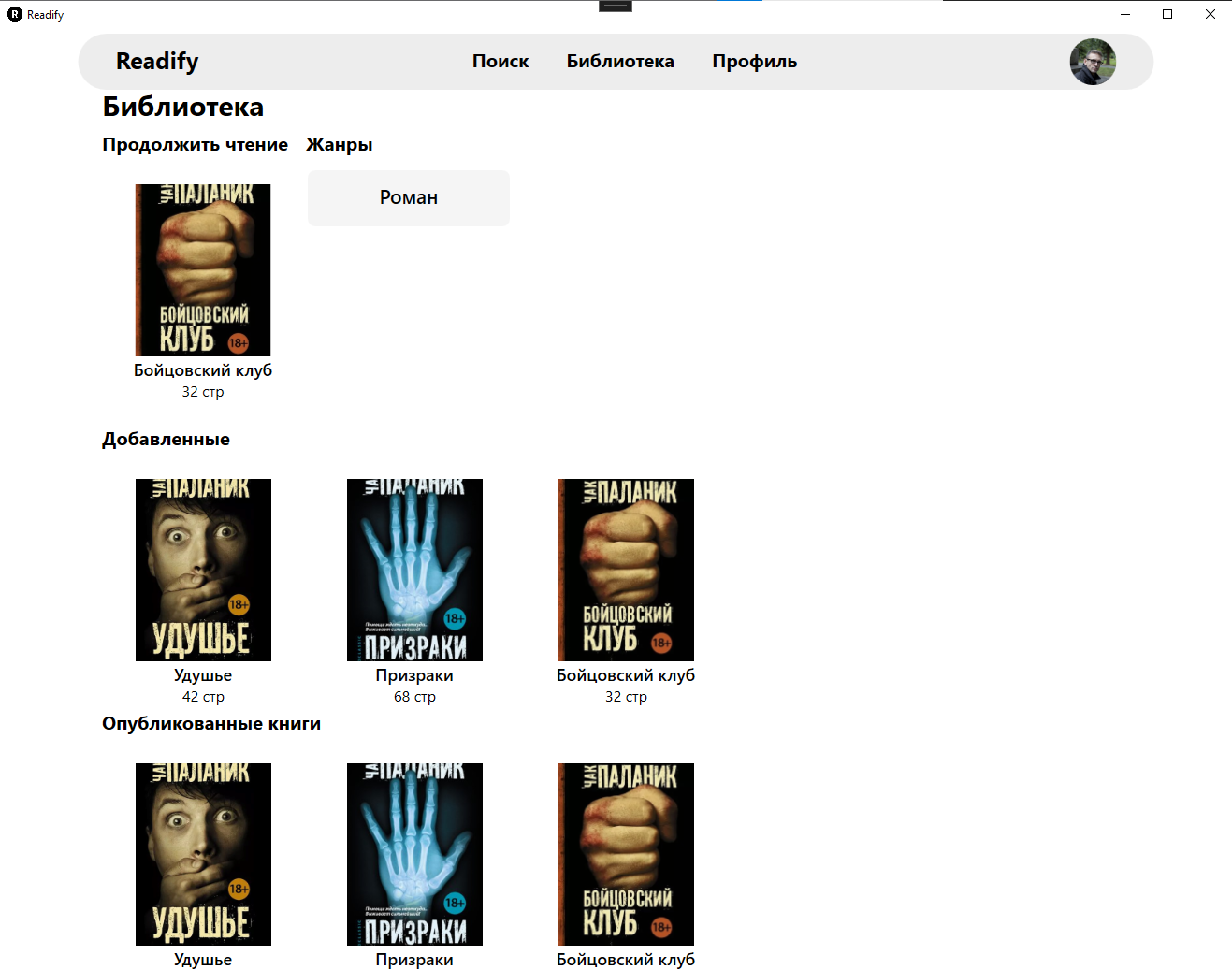


Рисунок – Окно библиотеки

Из меню поиска, библиотеки и профиля можно попасть в меню книги, нажав на блок с книгой. Окно книги представлено в соответствтии с рисунком 30.

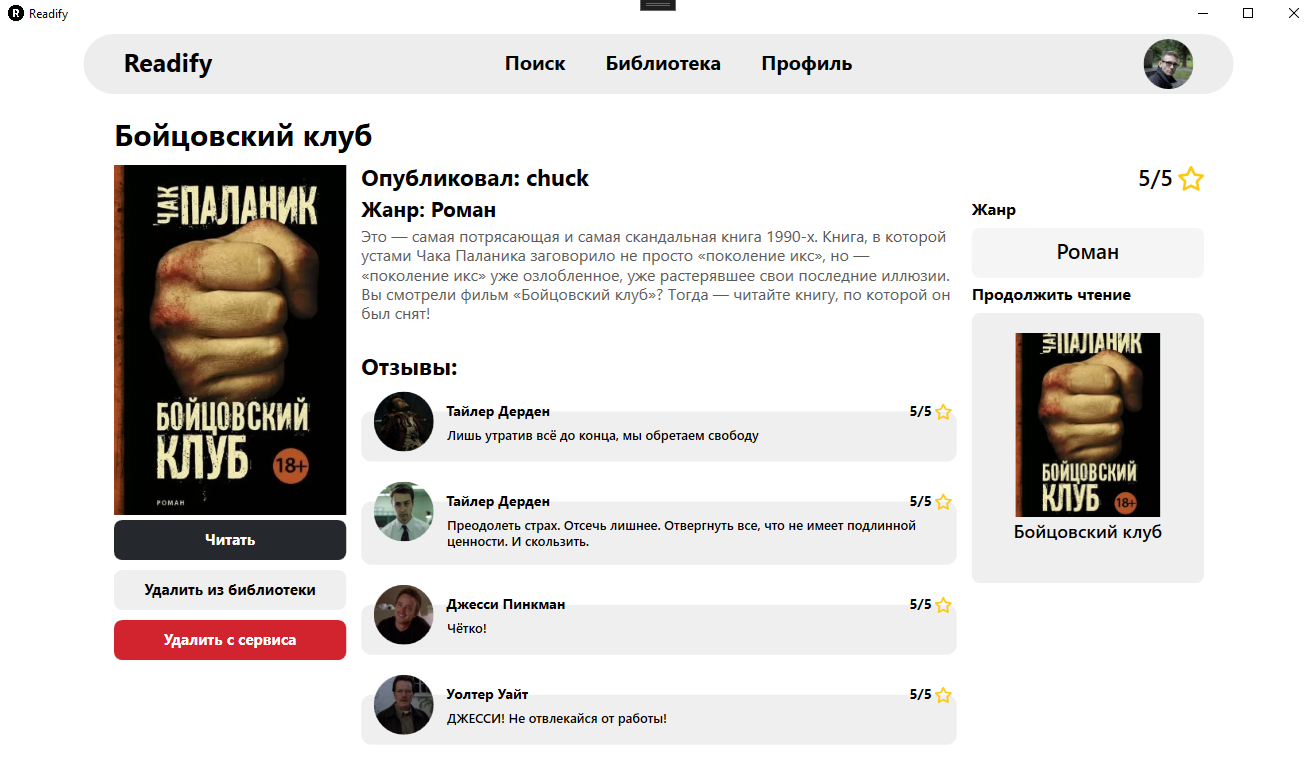


Рисунок – Окно книги

Книгу можно удалить с сервиса, нажав на кнопку «Удалить с сервиса», после этого появится окно подтверждения. Окно подтверждения удаления книги представлено в соответствии с рисунком 31.

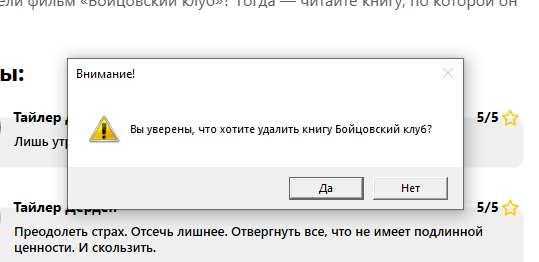


Рисунок – Окно подтверждения удаления книги

В меню пользователя можно изменять аккаунт, нажав на кнопку «Обновить профиль». Окно обновления профиля представлено в соответствии с рисунком 32.

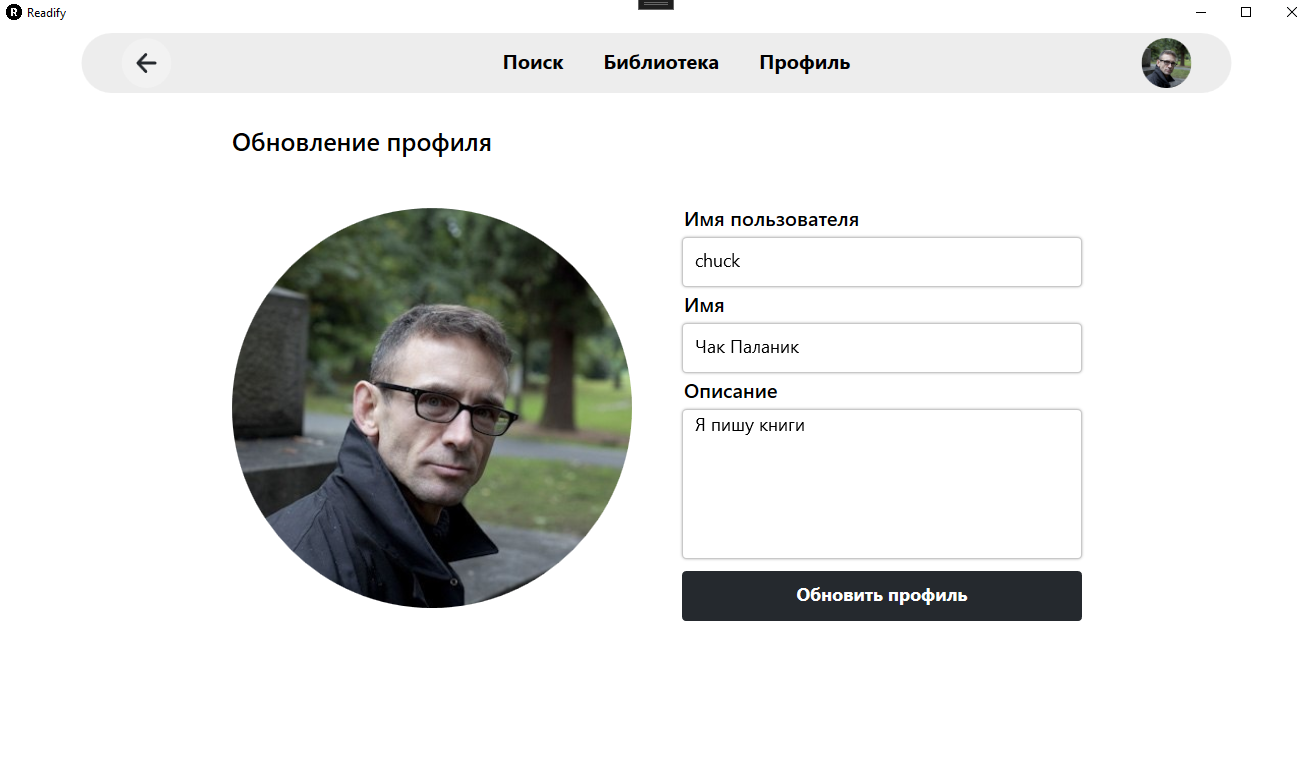


Рисунок – Окно обновления профиля

Также в окне профиля есть кнопка «Опубликовать книгу», нажав на неё можно опубликовать новую книгу. Окно публикации книги представлено в соответствии с рисунком 33.

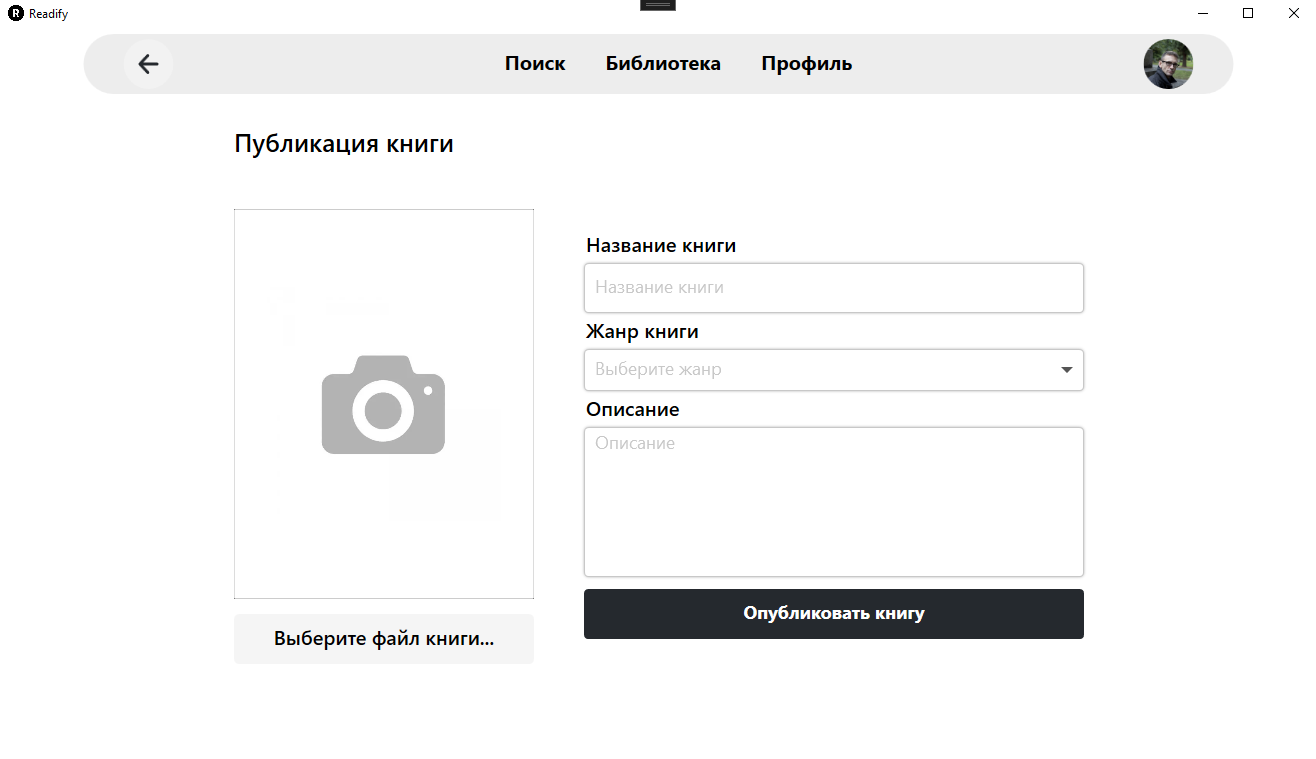


Рисунок – Окно публикации книги

ЗАКЛЮЧЕНИЕ

Целью курсового проекта была разработка модуля информационной системы для чтения книг. Это приложение позволит пользователям хранить, коммуницировать и читать книги.

В ходе работы были составлены требования к модулю информационной системы, проанализированы средства разработки программного обеспечения и разработан привлекательный графический дизайн.

Следующим шагом было проведено тестирование функционала всех разделов системы, чтобы подтвердить высокое качество разработанного проекта. Способность кода к сопровождению была повышена за счёт написания комментариев и неоднократного рефакторинга. По завершению работы над проектом была разработана обширная документация.

Таким образом, поставленная цель и задачи выполнены в полном объёме. В целом было проделано колоссальное количество работы, а приобретённые навыки значительно повысили мои профессиональные способности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

* + - 1. Microsoft Learn : официальный сайт – URL : <https://visualstudio.microsoft.com/ru/> (дата обращения: 10.03.2025). Текст – электронный.
      2. Microsoft Learn MS SQL : официальный сайт – URL : <https://www.microsoft.com/ru-ru/sql-server/sql-server-downloads> (дата обращения 10.03.2025). Текст – электронный.
      3. Разработчик программного обеспечения JetBrains Rider : официальный сайт – URL : <https://www.jetbrains.com/> (дата обращения: 10.03.2025). Текст – электронный.
      4. Разработчик программного обеспечения Eclipse : официальный сайт – URL : <https://www.eclipse.org/downloads/> (дата обращения: 10.03.2025). Текст – электронный.
      5. Разработчик программного обеспечения MySQL : официальный сайт – URL : <https://www.mysql.com/> (дата обращения: 10.03.2025). Текст – электронный.
      6. Разработчик программного обеспечения Oracle : официальный сайт – URL : <https://www.oracle.com/cis/database/> (дата обращения: 10.03.2025). Текст – электронный.
      7. Сообщество о программировании Metanit : официальный сайт. – URL : <https://metanit.com/cpp/> (дата обращения: 10.03.2025). Текст – электронный.
      8. Свободная энциклопедия Wikipedia : официальный сайт – URL : <https://ru.wikipedia.org/wiki/Pascal> (дата обращения: 10.03.2025). Текст – электронный.
      9. Свободная энциклопедия Wikipedia : официальный сайт – URL : <https://ru.wikipedia.org/wiki/C_sharp> (дата обращения: 10.03.2025). Текст – электронный.
      10. Форум разработчиков Laravel : официальный сайт. – URL : <https://laravel.ru/posts/1427> (дата обращения: 06.05.2025). Текст – электронный.

ПРИЛОЖЕНИЕ 1

(обязательное)

Бизнес-требования

Системные характеристики:

СХ-1. Платформа:

СХ-1.1. Приложение является настольным.

СХ-2. Технологический стек.

СХ-2.1. Разработка осуществляется с использованием .NET 7.0.

СХ-2.2. Для хранения данных используется СУБД Microsoft SQL Server 2022 Express Edition.

СХ-2.3. Целевая операционная система: Windows 11.

СХ-3. Авторизация:

СХ-3.1. Реализуется с использованием JWT-токенов.

Пользовательские требования

ПТ-1. Пользователи системы:

ПТ-1.1. Пользователи-читатели:

* могут читать книги;
* могут загружать книги (в PDF формате);
* могут оставлять отзывы о книгах;
* могут просматривать профили других пользователей;
* могут изменять информацию своего профиля;

ПТ-1.2. Администраторы:

* обладают всеми правами пользователей-читателей:
* могут удалять и редактировать книги.

ПТ-2. Запуск и закрытие приложения:

ПТ-2.1. Запуск приложения осуществляется двойным нажатием по исполняемому файлу приложения.

ПТ-2.2. Закрытие приложения происходит нажатием на кнопку «Закрыть» (крестик в правом верхнем углу).

ПТ-3. Регистрация:

ПТ-3.1. Пользователь может зарегистрироваться в системе, указав следующие данные:

* логин (только английские буквы и цифры, не менее 5 символов);
* email (валидный адрес электронной почты);
* имя (только буквы, не менее 2 символов);
* пароль (соответствует требованиям безопасности: минимум 8 символов, включая 1 цифру, 1 заглавную букву, 1 строчную букву и 1 специальный символ).

ПТ-3.2 После успешной регистрации автоматически происходит аутентификация.

ПТ-4. Вход в систему:

ПТ-4.1. При запуске приложения, если пользователь не аутентифицирован, появляется окно входа.

ПТ-4.2. Для входа пользователь должен ввести:

* логин/email;
* пароль.

ПТ-4.3. Перед успешным входом должна быть пройдена капча.

ПТ-4.4. Если логин или пароль неверны, выводится сообщение: «Неверный логин или пароль».

ПТ-4.5. Если подключение к серверу отсутствует, выводится сообщение: «Подключение к серверу не установлено».

ПТ-5 Выход из системы:

ПТ-5.1. Выход из системы осуществляется нажатием на кнопку «Выйти из аккаунта».

ПТ-6. Работа с данными:

ПТ-6.1. Профиль пользователя:

ПТ-6.1.1. Пользователь может просматривать свой профиль и профили других пользователей.

ПТ-6.1.2. Пользователь может изменять информацию в своем профиле:

* прозвище;
* имя;
* описание профиля;
* аватар (PNG формат, максимум 5 МБ).

ПТ-6.2 Книги:

ПТ-6.2.1. Пользователь может загружать книги (PDF формат).

ПТ-6.2.2. Пользователь может добавлять обложки для книг (PNG формат, максимум 5 МБ).

ПТ-6.2.3. Пользователь может оставлять отзывы о книгах (максимум 500 символов).

ПТ-7. Права администратора:

ПТ-7.1. Администратор может удалять книги.

Детальные спецификации:

ДС-1. Окно входа:

ДС-1.1. Окно содержит поля:

* логин/email.
* пароль;
* кнопка «Войти»;
* кнопка «Зарегистрироваться»;
* кнопка «Я забыл пароль».

ДС-1.2. При нажатии на кнопку «Войти»:

ДС-1.2.1. Если логин или пароль неверны, выводится сообщение: «Неверный логин или пароль»;

ДС-1.2.2. Если подключение к серверу отсутствует, выводится сообщение: «Подключение к серверу не установлено».

ДС-1.3. При нажатии на кнопку «Я забыл пароль» открывается окно восстановления пароля:

ДС-2. Окно регистрации:

ДС-2.1. Окно содержит поля:

* логин (только английские буквы и цифры, не менее 3 символов);
* email (валидный адрес электронной почты);
* имя (только буквы, не менее 2 символов);
* пароль (минимум 8 символов, включая 1 цифру, 1 заглавную букву, 1 строчную букву и 1 специальный символ);
* кнопка «Зарегистрироваться»;
* кнопка «У меня уже есть аккаунт»;

ДС-2.2. При нажатии на кнопку «Зарегистрироваться»:

ДС-2.2.1. Если какое-либо поле не заполнено, выводится соответствующее сообщение об ошибке.

ДС-2.2.2. Если данные корректны, пользователь регистрируется и автоматически аутентифицируется.

ДС-3. Меню информационной системы:

ДС-3.1. В верхней части окна расположены кнопки:

* «Свернуть»;
* «Развернуть»;
* «Закрыть».

Атрибуты качества:

АК-1. Фильтрация:

АК-1.1. Должна быть возможность фильтровать книги по жанру.

Требования к данным

ТД-1. Пароли:

ТД-1.1. Пароли в базе данных хранятся в зашифрованном виде.

ТД-2. Книги:

ТД-2.1. Допустимый формат: PDF.

ТД-3. Изображения:

ТД-3.1. Допустимый формат: PNG.

Ограничения:

О-1. Комментарии:

О-1.1. Максимальная длина: 500 символов.

О-3. JWT-токены:

О-3.1. Refresh-токен действителен 150 дней.

ПРИЛОЖЕНИЕ 2

Тестирование логики программы представлено в таблице 1.

Таблица 1 – Тестирование логики программы.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Описание | Действие | Ожидаемый результат | Фактический результат | Статус |
| 1 | Запуск приложения с корректным подключением к серверу | 1. Убедитесь, что системный администратор запустил сервер, что в файле MaxiSocialNetwork.dll.config введен правильный IP сервера  2. Запустите исполняемый файл приложения. | Открытие окна входа в систему. | Открытие окна входа в систему. | Пройден |
| 2 | Запуск приложения с некорректным IP адресом | 1. В файле MaxiSocialNetwork.dll.config поменяйте IP адрес на несуществующий.  2. Запустите исполняемый файл приложения. | Открытие диалогового окна с надписью «Подключение к серверу не установлено» и кнопкой «Ок». | Открытие диалогового окна с надписью «Подключение к серверу не установлено» и кнопкой «Ок». | Пройден |
| 3 | Попытка входа с неверным паролем. | 1. Запустите приложение.  2. Введите в поле ввода логина «mrbomba», а в поле ввода пароля «1111».  3. Нажмите кнопку «Войти». | Открытие диалогового окна с надписью: «Был введен неправильный пароль. Проверьте его, пожалуйста.» и кнопкой «Ок». | Открытие диалогового окна с надписью: «Был введен неправильный пароль. Проверьте его, пожалуйста.» и кнопкой «Ок». | Пройден |
| 4 | Попытка входа с несуществующим пользователем, но существующим паролем. | 1. Запустите приложение.  2. Введите в поле ввода логина «oooo», а в поле ввода пароля «Password1!».  3. Нажмите кнопку «Войти». | Открытие диалогового окна с надписью: «Был введен неправильный пароль. Проверьте его, пожалуйста.» и кнопкой «Ок». | Открытие диалогового окна с надписью: «Был введен неправильный пароль. Проверьте его, пожалуйста.» и кнопкой «Ок». | Пройден |
| 5 | Попытка зарегистрироваться с пустым логином | 1. Запустите приложение.  2. Перейдите в окно регистрации.  3. Заполните все поля, кроме логина.  4. Нажмите на кнопку «Зарегистрироваться» | Открытие диалогового окна с надписью: «Логин не может быть пустым!» и кнопкой «Ок». | Открытие диалогового окна с надписью: «Логин не может быть пустым!» и кнопкой «Ок». | Пройден |

Продолжение таблицы 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 | Попытка зарегистрироваться с пустым именем | 1. Запустите приложение.  2. Перейдите в окно регистрации.  3. Заполните все поля, кроме имени.  4. Нажмите на кнопку «Зарегистрироваться» | Открытие диалогового окна с надписью: «Имя не может быть пустым!» и кнопкой «Ок». | Открытие диалогового окна с надписью: «Имя не может быть пустым!» и кнопкой «Ок». | Пройден |
| 7 | Попытка зарегистрироваться с пустой фамилией | 1. Запустите приложение.  2. Перейдите в окно регистрации.  3. Заполните все поля, кроме фамилии.  4. Нажмите на кнопку «Зарегистрироваться» | Открытие диалогового окна с надписью: «Фамилия не может быть пустой!» и кнопкой «Ок». | Открытие диалогового окна с надписью: «Фамилия не может быть пустой!» и кнопкой «Ок». | Пройден |
| 8 | Попытка зарегистрироваться с пустым паролем | 1. Запустите приложение.  2. Перейдите в окно регистрации.  3. Заполните все поля, кроме пароля.  4. Нажмите на кнопку «Зарегистрироваться» | Открытие диалогового окна с надписью: «Длина пароля должна быть не менее 5 символов» и кнопкой «Ок». | Открытие диалогового окна с надписью: «Длина пароля должна быть не менее 5 символов» и кнопкой «Ок». | Пройден |
| 9 | Попытка зарегистрироваться с некорректным логином | 1. Запустите приложение.  2. Перейдите в окно регистрации.  3. Введите логин «kirya»  4. Нажмите на кнопку «Зарегистрироваться» | Открытие диалогового окна с надписью: «Логин должен содержать и цифры, и буквы» и кнопкой «Ок». | Открытие диалогового окна с надписью: «Логин должен содержать и цифры, и буквы» и кнопкой «Ок». | Пройден |

Продолжение таблицы 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 14 | Попытка редактиктирования профиля с неверной почтой | 1. Запустите приложение.  2. Перейдите в окно профиля  3. Поменяйте почту на «pochta»  4. Нажмите на кнопку «Сохранить». | Открытие диалогового окна с надписью: «Была введена некорректная почта» и кнопкой «Ок». | Открытие диалогового окна с надписью: «Была введена некорректная почта» и кнопкой «Ок». | Пройден |
| 15 | Попытка редактиктирования профиля с неверным телефоном | 1. Запустите приложение.  2. Перейдите в окно профиля  3. Поменяйте номер на «55555»  4. Нажмите на кнопку «Сохранить». | Открытие диалогового окна с надписью: «Был введен некорректный номер телефона» и кнопкой «Ок». | Открытие диалогового окна с надписью: «Был введен некорректный номер телефона» и кнопкой «Ок». | Пройден |
| 16 | Попытка редактиктирования профиля с пустым именем | 1. Запустите приложение.  2. Перейдите в окно профиля  3. Очистите имя  4. Нажмите на кнопку «Сохранить». | Открытие диалогового окна с надписью: «Укажите имя!» и кнопкой «Ок». | Открытие диалогового окна с надписью: «Укажите имя!» и кнопкой «Ок». | Пройден |
| 17 | Попытка редактиктирования профиля с пустой фамилией | 1. Запустите приложение.  2. Перейдите в окно профиля  3. Очистите фамилию  4. Нажмите на кнопку «Сохранить». | Открытие диалогового окна с надписью: «Укажите фамилию!» и кнопкой «Ок». | Открытие диалогового окна с надписью: «Укажите фамилию!» и кнопкой «Ок». | Пройден |

ПРИЛОЖЕНИЕ 3