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 сотрудник сможет управлять библиотекой, добавлять новые книги и читать их. Система также должна предоставлять инструменты для управления контентом, такие как добавление обложек, описаний и жанров.

Подробные требования к разрабатываемому программному обеспечению находятся 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 […]. 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 [...]. 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 […]. 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 […]. 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 […]. 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 […]. 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 […]. 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 […]. 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 […]. 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);
* Is\_banned – тип данных bit, содержит статус блокировки пользователя (по умолчанию FALSE);
* 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 BannedUsers

* Id – ключевое поле, тип данных int, уникальный идентификатор записи (автоинкремент);
* Id\_user – тип данных int, ссылается на таблицу Users (идентификатор заблокированного пользователя);
* Ban\_reason – тип данных nvarchar(200), содержит причину блокировки (может быть NULL);
* Banned\_at – тип данных datetime, заполняется датой и временем блокировки.

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 Bookmarks

* Id – ключевое поле, тип данных int, уникальный идентификатор записи (автоинкремент);
* Id\_library – тип данных int, ссылается на таблицу Libraries (идентификатор библиотеки, к которой относится закладка);
* Page – тип данных int, содержит номер страницы, на которую сделана закладка;
* Comment – тип данных nvarchar(100), содержит комментарий к закладке (может быть NULL);
* Created\_at – тип данных datetime, заполняется датой и временем создания закладки.

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 LikesReviews

* Id – ключевое поле, тип данных int, уникальный идентификатор записи (автоинкремент);
* Id\_author – тип данных int, ссылается на таблицу Users (идентификатор пользователя, который поставил реакцию);
* Id\_review – тип данных int, ссылается на таблицу BookReviews (идентификатор отзыва, к которому относится реакция);
* Reaction\_type – тип данных tinyint, содержит тип реакции (1 (лайк) или 0(дизлайк));
* Created\_at – тип данных datetime, заполняется датой и временем установки реакции.

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 1.

Изображение выглядит как текст, диаграмма, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1 – ER-диаграмма базы данных.

ПРИЛОЖЕНИЕ 1

Бизнес-требования

Суть проекта:

Разработка модуля информационной системы для сервиса чтения книг, который позволит пользователям читать книги, загружать их, оставлять отзывы и взаимодействовать с другими пользователями через комментарии и реакции (лайки/дизлайки). Администраторы смогут управлять аккаунтами пользователей и контентом.

Цели проекта:

* создание удобной платформы для чтения книг и взаимодействия с контентом;
* повышение вовлеченности пользователей за счет социальных функций (комментарии, рецензии);
* упрощение процесса поиска, чтения и управления книгами.

1. Системные характеристики

СХ-1. Платформа.

СХ-1.1. Приложение является настольным.

СХ-1… Возможно + веб, android

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 10.

СХ-3. Авторизация.

СХ-3.1. Реализуется с использованием JWT-токенов.

1. Пользовательские требования

ПТ-1. Пользователи системы.

ПТ-1.1. Пользователи-читатели:

* могут читать книги;
* могут загружать книги (в PDF формате);
* могут изменять загруженные книги;
* могут оставлять отзывы о книгах;
* могут оставлять реакции (лайки/дизлайки) на отзывы других пользователей;
* могут просматривать профили других пользователей;
* могут изменять информацию своего профиля;
* могут добавлять закладки в книги.

ПТ-1.2. Администраторы:

* обладают всеми правами пользователей-читателей:
* могут блокировать и разблокировать аккаунты пользователей;
* 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-6.2.4. Пользователь может ставить лайки/дизлайки на отзывы других пользователей.

ПТ-6.3. Закладки.

ПТ-6.3.1. Пользователь может добавлять закладки в книги с комментарием (максимум 100 символов).

ПТ-7. Права администратора

ПТ-7.1. Администратор может блокировать и разблокировать аккаунты пользователей.

ПТ-7.2. Администратор может удалять и редактировать книги.

Диаграмма вариантов использования представлена в соответствии с рисунком 1.

Изображение выглядит как текст, человек, Человеческое лицо, мультфильм

Автоматически созданное описание

Рисунок 1 – Диаграмма вариантов использования

1. 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».

ДС-3.2. Цвет меню соответствует основному цвету приложения. (#.....)

ДС-N

…

1. Атрибуты качества

АК-1. Фильтрация и сортировка:

АК-1.1. Должна быть возможность фильтровать книги по рейтингу, жанру.

АК-1.2. Должна быть возможность сортировать книги по алфавиту, дате добавления, популярности.

1. 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.

1. Ограничения

О-1. Комментарии.

О-1.1. Максимальная длина: 800 символов.

О-2. Лайки/дизлайки

О-2.1. Пользователь может поставить только одну реакцию на отзыв.

О-3. JWT-токены:

О-3.1. Refresh-токен действителен 150 дней.