Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліні “ Об’єктно -орієнтоване програмування”

ОСОБИСТА БІБЛІОТЕКА

Керівник, професор Бондарєв В. М.

Студент гр. ПЗПІ-19-4 Кошкіна Д. О.

Комісія:

ст. викл. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Черепанова Ю. Ю.

професор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бондарєв В. М.

доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Побіженко І. О.

Харків 2020

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра: *Програмної інженерії*

Дисципліна: *Об’єктно-орієнтоване програмування*

Спеціальність: *121*  *Інженерія програмного забезпечення*

Освітня програма: *Програмна інженерія*

Курс 1 .Група *ПЗПІ-19 -4* . Семестр *2* .

*ЗАВДАННЯ*

*на курсовий проект студента*

*Кошкіної Дар’ї Олександрівни*

(Прізвище, Ім'я, По батькові)

1 Тема проекту: «Особиста бібліотека».

2 Термін здачі студентом закінченого проекту: *“19” - червня - 2020 р*.

3 Вихідні дані до проекту:

*Специфікація програми, методичні вказівки до виконання курсової роботи*

4 Зміст розрахунково-пояснювальної записки:

*Вступ,* *специфікація програми, проектна специфікація, інструкція користувача, висновки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 Перелік графічного матеріалу:

\_\_ *Схема об’єктної моделі, діаграма класів, приклади екранних форм* \_\_\_\_ *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ КАЛЕНДАРНИЙ ПЛАН

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження і затвердження теми | 21-02-2020 р. |
| 2 | Формулювання вимог до програми | 30-03-2020 – 12-04-2020 р. |
| 3 | Створення класів та загальної ієрархії програми. | 13-04-2020 – 19-04-2020 р. |
| 4 | Розробка функції зчитування, зберігання та редагування даних. Робота з файлами. | 20-04-2020 – 26-04-2020 р. |
| 5 | Створення основних функцій бібліотеки, таких як додавання та видалення книг, пошук, сортування, функція інвентаризації. | 27-04-2020 – 03-05-2020 р. |
| 6 | Розробка інтерфейса користувача, обробка виключень при роботі з файлами. | 04-05-2020 – 15-05-2020 р. |
| 7 | Тестування і доопрацювання розробленої програмної системи. | 18-05-2020 – 24-05-2020 р. |
| 8 | Оформлення пояснювальної записки, додатків, графічного матеріалу | 25-05-2020 – 05-06-2020 р. |
| 9 | Захист | 01-06-2018 – 19-06-2020 р. |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Прізвище, Ім'я, По батькові)

« 21 »\_лютого\_\_\_\_\_\_\_\_\_\_ 2020 р.

РЕФЕРАТ

Пояснювальна записка до курсової роботи: 31 с., 23 рис., 2 діаграми, 1 додаток, 5 джерел.

ГІСТЬ, АДМІНІСТРАТОР, БІБЛІОТЕКА, КНИГА, ДОДАТОК, КЛАС, МОВА ПРОГРАМУВАННЯ C#, ОБ’ЕКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПРОГРАМА, ПЛАТФОРМА .NET.

Метою роботи є розробка програми “Особиста бібліотека” на засадах об'єктно-орієнтованого програмування.

Методи розробки базуються на використанні середи розробки Microsoft Visual Studio 2019, Windows Forms, платформи .NET Framework 4.7, мови програмування C#.

В результаті отримана програма під назвою “ Особиста бібліотека”, яка дозволяє зберігати та редагувати список книг, характеристики кожної книги, такі як: назва, автор, рік видання, жанр, анотація, кількість лайків. У адміністраторів є можливість реєструватися, додавати, видаляти та редагувати книги, також проводити інвентаризацію. Гості мають можливість переглядати книги по розділах, користуватися пошуком, ставити лайки та брати почитати книжки. Усі функції гостя доступні адміністратору.

ЗМІСТ

ВСТУП……………………………………………………………………………..6

1. СПЕЦІФІКАЦІЯ ПРОГРАМИ……………………………………………….8
   1. Опис програми на тему «Особиста бібліотека»……………………...8
   2. Користувачі програми…………………………………………………8
   3. Функції програми………………………………………………………8
   4. Функції адміністратора………………………………………………...9
   5. Функції гостя………………………………………………………….14
2. ПРОЕКТУВАННЯ ПРОГРАМИ……………………………………………17
   1. Вибір архітектури……………………………………………………..17
   2. Основна структура програми………………………………………...17
   3. Об’єктна модель………………………………………………………18
   4. Сховище даних та доступ до нього………………………………….20
3. ІНСТРУКЦІЯ КОРИСТУВАЧЕВІ……………………………………...…..21
   1. Інструкція адміністратору……………………………………………21
   2. Інструкція гостю………………………………………………………25

ВИСНОВКИ……………………………………………………………………...29

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ………………………………………………30

ДОДАТОК А……………………………………………………………………..31

ВСТУП

У наш час ми спостерігаємо технічний прорив. Розвиток технологій і винахід нових пристроїв стрімко зростає. Це також дозволяє цим пристроям бути доступними для більшості людей. Зараз кожен забезпечений достатньою кількістю гаджетів і наявністю інтернету. У зв'язку з цим актуальна розробка усіляких додатків. Розвиток інформаційних технологій сприяє еволюційній зміні у процесах зберігання і поширення інформації в електронній формі. Це дозволяє зберігати інформацію найбільш надійно і компактно, оперативніше і ширше поширювати її. Основними засобами для реалізації цих можливостей є електронні видання та електронні бібліотеки, які все частіше використовують мультимедійну складову в обслуговуванні своїх користувачів.

Метою моєї роботи стала розробка програми особистої бібліотеки. Такого роду електронна бібліотека дозволяє користувачеві зберігати списки книг, шукати потрібну книгу, додавати книги в список для читання або лайкати їх. Додаток розроблявся для двох типів користувачів: адміністраторів і гостей. Адміністратор редагує списки книг і проводить інвентаризацію, гість користується бібліотекою.

Окрім розробки даного додатку метою курсового проекту є здобути навички об’єктно-орієнтовного програмування на мові С#, навчитися працювати з Windows Forms.

Дана програма може служити шаблоном для повноцінного додатку електронної бібліотеки і широко використовуватися людьми для читання з електронних пристроїв, що є зручним і вигідним рішенням.

Задачі проекту:

* створення логіки роботи програми,
* додавання функції зчитування та збереження інформації за допомогою файлів. Робота з файлами,
* створення функцій користувача,
* тестування програми,
* створення графічного інтерфейсу,
* надання стійкості програмі.

1. СПЕЦИФІКАЦІЯ ПРОГРАМИ
   1. Опис завдання на тему «Особиста бібліотека»

У картотеці домашньої бібліотеки зберігаються книги, які додав користувач.

Користувач заповнює бібліотеку, має змогу додати/видалити книгу, знайти книгу, провести інвентаризацію.

Книга має такі вихідні данні: автор, назва, рік написання, жанр, анотація, є або немає у наявності, кількість лайків. У залежності від жанру, книги в бібліотеці зберігаються по розділах.

* 1. Користувачі програми

У проекта є дві категорії користувачів: володар бібліотеки та гості.

Користувач проходить регістрацію або входить у свій аккаунт, заповнює бібліотеку, має змогу додати/видалити книгу, знайти книгу (за назвою чи автором), провести інвентаризацію (отримати список усіх книг, що знаходяться у бібліотеці).

Питання безпеки, такі як хешування паролів та надійність даних у програмі не вирішуються.

Гість при реєстрації обирає ввійти як гість. Він може переглядати усі книги у бібліотеці та брати читати ці книги, на улюблених екземплярах гість може поставити лайк.

Належність користувача до гостя чи володаря визначається під час входу у програму.

1.3 Функції програми

Програма надає володарю наступні можливості:

1. Реєстрація
2. Вхід у програму
3. Додавання книги
4. Видалення книги
5. Пошук книги
6. Перегляд розділів
7. Проведення інвентаризації
8. Вихід із програми

Функції гостя:

1. Вхід у програму
2. Пошук книги
3. Можливість брати книгу в борг (додаючи її в розділ «читаю зараз»)
4. Ставити лайк на улюблені книги
5. Перегляд розділів
6. Вихід із програми

Загальні функції:

1. Загрузка даних із постійної пам’яті в оперативну.
2. Збереження змін даних у постійній пам’яті.

1.4 Функції адміністратора

Функція «Вхід у програму»

Користувач входить в програму за допомогою ім’я та паролю.

Імена та паролі зберігаються в даних програми. Користувач входить при запуску програми або (якщо є бажання змінити акаунт) через поле в меню «змінити користувача».

Основний сценарій

1. При ввімкненні програми відкривається вікно входу (рис 1.4.1).
2. Користувач заповнює поля та натискає кнопку «ОК».
3. Введені дані перевіряються, та відкривається основне вікно (рис 2).

Допоміжний сценарій

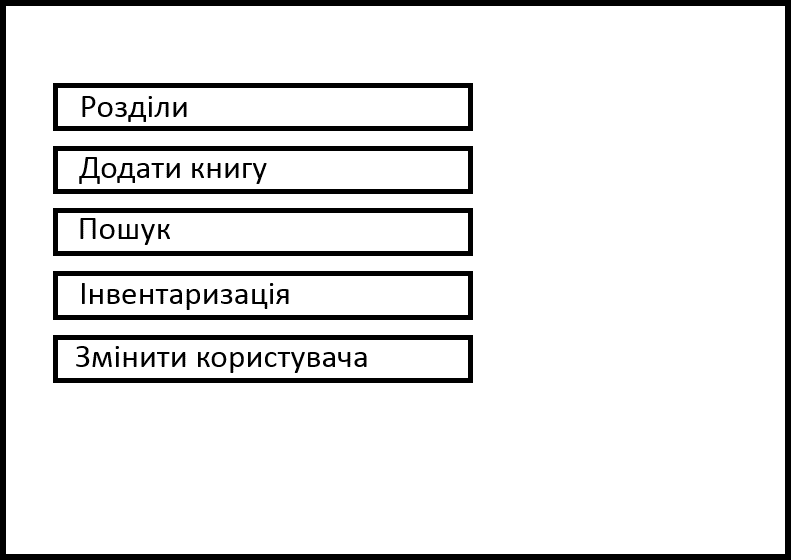
1.  Якщо дані не пройшли перевірку, на формі з’являється попередження про помилку та дозволяється спробувати ввійти ще раз.

Рис 1.4.1. - Форма входу.

Рис 1.4.2. – Головна форма.

Функція «Додавання/зміни/видалення книги»

Після входу користувач в меню обирає поле «додати нову книгу». Відкривається форма, яку він повинен заповнити та натиснути кнопку «Зберегти». Після чого дані збережуться та книга з’явиться у відповідному розділі.

Сценарій додавання книги

1. З головної форми (рис 1.4.2) користувач обирає поле «Додати нову книгу».
2. Відкривається пуста форма (рис 1.4.3).
3. Користувач заповнює вільні поля форми та натискає кнопку «Зберегти».
4. Нові дані збережуться. Якщо відшукати цю книгу в бібліотеці та відкрити її, то з’явиться вже заповнена форма.

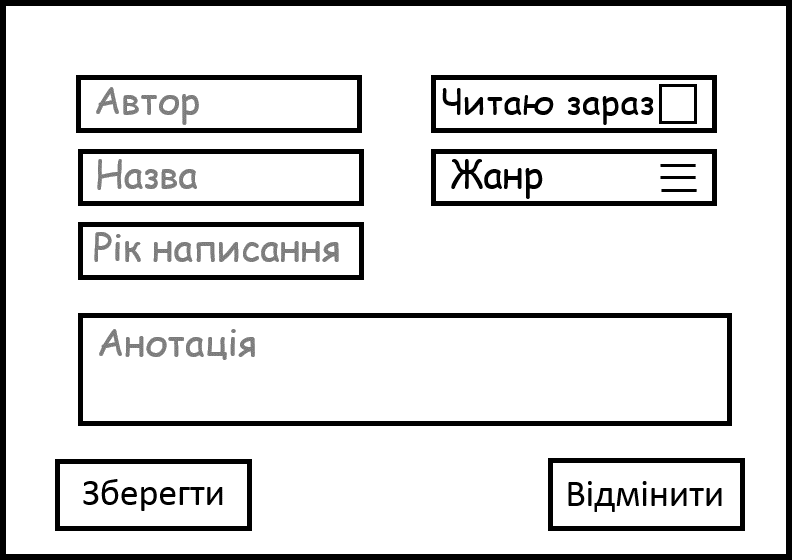


Рис 1.4.3. – Дані книги.

Сценарій зміни даних книги

1. Відшукуємо книгу за допомогою пошуку або переглядаючи розділи (рис 1.4.4).
2. Обираємо книгу, при цьому відкриється вже заповнена форма (рис 1.4.5).
3. Користувач може змінити жанр, анотацію або додати цю книгу в розділ «Читаю зараз».
4. Користувач в кінці повинен натиснути кнопку «Зберегти».

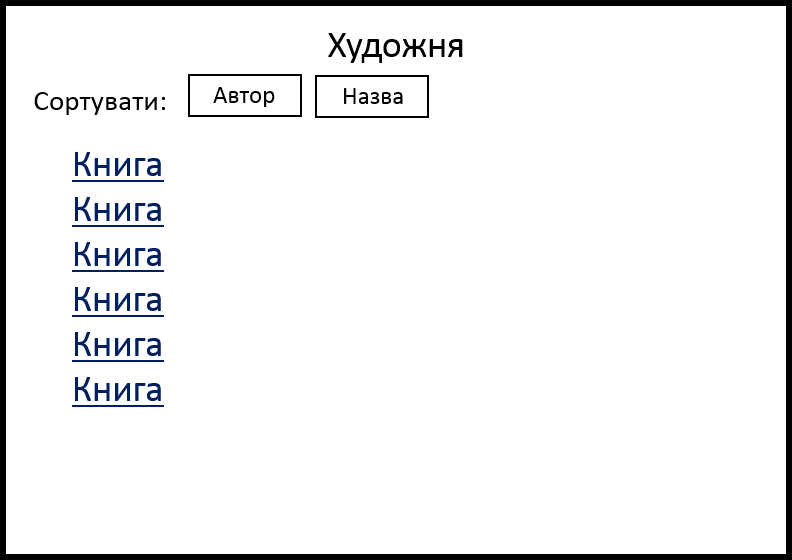


Рис 1.4.4. – Приклад розділу.

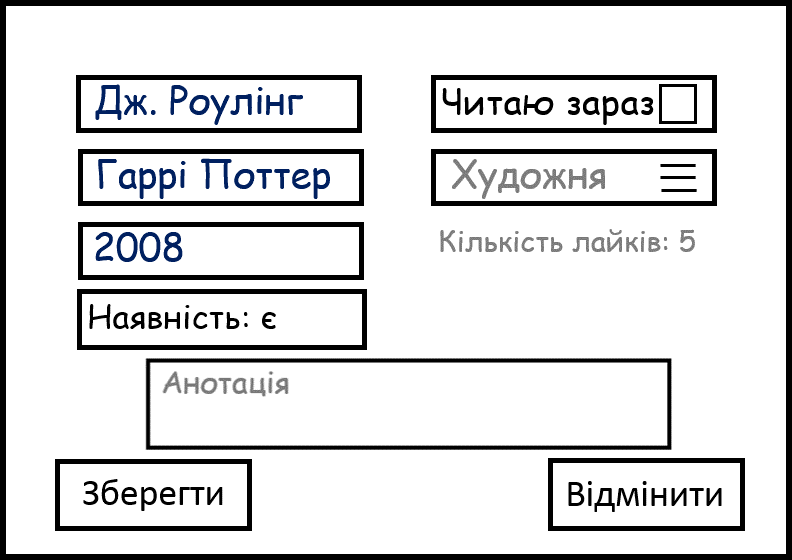


Рис 1.4.5. – Приклад заповненої форми.

Сценарій видалення книги

1. Користувач обирає книгу.
2. Натискає кнопку Delete.
3. Після отримання підтвердження книга буде видалена, список оновлюється.

Функція «Пошуку»

Функція «Пошуку» потрібна для швидкого переходу до потрібної книги. Також, таким чином можна з’ясувати чи присутня та чи інша книга у бібліотеці.

Основний сценарій «Пошуку»

1. Через головну форму (рис 1.4.2) користувач обирає поле «Пошук».
2. Відкривається форма з полем для вводу (рис 1.4.6).
3. Користувач вводить назву книги або її автора.
4. Натискає кнопку «пошук».
5. Форма оновлюється і з’являється перелік книг цього автора або книга з шуканою назвою.

Допоміжний сценарій

1. Якщо книги з шуканою назвою не існує в бібліотеці, у формі з’явиться відповідне оголошення.

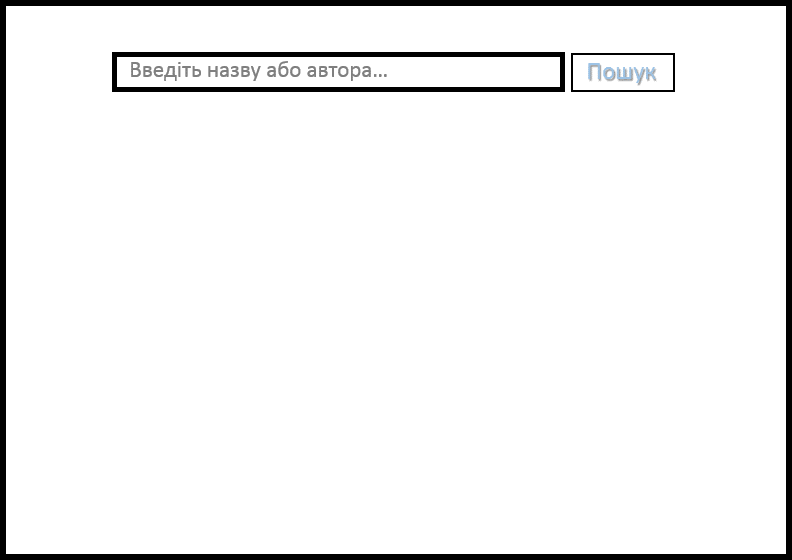


Рис 1.4.6. – Форма пошуку.

Функція «Інвентаризація»

Функція «Інвентаризації» дозволяє користувачу переглянути повний перелік книг. Це зручно для аналізу бібліотеки, користувач має змогу редагувати цей список книг, тобто реалізовувати функції додавання, зміни та видалення книг.

Сценарій проведення інвентаризації

1. Користувач з основної форми (рис 1.4.2) обирає поле «Інвентаризація».
2. Відкривається форма (рис 1.4.7), у якій є повний перелік усіх книг, що зберігаються у бібліотеці.
3. Користувач має змогу застосувати інші функції або просто переглянути свою особисту бібліотеку.

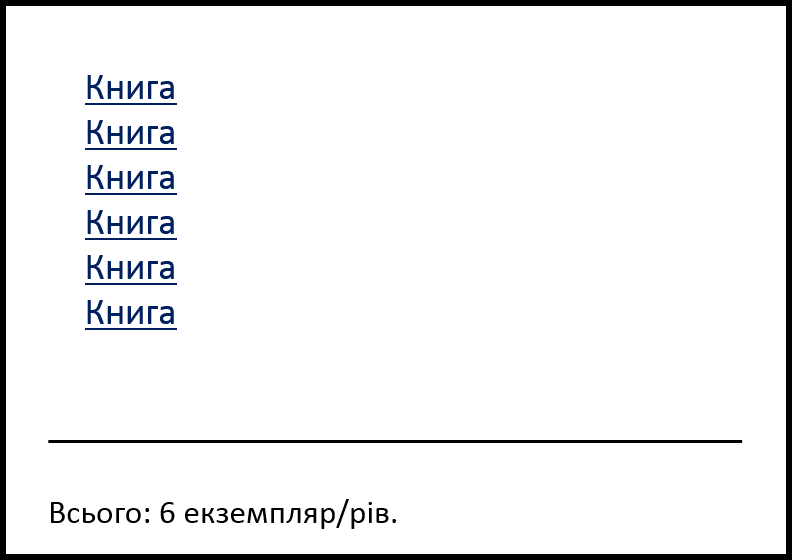


Рис 1.4.7. – Проведення інвентаризації.

Функція «Вихід із програми»

В ході роботи програми користувач змінює дані, що знаходяться в оперативній пам’яті. Тому при завершенні програми користувач отримає пропозицію зберегти зміни.

Сценарій збігається з виходом з будь-якого редактора, має стандартний вид, тому тут не наводиться.

1.5 Функції гостя

Функція «Вхід у програму»

Гість входить в програму без реєстрації, натискуючи кнопку «Ввійти як гість». Це дає йому змогу користуватися бібліотекою, але позбавляє можливості редагувати та контролювати вміст книг.

Основний сценарій

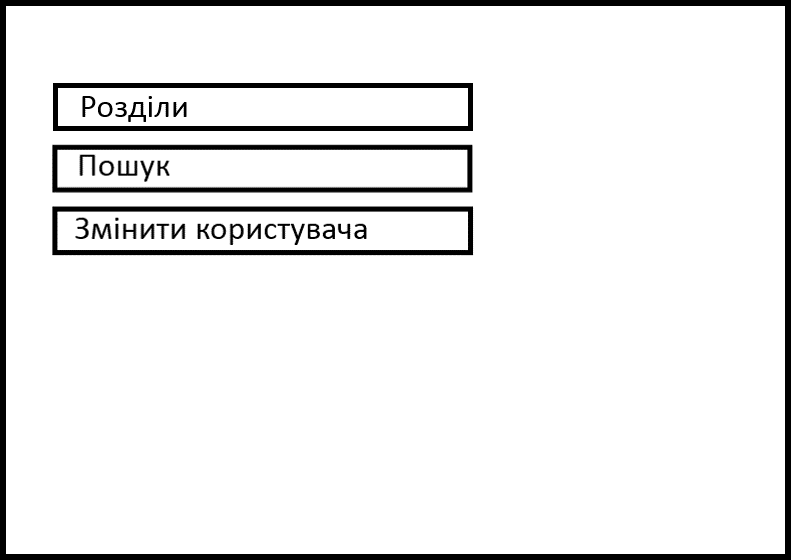
1. При ввімкненні програми відкривається вікно входу (рис 1.4.1).
2. Користувач заповнює поля та натискає кнопку «Ввійти як гість».
3. Введені дані перевіряються, та відкривається основне вікно (рис 1.5.1).

Рис 1.5.1. – Головна форма для гостя.

Функція «Пошук книги»

Функція пошуку книги збігається з такою функцією володаря бібліотеки.

Функція «Брати книгу в борг» та «ставити лайк»

Гість може взяти почитати книгу, що йому сподобалася. Для цього він знаходить книгу та додає її у розділ «читаю зараз». Таким чином книга стає автоматично недоступною для інших користувачів. Також гість може оцінити книгу, поставивши лайк.

Основний сценарій

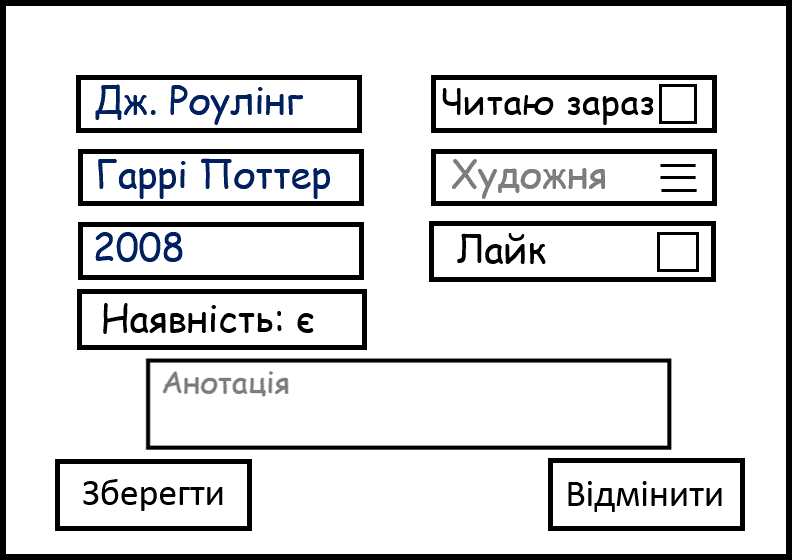
1. Відшукуємо книгу за допомогою пошуку або переглядаючи розділи (рис 1.4.4).
2. Обираємо книгу, при цьому відкриється вже заповнена форма (рис 1.5.2).
3. Гість може додати цю книгу в розділ «Читаю зараз» або поставити лайк.
4. Користувач в кінці повинен натиснути кнопку «Зберегти».

Рис 1.5.2. – Приклад заповненої форми для гостя гоиорпагостягостя.

Допоміжний сценарій

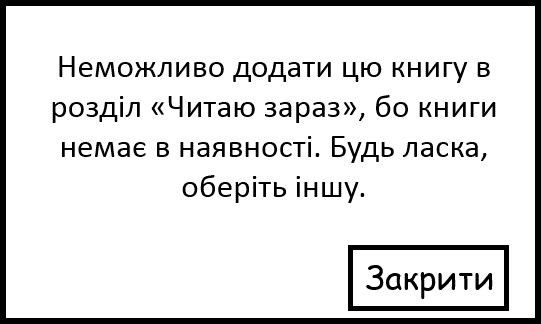
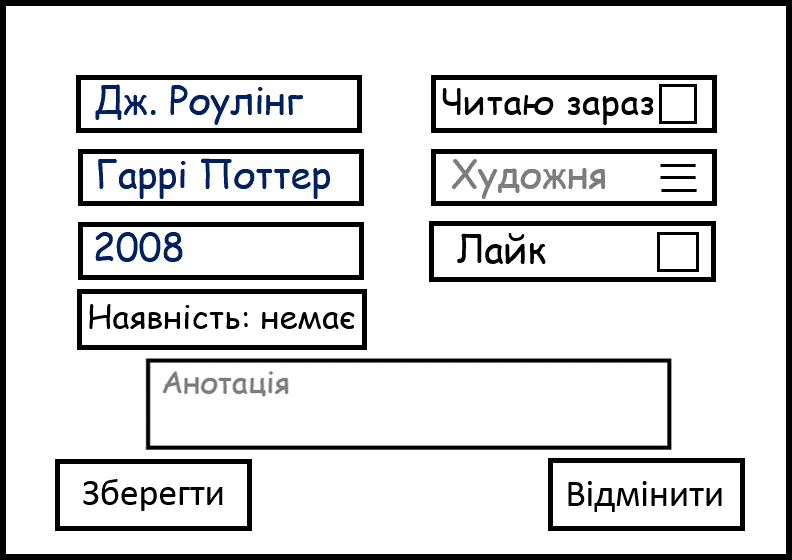
1. Якщо книги немає в наявності (рис 1.5.3), при спробі додати книгу в розділ «Читаю зараз» з’явиться застереження (рис 1.5.4), що неможливо додати цю книгу и краще обрати іншу.

Рис 1.5.4. – Застереження

Рис 1.5.3. – Приклад заповненої форми для гостя гоиорпагостягостя.

1. ПРОЕКТУВАННЯ ПРОГРАМИ

2.1 Вибір архітектури

Для реалізації проекту я обрала настільний додаток з призначеним для користувача інтерфейсом, так як це значно полегшить роботу користувача і надасть йому додаткові можливості. Так як проект замислювався для операційної системи Windows, я обрала систему керованих бібліотек Windows Forms платформи .NET Framework. Основними перевагами Windows Forms є можливість створення програми з повнофункціональним графічним інтерфейсом, великий вибір керованих елементів і стабільність роботи бібліотеки. Додаток писався на мові програмування C # в середовищі розробки Visual Studio.

2.2 Основна структура проекта

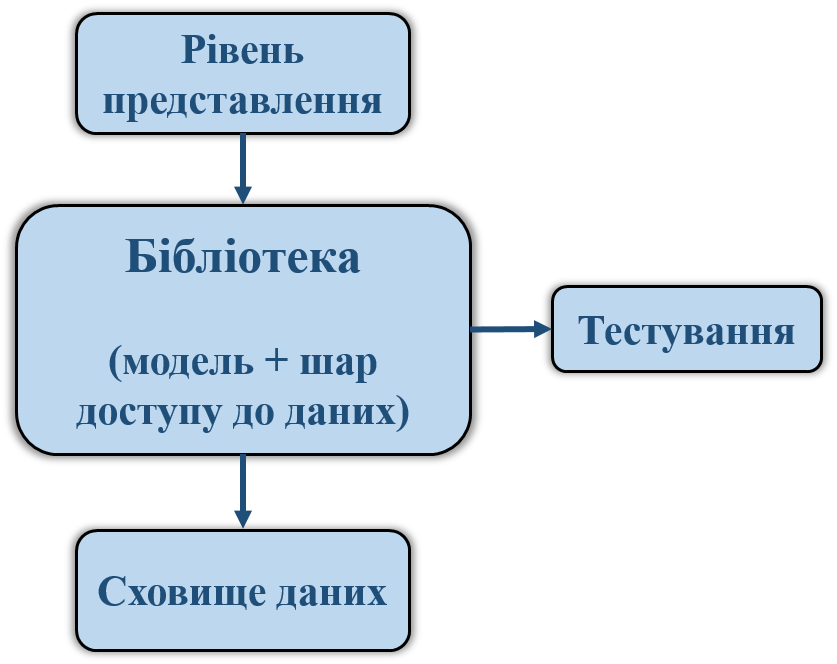
Проект курсової роботи включає в себе рівень уявлення у вигляді призначеного для користувача інтерфейсу Windows Forms, моделі проекту і шару доступу даних у вигляді бібліотеки, а також сховища даних.

В додатку існують два типи користувачів: адміністратори бібліотеки і гості. Так як дії в додатку у них схожі (гості мають деякі обмеження в порівнянні з адміністраторами), для рівня уявлення для обох користувачів було створено один додаток.

Модель проекту являє собою повну реалізацію функціоналу програми. Основним класом є клас Library, який зберігає колекції об'єктів інших допоміжних класів, таких як class Admin, class Guest, class Book, class Author, class Genre. Також class Library містить всі методи управління і зберігання даних.

Модель і шар доступу до даних знаходяться в окремій бібліотеці.

Також в проекті міститься додаток системи тестування для зручної перевірки справності програми.



Діаграма 2.2.1. – Основна структура проекту.

2.3 Об'єктна модель

Головний клас Library, який містить колекції адміністраторів, гостей, книг, авторів і жанрів, а також може керувати їх даними. Класи об'єднує відношення асоціації.

Оскільки класи Admin і Guest містять однакові властивості, був створений абстрактний клас User, який успадковують вище перераховані класи.

Адміністратор додає або видаляє книги, проводить інвентаризацію.

Гість проходить реєстрацію або авторизацію, бере читати книги, може ставити лайки, шукати книгу за допомогою пошуку або по секціях.

Адміністратор також має всі права гостя.

Про користувача відомо: ім'я, логін, пароль, унікальний ідентифікатор. Про книгу відомо: назва, автор, жанр, рік видання, анотація, унікальний ідентифікатор, унікальний ідентифікатор користувача, який її читає, колекція лайкнувших дану книгу користувачів. Про автора і жанр відомо: ім'я, унікальний ідентифікатор.

Бібліотека - кілька колекцій (адміністратори, гості, книги, автори, жанри).

Секція - колекція книг певного жанру.

Інвентаризація - виведення колекції книг в залежності від типу сортування і також виведення загальної кількості книг в бібліотеці.

Користувач = ім'я + логін + пароль + Id.

Книга = ім'я + автор + жанр + рік + анотація + Id + Id користувача + колекція Id лайкнувших користувачів.

Секція - колекція книг.

Інвентаризація - колекція книг + загальна кількість книг.

Виходячи з вище перерахованих функцій користувачів бібліотека включає в себе реалізацію відповідних методів. Основні методи бібліотеки:

1. Реєстрація (ім'я, логін, пароль, тип користувача).

2. Авторизація (логін, пароль).

3. Додавання книги (назва, автор, жанр, рік, анотація).

4. Видалення книги (Id книги).

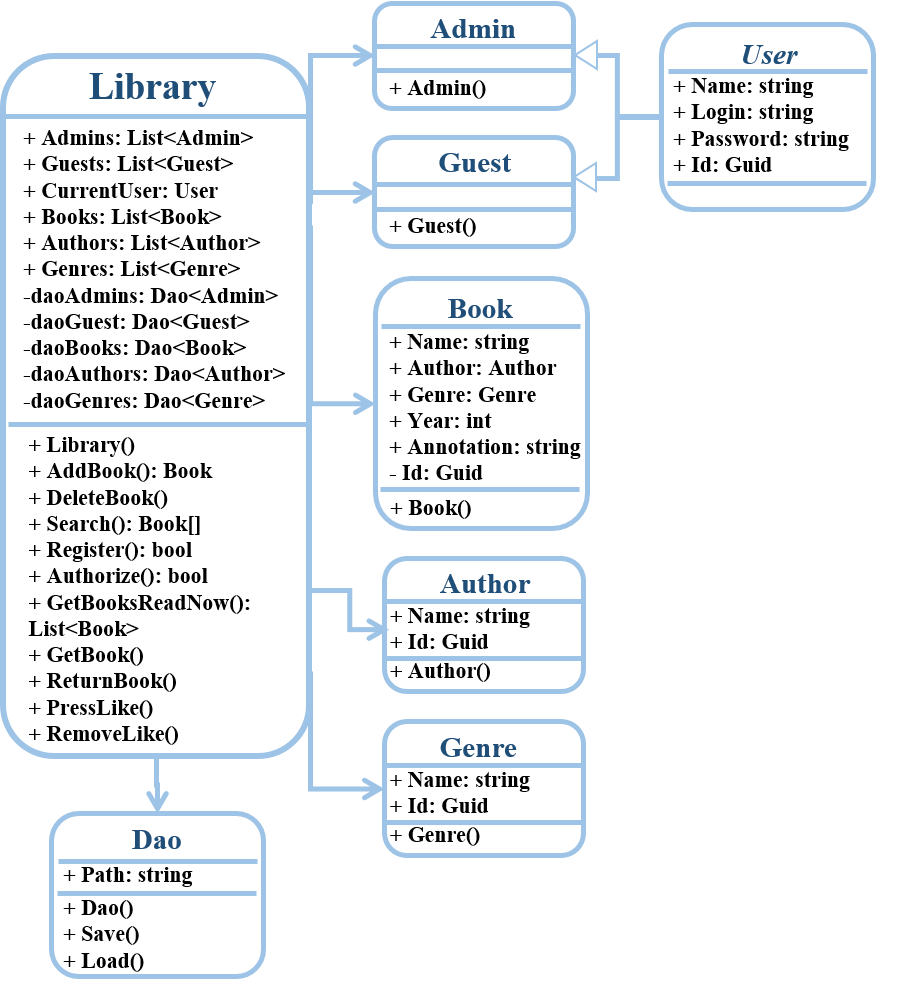
5. Пошук (введений користувачем рядок символів).

6. Взяття книги у бібліотеці (книга).

7. Повернення книги в бібліотеку (книга).

8. Додавання / видалення лайка (книга).

Також бібліотека зберігає колекції і завантажує дані, створюючи нову бібліотеку.



Діаграма 2.3.1. – Зображення відношень між класами.

2.4 Сховище даних та доступ до нього

Для зберігання і зміни даних я обрала бінарну серіалізацію - процес перетворення будь-якого об'єкта в потік байтів.

Всі колекції класів, які знаходяться в бібліотеці будуть зберігатися в окремих бінарних файлах. Серіалізація була написана в окремому класі Dao за допомогою класу BinaryFormatter.

Захист конфіденційності даних в роботі не розглядається.

1. ІНСТРУКЦІЯ КОРИСТУВАЧЕВІ

Установка додатку

Перейдіть за посиланням https://github.com/Daria-Koshkina/Library .

Завантажте архів з репозиторієм.

Запустіть файл «LibraryApp.exe» за таким шляхом: Library/LibraryApp/bin/Release/LibraryApp.exe .

* 1. Інструкція адміністратору

Авторизація

Основний сценарій

1. Запустіть програму, після чого відкриється вікно авторизації (рис 3.1.1).
2. Введіть логін та пароль.
3. Натисніть кнопку «Log in». Після чого відкриється головна форма адміністратора.

Допоміжний сценарій

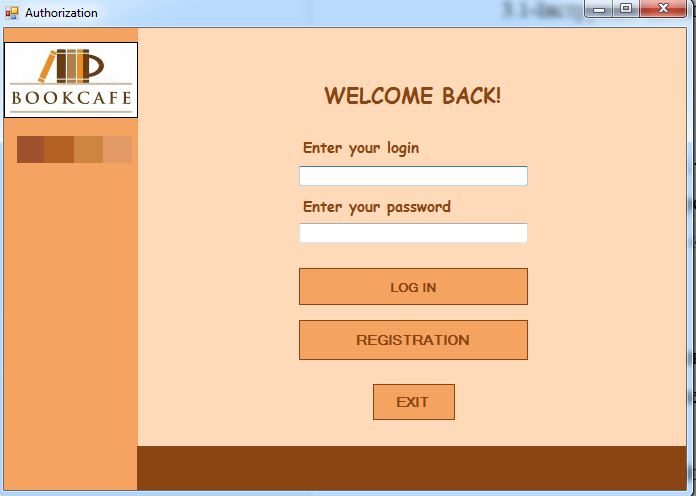
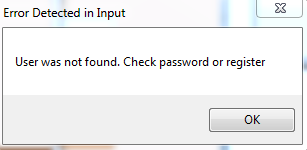
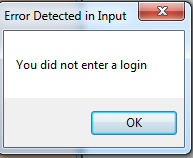
1. Якщо Ви ввели невірні дані, або намагаєтесь ввійти без реєстрації, з’явиться повідомлення про помилку з подальшими рекомендаціями (рис 3.1.2).
2. Якщо Ви хочете вийти, треба натиснути кнопку «Exit».
3. Якщо Ви хочете зареєструватися, натисніть кнопку «Registration».

Рис 3.1.2 – Приклад помилок.

Рис 3.1.1 – Форма авторизації.

Реєстрація

Основний сценарій

1. Натисніть кнопку Registration» на формі авторизації (рис 3.1.3).
2. Введіть своє ім’я. логін та пароль.
3. Оберіть роль адміністратора.
4. Натисніть кнопку «Register».

Допоміжний сценарій

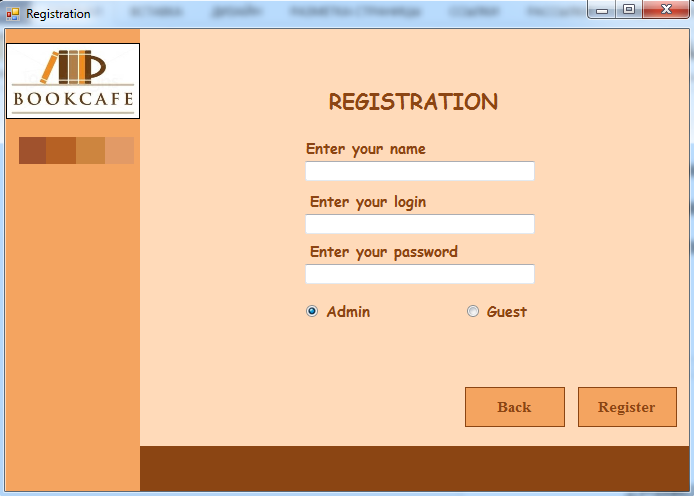
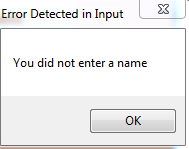
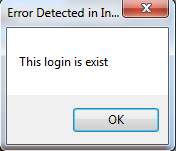
1. Якщо Ви ввели не всі дані, або такий користувач вже є в бібліотеці, з’явиться повідомлення про помилку з подальшими рекомендаціями (рис 3.1.4).
2. Якщо Ви хочете повернутися до авторизації або вийти, натисніть «Back».

Рис 3.1.4 – Приклад помилок.

Рис 3.1.3 – Форма реєстрації.

Робота з програмою

1. Після реєстрації або авторизації ви опиняєтесь на головній формі (рис 3.1.5).
2. Для швидкого орієнтування в додатку або якщо виникли питання, натисніть кнопку «Help», після чого відкриється форма з поясненням (рис 3.1.6).
3. Якщо бібліотека порожня, натисніть кнопку «Library Autocomplete», щоб наповнити бібліотеку тестовими даними. При кожному натисканні кнопки «Library Autocomplete» у бібліотеку додається встановлений перелік книг.

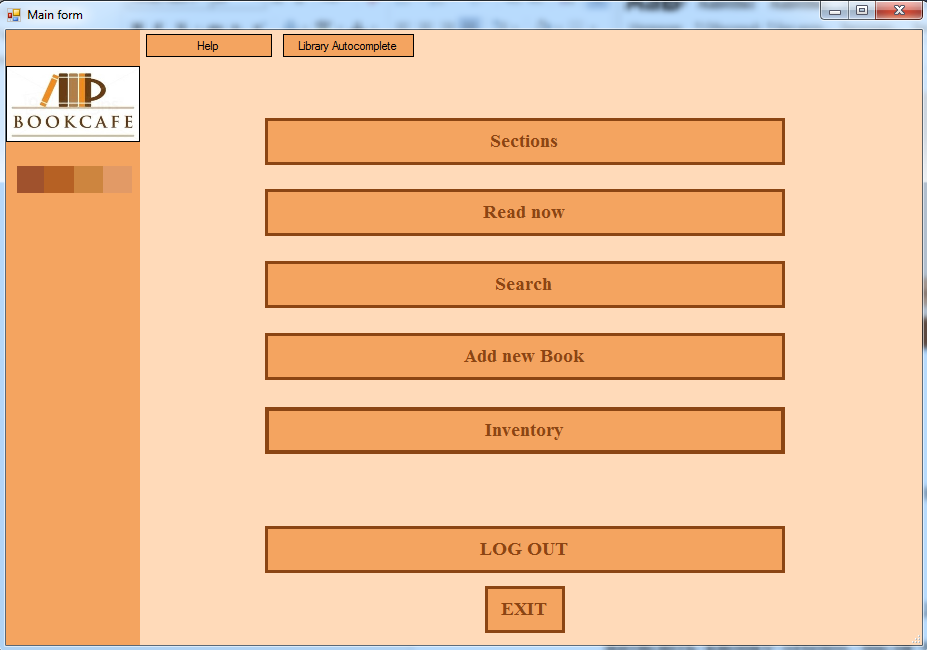
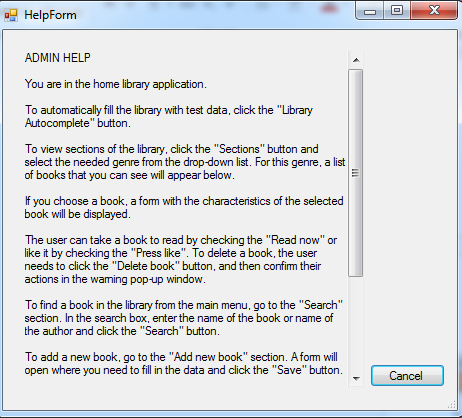


Рис 3.1.6 – Форма допомоги.

Рис 3.1.5 – Головна форма.

1. Для додавання нової книги у бібліотеки натисніть кнопку «Add new book», після чого відкриється форма заповнення даних книги (рис 3.1.7).

Заповніть поля – введіть фамілію автора, назву книги, рік видання, виберіть жанр зі списку або введіть свій, за бажанням додайте анотацію.

Натисніть кнопку «Save».

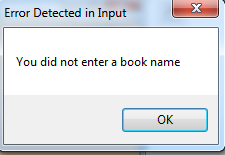
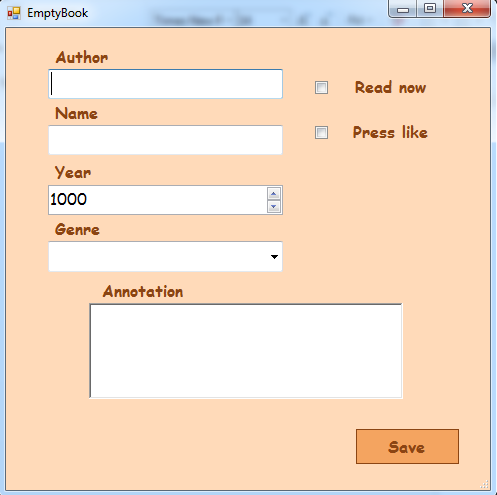
При помилці з’явиться вікно (рис 3.1.8) з описом помилок та рекомендаціями щодо їх виправлення.

Рис 3.1.7 – Форма для заповнення даних книги.

Рис 3.1.8 – Приклад помилок.

1. Для проведення інвентаризації, натисніть на головній формі кнопку «Inventory», після чого відкриється форма зі списком книг (рис 3.9). Зверху оберіть як відсортувати книги. При обранні сортування за автором чи жанром, з’явиться список відповідних структур. В цьому списку Ви можете обрати автора чи жанр та переглянути належні йому книги. За умовою обрано сортування за автором. Знизу форми Ви побачите загальну кількість книг в бібліотеці. Інвентаризація проводиться адміністратором для зручної та швидкої оцінки стану бібліотеки, а також для її редагування.

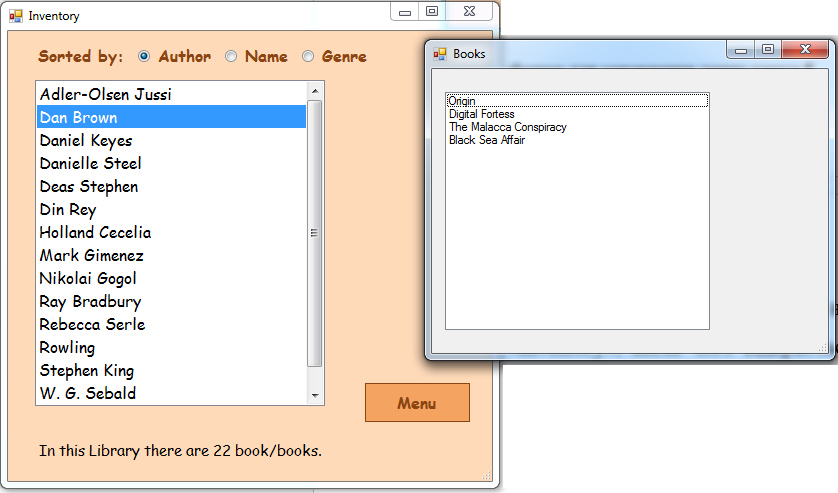


Рис 3.1.9 – Форма інвентаризації.

1. Для зміни акаунту оберіть в меню кнопку «Log out», після чого відкриється вікно авторизації.
2. Для виходу з додатку натисніть кнопку «Exit» на головній формі.

Також адміністратору доступні усі функції гостя: перегляд книг по секціях, пошук по ключових словах. обирання книг для читання, ставлення лайків (див. підрозділ 3.2).

* 1. Інструкція гостю

Авторизація та реєстрація

Авторизація та реєстрація гостя проходить аналогічно діям адміністратора (див. підрозділ 3.1). При реєстрації у пункті обрання ролі виберіть роль гостя.

Робота з програмою

1. Після реєстрації або авторизації ви опиняєтесь на головній формі (рис 3.1.5).
2. Для швидкого орієнтування в додатку або якщо виникли питання, натисніть кнопку «Help», після чого відкриється форма з поясненням (рис 3.1.6).
3. Для перегляду книг по розділах натисніть кнопку «Sections» на головній формі, після чого відкриється форма з жанрами (рис 3.2.1).

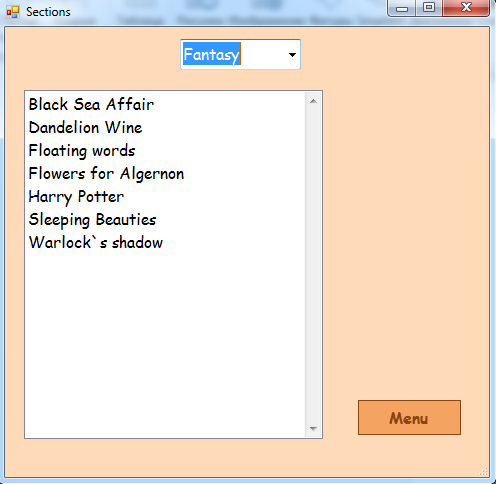
У випадному списку оберіть жанр, після чого на формі з’явиться перелік відповідних книг. Таким чином ви можете обирати книгу для читання з цього списку.

Рис 3.2.1 – Форма з жанрами.

1. Для перегляду списку книг, що ви взяли для читання натисніть кнопку «Read now» на головній формі, після чого з’явиться форма з відповідним списком (рис 3.2.2).

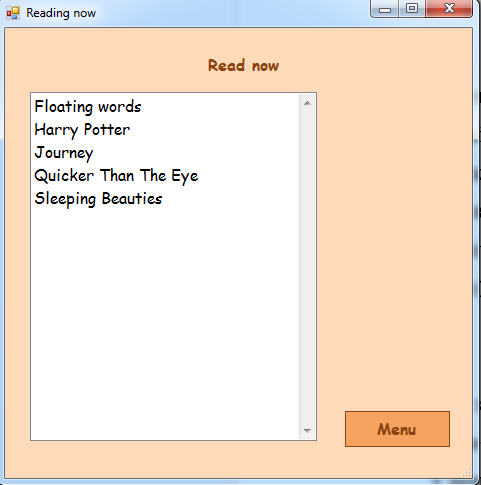
Для видалення книги з цього списку (повернення книги до бібліотеки) оберіть зі списку книгу та зніміть галочку з пункту «Read now» (читаю зараз).

Рис 3.2.2 – Форма з переліком взятих для читання книг.

1. Для зручного пошуку книг оберіть на головній формі пункт «Search», після чого відкриється форма пошуку (рис 3.2.3).

У пошуковій стрічці введіть назву книги, автора або ключове слово.

Натисніть кнопку «search».

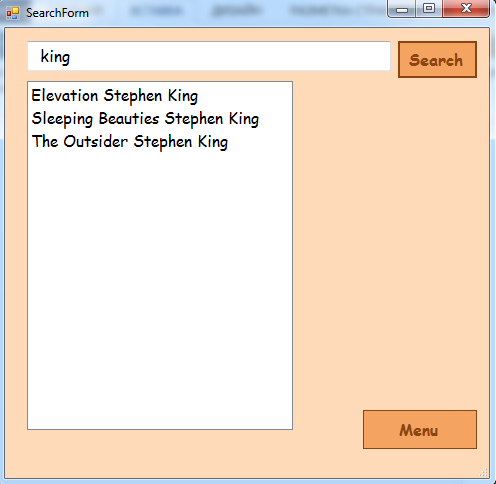
На формі з’явиться список книг з відповідними ключовими словами. У разі відсутності в бібліотеці шуканої книги або автора список буде порожній.

Рис 3.2.3 – Форма пошуку.

1. Для зміни акаунту оберіть в меню кнопку «Log out», після чого відкриється вікно авторизації.
2. Для виходу з додатку натисніть кнопку «Exit» на головній формі.

ВИСНОВКИ

У ході виконання курсового проекту були засвоєні навички об’єктно-орієнтовного програмування мовою C#, здобуті знання з організації логіки додатку, роботи з користувачем, поглиблені навички роботи з Windows Forms.

Був створений додаток «Особиста бібліотека», що відповідає меті роботи та специфікації проекту. Дана програма дозволяє користувачу переглядати книги та брати їх для читання. Повний набір функцій дозволен адміністраторам. Ці користувачі можуть додавати та видаляти книги, проводити інвентаризацію бібліотеки. Усі функції звичайних користувачів – гостей доступні адміністраторам. Бібліотека також має функції для поліпшення користування програмою, такі як пошук, наявність розділів і т. п. Через те, що вміст бібліотеки зберігається у бінарних файлах, програма захищена від редагування та пошкодження файлів. Якщо деякі файли зазнали змін, додаток відновить максимальну кількість інформації та буде працювати коректно.

Програма написана на мові програмування C# на платформі .NET Framework з використанням технологій Windows Forms у середі розробки Microsoft Visual Studio 2019.

Програма може бути поліпшена шляхом застосування бази даних, яка може синхронізуватися по мережі.

Галуззю застосування може стати будь-яка справжня бібліотека, яка хоче випустити свій електронний додаток для полегшення зберігання і перевірки даних, а також поява можливості електронного бронювання книг для залучення клієнтів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Троелсен, Э, Джепикс, Ф. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд., 2018.
2. Шилдт Г. C# 4.0.: Полное руководство : пер. с англ.- М.: Вильямс, 2011.
3. Бондарев, В.М. Объектно-ориентированное программирование на С# : учеб. пособ. / – Х. : Компания СМИТ, 2009.
4. Полное руководство по языку программирования С# 7.0 и платформе .NET 4.7 [Електронний ресурс] – Режим доступу до ресурсу: https://metanit.com/sharp/tutorial/.
5. Microsoft Docs, документация Microsoft для конечных пользователей, разработчиков и ИТ- специалистов : руководство по Windows Forms [Електронний ресурс] – Режим доступу до ресурсу: . https://docs.microsoft.com/en-us/dotnet/framework/winforms/

ДОДАТОК А

Посилання на репозиторій:

https://github.com/Daria-Koshkina/Library