

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та безпеки

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**ЕЛЕКТРОНА БІБЛІОТЕКА НА PYTHON ДЛЯ IOS ТА
ANDROID**

ELECTRONIC LIBRARY IN PYTHON FOR IOS AND ANDROID

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-42

Єрмакова Софія Русланівна

(підпис)

Керівник:

ст.викладач

Міскевич Оксана Іванівна

(підпис)

Кваліфікаційну роботу

допущено до захисту

« 04 » червня 2025 р.

Гарант освітньої програми:

к.т.н., доцент

Лавренчук Світлана Василівна

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Тарас ТЕРЛЕЦЬКИЙ

« 10 » 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Єрмакова Софія Русланівна

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Електронна бібліотека на Python для iOS та Android*

Керівник роботи *ст. викладач Міскевич Оксана Іванівна*

затверджені наказом закладу вищої освіти від «04» січня 2025 року № 11/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 10.06.2025р.

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Вступ

Аналіз сучасних підходів до створення мобільних електронних бібліотек та постановка завдань розробки мобільного застосунку

Теоретична частина

Практична частина

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Огляд існуючих мобільних електронних бібліотек і аналіз їхніх функціональних можливостей

Використані технології

Архітектура системи

Інтерфейс системи

Схема роботи програмного продукту

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз сучасних підходів до створення мобільних електронних бібліотек та постановка цілей і завдань розробки</i>	<i>Міскевич О. І., ст. викл.</i>		
<i>Теоретичні засади реалізації мобільних додатків електронних бібліотек та технічна реалізація проєкту</i>	<i>Міскевич О. І., ст. викл.</i>		
<i>Створення та тестування мобільного застосунку електронної бібліотеки з використанням Python та Kivu</i>	<i>Міскевич О. І., ст. викл.</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>		___%	
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст.викладач</i>		

7. Дата видачі завдання 10.01.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Розділ 1. Аналіз підходів до створення мобільних електронних бібліотек</i>	до 10.02.2025 р.	Виконано
2.	<i>Розділ 2. Теоретичні засади розробки застосунку на базі Kivu</i>	до 02.03.2025 р.	Виконано
3.	<i>Розділ 3. Реалізація мобільного застосунку електронної бібліотеки</i>	до 02.04.2025 р.	Виконано
4.	<i>Висновки та пропозиції</i>	до 10.04.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 15.04.2025 р.	Виконано
6.	<i>Формування додатків</i>	до 02.05.2025 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 10.05.2025 р.	Виконано
8.	<i>Нормоконтроль</i>	до 30.05.2025 р.	Виконано
9.	<i>Інструментальна перевірка на академічний плагіат</i>	до 03.06.2025 р.	Виконано
10.	<i>Представлення кваліфікаційної роботи бакалавра до захисту</i>	до 10.06.2025 р.	Виконано

Здобувач вищої освіти

_____ (підпис)

Єрмакова С. Р.

_____ (прізвище, ініціали)

Керівник кваліфікаційної роботи

_____ (підпис)

Міскевич О. І.

_____ (прізвище, ініціали)

АНОТАЦІЯ

Єрмакова С. Р. Електронна бібліотека на Python для iOS та Android.

Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

Дослідження включає аналіз сучасних підходів до створення мобільних застосунків, зокрема кросплатформних рішень на основі Python і фреймворку Kivy. У першому розділі висвітлено актуальність теми, можливості мови Python для мобільної розробки, особливості електронних бібліотек та вимоги до користувацького інтерфейсу.

Другий розділ присвячено проектуванню архітектури мобільного застосунку, структури даних, логіки роботи системи та дизайну інтерфейсу. Розглянуто реалізацію основних функцій електронної бібліотеки: каталог книг, пошук, персональна бібліотека, авторизація користувачів.

У третьому розділі подано практичну реалізацію застосунку за допомогою Python і Kivy, з локальною базою даних SQLite, підтримкою сканування книг і пошуку, а також тестуванням кросплатформної роботи на Android і iOS. Проведено аналіз ефективності функціоналу та визначено перспективи розвитку.

Ключові слова: електронна бібліотека, мобільний застосунок, Python, Kivy, iOS, Android, кросплатформна розробка, цифрові книги.

ANNOTATION

Ermakova S. R. Electronic library in Python for iOS and Android. Manuscript. Bachelor qualification work of the educational program "Computer Engineering", specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

The qualification work consists of an introduction, three chapters, conclusions, a list of references, and appendices.

The study explores modern approaches to mobile application development, with a focus on cross-platform solutions based on Python and the Kivy framework. The first chapter outlines the relevance of the topic, the capabilities of Python for mobile development, the specifics of electronic libraries, and user interface requirements.

The second chapter focuses on designing the architecture of the mobile application, data structure, system logic, and user interface. It covers the implementation of core features of the e-library: book catalog, search, personal library, and user authentication.

The third chapter presents the practical implementation of the application using Python and Kivy, with a local SQLite database, book scanning and search capabilities, and testing on Android and iOS platforms. The functionality is evaluated, and potential improvements are discussed.

Keywords: e-library, mobile application, Python, Kivy, iOS, Android, cross-platform development, digital books.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНОЛОГІЙ ТА ОСОБЛИВОСТЕЙ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ.....	10
1.1 Мобільні додатки та їх значення в цифрову епоху	10
1.2 Огляд підходів і фреймворків кросплатформеної розробки	10
1.3 Python для мобільної розробки	11
1.4 Електронні бібліотеки як приклад цифрових інформаційних систем.....	12
1.5 Аналіз вимог до інтерфейсу користувача в мобільному додатку електронної бібліотеки	13
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ.....	15
2.1 Постановка задачі	15
2.2 Архітектура мобільного додатку.....	16
2.3 Інтерфейс користувача	17
2.4 Архітектура та логіка роботи додатку	18
2.5 Порівняння з аналогами електронних бібліотек.....	20
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ.....	22
3.1 Ініціалізація проєкту та структура файлів.....	22
3.2 Розробка локальної бази даних.....	24
3.3 Реалізація авторизації та реєстрації	25
3.4 Каталог книг та інтерфейс перегляду	29
3.5 Персональна бібліотека користувача.....	33
3.6 Пошук та фільтрація та сканування книг	35
3.7 Розширені функціональні можливості.....	38
3.8 Кросплатформенність програмного засобу.....	40
ВИСНОВКИ.....	42
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	44

ВСТУП

Електронні бібліотеки набули значної популярності в останні десятиліття. Вони забезпечують швидкий та зручний доступ до великої кількості книг, дозволяють ефективно організовувати збереження інформації та відкривають нові можливості для навчання і досліджень. У сучасних реаліях користувачі очікують від бібліотечних систем не лише функціональності, а й зручного та інтуїтивно зрозумілого інтерфейсу. Для створення таких систем активно використовуються різні програмні засоби, серед яких важливе місце займають кросплатформні мобільні застосунки.

Одним із найбільш підходящих інструментів для розробки мобільних застосунків є Python. Завдяки великому вибору бібліотек та фреймворків, ця мова програмування дозволяє реалізувати як прості, так і складні мобільні рішення. Фреймворк Kivy є одним із найкращих варіантів для створення мобільних додатків, оскільки підтримує кросплатформність, має вбудовані елементи інтерфейсу та дозволяє реалізувати інтерактивні функції для користувачів.

Метою кваліфікаційної роботи є розробка кросплатформного мобільного застосунку електронної бібліотеки з використанням Python та Kivy, який дозволить користувачам зручно шукати, читати та організовувати електронні книги.

Для досягнення цієї мети в кваліфікаційній роботі будуть виконані наступні завдання:

- аналіз технологій для розробки мобільних застосунків на Python;
- вивчення можливостей та особливостей фреймворку Kivy;
- розробка архітектури мобільного додатка;
- реалізація основних функцій електронної бібліотеки, включаючи каталогізацію, пошук та управління книгами;
- оптимізація додатка та його тестування на мобільних пристроях;
- оцінка ефективності реалізованого рішення та визначення можливих напрямів його вдосконалення.

Кваліфікаційна робота складається з двох основних частин: теоретичної та практичної.

У теоретичній частині буде розглянуто сучасні технології для розробки мобільних додатків, особливості створення електронних бібліотек та можливості використання Python для цієї задачі.

У практичній частині буде реалізовано мобільний додаток, проаналізовано його ефективність та проведено тестування.

Результатом роботи стане функціональний мобільний застосунок, який може бути використаний для зручного збереження та читання електронних книг.

РОЗДІЛ 1

АНАЛІЗ ТЕХНОЛОГІЙ ТА ОСОБЛИВОСТЕЙ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ

1.1 Мобільні додатки та їх значення в цифрову епоху

Мобільні додатки стали невід'ємною частиною цифрової інфраструктури сучасного суспільства. Завдяки широкому розповсюдженню смартфонів та планшетів, більшість користувачів взаємодіють з інформаційними системами саме через мобільні пристрої. Мобільні застосунки охоплюють практично всі сфери життя – від освіти та охорони здоров'я до фінансів, торгівлі та розваг.

Особливу популярність здобули мобільні додатки, що дозволяють працювати з інформацією в зручному форматі та забезпечують персоналізовану взаємодію з користувачем. В умовах розвитку інформаційного суспільства саме мобільні застосунки стали одним із ключових інструментів доступу до цифрових бібліотек, що дозволяє швидко знаходити, читати та зберігати літературу без прив'язки до конкретного місця чи пристрою.

1.2 Огляд підходів і фреймворків кросплатформеної розробки

Одним із ключових викликів у мобільній розробці є забезпечення підтримки різних платформ – переважно Android та iOS. Створення окремих нативних застосунків для кожної платформи потребує подвійної роботи, що призводить до зростання витрат часу та ресурсів. Тому кросплатформна розробка, яка дозволяє писати один код для обох платформ, є актуальним підходом у сучасному розробництві [1].

Далі розглянемо найпоширеніші технології кросплатформної розробки є.

Flutter – фреймворк від Google, що використовує мову Dart. Забезпечує високопродуктивні інтерфейси та доступ до нативних функцій через власний рушій. Flutter має широку спільноту та активно розвивається, однак його навчання може бути складнішим через нову мову програмування [2].

React Native – бібліотека від Facebook, що базується на JavaScript та дозволяє створювати інтерфейси, схожі на нативні. React Native активно використовується у бізнес-середовищі, має велику кількість плагінів та підтримку спільноти [3]. Проте продуктивність може бути нижчою порівняно з нативною розробкою.

Kivy – фреймворк на Python з підтримкою OpenGL ES2, який дозволяє створювати кросплатформні додатки з єдиним кодом для Android, iOS, Windows, Linux та macOS. Kivy відзначається простотою синтаксису, можливістю швидкого створення прототипів, але поступається Flutter та React Native у продуктивності та естетичності інтерфейсів.

Вибір технології залежить від завдань, досвіду розробника та вимог до додатку [4]. У даній роботі обрано Kivy як найбільш придатний для швидкої реалізації мобільного застосунку на основі Python [5]. Kivy забезпечує базову кросплатформну функціональність і підтримує основні компоненти інтерфейсу [6].

1.3 Python для мобільної розробки

Python відомий своєю простотою, лаконічністю та широкою екосистемою бібліотек [7]. Це робить його однією з найпопулярніших мов програмування у світі [8]. Хоча він традиційно використовується для бекенд-розробки, наукових обчислень та автоматизації, останнім часом Python набуває все більшого поширення і в сфері мобільної розробки [9]. Як зазначає Д. Завацький, Python дає змогу швидко створювати адаптивні мобільні додатки навіть невеликими командами [10].

Серед основних переваг Python, які зумовлюють його популярність у розробці програмного забезпечення, варто зазначити легкість у вивченні, що робить мову доступною навіть для початківців, велику кількість бібліотек і модулів для роботи з базами даних, мережевими протоколами, графікою та

іншими напрямками, а також активну спільноту розробників і регулярне оновлення бібліотек і документації.

Основним фреймворком, що дозволяє створювати мобільні додатки на Python, є Kivy [11]. Це бібліотека з відкритим кодом, орієнтована на розробку мультитач-застосунків з багатим інтерфейсом. Вона підтримує запуск на Android, iOS, Windows, Linux та macOS. Однією з головних переваг Kivy є можливість писати універсальний код без потреби вносити зміни під конкретну платформу [12].

Для розробки мобільних застосунків на основі Kivy використовуються також такі інструменти як Buildozer – утиліта для збирання Android-пакетів (APK) з Python-коду; Plyer – бібліотека, що надає доступ до нативних функцій пристрою, таких як камера, GPS, повідомлення; Kivymd – бібліотека UI-компонентів у стилі Material Design; SQLite / SQLAlchemy – рішення для зберігання даних локально на пристрої.

Використання Python дає змогу швидко створювати прототипи, навчальні проєкти та навіть повноцінні застосунки для невеликих команд або індивідуальних розробників. Це робить його привабливим вибором у випадках, коли важлива швидкість розробки та кросплатформність.

1.4 Електронні бібліотеки як приклад цифрових інформаційних систем

Електронні бібліотеки – це інформаційні системи, що забезпечують збереження, пошук, перегляд і завантаження цифрових версій текстових ресурсів [13]. Вони можуть включати в себе книги, статті, енциклопедії, технічну документацію та інші форми текстових даних.

Сучасні електронні бібліотеки повинні мати зручний та інтуїтивно зрозумілий інтерфейс користувача, підтримувати багаторівневу систему пошуку з можливістю фільтрації за різними критеріями, забезпечувати авторизацію, персоналізацію та збереження налаштувань, надавати доступ до читання як у

режимі онлайн, так і офлайн, бути сумісними з популярними форматами документів, такими як PDF, ePub і txt, а також мати масштабовану архітектуру з належним захистом персональних даних.

Існують різноманітні приклади реалізації подібних систем – від академічних платформ на кшталт JSTOR чи Springer до мобільних додатків типу Google Play Books або FBReader [14]. Незалежно від розміру, структура таких систем базується на каталогах, системах метаданих, механізмах доступу до контенту та користувацькому досвіді (UX).

Розробка власної електронної бібліотеки у вигляді мобільного додатку дає можливість зосередитись на потребах конкретної аудиторії, розробити інтуїтивну навігацію, інтегрувати індивідуальні функції та досягнути автономної роботи без постійного з'єднання з сервером.

1.5 Аналіз вимог до інтерфейсу користувача в мобільному додатку електронної бібліотеки

Інтерфейс користувача (UI) відіграє визначальну роль у забезпеченні ефективної взаємодії користувача з мобільним додатком. У випадку електронної бібліотеки, де головним завданням є надання швидкого та зручного доступу до цифрових книг, саме якість інтерфейсу значною мірою визначає загальне враження від користування системою (UX). Важливими аспектами розробки такого інтерфейсу є простота навігації, що передбачає інтуїтивне розташування меню, розділів і кнопок для полегшення пошуку книг без потреби у виконанні зайвих дій. Також суттєвим є забезпечення адаптивності, оскільки інтерфейс має коректно відобразитись як на екранах смартфонів, так і планшетів. Візуальна ієрархія компонентів, включаючи назви книг, авторів, обкладинки та розділи, повинна бути логічною та зрозумілою для користувача. Значну увагу необхідно приділити зручності читання шляхом правильного підбору шрифтів, міжрядкових інтервалів, кольорів, а також впровадження можливості перемикання між темною і світлою темами та збереження місця читання. Крім

того, важливими є функції авторизації та персоналізації, що дозволяють зберігати особисті налаштування користувача, історію читання та забезпечувати синхронізацію між різними пристроями.

Для реалізації інтерфейсу користувача у Kivu використовується підтримка .kv файлів, що забезпечує розділення логіки та візуального представлення, а також бібліотека KivuMD, яка дозволяє оформити додаток у стилі Material Design. У першому розділі кваліфікаційної роботи було здійснено огляд сучасних підходів до розробки мобільних додатків із акцентом на кросплатформенні рішення, що дають змогу досягти ширшого охоплення аудиторії при мінімальних ресурсних витратах. Проведено аналіз найпопулярніших фреймворків, серед яких для розробки кваліфікаційного проєкту обрано Kivu як оптимальне рішення, що поєднує доступність мови Python із перевагами кросплатформної розробки.

Окремо розглянуто особливості електронних бібліотек як типу цифрових інформаційних систем, а також визначено основні вимоги до інтерфейсу користувача мобільного застосунку, серед яких особливе місце займають зручність навігації, оптимізація процесу читання та ефективна організація контенту. Сукупність отриманих результатів створює надійну теоретичну базу для подальшого переходу до етапу практичної реалізації мобільного додатку електронної бібліотеки, який буде детально описано у наступному розділі.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ

2.1 Постановка задачі

Розробка мобільного застосунку електронної бібліотеки передбачає створення системи, яка забезпечує користувачам зручний доступ до цифрових книг, інтуїтивну навігацію, можливість збереження матеріалів для офлайн-доступу, а також реалізацію базових функцій авторизації та персоналізації. Однією з ключових особливостей такого додатку є його здатність працювати на різних мобільних платформах, зокрема Android та iOS, що обумовлює необхідність використання кросплатформних інструментів розробки.

У межах проєкту було сформульовано низку функціональних вимог, серед яких каталогізація книг із відображенням обкладинок, назв і короткого опису, реалізація пошуку книг за ключовими словами, можливість перегляду вмісту книг у форматах PDF або TXT, збереження книг у локальну пам'ять пристрою для забезпечення офлайн-доступу, авторизація користувача з базовим збереженням персональних даних, додавання та видалення книг через інтерфейс застосунку, орієнтований на адміністратора або власника пристрою, а також адаптація графічного інтерфейсу під різні розміри екранів мобільних пристроїв.

Окрім функціональних характеристик, до системи були висунуті і нефункціональні вимоги. Серед них важливу роль відіграють простота процесу інсталяції та запуску програми, мінімальні затримки при відкритті сторінок, стабільна робота на платформі Android із подальшою генерацією APK-файлу, а також можливість повноцінного використання програми без постійного підключення до інтернету для читання книг та виконання базового пошуку.

Для реалізації зазначених вимог було обрано мову програмування Python у поєднанні з фреймворком Kivu, що забезпечує необхідну кросплатформність. Для збірки Android-версії застосовується утиліта Buildozer. З метою локального

збереження даних передбачається використання вбудованої бази даних SQLite або організація файлової структури для зберігання інформації.

2.2 Архітектура мобільного додатку

Мобільний додаток електронної бібліотеки створюється на основі архітектури, яка поєднує простоту, розширюваність та кросплатформну адаптацію. Його структура побудована відповідно до принципів розділення логіки, інтерфейсу та даних, що істотно полегшує подальший супровід і розвиток застосунку.

Користувацький інтерфейс реалізується за допомогою фреймворку Kivu із використанням мови опису інтерфейсу у .kv-файлах. Інтерфейс включає екран авторизації, екран реєстрації нового користувача, головний екран із вітанням, каталог доступних книг та екран перегляду обраної книги. Навігація між цими екранами організована за допомогою менеджера екранів (ScreenManager), що забезпечує централізоване керування переходами між вікнами без дублювання коду. Кожен екран має унікальне ім'я, яке використовується для звернення під час перемикання інтерфейсів.

База даних проєкту реалізована за допомогою SQLite і використовується для зберігання облікових записів користувачів, загального каталогу книг із зазначенням назви, автора, опису та шляху до файлу. У перспективі передбачається також організація персональних бібліотек для окремих користувачів. Доступ до бази здійснюється через окремий модуль database.py, який містить необхідні функції для перевірки логіну, реєстрації користувачів та отримання списку книг. Паролі зберігаються у вигляді хешів, що забезпечує підвищений рівень безпеки даних.

Логіка взаємодії реалізована через окремі контролери, кожен з яких відповідає за поведінку конкретного екрана. Наприклад, екран входу обробляє введення логіна та пароля, перевіряючи їх правильність, тоді як екран каталогу

завантажує список книг із бази даних і формує їх відображення у відповідному форматі.

Архітектура передбачає можливість подальшого розширення шляхом інтеграції додаткових модулів. Серед потенційних напрямів розвитку системи планується впровадження сканування штрих-кодів для автоматичного імпорту книг за ISBN, реалізація персоналізованих рекомендацій на основі вподобань користувача, а також збереження історії читання.

Загалом обрана архітектура відповідає сучасним вимогам кросплатформної розробки мобільних застосунків, забезпечуючи модульність, зрозумілу логіку роботи системи та можливість подальшого нарощування функціоналу відповідно до зростаючих потреб користувачів.

2.3 Інтерфейс користувача

Інтерфейс користувача є ключовою частиною будь-якого програмного забезпечення, що орієнтується на кінцевого користувача, адже саме з ним взаємодіє людина у процесі роботи з системою. У розробленому мобільному додатку електронної бібліотеки основна увага приділена інтуїтивності, простоті та функціональності, що дозволяє охопити широкий спектр користувачів із різним рівнем технічної підготовки.

Головне вікно додатку складається з навігаційної панелі, з якої користувач може перейти до каталогу книг, своєї персональної бібліотеки, а також здійснити вихід із профілю. Після входу або реєстрації користувач одразу потрапляє на домашній екран, де має змогу обрати подальші дії. Перехід до каталогу відкриває список усіх доступних книг, реалізований у вигляді прокручуваного контейнера з окремими блоками-картками, де кожна картка містить назву книги, автора та короткий опис.

Інтерфейс підтримує повнотекстовий пошук, що здійснюється за назвою або ім'ям автора. Введення даних у відповідне поле автоматично оновлює результати без потреби натискання кнопки підтвердження. Окрім того,

реалізовано динамічне автодоповнення авторів: у процесі введення у відповідному полі з'являються підказки з найближчими варіантами, які можна обрати в один клік. Це значно спрощує навігацію у випадку великої кількості авторів.

Додатково користувач має змогу активувати фільтр «Мої книги», який дозволяє переглядати лише ті книги, що вже додані у персональну бібліотеку. Кожну книгу можна переглянути окремо: у цьому режимі виводиться повний текст книги, а також кнопка для додавання або видалення її з особистої колекції. Кнопка змінюється динамічно в залежності від наявності книги у бібліотеці, що робить інтерфейс максимально адаптивним.

В цілому, інтерфейс побудований із використанням кросплатформного фреймворку Kivu, який дозволив реалізувати адаптивне масштабування елементів, прозору навігацію між екранами та ефективну обробку подій. Простота й логічна структура інтерфейсу дозволяє легко орієнтуватися в додатку навіть новим користувачам, а використання сучасних практик UI/UX-дизайну сприяє підвищенню задоволеності від роботи з мобільним застосунком.

2.4 Архітектура та логіка роботи додатку

Архітектура мобільного додатку електронної бібліотеки побудована з дотриманням принципів модульності, логічної ізоляції компонентів та простоти супроводу. Основною метою при формуванні структури стало забезпечення зрозумілої організації коду, що дозволяє ефективно розширювати функціонал у майбутньому, а також легко орієнтуватися в логіці роботи окремих модулів.

Додаток реалізований із використанням фреймворку Kivu, що дозволяє створювати кросплатформні інтерфейси. Основу навігації складає клас ScreenManager, який керує перемиканням між екранами. Кожен екран реалізовано як окремий клас, що наслідує Screen і відповідає за логіку відповідного інтерфейсу: екран авторизації, реєстрації, головний екран, каталог книг, перегляд книги та персональна бібліотека. Візуальне представлення

реалізоване у файлах з розширенням `.kv`, де за допомогою декларативного синтаксису описано компонування елементів інтерфейсу.

База даних додатку реалізована на основі SQLite, що дозволило легко інтегрувати локальне зберігання інформації без використання сторонніх серверів. Всі взаємодії з базою даних винесено в окремий модуль `database.py`, де описано функції для створення таблиць, перевірки користувачів, збереження книг, додавання книг до персональної бібліотеки та пошуку. Це дозволяє логічно відокремити обробку даних від інтерфейсу та контролерів.

Логіка роботи з книгами включає дві основні частини: роботу з глобальним каталогом та персональною бібліотекою користувача. Кожна книга містить базову інформацію: назву, автора, опис та шлях до текстового файлу. Користувач може додавати книги в особисту колекцію, після чого вони зберігаються у таблиці зв'язків між користувачем і книгою. Перевірка наявності книги в бібліотеці, як і логіка кнопки додавання/вилучення, реалізована динамічно та враховує поточний статус.

Інтерфейс каталогу має розширену логіку: при введенні пошукового запиту автоматично фільтрується список книг за назвою або автором. Реалізовано також фільтр за автором із автодоповненням, який оновлює список підказок у процесі введення. Функція «тільки мої книги» дозволяє обмежити виведення до персональної бібліотеки поточного користувача. Усі фільтри працюють у зв'язці, обробляючи запит на стороні бази даних і повертаючи лише релевантні результати.

Загальна архітектура додатку є гнучкою і модульною: нові екрани або функції (наприклад, сканування ISBN чи рекомендації на основі вподобань) можуть бути додані без суттєвих змін до існуючої структури. Такий підхід дозволяє масштабувати проєкт відповідно до потреб користувачів та потенційного розширення в майбутньому.

2.5 Порівняння з аналогами електронних бібліотек

У сучасному цифровому середовищі існує значна кількість електронних бібліотек та платформ для читання книг. Серед найпоширеніших рішень можна виділити такі сервіси, як Google Books, Amazon Kindle, Wattpad, а також мобільні додатки локальних онлайн-бібліотек, зокрема BiblioNet, Bookmate, Librarius або як приклад української реалізації можна навести мобільний застосунок eЧитальня, що є частиною цифрової бібліотечної інфраструктури [15]. Кожен із цих ресурсів має свої сильні сторони та унікальні функціональні можливості, проте не всі вони доступні для локального використання, не всі підтримують відкриті формати книг або не мають повноцінної офлайн-роботи без авторизації.

Amazon Kindle – один із найвідоміших у світі застосунків для читання, проте він прив'язаний до власної екосистеми Kindle Store і не дозволяє вільно працювати з книгами, що не були придбані через магазин. Google Books, хоч і підтримує офлайн-завантаження книг, має обмеження по доступності контенту залежно від регіону, а також не завжди надає користувачеві змогу імпортувати власні книги в бібліотеку.

Для наочності основні функціональні можливості зазначених платформ порівняно з розробленим мобільним застосунком зведено у таблицю 2.1.

Таблиця 2.1 – Порівняння функціональних можливостей популярних електронних бібліотек із розробленим застосунком

Критерій / Платформа	Amazon Kindle	Google Books	Librarius	Розроблений застосунок
Потрібна авторизація	Так	Так	Так	Ні
Офлайн-доступ	Частково	Частково	Так	Повний
Можливість імпорту книг	Ні	Частково	Ні	Так
Підтримка відкритих форматів	Ні	Частково	Частково	Так

Продовження таблиці 2.1

Критерій / Платформа	Amazon Kindle	Google Books	Librarius	Розроблений застосунок
Локальна база даних	Ні	Ні	Частково	Так
Інтеграція з ISBN/API	Ні	Так	Ні	Так

На відміну від перелічених рішень, розроблений у межах кваліфікаційної роботи мобільний додаток має низку суттєвих переваг у певних сценаріях використання. По-перше, він працює повністю локально – зберігання книг та облікових даних реалізовано через вбудовану базу даних SQLite. Це дозволяє використовувати додаток без постійного підключення до Інтернету. По-друге, система дозволяє імпортувати довільні книги в текстовому форматі та самостійно створювати бібліотеку. Користувач має повний контроль над своїм контентом.

Додатковою перевагою реалізованого рішення є відкритість архітектури: за потреби її можна доповнити модулями штучного інтелекту, наприклад, для генерації рекомендацій, або інтегрувати з зовнішніми API для пошуку книг за ISBN. Простота коду та зрозуміла структура дозволяють адаптувати програму до конкретних потреб – чого часто не дають готові комерційні платформи.

Отже, хоча розроблений застосунок не конкурує з глобальними платформами за масштабами чи функціональністю, він вирізняється локальністю, кастомізованістю та повною автономністю, що робить його цінним інструментом для освітніх закладів, бібліотек або індивідуального використання.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ

У цьому розділі наведено опис етапів практичної реалізації програмного забезпечення відповідно до концептуальної моделі, визначеної у попередніх частинах роботи. Метою є створення повноцінного мобільного застосунку, який забезпечує користувачу зручний інтерфейс для перегляду, збереження та управління електронними книгами, а також реалізує базові функції взаємодії з даними, авторизації, персоналізації та пошуку.

Реалізація проводилася з використанням мови програмування Python і кросплатформного фреймворку Kivy, що забезпечило підтримку розгортання на мобільних пристроях і персональних комп'ютерах. В якості сховища даних застосовано вбудовану базу даних SQLite, яка дозволяє зберігати облікові записи, метадані книг та зв'язки між користувачами і їх персональними бібліотеками. У структурі застосунку дотримано принципів модульності, кожен компонент реалізовано окремо з чітким розмежуванням функцій.

Детальний опис кожного етапу реалізації, включаючи створення інтерфейсів, побудову бази даних, логіку авторизації, пошуку, персоналізації та управління бібліотекою, наведено в підрозділах даного розділу.

3.1 Ініціалізація проєкту та структура файлів

На початковому етапі реалізації було створено окреме середовище розробки мобільного застосунку з використанням мови програмування Python і кросплатформного фреймворку Kivy. Проєкт побудовано за модульною структурою з чітким розмежуванням логіки, обробки даних та інтерфейсу користувача, що забезпечує гнучкість, масштабованість і зручність подальшої підтримки.

Головний файл програми, `main.py`, відповідає за ініціалізацію застосунку, підключення до бази даних, завантаження інтерфейсів, керування навігацією між екранами та збереження інформації про поточного користувача. Для роботи з локальною базою даних використовується окремий модуль `database.py`, який містить функції створення таблиць, обробки процесів реєстрації та авторизації, додавання книг до бази, управління персональними бібліотеками, реалізації пошуку, фільтрації та обробки закладок. Інтерфейси екранів реалізовані через окремі `.kv`-файли, згруповані у директорії `screens`. Серед них – екрани для авторизації, реєстрації нового користувача, головного меню, каталогу книг, перегляду вмісту книги з підтримкою закладок, персональної бібліотеки та сканування ISBN-кодів за допомогою камери.

Усі необхідні ресурси, включно з локальними файлами книг і збереженням закладок у форматі JSON, зберігаються у директорії `assets`. Для обробки додаткових задач, таких як імпорт літератури з відкритих бібліотек, очищення бази даних або отримання даних через API `OpenLibrary`, у проєкті передбачена окрема директорія `scripts` із допоміжними скриптами. Усі залежності проєкту описані у файлі `requirements.txt`, що дозволяє швидко відтворити середовище розробки. Серед основних бібліотек, які використовуються, – `kivy`, `pyzbar`, `opencv-python`, а також модулі для роботи з HTTP-запитами та базами даних, включаючи `requests`.

Візуальна частина додатку створюється за допомогою декларативних `.kv`-файлів, що дозволяє чітко розділити логіку застосунку та його інтерфейсну складову. Це не лише спрощує підтримку коду, а й забезпечує адаптивність інтерфейсу до різних розмірів екранів мобільних пристроїв і настільних систем. Навігація між екранами організована через об'єкт `ScreenManager`, ініціалізований у файлі `main.py`. Кожен екран представлений окремим класом, що взаємодіє зі своїм інтерфейсом через властивості, події та прив'язку до даних.

Особлива увага під час розробки приділялась забезпеченню кросплатформності: завдяки використанню `Kivy` додаток може бути перенесений як на `Android`, так і на `iOS` з мінімальними змінами в коді. Це дозволяє охопити

ширшу аудиторію без потреби у створенні окремих версій застосунку для кожної платформи.

Таким чином, загальна структура проєкту орієнтована на забезпечення модульності, масштабованості та зручності супроводу [16]. Це повністю відповідає поставленій меті створення повноцінного мобільного застосунку електронної бібліотеки.

3.2 Розробка локальної бази даних

Для зберігання користувацьких даних, інформації про книги та формування персональних бібліотек у мобільному застосунку реалізовано локальну базу даних на основі SQLite [17]. Вибір цієї технології обумовлений її легкістю, кросплатформністю, відсутністю потреби у зовнішньому сервері та повною інтеграцією з Python через стандартну бібліотеку `sqlite3`. Це дозволяє додатку працювати автономно, що є важливою перевагою для користувачів без постійного доступу до Інтернету.

База даних створюється автоматично під час першого запуску програми за допомогою вбудованого модуля `sqlite3` [18]. Відповідна логіка реалізована у файлі `database.py`, де функція `init_db()` відповідає за створення необхідних таблиць, якщо вони ще не існують. Основна структура бази включає таблицю користувачів, яка містить унікальні імена та хешовані паролі, таблицю книг із назвами, авторами, описами, шляхами до локальних текстових файлів та жанрами, що слугують основою для фільтрації, а також таблицю зв'язків між користувачами та книгами, яка дозволяє формувати індивідуальні добірки у межах персональної бібліотеки.

Уся взаємодія з базою даних інкапсульована у модулі `database.py` у вигляді функцій, кожна з яких відповідає за окрему операцію. Наприклад, функції реєстрації та перевірки користувачів відповідають за додавання нового облікового запису та перевірку правильності введених даних під час авторизації, а функції отримання переліку книг та пошуку забезпечують завантаження

відповідного контенту з урахуванням запитів користувача. Також реалізовано можливість додавання та видалення книг у персональну бібліотеку, а функція `get_user_books()` виводить список видань, прив'язаних до конкретного користувача.

Для зберігання паролів використовується алгоритм SHA-256, реалізований стандартним модулем `hashlib`, що забезпечує захист паролів у разі спроби несанкціонованого доступу до вмісту бази. Особливу увагу приділено безпеці під час роботи із запитом – усі SQL-операції побудовані з використанням параметризованих конструкцій, що унеможлиблює SQL-ін'єкції й тим самим підвищує надійність додатку.

Поточна структура бази даних передбачає можливість подальшого розширення, включно з додаванням таблиць для зберігання історії читання, рейтингів або системи рекомендацій на основі алгоритмів штучного інтелекту. Крім того, уже передбачена технічна можливість інтеграції із зовнішніми джерелами даних, зокрема онлайн-бібліотеками. Таким чином, реалізована база даних у повній мірі відповідає функціональним вимогам на поточному етапі розробки, забезпечуючи зберігання облікової інформації, метаданих про книги, персональних добірок, закладок, а також підтримуючи надійну й безпечну взаємодію користувача з програмою.

3.3 Реалізація авторизації та реєстрації

Система автентифікації є ключовою частиною персоналізованого функціоналу мобільного застосунку. У межах розробленого проекту реалізовано просту, надійну та автономну схему входу й реєстрації користувачів без залучення зовнішніх серверів, що забезпечує повну незалежність програми від сторонніх служб.

Функціональність авторизації реалізована через екран `LoginScreen` (рисунок 3.1), де користувач вводить свій логін і пароль для входу. Передані дані перевіряються у базі даних шляхом виклику функції `check_user`, яка порівнює

введений пароль із хешованим значенням, збереженим у базі, використовуючи алгоритм SHA-256 із модуля hashlib. У разі неправильного введення облікових даних користувачу відображається спливаюче вікно повідомлення про помилку (Popup), що інформує про невідповідність логіну або паролю.

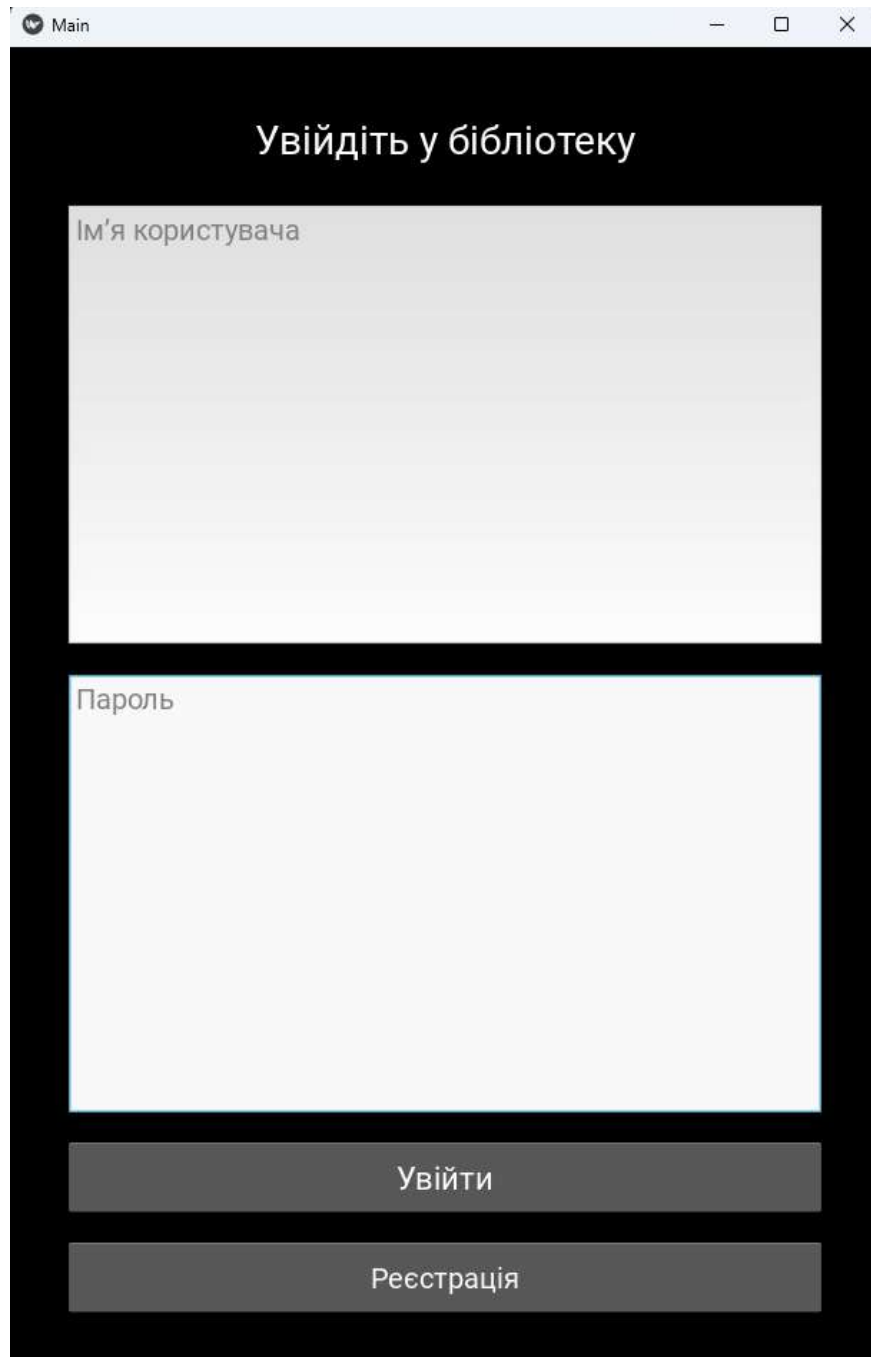
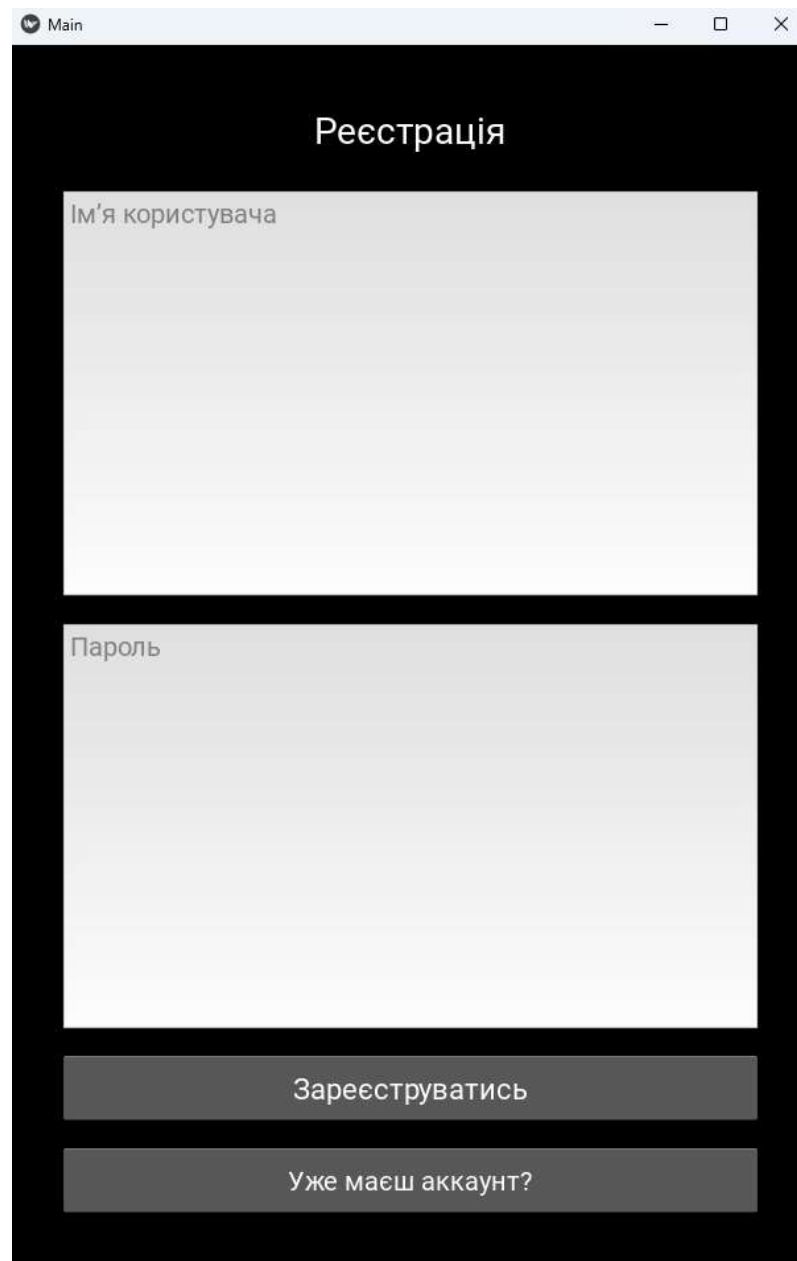


Рисунок 3.1 – Екран авторизації користувача

Реєстрація нового користувача відбувається через окремий екран RegisterScreen (рисунок 3.2). Під час реєстрації програма перевіряє унікальність

обраного логіна, а введені дані очищуються від зайвих пробілів для уникнення помилок. Пароль хешується за допомогою SHA-256 і тільки після цього записується до таблиці users бази даних. Якщо користувач намагається зареєструватися під вже існуючим логіном, система інформує його про це за допомогою відповідного повідомлення про помилку.



The image shows a web browser window titled "Main" with a registration form. The form is titled "Реєстрація" (Registration) and contains two input fields: "Ім'я користувача" (Username) and "Пароль" (Password). Below the fields are two buttons: "Зареєструватись" (Register) and "Уже маєш аккаунт?" (Already have an account?).

Рисунок 3.2 – Екран реєстрації нового користувача

Після успішного проходження процедури входу або реєстрації ім'я користувача відбувається перенаправлення на головну сторінку (рисунок 3.3),

зберігається у властивості `current_user` головного застосунку. Завдяки цьому застосунок може ідентифікувати активного користувача й відповідно формувати доступ до персональної бібліотеки книг, фільтрувати результати пошуку згідно з особистими даними користувача, а також автоматично завантажувати індивідуальні закладки під час відкриття книг.

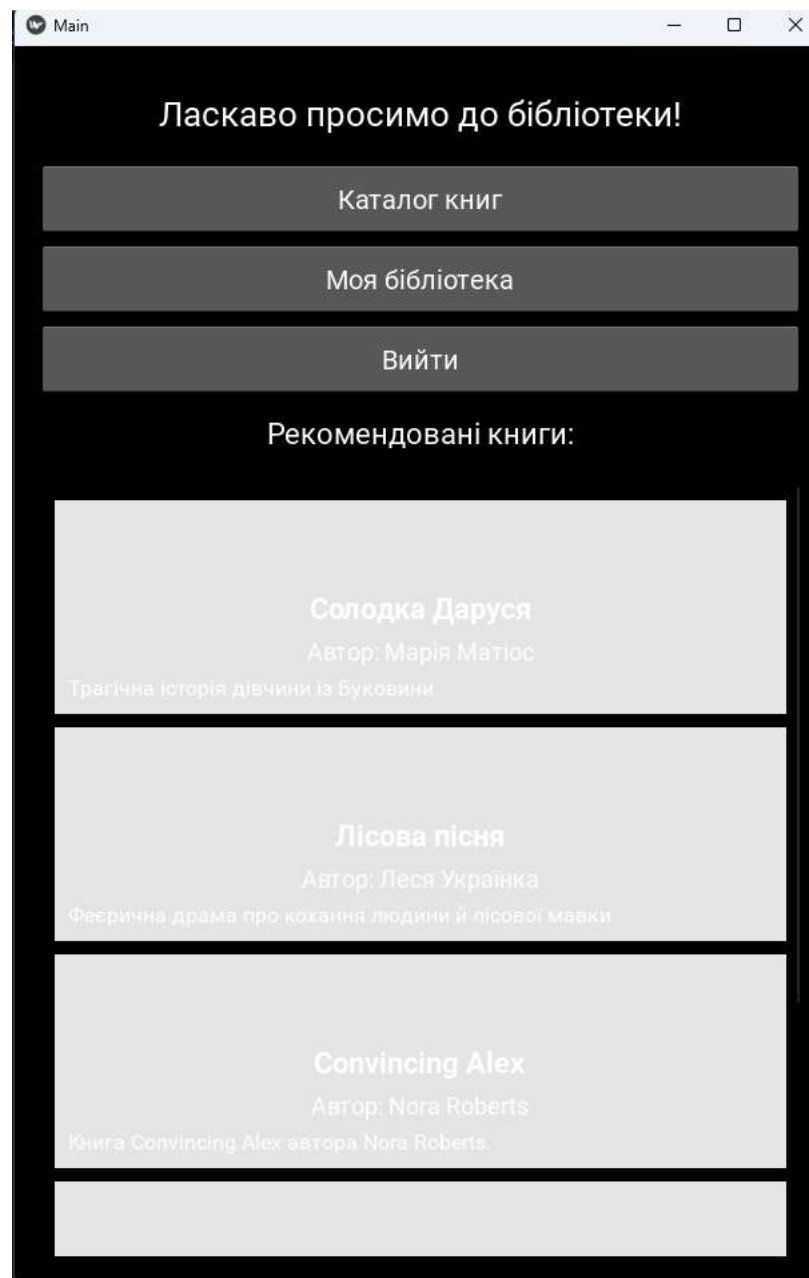


Рисунок 3.3 – Домашній екран застосунку

З архітектурної точки зору, система авторизації побудована на взаємодії між Python-класами, що містять логіку обробки даних, та відповідними .kv-

шаблонами, які визначають вигляд інтерфейсу. Такий підхід сприяє чіткому розділенню відповідальностей у програмі: логіка відокремлена від представлення, що значно полегшує супровід, налагодження та подальший розвиток системи. Завдяки цьому структура додатку є прозорою й масштабованою, що дозволяє без суттєвих змін інтегрувати нові можливості, як-от вхід через акаунти соціальних мереж, підтримку OAuth або впровадження двофакторної автентифікації.

Сама реалізація автентифікації дає змогу кожному користувачеві працювати у власному середовищі, зберігаючи персональні бібліотеки книг, уподобання та індивідуальні налаштування. При цьому система не залежить від зовнішніх серверів, що гарантує автономність роботи, конфіденційність та стабільність навіть за відсутності Інтернет-з'єднання.

3.4 Каталог книг та інтерфейс перегляду

Каталог книг є центральним елементом мобільного застосунку, який забезпечує інтуїтивну та зручну взаємодію користувача з електронною бібліотекою. Завдяки його реалізації користувач може легко орієнтуватися у великому обсязі доступної літератури, швидко знаходити потрібні книги за ключовими словами або критеріями фільтрації, а також користуватися персоналізованими можливостями вибору, що сприяє покращенню загального досвіду використання програми.

З технічного погляду, каталог реалізовано за допомогою екрана CatalogScreen, який відображає книги у форматі вертикального списку у вигляді окремих карток (BookCard), кожна з яких містить коротку інформацію про книгу та кнопку для перегляду повного вмісту (рисунки 3.4). Такий підхід забезпечує як візуальну привабливість, так і зручність навігації.



Рисунок 3.4 – Каталог книг із фільтрацією

Кожна картка містить базову інформацію про видання, зокрема назву книги, ім'я автора та короткий опис її змісту, що дозволяє користувачу швидко зорієнтуватися у доступному контенті. При натисканні на картку здійснюється автоматичний перехід до окремого екрана детального перегляду — BookViewScreen, де відображається повна інформація про книгу та надається можливість її читання (рисунок 3.5). Такий механізм взаємодії забезпечує послідовний і логічний користувацький шлях, підвищуючи зручність користування застосунком.

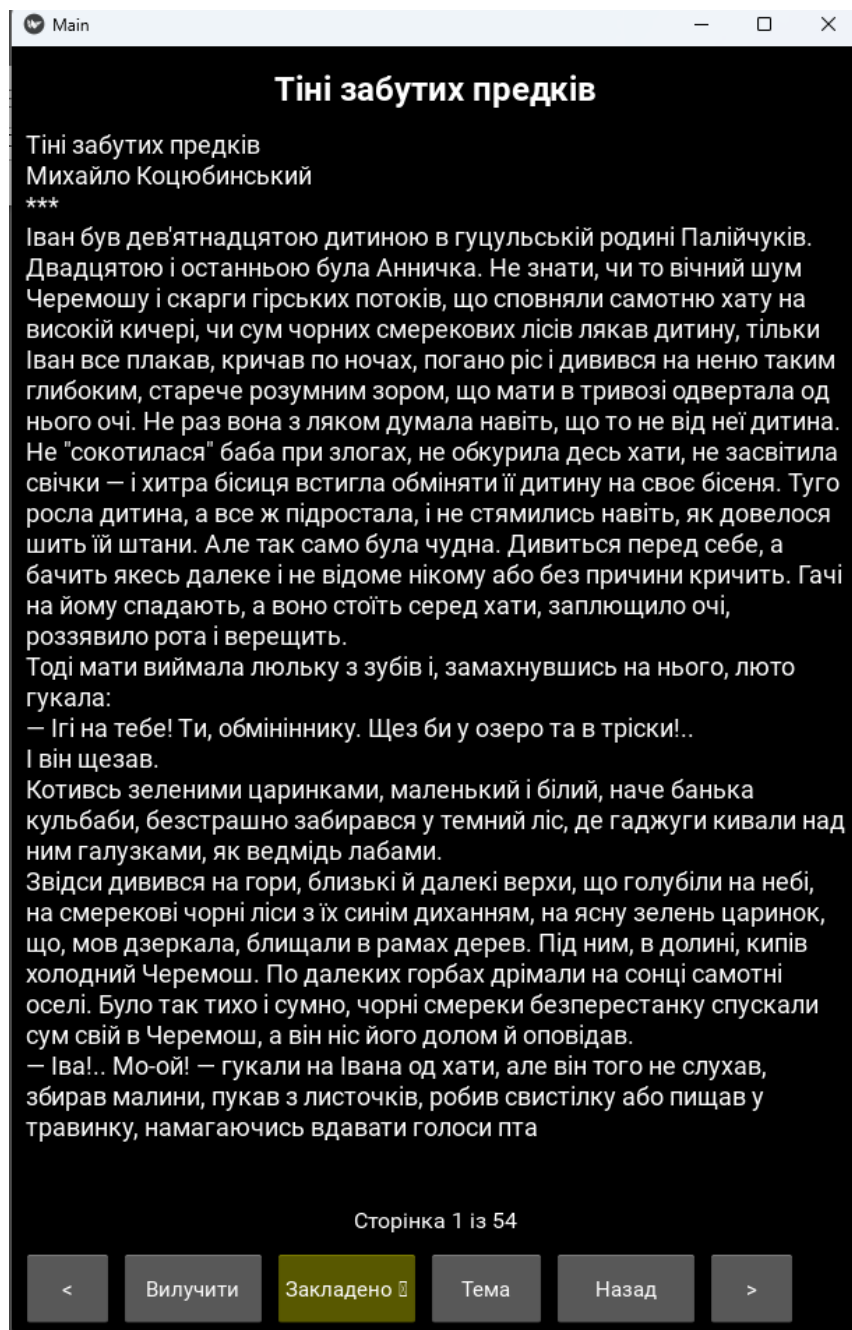


Рисунок 3.5 – Перегляд книги (темна тема)

Інформація про книги завантажується динамічно з локальної бази даних SQLite через функції `get_all_books_full()` та `search_books()`. Кожен запис у базі містить як основні метадані книги, так і шлях до локального текстового файлу, який використовується для читання її вмісту. Для підвищення зручності користування каталогом реалізовано механізм «живого пошуку», що оновлює список результатів при кожному введенні символу у відповідне поле пошуку без необхідності додаткового підтвердження запиту.

Пошук і фільтрація в каталозі побудовані таким чином, щоб надати користувачу максимальну гнучкість. Книги можна знаходити за назвою або ім'ям автора, причому система підтримує інтеграцію динамічних підказок за префіксом імені. Також передбачено можливість фільтрації за жанрами через спеціальний інтерфейсний елемент Spinner і за наявністю книги у персональній бібліотеці користувача, що активується відповідним чекбоксом «Мої книги». Усі пошукові запити виконуються без урахування реєстру символів, що робить систему більш зручною та інтуїтивною. При використанні кількох фільтрів одночасно результати оновлюються динамічно, відповідно до всіх обраних умов.

Інтерфейс перегляду книги організовано через екран BookViewScreen, який забезпечує повноцінний режим читання. Текст книги автоматично ділиться на сторінки об'ємом приблизно 1800 символів, що дає змогу користувачу зручно переходити між сторінками за допомогою навігаційних кнопок. Передбачено можливість встановлення закладки на поточній сторінці, яка зберігається у локальному файлі bookmarks.json і прив'язується до облікового запису користувача. Крім того, реалізовано функціонал перемикання між темною та світлою темою для адаптації під умови освітлення та особисті уподобання користувача. При відкритті книги система автоматично підвантажує останню закладку, дозволяючи продовжити читання саме з того місця, на якому було завершено сеанс.

Інтерфейс читання оптимізований для мобільних пристроїв: шрифти адаптивно масштабуються під різні розміри екранів, а кнопки управління розташовані компактно й інтуїтивно зрозуміло, що забезпечує комфортне використання навіть на невеликих смартфонах.

Додатково у функціонал каталогу інтегровано можливість сканування ISBN-кодів за допомогою камери пристрою. Завдяки цій можливості користувач може миттєво знайти відповідну книгу у базі даних або, за потреби, завантажити відсутню інформацію з відкритих джерел, таких як OpenLibrary.

Таким чином, реалізований каталог книг не лише забезпечує зручну навігацію серед електронних видань, але й поєднує функції повноцінної сучасної

електронної читалки, інтегруючи пошук, фільтрацію, персоналізацію користувацького досвіду та інтуїтивно зрозумілий, адаптивний інтерфейс.

3.5 Персональна бібліотека користувача

Персональна бібліотека є ключовим елементом мобільного застосунку, що дозволяє користувачу формувати власну колекцію книг для швидкого доступу та комфортного читання. Вона не лише забезпечує зручність користування, а й слугує основою для реалізації персоналізованих функцій, таких як рекомендації на основі індивідуальних інтересів, історії читання та вподобань.

З технічного боку, для зберігання персональної бібліотеки використовується локальна база даних, у якій реалізовано таблицю зв'язків `user_books`. Ця таблиця асоціює записи між основними таблицями `users` і `books`, що дає змогу кожному користувачу створювати власний набір літератури без дублювання самих книг у сховищі. Коли користувач додає книгу до своєї бібліотеки, система перевіряє її наявність у поточному наборі за допомогою функції `user_has_book`. У разі відсутності запису створюється новий зв'язок у таблиці `user_books`, а якщо книга вже є — користувачу пропонується можливість її вилучення шляхом повторного натискання на відповідну кнопку.

Уся логіка взаємодії з персональною бібліотекою зосереджена в межах методів класу `BookViewScreen`, що надає змогу змінювати статус книги — додавати її до добірки або вилучати — без необхідності повертатися до головного каталогу. Це дозволяє користувачу залишатися в контексті перегляду книги, спрощуючи роботу з інтерфейсом і зменшуючи кількість зайвих дій. Для перегляду всієї особистої колекції передбачено окремий екран `MyLibraryScreen`, на якому відображаються всі видання, що були додані до індивідуальної бібліотеки користувача (рисунок 3.6). Така структура сприяє інтуїтивній навігації, забезпечує швидкий доступ до улюблених книг та створює гнучке підґрунтя для реалізації додаткових функцій, таких як сортування, фільтрація або статистика читання.

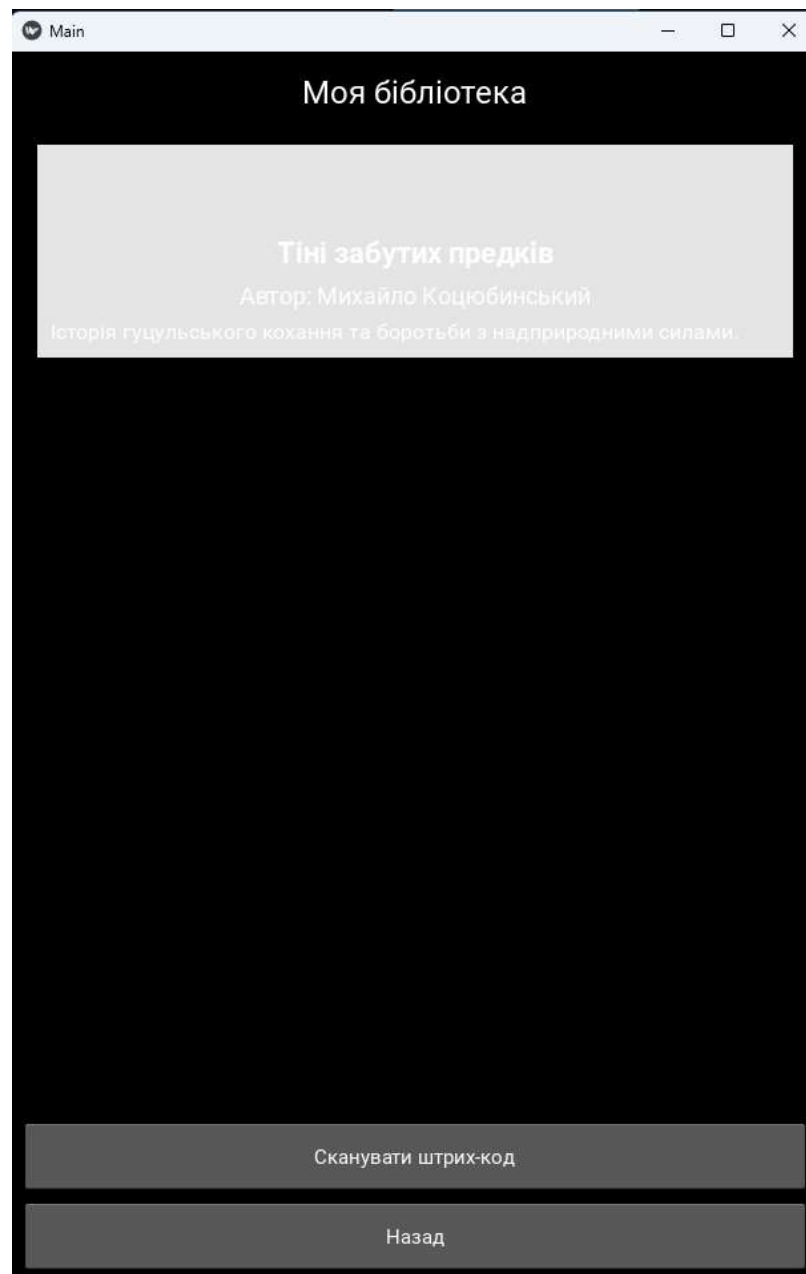


Рисунок 3.6 – Моя бібліотека список особисто доданих книг, можливість повернення або сканування нових.

Його функціонал дозволяє динамічно формувати список карток BookCard, які за структурою аналогічні карткам загального каталогу, проте відображають лише ті книги, які були додані конкретним користувачем. Інтерфейс персональної бібліотеки дозволяє прокручувати список власних книг, відкривати вибрану книгу для читання безпосередньо з персональної бібліотеки, а також автоматично завантажувати збережені закладки при відкритті книги. При цьому інтерфейс персональної бібліотеки підтримує єдину стилістику з основним

каталогом, що забезпечує цілісність користувацького досвіду та спрощує навігацію.

Особливістю реалізації персональної бібліотеки є її повна прив'язка до облікового запису користувача через властивість `current_user`. Це гарантує надійне розмежування даних між різними обліковими записами та забезпечує безпечну персоналізацію контенту. Після додавання або видалення книги статус бібліотеки оновлюється у реальному часі без потреби перезапуску застосунку, що сприяє підвищенню зручності користування. Система розроблена з урахуванням можливості масштабування, що дозволяє у майбутньому розширити її функціонал, наприклад, шляхом додавання можливості оцінювання книг, створення списків бажаного прочитання чи впровадження аналітики часу, витраченого на читання.

Таким чином, персональна бібліотека ефективно реалізує одну з ключових цілей застосунку, створюючи для кожного користувача індивідуальне середовище для збереження, перегляду та управління власною електронною літературою.

3.6 Пошук та фільтрація та сканування книг

Пошук і фільтрація є критичними елементами сучасних мобільних застосунків, що працюють із великими обсягами даних. У розробленому мобільному додатку реалізовано розширений функціонал швидкого пошуку, багаторівневої фільтрації та сканування ISBN-кодів для спрощення доступу до контенту.

Пошук у каталозі книг здійснюється у режимі реального часу. При кожному введенні символу в поле пошуку автоматично викликається функція `do_search()`, яка оновлює відображуваний список книг без необхідності перезавантаження екрана або підтвердження запиту. Пошукова система одночасно обробляє запити як за назвою книги, так і за іменем автора, що дозволяє користувачу швидко отримувати релевантні результати. Для

підвищення зручності реалізовано нечутливість до регістру введених символів завдяки використанню оператора LIKE ... COLLATE NOCASE у SQL-запитах, що дозволяє знаходити потрібні книги незалежно від введення великих чи малих літер.

З метою полегшення пошуку за автором інтегровано систему динамічних підказок. При введенні кількох перших символів у відповідне поле система у режимі реального часу формує список авторів, чий імена відповідають введеному префіксу. Обравши варіант із підказки, користувач миттєво отримує оновлений список книг, що відповідають обраному автору. Формування підказок здійснюється через виклик функції `find_authors_like()`, яка підтягує дані безпосередньо з бази даних.

Для розширення можливостей пошуку окремо реалізовано фільтрацію за жанром книги через інтеграцію елемента керування `Spinner`. При виборі конкретного жанру система автоматично оновлює список книг відповідно до обраного критерію. Якщо користувач обирає опцію «Всі жанри», фільтрація за цією ознакою скасовується, дозволяючи переглядати весь доступний каталог. Механізм фільтрації за жанрами повністю інтегрований у загальну пошукову систему і може комбінуватися з іншими умовами пошуку для досягнення більш точних результатів.

Додатково передбачено можливість фільтрації за персональними книгами користувача. Чекбокс «Мої книги» дозволяє обмежити пошук лише тими виданнями, які були раніше додані до персональної бібліотеки. При активації цього режиму результати пошуку формуються через об'єднання таблиць `user_books` та `users`, що гарантує правильне відображення індивідуальних даних для кожного користувача.

У рамках розширення пошукових можливостей реалізовано функціонал сканування ISBN-кодів книг за допомогою камери мобільного пристрою. (рисунок 3.7)

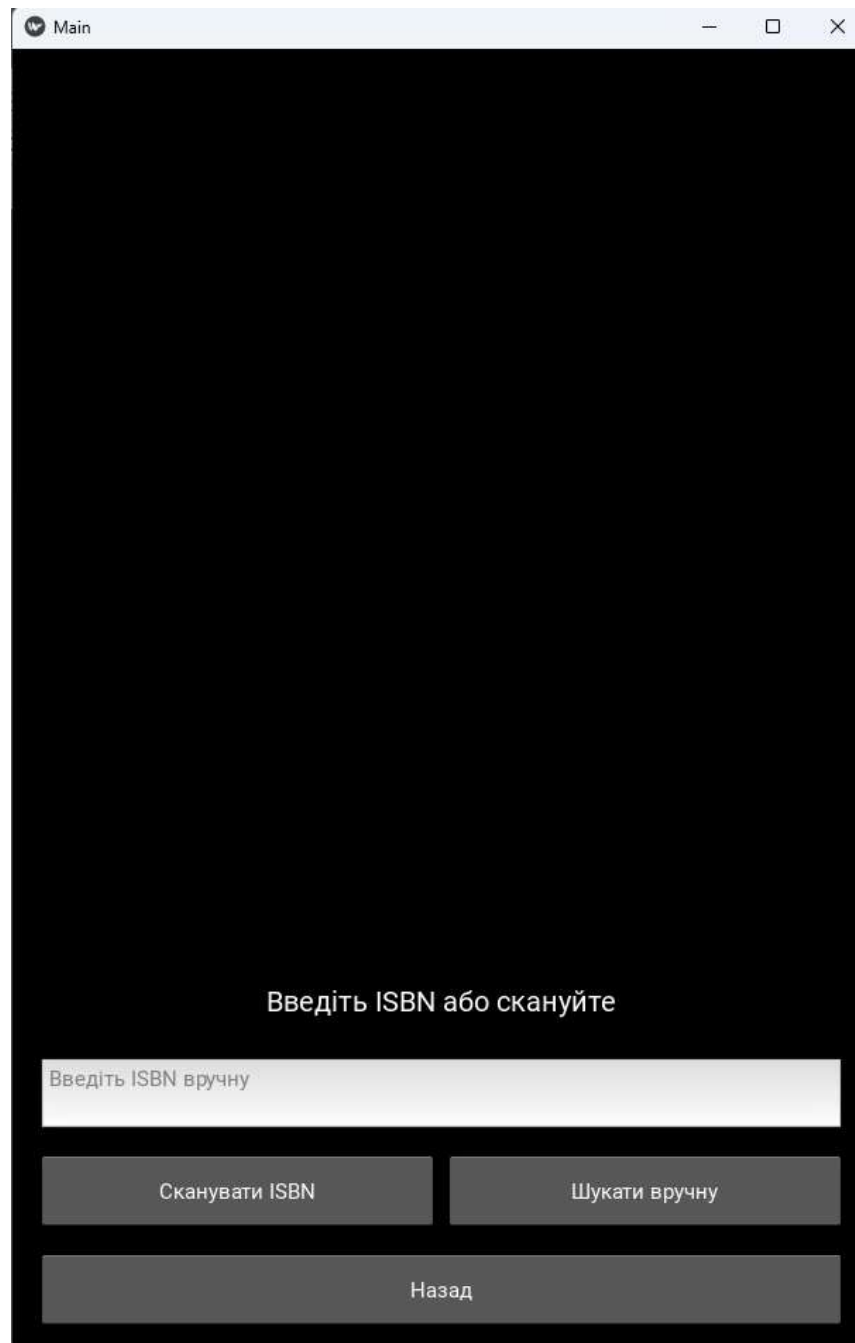


Рисунок 3.7 – Вікно сканування або введення ISBN інтерфейс пошуку книги за штрихкодом або ручним введенням коду.

Після сканування штрихкоду система спочатку здійснює пошук книги у локальній базі даних, а за відсутності відповідного запису автоматично звертається до відкритого API OpenLibrary для отримання додаткової інформації. Це дозволяє користувачу швидко знайти потрібну книгу як за текстовим запитом, так і за фізичним скануванням реального примірника.

Таким чином, реалізований механізм пошуку, фільтрації та сканування ISBN забезпечує швидкий, гнучкий та інтуїтивно зрозумілий доступ до великого обсягу літературного контенту, що відповідає сучасним вимогам до зручності та функціональності мобільних застосунків.

3.7 Розширені функціональні можливості

На завершальному етапі розробки мобільного застосунку було реалізовано низку додаткових функцій, що виходять за межі базового перегляду електронних книг. Метою впровадження розширених можливостей стало підвищення зручності використання застосунку, персоналізація досвіду користувачів, а також забезпечення інтеграції із зовнішніми джерелами даних. Основними напрямками розширення функціоналу стали інтеграція з відкритими бібліотечними API для автоматичного поповнення колекції книг за ISBN-кодами, реалізація механізму сканування ISBN через камеру пристрою для спрощення процесу додавання нових книг, побудова бази для створення рекомендаційної системи на основі особистих вподобань користувача, а також підготовка застосунку до кросплатформенного розгортання на Android та iOS.

Реалізація цих функцій дозволила не лише підвищити практичну цінність проєкту, а й продемонструвати можливості його масштабування відповідно до сучасних вимог, що висувуються до мобільних електронних бібліотек. У наступних підпідрозділах детально розглянуто кожну з реалізованих можливостей розширення.

Для забезпечення динамічного поповнення колекції книг у мобільному застосунку було реалізовано інтеграцію із відкритим бібліотечним API – OpenLibrary. Це рішення дало змогу автоматично отримувати дані про книги за їхніми унікальними ISBN-кодами. OpenLibrary є вільною онлайн-базою даних, що містить інформацію про мільйони книг, включаючи метадані, такі як назви, автори, описи, обкладинки, а в окремих випадках і доступ до текстів або їхніх фрагментів. Використання OpenLibrary API дозволяє значно розширити

можливості додатку без необхідності попереднього локального завантаження великої кількості літератури.

Реалізація інтеграції передбачає наступну логіку: після сканування або введення ISBN-коду система спочатку виконує локальний пошук книги в базі даних. У разі відсутності відповідного запису мобільний застосунок надсилає HTTP-запит до API OpenLibrary для пошуку потрібної інформації. Отримані дані, такі як назва книги, ім'я автора та короткий опис, автоматично додаються до локальної бази без потреби ручного введення. Якщо у відповіді API доступний унікальний ідентифікатор OpenLibrary (ocaid), система додатково намагається завантажити текст книги або доступний фрагмент із ресурсу Archive.org. У випадку невдачі використовується спеціальний файл-заглушка для демонстрації функціоналу. Технічна взаємодія із OpenLibrary API реалізована через бібліотеку requests, що дозволяє надсилати запити та обробляти відповіді у форматі JSON.

Інтеграція з OpenLibrary API забезпечує можливість розширення колекції без попереднього завантаження всієї бази книг, підтримує динамічне оновлення даних і підвищує користувацький досвід завдяки автоматизації процесу додавання нової літератури. Завдяки цій інтеграції мобільний застосунок значно розширив власну базу книг, підвищивши гнучкість, актуальність і практичну цінність розробленої електронної бібліотеки.

Окрім інтеграції з відкритими джерелами, для спрощення процесу додавання нових книг у застосунку було реалізовано можливість сканування ISBN-кодів за допомогою камери мобільного пристрою. Ця функціональність дозволяє користувачу швидко і зручно знаходити й додавати книги без необхідності вручну вводити дані. Сканування ISBN реалізовано із використанням бібліотек OpenCV для доступу до камери та обробки відеопотоку в реальному часі, а також ruzbar для декодування штрихкодів і QR-кодів, зокрема кодів ISBN у форматі EAN-13.

Алгоритм роботи сканера передбачає, що користувач переходить на окремий екран сканування, де відкривається відеопотік із камери пристрою. На

екрані відображається спеціальна рамка та текстова підказка для правильного розміщення ISBN-коду. Кожен кадр аналізується в реальному часі для виявлення й декодування штрихкоду. Після успішного зчитування ISBN система здійснює пошук книги у локальній базі даних і, у разі відсутності запису, автоматично звертається до OpenLibrary для отримання інформації. Користувачу одразу надається можливість переглянути знайдену книгу або додати її до персональної бібліотеки.

Для підвищення зручності користування сканером реалізовано можливість виходу без сканування, додано текстові інструкції, а також підтримку сканування навіть за умов низької якості відеопотоку шляхом обробки декількох кадрів. Особливістю реалізації є автоматичне закриття сканера після успішного зчитування та миттєва обробка даних без потреби у ручному введенні ISBN-коду. Також передбачена можливість подальшого розширення функціоналу для роботи із QR-кодами чи іншими типами ідентифікації книг.

Впровадження сканування ISBN-кодів значно підвищило рівень зручності використання застосунку, скоротивши кількість кроків, необхідних для додавання нової книги, що відповідає сучасним стандартам мобільних рішень у сфері цифрового контенту.

3.8 Кросплатформенність програмного засобу

У процесі розроблення електронної бібліотеки було закладено принцип кросплатформенності, що дозволяє працювати із застосунком на різних операційних системах. Базовим середовищем розробки стала Windows 11, де проводилась основна реалізація функціоналу та тестування. Окрім того, здійснювалась перевірка працездатності застосунку в середовищі Linux Mint, де він успішно працював без необхідності внесення змін у кодову базу.

Для забезпечення можливості використання на мобільних пристроях була виконана збірка програми у форматі APK для операційної системи Android із

використанням Buildozer та python-for-android. Застосунок був адаптований під мобільні обмеження і протестований у середовищі Android-пристрою.

Також було підготовлено процес побудови iOS-версії за допомогою інструменту Kivy-ios, який дозволяє створювати проекти для екосистеми Apple. Збірка застосунку для iOS є можливою за умови наявності офіційної ліцензії розробника Apple.

Оскільки демонстрація розробленої електронної бібліотеки планується проводитись на десктопній платформі, де застосунок має максимальну функціональність та зручність використання, публікація окремих мобільних версій наразі не є необхідною. Тим не менш, архітектура проєкту дозволяє у подальшому легко адаптувати його для повноцінного використання на мобільних пристроях.

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було проаналізовано сучасні технології розробки мобільних застосунків, зокрема кросплатформні рішення з використанням мови програмування Python. Особливу увагу приділено фреймворку Kivy, який забезпечує можливість створення додатків для Android, iOS, а також десктопних платформ, що дозволяє суттєво знизити витрати на розробку та підтримку.

Вивчено архітектуру та особливості використання Kivy у контексті побудови електронної бібліотеки. Досліджено можливості створення адаптивного інтерфейсу користувача, реалізації базових функцій взаємодії з книгами, авторизації, персоналізації та збереження даних на мобільному пристрої.

У процесі роботи спроектовано архітектуру мобільного застосунку з чітким розділенням компонентів: інтерфейс, логіка, база даних. Реалізовано основні функціональні елементи системи: екран реєстрації та входу, каталог електронних книг, інтерфейс перегляду, персональна бібліотека користувача, а також модулі для пошуку та фільтрації даних.

У розробленому додатку реалізовано повноцінну функціональність електронної бібліотеки. Зокрема, додано підтримку живого пошуку, фільтрацію за жанрами, відображення лише персональної бібліотеки, а також сканування ISBN-кодів із автоматичним імпортом метаданих книг через API OpenLibrary.

Оптимізовано процес розробки та виконано тестування мобільного застосунку на платформах Windows та Linux. Створено інсталяційний пакет (APK) для Android. Підготовлено інструкції для компіляції iOS-версії з використанням Kivy-ios. Така реалізація демонструє кросплатформність рішення та його адаптованість до різних типів пристроїв.

Розроблений застосунок може бути використаний як основа для подальших освітніх або практичних проєктів, пов'язаних із розробкою електронних бібліотек, мобільних систем керування інформацією або

персоналізованих додатків для збереження та читання цифрових книг. Його архітектура дозволяє масштабування і розширення функціональності, зокрема шляхом додавання штучного інтелекту для рекомендацій або інтеграції з онлайн-сервісами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. State-of-the-Art cross-platform mobile application development frameworks: a comparative study of market and developer trends. MDPI. URL: <https://www.mdpi.com/2227-9709/12/2/45> (дата звернення: 03.03.2025).
2. Flutter - Build apps for any screen. Flutter - Build apps for any screen. URL: <https://flutter.dev> (дата звернення: 04.03.2025).
3. React Native · Learn once, write anywhere. React Native · Learn once, write anywhere. URL: <https://reactnative.dev> (дата звернення: 04.03.2025).
4. The six most popular cross-platform app development frameworks | kotlin multiplatform development. Kotlin Multiplatform Development Help. URL: <https://www.jetbrains.com/help/kotlin-multiplatform-dev/cross-platform-frameworks.html> (дата звернення: 05.03.2025).
5. Comparative analysis of kivy and other mobile development platforms | international journal of scientific research and management (IJSRM). International Journal of Scientific Research and Management (IJSRM). URL: <https://ijsrm.net/index.php/ijsrm/article/view/5421> (дата звернення: 08.03.2025).
6. Створення мобільних додатків з Python. DevZone. URL: <https://devzone.org.ua/post/ctvorennia-mobilnykh-dodatkov-z-python> (дата звернення: 10.03.2025).
7. TIOBE Index TIOBE. TIOBE. URL: <https://www.tiobe.com/tiobe-index/> (дата звернення: 15.03.2025).
8. Top computer languages 2025 - statisticstimes.com. StatisticsTimes.com | Collection of Statistics and charts. URL: <https://statisticstimes.com/tech/top-computer-languages.php> (дата звернення: 1.04.2025).
9. Write once. Deploy everywhere. – BeeWare. URL: <https://beeware.org> (дата звернення: 3.04.2025).
10. Завацький Д. Створення мобільного додатка на Python - Wezom. IT-компанія повного циклу розробки програмних продуктів WEZOM - Київ,

Україна. URL: <https://wezom.com.ua/ua/blog/sozдание-mobilnogo-prilozhenija-na-python> (дата звернення: 4.04.2025).

11. Kivy: cross-platform python framework for NUI. Kivy: Cross-platform Python Framework for GUI apps Development. URL: <https://kivy.org> (дата звернення: 6.04.2025).

12. Introduction Kivy 2.3.1 documentation. Kivy: Cross-platform Python Framework for GUI apps Development. URL: <https://kivy.org/doc/stable/gettingstarted/intro.html> (дата звернення: 7.04.2025).

13. Guide to app architecture | App architecture | Android Developers. Android Developers. URL: <https://developer.android.com/topic/architecture> (дата звернення: 10.04.2025).

14. SQLite home page. SQLite Home Page. URL: <https://www.sqlite.org> (дата звернення: 11.04.2025).

15. Крупенко К. Мобільний додаток української цифрової бібліотеки можна завантажити на Google Play. Спільне Культура. URL: <https://suspilne.media/culture/193784-mobilnij-dodatok-ukrainskoi-cifrovoi-biblioteki-mozna-zavantaziti-na-google-rlay/> (дата звернення: 12.04.2025).

16. Sqlite3. DB-API 2.0 interface for SQLite databases. Python documentation. URL: <https://docs.python.org/3/library/sqlite3.html> (дата звернення: 14.04.2025).

17. Design and implementation of digital library | atlantis press. Atlantis Press | Atlantis Press Open Access Publisher Scientific Technical Medical Proceedings Journals Books. URL: <https://www.atlantis-press.com/proceedings/icemc-17/25881939> (дата звернення: 15.04.2025).

18. Mobile apps–based applications in libraries and information centers: a systematic review of the literature and future research agendas | international journal of librarianship. International Journal of Librarianship. URL: <https://journal.calaijol.org/index.php/ijol/article/view/294> (дата звернення: 18.04.2025).

ДОДАТКИ

Додаток А

Структура бази даних

У застосунку використовується вбудована база даних SQLite з такими основними таблицями:

1. Таблиця users

- id: INTEGER PRIMARY KEY
- username: TEXT (унікальне ім'я користувача)
- password: TEXT (зашифрований пароль)

2. Таблиця books

- id: INTEGER PRIMARY KEY
- title: TEXT (назва книги)
- author: TEXT (автор книги)
- description: TEXT (опис)
- filepath: TEXT (шлях до файлу)
- isbn: TEXT (опціонально)

3. Таблиця user_books

- id: INTEGER PRIMARY KEY
- user_id: INTEGER (ідентифікатор користувача)
- book_id: INTEGER (ідентифікатор книги)

Додаток В

Файли конфігурації середовища та складання застосунку

Фрагмент файлу requirements.txt

```
kivy==2.3.0
requests
Pillow
pyzbar
opencv-python
```

Фрагмент buildozer.spec (ключові налаштування):

```
source.include_exts = py,png,jpg,kv,atlas
requirements = python3,kivy,requests,opencv-python,pyzbar,Pillow
android.permissions = CAMERA, INTERNET
android.minapi = 21
android.target = 30
android.archs = arm64-v8a, armeabi-v7a
```