UNIVERSITY MALAYSIA TERENGGANU
FACULTY OF OCEAN ENGINEERING TECHNOLOGY & INFORMATICS

CSM3114
Framework Based Mobile Application Development

Project Submission
PROJECT 1
‘BOOKVAULT: BOOK REGISTRATION APPLICATION’

Prepared by:
Muhamad Arif Ajmal bin Afsanisham
S61658

Prepared for:
Dr. Mohamad Nor bin Hassan

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING)

WITH HONOURS

SEMESTER I 2023/2024

Table of Contents

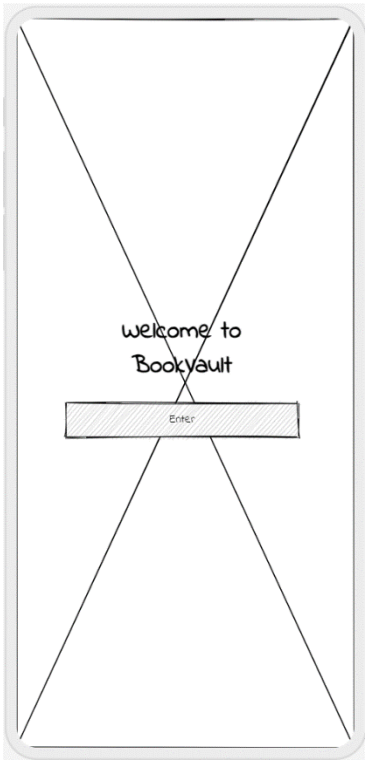
1 Summary of Prototype

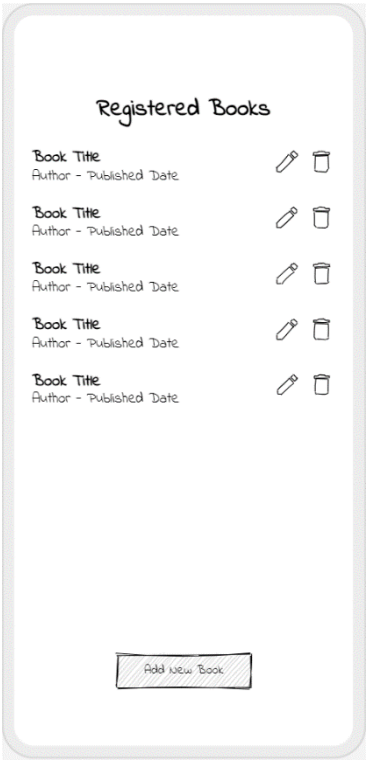
The Book Registration App, also known as "BookVault," is a mobile application designed to make it easy to manage library book collection. It is built on the Flutter framework, which allows for a consistent user experience across platforms. BookVault provides a user-friendly interface that makes it easy to register, edit, and view details of the books. User can input various book details, such as title, author, ISBN, publication date, edition, genre, synopsis, and shelf number. This ensures that the book library is organized in a structured way.

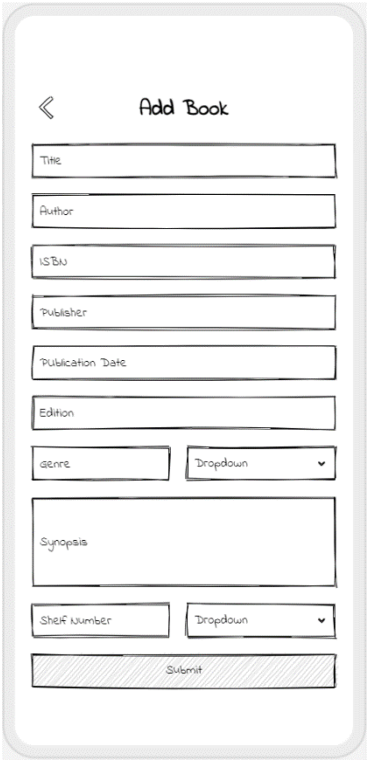
BookVault presents a comprehensive list of registered books. This makes it easy to navigate through the entries, edit information, and access complete book details. The app's welcome screen introduces the user to the platform and provides a seamless entry point to the book registration interface. Its visually appealing design, coupled with stylized components and intuitive navigation, ensures an engaging user experience.

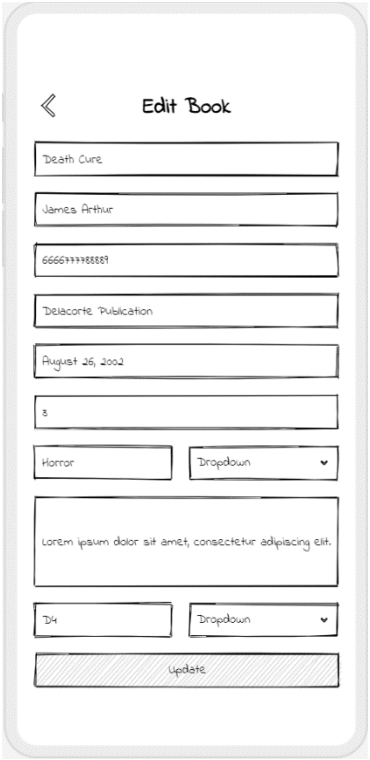
BookVault aims to simplify book collection management for enthusiasts. It offers a visually appealing and user-friendly platform for organizing, accessing, and exploring book library. With its current functionalities and future prospects, BookVault is positioned to become an indispensable tool for library management seeking a convenient way to manage their literary treasures.

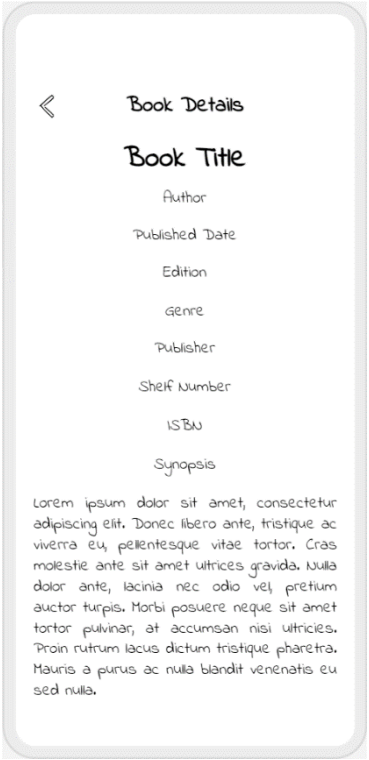
2 Prototype Design











3 Application User Interface

Figure 3.1

Figure 3.2

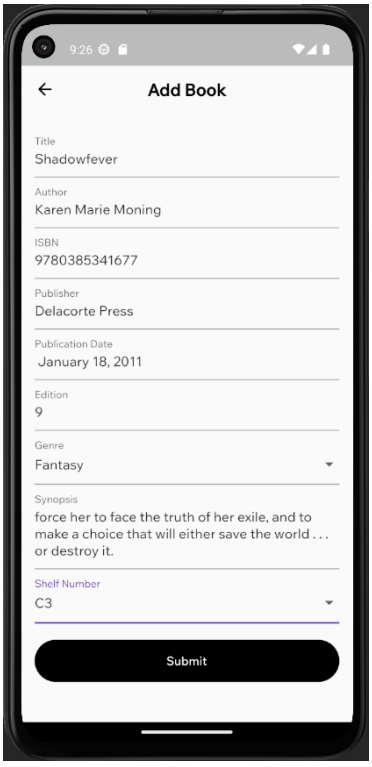


Figure 3.3

BookVault's application comprises of three core screens, one of them is Figure 3.1 which represents the welcome screen, serving as the initial interface upon app launch. This screen warmly greets users with a welcoming sentence, BookVault's branding, and an engaging background image. The inclusion of an 'Enter' button prompts users to transition to the subsequent screen seamlessly. Figure 3.2 illustrates the primary screen known as the 'Registered Books' screen, serving as the central hub for managing the user's book collection. Positioned at the bottom of the screen is a prominent floating action button, providing easy access for users to add new books to their library. Figure 3.3 showcases the 'Add Book' screen, facilitating the input of detailed book information. This screen incorporates various forms and dropdown menus, enabling users to enter essential book details like title, ISBN, publisher, and more. Notably, a 'Submit' button positioned at the screen's bottom finalizes the user's input, confirming the information's accuracy and readiness for inclusion in the 'Registered Books' screen's book list.

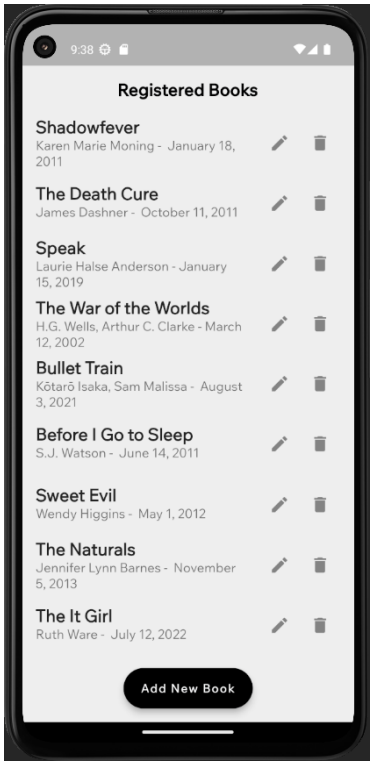


Figure 3.4

Figure 3.5

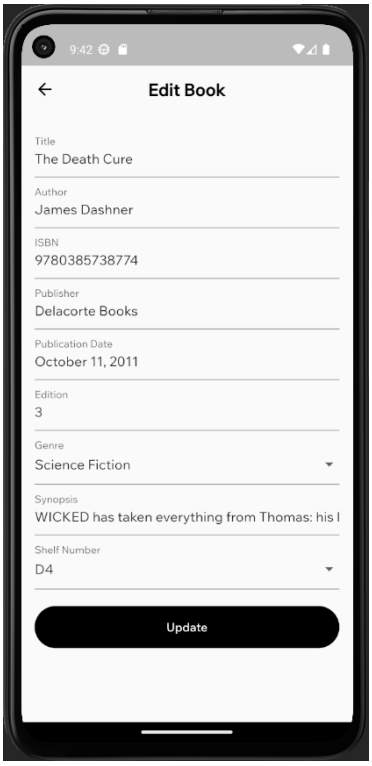


Figure 3.6

Figure 3.4 portrays the comprehensive 'Registered Books' list, showcasing books inputted by users. This catalogue displays vital book details such as the title, author, and publication date for each entry. Moreover, each book entry features convenient 'Edit' and 'Delete' buttons, empowering users to manage their library efficiently. The persistent 'Add New Book' floating action button remains accessible at the screen's bottom, facilitating seamless addition of new books to the list. Figure 3.5 represents the 'Book Details' screen, offering users a comprehensive view of the detailed information pertaining to a selected book. This screen consolidates all previously entered properties including edition number, genre, publisher details, and the book's synopsis, providing a comprehensive overview for users. Figure 3.6 highlights the 'Edit Book' screen, initiated when users engage the 'Edit' button aligned with their respective book entry on the 'Registered Books' screen. This interactive interface allows users to modify previously entered book information, updating it with new or revised details. Upon completion of edits, users can tap the 'Update' button positioned at the screen's bottom. Subsequently, the revised information seamlessly reflects on the 'Registered Books' list, ensuring accurate and up-to-date book details for the user's library.

4 Commercial Value and Pricing Potential of the Prototype

Market analysis reveals a substantial demand for book cataloging applications. BookVault's unique value proposition includes its ability to accommodate various user types, ranging from individual book collectors to institutional libraries. The app's competitive edge lies in its versatile functionality, allowing users to store and manage extensive book details effortlessly. Leveraging feedback from beta testing and user surveys, the app has demonstrated its appeal among diverse demographics, indicating promising market potential. Moreover, the prototype's adaptability and scalability position it favorably for expansion and integration with e-commerce platforms, potentially opening new revenue streams.

To optimize user adoption and ensure sustainability, multiple pricing models are under consideration. A freemium model, offering basic functionalities for free with premium features at a cost, appeals to individual users. Meanwhile, a subscription-based model tailored for institutional users, coupled with tiered pricing based on collection size, is envisaged for libraries. The pricing strategy is devised considering competitors' offerings, user willingness to pay, and perceived value. Extensive market research and user feedback inform the pricing structures, aiming to strike a balance between affordability and value.

5 Lesson Learned

The development process in Flutter revealed several critical lessons essential for building efficient and user-friendly applications. Understanding the fundamental concept of widget composition was pivotal, as it allowed the construction of intricate user interfaces by assembling basic widgets. Effective state management through mechanisms like 'setState' ensured real-time updates upon data changes, while the navigation system utilizing 'Navigator' facilitated seamless transitions between screens. Leveraging 'ListView.builder' demonstrated efficient rendering of dynamic content and scrollable widgets, maintaining performance even with extensive data. Additionally, handling user inputs using 'TextFormField' highlighted the importance of managing form fields and user interactions. Implementing consistent styling and theming through 'TextStyle' and 'ElevatedButton.styleFrom' contributed to a visually cohesive interface. Moreover, addressing scenarios like overflow issues in the Book Details screen emphasized the need for error handling and ensuring a smooth user experience.

6 Conclusion

In conclusion, the development journey through Flutter for the Book Registration App showcased the platform's robustness in building cross-platform, feature-rich applications. The project not only provided a comprehensive understanding of Flutter's widget-based architecture but also emphasized the significance of efficient state management, responsive UI design, and navigation flows. It highlighted the importance of responsive design principles and handling dynamic data with widgets like ‘ListView.builder’ and ‘TextFormField’ for a user-friendly experience. Additionally, the project underscored the need for meticulous error handling, especially with long texts causing overflow issues. Overall, this experience solidified the prowess of Flutter in facilitating rapid app development, enabling the creation of engaging, scalable, and aesthetically pleasing mobile applications.

7 Reference

1. Anurag. (2023, October 30). *Library Management System | Library management software in India*. SkoolBeep. https://www.skoolbeep.com/blog/library-management-system/
2. Bienvenu, M. (2017). LIBRARY MANAGEMENT SYSTEM. *www.academia.edu*. https://www.academia.edu/33632232/LIBRARY_MANAGEMENT_SYSTEM
3. *ButtonStyle class - material library - Dart API*. (n.d.). https://api.flutter.dev/flutter/material/ButtonStyle-class.html
4. Darji, P. (2024a, January 1). *Change floating action button color in flutter | Ultimate Guide*. FlutterBeads. https://www.flutterbeads.com/change-floating-action-button-color-in-flutter/
5. Darji, P. (2024b, January 1). *Customize TextField/TextFormField Border in flutter*. FlutterBeads. https://www.flutterbeads.com/textfield-textformfield-border-in-flutter/
6. *ElevatedButton class - material library - Dart API*. (n.d.). https://api.flutter.dev/flutter/material/ElevatedButton-class.html
7. *Expanded class - widgets library - Dart API*. (n.d.). https://api.flutter.dev/flutter/widgets/Expanded-class.html
8. *ExpansionPanelList class - material library - Dart API*. (n.d.). https://api.flutter.dev/flutter/material/ExpansionPanelList-class.html
9. *FloatingActionButton class - material library - Dart API*. (n.d.-a). https://api.flutter.dev/flutter/material/FloatingActionButton-class.html
10. *FloatingActionButton class - material library - Dart API*. (n.d.-b). https://api.flutter.dev/flutter/material/FloatingActionButton-class.html
11. *FontWeight class - dart:ui library - Dart API*. (n.d.). https://api.flutter.dev/flutter/dart-ui/FontWeight-class.html

12. GeeksforGeeks. (2020, October 21). *Flutter Using Google fonts*. https://www.geeksforgeeks.org/flutter-using-google-fonts/
13. *how to assign hexadecimal color code in primarySwatch in flutter?* (n.d.). Stack Overflow. https://stackoverflow.com/questions/62432229/how-to-assign-hexadecimal-color-code-in-primaryswatch-in-flutter
14. *How to change the appBar back button color*. (n.d.). Stack Overflow. https://stackoverflow.com/questions/51508257/how-to-change-the-appbar-back-button-color
15. *Interesting synopsis books*. (n.d.). https://www.goodreads.com/shelf/show/interesting-synopsis
16. *Make AppBar transparent and show background image which is set to whole screen*. (n.d.). Stack Overflow. https://stackoverflow.com/questions/53080186/make-appbar-transparent-and-show-background-image-which-is-set-to-whole-screen
17. *Make Extended FAB(FloatingActionButton) fluid-width with flutter*. (n.d.). Stack Overflow. https://stackoverflow.com/questions/75472255/make-extended-fabfloatingactionbutton-fluid-width-with-flutter
18. *Passing data between screens in Flutter*. (n.d.). Stack Overflow. https://stackoverflow.com/questions/53861302/passing-data-between-screens-in-flutter
19. *Styling widgets*. (n.d.). Flutter. https://docs.flutter.dev/ui/widgets/styling
20. Suragch. (2023, November 14). A visual guide to Input Decorations for Flutter TextField. *Medium*. https://medium.com/flutter-community/a-visual-guide-to-input-decorations-for-flutter-textfield-706cf1877e25

8 Code Submission

https://github.com/arifajmal/FrameworkMobileDevelopement.git