

IT2143 Visual Computing

Group Project

Group M1

University Library Management System

Group Members:

No	Registration No	Name
1	2020/ICT/19	K.A.D.S.D.Kandanaarachchi
2	2020/ICT/18	I.M.C.Jeewantha
3	2020/ICT/93	R.P.Jayathunga
4	2020/ICT/67	S.P.Malalarathna
5	2020/ICT/75	Hanzul Yeenas

Contents

1. Introduction	3
2. Objectives	4
3. Methodology.....	5
I. Requirement Gathering.....	5
II. Tools and Technologies	5
4. Implementation.....	7
I. Interface Design.....	7
II. Database	10
III. Output.....	10
5. Conclusion	11
6. References	12

1. Introduction

A university library management system (ULMS) is a software application that helps libraries manage their collections, patrons, and operations. ULMSs can be used to track books, journals, and other materials, as well as to automate tasks such as circulation, cataloging, and acquisitions. ULMSs can also provide a variety of services to patrons, such as searching, renewals, and holds. In addition, ULMSs can generate reports that can be used to track library usage and performance.

There are many benefits to using a ULMS. Some of the most important benefits include:

- Improved efficiency
- Increased accuracy
- Improved patron service
- Better decision-making

There are two main types of ULMSs: integrated and modular. Integrated ULMSs combine all of the functions of a library management system into a single package. Modular ULMSs allow libraries to choose the modules that they need.

Implementing a ULMS can be a complex process. It is important to carefully consider the needs of the library and to choose a ULMS that is a good fit. It is also important to involve staff in the implementation process and to provide adequate training.

2. Objectives

Improve efficiency and automation:

Automate tasks like borrowing, returning, renewing, searching, and reporting to save time and effort for both librarians and users.

Enhance resource management:

Track and manage books, journals, electronic resources, and other materials effectively, including acquisitions, cataloging, and inventory control.

Optimize user experience:

Provide a user-friendly interface for searching, browsing, requesting, and managing resources. Enable mobile access for on-the-go convenience.

Generate informative reports:

Provide reports on library usage, resource availability, overdue items, fines, and other relevant data for informed decision-making.

Integrate with existing systems:

Ensure smooth integration with university systems like student information systems and authentication services.

User management:

Manage user accounts, track borrowing history, and send notifications for overdue items and reservations. Offer self-service options for account management.

Resource management:

Efficiently catalogue and track physical and electronic resources, including e-books, journals, databases, and multimedia. Facilitate resource acquisition and budget planning.

Search and discovery:

Enable advanced search features with filters and facets to improve resource discovery. Recommend resources based on user interests and previous borrows.

3. Methodology

I. Requirement Gathering

- a. Interviews:
 - i. Conduct individual or group interviews with key stakeholders such as Librarians (circulation, acquisitions, cataloging, reference), Library administrator, Faculty members, Students and IT staff
- b. Observation:
 - i. Observe library operations firsthand identify pain points, bottlenecks, and areas for improvement.
- c. Document Analysis:
 - i. Review existing library documents such as Policies and procedures, Manuals and training materials, Reports and usage statistics and gain in-depth understanding of current processes and requirements.
- d. Questionnaires and Surveys:
 - i. Distribute questionnaires or online surveys to gather input from a wider audience.
- e. Prototyping:
 - i. Develop early prototypes or mockups to demonstrate potential features and workflows.
- f. Use Cases and Scenarios:
 - i. Develop detailed use cases and scenarios to describe how users will interact with the system

II. Tools and Technologies

Development Environment:

Visual Studio: Provides a comprehensive IDE for C# development.

Programming Language:

C#: Object-oriented language well-suited for building robust and scalable applications.

Database Management:

SQL: Structured Query Language for managing relational databases.

User Interface Design:

Figma: Collaborative design tool for creating UI prototypes and mockups.

Version Control:

GitHub

4. Implementation

- I. Our team successfully implemented core functionalities of the university library management system, including user registration, login, book searching, borrowing, and returning, as well as staff-level capabilities for adding, removing, and managing books and users. Visual Studio served as the primary development environment for C# coding and UI creation, while Figma facilitated initial UI design and workflow visualization. A SQL database with an .MDF file structure effectively stores and manages library data, ensuring seamless accessibility. Despite encountering minor challenges during specific function implementations, the team achieved a user-friendly interface with a simple design and intuitive navigation. The system demonstrates satisfactory performance under moderate data loads and exhibits potential for supporting future user growth.

II. Interface Design

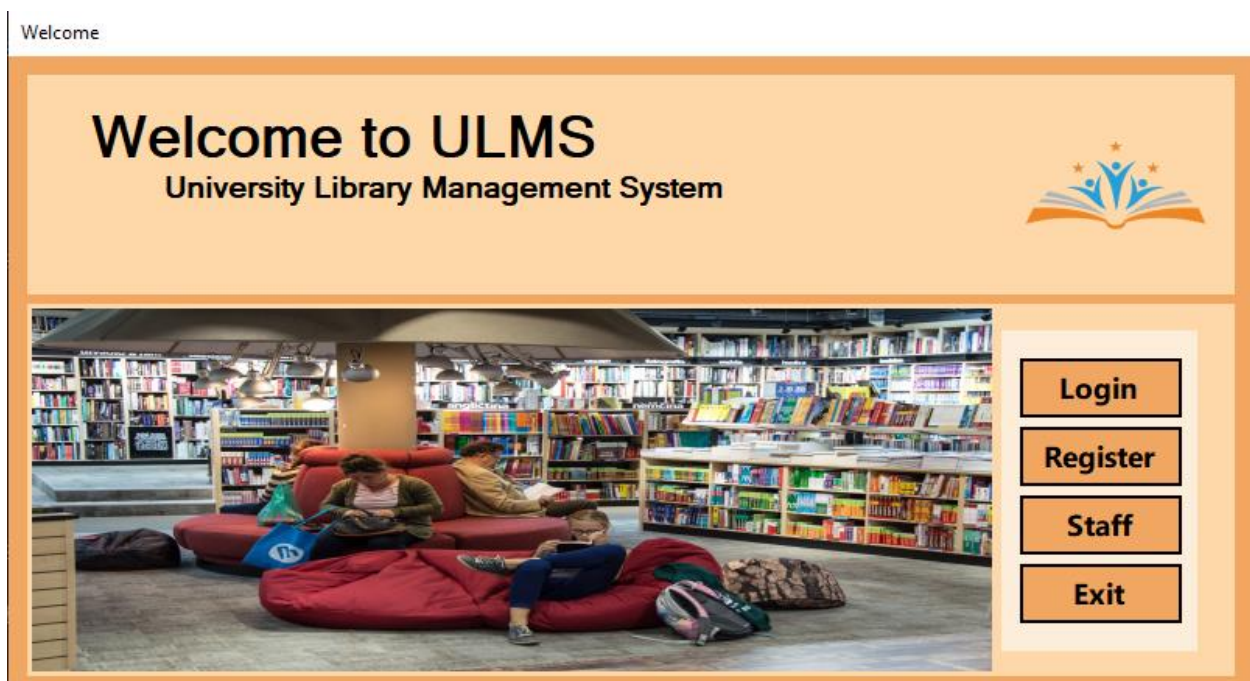


Figure 1.1: Welcome Page

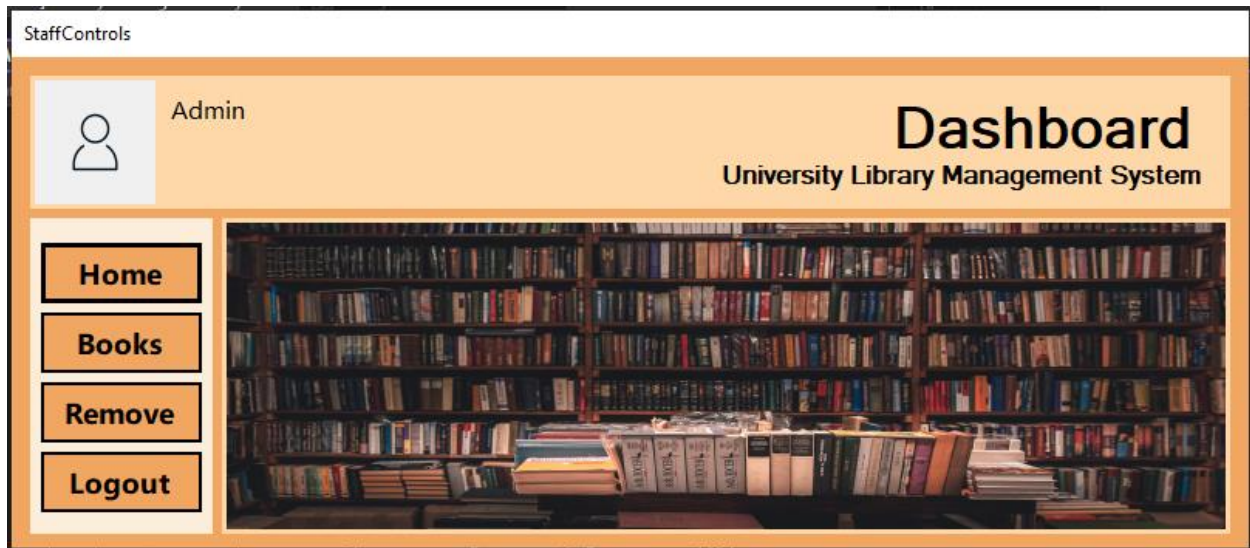


Figure 1.2: Staff Dashboard

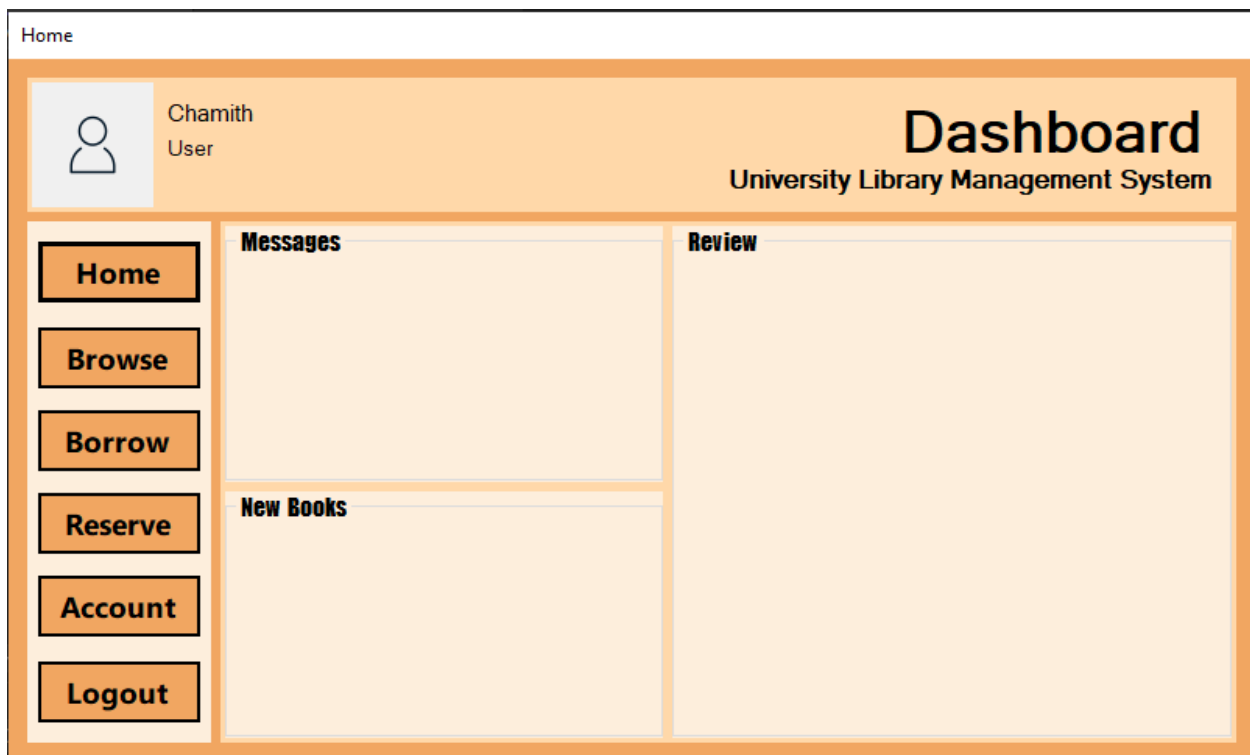


Figure 13: User Dashboard

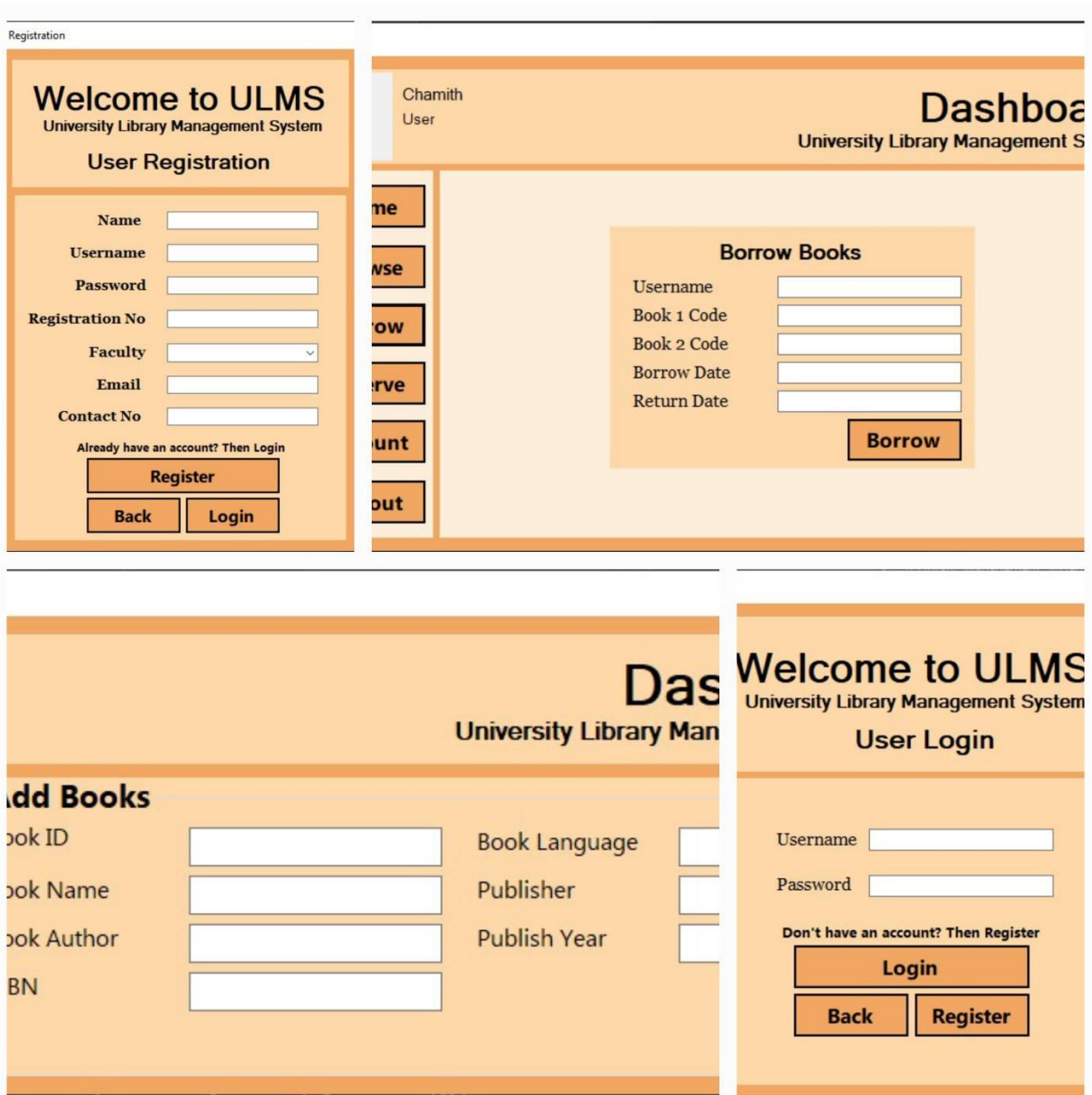


Figure 1.4: UI Interfaces

III. Database

The foundation of this library management system lies in a robust SQL database stored within an .MDF file. Three vital tables – users, books, and borrow – meticulously organize library data, ensuring efficient access and management. This structure facilitates functionalities like user registration, book tracking, and borrowing/returning processes, seamlessly supporting the system's core operations. Although some implementation challenges arose, the chosen structure exhibits potential for scalability and future data growth, paving the way for a comprehensive and reliable library solution.

IV. Output

University Library Management System(ULMS) stands as a testament to technological innovation, delivering impeccable user experience and enhanced usability. It seamlessly integrates with existing library infrastructures and operates as a scalable and adaptable system catering to diverse needs. ULMS goes beyond basic functionality, empowering librarians with data-driven insights for optimized resource allocation and user engagement. It fosters accessibility by providing remote access, digital resources, and a vibrant community platform. Furthermore, its cost-efficiency and future-proof design ensure sustainability and continuous innovation, making ULMS the cornerstone of a modern and thriving library landscape.

Table 4.1: User Table

Name	Username	Password	RegNo	Faculty	Email	ContactNo

BookName	BookID	Author	ISBN	Publisher	PublishYear	Language	Availability

Username	BookID1	BookID2	BorrowDate	ReturnDate

5. Conclusion

In conclusion, the University Library Management System (ULMS) represents a transformative leap forward for academic information access and management. Not just a functional tool, ULMS embodies a holistic vision, seamlessly blending intuitive user experience with robust data-driven insights. Beyond mere resource organization, it fosters connection and engagement, creating a vibrant hub for intellectual exploration and exchange. More than a technological marvel, ULMS stands as a catalyst for lifelong learning, empowering both students and librarians to navigate the ever-evolving landscape of knowledge and information. Its adaptable and scalable design ensures its adaptability to future needs, making it a cornerstone of academic excellence for generations to come. Therefore, ULMS isn't simply a solution; it's an invitation to step into the future of learning, where knowledge is not just stored, but experienced, shared, and transformed.

6. References

[1] Stack Overflow. Available online: <https://stackoverflow.com/> (Accessed December 24, 2023).

[2] C# Corner. Available online: <https://www.c-sharpcorner.com/chapters/> (Accessed December 25, 2023).

[3] Microsoft Docs. "Create a SQL database by using a designer." Available online: <https://m.youtube.com/watch?v=5nTlqgQLkIc> (Accessed December 20, 2023).

[4] Copy Programming. "Inserting Values into Database .mdf in C#." Available online: <https://stackoverflow.com/questions/32450651/how-to-copy-mdf-file-into-a-folder-with-c-sharp> (Accessed December 20, 2023).

[5] Microsoft Docs. "Add new connections." Available online: <https://medium.com/@andiaspin/how-to-create-connection-with-sql-server-in-visual-studio-ca6075b69515> (Accessed December 20, 2023).