MINI PROJECT – IS2106

LibraryMate - Library Management System
Group 25

Department of Computing & Information Systems

Faculty of Computing

Sabaragamuwa University of Sri Lanka

Prepared by

21CIS0217 - B.M.H.N.Ariyarathna

21CIS0213 - D.B.K.Jayathunga

21CIS0085 – K.P.B.Perera

Team Code Crusher

31st March,2024

1.0. INTRODUCTION

1.1 Purpose

This document outlines the requirements for developing a public library management system. It provides guidance for project design, development and testing.

1.2. Intended Audience

The intended audience here includes:
- System developers
- Project managers
- Users of the library management system such as librarians and students.

1.3 Scope

The project facilitates library operations including user authentication, student registration, book management, book ordering, periodic retrieval of books and communication between librarians and students in the library system.

1.4 Definitions, abbreviations and acronyms

SRS   : Software Requirements Specification

DBMS : Database Management System

UI   : User Interface

API   : Application Programming Interface

2.0 OVERALL DESCRIPTION

2.1 Product Perspective

The public library management system is accessible through standard web browsers. It interacts with a back-end database to store and retrieve data related to users, books, orders and deadlines. The system will be developed, scaled and maintained using modern web technologies.

2.2 Product Functions

The system will provide the following functionalities

- User Authentication and Authorization
- Registration of students
- Book catalog management
- Book ordering and order management
- Last date reminder for books received
- Communication between librarians and students

2.3 User Characteristics

The machine two number one person roles:

Students : users who browse, order and borrow books from the library.

Librarians : Users answerable for library management, which includes e book inventories, orders and closing dates.

2.4 Assumptions and Dependencies

Assumptions

The system calls for that:

- Users ought to have a contemporary internet browser and an internet connection.
- The database management machine (DBMS) used to keep the statistics is to be had and well configured.
- The hardware and infrastructure used within the website hosting of the web software is in vicinity.

Dependencies

- Implementation is dependent more on external elements. This is essentially because the gadget will rely on a number of outside API's or external libraries for functionality along with person authentication and sending e mail notifications among others.
- Stakeholders ought to additionally be able to quick provide remarks and to work in the improvement of the software.

3.0. SYSTEM FEATURES AND REQUIREMENTS

3.1 Functional Requirements

Functional requirements describe the specific tasks the system must perform to meet user needs. It includes:

- User authentication
- Student registration
- Book cataloging and management
- Order placement and management
- Notification of last date for delivery of books
- Communication functions

3.2 Non-Functional Requirements

Non-functional requirements indicate scenarios that can be used to test system functionality.

It includes:

- Performance   : The system should handle a large number of users efficiently.

- Security   : A robust security process should be followed to protect user data.
- Reliability   : The system should be reliable with reduced downtime and continuous data connectivity.
- Usability   : User interfaces should be intuitive and easy to navigate.

3.3 User Interfaces

This web system is designed to suit the needs of librarians and users. The interface is simple and user-friendly. Searching, ordering, and finding the due dates for books can be easily managed.

3.4 Software Interfaces

The system works with data systems such as database management systems to store and retrieve data. Works with library data systems and master data systems for user identification and email notifications.

4.0. SYSTEM DESIGN

4.1 Product Features

1. **User Authentication and Authorization:**
   Secure logins must exist for librarians and students accessing the system, ensuring that only authorized users can act in their roles.

2. **Student Registration:**
   The opportunity for students to register in the system, provide information required for library use, functionality of things like system contact information and student identification.

3. **Book Catalog Management:**
   Collecting new books for the library system, updating the information on existing books, managing the library, including book classification to update books.

4. **Order Placement and Management:**
   The feature enables users to create orders for the books they want and librarians can manage these requests, track their status and facilitate the delivery process.

5. **Notification System:**
   By setting up a notification system that reminds users of the upcoming due dates for borrowed books, they return their books on time and eliminate overdue fines.

6. **Communication Tools:**
   Integration of communication tools for librarians and students to exchange questions, help requests, and announcements within the information system.
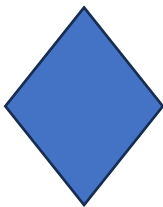
4.2 Product Environment

Entity-Relationship Diagram (ERD): Before physically creating a data warehouse, a necessary plan must be prepared. The plan prepared can then be used to physically create a data file. Thus, a graphical model used in creating a design to store data is known as an Entity-Relationship Diagram.

Symbols:

Entity

Relationship

Attribute

The Entity - Relationship Diagram (ERD) below shows the entities and their relationships within the Library Management System.

Relationships:

**User-Student**: One-to-One relationship indicating a user can be a student.

User-Librarian: One-to-One relationship indicating a user can be a librarian.

Student-Order: One-to-Many relationship indicating a student can have multiple orders.

Librarian-Order: One-to-Many relationship indicating a librarian can manage multiple orders.

Book-Order: One-to-Many relationship indicating multiple books can be part of multiple orders.

ER Diagram
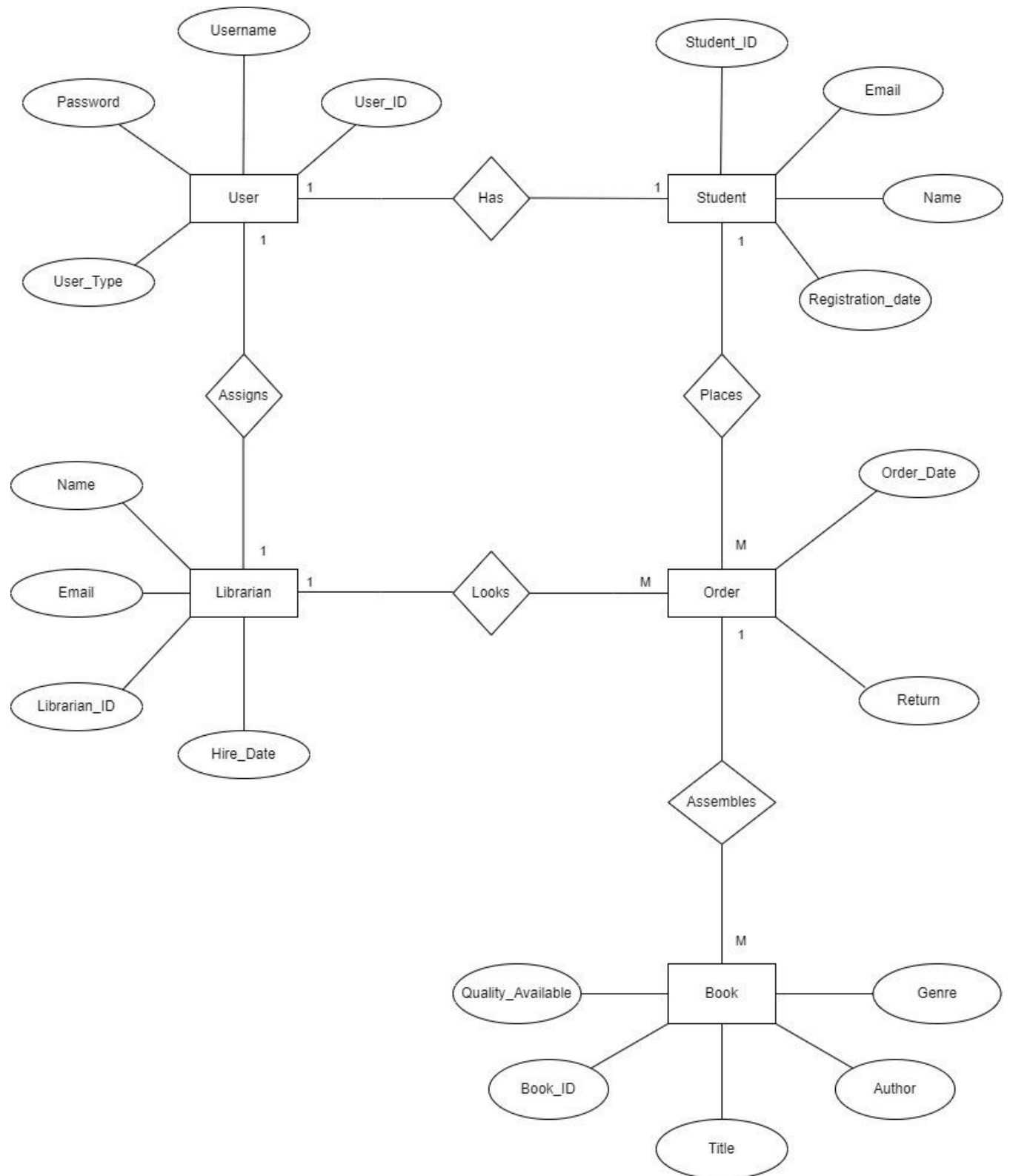
Entities:

- **User**

  Attributes: User_ID (Primary Key), Username, Password, User_Type

- **Student**

  Attributes: Student_ID (Primary Key), User_ID (Foreign Key), Name, Email, Registration_Date

- **Librarian**

  Attributes: Librarian_ID (Primary Key), User_ID (Foreign Key), Name, Email, Hire_Date

- **Book**

  Attributes: Book_ID (Primary Key), Title, Author, Genre, Quantity_Available

- **Order**

  Attributes: Order_ID (Primary Key), User_ID (Foreign Key), Book_ID (Foreign Key), Orde_rDate, Return_Date

4.3 Use cases

Use Scenario 1: To register as a student in the library system.

1. The student arrives at the library's registration page.
2. The student fills out the registration form and provides their name, email, desired username and password.
3. The system verifies the authenticity of the username and email and validates the entered information.
4. Once validated the system will create a new entry in the user table with a unique user ID along with username, password and user type provided as "Student".
5. The system creates a new entry with a student ID in the student table. It links to the user ID generated in the previous step and stores the student's name, email and registration date.

Use scenario 2: To add a new book to the library's collection.

1. The librarian moves to the "Add New Book" section of the system.
2. The librarian fills in the required information for the new book, including its title, author, genre, and basic quantity available.
3. The system validates the entered information.
4. Once validated the system creates a new entry in the book table with a BookID. Stores the provided book details.

Diagram
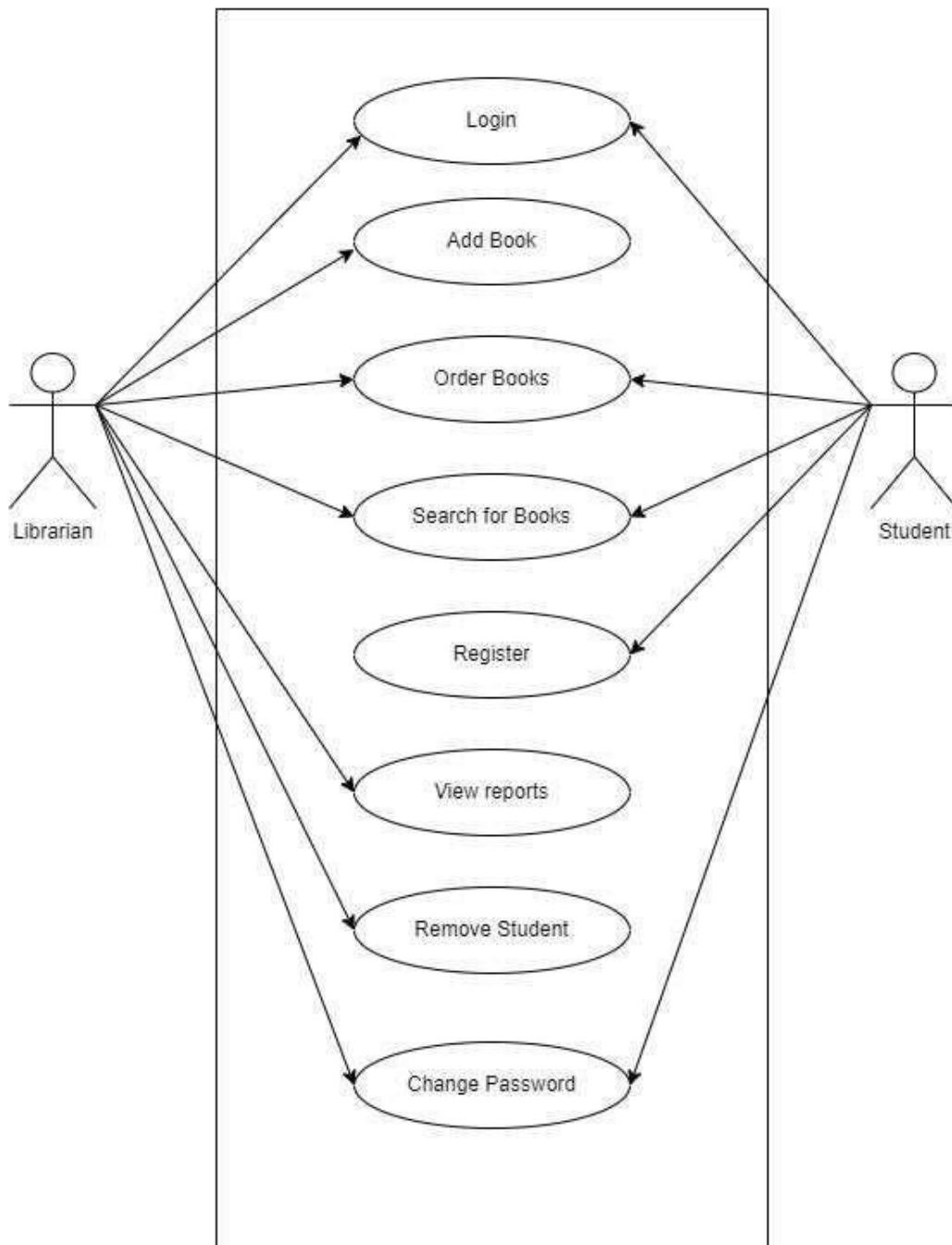
4.4 Database Tables

• **User Table:**

Columns: UserID (Primary Key), UserName, Password, UserType

Attribute	Data Type	Description
User_ID #	int(15)	Library users ID
Username	varchar(200)	Library users name
Password	varchar(15)	Library users login password
User_Type	varchar(200)	Library users type (Student or Librarian)

• **Student Table**

Columns: StudentID (Primary Key), UserID (Foreign Key), Name, Email, RegistrationDate

Attribute	Data Type	Description
Student_ID #	int(15)	Student ID
User_ID	int(15)	Library users ID
Name	varchar(200)	Student name
Email	varchar(30)	Student personal email address
Registration_Date	date	Student registration date

12

- **Librarian Table**

Columns: LibrarianID (Primary Key), UserID (Foreign Key), Name, Email, HireDate

Attribute	Data Type	Description
Librarian_ID #	int(15)	Librarian ID
User_ID	int(15)	Library users ID
Name	varchar(200)	Librarian name
Email	varchar(30)	Librariant personal email address
Hire_Date	date	Book issue date from librarian

- **Book Table**

Columns: Book_ID (Primary Key), Title, Author, Genre, Quantity_Available

Attribute	Data Type	Description
Book_ID	int(15)	Book ID
Title	varchar(200)	Book title
Author	varchar(200)	Book Author
Genre	varchar(200)	Book category
Quantity_Available	int(15)	Available book quantity

- **Order Table**

Columns: Order_ID (Primary Key), User_ID (Foreign Key), Book_ID (Foreign Key), Order_Date, Return_Date

Attribute	Data Type	Description
Order_ID #	int(15)	Book order ID
User_ID	int(15)	Library users ID
Book_ID	int(15)	Book ID
Order_Date	date	Book order date
Return_Date	date	Book return date

5.0 UI/UX DESIGN
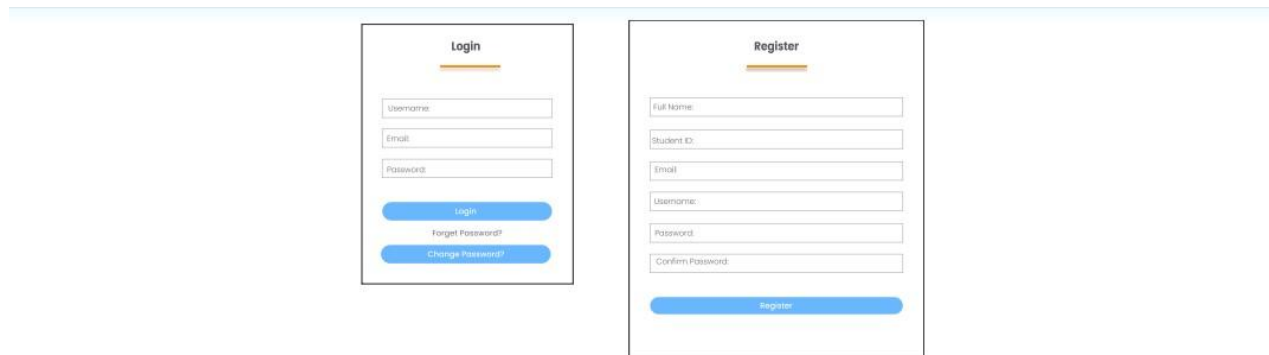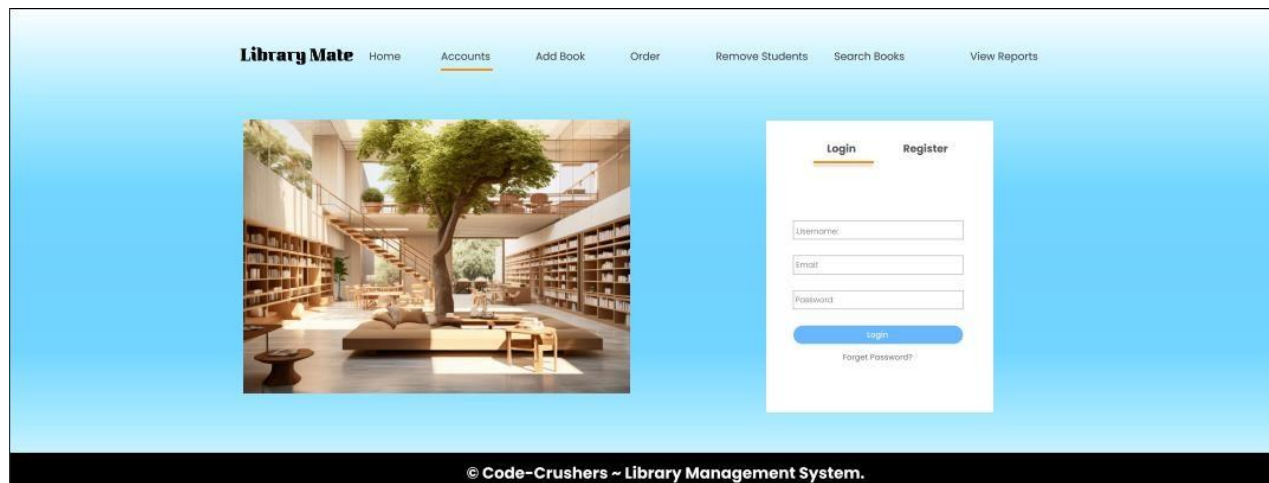
Change Password

Current Password:

New Password:

Confirm New Password:

Change Password

Library Mate Home Accounts Add Book Order Remove Students Search Books View Reports

Add Books

Book Title:

Author:

Book ID:

Genre:

Quantity:
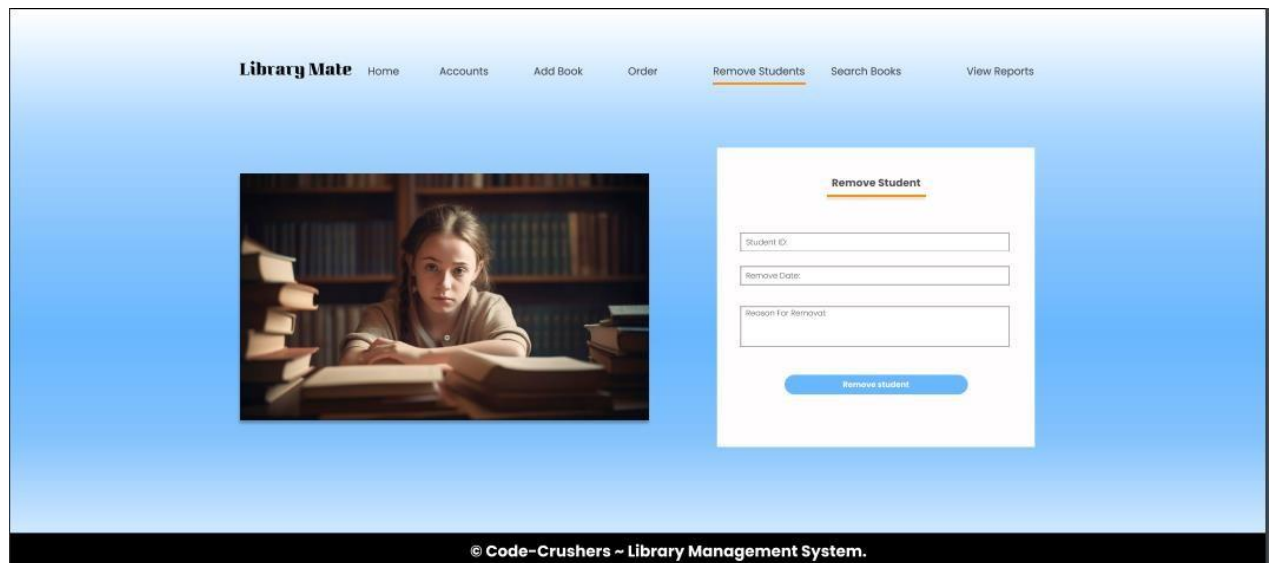
Add Date

Add Book

© Code-Crushers ~ Library Management System.

Library Mate Home   Accounts   Add Book   Order   Remove Students   Search Books   View Reports

Order Books

| Book Title 1: | Author: |
| Book Title 2: | Author: |

Order Date:

Return Date:

Additional Notes:

Place Order

© Code-Crushers ~ Library Management System.

Library Mate Home   Accounts   Add Book   Order   Remove Students   Search Books   View Reports

Remove Student

Student ID:

Remove Date:

Reason For Removal:

Remove student
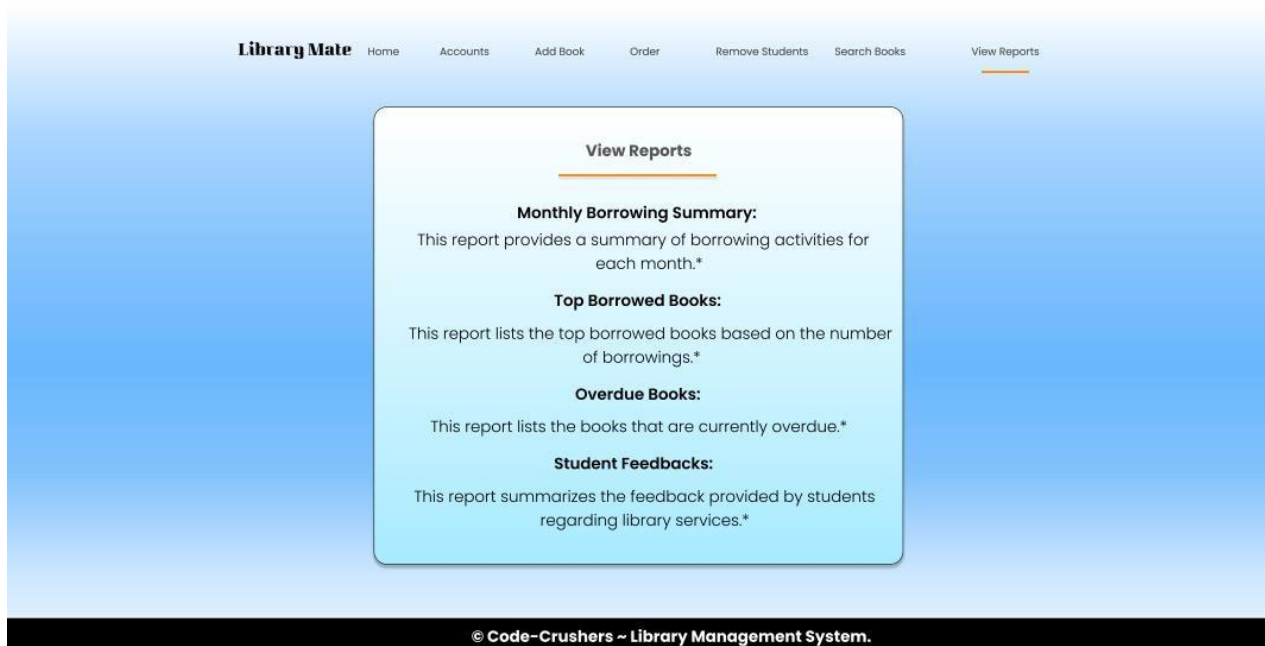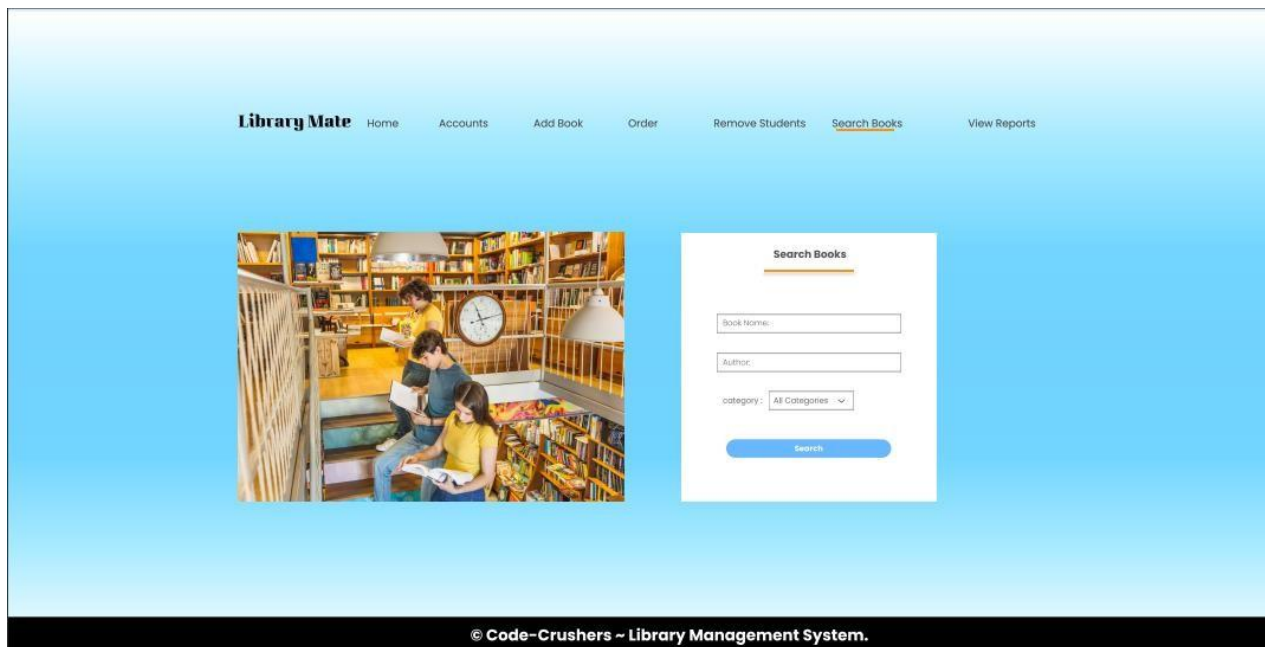
© Code-Crushers ~ Library Management System.

Figma Link

https://www.figma.com/file/YSH4E2ZKTvjfYbXPoLjc6G/Library-
managementSystem.?type=design&node-
id=0%3A1&mode=design&t=qy3eNSmLbZmsHnf3-1