LIBRARY SYSTEM

[CSCI313 Project]

Amr Ahmed – 211000270
Adham Hossam – 211001699
Eyad Yehia - 211001231
Omar Elhawwari – 211002152

Table of Contents

0. Table of Contents.....

1. Introduction.....

1.1. Purpose.....

1.2. Project Scope .....

1.3. Used Technologies.....

1.4. Intended Audience.....

2. Overall Description.....

2.1. Product Perspective.....

2.1.1 Product Function.....

2.2. User Characteristics.....

2.3 Constraints.....

2.4 Assumptions and Dependencies.....

3. Functional Requirements.....

3.1. User Class 1: Admin.....

3.2. User Class 2: The User.....

4. Non-Functional Requirements.....

4.1. Reliability.....

4.2. Recoverability.....

4.3. Performance.....

4.4. Availability.....

4.5. Usability.....

4.6. Maintainability.....

5. Interface (UI/UX).....

5.1. System Interface.....

5.2. Software Interface.....

5.3 *Hardware Interface*.....

6. Diagrams.....

6.1. Use Case Diagram.....

6.1.1. Use Case Scenarios.....

6.2. Sequence Diagram.....

6.3. Class Diagram.....

7. Test Cases.....

7.1. Test Case: #1.....

7.2. Test Case: #2.....

7.3. Test Case: #3.....

7.4. Test Case: #4.....

7.5. Test Case: #5.....

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of a library system application. In this document, we will outline the application's features, the system's user interface, and its intended function, as well as detail the admin's panel, testcases, and performance.

1.2 Project Scope

What we aim to achieve in this project is to build a stable, very well-optimized library system with many features that are yet to be decided. Security and access control: there will be various accounts to log in to the system one for the client and the other one for the admin to modify or update anything within the system. Reporting and analytics: this part is for the administrator which is going to show how many books have been borrowed or bought. and finally, the user interface we are going to make sure that it's a functional, useful GUI with a lot of features to use in a creative style.

1.3 Used Technologies

Mainly we are using Java as a programming language to build this library system as a main tool to build the system. MySQL database is also used as a primary database set to save the data in it.

1.4 Intended Audience

We are targeting institutions, universities, or any library that needs a fully functional, well-optimized library system that satisfies their requirements.

2. Overall Description

2.1 Product Perspective

• The "Library System" will be a desktop application that provides a virtual library service for a real library.

• The application has two different modes:

1- Admin Mode
2- User Mode

both have different features and structures.

2.1.1 Product Function

• After Opening the application, a loading screen will be displayed. The next screen will be a screen to choose between modes or to register as a new user.

• Registration:

• If this was the user first time to use the application, a registration panel will be provided to create an account.

• The user will be requested to enter his E-mail, Username, Password & Phone Number.

• An OTP will be sent the registered E-mail to check if the E-mail is correct and the user is requested to enter the OTP to continue using the application.

• After completing registering, the user will be redirected to the modes panel.

• Admin:

• If the user decided to choose the Admin Mode, he will be requested to enter his admin username and password (Generated automatically by the owner).

• After signing-in, a dashboard contains the main objects will be shown.

• Main objects:

1- Add New Book.
2- Remove Book.
3- Show Borrowed Books & The Borrowers.
4- Hold Borrowing Period.
5- Accounts Details.
6- Logout.

• **User:**

• If the user decided to choose the User Mode, he will be requested to enter his username and password then the OTP for more security.

• If the user forgot his password, the "Forgot Password?" feature will be provided with an OTP for more security.

• After signing-in, a user panel contains services will be shown.

• User Panel Services:

1- Show All Books.
2- Borrow Book / Return Book.
3- User Information
4- Logout.

2.2. User Characteristics

• The user should have basic knowledge of how to use online systems & registration process.

• The user should have accessibility to internet connection to connect with the database.

2.3. Constraints

• The PC will have to be connected to internet in order for the application to work normally with the database, otherwise the user won't be able to access the application.

• Sign-in and password are a main require to identify users and admins.

2.4. Assumptions and Dependencies

• The application will be available for download as soon as the official website is published.

• The Application will require internet connection, so users are expected to have an accessible internet connection to use the application.

3. Functional Requirements

3.1. User Class 1: Admin

3.1.1. Functional Requirements

Title: Login

Description: System will allow the admin to login if a valid username and password were entered.

3.1.2. Functional Requirements

Title: Show Books

Description: System will allow the admin show all the books in the library.

3.1.3. Functional Requirements

Title: Add Books

Description: System will allow the admin to add a new book to the library.

Required Information:

- Category
- Name
- Author Name
- Publisher (Admin Username)

3.1.4. Functional Requirements

Title: Remove Book

Description: System will allow the admin to remove a book by the book ID.

3.1.6. Functional Requirements

Title: Borrowed Books

Description: System will allow the admin to show the borrowed books with the borrower username & ID and the return date.

3.1.7. Functional Requirements

Title: Close Borrowing Period / Open Borrowing Period

Description: System will allow the admin to stop books from borrowing for a while and re-open the borrowing method again.

3.1.8. Functional Requirements

Title: View Accounts Details

Description: System will allow the admin to show all registered users information.

3.1.9. Functional Requirements

Title: Logout

Description: System will allow the admin to logout from the system.

3.2. User Class 2: The User

3.2.1. Functional Requirements

Title: Registration

Description: System will allow the user to create an account so that they can login later.

Required Information:

- Username
- Password
- Email
- Phone number (Egyptian numbers only)

3.2.2. Functional Requirements

Title: Forgot Password?

Description: System will allow the user to reset the password by username & OTP that will be sent the registered email for more security.

3.2.3. Functional Requirements

Title: Login

Description: System will allow the user to login if a valid username and password were entered.

3.2.4. Functional Requirements

Title: Show books status

Description: System will allow the user to show the books status to know which book is available to borrow.

3.2.5. Functional Requirements

Title: Borrow Book

Description: System will allow the user to borrow a book (availability is a require).

3.2.6. Functional Requirements

Title: Return Book

Description: System will allow the user to return a borrowed book.

3.2.7. Functional Requirements

Title: User Panel

Description: System will allow the user to show his profile and his information.

Available Information:

- User Info:
  - Username
  - Email
  - ID

3.2.8. Functional Requirements

Title: Logout

Description: System will allow the user to logout from the library.

4. Non-Functional Requirements

4.1. Reliability

- The system is estimated to fail at a mean time of 2 times per month.

4.2. Recoverability

- System should be able to recover after breakdown within maximum 2 hours.

4.3. Performance

- Pages should not take more than 10 seconds to load.

4.4. Availability

- The system should be available 24/7, with a regular maintenance plan.

- In the event of scheduled maintenance, users should be notified in advance, and the system should provide a temporary offline mode with a user-friendly message.

4.5. Usability

- The user interface should be intuitive and user-friendly, catering to users with basic knowledge of online systems.

- The system should provide clear error messages and prompts to guide users in case of mistakes or incorrect inputs.

- The user will need a maximum of $\frac{1}{2}$ hour to be able to use the system.

4.6. Maintainability

- Code should be well-documented to facilitate easy maintenance and updates by other developers.

- The system should support seamless updates without causing disruptions to ongoing operations.

5. Interface

5.1. System Interface

Upon opening the application, users will encounter a loading page displayed prior to the application's initiation, as depicted in Figure 1. Users are presented with three sign-up options: login as an admin, login as a regular user, or register as a new user, as illustrated in Figure 2.

Opting to register as a user involves completing a form that includes fields for the username, password, email address, and phone number, as indicated in Figure 3. While registering, the system will automatically send an OTP for the provided email address to confirm registering with this email to enhance system security.

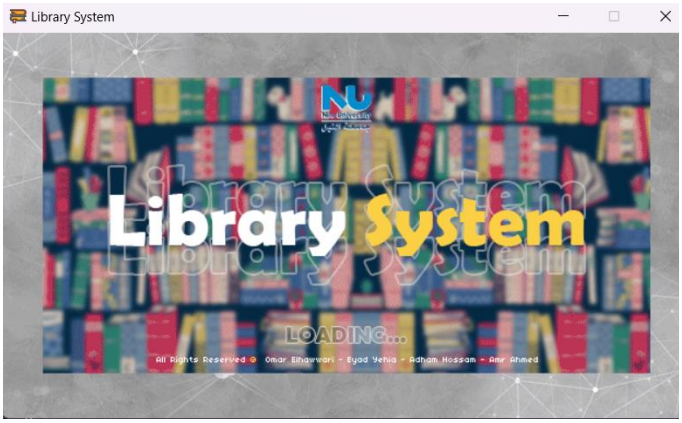


Figure 1: Loading Page

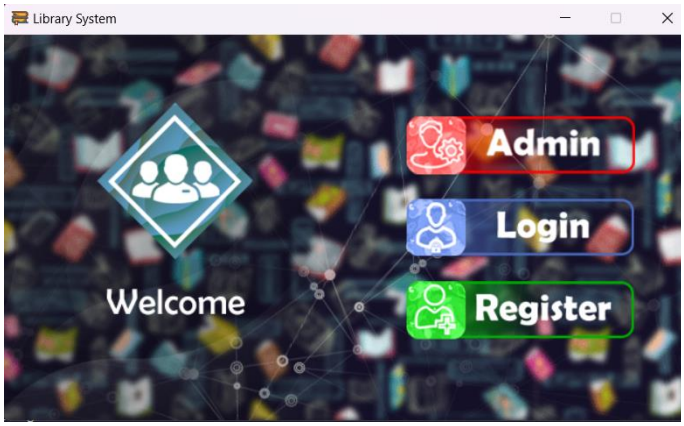


Figure 2: Modes Page

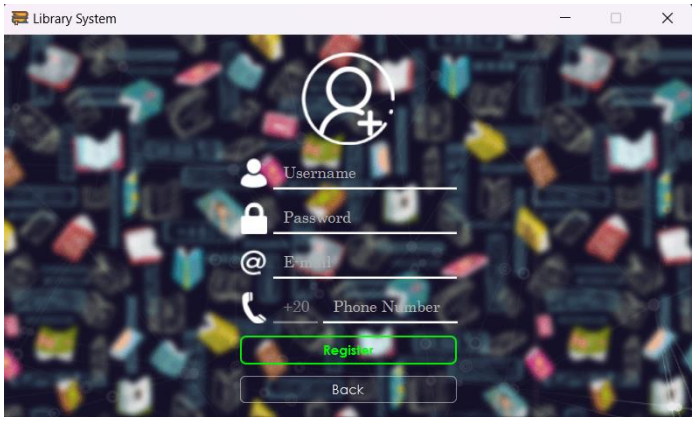


Figure 3: Register Page

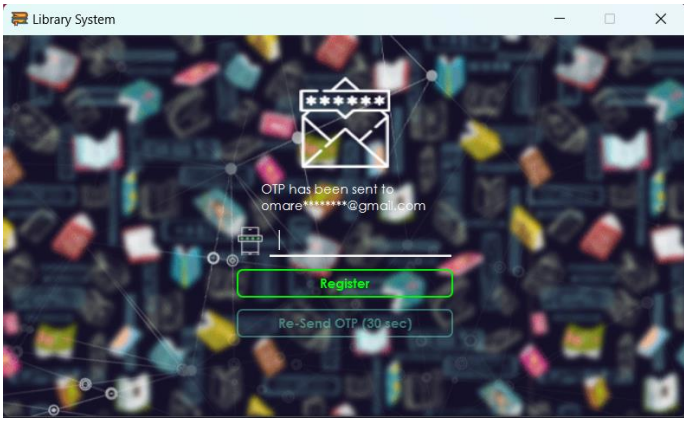


Figure 3.1: Register OTP Page

Upon successfully completing the registration process, the user gains the ability to access the system by logging in using the Login option depicted in Figure 2. Subsequently, the login form in Figure 4 is presented to the user. In the event that the user forgets their password, a feature has been implemented to address this scenario as in figures 4.1. Figure 5 presents a dashboard that encapsulates essential features available to the user upon logging in, including borrowing available books, returning borrowed books, and browsing the library's assortment to choose books for

borrowing. Additionally, users can view their own profiles, registered information, and activity within the application.

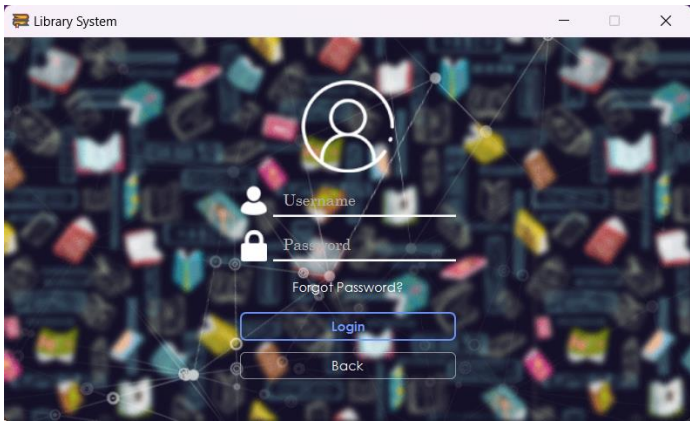


Figure 4: User Login

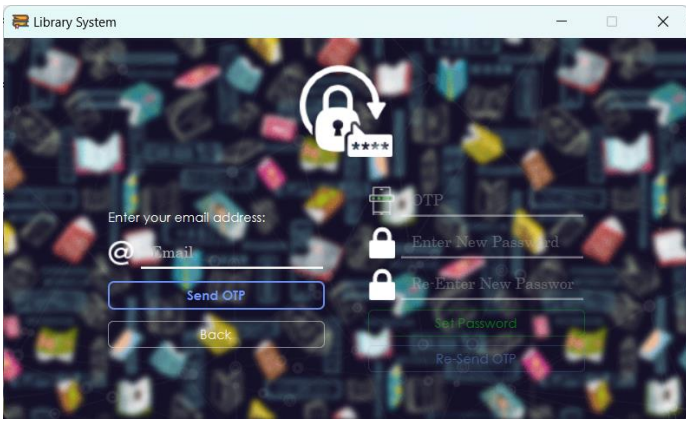


Figure 4.1: Forgot Password Page

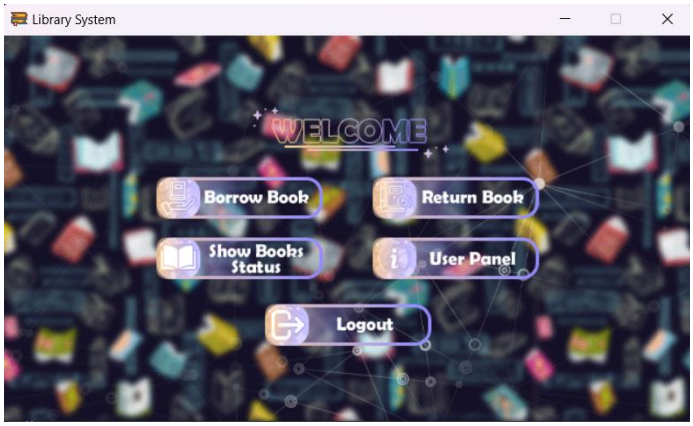


Figure 5: User Dashboard

The application includes a moderator account that allows administrators to update the system with necessary changes. To access these accounts, administrators should utilize the admin login option in Figure 2. As depicted in Figure 6, administrators must enter their provided username and password to log in and access the features of the moderator account.

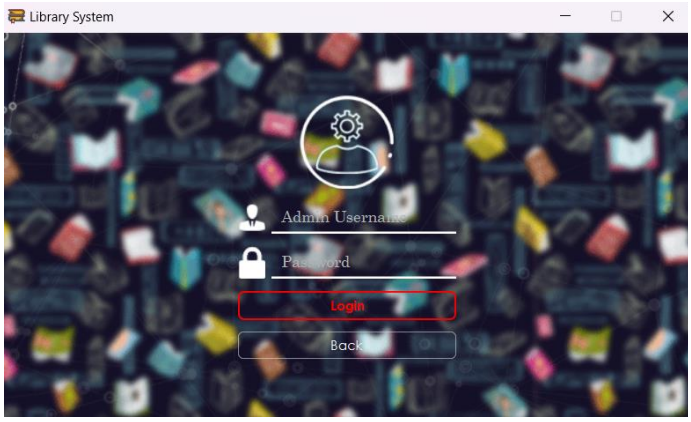


Figure 6: Admin Login

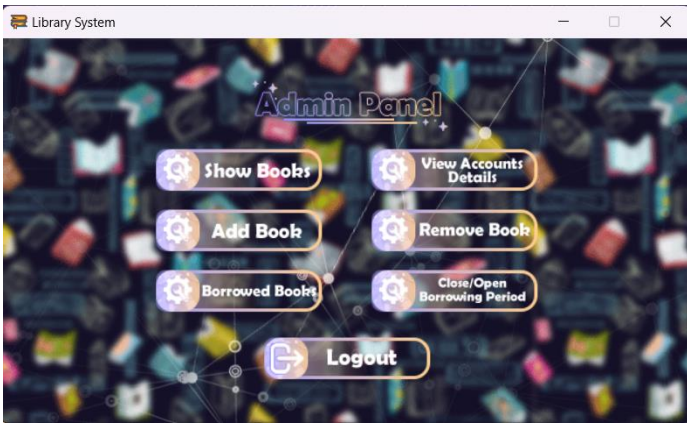


Figure 7: Admin Panel

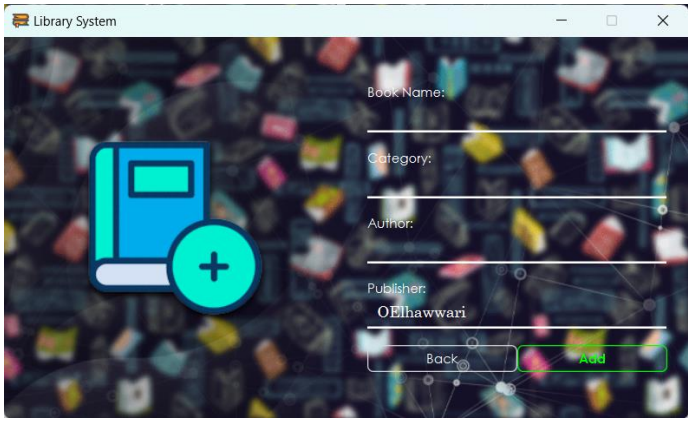


Figure 8: Add Book Feature (Admin)

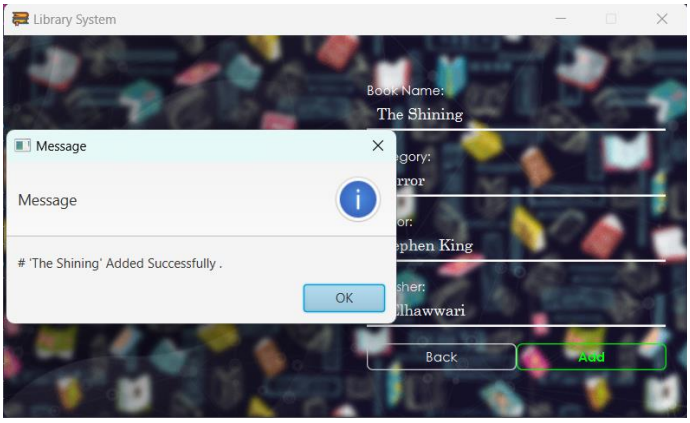


Figure 9: Book Added (Admin)

Upon successful admin login, a dashboard will be displayed, encompassing primary functions for the library system. These functions include adding books to the library, illustrated in Figures 8 and 9, removing books from the library as shown in Figure 10, displaying the current books in the library as depicted in Figure 11, showcasing currently borrowed books from the library as in Figure 11.1, and presenting information of registered users as seen in Figure 12.

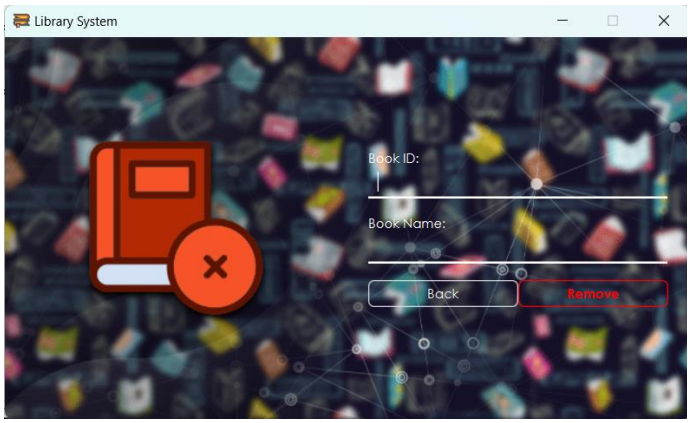


Figure 10: Remove Book Feature (Admin)

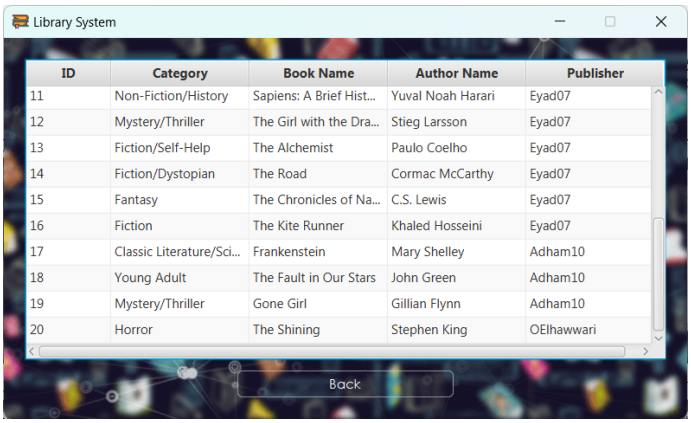


Figure 11: Show Books Feature (Admin)

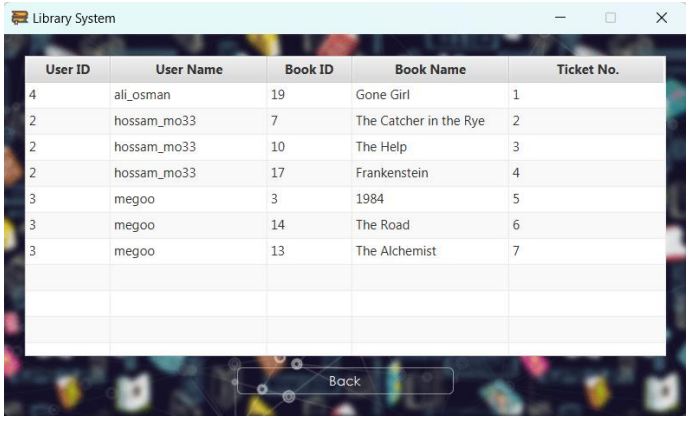


Figure 11.1: Show Borrowed Books Feature (Admin)

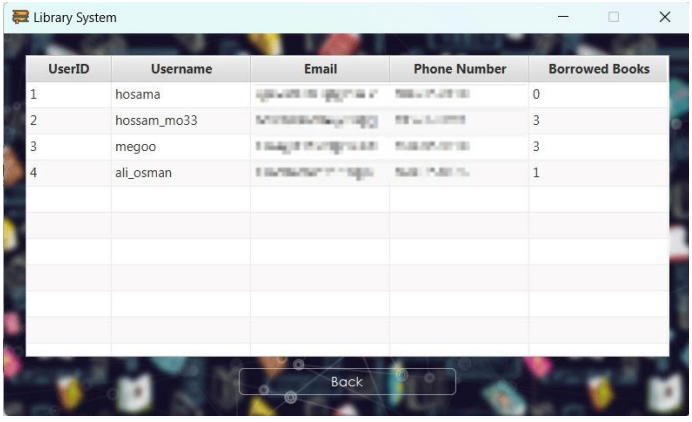


Figure 12: Show Accounts Feature (Admin)

In Figure 9, a book was successfully added to the library after entering the necessary information into the system, and the added book is displayed in Figure 11.

The admin has the capability to disable the borrowing option from the library if necessary, as illustrated in Figures 13 to 16. This feature can be valuable when all the books in the library are currently on loan.

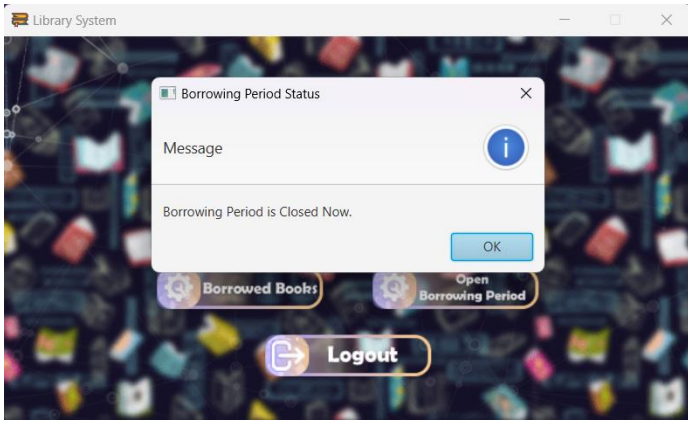


Figure 13: Closing Borrowing from Admin's pov

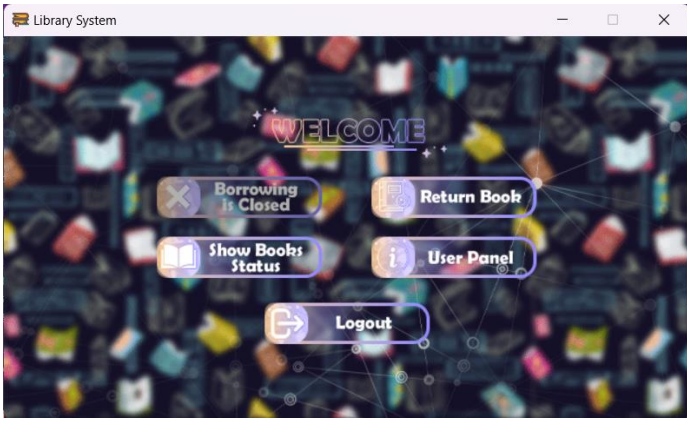


Figure 14: Closing Borrowing from User's pov

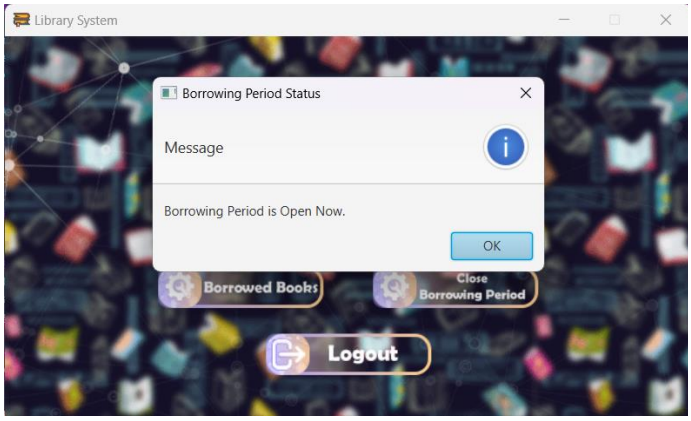


Figure 15: Open Borrowing from Admin's pov

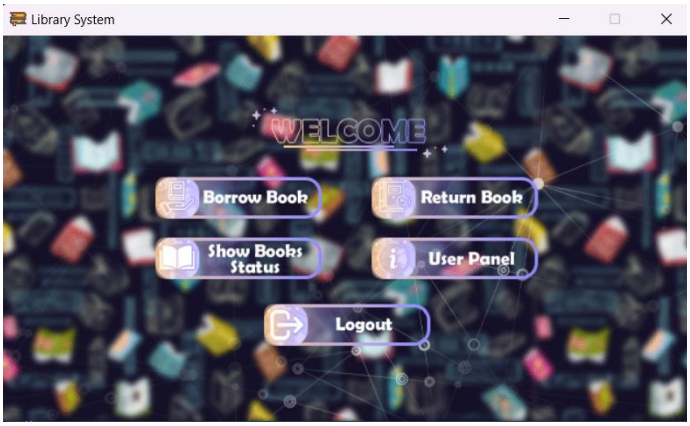


Figure 16: Closing Borrowing from User's pov

Additional features have been incorporated into the system; however, as for the timeline, it is still in the process of being developed and will be added as soon the update is done.

5.2. Software Interface

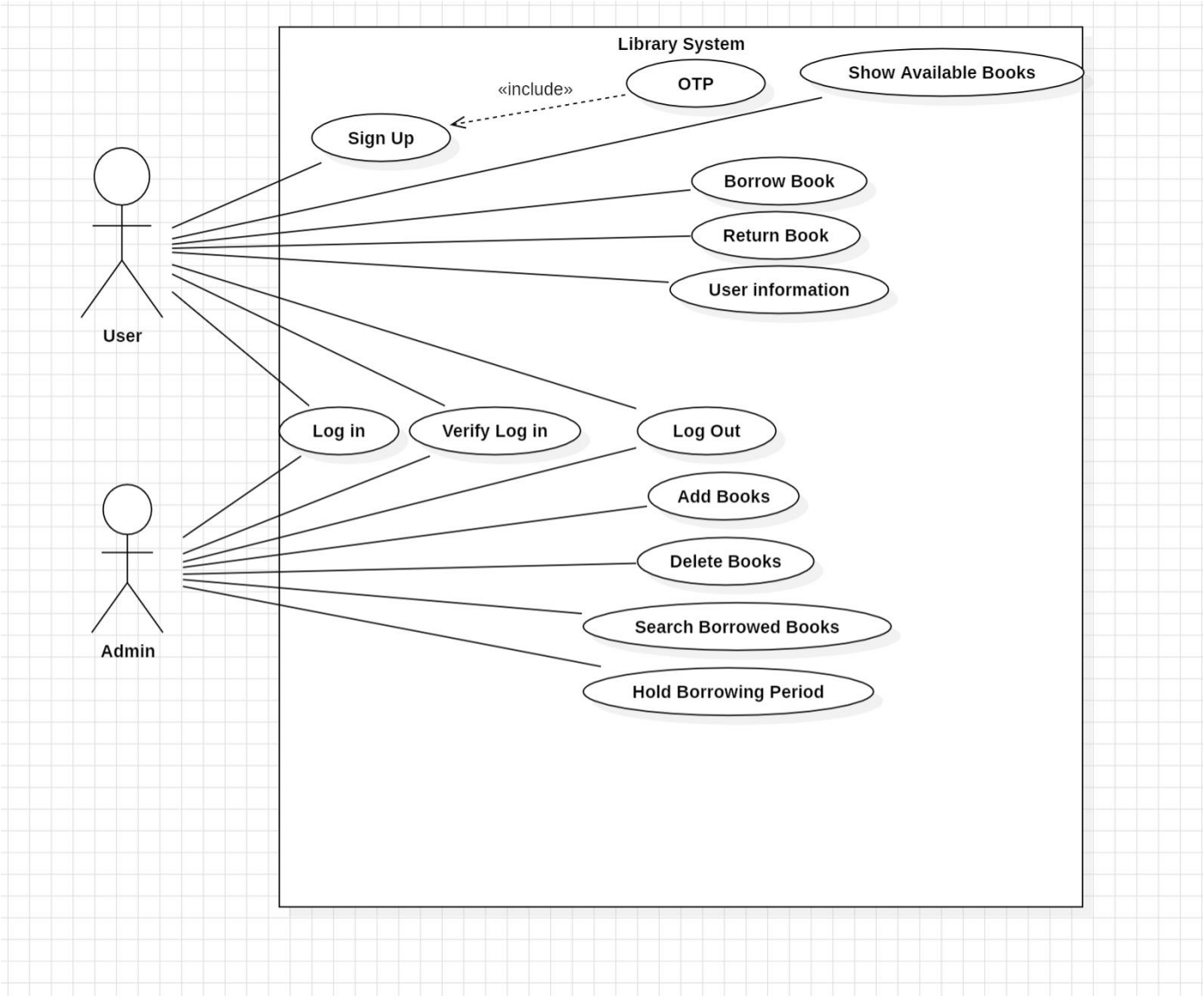
The library system application is a Java-based application developed using Visual Studio Code, Photoshop, incorporating various packages and libraries and using MySQL database. The application is designed to display datasets and facilitate interactions between users and the server/database, managing Outing data sent from the app and handling Incoming data received from the server and database.

5.3. Hardware Interface

An essential aspect is that the app needs an internet connection to operate seamlessly with the database.

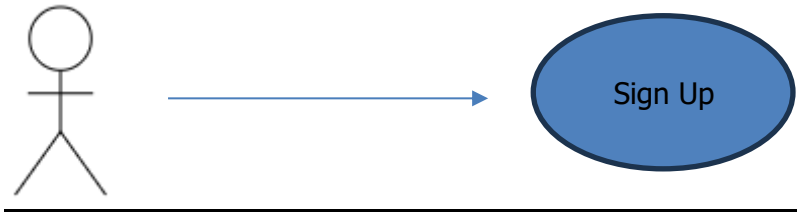
6. Diagrams

6.1 Use Case Diagram:



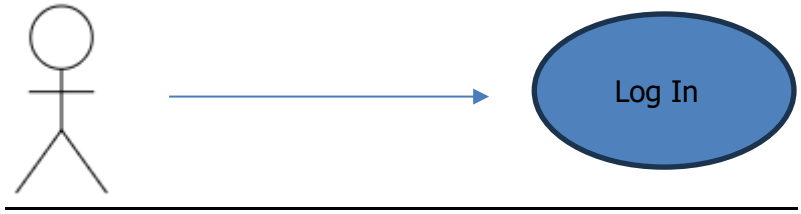
6.1.1 Use Case Scenarios

Use Case 1



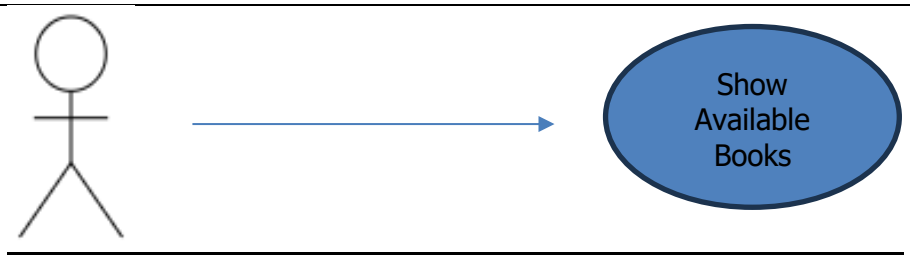
Usercase Name	**Sign-Up**
Actors	**User (Non-registered)**
Main Case Senario	**1. System asks the User to enter the email and password (if registered) or create a new account.** **2. User chooses to create a new account.** **3. System asks the User to give their information: first name, last name,password email (unregistered email), phone number.** **4. User gives the system the required information.** **5. User clicks on the sign up button.** **6. System checks that the email is not registered.** **7. System creates the account.**
Exception	**5a. User clicks on the sign up button without filling all of the required information.** **6a. Email is already registered**
Actions	**5.1. System display to the user an alert "Please fill in all the required fields".** **5.2. User fills the missing information.** **6.1. System display to the user "Email already registered".** **6.2. User enters another email.**
Pre-Condition	• **User had to download the application.**
Post Condition	• **Another user is added to the system.** • **User is successfully registered.** • **User information is stored in the database.**

Use Case 2



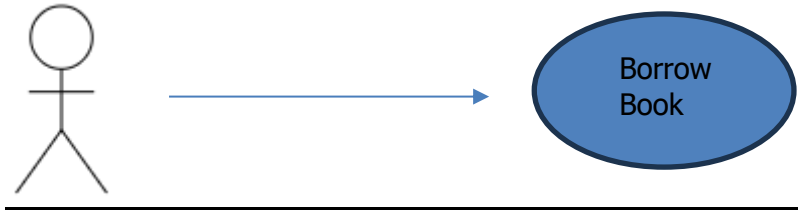
Usercase Name	**Log-In**
Actors	**User/Admin (Registered)**
Main Case Senario	**System asks the User/Admin to enter the email and password.** **User/Admin enters username and password.** **User clicks on the log in button.** **System checks that the email and password are valid.**
Exception	**2a. Email or password are not valid.** **3a. User/Admin clicks on the log in button without filling all of the required information.**
Actions	**2.1. System display to the User/Admin an alert "Email or Password are not valid".** **2.2. User/Admin enters the correct Email or password.** **3.1. System display to the User/Admin an alert "Please fill in all the required fields".** **3.2. User/Admin fills the missing information.**
Pre-Condition	**User had to download the application.** **User/Admin have to have an account**
Post Condition	**User/Admin are logged in to the system.** **User/Admin have access to the system functions.**

Use Case 3



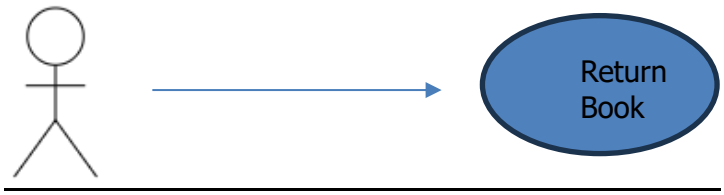
Usercase Name	**Show Available Books**
Actors	**User(Registered)**
Main Case Senario	**1.After Signing in user will use this function to display available books to borrow.** **2. All books will be displayed**
Exception	**1.There are no available books**
Actions	
Pre-Condition	**User had to download the application.** **User have to have an account**
Post-Condition	**User is logged in to the system.** **User have access to the system functions.**

Use Case 4



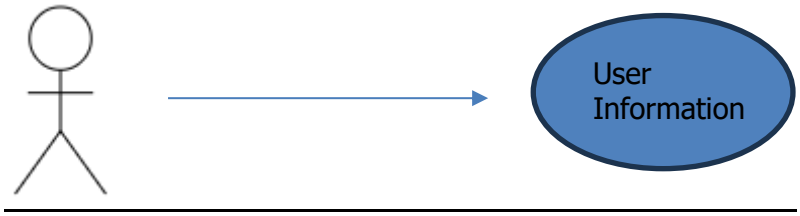
Usercase Name	**Borrowing Book**
Actors	**User (Registered)**
Main Case Senario	**1. User Search for the Book title** **2. User clicks on the Borrow button.** **3. User chooses the borrowing period (in days)**
Exception	
Actions	
Pre-Condition	• **User had to download the application.** • **User/Admin have to have an account**
Post Condition	• **Book is added to the Borrowed Books section for the student** • **Number of remaining books is decreased by 1** • **System Display the return date for the book**

Use Case 5



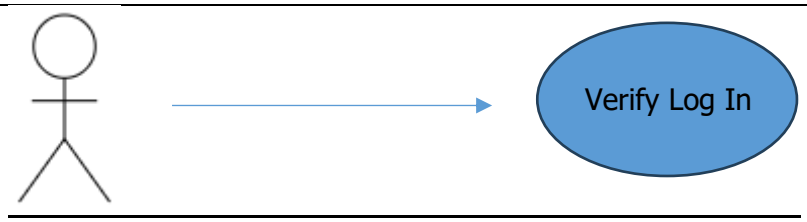
Usercase Name	**Return Book**
Actors	**User (Registered)**
Main Case Senario	**User Search for the Book title in the Borrowed Books section** **User clicks on the Return button.**
Exception	
Actions	
Pre-Condition	**User had to download the application.** **User/Admin have to have an account**
Post Condition	**Book is Removed from the Borrowed Books section for the student** **Number of remaining books is increased by 1**

Use Case 6



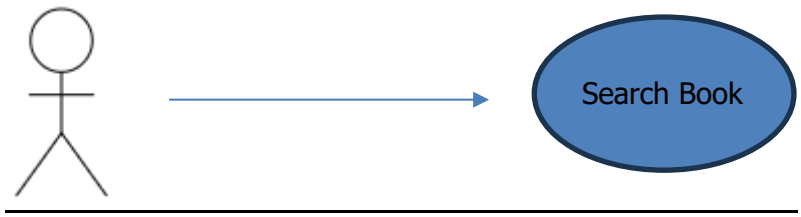
Usercase Name	**User Information**
Actors	**User (Registered)**
Main Case Senario	**1.After Logging in user will use this function to see his info.** **2.The System will display user profile**
Exception	**User isn't Registered**
Actions	
Pre-Condition	**User had to download the application.** **User have to have an account**
Post Condition	**User Can Edit any wrong info./ update any old information**

Use Case 7



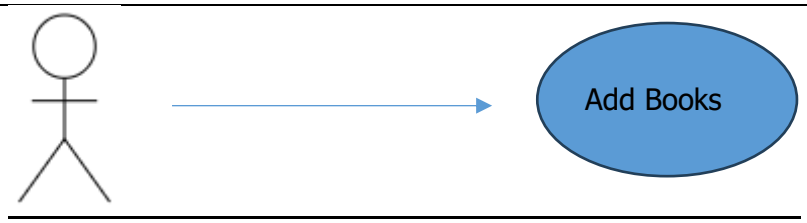
Usercase Name	**Verify Log In**
Actors	**User(Registered)/Admin**
Main Case Senario	**1.User/Admin will log in using his credentials** **2.The system will verify logging in by checking the database** **3.A One Time Password Will be Sent in order to check that the user is the one who is logging in**
Exception	➢ **User isn't Registered**
Actions	
Pre-Condition	• **User/Admin had to download the application.** • **User/Admin have to have an account**
Post Condition	• **User/Admin Successfully Logged in**

Use Case 8



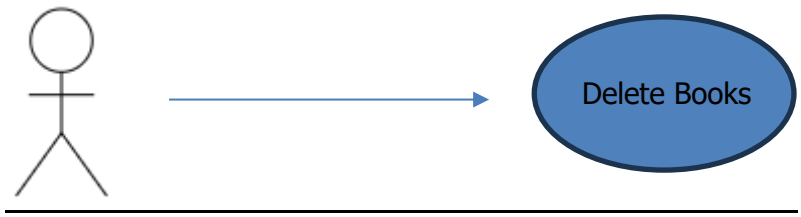
Usercase Name	**Search Book**
Actors	**Admin**
Main Case Senario	**1.Admin enters Book Title/Author etc.. in search bar** **2. List of searched books is displayed** **3. Admin can search which book he wants to pick**
Exception	➢ **Wrong title or book is searched**
Actions	➢ **No books will be displayed**
Pre-Condition	• **Admin had to download the application.** • **Admin have to have an account** • **Book is already added in database**
Post Condition	• **All books is displayed by the system**

Use Case 9



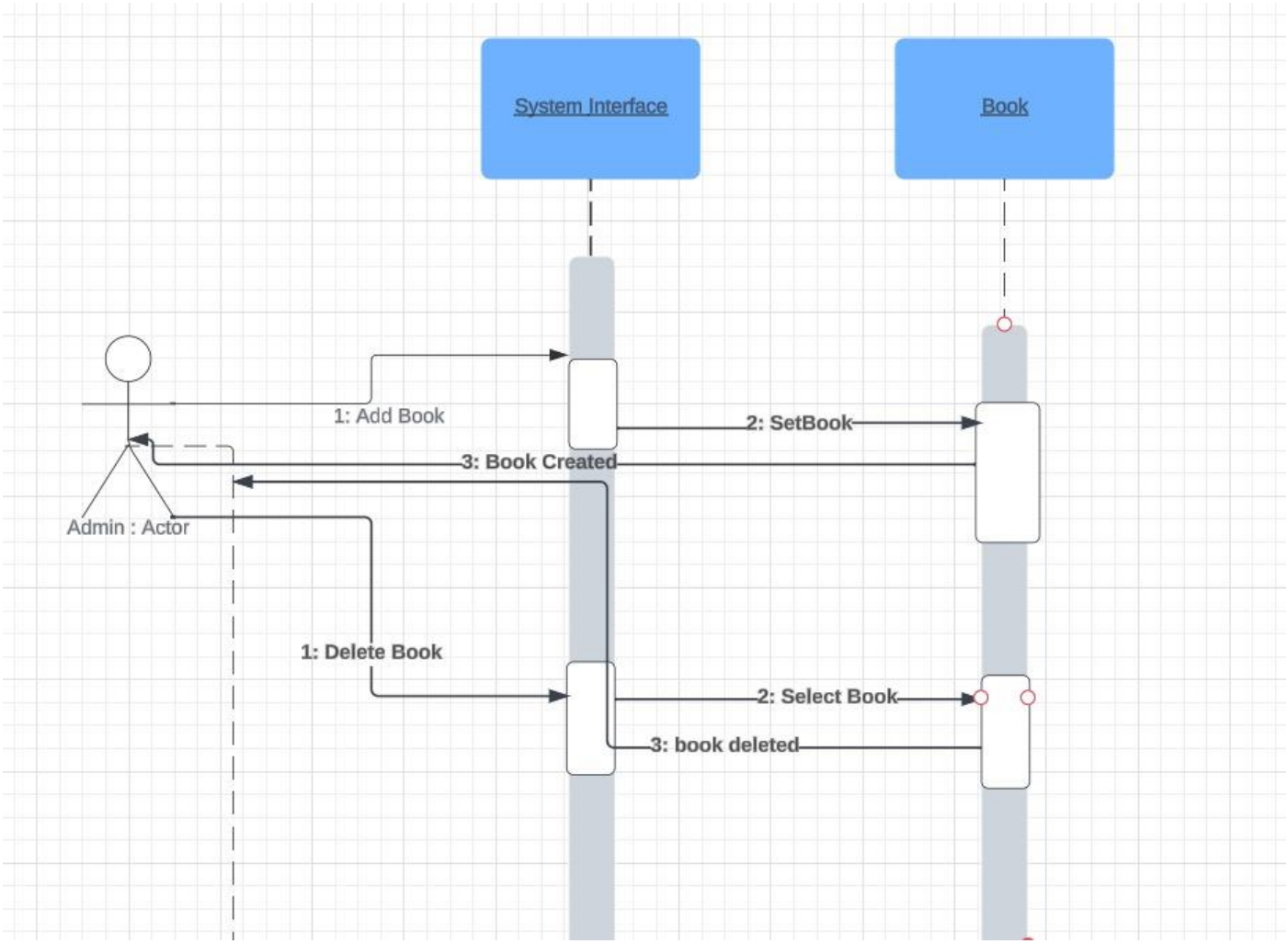
Usercase Name	**Add Books**
Actors	**Admin**
Main Case Senario	**1.Admin checks for new books online** **2. Admin access database in order to add new books**
Exception	
Actions	
Pre-Condition	• **Admin had to download the application.** • **Admin have to have an account**
Post Condition	• **All books is added to the system and database**

Use Case 10

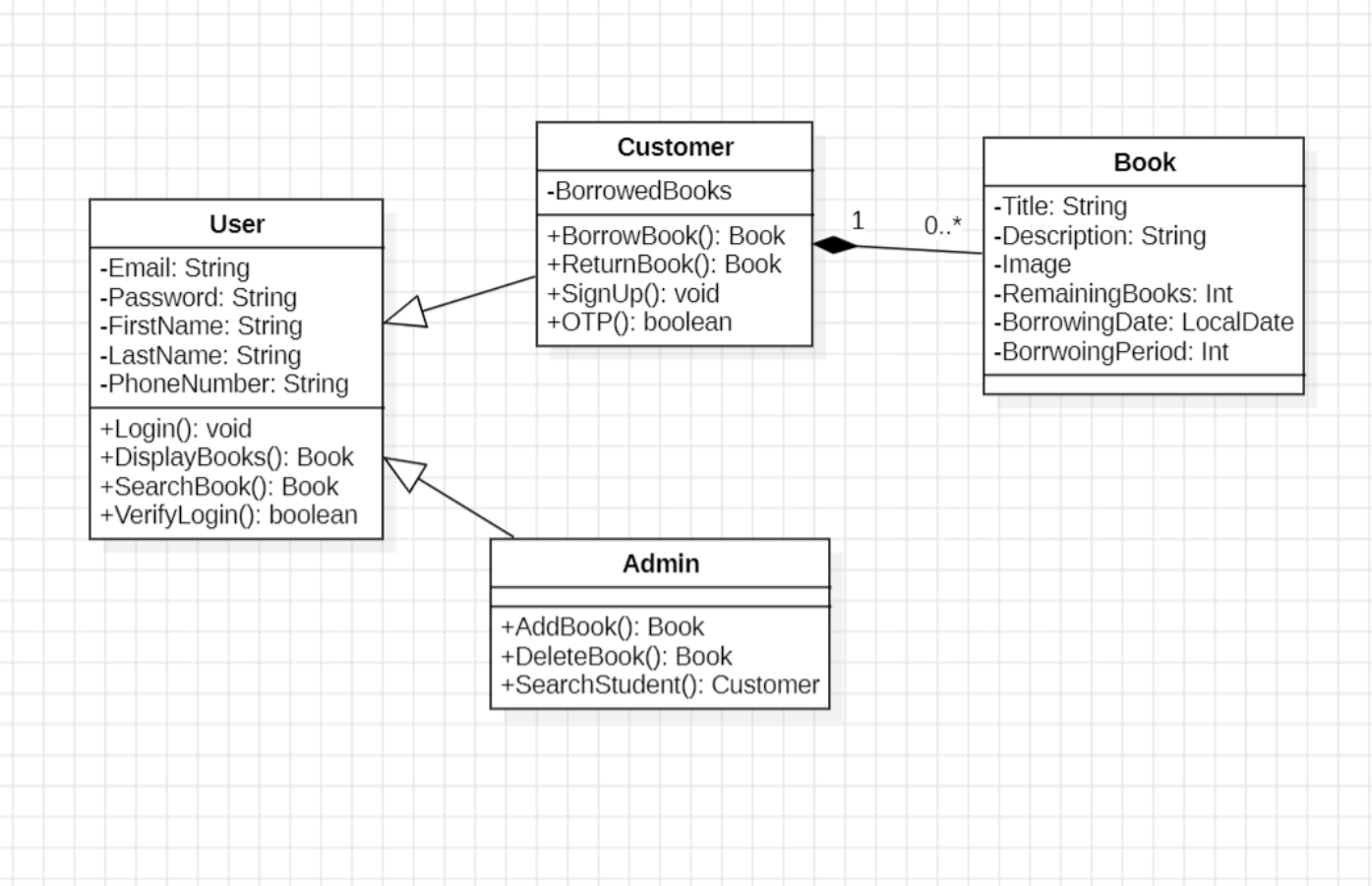


Usercase Name	**Delete Books**
Actors	**Admin**
Main Case Senario	**1.Admin searches for books he wants to delete** **2.Admin selects desired book that he wants to delete**
Exception	➢ **Book isn't found in the database**
Actions	➢ **Can't delete a book that isn't found in the database**
Pre-Condition	• **Admin had to download the application.** • **Admin have to have an account** • **The book needs to be found in the database**
Post Condition	• **All books is deleted from the system and database**

6.2. Sequence Diagram



6.3. Class Diagram



7. Test Cases

7.1. Test Case: #1

Test Case 1	
Test Case ID: LIB01	**Test Designed by:** Eyad Yehia
Test Priority (Low/Medium/High): High	**Test Designed date:** 28/12/2023
Module Name: Admin Login Panel	**Test Executed by:** Omar Elhawwari
Test Title: Verify login with valid username and password	**Test Execution date:** 28/12/2023
Description: Testing to see if the admin will log in with correct credentials	
Pre-conditions: Admin has valid username and password	
Dependencies: Admin Account is provided in database	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Press on the Admin button		User will be navigated to the admin login page	User is navigated to admin page	Pass	
2	Provide valid username	Username:Eyad07	Nothing will happen until providing password and logging in	N/A	N/A	
3	Provide valid password	Password : 2345	Nothing will happen until pressing on the login button	N/A	N/A	
4	Click on Login button		You will be successfully transferred to the admin panel	Admin can see all the functions that he can use	Pass	

Post-conditions:
Admin is validated with database and successfully logged in to account. The account session details are logged in database

7.2. Test Case: #2

Test Case 2	
Test Case ID: LIB02	**Test Designed by:** Adham Hossam
Test Priority (Low/Medium/High): High	**Test Designed date:** 28/12/2023
Module Name: User Panel	**Test Executed by:** Eyad Yehia
Test Title: Verify login with valid username and password	**Test Execution date:** 29/12/2023
Description: Testing to see if the user will log in with correct credentials	
Pre-conditions: User has valid username and password	
Dependencies: User is registered already and provided in the database	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Press on the login button		User will be navigated to the user login page	User is navigated to users page	Pass	
2	Provide valid username	Username:eyadyehia07	Nothing will happen until providing password and logging in	N/A	N/A	
3	Provide valid password	Password: eyadyehia07	Nothing will happen until pressing on the login button	N/A	N/A	
4	Click on Login button		OTP message has been sent to the user's email	OTP message has been sent to the user's email	N/A	
5	Click on Login button in OTP page		User will provide a valid OTP and will be successfully transferred to the Users panel	OTP is provided and user can see all the functions that he can use (Borrow/Return Book/User Panel)	Pass	

Post-conditions:
User is validated with database and successfully logged in to account. The account session details are logged into the database.

7.3. Test Case: #3

Test Case 3	
Test Case ID: LIB03	**Test Designed by:** Amr Ahmed
Test Priority (Low/Medium/High): High	**Test Designed date:** 28/12/2023
Module Name: Admin Panel - Adding Books	**Test Executed by:** Adham Hossam
Test Title: Successfully adding books to the system & database	**Test Execution date:** 29/12/2023
Description: Testing to see if the books are added and can be borrowed	
Pre-conditions: Admin is logged in	
Dependencies: Admin Account is provided in database	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Press on the admin button		User will be navigated to the admin login page	User is navigated to login page	Pass	
2	Provide valid username	Username: eyad07	Nothing will happen until providing password and logging in	N/A	N/A	
3	Provide valid password	Password: 2345	Nothing will happen until pressing on the login button	N/A	N/A	
4	Click on Login button		You will be successfully transferred to the admin panel	Admin can see all the functions that he can use.	Pass	
5	Click on AddBook button		Admin will be transferred to the addbook panel	Admin is navigated to admin page	Pass	
6	Fill the details of the book that you are adding(BookName, Category, Author) and the publisher is already filled with the admin username that is logged in (which in this case is eyad07)	BookName: The First 90 Days Category: Political Author : Michael D.Watkins	Book will be successfully added to the system and the database	an alert message will appear saying book is added successfully	Pass	

Post-conditions:
Admin has added a book successfully to the system. When checking the database the newly added book can be found

7.4. Test Case: #4

Test Case 4	
Test Case ID: LIB04	**Test Designed by:** Omar Elhawwari
Test Priority (Low/Medium/High): High	**Test Designed date:** 28/12/2023
Module Name: Admin Panel – Remove Book	**Test Executed by:** Omar Elhawwari
Test Title: Verify deleting book while it's borrowed	**Test Execution date:** 29/12/2023
Description: Testing to see if the admin will be able to delete the book from the list while a user is borrowing it	
Pre-conditions: Admin entered a valid book name and id	
Dependencies: Admin is logged in	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Press on the delete book button on admin panel		Admin will be navigated to the delete book page	Admin is navigated to delete book page	Pass	
2	Provide valid book name	Book Name : The Help	Nothing will happen until providing id and pressing on delete	N/A	N/A	
3	Provide valid book id	Book ID: 10	Nothing will happen until pressing on delete button	N/A	N/A	
4	Click on Delete button		An alert showing that deleting failed because the book is borrowed	An alert showing that deleting failed because the book is borrowed	Failed	

Post-conditions:
Admin is not able to delete the book from the list. No changes will be applied to the database

7.5. Test Case: #5

Test Case 5	
Test Case ID: LIB05	**Test Designed by:** Omar Elhawwari
Test Priority (Low/Medium/High): High	**Test Designed date:** 28/12/2023
Module Name: User Panel – Borrow Book	**Test Executed by:** Eyad Yehia
Test Title: Verify borrowing a book with an old return date	**Test Execution date:** 29/12/2023
Description: Testing to see if the user will be able to borrow the book from the list while an invalid/old return date is entered	
Pre-conditions: User is logged in	
Dependencies:	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Press on the borrow book button from user dashboard		Admin will be navigated to the delete book page	Admin is navigated to delete book page	Pass	
2	Provide valid book name	Book Name: The Help	Nothing will happen until providing id & return date and pressing on borrow	N/A	N/A	
3	Provide valid book id	Book ID: 10	Nothing will happen until providing return date and pressing on borrow button	N/A	N/A	
4	Provide valid return date	Return Date: 1/1/2009	Nothing will happen until pressing on borrow button	N/A	N/A	
5	Press on borrow book button		An alert showing that borrowing is failed because the return date is not valid	An alert showing that borrowing is failed because the return date is not valid	Failed	

Post-conditions:
User is not able to borrow the book from the list. No changes will be applied to the database for the user