**COMSATS UNIVERSITY ISLAMABAD**

ABBOTABAD CAMPUS



|  |  |  |
| --- | --- | --- |
| NAME | : | Muhammad Masood Khan (FA21-BSE-028)  Sofia Aamir (FA21-BSE-036)  Mahnoor Qazi (FA21-BSE-020) |
| SECTION | : | 4A |
| COURSE | : | OOSE |
| SUBMITTED TO | : | SIR Mukhtiar Zamin |

Table of Contents

[CHAPTER1: 3](#_Toc132641054)

[INTRODUCTION: 3](#_Toc132641055)

[VISION AND BUSINESS CASE: 3](#_Toc132641056)

[EXECUTIVE SUMMARY 3](#_Toc132641057)

[USE CASE MODEL: 4](#_Toc132641058)

[1. LOGIN: 4](#_Toc132641059)

[2. ADD BOOK TO LIBRARY: 4](#_Toc132641060)

[3. DELETE BOOK FROM LIBRARY: 4](#_Toc132641061)

[4. SEARCH BOOK BY NAME: 4](#_Toc132641062)

[5. DISPLAY BOOKS DETAILS: 4](#_Toc132641063)

[6. BORROW A BOOK: 4](#_Toc132641064)

[7. RETURN BOOK: 4](#_Toc132641065)

[8. REGISTER: 5](#_Toc132641066)

[9. REGISTER COMPLAINT: 5](#_Toc132641067)

[10. MANAAGE ACCOUNTS: 5](#_Toc132641068)

[11. FEEDBACK: 5](#_Toc132641069)

[12. ORDER BOOK: 5](#_Toc132641070)

[13. UPDATE BOOKS: 5](#_Toc132641071)

[SUPPLEMETARY SPECIFICATION: 5](#_Toc132641072)

[Introduction: 5](#_Toc132641073)

[Non-Functional Requirements 5](#_Toc132641074)

[1) SECURITY: 5](#_Toc132641075)

[2) RELAIBILITY/MAINTAINIBILITY: 5](#_Toc132641076)

[3) PERFOMANCE: 5](#_Toc132641077)

[4) UNDERSTANDABLE/USEABLE: 5](#_Toc132641078)

[5) FLEXIBILITY: 5](#_Toc132641079)

[GLOSSARY 6](#_Toc132641080)

[Risk List & Risk Management Plan 7](#_Toc132641081)

[1) Business Risks: 7](#_Toc132641082)

[2) Technical Risks: 7](#_Toc132641083)

[3) Resource Risks: 7](#_Toc132641084)

[4) Schedule Risks: 7](#_Toc132641085)

[CHAPTER 2 USE CASES 8](#_Toc132641086)

[USE CASE DIAGRAM: 8](#_Toc132641087)

[Use Cases Distribution 9](#_Toc132641088)

[Brief Level Use Cases 9](#_Toc132641089)

[Masood khan (FA21-BSE-028) 9](#_Toc132641090)

[Mahnoor Qazi (FA21-BSE-020) 9](#_Toc132641091)

[Sofia Aamir (FA21-BSE-036) 10](#_Toc132641092)

[Fully Dressed Use Cases 11](#_Toc132641093)

# CHAPTER1:

## INTRODUCTION:

A library management system is a computerized system which is used to have the record of the books ,students and staff records .which has a software installed in it which helps the user in handling all the books of library and with that keeping everything up to date,

It helps the manager to know how many books are there in that particular library, how many are given to the students, how many books of same author do we have and many other features that I will explain later.

This system is easy to use and at times helps both the members of the library and the manager. There will be different sections for the student and staff to use the applications. Librarian will use complete features of application.

Futermore, the system will be flexible it will run on many machines across the world, we are using iterative development strategy we are going to processed through requirements ,object oriented analysis design and implementation.

## VISION AND BUSINESS CASE:

Out Application will be fast, reliable, robust and flexible it will be very user friendly.

**Problems:** the speed will be decrease if there are multiple users using at a time too much load will reduce speed.

Our application is in English language when people don’t know English they will not understand the use of application.

**Constraints:** Including design constraints, performance, supportability, reuseability, and Documentation.

## EXECUTIVE SUMMARY

Our traditional libraries are getting hard to manage day by day as the world is advancing with every passing day. Student life is way too busy now and it is difficult to take out time from your busy schedule to go to library. While during studies, library always acts as ours best friend. Libraries help students in studies, it not only helps students but everyone that has keen interest of reading books.

In our library management system application, there will be an admin (librarian) login. He has the access to most of the functions of the library. He registers the students, keeps a check on the students whether they are returning books on time or not.

The librarian manages all the books in library. In this library management application, it is easy for the librarian to tell which book is available, which book is currently not available in library. How many books of same author are present in our library.

Librarian issues a certificate i.e. registration of the students who desires to be a permanent member of the library. After the registration the students are able to borrow a book. In our LMS application when a student gets registered, he/she is provided with a student login account. Through this account they can borrow any book and stay up to date.

In this library management application, there is a student login that makes this software different from others, way too much interesting and attractive for users. Student login account helps student to get information about any books within few seconds. There is a search bar where they can type author’s name, book id or book name and they can know if their desired book is available or not. Students can see the date they borrowed a book and the date when they have to return the book.

Students have to pay the fine if they fail to return the book on time. They can see all the information of fine through their accounts.

The UI (user interface) will be simple to understand easily by the user, no complicated buttons will be used which that can require any special training, everyone even the beginners will learn how to use the System easily and also the system will be programmed well to avoid cyber-attacks, the System will have data of many users.

To make it safe we will add authentication processes so that unauthorized people are unable to access the data.

## USE CASE MODEL:

### 1. LOGIN:

### 2. ADD BOOK TO LIBRARY:

When new books will come the system will update the list of books by adding all information of that book and at the end letting the manager know the total number of books in the library.

### 3. DELETE BOOK FROM LIBRARY:

If a book is no longer available in the library for one reason or other the system will delete it too

### 4. SEARCH BOOK:

If a student doesn’t knows the author name he/she will tell the librarian the book name and using thus software system, librarian will give the information about that particular book to the student.

### 5. BORROW A BOOK:

There will an option/feature of borrowing books. A user can borrow 2 books of their choice at a time.

6. RETURN BOOK:

When user borrows the book they have to return it after a certain mentioned time period. If they do not return the book due to one reason or other, fine will be imposed which will increase by 100RS with every passing day.  
7. REGISTER:

### 8. Manage Accounts

### 9. Allocate shift

### 10. Feedback

### 11. Order books

### 12. Logout

## SUPPLEMETARY SPECIFICATION:

### Introduction:

This document is the repository of all Library Management system not captured in the use cases.

### Non-Functional Requirements

### SECURITY:

My software has many accounts which keeps the user data their bank info, CNIC and other information which is confidential. So, lots of data must be kept safe avoid cyberattacks and recognize the unauthorized access directly and block access to that person.

### RELAIBILITY/MAINTAINIBILITY:

The software will be reliable and work properly make sure no errors and bugs will affect the performance of the system. If any Bug or Error arises the system will only make that Function out of service not the whole system and the system will do Maintenance of that certain Function which is causing error and bugs.

### PERFOMANCE:

We will make sure our system will do a certain operation faster and the start-up time of system will be less. Whenever the customer does a certain function, the operation will be done in a fraction of a second. Our goal: authorization in less than 1minute 90% of the time.

### UNDERSTANDABLE/USEABLE:

Make the system more user friendly more understandable by the user and easily to interact with.

### FLEXIBILITY:

The system can handle Thousands of users at a time.

**Human Factors:**

* Text should be visible from 1 meter
* Avoid colors associated with common forms of color blindness.

## GLOSSARY

|  |  |
| --- | --- |
| **SDLC** | Software development life cycle |
| **LMS** | Library Management System |
| **OOP** | Object Oriented Programming |
| **ICT** | Introduction to Computer Technology |
| **SSO** | Single Sign-on |

## Risk List & Risk Management Plan

A library management system is a critical system that help to manage the books resources and along that there are also many other difficulties faces by library. Just like other systems library also face the security issues, performance issues and also maintenance issues.

### Business Risks:

Business risk that a library can face in market is that they sometimes can’t meet the needs of targeted audience. Sometimes also the budget of one project is so much that it exceed the finance.

**Mitigation/Response:**

Analyse your customer base and also we can conduct the interviews to come to know their needs. Also before planning any project we have to look to our finance and then plan the things according to that.

### Technical Risks:

There are security risks the librarian managers can face. That other cyber attacker can also reach out the accounts in the library which is not good.

**Mitigation/Response:**

Regularly perform security audits and updates the system for its security. The library should heir the well qualified employees with technical skills that they have the knowledge to use the robust security protocols such as firewall, encryption to protect the system from security threats.

### Resource Risks:

The project may require a high qualified technical skills sometimes the resources are not allocate to the right person than they leave the project in midway.

**Mitigation/Response:**

Build a team with a diverse skills and all employees must be independent and they must be one alone individual.

### Schedule Risks:

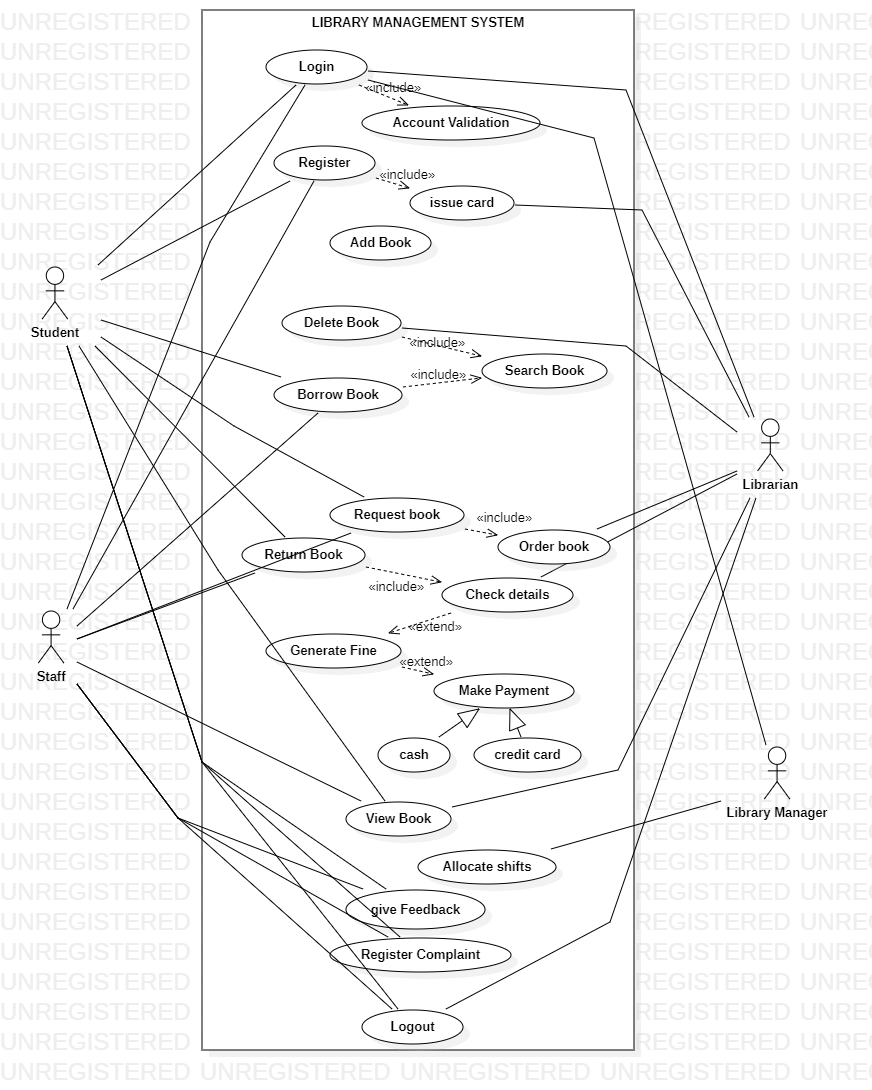
Sometimes the task assigned to one person may busy in some other project that he can’t give time to this project so there is delay in project.

**Mitigation/Response:**

Develop a project time by keeping the schedule in mind of all other staff. That they give their full time to the project.

# CHAPTER 2 USE CASES

USE CASE DIAGRAM:



## Use Cases Distribution

|  |  |  |
| --- | --- | --- |
| S#. | Group Member | Assigned Use Cases |
| 1 | Masood Khan | Add book  Delete Account  Order books  Reserve books  Block Account |
| 2 | Mahnoor Qazi | Search book  Login  Logout  Return book  Generate fine |
| 3 | Sofia Aamir | Register accounts  Delete book  Borrow book  Give feedback  payment |

## Brief Level Use Cases

### Masood khan (FA21-BSE-028)

#### ` Add book:

As a librarian or admin, I should be able to add new books to the library system by providing book details such as title, author, ISBN, category, publisher.

#### Delete account:

As a librarian or admin, I should be able to manage accounts of the users of the library.so a librarian can delete an account if he want to delete .

#### Block account:

As a librarian or admin ,if any user is doing any unsuspicious activity so librarian has the authority to block his account or the user is not paying fine.

#### Order books:

As a library member, I should be able to search for the book and if I can’t find that book will place order for the desired book and the librarian will order that book.

#### Reserve books:

As a library member, I will be able to reserve the book in library so that another person can’t borrow that book.

### Mahnoor Qazi (FA21-BSE-020)

#### Search book:

As a library member, I should be able to search the book by IBSN author name, title, publisher name.

#### Login:

As a library member or guest, I should be able to login in library by username and password. If I don’t have an account I have to find make an account and then I will be able to use the functionalities of library. After login I should be able to see my account details, books borrowing history etc.

#### Logout:

As a library member, I should be able to logout from my library account to ensure the security after using all the functionalities of library for that time.

#### Return book:

As library member, I should be able to return the book after borrowing the book and the status of book will be update to available and also there will be fine generated if I have return the book after the return date.

#### Generate fine:

As a library staff, I should be able to generate the fine or late fee for the overdue books based on the library policy and the number of days the book is overdue. The system should automatically calculate the fine amount and notify the member via email or SMS. The staff should also be able to mark the fine as paid once the member clears the dues.

### Sofia Aamir (FA21-BSE-036)

#### Register accounts:

As a library member, I should be able to create a new account by providing my personal details such as name, email, phone number, and address. After registration, I should be able to login to the library system and access the available features based on my account role and permissions.

#### Delete book:

As a librarian or staff, I should be able to delete the book from library details if it is borrowed, lost or damaged. The system should update the book details.

#### Borrow book:

As a library member, I should be able to borrowed the book from library by IBSN, title name, author name. the system should update the status of book to unavailable and should also generate the date for returning the book.

#### Give feedback:

As a library member, I should be able to give feedback on the books available in library after reading the book. That feedback should be visible to all other users also.

#### Payment:

As a library member, I should be able to pay the fines or overdue charges online using various payment modes such as credit/debit card, net banking, or digital wallets. The system should process the payment securely and update the payment status for the corresponding fines or dues.

## Fully Dressed Use Cases

### MASOOD KHAN (FA21-BSE-028)

| **Use Case UC1: ADD BOOK** |  |  |
| --- | --- | --- |
| **Scope**: Library Management application  **Level**: user goal  **Primary** **Actor**: Librarian  **Users and Interests**:  - Librarian: Wants accurate, fast entry, and well managed book in the library which can be easily accessible.  - User : want to buy borrow a book from the library  - University: Wants to accurately record book details so it can be easily available for the students and Faculty members to buy or borrow a book and will facilitate them better.  - User: he/she will be at the top level and can manage all the operations in the library can override all the features of librarian.  **Preconditions**: Librarian is authorized and authenticated. |  |  |

**Success Guarantee** (or Postconditions): Book will be added the Library Management system and will be assigned the correct shelf number and will be visible to the users of Library management system.

**Main Success Scenario (or Basic Flow):**

1. The librarian or library staff member logs into the library management system using their credentials.

2. Upon successful login, the system verifies the user's access rights and permissions to perform administrative tasks, such as adding new books.

3. The librarian selects the option to add a new book from the system's menu or interface.

4. The system prompts the librarian to enter the relevant details of the book, such as the title, author, ISBN, publication year, genre, and any other necessary information.

5. The librarian enters the book details into the system accurately.

6. The system validates the entered information to ensure it meets the required criteria. This may include checking for duplicate entries, verifying the ISBN format, or performing any other necessary checks.

7. If any errors or missing information are detected, the system prompts the librarian to correct or provide the necessary details.

8. Once all the information is correctly entered and validated, the system generates a unique identifier or barcode for the book.

9. The system updates the library's database or catalog with the new book entry, including all the relevant details and the generated identifier.

10. The book is assigned a suitable location within the library, such as a specific shelf or section, based on the library's classification system.

11. The system updates the book's availability status to indicate that it is now part of the library's collection.

12. If the library management system is integrated with other systems, such as an online catalog or circulation system, the new book's details are synchronized to ensure consistency across platforms.

13. The librarian or library staff member confirms the successful addition of the book and proceeds with other tasks or services as needed.

Following this scenario, the library management system ensures secure access to authorized users and facilitates the seamless addition of new books to the library's collection.

**Extensions (or Alternative Flows):**

\*a. At any time, User requests an override operation:

1. System enters Manager-authorized mode.
2. Librarian performs one Manager-mode operation.
3. System reverts to Librarian-authorized mode.

\*b. At any time, System fails:

If there occur any error while adding a book to the library there must be alternate method to do the operation

1. Librarian restarts the system and check the system if it works fine or not then enter prior state
2. System reconstructs prior state.

**Special Requirements:**

- Book will be added easily and fast within 30secounds 90%of time.

- better Machines will be provided to the librarian for fast data entry

- Language internationalization on the text displayed.

**Technology and Data Variations List**:

\*a. to add a book Multiple authentication process will be done. Fingerprint will be scanned .

3a. book identifier entered by bar code laser scanner (if bar code is present) or keyboard.

7a. Entry will be done through IBAN.

**Use Case UC2: BLOCK ACCOUNTS**

**Title:** Block Accounts

**Actor:** Librarian

**Preconditions:**

The librarian is logged into the LMS.

The user whose account is to be blocked has violated library policies, such as returning books late, damaging library property, or harassing other library users.

**Basic Flow:**

The main success scenario for blocking an account in a Library Management System (LMS) typically involves the following steps:

1. The librarian or library staff member logs into the LMS using their username and password.

2. Upon successful login, the system grants the user the necessary privileges to perform administrative tasks, including blocking user accounts.

3. The librarian accesses the account management section or menu of the LMS.

4. The librarian searches for the user account they wish to block, usually by entering the user's unique identifier, such as a library card number or username.

5. The system retrieves the user account based on the provided identifier.

6. The librarian selects the option to block or suspend the user account.

7. The system prompts the librarian to provide a reason or select a predefined reason for blocking the account. This information is helpful for record-keeping purposes and may be communicated to the user.

8. The librarian enters the reason for blocking the account or selects an appropriate reason from the predefined options.

9. The system confirms the blocking action and updates the user account's status to reflect that it has been blocked.

10. The blocked user account is prevented from accessing certain services or resources within the LMS, such as borrowing books, accessing online databases, or making reservations.

11. If the LMS is integrated with other systems or platforms, such as online catalogs or user authentication services, the blocked status is synchronized to ensure consistency across platforms.

12. The librarian may choose to communicate the account blocking to the user, either manually or through an automated notification sent by the system. This communication may include information on how to resolve the issue or the duration of the block if applicable.

13. The librarian or library staff member verifies and confirms the successful blocking of the account and proceeds with other tasks or services as needed.

By following this main success scenario, the LMS allows the library staff to effectively manage user accounts, ensuring appropriate actions are taken when necessary to maintain the integrity and security of the library's services.

**Alternate Flow:**

If the user disputes the blocking of their account, the librarian may review the evidence of the user's violation and decide to either unblock the account or keep it blocked.

**Post-conditions:**

The user's account is blocked and cannot be used until unblocked by the librarian.

The reason for blocking the user's account is logged in the LMS for future reference.

The user is informed of the reason for blocking their account and any steps they can take to unblock it

**Use Case UC3: DELETE ACCOUNTS**

**Title:** DELETE Accounts

**Primary Actor:** Librarian

**Goal:** To delete a user account from the Library Management System

**Preconditions:**

* The Librarian is logged in to the Library Management System.
* The user account to be deleted is not currently logged in.
* The Librarian has the necessary permissions to delete user accounts.

**Main Success Scenario:**

The main success scenario for deleting an account in a Library Management System (LMS) typically involves the following steps:

1. The librarian or library staff member logs into the LMS using their credentials.

2. Upon successful login, the system grants the user the necessary privileges to perform administrative tasks, including deleting user accounts.

3. The librarian accesses the account management section or menu of the LMS.

4. The librarian searches for the user account they wish to delete, usually by entering the user's unique identifier, such as a user isbn.

5. The system retrieves the user account based on the provided identifier.

6. The librarian selects the option to delete the user account.

7. The system displays a confirmation message, prompting the librarian to confirm the deletion of the account. This step serves as a safeguard to prevent accidental deletions.

8. The librarian confirms the deletion by selecting the appropriate option or entering a confirmation code, if required.

9. The system permanently removes the user account from the LMS, including all associated data and records.

10. If the LMS is integrated with other systems or platforms, such as online catalogs or user authentication services, the account deletion is synchronized to ensure consistency across platforms.

11. The system updates any relevant records, such as borrowing history, fines, or reservations, to reflect the removal of the account.

12. The librarian may choose to communicate the account deletion to the user, either manually or through an automated notification sent by the system. This communication may include information on how to resolve any pending issues or how to re-register if necessary.

13. The librarian or library staff member verifies and confirms the successful deletion of the account and proceeds with other tasks or services as needed.

By following this main success scenario, the LMS allows the library staff to effectively manage user accounts, ensuring that accounts are properly removed from the system when required, and maintaining the integrity and security of the library's services.

**Alternative Flow:**

* If the Librarian enters an invalid username or password, the LMS displays an error message and prompts the Librarian to try again.

**Postconditions:**

* The user account has been deleted from the LMS and all associated data has been removed.
* The user will no longer be able to log in to the LMS or use any of its services.
* The Librarian can no longer access the deleted user account's data.

**Use Case UC4: Order Books**

**Primary Actor:** Library Staff

**Goal in Context:** The goal of this use case is to enable library staff to order new books or replace damaged or lost books in the library system.

**Preconditions:** The library staff member has the necessary permissions to access the order books feature and has identified the need for new books or replacements.

**Trigger:** The trigger for this use case is the need to purchase new books or replace damaged or lost books in the library.

**Main Success Scenario:**

The main success scenario for ordering books in a Library Management System (LMS) typically involves the following steps:

1. The librarian or library staff member logs into the LMS using their credentials.

2. Upon successful login, the system grants the user the necessary privileges to perform administrative tasks, including ordering books.

3. The librarian accesses the book ordering section or menu of the LMS.

4. The librarian searches for the desired book either by entering specific details such as title, author, or ISBN, or by browsing through the library's catalog.

5. The system retrieves the book information based on the provided search criteria and displays the available options.

6. The librarian selects the book they wish to order from the displayed options.

7. The system prompts the librarian to provide additional information related to the order, such as the quantity needed, preferred edition or format, and any specific instructions or notes.

8. The librarian enters the required information accurately.

9. The system may provide real-time information on the availability and pricing of the book from various vendors or suppliers. The librarian selects the preferred vendor or supplier based on factors such as price, delivery time, and reliability.

10. The system generates an order request or purchase order with the necessary details, including the book information, quantity, vendor/supplier details, and any additional instructions.

11. The librarian reviews the order request for accuracy and completeness.

12. Once satisfied, the librarian submits the order request to the appropriate vendor or supplier.

13. The system records the order request and updates the order status to indicate that it is pending.

14. The vendor or supplier processes the order and ships the books to the library.

15. Upon receipt of the ordered books, the librarian verifies the delivered items against the order request to ensure accuracy.

16. If there are any discrepancies or issues with the delivered books, the librarian contacts the vendor or supplier to resolve the matter.

17. The librarian updates the order status in the LMS to indicate that the books have been received and added to the library's collection.

18. The system updates the library's database or catalog with the new book entries, including all relevant details and availability status.

19. The librarian or library staff member confirms the successful order and receipt of the books and proceeds with other tasks or services as needed.

By following this main success scenario, the LMS facilitates efficient book ordering processes, ensuring that the library can acquire the desired books for its collection and provide access to a diverse range of resources for library patrons.

**Alternative Scenarios:**

1a. If the desired book(s) are not available in the library catalog and a book request form is filled out, the staff member receives a notification when the book(s) are available to purchase. 5a. If the staff member reviews the order details and finds an error, they can edit the order before confirming it. 8a. If the delivered books are damaged or incomplete, the staff member can return them and request a replacement.

**Postconditions:**

The vendor(s) receive the purchase order and fulfill the order.

The library receives the ordered book(s) and adds them to the library collection.

The library catalog is updated to reflect the new additions.

**Assumptions**:

The library management system is fully functional and accessible to the library staff.

The vendor(s) can fulfill the purchase order within the expected time frame.

The library staff member has the necessary knowledge and training to navigate the order books feature and manage book orders.

**Use Case UC5: RESERVE Books**

**Title:** Reserve Books

**Primary Actor:** Librarian

**Goal in Context:** The goal of this use case is to enable librarian to reserve a copy of a book that is currently checked out.

**Preconditions:** The librarian has a valid library card and is logged into the library system. The book being reserved is currently checked out.

**Trigger:** The trigger for this use case is the library patron's desire to reserve a book that is currently checked out.

**Main Success Scenario:**

The main success scenario for reserving books in a Library Management System (LMS) typically involves the following steps:

1. The library user logs into the LMS using their credentials.

2. Upon successful login, the system grants the user the necessary privileges to perform actions, including reserving books.

3. The user searches for the desired book either by entering specific details such as title, author, or ISBN, or by browsing through the library's catalog.

4. The system retrieves the book information based on the provided search criteria and displays the available options.

5. The user selects the book they wish to reserve from the displayed options.

6. The system checks the availability of the selected book.

7. If the book is currently available, the system allows the user to borrow the book directly, bypassing the reservation process. The scenario for reserving the book ends here.

8. If the book is currently unavailable, the system provides an option to reserve the book.

9. The user selects the option to reserve the book.

10. The system prompts the user to confirm the reservation.

11. The user confirms the reservation.

12. The system records the reservation request and updates the book's status to indicate that it has been reserved.

13. The system may provide an estimated availability date or queue position to the user, indicating when the book is expected to be available for pickup.

14. When the reserved book becomes available, the system automatically notifies the user via email, notification within the LMS, or any other preferred communication method.

15. The user visits the library within a specified time frame (usually a few days) to pick up the reserved book.

16. The librarian or library staff member retrieves the reserved book from the hold shelf or designated area.

17. The librarian or library staff member verifies the user's identity and eligibility to borrow the reserved book.

18. The user receives the reserved book and completes the borrowing process, which may involve scanning their library card or confirming their details with the librarian.

19. The librarian or library staff member updates the book's status in the LMS to indicate that it has been borrowed.

20. The user enjoys the borrowed book for the designated loan period.

By following this main success scenario, the LMS enables users to reserve books that are currently unavailable and provides a streamlined process for notifying users when the reserved books become available for pickup.

**Alternative Scenarios:**

3a. If the book is not available in the library catalog, the patron can fill out a book request form with the required book information. 4a. If the patron decides not to confirm the reservation, the reservation is canceled. 5a. If the patron is not the first on the waitlist, the library system updates the waitlist to reflect the patron's position in the queue. 6a. If the reserved book is not returned within the designated time frame, the library staff may contact the borrower and request the return of the book. 7a. If the patron does not pick up the reserved book within the designated time frame, the reservation is canceled and the next patron on the waitlist is notified.

**Postconditions:**

The library patron has successfully reserved a copy of a book that is currently checked out.

The library system updates the book's availability status and waitlist.

The library patron picks up the reserved book within the designated time frame.

**Assumptions**:

The library management system is fully functional and accessible to the library patrons.

The reserved book will be returned within the designated time frame.

The library staff member has the necessary knowledge and training to manage the reservation process.

The patron has a valid library card and contact information.

### SOFIA AAMIR (FA21-BSE-036)

**Use Case UC1: REGISTER ACCOUNTS**

**Use case**: Register Account in Library Management System

**Actor**:

User: The individual who wants to register for an account in the LMS.

**Precondition:**

- The LMS is accessible and functioning properly.

- The user has not registered for an account in the LMS previously.

**Postcondition:**

- The user has successfully registered for an account in the LMS.

- The user can log into the LMS using their newly created account credentials.

**Main Success Scenario:**

1. The user accesses the library's LMS through an online portal or visits a physical library location.

2. The user selects the option to register for a new account.

3. The system presents a registration form to the user, prompting them to provide necessary information.

4. The user enters their full name, email address, and desired password into the registration form.

5. The user submits the registration form.

6. The system validates the entered information, ensuring that all required fields are filled correctly.

7. If any errors or missing information are detected, the system prompts the user to correct or provide the necessary details.

8. Once all the information is correctly entered and validated, the system generates a unique identifier for the user account, such as a library card number or username.

9. The system creates the user account in the LMS database and associates the entered information with the generated identifier.

10. The system sends a confirmation email to the user's provided email address, containing a verification link or a confirmation code.

11. The user receives the confirmation email and verifies their account by clicking on the verification link or entering the confirmation code.

12. The system verifies the account and updates the account status to "active" in the LMS.

13. The system displays a success message or redirects the user to a login page, indicating that the account registration process is complete.

14. The user can now log into the LMS using their newly created account credentials.

15. After logging in, the user gains access to their account dashboard and can explore the library's collection, search for books, make reservations, and utilize other available services.

16. The user can update their account information, such as contact details or preferences, through the account settings provided by the LMS.

17. Library staff periodically review and verify user accounts to ensure the validity and accuracy of information.

18. If any issues arise during the registration process, the user can contact the library's support desk for assistance.

**Extensions**:

- If the user enters an email address that is already registered in the system:

- The system displays an error message informing the user that the email address is already associated with an existing account.

- The user is prompted to either log into their existing account or recover/reset their password if needed.

**Exceptions**:

- If the LMS is experiencing technical difficulties, preventing the user from completing the registration process:

- The system displays an error message apologizing for the inconvenience and advises the user to try again later.

- The user is advised to contact the library's support desk if the issue persists.

This fully dressed use case outlines the typical steps involved in registering an account in the LMS, including validation, email verification, and account activation.

| **Use Case UC2: BORROW BOOK** |
| --- |
| **Scope**: Library Management application  **Level**: user goal  **Primary** **Actor**: User  **Users and Interests**:  - Librarian: will authorize the permission of the user when user wants to borrow a book from the library.  - User : wants to buy borrow a book from the library.  - University: Wants to accurately record book details, will request more when certain book is not available. So it can be easily available for the students and Faculty members to buy or borrow a book and will facilitate them better.  - User: he/she will get request by the librarian for new books then User will approve or reject the request.  **Preconditions**: User is authorized and authenticated. |

**Success Guarantee** (or Postconditions): Book will be borrowed and a time will be generated for the returning of book.

**Main Success Scenario (or Basic Flow):**

The main success scenario for borrowing a book in a Library Management System (LMS) typically follows these steps:

1. User login: The library user logs into the LMS using their unique credentials, such as a username and password.

2. Book search: The user searches for the desired book by entering relevant information like the book title, author name, or ISBN.

3. Book availability check: The LMS retrieves the search results and displays the availability of the book. It shows whether the book is currently available for borrowing or if it is already checked out by another user.

4. Book selection: The user selects the specific book they want to borrow from the available options.

5. Borrowing request: The user submits a borrowing request for the selected book. This action notifies the system and initiates the borrowing process.

6. Verification: The LMS verifies the user's borrowing eligibility. It checks if the user has any pending fines, overdue books, or other restrictions that may prevent them from borrowing the book. If the user meets all the criteria, the verification process proceeds successfully.

7. Loan confirmation: Once the verification is complete, the LMS confirms the loan request. The system records the user's borrowing details, such as the due date and any associated fees.

8. Book checkout: The user is notified that the book has been successfully borrowed. They may receive a physical copy of the book from the library staff or have access to an e-book version through the LMS, depending on the library's policies.

9. Return reminder: The LMS sends reminders to the user closer to the due date, reminding them to return the borrowed book in time.

10. Book return: Once the user has finished reading the book or the due date has arrived, they return the book to the library physically or through the LMS, depending on the library's policies.

11. Return processing: The library staff receives the returned book and checks it back into the LMS. The book's status is updated as "available" again, making it ready for other users to borrow.

Overall, the main success scenario for borrowing a book in an LMS ensures a streamlined process for users to search, request, and borrow books while maintaining proper record keeping and facilitating efficient library operations.

**Extensions (or Alternative Flows):**

\*b. At any time, when book cannot be borrowed any error occurred,

If there occur any error while adding a book to the library there must be alternate method to do the operation

1. User will restart the system and try again with better and fast internet connection .
2. If it does not work then user will submit a complaint in the application and can email the user for the desired book he want to borrow.

**Special Requirements:**

- Book will be borrowed easily and fast within 30secounds 90%of time.

- a simple UI will be designed for the users to easily understand the Library Management system features.

- better Machines will be provided to the librarian for fast data entry

- Language internationalization on the text displayed.

**Technology and Data Variations List**:

\*a. to borrow a book Multiple authentication process will be done. Fingerprint will be scanned.

3a. before borrowing a book user will get the approval from the librarian to borrow a book

7a. Entry will be done through IBSN.

**Use Case UC3: DELETE BOOK**

**Title:** Delete Books

**Primary Actor:** Library Staff

**Goal in Context:** The goal of this use case is to enable library staff to remove a book from the library's collection.

**Preconditions:** The library staff member is logged into the library system and has the necessary permissions to delete books. The book to be deleted is not checked out by any library patron.

**Trigger:** The trigger for this use case is the decision by the library staff to remove a book from the library's collection.

**Main Success Scenario:**

The main success scenario for deleting a book in a Library Management System (LMS) typically follows these steps:

1. User login: The library staff into the LMS using their unique credentials, such as a username and password.

2. Book search: The staff or administrator searches for the book they want to delete by entering relevant information like the book ISBN.

3. Confirmation: The staff confirms their intention to delete the book. This step is crucial as book deletion is a permanent action.

4. Book deletion: The LMS initiates the book deletion process based on the confirmation. The system removes the book's entry from the database and related records, such as availability status.

5. Removal of associated data: If the book had any associated data or metadata, such as reviews, ratings, or linked resources, the LMS also removes those records to maintain data consistency and integrity.

6. Update of book inventory: The LMS updates the book inventory count or statistics to reflect the deletion accurately. This ensures that the library's collection records remain up to date.

7. Confirmation of deletion: The LMS displays a confirmation message to the staff or administrator, indicating that the book has been successfully deleted from the system.

8. Optional: In some cases, the LMS may prompt the staff or administrator to take additional actions, such as removing physical copies of the book from the library shelves or updating related catalog records.

9. System synchronization: If the LMS is connected to other systems or platforms, such as online catalogs or external databases, it may trigger a synchronization process to ensure that the book's deletion is reflected across all relevant platforms.

By following this main success scenario, the LMS ensures that the book is effectively and accurately deleted from the system, maintaining data integrity and providing an up-to-date catalog for library users.

**Alternative Scenarios**:

2a. If the book is checked out by a library patron, the staff member cannot delete the book until it is returned. 2b. If the book has holds or reservations, the staff member cannot delete the book until all holds and reservations have been fulfilled or canceled. 2c. If the book has associated data such as circulation history or reviews, the staff member may choose to keep this data or delete it along with the book.

**Postconditions:**

The book has been successfully removed from the library's collection.

The library catalog and inventory have been updated to reflect the deletion.

**Assumptions:**

The library management system is fully functional and accessible to the library staff.

The staff member has the necessary knowledge and training to manage the deletion process.

The staff member has the necessary permissions to delete books from the library's collection.

The book to be deleted is not checked out by any library patron.

**Use Case UC4: GIVE FEEDBACK**

**Title:** Give Feedback

**Primary Actor:** User

**Goal in Context:** The goal of this use case is to enable library patrons to provide feedback to the library regarding their experience with the library services, facilities, or resources.

**Preconditions:** The library patron has a valid library card and is logged into the library system.

**Trigger:** The trigger for this use case is the library patron's decision to provide feedback regarding their experience with the library.

**Main Success Scenario:**

The main success scenario for giving feedback in a Library Management System (LMS) typically follows these steps:

1. User login: The library user logs into the LMS using their unique credentials, such as a username and password.

2. Feedback submission: The user navigates to the feedback section or option within the LMS interface.

3. Feedback form: The LMS presents a feedback form where the user can provide their feedback.

4. Feedback details: The user fills in the required fields, providing specific and detailed feedback about their experience, suggestions for improvement, or any issues they encountered and also enter their name and email address through which the librarian can contact them.

5. Submission: Once the user has completed the feedback form, they submit their feedback by clicking a "Submit" or similar button.

6. Confirmation message: The LMS displays a confirmation message to the user, acknowledging the successful submission of their feedback.

7. Feedback processing: The LMS processes the submitted feedback, which may involve storing the feedback details in a database or forwarding it to the appropriate staff or department responsible for handling feedback.

8. Review and analysis: The library staff or administrators periodically review and analyze the feedback received through the LMS. They examine the feedback for insights, identify recurring issues or suggestions, and use the feedback to inform decision-making and improvements in library services.

9. Response (optional): In some cases, the library staff or administrators may respond to the user's feedback. This response could be through the LMS interface itself, via email, or by other means of communication specified by the user.

10. Continuous improvement: Based on the feedback received, the library staff takes appropriate actions to address any identified issues, implement suggested improvements, or communicate necessary changes to library users.

By following this main success scenario, the LMS facilitates effective feedback collection and analysis, allowing library users to contribute their input and helping the library improve its services and user experience.

**Alternative Scenarios:**

3a. If the user does not want to fill out the feedback form, they can select a pre-written feedback option or skip the feedback process altogether. 4a. If the feedback form does not include a specific aspect of the library experience that the patron wants to provide feedback on, they can add it in a comment box or provide feedback in a different way (e.g., email, phone call). 5a. If the feedback requires follow-up action, such as a maintenance request or a request for additional resources, the appropriate library staff member will take action and respond to the patron when necessary. 6a. If the library staff member needs more information or clarification regarding the feedback, they may contact the patron to request more information.

**Postconditions:**

The library patron has successfully provided feedback to the library regarding their experience with the library services, facilities, or resources.

The library system has recorded the feedback and sent it to the appropriate library staff member for review and response.

**Assumptions:**

The library management system is fully functional and accessible to the library patrons.

The feedback form is user-friendly and accessible to all library patrons.

The library staff member has the necessary knowledge and training to manage the feedback process.

The library staff member will respond to the patron in a timely manner when necessary.

**Use Case UC5: PAYMENT**

**Title:** Payment

**Primary Actor:** user

**Goal in Context:** The goal of this use case is to enable library patrons to make payments for library fines, fees, or other charges.

**Preconditions:** The library patron has a valid library card and is logged into the library system. The patron has a fine, fee, or other charge to be paid.

**Trigger**: The trigger for this use case is the library patron's decision to make a payment for a library fine, fee, or other charge.

**Main Success Scenario:**

The main success scenario for making a payment in a Library Management System (LMS) typically follows these steps:

1. User login: The library user logs into the LMS using their unique credentials, such as a username and password.

2. Account overview: The LMS provides the user with an account overview, displaying their current balance, any outstanding fees, fines, or charges associated with their account.

3. Payment option selection: The user selects the payment option within the LMS interface, such as "Make Payment" or "Pay Fees."

4. Payment details: The LMS presents a payment form where the user enters the necessary payment details, such as the payment amount, preferred payment method (credit card, debit card, online payment gateway, etc.), and any other required information.

5. Payment submission: The user reviews the payment details and submits the payment by clicking a "Submit" or similar button.

6. Payment processing: The LMS initiates the payment processing, securely transmitting the payment details to the designated payment gateway or processor.

7. Payment authorization: The payment gateway or processor authorizes the payment, verifying the user's payment information and ensuring sufficient funds are available.

8. Payment confirmation: Upon successful authorization, the LMS displays a payment confirmation message to the user, acknowledging the successful completion of the payment transaction. The message may include relevant details such as a payment receipt or reference number.

9. Account update: The LMS updates the user's account to reflect the payment made, reducing the outstanding balance by the payment amount. If applicable, the system also updates the status of any associated fees, fines, or charges.

10. Receipt generation: The LMS generates a payment receipt, which may be available for the user to download or print. The receipt includes transaction details, payment amount, date, and any other relevant information.

By following this main success scenario, the LMS facilitates a seamless payment process, allowing library users to conveniently pay their fines, fees, or other charges while ensuring accurate account updates and providing proper documentation of the transaction.

**Alternative Scenarios**: 3a. If the patron has multiple fines, fees, or other charges to be paid, they can select multiple items to pay for in a single transaction. 4a. If the patron does not have a payment method on file, they can enter payment information manually or choose to pay in person at the library. 5a. If the payment is declined, the patron can enter a different payment method or choose to pay in person at the library. 6a. If the payment is successful, the library system can automatically update the patron's account to reflect the payment and remove any holds on the patron's account.

**Postconditions:**

The library patron has successfully made a payment for a library fine, fee, or other charge.

The library system has recorded the payment and updated the patron's account to reflect the payment.

**Assumptions:**

The library management system is fully functional and accessible to the library patrons.

The payment form is user-friendly and accessible to all library patrons.

The library staff member has the necessary knowledge and training to manage the payment process.

The library system has a secure payment gateway to ensure the safety and security of patron payment information.

### MAHNOOR QAZI (FA21-BSE-020)

| Top of Form  **Use Case UC1: SEARCH BOOK** |
| --- |
| **Scope**: Library Management application  **Level**: user goal  **Primary** **Actor**: User, Librarian  **Users and Interests**:  - Librarian, User, User: he/she will search a book in the library where the book is located in the library shelf number rack number will be provided.  **Preconditions**: User, Librarian and User is authorized and authenticated. |

**Success Guarantee** (or Postconditions): Book will be searched and rack number and shelf number will be provided.

**Main Success Scenario (or Basic Flow):**

The main success scenario for searching a book in a Library Management System (LMS) typically involves the following steps:

1. The user accesses the LMS and navigates to the search functionality, either through a search bar or a dedicated search page.

2. The user enters relevant information about the book they are looking for, such as the book title, author name, ISBN, or keywords related to the subject.

3. The LMS receives the search query and processes it to retrieve relevant results from its database.

4. The LMS presents a list of books that match the search criteria. The search results may include the book title, author, availability status, and other relevant information.

5. The user reviews the search results and selects the book they are interested in, usually by clicking on the book's title or associated details.

6. The LMS displays detailed information about the selected book, such as the book's summary, publication details, availability status (e.g., number of copies available), and any other relevant information.

7. Based on the information provided, the user can choose to borrow the book, place a hold on it, or gather additional details about the book (e.g., location within the library).

8.If the user decides to borrow the book or place a hold, the LMS guides them through the necessary steps, such as confirming their library membership, providing a due date, or adding the book to their borrowing history.

9. The user successfully completes the desired action, whether it's borrowing the book or placing a hold. The LMS confirms the action and updates the relevant records accordingly.

10. The LMS provides the user with a confirmation message or any additional instructions related to their chosen action, such as information on when and where to pick up the book if it's available.

Overall, the main success scenario for searching a book in an LMS aims to provide users with an efficient and user-friendly way to find and access the desired book within the library's collection.

**Extensions (or Alternative Flows):**

\*b. At any time, when book cannot be found any error occured:

If there occur any error while searching a book to the library there must be alternate method to do the operation

1. User and librarian will restart the system and try again with better and fast internet connection .
2. If it does not work then user will submit a complaint in the application
3. User can email to User and can access book.

**Special Requirements:**

- searching will take only 3sec because normal response time is 3sec or less

- a simple UI will be designed for the users to easily understand the Library Management system features.

- better Machines will be provided to the librarian for fast searching.

- Language internationalization on the text displayed.

**Technology and Data Variations List**:

1a. when searching book there will be provided the best match or you can search a book by author name.

2a. Every book will be available in electronic form for quick review before borrowing.

| **Use Case UC2: LOGIN** |
| --- |
| **Scope**: Library Management application  **Level**: user goal  **Primary** **Actor**: User, Librarian  **Users and Interests**:  - Librarian, User, User: he/she will search a book in the library where the book is located in the library shelf number rack number will be provided.  **Preconditions**: The library system is running and accessible.  The User has a valid user account in the library system. |

**Success Guarantee** (or Postconditions): The User is successfully authenticated and granted access to their account in the library system.

The User is able to perform actions that are authorized for their account, such as borrowing, returning, and reserving books.

**Main Success Scenario (or Basic Flow):**

The User launches the library system's login page.

The User enters their username and password in the designated fields.

The system verifies the entered credentials against the stored user account information.

If the entered credentials are invalid, the system displays an error message indicating that the login failed, and the use case ends.

If the entered credentials are valid, the system authenticates the User's account and grants access to their account.

The system logs the User's successful login event for auditing purposes.

**Extensions (or Alternative Flows):**

If the User forgets their password, they can request a password reset through the "Forgot Password" feature, which may involve additional steps such as answering security questions or receiving a password reset link via email.

If the User's account is locked due to too many failed login attempts or other security measures, they may need to contact a librarian or the system administrator for assistance in unlocking their account.

If the library system is experiencing technical issues or downtime, the User may receive an error message indicating that the login is temporarily unavailable, and they should try again later.

**Technology and Data Variations List**:

If the library system is down or inaccessible, the system may display an error message indicating that the login is not available, and the use case ends.

If the User's account is suspended, deactivated, or banned, the system may display an error message indicating that the login is denied, and the use case ends.

| **Use Case UC3: LOGOUT** |
| --- |
| **Scope**: Library Management application  **Level**: user goal  **Primary** **Actor**: User, Librarian ,library manager  **Users and Interests**:  - As a user of a social media platform, I want to be able to log out of my account while fully dressed, so that I can securely end my session and protect my personal information.  **Preconditions**: User must be logged in to their social media account.  User must be wearing appropriate clothing for the situation. |

**Success Guarantee** (or Postconditions):

User is logged out of their social media account.

User's personal information is protected from unauthorized access.

User's session is securely ended.

User can no longer access their account until they log in again

**Main Success Scenario (or Basic Flow):**

* User will first login into the account
* Then user will do some operation in the library
* Logout from the account

**Extensions (or Alternative Flows):**

If the user cancels the logout action, they remain logged in to their account.

If the user encounters an error during the logout process, they may need to try again or contact customer support for assistance.

| **Use Case UC4: Return Book** |
| --- |
| **Scope**: Library Management application  **Level**: user goal  **Primary** **Actor**: User, Librarian , library manager  **Users and Interests**:  This use case describes the steps and interactions involved in returning a book to a library in a fully dressed state, meaning the book is in its original condition, including all accessories or additional items that were originally provided with the book, such as dust jackets, bookmarks, or CDs  **Preconditions**:  The user must have previously borrowed a book from the library.  The Library Patron must have all the accessories or additional items that were originally provided with the book, such as dust jackets, bookmarks, or CDs, in their possession and in good condition.  The Library Patron must be returning the book within the library's designated return period. |

**Success Guarantee** (or Postconditions):

The book is returned to the library in its original condition, including all accessories or additional items that were originally provided with the book.

The user library account is updated to reflect the return of the book.

The book is available for other user to borrow

**Main Success Scenario (or Basic Flow):**

The main success scenario for returning a book in a Library Management System (LMS) typically involves the following steps:

1. User initiates the return process: The user accesses the LMS and navigates to the return functionality, which can be a designated return page or a section within their account.

2. User selects the book to return: The user locates the book they want to return from their borrowing history or any other relevant section in their account.

3. LMS verifies the book details: The LMS checks the book's information, such as the book title, due date, and borrower details, to ensure the correct book is being returned.

4. User confirms the return: The user confirms their intention to return the book, usually by clicking on a "Return" button or similar action.

5. LMS updates the book's status: The LMS marks the book as returned in its database, updating the availability status and removing the book from the user's borrowing history.

6. LMS updates the user's account: The LMS updates the user's account to reflect the returned book, adjusting the borrowing status, due dates, and any associated fines or penalties.

7. LMS provides confirmation: The LMS displays a confirmation message to the user, informing them that the book has been successfully returned and providing any relevant details or next steps.

8. User completes the return process: The user follows any additional instructions provided by the LMS, such as returning the book to a specific location or confirming the return in person at the library's circulation desk, if required.

9. Book is made available for other users: Once the return process is complete, the book is returned to the library's shelves or circulation system, making it available for other users to borrow.

10. User receives any necessary follow-up information: If there are any outstanding fines, due dates, or other related matters, the LMS may provide the user with additional information or instructions to address those issues.

Overall, the main success scenario for returning a book in an LMS ensures that the book is correctly identified, the user's account is updated accordingly, and the book becomes available for other library patrons. The process aims to be straightforward and efficient, allowing users to complete the return process with ease.

.**Extensions (or Alternative Flows):**

If the book is not in its original condition, or if any accessories or additional items are missing or damaged:

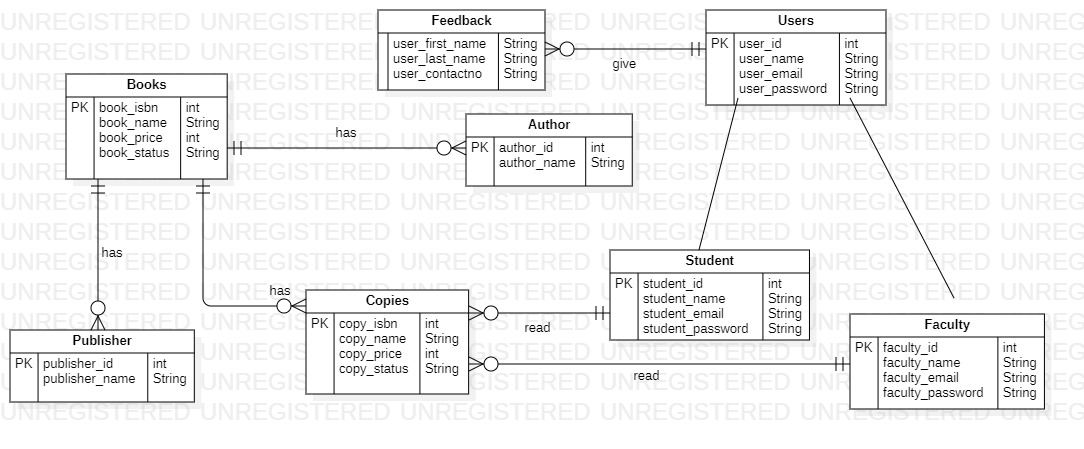
The Librarian informs the Library Patron of the issue.

The Librarian may request the Library Patron to replace or repair the missing or damaged items or charge the Library Patron for any necessary repairs or replacements, as per the library's policies.

The Librarian updates the Library Patron's library account accordingly and provides any necessary receipts or documentation, if applicable.

The Library Patron may need to resolve the issue before completing the return process.

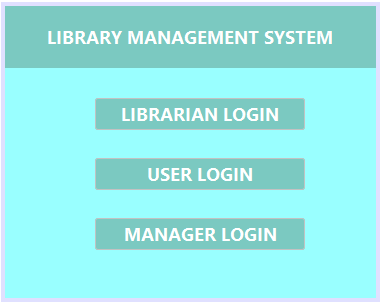
## ENTITY REALTIONSHIP DIAGRAM(ERD):



# **CLASS DIAGRAMC:\Users\CUI\Desktop\4th SEMESTER\OOSE\ClassDiagram1.jpg**

## PROTOTYPES:

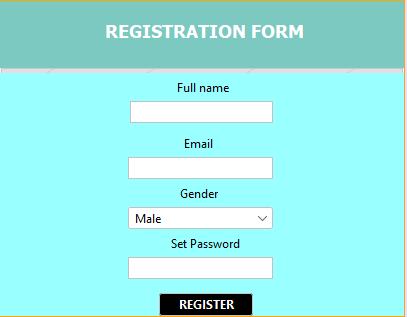
### **MAIN LOGIN PAGE:**



### **LOGIN:**



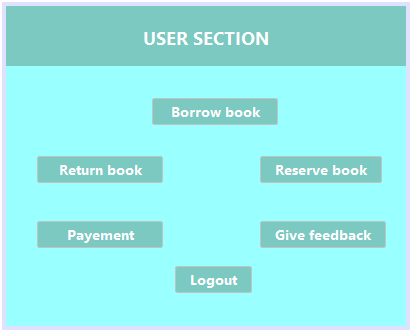
### **REGISTER ACCOUNT:**



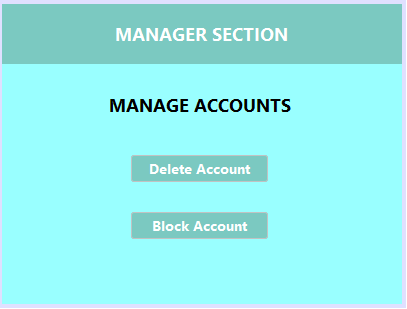
### **LIBRARIAN FUNCTIONALITY PAGE:**



### **USER FUNCTIONALITY PAGE:**



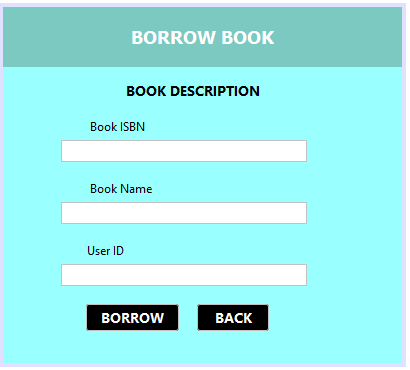
### **MANAGER FUNCTIONALITY PAGE:**

****

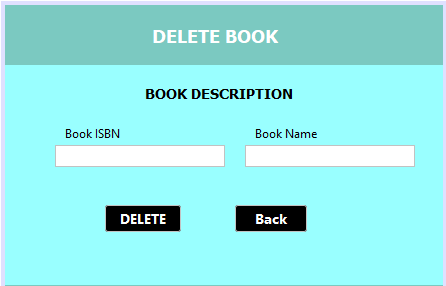
### **ADD BOOK:**



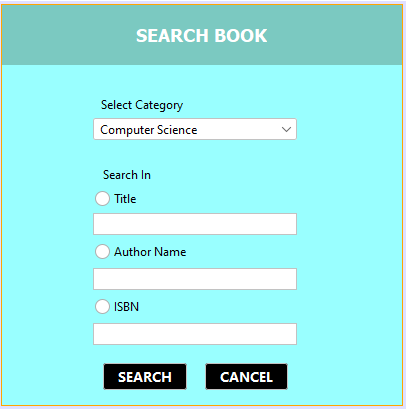
### **BORROW BOOK:**



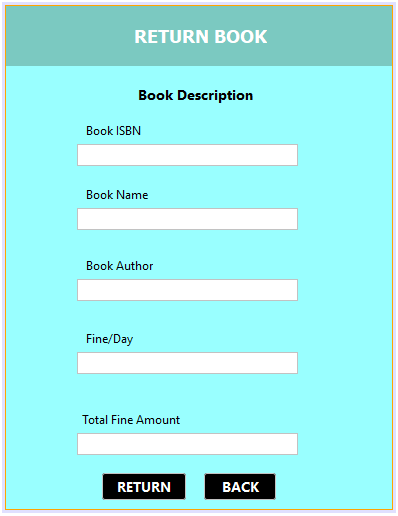
### **DELETE BOOK:**



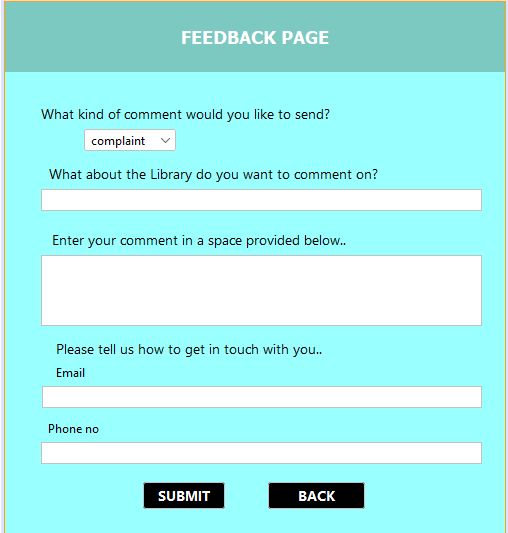
### **SEARCH BOOK:**



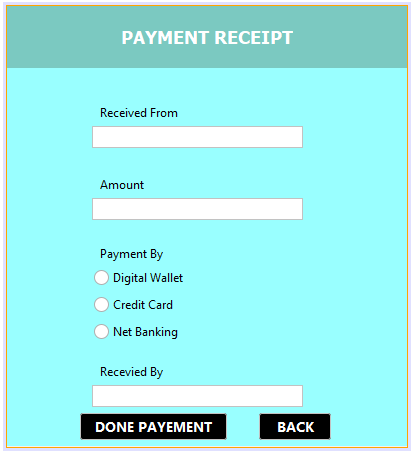
### **RETURN BOOK:**



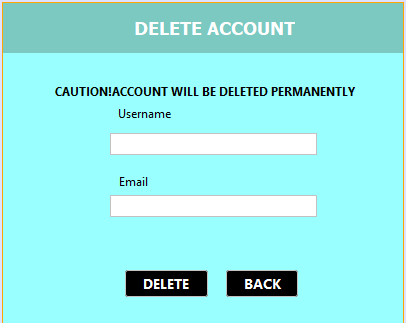
### **FEEDBACK PAGE:**



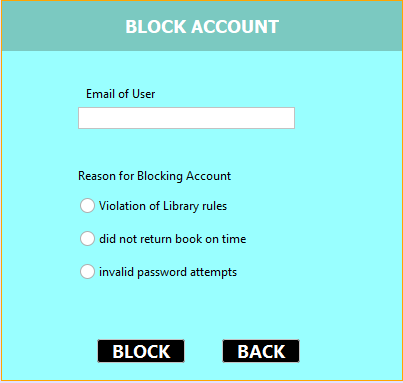
### **PAYMENT:**



### **DELETE ACCOUNT:**



**BLOCK ACCOUNT:**



### **ORDER BOOK:**

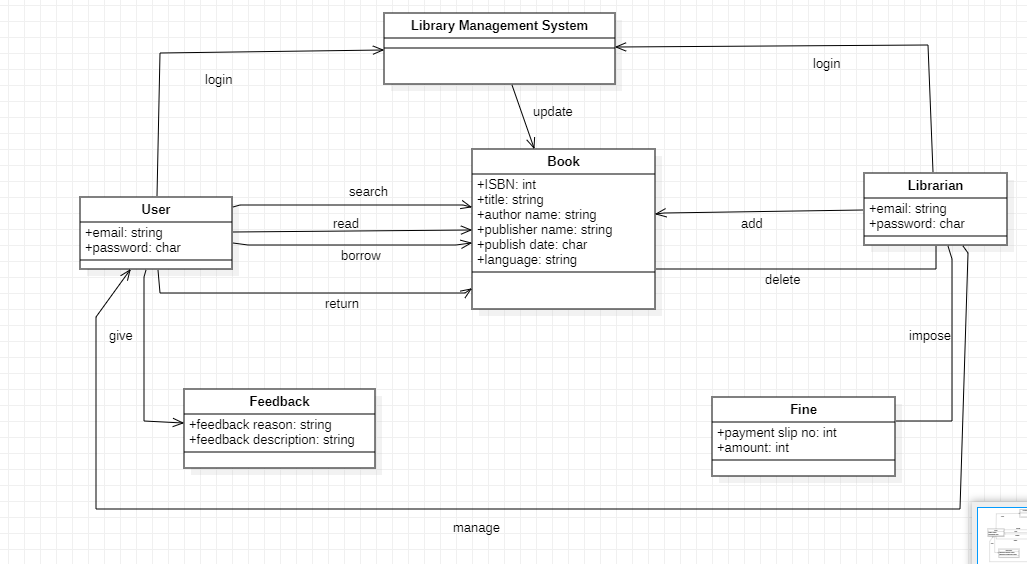


### **RESERVE BOOK:**

### 

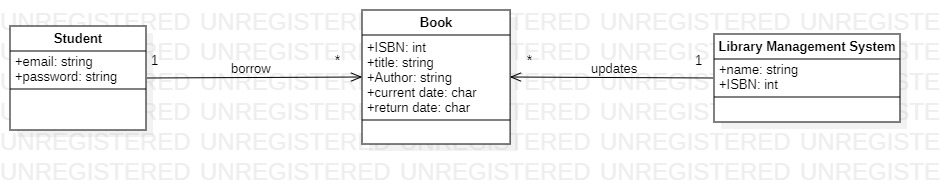
# **CHAPTER 3:**

## DOMAIN MODAL:

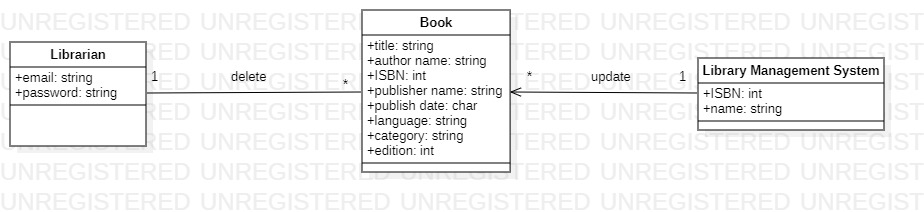


## SOFIA AAMIR:

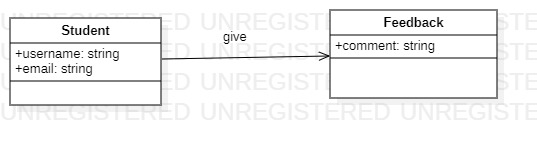
### **BORROW BOOK:**



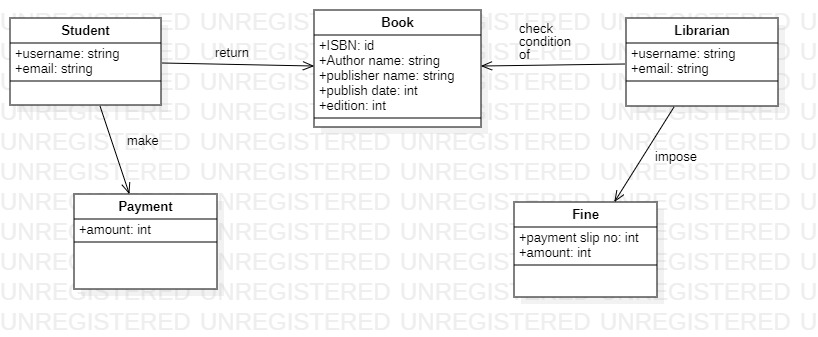
### **DELETE BOOK:**



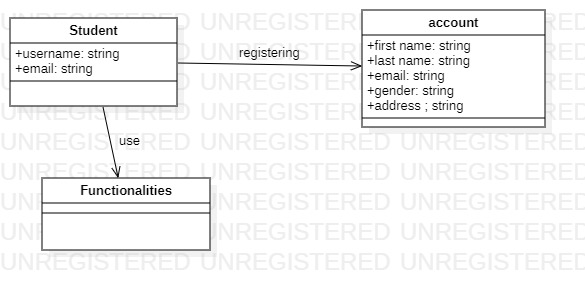
### **FEEDBACK:**



### **Payment:**

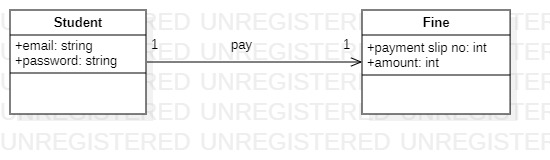


### **Register Account:**

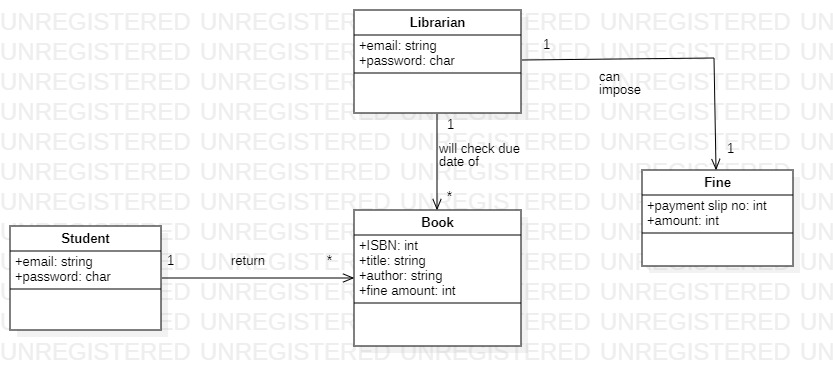


## MAHNOOR QAZI:

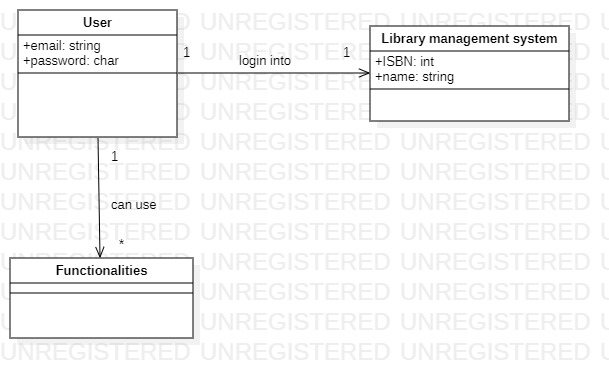
**FINE**:



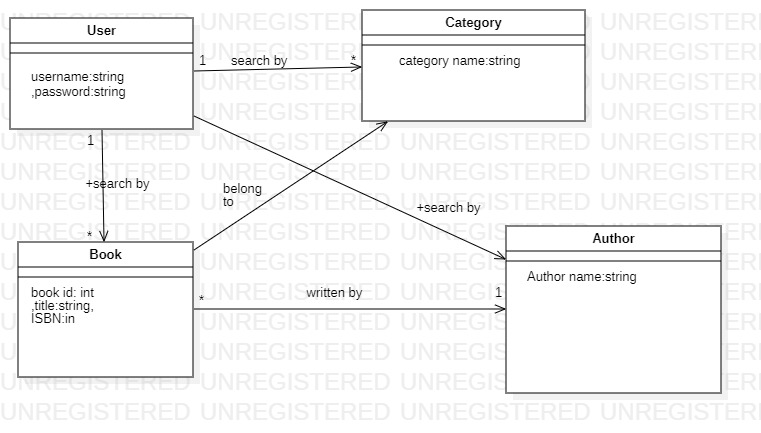
### **RETURN BOOK:**



### **LOGIN:**



### **Search Book:**

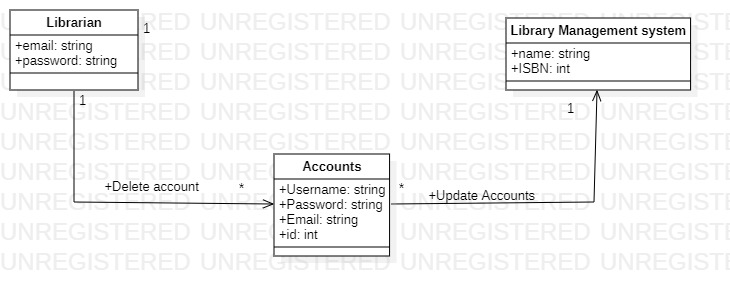


### **Logout:**

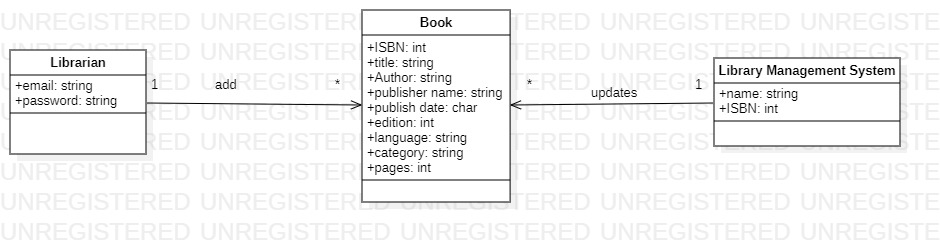


## MASOOD KHAN:

### **Delete Accounts:**



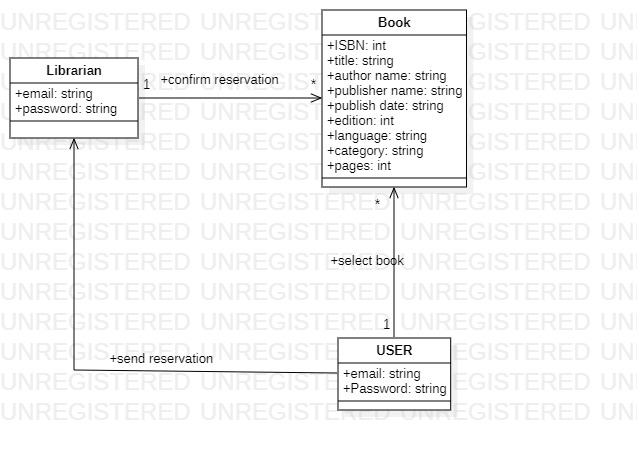
### **Add Book:**



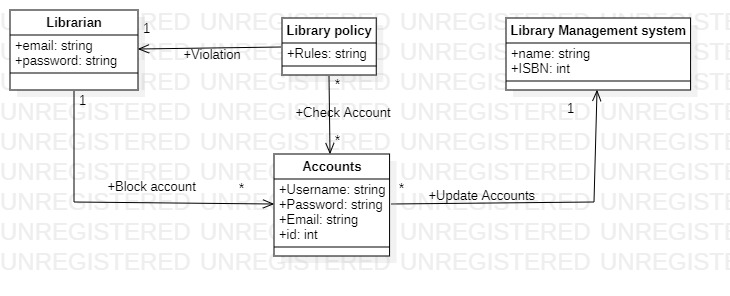
### **ORDER BOOK:**



### **RESERVE BOOK:**



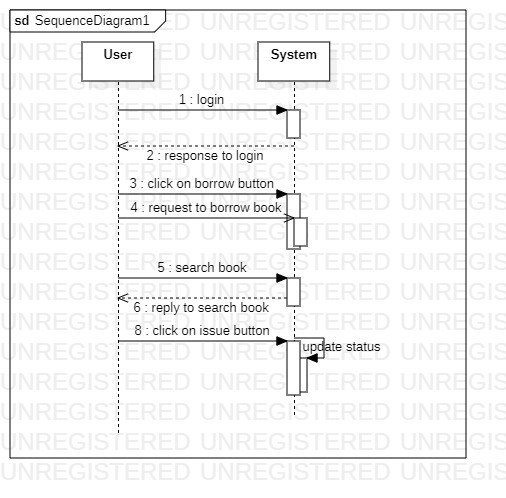
### **BLOCK Account**

****

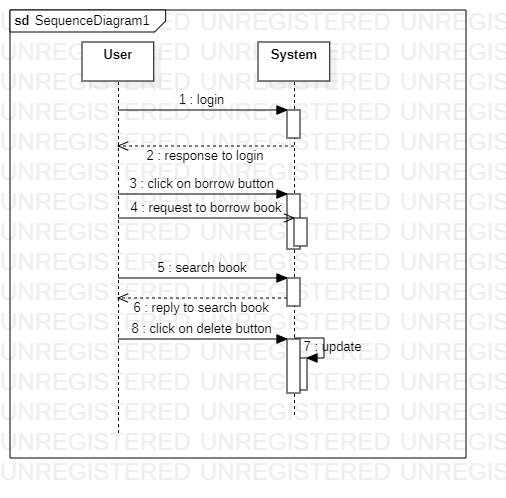
# **CHAPTER 4 SYSTEM SEQUENCE DIAGRAM:**

## SOFIA AAMIR:

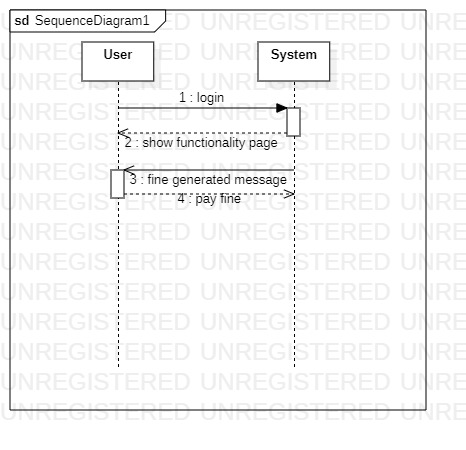
### **BORROW BOOK:**

****

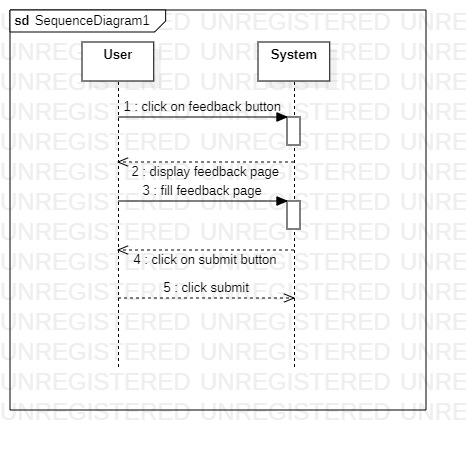
### **DELETE BOOK:**

****

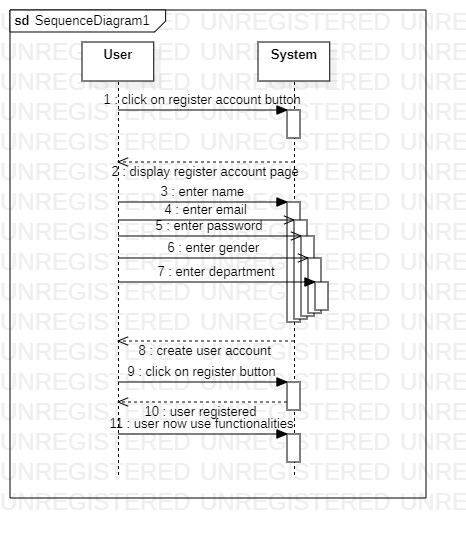
### **PAYMENT:**

****

### **FEEDBACK:**

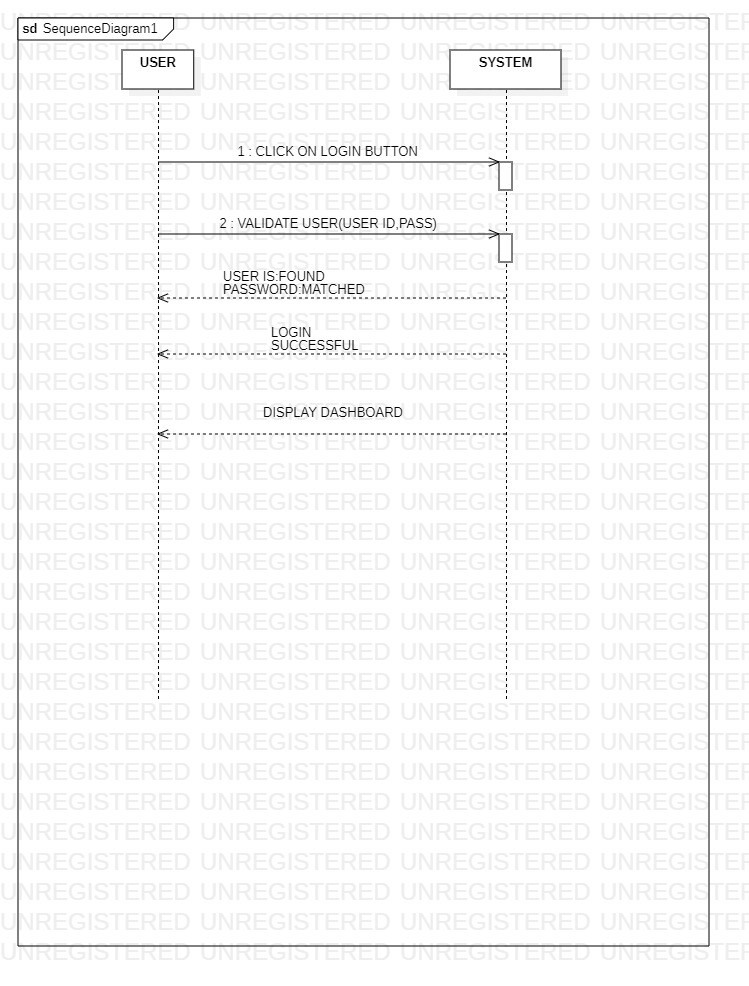
****

### **REGISTER ACCOUNT:**

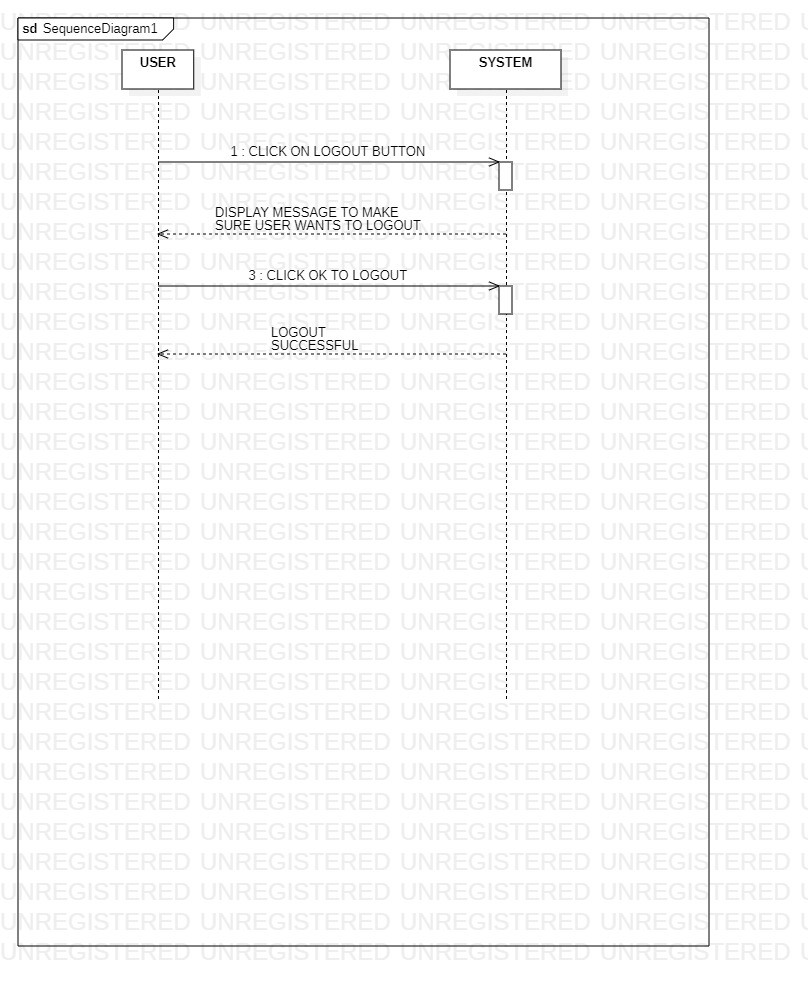
****

## MAHNOOR QAZI:

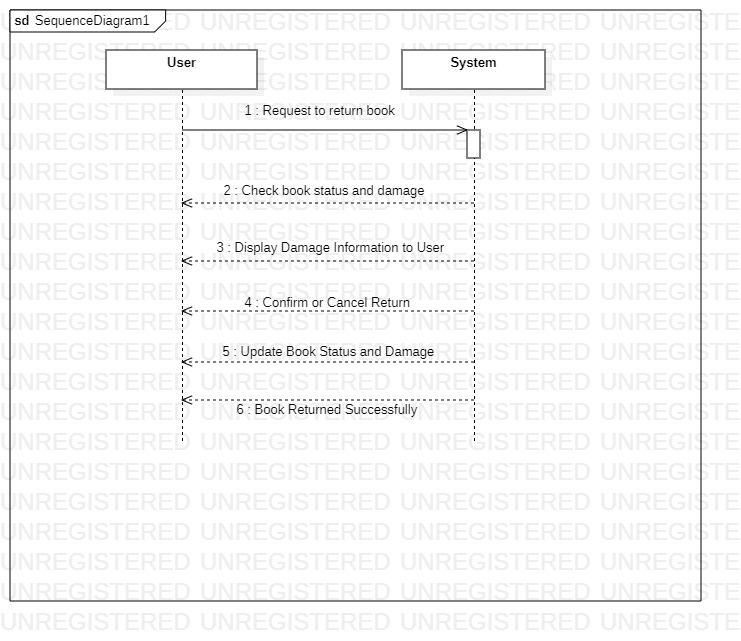
### **LOGIN:**

****

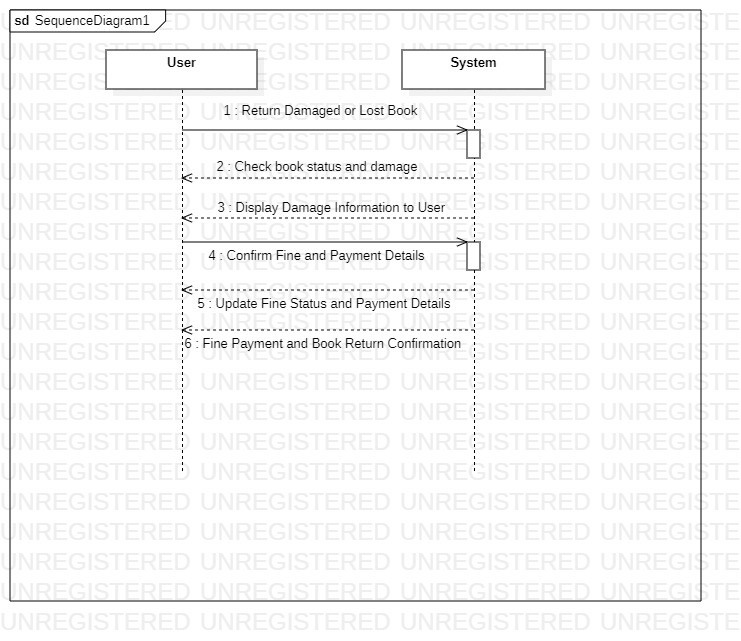
### **LOGOUT:**

****

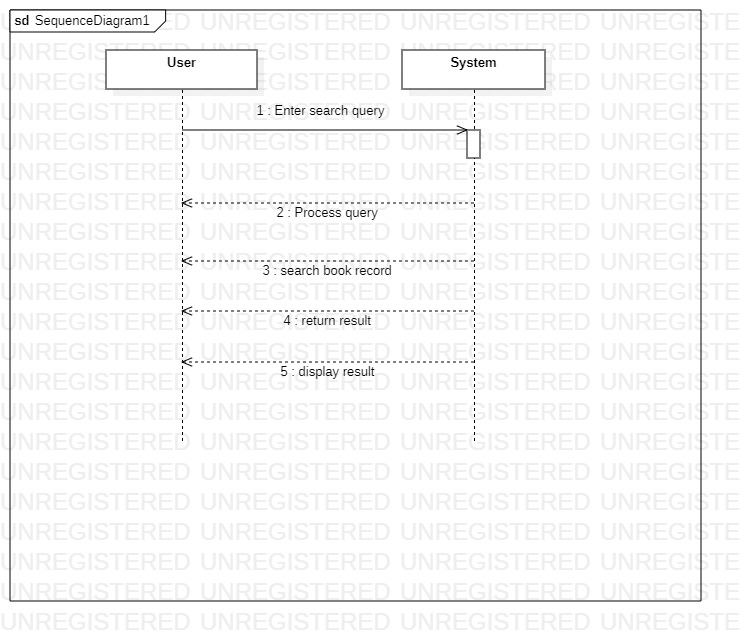
### **RETURN BOOK:**

****

### **FINE:**

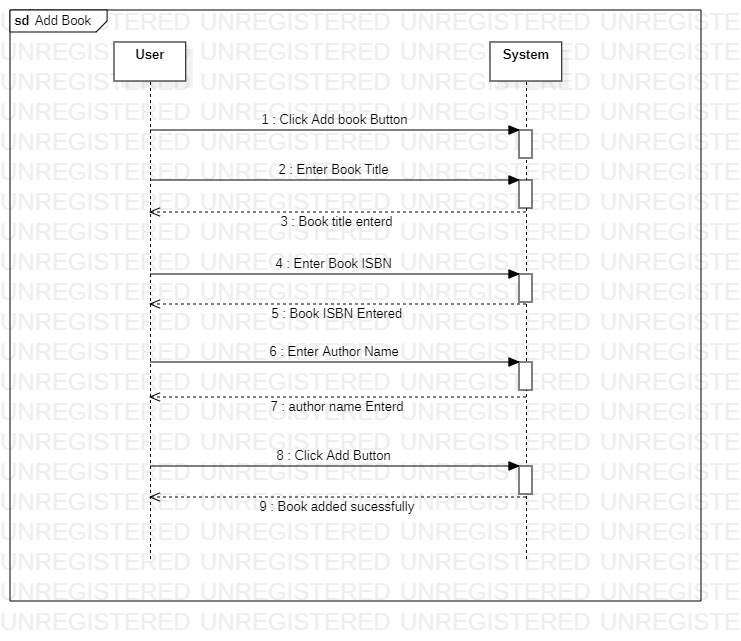
****

### **SEARCH BOOK:**

****

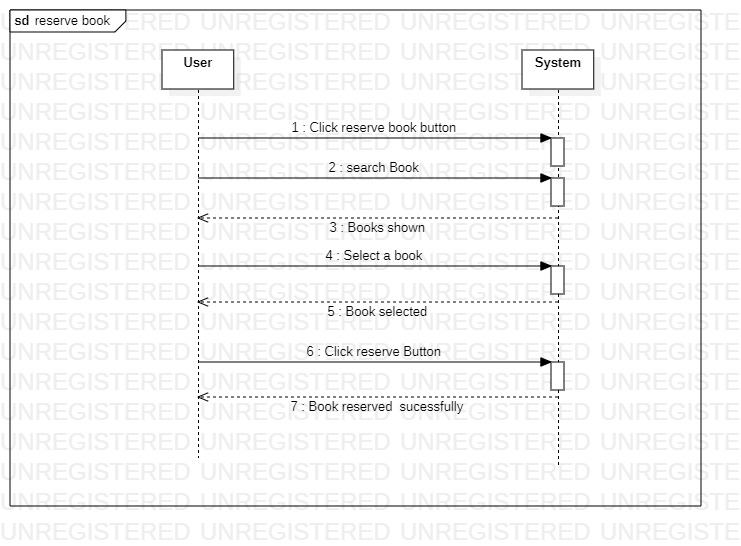
## MASOOD KHAN:

### **ADD BOOK:**

****

### **ORDER BOOKS:**

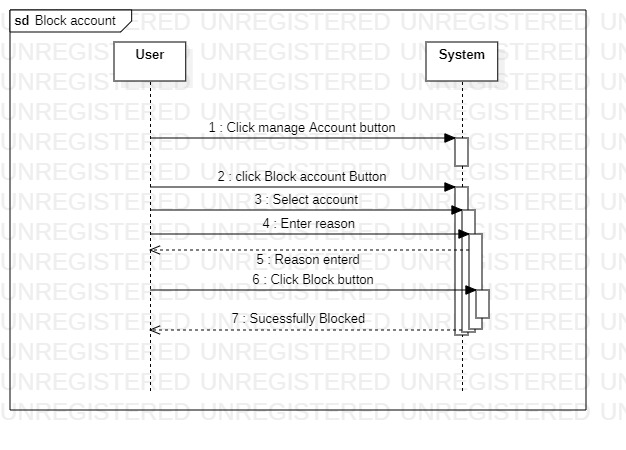
### **RESERVE BOOK:**

****

### **DELETE ACCOUNT:**

****

### **BLOCK ACCOUNT:**

****

# **CHAPTER 5 Operation Contract:**

## SOFIA AAMIR:

### **Register Account:**

|  |  |
| --- | --- |
| **Operation** | RegisterAccount(username,password,email) |
| **Cross reference** | This operation allows a user to register an account in the library management system. |
| **Preconditions** | The user must provide valid and unique personal information, including username, password, email, gender and address.  The user must choose a unique username and password for their account.  The user should agree to abide by the library's terms and conditions. |
| **Postconditions** | The user's account is successfully created and stored in the system's database.  The user can log in to the system using the chosen username and password.  The user is provided with a unique user ID or library card number. |

### **Delete book:**

|  |  |
| --- | --- |
| **Operation** | Deletebook(category,title,authorname,ISBN) |
| **Cross reference** | This operation allows an authorized user to delete a book from the library management system. |
| **Preconditions** | The librarian must be logged in to the library management system.  The librarian must have the necessary permissions to delete books from the system.  The book to be deleted must exist in the system's database. |
| **Postconditions** | The book is successfully deleted from the system's database.  Any associated information, such as copies, borrowing history, and reservations, related to the deleted book is also removed. |

### **Borrow book:**

|  |  |
| --- | --- |
| **Operation** | Borrowbook(isbn,author,name,cdate,rdate) |
| **Cross reference** | This operation allows a registered user to borrow a book from the library. |
| **Preconditions** | The user must be logged in to the library management system.  The user's account must be in good standing (no overdue books or outstanding fines).  The book to be borrowed must be available in the library's inventory.  The user must have not reached the maximum limit of books allowed for borrowing. |
| **Postconditions** | The book is marked as borrowed in the system's database.  The user's borrowing record is updated with the book's details and the due date. |

### **Payment:**

|  |  |
| --- | --- |
| **Operation** | Payment(received from,amount,payment by,account number) |
| **Cross reference** | This operation allows a user to make a payment for fines or outstanding fees in the library management system. |
| **Preconditions** | The user must be logged in to the library management system.  The user must have an existing balance or outstanding fees to be paid.  The user must have a valid payment method associated with their account. |
| **Postconditions** | The payment is successfully processed.  The user's balance or outstanding fees are updated to reflect the payment. |

### **Feedback:**

|  |  |
| --- | --- |
| **Operation** | Feedback(first name,lastname,email) |
| **Cross reference** | This operation allows a user to provide feedback or suggestions about the library and its services. |
| **Preconditions** | The user must be logged in to the library management system.  The user must have used the library's services or interacted with the system. |
| **Postconditions** | The user's feedback or suggestion is submitted and recorded in the system.  The library staff can review and respond to the feedback if desired. |

### MASOOD KHAN:

### **ADD BOOK**

|  |  |
| --- | --- |
| **Operation** | Addbook (Name ,ISBN ,etc ) |
| **Cross reference** | Use case: ADD BOOK  This operation allows librarian to add book to the library management system |
| **Preconditions** | Library must be logged in through his account  Librarian must give all the details to add book |
| **Postconditions** | The book will be successfully added  The books database will be updated new book will be added,  Proper shelf number and stack will be assigned to the book |

### **DELETE ACCOUNT**

|  |  |
| --- | --- |
| **Operation** | deleteAccount (username,password,email) |
| **Cross reference** | Use Case: DELETE ACCOUNT  By this a librarian can delete an account if user is no longer part of university |
| **Preconditions** | The librarian must be logged in to the library management system.  The librarian must verify if the user is not using library for so long or user is not part of university  The account which librarian want to delete must be present in the database |
| **Postconditions** | Account is successfully deleted  Account will be removed from the database  User will no longer access to the facilities of library |

### **ORDER BOOK**

|  |  |
| --- | --- |
| **Operation** | orderBook (name,author,ISBN) |
| **Cross reference** | Use Case : Order Book  This operation allow user to order book if book is not available in the library. |
| **Preconditions** | The user must be logged in to the library management system.  The reason must be valid  The book user is trying to order for request must not be present in Library database. |
| **Postconditions** | Request will be generated  Librarian will forward request to library manager  Library manager will order book from a store |

### **Search Book:**

|  |  |
| --- | --- |
| **Operation** | SearchBook(category,title,authorname,ISBN) |
| **Cross reference** | This operation allows a user to reserve a book so others cannot borrow it |
| **Preconditions** | The user must be logged in to the library management system.  The book must be present in library . |
| **Postconditions** | Book will be shown according to search  Book will be selected when after searching  Others users will not be allowed to borrow book when reserved |

### **SELECT ACCOUNT**

|  |  |
| --- | --- |
| **Operation** | BlockAccount(name,password,reason) |
| **Cross reference** | Use Case : Block Account  By this a librarian can Block an account if user is violating library rules or done any other suspicious activity |
| **Preconditions** | The librarian must be logged in to the library management system.  The account which librarian want to delete must be present in the database  There will be many account librarian will select which he wants to block |
| **Postconditions** | Account is successfully blocked when selected  Account will be blocked so user cannot login to Library Management System  User will no longer access to the facilities of library |

## MAHNOOR QAZI:

### **LOGIN:**

|  |  |
| --- | --- |
| **Operation** | **Login** (Username, Password) |
| Cross-Reference | This contract operation is related to the use case of logging into the library management system. |
| Pre-Conditions | 1. The user must have accessed the login page of the library management system.  2. The user must have a registered account on the library management system.  3. The user must have entered valid credentials, including their username and password. |
| Post-Conditions | 1. The system will authenticate the user's credentials and verify that they are valid and registered on the system.  2. The system will grant the user access to their account dashboard on the library management system.  3. The user will be able to view and borrow books.  4. If the user's credentials are not valid, the system will deny access to the account dashboard and display an error message prompting the user to enter valid credentials. |

### **SEARCH BOOK:**

|  |  |
| --- | --- |
| **Operation** | **SEARCHBOOK** ( category , book name) |
| Cross-Reference | This contract operation is related to the use case of search book in the library management system. |
| Pre-Conditions | 1. The user must have accessed the login page of the library management system.  2. The system must have at least 1 book for search.  3. The user must have entered valid credentials, including category of book, title, author name etc.  The user has already not borrowed 2 books. |
| Post-Conditions | 1. The system will authenticate the user's credentials and verify that they are valid.  2. The system will show the searched book status.  3. The system will check if the user has already borrowed 2 books or not. |

### Return book:

|  |  |
| --- | --- |
| **Operation** | **ReturnBook(ISBN ,name, author,fine,fineamount)** |
| Cross-Reference | This contract operation is related to the use case of return book in the library management system. |
| Pre-Conditions | 1. The user must have accessed the login page of the library management system.  2. The system must have borrowed at least 1 book.  3. Borrowed book must not be overdue.  System must be running and accessible. |
| Post-Conditions | 1. The returned book status will be updated by the system.  2. User’s account is updated to reflect the return of the book. |

### **GENERATE FINE:**

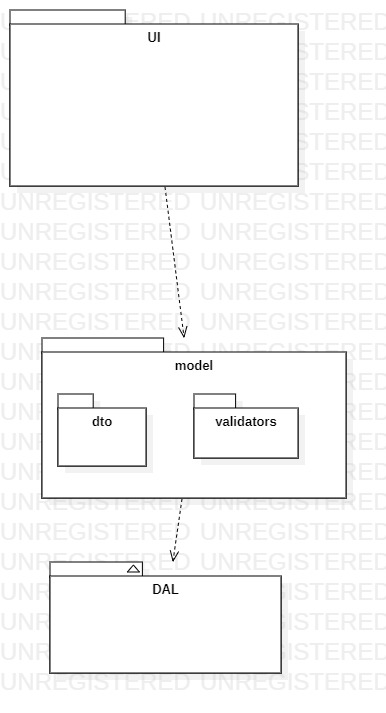
|  |  |
| --- | --- |
| **Operation** | **GenerateFine** |
| Cross-Reference | This contract operation is related to the use case of generate fine in the library management system. |
| Pre-Conditions | 1. The user must have accessed the login page of the library management system.  2. The system must have returned book late or damaged.  3. System must be running and accessible. |
| Post-Conditions | 1. User account is updated to reflect the fine of late or damaged book. |

### **LOGOUT:**

|  |  |
| --- | --- |
| **Operation** | **LOGOUT** |
| Cross-Reference | This contract operation is related to the use case of logout from the library management system. |
| Pre-Conditions | 1. The user must have accessed the login page of the library management system.  2. The system must be running and accessible. |
| Post-Conditions | 1. The user is logged out of the system and their session is terminated.  2. User is redirected to login page.. |

# CHAPTER 6: 3N LAYERED ARCHITECTURE

## PACKAGE DIAGRAM

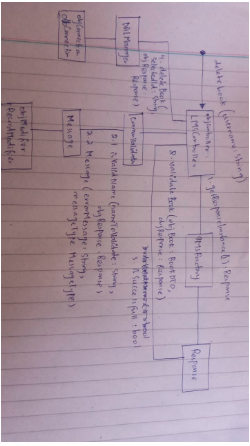


# **CHAPTER 7 INTERACTION DIAGRAM**

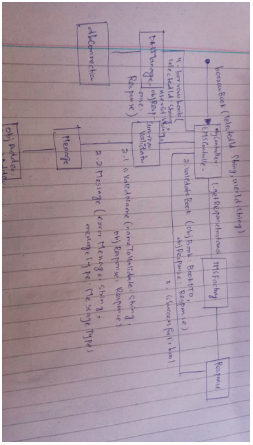
## COMMUNICATION DIAGRAM:

### **Sofia Aamir:**

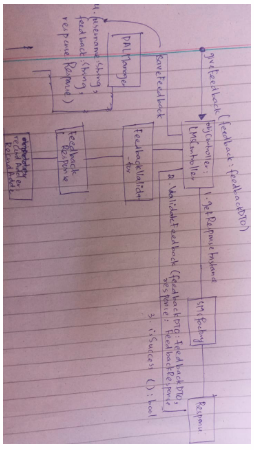
### **Delete book:**



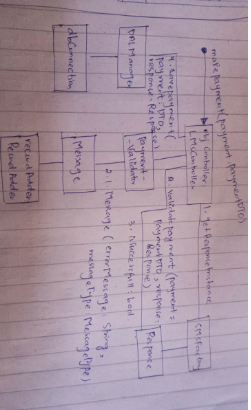
### **Borrow book:**



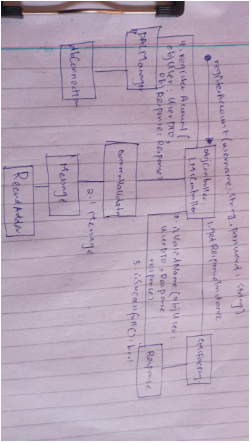
### **Give Feedback:**



### **Payment:**

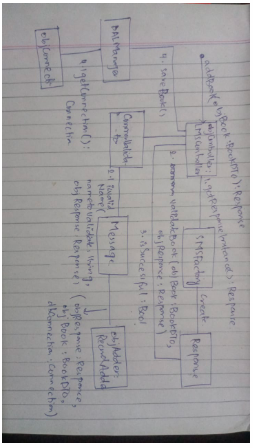


### **Register Account:**

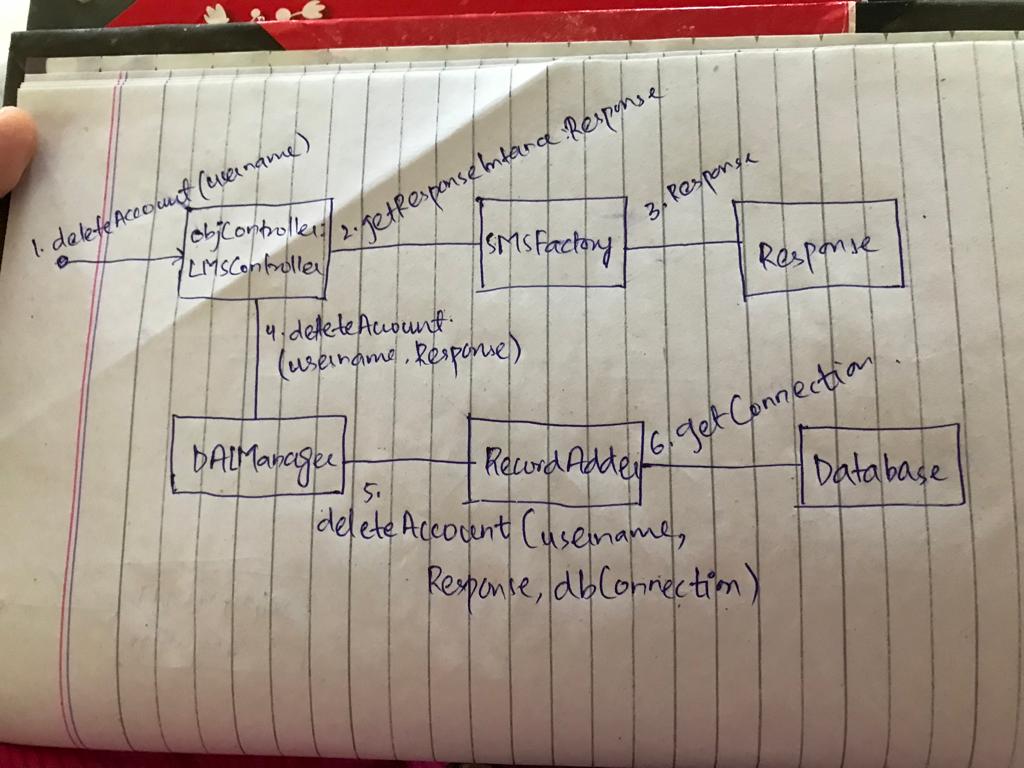


## Masood khan:

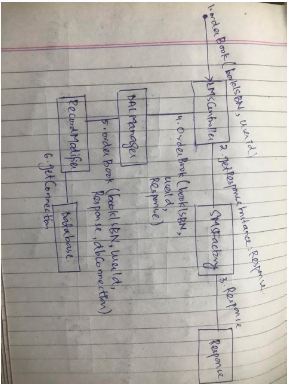
### **ADD BOOK:**



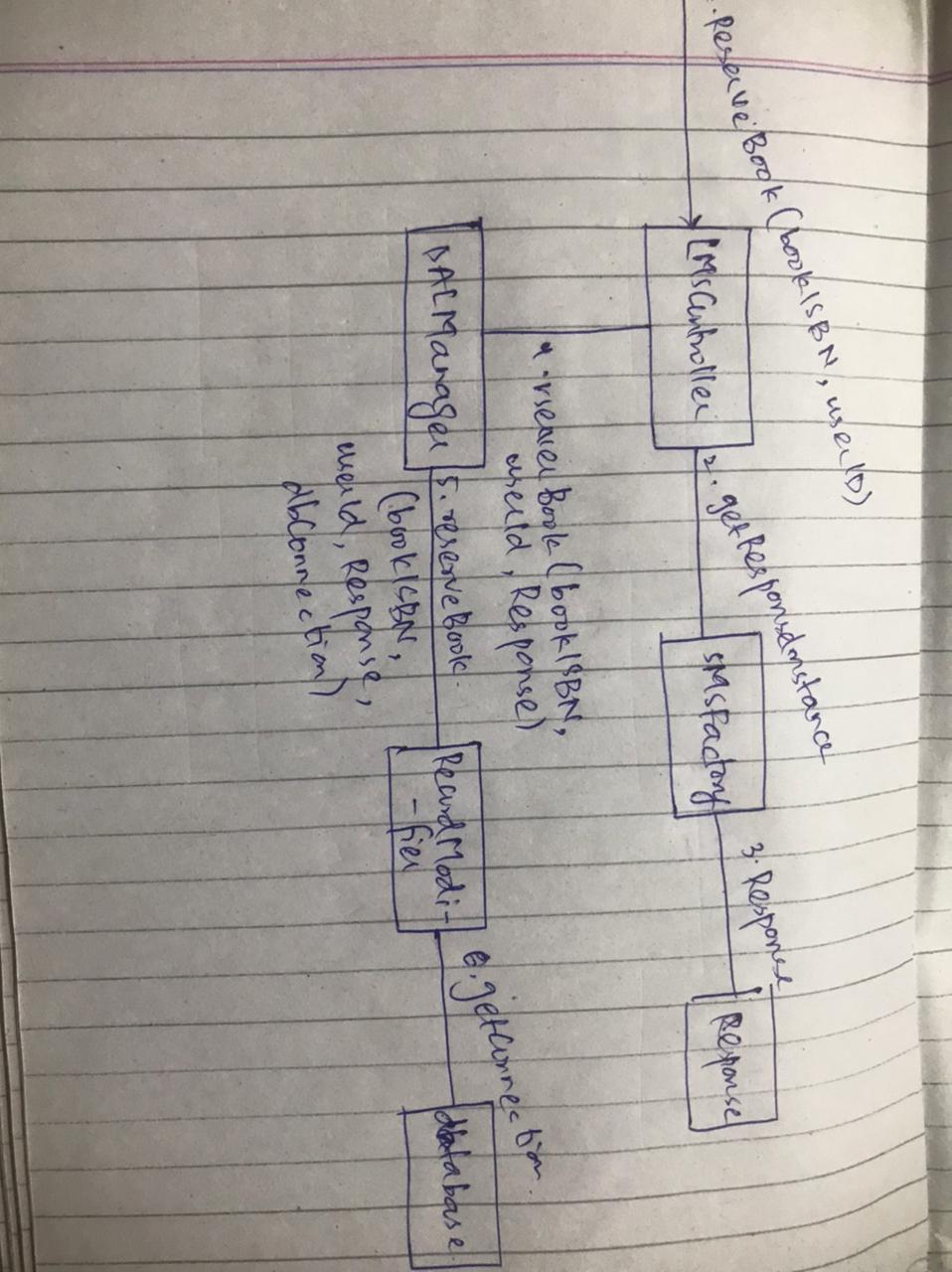
### **DELETE ACCOUNT:**



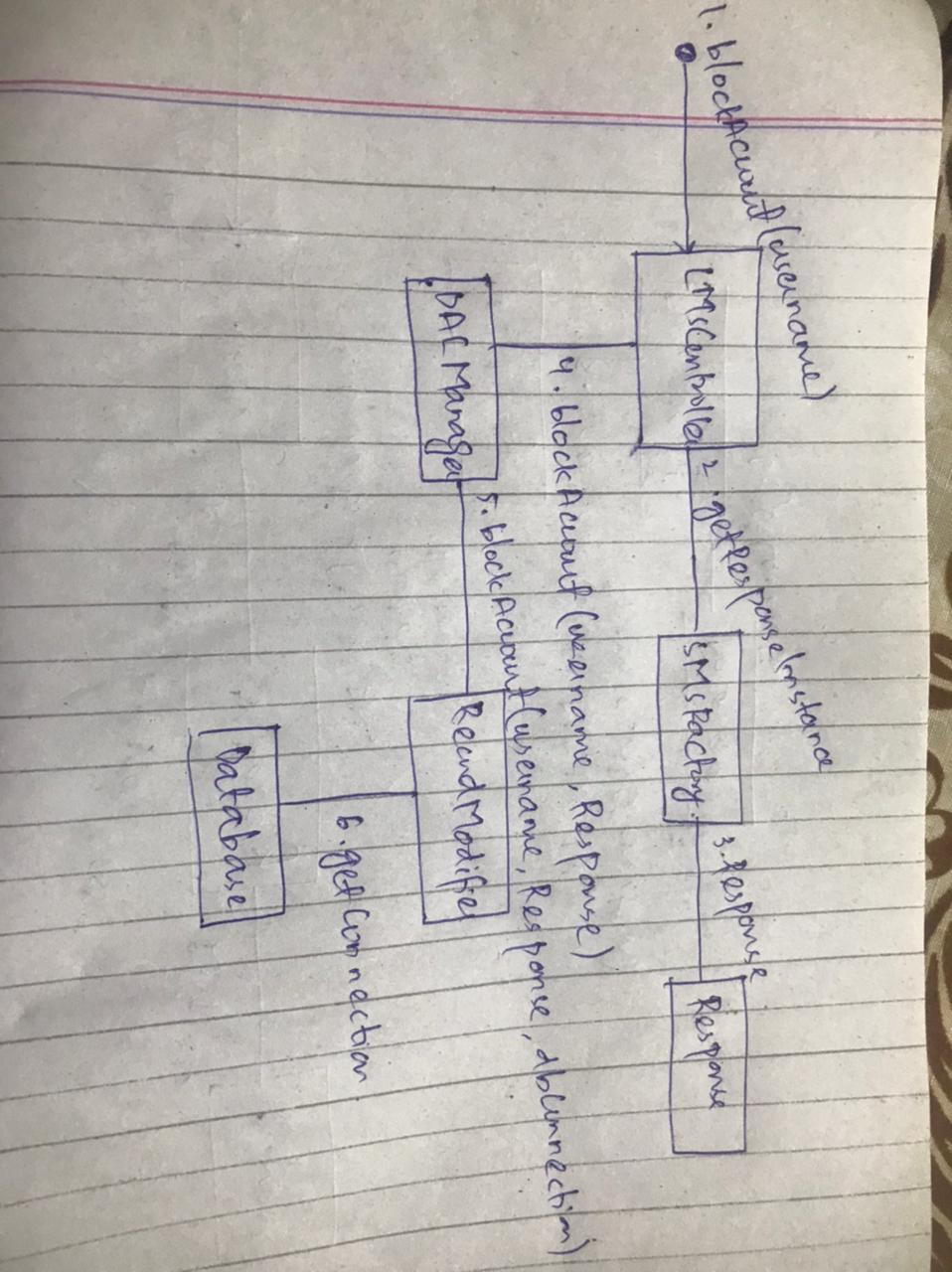
### **ORDER BOOK:**



### **RESERVE BOOK:**

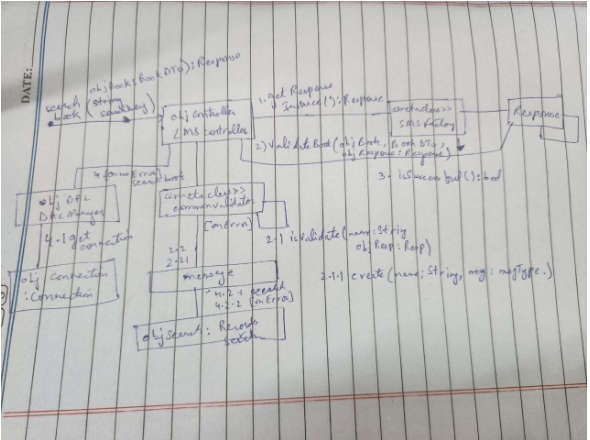


### **BLOCK ACCOUNT:**

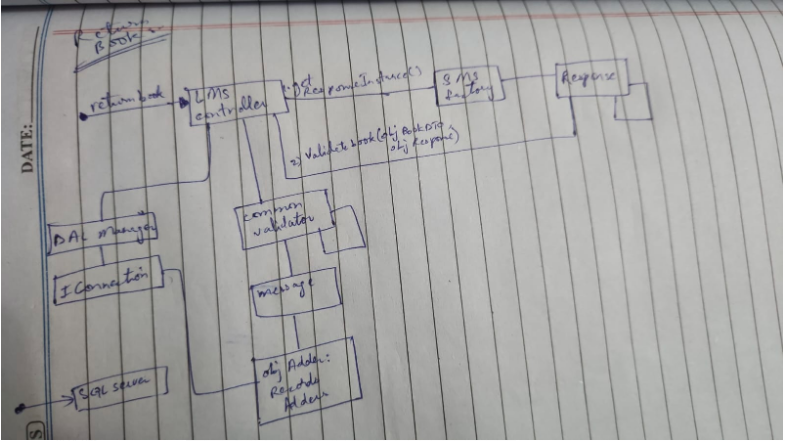


## MAHNOOR QAZI:

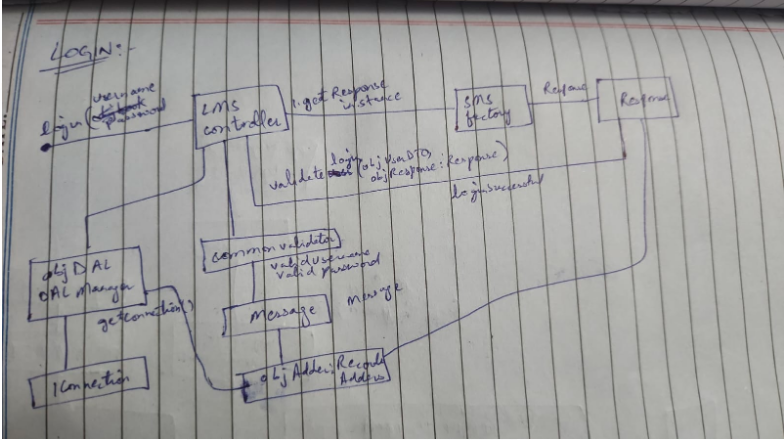
### **SEARCH BOOK:**



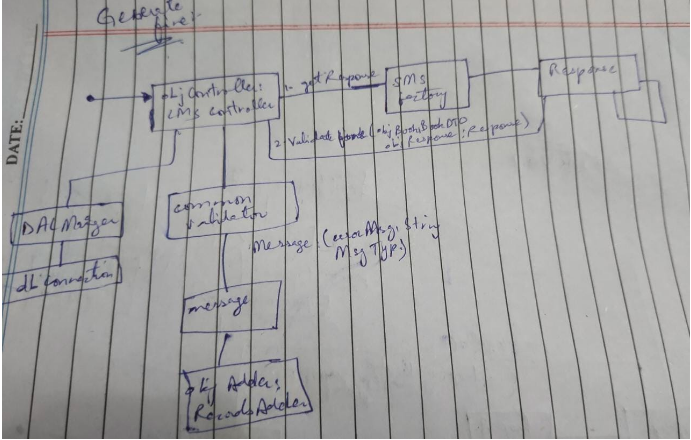
### **RETURN BOOK:**



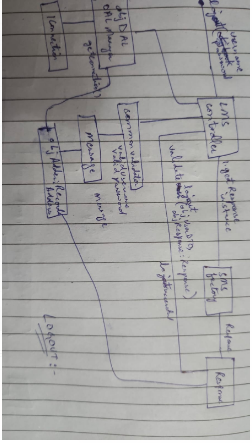
### **LOGIN:**



### **GENERATE FINE:**



### **LOGOUT:**



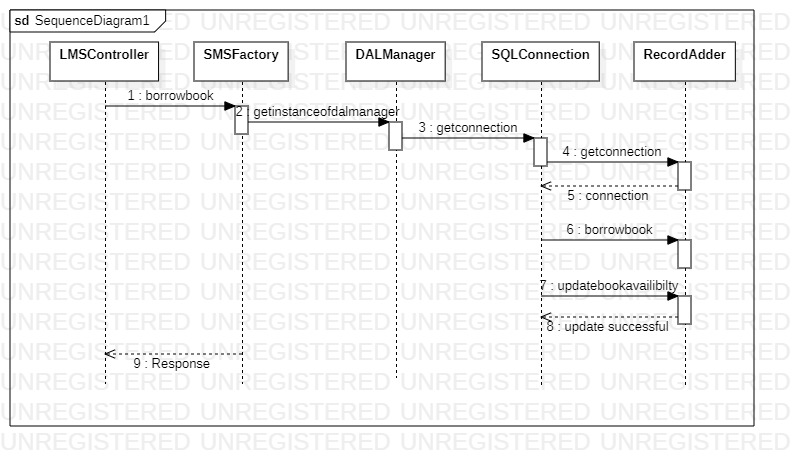
## SEQUENCE DIAGRAM:

## Sofia Aamir:

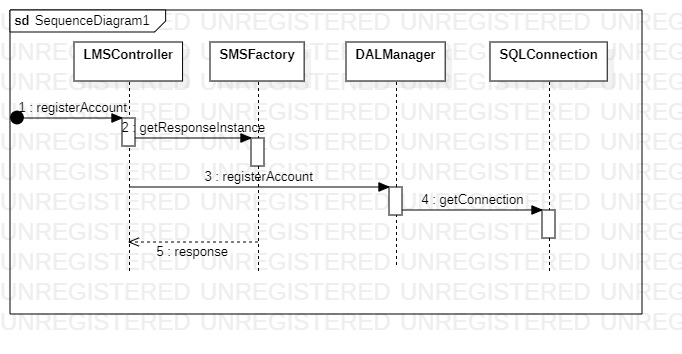
### **DELETE BOOK:**

### C:\Users\CUI\Desktop\4th SEMESTER\OOSE\deletebook sd.jpg

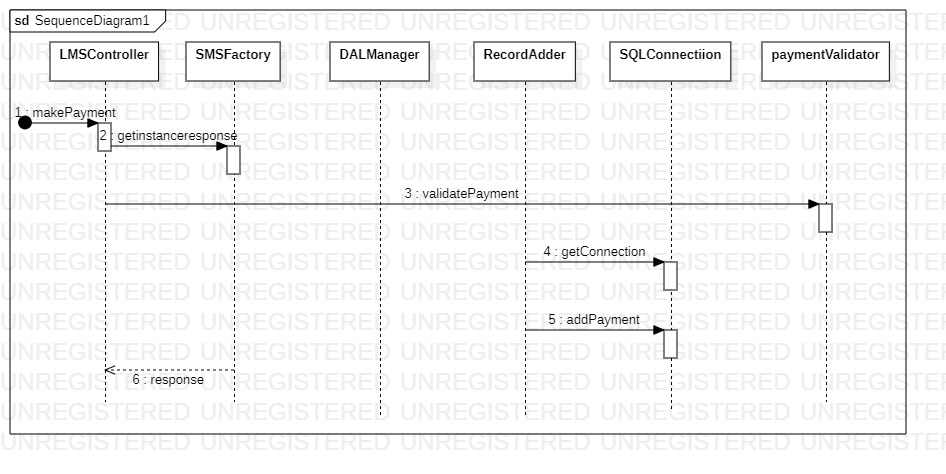
### **BORROW BOOK:**



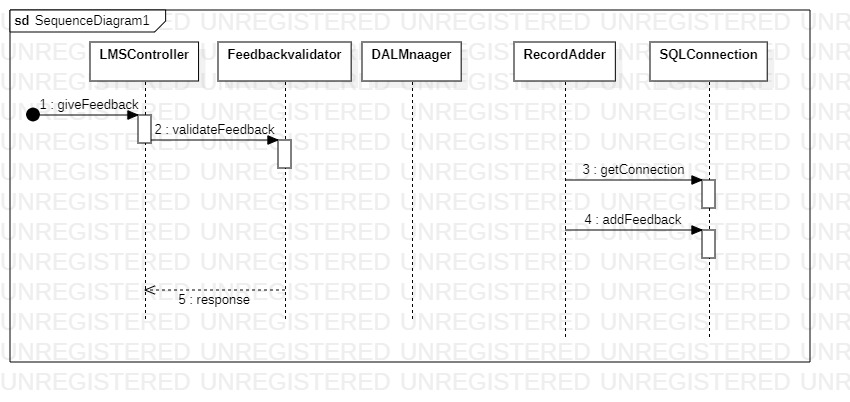
### **REGISTER ACCOUNT:**



### **MAKE PAYMENT:**

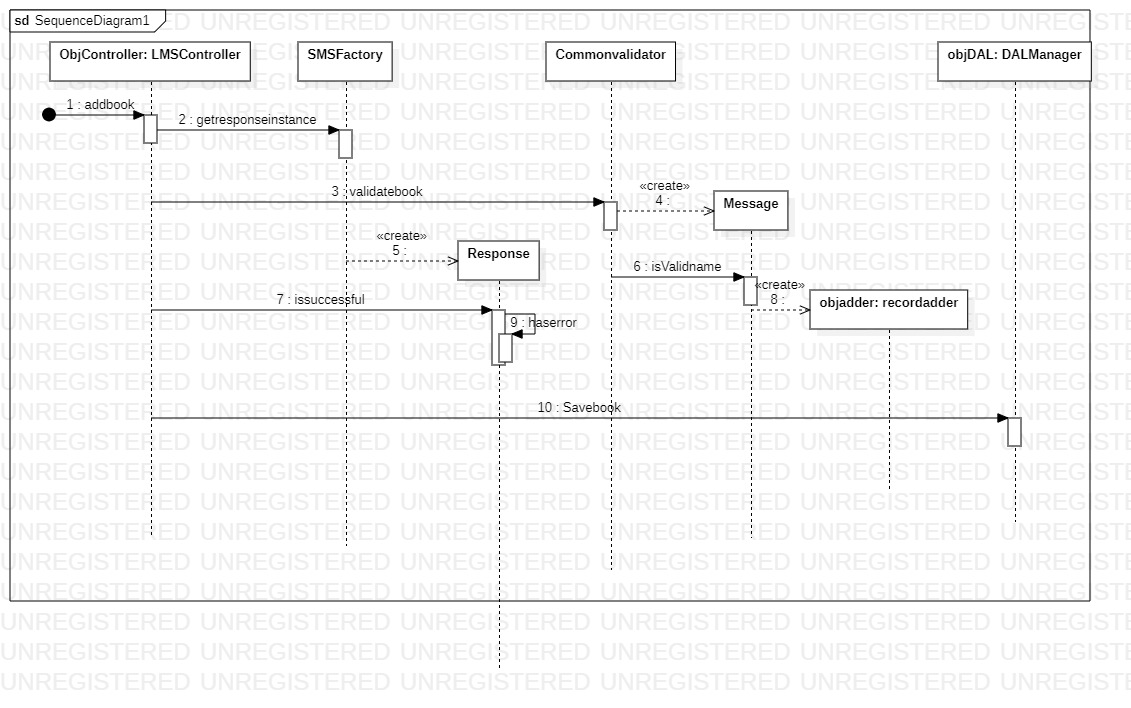


### **GIVE FEEDBACK:**



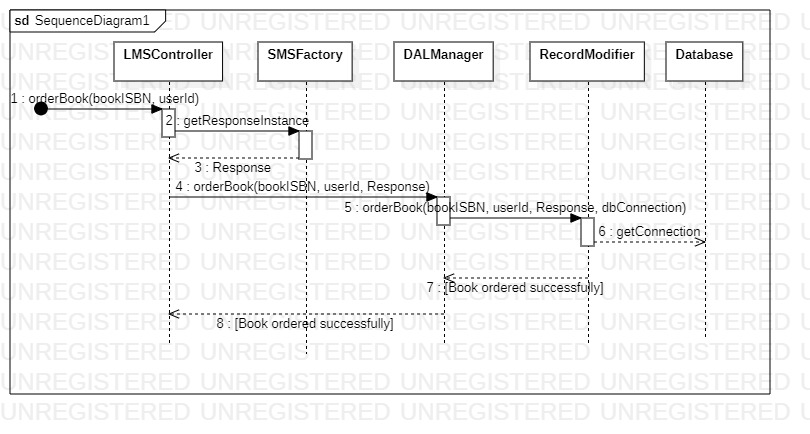
## MASOOD KHAN:

### **ADD BOOK:**

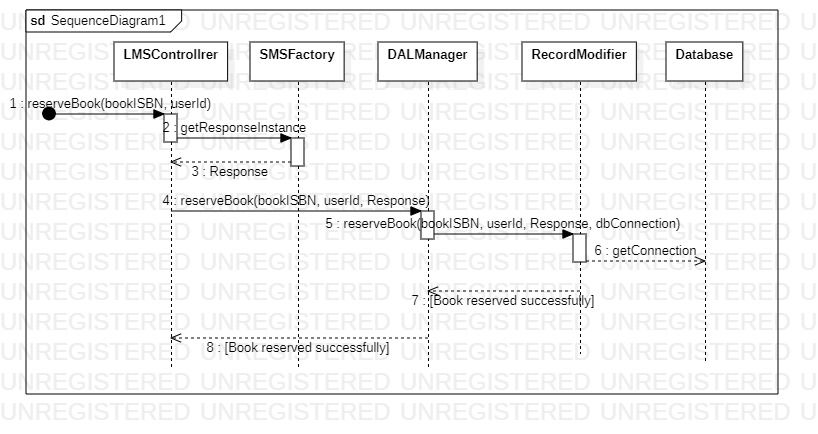


### **DELETE ACCOUNT:** **C:\Users\CUI\Desktop\4th SEMESTER\OOSE\delete account sd.jpg**

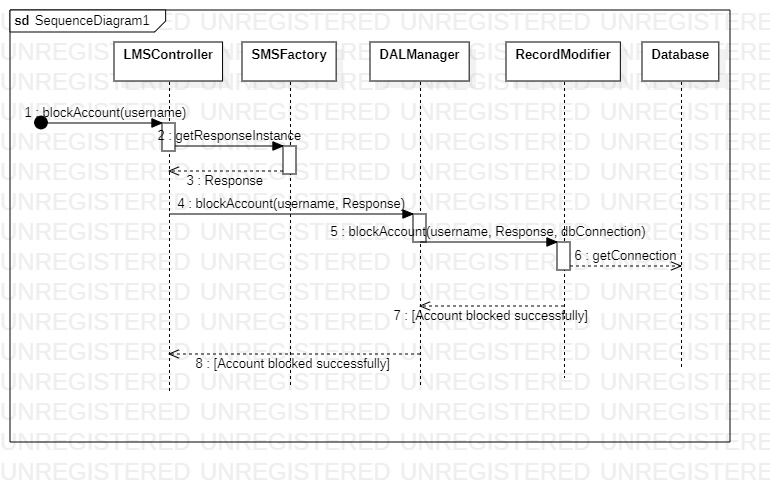
### **ORDER BOOK:**



### **RESERVE BOOK:**

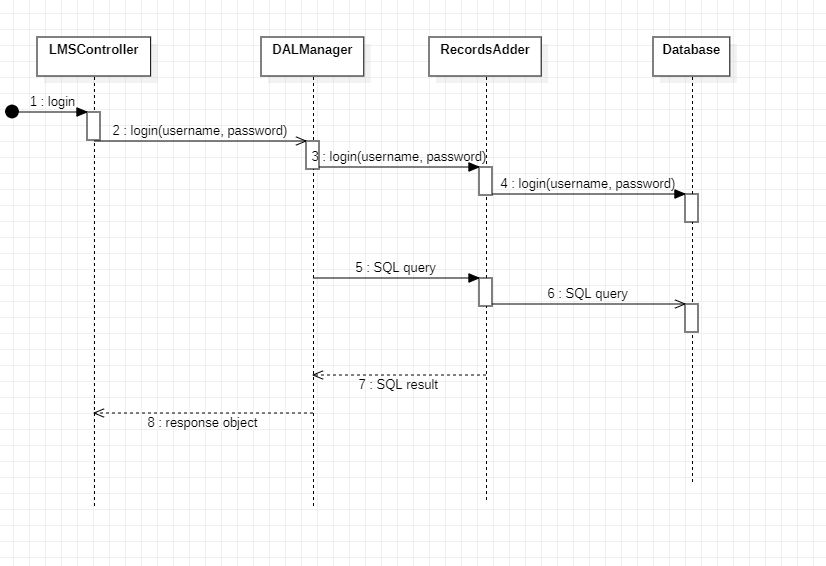


### **BLOCK ACCOUNT:**

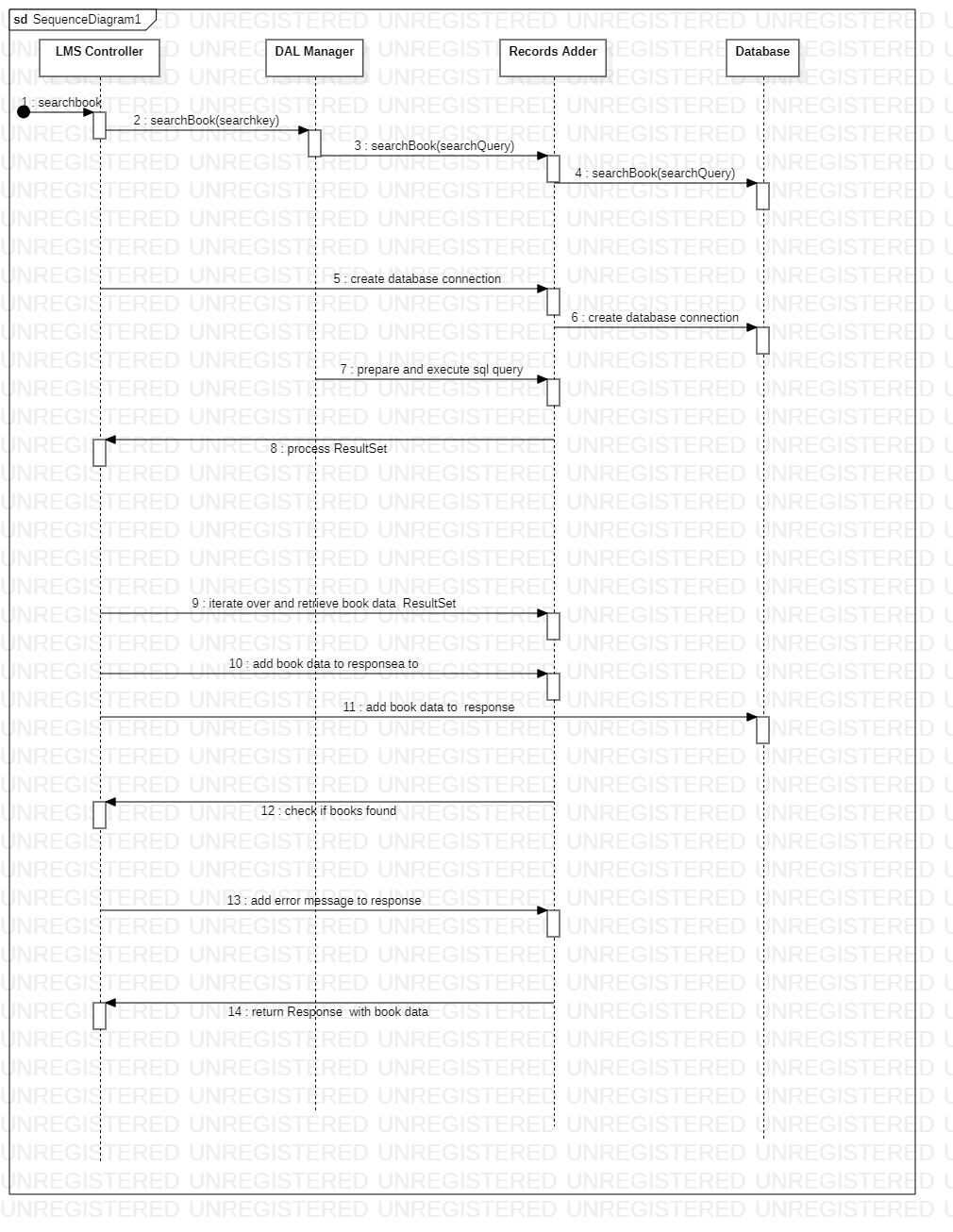


## MAHNOOR QAZI

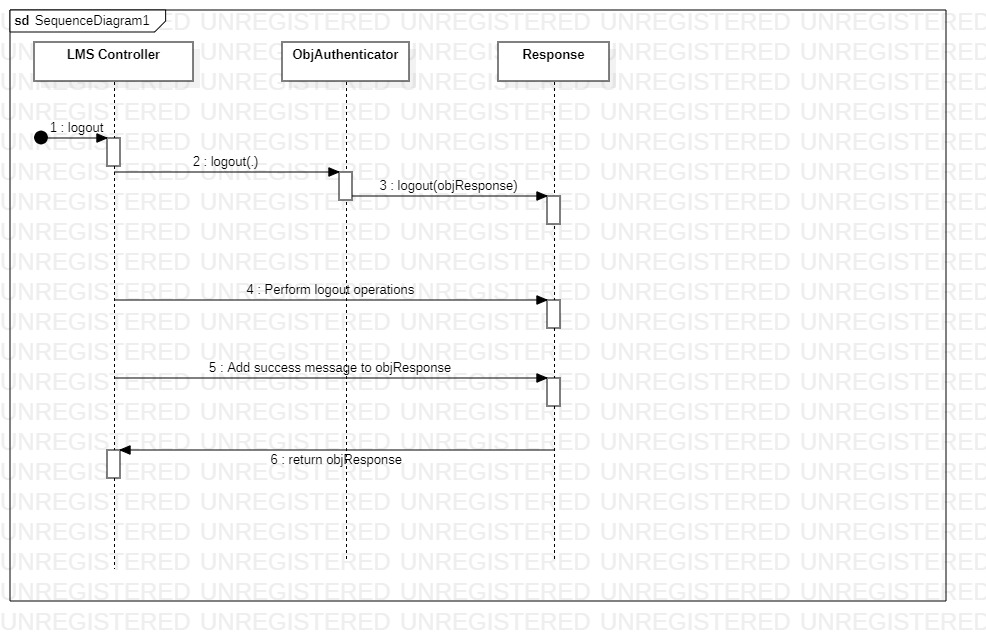
### **LOGIN**:



### **SEARCH BOOK:**



### **LOGOUT:**



### **RETURN BOOK:** **C:\Users\CUI\Downloads\returnlmsimg.jpg**

### **GENERATED FINE:** **C:\Users\CUI\Downloads\finegeneratesd.jpg**