Library Management System

Serah Michaels

Table of Contents

Library Management System Project
Serah Michaels

Problem Statement:
A library that uses an outdated manual record-keeping system. Because they are keeping track of their inventory manually, they are struggling with tracking overdue books, managing and tracking their inventory, and being able to locate available copies to their patrons in the most efficient manner. The current manual system is also prone to error and more time-consuming than a system that can provide real-time updates.

Glossary of Terms:
Library Management System (LMS): A system that automates and streamlines everyday library operations including cataloging and cardholder account management.
ISBN (International Standard Book Number): A unique identifier assigned to each book to organize them in the catalog.
Catalog: A comprehensive database of all books that are available in the library.
CH (Cardholder): A customer of the library who holds a current library card status.

Functional Requirements

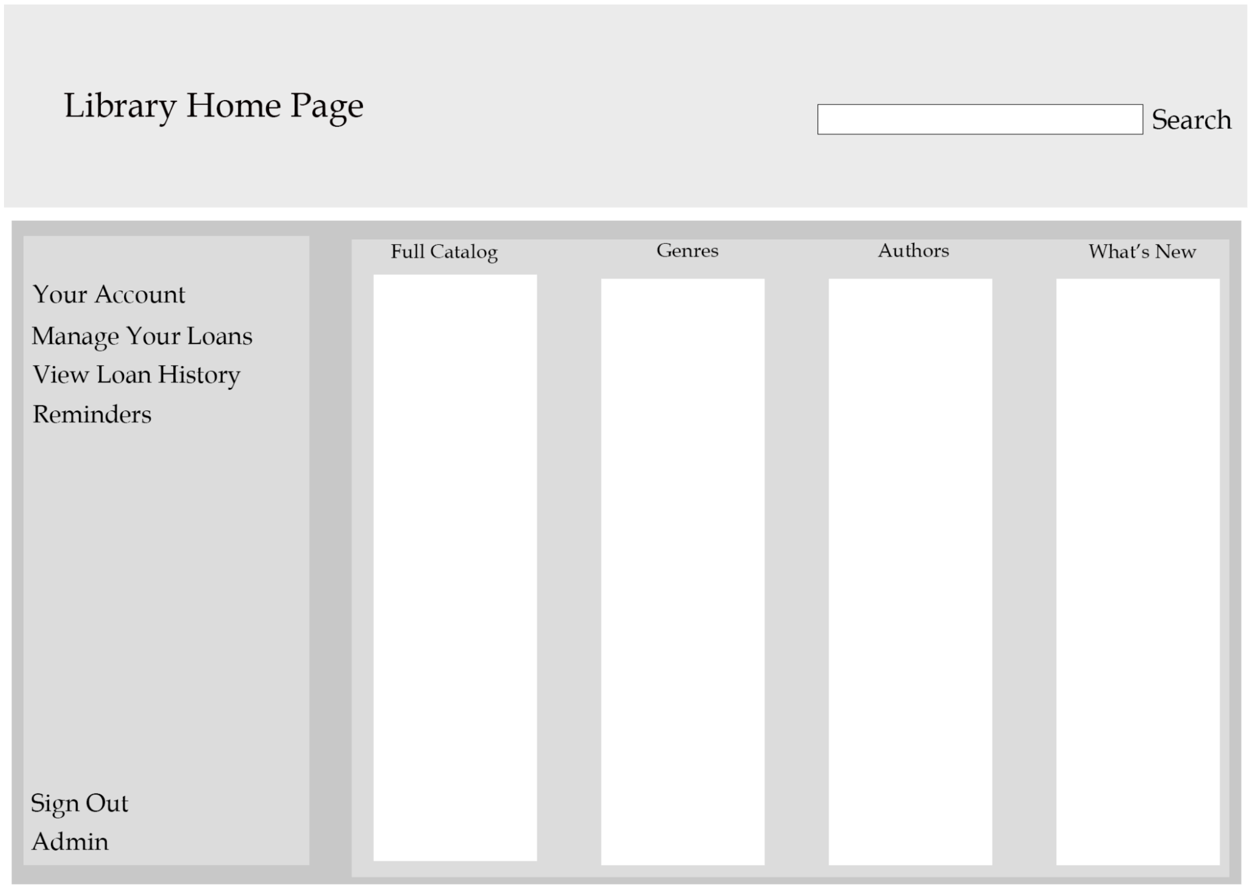
No.	**Priority**	**Description**
REQ-1	High	The catalog is able to be searched by genre, author, and title
REQ-2	Medium	The system supports automated checkout and return processing
REQ-3	High	Books can be added/removed in the library management system
REQ-4	Medium	Library staff and cardholders can track real-time book availability
REQ-5	High	Cardholders can manage their account details
REQ-6	Medium	Cardholders can view their loan history
REQ-7	High	Patrons can reserve and renew books online
REQ-8	Medium	Notifications are sent to cardholder’s accounts if a book is overdue or the return deadline is approaching
REQ-9	High	The system must generate reports on the inventory status

Nonfunctional Requirements

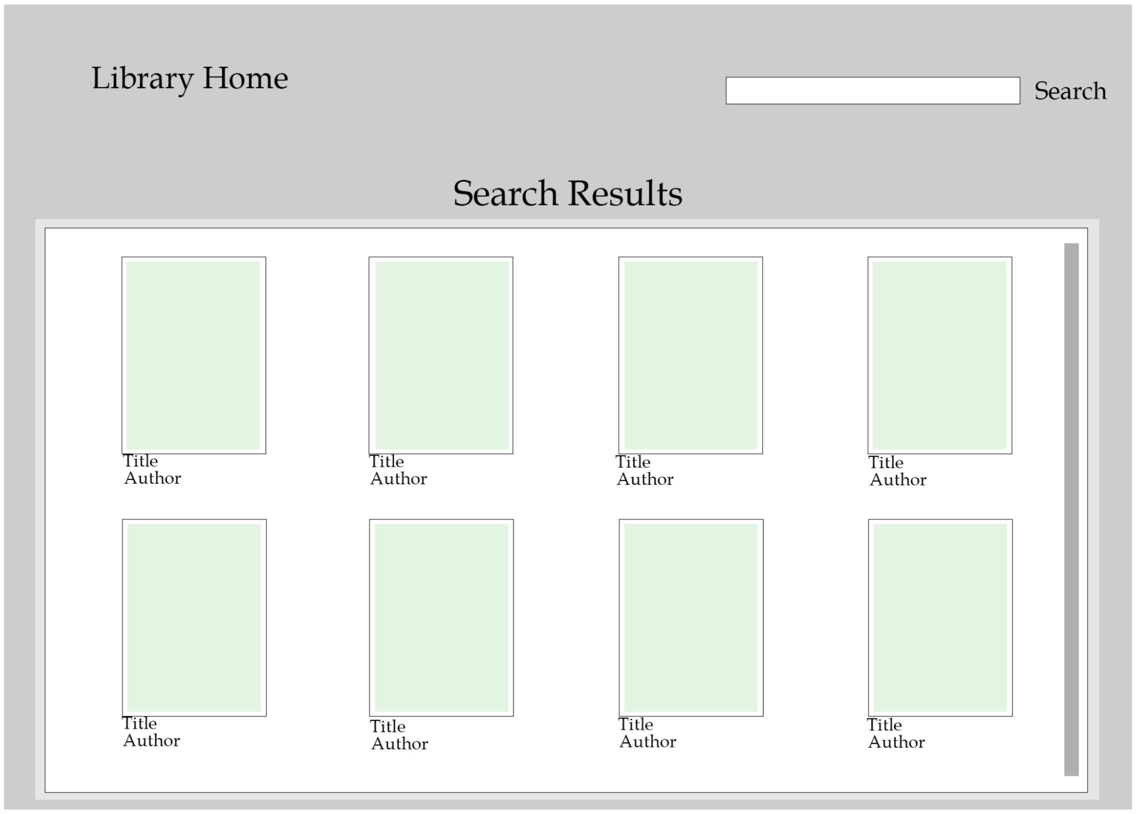
No.	Priority	Description
Functionality	High	The system can provide all library management features.
Usability	High	Cardholders must be able to reserve books online and library staff must be able to accurately track library's inventory.
Reliability	High	The system can accurately determine fines for overdue books and maintain accuracy on the library's inventory.
Performance	High	The system will be able to load queries and refresh without lags or bug interruptions.
Supportability	Medium	The system is maintainable and scalable based on a growing customer base and library catalog.

User Interface Requirements:

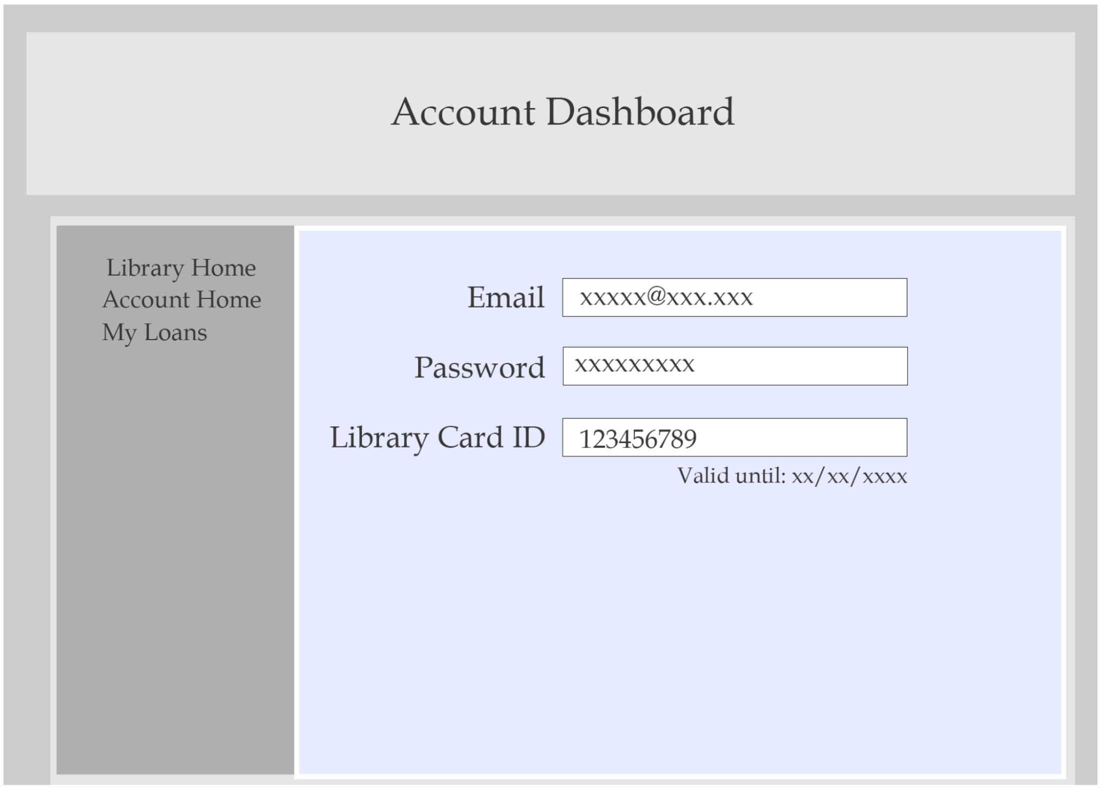
User-Friendly Navigation



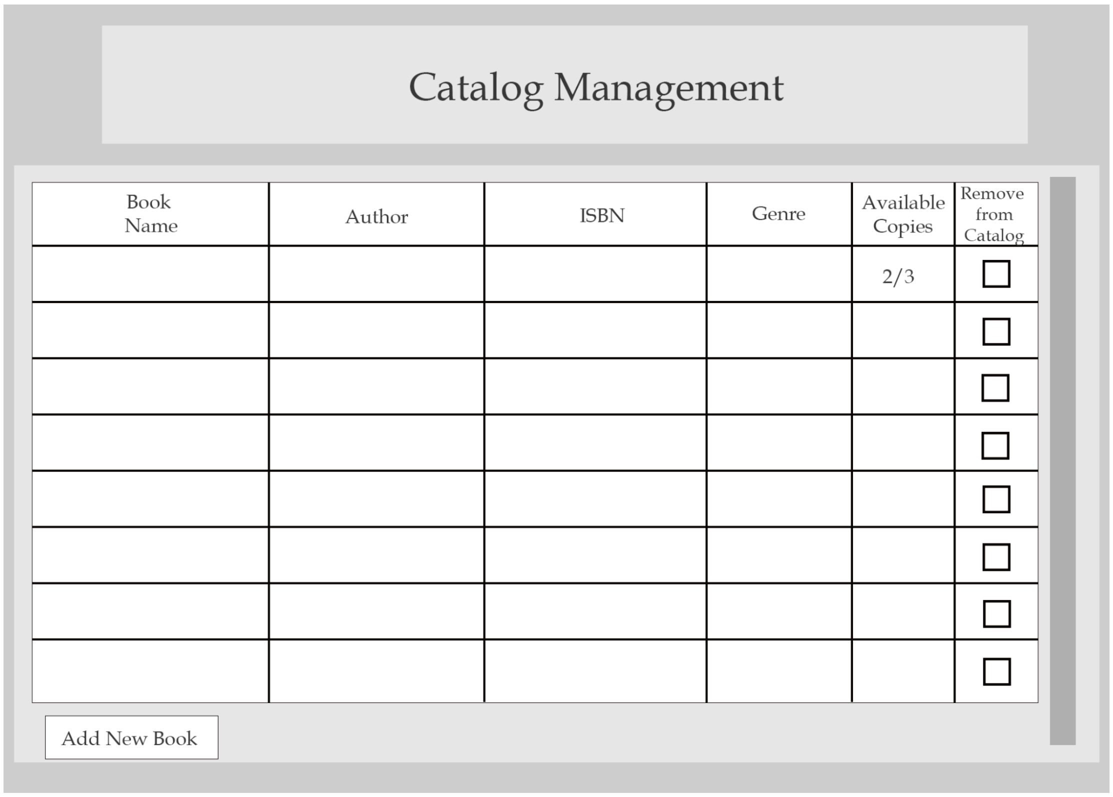
Search Functionality



Account Dashboard

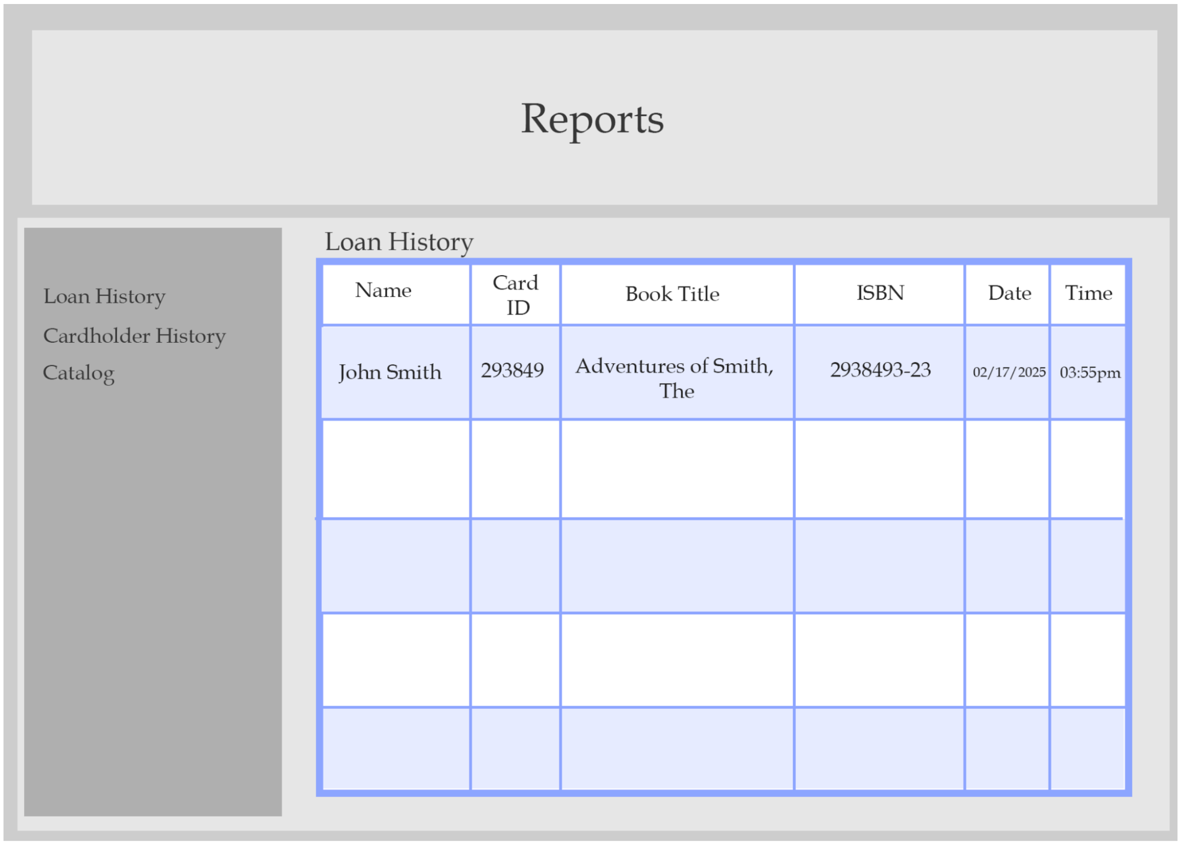


Book Management



Book Name	Author	ISBN	Genre	Available Copies	Remove from Catalog
				2/3	☐
					☐
					☐
					☐
					☐
					☐
					☐
					☐

Report Generation



Loan History

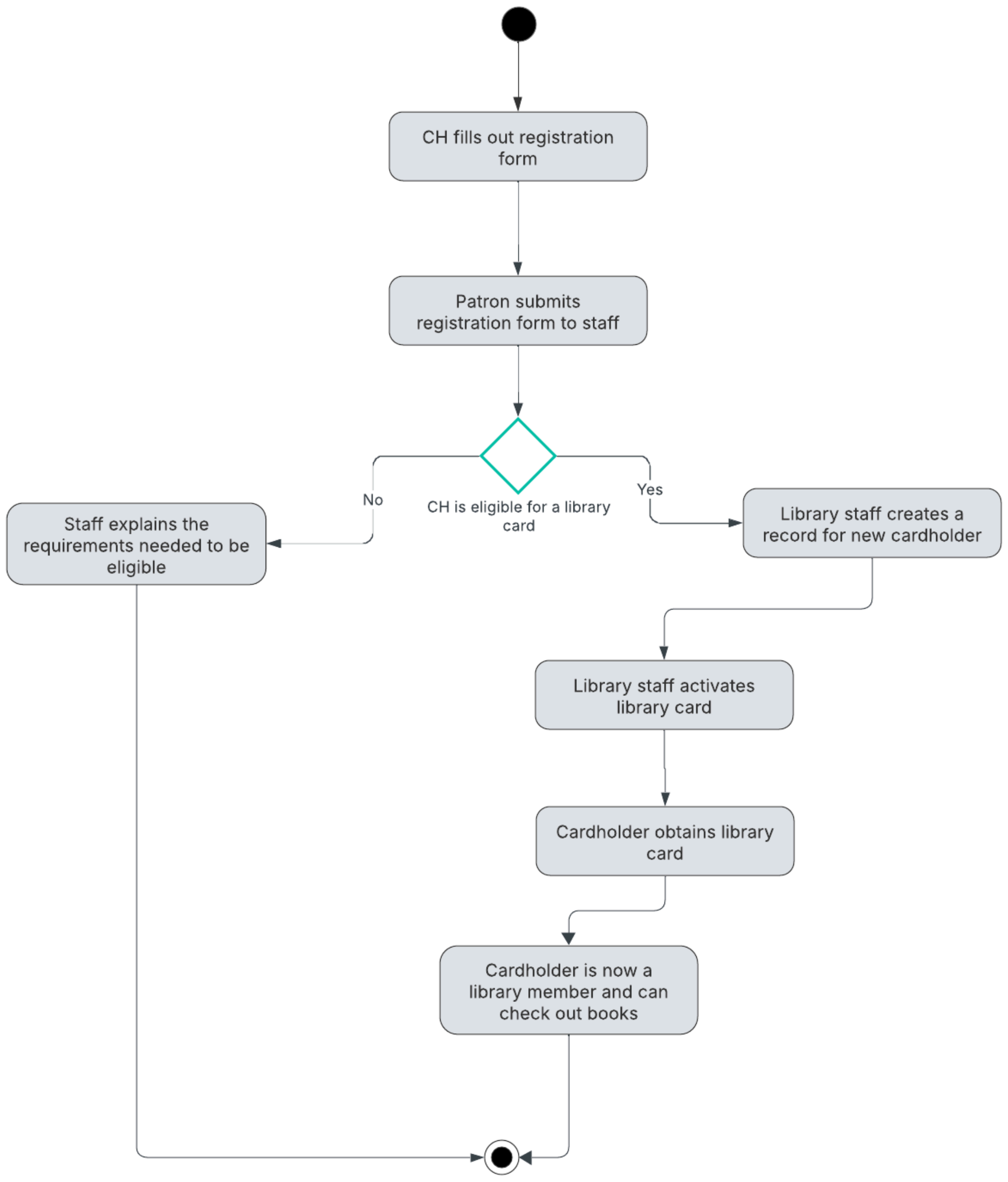
Name	Card ID	Book Title	ISBN	Date	Time
John Smith	293849	Adventures of Smith, The	2938493-23	02/17/2025	03:55pm

Plan of Work:
Weeks 1-2: setup the project, set up the development environment
I have set up my development environment so far. I will be using IntelliJ to write the code and Postgres to configure a database. I am currently trying to figure out how I want the interface to look using mockups.

Weeks 3-4: build login for users and staff, build book entry interface, build book catalog and book search features
Weeks 5-7: build online loan and return function, build user profile and loan history features
Week 8: initial testing of the system
Weeks 9-11: add notification capabilities, refine book search and management features
Weeks 12-14: testing of program, bug fixes, optimize performance
Week 15: final demo

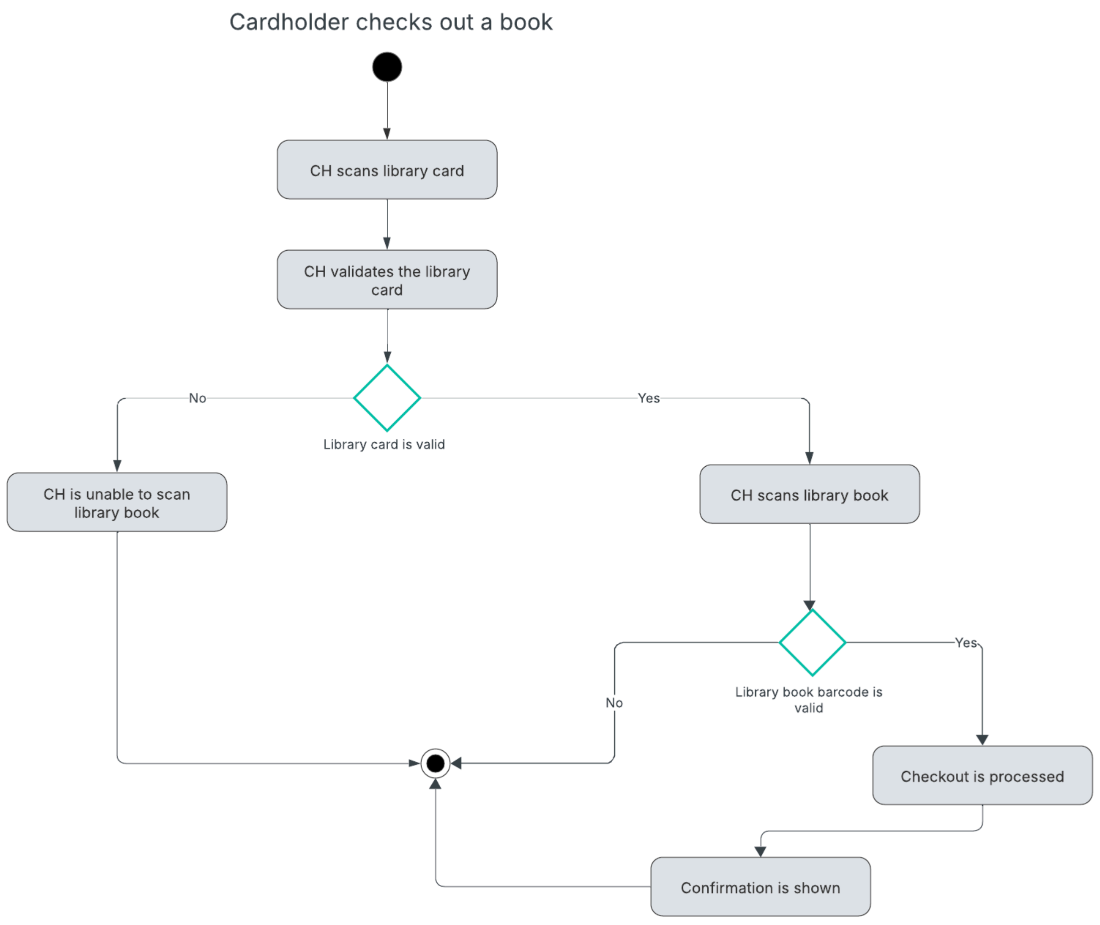
Module 5 - System Sequence Diagram and Activity Diagram
Serah Michaels

Cardholder obtains a library card

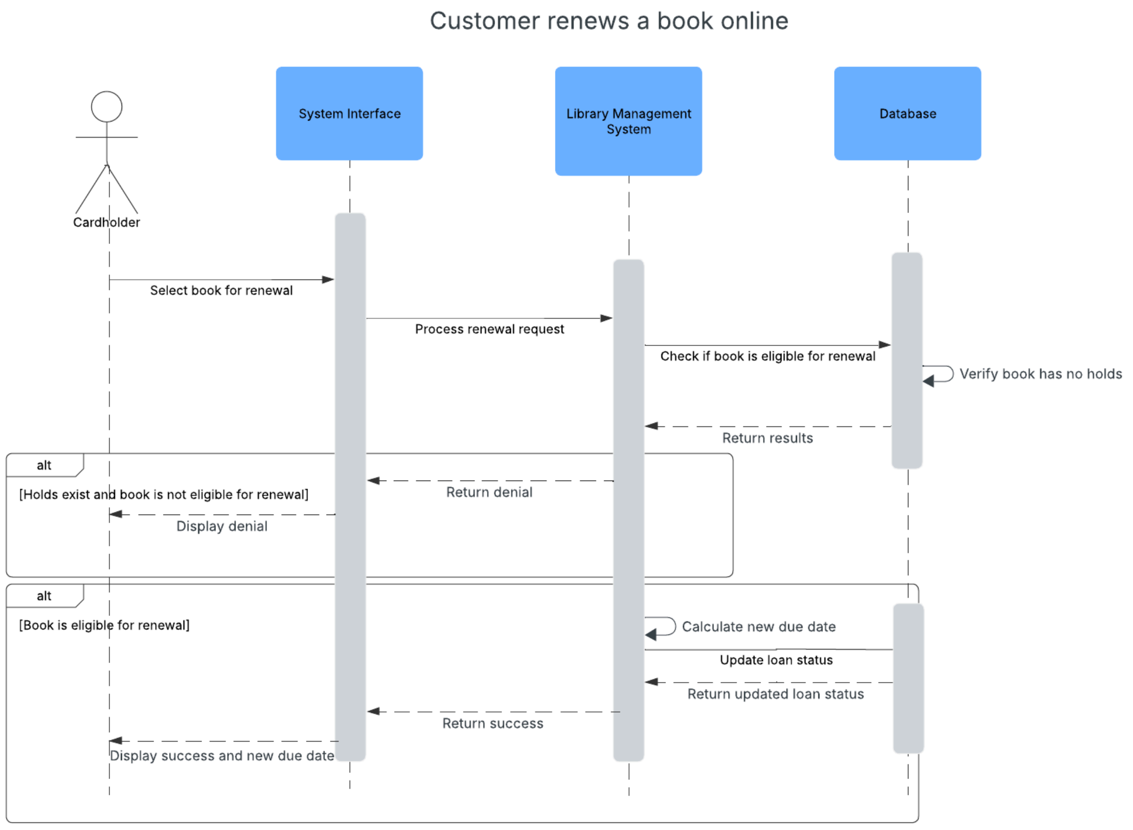

CH fills out registration form
Patron submits registration form to staff
CH is eligible for a library card
No
Yes
Staff explains the requirements needed to be eligible
Library staff creates a record for new cardholder
Library staff activates library card
Cardholder obtains library card
Cardholder is now a library member and can check out books

Actions:
The patron fills out a registration form to apply for a library card. The patron submits the form to the staff. If the patron is not eligible for a library card, the staff explains to the patron the eligibility requirements and the patron will need to come back at a later date if they can not meet the requirements. If the patron is eligible for a library card, the staff creates a record in the system for the new cardholder. A library card number is assigned and activated. The cardholder

obtains the library card, and they are now a library member and can use their card to check out books.


Cardholder checks out a book
CH scans library card
CH validates the library card
No
Yes
Library card is valid
CH is unable to scan library book
CH scans library book
Yes
No
Library book barcode is valid
Checkout is processed
Confirmation is shown

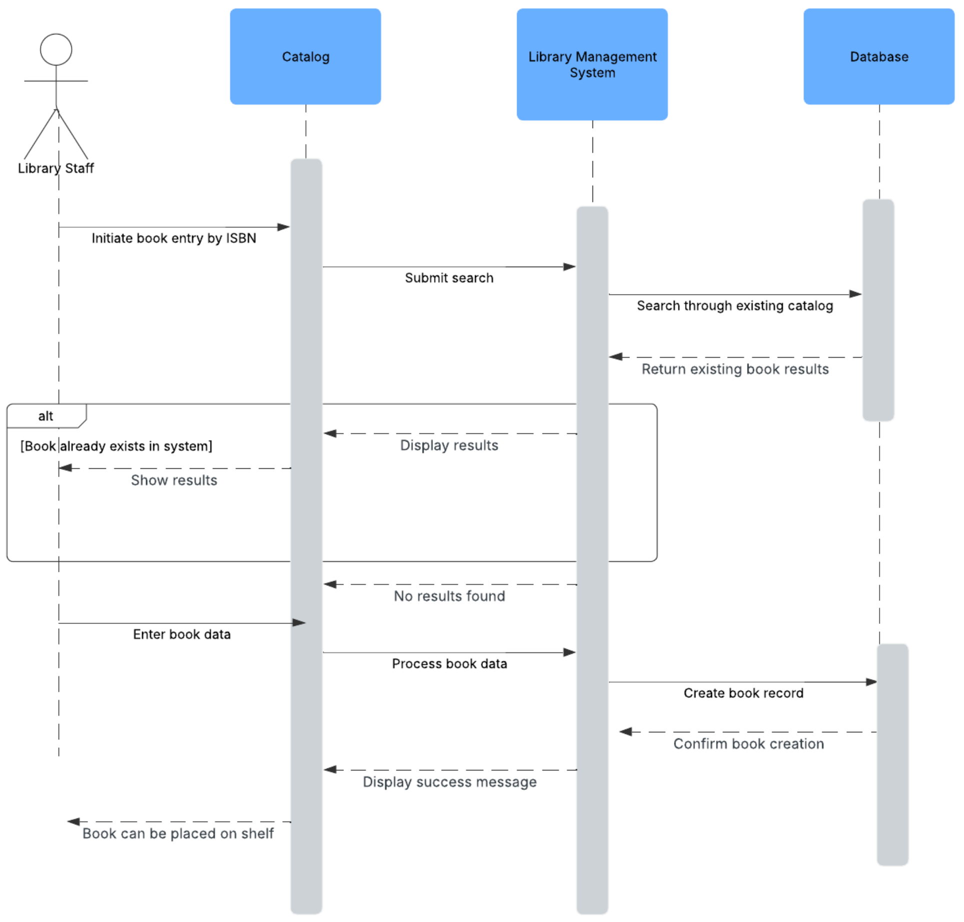
Actions:
To check out a book, the cardholder scans their library card. If their library card is not active, the cardholder is unable to continue to scan a library book. If their library card is active, the cardholder continues to scan a library book. If the book barcode is not valid, the cardholder can not check out the book. If the book barcode is valid, the checkout process is completed and confirmation is shown to the cardholder.


Customer renews a book online
System Interface
Library Management System
Database
Cardholder
Select book for renewal
Process renewal request
Check if book is eligible for renewal
Verify book has no holds
Return results
alt
[Holds exist and book is not eligible for renewal]
Return denial
Display denial
alt
[Book is eligible for renewal]
Calculate new due date
Update loan status
Return updated loan status
Return success
Display success and new due date

To renew a book online:

1. The cardholder selects the book they would like to renew.
2. The LMS processes the renewal request
3. The system searches the database to check if the book is eligible for renewal
4. The database verifies that the book has no existing holds
5. If holds exist and the book is not eligible for renewal:
    - The LMS returns the denial and displays it to the customer
6. If the book is eligible for renewal:
    - The LMS creates a new due date and updates the loan status in the database
7. The renewal is successfully placed and the success and new due date is displayed to the cardholder

Library adds a book to catalog

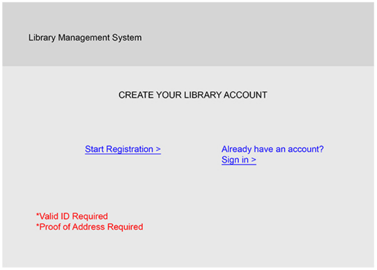


Steps for library staff to add a book to the catalog:

1. Staff inputs ISBN to begin book entry
2. The search is submitted to the LMS
3. The system searches the database for an existing record
4. If the book already exists in the system:
    - The result is displayed to the library staff and no record is input
5. If the book is not in the system:
    - The library staff continues to input book data
    - The book data is processed by the LMS
    - The book is added to the database
    - The confirmation for the book creation is sent to library staff
6. The book is ready to be placed on the shelf and checked out.

Use Case #1: A patron creates a new account

Screen 1: Account Creation



Library Management System

CREATE YOUR LIBRARY ACCOUNT

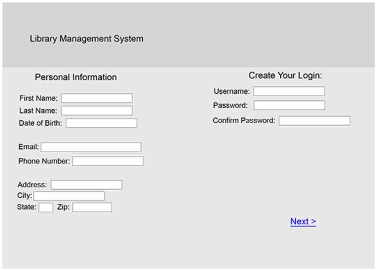
Start Registration >

Already have an account?
Sign in >

*Valid ID Required
*Proof of Address Required

>

Screen 2: Personal Info



Library Management System

Personal Information

First Name:
Last Name:
Date of Birth:

Email:
Phone Number:

Address:
City:
State: Zip:

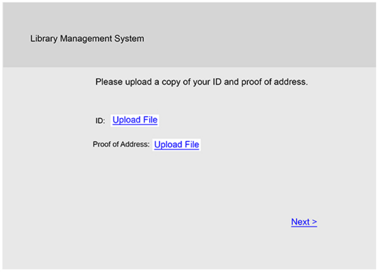
Create Your Login:

Username:
Password:
Confirm Password:

Next >

>

Screen 3: Upload Documents



Library Management System

Please upload a copy of your ID and proof of address.

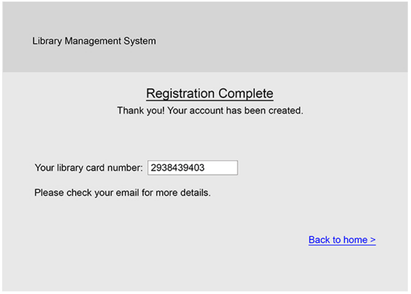
ID: Upload File

Proof of Address: Upload File

Next >

>

Screen 4: Confirmation



Library Management System

Registration Complete

Thank you! Your account has been created.

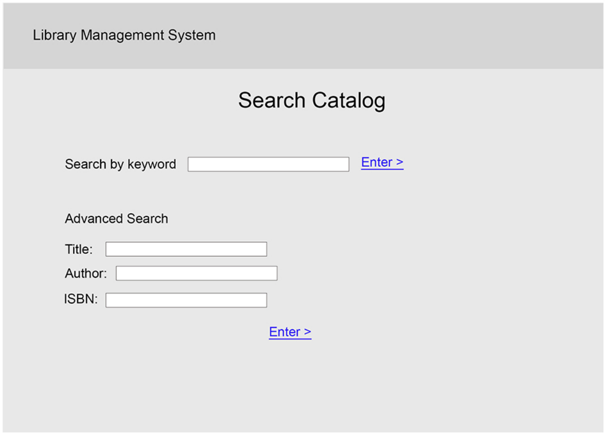
Your library card number: 2938439403

Please check your email for more details.

Back to home >

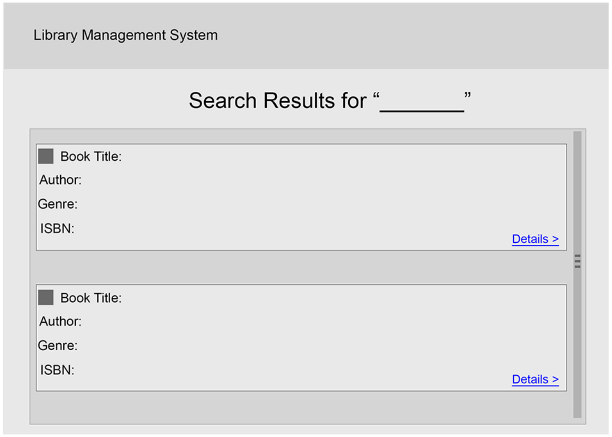
Use Case #2: Search for a title

Screen 1: Search Interface



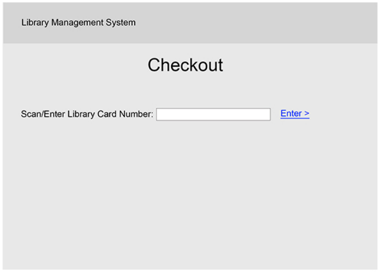
>

Screen 2: Search Results



Use Case #3: Checkout Process

Screen 1: Enter Library Card



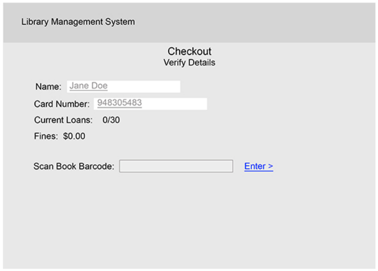
Library Management System

Checkout

Scan/Enter Library Card Number: ______ Enter >

>

Screen 2: Verify Information



Library Management System

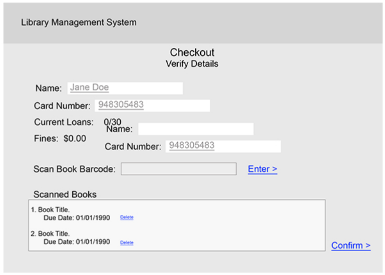
Checkout
Verify Details

Name: Jane Doe
Card Number: 948305483
Current Loans: 0/30
Fines: $0.00

Scan Book Barcode: ______ Enter >

>

Screen 3: Books Scanned



Library Management System

Checkout
Verify Details

Name: Jane Doe
Card Number: 948305483
Current Loans: 0/30
Name:
Fines: $0.00
Card Number: 948305483

Scan Book Barcode: ______ Enter >

Scanned Books

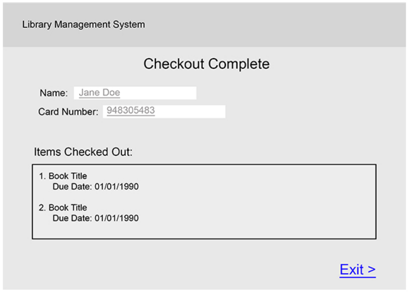
1. Book Title.
   Due Date: 01/01/1990 Delete

2. Book Title.
   Due Date: 01/01/1990 Delete

Confirm >

>

Screen 4: Checkout Complete



Library Management System

Checkout Complete

Name: Jane Doe
Card Number: 948305483

Items Checked Out:

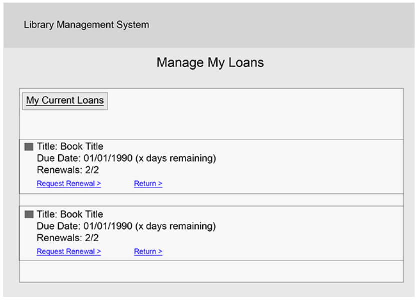
1. Book Title
   Due Date: 01/01/1990

2. Book Title
   Due Date: 01/01/1990

Exit >

Use Case #4: Renewing a Title

Screen 1: Loan Home


Library Management System

Manage My Loans

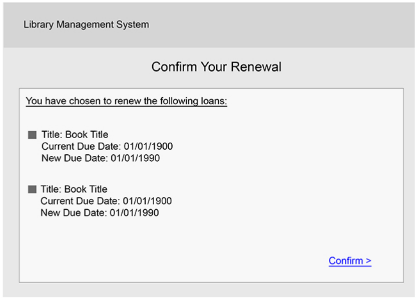
My Current Loans

■ Title: Book Title
Due Date: 01/01/1990 (x days remaining)
Renewals: 2/2

■ Title: Book Title
Due Date: 01/01/1990 (x days remaining)
Renewals: 2/2

>

Screen 2: Renewal Confirmation


Library Management System

Confirm Your Renewal

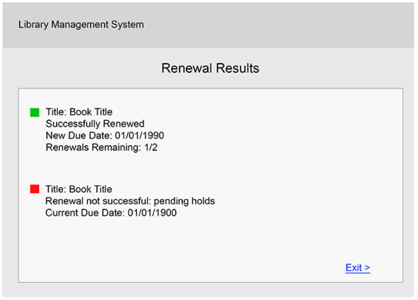
You have chosen to renew the following loans:

■ Title: Book Title
Current Due Date: 01/01/1900
New Due Date: 01/01/1990

■ Title: Book Title
Current Due Date: 01/01/1900
New Due Date: 01/01/1990

>

Screen 3: Renewal Results


Library Management System

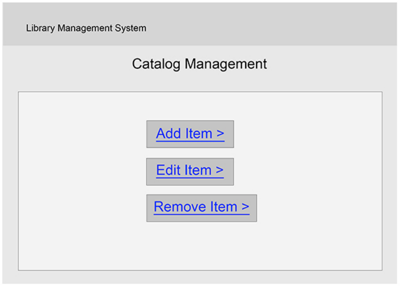
Renewal Results

🟩 Title: Book Title
Successfully Renewed
New Due Date: 01/01/1990
Renewals Remaining: 1/2

🟥 Title: Book Title
Renewal not successful: pending holds
Current Due Date: 01/01/1900

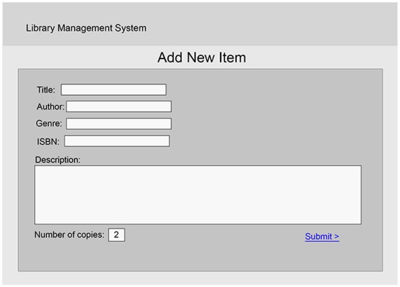
Use Case #5: Adding a Title to Catalog

Screen 1: Catalog Management



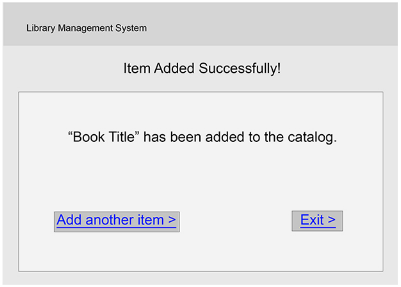
>

Screen 2: Add New Item



>

Screen 3: Confirmation


Library Management System
Item Added Successfully!
"Book Title" has been added to the catalog.
Add another item >
Exit >

User Effor Estimation

Use Case #1: A Patron Creates a New Account

The patron will need to click start registration. They will be taken to the account creation page and will need to enter their personal information (first name, last name, date of birth, email, phone number, address, city, state, zip). Then, they will need to create their username and password, and confirm their password. Then they will be taken to the ID verification page where they will need to click to browse files, choose file and upload (3 click). They will need to do this for their proof of address (3 click). Then, they click next and are taken to the registration complete screen.

Total for Use Case #1:

- Mouse clicks: 10
- Keystrokes: 85-100

Use Case #2: Search for a Title

The patron will click and enter keywords. The cardholder can also search by title, author, or ISBN (1-3 clicks). Then, the patron will select submit (1 click). The cardholder is taken to the search results page and any titles matching the keywords will return in the search results. The cardholder can click on details for each result (1 click).

Total for Use Case #2:

- Mouse clicks: 5
- Keystrokes: 15-30

Use Case #3: Checkout Process

The patron will need to add or scan their library card and click the enter button (1 click). They will be taken to the next screen which displays their information including their name, card number, their current loans, and any fines on their. The cardholder will then need to scan the book barcode and click enter (1 click). The screen will refresh with the books that they have scanned. Then, they will click confirm (1 click) and they will be taken to the checkout confirmation screen that displays their information and how many books they have checked out. Then, the patron can click exit.

Total for Use Case #3:

- Mouse clicks: 6

- Keystrokes: 45-70

Use Case #4: Renewing a Title

The patron will access their loan management screen. Here, their current loans will be displayed. The patron can click on each title and then request renewal or return on their loans. If they click request renewal, they will be taken to the renewal confirmation screen. The screen will display their selections. The patron will select confirm (1 click). Then, they are taken to the renewal confirmation screen that displays the loans they have successfully renewed or loans that have not been renewed. If the loan is unable to be renewed, a reason will be displayed. The patron can now exit (1 click).

Total for Use Case #4:

- Mouse clicks: 5
- Keystrokes: 0

Use Case #5: Adding a Title to Catalog

From the catalog management screen, staff can select to add an item to the catalog, edit an existing item, or remove an item. If the staff selects add item (1 click), they are taken to the next screen where they can enter the details. The staff will enter title, author, genre, ISBN, description if applicable, and the number of copies (6 click). Then, the staff will select submit (1 click). The next screen is the confirmation screen that the title has been added to the catalog successfully. The staff can select add another item or exit. If add another item is selected, they are returned to the second screen.

Total for Use Case #5:

- Mouse clicks: 5
- Keystrokes: 110-200

Traceability Matrix

REQ	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9
REQ 1	5		X	X			X			
REQ 2	4				X	X				
REQ 3	5		X		X					
REQ 4	5									X
REQ 5	3							X		
REQ 6	4								X	
REQ 7	4			X						
REQ 8	3									
REQ 9	4	X								
MAX PW		5	4	5	5	3	4	4	3	4
TOTAL PW		5	8	10	10	3	4	4	3	4

System Requirements

No	Priority Weight	Description
REQ1	5	Manage the library’s catalog and inventory
REQ2	4	Search through the library’s inventory by author, title, or genre
REQ3	5	Ability to reserve items
REQ4	5	Assign a library card number to a member
REQ5	3	Track overdue items
REQ6	4	Generate reports for current inventory
REQ7	4	Check availability of items
REQ8	3	Renew items online
REQ9	4	Ability to create online account

Use Cases

No	Description
UC1	User logs in to the system
UC2	User reserves an item
UC3	Available items are viewed in the LMS
UC4	A user renews an item online
UC5	A user searches for an item
UC6	Staff adds or removes an item from the inventory
UC7	Staff generates a report for overdue items
UC8	Staff generates a report for current inventory
UC9	A user obtains a library card

System Architecture and System Design
Library Management System - Serah Michaels

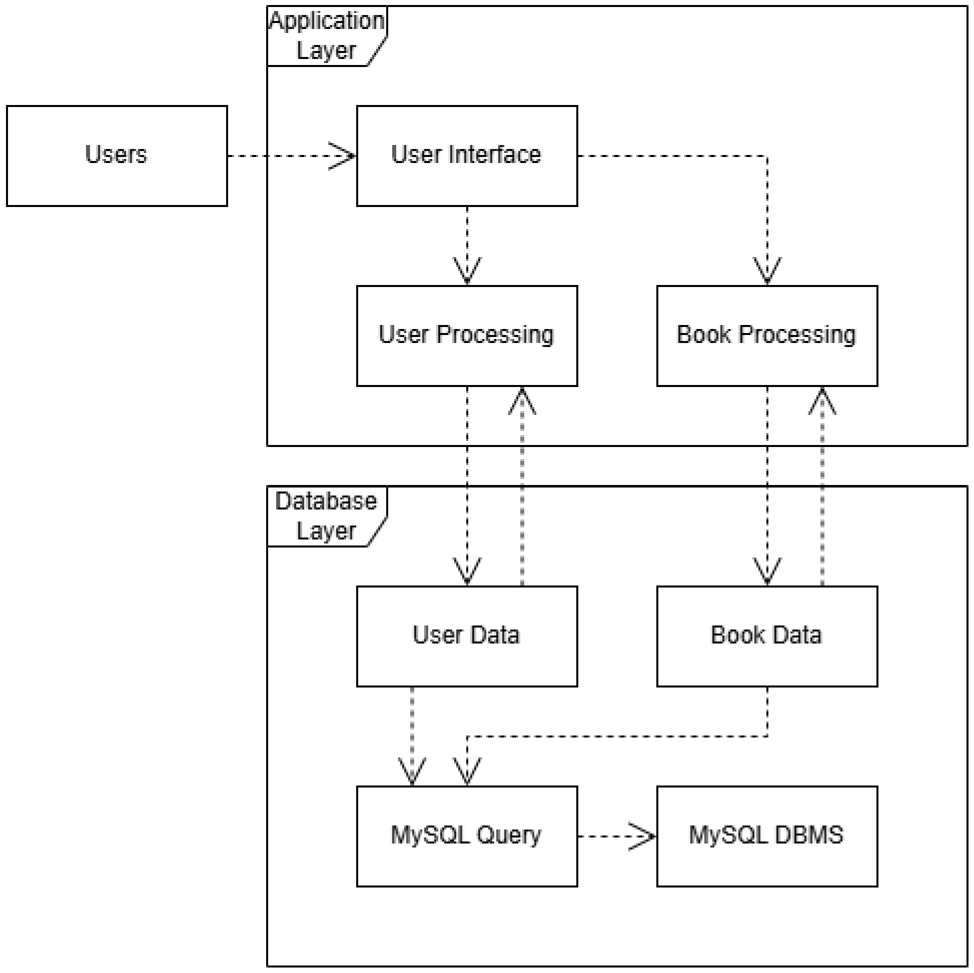
Architectural Styles

The Library Management System is a client-server architectural style. The client is the Java application running on the user's computer that sends requests to the MySQL database server and waits for the server to respond with data. The LibrarySystem.java serves as the entry point that initializes the client application by calling LibraryUI.createUI(). The LibraryUI.java creates all the graphical interface elements that users interact with on the client side, containing buttons for functions like "Add Book" and "Reserve Book" that trigger actions. When a user clicks these buttons, LibraryOperations.java handles these client-side requests by preparing the appropriate database operations. For example, when adding a book, LibraryOperations.addBook() collects input from the user through JOptionPane dialogs, then forms a SQL INSERT query to send to the server. The LibraryDatabase.java is the file that directly interfaces with the server side, establishing the JDBC connection to MySQL. It provides the getConnection() method that other client components use to obtain a connection to the database server. The server is the MySQL database that receives SQL queries from the client, processes them against the stored data in the 'books' and 'users' tables, and sends results back to the client. All data persistence happens on the server side, while all user interaction and application logic occur on the client side.

Identifying Subsystems

The Library Management System has four main files that work together to make the library system function. The main starting point of the system is the LibrarySystem.java file which allows for the initialization of the program. The User Interface package has the LibraryUI.java file which shows what the user sees on the screen. This interface has interactive buttons like "Register," "Login," "Add Book," and other options that the user can click on. When they click these buttons, the program communicates with the LibraryOperations.java file which handles processes such as adding new books, registering users, and reserving books. When the Operations file needs to save or retrieve information, it connects to the LibraryDatabase.java file. This package is responsible for connecting to the MySQL database where all the book information and user accounts are stored. It sends SQL queries to save new books or get information about existing ones.

UML Package Diagram:



Mapping Subsystems to Hardware

This system can run entirely on a single computer. Optionally, the MySQL database server can be hosted on a separate machine, but they can also run together on a single machine.

Persistent Data Storage

Yes, the system requires persistent data storage to maintain information about books and users. MySQL is being used as a relational database to maintain this data.

Network Protocol

The system uses JDBC protocol to communicate with the MySQL database server.

Global Control Flow

Execution Orders

The system is event-driven. The application initializes the UI and then waits for user actions. Users can generate actions in any order by clicking different buttons in the interface, and each action triggers a corresponding event handler that executes the requested operation.

Time Dependency

The current system is of event-response type with no timers implemented in the current version. The system responds to user inputs without time constraints. However, adding the future feature of the automatic notification of overdue books would require timer functionality.

Hardware Requirements

- **Color Display: Minimum resolution: 640 x 480 pixels**
- **Computer: Desktop/PC (no mobile capability at this time)**
- **Memory: 1 gb RAM**
- **Hard Drive: Minimum 1 gb hard drive disk space**

System Design Specification

Part A: System Overview

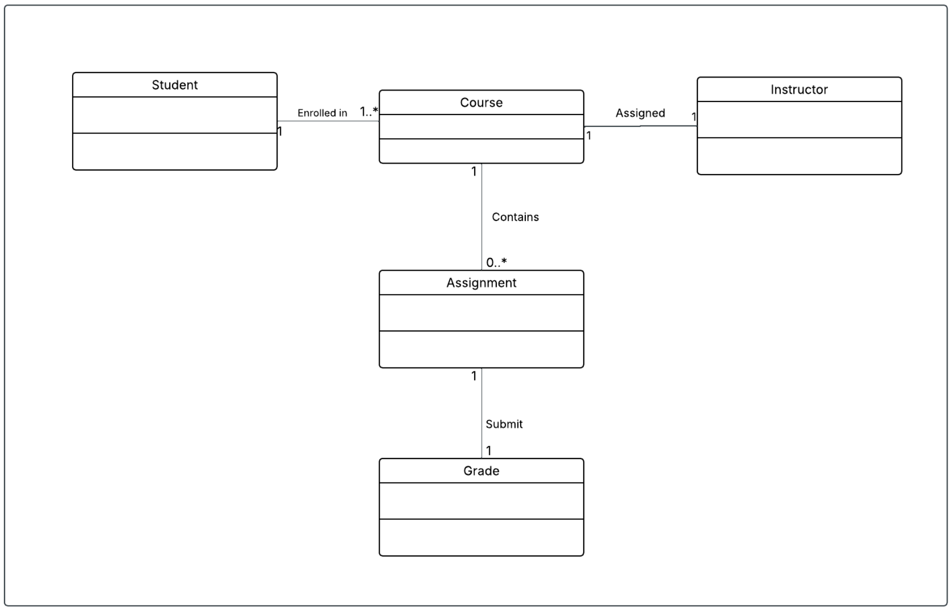
System Description: The Online Course Management System is a system that is intended for use by universities to manage operations for instructors and students. The system allows for course creation, student enrollment, assignment management, grading, and communication between students and instructors.

High-Level Goals and Objectives:

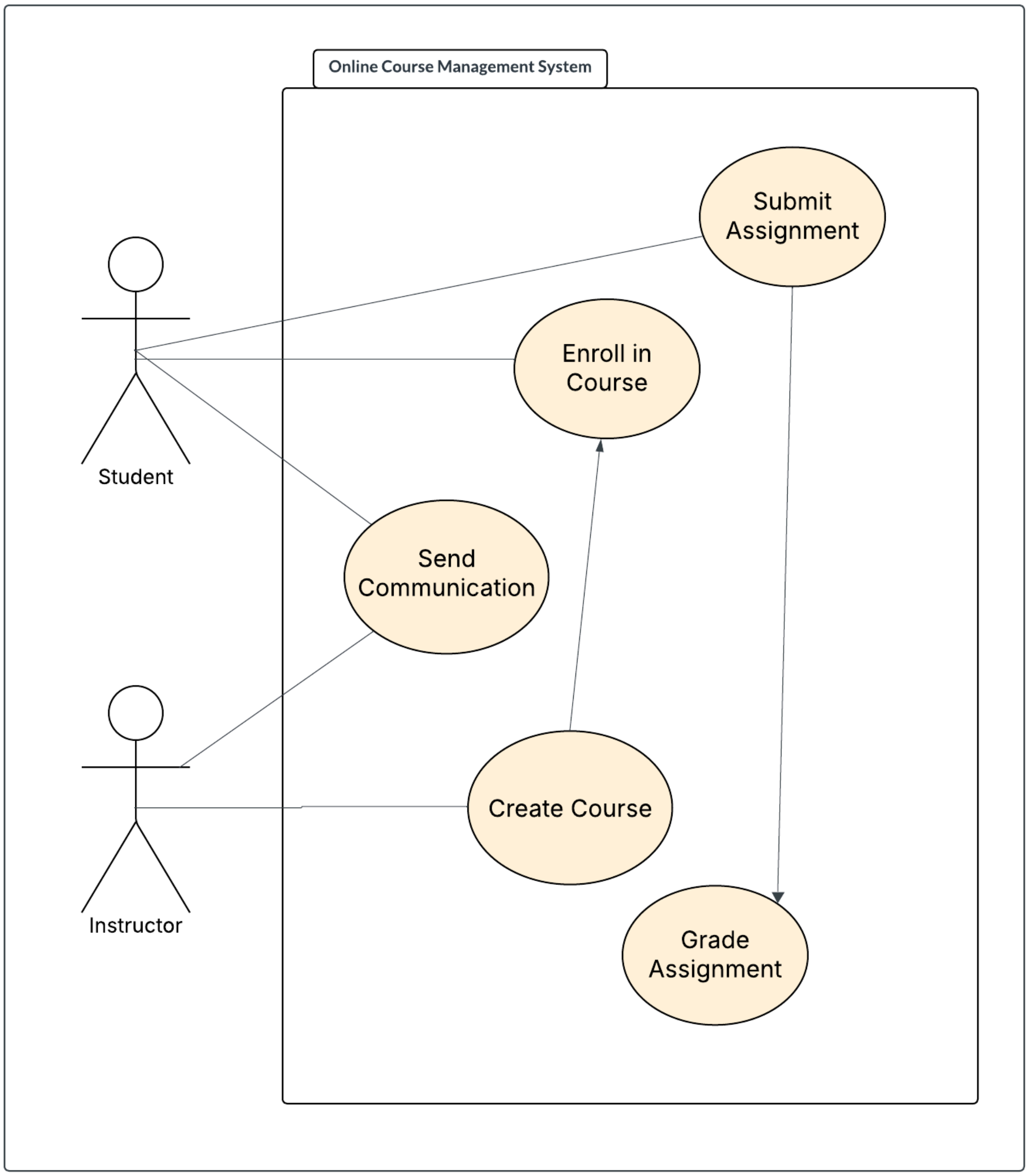
- Allow staff to create, update, and manage courses and assignments.
- Allow students to enroll in courses, submit assignments, and view grades.
- Provide communication tools (announcements, messages) between students and instructors.
- Offer secure, role-based access control for all users.
- Ensure system scalability and integration with existing university databases.

Part B: Design Integration

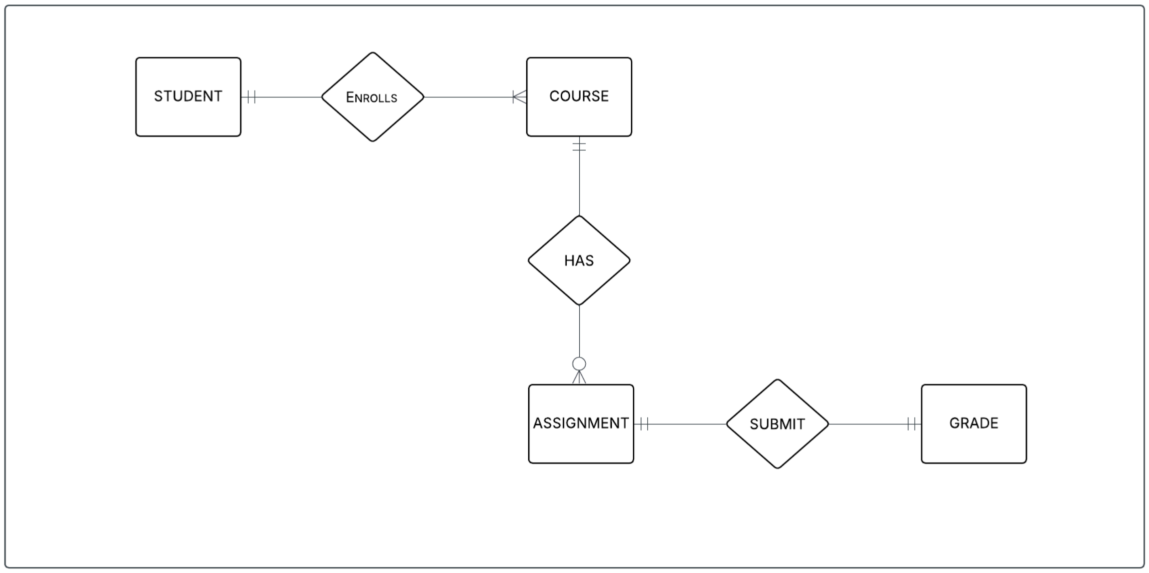
Class Diagram



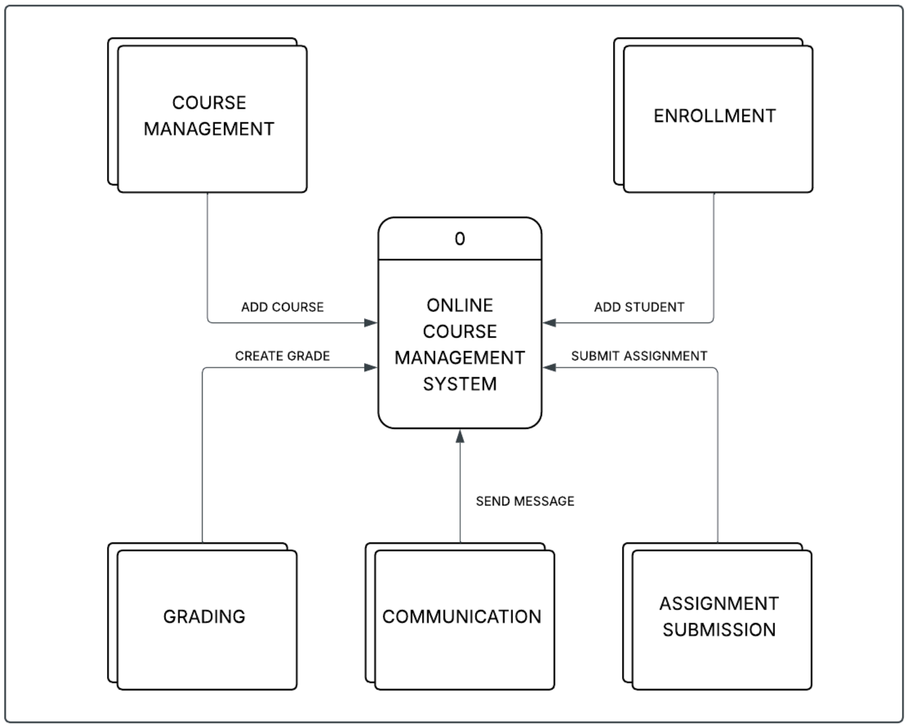
Use Case Diagram



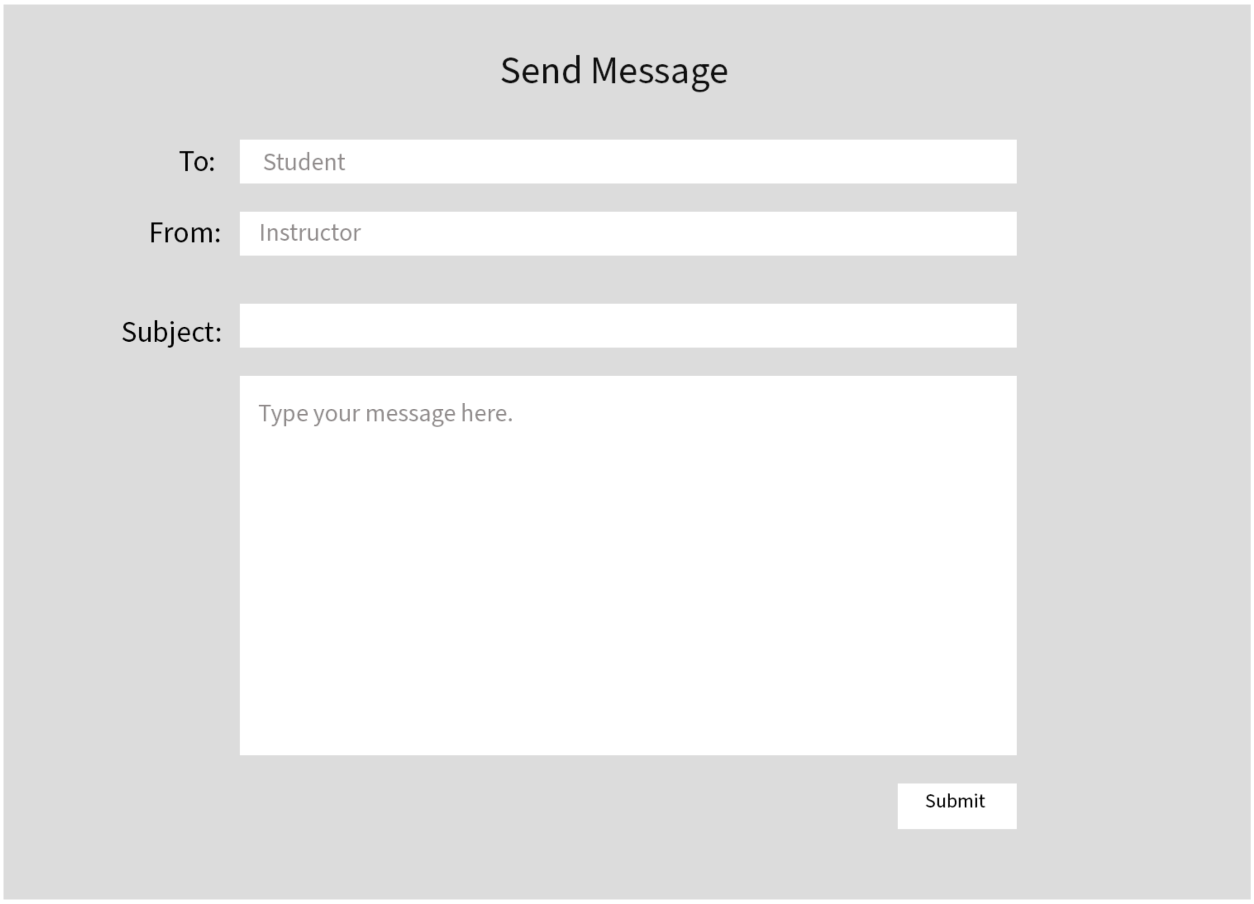
Entity-Relationship Diagram

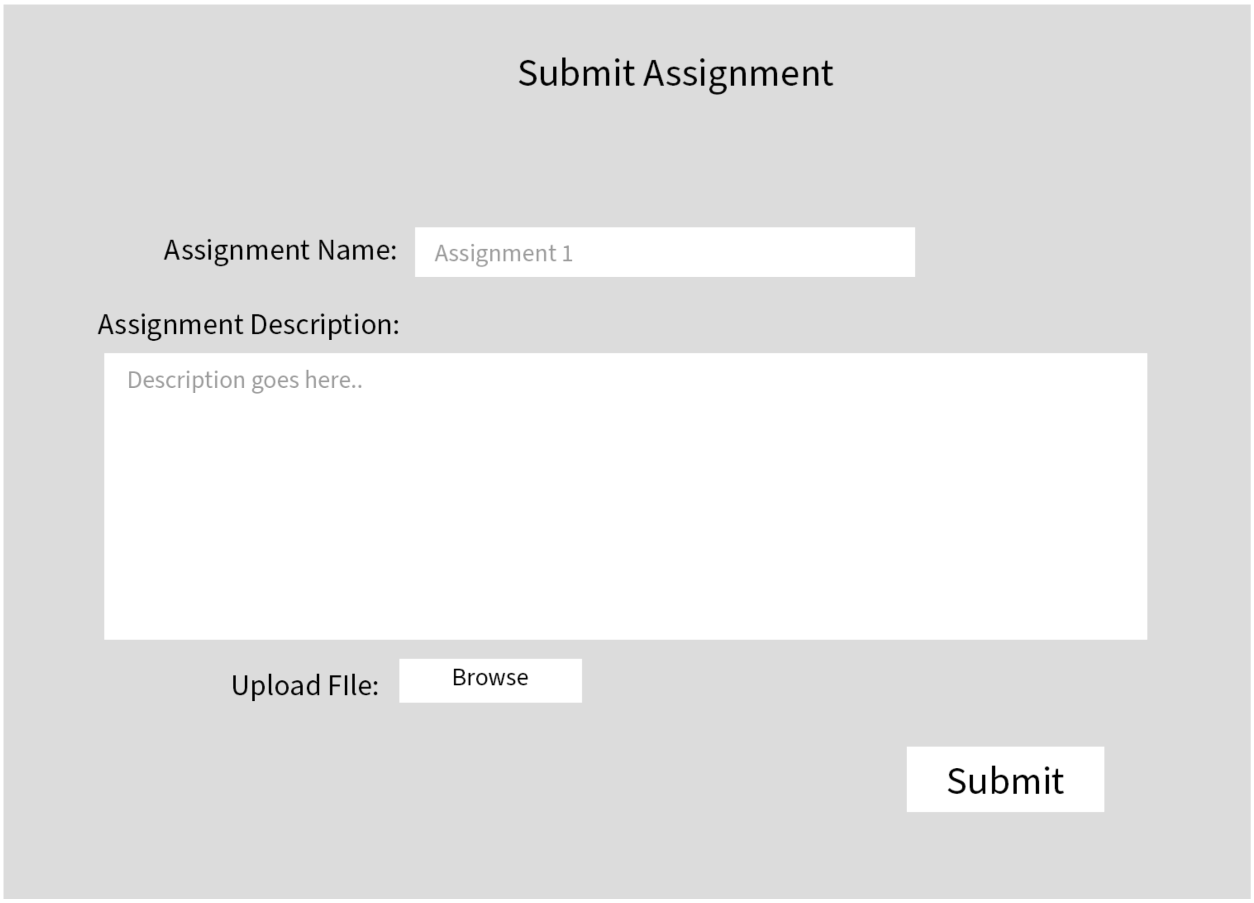


Context Diagram



Part C: Architecture and UI




Submit Assignment

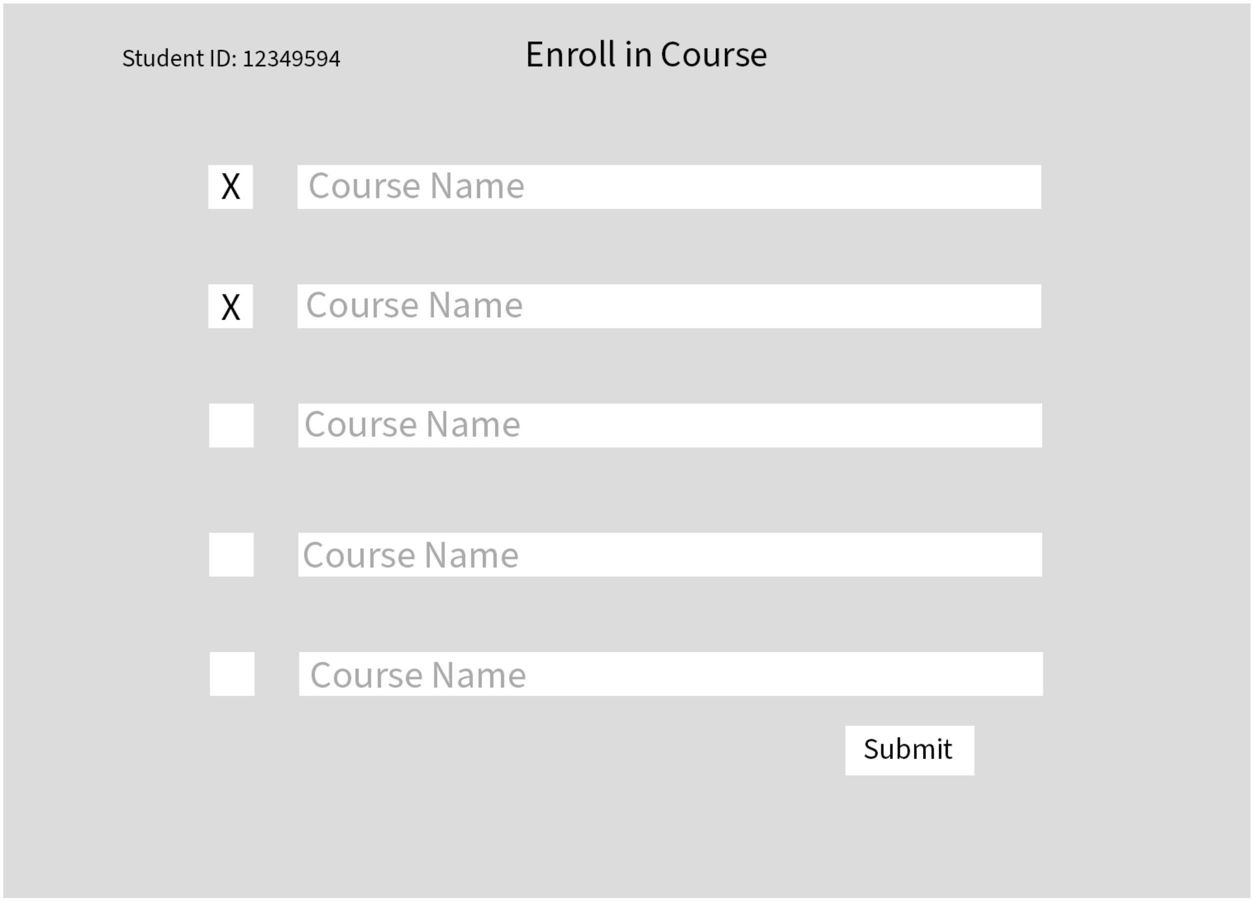
Assignment Name: Assignment 1

Assignment Description:

Description goes here..

Upload FIle: Browse

Submit



- Navigation flow and usability considerations
  - Students and instructors need to both be able to easily navigate through the interface.
  - Communications between students and instructors should be clearly labelled.

Part D: Security and Risk Considerations

Security Features:

- Only instructors have the authority to assign grades to assignments or add new courses.
- Password authentication - all users will need to set up a two-factor authentication method in order to log into their accounts.

Potential Design-Level Risks & Mitigations:

- Ensure data breaches are prevented with encryption.
- Set up backup servers to run during updates to avoid service disruptions.

Part E: Design Patterns and Best Practices

This system uses the model-view-controller design pattern. The model layer includes the data objects such as course information, assignments, and grade information. The view layer includes the user interface that students and instructors can access to view communications, grade information, and assignment details. The controller layer includes the handling of processes such as course creation, authenticating users, and the processing of assignment and grading submission.

Best Practices:

- Two-factor authentication and security measures in place to prevent data breach
- Consistent naming conventions and documentation in code
- Comprehensive testing done before deployment

Software Testing

- Unit testing will be done by testing each function individually.
- Integration testing will be done by ensuring that each module interacts with each other correctly. For example, LibraryUI should correctly interact with LibraryOperations.
- User Interface testing will be done to ensure that the correct functions are called when certain buttons are pressed, and alert dialogs pop up at appropriate times.

Error Handling

- SQLException is used.
- Try-catch blocks are used.
- Users must have valid library ID and logged in in order to reserve a book.
- Error alert dialogs are displayed with clear explanations. For example, “Book ID not found” and “You are not an admin”.

Collaboration & Code Integration

- Git is used to track and document changes to code.
- Detailed comments in the code explain what the intention is for each method.

Debugging & Logging

- Visual Studio Code was used for debugging.
- Implement logging for critical operations like user login and book reservations.
- Use log files to track issues and monitor system health.

Performance Optimization

- UI is clearly laid out and simply styled to ensure ease of access for users and staff.

Post-Delivery Maintenance Plan

- Maintain a bug tracking system to monitor and prioritize any issues.
- Set a schedule with staff for regular updates.
- Schedule meetings with library staff for any bug identification or potential improvements.
- Plan on adding new features, such as more refined search options or more intensive user profile management.
- Regularly verify the security of the database and ensure there are no breaches.
- Collect feedback from library users for future improvements.

Library Management System - Proposal

Serah Michaels

Problem Statement

For this project, I will be helping a library that uses an outdated manual record-keeping system. Because they are keeping track of their inventory manually, they are struggling with tracking overdue books, managing and tracking their inventory, and being able to locate available copies to their patrons in the most efficient manner. The current manual system is also prone to error and more time-consuming than a system that can provide real-time updates.

Objectives of the System

The library management system's objectives include providing a streamlined experience to track book inventory. Errors will become minimal since staff will be able to track inventory and overdue loans in real-time. Users will be able to loan books online without needing to visit the library. Users will also be able to determine if a book is available before visiting the library.

System Requirements

- Browse the library's book catalog
- Manage the library inventory
- Search for books by title, author, genre, or keywords
- Check book availability in real-time
- Reserve books online
- View loan history
- Renew book loans
- View due dates
- Track overdue books
- Manage book reservations

Typical Customers

The typical customers include library patrons, such as library card holders, students, and researchers. Other customers include the library staff, such as the librarians and library assistants.

Project Planning

For the software, the front-end and back-end will be developed using Java, and the database will be developed using SQL. The requirements for the hardware will include computers. The network will require a high-speed internet connection and SSL encryption to protect the user's data.

Development Approach

For this project, I will use the programming language Java and MySQL for the database. Java is flexible and versatile, and MySQL will be able to keep the book catalog well-organized. I will also use the Hibernate framework to interact Java with the database.

Development Plan

Weeks 1-2: setup the project, set up the development environment

Weeks 3-4: build login for users and staff, build book entry interface, build book catalog and book search features

Weeks 5-7: build online loan and return function, build user profile and loan history features

Week 8: initial testing of the system

Weeks 9-11: add notification capabilities, refine book search and management features

Weeks 12-14: testing of program, bug fixes, optimize performance

Week 15: final demo

References

https://www.geeksforgeeks.org/introduction-to-java-swing/
https://docs.oracle.com/javase/tutorial/uiswing/index.html
https://www.geeksforgeeks.org/java-joptionpane/
https://www.geeksforgeeks.org/java-database-connectivity-with-mysql/
https://docs.oracle.com/javase/8/docs/api/java/sql/PreparedStatement.html
https://www.geeksforgeeks.org/performing-database-operations-java-sql-create-insert-update-delete-select/
https://docs.oracle.com/javase/tutorial/uiswing/layout/border.html