Library Management System

System Requirements Documentation

Dalton Burge,
Monica Galvez, & William Applegate

Indiana University
INFO-C451: System Implementation
Instructor: Jafrina Jabin
4 May 2024

Table of Contents:

Customer Problem Statement

Problem Statement:

A library lets its members checkout books and return them on an agreed upon date. This process requires checkout procedures so that the library can lend books to its members while maintaining accountability of all books owned by the library. To complete this task, library staff must be able to track which members have checked out books, what those books are, and when they are due to be returned to the library. Library staff must have a clearly defined checkout procedure that identifies the member, the books, and the checkout and due dates. During peak hours of operation, the process to lookup the information required to complete a checkout needs to be quick. By ensuring a quick checkout process, fewer staff members are needed to handle more member check outs, and member wait times are reduced. Library staff will use a web application to quickly look up members and the books they are checking out. The web application will generate a default due date. To complete the checkout process, this information will be submitted in the web application and the application will be responsible for saving the check out details. These details will be accessible in the web application by the library staff and members.

Glossary of Terms:

Administrator	Library staff must each have their own individual account in the web application. An administrator can create staff accounts for library employees.
Library staff	Library staff are any library employee who can check books out to library members or access member records.
Member	A library customer who has registered an account with the library. Members may or may not be able to check out books depending on their account status with the library.
Guest	Non-member customers at the library who have not registered an account with the library. Guests cannot check out books.
Media	Library inventory that may be checked out to members. This includes books, CDs, DVDs, and Instruments.
Check-out process	The transaction in which the library permits a member to borrow a book from the library, only after both a library staff member and the member agree on what books are being borrowed and when they must be returned.
Account	The record of a library user, which could be either an account for a library member or a library employee.
Frozen/Active	Library staff may declare an account frozen which prevents the member from checking out books. Library staff can also return the account to an active status, thereby allowing the member to check out books again.
Due Date	The date a member is required to return checked out books by. Failure to return books by the due date will result in the user’s account being frozen and a fine accrued.

System Requirements:

- Functional Requirements:

Requirement No.	Priority	Description
REQ-1	High	The systems should prevent users with late or lost media from borrowing more.
REQ-2	High	Users should be able to search the libraries catalogue
REQ-3	Medium	Users should get a notification when the media is close to being due.
REQ-4	High	Online Payment can be made using a credit/debit card.
REQ-5	High	The four types of media are books, CDs, DVDs, and Instruments
REQ-6	Medium	Payment in the library can be made using either a credit/debit card or cash
REQ-7	High	A library employee can check media out to members using a user interface.
REQ-8	High	A library employee can look up library members and view their account details. Account details include member name, contact information, media the member currently has checked out, current fines, and if the account is frozen or active.
REQ-9	High	A library employee can look up media to determine if it is checked out. If the media is checked out, the library employee can see what member checked out the media.

- Nonfunctional Requirements:

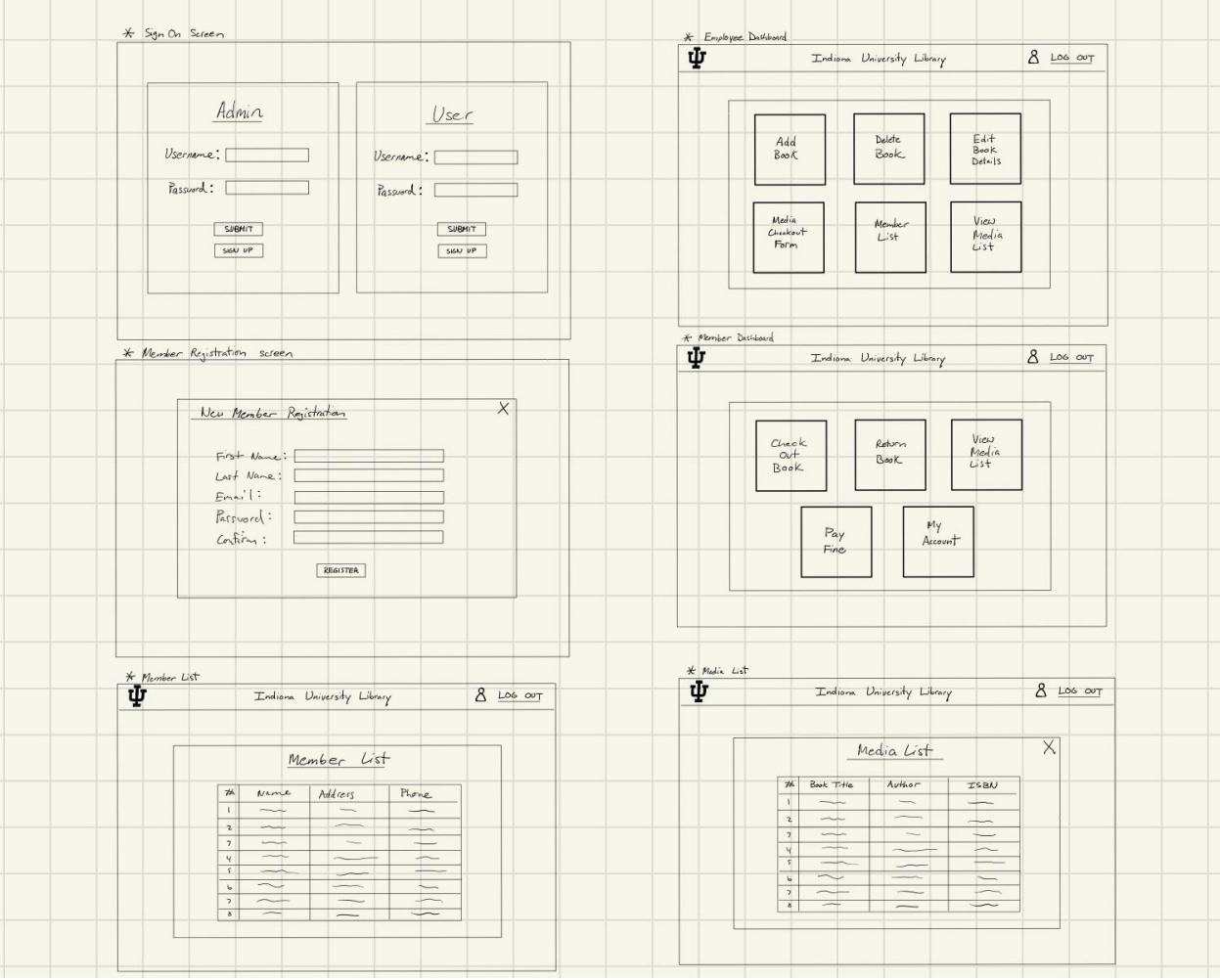
Functionality	• Data is encrypted • Server-side authentication • Session timeout after 5 minutes of inactivity
Usability	• Users will have to follow password guidelines. • System must be accessible from popular browsers, etc. Chrome, Firefox, edge.
Reliability	• The system should be available 24/7, with exclusions for maintenance and backups. • Backups should be done daily.
Performance	• Must be able to handle over 10,000 users. • The system must be scalable with increased books and users.
Supportability	• Separate environments will be used for testing, before releasing changes. • Must support English localization.

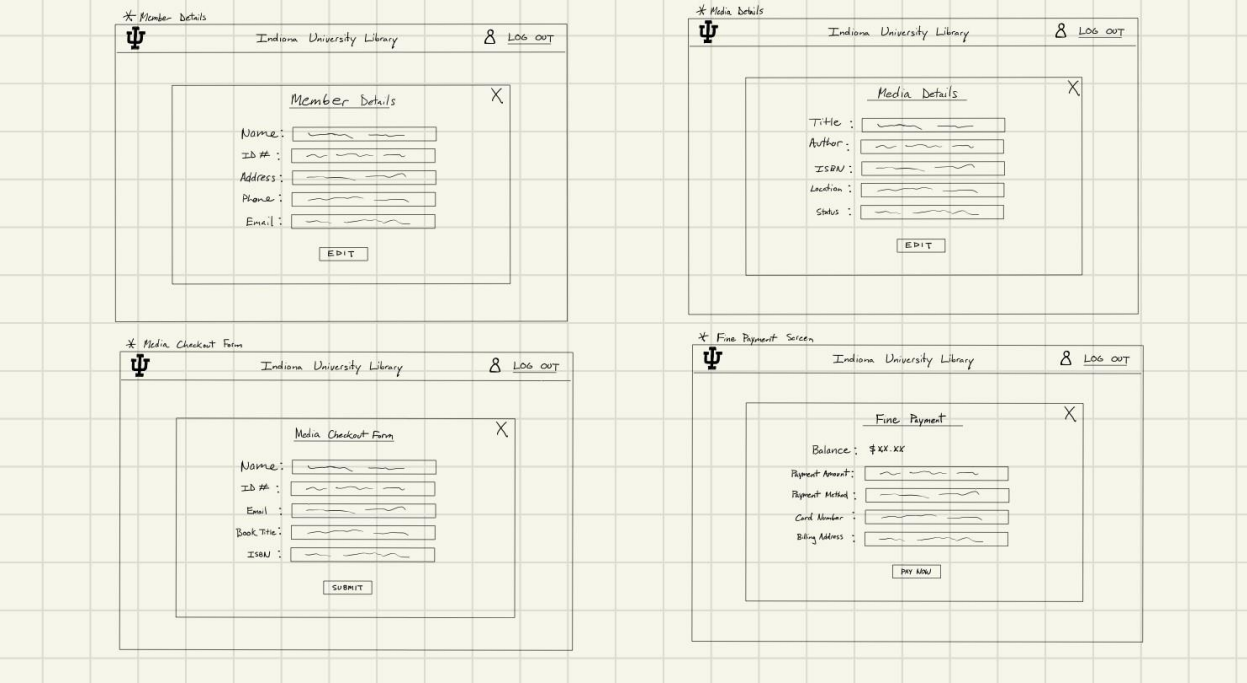
- User Interface Requirements:

The user interface will include the following screens:

- Sign On Screen
- Member registration screen

- User Home screen (Employees and Members)
- Member list (employees only)
- Member Details (Member can only view self, employees can view any)
- Media List
- Media Details
- Media checkout form (completed by the library employee)
- Fine payment screen





Plan of Work:
The developing environment should be finished by the end of the week. We have looked at our options and decided what would suit our team the most. We are also currently in the process of drafting up our sign-in page for the system to allow us to better communicate on how the system should function.

- User Registration and Sign on
- Access functionality from a user home-screen (for member and employee)
- Manage media content
    - Employees can create, read, update, and delete media
    - Members can read/view media
- Manage user account state (account balance, frozen/active, check out history)
    - User can view own account
    - Employees can view all accounts and set frozen/active
- Process checkouts via a web form
- Process returns via a web form
- Process payments to account balance (simulating payments)
- Accounts with past due checked-out media should be set to frozen status
- Administrators need the ability to create more employee / administrator accounts

Functional Requirements

Stakeholders

- Students
- General public (guests)
- University faculty
  - Library staff
  - Academia faculty (professors / deans/ advisors)
  - School administrators

Actors and Goals

Primary actors

- Member: This actor is able to log into the system to view rental items, as well as view and pay balance.

Secondary actors

- Library staff: Checks out items for members and can examine member details.
- Administrator: Views staff list and can add, search, or remove staff
- Database: Responsible for holding member and book data.

Use Cases

Member (total: 17)

- View library media list (2)
- Search library media (2)
- Check account status (active/frozen, account balance) (4)
- View checked out media and return due dates (3)
- View media checkout history (2)
- Login/Logout (2)
- Pay fee (2)

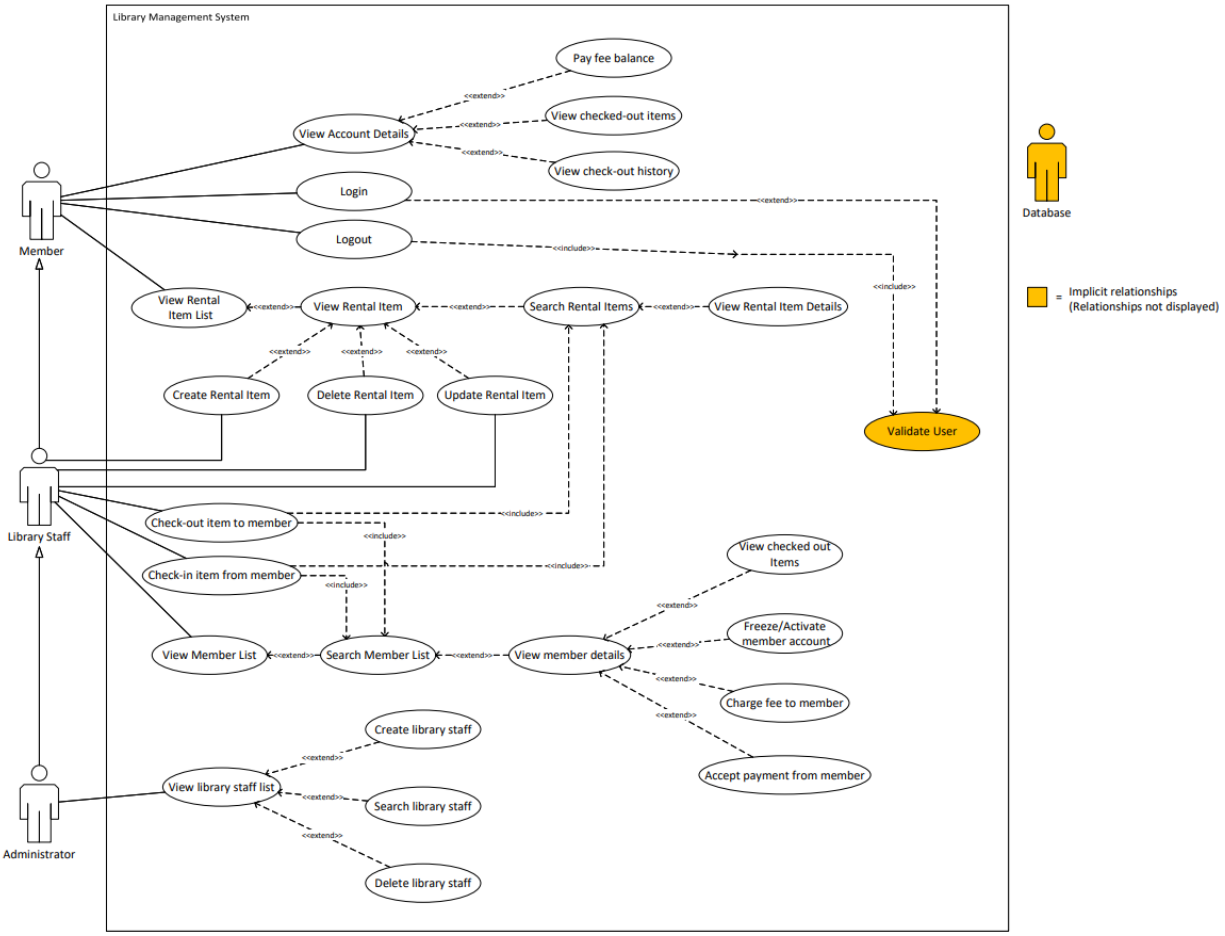
Library Staff (total: 32)

- Add media to inventory (3)
- Delete media from inventory (3)
- Update media in inventory (2)
- View library media list (2)
- Search library media (2)
- Checkout media to member (2)
- Check in returned media (2)
- View member list (2)
- Search member list (3)
- View member details (2)
- Freeze/Activate member account (3)
- Add fee to member account (2)
- Accept fee payments for Member (2)
- Login/Logout (2)

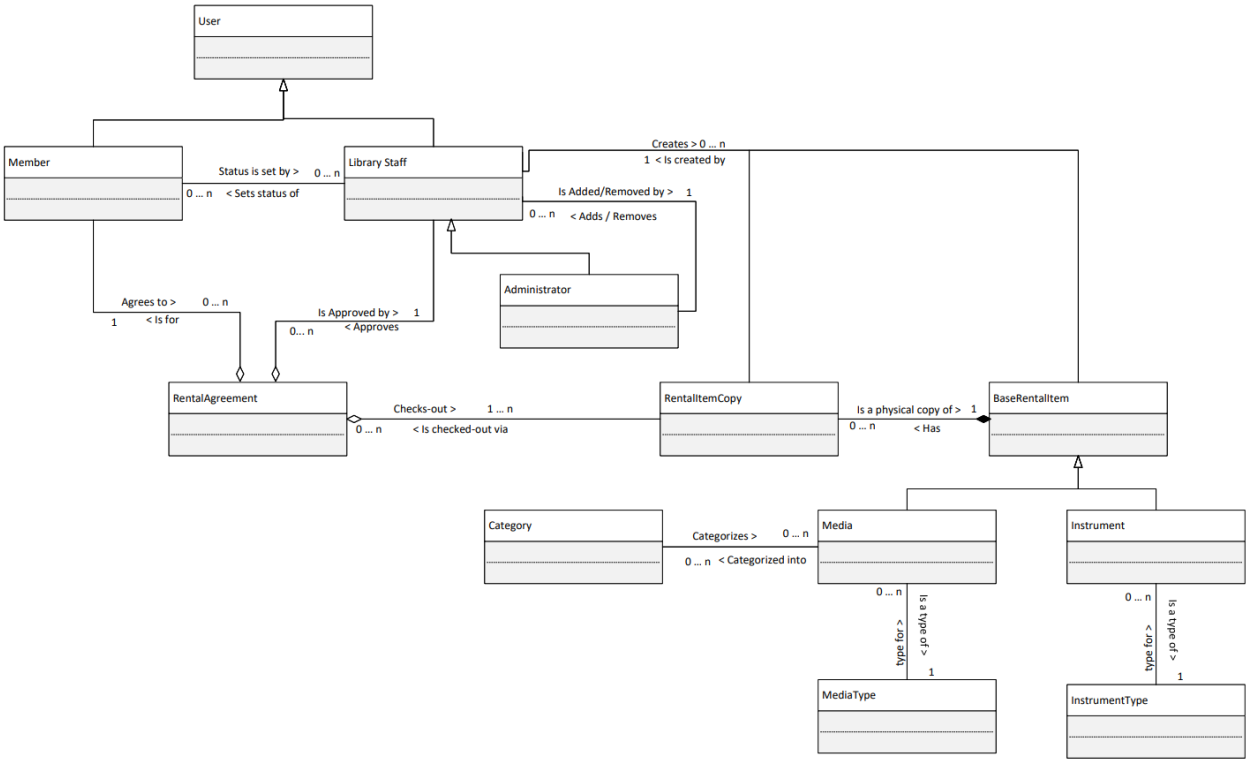
Library Administrator (total: 12)

- Add Library Staff (3)
- Delete library staff (3)
- View Library staff (2)
- Search library staff (2)
- Login/logout (2)

Use Case Diagram



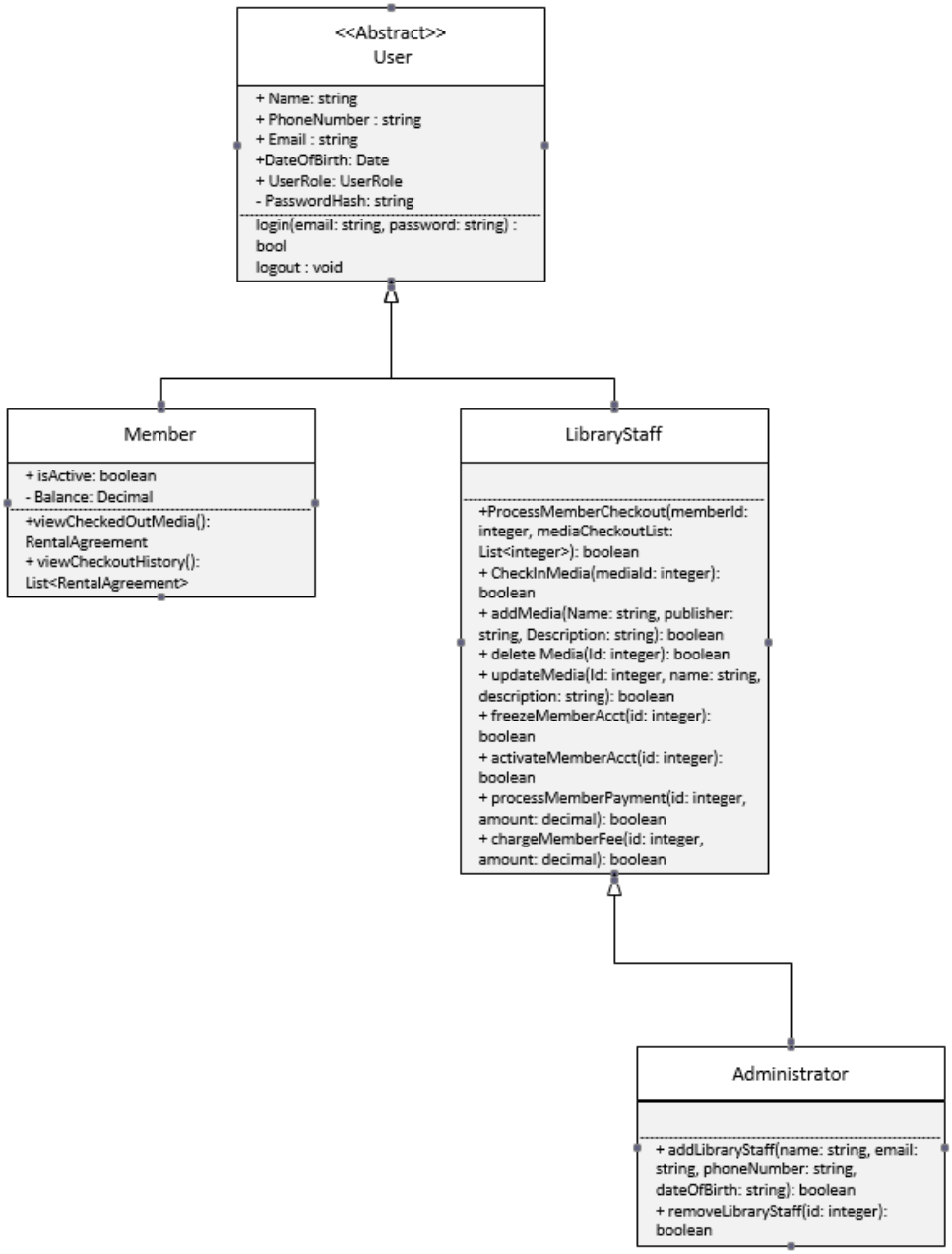
Class Diagram



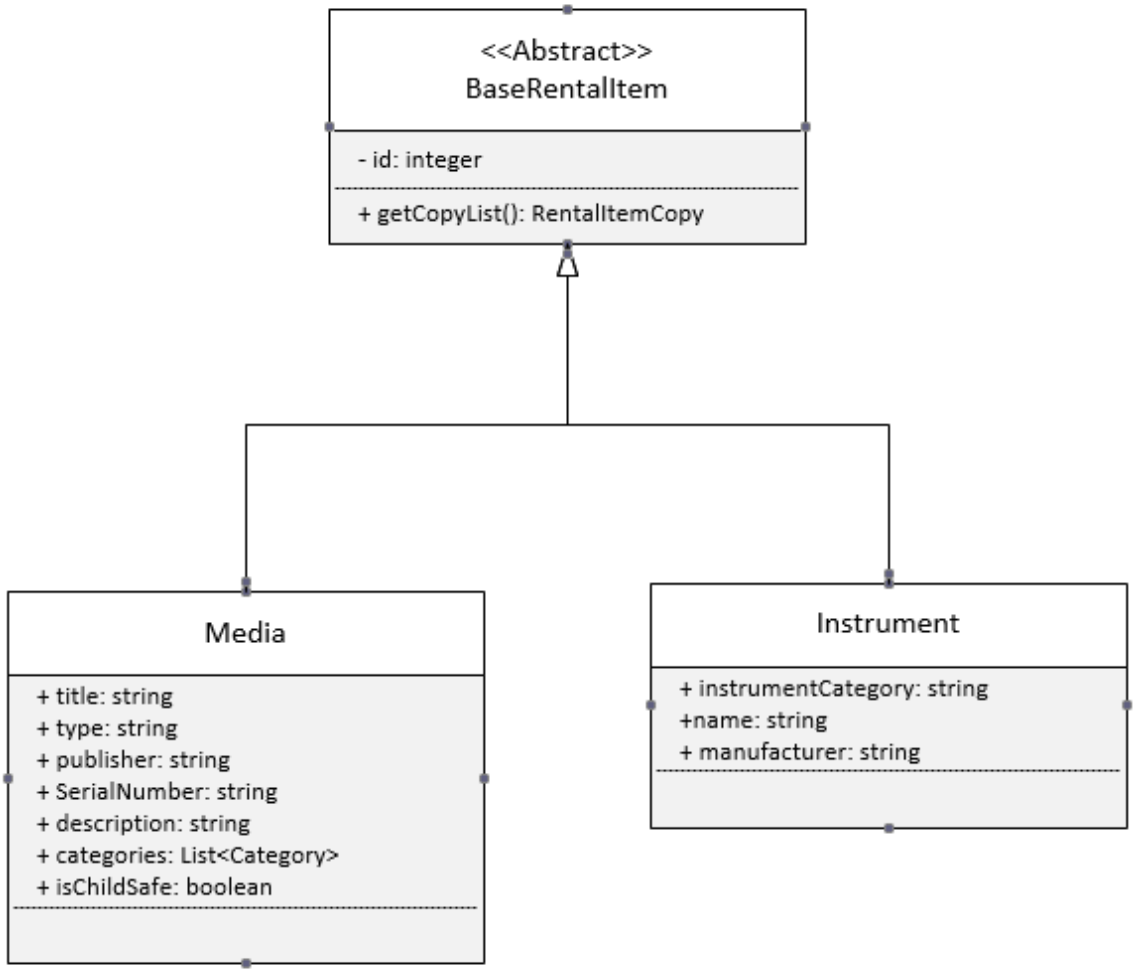
Data Types and Operation Signatures

User

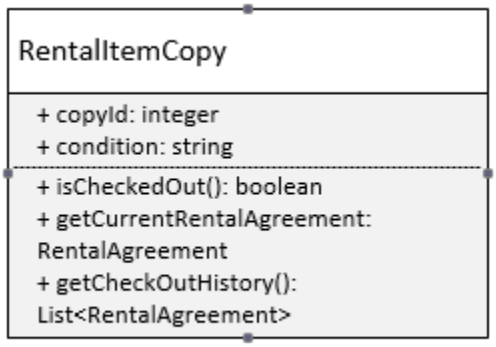
There are two types of users in the library management system, members, and library staff. Because all users must be either a member or a library staff member, the User class is abstract. Some, but not all, library staff are also system administrators with access to additional functionality.



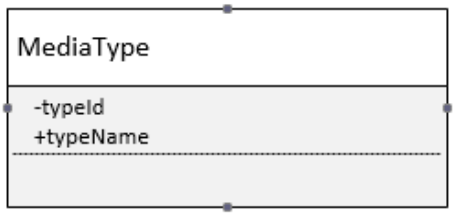
BaseRentalItem
The base rental item abstract class represents anything that can be rented or checked out of the library by members. There are two types of rental items, Media and Instruments. Media includes the enumerated types of books, CDs, DVDs, and any other form of published communication. Instruments represent musical instruments that can be checked out of the library.

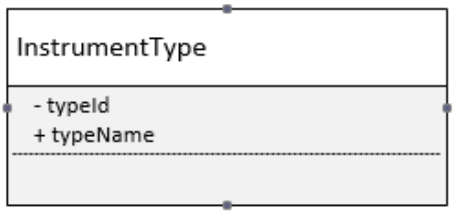


RentalItemCopy
A rental item copy represents the physical copy of a rental item. Just as a library can have many copies of the same book, there can be many rental item copies to one base rental item.

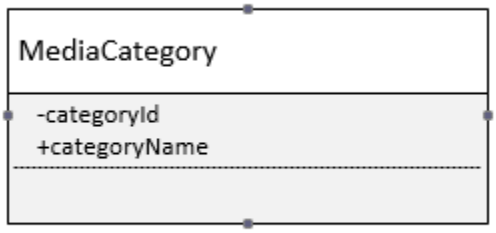


MediaType and InstrumentType
Media and instruments can both be classified by their type. Media and Instruments only belong to one type. The type is represented as an enumerated list of types. Media types may include books, DVDs, and CDs. Instrument types may include percussion, woodwind, brass, and strings for example.



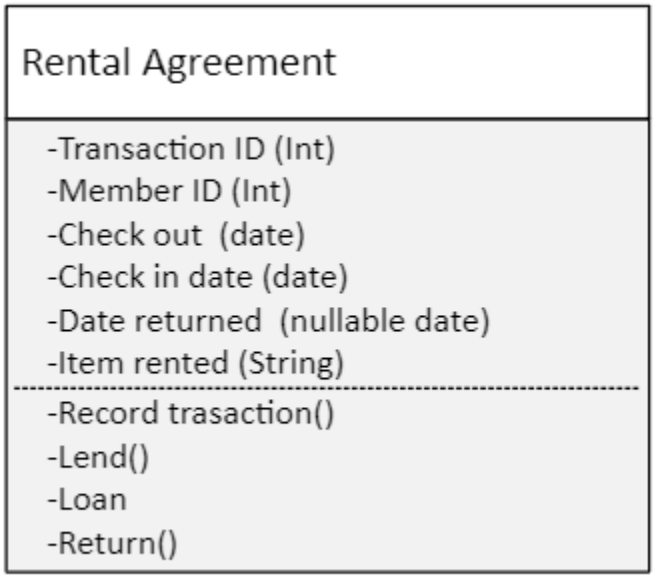


MediaCategory
Media can be classified by its category, which may include genre and/or its intended age. Categories are represented by an enumerated list of categories. Media is not limited to one category but can be labeled with many categories.

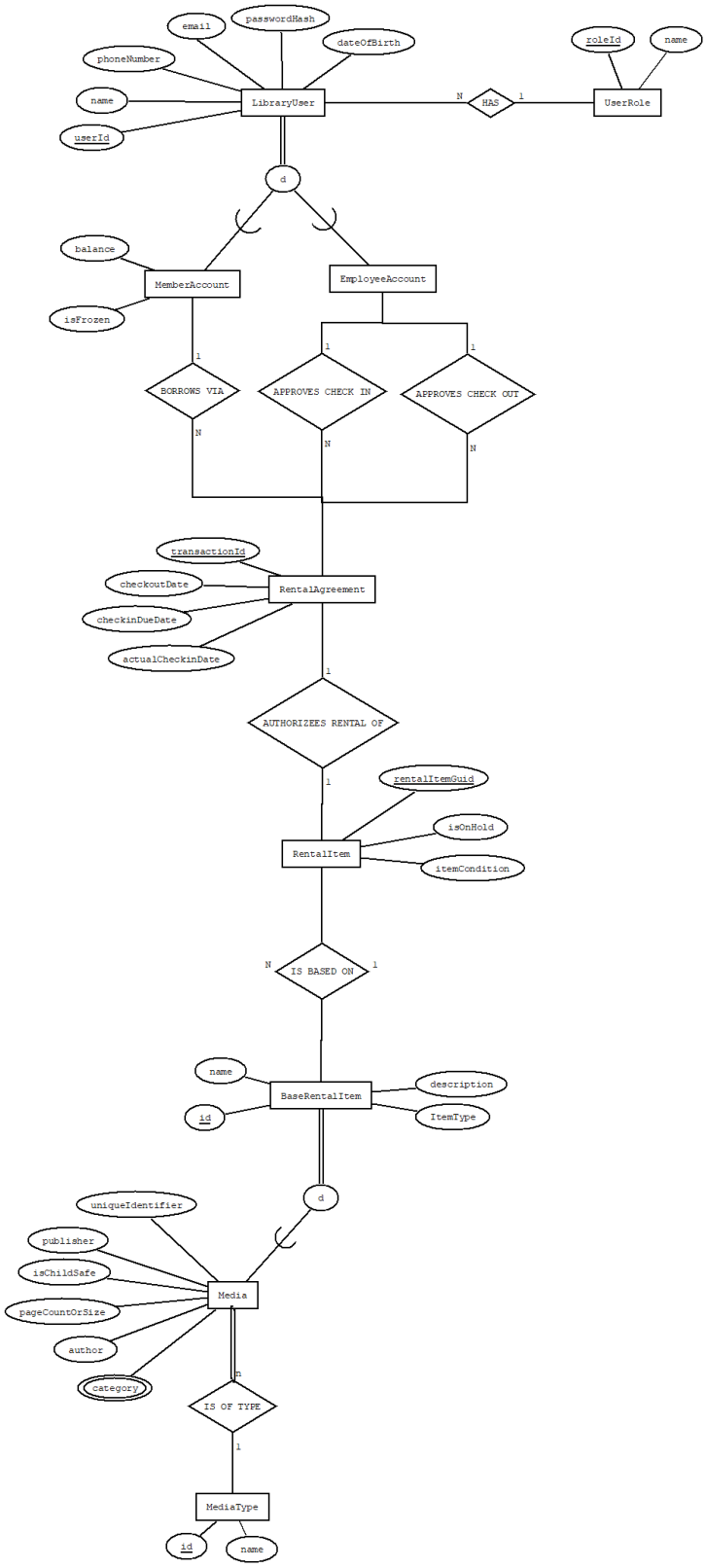


RentalAgreement

When a member checks out an item from the library, they agree to a rental agreement. The rental agreement provided details of that agreement, including references to the member and the item being rented.



Database Diagram



System Sequence Diagram and Design

Sequence Diagrams:

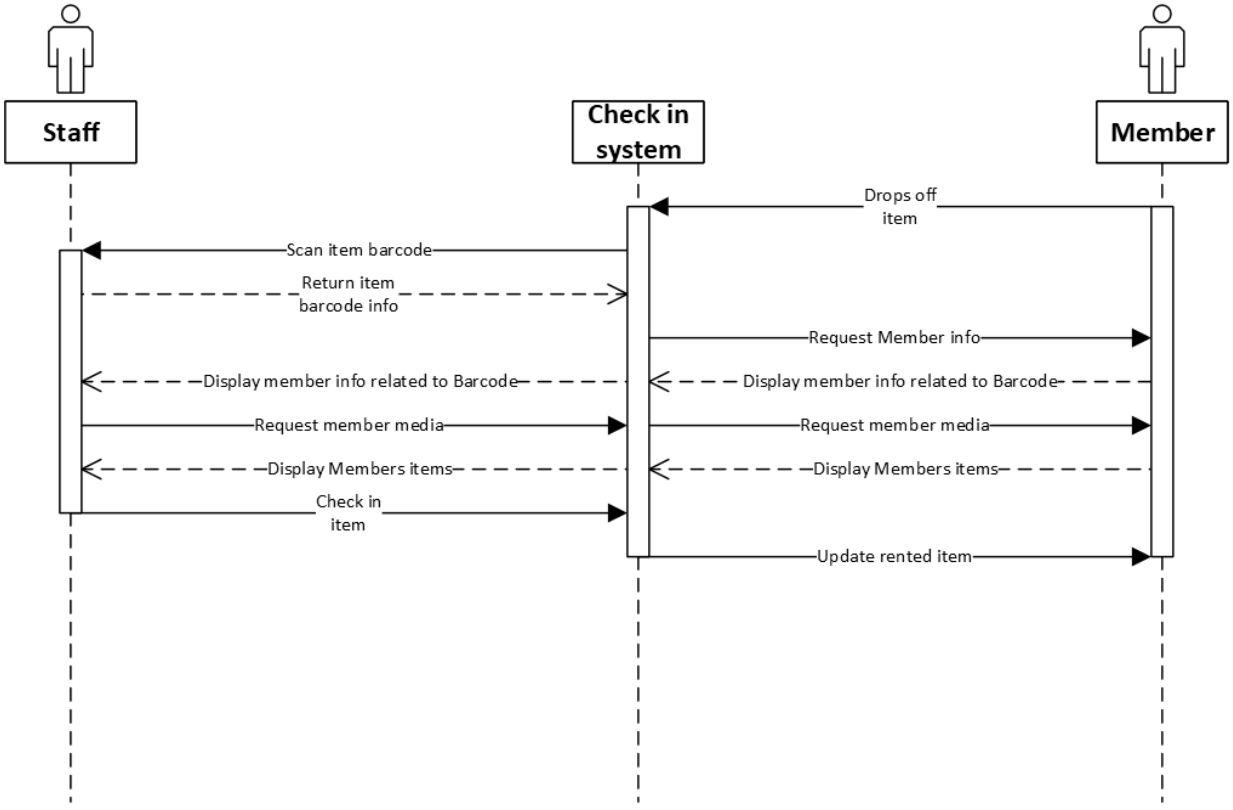
Sequence Diagram for Checking in media

Actor: Menmber
Object: Check-in counter
Interacts with the Check-in system or Staff

Steps for checking in:

1. The member drops off their item
2. Staff scans the item
3. The system reads the media's bar code information
4. The system requests the members information
5. The system displays the members information about the media
6. The staff requests the members items
7. The systems return the members item list
8. The staff checks-in the media
9. The system updates the members rented item status

Check-in system (Below)



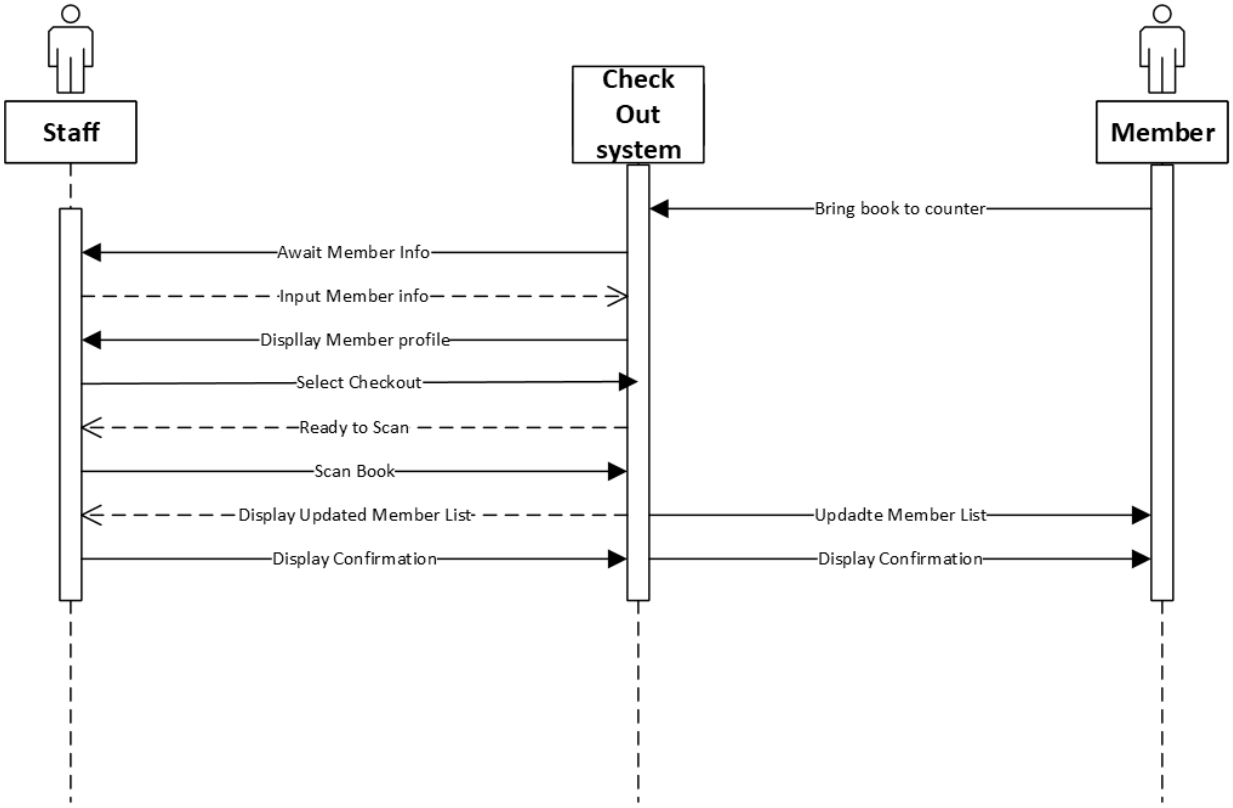
Sequence Diagram for Checking out media

Actor: Menmber
Object: Check-out counter
Interacts with the Check-out system or Staff

Steps for checking out:

1. Member brings media to counter
2. The checkout system request member information
3. Staff inputs the member information
4. The checkout displays the member's profile
5. The staff selects checkout on the system
6. The staff scans the media
7. The checkout system displays an updated list to the staff and the member's account
8. The checkout sends a confirmation that the checkout was successful to the staff and member

Check-out system (Below)



Activity Diagrams:

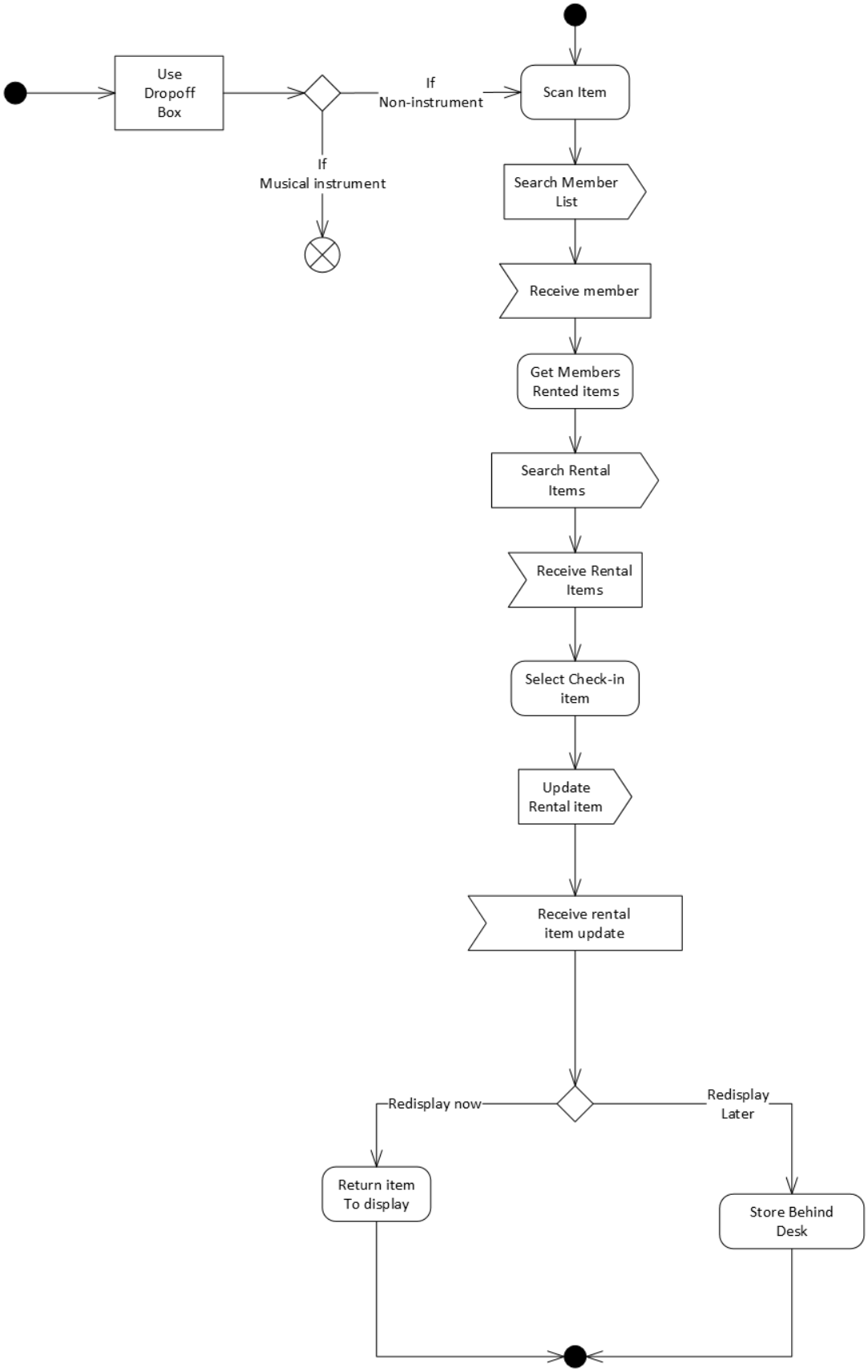
Activity diagram for checking-in media (image next page)

States:

- Initial state: The member brings media to be returned to the counter
- Final state: The staff updates the members account and returns the media to its shelf when able

Action:

The member begins the check-in process by returning the media to a drop-off point or at the check-in counter. The staff scans the item and searches the member list and receives the member's rented items list. The staff now searches the list for the item and checks in the item. The system updates the rental items status, and the staff returns the media back to its display when able.


Use
Dropoff
Box
If
Non-instrument
If
Musical instrument
Scan Item
Search Member
List
Receive member
Get Members
Rented items
Search Rental
Items
Receive Rental
Items
Select Check-in
item
Update
Rental item
Receive rental
item update
Redisplay now
Redisplay
Later
Return item
To display
Store Behind
Desk

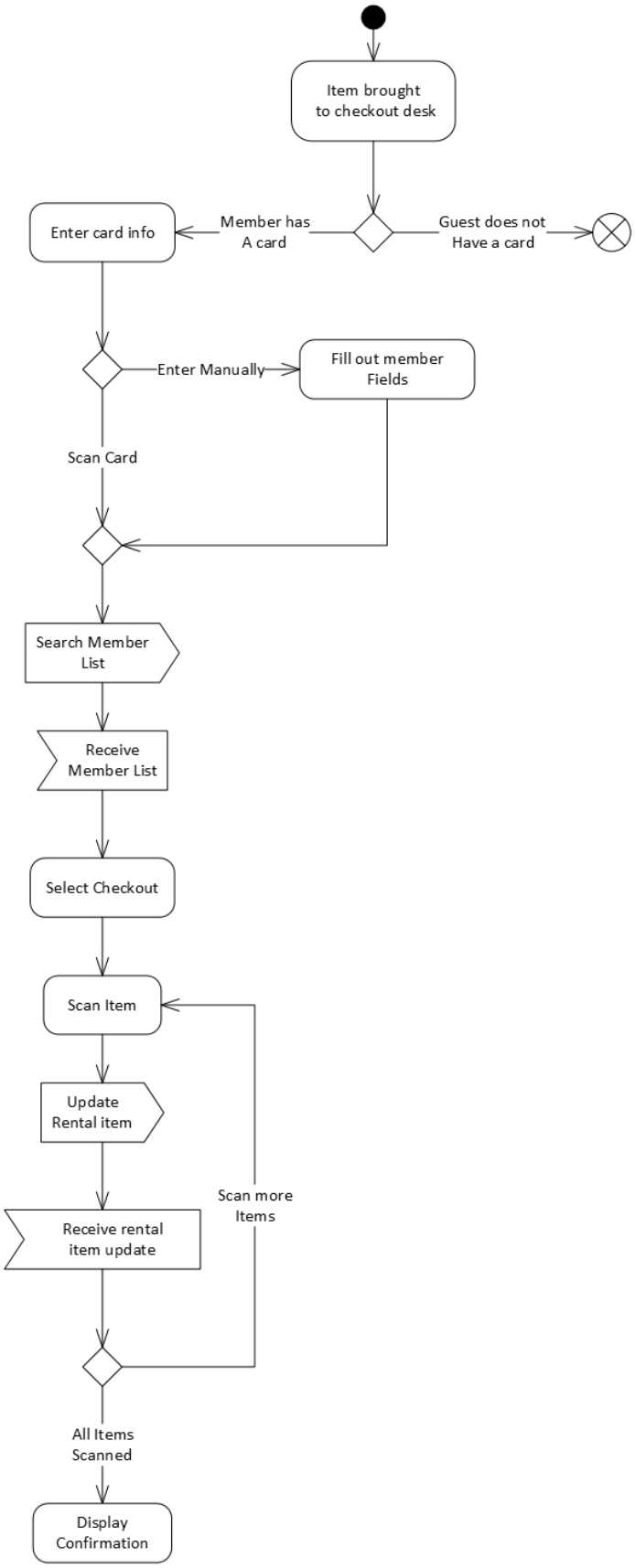
Activity diagram for checking-out media (image next page)

States:

- Initial state: The member brings an item to the checkout-desk
- Final state: 1. The member successfully checks out a book. 2. The member cannot check out a book without a card.

Action:

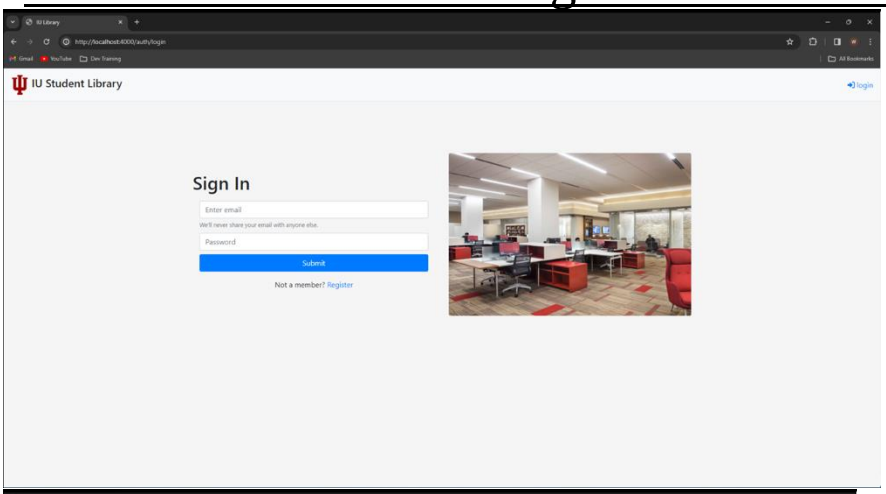
The member brings a media item to the check-out desk. The member presents their card, or they cannot checkout the media item. The staff scans the card or enters the information in manually. The checkout system searches the member list. The staff selects checkout and scans the item, and the system updates the members rental items list, then the step repeats for any additional items. When all items are scanned a confirmation is displayed and a confirmation is sent to the customer.

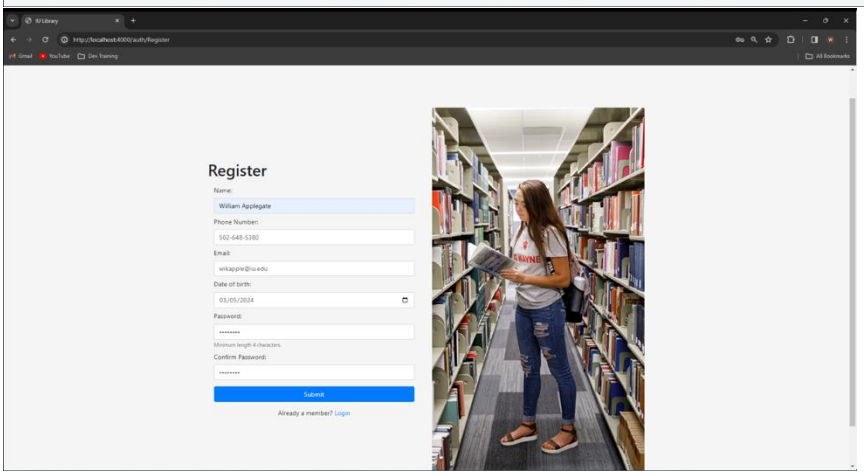

Item brought
to checkout desk
Enter card info
Member has
A card
Guest does not
Have a card
Enter Manually
Fill out member
Fields
Scan Card
Search Member
List
Receive
Member List
Select Checkout
Scan Item
Update
Rental item
Receive rental
item update
Scan more
Items
All Items
Scanned
Display
Confirmation

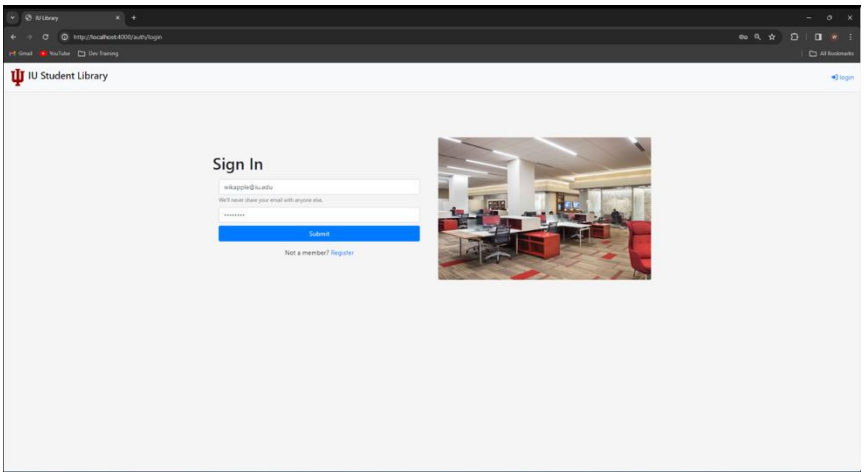
User Interface Specification

Preliminary Design

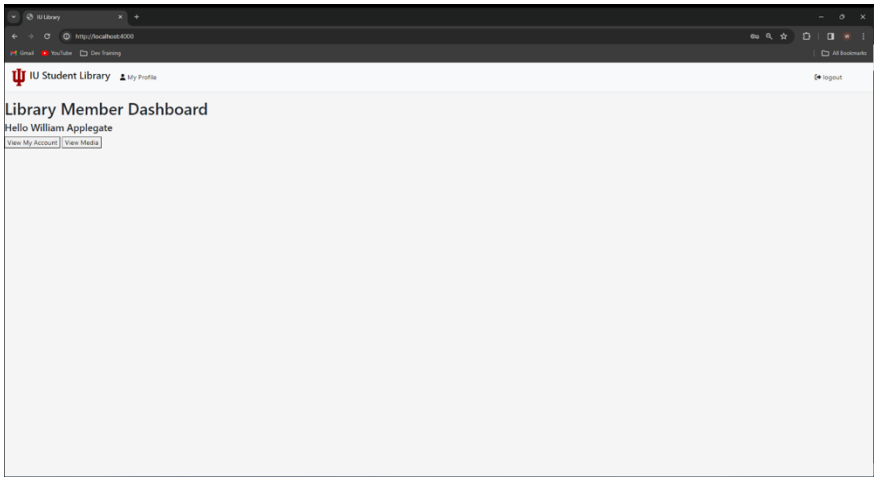
Use Case: A new user registers to become a member







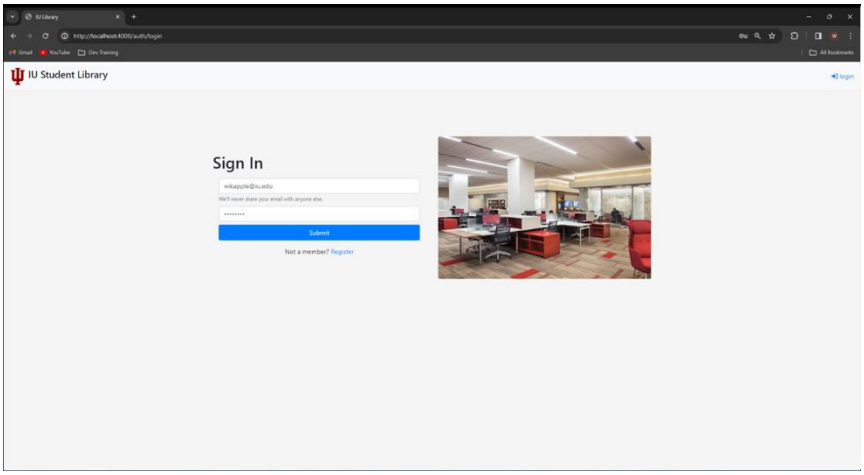
Clicks the ‘Register’ link in blue. Fills out the register form fields and then clicks blue ‘Submit’ button User is now able to sign in by submitting their email and password

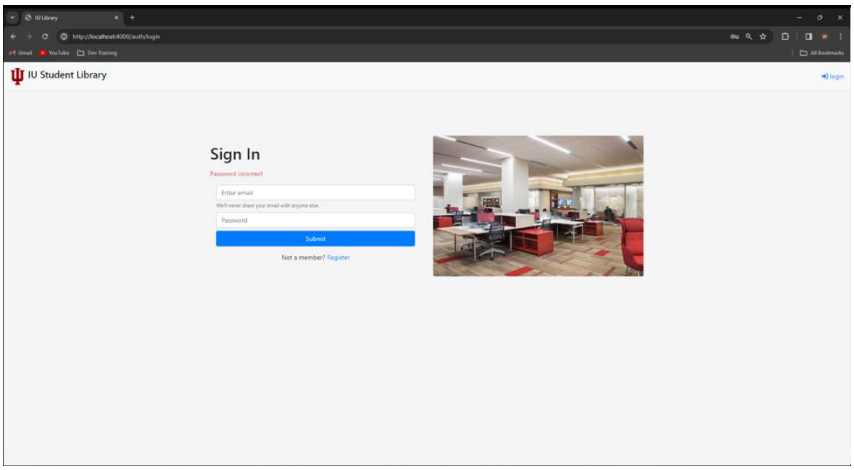


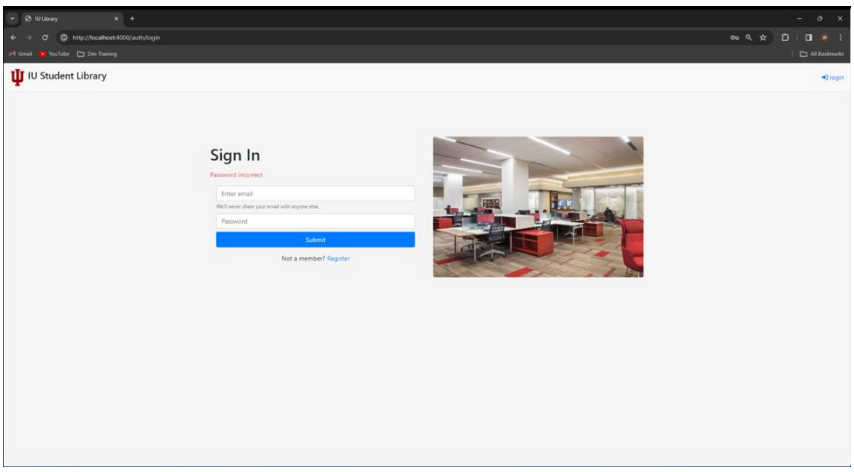
→

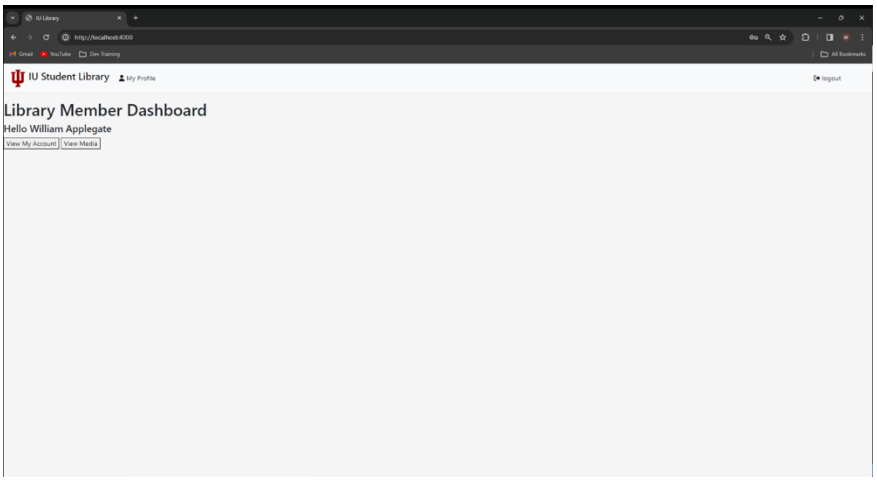
Member will be taken to their Library Member Dashboard

Use Case: User's password is incorrect



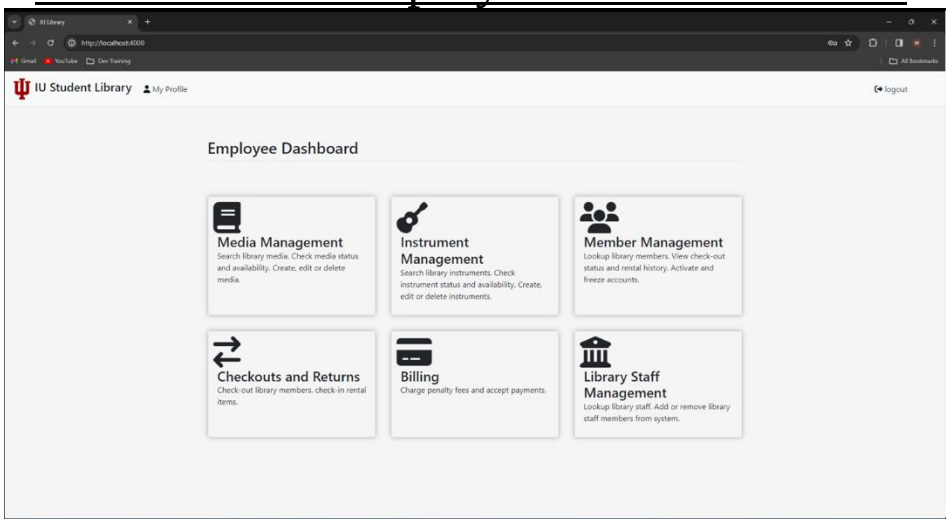


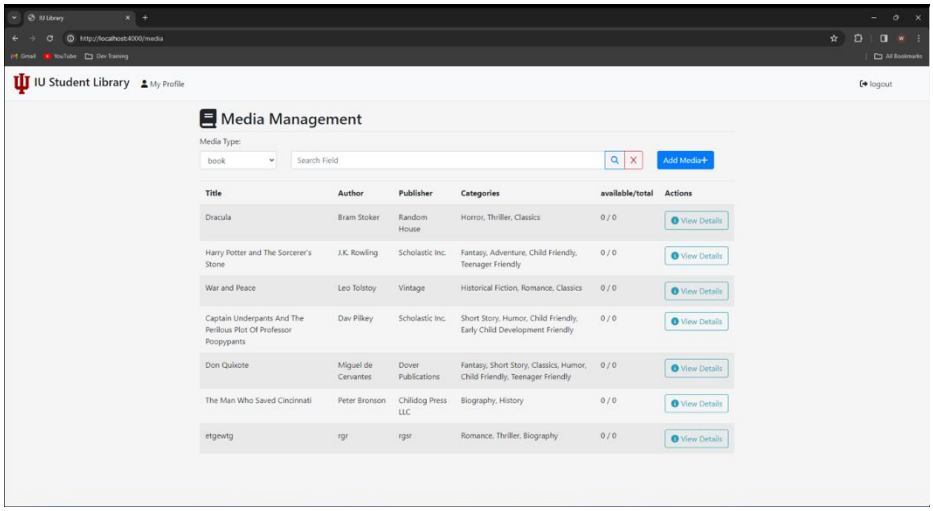


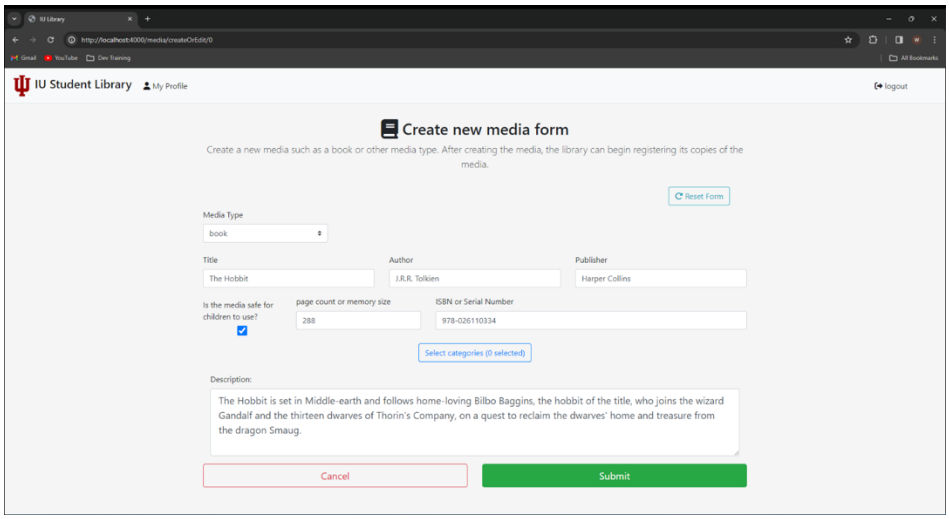


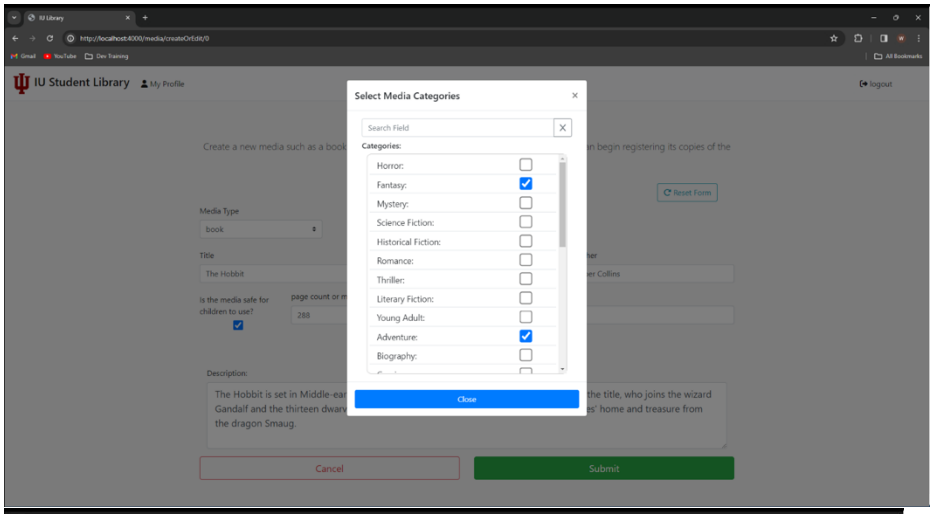
→

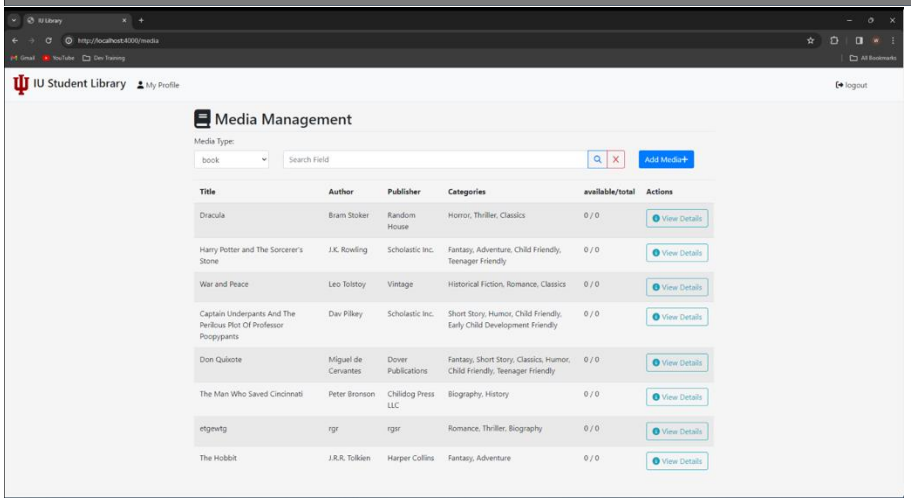
Use Case: An employee adds new media



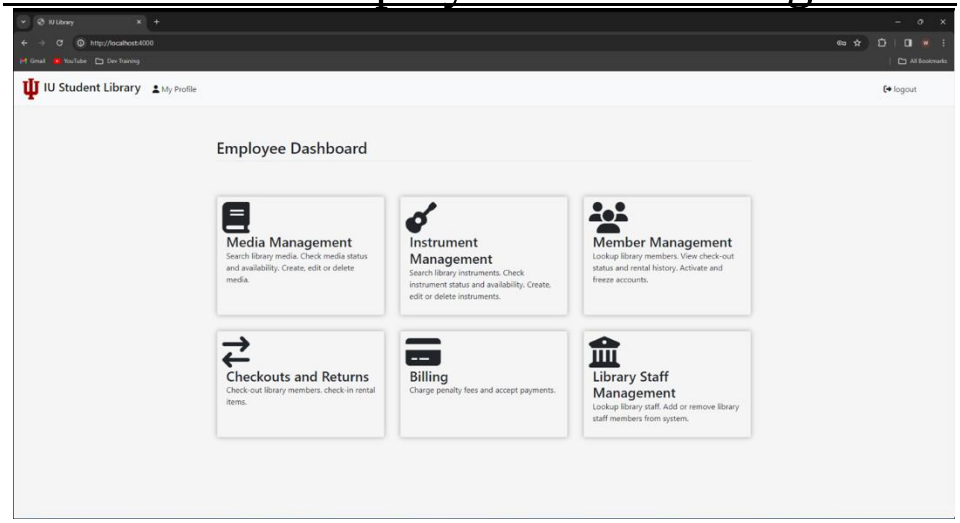


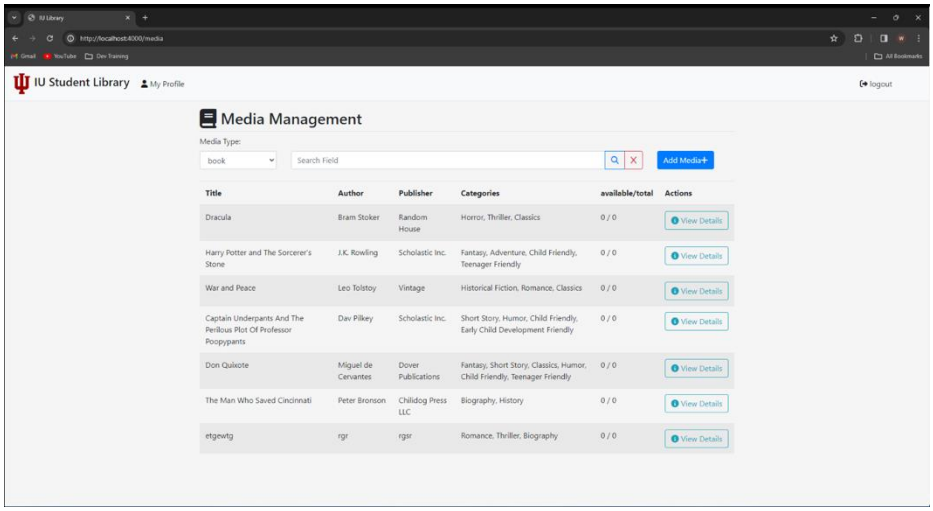

IU Student Library
My Profile
logout
Create new media form
Create a new media such as a book or other media type. After creating the media, the library can begin registering its copies of the media.
Reset Form
Media Type
book
Title
The Hobbit
Author
J.R.R. Tolkien
Publisher
Harper Collins
Is the media safe for children to use?
page count or memory size
288
ISBN or Serial Number
978-026110334
Select categories (0 selected)
Description:
The Hobbit is set in Middle-earth and follows home-loving Bilbo Baggins, the hobbit of the title, who joins the wizard Gandalf and the thirteen dwarves of Thorin's Company, on a quest to reclaim the dwarves' home and treasure from the dragon Smaug.
Cancel
Submit

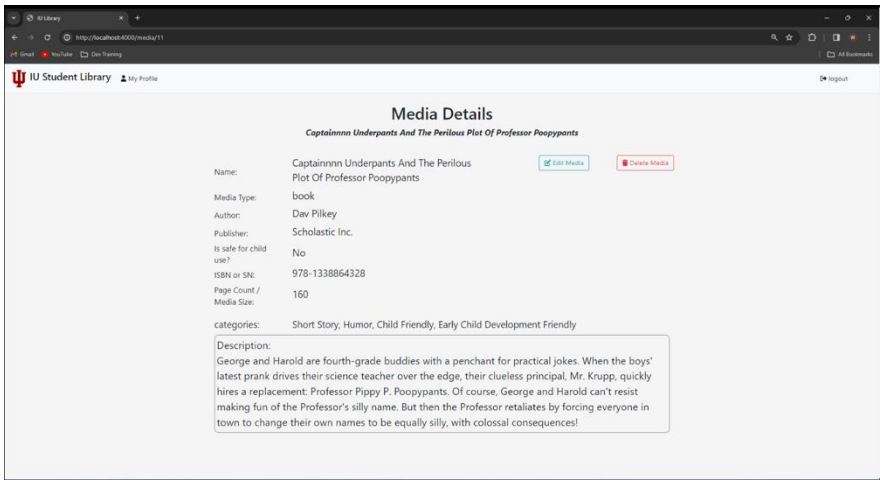

IU Student Library
My Profile
logout
Select Media Categories
Search Field
Categories:
Horror:
Fantasy:
Mystery:
Science Fiction:
Historical Fiction:
Romance:
Thriller:
Literary Fiction:
Young Adult:
Adventure:
Biography:
Close
Reset Form
Media Type
book
Title
The Hobbit
Is the media safe for children to use?
Description:
Cancel
Submit

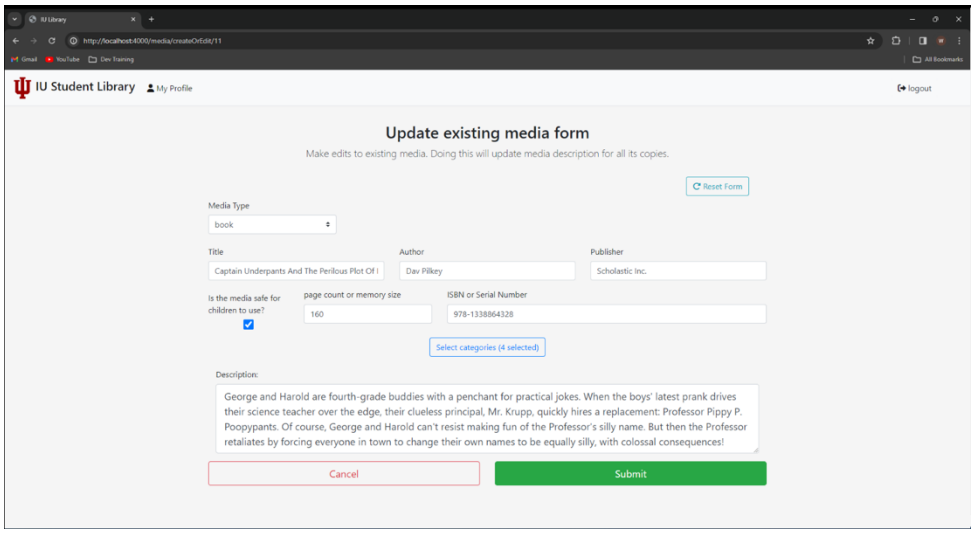

IU Student Library
My Profile
logout
Media Management
Media Type:
book
Search Field
Add Media+
Title
Author
Publisher
Categories
available/total
Actions
Dracula
Bram Stoker
Random House
Horror, Thriller, Classics
0 / 0
Harry Potter and The Sorcerer's Stone
J.K. Rowling
Scholastic Inc.
Fantasy, Adventure, Child Friendly, Teenager Friendly
0 / 0
War and Peace
Leo Tolstoy
Vintage
Historical Fiction, Romance, Classics
0 / 0
Captain Underpants And The Perilous Plot Of Professor Poopypants
Dav Pilkey
Scholastic Inc.
Short Story, Humor, Child Friendly, Early Child Development Friendly
0 / 0
Don Quixote
Miguel de Cervantes
Dover Publications
Fantasy, Short Story, Classics, Humor, Child Friendly, Teenager Friendly
0 / 0
The Man Who Saved Cincinnati
Peter Bronson
Childog Press LLC
Biography, History
0 / 0
etgewtg
rgr
rgsr
Romance, Thriller, Biography
0 / 0
The Hobbit
J.R.R. Tolkien
Harper Collins
Fantasy, Adventure
0 / 0
View Details

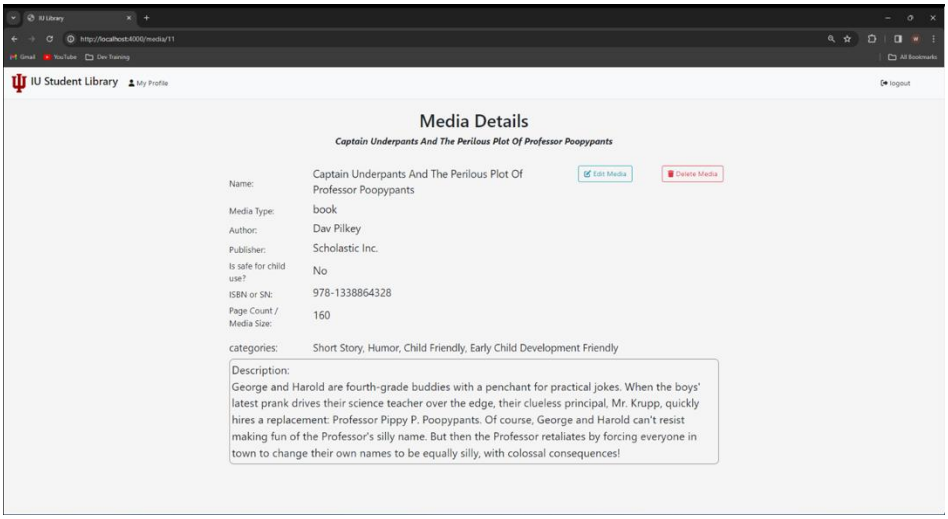
Use Case: An employee edits existing media



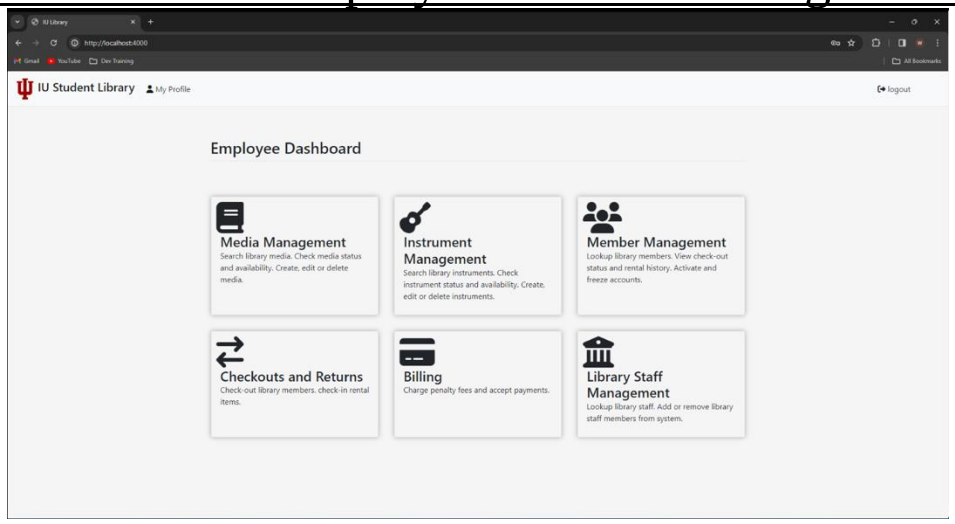


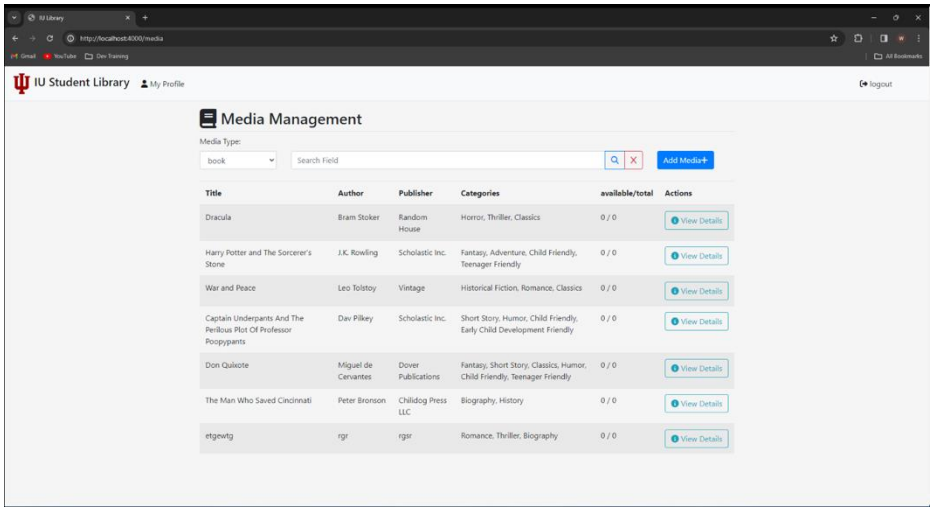


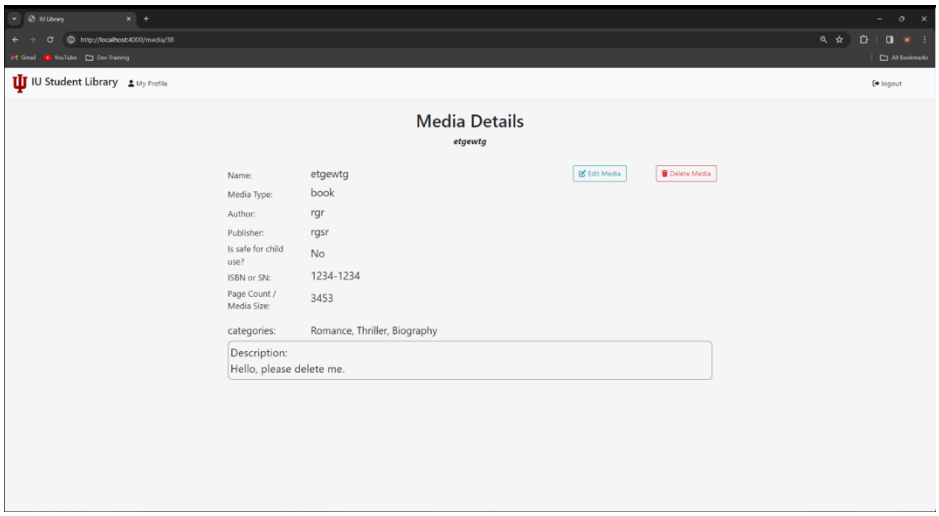

IU Student Library
My Profile
logout
Update existing media form
Make edits to existing media. Doing this will update media description for all its copies.
Reset Form
Media Type
book
Title
Captain Underpants And The Perilous Plot Of I
Author
Dav Pilkey
Publisher
Scholastic Inc.
Is the media safe for children to use?
page count or memory size
160
ISBN or Serial Number
978-1338864328
Select categories (4 selected)
Description:
George and Harold are fourth-grade buddies with a penchant for practical jokes. When the boys' latest prank drives their science teacher over the edge, their clueless principal, Mr. Krupp, quickly hires a replacement: Professor Pippy P. Poopypants. Of course, George and Harold can't resist making fun of the Professor's silly name. But then the Professor retaliates by forcing everyone in town to change their own names to be equally silly, with colossal consequences!
Cancel
Submit

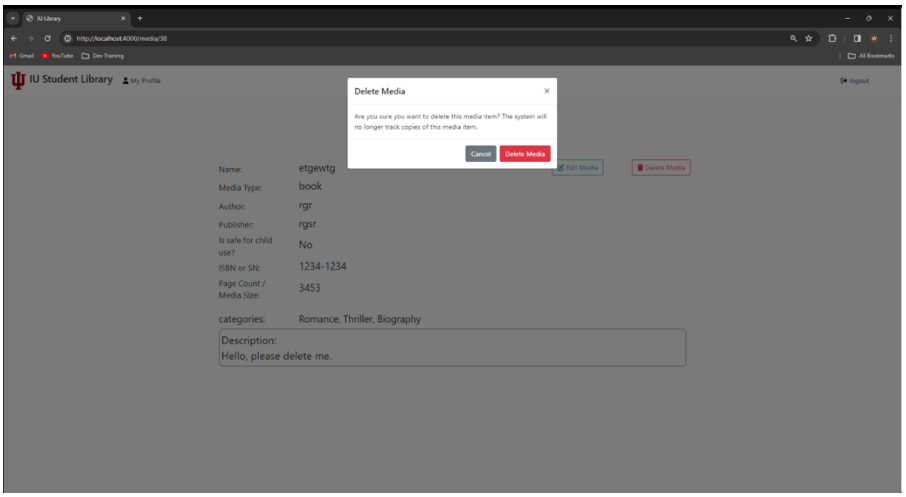

IU Student Library
My Profile
logout
Media Details
Captain Underpants And The Perilous Plot Of Professor Poopypants
Edit Media
Delete Media
Name:
Captain Underpants And The Perilous Plot Of Professor Poopypants
Media Type:
book
Author:
Dav Pilkey
Publisher:
Scholastic Inc.
Is safe for child use?
No
ISBN or SN:
978-1338864328
Page Count / Media Size:
160
categories:
Short Story, Humor, Child Friendly, Early Child Development Friendly
Description:
George and Harold are fourth-grade buddies with a penchant for practical jokes. When the boys' latest prank drives their science teacher over the edge, their clueless principal, Mr. Krupp, quickly hires a replacement: Professor Pippy P. Poopypants. Of course, George and Harold can't resist making fun of the Professor's silly name. But then the Professor retaliates by forcing everyone in town to change their own names to be equally silly, with colossal consequences!

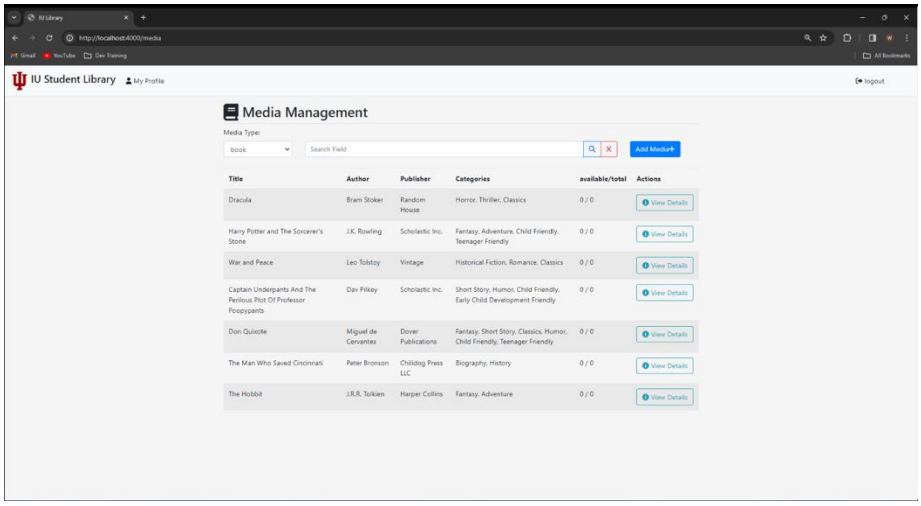
Use Case: An employee deletes existing media








IU Student Library
My Profile
logout
Delete Media
Are you sure you want to delete this media item? The system will no longer track copies of this media item.
Cancel
Delete Media
Edit Media
Delete Media
Name: etgewtg
Media Type: book
Author: rgr
Publisher: rgsr
Is safe for child use? No
ISBN or SN: 1234-1234
Page Count / Media Size: 3453
categories: Romance, Thriller, Biography
Description:
Hello, please delete me.


IU Student Library
My Profile
logout
Media Management
Media Type:
book
Search Field
Add Media
Title
Author
Publisher
Categories
available/total
Actions
Dracula
Bram Stoker
Random House
Horror, Thriller, Classics
0 / 0
Harry Potter and The Sorcerer's Stone
J.K. Rowling
Scholastic Inc.
Fantasy, Adventure, Child Friendly, Teenager Friendly
0 / 0
War and Peace
Leo Tolstoy
Vintage
Historical Fiction, Romance, Classics
0 / 0
Captain Underpants And The Perilous Plot Of Professor Poopypants
Dav Pilkey
Scholastic Inc.
Short Story, Humor, Child Friendly, Early Child Development Friendly
0 / 0
Don Quixote
Miguel de Cervantes
Dover Publications
Fantasy, Short Story, Classics, Humor, Child Friendly, Teenager Friendly
0 / 0
The Man Who Saved Cincinnati
Peter Bronson
Chilidog Press LLC
Biography, History
0 / 0
The Hobbit
J.R.R. Tolkien
Harper Collins
Fantasy, Adventure
0 / 0
View Details

User Effort Estimation

Usage Scenario	Navigation	Clicks	Keystrokes
New user registers to become member	Sign-in, Register form, Sign-in, Dashboard	3	<50
User's password is incorrect	Sign-in	2	<20
An employee adds new media	Dashboard, Media Management, Create new media form	8	<100
An employee edits existing media	Dashboard, Media Management, Media Details, Update exisiting media form	4	<10
An employee deletes media	Dashboard, Media Management, Media Details	4	0

Traceability Matrix

System Requirements

No.	Priority Weight (1-5: 1: lowest, 5: highest)	Description
REQ1	4	Only a library employee can check media out to members using a user interface.
REQ2	4	Users should be able to search the library's catalogue.
REQ3	5	Only a library employee can check media back in via a user interface.
REQ4	5	Only a library employee should be able to look up any library member and view their account details. Account details include member name, contact information, media the member currently has checked out, current fines, and if the account is frozen or active.
REQ5	5	Members and library employees can login and log out of application
REQ6	4	Only library employees should be able to create media items
REQ7	4	Only library employees should be able to update media items
REQ8	5	Only library employees should be able to delete media items
REQ9	4	Library members should be able to view details on their own library account, including fees and if their account is frozen
REQ10	5	Library members should not be able to view details of any other library accounts not belonging to them.
REQ11	3	A user must be able to use the application to pay any fees
REQ12	3	A should not be able to checkout media if their account is frozen or they have rental items past due.

Use Cases

No.	Description
UC1	Login and logout of application
UC2	Add media items for rent
UC3	Remove media items for rent

UC4	Edit media items for rent
UC5	View list of media items for rent
UC6	Checkout media items
UC7	Check-in media items
UC8	View checkout history of members and media
UC9	Freeze and unfreeze accounts
UC10	Assign fees to users
UC11	Pay Account Fees

Traceability Matrix

Req't	**PW**	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11
REQ1	4	X	X				X					
REQ2	4	X	X			X						
REQ3	5	X						X				
REQ4	5	X							X	X	X	
REQ5	5	X	X	X	X	X	X	X	X	X	X	X
REQ6	4	X	X									
REQ7	5	X			X							
REQ8	5	X		X								
REQ9	4	X										X
REQ10	5	X										
REQ11	3											X
REQ12	3						X					
Max PW		5	5	5	5	5	5	5	5	5	5	5
Total PW		46	17	9	9	9	12	9	10	10	10	17

System Architecture and System Design

Architecture Styles

The Library Management System uses a client-server architecture. The client side can be run on many computers with many users that send requests to the server. The client runs on a modern web browser. The server uses a Model View Controller (MVC) design pattern to separate concerns within the system. The MVC design includes the following separate components:

1. Router: The router received URL requests from the client and redirects the request to which controller should process each request.
2. Model: The model represents data from the business domain, such as media items and rental agreements.
3. Controller: The controller processes requests from the client, creates the model, and renders the view html pages for the client.
    a. API Controller: In addition to standard MVC controllers, the application also includes API controllers to process and return data to and from the client in JSON format.
4. View: The view is what is rendered to the client in the form of HTML, CSS, and JavaScript.

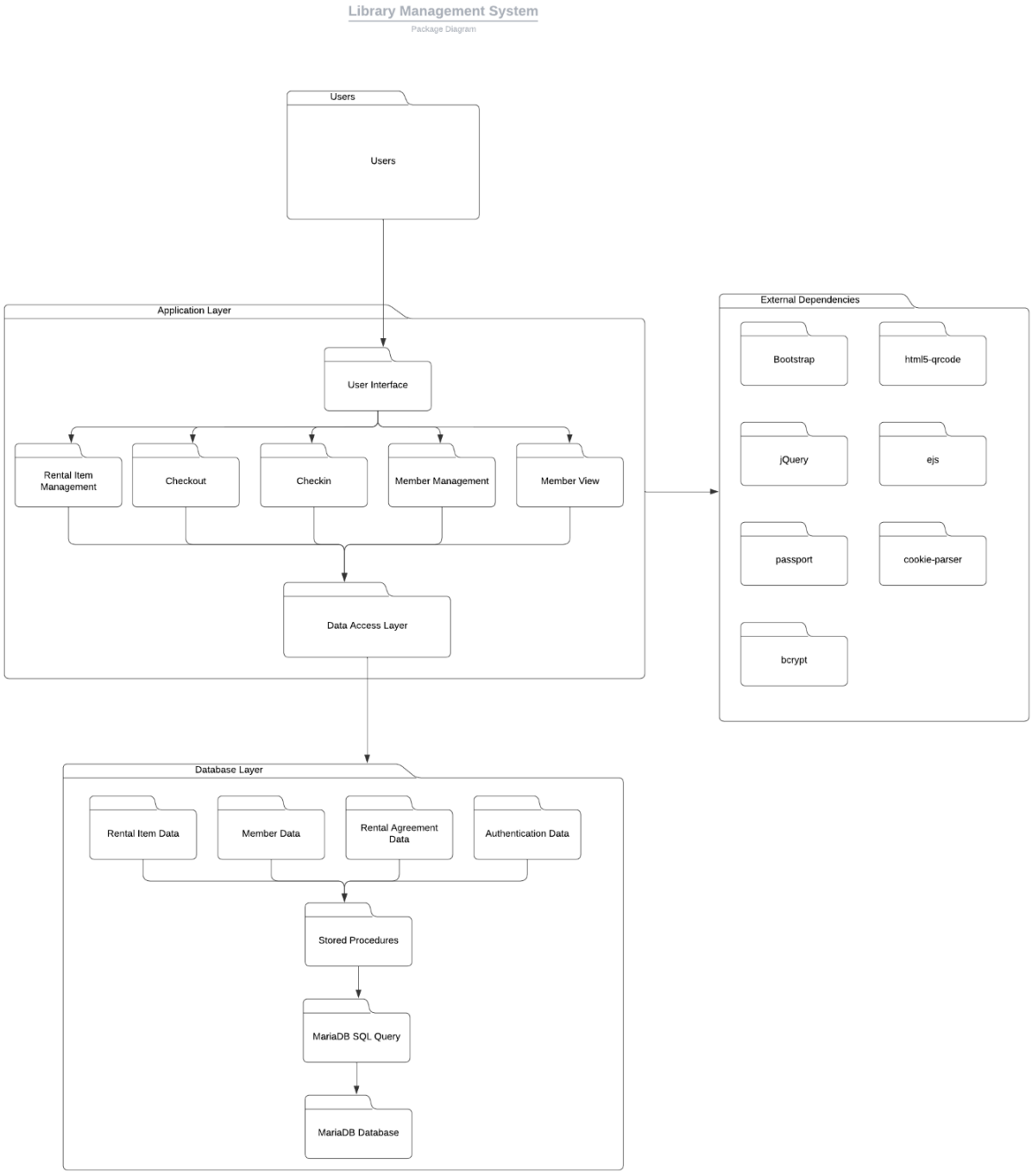
The server uses Node.js for the MVC application. The system stores data in a MariaDB database. The view of the MVC application uses embedded JavaScript (EJS) as a templating language to render dynamic HTML pages to the user.

The HTML pages rendered to the user include the Bootstrap CSS library for styling. The pages also include the jQuery JavaScript library to create more dynamic web pages and allow for JavaScript ajax API calls from the client to the server.

Identifying Subsystems

The subsystems of the application consist of the user, server, and database systems. The user interacts with the application via the user interface in the form of HTML, CSS, and JavaScript sent from the server to the user's client browser. The application layer consists of the Node.js application which processes requests, accesses the database, and returns information to the user. The Node.js application also includes external package dependencies. The database layer includes a MariaDB instance that persists the data for the application. The database layer also includes a folder of stored procedures that execute SQL queries on the database.

UML Package Diagram



Persistent Data Storage

This system requires data to be saved outside of a single execution. The persistent objects that are being saved are the users, fees, and catalogued items. These are stored in a relational database. The project will be using MariaDB.

Global Control Flow

Execution Orders

The library management system adopts an event-driven architecture, offering users a flexible and intuitive experience. Within the web interface, each interaction – navigating between pages, submitting forms, or initiating actions – generates an event that the system responds to. This design allows for dynamic page content that adapts to the user's context and the current database state. Users can seamlessly transition between core functionalities like adding books, checking outstanding fees, or editing item information without a rigid, sequential workflow. The system allows users to tailor their interactions according to their individual needs and preferences.

Time dependency

The system adopts an event-driven architecture, responding to user actions or system-generated events rather than employing explicit timers. While certain processes, such as overdue fee calculations, occur periodically, they are not constrained by strict real-time deadlines. The system prioritizes responding to events and completing tasks, as opposed to operating within rigid time intervals.

Concurrency

The Library Management System uses multiple threads to check for requests and then process them. A single thread runs in an event loop as it continually checks for requests that need to be processed. When it receives a request, it processes it asynchronously. The request is sent to another thread with a call back function to be executed when the task is completed. This allows the main thread to continue checking for requests while other threads handle processing them.

Hardware Requirements

The Library Management System is a web application. The benefits of a web application are that it can be used by any hardware or operating system that supports a modern and up-to-date web browser. The requirements for individual browsers may vary, however we recommend at least the following minimum hardware requirements:

- Client-side requirements:
  - Display: Minimum: 640 x 480 pixels recommended
  - Computing Device: Desktop/PC, laptop, or mobile device (tablet/smartphone)
  - Web Browser: Any modern web browser
  - Peripherals: Camera and scanner
- Server Requirements:
  - Memory: Minimum 1GB RAM
  - Storage: Minimum 1 GB hard drive
  - Web Server: MariaDB

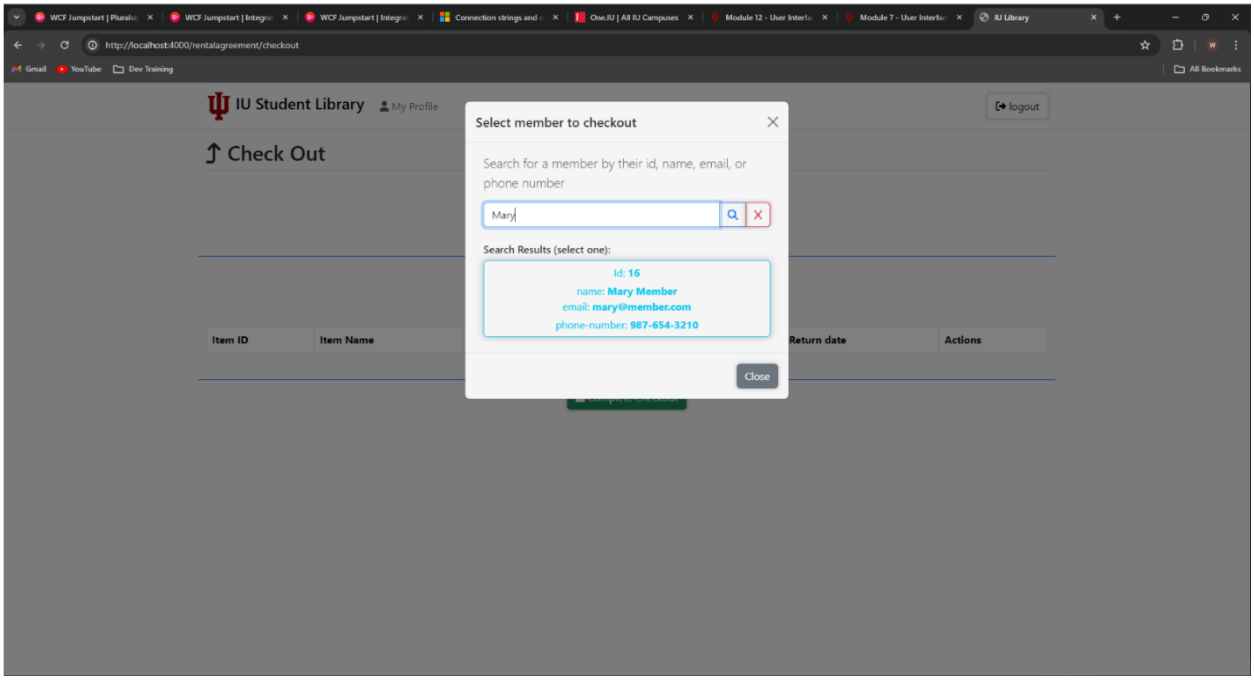
User Interface Design and Implementation

We have not made changes to the User Interface Specification in Module 7. The user interface has remained the same according to what was provided.

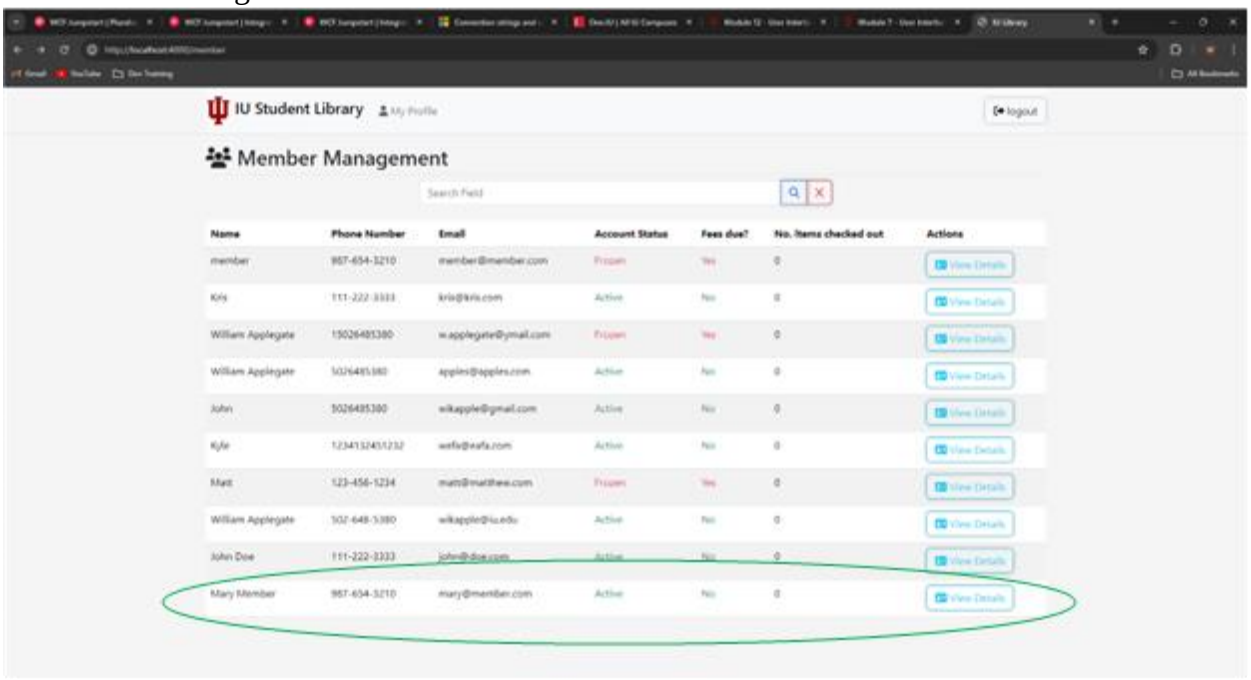
Screenshots:

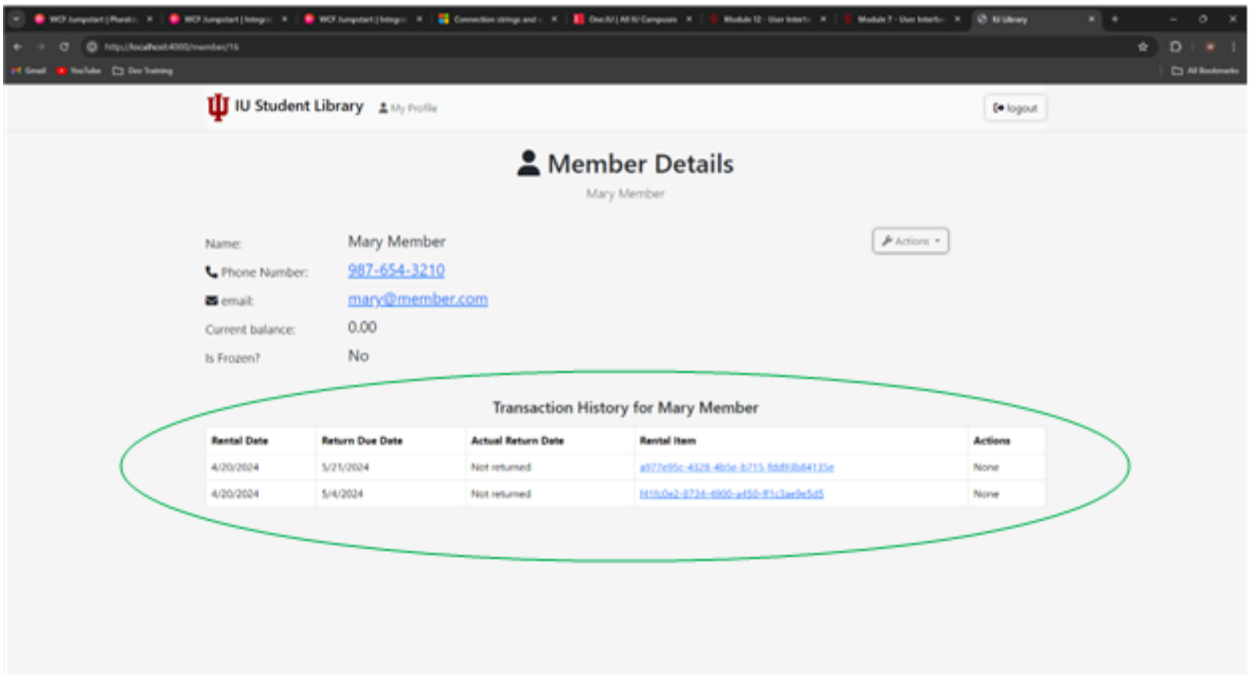
Checkout:

Complete Checkout Form:

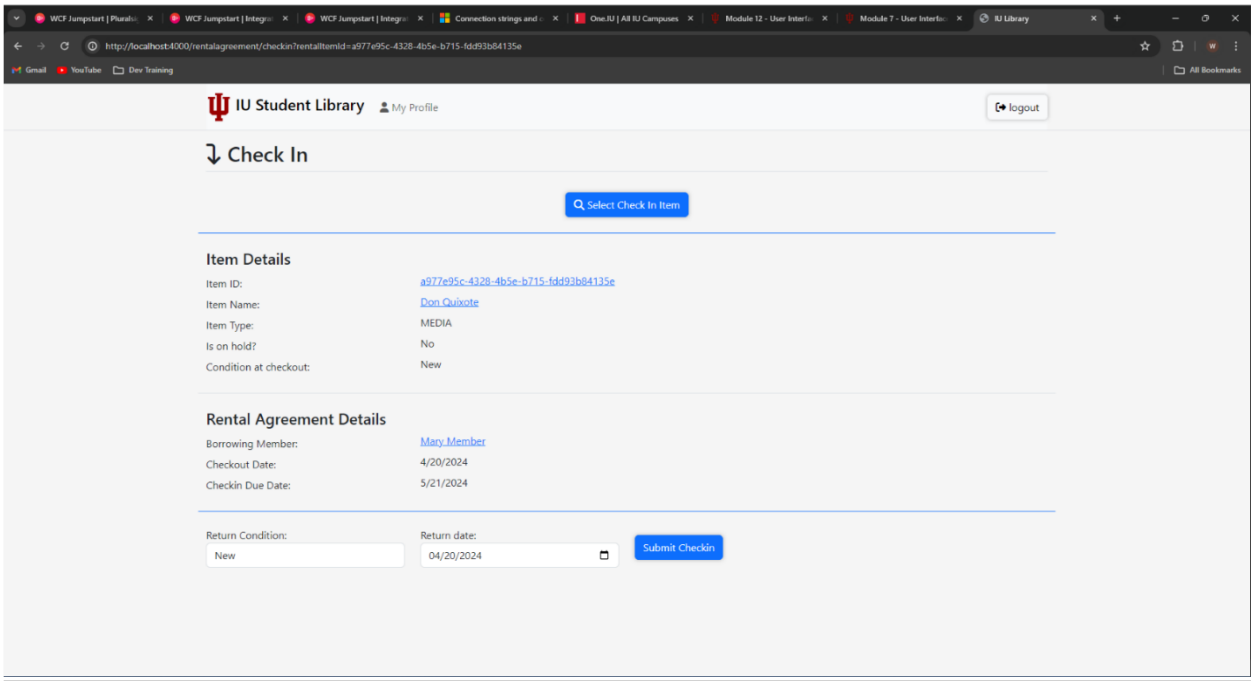


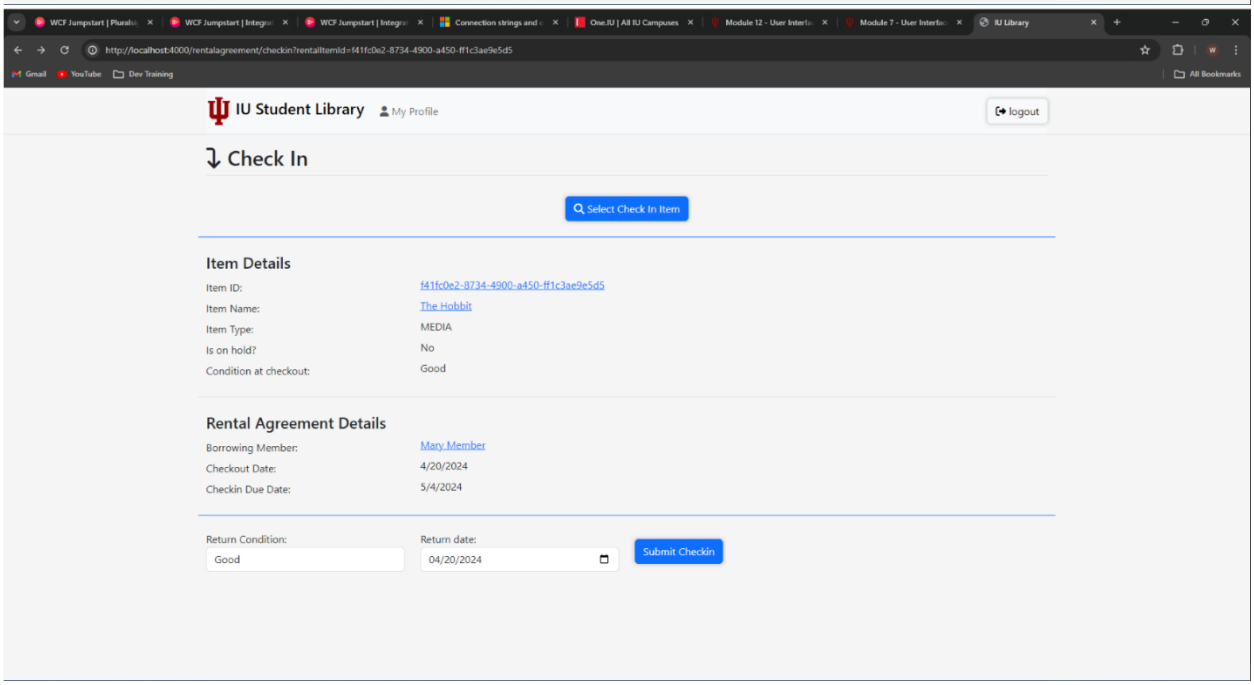
Confirm Rental Agreement exists:

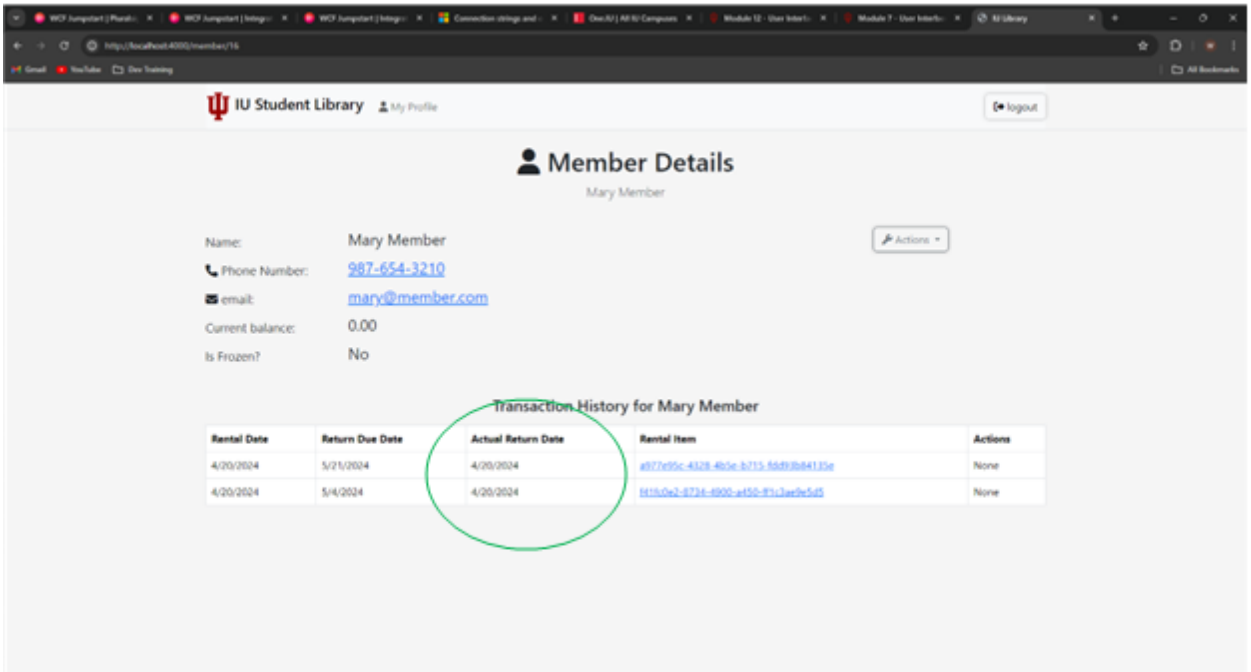




Check In:


IU Student Library
My Profile
logout
Check In
Select Check In Item
Item Details
Item ID: a977e95c-4328-4b5e-b715-fdd93b84135e
Item Name: Don Quixote
Item Type: MEDIA
Is on hold? No
Condition at checkout: New
Rental Agreement Details
Borrowing Member: Mary Member
Checkout Date: 4/20/2024
Checkin Due Date: 5/21/2024
Return Condition: New
Return date: 04/20/2024
Submit Checkin


IU Student Library
My Profile
logout
Check In
Select Check In Item
Item Details
Item ID: f41fc0e2-8734-4900-a450-ff1c3ae9e5d5
Item Name: The Hobbit
Item Type: MEDIA
Is on hold? No
Condition at checkout: Good
Rental Agreement Details
Borrowing Member: Mary Member
Checkout Date: 4/20/2024
Checkin Due Date: 5/4/2024
Return Condition: Good
Return date: 04/20/2024
Submit Checkin



Design of Tests

- Test cases that will be programmed and used for unit testing of the system:
  - Registration
    - Success – User fills out the register form and then clicks submit. The user is sent back to the login screen and their account is registered.
  - Sign in page
    - Sucess – The user should be able to access the dashboard by entering in valid credentials.
    - Failure – The user should be met with a notification saying the login credentials are invalid when using incorrect credentials or a nonexistent account.
    - Clicking the Register link will take the user to the registration page
  - Media management
    - Staff must be able to create, update, and delete media items.
    - Members must be able to view the list of media items available.
  - Member management
    - Staff must be able to look up members and see details on members, including their balance, contact information, and transaction history. Members must be able to view their own account details.
  - Checkouts and returns
    - Staff must be able to submit a form to check out rental items to members and then check those items back in when returned. The rental agreement should be viewable on the member's details screen and the rental item's details screen.

- Billing
  - Staff must be able to process payments from users. Processed payments should be reflected in the paying member's balance, which is displayed on the member details screen.

- Integration Testing strategy and plans on how it will be conducted:
  - We will use the incremental testing strategy starting with the login first, first with a failed attempt and then a successful attempt. After successfully logging in, testing will begin by ensuring all links and buttons work in the home menu.
  - Testing subsequent screens after the sign on process will ensure data created is reflected throughout the application.
    - Creating a new media item should allow members to check out the new item.
    - Checkouts should create rental agreements tied to the user and to the rental item. Rental Items should no longer be able to be checked out if they are already out.
    - Overdue items should be indicated in the rental agreement. A member should not be able to check out additional items if they have a past due rental item.
    - Checking in an item should update the actual check-in date for the rental agreement. The rental item should be able to be checked out again after being returned.
    - If a staff member processes a fee payment from a member for overdue rental items, the balance should be updated on the member's profile. If the member pays all their balance off, then they should be able to checkout items again.

- Testing Items:
  - User interface
    - Testing will involve going through various webpages, such as sign in, media list, and payment screens, and ensuring the UI works. Testing will include clicking button and filling in forms.
  - Non-functional requirements
    - Passwords must follow a guideline
    - System is displayed on multiple browsers
    - Short timeout for testing
    - Show system scalability by adding users and books

Project Plan

Project planning:

Software	• Front-end: HTML,CSS,JavaScript • Back-end: Node.js • Database: MariaDB
Hardware	PC and servers
Network	Local network and an internet connection with other I.U. libraries. Ethernet cables required.

Development approach:

- HTML/CSS/JavaScript with Bootstrap framework.
- JavaScript with Node.js.
- MariaDB for the database.
- Focus on web application to be used in the web browser

Development Timeline:

Weeks	Description
1-2	Determine the framework and establish the structure of the system, connecting front-end to back-end, connecting back-end to database.
3-4	Build the system login. Register management, staff, and known members
5-6	Implement basic features for staff and management. This will include signing up members, viewing accounts, checking out members, and checking book databases. The book database will allow for adding, deleting, and reservation of books.
7-8	Test and fix known issues. Record demo for the mid-term.
9-11	Improve the system based on feedback. Add basic user features such as reserving books, browsing the catalog, and paying fines.
12-14	Write test cases and add features
15	Record and submit final presentation

References

Bootstrap (2024). GetBootstrap.com v5.3.3. Views on 17 March, 2024 at: https://getbootstrap.com/

Delamater M. & Ruvalcaba Z. (2020). Murach's JavaScript and jQuery 4th Edition. Murach.

Elmasri R. & Navathe S. (2021). Fundamentals of Database Systems 7th Edition. Pearson.

Mebjas (2023). Html5-QRCode. Viewed on 10 March 2024 at: https://github.com/mebjas/html5-qrcode#readme

Mills J. (2021). Building Web Applications with Node.js and Express. PluralSight.com. Viewed on 4 February 2024.

Tilley S. & Rosenblatt H. (2017). System Analysis and Design. 11th Edition. Cengage Learning.