Software Requirements Specification

for

BookHive

Version 2.0 approved

Prepared by Lara Lim, Amani Farid, Samy Touabi, Kian Zahrai

University of Ottawa – CSI 3140

July 23, 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Group 33	03/06/2023	Initial SRS Draft	1.0

Group 33	06/06/2023	SRS Revision & Finalization	1.1
Group 33	23/07/2023	SRS Updated	2.0

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline the software requirements for the Library Management System (LMS). This document covers the initial release of the LMS, version 2.0. The scope of this SRS encompasses the entire library management system, including all subsystems and functionality.

The Library Management System is designed to provide efficient and effective management of library operations, including cataloging, and organizing items, managing borrower information, handling check-in and check-out processes, recommending books, and facilitating overall library administration. The system aims to streamline library operations, enhance accessibility to resources, and improve the user experience for both library staff and patrons.

This SRS serves as a comprehensive guide for the development team, stakeholders, and other interested parties involved in the design, implementation, and testing of the Library Management System. It outlines the functional and non-functional requirements, system constraints, and desired features, ensuring a clear understanding of the system's objectives and expectations.

1.2 Document Conventions

The following conventions have been followed in writing this Software Requirements Specification (SRS) document:

1. **Font Styles:** The document uses standard fonts, such as Times New Roman, for the main text. Titles, headings, and subheadings may be in bold or larger font sizes to improve readability.
2. **Section Numbering:** The SRS document follows a hierarchical numbering scheme for sections, using decimal notation (e.g., 1, 1.1, 1.2, 2, 2.1, etc.), allowing for easy navigation and cross-referencing.
3. **Requirement Priorities:** Each requirement statement is accompanied by its own priority level, indicated using a numerical or alphabetical system. The priority levels reflect the relative importance or urgency of the requirement, allowing for clear identification of critical features versus those that may be optional or lower priority.

These conventions aim to enhance the clarity and organization of the SRS document, facilitating effective communication between stakeholders, developers, and other parties involved in the project.

1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) document is intended for various stakeholders involved in the development and implementation of the Library Management System (LMS). The different types of readers that this document is targeted towards include:

1. **Developers**: The SRS provides developers with a detailed understanding of the functional and non-functional requirements of the system. It outlines the scope, features, and constraints that need to be considered during the development process.
2. **Project Managers**: Project managers can utilize the SRS to gain a comprehensive overview of the project's requirements and constraints. It helps in planning, allocating resources, and tracking progress throughout the development lifecycle.
3. **Testers**: Testers can refer to the SRS to understand the expected behavior of the system. It serves as a reference to develop test cases, ensuring that the system is thoroughly tested against the specified requirements.
4. **Users**: Users, including library staff and patrons, can benefit from reading the SRS to understand the system's capabilities, features, and intended usage. It provides insights into how the LMS will enhance library operations and improve the user experience.

The SRS document is organized into several sections, as follows:

1. **Introduction**: This section provides an overview of the SRS document, including its purpose, intended audience, and the scope of the LMS. It sets the context and introduces the key aspects of the document.
2. **Overall Description**: The Overall Description section offers a comprehensive overview of the LMS. It includes information about the system's objectives, goals, and the context in which it will be used. This section may also cover high-level requirements and constraints.
3. **External Interface Requirements**: This section focuses on the external interfaces of the LMS. It describes the interactions and interfaces between the LMS and external systems, such as databases, third-party services, or other library systems. It specifies the inputs, outputs, and protocols used for communication.
4. **System Features**: The System Features section outlines the specific functional requirements of the LMS. It provides detailed descriptions of the core features and functionalities of the system. This may include cataloging items, managing borrowers, handling check-in/check-out processes, generating reports, and other relevant capabilities.
5. **Other Nonfunctional Requirements**: This section covers the non-functional requirements of the LMS. It includes quality attributes such as performance, security, usability, reliability, and other aspects that are important for the system's overall effectiveness.
6. **Other Requirements**: The Other Requirements section encompasses any additional requirements or considerations not covered in the previous sections. This may include specific constraints, preferences, system configurations, or any other relevant information that doesn't fit into the previous sections.
7. **Appendices**: The Appendix section includes supplementary information that supports the main content of the SRS document. This may include a glossary and analysis models

For an effective reading sequence, it is suggested to start with the overview sections (Introduction and System Overview) to gain a holistic understanding of the LMS. Developers and testers can then focus on the Functional Requirements section, while project managers may also refer to the Non-Functional Requirements and System Constraints sections. Users can benefit from reading the entire document to gain a comprehensive understanding of the system's requirements and capabilities.

1.4 Product Scope

The Library Management System (LMS) is software designed to automate and improve library operations. It simplifies resource management, enhances borrowing processes, enables accurate inventory tracking, provides insightful recommendations, and enhances the overall user experience. By implementing the LMS, libraries can achieve cost savings, improve user satisfaction, and make data-driven decisions. The SRS document specifies the software requirements to develop and implement the LMS, ensuring it aligns with the desired objectives and benefits.

1.5 References

IEEE Standards Association (2011, Last Accessed June 2017) Systems and software engineering – Life cycle processes –Requirements engineering, Standard. (IEEE 29148-2011) - http://standards.ieee.org/findstds/standard/29148-2011.html.

World Wide Web Consortium (2011, Last Accessed June 2017) REST, Web Page. - http://www.w3.org/2001/sw/wiki/REST.

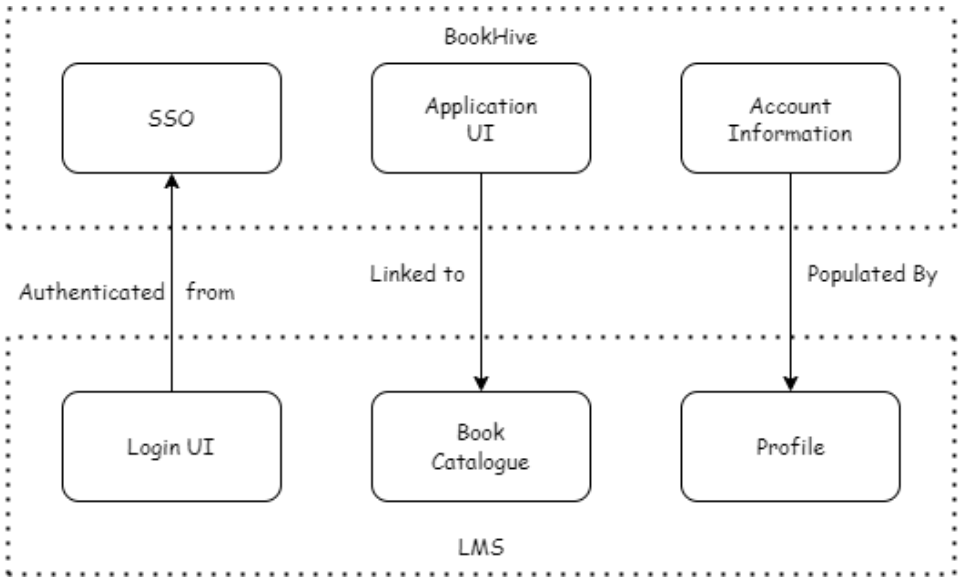
2. Overall Description

2.1 Product Perspective

BookHive is a new, self-contained library management system designed to streamline and enhance the management of library resources, services, and operations. It will function as an independent software solution, dedicated to efficiently handling various library-related tasks such as cataloging, inventory, and reporting. It aims to provide administrators and library staff with an intuitive and comprehensive tool to automate and simplify their daily workflows, ultimately improving user experiences and optimizing library operations. Although a standalone system, it can integrate with existing library databases and external cataloging services. The LMS will ensure seamless data exchange and interoperability with these external systems to facilitate efficient information retrieval, resource sharing, and collaborative services.

To visually illustrate the product perspective, a diagram depicting the major components of the overall system, their interconnections, and external interfaces will be included in subsequent sections of this SRS. To start, the following diagram will provide a clear overview of the LMS and its relationship to other systems within the library environment, aiding in the understanding of the software's functional context and its integration points.

Product Perspective of BookHive LMS



2.2 Product Functions

The BookHive Library Management System performs the following major functions:

- Account Management:
  - Register new users and maintain user information.
  - Authenticate users during login.
  - Manage user roles and permissions.
  - Manage system configurations and settings.
  - Configure account information.

- Inventory Management:
  - Add new items to the inventory with details such as title, author, ISBN, language, and publisher.
  - Search and retrieve item information based on title, author, or ISBN.
  - Update, edit, or delete existing items from the inventory.
  - Track item availability and status (e.g., checked out, available, on hold).
  - Manage due dates, renewals, and fines for late returns.

- Item Checkout:
  - Allow users to borrow items from the library.
  - Record and track borrowing history.
  - Process item returns and update availability in the inventory.

- Book Recommendation:

- Provide personalized book recommendations to users based on their preferences defined during the account creation process.
- Enhance the user experience by assisting users in discovering new books of interest within the library's catalog.

Appendix B provides visual representations of how the product functions interact within the system, and the users, portraying the flow of data throughout the system.

2.3 User Classes and Characteristics

The BookHive Library Management System is designed to cater to the following user classes with their respective characteristics:

- Librarian:
  - Responsible for overseeing the overall functioning of the library.
  - Full access to all system features and administrative functions.
  - Monitor system performance and handle exceptions or errors.
  - Highly familiar with library operations and procedures.
  - Responsible for managing the library's collection, assisting users, and maintaining inventory records.
  - Access to administrative functions for system configuration, user management, and monitoring system performance.

- Library Members:
  - Individuals who use the library services.
  - Access to borrowing, reservation, and search functionalities.
  - Can view their borrowing history, renew items, and receive notifications.

2.4 Operating Environment

The server-side software components will be run on a windows system, and the database will be hosted on a cloud platform. The client-side software components will be accessible from common browsers including Google Chrome (version 70+), Mozilla FireFox (version 60+), Apple Safari (version 11+), and Microsoft Edge (version 41+).

2.5 Design and Implementation Constraints

The system will use a relational database schema. It must be scalable to accommodate hundreds of users and new library branches. The development will take into consideration the variety of devices that may be used to access the system; advanced computation and graphics-intensive features will not be implemented. The system will use HTTPS to be security compliant. Timing constraints are associated with the project phases:

Phase 1: June 6, 2023 **Phase 3: July 23, 2023**
Phase 2: July 3, 2023 **Phase 4: July 22, 2023**

Customers (library branch staff) will be responsible for installing, configuring, and maintaining the software, as well as having at least one dedicated system administrator.

2.6 User Documentation

User documentation for the Library Management System consists of multiple parts available to the user classes of library staff and administrators. A README document is provided as an overview for software installation with the repository. A demonstration video will be released as Phase 4 of development and will showcase the system's functionality.

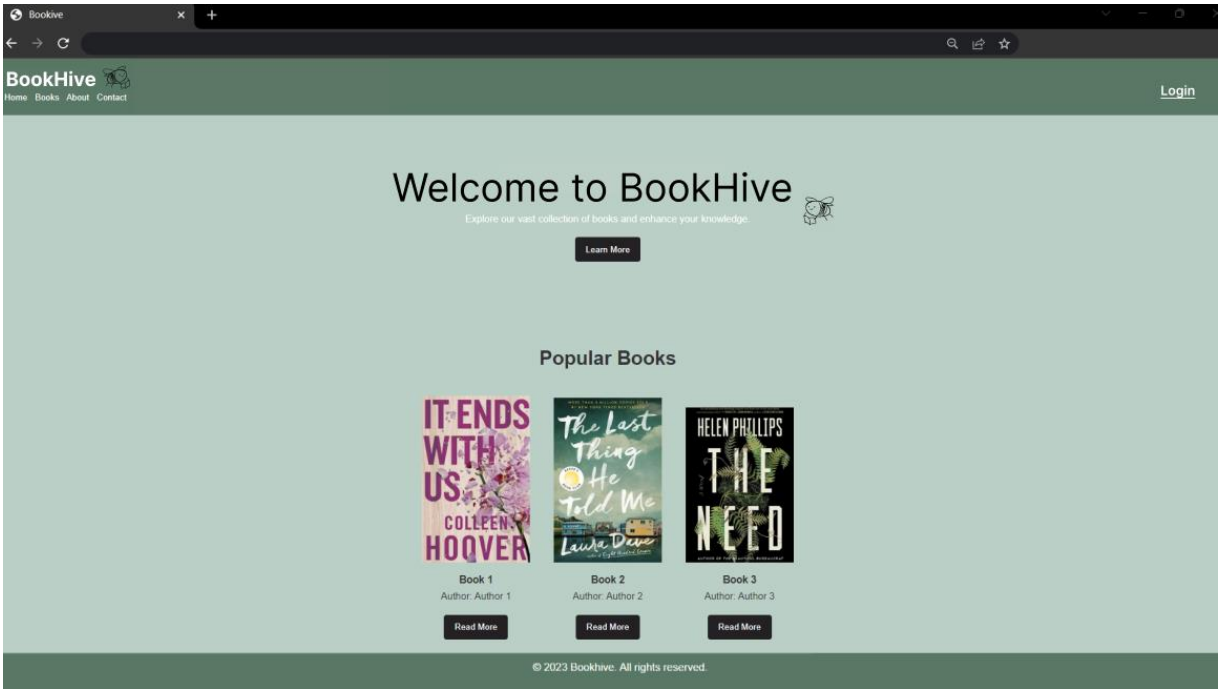
2.7 Assumptions and Dependencies

Assumed factors include the means of accessing the system (Section 2.4). It is also assumed that users of the system will be connected to the internet. Software component dependencies are related to the database and communications protocols (Section 2.5). The system is also dependent on third party libraries used for the system framework, as well as supporting the book recommendation feature.

3. External Interface Requirements

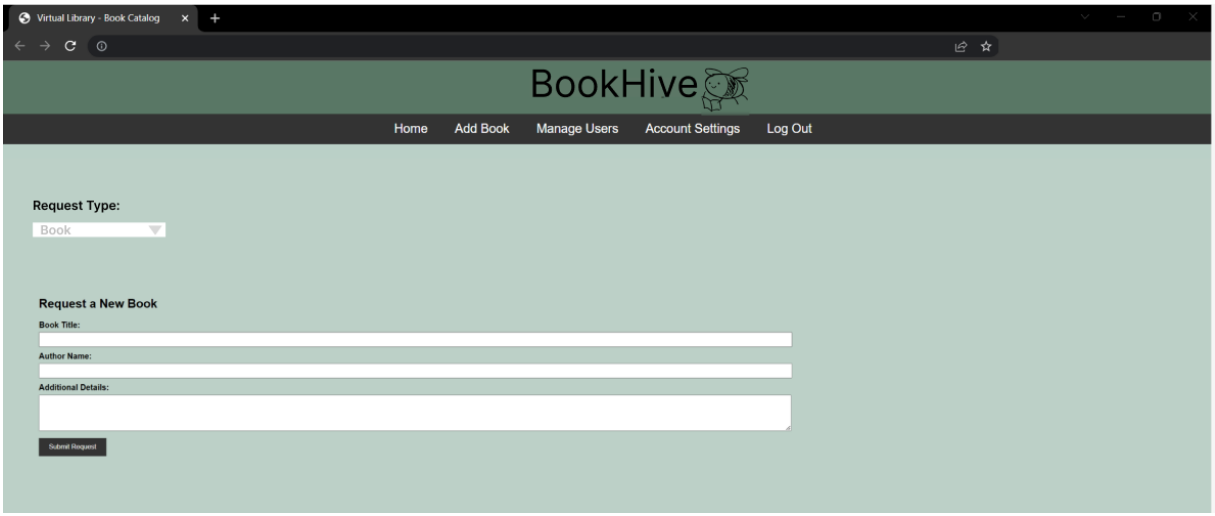
3.1 User Interfaces

The logical characteristics of the following interfaces adhere to GUI standards and design principles, aiming for a consistent and intuitive user experience. The layout of the screens is visually appealing and features key components such as the header, footer, content area, forms and carousels. The header includes varying sections and navigation options according to the account type of the user. Each page will adhere to the Arial font-family and will implement any generic sans-serif font as a fallback. The website pages utilize a pleasing color palette comprising of #507963 (dark green), #b6d0c7 (pale blue-green), #fff (white), and #333 (dark gray) to create a captivating atmosphere and evoke a sense of sophistication. Each page incorporates intuitive tooltips to provide assistance with navigation and enhance user interaction.



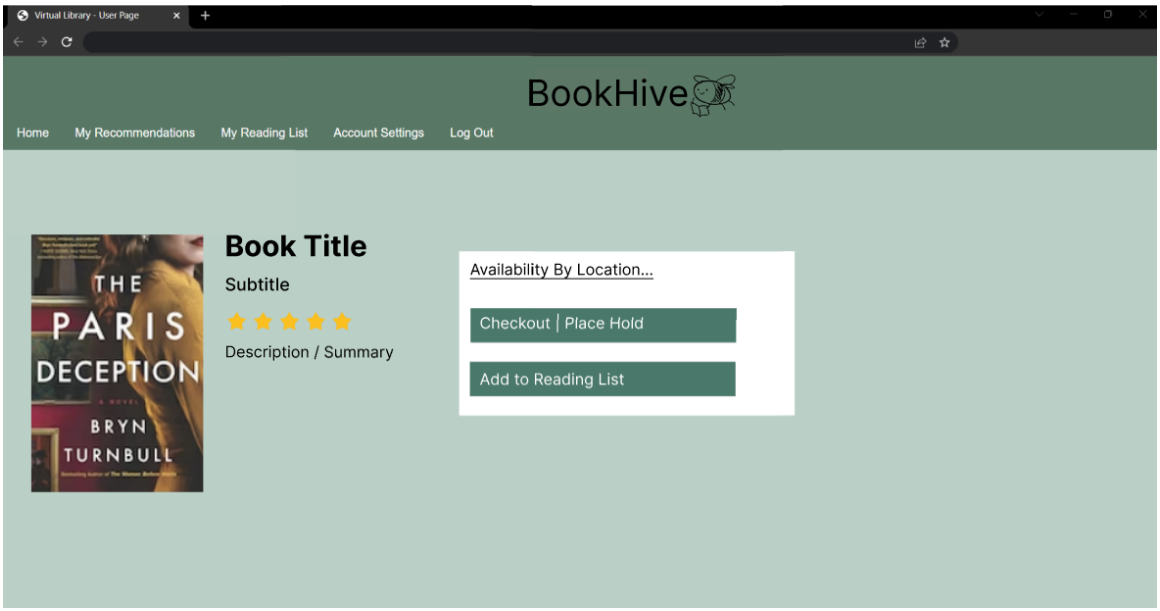
Screen 1: Guest Landing Page

The Visitor Page, also known as the Guest Landing Page, serves as the initial interface for visitors who have not logged in or created an account. The header displays the name of the system and its logo. It features a button for users to learn more about the system and encourages visitors to create an account. The Visitor Page also includes a list of popular books that the library has to offer. The footer section includes supplementary information such as copyright notices, links to terms of service or privacy policy pages, and contact details. Overall, the Visitor Page provides an engaging and informative entry point to the library's services while encouraging visitor registration.



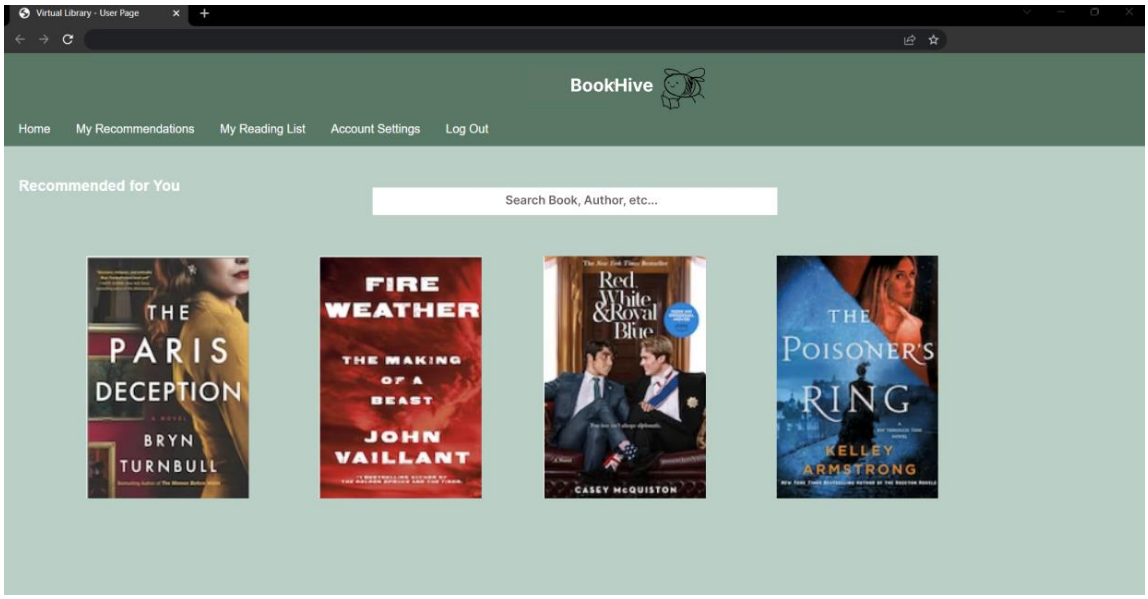
Screen 2: Book Request Page

The layout of the Book Recommendation page can be accessed once a library member logs into their account and visits the My Recommendations tab. It is designed to showcase recommended books in a visually appealing manner, with relevant information such as book covers, titles, authors, and brief descriptions. The interface provides options for users to explore additional details about recommended books, such as ratings, reviews, and availability. Users can also interact with the interface to provide feedback on the recommendations or refine their preferences to receive more accurate suggestions in the future. The book covers are displayed in a revolving carousel.



Screen 3: Book Checkout Page

The Book Checkout Page facilitates the smooth process of borrowing items from the library. The simple interface helps guide library members through the checkout process. When a user has selected an item to checkout, the page will display its title, author, and summary. Clear and prominent buttons are provided to confirm the checkout. If the item is unavailable, the button will display Place Hold.



Screen 4: Book Catalog Page

The Book Catalog Page serves as a dedicated interface for librarians to request new items to be added to the library's collection. The layout of the page is displayed as a form to allow the input of book details including the book title, author, and any additional relevant details or comments. Clear and prominent buttons are included to submit the request, as well as options to cancel or modify the request if required. The interface may also offer features such the ability to attach supporting documents or book cover images for the request.

3.2 Hardware Interfaces

Server-side components, responsible for handling data processing, storage, and retrieval, are exclusively executed on server-class computers which will provide the providing the necessary computational power, memory, and storage to process multiple client requests. Meanwhile, client-side components for user interaction are executed on workstation-class and personal-class computers.

3.3 Software Interfaces

The system implements several software components to enable its functionality. It leverages the PostgreSQL database system version 13.0, for efficient data storage and retrieval. The product utilizes SQL to perform database operations, such as data insertion, querying, updating, and deletion. The system is designed to run on the Windows 11 platform, macOS Monterey, and Linux Ubuntu 22.04. The system may implement NodeJS version 14.16 and ReactJS version 18.0 to facilitate web development and use jQuery version 3.6 and Bootstrap for client-side scripting and responsive design.

3.4 Communications Interfaces

The communication architecture follows the client-server model. The system shall utilize a REST-compliant web service and must be served by HTTPS. The product requires robust communication capabilities to support various functions. Firstly, it should integrate with email

systems to enable the sending and receiving of emails for user notifications, account management, and user communication. This requires adherence to standard email protocols such as SMTP for outgoing emails. The product involves FTP for file uploads and downloads.

4. System Features

For the following subsections, a user is intended to represent an individual with an account to use the functionalities and features of the library management system. This includes the administrators, library staff and library members.

4.1 Item Checkout

4.1.1 Description and Priority

Item Checkout allows library patrons to check out items from the library's collection. This feature enables a smooth and efficient borrowing process for end users, allowing them to select the desired items and complete the checkout procedure. The priority of this feature is High, as it is a core functionality of the library management system.

4.1.2 Stimulus/Response Sequences

The following sequences outline the user actions and system responses associated with the Item Checkout feature:

Stimulus: User selects the desired items for checkout, confirming their selection to then initiate the checkout process.
Response: System verifies the availability of the selected items, then updates the items' status as "checked out" and assigns a due date. A confirmation message is displayed to the user.

Stimulus: User selects the desired items for checkout, confirming their selection to then initiate the checkout process.
Response: System detects one or more items have no current availability. It prompts the user with an associated message, to then make alternative selections or modify their checkout request.

4.1.3 Functional Requirements

IC-1: The system shall allow users to search for items by the following criteria: title, author, and ISBN, to facilitate item selection during the checkout process.
IC-2: The system shall display the availability status of each item in real-time, indicating whether it is available for checkout or already checked out.
IC-3: The system shall enforce a checkout limit of 4 items at once per user.
IC-4: The system shall calculate and assign a due date for each checked-out item based on the library's borrowing policies.

4.2 Account Management

4.2.1 Description and Priority

The Account Management feature facilitates the creation, removal, and updates of library member and library staff accounts. Library members may perform these actions for their own accounts, and administrators may perform these actions for library staff accounts.

4.2.2 Stimulus/Response Sequences

The following sequences outline the user actions and system responses associated with the account management functionalities:

Stimulus: User requests to create a new account.
Response: System provides a form to enter account information.

Stimulus: User enters account information including name, email, and contact information, confirming the account creation.
Response: System verifies that this user's account does not already exist and creates the account.

Stimulus: User enters account information, including an email that already has an account associated to it.
Response: System detects that an account for this email already exists in the database and displays a message about this.

Stimulus: User requests to edit their personal information.
Response: System provides a form to edit personal information.

Stimulus: User makes the changes and requests to save them.
Response: System verifies that all information is in the intended format and saves the changes.

Stimulus: The user makes changes including at least one format issue such as phone number length or invalid email address, and requests to save them.
Response: System detects the format issue(s) and generates an error message indicating the fields that have issues.

Stimulus: Administrator requests to delete an account.
Response: System displays a warning message about the change being irreversible.

Stimulus: Administrator confirms the decision to delete an account.
Response: System removes the target account from the database.

4.2.3 Functional Requirements

AM-1: The system shall allow library members to edit their account information.

AM-2: The system shall allow librarians to view checked out items for client accounts.
AM-3 The system shall allow librarians to edit their personal information.
AM-4: The system shall allow librarians to create library member accounts.
AM-5: The system shall allow librarians to remove library member accounts.
AM-6: The system shall allow librarians to edit the personal information for library member accounts.
AM-7: The system shall allow librarians to edit the personal information for library member accounts.
AM-8: The system shall allow library members to modify their preferences for book recommendations.
AM-9: The system shall allow librarians to add books to the reading list of library members.
AM-10: The system shall include a password reset feature using account emails.
AM-11: If the user selects the password reset prompt, the system shall send an email to that user with instructions to facilitate the password reset process.

4.3 Book Recommendation Feature

4.3.1 Description and Priority

The book recommendation feature provides personalized book recommendations to users based on their preferences that are defined by the user during the creation of their account. It will enhance the user experience by assisting users to discover new books of interest within the book catalog.

4.3.2 Stimulus/Response Sequences

The following sequences outline the user actions and system responses associated with the book recommendation feature:

Stimulus: User navigates to the “Recommended for You” section.
Response: System generates personalized book recommendations based on the user’s preferences and displays the list of books tailored to the user’s interests.

Stimulus: User clicks on a recommended book.
Response: System displays details on the book, including its author, description/synopsis, cover image and ratings.

Stimulus: User clicks the checkout button for a book.
System: System brings the user to the checkout page where they can follow the steps to borrow the book.

Stimulus: User marks a book as “Not Interested”.

Response: System requests feedback from the user and adjusts future recommendations accordingly.

4.3.3 Functional Requirements

RE-1: The system shall allow library members to specify their book preferences during account registration.
RE-2: The system shall store library member preferences.
RE-2: The system shall allow library members to enter keywords related to their preferences.
RE-3: The system shall allow library members to specify topics of interest.
RE-4: The system shall allow library members to specify authors of interest.
RE-5: The system shall match keywords in the recommendation algorithm implemented to generate personalized recommendations.
RE-5: The system shall display a list of book recommendations to library members based on their preferences.
RE-6: The system shall allow users to provide feedback on its recommendations.
RE-7: The system shall employ a hybrid recommendation approach, combining collaborative filtering and content-based filtering techniques, to provide more accurate and diverse recommendations.

4.4 Inventory Management Feature

4.4.1 Description and Priority

The item inventory management feature aims to ensure efficient and accurate management of library resources, streamline item retrieval, enable data-driven decision-making, and enhance the overall organization and accessibility of the library's inventory.

4.4.2 Stimulus/Response Sequences

The following sequences outline the user actions and system responses associated with the inventory management functionalities:

Stimulus: User selects "Add New Item" option in the inventory management interface.
Response: System displays a form for entering item details such as title, author, ISBN, language, and publisher.

Stimulus: User searches for an item by title or ISBN in the inventory management interface.
Response: System retrieves and displays a list of matching items with their respective details.

Stimulus: User selects an item from the inventory list and clicks on "Edit" button.
Response: System opens a form pre-filled with the selected item's details, allowing the user to modify and save the changes.

Stimulus: User selects an item from the inventory list and clicks on "Delete" button.
Response: System prompts the user to confirm the deletion and, upon confirmation, removes the item from the inventory.

Stimulus: User applies a filter or sorting option in the inventory management interface.
Response: System updates the displayed inventory list based on the selected filter or sorting criteria.

Stimulus: User selects an item and clicks on the "View Details" button in the inventory management interface.
Response: System displays a detailed view of the selected item, including its description, availability status, and borrowing history.

4.4.3 Functional Requirements

IM-1: The system shall allow library members to browse the book catalogues of the library.
IM-2: The system shall allow librarians to track and manage the inventory of library items.
IM-3: The system shall automatically store account details of library members.
IM-4: The system shall support the classification and categorization of items based on the predefined scheme.
IM-5: The system shall display relevant information based on item search criteria.
IM-6: The system shall enable librarians to update the status of books.
IM-7: The system shall support the management of holds and reservations for items that are currently checked out.
IM-8: The system shall facilitate the withdrawal and disposal of items from the library's collection.
IM-9: The system shall notify library members if they have an overdue item.

5. Nonfunctional Requirements

5.1 Performance Requirements

PR-1: The system shall return query searches by users in under 5 seconds.
PR-2: The system shall change the availability status of items in under 10 seconds.
PR-3: In case of peak load, the system shall have a minimum transaction rate of 60 book registrations per 5 seconds.
PR-4: The system shall produce summary reports in less than 20 seconds for at least 95% of the cases.

5.2 Safety Requirements

SAFE-1: The system's defect rate shall be less than 1 error per 200 hours of operation.
SAFE-2: The system shall not be unavailable for more than 1 hour per 200 hours of operation.
SAFE-3: The system's code shall have a cyclomatic complexity of no more than 6.
SAFE-4: The system shall conduct data backups once every night upon downtime.

5.3 Security Requirements

SEC-1: The system shall detect at least 99% of intrusion attempts within 5 seconds.
SEC-2: The system shall require user authentication to access and perform actions.
SEC-3: The system shall allow librarians to implement role-based access control for defining different user roles.
SEC-4: The system shall maintain an audit trail of inventory-management activities.
SEC-5: The system shall log access authorization attempts upon every sign on.

5.4 Software Quality Attributes

SQ-1: The system shall have a page load time of less than 3 seconds for 95% of user requests.
SQ-2: The system shall have an average response time of less than 1 second for loading a book catalog page shall be less than 1 second.
SQ-3: The system shall enable novice library members to check out an item in less than 5 minutes.
SQ-4: The system shall enable novice library staff to request a new item in less than 10 minutes.
SQ-5: The system shall handle concurrent user requests without significant performance degradation or downtime.
SQ-6: The system shall be compatible with commonly used web browsers, such as Chrome, Firefox, Safari, and Edge.
SQ-7: The system shall include a version control system to track and manage code changes effectively.

5.5 Business Rules

BR-1: Users must supply complete information for their account.
BR-2: Librarians must supply complete information for all items in the inventory.
BR-3: Library members may not check out more than 4 items at once.

6. Other Requirements

6.1 Legal Requirements

LE-1: Third party software included in the system will be used in accordance with copyright laws and license agreements.
LE-2: The library inventory will only include books published through accredited methods.
LE-3: Software for the Book Recommendation feature will operate on data sets independent of the personal information of users.

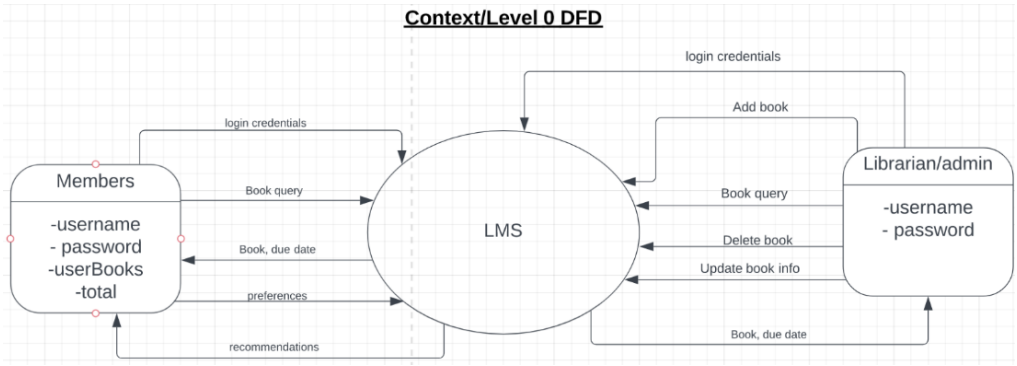
Appendix A: Glossary

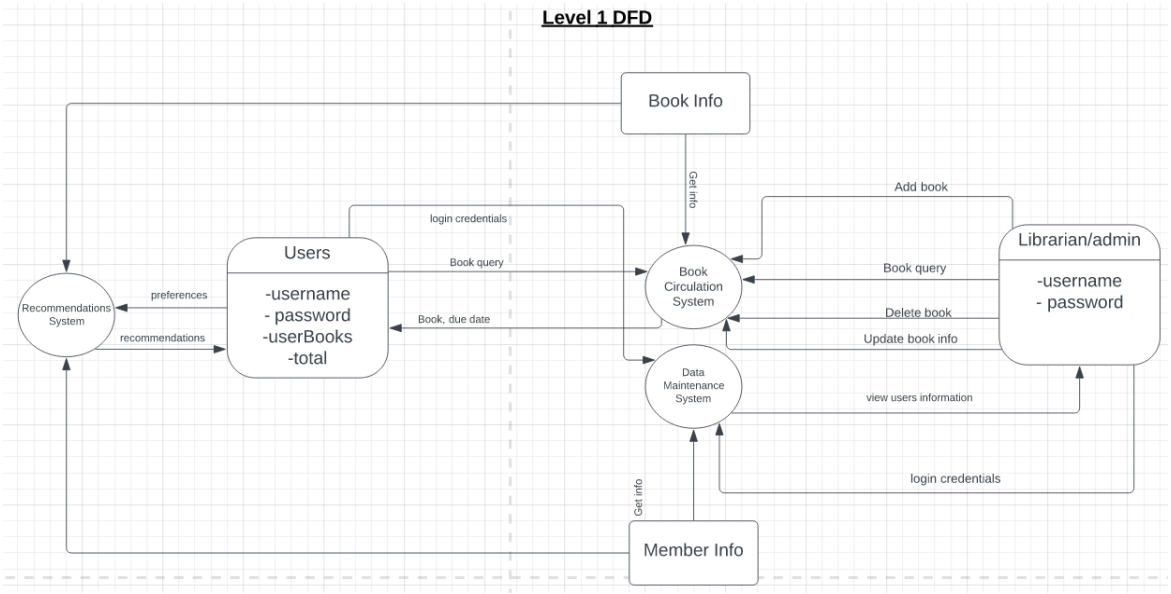
Requirement Identifiers:
AM – Account Management
BR – Business Rules
IC – Item Checkout
IM – Inventory Management
LE – Legal Requirements
PR – Performance Requirements
RE – Book Recommendations
SAFE – Safety Requirements
SEC – Security Requirements

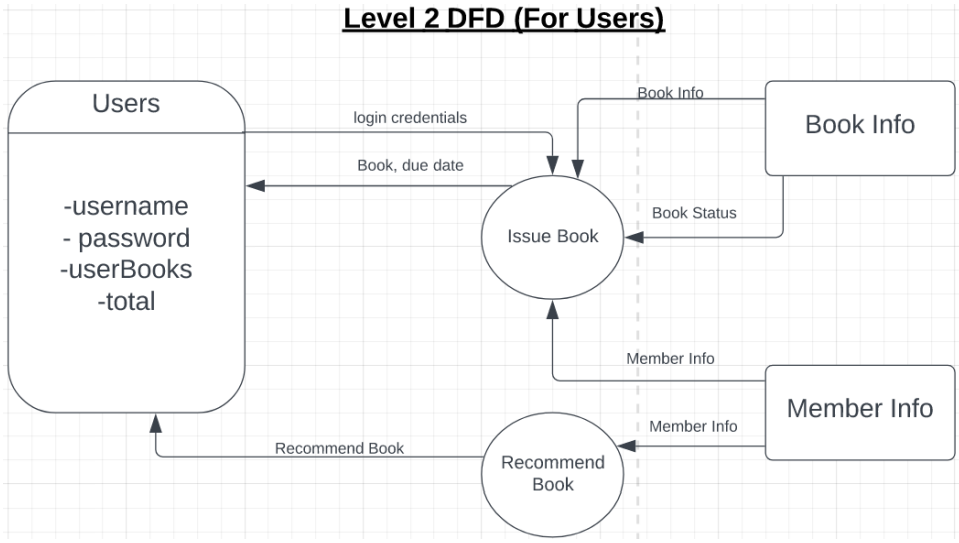
General:
FTP – File Transfer Protocol
HTTPS – Hypertext Transfer Protocol Secure
LMS – Library Management System
SMTP – Simple Mail Transfer Protocol
SRS – Software Requirements Specification

Appendix B: Analysis Models

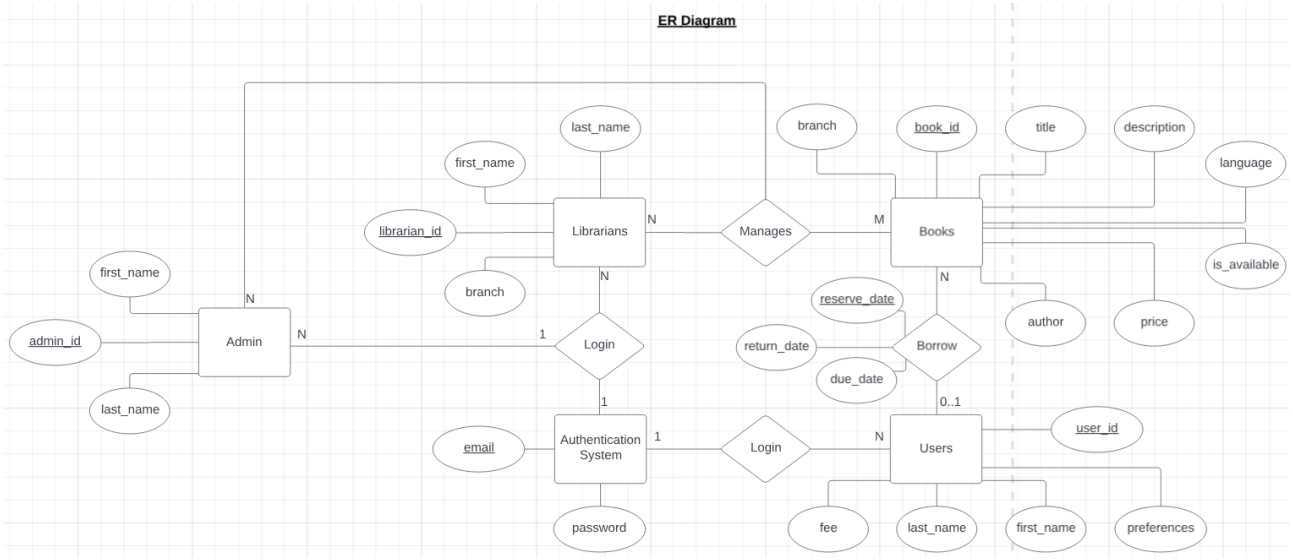
DataFlow Diagrams:



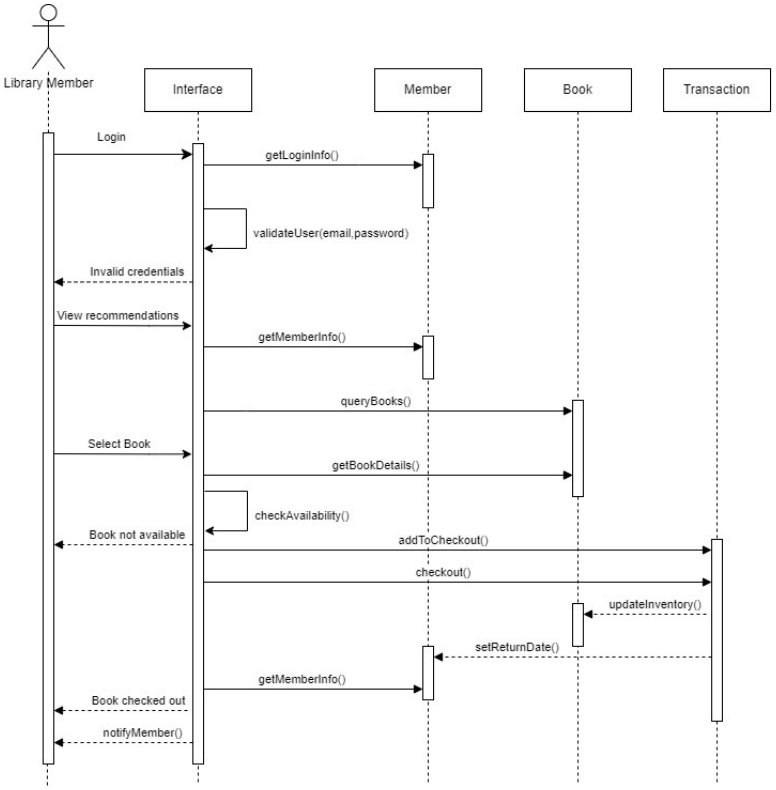


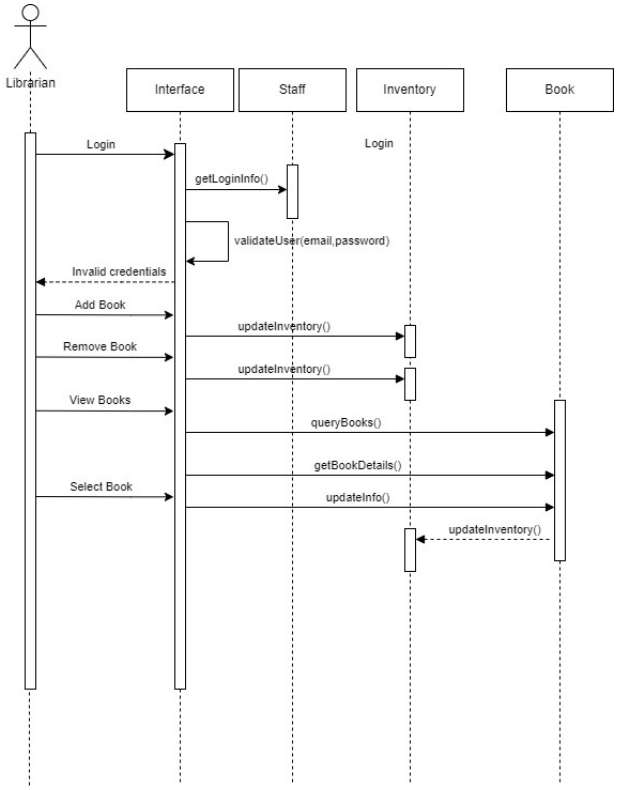


Entity-Relationship Diagram:

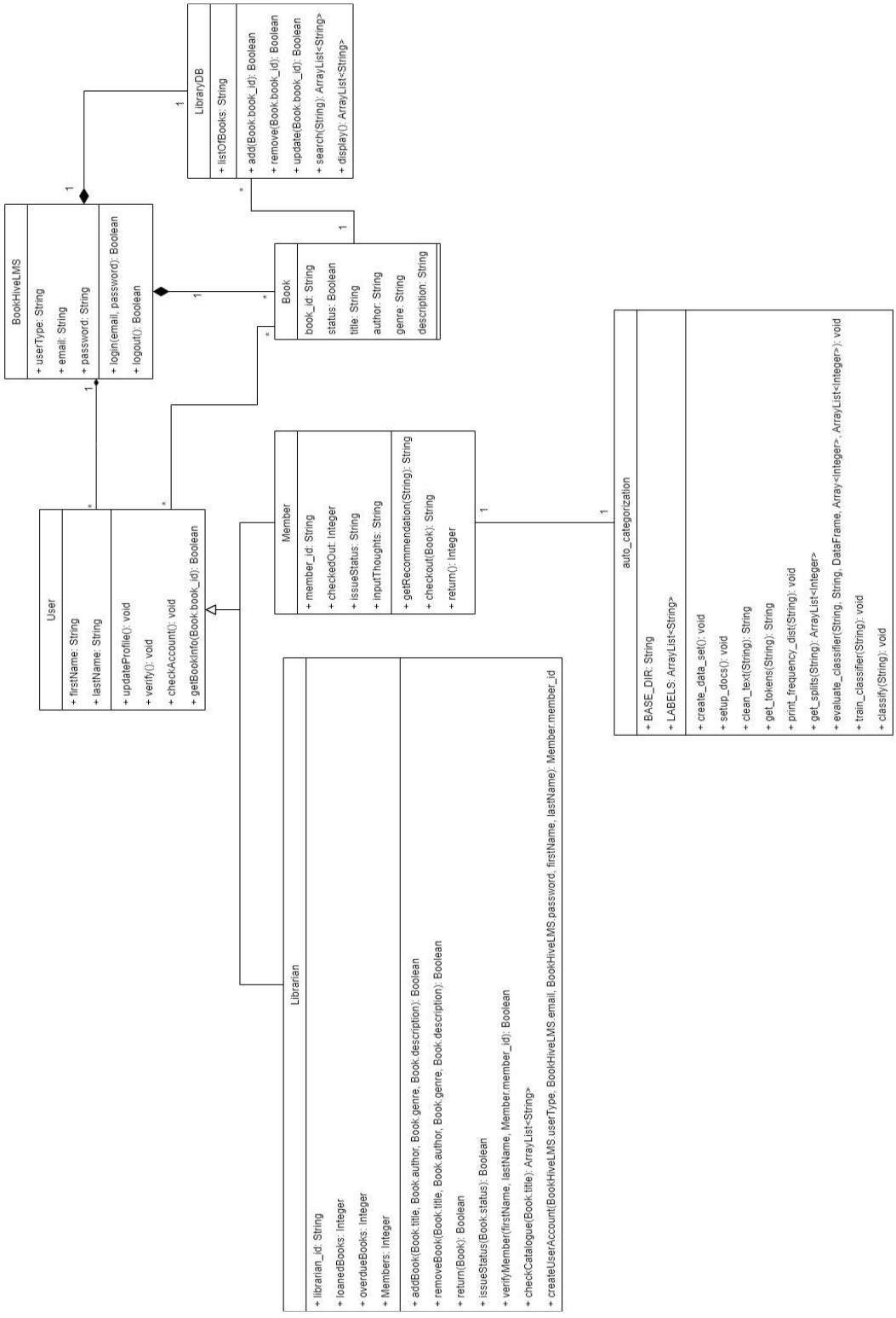


Sequence Diagram:





Class Diagram:


BookHiveLMS
+ userType: String
+ email: String
+ password: String
+ login(email, password): Boolean
+ logout(): Boolean
LibraryDB
+ listOfBooks: String
+ add(Book.book_id): Boolean
+ remove(Book.book_id): Boolean
+ update(Book.book_id): Boolean
+ search(String): ArrayList<String>
+ display(): ArrayList<String>
Book
book_id: String
status: Boolean
title: String
author: String
genre: String
description: String
User
+ firstName: String
+ lastName: String
+ updateProfile(): void
+ verify(): void
+ checkAccount(): void
+ getBookInfo(Book.book_id): Boolean
Member
+ member_id: String
+ checkedOut: Integer
+ issueStatus: String
+ inputThoughts: String
+ getRecommendation(String): String
+ checkout(Book): String
+ return(): Integer
auto_categorization
+ BASE_DIR: String
+ LABELS: ArrayList<String>
+ create_data_set(): void
+ setup_docs(): void
+ clean_text(String): String
+ get_tokens(String): String
+ print_frequency_dist(String): void
+ get_splits(String): ArrayList<Integer>
+ evaluate_classifier(String, String, DataFrame, Array<Integer>, ArrayList<Integer>): void
+ train_classifier(String): void
+ classify(String): void
Librarian
+ librarian_id: String
+ loanedBooks: Integer
+ overdueBooks: Integer
+ Members: Integer
+ addBook(Book.title, Book.author, Book.genre, Book.description): Boolean
+ removeBook(Book.title, Book.author, Book.genre, Book.description): Boolean
+ return(Book): Boolean
+ issueStatus(Book.status): Boolean
+ verifyMember(firstName, lastName, Member.member_id): Boolean
+ checkCatalogue(Book.title): ArrayList<String>
+ createUserAccount(BookHiveLMS.userType, BookHiveLMS.email, BookHiveLMS.password, firstName, lastName): Member.member_id

Activity Diagram: