CS 3354 Software Engineering
Final Project Deliverable 1

Bound 2 Be

Group 5: Charles Bacani, Victoria DeLozier, Peter Hoang,
Kim Le, Daniel Liu, Kim-Nhi Ngo, and Pranay Yadav

1. Addressing Proposal Feedback

Final Project Draft Description:

Group 5
Project Title
Bound 2 Be
Group Members
Charles Bacani, Victoria DeLozier, Peter Hoang, Kim Le, Daniel Liu, Kim-Nhi Ngo, and Pranay Yadav
Project Description
A book matching software where users can enter their interests and preferences to receive book recommendations and find places to purchase those books.
Motivation
We want to make it feasible to find books that match people's interests, fostering a love for reading and learning that will last a lifetime. This software could be used by book subscription services so they could provide books to their users that are catered to their individual preferences. Libraries and bookstores could also use it if someone asks for a book recommendation. The software can also be used by individuals who read for fun.
Tasks

- Charles Bacani
  - Management of project costs and pricing
  - Make an evaluation of work to conclude project
- Victoria DeLozier
  - Responsible for submitting each stage of the project to eLearning
  - Edit and put together presentation video
- Peter Hoang
  - Listing software requirements
  - Project scheduling
- Kim Le
  - Defining project scope and mapping out the general structure of the project.
- Daniel Liu
  - Create diagrams (case diagram, sequence diagram, class diagram).
  - Architectural design of the project.
  - Create the test plan.
- Kim-Nhi Ngo
  - Set up and manage Github repository
  - Create presentation on Canva
- Pranay Yadav
  - Help with creating diagrams
  - Comparison of similar designs

Presentation preference
Option 2: Video presentation with captions
Interested in scholarly paper?
We would not want to write a paper on this project.

Instructor Feedback:

> “A practical idea of a tool that promises a lot of potential use. In the final report, please make sure to include comparison with similar applications -if any-, make sure that you differentiate your design from those, and explicitly specify how. Fair delegation of tasks. Please share this feedback with your group members. You are good to go. Have fun with the project and hope everyone enjoys the collaboration."

Response:

The main component of the feedback that we received was that we need to be sure to include any similar applications and how Bound2Be is different in the final report. We have determined that there are a fair number of similar services, the main ones being GoodReads and Tailored Book Recommendations (TBR). When a user sets up an account with GoodReads, they are asked for their preferences but the questionnaire doesn’t go very in-depth, and as far as we could tell, is not dynamic as the user’s preferences change. TBR is a subscription-based service, providing 3 recommendations to the users every 3 months. When designing Bound2Be, we made sure to implement both a dynamic recommendation algorithm and an in-depth preferences questionnaire.

2. GitHub Repository

https://github.com/pranay-yadav/3354-Bound2Be

3. Delegation of Tasks

- Charles Bacani: Created functional requirements list
- Victoria DeLozier: Pushed ReadMe, wrote feedback to project proposal, created Architectural Design diagram
- Peter Hoang: Created nonfunctional requirements list
- Kim Le: Created sequence diagrams for use cases
- Daniel Liu: Created use case diagram for the system
- Kim-Nhi Ngo: Pushed project_scope document to GitHub repository, helped with sequence diagrams
- Pranay Yadav: Initialized GitHub repository, developed Class Diagram

4. Software Process Model

The project employs the waterfall process model in the development process, primarily because the project is heavily plan-driven and encourages documentation. Our team meets regularly to plan, delegate, and schedule our tasks and activities before starting development. In addition, by the end of each phase of the project, documents are produced and approved by a passing grade being given and prior documents are modified accordingly based on changes within our project. Much like how problems from previous stages become evident in future stages in the software development process when using a waterfall process model, our team has discovered a few new issues as the project develops and changes must be made.

5. Software Requirements

Functional

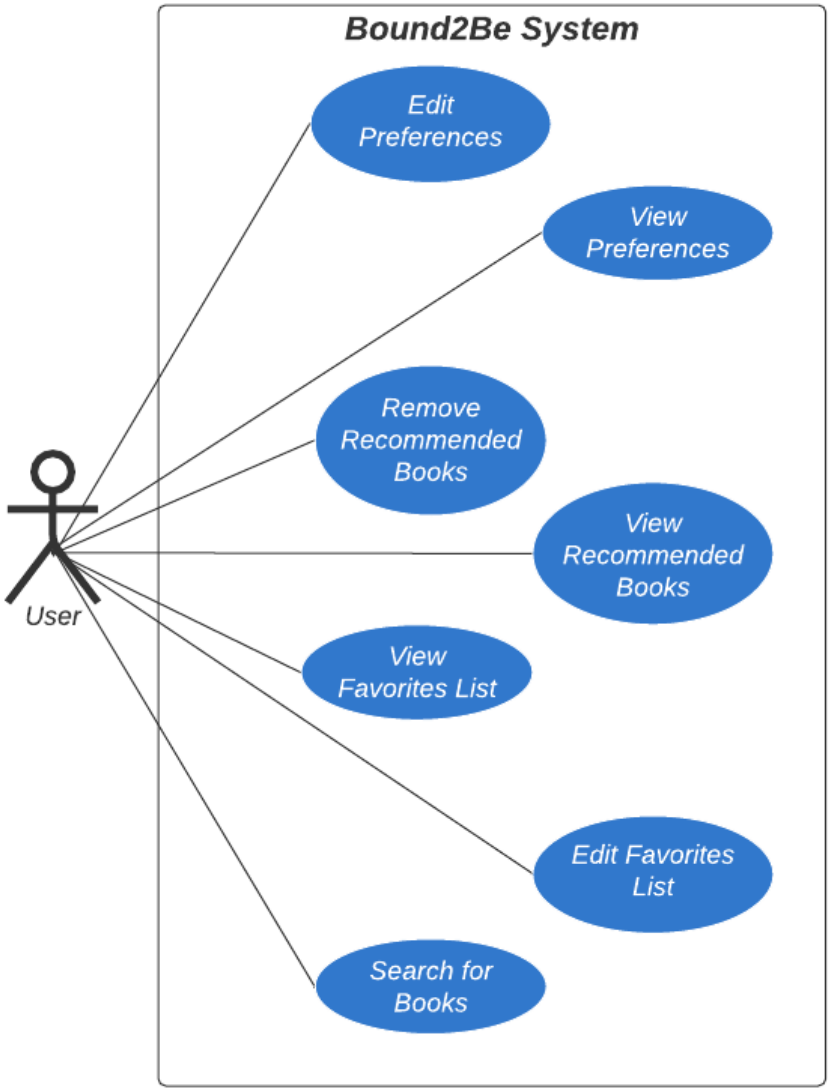
1. A user shall be able to easily find the links that are associated with the books they want to read more about and/or potentially purchase.
2. The system shall generate each login, for each user, a list of recommended books that match the preferences of the users if they are not already check marked as read.
3. Each user using the system shall be uniquely identified by their username and userID.
4. The system must allow users to register and log in into their accounts by entering their email and password.
5. The system must allow users to change their preferences at any time and update accordingly while also allowing users to change their list of favorite books.
6. The system must allow users to delete their accounts and associated information with ease.

Non-Functional

1. **Usability requirement:** 80% of users trying the application for the first time should be able to find at least 5 book recommendations within 5 minutes of software first time setup.
2. **Performance requirement:** The application should ensure that recommendations based on user input are matched, fetched from the database, and displayed within 1 second of system login.
3. **Space requirement:** The software will require no more than 5GB of storage on desktop systems to operate, including after subsequent updates.
4. **Dependability requirement:** The software will be down, unavailable, or under maintenance for less than 2% of operating and online time.
5. **Security requirement**: The system will support cloud-based authentication of user login credentials.

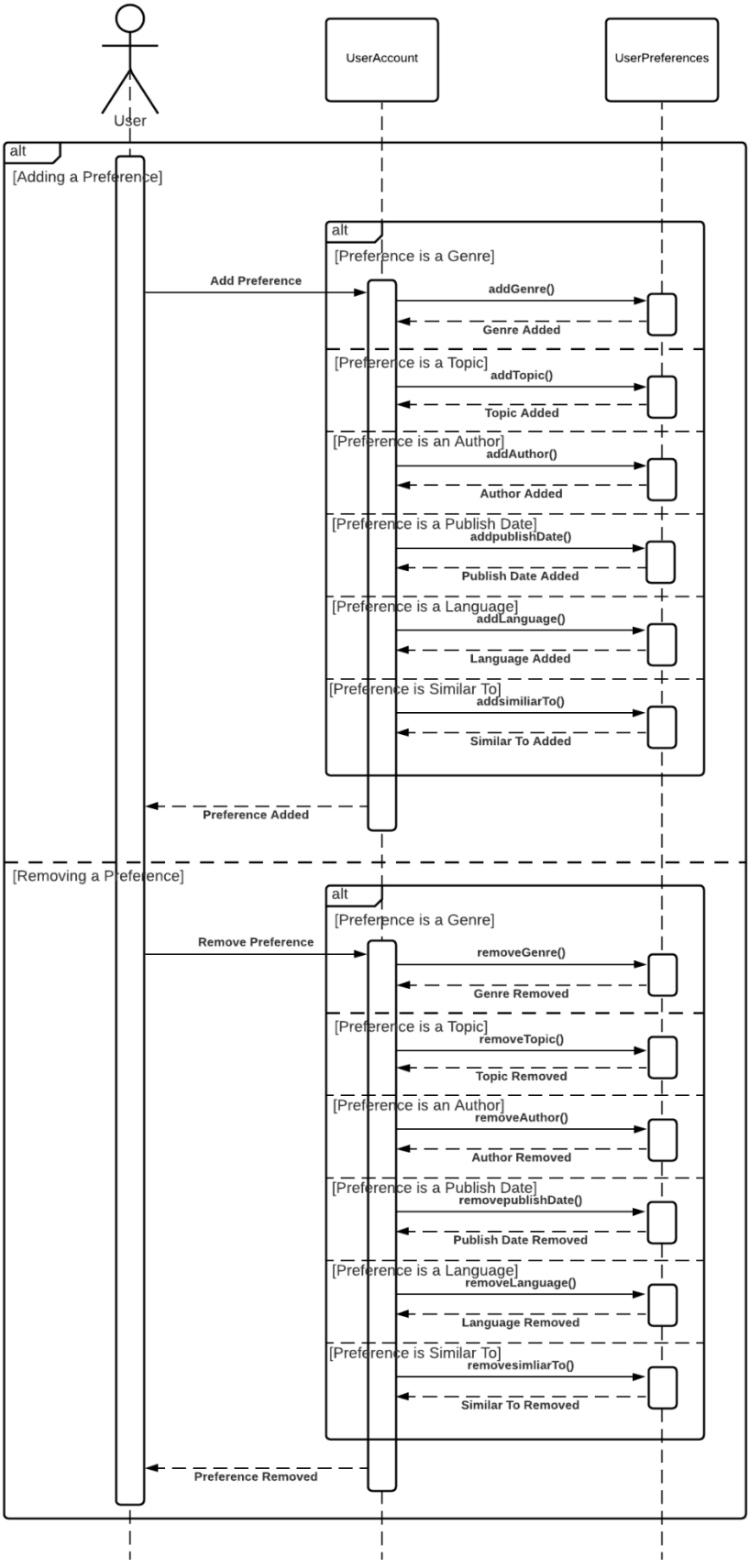
6. **Environmental requirement**: The software will run in various system environments, including macOS X or greater, Windows 8 or greater, and Linux and Unix based systems, as well as mobile platforms including iOS 12 or greater and Android 10 or greater.
7. **Operational requirement**: The software will be used by either individuals or libraries to recommend books based on given input and will be available as a stand-alone application for online download.
8. **Development requirement**: The software will be developed using JavaScript, and the MySQL framework will be used for interfacing with the database. The user interface will be developed using React.js and mobile implementations can be developed with React Native.
9. **Accounting requirement:** The system will track revenue generated through user purchases through referral links for books and/or advertisements and provide a monthly statement of expenses on the last Saturday of each month for accounting purposes as required by law.
10. **Ethical requirement:** The system will only use user data for book recommendations and diagnostic purposes. User data will not be sold or distributed and will be kept entirely confidential.
11. **Safety/security requirement:** The system will support database encryption and will not provide administrative accounts with more access than needed to prevent data leaks and security compromises.
12. **Regulatory requirement:** The software will adhere to a privacy policy informing users of the storage and collection of their account details and usage data in compliance with state data privacy laws such as the California Consumer Privacy Act (CCPA) and California Privacy Rights Act (CPRA).

6. Use Case Diagram

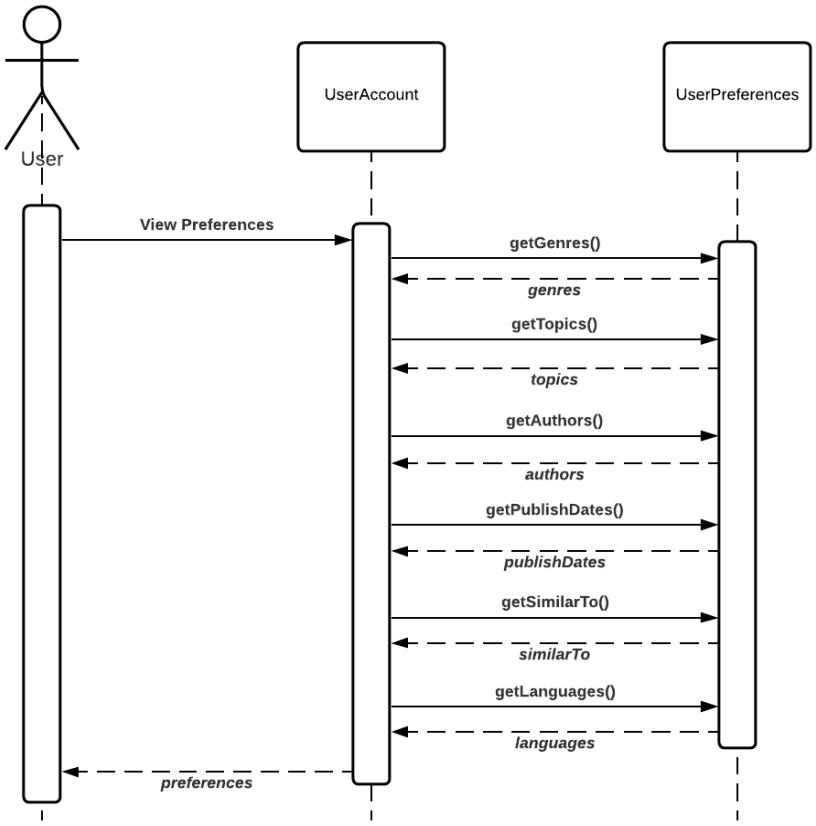


7. Sequence Diagrams

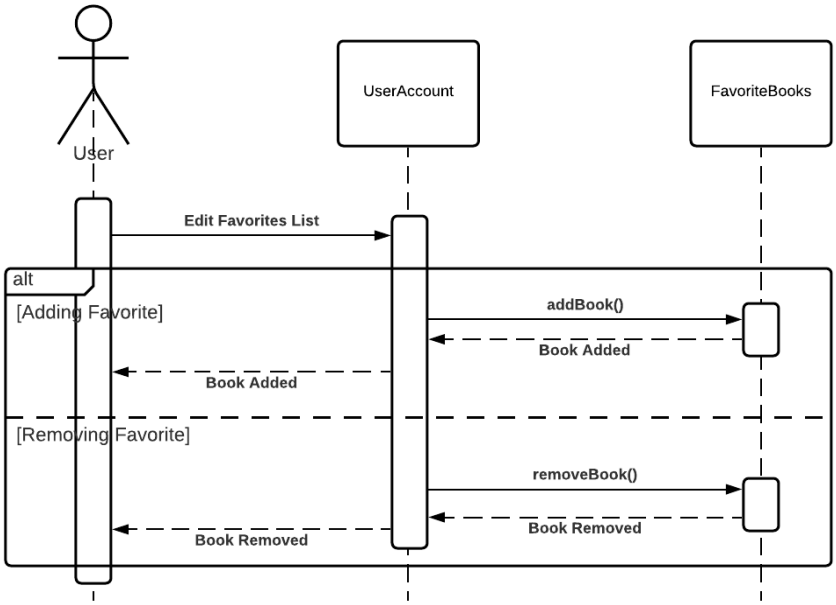
Edit Preferences



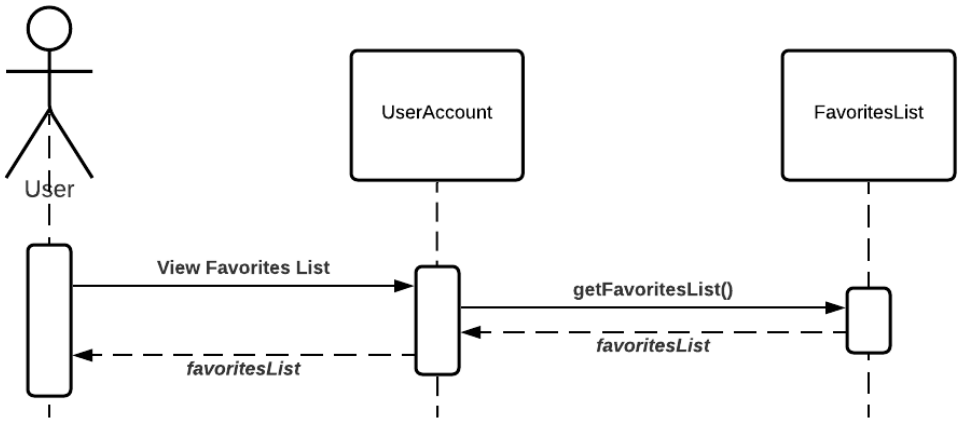
View Preferences



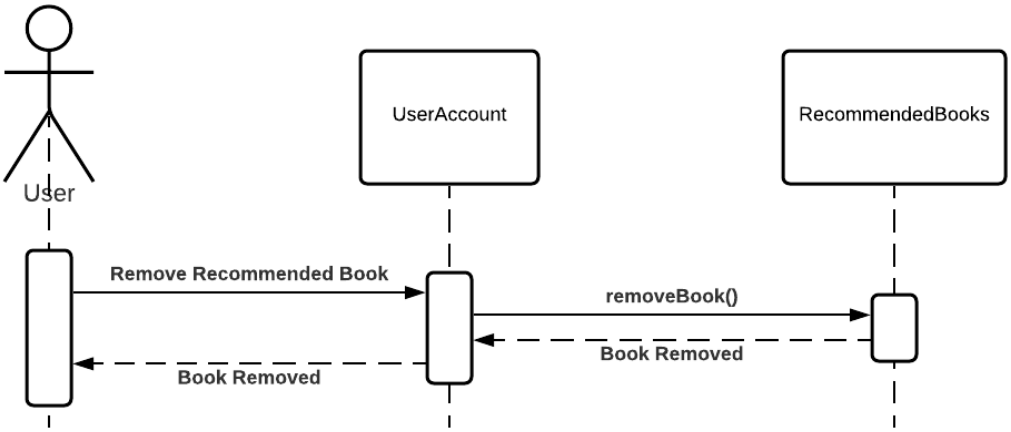
Edit Favorites List



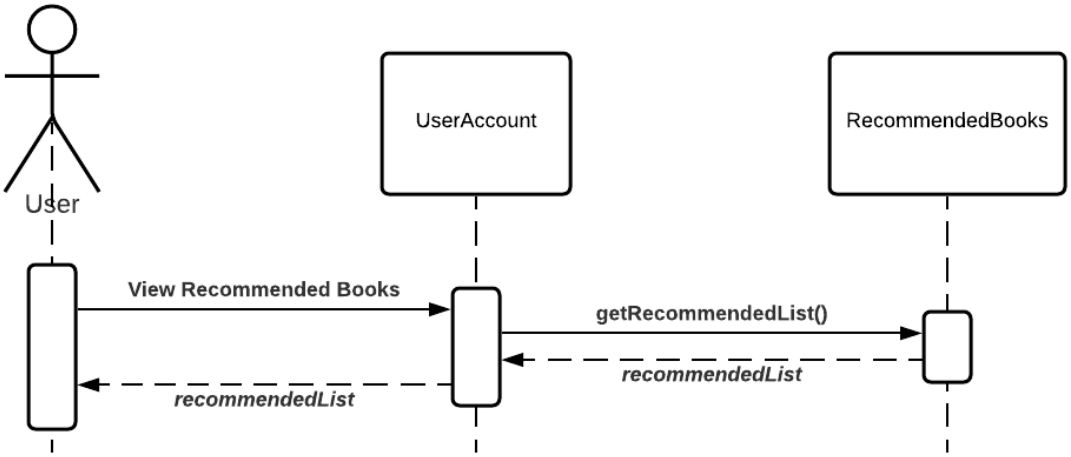
View Favorites List



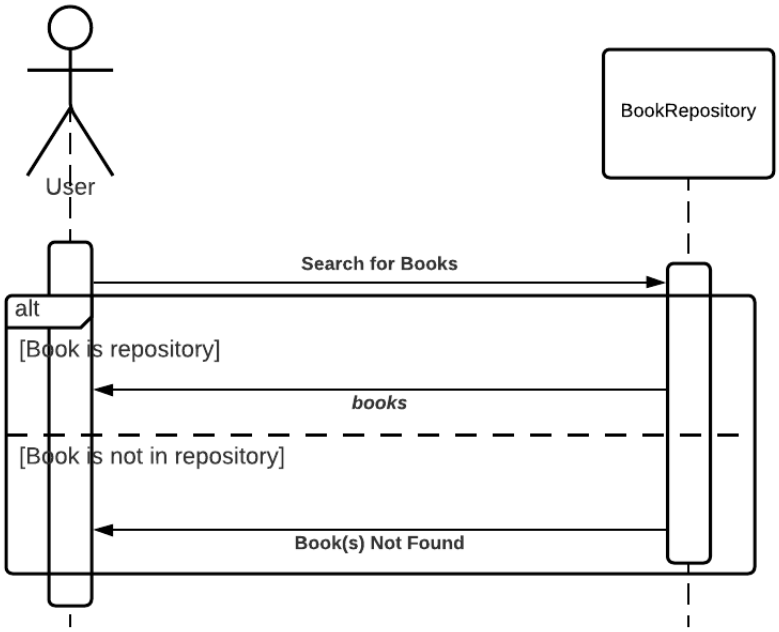
Remove Recommended Book



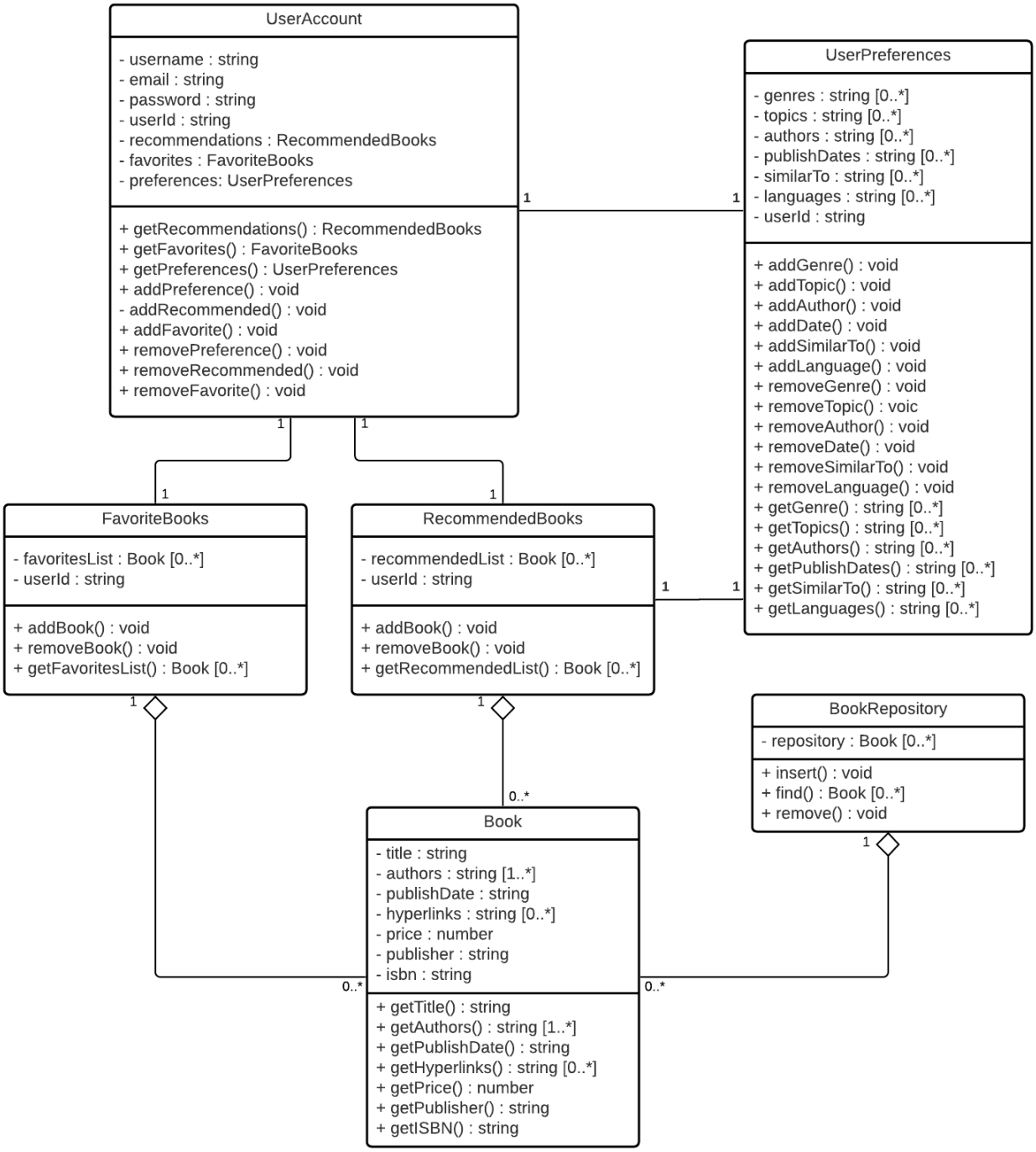
View Recommended Books



Search for Books



8. Class Diagram



9. Architectural Design