Software Requirements and Design Document

For

Group 17

Version 1.0

Authors:
Bryce B
Phuong N
Mason H

1. Overview (5 points).

We are developing a python web application named Scriptoria, which aims to allow users to find, review, track, and share books that they have read. We implemented the Google Books API which provides book information such as the description, cover image, and authors, which we feed into a database of books. Once a book has been added to our local database, users are able to add it to their reading lists and leave reviews and ratings which are reflected in the average scores given to the books on our website. Our aims are to allow users to interact with one another through groups called “Book Clubs”, create personalized profiles to show off their read books and posted reviews, and to suggest friends for users based on the ratings they’ve left on books. Our project arose from being disappointed by the social features present on other similar book review sites and wanting to improve upon those aspects in our project, while also allowing users the same base functionality present on similar website alternatives.

2. Functional Requirements (10 points)

Requirement ID	Requirement (_____ shall ____)	Priority (High/Med/Low)	Rationale (If needed)
FR001	The website shall allow users to see their stored reviews	High	Allow users to edit or delete their existing reviews
FR002	The website shall allow users to friend one another	High	Basis for other social features
FR003	The website shall allow users to form book clubs	High	
FR004	Users shall be able to customize and view their profiles	High	Core aspect of social features
FR005	Users shall be able to interact with reviews left by other users	Medium	
FR006	Users shall be able to message other users	Low	May become part of book club features (like a message bus)
FR007	Users shall get a “recommended” section of books similar to those they’ve already logged	Low	
FR008	Users shall be able to list whether they’ve liked or disliked the	High	Necessary for recommending other books/users

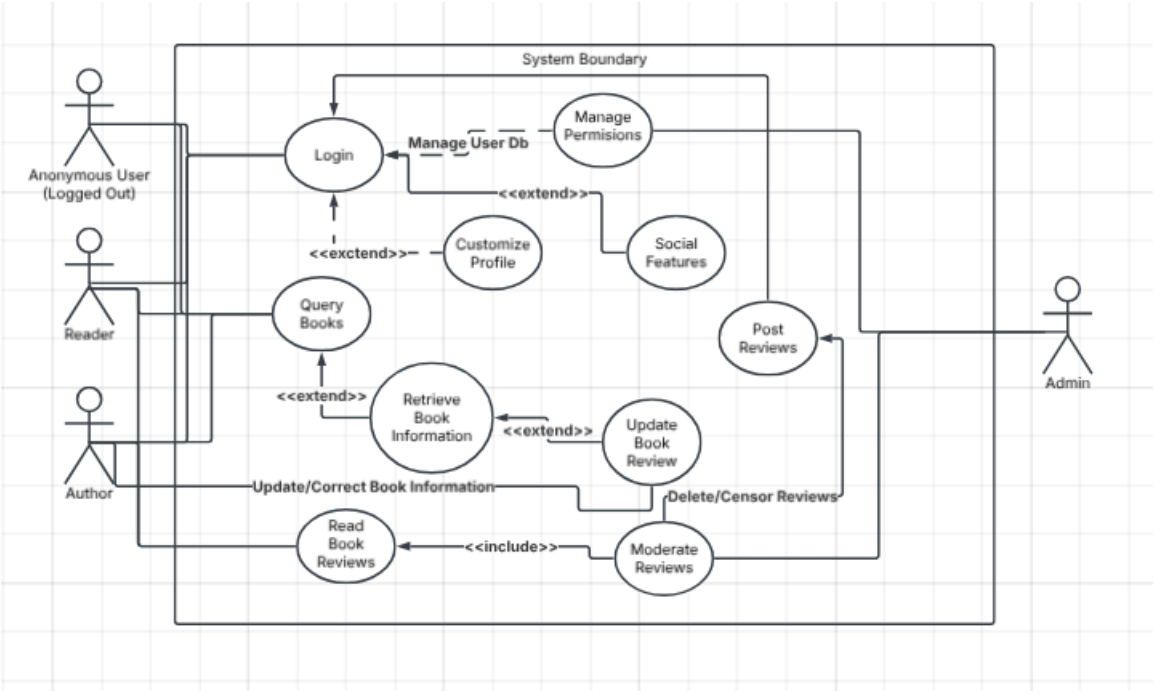
	books they log		
FR009	Users shall receive friend recommendations	Medium	
FR010	System shall allow users to pin reviews to their profiles	Medium	
FR011	System shall allow administrators to moderate discussions and reviews	High	Prevent misconduct or inappropriate speech
FR012	System shall allow users to flag and report inappropriate content	Low	Allows moderators to easily find questionably appropriate content
FR014	Administrators shall be able to add/remove books and suspend users	High	
FR015	Users with author permissions shall be able to modify information fields belonging to written books	High	

3. Non-functional Requirements (10 points)

Requirement ID	Non-Functional Requirement	Rationale
NFR001	Querying a book should take less than 3 seconds	
NFR002	User interface should be intuitive and accessible	Ease of use
NFR003	Information on retrieved books should be accurate, correct, and up to date	
NFR004	The system should be able to handle at least 1000 concurrent users without significant performance degradation	Prevents slow down as user load increases (ensure scalability)
NFR005	System should be responsive to different	

	screen and window sizes	
NFR006	System should be WCAG 2.1 AA compliant and accessible for users with disabilities	Allowing impaired users to interact with the website
NFR007	System should only allow authorized users to view private profile information	Private information should only be available to the user who inputs it
NFR008	System should continue to be built with a modular architecture	Enables new features to be added with minimal refactoring/restructuring
NFR009	System have 99% uptime to allow continuous access for user	
NFR010	System should allow users to have up to 99 friendships	Prevents high load on the database due to high friend counts

4. Use Case Diagram (10 points)



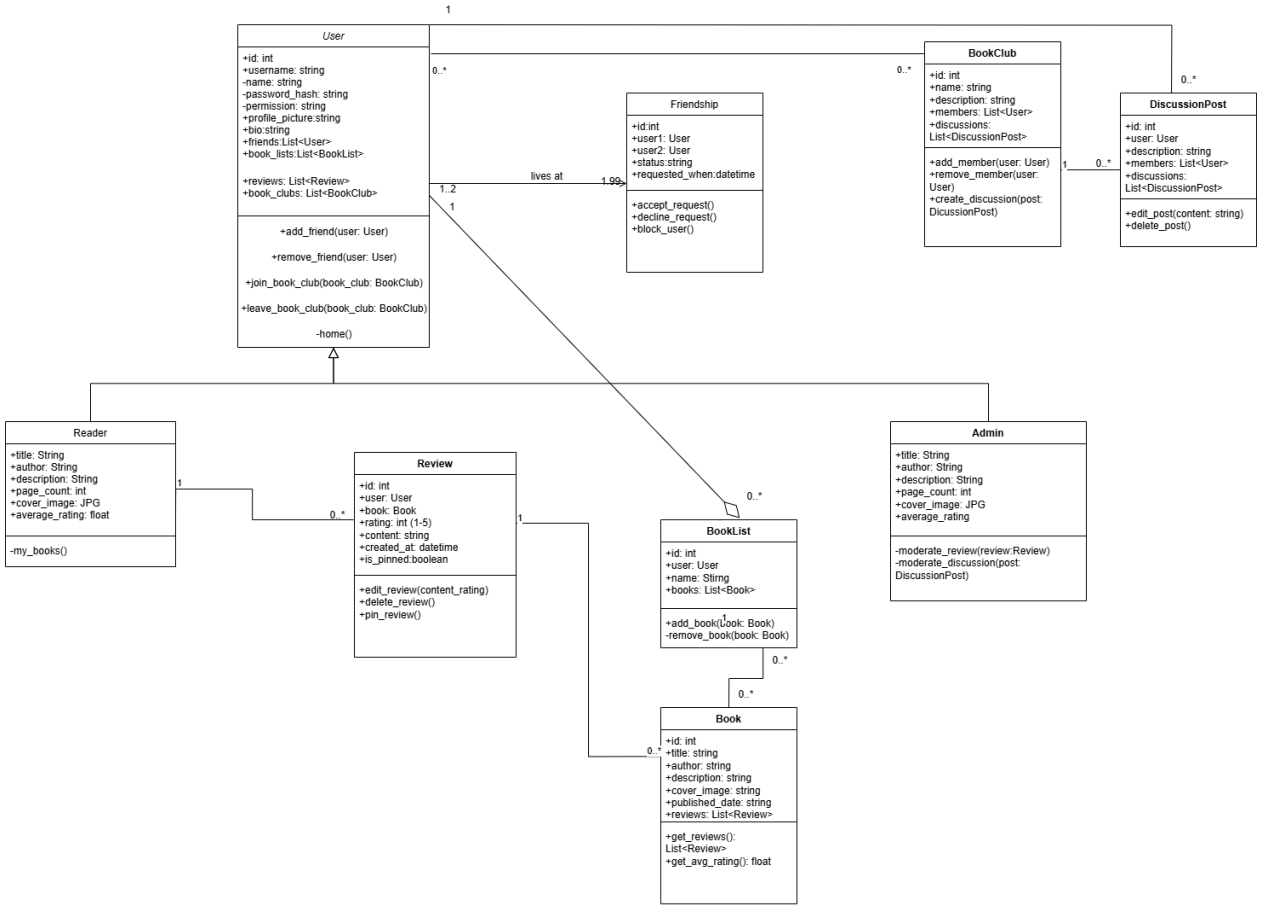
Textual Descriptions:

a) Anonymous user signs up and creates an account with a permission level of reader, the user then customizes their profile and utilizes the (not yet implemented) social features
b) Pre-existing Reader user signs into their account and queries books on the home page, they add a book to their reading list after browsing the query results, then they go to the review page and post a review and rating of the book and share the review to other users.

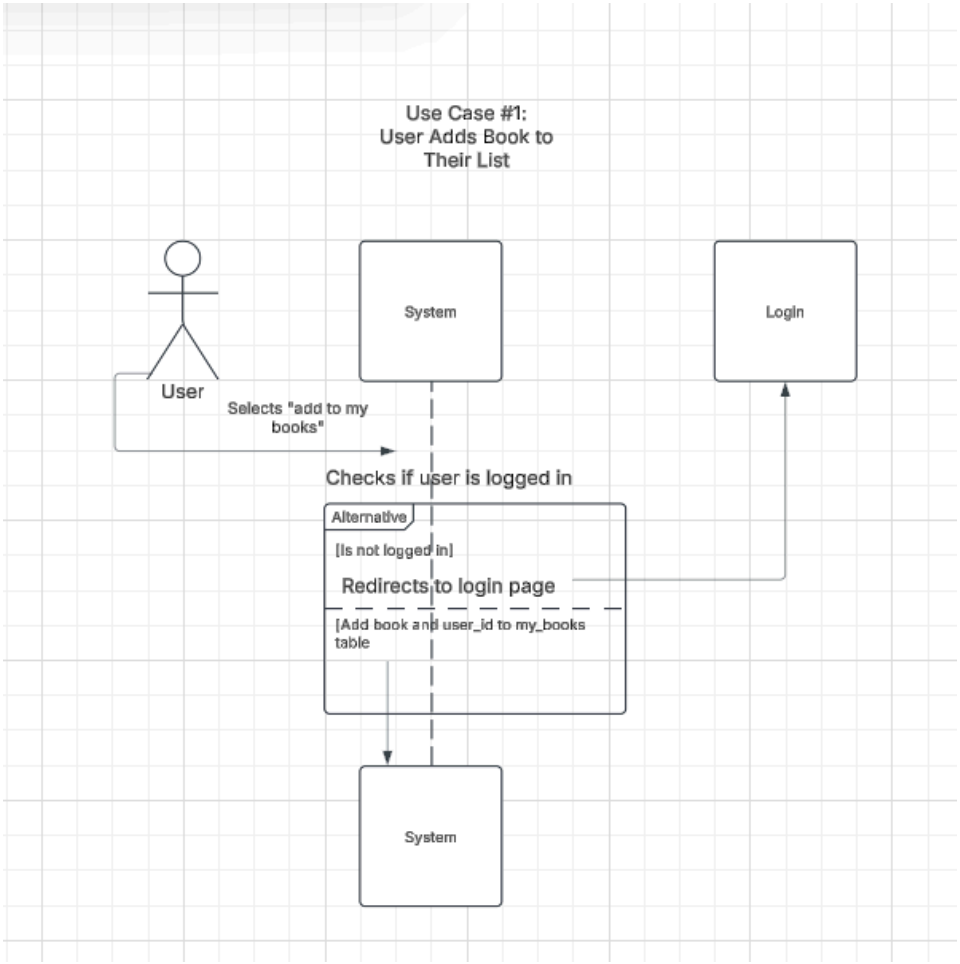
c) (Not Yet Implemented) An author logs into their account, searches for a book they're listed as the author of, and is able to edit the information stored about their book which will show the updated information to other users of the website.
d) An Admin logs in and is able to manage the reviews left by users, deleting the text associated with the ratings or making harmful or offensive reviews hidden to other users on the website.
e) Users who have logged in as Readers or Authors will be able to flag, or report, reviews to be seen by Administrators who can manage the reviews (Use Case d)

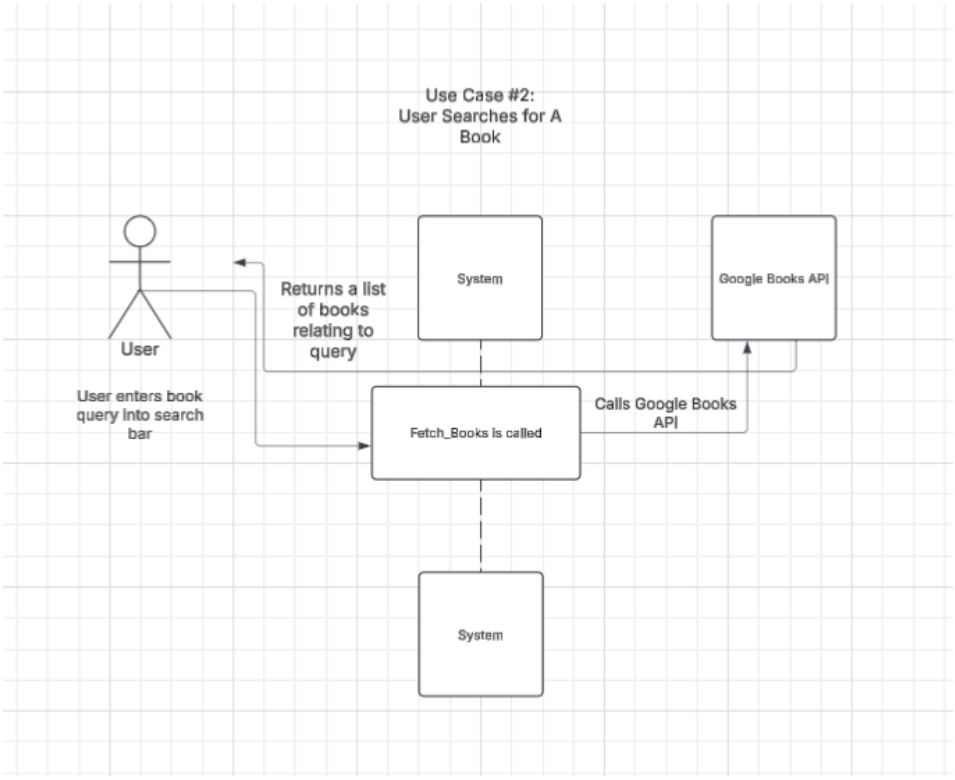
5. Class Diagram and/or Sequence Diagrams (15 points)

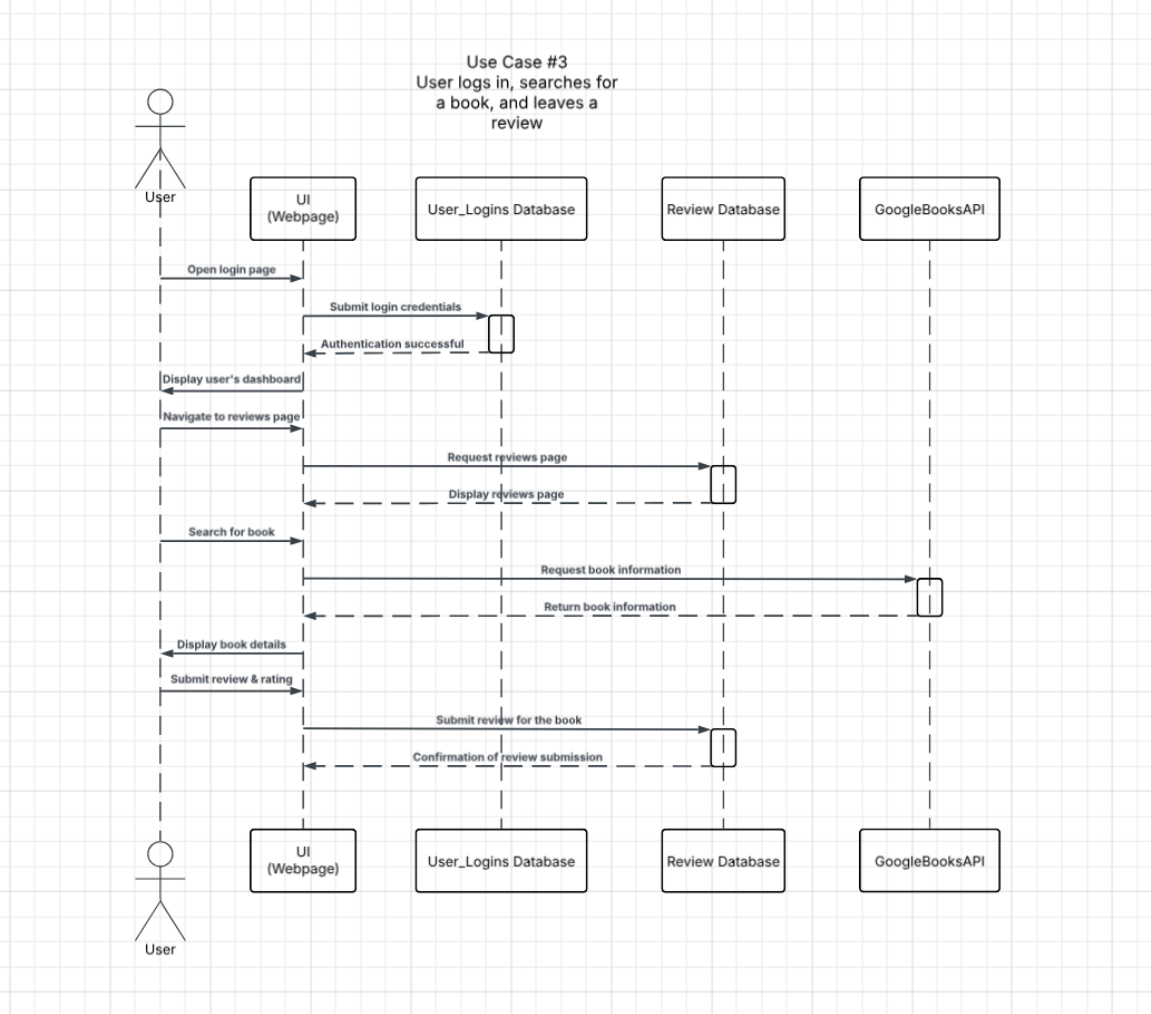
Class Diagram:



Sequence Diagram(s):




Use Case #2:
User Searches for A Book
User
System
Google Books API
Returns a list of books relating to query
User enters book query into search bar
Fetch_Books is called
Calls Google Books API
System



Use case #3 is completely different from Increment #1's implementation because we've since been able to implement the functionality it describes, and this new diagram is more accurate to our code.

6. Operating Environment (5 points)

Our project's operating environment is Windows 10 and runs on Python 3.12. Our website isn't being hosted on a server so therefore must be run from a local computer which has python installed and access to the repository. We do not have any specific required hardware platforms or codependent applications which our project relies on and it is unlikely that specific requirements in these areas will arise within the scope of our project. However, we haven't been able to explicitly isolate and test different hardware platforms to ensure our website's functionality on devices that aren't our own.

7. Assumptions and Dependencies (5 points)

This project utilizes the third party Google Books API and is dependent on that for searching and retrieving the information of books and storing those queried books in our local database, without a connection to this API, our app will not work. In our project's current iteration, we are also

dependent on the host/user of the app to run initializeDb.py in order to establish the database of debug users, enable the creation and storage of new users, allow book info to be stored locally, and handle any other SQL database-related functions of our application. Due to the constraints of our team's devices, we are also unable to confirm that the app works on operating systems aside from running locally on Windows computers. The code also requires the necessary imported packages to be installed on the local computer which is hosting the code, which is an error that we've encountered in sharing the code repository with one another and neglecting to mention new dependencies. Outside of the above dependencies, however, we have no reliance on other commercial or third-party components in our app and are currently unable to test other operating environments at the time of this increment's submission.