**Software Requirements and Design Document**

**For**

**Group 17**

Version 1.0

**Authors**:

Bryce Beck

Mason Hartle

Phuong Nguyen

# Overview (5 points)

We are developing a python web application which aims to allow users to find, review, and track their progress through books. We intend to implement this using a book API such as Google Books to produce a catalog of books, to which our website will add functionality for reviewing and marking books as read. Additionally, we are aiming to add the ability to connect with other users and manage friendships or organize “Book Clubs” to communicate between users and share reviews or thoughts on books. We also aim to allow users to display books they’ve read on their profile as well as manage lists of their favorite books, books they would like to read, and books they disliked, therefore enabling users to fully customize their profiles and how they appear to other users browsing their profile page. After being disappointed by the lack of cohesive social features on similar websites which deal with book reviews, we were motivated to make a website which shared similar base functionality but with a more accessible social aspect through the inclusion of features such as book clubs or pinning reviews to one’s profile.

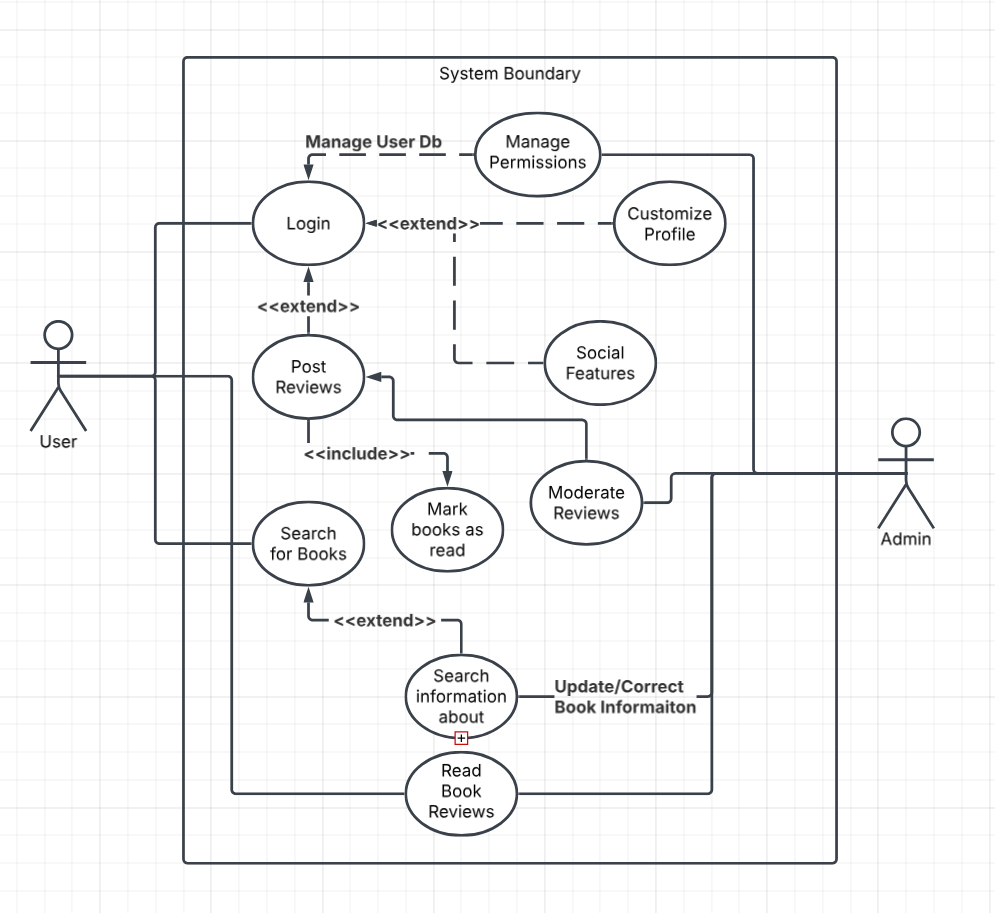
# Functional Requirements (10 points)

| Requirement ID | Requirement | Priority  (High/Med./Low) | Rationale (If needed) |
| --- | --- | --- | --- |
| FR001 | The website shall have a home page to allow users to login, signup, or browse reviews as a guest | High |  |
| FR002 | The website shall have a repository of books which users can track their progress | High | Provides context for user reviews to other readers |
| FR003 | The website shall allow users to post reviews of books | High | Other users can consult others before they start a book |
| FR004 | The website shall enable users to friend other users | High | Öne of the main objectives of the website |
| FR005 | The website shall allow users to form book clubs | High | Creating groups of users to interact with each other |
| FR006 | Users shall be able to view customize their profiles | Medium | Making the user’s experience more enjoyable |
| FR007 | Users shall be able to register with the website | High |  |
| FR008 | Users shall be able to interact with reviews left by other users | Medium |  |
| FR009 | Users shall be able to list books they would like to read, have read, and read but disliked | Medium | Enables users to further customize their profile’s appearance to other users |
| FR010 | Users shall be able to message other users | Low | Encourages repeat users |
| FR011 | Users shall get a “Recommended” section | Medium | Often, book review websites will have a recommended section that features book |
| FR012 | Users shall receive friend recommendations | Low | Allow users to see recommended users to friend |
| FR013 | The system shall allow users to search for books in the API | High |  |
| FR014 | The system shall allow users to rate the books that they have read | Medium | Enhances sharing of opinions between users and further fleshes out reviews |
| FR015 | The system shall enable users to pin reviews to their profile | Medium | Pin post tends to highlight what the user considered “most important |
| FR016 | The system shall allow administrators to moderate and remove inappropriate reviews or discussions | Medium | May become more necessary with a larger user count |
| FR017 | The system shall allow users to flag and report inappropriate content | Low | Enables ease of moderation |

# Non-functional Requirements (10 points)

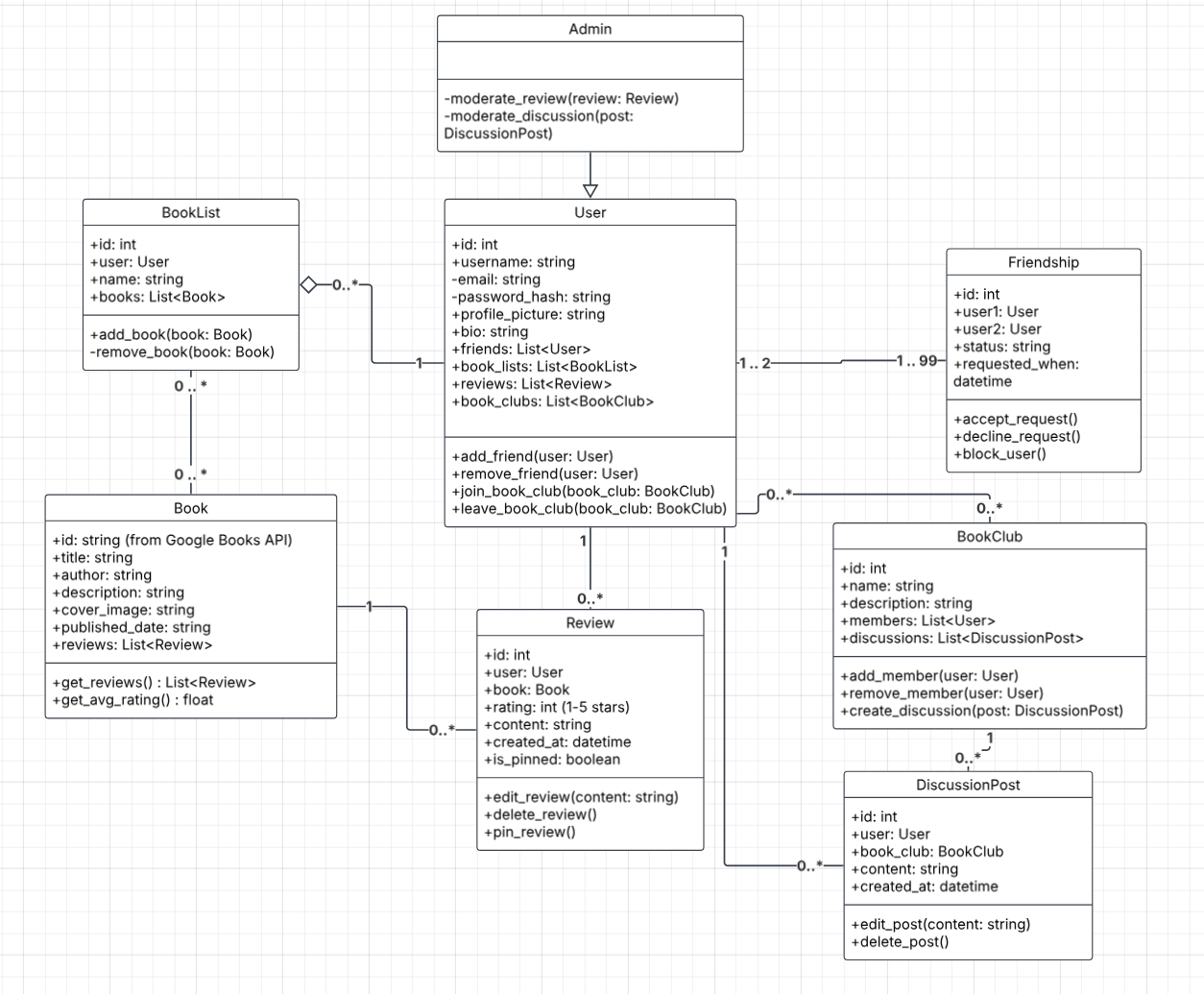
| Requirement ID | Non-functional Requirement | Rationale |
| --- | --- | --- |
| NFR001 | Information entered into the database must be encrypted | Ensures only the user and admin has access to a user’s credentials |
| NFR002 | Results fetched from Books API must load in under 3 seconds | Ensures the website is easy to use |
| NFR003 | User interface should be intuitive and accessible |  |
| NFR004 | Information on books being retrieved must be accurate and correct |  |
| NFR005 | The system should be able to handle at least 1000 concurrent users without significant performance degradation | Prevention against slowdowns as user count increases |
| NFR006 | The system should be responsive on different screen and window sizes |  |
| NFR007 | The website should be WCAG 2.1 AA compliant and accessible for users with disabilities | Allowing impaired users to interact with the website |
| NFR008 | The system should prevent unauthorized access to user’s accounts and private profile information | Private information should be exclusively accessed by the users themselves |
| NFR009 | The system should have 99% uptime to allow continuous access for users |  |
| NFR010 | The system should be built with a modular architecture to enable the addition of new features | Reduces the need for refactoring in the future |
| NFR011 | The system should log out after an extended amount of time of not using it | Prevents unnecessary load on server and db |
| NFR012 | The system should allow users to have up to 99 friendships | Upper limit placed to prevent problems with database |

# Use Case Diagram (10 points)

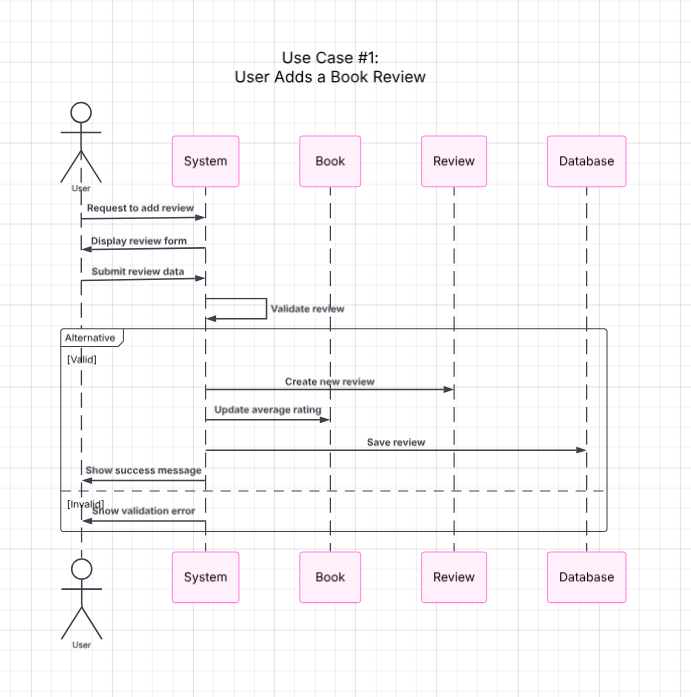


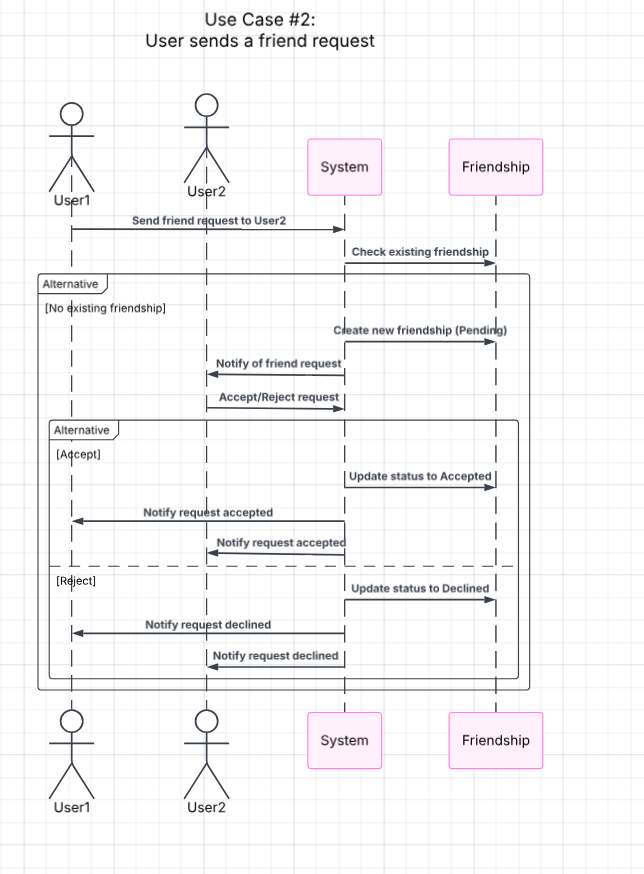
# (15 points)

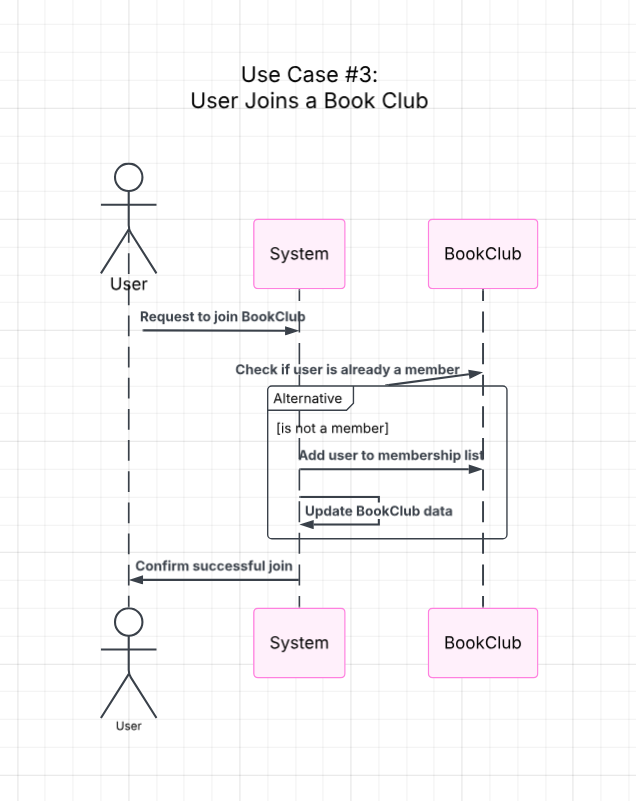
**Class Diagram:**



**Sequence Diagram(s):**

****

****

****

# Operating Environment (5 points)

In the project’s current state the website is not being hosted and therefore must be run from our computers before being accessible. Provided that the individual attempting to run the software has a computer with Python and access to our full repository, the program should encounter no errors when running. At the time of submission, we do not have any parts to the program which require specific hardware platforms or codependent applications although this may become a complication in later iterations. Additionally, as a result of our group all programming and testing on Windows, we have been unable to test and verify if the website is working on other OS during this increment.

# Assumptions and Dependencies (5 points)

In this increment, the program running successfully depends on the user running the “initializedb.py” file before the driver file “app.py”. Additionally, the project assumes the user has installed the full project repository since the project depends on folders such as static or templates for the website to display properly. Although it is not yet implemented in this increment, the project’s intended functionality will depend on the Google Books API which is managed by Google and therefore if it became nonfunctional then our project would become similarly unsuccessful. “signup.html” is dependent on being able to connect to “[..]maxcdn.bootstrapcdn[..]” for its stylesheet.