**CS 3354 Software Engineering**

**Project Deliverable 1 - 2**

Personal Virtual Bookshelf

Group #5

**Group Members**

* Abeel Khan
* Marvin Farinas
* Jay Trivedi
* Kevin Sun
* Christopher Castro
* Albion Krasniqi
* Syed Shaheer Ali
* Yusuf Kuzey

**Deliverable 1 (With Corrections):**

**1. Project Goal**

The goal for deliverable 1 is to:

* Create a repository for our website
* Use the Agile Process Model
* Make sure the functional and nonfunctional requirements of our website are figured out.
* Go over all the use cases that our users will go through
* Work over the architectural design of our website

**2.** ​ **REPOSITORY URL LINK:**​  
https://github.com/AlbionKr/virtual-bookshelf.git ​[​](https://github.com/Nasya-C/3354_Group4/)

**3. Delegation of tasks:**

|  |  |
| --- | --- |
|  |  |
| ***Abeel Khan:*** | Software Requirements/ Functional and non-functional / Requirements traceability diagram/ Delegation of tasks / finishing and putting together everyone’s works in document. |
| ***Kevin Sun:*** | Sequence Diagram |
| ***Jay Trivedi:***  ***Syed Shaheer Ali:*** | Software Process model and why we are choosing this model  Use case Diagram |
| ***Marvin Farinas:*** | Class Diagram |
|  |  |
| ***Chris Castro:*** | Architectural Diagram |
| ***Albion Krasniqi*** | Creating Repository Link |
| ***Yusuf Kuzey:*** | Conclusion |
|  |  |

# 4. Software Process Model​

Our team has chosen an agile-based approach to developing our virtual bookshelf. Our goal is first to create a functioning prototype and then iteratively add features. We anticipate encountering various challenges, and an agile methodology will allow us to adapt by adjusting or removing features that no longer align with our vision. Additionally, this approach enables us to focus on specific aspects of the project through sprints, breaking the workload into smaller, more manageable tasks. This makes development more efficient and less overwhelming. Lastly, since this approach is iterative, it allows us to continuously refine our user interface, ensuring a more user-friendly experience.

# 5.Software Requirements

# 5.a.) Functional Requirements:

# 5.b.) Non-functional requirements

**5.c) RTM**

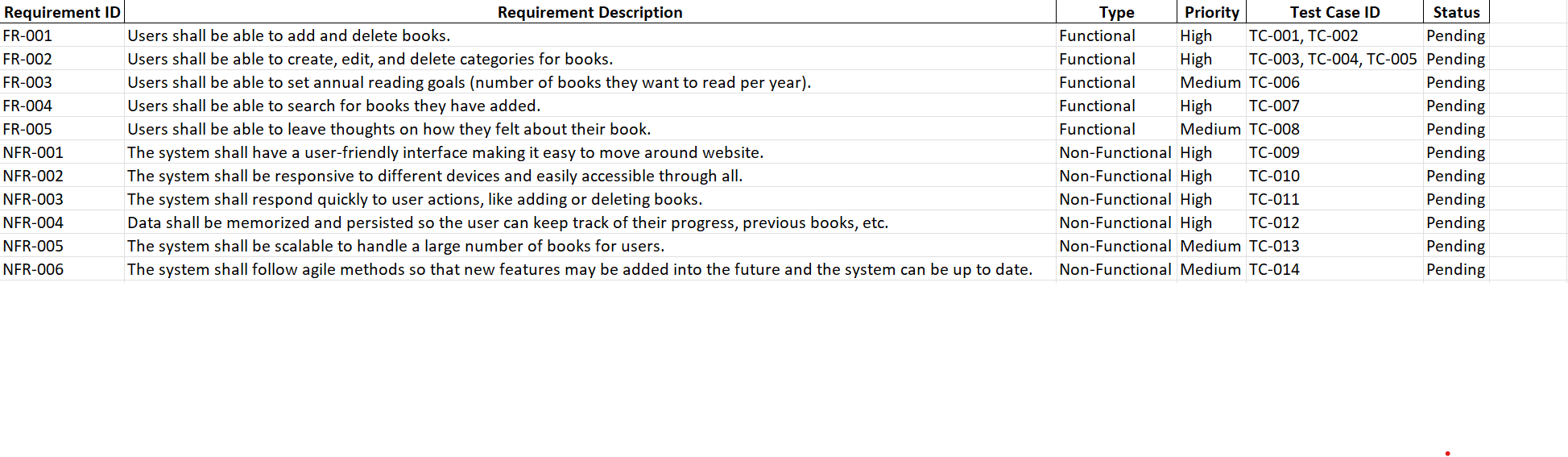
**Functional Requirements -**

* Users shall be able to **add and delete books**.
* Users shall be able to **create, edit, and delete categories** for books.
* Users shall be able to **set annual reading goals** (number of books they want to read per year).
* Users shall be able to **search for books** they have added.
* Users shall be able to **leave thoughts on how they felt about their book and rank them.**

**Non-Functional Requirements -**

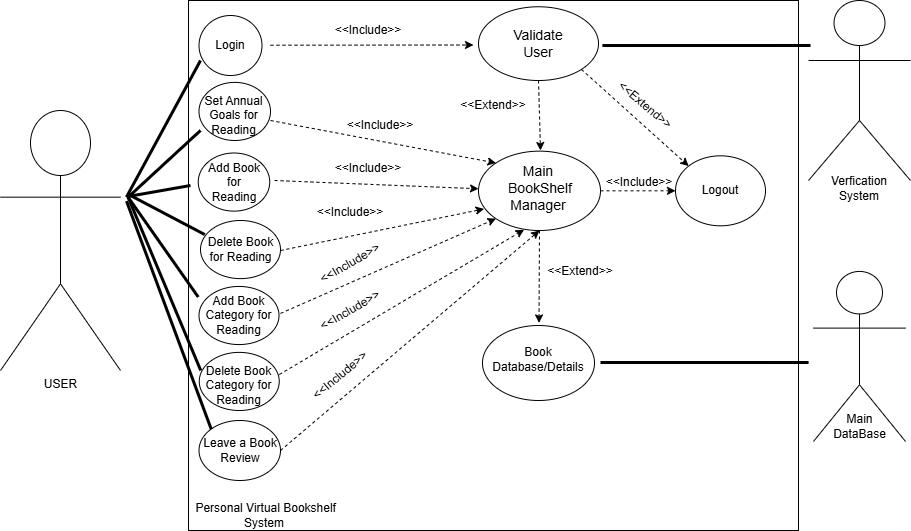
* The system shall have a **user-friendly interface making it easy to move around the website.**
* The system shall be responsive to different devices and easily accessible through all.
* The system shall **respond quickly** to user actions, like adding or deleting books.
* Data shall be memorized and persisted so the user can keep track of their progress, previous books, etc.
* The system shall be **scalable** to handle many books for users.
* The system shall follow agile methods so that new features may be added into the future and the system can be up to date.

**RTM:**

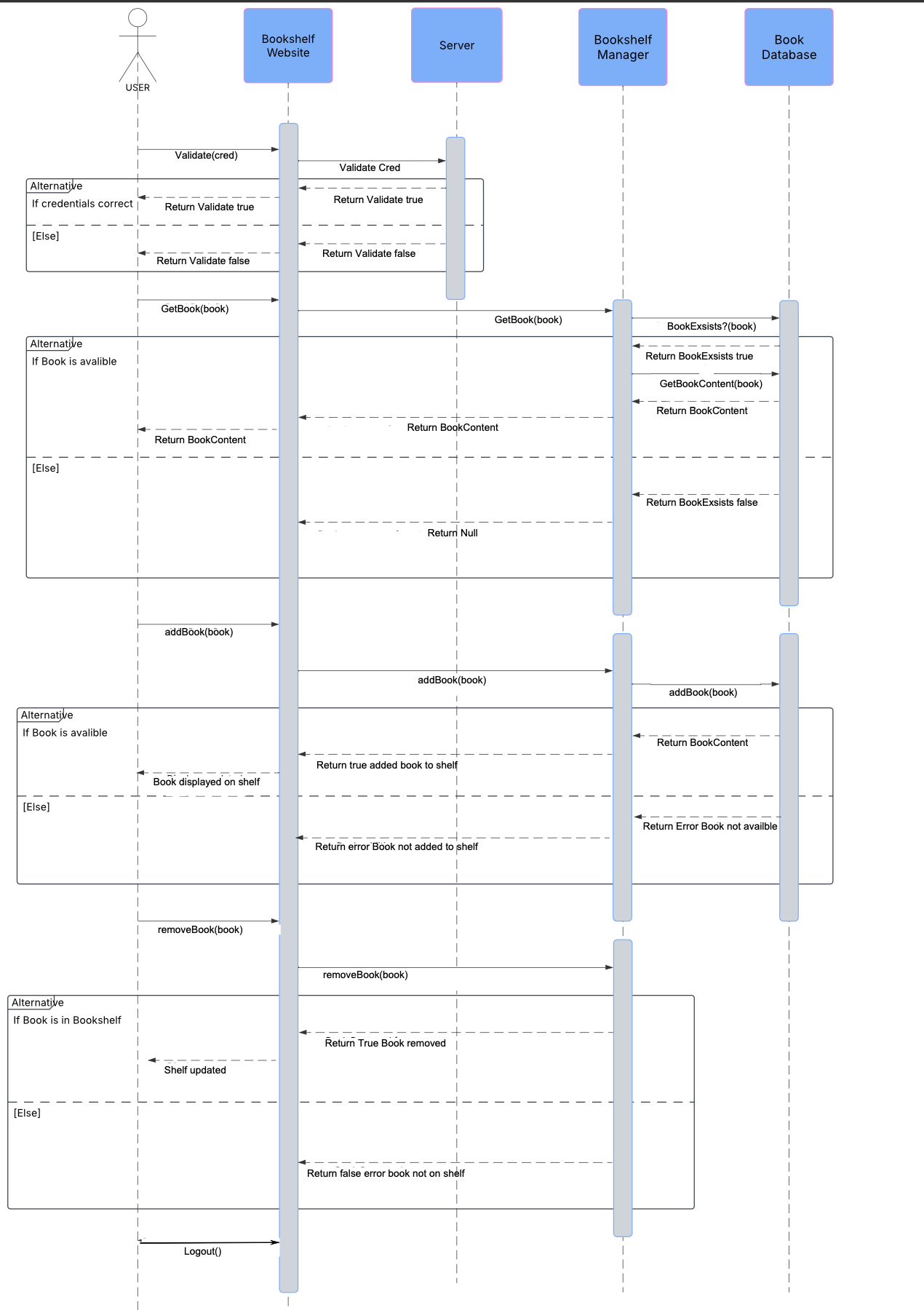


# 6. Diagrams (Use Case, Sequence Case, Class Diagram)

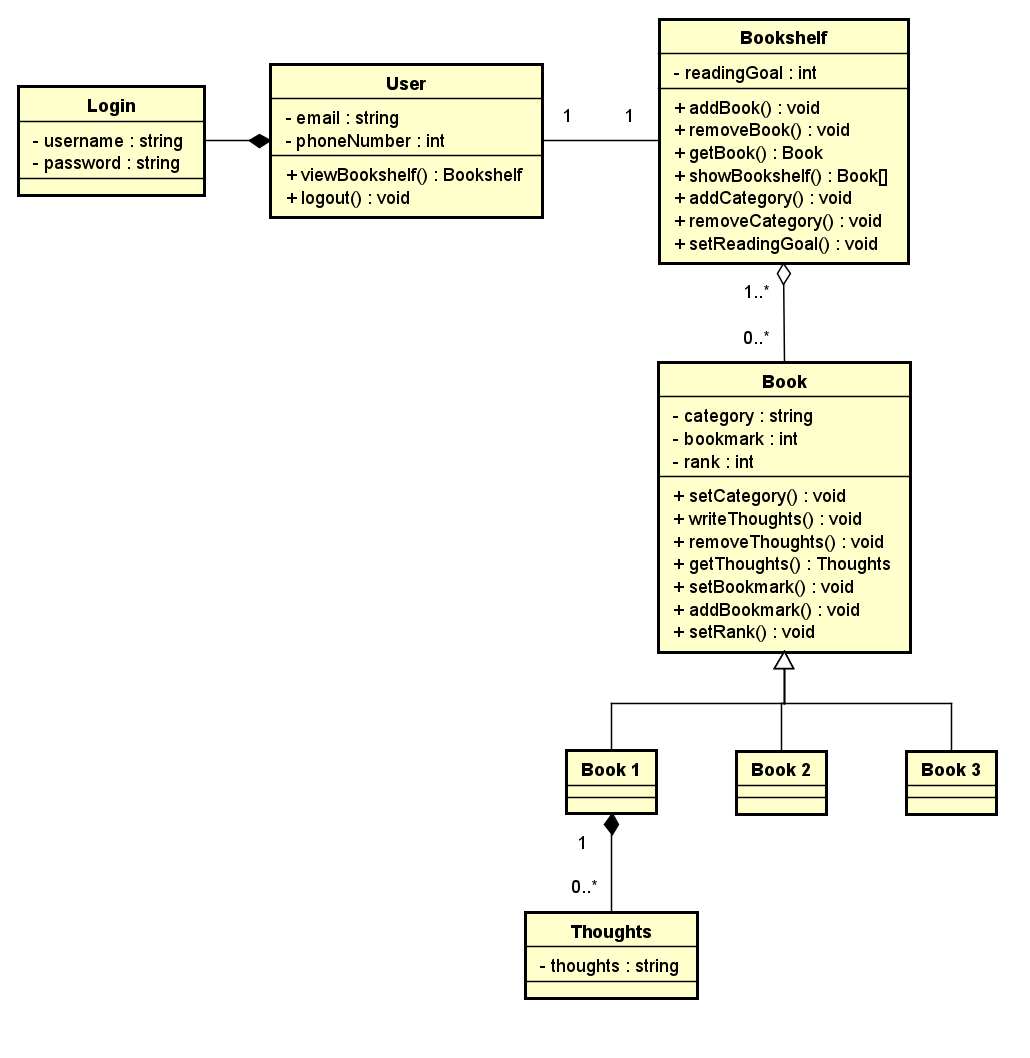
**Use Case:**



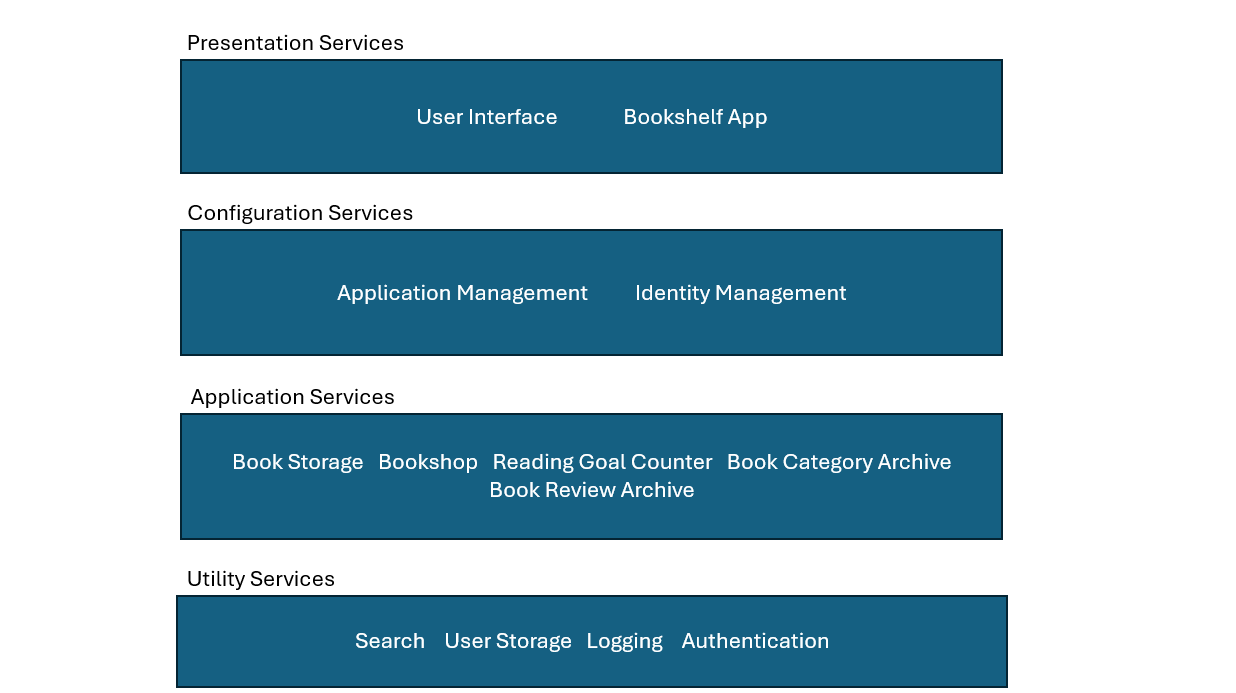
**Sequence Case:**



**Class Diagram:**



# 7. Architectural design



## 7.1 Describe why the pattern is selected

Our Team chose the Layered Architecture Pattern because it can be split so that each section can be tinkered with independently. The layered pattern also allows us to add or remove different new features, so the system stays up to date. The layered pattern can also have the same systems in each layer that help increase the dependability of the system.

**Deliverable 2**

**1. Delegation of tasks:**

* **Abeel Khan:** Delegating tasks, creating and preparing word documents, setting up deliverable 2, Comparison with similar Designs, A test plan for software, and references.
* **Marvin Farinas:** References and Project Scheduling
* **Jay Trivedi:** References and Presentation Slides

* **Kevin Sun:** References and Project Scheduling

* **Christopher Castro:** References.

* **Albion Krasniqi:** References.

* **Syed Shaheer Ali:** References and A Test Plan for Software

* **Yusuf Kuzey:** References and Conclusion

**2. Corrected Deliverable 1**

Shown above under the Deliverable 1 section.

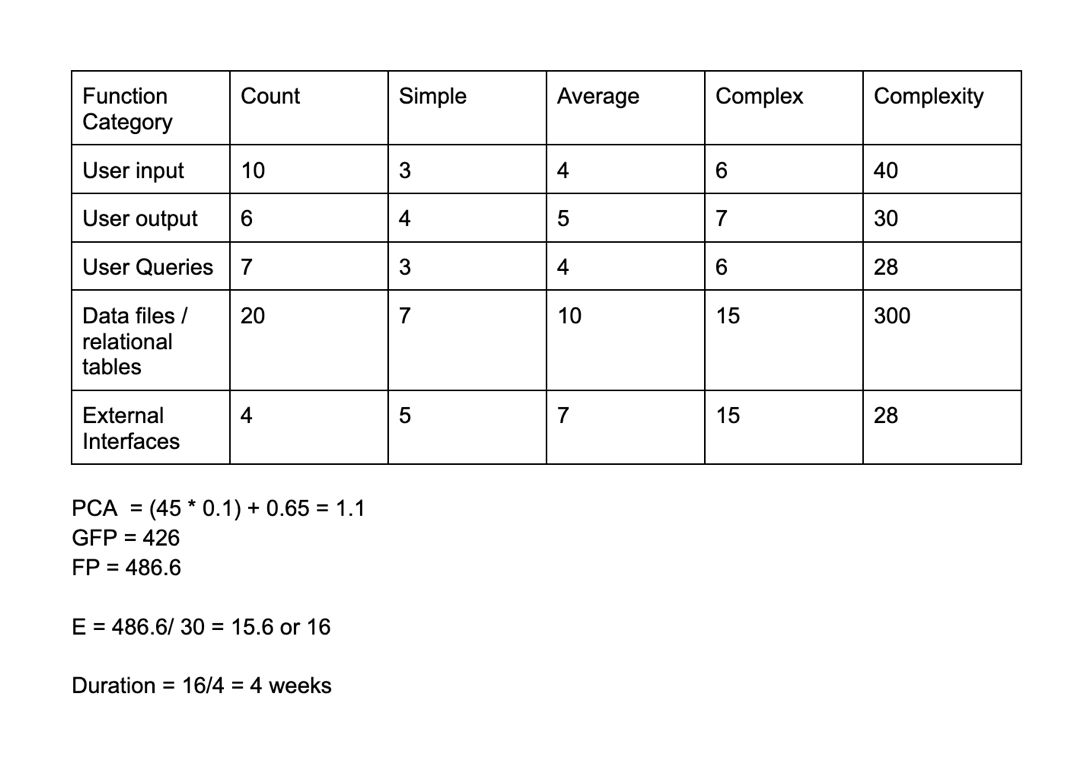
**3 – 4. Project Scheduling (Cost, Effort, Pricing Estimation, Project Duration, and Staffing.)**

**Estimated Cost of Hardware Products: $350/yr**

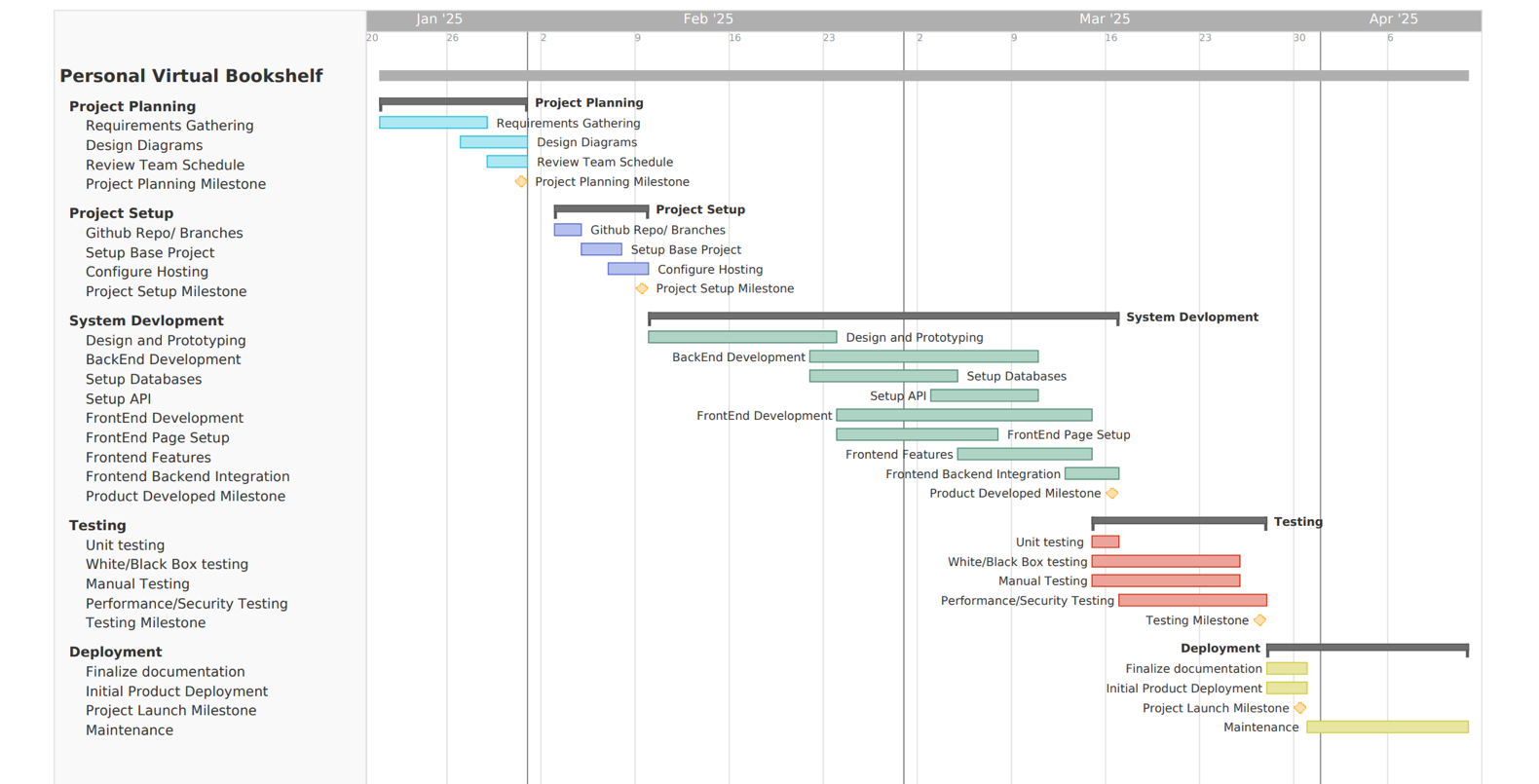
**Estimated Cost of Software Products: $40/yr**

**Estimated Cost of Personnel: $10,000**

* **Project Duration: 4 weeks**
* **Staffing: 4**

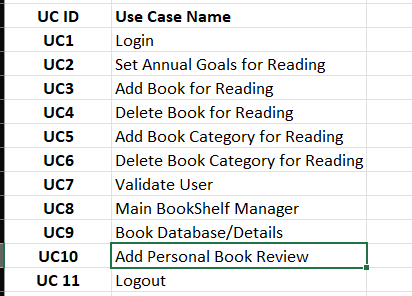


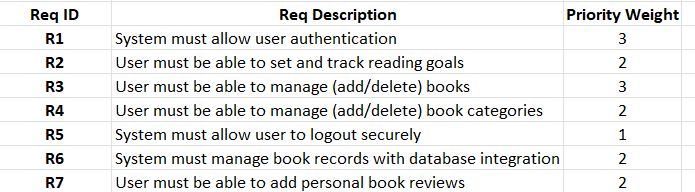
**Gantt Chart:**

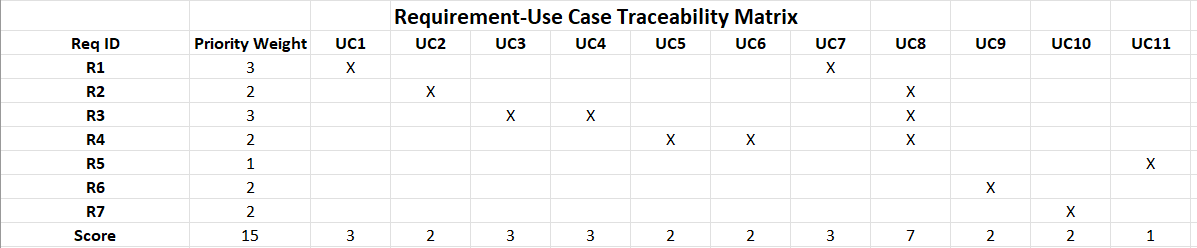


**5. A Test Plan for Software:**

**Requirement-Use Case Traceability Matrix:**

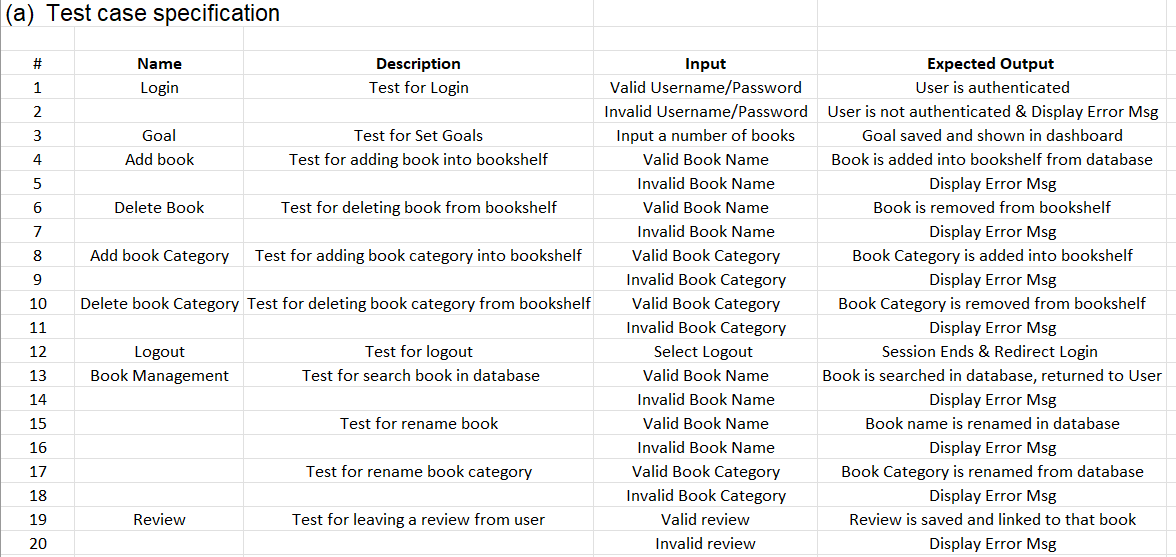


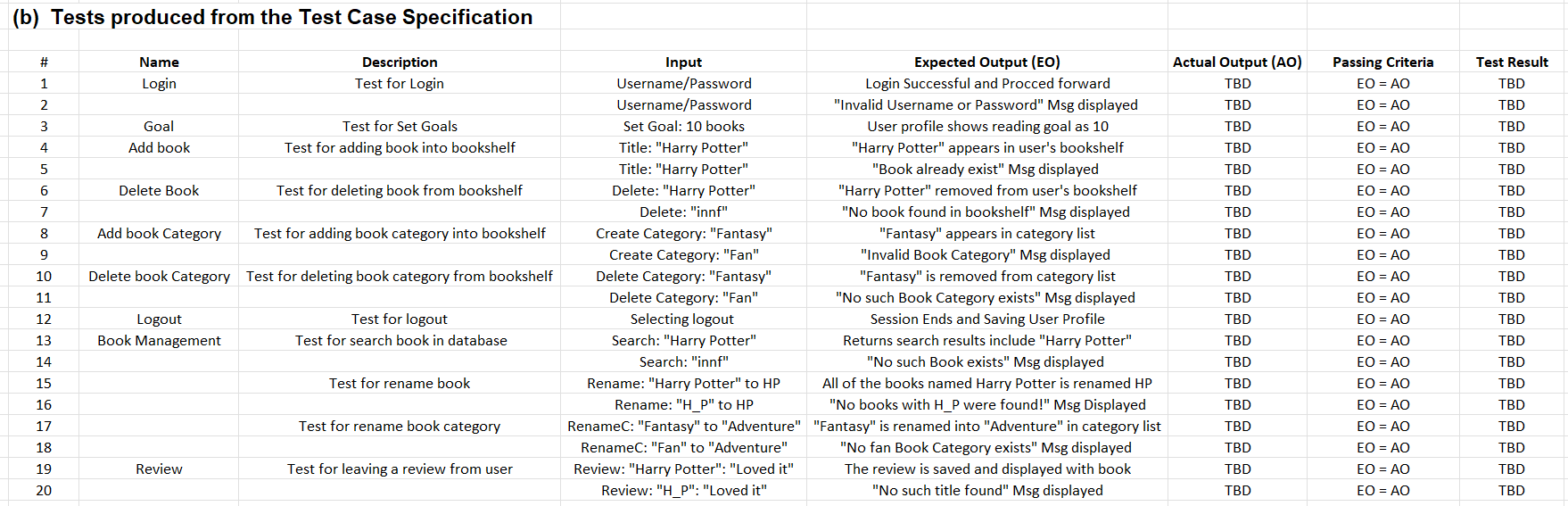




**Specification Table:**

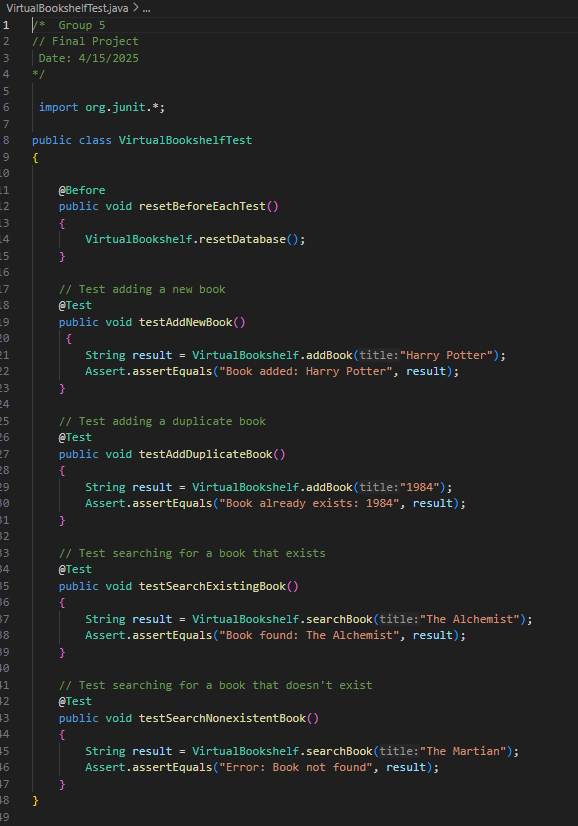
Part A:



Part B:  


Two Test Cases:





The test case specification tables presented here provide a structure for validating the functional correctness for the Personal Virtual Bookshelf System. Table (a) outlines the test case design based on the system's core requirements and associated use cases. Each test case includes a descriptive scenario, input conditions (both valid and invalid), and the expected system behavior, ensuring that the application handles typical, boundary, and erroneous conditions effectively.

Whereas Table (b) translates those specifications into executable tests by providing some sample input values, with their expected outputs (EO), and placeholders for actual results (AO), enabling clear criteria for pass/fail evaluation.

Finally, these tables support both black-box and white-box testing strategies by covering internal flow validation. Furthermore, this structured approach ensures traceability from requirements to test execution, aiding in quality assurance and system reliability.

**6. Comparison of Work with Similar Design:**

The website that has a similar goal and structure as what we are trying to achieve is Goodreads.com. Both Goodreads and our website share the idea that the user can add books to their reading list, bookmark upcoming books that the user wants to read, be able to send recommendations, and view comments about what others thought of the book.

One of the strengths that Goodreads has is the ability to join communities with people who have similar interests as you, so that they can talk about books that they enjoyed and recommend books to other users who enjoy that kind of genre. Another strength that Goodreads has is the function of browsing books with awards attached to them, making it so that people who want to read award-winning books are able to access them.

One weakness that Goodreads has is that they don’t have a personal rankings list where you leave your own private notes just for the user. They only have the option to leave reviews in public. One of our goals is to be able to rank and review books just for the user and the friends that they add.

Goodreads has a team who constantly updates and keeps up the website, which means it has excellent maintainability, something we hope to achieve as well. New books are added, new books are awarded, and new members have constant customer support.

The UI is simple and easy to follow. You log in and then have the option to go to your books, browse books, search for books, or join the community. It also shows a homepage of current popular books that Goodreads itself recommends. We want to have a similar UI that is fast and easy to navigate.

**7. Conclusion:**

Throughout this project, our goal was to design the foundational elements of a virtual bookshelf interface, outlining the structure, user interactions, and key features. Each team member contributed to different aspects of the design. Abeel Khan led the documentation and requirements, ensuring that all functional and non-functional aspects were clearly defined. Kevin Sun worked on the sequence diagrams, mapping out how users would interact with the system. Marvin Farinas focused on the class diagram, organizing the components of the system. Syed Shaheer Ali contributed with the use case diagram. Chris Castro designed the architecture, providing a structural overview. Jay Trivedi chose the Software Process Model and created the slideshow. Albion Krasniqi helped organize the repository and the overall project. As for me, Yusuf Kuzey, I worked on the conclusion, repository, and references.

Throughout the design process, we faced a few challenges. One of the main challenges was ensuring that our design was flexible and scalable, allowing for future additions and changes. We had to make sure the system could handle new features, such as user ranking and private notes, without compromising the performance. Another challenge was organizing user data in a way that was both efficient and easy to manage.

If we had more time, we would have refined the design further, focusing on areas like data storage and user interface elements. We would also have liked to explore additional features and improvements, ensuring the design was as user friendly as possible.

Overall, this project has allowed us to develop our skills in system design and collaborate as a team to create a solid foundation for a virtual bookshelf. We are proud of the work we have done, and we are excited to see how this concept could evolve with further developments.

**8. Link for Presentation Slides:**

[**https://docs.google.com/presentation/d/1-G2TTEycQhF04-ha04wOzkDSnyTtLPjrgK\_wB1RKPzU/edit**](https://docs.google.com/presentation/d/1-G2TTEycQhF04-ha04wOzkDSnyTtLPjrgK_wB1RKPzU/edit)

**9. Repository Submissions:**

<https://github.com/AlbionKr/virtual-bookshelf.git>

**10. References**

*Goodreads*. [www.goodreads.com](http://www.goodreads.com/). Accessed 27 Apr. 2025.

Burenko, S. (2025, April 10). *How much does it cost to develop software for a project?: Uptech*. RSS. <https://www.uptech.team/blog/software-development-costs>