

**Group Members:**

Vedanta Dhobley: vjd41@scarletmail.rutgers.edu  
Avani Bhardwaj: ab1572@scarletmail.rutgers.edu  
Shazidul Islam: si194@scarletmail.rutgers.edu  
Akshat Shah: avs91@scarletmail.rutgers.edu  
Kutay Kerimoglu: kk851@scarletmail.rutgers.edu  
Alan Patel: akp122@scarletmail.rutgers.edu  
Anthony Matos: amm720@scarletmail.rutgers.edu  
Joel Cruz: jc2125@scarletmail.rutgers.edu

**Project Title:** Recommending Alpha

**Group Number:** 16

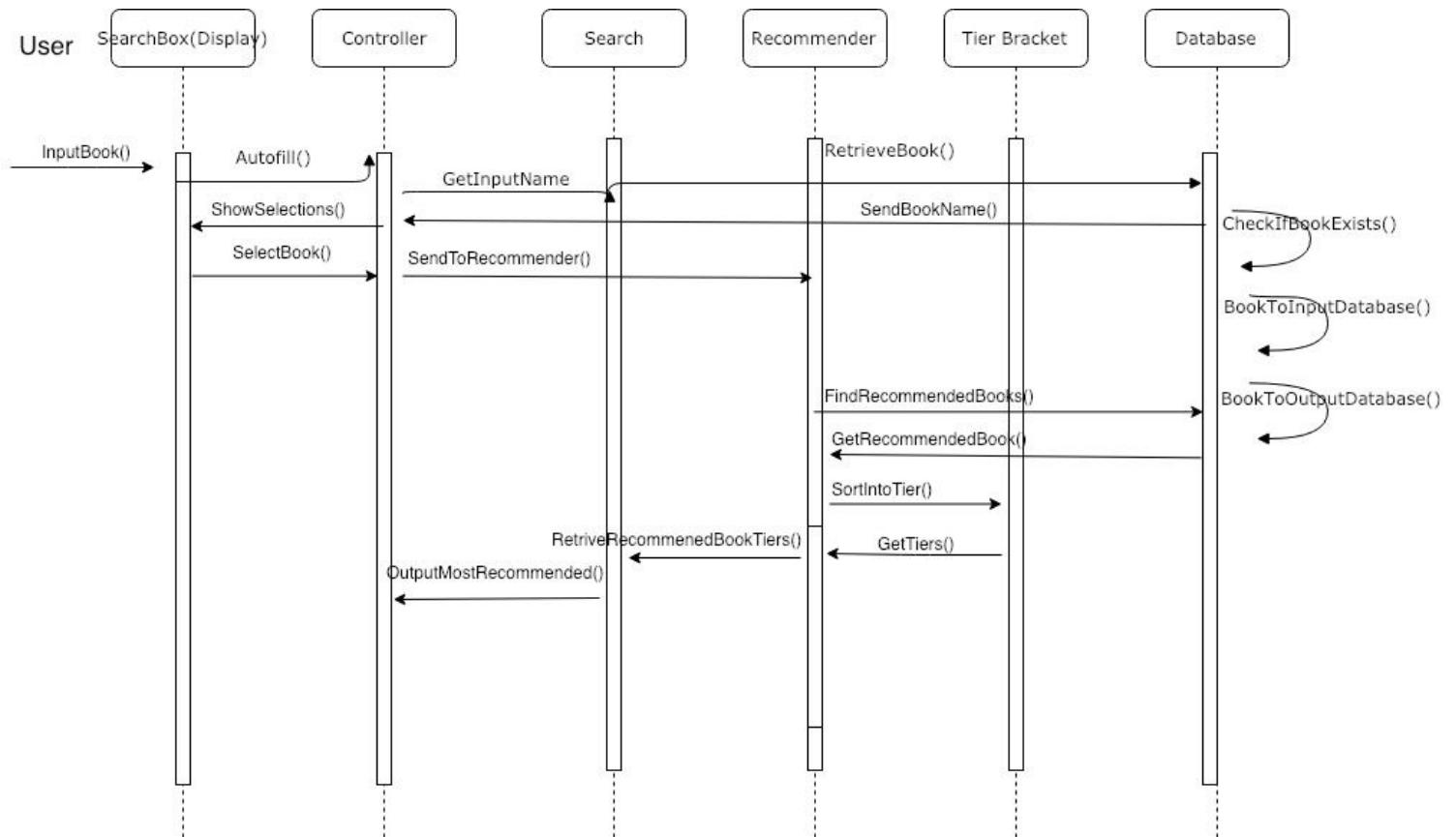
**Date:** March 3, 2019

**Report 2 Part 1**

## Interaction Diagrams

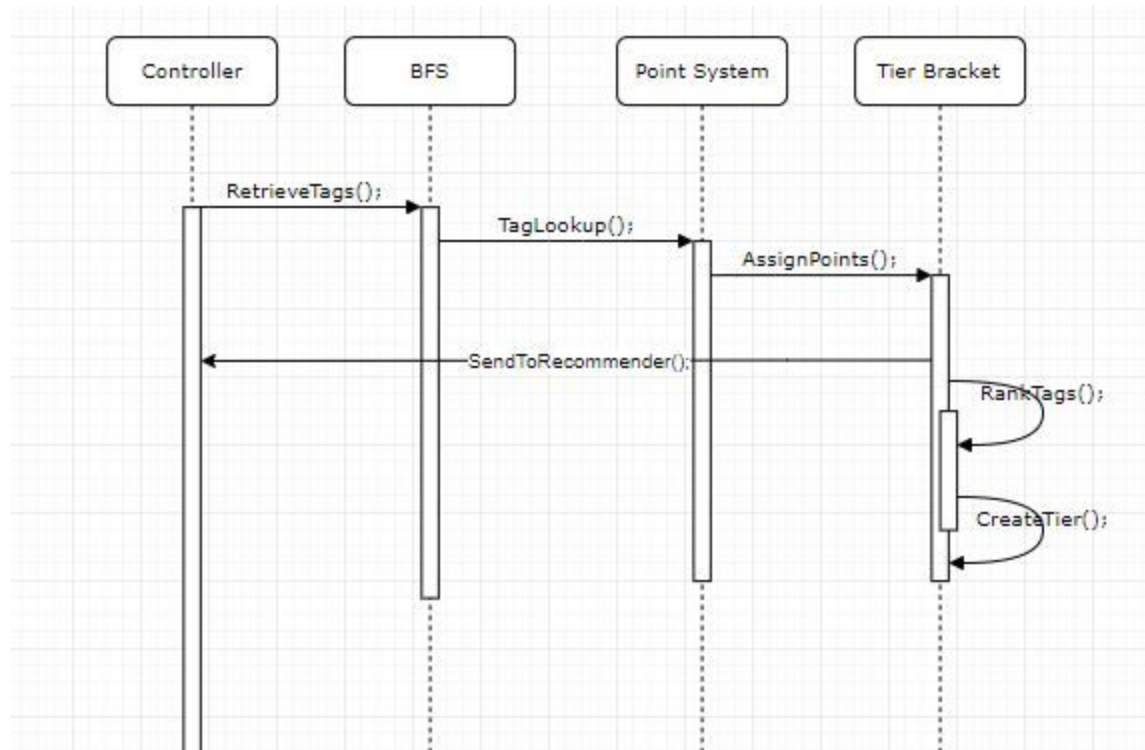
These are the use case diagrams for the two use cases we have chosen to fully address.

### Use Case 3



For use case 3, which is the most dense use case in regards to tasks that need to be done, the main focus was making many distinct objects who allows for use case 3's tasks to be broken down into as many singular tasks as possible. The main design principle that allowed for us to assign responsibilities to all the objects was the high cohesion principle, since our use case 3 involved the most computational and data manipulation out of all the use cases. The single responsibility principle from SOLID also helped us in breaking down the task into classes, followed by the high cohesion principle assisting in helping break down responsibilities of the concepts in each class. The goal was to make sure not a single class had objects responsible for too much at once, by creating several objects interacting with each other in order to make up that class.

## Use Case 4



Use case 4 had a lot more to do with the user interface, and communicating the correct results to the user. Given the task of use case 4, it was a dead giveaway to us to look into the low coupling principle when assigning responsibilities to objects because the low coupling principle mostly deals with communications in regards to one object being the primary one to delegate responsibilities to other objects. The goal here was to make sure that any object isn't responsible of communicating too many things independently, and that we made sure different things that needed to be communicated were each handled by single objects. Liskov's Substitution Principle also assisted with this use case because it allowed us to make sure that each object should operate independently, but should still interact with other objects. No changes in our software, system, or objects around a certain object should directly affect any object, only changes directly made to an object should affect its performance and outcome.

Interaction diagrams are most effective when they're manufactured off the basis of a design system sequence diagrams. What a design system sequence diagram does is it looks at all of the objects that help drive a system, and breaks down how exactly each object interacts with other objects. These interactions conclusively define how each object contributes to execute what the system is intended to do. They are very in depth and much more detailed visual representations of how each object plays a certain role in getting a task completed.

Objects are the different functions inside of a class (programming definition of class), which make the class fulfill its job. So we had to look at how each object interacts with one another, and break down a task into pieces and make sure each piece is given attention from objects.

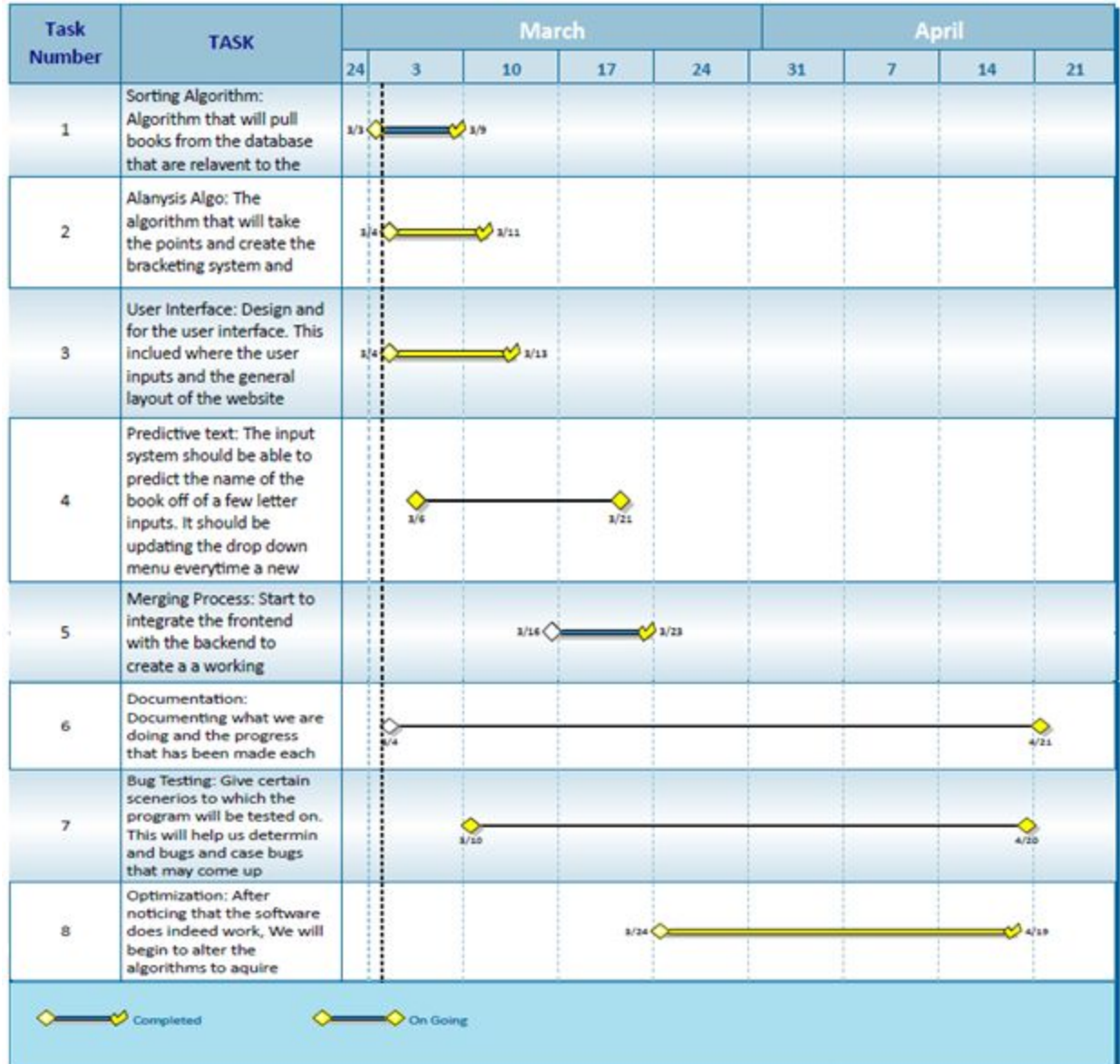
When assigning our objects to responsibilities, we used certain design principles in order to make sure that each object had a fair share of responsibilities and that the work between the system was fairly spread apart so no single object had too many tasks. We also used some of the techniques provided through the SOLID design principle to make sure that we were forming a diagram that does necessary tasks, and nothing was redundant or irrelevant.

### **Project Management and Plan of Work**

- For the project thus far, there have been no issues with managing individuals and mini groups. Everyone used the given references in the report format listed in the schedule. While there were some difficulties pertaining to how a certain section should be done, the entire group came together to help each other solve those problems given their diverse areas of expertise. People who were better at explaining the thorough descriptions of the algorithms that are included in this project were able to explain those ideas in words very well. People who were better at drawing diagrams and graphs helped create very organized displays which were very easy to read. Those who were good at photoshopping images made the example UI prototypes which came out to be very realistic.
- Having team meetings every week ensures that everyone is on top of their responsibilities. This ensured consistent work ethics and a well organized distribution of work. As for the formatting of the report, we looked at examples from previous years and wanted it to look as concise and easy to read as possible. By hyperlinking each section of the table of contents to the section's location, it became easier for everyone to navigate throughout the report. We followed the format exactly how it was described to do so, which kept the entirety of the report looking proper and organized. We will continue to work the same way, with everyone contributing to the best of their abilities, and keeping the same meeting times every week to ensure consistency for the future reports.
- Given the feedback we received for the first report, we will have a group meeting this week to see what we will be changing from that report and implemented into our next report and actual work. There were suggestions on doing the algorithm a different way using a hash tables instead of a breadth first search and reorganizing the requirements list that we have. We have to figure out how to implement

whether or not a user is a child or not in order to give him or her restrictive access.

## Projected Milestones



## Individual Plan of Work

Name	Did	Currently	Will Do
<b>Shazidul (Analysis Algorithm)</b>	Learned SQL and Python for compatibility Documentation	Working on Analysis Algo and sorting methods Documentation	Merge algo with database and site Documentation
<b>Vedanta (Analysis / Sorting Algorithm)</b>	Created Github repository. Documentation	Working on backend algorithm. Documentation.	Complete algorithm (backend/analysis) and merge with database
<b>Avani (Presentation and UI design)</b>	Documentation and Group Meeting Coordinator	Documentation and Group Meeting Coordinator.	Documentation and Group Meeting Coordinator Will add input for UI
<b>Anthony (Predictive Text)</b>	Contributed concepts for the functionality.	Working with front end for site development.	Will merge algo with site (he is the bridge between both)
<b>Alan (Database Entries)</b>	Database creator (local csv, static).	Maintain and organizing database	Will maintain and update database on new entries
<b>Akshat</b>	Learned SQL and Python, gained understanding of Bootstrap and AWS	Currently working on front end site.	Have predictive text up and merge with algo
<b>Seymour</b>	Research for point/tier system. Documentation	Working on sorting and selecting algo.	Help merge the algo with the analysis algo and the site
<b>Kutay (Presentation and UI design)</b>	Documentation	Documentation. Learning programming languages to help with bug testing. Cleaning up csv.	Documentation and bug tester

**What use cases have been implemented?**

None of the use cases have been implemented or completed; however we have begun tackling challenges leading to the completion of use case 3.

**What is already functional, what is currently being tackled?**

So far, the search, database (static, local) and retrieval for the recommender have been implemented abstractly. We've begun work on creating the object classes in order to store this information logically.

**List and describe other relevant project management activities.**

Branches and folders have been created for each development team (backend, recommendation, frontend, database) with ground broken in backend regarding retrieval and object creation. As far as project management is going, everyone has been contributing equally and uploading weekly information to our github repository. With upcoming deadlines, the use cases will be implemented soon and the sorting algorithm will be set in motion in about a week or so. After that has been done, we will start bug testing and checking the database to see that the information can be retrieved with ease. As for now, there are a lot of unreadable characters in the excel spreadsheet that need to be fixed for the search to work properly. We will go through the database to fix the errors which will be done during the bug testing portion of our project.

**Breakdown of Responsibilities**

**Names of modules and classes that each team member is currently responsible for developing, coding, and testing:**

- Coding - Shazidul and Vedanta will code the database and backend communication. Alan will structure the database and dynamically code it for new data entries
- Development -Akshat will develop the Predictive Text search bar and Anthony will develop other frontend site functioning with respect to the backend development
- Testing - Kutay and Avani will test input for UI functioning and bugging

**Integration Coordination:**

- Akshat, Anthony and Seymour will coordinate the integration of the backend algorithms with the front-end site functioning

**Performance and integration testing:**

- Kutay and Avani will be testing out implementation and outputs of use cases and multiple checkpoints, along with debugging sub functions and sub classes