

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: February 17, 2019

Stakeholders

- User
- Administrator
- Bookstore Owners

Actors and Goals

- User
 - User will be an initiating actor.
 - The user's goal will be to retrieve recommended books from the database which match the search that the user has made.
 - Includes: Parents, Children, Avid Readers, School teachers looking for books for their classrooms, People finding books to buy for their friends.
- Administrator
 - Administrator will be an initiating actor.
 - The administrator's goal will be to ensure that people are satisfied with their selections. He or she will make sure that the database is refreshed to a more up-to-date version of the database so that there are more options for users to choose from. He or she will also make sure that the tags on each book are still placed appropriately for the new incoming books.
- Bookstore Owner
 - Bookstore Owner is a participating actor.
 - The bookstore owner's goal is just to allow the system to be implemented within his or her store. He or she will just take a look around to make sure customers have easy access to the website on the computers in the store.
- Librarian
 - Librarian is a participating actor.

- The librarian's goal is just to allow the system to be implemented within his or her library. He or she will just take a look around to make sure customers have easy access to the website on the computers in the library.
- Website Developer
 - Website Developer is an initiating actor.
 - The website developer's goal is to keep the website up to date and refreshed with the latest books. He or she will check the frontend and backend parts of the system to ensure that the appearance of the UI looks the way that it should.
- Database Manager
 - Database Manager is an initiating actor.
 - The database manager's goal is to keep updating the list of books in the database every two weeks or so. He or she will make sure that the books are equipped with the appropriate tags. He or she will notify the updated database to the website developer, the store owner, librarian, etc.

Use Cases

1. Case 1 (Entering the List of Books)

- a. User→ Begins typing the book
- b. System← starts using predictive text, begins to suggest book name
- c. User→ select book which they are looking for.

2. Case 2 (Find books according to their list of read books)

- a. User→ Enter the name of book A into the search bar
- b. System← begins to find the name of the book in the database for predictive text
- c. User→ Selects the book from the predictive text
- d. System← Creates table that stores all the books that the user input
- e. System← Starts running the similar match algorithm, and then stores them in the same file with the books that were entered
- f. System← Use Analysis Algorithm to rank the books form S→ C

3. Case 3 (Decide which book is a better match for a single book)

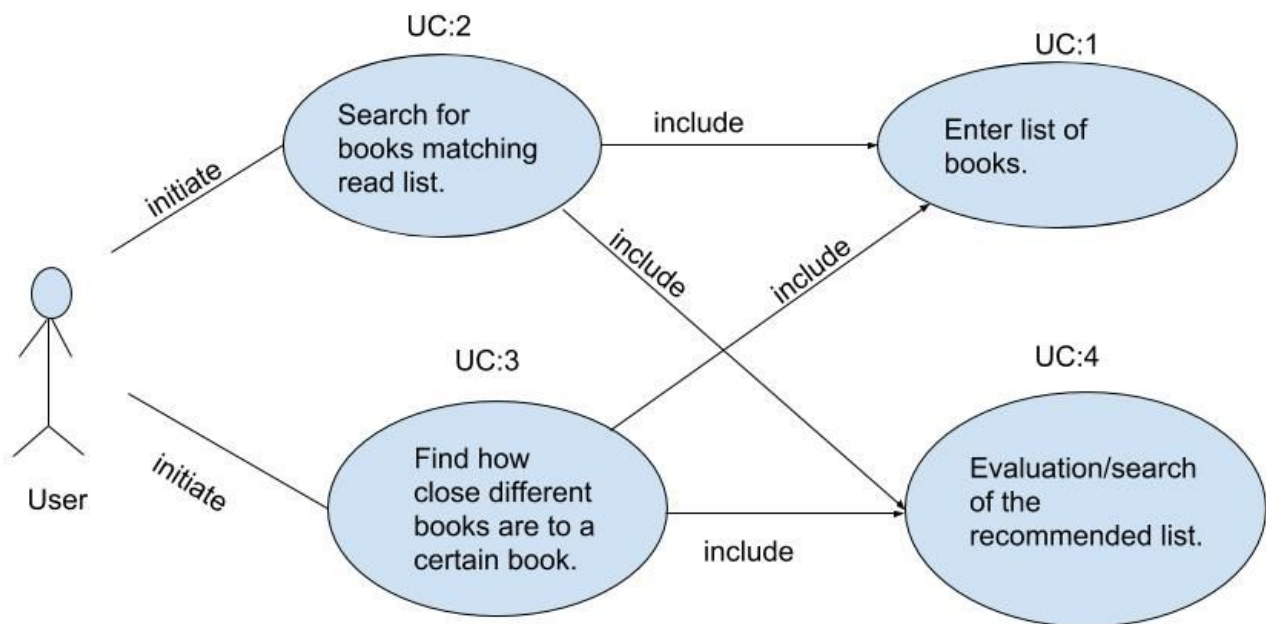
- a. User→ Enter the name of book A into the search bar
- b. System← begins to find the name of the book in the database for predictive text
- c. User→ Selects the book from the predictive text
- d. System← Creates table that stores all the books that the user input
- e. System← Starts running the similar match algorithm, and then stores them in the same file with the books that were entered
- f. System← Use Analysis Algorithm to rank the books form S→ C

- g. User → Use search function to find book B and C and see which book is more closely related to book A

4. Case 4 (Evaluate Recommendations)

- a. System ← Returns a list of books that are recommended based on what the user inputed
- b. User → Search/find books that they believe are good reads in their taste by scrolling through the list

Use Case Diagram:



Traceability Matrix

Traceability Matrix (For Enumerated Functional Requirements)

Requirements	Priority Weight (of Requirement)	UC-1	UC-2	UC-3	UC-4
REQ-1	2	✓	✓		✓
REQ-2	5		✓		
REQ-3	4		✓		✓
REQ-4	4	✓	✓		
REQ-5	5		✓	✓	✓
REQ-6	5		✓	✓	✓
REQ-7	1		✓	✓	
REQ-8	4	✓			
REQ-9	2	✓			
REQ-10	4			✓	✓
REQ-11	4			✓	✓
REQ-12	2				✓
REQ-13	4		✓	✓	✓

Max Priority Weight (of Use Case)	3	4	5	5
-----------------------------------	---	---	---	---

Use cases 3 and 4 are most definitely what makes our software unique compared to similar software available today, hence why they are given the highest weights. The specific detail of the use case 3 so essential to our idea is that it involves ranking the found books into tiers, and then comparing all the tiers to all the books who were entered. This specific tool is

what our software does, that no other current software considers, hence why it is one of the crucial and innovative pieces of our project.

Use case 4 is also one of the most important pieces of the software because it carries out the data from use case 3 to the user, and essentially displays to the user what our algorithms and findings would most recommend. Although use case 4 is typical with recommendation programs, it will function very specific and closely to how user case 3 will 'tell' it operate and display.

Use case 1 is the same type of implementation that can be found in practically all web searches, hence why it is important but not as specific to our design; and case 2 does what most recommendation software does when a user inputs single book to find recommendations for. Use cases 3 and 4 is what really sets our idea apart from what's currently available.

Fully Dressed Descriptions

Use Case 3: Search Prediction

Primary Actor: User

Goal: To narrow down the search results to the same type of book

Participating Actors: A computer/mobile device

Preconditions: Software has all popular search data and the necessary algorithm

Postconditions: The user has received a list of books with similar titles

Flow of Events:

1. User→ Enters Harry Potter into the search bar.
2. System← begins to find Harry Potter in the database for predictive text
3. User→ Selects Harry Potter from the predictive text
4. System← Creates table that stores all the books that the user input
5. System← All types of Harry Potter books regardless of the specific title are displayed on the user interface
6. System← Use Analysis Algorithm to rank the books from S→ C
7. User→ Use search function to find book B and C and see which book is more closely related to book A

Use Case 4: Evaluate Recommendations

Primary Actor: User

Goal: To find the desired book from the recommendations

Participating Actors: A computer/mobile device

Preconditions: A list of books with the similar titles are retrieved from the database and ready to be used

Postconditions: The user has found the desired book from the predicted searches

Flow of Events:

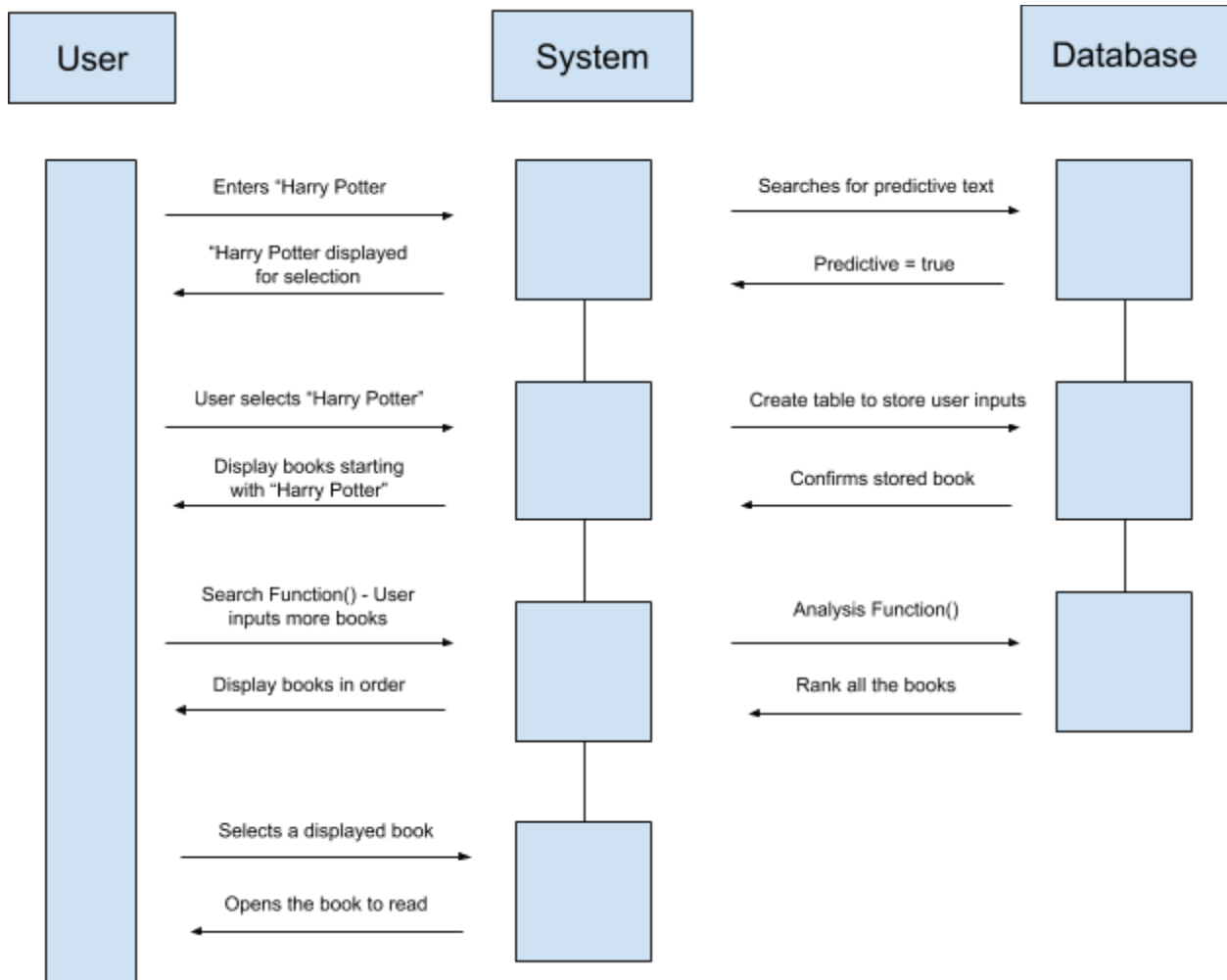
1. User→Wants *Harry Potter and the Sorcerer's Stone*
2. System← Performs algorithm and displays *Harry Potter and the Chamber of Secrets*, *Harry Potter and the Sorcerer's Stone*, *Harry Potter and the Goblet of Fire*. The system takes the key word Harry Potter and finds any book that starts with those keywords. Everything that appears after those keywords are irrelevant
3. User→ Selects *Harry Potter and the Sorcerer's Stone* since it is listed.
4. System← Opens the book for the user to read.

The two most important use cases, as shown from our traceability matrix, are use cases 3 and 4. In a nutshell, use case three takes all the data from use cases 1 and 2, then essentially creates the basis for which what our program will conclusively decide what will be recommended to the user. Use case 4 plays the role of communicating the results to the user and also giving a brief plot description of each of the options while also indicating which of the tiers each book falls under.

In order to describe use case 3 in detail, we must at first look at briefly describe what use cases 1 and 2 do. Use cases 1 and 2 for the most part store the user inputted books, and also creates a database for the books that are found through searching. These found books will then become the basis for which the program will rank (in terms of compatibility for the user based off of what the user likes)

System Sequence Diagrams

This is the system sequence diagram for User Cases 3 and 4:



User Interface Specification

Part A: Preliminary Design

First, the user enters the name of a book into the search bar.

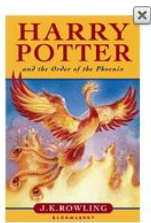


The query completion dropdown displays a list of available books.



The user then selects a book from the dropdown which is then displayed on the page.

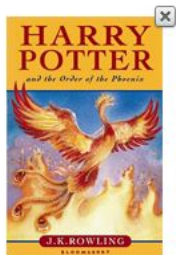
Book Recommendation



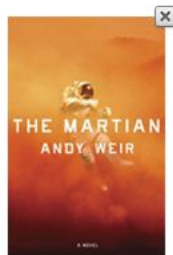
Harry Potter and the Order...

After selecting all the books they wish to use, the user clicks the search button for results.

Book Recommendation



Harry Potter and the Order...



The Martian



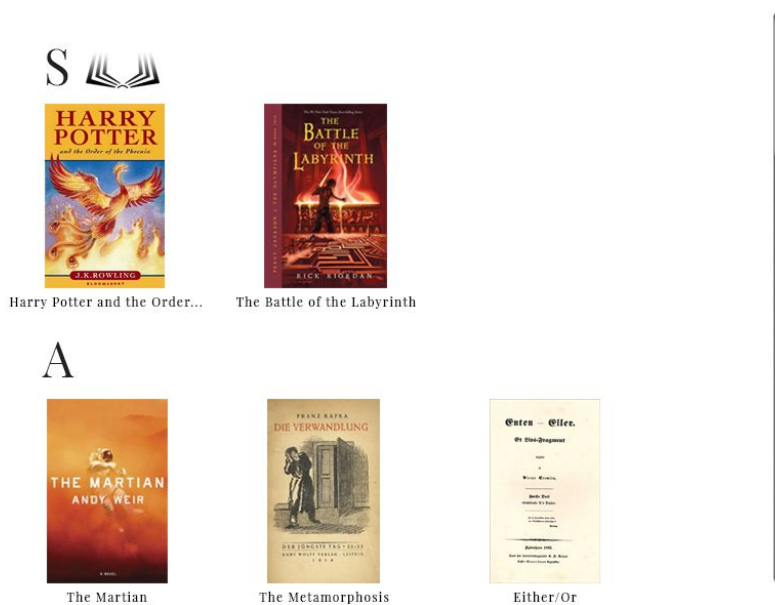
The Metamorphosis



Either/Or



Based on the user's entries, a page of recommendations is displayed, ranked from S to C.



Part B: User Effort Estimation

The scenario with the most usage is likely someone looking for books that explore the same themes and share the same qualities (author, genre, age, etc) as books they have enjoyed reading previously. In order for this scenario to be present, the user would have to enter the name of each book. Because we don't want them to enter the name of a book which we do not have in our database, the application will have a query completion dropdown menu from which they will have to select the book they wish to enter. This will decrease the number of keystrokes required to enter each book's title, and although it increases the number of mouse clicks (1 per book) it will decrease the number of inputs required overall.

After inputting all the books, one mouse click is required to execute the recommendation algorithm on the selected books. Therefore, between 5-10 keystrokes (estimate) and a mouse click may be required per book, and another mouse click to generate results. With 5 books being used to search for results, a total of 25-50 keystrokes and 6 mouse clicks are required. In the results page, after the search is completed, 1 mouse click is required to navigate to each book for information that the user wishes to view (not on our website). Of all of these inputs, none are required for user-interface navigation as the website has a search page and results page; rather all of the inputs are clerical data entry that is required for the algorithm to function.