



# BookWorm

Group 16

Avani Bhardwaj  
Vedanta Dhobley  
Akshat Shah  
Shazidul Islam  
Joel Cruz  
Kutay Kerimoglu  
Anthony Matos  
Alan Patel

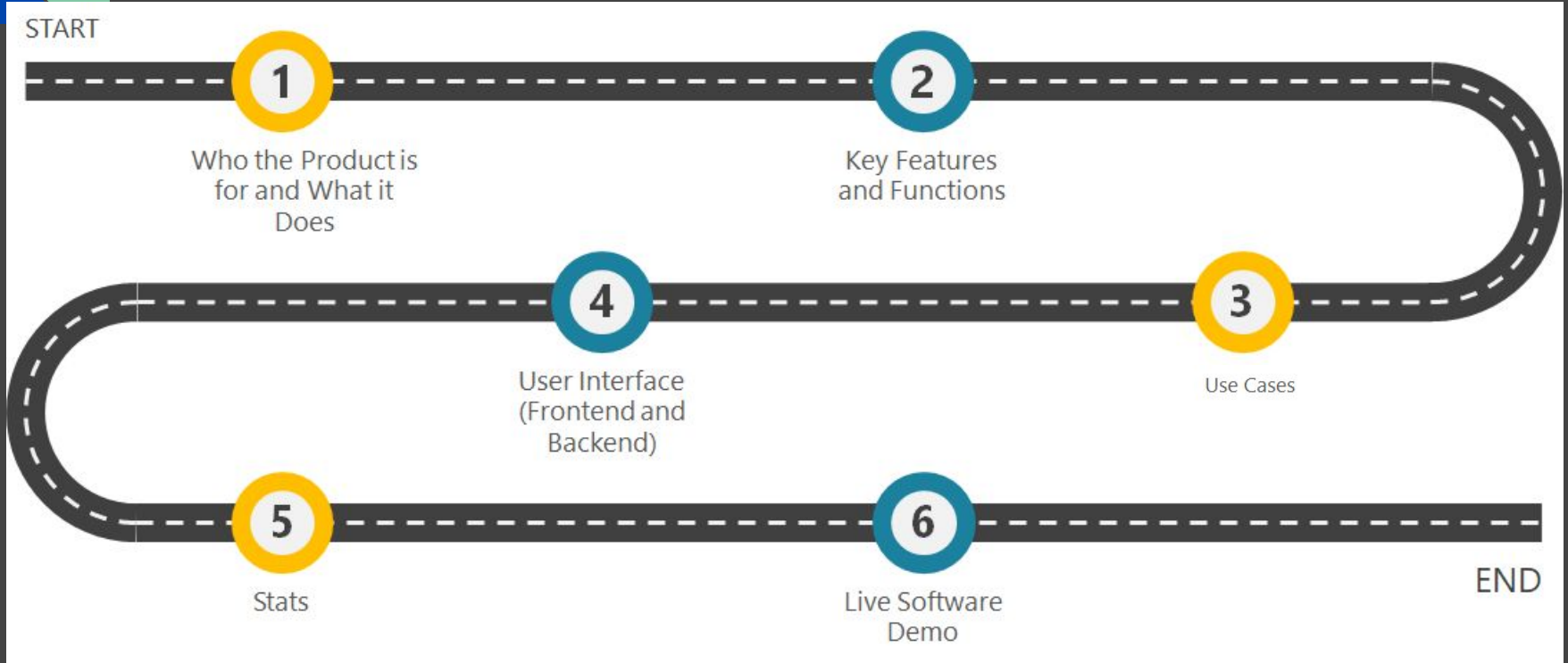
Demo 2

April 22, 2019

Github Link:

<https://github.com/vedantadhobley/SoftwareEngineeringProject2019>

# Roadmap





# Who is the product for? What is it for?

- A. Book fanatics
  - a. Book fanatics that wish to expand their library and find books similar to their taste
- B. You can use this site for course material as well
  - a. Instructors can use our product to create a lesson plan. This can be useful for Math, Science, Language, etc.
- C. This can be used by parents who want to have their child read specific types of books to enhance learning
  - a. Parents can search books that have similar learning properties as other books to introduce to their children
- D. The site is for all ages.
  - a. Our product will work for all ages. Based on our data and algorithm, we will be able to recommend ages specific books based on the users input

# Key features/functions:

## Manual Use:

- The user will manually load the site, and begin to type his/her book(s) name(s).
- He or she will manually select the book from the drop down menu.

## Automatic Features:

- Automatically, after the user begins to type in his or her book, the searching algorithm will run and display options for the user to choose from.
- The tier list will automatically show up after the user has selected his or her books.

## UI Design:

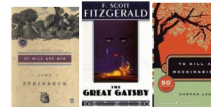


Tier S: The S tier represents books within the range of  $[95 \leq S \leq 100]\%$



Tier A: The A tier represents books within the range of  $[85 \leq A < 95]\%$

Tier B: The B tier represents books within the range of  $[60 \leq B < 85]\%$



Tier C: The C tier represents books within the range of  $[0 \leq C < 60]\%$





# Use Cases

## Step 1 (Entering the List of Books)

For this case, the user first begins typing the book into the search bar. The system responds by using predictive text to suggest the book's name. From there, the user selects the book that they are looking for from the dropdown menu that the system creates.

## Case 1 (Evaluate Recommendations)

For this case, the system returns a list of books that are recommended based on what the user has input. The user then searches or finds books that they want by scrolling through the list and evaluating which books have the most similar features.


## Case 3 (Find books according to their list of read books)

For this case, the user enters the books name into the search bar as previously done. The system responds by searching the database for predictive text. The user selects the book from the predictive text result. The system creates a table that stores all of the books that the user has input into the search bar. The system begins to run the similar match algorithm and stores them in the same file with the books that were previously entered. The system then runs the Analyzing Algorithm to rank the books in tiers from S to C



# User Interface, Frontend, Backend

- User Interface

- Simple
  - Local, Virtual Server
- 

- Frontend

- Predictive (in progress)

- Backend

- Fast  $\rightarrow O(1)$  search,  $O(n)$  storage
- Comprehensive  $\rightarrow$  10,000 book database (currently)

load all datasets

use TfidfVectorizer to convert authors and tag names into vectors

convert it to a matrix that holds all the values of the vectors

recommendation Algo

Find the cosine scores for each book

Add up the scores to have a list that contains the scores for each book and the book id

Sort the list to rank them from most similar to least similar

Tier System Algo

Take the first book and make it max limit

Take the 50th book and make that the lower bound

Get the difference in the scores and then divide them up into percentiles

Slice up the list to make sub groups

# Stats

Survey Results



■ Yes ■ Maybe ■ No

Total votes: 70

Yes: 53

Maybe: 14

No: 3

Disclaimer: People were not avid book readers and did not really know too much about the books.



# Software Demonstration