Team Meeting 2 February 6, 2019

Report 1

Deadline Website

<https://www.ece.rutgers.edu/~marsic/Teaching/SE/syllabus.html>

Report 1 Specifications:

<https://www.ece.rutgers.edu/~marsic/Teaching/SE/report1.html>

Team Splitting:

* Database (Shaz, Vedanta, Alan, Akshat)
  + Pulling information (how often we will update the library), Storing in one place, Organizing and cleaning so that it works with algorithm, result storage
* Frontend / User Interface (Anthony, Akshat, Joel, Avani)
  + Bootstrap, Book Autocompletion, Search, Output, Tier List display,
* Backend / Algorithm (Vedanta, Shaz, Alan, Joel, Akshat)
  + Tag Algorithm, Search, Tier Percentage Tile, Efficiency (time/space)
  + Point/Threshold System Employment
* Documentation (Avani and Kutay)
* Debugging (Kutay)

First Report – Part 1: (due Saturday 2/9/2019 @ 2:00pm latest)

1. Customer Problem Statement
   1. Problem Statement (Avani)
   2. Glossary of Terms (Kutay)
2. System Requirements (non-functional requirements describe how the system works, while functional requirements describe what the system should do)
   1. Enumerated Functional Requirements (Vedanta, Shaz)
      1. Search Function:
         1. Use a breadth for search for optimized time(Prior 2~3)
         2. The tags will be organized in a link list type of fashion that allows for easier more organized traversing. (Priority 5)
         3. Account for all distinct tags that associate with all the books entered (Priority 4)
      2. Multiple users:
         1. Create a new table every time a new user enters the list of books (New user just means a new list. Users are not recorded)(Prior 4)
         2. List should hold tags about the books (Prior 4)
         3. Spreadsheet will hold a list of books that have tags that are matching the distinct tags list(Prior 5)
         4. The list will include all books that have at least Age and Genre matching.(Prior 5)
         5. (Optional) Delete the last sheet to save memory allocation(Prior 1)
      3. Input Function:
         1. System will begin predicting user input(Prior 4)
         2. Update prediction with ever letter input(Prior 2)
      4. Analysis Function:
         1. The spreadsheet list will be organized via point system to rank them(Prior 4)
         2. Ranking will be broken via percentage tile of the highest point in that list 95%+ =S, 90%-94%=A, 80%-89% =B, 70%-80%=C (Prior 4)
         3. System will have a short description about the tier in the box located next to the letter(Prior 2)
   2. Enumerated Nonfunctional Requirements (Akshat, Alan)
      1. Database should be updated once week to account for new books and other updates (priority 3)
      2. System must be able to handle infinite amount of inputs and display finite amount of outputs for accurate representations. This should not take longer than 5 minutes (priority 5)
      3. System must display suggestions for spelling mistakes or if the inputs are not written exactly as the actual book name (even if few words are written, or should read 50 as the word “Fifty” in the book and vice-versa) (priority 4)
      4. System should display an error message for invalid inputs, even if there are few valid list of inputs followed by an invalid on (“Try again” button that refreshes the system.) (priority 5)
      5. System should prompt an option to start anew after displaying output (priority 5)
      6. System must be able output the recommended book using the algorithm in a quick fashion (Again limit is 5 min) (priority 5)
      7. Database is secured so the user cannot directly access it (priority 4)
      8. Program should be able to handle multiple clients accessing the program (priority 3)
      9. Application must be straightforward to use (priority 5)
      10. System code must be broken down into manageable and maintainable parts (priority 5)
      11. Must be able to run on web browsers without error (priority 5)
      12. Output should be displayed only after all desired inputs are submitted (priority 5)
   3. On-Screen Appearance Requirements (Anthony, Joel)
      1. The on-screen appearance aspect of our project is where most of the front-end design and development will be hosted. Along with the front-end work, this is also where the user interface aspects of our project will also reside. (Priority 5)
      2. The front-end design and development will drive how the visuals and website aesthetics will not only look, but also operate accordingly to the functional demands of our backend partition. All of the visuals of our website and user input will be held and driven through the on-screen requirements. (Priority 5)
      3. The on-screen appearance aspect of the project will also be where the user-interface will strive. So every input, whether it be from a input device or a smartphone screen, will be handled through our features of the user interface and will essentially allow the program to perform its job through user demands. (Priority 5)
      4. A list of books already read should be displayed horizontally (Priority 1).
      5. Below the list of books already read, a “Top Picks For You” section should be dedicated to all the recommended books based on the user’s interests (Priority 5).
      6. A trending section must list all books based on popularity and reviews (Priority 2).
      7. When scrolling through each book, there needs to be a brief summary of the book, the ratings, and it’s genre (Priority 5).
      8. Additional lists should have other genres for easy access to a book that’s not part of the user’s preferred genre. For example, such lists are comedy, horror, romance, drama etc. However, only a few genres should have their own list at this point. Otherwise, the user would have to scroll further down to see all genres which can ruin the user experience(Priority 4).
      9. A browse drop down menu should list all possible genres for users to further explore(Priority 2).
      10. In any list, the image of each book will be its cover so that the user can easily identify it (Priority 4).

Meeting Minutes: 60 minutes