CMPUT 291 Assignment 3 Design Document

Nomar Chavez
Adrian Leung
Nathan Vandermolen-Pater

March 7, 2019

Table of Contents

1. System overview and user guide

This system guides users, using a command line interface, through different tasks that parses enterprise data in a database and provides services to users. This design document will provide a brief overview of the layout of the system and will guide users through each individual tasks.

The system consisted of many different driving components, most notably, the libraries used. The libraries used in this system consists of pandas, matplotlib, and sqlite3. Pandas provide high-performance, easy to use data structures and data analysis tools. Pandas was used to analyze the data received from the Sqlite database. Matplotlib is a 2D plotting library that produces quality figures from data. Matplotlib was used to create charts of many forms. Sqlite3 is a small and versatile SQL database engine. Sqlite3 was used to interpret data from a database.

In order to be user and developer friendly, the data flow between each individual function is clean, well documented, and concise. Each individual functions were kept in individual files to allow easy editing, testing, and collaboration. Functions were then imported into the main program for further use. The menu is kept in the main program but isolated into its own function for easy editing and testing. The main function creates a Sqlite3 connection and cursor to the database, and passes these to each task function.

Before executing the program, ensure that the database is in the same directory as the program. To execute the program, run `python3 main.py <database>`, where `<database>` is the name of the database, in the command line. Example: `python3 main.py project.db`. Afterward, a menu will show with the following options:

```
=== Assignment 3 Project ===
(1) Show the email of all reviewers that have reviewed the paper
(2) Show all potential reviewers for that paper
(3) Find all reviewers whose number of reviews is in that range
(4) Show in how many sessions do authors participate in
(5) Top 5 most popular areas
(6) A bar chart of their average review scores for each category

(q) Quit

Enter command:
```

Using a keyboard, the user can choose from the options provided. Each individual functions will have their own implementation of this menu. Follow the instructions provided in each menu. For tasks that open a chart, the user can close the chart by clicking the ‘X’ in the top left or right (depending on operating system) corner of the window and return back to the menu.

2. Software design

2.1. Main function

The main function connects and disconnects the database, prints out a menu, and calls each individual functions. The path of the database is assumed to be in the same directory as the program. The function displays the menu with all possible options and accepts input from 1 to 6 and the letter ‘q’. All other inputs are rejected and the menu is displayed again. After quitting the menu, the main function commits any changes to the the database and then terminates the connection with the database before exiting the program.

2.2. Task 1

Task 1 displays a list of paper titles with a max of 5 titles per page. The function can take user inputs such as (n) next page, (p) previous page, (q) return to main function or a number assigned to the paper title. If the user inputs a number assigned to the paper title, the function then displays all the reviewers that have reviewed that paper.

2.3. Task 2

Task 2 displays a list of paper titles with a max of 5 titles per page. The function can take user inputs such as (n) next page, (p) previous page, (q) return to main function or a number assigned to the paper title. If the user inputs a number assigned to the paper title, the function then displays all potential reviewers for that page with expertise in the same area as the paper. (Potential reviewers are reviewers who have not reviewed that paper and have expertise in the same area as the paper)

2.4. Task 3

Task 3 takes as user input a range of integers, and outputs the email and number of reviews of each reviewer with number of reviews in that range.

2.5. Task 4

Task 4 implements two options to display the number of sessions each author participates in. To implement these, a menu is displayed upon calling task 4 which prompts the user to select an option.

> Option 1: Displays a bar plot of all authors and the number of sessions they participate in.

Option 2: Takes the email of an author as user input and outputs the number of sessions they participate in. Includes an option to display all authors to aid in selection.

Once an option is selected and its result outputted, the user is prompted to enter y/n for either ('y') entering another option to execute this task or ('n') finishing with the execution of this task.

2.6. Task 5

Task 5 creates a pie chart of the top 5 most popular areas based on the number of papers in each area. If there are less than 5 areas, the pie chart shows the number of papers per area. Initially, task 5 executes a query that returns the number of areas in the table areas and fetches a single row from the query into a variable called num_of_areas. Next, the function creates a data frame from another query that returns the name of each area and the number of papers per area. Afterward, the function will check if the condition holds for the number of areas and updates the range of the data frame if needed. Finally, the resulting data from the data frame is plotted on a pie chart and is displayed.

2.7. Task 6

Task 6 creates a grouped bar chart of each reviewer's average review scores for each category. Only those who reviewed papers are plotted in the grouped bar chart. The function creates a data frame from a query that returns the reviewer's email address and their average scores for each category by grouping each reviewer from the table reviews. Finally, the resulting data from the data frame is plotted on a grouped bar chart and is displayed.

3. Testing Strategy

All testing involved the sample database provided for Assignment 2. Testing for individual tasks was done as code was being written; this involved, in some cases, checking the output of SQL queries before outputting the result as a graph or chart. Once tasks were completed, testing was done using the sample database on each task individually; this involved manually inspecting the database contents for expected output of the task and running the task's code to ensure accuracy. When all tasks were completed and corroborated into the same program, final testing was done on each task to ensure accuracy.

4. Group work break-down

Nathan completed the code for tasks 3 and 4, conducted final testing on the completed project, created the layout for this design document and wrote sections 2.4, 2.5, and 3. Adrian completed the code for tasks 5 and 6, implemented the main function, menu, and dataflow between functions and the main function, and wrote sections 1, 2.1, 2.6, and 2.7. Nomar completed the code for tasks 1 and 2, and wrote sections 2.2 and 2.3. The group met three times, each for around an hour, to discuss the project and divide responsibilities. Each group member spent approximately 3 hours completing their portions.