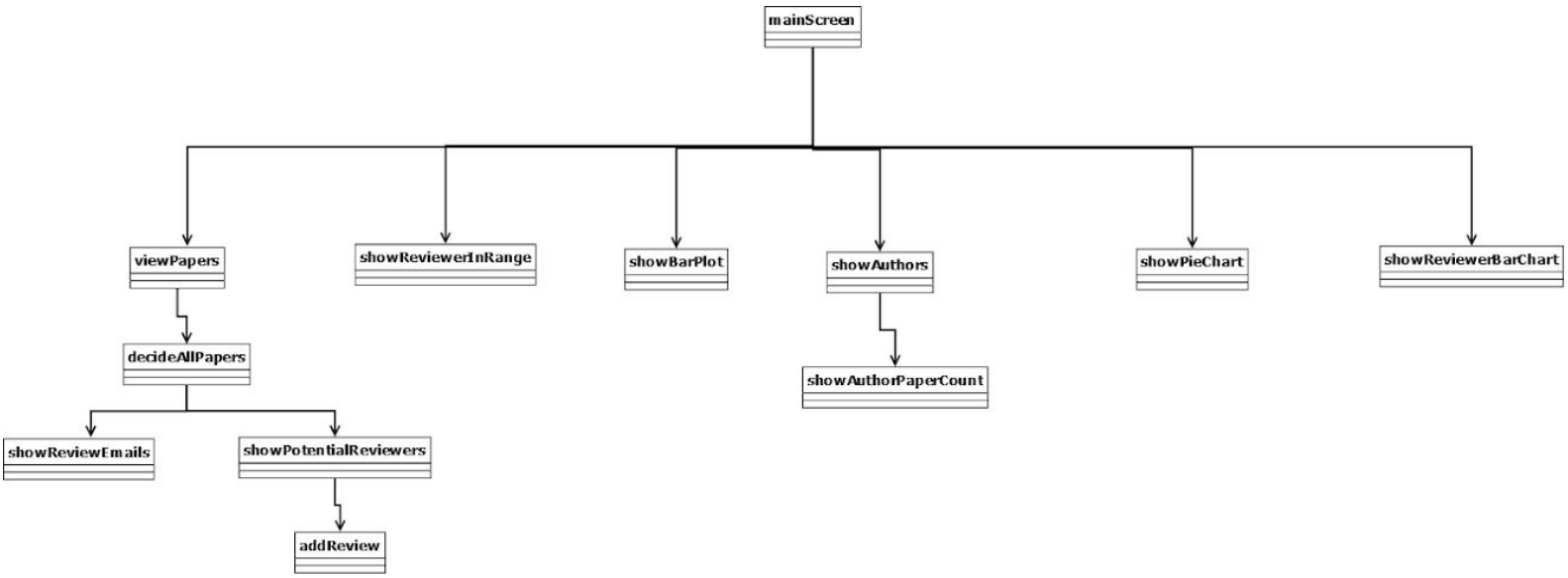
Overview and User Guide:

The application is organized into a series of “Views” following this pattern.



mainScreen:

- Serves as the home screen for the application, and presents the users with a list of all the functionality options of the program

viewPapers:

- Shows the user a list of all papers, and allows them to select one for further choices

decideAllPapers:

- Once the user has chosen a paper they are presented with the option to see all the reviewers that have reviewed that paper, or see a list of all potential reviewers for that paper

showReviewsEmails:

- Shows the user all the emails of the reviewers that have reviewed a paper

showPotentialReviewers:

- Shows a list of reviewers that are eligible to give a review for the current paper

addReview:

- Once a reviewer is selected, the user is prompted for the scores to create a new review

showReviewerInRange:

- The user is prompted to give an inclusive range, then they are presented with all the emails of the reviewers that have a total number of reviews within that range

showBarPlot:

- Creates and shows a bar plot of the number of sessions author has participated in

showAuthors:

- Shows a list of all authors and allows one to be selected

showAuthorPaperCount:

- Shows the number of accepted papers the selected author has

showPieChart:

- Creates and presents the user with a Pie chart of the top 5 most popular areas

showReviewerBarChart:

- Creates and presents a bar chart organized by reviewer of their average scores in each category

Software Design:

The design for this application is centered around the implementation of two classes. A UI class, whose job is to properly format and display the data to the user. And a Database class, whose responsibility is supplying the UI class with the correct data for each part of the UI.

Functions 1 and 2:

Function 1 and 2 were aggregated into the same flow pattern as they both require the user to see a list of all the papers in the database. Therefore initially a list of all the papers is fetched from the database, which are then formatted as a numbered list and waits for user input. Once the user picks a paper, they are presented with the option to execute function 1 or 2. If function 1 is picked, the emails of all reviewers that have reviewed that paper is fetched and presented as a list to the user, the user is then returned to the main screen. If function 2 is selected, the emails of all potential reviewers is fetched and presented to the user as a numbered list. Once the user chooses a reviewer, they are prompted to enter the scores for a new review. The scores are then inserted into the database as a new review for the selected reviewer on the originally selected paper

Function 3:

When function 3 is selected, the user is prompted to enter an inclusive range of integers. Then that range is used to fetch a list of reviewers whose total number of reviews in within the given range, and then presented as list to the user

Function 4:

Both parts of function 4 are presented as separate options in the main screen.
If part 1 is chosen. The number of accepted papers for each author is fetched into a pandas dataframe, and plotted as a bar chart, which is then presented to the user as a popup window displaying the bar chart.
If part 2 is chosen, a list of all authors is fetched and shown to the user, once the user picks an author, the number of accepted papers for that authors is fetched and shown to the user

Function 5:

If function 5 is selected, then the top 5 most popular areas are fetched into a pandas dataframe and plotted into a pie chart, which is then shown to the user as a popup window displaying the pie chart

Function 6:

If function 6 is selected the average scores in each field for all reviewers is fetched into a pandas dataframe and presented to the user in a popup window displaying the bar chart.

Testing Strategy:

Testing methods include programmed test cases and user simulation testing by the developer

Programmed test cases cover mainly any database accesses to ensure accuracy of SQL queries. Tests were formatted with expected outputs for each tested case, which were then compared to actual outputs and verified for equality.

Test included:

- testGetPaperReviewers()
    - Tests getPaperReviewers() returns the correct set of reviewers for each paper in the test database
- testGetAllPapers()
    - Tests that getAllPapers() returns all papers in the test database correctly
- testPotentialReviewers()
    - Tests getPotentialReviewers() returns the correct set of potential reviewers for each paper in the test database
- testInsert()
    - Gets all reviews from the database pre-insertion, then inserts a new review into the test database, then gets all reviews again and checks that the two list are not equal, verifying that the insert worked
- testGetRange()
    - Tests that getReviewersInRange() by checking a few select test cases, including the case of only reviewers with no reviews

User Simulation Testing:

Involved running the application and going through all potential user paths through the application, ensure no unexpected behaviour occurs.

Group Work Break-Down:

- Group work was mainly distributed based on point value for the assignment. Resulting in both partners doing 40 points worth of work

Brian:

- (10 marks) Show all papers* and allow one to be selected. Once a papers is selected, show all potential reviewers for that paper. Potential reviewers shown must have the same area of expertise as the paper. If reviewer has already reviewed the paper, they should not be able to review it again (either don't show them as a potential reviewer or give proper error once they try to input a review)
- (10 marks) Given a number range, find all reviewers whose number of reviews is in that range (the range should include the bounds)
- (2x 10 marks) Show in how many sessions do authors participate in? You must Implement two options: (1) a bar plot of all individual authors and how many sessions they participate in, and (2) just providing a number for a selected individual

Liam:

- (10 marks) Show all papers* and allow one to be selected. Once a paper is selected, show the email of all reviewers that have reviewed the paper.
- (15 marks) Create a pie chart of the top 5 most popular areas, popularity comes from the number of papers under the area. If there are less than 5 areas, show pie chart of however many areas that exist.
- (15 marks) For each reviewer, give a bar chart of their average review scores for each category. You must return a single grouped bar chart.