# **Team Members**

Xinran Tao,

Zain Raja,

Talal <add last name>,

Eamonn Mansour, 1836107

# **Introduction**

For this assignment, we were tasked with creating a GUI for exploring real-world properties available for temporary rent in London from AirBnB. The developed application features four panels that can be switched between, these include: A welcome panel, an interactive map panel, a statistics panel, and a User panel (details on each panel are discussed in the following section). Unit tests were also provided to test and verify that the functionality of the User panel is as intended. Since we were working as a team, we made use of the university’s Github Enterprise system to organise the application’s files, having divided the workload of the development process to each team member.

# **GUI Functionality**

This section will describe the functionality of each panel featured in the application’s final state.

The user is first greeted with the welcome panel. This panel allows the user to select a price range for the properties that they would like to view using two drop-down boxes, labelled “From” and “To” to minimize ambiguity. Without selecting a price range, the user cannot switch to another panel using the application’s navigation buttons shown at the bottom corners of the application window. If the user selects a “From” value that is greater than their chosen “To” value, the application will generate a warning dialog to inform the user that their selection is invalid, disabling the navigation buttons and resetting the selected price range values to null. Finally, the central area of the welcome panel displays the main welcome message along with a reactive price range label that displays the user’s currently selected price range.

The next panel is the map panel, where a visual representation of London’s boroughs is shown by various hexagonal buttons, each of which can be interacted with to display more information about the properties within the selected borough. This additional information is displayed in a new window that displays the name, price, number of reviews and minimum number of nights a person can stay per property, shown as a table. Furthermore, the newly opened window provides a drop-down box that allows the user to select how they would like the properties to be listed, either by number of reviews, price, or alphabetically by host name and a text field that allows the user to search for individual listings from the table.

When selecting a row in the table of listings, another window opens to display the information about the selected listing along with the ability to view this listing on a map. This window also gives the user the ability to navigate through the displayed listings using “Next” and “Previous” buttons, as well as a “Show on map” button that opens another window to display the chosen property’s location on a Google Map. This Google Map window can be interacted with and also allows the user to view directions from their location to the property’s location on the map.

Following that, the statistics panel allows the user to view different property statistics. This panel features four statistics boxes, each of which having navigational buttons to allow the user to switch between the various statistics. The base statistics include: Average number of reviews per property, total number of properties, the number of entire home and apartment properties, and the most expensive borough (additional statistics that were also implemented are discussed in the next section). The application prevents the user from displaying the same statistic on different statistics boxes by skipping the duplicate statistic content when navigating.

To allow the user to view all the statistics with greater ease, functionality has been provided for the user to click on a statistic in order to open a larger view of the selected statistic without opening a separate window. All the data represented in the statistics panel is based on the user’s selected price range.

# **Additional Statistics**

This section describes the additional statistics that were implemented as part of the statistics panel.

The first additional statistic is the number of properties in each borough displayed as a pie chart, where each borough has its own coloured section in the pie chart, and is appropriately labelled.

The next statistic is a bar graph representation of the room types in a borough. The user is able to select a borough that they would like to view the statistic on, before the statistic updates and displays the relevant bar graph. The bar graph displays the number of properties with the different room types.

Like the previous statistic, a bar graph for availability in each borough has also been implemented. This shows the user the number of properties with their relevant availabilities, within their selected borough (which can also be done using a drop-down box).

The final statistic is a comparison statistic that displays the common features of the top-rated listings. This shows the average price, the average minimum number of nights and the type of room of the top-rated listings.

# **Fourth Panel – User Panel**

This section describes the fourth, “User”, panel’s functionality.

The user panel allows users to create their own user account by providing a username and a password, both of which having restrictions on input length (e.g. between 5 and 15 characters). Having signed up or logged in, the user is presented with their “dashboard” which displays their account name, a list of favourite properties and a section of input fields for the creation of new listings.

The user is able to add properties to their “Favourites” list by selecting the “favourite” button when viewing a single listing on the map panel. If the user wants to remove a property from their favourites list, they can do so by simply selecting the property in the list and click on the “Delete” button under the list.

As well as adding favourite properties, users can create their own Airbnb listing and add that listing to the provided “airbnb-london” CSV file. This is done by filling in a set of text fields and selecting options from drop-down boxes on this panel.

# **Unit Testing**

This section describes the unit tests used on the *MapPanel* class. It was decided that the *MapPanel* should be tested to ensure that the link between the Java class and the FXML file is performed correctly without error.

The test class begins by creating the appropriate objects that will be used to run the various tests on within its *setUp* method. The following tests have been included in the test class:

* Null tests: Tests to ensure that various attributes are not null
* Size test: A test to ensure that the size of the *buttons* ArrayList has a size of 33 (since there are 33 borough buttons on the map)
* Exception tests: Tests to ensure that the *showViewInRange* method functions appropriately when passing valid and invalid parameters, executing without throwing any exceptions.

# **Known Bugs/Issues**

Due to time limitations, the following bugs/issues have remained present in the system’s final state.

TO BE COMPLETED