# Description of the GUI

The application we developed in response to the specification given to us consists of four graphical windows, with the vast majority of the application existing within the first window, called main window. The second window of the application is used to display a list of all available properties that are available within a borough, within the users given price range. The third window is used to see specific information about a selected property that has been selected by the user and the fourth and final window of the application allows a user to either create or login to an account that will allow the user to create their own property listings to be added to the market.

The main window is divided into four sub panels that can be used to organise different functionalities of the application. By default, the application always opens on the main window with the welcome panel activated. This panel provides the user with an introductory message instructing the user on how to use the application. Above this message are two combo buttons and a button. The two combo boxes allow the user to input both a minimum price bound, and an upper price bound. Once the user has selected these two values, they need to apply the upper and lower bounds by clicking the apply button. This button also enables the previous and next buttons.

Once the buttons are activated, the next and previous button allows the user change and navigate through the panels in the application in a cyclical manner. By this I mean, once the user has reached the last panel in the window, upon clicking the next button, the user is taken to the first panel in the window. This also works when reaching the first panel using the previous button. These buttons persist amongst all the panels in the application window.

The second panel in the window is the map window. This window is used to display a graphical representation of all the boroughs in the city of London. Each borough is represented by a button, that once clicked will open a new window. Each of these buttons will be coloured differently based on the number of properties that are available in that area. The darker the colour indicates a higher concentration of properties in a given borough whilst the lighter colours indicated a lower concentration of properties in a borough. If a borough has no properties at all, the button is disabled.

The third panel of the window is used to display statistical information about the information that is stored in the property database. This information is not in respect to a single borough, or property, but rather the entire collection of data that was provided. This panel consist of four labels and eight buttons. Each label is assigned two buttons that allow the user to scroll through the data that is displayed in the label. Each of the labels display distinct information about the data set. This application stores [NUMBER] of statistical information, [LIST OF INFORMATION]. Only four of these can be displayed at any one time.

The final panel of this window is used to implement the feature we had created for the challenge task laid out in the assignment specification. We decided that an application that was being used to view a list of available properties could also be used to add their own properties to the listing for the purpose of renting to patrons. The first step, we thought, in order to allow an owner to add their own property to the listing would be to allow them to log into an account, or if they didn’t yet have an account create one, that provided the user with a host ID and a window to import the information required. This panel consists of two labels, two buttons, one text input field and one password input field. The two labels indicate which of the text fields are used to input the username and password of the user. One text field masks the input from the user when entering the users password, preventing people from around the user from seeing their account password. One of the two buttons is used for logging into the users account. If the user doesn’t have an account, they are also able to click the create account button that takes them to a new panel that is not accessible from the next and previous buttons. The login button checks if the usr details are correct and then opens a new window where the user can create their property listing.

The panel that is used to create a new used account has four labels and four text input fields as well as two buttons. One button is used to create the user account and the other is to cancel account creation. The fours labels are used to describe the text fields and the text fields gather four key pieces of information. The users first name, surname, username and password. Once this information is gathered it is added to a database where an ID number is automatically appended.

The second window, which is opened by on of the map buttons that was mentioned when describing panel two, is used to select a property from a list of available properties based on the price range previously entered and to allow the user to organise these results based on two fields. Both of these functions are controlled by two combo boxes with an accompanying button. The first combo box is used to hold all the available properties and the second combo box is used to hold the available sorting options. When a sorting option is selected, the user must apply the sorting options which will re-organise the list of available properties. Once the user has selected the desired property, they must again click a button to view more details about the property. This function will open a new window.

The third window is used to display the information about the selected property. This is done through the use of various labels with the text being set to the value loaded from the data base. A single button also exists within this window that can be used to take the user to a google map representation of the property and its surrounding area.

# Unit Testing

The class chosen for the unit testing class is *SortingAirbnbLists* as it contains important calculations relating to the information displayed on the 3rd panel (Statistics) and also contains the most testable methods in the project. The purpose of Unit Testing is to break down the operation of classes into single or chains of method calls to see if the program is operating as intended. This has the benefit of allowing the programmer of seeing at a glance what methods stop working if any changes are made to the code.

The *SortingAirbnbListsTest* testing class is composed of 16 test methods that verify the values returned by *SortingAirbnbLists* class are those that are expected. The 2 *statisticsRefreshRange* tests check that the program can handle the user changing the price range and the displayed stats updating correctly after applying.

The following tests verify that each of the 8 method calls produce the correct values for *MostExpensiveBorough, AvailableProp, Home/Apartments, AverageReviews, AverageNights, HighestListing, PrivateRooms, SharedRoom.* The test class checks that when the price range is applied the program firstly returns a non null value, then proceeds to verify the return values using known values for each statistic.

The final tests check the strings returned by the *getString* methods from 1-8 which are called by the statistics box in the *MainWindow* class. This testing is done using a similar method as the one used for the other statistics by checking for non null and then testing returned strings containing the calculated values.

# Description of the Extension task.

For our extension task, my group and I believed that the application could be used, not only to view properties to rent, but to also allow a person to add their own properties to the database to be rented as well. For this task to be completed we identified several perquisites. First would be the ability to isolate users into their own accounts to prevent one user from creating a listing in another user’s name. As such, we needed a way to allow a user to log into their own respective account. We added a fourth panel to the window which would allow a user to either log into their own account or to create one.

If the user had chosen to create their own account, a window would open presenting them with several input boxes allowing them to input the required details for their account. All this information would be added to a data base (a .csv file in our group project) where a user ID would automatically be appended.

In order for the user to log into an account, they must enter their username and password. If their details are not found in the data base, or are not correct, the program will reject their permission to enter their own account. When the user successfully logs into an account, a new window will appear.

This new window will allow a user to input the name of the property, the neighbourhood it resides in and its corresponding longitude and latitude co-ordinates, the price of the property, the minimum duration of the stay as well as how many days it is available in a year. Once the user submits this information, several other pieces of information are automatically added to the listing before entering the data base. This information includes the host name and ID as well as the total number of reviews and the most latest review, which are both automatically set to zero and empty respectively.

On this window, the user also has the ability to log out of their account. This closes the current window and returns the user to the login screen.