KING'S College LONDON

4CCS1PPA

Programming Practice and Applications

<u>Report of Assignment 4</u>

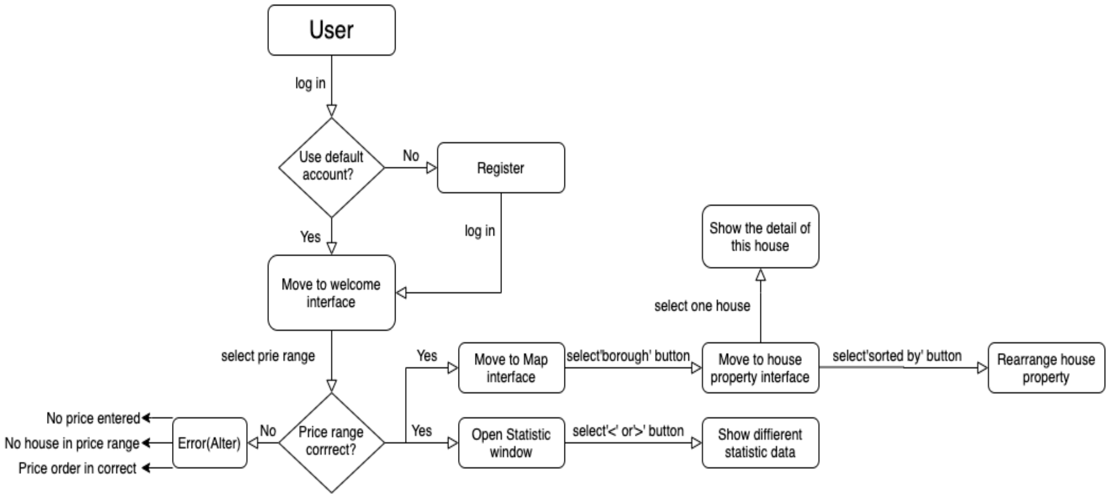
Author:
Haolun Shan (K1923585), Houkang Zhang (K1924824)
Yeshuai Cui (K1924221), Zhenyu Li (K1924125)

2020

26th, March

Introduction

This is a optimized GUI for exploring properties that are available for temporary rental in London. First the user logs in to the GUI through a registered account, and then they can check the houses in the various boroughs by selecting different price ranges. There will be a map showing the number of available houses in the borough by the color depth. At the same time, the system will pop up a new window to display some data characteristics of the houses in the selected price range. In addition, the attributes of all available houses in the current town will be displayed when the user clicks the town button on the map. Here is a rough flowchart：



Foundation of GUI

1. **Registration foundation**. It is a particularly important part in the entire system. Any house rental website has its own user registration system. Ours has no exception. Users need to log into their accounts before browsing the house. Users can log in by using default account accompany with password or they can register their own account. These will be stored in a csv file controlled by ***AccountDataManagement*** class.

2. **Welcome Panel**. This is basic task: Panel 1 Welcome. After the user logs in to their account, the interface will display a “ Welcome ! ” with a pink background. There is a price selection button in the upper right corner of this interface, users can select the items in the list or they can enter the price as

they want. After making a price selection, clicking the button **go** will move to the map interface and open a new statistics data window.
When button **go** is pressed, the class **PopertyFilter** can create an HashMap, In this HashMap, the key is neighborhood name and the value is an ArrayList. Inside this ArralyList, there are all the property in the selected price range and theirs neighborhood name is same as the key;

2.1 Error when selecting a wrong price range. In class **MainViewController**, methods **InvalidInPutAlter()**, **PriceRangeEmptyAlter()** and **AlertNoPrice()**, When the from price and to price = 0, from price>=to price will trigger the above method and trigger **Alert alert = new Alert(AlertType.ERROR)** which give a window with error.

3. **Map Panel.** This is the basic task, Panel 2 The map. After the user decides the price range, the interface will move to the map. There are 33 hexagon buttons according to the HashMap returned from **PopertyFilter** each button represents one neighborhood of properties. And we use to indicate the number of available properties, lighter the color is, more number of the houses. If this neighborhood has no properties in the given price range, the corresponding button will not be shown in the map and not set the click action.

3.1 How to determine the color of a borough button. In the class **MapController**, the method **update()**, there is a **while loop** to calculate the remainder to determine if the color is deepened. In the same area, use the method **setbackgroud()** of JavaFX : **BackgroundFill bgf = new BackgroundFill(Color.color**) to give a color to the button.

The user clicks on a different borough to move to the specific attributes of all the houses in that borough. The properties are host name, price, number of reviews and minimum rent time. There is also a select button on this interface to sort by name, price and review from bottom to top.

3.2 How to complete the sort. C**,** the method **sorting(),** use the method **Comparator() : compare(AirbnbListing one , AirbnbListing two).** There is a @Override means we can use a new completed method in the original method.

3.3 When one of the properties is clicked, all the descriptions of the property appear in a new window. The class **ListController** can create lots of GridPands and each GridPan show the basic information of on the property. This GridPan is linked to the whole information by a HashMap called **airbnbs.** The key is this GridPane itself and the value is the

AirbnblListing which store all the information of this property. When clicking the GridPan, all the information about the corresponding property will be shown on a new window.

Additional statistic

4. **Statistics Panel.** This is basic task: Panel 3 Statistic. After the user selects the price range, a new window will pop up, which is the data statistics window. These statistics are average view per property, avaliable property, number of private room and most expensive property price. In addition, users can switch to browse statistics of other properties by using the buttons next to the properties panel.

Fourth Panel

Our fourth panel is also our challenge task：user registration system. As described at the beginning, new users need to register an account before logging in to the system. Registration of an account requires your nickname, username(email) and password. These required data will have certain restrictions, such as the length of your nickname can not exceed 10 characters. This account information will be recorded and not stored in the system, so you cannot register for an account with the same nickname, just like we usually register for an account. All the account you created will be recorded even when you end the program.

Unit Test

In this code, we test the class: **PropertyFilter**, which has the method to calculate all the properties in a price range. In the method **getProperties()** using for loop to find all the **neighbourhood** of property when price is less than max price and more than min price, and add these properties in a HashMap.

For test,we create a test class: **filter_test** ,we use **size1,size2,size3,size4** to define the number of properties in different price combinations. And when these variables are less than or equal to 33, the result is true, otherwise it is false.

For bugs

Until now, we have checked the program lots of time and solve all the bugs we found, we hope there is no bug exist anymore.

Problem

submission of k1924824 have been split into to two different account, one is k1924824 and the other one is houkang zhang.