4CCS1PPA

Assignment 4: London property marketplace

Overview

Before starting this assignment, you should ensure you have completed the tasks in the Requirement 0 document on KEATS. You should be in a group of four, and have the group set up on TeamFeedback. You should have a working Git repository linked with TeamFeedback, with each member of the team having made a branch, which they have then merged.

The deadline for coursework is **Friday 30th March 5pm.**

Domain Description

Your task is to develop a GUI for exploring properties that are avaiable for temporary rental in London.

The Dataset

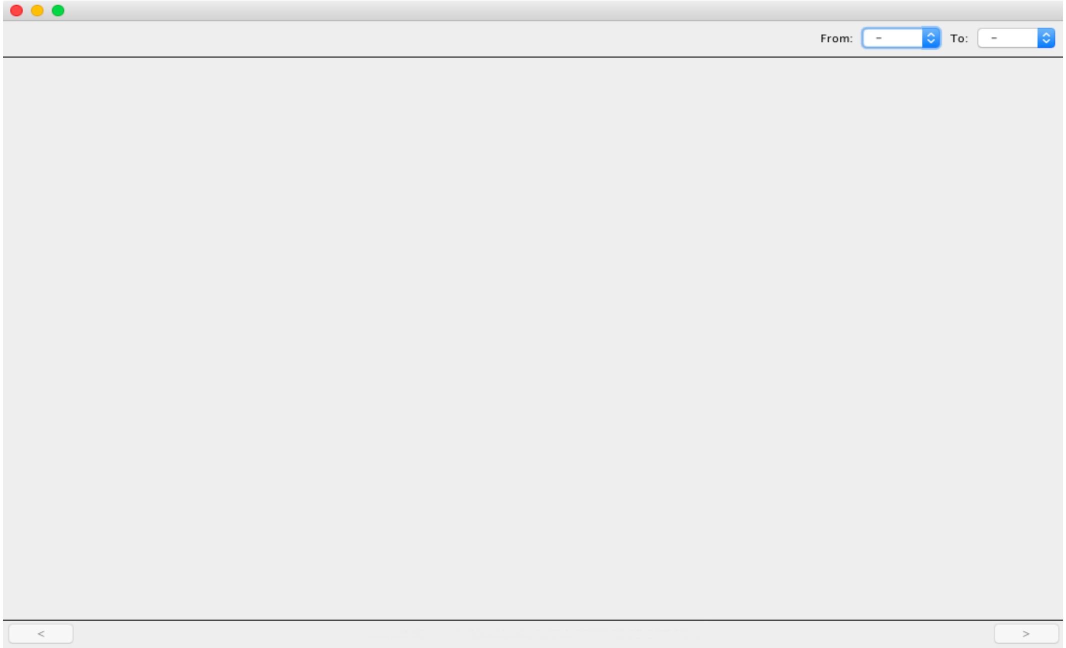
You will be working on a realworld dataset of the London property market, from AirBnB. This data is provided to you through existing Java classes that are available from the KEATS page. You should take some time to familiarise yourself with the methods that you have access to through these classes.

Base Tasks

You should aim to complete all core tasks. They are as follows.

- **Application window**
  The application is designed to process and display the data from the API in a digestible form for a user. To do this, we will create a multi-panel and multi-window application.

  The first, and main, window of this application should look something similar to the following:

  The window is designed to hold a series of different panels, each of which contain data relating to the retrieval and display of data from the API. Details of what these panels should contain is given in some of the remaining scetions of this brief.

  Key behaviour that should be offered by this frame itself is as follows:
  - There should be the ability to move left and right through the panels contained in the centre of the frame using "back" and "forward" buttons in appropriate positions. Naturally, each of these buttons will only work if there are still additional panels to access in the selected direction.

  - The top right of the frame should feature two drop-down boxes, appropriately labelled, allowing a user to select a price range for the properties they want to see statistics about.

  - The user should be alerted if they have selected a price range that is invalid, such as the "from" price they have selected is greater than the "end" price.

  - The "back" and "forward" buttons should be initially disabled, until the user has selected a price range. This is because the other frames available are going to process and display the data loaded when a user selects a price range, and are thus initially empty. The first panel, however, is unconnected to the data, and should thus be shown to the user when the frame first loads, as discussed in the next section.

- **Panel 1: Welcome**

  Having a user land on a blank screen like the one shown in the image above isn't too good in respect of usability.

  Instead, the first panel loaded into the Main window when a user loads the application should welcome to the application, and give them some instructions on its basic use.

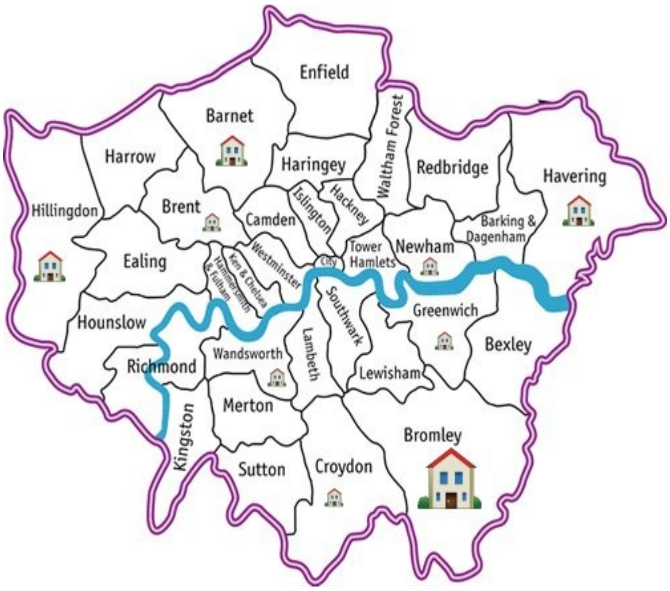
  Once a price range is selected by the user, the first window should additionally show which price range is currently selected.

- **Panel 2: Map**

  The first data related panel that should become available for access by the user once they have selected a date range, and the application has loaded this data, is a panel that visually demonstrates the content of the dataset to the user.

  This should occur by showing the user a map of London on the panel, with an appropriately sized property graphic (such as a house symbol) placed over a neighbourhood to indicate that properties are available in the neighbourhood, and the size of the graphic should correspond to the number of properties available in the neighbourhood. We will refer to these as markers. Please find and select an appropriate property graphics, and a map of London, onto which copies of the markers are to be placed.

  You'll need to think carefully about how the graphics are sized over each neighbourhood. We want some way to size each graphic relative to the others, using size to indicat when one neighbourhood has more properties available than another. We also don't want our graphics to be too small or too big.



If a neighbourhood has at least one property, then a marker should appear within the bounds of that neighbourhood on the image, in order to demonstrate this.

We will not be too concerned with where the marker appears within the neighbourhood boundaries, but it should appear be clear to which neighbourhood each graphic pertains.

The size of the marker should depend on the relative number of available properties in that neighbourhood. For example, in the graphic above, Bromley has a large number of properties available, Croydon has fewer, and Sutton has none.

<u>List of properties:</u>

Upon a user selecting a price range, and the appropriate markers being appended to the map of London, it should be possible for a user to click any of these markers in order to learn more about the properties in the neighbourhood.

This information should be presented in a new window, which pops up when each marker is clicked.

The title of the window should include the full name of the neighbourhood that the user has selected.

The window should display a simple list of available properties with the following details:

- Name of the host of the property
- Price of the property
- Number of reviews per property
- Minimum number of nights that someone can stay

The window should have an additional drop-down menu, that allows the user to select whether to sort the list by number of reviews, price, or alphabetically by host name. Making a selection from the drop-down menu should reorder the list accordingly.

When one of the properties in the list is clicked, the description of the property should appear in a new window.

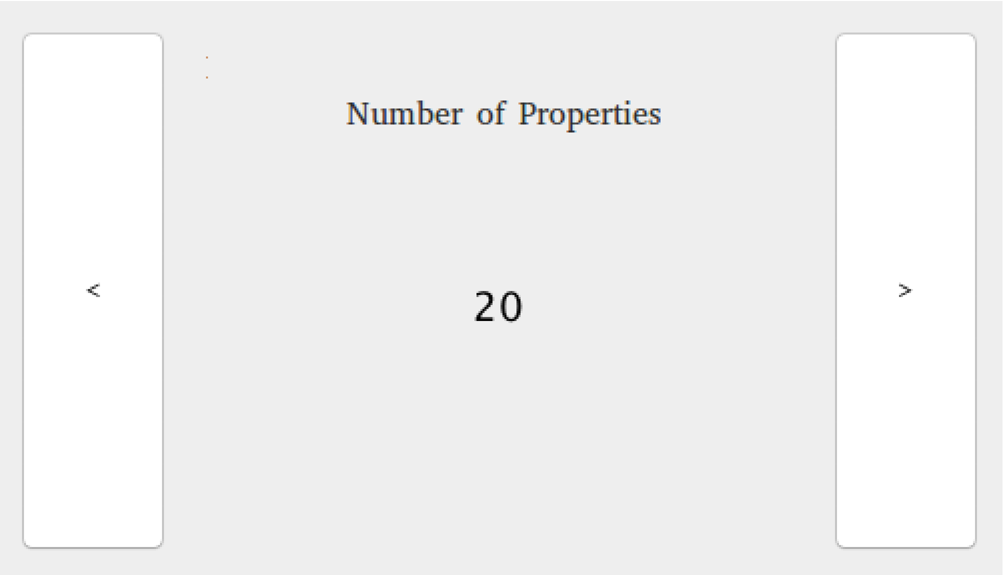
- **Panel 3: Statistics**

This panel will present a series of statistics, based on the information derived from the dataset, and should thus be avaiable at the same time as the map to be observed by the user, should they scroll to this panel.

We will derive eight statistics over the data avaiable.

Panel Behaviour

The panel should be separated into four distinct sections, which we will refer to as statistic boxes. Therefore, at any given time, only four of the available eight statistics are shown. Each statistic box should look similar to the following, with each box displaying a different statistic:



Using the buttons shown, the user should be able to click between the different available statistics, in a similar manner to the way in which they are able to click between the different panels of the main window.

At no point should the same statistic appear twice on the panel (as this would confuse the user), instead, when the panel is first shown, four different statistics should be selected. Moreover, when the user is clicking between the different statistics, it should not be possible for them to configure the panel such that it shows the same statistic in more than one box.

Statistics

The base statistics you need to implement are as follows:

- Average number of reviews per property.
- Total number of available properties.
- The number of entire homes and appartments (as opposed to private rooms).
- The priciest neighbourhood. The priciest neighbourhood should take into account the minimum number of nights.

You should also implement four additional statistics that a user can scroll to in any of the statistic boxes. The statistics should be signitifantly different to those above.

Each group member should take responsibility for coordinating the invention and development of each statistic. Your Github branches will be checked to ensure that the majority of the development work behind each statistic is undertaken by one group member.

A prize will be awarded to the team that derives the most insightful additional four statistics from the dataset.

- **Unit testing**

  You should provide suitable unit tests for one of the classes in your project. Your testing should be thorough and appropriate.

  You should not select one of the classes that were provided by us. You should pick a class of your own that is complex enough to warrant significant testing, e.g. one that provides some core functionality of your project.

Challenge Tasks

Once you have completed the base tasks, you can make a start with the fourth panel.

- **Panel 4: Surprise us!**
  The final panel is your blank canvas, designed for you to make your program do something additional and interesting, that it doesn't do already. This can include launching new windows, if you would like it to.

  Marks for this section will be awarded liberally, so be as creative or reserved as your like, but make suer your functional addition is evident.

  Only one functional addition is required per group.

  A prize will be awarded to the team that implementes the most interesting, inventive, or amusing "Surprise Us!" feature. Towards the end of the assignment, we will provide details of extra drop-in lab sessions to get additional support for your fourth panel.

Marking Criterea

Normally, each member of a group will receive the same mark. However, each team member’s contribution will be monitored through TeamFeedback, both via peer assesment, and via analysis of the content of Git commits of each member. If any team member is found to not be doing their fair share, they will be subject to a mark capping; the exact level of the capping will be decided on a case by case basis.

You must submit your code and a report. The code will be assessed for:

- Correctness
- Appropriate use of language constructs
- Style (commenting, indentation, etc.)
- Design (Code quality, appopriate use of inheritance)
- Difficulty (marks for extension task)

The report should include:

- A description of your GUI, including the functionalities providedd by the GUI.
- A description of the extension task.
- Known bugs or problems (Note: for a bug in your code that you document yourself, you may not lose many marks – maybe none, if it is part of the challenge task. For bugs that we find that you did not document you will probably lose marks).

The report should be no more than four pages long.

The code and report must be submitted via the Assignment 4 submission link on the PPA KEATS page, before the deadline, by **all members of your group**. Both the source code and the report should clearly state the names of all authors.