

Group Report

Coursework 4

April 7th, 2021

Group Name: KTAB

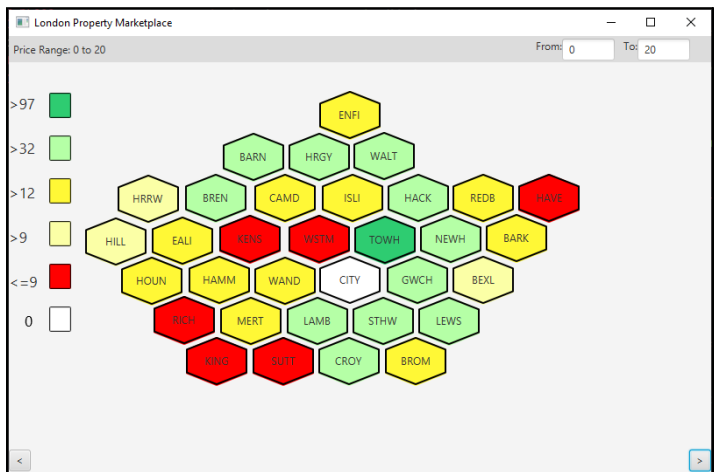
GUI Description

The London Property Marketplace program serves as a booking tool to show people different properties in London along with information about their boroughs. When the application is launched, the user is greeted with a home screen that shows instructions on how to use the application. The program consists of panels that the user can navigate back and forth through by clicking on the two buttons at the bottom of the window.

At the top of the welcome window, there are two text boxes that allow the user to enter a budget for the properties they would like to see. If a user enters data that is invalid, an error message will show. Once a range has been selected, the user will be able to navigate to the next panel.

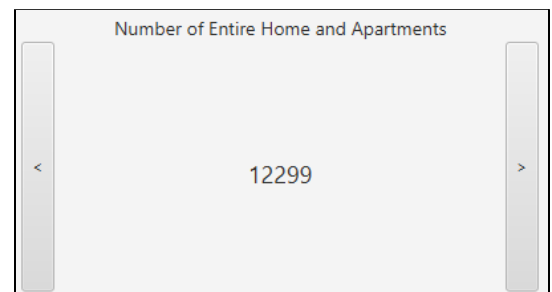
The second panel contains a map of London where each borough is represented by a hexagon. The colors of the hexagons indicate how many properties within the selected price range are in that borough. A key on the left of the panel shows the scale of the colours. The scale adjusts based on the number of properties in the price range.

When a borough hexagon is clicked on, a new window opens and shows different statistics, as well as the available properties in that borough.



This information provides the users with insight about the borough which can aid them in choosing where they would like to stay. On the right side of the window a list of properties in that borough is displayed, along with information about each property. The user can select a category (using the dropdown on the bottom left) they wish to sort the list of properties by, either number of reviews, price, or the name of the host.

The third panel is the statistics panel. This panel shows different statistics about the properties that are part of the user's selected price range in addition to statistics about the boroughs of London. The panel contains 4 different boxes that each display one stat. The user can use the buttons in each box to click between the different available statistics. **Note:** The GUI description and functionality provided by the fourth panel is described in depth on page 3



Additional Statistics

Safest Borough

A second dataset called 'london-boroughs.csv' is used for this statistic. This file contains a 'profile' for each borough in London that consists of various metrics such as population, area, crime rate, etc. In order to determine which borough is the safest, three metrics are considered (each measured in /1000s per population) and a weighted average is calculated. The metrics include: crime rate (60%), fire rate (25%), ambulance incidents (15%). Once the weighted average is computed and turned into a safety score, the borough with the lowest score is deemed the safest borough.

$$Score = (0.6 * crime\ rate) + (0.25 * fire\ rate) + (0.15 * ambulance\ incidents)$$

Best Property for Sightseeing

A third dataset called 'london-attractions.csv' is used for this statistic. The file contains the names of various different tourist attractions in London and their latitude and longitude coordinates. The statistic will show the id of the property that is closest (lowest average distance to all attractions) to the attractions. The distance used to compare properties with each other is the average distance in kilometers between the property and each attraction. The Haversine formula (seen below) was used in order to accurately calculate the distance between two coordinates.

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

Environmental Indicators

Using the same borough-profiles dataset, this set of 6 statistics displays the borough that ranks best in various environmental metrics. The metrics include: carbon emissions, recycling rate, cycling rate, public transport accessibility, green space area and number of cars. For each metric, a ranking is generated where the highest ranked borough is the one that has the best score for that metric.

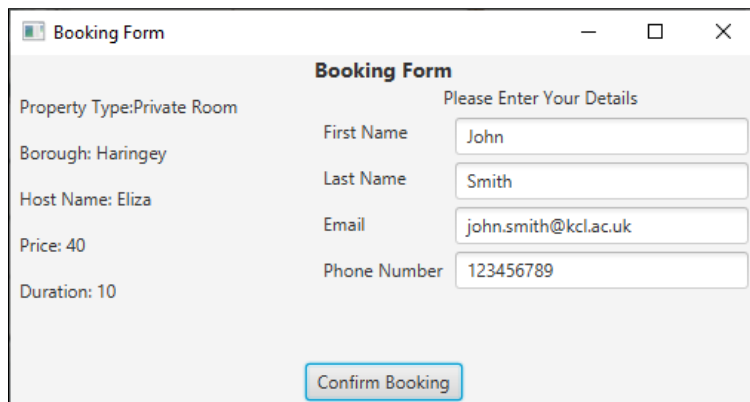
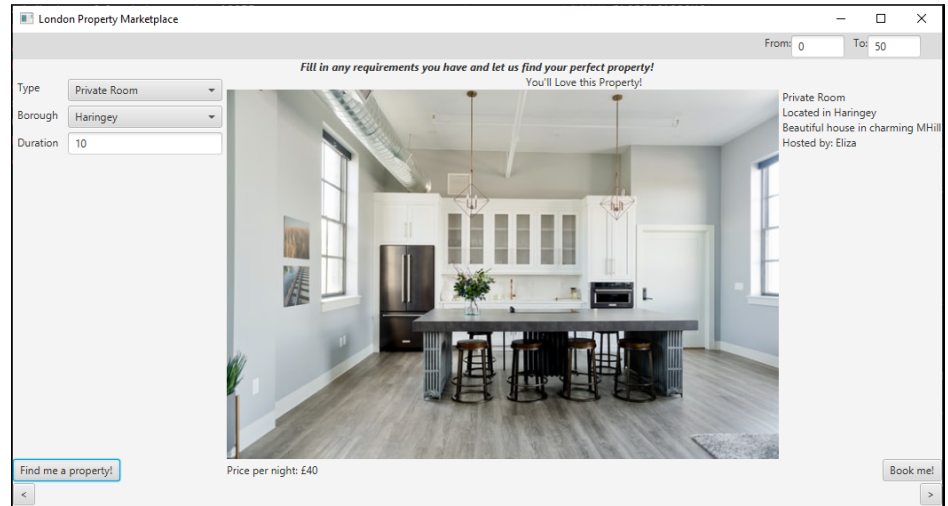
Greenest Borough

Using the individual rankings from the individual environmental indicators, an average ranking for each borough is computed by taking the sum of the borough's rank for each indicator and dividing it by the number of indicators. Once the average rank for each borough is calculated, the borough with the lowest average rank is considered the greenest/best environmental borough.

Fourth Panel

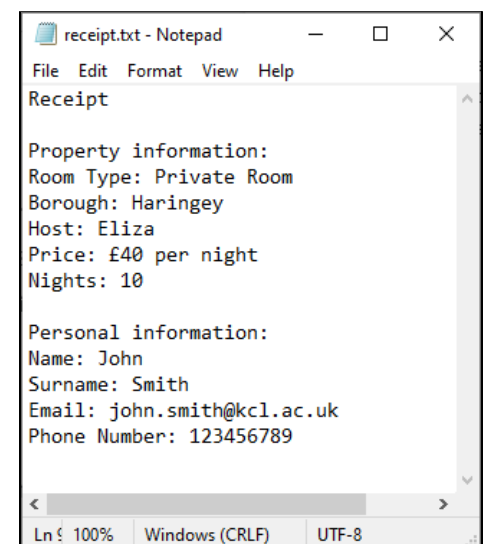
The fourth panel serves as a tool that finds a property that best suits the user's needs and then allows them to book a stay at that property. Using the inputs on the left of the screen, the user can select the type of property, borough, and duration for their stay. The user can then click on the 'Find me a property!' button and a property will be found. Information about the property and an image of the property (not a real image but a stock image) is displayed to the user.

If the property does not suit the user's needs, they can change any of the requirements and a new property will be found and displayed.



Once satisfied with the chosen property, the user can click on the 'Book me!' button in order to book a stay at that property. A new window will open with a booking form where the user can enter their name, email address, and phone number. The user has to ensure that their inputs are valid (e.g. email has an @ symbol, etc.) otherwise they will not be able to confirm their booking. At the bottom of the form there is a button that can be clicked to confirm the booking.

Once a booking is confirmed, a booking will be created and the user will receive a receipt of their booking. The program automatically creates a new text file and writes the details about the booking to that file. The txt file can be found in the same directory as the program's .jar file and is named 'propertyID-receipt.txt', where propertyID is the id of the property that has been booked. The receipt includes information about the property for the booking in addition to the user's personal information.



Unit Tests (use latest version of BlueJ to run tests)

Unit tests were performed on the 'Statistics Generator' class, which is responsible for creating instances of all other statistics-related classes and forming a list of all statistics that will be displayed to the user on the statistics panel. This class is integral to the statistics panel, as it combines a lot of different classes in one place, and therefore testing it properly was of significant importance.

The unit tests check that the objects for each statistic-related class (BoroughSafetyAnalyzer, GreenBoroughAnalyzer, AttractionDataAnalyzer) are created successfully and are not null. There are also tests in place that ensure that the properties and boroughProfiles lists are both passed through the constructors of other classes. There are also tests that check that the list of statistics is generated successfully without any missing or null elements.

Additionally, there are tests in place in order to check the value that is computed by the borough analyzer classes is equal to the expected value. For example, the borough in the dataset with the lowest carbon emissions is Barking and Dagenham and therefore the expected value for that stat is Barking and Dagenham.

Known Bugs or Problems

One minor problem has to do with the color scale on the map panel. When a small number of properties is chosen as part of the price range, the scale will show 0 for multiple colors.

When resizing certain panels, the buttons at the bottom of the screen will not resize properly and they will disappear. The user can adjust the window again in order to make the buttons reappear.