[redacted]

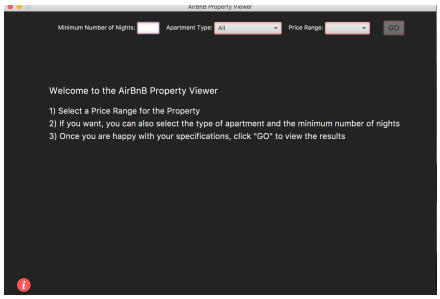
London Property Marketplace

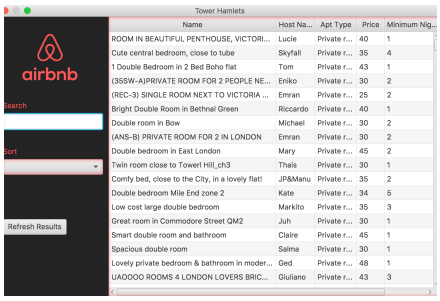
This is an app created to explore available properties inside London using the property dataset of Airbnb. It's designed to be easy to use, and powerful in the data it can display. Creating the piece of software used all the unique talents of Team Hummus, and we split the work evenly between us through GitHub and peer programming at Bush House.

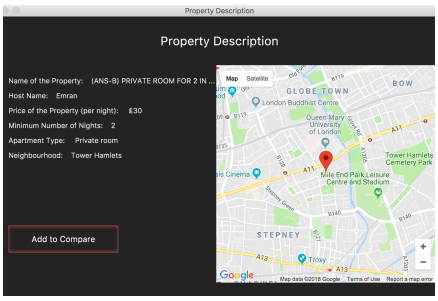
The GUI carries a modern design with "matte" black backgrounds, bright pink elements and minimalism. Since the application is designed to be for Airbnb, the official "Airbnb pink" was used for text and borders to adhere to company design standards. The base idea of the GUI is to disguise the complexity of the backend behind a deceptively simple user interface.

“Software is now so complex - requiring so many gazillions of tiny files all over your computer - that most consumers don’t want to bother to know what’s really going on”

-Clive Thompson







Welcome Panel

This panel is the first thing a user sees. It has three search options a user can use to specify the type of property they want before pressing the “GO” button to leave the panel. Two of the sorting options are optional, a user only needs to specify the price range. This panel also has an additional “info” button in the bottom corner to open the “About us” window.

Info & Alert Panels

These two are simple panels, with no buttons or elements offering interactivity, serving one purpose each. The info box shows an “About us” section with the names of the team and Alert Box is designed to pop up if the user entered incorrect values into the welcome panel search.

Statistics Panel

The statistics panel consists of four distinct sections that are designed to display eight statistics created from the 54,000 properties within the Airbnb dataset. With forward and back buttons, a user can cycle through the data and future programmers can easily add more statistics without having to create new panels. We also satisfied the criteria where no two statistics appear in a section twice. The current eight statistics display the following:

- The number of Entire Home/Apartments
- The priciest neighborhood

- The least expensive shared room
- The least expensive private room
- The most expensive Entire Home/Apartment
- The number of available properties (based on the user specification in the welcome panel)
- The average number of reviews per property
- The closest properties near the Big Ben

Compare Panel

The compare panel is the Panel 4 extension task for our assignment. Once the users have opened up the description of the property from the Property List Panel, they have an option to add the property to a compare list. This compare list allows the user to short list selected properties in order to make a more informed decision before booking a property. In the compare panel, there is a TableView which lists out the properties which the user has selected. The user even has an option to delete properties from the list if he wishes to.

Property List View Panel

We have used TableView to list out all the properties which satisfy the user specifications. There is a choice box on the side panel which allows the user to sort the results in any order they want and there is a search bar which searches through the list for what the user wants.

Property Description Panel

This Panel is designed to show extra information about one specific property. One **challenge task** we implemented ourselves here was the addition of Google maps inside our app. To achieve this we imported the Google maps API so a user can see the location of a property on the map of London with a pin dropped on there.

Extension Panel (Compare Panel)

We realized a user could view 54,000 properties in our software but never be able to compare them. For a user to be well informed before they select a property for their accommodation, we felt it was necessary for them to have a “Compare Panel”. Each time a user is viewing a property inside “Property Description”. This panel is capable of storing as many properties as possible inside a table view.

Code Design

To make an app that adhered to modern software design we used JavaFX instead of Swing to write our app. This included three different types of files that create different aspects of the app. FXML and CSS files help the design the look of the app with opacity, mouse hovering events, borders that glow and even editing effects into images for a better aesthetic design. The Java files were used to code the functionality of the app and were linked with the FXML files. This creates the front and back end of our application with a simplistic user interface in the front-end FXML and CSS files, and complex powerful programming code in the back-end Java files. Due to the nature of JavaFX we were not allowed to use inheritance within our program as that is not possible with JavaFX.