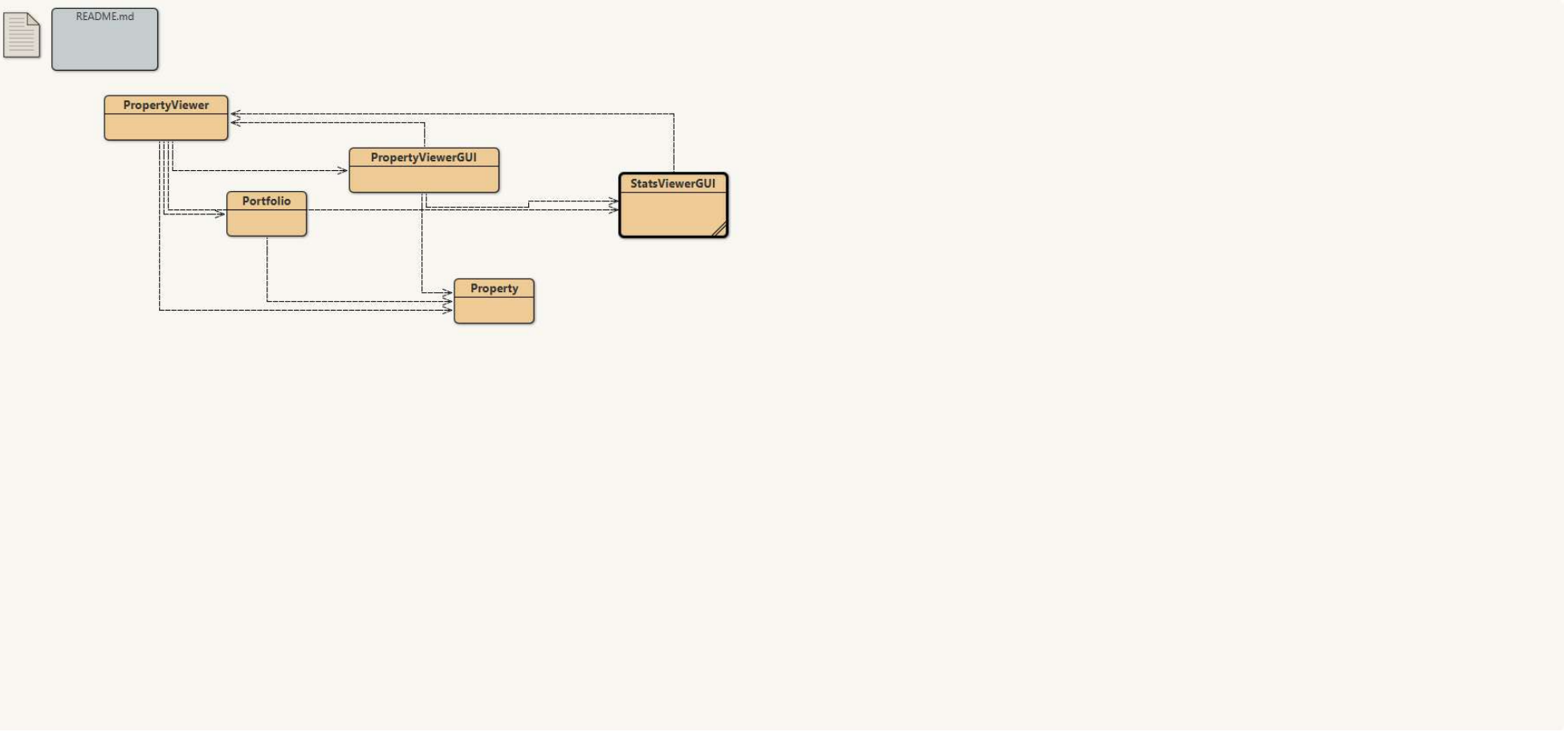
README.md
PropertyViewer
PropertyViewerGUI
StatsViewerGUI
Portfolio
Property

```
import java.awt.Desktop;
import java.net.URI;
import java.net.URISyntaxException;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
/**
 *
 *
 * Makes an Object for showing a graphical representation of the statistics.
 * @author Ronen Raj Roy (K21086768)
 */
public class StatsViewerGUI
{
    private JLabel NumOfProp_Label;
    private JLabel AvgPrice_Label;
    private PropertyViewer viewer;
    private JFrame frame;
    private boolean is_open = false;
    public StatsViewerGUI(PropertyViewer viewer)
    {
        this.viewer = viewer;
    }
    /** Actual function to make the jframe and display the statistics.
     *  Code referenced from the makeFrame() function from the PropertyViewerGUI
class authored by Michael Kölling, David J Barnes, and Josh Murphy.
     */
    public void showStats()
        {    if(is_open)
            {
                frame.toFront();
            }
            else{
                    is_open = true;
                    frame = new JFrame("Statistics");
                    JPanel contentPane = (JPanel)frame.getContentPane();
                    contentPane.setBorder(new EmptyBorder(6, 6, 6, 6));

                    contentPane.setLayout(new BorderLayout(6, 6));
                    NumOfProp_Label = new JLabel("default"); // Label is used to
show the Number of properties viewed by the user till now.
                    contentPane.add(NumOfProp_Label, BorderLayout.NORTH);
                    NumOfProp_Label.setFont(new Font("Arial", Font.BOLD, 26));

                    AvgPrice_Label = new JLabel(" ");   // Label is used to show
the Average Price of the properties viewed by the user until now.
                    contentPane.add(AvgPrice_Label, BorderLayout.SOUTH);
                    AvgPrice_Label.setFont(new Font("Arial", Font.BOLD, 26));

                    frame.setPreferredSize(new Dimension(610,135));

                    // building is done - arrange the components
                    frame.pack();
```

```
                    // place the frame at the center of the screen and show
                    Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
                    frame.setLocation(d.width/2 - frame.getWidth()/2, d.height/2 -
frame.getHeight()/2);
                    frame.setVisible(true);

                    ActionListener action = new
ActionListener(){
      //make an action listener to update the statistical values
                    public void actionPerformed(ActionEvent e)
                    {
                        NumOfProp_Label.setText("Number of Properties Viewed :" +
viewer.getNumberOfPropertiesViewed());
                        AvgPrice_Label.setText("Average Cost of the Properties
Viewed : " + viewer.averagePropertyPrice());
                    }
                };

                    Timer timer = new Timer(100,action);        //using a swing
timer to constantly update the JLabels every 100ms to keep up with the user
looking for new properties.
                    timer.setInitialDelay(0);                   //referenced from
https://docs.oracle.com/javase/tutorial/uiswing/misc/timer.html
                    timer.start();
                }
    }
}
```

```java
//Name - Ronen Raj Roy
//Student ID - K21086768
import java.awt.Desktop;
import java.net.URI;
import java.net.URISyntaxException;

/**
 * This project implements a simple application. Properties from a fixed
 * file can be displayed.
 *
 *
 * @author Michael Kölling and Josh Murphy,edited by Ronen Raj Roy (K21086768)
 * @version 1.0
 */
public class PropertyViewer
{
    private PropertyViewerGUI gui;     // the Graphical User Interface
    private Portfolio portfolio;
    private int property_no;            // stores the property number which is
currently being viewed by the user.
    private int number_of_properties;   // stores the number of properties viewed
by the user.
    private Property current_property;  // stores the property currently being
viewed by the user
    private int total_sum;              // stores the sum of the amounts of the
properties that the user views.
    private StatsViewerGUI stats_gui;
    /**
     * Create a PropertyViewer and display its GUI on screen.
     */
    public PropertyViewer()
    {
        number_of_properties = 1;   //stores the number of properties viewed.
        property_no = 0;            //stores current property number.
        gui = new PropertyViewerGUI(this);
        portfolio = new Portfolio("airbnb-london.csv");
        stats_gui = new StatsViewerGUI(this);
        displayProperty(property_no);
        total_sum = total_sum + current_property.getPrice();  //stores the price
of the first property viewed(we don't want to miss out on that!)
    }

    /**
     * Function used to display new Properties when the user
     * wants to go next or back.
     */
    public void displayProperty(int prop_no)
    {
        current_property = (portfolio).getProperty(prop_no);
        gui.showProperty(current_property);                    //calling all the
gui functions to display the property info on the window.
        gui.showID(current_property);
        gui.showFavourite(current_property);
    }
    /**
```

```
     *  Function which is used to move next in the airbnb list with a rollover
included so that it goes to the first property after clicking next on the last
property.
     */
    public void nextProperty()
    {
        property_no = (property_no + 1) % (portfolio.numberOfProperties());
        number_of_properties++;

displayProperty(property_no);

        total_sum = total_sum + (current_property).getPrice();
    }

    /**
     * Function which is used to move back in the airbnb list with a rollover
included so that it goes to the last property after clicking previous on the first
property.
     */
    public void previousProperty()
    {
        property_no = (property_no - 1) % (portfolio.numberOfProperties());
        number_of_properties++;
        displayProperty(property_no);
        total_sum = total_sum + (current_property).getPrice();
    }

    /**
     *  Function used to set the current property as the user's favourite.
     */
    public void toggleFavourite()
    {
        current_property.toggleFavourite();
        displayProperty(property_no);
    }


    //----- methods for challenge tasks -----

    /**
     * This method opens the system's default internet browser
     * The Google maps page should show the current properties location on the
map.
     */
    public void viewMap() throws Exception
    {
       double latitude = current_property.getLatitude();    // gets the latitude
of the current property being showed.
       double longitude = current_property.getLongitude();  // gets the longitiude
of the current property being showerd

       URI uri = new URI("https://www.google.com/maps/place/" + latitude + "," +
longitude);
       java.awt.Desktop.getDesktop().browse(uri);
    }
```