|  |
| --- |
| Abdulahi Sharif (22013666), Bartosz Glowacki (23010447), Muhammad Chikhoun (23013604), Saad Abdeen (23093538)  25/3/2024 |

|  |
| --- |
|  |
| London Covid19 Statistics |
| PPA – CW4 Covid Data GUI |

# Introduction

As a group we decided as the program requires four panels, and the group is made up of four people, we should each take responsibility of one panel. The remaining tasks were then split up and allocated depending on the workload so everyone makes a fair contribution.

# Tasks

## Base Tasks

### Panel 1: Welcome/Application Window (Abdulahi Sharif)

Initially, we created a superclass called "baseGUI," of which all of our other panels would be subclasses. The buttons used to cycle through the various panels and the date pickers at the top of the screen were added in this class. After developing the datepickers, we included ways to trigger an alert in the event that a user submits an incorrect date. The controller class's "updateData" method is called when the user chooses a valid date range, altering the data that is sent to the other classes and, ultimately, the user sees. The back and next buttons are disabled when the user first launches the application, but they become active again after the user provides a date.

Then, a "VBox" and many labels with information about the application are used to construct the welcome panel. The bottom label updates to reflect the date range the user has chosen as they submit a date range.

### Panel 2 : Map (Muhammad Chikhoun)

In order to create the map for the program. We initially experimented with a variety of approaches. This included the use of scene builder, and different approaches to how to design the map, in the end we agreed on the hexagonal pattern suggested in the task sheet. The pattern was created using a loop to avoid having to create each individual button separately. They were placed in a regular pane as it allows you to place nodes using coordinates so its easier to create the geographically accurate representation of London.

Once the buttons were made and placed. The data to be presented in the window once the button is pressed was created using a hash maps. The data from the controller class within the selected date range is placed in a hash map and the data for the required borough is isolated in a separate hash map. The new window displays the information in a table format using “CovidData” as the object type of the table. This allowed the use of “PropertyValueFactory” to call the respective get statements for the “CovidData” class to populate the table with the correct data. The data is sorted by date initially and as required, can be adjusted to be sorted for the other categories.

Lastly, in order to represent the total deaths of each borough, we used a choropleth map approach. The total deaths of each borough is calculated by subtracting the total deaths at the start date from the end date. The total deaths are then shown proportionally using opacity to adjust the red colour of the buttons. A larger opacity (stronger red colour) shows more total deaths and vice versa.

### Panel 3: Statistics (Saad Abdeen)

### Unit Testing

## Challenge Tasks

### Panel 4: Analytics (Bartosz Glowacki)

For the challenge task we created a panel with a simple simulation. To run the simulation a correct date must be declared and one of the London Boroughs must be chosen. After hitting the play button the Thread runs and the simulation runs in the background. After hitting stop button, running through 100 days, changing the panel, the date or borough the simulation stops. Each progress bar in the centre starts with the initial value of 1000 people and each day adjusts their values according to Google Mobility Data in the given dataset. Simulation runs from the start date chosen in the tab above. After choosing the borough, under the bars, the chart should appear. On it there are two lines: black one corresponding to the new number of deaths each day in the chosen borough and red one corresponding to the number of new infections each day in the chosen borough.

# Conclusion