GROUP 30 - REPORT

Oskar Bzymek - Screens, Buttons, Sliders.

Most of my work related to the project was firstly creating working screens and buttons which can help display our project. I would obtain the ready graph classes and implement them into the program by providing them with separate screens and working buttons that can scroll through the information. I also implemented other useful tools like a slider for the displayed data list which was created by Claire, or a working switch which enabled us to switch between the "graph" screen and the "heatmap" screen.

In the first few weeks of our project I merged the first few additions such as our first bar chart created by Ernest, or our first SQLite prototype provided by Claire. Luckily in the last weeks of the project Claire helped me with the merging and we managed to make all the data come together much faster. The biggest problem I ran into during merging was obtaining the libraries and code that was overriding functions, although we had some difficulties we managed to solve these problems as a team.
Most of the buttons I created work as an array list of widgets which the screen creates. The slider is an extension of the widget class and is created on the same basis.

Ernesto Ortiz - Barchart visualisation

I contributated by creating a class named BarChart that took in any arraylist representing the x axis and any arraylist representing the y axis and created a proportionally sized barchart that would fit in any screen. The barchart included all labels that can be seen in the final code.

I also initially made a list class that was omitted from the final code as I decided to focus on the barChart class. I thought of ways in how the raw data could be represented in form as a barchart. I created new classes to convert the raw data into two arraylists that the barChart class could work with.

This classes were, flightsPerState, distancePerAirline and flightsPerAirport. These all were able to successfully represent the full data set in a barchart. Finally I helped merge this with the final code so it could fit in our UI by making the barchart size customisable and therefore scrollable.

Claire McCooey - Backend and flight list

My contribution to the team came largely in the form of creating a more efficient backend for the application. We initially stored flight data in a CSV file then loaded it into an ArrayList of DataPoint objects. This was functional but very slow. Luckily, I had previous experience of working with SQLite. I knew it would be a perfect fit for the project as it is quick to set up, efficient, and portable. I created a SQLite database using Dylan’s pre-processed data and set up a SQL query generation system based on user input. I’m quite proud of this system, but we ended up dropping it because it didn’t interface well with the graphical outputs. I also created the flight list section of the program. Using the ControlP5 library’s ScrollableList and CheckBox controls, I created a quick interface that allows the user to select which fields they want to view and which parameter to sort the results by. On clicking a button, the program fetches the first 2000 records from the database and displays the selected fields in the specified order. Trying to get the displayed list to be scrollable was a big issue we faced. Eventually Oskar and I managed to come up with a solution; we created a PGraphics object with the records displayed in it, then drew that to the screen with its y-coordinate linked to the y-coordinate of Oskar’s Slider object. I also helped with general bug fixing and merge issues. None of us had really used version control systems before, so finding our way around Git and GitHub was a challenge. Luckily we persevered and got our program in working order!

Dylan Gallagher: Data Pre-processing and Heatmap Visualization

As part of my contribution to the team, I was responsible for data pre-processing, utilising Python and the pandas library to enhance the original dataset. I downloaded the dataset again from the source and added a flight delay column, providing more insights for the visualisations. Additionally, I was in charge of the heatmap visualisation, which required obtaining a dataset of US airports with latitude and longitude columns. I used pandas to pre-process the dataset to ensure compatibility with our program.

To accurately represent the geographical locations of airports, I obtained a map of the states with exact latitude and longitude boundaries using a Geospatial Imaging Software (QGIS). This allowed me to precisely place each airport in the correct pixel position by implementing simple scaling functions built into Java.

Additionally, by employing HashMaps within both the Heatmap class and the main program, I significantly improved its performance, making the rendering of heatmaps faster and more efficient. This optimization was essential in providing a seamless user experience, especially when working with large datasets, as it allowed the heatmap visualisation to update quickly and respond to user input without noticeable delays.