Programming Project Report

Introduction

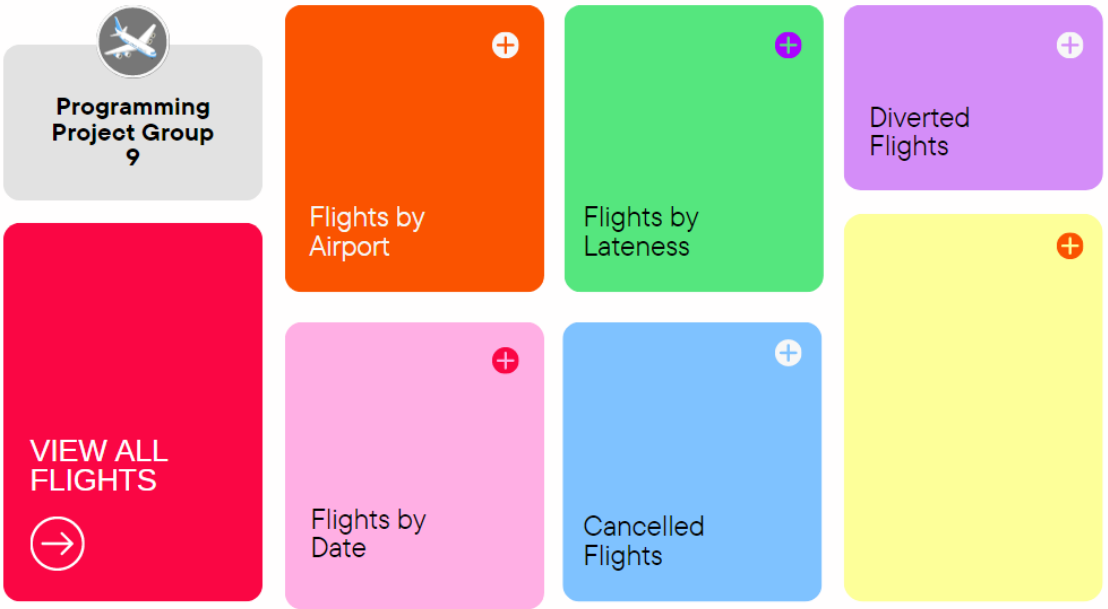
In today's world, efficient handling and presentation of data have become crucial for businesses and organisations. As part of our Programming Project, our team has developed an application that reads data from a CSV file and presents it in a user-friendly interface. This report outlines the goals, design, implementation, and evaluation of our application, highlighting its features and functionality.

The primary objective of our project was to create an application that simplifies the process of reading and displaying data from a CSV file, providing a seamless experience for users who may not have technical expertise. We utilised different libraries in order to build an application that offers an intuitive and visually appealing user interface, making it easy for users to navigate and interact with the data.

Investigating and Planning

Our team followed a systematic approach to investigate and plan for the development of our CSV data reading and display application for flights. Here are the key steps we took:

1. Defining the project scope: We agreed on the goals, requirements and limitations of our project, focusing on reading and displaying the data taking into account the user's query.
2. Brainstorming features: We generated ideas for features that would enhance the usability and functionality of our application, such as data filtering, sorting, and visualisation.
3. Planning user interface: We designed a visually appealing and intuitive user interface that would allow easy navigation and interaction with the data, incorporating features like data table display, interactive filtering, and visual representations.



4. Defining the project timeline: We established a project timeline with deadlines and assigned tasks to team members, ensuring clear role and responsibilities.

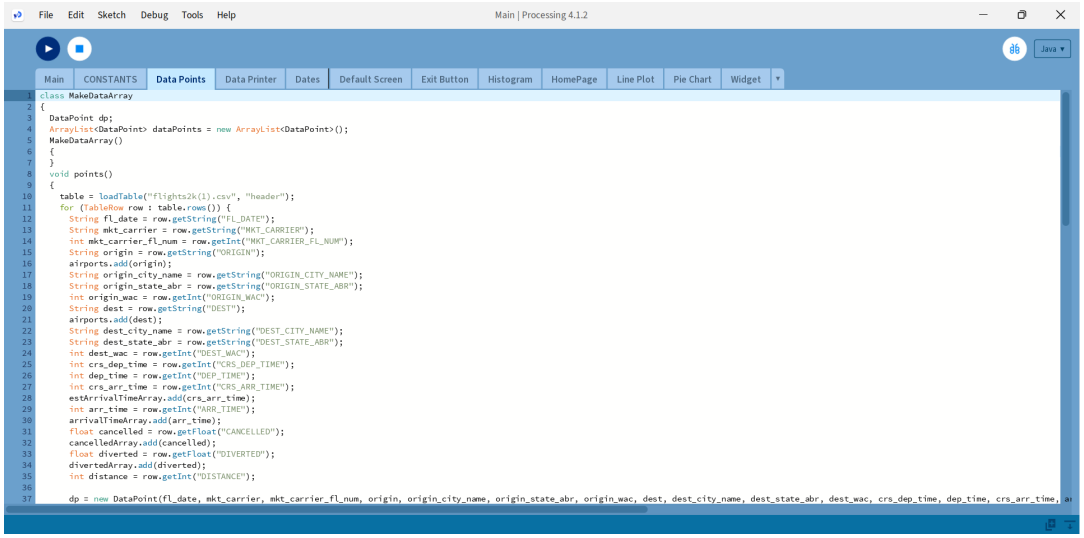
5. Testing and evaluation: We devised a plan for testing and evaluating our application, including thorough testing for bugs and issues, as well as evaluating performance, accuracy and usability from a user perspective.

By following a structured approach to investigate and plan, we laid a solid foundation for the successful development of our application, ensuring that it meets user needs and provides a seamless experience for reading and displaying CSV data related to flights.

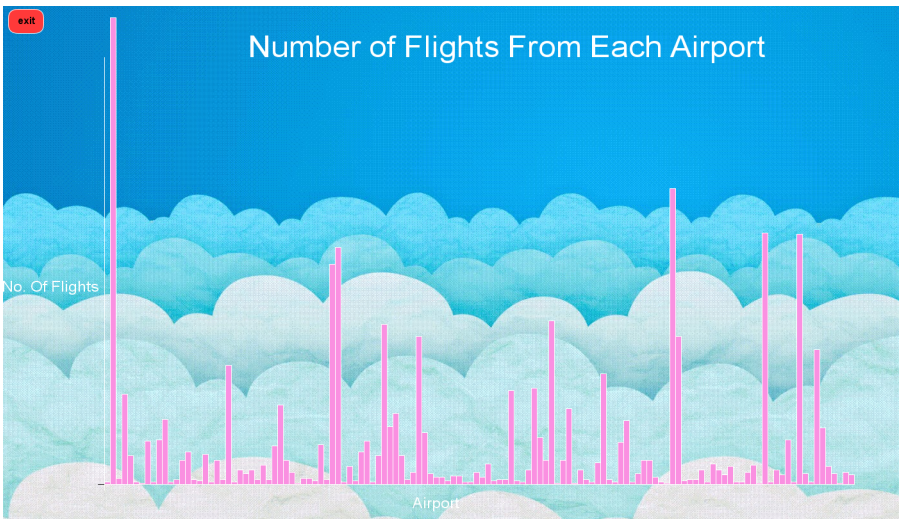
Design

Our project has successfully met the requirements outlined in the brief by implementing the following components:

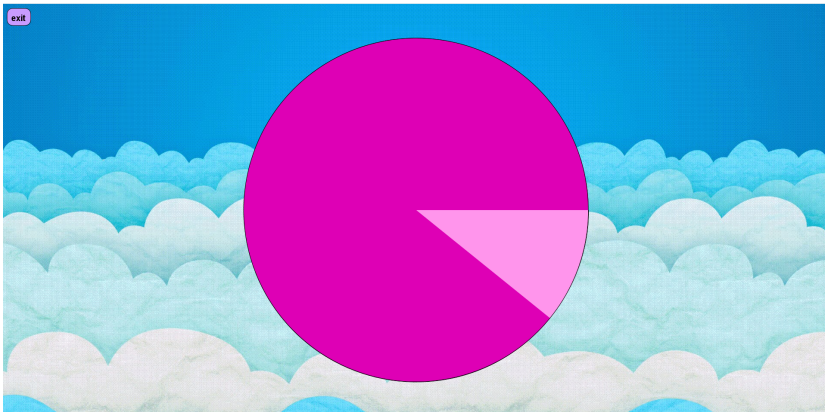
1. Data reading: We have written many different classes of code which all take different pieces of data from the file. We created a DataPoint class to represent each flight, with instances of the class created for each entry in the input file. This allowed for efficient data organisation and storage.



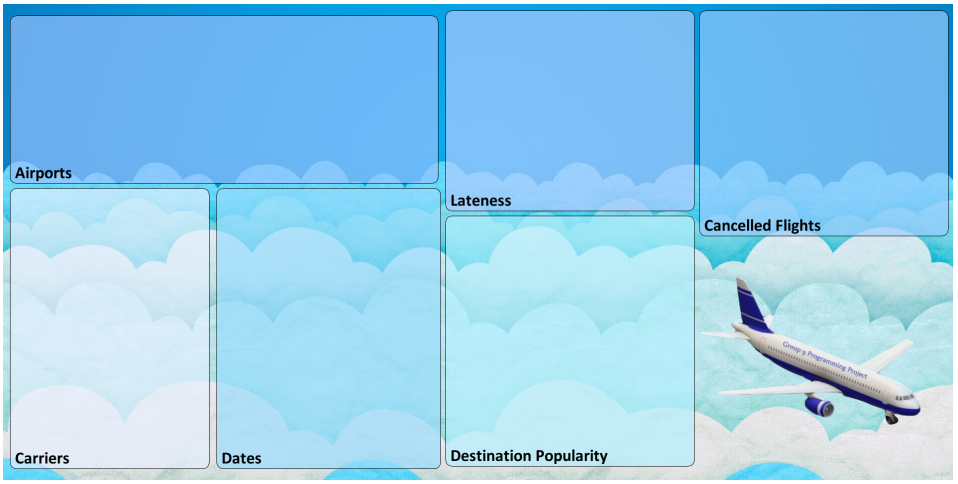
2. Queries: We have implemented code to select a subset of data based on various different queries. We have included queries for flights associated with a particular state and airports, flights within a certain date range, and flights sorted by lateness. These queries allow users to filter and sort the data based on their requirements, providing flexibility and versatility in data selection.



3. Graphs: We have written code to draw the data on the screen, using graphical representations. We have implemented numerous graphs such as pie charts and histograms in order to provide the user with a more friendly and visually appealing experience.



4. Integration: We have successfully integrated all the components together to create a cohesive and functional application. From the initial planning phase, we have outlined the architecture of the application, ensuring that all the components work together seamlessly to provide a complete solution for reading and displaying the data.



Our project has met the requirements of the brief by providing efficient data reading, queries, graphs and overall integration of all the components. This has resulted in an application that fulfils the project requirements.

As well as the basic requirements, we have decided to also implement a starting screen, simply for visual effects, in order to make the user experience even more pleasing to use.

Issues Encountered:

Throughout the project we encountered many difficulties. There have been times where we would have an idea for the project and spend a lot of time trying to implement the idea but ultimately it wouldn't work and we would have to scrap the idea. This took up a lot of our time. Also we struggled to get everything done, due to personal reasons such as members of our group getting sick. On a more technical aspect, at the very beginning, we struggled to use GitHub and pushing and pulling onto the repo, but as the weeks progressed we became more accustomed to it.

Contributions:

Rosemary Doyle:

- Created and implemented Histogram class to take in an arrayList of any size and display the frequency of the variables.
- Created and implemented Line Plot class to take in an arrayList of any size and display the frequency of the variables.
- Created method to display data on graphs by hovering over data points in both histogram and line graph.
- Created cancelled flight Pie Charts, including a hover method to display the title of each segment when the mouse is hovering over it.
- Made a second PieChart class which could take in ArrayList of any size and allowed the user to change the amount of variables being shown and showing the most significant values, in order to show most popular cities to fly to/from.
- Structured Main draw to allow users to switch between screens by creating a large switch statement in the draw method and assigning a screen variable to each screen.
- Created exit button, to return to homepage from any other screen.

Jason Qu:

- Created the DataPoints class.
- Created a drop down menu with all the dates.
- Created 31 separate classes that display the data by date.
- Implemented a scroll wheel.
- Fixed small issues such as the exit button not working for one of the pages.
- Edited the background and transparency of the widgets.
- Wrote the report.

Daniel FitzGerald:

- Created the”popularity” pie chart class, able to display an undefined number of variables.
- Colours in the pie chart would dynamically change and match legend regardless of data size.
- Created a second file reader class specifically for the pie chart.
- Created the landing screen at the beginning of the programming.
- Implemented code to display the gif in the loading screen.

James Hosey:

- Designed and coded the home screen
- Made the widget class and implemented all the widgets
- Designed each of the widgets to make it look better
- Tried to work on a map of the US with each state being a widget