Report

Team 16 report on CSU11013 programming project.

Team Members:

* David Sietko
* Lukas Maselsky
* Hezamary Paul
* Esosa Johnny
* Mark Varghese
* Joseph Reidy

For project setup please read the ReadME file located in the project repository.

Link: <https://github.com/davidtcd/programming_project_TEAM16.git>

For the design of this project, we wanted to make it simple and efficient. We focused a lot on user input and queries and wanted the user to have as many options as possible when viewing the data. All the buttons, dropdowns and other functions are all labeled so the user knows exactly what everything does. We started off with a selection screen where the user can choose which dataset they wanted to view. Whether it be 2k all the way to the full one. Upon choosing a dataset, a loading screen will pop up while the program loads the data. Once the program finishes loading the data, the user is then brought to the table screen which is just a standard table view of the dataset chosen. Across the top of the screen there is a navigation bar which allows the user to switch in-between “tabs” like on a search engine such as Google. Each “tab” corresponds to a different screen with a different data visualization being displayed. A list of the different screens that we have included in the project is given below.

* Bar Chart Screen
* Pie Chart Screen
* Tree Map Screen
* Line Graph Screen
* Search Screen
* Flight Map Screen

The Screens:

The name of each screen corresponds to the data visualization that is displayed on it. The screen name will be inside its corresponding tab so the user knows which tab leads where. Each of the screens embraces our core theme of user input. Each screen has multiple buttons, dropdowns, scrollbars, widgets and textboxes which allow the user to manipulate the data however they please. The user can sort the data from low to high or vice versa, choose a particular column of data they want to view (e.g. flight dates) or they can manually search for particular rows of data. The user is able to sort the data for every visualization, so the user can view the flight dates or times in a table-view, on a bar chart or pie chart or any of the other visualizations that our project has to offer. These are just a few examples some of the ways the user can manipulate the data, there is a bunch more that the user can do.

Splitting the work:

We decided to work on individual screens separately as we saw it as the most effective way to split work-load amongst the group. With this method each team member could design their own screen to make the project more unique and be able to make an ample contribution to the team. We also had one team member work on reading in the data and another who created the classes for the buttons and other interactive features.

Reading in data:

When a new dataset is added it is only processed once and stored into files to improve performance. We also have a dataset class that handles the data processing and contains methods to manipulate and extract that data (e.g. get the occurrence amount of a particular category).

Team contribution:

**David Sietko:**Designed and wrote all the methods for the bar chart screen and wrote the original navigation bar class.

**Lukas Maselsky:**Created the TreemapScreen class, the SearchScreen class, the base Widget class and Dropdown, Scrollbar and Button subclasses. Modified the navigation bar to have a more consistent design.

**Hezamary Paul:**Created the Screen class and extended the tableScreen class from it. Modified the tableScreen class and created 4 different ways to view it using dropdowns.

**Esosa Johnny:**Created and designed the Line Graph Screen and created constants for the screen size.

**Mark Varghese:**  
Handled data processing for the project, created the dataset selection and loading screens. Created the flights map screen and fixed minor bugs around the project.

A screenshot of a graph

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated