

TNPG: aTriceratopy, Roster: Brian Yang, Jonathan Song, Prattay Dey, Verit Li  
SoftDev pd7

P4 -- data visualization project design doc

2023-05-02

Time Spent: 3 hrs

Target Ship Date: 2023-05-23

## ***Aeroplano Graveyardo***

### **Abstract:**

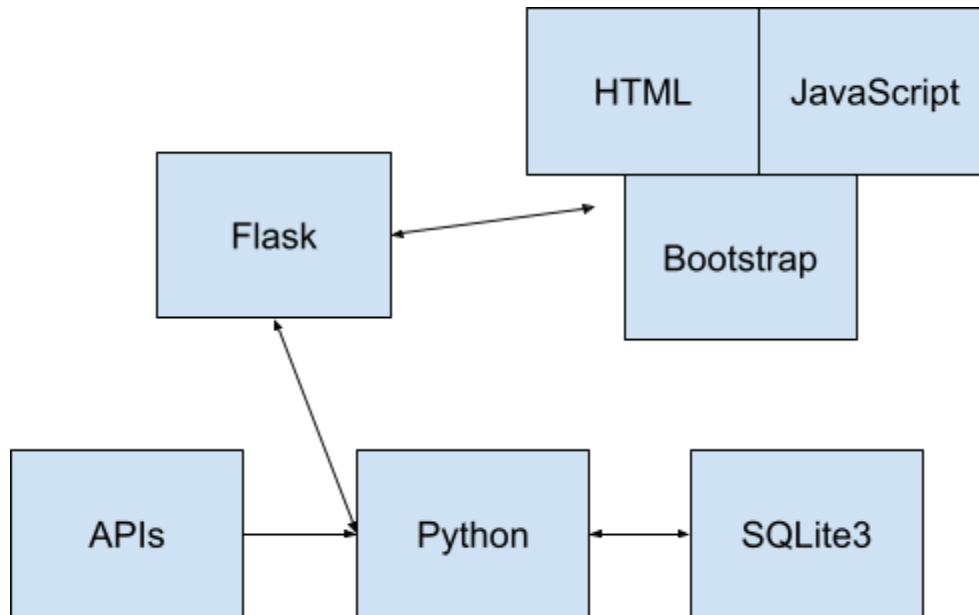
Aeroplano Graveyardo is a global map showing all civil aviation accidents from 1908 to 2019. Users will be able to see points on a globe of locations of crashes, select a crash and see the plane's intended path, interact with filters like aircraft type, country of crash, deadliness, and accident year. Users will also be able to see trends in the database file.

### **Program Components:**

- Javascript
  - Allows for more user interactivity
  - Cool visuals like animation
    - animation of plane flying its route
    - clouds parting affect when going to new page?
  - D3.js
    - Javascript library for our interactive map
    - different graphs
      - <https://d3-graph-gallery.com/>
- HTML
  - Jinja syntax to collaborate with Flask
  - Templates to be rendered by Flask
- Python
  - Makes API calls
  - Uses database functions
- Flask
  - Our web server and delivery framework
  - Renders html templates to browser
- SQLite3
  - Storage of airplane crash data based off of:
    - <https://www.kaggle.com/datasets/cgurkan/airplane-crash-data-since-1908>
- FEF (Bootstrap)
  - Used to style website
  - Chosen because of more experience using within the team

- Provides grid layout, navbar, dropdowns, range forms (a slider)
- APIs
  - <https://positionstack.com/>
    - To return coordinates when given the name of a location which can be then marked on D3.js map

### Component Map:



### Front End Site Map:

#### *map.html*

- Takes filtered data from the dataset and displays crashes it on an interactive map
- Clicking on a specific crash will allow you to view more detail on the incident (exact time/date and summary of the incident)
- Be able to filter the columns on the dataset to show what the user specifies

#### *summary.html*

- User is directed to this page when they click for more information on a specific flight
- A page that shows all the information provided by the dataset for a row (Date, Time, Airline/Operator, Flight number, Route, Aircraft Type, ICAO registration, total aboard, fatalities, summary)

#### *trends.html*

- Ranks airplane crashes depending on one factor (Time morning/afternoon/night, airline/operator)

- Be able to change time frame of the crashes
- Have a drop down bar to select which trend to visualize

### **Backend: Flask**

#### *\_\_init\_\_.py*

- The foundation of our flask app. Runs and displays web pages.

#### *rank.py*

- Calculates the ranking for all the trends provided at the start of the site
- Information is cached in the database to prioritize responsiveness
  - The rankings are stored so that they don't have to be calculated every time the trends page is loaded

#### *db.py*

- Contains function to create and maintain a database. Including the storage of ranking lists.
- Contains function to retrieve data from database

#### *api.py*

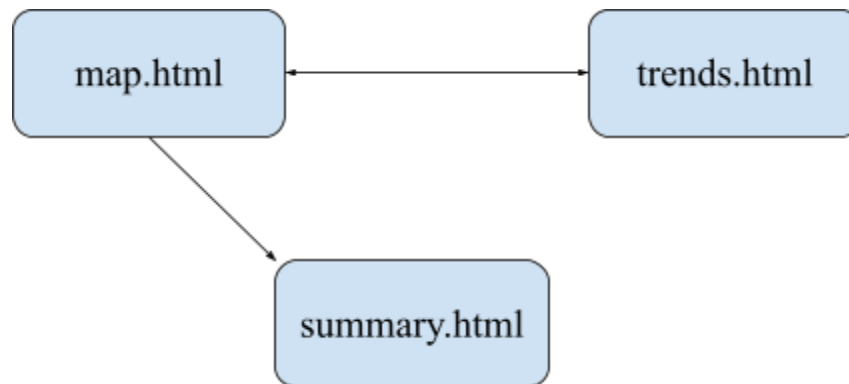
- Intermediary modules to allow getting information from api convenient and simple

### **Database Organization: SQLite3**

- Airplane crashes table consisting of:
  - Flight id
    - generated starting at 0 and adding 1 for every next flight
  - Date
  - Time
  - Location
  - Operator (Airline or operator of the aircraft)
  - Route
  - Aircraft type
  - # of Passenger/Crew aboard
- Crash coordinates table consisting of:
  - Flight id
  - Longitude
  - Latitude
    - (Both longitude and latitude generated from positionstack API)
- Ranking Tables
  - Table consisting of top 50 crashes due to time in day listed from descending order

- Table consisting of top 50 crashes due to airline and operator listed from descending order

Site Map:



#### Task List:

- Database (Verit)
  - functions for populating the database
  - functions for accessing data
- Map (Brian)
  - map.html template
  - javascript for interactive map
- Summary (Jonathan)
  - summary.html template
  - flask such that every crash has a route to a summary page
- Trends (Prattay)
  - trends.html template
  - function to calculate rankings of crashes based on a factor

#### Useful Links:

<https://www.kaggle.com/datasets/warcoder/civil-aviation-accidents>

<https://www.kaggle.com/datasets/cgurkan/airplane-crash-data-since-1908>

<https://www.nts.gov/safety/data/Pages/GeneralAviationDashboard.aspx#AVSpreadsheet>