Customizable Analysis and Visualization Tool for COVID Cases

2nd Semester Plan

Team Members

- Calvin Burns, cburns2017@my.fit.edu (Team Lead)
- Sam Hartle, shartle2017@my.fit.edu
- Nicole Wright, nwright2017@my.fit.edu
- Stian Olsen, shagboeolsen2017@my.fit.edu

Faculty Advisor/Client

- Dr. Philip Chan, pkc@cs.fit.edu

Goal and Motivation

- Goal:
  - A web application that shows COVID case data
  - Can do customizable analyses/visualization of results
  - Allows users to add additional pieces of data related to COVID
- Motivation:
  - Many COVID dashboards available
  - Analyses are pre-determined
  - Develop a dashboard where analyses are not predetermined
  - User has the ability to customize both results and visualizations

Features (1-4)

1. Users can select variables and perform custom operations
   a. Ranking, average, etc
   b. Allows for comparing across many different categories
2. Users can plot results from variable operations using various charts, graphs, and plots
3. Users can save custom visualizations to their unique workspace
4. Users can add additional types of datasets
   a. Airline travel data, school data, etc
   b. Could allow for analyzing infection rates in schools with face-to-face classes versus schools with online education

Features (5-7)

5. Users can make use of custom visualizations and layer plots on the visual for comparison
6. Users can
   a. Add/analyze multiple datasets
   b. Deleting existing datasets
   c. Have current datasets automatically updated daily
7. Users can apply to make private datasets public

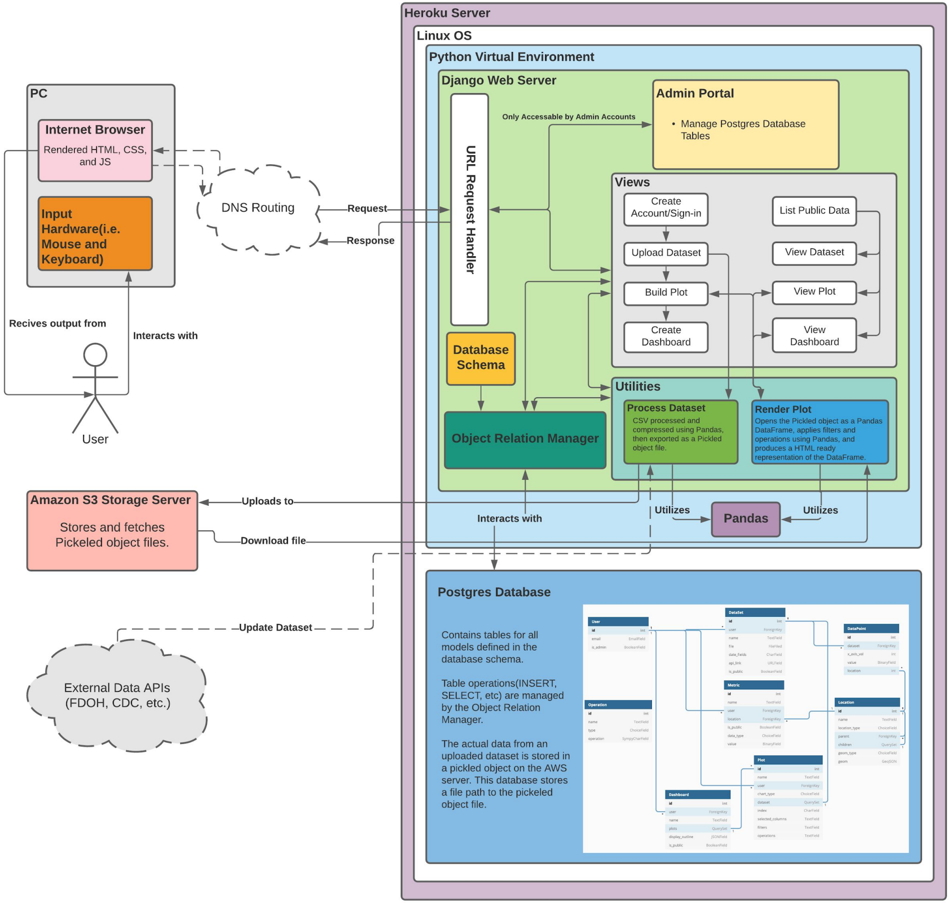
Novel Features

- Users being able to create/save custom plots is not available on other dashboards
  - Most plots are pre-set
- Users being able to layer plots shows relationships between data plotted on a shared x-axis
- Users being able to add additional types of datasets is not possible in other dashboards
  - Similar to pre-set plots

Technical Challenges

- Using Django/AWS/Pandas
- Learning various Javascript frameworks and integration with Django/AWS/Pandas
- Rest APIs and file formats for reading data (JSON, CSV, etc)
- Importance of design choice given amount of data in our system
- Exploring SQL and its possibility as an alternative to Pandas

System Architecture Diagram



- Manage Postgres Database Tables

Process Dataset: CSV processed and compressed using Pandas, then exported as a Pickled object file.

Render Plot: Opens the Pickled object as a Pandas DataFrame, applies filters and operations using Pandas, and produces a HTML ready representation of the DataFrame.

Amazon S3 Storage Server: Stores and fetches Pickeled object files.

Postgres Database

Contains tables for all models defined in the database schema.

Table operations(INSERT, SELECT, etc) are managed by the Object Relation Manager.

The actual data from an uploaded dataset is stored in a pickled object on the AWS server. This database stores a file path to the pickeled object file.

Evaluation

- Speed
  - Plot number of queries/plots vs. response time
  - Plot query size vs. response time
  - Plot number users (concurrent actions) vs. response time
- User Survey
- Accuracy
  - Smaller, more manageable dataset can be analyzed in Excel and then compared to results from our system
- Reliability
  - Can get logs of all system errors (404, 500, etc.)

Progress Summary Features 1-4

Feature	Completion %	To Do
1 - Select Variables and Perform Custom Operations	75%	Finalize Operations and perfect UI.
2 - Display their results on plots	75%	Finish Plot utility for Scatter Plots and Timelines.
3 - Save unique workspaces	15%	Page for creating unique workspaces(selecting plots and positioning them on the screen).
4 - Add additional datasets, auto update datasets	75%	Auto update datasets using an API endpoint.

Progress Summary Features 5-7

Feature	Completion %	To Do
5 - Layered Plots	15%	Add a page for layering plots. Update plot utility to work with layered plots.
6 - Multi dataset analysis and management	25%	Finish dataset management operations, multi dataset analysis.
7 - Application to make data public	25%	Create a page for submitting applications and a page for reviewing applications.

Milestone 4

- Finish, test, and demo *ability to plot data on graphs*
  - Feature 2
- Finish, test, and demo *select variables and perform custom operations*
  - Feature 1
- Finish, test, and demo *add additional datasets and auto update via API*
  - Feature 6
- Implement, test, and demo *application to make data public*
  - Feature 7

Milestone 5

- Implement, test, and demo *save unique workspaces*
  - Feature 3
- Implement, test, and demo *layering plots and layering on FL map*
  - Feature 5
- Implement, test, and demo *multi dataset analysis and management*
  - Features 4 and 7
- Preliminary testing to gather quantitative results on system performance

Milestone 6

- Test/demo of the entire system
- Evaluation results
- Create user/developer manual
- Create demo video

Task Matrix for Milestone 4

Task	Stian	Sam	Nicole	CJ
1. Continue work on scatter plot/plot utility	Example for Dr. Chan on why we would use a scatter plot for our application Finish development on plot type (85%)	Assist as needed (5%)	Assist as needed (5%)	Assist as needed (5%)
2. Continue work on plot creation tool	Assist with different chart types (10%)	Assist with various operations as needed (10%)	Add additional “cards” to Create Plot interface which shows general plot details (20%)	Work on Create Plot GUI and using specified filters/operations instead having to “program” (60%)
3. Continue work on creating lab testing plot/Start API auto update	Assist with different chart types and operations research (10%)	Plot positivity rate vs. date Focus on operations: count, sum, division (60%)	Assist as needed (5%)	Ideally this will be able to done through GUI as well Continue to build Create Plot GUI abstractly (25%)
4. Dataset management and application system			Create flows for user applying to make data public and admin review/approve application (80%)	Assist as needed (20%)

Questions?