Title: Customizable Analysis and Visualization Tool for COVID Cases

Team Members:

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

Advisor: Dr. Philip Chan, pkc@cs.fit.edu

Client: Dr. Philip Chan, pkc@cs.fit.edu

Date(s) of Meeting(s) with the Client for developing this Plan:

- Bi-weekly 12-1 PM on Monday

Goal and Motivation:

- **Goal:** To create a website that shows COVID case data, allows users to perform customizable analyses/visualization of results, and allows users to add additional pieces of data.
- **Motivation:** COVID is still ongoing. There are various COVID dashboards out there, but the analyses are pre-determined. We are developing a COVID dashboard where the analyses are not predetermined. The user has the ability to customize and get the results they are looking for rather than being stuck with certain results.

Approach (key features of the system):

Feature 1 (4.1 from Requirements Document):

Users can select specific variables and perform custom operations (ranking, average, etc) on the selected variables. This allows users to compare filtered data across a multitude of different categories (age, location, schools, month, hospitalized vs deaths, etc).

Feature 2 (4.2 from Requirements Document):

Users can display their results from variable operations using various charts, graphs, and plots. This provides the ability to see any trends from the selected data and does not limit users to using predetermined statistics and plots.

Feature 3 (4.3 from Requirements Document):

Users can save custom visualizations to their own unique workspace. These charts/graphs/plots will give the user the ability to easily access the custom analysis at any time and see the analysis with data updated daily.

Feature 4 (4.4 from Requirements Document):

Users can add additional types of datasets (airline travel data, school data, etc). Each user will be associated with a set of data, where their uploaded data is private to them and only them. For example, this feature could allow users to analyze the infection rates in areas with face-to-face classes versus areas with online education.

Feature 5 (4.5 from Requirements Document):

Users can make use of the custom visualizations and layer plots on the visual. Overlaying related data and plots makes statistical comparisons much more efficient and puts all the data in one location for users.

Feature 6 (4.6/4.7 from Requirements Document):

Users can add and analyze multiple datasets at a time, delete existing datasets, and have their current datasets automatically updated. For an existing dataset, automatic pulling of the most recent data will occur daily.

Feature 7 (4.6/4.7 from Requirements Document):

Users can submit an application to make their private datasets public. Users with admin level permissions can access a tool for reviewing and approving data sets. An application will require a sample of the data, source, API URL (if applicable), description, and contact information for further questions.

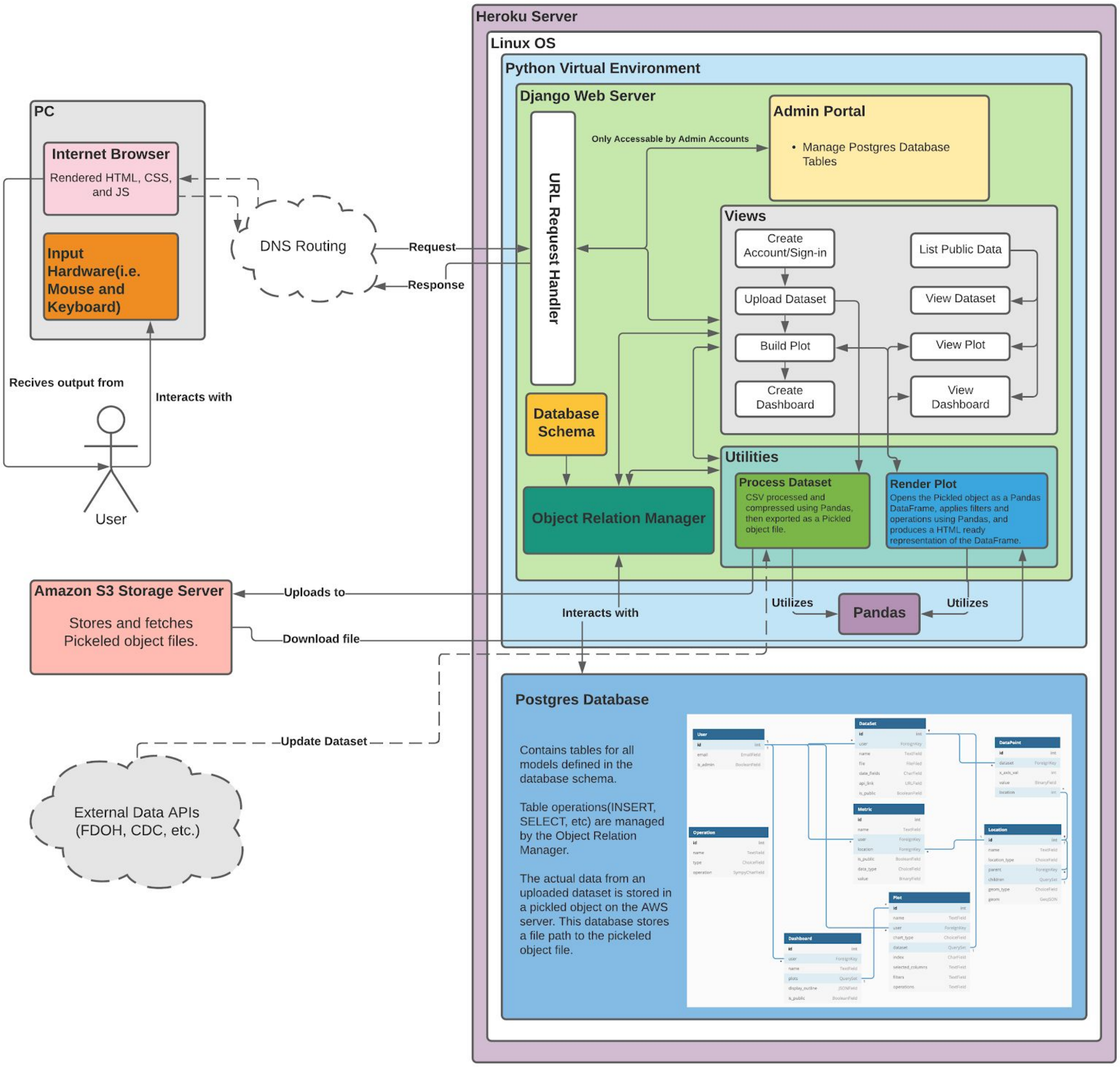
Novel features/functionalities:

1. Users being able to create/save custom graphs (Features 2 and 3) is not available on many other dashboards. Most of the plots are pre-set. Giving users the ability to pick their variables and operations to plot and show comparisons expands the functionality of a traditional dashboard.
2. Allowing users to layer plots (Feature 5) will show relationships between data plotted on a shared x-axis. This can help show correlations between trends. Other dashboards do not have this functionality.
3. Giving users the ability to add additional types of datasets is not possible in other dashboards (Features 4, 6, and 7). This feature allows users to perform analysis on data that is currently not available and potentially share data for other users to use.

Technical Challenges:

1. Learning how to use Django/AWS/Pandas
2. Learning about various Javascript frameworks and integration with Django/AWS
3. Learning about GIS and Spatial Database Relations
    a. Design considerations for the large amount of data we are working with
4. Learning about RestAPIs and file formats for reading data (JSON, CSV, etc)
5. Exploring SQL and its possibility as an alternative to Pandas

System Architecture Diagram:



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.

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.

User	
id	int
email	EmailField
is_admin	BooleanField

DataSet	
id	int
user	ForeignKey
name	TextField
file	FileField
date_fields	CharField
api_link	URLField
is_public	BooleanField

DataPoint	
id	int
dataset	ForeignKey
x_axis_val	int
value	BinaryField
location	int

Operation	
id	int
name	TextField
type	ChoiceField
operation	SympyCharField

Metric	
id	int
name	TextField
user	ForeignKey
location	ForeignKey
is_public	BooleanField
data_type	ChoiceField
value	BinaryField

Location	
id	int
name	TextField
location_type	ChoiceField
parent	ForeignKey
children	QuerySet
geom_type	ChoiceField
geom	GeoJSON

Plot	
id	int
name	TextField
user	ForeignKey
chart_type	ChoiceField
dataset	QuerySet
index	CharField
selected_columns	TextField
filters	TextField
operations	TextField

Dashboard	
id	int
user	ForeignKey
name	TextField
plots	QuerySet
display_outline	JSONField
is_public	BooleanField

Evaluation:

- speed (e.g. how fast does the system achieve each of its goals?
    - Speed is an important factor for this project. We're dealing with large amounts of data and we would like to provide analysis as fast as possible. To evaluate the speed, we will evaluate the data storage design model and determine if we can decrease the time by storing the data differently, run different queries, refactor the database, or use different algorithms.
    - Number of queries/plots
        - Plot number of queries/plots vs. response time
    - Size of queries
        - Plot query size vs. response time
    - Number of users (simultaneous queries/actions)
        - Plot number of users vs. response time
- user survey (e.g. rating of 1-5 on each of the different features)
- accuracy (e.g. how accurate is the system?)
    - In our system, the accuracy is evaluated mainly on the accuracy of the analyses provided to the user. The calculations need to be correct and the data we're using needs to be correct. This can be evaluated by comparing the analyses our system provides to the analyses provided by other covid dashboards.
    - Since our system is built in an abstract way, we can upload any CSV of data and perform analysis on it. A smaller, more manageable dataset can be analyzed in Excel and then compared to results from our system.
- reliability (e.g. out of 10 times, how many times does the system achieve each of its goals)
    - Using a system monitor, we can get logs of all system errors(404, 500, etc.). As of now, we don't expect reliability to be a major evaluation statistic.

Progress Summary:

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.
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 (Feb 15): itemized tasks:

A. Finish, test, and demo *ability to plot data on graphs(feature 2).*
B. Finish, test, and demo *select variables and perform custom operations(feature 1).*
C. Finish, test, and demo *add additional datasets and auto update via API (feature 6).*
D. Implement, test, and demo *application to make data public(feature 7).*

Milestone 5 (March 15): itemized tasks:

A. Implement, test, and demo *save unique workspaces(feature 3).*
B. Implement, test, and demo *layering plots and layering on FL map(feature 5).*
C. Implement, test, and demo *multi dataset analysis and management(features 4/7).*
D. Begin preliminary testing to gather quantitative results on system performance

Milestone 6 (April 12): itemized tasks:

A. Test/demo of the entire system
B. Evaluation results
C. Create user/developer manual
D. 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%)

Discussion of each planned task for Milestone 4:

- Task 1:

  This task will mainly revolve around Stian continuing work on determining if the scatter plot would be a good choice for our application. He needs to consider the variables that would be plotted on each axis. If so, he would be finishing development for support for this plot type. He was the sole initial developer for this task, so other team members will assist him as he needs.

- Task 2:

  This task will mainly revolve around CJ continuing work on the male/female pie chart via GUI operations. His work on this task will involve finishing the functionality of the Create Plot GUI and allowing users to specify specific filters/operations instead of having to "program" their own plot. Nicole will add additional "cards" to the Create Plot GUI which will allow the user to input general plot details such as name, type, etc. Stian will assist with different chart types as CJ and Nicole see fit. Sam will assist with various operations as CJ and Nicole see fit.

- Task 3:

  This task will mainly revolve around Sam continuing work on creating the lab testing line plot via GUI operations. This plot will show positivity rate over time. He will be focusing on the different backend operations needed to display this plot properly (count, sum, and division). Stian will assist with different chart types and operations research. CJ will be continuing to build the Create Plot GUI abstractly. This is so that the plot will ideally be able to be created through the interface once it has been initially programmed. Nicole will assist any of the other three group members as needed with this task.

- Task 4:

  This task will mainly revolve around Nicole and it implements feature 8. Additional functionality for managing all datatypes will be added(delete/edit). A flow for a user submitting an application to make their data public will be created. In addition, a flow for an admin to review and accept or reject applications will be created.