# Progress Evaluation - Milestone 5

**Title**: Customizable Analysis and Visualization Tool for COVID Cases

**Team Members:**

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

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

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

**Progress Matrix for Milestone 5:**

| Task | Completion % | Stian | Sam | Nicole | CJ | To do |
|---|---|---|---|---|---|---|
| 1) Continue work on scatter plot | 100% | 100% | - | - | - | None |
| 2) Operations card update to be more intuitive | 80% | - | - | - | 80% | |
| 3) Save unique workspaces | 0% | - | - | - | - | All |
| 4) Layering plots | 80% | 80% | - | - | - | Change UI so user can create multiple plots |
| 5) API auto updates for datasets | 50% | - | 50% | - | - | Convert/append JSON object received from API endpoint to a current CSV |
| 6) Finish Application Feature | 100% | - | - | 80% | 20% | None |

**Discussion of each accomplished task and obstacles for Milestone 5:**

- Task 1: This task revolved around finding a good use for the scatter plot and finishing up the plot utility. We decided to use a scatter plot to see if there is a correlation between positivity rate and mobility during the pandemic in Florida and in the USA. This task also revolved around adding more labels on the y-axis to make the plot more useful. We also encountered a problem with partial weeks when we were resampling the data. We fixed this problem by calculating the number of days in each week and then filtering out the partial weeks.

- Task 2: After meeting with our client, Dr. Chan, he suggested that we try to use a similar format to SQL for querying and manipulating a dataset. The work we had done for filters and operations was vaguely similarly to SQL but didn't work exactly how our client wanted it to. To move closer to the SQL method, we removed our filters and operations card and replaced them with a single "SQL Query" card. This form has fields for SELECT, FROM, WHERE, GROUPBY, and ORDERBY. Once this form was built, we began modifying our backend to use these fields. Building our own SQL evaluator for Pandas turned out to be a very involved task. After struggling for a few days, we took a step back and looked for 3rd party plugins that could help us. We found a pandas plugin called "dataframe_sql". This library allows us to pass a SQL query statement and it will return a filtered and manipulated dataframe. There are still quite a few remaining bugs. For example, the library has issues with some of our datatypes like BigInt64. Since we will now use SQL statements, we need to rework the interface for building Plots. In addition, the fields on Plot need to be modified to save the SQL statement. Finally, some type of SQL statement validator needs to be added to our front and backend. We want to catch errors in the SQL before we try to evaluate them.

- Task 3: Unfortunately, no progress was made on task 3 during this milestone.

- Task 4: This task revolves around figuring out which plots can be layered. After some research, we figured out that we can layer pie charts, bar charts, line charts, timelines. We still need to figure out if we can layer scatter plots.

- Task 5: This task revolves around the user uploading a dataset and setting a URL field that points to an API endpoint. Our system then will fetch the file from that endpoint and process the contents of the file to update the dataset. The research phase of this task is nearly complete. The backend implementation will include querying the API endpoint given by the user and converting the returned JSON object to a CSV object that will be appended to the end of a dataset. Frontend changes include changing the "Upload Dataset" page to add a text field for setting the URL that points to an API endpoint.

- Task 6: After meeting with Dr. Chan, he reminded us to use shared, curated, and private data, instead of just public/private. This task revolves around making the changes needed to include all three of these types of data visibilities. Our public data page now includes both shared and curated sections of data. On the public dataset application form, there is now an option to request review of your data. If not selected, the data will be uploaded to the shared data section. If selected and reviewed/approved, it will be uploaded to the curated data section.

**Discussion of contribution of each team member to Milestone 5:**

- Stian: Used datasets proposed by Dr.Chan to create different scatter plots. Encountered a problem with the display of partial weeks and fixed it with pandas. Used a Google mobility dataset to plot the correlation between positivity rate and change in grocery/pharmacy visitors. Did some research to figure out which plot types we can layer and made a demo with line and pie.
- CJ: Worked on the build plot page. After the first feedback meeting, I moved select columns to the bottom of the form. In addition, I started writing code to combine operations and filters into one card with drag and drop ordering.

  After the second feedback meeting, Dr. Chan recommended that we try to keep our implementation closer to SQL. “There is no reason to reinvent the wheel”. Operations and filters cards were removed. A new card for SQL queries was added. It has fields for SELECT, FROM, WHERE, GROUPBY, and ORDERBY. Writing the backend code to evaluate these fields on the actual data was very difficult and hard to make robust. I realized that other people have probably wanted to do the same thing I’m trying to do so I began looking for a Pandas method or third party plugin that would allow us to use SQL queries directly on the dataframe. I found `dataframe_sql` on GitHub. Currently working on updating our code base to work with this library.
- Sam: Used a lab testing dataset proposed by Dr.Chan to create a line graph showing weekly average positivity rate in FL. Had issues with the display of timestamps in the charting software and fixed it using various pandas functions for formatting datetime types. Found a frequently updated GitHub vaccine dataset to plot the relationship between the number of vaccines distributed per 100 people and the actual number of vaccines administered per 100 people. Did the majority of the required research for beginning the implementation of updating datasets currently active in our system. In addition, did the majority of the progress evaluation and presentation.

- Nicole: Updated the public data page to utilize both share and curated data. Did this by switching the 'is_public' field in the database to a visibility field that includes all 3 options: private, shared, and curated data. Then refactored the backend to use this visibility field instead of is_public. On the frontend worked in the html files to update the public data page. There are now 2 sections of tables: one with curated datasets, dashboards, and plots, and the other section with the shared data.

**Task Matrix for Milestone 6:**

| Task | Stian | Sam | Nicole | CJ |
|---|---|---|---|---|
| 1) Ensure that all showcase materials are completed and submitted | Demo Video (25%) | User/Developer manual (25%) | Clean up ebook page and assist Sam and/or Stian as needed (25%) | Clean up poster and assist Sam and/or Stian as needed (25%) |
| 2) Update Build Plot to use SQL statements | - | - | - | Implement evaluation of SQL queries on Pandas df (100%) |
| 3) Auto updates for datasets | - | Implement automatic dataset updates via API endpoints (80%) | - | Assist Sam as needed with implementation (20%) |
| 4) Update application process to work for plots & dashboards | - | - | Implement application flow for Plots and Dashboards (80%) | Assist Nicole as needed (20%) |
| 5) Test/demo of system for evaluation | Speed (25%) | Reliability (25%) | User survey (25%) | Accuracy (25%) |

**Discussion of each planned task for Milestone 6:**

- Task 1: This task will make sure that all necessary materials for the virtual senior design showcase are completed, reviewed by our client/advisor, and submitted to the proper submission link by the deadline. Stian will make the demo video, Sam will create the User/Developer Manual, Nicole will finalize the draft of the ebook submitted with this milestone, and CJ will finalize the draft of the poster submitted with this milestone. Nicole and CJ will help Stian and Sam as needed once their materials are finalized.

- Task 2: A new 3rd party plugin has been found to evaluate SQL queries on a Pandas Dataframe. The Plot fields need to be updated to save a SQL query. The constructor for a Plot dataframe needs to be updated to use the SQL query. Issues with compressed datatypes need to be resolved(i.e. Category fields).

- Task 3: This task will finish the requirement for auto-updating datasets currently on our system. The backend implementation will include querying the API endpoint entered by the user when uploading a dataset and converting the returned JSON object to a CSV object that will be appended to the end of a dataset. Frontend changes include changing the “Upload Dataset” page to add a text field for setting the URL that points to an API endpoint.

- Task 4: Our version 1 of the Application Flow gives functionality to the Dataset model. This task implements the application flow for both the Plot and Dashboard models.

- Task 5: This task is focused on testing and will involve a test/demo of our entire system in order to obtain evaluation results for each of the four categories mentioned in the task matrix. For speed, we will plot the relationship between the number of queries/plots vs. response time, the query size vs. response time, and the number of users vs. response time. For reliability, we can use a system monitor to get logs of all system errors. As of now, we don’t expect reliability to be a major evaluation statistic. For the user survey, we can gather qualitative results from users not familiar with our application. For accuracy, we can evaluate our system by comparing the analyses our system provides to the analyses provided by other covid dashboards. We can also 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.

**Date(s) of meeting(s) with Client/Advisor (same) during Milestone 5:**

- March 1st
- March 15th
    1. *Rescheduled to March 17th*

**Client feedback on the current milestone:** See Faculty Advisor Feedback below

**Faculty Advisor feedback for Milestone 5:**

- Notes from 3/1/2021
    1. Y-axis use positive cases in addition to using positivity rate (Stian)
    2. Remove the data points from a partial week, which is inconsistent with meaning of x-axis (full week) (Stian)
    3. Plot google mobility (Stian)
    4. API endpoints (Sam)
        - Possibly doing a full download to check for any previous changes
            - Weekly
        - Append by querying recent data (objectID or date)
            - Daily
    5. Focus on one round of operations with a max of 2 datasets (Calvin)
    6. Continue to separate shared vs. curated datasets (Nicole)
    7. Look into vaccines for plotting (Stian/Sam)
- Notes from 3/15/2021
    1. Use different colors/shapes for different months to distinguish between unexpected data points for scatter plot (Stian)
    2. Plot duration of the lockdown on top of number of cases to see if cases are dropping during period (Stian)
    3. Plot duration of lockdown vs. cases per capita across different counties/states (Stian)
    4. Color coding dropdown selection for shared, curated, and private datasets (Nicole)
        - Possible addition of “- datasetVisibility” to dataset name
    5. For the operations (CJ)
        - UI
            - Little typing of syntax
            - Plotting vs how to get the data
            - Plotting
                - Need to specify what kind of plot and what are the axis
                    - Labeling the x and y.

  - Getting the data, this produces a table and the columns will correspond to the x/y axis and labels from “Plotting” (By default the first col is x, second is y. Additional columns will be additional y axis/curves.)
    - Ordering of menus might be similar to the execution order of SQL
    - Deciding on datasets
    - Selecting variables(i.e. columns)
    - What rows to keep what rows to ignore
    - GroupBy/OrderBy
- Mapping of UI to execution of operations
- Execution of operations
  - SQL library for dataframes…
    - Syntax order is SELECT, FROM, WHERE,...
    - Execution order is FROM, WHERE, …, SELECT
  - Other Pandas things…

6. Divide Vaccine rate by distribution rate (Sam)
   - Perfect value is 100%
      - Current max around 50%
7. API specific to each website in terms of autoupdate (Sam)
   - Possibilities
      - Filter by age range (FDOH)
         - Geometric, geographic data to filter
      - Worker thread separate from main web thread