Project Sprint #4

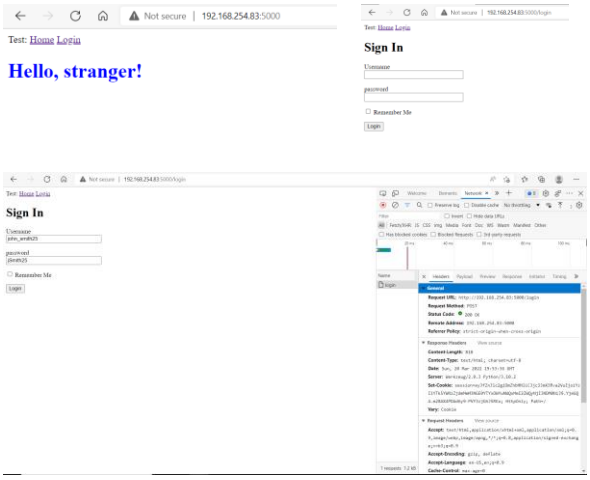
Aijing Gao, Shiyi Qin, Baiyan Ren, Yi Wang, Haojie Yu

agao48@gatech.edu

TASKS ACCOMPLISHED

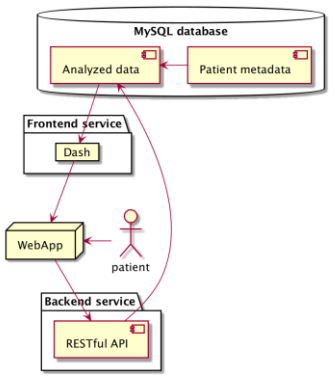
We have set up the structure of github repository, which has 4 branches. Main branch is where all changes eventually get merged back into. The other three branches are for individual modules such as database, frontend, and backend.

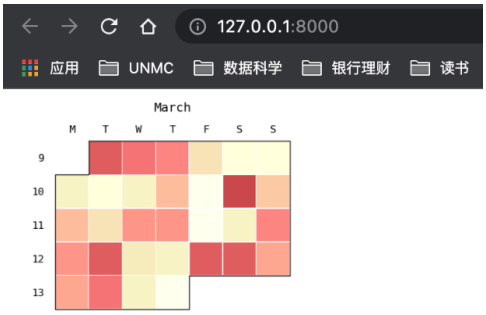
Haojie Yu has implemented authentication on the backend using Flask_Login and werkzeug.security and created Simple homepage and login page.



Baiyan Ren updated the README.md by including a flowchart as left below.

Baiyan Ren created static monthly calendar using matplotlib package. The color of each day reflects the number of appointments/events on the specific date. The plot is converted into html code then embedded into Dash dashboard as right below:



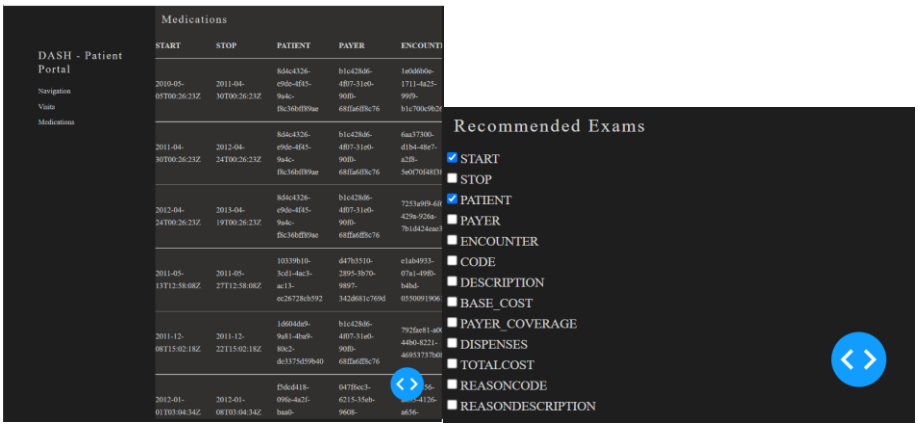


Aijing Gao created a mysql database, cs6440_sp22_team012, that contains 6 tables (see the below). Each table is corresponding to a FHIR resource. All team members will use this database for local code development. Aijing also finalized the logic for recommended tests or procedures based on patient's age, gender, and comorbidities.

```
+-------------------------------+
| Tables_in_cs6440_sp22_team012 |
+-------------------------------+
| Careplans                     |
| Conditions                    |
| Encounters                    |
| Medications                   |
| Observations                  |
| Patients                      |
| Procedures                    |
+-------------------------------+
```

Shiyi has extended the functionality of the previous bar-generator from a single bar to multiple bars. Additionally, Shiyi has developed a Python-based pipeline that obtains the desired data structure from the established database ("observations") to the visualization script. Below is an example of a test result visualization for the blood pressure from a sample patient.

Yi investigated using the Dash to make the front end, including the layout design, how to load the data for medication display, checklist for the recommended exams. Yi developed an initial front-end page based on the Dash. If using the Dash to deploy the app.py, it could include app, data, assets, components, callbacks.py.



CHALLENGES

It is challenging to create an interactive calendar using Python. There are two ways to embed matplotlib figures into Dash, one is using mpl_to_plotly() function, the other is using html code converted by the figure. The first function does not support heatmap. So, we used the second method, by which the embedded figure is not interactive on the dashboard. We have some challenges on how to integrate API into front-end. There are also some challenges with adopting framework to match the architecture of the backend.

PLANS AHEAD

We will first try building a dashboard using a flask framework template (https://github.com/app-generator/flask-adminator). In the mysql database, we are going to add one more table, user, which includes username and password for user authentication. Another function to be implemented on the backend is routing. If routing is implemented, we plan to develop the page after the user is logged in.

Updated timeline