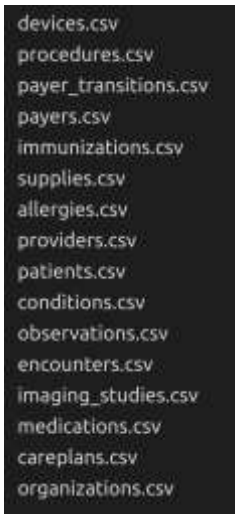
Status Check #1

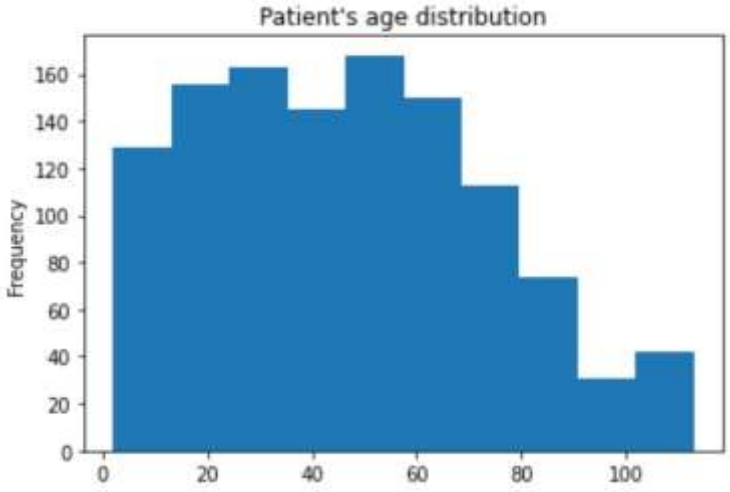
Tasks Accomplished

Aijing has been working on setting up MySQL database using Synthea data, which has 16 FHIR resources.

Required tables: patients (1,171 patients), observations, medications, conditions, and procedures

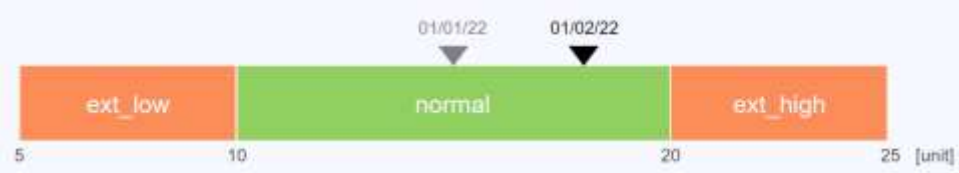
Possibly useful tables: care plan (e.g. Respiratory therapy, Routine antenatal care, Diabetes self management plan, Lifestyle education regarding hypertension), providers, immunizations





Baiyan figured out how to build the calendar to help patients visualize upcoming appointments. Specifically, there are two forms of calendars. The first one with HTML format is created by Python code. It highlights the dates with appointments. The second one is an embedded Dash component, which called date ranges (dcc.DatePickerRange). When patients select a date range, the web app show a list of appointments within this date range.

Shiyi has drafted a visualization for displaying lab result using Plotly from dummy data. The plot consists of three major components—a bar group that shows normal and extreme test ranges, a scatter group that points out 2 latest test results of the patient (if available), and an annotation group that specifies the date(s) of the test result(s) as well as the measurement values with unit.



Yi conducted the research on tools and libraries used for the project. Based on the flask template, the Open-source Flask Dashboard generated by the AppSeed platform on top of Black Dashboard was selected and set up for initial testing and concept development. The environment of the flash dashboard was set up with Docker. The dashboard includes a basic authorization module.

Haojie has established back-end services architecture by using Flask framework and implemented a proof-of-concept RESTFul api. The api is built with python, Flask framework and MySQL database. The api is successfully tested. The code is accessed via backend_poc branch: https://github.gatech.edu/gt-cs6440-hit-spring2022/Team-12-Connected-Patient-Experience/tree/backend_poc.

Apurva got acquainted with all the different features the team has been working on from the sample Flash template for the web application to the files in the GitHub repository. He will be working on the Python backend based off the backend_poc branch.

Challenges you have encountered.

The team had an addition of a new member this week. It brings a challenge to team dynamic and team collaboration style. We had explored several front-end framework and UI templates on the market and discussed a possible solution to adopting them.

The calendar in the format of HTML is static. Patients can only visualize the past and upcoming events on the calendar, instead of selecting a specific date and visualize the detail of the appointments.

Integrate MySQL to current template

How to select and filter the data to visualize, since there may be many test data. It is no necessary to show all of them.

Your plans moving forward.

Apurva & Haojie: backend development + integrate API into the frontend framework + simple visual display

Aijing: finish setting up mysql database + put together a list of logic for recommended services or tests

Yi & Baiyan & Shiyi: integrating the current codes in the Flask template

- Yi: integrate htmls (calendar+test results) into the current Flask template
- Baiyan: look into the template code, modify the appearance of the calendar, and look into how to generate a list of appointments besides the calendar
- Shiyi: make the current code into a class to generate htmls (multiple bars for different tests) and send to Yi, check the database to pick a patient profile to test the visualization. (Candidates, mainly)

For lab results visualization, the next step is to integrate the plot built by Plotly using Dash and Flask for generating the web application. In the coming week, our team will research on a few selected common lab tests, including blood sugar level test (What are names of the common tests for diabetes/high blood pressure?), xxxxx, and xxxxx. Based on the results, we will filter out the current database to highlight a few representing patient profiles, which will be used to test the functionality of the developed patient dashboard.

Updated timeline

We are on schedule for the planned tasks, including environmental and database setups. We have also started a few future tasks early, such as the visualizations for calendar visits and test results. The project timeline is updated to reflect these changes, mainly the shifts of the start dates for certain tasks. Accordingly, to integrate the visualizations, we have added one more week for the “environmental setup and proof-of-concept" task to ensure that the visualizations can indeed be integrated using the current Flask framework.