Sprint #2 - Design & Implementation Plan

Adarsh Koka
akoka7@gatech.edu

1 PROJECT DESIGN

1.1 Project Summary

The project is a web application that provides visualizations and data of the percent of deaths of different obesity related diseases depending on what the user selects. The main features of the application are selectors which will be presented in the form of checkboxes that the user can select for viewing data of the different obesity related diseases. The user should be able to select multiple diseases to view the all of the diseases in a visualization so that they can be compared. The user should also be able to switch between viewing data of deaths or diagnosed data of the diseases.

For example, say a user wants to understand the percent of people that were obese out of the people that died of cancer related deaths. The user should be able to select the cancer disease in the selector and then click on a "Deaths" radio button to view deaths. The web application should then display a visualization and data in tables presenting the number of people that were obese and died of cancer.

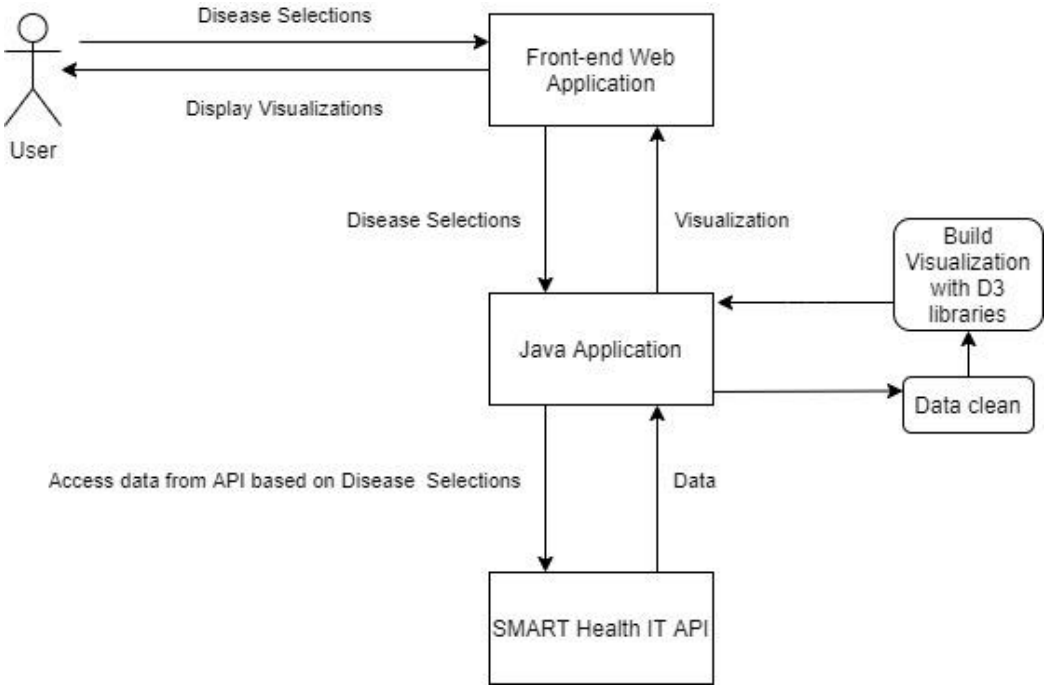
1.2 Tools and Technology

- Java as the programming language
- React for Front-end
- HL7 FHIR R4 Java Libraries
- D3 Visualization Library for visualizations
- Health Services Platform Consortium HSPC Client Java libraries
- Github

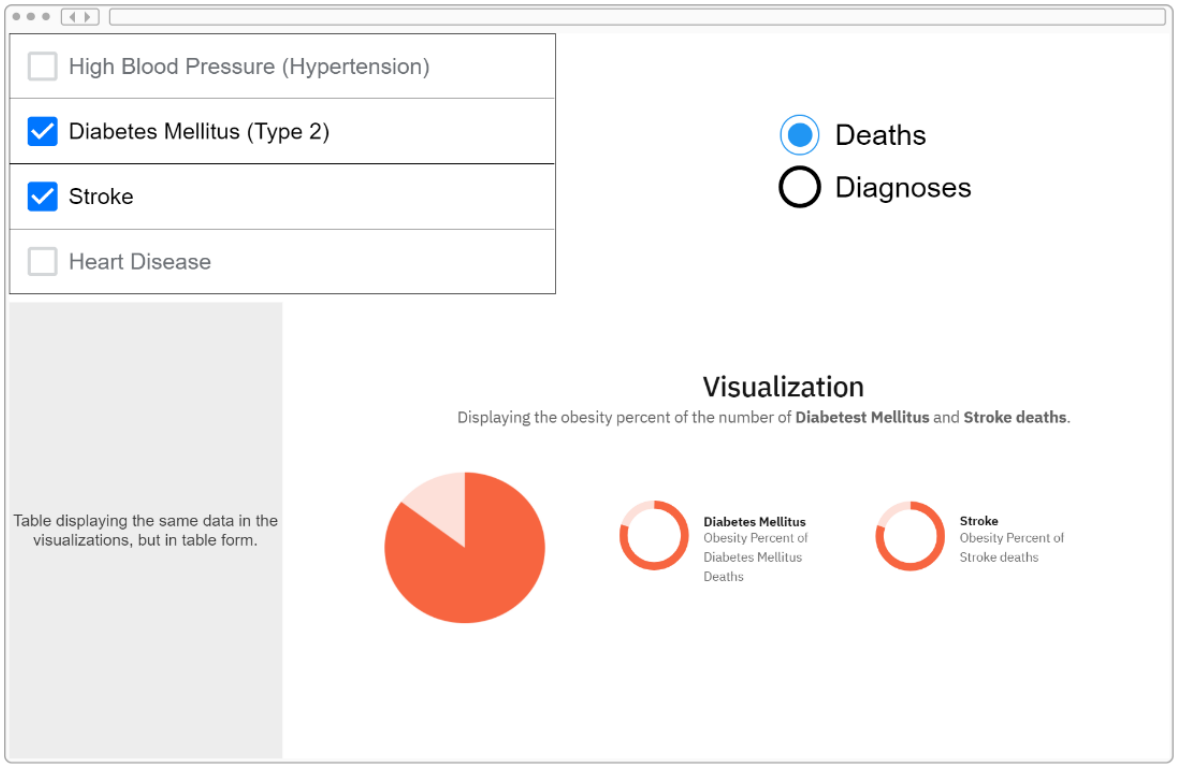
1.3 Data Sources

For data sources I plan on using the data available through SMART Health IT. I do not require a vast selection of data for my project because I only want to create a working solution at this time and the patient data available on SMART Health IT should work as a suitable representation (SMART on FHIR, n.d.).

1.4 Diagram



1.5 Screen Mockup



2 IMPLEMENTATION PLAN

2.1 Project Tasks

Week 11:

- Create Java Project
- Add HSPC Java libraries dependencies
- Demonstrate basic access to obesity data from API
- Demonstrate access to obesity data based on specific diseases

Week 12:

- Create classes such as “Disease"
- Create methods such as getDataByDisease()
- Clean data and put into arrays or whatever format that is usable in D3

Week 13:

- Set up web application host (use a github.io domain)
- Create selectors checklist for diseases
- Set up listeners for selected diseases
- Set up selected diseases as conditions in the API call

Week 14:

- Create radio buttons for deaths/diagnoses
- Set up listeners for deaths/diagnoses
- Set up condition for fetching data for deaths/diagnoses from API

Week 15:

- Set up D3 in Javascript, visualizations will be built through method calls
- Build Visualizations section of the web application
- Have D3 visualizations display in Visualizations section

Week 16:

- Stretch goal: Have table display numerical form of the same data
- Test different selected diseases
- Test diagnoses/deaths
- Ensure visualizations are displaying correctly based on user selections

2.2 Project Timeline

- Create Java Project (03/15 – 03/21)
- Add HSPC Java libraries dependencies (03/15 – 03/21)
- Demonstrate basic access to obesity data from API (03/15 – 03/21)
- Demonstrate access to obesity data based on specific diseases (03/15 – 03/21)
- Create classes such as “Disease" (03/22 – 03/28)
- Create methods such as getDataByDisease() (03/22 – 03/28)
- Clean data and put into arrays or whatever format that is usable in D3 (03/22 – 03/28)
- Set up web application host (use a github.io domain) (03/29 – 04/04)
- Create selectors checklist for diseases (03/29 – 04/04)
- Set up listeners for selected diseases (03/29 – 04/04)
- Set up selected diseases as conditions in the API call (03/29 – 04/04)
- Create radio buttons for deaths/diagnoses (04/05 – 04/11)
- Set up listeners for deaths/diagnoses (04/05 – 04/11)
- Set up condition for fetching data for deaths/diagnoses from API (04/05 – 04/11)
- Set up D3 in Javascript, visualizations will be built through method calls (04/12 – 04/18)
- Build Visualizations section of the web application (04/12 – 04/18)
- Have D3 visualizations display in Visualizations section (04/12 – 04/18)
- Stretch goal: Have table display numerical form of the same data (04/19 – 04/25)
- Test different selected diseases (04/19 – 04/25)
- Test diagnoses/deaths (04/19 – 04/25)
- Ensure visualizations are displaying correctly based on user selections (04/19 – 04/25)

2.3 Needs and Risks

For each task to work I will need to be able to get access to data from the SMART Health IT API. For the tasks concerned with getting data and setting conditions, it relies on the data for obesity (and the corresponding obesity related diseases) available in the API. A potential risk is that the time to build the data retrieval in the Java back-end taking extra time and then I would not have enough time for building the React front-end.

3 REFERENCES

1. SMART on FHIR javascript library. (n.d.). Retrieved March 21, 2021, from http://docs.smarthealthit.org/client-js/
2. Obaseki, Nosa. (2021, January 06). Using d3.js with REACT: A complete guide. Retrieved March 22, 2021, from https://blog.logrocket.com/data-visualization-in-react-using-react-d3-c35835af16d0/