4320 - Final Project

Cyclones

Table of Contents:

CS 4320 - Software Engineering 1
Dr. Sean Goggins

Sprint 1: Sunday, April 10th, 2022
Sprint 2: Sunday, April 18th, 2022
Sprint 3: Thursday, April 28th, 2022

Sprint 1

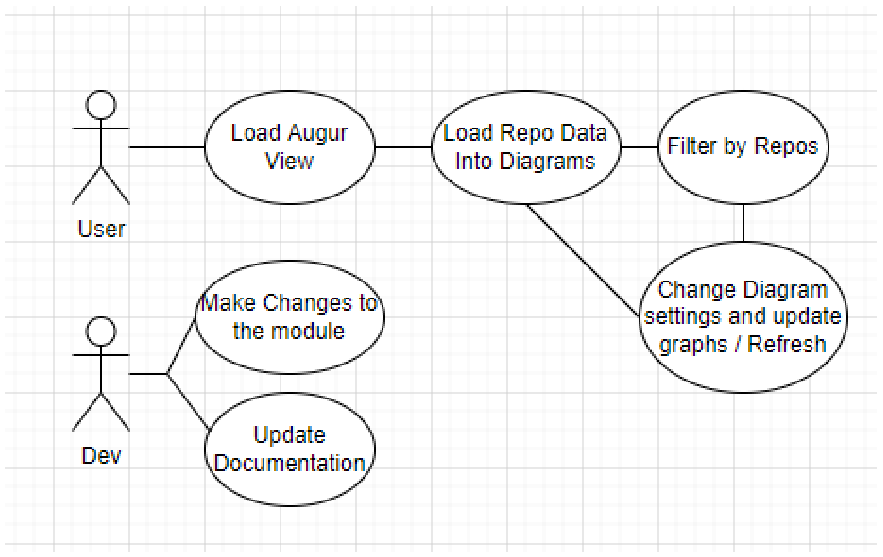
Important Links:

To access server:
- http://ec2-54-174-126-193.compute-1.amazonaws.com

To access Github Fork:
- https://github.com/BEBeach/augur/tree/BEBeach-Sprint-1

Basic Design:

Calendar:
- Based off of the Augur View project
- Main Goals:
    - Show Calendar view
        - Each shows the following data
            - Repo Name and data
            - Time Repo was accessed
            - Breakdown of server load
        - Keeps track of historical trends
            - Ratio_abs < 1 from issues
    - Show list of late repos
        - Repos that have not been checked in x time
    - Get statistics of server load
        - Provide data to detect trends in server usage
- Stretch Goals:
    - Create user profiles / Store data to be seen over time
- Use Case Diagram:


User
Load Augur View
Load Repo Data Into Diagrams
Filter by Repos
Change Diagram settings and update graphs / Refresh
Dev
Make Changes to the module
Update Documentation

Attributes:

PullRequest:

- Repo_id: int
- Repo_name: String
- Ratio_abs: Double
- Last collected: Date

Repos:

- Repo_id: int
- Repo_name: string
- Commits_all_time: int
- Issues_all_time: int
- Repos_status: string
- Url: string
- Color: string (not from json Data)

Requirements

User	**Activity**	**Associated Data**	**Description**
User	View Graphs made by Augur View	Module, Json data	The user can view specific data in the augur view.
Dev	Update documentation	Module	Can update the documentation of the module
User	Select Repo to view	Module, Json data	The user can select a repository on the augur view.

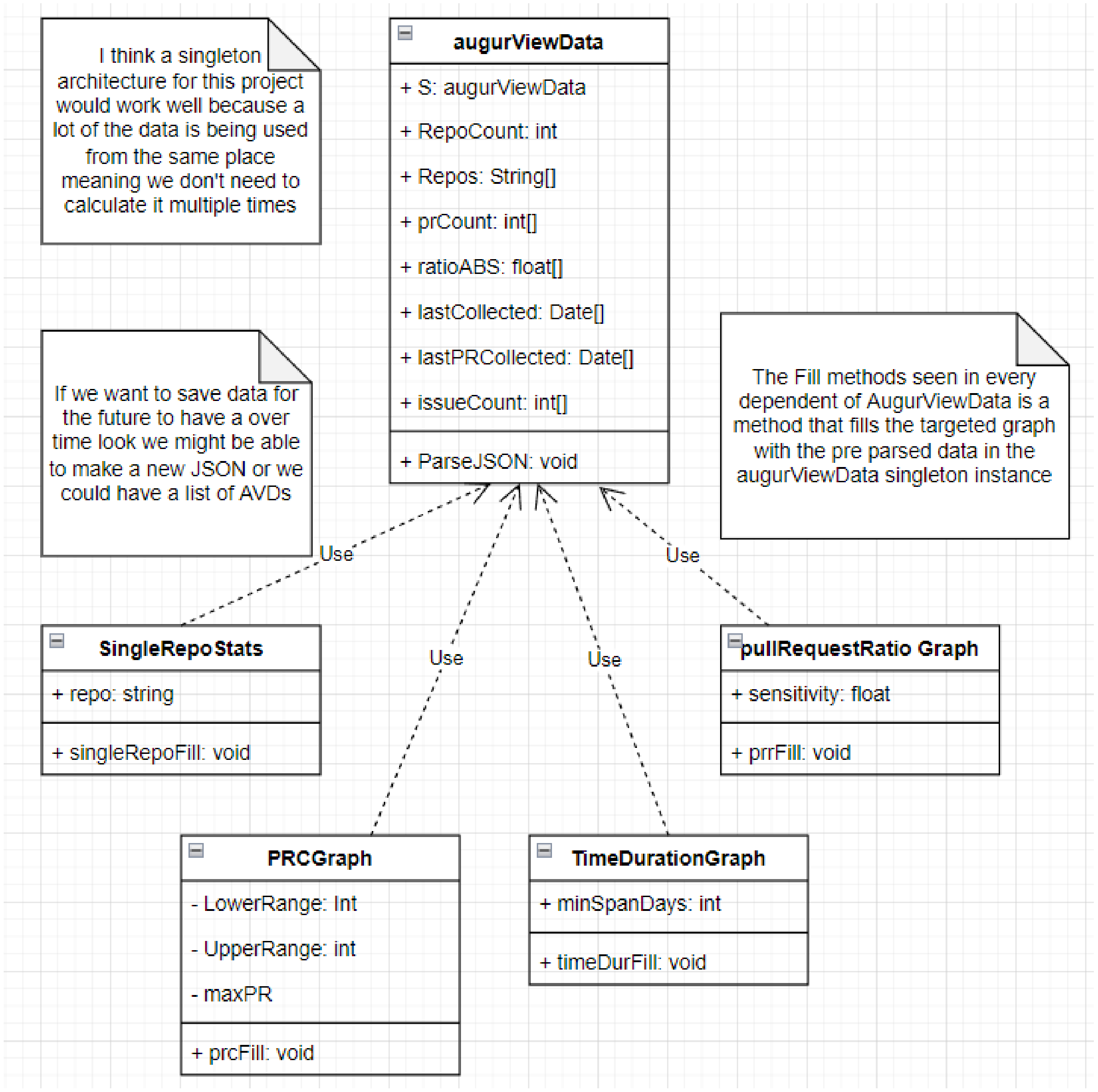
Sprint 2

Reworked Requirements:

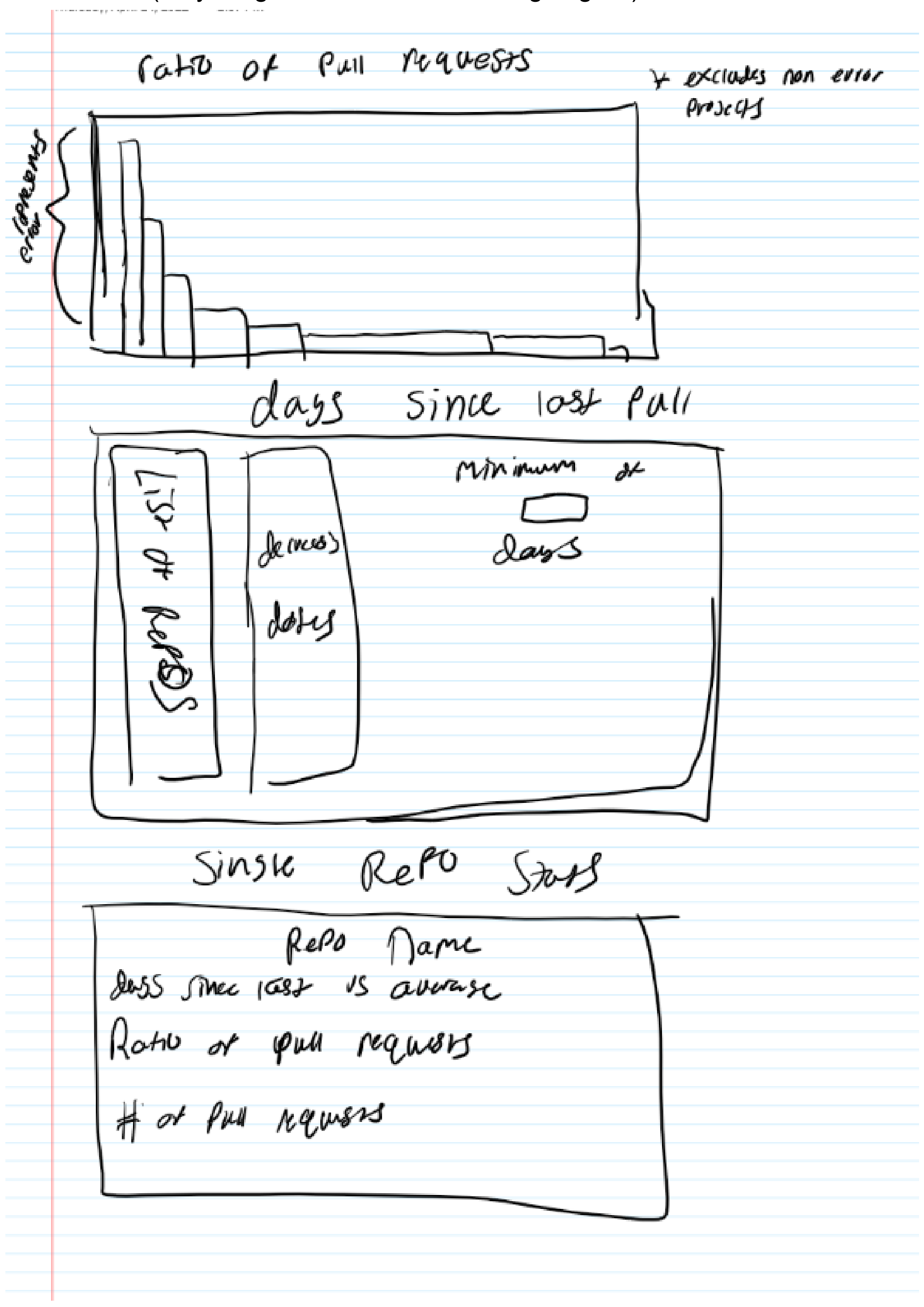
During this sprint, we decided to continue similarly to our original requirements; however, instead of making a calendar, we decided to show the data after the software has been run once. We will report any new issues that came up during the run, and we will still show the length of time that has passed between the run in question and its predecessor. This data will be shown for each repository.

Design Documents:

Data Structure

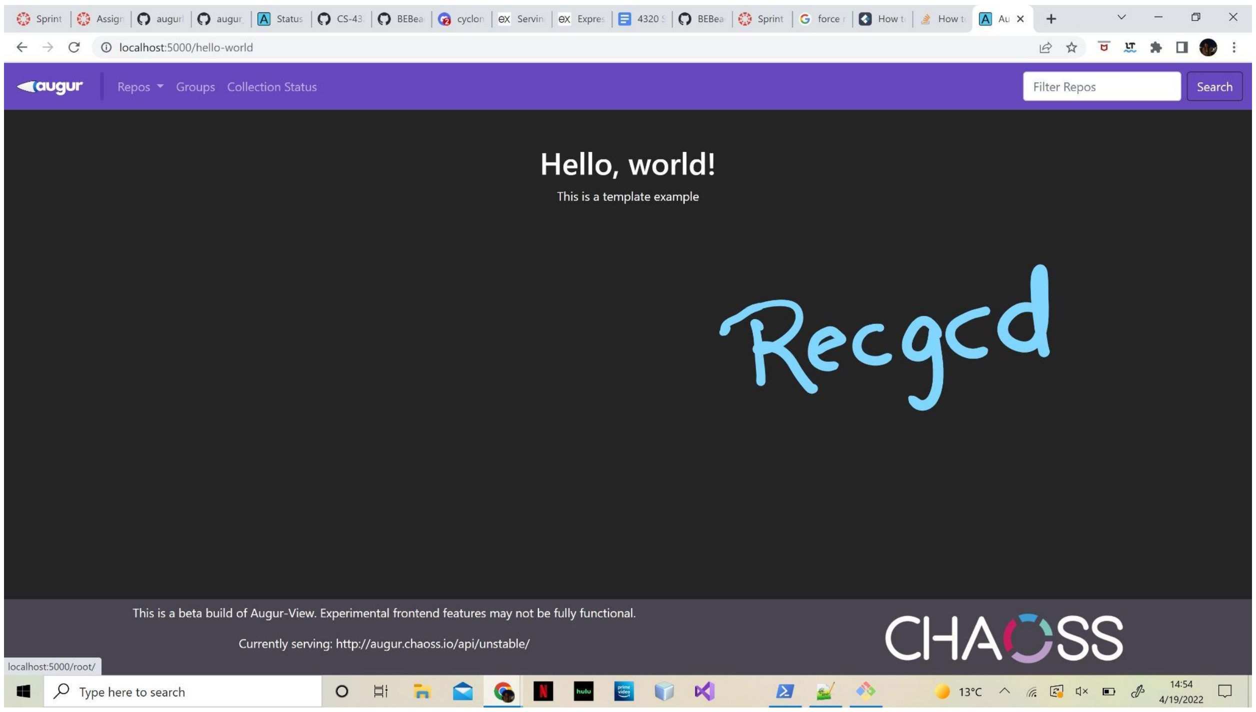

I think a singleton architecture for this project would work well because a lot of the data is being used from the same place meaning we don't need to calculate it multiple times
augurViewData
+ S: augurViewData
+ RepoCount: int
+ Repos: String[]
+ prCount: int[]
+ ratioABS: float[]
+ lastCollected: Date[]
+ lastPRCollected: Date[]
+ issueCount: int[]
+ ParseJSON: void
If we want to save data for the future to have a over time look we might be able to make a new JSON or we could have a list of AVDs
The Fill methods seen in every dependent of AugurViewData is a method that fills the targeted graph with the pre parsed data in the augurViewData singleton instance
Use
Use
Use
Use
SingleRepoStats
+ repo: string
+ singleRepoFill: void
pullRequestRatio Graph
+ sensitivity: float
+ prrFill: void
PRCGraph
- LowerRange: Int
- UpperRange: int
- maxPR
+ prcFill: void
TimeDurationGraph
+ minSpanDays: int
+ timeDurFill: void

New UI look (very rough draft of what we are going for)



Hello World Example:

This example demonstrates a locally hosted version of Augur View that has been modified. A live server implementation that pulls from the official Augur system will be implemented with help from the TA.


localhost:5000/hello-world
augur
Repos
Groups
Collection Status
Filter Repos
Search
Hello, world!
This is a template example
Recgcd
This is a beta build of Augur-View. Experimental frontend features may not be fully functional.
Currently serving: http://augur.chaoss.io/api/unstable/
CHAOSS
localhost:5000/root/
Type here to search
13°C
14:54
4/19/2022

To-do

- Name - Priority - Developer
- Launch Augur/augur view on remote server - 1 - Brian
  - Augur View Hello World
- General Graph Format - Olivia and Rachel
  - Order: least → greatest
  - Logarithmic view
- Pull Request Count Graph - 3 - Olivia and Rachel
  - General Graph Format
- Missing Pull Requests Graph - 1 Olivia and Rachel
  - General Graph Format
- Pull Request Ratio - 1 - Quinton Thuet
  - Remove all ones that have are >= 1
  - General Graph Format
- Issues - 4 - Brian
  - General graph update
- TimeLine - 2 - Jack
  - Complete Rehaul to make it more readable
  - Find greatest duration and list it
  - Click to expand list
  - General Graph Format
- Commit Completion - 5 - Brian
  - List incomplete collections
  - State if there are none
- Single Repo Status - 5 - Quinton
  - New Metrics to look at compared to averages

Sprint 3

Working Model:

- Go to https://github.com/BEBeach/augur_view to see our current codebase
- Finished Work
    - General Graph Format
    - Missing Issues Collected graph
    - Timeline
    - Single Repo
    - Completion Statuses
- To-Do
    - Final Review
    - Polish
- Changes made to each graph
    - Pull Request Count Graph
        - Sorted from least to greatest
        - Decided not to put this in logarithmic view because you can use the slider to zoom in on repos with less data.
    - Missing Pull Request Graph
        - Sorted from least to greatest
        - Put in logarithmic view
        - Switched the horizontal and vertical axes. We did this because it increased readability
    - Issues Graph
        - Sorted from least to greatest
        - Put in logarithmic view
        - Switched the horizontal and vertical axes. We did this because it increased readability
    - Timeline
        - Sorted from greatest to least
        - Click to expand
        - Shows only the largest 25% of values when expanded
    - Commits
        - Creates a list of all uncompleted collections
        - Shows a complete circle and completion message if all collections were successful

Design and Requirements Updates:

There are no major design or requirements updates from the Sprint 2 Design

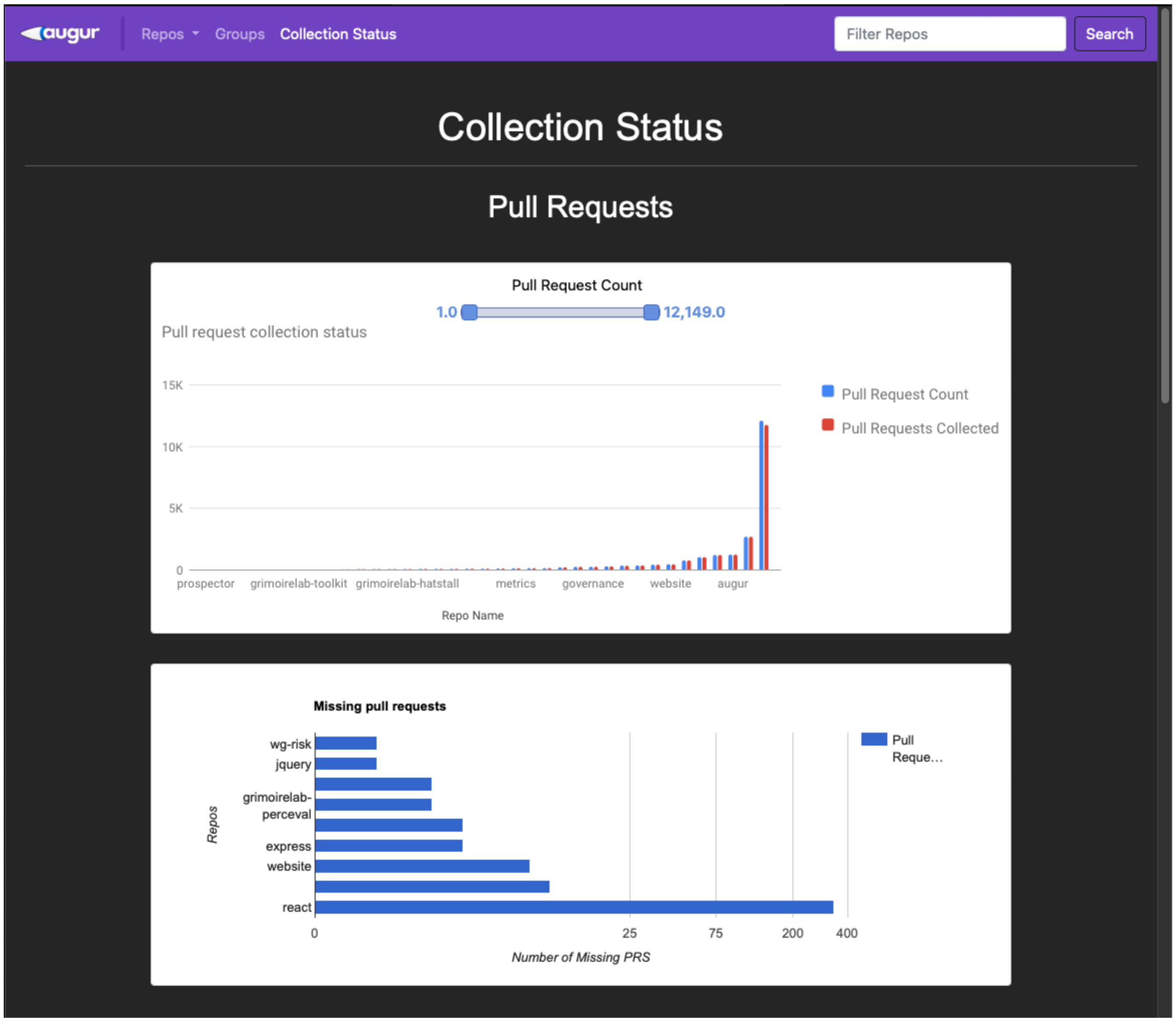
Testing Plan:

Human Test Case:

- Did data load successfully for you?
  - "Yes the data loaded."
- Was the data readable and well formatted?
  - "The data was formatted well. I did not understand augur so the data I was looking at was a little bit confusing. Over all it was displayed well."
- Find the Pull Request Collections Status graph and make the view around 100 to 2000. Were you able to do this? If not, what were the issues you had?
  - "I could find the graph but I did not know what the slider was for. After it was explained to me I could use the slider."
- Can you find the Repo Issues Ratio Graph
  - "Yes."
- Can you find the Single Repo Overview? Is the data readable?
  - "Yes, the information was displayed very well"
- Any suggestions or comments?
  - "If I understood augur more I would be able to understand this data more easily"

Use Test Case

1. Download and install augur view from the listed github page
   a. https://github.com/BEBeach/augur_view
2. After running the instructions on the GitHub you should be able to see the following page



3. There should be the following sections under the status page
    a. Pull Request Count Graph
        i. This graph should show all repos with pull requests in the serviced Augur Server
    b. Missing Pull Requests Graph
        i. This graph should show all repos with pull requests with missing information in the serviced Augur Server
    c. Issue Request Ration Graph
        i. This graph should show all pulled repos where the absolute ratio between the issues collected and the total number of repo issues.
    d. Issues Graph
        i. This graph should shows the total number of missing issue requests from repos in the serviced Augur Server
    e. Pull Request Timeline
        i. This shows a list of repos with that have not been pulled in the largest amount of time

    ii. This list should have a click to expand option

f. Commit Completion Status
    i. This section should list all repos that have not been successfully pulled
    ii. If all repos where successfully pulled it should show a complete circle
g. Single Repo Status
    i. There should be a search bar that is populated with the successfully pulled repos
    ii. When a repo is selected it should show all of the data represented in graphs above and provide a link to the github

Tools and Links

- Github Forks
  - Augur View Fork
    - https://github.com/BEBeach/augur_view
  - Augur Fork
    - https://github.com/BEBeach/augur
- EC2 Instance
  - Make sure the server is running
  - To connect to server via browser
    - ec2-44-201-91-139.compute-1.amazonaws.com
  - To connect to server via terminal:
    - Contact Brian Beach at bebz4t@umsystem.edu
    - Download the following pem key
      - 4320-Server-1.pem
    - Run this terminal command with the complete file path for the .pem file
    - ssh -i "4320-Server-1.pem" ubuntu@ec2-44-201-91-139.compute-1.amazonaws.com
    - Should look like this;
      - ssh -i /Users/brian/Programming/4320/4320-Server-1.pem ubuntu@ec2-3-236-246-18.compute-1.amazonaws.com