Sprint 1:

Daniel Fratila, Cameron Day, JP Pierce, Ryan Sellers

Other Frontends:
https://sandiego-explorer.osci.io/overview
http://new.augurlabs.io/
http://old.augurlabs.io/

https://github.com/Dfratila/augur/tree/main

Requirements:

- Login
    - username and password encrypted storage
    - change password?
- User Interface:
    - Visualizations: unique contributors, tickets opened, average time to close ticket, average commits per user distribution, lines of code per commit,
        - dynamic graphs?
        - color theme in the graphs
        - explanation on each visualization to explain what the numbers mean
    - Able to compare between different repos
    - support link to submit bugs
    - light mode/dark mode
    - main page to default to
        - search bar for repos
    - save/favorite repos
        - list of these at the main page
    - export to pdf?

System:

- Augur backend
- security/encryption
- database for repos

Requirements Spreadsheet:

Login	Username, Password	Allows a user to login (or create an account) so that saved repos can be accessed

		and preferences saved.
Add repo	Repo name, info from the database (ticket info, contributor info, commit info)	Adds a repo to look at the data from it through a search from the main page
Compare repo	Repo names, repo info	Adding another repo will allow the user to compare the info points.
Submit bug report	Username, bug info	users will be able to submit a bug report which would notify the developers about the bug.
Switch theme	username, theme	users will be able to switch from light to dark mode with the preference being saved to the user.
Save repo	Username, repo	Users will be able to save repos so that they are easily accessible in the future.
Export data to pdf	repo, repo info	Users will be able to save a repo's info to a pdf, allowing for easier sharing of the information to send thru email, print, etc.
Change password	username, password	Users will be able to change their password if they forgot it

Use Case Diagrams (at least one):

Add repo:

Description:
The user is able to add a repo to show the information of the repo (ticket info, contributor info, commit info, etc.) which will be shown graphically with various types of graphs relevant to the specific info.

Triggers:
The use case will start with the basic homepage and will be triggered when a repo is added through the search bar or clicked from the saved repos list.

Actors:

Users

Preconditions:
User has searched for a repo or has at least one saved repo in their list.

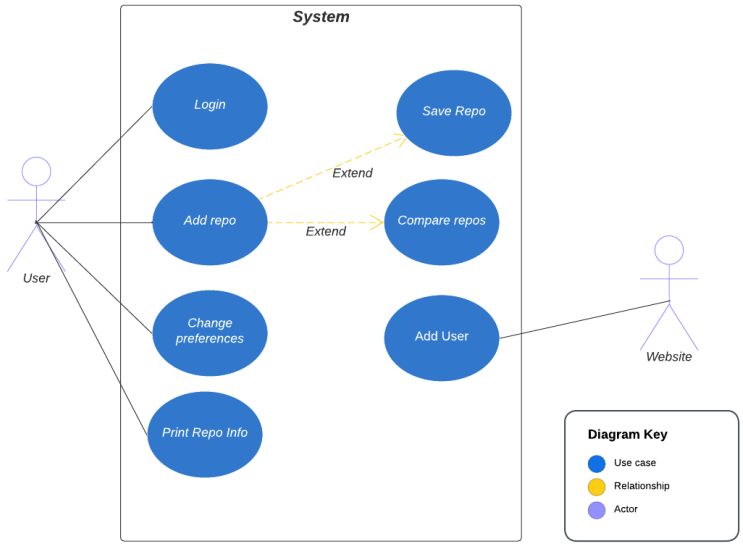
Main success scenario:
The repo shows up on the page with all of the visualizations working and set to their initial values, the information is accurate and up to date, and the user is able to interact and view these visualizations.

Fail conditions:
The repo cannot be found (private or does not exist), the visualizations do not display the information correctly or at all.

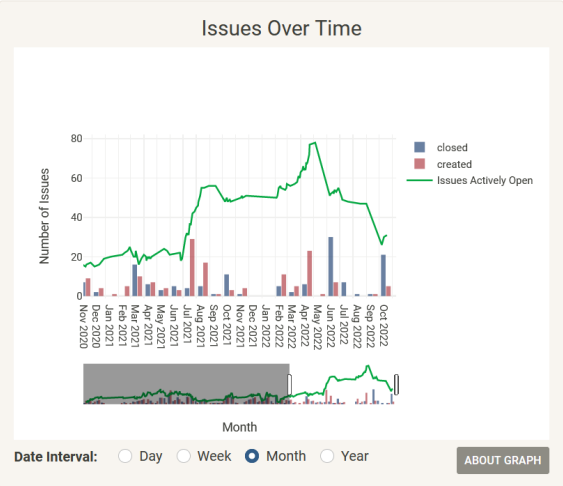
Steps of execution:

1. login
2. search for repo or click into saved repo from list
3. website retrieves repo info from database
4. website displays info in a graph
5. graphs are interactable
6. (optional) add another repo to compare the two

Use case diagram:



Dependent use cases:
compare repos
save repo
print to pdf
Description/Sketches of Frontend:



We like the San Diego Explorer frontend (pictured above) so we will probably take some ideas from that as the dynamic graphs and the color aspects of it are very appealing and greatly increases the usability and readability of the information.

We will have a few pages, one will be a home page which will be populated with the user's saved repos and a search bar for repos. They will be able to view quick basic analytics about each repo saved.

Once a user searches for one or more repos it will load either the single graphs, if they searched one repo, or the comparative graphs if they searched for at least two. The graphs will be in individual windows in a grid categorizing the data. It should be easy to understand how the data from different repos relate, as well as isolate the data for one repo at a time.

There will also be a page for a list of repos for the user to pick from as well as sort and filter the list to their liking to compare similar or related projects.

All graphs and charts should be able to be displayed in different formats/ scales, as well as be downloaded off the page. Since there is a time constraint on this project, the graph/ data manipulation won't be super in-depth, but it'll be enough to be able to view data in a different way.

The UI will try to strike a balance between being user friendly and accessible, as well as a high level tool. Our design philosophy will be leaning towards a more high level tool if some ease of use needs to go by the wayside, since the target audience for this site will be very tech literate and shouldn't have a hard time figuring out more advanced menus.

The main goal with our UI is to make the San Diego Explorer site more user friendly and easier on the eye like new.augurlabs.io while retaining the functionality.

Dev Process:
Agile style development and component based web dev.
We want to move relatively quickly and we don't need that much planning time so Agile feels the most viable. The first "sprint" will set up the repo in manageable chunks that we can do easily. The second sprint will get the framework of the frontend set up and be able to add features to it easily in the next sprint. The third sprint will add most of the features starting with the most crucial features like the data visualizations and adding and comparing repos. The fourth sprint will focus mostly on making the frontend look good and help it to be readable and intuitive along with adding more of the features. The final sprint should be reserved for final adjustments and cleaning up the code.