SOFTWARE REQUIREMENTS SPECIFICATION

Sweta Praharaj

Introduction

The document provides details about Requirements analysis for an Open Source Software Health and Sustainability Metrics Tool.

Software product overview

For our project, we'd like to create a website that uses existing endpoints from the Augur API, as well as new ones, to give insight to the health of a certain repo or repo group. Since this is a complicated and arbitrary determination, we will be focusing on a select few metrics that can help in determining overall code and project health.

System Use

- API – The Augur API is where we get our data from. We want to project data from this in a human-readable format.

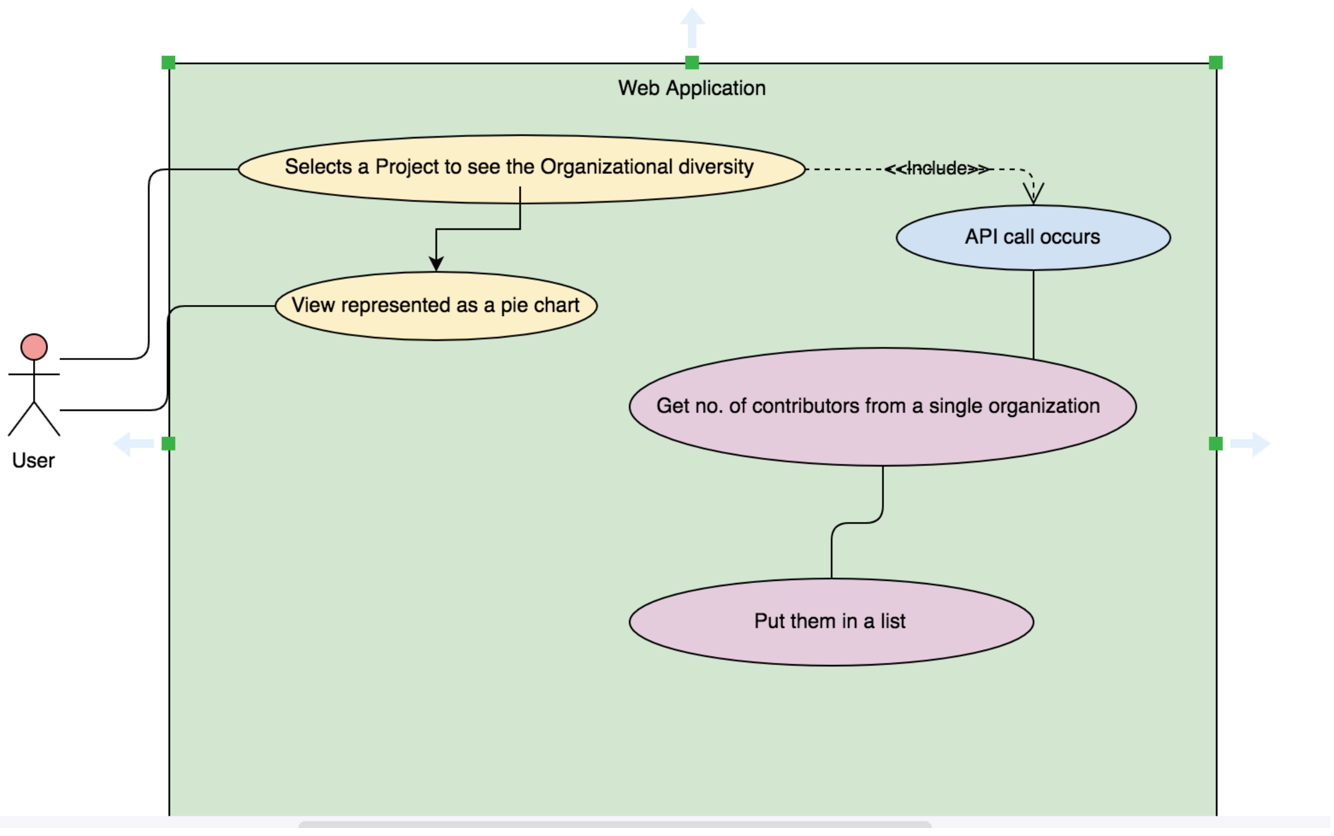
- DOM – The DOM is important when coding JavaScript, which since we are loading in data dynamically from the API, we will be using the DOM.

- User – The user is the person who is looking at our webpage and our data. They are the most important person because they are the reason we are making this project.

- Developer - The developer is who would be maintaining the site and taking care of any bugs that may arise.

Our system will be used by people who have an interest in the data that the Augur API collects. Our system will show this data in a human-readable format.

4. ***System Requirements*** (include 1 use case, a system functional specification, and a list of non-functional requirements)


Web Application
Selects a Project to see the Organizational diversity
<<Include>>
API call occurs
View represented as a pie chart
User
Get no. of contributors from a single organization
Put them in a list

USE CASE DIAGRAM FOR ORGANIZATIONAL DIVERSITY

System Functional Specification

We need to make various files that can be named as modules for different usage. We need to make

1. Service File(containing the Base API and the list of appended API calls)

2.View File(HTML and CSS)

3.View Logic Files(Javascript and other imported libraries of javascript)

4.API File(If we are creating new APIs)

<u>Non-Functional Requirements</u>

- Human-readable format
- Fast Client-side rendering
- Responsive and dynamic
- Platform-independent
- Free of bugs

<u>Design Constraints</u> (at least 5)

- The view components should look pleasing to the eyes of user.
- They should be easy and simple to understand with only the significant data displayed.
- They should have proper styling such that all the styling properties blend with each other fine to produce a good effect. If possible the components can have transitions and animations added to them.
- The UI should have standards of uniformity in all styling properties like color, fonts etc.
- The response data should be populated only after certain actions are performed by the user within a short span of time.
- If there is any server error, the errors should be handled in the front-end that notifies the user of non-existence of data.

Purchased Components

Servers: DATABASE: Postgre SQL, Augur.

Front-end: HTML,CSS,Javascript(along with other libraries)

Github for hosting the application.

Interfaces

The user selects the project it wants to see the metric: organizational diversity through a drop-down box that populates all the project groups from the Augur server.

The user sees the organizational diversity that is Ratio of contributors from a single company over all contributors, also described as: Maintainers from different companies,Diversity of contributor affiliation.

I plan to show the data in form of either a pie-chart or a dough-nut chart using Oracle JET or Google charts.

Identify what we will use from Augur ERD

augur_data.contributor_affiliations: ca_affiliation, ca_start_date