Brett Eckert

Assignment 9: Individual Requirements Analysis

Introduction

This document is a Requirement analysis for an Open Source Software Health and Sustainability Metrics Tool. The goal of this document is to specify requirements and details about this project. In this document we will outline the product overview, system use, system requirements, design constraints, purchased components, and interfaces.

Software product overview

With the ever increasing importance in open source software, it becomes more and more important to understand the health of these projects. This software production will develop metrics, methodologies, and software for expressing open source project health and sustainability. This project has several goals it aims to achieve. The first is to establish standard implementation-agnostic metrics for measuring community activity, contributions, and health. The second is to produce integrated open source software for analyzing software community development. Lastly, to build reproducible project health reports/containers. We hope to give open source contributors the power to know where they should place their efforts and know they will make an impact. We want to empower open source communities, attract new members, ensure consistent quality, and reward valuable members. This project will give companies the ability to know which communities and software to engage with, communicate the impact the organization has on the community, and evaluate the work of their employees within open source. Overall

this project achieves greater transparency of open source projects allowing others to make more informed decisions about how they want to engage with that open source project.

System Use

Project Manager: The project manager will oversee the software development process. They will create teams of developers, instruct developers on what tasks to complete, and evaluate development and maintenance across this project's lifespan. Their goal is to make sure the development process goes smoothly and coordinated.

Developer: The developer can have many roles. Overall the developer will be the one creating and maintaining the tools and website for the project. The developer does the initial creation of the project through web development, database design, and software design. Their goal is to work together as a team to create a seamless final piece of software.

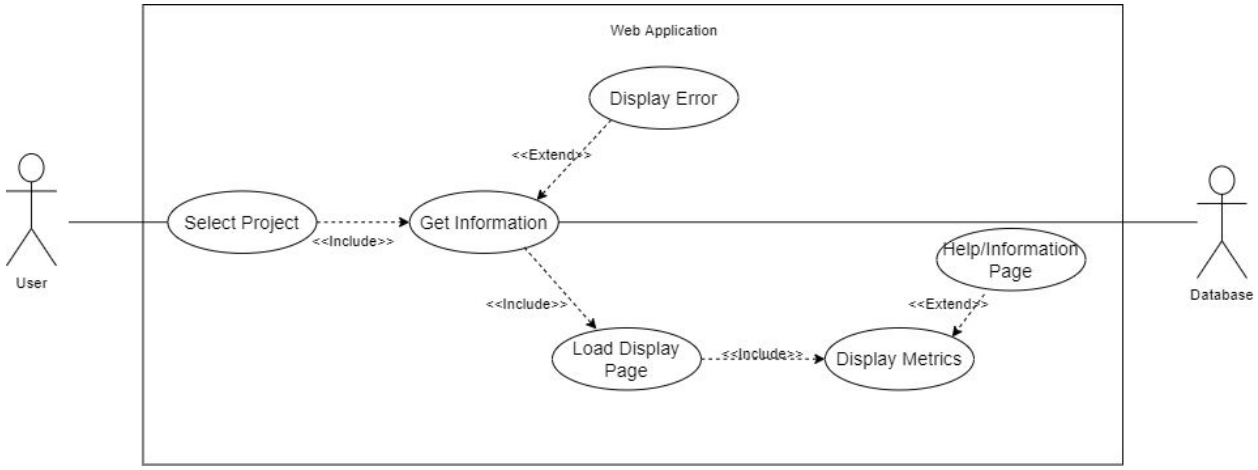
User: The user is the person using the metric tool for their own repository or viewing a website displaying statistics on other repositories. The project should be designed around this person’s experience when using the tools and site.

System Requirements

Use Case Diagrams:

Select Project

Diagram:



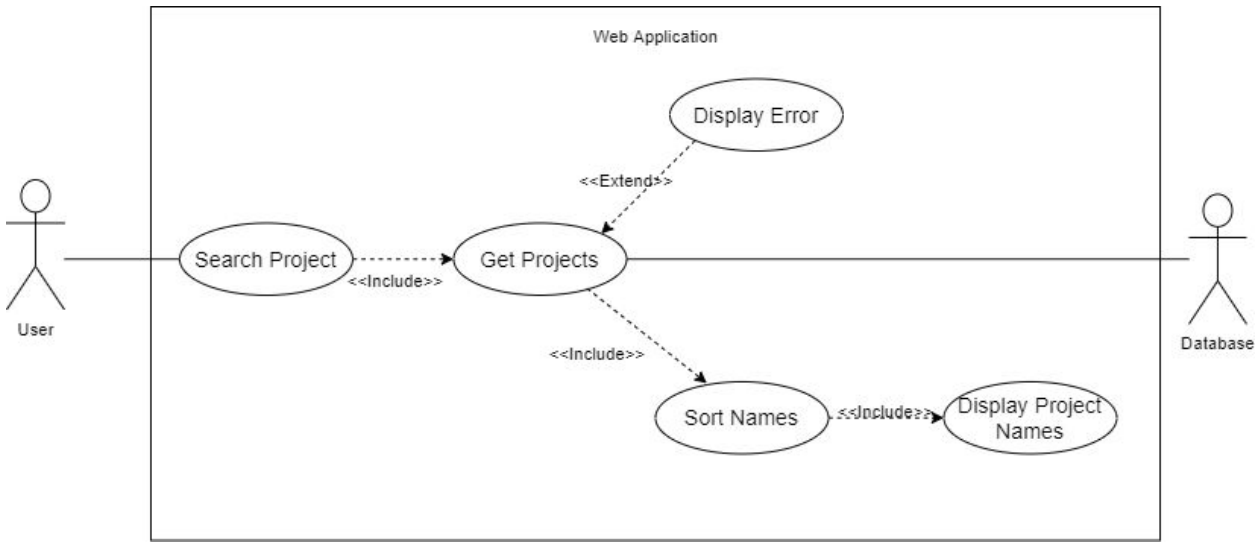
Description:

- System: Webpage
- Actors: User and Database
- Brief Description: This use case demonstrates the web application when a user selects a project from a list of projects to view data and metrics about that project.
- Flow of events:
    - 1. A user selects a project from a list of projects.
    - 2. The web application will pull the data from the database.
    - 3. The display page will begin to load and then the metrics will be displayed to the user.
- Alternate Flow of Events
    - Data Fails to Load

        - 1. A user selects a project from a list of projects.
        - 2. The web application will display an error to the user when trying to pull the data.
    - Help/Information Page
        - 1. A user selects a project from a list of projects.
        - 2. The web application will pull the data from the database.
        - 3. The display page will begin to load and then the metrics will be displayed to the user.
        - 4. The user selects the help/information page to learn more about the metrics and help deciphering a meaning.
- Pre-conditions: Must have a working internet connection.
- Post-conditions: The user will be on the display page and has the option to search for another project or navigate elsewhere.

<u>Search Project</u>

Diagram:



Description:

- System: Webpage
- Actors: User and Database
- Brief Description: This use case demonstrates the web application when a user searches for a project to view data about that project.
- Flow of events:
    - 1. A user searches for a project in a search bar.
    - 2. The web application will pull project names with names similar to the search.
    - 3. The web application will sort the projects by which ones match the closest to the search and secondarily by project popularity.
    - 4. The web application will display the names in that sorted order for the user to click on
- Alternate Flow of Events
    - Data Fails to Load
        - 1. A user searches for a project
        - 2. The web application will display an error to the user when trying to pull the data.
    - No Project Matches Search
        - 1. A user searches for a project
        - 2. Display no projects match the search
- Pre-conditions: Must have a working internet connection.

- Post-conditions: The user will be on the search results page and has the option to select a project and view it's metrics.

System Functional Specification:

- User functions
    - Search screen based on name of the project
    - Information page about this project to inform and teach user
    - View page for data pertaining to a selected open source project
    - Ability to see metrics on a specified open source project not in database
    - Comparison tool for multiple projects

Non-Functional Requirements:

- Usability
    - User friendly interface
    - Free of bugs
    - Standard web interfaces that are familiar to the user
- Reliability
    - Servers stay online 24/7, besides maintenance and special occurrences
    - Quick loading
    - Accurate data and information
- Performance
    - Handle expected user-load and expand server capacity as needed
    - Designed to work on all common operating systems (windows, mac, linux)
- Support

    - Code should be organized,readable, and standardized to allow for quick comprehension for maintenance, bug fixes, or new code additions. This will allow for quicker and more reliable support by the developers toward user problems.
    - All updates should be tested for accuracy and errors

Design Constraints

- The web interface should run on all major browsers (Edge, Firefox, Chrome, Safari) and work on their mobile counter parts.
- The interface should be easy to understand and navigate. The user should inherently be able to understand how to utilize the website for how they desire.
- The user interface should be pleasing to look at.
- The interface design and style should be uniform across all pages.
- Data should be returned to the user after certain actions are performed in a timely manner. The software should load data only the user selected to see.
- The user should be made aware when parts of the web page such as data are loading through loading animations and text.
- Any errors should be handled and displayed to the user properly.
    - Non-existence of data should inform the user on the front-end.
    - Server taking too long to respond should inform the user of the problem with the server.

- The design should contain no links to a foreign site that could potentially be harmful to the user.

Purchased Components

- Web Server
    - To host a website for the project
- Database Server
    - To store information and statistics on the open source projects

Interfaces

User Interface

Web interface will allow users to interact with the tools the project provides. They should be able to perform basic functions through this interface such as search for repos and view desired statistics about the repo. Asynchronous loading should be used, so the page doesn't need to fully reload when the user changes which repo data they wish to view.

Software Interface

The web application server will communicate with the database server.