

INDIVIDUAL REQUIREMENT ANALYSIS

JINGSONG TAN

INTRODUCTION

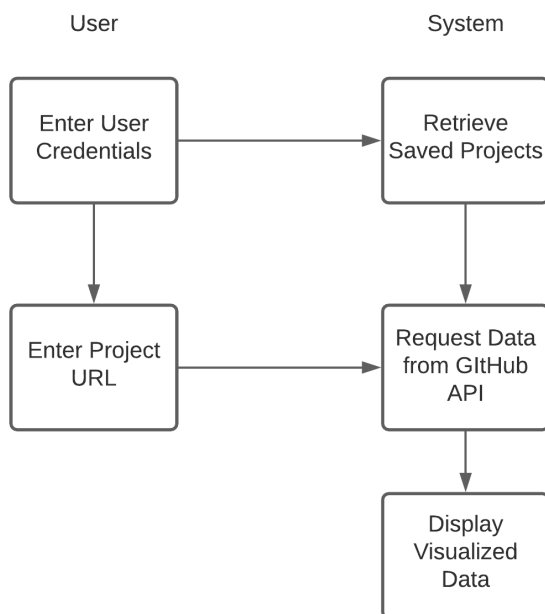
This system is built for technology companies and IT organizations to identify which open-source projects are more likely to be healthy and sustainable years to come. This system will provide metrics for measuring the health and sustainability of hundreds of thousands of open-source projects hosted on GitHub

SOFTWARE PRODUCT OVERVIEW

This software will be a web application. The system will pull information using GitHub API's to provide quantifiable metrics of open-source projects. A visual representation will be generated for each metrics to help compare different projects. The system also provides a convenient way of accessing previously viewed metrics.

SYSTEM USE

Users will be able to compare metrics for established open-source projects.



ACTOR SURVEY

USERS

Business managers, engineers, researchers, and students who rely on the performance and reliability of open-sourced projects could all be the users of this system.

Features:

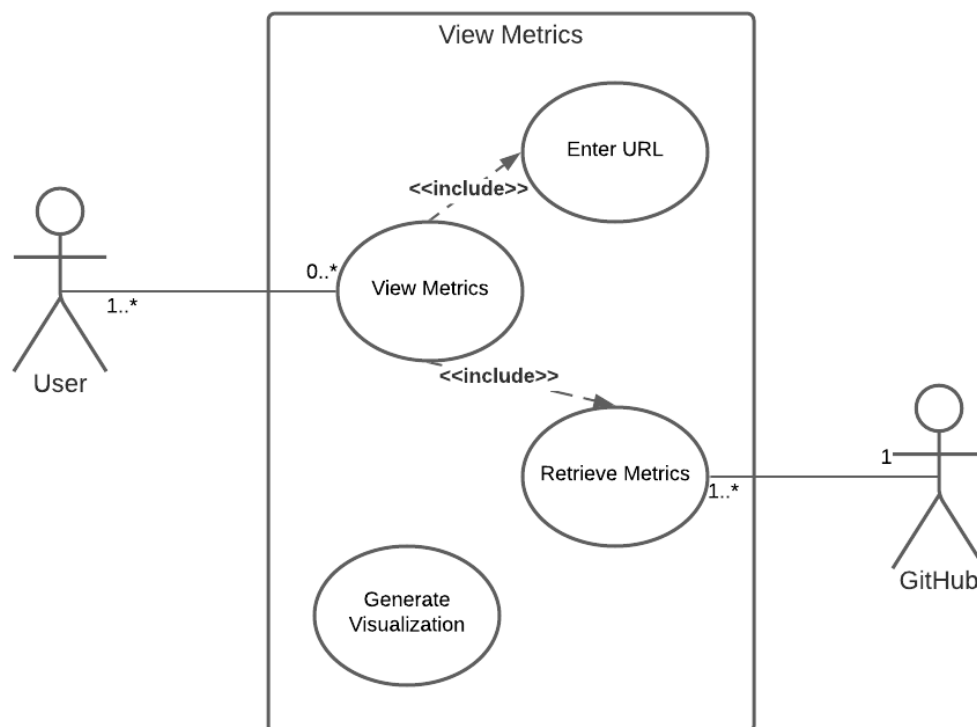
- Login
- View Metrics
- Compare Metrics

GITHUB

Open-sourced projects will be hosted on GitHub. GitHub will provide data that is necessary for generating health and sustainability metrics.

SYSTEM REQUIREMENTS

USE-CASE DIAGRAM



USE CASE: VIEW METRICS OF A PROJECT

DESCRIPTION

A user knows the link to an existing open-sourced project repository and wishes to view the health metrics of the project.

TRIGGERS

The user clicks the “View Metrics” button

ACTORS

Users of the software. (Researchers, Investors, Project Managers, Engineers, etc.)

PRECONDITIONS

- The user is logged in
- The user wished to view the metrics of a project.
- The open-source project is stored in a public git repository.
- The user has the URL of the project.

MAIN SUCCESS SCENARIO

The target project repository has been successfully accessed and health metrics are displayed with visual representations.

ALTERNATIVE SUCCESS SCENARIOS

N/A, there is only one success scenario for this use case.

FAILED END CONDITION

The software is not able to retrieve metric information with the provided URL, an error message is displayed.

EXTENSIONS

N/A

FUNCTIONAL REQUIREMENTS

1. A text field for users to enter URL
2. A “View Metrics” button
3. Display error messages for possible errors

4. The system shall pull data from the repository server
5. The system should display a chart for each metric
6. The charts should be organized in a tiled fashion

TECHNICAL REQUIREMENTS

1. Working webserver to handle user requests
2. The response time shouldn't be more than 5 seconds after the user pressed the "View Metrics" button

SYSTEM FUNCTIONAL SPECIFICATIONS:

1. Users should be able to create login credentials.
2. Users should be able to log in.
3. Users should be able to change their passwords
4. Users should be able to retrieve project information by providing a URL to the repository.
5. Users should be able to save project metrics to their account.
6. The system should save user data to a database.
7. The system should be able to call GitHub API to retrieve project data.
8. The system should be able to present a list of projects a user saved.
9. The system should be able to present a list of open-source projects with the highest health metrics measurement.
10. Users should be able to select 2 projects from a list of projects.
11. The system should be able to present metrics for two selected projects side by side.
12. The system should provide the total number of commits on the project.
13. The system should provide the total number of contributors to the project.
14. The system should provide a visual representation of the number of commits per week.
15. The system should provide a visual representation of the number of errors per week.
16. The system should provide a visual representation for lines of code by each top contributor.

NON-FUNCTIONAL REQUIREMENTS:

1. User interaction with the interface should feel responsive.
2. The system should retrieve and generate visual representations for a project in less than 5 seconds.
3. Users' information in the database should be secure and cannot be hacked.
4. The system should reliably work even when the user's network bandwidth is less than 1Mb/s.

DESIGN CONSTRAINTS

1. The system will run as a web application and be able to run on all major browsers including Chrome, Safari, Firefox, and Microsoft Edge.
2. Mobile browsing will not be supported as this system is intended for professional use only.
3. The database of this system will be developed using Mongo DB.
4. CRUD operations will be performed with REST API calls.
5. The user requests will be handled by a Node.js webserver.
6. The frontend of the web application will be using the Angular framework.

PURCHASED COMPONENTS

- A webserver to handle user requests.
- A database server to host user data.

INTERFACES

- Users will be interacting with the system with a website.
- The web server will provide REST APIs for the application to interact with data.

SAMPLE WEB INTERFACE

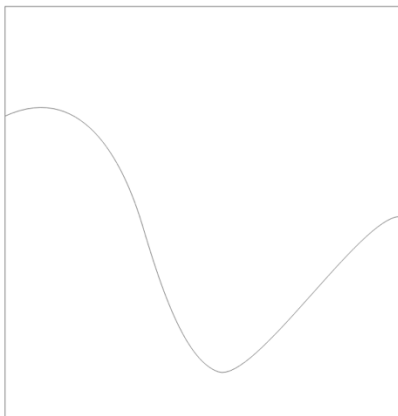
Metrics.Com

Show Metrics

Login

Project Name

Number of Commits over 7 Days



Number of Errors over 7 Days

