

Open Source Software Health and Sustainability Metrics Tool Requirement

Analysis

Prepared by

Jonah Marz

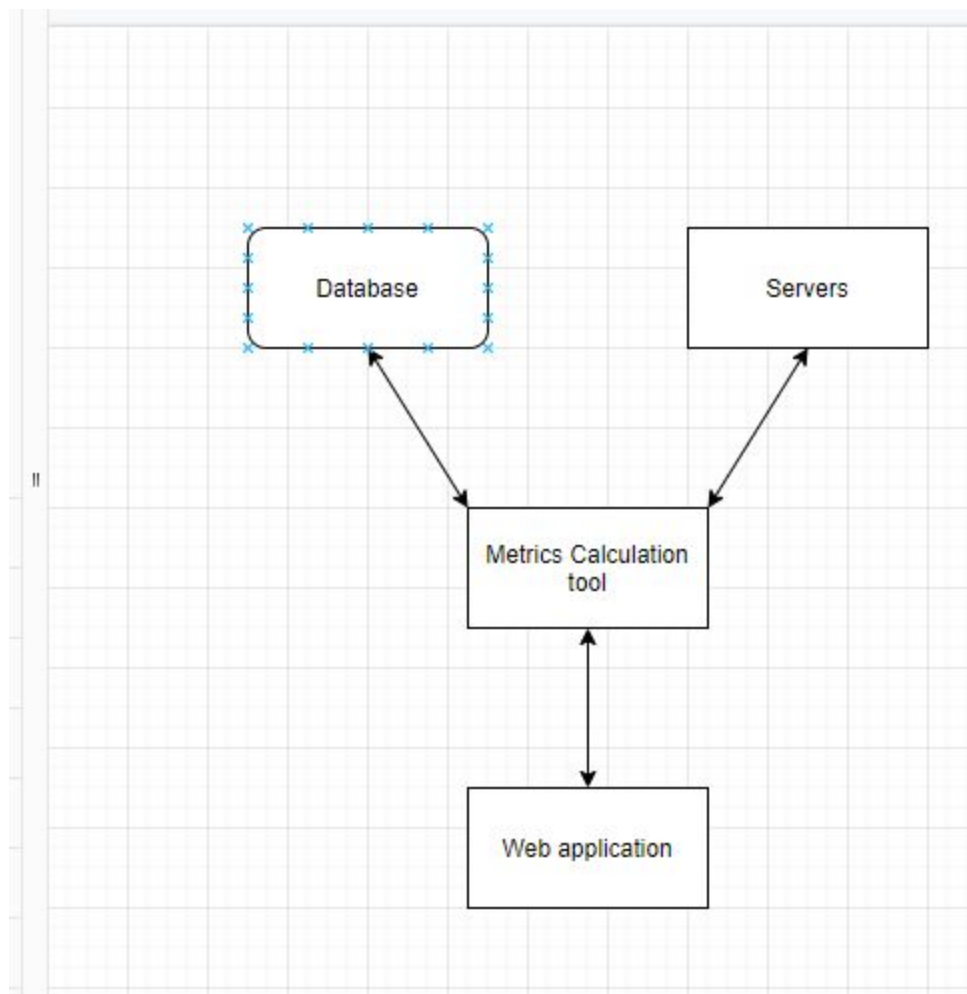
Introduction

- Purpose
 - This document is intended to lay out the requirements for an Open Source Software Health and Sustainability Metrics Tool. In summary this document will cover a brief overview of the product, system uses, system requirements, design constraints, purchased components, and an interface.
- Scope
 - This tool should be used to monitor the health and productivity of open source software projects in order to determine accurate predictions about the sustainability of said projects. Being able to accurately make these predictions will be central to the success of many technology driven companies.

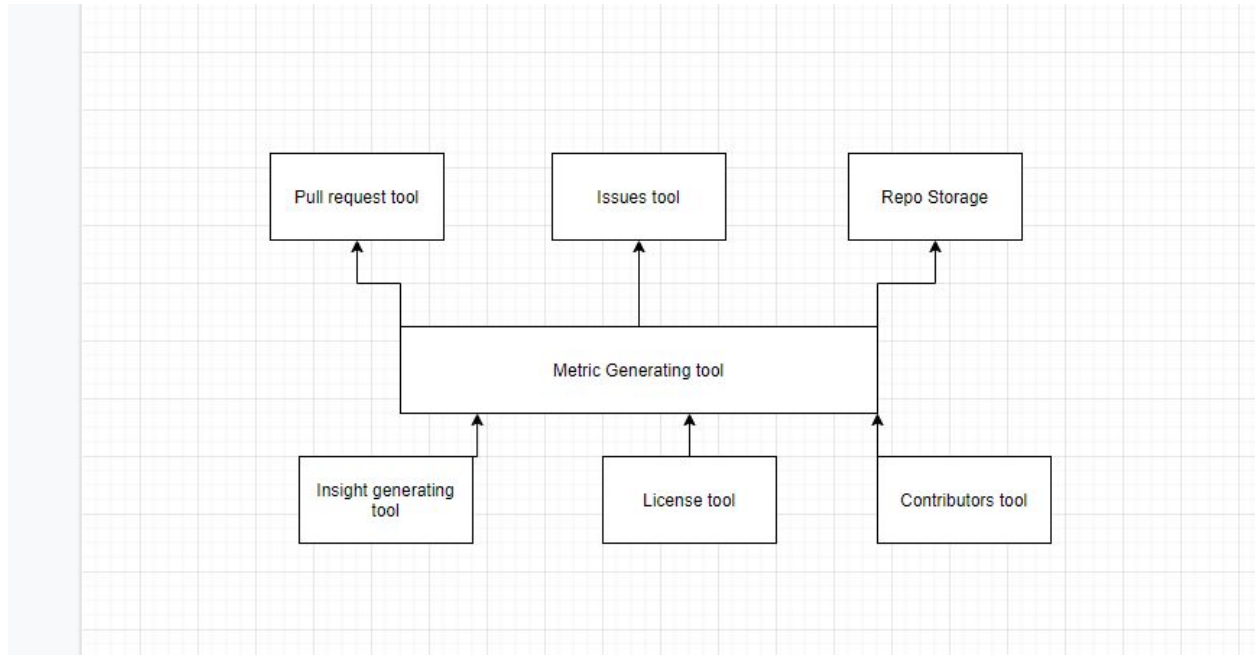
Software product overview

This section is meant to go over the basic architecture of this system. In addition this section goes over how the system works. It contains the system scope, the system architecture, and an overview of the system features.

- System Scope
 - This tool will be a web application mostly used for directly viewing the health of open source projects. The application will be able to provide metrics about the health and sustainability of open source projects.
- System Architecture
 - Physical system architecture



- Software system architecture



- Features overview
 - Provides Insights into popular or up and coming repositories
 - Can check what license a repository has attached
 - Can pull data about contributors to the repository
 - Can pull data about pull requests for a repository
 - Can pull data about issues in a repository
 - Stores various repositories and their related data
 - Uses a tool to generate figures and metrics with this given data

System Use

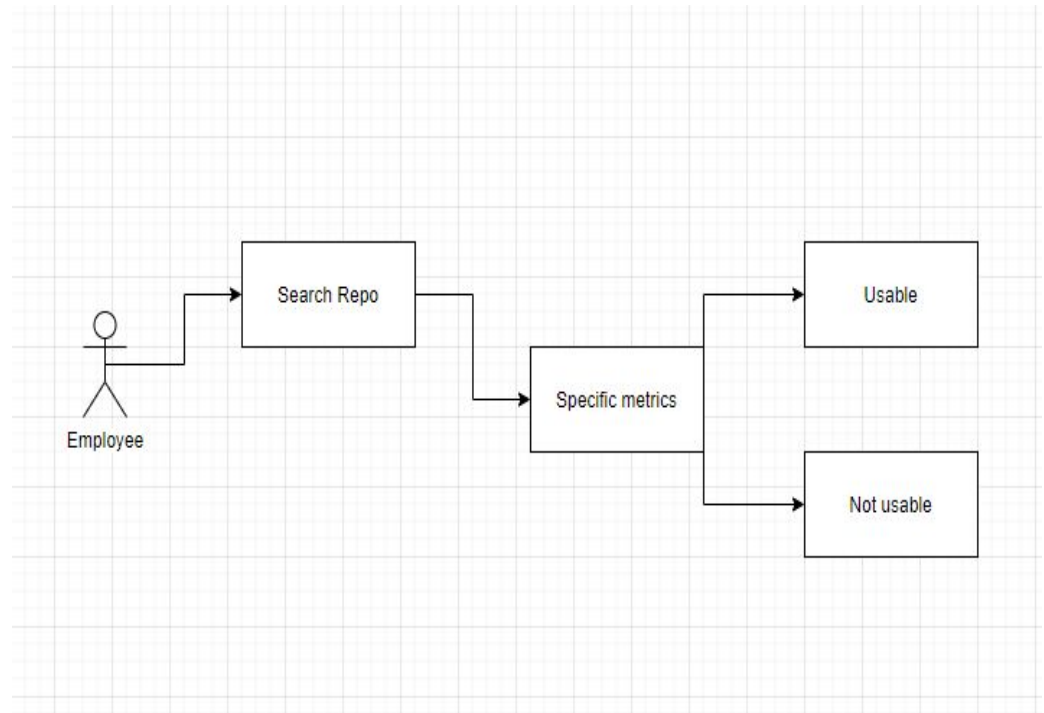
- Actor Survey
 - Manager

- Manager is responsible for using the metrics generated by the system in order to make decisions about open source software on Github
- Employee
 - An Employee should use the system to pull data into consolidated reports regarding open source software to present to a manager
- Customer
 - A customer receives the product generated by an employee and a Manager

System Requirements

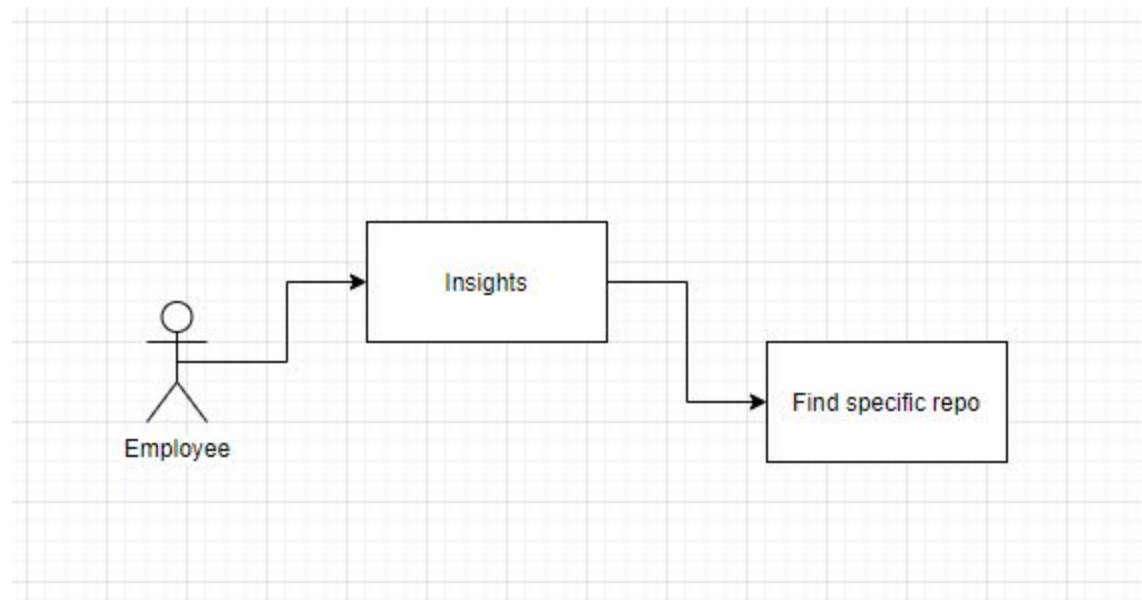
- Use Cases
 - Use Case 1

- Using metrics to determine the status of an open source project



- Use Case 2

- Using insights to see up and coming open source software



- System Functional Specification

- User Interface Functionality
- Data Pulling methods from websites like github
- Graph making with arbitrary data
- Updating repo list automatically
- Keeping track of the insights at a given time
- Giving a live feed of Issues and Pull Requests
- Non Functional Requirements
 - Reliability
 - The system needs to be up and running constantly in order to detect new data at all points
 - The system needs to be able to handle large amounts of repositories in its system at a time
 - Usability
 - User Interface should be concise and be easily accessible
 - Graphs should be aesthetic and easy to read
 - Capacity
 - The system needs to be able to handle a minimum of 1000 employees and a minimum of 100,000 simulations
 - The system needs to be able to handle at least 10,000,000 repositories at a time
 - Response Time
 - The system should always respond within 25 seconds

- Latency
 - A typical search should only take within 10 seconds
- Portability
 - The system should run on
 - Windows
 - Macintosh
 - Linux Environments
- Supportability
 - The system should be able to handle modifications automatically, and also keep backups for at least one year after release

Design Constraints

1. The system must be able to run as a web application on all major browsers (Internet Explorer Mozilla, Chrome, Microsoft Edge, and Safari)
2. The system must be developed using MySQL
3. The system must be hosted on an Apache Platform
4. The Apache Platform must be hosted from a Linux platform
5. The system must utilize modern frontend development practices such as Angular and React

Purchased Components

A component that this software will need to purchase is a server hosting machine to host our web application.

Interfaces

Here is an example user interface

