Individual Requirements Analysis

Project Name: Augur

Zach Burkholder

Revision History

Date	Revision	Description	Author
11/4/2019	1.0	Initial Version	Zach Burkholder

Table of Contents

1. Introduction

1.1 Purpose

This document is intended to specify the requirements and detail the software architecture of Augur. This Software Requirements Specification (SRS) will describe the external behavior of the application as well as the internal workings of the system. It will also describe the non-functional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements.

1.2 Scope

Augur is focused on prototyping open source software metrics. Functionally, Augur is a prototyped implementation of the Linux Foundation's CHAOSS Project on open source software metrics. Technically, Augur is a Flask web application, Python library and REST server that presents metrics on open source software development project health and sustainability.

2. Software Product Overview

This section provides an overview of Augur as a software product. It contains a summary of how the system works as a whole from all user perspectives. Included in this section is the system scope, the system architecture, and an overview of the system features that make up Augur.

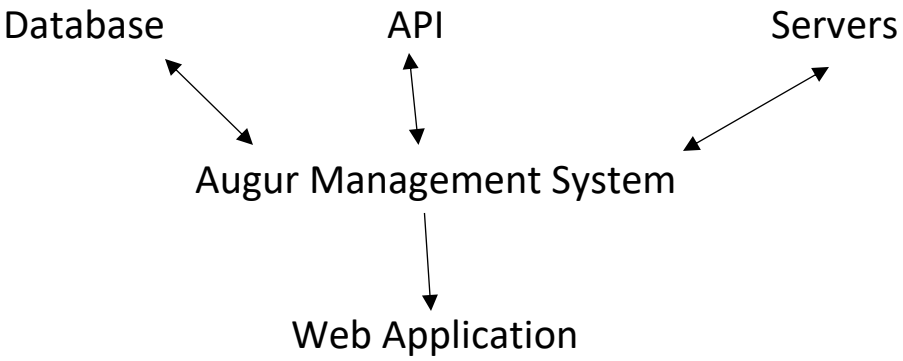
2.1 System Scope

This section proves the physical (architectural) system scope of Augur. Augur is a Python library and web service for Open Source Software Health and Sustainability metrics. It will provide data related to GitHub repositories.

2.2 System Architecture

This section defines the internal and external system architecture of Augur.

2.2.1 External View of Software Product



2.2.2 Internal View of Software Product

?

3. System Use

3.1 Actor Survey

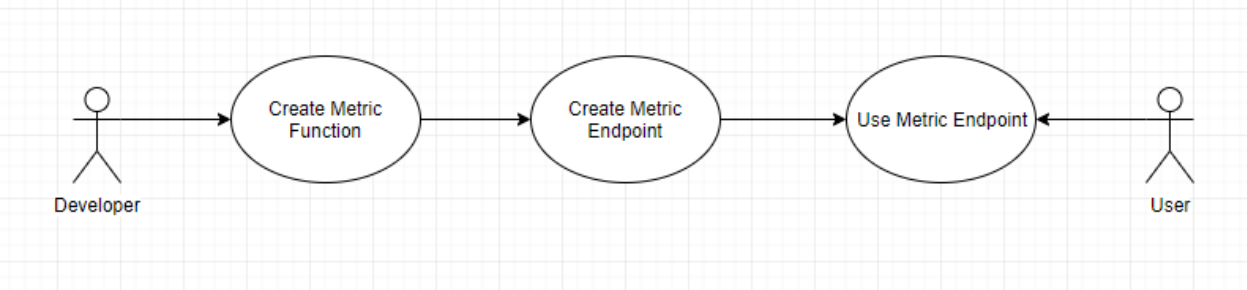
Systems Administrators receive help requests related to issues with Augur.

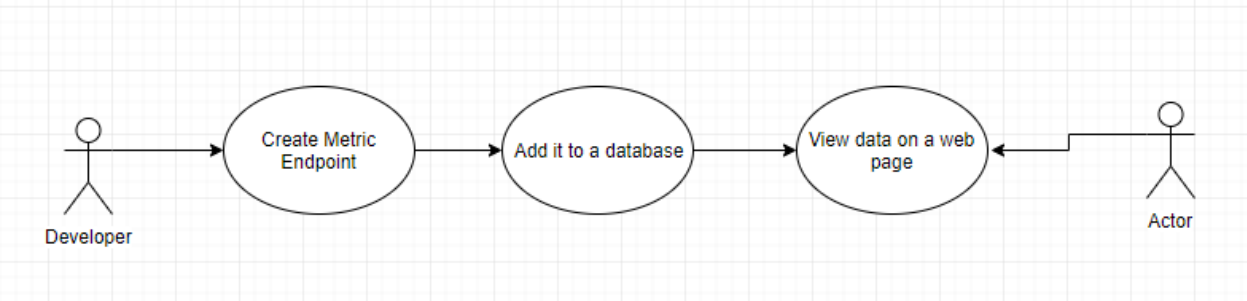
Developers are in charge of creating new metric functions and metric endpoints for the API. They are also responsible for the database and the server for the backend, and the web application on the frontend.

Users are the ones using Augur to receive data related to specific repositories in GitHub.

4. System Requirements

4.1 System Use-Cases





4.2 System Functional Specification

Users can select different repositories or repository groups, select an endpoint, and generate data based off of these selections.

Developers can do what Users can do, but they can also create their own metric functions and metric endpoints, alter the database, and alter the web page that is displayed for users.

4.3 Non-Functional Requirements

The web site must be compatible with the top web browsers (Chrome, Firefox, Microsoft Edge, Internet Explorer, Safari, etc.).

All users are required to be familiar with operating a PC. The expected education level of users should be of high school level.

5. Design Constraints

1. Database Management System (DBMS)

2. Web server software

3. Client computers with web browser and Internet access

4. Physical or virtual server(s) with compatible OS installed to host the web server and DBMS with network and internet connectivity

5. API endpoints

6. Purchased Components

The server that is hosting the web server and DBMS.

7. Interfaces

User interface on a web page where the user can select different repos/repo groups and an endpoint to generate data accordingly.