Software Requirements Specification

Project: Augur

Table of Contents

1. Introduction

Augur is focused on building human centered open source software health metrics defined by collaborations with the Linux Foundation's CHAOSS project and other open source stakeholders. Combining several of the different information of the “health” of the different open source projects should fulfill clients’ needs to be able to view an easily viewable and compact summary of the different projects that they are managing or a part of. This should take away some of the work for compiling information that clients want automated.

This document is intended to specify the requirements and detail the software architecture of Augur. This software requirements specification (SRS) document will cover the external behavior of the software, the internal systems at work inside of Augur. Also, this will go over the design constraints, non-functional requirements and comprehensive description of this software.

2. Software Product Overview

A. System Scope

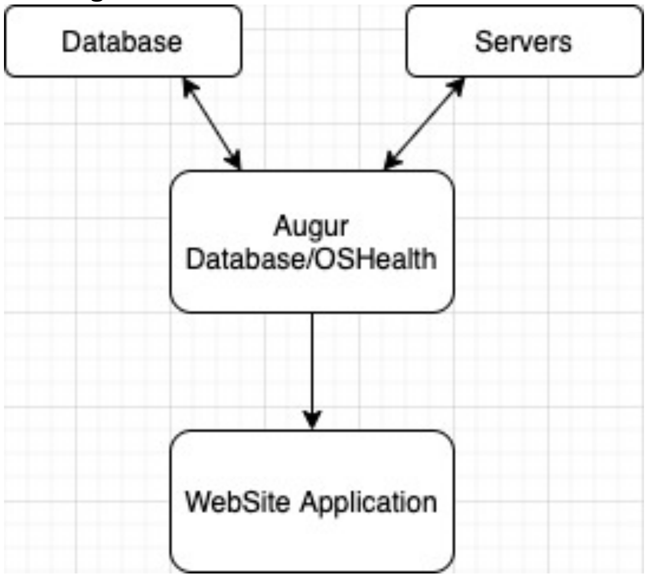
Augur is an application built to be able to determine the “health” of the different open-source projects that are in the system. Augur is available as a web application that automatically reads data from open source software projects and utilizes and displays health metrics based on those projects, with the goal of making sense of data.

The key strategies used by Augur in terms of making sense of data are 1.) Enabling comparisons, 2.) Make time a fundamental dimension in all metrics, 3.) Make all data driving visualizations downloadable as .csv, 4.) Make all visualizations downloadable as .svg

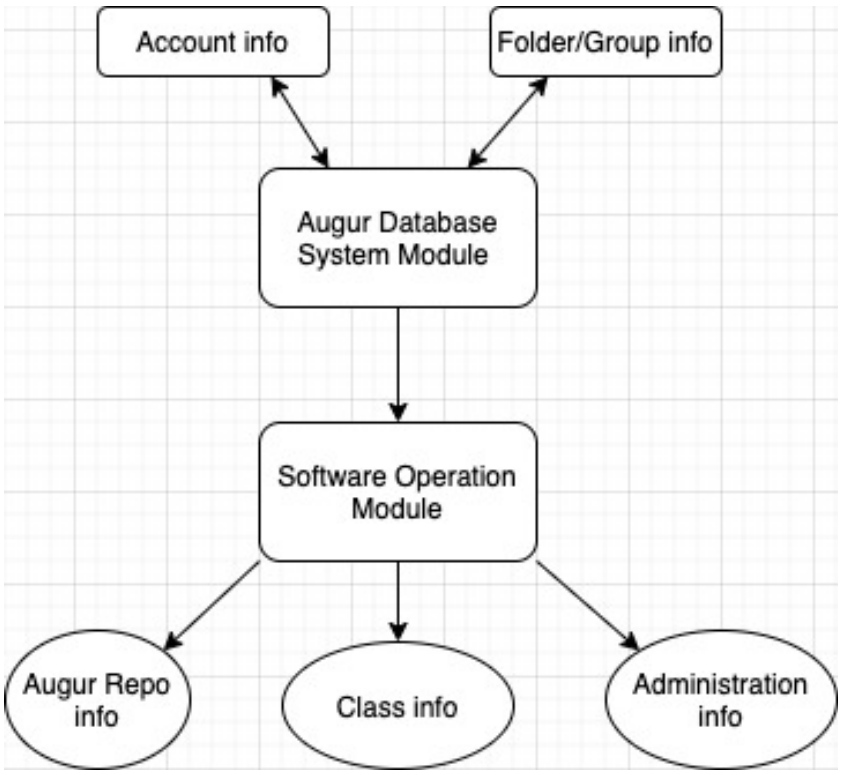
B. System Architecture

The system is designed as a web application that gets data from the projects that are to be analyzed with the internal and external architecture of Augur below.

Starting with External:



Internal design:



C. Attribute/Features Overview

This an overview of the features that user of Augur, and the developer who can contribute to repos/groups to Augur. For better understanding the section is divided into users and developers.

Users:

1. Overview - see analytics like lines of code added by different members, license coverage, top 10 authors, reviews per week, new issues per week, etc.
2. Compare Risk Metrics - compare different repos to each other by risk metrics.
3. Create/delete repo group- Creates a repository group, and allows you to add/delete repositories to/from that group.
4. Select Repo - allows user to choose which repo in the group to view
5. Store Repo- This feature allows the user to store a repo that they are a part of to be able to use currently stored ______ functions below.
6. Insights - not fully implemented, but will provide insights about different repos and allow comparisons
7. Compare Overviews - compare different repos to each other in the overview section.

Developers:

1. Stored Repos Sort- This allows the developer the ability to sort by the previously explained statistics from the currently stored repos function.
2. Make Docs - make and edit documentation of Augur
3. Testing - run testing on all aspects of Augur
4. Make Clean - remove logs, caches, and other junk
5. control the starting and stopping of the server
6. add groups of repos to the list of stored groups
7. start frontend and backend servers together

3. System Use

This explain how user can install and use Augur open software project. The installation enables a user to create all of the analyts and visualizations of data the user needs. Lastly the user can also build personal metrics to be included in augur open software ware community.

- Actor Survey's

Users: will install Augur system. This will enable them to view the collection of data documentation, optionally install the database schema and sample data, and the Augur frontend and its dependencies and generate the configuration file. hence the user can:

- Create user
- Add repo groups
- Store and remove Repo
- Start, Pause and Stop Repo
- Create Metric
- Test and Run Augur

4. System Requirement:

- **Case 1**

Case Name	Run the backend Serve
System	Augur
Actor	User
Description Summary	This describe logs in terminal and how user run backend server
Basic Flow of Events	1.Start running Augur (augur run) 2. then backend server starts up 3. then logs will appear in the terminal
Pre-conditions	User needs to be in augur directory to run database command

Pre-conditions	With command ctrl-c log will appear in terminal

- **Case 2**

Case name	Add Repo (UC01)
System	Augur
Actors	User
Description Summary	This case talks of how a user can add repos to augur
Basic Flow of events	1. Create/ set up all repo, outline all groups they belong. 2. Run augur with database add repo 3. The repo are added to their group
Pre-conditions	User Needs to in Augur directory to run the Augur Commands
Post-condtions	Specific forma is required Eg: <repo_group_id>, <repo_url>

- **System Functional Specification**:

Augur is a web application; the system architecture will be client-server.

User terminals

- T1: run augur backend servers
- T2: add repos
- T3: add repo groups
- T4: get repo groups
- T5: util shell
- T6: install project dependencies
- T7: clean project
- T8: rebuild project
- T9: start frontend and backend servers together
- T10: log frontend and backend
- T11: start the frontend server
- T12: start all installed data collection workers
- T13: check status
- T14: test
- T15: make docs
- T16: create a metric

Users Frontend:

- T17: it should possible for user to compare repos
- T18: fundamental part of all metrics should include TIME
- T19: Visualizations data should be downloadable as .csv
- T20: all visualizations should be downloadable as .svg

Non-Functional Requirements

- NF1: Augur must be open source and readable by users
- NF2: Augur must be readable and make sense to users
- NF3: Augur must not fail to run backend servers
- NF4: Augur must not fail to run frontend servers
- NF5: Augur must allow users to read and edit documentation
- NF6: Augur must allow users to create new metrics
- NF7: Augur must function on all modern browsers
- NF8: Users must be knowledgeable of command line, directories, Python

5. Design Constraints

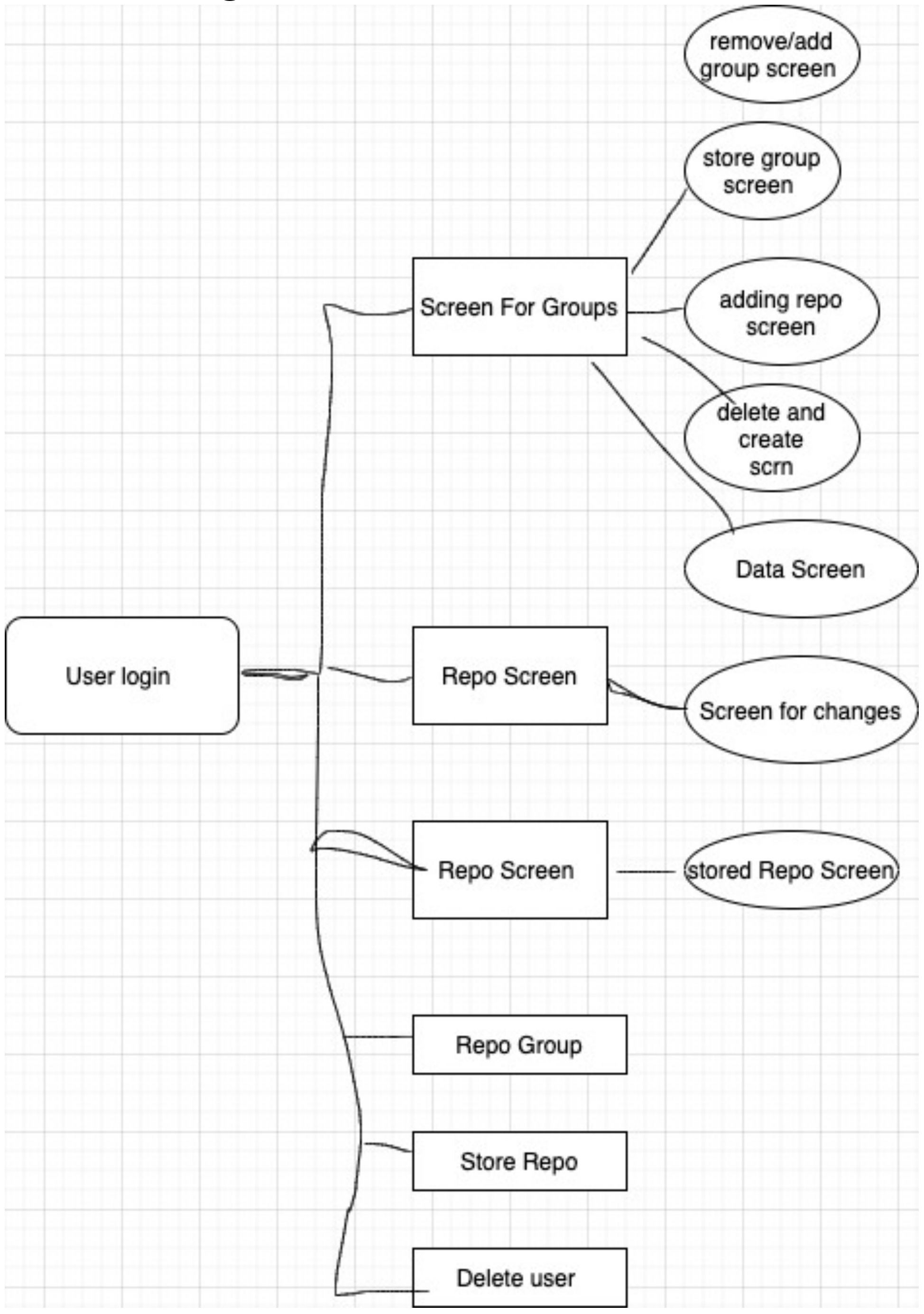
- Data must be brought together from Github, issue trackers, mailing lists, library dependency trees, the Linux Foundation's badging program, code complexity and contribution counting, and more
- Augur must be available on all major web browsers.
- System Codes must be open source and public on github
- Every function that is supported on the desktop version should also be supported on the desktop version.
- Differences between mobile and desktop websites should be minimal, but will be changed slightly for design and layout purposes.

6. Purchased Components

Augur is an open source software hence its free but other components and their price are as follows

1. Aws amazon server $100 monthly
2. External Data Storage (optional) $50

7. Design Interfaces


remove/add group screen
store group screen
Screen For Groups
adding repo screen
delete and create scrn
Data Screen
User login
Repo Screen
Screen for changes
Repo Screen
stored Repo Screen
Repo Group
Store Repo
Delete user