COMP SCI 4ZP6:

HubListener

Requirements Specifications Document

Authored By:
Ahmad, Zed
Jalan, Prakhar
Oliveira, Pedro
Selvathayabaran, Piranaven

Contents

1 Introduction

The following official document is the requirements specification document based on the Volere Template. It contains information regarding project drivers, constraints, functional/non-functional reqiurements and any other critical details that are defined under the templates defintion.The specifications document is subject to change and any modifications will be noted in the Revision History Section.

2 Project Drivers

2.1 The Purpose of the Project

2.2 The Client the Customer, and Other Stakeholders

2.3 Users of the Product

3 Project Constraints

3.1 Mandated Constraints

This section describes the constrains on the design of the HubListener service. They are the same as other requirements except they are mandatory:

Solutions Constraints:
Describe: The HubListener system shall have a command line interface that has access to the GithubAPI and associated analysis tools.
Rationale: This is central souce of the infromation needed.
Fit Criterion: The interface will conform to Github API and javascript standards

Implementaton Environment of the Current System
Describe: The HubListener system shall be written in the NodeJS environment.
Rationale: This is native way of accessing the githubAPI while also creating a web based tool.
Fit Criterion: The interface will conform to NodeJs and javascript policies and requirements.

Off-the-Shelf Software
Describe: The HubListener system shall utilize ay npm packages that aid in the retrieval and analysis of the repository.
Rationale: This is simplest way of retrieving and analyzing the repository. This will save labour time.
Fit Criterion: The interface will conform to NodeJs and javascript policies and requirements.

Schedule Constraints:
Description: The service shall be available January 31st 2018 as a version 1 release.
Rationale: We want to launch the service by this date so that we can do testing and iterate over the solution so that there is a polished solution for our April presentation.
Fit Criterion: The HubListener serice will be available for testing by January 31st 2018

Budget Constraints: The project has no financial budget.

3.2 Naming Conventions and Definitions

A glossary containing the meanings of all names, acronyms, and abbreviations sed within the requirements specification. The following is a running, ongoing

dictionary.

Naming Conventions List	
Naming Convention	Defintion
Github	The online, open-source repository that is available here.
NodeJS	tbd
HubListener	The command-line service that is to be created.
Repository	tbd
Command-Line	tbd
Cyclical Complexity	tbd

3.3 Relevant Facts and Assumptions

4 Functional Requirements

4.1 The Scope of the Work

The Current Situation: Currently, there is no way to compare github repositories against similar repositories within the ecosystem. There is no way to anaylze your code against similar project or see how your repository is trending in comparison to similar projects within the ecosystem.

Context Diagram

4.2 The Scope of the Service

HubListener aims to provide a service which allows the user to compare his/her repository or anyy open-source repository against similar repositories in the ecosystem. The end user will be able to attain meaningful information that they can use to improve their current repository and guage how they are trending.

Naming Conventions List	
Naming Convention	Defintion
Github	The online, open-source repository that is available here.

Input: Github Repo Clone Link

Output: - Cyclomatic Complexity - Essential Complexity - Integration Complexity - Cyclomatic Density - Lines of Code - Lines of Commens - Maintainability Index - Coupling - Nmber of Methods - Numbe rof Variables - Number of Issues - Number of Bugs - Number of Stars - Number of possible Logical errors - Functional Coverage Score - Condition Coverage Score

-

4.3 Functional and Data Requirements

5 Nonfunctional Reqiurements

5.1 Look and Feel Reqiurements

1. The application should provide an easy and clear command line interace for user to use the service. 2.The application shall comply with Open-source standards 3.Useful information(such as help, report issues, training) should be easily accessible. 4. When doing calculations or data handling like repository retrievals, the application should display an animated progress bar.

More to be determined at a later date.

5.2 Usability and Humanity Reqiurements

1. The software must be simple for a person aged above 18 years, and knowledge of Github/open-source technology, in able condition to understand and use all its features. 2. The application shall make it easy for the average user to find user-use guidelines. 3. The system must meet all open-source software accessibility standards enforced by the gouvernment of Canada.

5.3 Performance Requirements

1. After the user provides their repo link, the applicatoin shall generate charts and statisitcs ina timely manner. 10 seconds. 2. The applicaiton shall save the users last request and results. 3. The application shall analyze the results and provide recommendations and trends.

5.4 Operational Requirements

Expected physical environemnt -Users will use the application on their internet-connected computer Expected Technological Environemnt - The application should work on devices that have Node.JS installed on their machine

5.5 Maintainability and Support Reuquirements

Maintainability The sotware application is to be easily modifiable The application should notify user's to check for an npm update every 6 months. The application shall be ready to be deployed on any OS.

Portability The application shall be availble on any computer operating system such as MAC, Windows and Linux

5.6 Security Requirements

5.7 Cultural and Political Requirements

1. The application should not display any offensive text or informaoitn 2. The application should be available in English .

5.8 Legal Requirements

1. The application shall comply with all relevant information privacy acts. 2. The application shall comply wiht all relevant open-source lawas. 3. The application will abide by all developr guidelines as denoted by Windows, Apple, etc.

6 Project Issues

6.1 Open Issues

All open issues can be see on our Issue tracking board. (inserrt link)

6.2 Off-the-Shelf Solutions

Attempting to emulate similar applicaiton could greatly reduce th etime needed to design and implemnt the app. If an off the shelf solution is available and fit our reqiurements, we will use it as part of the software, providing the necessary credit as needd.

6.3 New Probelms

6.4 Tasks

All open taks can be seen our Issue tracking board (see link here)

6.5 Migration to the New Product

No new product to migrate too. This section may become obsolete.

6.6 Risks

6.7 Costs

There are no costs associated with the development of this project other than the time dedicated to development/documentation. It is developed under the open-source environemnet and therefore is useable for not-for profit purposes. In future, there may be a cost associated with maintaining the project, hosting the project or use by professional company.

6.8 User Documentatoin and Training

7 Revision History

Table 1: Revision History

Date	**Developer(s)**	**Change**
November 1st, 2018	Piranaven Selva	Make Foundation for Specifications Document as per Issue #6