**Software Requirement Specification**

**For**

**Source Code Management Github**

**Version 1.0 approved**

**Prepared by :: 1. Parth Panchal(CE 076)**

**2.Dhruv Patel (CE 081)**

**3. Akshay Patel(CE 080)**

**DDU,Nadiad**

**3 January 2017**

**Table of Contents**

**Table of Contents** [**ii**](#__RefHeading___Toc441230970)

**Revision History** [**ii**](#__RefHeading___Toc441230971)

**1. Introduction** [**1**](#__RefHeading___Toc441230972)

1.1 Purpose [1](#__RefHeading___Toc441230973)

1.2 Document Conventions [1](#__RefHeading___Toc441230974)

1.3 Intended Audience and Reading Suggestions [1](#__RefHeading___Toc441230975)

1.4 Product Scope [1](#__RefHeading___Toc441230976)

1.5 References [1](#__RefHeading___Toc441230977)

**2. Overall Description** [**2**](#__RefHeading___Toc441230978)

2.1 Product Perspective [2](#__RefHeading___Toc441230979)

2.2 Product Functions [2](#__RefHeading___Toc441230980)

2.3 User Classes and Characteristics [2](#__RefHeading___Toc441230981)

2.4 Operating Environment [2](#__RefHeading___Toc441230982)

2.5 Design and Implementation Constraints [2](#__RefHeading___Toc441230983)

2.6 User Documentation [2](#__RefHeading___Toc441230984)

2.7 Assumptions and Dependencies [3](#__RefHeading___Toc441230985)

**3. External Interface Requirements** [**3**](#__RefHeading___Toc441230986)

3.1 User Interfaces [3](#__RefHeading___Toc441230987)

3.2 Hardware Interfaces [3](#__RefHeading___Toc441230988)

3.3 Software Interfaces [3](#__RefHeading___Toc441230989)

3.4 Communications Interfaces [3](#__RefHeading___Toc441230990)

**4. System Features** [**4**](#__RefHeading___Toc441230991)

4.1 System Feature 1 [4](#__RefHeading___Toc441230992)

4.2 System Feature 2 (and so on) [4](#__RefHeading___Toc441230993)

**5. Other Nonfunctional Requirements** [**4**](#__RefHeading___Toc441230994)

5.1 Performance Requirements [4](#__RefHeading___Toc441230995)

5.2 Safety Requirements [5](#__RefHeading___Toc441230996)

5.3 Security Requirements [5](#__RefHeading___Toc441230997)

5.4 Software Quality Attributes [5](#__RefHeading___Toc441230998)

5.5 Business Rules [5](#__RefHeading___Toc441230999)

**6. Other Requirements** [**5**](#__RefHeading___Toc441231000)

**Appendix A: Glossary** [**5**](#__RefHeading___Toc441231001)

**Appendix B: Analysis Models** [**5**](#__RefHeading___Toc441231002)

**Appendix C: To Be Determined List** [**6**](#__RefHeading___Toc441231003)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

1. **Introduction**

1.1 **Purpose**

The purpose of this document to provide detailed description of Source code management system. This document will explain the purpose and features of the system, how the system performs, Its goals and user interface. This document is meant for Developers, Project Managers, Documentation Writers and marketing staff also.

1.2 **Document Conventions**

*There are two types of typographical conventions used in this document. First with bold letters and second with highlighted text. The one with bolded letters signifies the core idea and one with normal fonts describes the respective point with necessary details. The one with Bolded and Larger fonts describes the name of topic to be discussed.*

1.3 **Intended Audience and Reading Suggestions**

*This document is for Developers, Project Managers, user . This document has all the information related to the product and it is sequenced as per the life cycle of the product. All readers should start reading from the beginning of this document where developers should focus more on functional requirements, marketing staff on key features and documentation writers on all the information regarding the product.*

1.4 **Product Scope**

SCM Github is software which provide the functionality for project management and project version management .The main reason for developing such a software is that we can easily get the solution for any project which is uploaded by any user .Also we can get the any programming code which is uploaded by some user. This Software will be available for any public who has the account on github.

1.5 **References**

·0 *Software Engineering by Rogers Pressman.*

·1 *IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.*

2. **Overall Description**

2.1 **Product Perspective**

This software is inspired by the complex and clumsy project and version management Github allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of those lines of development takes seconds. Unlike traditional version controlling system such as bitkeeper and Bazaar, our scm github  is a fast, scalable, distributed revision control system with an unusually rich function set that provides both high-level operations and full access to internals.

2.2 Product Function

* **Frictionless Context Switching**. Create a branch to try out an idea, commit a few times, switch back to where you branched from, apply a patch, switch back to where you are experimenting, and merge it in.
* **Role-Based Codelines**. Have a branch that always contains only what goes to production, another that you merge work into for testing, and several smaller ones for day to day work.
* **Feature Based Workflow**. Create new branches for each new feature you're working on so you can seamlessly switch back and forth between them, then delete each branch when that feature gets merged into your main line.
* **Version control.** Allow to create new version of same project and allow to download various project available on github based on permission.

2.3 User classes and characteristic

The user classes that will anticipate in in the software are:

* User:

User can upload their project. And also they can download the project with higher version of that project. User can also upload their programming code and download the code which has been uploaded by some other user.

* Admin:

Admin can manage user. Also admin handle rating of user.

The admin will be able to interact with the software in order to know the user rating and to update the users list and remove the users if necessary.

3. **External Interface Requirement**

3.1 **Hardware Interfaces**

Ram : At least 500 MB

Free space: 5 GB

Processor: Higher or equal version to Pentium

3.2 **Software Interfaces**

* Operating Systems: Windows is recommended for better interface.
* Database: To save User details, Project details, Version details.
* Implementation Language: Python and Django framework.
* **Communications Interfaces**

The software would use HTTP for the tasks related to internet and would use FTP/IP for the communication through intranet.

4. **System Features**

4.1 **Users Can log in.**

Input: username, password

Output: Login\_Session

Description: This function will take username and password as a input and verify it with certain criteria and database and if username and password is valid login\_session will be created.

4.2 **System allow customer to create new repository or project.**

Input: Project or repository name

Output: req\_flag

Descrition: This Function allow users to create new repository or project and take project name as input and set req\_success flag if repository is created.

4.3 **System allow customer to clone or download projects.**

Description: User Can download or clone any project to his/her repository if project is public and permission is given.

4.4  **System allow client to track files from project.**

Input: File name and file path

Output: req\_success

Description: system will take file name and file path and set it in tracking file list so if any changes done to that files user can see changes.

4.5 **System allow client to search and reuse code.**

Input: search keyword

Output: Project List

Description: System will take keyword from user and gives list of all available projects based on SEO (search engine Optimization) very fast.

4.6 **System allow client to rate projects.**

Input: project Rate

Output: Updated Rate

Description: System allow client to rate other users project based on their point of view.

4.7 **System allow client to work on same project in group.**

Description: It allows that many user can work on same project in group and one user will be master who can see all the changes made by others in specific files.

5. **Other Non-functional Requirements**

5.1 **Performance Requirements**

The system would normally take 2 seconds to perform regular tasks. However in order to process large databases it might take more time.

5.2 **Safety Requirements**

The safety of data is taken care of as the database is updated in server with every new database related task.