Software Requirements Specification

for

CollabPro – Collaborative Editor

Version 1.0

Prepared by
Atharva Marathe (111903024)
Avishkar Andhale (111903025)

College of Engineering, Pune
INSTRUCTOR: Prof. Tanuja Pattanshetti

24 February, 2022

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for "CollabPro – Collaborative Code Editor". It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications.

1.2 Document Conventions

To build this template we followed IEEE standard Software Requirements Specification.

1.3 Intended Audience and Reading Suggestions

This document is made to guide developers and customers so that they can suggest changes in further scope of improvement. Also, this document specifies use case diagram, product scope, product functions, functional and non-functional requirements and technical issues related with the product.

The software requirement specification (SRS) document is written for a more general audience. This document is intended for individuals directly involved in the development of Collaborative Code Editor. It also includes software developers, project consultants, and team managers.

1.4 Product Scope

Building software projects by students, companies working remotely is common these days. Hence collaborative tools for coding are essential while working on group or team-based projects. Lot of Collaborative editing tools are available on the internet, but most of them are Rich Text based editors. CollabPro provides collaborative experience for coders with syntax highlighting, auto-suggestions features. This product makes it easy and convenient for teams working on a common project.

1.5 References

1] **Operational Transformation**: https://srijancse.medium.com/how-real-time-collaborative-editing-work-operational-transformation-ac4902d75682/

2] **Socket Programming:** https://socket.io/

3] **Code-Mirror** https://codemirror.net/

2. Overall Description

2.1 Product Perspective

CollabPro is a code editor with real-time collaborative features, unlike most of the editors supporting rich text, this software provides coding-specific tools and experience. It belongs to class of online editors with some added functionalities. OAuth based authentication provides quick and secure login to the application. Thus, it gives all the required features in a single application.

2.2 Product Functions

The main function of our product is to provide a platform to users, to code and complete the tasks online, without the need to install apps on the computer. It enables users to easily collaborate with team members in real-time. Users can communicate with each other through chat section.

2.3 User Classes and Characteristics

There are mainly two users for this software:

1. Host: The host creates coding sessions, invites other participants / team-members. The host has the rights to use all the features of the application.

2. Participant: The participant joins the session created by the host. The participant can use features of the application depending upon the session privileges granted by the host.

2.4 Operating Environment

- Web Browser
- Internet Connection
- Any operating system
- Compatible processor
- RAM
- Sufficient Hard-disk space

2.5 Design and Implementation Constraints

In order to use the application smoothly, a stable internet connection is required.
As the number of users increase, the traffic congestion may become severe, leading to decrease in performance. As the files are temporarily stored on the server, the server needs to have sufficient hard-disk space.

2.6 User Documentation

With the application separate user manual will be provided that will help the user to get familiar with the platform.

3. External Interface Requirements

3.1 User Interfaces

Various user interfaces for the product could be:

- Login Page
- Home Page
- Host Dashboard
- Code Editor

3.2 Hardware Interfaces

- Processor
- RAM
- Hard Drive
- Internet Connection

3.3 Software Interfaces

React: Front-end JS library to be used for frontend work
MongoDB: To be used to store databases
Express: Back-end JS library to design web application
Node.js: Back-end JavaScript runtime environment.
Web Browser

3.4 Communications Interfaces

HTTP connections will be setup between client and server of the application. Socket IO library will be used to implement real-time collaboration.

4. System Features

4.1 Login Authentication and Authorization

4.1.1 Description and priority:
Host/participant should login using their OAuth based authentication.
The feature has high priority.
4.1.2 Stimulus / Response Sequences:
Entering correct credentials. In response for the above stimulus user will be logged into the system.
Entering incorrect credentials.
In response for the above, it will ask to enter the credentials again.

4.2 Real-time Collaboration

4.2.1 Description and priority:
Real-time Collaborative environment in the code editor. Users can edit the code simultaneously. The feature has high priority.
4.2.2 Stimulus / Response Sequences:
User changes some part of the code
The code gets updated in each of the user present in the session.

4.3 Chat-Section

4.3.1 Description and priority:
Chat-section, where users can communicate with each other during the session. The feature has medium priority.
4.3.2 Stimulus / Response Sequences:
User sends a message in chat box.
All the joined users will receive the message in chat-section.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Real-time code editors require minimum latency, as many users may edit the contents simultaneously. As the number of users increase, the server must be able to handle the traffic congestion efficiently.

5.2 Safety Requirements

User-data should be stored temporarily in the server, in case if the user gets disconnected, and should be able to resume from where it left.

5.3 Security Requirements

Only authentic and authorized must be admitted to the session. The user data must be kept secure during the session. All the personal information of the user, passwords must be encrypted to prevent any kind of piracy.

Appendix A: Glossary

OAuth: Open Authorization, open standard authorization framework for token-based authorization.
Socket IO: JavaScript library for real-time web applications.
Rich Text: File format that lets you exchange text files between different word processors
HTTP: Hypertext Transfer Protocol, application level protocol in the internet.