

---

# **Software Requirements Specification**

## **Collaborative Coding and messaging Platform**

**Prepared by -**

**112103065 - Shriya Kale**

**112103067 - Himanshu Kamdi**

**112103076 - Soham Kumthekar**

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>ii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Product Scope	1
1.3 References	1
<b>2. Overall Description</b>	<b>2</b>
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Assumptions and Dependencies	3
<b>3. External Interface Requirements</b>	<b>3</b>
3.1 User Interfaces	3
3.2 Software Interfaces	3
3.3 Communications Interfaces	3
<b>4. System Features</b>	<b>4</b>
4.1 System Feature 1	4
4.2 System Feature 2	
4.3 System Feature 3	
<b>5. Other Nonfunctional Requirements</b>	<b>4</b>
5.1 Performance Requirements	4
5.2 Safety Requirements	
<b>Appendix A: Glossary</b>	<b>5</b>
<b>Appendix B: Analysis Models</b>	<b>5</b>
<b>Appendix C: To Be Determined List</b>	<b>6</b>

# 1. Introduction

## 1.1 Purpose

*The purpose of this collaborative development platform is to provide developers with a unified environment for project management, GitHub integration, and real-time code collaboration.*

## 1.2 Product Scope

*The platform will allow users to create project-specific groups, seamlessly integrate with GitHub repositories, and collaboratively edit code files in real-time.*

## 1.3 References

About Github API - [Github API Documentation](#)

About Firebase - [Firebase Documentation](#)

# 2. Overall Description

## 2.1 Product Perspective

*The platform will serve as an integrated solution for developers, connecting seamlessly with GitHub repositories and offering a live collaborative code editing environment. This platform is not a follow-on member of an existing product family but rather represents an innovative solution to address the evolving needs of modern software development practices.*

### **Origin:**

*The concept for this collaborative development platform originated from the recognition of the challenges faced by development teams in coordinating efforts, ensuring code quality, and streamlining project management which we realized from our personal experiences encountering collaboration and coordination challenges during college projects. Existing systems often lacked integration and real-time collaboration features. The need for a unified environment that seamlessly integrates with popular version control systems like GitHub while providing robust project management and real-time collaboration tools became evident.*

### **Product Placement:**

*This platform serves as a standalone product designed to meet the diverse needs of software development teams across industries. It is not a replacement for existing systems but rather an enhancement, offering a comprehensive suite of features to improve efficiency, communication, and code quality in a single, unified environment.*

## 2.2 Product Functions

### **Messaging Feature:**

*Communicate with other members privately or in groups*  
*Create Project Groups*  
*Share documents, images, or code files*  
*Add/Remove Members to Project Groups*

### **GitHub Integration:**

*Connect GitHub Repositories to Project Groups*  
*View Repository Information within the Platform*  
*Version Control for Edited Files*  
*Edit files and commit the changes*

### **Live Collaborative Editor:**

*Real-time Code Editing*

## 2.3 User Classes and Characteristics

*Administrator: Manages platform settings and user roles.*  
*Project Manager: Creates and manages project groups.*  
*Developer: Collaborates on code within project groups.*

## 2.4 Operating Environment

Operating System: Windows / Linux based  
Node: 16+  
Firebase: 7+

## 2.5 Assumptions and Dependencies

*The User must be familiar with the basic functionalities of Github.*  
*The project is dependent on the Firebase server for user authentication and real-time database.*

# 3. External Interface Requirements

## 3.1 User Interfaces

- *Login and register pages*
- *Messaging interface*
- *Files Interface*
- *Code Editor*

## 3.2 Software Interfaces

- *Firebase*
- *Github*

## 3.3 Communications Interfaces

- *HTTPS*
- *Websockets*

# 4. System Features

## 4.1 Unified Environment:

- **Objective:** To provide developers with a single, integrated platform for managing projects, code repositories, and collaboration.
- **Features:**  
Centralized project management tools.  
A consistent interface for accessing code repositories.

## 4.2 GitHub Integration:

- **Objective:** Seamless integration with GitHub to enhance version control, code review, and collaboration capabilities.
- **Features:**  
Direct integration with GitHub repositories.  
Real-time synchronization of code changes.  
Edit files and commit the changes to Github

## 4.3 Real-Time Code Collaboration:

- **Objective:** Facilitate effective communication and collaboration among developers,
- **Features:**  
Real-time code editing with other developers.  
Visibility into who is editing and connected users.

## **5. Other Nonfunctional Requirements**

### **5.1 Performance requirements**

- The system should respond to user actions, such as saving changes or opening files, within 1 second under normal load conditions.
- Code synchronization between collaborators should occur in real-time, with changes propagated to all connected users within milliseconds.
- The platform should support a minimum of 100 simultaneous users without experiencing significant degradation in performance.

### **5.2 Safety requirements**

- User authentication must be implemented using industry-standard encryption protocols to ensure the confidentiality of login credentials.
- Access to code repositories and collaboration features should be restricted based on user roles and permissions, with support for fine-grained access control.
- Communication between the client and server should be encrypted using SSL/TLS to prevent eavesdropping and data tampering.

### **5.3 Usability requirements**

- The user interface should be intuitive and responsive, providing clear feedback on user actions and system status.
- Collaborative editing features should be seamlessly integrated into the code editor, with intuitive controls for initiating, accepting, and rejecting changes from other users.
- Help documentation and tutorials should be easily accessible within the platform to assist users in getting started and mastering advanced features.
- 

### **5.4 Scalability requirements:**

- The architecture should support horizontal scaling to accommodate increases in user activity and data volume without degradation in performance.
- Load balancing mechanisms should be in place to distribute incoming traffic evenly across multiple servers and prevent bottlenecks.
- The platform should be capable of handling a 50% increase in concurrent users and code collaboration activity over the next two years.

## **5.5 Compatibility requirements:**

- The application should be compatible with the latest versions of major web browsers, including Chrome, Firefox, Safari, and Edge.
- Integration with popular version control systems and development tools, such as Git, GitHub, and Visual Studio Code, should be supported out of the box.

## **Appendix A: Glossary**

## **Appendix B: Analysis Models**

## **Appendix C: To Be Determined List**