Software Requirements Specification

For

Icollab

(A Collaboration Platform for Teams and Organizations)

Prepared by

Meet Antala (22CEUOS096)

Yash Gabani (22CEUOS137)

January 2025

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for Icollab, a platform designed to enhance collaboration, streamline workflows, and improve productivity for teams and organizations. Icollab aims to provide robust communication tools, real-time updates, and integration capabilities.

1.2 Document Conventions

- Entire Document Should be Justified
- Convention for main title
  Font : Arial
  Font style : medium
  Font size : 18
- Convention for sub title
  Font : Arial
  Font style : medium
  Font size : 12
- Convention for body
  Font : Arial
  Font style : small
  Font size : 12

1.3 Intended Audience and Reading Suggestions

- Admin
- User

1.4 Product Scope

Icollab will enable teams to communicate effectively via messaging, video calls, and file sharing. It will include features such as:

- Creation of workspaces and channels for specific projects.
- Integration with external tools (e.g., GitHub, Google Drive).
- Notifications and message summarization using machine learning.

1.5 References

1. MongoDB Documentation
2. ReactJS Official Guide
3. Node.js Documentation
4. WebRTC Documentation

2. Overall Description

2.1 Product Perspective

Icollab is a web-based platform built using the MERN stack (MongoDB, Express.js, ReactJS, Node.js). It is designed to integrate seamlessly with third-party APIs for enhanced functionality, such as authentication and storage.

2.2 Product Functions

- Workspace creation and channel management.
- Real-time messaging and notifications.
- File sharing with storage integration (e.g., Firebase, Google Drive).
- Message summarization using AI.
- Video calls with multiple participants.

2.3 User Classes and Characteristics

- **Admin**: Manage users, workspaces, and settings.
- **Team Members**: Create/join workspaces, participate in channels, and collaborate.

2.4 Operating Environment

- **Platform**: Web application
- **Browser Compatibility**: Chrome, Firefox, Edge, Safari

2.5 Design and Implementation Constraints

- Support for both desktop and mobile devices.

2.6 User Documentation

User manuals and help guides will be provided to assist users in navigating and utilizing the platform effectively.

2.7 Assumptions and Dependencies

- Internet availability is required for full functionality.
- Users should have basic knowledge of collaboration tools.

3. External Interface Requirements

3.1 User Interfaces

- **Login Page**: Clean and intuitive with multi-factor authentication.
- **Dashboard**: Displays workspaces, notifications, and recent activity.
- **Chat Interface**: With real-time updates.

3.2 Hardware Interfaces

No specific hardware requirements beyond internet-enabled devices.

3.3 Software Interfaces

- MongoDB for database management.
- WebRTC for video calls.

3.4 Communications Interfaces

- HTTPS for secure communication.
- WebSocket for real-time chat updates.

4. System Features

1. Workspace and Channel Creation
2. Real-time Messaging
3. Video Calling and Conferencing
4. File Sharing
5. Notification System
6. Message summarization
7. Integration with Github, Gmail etc.

5. Functional Requirements

5.1 User Authentication

Description: Enables users to securely log in, register, and recover passwords.

- **5.1.1 Login**
    - Input: Email and Password
    - Output: Access to the dashboard upon successful login.
- **5.1.2 Registration**
    - Input: Name, Email, Password
    - Output: Account created successfully.

5.2 Workspace Management

Description: Users can create or join workspaces for their teams.

- **5.2.1 Create Workspace**
    - Input: Workspace name, description.
    - Output: Workspace successfully created and listed on the dashboard.

5.3 Channel Management

Description: Workspaces consist of channels where members collaborate.

- **5.3.1 Create Channel**
    - Input: Channel name, type (public/private).
    - Output: Channel created successfully.

5.4 Messaging System

Description: Real-time chat with options for file sharing and reactions.

- **5.4.1 Send Message**
    - Input: Message text or file upload.
    - Output: Message displayed in the chat.

5.5 Video Calling

Description: Supports one-on-one and group video calls using WebRTC.

- **5.5.1 Initiate Call**
    - Input: Select participants.
    - Output: Video call session initiated.

5.6 Integration with Third-Party Apps

Description: Provides integrations with tools like GitHub, Gmail

- **5.6.1 GitHub Integration**
    - Input: Repository link, API key.
    - Output: Commits and issues visible in the workspace.
- **5.6.2 Gmail Integration**
    - Input: Google Account authentication.
    - Output: Workspace events synced with Gmail.

5.7 Message Summarization

Description:Message summarization provides a concise overview of lengthy conversations in a channel or workspace.

- **5.7.1 Summarize Channel Conversations**
    - Input: Chat of particular Channel
    - Output:A brief summary of the conversations.

6. Other Non-Functional Requirements

6.1 Performance Requirements

- The platform must deliver fast and interactive responses.
- Opening channels, sending messages, and retrieving chat histories should not exceed 1 second of delay.
- Database operations such as fetching user data, channel information, and file uploads must be fast.
- The summarization of messages for large conversations (should be processed fast.

6.2 Safety Requirements

- Only authorized administrators can manage and query the database contents.
- All users must be authenticated before accessing restricted features, such as private channels, workspaces, or messaging.
- Data protection is ensured during message transmission and storage.
- Any failed or interrupted payment transactions must have a refund or reversal system to ensure financial accountability.

6.3 Security Requirements

- Communication must be encrypted using HTTPS to secure data transmission.

6.4 Software Quality Attributes

- Browser Compatibility: Supports major browsers, including Chrome, Firefox, Safari, and Edge.
- Mobile Responsiveness: Fully responsive design ensures seamless access across Android and iOS devices.

6.5 Business Rules

- Trademark and Copyright: Ensures all intellectual property, such as the platform's logo and codebase, is trademarked and protected under copyright laws.
- File Sharing Restrictions: Limits the types of files shared in chats to prevent malware or illegal content.

7. Goals of Implementation

1. **Ensure Mobile Responsiveness**: Provide a seamless experience across devices, including smartphones, tablets, and desktops.
2. **Promote Marketing and Campaigns**: Enable announcements and promotional campaigns within channels for organizations.
3. **Workspace and Channel Customization**: Allow users to customize their workspace and channels with preferences, including themes and notification settings.
4. **Chatbot Functionality**: Introduce a chatbot to assist users with platform navigation, FAQs, and task automation.
5. **Optimized Performance**: Focus on delivering fast response times for message sending, file sharing, and workspace switching.

8. References

1. MongoDB Documentation
2. React Documentation
3. Node.js Documentation
4. WebRTC Documentation
5. OpenAI API Documentation (for message summarization)