TABLE OF ONTENTS

[Software Requirements Specification (SRS) 2](#_Toc3708)

[Collaborative Task Management Web Application 2](#_Toc24908)

[1. Introduction 2](#_Toc19647)

[1.1 Purpose 2](#_Toc26989)

[1.2 Scope 2](#_Toc5695)

[1.3 Definitions and Acronyms 2](#_Toc26474)

[2. System Overview 3](#_Toc3188)

[2.1 System Architecture 3](#_Toc13664)

[2.2 User Classes and Characteristics 3](#_Toc3439)

[3. Functional Requirements 3](#_Toc31739)

[3.1 User Authentication and Profile Management 3](#_Toc16026)

[3.2 Task Management 4](#_Toc13479)

[3.3 Team Collaboration 5](#_Toc7861)

[3.4 Notifications 5](#_Toc10777)

[3.5 Reports and Analytics 6](#_Toc24755)

[4. Non-Functional Requirements 6](#_Toc21063)

[4.1 Performance Requirements 6](#_Toc7762)

[4.2 Security Requirements 6](#_Toc29968)

[4.3 Usability Requirements 6](#_Toc10408)

[4.4 Reliability Requirements 7](#_Toc27931)

[4.5 Scalability Requirements 7](#_Toc11779)

[5. External Interfaces 7](#_Toc31927)

[5.1 User Interfaces 7](#_Toc26565)

[5.2 External Service Interfaces 7](#_Toc16073)

[6. CI/CD and Deployment Requirements 7](#_Toc22567)

[6.1 CI/CD Pipeline 7](#_Toc15361)

[6.2 Deployment Architecture 8](#_Toc7941)

[6.3 Monitoring and Logging 8](#_Toc12020)

[7. Data Requirements 8](#_Toc26147)

[7.1 Data Entities 8](#_Toc5583)

[7.2 Data Storage 8](#_Toc10384)

[8. System Constraints 8](#_Toc26291)

[9. Acceptance Criteria 9](#_Toc30214)

[10. Appendices 9](#_Toc9224)

[10.1 User Stories 9](#_Toc6786)

# **Software Requirements Specification (SRS)**

# **Collaborative Task Management Web Application**

## **1. Introduction**

### **1.1 Purpose**

This Software Requirements Specification (SRS) document describes the functional and non-functional requirements for a Collaborative Task Management Web Application. The system will allow users to manage tasks collaboratively within teams, providing features for task creation, assignment, tracking, communication, and notifications.

### **1.2 Scope**

The Collaborative Task Management Web Application will provide a comprehensive platform for team collaboration through task management. The system will include user authentication, task management, team collaboration, notifications, and reporting features. The application will be web-based and accessible across various devices.

### **1.3 Definitions and Acronyms**

JWT: JSON Web Token ; a compact URL safe means of representing claims to be transferred between two parties

CI/CD: Continuous Integration/Continuous Deployment

API: Application Programming Interface

UI: User Interface

UX: User Experience

## **2. System Overview**

### **2.1 System Architecture**

The application will follow a client-server architecture with:

* Frontend: React.js for state management
* Backend: Django
* Database: Postgresql database for flexible document storage
* Authentication: JWT and Google Oauth 2.0
* Real-time Communication: Socket.io for chat functionality
* Deployment: Docker containers with CI/CD pipeline on GitHub Actions(We tried using vasel for the deployment)

## **2.2 User Classes and Characteristics**

* Regular Users: Can create tasks, join teams, comment, and receive notifications.
* Team Administrators: Can manage team members, assign tasks, and oversee team activities.
* System Administrators: Have full access to system configuration and user management.

## **3. Functional Requirements**

### **3.1 User Authentication and Profile Management**

1. **User Registration**

* Users shall be able to register using email/password
* Users shall be able to register using Google authentication
* System shall validate all registration information
* System shall enforce password complexity requirements

1. **User Login**

* Users shall be able to log in using registered credentials
* Users shall be able to log in using Google authentication
* System shall provide password recovery functionality

1. **User Profile**

* Users shall be able to view and edit their profile information
* Users shall be able to upload a profile picture
* Users shall be able to set notification preferences
* Users shall be able to view their task history

### 3.2 Task Management

1. **Task Creation**

* Users shall be able to create tasks with title, description, and due date
* Users shall be able to set task priority (Low, Medium, High, Urgent)
* Users shall be able to attach files to tasks
* Users shall be able to add tags/labels to tasks

1. **Task Assignment**

* Users shall be able to assign tasks to team members
* Users shall be able to assign tasks to multiple users
* System shall notify users when tasks are assigned to them
* Task creators shall be able to change task assignments

**c)Task Tracking**

* Users shall be able to change task status (To Do, In Progress, Review, Completed)
* Users shall be able to set task progress percentage
* System shall track time spent on tasks
* System shall provide task history and audit trail

**d)Task Filtering and Search**

* Users shall be able to filter tasks by status, priority, assignee, and due date
* Users shall be able to search tasks by title and description
* Users shall be able to sort tasks by various attributes
* System shall provide saved filter/search functionality

### 3.3 Team Collaboration

1. **Team Creation and Management**

* Users shall be able to create teams
* Team administrators shall be able to invite members via email
* Team administrators shall be able to assign roles to team members
* Team administrators shall be able to remove members from teams
* Team Workspace
* Teams shall have dedicated workspaces
* Team members shall be able to view all team tasks
* Team members shall be able to view team activity

**b)Task Comments and Chat**

* Users shall be able to comment on specific tasks
* Comments shall support text formatting and reactions
* Users shall be able to mention other team members in comments
* System shall provide chat functionality within task threads

### **3.4 Notifications**

1. **Task Notifications**

* System shall notify users about task assignments
* System shall send reminders for approaching deadlines
* System shall notify task creators when assigned tasks are completed
* System shall notify users when they are mentioned in comments

1. **Team Notifications**

* System shall notify users about team invitations
* System shall notify team members about new tasks created within teams
* System shall provide digest notifications for team activities
* System shall allow customization of notification preferences

### 3.5 Reports and Analytics

1. **User Reports**

* System shall generate reports on individual task completion
* System shall track time spent on different tasks by users
* System shall provide personal productivity metrics
* Users shall be able to export their reports

**b)Team Reports**

* System shall generate team performance analysis
* System shall track task distribution amongst the members
* System shall provide deadline compliance data
* Team administrators shall be able to export team reports

## **4. Non-Functional Requirements**

### **4.1 Performance Requirements**

* System shall support up to 1000 concurrent users
* Page load time shall not exceed 2 seconds under normal conditions
* Database queries shall be complete within the required time
* Chat messages shall be delivered on time

### **4.2 Security Requirements**

* All passwords shall be hashed
* All communications shall be encrypted using HTTPS
* JWT tokens shall expire after 24 hours
* System shall implement rate limiting to prevent brute force attacks
* System shall log all authentication attempts

### **4.3 Usability Requirements**

* UI shall be responsive and compatible with all major browsers
* System shall be accessible according to the standards
* Interface will be understandable with minimal training required
* System shall provide help and tips

### **4.4 Reliability Requirements**

* System shall be available 99.9% of the time
* System shall implement automated database backups daily
* System shall recover from crashes easily
* No single point of failure shall exist within the system architecture

### **4.5 Scalability Requirements**

* System shall be able to handle increased load
* System shall implement strategies for improved performance
* Database shall be designed to handle growing data easily
* System shall support future expansion

## **5. External Interfaces**

### **5.1 User Interfaces**

1. Clean, modern UI with responsive design
2. Dark/light mode support
3. Customization of dashboard with widgets
4. list views for task management

### **5.2 External Service Interfaces**

* Google Oauth 2.0 for authentication
* Email service integration
* File storage service
* Calendar integration (Google Calendar)

## **6. CI/CD and Deployment Requirements**

### **6.1 CI/CD Pipeline**

* Automated testing on code commits
* Code quality checks
* Continuous integration with GitHub Actions
* Automated deployment to the environment
* Manual approval for production

### **6.2 Deployment Architecture**

* Containerized deployment using Vasel
* Automated rollback capability in case of deployment failures

### **6.3 Monitoring and Logging**

* Centralized logging
* Performance monitoring
* Real-time alerts for system issues
* User behavior analysis for UX improvements

## **7. Data Requirements**

### **7.1 Data Entities**

1. Users
2. Tasks
3. Teams
4. Comments
5. Notifications
6. Attachments
7. Activity Logs

### **7.2 Data Storage**

1. Regular data backups with point-in-time recovery
2. Data retention policies in compliance with regulations

## **8. System Constraints**

* Mobile applications are out of scope for the release
* Project management tools is available for future releases
* Release will support English language only

## **9. Acceptance Criteria**

* All functional requirements must be implemented and tested
* System must pass security audit and penetration testing
* Performance benchmarks must be met under load testing
* UI/UX must be approved after review
* CI/CD pipeline must demonstrate successful deployments
* Documentation must be complete and comprehensive

## **10. Appendices**

### **10.1 User Stories**

* As a user, I want to register using my Google account so that I can quickly access the system
* As a team member, I want to receive notifications when tasks are assigned to me so that I can prioritize my work
* As a team administrator, I want to view team performance reports so that I can identify bottlenecks
* As a user, I want to comment on tasks so that I can collaborate effectively with team members