

CEBU INSTITUTE OF TECHNOLOGY UNIVERSITY



COLLEGE OF COMPUTER STUDIES



Software Requirements Specification

for

CollaborAid

Developers

Jesson Chyd Cultura

Jhudiel Adrian Artezuela

Harold Destura

May 7, 2025

Table of Contents

Table of Contents	3
1. Introduction	5
1.1 Purpose	5
1.2 Scope	5
1.3 Definitions, Acronyms, and Abbreviations	6
2. Overall Description	6
2.1 Product Perspective	6
2.2 User Classes and Characteristics	6
2.3 Assumptions and Dependencies	7
3. System Features	8
3.1 Web Application Features (User and Admin Dashboard)	8
3.2 Messaging and Collaboration	9
3.3 AI Chat Support	9
3.4 Profile Management	10
3.5 Live Notifications	11
3.6 Info and Support Center	11
4. Mobile Application Features	11
4.1 Task Marketplace.....	11
4.2. Messaging and Collaboration	11
4.3. AI Chat Support.....	12
4.4. Live Notifications	12
4.5. Profile Management	12
5. Use Cases	13
6. System Architecture and Data Design	11
6.1. System Architecture	12
6.2. Data Design.....	12

7. Technical Specifications	11
7.1. Security	12
7.2.. Performance	12
7.3. Availability and Scalibility	20

1. Introduction

1.1. Purpose

CollaborAid is a platform that allows users to post tasks or activities they need help with and enables others to offer assistance. The platform includes web and mobile applications, providing a collaborative space with messaging, task management, AI support, and real-time notifications. The platform incorporates a reward system that motivates users to help others, creating a dynamic and supportive community.

1.2 Scope

CollaborAid includes the following core functionalities:

- **Web Application Features** (User and Admin Dashboard)
- **Mobile Application Features** (for General Users)
 - Messaging and Collaboration
 - AI Chat Support
 - Live Notifications
 - Profile Management
 - Info and Support Center

1.3 Definitions and Acronyms

- **AI** – Artificial Intelligence
- **JWT** – JSON Web Token
- **REST API** – Representational State Transfer Application Programming Interface
- **WebSocket** – Web communication protocol for real-time interaction
- **STOMP** – Simple Text Oriented Messaging Protocol
- **SockJS** – JavaScript library for WebSocket-like communication
- **Admin** – A system user responsible for managing users, tasks, and overseeing the platform
- **User** – A general user who posts tasks, accepts help requests, and engages with other users on the platform
- **Kotlin** – A statically typed programming language for modern Android development
- **Jetpack Compose** – A modern UI toolkit for building native Android apps in Kotlin

2. Overall Description

2.1 Product Perspective

CollaborAid is a web and mobile-based platform that includes:

- Frontend: Built with ReactJS, Tailwind CSS, and ShadCN UI components for the web interface. The mobile app is developed using Kotlin and Jetpack Compose.
- Backend: Developed with Spring Boot, REST API, and WebSocket for real-time communication.
- Database: Azure SQL Database for data storage.
- AI Integration: Powered by OpenAI for support-related queries.

2.2. User Classes and Characteristics

User Type	Description
User	Posts tasks, accepts help requests, communicates with other users, receives notifications, and updates profile
Admin	Manages the platform, oversees users and tasks, responds to support requests, and receives notifications

2.3 Assumptions and Dependencies

- Users must have an internet connection for accessing real-time communication features.
- The platform supports both web browsers and mobile devices.
- OpenAI API for AI chat responses.
- Integration with Azure SQL Database for data storage.
- The mobile app is built using Kotlin and Jetpack Compose for Android.

3. System Features

3.1 Web Application Features (User and Admin Dashboard)

The Web Application will include User and Admin Dashboards that provide distinct functionalities tailored to the different roles within the system.

3.1.1 User Dashboard:

- Task Management: Users can view and manage their personal tasks by adding new tasks, editing existing ones, or marking them as completed. They can also monitor their progress through task status updates.
- Recent Activity: A section will display the user's most recent actions, including accepted tasks, ongoing collaborations, and message history.
- Collaboration and Requests: Users can browse through posted tasks from others and request to participate in tasks based on their expertise.
- Personalized Insights: Users receive personalized recommendations on tasks or users they may be interested in collaborating with.

3.1.2 Admin Dashboard:

- User Management: Admins can manage user accounts, including activating/deactivating user accounts, assigning roles, and viewing user profiles.

- Task Overview: Admins can review all posted tasks on the platform, approve or reject help requests, and monitor task completion statuses.
- Reward Distribution: Admins are responsible for distributing rewards based on user participation, task completion, and overall engagement with the platform.
- Activity Monitoring: Admins can access detailed activity logs and data analytics to monitor system usage and performance.

3.2 Messaging and Collaboration

The Messaging and Collaboration feature allows users and admins to engage in real-time communication. It supports direct one-on-one messaging as well as group conversations where users can collaborate on tasks.

- One-on-One Communication: Users and admins can send private messages to each other, ask for assistance, and provide feedback.
- Messaging Interface: Built using WebSockets, the messaging interface allows for instant message delivery with minimal latency.

Technology: WebSocket with STOMP over SockJS ensures low-latency and real-time messaging.

3.3 AI Chat Support

The AI Chat Support feature provides an AI-powered support system to assist users with frequently asked questions and handle common inquiries without human intervention. This AI chat is integrated with a live chat feature for more complex issues that require human support.

3.3.1 AI-Driven Responses:

- For common queries, the AI system can provide pre-programmed or dynamically generated responses based on the user's input.
- The AI system is integrated with the OpenAI GPT model, allowing it to provide intelligent, context-aware assistance.

3.3.2 Live Chat Support:

- If the user's question is too complex or cannot be answered by AI, they are directed to a live chat support agent (admin or support staff) for further assistance.

Technology: OpenAI GPT for AI responses, integrated with the chat interface.

3.4 Profile Management

Users and admins can manage their profiles, including updating personal details, profile pictures, and passwords. The system also supports basic account settings, such as enabling/disabling notifications and privacy settings.

1. Profile Updates:

- Users can update their username, email, bio, and profile picture to keep their information current.
- Password recovery and security settings ensure that users can maintain account security.

2. Personal Information:

- Users can edit their personal bio and contact details, which helps in team formations or collaborations.

Technology: Frontend will be built using ReactJS with form validation and backend integration for storing updates in the database.

3.5 Live Notifications

The Live Notifications feature ensures that users and admins are kept up to date with important events, such as new messages, task updates, admin alerts, system announcements, and support replies.

3.5.1 Real-Time Updates:

- Notifications are delivered in real time, enabling users to respond quickly to changes such as new tasks, incoming messages, or system alerts.

3.5.2 Types of Notifications:

- Task Notifications: Notifications related to task acceptance, completion status, or new task postings.
- Message Notifications: Alerts when a new message is received, or a reply has been sent.
- System Notifications: Admin or system-generated messages that provide important updates.

Technology: WebSocket for notifications in the web app and for mobile push notifications.

3.6 Info and Support Center

The Info and Support Center provides users with essential platform documentation, FAQs, and user guides, helping them navigate the platform and find answers to common questions.

1. Documentation:

- Detailed user guides, FAQs, and terms of service will be available in this section.

2. Support Forms:

- Users can submit support tickets or inquiries directly through a contact form. Responses will be handled by admins or support staff.

Technology: Static pages for documentation and dynamic forms for submitting support inquiries.

4. Mobile Application Features (*Kotlin + Jetpack Compose*)

4.1 Task Marketplace (*Kotlin + Jetpack Compose*)

The Task Marketplace allows users to post tasks and browse tasks posted by others. They can accept tasks that match their expertise or interest.

- **Post Tasks:** Users can describe the task, set deadlines, and specify the required skills.
- **Browse and Accept Tasks:** Users can search for tasks they wish to contribute to or help with.

Technology: Built using Kotlin and Jetpack Compose for Android, providing a smooth and responsive user experience.

4.2 Messaging and Collaboration (*Kotlin + Jetpack Compose*)

The Messaging feature enables users to communicate in real time with other users for task-related discussions or general inquiries.

- **One-on-One Chat:** Private conversations between users or with admins for task collaboration.

Technology: Integrated using WebSockets for real-time messaging. Kotlin and Jetpack Compose for UI development.

4.3 AI Chat Support (Kotlin + Jetpack Compose)

The mobile app provides an AI-powered support chat for users to get quick answers to common questions.

- **AI Responses:** Automatically generated answers for frequently asked questions, based on AI algorithms.
- **Live Chat:** The AI chat redirects users to a live support agent for more complex inquiries.

Technology: OpenAI GPT integrated with the mobile app for AI-powered responses.

4.4 Live Notifications (Kotlin + Jetpack Compose)

Users will receive push notifications for task updates, new messages, and system alerts.

- **Push Notifications:** Delivered to ensure users receive updates in real-time.
- **Custom Alerts:** Users are notified when they are tagged in a task, when a new message is received, or when their task status changes.

Technology: Websocket for push notifications, Kotlin and Jetpack Compose for mobile UI.

4.5 Profile Management (Kotlin + Jetpack Compose)

Users can manage their profiles directly from the mobile app, including updating their photo, bio, and password.

- Profile Updates: Users can update their information from the mobile app, including setting privacy options and managing account settings.

Technology: Kotlin and Jetpack Compose for UI elements and backend integration.

5. Use Cases

Use Case ID	Title	Steps
UC1	Post a Help Request	<ol style="list-style-type: none">1. User logs into the system2. Fills out a form with task details (title, description, category, tags)3. Submits the form4. Task is posted to the marketplace
UC2	Real-Time Chat with Admin	<ol style="list-style-type: none">1. User logs into the system2. Opens the chat interface3. Selects Admin from the list4. Sends and receives messages in real time
UC3	Chat Between Users	<ol style="list-style-type: none">1. User logs into the system2. Opens the chat interface3. Selects another user4. Sends and receives messages in real time
UC4	AI Support Chat	<ol style="list-style-type: none">1. User opens the AI chat interface2. Submits a query3. AI system returns a response
UC5	Receive Real-Time Notification	<ol style="list-style-type: none">1. A triggering event occurs (e.g., new message, task accepted)2. System emits a notification

		3. User receives the notification in real time
--	--	--

6. System Architecture and Data Design

6.1 System Architecture

The system is composed of the following components:

Frontend: ReactJS, Tailwind CSS, ShadCN for UI components for web users.

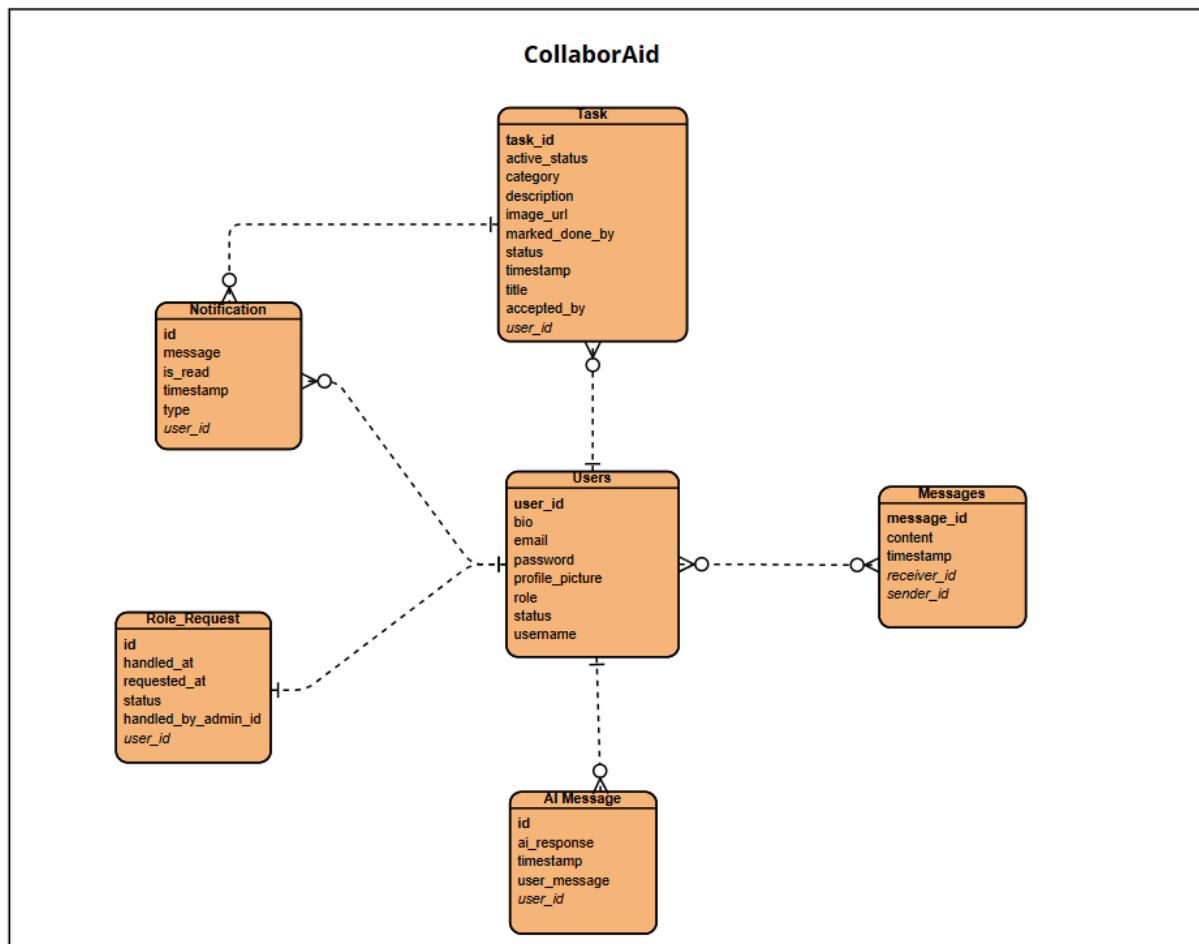
Mobile: Kotlin and Jetpack Compose for Android.

Backend: Spring Boot, WebSocket for real-time communication, REST API for service access.

Database: Azure SQL Database.

AI Integration : OpenAI API

6.2. Data Design



7. Technical Specifications

7.1 Security

- **JWT Authentication:**
 - All API endpoints and WebSocket connections are secured using JSON Web Tokens.
 - Tokens are verified with expiration time and user roles before granting access.
- **Data Encryption:**
 - Passwords are stored using BCrypt hashing.
 - Sensitive data is encrypted both in-transit (HTTPS) and at-rest (Azure SQL TDE).
- **OpenAI API Key Security:**
 - Stored securely in environment variables
 - Never exposed to the client side.

7.2 Performance

- **Real-time Communication:**
 - Messaging and notification updates are delivered within < 1 second latency under normal conditions.
- **Concurrency Support:**
 - Optimized to handle 100+ concurrent users using async processing and WebSocket event queues.
- **Load Management:**
 - Backend services are stateless and support horizontal scaling.

- Database supports read replicas to manage load on read-heavy operations.

7.3 Availability and Scalability

- **Cloud Hosting:**

- Hosted on Vercel, a modern frontend platform with automatic CI/CD, edge networks for fast delivery, and custom domain support.
- Hosted on Render, a cloud platform supporting containerized and scalable backend services with automated deploy pipelines and health checks

- **Database Redundancy:**

- Azure SQL provides high availability with geo-redundant backups and failover groups.

- **Disaster Recovery:**

- Nightly backups and snapshot storage ensure fast recovery in case of failure.

- **Scalability Design:**

- System follows microservices-inspired modularity, allowing separate services (auth, messaging, task) to scale independently.