Software Requirements Specification

for

Collab Nest

Version 1.0 approved

Prepared By

Saina Abdul Hamid -2024301008

Kalpak Patil-2024301020

15 August 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Collab Nest	15-08-2025	Initial Draft	1.0 draft 1

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) describes the functional and non-functional requirements for **CollabNest** version 1.0. CollabNest is a mobile application designed to help college and university students find collaborators, form teams, and coordinate events through a centralized digital platform.

This document is intended for developers, testers, project managers, institutional reviewers, and end-users who require a detailed understanding of the system's scope, features, and operational constraints. Unless otherwise stated, all requirements described are prioritized for implementation in the initial release.

1.2 Document Conventions

- **Font**: Times Roman, 12 pt.
- **Section numbering**: Follows hierarchical IEEE standard numbering (e.g., 1, 1.1, 1.1.1).
- **Requirement labels**: Functional requirements are prefixed with "REQ-" followed by a unique number.
- **Priority**: Each requirement is implicitly high priority unless stated otherwise.

1.3 Intended Audience and Reading Suggestions

- **Developers**: Use this document to implement all described features according to specifications.
- **Testers**: Use detailed functional requirements to design and execute test cases.
- **Project Managers**: Track scope, ensure compliance, and monitor deliverables.
- **End Users (Students, Event Organizers)**: Understand system capabilities and constraints.
- **Reviewers (Committee)**: Evaluate design completeness and academic alignment.

Reading Order Recommendation:

- Section 1 – Introduction (overview)
- Section 2 – Overall Description (context and major functions)
- Section 3 – External Interface Requirements (technical integrations)
- Section 4 – System Features (functional breakdown)
- Section 5 – Nonfunctional Requirements (performance, security, quality attributes)

1.4 Product Scope

CollabNest addresses recurring challenges in academic institutions, including:

- Difficulty in finding teammates with relevant skills or interests.
- Overlapping event schedules between departments.
- Inefficient communication among committees and participants.

The platform enables:

- Student profile creation (resume, skills, interest areas).
- Intelligent team matchmaking based on shared or complementary skills.
- A real-time master calendar for all college events.
- Built-in chat for group coordination.
- Live notifications for event updates and team invitations.

Technology Stack:

- **Frontend (Mobile)**: Android application developed in **Kotlin** using Android Studio.
- **Backend (Core Services)**: **Spring Boot** application providing REST APIs, event scheduling logic, authentication management, and data access control.

- **Cloud Services**: **Firebase** for real-time database, file storage, and push notifications (via Firebase Cloud Messaging).

Benefits include:

- Faster and more inclusive team formation.
- Reduced scheduling conflicts.
- Enhanced networking opportunities within the student community.

1.5 References

- IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications.
- Android Developers – Android Studio Documentation – https://developer.android.com/
- Google Firebase Documentation – https://firebase.google.com/docs/

2. Overall Description

2.1 Product Perspective

CollabNest is a **new, self-contained mobile application** designed for Android devices. It will use a **hybrid backend architecture** combining **Spring Boot** and **Firebase**.

- **Spring Boot** will provide REST APIs, handle business logic, enforce security rules, and manage smart scheduling (conflict detection for events).
- **Firebase** will provide cloud storage, real-time database functionality, file uploads (e.g., resumes), and push notifications via Cloud Messaging.
- The application will integrate these services to create a seamless collaboration experience for students and organizers.

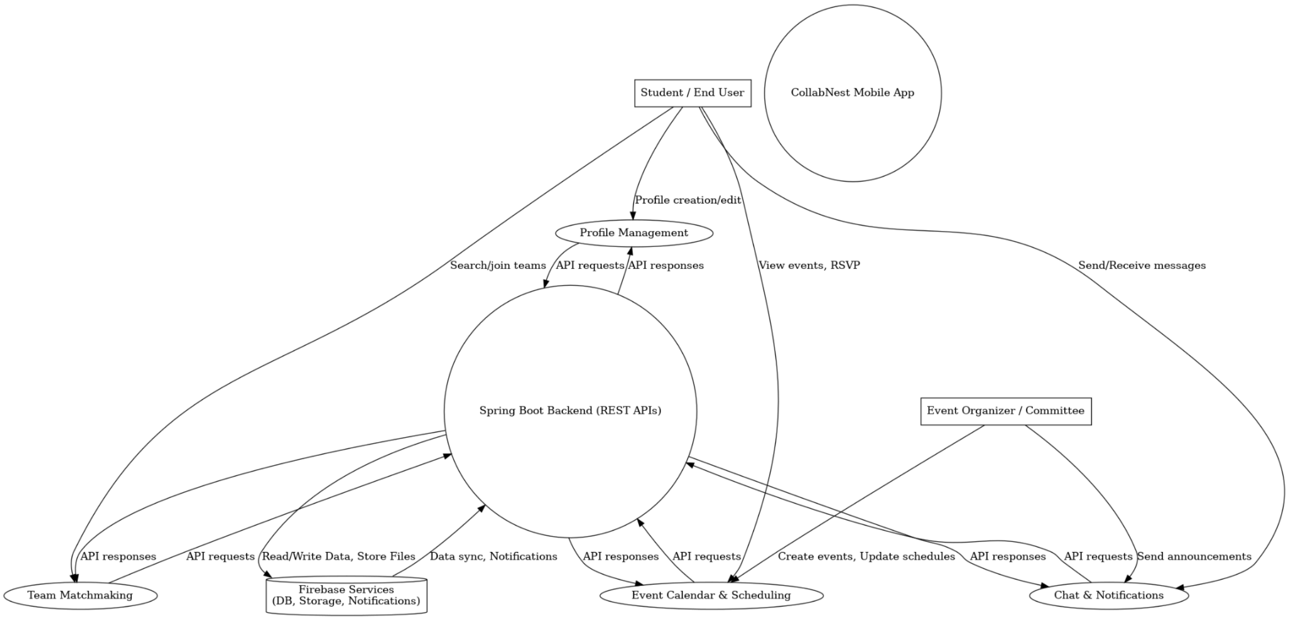
System Context:

- End-users interact with the mobile app (Android).
- App communicates with Spring Boot backend via secure HTTPS REST APIs.
- Backend connects to Firebase services for data storage, notifications, and file handling.

2.2 Product Functions

High-level features include:

- **Personalized User Profile Management** – Create/edit profile, upload resume, list skills/interests.
- **Team Discovery & Matchmaking** – Search/filter for potential teammates.
- **Individual Player Status** – Display current team membership/availability.
- **Real-Time Master Calendar** – View all scheduled college events; detect conflicts.
- **Built-in Chat** – Group messaging for teams and event committees.
- **Live Notifications** – Alerts for event updates, team invites, and scheduling changes.



2.3 User Classes and Characteristics

Students –

- Frequent users.
- Require access to matchmaking, profiles, and master calendar.
- Basic to intermediate technical skills.

Event Organizers / Committee Members –

- Moderate-frequency users.
- Require event creation, scheduling, and announcement features.
- May have higher access privileges for event management.

2.4 Operating Environment

- **Client Platform**: Android smartphones (Android 12 version or later).
- **Backend**: Firebase services (Authentication, Realtime Database, Cloud Messaging).
- **Internet Connectivity**: Required for profile updates, event synchronization, chat, and notifications.
- **Development Environment**: Android Studio IDE

2.5 Design and Implementation Constraints

- Must be implemented in **Kotlin** within Android Studio.
- Backend must use **Firebase** services; no alternative backend allowed for initial release.
- UI must be optimized for mobile screens (min. 5-inch display).
- Push notifications must comply with **Google FCM** policies.
- All data must be encrypted during transmission (HTTPS/TLS).

2.6 User Documentation

The following documentation will be provided:

- **User Manual** (PDF) – step-by-step guide for profile creation, matchmaking, event browsing, and chat.
- **FAQ Section** – common troubleshooting solutions accessible from the app menu.

2.7 Assumptions and Dependencies

- All users have Android devices with active internet access.
- Institutional event data will be provided by authorized organizers.
- Firebase services remain operational and accessible in the deployment region.
- The institution will promote app adoption among students and committees.

3. External Interface Requirements

3.1 User Interfaces

UI-1- **Login/Registration Screen** – Secure login via email/password, institutional ID.

UT2- **Dashboard** – Displays quick links to profile, teams, events, and chat.

UT3- **Profile Management Screen** – Allows upload of resume (PDF), adding skills, interests, and availability status.

UT4- **Team Discovery Screen** – Search/filter for available collaborators based on skills, interests, and event participation.

UT5- **Event Calendar Screen** – Displays upcoming events with filtering by department or type; conflict warnings for overlapping events.

UT6- **Chat Interface** – Group and direct messaging with real-time updates and read receipts.

UT7 - **Notifications Panel** – Displays event updates, team invites, and important announcements.

3.2 Hardware Interfaces

No hardware interfaces have been identified.

3.3 Software Interfaces

- **Operating System**: Android 8.0 or later for client devices.
- **Backend**: Spring Boot 3.x providing REST APIs for authentication, profile management, matchmaking, event scheduling, and chat integration.
- **Database and Storage**: Firebase Realtime Database / Firestore for structured data, Firebase Cloud Storage for file uploads.
- **Notification Service**: Firebase Cloud Messaging (FCM) for push notifications.
- **API Communication**: JSON over HTTPS between mobile app and Spring Boot backend.
- **Authentication**: Spring Security integrated with Firebase Authentication for secure user identity management.

3.4 Communications Interfaces

- **Protocols**:
  1. HTTPS for all client-server communication.

  2.WebSocket or Firebase Realtime Database event listeners for real-time chat and calendar updates.
- **Data Formats**: JSON for request/response payloads.
- **Encryption**: TLS 1.2+ for data in transit; Firebase-managed encryption for data at rest.
- **Notification Delivery**: Push notifications via Firebase Cloud Messaging (FCM).

4. System Features

4.1 Create accounts and log in securely.

4.1.1 Description and Priority

Priority: High (core access control for all other features).

4.1.2 Stimulus/Response Sequences

Stimulus: A new student clicks “Sign Up” and enters their details.
Response: The system validates the data, verifies the email, and creates the account.
Stimulus: A registered user enters login credentials.
Response: The system validates credentials and grants access.

4.1.3 Functional Requirements

REQ-1: The system shall allow new users to register using a valid college email address.

REQ-2: The system shall send a verification link to the provided email to activate the account.

REQ-3: The system shall authenticate returning users using email and password.

REQ-4: The system shall provide a “Forgot Password” option to reset passwords via email verification.

4.2 Personalized User Profile Management

4.2.1 Description and Priority

Allows users to create and update personal profiles with skills, interests, and availability status.

Priority: High (enables matchmaking and collaboration).

4.2.2 Stimulus/Response Sequences

Stimulus: A user opens the “Edit Profile” page and updates skills/interests.

Response: The system saves and displays updated details in their public profile.

4.2.3 Functional Requirements

REQ-1: The system shall allow users to upload resumes, portfolios, and certificates.

REQ-2: The system shall allow users to specify their skills, interests, and experience.

REQ-3: The system shall display a “Current Team Status” field (Available / In a Team).

REQ-4: The system shall let users control the visibility of personal information (public/private).

4.3 Team Discovery & Matchmaking

4.3.1 Description and Priority

Enables users to find peers for events and projects using search and filters.

Priority: High (core collaboration feature).

4.3.2 Stimulus/Response Sequences

Stimulus: A user searches for available teammates with “UI/UX design” skills.

Response: The system returns a list of profiles matching the criteria.

4.3.3 Functional Requirements

REQ-1: The system shall allow users to search teammates by skill, department, or event.

REQ-2: The system shall provide filters such as skills, availability, and event interest.

REQ-3: The system shall suggest potential matches based on profile similarity.

REQ-4: The system shall allow sending and accepting/declining team invitations.

4.4 Event & Master Calendar Management

4.4.1 Description and Priority

Provides a centralized calendar for all campus events to avoid scheduling conflicts.

Priority: High.

4.4.2 Stimulus/Response Sequences

Stimulus: An organizer posts a new hackathon date.

Response: The calendar updates instantly for all users.

4.4.3 Functional Requirements

REQ-1: The system shall allow authorized organizers to add/edit/delete events.

REQ-2: The system shall display all upcoming events in a master calendar view.

4.5 One-to-One Chat Communication

4.5.1 Description and Priority

Enables direct, real-time messaging between two users within the platform.

Priority: Medium.

4.5.2 Stimulus/Response Sequences

Stimulus: A user sends a message to another user via the one-to-one chat interface.

Response: The recipient sees the message instantly, along with a notification.

4.5.3 Functional Requirements

REQ-1: The system shall allow one-to-one chat between registered users.

REQ-2: The system shall allow sending text, images in a private conversation.

REQ-3: The system shall store the full message history for both participants to view later.

REQ-4: The system shall notify the recipient in real time when a new message arrives.

4.6 Live Notifications & Alerts

4.6.1 Description and Priority

Keeps users informed about important updates like team invites, event changes, or deadlines.

Priority: High.

4.6.2 Stimulus/Response Sequences

Stimulus: A user receives a team invite.

Response: The system shows a notification and allows the user to accept or decline.

4.6.3 Functional Requirements

REQ-1: The system shall provide real-time notifications for team requests, chat messages, and event changes.

REQ-2: The system shall allow users to customize notification preferences.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

PE-1: The system shall support up to 500 concurrent users during peak usage hours (8:00 AM to 10:00 AM local time), with an estimated average session duration of 12 minutes.

PE-2: All Resumes and portfolio related links / documents generated by the system shall be fully downloadable in no more than 5 seconds over a 5 Mbps internet connection.

PE-3: Responses to search queries, match recommendations, or profile lookups shall take no longer than 4 seconds to display after the user submits the request.

PE-4: The system shall deliver one-to-one chat messages to the recipient within 2 seconds of sending.

5.2 Safety Requirements

No safety requirements have been identified for this application, as its usage does not pose direct physical risks.

5.3 Security Requirements

SE-1: All network transactions involving personal data (e.g., name, profile details) shall be encrypted using TLS 1.3 or higher.

SE-2: Users shall be required to log in for all operations.

SE-3: The system shall permit only authenticated users to send messages or create events.

SE-4: Only authorized administrators may modify or delete platform-wide event related posts, user roles.

SE-5: The system shall ensure that private chat conversations can only be accessed by their respective participants.

5.4 Software Quality Attributes

SEQ-Availability-1: The application shall maintain 99.5% uptime during normal operation, excluding scheduled maintenance.
SEQ-Usability-1: The interface shall allow a first-time user to successfully create an account and join their first event within 5 minutes without external help.
SEQ-Maintainability-1: The system's codebase shall be modular and documented such that a new developer can add or modify a feature with minimal onboarding time.
SEQ-Portability-1: The mobile application shall be compatible with Android 12+ .

6. Other Requirements

Database: Firebase Realtime Database/Firestore for structured data; Firebase Cloud Storage for files.

Internationalization: Support for multiple languages; default is English (US).

Legal: Comply with relevant data protection laws; obtain user consent for personal data.

Reuse: Modular Spring Boot backend and reusable Firebase service classes.

Operations: Must work on both Wi-Fi and mobile data; target 99.9% cloud service uptime.

Appendix A: Glossary

Admin – A user role with permissions to manage events, users, and platform settings in CollabNest.

Attachment – Any file (e.g., PDF, image, document) shared within chats or events.

Chat – A real-time text communication feature between users, currently supporting only one-to-one messaging.

CollabNest – The name of the platform designed for event collaboration, team management, and communication.

Event – A scheduled activity created by users on CollabNest, which can have details such as description, date, and participants.

Event Creator – A user who creates and manages an event.

Event Lifecycle – The series of states an event goes through — from Draft to Published, Ongoing, Completed, or Canceled.

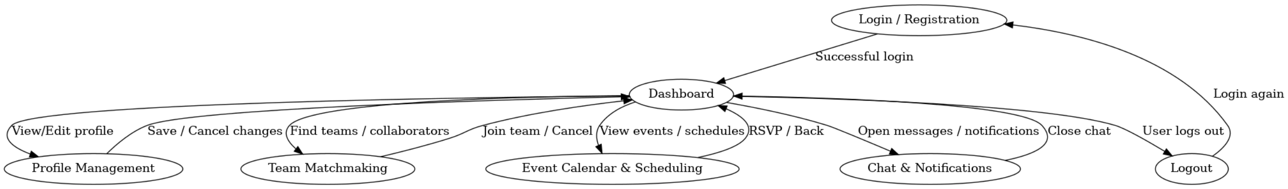
One-to-One Chat – Private messaging between two users. Group chat is not currently implemented.

Participant – A user who has joined an event but is not necessarily the event creator.

Team – A group of participants collaborating on an event or project.

User – Any registered individual who can create, join, and participate in events on CollabNest.

Appendix B: Analysis Models



Appendix C: To Be Determined List

TBD-1: Exact maximum number of concurrent users the system must support during peak load.