

---

# **Software Requirements Specification**

**for**

# **Project Management App**

**Version 1.0 approved**

**Prepared by:**

**Hanok Shrestha  
Padam Khatri  
Kul Bahadur Karki**

**Branch: 5CED**

**17-09-2023**

# Table of Contents

<b>1. Introduction</b>	<b>4</b>
1.1 Purpose	4
1.2 Document Conventions	4
1.3 Intended Audience and Reading Suggestions	4
1.4 Project Scope	4
1.5 References	4
<b>2. Overall Description</b>	<b>5</b>
2.1 Product Perspective	5
2.2 Product Features	5
2.2.1 User Registration	5
2.2.2 User Management	5
2.2.3 Task Management	5
2.2.4 Real-time Collaboration	5
2.2.5 Notifications	5
2.2.6 Reporting and Analytics	6
2.3 User Classes and Characteristics	6
2.3.1 Company Administrators	6
2.3.2 Project Managers	6
2.3.3 Team Members	6
2.4 Operating Environment	6
2.5 Design and Implementation Constraints	6
2.6 User Documentation	7
2.7 Assumptions and Dependencies	7
<b>3. System Features</b>	<b>7</b>
3.1 User Registration	7
3.1.1 User Registration Form	7
3.1.2 User Role Assignment	7
3.2 User Management	7
3.2.1 User Addition	7
3.2.2 User Editing	7
3.2.3 User Deletion	8
3.3 Task Management	8
3.3.1 Task Creation	8
3.3.2 Task Assignment	8
3.3.3 Task Tracking	8
3.4 Real-time Collaboration	8
3.4.1 Real-time Task Updates	8
3.5 Notifications	8

3.5.1 Task Assignment Notifications	8
3.5.2 Task Deadline Notifications	8
3.6 Reporting and Analytics	9
3.6.1 Report Generation	9
<b>4. External Interface Requirements</b>	<b>9</b>
4.1 User Interfaces	9
4.1.1 Web Interface	9
4.1.2 Mobile Interface	9
4.2 Hardware Interfaces	9
4.3 Software Interfaces	9
4.4 Communications Interfaces	9
<b>5. Other Nonfunctional Requirements</b>	<b>10</b>
5.1 Performance Requirements	10
5.2 Safety Requirements	10
5.3 Security Requirements	10
5.4 Software Quality Attributes	10
<b>6. Other Requirements</b>	<b>10</b>
<b>Appendix A: Glossary</b>	<b>10</b>
<b>Appendix B: Analysis Models</b>	<b>10</b>
<b>Appendix C: Issues List</b>	<b>11</b>

# 1. Introduction

## 1.1 Purpose

The purpose of this comprehensive document is to define the detailed requirements and specifications for the Project Management App. This document serves as a foundational guide for the development and deployment of the application. It outlines the desired functionality, architectural design, and performance expectations, with a specific focus on leveraging Firebase as the backend platform.

## 1.2 Document Conventions

This document follows a standardized set of conventions to ensure clarity and consistency in expressing requirements. "Must" indicates mandatory requirements that are essential for the system's core functionality. "Should" represents preferred, yet non-mandatory, features that enhance the user experience. "May" suggests optional features that could be considered for future iterations of the application.

## 1.3 Intended Audience and Reading Suggestions

The primary audience for this document includes developers, quality assurance professionals, project managers, and key stakeholders involved in the development and deployment of the Project Management App. It is recommended that readers have a foundational understanding of software development principles, Firebase, and project management concepts to fully grasp the contents of this document.

## 1.4 Project Scope

The Project Management App is envisioned as a comprehensive tool designed to streamline and optimize project management processes within organizations. It encompasses a wide array of features, including user management, task assignment, real-time collaboration, and reporting. Central to this solution is the utilization of Firebase as the backend platform, providing real-time data synchronization, robust authentication, and cloud-based storage capabilities.

## 1.5 References

This section is reserved for listing any external documents or resources that may be referred to throughout the development process. It may include documentation related to Firebase, Flutter, or project management methodologies.

## 2. Overall Description

### 2.1 Product Perspective

The Project Management App is positioned as a client-server application. The client-side, developed using the Flutter framework, is responsible for delivering a seamless and responsive user experience across both web and mobile platforms. The server-side leverages Firebase as the backend platform, facilitating real-time data storage, authentication, and notifications. This architectural approach ensures efficient data exchange between users and the cloud-hosted backend.

### 2.2 Product Features

#### 2.2.1 User Registration

- Must: Users can register with their company details, a crucial step in onboarding users into the system.
- Must: Users can be assigned after registration, enabling role-based access control and permissions.

#### 2.2.2 User Management

- Must: Administrators are empowered to add, edit, and delete users, ensuring robust user management capabilities.
- Must: User profiles are comprehensive, encompassing critical information such as name, email, role, and contact details, thereby facilitating effective communication within the system.

#### 2.2.3 Task Management

- Must: Users can create, assign, and meticulously track tasks, enhancing project planning and execution.
- Must: Tasks are imbued with attributes such as title, description, due date, and status, allowing for precise task management.

#### 2.2.4 Real-time Collaboration

- Must: Real-time collaboration is a core feature, enabling multiple users to collaborate on tasks concurrently.
- Must: Changes made by one user are immediately reflected to others, facilitating seamless teamwork and enhancing productivity.

#### 2.2.5 Notifications

- Must: Users receive notifications for task assignments and deadlines, ensuring that important project-related events are never missed.

### 2.2.6 Reporting and Analytics

- Should: The app features reporting capabilities that allow users to generate reports on various aspects, including project progress, user activity, and task completion.

## 2.3 User Classes and Characteristics

### 2.3.1 Company Administrators

- Company administrators occupy a critical role within the system, wielding the authority to manage users, tasks, and projects.
- They have full access to all features and settings, granting them the ability to configure the system to meet the organization's needs.

### 2.3.2 Project Managers

- Project managers play a pivotal role in overseeing project-specific tasks.
- They can create and manage projects and have the authority to assign tasks to team members, streamlining project planning and execution.

### 2.3.3 Team Members

- Team members constitute the majority of users within the system, focusing on task execution.
- They can view and update tasks assigned to them, enabling them to stay informed and engaged in project-related activities.
- They receive timely notifications for task assignments and deadlines, helping them stay organized and meet project milestones effectively.

## 2.4 Operating Environment

The Project Management App is designed to be accessible across various platforms, including:

- Web browsers such as Google Chrome, Firefox, and Safari.
- Mobile devices running iOS and Android operating systems.

## 2.5 Design and Implementation Constraints

- Front-end development is executed using the Flutter framework, ensuring a responsive and consistent user experience across both web and mobile platforms.
- Firebase serves as the backend platform, providing real-time database functionality, robust authentication, and cloud-based storage.
- Security is a paramount concern, necessitating the implementation of stringent measures, including user authentication and data encryption.

## 2.6 User Documentation

To ensure that users can effectively navigate and utilize the app, comprehensive user guides and tutorials will be provided. These resources will empower users to leverage the application's full potential while minimizing the learning curve.

## 2.7 Assumptions and Dependencies

- Assumptions are predicated on the assumption of basic user computer literacy, ensuring that users can interact with the system effectively.
- Dependencies encompass Firebase services for authentication, real-time database functionality, cloud-based storage, and cloud messaging for notifications. These services are central to the application's core functionality and are expected to function seamlessly.

## 3. System Features

### 3.1 User Registration

#### 3.1.1 User Registration Form

- Must: The system must provide users with a registration form where they can input their company details, including company name, email, and other relevant information.
- Must: The system must implement validation mechanisms to verify the accuracy and completeness of user registration data.

### 3.2 User Management

#### 3.2.1 User Addition

- Must: Administrators must have the capability to add new users to the system by providing their relevant details.
- Must: The system must validate user input during the user addition process to ensure data integrity.

#### 3.2.2 User Editing

- Must: Administrators must be able to edit user profiles, allowing them to update user information as needed.
- Must: User profile editing should be accompanied by validation mechanisms to maintain data accuracy.

#### 3.2.3 User Deletion

- Must: Administrators must possess the authority to deactivate or delete user accounts when necessary, maintaining user management flexibility.

### 3.3 Task Management

#### 3.3.1 Task Creation

- Must: Users should be able to create tasks, including details such as titles, descriptions, priorities, due dates, and assignment.
- Must: The system must enforce data integrity by validating task creation inputs.

#### 3.3.2 Task Assignment

- Must: Project managers must be able to assign tasks to team members, ensuring efficient task distribution.
- Must: Team members should have access to a dashboard where they can view tasks assigned to them.



### 3.3.3 Task Tracking

- Must: Users should be able to update task statuses, allowing for real-time task tracking and progress monitoring.
- Should: The system should support the attachment of files and comments to tasks, facilitating collaborative workflows.

## 3.4 Real-time Collaboration

### 3.4.1 Real-time Task Updates

- Must: The system should support real-time collaboration, enabling multiple users to work on tasks simultaneously.
- Must: Changes made by one user must be immediately synchronized and visible to others, fostering seamless teamwork.

## 3.5 Notifications

### 3.5.1 Task Assignment Notifications

- Must: Users must receive notifications when tasks are assigned to them, ensuring that they are promptly aware of their responsibilities.

### 3.5.2 Task Deadline Notifications

- Must: The system must send notifications as task due dates approach, assisting users in meeting project deadlines effectively.

## 3.6 Reporting and Analytics

### 3.6.1 Report Generation

- Should: The app should provide users with the ability to generate reports to track project progress, user activity, and task completion.
- Should: Data visualization tools, such as charts and graphs, should be integrated to help users derive meaningful insights from project data.

## 4. External Interface Requirements

### 4.1 User Interfaces

#### 4.1.1 Web Interface

- Must: The web interface should be accessible through modern web browsers, including but not limited to Google Chrome, Mozilla Firefox, and Safari.
- Must: The web interface should be responsive and compatible with a variety of screen sizes and resolutions.

#### 4.1.2 Mobile Interface

- Must: The mobile interface must be responsive and compatible with iOS and Android devices, ensuring a consistent user experience across mobile platforms.

### 4.2 Hardware Interfaces

- Must: The application should be designed to run efficiently on standard hardware configurations for web and mobile devices, optimizing user accessibility.

### 4.3 Software Interfaces

- Must: The application must effectively communicate with Firebase services, including Firebase Authentication, Firebase Realtime Database, Firebase Cloud Storage, and Firebase Cloud Messaging, to facilitate user authentication, data storage, and notifications.

### 4.4 Communications Interfaces

- External APIs may be integrated for email notifications, providing an additional communication channel for project-related updates and notifications.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

- Must: The application must exhibit responsive behavior, with user interactions yielding responses within a one-second time frame.
- Must: The system should be designed to handle concurrent users efficiently, supporting up to 1000 simultaneous users without compromising performance.

### 5.2 Safety Requirements

- Must: User data, including sensitive information, must be securely stored and encrypted to safeguard user privacy and protect against data breaches.

### 5.3 Security Requirements

- Must: The application must implement Firebase Authentication to ensure secure user authentication and access control.
- Must: Firebase security rules must be configured to restrict unauthorized data access and maintain data integrity.
- Must: Regular security audits and updates must be conducted to proactively identify and address potential security vulnerabilities.

### 5.4 Software Quality Attributes

- Must: The application should adhere to best practices in software development to ensure maintainability and extensibility, facilitating future updates and feature enhancements.
- Must: The system must be designed to be scalable, accommodating an increasing user base and growing data volumes.