Software Requirements Specification (SRS)

Project Title

Web-Based Project Management System (PMS)

Version

1.0

Prepared For

Rider

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document defines the functional and non-functional requirements for a Web-Based Project Management System. The system is intended to help organizations plan projects, manage tasks, collaborate with teams, track progress, manage files, identify risks, and generate reports.

1.2 Scope

The Project Management System (PMS) is a web application that allows organizations to: - Create and manage projects - Break projects into tasks and subtasks - Assign tasks to team members - Track progress and deadlines - Communicate within teams - Manage project-related documents - Identify risks and issues - Generate performance and status reports

The system will be accessible via modern web browsers and designed to be scalable and secure.

1.3 Definitions, Acronyms, and Abbreviations

- **PMS**: Project Management System
- **Admin**: User with full system control
- **PM**: Project Manager
- **RBAC**: Role-Based Access Control
- **SDLC**: Software Development Life Cycle

1.4 References

- IEEE 830 / ISO/IEC/IEEE 29148 SRS Standard

2. Overall Description

2.1 Product Perspective

The PMS is a standalone web-based system that may integrate with third-party services such as email, calendar tools, or cloud storage in future versions.

2.2 Product Functions

- User authentication and authorization
- Project and task management
- Team collaboration
- File storage and versioning
- Risk and issue management
- Reporting and analytics

2.3 User Classes and Characteristics

Actor	Description
Admin	Manages system settings, users, and reports
Project Manager	Creates projects, assigns tasks, monitors progress
Team Member	Works on assigned tasks and updates status
Viewer	Read-only access to projects and reports

2.4 Operating Environment

- Web browsers: Chrome, Firefox, Edge, Safari
- Server: Linux-based cloud or on-premise server
- Database: Relational (PostgreSQL/MySQL)

2.5 Design and Implementation Constraints

- Must support RBAC
- Must use secure authentication
- Must be scalable

2.6 Assumptions and Dependencies

- Users have internet access
- Email service available for notifications

3. System Actors

Primary Actors

- Admin
- Project Manager
- Team Member

Secondary Actors

- Notification Service
- File Storage Service

4. Functional Requirements

4.1 User Authentication

FR-1: The system shall allow users to register and log in using email and password.

FR-2: The system shall enforce role-based access control.

4.2 Project Management

FR-3: The system shall allow Project Managers to create, update, and delete projects.

FR-4: The system shall allow projects to have milestones and deadlines.

4.3 Task Management

FR-5: The system shall allow tasks and subtasks creation.

FR-6: The system shall allow assignment of tasks to users.

FR-7: The system shall track task status (To-Do, In Progress, Review, Done).

4.4 Communication

FR-8: The system shall provide task-level comments.

FR-9: The system shall log all user activities.

4.5 File Management

FR-10: The system shall allow file uploads linked to projects or tasks.

FR-11: The system shall maintain file version history.

4.6 Risk & Issue Management

FR-12: The system shall allow users to flag tasks as blocked or delayed.

FR-13: The system shall generate alerts for critical risks.

4.7 Reports

FR-14: The system shall generate project status reports.

FR-15: The system shall allow exporting reports in PDF and Excel formats.

---

5. Use Case Flows

5.1 Use Case: Create Project

Actor: Project Manager

Normal Flow: 1. Actor logs in 2. Selects "Create Project" 3. Enters project details 4. Saves project 5. System confirms creation

Alternative Flow: - 3a. Missing required fields → system prompts error

---

5.2 Use Case: Assign Task

Actor: Project Manager

Normal Flow: 1. Select project 2. Create or select task 3. Assign user 4. Set deadline 5. Save changes

Alternative Flow: - 3a. User unavailable → system shows warning

---

6. Non-Functional Requirements

6.1 Performance Requirements

- System shall support at least 1,000 concurrent users
- Page response time < 2 seconds

6.2 Security Requirements

- Passwords must be hashed
- HTTPS required
- RBAC enforced

6.3 Reliability

- System uptime ≥ 99.5%

6.4 Usability

- Intuitive UI
- Mobile-responsive design

6.5 Scalability

- Support horizontal scaling

---

7. SDLC Model

The system shall follow an **Agile SDLC**:

1. Requirement Analysis
2. System Design
3. Development (Sprint-based)
4. Testing
5. Deployment
6. Maintenance

---

8. Diagrams (Textual Representation)

8.1 Use Case Diagram (Textual)

Actors → Use Cases - Admin → Manage Users, View Reports, Configure System - Project Manager → Create Project, Assign Tasks, View Reports - Team Member → Update Task Status, Comment, Upload Files

---

8.2 High-Level System Architecture

[Client Browser] | [Web Application] | [Backend API] | [Database]

---

9. Data Requirements (High Level)

Entities: - User - Project - Task - File - Comment - Report

---

10. Future Enhancements

- AI-based task prioritization
- Voice assistant integration
- Mobile application

---

11. Approval

This document serves as the baseline agreement between stakeholders and the development team.

End of SRS Document