

# Software Requirements Specification (SRS) for Taskifi – Task Management System

## 1. Introduction

### 1.1 Purpose

Taskifi is a task management system designed to help teams **organize, track, and manage tasks efficiently**. It allows project managers to assign tasks, team members to update statuses, and administrators to manage user roles, ensuring seamless collaboration and project execution.

### 1.2 Scope

Taskifi will provide the following core functionalities:

- ✓ Task creation, assignment, and tracking
- ✓ Role-based access for Admins, Managers, and Team Members
- ✓ Task prioritization and deadline management
- ✓ Real-time notifications and reminders
- ✓ Collaboration through comments and file attachments
- ✓ Dashboard with reporting and analytics
- ✓ Secure authentication and role management

### 1.3 Intended Audience & Usage

- **Project Managers** – Assign and track tasks.
- **Team Members** – View and update tasks.
- **Administrators** – Manage users and system settings.
- **Clients (if applicable)** – Monitor progress and provide feedback.

### 1.4 Assumptions & Constraints

- Taskifi must support **at least 100 concurrent users**.
  - Task pages must load **within 2 seconds**.
  - **Role-based access control (RBAC)** will be implemented for security.
  - **Cloud-based deployment** for high availability.
- 

## 2. Overall Description

### 2.1 Product Perspective

Taskifi is a **web-based** application that can integrate with tools like **Google Calendar, Slack, and email services** for notifications.

## 2.2 Product Features

- ✓ **User Authentication & Roles (Admin, Manager, Team Member)**
  - ✓ **Task Management (Create, Assign, Edit, Delete)**
  - ✓ **Task Prioritization & Deadline Tracking**
  - ✓ **Task Status Updates (To-Do, In Progress, Blocked, Completed)**
  - ✓ **Team Collaboration (Comments, Attachments, Mentions)**
  - ✓ **Notification System (Email, In-App, Push Notifications)**
  - ✓ **Dashboard & Reporting (Progress Tracking, Charts, Filters)**
  - ✓ **File Attachments & Documentation**
- 

## 3. Functional Requirements

### 3.1 User Authentication & Role Management

- Users must **register and log in securely**.
- Three roles: **Admin, Manager, Team Member**.
- Admins can create and manage users.

### 3.2 Task Management

- Managers can **create, assign, edit, and delete tasks**.
- Users can view their assigned tasks.
- Tasks must have **title, description, priority, deadline, and assignee**.
- Users can update task status (**To-Do** → **In Progress** → **Blocked** → **Completed**).
- Comments and file attachments can be added to tasks.

### 3.3 Notifications

- Taskifi will send **email and in-app notifications** for:
  - New task assignments.
  - Task deadline reminders.
  - Status updates.

### 3.4 Reporting & Dashboard

- Managers can view **task progress, completion rates, and pending tasks**.
  - Users can filter tasks by **priority, status, deadline, and project**.
- 

## 4. Non-Functional Requirements

### 4.1 Performance Requirements

- Taskifi must support **100+ concurrent users**.
- Task pages should load in **under 2 seconds**.

## 4.2 Security Requirements

- **Role-Based Access Control (RBAC)** to restrict actions.
- **Two-factor authentication (2FA)** for admin accounts.
- **Data encryption** for sensitive user data.

## 4.3 Usability Requirements

- **Responsive design** for mobile, tablet, and desktop.
- **User-friendly UI** with drag-and-drop task management.

## 4.4 Availability & Reliability

- Taskifi must have **99.9% uptime**.
  - Automatic **daily backups** for data security.
- 

# 5. Use Cases

## Use Case 1: Assigning a Task

**Actors:** Project Manager, Team Member

**Preconditions:** Manager is logged in and has permission to assign tasks.

### Steps:

1. Manager selects a project.
2. Click on "Create Task."
3. Fills in task details (title, description, priority, deadline, assignee).
4. Click "Assign."
5. The system sends a **notification** to the assigned team member.

**Postconditions:** The task appears in the **team member's dashboard**.

---

## Use Case 2: Updating a Task Status

**Actors:** Team Member

**Preconditions:** The task is assigned to the user.

### Steps:

1. The team member logs in.

2. Navigates to their task list.
3. Selects a task and updates its status.
4. The system **logs the update and notifies relevant users**.

**Postconditions:** The task status is updated in the system.

---

## Use Case 3: Viewing Reports

**Actors:** Project Manager

**Preconditions:** The manager is logged in.

**Steps:**

1. The manager accesses the reporting dashboard.
2. Filters tasks by **status, priority, or team member**.
3. Views reports in **table or chart format**.

**Postconditions:** The system displays the requested report.

---

## 6. System Architecture

### 6.1 Technology Stack

- **Frontend:** React.js, Tailwind CSS
  - **Backend:** Node.js, Express.js
  - **Database:** MongoDB
  - **Authentication:** JWT
- 

## 7. Risks & Mitigation

Risk	Impact	Mitigation
High server load	System slowdown	Optimize database queries, use caching
Security threats	Data breach	Implement encryption, RBAC, and 2FA
User adoption issues	Low engagement	Provide user training and documentation

---

## 8. Conclusion

Taskifi is a **scalable, secure, and user-friendly** task management system that meets the needs of project managers, team members, and administrators. This SRS document ensures that all requirements are clear and well-defined before development begins.