To-do List API Documentation

1. Overview or Introduction

This document provides a detailed breakdown of the requirements, design specifications, and considerations for the development of a ToDo List API. Its aim is to offer a clear understanding of the project scope and objectives.

2. Summary of Existing Functionality

The ToDo List API extends the functionalities of task management systems by introducing new endpoints and features while aligning with established task management principles. It builds upon existing features such as task creation, update, deletion, user authentication, authorization, and task categorization.

3. Requirement Details

3.1 Task Endpoints Enhancement

New Endpoint: /tasks/{id}/complete

Description: Marks a task as completed.

Parameters: id (Task ID)

Enhancement: Allow filtering tasks by priority.

3.2 User Management

New Endpoint: /users/{id}/tasks

Description: Retrieve tasks associated with a specific user.

Parameters: id (User ID)

Enhancement: Implement role-based access control for task management operations.

4. Assumptions and Prerequisites

4.1 Assumptions:

Users possess basic familiarity with REST API principles.

Existing database infrastructure is available for task storage.

4.2 Prerequisites:

Integration of robust authentication mechanisms.

5. High-Level Design

The ToDo List API workflow encompasses:

Authentication and authorization checks.

Task management operations: retrieval, creation, update, and deletion.

Introduction of enhanced endpoints for specific task operations.

Integration with user management for access control.

6. Low-Level Design

6.1 Task Completion Endpoint Flow:

Verify user authentication.

Check task ownership.

Update task status in the database.

6.2 User Task Retrieval Flow:

Authenticate user.

Retrieve user's associated tasks from the database.

7. Impact Analysis

Product Versions: No significant impact expected on existing versions.

Service Interruption: Minimal to no interruption anticipated during implementation.

Traceability Matrix: Mapping requirements to test cases and documentation.

8. Out-of-scope

GUI implementation for task management.

Modifications to backend infrastructure unrelated to task management.

Notifications

New Feature: Implement task reminder notifications (sms or whatsapp).

Can be implemented using Twilio(trial) Soft.

9. Risks and Mitigation

9.1 Risks:

Dependency on third-party notification services.

Potential regression due to legacy system integration.

Paid Services for reliable integration

9.2 Mitigation:

Implement fallback mechanisms for notification services.

Rigorous testing and continuous monitoring for regression issues.

10. Appendices

Additional information:

Database schema for tasks and users.

API endpoint documentation with request/response examples.