# Task Priority System - Design and Development Document

## 1. Requirements Document

Feature: Task Priority System

Purpose:

To automatically suggest task priorities for a task management application based on deadlines, dependencies, and user preferences. The system helps teams focus on tasks that matter most, ensuring deadlines and dependencies are respected.

Functional Requirements:

- Input: Task name, description, assigned user role (e.g., developer, tester), deadline, task dependencies

- Output: Suggested priority (High, Medium, Low) and an explanation

Non-Functional Requirements:

- Fast computation (within 500ms)

- Scalability to handle large task lists

- Integration with task platforms like Jira, Trello

## 2. System Design Diagram

Component Layout:

[Task Input System (Form)]   
 ↓  
[Priority Engine (Java Application)]  
 ├─> Deadline Analyzer (Java Class)  
 ├─> Dependency Analyzer (Java Class)  
 └─> User Preference Analyzer (Java Class)  
 ↓  
[Priority Output (High/Medium/Low + Explanation)]

## 3. Java Code Implementation

Java Class: TaskPrioritySystem.java

public class TaskPrioritySystem {  
 public static String calculatePriority(LocalDate deadline, List<String> dependencies, String userRole) {  
 int urgencyScore = getUrgencyScore(deadline);  
 int dependencyScore = dependencies.size();  
 int userPreferenceScore = getUserPreferenceScore(userRole);  
 int totalScore = urgencyScore + dependencyScore + userPreferenceScore;  
  
 if (totalScore >= 8) return "High";  
 else if (totalScore >= 5) return "Medium";  
 else return "Low";  
 }  
 private static int getUrgencyScore(LocalDate deadline) {  
 long daysLeft = ChronoUnit.DAYS.between(LocalDate.now(), deadline);  
 return daysLeft <= 1 ? 5 : daysLeft <= 3 ? 3 : 1;  
 }  
 private static int getUserPreferenceScore(String role) {  
 return switch (role) {  
 case "Developer" -> 2;  
 case "Tester" -> 1;  
 case "PM" -> 3;  
 default -> 0;  
 };  
 }  
}

## 4. Test Cases

Test Case Table:

| Test Case | Input | Expected Output |  
|-----------|-------|-----------------|  
| 1 | Deadline = 1 day, no dependencies, role = Developer | High |  
| 2 | Deadline = 2 days, 2 dependencies, role = Tester | Medium |  
| 3 | Deadline = 10 days, no dependencies, role = PM | Low |

## 5. User Documentation

Feature: Task Priority System

Purpose: Suggest task priorities based on urgency, task dependencies, and role preferences.

How to Use:

- Fill in task details (name, deadline, role, dependencies)

- System outputs a priority level and reasoning

## 6. Deployment Considerations

- Deploy as a Java microservice

- REST endpoint: /api/priority

- Dockerize for containerized deployment

- Deploy to Kubernetes for scalability

- Secure with authentication and authorization