**TaskManagerPlus**

**Requirement:**

To develop a project adhering to the principles of Domain-Driven Design (DDD) and in accordance with industry best practices, SOLID principles, and utilizing the latest domain technologies, it is required to create an application for efficient management of user tasks. The user interface should be developed using Angular. The aim is to architect this project in a manner that it attains the status of a significant real-time project within the organization.

The application's functionalities should encompass the ability to perform the following actions:

* Create, delete, and modify tasks.
* Provide a comprehensive list of tasks for today and upcoming days, logically grouped by their respective dates and times.

Each task must encapsulate the following essential information:

* Associated User
* Date
* Start and End Times
* Subject
* Detailed Description

# Design Document: Task Management Application

## Project Overview

The Task Management Application is a web-based application designed to efficiently manage user tasks, adhering to the principles of Domain-Driven Design (DDD), industry best practices, SOLID principles, and utilizing the latest domain technologies. The user interface will be developed using Angular, and the project aims to attain the status of a significant real-time project within the organization.

## Functional Requirements

The application will have the following key functionalities:

1. \*\*Task Management:\*\*

- Create tasks

- Delete tasks

- Modify tasks

2. \*\*Task List:\*\*

- Provide a comprehensive list of tasks for today and upcoming days.

- Tasks will be logically grouped by their respective dates and times.

## Task Entity

Each task will be represented by the following attributes:

- Associated User

- Date

- Start Time

- End Time

- Subject

- Detailed Description

## Architecture and Technologies

### 1. Domain-Driven Design (DDD)

The project will be structured according to DDD principles, separating the application into distinct layers:

- \*\*Domain Layer:\*\* Contains domain models, entities, and value objects.

- \*\*Application Layer:\*\* Contains application services that coordinate actions across multiple domain objects.

- \*\*Infrastructure Layer:\*\* Manages external concerns like databases, file systems, etc.

### 2. SOLID Principles

The project will adhere to SOLID principles to ensure maintainability, extensibility, and readability of the codebase.

- \*\*Single Responsibility Principle (SRP)\*\*

- \*\*Open-Closed Principle (OCP)\*\*

- \*\*Liskov Substitution Principle (LSP)\*\*

- \*\*Interface Segregation Principle (ISP)\*\*

- \*\*Dependency Inversion Principle (DIP)\*\*

### 3. Angular UI

The user interface will be developed using Angular, providing a responsive and intuitive interface for users to interact with the application.

## Database

The application will utilize a PostgreSQL database to persist task data. Entity Framework will be used for object-relational mapping (ORM) to interact with the database.

## Technology Stack

- \*\*Frontend:\*\*

- Angular (Latest Version)

- HTML/CSS/TypeScript

- RxJS for reactive programming

- \*\*Backend:\*\*

- ASP.NET Core (Latest Version)

- C# for server-side logic

- Entity Framework Core for ORM

- RESTful API for communication between frontend and backend

- \*\*Database:\*\*

- PostgreSQL

## Testing

Unit tests will be written using xUnit for critical components, including domain logic, application services, and infrastructure components.

## Deployment

The application will be deployed on Amazon Web Services (AWS) for scalability and reliability. AWS services such as Elastic Beanstalk and RDS will be utilized for hosting the application and managing the database.

## Version Control

Git will be used for version control, with the repository hosted on GitHub for collaborative development and version tracking.

## Conclusion

The Task Management Application aims to provide an efficient and intuitive solution for managing user tasks, incorporating DDD, industry best practices, SOLID principles, and the latest domain technologies. The utilization of Angular for the UI ensures a modern and responsive user experience. By following this design, the project is poised to become a major real-time project within the organization.

---

\*Note: This design document serves as a high-level overview. Detailed design decisions and implementation specifics may evolve during the development process.\*