Timesheet Application Documentation:

Table of Contents

1. Introduction
2. System Requirements
3. High-Level Design (HLD)
4. Low-Level Design (LLD)
5. System Architecture
6. Class Diagram
7. Conclusion

1. Introduction

The **Timesheet Application** is a system designed to track and manage employees' working hours, project assignments. It allows employees to log their hours, managers to review them, and administrators to oversee the system.

2. System Requirements

2.1 Functional Requirements

- Employees can log their work hours (daily/weekly/Monthly).
- Admin can manage users, projects, and access rights.

2.2 Non-Functional Requirements

- Scalability: Support a large number of users concurrently.
- Performance: Response time should be less than 3 seconds.
- Security: Data encryption, role-based access control, authGuard.
- Reliability: Ensure high availability with backups.

2.3 Technology Stack

- **Frontend**: Angular/Tailwind css
- **Backend**: Node.js with Express.js
- **Database**: MongoDB
- **Email Service**: Nodemailer for notifications

3. High-Level Design (HLD)

3.1 System Modules

The Timesheet Application includes the following key modules:

1. **Authentication Module**
    - User login/logout and session management.
    - Role-based access (Employee, Admin).
2. **Timelog Management Module**
    - Employees can log working hours for projects.
3. **Project Management Module**
    - Admin can create, update, and manage projects.
    - Assign projects to employees.
4. **User Management Module**
    - Admin can create and manage users.
    - Assign roles and permissions

4. Low-Level Design (LLD)

4.1 Database Schema

1)User Model

Field	Data Type	Constraints
_id	ObjectId	Primary key
name	String	Required
email	String	Required
password	String	Required
department	String	Required
role	String	Required
businessUnit	String	Required

timestamp		true

2)Project Model

Field	Data Type	Constraints
_id	ObjectId	Primary key
name	String	Required
client	String	Required
address	String	Required
department	String	Required
businessUnit	String	Required
type	String	Required
users	String	Ref:User

3)Task Model

Field	Data Type	Constraints
_id	ObjectId	Primary key
name	String	Required
description	String	Optional
plannedHours	Number	Required
status	String	Required
project	String	Ref:Project
timestamps		true

4)Timelog Model

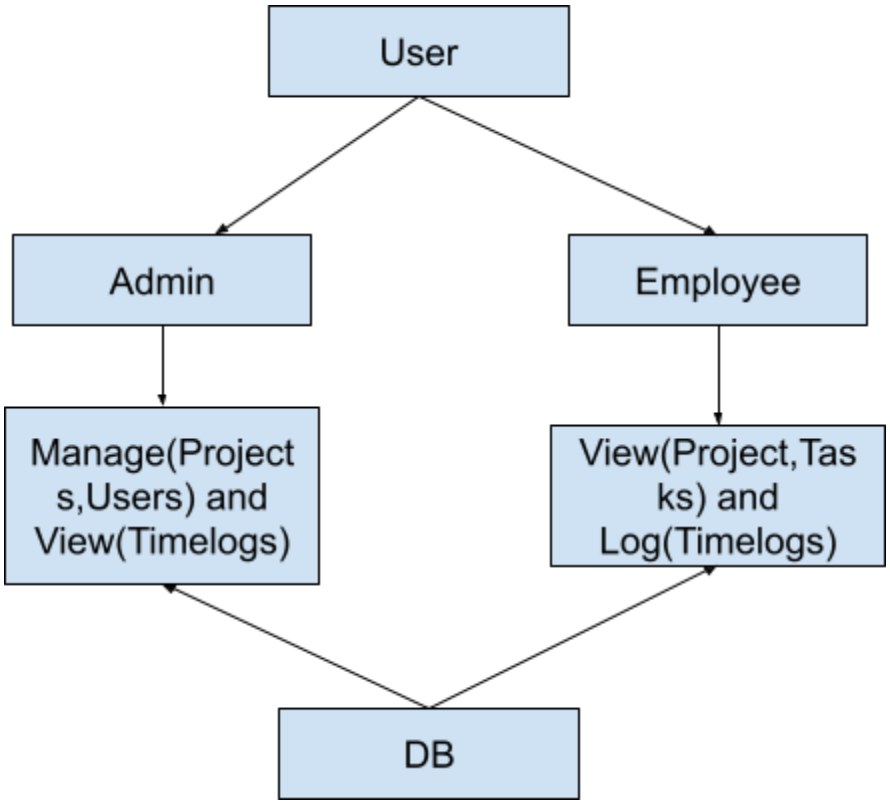
Field	Data Type	Constraints
_id	ObjectId	Primary key
date	Date	Required
task	String	Ref:Task
hours	Number	Required
status	String	Required

5)Dashboard Model

Field	Data Type	Constraints
_id	ObjectId	Primary key
name	String	Ref:Project
status	String	Required
plannedHours	Number	Required
actualHours	Number	Required

5. System Architecture

5.1 System Architecture Diagram

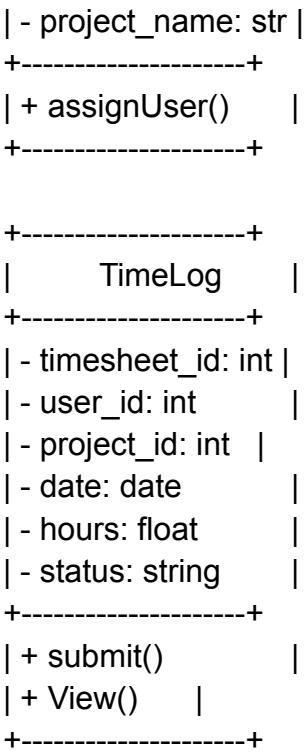


6. Class Diagram

Class Structure

```
+---------------------+
|        User         |
+---------------------+
| - user_id: int      |
| - name: string      |
| - email: string     |
| - role: string      |
| - password: string  |
+---------------------+
| + login()           |
+---------------------+

+---------------------+
|        Project      |
+---------------------+
| - project_id: int   |
```



Relationships

- **User**: Has many timesheets.
- **Project**: Assigned to a user and linked to timesheets.
- **Timelog**: Tracks the work logged by a user for a project.

7. Conclusion

The Timesheet Application is a robust system for managing and tracking employees' working hours. It ensures smooth communication between employees, managers, and administrators while offering Timelogs and project management features. With a modular design and secure architecture, the system is scalable and user-friendly.

By: Senbhagamoorthi R(23PCA132)