# Software Requirement Specification for Essential Staff Work Status Portal

**NAME :** PALANI BHARATHI T P

**ROLL NO :** 7376231CD501

**SEAT NO :** 218

**PROJECT ID :** 18

**PROJECT TITLE :** ESSENTIAL STAFF WORK STATUS PORTAL

## 1.0.INTRODUCTION:

### 1.1.Purpose:

The purpose of the Essential Staff Work Status Portal is to develop a comprehensive solution for monitoring and managing civil workers within the college campus environment. This project aims to address the challenges faced by supervisors and administrators in efficiently organizing and overseeing civil work activities, ensuring optimal utilization of resources, and maintaining a safe and productive work environment.Through the provision of real-time insights and task assignment and monitoring tools, the web application aims to improve decision-making capacities and streamline workflow procedures for supervisors and administrators.

### 1.2.Objectives:

- To provide supervisors with a user-friendly platform for scheduling and assigning tasks to civil workers for each workday.
- To enable administrators to gain insights into workforce distribution, task assignments, and work progress through a centralized dashboard.
- To streamline communication and coordination among supervisors, administrators, and civil workers, thereby improving overall productivity and efficiency.
- To enhance transparency and accountability in the management of civil work activities within the college campus.

## 2.0.PROJECT OVERVIEW:

### 2.1.Stakeholders:

- **College administration:** The college administration oversees the overall management and operations of the campus, making strategic decisions regarding infrastructure maintenance and development.

- **Supervisors:** Supervisors are responsible for overseeing the daily activities of civil workers, including task scheduling, assignment, and monitoring.

- **Administrators:** Administrators play a crucial role in overseeing the entire workforce and gaining insights into workforce distribution and task assignments.

## 2.2.Scope:

- User authentication and role-based access control for supervisors and administrators.
- Supervisor dashboard for scheduling and assigning tasks to civil workers.
- Admin dashboard for monitoring workforce distribution and task assignments.
- Database management system for storing and retrieving task-related data.
- Integration with existing college systems for seamless data exchange and communication.

## 2.3.Significance:

- Improve the efficiency and effectiveness of civil work management within the college campus.
- Enhance collaboration and communication among supervisors, administrators, and civil and electrical workers.
- Optimize resource utilization and allocation, leading to cost savings and improved operational efficiency.

- Foster a culture of transparency, accountability, and data-driven decision-making within the college administration.

## 3.0.USER REQUIREMENTS:

### 3.1.Supervisor:

#### **3.1.1.**Role Definition:

Supervisors are responsible for managing civil workers, scheduling their tasks, and ensuring that the daily work is completed efficiently.

#### **3.1.2.Responsibilities:**

- Schedule and assign tasks to workers for upcoming workdays.
- Monitor and update the status of tasks (completed or pending).
- Enter detailed task information such as date, time, nature of work, location, and worker assignments
- Conveying the feedback and the grievances of the workers to the higher authorities.

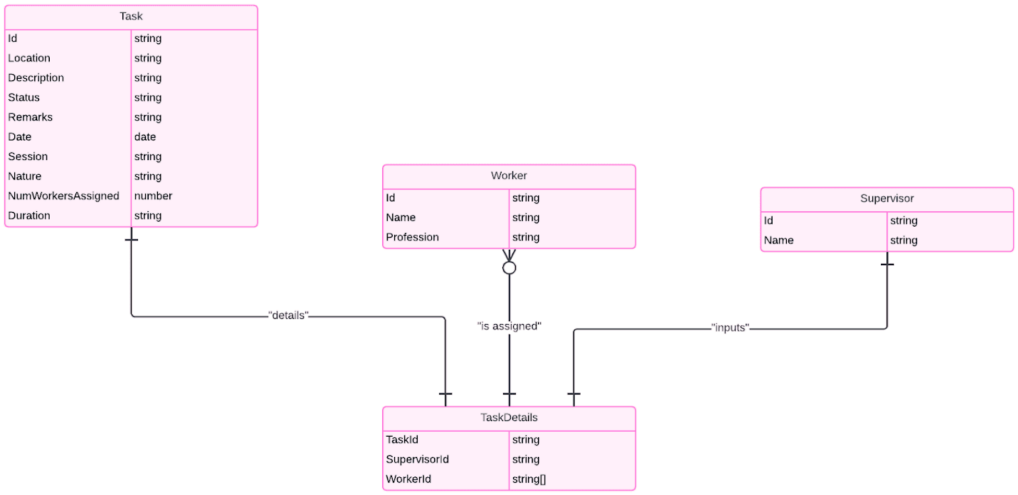



**Figure: 1. - Supervisor's Entity.**

## 3.2.Admin:

### **3.2.1.**Role Definition:

Admins oversee the overall operation within the campus, providing them with a broad view of activities and resource allocation.

### 3.2.2.Responsibilities:

- Access a dashboard summarizing all ongoing and completed tasks.
- View reports and analytics on worker distribution and task status.
- Search and retrieve specific data points such as worker locations, assigned tasks, and status updates.

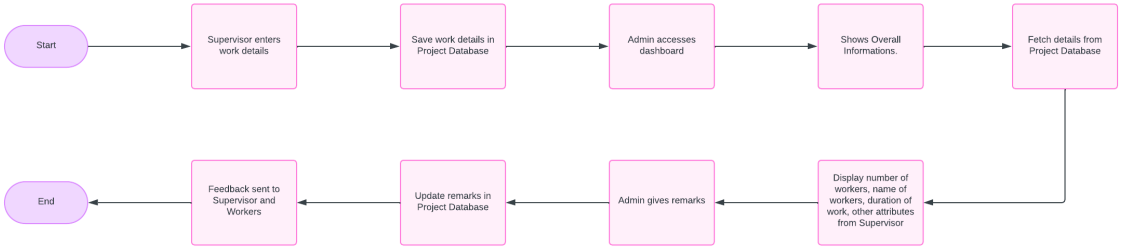



**Figure: 2 - Admin's Entity.**

## 3.3.Key Features and Functionalities:

### 3.3.1.Supervisor  Features :

- **Task Scheduling Interface:** A user-friendly interface to input and modify the schedule of tasks.

- **Task Management Dashboard:** A dashboard displaying all scheduled tasks and their statuses, allowing for updates and edits in real time.

- **Worker Assignment Tool:** A system to assign workers to specific tasks, including capabilities to adjust

assignments as needed based on worker availability and task demands.

### 3.3.2.Admin Features:

- **Overview Dashboard:** A high-level dashboard that provides an overview of all campus activities, focusing on worker distributions and task completions.

- **Advanced Search Capabilities:** Tools to search for tasks by ID, location, or worker names, providing detailed insights into the workforce's operations.

- **Analytics and Reporting Tools:** Generate reports on various metrics such as task completion rates, worker efficiency, and resource utilization to aid in decision-making.

- **Customization Settings:** Provide admins with the ability to customize various aspects of the dashboard and reports, such as setting preferences for data display, choosing specific metrics to prioritize, and configuring notification settings.

# 4.0.SYSTEM ARCHITECTURE:

## 4.1. Overview

The system architecture of our web application is designed to support robust, scalable, and secure operations tailored for effective management and monitoring of civil workers on a college campus. The architecture is divided into two main components: the front-end for user interactions and the back-end for processing and data management, all connected through secure APIs.

## 4.2.Frontend Architecture:

### 4.2.1.Technology Stack:

The front-end of our web application is built using **REACT .JS.** This choice was driven by React's efficiency in updating and rendering components, which is essential for our dynamic and interactive UI needs.

### 4.2.2.User Interfaces:

**Supervisor Interface:** Allows supervisors to enter and manage tasks for civil workers. Features include task creation forms, scheduling tools, and a dashboard for task tracking.

**Admin Interface:** Provides admins with a dashboard that offers a high-level view of activities, including analytics and worker task distributions.

### 4.3.Backend Architecture:

**Technology Stack:** The backend is implemented using **Spring Boot**, chosen for its capability to rapidly develop enterprise-grade applications. Spring Boot facilitates easy configuration and is extensively supported by a wide range of libraries and tools.

**Database: MySQL** is used for data persistence, handling data storage for tasks, worker profiles, and user information. This relational database management system was selected for its reliability, performance, and support for complex queries.

**APIs: RESTful APIs** are developed to facilitate communication between the frontend and backend. These APIs are secured with authentication and authorization mechanisms to ensure that only legitimate users can access sensitive data.

### 4.4.Communication Protocols and Data Flow:

- **HTTP/HTTPS Protocols:** The application uses HTTP and HTTPS protocols for communication between the client and server. HTTPS ensures that data transferred remains encrypted and secure.

- **DataFlow:**

- **From Frontend to Backend:** When a supervisor enters or updates a task, the frontend sends a POST or PUT request via HTTPS to the backend. The backend then processes these requests, updates the database, and sends back a response.
- **From Backend to Frontend:** The backend periodically sends data updates to the frontend, such as task statuses or alerts, using WebSocket for real-time communication to ensure that the UI is always current.
- **Admin Dashboard Data Retrieval:** Admins accessing the dashboard trigger GET requests that fetch aggregated data and analytics from the backend, ensuring that the dashboard displays the most updated information possible.

## 5.0.USER INTERFACE DESIGN:

### 5.1.Overview:

The user interface (UI) design for our web application is focused on creating an intuitive and efficient experience for both supervisors and admins. Our design principles emphasize clarity, ease of use, and responsiveness to accommodate various devices.

### 5.2.Login Page:

The login page features a simple and secure layout. It includes fields for username and password, accompanied by login buttons specific to either supervisors or admins to streamline access based on user roles.

### 5.3.Supervisor Dashboard:

**Mockup Details:** The Supervisor Dashboard is designed to provide a comprehensive view of daily tasks and worker statuses. It includes a calendar for scheduling, a list view for current and pending tasks, and easy access buttons for task modifications.

**Design Considerations:** The dashboard is designed for ease of use, with drag-and-drop features for rescheduling tasks and color-coded statuses for quick visual reference. Information is displayed hierarchically, ensuring that the most urgent tasks catch the supervisor's attention first.

### 5.4.Admin Dashboard:

**Mockup Details:** The Admin Dashboard offers a high-level overview of operations across the campus. It features widgets for quick stats, a map for real-time worker location tracking, and filters for detailed searches by task ID, worker name, or location.

**Design Considerations:** Emphasis is placed on real-time data updates, which are crucial for admins managing resources at a macro level. The interface includes customizable settings, allowing admins to tailor the dashboard to their preferences for displaying certain data types prominently.

## 5.5.LAYOUT OF SUPERVISOR'S VIEW PAGE:

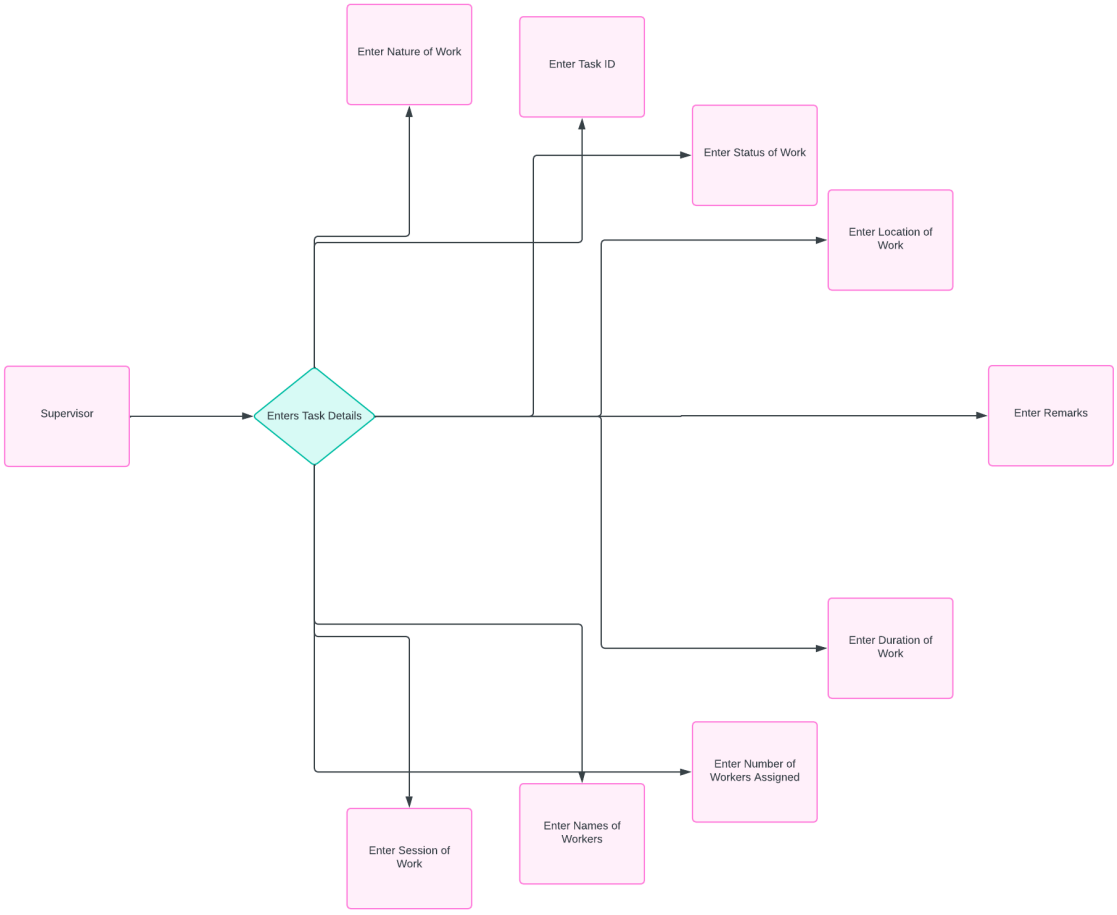



**Figure : 3 - Layout of Supervisor's ViewPage**

## 5.6.LAYOUT OF ADMIN'S VIEW PAGE:

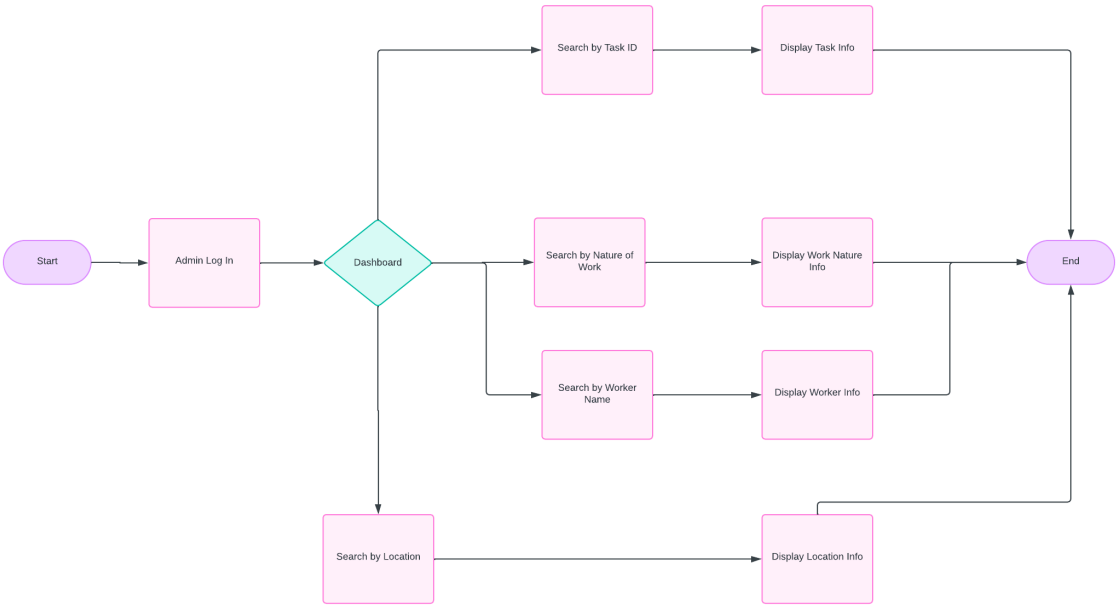



**Figure : 4 - Layout of Admin's View Page**

## 6.0.DATABASE DESIGN:

### 6.1.Overview:

The database design serves as the foundation of our web application, providing a structured framework for storing and organizing essential data related to tasks, workers, and user authentication. By carefully designing the database schema, we ensure efficient data management, maintain data integrity, and support scalability as our application grows.

## 6.2.Entity-Relationship Diagram:

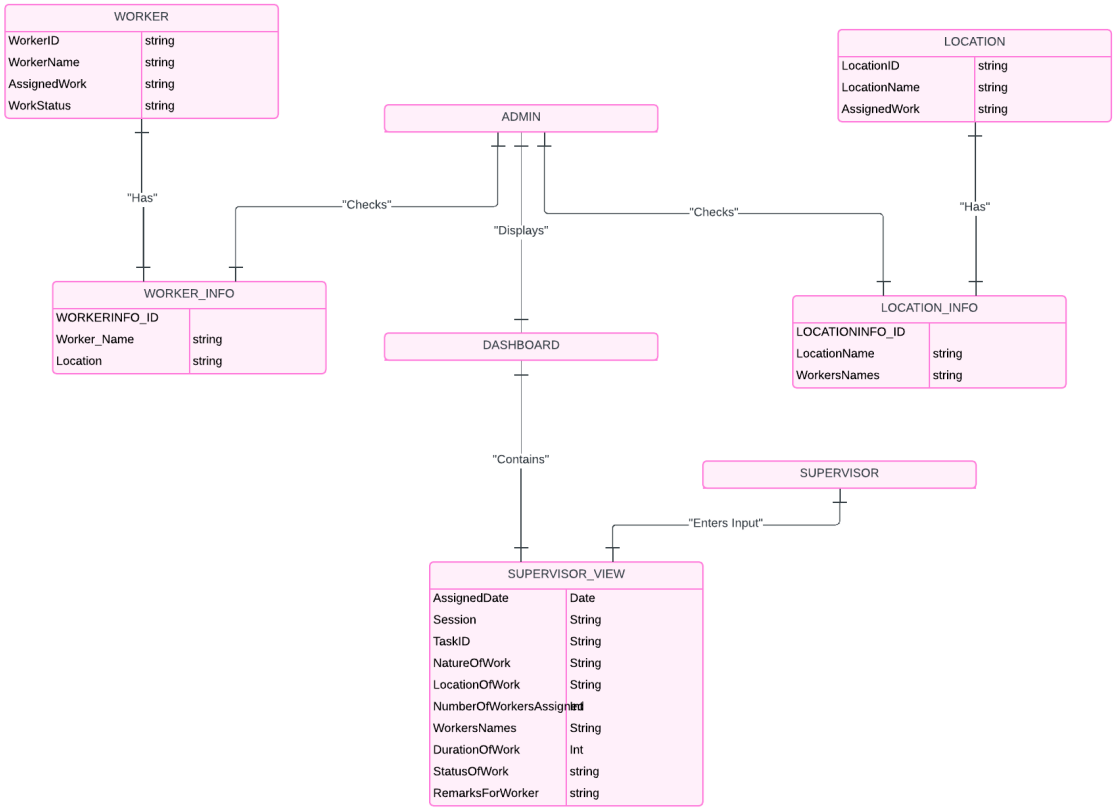
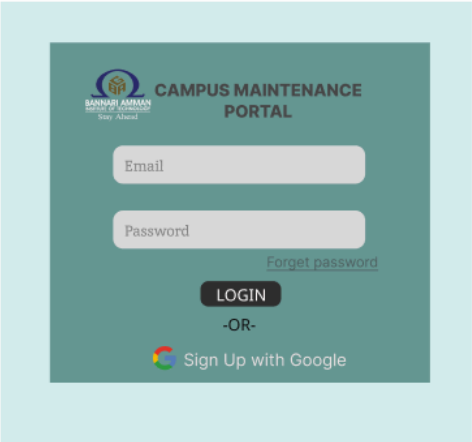



**Figure: 5 - Entity-Relationship Diagram.**

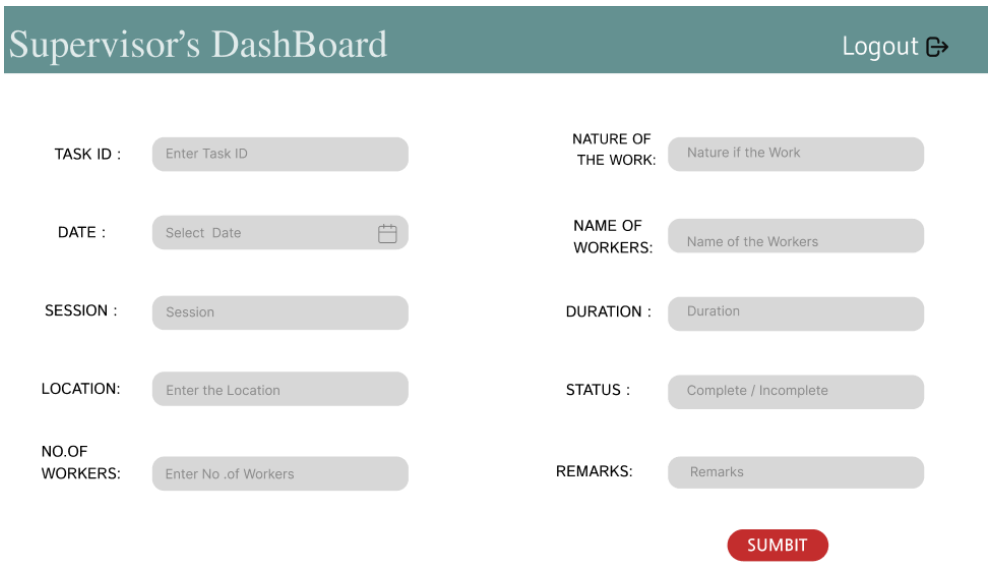
The Entity-Relationship Diagram (ERD) visually represents the structure of our database, illustrating the relationships between different entities and their attributes. This diagram serves as a valuable reference for understanding the database schema at a glance and guides the development of database tables and relationships.

## 7.0.LAYOUT :

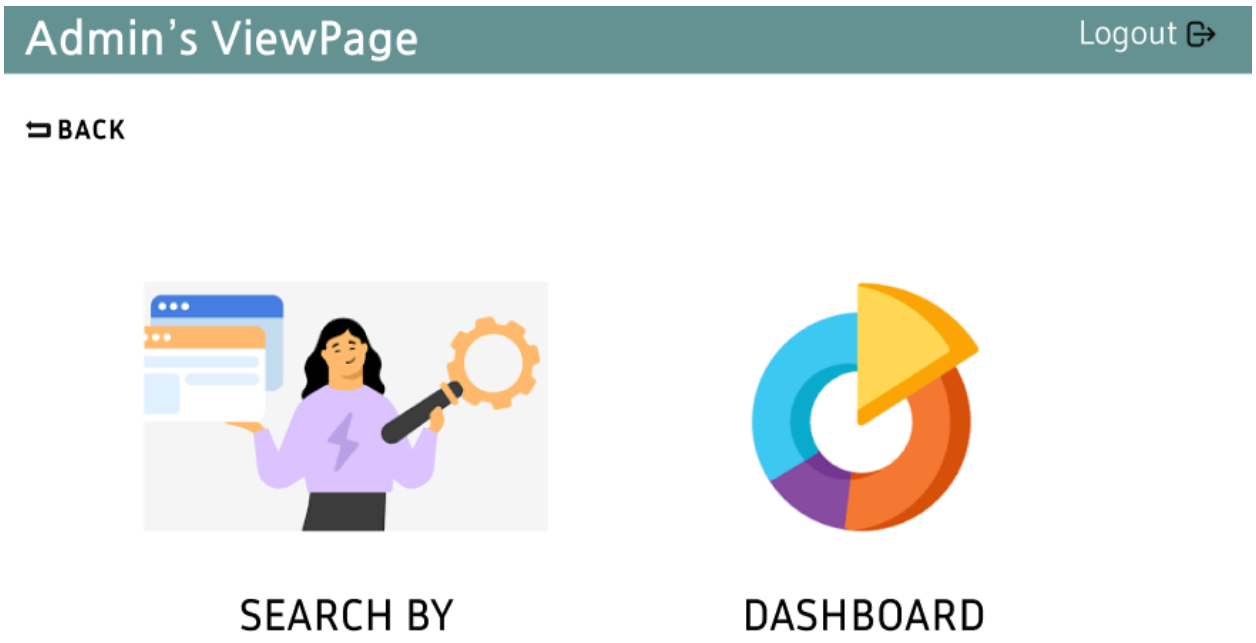
### 7.1.LOGIN PAGE :




### 7.2.SUPERVISOR'S DASHBOARD:



## 7.3.ADMIN'S VIEW PAGE:




## 7.4.ADMIN'S SEARCH PAGE :

## 7.5.ADMIN DASHBOARD :