****

**Personnel Management System**

**Software Requirement Specification (SRS) Document**

**Sprint Implementation-2**

**Project Timeline:07.09.2022 – 23.09.2022**

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Page no.** |
| 1 | Introduction | 3 |
| 1.1 | Purpose | 3 |
| 1.2 | Intended audience | 3 |
| 1.3 | Intended use | 3 |
| 1.4 | Project scope | 3 |
| 2 | Overall Description | 4 |
| 2.1 | User needs | 4 |
| 2.2 | Assumptions and Dependency | 4 |
| 3 | System features and requirements | 4 |
| 3.1 | Functional Requirements | 4 |
| 3.1.1 | Employee Details | 4 |
| 3.1.2 | Project Details | 5 |
| 3.2 | Non-functional Requirements | 6 |
| 3.3 | System requirements | 7 |
| 3.3.1 | Tools to be used | 7 |
| 4 | Data Flow Diagrams | 7 |
| 4.1 | DFD Level 0 (CAD) | 7 |
| 4.2 | DFD Level 1 | 7 |

**Introduction**

The introduction of the software requirement specification provides an overview of the   entire software. The entire SRS with overview description purpose, scope, tools used and basic description. The aim of this document is to gather, analyse and give an in-depth insight into the complete Personnel management application by providing the user required information through the application. The detailed requirements of the Personnel management application are provided in this document.

**1.1 Purpose**

The purpose of this document is to describe the requirements to track all information related to Employee(add employee, edit employee, view employee and delete employee (if needed)), Project details on which employee is working (add project, edit project, delete project(if needed),query an employee, giving the pay stub according to the working hours of the employee on the particular project, employee should be able to view the number of hours he spent on a particular project and the application should also provide the report for the employees who are working on more than one project.

The main purpose of the Personnel management system is to organize the employee data and make it accessible in an efficient way.

**1.2 Intended Audience**

This document is intended to be read by Client.

**1.3 Intended use**

* Development Team
* Maintenance Team
* Clients

**1.4 Project scope**

The project aims at the development of the online personnel management system   to assist Marspro organization to keep the records of the employee along with the project they are working on and to calculate their pay stub in a more robust and efficient manner. Along with that it also provides the list of employees who are working on more than one project and able to give the details of working hours on a project for an employee.

1. **Overall Description**

**2.1 User needs**

Any employee who manages data from the Marspro organization should be able to add, edit, view and delete the employee details as well as project details according to the requirements. To perform this functionality a menu is provided where the options are given for adding, editing, displaying, deleting records from employee and project list.

Along with that it also provides the option as query employee to view information of a single employee based on employee code and to calculate the pay stub according to the company policies which is stated as below:

The company pays Rs 45.00 per hour worked and a further Rs 10.00 per dependent per hour. 2 The company deducts 8.5% EPF from gross wage. 3 The company deducts 15% income tax from the gross wage.

Not only this, if a user wants to view how many hours an employee spent on a particular project, they can also retrieve that information and it also provides the list of employees working on more than one project at the moment in the form of a report.

**2.2 Assumptions and Dependency**

* User has the latest version of Ubuntu Linux installed.
* Client has either 4GB or more RAM.
* The service is used preferably on a desktop or laptop.

1. **System Features and Requirements**

**3.1 Functional Requirement**

**3.1.1Employee Details**

**3.1.1.1 PMS\_01 => add\_employee ()**

This feature adds the employee record to the file. It asks for employee\_id, employee\_name, department, salary, contact number whose record is to be created.

**3.1.1.2 PMS\_02 => edit\_employee ()**

This feature edits the employee record in the file. It asks for the employee\_id and asks the user to enter new employee\_name, department , salary, contact number and the list of projects done by the new employee by adding project\_id, project\_name, number of hours worked . Whose record is to be edited Upon successful editing of employee details into the employee database it will return to the main menu.

**3.1.1.3 PMS\_03 =>view\_employee ()**

This feature displays all the employee details present in the employee database. It displays employee\_id, employee\_name, department, contact number and list of projects done by the employee with project\_id, project name, number of hours worked.

**3.1.1.4 PMS\_04 => delete\_employee ()**

This feature deletes the employee details from the employee database. The user needs to provide the employee\_id to be deleted from the employee database. Upon successful deletion of employee details from the employee database it will return to the main menu.

**3.1.2 Project Details**

**3.1.2.1 PMS\_05 => add\_project()**

This feature adds the new project details into the project database. It asks for project\_id, project name and number hours to be worked on that project. Upon successful addition of a new project details into the project database then it returns to the main menu.

**3.1.2.2 PMS\_06 => edit\_project ()**

This feature edits the project's details from the project database. It asks for project\_id and asks the user to enter a new project name and number of hours to be extended to the project. Upon successful editing of project details into the project database then it returns to the main menu.

**3.1.2.3 PMS\_07 => view\_project ()**

This feature displays the project details in the project database. It displays project\_id, project name, number of hours to be worked on that project.

**3.1.2.4 PMS\_08 => delete\_project ()**

This feature deletes the project details from the project database. The user needs to provide the project\_id to be deleted from the project database. Upon successful deletion of project details from the employee database it will return to the main menu.

**3.1.2.5 PMS\_09 => query()**

This feature presents a query on top projects should show the project on which the highest number of employees are working. If the number is more than one, then all those projects should be displayed.

**3.1.2.6 PMS\_10 => pay\_report ()**

This function calls **salary\_view()** member function of SALARY class which displays the pay report for all the employees (id, name, basic salary, bonus, deduction and final salary) according to the criteria set by the Marspro organization.

**3.1.2.7 PMS\_10 => calc\_fi\_salary ()**

This feature generates the pay report for a particular employee (id, name, basic salary, bonus, deduction and final salary) according to the criteria set by the Marspro organization

**3.1.2.8 PMS\_11 => emp\_project report()**

This feature allows an employee to view how many hours he has already spent on which project through a menu option. This option requires him to enter his employee\_id, which is validated, and the details are shown.

**3.1.2.9 PMS\_12 => emp\_list\_with\_3\_projects ()**

This feature shows the details of all employees who are working on 3 projects at time.

**3.2 Non-functional Requirements**

•     **Supportability**: The system is easy to maintain.

•   **Design Constraints**: The system is built using only CPP language.

•   **Usability**: Personnel management system helps to manage the employee records and also displays the information according to the requirement of the moment. For example, if we want to calculate the pay of an employee, the system provides that functionality. Next time if you want to query an employee it also works like that. This function also adds to the usability of the system.

•   **Reliability & Availability**: The system shall provide employees along with project information with the creation of a file in every machine.

•   **Performance:** The system will work on the user’s terminal.

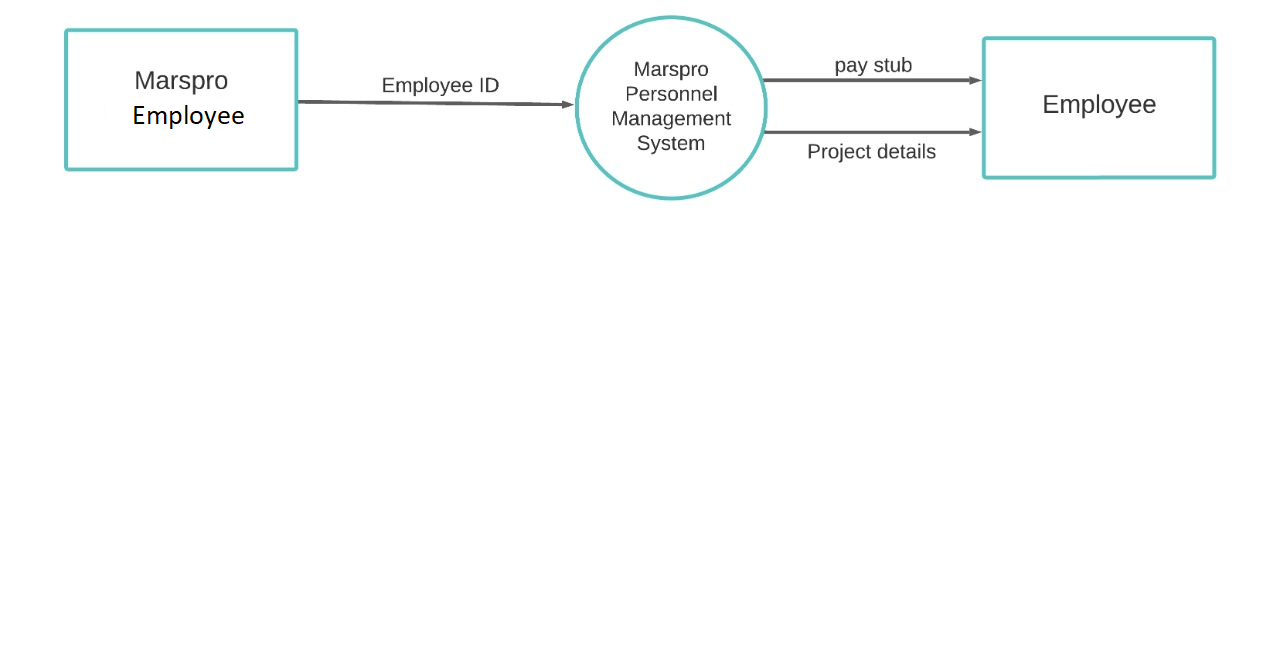
**3.3 System Requirements**

**3.3.1. Tools to be used**

* CPP File Handling
* CPP Language
* vi editor
* System Programming
* Valgrind
* GDB
* Splint
* Gcov
* CPP check
* CPPUnit
* Gprof

**4. Data Flow Diagram:**

**4.1 DFD Level 0:**

****

**4.2 DFD level 1:**

