**SOFTWARE PERSONNEL MANAGEMENT SYSTEM**

PROBLEM STATEMENT:

Software personnel management system allows employees to record time card electronically and automatically generates pay slips based on number of hours worked and total amount of sales.

The system will run on individual employee desktops where the employee can access and edit only their personal details. The system will maintain information of the employee in the company in order to calculate the payroll.

The employees will also be able to know from the system, the number of hours worked per day and total of all hours spent on a project and total pay received year-to-date etc.

Payroll administrators keep track of all the information including adding new employees, deleting employees, and edit information and run reports. The system will generate records and performance report of the employees.

Software system that will allow the human resource department to manage its employee in a better way.When needed,it will just take a few seconds to find out the background of an employee and his/her contribution to the organization,it will also facilitate keeping all the records of the employee,such as data leaving.So,all the information about an employee will be available in few seconds.It will also make it very easy to generate statistical data or custom data.Overall it will make human resource management an easier job.

**TABLE OF CONTENTS FOR SOFTWARE REQUIREMENTS SPECIFICATION:**

|  |  |
| --- | --- |
| CONTENTS | PAGE NO |
| Introduction | 5 |
| Purpose | 8 |
| Project Scope | 9 |
| References | 10 |
| Product Perspective | 10 |
| Tools to be used | 10 |
| Definitions | 11 |
| User Characteristics | 12 |
| Functional Requirements | 12 |
| Non Functional Requirements | 14 |
| Hardware Reqs | 16 |

SOFTWARE REQUIREMENT SPECIFICATION:

INTRODUCTION:

The Software Personnel Management system is an interface between Employee and the Administrator responsible for generation of payment slip. It aims at improving the efficiency in the generation of Pay slip and reduces the complexities involved in it to the maximum possible extent.

Software Personnel Management System allows human resource department to manage its employee in a better way.A software which contains all the employee details in a database manager is developed.Whenever an employee ID is entered,all the details about the employee are retrieved from the database manager.Then,according to the number of working days,number of working hours,a pay slip is generated.Thus,by the end of the month,the salary of the employee can automatically be calculated by the administrator from the instances of the database manager.This is a very convinient and efficient way of generating payslips to the employee.This procedure can be used when there are large number of employees.It is a very fast and efficient process.

PURPOSE:

The main reason for the development of this project is that it requires a lot of manpower to calculate the payslip of each and every employee in large industries.It is also a time consuming process and it may lead to errors as well.To overcome this problem,a software called “Software Personnel Management System” is developed.This uses a database manager.The administrator maintains a record of all the employees in a database manager.This contains the employee login,his personal details,number of working hours,number of working days and other details.Thus by the end of the month,the payslip is automatically generated and hence the admin can directly retrieve the data from database manager and generate the payslip of each and every employee in very less time without any errors and in a highly efficient manner.

PROJECT SCOPE:

In this project,we are going to achieve a software that can generate the payslip of each and every employee without any manual work and in a very efficient manner.

As soon as the employee ID is given,all the details about the employee are obtained from the database manager and hence it is very easy to generate a payslip without any errors. The main scope of this project would be to maintain a database for each employee which contains all his details,number of working hours,number of working days,etc and whenever required,update the database accordingly.

REFERENCES:

IEEE Software Requirement Specification format.

PRODUCT PERSPECTIVE:

The Software Personnel Management System acts as an interface between the ‘Employee’ and the 'Administrator’. This system tries to make the interface as simple as possible and at the same time not risking the work of data stored .

TOOLS TO BE USED:

• Eclipse IDE (Integrated Development Environment)

• Rational Rose tool (for developing UML Patterns)

DEFINITIONS:

* HTML-Hyper Text Markup Language
* HTTP-Hyper Text Transfer Protocol
* TCP/IP-Transmission Control Protocol/Internet Protocol is the communication protocol used to connect hosts on internet.
* J2EE-programing platform for developing distribute java applications
* SQL-Structured Query Language

USER CHARACTERISTICS:

EMPLOYEE:

He/She works for a company and all the details of employee including his/her login,total number of working hours,total number of working days,etc are stored in a database manager and are used whenever needed.

ADMINISTRATOR:

He/She check the employee details using a database manager and generate a payslip accordingly.

DATABASE MANAGER:

This is a database where we can store the details of each and every employee and hence we can easily retreive the information of each and every employee,whenever needed and generate a payslip accordingly.

FUNCTIONAL REQUIREMENTS:

A functional requirement describes an interaction between its environments.Functional requirements describes the system function in details, its input and output , exception,etc.The functional requirement are as follows:

Analysis:

In this phase, the project requirement is analyzed and the availability of the requirement is seen.

Design:

The design of a project is made by the project manager.

Implementation:

The construction of project is done and coding is developed.

Testing:

Testing activities are made,for which several types of testing is carried on.

Maintenance:

In this the software maintenance and the ways to avoid the drawback of the software is made.

Deployment:

IT is the process of installing and kick starting of the program.

NON-FUNCTIONAL REQUIREMENTS:

Non-functional requirements are requirements which are not directly concerned with the specific functions delivered by the system. They relate to emergent system properties such as reliability, response time and store occupancy. They relate to the system as whole other than to individual system features which means that they are more critical than individual functional requirements.

The non-functional requirement are as follows

Speed:

This software designed as a high speed software.

Size:

The size of this software is fixed.

Portability:

This software can be easily portable.

Reliability:

The rate of failure of this software is very low.

HARDWARE REQUIREMENTS:

1. Processor – Intel Pentium IV-2.0 GHZ.

2. Hardware – 40 GB.

3. RAM – 512mb

4. DVD RAM – 1 nos.

SOFTWARE REQUIREMENTS:

1. OS – windows XP/vista.

2. Front end Tool – Rational Rose Enterprise suite.

3. Back end Tool – Oracle 10i.

SOFTWARE CONFIGURATION MANAGEMENT:

Software Configuration Management is the arrangement of exercises which control the changes in the document and identifying the items for change. If something goes wrong, SCM can determine what was changed and who changed it. If a configuration is working well, SCM can determine how to replicate it across many hosts. It aims to control cost and work effort involved in making changes to the software system. The primary goal is to increase productivity with minimal mistakes.

TASKS IN SCM:

1)Configuration Identification-define exam types, categories, authorization validity, maintain security.

2) Change Control-A change request (CR) is submitted and evaluated to assess technical merit, potential side effects, overall impact on other configuration objects and system functions, and the projected cost of the change. The results of the evaluation are presented as a change report, which is used by a change control board (CCB) —a person or group who makes a final decision on the status and priority of the change.

3)Configuration Status- A software

Configuration audit complements the

Formal technical review of the process.

DESIGN DOCUMENT:

TITLE:

Design document for “Software Personnel Management System”.

INTRODUCTION:

The Software Personnel Management system is an interface between Employee and the Administrator responsible for generation of payment slip. It aims at improving the efficiency in the generation of Pay slip and reduces the complexities involved in it to the maximum possible extent.

Software Personnel Management System allows human resource department to manage its employee in a better way.A software which contains all the employee details in a database manager is developed.Whenever an employee ID is entered,all the details about the employee are retrieved from the database manager.Then,according to the number of working days,number of working hours,a pay slip is generated.Thus,by the end of the month,the salary of the employee can automatically be calculated by the administrator from the instances of the database manager.This is a very convinient and efficient way of generating payslips to the employee.This procedure can be used when there are large number of employees.It is a very fast and efficient process.

LIST OF DIAGRAMS:

1.Use case diagram

2.Class diagram

3.Sequence diagram

4.Collaboration diagram

5.Activity

6.State diagram

7.Component diagram

8.Deployment diagram

DESCRIPTION OF DIAGRAMS:

1.USE CASE DIAGRAM:

Use case diagram is the pictorial representation of how several users interact with the system.It contains actors,functionalities to represented as use case,relationships among use cases.

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases.

Use case diagrams specify the events of a system and their flows. But use case diagram never describes how they are implemented. Use case diagram can be imagined as a black box where only the input, output, and the function of the black box is known.



2.CLASS DIAGRAM:

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

A class notation consists of three parts:

1. **Class Name**
   * The name of the class appears in the first partition.
2. **Class Attributes**
   * Attributes are shown in the second partition.
   * The attribute type is shown after the colon.
   * Attributes map onto member variables (data members) in code.
3. **Class Operations** (Methods)
   * Operations are shown in the third partition. They are services the class provides.
   * The return type of a method is shown after the colon at the end of the method signature.
   * The return type of method parameters are shown after the colon following the parameter name.
   * Operations map onto class methods in code



3.SEQUENCE DIAGRAM:

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows.



4.COLLABORATION DIAGRAM:

Collaboration diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaborations are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case.

A collaboration diagram resembles aflowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. Themessages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.



5.ACTIVITY:

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

The basic purpose of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.



6.STATE DIAGRAM:

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It’s a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli. We can say that each and every class has a state but we don’t model every class using State diagrams.



7.COMPONENT DIAGRAM:

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.



8.DEPLOYMENT DIAGRAM:

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. Artifacts represent concrete elements in the physical world that are the result of a development process.

A variety of shapes make up deployment diagrams. This list offers an overview of the basic elements you may encounter, and you can see most of these items illustrated in the image below.

* **Artifact:** A product developed by the software, symbolized by a rectangle with the name and the word “artifact” enclosed by double arrows.
* **Association**: A line that indicates a message or other type of communication between nodes.
* **Component:** A rectangle with two tabs that indicates a software element.
* **Dependency:** A dashed line that ends in an arrow, which indicates that one node or component is dependent on another.
* **Interface:** A circle that indicates a contractual relationship. Those objects that realize the interface must complete some sort of obligation.
* **Node:** A hardware or software object, shown by a three-dimensional box.
* **Node as container:** A node that contains another node inside of it.
* **Stereotype:** A device contained within the node, presented at the top of the node, with the name bracketed by double arrows.

