**PROBLEM STATEMENT**  :

Software personnel management system allows to electronically record the details of the employee through the software hence making it easier to generate other information regarding the employee such as attendance, payroll, number of hours worked etc .Previously manually recording data proved to be very difficult to access data when needed and to generate other utility important information such as number of hours worked. Hence it marked a need for software maintenance system where information can be stored, manipulated to generate information about the required filed.

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 on 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 Requirement Specification** :

Purpose :

Every year the numbers of employees is increasing very rapidly. Having manually generated pay slip for all this ever increasing count is very tedious and impossible task. Hence a maintenance system is necessary to manage these needs.

Scope:

* We can get the information associated to the particular employee and his contribution to the organization in a matter of minutes
* Creation of custom data from the available data like general statistics is easier
* Information regarding an employee is organized effectively

**DEFINITIONS**

♣

Administrator:

Refers to the super user who is maintaining the employee details.

♣

Employee:

One who works for a software company

♣

SPMS :

Refers to this Software personnel management system

♣

HTML :

Markup Language used for creating web pages.

♣

J2EE :

Java 2 Enterprise Edition is a programming platform java platform for developing and running distributed java applications.

♣

HTTP :

Hyper Text Transfer Protocol.

**REFERENCES :**

IEEE Software Requirement Specification format.

**TECHNOLOGIES TO BE USED :**

• HTML

• JSP

• Javascript

• Java

• XML

• AJAX

**TOOLS TO BE USED :**

• Eclipse IDE (Integrated Development Environment)

• Rational Rose tool (for developing UML Patterns)

**PRODUCT PERSPECTIVE**

The SPMS acts as an interface between the 'ADMINISTRATOR' and the 'employee'. This system tries to make the interface as simple as possible and at the same time not risking the security of data stored in. This minimizes the time duration in which to manage the software personnel.

**SOFTWARE INTERFACE**

• Front End Client - The applicant and Administrator online interface is built using JSP and HTML. The ADMINISTRATOR's local interface is built using Java.

• Web Server – Apache Tomcat application server(Oracle Corporation).

• Back End – Oracle 11g database.

**HARDWARE INTERFACE**

The server is directly connected to the client systems. The client systems have access to the database in the server.

**FUNCTIONAL REQUIREMENTS** :

Payment Slip :

The payment module greatly reduces the workload of the ADMINISTRATOR department by automating the payroll process, allowing ADMINISTRATOR to ensure the payroll functions are completed on time and without errors.

View Salary :

The employee views the salary details efficiently from the SPMS.

**NON-FUNCTIONAL REQUIREMENTS:**

USABILITY **-** The software is easy to use without any particular manual or support.

PORTABLE **-** It can execute properly on multiple hardware platforms.

SECURE **–** Information is securely transferred to the server without any changes.

RELIABILITY **–** It assures no misuse of data and the rate of failure of this software is very low.

**USER CHARACTERISTICS**

>Employee

>Administrator

> Database manager

>Admin

**CONTRAINTS**

Administrator privilege are needed to monitor the information.

**ASSUMPTIONS AND DEPENDENECIES :**

>Users must have basic knowledge on working of computers and English language.

**Design Document**

**Title**  :

**Introduction :**

Stock manager is having the right to login and control into the software by entering his name and a valid password to tackle the problem of maintaining a

record of stack, several programming and database techniques are used to make process the easier.

**Persons Involved:**

1.Salesperson

2.Customer

3.Company

**Problem:**

The entire process of stock maintenance is done in a manual manner considering the fact that number of customers are increasing every year, an automated maintenance system is necessary.

**Expected Solution :**

An internet base automated stock Maintenance system is necessary to have the complexity of the problem availability of the product and product can be made available online this will allow the customer to buy the required product any time. A user interface should be provided to browse through the product and delivery system needs to be established.

**TABLE:**

|  |  |  |
| --- | --- | --- |
| **TASK** | **STARTING DATE** | **COMPLETION**  **DATE** |
| Developing the home page with all the desired options | Jan 11,2019 | Jan 12,2019 |
| Developing web page for registration of employees | Jan 13, 2019 | Jan 15,2019 |
| Adding additional Registration fields for the manager | Jan 16,2019 | Jan 16,2019 |
| Designing a log in page | Jan 17,2019 | Jan 17,2019 |
| Writing code for categorizing and sorting employees | Jan 18,2019 | Jan 21,2019 |
| Pictorial representation of information | Jan 22,2019 | Jan 23,2019 |
| Code for generating the pay | Jan 24,2019 | Jan 26,2019 |
| Linking code with  the webpages | Jan 27,2019 | Jan 28,2019 |
| Providing temporary dataset | Jan 29,2019 | Jan 29,2019 |
| Testing and Improvements | Jan 29,2019 | Feb 2,2019 |
| Handing over the software to the company | Feb 3,2019 | - |
| Updating | Whenever required | - |

**TOOLS USED:**

* Rational rose

**LIST OF DIAGRAMS:**

1.Use case diagram

2.Class diagram

3.Sequence diagram

4.Collaboration diagram

5.Activity

6.Component diagram

7.Deployment diagram

**1)USECASE DIAGRAM:**

* In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.
* An effective use case diagram represents

scenarios in which your system or application interacts with people, organizations, or external systems.

* Basic notations used in use case diagram:
* **Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.
* **System:** A specific sequence of actions and interaction between actors and the system. A system may also be referred to as a scenario.
* **Use Case** : They represent the system's functions.



**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 UML class diagram is made up of:
  + A set of classes and
  + A set of relationships between classes
* A class notation consists of three parts:
  + Class Name: The name of the class appears in the first partition.
  + Class Attributes: Attributes are shown in the second partition and map onto member variables (data members) in code.
  + Class Operations (Methods): Operations are shown in the third partition. They are services the class provides and map onto class methods in code
  + Class Relationships: A class may be involved in one or more relationships with other classes.



**3)SEQUENCE DIAGRAM**

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when



**4)COLLABORATION DIAGRAM**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). A collaboration diagram resembles a flowchart 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. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.



**5)ACTIVITY DIAGRAM**

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. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

The purpose of an activity diagram can be described as −

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.
* Describe the parallel, branched and concurrent flow of the system.

Before drawing an activity diagram, we should identify the following elements

* Activities
* Association
* Conditions
* Constraints

Once the above-mentioned parameters are identified, we need to make a mental layout of the entire flow. This mental layout is then transformed into an activity diagram.



**6)COMPONENT DIAGRAM**

Component diagrams are different in terms of nature and behavior. They are used to model the physical aspects of a system. Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Their purpose can be summarized as −

* Visualize the components of a system.
* Construct executables by using forward and reverse engineering.
* Describe the organization and relationships of the components.



**7)DEPLOYMENT DIAGRAM**

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. They are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

