SRS DOCUMENTATION

Time Cards Management System

Project by

--BhanuvikasYaganti

--Anish Reddy Ellore

--Anshul Garg

TABLE OF CONTENTS

1 INTRODUCTION

This section gives a scope description and overview of everything included in this SRS documents. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1 PURPOSE

The purpose of this document is to give the detailed description of the requirements for the "TIME CARDS MANAGEMENT SYSTEM" (TCM) software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions within the application. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

1.2 SCOPE

The "TIME CARDS MANAGEMENT SYTEM" is a stand-alone desktop application which helps organizations to manage salaries of employees based on leaves, working hours and holidays.

Employees can provide their daily working hours by logging into this application. This information will act as the basis for the calculation of their monthly salary. Employer can manage salaries of employees through this application.

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

Term	**Definition**
Employee	Person who works in the organization
Employer	Person who manages the organization

1.4 REFERENCES

[1] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

1.5 OVERVIEW

The remainder of this document includes three chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used to specify the requirements more precisely for different audiences.

The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen.
The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.

2. OVERALL DESCRIPTION

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1 PRODUCT PERSPECTIVE

This system will consist of only one parts: Desktop application and a database. The desktop application will be used by both employees and employer. Employees will be able to login into the application and so their daily working hours keeps counting till they logout. This application uses some clock & calendar to record the time . Employers will have an extra privilege to see the statistics of progress of employees and edit the stats and evaluate monthly salary of his employees.
Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. The application will communicate with the database. The employer will be able have more privileges in accessing the database. All the database communication will go via local system.

2.2 PRODUCT FUNCTIONS

With this desktop application the employee will be able to record their daily working hours by logging into the application as a employee and employer will be able to check the employee statistics and edit the stats or delete a particular employee.

Employer when clicks on a particular employee, he will be able to see the data pertaining to that particular employee and will be able to edit any of the data(base salary, sick leaves, casual leaves etc..). The changes made will be updated in the database as soon as the alert box is clicked ‘YES’.

2.3 USER CHARACTERISTICS

There are two types of users that interact with the system: Employer and employee. Each of these two types of users has different use of the system so each of them has their own requirements.

The employee of the organization can use the application to login and logout which helps employer to track his working hours. While, the employer will be able to edit the employee’s data such as number of sick leaves and casual leaves and will be able to add or delete the employee.

2.4 Constraints

The desktop application is constrained by the system interface to the calendar and clock within the system. Since there are multiple operating systems and multiple manufacturers, the interface will most likely not be the same for every one of them. Also, there may be a difference between what navigation features each of them provides.

2.5 Assumptions and dependencies

One assumption about the product is that date and time in the system are accurate, because the system will access that data to track the working hours of the employees.

3. SPECIFIC REQUIREMENTS

This section contains all the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

3.1 External interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface

3.1.1 User interfaces

A first-time user of the desktop application should see the log-in page when he/she opens the application, see Figure 2, Figure1. If the user has not registered, he/she should be able to do contact the employer so that he will be able to add the employee into the list of employees.

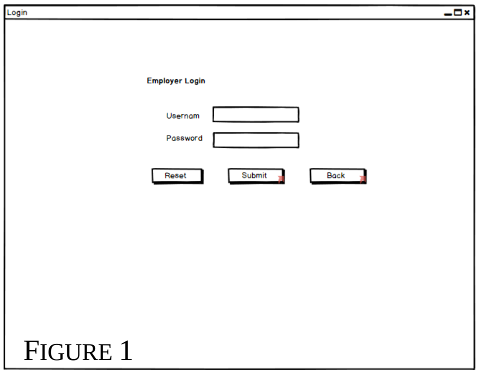


Figure 1

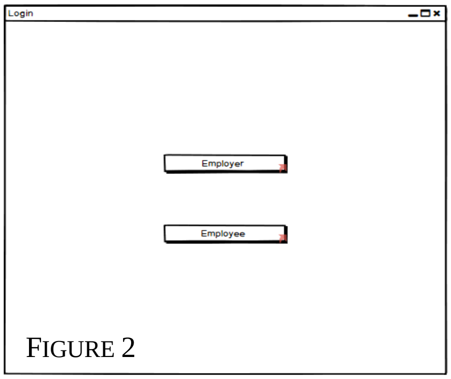
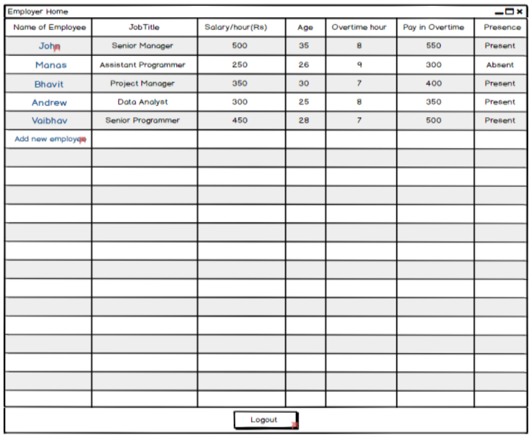


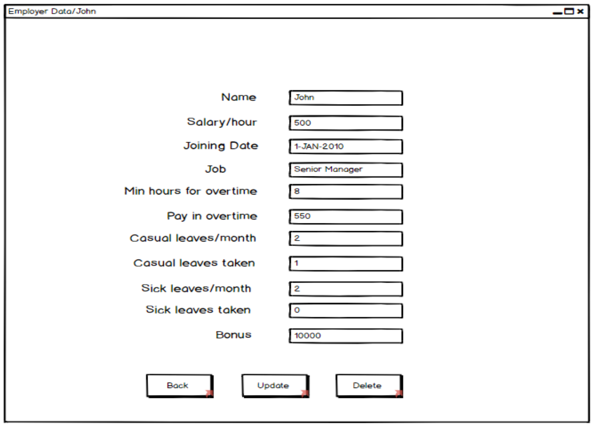
Figure 2

Employer can see the employee details (working hours, job title, salary etc.) when he logs into the system, see Figure 3. He can update the details of an existing employee, change the no. of given sick leaves or casual leaves, pay in overtime and can also delete the employee ,see Figure 4. He can add new users as shown in Figure 5.



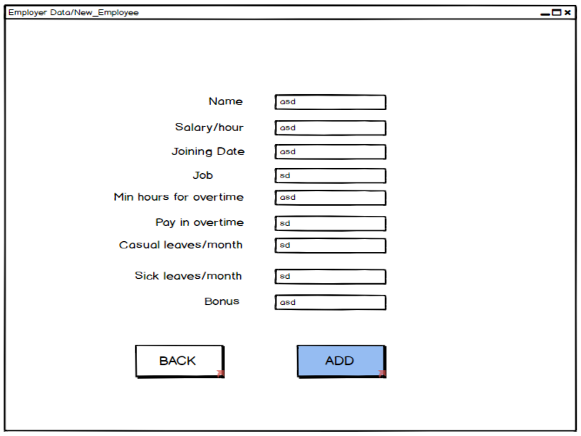
Name of Employee	JobTitle	Salary/hour(Rs)	Age	Overtime hour	Pay in Overtime	Presence
John	Senior Manager	500	35	8	550	Present
Manas	Assistant Programmer	250	26	9	300	Absent
Bhavit	Project Manager	350	30	7	400	Present
Andrew	Data Analyst	300	25	8	350	Present
Vaibhav	Senior Programmer	450	28	7	500	Present
Add new employee						

Figure 3



Name	John
Salary/hour	500
Joining Date	1-JAN-2010
Job	Senior Manager
Min hours for overtime	8
Pay in overtime	550
Casual leaves/month	2
Casual leaves taken	1
Sick leaves/month	2
Sick leaves taken	0
Bonus	10000

Figure 4



Name	asd
Salary/hour	asd
Joining Date	asd
Job	sd
Min hours for overtime	asd
Pay in overtime	sd
Casual leaves/month	sd
Sick leaves/month	sd
Bonus	asd

Figure 5

Employee should be able to see some details as shown in figure 6 when he logs in.

3.1.2 HARDWARE INTERFACES

Since neither the desktop application nor the database have any designated hardware, it does not have any direct hardware interfaces. The physical clock and calendar application in the desktop application and hardware connection to the database connection is managed by the underlying operating system on the application.

3.1.3 SOFTWARE INTERFACES

The desktop application communicates with the calendar and clock application in order to get the date and time information about the user logging see figure 6.
The communication between the database and the application consists of operation concerning both reading and modifying the data, while the communication between the database and the application consists of only reading operations.

3.1.4 COMMUNICATIONS INTERFACES

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for both the mobile application and database.

3.2 FUNCTIONAL REQUIREMENTS

This section includes the requirements that specify all the fundamental actions of the software system.

3.2.1 USER CLASS 1 – THE EMPLOYEE

3.2.1.1 Functional requirement 1.1

ID: FR1
TITLE: Employee registration - Desktop application
DESC: Given that a organization has taken the desktop application, then the employee should be able to register via the Employer. The Employer should only have the privilege to add the employee. The employee must provide user-name to employer and later he can reset his password.
RAT: For a user to register on the desktop application.
DEP: FR1

3.2.1.2 Functional requirement 1.2

ID: FR2
TITLE: Employee log-in - Desktop application
DESC: Given that a employee has registered by contacting the employer, then the user should be able to log in to the desktop application and he should be able to reset the password.
RAT: For a Employee to register on the desktop application.
DEP: FR1

3.2.1.3 Functional requirement 1.3

ID: FR3
TITLE: Reset password
DESC: Given that a employee has registered, then the user should be able to reset his password
RAT: For a user to reset his/her password.
DEP: FR1, FR2.

3.2.1.4 Functional requirement 1.4

ID: FR4
TITLE: DE application – Check in
DESC: Given that a employee is logged in to the desktop application, then the first page that is shown should be welcome page and the option to check in. The user should be able to check in and logout option.
RAT: For a user to check in
DEP: FR4

3.2.2 USER CLASS 2 - THE EMPLOYER

3.2.2.1 Functional requirement 2.1

ID: FR6

Feature: Restaurant owner log-in

In order to use the system, the employer should be logged in to the desktop application

Scenario: Successful log-in
Given the employer wants to log in, When the employer logs in with his/her account Then the employer should be logged in as a employer.
Scenario: Retrieve password
Given the employer wants to log in and has lost the password, When the employer clicks the “Reset” button and submits the new password, then the employer should be able to set the new password.

3.2.2.2 Functional requirement 2.2

ID: FR7
Feature: Manage information
In order to manage information employer should be logged in to the desktop application.

S**cenario: Show fields for managing information**
Given the employer is logged in, The employer wants to manage information
Then the employer owner should be able to manage information in a form in

which he should be able to see registered employees and on clicking a particular employee, the employer should see detailed description of employee.

Scenario: Editing information

Given the employer is logged in and wants to edit information of a employee, he should see the list of employees and on clicking on a particular employee he should see detailed description of the user details such as working hours, casual and sick leaves, base salary. When the employer edits information then the information should be edited by conforming with an alert box.

Scenario: Deleting employee

Given the employer is logged in and the employee list exists, when the employer deletes a particular, he should be able to delete by conforming with an edit box. Then the employee should be deleted.

3.2.1.5 Functional requirement 1.5

ID: FR8
TITLE: Desktop application - Profile page
DESC: On the Desktop application, the Employee should have a profile page which can be accessed by the employer. On the profile page the employer can edit information of employee, which includes the casual and sick leaves, base salary & extra working hours.
RAT: For a employee to have a profile page on the desktop application which can only be accessed by employer.
DEP: FR5, FR6.

3.3 Software system attributes

The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

3.5.1 Reliability

ID: QR1
TAG: System Reliability
GIST: The reliability of the system.
SCALE: The reliability that the system gives the right result on calculating the salary.
METER: Measurements obtained from # calculations during testing.
MUST: More than 100% of the searches.
PLAN: More than 100% of the searches.
WISH: 100% of the calculations.

3.5.2 Availability

ID: QR2
TAG: System Availability
GIST: The availability of the system when it is used.
SCALE: The average system availability (not considering database failure).
METER: Measurements obtained from # hours of usage during testing.
MUST: More than 98% of the time.
PLAN: More than 99%

ID: QR3
TITLE: Calendar and clock availability.
DESC: The application should be able to access the calendar and clock in the system.
RAT: For the application to calculate record exact date of execution.
DEP: none

3.5.3 Security

ID: QR4
TAG: EmployeeLoginAccountSecurity
GIST: Security of accounts.
SCALE: If an employee tries to log in to the application with a wrong password then the employee should not be logged in. The employee should be notified about log-in failure.
METER: # attempts to log-in with a non-existing user account during testing.
MUST: 100% of the time

ID: QR5
TAG: UserCreateAccountSecurity
IST: The security of creating account for users of the system.
SCALE: If the employer wants to create an account for a new employee and the desired user name is occupied, the employer should be asked to choose a different user name.
METER: Measurements obtained on # hours of usage during testing.
MUST: 100% of the time

3.5.4 Maintainability

ID: QR6
TITLE: Application extendibility
DESC: The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.
RAT: In order for future functions to be implemented easily to the application.
DEP: none

ID: QR7
TITLE: Application testability
DESC: Test environments should be built for the application to allow testing of the applications different functions.
RAT: To test the application.
DEP: none

3.5.5 Portability

ID: QR8
TITLE: Application portability
DESC: The application should be portable with all platforms of operating systems.
RAT: The adaptable platform for the application to run on.
DEP: none