**High Level Design - Employee Management System**

Table of Contents

[1. Revision History 2](#_Toc85392115)

[2. Approval 2](#_Toc85392116)

[3. Target Audience/Distribution List 3](#_Toc85392117)

[4. Reference Documents 3](#_Toc85392118)

[5. Introduction 3](#_Toc85392119)

[6. Requirement 3](#_Toc85392120)

[7. Purpose 4](#_Toc85392121)

[8. Scope 4](#_Toc85392122)

[9. Exclusions 4](#_Toc85392123)

[10. Architecture 4](#_Toc85392124)

[11. ER Diagram 4](#_Toc85392125)

[12. Requirements 5](#_Toc85392126)

[13. Constraints 6](#_Toc85392127)

[14. Technologies used 6](#_Toc85392128)

[15. Tools and Software required to develop 6](#_Toc85392129)

[16. S/w and H/w sizing 6](#_Toc85392130)

[17. Acronyms and Abbreviations 7](#_Toc85392131)

* Development
* Unit level testing
* EOI against Tender
* Project Discussion
* BID Submission
* Acceptance & Handshake

Database

* HLD
* LLD
* Testcase Document
* Delivery
* Feedback
* Warranty Support
* Payment and Closure
* System Integration Testing
* User Acceptance Testing

# Revision History

|  |  |  |
| --- | --- | --- |
| Revision Number | Change By | Changes Incorporated |
| PA1 |  | Initial Draft |
| PA2 | Saswata |  |
|  |  |  |
|  |  |  |
| PA10 |  |  |

# Approval

|  |  |  |
| --- | --- | --- |
| Approved By | Designation | Approval Type |
|  |  | Provisional/Final |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Target Audience/Distribution List

|  |  |
| --- | --- |
| Name | Designation |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Reference Documents

|  |  |
| --- | --- |
| Document Name | Location  Physical Location/Website |
| Scope Of Work |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Introduction

This High Level Design Document is intended to provide a brief summary of the project and setup a common agreement between the customer and the solution provider to finalize a design architecture so that there should not be any understanding and design gap that may erupt during development phase of the project. Employee Management System will manage the data of employees along with Department, Pay Scale and Product she worked on. Each employee must be working under his/her manager and there must be a hierarchical relationship among employees. Only CEO will not have any manager. There are different departments inside the organization and each employee will be working in any one department. So there is an one-to-one relationship between employee and department. Each employee must have only one Pay Scale through which she will be paid monthly salary. Each Pay Scale will be identified by a definite code and must contain all related information for each components of calculating salary. Employees will be working on one or multiple products which organization will be dealing with. There must be a start and end period information of each product the employees are working on.

On the report part, management should have the below reports any time.

* Departmentwise employee information
* Projectwise employee information

# Requirement

Since inception, Company ... is maintaining its employee information in excel files along with some other unformatted file, like pictures. This creates the problem of maintain employee information in an unified way and creates a lot of discrepancies and mis-understanding between HR personnel.

So a definite, proper and centrally managed system is required to put in place to replace the existing excel-managed system so that data should be managed in an efficient way.

# Purpose

A proper employee management system provides the below advantages which mostly not possible for any unstructured and file-based legacy systems:

* It provides data authenticity
* Provides data accessibility
* Hide sensitive data
* Removes concurrency
* Removes data duplication
* Avoid data loss

# Scope

**EMS (SPOC:) – Employee Management System** must have the facility to do the following:

* add an employee
* update an employee
* dormant any employee with fixed reasons (Retire, Resign, Layoff)
* sort the data based on employees
* view all employees with pagination
* search and generate report based on some query criteria
* each employee should have these data: id, name, dob, department code (FK to department), salary code (FK to salary), project code (FK to project), mail-id, mobile, address, etc.
* dob should have date validation property
* mail-id should be format-validation
* department code should be taken from a fixed predefined table with some department information (unique code, name, mail-id, phone)
* salary code should be taken from a fixed predefined table with some scale data (unique code, minbasic, maxbasic, da-percent, hra-percent, pf-percent)
* project code should be taken from a fixed predefined table with some project data (unique code, name, about)
* authentication and authorization of user during login to do the above operations

**LMS (SPOC: )– Library Management System** must have the facility to do the following:

* Admin user can do the below:
  + Add a book with count of books
  + Disable/enable a book to be issued
* Other Library user can do the below:
  + search a book
  + issue a book to a reader
  + refund a book from reader
* system will automatically keep track of the number of availability of a particular book
* system can provide report of a book on date of issue/refund
* system have facility to report book which is not refunded for a required number of days since issued
* there is a predefined reader information table having some data (unique id, name, address, phone)
* book must have the attributes: unique id, name, writer, isbn, publisher (FK to publisher), print-year
* publisher id should be taken from a fixed predefined table with some publisher information (unique code, name, mail-id, phone)
* generate report on books' names by publisher

**MMS (SPOC:) – Medical Management System** must have the facility to do the following:

* patients can book an appointment for a doctor
* doctors information must be kept in a fixed pre-defined table (code, name, specialization, fee)
* every doctor is assumed to visit for 6hours per day (mon to sat, 10am to 1pm and 2pm to 5pm) and each patient is allotted 15min for treatment
* no two patients will book a same slot in a day
* doctors can see their schedules
* patients can choose any doctor with available slots
* patients can cancel an appointment before 1 day of its schedule

**BMS (SPOC:) – Bus Management System** must have the facility to do the following:

* commuters can book a bus ticket from starting point to end point for a day
* there is a predefined bus information in the system (code, regn-number, routecode, start, end, facilities, seatcount, fare\_per\_km)
* routecode, start and end must be coming from route table
* there is a predefined route table (routecode, stopno, stopname, dist\_km)
* booking information for a bus can be retrieved for any day
* during booking, commuters can see the total fare depending on fare\_per\_km and dist\_km
* no waitlist ticket will be issued
* commuters can book at max 6 seats at a time
* after booking a PNR (unique 10digit alphanumeric data) will be generated for each ticket
* commuters can cancel its ticket as a whole, no partial cancellation
* report can be generated for busses with under-utilized seat (say less than 70% seat fillup)

# Exclusions

This system will not consider the below functionalities as part of development:

* This system will not keep employees' monthly payroll data and can't calculate salary.
* This system will not maintain any work related information in the developed system.

# Architecture

In the architecture part, the details of system and flow of the data is described. Employee management system consists of employee related information and their management – create update and delete – along with employee department, scale code and product information.

# ER Diagram

Required ER diagram is depicted below.



# Use Case

Below are the Use cases for development of employee management system.

1. All types of work must be done through web-based GUI form.
2. User can create employee information
3. User can update employee information.
4. User can delete employee information until her salary is generated or not worked in any product.
5. User can create department.
6. User can't modify any department name.
7. User can't delete department if at least one employee is assigned to that department.
8. User can create pay scale data.
9. User can create product information.
10. User can allocate product to an employee.
11. User can deallocate a product from an employee.

# Constraints

Constraints of this software development is defined here so that plan be defined beforehand. Some constraints of this development is described here.

* Development must be done inside the customer premises and the employee data is the sole property of customer.
* Access of Customer premises must be between morning 10.00am to 5p.m and no extended period of work is allowed inside customer premises.
* Customer side desktops, to be used for development, are quite old and using core-i3 CPU with only 2GB RAM.

# Technologies used

Here are the list of technologies used for this application:

* Frontend Development – HTML, CSS, Javascript, jQuery
* Middle tier business logic – Spring Framework, Hibernate ORM
* Backend database – PostgreSQL database
* Reporting(Excel) – Apache POI

# Tools and Software required to develop

Employee management system is a multiuser web-based software to maintain employee information efficiently. Data will be kept in database, PostgreSQL, and interaction between user and database will be done using a middle tier software to be developed in java. For Java/JSTL/HTML/JSP development, eclipse (newer version) will be used. For database development, pgAdmin will be used.

# S/w and H/w sizing

Software and Hardware must be specified to cater the SLA (Service Level Agreement) defined between Customer and Software supplier. To meet the pre-defined SLA, it is required to define the correct sizing of hardware to provide service withing stipulated time. For multiuser system and huge transaction rate software and hardware must be designed and tuned to the appropriate level.

S/w requirement:

* OS – Windows 10/linux
* Java – version 8
* Spring – 4.x
* Hibernate – (version)
* PostgreSQL – (version)

H/w requirement:

* RAM – 4GB
* Storage – 512 MB

# Acronyms and Abbreviations

* HLD – High Level Design
* SLA – Service Level Agreement
* JSTL – Java Standard Tag Libraries
* S/W – Software (Java version, Database version, Reporting tool software)
* H/W – Hardware (Server side CPU/RAM/LAN and Network architecture)
* ER – Entity Relationship