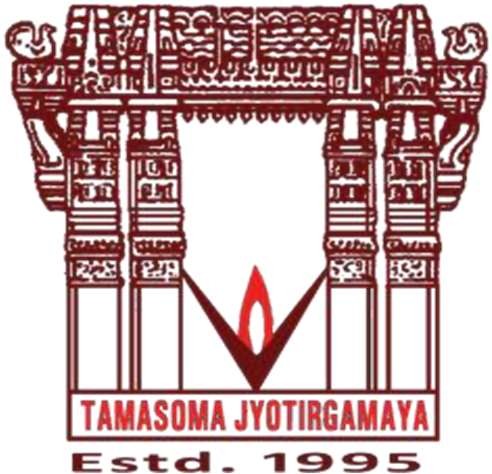


High Level Design Document

**On**

Employee Management System



VNR Vignana Jyothi Institute of Engineering & Technology Bachupally, Nizampet (S.O), Hyderabad–90

***Submitted By***

***Group Details:***

**Avuluri Nikhitha - 20075A0513**

**Damera Shyam Kumar - 20075A0514**

**Pooja Kumari Singh - 20075A0516**

**Shaik Yasmin Sulthana - 20075A0517**

**Siliveru Arun -20075A0518**

Contents

[Revision History](#_TOC_250027) 3

* 1. [Purpose](#_TOC_250026) 4
  2. [Audience](#_TOC_250025) 4
  3. [Design Process](#_TOC_250024) 4

1. [Requirements](#_TOC_250023) 5
   1. [Proposed Solution 5](#_TOC_250022)
   2. [Capacity Planning](#_TOC_250021) 6
2. [Architecture 6](#_TOC_250020)
   1. [Design 6](#_TOC_250019)

Use-Case Diagram

[Front end](#_TOC_250018) 7

* Client 7
* [Web Browser 7](#_TOC_250017)

Back end

* [Web Server](#_TOC_250016) 8
* Business Logic 8
* Storage 8

3.1.1[Version](#_TOC_250012) 9

* 1. [Access](#_TOC_250010) 9

3.3 [Hardware and Platform Requirements](#_TOC_250009) 9

[Hardware Requirements](#_TOC_250008) 8

* Processor 8
* Memory 8
* Display 8

[Packages Installed 9](#_TOC_250006)

* 1. [System Connectivity](#_TOC_250005) 10

1. [Standards](#_TOC_250004) 10
   1. [Security Standards](#_TOC_250003) 10

[4.1.1 Authorization and Logon](#_TOC_250002) 10

* 1. [Disaster Recovery](#_TOC_250001) 10

5.[Support](#_TOC_250000) 10

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Author | Section | Comments/Changes |
| 1.0.0 | 17 April 21 | All | All | Initial Revision |
| 1.1.0 | 21 April 21 | Train | All | Additional Content |
| 1.2.0 | 24 April 21 | All | All | Additional Content |
| 1.3.0 | 29 April 21 | Myers | Requirements | Updated section with Persistency information |
| 1.3.1 | 01 May 21 | Train | All | Edit for consistency and mistakes |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 1. Project Overview

**1.1 Purpose**

The purpose of this document is to specify the high-level design for the EMS (Employee Management System). This document will act as an outline for implementation and discuss the design considerations.

**1.2 Audience**

This high-level design is intended to be used by members of the development team that will implement the functionality of the EMS. This document will also be used to communicate the high-level design and design considerations to team members.

* Nikhitha, Development Team
* Shyam Kumar, Development Team
* Pooja, Development Team
* Yasmin, Development Team
* Arun, Development Team

**1.3 Design Process**

The high-level design was selected by deciding what aspects of the system were most important and then building an architecture around them. Architecture that was proposed: distributed system utilizing the single-page web applications architecture (SPAs) and technologies like angular, typescript and NodeJS. The pros and cons of each architecture and technology were discussed in meetings. For each technology proposed we explored and researched on feasibility and capability. Web app type is very agile, responsive, and lightweight.

AngularJS is a JavaScript-based front-end web framework based on bidirectional UI data binding and is used to design Single Page Applications. Single Page Applications are web applications that load a single HTML page and only a part of the page instead of the entire page gets updated with every click of the mouse. The page does not reload or transfer control to another page during the process. This ensures high performance and loading pages faster. Most modern applications use the concept of SPA. In the SPA, the whole data is sent to the client from the server at the beginning.

As the client clicks certain parts on the webpage, only the required part of the information is fetched from the server and the page is rewritten dynamically. This results in a lesser load on the server and is cost-efficient. SPAs use AJAX and HTML5 to create a fluid and responsive Web applications and most of the work happens on the client-side. Popular applications such as Face book, Gmail, Twitter, Google Drive, Netflix, and many more are examples of SPA.

**2. Requirements**

* The system will be having user privileges based menu.
* User will have to select the options form the given Menu.
* The system will be entering the information into the Database to generate reports.
* The forms will be designed to enter the data.
* Buttons will be used to insert, retrieve or modify the Data.

**2.1 Proposed Solution**

The proposed system is designed to eliminate all the Drawbacks of the existing employee management Software. The system shall be responsible for Maintaining information about employees, thus their Personal profile.

The system shall incorporate **Leave Management** all the way from application to Acceptance/rejection of leave requests as well.

Employee projects with close monitoring of the projects From creation to completion and trainings to assist in Monitoring active and inactive employees.

It also Includes The **Attendance management** where it displays the Number of working days of each Employee.

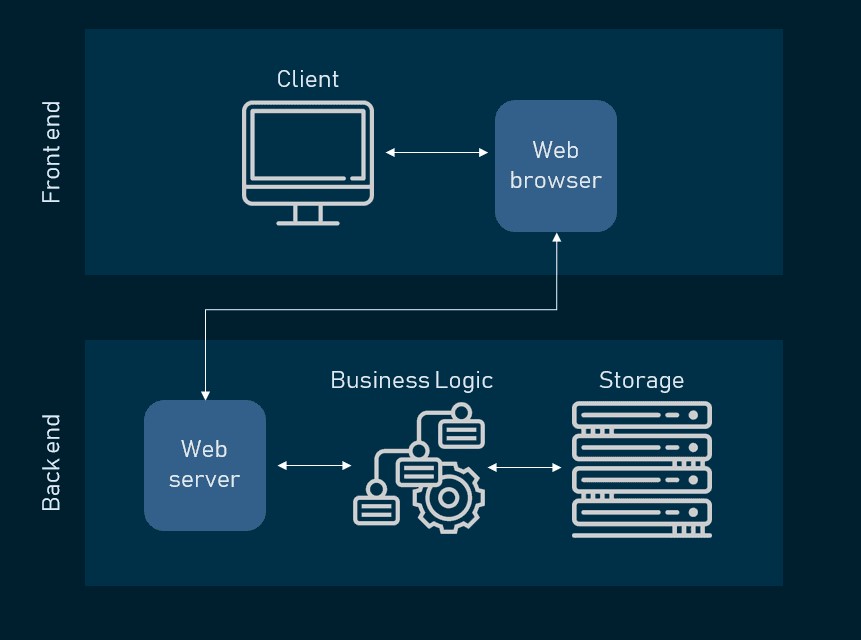
And the main aspect of the entire project comes with the **Salary management** Of the employee. It calculates the salary based on the Basic Salary, Dearness Allowance, HRA etc...

**2.2 Capacity Planning**

At a project level, capacity planning is all about determining whether you have enough resources to complete a given project .resources may mean people, equipment, software, rooms, and so on. We are using it to signify people with the skills necessary to meet the requirements of a project. Ex. Technical Skills ,Conceptual Skills, Interpersonal and Communication Skills, Decision-Making Skills.

**3. Architecture**

**3.1 Design**

****

**Use-Case Diagram**

**EMS**

****

**Front end**

* **Client**

The Client accesses a web application via a web browser, triggering a request to the web server over the internet. Web server forwards the request to business logic. Business logic performs the requested task such as querying the database or processing data and generates the results. Business logic sends the result back to the web server and web server delivers the requested information to the client by web browser

* **Web browser :**

A web browser that may be chrome, safari, Firefox on different devices like computer, mobile, Laptop. Web browser will act as an interface between the

client and web server so that whenever client needs to send any request to the web server, web browser will carry out the request to the web server and the served requests will be displayed on the desktop or mobile device to the client by web browser

**Back end**

* **Web Server**

The web server for the EMS provides an access for all authorized users to the EMS web application. Here Apache used as web server, its job is to establish a connection between a server and the browsers of website visitors. To access web server user need to login to the EMS web application. When the user login to the EMS then the web server starts serving based on the request given by the user

* **Business Logic**

Business Logic for the EMS, accepts user requests from the browser, processes them, and determines the routes through which the data will be accessed. The workflows by which the data and requests travel through the back end are encoded in a business layer

* **Storage**

The Database for the EMS provides all the necessary data to the web server. All the data will be edited by the administrator in the database. Here, MySQL is used as database,MySQL is a relational database management system based on SQL–Structured Query Language. When the employee or administrator requests to view the information, the data stored in the database by the administrator will be given to apache web server then it will display it to the user

**3.1.1 Version**

Version 1.0 – Initial version of the system that has the basic multi-page application.

Version 1.1 – After meeting with , the development team found out that there would be

Single-page application. The current version utilizes Angular technology to avoid page

reloading for every mouse click.

**3.2 Access**

The application will be accessible through web browser. The client will take the user credentials and login to the system and sends the request to the web server and web server retrieves data from the database and the retrieved data will be processed by the business logic written using nodejs, angular, javascript, typescript, html, css. This processed data will be sent to the web server and client will receive the data from web server and serves the user

* 1. **Hardware and Platform Requirements**

**Hardware Requirements**

* **Processor:**
  + Minimum: 1.9 gigahertz (GHz) x86- or x64-bit dual core processor with SSE2 instruction set
  + Recommended: 3.3 gigahertz (GHz) or faster 64-bit dual core processor with SSE2 instruction set
* **Memory:**
  + Minimum: 2-GB RAM
  + Recommended: 4-GB RAM or more
* **Display:** 
  + Minimum: Super VGA with a resolution of 1024 x 768
  + Recommended: Super VGA with a resolution of 1024 x 768

**Packages installed**

* Node js 14.16.1 LTS
* npm 6.14.12
* @angular/cli
* Typescript
  1. **.System Connectivity**

The client will connect with the system using Internet so both client and server needs to reside on the Internet and the necessary ports are available to both client and server.

**4. Standards**

**4.1 Security Standards**

**4.1.1 Authorization and Logon**

The system shall verify the username and password using the LDAP authentication.

**4.2 Disaster Recovery**

Restart server program as directed in the operational manual and the system will return to last safe state.

**5.Support**

The following support documentation will be provided: Code, Design Document, Operations Manual, and Deployment Plan. The system code shall be documented according to the "Code Conventions for the Front End (Angular,JavaScript,TypeScript,HTML,CSS),Back End(Node js) The system shall be described by a "Design Document." The system shall be accompanied by an "Operations Manual" describing proper use of the system. The system shall be deployed using operations described in the "Deployment Plan."