Employee Management System

# Technical Design Document

V.1

# Introduction:

This document covers details technical design of the employee management system. Employee management system allows you to manage various employees within an organization.

# Proposed Design:

Requirements:

* User can add/edit/delete(deactivate) employee.
* User can upload documents associated with the user in word,pdf,text format only.

## System Architecture

* Proposing design following layered architecture.
* There are following distinct layers
  + Front End
  + API Layer
  + Service
  + Data Access Layer
* Front End will be handled by Angular 9
* Angular 9 will talk with REST services created using ASP.NET Web API Core. These services will intern call logical layers like Service Layer and Data Access Layer.
* Data Access Layer will make use of entity framework core as a ORM tool and it will talk with SQL server database

System Architecture

ANGULAR APP

REST

SERVICE

SERVICE (Business Layer)

Data Access Layer

SQL Server

SQL Server

ANGULAR APP

Data Access Layer

SERVICE (Business Layer)

REST

SERVICE

Database Entities



### Technologies Used

* Single Page Web Application technology is used using Angular 9
* Angular app will consume ASP.NET web api core REST services.
* Entity Framework Core will be used as ORM framework for data access layer.
* NUnit and Moq will be used for unit testing and mocking.

### Design considerations

* Files are stored as a blob (nvarbinary(MAX)) instead of file share. This will enable us to deploy the application over the load balancer which will enable High Availably as well as enable performance.
* In first phase authentication and authorization will not be in scope however it is essential to put this module in the next phase to enable security and access control.
* ASP.NET core itself takes care of sanitization code to avoid any security related threats like SQL injection, script injection.
* Entity Framework core takes care of SQL injection vulnerability by using parameters.
* Highly recommended to use latest TLS on a production server with appropriate TLS certificate to avoid transport level security issue.
* Appropriate CORS policy will be implemented in production. In development, CORS policy is relieved .

Deployment Architecture

* Angular application and REST api will be deployed on a single web server as a single we application.
* There will be separate database server used to store the database.