Employee Portal

## Design Specification Document

Version 1.0

Document Control Information

### Document Information

|  |  |
| --- | --- |
| Document Identification |  |
| Document Name | EmployeePortal\_DesignSpecification\_v1.0.docx |
| Project Name | Employee Portal [Code Challenge] |
| Document Author | Team **GenY-Techies** |
| Document Version | 1.0 |
| Document Status | Valid |
| Date Released | 7th Feb 2018 |

### Document Edit History

| Version | Date | Additions/Modifications | Prepared/Revised by |
| --- | --- | --- | --- |
| 1.0 | 7th Feb 2018 | First Release | Team **GenY-Techies** |
|  |  |  |  |

### Team Details

|  |  |  |
| --- | --- | --- |
| Team Name : GenY-Techies | | |
| S.No. | Name | Email ID |
| 1 | Nikita Agrawal | nikiagrawal@deloitte.com |
| 2 | Deepak Ranjan Sahoo | deepsahoo@deloitte.com |
| 3 | Kritika Jain | kritikjain@deloitte.com |
| 4 | Nimit Saxena | [nimsaxena@deloitte.com](mailto:nimsaxena@deloitte.com) |

Table of Contents

[Design Specification Document 1](#_Toc505782511)

[Document Information 2](#_Toc505782512)

[Document Edit History 2](#_Toc505782513)

[1. Introduction 4](#_Toc505782514)

[1.1. Document Purpose 4](#_Toc505782515)

[1.2. Scope 4](#_Toc505782516)

[1.3. Maintenance 4](#_Toc505782517)

[2. Application Elements and Mappings 5](#_Toc505782518)

[2.1. Screen Navigation 5](#_Toc505782519)

[2.2. UI Design 5](#_Toc505782520)

[2.3. Field Format and Database Mapping 5](#_Toc505782521)

[2.4. Service and Business Layer Design 6](#_Toc505782522)

[2.5. Business Validations 7](#_Toc505782523)

[3. Database Design 8](#_Toc505782524)

[4. Process Flows 10](#_Toc505782525)

[4.1. Flow Diagrams 10](#_Toc505782526)

[4.2. Sequence Diagrams 10](#_Toc505782527)

# Introduction

## Document Purpose

This document conveys the design and implementation specific details about the **Employee Portal** Application that has been developed by Team **GenY-Techies** for the solution of code challenge problem.

## Scope

The scope of the application is limited to the set of requirements provided in the problem statement. Following are the high level scopes covered by this solution:

* + 1. Employee Sign Up
    2. Employee Sign In
    3. Employee Dashboard

## Maintenance

The application has been modelled as a distributed application. Front-end Components and Model-Business Backend implementation are separate deployable that can be hosted on different servers and interact through HTTP REST calls for above stated operations.

Thus, maintenance of this application involves

* + 1. Hosting the individual components on web hosting servers.
    2. Scale up and scale down can be configured for these components depending on the load increase and decrease respectively.
    3. PCF instances can be engaged for this activity.

# Application Elements and Mappings

## UIDesign

## Field Format and Database Mapping

Update the following properties file “src/main/resources/application.properties” with the database connection credentials:

**server.port=8080**

**spring.datasource.url=jdbc:mysql://localhost:3306/eoy**

**spring.datasource.username=root**

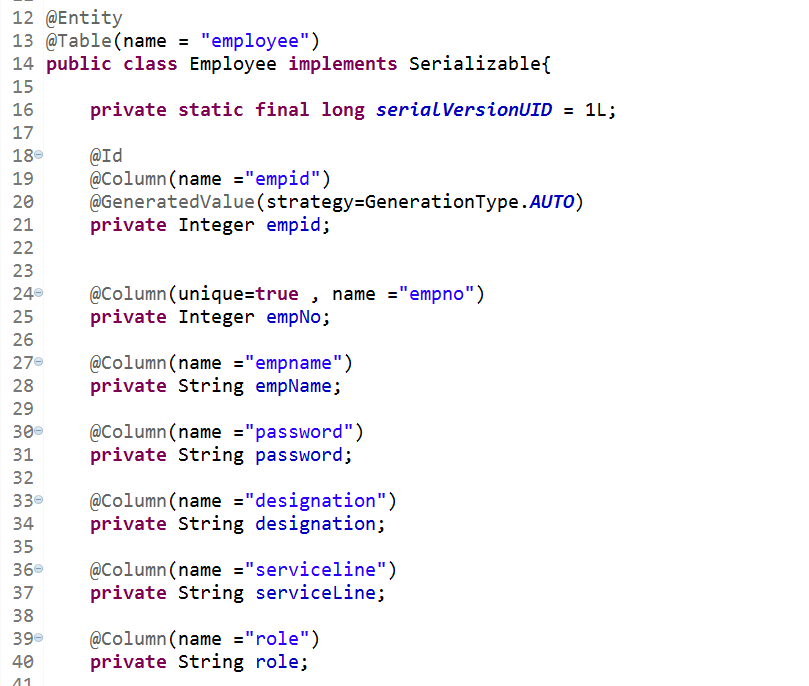
**spring.datasource.password=kjain28**

**spring.jpa.show-sql=false**

**spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect**

**spring.jpa.hibernate.ddl-auto=create**

The application engages ORM layer integration that maps the schema in the external database MySQL.



With table **employee,** following are the mapped attributes from Java Bean to the table scema

* + 1. empNo : Unique, Integer
    2. empName : String
    3. password : String | Encoded
    4. serviceLine : String
    5. role : String

## Service and Business Layer Design

* + 1. Back-end is designed as an autonomous and independently hosted RESTful service.
    2. These Spring based RESTful services consumes and produces JSON requests and responses respectively for their individual business calls from the front end.
    3. The application has been designed in a modular way where the business / backend layer is loosely coupled with the presentation/UI layer. Hence, these services can be integrated with presentation layer or any other business client.
    4. Exposed REST endpoint are:
       1. **http://<server>/register/signUp**

This accepts the input JSON object over HTTP POST, containing the employee details persists the same to the underlying database through ORM. For password encryption, the application has employed Base64 encoder format. Password is persisted to the database using Base64 encoded format.

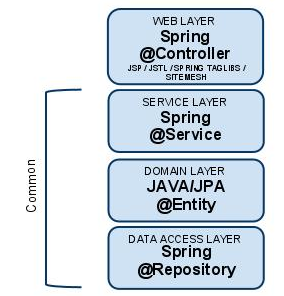
* + - 1. **http://<server>/register/signIn**

This accepts the input JSON object over HTTP POST, with input username and password this service authenticates the user with the provided input credentials. Encoded Password is fetched from Database, decoded and matched with the provided input password for authentication.

* + - 1. **http://<server>/employee/getAllEmployees**

This services fetches all the employees in the employee database that have successfully registered for the employee portal and produces a JSON Array in response.

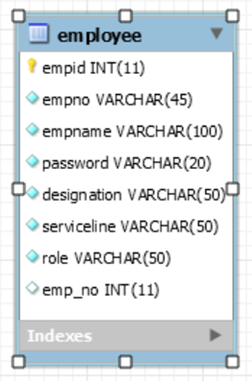
* + 1. Following is the representation of the laid out architecture:



## Business Validations

* + 1. All fields are mandatory for sign up.
    2. All fields are mandatory for sign in
    3. Sign up cannot proceed with a pre-existing employee number. An asynchronous call is fired from the signup screen while signup to check for the employee number to be pre-existing.

# Database Design



Attached is the DDL script for the above table:



CREATE TABLE `employee` (

`empid` int(11) NOT NULL auto\_increment,

`empno` varchar(45) NOT NULL,

`empname` varchar(100) NOT NULL,

`password` varchar(20) NOT NULL,

`designation` varchar(50) NOT NULL,

`serviceline` varchar(50) NOT NULL,

`role` varchar(50) NOT NULL,

`emp\_no` int(11) default NULL,

PRIMARY KEY (`empid`),

UNIQUE KEY `empno\_UNIQUE` (`empno`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Holds data for registered employee'; Process Flows

## Application Flows

1. Signup 🡪 SignIn 🡪 EmployeeDashboard 🡪 (Upon Logout) SignIn
2. SignIn 🡪 EmployeeDashboard 🡪 (Upon Logout) SignIn

\*\*\*\*End Of Document\*\*\*\*