**DB Table details**

Tables

Activity

|  |  |
| --- | --- |
| ID | Integer |
| Name | varchar |
| Description | varchar |
| Training flag | boolean |

Training\_Invitation

|  |  |
| --- | --- |
| Id | Integer |
| date | Varchar |
| Training\_id | Integer [fk training table] |

Employee

|  |  |
| --- | --- |
| Emp\_id | Integer |
| Employee Name | varchar |
| Date of birth | date |
| address | varchar |

Attendance Tracker

|  |  |
| --- | --- |
| ID | Integer |
| Emp Id | Integer [fk employee table] |
| Training Id | Integer [fk training table] |
| Last\_updated\_date | date |
| Start\_date | date |
| Date\_to\_complete | date |
| Completion\_status | varchar |

Role

|  |  |
| --- | --- |
| Role Id | Integer |
| Role Name | varchar |

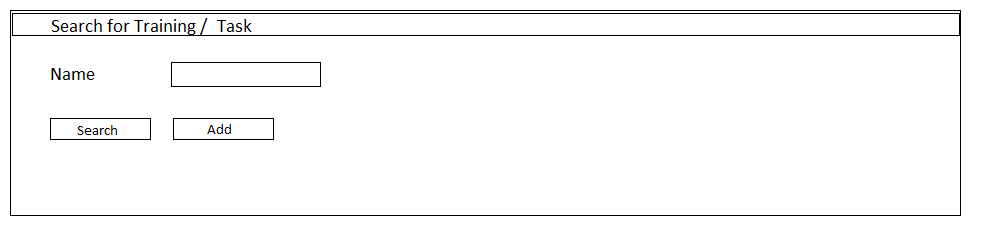
User Table

|  |  |
| --- | --- |
| id | Integer |
| Username | Varchar (Unique) |
| First name | varchar |
| Last Name | varchar |
| Password | Varchar (Encrypted String) |
| Created\_date | date |
| Role\_id | Integer [fk role table] |

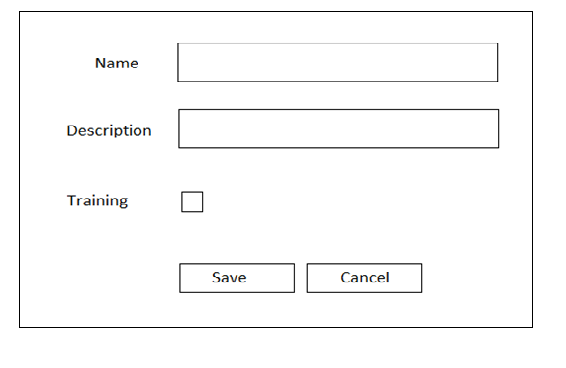
**UI Design Info for Angular**

* Login Page -Three types of user can log in to the system who having below three roles
  + Administrator
  + Manager
  + Trainee
* Home Page Modules

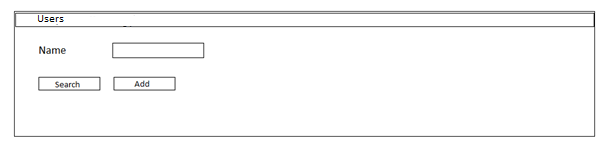
**Training / Task (Admin)**



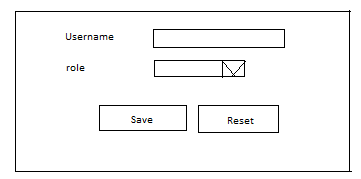
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Training | Date | Modify |
| Security training | Yes / No | 05/02/2021 | Edit/ Delete |



**Users (Admin)**

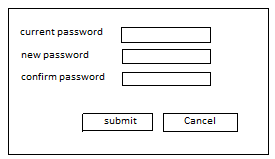
****

|  |  |  |  |
| --- | --- | --- | --- |
| User Name | Role | Last Updated | Modify |
| Shiva | Admin | 05/02/2021 | Edit/ Delete |

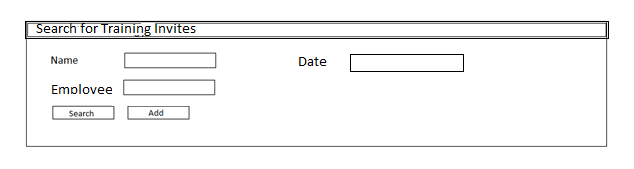


**Note**: Initially admin will create user and have default password, then later respective user can change the password.

**Change Password (Admin,Manager,Trainee)**

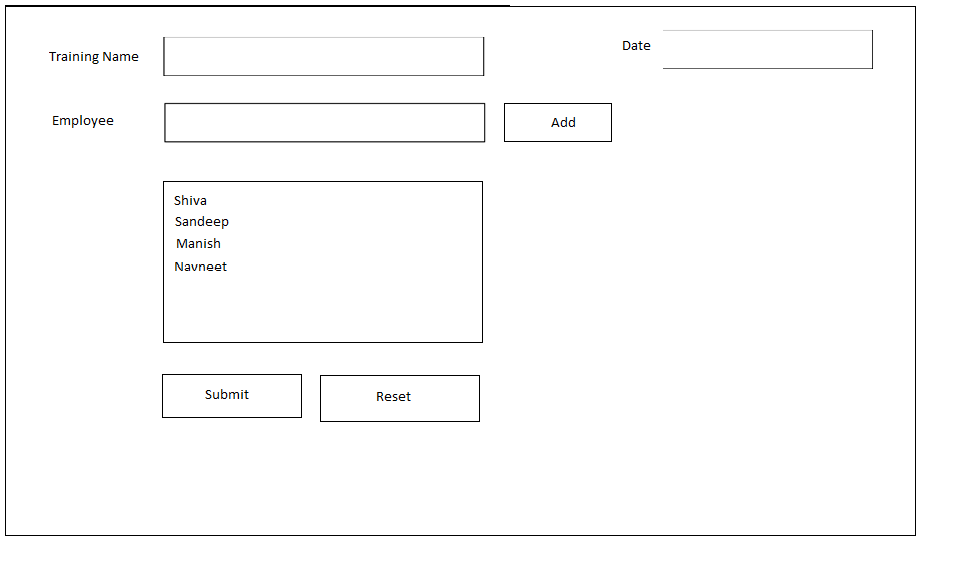
****

**Training Tracker (Admin / Manager)**



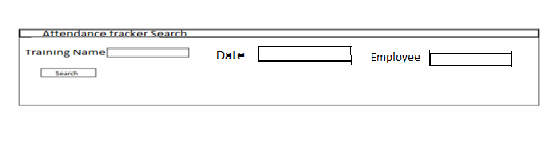
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Employee | Start Date | Modify |
| Security training | Shiva, Keshav, Sandeep, manish | 05/02/2021 | Edit/ Delete |

**Note**: Manager Role can only view doesn’t have modify permission



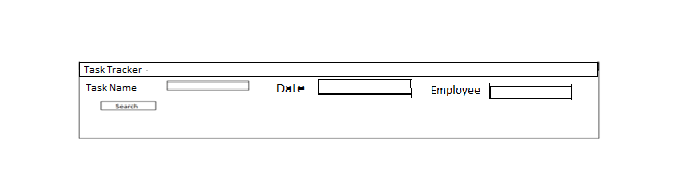
**Note**: While submitting the request email will be triggered to all the employee that are selected.

**Attendance Tracker (Admin, Manager)**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Employee | Start Date | Hours Spent | Status |
| Security training | Shiva | 05/02/2021 | 8 | completed |
| Security training | Manish | 05/02/2021 | 7 | In Progress |
| Security Training | Sandeep | 05/02/2021 | 0 | Not Started |

**Task Tracker (Manager)**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Employee | Start Date | Last Date to Complete | Hours Spent | Status |
| Module Development | Shiva | 05/02/2021 | 25/02/2021 | 11 | In Progress |
| Bug Fixing | Shiva | 05/02/2021 | 25/02/2021 | 12 | In Progress |
| Team Discussion on project schedules | Shiva | 05/02/2021 | 25/02/2021 | 3 | Completed |

**Training Details (Trainee)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Training Name** | **Status** | **Hours spent** | **Date** | **Last date to Complete** | **Action** |
| Security Training | In Progress | 10 Hours | 05/02/2021 | 25/02/2021 | Mark As Completed |
| Money Laundry Training | Not Started | 12 hours | 05/02/2021 | 25/02/2021 | Mark As Completed |
| Java Skill training | Completed | 9 Hours | 05/02/2021 | 25/02/2021 | Mark As Completed |

**Task Details( Trainee)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task Name** | **Status** | **Hours spent** | **Date** | **Last date to Complete** | **Action** |
| Module Development | In Progress | 11 Hours | 05/02/2021 | 25/02/2021 | Mark As Completed |
| Bug Fixing | Not Started | 12 hours | 05/02/2021 | 25/02/2021 | Mark As Completed |
| Team Discussion on project schedules | Completed | 3 Hours | 05/02/2021 | 25/02/2021 | Completed |

**Note**: As the Task details there was no proper specific requirement , I have created the screen for that and to utilize the trainee activity to track in more clear and efficient way.

**Backend Design**

This project should be implemented using spring boot with user authentication and authorization by JWT and bcrypt password encoding algorithm and spring-data for fetching data and Rest api should be used for fetching and should be documented using swagger configuration. Roles need to be applied based on the module level and respective api that we are calling like for admin, manager and trainee.

Database : MySql

Below are the coding package structure to be followed in Java.

App

-Src

com.app.base.

config

SwaggerConfig.java

controller

HomeController.java

AttendanceTrackerController.java

TrainingOrTaskController.java

TrainingInviteController.java

ExportDataController.java

UserController.java

Model

Activity.java

TrainingInvitation.java

AttendanceTracker.java

Employee.java

User.java

Role.java

JwtRequest.java

JwtResponse.java

Repository

UserRepository.java

EmployeeRepository.java

AttendanceTrackerRepository.java

ActivityTrackerRepository.java

TrainingInvitationRepository.java

Service

UserService.java

CommonService.java

CommonServiceImpl.java

UserServiceImpl.java

EmployeeService.java

EmployeeServiceImpl.java

AttendanceTracker.java

AttendanceTrankerImpl.java

TrainingTrackerService.java

TrainingTrackerServiceImpl.java

Security

JWTAuthenticationEntryPoint.java

JWTRequestFilter.java

JWTTokenUtil.java

WebSecurityConfig.java

SecurityConstants.java

Util

AppUtil.java

AppConstants.java

SendMailUsage.java

TraingingAppApplication.java

-Resource

Application.properties

-document

Pom.xml

**Dependencies**

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger2</artifactId>

<version>2.7.0</version>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger-ui</artifactId>

<version>2.7.0</version>

</dependency>

<dependency>

<groupId>org.springframework.data</groupId>

<artifactId>spring-data-jpa</artifactId>

<version>2.4.2</version>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.16</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

<version>2.4.2</version>

</dependency>

<dependency>

<groupId>org.springframework.security.oauth</groupId>

<artifactId>spring-security-oauth2</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-jwt</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

</dependency>

<dependency>

<groupId>jakarta.mail</groupId>

<artifactId>jakarta.mail-api</artifactId>

<version>2.0.0</version>

</dependency>

**Note**:There are few common methods to used across other modules each other like training and taks modules and exporting data these services we can generalize and reuse the codes.

**Sequence Diagram**

