**HRM Tool Documentation**

Contents

[1. Introduction 1](#_Toc153500509)

[1.1 Overview 1](#_Toc153500510)

[1.2 Purpose 1](#_Toc153500511)

[1.3 Goals 1](#_Toc153500512)

[2. **Project Structure** 2](#_Toc153500513)

[2.1 **Main Directories** 2](#_Toc153500514)

[2.2 **Key Directories in src/main/java** 2](#_Toc153500515)

[2.3 **Key Files in src/main/resources** 2](#_Toc153500516)

[2.5 **Security Configuration** 2](#_Toc153500517)

[2.6 **Project Configuration** 2](#_Toc153500518)

[2.7 **User Roles** 2](#_Toc153500519)

[2.8 **Authentication Success Handler** 2](#_Toc153500520)

[3. Key Features 3](#_Toc153500521)

[3.1 Employee Dashboard 3](#_Toc153500522)

[3.2 File Management 4](#_Toc153500523)

[3.3 Leave Management 4](#_Toc153500524)

[3.4 Attendance Tracking 4](#_Toc153500525)

[3.5 News and Announcements 4](#_Toc153500526)

[3.6 Training Records 4](#_Toc153500527)

[3.7 User Management 5](#_Toc153500528)

[Controller Classes Overview 5](#_Toc153500529)

[4.1 EmployeeController 5](#_Toc153500530)

[4.2 HRController 5](#_Toc153500531)

[4.3 ManagerController 6](#_Toc153500532)

[4.4 UserLRController 7](#_Toc153500533)

[5. **Service Interfaces** 8](#_Toc153500534)

[5.1 **AttendenceService** 8](#_Toc153500535)

[5.2 **EmployeeService** 8](#_Toc153500536)

[5.3 **FileService** 8](#_Toc153500537)

[5.5 **NewsService** 8](#_Toc153500538)

[5.6 **UserService** 8](#_Toc153500539)

[**6. SecurityConfig** 9](#_Toc153500540)

[7. Exception Handlers 11](#_Toc153500541)

[7.1 YourCustomAuthenticationSuccessHandler 11](#_Toc153500542)

[8. **Model Classes** 12](#_Toc153500543)

[8.1 Attendance 12](#_Toc153500544)

[8.2 Email 12](#_Toc153500545)

[8.3 Emplyees 13](#_Toc153500546)

[8.4 FileSystem 13](#_Toc153500547)

[8.5 FileSystem2 13](#_Toc153500548)

[8.6 LeaveApplication 13](#_Toc153500549)

[8.7 News 13](#_Toc153500550)

[8.8 TrainingRecord 13](#_Toc153500551)

[8.9 User 14](#_Toc153500552)

[9.Plugins and Dependencies 14](#_Toc153500553)

[10. Installation 16](#_Toc153500554)

# 

# 1. Introduction

## 1.1 Overview

The HRM Tool is a comprehensive Human Resource Management system designed to streamline and enhance HR processes within an organization. This tool provides a centralized platform for managing various aspects of employee data, leave applications, attendance, training records, and more.

## 1.2 Purpose

The primary purpose of the HRM Tool is to automate and simplify HR-related tasks, promoting efficiency and accuracy in managing workforce data. By leveraging technology, the tool aims to improve the overall HR workflow, reduce manual efforts, and ensure better organization-wide communication.

## 1.3 Goals

* **Efficient Data Management:** Facilitate the storage and retrieval of employee information, ensuring data accuracy and accessibility.
* **Leave Application Automation:** Streamline the process of applying for and approving leaves, reducing paperwork and manual intervention.
* **Attendance Tracking:** Provide a mechanism to monitor and manage employee attendance efficiently.
* **Training Record Maintenance:** Keep track of employee training records, both internal and external, to ensure continuous skill development.
* **Communication Enhancement:** Improve communication within the organization through features like news and announcements.
* **User Authentication and Authorization:** Implement a robust security system with role-based access control to protect sensitive HR data.

# 2. [**Project Structure**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#project-structure)

The HRM Tool project follows a structured organization to maintain a clear separation of concerns and ease of development. Here is an explanation of the project structure:

## 2.1 [**Main Directories**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#main-directories)

* **src/main/java**: Contains the Java source code for the project.
* **src/main/resources**: Holds configuration files, templates, and static resources.
* **src/test**: Houses the unit and integration tests.

## 2.2 [**Key Directories in src/main/java**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#key-directories-in-src-main-java)

* **com.example.hrm\_tool\_springboota.Controller**: Controllers handling HTTP requests.
* **com.example.hrm\_tool\_springboota.Modal**: Model classes representing entities.
* **com.example.hrm\_tool\_springboota.Repository**: Interfaces for database interactions.
* **com.example.hrm\_tool\_springboota.Service**: Service classes providing business logic.
* **com.example.hrm\_tool\_springboota.ExceptionHandler**: Custom exception handlers.

## 2.3 [**Key Files in src/main/resources**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#key-files-in-src-main-resources)

* **application.properties**: Configuration properties for the Spring Boot application.
* **templates**: Thymeleaf templates for rendering views.
* **static**: Static resources like CSS, JavaScript, and images.

2.4 [**Database Configuration**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#database-configuration)

* **com.example.hrm\_tool\_springboota.Configuration**: Configuration files for database setup.

## 2.5 [**Security Configuration**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#security-configuration)

* **com.example.hrm\_tool\_springboota.Configuration.SecurityConfig**: Configuration for Spring Security.

## 2.6 [**Project Configuration**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#project-configuration)

* **build.gradle**: Dependencies and build settings for the Gradle build tool.
* **plugins**: Additional plugins used in the project.
* **lombok.config**: Configuration for the Lombok library.

## 2.7 [**User Roles**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#user-roles)

* **com.example.hrm\_tool\_springboota.Controller.UserLRController**: User registration and roles assignment.

## 2.8 [**Authentication Success Handler**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#authentication-success-handler)

* **com.example.hrm\_tool\_springboota.ExceptionHandler.YourCustomAuthenticationSuccessHandler**: Redirects users based on roles after login.



# 3. Key Features

The HRM Tool is designed to provide comprehensive human resource management capabilities. Below are the key features along with detailed descriptions, benefits, and use cases:

## 3.1 [Employee Dashboard](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#employee-dashboard)

* **Description:** The Employee Dashboard provides a personalized view for each employee, displaying relevant information such as attendance, leave status, and announcements.
* **Benefits:**
  + Employees can easily track their attendance and leave balances.
  + Quick access to important announcements and news updates.
* **Use Cases:**
  + Employees can view their attendance history.
  + Access and apply for leaves directly from the dashboard.

## 3.2 [File Management](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#file-management)

* **Description:** The File Management feature allows users to upload, download, and view files related to their work, fostering efficient document sharing.
* **Benefits:**
  + Centralized storage for important documents.
  + Easy sharing of files among team members.
* **Use Cases:**
  + Uploading and sharing project-related documents.
  + Downloading files for reference or review.

## 3.3 [Leave Management](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#leave-management)

* **Description:** Leave Management enables employees to request leaves, and managers can approve or disapprove leave applications.
* **Benefits:**
  + Streamlined leave request and approval process.
  + Accurate tracking of employee leave balances.
* **Use Cases:**
  + Submitting leave requests with supporting documents.
  + Managers reviewing and managing leave applications.

## 3.4 [Attendance Tracking](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#attendance-tracking)

* **Description:** The Attendance Tracking feature allows employees to check in and out, and managers can monitor attendance records.
* **Benefits:**
  + Real-time attendance tracking.
  + Identifying patterns in employee attendance.
* **Use Cases:**
  + Employees checking in/out using the system.
  + Managers reviewing attendance reports.

## 3.5 [News and Announcements](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#news-and-announcements)

* **Description:** News and Announcements keep employees informed about company updates, events, and important news.
* **Benefits:**
  + Keeping employees engaged and informed.
  + Centralized communication platform.
* **Use Cases:**
  + Posting updates on company policies.
  + Announcing upcoming events or initiatives.

## 3.6 [Training Records](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#training-records)

* **Description:** Training Records track internal and external training programs attended by employees.
* **Benefits:**
  + Monitoring employee skill development.
  + Ensuring compliance with training requirements.
* **Use Cases:**
  + Recording details of attended training sessions.
  + Generating reports on employee training history.

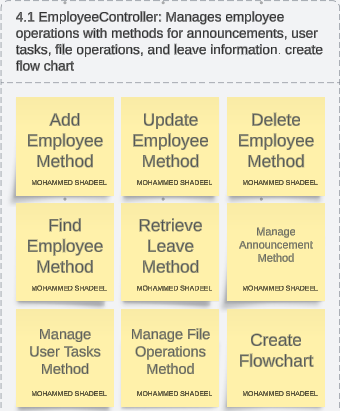
## 3.7 [User Management](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#user-management)

* **Description:** User Management allows administrators to manage user roles and permissions.
* **Benefits:**
  + Customizing access levels for different users.
  + Ensuring security and compliance.
* **Use Cases:**
  + Adding or removing user roles.
  + Adjusting permissions based on job roles.

# Controller Classes Overview

## 4.1 EmployeeController

* **Responsibilities:** Manages employee-related functions.
* **Key Methods:**
  + **getAnnouncements(Model model)**: Retrieves and displays announcements.
  + **user(Model model, Long id, LeaveApplication leaveApplication, @ModelAttribute("email") Email email)**: Handles user-specific tasks.
  + **getfilemanagement(Model model)**: Manages file-related operations.
  + **saveleave(...)**: Saves leave information.

****

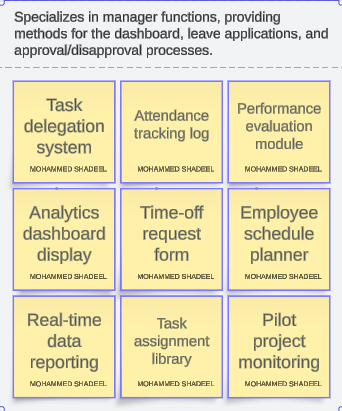
## 4.2 HRController

* **Responsibilities:** Handles HR-related tasks.
* **Key Methods:**
  + **dashboard(Model model)**: Displays HR dashboard statistics.
  + **getLeaveApplication(Model model)**: Manages leave applications.
  + **sendEmail(@ModelAttribute("Data\_info") Email email)**: Sends emails.
  + **getEmployee\_Data(Model model)**: Displays employee data.



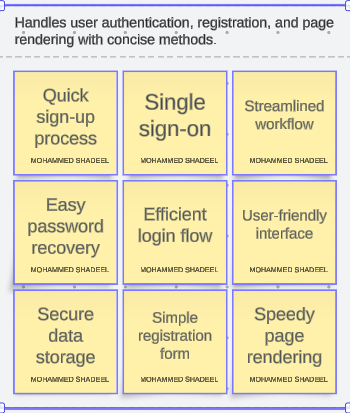
## 4.3 ManagerController

* **Responsibilities:** Manages manager-specific functions.
* **Key Methods:**
  + **getDashboard()**: Renders manager dashboard.
  + **getApplications(Model model)**: Handles leave applications.
  + **getApproval(...)** and **getDisapproval(...)**: Approves/disapproves leave applications.



## 4.4 UserLRController

* **Responsibilities:** Deals with user authentication and registration.
* **Key Methods:**
  + **home()**, **login()**, **register()**: Render pages.
  + **registerUser(...)**: Handles user registration.



# 5. [**Service Interfaces**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#service-interfaces)

In the HRM Tool project, service interfaces play a crucial role in encapsulating business logic and facilitating communication between controllers and repositories. Here's an overview of the service interfaces along with their responsibilities:

## 5.1 [**AttendenceService**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#attendenceservice)

* **Responsibilities:**
  + Saving check-in data for attendance.

## 5.2 [**EmployeeService**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#employeeservice)

* **Responsibilities:**
  + Retrieving various statistics related to employees (total, active, full-time, part-time, gender-wise).
  + Deleting an employee.
  + Handling pagination for employee data.

## 5.3 [**FileService**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#fileservice)

* **Responsibilities:**
  + Managing files, including listing, uploading, and retrieving file data.

5.4 [**LeaveService**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#leaveservice)

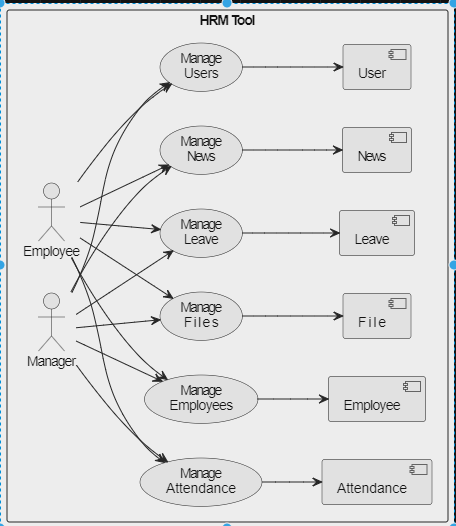
* **Responsibilities:**
  + Handling leave applications.
  + Saving, approving, and disapproving leave applications.
  + Handling email notifications for leave status.

## 5.5 [**NewsService**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#newsservice)

* **Responsibilities:**
  + Creating and retrieving news items.

## 5.6 [**UserService**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#userservice)

* **Responsibilities:**
  + Managing user registration.
  + Saving user details.
  + Handling user roles.



# **6. SecurityConfig**

The **SecurityConfig** class is responsible for configuring Spring Security settings and defining role-based access to different parts of the HRM Tool application.

Class Overview:

* **Package:** **com.example.hrm\_tool\_springboota.Configuration**
* **Annotations:**
  + **@Configuration**: Indicates that this class is a configuration class.
  + **@EnableWebSecurity**: Enables Spring Security for the web application.

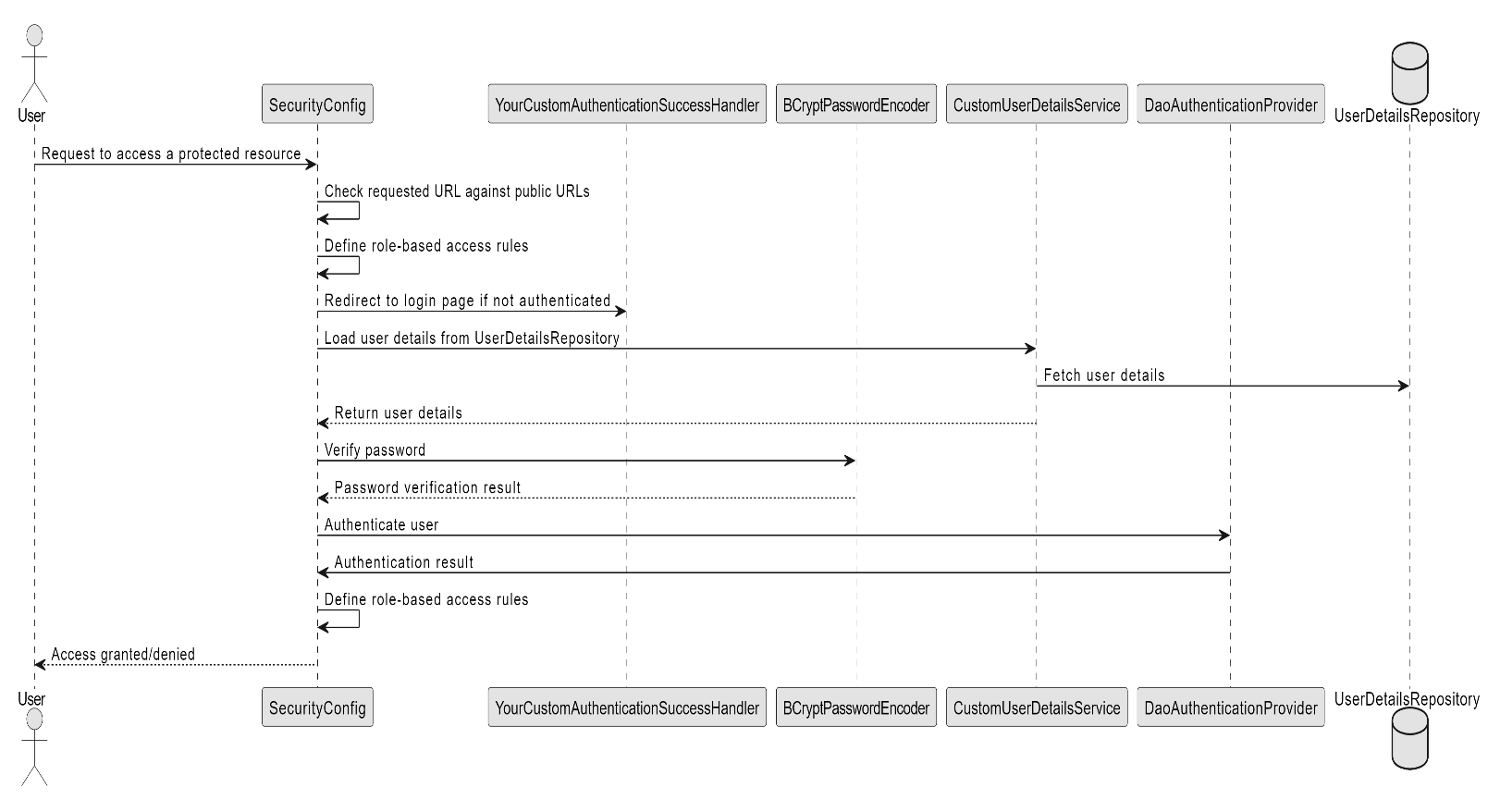
Methods:

1. **passwordEncoder():**
   * **Type:** Bean method
   * **Responsibility:** Provides a bean for the **BCryptPasswordEncoder** used for password encoding.
2. **getDetailsService():**
   * **Type:** Bean method
   * **Responsibility:** Provides a bean for the **CustomUserDetailsService**, which is a custom implementation of Spring Security's **UserDetailsService**.
3. **getDaoAuthenticationProvider():**
   * **Type:** Bean method
   * **Responsibility:** Provides a bean for the **DaoAuthenticationProvider**, configuring it with the custom user details service and password encoder.
4. **securityFilterChain(HttpSecurity http):**
   * **Type:** Bean method
   * **Responsibility:** Configures the security filter chain for HTTP security.
   * **Details:**
     + Defines role-based access control for different parts of the application.
     + Allows access to certain paths (like login and registration) without authentication.
     + Specifies that paths under "/api/v1/" should require the "HR" role.
     + Specifies that paths under "/api/v2/" should require the "EMPLOYEE" role.
     + Specifies that paths under "/api/v3/" should require the "MANAGER" role.
     + Configures form-based login with a custom login page and a success handler.
     + Disables CSRF protection for simplicity.

Dependencies:

* **YourCustomAuthenticationSuccessHandler**: A custom success handler for authentication success.
* **CustomUserDetailsService**: A custom implementation of **UserDetailsService** for loading user-specific data.
* **BCryptPasswordEncoder**: A password encoder using the bcrypt hashing algorithm.

This configuration ensures that the application has proper security measures in place, allowing or restricting access based on user roles.



# 7. Exception Handlers

This section covers the custom exception handling mechanism implemented in the HRM Tool project. Exception handling is vital to manage errors gracefully and provide users with meaningful feedback. The **YourCustomAuthenticationSuccessHandler** class is a key component in this process.

## 7.1 YourCustomAuthenticationSuccessHandler

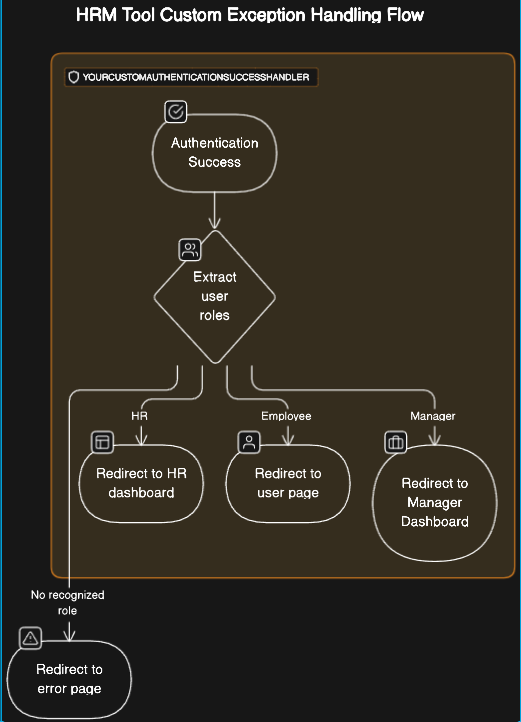
**Responsibility:** The **YourCustomAuthenticationSuccessHandler** class is responsible for managing the redirection of users after a successful authentication. It determines the appropriate destination based on the user's assigned role.

**Key Features:**

1. **Role-Based Redirection:**
   * If the authenticated user has the role "HR," they are redirected to the HR dashboard (**/api/v1/dashboard**).
   * If the user has the role "EMPLOYEE," the redirection is to the Employee dashboard (**/api/v2/user**).
   * For users with the role "MANAGER," the destination is the Manager dashboard (**/api/v3/Manager\_\_Dashboard**).
2. **Fallback Redirect:**
   * If the user's role is not recognized (falls into none of the specified roles), they are redirected to the error page (**/error**).

**Usage:** This handler is invoked upon a successful login, ensuring a seamless and role-specific user experience. By redirecting users to the appropriate dashboards, it enhances the user interface and ensures a contextually relevant environment.

**Implementation:** The class implements the **AuthenticationSuccessHandler** interface, allowing it to handle successful authentication events. It utilizes the Spring Security framework to extract the user's roles and determine the appropriate redirect based on those roles.



# 8. [**Model Classes**](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#model-classes)

* Overview of key model classes in the HRM Tool project.
* Fields and relationships within each model.

# 8.1 [Attendance](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#attendance)

* Fields:
  + **id**: Auto-generated identifier.
  + **employee**: Many-to-One relationship with **Emplyees**.
  + **checkinTime**: Check-in time.
  + **checkoutTime**: Check-out time.
  + **checkoutDate**: Check-out date.

## 8.2 [Email](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#email)

* Fields:
  + **id**: Identifier.
  + **from**: Sender's email.
  + **to**: Receiver's email.
  + **subject**: Email subject.
  + **body**: Email body.
  + **leaveApplication**: One-to-Many relationship with **LeaveApplication**.

## 8.3 [Emplyees](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#emplyees)

* Fields:
  + **Employee\_ID**: Auto-generated identifier.
  + **FName**: First name.
  + **LName**: Last name.
  + Other details such as **StartDate**, **ExitDate**, **Title**, **ADEmail**, etc.

## 8.4 [FileSystem](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#filesystem)

* Fields:
  + **id**: Auto-generated identifier.
  + **filedata**: Binary data of the file.
  + **filename**: Name of the file.
  + **fileContent**: Content of the file.

## 8.5 [FileSystem2](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#filesystem2)

* Fields:
  + **id**: Auto-generated identifier.
  + **filedata**: Binary data of the file.
  + **filename**: Name of the file.
  + **fileContent**: Content of the file.

## 8.6 [LeaveApplication](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#leaveapplication)

* Fields:
  + **id**: Identifier.
  + **file\_id**: ID of the associated file.
  + **full\_name**: Full name of the employee.
  + **additional\_comments**: Additional comments.
  + Other details such as **department**, **employee\_id**, **job\_title**, etc.
  + **file**: One-to-Many relationship with **FileSystem2**.
  + **emails**: Many-to-One relationship with **Email**.

## 8.7 [News](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#news)

* Fields:
  + **id**: Auto-generated identifier.
  + **Title**: News title.
  + **Content**: News content.

## 8.8 [TrainingRecord](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#trainingrecord)

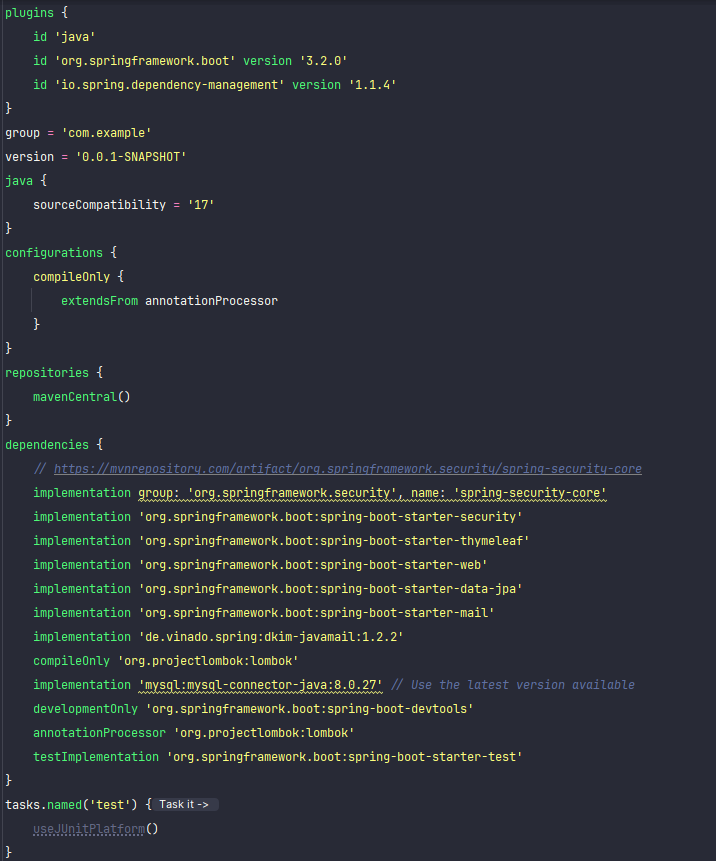
* Fields:
  + **Employee\_id**: Employee ID.
  + **Training\_Date**: Date of training.
  + **Training\_Program\_Name**: Name of the training program.
  + Other details such as **Training\_Type**, **trainingoutcome**, **Location**, etc.

## 8.9 [User](https://chat.openai.com/c/98f0ae21-4ace-4ffc-84dd-9448b95d0c41#user)

* Fields:
  + **id**: Auto-generated identifier.
  + **fullName**: Full name of the user.
  + **gender**: Gender of the user.
  + **confirmPass**: Confirmation password.
  + **mobileNo**: Mobile number of the user.
  + **password**: User password.
  + **email**: User email.
  + **roles**: User roles.

# 9.Plugins and Dependencies

**Build Configuration:**



**Notes:**

1. **Spring Boot Version (3.2.0):** Specifies the version of the Spring Boot framework.
2. **Source Compatibility (Java 17):** Ensures compatibility with Java 17.
3. **Dependency Management Plugin (1.1.4):** Manages project dependencies, simplifying version management.

**Key Dependencies:**

* **Spring Security Core:** Provides core security functionality.
* **Spring Boot Starters:** Simplify the configuration of Spring applications.
* **Thymeleaf:** A modern server-side Java template engine for web and standalone environments.
* **Spring Data JPA:** Simplifies data access using the JPA (Java Persistence API) specification.
* **Spring Boot DevTools:** Provides fast application restarts, among other development features.
* **Lombok:** Reduces boilerplate code in Java classes.
* **MySQL Connector:** Enables connectivity to a MySQL database.
* **DKIM for JavaMail (1.2.2):** Implements DomainKeys Identified Mail (DKIM) for JavaMail.

This configuration and these dependencies collectively form the foundation of the HRM Tool project, ensuring security, efficient development, and seamless integration with databases and mail services. Adjustments may be made to versions based on project requirements and compatibility.

# 10. Installation

1. **Clone the Repository:** Open a terminal or command prompt and navigate to the directory where you want to clone the project. Run the following command:

`***git clone <repository\_url***> `

Replace **<repository\_url>** with the actual URL of your GitHub repository. For example:

*`****git clone https://github.com/your-username/your-hrm-tool-project.git*** *`*

1. **Navigate to Project Directory:** Move into the project directory:

` ***cd your-hrm-tool-project*** `

1. **Build and Run the Project:** Use Gradle to build and run the project. In the project directory, run:

`***./gradlew bootRun*** `

On Windows, use:

`***gradlew.bat bootRun*** `

1. **Access the Application:** Once the project is successfully built and running, open a web browser and go to:

arduinoCopy code

http://localhost:8080

This assumes that the Spring Boot application is configured to run on the default port (8080). If you have configured a different port, adjust the URL accordingly.

You should now be able to access and interact with the HRM Tool application locally.

1. **Stop the Application:** To stop the running Spring Boot application, go back to the terminal where the application is running and press **Ctrl + C**.

That's it! You've successfully cloned, built, and run the Spring Boot HRM Tool project from GitHub using Gradle. Adjust the commands based on your project's specific requirements.