**Exercise 7: Employee Management System - Enabling Entity Auditing**

Business Scenario:

Implement auditing to track the creation and modification of employees and departments.

**1. Introduction**

This document provides an overview of the Employee Management System, detailing its main components, including entities, controllers, repositories, and configuration. The system is designed to manage employee and department data efficiently using Spring Boot and JPA.

**2. Key Components**

**2.1 Entities**

* **Employee**: Represents an employee with fields for id, name, email, and a reference to a Department. It also includes auditing fields for tracking creation and modification dates.
* **Department**: Represents a department with fields for id, name, and auditing fields similar to Employee.

**2.2 Repositories**

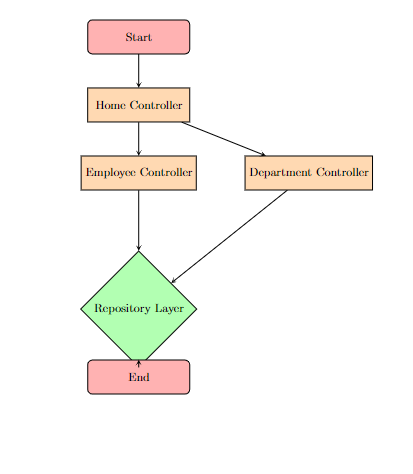
* **EmployeeRepository**: Interface for performing CRUD operations on Employee entities. Includes custom query methods to find employees by name or email.
* **DepartmentRepository**: Interface for performing CRUD operations on Department entities.

**2.3 Controllers**

* **EmployeeController**: Manages HTTP requests related to employees. Supports pagination and sorting of employee data.
* **DepartmentController**: Manages HTTP requests related to departments. Includes methods for creating, reading, updating, and deleting departments.

**2.4 Configuration**

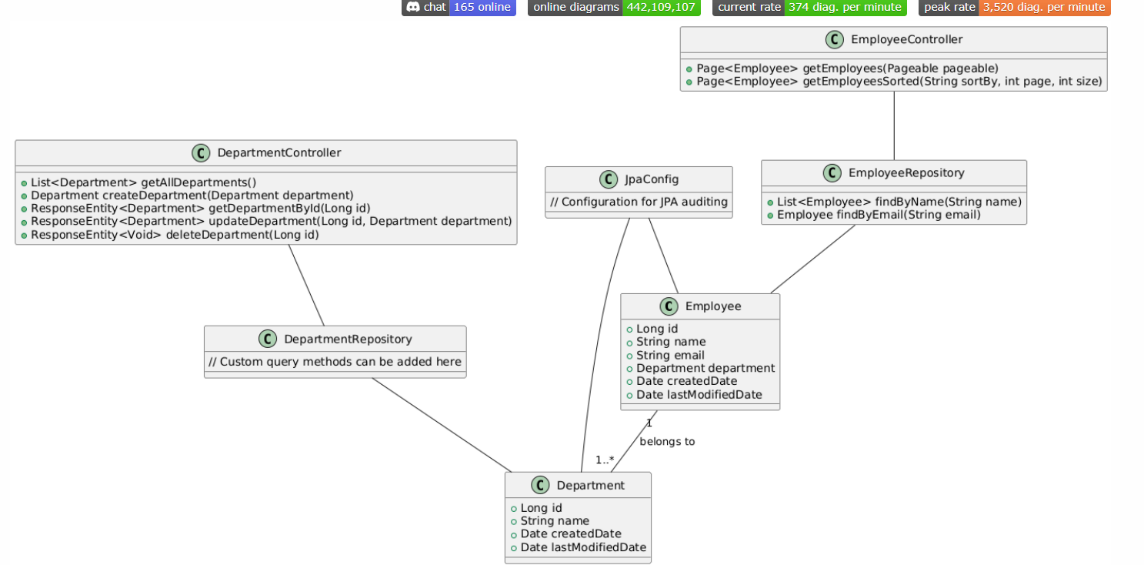
* **JpaConfig**: Configures JPA auditing for tracking the creation and modification timestamps of entities.

**FLOWCHART :**  
 

**Explanation**

* **Start**: The entry point of the system.
* **Home Controller**: Handles the root endpoint and provides a welcome message.
* **Employee Controller**: Manages requests related to employees, including CRUD operations and sorting.
* **Department Controller**: Manages requests related to departments, including CRUD operations.
* **Repository Layer**: Interacts with the database to perform CRUD operations on Employee and Department entities.
* **End**: Represents the conclusion of the flow.

**CLASS DIAGRAM :**



**Explanation**

* **Employee**: Represents an employee with fields like id, name, email, and a reference to a Department.
* **Department**: Represents a department with fields like id, name, and auditing fields (createdDate, lastModifiedDate).
* **EmployeeRepository**: Repository interface for Employee with custom query methods.
* **DepartmentRepository**: Repository interface for Department.
* **EmployeeController**: Manages HTTP requests related to employees, including pagination and sorting.
* **DepartmentController**: Manages HTTP requests related to departments, including CRUD operations.
* **JpaConfig**: Configuration class for enabling JPA auditing.

**Relationships**:

* An Employee belongs to a Department.
* EmployeeRepository and DepartmentRepository interact with their respective entities.
* EmployeeController and DepartmentController interact with their respective repositories.
* JpaConfig is linked to both Employee and Department for auditing.