**EXERCISE 4**

**1. Basic CRUD Operations with JpaRepository**

**JpaRepository:** The JpaRepository interface extends the CrudRepository interface and provides methods such as save(), findById(), findAll(), and deleteById(), among others. These methods allow us to perform basic CRUD operations on the database without writing boilerplate code.

**Service Layer**

The service layer acts as an intermediary between the controllers and the repositories. It contains the business logic for managing employees and departments.

**EmployeeService.java**

* saveEmployee(Employee employee): This method uses the save() method of JpaRepository to create a new employee or update an existing one.
* getEmployeeById(Long id): This method retrieves an employee by its ID using the findById() method.
* getAllEmployees(): This method retrieves all employees using the findAll() method.
* deleteEmployee(Long id): This method deletes an employee by its ID using the deleteById() method.

**DepartmentService.java**

* saveDepartment(Department department): Similar to EmployeeService, this method uses the save() method to create or update a department.
* getDepartmentById(Long id): This method retrieves a department by its ID.
* getAllDepartments(): This method retrieves all departments.
* deleteDepartment(Long id): This method deletes a department by its ID.

**Controller Layer**

The controller layer handles HTTP requests and maps them to the appropriate service methods.

**EmployeeController.java**

* createOrUpdateEmployee(@RequestBody Employee employee): Handles POST requests to create or update an employee. The employee data is sent in the request body, and the method calls saveEmployee() in the service layer.
* getEmployeeById(@PathVariable Long id): Handles GET requests to retrieve an employee by ID. It returns a ResponseEntity that wraps the employee data or a 404 status if not found.
* getAllEmployees(): Handles GET requests to retrieve all employees.
* deleteEmployee(@PathVariable Long id): Handles DELETE requests to remove an employee by ID. It returns a 204 status if successful.

**DepartmentController.java**

* createOrUpdateDepartment(@RequestBody Department department): Similar to EmployeeController, this method handles POST requests for departments.
* getDepartmentById(@PathVariable Long id): Handles GET requests to retrieve a department by ID.
* getAllDepartments(): Handles GET requests to retrieve all departments.
* deleteDepartment(@PathVariable Long id): Handles DELETE requests to remove a department by ID.

**NOTE:-**

**Repositories:** EmployeeRepository and DepartmentRepository extend JpaRepository to provide CRUD operations for Employee and Department entities.

**Service Classes:** EmployeeService and DepartmentService encapsulate business logic and interact with the repositories.

**Controller Classes:** EmployeeController and DepartmentController provide RESTful endpoints that expose CRUD operations via HTTP methods.