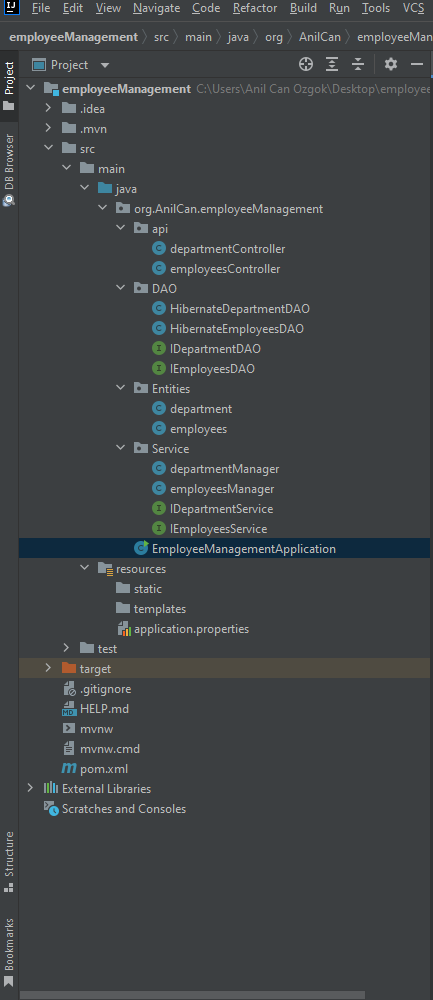
**Layers:**



**API LAYER**

**//empAPI**

package org.AnilCan.employeeManagement.api;  
  
import org.AnilCan.employeeManagement.Entities.employees;  
import org.AnilCan.employeeManagement.Service.IEmployeesService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/empAPI")  
public class employeesController {  
  
 private IEmployeesService employeesService;  
  
 @Autowired  
 public employeesController(IEmployeesService employeesService) {  
 this.employeesService = employeesService;  
 }  
  
 @GetMapping("/employees")  
 public List<employees> get(){  
 return employeesService.getAllData();  
 }  
  
 @PostMapping("/add")  
 public void add(@RequestBody employees emp){  
 this.employeesService.add(emp);  
 }  
  
 @PostMapping("/update")  
 public void update(@RequestBody employees emp){  
 this.employeesService.update(emp);  
 }  
  
 @PostMapping("/delete")  
 public void delete(@RequestBody employees emp){  
 this.employeesService.delete(emp);  
 }  
  
 @GetMapping("/employees/{id}")  
 public employees getByID(@PathVariable int id){  
 return this.employeesService.getByID(id);  
 }  
}

**//depAPI Layer**

package org.AnilCan.employeeManagement.api;  
  
import org.AnilCan.employeeManagement.Entities.department;  
import org.AnilCan.employeeManagement.Service.IDepartmentService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/depAPI")  
public class departmentController {  
  
 private IDepartmentService departmentService;  
  
 @Autowired  
 public departmentController(IDepartmentService departmentService) {  
 this.departmentService = departmentService;  
 }  
  
 @GetMapping("/department")  
 public List<department> get(){  
 return this.departmentService.getAllData();  
 }  
  
 @PostMapping("/add")  
 public void add(@RequestBody department dep){  
 this.departmentService.add(dep);  
 }  
  
 @PostMapping("/update")  
 public void update(@RequestBody department dep){  
 this.departmentService.update(dep);  
 }  
  
 @PostMapping("/delete")  
 public void delete(@RequestBody department dep){  
 this.departmentService.delete(dep);  
 }  
  
 @GetMapping("/department/{id}")  
 public department getByID(@PathVariable int id){  
 return this.departmentService.getByID(id);  
 }  
  
}

**DAO(DATA ACCESS OPERATIONS) LAYER**

**//HibernateDepartmentDAO**

package org.AnilCan.employeeManagement.DAO;  
  
import org.AnilCan.employeeManagement.Entities.department;  
import org.hibernate.Session;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Repository;  
import javax.persistence.EntityManager;  
import javax.transaction.Transactional;  
import java.util.List;  
  
@Repository  
public class HibernateDepartmentDAO implements IDepartmentDAO{  
  
 private EntityManager entityManager;  
  
 @Autowired  
 public HibernateDepartmentDAO(EntityManager entityManager) {  
 this.entityManager = entityManager;  
 }  
  
  
 @Transactional//beginTransaction ve endTransaction kodlarını otomatik olarak enjekte eder.  
 @Override  
 public List<department> getAllData() {  
 Session session=entityManager.unwrap(Session.class);  
 List<department> dep=session.createQuery("from department",department.class).getResultList();  
 return dep;  
 }  
   
 @Override  
 public void add(department dep) {  
 Session session=entityManager.unwrap(Session.class);  
 session.save(dep);  
 }  
  
 @Override  
 public void update(department dep) {  
 Session session=entityManager.unwrap(Session.class);  
 session.update(dep);  
 }  
  
 @Override  
 public void delete(department dep) {  
 Session session=entityManager.unwrap(Session.class);  
 department depDel=session.get(department.class,dep.getId());  
 session.delete(depDel);  
 }  
  
 @Override  
 public department getByID(int id) {  
 Session session=entityManager.unwrap(Session.class);  
 department dep=session.get(department.class,id);  
 return dep;  
 }  
}

**//HibernateEmployeesDAO**

package org.AnilCan.employeeManagement.DAO;  
  
import org.AnilCan.employeeManagement.Entities.employees;  
import org.hibernate.Session;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Repository;  
  
import javax.persistence.EntityManager;  
import javax.transaction.Transactional;  
import java.util.List;  
  
@Repository  
public class HibernateEmployeesDAO implements IEmployeesDAO{  
  
 private EntityManager entityManager;  
  
 @Autowired  
 public HibernateEmployeesDAO(EntityManager entityManager) {  
 this.entityManager = entityManager;  
 }  
  
 @Transactional  
 @Override  
 public List<employees> getAllData() {  
 Session session=entityManager.unwrap(Session.class);  
 List<employees> emp=session.createQuery("from employees",employees.class).getResultList();  
 return emp;  
 }  
  
 @Transactional  
 @Override  
 public void add(employees emp) {  
 Session session=entityManager.unwrap(Session.class);  
 session.saveOrUpdate(emp);  
 }  
  
 @Transactional  
 @Override  
 public void update(employees emp) {  
 Session session=entityManager.unwrap(Session.class);  
 session.saveOrUpdate(emp);  
 }  
  
 @Transactional  
 @Override  
 public void delete(employees emp) {  
 Session session=entityManager.unwrap(Session.class);  
 employees empDel=session.get(employees.class,emp.getId());  
 session.delete(empDel);  
 }  
  
 @Override  
 public employees getByID(int id) {  
 Session session=entityManager.unwrap(Session.class);  
 employees emp=session.get(employees.class,id);  
 return emp;  
 }  
}

**//IDepartmentDAO**

package org.AnilCan.employeeManagement.DAO;  
import org.AnilCan.employeeManagement.Entities.department;  
  
import java.util.List;  
  
public interface IDepartmentDAO {  
  
 List<department> getAllData();  
  
 void add(department dep);  
  
 void update(department dep);  
  
 void delete(department dep);  
  
 department getByID(int id);  
  
}

**//IEmployeesDAO**

package org.AnilCan.employeeManagement.DAO;  
import org.AnilCan.employeeManagement.Entities.employees;  
  
import java.util.List;  
  
  
public interface IEmployeesDAO {  
  
 List<employees> getAllData();

void add(employees emp);

void update(employees emp);

void delete(employees emp);

employees getByID(int id);

}

**ENTITY LAYER**

**//department**

package org.AnilCan.employeeManagement.Entities;  
  
import org.AnilCan.employeeManagement.DAO.IDepartmentDAO;  
import javax.persistence.\*;  
import java.util.List;  
  
//annotation bazlı injection  
@Entity  
@Table(name="department")  
public class department implements IDepartmentDAO {  
  
 //default empty constructor for hibernate  
 public department(){}  
  
 @Id  
 @Column(name="dep\_id")  
 private int id;  
  
 @Column(name="dep\_name")  
 private String name;  
  
 //Consturctor injection  
 public department(int id, String name) {  
 this.id = id;  
 this.name = name;  
 }  
  
//getters and setters  
 public int getId() { return id; }  
  
 public void setId(int id) { this.id = id; }  
  
 public String getName() { return name;}  
  
 public void setName(String name) { this.name = name; }  
@Override  
 public List<department> getAllData() { return null; }

//unit of work design patterninin uygulanması  
 @Override  
 public void add(department dep) {}

@Override  
 public void update(department dep) {}  
  
 @Override  
 public void delete(department dep) {}  
  
 @Override  
 public department getByID(int id) { return null; }

}

**//employees**

package org.AnilCan.employeeManagement.Entities;  
  
  
import org.AnilCan.employeeManagement.DAO.IEmployeesDAO;  
import javax.persistence.\*;  
import java.util.List;  
  
//annotation bazlı injection  
@Entity  
@Table(name="employees")//Hibernate ile verileri nesnelerle mapleme  
public class employees implements IEmployeesDAO {  
  
 //default empty constructor for hibernate  
 public employees(){ }  
  
 //Hibernate ile verileri nesnelerle mapleme  
 @Id  
 @Column(name="emp\_id")  
 private int id;  
  
 @Column(name="emp\_name")  
 private String name;  
  
 @Column(name = "emp\_lastname")//camel casing ile hatalı çalışıyor.  
 private String lastName;  
  
 @Column(name="mail")  
 private String mail;  
  
 @Column(name="phone")  
 private String phone;  
  
 @Column(name="salary")  
 private int salary;  
  
 @Column(name="manager\_id")  
 private int manager\_id;  
  
 @Column(name="dep\_id")  
 private int dep\_id;  
  
  
  
 //Consturctor injection  
 public employees(int id, int salary, int manager\_id, int dep\_id, String name, String lastName, String mail, String phone) {  
 this.id = id;  
 this.salary = salary;  
 this.manager\_id = manager\_id;  
 this.dep\_id = dep\_id;  
 this.name = name;  
 this.lastName = lastName;  
 this.mail = mail;  
 this.phone = phone;  
 }

**//employees**

//getters and setters  
 public int getId() { return id; }  
  
 public void setId(int id) { this.id = id; }  
  
 public String getName() { return name; }  
  
 public void setName(String name) { this.name = name; }  
  
 public String getLastName() { return lastName; }  
  
 public void setLastName(String lastName) { this.lastName = lastName; }  
  
 public String getMail() { return mail; }  
  
 public void setMail(String mail) { this.mail = mail; }  
  
 public String getPhone() { return phone; }  
  
 public void setPhone(String phone) { this.phone = phone; }  
  
 public int getSalary() { return salary; }  
  
 public void setSalary(int salary) { this.salary = salary; }  
  
 public int getManager\_id() { return manager\_id; }  
  
 public void setManager\_id(int manager\_id) {

this.manager\_id = manager\_id;  
 }  
  
 public int getDep\_id() { return dep\_id; }  
  
 public void setDep\_id(int dep\_id) { this.dep\_id = dep\_id; }  
  
  
 @Override  
 public List<employees> getAllData() { return null; }  
  
 @Override  
 public void add(employees emp) {}  
  
 @Override  
 public void update(employees emp) {}  
  
 @Override  
 public void delete(employees emp) {}  
  
 @Override  
 public employees getByID(int id) { return null;}

}

**SERVICE-BUISNESS LAYER**

**//departmentManager**

package org.AnilCan.employeeManagement.Service;  
  
import org.AnilCan.employeeManagement.DAO.IDepartmentDAO;  
import org.AnilCan.employeeManagement.Entities.department;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import javax.transaction.Transactional;  
import java.util.List;  
  
@Service  
public class departmentManager implements IDepartmentService{  
  
  
 private IDepartmentDAO departmentDAO;  
  
 @Autowired  
 public departmentManager(IDepartmentDAO departmentDAO) {  
 this.departmentDAO = departmentDAO;  
 }  
  
 @Transactional  
 @Override  
 public List<department> getAllData() {  
 return this.departmentDAO.getAllData();  
 }  
  
 @Transactional  
 @Override  
 public void add(department dep) {  
 this.departmentDAO.add(dep);  
 }  
  
 @Transactional  
 @Override  
 public void update(department dep) {  
 this.departmentDAO.update(dep);  
 }  
  
 @Transactional  
 @Override  
 public void delete(department dep) {  
 this.departmentDAO.delete(dep);  
 }  
  
 @Override  
 public department getByID(int id) {  
 return this.departmentDAO.getByID(id);  
 }  
}

**//employeesManager**

package org.AnilCan.employeeManagement.Service;  
  
import org.AnilCan.employeeManagement.DAO.IEmployeesDAO;  
import org.AnilCan.employeeManagement.Entities.employees;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import javax.transaction.Transactional;  
import java.util.List;  
  
@Service  
public class employeesManager implements IEmployeesService{  
  
 private IEmployeesDAO employeesDAO;  
  
 @Autowired  
 public employeesManager(IEmployeesDAO employeesDAO) {  
 this.employeesDAO = employeesDAO;  
 }  
  
 @Transactional  
 @Override  
 public List<employees> getAllData() {  
 return this.employeesDAO.getAllData();  
 }  
  
 @Transactional  
 @Override  
 public void add(employees emp) {  
 this.employeesDAO.add(emp);  
 }  
  
 @Transactional  
 @Override  
 public void update(employees emp) {  
 this.employeesDAO.update(emp);  
 }  
  
 @Transactional  
 @Override  
 public void delete(employees emp) {  
 this.employeesDAO.delete(emp);  
 }  
  
 @Override  
 public employees getByID(int id) {  
 return this.employeesDAO.getByID(id);  
 }  
}

**//IDepartmentService**

package org.AnilCan.employeeManagement.Service;  
  
import org.AnilCan.employeeManagement.Entities.department;  
  
import java.util.List;  
  
public interface IDepartmentService {  
  
 List<department> getAllData();

void add(department dep);

void update(department dep);

void delete(department dep);

department getByID(int id);

}

**//IEmployeesService**

package org.AnilCan.employeeManagement.Service;  
  
import org.AnilCan.employeeManagement.Entities.employees;  
  
import java.util.List;  
  
public interface IEmployeesService {

List<employees> getAllData();

void add(employees emp);

void update(employees emp);

void delete(employees emp);

employees getByID(int id);

}