@Entity

@Table(name = "users")

@Data

@AllArgsConstructor

@NoArgsConstructor

**public class User** {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private long id;

private String fullName;

private String email;

private String password;

private String address;

private String gender;

private Date dateOfBirth;

private String nationalId;

private String contact;

private double basicSalary;

private Date joinedDate;

private boolean isActive;

private Date createdAt;

private Date updatedAt;

private String profilePhoto;

@Enumerated(EnumType.STRING)

private Role role;

@ManyToOne(fetch = FetchType.EAGER)

@JoinColumn(name = "departmentId")

private Department department;

@ManyToOne(fetch = FetchType.EAGER)

@JoinColumn(name = "branchId")

private Branch branch;

}  
  
  
@Entity

@Table(name = "salaries")

@Data

@AllArgsConstructor

@NoArgsConstructor

public class Salary {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private long id;

private double medicare;

private boolean overTime;

private double fund;

private double netSalary;

private Date paymentDate;

private double insurance;

@ManyToOne(fetch = FetchType.EAGER)

@JoinColumn(name = "userId")

private User user;

@OneToOne(fetch = FetchType.EAGER)

@JoinColumn(name = "advancesalaryId")

private AdvanceSalary advanceSalary;

@OneToOne(fetch = FetchType.EAGER)

@JoinColumn(name = "bonusId")

private Bonus bonus;

}

@Entity

@Table(name = "branches")

@Data

@AllArgsConstructor

@NoArgsConstructor

public class Branch {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private long id;

private String name;

private String address;

private String city;

private String zipCode;

private String country;

private Date createdAt;

private Date updatedAt;

private String branchPhoto;

@ManyToOne(fetch = FetchType.EAGER)

@JoinColumn(name = "companyId")

private Company company;

@OneToMany(mappedBy = "branch", fetch = FetchType.LAZY)

private List<User> users;

}

@Entity

@Table(name = "departments")

@Data

@AllArgsConstructor

@NoArgsConstructor

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private long id;

private String name;

private String numberOfEmployees;

@ManyToOne(fetch = FetchType.EAGER)

@JoinColumn(name = "branchId")

private Branch branch;

@ManyToOne(fetch = FetchType.EAGER)

@JoinColumn(name = "companyId")

private Company company;

}

package com.mrahmed.HRandPayrollManagementSystem.entity;

import jakarta.persistence.\*;

import lombok.\*;

import java.util.List;

@Entity

@Data

@AllArgsConstructor

@NoArgsConstructor

public class Company {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

@OneToMany(mappedBy = "company", cascade = CascadeType.ALL, orphanRemoval = true, fetch = FetchType.EAGER)

private List<Branch> branches;

@OneToMany(mappedBy = "company", cascade = CascadeType.ALL, orphanRemoval = true, fetch = FetchType.EAGER)

private List<Department> departments;

@OneToMany(mappedBy = "company", cascade = CascadeType.ALL, orphanRemoval = true, fetch = FetchType.EAGER)

private List<User> users;

}