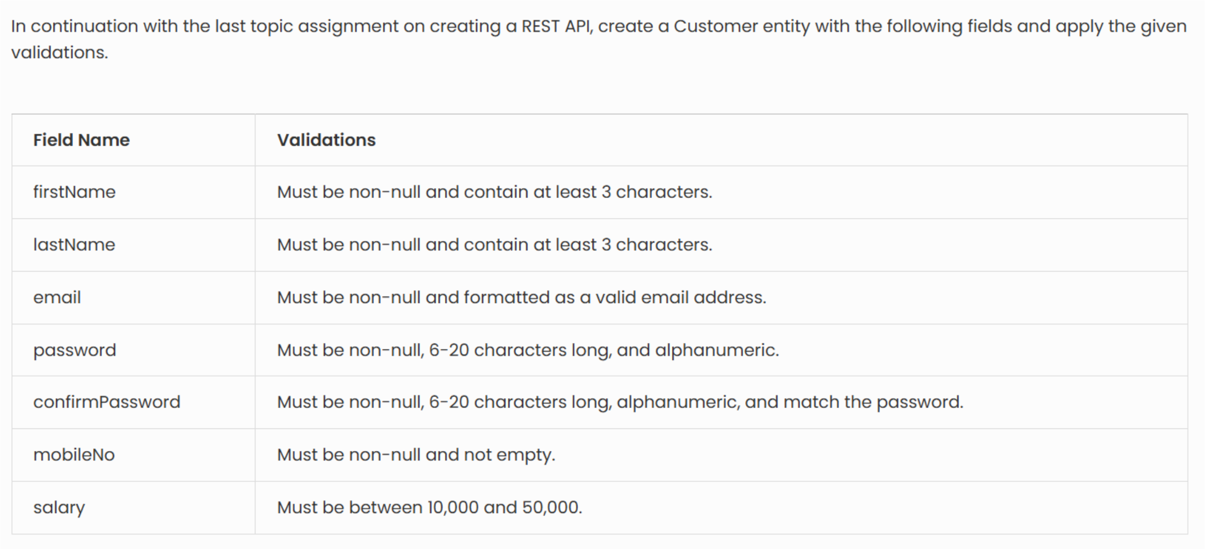


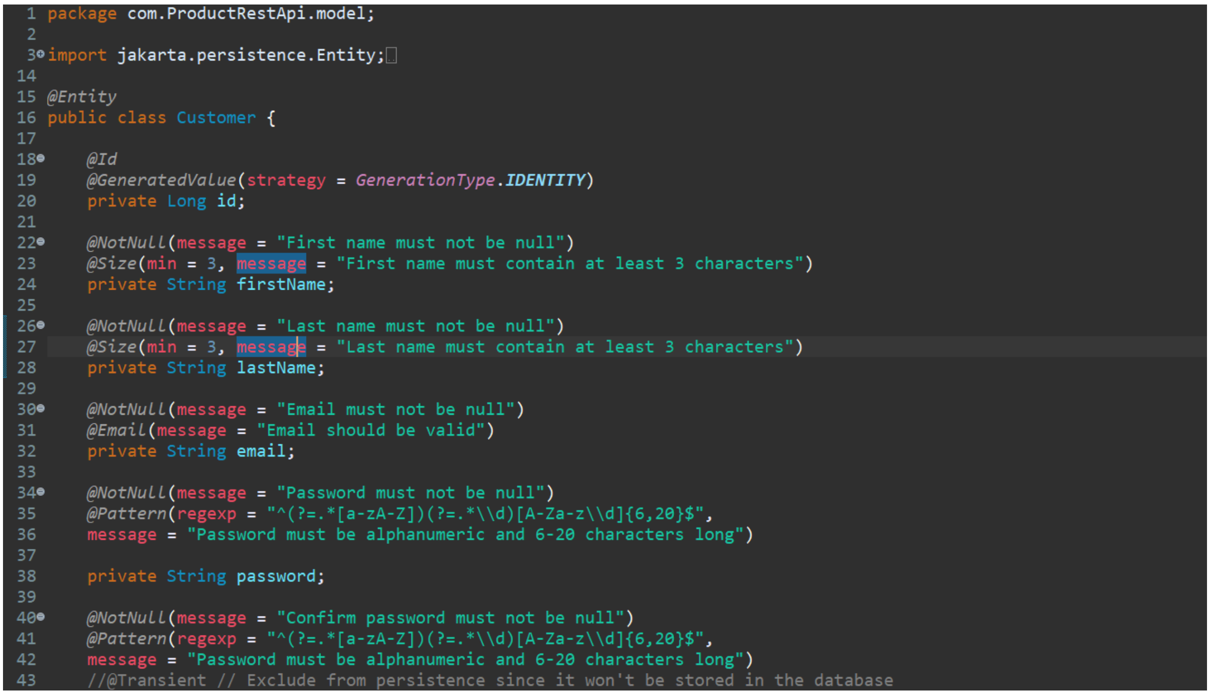
In continuation with the last topic assignment on creating a REST API, create a Customer entity with the following fields and apply the given validations.

Field Name	Validations
firstName	Must be non-null and contain at least 3 characters.
lastName	Must be non-null and contain at least 3 characters.
email	Must be non-null and formatted as a valid email address.
password	Must be non-null, 6-20 characters long, and alphanumeric.
confirmPassword	Must be non-null, 6-20 characters long, alphanumeric, and match the password.
mobileNo	Must be non-null and not empty.
salary	Must be between 10,000 and 50,000.

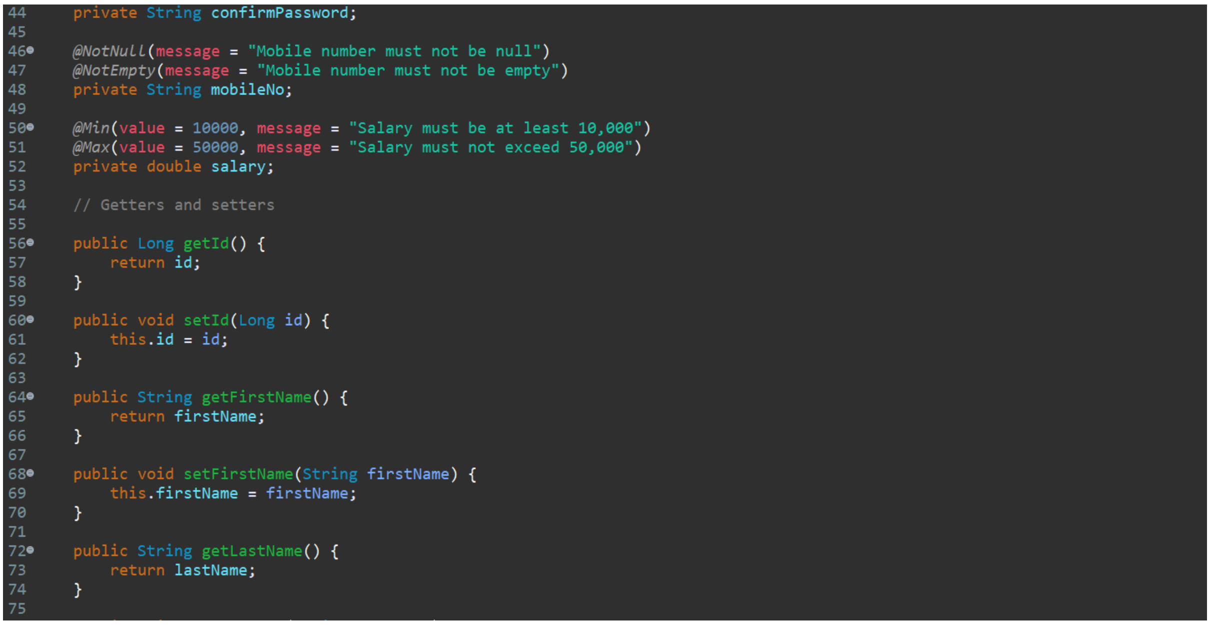
Ans –

Code –

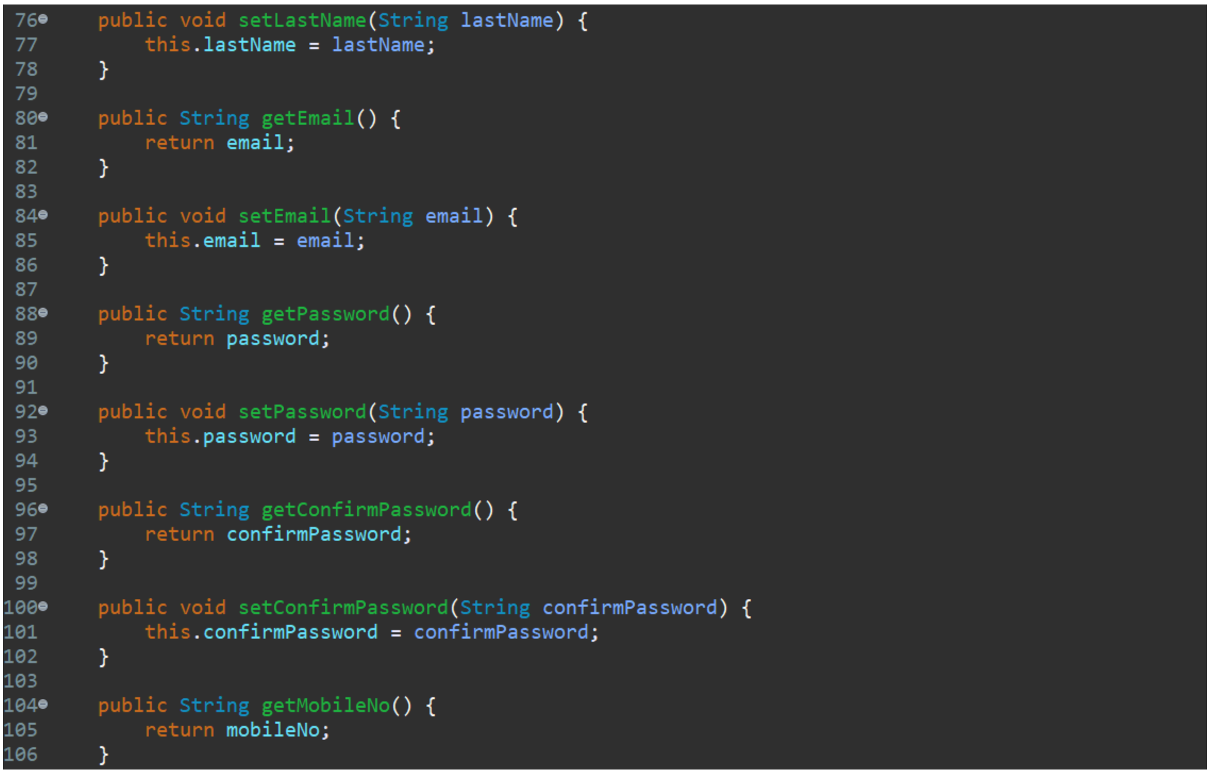
Customer.java (Entity)



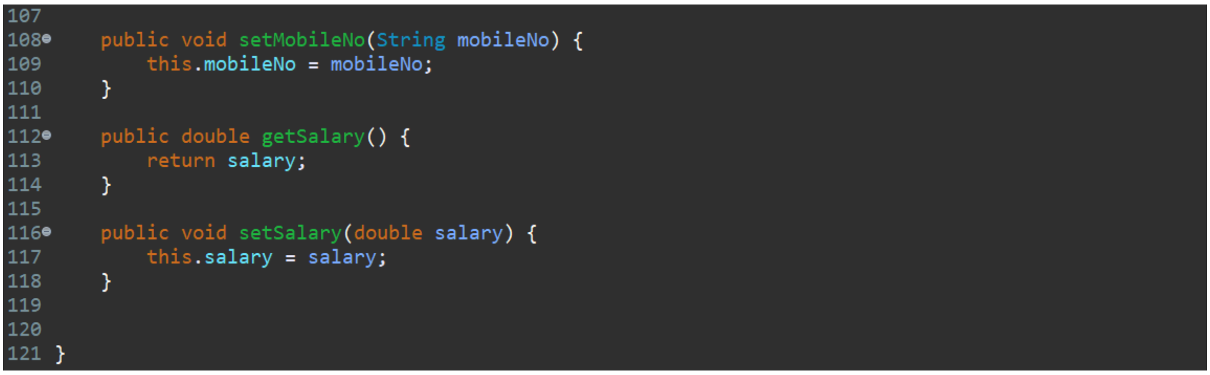
```java
package com.ProductRestApi.model;

import jakarta.persistence.Entity;

@Entity
public class Customer {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotNull(message = "First name must not be null")
    @Size(min = 3, message = "First name must contain at least 3 characters")
    private String firstName;

    @NotNull(message = "Last name must not be null")
    @Size(min = 3, message = "Last name must contain at least 3 characters")
    private String lastName;

    @NotNull(message = "Email must not be null")
    @Email(message = "Email should be valid")
    private String email;

    @NotNull(message = "Password must not be null")
    @Pattern(regexp = "^(?=.*[a-zA-Z])(?=.*\\d)[A-Za-z\\d]{6,20}$",
    message = "Password must be alphanumeric and 6-20 characters long")

    private String password;

    @NotNull(message = "Confirm password must not be null")
    @Pattern(regexp = "^(?=.*[a-zA-Z])(?=.*\\d)[A-Za-z\\d]{6,20}$",
    message = "Password must be alphanumeric and 6-20 characters long")
    //@Transient // Exclude from persistence since it won't be stored in the database
```



```java
    private String confirmPassword;

    @NotNull(message = "Mobile number must not be null")
    @NotEmpty(message = "Mobile number must not be empty")
    private String mobileNo;

    @Min(value = 10000, message = "Salary must be at least 10,000")
    @Max(value = 50000, message = "Salary must not exceed 50,000")
    private double salary;

    // Getters and setters

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

```



```java
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getConfirmPassword() {
        return confirmPassword;
    }

    public void setConfirmPassword(String confirmPassword) {
        this.confirmPassword = confirmPassword;
    }

    public String getMobileNo() {
        return mobileNo;
    }
```



```java

    public void setMobileNo(String mobileNo) {
        this.mobileNo = mobileNo;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }


}
```