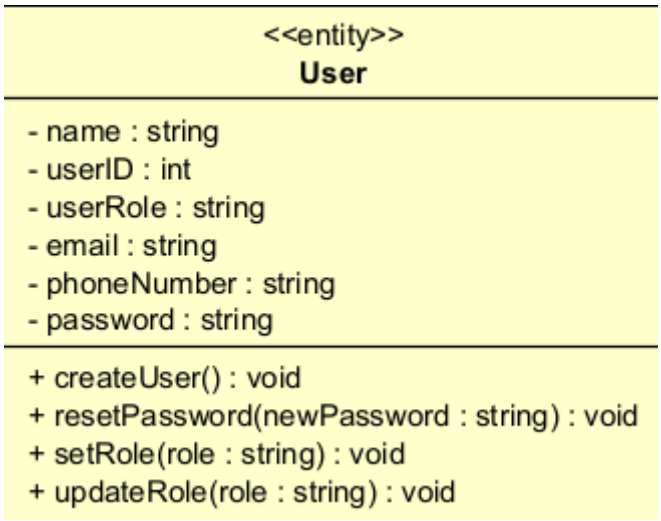
Design for class “User”



	Name	Data Type	Default value	Description
1	name	string		The name of the user
2	userID	int		A unique identifier for the user
3	userRole	string		The role of the user
4	email	string		The email address of the user
5	phoneNumber	string		The phone number of the user
6	password	string		The password for the user's account

Table 1. Example of attribute design

	Name	Return Type	Description
1	createUser		Creates a new user with provided attributes
2	setRole		Sets the role of the user
3	resetPassword		Resets the user's password
4	updateRole		Updates the user's role

Table 2. Example of operation design

Method

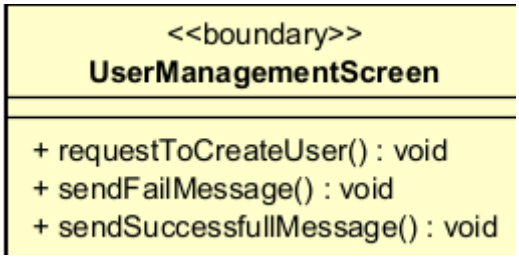
```
// Method to create a user
   public void createUser(String name, String role, String email, String
phoneNumber, String password) {
        this.name = name;
        this.userRole = role;
        this.email = email;
```

```
        this.phoneNumber = phoneNumber;
        this.password = password;
        System.out.println("User created successfully.");
    }

    // Method to set user role
    public void setRole(String role) {
        this.userRole = role;
        System.out.println("Role set to " + role);
    }

    // Method to reset user password
    public void resetPassword(String newPassword) {
        this.password = newPassword;
        System.out.println("Password reset successfully.");
    }

    // Method to update user role
    public void updateRole(String role) {
        this.userRole = role;
        System.out.println("User role updated to " + role);
    }
```

How to use parameters / attributes

```
private String name;
    private int userID;
    private String userRole;
    private String email;
    private String phoneNumber;
    private String password;

    // Constructor
    public User(String name, String userRole, String email, String phoneNumber,
String password) {
        this.name = name;
        this.userRole = userRole;
        this.email = email;
        this.phoneNumber = phoneNumber;
        this.password = password;
    }
```

Design for class “UserManagementScreen”



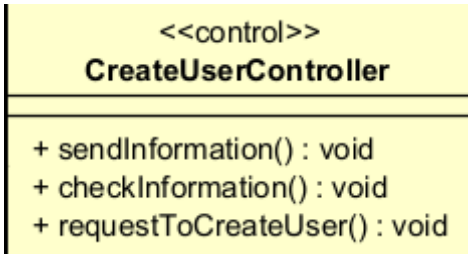
	Name	Return Type	Description
1	**requestToCreateUser**		**Requests the creation of a new user**
2	**sendFailMessage**		**Sends a failure message in case of errors**
3	**sendSuccessfullMessage**		Sends a success message when user creation succeeds

Table 2. Example of operation design

Method

```
public void requestToCreateUser() {
        System.out.println("Request to create a new user...");
        // Logic to collect user input and send it to controller
    }

    // Method to send failure message
    public void sendFailMessage() {
        System.out.println("User creation failed. Please try again.");
    }

    // Method to send success message
    public void sendSuccessfulMessage() {
        System.out.println("User created successfully!");
    }
```

Design for class “CreateUserController”



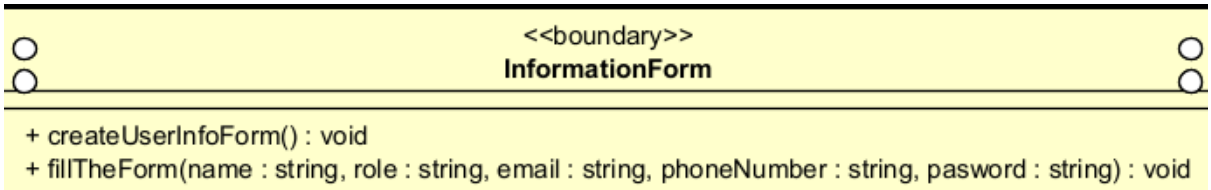
	Name	Return Type	Description
1	sendInformation		Sends the information related to user creation
2	checkInformation		Checks the information entered by the user
3	requestToCreateUser		Requests the creation of a user after verification

Table 2. Example of operation design

Method

```
// Method to send user information to be processed
   public void sendInformation(User user) {
      System.out.println("Sending user information to create user...");
      // Logic to send information to the User class for processing
   }

   // Method to check the information provided by the user
   public void checkInformation(String name, String email, String phoneNumber,
String password) {
      if (name.isEmpty() || email.isEmpty() || phoneNumber.isEmpty() ||
password.isEmpty()) {
         System.out.println("All fields are required.");
      } else {
         System.out.println("User information is valid.");
      }
   }

   // Method to request user creation
   public void requestToCreateUser(User user) {
      // Validate the information before passing to createUser
      checkInformation(user.getName(), user.getEmail(), user.getPhoneNumber(),
user.getPassword());
      sendInformation(user);
   }
```

Design for class “InformationForm”



	Name	Return Type	Description
1	createUserInfoForm		Creates the user information form
2	fillTheForm		Fills the information form with user data

Table 2. Example of operation design

Parameter

Name	Default Value	Description
name		The name of the user
role		The role of the user
email		The email address of the user
phoneNumber		The phone number of the user
password		The password for the user's account

Method

```
public void createUserInfoForm() {
        System.out.println("Creating user information form...");
        // Logic to create the form (e.g., GUI, console input)
    }

    // Method to fill out the form with user data
    public void fillTheForm(String name, String email, String phoneNumber, String
password) {
        System.out.println("Filling the form with user data...");
        // Logic to set the values in the form (could be filling in a GUI or console-based
inputs)
        System.out.println("User Name: " + name);
        System.out.println("Email: " + email);
        System.out.println("Phone Number: " + phoneNumber);
        System.out.println("Password: " + password);
    }
```

How to use parameters / attributes

```
    private String name;
    private int userID;
    private String userRole;
    private String email;
    private String phoneNumber;
    private String password;
```