Co
:

```
package buffer;


public class UserData {
    private String name;
    private String email;
    private String phone;
    public UserData(String name, String email, String phone)
{
        super();
        this.name = name;
        this.email = email;
        this.phone = phone;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getEmail() {
```

```
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
}


//AuthenticationCredential Class :
package buffer;

public class AuthenticationCredential {

    String email;
    String password;
```

```
    public AuthenticationCredential(String email, String
password) {
        super();
        this.email = email;
        this.password = password;
    }

    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 Object getHashedValue() {
        return null;
    }
}

//DigitalIdentityManagementSystem Class :
package buffer;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.Scanner;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import java.util.regex.*;

public class DigitalIdentityManagementSystem {
```

```
    // User data storage
    private Map<String, UserData> userDataTable;

    // Authentication credential storage
    private Map<String, AuthenticationCredential>
authCredentialTable;

    // Permissions management
    private Map<String, List<String>> resourcePermissions;

    public DigitalIdentityManagementSystem() {
        userDataTable = new HashMap<>();
        authCredentialTable = new HashMap<>();
        resourcePermissions = new HashMap<>();
    }

    //  sign up
    public void registerUser(String name, String email,
String phone, String password) {
        // Check if the user already exists
        if (userDataTable.containsKey(email)) {
            System.out.println("User already exists.");
```

```
            return;

        }

        UserData userData = new UserData(name, email,
phone);



        // Store the user data in the user data table

        userDataTable.put(email, userData);

        // Hash the password and store the authentication
credential in the auth-credential table



        AuthenticationCredential authCredential = new
AuthenticationCredential(email, password);

        authCredentialTable.put(email, authCredential);

        System.out.println("User registered successfully.");

    }



    // Log in an existing user

    public boolean login(String email, String password) {

        // Check if the user exists

        if (!userDataTable.containsKey(email)) {

            System.out.println("User does not exist.");

            return false;
```

```
        }

        // Retrieve the authentication credential from the
auth credential table

        AuthenticationCredential authCredential =
authCredentialTable.get(email);



        if
(!(authCredential.getPassword()).equals(password)) {

            System.out.println("Invalid password.");

            return false;

        }



        System.out.println("Login successful.");

        return true;

    }



    // Grant access to a resource---give it to manager,ceo



    public void grantAccess(String email, String
resourceName) {

        // Check if the user and resource exist

        if (!userDataTable.containsKey(email)) {

            System.out.println("User does not exist.");
```

```
            return;

        }

        if (!resourcePermissions.containsKey(resourceName))
{

            resourcePermissions.put(resourceName, new
LinkedList<>());

        }




        // Add the user to the linked list of users who have
permission to access the resource

        List<String> users =
resourcePermissions.get(resourceName);

        if (!users.contains(email)) {

            users.add(email);

        }

        System.out.println("Access granted to " +
resourceName + " for " + email);

    }



    private void permissons(String email){

        List<String>manager=new LinkedList<>();

        manager.add( "project Management");

        manager.add("grant-access");
```

```
        manager.add("Team management");

        manager.add("Resource allocation");////here we can
call grant access

        manager.add("Technical expertise");



        List<String>CEO=new LinkedList<>();

        CEO.add("grant-access");

        CEO.add( "Strategic planning");

        CEO.add("Financial management");

        CEO.add("Human resources");

        CEO.add("Stakeholder management");

        CEO.add("Public relations");



        List<String>HR=new LinkedList<>();

        HR.add( "Recruitment and staffing");

        HR.add("Employee onboarding and training");

        HR.add("Performance management and employee
development");

        HR.add("Employee relations");

        HR.add("Workplace policies and procedures");



        List<String>projectLead=new LinkedList<>();
```

```
        projectLead.add( "Team Management");

        projectLead.add("Planning and Strategy");

        projectLead.add("Budgeting and Resource");

        projectLead.add("Reporting of project");




        List<String>softwareEngineer=new LinkedList<>();

        softwareEngineer.add( "Requirements gathering");

        softwareEngineer.add("Design software
architecture");

        softwareEngineer.add("Development of program");

        softwareEngineer.add("Quality assurance");




        if(email.contains("manager")) {

            System.out.println("************ MANAGER SECTION
************");

            System.out.println("Manager has access to
following resources  :\n");

            for(int i=0;i<manager.size();i++) {


System.out.println((i+1)+"."+manager.get(i));

            }
```

```
        }


        else if(email.contains("ceo")) {

            System.out.println("************ CEO SECTION
************");

            System.out.println("CEO has access to following
resources  :\n");

            for(int i=0;i<CEO.size();i++) {

                System.out.println((i+1)+"."+CEO.get(i));

            }

        }


        else if(email.contains("hr")) {

            System.out.println("************ HR SECTION
************");

            System.out.println("HR has access to following
resources  :\n");

            for(int i=0;i<HR.size();i++) {

                System.out.println((i+1)+"."+HR.get(i));

            }

        }


        else if(email.contains("projectlead")) {
```

```
            System.out.println("************ PROJECT LEAD
************");

            System.out.println("ProjectLead has access to
following resources  :\n");

            for(int i=0;i<projectLead.size();i++) {


System.out.println((i+1)+"."+projectLead.get(i));

            }

        }



        else if(email.contains("sde")) {

            System.out.println("************ SDE SECTION
************");

            System.out.println("Software Engineer has access
to following resources  :\n");

            for(int i=0;i<softwareEngineer.size();i++) {


System.out.println((i+1)+"."+softwareEngineer.get(i));

            }

        }

        return;

    }



    // Main method
```

```
    public static void main(String[] args) {

        DigitalIdentityManagementSystem system = new
DigitalIdentityManagementSystem();

        Scanner sc = new Scanner(System.in);

        int choice=0;

        while (choice!=3) {



            System.out.println("************ WELCOME TO
PROBUILD SOFTWARE LIMITED COMPANY ************");

            System.out.println("\nEnter a command: \n1.sign
in\n2.login\n3.exit");

            choice=sc.nextInt();

            switch(choice) {

                case 1:

                    System.out.println("************ SIGN IN
************");

                    System.out.println("enter your name");

                    String name=sc.next();

                    System.out.println("enter your phone
number");

                    String phone =sc.next();

                    do{

                        if(phone.length()!=10) {
```

```
                            System.out.println("Invalid
Phone number\nPlease enter valid Phone number");

                            phone =sc.next();

                        }

                    }while(phone.length()!=10);

                    System.out.println("enter your email");

                    String empemail=sc.next();

                    do {

                        if ((!empemail.contains("@") ||
!empemail.contains(".com")) ||
(!empemail.contains("manager") && !empemail.contains("hr")
&& !empemail.contains("ceo") && !empemail.contains("SDE") &&
!empemail.contains("projectlead"))) {

                            System.out.println("Invalid
Email adrress\nPlease enter  valid  email address :");

                            empemail = sc.next();

                        }

                    }while((!empemail.contains("@") ||
!empemail.contains(".com")) ||
(!empemail.contains("manager") && !empemail.contains("hr")
&& !empemail.contains("ceo") && !empemail.contains("SDE") &&
!empemail.contains("projectlead")));



                    System.out.println("enter your
password");

                    String emppassword=sc.next();

                    do{
```

```
                        if(emppassword.length()<4 ||
emppassword.length()>15) {

                            System.out.println("Invalid
password\nPlease enter password having length between 4 to
15");

                            emppassword =sc.next();

                        }

                    }while(emppassword.length()<4 ||
emppassword.length()>15);



                    system.registerUser(name, empemail,
phone, emppassword);

                    break;

                case 2:

                    System.out.println("************ LOG IN
************");

                    System.out.println("enter your email");

                    String email=sc.next();

                    System.out.println("enter your
password");

                    String password=sc.next();

                    if(system.login(email,password)) {

                        system.permissons(email);

                    }
```

```
                    break;
                case 3:
                    System.out.println("Exiting...");
                    break;
                default:
                    break;
            }
        }
    }
}
```