```
/*---------------------------------------------------------------------------

    LAB IA

    Date: 20-5-2022

    Author: Meeth Sakaria
    USN: 2SD20CS054
    Div: A
    Sem: 4th

---------------------------------------------------------------------------*/

import java.text.SimpleDateFormat;
import java.util.*;

class Department {
    private String name;
    private int number;
    private Employee manager;
    public List<Project> projectdeppool = new ArrayList<Project>();
    public List<Employee> employeedeppool = new ArrayList<Employee>();
    public int b=0;
    public int w=0;

    public Department(String name,int number){
        this.name=name;
        this.number=number;
    }

    public void setName(String name){
        this.name=name;
    }

    public String getName(){
        return name;
    }

    public void setManager(Employee manager) {
        this.manager=manager;
    }

    public Employee getManager(){
        return manager;
    }

    public int getNumber(){
        return number;
    }

    public void assignprojectdep(Project p){
        projectdeppool.add(b,p);
        b++;
    }

    public void withdrawProjectDep(int f){
        projectdeppool.remove(f);
        b--;
```

```
    }

    public void assignempdep(Employee e){
        employeedeppool.add(w,e);
        w++;
    }

    public String toString(){
        return ("\n Department name: " + name +
                "\n Department number: " + number +
                "\n Manager: " + manager );
    }

    public String printProject(){
        String s="";
        for(Project p1 : projectdeppool)
        {
            s += p1.toString() + "\n";
        }
        return "\n\n The details of projects assigned to " + name + "
are: \n" + s;
    }

    public String printEmployee(){
        String s="";
        for(Employee e1 : employeedeppool)
        {
            s += e1.toString() + "\n";
        }
        return "\n\n The details of employee assigned to " + name + "
are: \n" + s;
    }
}

class Project{
    private String name;
    private String location;

    public Project(String name,String location){
        this.name=name;
        this.location=location;
    }

    public String getName(){
        return name;
    }

    public String getLocation(){
        return location;
    }

    public String toString(){
        return ("\n Project name: " + name +
                "\n Project location: " + location);
    }
}

class Employee{
```

```
    private String name;
    private int SSN;
    private String address;
    private float salary;
    private String gender;
    private Date DOB;
    public int man;
    public List<Project> projectemppool = new ArrayList<Project>();
    public List<Dependents> empdependents = new ArrayList<Dependents>();
    public int dep=0;
    public int c=0;

    public Employee(String name,int SSN,String address,float
salary,String gender,Date DOB){
        this.name=name;
        this.SSN=SSN;
        this.address=address;
        this.salary=salary;
        this.gender=gender;
        this.DOB=DOB;
    }

    public void setName(String name){
        this.name=name;
    }

    public void setSSN(int SSN){
        this.SSN=SSN;
    }

    public int getSSN(){
        return SSN;
    }

    public void setAddress(String address){
        this.address=address;
    }

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

    public void setgender(String gender){
        this.gender=gender;
    }

    public void setDOB(Date DOB){
        this.DOB=DOB;
    }

    public void assignprojectemp(Project p){
        projectemppool.add(c,p);
        c++;
    }

    public void withdrawProjectEmp(int f){
        projectemppool.remove(f);
        c--;
```

```
    }

    public void assigndepend(Dependents de){
        empdependents.add(dep,de);
        dep++;
    }

    public String toString(){
        return ("\n Employee SSN: " + SSN +
                "\n Employee name: " + name +
                "\n Employee address: " + address +
                "\n Employee Salary: " + salary +
                "\n Employee gender: " + gender +
                "\n Employee date of birth: " + DOB );
    }

    public String printProject(){
        String s="";
        for(Project p1 : projectemppool)
        {
            s +=p1.toString() + "\n";
        }
        return ("\n\n The details of projects assigned to " + name + "
are: \n" + s);
    }

    public String printDependents(){
        String s="";
        for(Dependents d1 : empdependents)
        {
            s += d1.toString() + "\n";
        }
        return ("\n\n The details of the dependents are: \n" + s);
    }
}

class Dependents {
    private String name;
    private String gender;
    private Date DOB;
    private String relationship;

    public Dependents(String name,String gender,Date DOB,String
relationship){
        this.name=name;
        this.gender=gender;
        this.DOB=DOB;
        this.relationship=relationship;
    }

    public String toString(){
        return ("\n Dependent's name: " + name +
                "\n Dependent's gender: " + gender +
                "\n Dependent's date of birth: " + DOB +
                "\n Dependent's relation: " + relationship + "\n");
    }
}
```

```
public class mainclass{
    private static int i=0,j=0,k=0;
    private static int empid;
    private static int depid;
    private static String empname;
    private static String depname;
    private static String proname;
    private static String dependname;
    private static Date empdate;
    private static Date dependdate;
    static int employeeindex=-1;
    static int departmentindex=-1;
    static int projectindex=-1;
    static int depprojectindex=-1;
    static int empprojectindex=-1;
    static int depemployeeindex=-1;
    private static List<Department> departmentpool = new
ArrayList<Department>();
    private static List<Employee> employeepool = new
ArrayList<Employee>();
    private static List<Project> projectpool = new ArrayList<Project>();
    static boolean projectfound=false;
    static boolean employeefound=false;
    static boolean departmentfound=false;
    static boolean depprojectfound=false;
    static boolean empprojectfound=false;
    static boolean depemployeefound=false;
    static Scanner in = new Scanner(System.in);

    public static void main(String[] args){
        int menuChoice=0;
        while(menuChoice!=8){
            try
            {
            System.out.println("\nMain Menu");
            System.out.println("------------------");
            System.out.println("\n1: Register department");
            System.out.println("2: Register and unregister employee");
            System.out.println("3: Register and unregister projects");
            System.out.println("4: Assign employee to department");
            System.out.println("5: Assign/withdraw projects to the
department");
            System.out.println("6: Assign/withdraw projects to the
employee");
            System.out.println("7: Details of Employees, Projects and
Departments");
            System.out.println("8: Exit\n");

            System.out.print("Enter an your option : ");
            menuChoice = in.nextInt();

                switch(menuChoice)
                {
                    case 1:{
                        registerDep();
                        break;
                    }
```

```
                case 2:{
                    registerEmp();
                    break;
                }

                case 3:{
                    registerPro();
                    break;
                }

                case 4:{
                    assignEmployeeToDepartment();
                    break;
                }

                case 5:{
                    assignProjectDep();
                    break;
                }

                case 6:{
                    assignProjectEmp();
                    break;
                }

                case 7:{
                    details();
                    break;
                }
            }
        }finally{}
    }
}

private static void registerDep(){
    int choice=0;
    while(choice!=4){
        System.out.println("\n1: Register department");
        System.out.println("2: Assign manager");
        System.out.println("3: Withdraw manager");
        System.out.println("4: Return to Main menu\n");
        System.out.print("Enter an your option : ");
        choice=in.nextInt();
        switch(choice){
            case 1:{
                registerDeparment();
                break;
            }
            case 2:{
                assignManager();
                break;
            }
            case 3:{
                withdrawManager();
                break;
            }
        }
    }
```

```
    }

    private static void registerDeparment(){
        System.out.print("\nEnter Department Id: ");
        depid = in.nextInt();

        if(isDepartment(depid) == true){
            System.out.println("\nDepartment Id already exist");
        }
        else {
            in.nextLine();

            System.out.print("Enter Department name: ");
            depname = in.nextLine();
            Department d = new Department(depname,depid);
            departmentpool.add(j,d);
            System.out.println("\nThe Department is created");
            j++;
        }
    }

    private static void assignManager(){
        System.out.print("\nEnter Department Id: ");
        depid = in.nextInt();

        if(isDepartment(depid) == true){
            System.out.print("Enter the employee Id for manager: ");
            empid=in.nextInt();

            if(isEmployee(empid) == true){
                Employee depmanager=employeepool.get(employeeindex);
                employeepool.get(employeeindex).man=departmentindex;

departmentpool.get(departmentindex).setManager(depmanager);
            }
            else {
                System.out.println("\nManager not assigned");
                System.out.println("Employee Id does not exist");
            }
        }
        else{
            System.out.println("Department Id does not exist");
        }
    }

    private static void withdrawManager(){
        System.out.print("\nEnter Department Id: ");
        depid = in.nextInt();

        if(isDepartment(depid) == true){
            Employee e =
departmentpool.get(departmentindex).getManager();
            isEmployee(e.getSSN());
            employeepool.get(employeeindex).man=0;
            departmentpool.get(departmentindex).setManager(null);
        }
        else {
            System.out.println("\nDepartment Id does not exist");
```

```
    }
}

private static void registerEmp(){
    int choice=0;
    while(choice!=4){
        System.out.println("\n1: Register employee");
        System.out.println("2: Unregister employee");
        System.out.println("3: Assign dependents of employee");
        System.out.println("4: Return to Main menu\n");
        System.out.print("Enter an your option : ");
        choice=in.nextInt();

        switch(choice){
            case 1:{
                registerEmployee();
                break;
            }
            case 2:{
                unregisterEmploye();
                break;
            }
            case 3:{
                assignDependent();
                break;
            }
        }
    }
}

private static void registerEmployee(){
    System.out.print("\nEnter Employee Id: ");
    empid = in.nextInt();

    if(isEmployee(empid) == true){
        System.out.println("\nEmployee Id already exist");
    }
    else {
        in.nextLine();

        System.out.print("Enter Employee name: ");
        empname = in.nextLine();

        System.out.print("Enter address: ");
        String empaddress = in.nextLine();

        System.out.print("Enter salary: ");
        float empsalary = in.nextFloat();

        in.nextLine();

        System.out.print("Enter Gender: ");
        String empgender = in.nextLine();

        System.out.print("Enter Date of birth: ");
        String empdob = in.nextLine();

        try {
```

```
                empdate=new SimpleDateFormat("dd/MM/yyyy").parse(empdob);
            }catch (Exception e) {
                System.out.println("Invalid Date of birth");
                System.exit(0);
            }

            Employee e1 = new
Employee(empname,empid,empaddress,empsalary,empgender,empdate);
            employeepool.add(i,e1);
            System.out.println("\nThe employee is created");
            i++;
        }
    }

    private static void unregisterEmploye(){
        System.out.print("\nEnter the employee id to unregister: ");
        empid=in.nextInt();

        if(isEmployee(empid) == true){

departmentpool.get(employeepool.get(employeeindex).man).setManager(null);
            employeepool.get(employeeindex).man=0;
            employeepool.remove(employeeindex);

            System.out.println("\nEmployee unregisterd");
            i--;
        }
        else {
            System.out.println("\nEmployee Id does not exist");
        }
    }

    private static void assignDependent(){
        System.out.print("\nEnter the employee id to assign dependents:
");
        empid=in.nextInt();

        if(isEmployee(empid)){
            System.out.println("\nEnter the number of dependents: ");
            int numdep = in.nextInt();
            for(int y=0;y<numdep;y++){
                Dependents de=assignDependents();
                employeepool.get(employeeindex).assigndepend(de);
            }
            System.out.println("\nDependents assigned");
        }
        else{
            System.out.println("\nEmployee Id does not exist");
        }
    }

    private static Dependents assignDependents(){
        in.nextLine();
        System.out.print("Enter the name of the dependents: ");
        dependname=in.nextLine();
        System.out.print("Enter the gender of the dependents: ");
        String dependgender=in.nextLine();
        System.out.print("Enter the date of birth of the dependents: ");
```

```
        String dependdob=in.nextLine();
        try {
            dependdate=new
SimpleDateFormat("dd/MM/yyyy").parse(dependdob);
        }catch (Exception e) {
            System.out.println("Invalid Date of birth");
            System.exit(0);
        }
        System.out.print("Enter the relationship of the dependents: ");
        String dependrelation=in.nextLine();
        Dependents d1 = new
Dependents(dependname,dependgender,dependdate,dependrelation);
        return d1;
    }

    private static void registerPro(){
        int choice=0;
        while(choice!=3){
            System.out.println("1: Register project");
            System.out.println("2: Unregister project");
            System.out.println("3: Return to Main menu\n");
            System.out.print("\nEnter an your option : ");
            choice=in.nextInt();

            switch(choice){
                case 1:{
                    registerProject();
                    break;
                }
                case 2:{
                    unregisterProject();
                    break;
                }
            }
        }
    }

    private static void registerProject(){
        System.out.print("\nEnter project name: ");
        proname = in.next();

        if(isProject(proname) == true){
            System.out.println("\nProject name already exist");
        }
        else {
            in.nextLine();

            System.out.print("Enter Project location: ");
            String prolocation = in.nextLine();
            Project p = new Project(proname,prolocation);
            projectpool.add(k,p);
            System.out.println("\nThe project is created.");
            k++;
        }
    }

    private static void unregisterProject(){
        in.nextLine();
```

```
        System.out.print("Enter the project name to unregister: ");
        proname=in.nextLine();

        if(isProject(proname) == true){
            projectpool.remove(projectindex);
            System.out.println("\nEmployee is unregister");
            k--;
        }
        else{
            System.out.println("\nEmployee Id does not exist");
        }
    }

    private static void assignEmployeeToDepartment(){
        System.out.print("Enter the deparment Id: ");
        depid=in.nextInt();

        in.nextLine();

        System.out.print("Enter the employee Id to be assigned: ");
        empid=in.nextInt();

        if(isEmployee(empid)&&isDepartment(depid)){
            if(isdepEmployee(departmentpool.get(departmentindex),
empid)){
                System.out.println("\nEmployee already assigned");
            }
            else{
                Employee depemployee = employeepool.get(employeeindex);

departmentpool.get(departmentindex).assignempdep(depemployee);
                System.out.println("\nEmployee is assigned to
department");
            }
        }
        else if(!isEmployee(empid)){
            System.out.println("\nThe employee Id does not exist");
        }
        else if(!isDepartment(depid)){
            System.out.println("\nThe Department Id does not exist");
        }
    }

    private static void assignProjectDep(){
        int choice=0;
        while(choice!=3){
            System.out.println("1: Assign project to Department");
            System.out.println("2: Withdraw project from Department");
            System.out.println("3: Return to Main menu\n");
            System.out.print("Enter an your option : ");
            choice=in.nextInt();
            switch(choice){
                case 1:{
                    assignProjectToDepartment();
                    break;
                }
                case 2:{
```

```
                    withdawProjectFromDepartment();
                    break;
                }
            }
        }
    }

    private static void assignProjectToDepartment(){
        System.out.print("Enter the deparment Id: ");
        depid=in.nextInt();

        in.nextLine();

        System.out.print("Enter the project to be assigned: ");
        proname=in.nextLine();

        if(isProject(proname)&&isDepartment(depid)){
            if(isdepProject(departmentpool.get(departmentindex),
proname)){
                System.out.println("\nProject already assigned");
            }
            else{
                Project depproject=projectpool.get(projectindex);

departmentpool.get(departmentindex).assignprojectdep(depproject);
                System.out.println("\nProject is assigned to
department");
            }
        }
        else if(!isProject(proname)){
            System.out.println("\nThe project does not exist");
        }
        else if(!isDepartment(depid)){
            System.out.println("\nThe Department Id does not exist");
        }
    }

    private static void withdawProjectFromDepartment(){
        System.out.print("Enter the deparment Id: ");
        depid=in.nextInt();

        in.nextLine();

        System.out.print("Enter the project to be Withdrawn: ");
        proname=in.nextLine();

        if(isDepartment(depid)&&isProject(proname)){

if(isdepProject(departmentpool.get(departmentindex),proname)){

departmentpool.get(departmentindex).withdrawProjectDep(depprojectindex);
                System.out.println("\nProject withdrawn from
department");
            }
            else{
                System.out.println("\nProject not found");
            }
        }
```

```
        else if(!isProject(proname)){
            System.out.println("\nThe project does not exist");
        }
        else if(!isDepartment(depid)){
            System.out.println("\nThe Department Id does not exist");
        }
    }

    private static void assignProjectEmp(){
        int choice=0;

        while(choice!=3){
            System.out.println("\n1: Assign project to Employee");
            System.out.println("2: Withdraw project from Employee");
            System.out.println("3: Return to Main menu\n");
            System.out.print("Enter an your option : ");
            choice=in.nextInt();
            switch(choice){
                case 1:{
                    assignProjectToEmployee();
                    break;
                }
                case 2:{
                    withdrawProjectFromEmployee();
                    break;
                }
            }
        }
    }

    private static void assignProjectToEmployee(){
        System.out.print("Enter the Employee Id: ");
        empid=in.nextInt();

        in.nextLine();

        System.out.print("Enter the project to be assigned: ");
        proname=in.nextLine();

        if(isProject(proname)&&isEmployee(empid)){
            if(isempProject(employeepool.get(employeeindex), proname)){
                System.out.println("\nProject alredy assigned");
            }
            else{
                Project empproject=projectpool.get(projectindex);

(employeepool.get(employeeindex)).assignprojectemp(empproject);
                System.out.println("\nProject is assigned to Employee");
            }
        }
        else if(!isProject(proname)){
            System.out.println("\nThe project does not exist");
        }
        else if(!isEmployee(empid)){
            System.out.println("\nThe Employee Id does not exist");
        }
    }
```

```
    private static void withdrawProjectFromEmployee(){
        System.out.print("Enter the Employee Id: ");
        empid=in.nextInt();

        in.nextLine();

        System.out.print("Enter the project to be Withdrawn: ");
        proname=in.nextLine();

        if(isEmployee(empid)){
            if(isempProject(employeepool.get(employeeindex),proname)){

employeepool.get(employeeindex).withdrawProjectEmp(empprojectindex);
                System.out.println("\nProject withdrawn from Employee");
            }
            else{
                System.out.println("\nProject not found");
            }
        }
        else if(!isProject(proname)){
            System.out.println("\nThe project does not exist");
        }
        else if(!isEmployee(depid)){
            System.out.println("\nThe Employee Id does not exist");
        }
    }

    private static void details(){
        int choice=0;
        while(choice!=4){
            System.out.println("\n1: Employee details");
            System.out.println("2: Project details");
            System.out.println("3: Department details");
            System.out.println("4: Return to Main menu\n");
            System.out.print("Enter an your option : ");
            choice=in.nextInt();

            switch(choice){
                case 1:{
                    printEmployeeDetails();
                    break;
                }
                case 2:{
                    printProjectDetails();
                    break;
                }
                case 3:{
                    printDepartmentDetails();
                    break;
                }
            }
        }
    }

    private static void printEmployeeDetails(){
        if(i>0){
            System.out.println("\nThe details of the employee are: ");
            for(int a=0;a<i;a++)
```

```
            {
                System.out.println(employeepool.get(a).toString());
                if(employeepool.get(a).c>0){

System.out.println(employeepool.get(a).printProject());
                }
                else{
                    System.out.println("\nNo projects assigned");
                }
                if(employeepool.get(a).dep>0){

System.out.println(employeepool.get(a).printDependents());
                }
                else{
                    System.out.println("\nNo Dependents assigned");
                }
            }
        }
        else {
            System.out.println("\nNo Employee created");
            System.out.println("Employee pool is empty");
        }
    }

    private static void printProjectDetails(){
        if(k>0){
            System.out.println("\nThe details of the projects are: ");
            for(int a=0;a<k;a++)
            {
                System.out.println(projectpool.get(a).toString());
            }
        }
        else {
            System.out.println("\nNo Project created");
            System.out.println("Project pool is empty");
        }
    }

    private static void printDepartmentDetails(){
        if(j>0){
            System.out.println("\nThe details of department are: ");
            for(int a=0;a<j;a++)
            {
                System.out.println(departmentpool.get(a).toString());
                if(departmentpool.get(a).b>0){

System.out.println(departmentpool.get(a).printProject());
                }
                else{
                    System.out.println("\nNo projects assigned");
                }
                if(departmentpool.get(a).w>0){

System.out.println(departmentpool.get(a).printEmployee());
                }
                else{
                    System.out.println("\nNo employee assigned");
                }
```

```
        }
    }
    else {
        System.out.println("\nNo Department created");
        System.out.println("Department pool is empty");
    }
}

private static boolean isEmployee(int empid){
    int m=0;
    employeefound=false;

    while((employeefound == false)&&(m<i)){
        if(empid == employeepool.get(m).getSSN()){
            employeefound = true;
            employeeindex = m;
            break;
        }
        m++;
    }
    return employeefound;
}

private static boolean isDepartment(int depid){
    int n=0;
    departmentfound=false;

    while((departmentfound == false)&&(n<j)){
        if(depid == departmentpool.get(n).getNumber()){
            departmentfound = true;
            departmentindex = n;
            break;
        }
        n++;
    }
    return departmentfound;
}

private static boolean isProject(String proname){
    int q=0;
    projectfound=false;

    while((projectfound == false)&&(q<k)){
        if(proname.equals(projectpool.get(q).getName())){
            projectfound = true;
            projectindex = q;
            break;
        }
        q++;
    }
    return projectfound;
}

private static boolean isdepProject(Department d,String proname){
    int r=0;
    depprojectfound=false;

    while((depprojectfound == false)&&(r<d.b)){
```

```
            if(proname.equals(d.projectdeppool.get(r).getName())){
                depprojectfound = true;
                depprojectindex = r;
                break;
            }
            r++;
        }
        return depprojectfound;
    }

    private static boolean isempProject(Employee e,String proname){
        int s=0;
        empprojectfound=false;

        while((empprojectfound == false)&&(s<e.c)){
            if(proname.equals(e.projectemppool.get(s).getName())){
                empprojectfound = true;
                empprojectindex = s;
                break;
            }
            s++;
        }
        return empprojectfound;
    }

    private static boolean isdepEmployee(Department d,int empid){
        int u=0;
        empprojectfound=false;

        while((depemployeefound == false)&&(u<d.w)){
            if(empid == (d.employeedeppool.get(u).getSSN())){
                depemployeefound = true;
                depemployeeindex = u;
                break;
            }
            u++;
        }
        return depemployeefound;
    }
}
```