/\*Title:Employee Management System

\* Author:Priya Mahalakshmi S

\* Created at:09.10.2021

\* Updated at:12.10.2021

\* Reviewed by:Akshaya

\*/

**package** employeeDetails;

**public** **class** Main {

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

UserAction user=**new** UserAction();

user.userAction();

}

}-Main.java

**package** employeeDetails;

**import** java.util.Scanner;

**public** **class** UserAction {

**public** **void** userAction() {

Scanner scanner = **new** Scanner(System.***in***);

Base base = **new** Base();

**int** choice;

**do** {

System.***out***.println("Enter your choice of action:\n1.Add Employee Details\n2.Display Employee Details\n3.Delete Employee Details\n4.Exit");

choice=scanner.nextInt();

**if**(choice==1) {

System.***out***.println("Number of Employees to be added");

**int** number = scanner.nextInt();

**int** index = 0;

**while** (index < number) {

EmployeeDetails employeeDetails = **new** EmployeeDetails(base);

System.***out***.println("Enter " + (index + 1) + "st Employee Details");

employeeDetails.getEmployeeDetails();

employeeDetails.storeEmployeeDetails();

index++;

}

}**else** **if**(choice==2) {

EmployeeDetails employeeDetails = **new** EmployeeDetails(base);

employeeDetails.displayEmployeeDetails();

}**else** **if**(choice==3) {

String employeeId;

EmployeeDetails employeeDetails = **new** EmployeeDetails(base);

System.***out***.println("Enter Id to delete employee details");

employeeId = scanner.nextLine();

employeeDetails.removeEmployeeDetails(employeeId);

}

}**while**(choice<4);

}

}-UserAction.java

**package** employeeDetails;

**public** **class** Base {

EmployeeDetailsStore employeeDetailsStore = **new** EmployeeDetailsStore();

**public** EmployeeDetailsStore getEmployeeDetailsStore(){

**return** employeeDetailsStore;

}

}-Base.java

**package** employeeDetails;

**public** **class** Employee {

**private** String id;

**private** String name;

**private** String emailId;

**private** String dateOfBirth;

**private** String dateOfJoining;

Employee(String id,String name,String emailId,String dateOfBirth,String dateOfJoining){

**this**.id=id;

**this**.name=name;

**this**.emailId=emailId;

**this**.dateOfBirth=dateOfBirth;

**this**.dateOfJoining=dateOfJoining;

}

**public** String getId() {

**return** id;

}

**public** String getName() {

**return** name;

}

**public** String getEmailId() {

**return** emailId;

}

**public** String getDateOfBirth() {

**return** dateOfBirth;

}

**public** String getDateOfJoining() {

**return** dateOfJoining;

}

}-Employee.java

package employeeDetails;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Iterator;

import java.util.Map;

import java.util.Scanner;

public class EmployeeDetails extends EmployeeDetailsValidation implements BasicActions{

private String employeeId;

private String employeeName;

private String employeeEmailId;

private String employeeDateOfBirth;

private String employeeDateOfJoining;

private Base base;

public EmployeeDetails(Base base) {

this.base=base;

}

public void getEmployeeDetails(){

//EmployeeDetailsValidation edv=new EmployeeDetailsValidation();

Scanner scanner = new Scanner(System.in);

System.out.println("Enter Employee Id");

while(true) {

employeeId=scanner.nextLine();

if(checkEmployeeIdFormat(employeeId))

break;

System.out.println("Employee Id must start with 'ACE' followed by four digits.Do not include any special characters.");

}

System.out.println("Enter Employee Name");

while(true) {

employeeName=scanner.nextLine();

if(checkEmployeeNameFormat(employeeName))

break;

System.out.println("Employee name should contain only alphabets do not include special characters or numerics");

}

System.out.println("Enter Employee Email ID");

while(true) {

employeeEmailId=scanner.nextLine();

if(checkEmployeeEmailIdFormat(employeeEmailId))

break;

System.out.println("Domain name should contain only alphabets");

}

System.out.println("Enter Employee Date of Birth (in YYYY-MM-DD format)");

while(true) {

employeeDateOfBirth=scanner.nextLine();

if(checkEmployeeDOBFormat(employeeDateOfBirth))

break;

System.out.println("Enter valid Dob");

}

System.out.println("Enter Employee Date of Joining (in YYYY-MM-DD format)");

while(true) {

employeeDateOfJoining=scanner.nextLine();

if(checkEmployeeDOJFormat(employeeDateOfJoining))

break;

System.out.println("please enter the valid date format is 'YYYY-MM-DD'\nAge must be between 18 to 60");

}

try {

Class.forName("com.mysql.jdbc.Driver");

Connection connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/query1","root","Priya2001");

PreparedStatement preparedStatement=connection.prepareStatement("INSERT INTO employee (id,name,emailid,dob,doj) VALUES (?,?,?,?,?)");

preparedStatement.setString(1,employeeId);

preparedStatement.setString(2, employeeName);

preparedStatement.setString(3, employeeEmailId);

preparedStatement.setString(4, employeeDateOfBirth);

preparedStatement.setString(5, employeeDateOfJoining);

preparedStatement.executeUpdate();

System.out.println("Records are added in the database!");

connection.close();

}

catch(Exception exception) {

System.out.println(exception);

}

}

public void storeEmployeeDetails() {

EmployeeDetailsStore employeeDetailsStore=base.getEmployeeDetailsStore();

Employee newEmployee=new Employee(employeeId,employeeName,employeeEmailId,employeeDateOfBirth,employeeDateOfJoining);

employeeDetailsStore.addNewEmployeeDetail(employeeId,newEmployee);

}

public void displayEmployeeDetails() {

EmployeeDetailsStore employeeDetailsStore=base.getEmployeeDetailsStore();

employeeDetailsStore.displayEmployeeDetails();

}

public void updateEmployeeDetails() {

}

public void removeEmployeeDetails(String employeeId) {

EmployeeDetailsStore employeeDetailsStore=base.getEmployeeDetailsStore();

employeeDetailsStore.removeEmployeeDetail(employeeId);

}

}-EmployeeDetails.java

package employeeDetails;

import java.util.HashMap;

import java.util.Map;

public class EmployeeDetailsStore {

public HashMap<String,Employee> dataBase=new HashMap<>();

public Employee getEmployeeDetails (String employeeID) {

return dataBase.get(employeeID);

}

public void addNewEmployeeDetail(String employeeID, Employee employee) {

dataBase.put(employeeID,employee);

}

public void removeEmployeeDetail(String employeeID) {

dataBase.remove(employeeID);

}

public void displayEmployeeDetails(){

for(Map.Entry<String,Employee> entry : dataBase.entrySet()){

Employee employee=entry.getValue();

System.out.println("--------------------------------------------------\nEmployee id:" +employee.getId()+

"\nEmployee name:" +employee.getName()+

"\nEmployee emailId:" +employee.getEmailId()+

"\nEmployee dateOfBirth:" +employee.getDateOfBirth()+

"\nEmployee dateOfJoining:"+employee.getDateOfJoining()+

"\n--------------------------------------------------"

);

}

}

}-EmployeeDetailsStore.java

package employeeDetails;

import java.time.LocalDate;

import java.time.Period;

import java.time.ZoneId;

import java.time.format.DateTimeFormatter;

public class EmployeeDetailsValidation extends EmployeeDetailsStore {

public boolean checkEmployeeIdFormat(String employeeId) {

return employeeId.matches("^ACE\\d{4}$");

}

public boolean checkUniqueId(String employeeId)

{

if(dataBase.containsKey(employeeId))

return true;

else

return false;

}

public boolean checkEmployeeNameFormat(String employeeName) {

return employeeName.matches("^[a-zA-Z]\*$");

}

public boolean checkEmployeeEmailIdFormat(String employeeEmailId) {

return employeeEmailId.matches("^[a-zA-Z0-9\_.]+@gmail.com$");

}

public boolean checkEmployeeDOBFormat(String employeeDateOfBirth) {

LocalDate dob = LocalDate.parse(employeeDateOfBirth);

LocalDate curDate = LocalDate.now();

int age=Period.between(dob, curDate).getYears();

return age>18&&age<60;

}

public boolean checkEmployeeDOJFormat(String employeeDateOfJoining){

String dateFormat="yyyy-MM-dd";

LocalDate localDate = LocalDate.now(ZoneId.systemDefault());

DateTimeFormatter dtf = DateTimeFormatter.ofPattern(dateFormat);

LocalDate inputDate = LocalDate.parse(employeeDateOfJoining, dtf);

return !inputDate.isAfter(localDate);

}

}-EmployeeDetailsValidation.java

**package** employeeDetails;

**public** **interface** BasicActions {

**public** **void** getEmployeeDetails();

**public** **void** storeEmployeeDetails();

**public** **void** displayEmployeeDetails();

**public** **void** updateEmployeeDetails();

**public** **void** removeEmployeeDetails(String employeeId);

}-BasicActions.java