

Crime Reporting System

entity

Evidence.java

```
package entity;
```

```
public class Evidence {
```

```
    private int EvidenceID;
```

```
    private int IncidentID;
```

```
    private String LocationFound;
```

```
    private String Description;
```

```
    public Evidence(int EvidenceID,int IncidentID,String LocationFound,String Description)
```

```
    {
```

```
        this.EvidenceID=EvidenceID;
```

```
        this.IncidentID=IncidentID;
```

```
        this.LocationFound=LocationFound;
```

```
        this.Description=Description;
```

```
    }
```

```
    public int getEvidenceID()
```

```
    {
```

```
        return EvidenceID;
```

```
    }
```

```
    public int getIncidentID()
```

```
    {
```

```
        return IncidentID;
```

```
    }
```

```
    public String getLocationFound()
```

```
    {
```

```
        return LocationFound;
```

```
    }

    public String getDescription()
    {
        return Description;
    }

    public void setEvidenceID(int EvidenceID)
    {
        this.EvidenceID=EvidenceID;
    }

    public void setIncidentID(int IncidentID)
    {
        this.IncidentID=IncidentID;
    }

    public void setLocationFound(String LocationFound)
    {
        this.LocationFound=LocationFound;
    }

    public void setDescription(String Description)
    {
        this.Description=Description;
    }
}
```

Incidents.java

package entity;

import java.time.LocalDate;

```
public class Incidents {  
    private int IncidentID;  
    private String IncidentType;  
    private String IncidentDate;  
    private String Location;  
    private String Description;  
    private String Status;  
    private int VictimID;  
    private int SuspectID;  
    private int OfficerID;  
  
    public Incidents(int IncidentID,String IncidentType,String IncidentDate,String  
Location,String Description,String Status,int VictimID,int SuspectID,int OfficerID )  
    {  
        this.IncidentID=IncidentID;  
        this.IncidentType=IncidentType;  
        this.IncidentDate=IncidentDate;  
        this.Location=Location;  
        this.Description=Description;  
        this.Status=Status;  
        this.VictimID=VictimID;  
        this.SuspectID=SuspectID;  
        this.OfficerID=OfficerID;  
    }  
  
    public int getIncidentID() {  
        return IncidentID;  
    }  
  
    public void setIncidentID(int incidentID) {  
        IncidentID = incidentID;  
    }  
}
```

```
}
```

```
public String getIncidentType() {  
    return IncidentType;  
}
```

```
public void setIncidentType(String incidentType) {  
    IncidentType = incidentType;  
}
```

```
public String getIncidentDate() {  
    return IncidentDate;  
}
```

```
public void setIncidentDate(String localDate) {  
    IncidentDate = localDate;  
}
```

```
public String getLocation() {  
    return Location;  
}
```

```
public void setLocation(String location) {  
    Location = location;  
}
```

```
public String getDescription() {  
    return Description;  
}
```

```
public void setDescription(String description) {  
    Description = description;  
}
```

```
}
```

```
public String getStatus() {
```

```
    return Status;
```

```
}
```

```
public void setStatus(String status) {
```

```
    Status = status;
```

```
}
```

```
public int getVictimID() {
```

```
    return VictimID;
```

```
}
```

```
public void setVictimID(int victimID) {
```

```
    VictimID = victimID;
```

```
}
```

```
public int getSuspectID() {
```

```
    return SuspectID;
```

```
}
```

```
public void setSuspectID(int suspectID) {
```

```
    SuspectID = suspectID;
```

```
}
```

```
public int getOfficerID() {
```

```
    return OfficerID;
```

```
}
```

```
public void setOfficerID(int officerID) {
```

```
    OfficerID = officerID;
```

```
}  
}
```

LawEnforcementAgency.java

```
package entity;
```

```
public class LawEnforcementAgency {
```

```
    private int AgencyID;
```

```
    private String AgencyName;
```

```
    private String Jurisdiction;
```

```
    private String Address;
```

```
    private String PhoneNumber;
```

```
    private int officer;
```

```
    public LawEnforcementAgency(int AgencyID, String AgencyName, String Jurisdiction,  
                                String Address, String PhoneNumber, int officer) {
```

```
        this.AgencyID = AgencyID;
```

```
        this.AgencyName = AgencyName;
```

```
        this.Jurisdiction = Jurisdiction;
```

```
        this.Address = Address;
```

```
        this.PhoneNumber = PhoneNumber;
```

```
        this.officer = officer;
```

```
    }
```

```
    public int getAgencyID() {
```

```
        return AgencyID;
```

```
    }
```

```
    public void setAgencyID(int agencyID) {
```

```
        AgencyID = agencyID;
    }

    public String getAgencyName() {
        return AgencyName;
    }

    public void setAgencyName(String agencyName) {
        AgencyName = agencyName;
    }

    public String getJurisdiction() {
        return Jurisdiction;
    }

    public void setJurisdiction(String jurisdiction) {
        Jurisdiction = jurisdiction;
    }

    public String getAddress() {
        return Address;
    }

    public void setAddress(String address) {
        Address = address;
    }

    public String getPhoneNumber() {
        return PhoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
```

```
        PhoneNumber = phoneNumber;
    }

    public int getOfficer() {
        return officer;
    }

    public void setOfficer(int officer) {
        this.officer = officer;
    }
}
```

Officers.java

```
package entity;

public class Officers {
    private int OfficerID; //
    private String FirstName;
    private String LastName;
    private String BadgeNumber;
    private String Rank;
    private String Address;
    private String PhoneNumber;
    private int AgencyID;

    // Constructor
    public Officers(int OfficerID, String FirstName, String LastName, String BadgeNumber,
        String Rank, String Address, String PhoneNumber, int AgencyID) {
        this.OfficerID = OfficerID;
        this.FirstName = FirstName;
        this.LastName = LastName;
        this.BadgeNumber = BadgeNumber;
        this.Rank = Rank;
    }
}
```



```
    this.Address = Address;
    this.PhoneNumber = PhoneNumber;
    this.AgencyID = AgencyID;
}
```

```
public int getOfficerID() {
    return OfficerID;
}
```

```
public void setOfficerID(int officerID) {
    OfficerID = officerID;
}
```

```
public String getFirstName() {
    return FirstName;
}
```

```
public void setFirstName(String firstName) {
    FirstName = firstName;
}
```

```
public String getLastName() {
    return LastName;
}
```

```
public void setLastName(String lastName) {
    LastName = lastName;
}
```

```
public String getBadgeNumber() {
    return BadgeNumber;
}
```

```
}
```

```
public void setBadgeNumber(String badgeNumber) {
```

```
    BadgeNumber = badgeNumber;
```

```
}
```

```
public String getRank() {
```

```
    return Rank;
```

```
}
```

```
public void setRank(String rank) {
```

```
    Rank = rank;
```

```
}
```

```
public String getAddress() {
```

```
    return Address;
```

```
}
```

```
public void setAddress(String address) {
```

```
    Address = address;
```

```
}
```

```
public String getPhoneNumber() {
```

```
    return PhoneNumber;
```

```
}
```

```
public void setPhoneNumber(String phoneNumber) {
```

```
    PhoneNumber = phoneNumber;
```

```
}
```

```
public int getAgencyID() {
```

```
    return AgencyID;
```

```
}

    public void setAgencyID(int agencyID) {
        AgencyID = agencyID;
    }
}
```

Reports.java

```
package entity;
```

```
public class Reports {
    private int ReportID;
    private int IncidentID;
    private int ReportingOfficer;
    private String ReportDate;
    private String ReportDetails;
    private String Status;

    public Reports(int ReportID, int IncidentID, int ReportingOfficer, String ReportDate,
        String ReportDetails, String Status) {
        this.ReportID = ReportID;
        this.IncidentID = IncidentID;
        this.ReportingOfficer = ReportingOfficer;
        this.ReportDate = ReportDate;
        this.ReportDetails = ReportDetails;
        this.Status = Status;
    }

    public int getReportID() {
        return ReportID;
    }
}
```

```
}
```

```
public void setReportID(int reportID) {
```

```
    ReportID = reportID;
```

```
}
```

```
public int getIncidentID() {
```

```
    return IncidentID;
```

```
}
```

```
public void setIncidentID(int incidentID) {
```

```
    IncidentID = incidentID;
```

```
}
```

```
public int getReportingOfficer() {
```

```
    return ReportingOfficer;
```

```
}
```

```
public void setReportingOfficer(int reportingOfficer) {
```

```
    ReportingOfficer = reportingOfficer;
```

```
}
```

```
public String getReportDate() {
```

```
    return ReportDate;
```

```
}
```

```
public void setReportDate(String reportDate) {
```

```
    ReportDate = reportDate;
```

```
}
```

```
public String getReportDetails() {
```

```
    return ReportDetails;
```

```
}

public void setReportDetails(String reportDetails) {
    ReportDetails = reportDetails;
}

public String getStatus() {
    return Status;
}

public void setStatus(String status) {
    Status = status;
}
}
```

Suspects.java

```
package entity;
```

```
public class Suspects {
    private int SuspectID;
    private String FirstName;
    private String LastName;
    private String DateOfBirth;
    private String Gender;
    private String Address;
    private String PhoneNumber;

    public Suspects(int SuspectID, String FirstName, String LastName, String DateOfBirth,
        String Gender, String Address, String PhoneNumber) {
        this.SuspectID = SuspectID;
        this.FirstName = FirstName;
```

```
    this.LastName = LastName;
    this.DateOfBirth = DateOfBirth;
    this.Gender = Gender;
    this.Address = Address;
    this.PhoneNumber = PhoneNumber;
}
```

```
public int getSuspectID() {
    return SuspectID;
}
```

```
public void setSuspectID(int suspectID) {
    SuspectID = suspectID;
}
```

```
public String getFirstName() {
    return FirstName;
}
```

```
public void setFirstName(String firstName) {
    FirstName = firstName;
}
```

```
public String getLastName() {
    return LastName;
}
```

```
public void setLastName(String lastName) {
    LastName = lastName;
}
```

```
public String getDateOfBirth() {  
    return DateOfBirth;  
}  
  
public void setDateOfBirth(String dateOfBirth) {  
    DateOfBirth = dateOfBirth;  
}  
  
public String getGender() {  
    return Gender;  
}  
  
public void setGender(String gender) {  
    Gender = gender;  
}  
  
public String getAddress() {  
    return Address;  
}  
  
public void setAddress(String address) {  
    Address = address;  
}  
  
public String getPhoneNumber() {  
    return PhoneNumber;  
}  
  
public void setPhoneNumber(String phoneNumber) {  
    PhoneNumber = phoneNumber;  
}  
}
```

Victims.java

```
package entity;
```

```
public class Victims {
```

```
    private int VictimID;
```

```
    private String FirstName;
```

```
    private String LastName;
```

```
    private String DateOfBirth;
```

```
    private String Gender;
```

```
    private String Address;
```

```
    private String PhoneNumber;
```

```
    public Victims(int VictimID, String FirstName, String LastName, String DateOfBirth,String  
Gender, String Address, String PhoneNumber) {
```

```
        this.VictimID = VictimID;
```

```
        this.FirstName = FirstName;
```

```
        this.LastName = LastName;
```

```
        this.DateOfBirth = DateOfBirth;
```

```
        this.Gender = Gender;
```

```
        this.Address = Address;
```

```
        this.PhoneNumber = PhoneNumber;
```

```
    }
```

```
    public int getVictimID() {
```

```
        return VictimID;
```

```
    }
```

```
    public void setVictimID(int victimID) {
```

```
        VictimID = victimID;
```



```
}
```

```
public String getFirstName() {
```

```
    return FirstName;
```

```
}
```

```
public void setFirstName(String firstName) {
```

```
    FirstName = firstName;
```

```
}
```

```
public String getLastName() {
```

```
    return LastName;
```

```
}
```

```
public void setLastName(String lastName) {
```

```
    LastName = lastName;
```

```
}
```

```
public String getDateOfBirth() {
```

```
    return DateOfBirth;
```

```
}
```

```
public void setDateOfBirth(String dateOfBirth) {
```

```
    DateOfBirth = dateOfBirth;
```

```
}
```

```
public String getGender() {
```

```
    return Gender;
```

```
}
```

```
public void setGender(String gender) {
```

```
    Gender = gender;
```

```

    }

    public String getAddress() {
        return Address;
    }

    public void setAddress(String address) {
        Address = address;
    }

    public String getPhoneNumber() {
        return PhoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        PhoneNumber = phoneNumber;
    }
}

```

dao

CrimeAnalysisDAO.java

```

package dao;

import entity.*;
import java.util.Collection;
import java.util.List;

public interface CrimeAnalysisDAO {

    public List<Incidents> searchIncidents(String IncidentType);
}

```

```
        public Reports generateIncidentReport(Incidents incident);

    }
}
```

CrimeAnalysisImpl.java

```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.List;
import java.util.ArrayList;
import java.sql.ResultSet;
import entity.Incidents;
import util.DBConnUtil;
import exception.*;
import entity.*;

public abstract class CrimeAnalysisImpl implements CrimeAnalysisDAO{

    private static Connection connection;

    static
    {
        connection = DBConnUtil.getConnectionObject();
    }

    public static boolean createIncident(Incidents incident)
    {
    }
```

```
String query="INSERT INTO INCIDENTS VALUES(?,?,?,?,?,?,?,?)";
```

```
try(PreparedStatement stmt=connection.prepareStatement(query))
```

```
{
```

```
    stmt.setInt(1, incident.getIncidentID());
```

```
    stmt.setString(2, incident.getIncidentType());
```

```
    stmt.setString(3, incident.getIncidentDate());
```

```
    stmt.setString(4, incident.getLocation());
```

```
    stmt.setString(5, incident.getDescription());
```

```
    stmt.setString(6, incident.getStatus());
```

```
    stmt.setInt(7, incident.getVictimID());
```

```
    stmt.setInt(8, incident.getSuspectID());
```

```
    stmt.setInt(9, incident.getOfficerID());
```

```
    stmt.executeUpdate();
```

```
    return true;
```

```
}
```

```
catch(SQLException sql)
```

```
{
```

```
    sql.printStackTrace();
```

```
}
```

```
return false;
```

```
}
```

```
public static boolean updateIncidentStatus(String ReportStatus,int IncidentID)
```

```
{
```

```
    String query1 = "UPDATE INCIDENTS SET STATUS=? WHERE INCIDENTID=?";
```

```
    String query2 = "UPDATE REPORTS SET STATUS=? WHERE INCIDENTID=?";
```

```

try
(PreparedStatement stmt1 = connection.prepareStatement(query1);
PreparedStatement stmt2 = connection.prepareStatement(query2))
{
    stmt1.setString(1, ReportStatus);
    stmt1.setInt(2, IncidentID);

    stmt2.setString(1, ReportStatus);
    stmt2.setInt(2, IncidentID);

    stmt1.executeUpdate();
    stmt2.executeUpdate();

    System.out.println("Status has been Updated Successfully...");
    return true;

}
catch(SQLException sql)
{
    sql.printStackTrace();
}

return false;

}

public static List<Incidents> getIncidentInDateRange(String startDate,String endDate)
{

    List<Incidents> incidents=new ArrayList<>();

```

```

AND ?";

String query="SELECT * FROM INCIDENTS WHERE INCIDENTDATE BETWEEN ?

try(Connection connection=DBConnUtil.getConnectionObject())
{
    PreparedStatement stmt=connection.prepareStatement(query);
    stmt.setString(1, startDate);
    stmt.setString(2, endDate);

    ResultSet rs=stmt.executeQuery();

    while(rs.next())
    {
        while (rs.next()) {
            Incidents incident = new Incidents(
                rs.getInt("IncidentID"),
                rs.getString("IncidentType"),
                rs.getString("IncidentDate"),
                rs.getString("Location"),
                rs.getString("Description"),
                rs.getString("Status"),
                rs.getInt("VictimID"),
                rs.getInt("SuspectID"),
                rs.getInt("OfficerID")
            );
            incidents.add(incident);
        }
    }
}
catch(SQLException sql)
{
    sql.printStackTrace();
    return null;
}

```

```
        return incidents;

    }

    public static List<Incidents> getIncidentsByIncidentType(String IncidentType)
    {
        String query="SELECT * FROM INCIDENTS WHERE INCIDENTTYPE = ?";
        List<Incidents> incidents=new ArrayList<>();

        try(Connection connection=DBConnUtil.getConnectionObject())
        {
            PreparedStatement stmt=connection.prepareStatement(query);
            stmt.setString(1,IncidentType);

            ResultSet rs=stmt.executeQuery();

            while (rs.next()){
                Incidents incident = new Incidents(
                    rs.getInt("IncidentID"),
                    rs.getString("IncidentType"),
                    rs.getString("IncidentDate"),
                    rs.getString("Location"),
                    rs.getString("Description"),
                    rs.getString("Status"),
                    rs.getInt("VictimID"),
                    rs.getInt("SuspectID"),
                    rs.getInt("OfficerID")
                );
                incidents.add(incident);
            }
        }
    }
}
```

```

    }
    catch(SQLException sql)
    {
        sql.printStackTrace();
        return null;
    }

    return incidents;
}

public static List<Reports> getReportByIncidentID(int IncidentType)
{

    String query="SELECT * FROM REPORTS WHERE INCIDENTID=?";
    List<Reports> reports=new ArrayList<>();

    try(Connection connection=DBConnUtil.getConnectionObject())
    {
        PreparedStatement stmt=connection.prepareStatement(query);
        stmt.setInt(1, IncidentType);

        ResultSet rs=stmt.executeQuery();

        while(rs.next())
        {
            Reports report = new Reports(
                rs.getInt("ReportID"),
                rs.getInt("IncidentID"),
                rs.getInt("ReportingOfficer"),
                rs.getString("ReportDate"),
                rs.getString("ReportDetails"),
            );
        }
    }
}

```



```

        rs.getString("Status")
    );

    reports.add(report);
}

return reports;

}
catch(SQLException sql)
{
    sql.printStackTrace();
    return null;
}
}

public static void printArrayList(List<Incidents> incidents) {
    System.out.println("Incident Details:");
    for (Incidents incident : incidents) {
        System.out.println("Incident ID: " + incident.getIncidentID());
        System.out.println("Type: " + incident.getIncidentType());
        System.out.println("Date: " + incident.getIncidentDate());
        System.out.println("Location: " + incident.getLocation());
        System.out.println("Description: " + incident.getDescription());
        System.out.println("Status: " + incident.getStatus());
        System.out.println("Victim ID: " + incident.getVictimID());
        System.out.println("Suspect ID: " + incident.getSuspectID());
        System.out.println("Officer ID: " + incident.getOfficerID());
        System.out.println("-----");
    }
}

```

```

    }

    public static void printReportsArrayList(List<Reports> reports) {
        System.out.println("Report Details:");
        for (Reports report : reports) {
            System.out.println("Report ID: " + report.getReportID());
            System.out.println("Incident ID: " + report.getIncidentID());
            System.out.println("Reporting Officer: " + report.getReportingOfficer());
            System.out.println("Report Date: " + report.getReportDate());
            System.out.println("Report Details: " + report.getReportDetails());
            System.out.println("Status: " + report.getStatus());
            System.out.println("-----");
        }
    }
}

```

```

}

```

getInputs.java

```

package dao;

import entity.*;
import java.util.Scanner;

public class getInputs {

    public static Scanner scanner=new Scanner(System.in);

    public static Incidents createIncidentInput()
    {
        System.out.println("Enter Incident ID : ");
        int IncidentID = scanner.nextInt();
    }
}

```

```
scanner.nextLine();
```

```
System.out.println("Enter IncidentType (e.g., Robbery, Homicide, Theft) :");
```

```
String IncidentType = scanner.nextLine();
```

```
System.out.println("Enter Incident Date (yyyy-mm-dd) :");
```

```
String IncidentDate = scanner.nextLine();
```

```
System.out.println("Enter Location (Geospatial Data: Latitude and Longitude) :  
");
```

```
String Location = scanner.nextLine();
```

```
System.out.println("Enter Description : ");
```

```
String Description = scanner.nextLine();
```

```
System.out.println("Enter Status (e.g., Open, Closed, Under Investigation) : ");
```

```
String Status = scanner.nextLine();
```

```
System.out.println("Enter Victim ID : ");
```

```
int VictimID = scanner.nextInt();
```

```
scanner.nextLine();
```

```
System.out.println("Enter Suspect ID : ");
```

```
int SuspectID = scanner.nextInt();
```

```
scanner.nextLine();
```

```
System.out.println("Enter Officer ID : ");
```

```
int OfficerID = scanner.nextInt();
```

```
Incidents incident=new  
Incidents(IncidentID,IncidentType,IncidentDate,Location,Description,Status,VictimID,SuspectI  
D,OfficerID);
```

```
        return incident;
    }
}
```

exception

IncidentNotFoundInDateRangeException.java

```
package exception;

public class IncidentNotFoundInDateRangeException extends Exception {

    private static final long serialVersionUID = 1L;

    public IncidentNotFoundInDateRangeException(String message) {
        super(message);
    }
}
```

IncidentNumberNotFoundException.java

```
package exception;

public class IncidentNumberNotFoundException extends Exception {
    public IncidentNumberNotFoundException(String message) {
        super(message);
    }
}
```

main

main_module.java

```
package main;

import java.sql.*;
import java.util.*;
```

```
import entity.*;
import util.*;
import dao.*;
import java.util.List;
import exception.*;

public class main_module {

    public static void main(String[] args) {

        Scanner scanner=new Scanner(System.in);
        boolean exit=false;
        while(!exit)
        {
            System.out.println("\n=== Crime Analysis and Reporting System ===");
            System.out.println("1. Create Incident");
            System.out.println("2. Update Incident Status");
            System.out.println("3. Get Incidents in Date Range");
            System.out.println("4. Search Incidents");
            System.out.println("5. Generate Incident Report");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");
            int choice=scanner.nextInt();

            switch(choice)
            {
                case 1:
                    Incidents incident=getInputs.createIncidentInput();
                    if(CrimeAnalysisImpl.createIncident(incident))
                    {
                        System.out.println("Incident Created Successfully...!");
                    }
                }
            }
        }
    }
}
```

```
}  
break;
```

case 2:

```
System.out.println("Enter Incident ID :");  
int IncidentID=scanner.nextInt();  
scanner.nextLine();  
System.out.println("Enter current Status of Incident :");  
String ReportStatus=scanner.nextLine();  
  
CrimeAnalysisImpl.updateIncidentStatus(ReportStatus, IncidentID);  
break;
```

case 3:

```
System.out.println("Enter a start Date(yyyy-mm-dd) : ");  
String startDate=scanner.nextLine();  
scanner.nextLine();  
  
System.out.println("Enter a end Date (yyyy-mm-dd) : ");  
String endDate=scanner.nextLine();  
  
List<Incidents> incidents = null;  
  
try {  
incidents = CrimeAnalysisImpl.getIncidentInDateRange(startDate, endDate);  
  
if (incidents != null && !incidents.isEmpty()) {  
    CrimeAnalysisImpl.printArrayList(incidents);  
} else {  
    throw new IncidentNotFoundInDateRangeException("No incidents found in the  
given date range.");  
}  
  
} catch (IncidentNotFoundInDateRangeException e) {
```

```
        System.err.println(e.getMessage());
    } catch (Exception e) {
        e.printStackTrace();
    }
```

```
break;
```

case 4:

```
    scanner.nextLine();
    System.out.println("Enter Incident Type to search Incident : ");
    String IncidentType=scanner.nextLine();
```

```
    List<Incidents> inci =
    CrimeAnalysisImpl.getIncidentsByIncidentType(IncidentType);
```

```
    if(inci!=null && !inci.isEmpty())
    {
        CrimeAnalysisImpl.printArrayList(inci);
    }
    else
    {
        System.out.println("No Incident of type "+IncidentType+"\n");
    }
```

```
break;
```

case 5:

```
    try {
        scanner.nextLine();
```

```

        System.out.println("Enter your Incident ID to view the current Report: ");
        int IncidentID1 = scanner.nextInt();

        List<Reports> reports =
CrimeAnalysisImpl.getReportByIncidentID(IncidentID1);

        if (reports != null && !reports.isEmpty()) {
            CrimeAnalysisImpl.printReportsArrayList(reports);
        } else {
            throw new IncidentNumberNotFoundException("No Incident with ID " +
IncidentID1 + " found.");
        }
    } catch (IncidentNumberNotFoundException e) {
        System.err.println(e.getMessage());
    } catch (Exception e) {
        e.printStackTrace();
    }

    break;

    case 6:
        System.out.println("Exiting... Goodbye!");
        exit = true;
        break;

    default:
        System.out.println("Invalid choice. Please try again.");
    }
}
}

```



```
}
```

util

DBConnUtil.java

```
package util;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.SQLException;
```

```
public class DBConnUtil
```

```
{
```

```
    public static Connection getConnectionObject()
```

```
    {
```

```
        String url = DBPropertyUtil.getConnectionString("db.properties");
```

```
        Connection connection = null;
```

```
        try
```

```
        {
```

```
            connection = DriverManager.getConnection(url);
```

```
            System.out.println("Connection Established Successfully...");
```

```
        }
```

```
        catch(SQLException sql)
```

```
        {
```

```
            sql.printStackTrace();
```

```
        }
```

```
        return connection;
```

```
    }
```

```
}
```

DBPropertyUtil.java

```
package util;

import java.io.FileInputStream;
import java.io.IOException;
import java.util.Properties;

public class DBPropertyUtil
{
    public static String getConnectionString(String DBFileName)
    {
        Properties properties=null;
        String protocol=null;
        String dbname=null;
        String user=null;
        String password=null;

        DBFileName="./src/"+DBFileName;

        try(FileInputStream input = new FileInputStream(DBFileName))
        {
            properties = new Properties();
            properties.load(input);

            protocol=properties.getProperty("protocol");
            dbname=properties.getProperty("dbname");
            user=properties.getProperty("user");
            password=properties.getProperty("password");

        }
        catch(IOException ex)
        {
```

```

        ex.printStackTrace();
    }

    if(protocol!=null && dbname!=null && user!=null && password!=null)
    {
        return protocol+"/"+dbname+"?user="+user+"&password="+password;
    }

    return null;
}
}

```

module-info.java

```

/**
 *
 */
/**
 *
 */
module crime {
    requires java.sql;
    requires org.junit.jupiter.api;

    requires org.junit.platform.commons; // Platform common utilities (optional for most use
cases)
    requires org.junit.jupiter.params;
}

```

db.properties

```
protocol = jdbc:mysql://localhost:3306
```

```
dbname = crime
```

```
user = root
```

```
password = Charu@07
```

JUNIT

```
IncidentServiceTest.java
```

```
package JUNIT;
```

```
import static org.junit.jupiter.api.Assertions.*;
```

```
import dao.CrimeAnalysisImpl;
```

```
import org.junit.jupiter.api.Test;
```

```
import java.util.List;
```

```
import entity.*;
```

```
import exception.*;
```

```
public class IncidentServiceTest {
```

```
    @Test
```

```
    public void testCreateIncident() {
```

```
        Incidents newIncident = new Incidents(108, "Robbery", "2024-11-23", "123, ABC Street",  
        "Stolen items", "Open", 1, 1,301);
```

```
        boolean isCreated = CrimeAnalysisImpl.createIncident(newIncident);
```

```
        assertTrue(isCreated, "Incident should be created successfully.");
```

```
    }
```

```
    @Test
```

```
    public void testUpdateIncidentStatus_Success() {
```

```
        int incidentID = 101;
```

```
String newStatus = "Closed";
```

```
boolean isUpdated = CrimeAnalysisImpl.updateIncidentStatus(newStatus, incidentID);
```

```
assertTrue(isUpdated, "The incident status should be updated successfully.");
```

```
}
```

```
}
```