“Amazon”

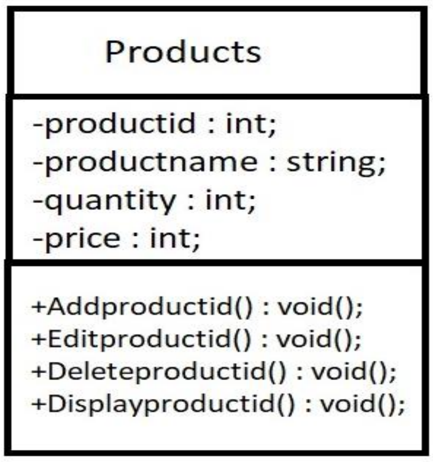
Products Class

Code:

```
class Products
{
   private int productid;
   private string productname;
   private int quantity;
   private int price;

   Public static void Addproductid()
   {
      // TO DO
   }
    Public static void Deleteproductid()
   {
      // TO DO
   }
    Public static void Editproductid()
   {
      // TO DO
   }
    Public static void Displayproductid()
   {
      // TO DO
   }
}
```

UML Diagram:



Products
-productid : int; -productname : string; -quantity : int; -price : int;

+Addproductid() : void();
+Editproductid() : void();
+Deleteproductid() : void();
+Displayproductid() : void();

“Amazon”

User Class

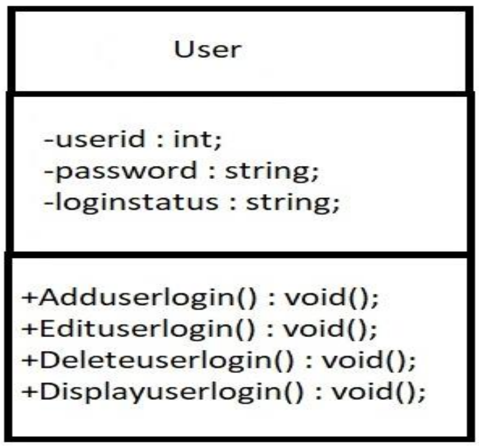
Code:

```
class User
{
    private string userid;
    private string password;
    private string loginstatus;

     Public static void Adduserlogin()
    {
        // TO DO
    }
     Public static void Deleteuserlogin()
    {
        // TO DO
    }
     Public static void Edituserlogin()
    {
        // TO DO
    }
     Public static void Displayuserlogin()
    {
        // TO DO
    }

}
```

UML Diagram:



User

-userid : int;
-password : string;
-loginstatus : string;

+Adduserlogin() : void();
+Edituserlogin() : void();
+Deleteuserlogin() : void();
+Displayuserlogin() : void();

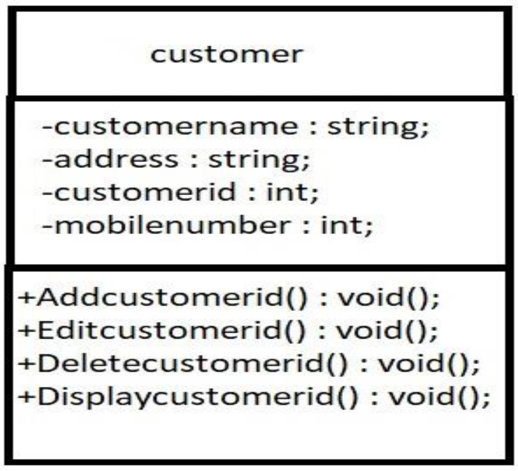
“Amazon”
Customer Class
Code:

```
class Customer
{
   private string customername;
    private string address;
    private int customerid;
    private int mobilenumber;

      Public static void Addcustomerid()
    {
       // TO DO
    }
     Public static void Deletecustomerid ()
    {
       // TO DO
    }
     Public static void Editcustomerid()
    {
       // TO DO
    }
     Public static void Displaycustomerid()
    {
       // TO DO
    }

}
```

UML Diagram:



customer

-customername : string;
-address : string;
-customerid : int;
-mobilenumber : int;

+Addcustomerid() : void();
+Editcustomerid() : void();
+Deletecustomerid() : void();
+Displaycustomerid() : void();

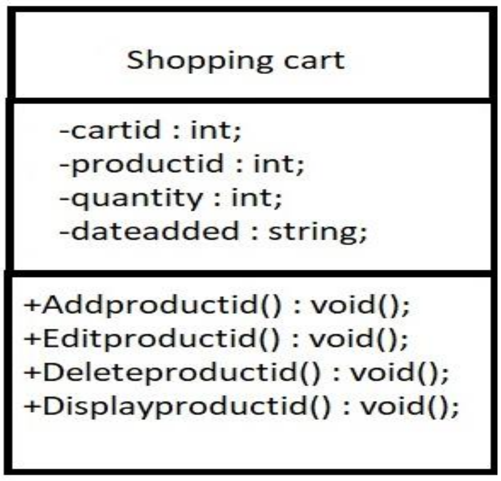
“Amazon”
Shopping Cart Class
Code:

```
class Shoppingcart
{
    private int cartid;
    private int productid;
    private int quantity;
    private string dateadded;

      Public static void Addproductid()
    {
        // TO DO
    }
     Public static void Deleteproductid()
    {
        // TO DO
    }
     Public static void Editproductid()
    {
        // TO DO
    }
     Public static void Displayproductid()
    {
        // TO DO
    }

}
```

UML Diagram:



Shopping cart

-cartid : int;
-productid : int;
-quantity : int;
-dateadded : string;

+Addproductid() : void();
+Editproductid() : void();
+Deleteproductid() : void();
+Displayproductid() : void();

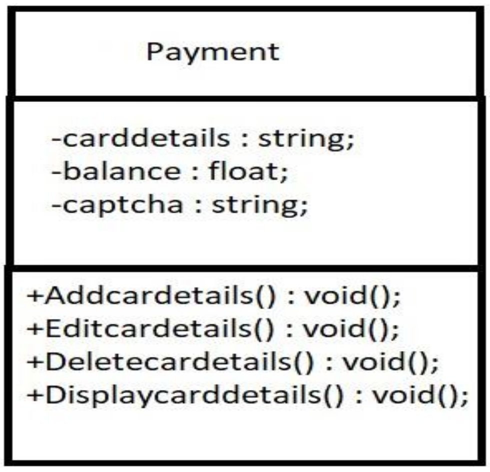
“Amazon”
Payment Class
Code:

```
class Payment
{
   private string carddetails;
   private float balance;
   private string captcha;

    Public static void Addcarddetails()
   {
      // TO DO
   }
    Public static void Deletecarddetails()
   {
      // TO DO
   }
    Public static void Editcarddetails()
   {
      // TO DO
   }
    Public static void Displaycarddetails()
   {
      // TO DO
   }

}
```

UML Diagram:



Payment

-carddetails : string;
-balance : float;
-captcha : string;

+Addcardetails() : void();
+Editcardetails() : void();
+Deletecardetails() : void();
+Displaycarddetails() : void();

“Police Station”

FIR Class

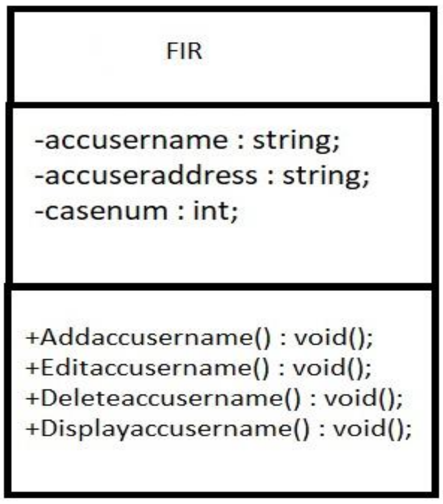
Code:

```
class FIR
{
   private string accusername;
   private string accuseraddress;
   private int casenum;

    Public static void Addaccusername()
   {
      // TO DO
   }
    Public static void Deleteaccusername()
   {
      // TO DO
   }
    Public static void Editaccusername()
   {
      // TO DO
   }
    Public static void Displayaccusername()
   {
      // TO DO
   }

}
```

UML Diagram:



FIR

-accusername : string;
-accuseraddress : string;
-casenum : int;

+Addaccusername() : void();
+Editaccusername() : void();
+Deleteaccusername() : void();
+Displayaccusername() : void();

“Police Station”

Guns Class

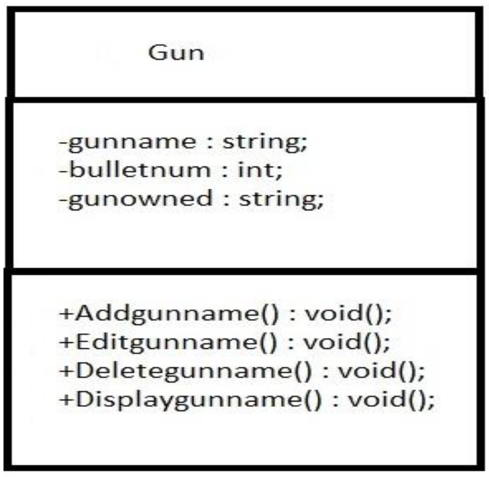
Code:

```
class Guns
{
    private string gunname;
    private int bulletnum;
    private string gunowned;

     Public static void Addgunname()
    {
        // TO DO
    }
     Public static void Deletegunname()
    {
        // TO DO
    }
     Public static void Editgunname()
    {
        // TO DO
    }
     Public static void Displaygunname()
    {
        // TO DO
    }

}
```

UML Diagram:



Gun

-gunname : string;
-bulletnum : int;
-gunowned : string;

+Addgunname() : void();
+Editgunname() : void();
+Deletegunname() : void();
+Displaygunname() : void();

“Police Station”

employee Class

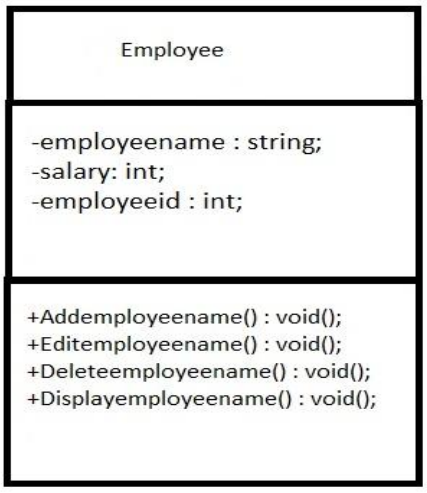
Code:

```
class Employee
{
    private string employeename;
    private int salary;
    private int employeeid;

     Public static void Addemployeename()
    {
       // TO DO
    }
     Public static void Deleteemployeename()
    {
       // TO DO
    }
     Public static void Editemployeename()
    {
       // TO DO
    }
     Public static void Displayemployeename()
    {
       // TO DO
    }

}
```

UML Diagram:


Employee
-employeename : string;
-salary: int;
-employeeid : int;
+Addemployeename() : void();
+Editemployeename() : void();
+Deleteemployeename() : void();
+Displayemployeename() : void();

“Police Station”

Jail Class

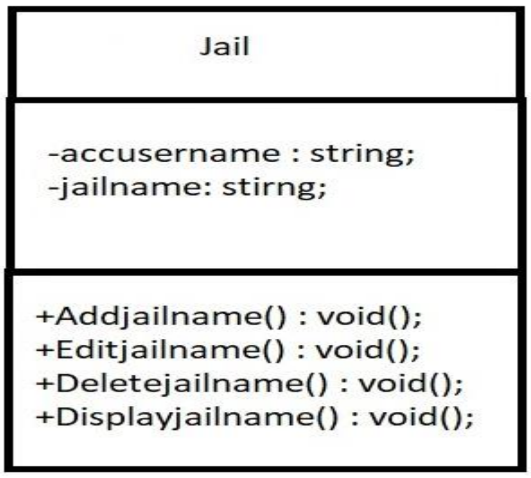
Code:

```
class Jail
{
    private string accusername;
    private string jailname;

     Public static void Addjailname()
    {
        // TO DO
    }
     Public static void Deletejailname()
    {
        // TO DO
    }
     Public static void Editjailname()
    {
        // TO DO
    }
     Public static void Displayjailname()
    {
        // TO DO
    }

}
```

UML Diagram:



Jail

-accusername : string;
-jailname: stirng;

+Addjailname() : void();
+Editjailname() : void();
+Deletejailname() : void();
+Displayjailname() : void();

"Police Station"

Complaint Class

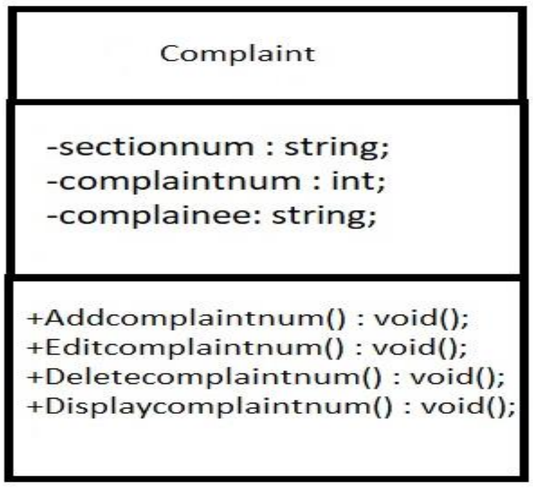
Code:

```
class Complaint
{
    private string sectionnum;
    private string complainee;
    private int complaintnum;

     Public static void Addcomplaintnum()
    {
        // TO DO
    }
     Public static void Deletecomplaintnum()
    {
        // TO DO
    }
     Public static void Editcomplaintnum()
    {
        // TO DO
    }
     Public static void Displaycomplaintnum()
    {
        // TO DO
    }

}
```

UML Diagram:



Complaint
-sectionnum : string;
-complaintnum : int;
-complainee: string;
+Addcomplaintnum() : void();
+Editcomplaintnum() : void();
+Deletecomplaintnum() : void();
+Displaycomplaintnum() : void();

"Apollo Hospitals"

Patient Class

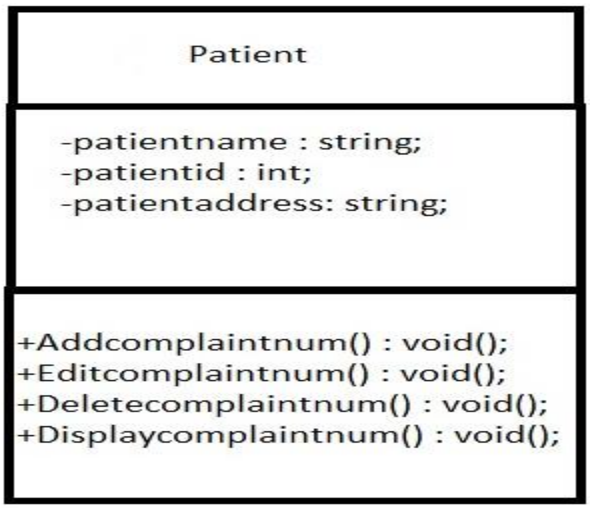
Code:

```
class Patient
{
    private string patientname;
    Private int patientid;
    private string patientaddress;

     Public static void Addpatientname()
    {
       // TO DO
    }
     Public static void Deletepatientname()
    {
       // TO DO
    }
     Public static void Editpatientname()
    {
       // TO DO
    }
     Public static void Displaypatientname()
    {
       // TO DO
    }

}
```

UML Diagram:


Patient
-patientname : string;
-patientid : int;
-patientaddress: string;
+Addcomplaintnum() : void();
+Editcomplaintnum() : void();
+Deletecomplaintnum() : void();
+Displaycomplaintnum() : void();

“Apollo Hospitals”

Doctor Class

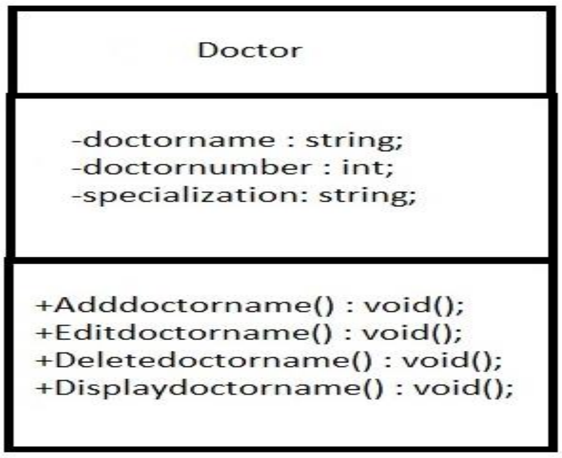
Code:

```
class Doctor
{
    private string doctorname;
    private int doctornumber;
    private string specialization;

     Public static void Adddoctorname()
    {
        // TO DO
    }
     Public static void Deletedoctorname()
    {
        // TO DO
    }
     Public static void Editdoctorname()
    {
        // TO DO
    }
     Public static void Displaydoctorname()
    {
        // TO DO
    }

}
```

UML Diagram:


Doctor
-doctorname : string;
-doctornumber : int;
-specialization: string;
+Adddoctorname() : void();
+Editdoctorname() : void();
+Deletedoctorname() : void();
+Displaydoctorname() : void();

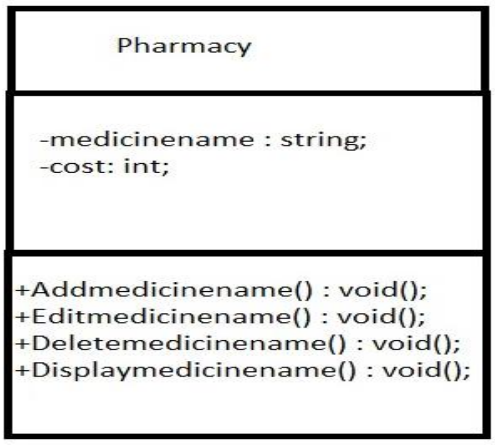
“Apollo Hospitals”
Pharmacy Class
Code:

```
class Pharmacy
{
    private string medicinename;
    private int cost;

     Public static void Addmedicinename()
    {
        // TO DO
    }
     Public static void Deletemedicinename()
    {
        // TO DO
    }
     Public static void Editmedicinetname()
    {
        // TO DO
    }
     Public static void Displaymedicinename()
    {
        // TO DO
    }

}
```

UML Diagram:



Pharmacy

-medicinename : string;
-cost: int;

+Addmedicinename() : void();
+Editmedicinename() : void();
+Deletemedicinename() : void();
+Displaymedicinename() : void();

"Apollo Hospitals"

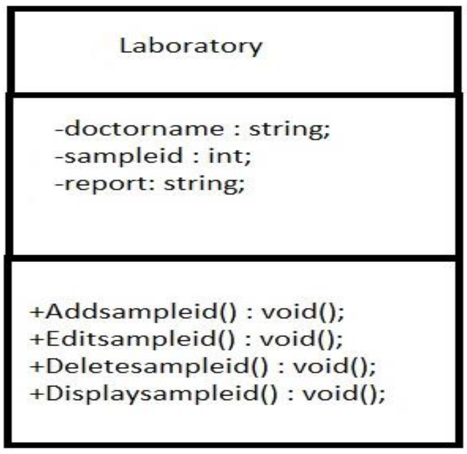
Laboratory Class

Code:

```
class Laboratory
{
    private int sampleid;
    private string report;
    private string doctorname;

     Public static void Addsampleid()
    {
        // TO DO
    }
     Public static void Deletesampleid()
    {
        // TO DO
    }
     Public static void Editsampleid()
    {
        // TO DO
    }
     Public static void Displaysampleid()
    {
        // TO DO
    }
```

UML Diagram:



```
Laboratory

-doctorname : string;
-sampleid : int;
-report: string;

+Addsampleid() : void();
+Editsampleid() : void();
+Deletesampleid() : void();
+Displaysampleid() : void();
```

“Apollo Hospitals”

Billing Class

Code:

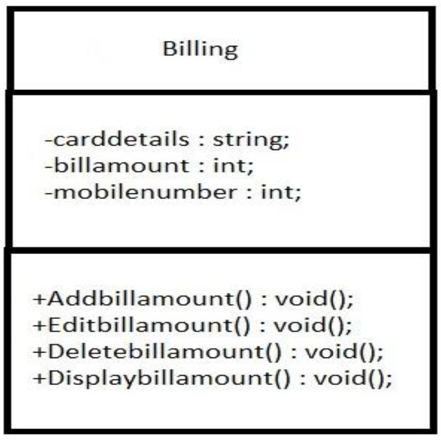
```
class Billing
{
    private string carddetails;
    private int billamount;
    private int mobilenumber;

     Public static void Addbillamount()
    {
        // TO DO
    }
     Public static void Deletebillamount()
    {
        // TO DO
    }
     Public static void Editbillamount()
```

```
        {
           // TO DO
        }
         Public static void Displaybillamount()
        {
           // TO DO
        }

     }
```

UML Diagram:



Billing

-carddetails : string;
-billamount : int;
-mobilenumber : int;

+Addbillamount() : void();
+Editbillamount() : void();
+Deletebillamount() : void();
+Displaybillamount() : void();

"Police Station"

FIR Class

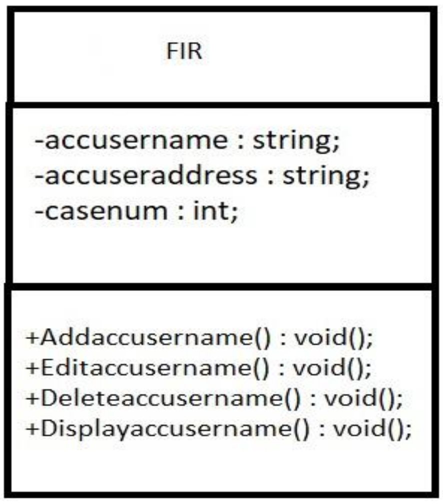
Code:

```
     class FIR
     {
        private string accusername;
        private string accuseraddress;
        private int casenum;

         Public static void Addaccusername()
        {
           // TO DO
        }
         Public static void Deleteaccusername()
        {
           // TO DO
        }
         Public static void Editaccusername()
        {
           // TO DO
        }
         Public static void Displayaccusername()
        {
           // TO DO
        }
```

```
    }
```

UML Diagram:



FIR

-accusername : string;
-accuseraddress : string;
-casenum : int;

+Addaccusername() : void();
+Editaccusername() : void();
+Deleteaccusername() : void();
+Displayaccusername() : void();

“Police Station”

Guns Class

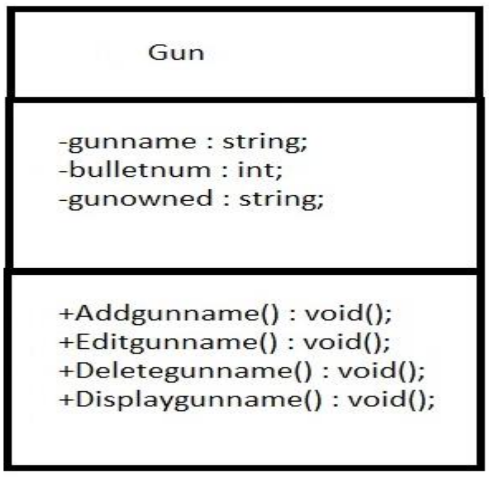
Code:

```
    class Guns
    {
       private string gunname;
       private int bulletnum;
       private string gunowned;

        Public static void Addgunname()
       {
          // TO DO
       }
        Public static void Deletegunname()
       {
          // TO DO
       }
        Public static void Editgunname()
       {
          // TO DO
       }
        Public static void Displaygunname()
       {
          // TO DO
       }

    }
```

UML Diagram:



Gun

-gunname : string;
-bulletnum : int;
-gunowned : string;

+Addgunname() : void();
+Editgunname() : void();
+Deletegunname() : void();
+Displaygunname() : void();

“Police Station”

employee Class

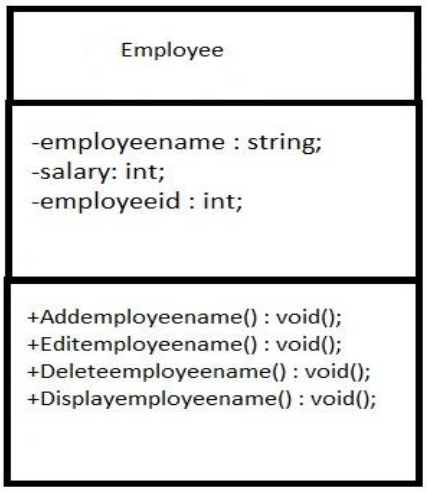
Code:

```
class Employee
{
   private string employeename;
   private int salary;
   private int employeeid;

    Public static void Addemployeename()
   {
      // TO DO
   }
    Public static void Deleteemployeename()
   {
      // TO DO
   }
    Public static void Editemployeename()
   {
      // TO DO
   }
    Public static void Displayemployeename()
   {
      // TO DO
   }

}
```

UML Diagram:



Employee

-employeename : string;
-salary: int;
-employeeid : int;

+Addemployeename() : void();
+Editemployeename() : void();
+Deleteemployeename() : void();
+Displayemployeename() : void();

“Police Station”

Jail Class

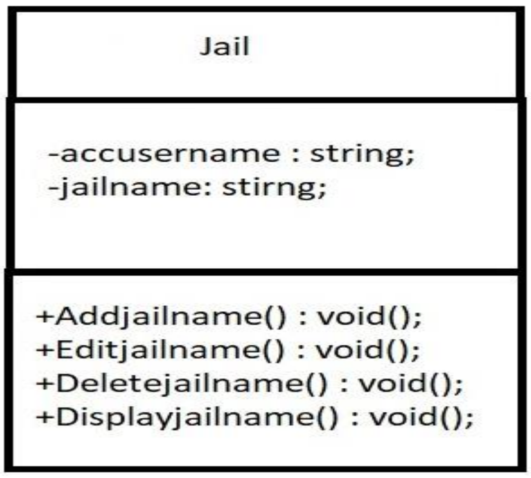
Code:

```
class Jail
{
    private string accusername;
    private string jailname;

     Public static void Addjailname()
    {
        // TO DO
    }
     Public static void Deletejailname()
    {
        // TO DO
    }
     Public static void Editjailname()
    {
        // TO DO
    }
     Public static void Displayjailname()
    {
        // TO DO
    }

}
```

UML Diagram:


Jail
-accusername : string;
-jailname: stirng;
+Addjailname() : void();
+Editjailname() : void();
+Deletejailname() : void();
+Displayjailname() : void();

“Police Station”

Complaint Class

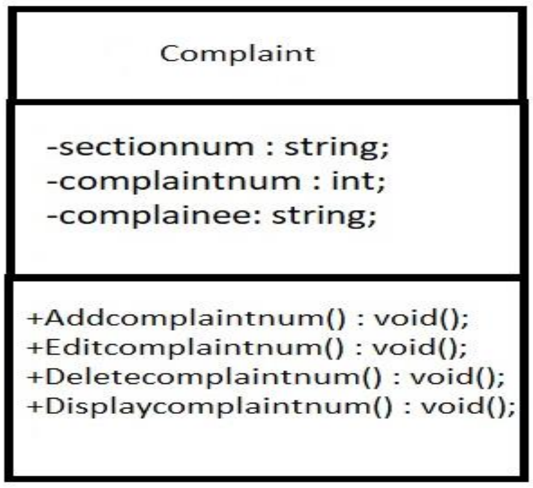
Code:

```
class Complaint
{
   private string sectionnum;
   private string complainee;
   private int complaintnum;

    Public static void Addcomplaintnum()
   {
      // TO DO
   }
    Public static void Deletecomplaintnum()
   {
      // TO DO
   }
    Public static void Editcomplaintnum()
   {
      // TO DO
   }
    Public static void Displaycomplaintnum()
   {
      // TO DO
   }

}
```

UML Diagram:



Complaint

-sectionnum : string;
-complaintnum : int;
-complainee: string;

+Addcomplaintnum() : void();
+Editcomplaintnum() : void();
+Deletecomplaintnum() : void();
+Displaycomplaintnum() : void();