

INDEX**Name of the Student:****Department:****Roll No:****Reg.No:****Lab Incharge:**

S.No.	Date	Name of the Experiment	Page No	Date of Sub	Marks	Staff Sign	Date
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							

CS8582 OBJECT ORIENTED ANALYSIS AND DESIGN LABORATORY**L T P C****0 0 4 2****OBJECTIVES:**

- To capture the requirements specification for an intended software system
- To draw the UML diagrams for the given specification
- To map the design properly to code
- To test the software system thoroughly for all scenarios
- To improve the design by applying appropriate design patterns.

Draw standard UML diagrams using an UML modeling tool for a given case study and map design to code and implement a 3 layered architecture. Test the developed code and validate whether the SRS is satisfied.

1. Identify a software system that needs to be developed.
2. Document the Software Requirements Specification (SRS) for the identified system.
3. Identify use cases and develop the Use Case model.
4. Identify the conceptual classes and develop a Domain Model and also derive a Class Diagram from that.
5. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence and Collaboration Diagrams
6. Draw relevant State Chart and Activity Diagrams for the same system.
7. Implement the system as per the detailed design
8. Test the software system for all the scenarios identified as per the usecase diagram
9. Improve the reusability and maintainability of the software system by applying appropriate design patterns.
10. Implement the modified system and test it for various scenarios

SUGGESTED DOMAINS FOR MINI-PROJECT:

Passport automation system.

Book bank

Exam registration

Stock maintenance system.

Online course reservation system

Airline/Railway reservation system

Software personnel management system

Credit card processing

e-book management system

Recruitment system

Foreign trading system

Conference management system

BPO management system

Library management system

Student information system

TOTAL: 60 PERIODS

OUTCOMES:

- Upon completion of this course, the students will be able to:
- Perform OO analysis and design for a given problem specification.
- Identify and map basic software requirements in UML mapping.
- Improve the software quality using design patterns and to explain the rationale behind applying specific design patterns
- Test the compliance of the software with the SRS.

HARDWARE REQUIREMENTS Standard PC

SOFTWARE REQUIREMENTS 1. Windows 7 or higher, 2. ArgoUML that supports UML 1.4 and higher, 3. Selenium, JUnit or Apache JMeter

LIST OF EXPERIMENTS

S.No	Experiment Name	Page No
1.	Passport Automation System	1
2.	Book Bank	9
3.	Exam Registration	15
4.	Stock maintenance system	19
5.	Online course reservation system	24
6.	E-ticketing	29
7.	Software personnel management system	34

EX.NO: 1 PASSPORT AUTOMATION SYSTEM**AIM:**

To create an application for passport automation system using uml concept

SCOPE:

The passport automation system provides an online interface to the user where they can fill in their personal details and submit the necessary documents to apply for passport.

DEFINITIONS, ACRONYMS AND ABBREVIATIONS:

Administrator – Refers to the super user who is the central authority who has been nested with the privilege to manage entire system.

Applicant – One who wishes to obtain the passport.

OVERVIEW:

Classified into two main sections namely, Overall Description and Overall Requirements.

OVERALL DESCRIPTION:

This section describes about the use case modeling survey, assumption and dependencies.

❖ Use- Case Model Survey:

Applicant applies for passport online, which is verified by the administrator; the police officer has the authority to issue/reject the passport. After the interview of an applicant the passport is issued by post.

- **Introduction:**

Use case model is a part of UML that defines the functionality of each actor or person involved in the project, it is done by defining use case for each actor, one or more actors can be given an use case.

- **Survey Description:**

Enter information:

The user enters his/her personal information with proof and references.

Queries:

The user can enquire about the issue of passport.

Transaction:

The user can view about the status of passport issue.

Proceed/Reject:

The passport can be issued or rejected by the passport officer and police.

❖ Assumptions and Dependencies:

The applicant and administrator must have basic knowledge of computers and language in the internet. Applicants may be required to scan the documents and send.

Overall Requirements:

The system gathers requirements such as software to run the database for storing the details of the customer.

❖ Use-Case Specification:

The candidate enters into the passport site and feed the personal information. The passport department forwards the information to the police dept. to enquire about the candidate. If everything goes well the acknowledgement number is provided to the candidate for further clarification with which the candidate can check his/her status.

❖ Functionality:

The passport automation system is used to facilitate quick processing of acquiring passport.

❖ Usability:

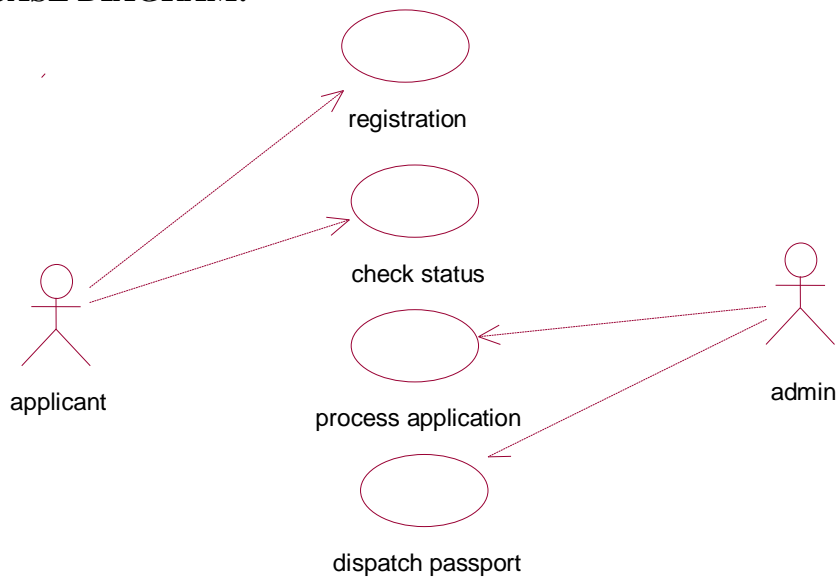
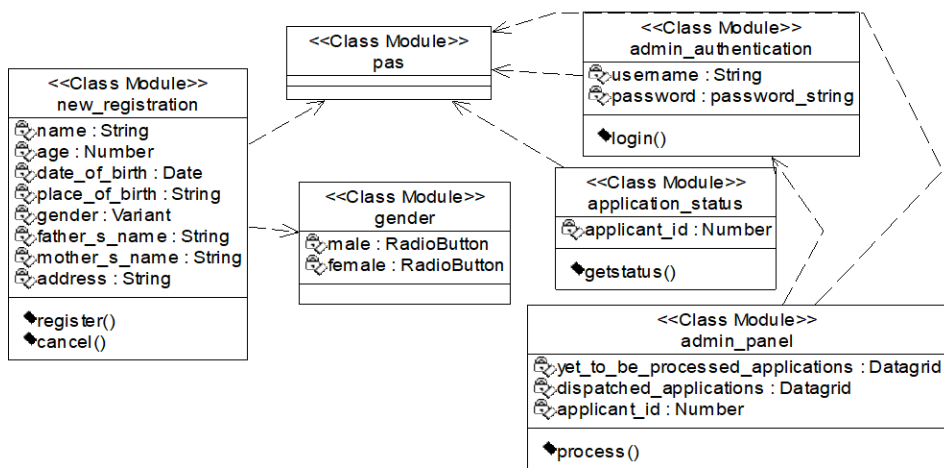
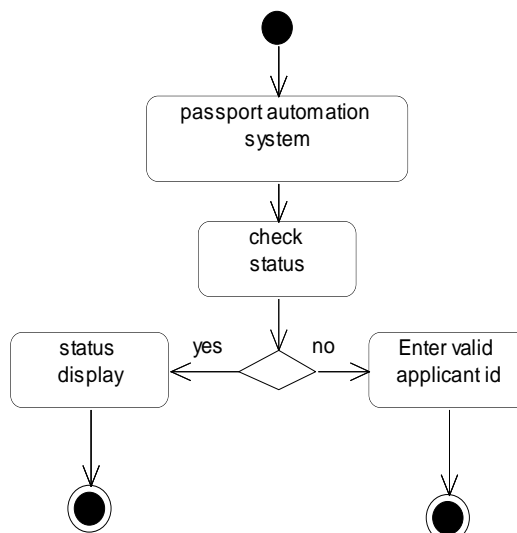
The passport automation system is used to get passport with quick processing and delivery.

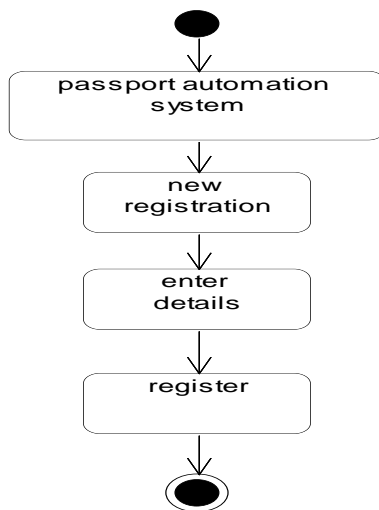
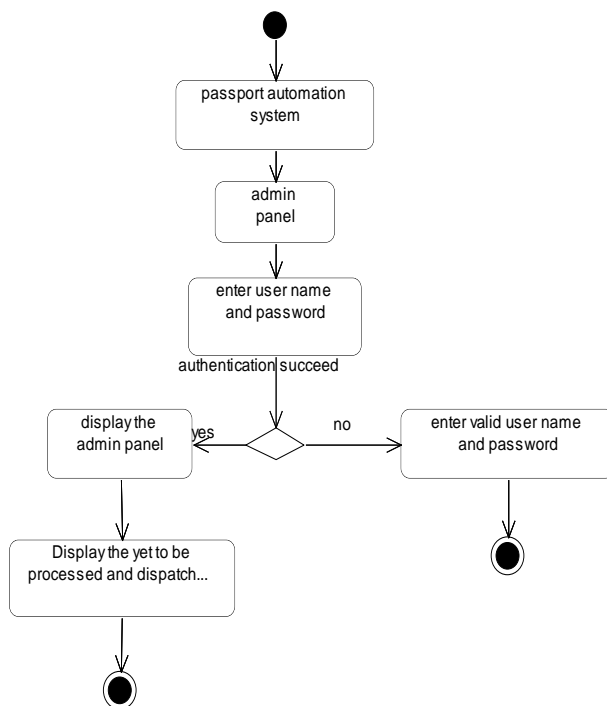
❖ Reliability:

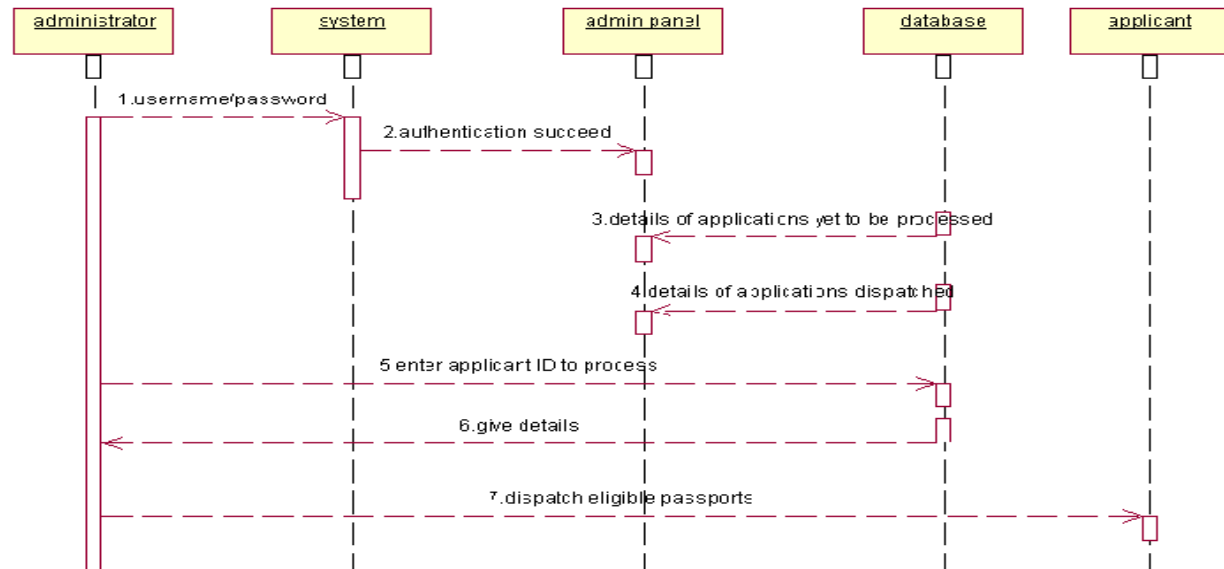
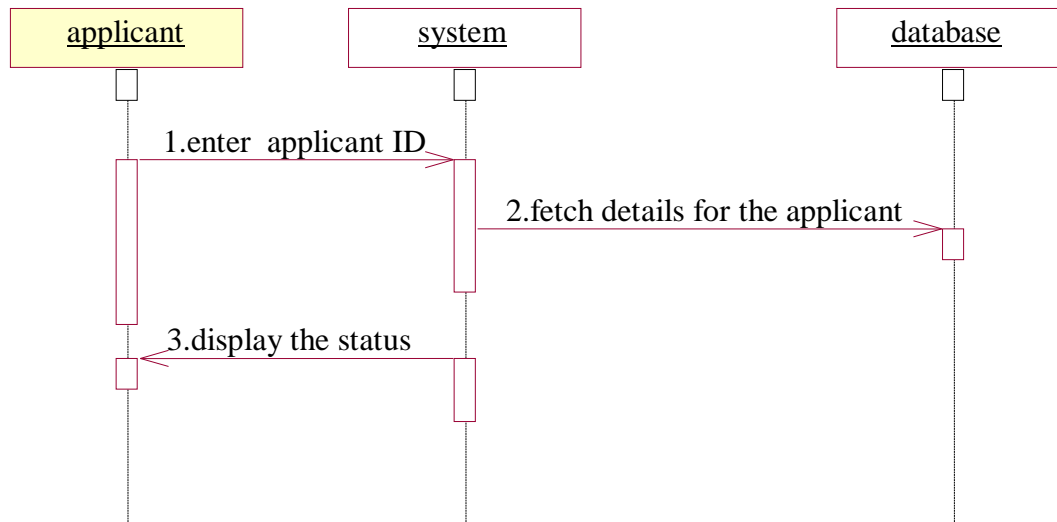
The passport automation system assures to provide authenticated passports and confidential.

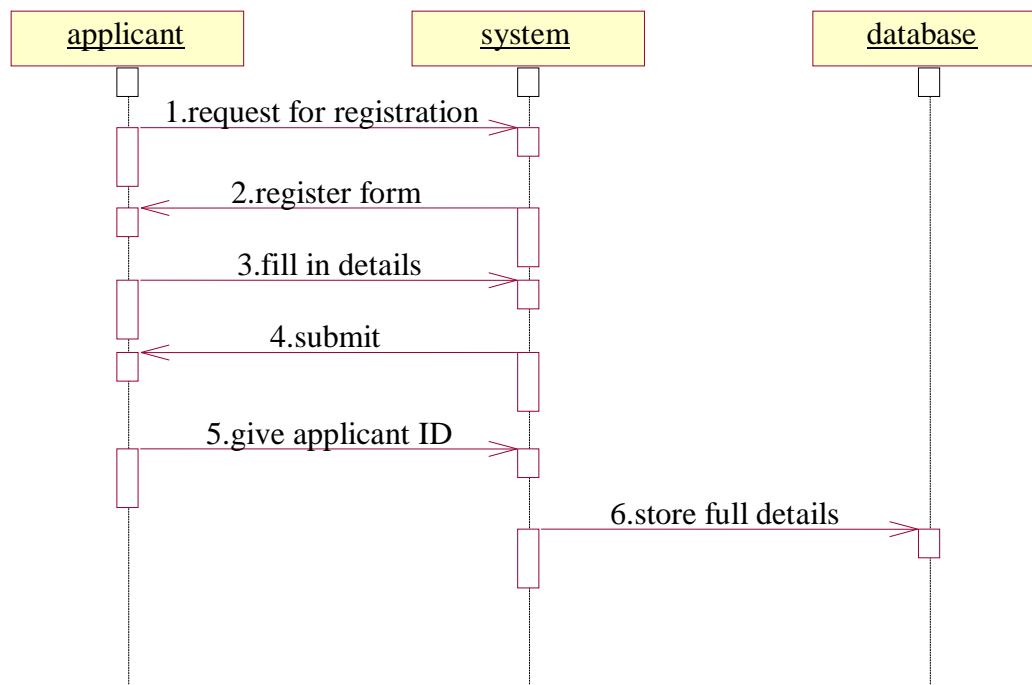
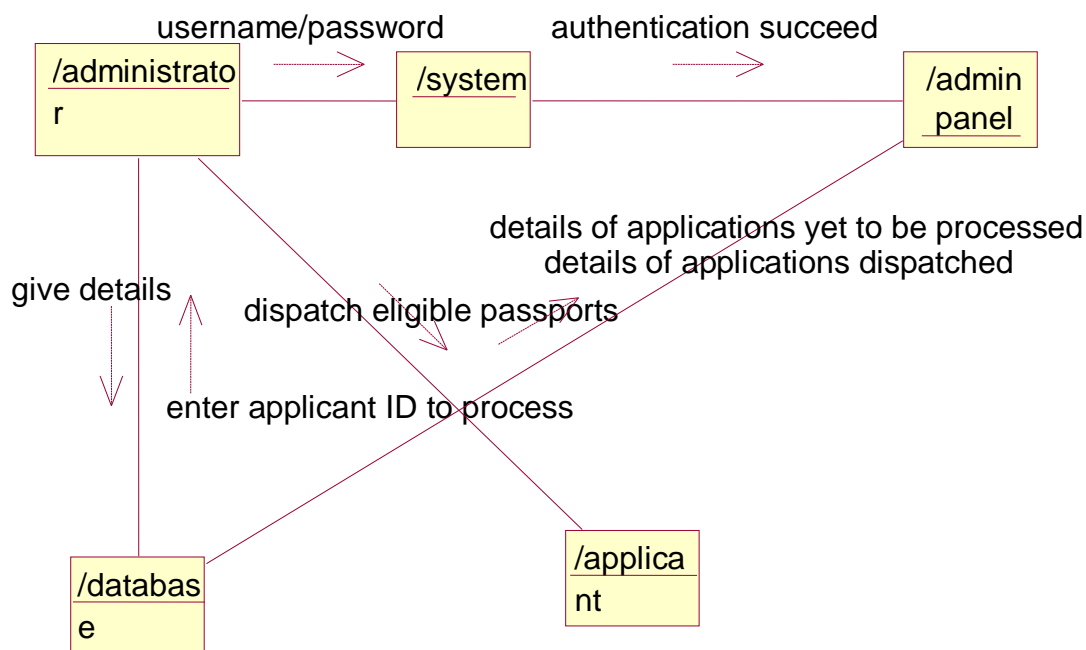
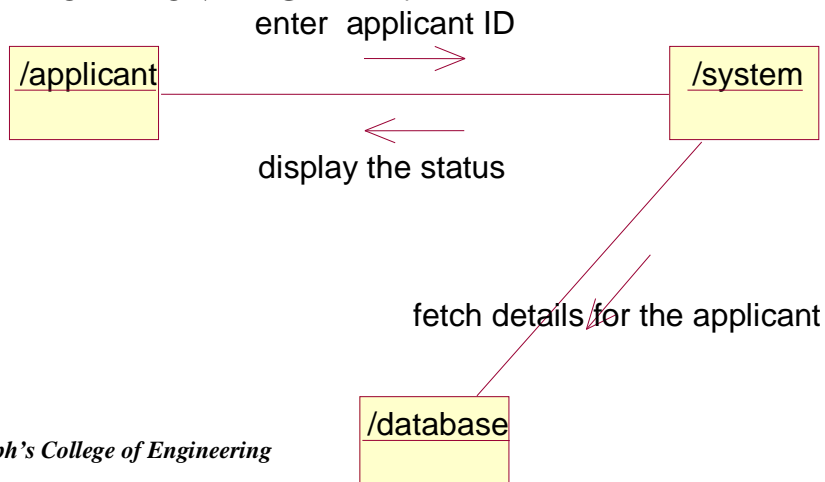
❖ Performance:

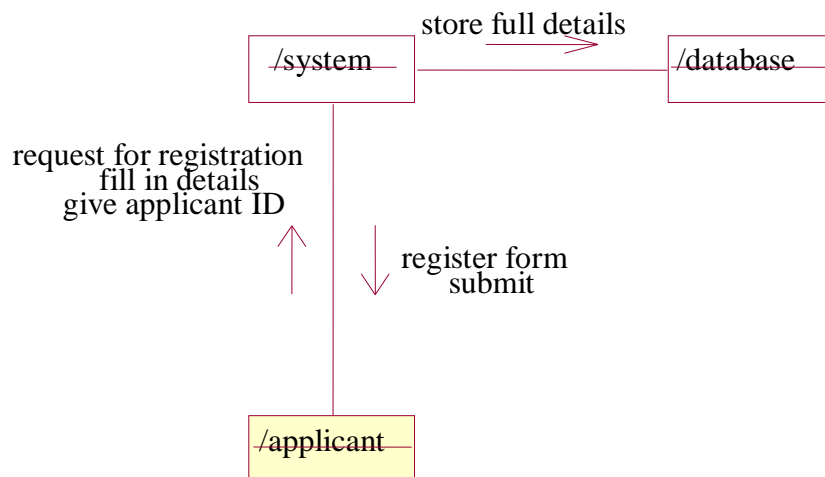
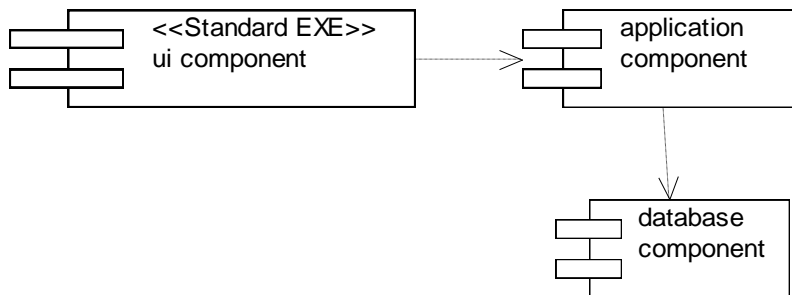
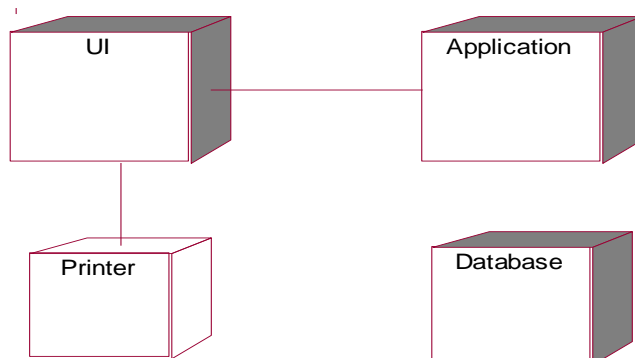
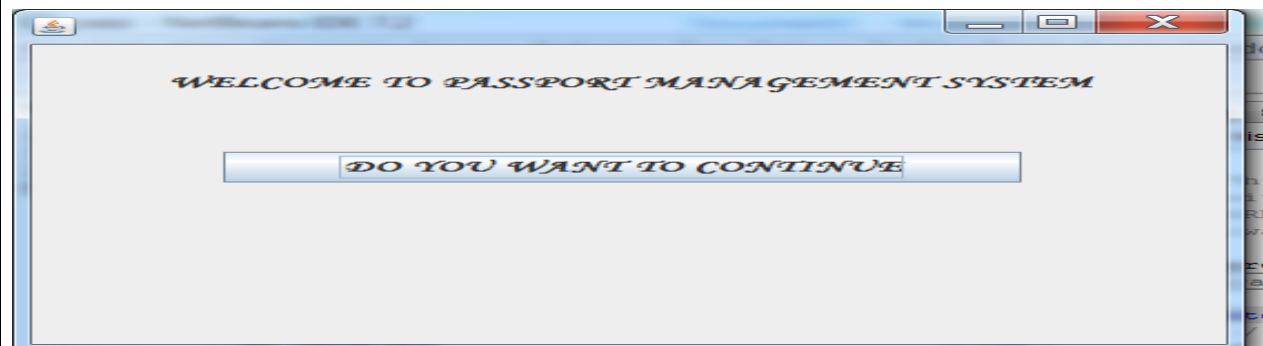
The passport automation system performs immediate actions once the registrations are done.

USE CASE DIAGRAM:**CLASS DIAGRAM:****ACTIVITY DIAGRAM 1:**

ACTIVITY DIAGRAM 2:**ACTIVITY DIAGRAM 3:**

SEQUENCE DIAGRAM 1:**SEQUENCE DIAGRAM 2:****SEQUENCE DIAGRAM 3:**

**COLLABORATION DIAGRAM 1:****COLLABORATION DIAGRAM 2:**

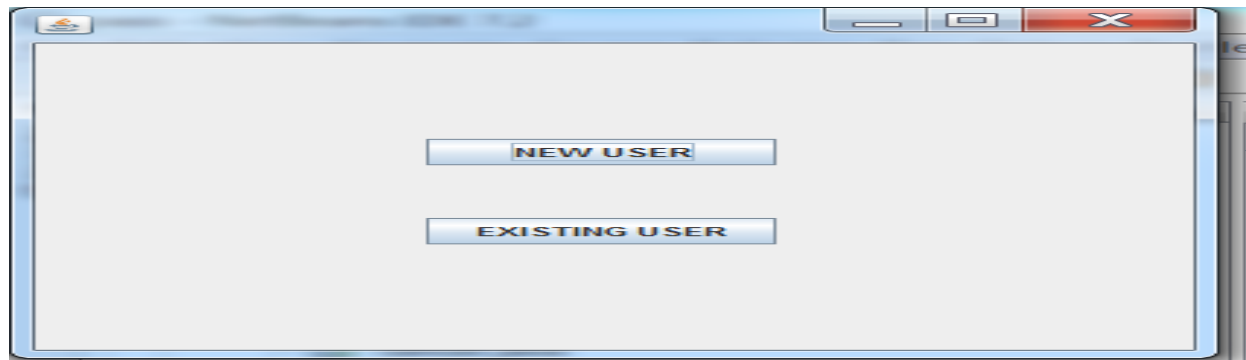
COLLABORATION DIAGRAM 3:**COMPONENT DIAGRAM:****DEPLOYMENT DIAGRAM:****CODING AND FORMS**

package pass;

```
public class Main {  
    public static void main(String[] args) {  
        // TODO code application logic here  
        new pass().setVisible(true);  
    }  
}
```



```
package pass;  
  
public class select extends javax.swing.JFrame {  
    /** Creates new form select */  
    public select() {  
        initComponents();  
    }  
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
        // TODO add your handling code here:  
        setVisible(false);  
        new user().setVisible(true);  
    }  
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
        // TODO add your handling code here:  
        setVisible(false);  
        new login().setVisible(true);  
    }  
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
        // TODO add your handling code here:  
        setVisible(false);  
        new login().setVisible(true);  
    }  
    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
        // TODO add your handling code here:  
        setVisible(false);  
        new login().setVisible(true);  
    }  
}
```



```

package pass;
public class user extends javax.swing.JFrame {
    /** Creates new form user */
    public user() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        new registre().setVisible(true);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        new alogin().setVisible(true);
    }
    import java.sql.Connection;
    import java.sql.DriverManager;
    import java.sql.PreparedStatement;
    import java.sql.ResultSet;
    import java.sql.SQLException;
    import java.sql.Statement;
    import javax.swing.JOptionPane;
    import net.proteanit.sql.DbUtils;
    public class registre extends javax.swing.JFrame {
        /** Creates new form registre */
        public registre() {
            initComponents();
        }
        private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling code here:
            String name=jTextField3.getText();
            String dob=jTextField1.getText();
            String username=jTextField4.getText();
            String password=jPasswordField1.getText();
            String adds=JTextArea1.getText();
            String id=jTextField2.getText();
            try {

```

```
PreparedStatement psm=null;
Class.forName("com.mysql.jdbc.Driver")
conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/stock", "root", "");
String sql="insert into pass
values('"+name+"','"+dob+"','"+username+"','"+password+"','"+adds+"','"+id+"')";
Statement stm=conn.createStatement();
stm.executeUpdate(sql);
JOptionPane.showMessageDialog(null, "Successfully Registered!!");
new cancel().setVisible(true);
} catch(Exception e) {
JOptionPane.showMessageDialog(null, "Error Occured!");
}
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
setVisible(true);
new cancel().setVisible(true);
}
```

RESULT:

Thus the application for passport automation system using uml concept has been executed.

Viva Questions:

1. What is UML?
2. Define modeling in UML.
3. What are the advantages of UML?
4. What are the different views in UML?
5. Define SDLC

EX NO: 2**BOOK BANK****AIM**

To create an application for book bank using uml concept.

PROBLEM STATEMENT:

The main aim of the project deals with checking for availability of books and there by supplying and returning of the books .The main objective of designing and developing a book bank management system which proves to be manually beneficial. (i.e) benefits both the students and the administration. This provides the overall information of the supplying and returning of books book bank system is where the books can be collected every semester and must be returned at the end of semester. The system must have option for new members to enroll for membership by paying deposit. A provision for getting six or seven books per semester. Membership can be renewed by using register number. A database must be maintained to guide the issue to mark the details of students. Search option must be provided so that the members can search for the availability of a particular book.

MODULE DESCRIPTION:**MODULE 1: (HOME PAGE)**

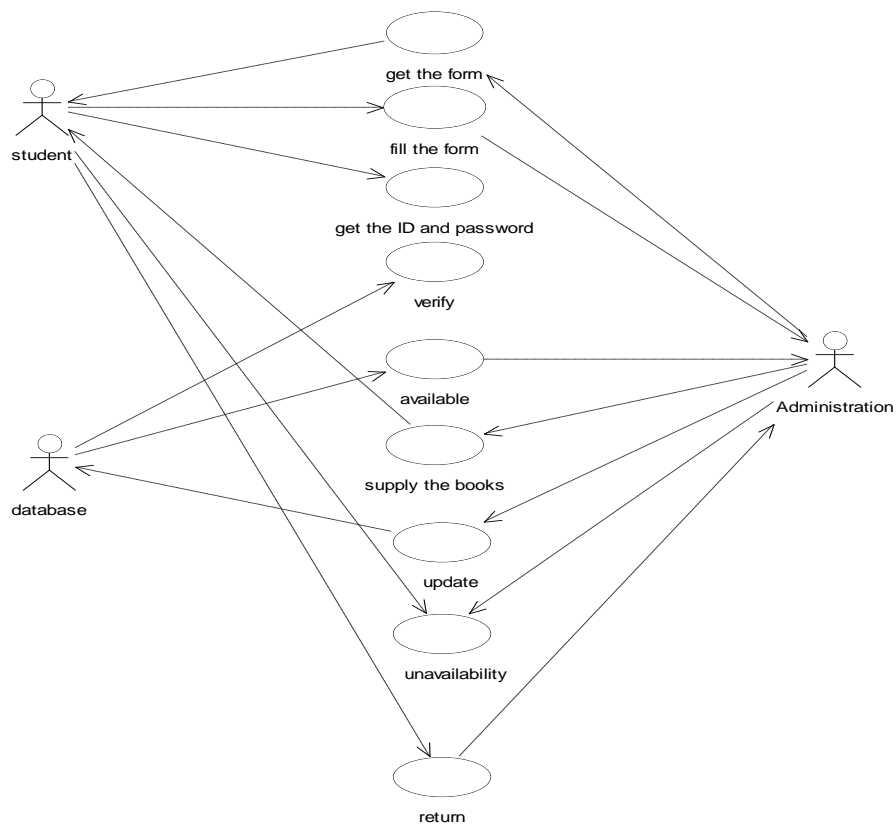
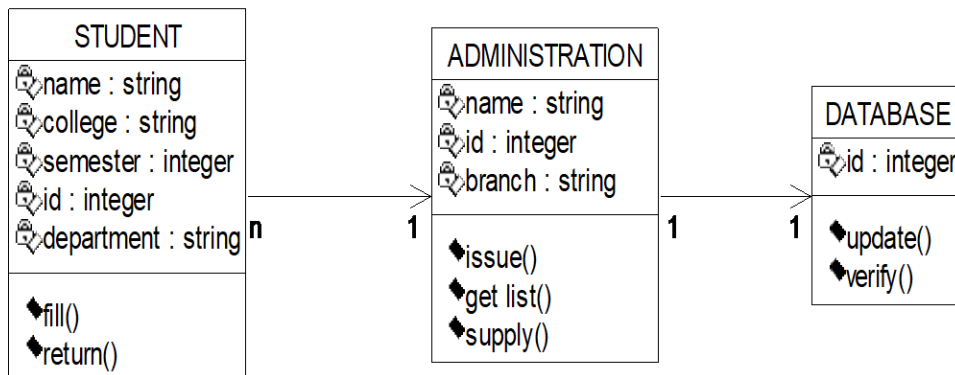
It contains the home page which has the login and registration.

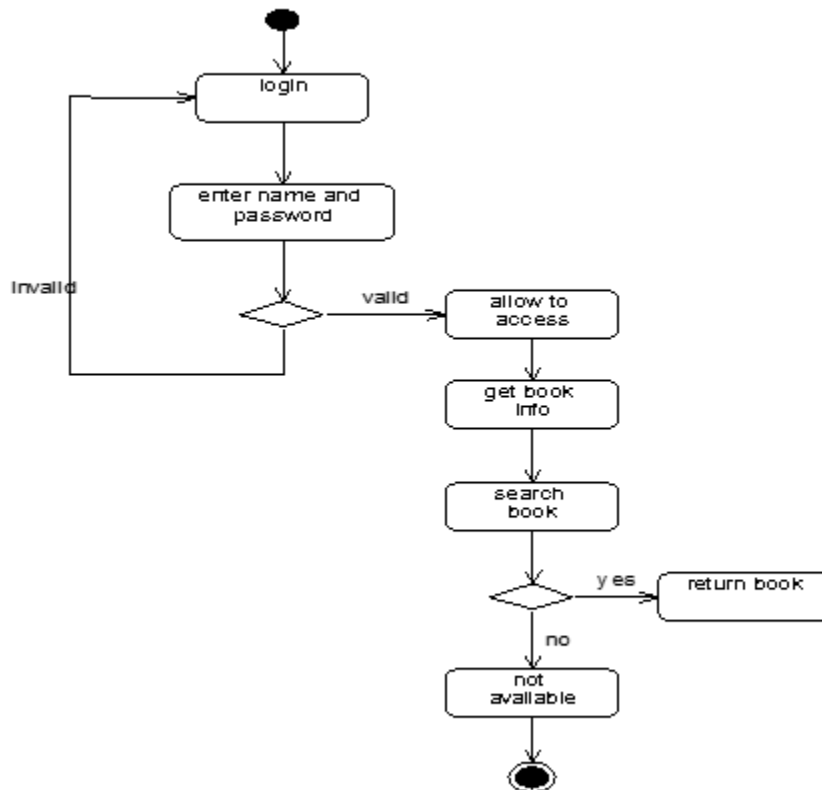
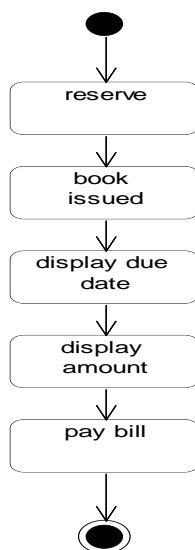
MODULE 2: (ADMINISTRATION FORM)

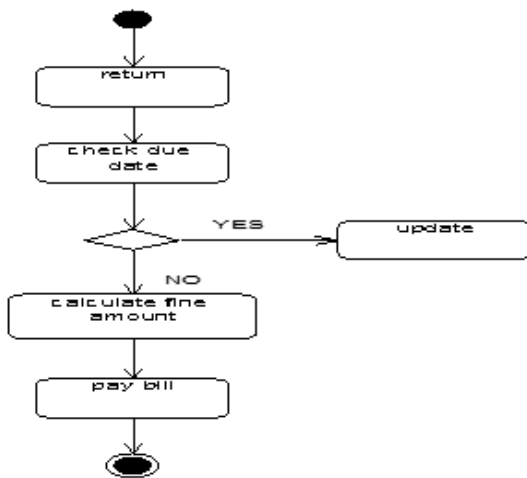
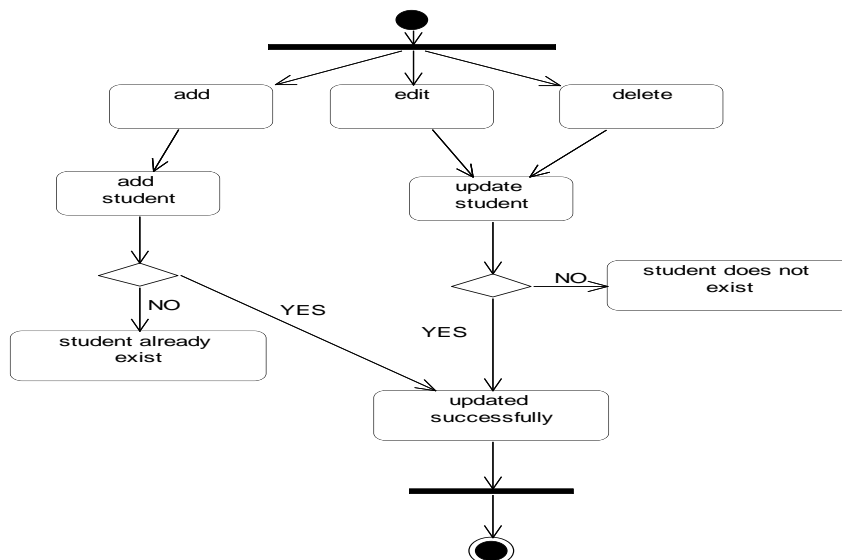
The students' details are gathered and the information is updated in the database.

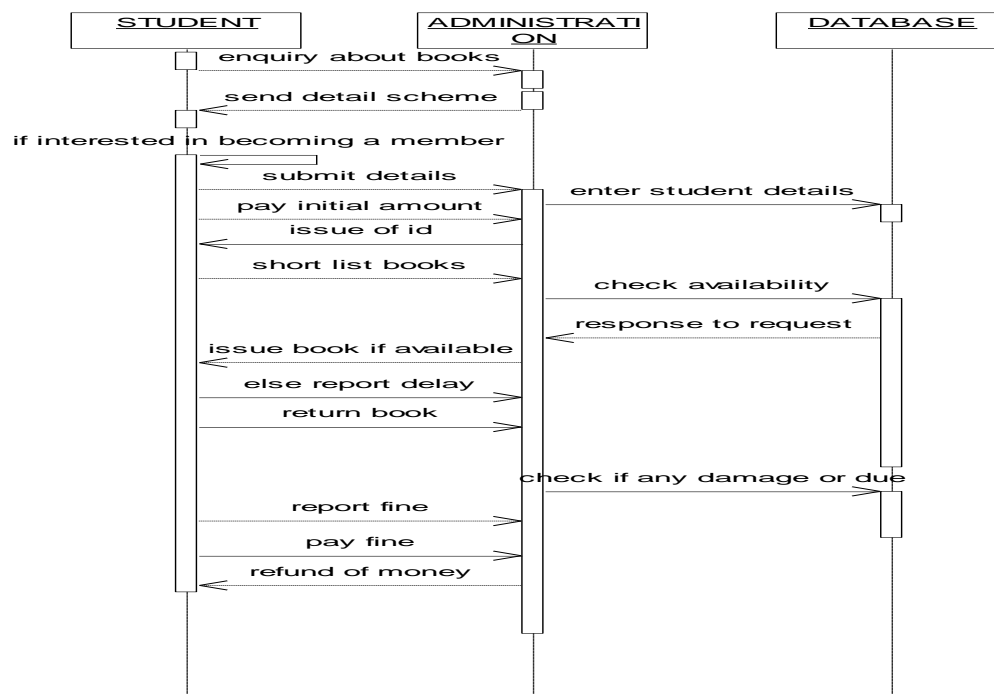
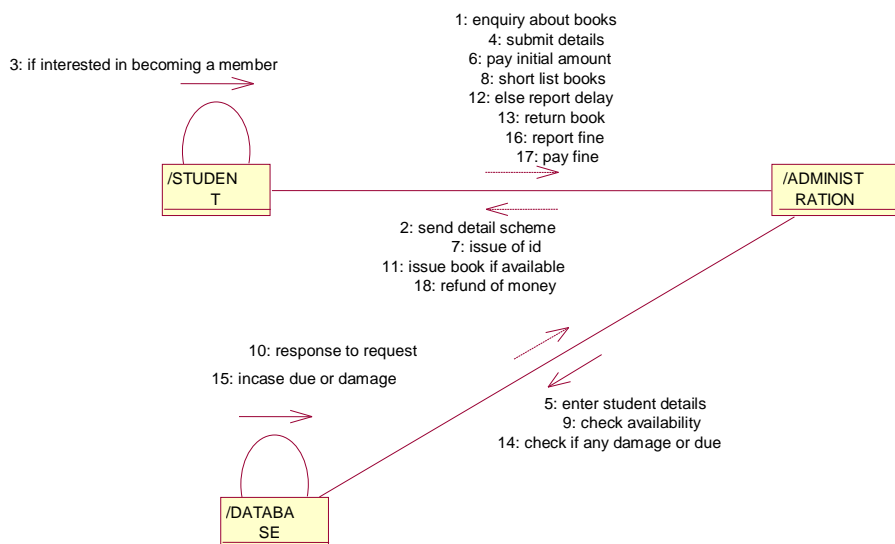
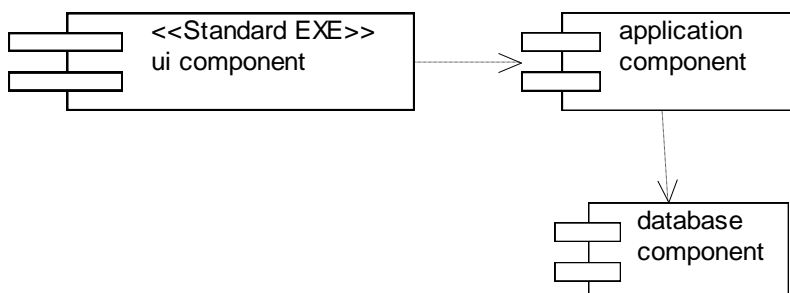
MODULE 3:

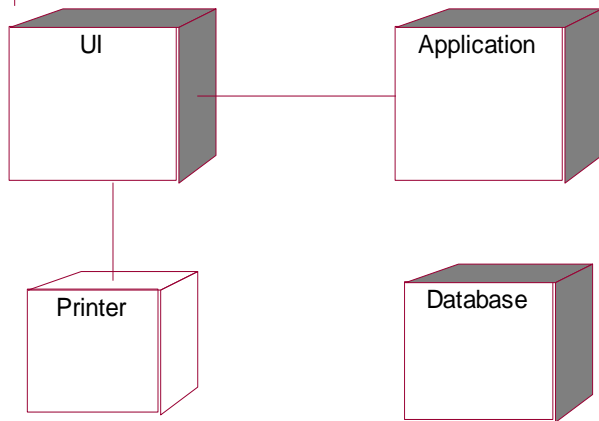
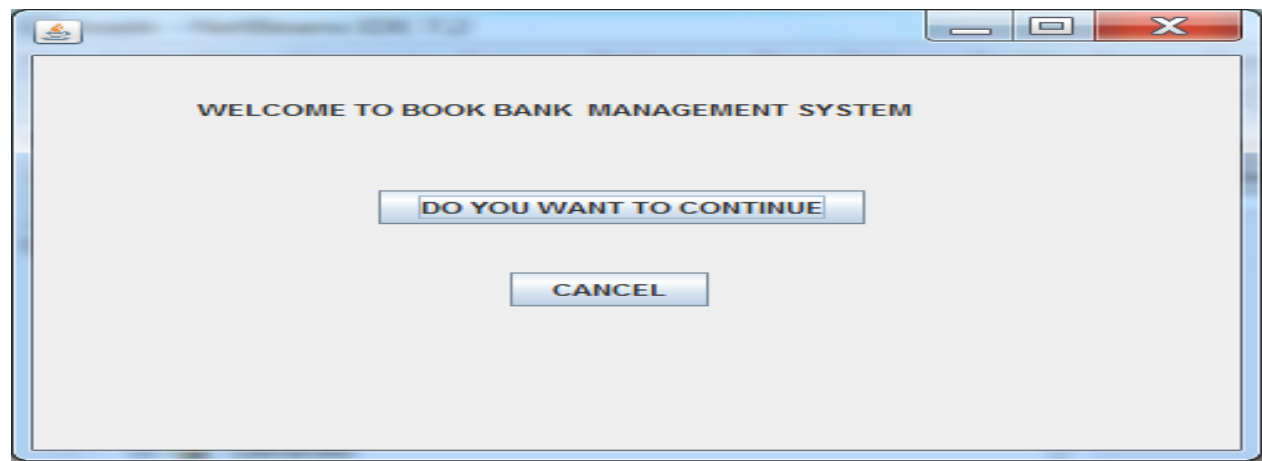
The member can login and check the availability of books.

USE CASE DIAGRAM:**CLASS DIAGRAM:**

ACTIVITY DIAGRAM 1: (LOGIN)**ACTIVITY DIAGRAM 2: (ISSUE A BOOK)**

ACTIVITY DIAGRAM 3: (RETURN A BOOK)**ACTIVITY DIAGRAM 3: (STUDENT DETAILS)**

SEQUENCE DIAGRAM:**COLLABORATION DIAGRAM:****COMPONENT DIAGRAM:**

DEPLOYMENT DIAGRAM:**CODING AND FORMS**

```

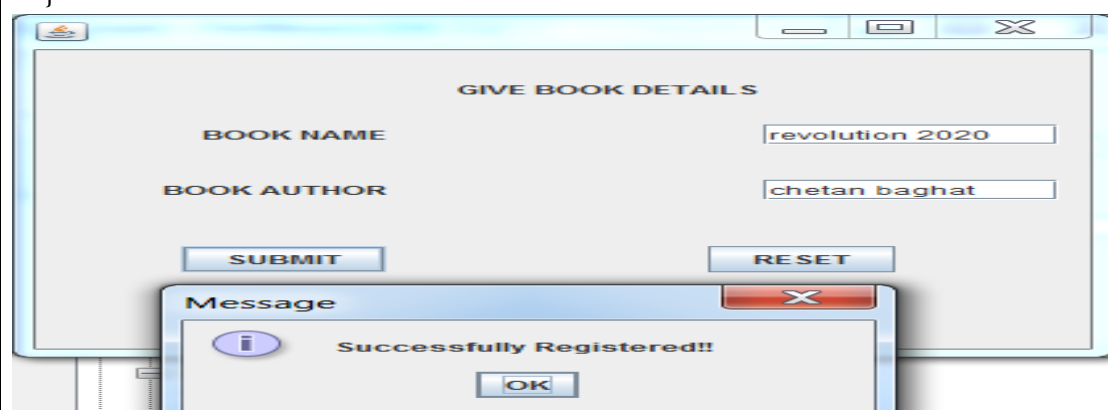
public class book extends javax.swing.JFrame {
    public book() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        new login().setVisible(true);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new cancel().setVisible(true); }
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
public class login extends javax.swing.JFrame {
    /** Creates new form login */

```

```

public login() {
    initComponents();
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String s1=jTextField1.getText();
    String s2=jPasswordField1.getText();
    System.out.println(s1);
    System.out.println(s2);
    try
    {
        ResultSet rs;
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection connect=DriverManager.getConnection("jdbc:odbc:bookbank");
        String s="select * from pass where username = '" +s1+"'and password = '"+s2+"'";
        Statement stmt=connect.createStatement();
        rs=stmt.executeQuery(s);
        int count=0;
        while(rs.next())
        {
            count++;
        }
        if(count==1)
            new bb().setVisible(true);
        else
            JOptionPane.showMessageDialog(null,"Wrong Password");
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setVisible(false);
    new login().setVisible(true);
}

```

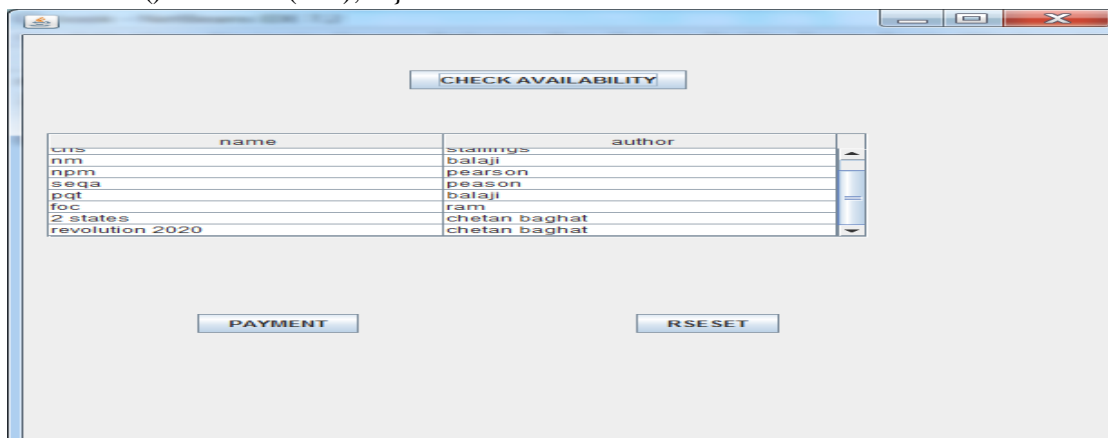


```
import java.sql.Connection;
```

```

import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
public class bb extends javax.swing.JFrame {
/** Creates new form bb */
    public bb() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String name=jTextField1.getText();
        String author=jTextField2.getText();
        try
        {
            PreparedStatement psm=null;
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection conn=DriverManager.getConnection("jdbc:odbc:bookbank");
            String sql="insert into book values('"+name+"','"+author+"')";
            Statement stm=conn.createStatement();
            stm.executeUpdate(sql);
            JOptionPane.showMessageDialog(null, "Successfully Registered!!");
            setVisible(false);
            new verify().setVisible(true);
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null, " Successfully Registered!!");
            new verify().setVisible(true);
        }
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new bb().setVisible(true);
    }

```



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

```
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;
public class verify extends javax.swing.JFrame {
    /** Creates new form verify */
    public verify() {
        initComponents();
    }
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new verify().setVisible(true);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new payment().setVisible(true);
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try
        {
            ResultSet rs;
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection connect=DriverManager.getConnection("jdbc:odbc:bookbank");
            String sq="select * from book" ;
            Statement stmt=connect.createStatement();
            rs=stmt.executeQuery(sq);
            System.out.println("query executed");
            jTable1.setModel(DbUtils.resultSetToTableModel(rs));
        } catch(Exception e)
        {
            System.out.println(e.getMessage()); } }
}
```

RESULT:

Thus the application for book bank using uml concept has been executed.

Viva Questions:

1. Explain the types of diagrams in UML.
2. Building blocks of UML
3. Relationship in UML.
4. Diagrams in UML.
5. What are the three types of modeling in UML?

EX NO:3**EXAM REGISTRATION SYSTEM****AIM:**

To create an application for exam registration system using uml concept.

PROBLEM STATEMENT:

To create an Exam Registration software that will meet the needs of the applicant and help them in registering for the exam ,enquiry about the registered subject,modification in database and cancellation for the registered project.

OVERALL DESCRIPTION:

The Exam Registration system is an integrated system that has four modules as part of it.The four modules are:

❖ **REGISTRATION FOR EXAMS**

- In this module,the user can select the subject to register for the exam,enquire about the registered subject,modification in the student database,cancelling the registered subject.

❖ **FORM FOR REGISTRATION**

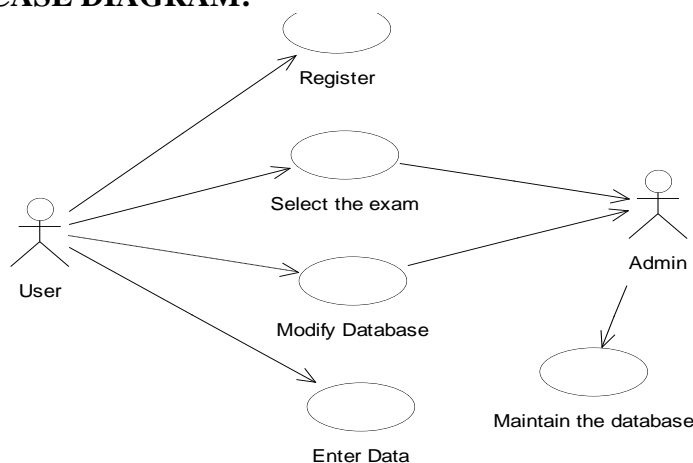
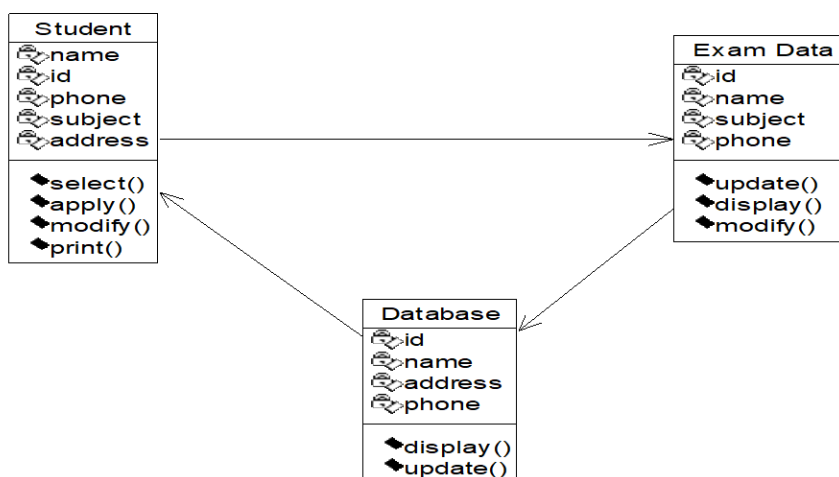
- In this module the user can apply for the exam by giving the details about the candidate and selecting the subject for registration.

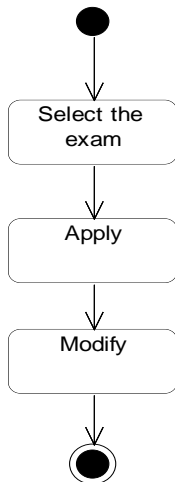
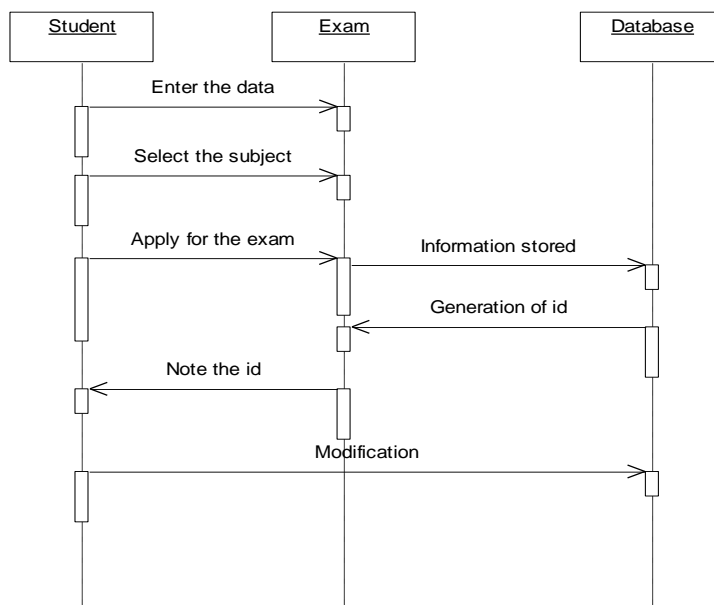
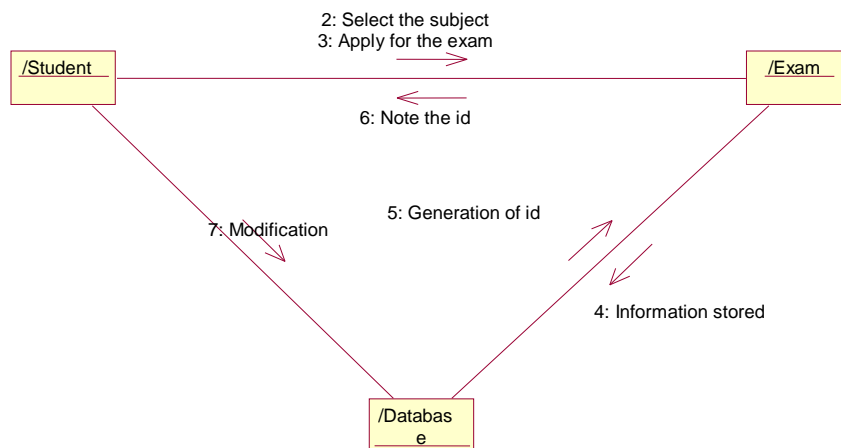
❖ **MODIFICATION IN THE DATABASE**

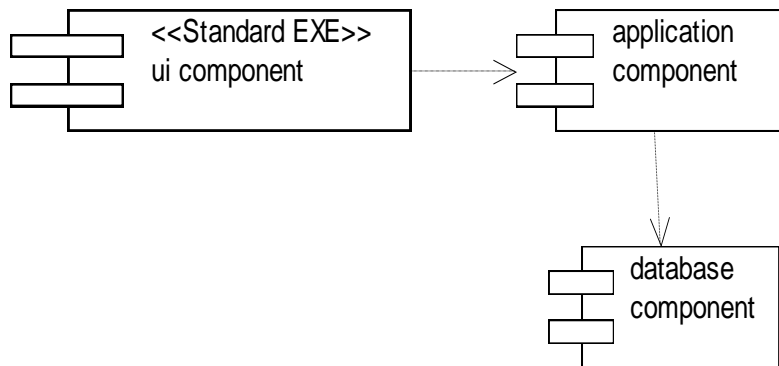
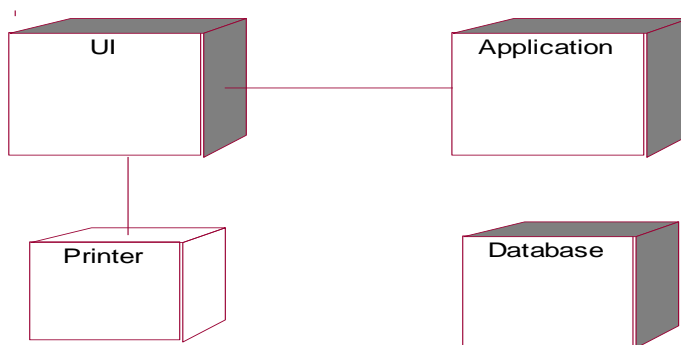
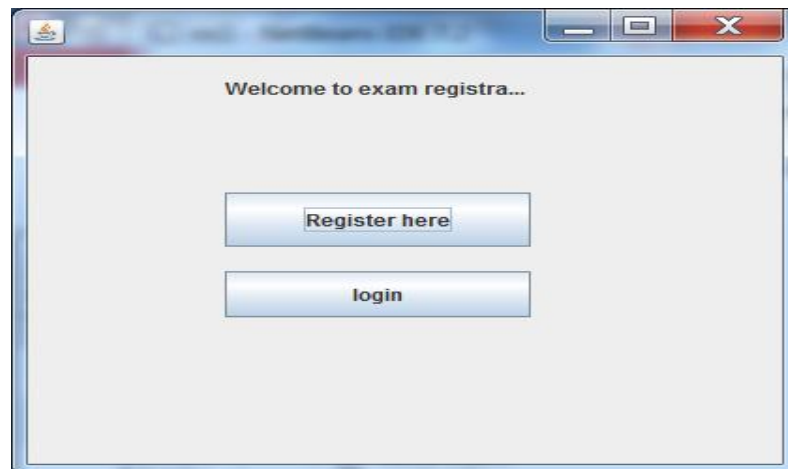
- In this module the user can change the data like the phone number, address can be done.

❖ **CANCELLATION FOR THE REGISTERED SUBJECT**

In this module the user can cancel the name which is registered for the exam.

USE CASE DIAGRAM:**CLASS DIAGRAM:**

ACTIVITY DIAGRAM:**SEQUENCE DIAGRAM:****COLLABORATION DIAGRAM:**

COMPONENT DIAGRAM:**DEPLOYMENT DIAGRAM:****IMPLEMENTATION:**

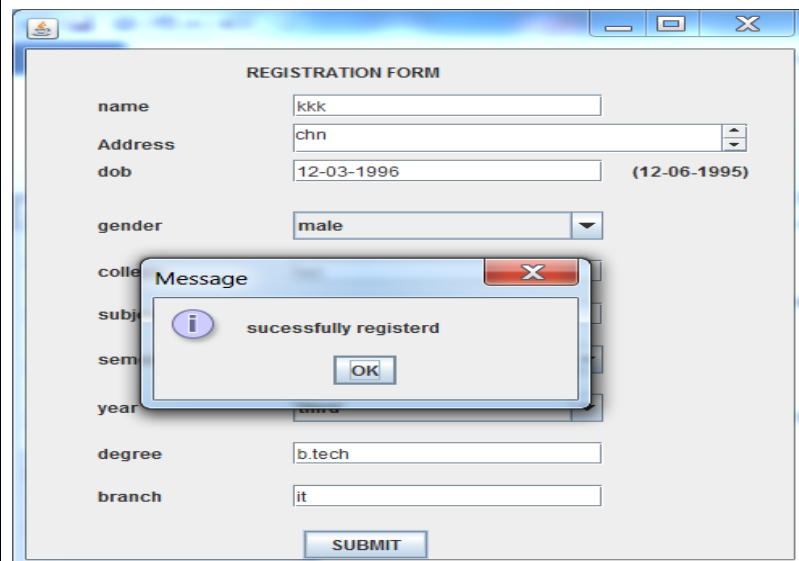
```

public class opening extends javax.swing.JFrame {
    public opening() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new register().setVisible(true);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new login().setVisible(true);
    }
}
  
```

```

    }
    public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
    new opening().setVisible(true);
    }
    });
    }
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration}

```



```

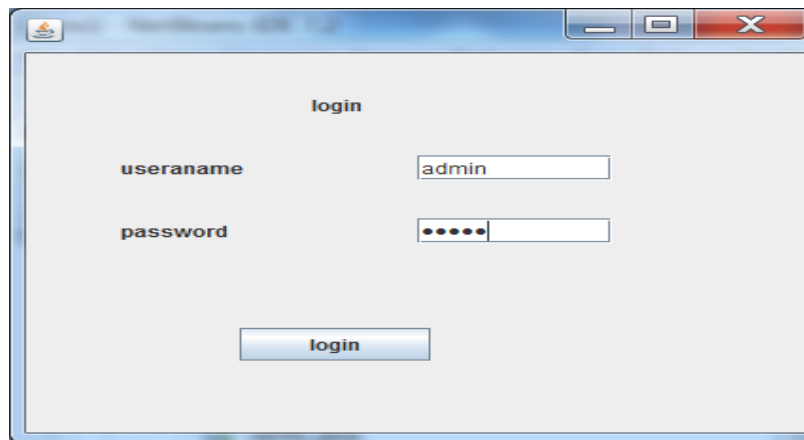
public class register extends javax.swing.JFrame {
    public register() {
    initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    // student sd=new student();
    String a=jTextField1.getText();
    String b=JTextArea1.getText();
    String c=jTextField2.getText();
    String d=(String) jComboBox1.getSelectedItem();
    String e=jTextField3.getText();
    String f=jTextField4.getText();
    String g=(String) jComboBox2.getSelectedItem();
    String h=(String) jComboBox3.getSelectedItem();
    String i=jTextField5.getText();
    String j=jTextField6.getText();
    student s=new student(a,b,c,d,e,f,g,h,i,j);
    public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {

```



```
public void run() {
new register().setVisible(true);
    }
});
}

STUDENT:
import java.sql.*;
import javax.swing.JOptionPane;
public class student
{
public String name;
private String address;
private String dob;
private String gender;
private String college;
private String sub;
private String sem;
private String year;
private String deg;
private String branch;
private int pay;
    //public database theDatabase;
    /**
    @roseuid 54BE161601DA
    */
    public student(String
name,Stringaddress,Stringdob,Stringgender,Stringcollege,Stringsubjects,Stringsem,Stringyear,Stringdeg,
String branch )
    {
try{
System.out.println(name+address+dob+gender+college+subjects+sem+year+deg+branch);
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection conn=DriverManager.getConnection("jdbc:odbc:exam");
    String sql="insert into studentdetails
values('"+name+"','"+address+"','"+dob+"','"+gender+"','"+college+"','"+subjects+"','"+sem+"','"+year+"','"+deg+"','"+branch+"')";
    Statement stm=conn.createStatement();
stm.executeUpdate(sql);
JOptionPane.showMessageDialog(null, "sucessfullyregisterd") }
catch(Exception ex)
    {
System.out.println(ex.getMessage());
JOptionPane.showMessageDialog(null, "error occured in connection");
    }
}
```



```

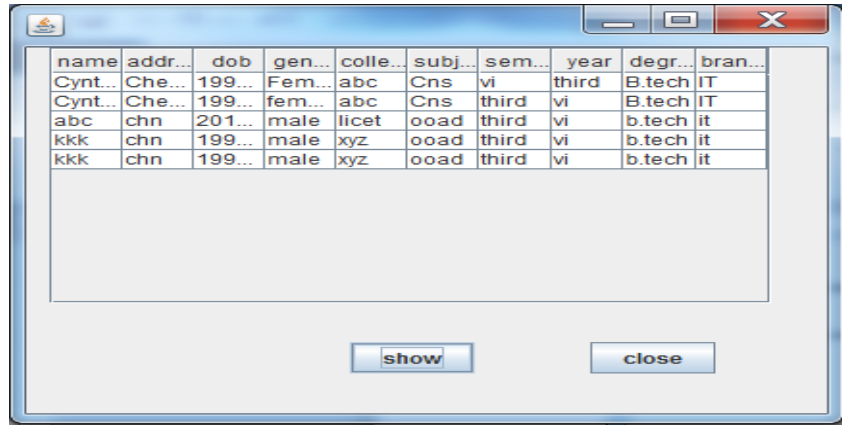
public class login extends javax.swing.JFrame {
    public login() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String s1=jTextField1.getText();
        String s2=jPasswordField1.getText();
        databasedb=new database(s1,s2);
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new login().setVisible(true);
            }
        });
    }
}
DATABASE:
import java.sql.*;
import javax.swing.JOptionPane;
public class database
{
    /**
     * @roseuid 54BE1616026E
     */
    public database(String s1,String s2)
    {
        try{
            PreparedStatement pstmt;
            ResultSet rs;
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection connect=DriverManager.getConnection("jdbc:odbc:exam");
            String sq="select * from userdetails where username = '"+s1+"'and password = '"+s2+"'";
            Statement stmt=connect.createStatement();
            rs=stmt.executeQuery(sq);
            System.out.println("query executed");
            int count = 0;
            while(rs.next())
            {

```

```

count ++;
    }
if(count == 1)
{
new verify().setVisible(true);
}
else
{
JOptionPane.showMessageDialog(null, "INVALID PASSWORD");
}}

```



```

import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;
import java.sql.*;

public class verify extends javax.swing.JFrame {
    public verify() {
        initComponents();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try
        {
            //String s=jTextField1.getText();
            ResultSets;
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection connect=DriverManager.getConnection("jdbc:odbc:exam");
            String sq="select * from studentdetails ";
            Statement stmt=connect.createStatement();
            rs=stmt.executeQuery(sq);
            System.out.println("query executed");
            // System.out.println("check:"+rs.getInt("count")
            jTable1.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch(Exception ex){
            JOptionPane.showMessageDialog(null, "error occurs");}
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

```

```

new opening().setVisible(true);
    }
    public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new verify().setVisible(true);
    }
}); }

```

RESULT:

Thus the application for exam registration system using uml concept has been executed.

Viva Questions:

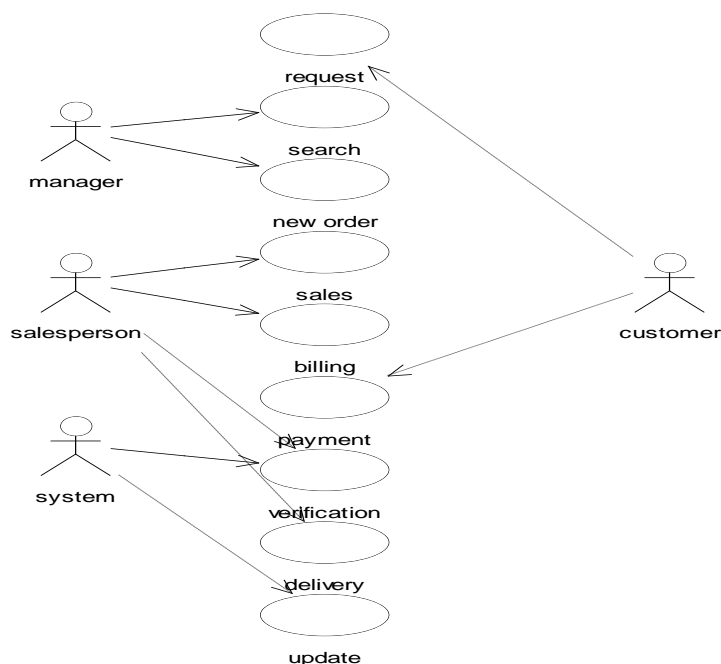
- 1.What is architecture?
2. What are messages?
- 3.What is component diagram?
4. What are the various components in sequence diagram?
- 5.What are the parts of the deployment diagram?

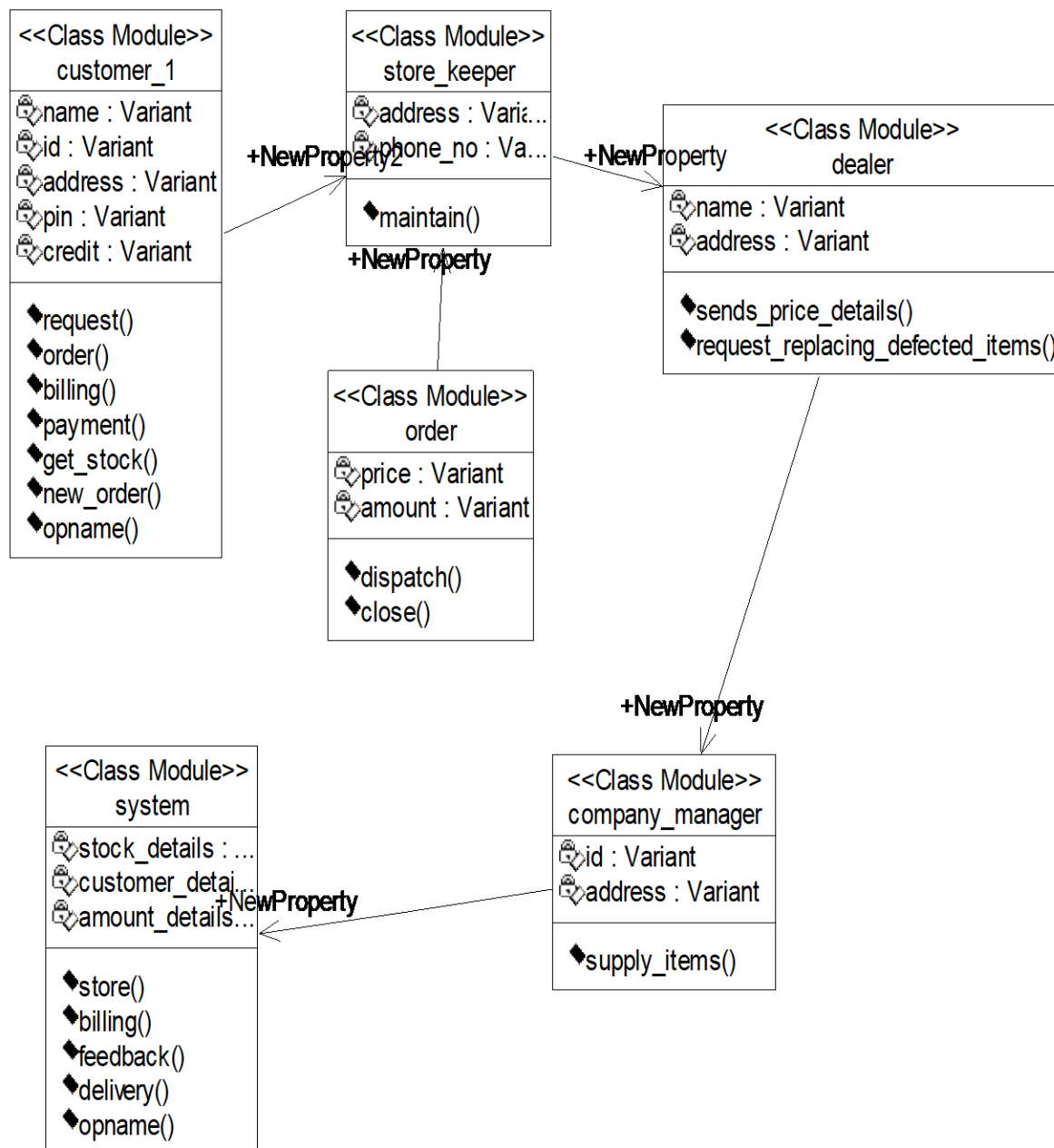
EX NO: 4 STOCK MAINTAINANCE SYSTEM**AIM:**

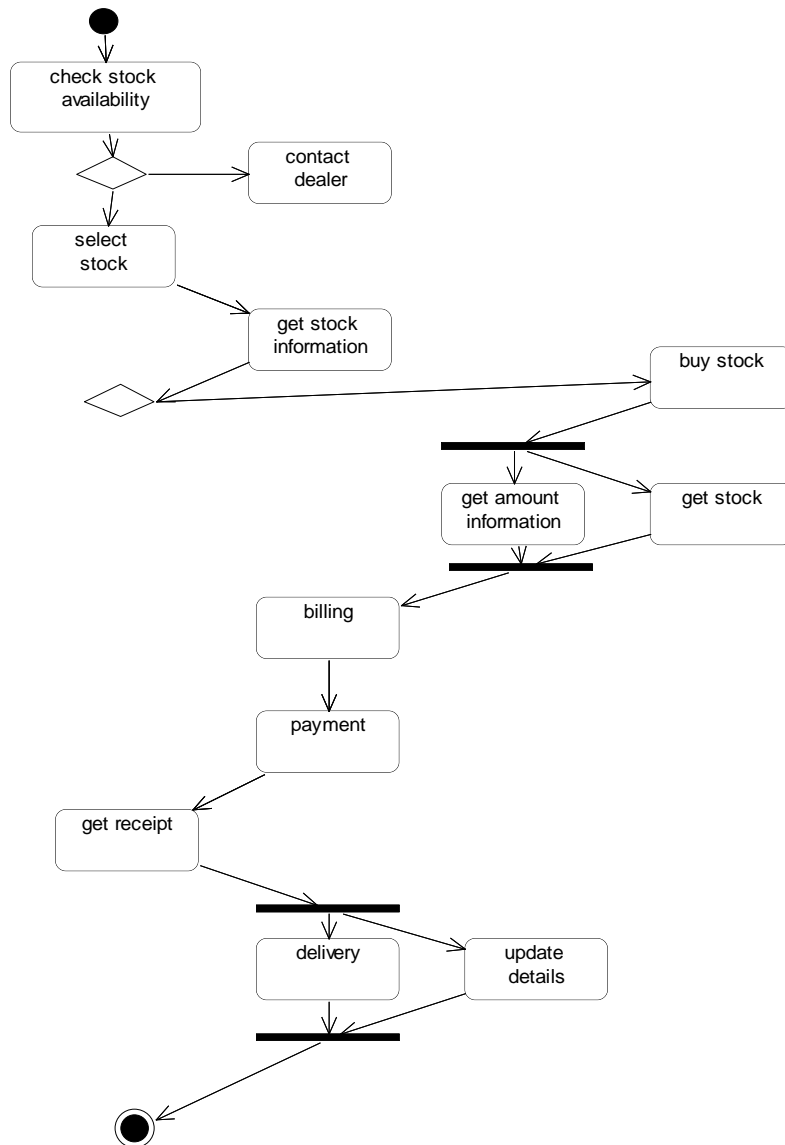
To create an application for Stock maintenance system using uml concept

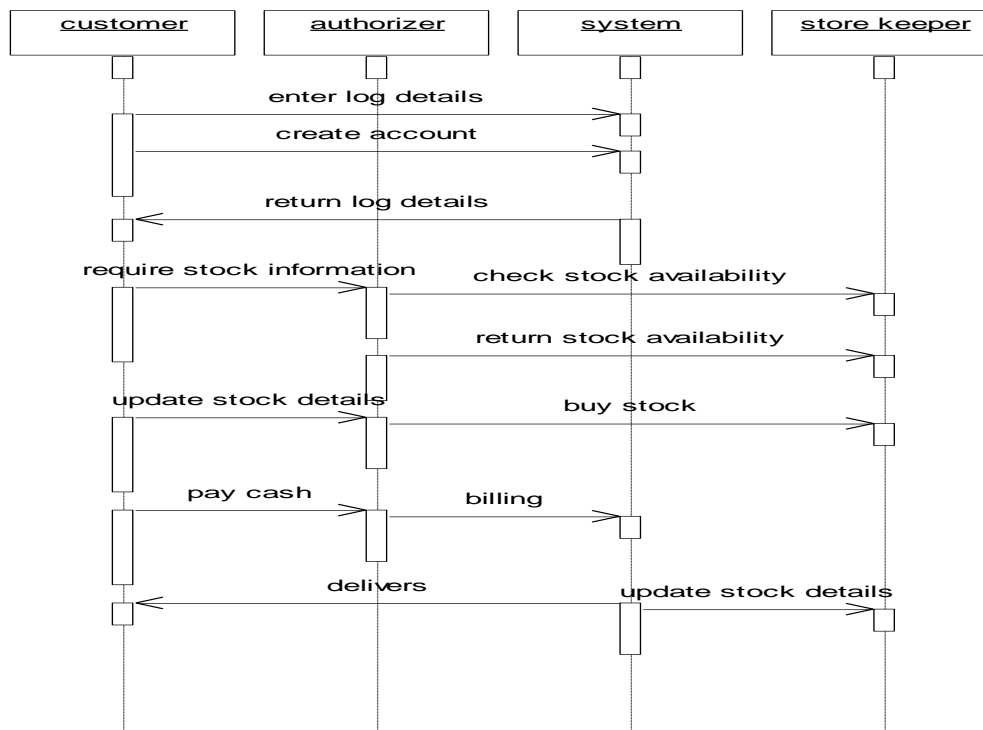
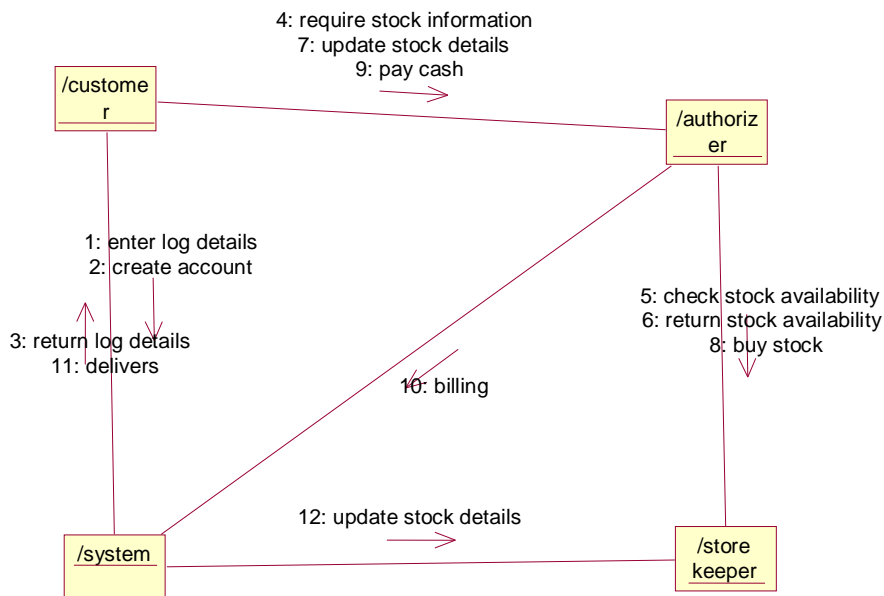
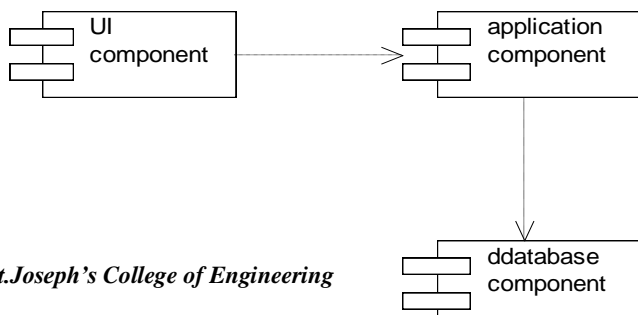
PROBLEM STATEMENT:

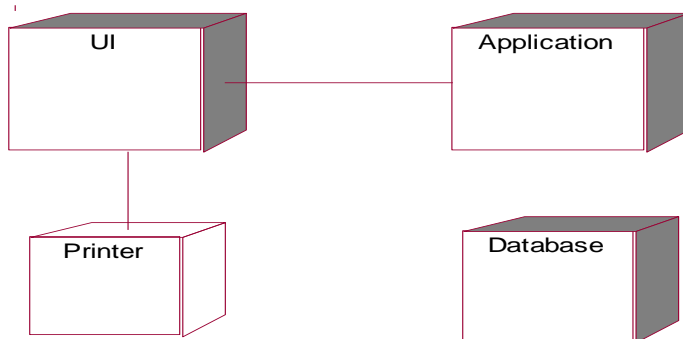
Stock maintenance system is the function of understanding the stock mix of a company and different demands on stock. The demands are influenced by both external and internal factor and are balanced by certain purchase and request to keep supplies at a reasonable or prescribed level. They are starting from request for new stock from stores to head office issues purchase order to vendor, the vendor ships the goods, ware house receives the goods. Ware house receives the goods, warehouse stocks and distributes to the stores. The whole economic status is being improved properly if stock is maintained.

USECASE DIAGRAM:

CLASS DIAGRAM:

ACTIVITY DIAGRAM:

SEQUENCE DIAGRAM:**COLLABORATION DIAGRAM:****COMPONENT DIAGRAM:**

DEPLOYMENT DAIGRAM:**CODING AND FORMS**

```

package Sample;
import java.sql.*;
import javax.swing.JOptionPane;
public class Login extends javax.swing.JFrame {
    public Connection con;
    public Statement st;
    public Login() {
        initComponents();
        try
        {
            Class.forName("com.mysql.jdbc.Driver")
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/stock", "root", "");
            st=con.createStatement();
            JOptionPane.showMessageDialog(null, "You are now connected...!!!");
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null, "Unable to connect");
        } }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try{
            String user = jTextField1.getText();
            String pass = String.valueOf(jPasswordField1.getText());
            String sqlquery1 = "Select * from login where username='"+user+"'and password='"+pass+"'";

```



```

ResultSet rs = st.executeQuery(sqlquery1);
if(rs.next())
{
    setVisible(false);
    new New().setVisible(true);
}
else{
    JOptionPane.showMessageDialog(null, "Wrong username or password");
}
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null, "Unable to login");
}
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setVisible(false);
    new Applicant().setVisible(true);
}

```

```

package Sample;
import java.sql.*;
import javax.swing.JOptionPane;
public class Applicant extends javax.swing.JFrame {
    public Connection con;
    public Statement st;
    /** Creates new form start */
    public Applicant() {
        initComponents();
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/stock", "root", "");
            st=con.createStatement();
            JOptionPane.showMessageDialog(null, "You are now connected...!!!");
        }
        catch(Exception e)

```

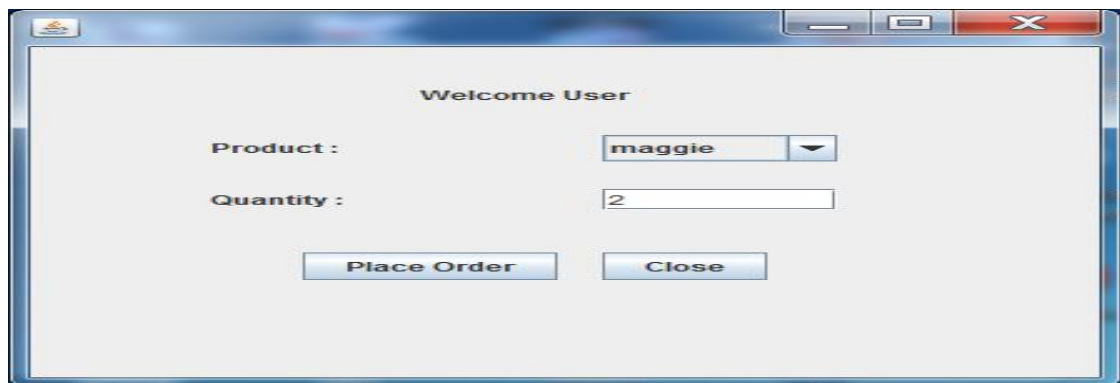
```

    {
        JOptionPane.showMessageDialog(null, "Unable to connect");
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String name = jTextField1.getText();
        String dob = jTextField2.getText();
        String addr = jTextArea1.getText();
        String email = jTextField5.getText();
        String mob = jTextField4.getText();
        String sqlquery1 = "Insert into register
values('"+name+"','"+dob+"','"+addr+"','"+email+"','"+mob+"')";
        st.executeUpdate(sqlquery1);
        JOptionPane.showMessageDialog(null, "Data successfully inserted");
        setVisible(false);
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, "Data insertion failure");
    }
    new Login().setVisible(true);
}

package Sample;

```



```

import java.sql.*;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;
import java.lang.String.*;
/**
 *
 * @author Keerthana
 */
public class Status extends javax.swing.JFrame {
    public Connection connect;
    public Statement st;
    String category;

```

```
/** Creates new form Status */
public Status(String c) {
    initComponents();
    category=c;
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        connect = DriverManager.getConnection("jdbc:mysql://localhost:3306/stock", "root", "");
        st=connect.createStatement();
        JOptionPane.showMessageDialog(null, "You are now connected...!!!");
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, "Unable to connect");
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String sqlquery1 = "Select * from details where category='"+category+"'";
        ResultSet rs = st.executeQuery(sqlquery1);
        jTable1.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, "Unable to display stock details");
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        int count,count2;
        String stock = jTextField3.getText();
        int qty = Integer.parseInt(jTextField1.getText());
        String sqlquery1 = "Select * from details where stock='"+stock+"'";
        ResultSet rs1 = st.executeQuery(sqlquery1);
        while(rs1.next()){
            if(qty<=(rs1.getInt(3))){
                count=rs1.getInt(3);
                count-=qty;
                String sqlquery2 = "Update details set qty='"+count+"' where stock='"+stock+"'";
                st.executeUpdate(sqlquery2);
                JOptionPane.showMessageDialog(null, "Stocks purchased");
                setVisible(false);
                new Thanks().setVisible(true);
            }
            else
                JOptionPane.showMessageDialog(null, "No stock available");
        }
    }
}
```

```
    }  
    }  
    catch(Exception e)  
    {  
        //JOptionPane.showMessageDialog(null, "Unable to check details");  
    }  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String reg = jTextField1.getText();  
    setVisible(false);  
    new Thanks().setVisible(true);  
}
```

RESULT:

Thus the application for Stock maintenance system using uml concept has been executed.

Viva Questions:

1. What are the different elements of a collaboration diagram?
2. What are the elements of activity diagram?
3. What are the different elements of use case?
4. What is the difference between activity and use case diagram?
5. What is component diagram?

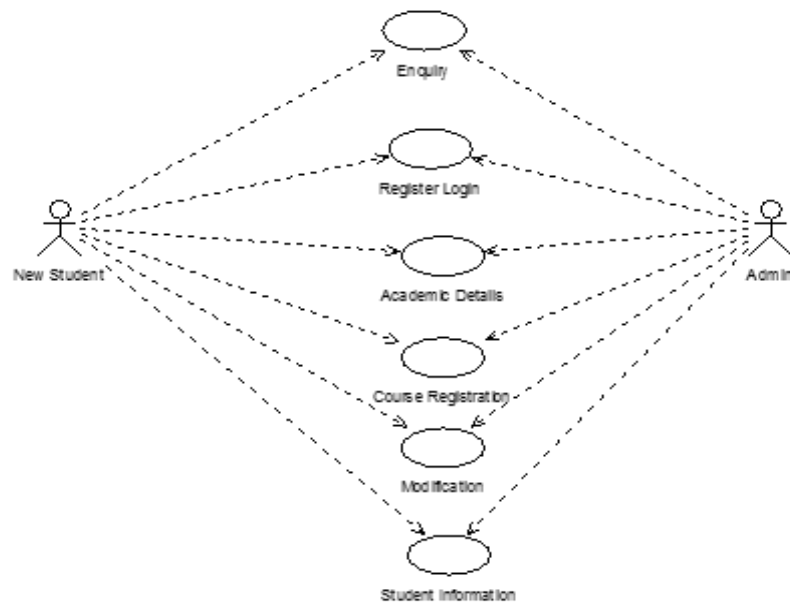
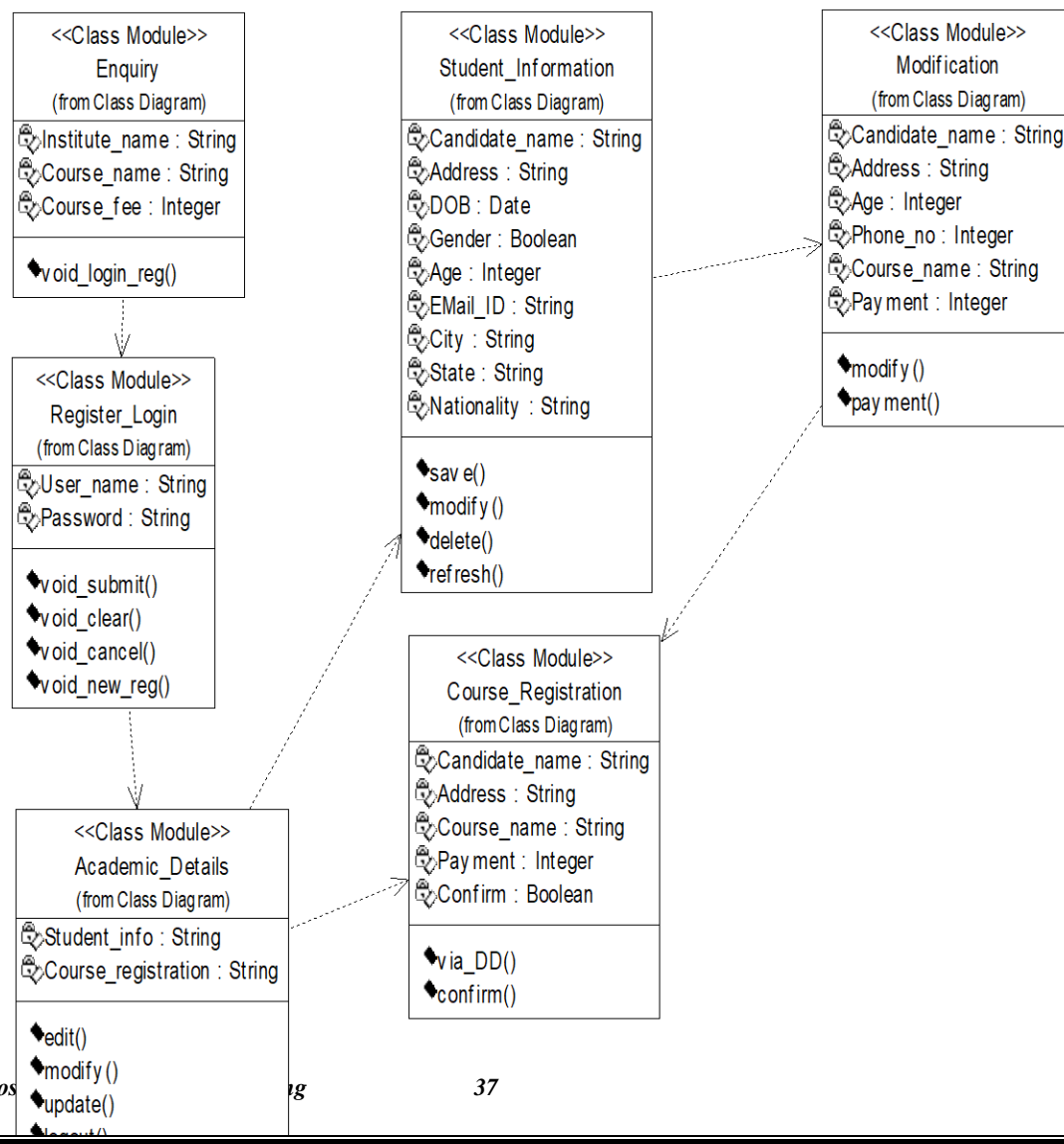
EX NO: 5 ONLINE COURSE RESERVATION SYSTEM**AIM:**

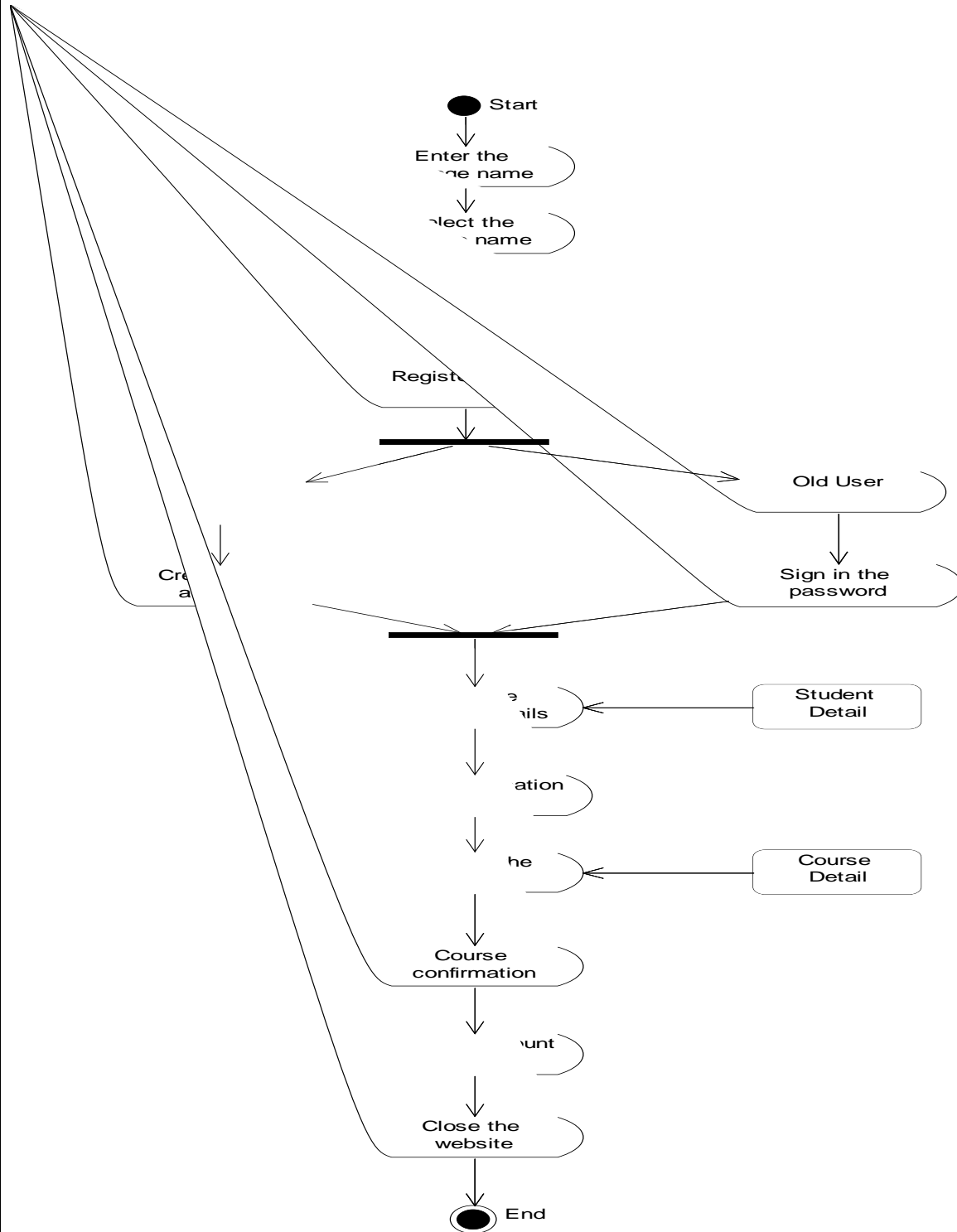
To create an application for online course reservation system using uml concept.

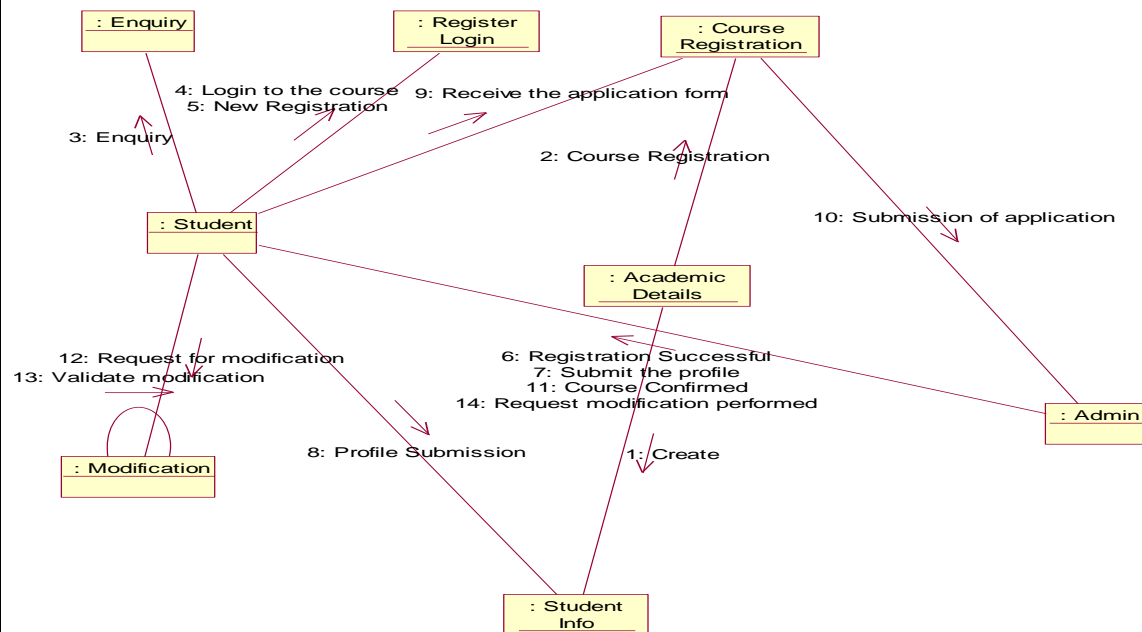
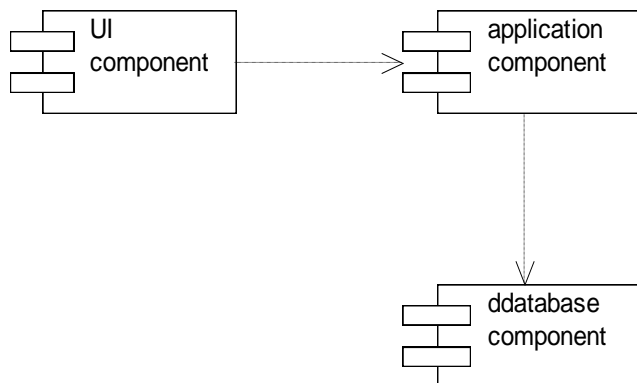
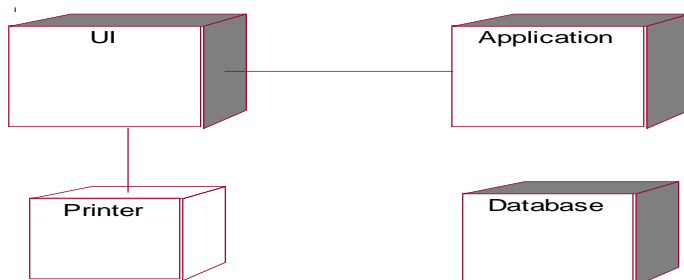
PROBLEM STATEMENT:

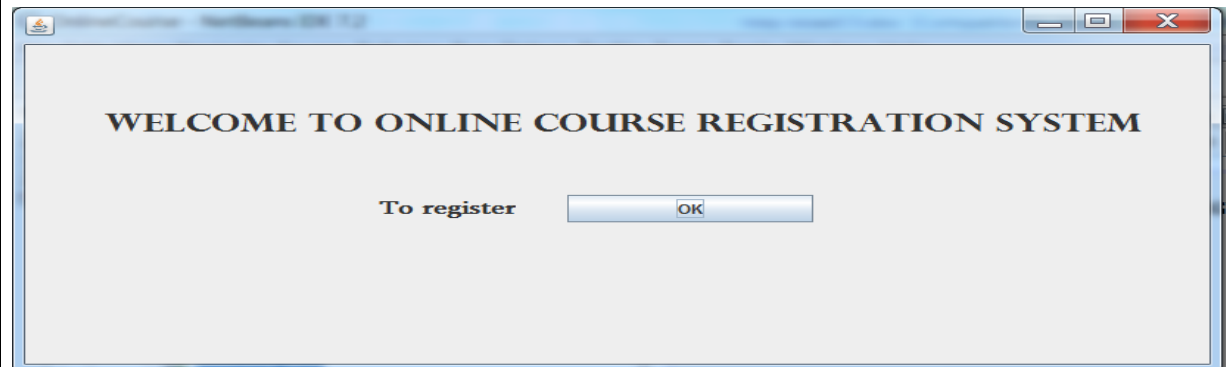
The software which displays the list of courses available for the mark that the student get and the student can able to allocate best course from the choice available. It displays and allocates courses based on student ranking. The student ranking is based on marks, caste and community. Based on caste and community the ranking may varied. This software allows the student to choose the best college for the available courses.

USE CASE DIAGRAM:

**CLASS DIAGRAM:**

ACTIVITY DIAGRAM:

COLLABORATION DIAGRAM:**COMPONENT DIAGRAM:****DEPLOYMENT DIAGRAM:**

CODING AND FORMS

```
package Sample;
public class Welcome extends javax.swing.JFrame {
    /** Creates new form Welcome */
    public Welcome() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        new Applicant().setVisible(true);
    }
}
```

```
package Sample;
import java.sql.*;
import javax.swing.JOptionPane;
public class Applicant extends javax.swing.JFrame {
    public Connection con;
    public Statement st;
    /** Creates new form start */
    public Applicant() {
        initComponents();
    }
}
```



```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    con = DriverManager.getConnection("jdbc:mysql://localhost:3306/course", "root", "");
    st=con.createStatement();
    JOptionPane.showMessageDialog(null, "You are now connected...!!!");
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null, "Unable to connect");
}
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String name = jTextField1.getText();
        String fname = jTextField2.getText();
        String dob = jTextField3.getText();
        String addr = jTextArea1.getText();
        String clgcode = jTextField7.getText();
        String drg = JComboBox3.getSelectedItem().toString();
        String yr = jTextField6.getText();
        String email = jTextField4.getText();
        String mob = jTextField5.getText();
        String course = JComboBox1.getSelectedItem().toString();
        String sch = JComboBox2.getSelectedItem().toString();
        String payment="Unpaid";
        String sqlquery1 = "Insert into register
values('"+name+"','"+fname+"','"+dob+"','"+addr+"','"+clgcode+"','"+drg+"','"+yr+"','"+email+"','"+mob+"',
','"+course+"','"+sch+"','"+payment+"')";
        st.executeUpdate(sqlquery1);
        JOptionPane.showMessageDialog(null, "Data successfully inserted");
        setVisible(false);
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, "Data insertion failure");
    }
    new Register().setVisible(true);
}

package Sample;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
public class Register extends javax.swing.JFrame {
    public Connection connect;
```

```
public Statement st;
/** Creates new form Register */
public Register() {
    initComponents();
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        connect = DriverManager.getConnection("jdbc:mysql://localhost:3306/course", "root", "");
        st=connect.createStatement();
        JOptionPane.showMessageDialog(null, "You are now connected...!!!");
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, "Unable to connect");
    }
}
```

RESULT:

Thus the application for online course reservation system using uml concept has been executed.

Viva Questions:

- 1.What is the difference between uses and extends?
- 2.How do we represent private, public and protected in class diagrams?
- 3.What does association in a class diagram mean?
- 4.What is uml use case diagram?
- 5.What does aggregation mean in a class diagram?

EXNO: 6 E-TICKETING SYSTEMS**AIM:**

To create an application for E-Ticketing systems using uml concept.

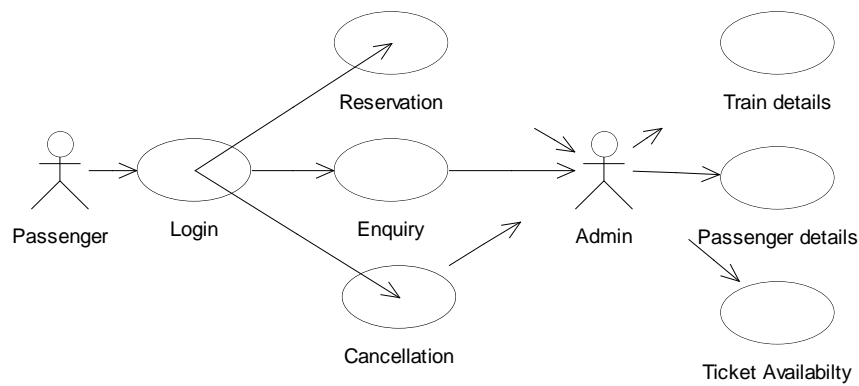
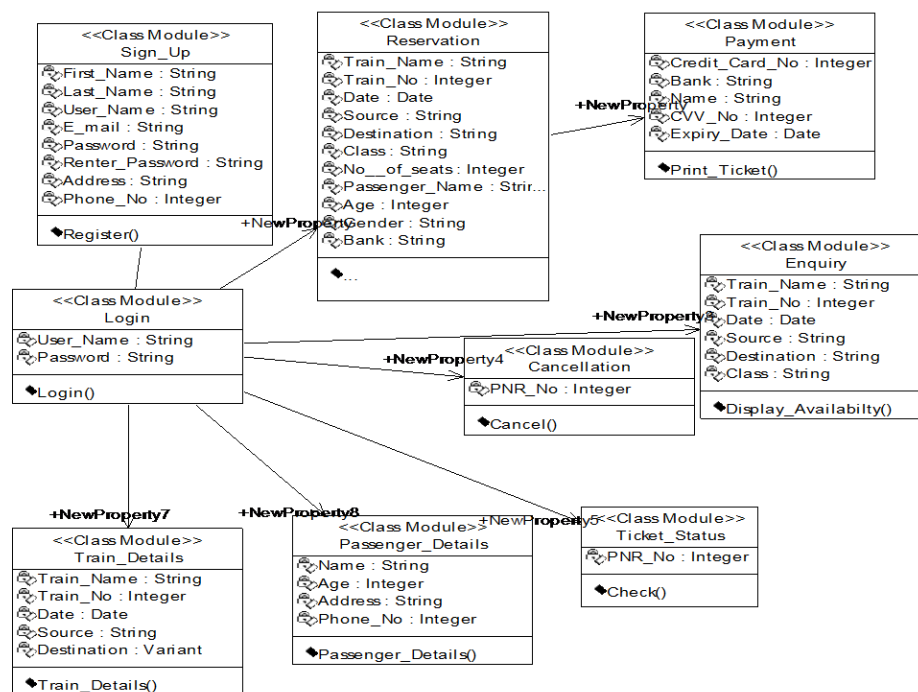
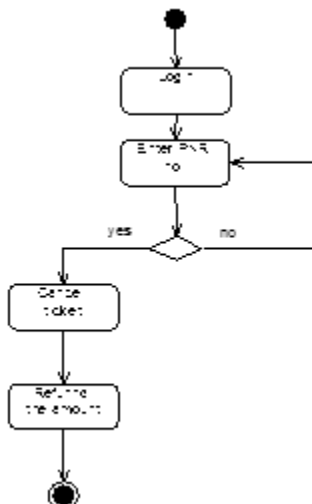
PROBLEM STATEMENT:

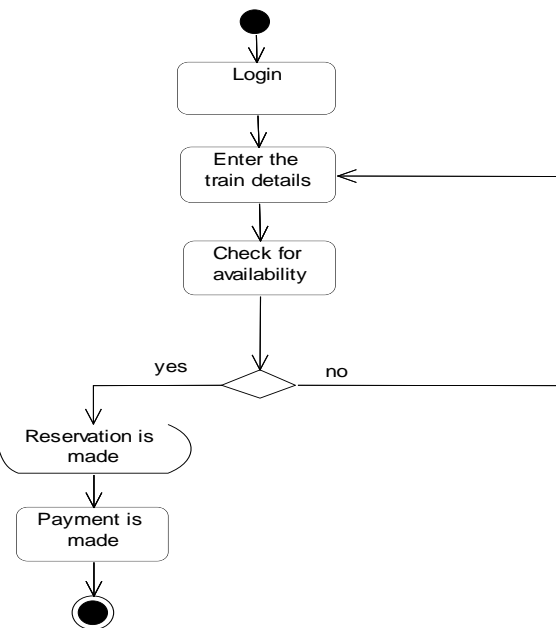
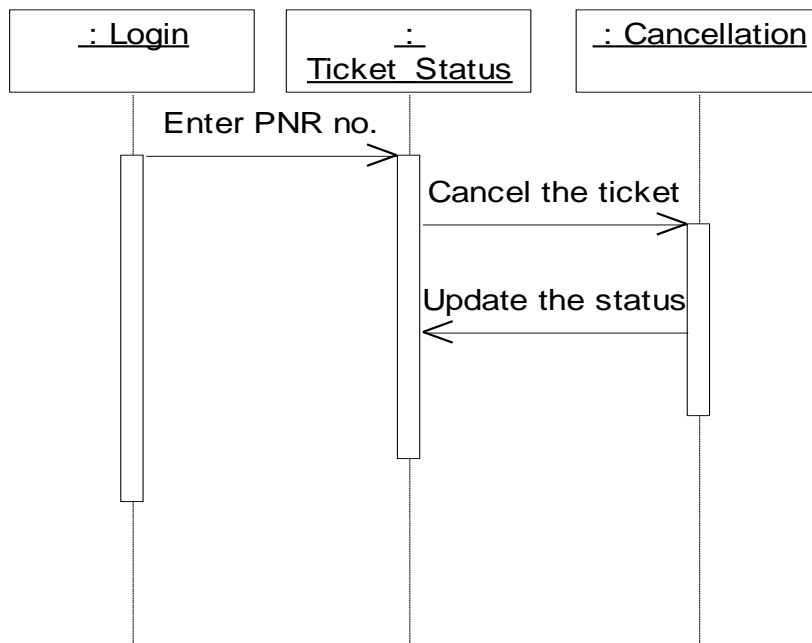
This project enables a user to perform the following operations.

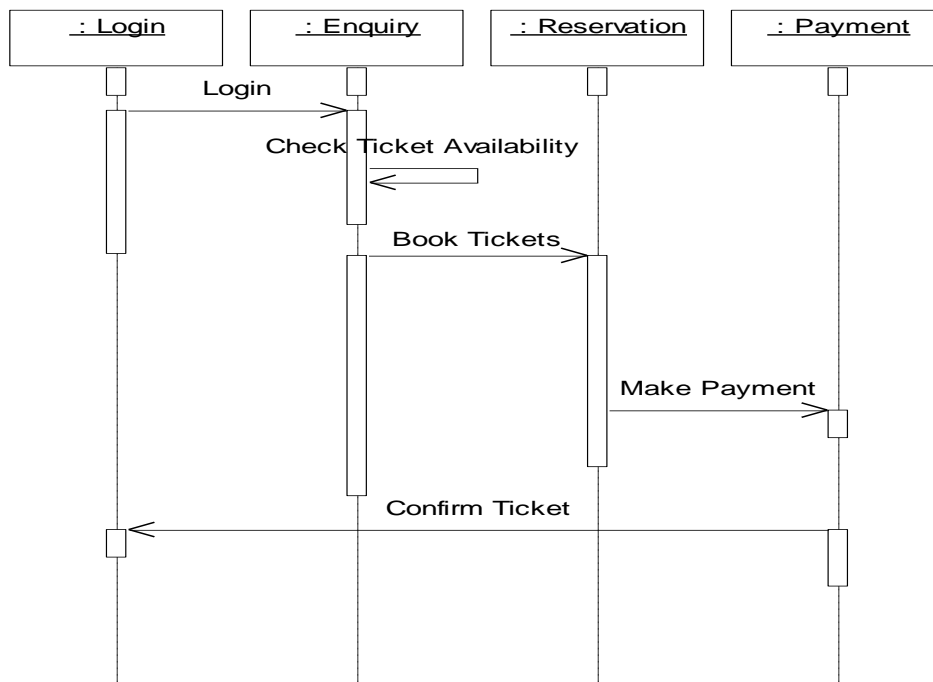
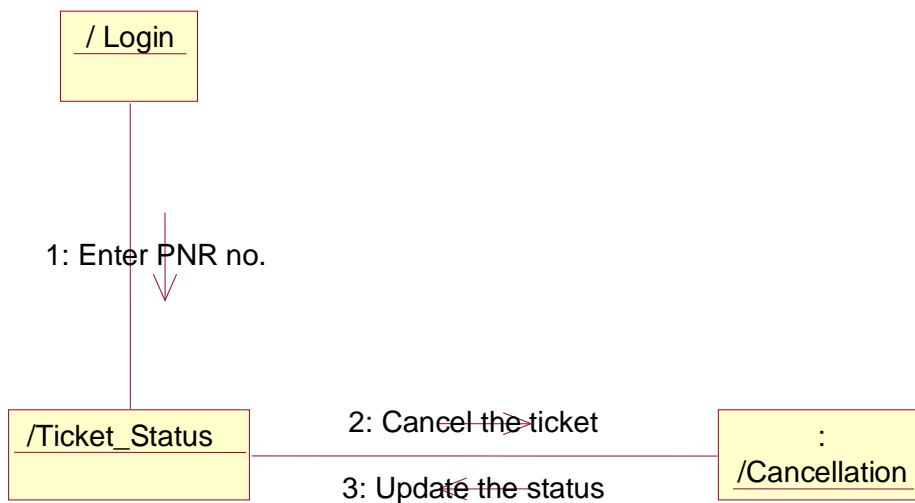
- Online Booking of tickets
- Online cancellation of tickets
- Online Enquiry system
- Online ticket status checking

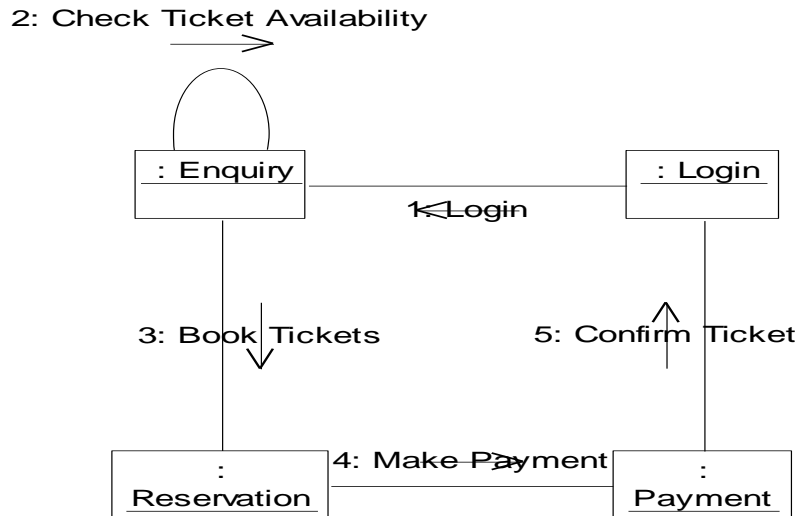
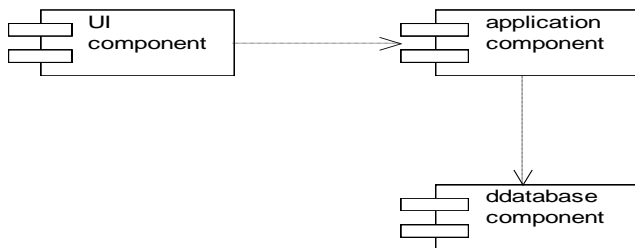
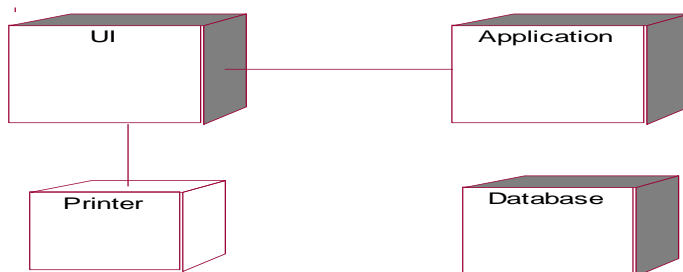
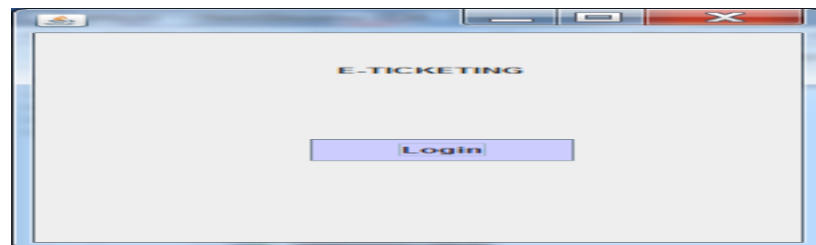
The booking service accepts the source;destination derails along with the train name and date of service. According to the availability of tickets, the particular passenger is allotted a seat on the particular train. The tickets would be displayed in a particular passenger is allotted a seat on the particular train. The ticket would redisplay in a printable format along with a unique seat number to differentiate between online ticket reservations. The cancellation services comes in very handy in case of emergency cancellation where in a passenger who has already booked his tickets can follow easy steps to cancel his ticket. The passenger specifies his seat no and other unique identification parameter like credit card number and the details are validated. If found legal, the ticket is cancelled and the transaction with the bank is nullified. The train schedule and enquiry features of this system enable the user to interactively identify a suitable configuration of train timings and fares. This module gives the user complete train details and facilities the passenger to choose a train of his choice.

USE CASE DIAGRAM:

**CLASS DIAGRAM:****ACTIVITY DIAGRAM: (CANCELLATION)****ACTIVITY DIAGRAM : (RESERVATION)**

**SEQUENCE DIAGRAM: (CANCELLATION)**

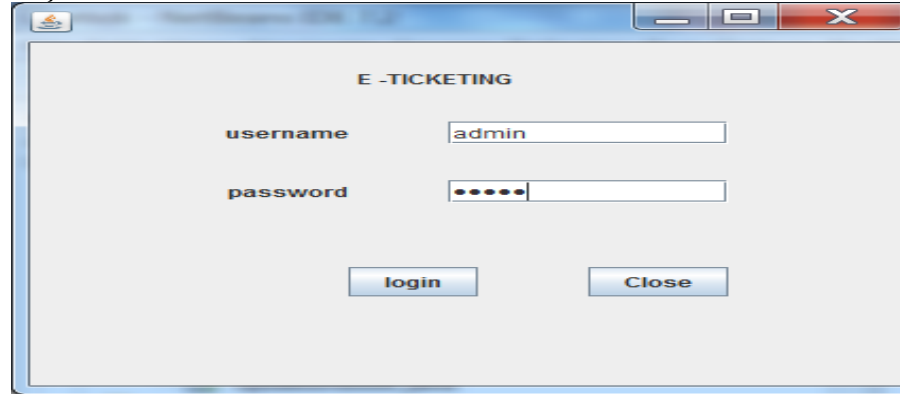
SEQUENCE DIAGRAM: (RESERVATION)**COLLABORATION DIAGRAM: (CANCELLATION)**

COLLABORATION DIAGRAM: (RESERVATION)**COMPONENT DIAGRAM:****DEPLOYMENT DIAGRAM:****FORMS AND CODING:**

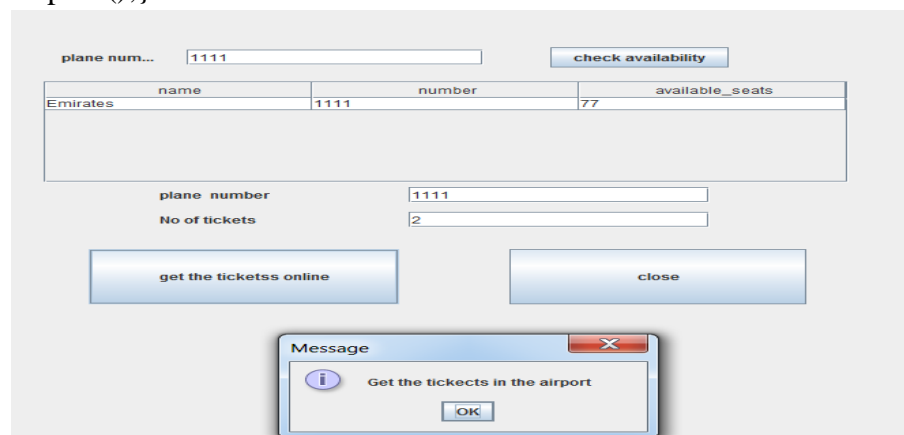
```

public class opening extends javax.swing.JFrame {
    /** Creates new form opening */
    public opening() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
  
```

```
// TODO add your handling code here:
new login().setVisible(true);
}
```



```
public class login extends javax.swing.JFrame {
/** Creates new form login */
public login() {
    initComponents();
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here
    String s1=jTextField1.getText();
    String s2=jPasswordField1.getText();
    if((s1.compareTo("admin")==0)&&(s2.compareTo("admin")==0))
    {
        new update().setVisible(true);
    }
    else{
        customer s=new customer(s1,s2);
    }
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    dispose();}
```



```
import java.sql.*;
import javax.swing.table.TableModel;
//import org.apache.commons.dbutils.DbUtils;
import net.proteanit.sql.DbUtils;
```

```

public class NewJFrame extends javax.swing.JFrame {
/** Creates new form NewJFrame */
public NewJFrame() {
initComponents();
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
try
{
String s=jTextField1.getText();
ResultSet rs;
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection connect=DriverManager.getConnection("jdbc:odbc:aero");
String sq="select * from planedetails where number = " +s+" ";
Statement stmt=connect.createStatement();
rs=stmt.executeQuery(sq);
System.out.println("query executed");
// System.out.println("check:"+rs.getInt("count"));
jTable1.setModel(DbUtils.resultSetToTableModel(rs));
}
catch(Exception e)
{ System.out.println(e.getMessage());
}
}
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
new opening().setVisible(true);
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
String s1=jTextField2.getText();
int j=Integer.parseInt(jTextField3.getText());
System.out.println("AD"+s1);
airdatabase l=new airdatabase(s1,j);
} //DECREASE IN TICKET COUNT

```

name	number	available_seats
Emirates	1111	75

RESULT:

Thus the application for E-Ticketing systems using uml concept has been executed.

Viva Questions:

- 1.What does composition mean in a class diagram?
2. How a uml use case diagram is different from traditional flow chart?
3. Who maintains the UML specification?
- 4.How the UML specification is updated?
- 5.On what basis UML modeling tool is selected?

EX:NO:7 SOFTWARE PERSONNEL MANAGEMENT SYSTEM**AIM:**

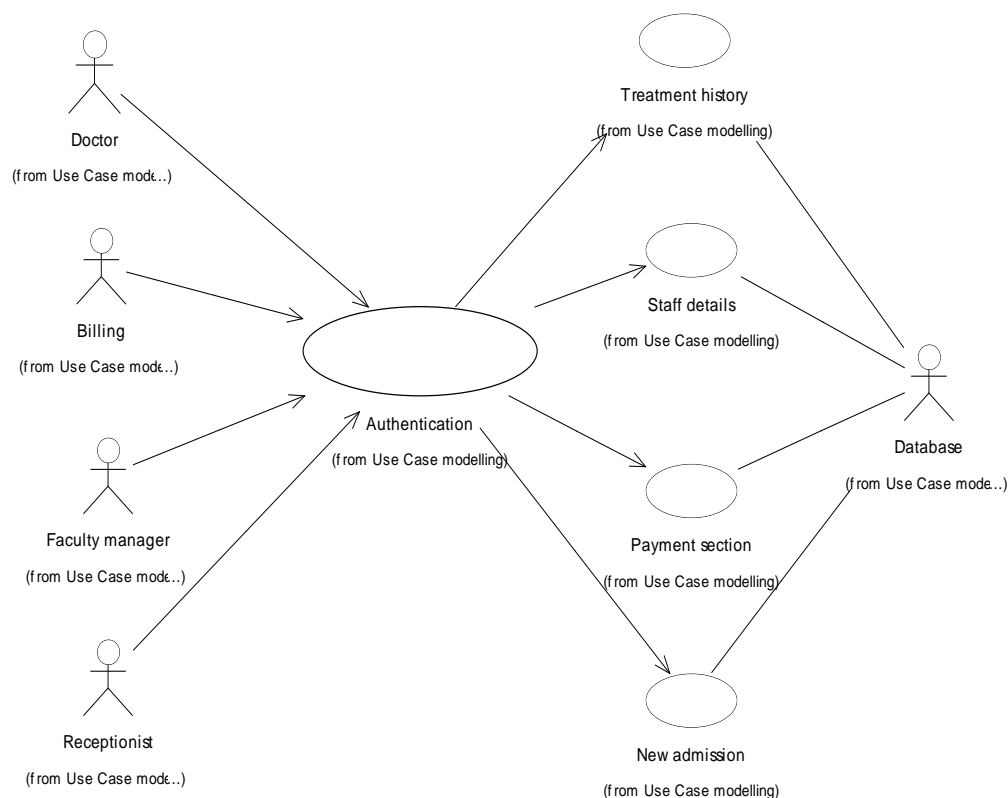
To create an application for Software personnel management system using uml concept

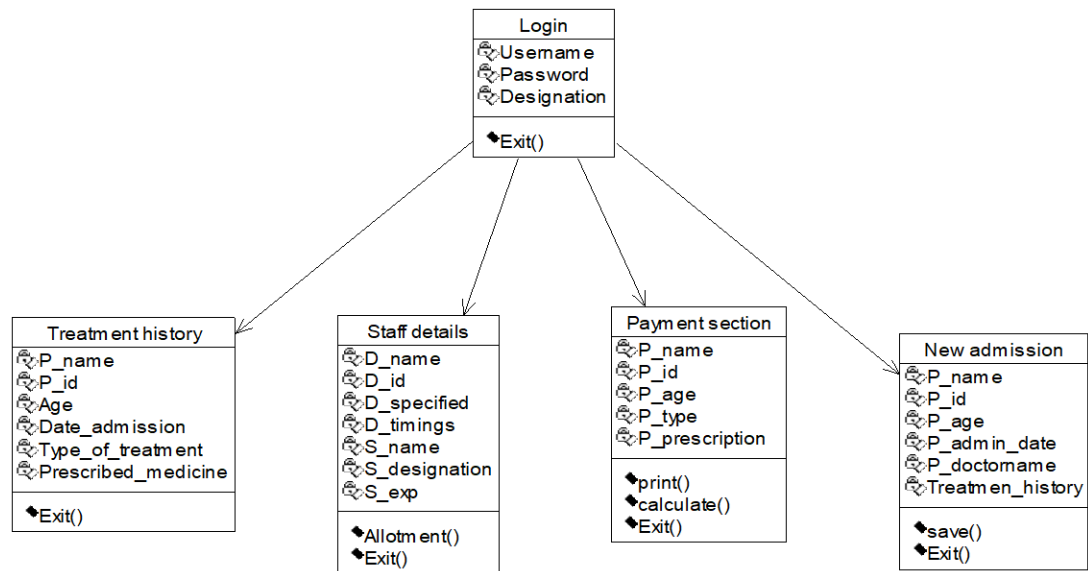
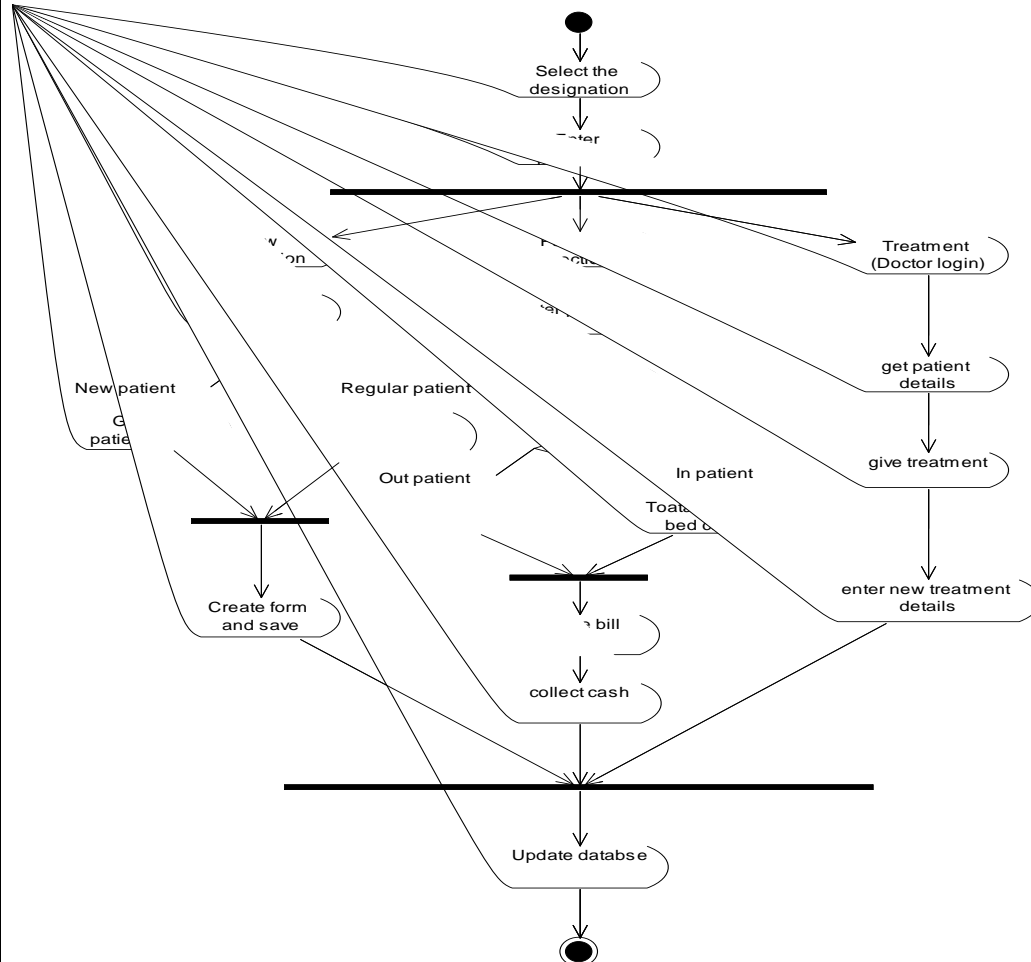
PROBLEM STATEMENT:

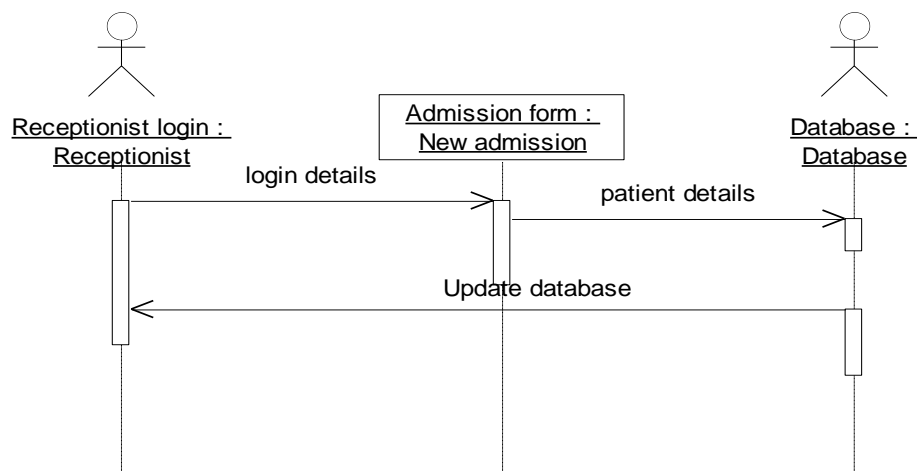
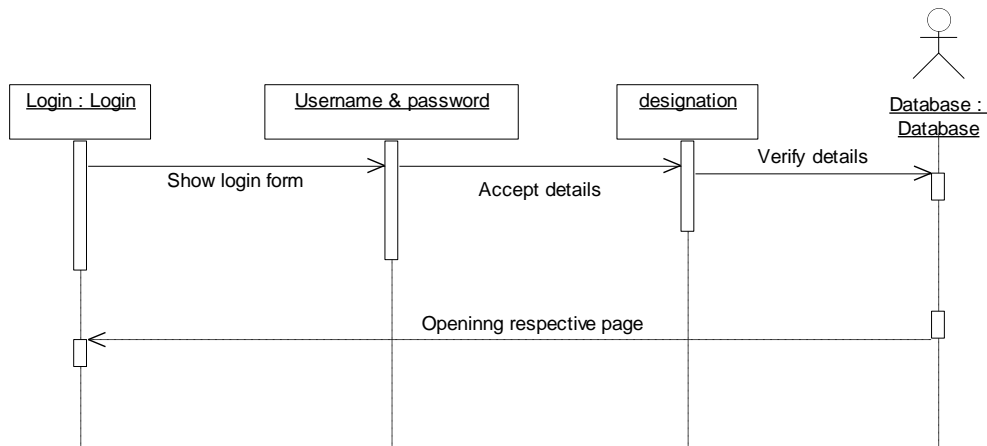
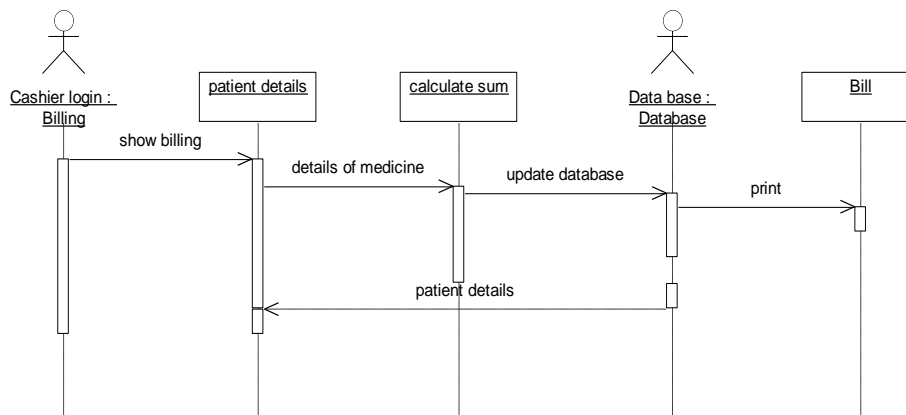
The module should provide users with a system to keep trace of hospitals,features,availability,staffing and report problems.

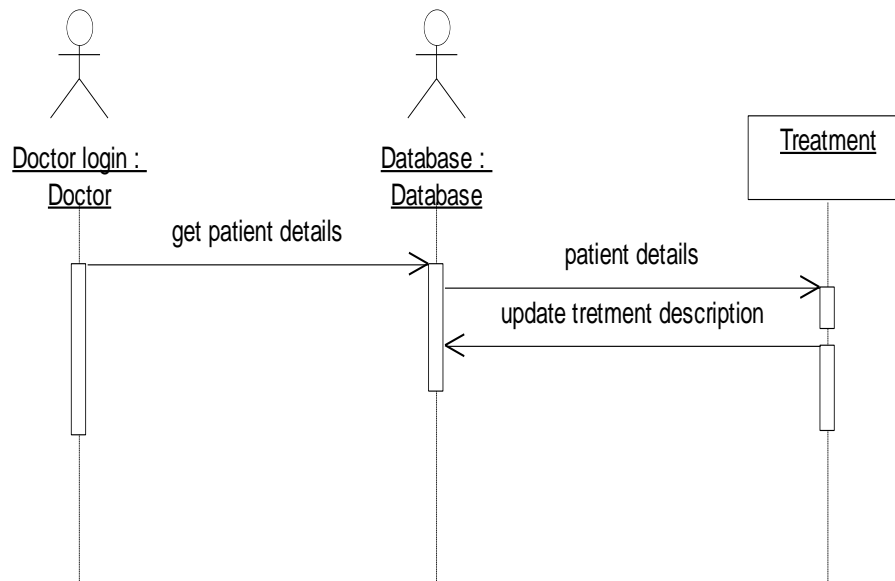
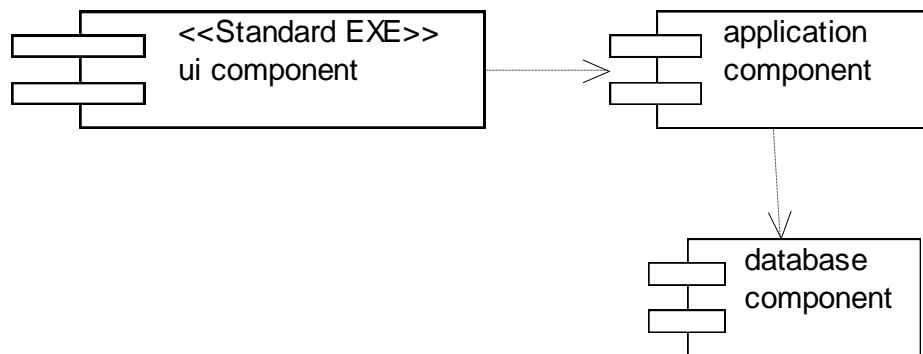
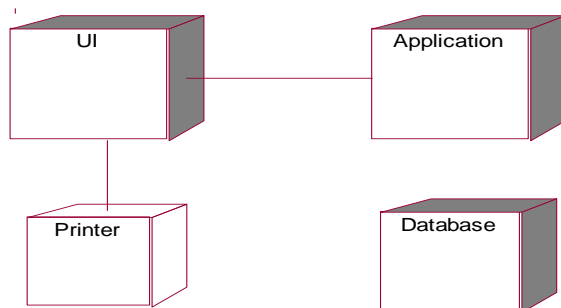
The hospital management system helps to make updated data base about the status of faculty members. The patient records have to be monitored more accurately and with frequent updation of the treatment details all these requirements have been tried to satisfied project.

It involved in the bus system are doctor,receptionist,cashier and they perform the actions such as giving treatment payment receiving entering the patient details to new admission all these actions will be executed.

USE CASE**DIAGRAM:**

CLASS DIAGRAM:**ACTIVITY DIAGRAM:**

SEQUENCE DIAGRAM:(LOGIN)**NEW ADMISSION:****PAYMENT SECTION:**

TREATMENT SECTION:**COMPONENT DIAGRAM:****DEPLOYMENT DIAGRAM:****FORMS AND CODING:**

```
package personnel;
```

```
/**
```

```
*
```

```
* @author Elcot
```



```

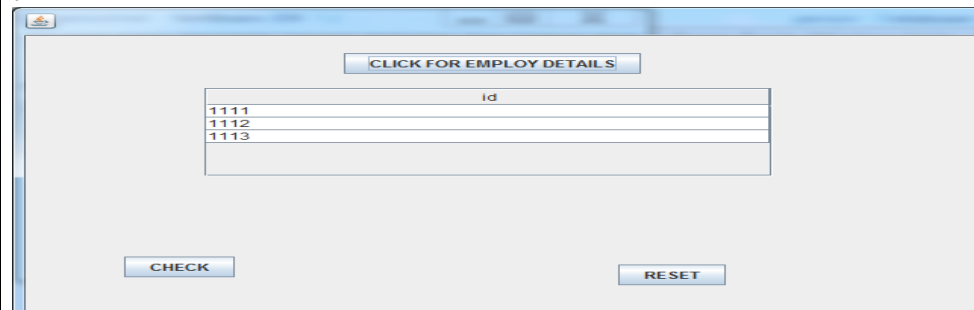
*/
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
public class login extends javax.swing.JFrame {
/** Creates new form login */
    public login() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String s1=jTextField1.getText();
        String s2=jPasswordField1.getText();
        System.out.println(s1);
        System.out.println(s2);
        try {
            ResultSet rs;
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection connect=DriverManager.getConnection("jdbc:odbc:pers");
            String s="select * from pass where username = '"+s1+"'and password = '"+s2+"'";
            Statement stmt=connect.createStatement();
            rs=stmt.executeQuery(s);
            int count=0;
            while(rs.next()) {
                count++;
            }
            if(count==1)
                new employ().setVisible(true);
            else
                JOptionPane.showMessageDialog(null,"Wrong Password");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

```

```

    setVisible(false);
    new login().setVisible(true);
}

```



```

package personnel;
/**
 *
 * @author Elcot
 */
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;
public class employ extends javax.swing.JFrame {
/** Creates new form employ */
    public employ() {
        initComponents();
    }
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        setVisible(false);
        new employ().setVisible(true);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        setVisible(false);
        new details().setVisible(true);
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        try
        {
            ResultSet rs;
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection connect=DriverManager.getConnection("jdbc:odbc:pers");
            String sq="select id from empo" ;
            Statement stmt=connect.createStatement();
            rs=stmt.executeQuery(sq);
            System.out.println("query executed");
            jTable1.setModel(DbUtils.resultSetToTableModel(rs));
        } catch (Exception e)
        {
            System.out.println(e.getMessage());
        }
    }
}

```

empname	id	address	salary	department	service
aaa	1111	chennai	500000	it	2

```

package personnel;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;
public class details extends javax.swing.JFrame {
    /** Creates new form details */
    public details() {
        initComponents();
    }
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            String s=jTextField1.getText();
            ResultSet rs;
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection connect=DriverManager.getConnection("jdbc:odbc:pers");
            String sq="select * from empo where id = '" +s+"'" ;
            Statement stmt=connect.createStatement();
            rs=stmt.executeQuery(sq);
            System.out.println("query executed");
            jTable1.setModel(DbUtils.resultSetToTableModel(rs));
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        new employ().setVisible(true);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        new logout().setVisible(true);
    }
}

```

RESULT:

Thus the application for Software personnel management system using uml concept has been executed.

Viva Questions:

1. Define Object Oriented Analysis.
2. What are the characteristics of an object?
3. Define class.
4. Name two types of object diagram.
5. What is an attribute? Give example.