

# **JAVA SWINGS BASED- STUDENT ASSIGNMENTS TRACKER**

**- SQL CONNECTIVITY USING JDBC**

*A*

*Report*

*Submitted in partial fulfillment of the  
Requirements for the award of the Degree  
of*

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**By**

M Amruth Sai <1602-19-737-066> Under the  
guidance of Ms B. Leelavathy



Department of Information Technology

Vasavi College of Engineering

(Autonomous) (Affiliated to Osmania

University) Ibrahimbagh, Hyderabad-31

## BONAFIDE CERTIFICATE

This is to certify that this project report titled '**STUDENT ASSIGNMENTS TRACKER**' is a project work of Mr. M Amruth Sai bearing roll no. 1602-19-737-066 who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature

Internal Examiner  
Examiner

Signature

External

**ABSTRACT:**

The main aim of Courier Management System is to deliver the parcel. We aim to demonstrate the use of create, read, update and delete SQL operations through this project. The project starts by adding details of courier branch and staffs working in their respective branches. The staff then adds sender and receiver details of that parcel i.e, Booking details. Then, the parcel is scheduled for delivery. Now, when the parcel is received by the receiver branch, they deliver the parcel and delivery status is shown

## REQUIREMENT ANALYSIS:

### List of Tables:

- Branch
- Delivery
- Staff
- Booking
- Schedule

### List of attributes with their domain types:

#### Branch

- Id INT(10)
- branchCode VARCHAR2(15)
- branchName VARCHAR2(25)
- branchAddress VARCHAR2(100)

#### Delivery

- Did INT(10)
- Bid INT(10)
- Selection VARCHAR2(50)

#### Staff

- Sid INT(5)
- Staff\_name VARCHAR2(50)
- Staff\_Contact VARCHAR2(20)

#### Booking

- Bid varchar2(30)
- ToName VARCHAR2(25)
- ToAddress VARCHAR2(100)
- Id Number(5)
- Weight INT(5)

## Student Assignments

### Schedule

- ShId INT(11)
- Bid INT(5)
- Sid INT(5)
- Id INT(5)

## AIM AND PRIORITY OF THE PROJECT

To create a Java GUI based student assignments tracker which takes the values like: user id, password for login and stores details of all students, faculty and student assignments their marks in those and staff can assign assignments to students. These values are to be updated in the database using JDBC connectivity.

## ARCHITECTURE AND TECHNOLOGY

**Software used:** Java Eclipse, Oracle 11g Database, Java SE version 13, SQL\*Plus.

### **Java SWINGS:**

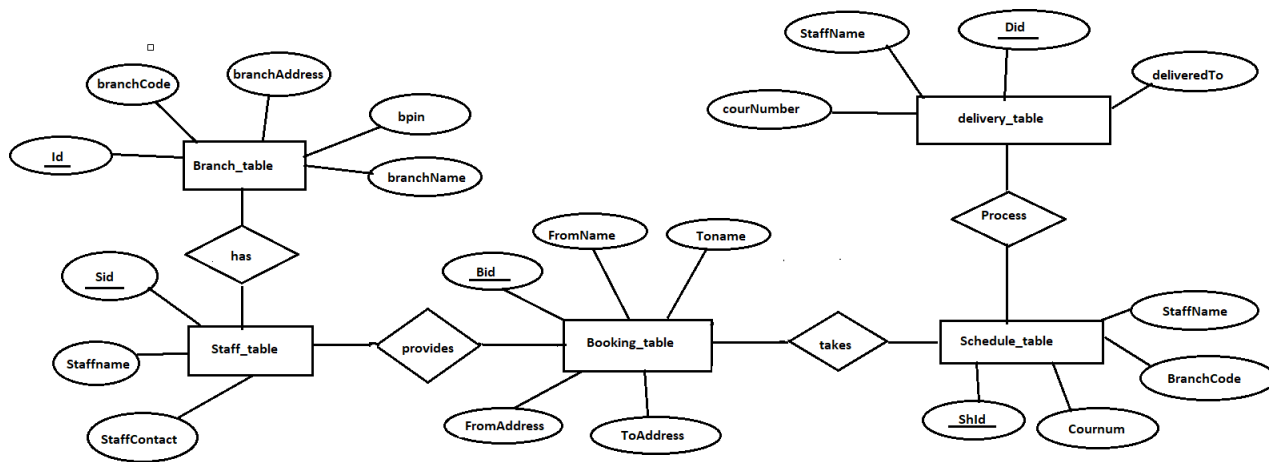
**Java SWINGS** is an API to develop GUI or window-based applications in java. Java SWING components are platform-independent. It is lightweight. The javax.swing package provides classes for SWING API such as JTextField, JLabel, JTextArea, JRadioButton, JCheckBox, JChoice, JList etc.

### **SQL:**

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySQL, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

## DESIGN:

### ER DIAGRAM:



### DDL COMMANDS:

```

SQL> desc Branch_table;
Name                                     Null?   Type
-----
ID                                     NOT NULL NUMBER(38)
BRANCHCODE                           VARCHAR2(15)
BRANCHNAME                           VARCHAR2(25)
BRANCHADDRESS                         VARCHAR2(100)
BPIN                                  VARCHAR2(10)

SQL> desc Booking_table;
Name                                     Null?   Type
-----
BID                                   NOT NULL VARCHAR2(10)
FROMNAME                             NOT NULL VARCHAR2(25)
TONAME                               NOT NULL VARCHAR2(25)
FROMADDRESS                           NOT NULL VARCHAR2(100)
TOADDRESS                             NOT NULL VARCHAR2(100)

SQL> desc Delivery_table;
Name                                     Null?   Type
-----
DID                                   NOT NULL NUMBER(38)
COURNUMBER                           VARCHAR2(15)
STAFFNAME                            VARCHAR2(50)
DELIVEREDTO                           VARCHAR2(50)
    
```

## Student Assignments

```
SQL> create table Branch_table(  
  2 Id INT,  
  3 branchCode varchar2(15),  
  4 branchName varchar2(25),  
  5 branchAddress varchar2(100),  
  6 bpin varchar2(10),  
  7 PRIMARY KEY(Id));
```

```
SQL> create table Delivery_table(  
  2 Did INT,  
  3 courNumber varchar2(15),  
  4 StaffName varchar2(50),  
  5 deliveredTo varchar2(50),  
  6 PRIMARY KEY(Did));
```

Table created.

```
SQL> create table Staff_table(  
  2 Sid INT,  
  3 Staffname varchar2(50),  
  4 StaffContact varchar2(10),  
  5 PRIMARY KEY(Sid));
```

Table created.

```
SQL> create table Schedule_table(  
  2 ShId INT,  
  3 Cournum varchar2(15),  
  4 BranchCode varchar2(15),  
  5 StaffName varchar2(15),  
  6 PRIMARY KEY(ShId));
```

Table created.

```
SQL> desc Branch_table;
```

Name	Null?	Type
ID	NOT NULL	NUMBER(38)
BRANCHCODE		VARCHAR2(15)
BRANCHNAME		VARCHAR2(25)
BRANCHADDRESS		VARCHAR2(100)
BPIN		VARCHAR2(10)

```
SQL> desc Schedule_table;
```

Name	Null?	Type
SHID	NOT NULL	NUMBER(38)
COURNUM		VARCHAR2(15)
BRANCHCODE		VARCHAR2(15)
STAFFNAME		VARCHAR2(15)

```
SQL> desc Staff_table;
```

Name	Null?	Type
SID	NOT NULL	NUMBER(38)
STAFFNAME		VARCHAR2(50)
STAFFCONTACT		VARCHAR2(10)

## Student Assignments

```
SQL> create table Booking_table(  
2 Bid varchar2(10),  
3 FromName varchar2(25) NOT NULL,  
4 ToName varchar2(25) NOT NULL,  
5 FromAddress varchar2(100) NOT NULL,  
6 ToAddress varchar2(100) NOT NULL,  
7 PRIMARY KEY(Bid));
```

Table created.

## DML COMMANDS:

```
SQL> select * from branch;
```

ID	BRANCH_NAME	BRANCH_CODE	BRANCH_ADDRESS
1	branch1	BR100	abc xyz
2	branch2	BR101	abc abc

```
SQL> insert into branch values(&Id,'&Branch_name','&Branch_code','&Branch_address');
```

Enter value for id: 3

Enter value for branch\_name: branch3

Enter value for branch\_code: BR102

Enter value for branch\_address: ghl xyz

old 1: insert into branch values(&Id,'&Branch\_name','&Branch\_code','&Branch\_address')

new 1: insert into branch values(3,'branch3','BR102','ghl xyz')

1 row created.

```
SQL> /
```

Enter value for id: 4

Enter value for branch\_name: branch4

Enter value for branch\_code: BR103

Enter value for branch\_address: jkl xyz

old 1: insert into branch values(&Id,'&Branch\_name','&Branch\_code','&Branch\_address')

new 1: insert into branch values(4,'branch4','BR103','jkl xyz')

1 row created.

```
SQL> /
```

Enter value for id: 5

Enter value for branch\_name: branch5

Enter value for branch\_code: BR104

Enter value for branch\_address: mno xyz

old 1: insert into branch values(&Id,'&Branch\_name','&Branch\_code','&Branch\_address')

new 1: insert into branch values(5,'branch5','BR104','mno xyz')

1 row created.



## Student Assignments

```
SQL> select * from staff;
```

SID	STAFF_NAME	STAFF_CONTACT
21	Sachin	9999999991
22	jack	9999999992
23	john	9999999998

```
SQL> desc staff;
```

Name	Null?	Type
SID	NOT NULL	NUMBER
STAFF_NAME	NOT NULL	VARCHAR2(20)
STAFF_CONTACT	NOT NULL	VARCHAR2(20)

```
SQL> insert into staff values(&SID, '&Staff_name', '&Staff_contact');
```

```
Enter value for sid: 24
```

```
Enter value for staff_name: warner
```

```
Enter value for staff_contact: 9999999993
```

```
old 1: insert into staff values(&SID, '&Staff_name', '&Staff_contact')
```

```
new 1: insert into staff values(24, 'warner', '9999999993')
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for sid: 25
```

```
Enter value for staff_name: smith
```

```
Enter value for staff_contact: 9999999994
```

```
old 1: insert into staff values(&SID, '&Staff_name', '&Staff_contact')
```

```
new 1: insert into staff values(25, 'smith', '9999999994')
```

```
1 row created.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL>
```

## Student Assignments

```
SQL> desc booking;
Name                                     Null?    Type
-----
BID                                     NOT NULL NUMBER
TONAME                                VARCHAR2(15)
TOADDRESS                             VARCHAR2(20)
WEIGHT                                NUMBER
ID                                     NOT NULL NUMBER

SQL> insert into booking values(&BID,'&toname','&toaddress',&weight,&Id);
Enter value for bid: 33
Enter value for toname: yuvi
Enter value for toaddress: xyz jkl
Enter value for weight: 20
Enter value for id: 3
old 1: insert into booking values(&BID,'&toname','&toaddress',&weight,&Id)
new 1: insert into booking values(33,'yuvi','xyz jkl',20,3)
insert into booking values(33,'yuvi','xyz jkl',20,3)
*
ERROR at line 1:
ORA-00001: unique constraint (AMRUTH.PK_BOOKINGID) violated

SQL> /
Enter value for bid: 34
Enter value for toname: yuvi
Enter value for toaddress: xyz jkl
Enter value for weight: 20
Enter value for id: 3
old 1: insert into booking values(&BID,'&toname','&toaddress',&weight,&Id)
new 1: insert into booking values(34,'yuvi','xyz jkl',20,3)

1 row created.

SQL> /
Enter value for bid: 35
Enter value for toname: chahal
Enter value for toaddress: xyz mno
Enter value for weight: 25
Enter value for id: 4
old 1: insert into booking values(&BID,'&toname','&toaddress',&weight,&Id)
new 1: insert into booking values(35,'chahal','xyz mno ',25,4)

1 row created.

SQL> commit;

Commit complete.
```

## Student Assignments

```
SQL> desc schedule;
```

Name	Null?	Type
SHID	NOT NULL	NUMBER
BID		NUMBER
SID		NUMBER
ID		NUMBER

```
SQL> select * from schedule;
```

SHID	BID	SID	ID
41	31	21	1
42	32	22	2
43	33	21	2

```
SQL> insert into schedule values(&shid,&bid,&sid,&id);
```

```
Enter value for shid: 44
```

```
Enter value for bid: 34
```

```
Enter value for sid: 24
```

```
Enter value for id: 3
```

```
old 1: insert into schedule values(&shid,&bid,&sid,&id)
```

```
new 1: insert into schedule values(44,34,24,3)
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for shid: 45
```

```
Enter value for bid: 35
```

```
Enter value for sid: 25
```

```
Enter value for id: 4
```

```
old 1: insert into schedule values(&shid,&bid,&sid,&id)
```

```
new 1: insert into schedule values(45,35,25,4)
```

```
1 row created.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL>
```

```

SQL> desc delivery;
Name                               Null?    Type
-----
DID                                NOT NULL NUMBER
BID                                NOT NULL NUMBER
SELECTION                          VARCHAR2(10)

SQL> select * from delivery;

      DID      BID SELECTION
-----
       51       31 YES
       54       31 NO
       52       31 YES
       53       33 No

SQL> insert into delivery values(&did,&bid,&selection');
Enter value for did: 55
Enter value for bid: 34
Enter value for selection: NO
old 1: insert into delivery values(&did,&bid,&selection')
new 1: insert into delivery values(55,34,'NO')

1 row created.

SQL> /
Enter value for did: 56
Enter value for bid: 35
Enter value for selection: YES
old 1: insert into delivery values(&did,&bid,&selection')
new 1: insert into delivery values(56,35,'YES')

1 row created.

SQL> commit;

Commit complete.

SQL>

```

## Student Assignments

```
SQL> select * from schedule;
```

SHID	BID	SID	ID
41	31	21	1
44	34	24	3
42	32	22	2
43	33	21	2
45	35	25	4

```
SQL> select * from delivery;
```

DID	BID SELECTION
51	31 YES
54	31 NO
52	31 YES
53	33 No
55	34 NO
56	35 YES

6 rows selected.

```
SQL> select * from branch;
```

ID	BRANCH_NAME	BRANCH_CODE
-----		
BRANCH_ADDRESS		
-----		
1	branch1	BR100
abc xyz		
3	branch3	BR!02
ghl xyz		
4	branch4	BR103
jkl xyz		
ID	BRANCH_NAME	BRANCH_CODE
-----		
BRANCH_ADDRESS		
-----		
2	branch2	BR101
abc abc		
5	branch5	BR104
mno xyz		

## Student Assignments

```
SQL> select * from staff;
```

SID	STAFF_NAME	STAFF_CONTACT
21	Sachin	9999999991
22	jack	9999999992
23	john	9999999998
24	warner	9999999993
25	smith	9999999994

```
SQL> select * from booking;
```

BID	TONAME	TOADDRESS	WEIGHT	ID
31	root	xyz abc	10	1
33	ramesh	ptr ptr	55	1
32	amruth	pqr pqr	54	2
34	yuvi	xyz jkl	20	3
35	chahal	xyz mno	25	4

```
SQL>
```

## IMPLEMENTATION

### Front end programs and its connectivity

**Java Database Connectivity (JDBC)** is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

```
public void connectToDB()
{
    try {
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","amruth","vasavi");
        statement=con.createStatement();
        statement.executeUpdate("commit");
    }
    catch (SQLException connectException)
    {
        System.out.println(connectException.getMessage());
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
        System.exit(1);
    }
}
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

### MAIN METHOD:

```
package dbms;

public class main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        new dbmsUI();
    }

}

package dbms;
import java.awt.*;
import javax.swing.*;
import java.awt.Color;

1602-19-737-066
M Amruth Sai
```

## Student Assignments

```
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class dbmsUI extends JFrame{
    /**
     *
     */
    private static final long serialVersionUID = 1L;

    private JMenuBar mnu;

    private JMenu mnuBranch;
    private JMenu mnuSchedule;
    private JMenu mnuBooking;
    private JMenu mnuStaff;
    private JMenu mnuDelivery;
    // private JMenu mnuorder_Det;
    // private JMenu mnuuser_orderDet;

    private JMenuItem insert1,update1,delete1,view1;
    private JMenuItem insert2,update2,delete2,view2;
    private JMenuItem insert3,update3,delete3,view3;
    private JMenuItem insert4,update4,delete4,view4;
    private JMenuItem insert5,update5,delete5,view5;
    // private JMenuItem insert6,update6,delete6,view6;
    // private JMenuItem insert7,update7,delete7,view7;

    private JLabel labelName;

    private static JPanel p0,p1;

    void initialize() {
        mnu=new JMenuBar();
        mnuBranch= new JMenu("Branch");
        mnuStaff= new JMenu("Staff");
        mnuSchedule= new JMenu("Schedule");
        mnuDelivery= new JMenu("Delivery");
        mnuBooking= new JMenu("Booking");
    // mnuuser_orderDet= new JMenu("User_orderDet");
        labelName=new JLabel("COURIER MANAGEMENT SYSTEM");
        p1=new JPanel();
        p0=new JPanel();
        insert1=new JMenuItem("Insert");
        update1=new JMenuItem("Update");
        delete1=new JMenuItem("Delete");
        view1=new JMenuItem("View");
        insert2=new JMenuItem("Insert");
        update2=new JMenuItem("Update");
        delete2=new JMenuItem("Delete");
        view2=new JMenuItem("View");
    }
}
```



## Student Assignments

```
        insert3=new JMenuItem("Insert");
        update3=new JMenuItem("Update");
        delete3=new JMenuItem("Delete");
        view3=new JMenuItem("View");
        insert4=new JMenuItem("Insert");
        update4=new JMenuItem("Update");
        delete4=new JMenuItem("Delete");
        view4=new JMenuItem("View");
        insert5=new JMenuItem("Insert");
        update5=new JMenuItem("Update");
        delete5=new JMenuItem("Delete");
        view5=new JMenuItem("View");
        insert6=new JMenuItem("Insert");
        update6=new JMenuItem("Update");
        delete6=new JMenuItem("Delete");
        view6=new JMenuItem("View");
        insert7=new JMenuItem("Insert");
        update7=new JMenuItem("Update");
        delete7=new JMenuItem("Delete");
        view7=new JMenuItem("View");
        insert1=new JMenuItem("Insert");
        update1=new JMenuItem("Update");
        delete1=new JMenuItem("Delete");
        view1=new JMenuItem("View"); */
    }
    void addComponentsToFrame() {
        mnuBranch.add(insert1);
        mnuBranch.add(delete1);
        mnuBranch.add(update1);
        mnuBranch.add(view1);
        mnuStaff.add(insert2);
        mnuStaff.add(delete2);
        mnuStaff.add(update2);
        mnuStaff.add(view2);
        mnuSchedule.add(insert3);
        mnuSchedule.add(delete3);
        mnuSchedule.add(update3);
        mnuSchedule.add(view3);
        mnuDelivery.add(insert4);
        mnuDelivery.add(delete4);
        mnuDelivery.add(update4);
        mnuDelivery.add(view4);
        mnuBooking.add(insert5);
        mnuBooking.add(delete5);
        mnuBooking.add(update5);
        mnuBooking.add(view5);
        mnuorder_Det.add(insert6);
        mnuorder_Det.add(delete6);
        mnuorder_Det.add(update6);
        mnuorder_Det.add(view6);
        mnuuser_orderDet.add(insert7);
        mnuuser_orderDet.add(delete7);
        mnuuser_orderDet.add(update7);
        mnuuser_orderDet.add(view7);
        mnu.add(mnuDelivery);
        mnu.add(mnuBooking);
        mnu.add(mnuStaff);
        mnu.add(mnuSchedule);
        mnu.add(mnuBranch);
        mnu.add(mnuorder_Det);
        mnu.add(mnuuser_orderDet);
        setJMenuBar(mnu);
        p1.add(labelName);p1.setAlignmentY(CENTER_ALIGNMENT);
        p0.add(p1);
        p0.setBounds(500,500,800,100);
    }
}
```

1602-19-737-066  
M Amruth Sai

## Student Assignments

```
        p0.setBackground(Color.lightGray);
        add(p0);
    }
    void closeWindow(){
        try {
            int a=JOptionPane.showConfirmDialog(this,"Are you sure want to Quit Courier Management System:");
            if(a==JOptionPane.YES_OPTION){
                JOptionPane.showMessageDialog(this,
                    "Thank you!\nExiting Courier Management System","Quit",
                    JOptionPane.WARNING_MESSAGE);
                System.exit(0);
            }
            else if (a== JOptionPane.NO_OPTION) {
                setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
            }
            else if (a== JOptionPane.CANCEL_OPTION) {
                setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
            }
        }
        catch(Exception e) {
            System.out.println(e);
        }
    }
    void register() {
        Branch log=new Branch(p0,dbmsUI.this,insert1,delete1,update1,view1);
        log.buildGUI();
        Staff user=new Staff(p0,dbmsUI.this,insert2,delete2,update2,view2);
        user.buildGUI();
        Booking umed=new Booking(p0,dbmsUI.this,insert5,update5,view5,delete5);
        umed.buildGUI();
        Delivery log_user=new Delivery(p0,dbmsUI.this,insert4,update4,view4,delete4);
        log_user.buildGUI();
        Schedule med=new Schedule(p0,dbmsUI.this,insert3,update3,view3,delete3);
        med.buildGUI(); /*
        Booking umed=new Booking(p0,dbmsUI.this,insert5,delete5,update5,view5);
        umed.buildGUI();
        orderDetails orders=new orderDetails(p0,EMedUI.this,insert6,delete6,update6,view6);
        orders.buildGUI();
        UserOrderDetails u_orders=new UserOrderDetails(p0,EMedUI.this,insert7,delete7,update7,view7);
        u_orders.buildGUI(); */
        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent we)
            {
                closeWindow();
            }
        });
    }
}
public dbmsUI() {
    initialize();
    addComponentsToFrame();
    register();
    pack();
    //
    setBackground(Color.RED);
    setTitle("Courier Management System");
    setSize(800,800);
    setVisible(true);
}
}
```

**BRANCH:**

; 1602-19-737-066

M Amruth Sai

## Student Assignments

```
package dbms;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public class Branch{
    /**
     *
     */
    // private static final long serialVersionUID = 1L;
    private JButton insertButton,deleteButton,viewButton,updateButton;
    private JPanel p1,p2,p3,p;
    private JLabel lblid,lblname,lblcode,lbladdress;
    private JTextField txtid,txtname,txtcode,txtaddress;

    private List BIDList;
    Connection con;ResultSet rs;
    Statement statement;
    private JFrame frame;
    private JMenuItem insert,delete,update,view;
    public Branch(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem
view)
    {

        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectToDB();

        this.frame=frame;
        this.insert=insert;
        this.delete=delete;
        this.update=update;
```

1602-19-737-066  
M Amruth Sai

## Student Assignments

```
this.view=view;

lblid=new JLabel("Branch ID");
lblname=new JLabel("Branch Name");
lblcode=new JLabel("Branch Code");
lbladdress=new JLabel("Branch Address");

txtid=new JTextField(15);
txtname=new JTextField(15);
txtcode=new JTextField(8);
txtaddress=new JTextField(15);

this.p=p;

}

public void connectToDB()
{
    try {

        Connection con=DriverManager.getConnection(
            "jdbc:oracle:thin:@localhost:1521:xe","amruth","vasavi");

        statement=con.createStatement();
        statement.executeUpdate("commit");

    }
    catch (SQLException connectException)
    {
        System.out.println(connectException.getMessage());
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
        System.exit(1);
    }
}

private void displaySQLErrors(SQLException e)
{
    JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState: " +
e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");

}

public void loadBIDs() {
    try {
        BIDList.removeAll();
        rs=statement.executeQuery("select Id from Branch");
        while(rs.next()) {
            BIDList.add(rs.getString("ID"));
        }
    }
    catch(SQLException e) {
        displaySQLErrors(e);
    }
}
```

1602-19-737-066  
M Amruth Sai

## Student Assignments

```
}
```

```
public void buildGUI() {
```

```
    insert.addActionListener(new ActionListener() {
```

```
        @Override
```

```
        public void actionPerformed(ActionEvent arg0) {
```

```
            // TODO Auto-generated method stub
```

```
            insertButton=new JButton("Insert");
```

```
            txtid.setText(null);
```

```
            txtname.setText(null);
```

```
            txtcode.setText(null);
```

```
            txtaddress.setText(null);
```

```
            p.removeAll();
```

```
            frame.invalidate();
```

```
            frame.validate();
```

```
            frame.repaint();
```

```
            p1=new JPanel();
```

```
            p1.setLayout(new GridLayout(4,2));
```

```
            p1.add(lblid);
```

```
            p1.add(txtid);
```

```
            p1.add(lblname);
```

```
            p1.add(txtname);
```

```
            p1.add(lblcode);
```

```
            p1.add(txtcode);
```

```
            p1.add(lbladdress);
```

```
            p1.add(txtaddress);
```

```
            p3=new JPanel(new FlowLayout());
```

```
            p3.add(insertButton);
```

```
            //p1.add(txtf1);
```

```
            p3.setBackground(Color.yellow);
```

```
            p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);
```

```
            p1.setBackground(Color.pink) ;
```

```
            p2 = new JPanel(new FlowLayout());
```

```
                BIDList=new List(10);
```

```
                loadBIDs();
```

```
                p2.add(BIDList);p2.setBackground(Color.cyan) ;
```

```
                p2.setBounds(450,150,350,180);
```

```
            p.add(p1);p.add(p3);
```

```
            p.add(p2);
```

## Student Assignments

```
p.setLayout(new BorderLayout());

frame.add(p);
frame.setSize(800,800);
frame.validate();

insertButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try {
            String query="INSERT INTO BRANCH
VALUES("+txtid.getText()+"','"+txtname.getText()+"','"+txtcode.getText()+"','"+txtaddress.getText()+"");

            int i=statement.executeUpdate(query);
            JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows
succesfully");loadBIDs();

        }
        catch(SQLException insertException){
            displaySQLErrors(insertException);
        }
    }

});

delete.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        deleteButton=new JButton("Delete");

        txtid.setText(null);
        txtname.setText(null);
        txtcode.setText(null);
        txtaddress.setText(null);

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        p1=new JPanel();

        p1.setLayout(new GridLayout(4,2));
        p1.add(lblid);
        p1.add(txtid);
        p1.add(lblname);
        p1.add(txtname);
```

## Student Assignments

```
p1.add(lblcode);
p1.add(txtcode);
p1.add(lbladdress);
p1.add(txtaddress);

p3=new JPanel(new FlowLayout());
p3.add(deleteButton);
//p1.add(txtf1);
p3.setBackground(Color.yellow);
p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);
p1.setBackground(Color.pink) ;

// p1.setBounds(100,100,500,300);

p2 = new JPanel(new FlowLayout());

    BIDList=new List(10);
    loadBIDs();
    p2.add(BIDList);p2.setBackground(Color.cyan) ;

    p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);
p. add(p2);
    BIDList.addItemListener(new ItemListener()
    {
        public void itemStateChanged(ItemEvent e)
        {
            try
            {
                rs=statement.executeQuery("select * from Branch");
                while (rs.next())
                {
                    if
                    {
                        break;
                    }
                    if (!rs.isAfterLast())
                    {
                        txtid.setText(rs.getString("Id"));

                        txtname.setText(rs.getString("branch_name"));

                        txtcode.setText(rs.getString("branch_code"));

                        txtaddress.setText(rs.getString("branch_address"));
                    }
                }
            }
            catch (SQLException selectException)
            {
                displaySQLErrors(selectException);
            }
        }
    });

p.setLayout(new BorderLayout());
```

## Student Assignments

```
frame.add(p);
frame.setSize(800,800);
frame.validate();

deleteButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try {

            int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
            if(a==JOptionPane.YES_OPTION){
                String query="DELETE FROM BRANCH WHERE
ID="+BIDList.getSelectedItem();

                int i=statement.executeUpdate(query);
                JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows
succesfully");loadBIDs();
            }

        }
        catch(SQLException deleteException){
            displaySQLErrors(deleteException);
        }
    }
});

update.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        JButton updateButton = new JButton("Update");
        txtid.setText(null);
        txtname.setText(null);
        txtcode.setText(null);
        txtaddress.setText(null);

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        p1=new JPanel();

        p1.setLayout(new GridLayout(4,2));
        p1.add(lblid);
        p1.add(txtid);
        p1.add(lblname);
        p1.add(txtname);
        p1.add(lblcode);
```



## Student Assignments

```
p1.add(txtcode);
p1.add(lbladdress);
p1.add(txtaddress);
p3=new JPanel(new FlowLayout());
p3.add(updateButton);
//p1.add(txtf1);
p3.setBackground(Color.yellow);
p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);
p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

        BIDList=new List(10);
        loadBIDs();
        p2.add(BIDList);p2.setBackground(Color.cyan) ;

        p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);
p. add(p2);
        BIDList.addItemListener(new ItemListener()
        {
            public void itemStateChanged(ItemEvent e)
            {
                try
                {
                    rs=statement.executeQuery("select * from Branch");
                    while (rs.next())
                    {
                        if
                        {
                            break;
                        }
                        if (!rs.isAfterLast())
                        {
                            txtid.setText(rs.getString("ID"));

                            txtname.setText(rs.getString("Branch_name"));

                            txtcode.setText(rs.getString("Branch_code"));

                            txtaddress.setText(rs.getString("Branch_address"));

                        }
                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        });

p.setLayout(new BorderLayout());

frame.add(p);
frame.setSize(800,800);
frame.validate();
```

## Student Assignments

```
updateButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try {

            int a=JOptionPane.showConfirmDialog(p,"Are you
sure want to update:");

            if(a==JOptionPane.YES_OPTION){
                String query="update Branch set
Branch_name='"+txtname.getText()+"',Branch_code='"+txtcode.getText()+"',Branch_address='"+txtaddress.getText()+"
WHERE ID="+BIDList.getSelectedItem();

                int i=statement.executeUpdate(query);
                JOptionPane.showMessageDialog(p,"\nupdated
"+i+" rows succesfully");loadBIDs();

            }

        } catch(SQLException deleteException){
            displaySQLErrors(deleteException);
        }

    }

});

view.addActionListener(new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        Label view1=new Label("View");
        //view1.setAlignment(Label.CENTER);
        Font myFont = new Font("Serif",Font.BOLD,25);
        view1.setFont((myFont));
        viewButton=new JButton("View");
        p1=new JPanel();
        p2=new JPanel();
        p1.add(view1);
        p2.add(viewButton);p1.setBackground(Color.cyan) ;p2.setBackground(Color.cyan) ;
        p.add(p1);p.add(p2); p.setLayout(new FlowLayout());
        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
        viewButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
```

## Student Assignments

```
JFrame f;  
  
JTable j;  
  
f = new JFrame();  
  
f.setTitle("Branch details");  
  
DefaultTableModel model = new DefaultTableModel();  
j = new JTable(model);  
model.addColumn("Branch Id");  
model.addColumn("Branch Name");  
model.addColumn("Branch Code");  
model.addColumn("Branch Address");  
  
try {  
  
        rs=statement.executeQuery("select * from  
Branch");  
        while(rs.next()) {  
            model.addRow(new  
Object[]{rs.getString("Id"), rs.getString("Branch_name"),rs.getString("Branch_code"),rs.getString("Branch_address")});  
        }  
        catch(SQLException viewException) {  
            displaySQLErrors(viewException);  
        }  
        j.setEnabled(false);  
j.setBounds(30, 40, 300, 300);  
  
JScrollPane sp = new JScrollPane(j);  
f.add(sp);  
  
f.setSize(800, 400);  
  
f.setVisible(true);  
  
}  
  
});  
  
}
```

```
}
```

## STAFF:

```
package dbms;
```

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;
```

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;
```

```
public class Staff{
    /**
     *
     */
    // private static final long serialVersionUID = 1L;
    private JButton insertButton,deleteButton,viewButton,updateButton;
    private JPanel p1,p2,p3,p;
    private JLabel lblid,lblname,lblcontact;
    private JTextField txtid,txtname,txtcontact;

    private List SIDList;
    Connection con;ResultSet rs;
    Statement statement;
    private JFrame frame;
    private JMenuItem insert,delete,update,view;
    public Staff(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem view)
    {

        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
    }
}
```

1602-19-737-066  
M Amruth Sai

## Student Assignments

```
    }
    connectToDB();

    this.frame=frame;
    this.insert=insert;
    this.delete=delete;
    this.update=update;
    this.view=view;

    lblid=new JLabel("Staff ID");
    lblname=new JLabel("Staff Name");
    lblcontact=new JLabel("Staff Contact");

    txtid=new JTextField(15);
    txtname=new JTextField(15);
    txtcontact=new JTextField(8);
//    txtaddress=new JTextField(15);

    this.p=p;

}

public void connectToDB()
{
    try {

        Connection con=DriverManager.getConnection(
            "jdbc:oracle:thin:@localhost:1521:xe","amruth","vasavi");

        statement=con.createStatement();
        statement.executeUpdate("commit");

    }
    catch (SQLException connectException)
    {
        System.out.println(connectException.getMessage());
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
        System.exit(1);
    }
}

private void displaySQLErrors(SQLException e)
{
    JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState: " +
e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");

}

public void loadSIDs() {
    try {
        SIDList.removeAll();
        rs=statement.executeQuery("select sid from Staff");
        while(rs.next()) {
            1602-19-737-066
            M Amruth Sai
```

## Student Assignments

```

        SIDList.add(rs.getString("sid"));
    }
}
catch(SQLException e) {
    displaySQLErrors(e);
}
}

public void buildGUI() {

    insert.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent arg0) {
            // TODO Auto-generated method stub
            insertButton=new JButton("Insert");
            txtid.setText(null);
            txtname.setText(null);
            txtcontact.setText(null);
            txtaddress.setText(null);

            p.removeAll();
            frame.invalidate();
            frame.validate();
            frame.repaint();

            p1=new JPanel();

            p1.setLayout(new GridLayout(4,2));
            p1.add(lblid);
            p1.add(txtid);
            p1.add(lblname);
            p1.add(txtname);
            p1.add(lblcontact);
            p1.add(txtcontact);
            p1.add(lbladdress);
            p1.add(txtaddress);

            p3=new JPanel(new FlowLayout());
            p3.add(insertButton);
            //p1.add(txtf1);
            p3.setBackground(Color.yellow);
            p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);
            p1.setBackground(Color.pink) ;

            p2 = new JPanel(new FlowLayout());

            SIDList=new List(10);
            loadSIDs();
            p2.add(SIDList);p2.setBackground(Color.cyan) ;
        }
    });
}
```

## Student Assignments

```
p2.setBounds(450,150,350,180);

p.add(p1);p.add(p3);
p.add(p2);

p.setLayout(new BorderLayout());

frame.add(p);
frame.setSize(800,800);
frame.validate();

insertButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try {
            String query="INSERT INTO STAFF
VALUES("+txtid.getText()+","+txtname.getText()+","+txtcontact.getText()+")";

            int i=statement.executeUpdate(query);
            JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows
succesfully");loadSIDs();

        }
        catch(SQLException insertException){
            displaySQLErrors(insertException);
        }
    }

});

}
});
/*
delete.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        deleteButton=new JButton("Delete");

        txtid.setText(null);
        txtname.setText(null);
        txtcode.setText(null);
        txtaddress.setText(null);

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
```

## Student Assignments

```
p1.setLayout(new GridLayout(4,2));
p1.add(lblid);
p1.add(txtid);
p1.add(lblname);
p1.add(txtname);
p1.add(lblcode);
p1.add(txtcode);
p1.add(lbladdress);
p1.add(txtaddress);

p3=new JPanel(new FlowLayout());
p3.add(deleteButton);
//p1.add(txtf1);
p3.setBackground(Color.yellow);
p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);
p1.setBackground(Color.pink) ;

// p1.setBounds(100,100,500,300);

p2 = new JPanel(new FlowLayout());

    BIDList=new List(10);
    loadBIDs();
    p2.add(BIDList);p2.setBackground(Color.cyan) ;

    p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);
p. add(p2);
    BIDList.addItemListener(new ItemListener()
    {
        public void itemStateChanged(ItemEvent e)
        {
            try
            {
                rs=statement.executeQuery("select * from Branch");
                while (rs.next())
                {
                    if
                    break;
                }
                if (!rs.isAfterLast())
                {
                    txtid.setText(rs.getString("Id"));

                    txtname.setText(rs.getString("branch_name"));

                    txtcode.setText(rs.getString("branch_code"));

                    txtaddress.setText(rs.getString("branch_address"));

                }
            }
            catch (SQLException selectException)
            {
                displaySQLErrors(selectException);
            }
        }
    });
(rs.getString("Id").equals(BIDList.getSelectedItem()))
```

1602-19-737-066  
M Amruth Sai



## Student Assignments

```
        }
    }
});

p.setLayout(new BorderLayout());

frame.add(p);
frame.setSize(800,800);
frame.validate();

deleteButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
    try {

        int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
        if(a==JOptionPane.YES_OPTION){
            String query="DELETE FROM BRANCH WHERE
ID="+BIDList.getSelectedItem();

            int i=statement.executeUpdate(query);
            JOptionPane.showMessageDialog(p,"nDeleted "+i+" rows
succesfully");loadBIDs();

        }

        catch(SQLException deleteException){
            displaySQLErrors(deleteException);
        }

    }

});

}
});
*/

delete.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        deleteButton=new JButton("Delete");

        txtid.setText(null);
        txtname.setText(null);
        txtcontact.setText(null);
        txtaddress.setText(null);

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
    }
});

//

p1=new JPanel();
```

## Student Assignments

```
p1.setLayout(new GridLayout(4,2));
p1.add(lblid);
p1.add(txtid);
p1.add(lblname);
p1.add(txtname);
p1.add(lblcontact);
p1.add(txtcontact);
p1.add(lbladdress);
p1.add(txtaddress);

//
//

p3=new JPanel(new FlowLayout());
p3.add(deleteButton);
//p1.add(txtf1);
p3.setBackground(Color.yellow);
p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);
p1.setBackground(Color.pink) ;

// p1.setBounds(100,100,500,300);

p2 = new JPanel(new FlowLayout());

    SIDList=new List(10);
    loadSIDs();
    p2.add(SIDList);p2.setBackground(Color.cyan) ;

    p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);
p. add(p2);
    SIDList.addItemListener(new ItemListener()
    {
        public void itemStateChanged(ItemEvent e)
        {
            try
            {
                rs=statement.executeQuery("select * from Staff");
                while (rs.next())
                {
                    if
                    break;
                }
                if (!rs.isAfterLast())
                {
                    txtid.setText(rs.getString("sid"));
                    txtname.setText(rs.getString("staff_name"));

                    txtcontact.setText(rs.getString("staff_contact"));
                    txtaddress.setText(rs.getString("branch_address"));
                }
            }
            catch (SQLException selectException)
            {
                displaySQLErrors(selectException);
            }
        }
    })
(rs.getString("sid").equals(SIDList.getSelectedItem()))

//

1602-19-737-066
M Amruth Sai
```

## Student Assignments

```
        }
    });

    p.setLayout(new BorderLayout());

    frame.add(p);
    frame.setSize(800,800);
    frame.validate();

    deleteButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // TODO Auto-generated method stub
            try {

                int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
                if(a==JOptionPane.YES_OPTION){
                    String query="DELETE FROM STAFF WHERE
SID="+SIDList.getSelectedItemId();

                    int i=statement.executeUpdate(query);
                    JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows
succesfully");loadSIDs();
                }

            }
            catch(SQLException deleteException){
                displaySQLErrors(deleteException);
            }
        }
    });

    update.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent arg0) {
            // TODO Auto-generated method stub
            JButton updateButton = new JButton("Update");
            txtid.setText(null);
            txtname.setText(null);
            txtcontact.setText(null);
            txtaddress.setText(null);

            p.removeAll();
            frame.invalidate();
            frame.validate();
            frame.repaint();

            p1=new JPanel();

            p1.setLayout(new GridLayout(4,2));
```

## Student Assignments

```
p1.add(lblid);
p1.add(txtid);
p1.add(lblname);
p1.add(txtname);
p1.add(lblcontact);
p1.add(txtcontact);
//
//
p1.add(lbladdress);
p1.add(txtaddress);
p3=new JPanel(new FlowLayout());
p3.add(updateButton);
//p1.add(txtf1);
p3.setBackground(Color.yellow);
p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);
p1.setBackground(Color.pink) ;

p2 = new JPanel(new FlowLayout());

SIDList=new List(10);
loadSIDs();
p2.add(SIDList);p2.setBackground(Color.cyan) ;

p2.setBounds(450,150,350,180);

p. add(p1);p.add(p3);
p. add(p2);
SIDList.addItemListener(new ItemListener()
{
    public void itemStateChanged(ItemEvent e)
    {
        try
        {
            rs=statement.executeQuery("select * from staff");
            while (rs.next())
            {
                if
                (rs.getString("sid").equals(SIDList.getSelectedItem()))
                {
                    break;
                }
                if (!rs.isAfterLast())
                {
                    txtid.setText(rs.getString("sid"));
                    txtname.setText(rs.getString("staff_name"));
                    txtcontact.setText(rs.getString("staff_contact"));
                    //
                    txtaddress.setText(rs.getString("Branch_address"));
                }
            }
        }
        catch (SQLException selectException)
        {
            displaySQLErrors(selectException);
        }
    }
});

p.setLayout(new BorderLayout());

frame.add(p);
```

1602-19-737-066  
M Amruth Sai

## Student Assignments

```
frame.setSize(800,800);
frame.validate();

updateButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try {

            int a=JOptionPane.showConfirmDialog(p,"Are you
sure want to update:");

            if(a==JOptionPane.YES_OPTION){
                String query="update Staff set
staff_name='"+txtname.getText()+"',staff_contact='"+txtcontact.getText()+"' WHERE SID='"+SIDList.getSelectedItem();

                int i=statement.executeUpdate(query);
                JOptionPane.showMessageDialog(p,"\nupdated
"+i+" rows succesfully");loadSIDs();

            }

        } catch(SQLException deleteException){
            displaySQLErrors(deleteException);
        }

    }

});

}

});

view.addActionListener(new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        Label view1=new Label("View");
        //view1.setAlignment(Label.CENTER);
        Font myFont = new Font("Serif",Font.BOLD,25);
        view1.setFont((myFont));
        viewButton=new JButton("View");
        p1=new JPanel();
        p2=new JPanel();
        p1.add(view1);
        p2.add(viewButton);p1.setBackground(Color.cyan) ;p2.setBackground(Color.cyan) ;
        p.add(p1);p.add(p2); p.setLayout(new FlowLayout());
        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
        viewButton.addActionListener(new ActionListener() {
```

1602-19-737-066  
M Amruth Sai

## Student Assignments

```
@Override
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    JFrame f;

    JTable j;

    f = new JFrame();

    f.setTitle("Staff details");

    DefaultTableModel model = new DefaultTableModel();
    j = new JTable(model);
    model.addColumn("sid");
    model.addColumn("staff_name");
    model.addColumn("staff_contact");
    model.addColumn("Branch Address");

    //

    try {

        rs=statement.executeQuery("select * from
Staff");

        while(rs.next()) {
            model.addRow(new
Object[]{rs.getString("sid"), rs.getString("staff_name"),rs.getString("staff_contact")});
        }
    } catch(SQLException viewException) {
        displaySQLErrors(viewException);
    }
    j.setEnabled(false);
    j.setBounds(30, 40, 300, 300);

    JScrollPane sp = new JScrollPane(j);
    f.add(sp);

    f.setSize(800, 400);

    f.setVisible(true);

}

});

}
```

```
}
```

```
}
```

## BOOKING:

```
package dbms;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.Properties;
```

```
public class Booking
{
```

```
    private JPanel pn,pn1,pn2,pn3;
    private JFrame jframe;
    private JButton JB_insert,JB_modify,JB_view,JB_delete;
    private JLabel JL_Booking_id,JL_Toname,JL_Weight,JL_Toaddress,JL_Branch_id;
    private JTextField JTF_Booking_id,JTF_Toname,JTF_Weight,JTF_Toaddress,JTF_Branch_id;
    Connection con;
    ResultSet rs;
    Statement stmt;
    private JMenuItem insert2,update2,view2,delete2;
    private List BookingList;
    private Choice Branchid;//d_1
```

```
    public Booking(JPanel pn,JFrame jframe,JMenuItem insert2,JMenuItem update2,JMenuItem
view2,JMenuItem delete2)
    {
```

```
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectDatabase();
```

```
        this.jframe=jframe;
        this.insert2=insert2;
        this.update2=update2;
        this.view2=view2;
        this.delete2=delete2;
```

```
        JL_Booking_id=new JLabel("Booking_id:");
        JTF_Booking_id=new JTextField(10);
        JL_Toname=new JLabel("To Name:");
        JTF_Toname=new JTextField(10);
        JL_Weight=new JLabel("Weight:");
```

```
        JTF_Weight=new JTextField(10);
        JL_Toaddress=new JLabel("To Address:");
        JTF_Toaddress=new JTextField(10);
```

```
        JL_Branch_id=new JLabel("Branch_id:");
        JTF_Branch_id=new JTextField(10);
```

## Student Assignments

```
this.pn=pn;

}
public void connectDatabase()
{
    try
    {
        Connection con=DriverManager.getConnection(
            "jdbc:oracle:thin:@localhost:1521:xe","amruth","vasavi");

        stmt=con.createStatement();
        stmt.executeUpdate("commit");

    }
    catch (SQLException connectException)
    {
        System.out.println(connectException.getMessage());
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
        System.exit(1);
    }
}
private void displaySQLErrors(SQLException e)
{
    JOptionPane.showMessageDialog(pn1,"nSQLException: " + e.getMessage() + "n"+"SQLState: " +
e.getSQLState() + "n"+"VendorError: " + e.getErrorCode() + "n");
}
public void loadbranch()
{
    try
    {
        Branchid=new Choice();
        Branchid.removeAll();
        rs=stmt.executeQuery("select * from branch");
        while(rs.next())
        {
            Branchid.add(rs.getString("Id"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
public void loadbooking()
{
    try
    {
        BookingList=new List();
        BookingList.removeAll();
        rs=stmt.executeQuery("select * from booking");
        while(rs.next())
        {
            BookingList.add(rs.getString("Bid"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
```

1602-19-737-066  
M Amruth Sai



## Student Assignments

```
        displaySQLErrors(e);
    }
}
public void buildGUI()
{
    insert2.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent aevt)
        {
            JB_insert=new JButton("Insert");

            JTF_Booking_id.setText(null);
            JTF_Toname.setText(null);
            JTF_Weight.setText(null);
            JTF_Toaddress.setText(null);
            JTF_Branch_id.setText(null);

            loadbooking();

            pn.removeAll();
            jframe.invalidate();
            jframe.validate();
            jframe.repaint();

            pn1=new JPanel();
            pn1.setLayout(new GridLayout(10,10));
            pn1.add(JL_Booking_id);
            pn1.add(JTF_Booking_id);
            pn1.add(JL_Toname);
            pn1.add(JTF_Toname);
            pn1.add(JL_Weight);
            pn1.add(JTF_Weight);
            pn1.add(JL_Toaddress);
            pn1.add(JTF_Toaddress);
            pn1.add(JL_Branch_id);
            pn1.add(JTF_Branch_id);

            pn3=new JPanel(new FlowLayout());
            pn3.add(JB_insert);
            pn1.setBounds(115,80,300,250);
            pn3.setBounds(200,350,75,35);

            pn2=new JPanel(new FlowLayout());
            BookingList=new List(10);
            loadbooking();
            pn2.add(BookingList);
            pn2.setBounds(200,350,300,180);

            pn.add(pn1);
            pn.add(pn3);
            pn.add(pn2);

            pn.setLayout(new BorderLayout());
            jframe.add(pn);
            jframe.setSize(800,800);
            jframe.validate();
        }
    });
}
```

## Student Assignments

```
JB_insert.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        try
        {
            String query= "INSERT INTO Booking VALUES("+
JTF_Booking_id.getText() + ","+JTF_Toname.getText() +","+JTF_Toaddress.getText()+","+JTF_Weight.getText()
+","+JTF_Branch_id.getText()+")";

            int i = stmt.executeUpdate(query);
JOptionPane.showMessageDialog(pn,"\nInserted "+i+" rows
successfully");

            loadbooking();
            System.out.println("Done");
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
    }
});

});
update2.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_modify=new JButton("Modify");

        JTF_Booking_id.setText(null);
        JTF_Toname.setText(null);
        JTF_Weight.setText(null);
        JTF_Toaddress.setText(null);
        JTF_Branch_id.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));

        pn1.add(JL_Booking_id);
        pn1.add(JTF_Booking_id);
        pn1.add(JL_Toname);
        pn1.add(JTF_Toname);
        pn1.add(JL_Weight);
        pn1.add(JTF_Weight);
        pn1.add(JL_Toaddress);
        pn1.add(JTF_Toaddress);
        pn1.add(JL_Branch_id);
        pn1.add(JTF_Branch_id);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_modify);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);
```

## Student Assignments

```
pn2=new JPanel(new FlowLayout());
BookingList=new List(10);
loadbooking();
pn2.add(BookingList);
pn2.setBounds(200,350,300,180);

pn.add(pn1);
pn.add(pn3);
pn.add(pn2);

pn.setLayout(new BorderLayout());
jframe.add(pn);
jframe.setSize(800,800);
jframe.validate();

BookingList.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent ievt)
    {
        try
        {
            rs=stmt.executeQuery("select * from Booking");
            while (rs.next())
            {
                if
(rs.getString("Bid").equals(BookingList.getSelectedItem()))
                break;
            }
            if (!rs.isAfterLast())
            {
                JTF_Booking_id.setText(rs.getString("Bid"));
                JTF_Toname.setText(rs.getString("ToName"));
                JTF_Weight.setText(rs.getString("Weight"));
                JTF_Toaddress.setText(rs.getString("ToAddress"));
                JTF_Branch_id.setText(rs.getString("Id"));
            }
        }
        catch (SQLException selectException)
        {
            displaySQLErrors(selectException);
        }
    }
});
JB_modify.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        try
        {
            int a=JOptionPane.showConfirmDialog(pn,"Are you sure
want to update:");

            if(a==JOptionPane.YES_OPTION)
            {
                String pack=JOptionPane.showInputDialog(pn,"Enter
New Name:");

                JTF_Toname.setText(pack);
                String query="update booking set
ToName="+pack+" where Bid="+JTF_Booking_id.getText();
M Amruth Sai
```

## Student Assignments

```
int i=stmt.executeUpdate(query);
OptionPane.showMessageDialog(pn,"nUpdated

"+i+" rows succesfully");

loadbooking();

}
}
catch(SQLException e)
{
    displaySQLErrors(e);
}
}
});
}
});
delete2.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_delete=new JButton("Delete");

        JTF_Booking_id.setText(null);
        JTF_Toname.setText(null);
        JTF_Weight.setText(null);
        JTF_Toaddress.setText(null);
        JTF_Branch_id.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_Booking_id);
        pn1.add(JTF_Booking_id);
        pn1.add(JL_Toname);
        pn1.add(JTF_Toname);
        pn1.add(JL_Weight);
        pn1.add(JTF_Weight);
        pn1.add(JL_Toaddress);
        pn1.add(JTF_Toaddress);
        pn1.add(JL_Branch_id);
        pn1.add(JTF_Branch_id);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_delete);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        BookingList=new List(10);
        loadbooking();
        pn2.add(BookingList);
        pn2.setBounds(200,350,300,200);

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);
```

## Student Assignments

```
pn.setLayout(new BorderLayout());
jframe.add(pn);
jframe.setSize(800,800);
jframe.validate();

BookingList.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent ievt)
    {
        try
        {
            rs=stmt.executeQuery("select * from Booking");
            while (rs.next())
            {
                if
(rs.getString("Bid").equals(BookingList.getSelectedItem()))
                    break;
            }
            if (!rs.isAfterLast())
            {
                JTF_Booking_id.setText(rs.getString("Bid"));
                JTF_Toname.setText(rs.getString("ToName"));
                JTF_Weight.setText(rs.getString("weight"));
                JTF_Toaddress.setText(rs.getString("ToAddress"));
                JTF_Branch_id.setText(rs.getString("Id"));
            }
        }
        catch (SQLException selectException)
        {
            displaySQLErrors(selectException);
        }
    }
});
JB_delete.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        try
        {
            int a=JOptionPane.showConfirmDialog(pn,"Are you sure
want to Delete:");

            if(a==JOptionPane.YES_OPTION)
            {
                //String query="DELETE FROM employee WHERE
                String query="DELETE FROM Booking WHERE
                employee_id="+employeeList.getSelectedItem();
                Bid="+JTF_Booking_id.getText();
                "+i+" rows succesfully");

                int i=stmt.executeUpdate(query);
                JOptionPane.showMessageDialog(pn,"nDeleted
                loadbooking();
            }
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
    }
});
```

## Student Assignments

```
        });
    }
});
view2.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        JLabel view=new JLabel("Booking View");
        JB_view=new JButton("View");
        Font myFont = new Font("Serif",Font.BOLD,25);
        // p2.setBackground(Color.cyan) ;
        view.setFont((myFont));

        pn1=new JPanel();
        pn2=new JPanel();
        pn1.add(view);
        pn2.add(JB_view);
        pn1.setBackground(Color.cyan) ;
        pn2.setBackground(Color.cyan) ;
        pn.add(pn1);
        pn.add(pn2);
        pn.setBounds(500,800,300,300);
        pn.setLayout(new FlowLayout());
        // pn.setBackground(Color.cyan) ;

        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        JB_view.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                JFrame jf=new JFrame("Booking Details");
                JTable jt;
                DefaultTableModel model = new DefaultTableModel();
                jt = new JTable(model);
                model.addColumn("Booking_id");
                model.addColumn("To name");
                model.addColumn("Weight");
                model.addColumn("To address");
                model.addColumn("Branch Id");
                try
                {
                    rs=stmt.executeQuery("select * from booking");
                    while(rs.next())
                    {
                        model.addRow(new Object[]{rs.getString("Bid"),

rs.getString("ToName"),rs.getString("Weight"),rs.getString("ToAddress")
,rs.getString("Id")});
                    }
                }
                catch(SQLException e)
            }
        });
    }
});
```

## Student Assignments

```
        {
            displaySQLErrors(e);
        }
        jt.setEnabled(false);
        jt.setBounds(30, 40, 300, 300);
        JScrollPane jsp = new JScrollPane(jt);
        jf.add(jsp);
        jf.setSize(800, 400);
        jf.setVisible(true);
    }
    });
}
});
}
```

## SCHEDULE:

```
package dbms;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.Properties;

public class Schedule
{
    private JPanel pn,pn1,pn2,pn3;
    private JFrame jframe;
    private JButton JB_insert,JB_modify,JB_view,JB_delete;
    private JLabel JL_Schedule_id,JL_Booking_id,JL_Branch_id,JL_Staff_id;
    private JTextField JTF_Schedule_id,JTF_Booking_id,JTF_Branch_id,JTF_Staff_id;
    Connection con;
    ResultSet rs;
    Statement stmt;
    private JMenuitem insert4,update4,view4,delete4;
    private List ScheduleList;
    private Choice BookingId,BranchId,StaffId;

    public Schedule(JPanel pn,JFrame jframe,JMenuitem insert4,JMenuitem update4,JMenuitem
view4,JMenuitem delete4)
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectDatabase();

        this.jframe=jframe;
        this.insert4=insert4;
        this.update4=update4;
        this.view4=view4;
        this.delete4=delete4;
    }
}
```

1602-19-737-066

M Amruth Sai

## Student Assignments

```
JL_Schedule_id=new JLabel("Schedule Id :");
JTF_Schedule_id=new JTextField(10);
JL_Booking_id=new JLabel("Booking Id:");
JTF_Booking_id=new JTextField(10);
JL_Branch_id=new JLabel("Branch Id:");
JTF_Branch_id=new JTextField(10);
JL_Staff_id=new JLabel("Staff Id:");
JTF_Staff_id=new JTextField(10);

this.pn=pn;

}
public void connectDatabase()
{
    try
    {
        Connection con=DriverManager.getConnection(
            "jdbc:oracle:thin:@localhost:1521:xe","amruth","vasavi");

        stmt=con.createStatement();
        stmt.executeUpdate("commit");

    }
    catch (SQLException connectException)
    {
        System.out.println(connectException.getMessage());
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
        System.exit(1);
    }
}
private void displaySQLErrors(SQLException e)
{
    JOptionPane.showMessageDialog(pn1,"\\nSQLException: " + e.getMessage() + "\\n"+"SQLState: " +
e.getSQLState() + "\\n"+"VendorError: " + e.getErrorCode() + "\\n");
}
public void loadBooking()
{
    try
    {
        BookingId=new Choice();
        BookingId.removeAll();
        rs=stmt.executeQuery("select * from Booking");
        while(rs.next())
        {
            BookingId.add(rs.getString("Bid"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
public void loadbranch()
{
    try
    {
        BranchId=new Choice();
        BranchId.removeAll();
        rs=stmt.executeQuery("select * from Branch");
        while(rs.next())
        {
            BranchId.add(rs.getString("BranchId"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
}
```

1602-19-737-066  
M Amruth Sai



## Student Assignments

```
        BranchId.removeAll();
        rs=stmt.executeQuery("select * from Branch");
        while(rs.next())
        {
            BranchId.add(rs.getString("Id"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
public void loadstaff()
{
    try
    {
        StaffId=new Choice();
        StaffId.removeAll();
        rs=stmt.executeQuery("select * from Staff");
        while(rs.next())
        {
            StaffId.add(rs.getString("Sid"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
public void loadschedule()
{
    try
    {
        ScheduleList=new List();
        ScheduleList.removeAll();
        rs=stmt.executeQuery("select * from Schedule");
        while(rs.next())
        {
            ScheduleList.add(rs.getString("Shid"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
public void buildGUI()
{
    insert4.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent aevt)
        {
            JB_insert=new JButton("Insert");

            JTF_Schedule_id.setText(null);
            JTF_Booking_id.setText(null);
            JTF_Branch_id.setText(null);
            JTF_Staff_id.setText(null);
        }
    });
}
```

```

        loadschedule();

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_Schedule_id);
        pn1.add(JTF_Schedule_id);
        pn1.add(JL_Booking_id);
        pn1.add(JTF_Booking_id);
        pn1.add(JL_Branch_id);
        pn1.add(JTF_Branch_id);
        pn1.add(JL_Staff_id);
        pn1.add(JTF_Staff_id);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_insert);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        ScheduleList=new List(10);
        loadschedule();
        pn2.add(ScheduleList);
        pn2.setBounds(200,350,300,180);

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        JB_insert.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                try
                {
                    String query= "INSERT INTO Schedule VALUES(" +
JTF_Schedule_id.getText() + ","+JTF_Booking_id.getText()+"," +
+JTF_Staff_id.getText()+","+JTF_Branch_id.getText()+")";
                    int i = stmt.executeUpdate(query);
                    JOptionPane.showMessageDialog(pn,"Inserted "+i+" rows
successfully");

                    loadschedule();
                    System.out.println("Done");
                }
                catch(SQLException e)
                {
                    displaySQLErrors(e);
                }
            }
        });
    }
}

```

## Student Assignments

```
        }
    });
}

});
/*
update4.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_modify=new JButton("Modify");

        JTF_Schedule_id.setText(null);
        JTF_Booking_id.setText(null);
        JTF_Branch_id.setText(null);
        JTF_Staff_id.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_Schedule_id);
        pn1.add(JTF_Schedule_id);
        pn1.add(JL_Booking_id);
        pn1.add(JTF_Booking_id);
        pn1.add(JL_Branch_id);
        pn1.add(JTF_Branch_id);
        pn1.add(JL_Staff_id);
        pn1.add(JTF_Staff_id);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_modify);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        ScheduleList=new List(10);
        loadschedule();
        pn2.add(ScheduleList);
        pn2.setBounds(200,350,300,180);

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        ScheduleList.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent ievt)
            {
                try
                {
                    rs=stmt.executeQuery("select * from Schedule");
                    while (rs.next())
                    {
```

## Student Assignments

```

        if
(rs.getString("Schedule_id").equals(ScheduleList.getSelectedItem()))
            break;
        }
        if (!rs.isAfterLast())
        {
            JTF_Schedule_id.setText(rs.getString("Shid"));
            JTF_Booking_id.setText(rs.getString("Bid"));
            JTF_Branch_id.setText(rs.getString("Id"));
            JTF_Staff_id.setText(rs.getString("Sid"));
        }
    }
    catch (SQLException selectException)
    {
        displaySQLErrors(selectException);
    }
});
JB_modify.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        try
        {
            int a=JOptionPane.showConfirmDialog(pn,"Are you sure
want to update:");

            if(a==JOptionPane.YES_OPTION)
            {
                String pack=JOptionPane.showInputDialog(pn,"Enter
New :");

                JTF_Schedule_id.setText(pack);
                String query="update Schedule set
insurance_type='"+pack+"' where insurance_num="+JTF_insurance_num.getText();
                int i=stmt.executeUpdate(query);
                JOptionPane.showMessageDialog(pn,"nUpdated
"+i+" rows succesfully");

                loadinsurance();
            }
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
    }
});
});
*/
delete4.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_delete=new JButton("Delete");

        JTF_Schedule_id.setText(null);
        JTF_Booking_id.setText(null);
        JTF_Branch_id.setText(null);
        JTF_Staff_id.setText(null);
    }
});

```

```
pn.removeAll();
jframe.invalidate();
jframe.validate();
jframe.repaint();

pn1=new JPanel();
pn1.setLayout(new GridLayout(10,10));
pn1.add(JL_Schedule_id);
pn1.add(JTF_Schedule_id);
pn1.add(JL_Booking_id);
pn1.add(JTF_Booking_id);
pn1.add(JL_Branch_id);
pn1.add(JTF_Branch_id);
pn1.add(JL_Staff_id);
pn1.add(JTF_Staff_id);

pn3=new JPanel(new FlowLayout());
pn3.add(JB_delete);
pn1.setBounds(115,80,300,250);
pn3.setBounds(200,350,75,35);

pn2=new JPanel(new FlowLayout());
ScheduleList=new List(10);
loadschedule();
pn2.add(ScheduleList);
pn2.setBounds(200,350,300,200);

pn.add(pn1);
pn.add(pn3);
pn.add(pn2);

pn.setLayout(new BorderLayout());
jframe.add(pn);
jframe.setSize(800,800);
jframe.validate();

ScheduleList.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent ievt)
    {
        try
        {
            rs=stmt.executeQuery("select * from Schedule");
            while (rs.next())
            {
                if
(rs.getString("Shid").equals(ScheduleList.getSelectedItem()))
                break;
            }
            if (!rs.isAfterLast())
            {
                JTF_Schedule_id.setText(rs.getString("Shid"));
                JTF_Booking_id.setText(rs.getString("Bid"));
                JTF_Branch_id.setText(rs.getString("Id"));
                JTF_Staff_id.setText(rs.getString("Sid"));
            }
        }
    }
});
```

## Student Assignments

```
        }
    }
    catch (SQLException selectException)
    {
        displaySQLErrors(selectException);
    }
}

});
JB_delete.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        try
        {
            int a=JOptionPane.showConfirmDialog(pn,"Are you sure
want to Delete:");

            if(a==JOptionPane.YES_OPTION)
            {
                //String query="DELETE FROM insurance WHERE
insurance_num="+insuranceList.getSelectedItemAt();
                String query="DELETE FROM Schedule WHERE
Shid="+JTF_Schedule_id.getText();

                int i=stmt.executeUpdate(query);
                JOptionPane.showMessageDialog(pn,"\nDeleted
"+i+" rows succesfully");

                loadschedule();
            }
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
    }
});

});
view4.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        JLabel view=new JLabel("Schedule View");
        JB_view=new JButton("View");
        Font myFont = new Font("Serif",Font.BOLD,25);
        view.setFont((myFont));

        pn1=new JPanel();
        pn2=new JPanel();
        pn1.add(view);
        pn2.add(JB_view);
        pn1.setBackground(Color.cyan) ;
        pn2.setBackground(Color.cyan) ;
        pn.add(pn1);
        pn.add(pn2);
    }
});
```

## Student Assignments

```
pn.setBounds(500,800,300,300);
pn.setLayout(new FlowLayout());

jframe.add(pn);
jframe.setSize(800,800);
jframe.validate();

JB_view.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JFrame jf=new JFrame("Schedule Details");
        JTable jt;
        DefaultTableModel model = new DefaultTableModel();
        jt = new JTable(model);
        model.addColumn("Schedule Id");
        model.addColumn("Booking Id");
        model.addColumn("Branch Id");
        model.addColumn("Staff Id");
        try
        {
            rs=stmt.executeQuery("select * from Schedule");
            while(rs.next())
            {
                model.addRow(new Object[]{rs.getString("Shid"),
rs.getString("Shid"),rs.getString("Bid"),rs.getString("id"),rs.getString("sid")});
            }
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
        jt.setEnabled(false);
        jt.setBounds(30, 40, 300, 300);
        JScrollPane jsp = new JScrollPane(jt);
        jf.add(jsp);
        jf.setSize(800, 400);
        jf.setVisible(true);
    }
});
}
```

## DELIVERY:

```
package dbms;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.Properties;
import java.util.StringTokenizer;

public class Delivery
{
    private JPanel pn,pn1,pn2,pn3;
    private JFrame jframe;
    private JButton JB_insert,JB_modify,JB_view,JB_delete;
    private JLabel JL_Did,JL_Selection,JL_Bid;
    private JTextField JTF_Did,JTF_Bid,JTF_Selection;
    Connection con;
    ResultSet rs;
    Statement stmt;
    private JMenuItem insert3,update3,view3,delete3;
    private List hotelsList;
    private Choice statelId,hotelId;

    public Delivery(JPanel pn,JFrame jframe,JMenuItem insert3,JMenuItem update3,JMenuItem
view3,JMenuItem delete3)
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectDatabase();

        this.jframe=jframe;
        this.insert3=insert3;
        this.update3=update3;
        this.view3=view3;
        this.delete3=delete3;

        JL_Bid=new JLabel("BId:");
        JTF_Bid=new JTextField(10);
        JL_Did=new JLabel("DId:");
        JTF_Did=new JTextField(10);
        JL_Selection=new JLabel("Selection:");
        JTF_Selection=new JTextField(10);
        // JL_noOfRooms=new JLabel("No Of Rooms:");
        // JTF_noOfRooms=new JTextField(10);

        this.pn=pn;
    }
    public void connectDatabase()
```



## Student Assignments

```
{
    try
    {
        Connection con=DriverManager.getConnection(
            "jdbc:oracle:thin:@localhost:1521:xe","amruth","vasavi");

        stmt=con.createStatement();
        stmt.executeUpdate("commit");

    }
    catch (SQLException connectException)
    {
        System.out.println(connectException.getMessage());
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
        System.exit(1);
    }
}

private void displaySQLErrors(SQLException e)
{
    JOptionPane.showMessageDialog(pn1,"nSQLException: " + e.getMessage() + "n"+"SQLState: " +
e.getSQLState() + "n"+"VendorError: " + e.getErrorCode() + "n");
}

public void loadStatesHotels()
{
    try
    {
        hotelsList=new List();
        hotelsList.removeAll();
        rs=stmt.executeQuery("select * from Delivery");
        while(rs.next())
        {
            //      hotelsList.add(rs.getString("Did")+"->" +rs.getString("Bid"));
            hotelsList.add(rs.getString("Did"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}

public void loadStates()
{
    try
    {
        statelD=new Choice();
        statelD.removeAll();
        rs=stmt.executeQuery("select * from Booking");
        while(rs.next())
        {
            statelD.add(rs.getString("Bid"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
```

## Student Assignments

```
public void loadHotels()
{
    try
    {
        hotelId=new Choice();
        hotelId.removeAll();
        rs=stmt.executeQuery("select * from Delivery");
        while(rs.next())
        {
            hotelId.add(rs.getString("Did"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
public void buildGUI()
{
    insert3.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent aevt)
        {
            JB_insert=new JButton("Insert");

            JTF_Did.setText(null);
            JTF_Bid.setText(null);
            JTF_Selection.setText(null);
            JTF_noOfRooms.setText(null);

            loadStates();
            loadHotels();

            pn.removeAll();
            jframe.invalidate();
            jframe.validate();
            jframe.repaint();

            pn1=new JPanel();
            pn1.setLayout(new GridLayout(10,10));
            pn1.add(JL_Did);
            pn1.add(JTF_Did);
            pn1.add(JL_Bid);
            pn1.add(JTF_Bid);
            //
            //
            pn1.add(JL_Selection);
            pn1.add(JTF_Selection);
            //
            //
            pn1.add(JL_noOfRooms);
            pn1.add(JTF_noOfRooms);

            pn3=new JPanel(new FlowLayout());
            pn3.add(JB_insert);

            pn1.setBounds(115,80,300,250);
            pn3.setBounds(200,350,75,35);

            pn2=new JPanel(new FlowLayout());
```

## Student Assignments

```
        hotelsList=new List(10);
        loadStatesHotels();
        pn2.add(hotelsList);
        pn2.setBounds(200,350,300,180);

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

        //hotelsList.setEnabled(false);
        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        JB_insert.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                try
                {
                    String query= "INSERT INTO Delivery VALUES("+
JTF_Did.getText() + ","+JTF_Bid.getText() + ","+ JTF_Selection.getText() +")";
                    int i = stmt.executeUpdate(query);
                    JOptionPane.showMessageDialog(pn,"\\nInserted "+i+" rows
successfully");

                    loadStatesHotels();
                    System.out.println("Done");
                }
                catch(SQLException e)
                {
                    displaySQLErrors(e);
                }
            }
        });

    });
    update3.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent aevt)
        {
            JB_modify=new JButton("Modify");

            JTF_Bid.setText(null);
            JTF_Did.setText(null);
            JTF_Selection.setText(null);
            JTF_noOfRooms.setText(null);

            //

            pn.removeAll();
            jframe.invalidate();
            jframe.validate();
            jframe.repaint();

            pn1=new JPanel();
            pn1.setLayout(new GridLayout(10,10));
            pn1.add(JL_Bid);
            pn1.add(JTF_Bid);
            pn1.add(JL_Did);
            pn1.add(JTF_Did);
```

## Student Assignments

```
pn1.add(JTF_Did);
pn1.add(JL_Selection);
pn1.add(JTF_Selection);
pn1.add(JL_noOfRooms);
pn1.add(JTF_noOfRooms);

//
//

pn3=new JPanel(new FlowLayout());
pn3.add(JB_modify);
pn1.setBounds(115,80,300,250);
pn3.setBounds(200,350,75,35);

pn2=new JPanel(new FlowLayout());
hotelsList=new List(10);
loadStatesHotels();
pn2.add(hotelsList);
pn2.setBounds(250,350,300,180);

pn.add(pn1);
pn.add(pn3);
pn.add(pn2);

pn.setLayout(new BorderLayout());
jframe.add(pn);
jframe.setSize(800,800);
jframe.validate();

hotelsList.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent ievt)
    {
        try
        {
            rs=stmt.executeQuery("select * from Delivery");
            while (rs.next())
            {
                if
                (rs.getString("Did").equals(hotelsList.getSelectedItem()))
                {
                    break;
                }
                if (!rs.isAfterLast())
                {
                    JTF_Did.setText(rs.getString("Did"));
                    JTF_Selection.setText(rs.getString("Selection"));
                }
            }
        }
        catch (SQLException selectException)
        {
            displaySQLErrors(selectException);
        }
    }
});
JB_modify.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
```

## Student Assignments

```
        {
            try
            {
                int a=JOptionPane.showConfirmDialog(pn1,"Are you sure do
you want to update:");

                if(a==JOptionPane.YES_OPTION)
                {
                    String
rooms=JOptionPane.showInputDialog(pn,"Enter selection:");
                    JTF_Selection.setText(rooms);
                    String query="update Delivery set
                    " where Did="+JTF_Did.getText());

                    int i=stmt.executeUpdate(query);
                    JOptionPane.showMessageDialog(pn,"\\nUpdated
"+i+" rows successfully");

                    loadStatesHotels();
                }
            }
            catch(SQLException exp)
            {
                displaySQLErrors(exp);
            }
        }
    });
});

delete3.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_delete=new JButton("Delete");

        JTF_Bid.setText(null);
        JTF_Did.setText(null);
        JTF_Selection.setText(null);
        JTF_noOfRooms.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_Bid);
        pn1.add(JTF_Bid);
        pn1.add(JL_Did);
        pn1.add(JTF_Did);
        pn1.add(JL_Selection);
        pn1.add(JTF_Selection);
        pn1.add(JL_noOfRooms);
        pn1.add(JTF_noOfRooms);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_delete);
```

## Student Assignments

```
pn1.setBounds(115,80,300,250);
pn3.setBounds(180,350,75,35);

pn2=new JPanel(new FlowLayout());
hotelsList=new List(10);
loadStatesHotels();
pn2.add(hotelsList);
pn2.setBounds(250,350,300,180);

pn.add(pn1);
pn.add(pn3);
pn.add(pn2);

pn.setLayout(new BorderLayout());
jframe.add(pn);
jframe.setSize(800,800);
jframe.validate();

hotelsList.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent e)
    {
        try
        {
            rs=stmt.executeQuery("select * from Delivery");
            StringTokenizer st=new
StringTokenizer(hotelsList.getSelectedItem());

            //String p=st.nextToken();
            String q=st.nextToken();

            while (rs.next())
            {

                //if (rs.getString("state_Id").equals(p) &&
                if (rs.getString("Did").equals(q))
                    break;
            }
            if (!rs.isAfterLast())
            {
                JTF_Did.setText(rs.getString("Did"));
                JTF_Selection.setText(rs.getString("Selection"));
            }
            //
            JTF_noOfRooms.setText(rs.getString("no_Of_Rooms_Available"));
            JTF_Bid.setText(rs.getString("Bid"));

        }
        catch (SQLException selectException)
        {
            displaySQLErrors(selectException);
        }
    }
});
JB_delete.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
```

## Student Assignments

```
        try
        {
            int a=JOptionPane.showConfirmDialog(pn,"Are you sure do
you want to delete:");

            if(a==JOptionPane.YES_OPTION)
            {
                StringTokenizer st=new
StringTokenizer(hotelsList.getSelectedItem(),"->");

                String query="DELETE FROM Delivery WHERE
Did="+st.nextToken();

                int i=stmt.executeUpdate(query);
                JOptionPane.showMessageDialog(pn,"\\nDeleted
"+i+" rows successfully");

                loadStatesHotels();
            }
        }
        catch(SQLException exp)
        {
            displaySQLErrors(exp);
        }
    }
});
});
view3.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        JB_view=new JButton("View");
        JLabel view=new JLabel("Delivery View");
        Font myFont = new Font("Serif",Font.BOLD,25);
        view.setFont((myFont));

        pn1=new JPanel();
        pn2=new JPanel();
        pn1.setBackground(Color.cyan) ;
        pn2.setBackground(Color.cyan) ;
        pn1.add(view);
        pn2.add(JB_view);
        pn.add(pn1);
        pn.add(pn2);
        pn.setLayout(new FlowLayout());

        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        JB_view.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                JFrame jf=new JFrame("Delivery Details");
                JTable jt;
```

```
DefaultTableModel model = new DefaultTableModel();

jt= new JTable(model);
model.addColumn("Delivery Id");
model.addColumn("selection");
model.addColumn("Nos");
model.addColumn("Booking Id");

//

try
{
    rs=stmt.executeQuery("select * from Delivery");
    while(rs.next())
    {
        model.addRow(new Object[]{rs.getString("Did"),
        rs.getString("Selection"),rs.getString("Bid")});
    }
}
catch(SQLException exp)
{
    displaySQLErrors(exp);
}
jt.setEnabled(false);
jt.setBounds(30, 40, 180, 150);
JScrollPane sp = new JScrollPane(jt);
jf.add(sp);
jf.setSize(800, 400);
jf.setVisible(true);
}
});
}
}
```

## Github links and folder

**structure:** Link:




<https://github.com/Amruth066/>

[DBMS 2](#)

Folder structure:

























## Student Assignments


 |  |  | Courier Management

**File** Home Share View

← → ↑ > Courier Management

 Quick access

-  Desktop
-  Downloads
-  Documents
-  Pictures
-  .settings
-  DBMS Lab
-  diagrams
-  SE
-  OneDrive
-  This PC
-  3D Objects
-  Desktop
-  Documents
-  Downloads
-  Music
-  Pictures
-  Videos
-  Local Disk (C:)
-  Local Disk (E:)
-  Local Disk (F:)
-  Network

Name	Date modified	Type	Size
 CMS	6/24/2021 12:20 PM	File folder	

< > Courier Management > CMS > src > dbms








ess

ads

nts

ab

s

Name	Date modified	Type	Size
 Booking	6/19/2021 11:15 PM	JAVA File	13 KB
 Branch	6/19/2021 11:02 PM	JAVA File	13 KB
 dbmsUI	6/19/2021 12:12 PM	JAVA File	6 KB
 Delivery	6/19/2021 11:16 PM	JAVA File	13 KB
 main	6/17/2021 12:55 PM	JAVA File	1 KB
 Schedule	6/19/2021 11:17 PM	JAVA File	13 KB
 Staff	6/19/2021 11:03 PM	JAVA File	16 KB

## Student Assignments

File Explorer interface showing the contents of the CMS directory.

Address bar: CMS > Courier Management > CMS

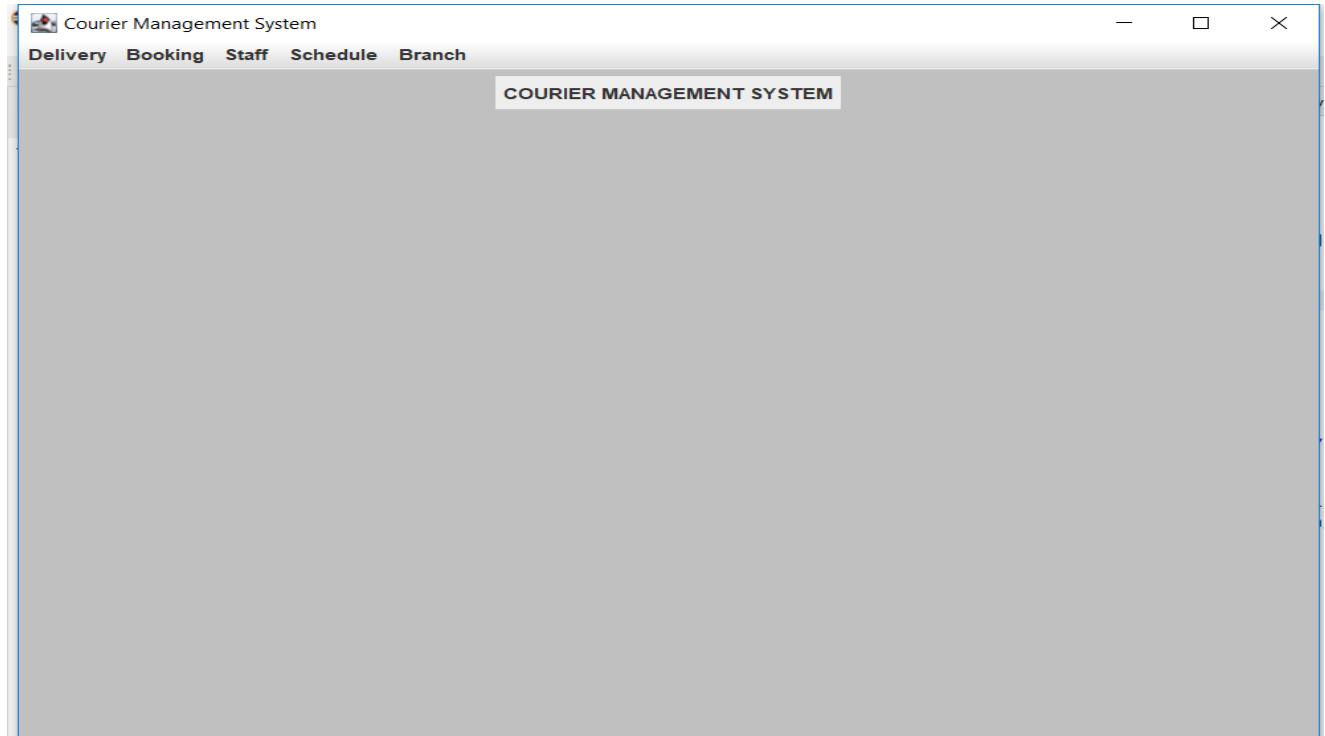
Name	Date modified	Type	Size
.settings	6/24/2021 12:18 PM	File folder	
bin	4/14/2020 9:42 PM	File folder	
src	6/24/2021 12:19 PM	File folder	
.classpath	6/16/2021 4:42 PM	CLASSPATH File	1 KB
.project	6/16/2021 4:39 PM	PROJECT File	1 KB

Left sidebar (Quick access):

- Desktop
- Downloads
- Documents
- Pictures
- .settings
- DBMS Lab
- diagrams
- SE
- OneDrive
- This PC
  - 3D Objects
  - Desktop
  - Documents
  - Downloads
  - Music
  - Pictures
  - Videos
- Local Disk (C:)
- Local Disk (E:)
- Local Disk (F:)
- Network

## TESTING

### *OUTPUT SCREENSHOTS:*



## Student Assignments

Courier Management System

Delivery Booking Staff Schedule Branch

Branch ID	6
Branch Name	branch6
Branch Code	BR105
Branch Address	abc abc

Insert

1  
2  
3  
4  
5

Message

inserted 1 rows succesfully

OK

## Student Assignments

Courier Management System

Delivery Booking Staff Schedule Branch

Branch ID	5
Branch Name	branch5
Branch Code	BR104
Branch Address	mno xyz

1  
2  
3  
4  
5  
6

Delete

Select an Option

Are you sure want to delete:

Yes No Cancel

## Student Assignments

Courier Management System

Delivery Booking Staff Schedule Branch

Branch ID	5
Branch Name	branch6
Branch Code	BR105
Branch Address	abc abc

Update

Message

updated 1 rows succesfully

OK

1  
2  
3  
4  
6

## Student Assignments



Courier Management System

**Delivery** **Booking** **Staff** **Schedule** **Branch**

### Booking View


View



Booking Details

Booking_id	To name	Weight	To address	Branch
31	root	10	xyz abc	1
32	morgan	12	xyz def	1
33	Amruth	15	xyz ghi	2
34	Yvuvi	20	xyz jkl	4
35	amruth	56	abc abc	3

## Student Assignments

 Courier Management System

**Delivery** **Booking** **Staff** **Schedule** **Branch**

**Booking\_id:**

35

**To Name:**

amruth

**Weight:**

56

**To Address:**

abc abc

**Branch\_id:**

3

Message



Inserted 1 rows successfully

OK

Insert

31  
32  
33  
34



## Student Assignments



Courier Management System

**Delivery** **Booking** **Staff** **Schedule** **Branch**

**View**


**View**



Staff details

sid	staff_name	staff_contact
21	Sachin	9999999991
22	jack	9999999992
23	john	9999999993
24	warner	9999999994
25	smith	9999999997
26	amruth .p	654564564

## Student Assignments

 Courier Management System

Delivery Booking Staff Schedule Branch

Staff ID

26

Staff Name

amruth .p

Staff Contact

654564564

21

22


23

24

25


Insert

Message

 inserted 1 rows succesfully

OK


## Student Assignments

 Courier Management System

[Delivery](#) [Booking](#) [Staff](#) [Schedule](#) [Branch](#)

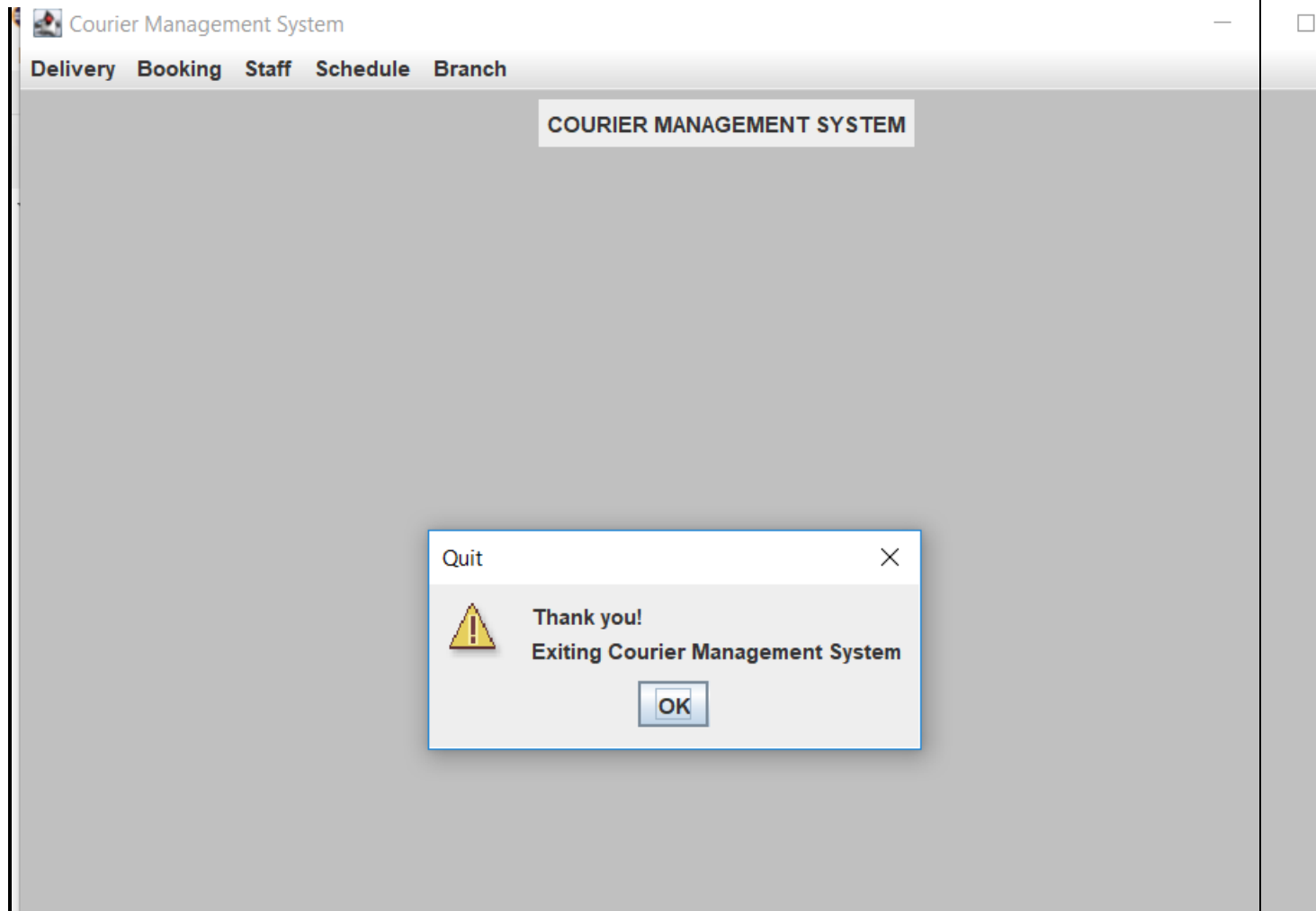
View

View

 Branch details

Branch Id	Branch Name	Branch Code	Branch Address
1	branch1	BR100	abc xyz
2	branch2	BR101	def xyz
3	branch3	BR102	ghi xyz
4	branch4	BR103	jkm xyz
6	branch6	BR105	abc abc

## Student Assignments



## Student Assignments



Courier Management System

**Delivery** Booking Staff Schedule Branch

### Delivery View

View



#### Delivery Details

Delivery Id	selection	Booking Id
51	YES	31
52	NO	32
55	No	34
54	Yes	34
53	Yes	33

## Student Assignments



Courier Management System

**Delivery** **Booking** **Staff** **Schedule** **Branch**

**Booking\_id:**

32

**To Name:**

Root

**Weight:**

54

**To Address:**

abc abc

**Branch\_id:**

2

Input



Enter New Name:


OK

Cancel

Modify

31

## Student Assignments

 Courier Management System

**Delivery** Booking Staff Schedule Branch

**DId:**

52

**BId:**

31

**Selection:**

YES

Message



Inserted 1 rows successfully

OK

Insert

51

## Student Assignments

 Courier Management System

**Delivery** Booking Staff Schedule Branch

**Booking\_id:**

32

**To Name:**

morgan

**Weight:**

12

**To Address:**

xyz def

**Branch\_id:**

1

Message



Deleted 1 rows succesfully

OK

Delete

31  
32



## Student Assignments



Courier Management System

**Delivery** Booking Staff Schedule Branch

**Staff ID**

23

**Staff Name**

john

**Staff Contact**

9999999998

Update

Message



updated 1 rows succesfully


OK

21

22

23

## Student Assignments

 Courier Management System

**Delivery** Booking Staff Schedule Branch


### Delivery View

View



#### Delivery Details

Delivery Id	selection	Booking Id
51	YES	31
52	YES	31

 Courier Management System

Delivery

Booking


Staff

Schedule

Branch

Schedule View

View

 Schedule Details

Schedule Id	Booking Id	Branch Id	Staff Id
41	41	31	1
42	42	32	2
43	43	33	3

## DATABASE OUTPUTS:

BEFORE :

```
SQL> select * from branch;
```

ID	BRANCH_NAME	BRANCH_CODE	BRANCH_ADDRESS
1	branch1	BR100	abc xyz

```
SQL> select * from staff;
```

SID	STAFF_NAME	STAFF_CONTACT
21	Sachin	9999999991
22	jack	9999999992
23	john	9999999993
24	warner	9999999994
25	smith	9999999997
26	amruth .p	654564564

6 rows selected.

```
SQL> select * from schedule;
```

SHID	BID	SID	ID
41	31	21	1

:

```
SQL> select * from Booking;
```

BID	TONAME	TOADDRESS	WEIGHT	ID
31	root	xyz abc	10	1

## Student Assignments

```
SQL> select * from delivery;
```

DID	BID SELECTION
51	31 YES
52	31 YES

AFTER :

```
SQL> select * from branch;
```

ID	BRANCH_NAME	BRANCH_CODE
1	branch1	BR100
2	branch2	BR101

```
SQL> select * from staff;
```

SID	STAFF_NAME	STAFF_CONTACT
21	Sachin	9999999991
22	jack	9999999992
23	john	9999999993

```
SQL> select * from booking;
```

BID	TONAME	TOADDRESS	WEIGHT	ID
31	root	xyz abc	10	1
33	ramesh	ptr ptr	55	1
32	amruth	pqr pqr	54	2

```
SQL>
```

```
SQL> select * from schedule;
```

SHID	BID	SID	ID
41	31	21	1
42	32	22	2
43	33	21	2

```
SQL> select * from delivery;
```

DID	BID	SELECTION
51	31	YES
54	31	NO
52	31	YES
53	33	No

## DISCUSSION AND FUTURE WORK

So far this project has helped us assign assignments to students by staff and students will be able to view assignments their marks, due dates and feedback

In future we try to include an option which allows students to know their mistakes in their assignments and staff can highlight the mistakes by students in their respective submissions.

## REFERENCES:

1. Abraham Silberschatz, Henry F Korth, S. Sudarshan, Database System Concepts, 6th Edition, McGraw-Hill International Edition, 2010.
2. [https://docs.oracle.com/cd/E11882\\_01/server.112/e40540/intro.htm#CNCPT001](https://docs.oracle.com/cd/E11882_01/server.112/e40540/intro.htm#CNCPT001)

## Student Assignments

## Student Assignments



## Student Assignments

## Student Assignments

## Student Assignments

## Student Assignments

## Student Assignments

## Student Assignments

## Student Assignments

## Student Assignments



## Student Assignments