

UCS1412 – Database Lab

Mini Project

Title: Food Delivery Service And Database Management System

Abstract

Description

In today's world, there is an increasing amount of innovation that centers on delivering goods to a person's doorstep, whether that may be groceries, clothes, furniture, etc. Hence, we decided to build an application for a food delivery service company to help them analyze data about their business transactions. This app provides a UI where the company can view information about customers (including their names, phone numbers, and addresses), employees (their names and job positions), and restaurants participating in the service (their addresses and menu items). As such, the app's intended clients are the managers of the company, since they will likely be responsible for keeping track of current employees, customers, and restaurant data. For example, they can query each customer's order history and the number of orders fulfilled within a given time period. These queries will assist managers in making decisions about employee bonuses, customer discounts, etc.

For overall functionality of the application, we would have to be able to query how many hours each employee has worked, be able to create a contract with a new restaurant to join the delivery program, and be able to view the order histories of individual customers (more functionalities are listed below). We would also take into consideration limitations such as how many hours an employee can work per day and that drivers in a particular region can only deliver food from restaurants from that region to customers in the same region (constraints listed below).

Functionalities

a) Transactions:

Employee transactions:

1. Generate a list of all the employees.
2. Show all the employees who work in a given region
3. Generate a list of supervisors and the drivers they supervise.
4. Create a new employee contract (input employee name, phone number, region, etc).
5. Ratings For Driver.

Restaurant transactions:

1. Generate a list of all restaurants that have partnered with the delivery service.
2. Generate a list of restaurants located in a given region.
3. Generate a list of menu items and their prices from a given restaurant.

4. Ratings For Restaurants are shown in the place order screen.

Customer transactions:

1. Create customer account
2. Customer Sign In
3. Show all the restaurants in the same region as the customer.
4. Show the order history of a given customer.

b) Constraints:

1. There is only one supervisor per region.
Enforced with an insert trigger on the Supervisors table. (check_spvsvr)
2. A customer can only order food from restaurants in his/her region.
Enforced with an insert trigger on the Orders table. (fromSameRegion)
And also within the application.
3. Each driver can only complete orders that are in the same region as the restaurant and the customer, and himself/herself.
Enforced with an insert trigger on the Orders table. (fromSameRegion)
4. A driver can only be employed in one region
Enforced through FD empID->empRegion. User of application is also only allowed to select one option from a given drop-down list of regions.
5. A driver cannot work for more than 8 hours per day.
Enforced with a check() constraint during relation creation.

FDs and Normalization

Employees, Supervisors, Drivers Tables

empId -> empName, empRegion, empPhone, empHours_worked, salary, licenseNo, wage, supervisorId, dRating

Customers Table

cId -> cName, cRegion, cPhone, cAddress

Orders Table

orderId -> total, cId, time_made, rId, empId, time_delivered

Restaurants Table

rId -> rName, rCuisine, rRegion, rPhone, rAddress, rRating

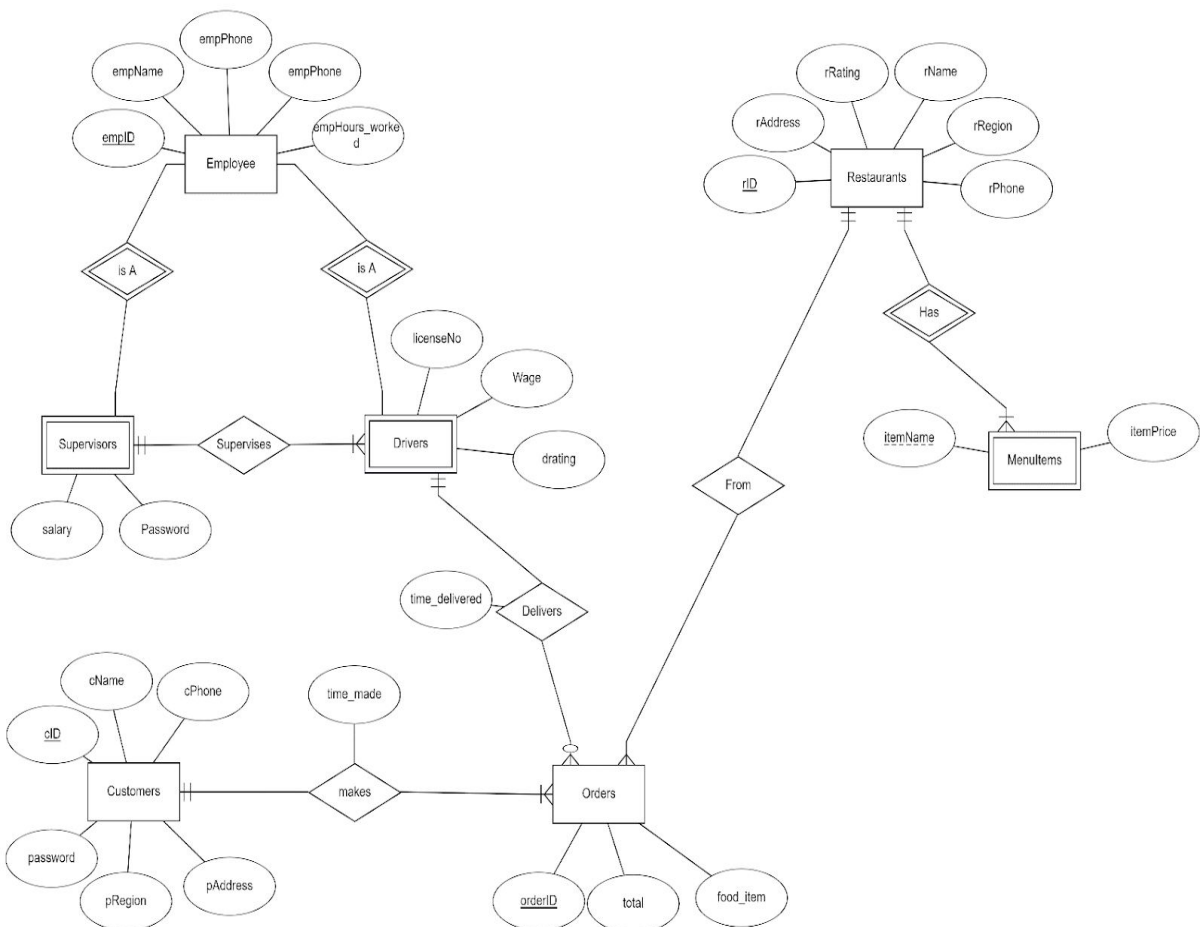
MenuItems table

rId, itemName -> itemPrice

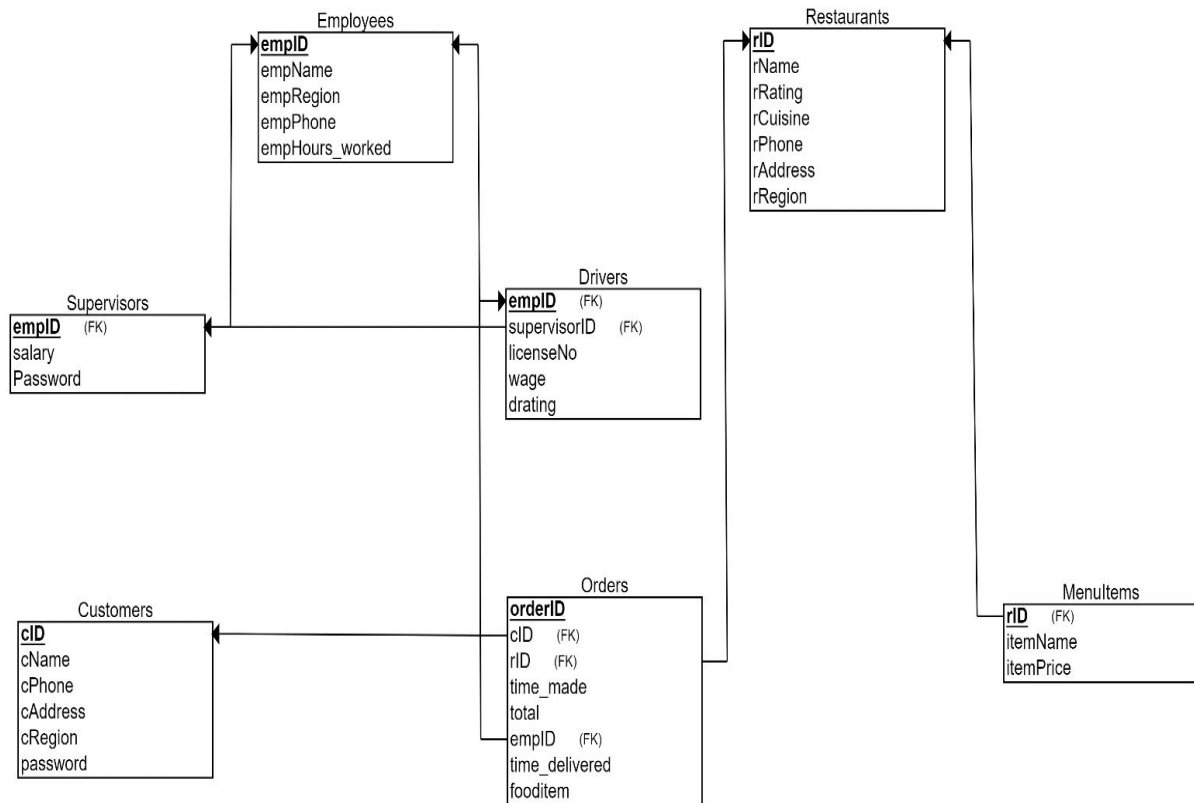
Normalization:

All determinants are **super keys to their respective tables**, therefore all relations are in **BCNF** form.

Er Diagram



ER to Relational Schema



Table, Procedures and Functions Definitions

-- Employee Table

```
Create table Employees(  
empId varchar2(10) constraint EMP_ID_PK primary key,  
empName varchar2(20),  
empRegion varchar2(15),  
empPhone varchar2(13),  
empHours_worked number(3,2),  
constraint CHK_EMPHRS check (empHours_worked <= 8));
```

-- Supervisor Table

```
Create table Supervisors(  
empId varchar2(10),  
salary number(9, 2),  
password varchar2(20),  
constraint SUP_PK primary key (empId),  
constraint SUP_EID_FK foreign key (empId) references Employees  
(empId));
```

-- Driver Table

```
Create table Drivers(  
empId varchar2(10),  
licenseNo varchar2(8),  
wage number(4, 2),  
drating float,  
supervisorId varchar2(10),  
constraint DRI_PK primary key (empId),  
constraint DRI_EID_FK foreign key (empId) references Employees  
(empId),  
constraint DRI_SID_FK foreign key (supervisorId) references  
Supervisors (empId));
```

-- Customers Table

```
Create table Customers(  
cId varchar2(5) constraint CID_PK primary key,  
cName varchar2(20),  
cPhone varchar2(10),  
cAddress varchar2(25),  
cRegion varchar2(15),  
password varchar2(20));
```

-- Restaurants Table

```
Create table Restaurants(  
rId varchar2(5) constraint RtID_pk primary key,  
rName varchar2(20),  
rrating float,  
rCuisine varchar2(15),  
rPhone varchar2(13),
```

```

rAddress varchar2(30),
rRegion varchar2(15));

-- Menu Items Table
Create table MenuItems(
rId varchar2(5),
itemName varchar2(25),
itemPrice number(5,2),
constraint MENU_ITMS_PK primary key (rId, itemName),
constraint MENU_ITEMS_RID_FK foreign key (rId) references
Restaurants (rId));

-- Orders Table
Create table Orders(
orderId varchar2(5) constraint ORDID_PK primary key,
cId varchar2(5),
rId varchar2(5),
time_made Date,
total number(5,2),
empId varchar2(10),
time_delivered Date,
fooditem varchar2(100),
constraint ORD_CID_FK foreign key      (cId) references Customers
(cId),
constraint ORD_RID_FK foreign key      (rId) references Restaurants
(rId),
constraint ORD_EID_FK foreign key      (empId) references Drivers
(empId));

-- SEQUENCES
-- Employee Sequence
Create sequence emp_seq
start with 1
increment by 1
minvalue 0
maxvalue 500
NOCACHE
NOCYCLE;

-- Restaurant Sequence
Create sequence rest_seq
start with 1
increment by 1
minvalue 0
maxvalue 500
NOCACHE
NOCYCLE;

-- TRIGGERS --

```

-- trigger to check whether while inserting a supervisor if there already exists a supervisor from the region

CREATE OR REPLACE TRIGGER CHECK_SUPVSR

BEFORE INSERT ON Supervisors

FOR EACH ROW

DECLARE

ct integer :=0;

sreg varchar(15);

BEGIN

**/*get the region of the supervisor trying to be inserted
(assumes the new supervisor is already an employee) */**

Select empRegion

into sreg

from Employees

where empId = :NEW.empId;

/* find the count of supervisors who has the same region as the inserted supervisor */

Select count(empId)

into ct

FROM Employees natural join Supervisors

Where empRegion = sreg;

if ct > 0 then

RAISE_APPLICATION_ERROR(-20000, 'Cannot Insert: The given region already has a supervisor.');

END IF;

END;

/

show errors;

--Trigger to check whether the customer and the restaurants are in the same region

CREATE OR REPLACE TRIGGER fromSameRegion

BEFORE INSERT ON Orders

FOR EACH ROW

DECLARE

l_cRegion Customers.cRegion%type;

l_empRegion Employees.empRegion%type;

l_rRegion Restaurants.rRegion%type;

BEGIN

Select cRegion into l_cRegion

From Customers

```

Where cID = :new.cID;

Select empRegion into l_empRegion
From Employees
Where empId = :new.empID;

Select rRegion into l_rRegion
From Restaurants
Where rId = :new.rID;

IF (l_cRegion <> l_empRegion) or (l_cRegion <> l_rRegion) or
(l_empRegion <> l_rRegion) THEN
    RAISE_APPLICATION_ERROR(-20001, 'Cannot Insert: Customer,
driver, and/or restaurant are not from the same region.');
```

END IF;

END;

/

show errors;

-- insert values

--procedure to insert employee information

CREATE OR REPLACE PROCEDURE loadEmployeeData

AS

BEGIN

insert into Employees values('e'||emp_seq.nextval, 'Olivia Liu',
'San Jose', '4082345679', 8); -- 1 SUPERVISOR

insert into Employees values('e'||emp_seq.nextval, 'Adam Jones',
'San Jose', '4082225678', 4); -- 2 DRIVER

insert into Employees values('e'||emp_seq.nextval, 'Maggie
Jones', 'San Jose', '4088458323', 4.5); -- 3 DRIVER

insert into Employees values('e'||emp_seq.nextval, 'Sherene
Bennett', 'San Jose', '4086341212', 5); -- 4 DRIVER

insert into Employees values('e'||emp_seq.nextval, 'Carla
Jepsen', 'San Francisco', '4158948003', 8); -- 5 SUPERVISOR

insert into Employees values('e'||emp_seq.nextval, 'David Kim',
'San Francisco', '4087896478', 6.5); -- 6 DRIVER

insert into Employees values('e'||emp_seq.nextval, 'Luis
Garcia', 'Fremont' , '5104328006', 8); -- 7 SUPERVISOR

insert into Employees values('e'||emp_seq.nextval, 'Jess
Applegate', 'Fremont' , '5104328006', 5); -- 8 DRIVER


```

insert into Employees values('e'||emp_seq.nextval, 'Doug White',
'Santa Clara', '4083345432', 8); -- 9 SUPERVISOR
insert into Employees values('e'||emp_seq.nextval, 'Ira Kassen',
'Santa Clara', '4082553399', 4.5); -- 10 DRIVER

```

```

insert into Employees values('e'||emp_seq.nextval, 'Luna
Granger', 'Palo Alto', '4085653432', 8); -- 11 SUPERVISOR
insert into Employees values('e'||emp_seq.nextval, 'Ron Potter',
'Palo Alto', '6152553389', 4.5); -- 12 DRIVER

```

-- Supervisors

```

insert into Supervisors values ('e1', 85000,'e1admin'); -- SJ
insert into Supervisors values ('e5', 80000,'e5admin'); -- SF
insert into Supervisors values ('e7', 80000,'e7admin'); --
Fremont
insert into Supervisors values ('e9', 80000,'e9admin'); -- Santa
Clara
insert into Supervisors values ('e11', 80000,'e11admin'); --
Palo Alto

```

--Drivers

```

insert into Drivers values ('e2', 'F5628921', 13.50, 1,'e1');
insert into Drivers values ('e3', 'F8214211', 13.50, 1,'e1');
insert into Drivers values ('e4', 'F7757382', 14.00, 1,'e1');
insert into Drivers values ('e6', 'F3226881', 13.00, 1,'e5');
insert into Drivers values ('e8', 'F3456881', 13.00, 1,'e7');
insert into Drivers values ('e10', 'F3676881', 13.00, 1,'e9');
insert into Drivers values ('e12', 'F8921811', 13.00, 1,'e11');

```

END;

/

show errors;

exec loadEmployeeData;

--procedure to insert restraurant data

CREATE OR REPLACE PROCEDURE loadRestaurantData

AS

BEGIN

```

insert into Restaurants values ('r'||rest_seq.nextval, 'Via Mia
Pizza',1, 'Italian', '4082444100', '1150 Saratoga Ave 95129', 'San
Jose'); -- 1

```

```

insert into Restaurants values ('r'||rest_seq.nextval, 'La
Victoria', 1,'Mexican', '4082985335', '140 E San Carlos St 95112',
'San Jose'); -- 2

```

```

insert into Restaurants values ('r'||rest_seq.nextval, 'La
Victoria', 1,'Mexican', '4089938230', '131 W Santa Clara St
95113', 'San Jose'); -- 3

```

```

insert into Restaurants values ('r'||rest_seq.nextval, 'Athena
Grill', 1, 'Italian', '4085679144', '1505 Space Park Dr 95054',
'Santa Clara'); -- 4
insert into Restaurants values ('r'||rest_seq.nextval, 'Pizza My
Heart', 1, 'American', '4082410000', '700 Bellomy St 95050', 'Santa
Clara'); -- 5
insert into Restaurants values ('r'||rest_seq.nextval, 'Eastern
Winds', 1, 'Chinese', '5102261588', '6997 Warm Springs Blvd 94539',
'Fremont'); -- 6
insert into Restaurants values ('r'||rest_seq.nextval, 'Chaat
Bhavan', 1, 'Indian', '5107951100', '5355 Mowry Ave 94538',
'Fremont'); -- 7
insert into Restaurants values ('r'||rest_seq.nextval, 'Osaka
Sushi', 1, 'Japanese', '4152558828', '460 Castro St 95114', 'San
Francisco'); -- 8
insert into Restaurants values ('r'||rest_seq.nextval,
'Farmhouse Kitchen', 1, 'Thai', '4158142920', '710 Florida St
94110', 'San Francisco'); -- 9
insert into Restaurants values ('r'||rest_seq.nextval, 'Oren''s
Hummus Shop', 1, 'Israeli', '6507526492', '261 University Ave
94301', 'Palo Alto');--10
insert into Restaurants values ('r'||rest_seq.nextval, 'Tamarine
Restaurant', 1, 'Vietnamese', '6503258500', '560 University Ave',
'Palo Alto');
END;
/
show errors;

```

```

exec loadRestaurantData;

```

--procedure to insert Menu Items

```

-- MenuItems (restaurantId, itemName, itemPrice)
CREATE OR REPLACE PROCEDURE loadMenuItems
AS
BEGIN
insert into MenuItems values ('r1', 'Italiana Pizza', 7.99);
insert into MenuItems values ('r1', 'Stromboli Pizza', 8.99);
insert into MenuItems values ('r1', 'Siciliana Beef Pizza',
8.99);
insert into MenuItems values ('r1', 'Traditional Calzone',
6.99);

insert into MenuItems values ('r2', 'Regular Burrito', 5.45);
insert into MenuItems values ('r2', 'Super Burrito', 6.45);
insert into MenuItems values ('r2', 'Veggie Burrito', 5.45);

```

```

insert into MenuItems values ('r3', 'Regular Burrito', 5.45);
insert into MenuItems values ('r3', 'Super Burrito', 6.45);
insert into MenuItems values ('r3', 'Veggie Burrito', 5.45);

insert into MenuItems values ('r5', 'Big Sur - 12 in', 17.00);
insert into MenuItems values ('r5', 'Davenport - 12 in', 17.00);
insert into MenuItems values ('r5', 'Maui Wowie - 12 in',
15.00);

insert into MenuItems values ('r4', 'Gyro Plate', 14.50);
insert into MenuItems values ('r4', 'Chicken Souvlaki Plate',
14.50);
insert into MenuItems values ('r4', 'Beef Souvlaki Plate',
18.95);

insert into MenuItems values ('r6', 'Orange Chicken', 8.95);
insert into MenuItems values ('r6', 'Chicken Chow Mein', 7.95);

insert into MenuItems values ('r7', 'Samosa', 3.99);
insert into MenuItems values ('r7', 'Dabeli', 5.49);
insert into MenuItems values ('r7', 'Misal Pav', 6.99);

insert into MenuItems values ('r8', 'Chicken Teriyaki Plate',
14.95);
insert into MenuItems values ('r8', 'Salmon Teriyaki Plate',
16.95);
insert into MenuItems values ('r8', 'Agedashi Tofu', 5.95);
insert into MenuItems values ('r8', 'Assorted Tempura', 6.95);

insert into MenuItems values ('r9', 'Fish Cake', 9.00);
insert into MenuItems values ('r9', 'Tom Yum Kai', 6.00);
insert into MenuItems values ('r9', 'Pad Thai', 14.00);

insert into MenuItems values ('r10', 'Hummus Classic', 8.95);
insert into MenuItems values ('r10', 'Falafel Sandwich', 9.95);

insert into MenuItems values ('r11', 'Banh Mi Roti', 10.00);
insert into MenuItems values ('r11', 'Shrimp Spring Rolls',
12.00);
insert into MenuItems values ('r11', 'Wok Beef Noodles', 18.00);

END;
/
show errors;
exec loadMenuItems;

```

--procedure to update ratings(driver)

```
CREATE OR REPLACE PROCEDURE updateRatings
(Rest_id IN Restaurants.rId%TYPE, Dr_id IN Drivers.empId%TYPE, nrr
IN float, ndr IN float) IS
    oldrr float;
    olddr float;
BEGIN
    select rrating into oldrr from Restaurants where rid=Rest_id;
    select drating into olddr from Drivers where empId=Dr_id;
    update Restaurants set rrating=(oldrr+nrr)/2 where
rid=Rest_id;
    update Drivers set drating=(olddr+ndr)/2 where empId=Dr_id;
END;
commit;
```

--procedure to delete driver info from the table

```
Create or Replace Procedure delete_driver(eid in
drivers.empid%TYPE)
AS
BEGIN
delete from drivers
where empid = eid;
delete from employees
where empid = eid;
END;
/
show errors;
```

--procedure to delete supervisor information

```
Create or Replace Procedure delete_supervisor(eid in
drivers.empid%TYPE)
AS
BEGIN
update drivers set supervisorId = null where supervisorId = eid;
delete from supervisors
where empid = eid;
delete from employees
where empid = eid;
END;
/
```

-- insert values

--procedures to insert new supervisors and drivers

```

CREATE OR REPLACE PROCEDURE add_supervisor(eid in
employees.empid%TYPE,name in employees.empName%TYPE, region in
employees.empRegion%TYPE, phone_no in employees.empPhone%TYPE,
hours_worked in employees.empHours_worked%TYPE, sup_salary in
supervisors.salary%TYPE,pwd in supervisors.password%TYPE)
AS
e_id employees.empid%TYPE;
CURSOR c1 IS SELECT empid from employees where empid = eid;
BEGIN
    open c1;
    FETCH c1 INTO e_id;
    if c1%found then
        insert into Supervisors values (eid, sup_salary,pwd);
    else
        insert into Employees values(eid, name, region,
phone_no, hours_worked);
        insert into Supervisors values (eid, sup_salary,pwd);
    end if;
    update drivers set supervisorId = eid where supervisorId =
null;
END;
/
-- insert values
CREATE OR REPLACE PROCEDURE add_driver(eid in
employees.empid%TYPE,name in employees.empName%TYPE, region in
employees.empRegion%TYPE, phone_no in employees.empPhone%TYPE,
hours_worked in employees.empHours_worked%TYPE, license_no in
drivers.licenseNo%TYPE, driver_wage in drivers.wage%TYPE,
driver_rating in drivers.drating%TYPE, supervisor_id in
drivers.supervisorId%TYPE)
AS
e_id employees.empid%TYPE;
CURSOR c1 IS SELECT empid from employees where empid = eid;
BEGIN
    open c1;
    FETCH c1 INTO e_id;
    if c1%found then
        insert into Drivers values (eid,
license_no,driver_wage,driver_rating,supervisor_id);
    else
        insert into Employees values(eid, name, region,
phone_no, hours_worked);
        insert into Drivers values (eid,
license_no,driver_wage,driver_rating,supervisor_id);
    end if;
END;

```

Main Connection Code Used in all Operations:

```
import java.sql.*;           // Import java's SQL library

Connection conn = NULL;      // Declare connection variable

Class.forName("oracle.jdbc.driver.OracleDriver");

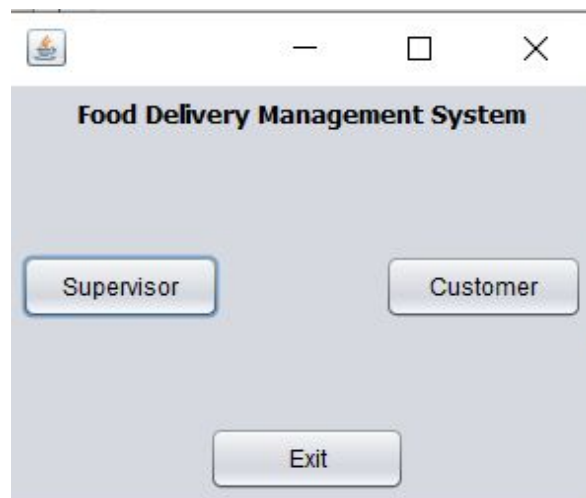
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521
:orcl1", "user", "password"); // Establishing
connection

conn.close()    // Closing the connection after operation is
complete

/* Note: This code should always be kept in a try{}catch(){} block
*/
```

Screenshots of the working and the GUI code:

Home Screen:



Home screen has three buttons:

1) Supervisor

Takes to supervisor login

Code:

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

```
}
```

2)Customer

Takes to customer login

Code:

```
private void  
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    new CustomerLogin().setVisible(true);  
    dispose();  
}
```

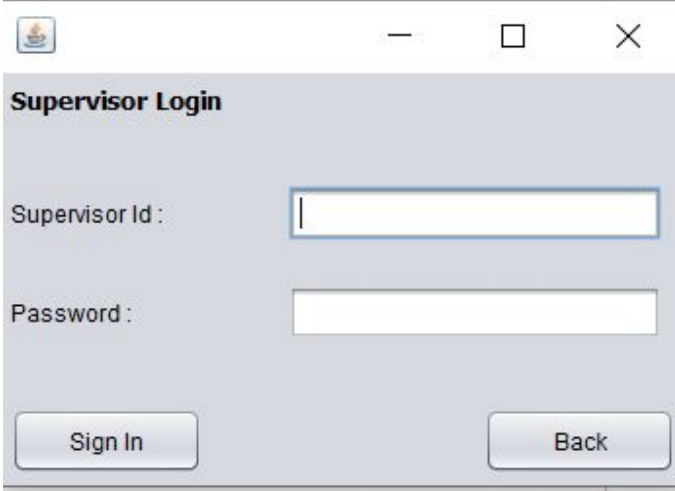
3)Exit

Closes the application:

Code:

```
private void  
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

Supervisor login screen:



Supervisor login has two buttons:

1)Sign In

Helps in signing in

Code:

```
private void  
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{  
        Class.forName("oracle.jdbc.driver.OracleDriver");  
        Connection  
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521  
:orcl1","scott","a1028");  
        String sql ="select * from Supervisors where empid=?";
```

```

        pst=conn.prepareStatement(sql);
        pst.setString(1,sid.getText());
        rs=pst.executeQuery();
        if(rs.next()){
            String uid = rs.getString(1);
            String pass = rs.getString(3);
            if (uid.equals(sid.getText()) &&
pass.equals(jPasswordField1.getText())){
                new supervisor().setVisible(true);
                dispose();
            }
            else{
                JOptionPane.showMessageDialog(null,"Incorrect
Id or password!!");
            }
        }
        conn.close();
        rs.close();
        pst.close();
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null,e);
        e.printStackTrace();
    }
}

```

2) Back

Takes back to the home screen:

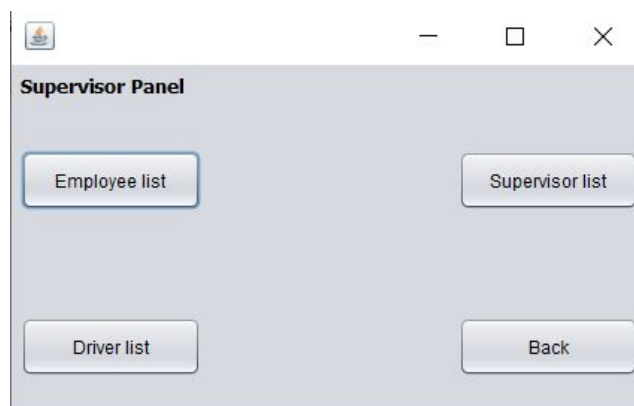
Code:

```

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new HomeScreen().setVisible(true);
    dispose();
}

```

Supervisor Panel screen:



There are four buttons on the screen

1)Employee list

Takes to the employees list table

Code:

```
private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new employee_list().setVisible(true);
    this.dispose();
}
```

2)Supervisor list

Takes to the supervisor list table

Code:

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

3)Drivers list

Takes to the drivers list table

Code:

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

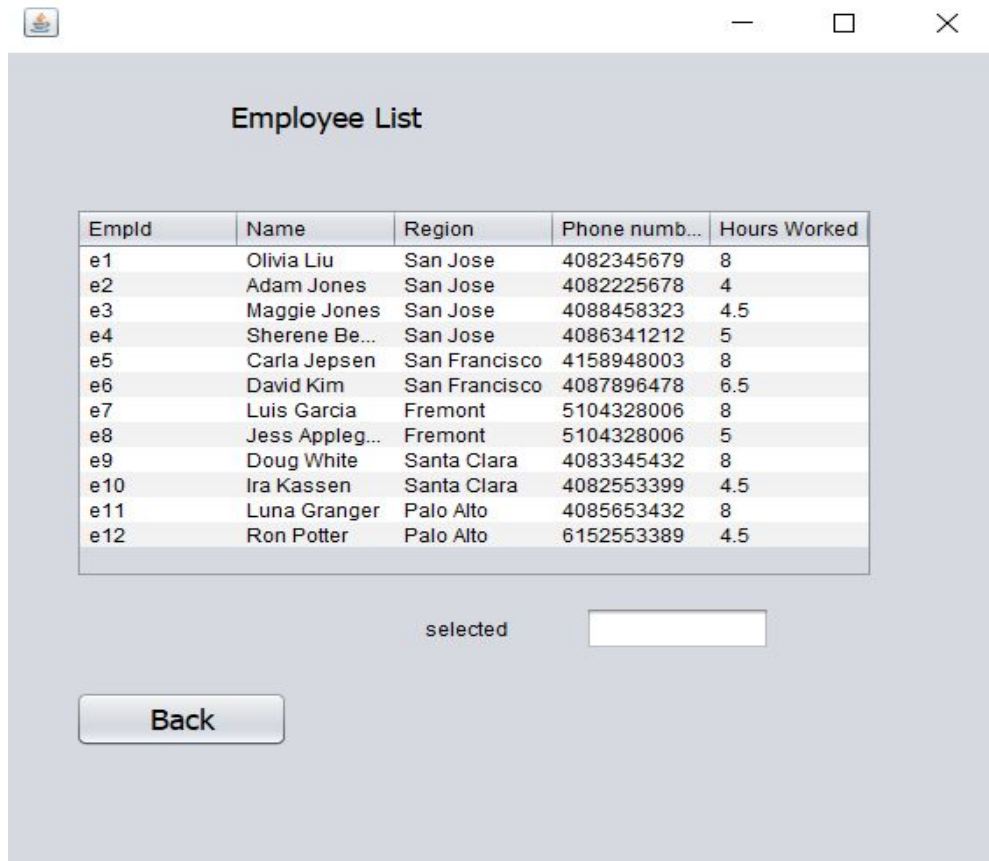
4)Back

Takes back to the supervisor login page

Code:

```
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new Supervisorlogin().setVisible(true);
    dispose();
}
```

Employee list screen:



Empld	Name	Region	Phone numb...	Hours Worked
e1	Olivia Liu	San Jose	4082345679	8
e2	Adam Jones	San Jose	4082225678	4
e3	Maggie Jones	San Jose	4088458323	4.5
e4	Sherene Be...	San Jose	4086341212	5
e5	Carla Jepsen	San Francisco	4158948003	8
e6	David Kim	San Francisco	4087896478	6.5
e7	Luis Garcia	Fremont	5104328006	8
e8	Jess Appleg...	Fremont	5104328006	5
e9	Doug White	Santa Clara	4083345432	8
e10	Ira Kassen	Santa Clara	4082553399	4.5
e11	Luna Granger	Palo Alto	4085653432	8
e12	Ron Potter	Palo Alto	6152553389	4.5

selected

Back

Code to populate the table

```
private void populate_table() {
    DefaultTableModel
model=(DefaultTableModel) jTable1.getModel();
    try{
        Class.forName("oracle.jdbc.OracleDriver");
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:
orcl2","scott","123");
        while(model.getRowCount()>0)
            model.removeRow(0);
        String query = "select * from Employees";
        PreparedStatement st = con.prepareStatement(query);
        //st.setString(1, category);
        ResultSet rs = st.executeQuery();
        while(rs.next()) {
            String s1=rs.getString("empId");
            String s2=rs.getString("empName");
            String s3=rs.getString("empRegion");
            String s4=rs.getString("empPhone");
            String s5=rs.getString("empHOURS_worked");
            model.addRow(new Object[] {s1,s2,s3,s4,s5});
        }
    }
```

```

    }

    }
    catch(Exception e){

        JOptionPane.showMessageDialog(this,e.getMessage());
    }
}

```

Code to make selection in the table:

```

private void jTable1MouseClicked(java.awt.event.MouseEvent
evt) {
    // TODO add your handling code here:
    int index = jTable1.getSelectedRow();
    eid = jTable1.getModel().getValueAt(index,0).toString();

    jTextField1.setText(eid);
}

```

Only one button is present

1)back

Takes back to the supervisor panel

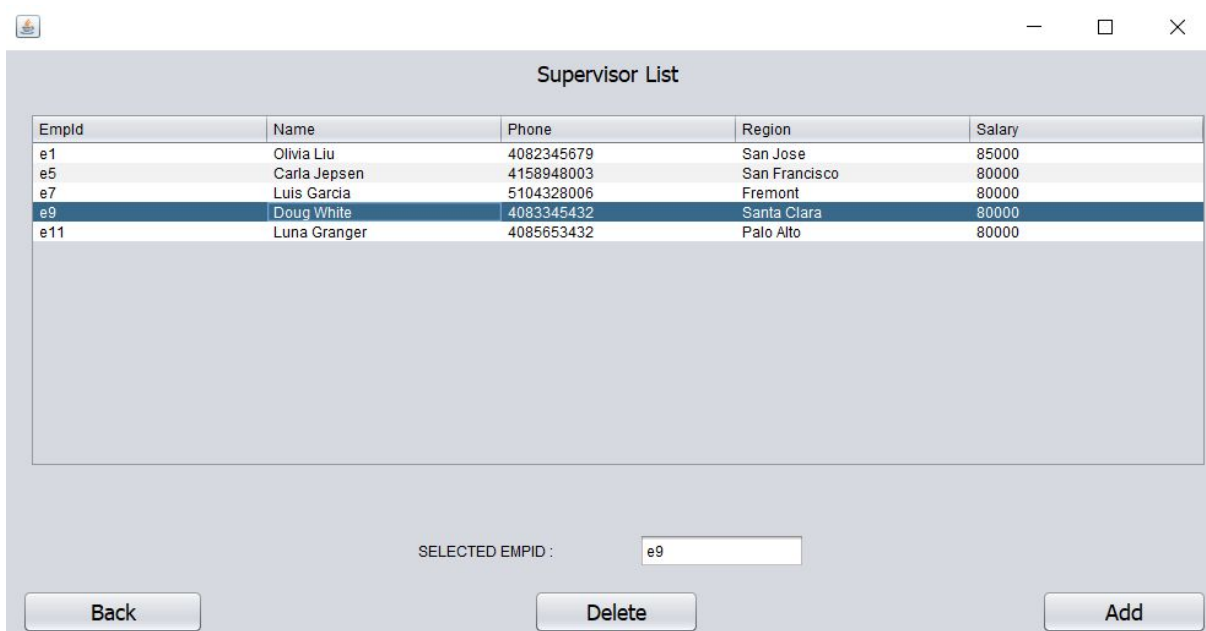
Code:

```

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new supervisor().setVisible(true);
    this.dispose();
}

```

Supervisor list screen:



EmpId	Name	Phone	Region	Salary
e1	Olivia Liu	4082345679	San Jose	85000
e5	Carla Jepsen	4158948003	San Francisco	80000
e7	Luis Garcia	5104328006	Fremont	80000
e9	Doug White	4083345432	Santa Clara	80000
e11	Luna Granger	4085653432	Palo Alto	80000

SELECTED EMPID :

Back Delete Add

Code to populate the table

```

private void populate_table() {
    DefaultTableModel
model=(DefaultTableModel) jTable1.getModel();
    try{
        Class.forName("oracle.jdbc.OracleDriver");
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:
orcl2","scott","123");
        while(model.getRowCount()>0)
            model.removeRow(0);
        String query = "select * from Supervisors,Employees
where Supervisors.empId = Employees.empId";
        PreparedStatement st = con.prepareStatement(query);
        //st.setString(1, category);
        ResultSet rs = st.executeQuery();
        while(rs.next()){
            String s1=rs.getString("empId");
            String s2=rs.getString("salary");
            String s3=rs.getString("empName");
            String s4=rs.getString("empPhone");
            String s5=rs.getString("empRegion");

            model.addRow(new Object[] {s1,s3,s4,s5,s2});
        }

    }
    catch(Exception e){

        JOptionPane.showMessageDialog(this,e.getMessage());
    }
}

```

Code to make selection in the table:

```

private void jTable1MouseClicked(java.awt.event.MouseEvent
evt) {
    // TODO add your handling code here:
        int index = jTable1.getSelectedRow();
        eid = jTable1.getModel().getValueAt(index,0).toString();

        jTextField1.setText(eid);
    }
}

```

Three buttons are present:

1)Delete

Helps in deleting supervisor info from the table

Code:

```

private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

```

```

        // TODO add your handling code here:
        System.out.println(eid);
        if (eid == null ){
            JOptionPane.showMessageDialog(this,"ROW not selected
\n please select a row");
        }
        else{
            try{

                Class.forName("oracle.jdbc.OracleDriver");
                Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521
:orcl2","scott","123");
                //int temp = Integer.parseInt(eid);
                CallableStatement cst=conn.prepareCall("{call
delete_supervisor(?)}");
                cst.setString(1,eid);
                cst.execute();
                JOptionPane.showMessageDialog(null,"Record deleted
successfully");
                populate_table();
            }
            catch(Exception e)
            { JOptionPane.showMessageDialog(this,e.getMessage());}
        }
    }
}

```

2)Add

Helps in adding supervisor info to the table

Code:

```

        private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling code here:
            new add_employee('s').setVisible(true);
            this.dispose();
        }
    }

```

3) Back

Takes back to the supervisor screen

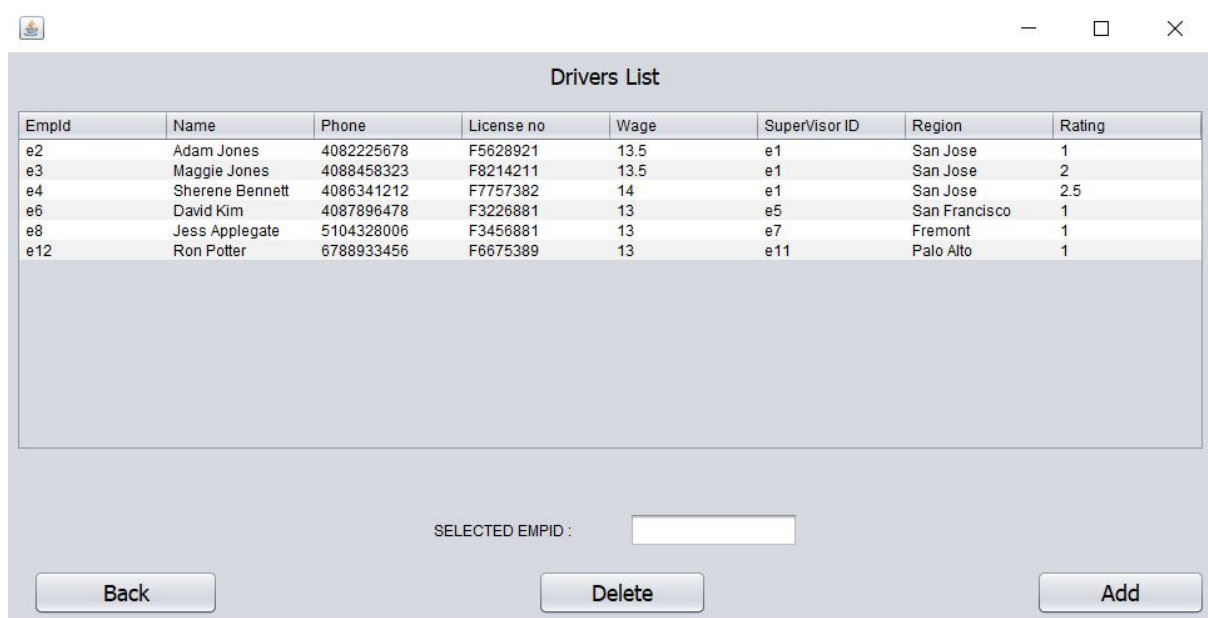
Code:

```

        private void jButton2ActionPerformed(java.awt.event.ActionEvent
evt) {
            // TODO add your handling code here:
            new supervisor().setVisible(true);
            this.dispose();
        }
    }

```

Driver list screen:



EmpId	Name	Phone	License no	Wage	SuperVisor ID	Region	Rating
e2	Adam Jones	4082225678	F5628921	13.5	e1	San Jose	1
e3	Maggie Jones	4088458323	F8214211	13.5	e1	San Jose	2
e4	Sherene Bennett	4086341212	F7757382	14	e1	San Jose	2.5
e6	David Kim	4087896478	F3226881	13	e5	San Francisco	1
e8	Jess Applegate	5104328006	F3456881	13	e7	Fremont	1
e12	Ron Potter	6788933456	F6675389	13	e11	Palo Alto	1

SELECTED EMPID :

Code to populate the table

```
private void populate_table(){

    DefaultTableModel
model=(DefaultTableModel) jTable1.getModel();

    try{
        Class.forName("oracle.jdbc.OracleDriver");
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:
orcl2","scott","123");
        while(model.getRowCount()>0)
            model.removeRow(0);
        String query = "select * from Drivers,Employees where
Drivers.empId = Employees.empId";
        PreparedStatement st = con.prepareStatement(query);
        //st.setString(1,category);
        ResultSet rs = st.executeQuery();
        while(rs.next()){
            String s1=rs.getString("empId");
            String s2=rs.getString("licenseNo");
            String s3=rs.getString("wage");
            String s4=rs.getString("supervisorId");
            String s5=rs.getString("drating");
            String s6=rs.getString("empName");
            String s7=rs.getString("empRegion");
            String s8=rs.getString("empPhone");
            model.addRow(new Object[]
{s1,s6,s8,s2,s3,s4,s7,s5});
        }
    }
```

```

    }
    catch(Exception e){

        JOptionPane.showMessageDialog(this,e.getMessage());
    }
}

```

Code to make selection in the table:

```

private void jTable1MouseClicked(java.awt.event.MouseEvent
evt) {
    // TODO add your handling code here:
    int index = jTable1.getSelectedRow();
    eid = jTable1.getModel().getValueAt(index,0).toString();

    jTextField1.setText(eid);
}

```

Three buttons are present:

1)Delete

Helps in deleting Driver info from the table

Code:

```

private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    System.out.println(eid);
    if (eid == null ){
        JOptionPane.showMessageDialog(this,"ROW not selected
\n please select a row");
    }
    else{
        try{

            Class.forName("oracle.jdbc.OracleDriver");
            Connection conn =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl
2","scott","123");
            //int temp = Integer.parseInt(eid);
            CallableStatement cst=conn.prepareCall("{call
delete_driver(?)}");
            cst.setString(1,eid);
            cst.execute();
            JOptionPane.showMessageDialog(null,"Record deleted
successfully");
            populate_table();
        }
        catch(Exception e)
        { JOptionPane.showMessageDialog(this,e.getMessage());}
    }
}

```

```

    }
}

```

2)Add

Helps in adding driver info to the table

Code:

```

private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new add_employee('d').setVisible(true);
    this.dispose();
}

```

3) Back

Takes back to the supervisor panel:

Code:

```

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new supervisor().setVisible(true);
    this.dispose();
}

```

Add Employee Screen:

Add Employee

EID: Region:

Name: Hours Worked:

Phone no:

Choose One: ☐ Driver ☐ Supervisor

Driver

License No:

Wage:

Supervisor Id:

Supervisor

Salary:

Password:

Code For the two Radio buttons:

```

private void
jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    choice = "driver";
    jRadioButton1.setSelected(true);
}

```



```

        jButton2.setSelected(false);
    }

    private void
jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    choice = "supervisor";
    jRadioButton1.setSelected(false);
    jRadioButton2.setSelected(true);
}

```

1) Back Button:

Takes back to the screen from where it is called.(Driver_List or Supervisor_list)

Code:

```

    private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (typ == 's'){
        new supervisor_list().setVisible(true);
        this.dispose();
    }
    else if (typ == 'd'){
        new drivers_list().setVisible(true);
        this.dispose();
    }
}

```

2) Add Button

Adds the selected type of employee into the database.

Code:

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:

    String name=Name.getText();
    String region= Region.getText();
    String phone_no=pno.getText();
    String hours_worked=hrs.getText();
    String eid=eId.getText();
    //String query = "insert into books value
    (? , ? , ? , ? , ? , ? , ? , ? ) ";
    if(name.isEmpty()||region.isEmpty()|| phone_no.isEmpty()||
hours_worked.isEmpty()){
        JOptionPane.showMessageDialog(this,"ERROR:All Fields
Are Mandatory");
    }
    else{
        if (choice.equals("supervisor")){

```

```

        String Salary=salary.getText();
        String pwd = pass.getText();
        if(Salary.isEmpty() || pwd.isEmpty())
            JOptionPane.showMessageDialog(this,"ERROR:All
Fields Are Mandatory");
        else{
            try{
                Class.forName("oracle.jdbc.OracleDriver");
                Connection conn =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl
1","scott","a1028");

                //int temp = Integer.parseInt(eid);
                CallableStatement
cst=conn.prepareCall("{call add_supervisor(?,?,?,?,?,?,?)}");
                cst.setString(1,eid);
                cst.setString(2,name);
                cst.setString(3,region);
                cst.setString(4,phone_no);
                cst.setString(5,hours_worked);
                cst.setString(6,Salary);
                cst.setString(7,pwd);
                cst.execute();

JOptionPane.showMessageDialog(null,"Supervisor added
successfully");

            }
            catch(Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

            }
        }
    }
    else if(choice.equals("driver")){
        String license_no=license.getText();
        String Wage=wage.getText();
        float drating=1;
        String supervisor_ID=sid.getText();
        if(license_no.isEmpty() || Wage.isEmpty() ||
supervisor_ID.isEmpty())
            JOptionPane.showMessageDialog(this,"ERROR:All
Fields Are Mandatory");
        else{
            try{

                Class.forName("oracle.jdbc.OracleDriver");

```


Customer Login

Customer Id :

Password :

There are three buttons:

1)Sign In

Takes to the customer services:

Code:

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    try{
        Class.forName("oracle.jdbc.OracleDriver");
        Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521
:orcl2","scott","123");
        String sql ="select * from Customers where cid=?";
        pst=conn.prepareStatement(sql);
        pst.setString(1,jTextField1.getText());
        rs=pst.executeQuery();
        if(rs.next()){
            String uid = rs.getString(1);
            String pass = rs.getString(6);
            if (uid.equals(jTextField1.getText()) &&
pass.equals(jPasswordField1.getText())){
                new CustomerService().setVisible(true);
                dispose();
            }
            else{
                JOptionPane.showMessageDialog(null,"Incorrect
Id or password!!");
            }
        }
    }
}
```

```

        conn.close();
        rs.close();
        pst.close();
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null,e);
    }
}

```

2)Sign Up

Takes to the customer signup page:

Code:

```

    private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new CustomerSignUp().setVisible(true);
    dispose();
}

```

3)back

Takes back to the home screen

Code:

```

    private void jButton3ActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    new HomeScreen().setVisible(true);
    dispose();
}

```

Customer Signup screen

Create New Account

User Id: Password:

Name: Address:

Mobile No: Region:

There are three buttons:

1)Create

Code:

```

    private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    try{

```

```

        Class.forName("oracle.jdbc.OracleDriver");
        Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521
:orcl2","scott","123");
        String sql = "insert into Customers"

+"(cid,cname,cPhone,cAddress,cRegion,password) "
        +"values (?, ?, ?, ?, ?, ?)";
        pst=conn.prepareStatement(sql);
/* Adding data to the table */
        pst.setString(1,jTextField1.getText());
        pst.setString(2,jTextField2.getText());
        pst.setString(3,jTextField3.getText());
        pst.setString(4,jTextField4.getText());
        pst.setString(5,jTextField5.getText());
        pst.setString(6,jPasswordField1.getText());
        pst.execute();
        JOptionPane.showMessageDialog(null,"Account Created
successfully");
        new CustomerLogin().setVisible(true);
        dispose();
        conn.close();
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null,e);
    }
}

```

2) Back

Code:

```

        private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new CustomerLogin().setVisible(true);
        dispose();
    }

```

3) Clear

Code:

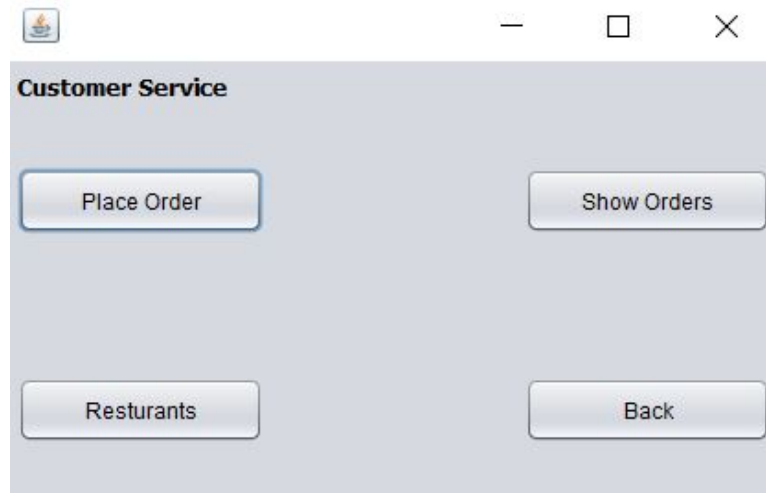
```

        private void jButton2ActionPerformed(java.awt.event.ActionEvent
evt) {
        // TODO add your handling code here:
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");
        jTextField4.setText("");
        jTextField5.setText("");
        jPasswordField1.setText("");
    }

```

}

Customer services Screen:



There are four buttons

1) Place order

Code:

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

2) Show order

Code:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    new ShowOrders().setVisible(true);
    dispose();
}
```

3) Restaurants

Code:

```
private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new ShowRestaurants().setVisible(true);
    dispose();
}
```

4) Back

Code:

```

private void
jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new CustomerLogin().setVisible(true);
    dispose();
}

```

Place order screen:

Place Order

Customer Id :

Choose Restaurant

Items Available :

Name	Price

Order List :

Address :

Total :

Combobox to chose restaurants:

Code:

```

private void
jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521
:orcl1","scott","a1028");
        DefaultTableModel model=(DefaultTableModel)
jTable1.getModel();
        if
(!jComboBox1.getItemAt(jComboBox1.getSelectedIndex()).equals("Choose Restaurant") ||
!jComboBox1.getItemAt(jComboBox1.getSelectedIndex()).equals("")){
            String []arr =
jComboBox1.getItemAt(jComboBox1.getSelectedIndex()).split(" ");
            restid = arr[0];
            model.setRowCount(0);
            String sql ="select * from MenuItems where rid=?";
            pst=conn.prepareStatement(sql);
            pst.setString(1,arr[0]);
            rs=pst.executeQuery();
            while(rs.next()){
                String name=rs.getString(2);
                float price=Float.parseFloat(rs.getString(3));
                model.addRow(new Object[]
                    {name,price});
            }
            conn.close();
            rs.close();
            pst.close();
        }
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null,e);
    }
}

```

There are five buttons:**1) Ok****Code:**

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:

```

```

        try{
            Class.forName("oracle.jdbc.OracleDriver");
            Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521
:orcl2","scott","123");

            String sql ="select * from Customers where cid=?";
            pst=conn.prepareStatement(sql);
            pst.setString(1,jTextField1.getText());
            rs=pst.executeQuery();
            if(rs.next()){
                String add = rs.getString(4);
                region = rs.getString(5);
                jTextArea1.setText(add);
            }
            conn.close();
            rs.close();
            pst.close();
            Class.forName("oracle.jdbc.OracleDriver");

conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521
:orcl2","scott","123");
            sql ="select * from Restaurants where rRegion=?";
            pst=conn.prepareStatement(sql);
            pst.setString(1,region);
            rs=pst.executeQuery();
            while(rs.next()){
                jComboBox1.addItem(rs.getString(1)+"
"+rs.getString(2)+"      (" +Float.parseFloat(rs.getString(3))+")");
            }
            conn.close();
            rs.close();
            pst.close();
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        }
    }
}

```

2) Add

Code:

```

private void AddButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    DefaultTableModel model=(DefaultTableModel)
jTable1.getModel();
}

```

```

int[] index = jTable1.getSelectedRows();
Object[] row = new Object[2];
String str="";
for(int i=0;i< index.length;i++){
    str = str + model.getValueAt(index[i],0);
    str = str + "    " + model.getValueAt(index[i],1);
}
String []arr = str.split("    ");
total += Float.parseFloat(arr[1]);
dlm.addElement(str);
jTextField2.setText(Float.toString(total));
}

```

3) Remove

Code:

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    int Index = orderlist.getSelectedIndex();
    //System.out.println(Index);
    String str;
    if (Index != -1) {
        str = dlm.remove(Index);
        String []arr = str.split("    ");
        total -= Float.parseFloat(arr[1]);
        jTextField2.setText(Float.toString(total));
    }
}

```

4) Place Order

Code:

```

private void jButton3ActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    try{
        Class.forName("oracle.jdbc.OracleDriver");
        Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521
:orcl2","scott","123");
        String Eid="";
        Random rand = new Random();
        int ind = 0;
        String sql = "select * from drivers d,employees e
where d.empid = e.empid and e.empRegion=? and e.empHours_worked <
8";

        pst=conn.prepareStatement(sql);
        pst.setString(1,region);
        rs=pst.executeQuery();
    }
}

```

```

        while(rs.next()){
            Eid += rs.getString(1);
            Eid += " ";
        }
        String []eid=Eid.split(" ");
        ind = rand.nextInt(eid.length);
        rs.close();
        pst.close();
        java.util.Date utilDate = new java.util.Date();
        java.sql.Date date = new
java.sql.Date(utilDate.getTime());
        int oid = rand.nextInt(100000);
        int size = dlm.getSize();
        String ol="";
        for(int i=0;i<size;i++){
            ol += dlm.get(i).split("    ")[0];
            ol += ",";
        }

        ol = ol.substring(0,ol.length()-1);
        sql = "insert into Orders"

+ "(orderId,cid,rid,time_made,total,empId,time_delivered,fooditem) "
        + "values (?,?,?,?,?,?,?,?)";
        pst=conn.prepareStatement(sql);
/* Adding data to the table */
        pst.setString(1,Integer.toString(oid));
        pst.setString(2,jTextField1.getText());
        pst.setString(3,restid);
        pst.setDate(4,date);
        pst.setString(5,jTextField2.getText());
        pst.setString(6,eid[ind]);
        pst.setDate(7,date);
        pst.setString(8,ol);
        pst.execute();
        JOptionPane.showMessageDialog(null,"Order Placed
successfully");
        int response = JOptionPane.showConfirmDialog(this,"Do
you want to rate your
experience?","Rating",JOptionPane.YES_NO_OPTION,JOptionPane.QUESTI
ON_MESSAGE);
        if (response == JOptionPane.YES_OPTION){
            new Rating(restid,eid[ind]).setVisible(true);
            dispose();
        }
        else{


```

5) Back

Code:

```
private void
jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new CustomerService().setVisible(true);
    dispose();
}
```

Show orders screen:

— □ ×

Show Orders

Customer Id :

Search

Order ID	R_ID	Made	Delivered	D_ID	Food Items	Total
----------	------	------	-----------	------	------------	-------

Back

There are two buttons

1) Search

Code:

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

```

Class.forName("oracle.jdbc.OracleDriver");
        Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521
:orcl2","scott","123");
        DefaultTableModel model=(DefaultTableModel)
jTable1.getModel();
        model.setRowCount(0);
        String sql ="select * from Orders where cId=? ";
        pst=conn.prepareStatement(sql);
        pst.setString(1,jTextField1.getText());
        rs=pst.executeQuery();
        while(rs.next()){
            String oid=rs.getString(1);
            String rid=rs.getString(3);
            String mdate=rs.getString(4);
            mdate = mdate.substring(0,10);
            String ddate=rs.getString(7);
            ddate = ddate.substring(0,10);
            String did=rs.getString(6);
            String food=rs.getString(8);
            float total =
Float.parseFloat(rs.getString(5));
            model.addRow(new Object[]
                {oid,rid,mdate,ddate,did,food,total});
        }

        conn.close();
        pst.close();
        rs.close();
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null,e);
    }
}

```

2) Back

Code:

```

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

```

Show Restaurants screen:


```

{rid,name,cuisine,phone,add,region,rating});
    }

    conn.close();
    pst.close();
    rs.close();
}

catch(Exception e){
    JOptionPane.showMessageDialog(null,e);
}

```

2) Back

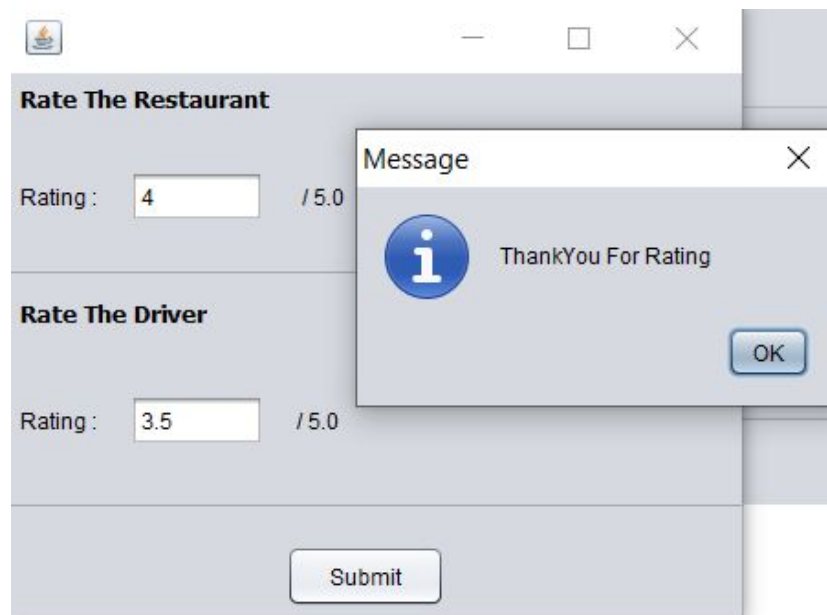
Code:

```

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

```

Rating Screen



Code For Submit Button:

```

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

```



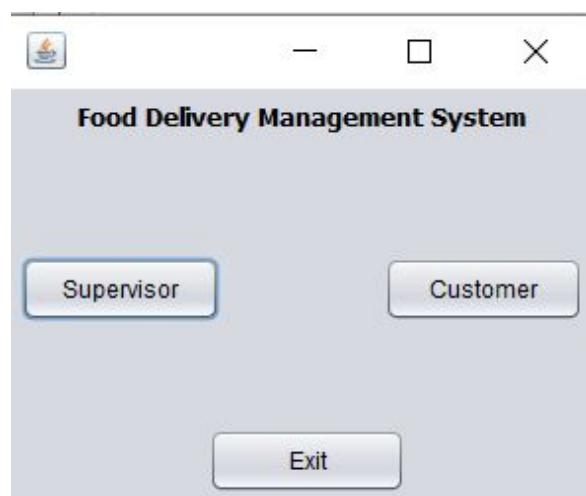
```

        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection
conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521
:orcl1","scott","a1028");
        cst=conn.prepareCall("{call updateRatings(?,?,?,?)}");
/* Execution the procedure calNet_Pay */
        cst.setString(1,R_id);
        cst.setString(2,E_id);
        cst.setFloat(3,Float.parseFloat(R_Rating.getText()));
        cst.setFloat(4,Float.parseFloat(D_Rating.getText()));
        cst.executeUpdate();
        JOptionPane.showMessageDialog(null,"ThankYou For
Rating");
        new CustomerService().setVisible(true);
        dispose();
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null,e);
    }
}

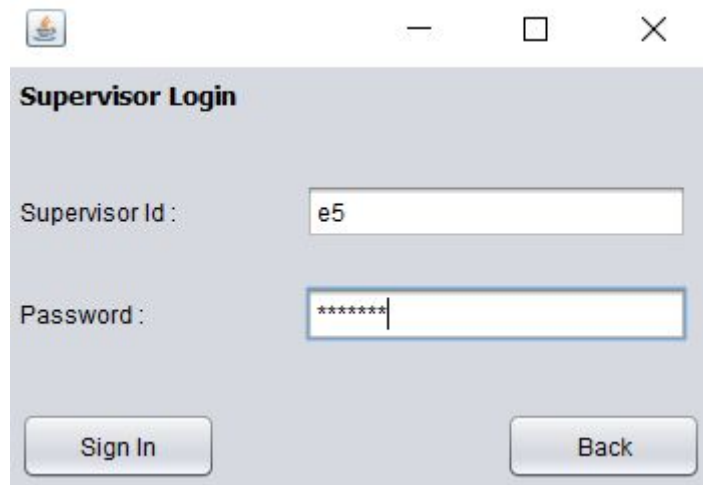
```

Working

1) Home Screen



2) Supervisor Login



A screenshot of a 'Supervisor Login' window. The window has a title bar with a small icon, a minus sign, a maximize button, and a close button. The main area has a light gray background. At the top, the text 'Supervisor Login' is displayed in bold. Below this, there are two input fields. The first is labeled 'Supervisor Id :' and contains the text 'e5'. The second is labeled 'Password :' and contains seven asterisks '*****'. At the bottom, there are two buttons: 'Sign In' on the left and 'Back' on the right.

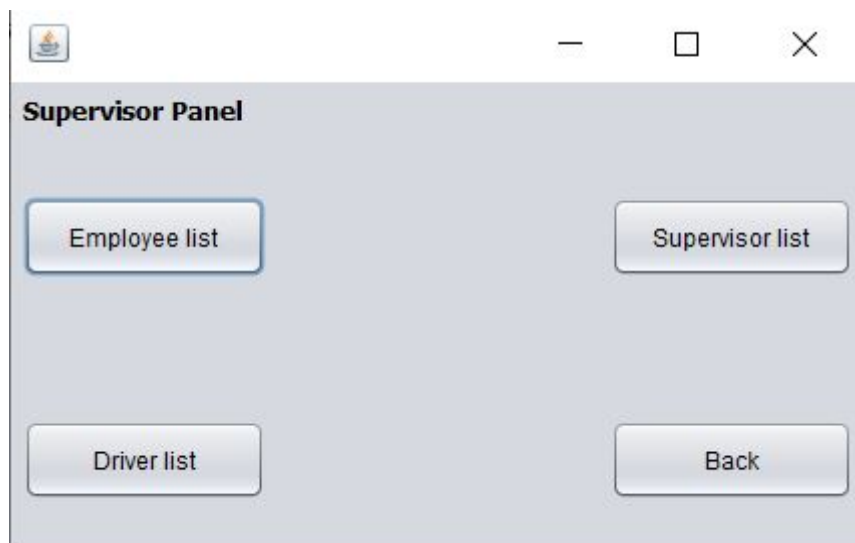
Supervisor Login

Supervisor Id : e5

Password : *****

Sign In Back

3) Supervisor Panel




A screenshot of a 'Supervisor Panel' window. The window has a title bar with a small icon, a minus sign, a maximize button, and a close button. The main area has a light gray background. At the top, the text 'Supervisor Panel' is displayed in bold. Below this, there are four buttons arranged in a 2x2 grid. The top-left button is labeled 'Employee list', the top-right button is labeled 'Supervisor list', the bottom-left button is labeled 'Driver list', and the bottom-right button is labeled 'Back'.

Supervisor Panel

Employee list Supervisor list

Driver list Back

4) Employee List


—
□
×


Employee List

Empld	Name	Region	Phone numb...	Hours Worked
e1	Olivia Liu	San Jose	4082345679	8
e2	Adam Jones	San Jose	4082225678	4
e3	Maggie Jones	San Jose	4088458323	4.5
e4	Sherene Be...	San Jose	4086341212	5
e5	Carla Jepsen	San Francisco	4158948003	8
e6	David Kim	San Francisco	4087896478	6.5
e7	Luis Garcia	Fremont	5104328006	8
e8	Jess Appleg...	Fremont	5104328006	5
e9	Doug White	Santa Clara	4083345432	8
e10	Ira Kassen	Santa Clara	4082553399	4.5
e11	Luna Granger	Palo Alto	4085653432	8
e12	Ron Potter	Palo Alto	6152553389	4.5

selected

Back

5) Driver List


—
□
×

Drivers List

Empld	Name	Phone	License no	Wage	SuperVisor ID	Region	Rating
e2	Adam Jones	4082225678	F5628921	13.5	e1	San Jose	1
e3	Maggie Jones	4088458323	F8214211	13.5	e1	San Jose	2
e4	Sherene Bennett	4086341212	F7757382	14	e1	San Jose	2.5
e6	David Kim	4087896478	F3226881	13	e5	San Francisco	1
e8	Jess Applegate	5104328006	F3456881	13	e7	Fremont	1
e12	Ron Potter	6788933456	F6675389	13	e11	Palo Alto	1

SELECTED EMPID :

Back
Delete
Add

6) Driver Delete Operation

The screenshot shows the 'Drivers List' application window. A modal message box is displayed in the center, indicating that the record for employee e12 has been deleted successfully. The message box contains an information icon and an 'OK' button. Below the message box, the 'SELECTED EMPID' field shows 'e12'. The 'Drivers List' table is visible in the background, with the row for e12 highlighted in blue. At the bottom of the window are three buttons: 'Back', 'Delete', and 'Add'.

EmpId	Name	Phone	License no	Wage	SuperVisor ID	Region	Rating
e2	Adam Jones	4082225678	F5628921	13.5	e1	San Jose	1
e3	Maggie Jones	4088458323	F8214211	13.5	e1	San Jose	2
e4	Sherene Bennett	4086341212	F7757382	14	e1	San Jose	2.5
e6	David Kim	4087896478	F3226881	13	e5	San Francisco	1
e8	Jess Applegate	5104328006	F3456881	13	e7	Fremont	1
e12	Ron Potter	6788933456	F6675389	13	e11	Palo Alto	1

Message

Record deleted successfully

OK

SELECTED EMPID : e12

Back Delete Add

7) Supervisor List


The screenshot shows the 'Supervisor List' application window. The 'Supervisor List' table is displayed, with the row for employee e9 highlighted in blue. Below the table, the 'SELECTED EMPID' field shows 'e9'. At the bottom of the window are three buttons: 'Back', 'Delete', and 'Add'.

EmpId	Name	Phone	Region	Salary
e1	Olivia Liu	4082345679	San Jose	85000
e5	Carla Jepsen	4158948003	San Francisco	80000
e7	Luis Garcia	5104328006	Fremont	80000
e9	Doug White	4083345432	Santa Clara	80000
e11	Luna Granger	4085653432	Palo Alto	80000

SELECTED EMPID : e9

Back Delete Add

8) Supervisor Delete Operation



— □ ×

Supervisor List

EmpId	Name	Phone	Region	Salary
e1	Olivia Liu	4082345679	San Jose	85000
e5	Carla Jepsen	4158948003	San Francisco	80000
e7	Luis Garcia	5104328006	Fremont	80000
e9	Doug White	4083345432	Santa Clara	80000
e11	Luna Granger	6678955432	Palo Alto	80000

Message

×


Record deleted successfully

OK


SELECTED EMPID :

Back

Delete

Add

9) Add Employee


— □ ×

Add Employee

EID :
Region :

Name :
Hours Worked :

Phone no :

Choose One :

Driver

License No :
Wage :
Supervisor Id :

Salary :
Password :

Back

Add

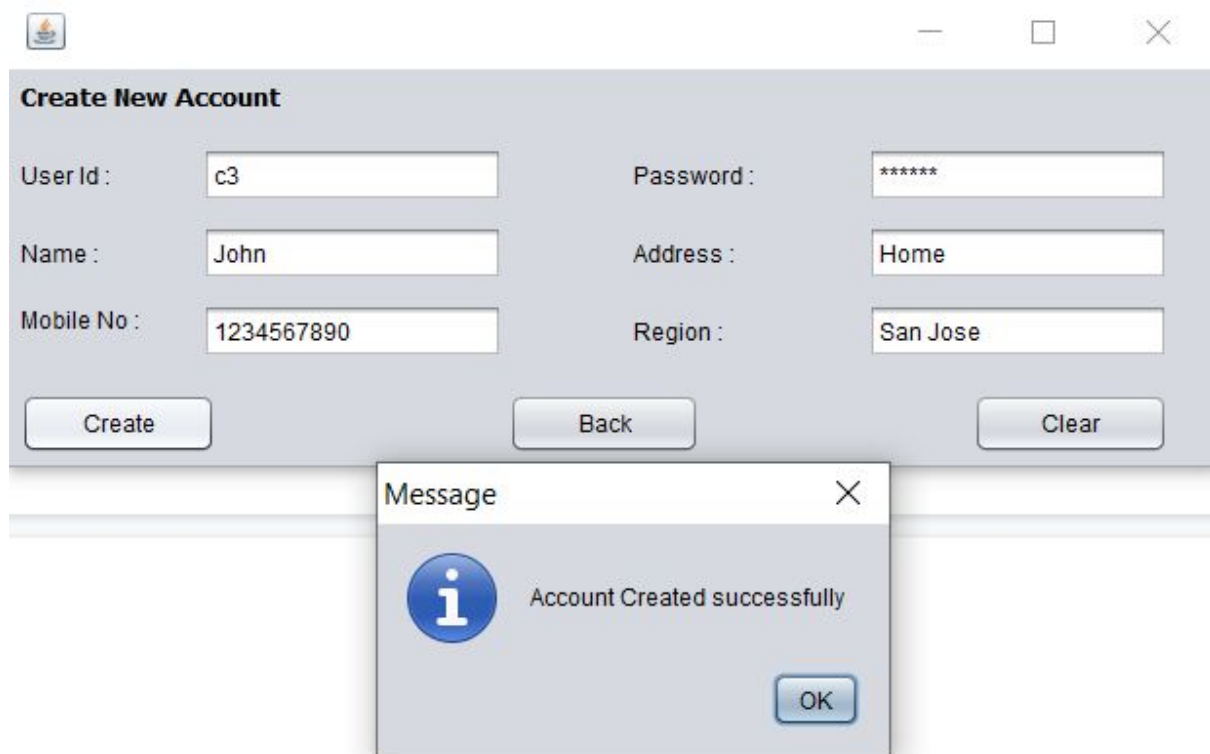
Message

×


Supervisor added successfully

OK

10) Customer Account Creation



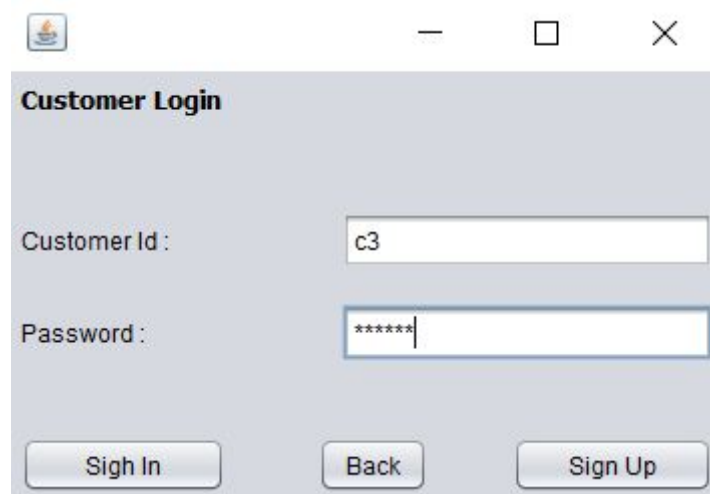
The image shows a 'Create New Account' form with the following fields and values:

Field	Value
User Id :	c3
Password :	*****
Name :	John
Address :	Home
Mobile No :	1234567890
Region :	San Jose

Buttons at the bottom: Create, Back, Clear.

A 'Message' dialog box is overlaid on the form, displaying an information icon and the text: 'Account Created successfully'. An 'OK' button is at the bottom right of the dialog.

11) Customer Login

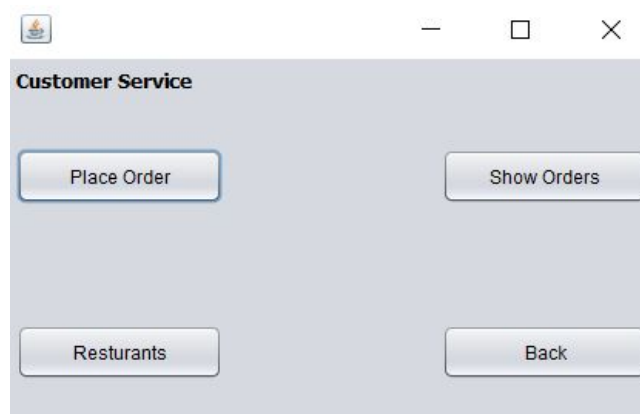


The image shows a 'Customer Login' form with the following fields and values:

Field	Value
Customer Id :	c3
Password :	*****

Buttons at the bottom: Sign In, Back, Sign Up.


12) Customer Service



The image shows a 'Customer Service' form with the following buttons:

Place Order	Show Orders
Restaurants	Back


13) Show Restaurants


— □ ×

Restaurants

Region :

R_ID	Name	Cuisine	Phone	Address	Region	Rating
r1	Via Mia Pizza	Italian	4082444100	1150 Saratoga Ave 9...	San Jose	2.5
r2	La Victoria	Mexican	4082985335	140 E San Carlos St...	San Jose	1.0
r3	La Victoria	Mexican	4089938230	131 W Santa Clara ...	San Jose	2.5



— □ ×

Restaurants

Region :

R_ID	Name	Cuisine	Phone	Address	Region	Rating
r8	Osaka Sushi	Japanese	4152558828	460 Castro St 95114	San Francisco	1.0
r9	Farmhouse Kitchen	Thai	4158142920	710 Florida St 94110	San Francisco	1.0

14) Show Orders


— □ ×

Show Orders

Customer Id :

Order ID	R_ID	Made	Delivered	D_ID	Food Items	Total
36689	r1	2020-05-03	2020-05-03	e2	Italiana Pizza	7.99
2798	r3	2020-05-03	2020-05-03	e4	Regular Burrito	5.45
24432	r1	2020-05-03	2020-05-03	e2	Italiana Pizza,Sici...	32.96

— □ ×

Show Orders

Customer Id : Search

Order ID	R_ID	Made	Delivered	D_ID	Food Items	Total
58927	r2	2020-05-05	2020-05-05	e3	Super Burrito,Veggie Burrito	11.9

Back

15) Place Order

— □ ×

Place Order

Customer Id : OK

La Victoria (1.0) ▼

Items Available :

Name	Price
Regular Burrito	5.45
Super Burrito	6.45
Veggie Burrito	5.45

Add

Order List :

Super Burrito 6.45

Veggie Burrito 5.45

Remove

Address :

Home

Total : Back Place Order

×

Message

Order Placed successfully

OK

Place Order

Customer Id : OK

La Victoria (1.0)

Items Available :

Name	
Regular Burrito	5.45
Super Burrito	6.45
Veggie Burrito	5.45

Order List :

Super Burrito	6.45
Veggie Burrito	5.45

Address :

Total :

16) Rating


Rate The Restaurant

Rating : / 5.0

Rate The Driver

Rating : / 5.0

Message

 ThankYou For Rating