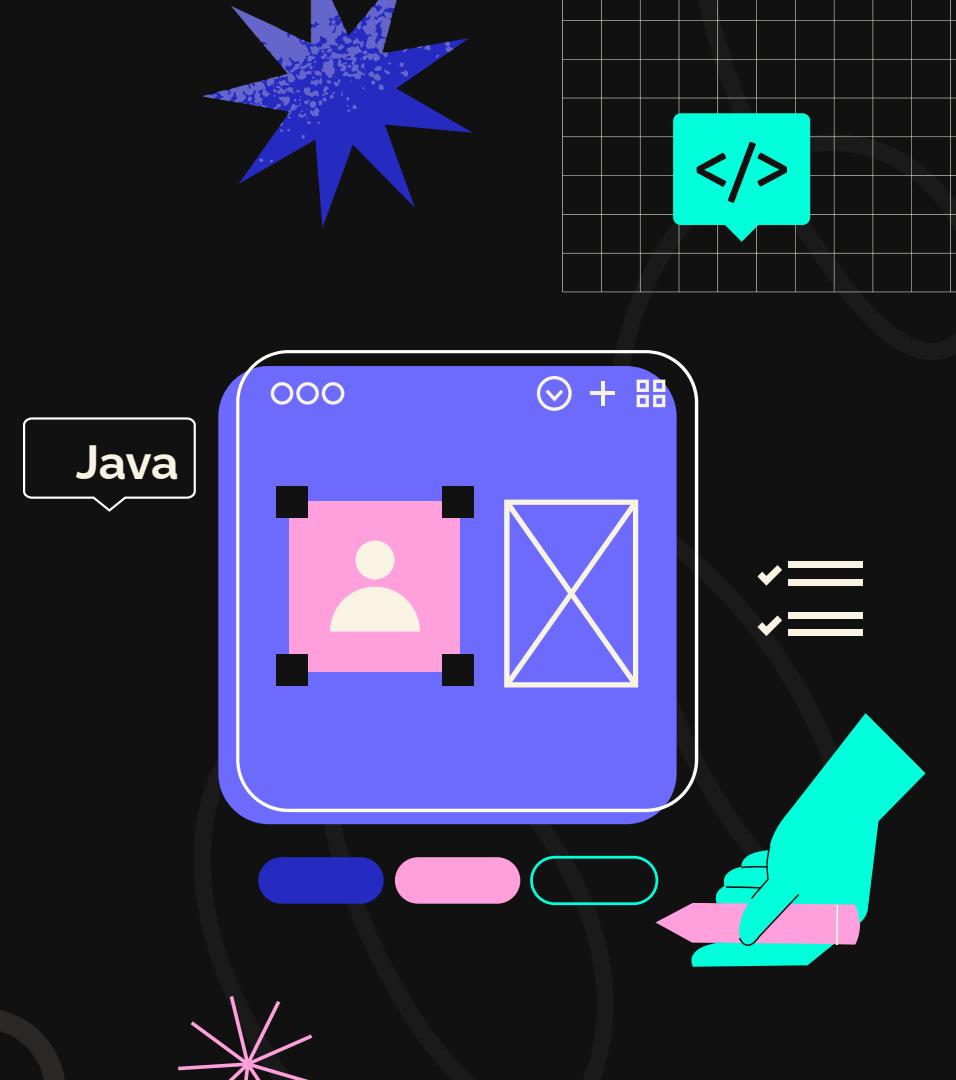


# Pets Shop Interface

LY PHUONG KHAI B2111930  
TRAN MINH QUANG B2111946



# Funcion

01

Function

02

Database

03

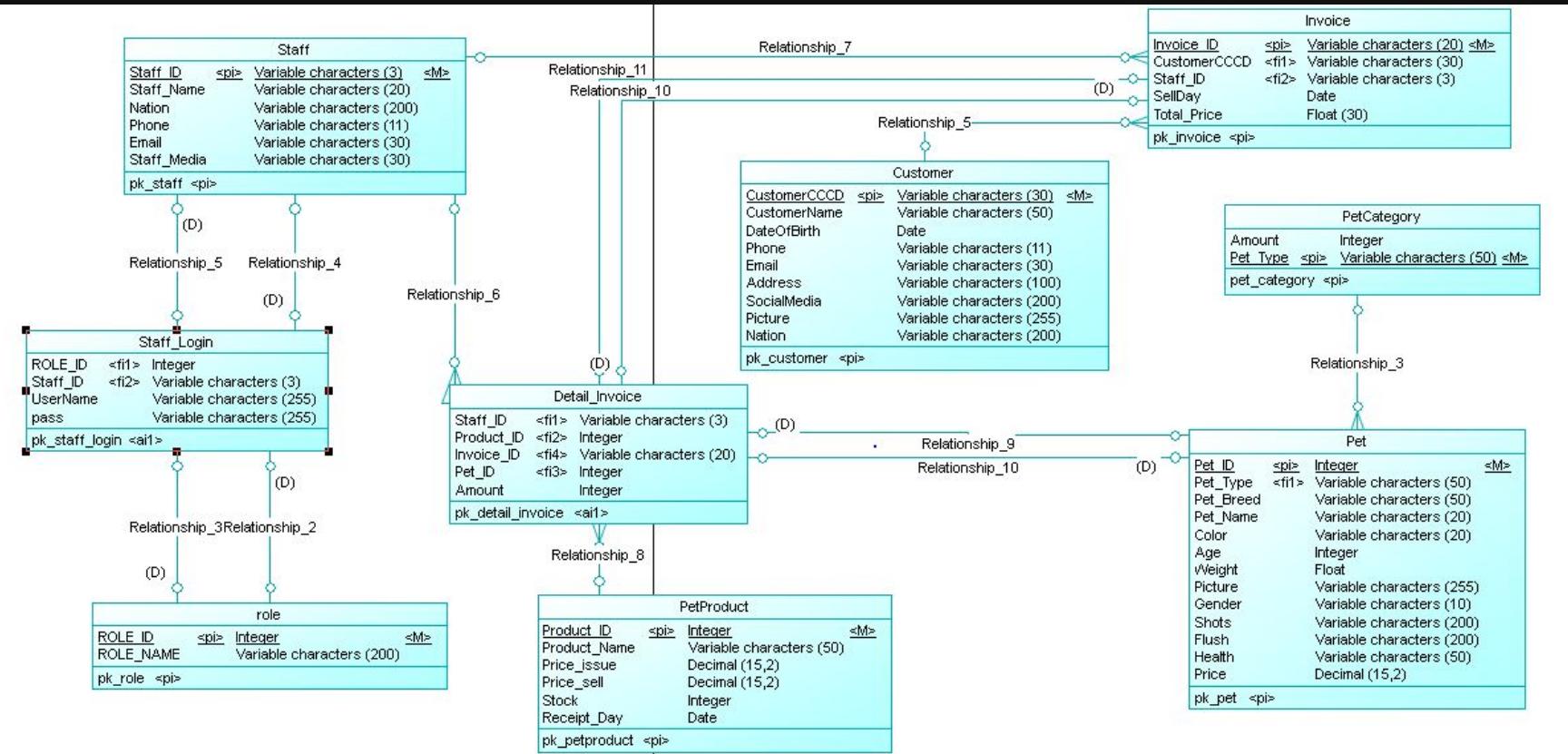
Interfaces



# Function

- Login Logout Staff and Admin.
- Search, Insert, Update and Delete User
- Search, Insert, Update and Delete Staff.
- Search, Insert, Update and Delete Customers.
- Search, Insert, Update and Delete Pet Product.
- Search, Insert, Update and Delete Pet.
- Insert, Update Pet Image.
- Invoicing.
- Print Invoice
- View Invoice
- View Monthly Revenue

# Power Designer DataBase



# Login Staff and Admin

```
private void LoginButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    UsersDAO UserDATA = new UsersDAO();  
    Users user = null;  
    String username = usernametxt.getText();  
    String password = passwordtxt.getText();  
    if(usernametxt.getText().isBlank() || passwordtxt.getText().isBlank()) {  
        JOptionPane.showMessageDialog(parentComponent, "Please enter both username and password");  
    } else {  
        user = UserDATA.Login(username, password);  
        if(user == null) {  
            JOptionPane.showMessageDialog(parentComponent, "Invalid username or password");  
        } else if(user.getRole_id() == 1) {  
            this.dispose();  
            new AdminPage().setVisible(true);  
        } else {  
            this.dispose();  
            new StaffPage().setVisible(true);  
        }  
    }  
}
```

Only Pet

Only Pet

Username

Password



```
private void LogoutActionPerformed(java.awt.event.ActionEvent evt) {  
    this.dispose();  
    new Login().setVisible(true);  
}
```

# Insert User

```
private void ConfirmActionPerformed(java.awt.event.ActionEvent evt) {  
    UsersDAO userDATA = new UsersDAO();  
    if(passwordtxt.getText().isBlank() || passwordcftxt.getText().isBlank() || usernametxt.getText().isBlank()){  
        JOptionPane.showMessageDialog(parentComponent: this, message: "Missing information");  
    }else if(!passwordcftxt.getText().equals(anObject.passwordtxt.getText())){  
        JOptionPane.showMessageDialog(parentComponent: this, message: "Invalid confirm password");  
    }else {  
        String id = String.valueOf(obj: IDBox.getSelectedItem());  
        String username = usernametxt.getText();  
        String password = passwordtxt.getText();  
        int role = Integer.parseInt(s: String.valueOf(obj: RoleBox.getSelectedItem()));  
        Users user = new Users(id,username,password,role_id: role);  
        int result = userDATA.insert(o: user);  
        if(result == 0){  
            JOptionPane.showMessageDialog(parentComponent: this, message: "Add Failed!!");  
        }else  
        {  
            JOptionPane.showMessageDialog(parentComponent: this, message: "Add Successful!!");  
            this.setVisible(b: false);  
            table.setRowCount(rowCount:0);  
            ArrayList<Users> list = userDATA.getAll();  
            addData(list);  
        }  
    }  
}
```

# Update User

```
public int update(Users o) {  
    PreparedStatement st = null;  
    int result = 0; //result == 1 => thanh cong // result == 0 => that bai  
    try{  
        Connection conn = Connect.Conn();  
        st = conn.prepareStatement("UPDATE Staff_Login \n" +  
                "Set username = ?,\n" +  
                "    pass = ?\n" +  
                "Where Staff_id = ?").  
        st.setString(1, string: o.getUsername());  
        st.setString(2, string: o.getPassword());  
        st.setString(3, string: o.getId());  
        result = st.executeUpdate();  
    }catch(Exception e){  
        System.out.println("Cannot Connect" + e);  
    }  
    return result;  
}
```

```
private void UpdateActionPerformed(java.awt.event.ActionEvent evt) {  
    UsersDAO UserDATA = new UsersDAO();  
    int result = 0;  
    for(int i = 0 ; i < table.getRowCount();i++) {  
        if(table.getValueAt(row: i, column: 0).equals(obj: "") ||  
            table.getValueAt(row: i, column: 1).equals(obj: "") ||  
            table.getValueAt(row: i, column: 2).equals(obj: "") ){  
            JOptionPane.showMessageDialog(parentComponent: UserForm.this, message: "Please enter full information");  
            break;  
        }else{  
            Users u = new Users();  
            u.setId((String)table.getValueAt(row: i, column: 0));  
            u.setUsername((String)table.getValueAt(row: i, column: 1));  
            u.setPassword((String)table.getValueAt(row: i, column: 2));  
            result = UserDATA.update(o: u); //result = 0 => that bai , result > 0 => thanh cong  
        }  
    }  
    if(result == 0 ){  
        JOptionPane.showMessageDialog(parentComponent: UserForm.this, message: "Cannot Update");  
    }else {  
        JOptionPane.showMessageDialog(parentComponent: UserForm.this, message: "Updated!!!");  
    }  
    table.setRowCount(rowCount:0);  
    ArrayList<Users> list = UserDATA.getAll();  
    addData(list);  
}
```

# Delete User

```
public int delete(String id) {  
    PreparedStatement stmt = null;  
    int result = 0;//result == 1 => thanh cong // result == 0 => that bai  
    try{  
        Connection conn = Connect.Conn();  
        stmt = conn.prepareStatement(string: "DELETE FROM Staff_Login where staff_id = ?");  
        stmt.setString(i:1,string: id);  
        result = stmt.executeUpdate();  
    }catch(Exception e){  
        System.out.println("Cannot connect: "+ e);  
    }  
    return result;  
}  
  
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    UsersDAO UserDATA = new UsersDAO();  
    int row = UsersTable.getSelectedRow();  
    if(row == -1){  
        JOptionPane.showMessageDialog(parentComponent: UserForm.this, message: "Please choose account to delete!");  
    }else {  
        int cf = JOptionPane.showConfirmDialog(parentComponent: UserForm.this, message: "Are you sure to delete this account?");  
        if(cf == JOptionPane.YES_OPTION) {  
            String id = String.valueOf(obj: table.getValueAt(row, column: 0));  
            UserDATA.delete(id);  
            table.setRowCount(rowCount:0);  
            ArrayList<Users> list = UserDATA.getAll();  
            addData(list);  
        }  
    }  
}
```

# Search User

```
private void SearchtxtKeyReleased(java.awt.event.KeyEvent evt) {  
    UsersDAO UserDATA = new UsersDAO();  
    ArrayList<Users> list = UserDATA.search(condition: "Staff_id", things: Searchtxt.getText());  
    table.setRowCount(rowCount: 0);  
    addData(list);  
    if(Searchtxt.getText().equals(anObject: "")){  
        list = UserDATA.getAll();  
        table.setRowCount(rowCount: 0);  
        addData(list);  
    }  
}  
  
public ArrayList<Users> search(String condition, String things) {  
    ArrayList<Users> list = new ArrayList();  
    Statement stmt = null;  
    try{  
        Connection conn = Connect.Conn();  
        stmt = conn.createStatement();  
        ResultSet rs = stmt.executeQuery("Select * from Staff_Login where " + condition + " LIKE '%" + things + "%'");  
        while(rs.next()){  
            Users u = new Users(  
                id: rs.getString(string: "Staff_id"),  
                username: rs.getString(string: "username"),  
                password: rs.getString(string: "pass"),  
                role_id: rs.getInt(string: "role_id")  
            );  
            list.add(e: u);  
        }  
    }catch(Exception e){  
        System.out.println("Cannot connect: "+ e);  
    }  
    return list;  
}
```

# Insert Staff

```
private void ConfirmStaffActionPerformed(java.awt.event.ActionEvent evt) {  
    StaffDAO staffDATA = new StaffDAO();  
    if( Staffidtext.getText().equals(anObject:"") ||  
        Staffnametxt.getText().equals(anObject:"") ||  
        StaffNationtxt.getText().equals(anObject:"") ||  
        Staffphonetxt.getText().equals(anObject:"") ||  
        Staffmailtxt.getText().equals(anObject:"") ||  
        StaffSocialMediatxt.getText().equals(anObject:"")){  
        JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Missing information");  
    }else if(Staffidtext.getText().length() > 3){  
        JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Invalid ID");  
    }else if(!Staffmailtxt.getText().endsWith(suffix: "@gmail.com")){  
        JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Invalid gmail");  
    }else{  
        String id = Staffidtext.getText();  
        String name = Staffnametxt.getText();  
        String Nation = StaffNationtxt.getText();  
        String phone = Staffphonetxt.getText();  
        String email = Staffmailtxt.getText();  
        String SocialMedia = StaffSocialMediatxt.getText();  
        Staff st = new Staff(id, name, nation, Nation, phone, email, socialmedia: SocialMedia);  
        int result = staffDATA.insert(o: st);  
        if(result == 0){  
            JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Cannot Add Staff");  
        }else{  
            JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Add Staff Successful!!!");  
        }  
    }  
}  
  
private void addStaffActionPerformed(java.awt.event.ActionEvent evt) {  
    AddStaffDialog.setVisible(b: true);  
}
```

# Insert Staff

Add Staff

ID	<input type="text" value="ST4"/>
Name	<input type="text" value="Nguyen Van B"/>
Nation	<input type="text" value="Viet Nam"/>
Phone	<input type="text" value="09876543671"/>
Email	<input type="text" value="nguyenvanb@gmail.com"/>
Social Media	<input type="text" value="https://www.facebook.com/NguyenVanB/"/>

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

Log Out

## Staff List

Search

Staff_id	Staff_name	Nation	phone	email	Staff_Media
AMD	Lee Sang-hyeok	Korea	0999999999	Faker@gmail.com	<a href="https://www.facebook.com/Faker">https://www.facebook.com/Faker</a>
ST1	Jack	Viet Nam	09667756733	jack@gmail.com	<a href="https://www.facebook.com/jack">https://www.facebook.com/jack</a>
ST2	Jane Smith	Canada	0987654321	jane.smith@gmail.com	<a href="https://www.facebook.com/janesmith">https://www.facebook.com/janesmith</a>
ST3	Bray	Viet Nam	09765874911	baothanhthien@gmail.com	<a href="https://www.facebook.com/baothanhthien">https://www.facebook.com/baothanhthien</a>
ST4	Nguyen Van B	Viet Nam	09876543671	nguyenvanb@gmail.com	<a href="https://www.facebook.com/nguyenvanb">https://www.facebook.com/nguyenvanb</a>

Message

 Add Staff Successful!!!

# Update Staff

```
private void UPDATEStaffActionPerformed(java.awt.event.ActionEvent evt) {  
    StaffDAO staffDATA = new StaffDAO();  
    int result = 0;  
    for(int i = 0 ; i < table.getRowCount();i++) {  
        if(table.getValueAt(row: i, column: 0).equals(obj: "") ||  
        table.getValueAt(row: i, column: 1).equals(obj: "") ||  
        table.getValueAt(row: i, column: 2).equals(obj: "") ||  
        table.getValueAt(row: i, column: 3).equals(obj: "") ||  
        table.getValueAt(row: i, column: 4).equals(obj: "") ||  
        table.getValueAt(row: i, column: 5).equals(obj: "")){  
            JOptionPane.showMessageDialog(parentComponent: StaffForm.this,message: "Please enter full information");  
            break;  
        }else{  
            Staff st = new Staff();  
            st.setId((String)table.getValueAt(row: i, column: 0));  
            st.setName((String)table.getValueAt(row: i,column: 1));  
            st.setNation((String)table.getValueAt(row: i,column: 2));  
            st.setPhone((String)table.getValueAt(row: i,column: 3));  
            st.setEmail((String)table.getValueAt(row: i,column: 4));  
            st.setSocialmedia((String)table.getValueAt(row: i,column: 5));  
            result = staffDATA.update(o:st); //result = 0 => that bai , result > 0 => thanh cong  
        }  
    }  
    if(result == 0 ){  
        JOptionPane.showMessageDialog(parentComponent: StaffForm.this,message: "Cannot Update");  
    }else {  
        JOptionPane.showMessageDialog(parentComponent: StaffForm.this,message: "Updated!!!");  
    }  
    table.setRowCount(rowCount:0);  
    ArrayList<Staff> list = staffDATA.getAll();  
    addData(list);  
}
```

# Update Staff

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

Log Out

Staff List

Search Staff\_id

Staff_id	Staff_name	Nation	phone	email	Staff_Media
AMD	Lee Sang-hyeok	Korea	0999999999	Faker@gmail.com	https://www.faceb...
ST1	Jack	Viet Nam	09667756733	jack@gmail.com	https://www.faceb...
ST2	Jane Smith	Canada	0987654321	jane.smith@gmail...	https://www.faceb...
ST3	Bray	Viet Nam	09765874911	baothanhthien@g...	https://www.faceb...
ST4	Nguyen Van B	Canada	09876543671	nguyenvanb@gm...	https://www.faceb...

ADD

UPDATE

DELETE

Message

Updated!!!

OK

# Delete Staff

```
private void DeleteStaffActionPerformed(java.awt.event.ActionEvent evt) {
    StaffDAO StaffDATA = new StaffDAO();
    int row = StaffTable.getSelectedRow();
    if(row == -1){
        JOptionPane.showMessageDialog(parentComponent: StaffForm.this, message: "Please choose staff to delete!");
    }else {
        int cf = JOptionPane.showConfirmDialog(parentComponent: StaffForm.this, message: "Are you sure to delete this staff?");
        if(cf == JOptionPane.YES_OPTION) {
            String id = String.valueOf(obj: table.getValueAt(row, column: 0));
            StaffDATA.delete(id);
            table.setRowCount(rowCount.0);
            ArrayList<Staff> list = StaffDATA.getAll();
            addData(list);
        }
    }
}

public int delete(String id) {
    PreparedStatement stmt = null;
    int result = 0;//result == 1 => thanh cong // result == 0 => that bai
    try{
        Connection conn = Conn();
        stmt = conn.prepareStatement(string: "DELETE FROM Staff where staff_id = ?");
        stmt.setString(i: 1, string: id);
        result = stmt.executeUpdate();
    }catch(Exception e){
        System.out.println("Cannot connect: "+ e);
    }
    return result;
}
```

# Delete Staff

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

Log Out

## Staff List

Search Staff\_id

Staff_id	Staff_name	Nation	phone	email	Staff_Media
AMD	Lee Sang-hyeok	Korea	0999999999	Faker@gmail.com	https://www.facebook.com/faker
ST1	Jack	Viet Nam	09667756733	jack@gmail.com	https://www.facebook.com/jack
ST2	Jane Smith	Canada	0987654321	jane.smith@gmail.com	https://www.facebook.com/janesmith
ST3	Bray	Viet Nam	09765874911	baothanhthien@gmail.com	https://www.facebook.com/baothanhthien
ST4	Nguyen Van B	Canada	09876543671	nguyenvanb@gmail.com	https://www.facebook.com/nguyenvanb

ADD

Select an Option

Are you sure to delete this staff?

Yes No Cancel

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

Log Out

## Staff List

Search Staff\_id

Staff_id	Staff_name	Nation	phone	email	Staff_Media
AMD	Lee Sang-hyeok	Korea	0999999999	Faker@gmail.com	https://www.facebook.com/faker
ST1	Jack	Viet Nam	09667756733	jack@gmail.com	https://www.facebook.com/jack
ST2	Jane Smith	Canada	0987654321	jane.smith@gmail.com	https://www.facebook.com/janesmith
ST3	Bray	Viet Nam	09765874911	baothanhthien@gmail.com	https://www.facebook.com/baothanhthien

ADD

UPDATE

DELETE

# Search Staff

```
public ArrayList<Staff> search(String condition, String things) {
    ArrayList<Staff> list = new ArrayList();
    Statement stmt = null;
    try{
        Connection conn = Connect.Conn();
        stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("Select * from Staff where " + condition +" LIKE '%"+ things +"%'");
        while(rs.next()){
            Staff st = new Staff(
                id: rs.getString(string: "Staff_id"),
                name: rs.getString(string: "Staff_name"),
                nation: rs.getString(string: "Nation"),
                phone: rs.getString(string: "phone"),
                email: rs.getString(string: "email"),
                socialmedia: rs.getString(string: "Staff_Media")
            );
            list.add(e: st);
        }
    }catch(Exception e){
        System.out.println("Cannot connect: "+ e);
    }
    return list;
}

private void searchStafftxtKeyReleased(java.awt.event.KeyEvent evt) {
    StaffDAO StaffDATA = new StaffDAO();
    ArrayList<Staff> list = StaffDATA.search((String)SearchStaffBox.getSelectedItem(),things: searchStafftxt.getText());
    table.setRowCount(rowCount:0);
    addData(list);
    if(searchStafftxt.getText().equals(anObject: "")){
        list = StaffDATA.getAll();
        table.setRowCount(rowCount:0);
        addData(list);
    }
}
```

# Search Staff

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

Log Out

## Staff List

Search Staff\_name ▾ Lee

Staff_id	Staff_name	Nation	phone	email	Staff_Media
AMD	Lee Sang-hyeok	Korea	0999999999	Faker@gmail.com	<a href="https://www.facebook.com/faker">https://www.facebook.com/faker</a>

ADD

UPDATE

DELETE

# Insert Customer

```
private void ConfirmButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    CustomersDAO cusDATA = new CustomersDAO();  
    if(CCCDtxt.getText().isBlank() || Nametxt.getText().isBlank() || Birthdaytxt.getText().isBlank() ||  
        Phonetxt.getText().isBlank() || Emailtxt.getText().isBlank() || Addresstxt.getText().isBlank() ||  
        SocialMediatxt.getText().isBlank() || Nationtxt.getText().isBlank() ||  
        (!GenderButtonGroup.isSelected(m_Button1.getModel()) && !GenderButtonGroup.isSelected(m_Button2.getModel()))){  
        JOptionPane.showMessageDialog(parentComponent: this, message: "Missing information");  
    }else {  
        try {  
            String CCCD = CCCDtxt.getText();  
            String Name = Nametxt.getText();  
            Date Birthday = dateFormat.parse(source: Birthdaytxt.getText());  
            String Phone = Phonetxt.getText();  
            String Email = Emailtxt.getText();  
            String Address = Addresstxt.getText();  
            String SocialMedia = SocialMediatxt.getText();  
            String Gender = "";  
            if(Button1.isSelected()){  
                Gender += "Male";  
            }else if(Button2.isSelected()){  
                Gender += "Female";  
            }  
            String Nation = Nationtxt.getText();  
            Customers cus = new Customers(CustomerCCCD: CCCD, CustomerName: Name, Cus_Phone: Phone, Email, Address, SocialMedia, Gender, Nation, DateOfBirth: null);  
  
            int result = cusDATA.insert(o: cus);  
            if(result == 0){  
                JOptionPane.showMessageDialog(parentComponent: this, message: "Add Failed!!");  
            }else {  
                JOptionPane.showMessageDialog(parentComponent: this, message: "Add Successful!!");  
                table.setRowCount(rowCount: 0);  
                ArrayList<Customers> list = cusDATA.getAll();  
                addData(list);  
            }  
        }catch (ParseException ex) {  
            Logger.getLogger(name: CustomersForm.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);  
        }  
    }  
}
```

# Insert Customer

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

Search CustomerCCCD

## Customers Information

CustomerC...	CustomerN...	DateOfBirth	Cus_Phone	Email	Address	SocialMedia	Gender	Nation
09220307654	Ryu Min-seok	2002-10-14	0123456789	champion@...	Busan	https://www...	Male	Korea
09220401334	Eva Brown	1995-03-10	01122334455	eva.brown@...	789 Pine St, ...	https://www...	Female	UK
09220402634	Nguyen Kim...	2005-08-06	0887654321	trong@gmail...	TPHCM	https://www...	Male	Viet Nam

Message

Add Successfull!!

OK

Add Customer

CCCD: 09220678593

Address: Quan 7 TPHCM

Name: Nguyen Van B

SocialMedia: facebook.com/nguyenvanb/

Birthday: Nov 23, 2005

Gender: Male

Phone: 0892223856

Nation: Viet Nam

Email: nguyenvanb@gmail.com

Confirm

# Update Customer

```
private void UpdateActionPerformed(java.awt.event.ActionEvent evt) {  
    CustomersDAO cusDATA = new CustomersDAO();  
    int result = 0;  
    for(int i = 0 ; i < table.getRowCount();i++) {  
        if(table.getValueAt(row: i, column: 0).equals(obj: "") ||  
        table.getValueAt(row: i, column: 1).equals(obj: "") ||  
        table.getValueAt(row: i, column: 2).equals(obj: "") ||  
        table.getValueAt(row: i, column: 3).equals(obj: "") ||  
        table.getValueAt(row: i, column: 4).equals(obj: "") ||  
        table.getValueAt(row: i, column: 5).equals(obj: "") ||  
        table.getValueAt(row: i, column: 6).equals(obj: "") ||  
        table.getValueAt(row: i, column: 7).equals(obj: "") ||  
        table.getValueAt(row: i, column: 8).equals(obj: ""){  
            JOptionPane.showMessageDialog(parentComponent: CustomersForm.this,message: "Please enter full information");  
            break;  
        }else{  
            try{  
                Customers cus = new Customers();  
                cus.setCustomerCCC((String)table.getValueAt(row: i, column: 0));  
                cus.setCustomerName((String)table.getValueAt(row: i, column: 1));  
                String DateValue = (String)table.getValueAt(row: i, column: 2);  
                Date birth = dateFormat.parse(source: DateValue);  
                cus.setDateOfBirth(dateOfBirth: birth);  
                cus.setCus_Phone((String)table.getValueAt(row: i, column: 3));  
                cus.setEmail((String)table.getValueAt(row: i, column: 4));  
                cus.setAddress((String)table.getValueAt(row: i, column: 5));  
                cus.setSocialMedia((String)table.getValueAt(row: i, column: 6));  
                cus.setGender((String)table.getValueAt(row: i, column: 7));  
                cus.setNation((String)table.getValueAt(row: i, column: 8));  
                result = cusDATA.update(o: cus); //result = 0 => that bai , result > 0 => thanh cong  
            } catch (ParseException ex) {  
                Logger.getLogger(name:CustomersForm.class.getName()).log(level:Level.SEVERE, msg: null, thrown: ex);  
            }  
        }  
        if(result == 0 ){  
            JOptionPane.showMessageDialog(parentComponent: CustomersForm.this,message: "Cannot Update");  
        }else {  
            JOptionPane.showMessageDialog(parentComponent: CustomersForm.this,message: "Updated!!!");  
        }  
        table.setRowCount(rowCount: 0);  
        ArrayList<Customers> list = cusDATA.getAll();  
        addData(list);  
    }  
}
```

# Delete Customer

```
public int delete(String id) {  
    PreparedStatement stmt = null;  
    int result = 0;//result == 1 => thanh cong // result == 0 => that bai  
    try{  
        Connection conn = Conn();  
        stmt = conn.prepareStatement(string: "DELETE FROM Customer where CustomersCCCD = ?");  
        stmt.setString(i: 1, string: id);  
        result = stmt.executeUpdate();  
    }catch(Exception e){  
        System.out.println("Cannot connect: "+ e);  
    }  
    return result;  
}  
  
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    CustomersDAO cusDATA = new CustomersDAO();  
    int row = CustomersTable.getSelectedRow();  
    if(row == -1){  
        JOptionPane.showMessageDialog(parentComponent: CustomersForm.this, message: "Please choose customer to delete!");  
    }else {  
        int cf = JOptionPane.showConfirmDialog(parentComponent: CustomersForm.this, message: "Are you sure to delete this customer?");  
        if(cf == JOptionPane.YES_OPTION) {  
            String id = String.valueOf(obj: table.getValueAt(row, column: 0));  
            cusDATA.delete(id);  
            table.setRowCount(rowCount:0);  
            ArrayList<Customers> list = cusDATA.getAll();  
            addData(list);  
        }  
    }  
}
```

# Search Customer

```
public ArrayList<Customers> search(String Condition, String things) {  
    ArrayList<Customers> list = new ArrayList();  
    Statement st = null;  
    try{  
        Connection conn = Conn();  
        st = conn.createStatement();  
        ResultSet rs = st.executeQuery("Select * from Customer where " + Condition + " LIKE '%" + things + "%'");  
        while(rs.next()) {  
            String CCCD = rs.getString("CustomerCCCD");  
            String name = rs.getString("CustomerName");  
            Date birth = rs.getDate("DateOfBirth");  
            String phone = rs.getString("Cus_Phone");  
            String email = rs.getString("Email");  
            String Address = rs.getString("Address");  
            String socialmedia = rs.getString("SocialMedia");  
            String Gender = rs.getString("Gender");  
            String Nation = rs.getString("Nation");  
            Customers cus = new Customers(CustomerCCCD, CCCD, CustomerName, name, Cus_Phone, phone, Email, email, Address, SocialMedia, socialmedia, Gender, Nation);  
            list.add(cus);  
        }  
    }catch(Exception e){  
        System.out.println("Cannot Connect" + e);  
    }  
    return list;  
}  
  
private void SearchtxtKeyReleased(java.awt.event.KeyEvent evt) {  
    CustomersDAO cusDATA = new CustomersDAO();  
    ArrayList<Customers> list = cusDATA.search((String)SearchBox.getSelectedItem(),things, Searchtxt.getText());  
    table.setRowCount(rowCount:0);  
    addData(list);  
    if(Searchtxt.getText().equals(anObject:" ")){  
        list = cusDATA.getAll();  
        table.setRowCount(rowCount:0);  
        addData(list);  
    }  
}
```

# Search Customer

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

## Customers Information

Search CustomerName ryu

CustomerC...	CustomerN...	DateOfBirth	Cus_Phone	Email	Address	SocialMedia	Gender	Nation
09220307654	Ryu Min-seok	2002-10-14	0123456789	champion@...	Busan	https://www...	Male	Korea

ADD UPDATE DELETE

# Insert PetProduct

```
private void ConfirmActionPerformed(java.awt.event.ActionEvent evt) {  
    PetProductDAO productDATA = new PetProductDAO();  
    ArrayList<PetProduct> list = productDATA.getbyID(id: IDtxt.getText());  
    if(IDtxt.getText().isBlank() ||  
        Nametxt.getText().isBlank() ||  
        Receiptdate.getDate().equals(obj: "") || Priceissuetxt.getText().isBlank() || Priceselltxt.getText().isBlank())  
        JOptionPane.showMessageDialog(parentComponent: this, message: "Missing information");  
    else if(!list.isEmpty()){  
        JOptionPane.showMessageDialog(parentComponent: this, message: "ID already used");  
    }else {  
        try{  
            PetProduct pp = new PetProduct();  
            pp.setProduct_ID(Product_ID: IDtxt.getText());  
            pp.setProduct_Name(Product_Name: Nametxt.getText());  
            pp.setPrice_issue(Price_issue: Float.parseFloat(s: Priceissuetxt.getText()));  
            pp.setPrice_sell(Price_sell: Float.parseFloat(s: Priceselltxt.getText()));  
            pp.setStock(Stock: Integer.parseUnsignedInt(s: Stocktxt.getText()));  
            String date = dateFormat.format(date: Receiptdate.getDate());  
            pp.setReceipt_Day(Receipt_Day: dateFormat.parse(source: date));  
            int result = productDATA.insert(o: pp);  
            if(result == 0){  
                JOptionPane.showMessageDialog(parentComponent: this, message: "Add Failed!!");  
            }else {  
                JOptionPane.showMessageDialog(parentComponent: this, message: "Add Successful!!");  
                table.setRowCount(rowCount: 0);  
                ArrayList<PetProduct> list = productDATA.getAll();  
                addData(list);  
            }  
        }catch(Exception e){  
            System.out.println("Error :" + e);  
        }  
    }  
}
```

# Insert PetProduct

Add Product

ID	<input type="text" value="IT_4"/>	Price issue	<input type="text" value="5"/>
Name	<input type="text" value="Ball"/>	Price sell	<input type="text" value="5"/>
Receipt day	<input type="text" value="2023-11-26"/>	Stock	<input type="text" value="30"/>
<input type="button" value="Close"/>		<input type="button" value="Confirm"/>	

← Add

Product_ID	Product_Name	Price_issue	Price_sell	Stock	Receipt_Day
IT_1	Dog Food	200000.0	250000.0	88	2023-11-15
IT_2	Cat Toy	100000.0	150000.0	38	2023-11-15
IT_3	Bird Cage	300000.0	400000.0	13	2023-11-15
IT_4	Ball	5.0	5.0	30	2023-11-26

Database →

# Update PetProduct

```
private void UpdateActionPerformed(java.awt.event.ActionEvent evt) {  
PetProductDAO productDATA = new PetProductDAO();  
    int result = 0;  
    for(int i = 0 ; i < table.getRowCount();i++) {  
        if(table.getValueAt(row: i, column: 0).equals(obj: "") ||  
        table.getValueAt(row: i, column: 1).equals(obj: "") ||  
        table.getValueAt(row: i, column: 2).equals(obj: "") ||  
        table.getValueAt(row: i, column: 3).equals(obj: "") ||  
        table.getValueAt(row: i, column: 4).equals(obj: "") ||  
        table.getValueAt(row: i, column: 5).equals(obj: "")){  
            JOptionPane.showMessageDialog(parentComponent: ProductsForm.this,message: "Please enter full information");  
            break;  
        }else{  
            try{  
                PetProduct pp = new PetProduct();  
                pp.setProduct_ID(Product_ID: String.valueOf(obj: ProductTable.getValueAt(row: i, column: 0)));  
                pp.setProduct_Name(Product_Name: String.valueOf(obj: ProductTable.getValueAt(row: i, column: 1)));  
                pp.setPrice_issue(Price_issue: Float.parseFloat(String.valueOf(obj: ProductTable.getValueAt(row: i, column: 2))));  
                pp.setPrice_sell(Price_sell: Float.parseFloat(String.valueOf(obj: ProductTable.getValueAt(row: i, column: 3))));  
                pp.setStock(Stock: Integer.parseInt(String.valueOf(obj: ProductTable.getValueAt(row: i, column: 4))));  
                String DateValue = String.valueOf(obj: ProductTable.getValueAt(row: i, column: 5));  
                Date birth = dateFormat.parse(source: DateValue);  
                pp.setReceipt_Day(Receipt_Day: birth);  
                result = productDATA.update(o: pp); //result = 0 => that bai , result > 0 => thanh cong  
            }catch (ParseException ex) {  
                System.out.println("Error " + ex);  
            }  
        }  
    }  
    if(result == 0 ){  
        JOptionPane.showMessageDialog(parentComponent: ProductsForm.this,message: "Cannot Update");  
    }else {  
        JOptionPane.showMessageDialog(parentComponent: ProductsForm.this,message: "Updated!!!");  
    }  
    table.setRowCount(rowCount:0);  
    ArrayList<PetProduct> list = productDATA.getAll();  
    addData(list);  
}
```

# Update PetProduct

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

## Product Information

Search Product\_ID

Product_ID	Product_Name	Price_issue	Price_sell	Stock	Receipt_Day
IT_1	Dog Food	200000.0	20	88	2023-11-15
IT_2	Cat Toy	100000.0	200	38	2023-11-15
IT_3	Bird Cage	300000.0	50	13	2023-11-15
IT_4	Ball	5.0	5.0	30	2023-11-26

ADD UPDATE DELETE

Message X

Updated!!! i

OK

# Delete PetProduct

```
public int delete(String id) {  
    PreparedStatement stmt = null;  
    int result = 0;//result == 1 => thanh cong // result == 0 => that bai  
    try{  
        Connection conn = Conn();  
        stmt = conn.prepareStatement(string: "DELETE FROM PetProduct where Product_ID = ?");  
        stmt.setString(i: 1, string: id);  
        result = stmt.executeUpdate();  
    }catch(Exception e){  
        System.out.println("Cannot connect: "+ e);  
    }  
    return result;  
}  
  
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    PetProductDAO cusDATA = new PetProductDAO();  
    int row = ProductTable.getSelectedRow();  
    if(row == -1){  
        JOptionPane.showMessageDialog(parentComponent: ProductsForm.this, message: "Please choose product to delete!");  
    }else {  
        int cf = JOptionPane.showConfirmDialog(parentComponent: ProductsForm.this, message: "Are you sure to delete this product?");  
        if(cf == JOptionPane.YES_OPTION) {  
            String id = String.valueOf(obj: table.getValueAt(row, column: 0));  
            cusDATA.delete(id);  
            table.setRowCount(rowCount: 0);  
            ArrayList<PetProduct> list = cusDATA.getAll();  
            addData(list);  
        }  
    }  
}
```

# Search PetProduct

```
public ArrayList<PetProduct> search(String Condition, String things) {  
    ArrayList<PetProduct> list = new ArrayList();  
    Statement st = null;  
    try{  
        Connection conn = Conn();  
        st = conn.createStatement();  
        ResultSet rs = st.executeQuery("Select * from PetProduct where " + Condition + " LIKE '%" + things + "%'");  
        while(rs.next()) {  
            PetProduct pp = new PetProduct();  
            pp.setProduct_ID(Product_ID: rs.getString(string: "Product_ID"));  
            pp.setProduct_Name(Product_Name: rs.getString(string: "Product_Name"));  
            pp.setPrice_issue(Price_issue: rs.getFloat(string: ""));  
            pp.setPrice_sell(Price_sell: rs.getBigDecimal(string: "Price_sell").floatValue());  
            pp.setStock(Stock: rs.getInt(string: "Stock"));  
            pp.setReceipt_Day(Receipt_Day: rs.getDate(string: "Receipt_day"));  
            list.add(e: pp);  
        }  
    }catch(Exception e){  
        System.out.println("Cannot Connect" + e);  
    }  
    return list;  
}  
  
private void SearchtxtKeyReleased(java.awt.event.KeyEvent evt) {  
    PetProductDAO productDATA = new PetProductDAO();  
    ArrayList<PetProduct> list = productDATA.search((String)SearchBox.getSelectedItem(),things: Searchtxt.getText());  
    table.setRowCount(rowCount:0);  
    addData(list);  
    if(Searchtxt.getText().equals(anObject: ""){  
        list = productDATA.getAll();  
        table.setRowCount(rowCount:0);  
        addData(list);  
    }  
}
```

# Insert Pet

```
private void ConfirmActionPerformed(java.awt.event.ActionEvent evt) {
PetDAO petDATA = new PetDAO();
    if(petidtxt.getText().isBlank() || pettypetxt.getText().isBlank() || petbreedtxt.getText().isBlank() || petnametxt.getText().isBlank() || colortxt.getText().isBlank() || agetxt.getText().isBlank() || weighttxt.getText().isBlank() || healthtxt.getText().isBlank() || pricetxt.getText().isBlank() ||
(!MaleRadioButton.isSelected() && !FemaleRadioButton.isSelected()) ||
(!ShotsYesButton.isSelected() && !ShotsNoButton.isSelected()) ||
(!FlushYesButton.isSelected() && !FlushNoButton.isSelected())){
 JOptionPane.showMessageDialog(parentComponent: this, message: "Missing information");
} else {
    String PetId = petidtxt.getText();
    String pettype = pettypetxt.getText();
    String petbreed = petbreedtxt.getText();
    String petname = petnametxt.getText();
    String color = colortxt.getText();
    int age = Integer.parseInt(agetxt.getText());
    Float weight = Float.parseFloat(weighttxt.getText());
    String health = healthtxt.getText();
    Float price = Float.parseFloat(pricetxt.getText());
    String picture = imagepath;
    if(picture.equals("C:\\\\Users\\\\84896\\\\Documents\\\\NetBeansProjects\\\\DBMS\\\\src\\\\icons\\\\dog.png")){
        JOptionPane.showMessageDialog(parentComponent: this, message: "Please choose picture");
        return;
    }
    String Gender = "";
    if(MaleRadioButton.isSelected()){
        Gender += "Male";
    } else if(FemaleRadioButton.isSelected()){
        Gender += "Female";
    }
    String Shots = "";
    if(ShotsYesButton.isSelected()){
        Shots += "Yes";
    } else if(ShotsNoButton.isSelected()){
        Shots += "No";
    }
    String Flush = "";
    if(FlushYesButton.isSelected()){
        Flush += "Yes";
    } else if(FlushNoButton.isSelected()){
        Flush += "No";
    }
    Pet p = new Pet(Pet_ID: PetId, Pet_Type: pettype, Pet_Breed: petbreed, Pet_Name: petname, Color: color, Age: age, Weight: weight,
Picture: picture, Gender: Gender, Shots: Shots, Flush: Flush, Health: health, Price: price);
    int result = petDATA.insert(o p);
    if(result == 0){
        JOptionPane.showMessageDialog(parentComponent: this, message: "Add Failed!!!");
    } else {
        JOptionPane.showMessageDialog(parentComponent: this, message: "Add Successful!!!");
        table.setRowCount(rowCount: 0);
        ArrayList<Pet> list = petDATA.getAll();
        addData(list);
    }
}
```

# Insert Pet

## Add Pet

Picture

Pet_ID	C1	Gender	<input checked="" type="radio"/> Male	<input type="radio"/> Female	
Pet_Type	Cat	Shots	<input checked="" type="radio"/> Yes	<input type="radio"/> No	
Pet_Breed	English shorthair	Flush	<input checked="" type="radio"/> Yes	<input type="radio"/> No	
Pet_Name	Liked	Health	<input checked="" type="radio"/> Healthy	<input type="radio"/> Unhealthy	
Color	Brown and white	Age	4	Price	100
Weight	6				

Database →

← Add

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

List of Pet

Update Picture

Search... Pet\_ID

Pet_ID	Pet_Type	Pet_Bre...	Pet_Name	Color	Age	Weight	Picture	Gender	Shots	Flush	Heath	Price
1	Horse	Camarillo	Jolly Ju...	White	8	300.0	C:\Users...	Male	Yes	Yes	Healthy	10800.0
2	Cat	British L...	Lion	White	3	4.0	C:\Users...	Male	No	No	Healthy	200.0
4	Dog	Husky	Mickey	Black an...	1	2.0	C:\Users...	Female	Yes	No	Eye prob...	249.0
5	Bird	Parrot	Jack Sp...	Orange ...	3	4.0	C:\Users...	Male	No	Yes	Healthy	199.0
6	Rabbit	English ...	Hopper	Brown a...	2	3.0	C:\Users...	Female	No	Yes	Healthy	99.0
7	Snake	Rough ...	Syther	Green	2	1.0	C:\Users...	Female	No	No	Healthy	139.0
8	Rabbit	New Zel...	Bonnie	White	3	3.0	C:\Users...	Male	No	No	Healthy	99.0
9	Cat	Exotic	Kingen	White an...	5	20.0	C:\Users...	Male	No	No	Healthy	499.0
C1	Cat	English ...	Liked	Brown a...	4	6.0	C:\Users...	Male	Yes	Yes	Healthy	100.0

ADD UPDATE DELETE

# Update Pet

```
private void UpdateActionPerformed(java.awt.event.ActionEvent evt) {  
    PetDAO PetDATA = new PetDAO();  
    int result = 0;  
    for(int i = 0 ; i < table.getRowCount();i++) {  
        if(table.getValueAt(row: i, column: 0).equals(obj: "") ||  
        table.getValueAt(row: i, column: 1).equals(obj: "") ||  
        table.getValueAt(row: i, column: 2).equals(obj: "") ||  
        table.getValueAt(row: i, column: 3).equals(obj: "") ||  
        table.getValueAt(row: i, column: 4).equals(obj: "") ||  
        table.getValueAt(row: i, column: 5).equals(obj: "") ||  
        table.getValueAt(row: i, column: 6).equals(obj: "") ||  
        table.getValueAt(row: i, column: 7).equals(obj: "") ||  
        table.getValueAt(row: i, column: 8).equals(obj: "") ||  
        table.getValueAt(row: i, column: 9).equals(obj: "") ||  
        table.getValueAt(row: i, column: 10).equals(obj: "") ||  
        table.getValueAt(row: i, column: 11).equals(obj: "") ||  
        table.getValueAt(row: i, column: 12).equals(obj: ""){  
            JOptionPane.showMessageDialog(parentComponent: PetForm.this,message: "Please choose Picture to update!");  
            return;  
        }else{  
            Pet pet = new Pet();  
            pet.setPet_ID((String)table.getValueAt(row: i, column: 0));  
            pet.setPet_Type((String)table.getValueAt(row: i, column: 1));  
            pet.setPet_Breed((String)table.getValueAt(row: i, column: 2));  
            pet.setPet_Name((String)table.getValueAt(row: i, column: 3));  
            pet.setColor((String)table.getValueAt(row: i, column: 4));  
            pet.setAge(Age: Integer.parseInt((String)table.getValueAt(row: i, column: 5)));  
            pet.setWeight(Weight: Float.parseFloat((String)table.getValueAt(row: i, column: 6)));  
            pet.setPicture((String)table.getValueAt(row: i, column: 7));  
            pet.setGender((String)table.getValueAt(row: i, column: 8));  
            pet.setShots((String)table.getValueAt(row: i, column: 9));  
            pet.setFlush((String)table.getValueAt(row: i, column: 10));  
            pet.setHeath((String)table.getValueAt(row: i, column: 11));  
            pet.setPrice(Price: Float.parseFloat((String)table.getValueAt(row: i, column: 12)));  
            result = PetDATA.update(o: pet);  
        }  
        JOptionPane.showMessageDialog(parentComponent: PetForm.this,message: "Updated!!!");  
        Refresh();  
    }  
}  
  
private void UpdatepictureActionPerformed(java.awt.event.ActionEvent evt) {  
    if(current.equals(anObject: "")){  
        JOptionPane.showMessageDialog(parentComponent: PetForm.this, message: "Please choose Picture to update!");  
    }else{  
        try{  
            JFileChooser k = new JFileChooser(currentDirectoryPath: "C:\\\\Users\\\\84896\\\\Documents\\\\NetBeansProjects\\\\DBMS\\\\src\\\\icons");  
            FileNameExtensionFilter imageFilter = new FileNameExtensionFilter(description: "picture",extensions: "png", extensions: "jpg");  
            k.setFileFilter(filter: imageFilter);  
            k.setDialogTitle(dialogTitle: "Open file");  
            k.showOpenDialog(parent: this);  
            File filename = k.getSelectedFile();  
            String newImage = filename.getAbsolutePath();  
            PetDAO petDATA = new PetDAO();  
            ArrayList<Pet> getpet = petDATA.getbyID(id: current);  
            getpet.get(index: 0).setPicture(picture: newImage);  
            petDATA.update(o: getpet.get(index: 0));  
            PictureLabel.setIcon(icon: ResizeImage(imagePath: newImage,picture: PictureLabel));  
            Refresh();  
        }catch(Exception e){  
            System.out.println(x: e);  
        }  
    }  
}
```

# Delete Pet

```
public int delete(String id) {
    PreparedStatement stmt = null;
    int result = 0;//result == 1 => thanh cong // result == 0 => that bai
    try{
        Connection conn = Conn();
        stmt = conn.prepareStatement(string: "DELETE FROM Pet where Pet_ID = ?");
        stmt.setString(1, string: id);
        result = stmt.executeUpdate();
    }catch(Exception e){
        System.out.println("Cannot connect: "+ e);
    }
    return result;
}

private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    PetDAO petDATA = new PetDAO();
    int row = PetTable.getSelectedRow();
    if(row == -1){
        JOptionPane.showMessageDialog(parentComponent: PetForm.this, message: "Please choose Pet to delete!");
    }else {
        int cf = JOptionPane.showConfirmDialog(parentComponent: PetForm.this, message: "Are you sure to delete this Pet?");
        if(cf == JOptionPane.YES_OPTION) {
            String id = String.valueOf(obj: PetTable.getValueAt(row, column: 0));
            petDATA.delete(id);
            table.setRowCount(rowCount: 0);
            ArrayList<Pet> list = petDATA.getAll();
            addData(list);
        }
    }
}
```

# Search Pet

```
public ArrayList<Pet> search(String Condition, String things) {  
    ArrayList<Pet> list = new ArrayList<Pet>();  
    Statement stm = null;  
    try{  
        Connection c = Conn();  
        stm = c.createStatement();  
        ResultSet rs = stm.executeQuery("Select * from Pet where " + Condition + " LIKE '%" + things + "%'");  
        list.clear();  
        while(rs.next()){  
            String ID = rs.getString(string: "Pet_ID");  
            String Pet_Type = rs.getString(string: "Pet_Type");  
            String Pet_Breed = rs.getString(string: "Pet_Breed");  
            String Pet_Name = rs.getString(string: "Pet_Name");  
            String Color = rs.getString(string: "Color");  
            int Age = rs.getInt(string: "Age");  
            float Weight = rs.getFloat(string: "Weight");  
            String Picture = rs.getString(string: "Picture");  
            String Gender = rs.getString(string: "Gender");  
            String Shots = rs.getString(string: "Shots");  
            String Flush = rs.getString(string: "Flush");  
            String Heath = rs.getString(string: "Health");  
            float Price = rs.getFloat(string: "Price");  
            Pet p = new Pet(Pet_ID: ID, Pet_Type,Pet_Breed, Pet_Name, Color, Age , Weight, Picture, Gender, Shots, Flush, Heath, Price);  
            list.add(e..p);  
        }  
    }catch(Exception e) {  
        System.out.println("Cannot connect" + e);  
    }  
    return list;  
}  
  
private void SearchtxtKeyReleased(java.awt.event.KeyEvent evt) {  
    PetDAO petDATA = new PetDAO();  
    ArrayList<Pet> list = petDATA.search((String)SearchBox.getSelectedItem(),things: Searchtxt.getText());  
    table.setRowCount(rowCount.0);  
    addData(list);  
    if(Searchtxt.getText().equals(anObject: " ")){  
        list = petDATA.getAll();  
        table.setRowCount(rowCount.0);  
        addData(list);  
    }  
}
```

# Search Pet

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

## List of Pet

Sear... Pet\_Type ▼ cat

Update Picture



Pet_ID	Pet_Type	Pet_Bre...	Pet_Name	Color	Age	Weight	Picture	Gender	Shots	Flush	Heath	Price
2	Cat	British L...	Lion	White	3	4.0	C:\Users...	Male	No	No	Healthy	200.0
9	Cat	Exotic	Kingen	White an...	5	20.0	C:\Users...	Male	No	No	Healthy	499.0
C1	Cat	English ...	Liked	Brown a...	4	8.0	C:\Users...	Male	Yes	Yes	Healthy	100.0

ADD

UPDATE

DELETE

# Insert Pet Image

```
private void ChoosePictureButtonActionPerformed(java.awt.event.ActionEvent evt) {  
JFileChooser k = new JFileChooser(currentDirectoryPath: "C:\\Users\\\\84896\\\\Documents\\\\NetBeansProjects\\\\DBMS\\\\src\\\\icons");  
FileNameExtensionFilter imageFilter = new FileNameExtensionFilter(description: "picture", extensions: "png", extensions: "jpg");  
k.setFileFilter(filter: imageFilter);  
k.setDialogTitle(dialogTitle: "Open file");  
k.showOpenDialog(parent: null);  
File filename = k.getSelectedFile();  
imagepath = filename.getAbsolutePath();  
if (imagepath != null){  
    DialogPictureLabel.setIcon(icon: ResizeImage(imagePath: String.valueOf(obj: imagepath), picture: DialogPictureLabel));  
    System.out.println(x: imagepath);  
} else{  
    JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Please select your picture!!");  
}  
}
```

# Update Pet Image

```
private void UpdatepictureActionPerformed(java.awt.event.ActionEvent evt) {  
    if(current.equals("")){
        JOptionPane.showMessageDialog(parentComponent: PetForm.this, message: "Please choose Picture to update!");
    }else{
        try{
            JFileChooser k = new JFileChooser(currentDirectoryPath: "C:\\\\Users\\\\84896\\\\Documents\\\\NetBeansProjects\\\\DBMS\\\\src\\\\icons");
            FileNameExtensionFilter imageFilter = new FileNameExtensionFilter(description: "picture", extensions: "png", extensions: "jpg");
            k.setFileFilter(filter: imageFilter);
            k.setDialogTitle(dialogTitle: "Open file");
            k.showOpenDialog(parent: this);
            File filename = k.getSelectedFile();
            String newImage = filename.getAbsolutePath();
            PetDAO petDATA = new PetDAO();
            ArrayList<Pet> getpet = petDATA.getbyID(ID: current);
            getpet.get(index: 0).setPicture(picture: newImage);
            petDATA.update(o: getpet.get(index: 0));
            PictureLabel.setIcon(icon.ResizeImage(ImagePath: newImage, picture: PictureLabel));
            Refresh();
        }catch(Exception e){
            System.out.println(x: e);
        }
    }
}
```

# Invoicing

```
private void SaveActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        if(invoiceidtxt.getText().isBlank() ||  
        customeridcbiv.getSelectedItem().equals(obj. "Select") ||  
        staffidcbiv.getSelectedItem().equals(obj. "Select") || table.getRowCount() == 0){  
            JOptionPane.showMessageDialog(parentComponent: this, message: "Missing information to save"  
        }  
        int result = 0;  
        Detail_InvoiceDAO diDATA = new Detail_InvoiceDAO();  
        InvoiceDAO ivDATA = new InvoiceDAO();  
        String InvoiceID = invoiceidtxt.getText();  
        if(!ivDATA.getbyID(id: invoiceidtxt.getText()).isEmpty()){  
            JOptionPane.showMessageDialog(parentComponent: this, message: "ID has already used");  
            return;  
        }else{  
            String DateValue = dateFormat.format(date.date.getDate());  
            Date day = dateFormat.parse(source. DateValue);  
            String StaffID = String.valueOf(obj. staffidcbiv.getSelectedItem());  
            String CustomerID = String.valueOf(obj. customeridcbiv.getSelectedItem());  
            float Totalcost = Float.parseFloat(s. totalcosttxt.getText());  
            Invoice iv = new Invoice(InvoiceID,StaffID,CustomerID,sellDay: day, Total_Price: Totalcost);  
            result += ivDATA.insert(o: iv);  
            for(int i = 0 ; i < table.getRowCount(); i++){  
                PetDAO petDATA = new PetDAO();  
                PetProductDAO productDATA = new PetProductDAO();  
                ArrayList<Pet> pet = petDATA.getbyID(id: String.valueOf(obj. table.getValueAt(row: i, column: 0)));  
                ArrayList<PetProduct> product = productDATA.getbyID(id: String.valueOf(obj. table.getValueAt(row: i, column: 0)));  
                if(pet.isEmpty()){  
                    String ProductID = product.get(index: 0).getProduct_ID();  
                    int amount = Integer.parseInt(s. String.valueOf(obj. table.getValueAt(row: i, column: 2)));  
                    product.get(index: 0).setStock(product.get(index: 0).getStock() - amount);  
                    productDATA.update(o: product.get(index: 0));  
                    float price = Float.parseFloat(s. String.valueOf(obj. table.getValueAt(row: i, column: 3)));  
                    Detail_Invoice di = new Detail_Invoice(Invoice_ID: InvoiceID,Product_ID: ProductID,Pet_ID: null,Amount: amount,Price: price);  
                    result += diDATA.insert(o: di);  
                }else if(product.isEmpty()){  
                    String PetID = pet.get(index: 0).getPet_ID();  
                    int amount = 1;  
                    petDATA.delete(id: PetID);  
                    float price = Float.parseFloat(s. String.valueOf(obj. table.getValueAt(row: i, column: 3)));  
                    Detail_Invoice di = new Detail_Invoice(Invoice_ID: InvoiceID,Product_ID: null,Pet_ID: PetID,Amount: amount,Price: price);  
                    result += diDATA.insert(o: di);  
                }  
                if(result == table.getRowCount()+1){  
                    JOptionPane.showMessageDialog(parentComponent: this, message: "Save Failed!!");  
                    System.out.println(o: result);  
                }else{  
                    JOptionPane.showMessageDialog(parentComponent: this, message: "Save Successful!!");  
                    totalprice = dummy;  
                    dummy = 0;  
                }  
            }  
        }  
    }catch (ParseException ex) {  
        Logger.getLogger(name: CustomersForm.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);  
    }  
}
```

# Invoicing

```
private void AddActionPerformed(java.awt.event.ActionEvent evt) {  
  
    if(!productidcbiv.getSelectedItem().equals(obj: "Select")){  
        PetProductDAO productDATA = new PetProductDAO();  
        ArrayList<PetProduct> listp = productDATA.getbyID(id: String.valueOf(obj: productidcbiv.getSelectedItem()));  
        if(amountivtxt.getText().isBlank()){  
            JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Please enter the amount");  
            return;  
        }else if(listp.get(index: 0).getStock() - Integer.parseInt(s: amountivtxt.getText()) < 0 || listp.get(index: 0).getStock() == 0){  
            JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Out of stock");  
        }else{  
            table.addRow(new String[]{String.valueOf(obj: productidcbiv.getSelectedItem()),  
                String.valueOf(obj: listp.get(index: 0).getProduct_Name()),amountivtxt.getText(),productpriceivtxt.getText()});  
            dummy += Float.parseFloat(s: productpriceivtxt.getText());  
            totalcosttxt.setText(t: String.valueOf(f: dummy));  
        }  
    }if(!petidcbiv.getSelectedItem().equals(obj: "Select")){  
        PetDAO petDATA = new PetDAO();  
        ArrayList<Pet> listp = petDATA.getbyID(id: String.valueOf(obj: petidcbiv.getSelectedItem()));  
        table.addRow(new String[] {String.valueOf(obj: petidcbiv.getSelectedItem()),  
            String.valueOf(obj: listp.get(index: 0).getPet_Name()),"1",petpriceivtxt.getText()});  
        dummy += Float.parseFloat(s: petpriceivtxt.getText());  
        totalcosttxt.setText(t: String.valueOf(f: dummy));  
    }else if(!petidcbiv.getSelectedItem().equals(obj: "Select") && !productidcbiv.getSelectedItem().equals(obj: "Select")){  
        JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Please choose your item or pet before add");  
    }  
}
```

# Invoicing

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

InvoiceID: B13 StaffID: ST1 CustomerID: 09220307654 Refresh

Date: Nov 23, 2023 StaffName: Jack CustomerName: Ryu Min-seok

ProductID: IT\_1 ProductName: Dog Food Amount: 3 Price: 750000.0

PetID: 3 PetName: Mouse Price: 2.0

Pet/Product ID	Pet/Product	Price
IT_1	Dog Food	750000.0
3	Niggi	2.0

Message: Save Successfull! OK

Delete Add Save Print Total Price: 750002.0

# Print Invoice

```
private void printActionPerformed(java.awt.event.ActionEvent evt) {  
    try{  
        HSSFWorkbook workbook = new HSSFWorkbook();  
        HSSFSheet sheet = workbook.createSheet(sheetname: "Invoice");  
        HSSFRow row = null;  
        Cell cell = null;  
        row = sheet.createRow(rownum: 0);  
        cell = row.createCell(columnIndex: 0,type:CellType.STRING);  
        cell.setCellValue(string: "Staff Name ");  
        cell = row.createCell(columnIndex: 1,type:CellType.STRING);  
        cell.setCellValue(string: staffnametxt.getText());  
        row = sheet.createRow(rownum: 1);  
        cell =row.createCell(columnIndex: 0,type:CellType.STRING);  
        cell.setCellValue(string: "Customer Name ");  
        cell =row.createCell(columnIndex: 1,type:CellType.STRING);  
        cell.setCellValue(string: customernametxt.getText());  
        row = sheet.createRow(rownum: 2);  
        cell = row.createCell(columnIndex: 2,type:CellType.STRING);  
        cell.setCellValue(string: "INVOICE");  
        row = sheet.createRow (rownum: 3);  
        cell = row.createCell(columnIndex: 0, type:CellType.STRING);  
        cell.setCellValue(string: "STT");  
        cell = row.createCell(columnIndex: 1, type:CellType.STRING);  
        cell.setCellValue(string: "ID");  
        cell = row.createCell(columnIndex: 2, type:CellType.STRING);  
        cell.setCellValue(string: "Name");  
        cell = row.createCell(columnIndex: 3, type:CellType.STRING);  
        cell.setCellValue(string: "Amount");  
        cell = row.createCell(columnIndex: 4, type:CellType.STRING);  
        cell.setCellValue(string: "Price");  
        int i;  
        for(i = 0 ; i < table.getRowCount(); i++){  
            row = sheet.createRow(4+i);  
            cell = row.createCell(columnIndex: 0, type:CellType.NUMERIC);  
            cell.setCellValue(i+1);  
            cell = row.createCell(columnIndex: 1,type:CellType.STRING);  
            cell.setCellValue(string: String.valueOf(obj: table.getValueAt(row: i, column: 0)));  
            cell = row.createCell(columnIndex: 2, type:CellType.STRING);  
    }  
}
```



# Monthly Revenue

```
private void CalculateActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        float totalcost = 0;
        String sDateValue = dateFormat.format(date.Startday.getDate());
        Date StartDay = dateFormat.parse(source.sDateValue);
        String fDateValue = dateFormat.format(date.Finishday.getDate());
        Date FinishDay = dateFormat.parse(source.fDateValue);
        InvoiceDAO iv = new InvoiceDAO();
        ArrayList<Invoice> list = iv.findByDate(start.StartDay, finish.FinishDay);
        for(Invoice i : list){
            Revenuetable.addRow(new String[] {i.getInvoiceID(),
                i.getStaffID(), i.getCustomerCCCD(),
                String.valueOf(i.getSellDay()),
                String.valueOf(i.getTotal_Price())});
            totalcost += i.getTotal_Price();
        }
        Revenuetxt.setText("Total Cost: " + String.valueOf(totalcost));
    } catch (ParseException ex) {
        java.util.logging.Logger.getLogger(AdminPage.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

```
public ArrayList<Invoice> findByDate(Date Start, Date Finish){
    PreparedStatement stmt = null;
    ArrayList<Invoice> list = new ArrayList();
    try{
        Connection conn = Connect.Conn();
        stmt = conn.prepareStatement(string: "SELECT * FROM Invoice where SellDay BETWEEN ? and ?");
        stmt.setDate(i: 1, new java.sql.Date(date.Start.getTime()));
        stmt.setDate(i: 2, new java.sql.Date(date.Finish.getTime()));
        ResultSet rs = stmt.executeQuery();
        while (rs.next()) {
            String Invoice_ID = rs.getString(string: "Invoice_ID");
            String CustomerCCCD = rs.getString(string: "CustomerCCCD");
            String Staff_ID = rs.getString(string: "Staff_ID");
            Date SellDay = rs.getDate(string: "SellDay");
            Float Total_Price = rs.getFloat(string: "Total_Price");
            Invoice i = new Invoice(InvoiceID:Invoice_ID, staffID: Staff_ID, CustomerCCCD, SellDay, Total_Price);
            list.add(e: i);
        }
    }catch(Exception e){
        System.out.println("Cannot connect: " + e);
    }
    return list;
}
```



# Monthly Revenue

Welcome to Only Pet

Staff Account Customer Pet Product Invoicing Revenue

From Date: Nov 23, 2023

To Date: Nov 26, 2023

Revenue: 9711400.0

InvoiceID	StaffID	CustomerID	SellDay	TotalCost
B1	ST1	09220307654	2023-11-25	1200200.0
B1	ST1	09220307654	2023-11-25	450199.0
B12	ST3	09220401334	2023-11-26	2950000.0
B13	ST1	09220307654	2023-11-23	750002.0
B2	ST1	09220401334	2023-11-25	760800.0
B6	AMD	09220307654	2023-11-25	750199.0
B7	ST1	09220307654	2023-11-25	1250000.0
B8	ST1	09220307654	2023-11-25	1050000.0
B9	ST1	09220307654	2023-11-25	550000.0

**Thank you!!!**