

Final Documentation

CS 3320 Files & Database

Alin Bocardo, Katelyn Espinoza, Savannah Lopez, Nadia Morrow

November 15, 2022

Table of Contents

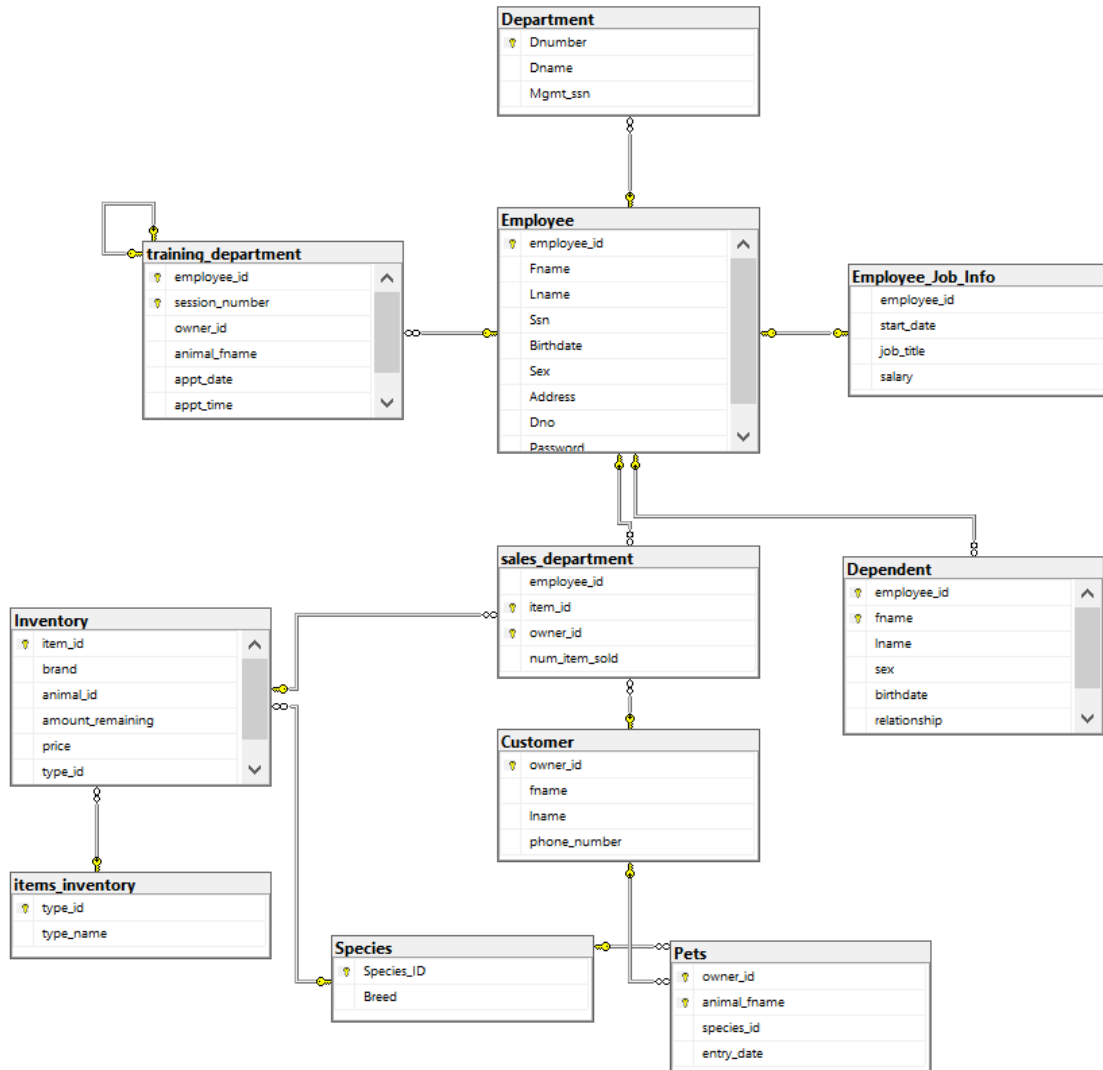
Introduction	3
Project Specifications	3
Design	5
User Manual	8
Summary and Conclusion	12
Future Work	13
References	14
Code Listing	15

Introduction

Owning a pet requires numerous elements to ensure a pet is living a healthy life. Some of the elements required of owning a pet include food, grooming supplies, supplements, water and food bowl, and bathroom essentials¹. A pet store acts as a retail business which allows a pet owner to shop all in one place for these essential elements and in some cases a pet store may offer training sessions for a pet². Thus, we propose the creation of a Pet Store Database System which will fulfill the capabilities of a simple computer-based system which will be used by a pet store and the pet store's in-house training facility.

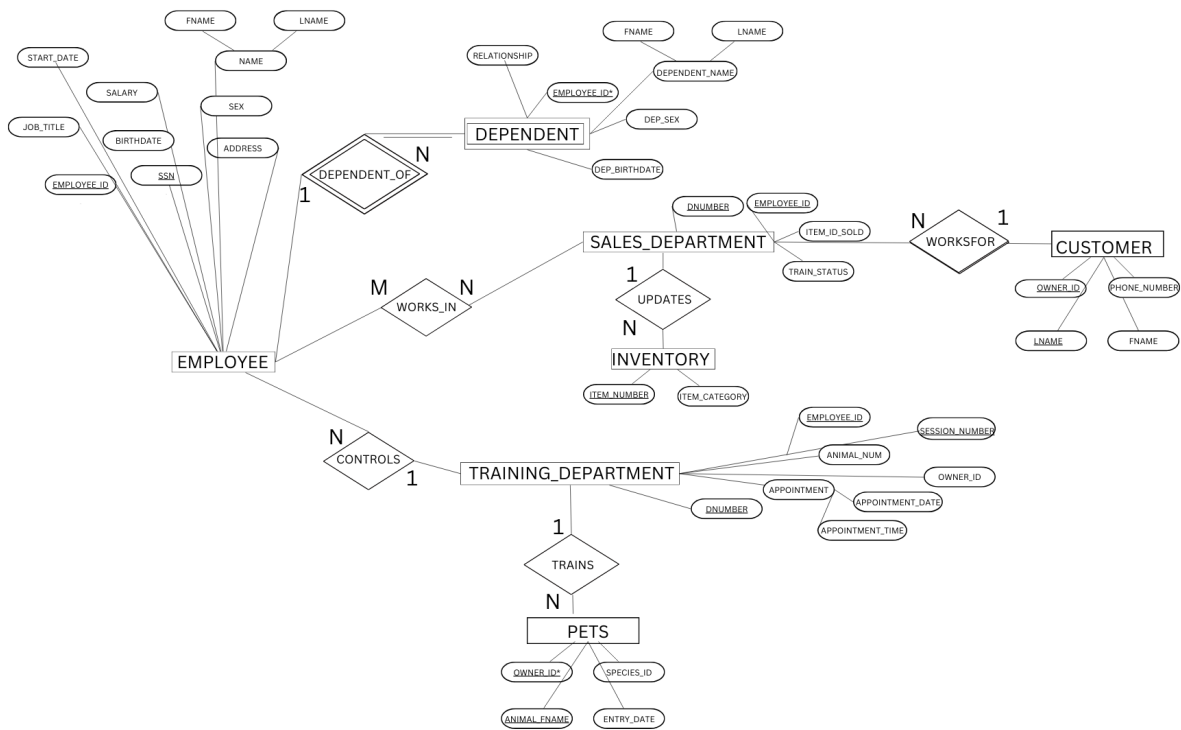
Project Specifications

The Pet Database System will fulfill the capabilities of a simple computer-based system used by a Pet Store and Training Facility. It will allow employees to access various forms of information, including their own, the store's inventory, all animal information and customer history. Employees will also be able to update the information about sales and scheduling. An employee also will be able to add a new customer, load, or update an existing customer. The application will be using the languages C# and SQL, will be developed on operating systems of Windows 10 and 11, and the environments used will include Visual Studio and Microsoft SQL Server Management Studio.



Design

(I) ERD



(II) Schema and data types for each attribute:

EMPLOYEE

<u>Employee_ID</u>	Fname	Lname	Start_date	Job_title	Ssni	Salary	Birthdate	Sex	Address
int	varchar(50)	varchar(50)	varchar(10)	varchar(50)	nt	int	varchar(10)	char(1)	nvarchar(50)

DEPENDENT

<u>Employee_ID</u>	<u>Depen_fname</u>	Depen_lname	Dep_sex	Dep_Birthdate	Relationship
int	varchar(50)	varchar(50)	char(1)	varchar(10)	varchar(50)

DEPARTMENT

<u>Dnumber</u>	Dname
int	varchar(50)

MANAGEMENT

<u>Dnum</u>	Mgmt._ssn
<i>int</i>	<i>int</i>

TRAINING DEPARTMENT

<u>Employee_ID</u>	<u>Session_number</u>	Animal_num	Appt_date	Appt_time
<i>int</i>	<i>int</i>	<i>int</i>	<i>varchar(10)</i>	<i>varchar(10)</i>

SALES DEPARTMENT

<u>Employee_ID</u>	<u>Item_id_sold</u>	Train_status
<i>int</i>	<i>int</i>	<i>varchar(50)</i>

ANIMAL

<u>Animal_ID</u>	<u>Species</u>
<i>int</i>	<i>varchar(50)</i>

INVENTORY

<u>Item_number</u>	Item_Category
<i>int</i>	<i>varchar(50)</i>

FOOD

<u>Item_number</u>	Food_Brand	Type_of_Animal	Amount_Remaining	Item_Price
<i>int</i>	<i>varchar(50)</i>	<i>varchar(50)</i>	<i>int</i>	<i>decimal(18,2)</i>

CLEANING SUPPLY

<u>Item_number</u>	Cleaning_Brand	Type_of_Animal	Amount_Remaining	Item_Price
<i>int</i>	<i>varchar(50)</i>	<i>varchar(50)</i>	<i>int</i>	<i>decimal(18,2)</i>

TOY

<u>Item_number</u>	Toy_Brand	Type_of_Animal	Amount_Remaining	Item_Price
<i>int</i>	<i>varchar(50)</i>	<i>varchar(50)</i>	<i>int</i>	<i>decimal(18,2)</i>

(III)Relationships:

- Employee_ID – Fname (1:1)

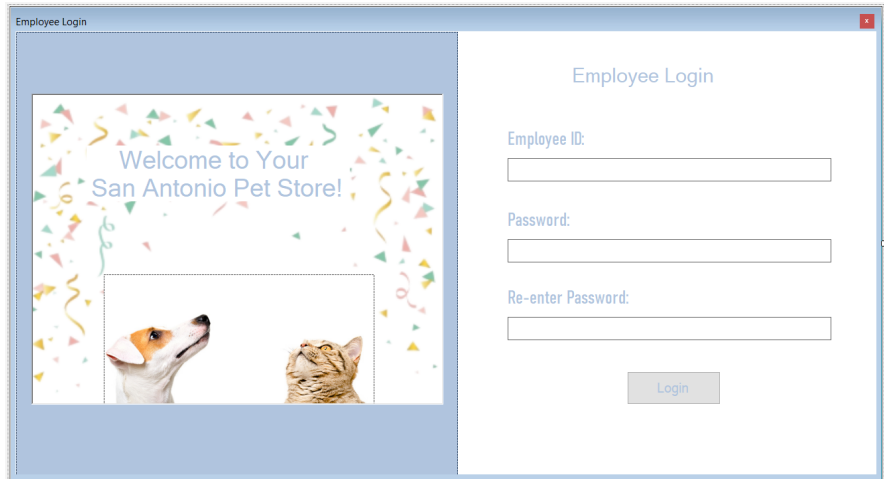
- Employee_ID – Lname (1:1)
- Employee_ID – Start_date (1:1)
- Employee_ID – Job_title (1:1)
- Employee_ID – Ssn (1:1)
- Employee_ID – Salary (1:1)
- Employee_ID – Birthdate (1:1)
- Employee_ID – Sex (1:1)
- Employee_ID – Address (1:1)
- Employee_ID – Session_number (1:1)
- Employee_ID – Depen_fname (1:M)
- Employee_ID – Depen_lname (1:M)
- Employee_ID – Depen_sex (1:M)
- Employee_ID – Depen_Birthdate (1:M)
- Employee_ID – Relationship (1:M)
- Dnumber – Dname (1:1)
- Dnum – Mgmt_ssn (1:1)
- Employee_ID – Session_number (1:M)
- Employee_ID – Animal_num (1:M)
- Employee_ID – Appt_date (M:M)
- Employee_ID – Appt_time (M:M)
- Employee_ID – Item_id_sold (M:M)
- Employee_ID – Train_status (1:1)
- Animal_ID – Species (1:1)
- Item_number – Item_Category (1:1)
- Animal_ID – Cat_fname (1:1)
- Animal_ID – Entry_date (1:1)
- Animal_ID – Cat_Breed (1:1)
- Animal_ID – Adoption_Status (1:1)
- Animal_ID – Ownership_Status (1:1)
- Animal_ID – Dog_fname (1:1)
- Animal_ID – Dog_Breed (1:1)
- Animal_ID – Bird_fname (1:1)
- Animal_ID – Type_of_Bird (1:1)
- Animal_ID – Animal_fname (1:1)
- Animal_ID – Animal_Type (1:1)
- Item_number – Type_of_Animal (1:1)
- Item_number – Amount_Remaining (1:M)
- Item_number – Item_Price (1:1)
- Item_number – Food_brand (1:1)
- Item_number – Cleaning_brand (1:1)
- Item_number – Toy_brand (1:1)

(IV) Functional Dependencies

User Manual

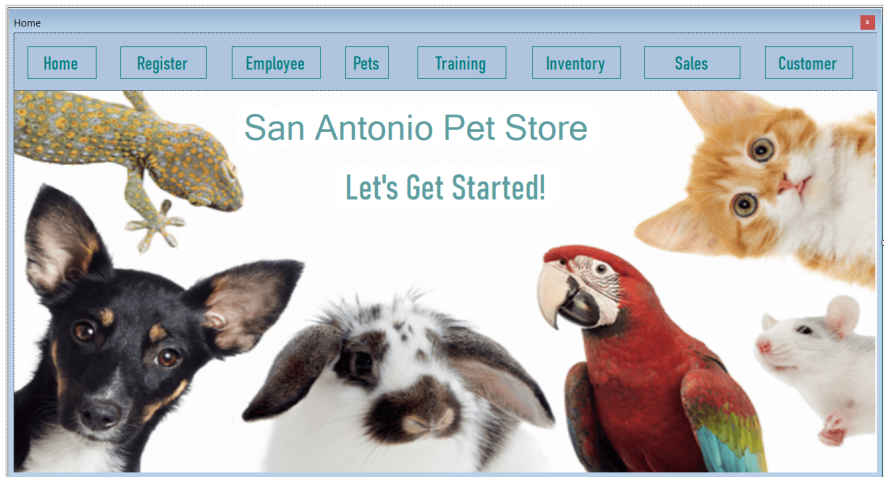
The following screenshots are from the actual database application. They all serve all purpose for the Pet Store and will be explained as they go through.

The first image is the Employee Login page. It requires verification as it allows the user to edit data in the database.



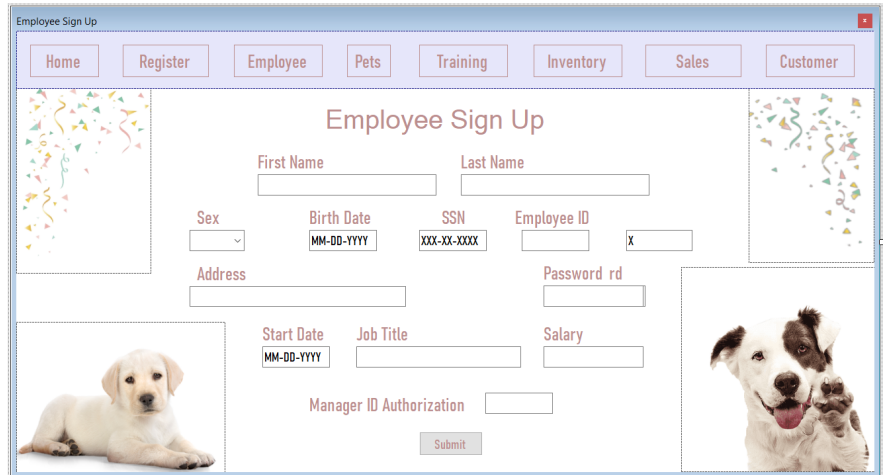
The screenshot shows the 'Employee Login' window. On the left, there is a decorative banner with confetti and the text 'Welcome to Your San Antonio Pet Store!' above a photo of a dog and a cat. On the right, the login form includes fields for 'Employee ID:', 'Password:', and 'Re-enter Password:', followed by a 'Login' button.

Once the employee or manager logs in, they are welcomed by the home page. The Home page holds a directory to any of the other forms.



The screenshot shows the 'Home' page of the San Antonio Pet Store. It features a navigation bar with buttons for 'Home', 'Register', 'Employee', 'Pets', 'Training', 'Inventory', 'Sales', and 'Customer'. Below the navigation bar is a large banner with the text 'San Antonio Pet Store' and 'Let's Get Started!' surrounded by images of various pets: a lizard, a dog, a rabbit, a parrot, a cat, and a mouse.

Managers are also allowed to create new employee profiles, as long as they hold a manager number for verification and a new employee number to create the profile under.



The screenshot shows the 'Employee Sign Up' window. It includes a navigation bar with buttons for 'Home', 'Register', 'Employee', 'Pets', 'Training', 'Inventory', 'Sales', and 'Customer'. The main form area is titled 'Employee Sign Up' and contains fields for 'First Name', 'Last Name', 'Sex', 'Birth Date', 'SSN', 'Employee ID', 'Address', 'Password', 'Start Date', 'Job Title', 'Salary', and 'Manager ID Authorization'. There are also decorative elements: confetti on the left and a photo of a dog on the right. A 'Submit' button is at the bottom.

Forms that follow the creation of employee profiles are the Employee Profile page itself, confidential to managers, and the Pet Store Employees Page which displays all employees without confidential information in a list view. The Employee Profile page displays all dependents for the employee queried, they can be added, deleted or edited. The San Antonio Pet Store Employees page broader and mostly serves see if the employee was added to the database, and whether they can now log themselves.

Profile

Retrieve Dependents

Return to Home

Return to Employees



Employee Profile

Employee First Name

Employee First Name

Employee Last Name

Employee Last Name

Employee Job Title

Employee Job Title

Employee Salary

Employee Salary

Employee ID

Employee ID

Employee Dependents:

First Name

Last Name

Sex

MM/DD/YY

Relationship

Add

Delete

First Name	Last Name	Sex	Birth Date	Relationship

Employee

Home

Register

Employee

Pets

Training

Inventory

Sales

Customer



San Antonio Pet Store Employees

Employee ID Chosen:

Manager ID Authorization:

See Profile

Delete Employee

Employee ID	First Name	Last Name	Sex	Birthdate	Address	Start Date	Job Title

Retrieve All



is to in

The application then holds several abilities for the employees themselves. Employees are able to add, edit, search and delete customer information. Customers are important to database since they bring in the pets, and so keeping their information in order is vital to the database.

CustomerInformation



Delete

Clear

Edit

Return to Menu

Customer Information

Search With Owner ID

Find

Owner ID

First Name

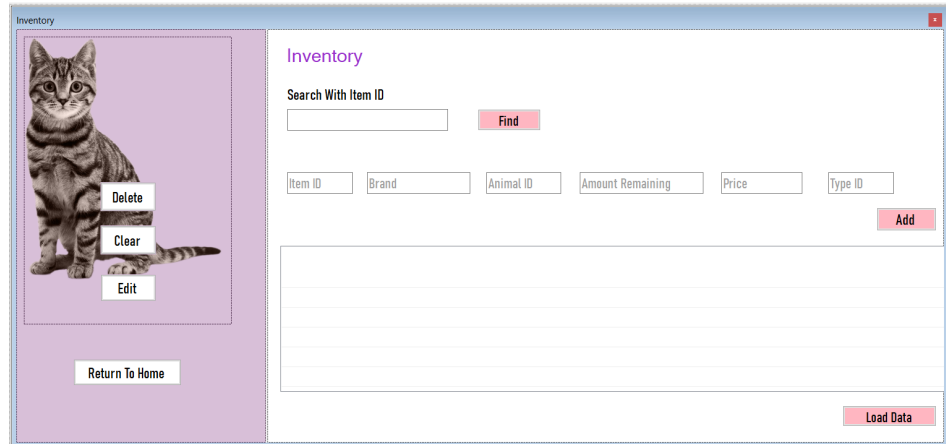
Last Name

Phone Number

Add

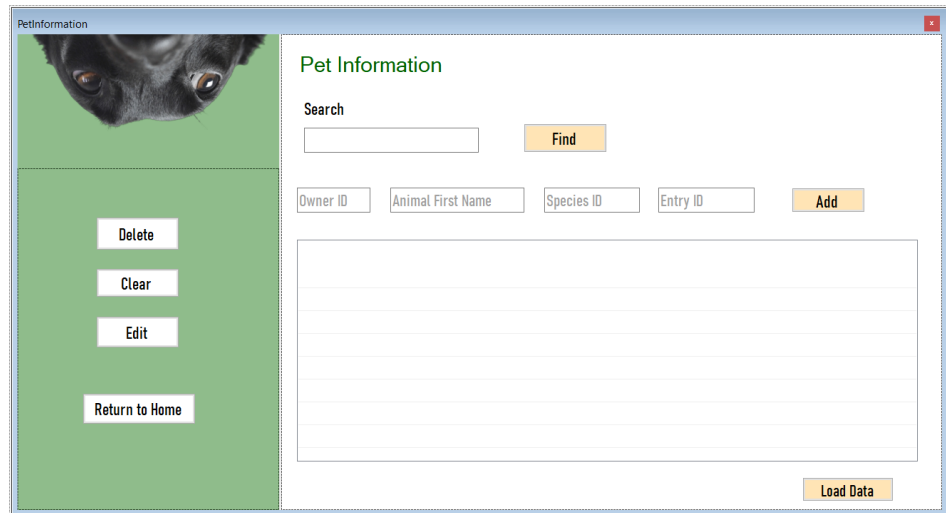
Load Data

The inventory page allows employees in charge of stock to update the system. The page allows users to add, delete and edit inventory in the database. They can also query a piece.



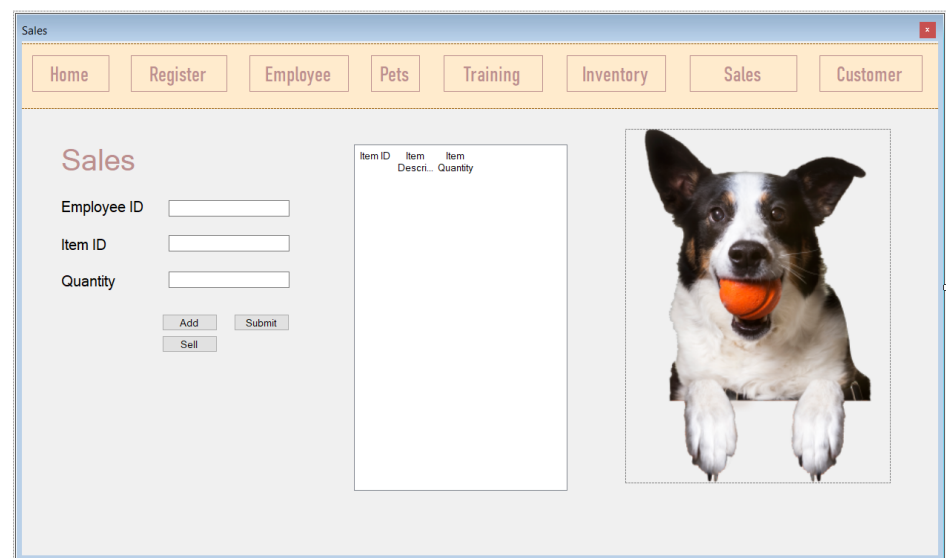
The Inventory page features a sidebar on the left with a cat image and buttons for 'Delete', 'Clear', 'Edit', and 'Return To Home'. The main area is titled 'Inventory' and includes a search bar labeled 'Search With Item ID' with a 'Find' button. Below the search bar are input fields for 'Item ID', 'Brand', 'Animal ID', 'Amount Remaining', 'Price', and 'Type ID', followed by an 'Add' button. A table with multiple rows is positioned below these fields, and a 'Load Data' button is at the bottom right.

The Pet Information page allows the employee to access pet information necessary for animals coming in to the training center, as well as those being sold. Animals have their own ID, as well as species ID and owner ID for those applicable. Employees are able to add new pets, as well as edit, delete and query existing customers.



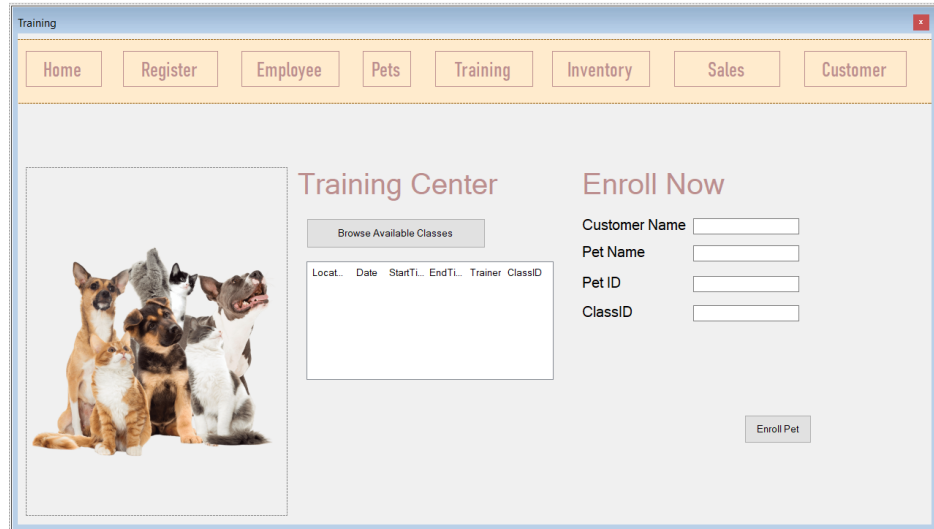
The Pet Information page has a sidebar on the left with a dog's face image and buttons for 'Delete', 'Clear', 'Edit', and 'Return to Home'. The main area is titled 'Pet Information' and includes a search bar with a 'Find' button. Below the search bar are input fields for 'Owner ID', 'Animal First Name', 'Species ID', and 'Entry ID', followed by an 'Add' button. A table with multiple rows is positioned below these fields, and a 'Load Data' button is at the bottom right.

As employees receive and complete different tasks, they will be able to see the Sales page. This form allows the user to edit the inventory through the sales portal. The user will enter their employee ID for records of sales, then the item ID and quantity sold. There is an option to add simply to make returns easier, but all should update the inventory list view equally, which is also visible to the right of the text boxes.



The Sales page features a top navigation bar with buttons for 'Home', 'Register', 'Employee', 'Pets', 'Training', 'Inventory', 'Sales', and 'Customer'. The main area is titled 'Sales' and includes input fields for 'Employee ID', 'Item ID', and 'Quantity'. Below these fields are buttons for 'Add', 'Submit', and 'Sell'. To the right of the input fields is a table with columns 'Item ID', 'Item Descr...', and 'Quantity'. Further right is a large image of a dog holding an orange ball.

Finally, employees and trainers, if available, are able to enroll the pets and customers for the training center. The Browse Available Classes button allows the user to see all current appointments in order to work around them. On the right, the user is able to enter information of the pet and customer in order to enlist them for a class, which then updates the list view.



The screenshot shows a web application window titled "Training". It features a navigation bar with buttons for Home, Register, Employee, Pets, Training, Inventory, Sales, and Customer. The main content area is divided into two sections: "Training Center" and "Enroll Now".

Training Center

Below the "Browse Available Classes" button is a table with the following headers: Locat..., Date, StartTi..., EndTi..., Trainer, and ClassID. The table body is currently empty.

Enroll Now

This section contains form fields for Customer Name, Pet Name, Pet ID, and ClassID, each followed by an input box. An "Enroll Pet" button is located at the bottom right of this section.

On the left side of the Training Center section, there is a placeholder image showing a group of dogs.

Summary and Conclusion

The team, composed of Alin Bocardo, Nadia Morrow, Katelyn Espinoza and Savannah Lopez would all like to thank Dr. Ayad Barsoum for the mentoring behind the project and ability to present at the St. Mary University's 2022 Computer Science Symposium. The project proved to be a great opportunity for this team to work together. The list of tasks below is the list included in the proposal, followed by concluding statements for the project:

Tasks:

- Requirements Design 9/12 – 9/15
- Project Proposal Write-up 9/15 – 9/20
- Database Design 9/21 – 9/29
- GUI Design 9/21 – 9/29
- Database Development 9/30 – 10/13
- GUI Development 9/30 – 10/13
- Integration of Database and GUI 10/14 – 10/20
- Create User Manual 10/21 – 10/28
- Write Summary and Conclusion 10/21 – 10/28
- Write Future Work Section 10/21 – 10/26
- Edit Sections from Project Proposal 10/21 – 10/25
- Integrate Sections for Final Documentation 10/31 – 11/2

All design phases were a learning experience for the team, both in working together and creating the base for what would become the database. The development of the Gui and creation of the database were divided tasks, which thanks to the ability of the team, were able to be integrated into a functional database. The final documentation was completed at an ease as it was broken down and allowed the team members to thoughtfully submit final renditions.

Future Work

Collegiate Level	At a collegiate level, the team hopes to fix bugs that were found during a presentation of the software. The application will run smoother after several issues have been assessed, and the database will be automated to a more precise level. The team hopes to better the interface of several of the forms and create an experience that users would enjoy. The team also plans to present the application at the St. Mary's University Fall 2022 Computer Science Symposium.
Real World Level	It is a developing possibility that customers will be able to access their pet's information as well, in accordance with scheduling and store updates. The team would hope to develop an interface for the customers to access as well, with more security features, portability and access to personal records. It is the teams' goal to ease the process of training a pet for the customer.

References

1. My Pet Naturally. “12 Pet Essentials Every Pet Owner Needs.” *My Pet Naturally*, 15 Nov. 2021, <https://mypetnaturally.com/2020/11/03/12-pet-essentials-every-pet-owner-needs/>.
2. “Pet Store.” *Wikipedia*, Wikimedia Foundation, 5 Mar. 2022, https://en.wikipedia.org/wiki/Pet_store.

Code Listing

Home.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Data.SqlClient;

namespace Pet_Store_App
{
    public partial class Home : Form
    {
        SqlConnection conn;
        public Home(SqlConnection c)
        {
            InitializeComponent();
            conn = c;
        }

        private void labelTitle_Click(object sender, EventArgs e)
        {
        }

        private void buttonHome_Click(object sender, EventArgs e)
        {
        }

        private void buttonRegister_Click(object sender, EventArgs e)
        {
            this.Hide();
            EmployeeCreation frm = new EmployeeCreation(conn);
            frm.ShowDialog();
        }

        private void buttonEmployee_Click(object sender, EventArgs e)
        {
            this.Hide();
            EmployeeProfile frm = new EmployeeProfile(conn);
            frm.ShowDialog();
        }

        private void buttonPets_Click(object sender, EventArgs e)
        {
            this.Hide();
            PetInformation frm = new PetInformation(conn);
        }
    }
}
```

```

        frm.ShowDialog();
    }

    private void buttonCustomer_Click(object sender, EventArgs e)
    {
        this.Hide();
        CustomerInformation frm = new CustomerInformation(conn);
        frm.ShowDialog();
    }

    private void buttonInventory_Click(object sender, EventArgs e)
    {
        this.Hide();
        Inventory frm = new Inventory(conn);
        frm.ShowDialog();
    }

    private void buttonDepartments_Click(object sender, EventArgs e)
    {
        this.Hide();
        Sales frm = new Sales(conn);
        frm.ShowDialog();
    }

    private void buttonTraining_Click(object sender, EventArgs e)
    {
        this.Hide();
        Training frm = new Training(conn);
        frm.ShowDialog();
    }

    private void pictureBox2_Click(object sender, EventArgs e)
    {
    }

    private void Home_Load(object sender, EventArgs e)
    {
    }
}

```

EmployeeProfile.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

using System.Data.SqlClient;

namespace Pet_Store_App
{
    public partial class EmployeeProfile : Form
    {
        SqlConnection conn;
        public EmployeeProfile(SqlConnection c)
        {
            InitializeComponent();
            conn = c;
        }

        private void labelTitle_Click(object sender, EventArgs e)
        {
        }

        private void listViewEmployees_SelectedIndexChanged(object sender, EventArgs e)
        {
        }

        private void labelName_Click(object sender, EventArgs e)
        {
        }

        private void buttonHome_Click(object sender, EventArgs e)
        {
            this.Hide();
            Home frm = new Home(conn);
            frm.ShowDialog();
        }

        private void buttonRegister_Click(object sender, EventArgs e)
        {
            this.Hide();
            EmployeeCreation frm = new EmployeeCreation(conn);
            frm.ShowDialog();
        }

        private void buttonCustomer_Click(object sender, EventArgs e)
        {
            this.Hide();
            CustomerInformation frm = new CustomerInformation(conn);
            frm.ShowDialog();
        }

        private void buttonPets_Click(object sender, EventArgs e)
        {
            this.Hide();
            PetInformation frm = new PetInformation(conn);
            frm.ShowDialog();
        }

        //see employee profile
        private void buttonProfile_Click(object sender, EventArgs e)

```

```

{
    try
    {
        String query = "SELECT employee_id FROM Employee WHERE EXISTS(SELECT * FROM
Department WHERE Ssn = Mgmt_ssn)";
        String query2 = "SELECT employee_id FROM Employee";
        SqlDataAdapter sda = new SqlDataAdapter(query, conn);
        SqlDataAdapter adapter = new SqlDataAdapter(query2, conn);

        DataTable dtable = new DataTable();
        DataTable dta = new DataTable();

        sda.Fill(dtable);
        adapter.Fill(dta);

        int manageID;
        manageID = Int32.Parse(textBoxManagerConfirm.Text);

        if (dtable.Rows.Count > 0)
        {
            DataRow[] foundID = dtable.Select("employee_id = " + manageID + "");
            //verify manager authorization
            if (foundID.Length != 0)
            {
                int IDchosen;
                string text = textBoxIDChosen.Text;
                IDchosen = Int32.Parse(text);

                DataRow[] foundEmp = dta.Select("employee_id = " + IDchosen + "");
                //verify employee id
                if (foundEmp.Length != 0)
                {
                    this.Hide();
                    Profile frm = new Profile(conn, IDchosen);
                    frm.ShowDialog();
                }
                else
                {
                    MessageBox.Show("Employee ID does not exist.", "Error", MessageBoxButtons.OK);
                }
            }
            else
            {
                MessageBox.Show("Manager ID does not exist.", "Error", MessageBoxButtons.OK);
            }
        }

    }
    catch
    {
        MessageBox.Show("Error.", "Error", MessageBoxButtons.OK);
    }
    finally

```

```

        {
            conn.Close();
        }

    }

    private void buttonInventory_Click(object sender, EventArgs e)
    {
        this.Hide();
        Inventory frm = new Inventory(conn);
        frm.ShowDialog();
    }

    private void buttonDepartments_Click(object sender, EventArgs e)
    {
        this.Hide();
        Sales frm = new Sales(conn);
        frm.ShowDialog();
    }

    private void buttonTraining_Click(object sender, EventArgs e)
    {
        this.Hide();
        Training frm = new Training(conn);
        frm.ShowDialog();
    }

    private void EmployeeProfile_Load(object sender, EventArgs e)
    {

    }

    //fills table with employees
    private void buttonRetrieve_Click(object sender, EventArgs e)
    {

        String q= @"SELECT Employee.employee_id, Fname, Lname, Sex, Birthdate,Address, start_date
        ,job_title,salary
        FROM Employee JOIN Employee_Job_Info ON Employee.employee_id =
        Employee_Job_Info.employee_id ";
        SqlDataAdapter brandnew = new SqlDataAdapter(q, conn);
        DataTable table = new DataTable();
        brandnew.Fill(table);
        listViewEmployees.Items.Clear();

        foreach (DataRow row in table.Rows)
        {
            ListViewItem item = new ListViewItem(row[0].ToString());
            for (int i = 1; i < table.Columns.Count; i++)
            {
                item.SubItems.Add(row[i].ToString());
            }
            listViewEmployees.Items.Add(item);
        }
    }

```

```

    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {

    }

    private void buttonDelete_Click(object sender, EventArgs e)
    {

        try
        {

            String query = "SELECT employee_id FROM Employee WHERE EXISTS(SELECT * FROM
Department WHERE Ssn = Mgmt_ssn)";
            String query2 = "SELECT employee_id FROM Employee";
            SqlDataAdapter sda = new SqlDataAdapter(query, conn);
            SqlDataAdapter adapter = new SqlDataAdapter(query2, conn);

            DataTable dtable = new DataTable();
            DataTable dta = new DataTable();

            sda.Fill(dtable);
            adapter.Fill(dta);

            int manageID;
            manageID = Int32.Parse(textBoxManagerConfirm.Text);

            if (dtable.Rows.Count > 0)
            {
                DataRow[] foundID = dtable.Select("employee_id = " + manageID + "");
                //verify manager authorization
                if (foundID.Length != 0)
                {
                    int IDchosen;
                    string t = textBoxIDChosen.Text;
                    IDchosen = Int32.Parse(t);

                    DataRow[] foundEmp = dta.Select("employee_id = " + IDchosen + "");
                    //verify employee id
                    if(foundEmp.Length != 0)
                    {
                        DialogResult c = new DialogResult();
                        c = MessageBox.Show("Are you sure you want to delete Employee ID: " + IDchosen + " from
the system?", "Confirmation", MessageBoxButtons.YesNo);
                        if (c == DialogResult.Yes)
                        {
                            String query3 = @"DELETE FROM Employee
WHERE employee_id = " + IDchosen;
                            SqlCommand cmd;

                            conn.Open();
                            cmd = new SqlCommand(query3, conn);
                            cmd.ExecuteNonQuery();
                            MessageBox.Show("Employee ID: " + IDchosen + " has been deleted.", "Confirmation",
MessageBoxButtons.OK);
                            conn.Close();

```

```

        }
        else
        {
            return;
        }
    }
    else
    {
        MessageBox.Show("Employee ID does not exist.", "Error", MessageBoxButtons.OK);
    }

}
else
{
    MessageBox.Show("Manager ID does not exist.", "Error", MessageBoxButtons.OK);
}

}

}
catch
{
    MessageBox.Show("Error.", "Error", MessageBoxButtons.OK);
}

}

private void textBox1_TextChanged_1(object sender, EventArgs e)
{
}

}
}
}

```

EmployeeLogin.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Data.SqlClient;

using System.Data.SqlClient;

namespace Pet_Store_App
{
    public partial class Form1 : Form

```

```

{
    SqlConnection conn;

    public Form1(SqlConnection c)
    {
        SqlConnection sqlConnection1 = new SqlConnection(@"Data Source=KATELYN;Initial
        Catalog=petStore;Integrated Security=True");
        public Form1()
        {
            InitializeComponent();
            conn = c;
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }

        private void label2_Click(object sender, EventArgs e)
        {
        }

        private void label3_Click(object sender, EventArgs e)
        {
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
        }

        private void button1Login_Click(object sender, EventArgs e)
        {
            Home frm = new Home(conn);
            frm.ShowDialog();
            String id, password, rePassword;

            id = textBoxEmployeeID.Text;
            password = textBoxPassword.Text;
            rePassword = textBox1.Text;

            try
            {
                String query = "SELECT * FROM EMPLOYEE WHERE Employee_ID='" +
                textBoxEmployeeID.Text + "' AND Password='" + textBoxPassword.Text + "'";
                SqlDataAdapter sda = new SqlDataAdapter(query, sqlConnection1);

                DataTable dtable = new DataTable();
                sda.Fill(dtable);

                if(dtable.Rows.Count > 0)

```

```

        {
            if (password != rePassword)
            {
                MessageBox.Show("Passwords do not match", "Error", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            }
            else
            {
                id = textBoxEmployeeID.Text;
                password = textBoxPassword.Text;

                //page that needed to be load next
                Home frm = new Home(sqlConnection1);
                frm.ShowDialog();
                this.Hide();
            }
        }
    }
    else
    {
        MessageBox.Show("Invalid login details", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        textBoxEmployeeID.Clear();
        textBoxPassword.Clear();

        //to focus username
        textBoxEmployeeID.Focus();
    }
}
}
catch
{
    MessageBox.Show("Error");
}
finally
{
    {
        sqlConnection1.Close();
    }
}
}

private void textBoxEmployeeID_TextChanged(object sender, EventArgs e)
{
}

private void textBox1_TextChanged_1(object sender, EventArgs e)
{
}
}
}
}

```

EmployeeCreation.cs

```
using System;
```

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Data.SqlClient;

namespace Pet_Store_App
{
    public partial class EmployeeCreation : Form
    {
        SqlConnection conn;
        public EmployeeCreation(SqlConnection c)
        {
            InitializeComponent();
            conn = c;
        }

        private void labelName_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void labelSex_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click_1(object sender, EventArgs e)
        {

        }

        private void textBox4_TextChanged(object sender, EventArgs e)
        {

        }

        private void Form2_Load(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {

        }

        private void buttonEmployee_Click(object sender, EventArgs e)
```

```

    {
        this.Hide();
        EmployeeProfile frm = new EmployeeProfile(conn);
        frm.ShowDialog();
    }

    private void buttonHome_Click(object sender, EventArgs e)
    {
        this.Hide();
        Home frm = new Home(conn);
        frm.ShowDialog();
    }

    private void buttonCustomer_Click(object sender, EventArgs e)
    {
        this.Hide();
        CustomerInformation frm = new CustomerInformation(conn);
        frm.ShowDialog();
    }

    private void buttonPets_Click(object sender, EventArgs e)
    {
        this.Hide();
        PetInformation frm = new PetInformation(conn);
        frm.ShowDialog(this);
    }

    private void buttonInventory_Click(object sender, EventArgs e)
    {
        this.Hide();
        Inventory frm = new Inventory(conn);
        frm.ShowDialog();
    }

    private void buttonTraining_Click(object sender, EventArgs e)
    {
        this.Hide();
        Training frm = new Training(c: conn);
        frm.ShowDialog();
    }

    private void buttonDepartments_Click(object sender, EventArgs e)
    {
        this.Hide();
        Sales frm = new Sales(conn);
        frm.ShowDialog();
    }

    public string conString = "Data Source=LAPTOP-MT7MRPHE; Initial Catalog=petStore; Integrated
Security=True";

    private void buttonSubmit_Click(object sender, EventArgs e)
    {
        String query = @"SELECT employee_id FROM Employee";

        SqlConnection conn = new SqlConnection(conString);
        SqlDataAdapter sda = new SqlDataAdapter(query, conn);
        DataTable dTable = new DataTable();

```

```

sda.Fill(dTable);
String ID = textBoxEmployeeID.Text;
DataRow[] findDuplicateID = dTable.Select("employee_ID = " + ID + "");
if (findDuplicateID.Length == 0)
{
    //table of all Dno to validate
    String query2 = @"SELECT Dnumber FROM Department";
    SqlDataAdapter adapter = new SqlDataAdapter(query2, conn);
    DataTable dta = new DataTable();
    adapter.Fill(dta);
    String dno = textBoxDno.Text;
    DataRow[] findDuplicatDno = dta.Select("Dnumber = " + dno + "");

    if (findDuplicatDno.Length != 0)
    {
        //validate manager id
        String query3 = @"SELECT employee_id FROM Employee WHERE EXISTS(SELECT* FROM
Department WHERE Ssn = Mgmt_ssn)";
        SqlDataAdapter dsa = new SqlDataAdapter(query3, conn);
        DataTable dta3 = new DataTable();
        dsa.Fill(dta3);
        String managerid = textBoxManagerID.Text;
        DataRow[] findManagerID = dta3.Select("employee_ID=" + managerid + "");
        if (findManagerID.Length != 0)
        {
            /*String q = @"INSERT INTO Employee(employee_id, Fname, Lname, Ssn, Birthdate, Sex,
Address, Dno, Password)
VALUES(" + textBoxEmployeeID.Text + ", " + textBoxFname.Text + ", " +
textBoxLname.Text + ", " + textBoxSSN.Text +
", " + textBoxBDate.Text + ", " + comboBoxSex.Text + ", " + textBoxAddress.Text +
", " + textBoxDno.Text + ", " + textBoxPass.Text + ")";
*/

            conn.Open();

            using (SqlCommand cmd = new SqlCommand())
            {
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;

                cmd.CommandText = query;
                cmd.Parameters.AddWithValue("@empId", textBoxEmployeeID.Text);
                cmd.Parameters.AddWithValue("@firstn", textBoxFname.Text);
                cmd.Parameters.AddWithValue("@lastn", textBoxLname.Text);
                cmd.Parameters.AddWithValue("@ssn", textBoxSSN.Text);
                cmd.Parameters.AddWithValue("@bdate", textBoxBDate.Text);
                cmd.Parameters.AddWithValue("@sex", comboBoxSex.Text);
                cmd.Parameters.AddWithValue("@addr", textBoxAddress.Text);
                cmd.Parameters.AddWithValue("@dnum", textBoxDno.Text);
                cmd.Parameters.AddWithValue("@passw", textBoxPass.Text);

                try { cmd.ExecuteNonQuery(); }
                catch (Exception) { }

                cmd.CommandText = query2;
                cmd.Parameters.AddWithValue("@empId", textBoxEmployeeID.Text);
                cmd.Parameters.AddWithValue("@sDate", textBox_startDate.Text);

```

```

cmd.Parameters.AddWithValue("@jobT", textBoxJobTitle.Text);
cmd.Parameters.AddWithValue("@sal", textBoxSalary.Text);

try { cmd.ExecuteNonQuery(); }
catch (Exception)
{ /* your error handling */

}

}

}
MessageBox.Show("New dependent for Employee ID: " + ID + " has been inserted.",
"Confirmation", MessageBoxButtons.OK);
conn.Close();
}
else
{
    MessageBox.Show("Not a valid manager ID.", "Error", MessageBoxButtons.OK);
}

}
else
{
    MessageBox.Show("Not a valid department number.", "Error", MessageBoxButtons.OK);
}
}
else
{
    MessageBox.Show("That Id is already in use.", "Error", MessageBoxButtons.OK);
}
/*
String query = @"INSERT INTO Employee (employee_id, Fname, Lname, Ssn, Birthdate, Sex, Address,
Dno, Password)
VALUES (@empId, @firstn, @lastn, @ssn, @bdate, @sex, @addr, @dnum, @passw)";
String query2 = @"INSERT INTO Employee_Job_Info (employee_id, start_date, job_title, salary)
VALUES (@empId, @sDate, @jobT, @sal)";

using (SqlConnection conn = new SqlConnection(conString))
{
    conn.Open();
    using (SqlCommand cmd = new SqlCommand())
    {
        cmd.Connection = conn;
        cmd.CommandType = CommandType.Text;

        cmd.CommandText = query;
        cmd.Parameters.AddWithValue("@empId", textBoxEmployeeID.Text);
        cmd.Parameters.AddWithValue("@firstn", textBoxFname.Text);
        cmd.Parameters.AddWithValue("@lastn", textBoxLname.Text);
        cmd.Parameters.AddWithValue("@ssn", textBoxSSN.Text);
        cmd.Parameters.AddWithValue("@bdate", textBoxBDate.Text);
        cmd.Parameters.AddWithValue("@sex", comboBoxSex.Text);
        cmd.Parameters.AddWithValue("@addr", textBoxAddress.Text);
        cmd.Parameters.AddWithValue("@dnum", textBoxDno.Text);
        cmd.Parameters.AddWithValue("@passw", textBoxPass.Text);

        try { cmd.ExecuteNonQuery(); }

```

```

        catch (Exception) { }

        cmd.CommandText = query2;
        cmd.Parameters.AddWithValue("@empId", textBoxEmployeeID.Text);
        cmd.Parameters.AddWithValue("@sDate", textBox_startDate.Text);
        cmd.Parameters.AddWithValue("@jobT", textBoxJobTitle.Text);
        cmd.Parameters.AddWithValue("@sal", textBoxSalary.Text);

        try { cmd.ExecuteNonQuery(); }
        catch (Exception) { /* your error handling */
    }

    // }
    //}

    //MessageBox.Show("New Employee" + textBoxEmployeeID.Text + "has been added.", "Confirmation",
    MessageBoxButtons.OK);
    //conn.Close();

    }
}

```

CustomerInformation.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Data.SqlClient;

namespace Pet_Store_App
{
    public partial class CustomerInformation : Form
    {
        SqlConnection conn;

        public CustomerInformation(SqlConnection c)
        {
            InitializeComponent();
            conn = c;
        }

        private void CustomerInformation_Load(object sender, EventArgs e)
        {
            listView_customerInfo.GridLines = true;
            listView_customerInfo.View = View.Details;

            listView_customerInfo.Columns.Add("Owner ID", 180);

```

```

listView_customerInfo.Columns.Add("First Name", 200);
listView_customerInfo.Columns.Add("Last Name", 200);
listView_customerInfo.Columns.Add("Phone Number", 230);

}

public string conString = "Data Source=LAPTOP-MT7MRPHE; Initial Catalog=petStore; Integrated
Security=True";

private void add_button_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(conString);
    String query = @"INSERT INTO Customer(owner_id, fname, lname, phone_number)
VALUES ('" + textBox_ownerId.Text + "', '" + textBox_firstName.Text + "', '" +
textBox_lastName.Text + "', '" + textBox_phoneNumber.Text + "')";
    SqlCommand cmd;

    conn.Open();
    cmd = new SqlCommand(query, conn);
    cmd.ExecuteNonQuery();
    MessageBox.Show("New Customer" + textBox_ownerId.Text + "has been added.", "Confirmation",
    MessageBoxButtons.OK);
    conn.Close();

}

private void button_returnToMenu_Click(object sender, EventArgs e)
{
    this.Hide();
    Home frm = new Home(conn);
    frm.ShowDialog();
}

private void button_loadData_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(conString);
    conn.Open();

    string sql = "Select owner_id, fname, lname, phone_number From Customer";

    SqlCommand cmd = new SqlCommand(sql, conn);
    SqlDataReader rd;
    rd = cmd.ExecuteReader();
    listView_customerInfo.Items.Clear();

    while (rd.Read())
    {
        ListViewItem lv = new ListViewItem(rd.GetDecimal(0).ToString());
        lv.SubItems.Add(rd.GetString(1).ToString());
        lv.SubItems.Add(rd.GetString(2).ToString());
        lv.SubItems.Add(rd.GetString(3).ToString());

        listView_customerInfo.Items.Add(lv);
    }
}

```

```

        rd.Close();
        cmd.Dispose();
        conn.Close();
    }

    private void textBox_ownerId_Enter(object sender, EventArgs e)
    {
        if (textBox_ownerId.Text == "Owner ID")
        {
            textBox_ownerId.Text = "";
            textBox_ownerId.ForeColor = Color.Black;
        }
    }

    private void textBox_ownerId_Leave(object sender, EventArgs e)
    {
        if (textBox_ownerId.Text == "")
        {
            textBox_ownerId.Text = "Owner ID";
            textBox_ownerId.ForeColor = Color.Silver;
        }
    }

    private void textBox_firstName_Enter(object sender, EventArgs e)
    {
        if (textBox_firstName.Text == "First Name")
        {
            textBox_firstName.Text = "";
            textBox_firstName.ForeColor = Color.Black;
        }
    }

    private void textBox_firstName_Leave(object sender, EventArgs e)
    {
        if (textBox_firstName.Text == "")
        {
            textBox_firstName.Text = "First Name";
            textBox_firstName.ForeColor = Color.Silver;
        }
    }

    private void textBox_lastName_Enter(object sender, EventArgs e)
    {
        if (textBox_lastName.Text == "Last Name")
        {
            textBox_lastName.Text = "";
            textBox_lastName.ForeColor = Color.Black;
        }
    }

    private void textBox_lastName_Leave(object sender, EventArgs e)
    {
        if (textBox_lastName.Text == "")
        {
            textBox_lastName.Text = "Last Name";
            textBox_lastName.ForeColor = Color.Silver;
        }
    }

```

```

private void textBox_phoneNumber_Enter(object sender, EventArgs e)
{
    if (textBox_phoneNumber.Text == "Phone Number")
    {
        textBox_phoneNumber.Text = "";
        textBox_phoneNumber.ForeColor = Color.Black;
    }
}

private void textBox_phoneNumber_Leave(object sender, EventArgs e)
{
    if (textBox_phoneNumber.Text == "")
    {
        textBox_phoneNumber.Text = "Phone Number";
        textBox_phoneNumber.ForeColor = Color.Silver;
    }
}

private void textBox_firstName_TextChanged(object sender, EventArgs e)
{
}

private void button_delete_Click(object sender, EventArgs e)
{
    if (listView_customerInfo.SelectedItems.Count > 0)
    {
        listView_customerInfo.SelectedItems[0].SubItems[0].Text = textBox_ownerId.Text;
        listView_customerInfo.SelectedItems[0].SubItems[1].Text = textBox_firstName.Text;
        listView_customerInfo.SelectedItems[0].SubItems[2].Text = textBox_lastName.Text;
        listView_customerInfo.SelectedItems[0].SubItems[3].Text = textBox_phoneNumber.Text;
    }

    SqlConnection conn = new SqlConnection(conString);

    SqlCommand cmd = conn.CreateCommand();

    conn.Open();

    cmd.CommandText = "Delete From Customer Where owner_id = @owner_Id";
    cmd.Parameters.AddWithValue("@owner_Id", textBox_ownerId.Text);
    cmd.ExecuteNonQuery();

    MessageBox.Show("Customer" + textBox_ownerId.Text + "has been deleted.", "Confirmation",
    MessageBoxButtons.OK);
    conn.Close();
}

private void clear_button_Click(object sender, EventArgs e)
{
    textBox_ownerId.Text = "";
    textBox_firstName.Text = "";
    textBox_lastName.Text = "";
    textBox_phoneNumber.Text = "";
}

private void listView_customerInfo_Click(object sender, EventArgs e)

```

```

{
    if (listView_customerInfo.SelectedItems.Count > 0)
    {
        textBox_ownerId.Text = listView_customerInfo.SelectedItems[0].SubItems[0].Text;
        textBox_firstName.Text = listView_customerInfo.SelectedItems[0].SubItems[1].Text;
        textBox_lastName.Text = listView_customerInfo.SelectedItems[0].SubItems[2].Text;
        textBox_phoneNumber.Text = listView_customerInfo.SelectedItems[0].SubItems[3].Text;
    }
}

private void edit_button_Click(object sender, EventArgs e)
{
    if (listView_customerInfo.SelectedItems.Count > 0)
    {
        listView_customerInfo.SelectedItems[0].SubItems[0].Text = textBox_ownerId.Text;
        listView_customerInfo.SelectedItems[0].SubItems[1].Text = textBox_firstName.Text;
        listView_customerInfo.SelectedItems[0].SubItems[2].Text = textBox_lastName.Text;
        listView_customerInfo.SelectedItems[0].SubItems[3].Text = textBox_phoneNumber.Text;
    }

    SqlConnection conn = new SqlConnection(conString);

    SqlCommand cmd = conn.CreateCommand();

    conn.Open();

    cmd.CommandText = "Update Customer set fname = @firstName, lname = @lastName, phone_number = @phoneNumber Where owner_id = @owner_Id";
    cmd.Parameters.AddWithValue("@owner_Id", textBox_ownerId.Text);
    cmd.Parameters.AddWithValue("@firstName", textBox_firstName.Text);
    cmd.Parameters.AddWithValue("@lastName", textBox_lastName.Text);
    cmd.Parameters.AddWithValue("@phoneNumber", textBox_phoneNumber.Text);
    cmd.ExecuteNonQuery();

    MessageBox.Show("Customer" + textBox_ownerId.Text + "has been updated.", "Confirmation",
    MessageBoxButtons.OK);
    conn.Close();
}

private void button_find_Click(object sender, EventArgs e)
{
    listView_customerInfo.SelectedItems.Clear();

    int i;
    for (i = 0; i < listView_customerInfo.Items.Count - 1; i++)
    {
        if (listView_customerInfo.Items[i].SubItems[0].Text == textBox_search.Text)
        {
            listView_customerInfo.Items[i].Selected = true;
            textBox_ownerId.Text = listView_customerInfo.SelectedItems[0].SubItems[0].Text;
            textBox_firstName.Text = listView_customerInfo.SelectedItems[0].SubItems[1].Text;
            textBox_lastName.Text = listView_customerInfo.SelectedItems[0].SubItems[2].Text;
            textBox_phoneNumber.Text = listView_customerInfo.SelectedItems[0].SubItems[3].Text;
        }
    }

    listView_customerInfo.Focus();
}

```

```

    }

    private void splitContainer1_Panel2_Paint(object sender, PaintEventArgs e)
    {

    }

}

}

```

Inventory.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Data.SqlClient;

namespace Pet_Store_App
{
    public partial class Inventory : Form
    {
        SqlConnection conn;

        public Inventory(SqlConnection c)
        {
            InitializeComponent();
            conn = c;
        }

        private void button_returnToHome_Click(object sender, EventArgs e)
        {
            this.Hide();
            Home frm = new Home(conn);
            frm.ShowDialog();
        }

        private void Inventory_Load(object sender, EventArgs e)
        {
            listView_inventory.GridLines = true;
            listView_inventory.View = View.Details;

            listView_inventory.Columns.Add("Item ID", 140);
            listView_inventory.Columns.Add("Brand", 200);
            listView_inventory.Columns.Add("Animal ID", 180);
            listView_inventory.Columns.Add("Amount Remaining", 300);
            listView_inventory.Columns.Add("Price", 120);
            listView_inventory.Columns.Add("Type ID", 140);
        }
    }
}

```

```

    }

    public string conString = "Data Source=LAPTOP-MT7MRPHE; Initial Catalog=petStore; Integrated
Security=True";

    private void button_add_Click(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection(conString);
        String query = @"INSERT INTO Inventory(item_id, brand, animal_id, amount_remaining, price, type_id)
VALUES ('" + textBox_itemID.Text + "', '" + textBox_brand.Text + "', '" +
textBox_animalID.Text + "', '" + textBox_amountRemaining.Text + "', '" +
+ textBox_price.Text + "', '" + textBox_typeID.Text + "')";
        SqlCommand cmd;

        conn.Open();
        cmd = new SqlCommand(query, conn);
        cmd.ExecuteNonQuery();
        MessageBox.Show("New Item" + textBox_itemID.Text + "has been added.", "Confirmation",
MessageBoxButtons.OK);
        conn.Close();
    }

    private void button_loadData_Click(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection(conString);
        conn.Open();

        string sql = "Select item_id, brand, animal_id, amount_remaining, price, type_id From Inventory";

        SqlCommand cmd = new SqlCommand(sql, conn);
        SqlDataReader rd;
        rd = cmd.ExecuteReader();
        listView_inventory.Items.Clear();

        while (rd.Read())
        {
            ListViewItem lv = new ListViewItem(rd.GetDecimal(0).ToString());
            lv.SubItems.Add(rd.GetString(1).ToString());
            lv.SubItems.Add(rd.GetDecimal(2).ToString());
            lv.SubItems.Add(rd.GetDecimal(3).ToString());
            lv.SubItems.Add(rd.GetDecimal(4).ToString());
            lv.SubItems.Add(rd.GetDecimal(5).ToString());

            listView_inventory.Items.Add(lv);
        }

        rd.Close();
        cmd.Dispose();
        conn.Close();
    }

    private void textBox_itemID_Enter(object sender, EventArgs e)
    {
        if (textBox_itemID.Text == "Item ID")
        {
            textBox_itemID.Text = "";
            textBox_itemID.ForeColor = Color.Black;
        }
    }

```

```

    }
}

private void textBox_itemID_Leave(object sender, EventArgs e)
{
    if (textBox_itemID.Text == "")
    {
        textBox_itemID.Text = "Owner ID";
        textBox_itemID.ForeColor = Color.Silver;
    }
}

private void textBox_brand_Enter(object sender, EventArgs e)
{
    if (textBox_brand.Text == "Brand")
    {
        textBox_brand.Text = "";
        textBox_brand.ForeColor = Color.Black;
    }
}

private void textBox_brand_Leave(object sender, EventArgs e)
{
    if (textBox_itemID.Text == "")
    {
        textBox_itemID.Text = "Brand";
        textBox_itemID.ForeColor = Color.Silver;
    }
}

private void textBox_animalID_Enter(object sender, EventArgs e)
{
    if (textBox_animalID.Text == "Animal ID")
    {
        textBox_animalID.Text = "";
        textBox_animalID.ForeColor = Color.Black;
    }
}

private void textBox_animalID_Leave(object sender, EventArgs e)
{
    if (textBox_animalID.Text == "")
    {
        textBox_animalID.Text = "Animal ID";
        textBox_animalID.ForeColor = Color.Silver;
    }
}

private void textBox_amountRemaining_Enter(object sender, EventArgs e)
{
    if (textBox_amountRemaining.Text == "Amount Remaining")
    {
        textBox_amountRemaining.Text = "";
        textBox_amountRemaining.ForeColor = Color.Black;
    }
}

private void textBox_amountRemaining_Leave(object sender, EventArgs e)

```

```

    {
        if (textBox_amountRemaining.Text == "")
        {
            textBox_amountRemaining.Text = "Amount Remaining";
            textBox_amountRemaining.ForeColor = Color.Silver;
        }
    }

private void textBox_price_Enter(object sender, EventArgs e)
{
    if (textBox_price.Text == "Price")
    {
        textBox_price.Text = "";
        textBox_price.ForeColor = Color.Black;
    }
}

private void textBox_price_Leave(object sender, EventArgs e)
{
    if (textBox_price.Text == "")
    {
        textBox_price.Text = "Price";
        textBox_price.ForeColor = Color.Silver;
    }
}

private void textBox_typeID_Enter(object sender, EventArgs e)
{
    if (textBox_typeID.Text == "Type ID")
    {
        textBox_typeID.Text = "";
        textBox_typeID.ForeColor = Color.Black;
    }
}

private void textBox_typeID_Leave(object sender, EventArgs e)
{
    if (textBox_typeID.Text == "")
    {
        textBox_typeID.Text = "Type ID";
        textBox_typeID.ForeColor = Color.Silver;
    }
}

private void button_delete_Click(object sender, EventArgs e)
{
    if (listView_inventory.SelectedItems.Count > 0)
    {
        listView_inventory.SelectedItems[0].SubItems[0].Text = textBox_itemID.Text;
        listView_inventory.SelectedItems[0].SubItems[1].Text = textBox_brand.Text;
        listView_inventory.SelectedItems[0].SubItems[2].Text = textBox_animalID.Text;
        listView_inventory.SelectedItems[0].SubItems[3].Text = textBox_amountRemaining.Text;
        listView_inventory.SelectedItems[0].SubItems[4].Text = textBox_price.Text;
        listView_inventory.SelectedItems[0].SubItems[5].Text = textBox_typeID.Text;
    }
}

SqlConnection conn = new SqlConnection(conString);

```

```

SqlCommand cmd = conn.CreateCommand();

conn.Open();

cmd.CommandText = "Delete From Inventory Where item_id = @item_Id";
cmd.Parameters.AddWithValue("@item_Id", textBox_itemID.Text);
cmd.ExecuteNonQuery();

MessageBox.Show("Item" + textBox_itemID.Text + "has been deleted.", "Confirmation",
MessageBoxButtons.OK);
conn.Close();
}

private void button_clear_Click(object sender, EventArgs e)
{
    textBox_itemID.Text = "";
    textBox_brand.Text = "";
    textBox_animalID.Text = "";
    textBox_amountRemaining.Text = "";
    textBox_price.Text = "";
    textBox_typeID.Text = "";
}

private void listView_inventory_Click(object sender, EventArgs e)
{
    if (listView_inventory.SelectedItems.Count > 0)
    {
        textBox_itemID.Text = listView_inventory.SelectedItems[0].SubItems[0].Text;
        textBox_brand.Text = listView_inventory.SelectedItems[0].SubItems[1].Text;
        textBox_animalID.Text = listView_inventory.SelectedItems[0].SubItems[2].Text;
        textBox_amountRemaining.Text = listView_inventory.SelectedItems[0].SubItems[3].Text;
        textBox_price.Text = listView_inventory.SelectedItems[0].SubItems[4].Text;
        textBox_typeID.Text = listView_inventory.SelectedItems[0].SubItems[5].Text;
    }
}

private void button_edit_Click(object sender, EventArgs e)
{
    if (listView_inventory.SelectedItems.Count > 0)
    {
        listView_inventory.SelectedItems[0].SubItems[0].Text = textBox_itemID.Text;
        listView_inventory.SelectedItems[0].SubItems[1].Text = textBox_brand.Text;
        listView_inventory.SelectedItems[0].SubItems[2].Text = textBox_animalID.Text;
        listView_inventory.SelectedItems[0].SubItems[3].Text = textBox_amountRemaining.Text;
        listView_inventory.SelectedItems[0].SubItems[4].Text = textBox_price.Text;
        listView_inventory.SelectedItems[0].SubItems[5].Text = textBox_typeID.Text;
    }
}

SqlConnection conn = new SqlConnection(conString);

SqlCommand cmd = conn.CreateCommand();

conn.Open();

cmd.CommandText = "Update Inventory set item_id = @itemId, brand = @br, animal_id = @animalID,
amount_remaining = @amrem, price = @pri, type_id = @typeId Where item_id = @itemId";
cmd.Parameters.AddWithValue("@itemId", textBox_itemID.Text);

```

```

cmd.Parameters.AddWithValue("@br", textBox_brand.Text);
cmd.Parameters.AddWithValue("@animalID", textBox_animalID.Text);
cmd.Parameters.AddWithValue("@amrem", textBox_amountRemaining.Text);
cmd.Parameters.AddWithValue("@pri", textBox_price.Text);
cmd.Parameters.AddWithValue("@typeID", textBox_typeID.Text);

cmd.ExecuteNonQuery();

MessageBox.Show("Item" + textBox_itemID.Text + "has been updated.", "Confirmation",
MessageBoxButtons.OK);
conn.Close();
}

private void button_find_Click(object sender, EventArgs e)
{
    listView_inventory.SelectedItems.Clear();

    int i;
    for (i = 0; i < listView_inventory.Items.Count - 1; i++)
    {
        if (listView_inventory.Items[i].SubItems[0].Text == textBox_search.Text)
        {
            listView_inventory.Items[i].Selected = true;
            textBox_itemID.Text = listView_inventory.SelectedItems[0].SubItems[0].Text;
            textBox_brand.Text = listView_inventory.SelectedItems[0].SubItems[1].Text;
            textBox_animalID.Text = listView_inventory.SelectedItems[0].SubItems[2].Text;
            textBox_amountRemaining.Text = listView_inventory.SelectedItems[0].SubItems[3].Text;
            textBox_price.Text = listView_inventory.SelectedItems[0].SubItems[4].Text;
            textBox_typeID.Text = listView_inventory.SelectedItems[0].SubItems[5].Text;
        }
    }

    listView_inventory.Focus();
}
}
}

```

PetInformation.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Data.SqlClient;

namespace Pet_Store_App
{
    public partial class PetInformation : Form
    {
        SqlConnection conn;
    }
}

```

```

public PetInformation(SqlConnection c)
{
    InitializeComponent();
    conn = c;
}

private void button_returnToHome_Click(object sender, EventArgs e)
{
    this.Hide();
    Home frm = new Home(conn);
    frm.ShowDialog();
}

private void PetInformation_Load(object sender, EventArgs e)
{
    listView_pet.GridLines = true;
    listView_pet.View = View.Details;

    listView_pet.Columns.Add("Owner ID", 180);
    listView_pet.Columns.Add("Animal First Name", 300);
    listView_pet.Columns.Add("Species ID", 200);
    listView_pet.Columns.Add("Entry Date", 180);
}

public string conString = "Data Source=LAPTOP-MT7MRPHE; Initial Catalog=petStore; Integrated
Security=True";

private void textBox_ownerId_Enter(object sender, EventArgs e)
{
    if (textBox_ownerId.Text == "Owner ID")
    {
        textBox_ownerId.Text = "";
        textBox_ownerId.ForeColor = Color.Black;
    }
}

private void textBox_ownerId_Leave(object sender, EventArgs e)
{
    if (textBox_ownerId.Text == "")
    {
        textBox_ownerId.Text = "Owner ID";
        textBox_ownerId.ForeColor = Color.Silver;
    }
}

private void textBox_aFname_Enter(object sender, EventArgs e)
{
    if (textBox_aFname.Text == "Animal First Name")
    {
        textBox_aFname.Text = "";
        textBox_aFname.ForeColor = Color.Black;
    }
}

private void textBox_aFname_Leave(object sender, EventArgs e)
{

```

```

        if (textBox_aFname.Text == "")
        {
            textBox_aFname.Text = "Animal First Name";
            textBox_aFname.ForeColor = Color.Silver;
        }
    }

    private void textBox_speciesId_Enter(object sender, EventArgs e)
    {
        if (textBox_speciesId.Text == "Species ID")
        {
            textBox_speciesId.Text = "";
            textBox_speciesId.ForeColor = Color.Black;
        }
    }

    private void textBox_speciesId_Leave(object sender, EventArgs e)
    {
        if (textBox_speciesId.Text == "")
        {
            textBox_speciesId.Text = "Species ID";
            textBox_speciesId.ForeColor = Color.Silver;
        }
    }

    private void textBox_entryId_Enter(object sender, EventArgs e)
    {
        if (textBox_entryId.Text == "Entry ID")
        {
            textBox_entryId.Text = "";
            textBox_entryId.ForeColor = Color.Black;
        }
    }

    private void textBox_entryId_Leave(object sender, EventArgs e)
    {
        if (textBox_entryId.Text == "")
        {
            textBox_entryId.Text = "Entry ID";
            textBox_entryId.ForeColor = Color.Silver;
        }
    }

    private void button_add_Click(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection(conString);
        String query = @"INSERT INTO Pets(owner_id, animal_fname, species_id, entry_date)
            VALUES ('" + textBox_ownerId.Text + "', '" + textBox_aFname.Text + "', '" +
textBox_speciesId.Text + "', '" + textBox_entryId.Text + "')";
        SqlCommand cmd;

        conn.Open();
        cmd = new SqlCommand(query, conn);
        cmd.ExecuteNonQuery();
        MessageBox.Show("New Pet" + textBox_ownerId.Text + "has been added.", "Confirmation",
        MessageBoxButtons.OK);
        conn.Close();
    }

```

```

private void button_loadData_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(conString);
    conn.Open();

    string sql = "Select owner_id, animal_fname, species_id, entry_date From Pets";

    SqlCommand cmd = new SqlCommand(sql, conn);
    SqlDataReader rd;
    rd = cmd.ExecuteReader();
    listView_pet.Items.Clear();

    while (rd.Read())
    {
        ListViewItem lv = new ListViewItem(rd.GetDecimal(0).ToString());
        lv.SubItems.Add(rd.GetString(1).ToString());
        lv.SubItems.Add(rd.GetDecimal(2).ToString());
        lv.SubItems.Add(rd.GetString(3).ToString());

        listView_pet.Items.Add(lv);
    }

    rd.Close();
    cmd.Dispose();
    conn.Close();
}

private void button_delete_Click(object sender, EventArgs e)
{
    if (listView_pet.SelectedItems.Count > 0)
    {
        listView_pet.SelectedItems[0].SubItems[0].Text = textBox_ownerId.Text;
        listView_pet.SelectedItems[0].SubItems[1].Text = textBox_aFname.Text;
        listView_pet.SelectedItems[0].SubItems[2].Text = textBox_speciesId.Text;
        listView_pet.SelectedItems[0].SubItems[3].Text = textBox_entryId.Text;
    }

    SqlConnection conn = new SqlConnection(conString);

    SqlCommand cmd = conn.CreateCommand();

    conn.Open();

    cmd.CommandText = "Delete From Pets Where owner_id = @owner_Id and animal_fname = @animalFname";
    cmd.Parameters.AddWithValue("@owner_Id", textBox_ownerId.Text);
    cmd.Parameters.AddWithValue("@animalFname", textBox_aFname.Text);
    cmd.ExecuteNonQuery();

    MessageBox.Show("Pet" + textBox_ownerId.Text + "has been deleted.", "Confirmation",
    MessageBoxButtons.OK);
    conn.Close();
}

private void listView_pet_Click(object sender, EventArgs e)
{

```

```

        if (listView_pet.SelectedItems.Count > 0)
        {
            textBox_ownerId.Text = listView_pet.SelectedItems[0].SubItems[0].Text;
            textBox_aFname.Text = listView_pet.SelectedItems[0].SubItems[1].Text;
            textBox_speciesId.Text = listView_pet.SelectedItems[0].SubItems[2].Text;
            textBox_entryId.Text = listView_pet.SelectedItems[0].SubItems[3].Text;
        }
    }

    private void button_clear_Click(object sender, EventArgs e)
    {
        textBox_ownerId.Text = "";
        textBox_aFname.Text = "";
        textBox_speciesId.Text = "";
        textBox_entryId.Text = "";
    }

    private void button_edit_Click(object sender, EventArgs e)
    {
        if (listView_pet.SelectedItems.Count > 0)
        {
            listView_pet.SelectedItems[0].SubItems[0].Text = textBox_ownerId.Text;
            listView_pet.SelectedItems[0].SubItems[1].Text = textBox_aFname.Text;
            listView_pet.SelectedItems[0].SubItems[2].Text = textBox_speciesId.Text;
            listView_pet.SelectedItems[0].SubItems[3].Text = textBox_entryId.Text;
        }

        SqlConnection conn = new SqlConnection(conString);

        SqlCommand cmd = conn.CreateCommand();

        conn.Open();

        cmd.CommandText = "Update Pets set animal_fname = @animalFName, species_id = @speciesId,
entry_date = @entryD Where owner_id = @owner_Id";
        cmd.Parameters.AddWithValue("@owner_Id", textBox_ownerId.Text);
        cmd.Parameters.AddWithValue("@animalFname", textBox_aFname.Text);
        cmd.Parameters.AddWithValue("@speciesId", textBox_speciesId.Text);
        cmd.Parameters.AddWithValue("@entryD", textBox_entryId.Text);
        cmd.ExecuteNonQuery();

        MessageBox.Show("Pet" + textBox_ownerId.Text + "has been updated.", "Confirmation",
        MessageBoxButtons.OK);
        conn.Close();
    }

    private void button_find_Click(object sender, EventArgs e)
    {
        listView_pet.SelectedItems.Clear();

        int i;
        for (i = 0; i < listView_pet.Items.Count - 1; i++)
        {
            if (listView_pet.Items[i].SubItems[0].Text == textBox_search.Text)
            {
                listView_pet.Items[i].Selected = true;
                textBox_ownerId.Text = listView_pet.SelectedItems[0].SubItems[0].Text;
                textBox_aFname.Text = listView_pet.SelectedItems[0].SubItems[1].Text;
            }
        }
    }

```

```

        textBox_speciesId.Text = listView_pet.SelectedItems[0].SubItems[2].Text;
        textBox_entryId.Text = listView_pet.SelectedItems[0].SubItems[3].Text;
    }
}

listView_pet.Focus();
}
}
}

```

Profile.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Data.SqlClient;

namespace Pet_Store_App
{
    public partial class Profile : Form
    {
        SqlConnection conn;
        int ID;
        public Profile(SqlConnection con, int IDChosen)
        {
            InitializeComponent();
            ID = IDChosen;
            conn = con;
        }

        private void label_selectedRow_Click(object sender, EventArgs e)
        {
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
        }

        private void button_returnToHome_Click(object sender, EventArgs e)
        {
            this.Hide();
            Home frm = new Home(conn);
            frm.ShowDialog();
        }

        private void Profile_Load(object sender, EventArgs e)

```

```

{

//display Employee First Name
SqlCommand cmdLoadFname = conn.CreateCommand();
cmdLoadFname.CommandText = @"SELECT fname
                           FROM Employee
                           WHERE employee_ID = " + ID;
conn.Open();
SqlDataReader reader = cmdLoadFname.ExecuteReader();

while (reader.Read())
{
    labelFname.Text = reader.GetString(0);
}
reader.Close();
conn.Close();

//display Employee Last Name
SqlCommand cmdLoadLname = conn.CreateCommand();
cmdLoadLname.CommandText = @"SELECT lname
                           FROM Employee
                           WHERE employee_ID = " + ID;
conn.Open();
SqlDataReader reader1 = cmdLoadLname.ExecuteReader();

while (reader1.Read())
{
    labelLname.Text = reader1.GetString(0);
}
reader1.Close();
conn.Close ();

//display job title
SqlCommand cmdLoadJob = conn.CreateCommand();
cmdLoadJob.CommandText = @"SELECT job_title
                           FROM Employee_Job_Info
                           WHERE employee_ID = " + ID;
conn.Open ();
SqlDataReader reader2 = cmdLoadJob.ExecuteReader();

while (reader2.Read())
{
    labelJobs.Text = reader2.GetString(0);
}
reader2.Close();
conn.Close();

//display Salary
SqlCommand cmdLoadSalary = conn.CreateCommand();
cmdLoadSalary.CommandText = @"SELECT salary
                           FROM Employee_Job_Info
                           WHERE employee_ID = " + ID;
conn.Open();
SqlDataReader reader3 = cmdLoadSalary.ExecuteReader();

while (reader3.Read())
{

```

```

        labelSal.Text = reader3.GetDecimal(0).ToString();
    }
    reader3.Close();
    conn.Close();

    //display ID
    labelEmpID.Text = " " + ID;

}

private void listView1_SelectedIndexChanged(object sender, EventArgs e)
{

}

private void buttonRetrieveDepen_Click(object sender, EventArgs e)
{
    listViewDependents.Items.Clear();
    String query = @"SELECT  Fname, Lname, Sex, Birthdate, Relationship
                        FROM Dependent WHERE employee_id = " + ID + """;
    SqlDataAdapter sda = new SqlDataAdapter(query, conn);
    DataTable dtable = new DataTable();
    sda.Fill(dtable);

    foreach (DataRow row in dtable.Rows)
    {
        ListViewItem item = new ListViewItem(row[0].ToString());
        for (int i = 1; i < dtable.Columns.Count; i++)
        {
            item.SubItems.Add(row[i].ToString());
        }
        listViewDependents.Items.Add(item);
    }
}

private void labelFname_Click(object sender, EventArgs e)
{

}

private void textBoxLastName_TextChanged(object sender, EventArgs e)
{

}

private void labelLname_Click(object sender, EventArgs e)
{

}

private void label1_Click(object sender, EventArgs e)
{

}

```

```

private void buttonEmployees_Click(object sender, EventArgs e)
{
    this.Hide();
    EmployeeProfile frm = new EmployeeProfile(conn);
    frm.ShowDialog();
}

private void button_add_Click(object sender, EventArgs e)
{
    String query = @"INSERT INTO Dependent(employee_id, Fname, Lname, Sex, Birthdate, Relationship)
VALUES("+ID+", "" +textBoxDepen_fname.Text+"", ""+textBoxLName.Text+"",
""+comboBoxSex.Text+"", ""+textBoxBirthdate.Text+"", ""+textBoxRelationship.Text+"")";
    SqlCommand cmd;

    conn.Open();
    cmd= new SqlCommand(query,conn);
    cmd.ExecuteNonQuery();
    MessageBox.Show("New dependent for Employee ID: " + ID + " has been inserted.", "Confirmation",
    MessageBoxButtons.OK);
    conn.Close();
}

private void textBox3_TextChanged(object sender, EventArgs e)
{
}

private void buttonDelete_Click(object sender, EventArgs e)
{
    DialogResult choice = new DialogResult();
    choice = MessageBox.Show("Are you sure you want to delete " + textBoxDepen_fname.Text + " " +
textBoxLName.Text + " as a dependent?", "Confirmation", MessageBoxButtons.YesNo);
    if (choice == DialogResult.Yes)
    {
        String query = @"DELETE FROM Dependent
WHERE employee_id = " + ID + " AND fname = "" + textBoxDepen_fname.Text + """;
        SqlCommand cmd;

        conn.Open();
        cmd = new SqlCommand(query, conn);
        cmd.ExecuteNonQuery();
        MessageBox.Show("Dependent:" + textBoxDepen_fname.Text + " " + textBoxLName.Text + " for
Employee ID: " + ID + " has been deleted.", "Confirmation", MessageBoxButtons.OK);
        conn.Close();
    }
    else
    {
        return;
    }
}

private void splitContainer1_Panel2_Paint(object sender, PaintEventArgs e)
{
}

private void labelJobs_Click(object sender, EventArgs e)
{
}

```

```

    }
}
}

```

Sales.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Data.SqlClient;

namespace Pet_Store_App
{
    public partial class Sales : Form
    {
        SqlConnection conn;
        public Sales(SqlConnection c)
        {
            InitializeComponent();
            conn = c;
        }

        private void labelTitle_Click(object sender, EventArgs e)
        {
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }

        private void label2_Click(object sender, EventArgs e)
        {
        }

        private void button3_Click(object sender, EventArgs e)
        {
        }

        private void Sales_Load(object sender, EventArgs e)
        {
        }

        private void buttonHome_Click(object sender, EventArgs e)

```

```

    {
        this.Hide();
        Home frm = new Home(conn);
        frm.ShowDialog();
    }

    private void buttonRegister_Click(object sender, EventArgs e)
    {
        this.Hide();
        EmployeeCreation frm = new EmployeeCreation(conn);
        frm.ShowDialog();
    }

    private void buttonEmployee_Click(object sender, EventArgs e)
    {
        this.Hide();
        EmployeeProfile frm = new EmployeeProfile(conn);
        frm.ShowDialog();
    }

    private void buttonPets_Click(object sender, EventArgs e)
    {
        this.Hide();
        PetInformation frm = new PetInformation(conn);
        frm.ShowDialog();
    }

    private void buttonTraining_Click(object sender, EventArgs e)
    {
        this.Hide();
        Training frm = new Training(conn);
        frm.ShowDialog();
    }

    private void buttonInventory_Click(object sender, EventArgs e)
    {
        this.Hide();
        Inventory frm = new Inventory(conn);
        frm.ShowDialog();
    }

    private void buttonCustomer_Click(object sender, EventArgs e)
    {
        this.Hide();
        CustomerInformation frm = new CustomerInformation(conn);
        frm.ShowDialog();
    }

    private void buttonDepartments_Click(object sender, EventArgs e)
    {
    }
}

```

Training.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Data.SqlClient;

namespace Pet_Store_App
{
    public partial class Training : Form
    {
        SqlConnection conn;
        public Training(SqlConnection c)
        {
            InitializeComponent();
            conn = c;
        }

        private void labelTitle_Click(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
        }

        private void button2_Click(object sender, EventArgs e)
        {
        }

        private void listView1_SelectedIndexChanged(object sender, EventArgs e)
        {
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
        }

        private void label2_Click(object sender, EventArgs e)
        {
        }

        private void Training_Load(object sender, EventArgs e)
        {
        }
    }
}
```

```

private void buttonHome_Click(object sender, EventArgs e)
{
    this.Hide();
    Home frm = new Home(conn);
    frm.ShowDialog();
}

private void buttonRegister_Click(object sender, EventArgs e)
{
    this.Hide();
    EmployeeCreation frm = new EmployeeCreation(conn);
    frm.ShowDialog();
}

private void buttonEmployee_Click(object sender, EventArgs e)
{
    this.Hide();
    EmployeeProfile frm = new EmployeeProfile(conn);
    frm.ShowDialog();
}

private void buttonPets_Click(object sender, EventArgs e)
{
    this.Hide();
    PetInformation frm = new PetInformation(conn);
    frm.ShowDialog();
}

private void buttonInventory_Click(object sender, EventArgs e)
{
    this.Hide();
    Inventory frm = new Inventory(conn);
    frm.ShowDialog();
}

private void buttonDepartments_Click(object sender, EventArgs e)
{
    this.Hide();
    Sales frm = new Sales(conn);
    frm.ShowDialog();
}

private void buttonCustomer_Click(object sender, EventArgs e)
{
    this.Hide();
    CustomerInformation frm = new CustomerInformation(conn);
    frm.ShowDialog();
}
}
}

```