RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

PET MANAGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted by

M N Chandni	**230701059**
R U Amirthavarshini	**230701027**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

(AUTONOMOUS) THANDALAM

CHENNAI-602105
2024 - 2025

BONAFIDE CERTIFICATE

Certified that this project report "**Pet Management System**" is the Bonafide work of

"M N Chandni (230701059), R U Amirthavarshini (230701027)" who carried out the project work under my supervision.

Submitted for the Practical Examination held on ________________________

SIGNATURE

Mrs.B.Deepa
Professor (SS) CSE,
Rajalakshmi Engineering College
(Autonomous),
Thandalam, Chennai - 602 105

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The Pet Management System is a desktop-based application developed using Java Swing, JDBC, and Oracle Database, designed to streamline the management of pet records in veterinary clinics. The system provides an easy-to-use interface for clinic administrators to manage pet details, owner information, and associated records efficiently.

Key features include the ability to add, update, and delete pet records, including information such as pet ID, name, breed, age, owner name, and contact details. The system ensures secure data handling and offers real-time updates to the database using JDBC for seamless integration with Oracle Database.

The front-end leverages Java Swing for an intuitive graphical user interface, ensuring an organized and user-friendly experience. This system is a robust solution for veterinary clinics, demonstrating the practical application of Java technologies in streamlining pet management operations while ensuring reliability and ease of use.

I. INTRODUCTION

In today's fast-paced world, managing pet information efficiently has become a critical aspect of veterinary clinic operations. The Pet Management System is a desktop-based application designed to provide a comprehensive solution for managing pet records and streamlining clinic workflows.

Developed using Java technologies, including Swing for the user interface and JDBC for database connectivity, this system offers a secure and user-friendly platform for veterinary administrators. It features two main modules: an administrative panel for managing pet details and owner information, and a data visualization interface for easy access and updates to records.

With a focus on simplicity and functionality, the system utilizes Java Swing for an intuitive front-end, ensuring responsiveness and ease of use. By integrating Oracle Database, it ensures efficient and reliable data management, supporting operations such as adding, updating, and deleting pet records.

This project showcases the practical application of Java development and database integration to address real-world challenges in pet management, demonstrating its relevance and effectiveness in veterinary practices.

OBJECTIVES

Primary Objectives

- Develop a user-friendly system for managing pet records in veterinary clinics.
- Create a secure platform for storing and updating pet and owner details.
- Ensure efficient data retrieval and management using a reliable database system.
- Provide streamlined operations for adding, updating, and deleting records.

Business Objectives

- Minimize manual record-keeping efforts.
- Enhance accessibility and organization of clinic operations.
- Improve accuracy in managing pet and owner information.
- Support better decision-making through comprehensive record maintenance.

MODULES

Admin Module

- **Login/Authentication**: Secure login for administrators.
- **Pet Management**:
  - Add new pet records.
  - Update existing pet details.
  - Remove outdated or duplicate records.
- **Owner Management**: Maintain owner contact details for reference.
- **Report Generation**:
  - Generate pet and owner reports.
  - Monitor active and past records.

Pet Records Module

- **Pet Details**:
  - Store and manage pet ID, name, breed, age, and other attributes.

- **Owner Details**:
    - Maintain owner information for each pet.

Database Module

- **Data Management**:
    - Store and organize pet and owner records.
    - Maintain a history of updates.
- **Record Integrity**: Ensure the consistency and accuracy of stored data.

Security Module

- **User Authentication**: Secure login credentials for authorized users.
- **Access Control**: Restrict unauthorized actions to sensitive data.
- **Session Management**: Maintain secure sessions for logged-in users.

This modular design ensures the system is scalable, efficient, and secure while addressing the needs of veterinary clinics.

II. SURVEY OF TECHNOLOGIES

The Pet Management System employs a variety of software tools and programming languages to create an efficient, user-friendly, and secure solution for managing pet records. Below is an overview of the technologies used:

SOFTWARE DESCRIPTION:

Visual Studio Code (VS Code):

Visual Studio Code is a lightweight, open-source, and highly extensible integrated development environment (IDE) widely used for various programming languages and frameworks.

- Key Features for the Project:
    - Support for Java development with extensions like Language Support for Java by Red Hat.
    - Integrated terminal for running and debugging Java applications.
    - Rich ecosystem of plugins for database integration, code formatting, and version control (e.g., Git).
    - Easy customization and cross-platform compatibility.

VS Code offers a streamlined development experience, ensuring efficiency and productivity while building the Pet Management System.

LANGUAGES AND TECHNOLOGIES:

1. MySQL
MySQL is an open-source relational database management system that organizes data into structured tables with defined relationships.

- Usage in the Project:
    - Managing pet and owner details, and maintaining transaction records.
    - Supporting data retrieval and updates using SQL queries.

2. Java
Java is a versatile, platform-independent programming language used to develop the core functionalities of the Pet Management System.

- Usage in the Project:
    - Building the GUI using Java Swing.
    - Integrating database operations through JDBC.
    - Ensuring the application runs seamlessly across different operating systems.

3. HTML
HTML is the standard markup language for structuring and presenting content on the web.

- Usage in the Project:
    - Designing documentation or any potential web-based components.

4. CSS
CSS is used to style and format HTML elements for improved presentation.

- Usage in the Project:
    - Enhancing the visual appeal of any web-based components or documentation.

5. JavaScript
JavaScript is a dynamic programming language for adding interactivity to web

pages.

- Usage in the Project:
  - Potential future enhancements for a web interface of the system.

- **REQUIREMENTS AND ANALYSIS**

HARDWARE AND SOFTWARE REQUIREMENTS

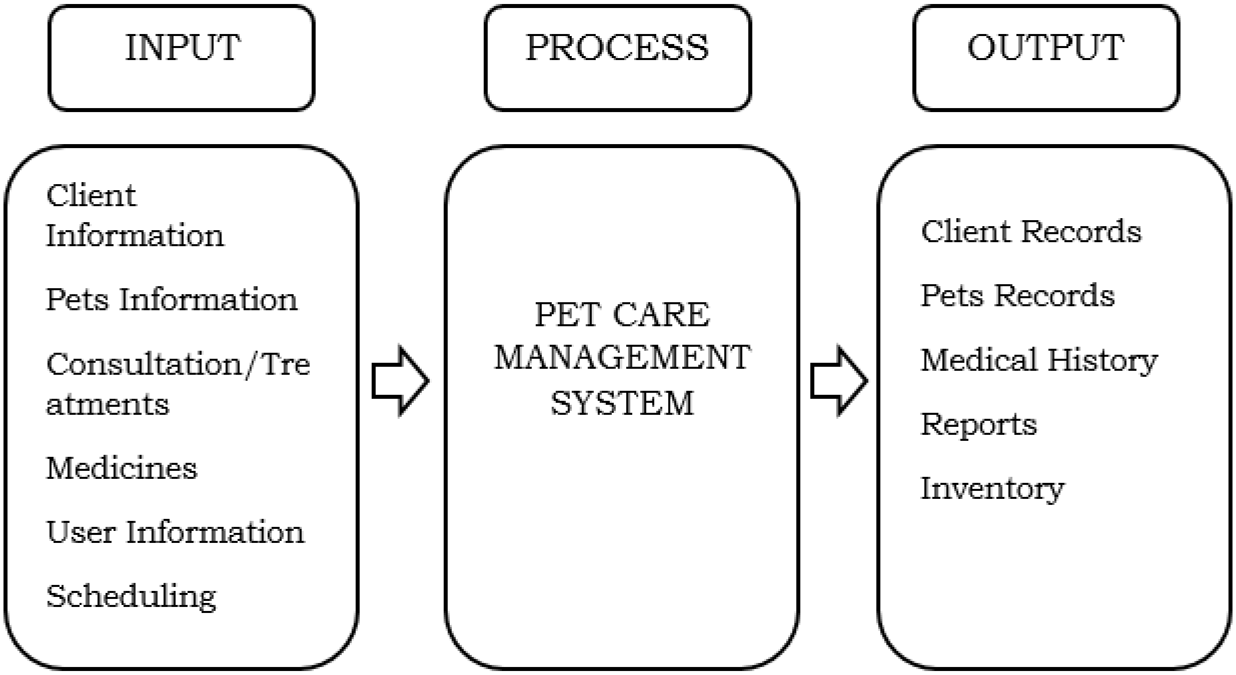
Software Requirements

- Operating System: Windows 10 or higher.
- Development Environment: Visual Studio Code with Java support.
- Front End: Java Swing for GUI development.
- Back End: Java (JDBC) and Oracle Database for data management.

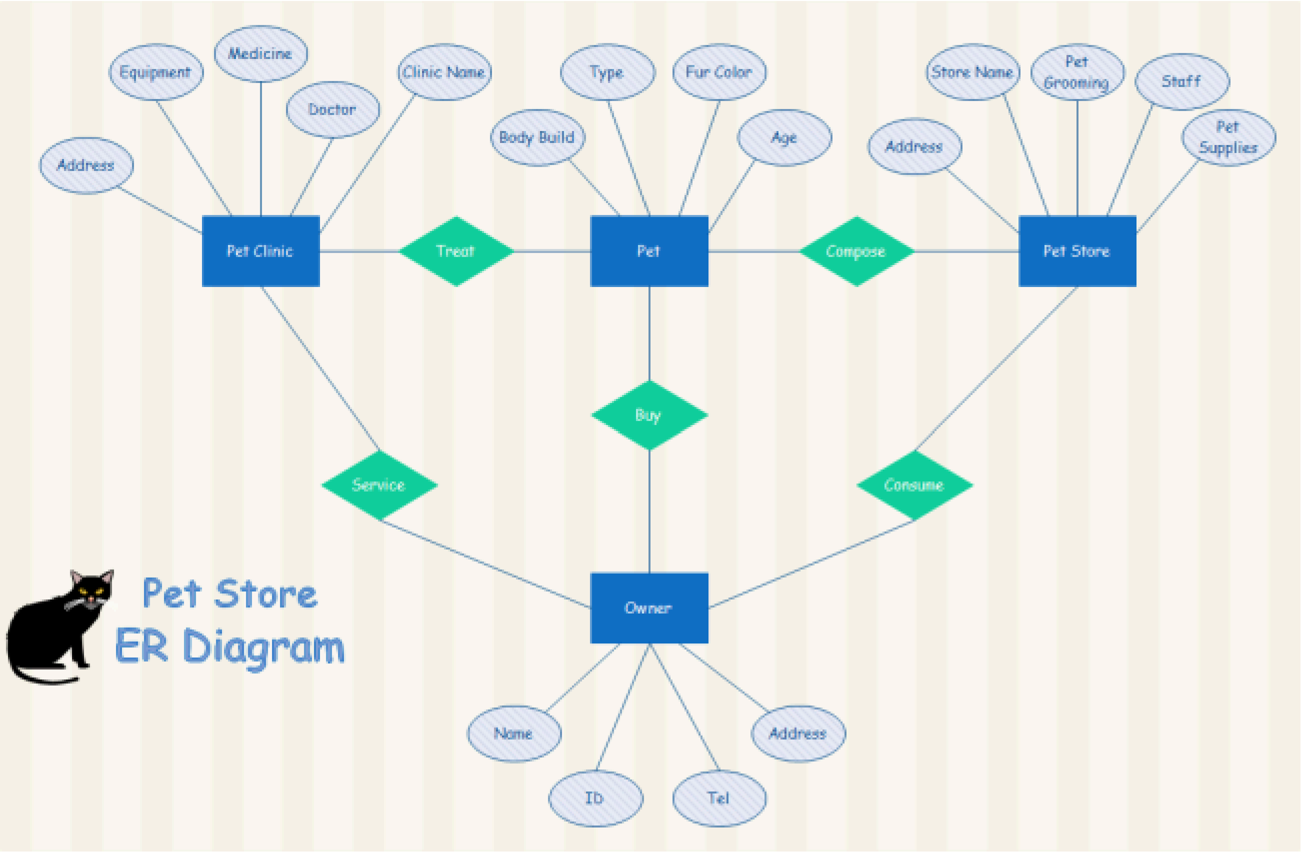
Hardware Requirements

- Processor: Intel® Core™ i3-6006U CPU @ 2.00GHz or higher.
- Memory: 4 GB RAM or higher for smooth application execution.
- Storage: Minimum 10 GB free space for database storage and system files.
- Display: Monitor resolution of 1024 x 768 or higher.
- Peripherals: Keyboard and mouse for input; Printer (optional) for report printing.

Architecture:



ER DIAGRAM


Equipment
Medicine
Doctor
Clinic Name
Address
Pet Clinic
Treat
Type
Fur Color
Body Build
Age
Pet
Compose
Store Name
Pet Grooming
Staff
Address
Pet Supplies
Pet Store
Buy
Service
Consume
Pet Store
ER Diagram
Owner
Name
Address
ID
Tel

CODE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.table.DefaultTableModel;

public class VetClinicSystem {
    private JFrame frame;
    private JTable table;
    private DefaultTableModel tableModel;
    private Connection conn;
    private JTextField txtPetID;

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new VetClinicSystem().initialize());
    }
```

```
 public VetClinicSystem()
{
       try {
        conn=
DriverManager.getConnection("jdbc:oracle:thin:@//AMIRTHAVARSHINI:1
521/XE", "project", "amir");
       } catch (SQLException e) {
          e.printStackTrace();
          JOptionPane.showMessageDialog(null, "Database connection error.");
       }
   }

   public void initialize() {
       frame = new JFrame("Pet Management System");
       frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       frame.setSize(1000, 600);

       JPanel mainPanel = new JPanel(new BorderLayout());
       mainPanel.setBackground(new Color(173, 216, 230));
       frame.getContentPane().add(mainPanel);
```

```
        // Header panel with title and Pet ID input field
        JPanel headerPanel = new JPanel(new BorderLayout());
        headerPanel.setBackground(new Color(173, 216, 230));

        // Centered title label
              JLabel header = new JLabel("Pet Management System",
JLabel.CENTER);
        header.setFont(new Font("Arial", Font.BOLD, 24));
        header.setForeground(new Color(0, 0, 139));
        headerPanel.add(header, BorderLayout.CENTER);

        // Panel for Pet ID label and text field on the right
                      JPanel petIDPanel = new JPanel(new
FlowLayout(FlowLayout.RIGHT));
        petIDPanel.setBackground(new Color(173, 216, 230));
        JLabel lblPetID = new JLabel("Pet ID:");
        lblPetID.setFont(new Font("Arial", Font.PLAIN, 16));
        txtPetID = new JTextField(10);
        petIDPanel.add(lblPetID);
        petIDPanel.add(txtPetID);

        headerPanel.add(petIDPanel, BorderLayout.EAST);
        mainPanel.add(headerPanel, BorderLayout.NORTH);
```

```
// Table and table header styles
        tableModel = new DefaultTableModel(new Object[]{"Pet ID", "Name",
"Breed", "Age", "Owner Name", "Owner Contact"}, 0);
      table = new JTable(tableModel);
      table.setBackground(new Color(240, 248, 255));
      table.getTableHeader().setBackground(new Color(100, 149, 237));
      table.getTableHeader().setForeground(Color.WHITE);
      JScrollPane scrollPane = new JScrollPane(table);
      mainPanel.add(scrollPane, BorderLayout.CENTER);

      // Bottom button panel
                              JPanel buttonPanel = new JPanel(new
FlowLayout(FlowLayout.CENTER, 10, 10));
      buttonPanel.setBackground(new Color(70, 130, 180));
      addButtons(buttonPanel);

      mainPanel.add(buttonPanel, BorderLayout.SOUTH);

      frame.setVisible(true);
  }


private void addButtons(JPanel buttonPanel) {
        String[] buttonNames = {"Add Pet", "Add Appointment", "Search Pet",
"View Appointments", "Add Medical Record", "View Medical Records",
"Generate Bill", "Update Pet", "Delete Pet"};
```

```
        ActionListener[] buttonActions = {
            e -> addPet(), e -> addAppointment(), e -> searchPetByName(),
                      e -> viewAppointments(), e -> addMedicalRecord(), e ->
viewMedicalRecords(), e -> generateBill(),
            e -> updatePet(), e -> deletePet()
        };

        Color buttonColor = new Color(0, 191, 255);

        for (int i = 0; i < buttonNames.length; i++) {
            JButton button = new JButton(buttonNames[i]);
            button.setFont(new Font("Arial", Font.BOLD, 14));
            button.setBackground(buttonColor);
            button.setForeground(Color.WHITE);
            button.addActionListener(buttonActions[i]);
            buttonPanel.add(button);
        }
    }

private int getPetIDInput() {
        try {
            return Integer.parseInt(txtPetID.getText());
        } catch (NumberFormatException e) {
                JOptionPane.showMessageDialog(frame, "Please enter a valid Pet
ID.");
            return -1;
        }
    }

    private void addPet() {
        int petID = getPetIDInput();
        if (petID == -1) return;

            String name = JOptionPane.showInputDialog(frame, "Enter Pet
Name:");
            String breed = JOptionPane.showInputDialog(frame, "Enter Pet
Breed:");
          int age = Integer.parseInt(JOptionPane.showInputDialog(frame, "Enter
Pet Age:"));
           String ownerName = JOptionPane.showInputDialog(frame, "Enter
Owner Name:");
          String ownerContact = JOptionPane.showInputDialog(frame, "Enter
Owner Contact:");



try {
             String query = "INSERT INTO pets (pet_id, name, breed, age,
owner_name, owner_contact) VALUES (?, ?, ?, ?, ?, ?)";
          PreparedStatement stmt = conn.prepareStatement(query);
          stmt.setInt(1, petID);
          stmt.setString(2, name);
          stmt.setString(3, breed);
          stmt.setInt(4, age);
```

```
            stmt.setString(5, ownerName);
            stmt.setString(6, ownerContact);
            stmt.executeUpdate();

            JOptionPane.showMessageDialog(frame, "Pet Added Successfully!");
            refreshPetTable();

        } catch (SQLException e) {
            e.printStackTrace();
             JOptionPane.showMessageDialog(frame, "Error while adding pet: " +
e.getMessage());
        }
    }

 private void addAppointment() {
        int petID = getPetIDInput();
        if (petID == -1) return;

              String veterinarian = JOptionPane.showInputDialog(frame, "Enter
Veterinarian:");
                 String dateInput = JOptionPane.showInputDialog(frame,"Enter
Appointment Date (YYYY-MM-DD):");
        java.sql.Date appointmentDate = java.sql.Date.valueOf(dateInput);
        String status = JOptionPane.showInputDialog(frame, "Enter Status:");

        try {
                 String query = "INSERT INTO appointments (appointment_id,
pet_id, veterinarian,appointment_date, status) VALUES
(appointment_seq.NEXTVAL, ?, ?, ?, ?)";
         PreparedStatement stmt = conn.prepareStatement(query);
         stmt.setInt(1, petID);
         stmt.setString(2, veterinarian);
         stmt.setDate(3, appointmentDate);
         stmt.setString(4, status);
         stmt.executeUpdate();
                JOptionPane.showMessageDialog(frame, "Appointment Added
Successfully!");


// Automatically generate a bill for the appointment
           autoGenerateBill(petID, 50.0, appointmentDate);  // Fixed charge of
$50
      } catch (SQLException e) {
         e.printStackTrace();
                 JOptionPane.showMessageDialog(frame, "Error while adding
appointment: " +
e.getMessage());
      }
   }

      private void autoGenerateBill(int petID, double amount, java.sql.Date
date) {
      try {
             String query = "INSERT INTO bills (bill_id, pet_id, bill_amount,
bill_date) VALUES (bill_seq.NEXTVAL, ?, ?, ?)";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setInt(1, petID);
            stmt.setDouble(2, amount);
            stmt.setDate(3, date);
            stmt.executeUpdate();

                          //JOptionPane.showMessageDialog(frame,"BillGenerated
                  Automatically for $" + amount);
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(frame, "Error while generating bill:
" + e.getMessage());
        }
    }

    private void searchPetByName() {
                  String  name  =  JOptionPane.showInputDialog(frame,  "Enter  Pet
Name:");
        try {
            String query = "SELECT * FROM pets WHERE name LIKE ?";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setString(1, "%" + name + "%");
            ResultSet rs = stmt.executeQuery();
          tableModel.setRowCount(0);
            while (rs.next()) {
                tableModel.addRow(new Object[]{
```

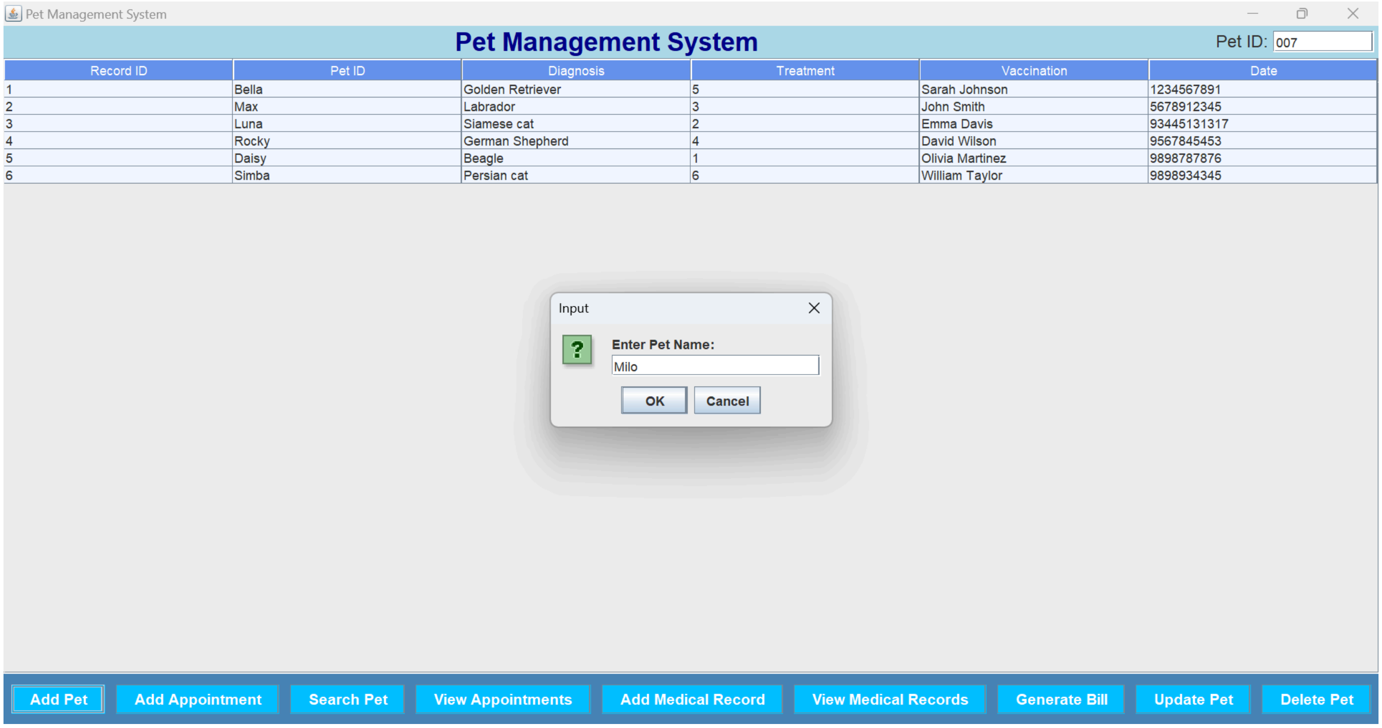
```
            rs.getInt("pet_id"),
            rs.getString("name"),
            rs.getString("breed"),
            rs.getInt("age"),
            rs.getString("owner_name"),
            rs.getString("owner_contact")
        });
      }
    } catch (SQLException e) {
      e.printStackTrace();
       JOptionPane.showMessageDialog(frame, "Error while searching pet:
" + e.getMessage());
    }
  }
    private void viewAppointments() {
    int petID = getPetIDInput();
    if (petID == -1) return;

    try {
        String query = "SELECT * FROM appointments WHERE pet_id =
?";
      PreparedStatement stmt = conn.prepareStatement(query);
      stmt.setInt(1, petID);
      ResultSet rs = stmt.executeQuery();

      tableModel.setRowCount(0);
        tableModel.setColumnIdentifiers(new Object[]{"Appointment ID",
"Pet ID", "Veterinarian",
"Date", "Status"});
            while (rs.next()) {
                tableModel.addRow(new Object[]{
                        rs.getInt("appointment_id"),
                        rs.getInt("pet_id"),
                        rs.getString("veterinarian"),
                        rs.getDate("appointment_date"),
                        rs.getString("status")
                });
            }
        }


             catch (SQLException e) {
            e.printStackTrace();
                   JOptionPane.showMessageDialog(frame, "Error while retrieving
appointments: " +
e.getMessage());
        }
    }
private void addMedicalRecord() {
        int petID = getPetIDInput();
        if (petID == -1) return;

                  String diagnosis = JOptionPane.showInputDialog(frame, "Enter
Diagnosis:");
```

```
            String treatment = JOptionPane.showInputDialog(frame, "Enter
Treatment:");
           String vaccination = JOptionPane.showInputDialog(frame, "Enter
Vaccination:");
       String dateInput = JOptionPane.showInputDialog(frame, "Enter Record
Date (YYYY-MM-DD):");
      java.sql.Date recordDate = java.sql.Date.valueOf(dateInput);
       try {
           String query = "INSERT INTO medical_records (record_id, pet_id,
diagnosis, treatment,vaccination, record_date) VALUES
(medical_record_seq.NEXTVAL, ?, ?, ?, ?, ?)";
         PreparedStatement stmt = conn.prepareStatement(query);
         stmt.setInt(1, petID);
         stmt.setString(2, diagnosis);
         stmt.setString(3, treatment);
         stmt.setString(4, vaccination);
         stmt.setDate(5, recordDate);
         stmt.executeUpdate();
            JOptionPane.showMessageDialog(frame, "Medical Record Added
Successfully!");

         // Automatically generate a bill for the treatment
         autoGenerateBill(petID, 100.0, recordDate);  // Fixed charge of $100
      } catch (SQLException e) {
         e.printStackTrace();
               JOptionPane.showMessageDialog(frame, "Error while adding
medical record: " + e.getMessage());
```

```
        }
    }




         private void viewMedicalRecords() {
        int petID = getPetIDInput();
        if (petID == -1) return;

        try {
            String query = "SELECT * FROM medical_records WHERE pet_id =
?";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setInt(1, petID);
            ResultSet rs = stmt.executeQuery();

            tableModel.setRowCount(0);
               tableModel.setColumnIdentifiers(new Object[]{"Record ID", "Pet
ID", "Diagnosis",
"Treatment", "Vaccination", "Date"});


         while (rs.next()) {
              tableModel.addRow(new Object[]{
                   rs.getInt("record_id"),
                   rs.getInt("pet_id"),
                        rs.getString("diagnosis"),
                        rs.getString("treatment"),
                        rs.getString("vaccination"),
                        rs.getDate("record_date")
                });
            }
        } catch (SQLException e) {
            e.printStackTrace();
                    JOptionPane.showMessageDialog(frame, "Error while retrieving
medical records: " +
e.getMessage());
        }
    }

    private void generateBill() {
        int petID = getPetIDInput();
        if (petID == -1) return;

        try {
                String query = "SELECT SUM(bill_amount) FROM bills WHERE
pet_id = ?";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setInt(1, petID);
            ResultSet rs = stmt.executeQuery();

            if (rs.next()) {
                double totalAmount = rs.getDouble(1);
                JOptionPane.showMessageDialog(frame, "Total Bill: " +
totalAmount);
            }
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(frame, "Error while generating bill:
" + e.getMessage());
        }
    }

    private void deletePet() {
        int petID = getPetIDInput();
        if (petID == -1) return;

        try {
            // Begin transaction
            conn.setAutoCommit(false);

              // Step 1: Delete related records in child tables (adjust table names as
necessary)
                String deleteMedicalRecordsQuery = "DELETE FROM
medical_records WHERE pet_id = ?";
                                        PreparedStatement stmt1 =
conn.prepareStatement(deleteMedicalRecordsQuery);
            stmt1.setInt(1, petID);
            stmt1.executeUpdate();
              String deleteAppointmentsQuery = "DELETE FROM appointments
WHERE pet_id = ?";
                                            PreparedStatement stmt2 =
conn.prepareStatement(deleteAppointmentsQuery);
            stmt2.setInt(1, petID);
            stmt2.executeUpdate();

        String deleteBillsQuery = "DELETE FROM bills WHERE pet_id = ?";
            PreparedStatement stmt3 = conn.prepareStatement(deleteBillsQuery);
            stmt3.setInt(1, petID);
            stmt3.executeUpdate();

            // Step 2: Delete the pet record from the pets table
            String deletePetQuery = "DELETE FROM pets WHERE pet_id = ?";
            PreparedStatement stmt4 = conn.prepareStatement(deletePetQuery);
            stmt4.setInt(1, petID);
            int rowsAffected = stmt4.executeUpdate();

            // Commit transaction
            conn.commit();

    if (rowsAffected > 0) {
                          JOptionPane.showMessageDialog(frame, "Pet Deleted
Successfully!");
              refreshPetTable();
            } else {
              JOptionPane.showMessageDialog(frame, "Pet ID not found.");
            }
        } catch (SQLException e) {
            e.printStackTrace();
            try {
                // Rollback transaction in case of error
                conn.rollback();
            } catch (SQLException rollbackEx) {
                rollbackEx.printStackTrace();
            }
              JOptionPane.showMessageDialog(frame, "Error while deleting pet: "
+ e.getMessage());
        }
          finally {
            try {
                // Restore default auto-commit behavior
                conn.setAutoCommit(true);
            } catch (SQLException ex) {
                ex.printStackTrace();
            }
        }
    }

    private void refreshPetTable() {
        try {
            String query = "SELECT * FROM pets";
            PreparedStatement stmt = conn.prepareStatement(query);
            ResultSet rs = stmt.executeQuery();
            tableModel.setRowCount(0);


        while (rs.next()) {
                tableModel.addRow(new Object[]{
                        rs.getInt("pet_id"),
                        rs.getString("name"),
                        rs.getString("breed"),
                        rs.getInt("age"),
                        rs.getString("owner_name"),
                        rs.getString("owner_contact")
                });
            }
        } catch (SQLException e) {
            e.printStackTrace();
             JOptionPane.showMessageDialog(frame, "Error while refreshing pet
table: " +
e.getMessage());
        }
    }

    private void updatePet() {
        int petID = getPetIDInput();
        if (petID == -1) return;

        String name = JOptionPane.showInputDialog(frame, "Enter New Pet
Name:");
            String breed = JOptionPane.showInputDialog(frame, "Enter New Pet
Breed:");
           int age = Integer.parseInt(JOptionPane.showInputDialog(frame, "Enter
New Pet Age:"));
          String ownerName = JOptionPane.showInputDialog(frame, "Enter New
Owner Name:");
             String ownerContact = JOptionPane.showInputDialog(frame, "Enter
New Owner Contact:");

        try {
                String query = "UPDATE pets SET name = ?, breed = ?, age = ?,
owner_name = ?, owner_contact = ? WHERE pet_id = ?";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setString(1, name);
            stmt.setString(2, breed);
            stmt.setInt(3, age);
            stmt.setString(4, ownerName);
            stmt.setString(5, ownerContact);
            stmt.setInt(6, petID);
            int rowsAffected = stmt.executeUpdate();

            if (rowsAffected > 0) {
                        JOptionPane.showMessageDialog(frame,  "Pet  Updated
Successfully!");
              refreshPetTable();
            } else {
              JOptionPane.showMessageDialog(frame, "Pet ID not found.");
            }
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(frame, "Error while updating pet: "
+ e.getMessage());
        }
    }
}
```

OUTPUT

i) Insert pet details:



Pet Management System

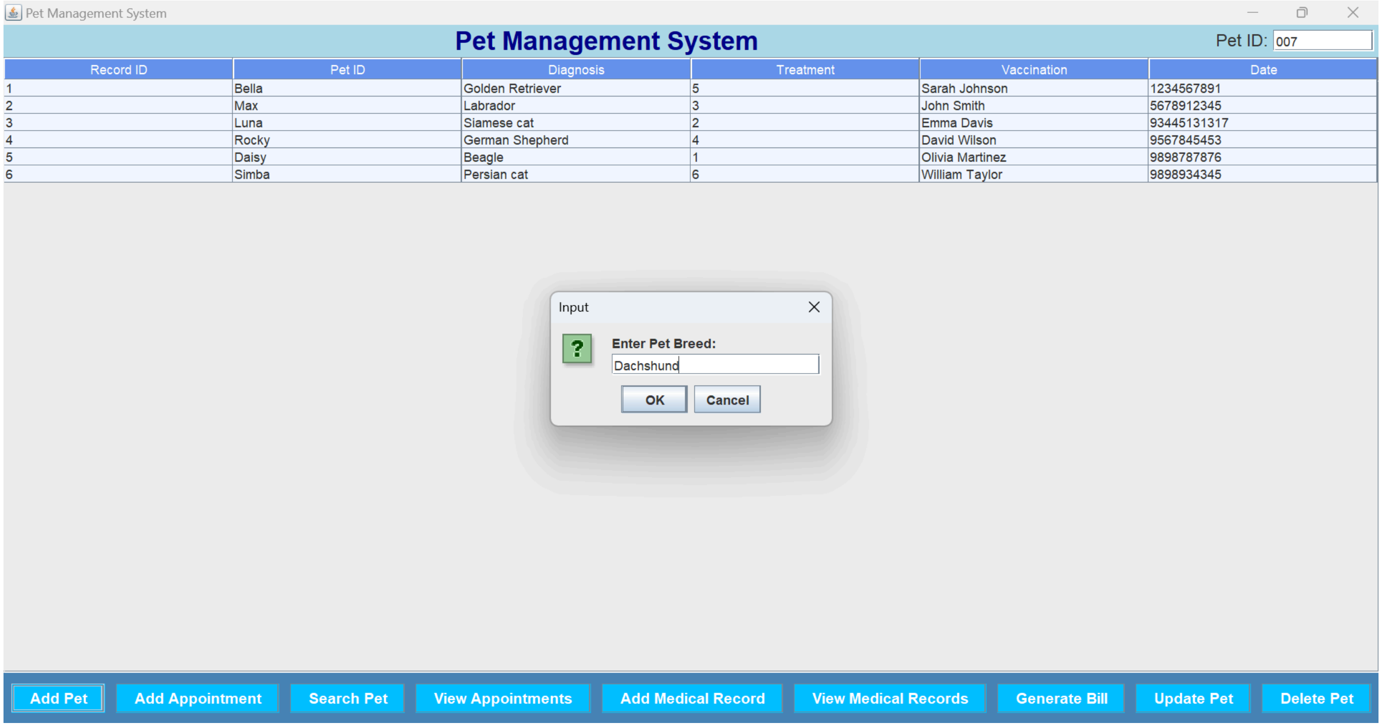
Pet ID: 007

Record ID	Pet ID	Diagnosis	Treatment	Vaccination	Date
1	Bella	Golden Retriever	5	Sarah Johnson	1234567891
2	Max	Labrador	3	John Smith	5678912345
3	Luna	Siamese cat	2	Emma Davis	93445131317
4	Rocky	German Shepherd	4	David Wilson	9567845453
5	Daisy	Beagle	1	Olivia Martinez	9898787876
6	Simba	Persian cat	6	William Taylor	9898934345

Input

Enter Pet Name: Milo

OK Cancel



Pet Management System

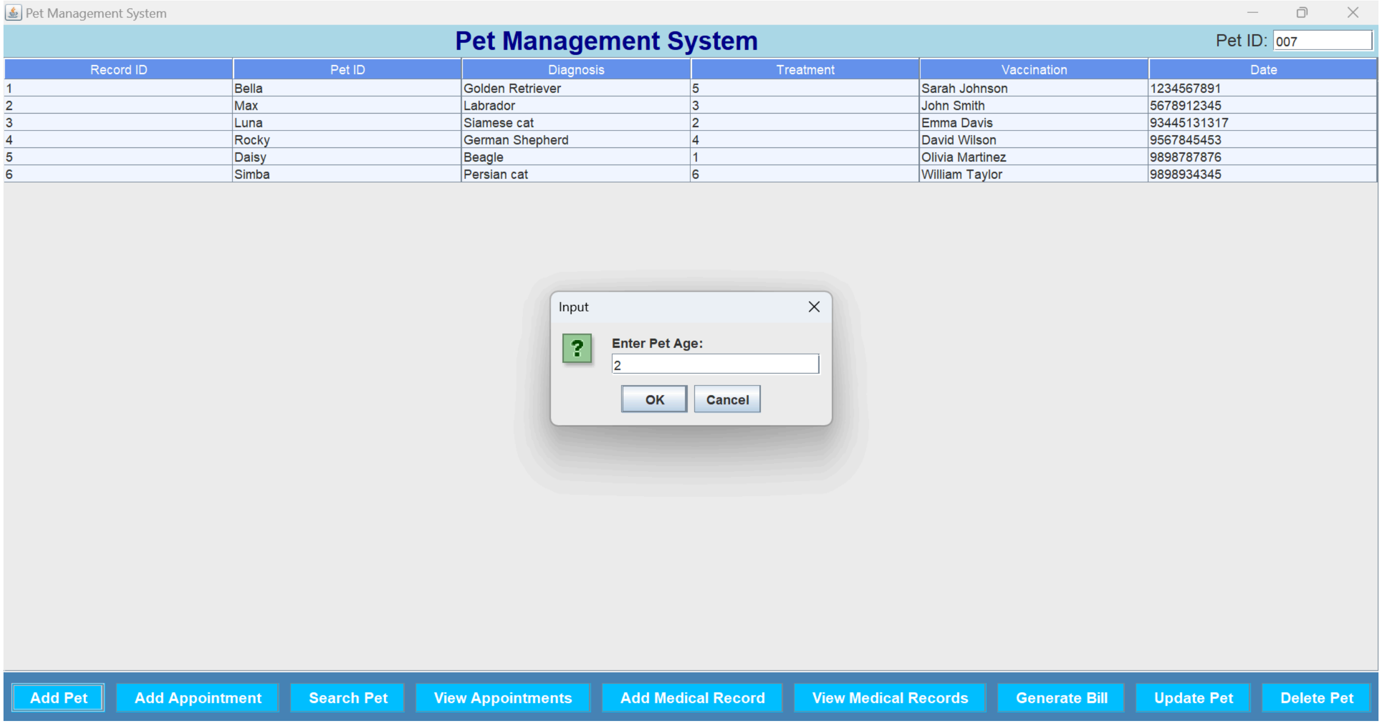
Pet ID: 007

Record ID	Pet ID	Diagnosis	Treatment	Vaccination	Date
1	Bella	Golden Retriever	5	Sarah Johnson	1234567891
2	Max	Labrador	3	John Smith	5678912345
3	Luna	Siamese cat	2	Emma Davis	93445131317
4	Rocky	German Shepherd	4	David Wilson	9567845453
5	Daisy	Beagle	1	Olivia Martinez	9898787876
6	Simba	Persian cat	6	William Taylor	9898934345

Input

Enter Pet Breed: Dachshund

OK Cancel



Pet Management System

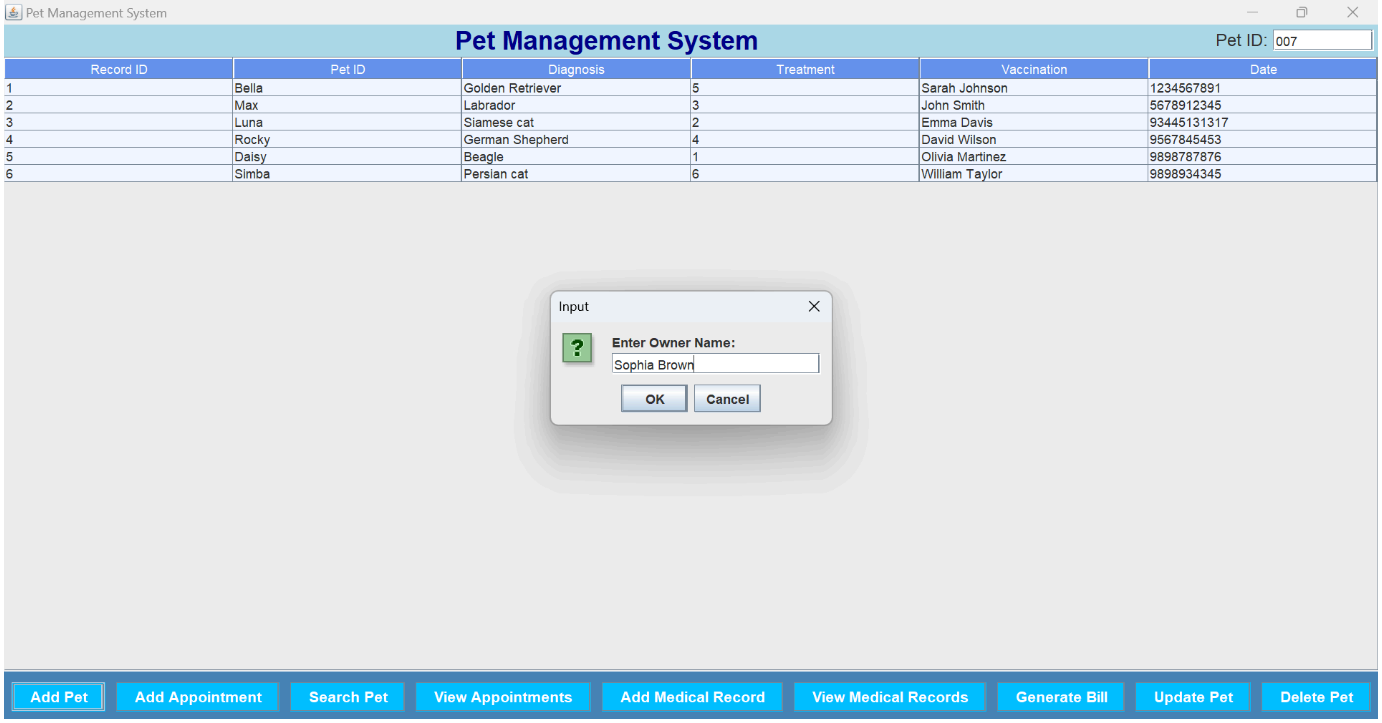
Pet ID: 007

Record ID	Pet ID	Diagnosis	Treatment	Vaccination	Date
1	Bella	Golden Retriever	5	Sarah Johnson	1234567891
2	Max	Labrador	3	John Smith	5678912345
3	Luna	Siamese cat	2	Emma Davis	93445131317
4	Rocky	German Shepherd	4	David Wilson	9567845453
5	Daisy	Beagle	1	Olivia Martinez	9898787876
6	Simba	Persian cat	6	William Taylor	9898934345

Input

Enter Pet Age:

2



Pet Management System

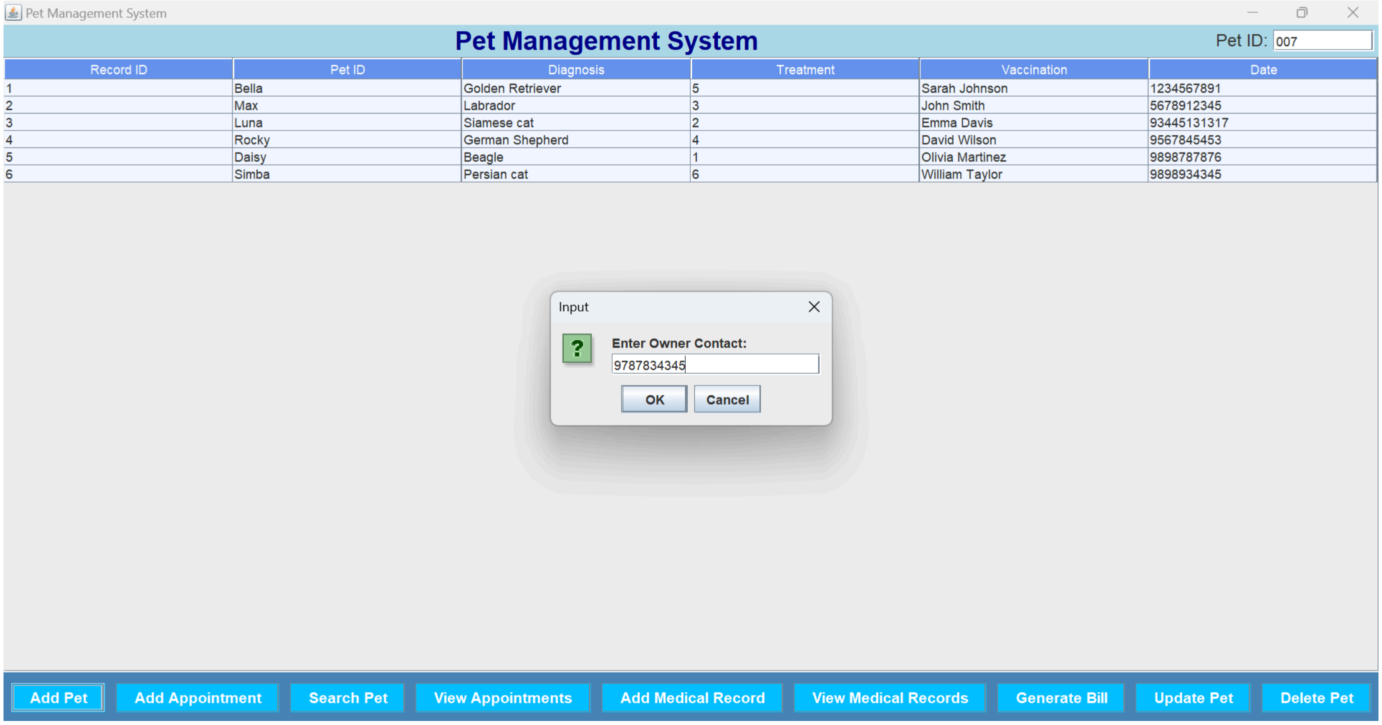
Pet ID: 007

Record ID	Pet ID	Diagnosis	Treatment	Vaccination	Date
1	Bella	Golden Retriever	5	Sarah Johnson	1234567891
2	Max	Labrador	3	John Smith	5678912345
3	Luna	Siamese cat	2	Emma Davis	93445131317
4	Rocky	German Shepherd	4	David Wilson	9567845453
5	Daisy	Beagle	1	Olivia Martinez	9898787876
6	Simba	Persian cat	6	William Taylor	9898934345

Input

Enter Owner Name:

Sophia Brown



Pet Management System

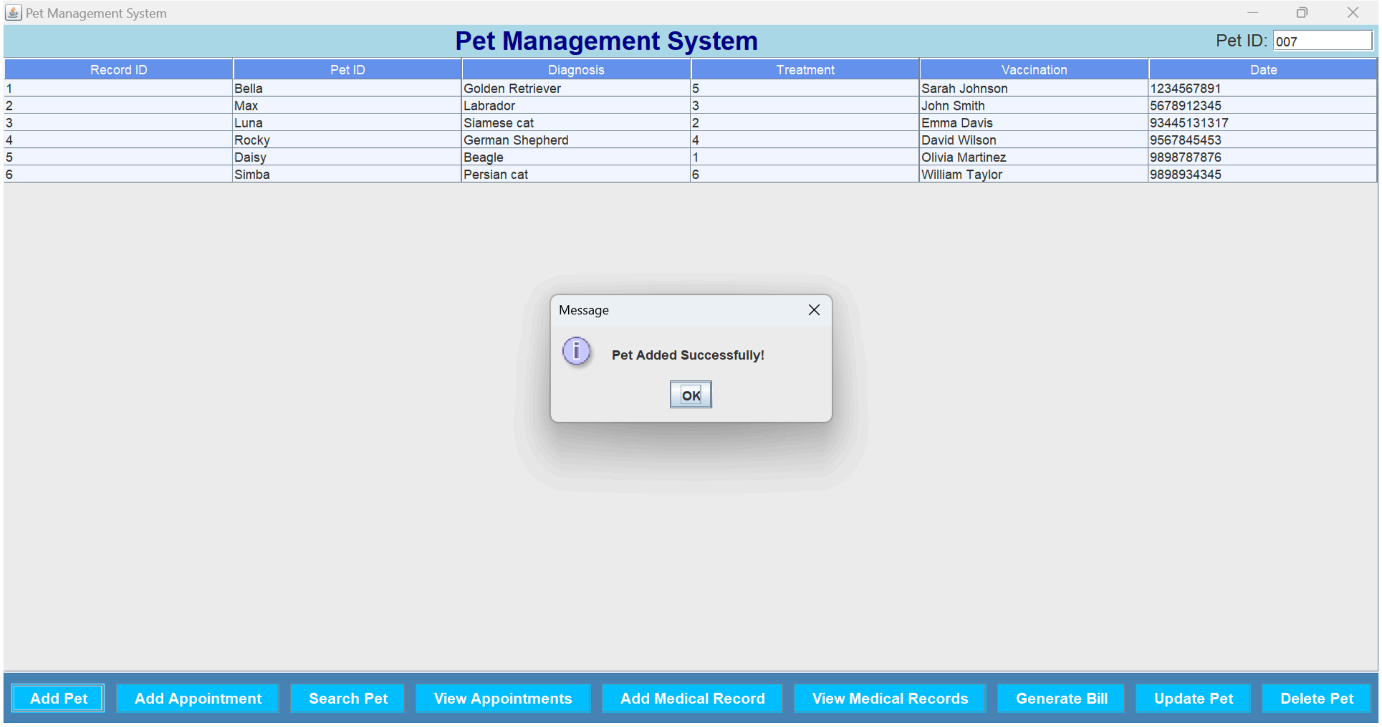
Pet Management System

Pet ID: 007

Record ID	Pet ID	Diagnosis	Treatment	Vaccination	Date
1	Bella	Golden Retriever	5	Sarah Johnson	1234567891
2	Max	Labrador	3	John Smith	5678912345
3	Luna	Siamese cat	2	Emma Davis	93445131317
4	Rocky	German Shepherd	4	David Wilson	9567845453
5	Daisy	Beagle	1	Olivia Martinez	9898787876
6	Simba	Persian cat	6	William Taylor	9898934345

Input

Enter Owner Contact: 9787834345



Pet Management System

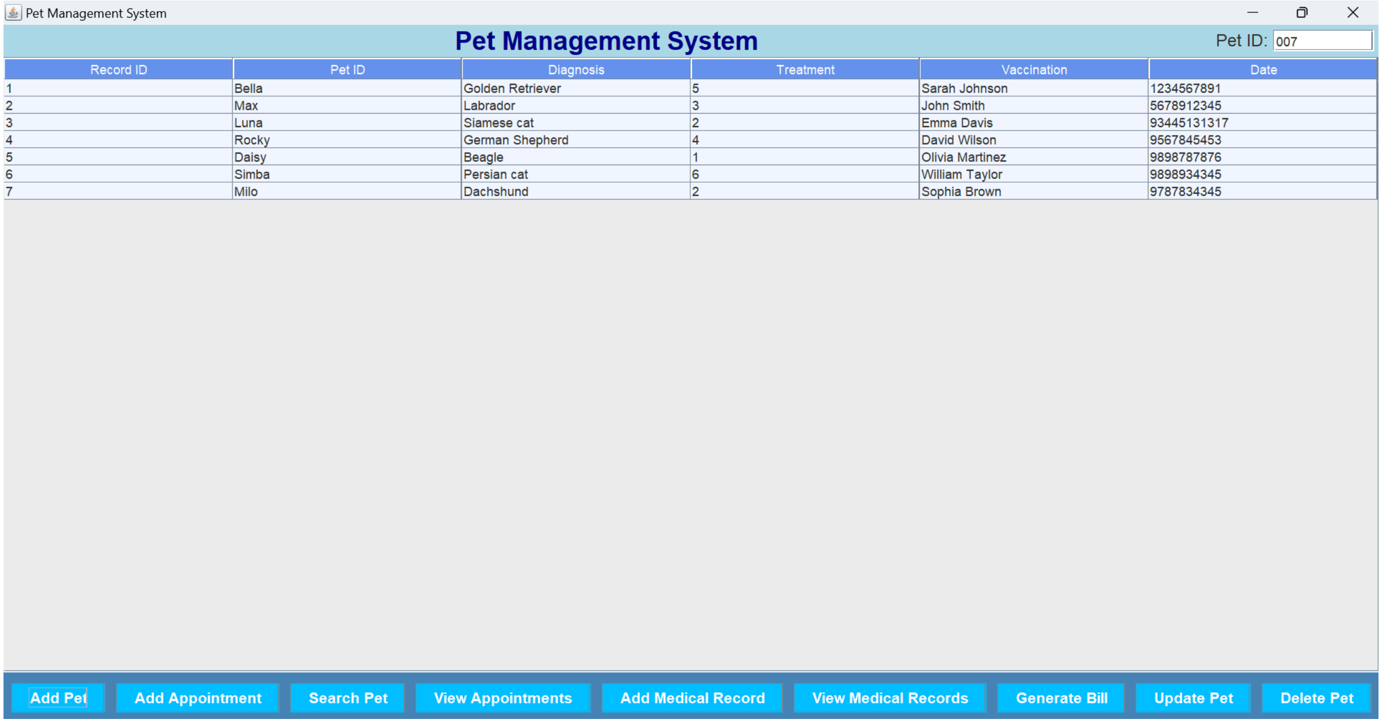
Pet Management System

Pet ID: 007

Record ID	Pet ID	Diagnosis	Treatment	Vaccination	Date
1	Bella	Golden Retriever	5	Sarah Johnson	1234567891
2	Max	Labrador	3	John Smith	5678912345
3	Luna	Siamese cat	2	Emma Davis	93445131317
4	Rocky	German Shepherd	4	David Wilson	9567845453
5	Daisy	Beagle	1	Olivia Martinez	9898787876
6	Simba	Persian cat	6	William Taylor	9898934345

Message

Pet Added Successfully!

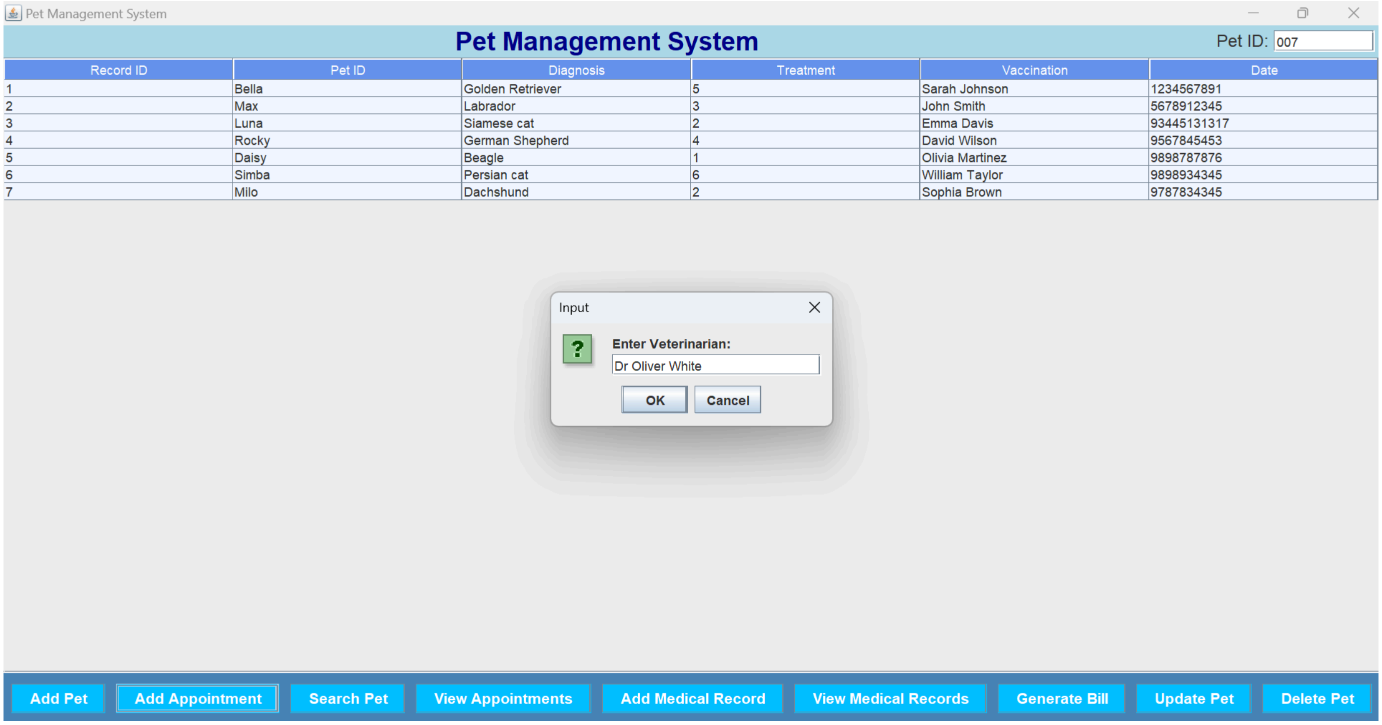


Pet Management System

Pet ID: 007

Record ID	Pet ID	Diagnosis	Treatment	Vaccination	Date
1	Bella	Golden Retriever	5	Sarah Johnson	1234567891
2	Max	Labrador	3	John Smith	5678912345
3	Luna	Siamese cat	2	Emma Davis	93445131317
4	Rocky	German Shepherd	4	David Wilson	9567845453
5	Daisy	Beagle	1	Olivia Martinez	9898787876
6	Simba	Persian cat	6	William Taylor	9898934345
7	Milo	Dachshund	2	Sophia Brown	9787834345

ii) Insert appointment details:



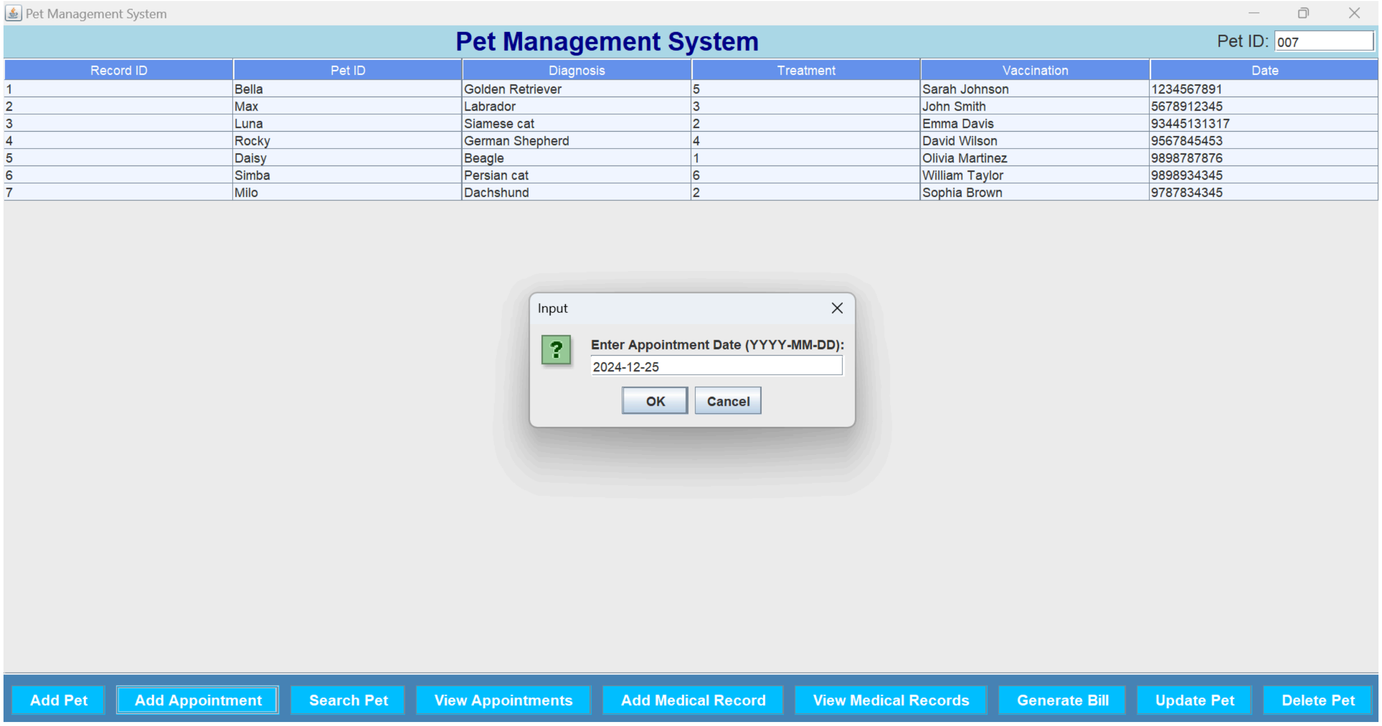
Pet Management System

Pet ID: 007

Record ID	Pet ID	Diagnosis	Treatment	Vaccination	Date
1	Bella	Golden Retriever	5	Sarah Johnson	1234567891
2	Max	Labrador	3	John Smith	5678912345
3	Luna	Siamese cat	2	Emma Davis	93445131317
4	Rocky	German Shepherd	4	David Wilson	9567845453
5	Daisy	Beagle	1	Olivia Martinez	9898787876
6	Simba	Persian cat	6	William Taylor	9898934345
7	Milo	Dachshund	2	Sophia Brown	9787834345

Input

Enter Veterinarian: Dr Oliver White



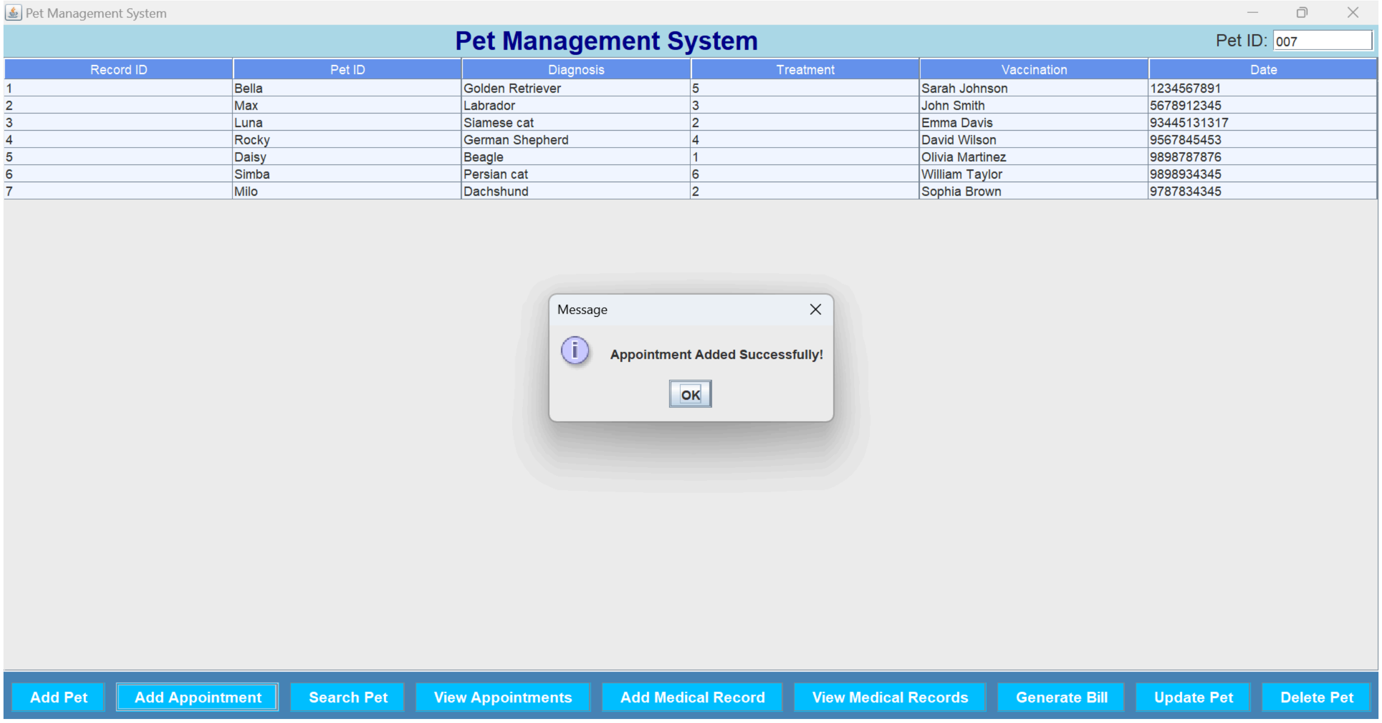
Pet Management System

Pet ID: 007

Record ID	Pet ID	Diagnosis	Treatment	Vaccination	Date
1	Bella	Golden Retriever	5	Sarah Johnson	1234567891
2	Max	Labrador	3	John Smith	5678912345
3	Luna	Siamese cat	2	Emma Davis	93445131317
4	Rocky	German Shepherd	4	David Wilson	9567845453
5	Daisy	Beagle	1	Olivia Martinez	9898787876
6	Simba	Persian cat	6	William Taylor	9898934345
7	Milo	Dachshund	2	Sophia Brown	9787834345

Input

Enter Appointment Date (YYYY-MM-DD): 2024-12-25



Pet Management System

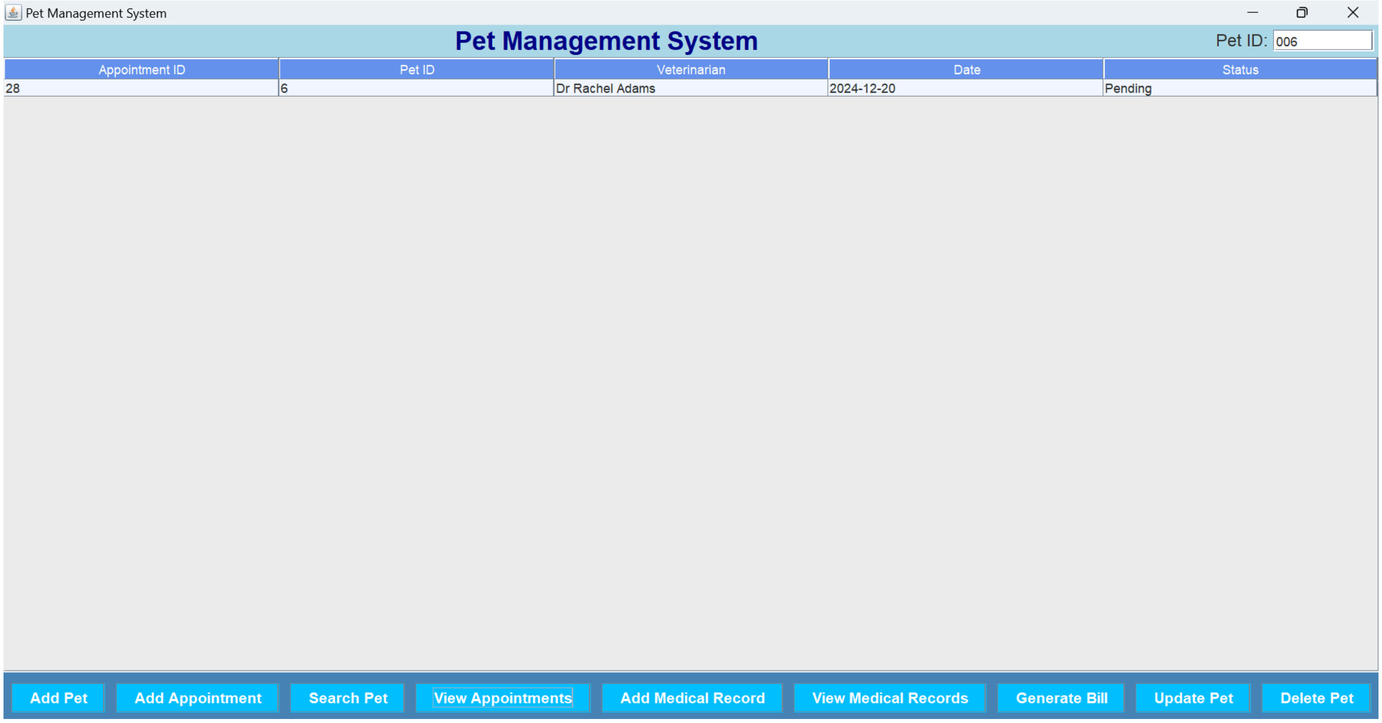
Pet ID: 007

Record ID	Pet ID	Diagnosis	Treatment	Vaccination	Date
1	Bella	Golden Retriever	5	Sarah Johnson	1234567891
2	Max	Labrador	3	John Smith	5678912345
3	Luna	Siamese cat	2	Emma Davis	93445131317
4	Rocky	German Shepherd	4	David Wilson	9567845453
5	Daisy	Beagle	1	Olivia Martinez	9898787876
6	Simba	Persian cat	6	William Taylor	9898934345
7	Milo	Dachshund	2	Sophia Brown	9787834345

Message

Appointment Added Successfully!

OK

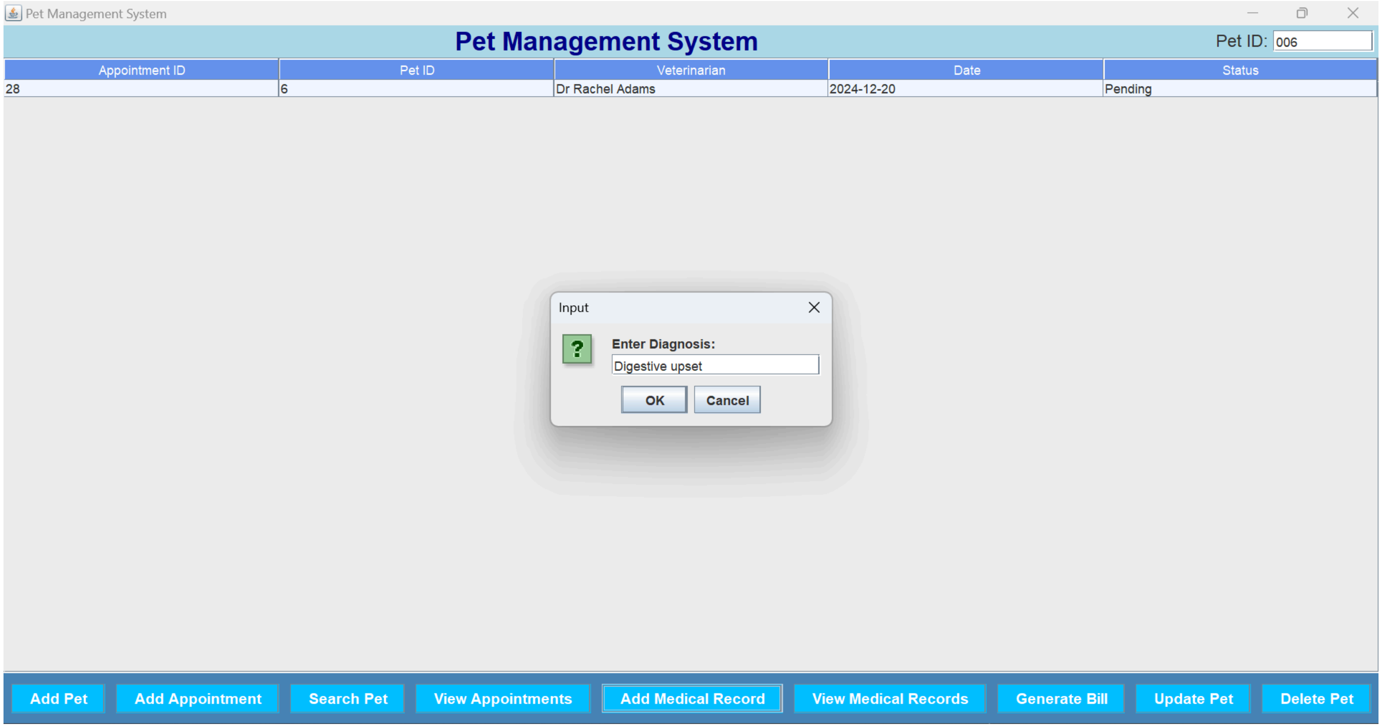


Pet Management System

Pet ID: 006

Appointment ID	Pet ID	Veterinarian	Date	Status
28	6	Dr Rachel Adams	2024-12-20	Pending

iii)Inserting medical report details:

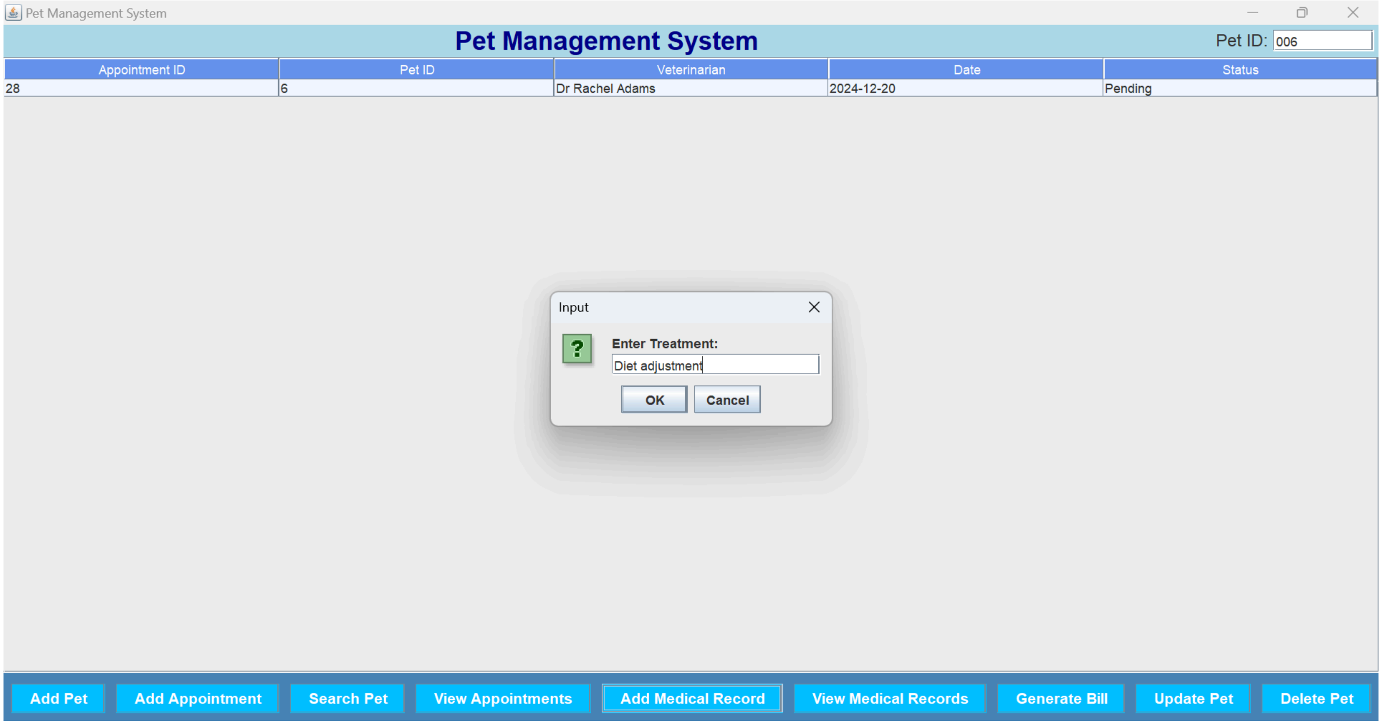


Pet Management System

Pet ID: 006

Appointment ID	Pet ID	Veterinarian	Date	Status
28	6	Dr Rachel Adams	2024-12-20	Pending

Enter Diagnosis: Digestive upset

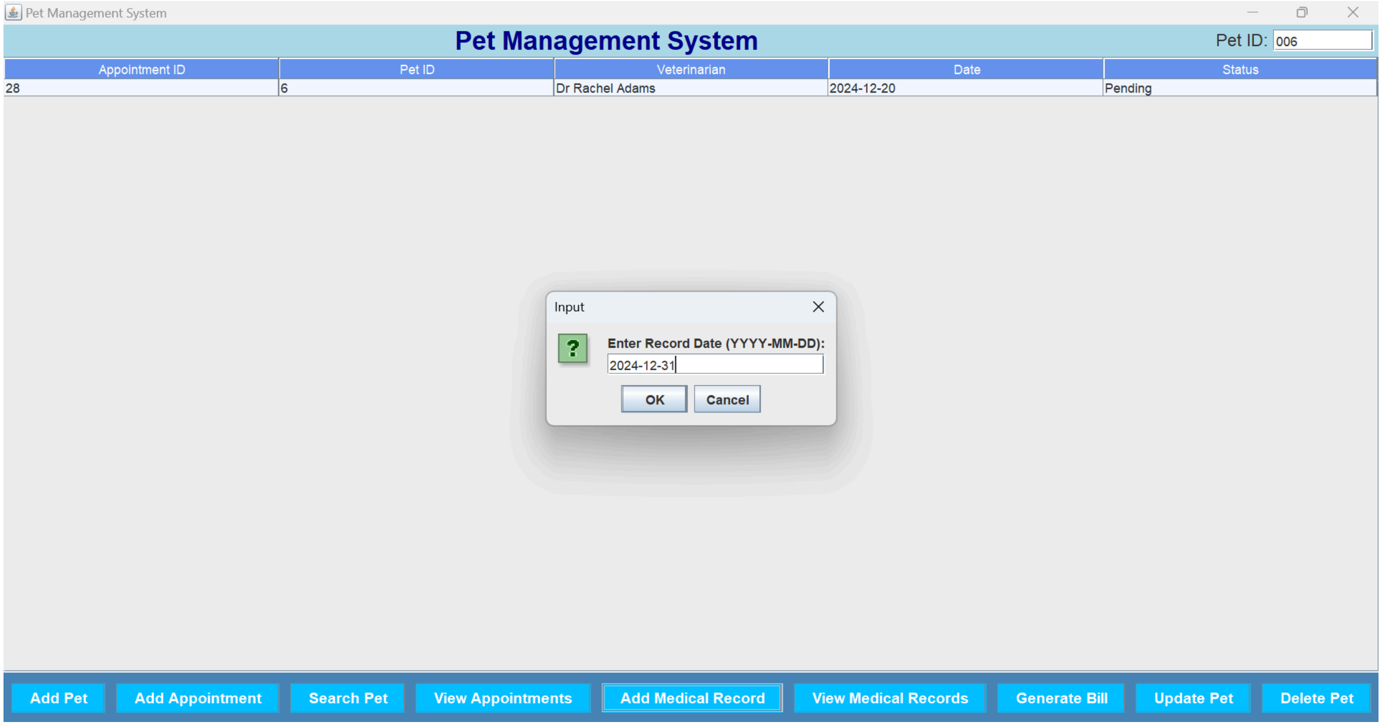


Pet Management System

Pet ID: 006

Appointment ID	Pet ID	Veterinarian	Date	Status
28	6	Dr Rachel Adams	2024-12-20	Pending

Enter Treatment: Diet adjustment



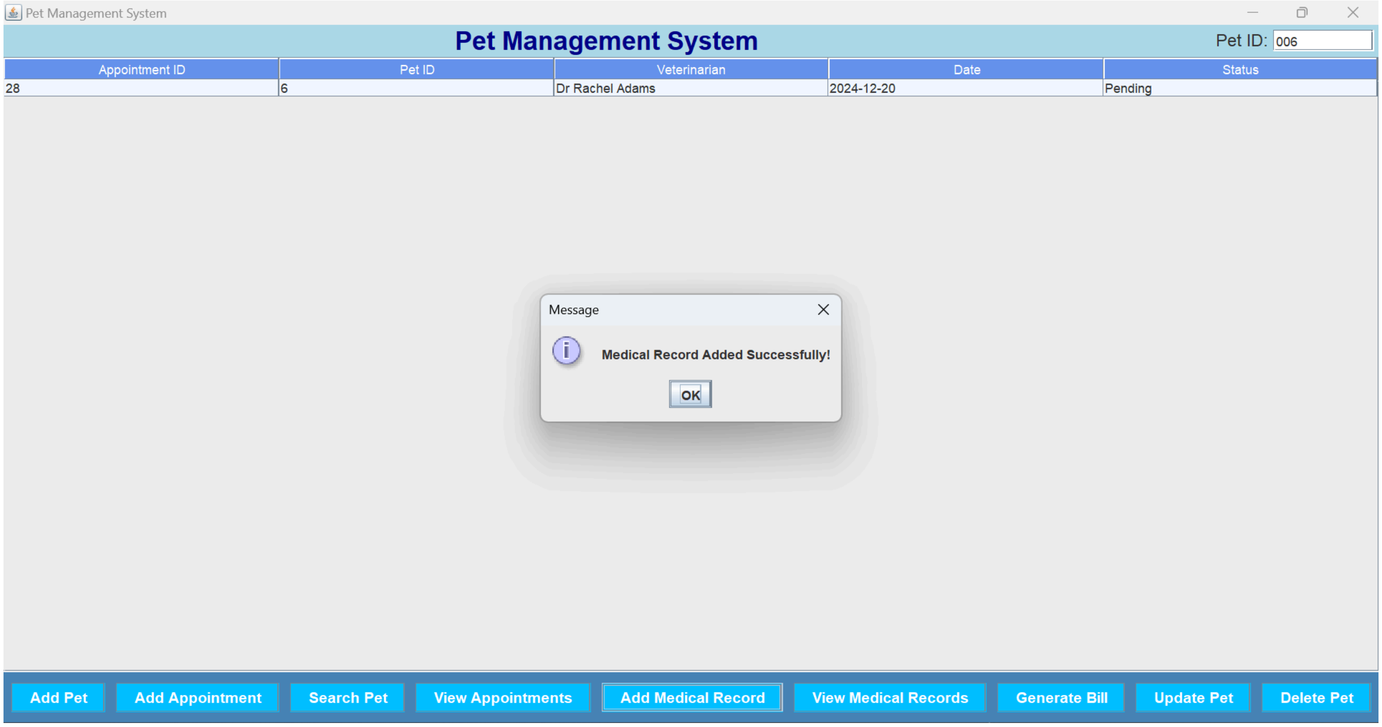
Pet Management System

Pet ID: 006

Appointment ID	Pet ID	Veterinarian	Date	Status
28	6	Dr Rachel Adams	2024-12-20	Pending

Input

Enter Record Date (YYYY-MM-DD): 2024-12-31



Pet Management System

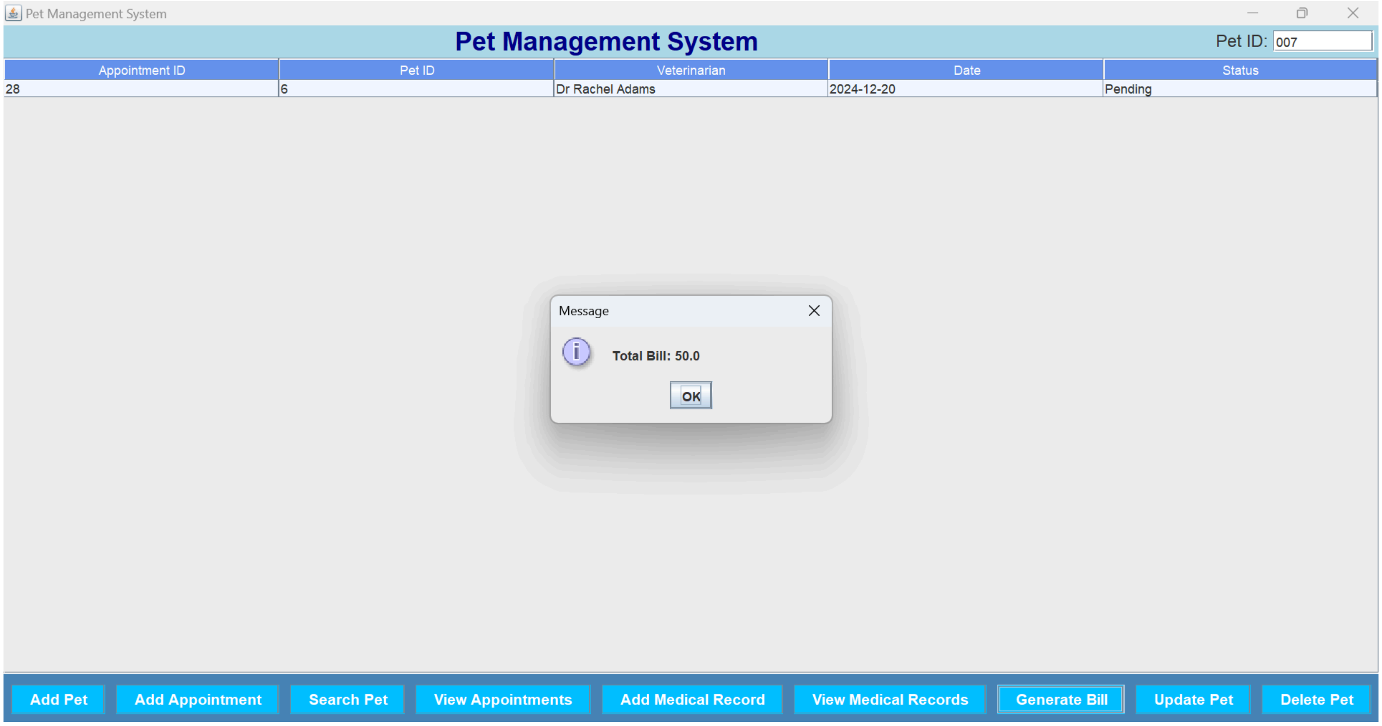
Pet ID: 006

Appointment ID	Pet ID	Veterinarian	Date	Status
28	6	Dr Rachel Adams	2024-12-20	Pending

Message

Medical Record Added Successfully!

iv)Bill:



Pet Management System

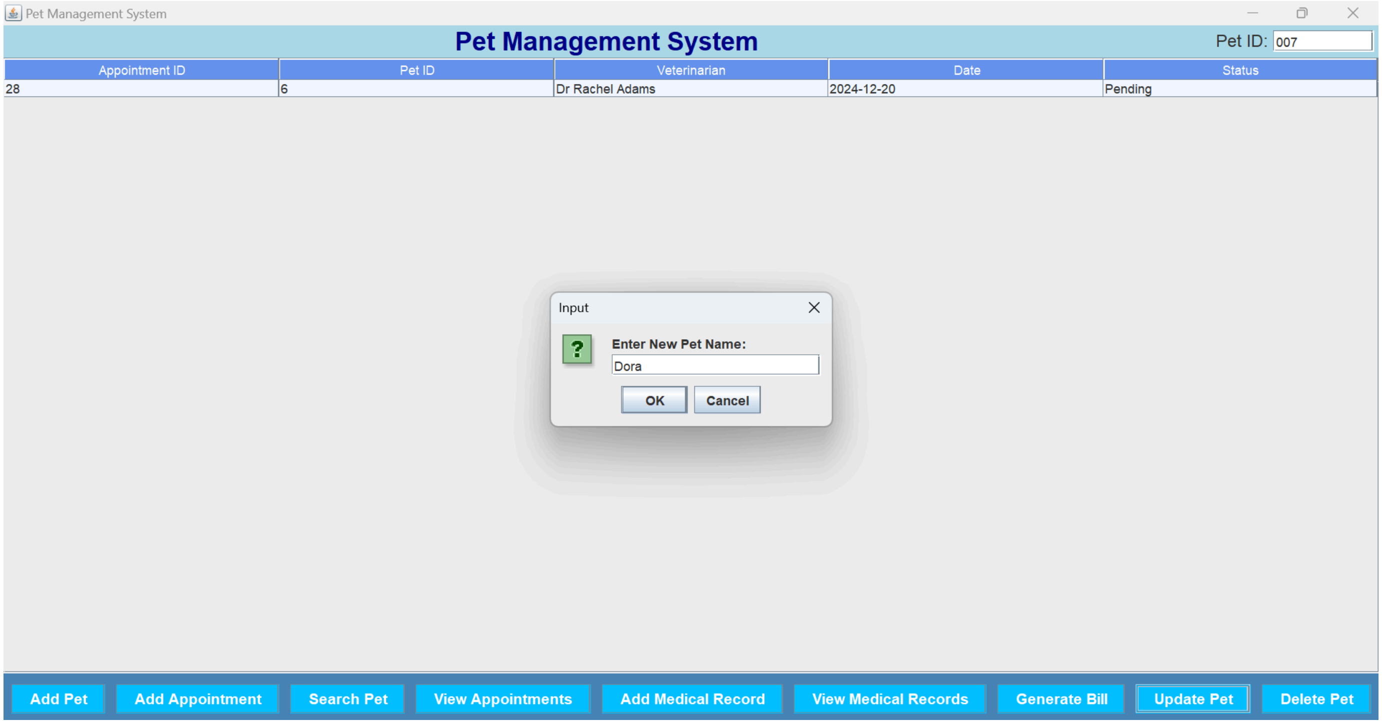
Pet ID: 007

Appointment ID	Pet ID	Veterinarian	Date	Status
28	6	Dr Rachel Adams	2024-12-20	Pending

Message

Total Bill: 50.0

v)Update:



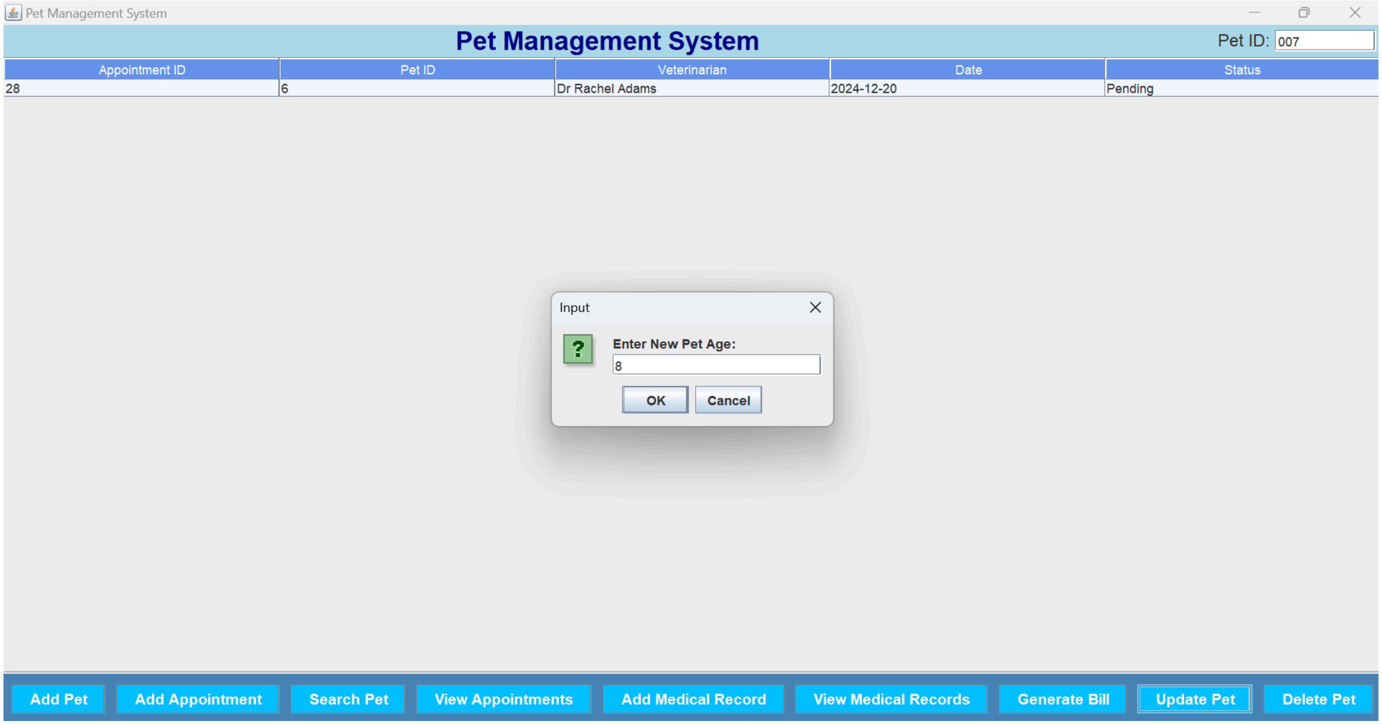
Pet Management System

Pet ID: 007

Appointment ID	Pet ID	Veterinarian	Date	Status
28	6	Dr Rachel Adams	2024-12-20	Pending

Input

Enter New Pet Name: Dora

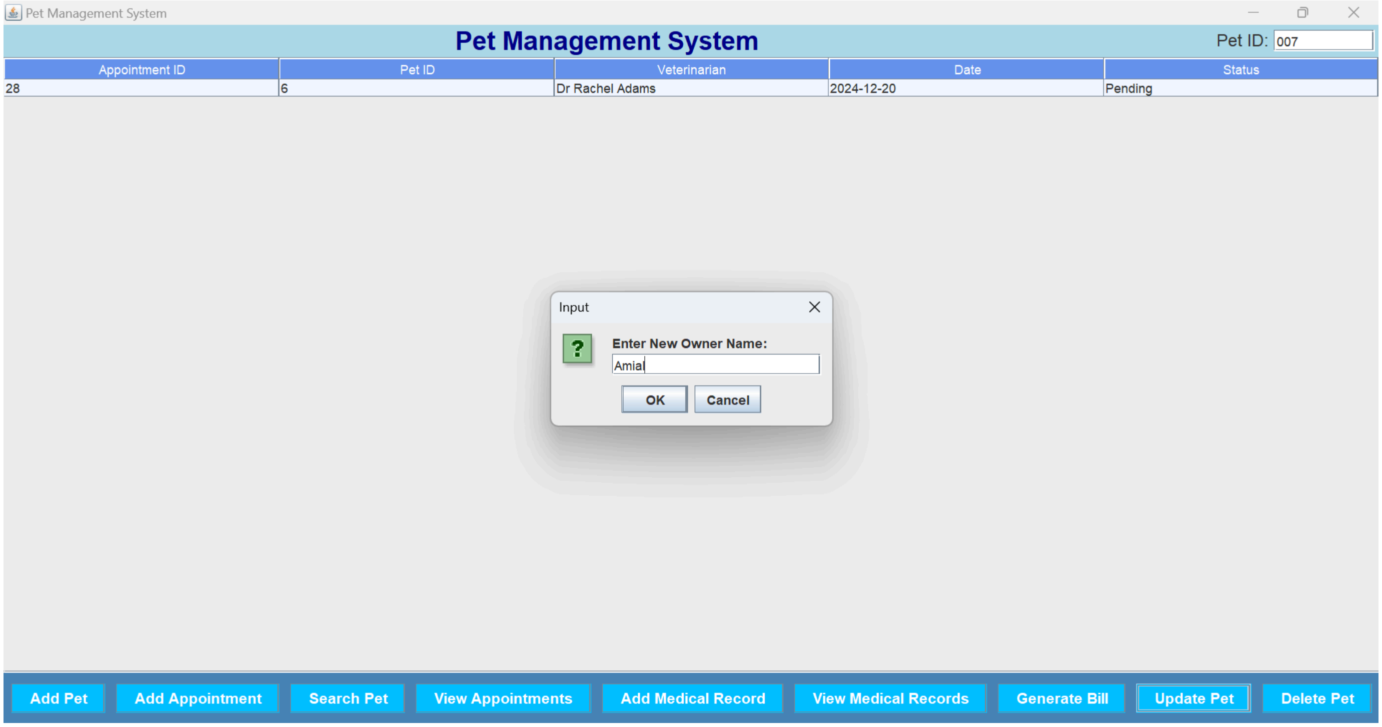


Pet Management System

Pet ID: 007

Appointment ID	Pet ID	Veterinarian	Date	Status
28	6	Dr Rachel Adams	2024-12-20	Pending

Enter New Pet Age:
8

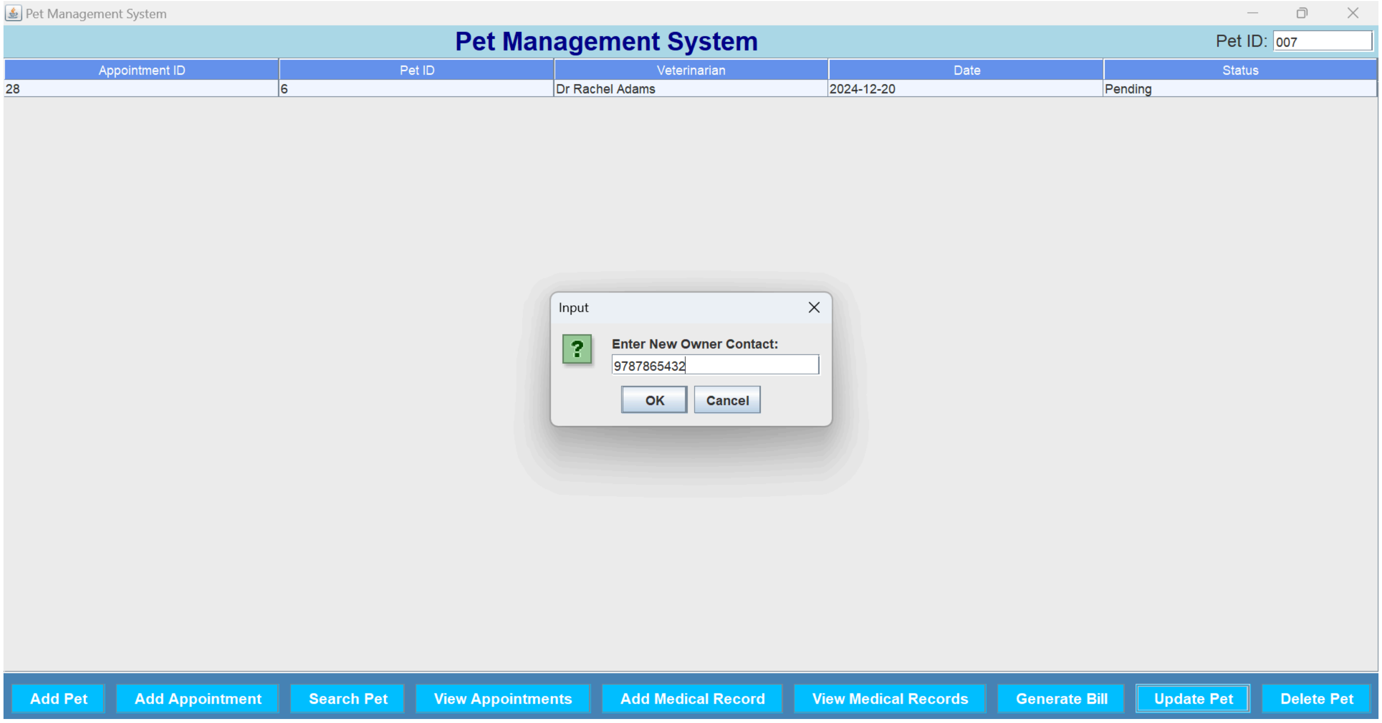


Pet Management System

Pet ID: 007

Appointment ID	Pet ID	Veterinarian	Date	Status
28	6	Dr Rachel Adams	2024-12-20	Pending

Enter New Owner Name:
Amial



Pet Management System

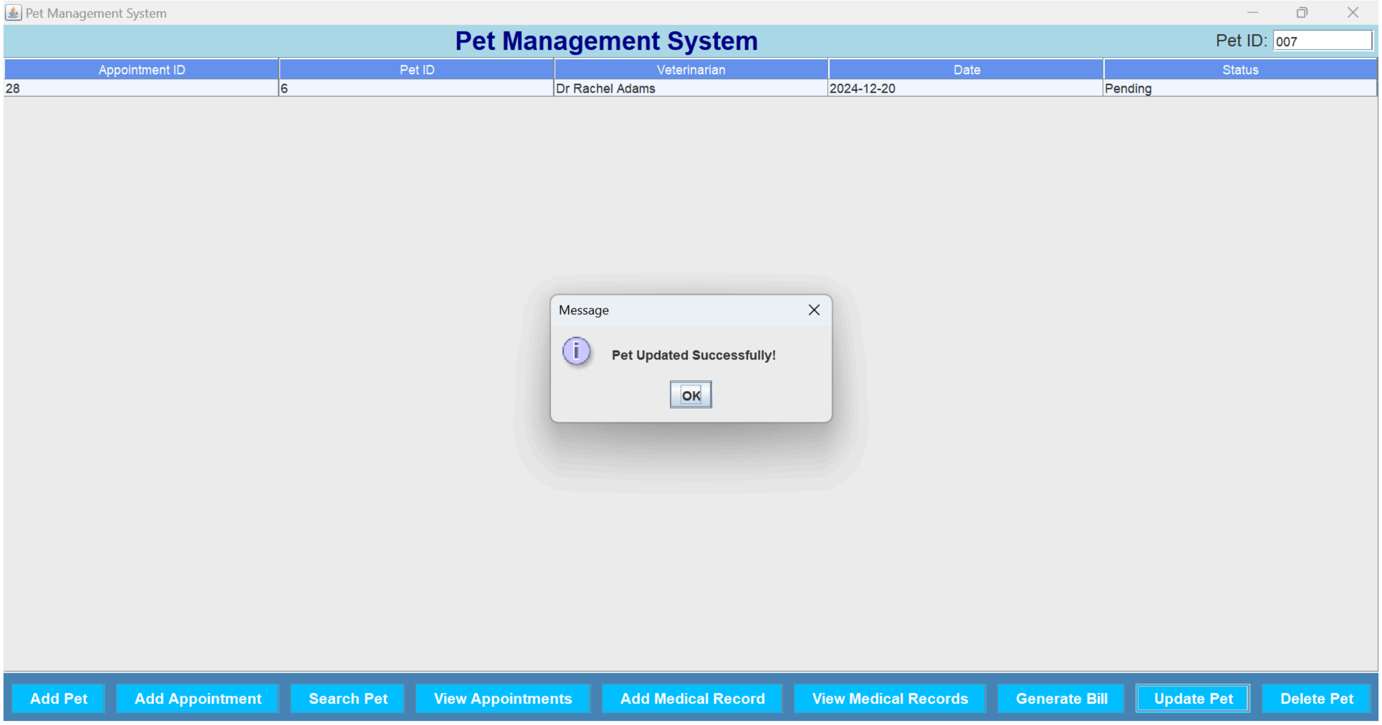
Pet ID: 007

Appointment ID	Pet ID	Veterinarian	Date	Status
28	6	Dr Rachel Adams	2024-12-20	Pending

Input

Enter New Owner Contact:

9787865432



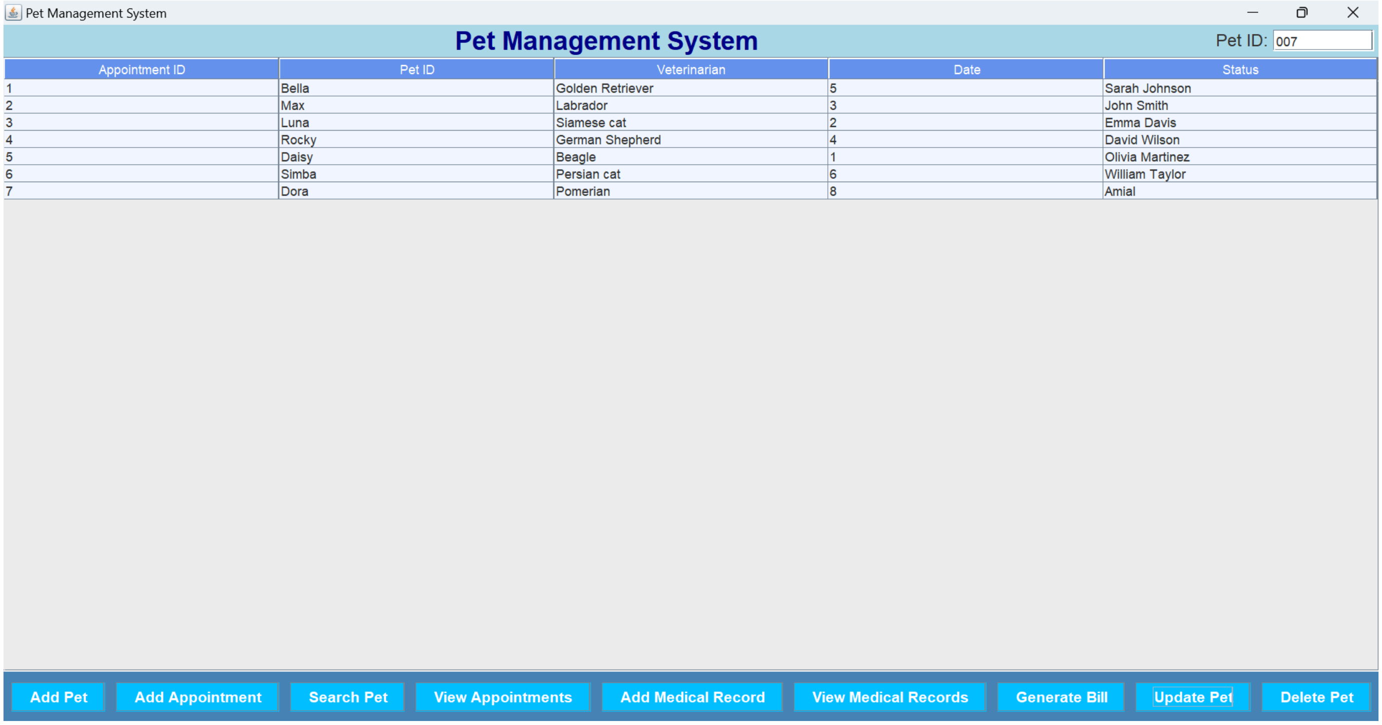
Pet Management System

Pet ID: 007

Appointment ID	Pet ID	Veterinarian	Date	Status
28	6	Dr Rachel Adams	2024-12-20	Pending

Message

Pet Updated Successfully!

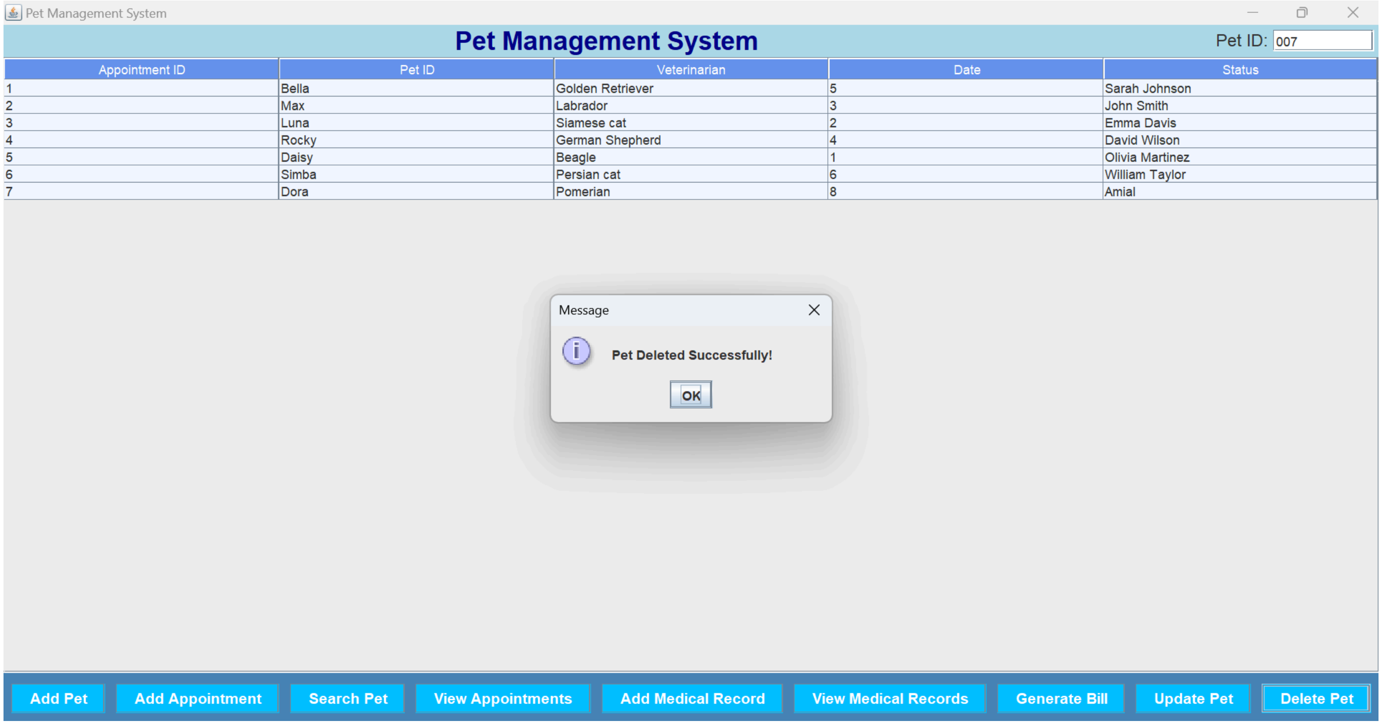


Pet Management System

Pet ID: 007

Appointment ID	Pet ID	Veterinarian	Date	Status
1	Bella	Golden Retriever	5	Sarah Johnson
2	Max	Labrador	3	John Smith
3	Luna	Siamese cat	2	Emma Davis
4	Rocky	German Shepherd	4	David Wilson
5	Daisy	Beagle	1	Olivia Martinez
6	Simba	Persian cat	6	William Taylor
7	Dora	Pomerian	8	Amial

vi)Delete:

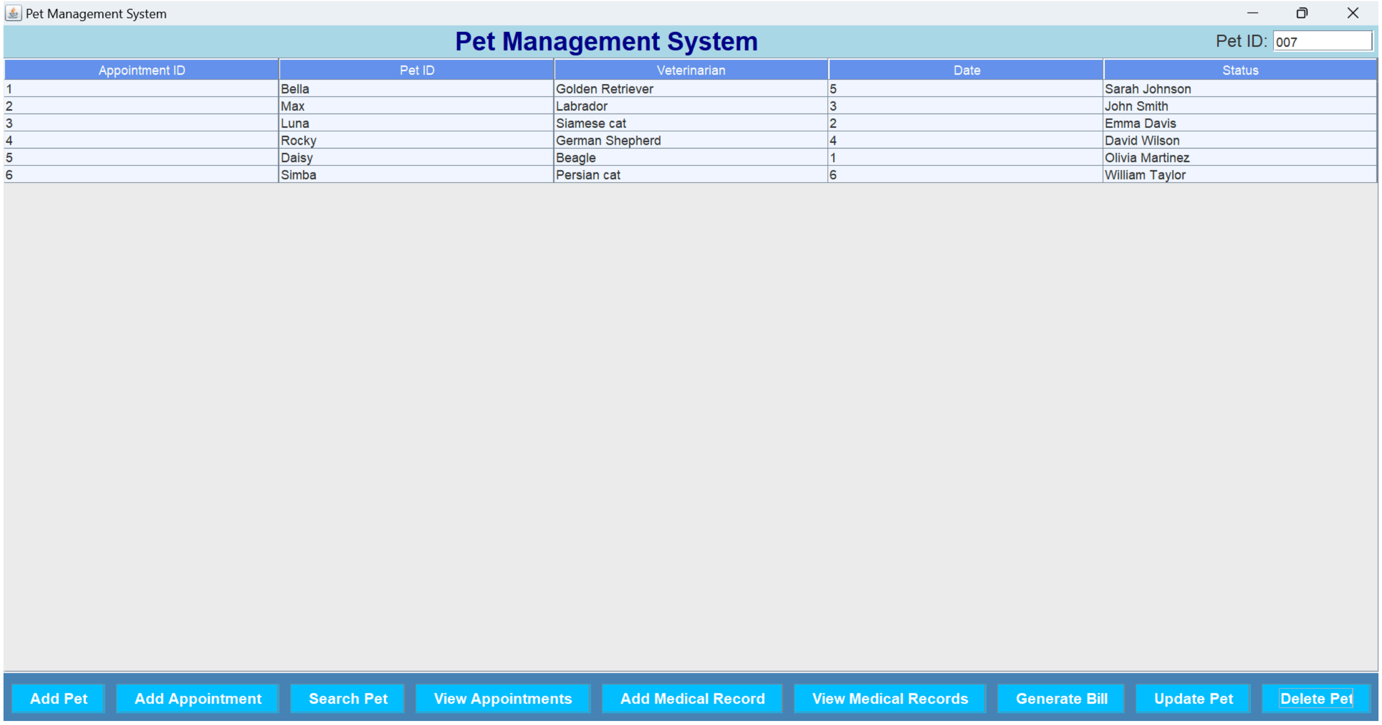


Pet Management System

Pet ID: 007

Appointment ID	Pet ID	Veterinarian	Date	Status
1	Bella	Golden Retriever	5	Sarah Johnson
2	Max	Labrador	3	John Smith
3	Luna	Siamese cat	2	Emma Davis
4	Rocky	German Shepherd	4	David Wilson
5	Daisy	Beagle	1	Olivia Martinez
6	Simba	Persian cat	6	William Taylor
7	Dora	Pomerian	8	Amial

Pet Deleted Successfully!



Pet Management System

Pet ID: 007

Appointment ID	Pet ID	Veterinarian	Date	Status
1	Bella	Golden Retriever	5	Sarah Johnson
2	Max	Labrador	3	John Smith
3	Luna	Siamese cat	2	Emma Davis
4	Rocky	German Shepherd	4	David Wilson
5	Daisy	Beagle	1	Olivia Martinez
6	Simba	Persian cat	6	William Taylor

CONCLUSION

The Pet Management System is a well-rounded desktop application that excels in managing the complex data needs of veterinary clinics. With its core functionality, including the ability to add, update, and delete pet and owner records, it efficiently addresses the key tasks required for day-to-day operations. The system's design ensures that staff with varying levels of technical expertise can navigate it with ease, providing an intuitive user experience.

Built with a robust technology stack of Java, JDBC, and an Oracle database, the application is both reliable and scalable. Its secure architecture, featuring role-based access control and modular design, ensures that the system can grow and adapt as future requirements emerge. This makes the application not only efficient but also secure and maintainable over time.

Looking ahead, the Pet Management System shows great potential for expansion. Features like appointment scheduling, payment system integration, and mobile/web extensions can be added to further enhance its capabilities. These future improvements, combined with the solid foundation of the current system, make it a promising tool for real-world veterinary clinics, providing flexibility for continuous development.

REFERENCES

1)https://docs.oracle.com/en/database/oracle/oracle-database/

2)https://www.amazon.com/Head-First-Java-Kathy-Sierra/dp/0596009208

3)https://www.amazon.com/Java-Swing-Beginners-Guide-Programming/dp/0072224842

4)https://docs.oracle.com/javase/tutorial/uiswing/

5)https://www.amazon.com/Java-Database-Programming-JDBC-Developers/dp/1565924232