Fall 2023: CS5200 Final Project Report - Wee Veterinary Clinic

Group Name: ShiYWangBZhangY
Group Members: Beini Wang, Yi Shi, Yue Zhang

Link to the Final Project Presentation:
https://drive.google.com/file/d/1XFNyTRLiGY8O1D5NRFDEs-nFHDG6qjc2/view?usp=sharing

Database Introduction

This project builds a management system for Wee Veterinary Clinic. The system is based on data of clinic organization, facilities, services, clients, appointments and activities. We keep track of all the data in daily operation to help make improvement on resource allocation, service quality and financial performance. The ideal users of this database are clinic employees, including admin staff and veterinarians and visitors with pets. For clinic staff, they have access to the appointment data regarding the clinic branch they work at and vets are authorized to view their appointments.

Management on the system includes create, read, update and delete. Registered pet owners have access to the data system for all the appointments info, treatment records of their pets, invoices and can pay their invoices. New clients, pet owners can use the sign up feature to sign up. Admin staff have access to all appointments data regarding the clinic branch they work at and can make and cancel appointments for the registered pet owners. When the pet owner with a pet(patient) shows up for the appointment, they are authorized to check-in for them and mark the appointment as show-up as an update. Vets have the access to all assigned appointments of theirs and available treatments, added treatments of the appointment. They are authorized to add, delete treatments, medications of the appointments, which will update the invoice amount at the same time and can complete the appointment by releasing the room assigned to it.

Bonus work for the project: Complicated schema, additional front end GUI, multiple user roles

Description of updated conceptual design

Every branch of Wee Veterinary Clinic has a unique clinic id, an address(street_number, street_name, town, state, zip code), a landline phone number and a clinic type (ex. emergency, regular).
Each branch has many employees, but each employee can only work in one branch. Each employee is associated with a unique employee id, a contact address and position type. The clinic also includes several vets, each has a unique license number, certification file and description of specialization.
Each branch maintains several rooms, each associated with a room number and state of use.
Each branch also provides a series of treatment services (unique service id, service name and regular price) and each treatment service may include a series of treatments.
Each appointment is associated with a unique appointment id, a record of appointment date time, a description and a showup boolean that indicates if this is a shows-up appointment. Each

appointment may have many treatment services with an actual charge and is assigned to one vet and happens in one room in the clinic.
The branch has a storage of medication that may be used in a treatment. Each treatment may contain several types of medication with a given quantity. Each type of medication has a unique scientific name, description of utility, a retail price and time interval indicated in days number.
An owner's pet may have appointments in many clinic branches, and each appointment is with one pet and in one clinic branch. Each pet owner has a unique id, name, a credit card (card number, card type) and corresponding billing address. Each pet belongs to one pet category which is associated with a unique category id, a breed name and a description.
Invoice is issued when a pet shows up for the appointment and is dependent on the appointment, it has an issued date, a received amount and date from the system log and a total payment for one pet per visit. The total amount on the invoice can be traced through all the treatment service and medication charges for the appointment. The owners pay for their invoices.

Application Language

Software: SQL Workbench, interillJIDA

Storage: We choose SQL Storage

Language: We will use java for building up the back-end and also java to build GUI as the front-end, which allows the users to interact with, for this project.

Library used for building GUI:

```
javax.swing.*;
java.awt.*;
java.awt.event.ActionEvent;
ava.awt.event.ActionListener;
java.sql.CallableStatement;
java.sql.Connection;
java.sql.ResultSet;
java.sql.SQLException;
java.sql.Timestamp;
java.sql.Types;
java.text.SimpleDateFormat;
java.util.List;
java.util.ArrayList;
java.sql.Date;
java.sql.DriverManager;
```

Machine Restrictions: There are no strict restrictions for the hardware and machine.

Why does this project or this data domain interest you?

We want to build a clear database system that helps all pet-loving people understand how pet clinics work and how to make full use of the resources in the clinic. Since databases are widely used in managing a workflow of commercial activities, such as trades, inventory management and so on, a veterinary system is also a good topic for us to practice to get familiar with this kind of database management, which is fit in with our normal life. It can be a good foundation for us to better know the daily database application in the future. As pet lovers, it's a two-for-one deal.

README:

Overview:

- WePC Veterinary Management System is a comprehensive Java-based application designed to manage various aspects of a veterinary clinic. This system includes separate interfaces for pet owners, veterinarians, and administrative employees, each with tailored functionalities. GUI and Command-Line Interface applications are available and which offer the same functionalities.

GUI components

- WePCGUI: The main entry point of the application, handling user sign-in and sign-up.
- PetOwnerGUI: Interface for pet owners to view pet information, appointments, and view and pay invoices.
- VetGUI: Interface for veterinarians to view appointments, manage it with adding, deleting treatments, and medications.
- EmpGUI: Interface for administrative employees to manage appointments and access detailed appointment information.

Prerequisites

- Java Runtime Environment, MySQL database server and with the we_pc dump file imported. JDBC driver for MySQL, you can find the jar file in the lib folder.

STEP 0: Preparation

- Create the MySQL Database for this project on your computer if you don’t have it yet.
- Open the Dump file and run the code for create tables, insert data and create procedures and functions.
- Open the Java project and connect your IDE with the ‘mysql-connector-j-8.2.0.jar’ as dependency.
- How to connect your IntelliJ IDE with the sql java connector:
    a. Open IntelliJ IDEA and load your project.

b. Go to the "File" menu and select "Project Structure."
c. In the "Project Structure" dialog, select "Libraries" under the "Project Settings" section.
d. Click the "+" button to add a new library.
e. Select "Java" from the dropdown menu.
f. Navigate to the location where you downloaded the JDBC driver and select the JDBC driver JAR file.
g. Click "OK" to add the JAR file to your project.

STEP 1: Connect to your MySQL Database and start the application

- Open the java program and Run the Main.java
- If CLI is preferred, you may also run we_pc.java to start the CLI application.
- The Java GUI will show up, type in your database username and password to connect with your DB, if wrong, the project will automatically stop.

STEP 2: Signup as a Pet Owner User

- Although there is already some pet owner data inside the DB, you can sign up for a new pet owner user. You can sign up a new user by clicking the Sign up button, at this point, you can only sign up as a pet owner user.
- If your sign up information is not correct, the error will show up, you need to re-sign up.
- When you sign up successfully by clicking the sign up button, the username for a pet owner will automatically add ‘CUS’ , so when a pet owner login, he needs to add ‘CUS’ in front of his username. For example, he signs up ‘user1’ as his username, when logging in, he should type ‘CUSuser1’ as his log in username.

STEP 3: Log in

- If the username starts with ‘CUS’, it means the user is a pet owner. The it will turn to the pet owner user page.
- If the username starts with a number, then the user is an employee, the start number following with ‘emp’ represents the clinic that the employee belongs to. For example, when the username of an employee is ‘01emp00008’, then this employee belongs to the clinic with id 1.
- If the username is an employee username, then it will check whether it is a normal employee or a vet. When it checks that the user is a vet, then it will turn to the vet page, if the employee is not a vet, then it will turn to the employ page.

Pet Owner Page:

- **Sample owner credential to login:**
  Username: CUSuser1

Password: password123

Username: CUSuser3
Password: securepass1

- **Pet owner will have 4 choices:**

1. **View Pet Information Button (display_pet_info procedure):** will show all the pets you have and their information, which includes pet id, name, breed, height, weight and status.
2. **View Appointment Information Button (display_appointments_for_owner):** it will display all the appointment you have, normal employee of the clinic can make new appointment for the pet owner, and for each appointment, you have 3 choices:
    a. **View invoice (display_invoices_for_appointment procedure):** you can view the invoice information, which includes appointment ID, time, Description, patient ID, pet name, clinic id, show up status, vet id, vet name and room id. You can see the name of the treatment/ medication, the description, price for it.
    b. **View the treatments (display_treatments_and_medications_for_appointment procedure):** treatments and medications will display in this section.
    c. **Pay for the invoice (update_payment_received):** you can type in the amount you will pay for the invoice, then the invoice's received amount and date will change. If the pay amount is bigger than the total you need to pay, then the received amount will equal the total amount.
3. **Reset your password Button (update_owner_password procedure):** the user can reset his password.
4. **Sign out Button:** the program will end.

Admin Employee Page

- **Log In:**
  Given the employee table, there are two employee types (emp_type field), the administrative employee is authorized to specific procedures. Here two examples of employee id and password are provided for log-in.

  Username: 05emp00014
  Password: coolpet

  Username: 01emp00011
  Password: jaych08
  The Employee class contains several methods that can be used by only an administrative staff in terms of making a new appointment, deleting an appointment, displaying all the appointments in the clinic and displaying the detailed information. On SQL level, the procedures and functions created for administrative staff have staff username (unique) as

the input parameter so that all the reads and modifications are based on data of the clinic where the staff works.

1. **Make an appointment:** the staff should first ensure the schedule is not full for the datetime provided by the pet owner. A function (is_book_full) is created to check if the number of vets that work in the clinic are not all booked on the specific time, which indicates the appointment is available. Another function (veterinarian_match) returns one of the available vets to the appointment. It is worth noticing that after a tuple is inserted into the appointment table, the show-up field value is 0 and the room field value is null. The state of both fields will only change when the pet shows up as scheduled. The procedure (make_appointment) signals error when making duplicate appointments or the pet id is invalid.

2. **Cancel an appointment:** the procedure (cancel_appointment) finds the appointment by the given time and pet id. Given the appointment time, clinic id and pet id form the alternative key in the appointment table, the target appointment will be found if it already exists. Also, the application is designed in a way that all appointments associated with expenditure should never be deleted through this procedure. Thus, the procedure signals error when an appointment is not found or when an appointment is associated with treatment expenditure.

3. **Display all the appointments:** the procedure (display_appointment_for_employee) displays all the basic information for the staff by joining tables on references. It may signal error when the staff can't be traced to the clinic id, which is possible in real life when an employee loses authorization to the database in the clinic.

4. **Display appointment for details:** the procedure (display_appointment_detail) provides all information about one appointment for the employee. Given some appointments are not associated with invoices since the pet didn't show up, a left outer join with the invoice table is required to implement the procedure. The procedure signals error when the employee is trying to read an appointment that doesn't belong to the clinic or when employee id is invalid.

5. **Show-up Button:** when a new appointment is created, the show_up field is valued at 0 for the appointment as well as the roomID. This is the initial state of a new appointment. If the pet shows up as scheduled, the admin staff needs to click the show-up button so that an invoice is created for the pet, a room is assigned to the pet with a room id and the show_up value gets updated into 1.

On Java level, the application handles errors with wrong inputs during interaction with the user by the try-catch technique overall.

Vet Page:

- **Sample vet credential to login:**
  Username: 05emp00010
  Password: dogdd34

  Username: 02emp00003
  Password: newmoon95

  Vets are authorized to read and write treatments and medications on their appointments and release rooms when they complete their appointment.

1. **Appointment Management:** View and select veterinary appointments.
   Vets will need to first select an appointment to proceed to the following options as the vet manages the treatments and medications record for the appointments. Click on Select appointment and all appointments that are assigned to the vet with the credential will be displayed, click on one, that appointment will be selected. The header of the window will show the selected appointment id.
2. **Treatment Handling:** Add, view, and delete treatments for selected appointments.
   Vet have the options to view available treatment services that have not been added yet for the appointment/add treatment with an actual charge to the selected appointment. Click on add a treatment, application will ask for valid treatment service id and actual charge for the treatment. If the inputs are valid, the treatment id, the charge will also be added to the total amount on the appointment invoice on invoice table.
   Vet also views added treatments of the selected appointment/delete treatments added if the pet owner refused, the charge on the treatment will be deducted on the total amount on the appointment invoice.
3. **Medication Management:** Add medications and view available medications for pets.
   Vet have options to view available medications that have not been prescribed for this appointment/prescribe a medication with a quantity and a time interval in days. The total charge of the medication will be added on the invoice.
4. **Complete an appointment:** When the pet shows up, an available room will be assigned to the appointment and when the vet completes the appointment, the room will be released and the state of use would be available. If the room has already been released, the application will show the appointment has been completed or it’s a no show appointment.

Lesson learned:

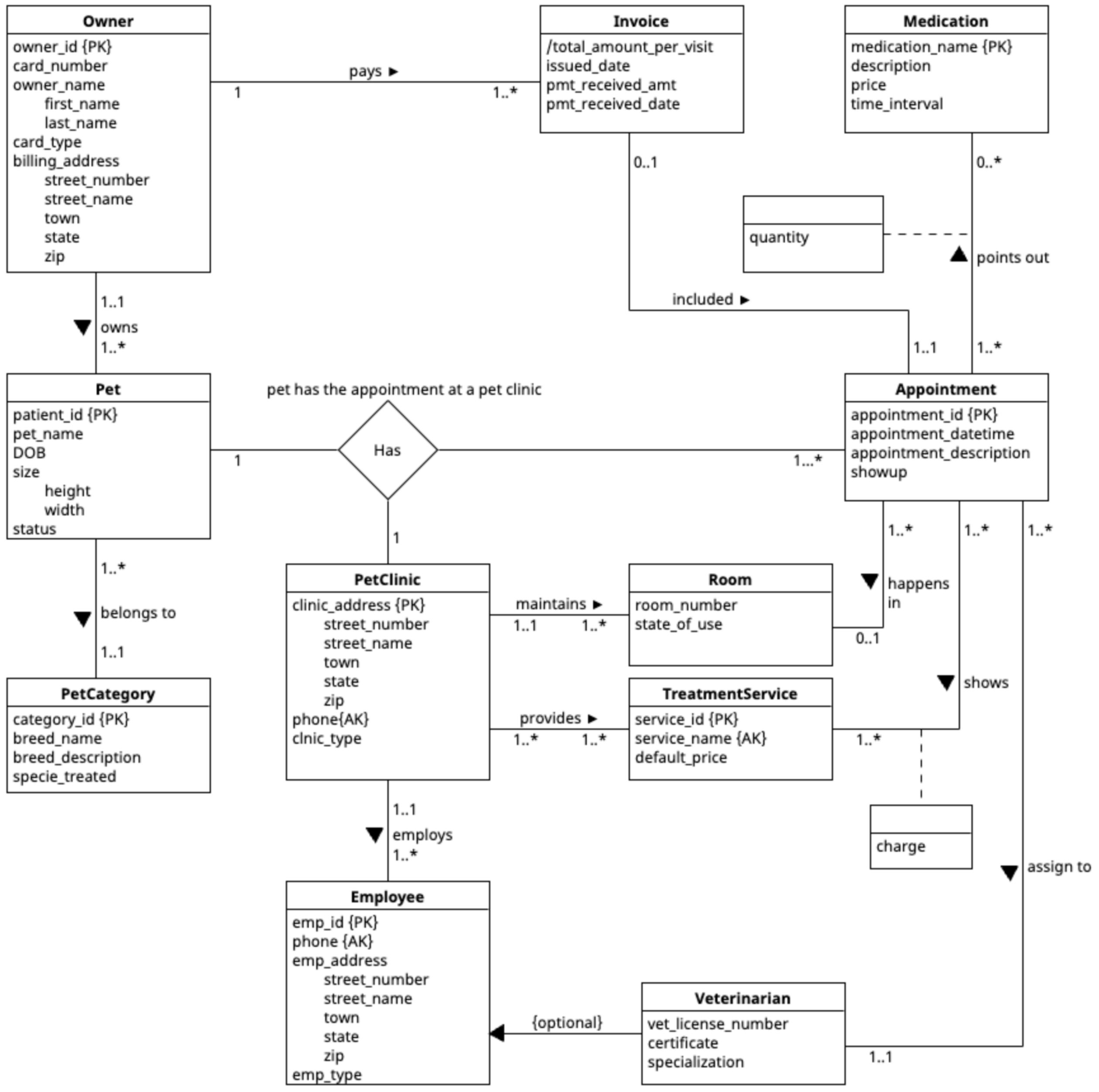
- When transitioning from a SQL database with stored procedures, functions to a CLI (Command Line Interface) and eventually adapting it to a Java Swing GUI, there are several valuable lessons and insights that can be gained from the process.
- **Technical expertise gained:** Better understanding on how to create and effectively use stored procedures in SQL. Enhanced Java skills, particularly in Swing for GUI development, and understanding the nuances of event-driven programming. Gained understanding on integrating a backend database with a frontend application. Advanced techniques in managing errors and exceptions in both Java and SQL by using try and catch mechanisms and using SIGNAL SQLSTATE statement to signal self-defined errors in SQL.
- **Insight:** Importance of allocating sufficient time for design, development, and testing, especially when dealing with GUIs which are more time-consuming than CLI. And the testing also plays an important part of building a team project. To synchronize everyone's work and make sure each member is on the same page and complete a whole project is time consuming and is different from an individual's project. We learned how to quickly read other people's code and communicate efficiently. For example, the more pre-inserted data we put into the schema, the more efficiency we can achieve in error-handling and testing.
- **Alternative Design:** Consideration of using other technologies for GUI, such as JavaFX, which might offer more features or better performance than Swing.The project can also be implemented with visualization tools as more data is inserted into the system so that the analysis of data will play a more important role.
- **Document any code not working in this section:**
  In creating methods to call the sql procedures and where print the result set is not needed, the update doesn't show in the sql database. Later I realized that I didn't have "stmt.executeQuery();" to actually execute the query. And when I debugged the method, I realized I didn't have the line and solved the update issue.

Future Work:

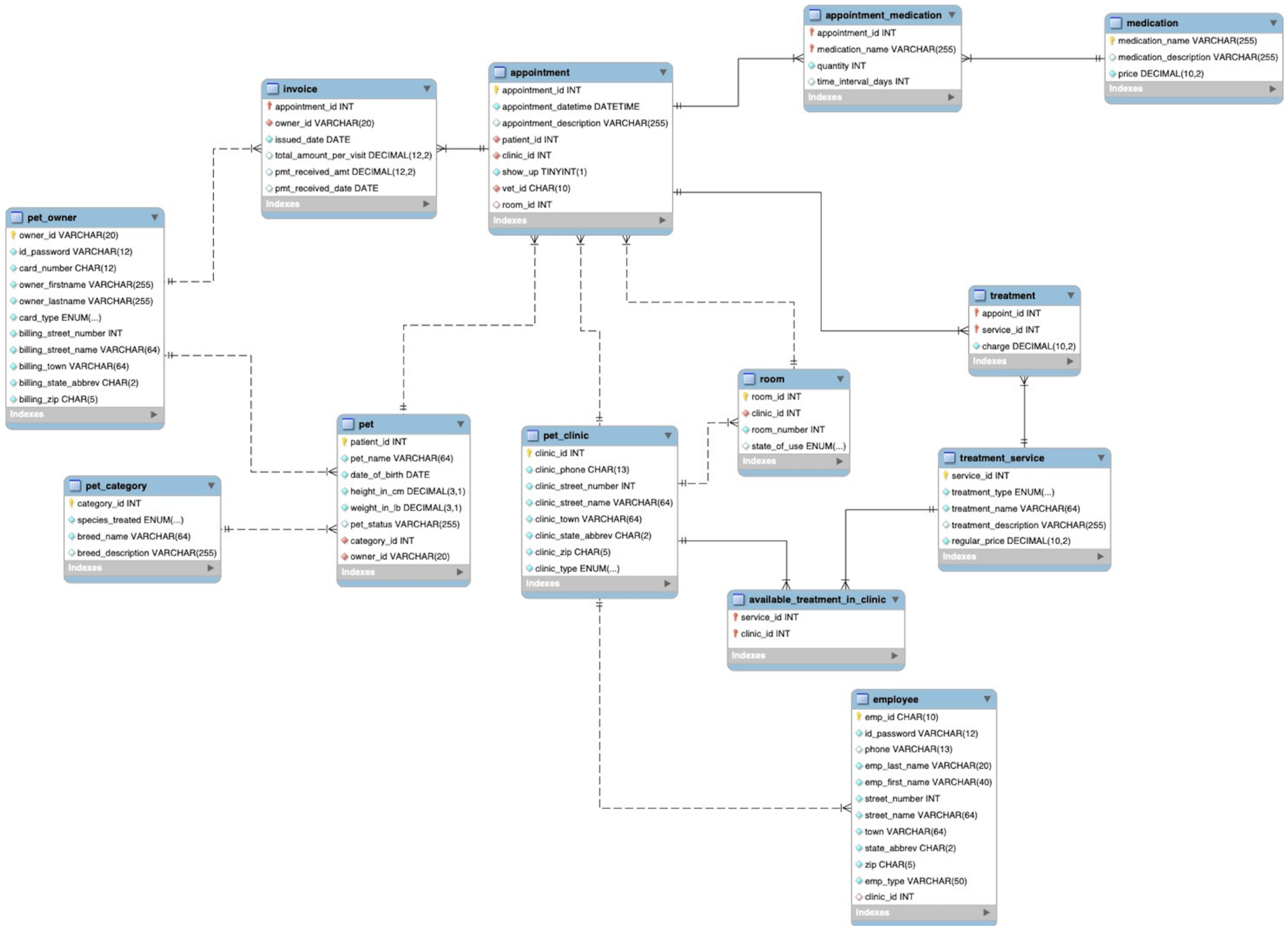
- **More Analytical Functionality to Existing data:**The project can be implemented with more functionality since not all current fields are used in the program. For example, all the appointments contain the appointment date so that analyzing the number of appointments made by each pet id can help the clinic decide its most valuable client. Also, the number of appointments each vet has involved with can help the clinic decide its most valuable employee. For another, The cancel appointment procedure can be further complicated by keeping all the expired appointments (pet not showing up case) so that it can be used to set up a priority list for existing clients, improving the efficiency in making appointments.

- **User Interface and Experience Improvement:** Web applications can be accessed from any device which offers more flexibility and the web user interface is more dynamic and responsive. Also the web applications are more open to the updates where Java GUI or CLI have limitations because of the original designs on the project.
- **Scalability of the application:** The current read and display of data are only based on the mock data with limited size. NoSQL databases and Web applications can be more easily scaled to handle an increasing number of users or data load, especially if hosted on cloud services. It might be necessary to make the transition.
- **Performance Optimization:** There are possible overlaps on the functionality and procedures created in this project, so optimization on the SQL procedures and Java designs are necessary for aiming on faster response time.
- **Security Enhancement:** Since the username and password are all stored in the database, it's crucial to separate the user information from other data in the system. For example, some data such as contact information of the employees and customers, payment information are more likely to be targeted by hackers in real life. Future work may involve setting hierarchy levels of data with different levels of recovery management and application views.

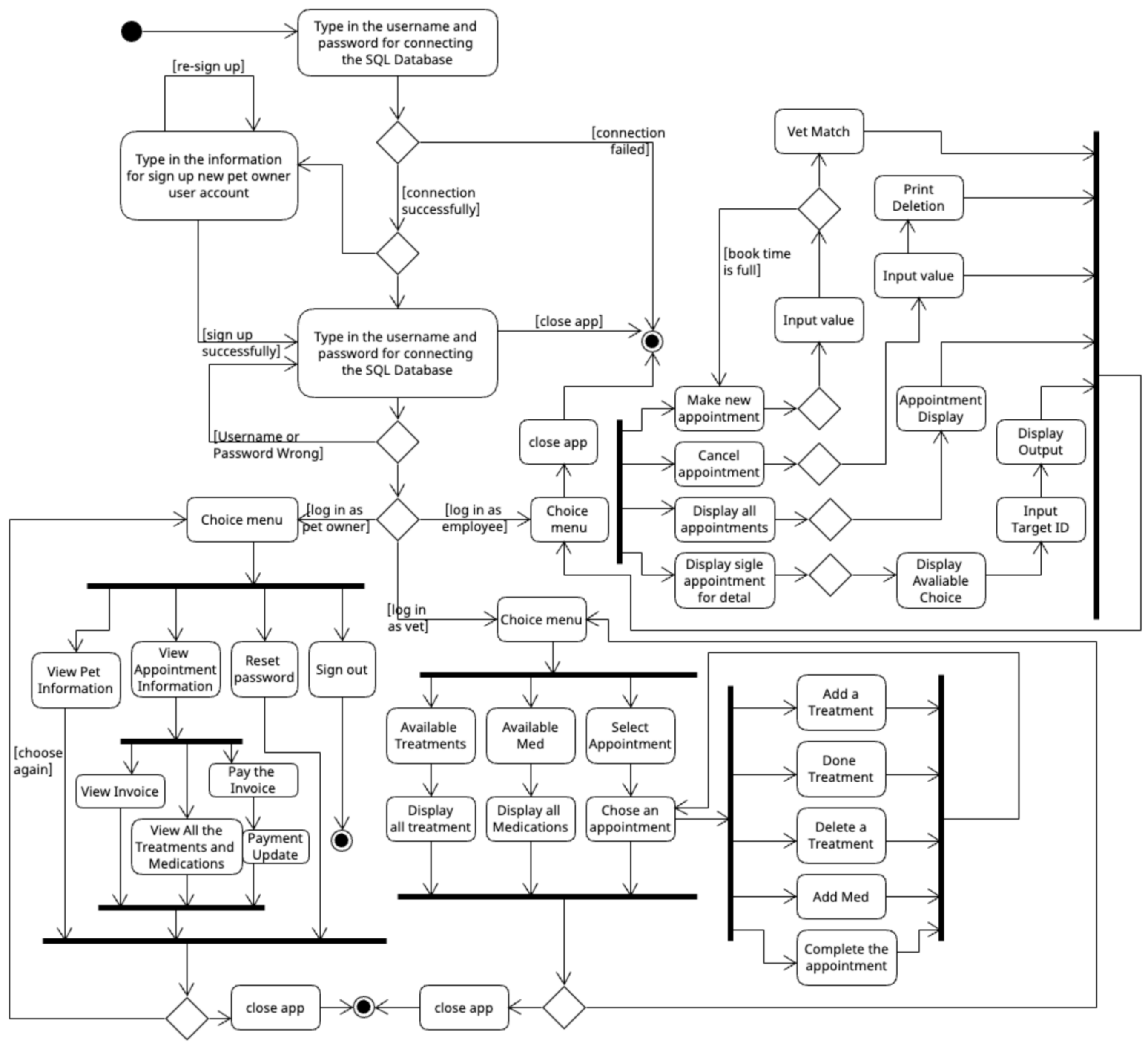
The UML Diagram of the Conceptual Design:



Logical Design for the Submitted Database Schema：



User Flow Chart of the System:


Type in the username and password for connecting the SQL Database
[re-sign up]
Type in the information for sign up new pet owner user account
[connection failed]
[connection successfully]
[sign up successfully]
Type in the username and password for connecting the SQL Database
[close app]
[Username or Password Wrong]
[log in as pet owner]
[log in as employee]
[log in as vet]
Choice menu
Choice menu
Choice menu
close app
Vet Match
Print Deletion
[book time is full]
Input value
Input value
Make new appointment
Cancel appointment
Display all appointments
Display sigle appointment for detal
Appointment Display
Display Output
Input Target ID
Display Avaliable Choice
View Pet Information
View Appointment Information
Reset password
Sign out
[choose again]
View Invoice
Pay the Invoice
View All the Treatments and Medications
Payment Update
Available Treatments
Available Med
Select Appointment
Display all treatment
Display all Medications
Chose an appointment
Add a Treatment
Done Treatment
Delete a Treatment
Add Med
Complete the appointment
close app
close app