**Group Project Documentation   
Advanced Cross-Platform Application Programming with .NET**

**Pet Health Care System**

**Prepared by Group 2**

***Bùi Duy Anh***

***Trần Đình Minh Quang***

***Nguyễn Đặng Phương Nam***

***Lê Hoàng Sơn***

**Ho Chi Minh City, 2024**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1.** **Project Introduction 1**

1.1 Product Perspective 1

1.2 User Classes and Characteristics 1

**2.** **Database Design 1**

**3.** **System Architecture 1**

**4.** **Implementation 2**

4.1. Deployment Considerations [2](#_2s8eyo1)

4.2. Screenshots and explanations [2](#_17dp8vu)

**5.** **References 2**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Quang | 20 May 2024 | Initial create | 1.0 |
| Quang | 10 Jul 2024 | Add BR | 1.0 |
| Anh | 10 Jul 2024 | Add references | 1.0 |
|  |  |  |  |

# **Project Introduction**

## **Product Perspective**

This project proposes a pet health care system, a B2C (Business-to-Consumer) platform that simplifies pet care for owners. The system allows users to conveniently book appointments online, manage their pet's medical records, and access relevant information - all in one place. This not only empowers pet owners to be proactive about their furry companions' health but also streamlines operations for veterinary clinics and staff by providing a centralized system for pet medical data.

## **User Classes and Characteristics**

User (Customer/Staff/Veterinarian/Admin) authentication by Email/Password

I Customer

II Staff

III Veterinary

IV Admin

1. **Customer:**
2. Login
3. Manage Pet
4. Book, View, Cancel, Appointment
5. View Service
6. Modify profile info

**II. Staff:**

1. Login
2. Manage Customer
3. Manage Appointment
4. Manage Pet
5. View Medical Record
6. Manage Hospitalization

**III. Vet:**

1. Login
2. View TimeTable (Appointment + Hospitalization time list)
3. Manage Medical Record

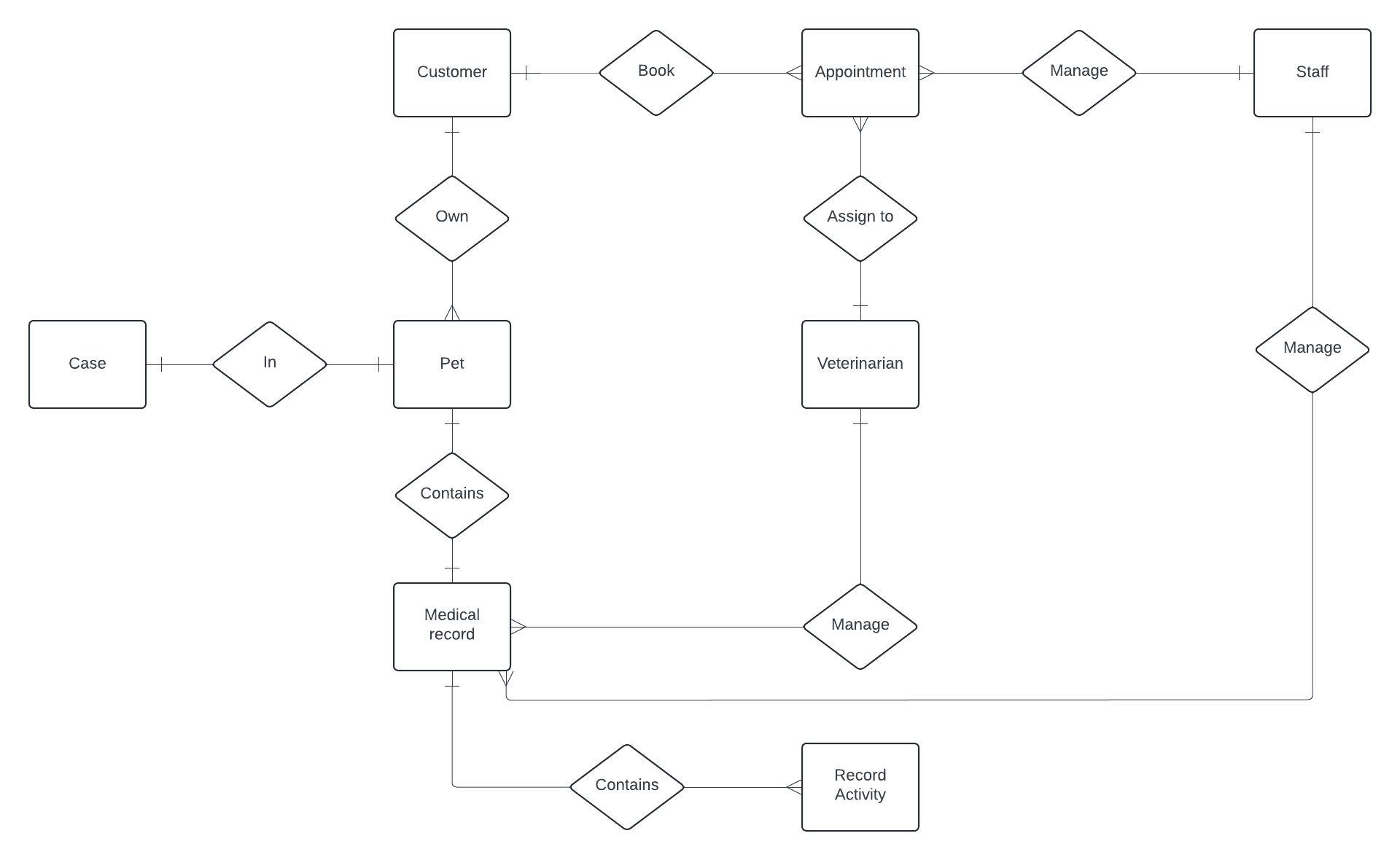
**IV. Admin:**

1. Login
2. View Appointment List
3. Manage Account
4. Manage Service
5. Manage Items
6. Manage System Configuration

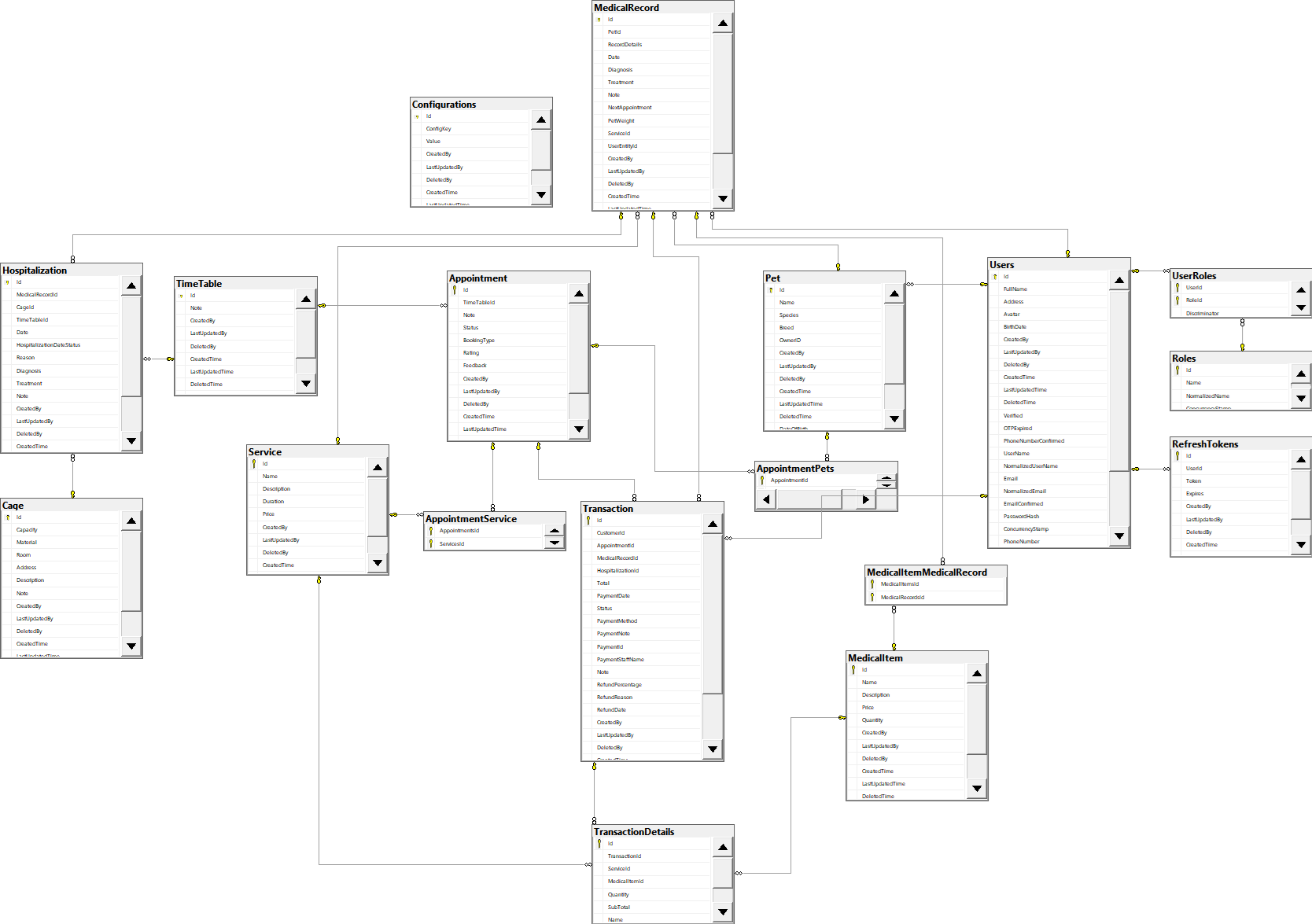
### 

| **ID** | **Rule definition** |
| --- | --- |
| BR-01 | The system must validate all the guests’ sign-up information before creating the guest accounts. |
| BR-02 | All information entered must comply with the constraints given by the system. |
| BR-03 | Users are only allowed to choose an appointment time based on the doctor's work shift suggested by the system. |
| BR-04 | The system must ensure that user has the rights to use authenticated features |
| BR-05 | The system must ensure that the healing progress data is up-to-date and accurate. |
| BR-06 | The system must ensure that the feedback data is validated and recorded accurately. |
| BR-07 | Feedback should be linked to the specific appointment with the veterinarian. |
| BR-08 | The system must ensure that the appointment data is validated and recorded accurately. |
| BR-09 | Appointment slots must be checked for availability before confirming the booking. |
| BR-10 | The system must ensure that the user's session is properly terminated. |
| BR-11 | The system should securely clear any session-related data. |
| BR-12 | Ensure data to be private and secured. |
| BR-13 | Any change on a pet medical record must be notified to the customer. |
| BR-14 | Data reports can not be deleted, if they somehow got modified, all previous versions are still kept. |
| BR-15 | The system must ensure that a room with existing bookings cannot be deleted. |
| BR-16 | Deletions must be confirmed by the staff member to prevent accidental removal. |
| BR-17 | The system must log all deletion activities for audit purposes. |
| BR-18 | Room deletion should adhere to the data retention policies defined by the organisation. |
| BR-19 | The system must ensure that a vet cannot be assigned to more than one room at the same time. |
| BR-20 | The system must log all assignment activities for audit purposes. |
| BR-21 | Room and vet assignments should adhere to the scheduling and availability policies defined by the organisation. |
| BR-22 | The system should be displayed in a user-friendly format |
| BR-23 | The system must log all actions related to viewing appointments for audit purposes. |
| BR-24 | Staff must validate the customer's identity before scheduling an appointment. |
| BR-25 | The system must ensure no double-booking of appointment slots. |
| BR-26 | All appointments must be confirmed with the customer before finalising. |
| BR-27 | The data must be retrieved in accordance with the data access and privacy policies defined by the organisation. |
| BR-28 | Appointment details should be accessible for modification or cancellation by the staff up to 24 hours before the appointment time. |
| BR-29 | Staff must follow data privacy regulations when handling customer information. |
| BR-30 | The system must ensure that only valid appointments can be cancelled. |
| BR-31 | Cancellations must be confirmed by the staff member to prevent accidental cancellations. |
| BR-32 | The system must log all cancellation activities for audit purposes. |
| BR-33 | Appointment cancellations should adhere to the organisation’s cancellation policies and procedures. |
| BR-34 | Only authorised staff members can assign or reassign veterinarians to rooms. |
| BR-35 | The system must prevent double-booking of both vets and rooms. |
| BR-36 | Assignments must be confirmed with both the veterinarian and the room scheduler. |
| BR-37 | Any changes to assignments must be logged and communicated to the affected parties immediately. |
| BR-38 | The system must comply with all applicable regulations regarding staff scheduling and room management. |
| BR-39 | Only authenticated veterinarians can view their appointment schedules. |
| BR-40 | The system must ensure that the appointment data is accurate and up-to-date. |
| BR-41 | Appointment details must comply with data privacy regulations. |
| BR-42 | The system must provide an option for veterinarians to print or export their schedule. |
| BR-43 | Any changes to the schedule (e.g., cancellations, rescheduling) must be promptly reflected in the system. |
| BR-44 | Only authenticated veterinarians can add or edit medical records. |
| BR-45 | All medical records must comply with data privacy and security regulations. |
| BR-46 | The system must ensure that required fields are completed before allowing the record to be saved. |
| BR-47 | Changes to the medical record must be logged with a timestamp and the user’s ID. |
| BR-48 | The system should allow veterinarians to update or append additional information to the medical record if needed |
| BR-49 | Specific data formats may be required for certain service information fields (e.g., date format, phone number format). |
| BR-50 | Certain service information fields may be mandatory and cannot be left blank. (e.g., service description, pricing, availability) |
| BR-51 | Staff permissions may restrict which service information fields they can edit. |
| BR-52 | A service cannot be deleted if it has any active bookings associated with it. |
| BR-53 | The system should ensure that deleting a service does not cause any data inconsistencies or corrupt other system data. |
| BR-54 | The price must be entered in a specific format (e.g., currency symbol, decimal places). |
| BR-55 | **Impact on Existing Subscriptions/Bookings:** Define how price changes will affect existing subscriptions or bookings. Options include:   * No impact: Existing subscriptions/bookings continue at the original price they were booked under. * Prorated adjustment: The price change is prorated for existing subscriptions to reflect the remaining duration. (This may require additional business logic) * Customer notification: The system may require notifying customers about upcoming price changes for their existing subscriptions/bookings. |

# **Database Design**

*ERDConceptual*

*Database Diagram*



# **System Architecture**

## **System Environment**

**Hardware Platform:** Web-based application accessible through various devices like desktops, laptops, tablets, and smartphones with internet access.

**Operating Systems:**

* Client-side (user devices): No specific requirement, will function on most popular operating systems (Windows, macOS, Android, iOS).
* Server-side: Operating system will depend on the deployment platform. If using Azure, it will be a cloud-based virtual machine with a server OS like Windows Server or Linux.

**Geographical Locations:**

* Users: Accessible globally with internet access.
* Servers and Database: Can be deployed on-premises or in a cloud platform like Azure. Location can be chosen based on factors like user base concentration, data privacy regulations, and latency requirements.

**Organizations Hosting Infrastructure:**

* Database: Primarily hosted on a SQL Server instance (can be on-premises or on Azure).
* Servers and Website: Can be hosted on-premises servers or a cloud platform like Azure App Service.

## **System Architecture (3-Layer with Razor)**

This system will utilize a 3-layer architecture with Razor as the frontend framework. Here's a breakdown of the layers:

1. **Presentation Layer (Razor):**
   * Developed using Razor, a web framework that allows creating interactive web UIs using C#.
   * Responsible for handling user interactions, displaying data, and sending requests to the Business Logic Layer.
2. **Business Logic Layer:**
   * Written in C#.
   * Contains the core application logic for processing user requests, interacting with the Data Access Layer, and managing business rules.
   * Directly interacts with the Data Access Layer.
3. **Data Access Layer:**
   * Responsible for interacting with the database (SQL Server).
   * Handles data retrieval, manipulation, and storage.
   * Uses libraries like Entity Framework to connect and communicate with the SQL Server database.

**Benefits of using a 3-layer architecture:**

* Improved modularity and maintainability
* Clear separation of concerns
* Easier to test and deploy individual layers

**Why Razor?**

* Razor allows creating interactive web UIs with C#, familiar to many developers.
* Reduces the need for separate frontend and backend development tools.
* Enables real-time updates without full page reloads, enhancing user experience.

**Deployment (Optional - Azure):**

If time permits, the system can be deployed on Microsoft Azure, a cloud platform offering various services like:

* Azure App Service: For hosting the Razor application.
* Azure SQL Database: Managed SQL Server service for cloud-based database needs.

# **Implementation**

## **Deployment Considerations**

**Effective Deployment:**

* **Pre-deployment:**
  + System testing in a staging environment to identify and fix bugs before launch.
  + Data migration plan for transferring pet medical records to the new system (if applicable).
  + Security configuration to ensure data privacy and system integrity.
* **Deployment:**
  + Choosing a deployment window with minimal disruption to potential users (e.g., evenings or weekends).
  + Monitoring system performance after deployment to identify and address any issues.

**User Access:**

* The system is designed for global accessibility with an internet connection.
* Users can be distributed across various time zones, requiring 24/7 system availability.
* The system should be designed to handle peak usage times, which might vary depending on user location and pet care routines (e.g., appointment bookings could be higher during evenings or weekends).

**Infrastructure Changes:**

* **Capacity:**
  + Server capacity may need to be scaled based on user base growth. Cloud platforms like Azure offer auto-scaling features to handle fluctuating demand.
* **Network Access:**
  + Reliable internet access is required for users and for communication between the application and database.
* **Data Storage:**
  + Storage requirements will increase with user base and pet medical records. Upgrading storage capacity or utilizing cloud-based storage solutions might be necessary.
* **Data Migration:**
  + If migrating existing pet medical records from another system, a secure and well-tested data migration plan needs to be established.

**Training and Business Process Changes:**

* **Training Materials:**
  + User guides and training materials will be needed for pet owners and veterinary clinic staff to navigate the system effectively.
  + Training should cover appointment booking, pet medical record management, and any new functionalities introduced by the system.
* **Business Process Changes:**
  + The system might necessitate adjustments to existing workflows in veterinary clinics. Training should address these changes and ensure smooth adoption of the new system.

**Additional Notes:**

* Data security and privacy compliance are crucial considerations throughout deployment and ongoing operation.
* A communication plan should be established to keep users informed about the deployment process, potential downtime, and any new features or functionalities.

## **Screenshots and explanations**

*<Screen flow | Dialog Map>*

*<The screenshots and explanations>*

* + 1. Book Appointment Online: Login -> Manage pet -> Add pet -> Book appointment + Choose payment method by cash, time, vet, pet -> Create Transaction -> Login Staff -> Staff dashboard -> Manage appointment -> Choose appointment that did not paid -> Change status Paid
    2. Book Appointment Offline: Login -> Staff dashboard -> Manage customer -> Check if user is exist -> Manage Booking -> Create Appointment -> Choose customer payment method, time, vet, pet, status -> Confirm
    3. Manage Medical Record: Login -> Vet dashboard -> TimeTable -> Choose appointment -> Choose pet -> Create medical record -> Done appointment
    4. Manage Hospitalisation: Login -> Staff dashboard -> Hospitalization -> Medical record list -> create hospitalization for medical record has hospitalisation date -> Login Vet -> Vet dashboard -> medical record list > hospitalization list > update hospitalization

# **References**

Web Template: <https://themewagon.com/themes/free-bootstrap-4-html5-pet-clinic-website-template-petcare/>

Github:  
[quangtrandinhminh/PetHealthCareSystemRazorPages (github.com)](https://github.com/quangtrandinhminh/PetHealthCareSystemRazorPages)