Pet Clinic Management System

TEAM 6

# Team members:

* Elias Barrellet 2072683
* Claudiu Mihael Terenche 6268599
* Qingjun Bao 2398022

# Project Description:

Pet Clinic Management System is a WPF application that offers a modern, efficient solution for managing a pet clinics and veterinary hospitals. The system provides a modern interface for managing various clinic operations. The application aims to simplify tasks such as appointment scheduling, patient (pet) record management, inventory tracking, and financial transactions, thereby improving the efficiency of clinic staff and enhancing the overall customer experience.

# Technologies Used:

* WPF (Windows Presentation Foundation) for the user interface
* C# as the programming language
* Azure SQL Database for data storage
* Entity Framework for database operations
* Guna framework for components/icons

# Special Features:

* **Appointment Scheduling:**

Enables staff to book, view, and manage appointments through an intuitive calendar interface. Includes functionality for appointment reminders and scheduling recurring visits. Appointments emails integration.

* **Patients (For pets) Record Management:**

Allows for the comprehensive management of pet records, including medical history, vaccination records, visit notes, and owner information.

* **Inventory Management:**

Enables staff to track and manage the stock of pets, food, toys, and other accessories. Features include adding new items, updating stock levels, and monitoring inventory levels to prevent shortages.

* **Sales Transaction Management / Billing and Invoicing :**

Facilitates the processing of sales transactions, including the sale of pets and pet products. It supports various payment methods and generates sales receipts.

* **Customer Management:**Admin will manage customer information, including contact information, purchase history, etc.
* **Reporting:**

Generates detailed and printable reports on sales, inventory levels, and customer interactions.

* **User Access Control:**

Implements role-based access control (RBAC) to ensure that only authorized personnel can access sensitive information and perform certain operations within the system.

# Additional Libraries:

* Bcrypt for password hashing
* MVVM Frameworks
* Entity Framework / ORM framework for simplifies data access: Dapper
* UI/UX Design Libraries: Material Design In XAML
* Charting and Reporting: LiveCharts, OxyPlot
* Logging and Diagnostics: Serilog / NLog
* Testing Frameworks: NUnit / xUnit
* Print & PDF: PrintDialog / PdfSharp / MigraDoc / Crystal Reports / Microsoft Reporting (RDLC)
* Payment: Stripe .NET Library / PayPal SDK
* Email notifications: SendGrid / MailKit / Firebase Cloud Messaging (FCM)

# Challenging Items:

* **Complex UI/UX Design:**

Challenge: Creating an intuitive, responsive, and attractive user interface that meets the expectations of modern users can be challenging.

Strategy: Utilize Material Design In XAML or MahApps.Metro to streamline the design process.

* **Data Management and Integration:**

Challenge: Efficiently managing and integrating various data sources, such as inventory, sales, and customer information, while ensuring data integrity and performance.

Strategy: Use robust ORM tools like Entity Framework Core for seamless data operations.

* **Security and Privacy:**

Challenge: Protecting sensitive customer and business data against unauthorized access, breaches, and other security threats.

Strategy: Implement strong authentication and authorization mechanisms.

* **Appointment Scheduling**

Challenge: Handling complex scheduling requirements, such as managing overlapping appointments, varying appointment lengths, and resource (e.g., specific veterinarians or equipment) availability.

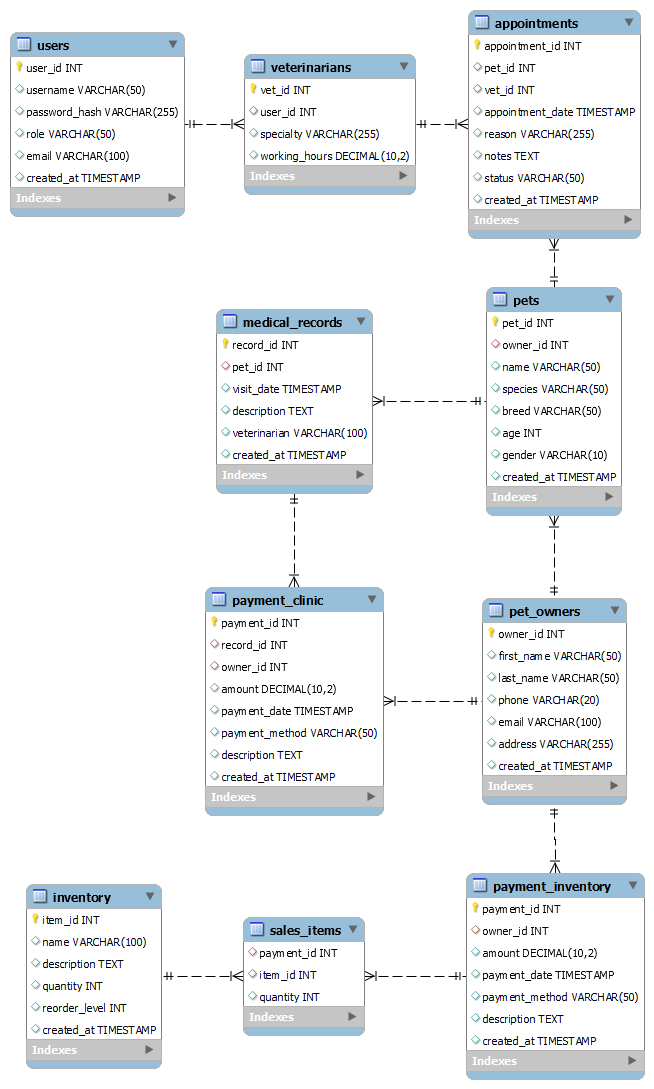
Solution: Design a flexible scheduling system that allows for easy adjustments and viewing of appointments in different formats (e.g., daily, weekly, monthly views). Consider using existing scheduling libraries that can be adapted to your needs.

* **Testing and Quality Assurance:**

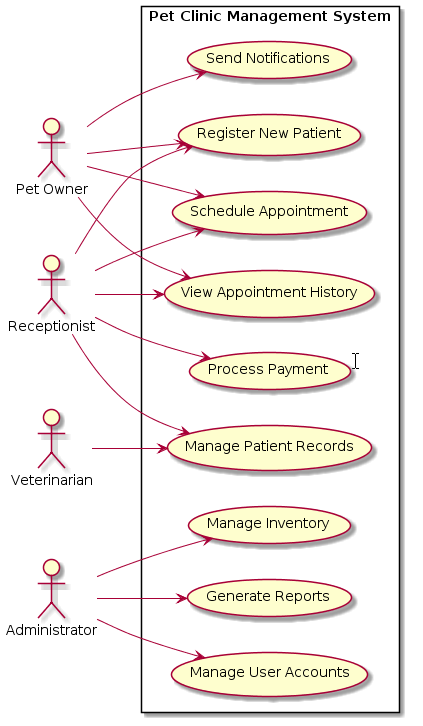
Challenge: Ensuring the application is bug-free and performs as expected under various scenarios.

Strategy: Implement a comprehensive testing strategy, including unit tests, integration tests, and UI tests.

# Database Design:



# Use case diagram:



# Mockups:

