Software Requirements Specification

for

VetCare – Online Vet Clinic Management System

Version 1.0 approved

Prepared by Group-P08-03

RMIT University

17th August 2024

# Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 2

1.5 References 2

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 3

2.3 User Classes and Characteristics 5

2.4 Operating Environment 8

2.5 Design and Implementation Constraints 8

2.6 User Documentation 9

2.7 Assumptions and Dependencies 9

3. External Interface Requirements 9

3.1 User Interfaces 9

3.2 Hardware Interfaces 9

3.3 Software Interfaces 9

3.4 Communications Interfaces 9

4. Nonfunctional Requirements 20

4.1 Performance Requirements 20

4.2 Safety Requirements 20

4.3 Security Requirements 20

4.4 Software Quality Attributes 21

4.5 Business Rules 22

5. Other Requirements 24

6. System Architecture 26

6.1 Architecture Overview 26

6.2 Architectural Decisions 27

7. User Interface Design 29

Appendix A: Glossary 39

Appendix B: To Be Determined List 46

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of all core and additional features in VetCare. This document will cover each of the application’s functional and non-functional requirements and offer, as well as a preliminary glimpse of the user interface design. The document will also cover the system architecture, hardware, software, and other technical requirements.

## Document Conventions

This document used terminology which readers may be unfamiliar with. See Appendix A (Glossary) for these terms and their definitions.

This document includes a list of the TBD (to be determined) references for tracking purposes. See Appendix B (To Be Determined List) for complete list of TBD references.

## Intended Audience and Reading Suggestions

This document is intended for all individuals participating/supervising in building VetCare. The intended readers include, but are not limited to product owners, scrum masters, and software engineers.

Readers who wish to explore the features of VetCare in more detail should read Part 2.2 (Product Functions), and Part 3 (External Interface Requirements) offers more technical details, including information on the user interface and the hardware and software platforms on which the application will run.

Readers who are testers or quality assurance professionals or interested in the project's non-technical aspects should read Part 4, which covers performance, safety, security, software quality attributes, and business rules. Readers who have not found the information they are looking for should check Part 5 (Other Requirements), which includes any additional information which does not fit logically into the other sections.

## Product Scope

Vetcare is composed of 3 main components: a client-side, an admin/owner side, and a veterinarian side.

1. Client-side includes appointment scheduling, prescription refills, educational resources, and updating pet medical records, all aimed at improving convenience and efficiency.
2. Admin-Side manages user, pet, and vet information, oversees clinic operations like appointments and prescription stock, and ensures accurate and accessible medical records.
3. Veterinarian-Side allows vets to manage their schedules, view and update pet medical records, and prescribe medications. It ensures that veterinarians have the tools and information they need to provide the best possible care for their patients.

## References

Australian Government (2024) Privacy Act 1988 (Cth), Compilation No. 79, 22 May 2024, including amendments up to Act No. 24, 2024, Federal Register of Legislation website, accessed 23 August 2024. <https://www.legislation.gov.au/C2004A03712/latest/text>

Maoz (2022) Veterinarian Clinic Dashboard, Behance website, accessed 20 August 2024. <https://www.behance.net/gallery/153410139/Veterinarian-Clinic-Dashboard>

Perforce (2023) How to Write an SRS Document (Software Requirements Specification Document), Perforce website, accessed 17 August 2024. <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#what-is-a-software-requirements-specification-document-01>

Victorian Legislation (1997) Veterinary Practice Act 1997, Victorian Legislation website, accessed 21 August 2024. <https://www.legislation.vic.gov.au/in-force/acts/veterinary-practice-act-1997/044>

# Overall Description

## Product Perspective

VetCare is a new, self-contained software product designed to streamline veterinary practice management and enhance the quality of care for pets. It is a comprehensive system that integrates various functionalities required for efficient operation of a veterinary clinic, including appointment scheduling, patient management, medical record-keeping, and prescription delivery.

VetCare is not a replacement for any existing systems but rather a new solution developed to address the evolving needs of veterinary practices through an innovative and all-inclusive system. Previous systems used by veterinary clinics usually used manual processes that increase the possibility of inefficiencies and errors. Therefore, VetCare seeks to combine these features into a single integrated platform.

**Relation to Larger Systems:**

* Integration Points: VetCare is designed to interface with various external systems to enhance its functionality. Key integration points include:
  + Payment Processing Systems: To handle billing and payments efficiently.
  + Electronic Health Record (EHR) Systems: For comprehensive patient record management and integration.
  + Appointment Management Systems: To allow clients to book appointments online.
* **Subsystems and Interfaces:**
  + User Management System: Handles registration, login, and profile management for both pet owners and veterinarians.
  + Medical Record System: Stores and manages pet medical data, including immunization and other medical histories.
  + Prescription Management System: Facilitates prescription details, refill requests, and delivery arrangements.
  + Educational Resource System: Provides access to pet care and veterinary practice resources.

## Product Functions

The following list offers a brief outline and description of the core features and additional features of VetCare. Core features are essential to the application’s operation, whereas additional features simply add new functionalities. The latter features will only be implemented as time permits.

### Core Features

1. Pet Owner Registration and Welcome
   1. Allows pet owner to register new account
   2. Allows pet owner to register pet(s) account
   3. Welcome page is directed after successful completion of registration
2. Veterinarian Registration and Welcome
   1. Allows veterinarian to register new account
   2. Welcome page is directed after successful completion of registration
3. User Login and Welcome
   1. Allows pet owner and veterinarian to enter their credential details to access VetCare
   2. Welcome page is directed after successful login
4. Appointment Scheduling
   1. Enables pet owner to schedule new appointment with veterinarian of their choice at available date and time slot
   2. Enables pet owner to reschedule existing appointment at other date and time slot
   3. Enables pet owner to cancel existing appointment
   4. Allows veterinarian to view his/her today’s scheduled appointment
5. Pet Medical Record
   1. Store and manage the pet's medical history, immunization records, and treatment plans
   2. Download and share the pet's medical history in specified file format or link
6. Pet Prescription Management
   1. Allow pet owner to access information about the pet’s medication(s) and dosage instruction
   2. Enable pet owner to request and purchase for prescription refills for the pet’s medication(s), and have them delivered to the shipping address
   3. Allow veterinarian to add new prescription for the pet
7. Education Resources
   1. Allow pet owners to access library of articles, videos, and guides on pet care and wellness
   2. Allow veterinarian to access the latest trends and best practices in veterinary medicine

### Additional Features

1. Pet Owner Appointment Reminder
   1. Reminds pet owner of upcoming appointment 24 hours before the time of appointment
   2. Reminder is received via application notification on pet owner device
2. Pet Owner / Veterinarian Password Reset
   1. Allow pet owner and veterinarian to set a new password for their account
3. Administrator Dashboard
   1. Allow clinic administrator to access and/or manage:
      1. Scheduled appointment
      2. Clinic staff
      3. Client
      4. Patient
      5. Prescription inventory stock
      6. Prescription order

## User Classes and Characteristics

VetCare is designed to provide a seamless, user-friendly platform that satisfies the needs of pet owners and veterinarians. It aims to simplify pet care management by offering a range of essential features, from appointment scheduling to medical record management. The primary users of VetCare can be classified based on their roles and how they will interact with the application, which are defined below:

1. **Pet owners**

Pet owners are the primary users of VetCare and utilises the application to manage their pets' wellbeing. This group is diverse, ranging from tech-savvy individuals to those who may have limited experience with web applications. Regardless of their technical proficiency, the platform is designed to be intuitive and easy to navigate, so users can efficiently and conveniently manage their pets' healthcare needs.

* **Key functions:**
  + Register and manage accounts for their pets.
  + Schedule, reschedule, and cancel veterinary appointments.
  + Access and manage their pets’ medical records and prescriptions.
  + Request and purchase prescription refills, with delivery to a specified address.
  + Integrated secure payment system for purchasing prescription refills, supporting payment methods using credit/debit cards.
  + End-to-end encryption to protect financial transactions and user data.
  + Receive reminders for upcoming appointments.
  + Access educational resources on pet care and wellness, and best veterinary practices.
* **Requirements:**
  + Simple and intuitive registration process for both pet owners and their pets.
  + Easy-to-use appointment scheduling system with real-time availability.
  + Secure storage and retrieval of pet medical records.
  + Notifications and reminders for upcoming appointments.
  + Access to educational content within the web application.
* **User Characteristics:**
  + May have varying levels of technical expertise.
  + Primarily seeking convenience, security, and ease in managing their pet’s health.
  + Likely to value a secure and straightforward payment process for online transactions.

1. **Veterinarians**

Veterinarians are professional users who interact with the application primarily to manage their schedules, maintain and view patients’ medical records, and stay updated with the latest veterinary practices. The platform is designed to accommodate their need for efficiency and reliability to allow them to focus more on providing care to their patients.

* **Key functions:**
  + Register an account and access the web application’s dashboard.
  + Manage appointments schedules and availability.
  + Record and update pet medical records.
  + Provide detailed medication and dosage instructions for pets.
  + Access the latest trend and best practices in veterinary medicine.
* **Requirements:**
  + Professional registration process with verification systems.
  + User-friendly interface for managing appointments and medical records.
  + Secure and compliant storage of sensitive medical data.
  + Access to a library of professional development resources.
* **User Characteristics:**
  + Needs reliability, security, and compliance in handling patient data.
  + Values professional development and access to up-to-date information.

1. **Clinic Administrator**

This group may include clinic managers or administrative staff who assist in managing appointments, patient records, and overall clinic operations. They would require access to a broader range of features within the application, including financial management.

* **Key functions:**
  + Oversee the scheduling and coordination of multiple veterinarians.
  + Manage pet owner accounts and records.
  + Manage veterinarian accounts and records.
  + Manage inventory and supplies of medications available.
  + Handle billing and payments for services, such as prescription refills.
* **Requirements:**
  + Administrative dashboard with an overview of all appointments, records, and transactions.
  + Tools for managing and organising pet owners’ and veterinarians’ accounts.
  + Secure payment processing systems with access controls to protect financial data.
* **User Characteristics:**
  + Likely to have a background in office administration or clinic management.
  + Knowledgeable about healthcare regulations, including data protection laws and veterinary-specific requirements.
  + Maintains positive relationships with pet owners.

## Operating Environment

The VetCare application is a web-based platform designed to operate in a robust and scalable environment. The key components of the operating environment are as follows:

* **Hardware Platform**: Depending on deployment preferences, the application will run on standard web servers, either on-premises or in the cloud. The hardware should support Java-based applications and containerized services.
* **Operating System**: The application is platform-agnostic and can operate on any server running a compatible Java Runtime Environment (JRE) version 17 or later. This includes popular operating systems like Linux, Windows Server, and macOS.
* **Database**: The application will utilize MySQL as its primary relational database. The database server should be configured to handle requests efficiently and support the expected load.
* **Web Framework**: The application is built using the Spring Boot framework, which simplifies the creation of production-ready applications and ensures compatibility with modern web standards.
* **Build and Deployment**:
  + **Build Tool**: Maven is used as the build tool to manage project dependencies and automate the build process.
  + **CI/CD**: GitHub Actions will be employed for Continuous Integration and Continuous Deployment, ensuring that the application is tested and deployed efficiently.
* **Unit Testing**: The application will use JUnit5 for unit testing to ensure code quality and reliability.
* **Containers**: Docker will be used to containerize the application, which will enable consistent deployment across different environments and simplify the development and testing process.

## Design and Implementation Constraints

The main limitation in designing the VetCare application is its reliance on the internet, which needs to accommodate both users on desktops and mobile devices. Meeting the need for compatibility on two platforms presents substantial obstacles in guaranteeing that the user interface remains efficient and user-friendly on various screen sizes and resolutions. Creating a user interface that is both responsive and maintains functionality and ease of use on smaller mobile screens will present a significant challenge.

The app needs to be created in a way that allows it to easily expand to handle more users and data as the service becomes larger. This involves making sure that the basic database and server infrastructure can manage increased loads without a decrease in performance.

To make updates and maintenance of the application easier in the future, the codebase needs to follow software engineering best practices, such as modular design, consistent coding standards, and comprehensive documentation.

## User Documentation

The user documentation for the VetCare will include the following components to ensure that users can effectively utilize all features and functionalities:

**FAQs (Frequently Asked Questions)**:

* **Description**: A list of common questions and issues that users might encounter, with clear and concise answers.
* **Format**: Web-based format, accessible via the VetCare website.

## Assumptions and Dependencies

### Time Dependencies

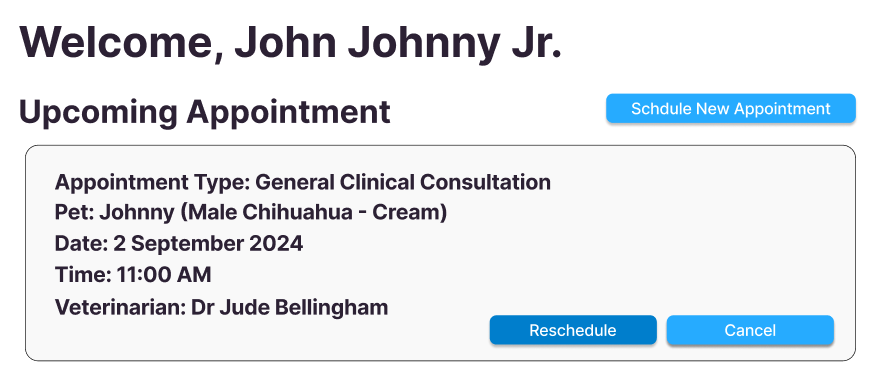
As previously mentioned, the features of VetCare are categorized into two groups: core features and additional features. Core features are essential to the fundamental operation of the VetCare web application. These functionalities must be in place for the application to perform as intended. On the other hand, additional features are not vital to the application's core operations. They serve as enhancements that can be introduced once the primary features are fully developed.

The decision to include these optional features will depend on the time allocated to designing and implementing the core functionalities. The final determination on whether to incorporate these features will be made in the later stages of the design phase.

# External Interface Requirements

## User Interfaces

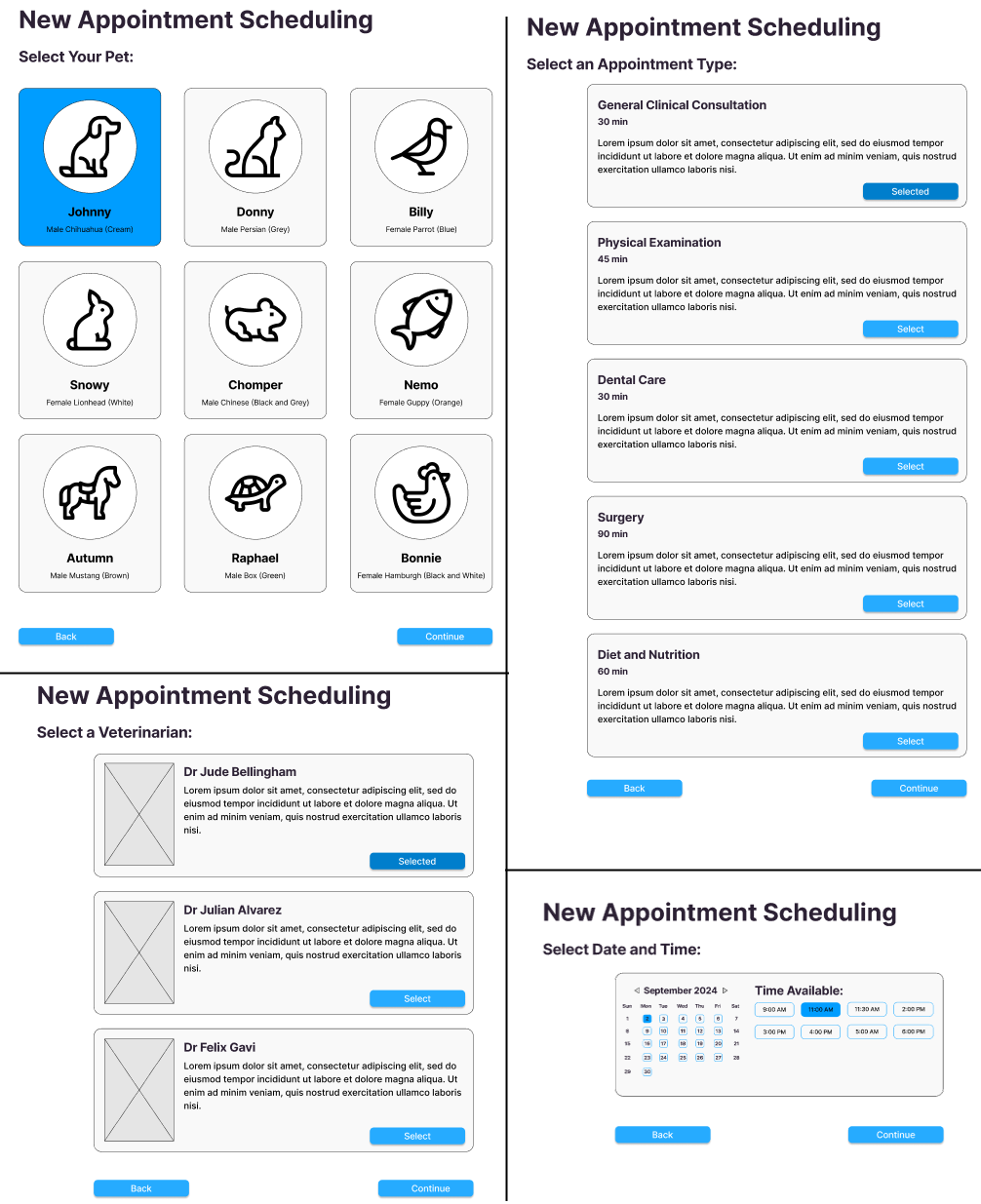
### Dashboard Page



On the dashboard page, users will be able to book a new appointment as well as change that date and time for an existing appointment. This page will include:

* Upcoming Appointment: This section of the page shows users their upcoming appointments, shown in order from closest to furthest in time.
* Schedule New Appointment: The users will be able to click the “Schedule New Appointment” Button which will be a direct link to appointment scheduling page.
* Reschedule Appointment: This button is shown on every upcoming appointment. Clicking it will take users to a page that lets them change the time and date of a given appointment.
* Cancel Appointment: This button is shown on every upcoming appointment. Clicking it will let users cancel a given appointment.

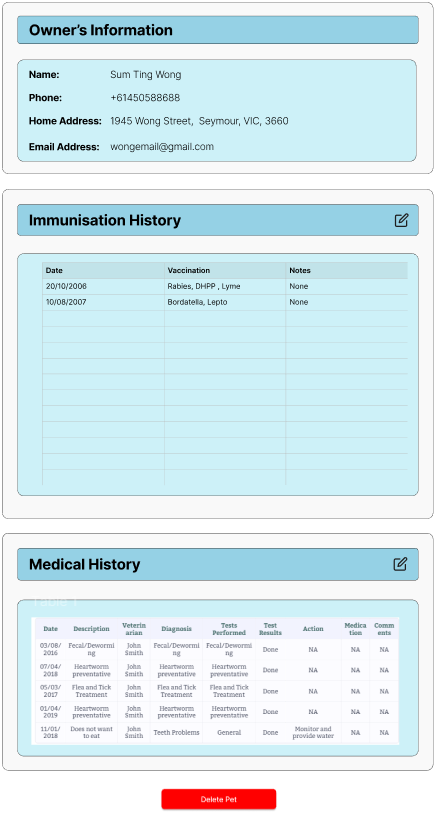
### Appointment Scheduling Page



On the appointment scheduling page, users can schedule a new appointment for their pets. This page will include the following:

* Pet selection: Users will select one of their pets for the upcoming appointment.
* Appointment type: Users will select the type of appointment they want from a list of options. These options include a general consultation, a physical examination, dental care, surgery, or an appointment for diet and nutrition.
* Vet selection: Users will choose the vet they want for their pet from a list of options. The available vets are ones who can carry out the appointment – i.e., if the user wants an appointment for their cat, the available vets are ones who are qualified to work with cats.
* Date and time: Once the user has chosen a vet, they will then select a time and date said vet is available for their appointment.
* Final confirmation: Users will be able to review the details of their appointment and can either confirm the appointment or go back and make any changes that are necessary.

### Pet Profile Page

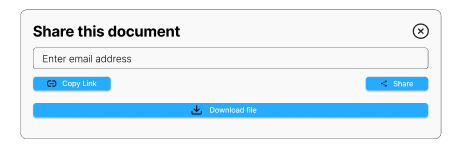


On this page, users can see detailed information about a certain pet, including the pet’s details, any ongoing treatment plans and medical history. The pet’s owner can edit any information they want on this page. It includes the following:

* Pet information: This section shows details about the pet, including name, gender, species, age, etc.
* Pet medication: This section shows any ongoing treatment plans for the given pet, including the name and photo of the medication for the treatment, as well as frequency and dosage.
* Health concerns: This section shows any current health concerns for the pet, including allergies, existing health conditions and vets who have examined the pet.
* Immunisation history: This section shows details about vaccines the pet has taken, including the type, date of vaccination, vet who gave the vaccination and any additional notes.
* Medical history: This section shows details for all appointments the pet has gone through. These include dates of appointments, brief descriptions of appointments, the vets who conducted the appointments, diagnoses and tests from the appointments, recommended actions to take after the appointments and additional notes.

Although the pet owner’s information is displayed on this page, it can only be edited through their profile page.

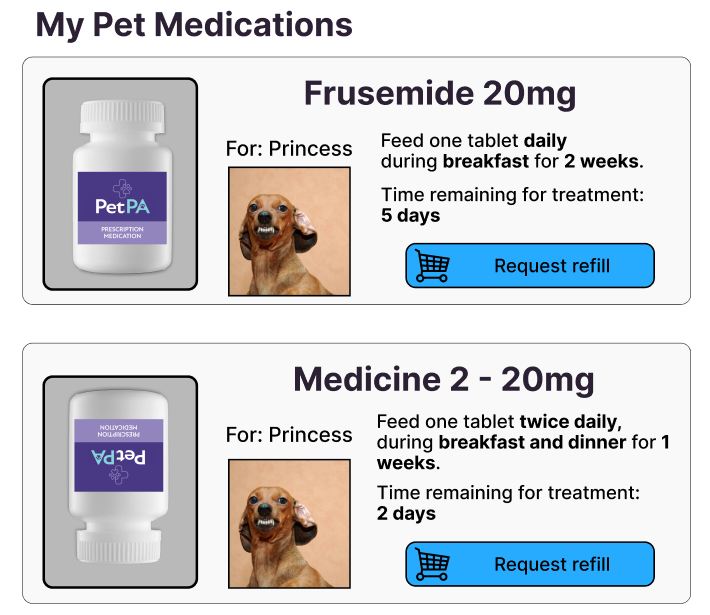
### Share Pet Profile Page



On this page, users can share a pet’s profile. The profile shared will contain the details shown in said pet’s medical history page. We have three methods planned for sharing:

* A direct link to the pet medical history page.
* Sending the pet’s profile through e-mail.
* Downloading pet’s profile in the form of a document.

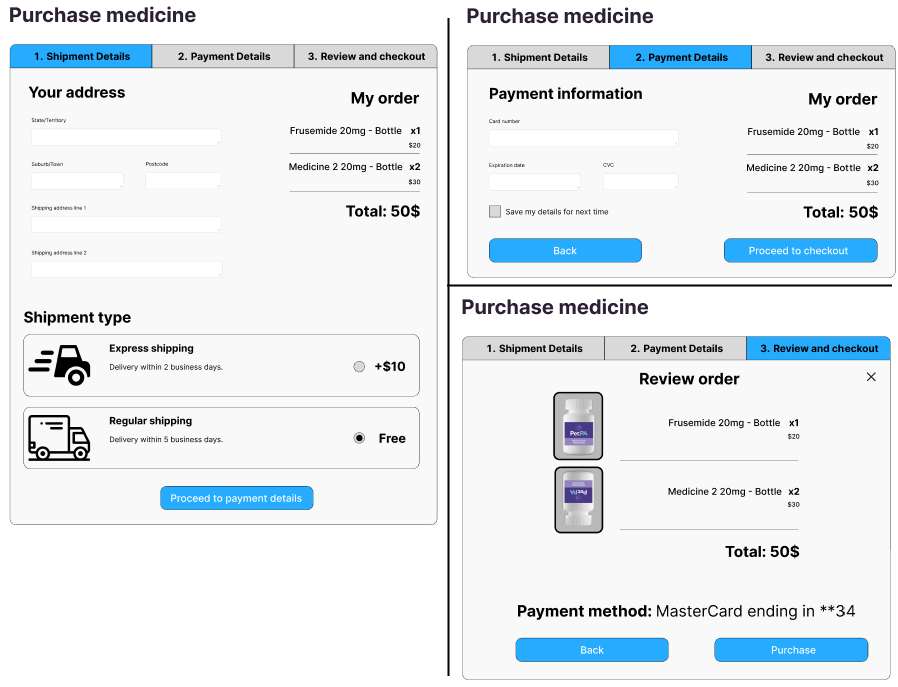
### Medication Management Page



On the Medication Management page, users can see what medications they should give their pets as well as the treatment plan (frequency and dosage). This page includes:

* Medication list: Users can see each medication that is part of an active treatment plan. A photo and the name of each medicine is included for the user’s convenience.
* Treatment plan: Users are shown the treatment plan for each medication. This includes the pet that is receiving treatment, the frequency at which they should take the medicine and how much time is remaining for treatment.
* Refill request: Using this button, users are taken to a page where they can order a refill of a given medication should they need one.

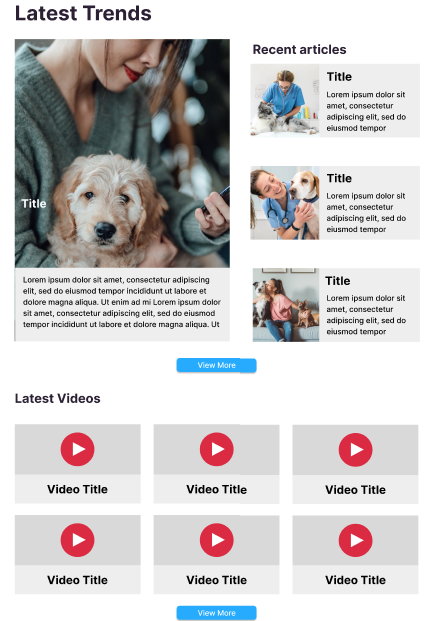
### Purchase Medicine Page



Purchase Medicine Page is where users can buy medication for their pets. The page follows a typical e-commerce flow for user ease of use. The page includes:

* Medication List: Users can see a summary of the medication they’ve added to their cart. Each item in the list includes the medications name, quantity and price.
* Total Cost Summary: Below the list of medication, there will be a summary of the total price. The total is calculated dynamically based on the items in the cart.
* Shipping Details: Users will be prompted to fill in their shipping details, including state/territory, suburb/town, postcode, shipping address line. The users are provided options for regular shipping (free) or express shipping (additional cost).
* Payment Details: After entering shipping information, users will then move to the payment section. Where they will insert their credit/debit card information. This section will ensure that the payment is securely processed before the order in completed.
* Review and Checkout: Before finalising the purchase, users are given a final review of their order. They can then confirm the purchase by clicking on the Purchase button.

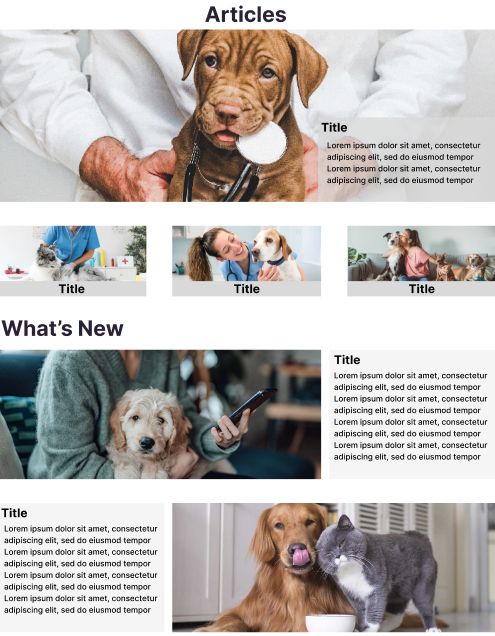
### Latest Trends Page



The Latest Trends page is created to keep users informed about the most recent developments in pet care. The page includes:

* Recent and Trending Articles: Vets will be able to view articles and guides on the latest trends and practices in veterinary medicine.
* View More Button: Vets will be able to press these button that is located under each section, which will lead them to their respective pages.

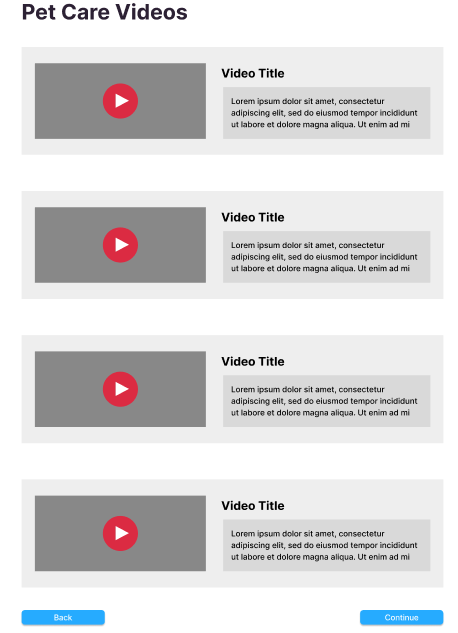
### Articles Page



The article page aims to provide educational and informative content related to pet care. The page includes:

* Hero Article: This will display the most popular article at the given time at the top of the page. The article with have its title and a short description.
* Article List: Under the hero article, users will be able to view articles related to the one displayed in the hero section as well as the newest articles from the world of pet care.
* Article Details: Clicking on the articles will bring up the full text, along with accompanying images or other media. Articles cover a wide range of topics, form general pet care advice to more specific medical or wellness tips.

### Videos Page



This Videos page serves as a library for users to watch various education or instructional videos related to pet care. This page includes:

* Video thumbnails: The videos are displayed as clickable thumbnails with its corresponding title and description on the right. This format makes it easy for users to browse and find a video that interests them
* Next and Back buttons: These buttons will be located at the bottom of the page to help guide users to the next or previous page of videos.

## Hardware Interfaces

Currently, we intend our software to be used as a web application for desktop browsers. As such, all interactions between software and hardware should be doable with a mouse and keyboard. Interactions will include typing out information in forms and clicking on buttons.

## Software Interfaces

The VetCare system interfaces with various software components. These components will be essential for the platform to function and handle data throughout the application.

### Database

MySQL Database: The VetCare system will utilize an SQL database for data storage. The database will store user data, including pet records, prescription details, and appointment history.

All data will be handled in a relational format, ensuring proper normalization and integrity. Communication with the database will occur via Spring Boot framework, which will manage queries, updates and data transactions.

### Operating Systems

Supported operating systems: VetCare is primarily deployed and developed on Windows and macOS. The software will be cross-platform compatible for users accessing it through web browsers.

### Tools and Libraries

**Spring Boot:**

VetCare is developed using the Spring Boot framework, a Java-based framework that offers comprehensive infrastructure support for building web applications.

**Java:**

The main programming language for the VetCare platform will be Java, used with the Spring Boot framework. Java will handle the back-end logic, including API request, business logic, and database interactions.

## Communications Interfaces

Our product will use web browsing and e-mail for communication. The product will be accessed through the browser, and e-mail will be used for certain features (e.g., sharing medical history). We will collect data from users via electronic forms, where users will provide us with the necessary details for us to deliver the best service possible. This data will be stored in our database.

The communication protocols that we use will include HTTPS and SFTP. HTTPS will be used for displaying the application on users’ browsers, and SFTP will be necessary to allow users to download their pets’ medical history.

We must ensure that pet and user data is stored securely in our database as it contains sensitive information, including personal details of the user and their pets, medical history about their pets and any upcoming appointments. We intend to address these issues by implementing secure communication protocols (HTTPS and SFTP) over their less secure counterparts (HTTP and FTP).

Additionally, we must ensure that all data is synchronized correctly between the users and the database, meaning that whenever users add or edit data (e.g. scheduling or rescheduling appointments), the records get updated in the database accordingly.

# Nonfunctional Requirements

## Performance Requirements

VetCare is expected to have at most a 5-second response for computationally heavy tasks which require queries and updates of enormous amounts of data from remote databases. Redirection of screens will require very little computation and thus will occur very quickly. The application’s response should not have unexpected interruptions if the database maintains a steady connection with the application. The algorithms used in VetCare application will be highly efficient, taking only a fraction of a second to compute.

## Safety Requirements

VetCare will not affect any other web application and browser running on the user’s device or the data stored outside its database. It should not and cannot cause any damage to the device’s operating system or its hardware components. The only potential safety concern associated with this application applies to virtually all web applications: VetCare should not be used in situations where the user’s attention must be focused elsewhere, e.g. driving.

## Security Requirements

This application implements and enforces Role-Based Access Control (RBAC) to restrict access to only relevant information based on user roles. For user identity authentication, verification checks on user credential information are implemented to ensure that only authorized users can access pet medical information. The verification checks will involve comparing the hashed users’ passwords when they create an account and when they login. Furthermore, the application should implement strong password policies, including minimum length requirements and rules such as the inclusion of upper- and lower-case letters, numbers, and special characters) to further mitigate the risk of compromised credentials.

This could only pose a security threat if unauthorized individuals have access to user credentials. Therefore, the application must ensure compliance with the Privacy Act 1988 for handling the personal data of users in Australia. If applicable, ensure compliance with veterinary practice standards issued under the Veterinary Practice Act 1997. For security measures, ensure the application’s dependencies are regularly updated with the latest software updates and patches to protect against new cyber threats. Any external API usage must be checked for security vulnerabilities before its integration into VetCare.

## Software Quality Attributes

VetCare’s user interface is designed with strong emphasis on its usability for all users. The application will be organized in an easy way for the user to navigate to ensure seamless user experiences. The design layout will be consistent throughout the application. There should not be any confusion on interpretation of the user interfaces.

To ensure high reliability, the application should handle up to 500 users without performance degradation at the same time. Recovery from application failure should be within 5 minutes. VetCare software implementation should reflect its core functionality accurately at any time to achieve correctness. Bugs reported by the user or team should be fixed with the patch fixes to be released as soon as possible. To achieve high maintainability and reusability, the codebase should follow the system architecture guidelines, object-oriented programming principles, and standard coding practices adopted in Oracle Code Conventions for the Java Programming Language. The application should allow easy reuse of code across different modules. Readers who wish to explore the details of system architecture guidelines should read Part 6 (System Architecture). The codebase should also allow for easy bug fixes and feature updates when necessary.

VetCare should achieve interoperability through supporting import/export of data from/to APIs and database using standardized data formats such as CSV, JSON and XML. This practice ensures consistency in data exchange within the application, and with external systems. To maintain adaptability, VetCare will take account scenarios in which the application loses connectivity to the Internet or the database for whatever reason. The users should be informed of the connectivity issues and the operations will be continued once the connection is established with the Internet or the database. Furthermore, the application should support the addition of new functionality with minimal modification to the existing codebase.

VetCare should handle incorrect inputs from its user and display meaningful error messages to ensure robustness. The application must be able to recover from any crashes without data loss in the database. For testability, the application’s feature should have at least 90% coverage with automated tests. All core features should be tested before final deployment. Overall, the application aims to meet the requirements of users which results in a well-rounded online vet clinic management system solution.

## Business Rules

The following business rules govern the operation of the VetCare application, specifying which individuals or roles can perform specific functions under certain circumstances. These rules help ensure that the application operates within the intended scope and maintains security, efficiency, and compliance with relevant regulations.

1. **User Roles and Permissions:**

* **Pet Owners**:
  + Can register and manage their accounts.
  + Can add, edit, and delete their pets’ medical history.
  + Can schedule, reschedule, and cancel appointments for their pets.
  + Can request prescription refills and choose delivery options.
  + Can view, download, and share their pets' medical records with veterinarians.
  + Can access educational resources, including articles and videos.
* **Veterinarians**:
  + Can register an account and offer services on the VetCare platform.
  + Can view and manage appointments with their patients (pets).
  + Can update pet medical records and prescribe medications.
  + Can access the latest trends and best practices in veterinary medicine through the educational section.
  + Cannot modify or delete pet medical records without owner consent.
* **Clinic Administrators**:
  + Can manage all user accounts (both pet owners and veterinarians).
  + Can oversee and manage appointment schedules for all veterinarians.
  + Can manage prescription stock, including processing refill requests and monitoring inventory levels.
  + Have the authority to access and update all medical records within the clinic.

# Appointment Management:

* **Booking**: Pet owners can book appointments online at any time. Appointments must be confirmed by the veterinarian or clinic administrator.
* **Rescheduling**: Pet owners can reschedule appointments up to 24 hours before the scheduled time without penalty. Rescheduling within 24 hours may be subject to a fee, as determined by the clinic.
* **Cancellation**: Pet owners can cancel appointments up to 24 hours in advance without penalty. Late cancellations may incur a fee.

# Prescription Management:

* **Refill Requests**: Pet owners can request prescription refills through the application. These requests must be reviewed and approved by the veterinarian before being processed.
* **Delivery**: Pet owners can opt for prescription delivery. Once the prescription is approved and processed, the clinic administrator is responsible for ensuring it is dispatched.
* **Stock Control**: Clinic administrators must monitor prescription stock levels and update the system to reflect real-time inventory. The system should alert the administrator when stock levels are low.

# Medical Record Management:

* **Access and Update**: Pet owners have full access to their pets' medical records and can update or delete information as needed. Veterinarians can update records following appointments or treatments but cannot delete records without the owner’s consent.
* **Sharing Records**: Pet owners can share their pets' medical records with veterinarians directly through the application. Shared records are view-only for veterinarians unless explicitly given permission by the pet owner.

## Data Privacy and Security:

## Confidentiality: All user data, including medical records, prescriptions, and personal information, must be encrypted and stored securely. Access to this data is restricted based on user roles.

* **Compliance**: The application must comply with relevant data protection regulations.

# Educational Resources:

* **Content Access**: Educational resources, such as articles and videos, are accessible to all registered users. Pet owners can search for, and view content related to pet care, while veterinarians can access content related to veterinary best practices.
* **Content Management**: Only clinic administrators or designated content managers can add or update educational content in the application.

# 5. Other Requirements

## 5.1, Database Requirements

* **Data Storage:** The system should securely store patient medical records, appointment history, pet owner information (such as billing information) veterinarian personal information (such as degree testamur, personal resume, etc.) in a relational database MySQL.
* **Data Consistency:** The database should be consistent and reliable. Therefore, it should be designed in Third Normal Form (3NF) to eliminate redundancy and ensure that data dependencies are logical and only related to the primary key, consequently, minimizes the chances of anomalies during insertions, updates, and deletions.
* **Referential Integrity:** Enforce referential integrity by using foreign keys to maintain the relationships between tables.
* **Validation Rules:** Enforce validation rules at the database level, other than at the application level, to ensure a strong data integrity. For example, ensure that fields like email addresses, phone numbers, and date of birth conform to the correct formats and are within acceptable ranges.

## 5.2. Legal and Compliance Requirements

* **Data Privacy:** The system must comply with data protection regulations, which for Australia is the Privacy Act 1988. This is because VetCare stores personal information, including clients' and veterinarians’ names, addresses, contact details, payment information, sensitive medical information related to clients’ pets, and important personal documents of veterinarians. Therefore, the system must ensure that all personal data is collected, processed, stored, and shared following the principles outlined in the Privacy Act, including the secure handling of data, limiting access to authorized personnel only, and ensuring data is used only for its intended purposes.
* **User Consent:** The system should obtain explicit consent from users before collecting, storing, or processing their data. This will be done whenever the user registers an account, books an appointment, or submits any personal information through the system. The consent process should include a clear explanation of what data is being collected, how it will be used, and how users can manage their consent preferences or request data deletion. Additionally, users should be informed of their rights under the Privacy Act 1988, including the right to access their data, request corrections, and lodge complaints if they believe their data has been mishandled.

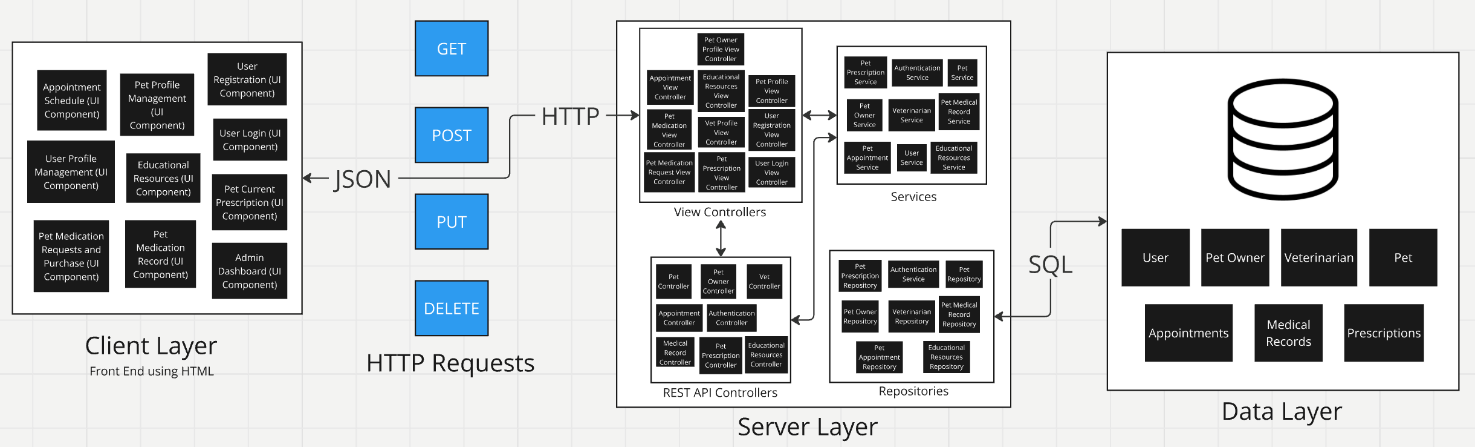
The consent request will be presented to users through a pop-up that requires them to make an explicit choice by clicking either an "I Agree" or "I Disagree" button. The pop-up will include links to the full Privacy Policy and Terms of Service for users who wish to review additional details before making their decision. If a user chooses "I Disagree", the system should explain the consequences of this decision, such as being unable to book an appointment, prescription delivery, and so on, but details on how they can proceed will be given.

* **Access Control:** Implement strict access controls to ensure that only authorized personnel can access sensitive data, such as only the admin can access every pet owner, patient, and veterinarian that is currently registered in the system, whereas veterinarians are only able to see and edit the detailed data of pets that they have had or will have an appointment with. However, veterinarians are still able to see the pet owner’s details, such as full name, phone number, home address, and email address.

# System Architecture

The VetCare web application is designed to provide a robust and intuitive platform for managing veterinary clinic appointments and prescriptions online. The architecture utilises a Client-Server design pattern, which is ideal for this type of application as it enables efficient communication between the client (user interface) and the server (backend). This architecture leverages modern web and database technologies to ensure high availability, reliability, and performance.

## 6.1 Architecture Overview



The VetCare web application follows a Client-Server architecture, structured using the Model-View-Controller (MVC) design pattern, implemented with Spring Boot as the server framework and MySQL as the relational database. It composes of 3 layers:

1. **Client Layer**

**Components:** Web browsers

**Role:** Interacts with the user, sending HTTP requests to the server and receiving HTTP responses. Moreover, the frontend is likely built using HTML, CSS, and JavaScript.

**Communication:** Communicates with the server layer via HTTP/HTTPS and exchange data typically in JSON format.

1. **Server Layer**

**Framework:** Spring Boot

**Role:** The core of the application, which is responsible for processing client requests, executing business logic, and interacting with the database.

**Components:**

* Controllers:
  + View Controllers: Handle web page requests and responses.
  + REST API Controllers: Handle API requests from the client, providing JSON responses.
* Services: Contain business logic and manage interactions between controllers and the database.
* Repositories: Provide an abstraction for data access to allow the service layer to perform CRUD operations on the database using ORM (like Hibernate). ￼

1. **Data Layer**

* **Database:** MySQL
* **Role:** The database of the application, which stores the information of pet owners, veterinarians, pets, and other information needed for the system to operate.

## 6.2 Architectural Decisions

1. **Client-Server Architecture**

The architecture promotes the separation of VetCare functionality by having the client side manage user interface and input validation logic, while the server-side res responds to the client’s requests via a standard protocol. These layers are put together to ensure the application runs as required. Besides that, the architecture allows scalability of each layer to respond to additional requirements, and easier codebase maintenance for software engineers.

1. **Model-View-Controller (MVC) Design Pattern**

The design pattern ensures clear separation of components handling business logic, user interface and data. Hence, it allows software engineers to develop and test their assigned features independently. The pattern promotes reusability of modules in different sections of the codebase, thus improving workflow efficiency. Furthermore, it enables scalability of the application through allowing easier addition and modification of new and existing features without affecting other features. Therefore, software engineers work in a parallel development setting which enables them to develop the application collaboratively.

1. **Spring Boot**

Implementation of the server framework assist software engineers in the development of VetCare through providing an opinionated configuration approach, built-in autoconfiguration feature and the ability to build stand-alone web applications. It also offers support for integrating dependency injection, REST API, and security which are aligned with the application’s requirements. Thus, engineers were provided a tool to quickly build and run production-grade web applications with minimal setup.

1. **MySQL**

MySQL provides a relational database structure to store all information required to run VetCare. The database allows servers to execute complex and flexible queries to display relevant information on the application. Spring Boot allows integration of MySQL database which enables the server to query for data.

1. **REST API**

REST API allows data exchange (in JSON) between the client and the server using HTTP methods. Its utilization of standardized set of data format establishes consistency in server’s and client’s communication which allow easier implementation and maintainability for software engineers. Communication can be secured using standard web security protocols to ensure user data is protected when transmitting across the Internet.

# User Interface Design

The interface for the VetCare is designed to be as intuitive and user-friendly as possible. Its interface is responsive and easily navigable so that users can effectively take care of their tasks, whether it be booking an appointment or viewing medical records and educational resources.

UI Design Principles

The design of the PetCare system follows key user-centric principles to make sure that the application is accessible, functional, and visually appealing across various devices.

1. Consistency

The interface elements on the VetCare platform maintain a consistent layout and style. There is a uniform use of typography, color schemes, and button designs. This quality aids in getting the flow of the application and, therefore, reduces the learning curve for new users.

1. Responsiveness

The VetCare platform is designed in a way that will adapt to different screen sizes. Responsive design ensures the system is usable on these devices in a friendly way and fully functional whether users access it from a desktop, tablet, or mobile device.

1. Simplicity

The user interface is kept purposefully simple with minimal distractions. Every page is designed to serve a single function and offers only the information and actions required. This avoids complexity for users, especially for those who are not very tech-savvy.

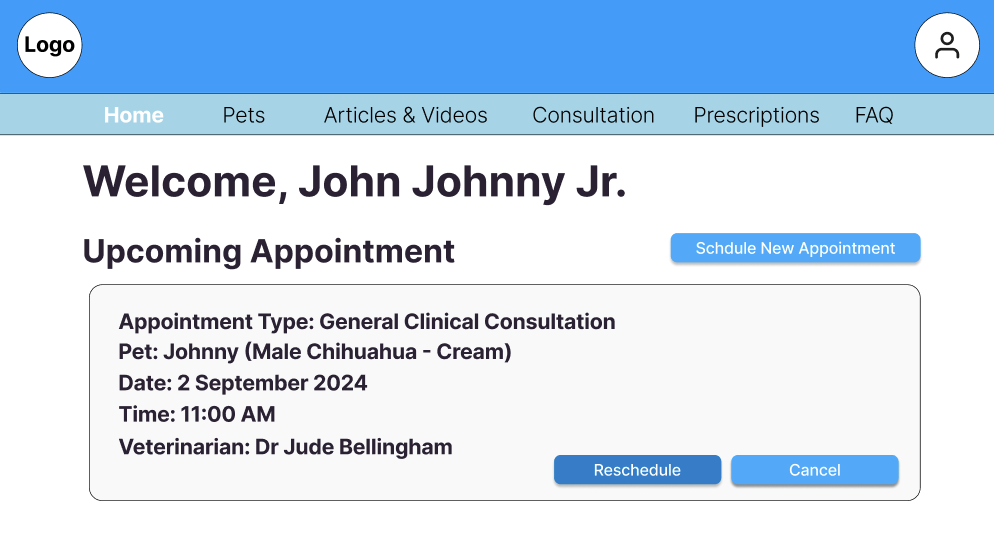
1. Accessibility

This platform is designed so it is easily accessible to everyone. It includes features such as user-friendly fonts, excellent contrast ratios, and easily navigable elements. This will ensure a great user experience.

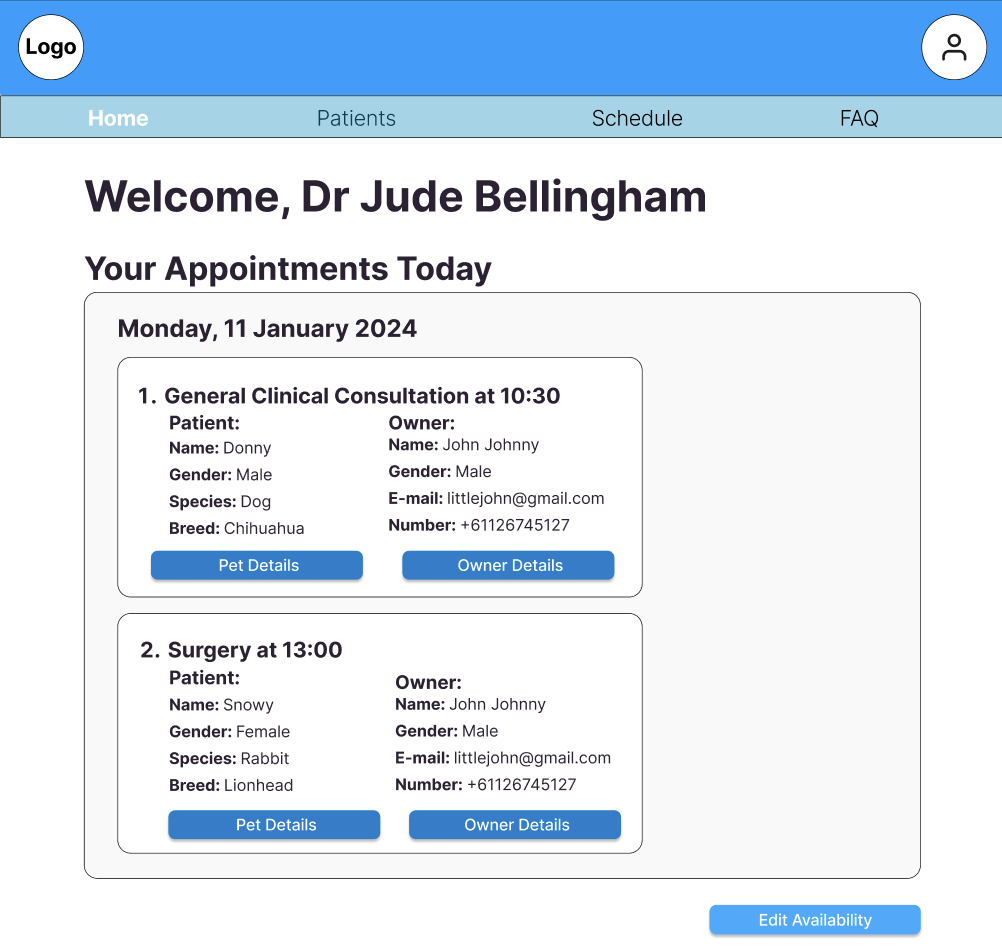
Main User Interface Pages

1. Dashboard Page

* For Pet Owners: Displays upcoming appointments with the ability to book, reschedule, and cancel. Users can access their medical records, prescriptions, and educational resources via navigation bar.

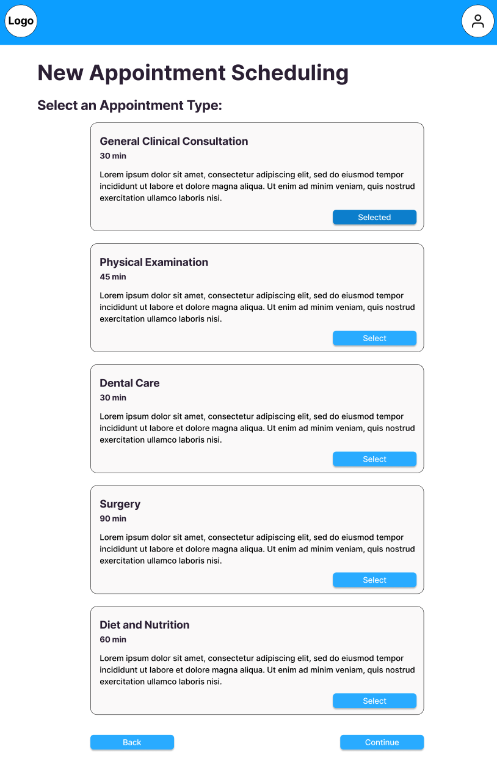
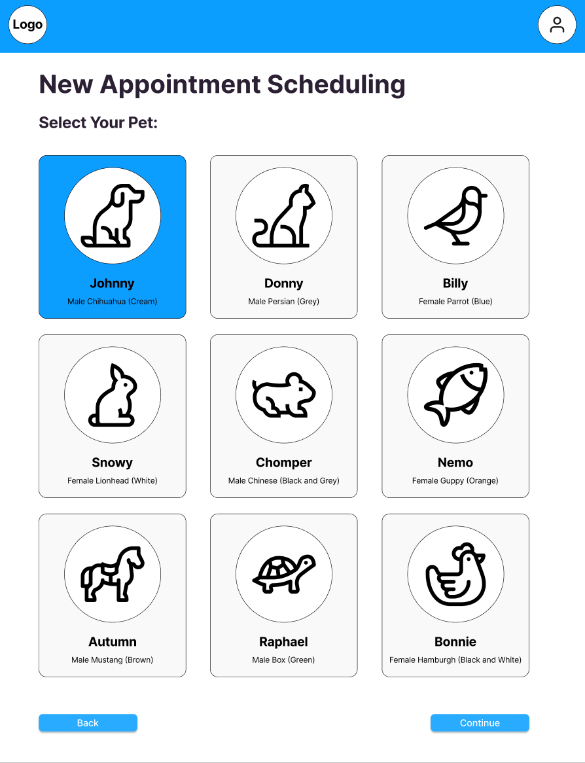


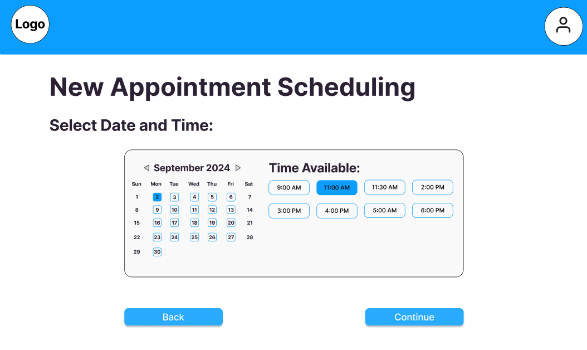
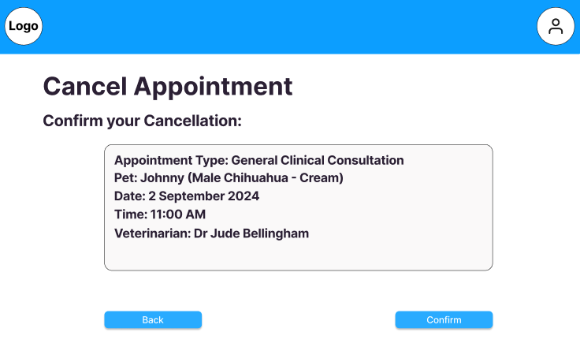
* For Veterinarians: Displays appointments and tasks for the day with links to patient medical records and prescriptions. Detailed patient history can be viewed, and pet information updated.



1. Appointment Scheduling Page

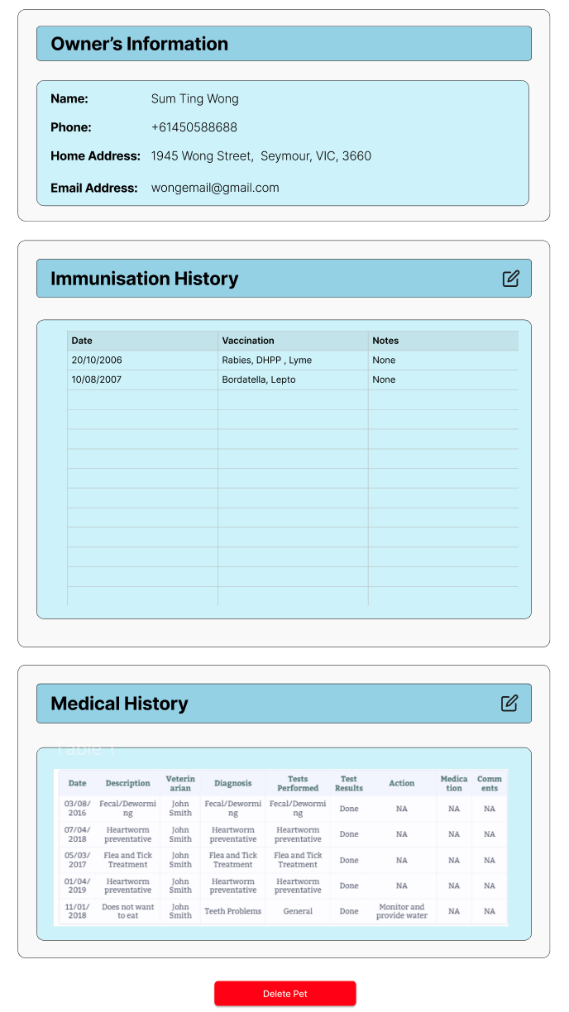
* For Pet Owners: Intuitive and user-friendly design for easy scheduling, rescheduling, or cancelation of appointments. The integrated calendar view shows available time slots according to the selected veterinarian and pet. It also includes confirmation and reminder functions for appointments.

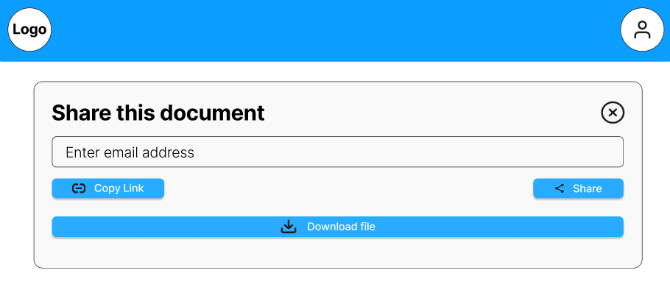




1. Pet Profile Page

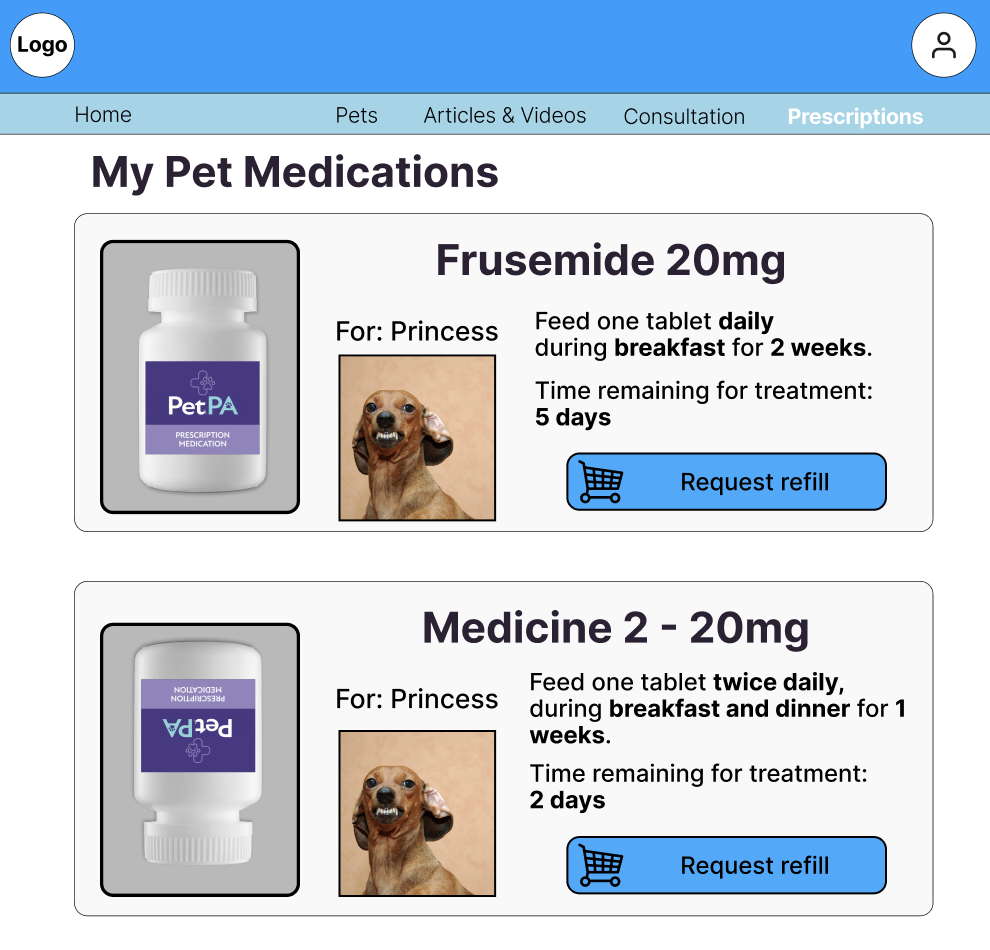
* Displays all relevant information about a pet, such as name, breed, age, medical history, and treatment plans. Editable areas allow users to update pet information, including health conditions and medications. Users can also view and share medical records in PDF format or via email.





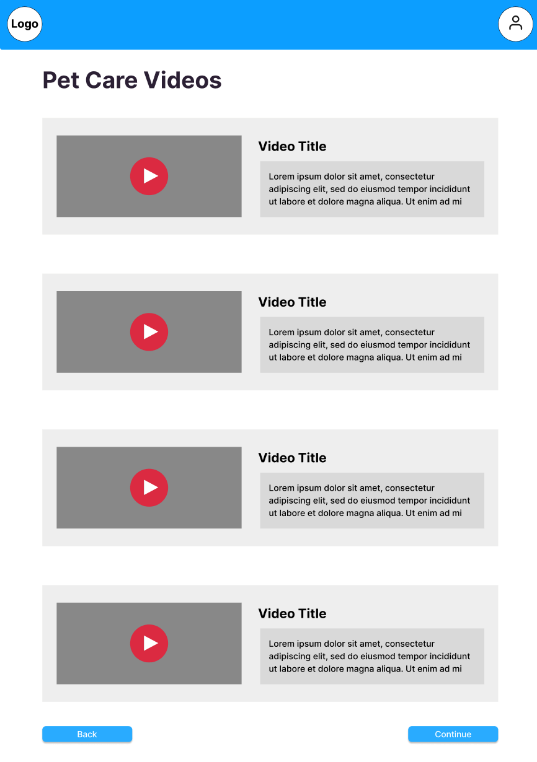
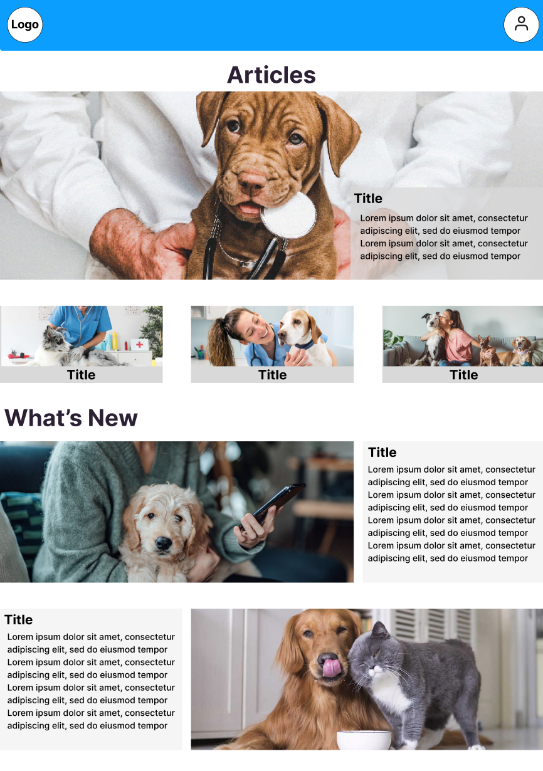
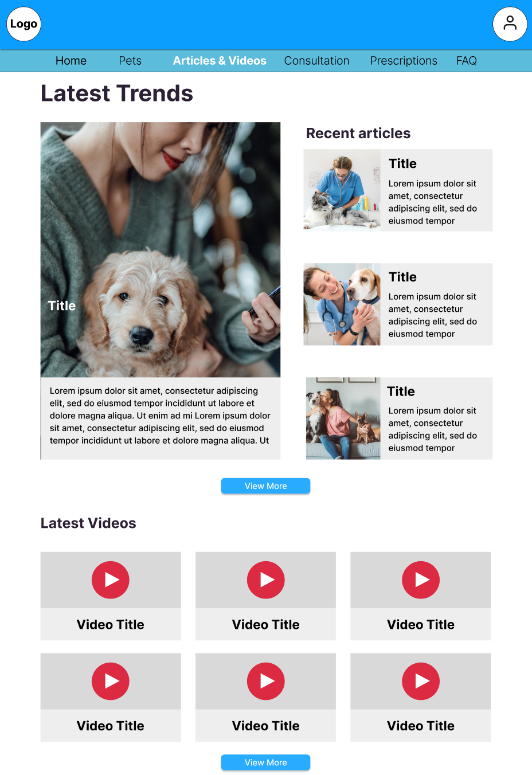
1. Medication Management Page

* A clear and concise list of all medications prescribed to the pet, including dosage instructions and frequency. Pet owners can request prescription refills.



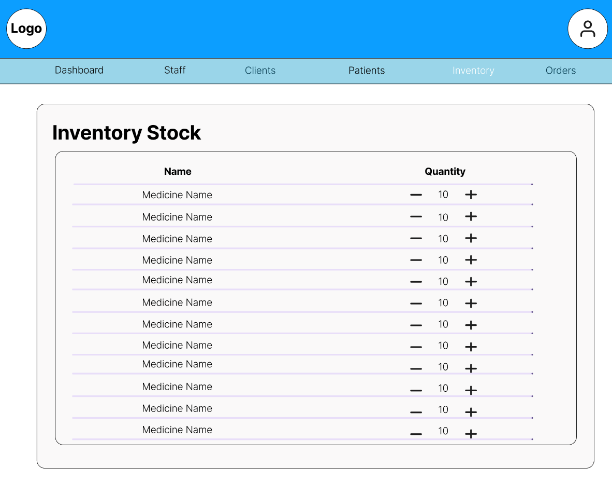
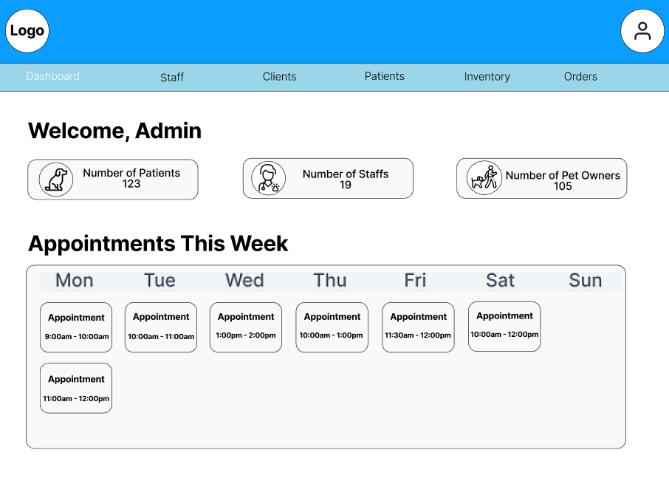
1. Educational Resource Pages

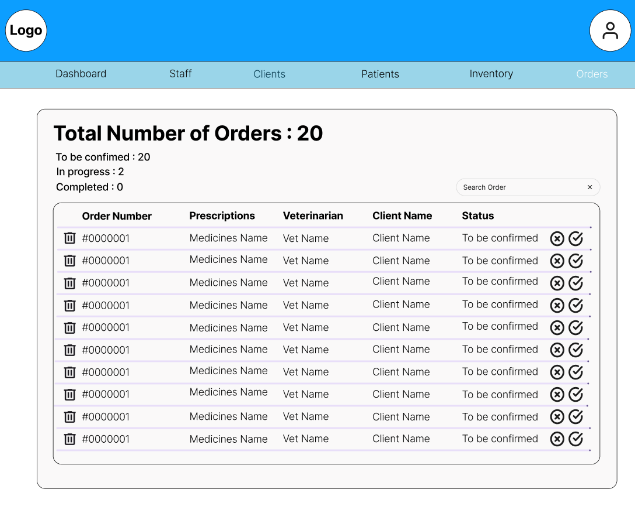
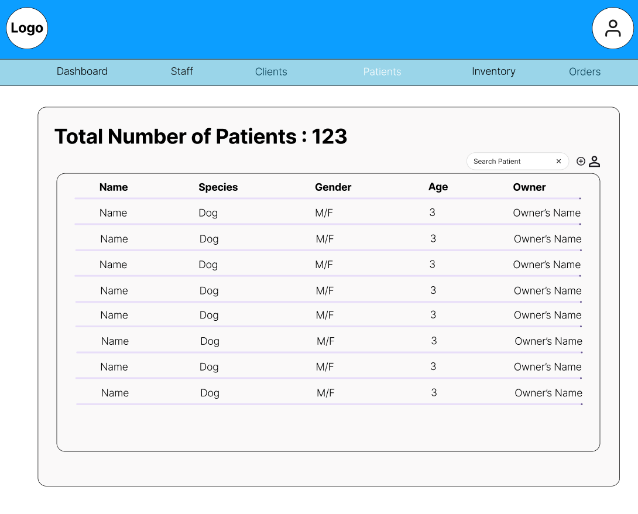
* A library of articles, videos, and guides on pet care and wellness. Users can filter content by category or search for specific topics of interest.



1. Admin Dashboard

* A central management hub for clinic administrators, allowing them to oversee appointments, pet records, and prescription stock.

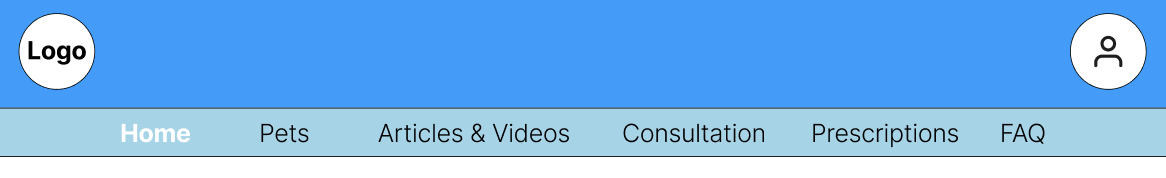




Navigation and Layout

The UI of VetCare can be easily navigated using the consistent top navigation bar throughout the application, featuring links to important sections such as:

* Home
* Pets
* Prescriptions
* Articles & Videos
* FAQ
* Profile



For mobile users, the top navigation collapses into a hamburger menu for easy access. Every page includes buttons with clear call-to-action labels like “Schedule Appointment,” “Edit Profile,” “Download Medical Records,” and “Request Refill,” ensuring that tasks can be completed efficiently.

Visual Design

The modern and clean aesthetic uses soft color palettes, such as light blues, which are soothing and resonate with the veterinary healthcare theme. The design promotes trust and professionalism while remaining approachable for users of all backgrounds.

* Typography: A combination of sans-serif fonts ensures superior legibility.
* Icons and Graphics: Familiar icons like calendars, medical symbols, and pills help users easily recognize key functions.
* Spacing and Alignment: Generous spacing ensures the interface is not cluttered, and actions are easy to find.

Error Handling and Notifications

The UI provides clear feedback for errors like invalid form submissions or network issues. Error messages are well-positioned with instructions on how to correct them. Notifications for upcoming appointments, prescription refills, or system updates are delivered via:

* In-App Notifications: Displayed at the top of the screen for quick visibility.
* Email Alerts: Sent for critical events like appointment confirmations or shared medical records.

Appendix A: Glossary

## A

* **Abstraction:** The process of hiding complex implementation details from the user and only exposing the necessary components. Repositories in the VetCare application provide an abstraction for data access.
* **Access Control:** Security measures that restrict access to data and resources to authorized users only. In VetCare, access control mechanisms ensure that only appropriate personnel can access sensitive information.
* **Account Management:** The functionality that allows users to create, update, and manage their personal and pet-related accounts within the VetCare system.
* **API (Application Programming Interface):** A set of protocols and tools that allow different software applications to communicate with each other. In VetCare, APIs are used to integrate with external systems such as payment processors and electronic health record (EHR) systems.
* **API Request:** A call made by the application to an Application Programming Interface (API) to retrieve or send data. In VetCare, API requests are handled by the Java backend.
* **Application Server:** The software framework that hosts and runs web applications. In this context, it refers to the servers running the VetCare application.
* **Appointment Management System:** A system that facilitates the scheduling, rescheduling, and cancellation of appointments. VetCare integrates with these systems to allow clients to book appointments online.
* **Autoconfiguration:** A feature of Spring Boot that automatically configures the application based on the dependencies present in the project, reducing the need for manual configuration.

## B

* **Back-end Logic:** The server-side operations that manage data processing, business rules, and database interactions. In VetCare, this is implemented using Java and Spring Boot.
* **Billing System:** A subsystem responsible for handling invoicing and payment processing for services provided. VetCare interfaces with external payment processing systems to manage billing.
* **Build Tool:** A software tool used to automate the process of compiling source code into executable programs and managing project dependencies. In this case, Maven is used.
* **Business Logic:** The part of the application that handles the core functionality and rules that govern how data is created, displayed, stored, and changed.
* **Business Rules:** The predefined conditions that govern the actions users can perform within the VetCare system, ensuring compliance with regulations and maintaining system security and efficiency.

## C

* **Client:** In the context of a REST API, the client refers to the application or system that makes requests to the server for data or services.
* **Client-Server Architecture:** A network architecture where client devices request resources or services, and the server provides them. In VetCare, the client layer interacts with the server layer, which processes requests and communicates with the database.
* **Clinic Administrator:** A role that may include clinic managers or administrative staff responsible for managing appointments, patient records, and overall clinic operations. They may also handle financial management and other administrative tasks.
* **Codebase:** The complete collection of source code used to build the VetCare application, which should follow standard coding practices for maintainability and reusability.
* **Continuous Deployment (CD):** The practice of automatically deploying code changes to production environments after passing CI tests, ensuring that updates are released quickly and efficiently.
* **Continuous Integration (CI):** The practice of automatically testing and integrating code changes into a shared repository multiple times a day to detect issues early.
* **Controller:** A component in the MVC design pattern that processes user input, interacts with models, and returns the appropriate view or data.
* **CRUD Operations:** The four basic operations of persistent storage—Create, Read, Update, and Delete. These operations are performed on the MySQL database by the repository layer using ORM.
* **CTH (Commonwealth of Australia):** Refers to the federal government of Australia. Relevant in the context of compliance with data protection regulations such as the Privacy Act 1988.
* **Cross-Platform Compatibility:** The ability of the VetCare system to operate on different operating systems (e.g., Windows, macOS) and be accessed through various web browsers without compatibility issues.

## D

* **Dashboard:** The main user interface screen of the web application where veterinarians can access and manage various features such as schedules, medical records, and professional resources.
* **Data Layer:** The layer in the VetCare application responsible for interacting with the database (MySQL), storing and retrieving data as needed.
* **Data Privacy:** Legal and ethical obligations to protect personal information from unauthorized access, disclosure, or misuse, in compliance with regulations like the Privacy Act 1988.
* **Data Protection Regulations:** Laws and regulations that govern the collection, storage, and processing of personal data. For VetCare, this includes compliance with the Privacy Act 1988.
* **Data Synchronization:** The process of ensuring that all data changes (e.g., scheduling appointments) made by users are accurately reflected and updated in the database.
* **Database Normalization:** A process in database design that organizes data to reduce redundancy and improve data integrity. VetCare's database uses normalization techniques to ensure efficient data storage.
* **Database Server:** The server that manages the database, handling queries, and storing data. For VetCare, MySQL is used.
* **Dependencies:** External libraries, frameworks, or software components that the VetCare application relies on for its functionality. These must be kept up to date to ensure security and performance.
* **Dependency Injection:** A design pattern used in Spring Boot to manage components and their dependencies, promoting loose coupling and making the code easier to test and maintain.
* **Docker:** A platform used for containerizing applications, ensuring that they work consistently across different computing environments.

## E

* **EHR (Electronic Health Record):** A digital version of a patient's medical history and treatment records. VetCare interfaces with EHR systems to manage and retrieve pet medical records.
* **Electronic Forms:** Forms presented online through the VetCare application for users to fill out and submit information, which is then stored in the database.
* **End-to-End Encryption:** A security measure that ensures data transmitted between the user's device and VetCare is encrypted to protect it from unauthorized access or tampering.

## F

* **FAQs (Frequently Asked Questions):** A list of common queries and issues that users might face, along with their answers, provided in a web-based format.
* **Foreign Key:** A field in a relational database table that creates a relationship between two tables, enforcing referential integrity by linking to the primary key of another table.

## H

* **Hashed Passwords:** Passwords that have been converted into a fixed-length string of characters, which cannot be directly converted back to the original password. Hashing is used to securely store passwords in the database.
* **Hero Article:** The most popular or featured article displayed prominently at the top of the article page. It includes the article's title and a short description to attract user attention.
* **HTTP Methods:** The methods used in REST APIs to perform actions, such as GET (retrieve data), POST (create data), PUT (update data), and DELETE (remove data).
* **HTTPS (HyperText Transfer Protocol Secure):** A secure version of HTTP used for transmitting data over the web. VetCare uses HTTPS to ensure secure communication between users' browsers and the server.

## I

* **Interoperability:** The ability of VetCare to exchange and use information with other systems and applications using standard data formats like CSV, JSON, and XML.

## J

* **Java:** The primary programming language used to develop VetCare, handling business logic, API requests, and database interactions.
* **Java Runtime Environment (JRE):** The environment required to run Java applications. VetCare requires JRE version 17 or later.
* **JSON (JavaScript Object Notation):** A lightweight data interchange format that is easy for humans to read and write and easy for machines to parse and generate. VetCare uses JSON for data exchange between the client and server.

## M

* **Maven:** A build automation tool used to manage project dependencies and the build process.
* **Medical Record System:** The component of VetCare responsible for storing and managing pet medical records, including vaccination history, treatment plans, and health summaries.
* **Model-View-Controller (MVC):** A design pattern used in software engineering to separate the application into three interconnected components: Model (data), View (user interface), and Controller (logic that handles input).
* **Modular Design:** A software design principle that breaks down the application into smaller, manageable, and independently functioning modules, which makes the system easier to maintain and update.
* **MySQL Database:** An open-source relational database management system used by VetCare for storing user data, including pet records, prescriptions, and appointment histories.

## O

* **Object-Oriented Programming (OOP):** A programming paradigm based on the concept of "objects," which contain data and methods to manipulate that data, aimed at enhancing reusability and maintainability.
* **ORM (Object-Relational Mapping):** A programming technique used to convert data between incompatible systems (object-oriented programming languages and relational databases). In VetCare, ORM (like Hibernate) is used for database interactions.

## P

* **Patch Fixes: Software updates released to fix bugs or vulnerabilities after** the initial deployment of the application.
* **Pop-Up:** A user interface element that appears in a small window on the screen, used in VetCare to present important information such as consent requests to users.
* **Prescription Management System:** A component of VetCare responsible for managing prescription details, facilitating refill requests, and handling delivery arrangements.
* **Privacy Act 1988:** Australian legislation that governs the handling of personal information. VetCare must comply with this act to ensure proper data protection and user privacy.

## R

* **Referential Integrity:** A database concept that ensures relationships between tables remain consistent, typically enforced through foreign keys.
* **Relational Database:** A type of database that stores data in tables with rows and columns, allowing for relationships between different tables. VetCare uses a relational database to manage its data.
* **Reminder Notification:** A feature in VetCare that sends alerts to pet owners about upcoming appointments or other important events. Notifications are sent via the application to ensure timely reminders.
* **Responsive Design:** A design approach that ensures the application’s user interface adjusts smoothly and remains user-friendly across different screen sizes and resolutions, particularly for mobile devices.
* **REST (Representational State Transfer):** An architectural style for designing networked applications. RESTful services in VetCare handle HTTP requests and provide JSON responses.
* **REST API (Representational State Transfer Application Programming Interface):** An architectural style for designing networked applications that provides a way of communication between client and server through standard HTTP methods.
* **Robustness:** The application's ability to handle incorrect inputs or unexpected conditions gracefully, without crashing.
* **Role-Based Access Control (RBAC):** A security mechanism that restricts access to data and functionality within the application based on the user's role, ensuring that users can only access information relevant to their job function.

## S

* **Scalability:** The ability of the VetCare application’s database and server infrastructure to handle increased loads, such as more users and data, without decreasing performance.
* **Secure Storage:** The practice of protecting sensitive user data, such as medical records and payment information, from unauthorized access through various security measures.
* **Sensitive Medical Data:** Personal and health-related information that requires secure handling and compliance with privacy regulations.
* **Server:** In the context of REST APIs and web applications, the server is the system that processes requests from the client and returns the appropriate response.
* **Service Layer:** The layer in the MVC pattern that contains the business logic of the application and acts as a mediator between the controllers and the database.
* **SFTP (Secure File Transfer Protocol):** A secure protocol used by VetCare to allow users to download files, such as medical history records, safely.
* **Spring Boot:** A Java-based framework used to create production-ready applications with minimal configuration, facilitating the development of web applications.
* **SRS (Software Requirements Specification):** A document that outlines the requirements and functionality of the VetCare system and serves as a guide for development, testing, and implementation.
* **Standardized Data Formats:** Commonly accepted formats for data exchange, such as CSV, JSON, and XML, used to ensure compatibility and consistency across different systems.
* **Subsystem:** A smaller, self-contained system or component that integrates with VetCare to provide specific functionalities. Examples include the User Management System and the Prescription Management System.

## T

* **Thumbnail:** A small, clickable image that represents a video and are typically used to give users a preview or visual representation of the video content before they decide to watch it.

## U

* **Unit Testing:** A software testing method where individual components of the application are tested in isolation to ensure they work correctly. JUnit5 is used for this purpose.
* **User Characteristics:** Traits and preferences of the system’s users, including their level of technical expertise and their needs for convenience, security, and ease of use.
* **User Consent:** The process of obtaining explicit permission from users before collecting, storing, or processing their personal data. In VetCare, this includes providing clear information about data usage and obtaining consent through a pop-up with "I Agree" or “I Disagree” options.
* **User Documentation:** The materials provided to help users understand and effectively utilize the VetCare application, including FAQs and other guides.
* **User Interface (UI):** The visual part of the application that users interact with. It includes elements like buttons, menus, and dashboards designed to allow easy navigation and efficient management of tasks.
* **User Management System:** Manages user registration, login, and profile management for both pet owners and veterinarians in VetCare.
* **User Roles:** Defined categories of users in VetCare with different levels of access and permissions, such as admin, veterinarians, and pet owners.

## V

* **Validation Rules:** Constraints set at the database level to ensure that data entered into the system conforms to specified formats and standards, such as valid email addresses and phone numbers.
* **Verification Checks:** Processes used to confirm the identity of users, typically through the comparison of entered credentials with stored, hashed passwords during login.
* **Veterinary Practice Act 1997:** Legislation that governs the standards and practices for veterinary professionals to ensure that the VetCare application complies with relevant professional regulations.
* **View Controllers:** Components in the MVC pattern that handle web page requests and responses to determine what the user sees.

## W

* **Web Application:** A software application that operates through a web browser which provides a platform for users to interact with VetCare’s features and services.
* **Web Framework:** The framework used to build and manage web applications. Spring Boot is the web framework for VetCare.
* **Workflow Efficiency:** The optimization of the sequence of processes through which a piece of work passes from initiation to completion which is often improved using design patterns like MVC.

Appendix C: To Be Determined List