Software Requirements Specification

for

VetCare – Online Vet Clinic Management System

Version 1.0 approved

Prepared by Group 05_06

RMIT University

18/08/2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document describes the software requirements for the VetCare – Online Vet Clinic Management System, version 1.0. The VetCare application is designed to provide pet owners with a comprehensive platform for managing their pets' health care conveniently. They can access numerous features such as appointment scheduling, access to medical records, prescription management, and more. This document covers the entire system, detailing the requirements for all functionalities within the scope of the VetCare project.

1.2 Document Conventions

This SRS follows the IEEE standard for documenting software requirements. Key terms are highlighted in bold and italicized to emphasize their importance.

1.3 Intended Audience and Reading Suggestions

This document is intended for the following audiences:

- **Developers**: To understand the system requirements and implement the software.
- **Testers**: To create test cases and ensure the system meets the requirements.
- **Documentation Writers**: To prepare user manuals and help guides.
- **Stakeholders and Clients**: To review the system's features and ensure it meets their expectations.

The document is organized starting with the system overview, followed by detailed requirements. It is recommended to begin with the introduction and product scope, then proceed to the relevant sections.

1.4 Product Scope

VetCare is a web-based application designed to connect pet owners with local veterinary clinics and pet care providers. The application enables users to book appointments, access their pets' medical records, manage prescriptions, and explore educational resources on pet care. The system is aligned with the goal of improving pet health and wellness by providing a user-friendly platform that integrates with various service providers in the local area.

1.5 References

- **IEEE SRS Standard**: IEEE 830-1998, Recommended Practice for Software Requirements Specifications.

2. Overall Description

2.1 Product Perspective

The VetCare application is a new product designed to revolutionize the way pet owners interact with veterinary services. It is not a follow-on product or a replacement for existing systems but is intended to fill a gap in the market by offering comprehensive vet clinic management functionalities. The system interfaces with various veterinary clinics, pet care providers, and pharmacies, enabling seamless integration and data exchange. The application will be deployed on a cloud-based platform, ensuring scalability and accessibility.

2.2 Product Functions

Functional Requirements

Creating an Account

- The user can create an account with an email and password
- The system will create the account and provide a validation email for security reasons
- The system checks the password before validating making sure the password is not easily hackable.

Logging into an Account

- Users can log into their registered accounts
- The system will check the login information and log the user in if correct
- The system will have options for the user to reset their password if forgotten

Creating a pet profile

- The system provides a Pets section and an add a pet profile option
- The system will give a pet details form for the user to complete
- The System will save the pet profile information and show a confirmation message

Scheduling an Appointment with a Vet

- The system will provide a "Schedule an Appointment" page
- The system will provide users with an interface to select a vet clinic, choose a free date and time, and confirm the appointment
- The system will ensure the time and date is free before confirming the appointment
- The system should save the appointment details and display a confirmation message.
- The system will add the appointment to an "upcoming appointments" page

Scheduling Consultations for Vets

- The system will provide a Consultation Schedule page for the vets to access.
- The system should allow Vets to select dates and times they are available for consultations.
- The system will allocate the selected time slots and mark them as reserved or open
- The System will provide the option for Vets to offer in-person or virtual consultations which will be easily differentiated in the schedule view.

Canceling an Appointment

- The system will allow pet owners to cancel an appointment during the booking stage
- The system should display a confirmation message when the cancel button is clicked
- The System will navigate the user back to the beginning of the appointment book page

Searching for specific Vet and Location and Booking

- The system will provide an option for users to search for vets and clinics by name or specialty
- The system will show a list of matches specific to the user's criteria
- The system will display the selected vet's profile with available dates and booking times.
- The system will allow users to complete the booking process after selecting an available date and time.

Setting Up Notifications for Pets Vaccines and Treatments

- The system will allow users to access their pet's medical profile and click the notification settings section
- The system will display options for setting up notifications for the pet's upcoming vaccinations and treatments
- The system will provide an option for users to select specific vaccines and treatments that they want to be notified on
- The system will provide options for different notification methods

Accessing Pet Medical Records

- The system will provide an option that allows pet owners to access the My Pets section and click a specific pet profile.
- The system will display information related to the pet including medical records, vaccination history, treatments, and allergies
- The system will allow users to download, print, or view this information

Viewing Pet Medication and Dosage Instructions

- The system should provide users with the ability to access the Medications tab within a pet profile
- The system will display a list and information on the pet's current and past medication.
- The system will allow users to download, view, or print this information

Vets securely accessing pet's Medical Records

- The system will allow vets to access medical records provided by pet owners for consultations.
- The system will display the medical information required for medical information

Requesting Prescription Refills Online

- The system will allow users to request prescription refills through the Medicines tab in the pet profile.
- The system should provide a home delivery option
- The system should display a delivery form for the user to fill out
- The system should process the form and provide the user with an estimated delivery date
- The system should provide the user with a confirmation message once everything is processed

Searching and Filtering Education Content

- The system will provide a large amount of educational content related to animal care.
- The system will allow users to search and filter the education content based on many parameters such as topics, pet breed, age, and condition
- The system will display the required content based on the selected filters

Saving Educational Resources for Offline Access

- The system will allow users to save educational content to access offline
- The system will download the selected content to the users device
- The system will display downloaded content in an offline downloads tab

User-Friendly Interface

- The system should provide clear and concise labels for each option across the website
- The system will display feedback in case errors occur

The major functions of the VetCare system include:

- **Appointment Scheduling**: Book, reschedule, and cancel appointments with veterinary clinics.
- **Medical Records Access**: Securely view, manage, and share pets' medical records.
- **Prescription Management**: Request refills and access medication information.
- **Educational Resources**: Explore articles, videos, and guides on pet care.
- **Reminders and Notifications**: Receive alerts for appointments and medication schedules.
- **Online Payment**: Pay for services and products securely through the platform.

2.3 User Classes and Characteristics

The primary user classes anticipated for this product are:

- **Pet Owners**: Users who book appointments, access medical records, and manage prescriptions.
- **Veterinarians**: Professionals who provide services, manage appointments, and update medical records.
- **Clinic Administrators**: Users responsible for managing clinic operations, scheduling, and patient records.
- **Pharmacy Staff**: Users who manage prescriptions and process medication orders.
- **System Administrators**: Technical staff responsible for maintaining and updating the VetCare system.

Each user class will have different access levels and functionalities based on their role.

2.4 Operating Environment

The VetCare system will operate on a cloud-based platform compatible with the following:

- **Hardware**: Web servers, cloud storage solutions.
- **Operating Systems**: Linux-based servers, Windows, macOS, iOS, Android for client devices.
- **Web Browsers**: Chrome, Firefox, Safari, Edge (latest versions).
- **Database**: MySQL for backend data management.
- **Security Protocols**: HTTPS for secure data transmission.

2.5 Design and Implementation Constraints

The design and implementation of VetCare will be constrained by:

- **Regulatory Requirements**: Compliance with data protection laws, such as GDPR.
- **Security Considerations**: Implementation of encryption for data storage and transmission.
- **Performance Requirements**: The system must support a high volume of concurrent users without performance degradation.
- **Technology Stack**: The use of a specific set of technologies, including Springboot and Thymeleaf, and MySQL for the database.
- **User Interface Standards**: Adherence to the VetCare UI Standards document.

2.6 User Documentation

The following user documentation will be delivered with the VetCare system:

- **User Manual**: A comprehensive guide for pet owners on how to use the application’s features.
- **Admin Guide**: Instructions for clinic administrators on managing the system.
- **Online Help**: Context-sensitive help integrated within the application

2.7 Assumptions and Dependencies

Key assumptions and dependencies include:

- **Third-Party APIs**: Integration with external APIs for appointment scheduling and payment processing.
- **Internet Access**: The system assumes stable internet connectivity for all users.
- **Clinic Participation**: The success of the system depends on the participation of local veterinary clinics and pharmacies.
- **Browser Compatibility**: The system assumes users will access it through supported browsers.

3. External Interface Requirements

3.1 User Interfaces

The user interface should be consistent across all pages. Additonally the user interface should follow UI design concepts to ensure the best interactions with users.

3.2 Hardware Interfaces

3.3 Software Interfaces

3.4 Communications Interfaces

4. Nonfunctional Requirements

4.1 Performance Requirements

- **Creating an Account:** Creating an account will be straightforward and efficient.
- **Logging into an Account:** The login process should be validated within 10 seconds.
- **Creating a Pet Profile:** The system should be able to handle multiple pets in one user account without performance issues.
- **Scheduling an Appointment with a Vet:** Scheduling and confirming appointments should be processed quickly (efficiently).
- **Searching for a Specific Vet and Location and Booking:** Search results should be displayed within 3 seconds.
- **Setting Up Notifications for Pets Vaccines and Treatments:** Notification delivery should be close to the scheduled notification time.
- **Accessing Pet Medical Records**: The pet information should be displayed within 2 seconds of opening the pet profile.
- **Viewing Pet Medication and Dosage Instructions:** The system should display information quickly after the medications tab is clicked.
- **Requesting Prescription Refills Online:** The system should process prescription refills within 1 minute.
- **Searching and Filtering Educational Content:** The system should be able to search multiple parameters simultaneously without performance lag.
- **Saving Educational Resources for Offline Access:** No download should take longer than 20 minutes.
- **User-Friendly Interface:** The home page interface should load within 2 seconds.
- **Requesting Prescription Refills Online:** The system can hold multiple prescription refill requests within a performance drop-off or crash.

4.2 Safety Requirements

- **Setting Up Notifications for Pets Vaccines and Treatments:** Notifications should be sent at least 48 hours before a scheduled vaccination or appointment.

4.3 Security Requirements

- **Creating an Account:** The system will store account data securely.
- **Creating a Pet Profile:** The system should save and encrypt the information (security).
- **Accessing Pet Medical Records:** The system should store all pet information securely with encryption.
- **Vets Securely Accessing Pet's Medical Records:** The system should ensure data security so that only the vets can access the records.
- **Requesting Prescription Refills Online:** The system will ensure payment details are encrypted and processed safely.

4.4 Software Quality Attributes

- **Canceling an Appointment:** A user-friendly UI should be provided so the user is not confused at any step in the process.
- **Scheduling an Appointment with a Vet:** The system should handle multiple account appointment books simultaneously with high performance (Scalability) (Efficiency).
- **Scheduling Consultations for Vets:** The schedule interface should be clear and display the different consultation types.
- **User-Friendly Interface:** The website should maintain consistent design across all pages of the website.
- **User-Friendly Interface:** The system will be available across multiple devices.

4.5 Business Rules

- **Scheduling an Appointment with a Vet:** Data must be managed well to ensure no double bookings (Reliability).
- **Canceling an Appointment:** The system should ensure that canceled booking data is not saved (Reliability).

5. Other Requirements

6. System Architecture

Summary of the Fundamental Decisions and Solution Strategies that Shape the Architecture

The VetCare application follows a modular N-tier architecture, leveraging Spring Boot as the backbone for backend services and Thymeleaf for the frontend. The architecture prioritizes scalability, security, and maintainability while enabling seamless integration with external systems and services. Key decisions involve the use of microservices for core features, containerization for deployment, and cloud-based tools for continuous integration and delivery.

Technology Choices:

- The backend is built using Java with Spring Boot, ensuring efficient management of business logic and service orchestration. The frontend is rendered using Thymeleaf templates for dynamic content generation.
- Data management is handled with MySQL, ensuring robust and scalable relational data storage.
- Redis is employed for caching to enhance performance by reducing database load.
- Docker and Docker Compose are used for containerization, enabling consistent development, testing, and deployment across environments.
- GitHub Actions is utilized for CI/CD pipelines, ensuring automated testing and deployment processes.

Top-Level Decomposition: The system is divided into the following tiers and components:

- Presentation Tier (UI/UX Components): Manages user interactions and provides dynamic content using Thymeleaf.
- Security and Authentication Components: Implements authentication and authorization using Spring Security with OAuth 2.0 and JWT.
- Application Tier (Business Logic & Service Layer): Orchestrates business logic, service communication, and API integrations.
- Data Tier (Database, APIs, and Data Access Layer): Handles data persistence, external API communication, and database interactions.
- Plugins for Extended Features: Provides modular integration points for additional features like educational resources and notifications.
- Kernel (Core Logic): Manages core service routing and request processing.
- Caching Layer: Utilizes Redis for caching frequently accessed data, improving response times.

Approaches to Achieve Top Quality Goals:

- Scalability: The microservices-based architecture, combined with Docker for containerization and Redis for caching, ensures that the system can scale horizontally.
- Security: The application enforces robust security protocols using Spring Security, OAuth 2.0, and JWT, ensuring secure data access and management.

- Flexibility: The modular architecture with plugins allows for easy integration of new features without disrupting the existing system.
- Reliability: CI/CD pipelines powered by GitHub Actions, combined with automated testing, ensure consistent and reliable releases.
- Organizational Decisions: The project adheres to Agile principles, with Scrum as the project management methodology. Continuous progress is ensured through frequent sprints, daily standups, and iterative releases.

6.1 Architecture Overview

The architecture is represented as a hierarchy of components, with white boxes encapsulating black boxes, depicting the system's internal structure.

Presentation Tier (UI/UX Components):
UI Components: Uses Thymeleaf for server-side rendering of dynamic HTML pages.
Controllers: Handle HTTP requests and return appropriate Thymeleaf views.
Form Handling: Server-side validation and processing of user inputs (e.g., appointment scheduling).

Security and Authentication Components:
Spring Security: Implements secure authentication and authorization using OAuth 2.0 and JWT.
User Services: Provides secure login, registration, and user management workflows.

Application Tier (Business Logic & Service Layer):
Service Layer: Manages business rules and coordinates between different microservices.
REST APIs: Facilitates communication between the application and external systems (e.g., third-party service providers).

Data Tier (Database, APIs, and Data Access Layer):
MySQL: Manages structured data storage for user profiles, appointments, and medical records.
Spring Data JPA: Handles data persistence and repository management for efficient data access.
API Integrations: Manages data exchange with external services (e.g., education material on pets).

Plugins for Extended Features:
Modular Features: Supports additional functionalities like notifications, educational content, and reminders.
Extension Points: Allows seamless addition of new services without affecting existing modules.

Kernel (Core Logic):
Service Orchestration: Manages service communication and load balancing using Spring Cloud.
Request Handling: Directs incoming requests to appropriate microservices and handles responses.

Caching Layer:
Redis: Caches frequently accessed data, such as appointment availability, to improve performance and reduce database load.

6.2 Architectural Decisions

Java with Spring Boot and Thymeleaf:

Rationale: Provides a cohesive environment for managing both backend services and frontend rendering, simplifying integration and maintenance.
Risk: Server-side rendering might be less suitable for highly dynamic UIs, requiring additional optimization.

Spring MVC for Presentation Layer:
Rationale: Offers seamless integration with Thymeleaf for generating dynamic HTML content.
Risk: Limited flexibility for handling client-side interactions compared to modern JavaScript frameworks.

MySQL for Data Management:
Rationale: Provides a reliable and scalable relational database solution, ideal for managing structured data in the VetCare application.
Risk: Requires careful schema design and indexing to maintain performance with growing data volumes.

Redis for Caching:
Rationale: Enhances performance by caching frequently accessed data, reducing the need for repetitive database queries.
Risk: Cache invalidation must be handled carefully to avoid stale data issues.

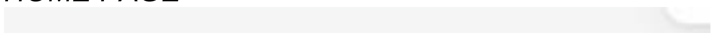
Spring Security with OAuth 2.0 and JWT:
Rationale: Ensures secure authentication and access control, with the flexibility to handle different user roles and permissions.
Risk: Requires careful configuration to avoid vulnerabilities related to token management and access control.

Containerization with Docker and CI/CD with GitHub Actions:
Rationale: Ensures consistent deployment across environments and supports automated testing and deployment pipelines.
Risk: Requires proper management of container resources and dependency versions across different environments.

7. User Interface Design

PET PROFILE

HOME PAGE



LOGIN

USER PROFILE

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>