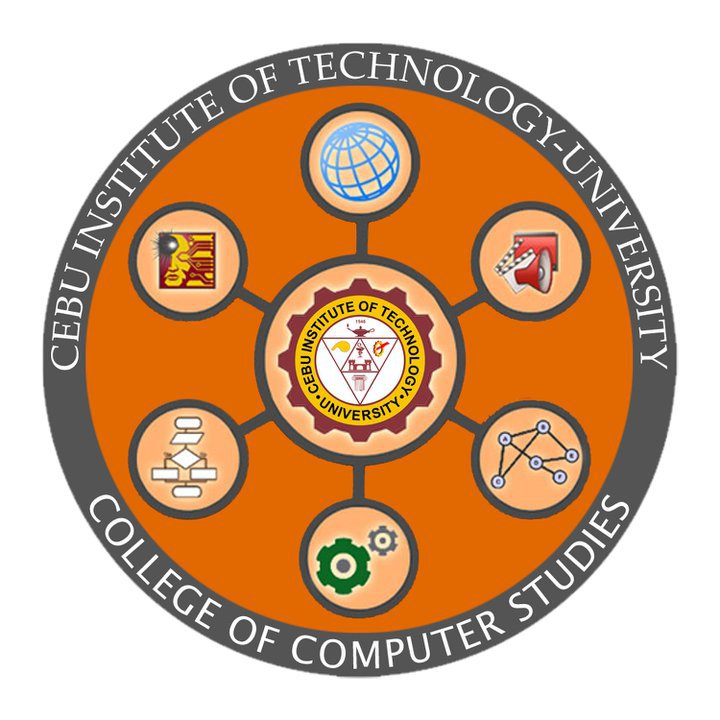


**CEBU INSTITUTE OF TECHNOLOGY**

**UNIVERSITY**

****

COLLEGE OF COMPUTER STUDIES

Software Requirements Specifications

for

FurrEverCare

prepared by:

Genosolango, Selina Mae V.

Melocoton, Jeric Kiel B.

Quilo, Jan Isaac S.

**Change History**

**Table of Contents**

**Change History 2**

**Table of Contents 3**

**1.** **Introduction 4**

1.1. Purpose 4

1.2. Scope 4

1.3. Definitions, Acronyms and Abbreviations 4

1.4. References 4

**2.** **Overall Description 5**

2.1. Product perspective 5

2.2. User characteristics 5

2.4. Constraints 5

2.5. Assumptions and dependencies 6

**3.** **Specific Requirements 7**

3.1. External interface requirements 7

*3.1.1.* *Hardware interfaces 7*

*3.1.2.* *Software interfaces 7*

*3.1.3.* *Communications interfaces 7*

3.2. Functional requirements 7

*Module 1 7*

*Module 2 8*

3.4 Non-functional requirements 8

*Performance 8*

*Security 8*

*Reliability 8*

# Introduction

**FurrEverCare** is a mobile and web-based application designed to help pet owners manage their pets’ health more easily. It provides tools for tracking medical information, handling emergencies, and accessing reliable pet health resources. With a focus on convenience and accessibility, **FurrEverCare** supports pet owners in making informed decisions and ensuring the well-being of their pets.

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the functional and nonfunctional requirements for the **FurrEverCare** application. This document serves as a comprehensive reference for the development team, stakeholders, and end-users, ensuring a clear understanding of the system’s capabilities, limitations, and objectives.

The intended audience for this document includes:

* **Developers –** To implement features and ensure alignment with project goals.
* **Professor –** To assess the system’s completeness, functionality, and adherence to requirements.
* **Pet Owners(End-Users) –** To represent the target audience for usability testing and feature validation.

## Scope

The software product to be produced is **FurrEverCare**, a mobile and web-based application designed to support pet owners in managing their pets’ health and overall well-being.

**1.2.1 In Scope**

a. Health tracking to log treatments, medications, and appointments for pets.

b.Integrate an emergency pet profile card systemfor quick access to critical pet health information in emergencies.

c. Real-time notifications for treatment schedules, check-ups, and medication.

d. Interactive pet wellness timeline to visually track and monitor pets' medical history and health progress.

e. Integration of a symptom checker powered by AI.

f. Searchable pet health condition and disease encyclopedia for pet owners to explore common conditions, symptoms, and treatment options.

g. First-Aid Visual Instructions with offline access to guide pet owners through emergency situations.

h. Offline mode functionality for the first-aid visual instructions.

**1.2.2 Out-of-Scope**

a. Live veterinary consultation**s** or direct messaging with veterinary professionals.

b. Financial transactions related to veterinary services or pet care products.

c. Advanced biometric authentication beyond facial recognition.

## Definitions, Acronyms and Abbreviations

* **Veterinarian / Vet:** A licensed medical professional specializing in animal health diagnosis, treatment, and surgery. In FurrEverCare, vets may have elevated access to pet health timelines under RBAC rules.
* **Health Tracking:** The systematic recording and monitoring of a pet’s health-related activities, including vet visits, medication administration, symptom progression, and diagnostic test results
* **First-Aid Visual Instructions:** Step-by-step graphical guides within the app demonstrating emergency procedures for pets, such as CPR, wound cleaning, or choking relief
* **Pet Wellness:** The holistic management of a pet’s physical, mental, and emotional health through preventive care, including nutrition, exercise, dental hygiene, and stress reduction.
* **Emergency Pet Profile Card System:** A digital profile containing critical pet information (e.g. blood type, chronic conditions, emergency contacts) designed for rapid access during crises.
* **Core (Vaccines/Procedures):** Essential vaccines, treatments, or preventive care measures universally recommended for all pets regardless of geographic location, lifestyle, or breed. These protect against severe, life-threatening, or legally mandated diseases.
* **Non-Core (Vaccines/Procedures):** Optional vaccines, treatments, or preventive care recommended based on pet’s specific risk factors, such as geographic exposure, lifestyle (e.g. outdoors access), or breed susceptibility. Administered after vet consultation.
* **Vaccination Record:** A verified digital log of vaccines administered to a pet, including dates, types (core/non-core), and next due dates.
* **Vaccination:** The process of administering antigenic material (vaccines) to stimulate immunity against specific diseases. Core vaccines (e.g. rabies) are legally required; non-core (e.g., Lyme disease are optional based on risk factors.
* **Grooming:** Routine hygiene and maintenance activities for pets, such as bathing, nail trimming, ear cleaning, and coat brushing, documented via the app’s grooming scheduler.
* **Medication Doses:** The precise quantity, frequency, and administration method of drugs prescribed for a pet.
* **MFA (Multi-Factor Authentication):** A login process requiring two or more verification methods (e.g. password + SMS code).
* **RBAC (Role-Based Access Control):** Restricting system based on user roles (e.g. owner vs. vet).
* **Deworming Schedule:** Appointed timetable for administering antiparasitic treatments
* **Triage:** Prioritizing emergency care based on severity of a pet’s condition.

## References

Firebase quickly syncs data via Android, iOS, and JavaScript SDKs, allowing for expressive queries that scale with the size of the results set. Users who don’t require real-time data syncs might still benefit from the one-time reads feature. Firebase also has real-time data syncing capabilities-all changes made to the stored data are instantly propagated to any connected service-which is ideal for cross-platform mobile application development.

Firebase also has horizontal scalability compared to MySQL which is vertically scalable.

<https://www.integrate.io/blog/firebase-vs-mysql/>

Vertical Scaling has a higher possibility for downtime and unless we have a backup server that can handle operations and requests,we will need some considerable downtime to upgrade your machine. Additionally, there is a limitation to how much we can upgrade a machine as every machine has its threshold for RAM, storage, and processing power. Whereas Horizontal scaling has increased resilience and fault tolerance which is by relying on a single node for all our data and operations puts us at a high risk of losing it all when it fails. Distributing it among several nodes saves us from losing it all. Additionally, because if we will be adding a machine, we don’t have to switch the old machine off while scaling. If done effectively, there may never be a need for downtime and clients are less likely to be impacted.

This fits FurrEverCare due to:

1. Traffic Spikes: mobile users might trigger sudden surges (e.g. emergencies requiring First-Aid Instructions). Horizontal scaling handles unpredictable loads.
2. Cross-Platform Consistency: Web and mobile clients share the same backend. Horizontal scaling ensures synchronized data (e.g., Health Timeline updates).
3. Firebase’s Native Support: Firebase Firestore and Cloud Functions auto-scale horizontally, aligning with our requirement to handle a 50% user increase effortlessly.

<https://www.cloudzero.com/blog/horizontal-vs-vertical-scaling/#:~:text=While%20horizontal%20scaling%20refers%20to,%2C%20storage%2C%20or%20network%20speed>.

DeepSeek, compared to ChatGPT, has 90% accuracy surpassing GPT-4o’s 83% accuracy on advanced benchmarks in Mathematics Metric. DeepSeek also features: faster processing for task-specific solutions, more affordable, with efficient resource usage, optimized for technical and specialized queries and provides well-structured and task-oriented responses.

<https://www.geeksforgeeks.org/deepseek-vs-chatgpt/>

# Overall Description

**FurrEverCare** is a mobile and web-based application that helps pet owners manage their pets' health and access important information when needed. The mobile app offers tools for tracking medical and treatment details, managing emergencies with an emergency pet profile card system, and providing first-aid visual instructions.

Both the mobile and web platforms feature an interactive pet health timeline, while the web platform also includes resources for checking symptoms, exploring pet health conditions, and monitoring overall well-being. Designed for convenience and reliability, **FurrEverCare** supports pet owners in making informed decisions about their pets' care.

## Product perspective

**FurrEverCare** is a self-contained mobile and web-based application designed specifically to assist pet owners in managing their pets’ health, wellness, and emergency care. Unlike other pet care apps, FurrEverCare provides an integrated solution that combines health tracking, emergency management, AI-driven symptom checking, and wellness timelines in one platform. It operates as an independent product, which does not rely on external systems to provide core functionalities.

While the product is self-contained, it interacts with external services for certain features such as mapping services (Google Maps for location-based services) and AI-powered tools (DeepSeek for symptom checking). Additionally, the app can sync data across different devices, allowing for real-time updates and cross-platform accessibility. The modular decomposition of FurrEverCare breaks down the app into distinct, functional modules to ensure maintainability, scalability, and ease of use.

**Module 1: User Profile Management & Authentication**

* Transaction 1.1: **User Registration & Login** 
  + Handles user account creation, authentication (with multi-factor authentication), and secure login.
* Transaction 1.2: **Pet Profile Creation & Management** 
  + Allows users to create and manage profiles for each pet, including breed, age, medical history, etc.
* Transaction 1.3: **Profile Synchronization Across Devices**
  + Ensures that all user and pet data sync seamlessly between mobile and web platforms.

**Module 2: Health Tracking & Emergency Profile**

* Transaction 2.1: **Medication & Treatment Scheduling** 
  + Allows users to schedule and track treatments, medications, and appointments with reminders.
* Transaction 2.2: **Emergency Pet Profile Card System**
  + Generates a digital emergency ID card for each pet with crucial health information (allergies, vet contacts, vaccination records) accessible offline and shareable via QR code.
* Transaction 2.3: **Notifications & Alerts** 
  + Sends real-time notifications for upcoming treatments, appointments, and medication refills.

**Module 3: AI-Powered Symptom Checker & Encyclopedia**

* Transaction 3.1: **AI Symptom Analysis** 
  + A chatbot-driven feature powered by AI (DeepSeek) that analyzes symptoms entered by users and suggests possible conditions based on breed and age.
* Transaction 3.2: **Pet Health Encyclopedia**
  + A searchable, veterinary-reviewed database of common pet conditions, symptoms, and treatments.

**Module 4: First-Aid Visual Guides**

* Transaction 4.1: **First-Aid Guides & Tutorials**
  + Provides step-by-step, video-based visual guides for common emergencies such as choking, wounds, or seizures.
* Transaction 4.2: **Offline Access**
  + Ensures that first-aid guides are available offline, especially in areas with poor connectivity.

**Module 5: Wellness Timeline**

* Transaction 5.1: **Medical History Tracking** 
  + Tracks and visualizes the pet’s medical history, including vet visits, treatments, and diagnoses.
* Transaction 5.2: **Timeline Export & Sharing** 
  + Allows users to export their pet’s wellness timeline (e.g. as PDFs) for sharing with vets, pet sitters, or insurance companies.
* *Put software product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the SRS defines a product that is a component of a larger system, as frequently occurs, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software.*
* *A block diagram showing the major components of the larger system, interconnections, and external inter- faces can be helpful.*
* *Describe the modular decomposition of the components using the format below:*

*Module 1*

*Transaction 1.1*

*Transaction 1.2*

*Module 2*

*Transaction 2.1*

*Transaction 2.2*

*. . .*

## User characteristics

### 2.2.1. Pet Owner

The Pet Owner is the primary user of FurrEverCare, responsible for managing their pet’s health data, tracking medications, and accessing emergency resources.

**Privileges:**

1. Create, edit, and delete pet profiles (e.g., breed, age, medical history).
2. Schedule medications, vet appointments, and set reminders.
3. Generate and share emergency profile cards via QR code or link.
4. Access offline first-aid guides and symptom-checking tools.
5. View and update the interactive pet health timeline (e.g., vet visits, treatments).
6. Submit inquiries or feedback through the app.

### 2.2.2. Administrator

The Administrator oversees system operations, manages user accounts, and ensures platform stability.

**Privileges:**

1. Create, suspend, or delete user accounts (e.g., reset passwords for demo purposes).
2. Monitor system performance and resolve technical issues.
3. Manage database backups and ensure data integrity for the deployed prototype.
4. Review basic usage analytics (e.g., number of registered pets, active users).
5. Configure system settings (e.g., enable/disable features for testing).

## 2.4. Constraints

### 2.4.1. Development

### 2.4.1.1. Time and Resource Limitations

* Three-month development timeline with concurrent academic responsibilities
* Three-person development team with varying technical expertise
* Limited budget for development tools and third-party services
* Development must be completed within the academic semester

### 2.4.1.2. Technical Experience

* Team members' varying familiarity with required technologies
* Learning curve for implementing AI features and mobile development
* Limited experience with database optimization and security implementation

### 2.4.2. Platform

### 2.4.2.1. Mobile

* Optimized for mid-range Android/iOS devices (2GB+ RAM).
* Offline-first features require 50MB+ storage.

### 2.4.2.2. Web

* Compatible with Chromium browsers (latest versions).
* Limited Support for Non-Chromium browsers.
* No support for legacy browsers.

### 2.4.3. Security

### 2.4.3.1. Data Protection

* Default Firebase security rules (no custom encryption for stored data.)
* Emergency profiles and guides stored locally on mobile devices (no encryption).

### 2.4.3. Other

* **Non-Critical Prototype**: Not intended for real-world emergency use. Errors in medication tracking or symptom advice are acceptable for academic evaluation.
* **Uptime**: No guaranteed uptime (free hosting services may have downtime).

## 2.5. Assumptions and dependencies

This section outlines the factors that influence the requirements specified in this Software Requirements Specification (SRS). These are not design constraints but conditions that, if altered, could impact the requirements.

### 2.5.1 Assumptions

* **Operating System Availability:** The software will operate on the latest versions of Android operating systems.
* **Internet Connectivity:** Users will have access to a stable internet connection for the features that require real-time data synchronization
* **User Device Capabilities:** Target devices will support necessary hardware features, may it be for cameras for QR code scanning, or fingerprint sensors for biometric authentication.
* **User Consent:** Users will provide informed consent for data collection and usage as per the application's privacy policy.

### 2.5.1 Dependencies

* **Technical Dependencies:**
  + **Backend:** Java Spring Boot
  + **Frontend:** ReactJS + Android Kotlin
  + **Database:** Firebase
* **Third-Party APIs:**
  + **Mapping Services:** Google
  + **AI:** DeepSeek
* **Infrastructure:**
  + **Availability of free hosting services for web deployment.**
* **Hardware:**
  + **Access to Android devices for testing the mobile app.**

# Specific Requirements

## External interface requirements

### 3.1.1. Hardware interfaces

* Mobile Devices
* **Fingerprint Sensors:** Employed for biometric authentication to enhance security.
* **Touchscreens:** Facilitate user interaction through touch inputs.
* **Cameras:** Utilized for QR code scanning.
* Web Platform
* **Desktops/Laptops:** Standard hardware configurations with keyboard and mouse inputs.

### 3.1.2. Software interfaces

* Operating Systems
  + **Android**
  + **Web Browsers** (Google Chrome, Safari, Microsoft)

### 3.1.3. Communications interfaces

* Network Protocols
  + **Wi-Fi:** For internet connectivity, enabling features like data synchronization.
  + **Cellular Data:** For users without Wi-Fi access, ensuring continuous functionality.

## Functional requirements

### 3.2.1. User Registration & Pet Profile Management (Web and Mobile)

* Customers can register, log in, and manage personal and pet profiles (e.g., breed, age, medical history, grooming preferences).
* Create and edit multiple pet profiles with customizable health alerts.
* Admins oversee user accounts, manage access permissions, and monitor platform activity.
* Profiles sync seamlessly across devices for real-time updates.

### 3.2.2. Emergency Pet Profile Card System (Mobile)

* Generates a digital emergency ID card with critical pet details (e.g., allergies, vet contacts, vaccination records).

##### Accessible offline and shareable via QR code or link during emergencies.

* Integrates alerts for expired medications or upcoming vaccinations.

### 3.2.3. Medication & Treatment Tracker (Web and Mobile)

* Schedule medication doses, vet appointments, and treatment routines with automated reminders.

##### Log administered treatments and track progress via a centralized dashboard.

* Syncs data across devices to ensure caregivers and vets stay updated.

### 3.2.4. AI-Powered Symptom Checker & Condition Encyclopedia (Web)

##### Chat-based AI virtual assistant analyzes symptoms and filters results by severity, breed, or age.

##### Searchable veterinary encyclopedia with vet-reviewed articles, visual aids, and treatment protocols.

##### Provides actionable recommendations (e.g., “Monitor at home” or “Seek urgent care”).

### 3.2.5 First-Aid Visual Guides with Offline Access (Mobile)

##### Step-by-step visual guides and video tutorials for common emergencies (e.g., choking, wounds).

##### Downloadable content for offline use in low-connectivity scenarios.

### 3.2.4. Interactive Pet Wellness Timeline (Web and Mobile)

##### Visual timeline tracks medical history, including vet visits, diagnoses, and treatments.

##### Attach lab reports, prescriptions, or images to timeline entries for comprehensive records.

##### Export timelines as PDFs for sharing with pet sitters, insurers, or veterinarians.

## Non-functional requirements

### Performance

* **Response Time:** The system should respond to user inputs within 2 seconds to ensure a seamless user experience.
* **Throughput:** The application must handle up to 1,000 concurrent users without degradation in performance.
* **Scalability:** The system should be able to scale horizontally to accommodate a 50% increase in user base without significant reconfiguration.

### Security

* **Authentication:** The application should support multi-factor authentication (MFA) to verify user identities, enhancing access control.
* **Real-Time Alerts:** Trigger alerts for suspicious activities (e.g., repeated failed login attempts, bulk data exports).
* **Account Lockout:** Temporarily lock accounts after 5 failed login attempts (30-minute lockout).
* **Role-based Access Control (RBAC):** Implement granular permissions based on user roles (e.g., pet owner, veterinarian, admin) to restrict access to sensitive features like medical history or emergency profiles.
* **Least Privilege Principle:** Users and services should only have access to the minimum data required for their function.

### Reliability

* **Data Integrity:** The system should ensure that data is accurate and consistent, with mechanisms in place to detect and correct errors.
* **Fault Tolerance:** The application must continue to function correctly even if one component fails, ensuring uninterrupted service.
* **Maintainability:** The application should be designed for easy maintenance, with clear documentation and modular architecture to facilitate updates and bug fixes.