

Software Requirements Specification

Version 1.0



FurShield

Every Paw/Wing Deserves a Shield of Love

Theme: **Pet Care**

Category: **Full-Stack Application Development**

Table of Contents

1.1 Background and Necessity for the Full-Stack Web Application

1.2 Proposed Solution

1.3 Purpose of this Document

1.4 Scope of Project

1.5 Constraints

1.6 Functional Requirements

1.7 Non-Functional Requirements

1.8 Interface Requirements

➤ **Hardware**

➤ **Software**

➤ **Database Design**

1.9 Project Deliverables

1.1

Background and Necessity for a Full-Stack Web Application



As pet ownership continues to grow, so does the necessity for a reliable and organized platform to manage pet care activities. Pet owners often face difficulties in keeping track of essential tasks such as feeding schedules, vaccinations, medical records, grooming routines, finding pet products, and vet visits. Information is usually scattered across multiple sources, that leads to confusion and inconsistent care.

A full-stack Web application is required to bring these users together through a single platform. Through this platform, pet owners can manage their pets' profiles and health updates, veterinarians can view and record treatment details, and animal shelters can showcase adoptable pets and monitor care activities. By offering centralized access to accurate and timely information, the application helps improve the quality of care, promotes responsible pet ownership, and supports efficient shelter management.

1.2

Proposed Solution

The proposed full-stack Web application titled '**FurShield**' aims to provide a unified platform that caters to the requirements of pet owners, veterinarians, and animal shelters while also offering a section for purchasing essential pet products. Pet owners can create and manage pet profiles, track health records, book appointment with vets, receive vaccination and grooming reminders, and access expert care tips.

Veterinarians can view medical histories, log treatments, and manage appointments. Animal shelters can showcase adoptable pets, update their care status, and coordinate with potential adopters and veterinary professionals.

Pet owners will also be able to browse and purchase pet care products such as food, grooming items, accessories, and health supplies directly through the platform. By combining information management, service coordination, and product access in one system the application enhances the overall pet care experience with convenience, accuracy, and efficiency.

1.3

Purpose of the Document



This document outlines the design, functional expectations, non-functional criteria, and implementation guidelines for the **'FurShield'** Web application. It serves as a reference point for developers, testers, project managers, and stakeholders involved in the creation and evaluation of the system.

1.4

Scope of the Project

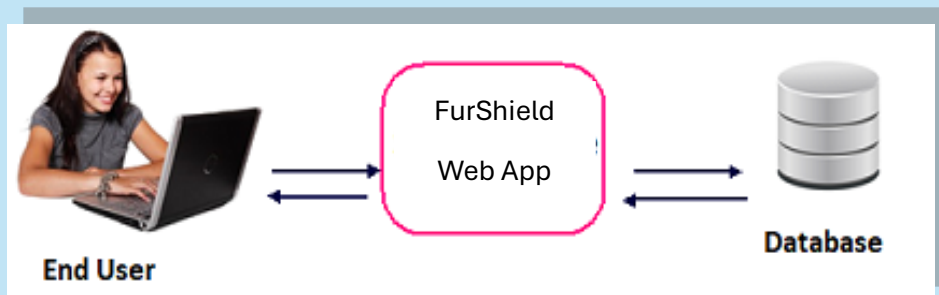
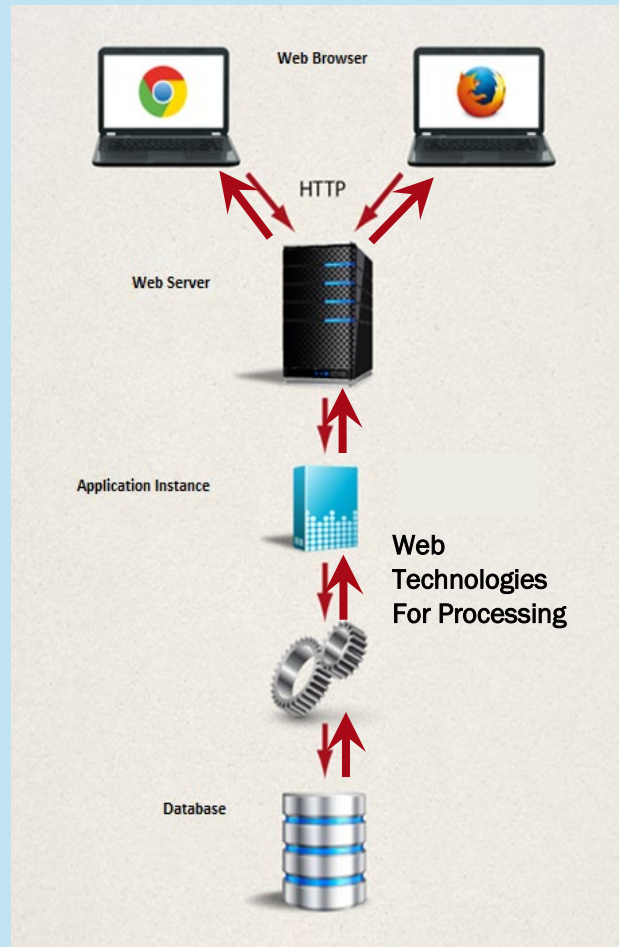
This project involves the development of a comprehensive full-stack Web-based application that simplifies and centralizes pet care management for pet owners, veterinarians, and animal shelters. The application allows users to create pet profiles, manage medical records, and book or cancel vet appointments. It provides care tips and reminders for feeding, grooming, and visits. Veterinarians can log treatments, review medical histories, and manage appointment schedules. Animal shelters can list adoptable pets, track health records, and coordinate with adopters.

The platform also includes browsing and purchasing pet food, grooming supplies, health products, and training aids. It supports secure registration, role-based access, and interactive dashboards.

The solution is responsive, intuitive, and scalable, ensuring smooth communication, better record-keeping, and convenient product access within a single platform.

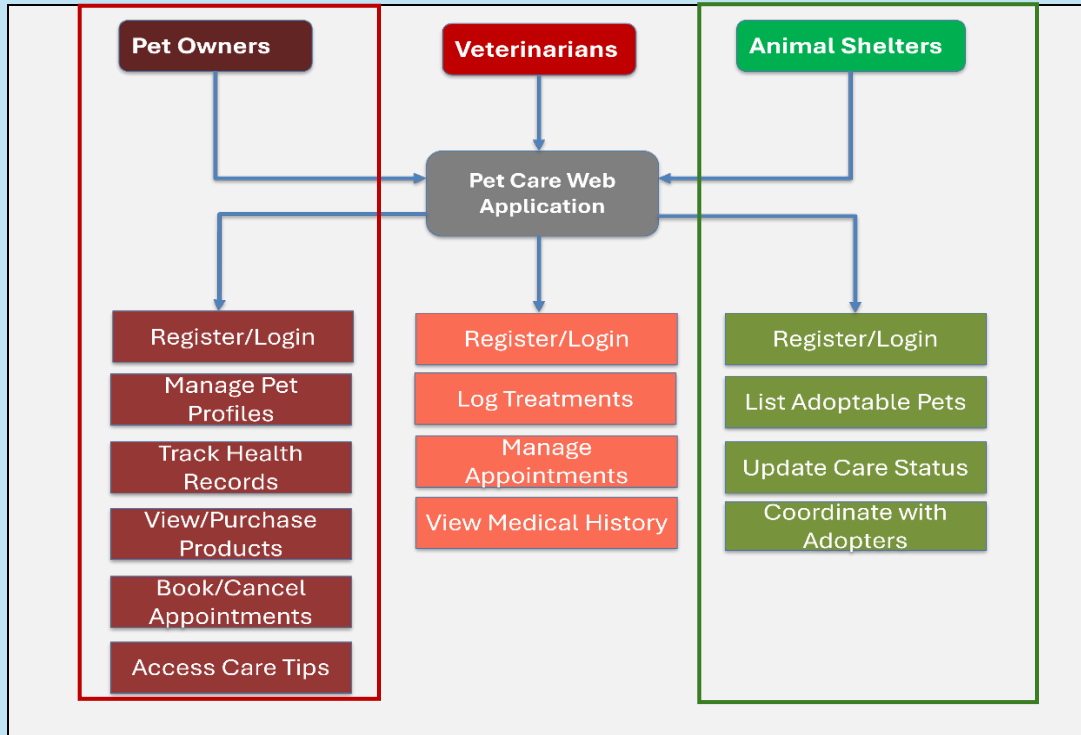


Architecture Diagram for FurShield



Flow Diagram

A sample flow diagram depicting the interaction between various entities and the application is shown here:



1.5

Constraints



The development of the **'FurShield'** Web application must adhere to several constraints to ensure its successful implementation and operation. Technically, the application must be compatible with major Web browsers and should be responsive across various devices. You could encounter constraints related to data storage, data synchronization, and backup procedures. Usage of various images and videos can be subject to licensing agreements and copyright restrictions. It is important to understand and comply with these constraints to avoid legal issues. The Web application will not have any functionality for payment gateways. Authentication of veterinarian credentials will not be a part of Web application functionality.

Functional Requirements

Application will be designed with a set of forms/pages with menus representing the choice of activities to be performed.



Pet Owner Features



User Registration and Login

- Pet owners must be able to register, log in, and securely access their dashboard.
- Name of the owner, contact number, email ID, and address must be supplied during registration.
- Registration for multiple pets by a single owner can be allowed.
- Sharing of accounts among family members can be permitted (optional).

Manage Pet Profiles

- Users can add, edit, view, or delete pet details including name, species, breed, age, and medical history. Also, they can create and manage image galleries.
- Users can manage multiple pets with tabbed interface.

Track Health Records

- Pet owners can create, update, and view vaccination dates, allergies, illnesses, and treatment history.
- Visual timeline showing vaccination history, treatments, and milestones should be displayed.
- Pet owners can scan and store vet certificates, X-rays, lab reports, and so on.
- Pet owners can store and manage insurance policy details and claims. (Only upload, storing, and viewing features will be included other features are beyond the scope of the application)

View/Purchase Products

- Users can browse categories (food, grooming, toys, and so on) with filters, view product details, and add/remove/modify products to a cart.
- Payment functionality and physical delivery of the products will not be included.

Access Care Options

- Owners can view categorized care information (feeding, hygiene, exercise, and so on) through articles, videos, and FAQs.
- AI-powered chatbot for pet care queries (optional).

Appointment Booking

- Users can request or book appointments with listed veterinarians.
- The app should auto-suggest vets based on pet's condition or location.



Veterinarian Features

Veterinarian Registration and Profile Management

- At the time of registration, vets must provide name, contact number, email ID, and address.
- Once they are logged in, vets can enhance their profiles including specialization, experience, and available time slots.

View Pet Medical History

- Vets can access health data of pets owned by users who have booked an appointment with them.

Log Treatments and Observations

- After appointments, vets can add notes on diagnosis, prescribed medication, and follow-up actions.
- Structured view with sections for symptoms, past treatments, lab results, and prescriptions should be displayed.



Manage Appointments

- Vets can view upcoming bookings, approve or reschedule appointments, and update availability.



Animal Shelter Features

Registration

- Shelters can register with shelter name, contact person name, email address, contact number, and address.

Shelter Account Access

- Shelters can log in with a verified account to manage adoption listings and pet records.

List Adoptable Pets

- Shelters can add profiles with images, age, breed, health status, and other relevant details.



Update Pet Care Status

- They can maintain and update logs for feeding, grooming, and medical attention for each shelter animal.



Coordinate with Adopters

- They can also view adopter interest forms, respond to queries, and finalize adoption status via email/push notifications.

Common Features (All Users)

Role-Based Access Control

- Users can access only those features relevant to their role.

Search, Sort, Filter

- Search pets, products, care articles, or vets based on filters like location, breed, type, category, and so on.

Responsive Design

- The application will be mobile-friendly and accessible across various devices.

Notifications

- Email or in-app alerts for vaccination due dates, appointment confirmations, new product arrivals, and so on.

Feedback and Ratings

- Users can rate veterinarians, shelters, or products and leave comments.

About Us

- Information about the team and the portal should be displayed.

Contact Us

- Display static team contact information with Google maps showing location.

Note: Boilerplate or readymade HTML template can be used, provided it is only for design aspect and not for implementing application functionality.

Generating/copying content or code from AI tools is prohibited.

Do NOT copy content or code from GPTs or other AI tools, although you are permitted to use images generated by AI tools for any visual representation purposes. It is mandatory to mention such tools used in case you add any AI generated images.

1.7

Non-Functional Requirements



There are several non-functional requirements that should be fulfilled by the application. They include:

- **Safe to use:** The application should not result in any malicious downloads or unnecessary file downloads.
- **Accessibility:** The application should have clear and legible fonts, user-interface elements, and navigation elements.
- **User-friendliness:** The application should be easy to navigate with clear menus and other elements and easy to understand.
- **Operability:** The application should be reliable and efficient.
- **Performance:** The application should demonstrate high value of performance through speed and throughput. In simple terms, the application should have minimal load time and smooth page redirection.
- **Scalability:** The application architecture and infrastructure should be designed to handle increasing user traffic, data storage, and feature expansions.
- **Security:** The application should implement adequate security measures such as authentication. For example, only registered users can access certain features.
- **Availability:** The application should be available 24/7 with minimum downtime.
- **Compatibility:** The application should be compatible with latest browsers and various devices.



These are the bare minimum expectations from the project. It is a must to implement the **FUNCTIONAL** and **NON-FUNCTIONAL** requirements given in this SRS. Once they are complete, you can use your own creativity and imagination to add more features if required.

1.8

Interface Requirements

Hardware

- Intel Core i5/i7 Processor or higher
- 8 GB RAM or higher
- Color SVGA monitor
- 500 GB Hard Disk space
- Mouse
- Keyboard

Software

IDE: Appropriate IDE as per the platform

Frontend: HTML5, CSS3, Bootstrap, ReactJS/AngularJS/Angular/TypeScript, JavaScript, jQuery, and XML

Backend:

Java SDK with Apache NetBeans or Eclipse, Jakarta EE

OR

C# with ASP.NET MVC and ASP.NET MVC Core (optional), Visual Studio IDE

OR

PHP with Laravel Framework

OR

Python with Flask or Django

OR

MongoDB, Express.js, Angular, Node.js

OR

MongoDB, Express.js, React, Node.js

Database: MySQL / SQL Server

For local hosting (optional):
XAMPP latest version

Maps and Geolocation:

- Google Maps API or OpenStreetMap for location services

Database Design

Based on the given specifications, you can define suitable entities, attributes for these entities, and identify relationships between the entities. For example, some entities along with their attributes can be identified as follows:

Owners	Pets	HealthRecords	Appointments
owner_id (PK)	pet_id (PK)	record_id (PK)	appt_id (PK)
name	owner_id (FK)	pet_id (FK)	pet_id (FK)
email (U)	name	vet_id (FK)	owner_id (FK)
password_hash	species	visit_date	vet_id (FK)
role	breed	diagnosis	appointment_time
address	age	treatment	status
	gender		

Products	Orders	Order_Items	AdoptionListings
product_id (PK)	order_id (PK)	order_item_id (PK)	listing_id (PK)
name	owner_id (FK)	order_id (FK)	shelter_id (FK)
category	total_amount	product_id (FK)	pet_name
price	status	quantity	species
description	order_date	price_each	breed
stock_quantity			age
			status

Abbreviations:

PK: Primary Key

FK: Foreign Key

U: Unique

Similarly, you can define other entities and relationships between entities and methods representing activities on the entities.

Note: These are just examples, you do not have to adhere to these structures and can design your own table structure with different columns.

Project Deliverables

You will require to design and build the project and submit it along with a complete project report that includes:

- Problem Definition
- Design Specifications
- Diagrams such as Flowcharts for various Activities, Data Flow Diagrams, and so on
- Database Design
- Test Data Used in the Project
- Project Installation Instructions (if any)
- **User Credentials for all Types of Users with Passwords**

Documentation is considered as a very important part of the project. Ensure that documentation is complete and comprehensive.

Documentation should not contain any source code.

The consolidated project will be submitted as a zip file with a ReadMe.doc file listing assumptions (if any) made at your end and SQL scripts files (.sql) containing database and table definitions.

Note: Preferably, host the working Web application on a Website and share the URL for evaluation.

Submit a video (.mp4 file) demonstrating the working of the Web application, including all features under Functional Requirements. This is MANDATORY.

Over and above the given specifications, you can apply your creativity and logic to improve the system.
