# **Software Requirements Specification (SRS)**

**Project Title**: PetStore Application

**Document Version**: 1.0

**Prepared By**: Shriya Jorrigala

**Table of Contents**

1. **Introduction**
2. 1.1 Purpose
3. 1.2 Intended Audience
4. 1.3 Scope
5. 1.4 References
6. 1.5 Overview
7. **Overall Description**

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 Constraints

2.5 Assumptions and Dependencies

1. **Specific Requirements**

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2 Functional Requirements

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software System Attributes

3.6 Other Requirements

## **1. Introduction**

### **1.1 Purpose**

This document defines the software requirements for the PetStore application—an e-commerce and pet adoption platform that allows users to browse, buy, or adopt pets and purchase pet-related supplies and services. It will guide developers, testers, project managers, and stakeholders in understanding the system’s features, functionalities, and limitations.

### **1.2 Intended Audience**

* End Users (Pet owners, Adopters)
* Admin Users (Inventory Managers, Support Staff)
* Developers & Testers
* Project Stakeholders

### **1.3 Scope**

PetStore is a web-based platform that:

* Facilitates pet browsing, adoption, and supply purchases.
* Supports user registration, login, and profile management.
* Enables secure order placement and payment.
* Provides an admin dashboard for managing inventory and orders.
* Supports mobile responsiveness and user support features. Exclusions: In-house delivery and payment processing are managed by third-party services.

### **1.4 References**

* IEEE 830-1998 Standard for SRS
* <https://petstore.octoperf.com/actions/Catalog.action>
* OWASP Security Guidelines
* UX/UI Design Best Practices

### **1.5 Overview**

This SRS outlines the application’s architecture, functional modules, interfaces, and attributes to serve as a baseline for development and validation.

## **2. Overall Description**

### **2.1 Product Perspective**

PetStore is a full-stack, responsive, browser-based application designed using a 3-tier architecture:

* **Frontend**: ReactJS/Angular
* **Backend**: Spring Boot
* **Database**: MySQL

### **2.2 Product Functions**

* User Account Management (Register/Login/Profile)
* Product & Pet Catalog
* Search and Filter
* Shopping Cart and Wishlist
* Order and Payment Processing
* Pet Adoption Workflow
* Admin Panel for Inventory & User Management
* Review, Rating & Feedback
* FAQ and Help Center

### **2.3 User Characteristics**

* End Users: Basic digital literacy required; familiar with online shopping.
* Admins: Moderate technical knowledge; experience with dashboards and analytics.

### **2.4 Constraints**

* Compatible with modern browsers (Chrome, Firefox, Edge).
* Backend must support Java EE and Spring Boot.
* Mobile responsiveness required.
* Transactions must be secured using HTTPS.

### **2.5 Assumptions and Dependencies**

* Users have internet access.
* Email services and payment gateways (Stripe, PayPal) are available.
* Hosting platform supports Java and MySQL stack.

## **3. Specific Requirements**

### **3.1 External Interface Requirements**

#### **3.1.1 User Interfaces**

* Home Page
* Product Listings and Detail Pages
* User Profile and Dashboard
* Admin Panel
* Shopping Cart and Checkout Pages
* FAQ & Support Pages

#### **3.1.2 Hardware Interfaces**

* Standard desktops, laptops, tablets, and smartphones.
* Touchscreen support on mobile devices.

#### **3.1.3 Software Interfaces**

* Frontend: ReactJS/Angular
* Backend: Spring Boot
* Database: MySQL
* External APIs: Payment Gateway (PayPal/Stripe), Email (SMTP)

#### **3.1.4 Communication Interfaces**

* RESTful APIs for frontend-backend communication.
* HTTPS for secure data transmission.

### **3.2 Functional Requirements**

#### **3.2.1 User Registration & Login**

* Register with name, email, password, contact info.
* Password encryption using hashing and salting.
* Forgot password via email verification.

#### **3.2.2 Product & Pet Catalog**

* Categorized listing of pets and supplies.
* Filters: Category, Price, Availability.
* Product details: image, price, description, stock status.

#### **3.2.3 Cart & Wishlist**

* Add/remove items to/from cart or wishlist.
* View cart summary and dynamic price calculation.

#### **3.2.4 Checkout and Payment**

* Collect shipping and billing info.
* Secure integration with payment gateway.
* Confirmation page and receipt via email.

#### **3.2.5 Order History & Tracking**

* Display order list with status updates.
* Option to cancel orders before dispatch.

#### **3.2.6 Admin Dashboard**

* CRUD operations for inventory.
* View and update order statuses.
* Sales reports and customer feedback analytics.
* Manage FAQs and user reviews.

#### **3.2.7 Pet Adoption Module**

* Application form submission.
* Admin approval process.
* Adoption status tracking.

### **3.3 Performance Requirements**

* Page load ≤ 2 seconds.
* Payment and order confirmation ≤ 5 seconds.
* Handle up to 2000 concurrent users.

### **3.4 Design Constraints**

* MySQL as DBMS.
* Browser support: Chrome, Firefox, Edge.
* MVC design pattern.
* Adhere to PCI-DSS for payment handling.

### **3.5 Software System Attributes**

#### **3.5.1 Reliability**

* System uptime ≥ 99.9%
* Daily backups of critical data

#### **3.5.2 Availability**

* 24/7 availability except for planned maintenance.

#### **3.5.3 Security**

* SSL/TLS secured.
* Password hashing and salting.
* Two-Factor Authentication (future release)

#### **3.5.4 Maintainability**

* Modular, clean codebase with documentation.
* Error logs and monitoring tools.

#### **3.5.5 Portability**

* Compatible with Windows, Linux, and MacOS servers.
* Cross-browser frontend compatibility.

### **3.6 Other Requirements**

* Multilingual support.
* User guides and admin manuals.
* Post-order feedback and review system.