# **Software Requirements Specification (SRS)**

Contents

1. Overview

1.1 Scope

2. References

3. Definitions

4. Considerations for producing a good SRS

4.1 Nature of the SRS

4.2 Environment of the SRS

4.3 Characteristics of a good SRS

4.4 Joint preparation of the SRS

4.5 SRS evolution

4.6 Prototyping

4.7 Embedding design in the SRS

4.8 Embedding project requirements in the SRS

## **1. Overview**

The Pet Store application provides a user-friendly online platform for customers to browse, search, and purchase various pet products. It supports product listing, shopping cart management, user registration, and order processing functionalities.

### **1.1 Scope**

This SRS defines the functional and non-functional requirements of the Pet Store system. It targets end-users (customers), administrators, and system integrators. The document aims to ensure a common understanding among stakeholders for a smooth development, testing, and maintenance process.

## **2. References**

* [Pet Store Application](https://petstore.octoperf.com/actions/Catalog.action)
* IEEE Standard 830-1998, "IEEE Recommended Practice for Software Requirements Specifications"
* Agile Manifesto Principles
* REST API best practices for Web Applications

## **3. Definitions**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| SRS | Software Requirements Specification |
| SKU | Stock Keeping Unit, unique identifier for a product |
| Cart | Virtual basket for holding selected products |
| Checkout | Process of purchasing the products in the cart |

## **4. Considerations for Producing a Good SRS**

### **4.1 Nature of the SRS**

The SRS will serve as the base agreement between stakeholders. It will be detailed enough to provide a clear roadmap for developers and testers, yet flexible to accommodate evolving needs.

### **4.2 Environment of the SRS**

The Pet Store system is designed as a web application accessed via internet browsers on desktop and mobile devices. The back-end uses a server-side engine to process requests, and the front-end is rendered in HTML, CSS, and JavaScript.

**Tech stack assumptions**:

* Frontend: HTML5, CSS3, JavaScript
* Backend: Java Servlet/JSP (typical for OctoPerf sample apps)
* Database: Relational (e.g., MySQL or PostgreSQL)

### **4.3 Characteristics of a Good SRS**

|  |  |
| --- | --- |
| **Characteristic** | **How It Applies to Pet Store** |
| Correct | Requirements match the site's available features like catalog browsing, user login, and cart management. |
| Unambiguous | Terms like "Add to Cart", "Checkout" will be clearly defined. |
| Complete | All pages and features will be covered, including error handling (e.g., login failures). |
| Consistent | Naming conventions and feature descriptions are used uniformly. |
| Verifiable | Requirements are testable through actions like product search, cart addition, and order confirmation. |
| Modifiable | The document structure allows easy updates if new features (like wishlist) are added later. |
| Traceable | Each feature request can be mapped to specific functionality in the app (e.g., "View Fish category" ➔ Product Listing Page). |

### **4.4 Joint Preparation of the SRS**

Stakeholders including product owners, developers, testers, and UI/UX designers should collaboratively review and finalize the SRS. Regular feedback cycles (every 2 weeks) are suggested.

### **4.5 SRS Evolution**

This SRS will evolve through iterative updates during the development life cycle, incorporating stakeholder feedback, usability studies, and user acceptance tests.

Versioning will be maintained with a version history log.

Example:

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| 1.0 | Apr 28, 2025 | Initial creation |

### **4.6 Prototyping**

Wireframes and mockups based on the existing Pet Store interface can serve as prototypes. Changes in UI can be validated using clickable prototypes before being coded.

Suggested tools: Figma, Adobe XD.

### **4.7 Embedding Design in the SRS**

Key design elements like homepage layout, product card structure, and checkout flow will be documented with annotated screenshots or diagrams.

Example elements:

* Top Navigation (Home, Sign In, Cart)
* Categories (Fish, Dogs, Reptiles, Cats, Birds)
* Search functionality
* Product detail page

### **4.8 Embedding Project Requirements in the SRS**

In addition to user requirements, system constraints and performance targets will be captured:

* Load time for any page must be < 3 seconds.
* Cart data must persist for at least 30 minutes of inactivity.
* Login sessions expire after 15 minutes of inactivity.
* Secure login with password masking.

## **5.1 Introduction (Section 1 of the SRS)**

The introduction provides an overview of the entire Software Requirements Specification for the Pet Store web application. It defines the purpose, scope, and structure of the document to facilitate a clear and common understanding among all stakeholders.

## **5.2 Overall Description (Section 2 of the SRS)**

This section provides background information, helping to make the specific requirements easier to understand. It identifies factors that influence product requirements without detailing specific requirements themselves.

* **a) Product Perspective:**  
   The Pet Store system is a self-contained web application using a typical client-server architecture. It interacts with users via a web browser and connects to a backend server for processing and a database for storage.
* **b) Product Functions:**  
   Major functions of the Pet Store system include:
  + User registration and login
  + Browsing products by category
  + Searching products by keyword
  + Adding/removing items to/from the shopping cart
  + Completing purchases through the checkout process
  + Viewing past orders
* **c) User Characteristics:**
  + General Users: Basic computer and internet browsing skills.
  + Administrators (optional scope): Familiarity with product catalog management.
  + Devices: Desktop and mobile browsers.
* **d) Constraints:**
  + Browser compatibility (Chrome, Firefox, Safari, Edge)
  + Session timeout after 15 minutes of inactivity
  + Page load times should not exceed 3 seconds
  + Compliance with basic accessibility standards (WCAG 2.1)
* **e) Assumptions and Dependencies:**
  + Users have access to a stable internet connection.
  + Backend server and database are operational.
  + No offline support is required.
  + Payment system (if added) will rely on third-party integration.
* **f) Apportioning of Requirements:**  
   Future versions may include:
  + Wishlist functionality
  + Integration with third-party pet service providers
  + Mobile app versions These features are out of scope for the current release but may be developed later.

## **5.3 Specific Requirements (Section 3 of the SRS)**

This section contains all functional and non-functional software requirements in enough detail to allow system designers and testers to proceed confidently.

### **5.3.1 Functional Requirements**

Each functional requirement is externally perceivable by users or other systems.

|  |  |
| --- | --- |
| **Requirement ID** | **Requirement Description** |
| FR-1 | The system shall allow users to create an account with a unique username and password. |
| FR-2 | The system shall authenticate users during login using their credentials. |
| FR-3 | The system shall allow users to browse products categorized by pet type. |
| FR-4 | The system shall allow users to search for products by keyword. |
| FR-5 | The system shall allow users to add products to a shopping cart. |
| FR-6 | The system shall allow users to update quantities or remove products from the cart. |
| FR-7 | The system shall allow users to complete checkout by submitting shipping information. |
| FR-8 | The system shall send order confirmation to users after a successful checkout. |
| FR-9 | The system shall allow users to view their past order history when logged in. |
| FR-10 | The system shall maintain the shopping cart for at least 30 minutes during a session. |

### **5.3.2 Non-Functional Requirements**

|  |  |
| --- | --- |
| **Requirement ID** | **Requirement Description** |
| NFR-1 | The system shall ensure that all pages load within 3 seconds under normal load. |
| NFR-2 | The system shall be available 99.5% of the time, excluding planned maintenance. |
| NFR-3 | The system shall use HTTPS protocol for secure data transmission. |
| NFR-4 | The system shall hash and store user passwords securely. |
| NFR-5 | The system shall be compatible with major web browsers. |
| NFR-6 | The system shall timeout inactive sessions after 15 minutes. |
| NFR-7 | The system shall provide accessible features in compliance with WCAG 2.1 AA. |

## **5.4 Supporting Information**

Supporting sections to make the SRS easier to use.

* **a) Table of Contents:**  
   Provided at the beginning of the document.
* **b) Index:**  
   Index of key terms and requirement IDs (Optional in simple SRS but recommended in large documents).
* **c) Appendices:**
  + A. Sample Screenshots of Catalog and Cart Pages
  + B. Future Enhancements and Out-of-Scope Features
  + C. Glossary of Terms