**Software Requirements Specification Document**

### **1. Overview**

This Software Requirements Specification (SRS) outlines the key functional and non-functional requirements of the Petstore Catalog web application, available at <https://petstore.octoperf.com/actions/Catalog.action>. It acts as a shared reference for developers, testers, and stakeholders. The document defines system behavior, expected features, and user interactions. It ensures alignment across the project lifecycle for effective development and validation.

#### **1.2 Scope**

The Petstore Catalog is a web-based application designed to allow users to browse and purchase pet-related products and services. Key functionalities include:

* Viewing categorized product listings
* Searching and filtering by keywords or category
* Viewing product details (e.g., description, price, availability)
* User login and session handling
* Processing orders, reviews, and ratings
* Payment gateway integration (if applicable)

Backend implementation, detailed security architecture, and third-party integrations are outside the scope unless otherwise stated.

### **2. References**

* IEEE Std 830-1998 - Software Requirements Specifications
* WCAG 2.1 - Web Content Accessibility Guidelines
* ISO/IEC 9126 - Software Product Quality
* Petstore Catalog URL: <https://petstore.octoperf.com/actions/Catalog.action>

### **3. Definitions**

Definitions relevant to the Petstore Catalog system include:

* **Catalog**: A structured listing of pet-related products available for users to view and purchase.
* **Shopping Cart**: A temporary holding area for selected items before checkout.
* **Checkout**: The process of reviewing the shopping cart and completing a purchase.
* **Order History**: A record of all previous purchases made by a user.

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

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

The SRS is a structured document that clearly defines both functional and non-functional system requirements. It serves as a mutual agreement between stakeholders, developers, and testers throughout the development lifecycle.

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

The SRS should be created collaboratively with input from business analysts, developers, testers, and customers. It must be easily accessible and support version control for updates and revisions.

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

An effective SRS must be:

* **Unambiguous**: Clear and open to only one interpretation
* **Complete**: Covers all functional scenarios
* **Consistent**: Free from conflicting requirements
* **Verifiable**: Testable and measurable
* **Feasible**: Realistic within constraints
* **Modifiable**: Easy to update
* **Traceable**: Each requirement has a defined source

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

The SRS should be collaboratively developed by the project team and stakeholders, ensuring that all functional and business perspectives are accurately represented.

### **4.5 SRS Evolution**

The SRS is a dynamic document that must evolve with the project. As new requirements arise, updates should be managed with proper version control.

### **4.6 Prototyping**

Prototypes can be used to validate and refine requirements early in the process. This helps stakeholders visualize the system and provide feedback before finalizing the SRS.

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

While the SRS focuses on requirements, including high-level design elements can aid stakeholder understanding. Detailed design aspects, however, should be documented separately.

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

Beyond system functionality, the SRS should also capture project-specific needs like performance benchmarks, security standards, and scalability expectations in the non-functional section.

**5. The Part of an SRS**

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

This section introduces the Petstore Catalog system. It outlines the purpose of the SRS document, the scope of the system, and the key definitions and references needed to understand the system. It establishes the context for all readers, helping stakeholders understand the intention and objectives behind the application.

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

This section introduces the Petstore Catalog system. It outlines the purpose of the SRS document, the scope of the system, and the key definitions and references needed to understand the system. It establishes the context for all readers, helping stakeholders understand the intention and objectives behind the application.

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

This section provides a broad overview of the Petstore Catalog system. It includes:

* **Product Perspective**: How the system fits within a larger e-commerce ecosystem, including integration with payment and inventory systems.
* **User Classes and Characteristics**: Details about the different users such as guests, registered users, and administrators, their roles, and their interactions with the system.
* **System Features**: A summary of functionalities including product browsing, filtering, order processing, and user management.
* **Assumptions and Dependencies**: Lists external systems (e.g., third-party payment gateways), browsers supported, and any other prerequisites for the system’s functionality.

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

This section details all the functional and non-functional requirements of the system:

* **Functional Requirements**: Include product search, user authentication, shopping cart functionality, order placement, product review and rating, and integration with payment gateways.
* **Non-Functional Requirements**:
  + **Performance**: The system should support a minimum of 500 concurrent users.
  + **Security**: Passwords must be hashed and secured; data in transit should be encrypted.
  + **Usability**: Interface should be intuitive, with consistent navigation and accessibility support per WCAG 2.1.
  + **Scalability and Reliability**: The system must scale to accommodate growing user base without service interruption.

#### **5.4 Supporting Information**

This section includes supplementary artifacts that enhance the understanding of requirements:

* **Use Case Diagrams**: Illustrate key interactions between users and the system.
* **Flowcharts**: Depict business logic and system processes.
* **Data Models**: Entity-relationship diagrams representing the data architecture.
* **UI Wireframes or Mockups**: Visual representations of user interface designs to aid stakeholders in envisioning the end product.