**SRS for JPetStore**

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, acronyms, and abbreviations
   4. Reference
   5. Overview
2. 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
   1. External interface requirements
      1. User interfaces
      2. Hardware interfaces
      3. Software interfaces
      4. Communication interfaces
   2. Functional requirement
      1. User Registration
      2. Login/Sign In
      3. Product listing and search
      4. Add products to shopping cart
      5. Order processing
      6. Payment processing
      7. Admin – Inventory Management
   3. Performance requirements
   4. Design constraints

3.5 Software system attributes

3.6 Other requirements

1. **Introduction** 
   1. Purpose

The purpose of this document is to build an online pet store for purchasing pets and pet-related products. This document helps the developers and testers throughout the development process.

* 1. Scope

JPetStore allows the user to browse and purchase pets, pet supplies, and accessories. The system has user accounts, product management, and order processing.

* 1. Definitions, acronyms, and abbreviations

SRS: Software Requirements Specification

UI: User Interface

API: Application Programming Interface

Gpay: Google Pay

* 1. Reference

PetStore Application Framework Documentation

* 1. Overview

This document describes the overall functionality of the JPetStore application, its features, and the interactions between the users and the system.

1. **Overall description**
   1. Product perspective

JPetStore is a Java based web application that is built with various payment gateways and third-party services for inventory management.

* 1. Product functions
* User Registration
* Login/Sign In
* Product listing and search
* Add products to shopping cart
* Order processing
* Payment processing
* Admin – Inventory Management
  1. User Characteristics
* Customers: Users who browse and purchase products.
* Admins: Users who are responsible for managing the store’s inventory and user accounts.

2.4 Constraints

* The system must be deployed using Java frameworks.
* The application must be compatible with mobile and desktop browsers.

2.5 Assumptions and dependencies

* A stable internet connection is required for users to access the system.
* Integration with third-party payment gateways is assumed to be operational.

1. **Specific requirements**

3.1 External interface requirements

3.1.1 User interfaces

* Home Page: Displays featured products and categories.
* Login/Registration Page: Allows users to log in or create an account.
* Product Page: Shows product details with images and prices.
* Shopping Cart Page: Displays items added with total price.
* Admin Dashboard: Provides product and order management tools

3.1.2 Hardware interfaces

* Supports devices with a minimum resolution of 720x1280 pixels.

3.1.3 Software interfaces

* Frontend: JavaScript-based (React or Angular).
* Backend: Spring Boot.
* Database: MySQL.
* Payment Gateway API: Integration with Gpay

3.1.4 Communication interfaces

* HTTPS Protocol for secure data transmission.
* RESTful API for backend and frontend communication.

3.2 Functional requirement

3.2.1 User Registration:

* The new user should be able provide his details and create new accounts.
* After successful creation, the user will be able have their account credentials.

3.2.2 Login/Sign In:

* After the account creation, users must log in using registered email and password.

3.2.3 Product Listing and Search

* Users can browse products by categories (e.g., fish, dogs).
* Users can search for products by name or keywords.

3.2.4 Add Products to shopping cart

* Users can add, update, or remove items in the cart.
* Cart must display item quantity and total price.

3.2.5 Order Processing

* Users can place orders with delivery and billing addresses.
* Order confirmation should be sent via email.

3.2.5 Payment Processing

* Payment gateway integration to accept credit/debit cards or UPI.
* Users should receive payment confirmation after a successful transaction.

3.2.5 Admin – Inventory management

* Admins can add, update, or delete products.
* The system must alert admins when stock levels are low.
* Admins can update the status of orders (e.g., pending, shipped).

3.3 Performance requirements

* The system should respond to user actions within 2 seconds.
* Product searches should return results in under 3 seconds.

3.4 Design constraints

* The backend must be implemented using Java Spring Boot.
* All communication between frontend and backend must be secured using SSL/TLS.

3. 5 Software system attributes

* The system must be easy to navigate for both customers and administrators.
* The UI should be responsive across all device sizes.

3.6 Other requirements

* The system must provide a user manual for customers and admins.
* There should be customer feedback collection after orders.