

Software Requirements Specification (SRS) Document

Pet-Shop Web Application

Table of Contents
Introduction
a. Purpose
b. Scope
c. Definitions, Acronyms, and Abbreviations
d. Overview
Overall Description
a. Product Perspective
b. Product Features
c. User Classes and Characteristics
d. Operating Environment
e. Design and Implementation Constraints
f. User Documentation
System Features
a. User Registration and Login
b. Product Search and Purchase
c. Order Management
d. Order History
e. User Logout
External Interface Requirements
a. User Interfaces
b. Hardware Interfaces
c. Software Interfaces
d. Communication Interfaces
Non-Functional Requirements
a. Performance
b. Security
c. Reliability
d. Availability
e. Scalability
f. Maintainability
Database Design
a. Entity-Relationship Diagram
b. Database Tables and Fields
Appendices
a. Glossary

Pet-Shop

1. Introduction

a. Purpose

The purpose of this document is to provide a comprehensive overview of the requirements for the Pet-Shop Web Application. It outlines the functionalities, interfaces, constraints, and specifications to be followed during the development process.

b. Scope

The scope of the Pet-Shop Web Application focuses on providing customers with an intuitive and efficient platform for browsing and purchasing pet-related products. The application aims to offer a seamless shopping experience, enabling users to explore a wide range of pet products and make purchases online.

c. Definitions, Acronyms, and Abbreviations

SRS: Software Requirements Specification

API: Application Programming Interface

UI: User Interface

ER Diagram: Entity-Relationship Diagram

Pet-Shop

d. Overview

The Pet-Shop Web Application aims to create an intuitive and convenient online platform that allows customers to explore, select, and purchase a diverse range of pet-related products. This application is designed to cater to the needs of pet owners and enthusiasts who are seeking high-quality products for their beloved animals. By providing a user-friendly interface, comprehensive product information, and seamless shopping experience, the Pet-Shop Web Application aims to deliver convenience and satisfaction to its users.

2. Overall Description

a. Product Perspective

The Pet-Shop Web Application is a standalone system that allows customers to interact with the pet shop's inventory and services. It provides a user-friendly interface for registering, browsing products, making purchases, and managing orders.

b. Product Features

The primary features of the Pet-Shop Web Application include:

User registration and login

Product search and purchase

Order management and tracking

Order history viewing

User logout

c. User Classes and Characteristics

There are two main user classes:

Customers: These are the individuals who use the application to browse products and manage their orders.

Pet-Shop

d. Operating Environment

The application will run within standard web browsers. It will be developed using Spring Boot for the backend and React for the frontend. Communication will occur through RESTful APIs, with data being stored and retrieved from a MySQL database.

e. Design and Implementation Constraints

The application should be responsive and accessible across different devices and screen sizes.

The application should handle concurrent user interactions and maintain data consistency.

f. User Documentation

The application will be accompanied by user documentation that explains how to use the features, troubleshoot common issues, and contact support.

3. System Features

a. User Registration and Login

Users can create accounts by providing their email and password.

Registered users can access the application by logging in with their credentials.

b. Product Search and Purchase

Users can search for products based on categories and keywords.

Detailed product information is displayed, and users can add products to their cart.

Users can proceed to checkout to confirm their selections.

Pet-Shop

c. Order Management

Users can view and manage their active orders, tracking order status and details.

d. Order History

Users can access their order history, viewing past purchases and order statuses.

e. User Logout

Users can log out of the application.

4. External Interface Requirements

a. User Interfaces

The user interface will be designed with an intuitive and clean layout. It will include screens for registration, login, product browsing, cart management, checkout, order history, and profile management.

b. Hardware Interfaces

The application will be compatible with standard web browsers and will not require any specialized hardware.

c. Software Interfaces

Backend: Utilizes Spring Boot for RESTful API development.

Frontend: Developed using React framework for building user interfaces.

Database: Relies on MySQL for data storage and retrieval.

d. Communication Interfaces

The frontend will communicate with the backend through RESTful API endpoints. The application will use HTTPS for secure data transmission.

Pet-Shop

5. Non-Functional Requirements

a. Performance

The application should respond to user actions promptly.

The system should handle multiple concurrent users without significant performance degradation.

b. Security

User data must be securely stored and transmitted.

Access control measures should prevent unauthorized access.

c. Reliability

The application should be stable and reliable, minimizing downtime and errors.

d. Availability

The application should be available to users with minimal planned maintenance periods.

e. Scalability

The system architecture should support easy scaling to accommodate increased user loads.

f. Maintainability

The codebase should be well-structured and documented to facilitate future maintenance and updates.

Pet-Shop

6. Database Design

a. Entity-Relationship Diagram

(TBD - Entity-Relationship Diagram illustrating the database structure)

b. Database Tables and Fields

- Customer: customer_id, customer_name, customer_password
- Users: user_id, user_name, user_type, customer_customer_id
- Orders: order_id, dispatch_date, order_date, total_cost, customer_customer_id, products_product_id
- Category: category_id, category_name
- Product: product_id, category, product_cost, description, product_name