**Experiment No 6**

**Name : Shaikh Zaid Muddaris Husain Roll No.: 23dco06 Batch : 3**

**Aim:** To develop Software Requirement Specification (SRS) document in IEEE format for the selected case study.

**Theory:**

An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time. The SRS document itself states in precise and explicit language those functions and capabilities a software system must provide, as well as states any required constraints by which the system must abide. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project Software Engineering Lab management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. SRS should address the following: a) Functionality. What is the software supposed to do? b) External interfaces. How does the software interact with people, the system„s hardware, other hardware, and other software? c) Performance. What is the speed, response time, recovery time of various software functions, etc.? d) Attributes. What are the portability, correctness, maintainability, security, etc. considerations? e) Design constraints imposed on an implementation. Are there any required standards in effect, implementation language, policies for database integrity, operating environment(s) etc.?

Procedure:

Prepare the SRS document as per the following guidelines: A sample of basic SRS Outline

1. Introduction

1.1 Purpose

1.2 Document conventions

1.3 Intended audience

1.4 Contact information/SRS team members

1.5 References

2. Overall Description

2.1 Product perspective

2.2 Product features

2.3 User classes and characteristics

2.4 Operating environment

2.5 Design and Implementation constraints

2.6 User documentation

2.7 Assumptions and dependencies

3. External Interface Requirements

3.1 User interfaces

3.2 Hardware interfaces

3.3 Software interfaces

3.4 Communication interfaces

4. System Features

4.1 System feature A

4.1.1 Description and priority

4.1.2 Action/result

4.2 System feature B

5. Other Nonfunctional Requirements

5.1 Performance requirements

5.2 Safety requirements

5.3 Security requirements

5.4 Software quality attributes

6. Other Requirements

Appendix A: Terminology/Glossary/Definitions list

Appendix B: Analysis Model

Appendix C: Issues list

**Output:**

# **Software Requirements Specification (SRS)**

# **E-Commerce Website**

# **1. Introduction**

# **1.1 Purpose**

# This SRS document outlines the requirements for the E-Commerce Website (ECW), which will provide an online platform for users to browse and purchase products. The purpose of this document is to define the functionalities, interfaces, and constraints of the system.

# **1.2 Document Conventions**

# - **Bold**: Titles and section headings

# - *Italic*: Emphasized text

# - Code Font: Code snippets or commands

# - Numbering is used to structure the document for clarity

# 

# **1.3 Intended Audience**

# This document is intended for:

# Web Developers

# System Architects

# Quality Assurance Teams

# Business Analysts

# Stakeholders involved in the development and deployment of the ECW

# **1.4 Contact Information/SRS Team Members**

# [Zaid Shaikh] - Project Manager (Email: 23dco06@aiktc.ac.in)

# [Team Member 1] - Arhaan SOlkar (Email: 22co062@aiktc.ac.in)

# [Team Member 2] - Shahid Karche (Email: 23dco02@aiktc.ac.in)

# **1.5 References**

# IEEE Standard for SRS Documentation

# E-commerce Best Practices Guidelines

# User Interface Design Guidelines (Version X)

# **2. Overall Description**

# **2.1 Product Perspective**

# The ECW is a web-based application that allows users to browse, select, and purchase products online. It integrates with payment gateways and delivery services. The platform will also include an admin interface for managing products, orders, and users.

# **2.2 Product Features**

# User registration and login

# Product search and filtering

# Shopping cart and checkout

# Payment integration (e.g., PayPal, Stripe)

# Order tracking

# Admin interface for managing inventory, orders, and users

# Customer reviews and ratings

# 

# **2.3 User Classes and Characteristics**

# Customers: Can browse products, add items to the cart, make purchases, and track orders.

# Admin: Can manage products, orders, users, and handle refunds/returns.

# Suppliers: Can manage product listings and view order statuses.

# **2.4 Operating Environment**

# Operating Systems: Windows, macOS, Linux

# Browsers: Chrome, Firefox, Safari, Edge

# Database: MySQL or PostgreSQL

# Servers: Apache or Nginx

# **2.5 Design and Implementation Constraints**

# Must support integration with multiple payment gateways

# Compliance with local and international e-commerce regulations

# Must adhere to security standards such as PCI DSS for handling payments

# **2.6 User Documentation**

# User Manual

# Admin Guide

# API Documentation

# **2.7 Assumptions and Dependencies**

# Users have basic knowledge of online shopping

# The system will require a reliable internet connection

# The project depends on third-party APIs for payment and shipping services

# **3. External Interface Requirements**

# **3.1 User Interfaces**

# Responsive web interface for browsing and shopping

# Admin dashboard for managing products, users, and orders

# Mobile-friendly interface for on-the-go shopping

# 

# **3.2 Hardware Interfaces**

# Supports standard peripherals (keyboard, mouse, touchscreen)

# **3.3 Software Interfaces**

# Integration with payment APIs (e.g., Stripe, PayPal)

# API endpoints for integrating with delivery services

# **3.4 Communication Interfaces**

# Email notifications for order confirmations and status updates

# WebSocket for real-time inventory updates

# **4. System Features**

# **4.1 System Feature A: Product Management**

# **4.1.1 Description and Priority**

# Admins can add, update, and delete products. This is a high-priority feature, as it is essential for the business.

# **4.1.2 Action/Result**

# Admins can manage product listings, including descriptions, prices, and inventory. The system will update the product catalog in real-time.

# **4.2 System Feature B: Checkout and Payment**

# **4.2.1 Description and Priority**

# Customers can review their cart, select payment methods, and complete purchases. This is a high-priority feature for the website.

# **4.2.2 Action/Result**

# Customers can select products, complete the purchase, and receive order confirmation. Payment processing will be done securely through integrated payment gateways.

# **5. Other Nonfunctional Requirements**

# **5.1 Performance Requirements**

# The system should support up to 500 concurrent users

# Response time for adding items to the cart should be less than 1 second

# **5.2 Safety Requirements**

# Daily backups to prevent data loss

# Secure handling of user and payment information

# **5.3 Security Requirements**

# Must support SSL encryption for secure transactions

# Two-factor authentication for admin access

# **5.4 Software Quality Attributes**

# Reliability: 99.9% uptime

# Scalability: The system should support growth as the user base increases

# Usability: Intuitive user interface with minimal training required

# **6. Other Requirements**

# Any additional requirements will be documented here as the project evolves.

# **Appendix A: Terminology/Glossary/Definitions List**

# ECW: E-Commerce Website

# Cart: The virtual basket where users add products before purchasing

# Payment Gateway: A service that authorizes payments

# 

# **Appendix B: Analysis Model**

# Diagrams and models showing system architecture, user interaction, and data flow.

# **Appendix C: Issues List**

# A list of known issues and their resolutions throughout development.

# **Conclusion:**

# The SRS for the E-Commerce Website provides a detailed overview of the system’s functionalities, constraints, and requirements. By following this document, the development team can ensure that the project is completed efficiently, offering a user-friendly, secure, and scalable online shopping platform.

# 