Project Title: International Expansion of E-Commerce Platform  
Purpose: Enhance an existing e-commerce platform to support international users with features like localization, multi-language, and multi-currency in phases.

**Software Requirements Specification (SRS)**

**Project Name: International Expansion of E-Commerce Platform**

**Version: 1.0**

**Author: Akshata R H**

**Date: 25-04-2025**

### Objective:

Review IEEE 830 standards and prepare an SRS template.

### Explanation:

The IEEE 830-1998 standard outlines how to prepare a high-quality Software Requirements Specification (SRS). It ensures that requirements are:

* Correct – Accurately describe the intended functionality
* Unambiguous – Only one interpretation
* Complete – All significant requirements are included
* Consistent – No conflicting requirements
* Verifiable – Requirements can be tested
* Modifiable – Organized for easy updates
* Traceable – Can be traced back to origin

### SRS Template Structure (IEEE 830 Compliant):

1. Introduction
   * Purpose
   * Scope
   * Definitions, Acronyms, Abbreviations
   * References
   * Overview of Document
2. Overall Description
   * Product Perspective
   * Product Functions
   * User Characteristics
   * Constraints
   * Assumptions and Dependencies
3. Specific Requirements
   * Functional Requirements
   * Non-functional Requirements (Performance, Security, Usability, etc.)
   * Interface Requirements (User, Hardware, Software)
   * Logical Database Requirements
4. Appendices
   * Glossary
   * Supporting Information

### **Problem Statement:**

Upgrade an existing e-commerce platform to support international expansion by adding localization, multi-language, and multi-currency support.

Model : incremental model.

#### 1. Introduction

* Purpose: Define the functional and non-functional requirements of the upgraded e-commerce system.
* Scope: Applies to enhancements that allow international customers to interact with the platform in their language and currency.
* Overview: Document presents incremental features delivered in phases.

#### 2. Overall Description

* Product Perspective: An upgrade to an existing live e-commerce platform.
* Product Functions:
  + Detect user’s region and show localized content
  + Enable payments through regional gateways
  + Display prices in user-selected currencies
  + Allow switching between multiple languages
* User Characteristics: Internet-savvy, from different countries
* Constraints:
  + Must maintain backward compatibility
  + Must comply with GDPR and PCI-DSS
* Assumptions:
  + Third-party APIs for translation and currency exchange will be reliable

#### 3. Specific Requirements

* Functional Requirements:
  + FR1: System detects user's region automatically
  + FR2: User can manually select language/currency
  + FR3: Admin can configure supported locales
  + FR4: Prices are converted using exchange rates
  + FR5: Support payment gateways like PayPal, Stripe, Razorpay
* Non-functional Requirements:
  + NFR1: Response time < 2 seconds for 90% of transactions
  + NFR2: 99.9% system uptime
  + NFR3: UI supports left-to-right and right-to-left scripts
  + NFR4: Accessibility compliance (WCAG 2.1)
* External Interfaces:
  + Google Translate API for language support
  + CurrencyLayer API for exchange rates
  + Payment SDKs

### **Techniques Used:**

* Stakeholder Brainstorming  
  Held meetings with users, admin staff, and dev team to identify features needed by international users.
* Use Case Modeling  
  Used to capture key system interactions.

### 

### **Key Use Cases**:

* Use Case 1: “Select Language”
  + Actor: User
  + Description: User selects preferred language from dropdown
* Use Case 2: “Convert Price”
  + Actor: System
  + Description: System auto-converts and displays product prices
* Use Case 3: “Configure Locale”
  + Actor: Admin
  + Description: Admin selects available currencies/languages
* Use Case 4: “Pay with Local Gateway”
  + Actor: Customer
  + Description: Customer chooses a region-specific payment method

### Requirement Categorization:

* Functional: Region detection, language selector, payment methods
* Non-Functional: Availability, performance, accessibility
* Domain-specific: ISO 4217 currency codes, multilingual SEO

Software Requirements Verification

### **Techniques Used:**

* Peer Review: Team reviewed SRS using a checklist
* Checklist Items Included:
  + Are all requirements traceable to use cases?
  + Are non-functional requirements measurable?
  + Is terminology consistent?
* Traceability Matrix:  
  Each use case was linked to at least one functional requirement.

## Software Requirements Validation

### Stakeholder Involvement:

* Admins, Dev team, and Pilot Users reviewed requirements.
* Requirements walkthrough was conducted using Zoom/Google Meet.
* Stakeholders validated that all expected features were captured.

### Changes After Validation:

* Added feature: Allow fallback to default currency/language
* Modified: Language selection to persist in user profile

### Validation Techniques:

* Stakeholder reviews
* Live walkthroughs of SRS document
* Approval through a sign-off form

### Documentation:

* A validation report was created with notes and stakeholder feedback
* All changes were versioned in Git

## SRS Documentation and Versioning

### **Finalization:**

* Final SRS was reviewed, baselined, and saved as version v1.0
* All requirements validated and verified

### **Version Control:**

* Git used to track all changes with descriptive commit messages
* Changelog maintained:
  + v0.1: Initial Draft
  + v0.5: Added use cases and non-functional requirements
  + v1.0: Finalized after verification and validation

### **Submission Package:**

* Final SRS (v1.0)
* Git repository with commit history
* Verification report
* Validation report
* Change log