

## Software Requirements Specification (SRS) Template

**Project:** Online Marketplace for Handcrafted Goods

**Version:** 1.0

**Authors:** < Instructor / Student >

**Date:** 01-10-2025

**Status:** Model Answer / Approved

### Revision history

Version	Date	Author	Change summary	Approval
1.0	01-10-2025	Instructor	SRS with diagrams embedded	

### Approvals

Role	Name	Signature / Email	Date
Course Coordinator			

### Table of Contents

1. Introduction
2. Overall description
3. External interfaces
4. System features (detailed)
5. Non-functional requirements (detailed)
6. Quality attributes & Acceptance tests
7. UML Use-Case Diagram
8. Requirements Traceability Matrix (RTM)

## 1. Introduction

### 1.1 Purpose

This SRS defines requirements for a Handmade E-commerce Storefront that connects artisans with customers, enabling browsing of artisan profiles, viewing products, managing favorites, adding to cart, and secure checkout.

### 1.2 Scope

The platform covers artisan registration, product listing, search & filters, customer authentication, cart, order placement, and admin monitoring. It excludes logistics/delivery management and offline sales.

### 1.3 Audience

Developers, QA Engineers, Admins, End Users

### 1.4 Definitions

List of acronyms: UI,API,DBMS

## 2. Overall description

### 2.1 Product perspective

The Handmade Storefront is a standalone web-based application that functions as an online marketplace for artisans and customers. It is an independent product and not a component of a larger system. The application provides two main perspectives:

- Customer Perspective: Customers can browse products, search by category, view artisan profiles, add products to cart, and complete secure purchases.
- Artisan Perspective: Artisans can create and manage profiles, upload product details, update stock, and track their product listings.

### 2.2 Major product functions (detailed)

- User registration and login (customers & artisans)
- Product addition, update, and removal by artisan
- Product browsing, search, and category-based filtering
- Artisan profile pages with product listings
- Shopping cart and secure checkout
- Order placement and tracking
- Notifications

### 2.3 User roles and characteristics (expanded)

- Customer – Browses products, adds to cart, places orders, tracks deliveries.
- Artisan – Creates and manages product listings, updates stock, views orders.
- Admin – Manages users, products, categories, and overall system monitoring.

### 2.4 Operating environment

Platform: Web-based application built on Next.js framework.

Frontend: Runs in modern web browsers.

Backend: Node.js environment.

Database: MySQL

### 2.5 Constraints

- Must support secure payments.
- Must be accessible.
- Must handle at least 100 concurrent users.

## 3. External interface requirements

### 3.1 User interfaces

Responsive web app

### 3.2 Software interfaces

Payment Gateway API, Email/Notification service.

### 3.3 Communications

HTTPS (TLS 1.2+), REST APIs.

<< Make sure overall there are at least 15 FRs for overall project, 5 NFRs, 2 security objectives and 5 Security requirements>>

## 4. System features (detailed)

Each requirement below includes acceptance criteria and a reference test case. IDs follow HME-F-###.

Req ID	Requirement	Type	Priority	Source/Stakeholder	Acceptance criteria / Test case ref	Comments / Dependencies
HME-F-001	System shall allow user registration/login (email & password)	Functional	High	Customer/User	User can register/login with valid email & password; TC-F001	Depends on database setup, email validation
HME-F-002	System shall support OAuth login	Functional	Medium	User/Customer	User can login via Google OAuth; TC-F002	Requires Google API integration
HME-F-003	System shall allow artisans to create/update profiles.	Functional	High	Artisan	Artisans can create/update profile details; TC-F003	Requires user login (HME-F-001)
HME-F-004	System shall allow artisans to list products with name, description, price, images	Functional	High	Artisan	Artisan can add/edit products; TC-F004	Image upload & storage dependencies
HME-F-005	System shall allow customers to browse products by category	Functional	High	Customer/User	Customer can view products filtered by category; TC-F005	Depends on product listing (HME-F-004)

HME-F-006	System shall allow customers to view artisan profiles	Functional	Medium	Customer/User	Customer can view artisan profile details; TC-F006	Requires artisan profiles to exist (HME-F-003)
HME-F-007	System shall allow adding/removing items to favorites	Functional	Medium	Customer/User	Customer can add/remove items to/from favorites; TC-F007	Depends on product listing (HME-F-004)
HME-F-008	System shall allow adding/removing items to cart	Functional	High	Customer/User	Customer can add/remove products to/from cart; TC-F008	Depends on product listing (HME-F-004)
HME-F-009	System shall allow checkout with order confirmation	Functional	High	Customer/User	Customer can complete checkout and receive confirmation; TC-F009	Depends on cart functionality (HME-F-008)
HME-F-010	System shall integrate with payment gateway for secure payments	Functional	High	Customer/User	Payment processed successfully; TC-F010	Requires payment gateway integration
HME-F-011	System shall allow customers to rate & review artisans/products	Functional	Medium	Customer/User	Customer can submit/view ratings & reviews; TC-F011	Depends on order completion (HME-F-009)

HME-F-012	System shall allow admin approval of artisan accounts	Functional	High	Admin	Admin can approve/reject artisan accounts ;TC-F012	Depends on artisan registration (HME-F-003)
HME-F-013	System shall display product stock availability	Functional	Medium	Customer/User	Product availability shown accurately; TC-F013	Depends on product inventory data
HME-F-014	System shall allow search by artisan/product name	Functional	Medium	Customer/User	Search returns correct products/artisans; TC-F014	Requires database indexing/search integration
HME-F-015	System shall provide notifications for order updates	Functional	Medium	Customer/User	Notifications sent on order status change; TC-F015	Depends on order & checkout flow (HME-F-009)

#### 4.1 Maintenance & Monitoring

The system will be regularly maintained to fix bugs, update features, ensure security, and keep the website compatible with modern browsers and devices. Monitoring will track server performance, uptime, order processing, and errors, with alerts for critical issues to ensure smooth operation and a reliable user experience.

#### 5. Non-functional requirements (detailed)

NFRs below are measurable and tied to test plans. IDs HME-NF-###.

Req ID	Requirement	Category	Priority	Acceptance criteria / Measurement
HME-NF-001	90% of page loads shall complete	Performance	High	Measure page load times; 90% of

	within 3 seconds under normal load			requests should be ≤3s under normal load.
HME-NF-002	System shall provide 99.5% uptime monthly	Reliability	High	Monthly uptime logs should show ≥99.5% availability.
HME-NF-003	System shall encrypt all data in transit (TLS 1.2+)	Security	High	All network traffic should use TLS 1.2 or higher; verified via security scan.
HME-NF-004	System shall support up to 500 concurrent users	Scalability	Medium	Load test with 500 simultaneous users should not cause failures or degradation.
HME-NF-005	System shall meet WCAG 2.1 AA accessibility	Usability/Accessibility	Medium	Accessibility audit confirms compliance with WCAG 2.1 AA standards.

## 5.1. Security

### 5.1.1 Security Objectives

Protect customer personal & payment data.

Prevent unauthorized access to artisan accounts.

### 5.1.2 Security Requirements

Req ID	Requirement (shall...)	Type	Priority	Acceptance criteria / Test case ref
HME-SR-001	All API endpoints must use TLS 1.2+	Security	High	All API requests are over HTTPS using TLS 1.2+;

				verified via security scan; TC-SR-001
HME-SR-002	System shall hash passwords using bcrypt	Security	High	Passwords stored in DB are hashed with bcrypt; TC-SR-002
HME-SR-003	System shall support role-based access control	Security	High	Users can access only authorized resources based on role; TC-SR-003
HME-SR-004	System shall implement JWT-based authentication	Security	High	Authentication uses JWT tokens; expired/invalid tokens are rejected; TC-SR-004
HME-SR-005	System shall lock account after 5 failed login attempts	Security	High	Account is locked after 5 consecutive failed logins; unlock requires admin action or timeout; TC-SR-005

## 6. Quality attributes & Acceptance tests

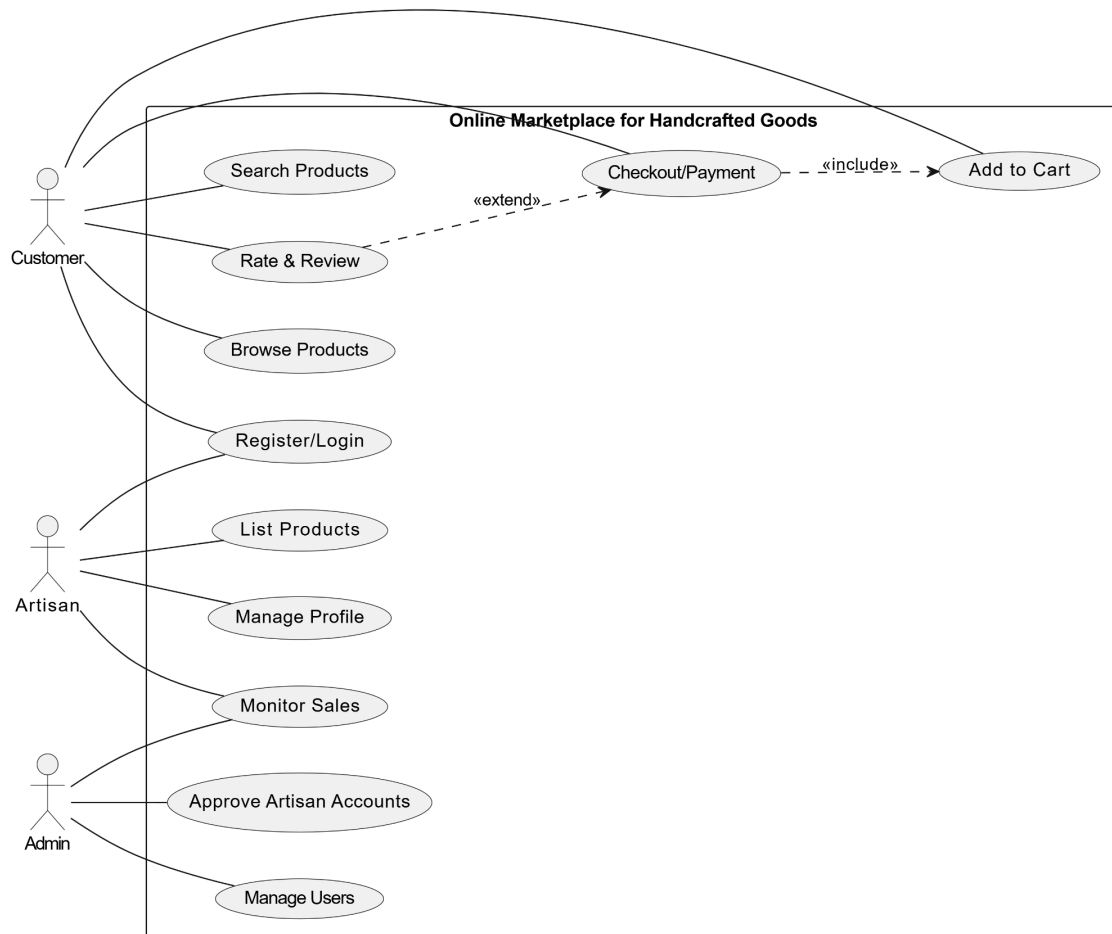
Exit criteria for acceptance: All high-priority functional requirements implemented and verified, no critical non-functional requirement failures, and all test cases in the RTM passed.

Acceptance test suites: User Registration/Login, Product Management, Cart & Checkout, Payment, Ratings & Reviews, Performance, Security, and Accessibility tests.



## 7. System models and diagrams

### 7.1 UML Use-Case diagram



## 8. Requirements Traceability Matrix (RTM)

Req ID	Requirement short	Section ref / Design Spec	Module	Test case(s)	Status (N/P/A)	Comments

HME-F-001	System shall allow user registration/login	Auth Module, Sec 3.1	Authentication	TC-01, TC-02	P	Basic login works, OAuth pending
HME-F-002	System shall support OAuth login	Sec 3.1 (Auth Spec)	Authentication	TC-03	N	To be implemented in next sprint
HME-F-004	System shall allow artisans to list products	Sec 3.3 (Product Spec)	Product Catalog	TC-05, TC-06	A	Verified successfully
HME-F-009	System shall allow checkout with order confirmation	Sec 3.5 (Checkout Spec)	Checkout/Payment	TC-10, TC-11	P	Payment integration WIP
HME-F-011	System shall allow customers to rate & review	Sec 3.6 (Review Spec)	Review/Feedback	TC-13	N	Pending after checkout completion
HME-NF-001	90% of page loads within 3s	Sec 4 (Performance)	Performance Layer	TC-20	A	Passed load testing
HME-SR-002	System shall hash passwords using bcrypt	Sec 5.2 (Security Spec)	Security	TC-25	A	Verified by security tests