Software Requirements Specification

for

E-commerce web application

Version 1.0 approved

Prepared by Jacob Roberts, Ethan Mitchell, Jacob Hollis, Ashanti Cocroft, and Matthew Rester

Intro to SE Group 6

28 August 2024

1. Introduction

1.1 Purpose

Introduction to Software Engineering Group 6 is creating the first revision of a web application for e-commerce. This document is the Software Requirement Specification (SRS) for the e-commerce web application. It describes all of the features of the website.

The purpose of the SRS is to provide detailed requirements for the e-commerce web site. The website will support three user roles: Buyer, Seller, and Admin. Each user role will allow for different actions to be completed, and all users will be able to login.

Buyer actions:

1. Search products
2. Compare products
3. Buy products
4. Return products

Seller actions:

1. Add products
2. Sell products
3. Receive payments

Admin action:

1. Approve/block new user accounts
2. Oversee user actions

1.2 Document Conventions

Requirement Id takes the form of *REQ-UseCase-#*.

1.3 Intended Audience and Reading

The intended audience of this document is Dr. Charan Gudla and Lab Instructor Abhilash Kanduri. As well as the members of Group 6. When reading this document it is recommended to start with the overview sections and then proceed to sections pertinent to the reader.

1.4 Product Scope

The e-commerce website sells books adapted to film or television. It will allow sellers to add and remove stock. It will allow customers to search for content by genre, release date, etc. By serving this specific niche we hope to attract readers who would like to watch what they have read and watchers who would like to read the original inspiration.

2. Overall Description

2.1 Product Perspective

The e-commerce platform described in this SRS is a standalone product designed for buyers, sellers, and administrators. It offers essential features like user authentication, product management, and transaction processing. Though it can work on its own, it can also be integrated into larger systems, interacting smoothly with inventory management, CRM tools, and payment gateways through clear interfaces.

2.2 Product Functions

The e-commerce platform needs to handle some essential tasks based on different user roles. Users, whether they're Buyers, Sellers, or Admins, should be able to log in and out easily. Buyers should be able to search for products, compare different options, buy items, and return them if needed. Sellers should be able to list their products, manage sales, and get paid. Admins will have the ability to approve or block user accounts and product listings and keep an eye on what everyone's doing to make sure everything runs smoothly. All these functions are linked, with each role doing their part to keep the platform working as it should.

2.3 User Classes and Characteristics

- Admin
  - Approve/block new user accounts and products
  - Oversee different user actions
- Seller (Most important user class)
  - Add new products, sell and receive payments for their available products
- Buyer
  - Able to search, compare, buy and return different types of products from different sellers

2.4 Operating Environment

The e-commerce will operate and run in a python-based environment. The platform will be developed and running using python while utilizing the framework Django for web development and for the database SQLite.

2.5 Design and Implementation Constraints

Design: Language Python

- Regulatory Policies: The platform must comply with data protection laws like GDPR or CCPA, especially when handling user data and transactions.

- Security Considerations: Developers will need to implement strong security features, such as SSL encryption, secure payment methods, and user authentication to protect user information and prevent security breaches.

- Hardware Limitations: The platform must be optimized to run smoothly on different devices, including those with limited processing power and memory, to ensure a good user experience.

- Integration with Other Systems: The platform may need to connect with existing systems like inventory management, and payment gateways, which might require sticking to specific APIs and protocols, limiting some of the flexibility in how things are built.

- Technology Stack: The choice of programming languages, frameworks, and databases might be limited by what the development team knows best or what the customer's current setup supports, which could influence the overall approach.

- Design Conventions and Coding Standards: Developers will need to follow certain design and coding standards, especially if the customer’s team will be maintaining the software afterward. This includes the following best practices for coding, documentation, and version control.

- Parallel Operations**:** The platform needs to handle multiple users doing things at the same time, so developers will have to make sure it can manage parallel processes efficiently to avoid slowdowns and ensure it can scale as needed.

3. System Features

3.1 Login

3.1.1 Description and Priority

The login feature allows people to use the website as a registered user. This is a high priority feature.

3.1.2 Stimulus/Response Sequences

1. User selects login
2. System shows fields for email and password
3. User fills in fields
4. System validates user as registered user

3.1.3 Functional Requirements

Requirement ID	Requirement
REQ-Login-1	The website shall have a login page.
REQ-Login-2	The login page shall have an email field.

REQ-Login-3	The login page shall have a password field.
REQ-Login-4	The login page shall have a submit button.
REQ-Login-5	If the email and password combination does not exist as a user account, inform the user.

3.2 Registration

3.2.1 Description and Priority

The registration feature allows users without an account to create an account. This is a high priority feature.

3.2.2 Stimulus/Response Sequences

1. User selects register
2. System shows fields for email and password
3. User fills in fields
4. System creates account with given email and password combination

3.2.3 Functional Requirements

Requirement ID	Requirement
REQ-Register-1	The website shall have a register page.
REQ- Register -2	The register page shall have an email field.
REQ- Register -3	The register page shall have a password field.
REQ- Register -4	The register page shall have a submit button.
REQ- Register -5	If the email is already used, inform the user.

3.3 Search products

3.3.1 Description and Priority

The search products feature allows buyers to explore available products by using keywords. This is a medium priority feature.

3.3.2 Stimulus/Response Sequences

1. User enters search phrase in field.
2. System displays page with matching products.

3.3.3 Functional Requirements

Requirement ID	Requirement
REQ-Search-1	The website shall have a search results page.
REQ-Search -2	The website shall have a search bar on top of each page.
REQ-Search -3	The website shall have a submit search button on top of each page.

REQ-Search -4	The search page shall display products with names that match the search phrase.
REQ-Search -5	If a product with that search phrase does not exist, inform the user.

3.4 Compare products

3.4.1 Description and Priority

The compare products feature allows buyers to look at products similar to the one they are already looking at and compare different features such as the price, description, and rating. This is a low priority feature.

3.4.2 Stimulus/Response Sequences

1. User selects product.
2. System shows product details.
3. User scrolls to see similar products.
4. System shows products of a similar type.

3.4.3 Functional Requirements

Requirement ID	Requirement
REQ-Compare-1	The website shall have a page for each product.
REQ-Compare-2	Each product page shall have a compare products section.
REQ-Compare-3	The compare products section shall have up to three similar products.
REQ-Compare-4	Each similar product should show the title, seller, price, and rating.
REQ-Compare-5	If a similar product does not exist, inform the user.

3.5 Buy products

3.5.1 Description and Priority

The buy products feature allows buyers to give their payment and shipping information to the seller of a product. This is a medium priority feature.

3.5.2 Stimulus/Response Sequences

1. User selects product.
2. System shows product details.
3. User selects buy product.
4. System adds product to checkout cart.
5. User selects checkout page.
6. System shows page with payment and shipment fields.
7. User fills in fields.
8. System shows confirmation page.

3.5.3 Functional Requirements

Requirement ID	Requirement

REQ-Buy-1	The website shall have a checkout page.
REQ-Buy-2	Each product page shall have a buy product button.
REQ-Buy-3	The buy product button shall add a product to the checkout cart.
REQ-Buy-4	The checkout page shall have fields for payment information.
REQ-Buy-5	The checkout page shall have fields for shipping information.
REQ-Buy-6	The checkout page shall have a checkout button.

3.6 Return products

3.6.1 Description and Priority

The return products feature allows buyers to return items if they have any issues with their received product.

3.6.2 Stimulus/Response Sequences

1. User selects product from orders.
2. System shows order details.
3. User selects return product.
4. System checks how long a product has been received.
5. If the product received exceeds a certain amount of time, then the request will be denied.
6. If a product is eligible for return, provide instructions for return delivery.
7. Once the product has been received, a refund and confirmation will be issued.

3.6.3 Functional Requirements

Requirement ID	Requirement
REQ-Return-1	The website shall have an orders page.
REQ-Return-2	Each order will be able to check if it is returnable.
REQ-Return-3	The return product button shall provide instructions on how to return the product.
REQ-Return-4	The customer will be issued a refund and confirmation upon verification of the returned product.

3.7 Add products

3.7.1 Description and Priority

The add products feature allows sellers to add items they want to sell.

3.7.2 Stimulus/Response Sequences

1. Seller selects add products.
2. System shows product fill out sheet.
3. Seller fills out sheet.
4. System adds product to database.
5. System displays product as in stock.

3.7.3 Functional Requirements

Requirement ID	Requirement
REQ-Add-1	The website shall have an add products page.
REQ-Add-2	Each seller will be able to add more products.

3.8 Approve/Block user accounts

3.8.1 Description and Priority

The approve/block feature allows admins to block suspicious user accounts and approve user accounts to be upgraded to seller or admin status.

3.8.2 Stimulus/Response Sequences

1. Admin finds account.
2. Admin blocks or upgrades account.
3. System updates user privileges.

3.8.3 Functional Requirements

Requirement ID	Requirement
REQ-Approve/Block-1	The website shall have an account list accessible by admins.
REQ-Approve/Block-2	Each account will have a block and upgrade button.

3.9 Oversee user actions

3.8.4 Description and Priority

The oversee feature will allow an admin to view user's and seller’s order history and seller’s product inventory.

3.8.5 Stimulus/Response Sequences

1. Admin selects oversee.
2. System shows user and seller list.
3. Admin selects user or seller.
4. System shows order history and/or inventory.

3.8.6 Functional Requirements

Requirement ID	Requirement

REQ-Oversee-1	The website shall have an oversee page that shows all users.
REQ-Oversee-2	Each user will have a list of order history and/or inventory

4. Other Nonfunctional Requirements

4.1 Performance Requirements

Given that many users may access this site simultaneously, the platform must have the ability to accommodate many concurrent users. The performance of the website should not substantially reduce in response to concurrent users. Also, the timing of user interaction needs to be accurately tracked so that features work as intended. For example, if two users place the same item in their cart while there is only one left in inventory, then whichever buyer completes the purchase first should be able to proceed, while the other is accurately denied access due to empty inventory.

The Django framework includes built in tools to evaluate the performance of code. For instance, one tool displays the number of SQL queries the page is generating, alongside the amount of time each one takes. It also displays the number of queries that are similar or identical. These tools can be used by developers to rigorously test the operation of the website so that all demands are adequately met.

4.2 Security Requirements

The website must be made in a secure way that adequately protects the private information of all users. To create an account, users have to provide certain pieces of information such as an email address. This information should not readily be available to other user accounts. Also, all buyers and sellers have to provide information to participate in financial transactions. Ensuring that this information is kept safe is a top priority for the operation of this website.

Therefore, access to read/write any information from/to a user's account is restricted to only that user. Only admin accounts are about to view and oversee certain actions committed by other user accounts. The ability to read product information is in no way restricted, but the ability to write to product information is exclusive to the seller account that posted the product.

The Django framework is used, which includes many safety features, including protection against cross site scripting, cross site request forgery, SQL injection, and clickjacking.

4.3 Software Quality Attributes

The website should be available to users at any time of day and should be automated to work without the presence of workers. The website should work reliably and accurately regardless to the number of concurrent users. When information is changed, such as inventory of products, this information update should occur quickly and be immediately reflected to any viewer of the website without significant delay. Also, the UI should allow users to interact with the website easily without any steep learning curves. Basic operation of the website should be easy for users so that their experience is efficient and positive.

5. Other Requirements

The Website will pull from a SQL relational Database. The database will use tables and indexes to connect all the data together. The database will be splitting larger databases into smaller ones, to improve the performance of the website. The website will also include multilingual, local currency, local tax laws, and international shipping options, so the website can be used from any country.

Appendix A: Glossary

Django: Software website framework that works with Python
Python: General-purpose, high-level coding language

Acronym	**Definition**
API	Application Programming Interface
CCPA	California Consumer Privacy Act
CRM	Customer Relationship Management
GDPR	General Data Protection Regulation
REQ	Required
SSL	Secure Sockets Layer
SQL	Structured Query Language

Term	**Definition**
Admin(administrator)	Someone who is responsible for the maintenance of the website
Buyer	Someone who uses the website to buy something or in-search of something
Database	A collection of data
Developer	Someone who is a part of building the website
Django	Software website framework that works with python
Field	Blank space where user can enter information
Framework	A foundation for programming applications
Memory	A capacity of stored data
Priority Feature	Aspect of program that is wanted over other non-priority features
Processing Power	Computer’s ability to process information
Programming Language	Language used to write a program
Python	General-purpose high-level coding language
Seller	Someone who uses the website to sell a product on the website
System	Program/Website application
User	Someone who uses the website like Admins, Buyers, and Sellers.

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>