Software Requirements Specification (SRS)

Project Title:

Storify – Smart E-Commerce Platform

Document Title:

Software Requirements Specification (SRS)

Course Name:

CSAI 203 – Introduction to Software Engineering

Instructor:

Dr. Mohamed Sami Rakha

Team Number:

Team #28

Team Members:

- Ahmed Tamer — ID:202401457
- Omar Ahmed — ID:202400354
- Samaa Khaled — ID:202401280

Table of Contents:

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 5

1.5 Overview 5

2. Overall Description 5

2.1 Product Perspective 5

2.2 Product Functions 6

2.3 User Classes and Characteristics 7

2.4 Operating Environment 7

2.5 Design and Implementation Constraints 8

2.6 User Documentation 8

2.7 Assumptions and Dependencies 8

3. Specific Requirements 9

3.1 Functional Requirements 9

3.2 Use Case Model 9

3.2.1 Use Case Diagrams 10

3.2.2 Use Case Descriptions 11

3.3 Domain Model 12

3.3.1 Conceptual Class Diagram 12

3.3.2 Class Descriptions 13

3.4 Non-Functional Requirements 13

3.5 External Interface Requirements 14

3.5.1 User Interface 14

3.5.2 Hardware Interface 14

3.5.3 Software Interface 14

3.5.4 Communication Interface 14

4. Appendices 15

4.1 Appendix A: Data Dictionary 15

4.2 Appendix B: Glossary 15

1. Introduction:

1.1 Purpose

The purpose of this document is to specify, describe, and define the functional and non-functional requirements of the **Storify – Smart E-Commerce Platform**.
Storify aims to provide an intelligent, secure, and user-friendly e-commerce environment that connects **customers**, **sellers**, and the **admin** in one integrated system.

The platform allows **customers** to browse and purchase products, **sellers** to manage and sell their items, and the **admin** to monitor and control the entire platform's operations.
This document serves as a communication bridge between developers, stakeholders, and the client, ensuring that all software requirements are clearly understood and properly documented before the design and implementation phases.

1.2 Scope

Storify is an online e-commerce platform designed to simplify and enhance the online shopping experience by connecting three main types of users — customers, sellers, and the admin — through one smart, secure, and scalable system.

The system provides the following main functionalities:

For Customers:

- Register and log in to their accounts.
- Browse and search for products using filters such as category, price, and rating.
- Add products to the shopping cart and place orders.
- Make secure online payments.
- View purchase history and order status.
- Review and rate purchased products.

For Sellers:

- Create and manage store profiles.

- Add, edit, or remove products with details (name, description, price, stock).
- View customer orders and manage deliveries.
- Receive notifications about low-stock items or customer feedback.

For Admin:

- Monitor system activity and user behavior.
- Manage and verify sellers' accounts and their product listings.
- Handle user reports, complaints, and system maintenance.
- Generate analytical reports about sales and performance.

1.3 Definitions, Acronyms, and Abbreviations:

Term / Acronym	**Definition**
Admin	The system administrator manages the entire platform and oversees users and sellers.
Customer	The user who browses, buys, and reviews products.
Seller	The user who owns a store and manages products and orders.
Product	Any item listed by a seller is available for purchase.
Cart	A virtual basket where customers can temporarily store products before checkout.
Order	A confirmed request from a customer to purchase specific products.
Payment Gateway	A secure service that processes online payments.
Notification	A system message sent to inform users about updates, stock changes, or order status.
Authenticator	The module is responsible for login, signup, and user verification.
Checker	A background process that monitors stock levels and sends low-stock alerts to sellers.

1.4 References:

- IEEE Std 830-1998 — IEEE Recommended Practice for Software Requirements Specifications.

1.5 Overview:

The remainder of this document provides a comprehensive description of the Storify System, detailing both its functional and non-functional requirements.
Section 2 outlines the overall description of the system, including its perspective, functions, and user characteristics.
Section 3 provides the specific requirements of the system, including functional requirements, detailed Use Case Diagrams and descriptions, and the Conceptual Class Diagram with their relationships.
Non-functional and interface requirements are also defined to ensure performance, usability, and system constraints are met.
Finally, the appendices include supporting materials such as the Data Dictionary and Glossary to facilitate understanding and traceability of the requirements throughout the system lifecycle.

2. Overall Description:

2.1 Product Perspective

Storify is a web-based e-commerce system that connects three main stakeholders: customers, sellers, and administrators.
The system serves as a platform for online product management, purchase, and communication between users.
It allows customers to browse, search, and purchase products; sellers to add, manage, and track their products; and administrators to monitor system activities, manage users, and handle reports or issues.

Storify follows a modular architecture divided into several key subsystems:

- Authentication Module: Handles of login, registration, and account management for all user types.

- Product Management Module: Enables sellers to add, edit, and remove products, and allows customers to browse available items.
- Order & Payment Module: Supports cart management, order placement, and payment processing.
- Notification & Monitoring Module: Alerts sellers when product stock is low and notifies customers about order updates.
- Customer Support Module: Manages communication between users and support teams through queries and responses.
- Admin Control Module: Gives administrators authority to verify products, approve sellers, and respond to system-wide reports.

The system is designed to be scalable, secure, and user-friendly, with a structured interaction between different roles using predefined interfaces and workflows.

2.2 Product Functions

The Storify System provides a set of core functions categorized by user type:

1. Customer Functions

- Register and Login: Customers can create an account or log in securely using their credentials.
- Browse Products: View available products, search by category, and filter by price, brand, or rating.
- View Product Details: See detailed descriptions, prices, and seller information.
- Add to Cart / Remove from Cart: Manage selected items before purchase.
- Make Payment: Choose a payment method and complete the purchase securely.
- Track Orders: View order status, shipment, and delivery details.
- Submit Queries or Complaints: Contact customer support for help or inquiries.

2. Seller Functions

- Register and Login: Sellers create accounts and verify their profiles.
- Add Products: Upload product details including name, description, price, and stock quantity.
- Edit or Remove Products: Update or delete products as needed.
- View Orders: Check customer purchases and manage deliveries.

- Respond to Notifications: Receive alerts when stock is low (from the checker module).
- Reply to Customer Queries: Respond to customer inquiries or complaints received via admin or directly.

3. Administrator Functions

- User Management: Verify, activate, or deactivate seller and customer accounts.
- Product Verification: Approve or reject new products before they appear on the store.
- Monitor System Activity: Track transactions, logs, and system usage.
- Respond to Reports: Handle issues reported by customers or sellers.
- Manage Notifications: Control system alerts and announcements.

4. System Functions

- Notification System: Sends real-time alerts to users for order updates or low stock.
- Checker Module: Monitors product quantities and alerts sellers automatically.
- Database Management: Stores and retrieves all user, product, and transaction data securely.

2.3 User Classes and Characteristics:

User Type	Description	Experience Level	Main Responsibilities
Customer	End-user who visits the platform to browse and buy products.	Basic computer and internet skills.	Browse products, make purchases, and submit feedback.
Seller	Business owner or individual who uploads and manages products.	Intermediate knowledge of online selling tools.	Managing product listings, checking notifications, handling customer orders
Administrator	Site owner and system supervisor.	Technical and managerial	Managing users, verifying products,

		experience.	and monitoring system operations.

2.4 Operating Environment:

- Frontend: Web browsers (Chrome, Edge, Firefox, Safari).
- Backend: Flask
- Operating Systems: Compatible with Windows
- Network: Requires stable internet connection for all operations.

2.5 Design and Implementation Constraints:

- The system must comply with security standards (HTTPS, password encryption).
- The database must handle multiple concurrent users efficiently.
- Systems should follow responsive design for various screen sizes.
- All components must integrate seamlessly within a modular architecture.

2.5 Design and Implementation Constraints:

- The system must comply with security standards (HTTPS, password encryption).
- The database must handle multiple concurrent users efficiently.
- Systems should follow responsive design for various screen sizes.
- All components must integrate seamlessly within a modular architecture.

2.6 User Documentation:

- A user manual will be provided explaining how to register, log in, add products, and make purchases.
- Help pages and FAQs will be integrated into the platform.
- Technical documentation for developers will describe APIs and database schema.

2.7 Assumptions and Dependencies:

- Users must have access to a stable internet connection.
- All users should provide valid registration information.
- Payment gateways (e.g., PayPal, Stripe) must be available and operational.

3.1 Functional Requirements

The functional requirements of Storify are categorized by user type and define the core features the system must provide.

1. Customer Functional Requirements:
- Register, login, and logout securely.
- Browse and search for products using filters.
- Add items to the cart and place orders.
- Make secure online payments.
- View order history and order details.
- Review and rate purchased products.

2. Seller Functional Requirements:
- Register, login, and manage store profile.
- Add, update, or remove products.
- View customer orders and respond to return queries.
- Receive notifications about low stock or customer feedback.

3. Administrator Functional Requirements:
- Manage users (view, block/unblock, or delete).
- Verify and approve seller accounts and products.
- View system reports and respond to user queries.
- Handle complaints and maintain system activity.

4. System Functional Requirements:
- Manage notifications and alerts.
- Monitor stock using the background checker module.

- Handle all database transactions securely and efficiently.

3.2 Use Case Model

3.2.1 Use Case Diagram

The following diagram illustrates the main actors and their interactions within the Storify System.

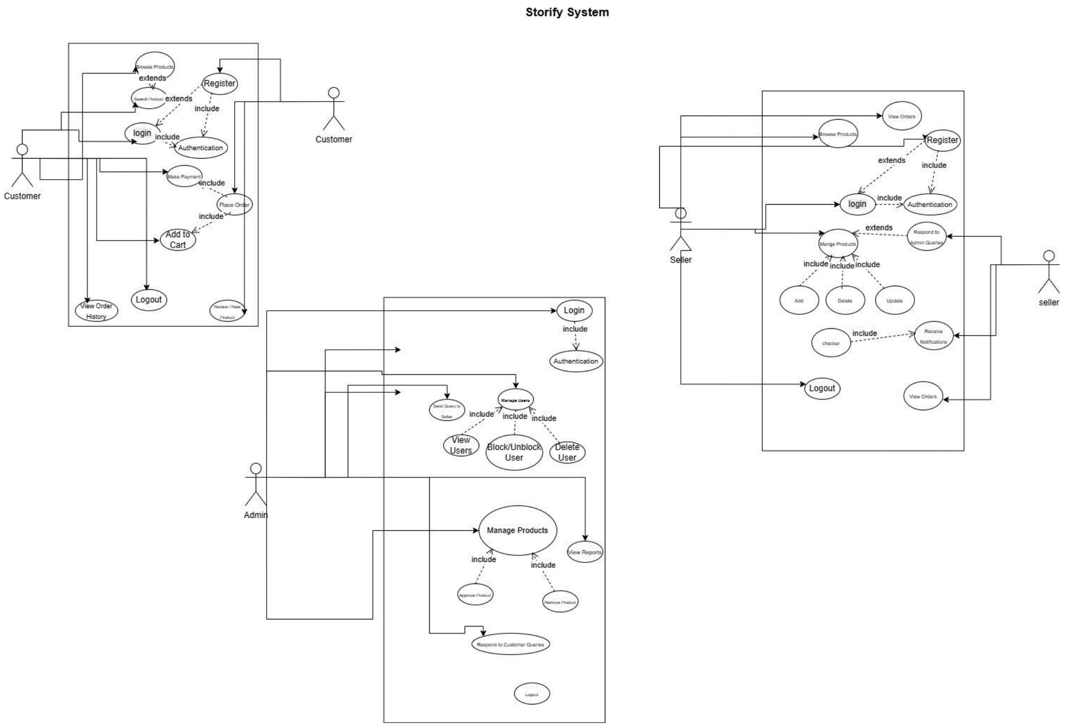

Figure 3.1: Storify System Use Case Diagram

3.2.2 Use Case Descriptions

Use Case 1: Customer

Template Section	**Description**
Use Case ID	UC-C01
Use Case Name	Customer Shopping and Order Management
Primary Actor	Customer
Stakeholders and Interests	Customer wants to browse, purchase, and manage orders easily and securely
Preconditions	Customer must be registered and logged in successfully.

Postconditions	Order placed successfully, cart is cleared, and confirmation is sent.
Main Success Scenario (Basic Flow	Customer logs in → browses products → adds to cart → places order → makes payment → receives confirmation.
Extensions (Alternative Flows)	Invalid login, failed payment, or out-of-stock item.
Includes / Extends (Relationships)	Include: Authentication, Make Payment. Extend: Search Products(extends Browse
Special Requirements	Secure payment processing, user-friendly UI, session timeout for security.
Assumptions	Customer has stable internet and valid payment method.
Frequency of Use	Multiple times daily by active customers.

Use Case 2: Seller

Template Section	Description
Use Case ID	UC-S01
Use Case Name	Product and Order Management for Seller
Primary Actor	Seller
Stakeholders and Interests	Sellers manage their stores, update products, and handle customer orders efficiently.
Preconditions	Seller must be registered and logged in.
Postconditions	Product list updated or order queries resolved.
Main Success Scenario (Basic Flow	Seller logs in → manages products → views orders → responds to customer queries.
Extensions (Alternative Flows)	Invalid login, invalid product data, or network failure.
Includes / Extends (Relationships)	Include: Authentication, Add Product, Delete Product, Update Product, Check Stock.
Special Requirements	Data validation for products, notification system, secure session.
Assumptions	Seller has verified business account and active listings.
Frequency of Use	Used daily or weekly by sellers to manage their stores.

Use Case 3: Admin

Template Section	Description
Use Case ID	UC-A01
Use Case Name	System Administration and User/Product Control
Primary Actor	Admin
Stakeholders and Interests	Admin monitors and maintains the entire system, ensuring smooth and secure operations.
Preconditions	Admin must log in with valid credentials.
Postconditions	System operations updated and logged
Main Success Scenario (Basic Flow	Admin logs in → manages users → verifies products → views reports → responds to queries.

Extensions (Alternative Flows)	Unauthorized access attempt, system error
Includes / Extends (Relationships)	Authentication, Manage Users, Approve Products
Special Requirements	High-level security and activity logging.
Assumptions	Customer has stable internet and valid payment method.
Frequency of Use	Multiple times daily by active customers.

3.3 Domain Model

3.3.1 Conceptual Class Diagram:

The conceptual class diagram provides a high-level overview of the main entities in the **Storify System** and how they are related.
It illustrates the core classes: User, Customer, Seller, Admin, Product, Order, Payment, and Notification — and shows the key relationships between them.

The **User** class is abstract and represents the base for all user types.

- **Customers** can browse products, place orders, and make payments.
- **Sellers** manage products and receive notifications.
- **Admins** monitor and manage the entire platform.

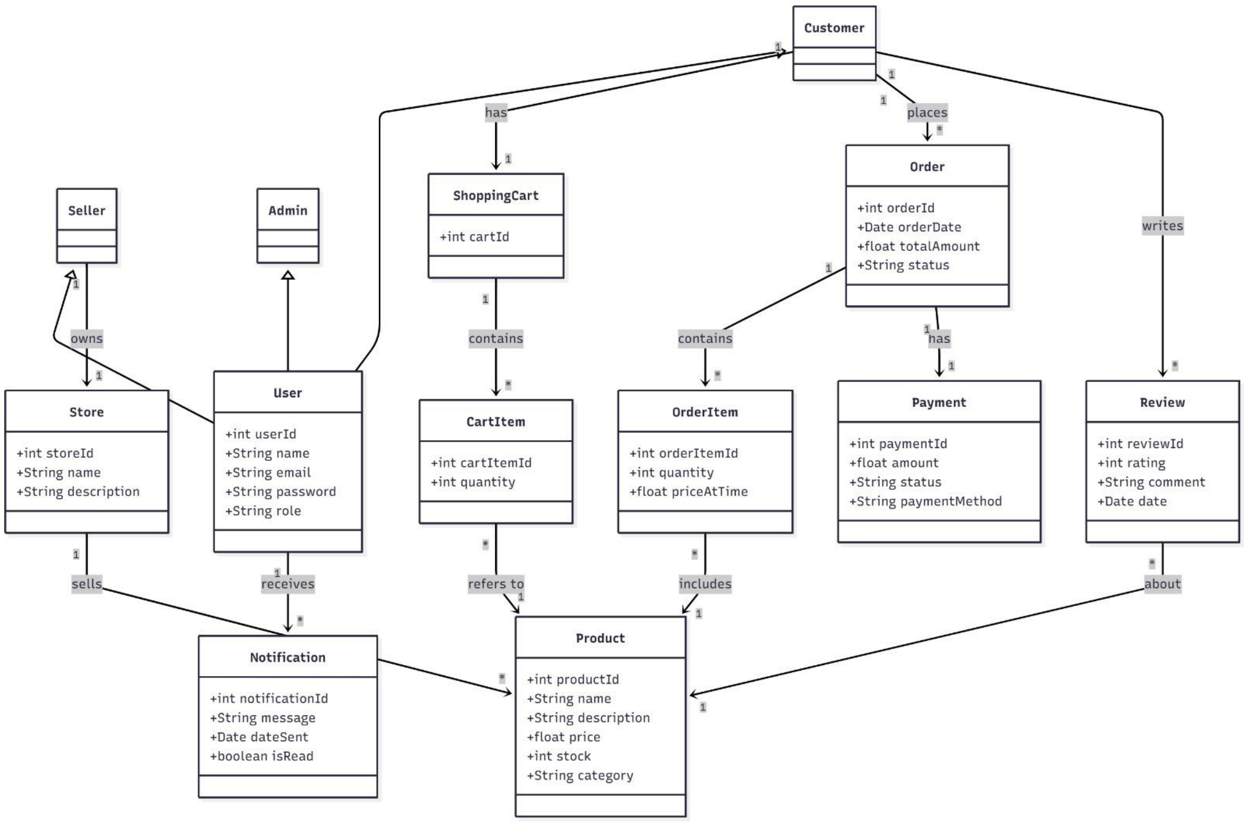

Figure 3.2 shows the conceptual relationships between the main system classes.

3.3.3 Class Descriptions:

Class Name	Attributes	Responsibilities
User (Abstract)	userID, name, email, password	Base class for all user types. Handles authentication and common info.
Customer	address, phone, orderHistory	Browse, purchase products, make payments, and review items.
Seller	storeName, productList	Add, edit, or remove products and manage sales.
Admin	adminID, role	Manage users, verify

		products, and monitor system activity.
Product	productID, name, price, description, stock	Represents an item for sale by a seller.
Order	orderID, date, totalAmount, status	Records details of customer purchases.
Payment	paymentID, method, amount, date	Stores payment information for completed orders.
Notification	notificationID, message, date	Sends system alerts or stock updates to users.

3.4 Non-Functional Requirements

1. Performance Requirements:
   - The system should support at least 200 concurrent users with minimal latency.
2. Security Requirements:
   - All passwords must be encrypted, and transactions must use HTTPS.
3. Usability Requirements:
   - The interface should be intuitive and responsive across all devices.
4. Reliability Requirements:
   - The system uptime must exceed 99% with automated backup every 24 hours.
5. Maintainability Requirements:
   - Modular architecture for easy updates and debugging.
6. Scalability Requirements:
   - Should handle future growth in products and users.

3.5 External Interface Requirements

3.5.1 User Interface

The web interface should be simple, interactive, and consistent. It includes navigation menus, product filters, shopping cart, and checkout pages.

3.5.2 Hardware Interface

The system requires a web server with sufficient processing power and storage capacity.

Users can access the platform through devices like PCs, laptops, or smartphones.

3.5.3 Software Interface

Frontend: HTML, CSS, JavaScript
Backend: Flask (Python)
Database: MySQL
External APIs: Payment Gateway API (e.g., PayPal, Stripe)

3.5.4 Communication Interface

The system communicates over secure HTTPS connections. Notification and email services use the SMTP protocol for alerts.

4. Appendices

4.1 Appendix A: Data Dictionary

Table: User
- UserID (PK), Name, Email, Password, Role

Table: Product
- ProductID (PK), Name, Description, Price, Stock, SellerID (FK)

Table: Order
- OrderID (PK), CustomerID (FK), TotalAmount, PaymentID (FK), Date

Table: Payment
- PaymentID (PK), Method, Status, Amount

Table: Notification

- NotificationID (PK), UserID (FK), Message, Date

4.2 Appendix B: Glossary

Customer: User who browses and purchases products.
Seller: User who lists and manages products.
Admin: Supervisor who oversees the system.
Order: Confirmed purchase made by a customer.
Cart: Temporary storage for selected items before checkout.
Payment Gateway: Service used to process secure payments.
Notification: Alert sent to users regarding updates.