

**Project Name:**  
**Multi-vendor e-commerce platform**

## **Software Requirements Specification**

**Course Code : INT220**  
**SERVER SIDE SCRIPTING**

**AMAN KUMAR SINGH**  
**REG. NO: 12301704**

Prepared for  
Continuous Assessment 3  
Spring 2025

### **Table of Contents**

<b>REVISION HISTORY .....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 PURPOSE .....	1
1.2 SCOPE .....	1

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS .....	1
1.4 REFERENCES .....	2
1.5 OVERVIEW .....	2
<b>2. GENERAL DESCRIPTION .....</b>	<b>2</b>
2.1 PRODUCT PERSPECTIVE .....	3
2.2 PRODUCT FUNCTIONS .....	3
2.3 USER CHARACTERISTICS .....	3
2.4 GENERAL CONSTRAINTS.....	3
2.5 ASSUMPTIONS AND DEPENDENCIES .....	3
<b>3. SPECIFIC REQUIREMENTS.....</b>	<b>3</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
3.1.1 <i>User Interfaces</i> .....	4
3.1.2 <i>Hardware Interfaces</i> .....	4
3.1.3 <i>Software Interfaces</i> .....	4
3.1.4 <i>Communications Interfaces</i> .....	4
3.2 FUNCTIONAL REQUIREMENTS.....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
3.2.1 <Functional Requirement or Feature #1> .....	<b>Error! Bookmark not defined.</b>
3.2.2 <Functional Requirement or Feature #2> .....	<b>Error! Bookmark not defined.</b>
3.5 NON-FUNCTIONAL REQUIREMENTS .....	3
3.5.1 <i>Performance</i> .....	3
3.5.2 <i>Reliability</i> .....	3
3.5.3 <i>Availability</i> .....	3
3.5.4 <i>Security</i> .....	3
3.5.5 <i>Maintainability</i> .....	3
3.5.6 <i>Portability</i> .....	3
3.7 DESIGN CONSTRAINTS .....	3
3.9 OTHER REQUIREMENTS .....	3
<b>4. ANALYSIS MODELS.....</b>	<b>5</b>
4.1 DATA FLOW DIAGRAMS (DFD) .....	4
<b>5. GITHUB LINK.....</b>	<b>5</b>
<b>6. DEPLOYED LINK.....</b>	<b>6</b>
<b>7. CLIENT APPROVAL PROOF.....</b>	<b>7</b>
<b>8. CLIENT LOCATION PROOF.....</b>	<b>8</b>
<b>9. TRANSACTION ID PROOF.....</b>	<b>9</b>
<b>10. EMAIL ACKNOWLEDGEMENT.....</b>	<b>10</b>
<b>11. GST No.....</b>	<b>11</b>
<b>A. APPENDICES.....</b>	<b></b>
A.1 APPENDIX 1.....	
A.2 APPENDIX 2.....	

## 1. Introduction

*This document formally specifies all requirements for Snapkart's multi-vendor e-commerce platform. It serves as the single source of truth for developers to understand exactly what to build, ensuring all stakeholders share the same vision.*

*The SRS will be used throughout the development lifecycle - from initial coding to final testing. It prevents scope creep by clearly defining system boundaries and functionality.*

### 1.1 Purpose

*The purpose of this SRS document is to define the requirements for the development of an e-commerce web application. It serves as a reference for all stakeholders including developers, designers, testers, and clients. This document ensures that all functionalities and expectations are clearly outlined to guide the software development life cycle.*

### 1.2 Scope

*The system to be developed is a full-stack e-commerce web application that enables users to browse products, add them to a shopping cart, and proceed with a secure checkout. It also provides an admin panel to manage products, categories, orders, and customer details. The*

*project will focus on usability, performance, and secure transaction handling through an intuitive UI.*

### **1.3 Definitions, Acronyms, and Abbreviations**

- *SRS: Software Requirements Specification*
- *UI: User Interface*
- *UX: User Experience*
- *DBMS: Database Management System*
- *CRUD: Create, Read, Update, Delete*
- *HTTPS: Hypertext Transfer Protocol Secure*
- *MVC: Model View Controller*

### **1.4 References**

- *IEEE SRS Standard (IEEE 830-1998)*
- *Tailwind CSS Documentation (<https://tailwindcss.com/docs>)*
- *PHP Manual (<https://www.php.net/manual/en/>)*

### **1.5 Overview**

*This document is organized into sections detailing the overall system characteristics, specific functional and non-functional requirements, user interaction, design constraints, and analysis*

*models. It aims to capture every aspect necessary to develop the application in accordance with client expectations and industry best practices.*

## **2. General Description**

*Snapkart operates as an independent web application with PHP backend and MySQL database. It follows the MVC pattern with clear separation between presentation, business logic, and data layers.*

*The system currently integrates with SMTP for emails but is designed for future expansion. All external integrations use standardized APIs with proper authentication mechanisms.*

### **2.1 Product Perspective**

*The product is a new, standalone web application, not dependent on any existing platform or software. It will interact with a MySQL database and possibly a payment gateway API for transaction processing. It is built using a modern tech stack including JavaScript, PHP, and Tailwind CSS.*

### **2.2 Product Functions**

*The system allows users to register, log in, view and search products, and complete purchases. Admins can add new products, manage inventory, and track orders. The application also includes features like a secure payment system, order history, and customer support integration.*

### **2.3 User Characteristics**

*Target users include general online shoppers with basic internet literacy and administrative users with access to backend management tools. The UI will be designed for ease of use with minimal learning curve. Accessibility for differently-abled users will also be considered.*

### **2.4 General Constraints**

*The application must work on various screen sizes (responsive design) and be compatible with all major browsers. It should comply with legal and security standards, including data protection laws. Resource constraints like hosting and budget will influence the tech stack and deployment strategy.*

### **2.5 Assumptions and Dependencies**

*We assume users have access to reliable internet and modern web browsers. The server environment will support PHP and MySQL. Payment processing is dependent on third-party APIs, whose availability and uptime are out of our control.*

## **3. Specific Requirements**

### **3.1.1 User Interfaces**

*the user interface will be clean and user-friendly, built using Tailwind CSS for styling and responsiveness. Forms will be validated both client-side and server-side for robust input handling. The admin panel will feature dashboards with charts and data tables.*

### **3.1.2 Hardware Interfaces**

*The application will be accessed using personal computers or mobile devices with internet connectivity. No specialized hardware is required. Server-side will run on Linux/Windows hosting environments.*

### **3.1.3 Software Interfaces**

*The application will be accessed using personal computers or mobile devices with internet connectivity. No specialized hardware is required. Server-side will run on Linux/Windows hosting environments.*

### **3.1.4 Communications Interfaces**

*All communication will occur over HTTPS to ensure secure data transmission. Web hosting must support SSL certification and secure sockets layer encryption.*

## **3.2 Functional Requirements**

**3.2.1 User Registration and Authentication** *Users must be able to sign up and log in securely. Email validation and password encryption will be implemented. Session management will ensure persistent login where required.*

**3.2.2 Product Management** *Admins can create, update, and delete product listings from the admin panel. Products will include descriptions, images, prices, and stock availability. All operations will be logged.*

**3.2.3 Shopping Cart and Checkout** *Users can add items to their cart, change quantities, or remove items. At checkout, user data is validated and stored. Integration with a payment gateway will allow real-time payment processing.*

## **3.5 Non-Functional Requirements**

**3.2.1 User Registration and Authentication** *Users must be able to sign up and log in securely. Email validation and password encryption will be implemented. Session management will ensure persistent login where required.*

**3.2.2 Product Management** *Admins can create, update, and delete product listings from the admin panel. Products will include descriptions, images, prices, and stock availability. All operations will be logged.*

**3.2.3 Shopping Cart and Checkout** *Users can add items to their cart, change quantities, or remove items. At checkout, user data is validated and stored. Integration with a payment gateway will allow real-time payment processing.*

## **3.5 Non-Functional Requirements**

**3.5.1 Performance** *The system should respond to user actions in under 2 seconds. It should support concurrent usage by thousands of users. Static resources will be cached to optimize performance.*

**3.5.2 Reliability** *System availability must be 99.9% with minimal downtime. Failover procedures and data redundancy will ensure business continuity in case of failure.*

**3.5.3 Availability** *The application must be available 24/7. Hosting services will include uptime monitoring and auto-scaling features to ensure consistent availability.*

**3.5.4 Security** *User data will be encrypted using industry standards. The system will implement protection against XSS, CSRF, and SQL injection. Admin access will be role-based.*

**3.5.5 Maintainability** *Code will follow modular architecture and documentation standards to ensure easy maintenance. Logs will help trace and debug issues efficiently.*

**3.5.6 Portability** *The application will be tested on different operating systems and browsers. Docker containers may be used to ensure environment independence.*

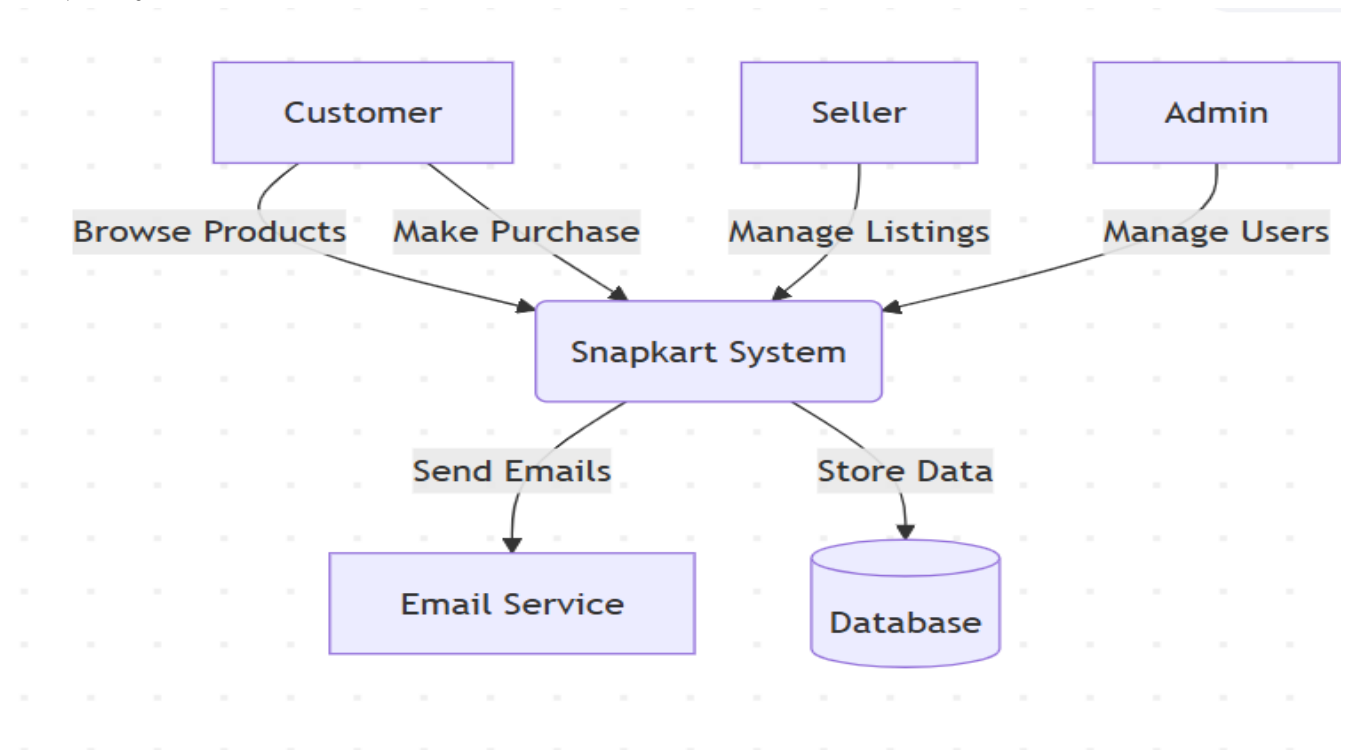
**3.7 Design Constraints** *Design must follow MVC architecture. The project must use open-source technologies . And Should as low cost as possible.*

**3.9 Other Requirements** *Support for localization and multi-currency should be available in future updates. The app must provide a basic analytics dashboard for the admin.*

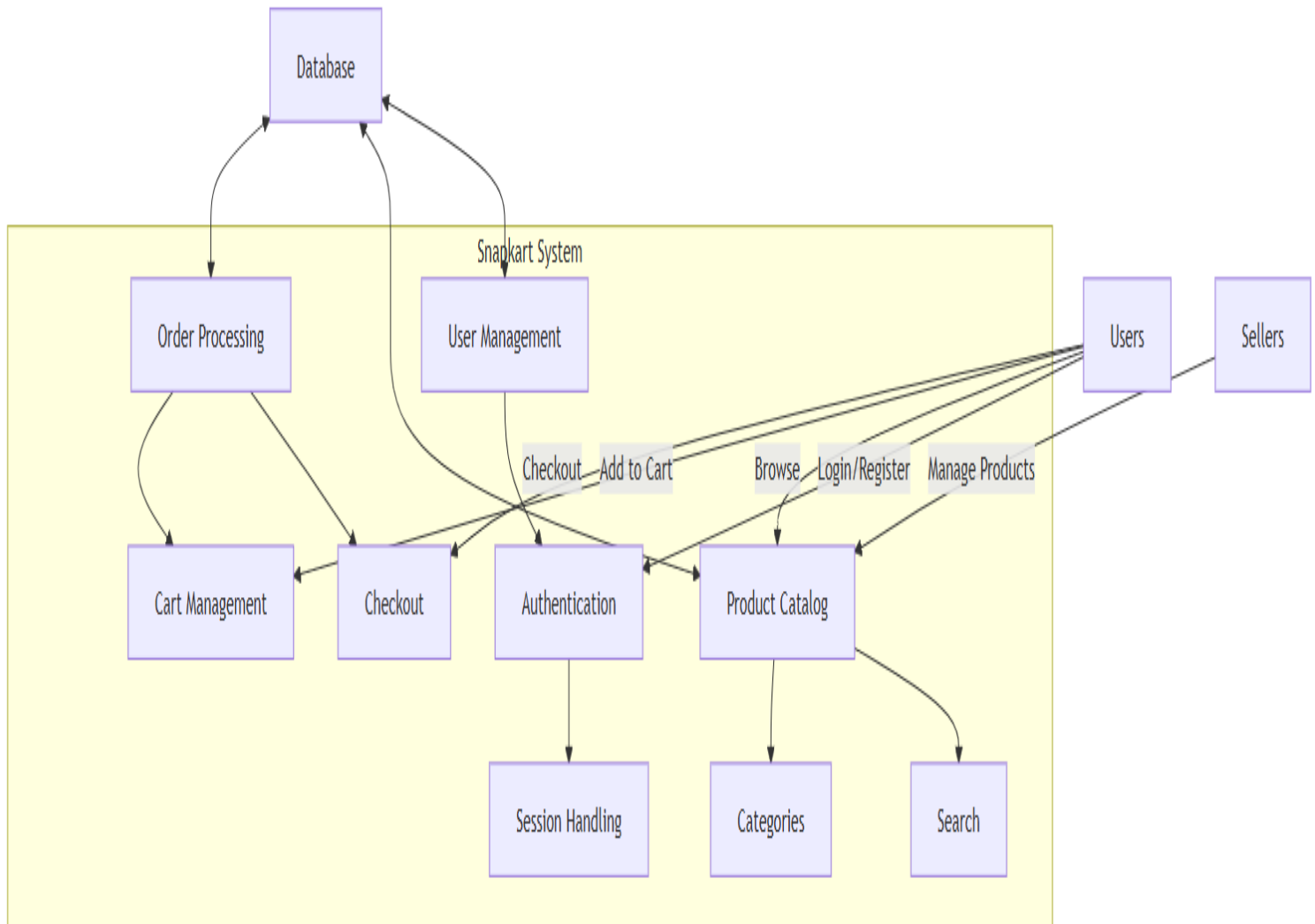
## 4. Analysis Models

### 4.1 Data Flow Diagrams (DFD)

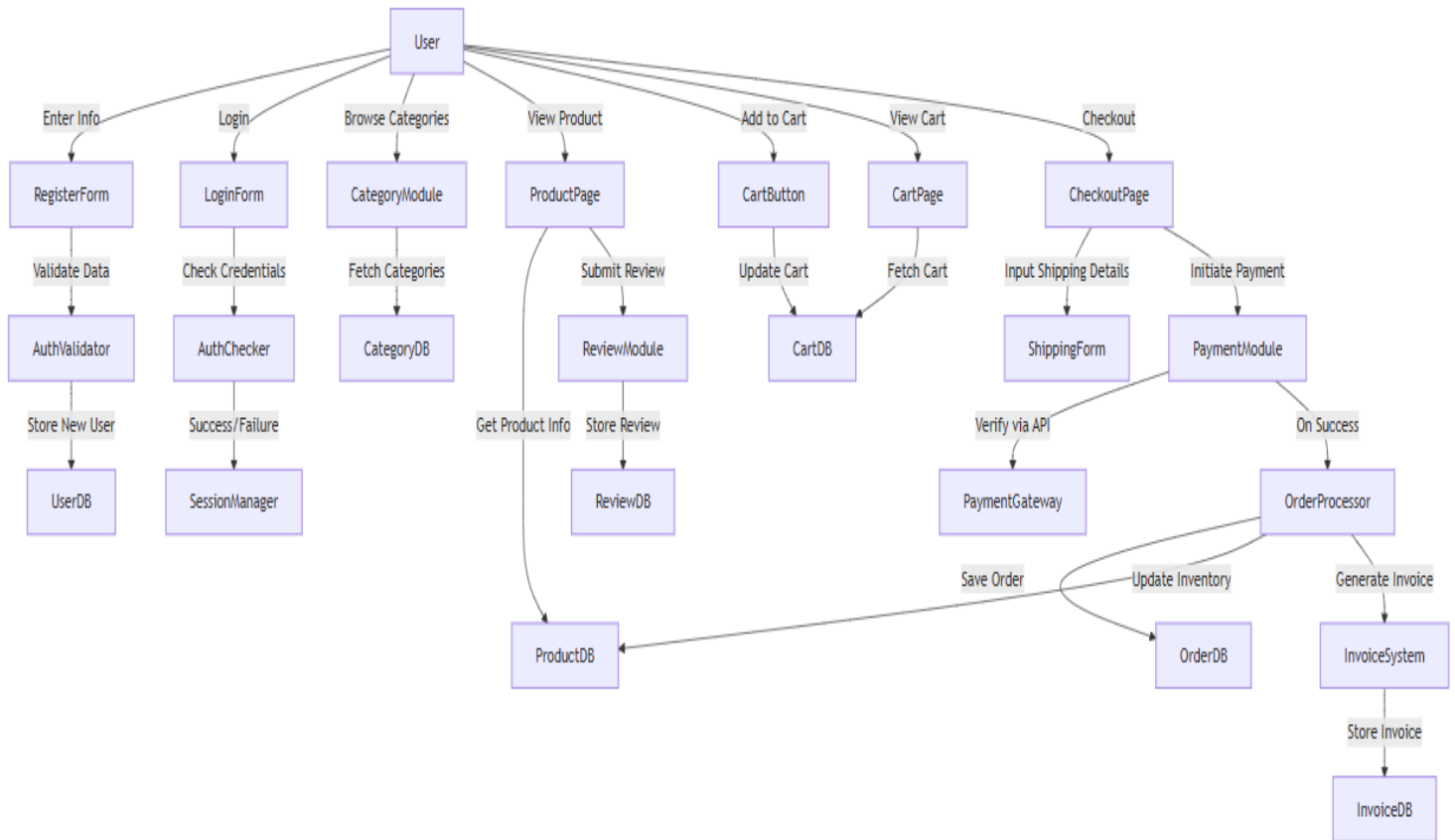
#### LEVEL 0



## LEVEL 1





**LEVEL 2****Appendices**

**A.1 Appendix 1** This appendix includes initial design wireframes created for the e-commerce platform. It shows mockups of core pages such as home, product listing, product details, user login, and shopping cart. These visuals served as a reference for the frontend development team during implementation.

**A.2 Appendix 2** This section contains detailed records of QA test cases, including functionality, UI, and performance tests. Also included are bug reports discovered during the testing phase and user feedback obtained during prototype testing. These were crucial in improving final system quality and usability.

