VISHWAKARMA UNIVERSITY
Maximising Human Potential

Activity based

Project Report on

Software Engineering

Project Module - II

Submitted to Vishwakarma University, Pune

Under the Initiative of

Contemporary Curriculum, Pedagogy, and Practice (C2P2)

By

Digvijay Balasaheb Gadhave

SRN No : 202101041

Roll No : 18

Div : A

Third Year Engineering

Department of Computer Engineering

Faculty of Science and Technology

Academic Year

2023-2024

Software Engineering: Project Module II

Project Name : SRS Plan and UML Diagram for Shopify

Part I : Write a Software Requirement Specification (SRS) Plan for Shopify.

Explanation of each technique and its applicability in mentioned project

- **Introduction**

Purpose of Shopify:

Enable businesses to set up and manage online stores easily.

Simplify tasks such as product listing, order processing, and marketing.

Provide a platform for businesses to sell products and services online.

Offer tools for inventory management, customer communication, and sales analytics.

Scope of Shopify:

Store Creation and Customization: Customize the look and feel of the online store.

Product Management: Add, edit, and categorize products for sale.

Order Processing: Track orders, manage fulfillment, and process payments.

Inventory Management: Track stock levels, receive low stock alerts, and manage product variants.

Marketing Tools: Optimize for search engines, run email marketing campaigns, and integrate with social media.

Sales Features: Offer discounts, promotions, and recover abandoned carts.

Customer Management: Manage customer accounts, view order history, and communicate with customers.

Analytics and Reporting: Generate sales reports, analyze customer data, and track conversions.

❖ Functional Requirements-

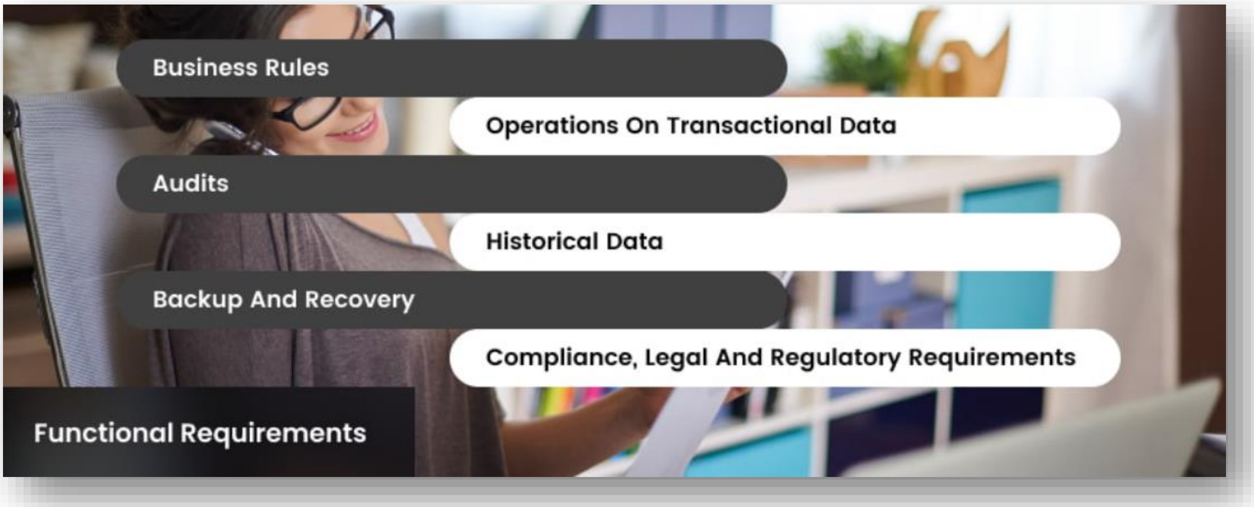
SRS (Software Requirements Specifications) or FS (Functional Specifications) defines the behavior of the system and also the functioning of the Shopify application and web design. These documents are termed Functional Requirements. The SRS is a document that defines the layout and technical terms/parameters on which the software is to be designed. There are several methodologies available in the industry to write and construct functional requirements documents.

Types of functional requirements in SRS:

- Business Rules - What features do you want to embed in your system?
- Operations on Transactional data - It aims at examining every transactional entry, modification, and deletion of data entry.
- Authentication and Authorization - The concern is regarding the security at different levels and provision of authority to users to apply_CRUD (Change, Read, Update and Delete) on operations.
- Audits - The tracking of critical data and various authority checks performed.
- Historical data - The exponential growth of data needs to be managed effectively and stored.
- Backup and Recovery - If the system crashes and the data wipes out then the mechanism for backup and recovery is a savior.
- Compliance, legal and regulatory requirements - Governance regulations, internal policies, and laws that are centralized within the organization

Functional requirements more have to do with the quantity factors of the Shopify application and web design developed. The focus is the **'WHAT'** here rather than the **'WHY'.**

FIG.1


Business Rules
Operations On Transactional Data
Audits
Historical Data
Backup And Recovery
Compliance, Legal And Regulatory Requirements
Functional Requirements

❖ Non-Functional Requirements

Non-Functional requirements focus on the **‘HOW’** aspect of system functionalities. It is also driven by user experiences and the ability of users perception about the Shopify application and web design. The definition is a bit difficult and doesn’t have direct measures to identify the performance.

Below enlisted are a few non-functional requirements:

- Usability - user’s ability to interact or communicate with the product/software
- Reliability - sensitivity in handling the data and information
- Performance - speedy interactions with the user are the new normal
- Error handling - the ability of the system to withstand error

FIG.2

❖ System Requirements

- *Internet Connection:* A stable internet connection is required to access and manage your Shopify store.
- *Supported Web Browser:* Shopify supports modern web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
- *Device Compatibility:* Shopify can be accessed from desktop computers, laptops, tablets, and smartphones.
- *Operating System:* As Shopify is web-based, it is compatible with various operating systems including Windows, macOS, iOS, and Android.

- *Hardware Requirements:* No specific hardware requirements are needed beyond what is necessary to run the supported web browsers and operating systems.

❖ User Requirements

  - *Ease of Use:* Users expect Shopify to be user-friendly and intuitive, allowing them to easily navigate and manage their online store without needing extensive technical knowledge.

  - *Customization:* Users may require the ability to customize their store's appearance, including themes, colors, and layouts, to reflect their brand and meet their design preferences.

  - *Product Management:* Users need tools to easily add, edit, and manage their products, including descriptions, prices, images, and inventory levels.

  - *Order Management:* Users expect features for managing orders, including processing payments, tracking shipments, and handling returns and exchanges.

  - *Marketing and SEO:* Users may require tools for marketing their store, including SEO optimization, email marketing, and social media integration, to attract and retain customers.

  - *Analytics and Reporting:* Users need access to analytics and reporting tools to track their store's performance, including sales data, traffic sources, and customer behavior.

❖ Constraints

- *Budget*: The budget available for developing and maintaining the Shopify store may constrain the choice of features, customization options, and marketing efforts.

- *Time:* The timeline for launching the Shopify store may constrain the amount of time available for development, testing, and implementation, which could impact the scope of the project.

- *Technical Limitations*: Constraints related to the Shopify platform itself, such as limitations on customization options, API restrictions, or compatibility issues with third-party apps or integrations.

- *Resource Availability:* Constraints related to the availability of resources, such as skilled developers, designers, or content creators, which may impact the speed and quality of development.

- *Legal and Regulatory Requirements:* Constraints related to legal and regulatory requirements, such as data protection laws, which may impact the design and functionality of the Shopify store.

- *Compatibility:* Constraints related to compatibility with existing systems, tools, or processes, which may limit the choice of Shopify apps or integrations that can be used.

    - *Scalability:* Constraints related to the scalability of the Shopify store, such as limitations on the number of products, transactions, or users that the store can handle, which may impact long-term growth and expansion plans.

❖ Assumptions and Dependencies

Assumptions and dependencies for Shopify can include:

Assumptions:

- *Internet Connection:* Users have a stable internet connection to access and manage their Shopify store.
- Supported Web Browser: Users are using a modern web browser supported by Shopify, such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge.
- *Device Compatibility:* Users are accessing Shopify from a device (desktop, laptop, tablet, or smartphone) that meets Shopify's compatibility requirements.
- *Operating System:* Users' devices are running an operating system (Windows, macOS, iOS, Android) that is compatible with Shopify.

- *Product Information:* Users have accurate and up-to-date information about their products, including descriptions, prices, and inventory levels.
- *Payment Gateway:* Users have set up a payment gateway to process transactions on their Shopify store.
- *Legal Compliance:* Users comply with legal and regulatory requirements related to online selling, including data protection and consumer rights laws.

Dependencies:

- *Third-Party Apps:* Shopify stores may depend on third-party apps for additional features and functionality, such as marketing tools, analytics, or inventory management.
- *Payment Processors:* Shopify stores depend on payment processors to securely process transactions and handle payment-related tasks.
- *Shipping Providers:* Shopify stores depend on shipping providers to deliver products to customers and manage shipping logistics.
- *Theme Customization:* Shopify stores may depend on theme customization to create a unique and branded storefront.
- *API Integrations:* Shopify stores may depend on API integrations to connect with external systems or services, such as accounting software or CRM systems.
- *Customer Support:* Shopify stores may depend on Shopify's customer support for assistance with technical issues or questions about the platform.

❖ Functional Models

Use Case Diagrams: Use case diagrams help identify the different types of users (actors) interacting with the system and the various tasks (use cases) they perform. This includes actors like administrators, store owners, customers, and suppliers, and use cases such as managing products, processing orders, and managing customer accounts

.

Class Diagrams: Class diagrams help model the static structure of the system by identifying the classes, attributes, methods, and relationships between objects. This includes classes such as products, orders, customers, sellers, and various modules within the system.

Sequence Diagrams: Sequence diagrams depict the interactions between different objects or components over time, showing the sequence of messages exchanged during a particular use case or scenario. This includes sequences like placing an order, processing payments, or managing inventory.

Entity-Relationship Diagrams : ER diagrams depict the relationships between different entities or tables in the database schema. This includes entities such as products, orders, customers, suppliers, payments, and other relevant data entities

- **Use Case**

Actors:

1.InventoryManager
2.Payment Gateway
3.Shipping Service
4.Customer
5.Actor

Relationships:

Associations: Actors interact with use cases. For example, Customer interacts with Browse Products, Add to Cart, Remove from Cart, View Cart, and Checkout.

Generalization: One actor is a specialized version of another. For example, Customer and Admin are both Users.

Extends: Additional functionality added to use cases under certain conditions. For example, Add Discount adds functionality to Manage Products, Apply Discount adds functionality to Checkout, View Order History adds functionality to Customer, and Notify Low Inventory adds functionality to InventoryManager.

Cases:

- **Browse Products:** Customer can browse the products available in the store.
- **Add to Cart**: Customer can add products to their shopping cart.
- **Remove from Cart:** Customer can remove products from their shopping cart.
- **View Cart:** Customer can view the products currently in their shopping cart.
- **Checkout:** Customer can proceed to checkout to purchase the products in their cart.
- **Manage Products:** Admin can manage (add, edit, delete) products in the store.
- **Manage Inventory:** InventoryManager can manage the inventory of products in the store.
- **Process Payment:** PaymentGateway can process payments for the purchased products.
- **Manage Shipping**: ShippingService can manage the shipping of products to customers.
- **Add Discount** (extends Manage Products): Admin can add discounts to products.
- **Apply Discount** (extends Checkout): Customer can apply discounts to their purchase during checkout.
- **View Order History** (extends Customer): Customer can view their order history.
- **Notify Low Inventory** (extends InventoryManager): InventoryManager can receive notifications about low inventory.

.

❖ Traceability Matrix.

ID	Description	Design	Implementation	Requirement
1	Frontend	✓		✓
2	Backend	✓		✓
3	Allow users to create accounts	✓	✓	✓
4	Enable shopping cart functionality	✓		✓
5	Provide product catalog	✓	✓	✓
6	Implement order processing	✓		
7	Offer discounts and promotions	✓		
8	Support multiple payment methods	✓		

❖ Glossary

- *Shopify:* An e-commerce platform that allows businesses to create online stores and sell products.

- *Online Store:* A website where customers can browse and purchase products from a Shopify merchant.

- *Product:* An item that is sold by a Shopify merchant through their online store.

- *Storefront:* The customer-facing part of a Shopify store where products are displayed and can be purchased.

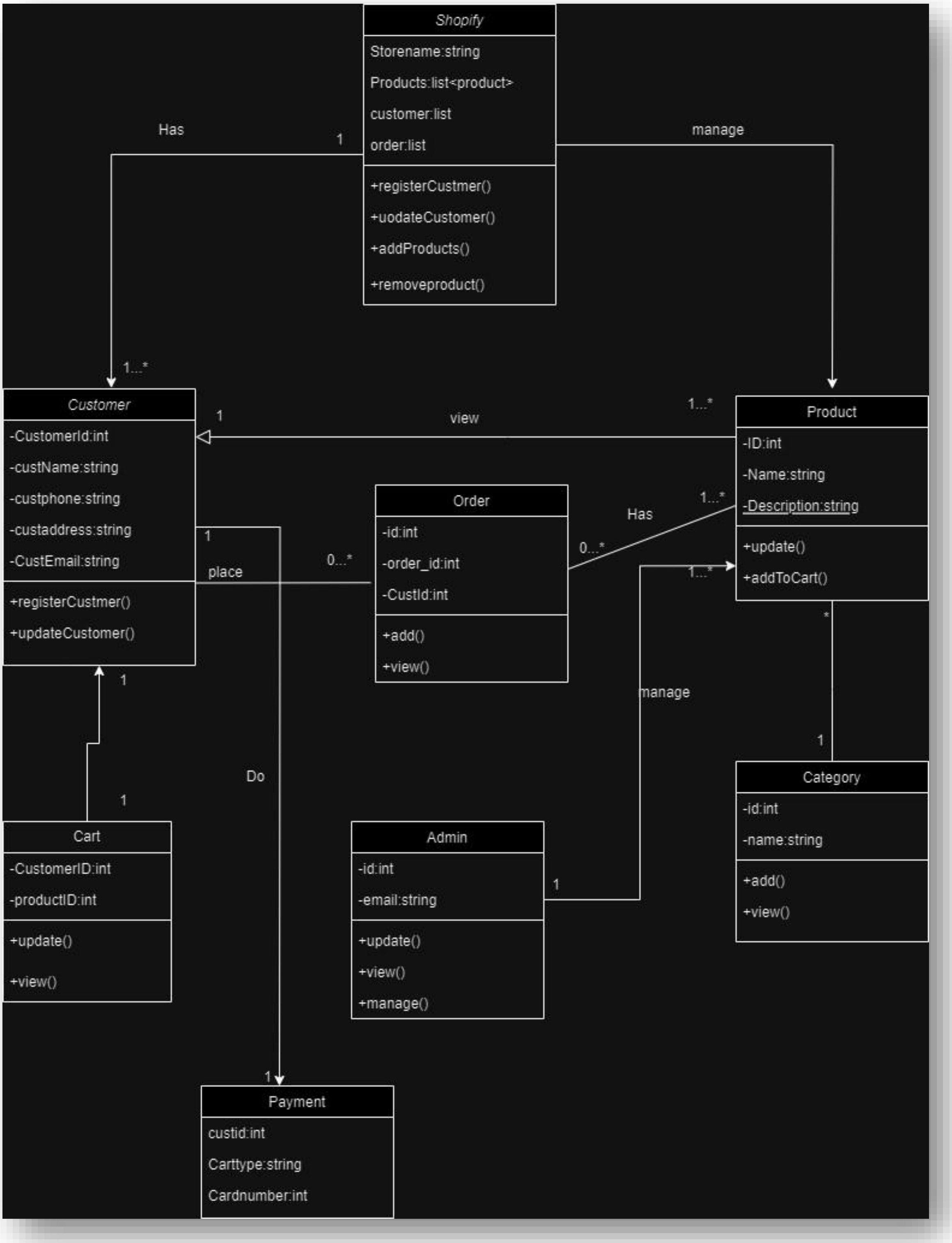
- *Dashboard:* The main administrative interface of a Shopify store where merchants can manage their products, orders, and customers.

- *Theme:* A pre-designed template that determines the look and feel of a Shopify store.

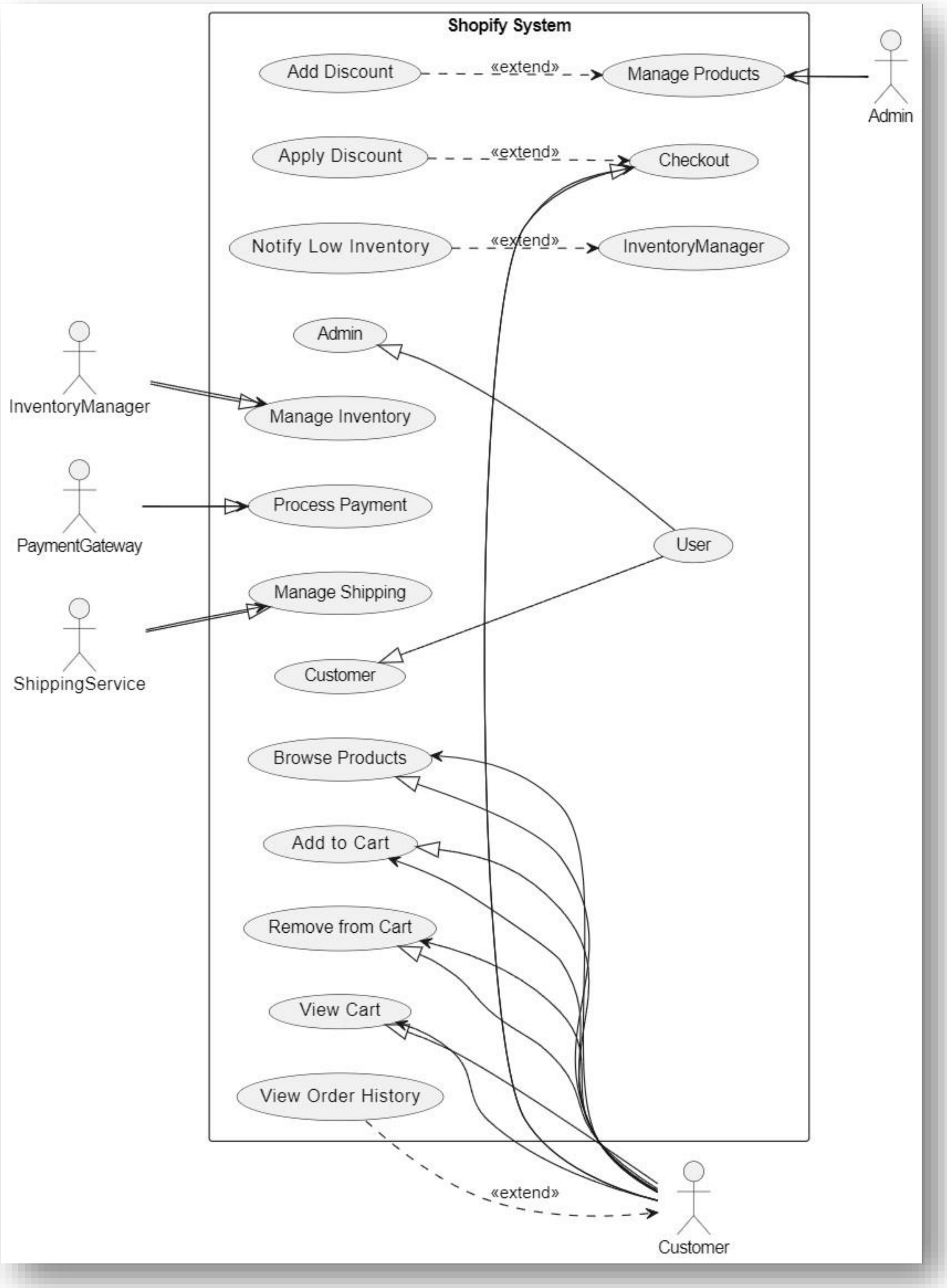
- *Order:* A customer's request to purchase one or more products from a Shopify store.
- *Inventory*: The stock of products that a Shopify merchant has available for sale.
- *Payment Gateway:* A service that processes payments for Shopify stores, allowing customers to pay for their orders securely.
- *Shipping:* The process of delivering products to customers after they have been purchased from a Shopify store.
- *App:* A software application that can be installed on a Shopify store to add new features and functionality.
- *Customer:* A person who purchases products from a Shopify store.
- *Marketing:* The process of promoting a Shopify store and its products to attract customers and drive sales.
- *Analytics:* Data and statistics about a Shopify store's performance, such as sales, traffic, and customer behavior.
- *Theme Customization:* The process of modifying a Shopify store's theme to customize its appearance and layout.

Part II : Draw following UML diagram for above mentioned project

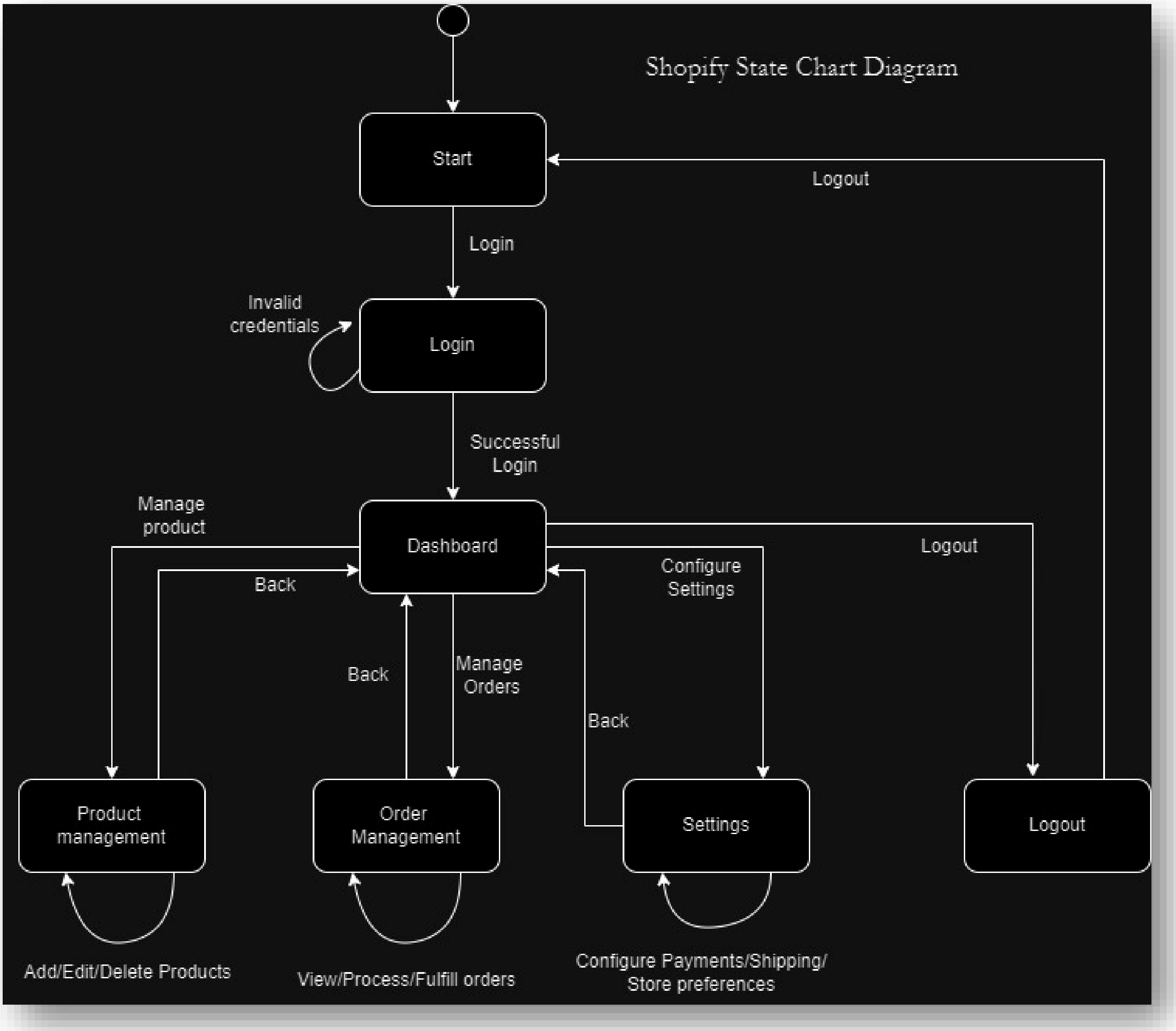
- **Class Diagram**



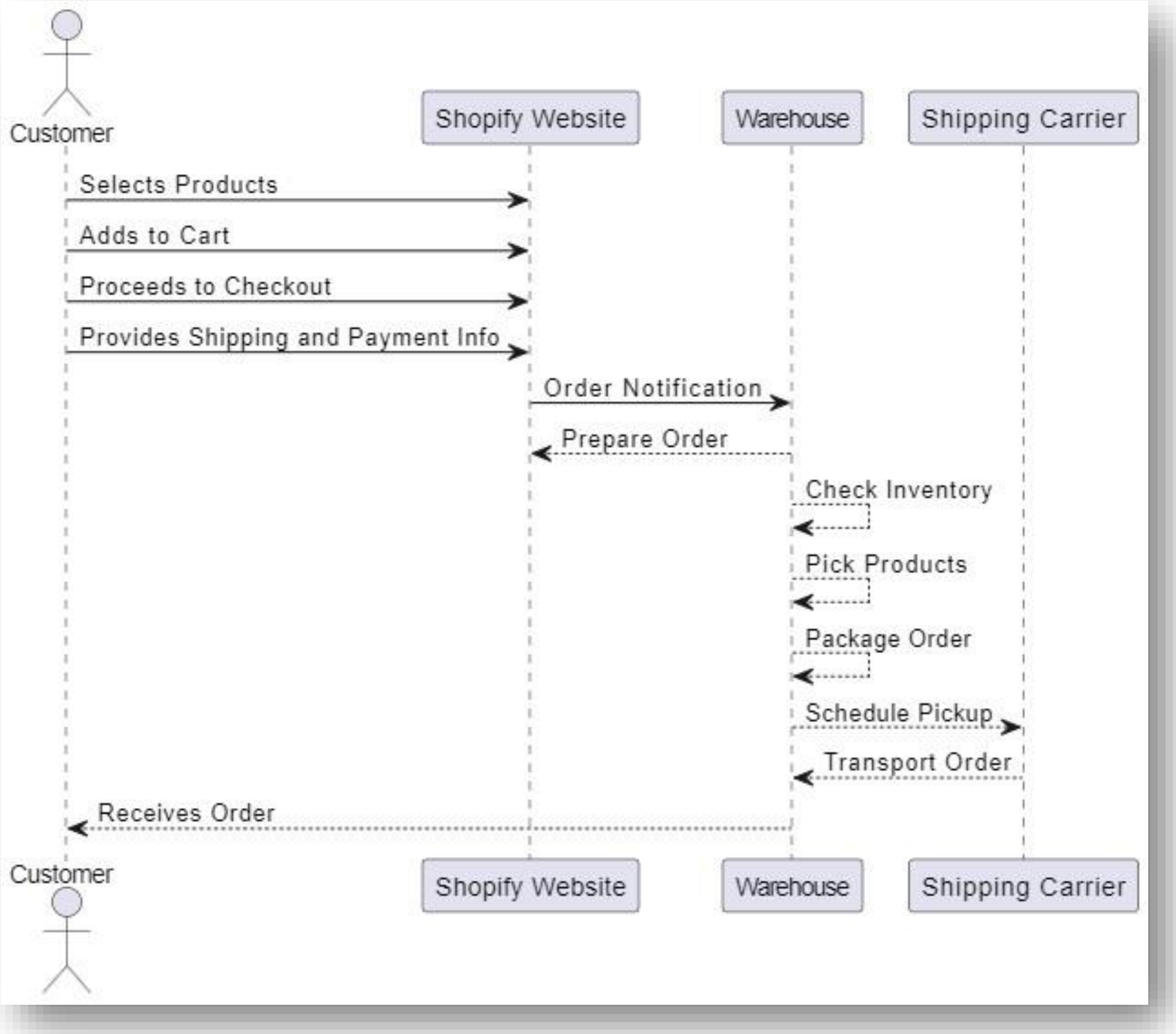
- **Use Case Diagram**



- **State Chart Diagram**



Sequence Diagram



-

Conclusion :

The project report presents a detailed Software Requirement Specification (SRS) plan for Shopify, including its purpose, scope, functional and non-functional requirements, system requirements, user requirements, constraints, assumptions, dependencies, and four diagrams (Class Diagram, Use Case Diagram, State Chart Diagram, and Sequence Diagram). These elements provide a clear framework for the development and implementation of the Shopify e-commerce platform.