



University of Bahrain  
College of Information Technology  
Department of Computer Science  
ITCS 389: Software Engineering I

# Online Shopping System

**Phase Number:** 1, 2 and 3

**Phase Title:** Final Report

**Due Date:** 12/12/2024

**Team Members:**

**Project Manager:**

Shahad Yaser Abdulrahman Nassar	202210309
---------------------------------	-----------

**Members:**

Maimoona Abdulrahman Alawadhi	202208599
-------------------------------	-----------

Amal Abdulhameed Ahmed	202204700
------------------------	-----------

Maryam Ali Jama	202107498
-----------------	-----------

# Table of Contents

<b>Chapter 1   Project Management</b> .....	5
<b>1.1 Introduction</b> .....	5
Project Overview .....	5
Client/Organization .....	5
Stakeholders .....	5
Client's Business .....	6
Existing System .....	6
Business Goals .....	6
<b>1.3 Problem Definition</b> .....	7
Specific Issues .....	7
Impact Analysis.....	7
Desired Outcomes .....	8
<b>1.4 Project Objectives</b> .....	8
<b>1.5 DevOps Model</b> .....	9
<b>Chapter 2   System Analysis</b> .....	11
<b>2.1 Requirements Collection</b> .....	11
<b>2.2 Functional Requirements</b> .....	14
2.2.1 Detailed Descriptions .....	14
2.2.2 Prioritization .....	21
<b>2.3 Non-Functional Requirements</b> .....	22
<b>2.4 Data Flow Diagrams (DFD)</b> .....	24
2.4.1 Context Level DFD .....	24
2.4.3 Level 1 DFDs .....	26
2.4.3.1 Process 1.0 .....	26
2.4.3.2 Process 2.0 .....	26
2.4.3.3 Process 3.0 .....	27
2.4.3.5 Process 5.0 .....	28

<b>2.5 Process Specification .....</b>	<b>28</b>
<b>2.5.1 Structured English .....</b>	<b>28</b>
<b>2.5.2 Decision Tables .....</b>	<b>30</b>
<b>2.6 Entity-Relationship Diagram (ERD) .....</b>	<b>31</b>
<b>Chapter 3   Object-Oriented Analysis .....</b>	<b>32</b>
<b>3.1 Use Case Diagram .....</b>	<b>32</b>
<b>3.2 State Transition Diagram (STD) .....</b>	<b>35</b>
<b>3.2.3 STD for Admins .....</b>	<b>36</b>
<b>3.3 Class Diagram .....</b>	<b>37</b>
<b>3.4 Sequence Diagram .....</b>	<b>37</b>
<b>3.5 Activity Diagram .....</b>	<b>40</b>
<b>References .....</b>	<b>43</b>

## Table of Figures

Figure 1 DevOps .....	9
Figure 2 context level DFD .....	24
Figure 3 level 0 DFD.....	25
Figure 4 level 1 DFD.....	26
Figure 5 Entity-relationship diagram .....	31
Figure 6 UML use case diagram .....	32
Figure 7 View Item Use Case .....	33
Figure 8 Checkout Use Case.....	33
Figure 9 Contact Use Case.....	34
Figure 10 User Registration Use Case .....	34
Figure 11 State transition diagram for customers.....	35
Figure 12 State transition diagram for suppliers.....	35
Figure 13 State transition diagram for admin.....	36
Figure 14 Class diagram of the system.....	37
Figure 15 Sequence diagram for customer .....	38
Figure 16 Sequence diagram for admin .....	39
Figure 17 Sequence diagram for suppliers.....	40
Figure 18 User login and registration Activity Diagram .....	40
Figure 19 Browsing and Searching Activity Diagram .....	41
Figure 20 Adding items to cart and checkout Activity Diagram .....	41
Figure 21 Supplier Activity Diagram.....	42
Figure 22 Admin Activity Diagram.....	42

## Table of Tables

Table 1 Interview outline .....	13
Table 2 Interview questions.....	14
Table 3 Process 1.0 Details.....	28
Table 4 Process 2.0 Details.....	29
Table 5 Process 3.0 Details.....	29
Table 6 Process 4.0 Details.....	30
Table 7 Process 5.0 Details.....	30
Table 8 Process 1.0 Decision Table (Manage User) .....	30
Table 9 Process 5.0 Decision Table (Manage Inventory) .....	30

# Chapter 1 | Project Management

## 1.1 Introduction

### Project Overview

The Online Shopping System is comprehensive software designed to make it easier to purchase and sell products and services over the internet. The project's scope includes creating software that allows customers to register, explore products in the catalogue, add items to cart, make secure payments, track orders, and to provide customer service assistance. The primary purpose of this project is to provide customers with a user-friendly, safe, seamless, and efficient online purchasing experience, and ensure data security, as well as a strong platform for businesses to manage their online sales operations, leading to increased sales and customer satisfaction.

The aim of this phase is managing the project which means following a set of principles and procedures to manage a project. The main purpose of this phase is collecting information and requirements for a new system and determining the existing problems in the current systems to develop a better and high-quality system.

### Client/Organization

The system is developed to serve customers interested in online shopping of all ages and also to serve all e-commerce companies, which are medium-sized retail companies that seek to increase their market reach via an online platform and aim to improve their digital presence and provide customers with an effective online shopping experience in order to attract more buyers and enhance their overall sales performance.

### Stakeholders

The main stakeholders in the Online Shopping System project are:

- Project admins: leadership team who fund and manage the project.
- Customers: Online shoppers.
- Suppliers (Vendors): Goods providers.

Each of these stakeholders plays a crucial role in the overall functionality and will guarantee that the Online Shopping System satisfies the demands of both the organization and its consumers, leading to a successful digital buying experience.

## 1.2 Background

### Client's Business

The client operates within the e-commerce industry, specifically targeting medium-sized retail businesses. Their operations focus on providing a wide array of products and services through an online platform, allowing customers to shop conveniently from any location. The target market includes tech-savvy consumers who value the accessibility and efficiency of online shopping, ranging from busy professionals to families seeking convenience. By utilizing digital platforms, the client aims to enhance customer engagement and foster long-term relationships with their audience.

### Existing System

Currently, the client employs a mix of manual processes and outdated software to manage their online sales. This includes basic website functionalities that do not integrate well with inventory management, payment processing, or customer service tools. Key limitations of the existing system include:

- **User Experience:** The current interface is not user-friendly, which can lead to customer frustration. Navigation is often cumbersome, potentially deterring purchases.
- **Payment Security:** Existing payment methods lack robust security features, raising concerns about data protection and customer trust.
- **Order Tracking:** Customers face challenges in effectively tracking their orders, resulting in increased inquiries to customer support.
- **Inventory Management:** Manual inventory tracking processes can lead to discrepancies, negatively impacting sales and customer satisfaction.

These inefficiencies hinder the client's ability to compete effectively in a rapidly evolving digital marketplace.

### Business Goals

The new Online Shopping System is designed to align closely with the client's overarching business goals, which include:

- **Enhancing Digital Presence:** By implementing a modern, user-friendly platform, the client aims to attract a broader audience and improve brand visibility.
- **Increasing Sales:** Streamlined purchasing processes and enhanced customer engagement are expected to drive higher conversion rates (percentage of potential customers who take a desired action, such as making a purchase) and thus a high return on investment and boost overall sales performance.

- **Improving Customer Satisfaction:** The system will be easy to navigate (user-friendly), provide secure payment options, efficient order tracking, and responsive customer support, leading to a more satisfying shopping experience.
- **Data Security:** With a high focus on robust security, the new system will ensure that sensitive customer data such as personal and payment information is protected, increasing trust and loyalty from the customer.

## 1.3 Problem Definition

### Specific Issues

- Complicated user interface design that does not consider usability fundamentals.
- Listed stock is not consistently updated, leading to inaccurate information about available products.
- The quality of items may be inaccurately presented and (for edible products) the expiry date may be incorrect, due to lack of dedicated administration.
- Delivery time is not clearly specified and/or may be incorrect.
- Inefficient service quality, especially with regards to customer support, which is not always available.
- Payment gateways are not always secured and/or may have bugs.

### Impact Analysis

The above issues have the following negative consequences:

- Users find it difficult and cumbersome to traverse complex system interfaces, wasting their time with actions that should ideally be quick and effortless.
- Mistakenly labeling unavailable products as available may cause issues with orders and therefore lead to unsatisfied customers.
- Inaccurate information about the quality of products may lead to unsatisfied customers and potential legal issues.
- Incorrect delivery information may cause issues when it comes to urgent orders.
- Unsecured payment gateways often lead users to fall into phishing attacks and fraudulent transactions.

## **Desired Outcomes**

- To design an easy-to-use system that helps users find their desired products with minimal time and effort.
- To develop an easy-to-maintain system that can be easily updated with up-to-date stock and order information.
- To provide users with accurate information about the system, its products, and the user's order.
- To equip the system with a satisfactory customer support service that speedily responds to users' problems and queries.
- To implement secure payment gateways that allow users to safely complete their transactions.

## **1.4 Project Objectives**

The project aims to achieve the following objectives by the deadline specified in the contract:

- Integrate all business operations, such as selling products, inventory management, payment, orders delivery, return policy, and customer support into the online shopping system to be available at any time and any place.
- Minimize the cost and time needed by all business operations.
- Reduce chances of mistakes by humans (customers, staff, administrators) by implementing safety measures and checks into the system.
- Provide accurate real-time statistics for the business operations and generate readable, presentable reports.
- Increase security of operations such as payment.
- Faster response times to customer requests, complaints, and feedback.

These project objectives are well aligned with the goals of online shopping businesses, which aim to increase the ROI and customer sales. These objectives prioritize creating an efficient system that's easy to use and stable in the long run. The system is made to be attractive to the customer, simple to do processes through, and provides all the processes needed with minimal need for customer support. This should increase the returns of investment in this system and the sales for the businesses using it to sell products.



## 1.5 DevOps Model

The DevOps iterative model was chosen to meet business objectives in an efficient and rapid manner. Modern tools and solutions are key for a modern system to be developed, deployed, and maintained in the long run. Since the system is intended for a quickly changing market and industry, and the fact that the business itself is fundamentally a large-scale online business; a modern and quick lifecycle such as DevOps is needed to fulfill the project objectives. [1]

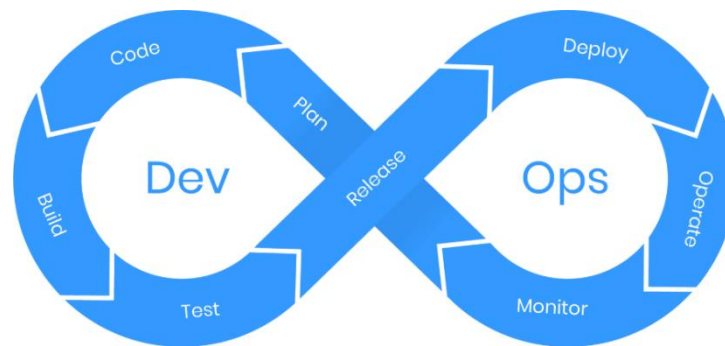


Figure 1DevOps

**DevOps** is a collaborative approach to software development and IT operations that aims to shorten and automate the entire software development lifecycle (SDLC) and provide continuous integration, delivery, and feedback, which ensures fast, reliable, and efficient software development and deployment.

DevOps fosters a culture of collaboration between development (Dev) and operations (Ops) teams, promoting automation, continuous improvement and delivering high-quality software faster.

The iterative phases of the DevOps model are as follows:

- **Plan:** Defining requirements and planning the project by collecting end-user feedback, then creating a project roadmap.
- **Code:** Writing and developing code for the features planned in the previous phase.
- **Build:** Automating builds, generating executables, and preparing the software for deployment.
- **Test:** Verify the quality of the code by testing it in various environments.
- **Release:** Prepare the software for deployment and release it to production.
- **Deploy:** Deploy the software to the production environment in an automated and reliable manner.

- **Operate:** Managing infrastructure, scaling, and monitoring production environments.
- **Monitor:** Continuously monitor the application's performance and infrastructure for any issues or potential improvements.

DevOps involves multiple roles across both development and operations, as well as some new roles that are specific to DevOps:

- **Software Developer:** Writes, commits and tests code frequently and collaborates with DevOps engineers to integrate code into the shared repository.
- **Operations Engineer:** Manages infrastructure, deployment, and monitoring and ensuring system availability and performance. In addition, collaborates with DevOps engineers on deployment, scalability, and system maintenance.
- **DevOps Engineer:** Facilitates collaboration between development and operations, and responsible for implementing automation tools, processes, and promotes continuous improvement.
- **Quality Assurance (QA) Engineer:** Ensures the quality of the software through writing and automating test cases, analyzes test reports and identifies issues for developers. Works closely with developers and DevOps engineers to ensure continuous testing.
- **Security Engineer:** Protects the application and infrastructure from security threats by ensuring compliance with security standards and conducting penetration testing and security audits.
- **Release Manager:** Oversees the release process and coordinates between development, operations, and stakeholders.

Artifacts in DevOps are the essential deliverables created throughout the DevOps lifecycle, facilitating continuous integration and delivery. These include the source code, which developers produce during the coding phase, and build artifacts, such as executables or libraries compiled during the build process. Deployment scripts automate tasks like deployment through tools such as Docker or Kubernetes. Test reports summarize the outcomes of testing, while release notes document the changes in each release, such as new features or bug fixes. Additionally, monitoring dashboards provide real-time insights into system performance, and logs capture system activity for debugging and monitoring purposes.

## Why is DevOps the ideal choice for this system?

The DevOps model has been chosen as the process model for this system because it gives clients faster and better results compared to the other models. Because it is based on process automation, continuous collaboration and sharing of feedback between the Development team and the Operations team, the process of implementing code and testing it becomes much faster, and developers can maintain the system and fix its problems (if found) quickly and easily.

This quality is very important for large-scale systems such as online shopping systems, because the clients prefer to receive speedy results when first ordering development of the system, and speedy maintenance should any problems arise. This is different from other models like the Waterfall Model, where implementation of the system occurs very late into the process, and undetected blunders can have terrible effects, making it unideal for large-scale projects.

Merging both the operations and development teams also helps enhance teamwork and collaboration skills by increasing communication between them, leading to a tighter knit work team where there is stronger trust between everyone involved and less misunderstandings.

## Chapter 2 | System Analysis

### 2.1 Requirements Collection

Requirements collection is an important process in software engineering that aims at gathering all the information needed to understand the deliverables of the project. This process is vital in project management because it enables the project team members to understand the end goal of the project and what the stakeholders require of them.

Requirements collection involves various methods, with each of them having its own advantages and disadvantages:

- **Interviews:** one of the most direct methods, interviews enable analysts and/or project managers to engage with stakeholders in one-on-one conversations to extract detailed insights. However, they may be difficult to schedule and there may be inconsistencies in the information gathered when interviewing individuals.
- **Observing workers:** directly observing workers gives more firsthand and objective measures of the way employees interact with information systems. However, it is time consuming and can cause people to change their normal behavior (upon finding out that they are being observed).
- **Document analysis** helps to uncover existing requirements or constraints by reviewing relevant documentation, such as business plans, regulations, and previous project reports. However, some disadvantages are that the information gathered through documents may be out of date and may contradict information gathered through interviews.
- **Questionnaires:** a method that is often used to collect quantitative responses from large groups of stakeholders. Workshops bring together key stakeholders in a

collaborative setting to brainstorm, discuss, and prioritize requirements in real-time. However, there may be issues with the information collected through this method due to a lack of response time and options usually provided in questionnaires.

A combination of these methods is often used to ensure that all perspectives are considered and that requirements are both complete and aligned with business goals. As interviews are the best tool for gathering detailed information, an interview was conducted with an experienced IT support staff member who oversees developing and maintaining the functionalities of the system, whose experience and position as a stakeholder in the system provided valuable insight. The analyst's skill and the quality of the questions asked directly affect the quality of the information gathered, so the questions and topics to be discussed were prepared beforehand.

<b>Interview outline</b>	
<b>Interviewee</b> Amal A.	<b>Interviewer</b> Amal Abdulhameed
<b>Location:</b> Saar Branch	<b>Appointment:</b> 07-Nov-2024 <b>Start time:</b> 6:00pm <b>End time:</b> 6:23pm
<b>Objectives:</b> <ul style="list-style-type: none"> <li>- Collect detailed information about the current online shopping system and learn more about its background.</li> <li>- Discuss the functionalities of the current system.</li> <li>- Discuss the issues faced in the current system.</li> <li>- Learn more about the interviewee's experiences with the system.</li> <li>- Discuss interviewee's future system development goals</li> </ul>	<b>Reminders:</b> None
<b>Agenda:</b> <ul style="list-style-type: none"> <li>- Background information about the system</li> <li>- Functionalities of the system</li> <li>- Issues with the system</li> <li>- Interviewee's experiences with the system</li> <li>- Future development plans</li> </ul>	<b>Approximate time:</b> <ul style="list-style-type: none"> <li>- 3 minutes</li> <li>- 6 minutes</li> <li>- 6 minutes</li> <li>- 5 minutes</li> <li>- 3 minutes</li> </ul>
<b>General observations:</b> The interviewee gave us some very valuable insight regarding the system and her answers were concise and helpful despite being a little short sometimes. She considers	

the system to be satisfactory despite some of its issues and emphasized its usability and user friendliness many times throughout the interview.

Table 1 Interview outline

<b>Q1:</b> How long have you been using this system, and what kind of software and hardware does it depend on?	<b>Answer:</b> We have been using the system since July 2024. It is built on the Shopify software and utilizes various kinds of hardware such as computer devices, mobile devices, and printers.
<b>Q2:</b> What are the functionalities of your system?	<p><b>Answer:</b> Our system has many functionalities such as “add to cart”, “browse and search for products” and “enter promotional code” functionalities. These are provided by the Shopify software, which is very flexible and easy to use, and offers many great features to implement into our system.</p> <p><b>Observation:</b> Other than those mentioned above, we later observed that the system utilizes many other functionalities such as “log in/register” and “customer reviews” functionalities.</p>
<b>Q3:</b> Are there any issues in the system? How do you manage them?	<b>Answer:</b> Yes, we ran into some problems regarding the system’s language settings. The Shopify software does not provide any language options other than English, so for a while the system was unavailable in Arabic, which caused Arabic-speaking customers some difficulties when using the system. We had to contact experienced freelance software developers to implement an Arabic-language option in the system, and the issue was finally fixed.
<b>Q4:</b> What are your experiences with the system? Do you consider it to be satisfactory?	<b>Answer:</b> Yes, the system is excellent. It is flexible, easy to use, and very reliable. Despite being new to my job when we first started using this system, I found it very easy to familiarize myself with its features and got used to its development and maintenance very quickly.
<b>Q5:</b> Do you have any future development plans in mind for the system? What are they?	<b>Answer:</b> Yes, we have some ideas about what to implement next. We are planning on expanding our shipping services to more countries and adding more language options to enhance the user experience and reach more customers around the world.

	<b>Observation:</b> The interviewee didn't specify many details about their future development plans, or the timeframe by which they were hoping to implement them.
--	---

Table 2 Interview questions

## Requirement Prioritization Criteria

The criteria we used to prioritize requirements are focused on the important and feasible functionality to implement, as in, how integral the functionality is to the experience of online shopping.

Because the main goal of an online shopping system is to provide users with a platform where they can browse, examine, and purchase products, the highest priority was given to the functionalities that allow users to do these things, such as the search for/browse products functionality, the checkout functionality, and the shopping cart functionality.

On the other hand, functionalities like customer reviews, promotional codes, and shipping method selection are not necessary for the main goal of the system, which is to enable users to browse and purchase products. In addition to that, some functionalities like customer reviews and shipping method selection are not feasible to implement early in the project, because they depend on the higher-priority functionalities like the checkout and browsing functionalities.

## 2.2 Functional Requirements

### 2.2.1 Detailed Descriptions

- 1. The user (customer, staff member, or otherwise) must be able to log in, log out, or register into the system.**

#### Input:

- Username
- Password
- Email (for the registration process)

#### Output:

- A successful login response, including user information (name, email, user ID, etc.).
- Error messages (incorrect credentials, missing information).
- A redirect to the homepage or dashboard (if the login credentials were correct).
- Account creation confirmation email (for registration).

### **Processing Logic:**

#### Registration:

- User inputs registration details
- System makes sure the email/username are not already taken.
- User data is stored in the database with password hashing.

#### Login:

- The system makes sure login credentials are correct.
- If the credentials are correct, a session or JWT token is created for the user.
- Session expires after a certain time or upon user logout.

#### Logout:

- The session is invalidated when the user logs out.
- The User is redirected to the homepage or login screen.

## **2. The user must be able to browse products, search by categories or keywords, and view the product's details.**

### **Input:**

- The user's search query (consisting of keywords or the product name)
- Filters (price range, color, size, ratings)
- Sort criteria (price low to high, highest rating, newest products, etc.)

### **Output:**

- A list of products that appropriately match the search criteria.
- A Detailed product page with the image, description, price, and customer reviews of the product

### **Processing Logic:**

#### Search:

- The system takes the user's search query and returns matching results from the database (The query is altered appropriately in case of filters/sorting criteria being provided)

#### Product Details:

- When the user clicks/taps on the product
- The system checks the available amount of the product.

## **3. The user must be able to add, remove and update items in their shopping cart.**

### **Input:**

- The product name/ID
- The product quantity
- Desired action (add/remove/update product quantity)

**Output:**

- The updated shopping cart with the products, quantities, and prices.
- The updated total price calculation.
- The option to proceed to checkout.

**Processing Logic:**Add to Cart:

- The user specifies a product and the quantity they want to add of said product.
- If the product is already in the cart, the quantity is updated.
- The cart is saved in a session or the user's account if he/she is logged in.

View Cart:

- The system retrieves and displays the current cart contents, including product name, quantity, price per item, and total price.

Update Cart:

- The user can change the quantity of items and remove items from his/her cart.
- The total price is recalculated based on the updated cart.
- The user must be able to

**4. The user must be able to check out and enter his/her payment and shipping information.****Input:**

- The user's shipping address (name, address, phone number)
- The user's payment information (credit/debit card information, Benefit, etc.)

**Output:**

- Order confirmation with overview of order details.
- Email or SMS message with order confirmation.
- A success/error message regarding payment processing.

**Processing Logic:**Enter/update Shipping Information:

- The user provides his/her delivery address and contact details.
- The system validates the address format and availability.
- Enter/update payment information.
- The user selects a payment method (credit/debit card, BenefitPay, cash).
- The system securely processes the payment through a payment gateway.

Payment Processing:

- The system communicates with the payment gateway to authorize and process the payment.
- The payment status (approved/declined) is returned.

Order Confirmation:



- The order is placed if the payment is successful.
- A confirmation email/SMS message is sent to the user, and the order status is updated in the database.

## **5. The user must be able to review and rate products.**

### **Input:**

- The user's Product rating
- The user's written review

### **Output:**

- Display of the average product rating.
- Display of customer reviews.

### **Processing Logic:**

#### Submit Review:

- The user provides a rating and/or a written review.
- The review is saved and associated with the product in the database.

#### View Reviews:

- The average rating is calculated and displayed on the product page along with a list of customers reviews.
- Reviews may be sorted by date or rating.

## **6. The user must be able to choose between shipping methods and track his/her deliveries.**

### **Input:**

- The user's shipping address
- The user's preferred shipping method

### **Output:**

- The estimated delivery time.
- The tracking number of the shipment and shipping provider
- Shipping status updates (shipped/out for delivery/delivered)

### **Processing Logic:**

#### Shipping Method Selection:

- User selects a shipping method during checkout.
- The system calculates shipping costs and delivery times.

#### Tracking:

- The system provides tracking details for the user to track their shipment once the order is shipped.
- Delivery status updates are sent to the user.

**7. The user must be able to apply promotional codes and discounts.**

**Input:**

- The promo code is supplied by the user at checkout.

**Output:**

- The total (discounted) price after applying the promotional code.
- An error message if the promotional code is invalid.

**Processing Logic:**

Apply Promo Code:

- The user enters a promotional code during checkout.
- The system validates the promotional code.
- The system applies the appropriate discount and calculates the new (discounted) price.

Invalid Code:

- An error message is displayed if the promotional code is invalid.

**8. The user must be able to contact customer support and communicate with the staff.**

**Input:**

- The user's questions about products, orders, returns, and/or other issues.
- The user's email, phone number, live chat messages

**Output:**

- A support request ticket with a unique ID is generated.
- The customer support's response and resolution to the issue

**Logic Processing:**

- The system provides a ticketing system for managing customer inquiries.
- For common issues like shipping details, automated responses are sent.
- For more complex or unresolved issues, a customer service agent is dispatched to the user.
- History of customer interactions is available for viewing.

**9. The supplier must be able to add/remove products.**

**Input:**

- Product details (name, category, description, price, available quantity)
- Desired updates (adding or deleting new products, updating the quantity)

**Output:**

- The updated inventory list
- Alerts for low stock items

**Processing Logic:****Add Products:**

- The supplier can add new products to the catalog, including details like price, description, and stock level.

**Update Product Information:**

- details of existing products (price changes, stock adjustments, etc.) are.

**Stock Management:**

- The system updates stock quantities in real time as orders are placed.
- Low stock alerts are sent to admins when stock falls below a set threshold.

**10. The supplier must be able to generate reports and view statistics.****Input:**

- Parameters for reports (e.g., product sales, inventory levels).

**Output:**

- Detailed reports on product performance and inventory status.

**Processing Logic:**

- The supplier selects metrics and timeframe for generating reports.
- The system processes and compiles the data into a report format.

**11. The admin must be able to authorize or block users (customers or suppliers).****Input:**

- User account details (username, email, user type).
- Desired action (authorize/block).

**Output:**

- Updated user status (authorized/blocked).
- Notifications sent to users about their account status.

**Processing Logic:**

- The admin reviews user accounts based on activity or reports.
- User accounts can be authorized for access or blocked as necessary.

## **12. The admin must be able to manage the system and generate reports.**

### **Input:**

- Parameters for generating reports (e.g., sales data, user activity, system usage).

### **Output:**

- Detailed reports showing insights into system performance, user engagement, and financial data.

### **Processing Logic:**

- The admin selects the type and scope of report to generate.
- The system processes and compiles the data into a readable format.

## **13. The admin user must be able to view and update customer orders.**

### **Input:**

- Order ID
- Order status (pending/shipped/delivered)
- Tracking details (for shipped orders)

### **Output:**

- A list of orders with status (pending/shipped/delivered)
- The updated order statuses and shipping information
- Notifications for shipment and delivery

### **Processing Logic:**

#### View Orders:

- The admin views all incoming orders, filterable by status, date, customer, etc.

#### Update Order Status:

- The admin can change the order status (e.g., from "Pending" to "Shipped").
- Admin may update tracking numbers and shipping provider information.

#### Order Fulfillment:

- Once shipped, a notification (email/SMS) is sent to the customer.
- Admin can generate invoices for each order.

### 2.2.2 Prioritization

1. **The user (customer, staff member, or otherwise) must be able to log in, log out, or register into the system.**

**Priority:** High, because user authentication is important for customers to make purchases, and important for admins to be able to access their system privileges. This is vital for both functionality and security purposes.

2. **The user must be able to browse products, search by categories or keywords, and view the product's details.**

**Priority:** High, because it is important for users to be able to find products and their information in an online shopping system. Without it, the system would be confusing to the user, making finding and purchasing the desired products harder for the user and thus rendering the system purposeless.

3. **The user must be able to add, remove, and update items in his/her shopping cart.**

**Priority:** High, because the user needs the shopping cart feature to be able to view and manage his/her selected items before checkout. It is also important for viewing item prices and conversion rates.

4. **The user must be able to enter (and update) shipping information and make payments when checking out.**

**Priority:** High, because payment and shipping information are essential in completing the order, as without paying and specifying their shipping and delivery information, the customer will be unable to successfully complete their purchase, and thus the system would be purposeless. This feature also needs to be secure and reliable to prevent any further complications with the order.

5. **The supplier must be able to add/remove products.**

**Priority:** High, because suppliers are responsible for maintaining the product catalog and ensuring products are updated and available for customers. This is critical for the system's functionality.

6. **The admin must be able to authorize or block users (customers or suppliers).**

**Priority:** High, because admins need to manage access to the system and maintain security by addressing unauthorized or problematic users.

7. **The admin user must be able to view and update customer orders.**

**Priority:** High, because admins need to monitor and manage orders to ensure the system processes them correctly and customer satisfaction is maintained.

8. **The user must be able to review and rate products.**

**Priority:** Medium, because while reviews and ratings are very useful for customers to learn more about the products and hear others' experiences with them, they are not integral to the system's operations and main functionalities. Therefore, this

feature can be implemented after the basic transactional features are implemented.

9. **The user must be able to choose between shipping methods and track his/her deliveries.**

**Priority:** Medium, because selecting preferred shipping options is not essential for the basic operation of the system, so it can be implemented later.

10. **The admin must be able to manage the system and generate reports.**

**Priority:** Medium, because although it helps admins oversee the system's operations, it is not immediately essential for the system's basic functionality and can be implemented after core features.

11. **The supplier must be able to generate reports and view statistics.**

**Priority:** Medium, because while this functionality is not essential for the basic operation of the system, it provides suppliers with useful insights into their sales and inventory, which aids in decision-making.

12. **The user must be able to apply promotional codes and discounts.**

**Priority:** Medium, because despite promotional codes being a good tool to boost sales and increase conversions, the system can function without them, and therefore the feature can be implemented after the other, higher-priority features.

13. **The user must be able to contact customer support and communicate with the staff.**

**Priority:** Low, because a customer support feature is not required for the system to function, as online shopping systems can facilitate customer-staff communication through other methods such as emails, thus it is not necessary for the basic operation of the system.

## 2.3 Non-Functional Requirements

Non-functional requirements (NFRs) specify criteria that can be used to judge the operation of a system, rather than specific behaviors. For an online shopping system, NFRs are crucial for ensuring the system is reliable, efficient, secure, and user-friendly.[2]

### Performance:

- The website should load within 2-3 seconds.
- The checkout process should not take more than 5 minutes.
- The system should handle a high number of concurrent users without noticeable performance degradation.
- The suppliers should be able to upload products and manage inventory quickly and efficiently.

- The admins should well-document the codebase to facilitate easy maintenance and upgrades.

**Security:**

- Each user should have a unique email and a strong password.
- The user and admin should be able to reset their password.
- The user should receive an email when registering or an order is made.
- The user should make payments through BENEFIT payment gateway.
- The system should ensure that only authorized users can view and modify their personal information.
- The data related to products and transactions must be secure from unauthorized access.

**Usability:**

- The system should be intuitive and easy to navigate, enabling users to perform actions with minimal effort.
- The user should be able to access the website through any device.
- The system should provide clear instructions and feedback for user actions.
- The suppliers should be able to easily update product information, track orders, and manage returns.
- For admins, the development tools and environments should be user-friendly and efficient.

**Reliability:**

- The system must be available 99.9% of the time, so it can be accessible by any customer, supplier or admin.
- Implement backup and recovery systems to prevent data loss and ensure continuity.
- The website should not have critical failures.
- Provide a responsive support system to address user issues promptly, at non-official hours.

**Compatibility:**

- The website must be compatible with all modern browsers (Chrome, Firefox, Safari, Edge) and versions.
- The website must be responsive to all hardware, operating systems, devices and kept up to date.

- The system should support integration with payment methods and shipping services.

## 2.4 Data Flow Diagrams (DFD)

### 2.4.1 Context Level DFD

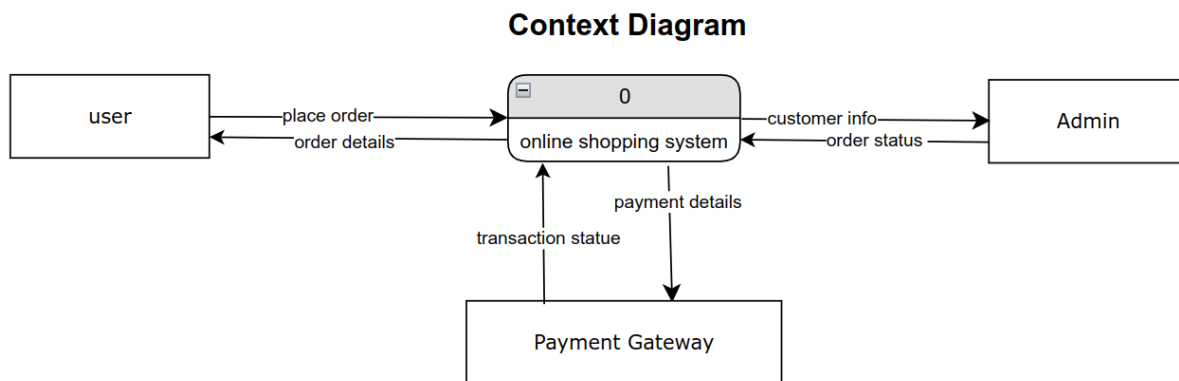


Figure 2 context level DFD



## 2.4.2 Level 0 DFD

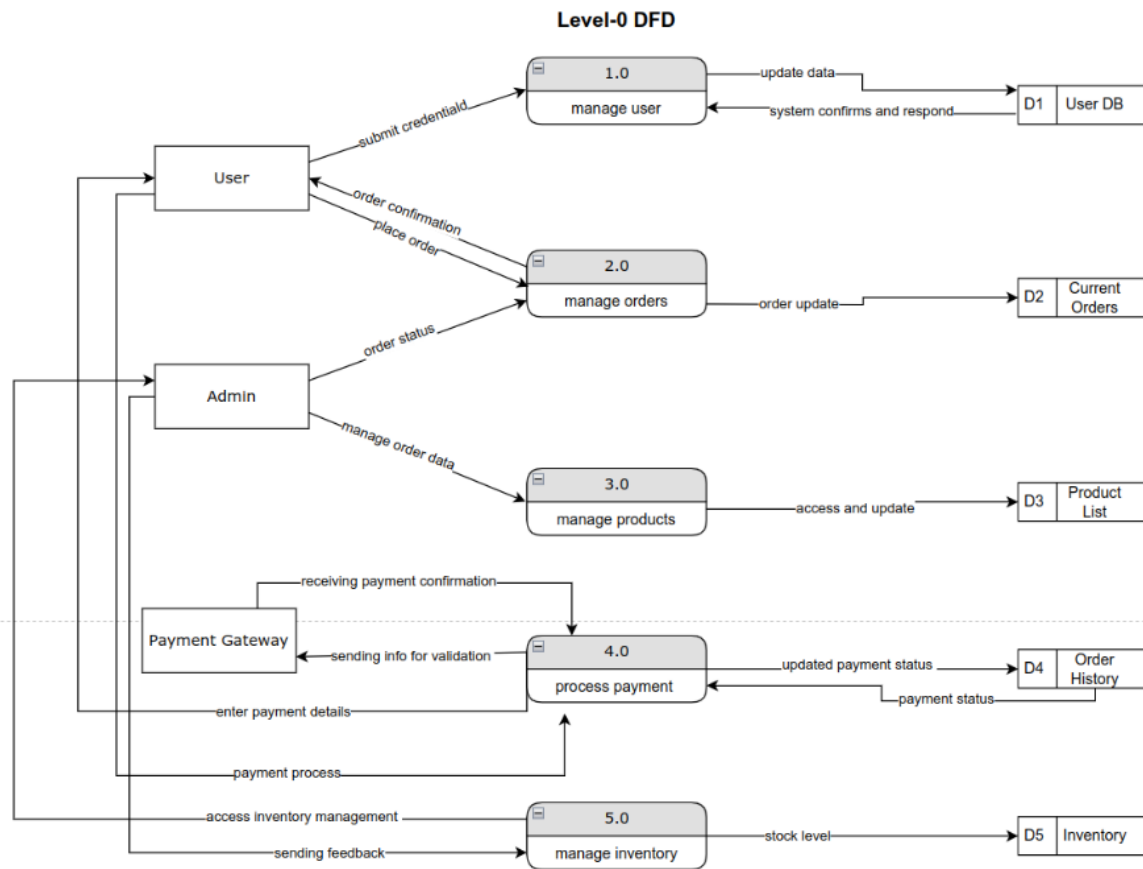


Figure 3 level 0 DFD

## 2.4.3 Level 1 DFDs

### 2.4.3.1 Process 1.0

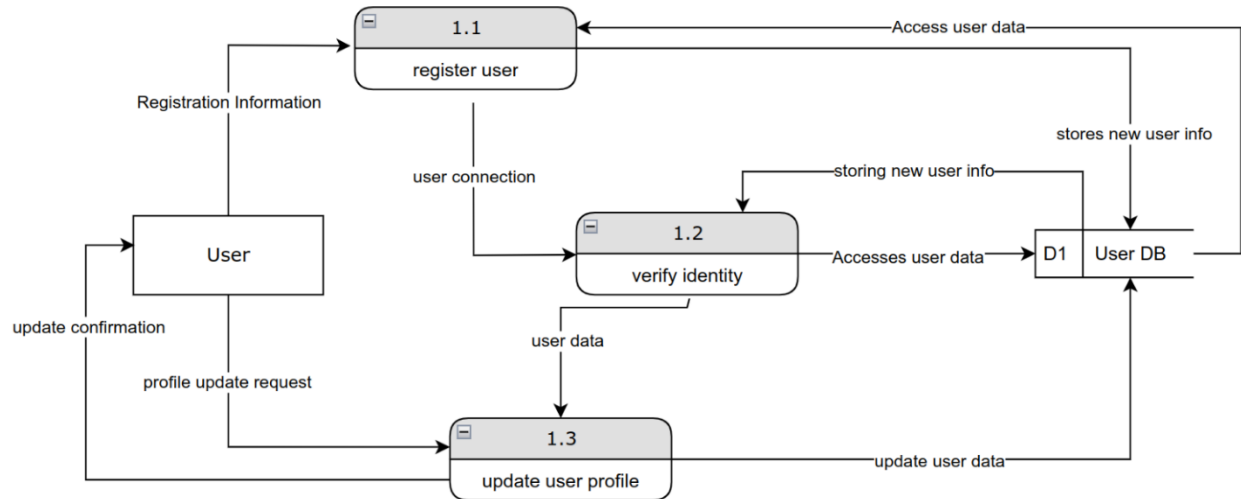
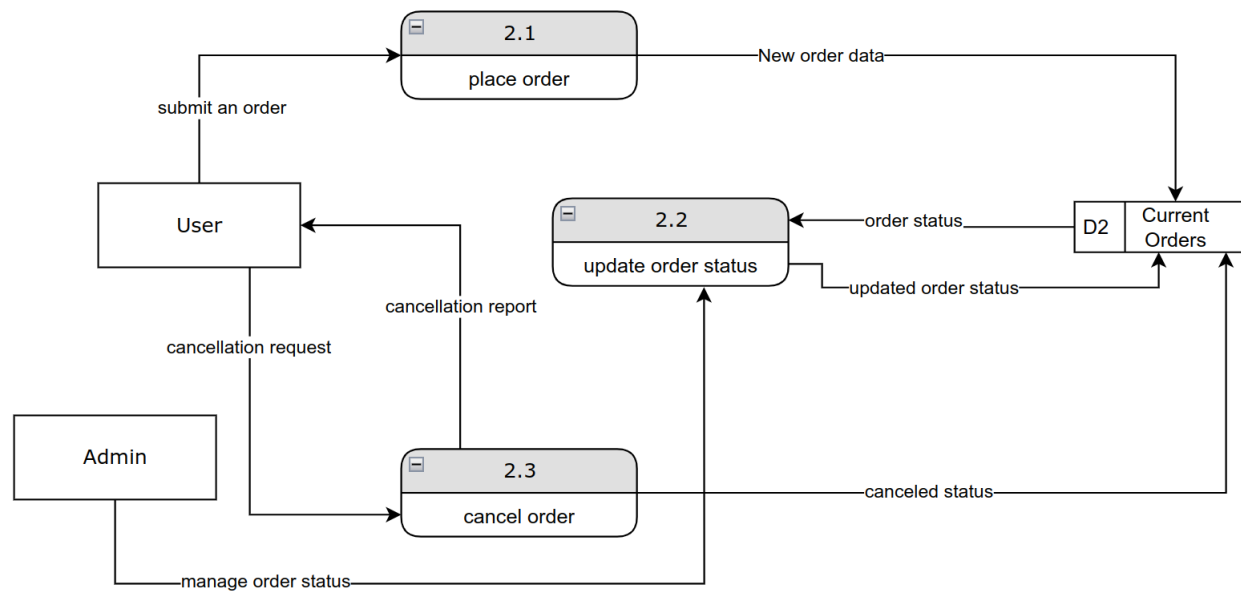
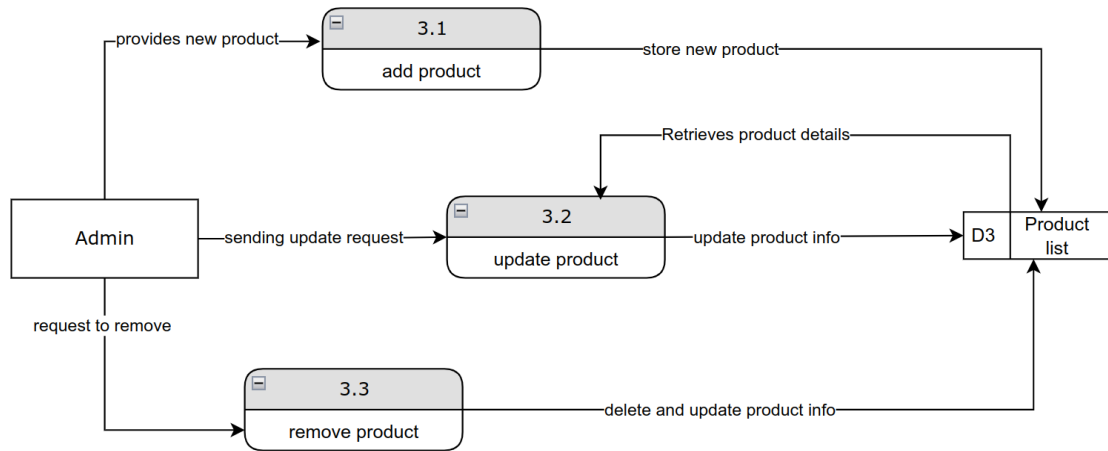


Figure 4 level 1DFD

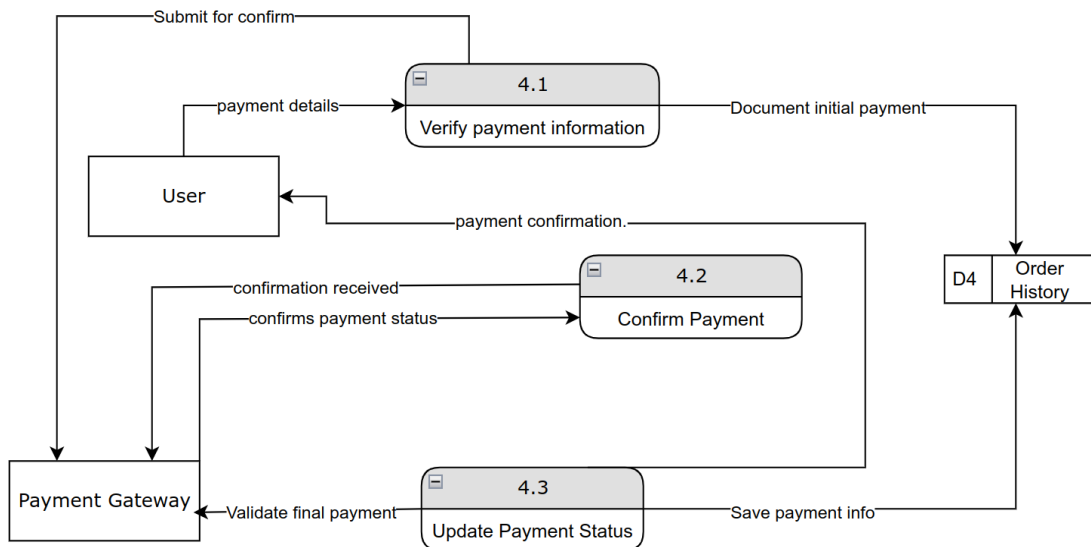
### 2.4.3.2 Process 2.0



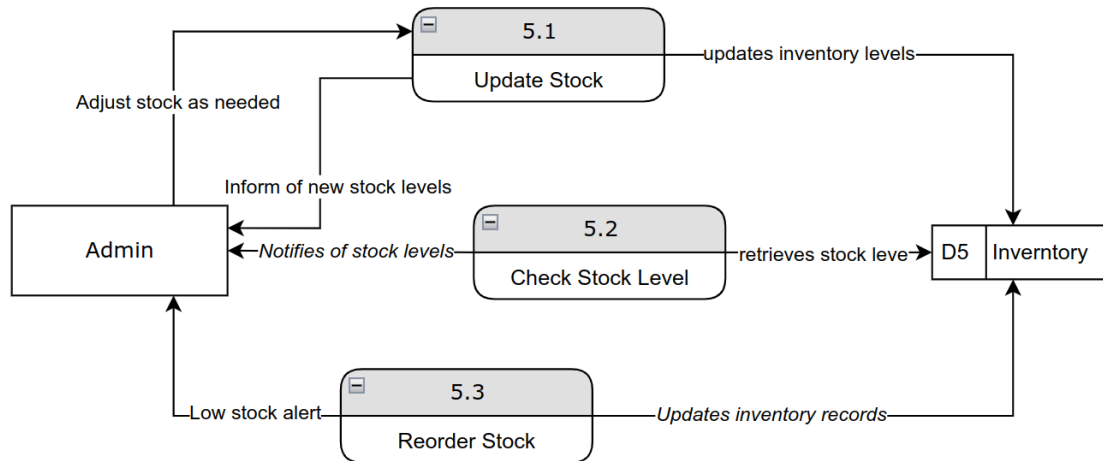
### 2.4.3.3 Process 3.0



### 2.4.3.4 Process 4.0



### 2.4.3.5 Process 5.0



## 2.5 Process Specification

### 2.5.1 Structured English

Structured English is a method of writing in plain English, organized into blocks with indentation and capitalization to visually display the hierarchical structure of logic. It relies on structured logic and is particularly useful for describing processes that involve conditional statements or repeated actions, such as IF-THEN-ELSE decisions. This approach simplifies complex process flows by breaking them down into clear, easily understandable steps.

Process ID	1.0
Process Name	Manage User
Description	To manage user account creation, updates, and validation
Input	User credentials, account details
output	User confirmation, error messages
Logic	<ul style="list-style-type: none"> <li>• <b>READ</b> User credentials.</li> <li>• <b>IF</b> credentials are valid <b>THEN</b> <ul style="list-style-type: none"> <li>• <b>UPDATE</b> user information in User DB</li> <li>• <b>DISPLAY</b> confirmation message to User.</li> </ul> </li> <li>• <b>ELSE</b> <ul style="list-style-type: none"> <li>• <b>DISPLAY</b> error message</li> </ul> </li> </ul>

Table 3 Process 1.0 Details.

Process ID	2.0
Process Name	Manage orders
Description	To handle the creation, update, and status of orders placed by users.
Input	Order details from the User
output	Order Confirmation, updated order status
Logic	<ul style="list-style-type: none"> <li>• <b>READ</b> Order Details from User</li> <li>• <b>ADD</b> Order Data to Records</li> <li>• <b>DISPLAY</b> Order Confirmation to User</li> <li>• <b>IF</b> order update is requested <b>THEN</b> <ul style="list-style-type: none"> <li>◦ <b>UPDATE</b> order information in Records.</li> <li>◦ <b>DISPLAY</b> updated order status to User</li> </ul> </li> </ul>

Table 4 Process 2.0 Details.

Process ID	3.0
Process Name	Manage Products
Description	To manage the product catalog, including additions, updates, and deletions.
Input	Product details from Admin
Output	Updated Product List
Logic	<ul style="list-style-type: none"> <li>• <b>READ</b> Product Details</li> <li>• <b>IF</b> new product <b>THEN</b> <ul style="list-style-type: none"> <li>◦ <b>ADD</b> product to Product List</li> </ul> </li> <li>• <b>IF</b> product update requested <b>THEN</b> <ul style="list-style-type: none"> <li>◦ <b>UPDATE</b> Product Information in Product List</li> </ul> </li> <li>• <b>IF</b> product removal requested <b>THEN</b> <ul style="list-style-type: none"> <li>◦ <b>DELETE</b> product from Product List</li> </ul> </li> </ul>

Table 5 Process 3.0 Details.

Process ID	4.0
Process Name	Process Payment
Description	To validate and confirm payment information through a gateway.
Input	Payment details from User
Output	Payment confirmation, updated payment status
Logic	<ul style="list-style-type: none"> <li>• <b>RECEIVE</b> Payment Details from User</li> <li>• <b>SEND</b> Payment Details to Payment Gateway for validation.</li> <li>• <b>IF</b> Payment Confirmation <b>RECEIVED THEN</b> <ul style="list-style-type: none"> <li>◦ <b>UPDATE</b> Payment Status in Order History</li> <li>◦ <b>DISPLAY</b> payment confirmation to User.</li> </ul> </li> <li>• <b>ELSE</b></li> </ul>

	o <b>DISPLAY</b> error message to User
--	--

Table 6 Process 4.0 Details.

Process ID	5.0
Process Name	Manage Inventory
Description	To monitor and adjust inventory levels based on stock and orders.
Input	Inventory adjustments
output	Updated inventory levels
Logic	<ul style="list-style-type: none"> <li>• <b>READ</b> Inventory Levels</li> <li>• <b>IF</b> stock level is low <b>THEN</b> <ul style="list-style-type: none"> <li>o <b>GENERATE</b> reorder request.</li> </ul> </li> <li>• <b>UPDATE</b> Inventory Records</li> </ul>

Table 7 Process 5.0 Details.

## 2.5.2 Decision Tables

Complex tasks are represented using tables. Highlighted cells represent the action to be taken depending on the conditions present in the system.

Conditions / Courses of Action	Rules			
	1	2	3	4
Username	Exists		Doesn't Exist	
Password	Valid	Invalid	Valid	Invalid
Register User				
Display Error Message				
Login User				

Table 8 Process 1.0 Decision Table (Manage User)

Conditions / Courses of Action	Rules			
	1	2	3	4
Average daily sales	X (Calculated field)			
Today's product sales	Less than X		More than X	
Product stock level	Less than X	More than X	Less than X	More than X
Do not restock				
Restock X amount				
Restock 2X amount				

Table 9 Process 5.0 Decision Table (Manage Inventory)

## 2.6 Entity-Relationship Diagram (ERD)

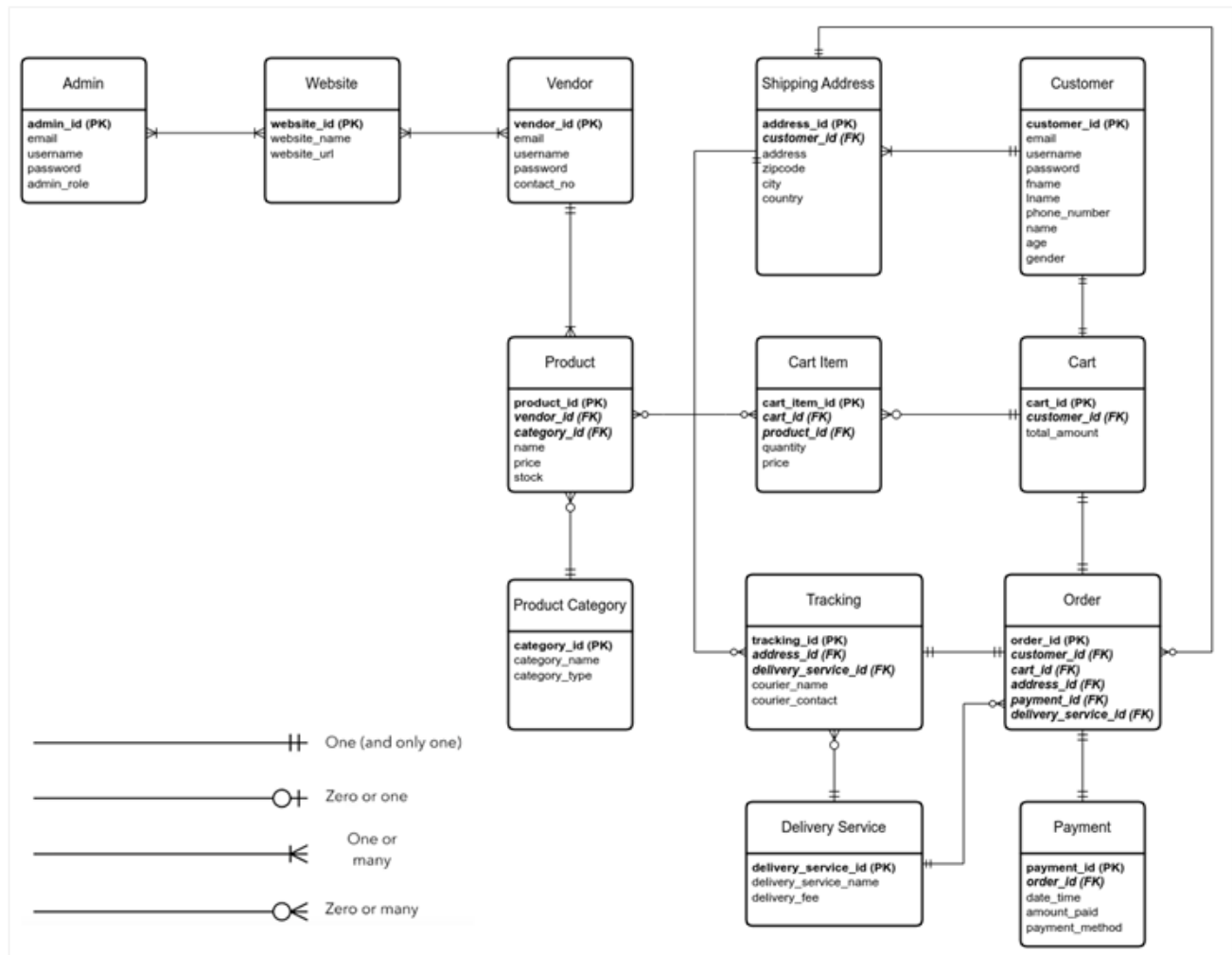


Figure 5 Entity-relationship diagram

ERD diagrams represent the schema of the system's database. Each entity represents a table to be created inside the database, while the attributes are the columns of that table.

Primary keys (a unique attribute that defines an instance of the entity) are **bold**.

Foreign keys (an attribute that relates one instance to another instance in another tables) are **bold and italic**.

Relationships between entities are represented using lines between them, with special notation to represent cardinality (one or many) and modality (optional or mandatory).

## Chapter 3 | Object-Oriented Analysis

### 3.1 Use Case Diagram

A UML (Unified Modeling Language) use case diagram visually represents the interactions between users (actors) and a system, illustrating the system's functionalities (use cases) and how users engage with them. It helps identify the requirements and scope of the system. [3]

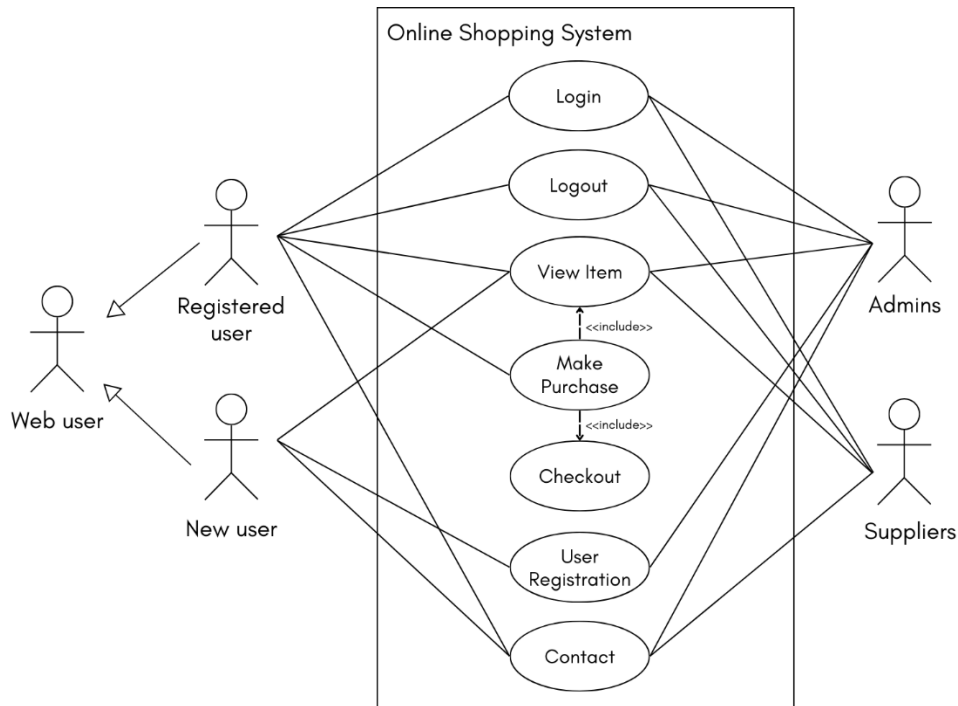


Figure 6 UML use case diagram

It represents the use case diagram of an online shopping system. A new customer can view items but if he wants to make a purchase, he must register. A previously registered customer, admins and suppliers can login and logout. Also, registered customers can browse items, make purchases, proceed to checkout and payment, and contact the concerned parties in case of query or error.



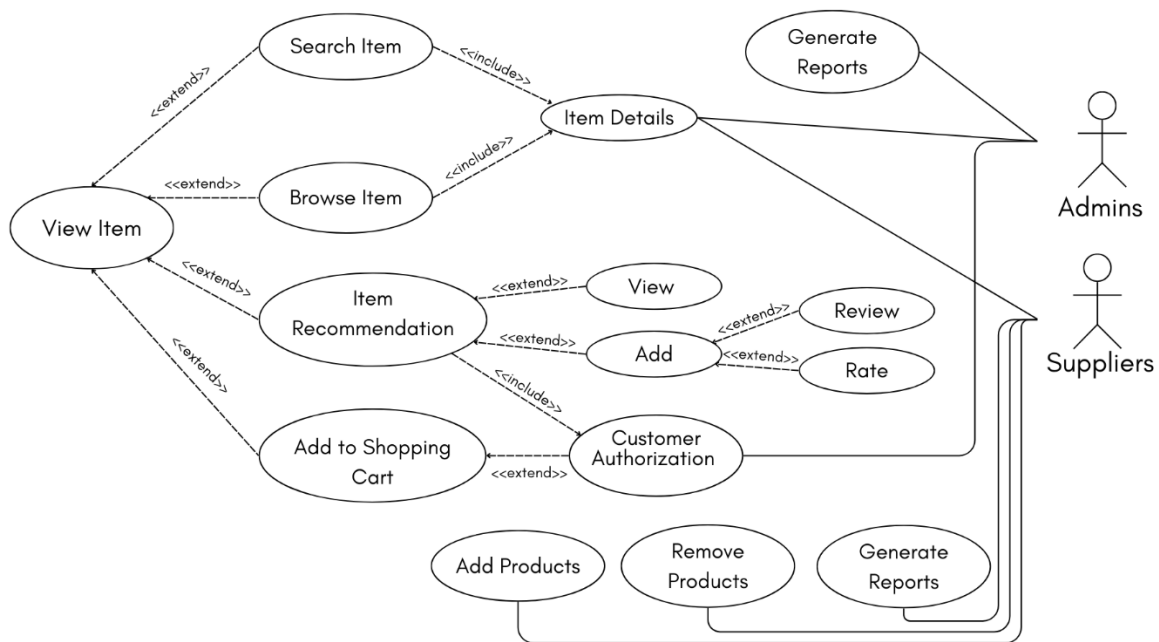


Figure 7 View Item Use Case

The figure shows that the customer can search/browse for an item, add or view a recommendation, and add an item to the shopping cart. Also, the supplier can add/remove products and generate reports. As well as the admin can generate reports.

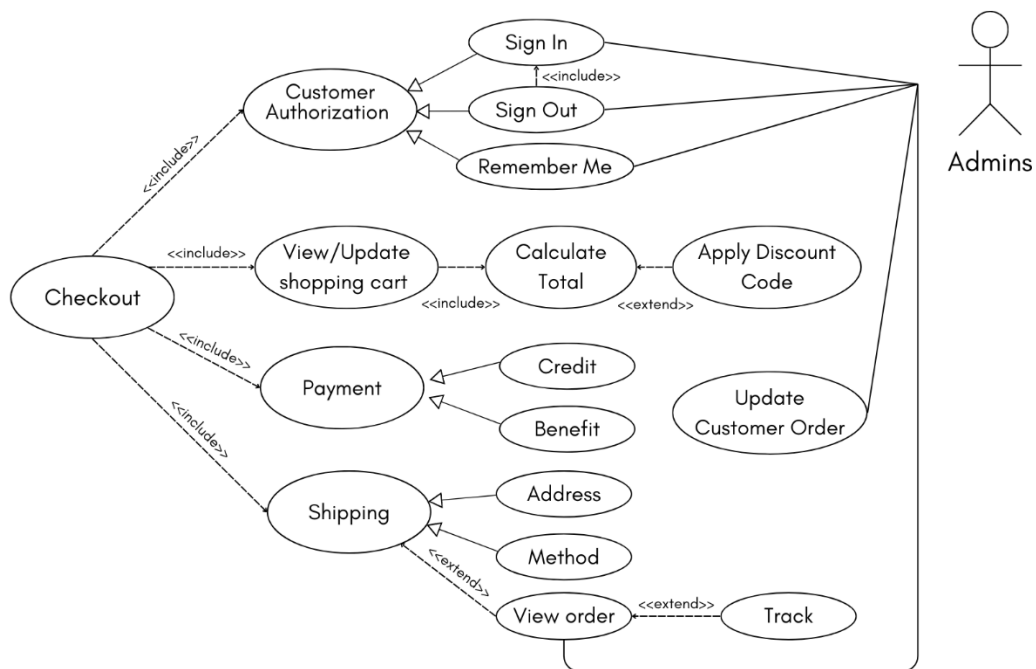


Figure 8 Checkout Use Case

The Figure shows that the customer should be registered to update the shopping cart and apply a discount code to calculate the total amount. Also, the customer can choose the payment and shipping method, and he can track his order. While admins can view and update the customers' orders.

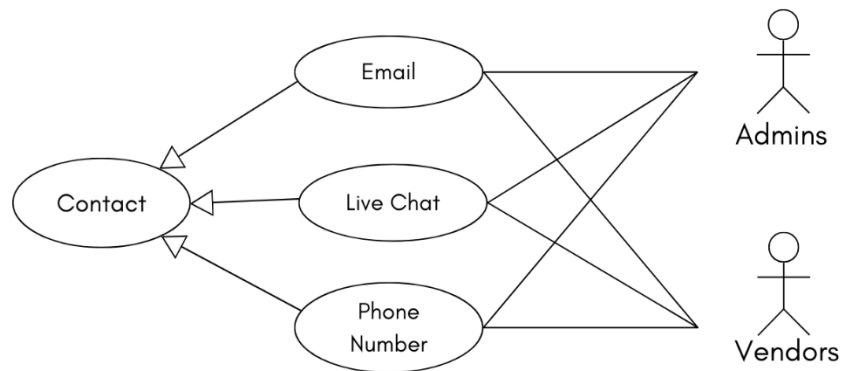


Figure 9 Contact Use Case

The figure shows that the customer can contact any of the parties concerned to get the necessary assistance and answers to inquiries.

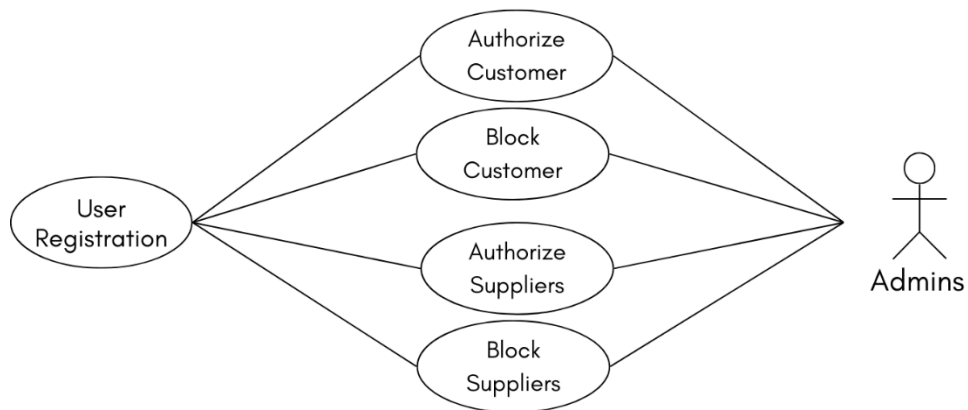


Figure 10 User Registration Use Case

The figure shows that the admin can Authorize/Block customers and suppliers.

### 3.2 State Transition Diagram (STD)

The diagram of the state machine is a behavior diagram that illustrates discrete behavior by finite state transformations of a portion of designed system. [4]

#### 3.2.1 STD for customers

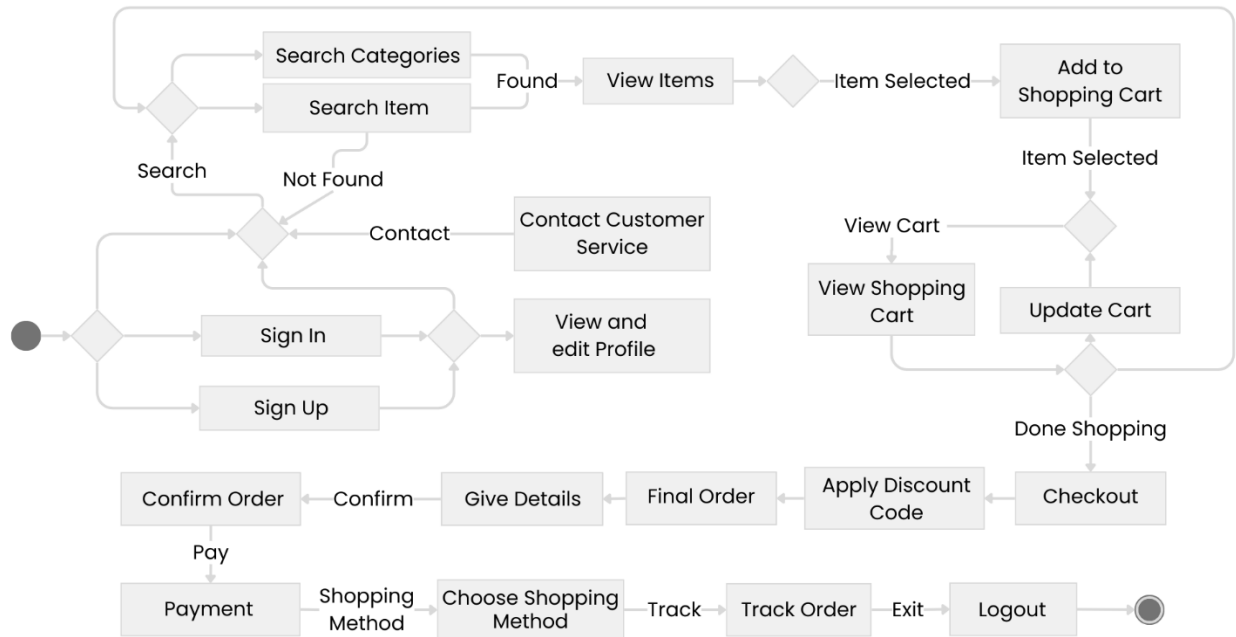


Figure 11 State transition diagram for customers

#### 3.2.2 STD for Suppliers

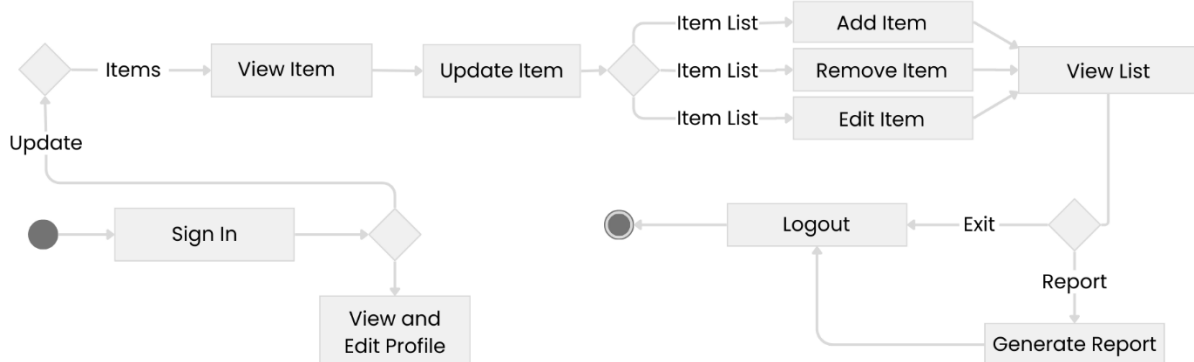


Figure 12 State transition diagram for suppliers

### 3.2.3 STD for Admins

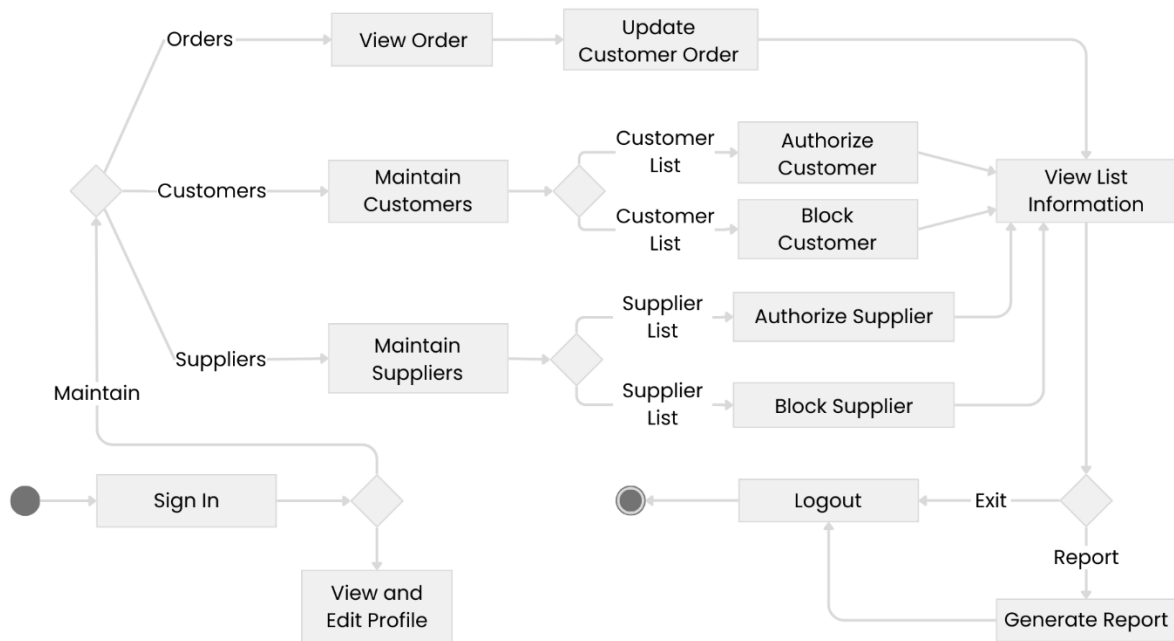


Figure 13 State transition diagram for admins

### 3.3 Class Diagram

The system will follow an object-oriented design, this means that the primary drivers of the system functionalities will be objects constructed from classes.

The following class diagram (Figure 8) illustrates the structure of this system.

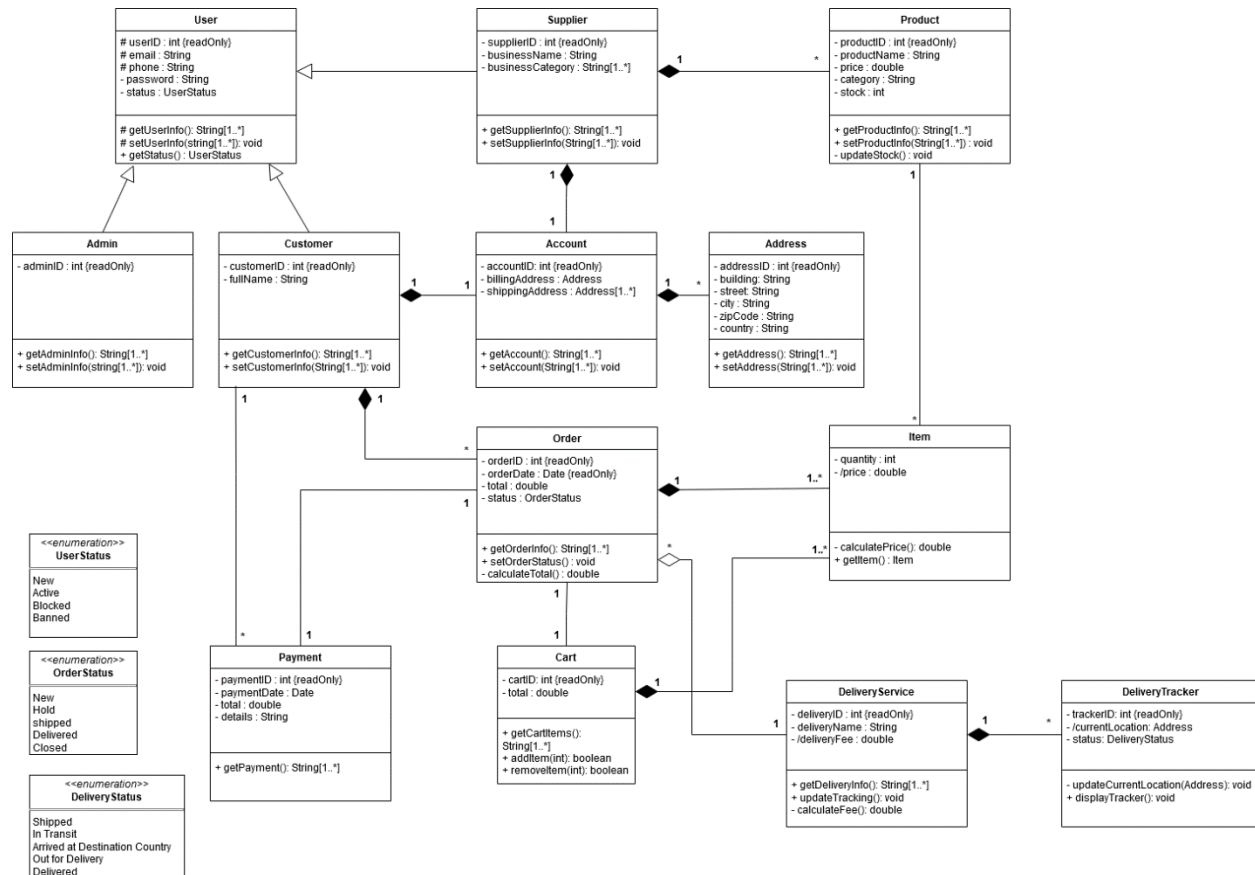


Figure 14 Class diagram of the system

### 3.4 Sequence Diagram

The sequence diagram is an interaction diagram that illustrates the sequence of messages between objects. It consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

### 3.4.1 Customers Sequence Diagram

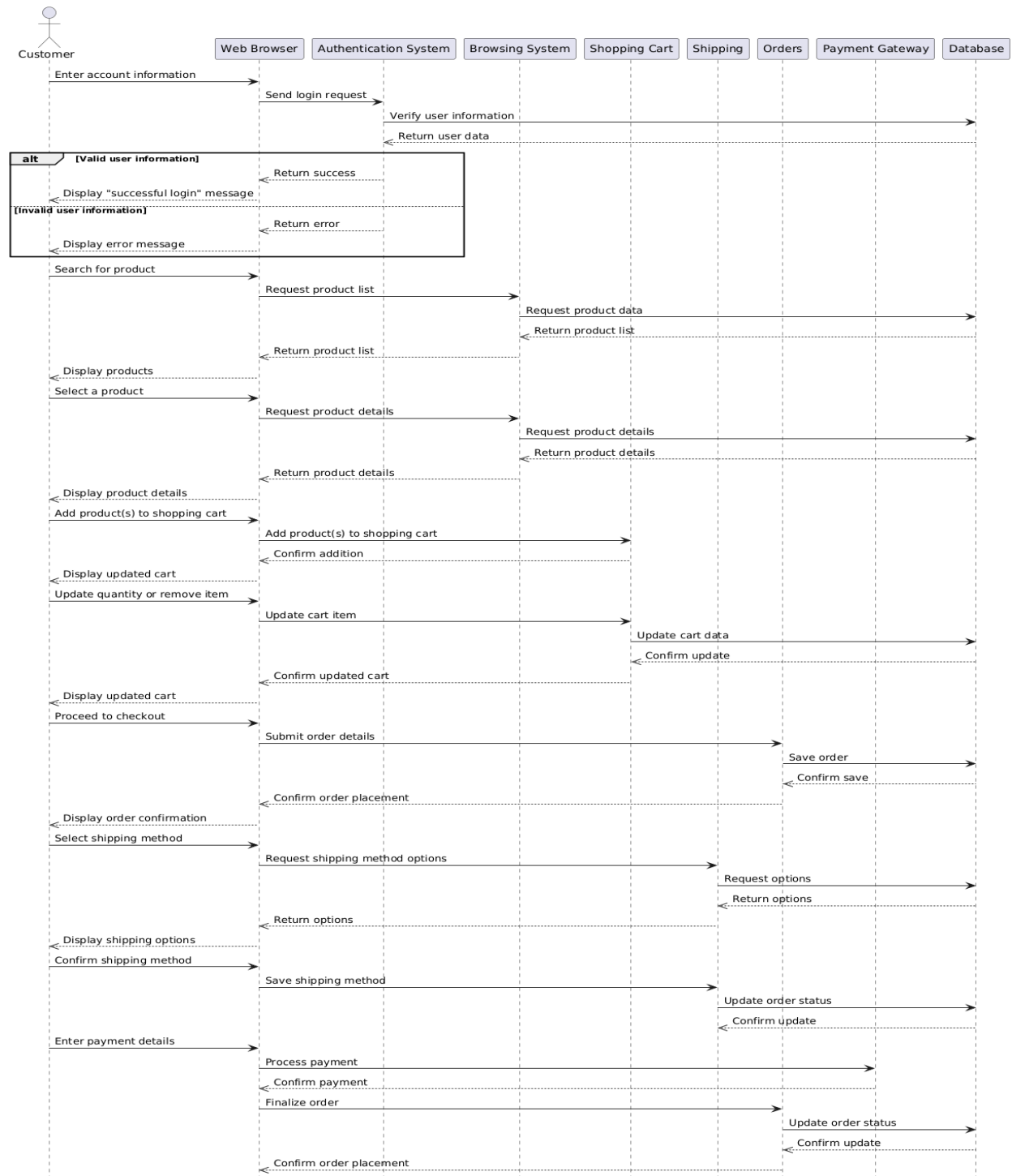


Figure 15 Sequence diagram for customer

### 3.4.2 Admins Sequence Diagram

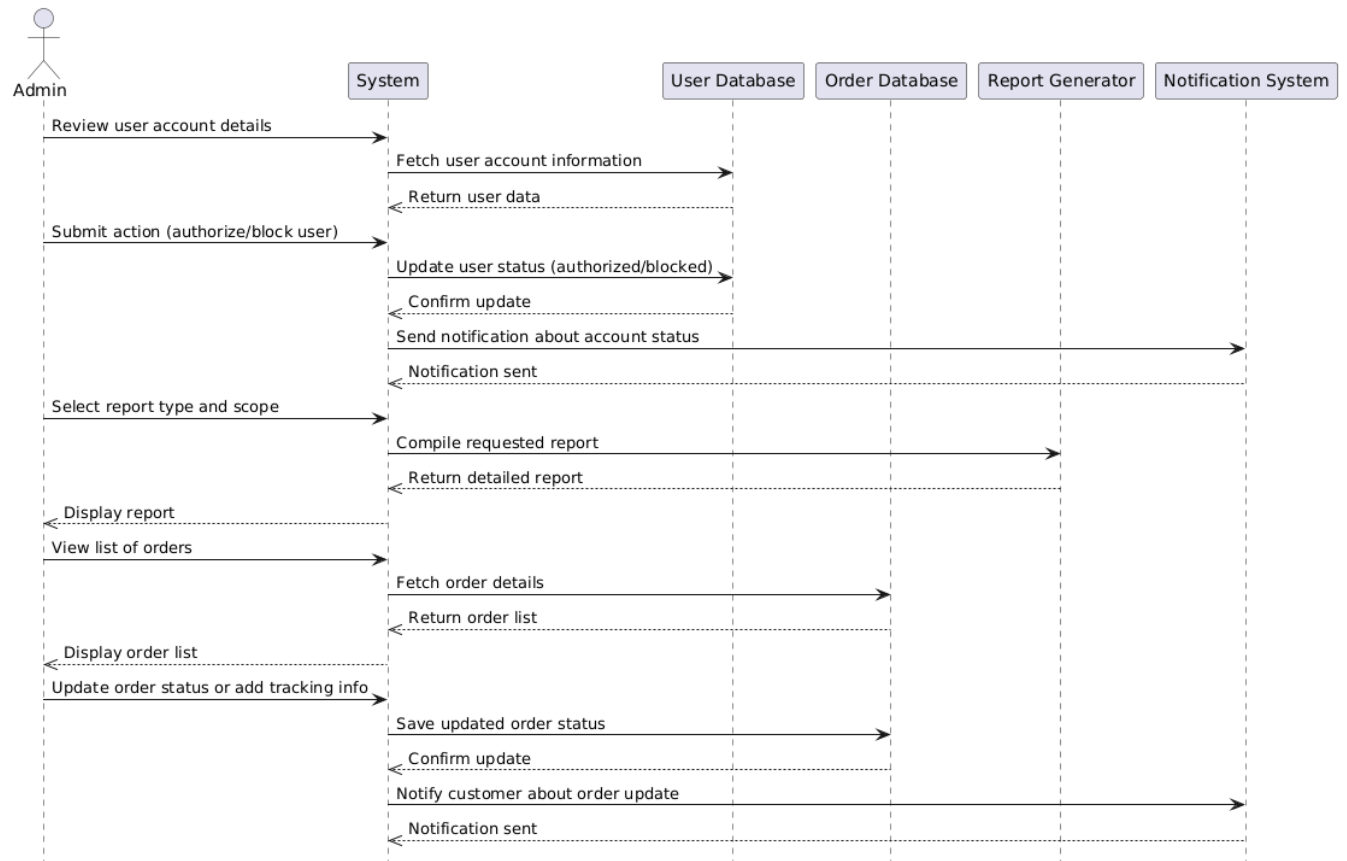


Figure 16 Sequence diagram for admin

### 3.4.1 Suppliers Sequence Diagram

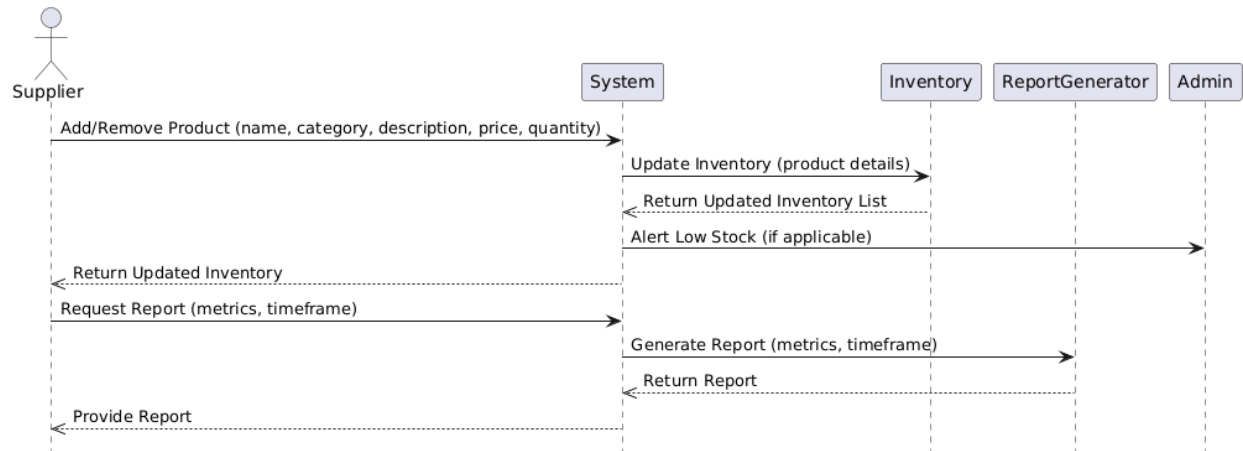


Figure 17 Sequence diagram for suppliers

## 3.5 Activity Diagram

Activity diagrams are behavioral diagrams that depict the movement of activities or processes within a system, highlighting control flow and decision-making phases. They use symbols like swimlanes, control flows, action nodes, and decision nodes.

### 3.5.1 User Login and Registration Activity Diagram

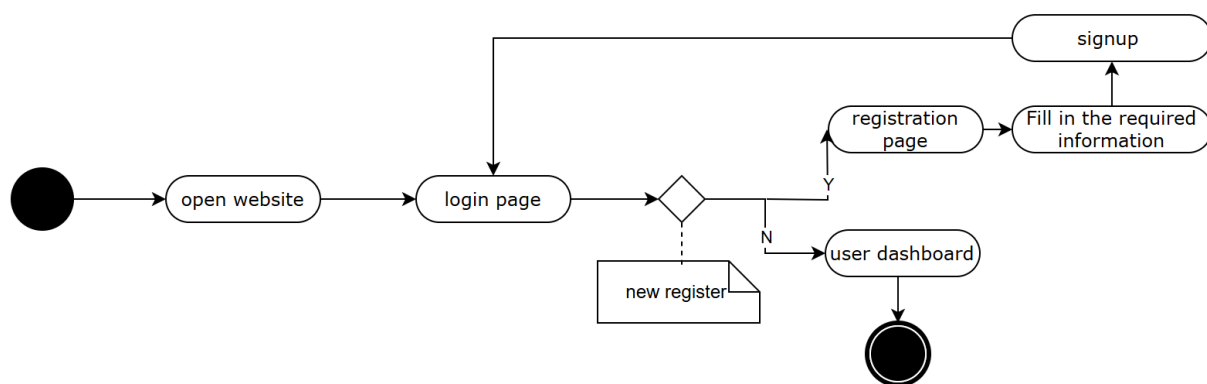


Figure 18 User login and registration Activity Diagram



### 3.5.2 Browsing and Searching Activity Diagram

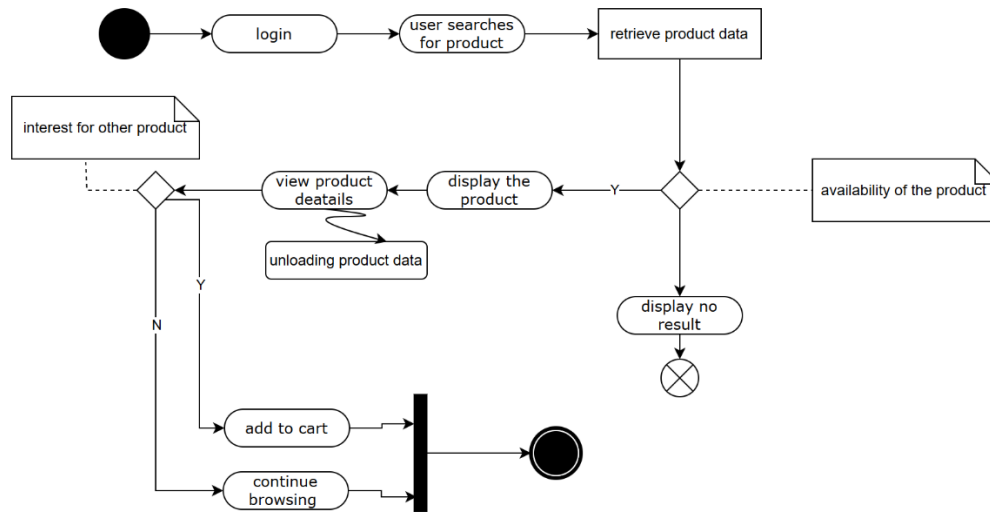


Figure 19 Browsing and Searching Activity Diagram

### 3.5.3 Adding Items and Checkout Activity Diagram

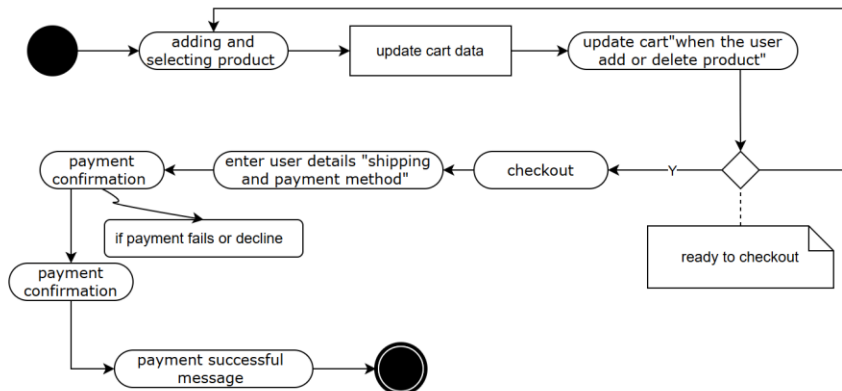


Figure 20 Adding items to cart and checkout Activity Diagram

### 3.5.4 Supplier Activity Diagram

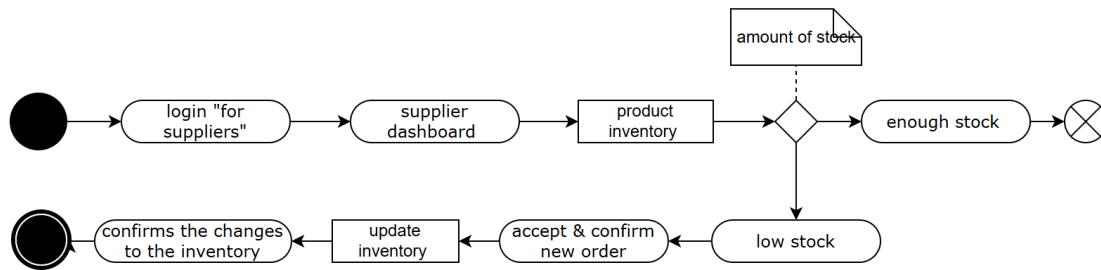


Figure 21 Supplier Activity Diagram

### 3.5.5 Admin Activity Diagram

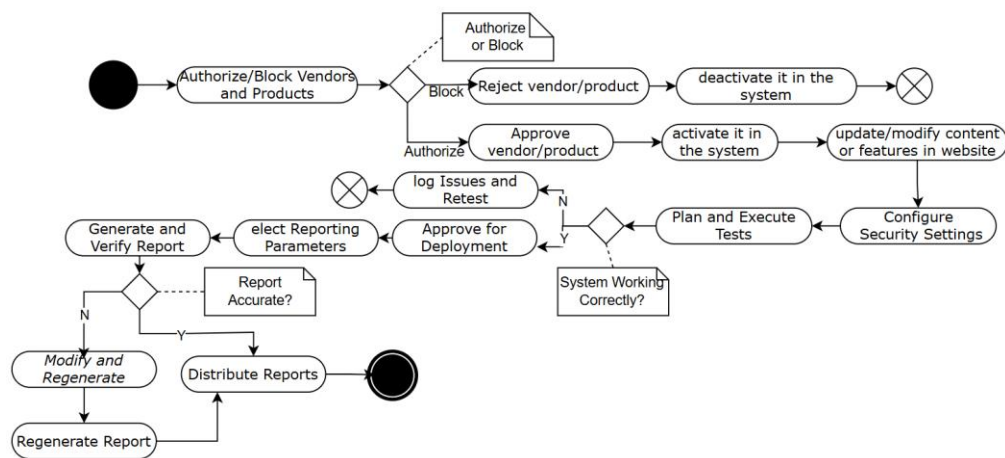


Figure 22 Admin Activity Diagram

## References

[1] **“What is DevOps?”** 23 5 2021. [Online]. Available:

<https://www.betsol.com/blog/what-is-devops/>.

[2] **“Non-Functional Requirements in Software Engineering”** 4 10 2024. [Online].

Available: [Non-Functional Requirements in Software Engineering - GeeksforGeeks](#).

[3] **“Use Case Diagram - Unified Modeling Language (UML)”** 14 10 2024. [Online].

Available: [Use Case Diagram - Unified Modeling Language \(UML\) - GeeksforGeeks](#)

[4] **“State Machine Diagram”**. [Online]. Available: [UML State Machine Diagrams - Overview of Graphical Notation](#)