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

Online Shopping System

Phase Number: 2

Phase Title: System Analysis Document

Due Date: 11/11/2024

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

2.1 Requirements Collection ... 4

2.2 Functional Requirements ... 7

2.2.1 Detailed Descriptions ... 7

2.2.2 UML Use Case Diagram ... 14

2.2.3 Prioritization ... 16

2.3 Non-Functional Requirements ... 18

2.4 Data Flow Diagrams (DFD) ... 21

2.4.1 Context Level DFD ... 21

2.4.2 Level 0 DFD ... 21

2.4.3 Level 1 DFDs ... 22

2.4.3.1 Process 1.0 ... 22

2.4.3.2 Process 2.0 ... 22

2.4.3.3 Process 3.0 ... 23

2.4.3.4 Process 4.0 ... 23

2.4.3.5 Process 5.0 ... 24

2.5 Process Specification ... 25

2.5.1 Structured English ... 25

2.5.2 Decision Tables ... 27

2.6 Entity-Relationship Diagram (ERD) ... 28

Table of Figures

Figure 1 illustrates a UML use case diagram. 14
Figure 2: View Item Use Case 15
Figure 3: Checkout Use Case.................... 15
Figure 4: Contact Use Case 16
Figure 5: Context Level DFD.................... 21
Figure 6: Level 0 DFD.................... 21
Figure 7: Level 1 DFD: Process 1.0 22
Figure 8: Level 1 DFD: Process 2.0 22
Figure 9: Level 1 DFD: Process 3.0 23
Figure 10: Level 1 DFD: Process 4.0 23
Figure 11: Level 1 DFD: Process 5.0 24
Figure 12: Entity-Relationship Diagram (ERD) 28

Table 1: Interview outline.................... 5
Table 2: Interview questions..................... 6
Table 3: Process 1.0 Details. 25
Table 4: Process 2.0 Details. 25
Table 5: Process 3.0 Details. 26
Table 6: Process 4.0 Details. 26
Table 7: Process 5.0 Details. 26
Table 8: Process 1.0 Decision Table (Manage User) 27
Table 9: Process 5.0 Decision Table (Manage Inventory).................... 27

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.

Table 1: Interview outline

Interview outline	
Interviewee Amal A.	**Interviewer** Amal Abdulhameed
Location: Saar Branch	**Appointment:** 07-Nov-2024 **Start time:** 6:00pm **End time:** 6:23pm
Objectives: - Collect detailed information about the current online shopping system and learn more about its background - Discuss the functionalities of the current system - Discuss the issues faced in the current system - Learn more about the interviewee's experiences with the system - Discuss interviewee's future system development goals	**Reminders:** None
Agenda: - Background information about the system - Functionalities of the system - Issues with the system - Interviewee's experiences with the system - Future development plans	**Approximate time:** - 3 minutes - 6 minutes - 6 minutes - 5 minutes - 3 minutes
General observations: 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 2: Interview questions.

Q1: How long have you been using this system, and what kind of software and hardware does it depend on?	**Answer:** 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.
Q2: What are the functionalities of your system?	**Answer:** 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. **Observation**: Other than those mentioned above, we later observed that the system utilizes many other functionalities such as “log in/register” and “customer reviews” functionalities.
Q3: Are there any issues in the system? How do you manage them?	**Answer:** 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 experience freelance software developers to implement an Arabic-language option in the system, and the issue was finally fixed.
Q4: What are your experiences with the system? Do you consider it to be satisfactory?	**Answer:** Yes, the system is excellent, in my opinion. It is flexible, easy to use, and very reliable. Despite being relatively 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.
Q5: Do you have any future development plans in mind for the system? What are they?	**Answer:** 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. **Observation:** The interviewee didn’t specify many details about their future development plans, or the timeframe by which they were hoping to implement them.

Requirement Prioritization Criteria:

The criteria we used to prioritize requirements are focused on the important and feasible functionality is 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:

- 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)

<u>Product Details:</u>

- 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:

<u>Add to Cart:</u>

- 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

<u>View Cart:</u>

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

<u>Update Cart:</u>

- 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:

<u>Enter/update Shipping Information:</u>

- 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.

<u>Payment Processing:</u>

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

<u>Order Confirmation:</u>

- 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 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.

6. The admin should 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:

- Admin 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.

7. 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 customer reviews
- Reviews may be sorted by date or rating

8. **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.

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

Input:

- The promo code 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:

<u>Apply Promo Code:</u>

- 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

<u>Invalid Code:</u>

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

10. 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

2.2.2 UML 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.

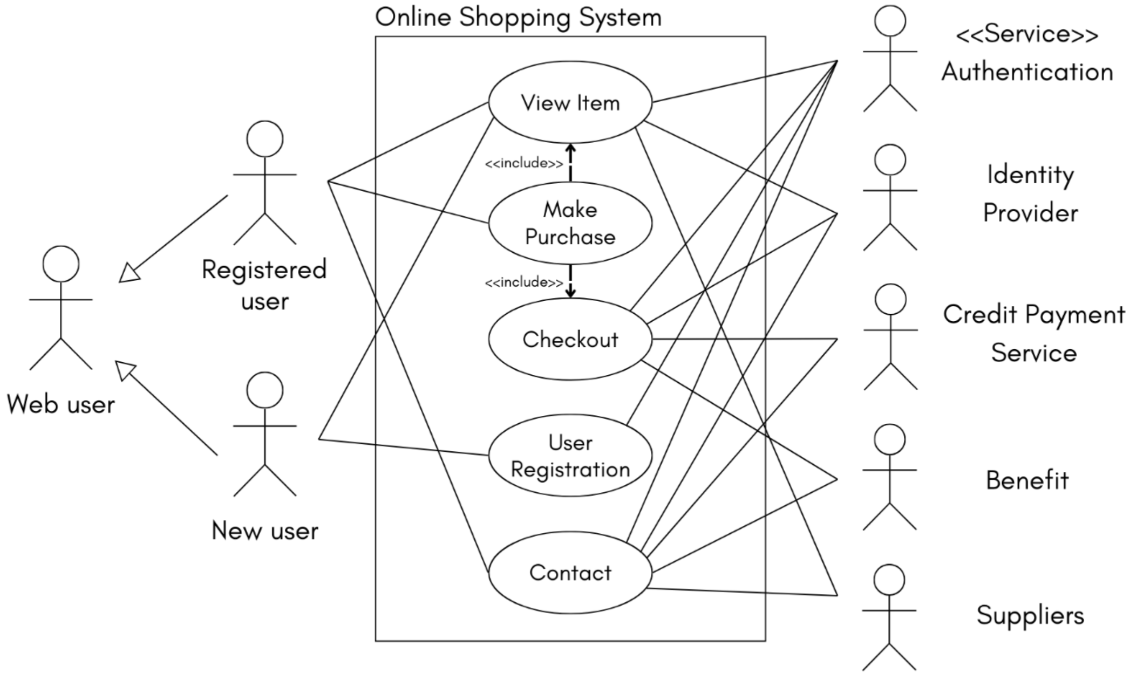

Figure 1 illustrates a UML use case diagram.

It represents the use case diagram of an online shopping system. It represents one of the main actors, the customer. A new customer can view items but if he wants to buy, he must register. A previously registered customer can browse items, add to cart, proceed to checkout and payment, and contact the concerned parties in case query or error.

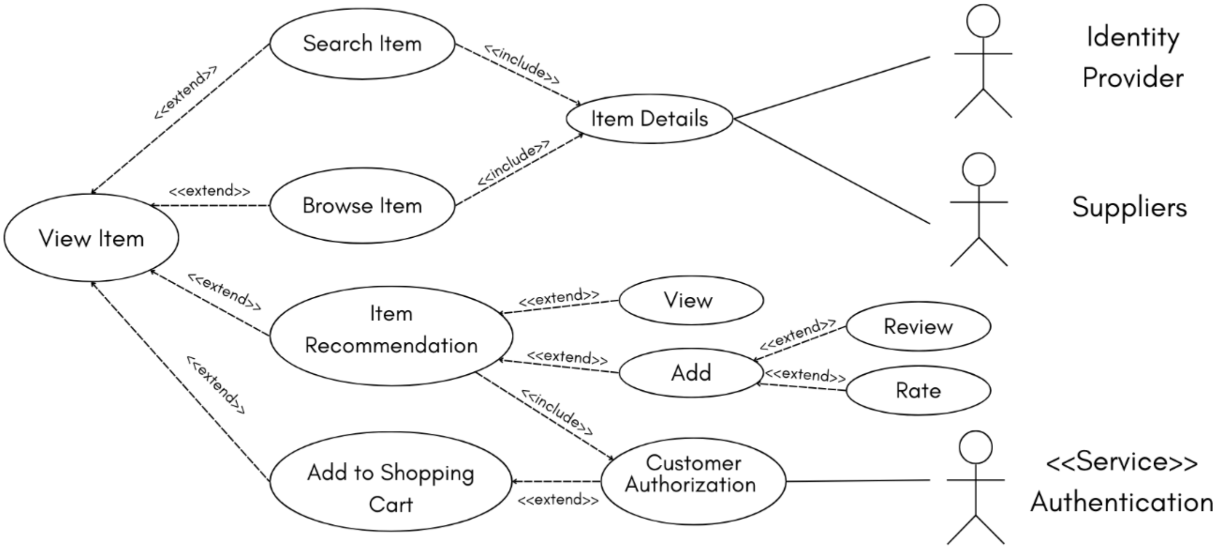

Figure 2: View Item Use Case

The figure shows that the customer can search or browse for an item, add a recommendation, view a previous recommendation, and add an item to the shopping cart.

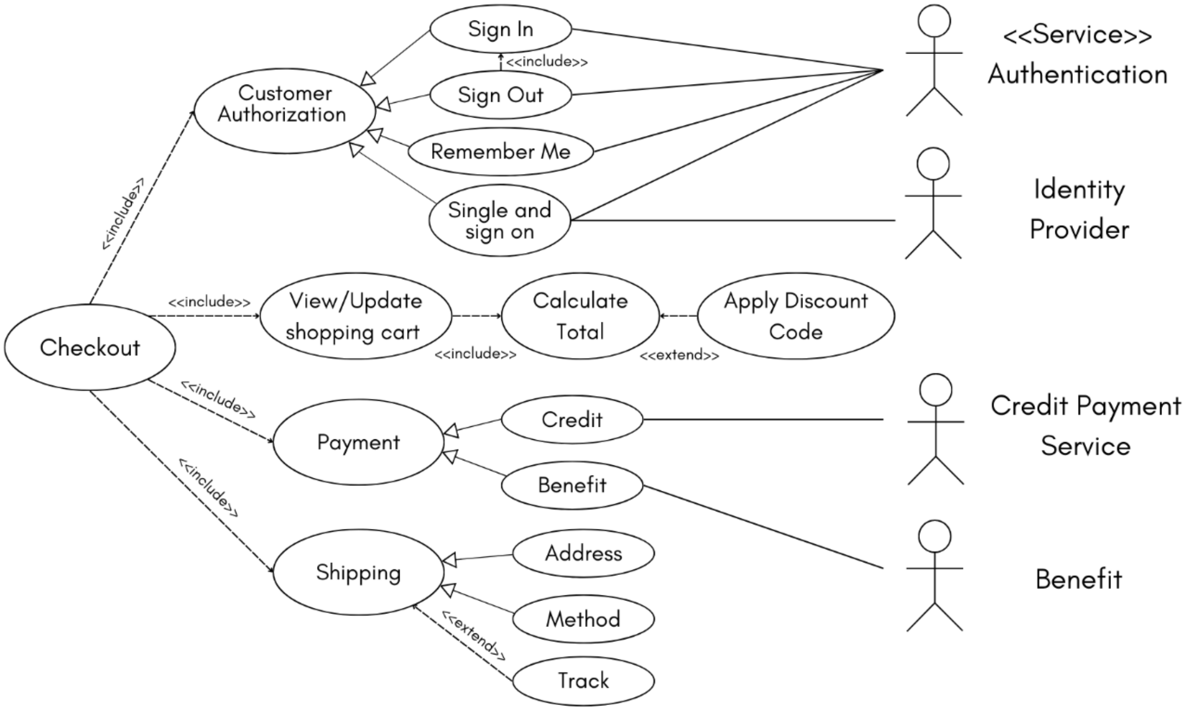

Figure 3: Checkout Use Case

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.

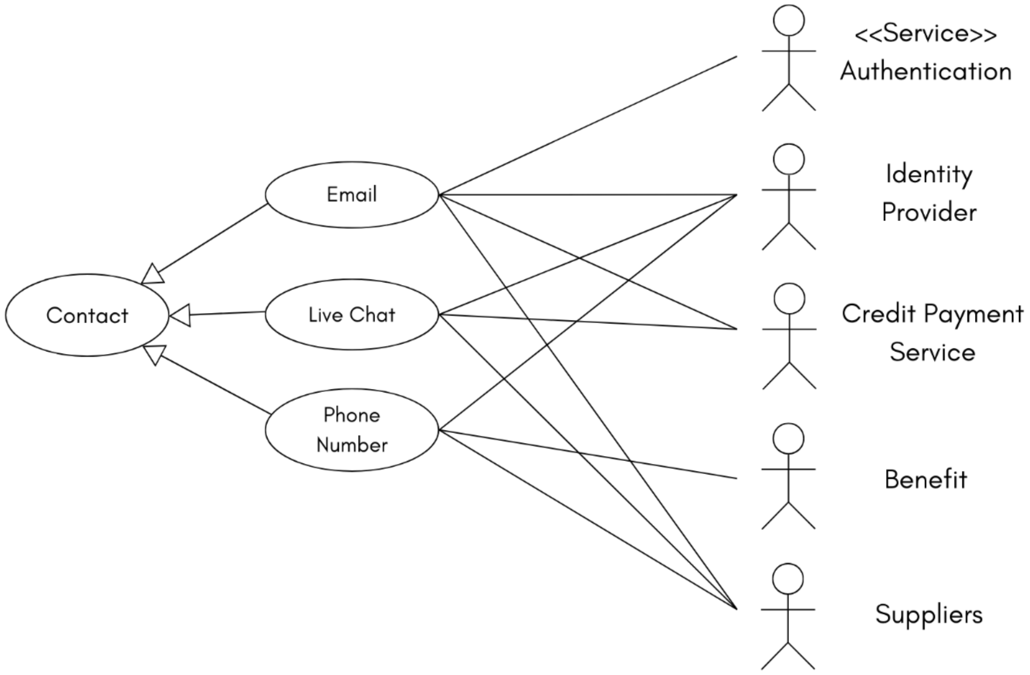

Figure 4: Contact Use Case

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

2.2.3 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 admin must be able to view and update customer orders.**
   Priority: high, because the admins need to have control over the inventory, customer service, and overall operations of the system, to ensure that the system is processing orders correctly and that everything in it is working correctly.
6. **The admin must be able to add/remove products.**
   Priority: medium, because although it is important for managing the product catalogue, it doesn't directly immediately affect customer experience unless the catalogue is limited or has not been updated. Therefore, implementing this feature can wait until the other, more important features are put in place.
7. **The user must be able to review and rate products.**
   Priority: medium, because while reviews and rating 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.

8. **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.
9. **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 increasing conversions, the system can function without them, and therefore the feature can be implemented after the other, higher-priority features.
10. **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.

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.

- For developers, the codebase should be well-documented to facilitate easy maintenance and upgrades.

Security:

- The developers should design the system by implementing firewalls, secure coding practices, and regular security audits.
- 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.
- For suppliers, 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 supplier should be able to easily update product information, track orders, and manage returns.
- For developers, 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 user, supplier or developer.
- 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 third-party payment gateways and shipping services.

2.4 Data Flow Diagrams (DFD)

2.4.1 Context Level DFD

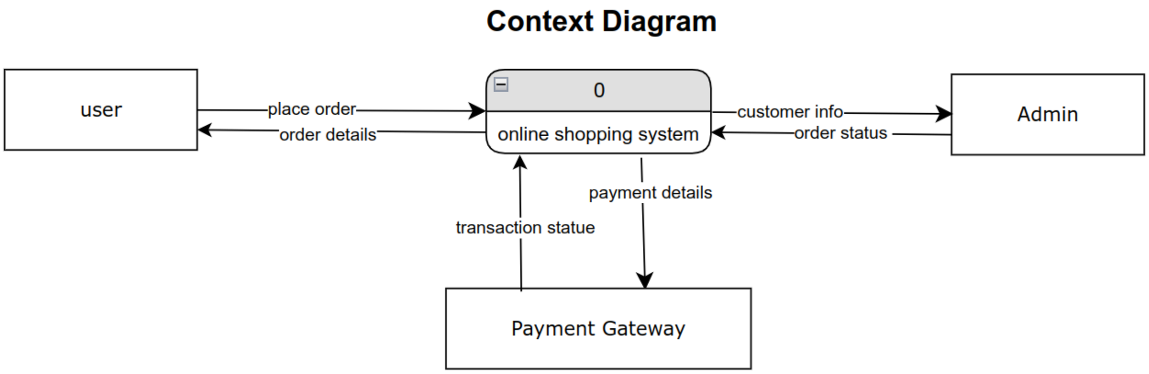

Figure 5: Context Level DFD

2.4.2 Level 0 DFD

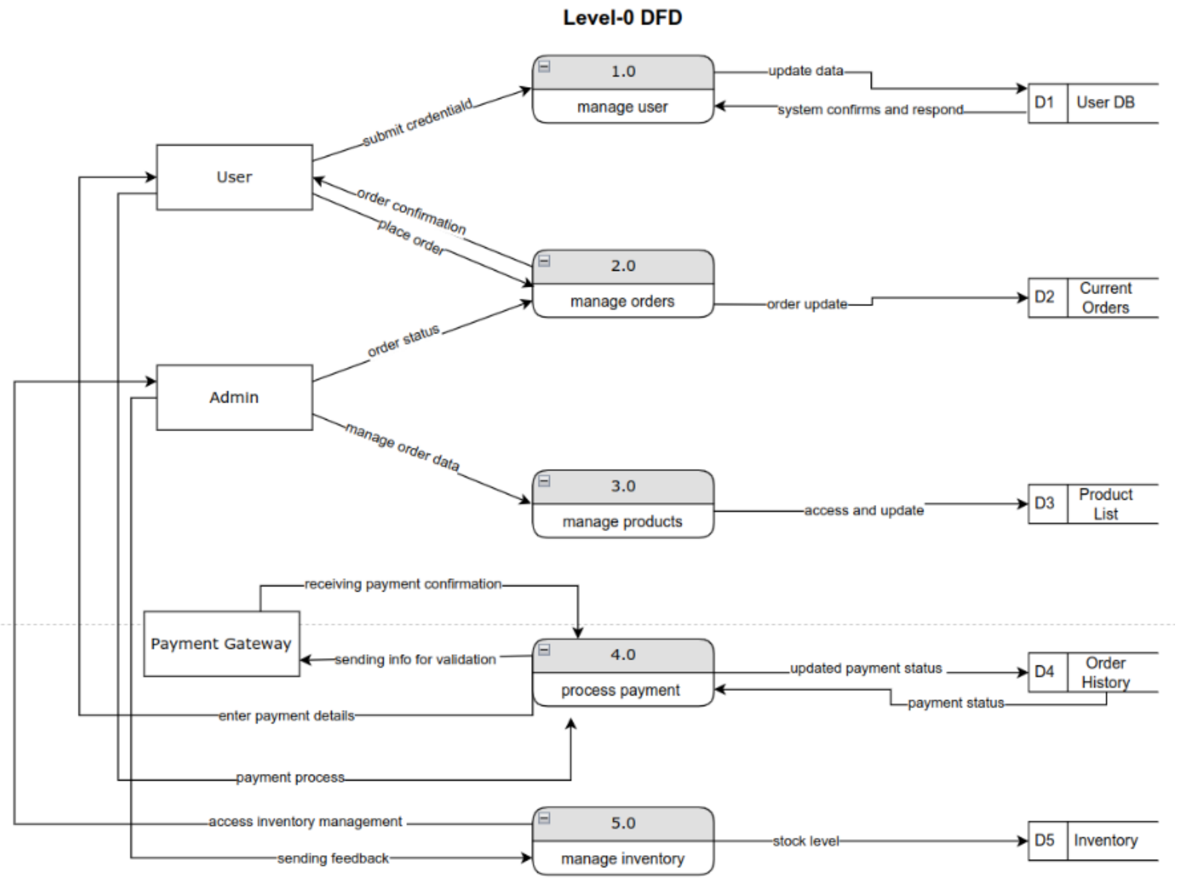

Figure 6: Level 0 DFD

2.4.3 Level 1 DFDs

2.4.3.1 Process 1.0

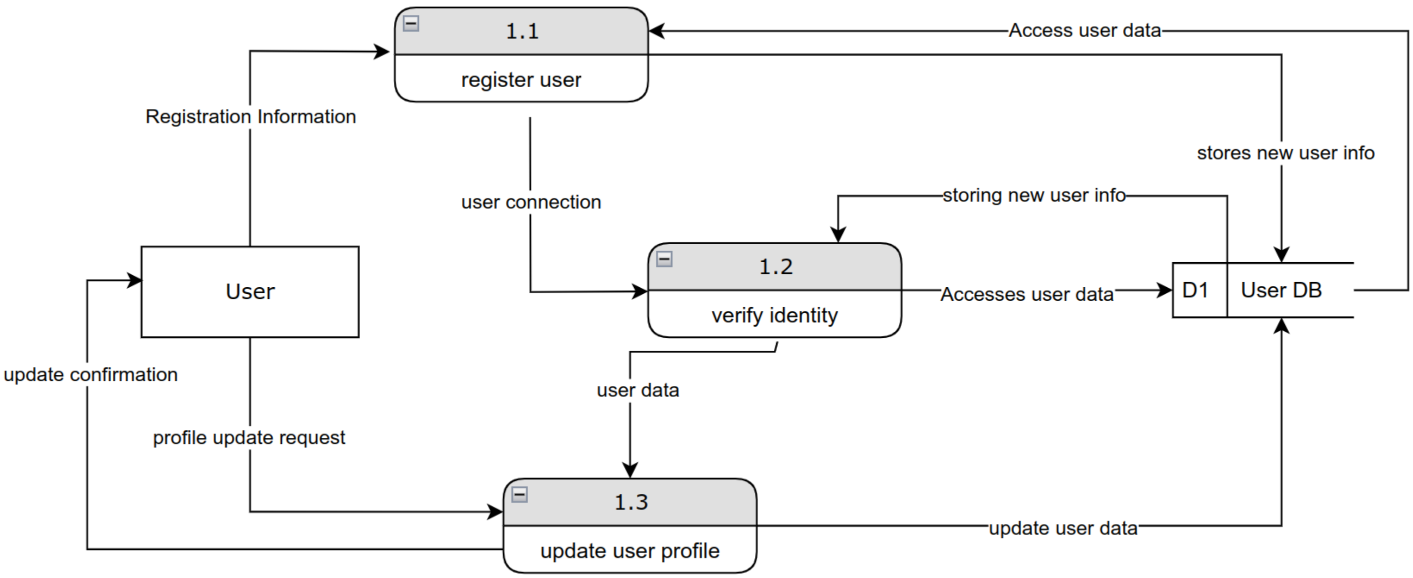

Figure 7: Level 1 DFD: Process 1.0

2.4.3.2 Process 2.0

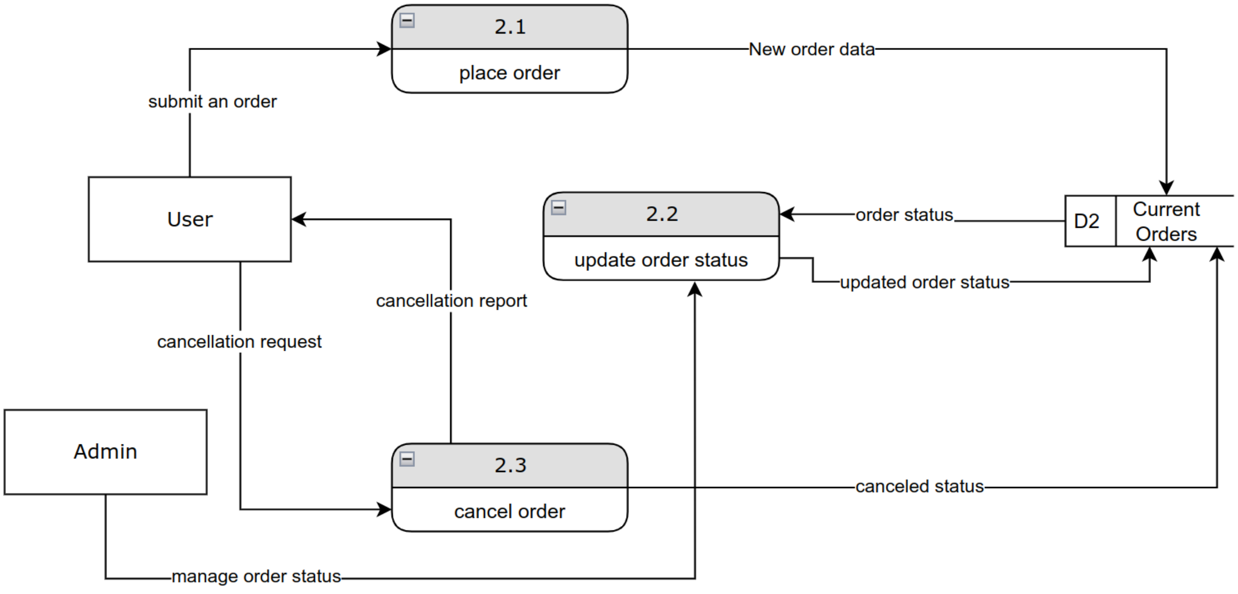

Figure 8: Level 1 DFD: Process 2.0

2.4.3.3 Process 3.0

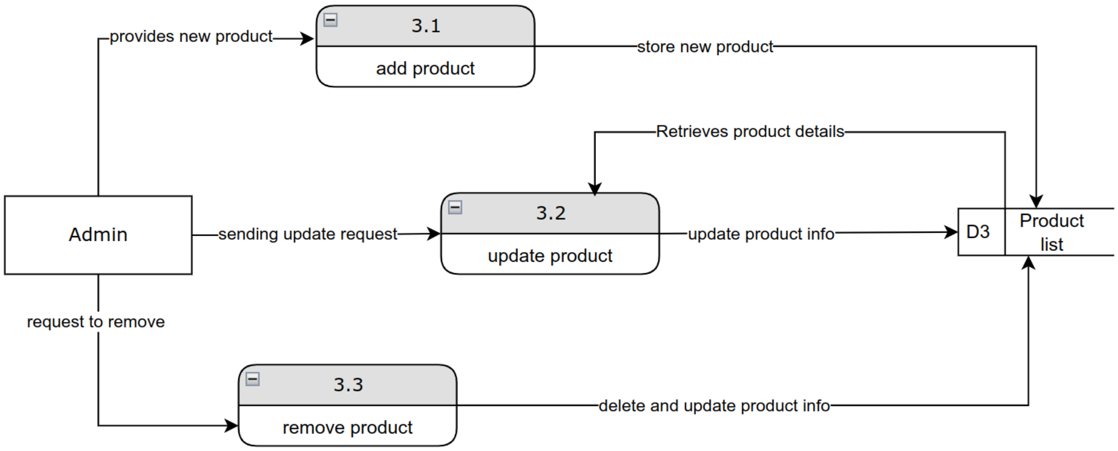

Figure 9: Level 1 DFD: Process 3.0

2.4.3.4 Process 4.0

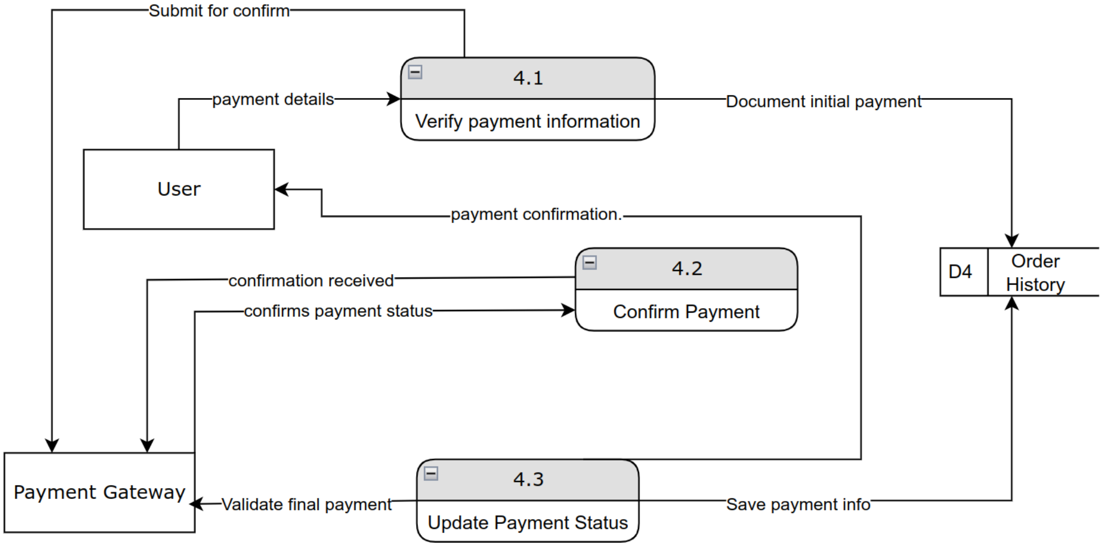

Figure 10: Level 1 DFD: Process 4.0

2.4.3.5 Process 5.0

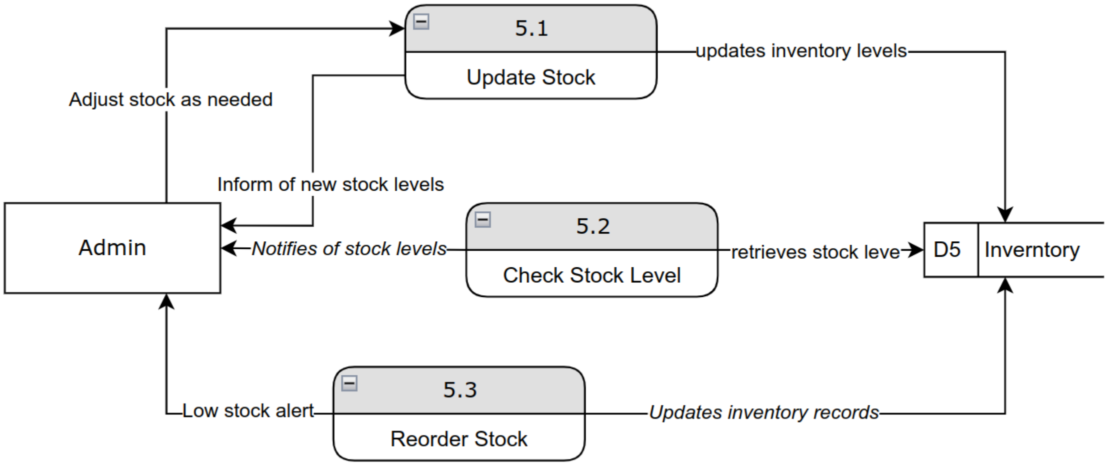

Figure 11: Level 1 DFD: Process 5.0

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.

Table 3: Process 1.0 Details.

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	• READ User credentials • IF credentials are valid THEN • UPDATE user information in User DB • DISPLAY confirmation message to User • ELSE • DISPLAY error message

Table 4: Process 2.0 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	• READ Order Details from User • ADD Order Data to Records • DISPLAY Order Confirmation to User • IF order update is requested THEN o UPDATE order information in Records o DISPLAY updated order status to User

Table 5: Process 3.0 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	• READ Product Details • IF new product THEN o ADD product to Product List • IF product update requested THEN o UPDATE Product Information in Product List • IF product removal requested THEN o DELETE product from Product List

Table 6: Process 4.0 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	• RECEIVE Payment Details from User • SEND Payment Details to Payment Gateway for validation • IF Payment Confirmation RECEIVED THEN o UPDATE Payment Status in Order History o DISPLAY payment confirmation to User • ELSE o DISPLAY error message to User

Table 7: Process 5.0 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	• READ Inventory Levels • IF stock level is low THEN o GENERATE reorder request • UPDATE Inventory Records

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.

Table 8: Process 1.0 Decision Table (Manage User)

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 9: Process 5.0 Decision Table (Manage Inventory)

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 2**X** amount			🟩	

2.6 Entity-Relationship Diagram (ERD)

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).

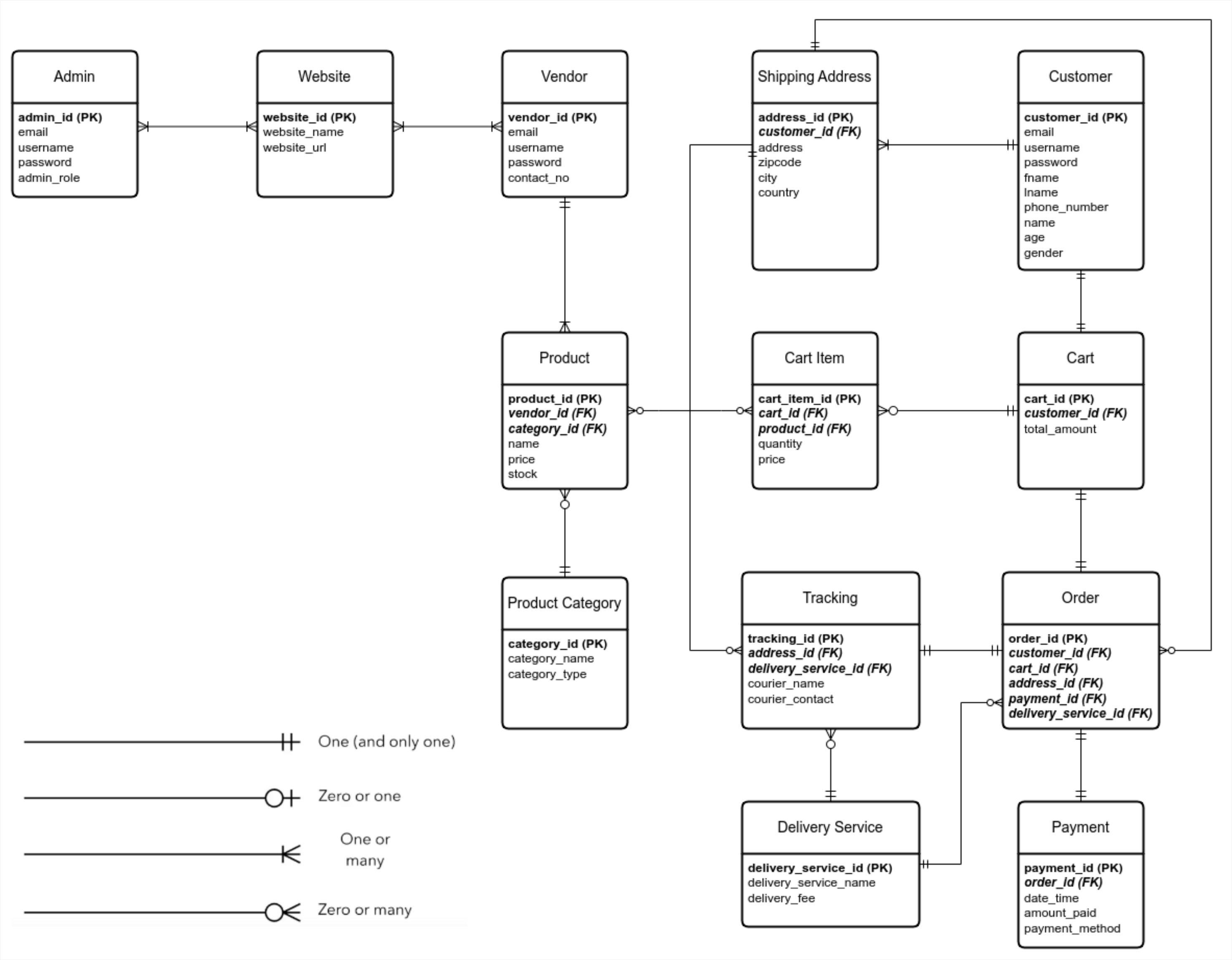

Figure 12: Entity-Relationship Diagram (ERD)