**GadgetHub Website Doucumentation**

**Prepared by members**

* Simon Muturi
* Elvis Gisiora
* Obed Keritich
* Kuku Ali ngocho

# INTRODUCTION

Electronic commerce, also known as e-commerce, involves the buying and selling of goods and services, or the transfer of funds or data, through electronic means, mainly over the internet. These transactions can occur between businesses, businesses and consumers, between consumers, or between consumers and businesses.

E-commerce has become a regular part of our daily routines, and advancements in technology allow people to shop online from the comfort of their homes without having to visit a physical store. This is a platform for people all around the world to be able to purchase goods and services online.

The project is divided into two main categories: Administrators and Customers/Users. The administrators, consisting of the store manager and staff members, have the ability to add, edit, update, or delete products, as well as change product names, prices, and availability. The customers have the ability to search for products, update their shopping carts, remove items, and check out. They can also update their personal information such as their name and address. Users are limited to only browsing the online shop and adding items to their cart.

**2.Functional and Non-functional Requirements of the system**

**Functional requirements** are the specific features and capabilities that an e-commerce site must have to meet the needs of its users and achieve its business goals. The standard functional requirements for this e-commerce site include the following:

* Product listings and catalog management
* Shopping cart and checkout
* Payment processing
* Order management and fulfillment
* Customer account management
* Search and filtering functionality
* Product reviews and ratings
* Product recommendations and up selling
* Marketing and promotional tools
* Customer service and support

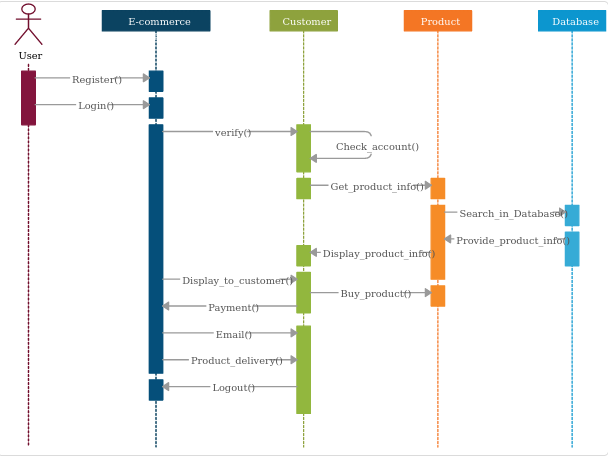
**Non-functional requirements** are the underlying characteristics of an e-commerce site that impact its overall performance, security, and usability. Some standard non-functional requirements for this e-commerce site include the following:

* Performance and scalability
* Security and privacy
* Usability and accessibility
* Mobile responsiveness
* Search engine optimization (SEO)
* Integration with other systems and platforms
* Data management and reporting
* Disaster recovery and backup
* Compliance with legal and industry standards

# 3. sequence diagram

# A sequence diagram is a type of interaction diagram that shows the interactions between objects or components in a system over time. In the case of an E-commerce website, a sequence diagram could be used to show how a customer interacts with the website to complete a purchase.

Here is a sequence diagram for our e commerce website:



In this diagram, we see the interactions between the customer, the website, and the payment gateway.

1. The customer begins by browsing the website and selecting items they wish to purchase.

2. The website sends a request to the inventory system to check if the items are available.

3. The inventory system responds with the availability of the items.

4. The customer proceeds to the checkout and enters their shipping and payment information.

5. The website sends a request to the payment gateway to process the payment.

6. The payment gateway responds with the status of the payment.

7. If the payment is successful, the website sends a request to the shipping system to process the order and send it to the customer.

8. The shipping system responds with the status of the shipment.

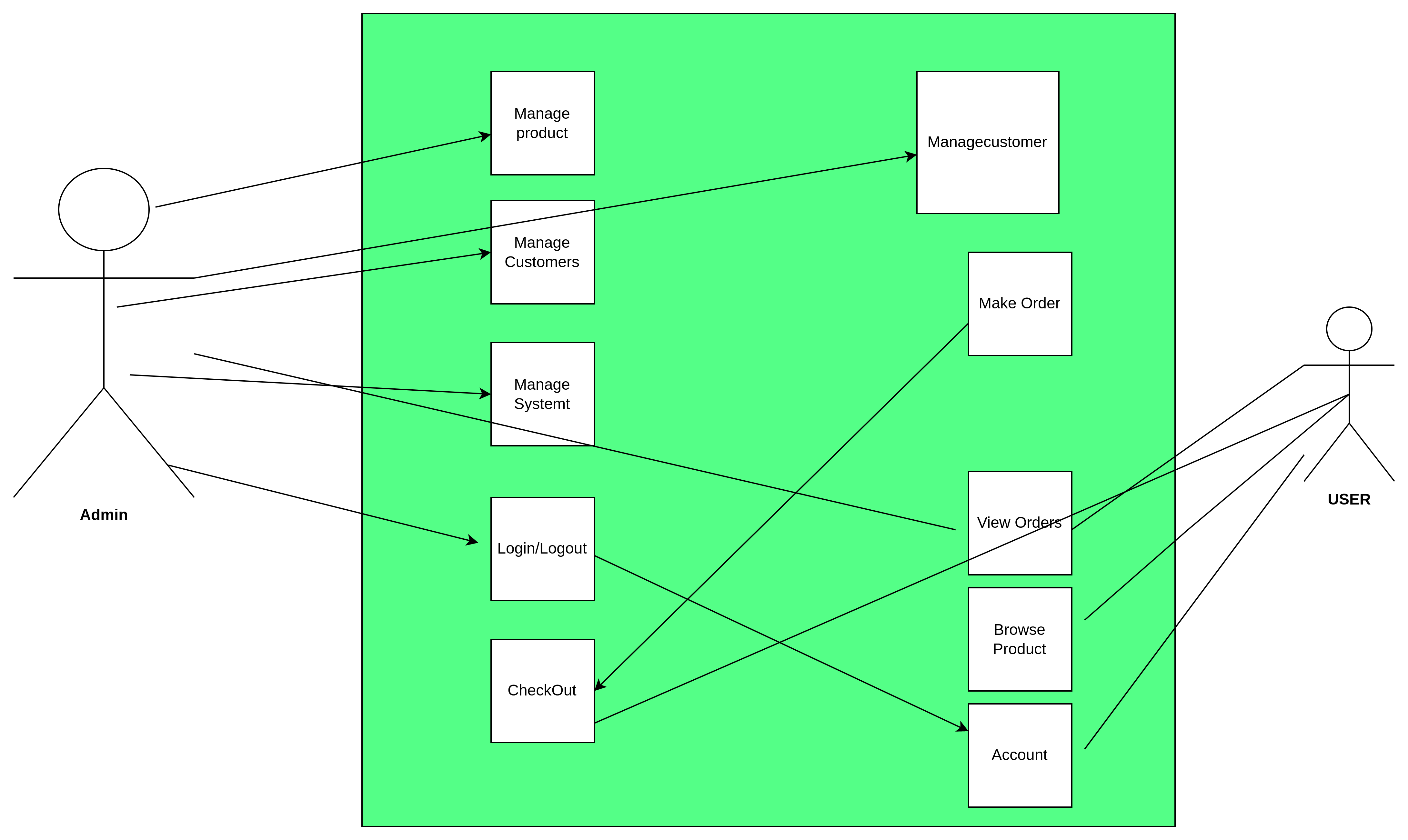
9. The website displays the confirmation page to the customer.

This sequence diagram provides a high-level overview of the interactions between the various components involved in a purchase on an e-commerce website. It helps to visualize the flow of information and actions between the customer, the website, and other systems, making it easier to identify potential issues or areas for improvement.

4. USE CASE diagram

A use case diagram is a type of behavior diagram that describes the interactions between actors (users) and a system. In the case of an E-commerce website, a use case diagram could be used to show the various actions that users can take on the website and how those actions relate to the overall functionality of the system.

Here is a use case diagram for our E-commerce website:

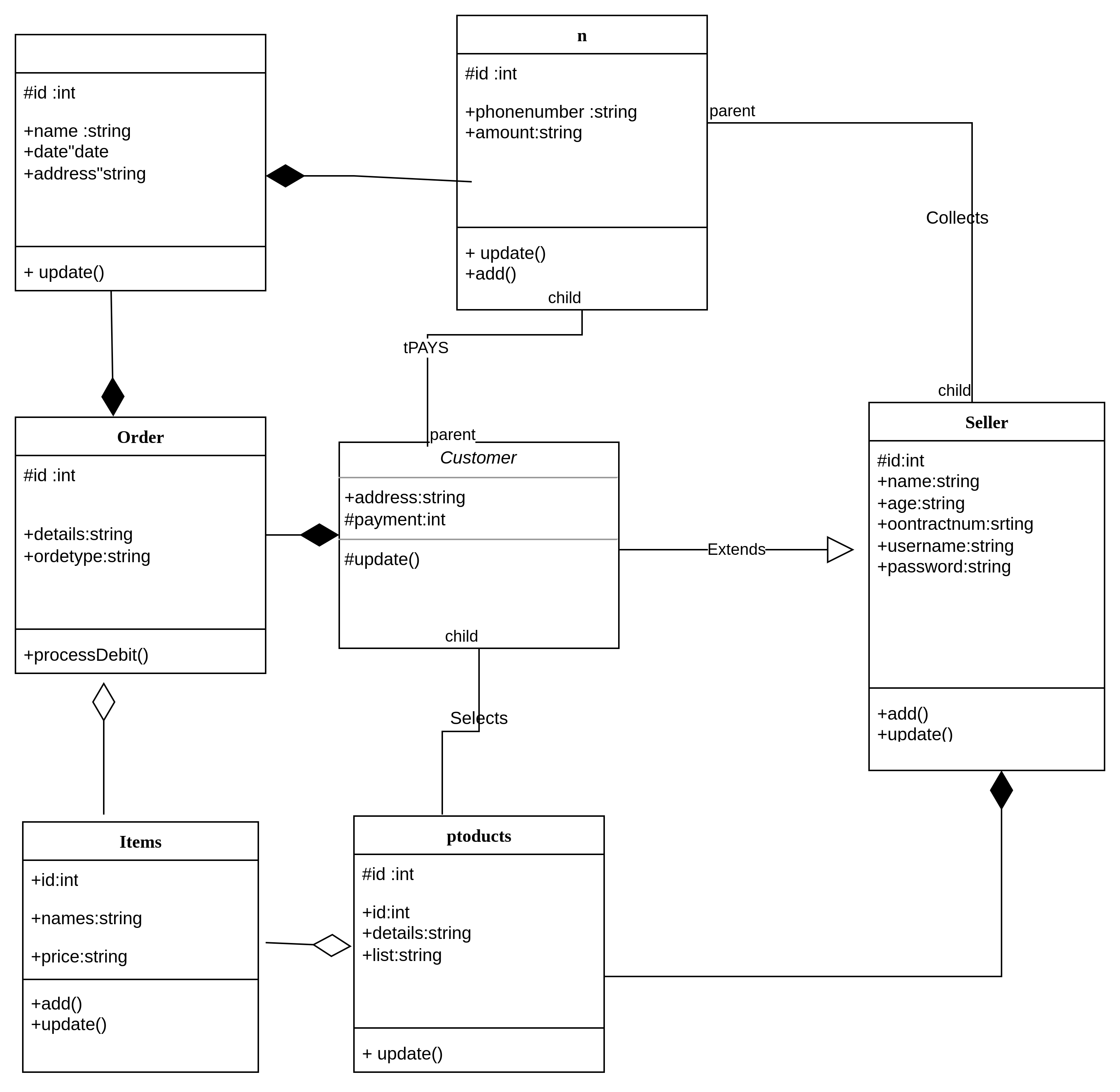


In this diagram, we see the various actors (users) who interact with the website and the use cases (actions) that they can perform.

1. The "Customer" actor can perform the following use cases:
   * Search for Products: Allows the customer to search for products on the website using keywords, categories, or filters.
   * View Product Details: Allows the customer to view the details of a product, including the price, description, and availability.
   * Add to Cart: Allows the customer to add a product to their shopping cart.
   * Remove from Cart: Allows the customer to remove a product from their shopping cart.
   * Checkout: Allows the customer to complete the purchase by entering their shipping and payment information.
   * Track Order: Allows the customer to track the status of their order and delivery.
2. The "Admin" actor can perform the following use cases:
   * Manage Products: Allows the admin to add, update, or remove products from the website.
   * Manage Orders: Allows the admin to view and manage orders, including updating the status and tracking information.
   * Manage Customers: Allows the admin to view and manage customer accounts, including adding or removing users.
3. The "System" actor represents the various systems and third-party services that interact with the website, such as the payment gateway, inventory system, and shipping system.

This use case diagram provides a high-level overview of the different actors and their interactions with the E-commerce website. It helps to identify the different use cases and how they relate to the overall functionality of the system. By visualizing the interactions between actors and the system, it is easier to design and develop the E-commerce website and ensure that it meets the needs of its users.

**5. A class diagram**

A class diagram is a type of static structure diagram that shows the classes, attributes, methods, and relationships between objects in a system. In the case of an E-commerce website, a class diagram could be used to represent the various entities and their relationships within the system.Here is a class diagram for our E-commerce website:

In this diagram, we see the different classes that represent the entities within the E-commerce website and their relationships.

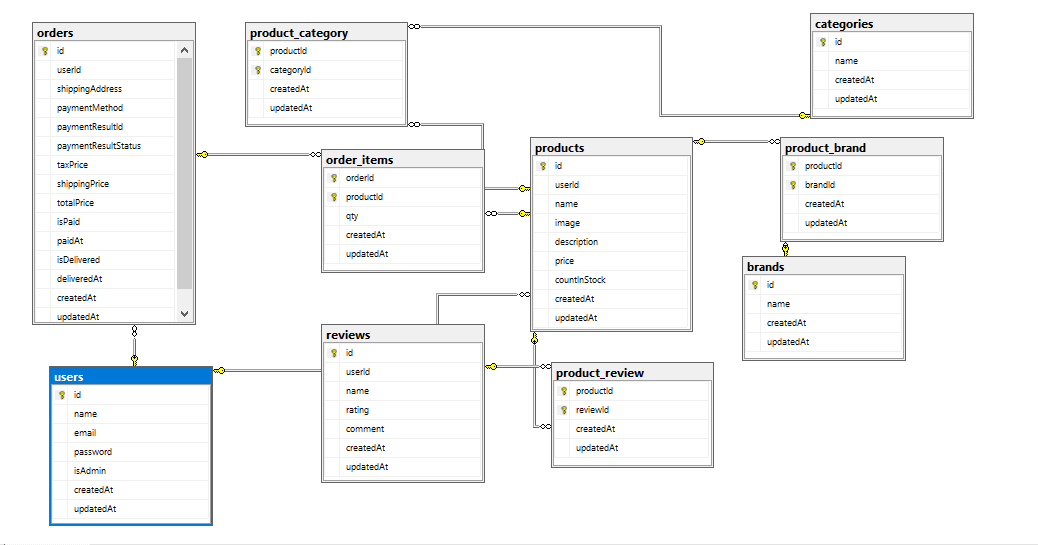
1. The "Customer" class represents the customer entity and has attributes such as "customerID", "name", "email", and "password". It also has methods such as "placeOrder()" and "trackOrder()".
2. The "Product" class represents the product entity and has attributes such as "productID", "name", "description", "price", and "quantity". It also has methods such as "addProduct()" and "removeProduct()".
3. The "Order" class represents the order entity and has attributes such as "orderID", "date", "status", and "totalPrice". It also has methods such as "placeOrder()" and "cancelOrder()".
4. The "ShoppingCart" class represents the shopping cart entity and has attributes such as "cartID" and "totalPrice". It also has methods such as "addProduct()" and "removeProduct()".
5. The "Payment" class represents the payment entity and has attributes such as "paymentID", "amount", and "paymentMethod". It also has methods such as "processPayment()" and "refundPayment()".
6. The "Inventory" class represents the inventory entity and has attributes such as "productID" and "quantity". It also has methods such as "checkAvailability()" and "updateQuantity()".
7. The "CustomerAccount" class represents the customer account entity and has attributes such as "customerID", "name", "email", and "password". It also has methods such as "login()" and "register()".

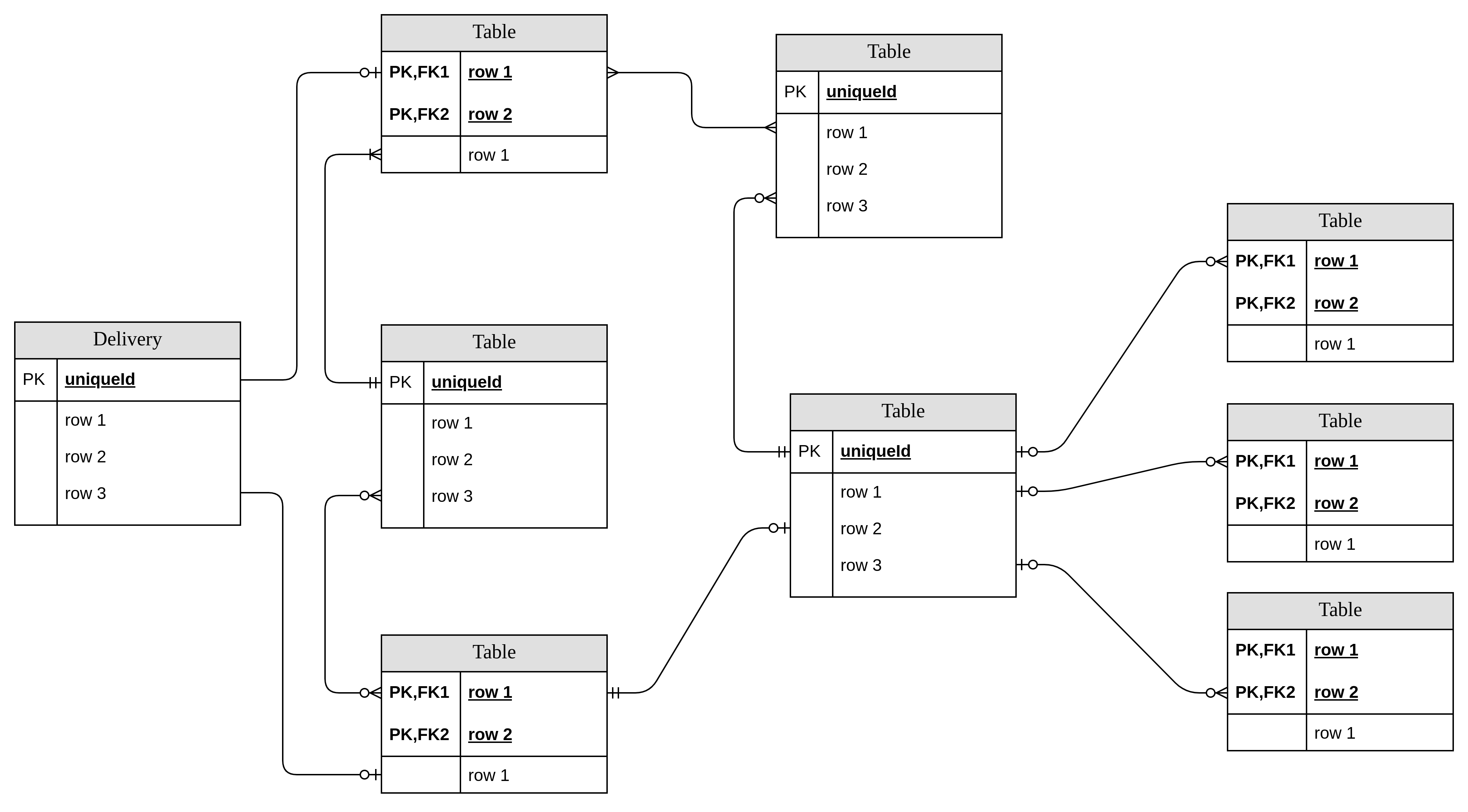
This class diagram provides a high-level overview of the different classes and their relationships within the E-commerce website. It helps to identify the different entities and their attributes and methods, as well as how they relate to each other. By visualizing the entities and their relationships, it is easier to design and develop the E-commerce website and ensure that it meets the needs of its users

**6. Entity Relationship Diagram**

An entity-relationship diagram (ERD) is a graphical representation of entities and their relationships to one another in a system or organization. In the context of an e-commerce website, an ERD can be used to model the various entities that are involved in the website, including customers, products, orders, and payments.

Here is an example of an ERD for an e-commerce website:





The diagram includes four main entities: Customers, Products, Orders, and Payments. Each of these entities has its own set of attributes, which are listed below:

1. Customers: The customers entity represents the people who use the e-commerce website to purchase products. The attributes of this entity include:

* Customer ID (primary key)
* Name
* Email
* Phone Number
* Shipping Address

1. Products: The products entity represents the various items that are available for purchase on the e-commerce website. The attributes of this entity include:

* Product ID (primary key)
* Name
* Description
* Price
* Category

1. Orders: The orders entity represents the purchases made by customers on the e-commerce website. The attributes of this entity include:

* Order ID (primary key)
* Customer ID (foreign key)
* Order Date
* Total Price

1. Payments: The payments entity represents the various payment methods that customers can use to complete their purchases on the e-commerce website. The attributes of this entity include:

* Payment ID (primary key)
* Order ID (foreign key)
* Payment Method
* Amount

The relationships between these entities are also represented in the ERD. For example, there is a one-to-many relationship between the Customers and Orders entities, meaning that each customer can have multiple orders. Similarly, there is a many-to-many relationship between the Products and Orders entities, meaning that each order can contain multiple products and each product can be included in multiple orders.

Overall, the ERD provides a visual representation of the various entities and relationships involved in an e-commerce website, making it easier to understand and manage the system as a whole.

**7. CONCLUSION**

The electronic website is developed using MySQL, HTML,Typescript,JavaScript and CSS technology. Any consumer can browse products, add, replace or delete a product from the cart.

The consumer can log in, with his information such as his email and password. If the login does not go through, the user can re-register or ask to change the password. After login, the user can see the product in the cart and proceed onward. The product can be paid with

M-PESA Express. The administrator can verify the order, However

the consumer can still look at the orders in his or her account. The ordered price is saved in the database.

**FUTURE IMPROVEMENT**

Invoices need to be implemented in the websites, emails and notifications need's to be sent to customers for new arrivals or discount. The shop has to have a search engine where users and customers can search for the various product from the shop. Debit and credit cards need's to be implemented in the shop as well. There have to be language varieties so that none-English users and customers can shop easily without any difficulty.