A picture containing text

Description automatically generated

E-Commerce

Team ID:32

Icon

Description automatically generated with medium confidence

|  |  |  |
| --- | --- | --- |
|  | **ID**  **[Ordered by ID]** | **Full Name**  **[In Arabic]** |
| **1** | **20210883** | **محمود نصر عبدالعزيز** |
| **2** | **20210192** | **أنس صالح موسي الصادق** |
| **3** | **20210191** | **أندرو عاطف شكرالله** |
| **4** | **20210753** | **محمد ايهاب محمد فريد** |
| **5** | **20210742** | **محمد اشرف ابوالمعاطى** |
| **6** | **20211041** | **يارا محمد عطية الطوخى** |
| **7** | **20210959** | **منةالله عصام عبدالعزيز** |

**1. Requirements Specification**

**1.1 Functional Requirements**

**User:**

* Sign Up

User should provide his information including (Full Name, username, Password,..).

* login

After signup user can input his username and password and log in.

* Setting

User has an option for an alternate email address, phone number, preferred language, or change password.

* Change password

User has an ability to change the old password.

* Change email

User can change the old email.

* Change mobile number

User can change the telephone number

* Alert and Notification

User can specific notifications in the application

* logout

User can logout from his session on website.

**Admin:**

* Add Category

The administrator shall be able to add a new Category.

* Update Category

The administrator shall be able to modify/update the Category.

* Delete Category

The administrator shall be able to delete items from the main page of the

shopping-cart application.

The administrator shall be able to delete items from the main page of the

shopping-cart application.

The administrator shall be able to delete Category.

* Block User

The administrator can block user from getting into his website.

* Add Product

The seller shall be able to add new products or items to the list of shopping items.

* Update Product

The seller shall be able to modify/update products or items price and description.

* Delete Product

The administrator shall be able to delete items from the main page of the

shopping-cart application.

The administrator shall be able to delete items from the main page of the

shopping-cart application.

The seller shall be able to delete products or items from website

**Buyer:**

* Search

buyer can type name of product to search.

buyer should be able to view their favorite categories or product or sellers.

* View Category

The buyer shall be able to view the categories on home.

* View Product

The buyer can view more information about an item before adding it to the cart.

* Change color

buyer can change the color of product.

* Change size

buyer can change the size of product.

* View History

buyer can see earlier product history by pressing history button.

* Add to cart

buyer can add item to the cart.

* Delete from cart

The buyer shall be able to delete items from the cart.

* View cart

The buyer shall be able to view the items added to the cart.

* Order

buyer can press order now button in the cart to buy the product.

* Cancel Order

buyer should be able to cancel an order before it is shipped.

* Choose Payment Method

buyer should be able to view their preferred payment option.

* Choose Delivery Option

buyer should be able to view their preferred delivery option.

* Refund

buyer can refund the product cost by button refund.

* Purchase History

buyer should be able to view their purchase history.

* Report for a problem

buyer should be able to report any issues or problem with transaction.

The administrator shall be able to delete items from the main page of the

shopping-cart application.

The administrator shall be able to delete items from the main page of the

shopping-cart applicatio

**2) Non-Functional Requirements**

1) Performance

1.1) The system should be compatible with all modern browsers.

1.2) The system must respond to the operation in less than 3 Seconds

1.3) Each page must load within 2 seconds.

1.4) The system should be reliable.

2) Security

2.1) The system must handle safe login and logout through session.

2.2) System should protect data from unauthorized access.

2.3) The database should be secured from SQL injection to prevent leak or

loss of information.

3) Reliability

3.1) The system should be reliable and provide catching of exceptions so that unintended results do not occur such as system crashes, or data validation failures.

3.2) The system must be available 24/7.

4) Usability

4.1) The system should have user-friendly interface.

5) ease of use

5.1) The system shall provide help link on each page providing help for functions that can be performed from that page.

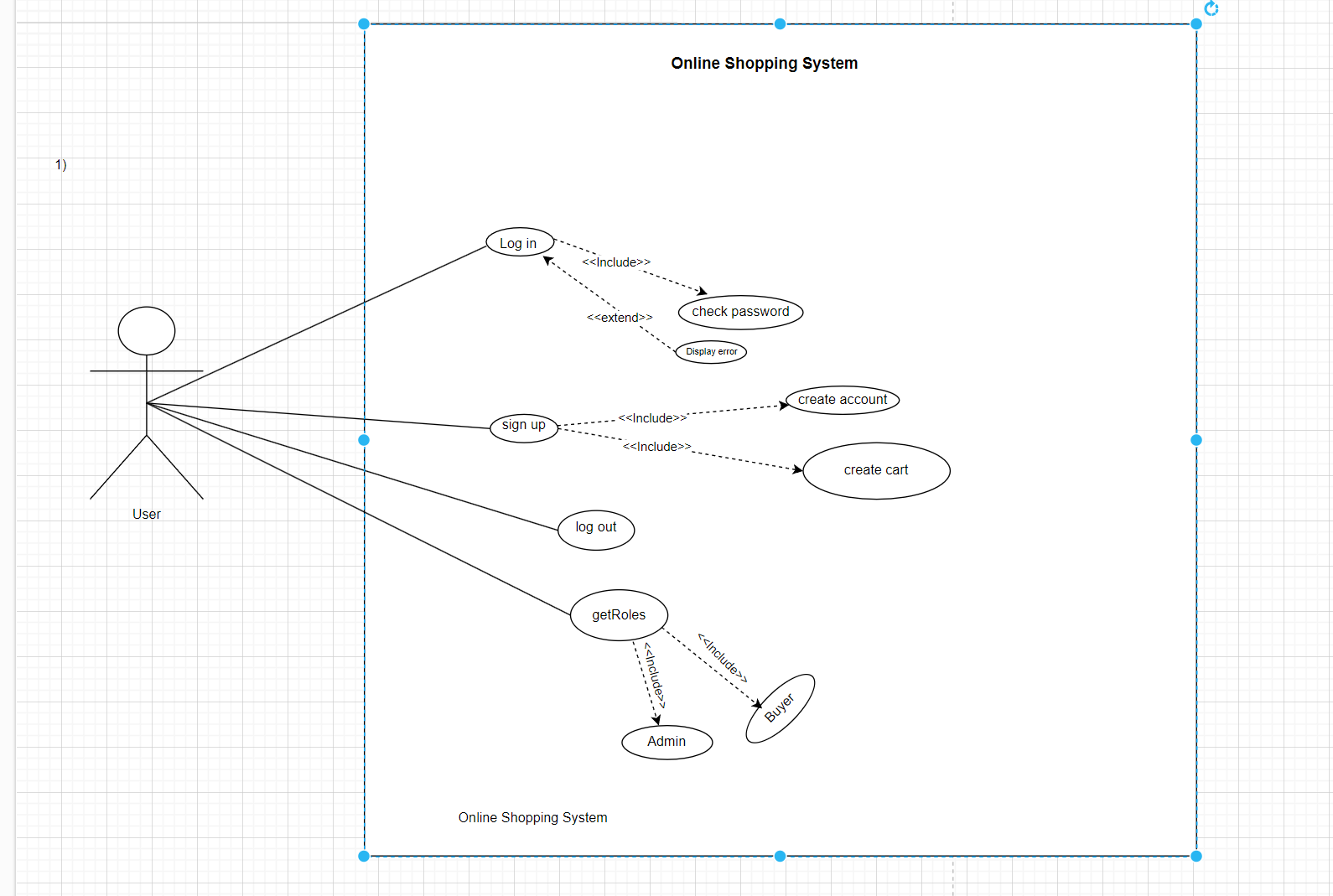
5.2) The system shall provide menu bar for easy traversal from page to page.

5.3) Depending on the type of user system shall direct the user to appropriate user interface.

5.4) System shall use consistent layout between user interfaces

**2.Use-Case Diagrams**

**User - Use Case**

****

**User - Use Case**

|  |  |
| --- | --- |
| ID Number – Name: | Us\_001–Login |
| Short-Description / Goal: | To get into the Web app |
| Preconditions: | User must have an account so he/she can login |
| Postconditions: | Logged in successfully and he/she can get benefits of the application |
| Error Situations – System  State in the Event of an Error: | The username or password is invalid ( not exists in the database ) |
| Actor(s) / Initiator(s): | Any User (Admin, Buyer ) |
| Trigger: | user enters his username and password in the login form and clicks the "Login". |
| Standard Process  (Main Success Scenario): | 1.the user types in his credentials  2.the system is going to check if it exists in the database 3.eventually user logged in successfully |
| Alternative Processes  (Alternative/Other Scenarios): | Credentials doesn’t exist so user must keep trying |

|  |  |
| --- | --- |
| ID Number – Name: | Us\_002– Sign up |
| Short-Description / Goal: | User creates an account |
| Preconditions: | User must be connected to the Internet |
| Postconditions: | The user created an account and can login to the system |
| Error Situations – System  State in the Event of an Error: | The username or email is already exists in the database |
| Actor(s) / Initiator(s): | Any User ( Buyer ) |
| Trigger: | User wants to have an account on the website application |
| Standard Process  (Main Success Scenario): | 1.user opens the Web application  2.user adds his information such as username and password  3.the account is created successfully |
| Alternative Processes  (Alternative/Other Scenarios): | —— |

|  |  |
| --- | --- |
| ID Number – Name: | Us\_003–Check password |
| Short-Description / Goal: | To check if the password is correct |
| Preconditions: | User must have an account and already entered his credentials |
| Postconditions: | If the password is correct then user is logged in successfully or else cannot login |
| Error Situations – System  State in the Event of an Error: | Wrong password so Cannot login to the system so try again |
| Actor(s) / Initiator(s): | Any User (Admin, Buyer and Seller) |
| Trigger: | User added his credentials |
| Standard Process  (Main Success Scenario): | 1.user must have an account  2.add his credentials |
| Alternative Processes  (Alternative/Other Scenarios): | —— |

|  |  |
| --- | --- |
| ID Number – Name: | Us\_004–Log out |
| Short-Description / Goal: | To logout from the Web application |
| Preconditions: | User in logged in to the application |
| Postconditions: | User logged out successfully |
| Error Situations – System  State in the Event of an Error: | —— |
| Actor(s) / Initiator(s): | Any User (Admin, Buyer and Seller) |
| Trigger: | The User clicks on the " logout " button. |
| Standard Process  (Main Success Scenario): | 1.user must have an account  2.user is done with his needs and wants to logout  3.user logged out successfully |
| Alternative Processes  (Alternative/Other Scenarios): | —— |

**Admin - Use Case**

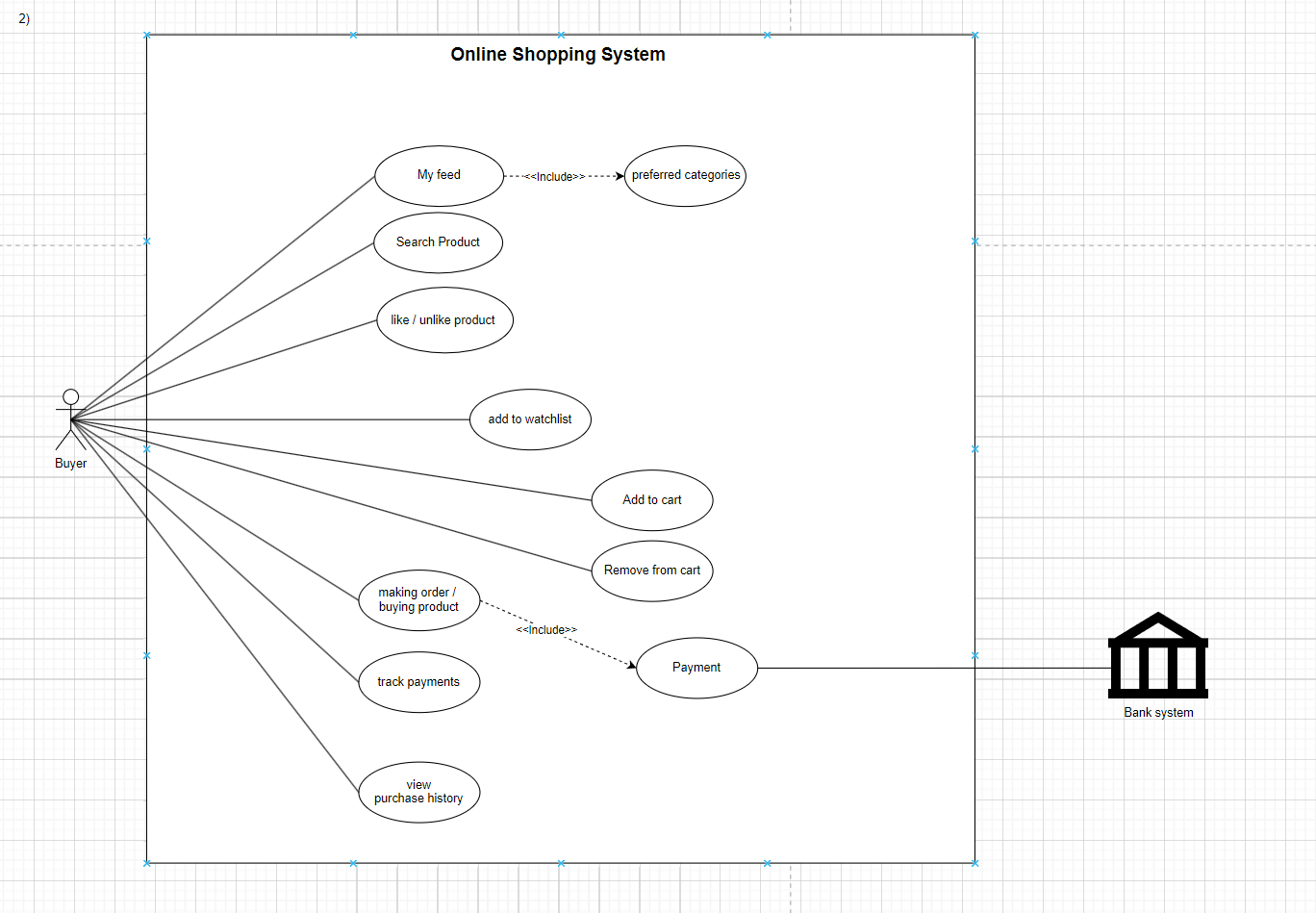
**A diagram of a product

Description automatically generated**

**Admin - Use Case**

|  |  |
| --- | --- |
| ID Number – Name: | Ad\_001–Add category |
| Short-Description / Goal: | Categories can be added, deleted and updated |
| Preconditions: | Admin logged in to the system and wants to manage categories |
| Postconditions: | Admin added, deleted or updated category. |
| Error Situations – System  State in the Event of an Error: | Changed categories are not modified |
| Actor(s) / Initiator(s): | Admin |
| Trigger: | The Admin clicks on the " added, deleted or updated " button. |
| Standard Process  (Main Success Scenario): | 1.admin logged in to the system  2.admin needs to manage categories  3. Categories are added, deleted and updated successfully |
| Alternative Processes  (Alternative/Other Scenarios): | —— |

**Buyer - Use Case**

****

|  |  |
| --- | --- |
| ID Number – Name: | CT\_001– AddToCart |
| Short-Description / Goal: | The goal of this use case is to allow the buyer to add an item to their shopping cart for purchase. |
| Preconditions: | The buyer has selected at least one item to purchase. |
| Postconditions: | The selected item(s) are added to the shopping cart. |
| Error Situations – System  State in the Event of an Error: | If the item is out of stock, the system displays an error message and does not allow the user to add it to the cart. |
| Actor(s) / Initiator(s): | buyer |
| Trigger: | The buyer clicks on the "Add to Cart" button for the selected item. |
| Standard Process  (Main Success Scenario): | 1.The system displays the details of the selected item, including its name, description, and price.  2.The user clicks on the "Add to Cart" button.  3. The system checks if the item is in stock.  4. If the item is in stock, the system adds the item to the cart and displays a confirmation message. |
| Alternative Processes  (Alternative/Other Scenarios): | If the user wants to change the quantity of the item before adding it to the cart. |

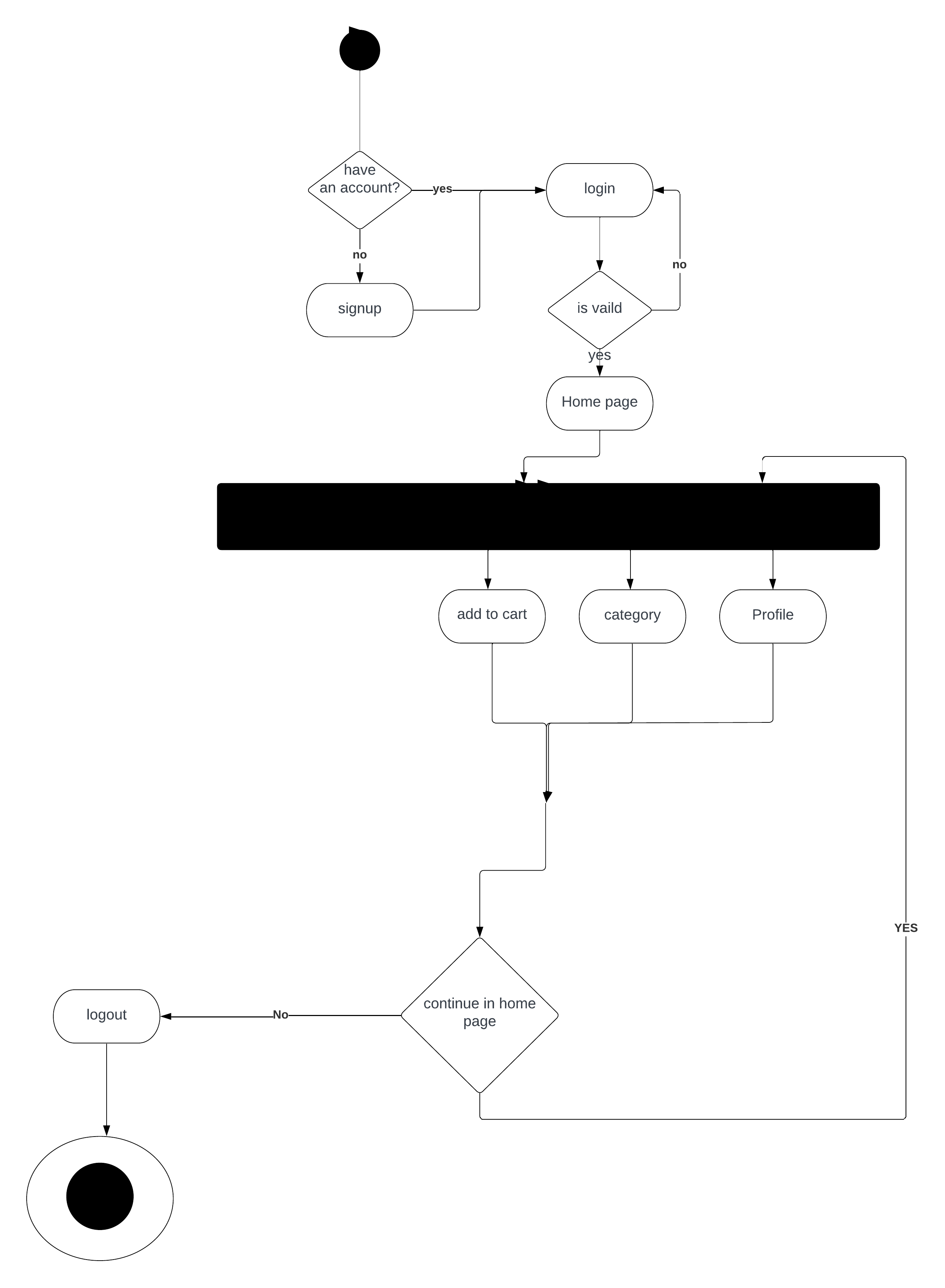
|  |  |
| --- | --- |
| ID Number – Name: | CT\_002– RemoveFromCart |
| Short-Description / Goal: | The goal of this use case is to allow the buyer to remove an item from their shopping cart. |
| Preconditions: | The buyer has added at least one item to the cart. |
| Postconditions: | The selected item(s) are removed from the shopping cart. |
| Error Situations – System  State in the Event of an Error: | If the item is not found in the shopping cart, the system displays an error message and does not remove any items. |
| Actor(s) / Initiator(s): | buyer |
| Trigger: | The buyer clicks on the "Remove From Cart" button for the selected item. |
| Standard Process  (Main Success Scenario): | 1.The system displays the contents of the user's shopping cart.  2.The user selects the item they wish to remove from the cart by clicking on the "Remove" button next to the item.  3. The system prompts the user to confirm  4. The user confirms by clicking on the "Yes" button |
| Alternative Processes  (Alternative/Other Scenarios): | 1.If the user selects the wrong item, they can click on the "Cancel" button to go back to the shopping cart without removing any items.  2.If the user wants to remove all items from the cart, they can click on the "Empty Cart" button to remove all items from the cart. |

|  |  |
| --- | --- |
| ID Number – Name: | CT\_003– Making Order |
| Short-Description / Goal: | The goal of this use case is to allow the user to place an order for the items in their shopping cart. |
| Preconditions: | 1. The buyer has added at least one item to the cart. 2. The user has provided their shipping and payment information. |
| Postconditions: | 1. The system confirms the order and sends an order confirmation email to the user. 2. The system updates the inventory of the purchased items. 3. The user can view their order details in their account. |
| Error Situations – System  State in the Event of an Error: | If the user's payment information is invalid, the system displays an error message and asks the user to update their payment information. |
| Actor(s) / Initiator(s): | buyer |
| Trigger: | The user clicks on the "Place Order" button after reviewing their shopping cart and confirming their shipping and payment information. |
| Standard Process  (Main Success Scenario): | 1. The system displays a summary of the user's order. 2. The user confirms the order by clicking on the "Place Order" button. 3. The system checks the user's payment information and confirms that the payment is successful. 4. The system updates the inventory of the purchased items. 5. The system confirms the order and sends an order confirmation email to the user. 6. The user can view their order details in their account. |
| Alternative Processes  (Alternative/Other Scenarios): | 1. If the user wants to update their shipping or payment information, they can click on the "Edit" button to make changes before placing the order. 2. If the user wants to cancel the order, they can click on the "Cancel Order" button before the order is confirmed. |

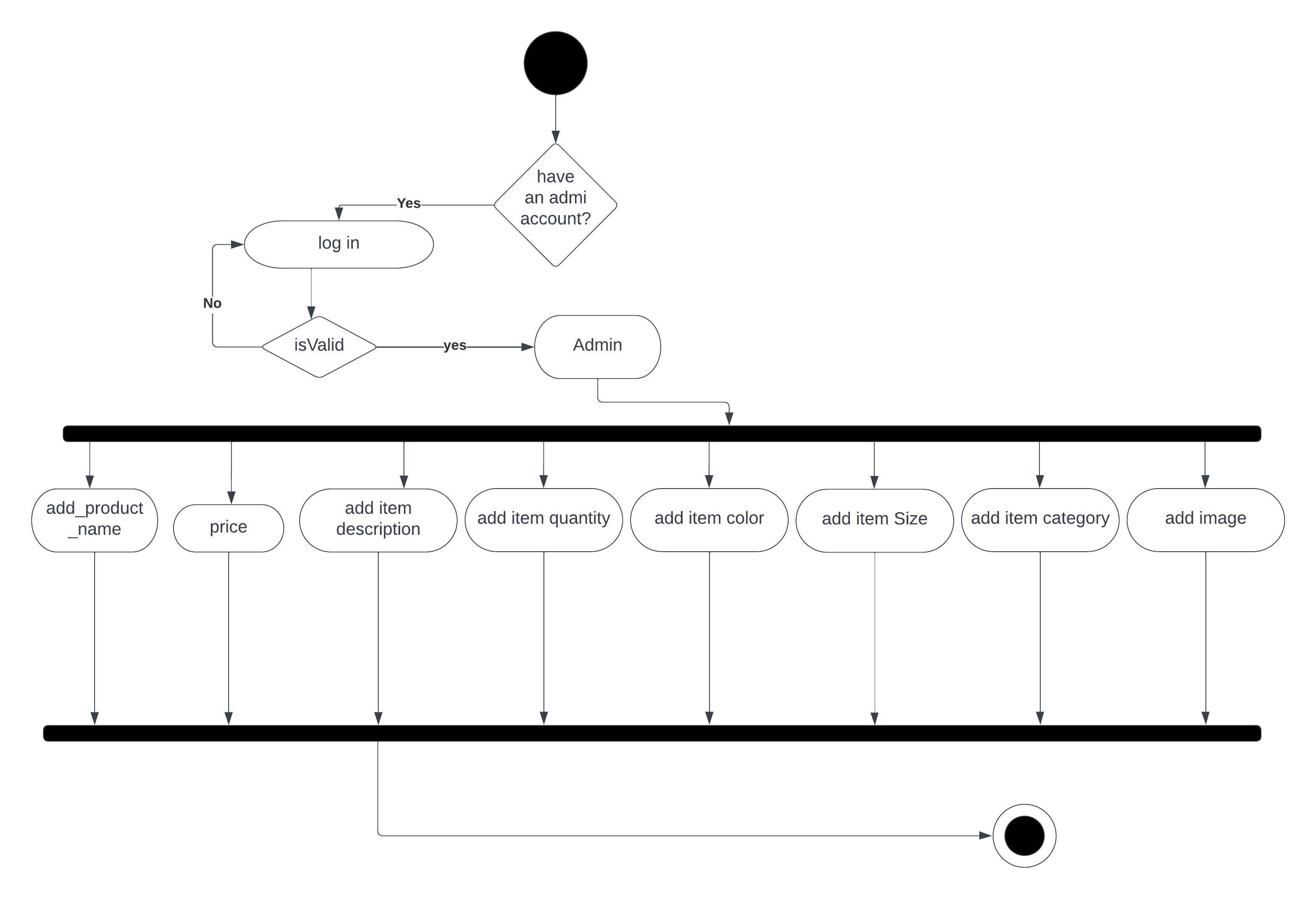
|  |  |
| --- | --- |
| ID Number – Name: | PurchaseHistory\_001 - Purchase History |
| Short-Description / Goal: | The goal of this use case is to allow the user to view their purchase history on the online store. |
| Preconditions: | 1. The user is logged in to their account. 2. The user has made at least one purchase on the online store. |
| Postconditions: | 1. The user can view their purchase history in their account |
| Error Situations – System  State in the Event of an Error: | If there is a technical error |
| Actor(s) / Initiator(s): | buyer |
| Trigger: | The user clicks on the "Purchase History" button in their account settings. |
| Standard Process  (Main Success Scenario): | 1. The system displays a list of the user's past orders, including the order details. 2. The user can click on the "Reorder" button to add the items from a previous order to their shopping cart. |
| Alternative Processes  (Alternative/Other Scenarios): | 1. If the user has not made any purchases on the online store, the system displays a message indicating that there is no purchase history available. |

**4. Activity Diagram**

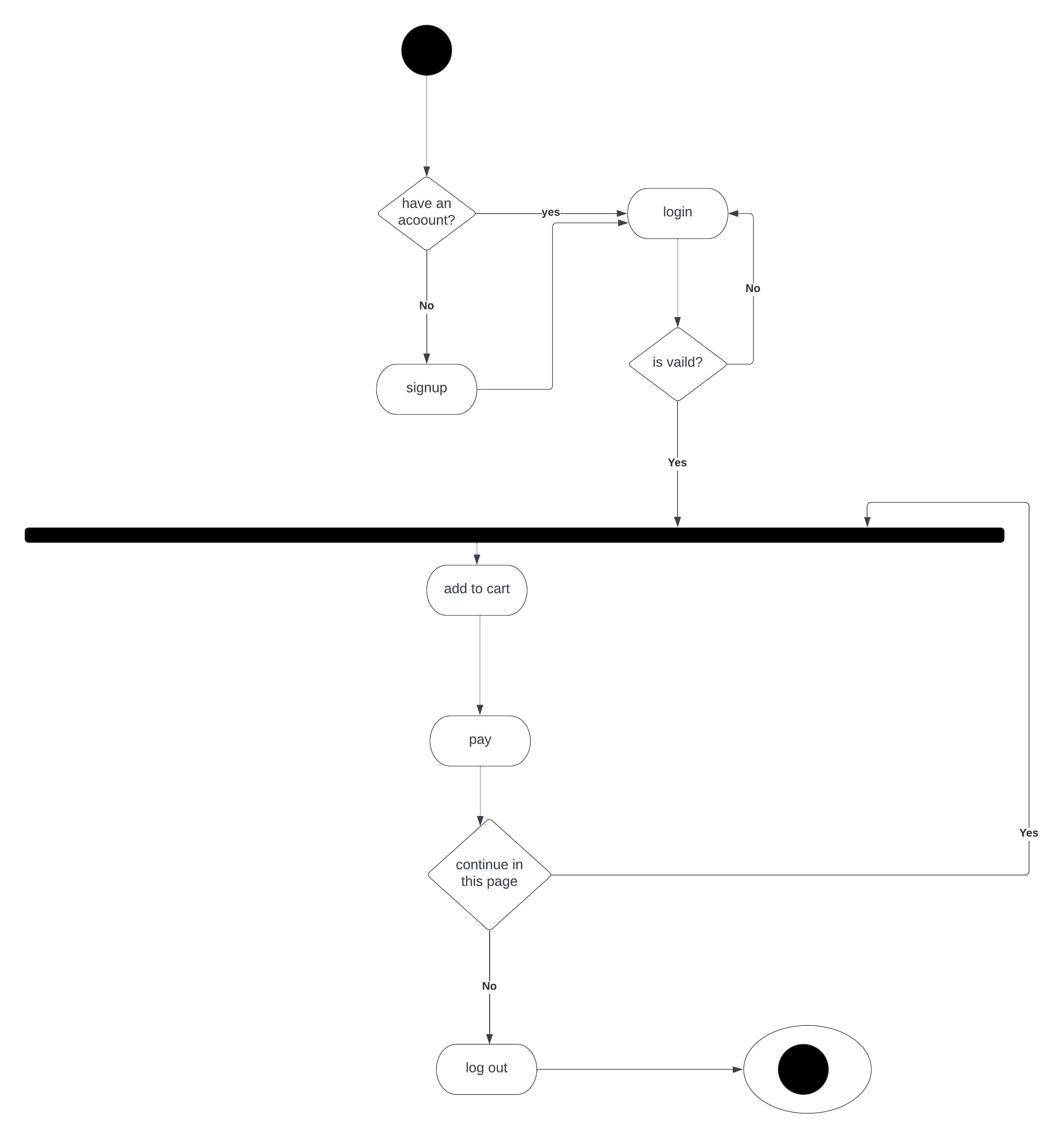
**User**

****

**Admin**

****

**Buyer**

****

**Diagram

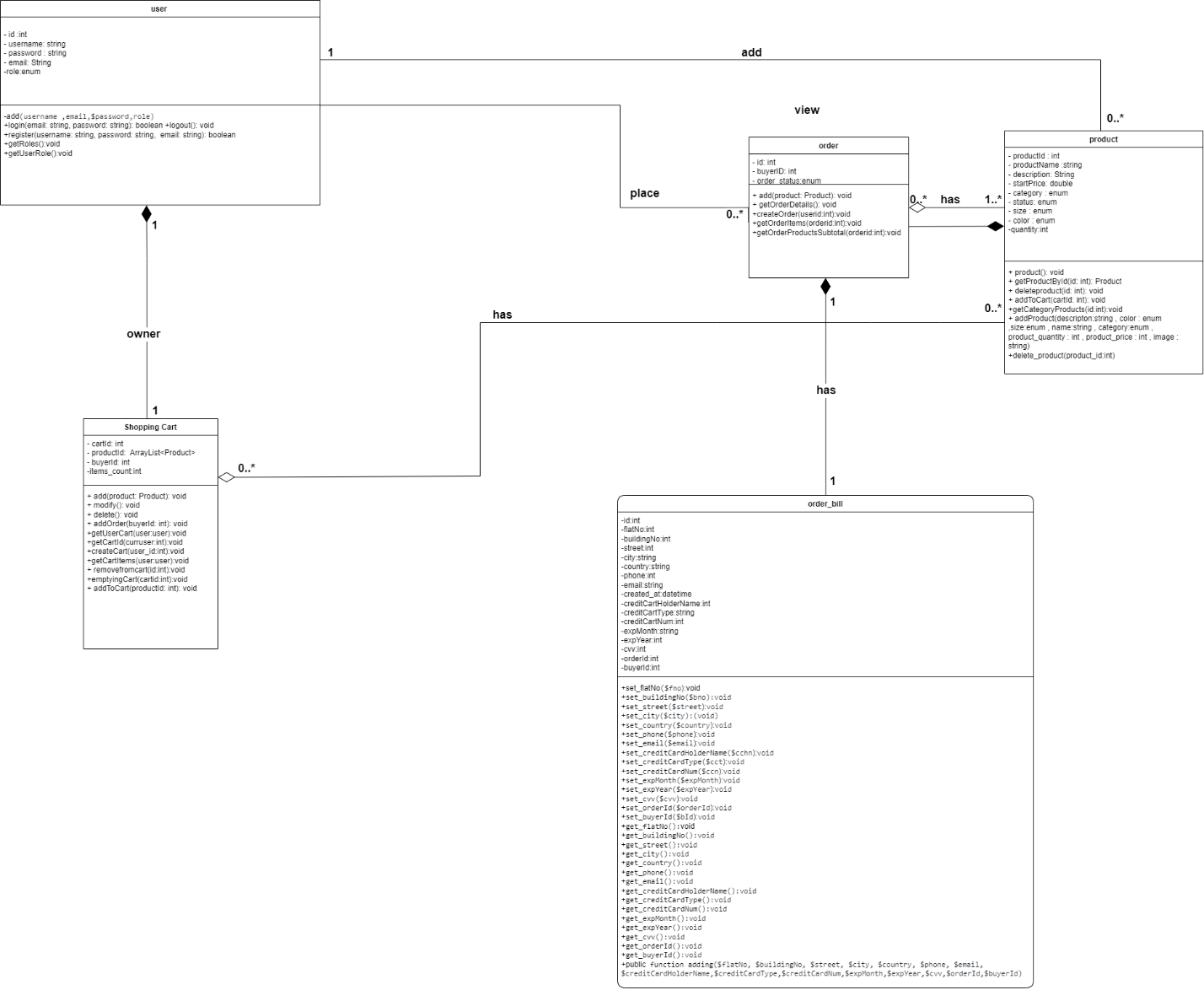
Description automatically generated**

1. **Class Diagram v1**

**Diagram

Description automatically generated**

**Class Diagram v2**

****

**OCL:**

context User :: register(username: string, password: string, email: string)

pre: (self.password)->size() > 8

context User :: addProduct(descripton:string , color : enum ,size:enum , name:string , category:enum , product\_quantity : int , product\_price : int , image : string)

pre: self.role==ADMIN

context User :: new\_order(id , buyer\_id,status)

pre: owner.items\_count->size() > 0;

context Admin :: delete\_product(product\_id:int)

post: product.product\_quantity = 0;

context Product :: addToCart(id , buyer\_id,status)

pre: product\_quantity > 0;

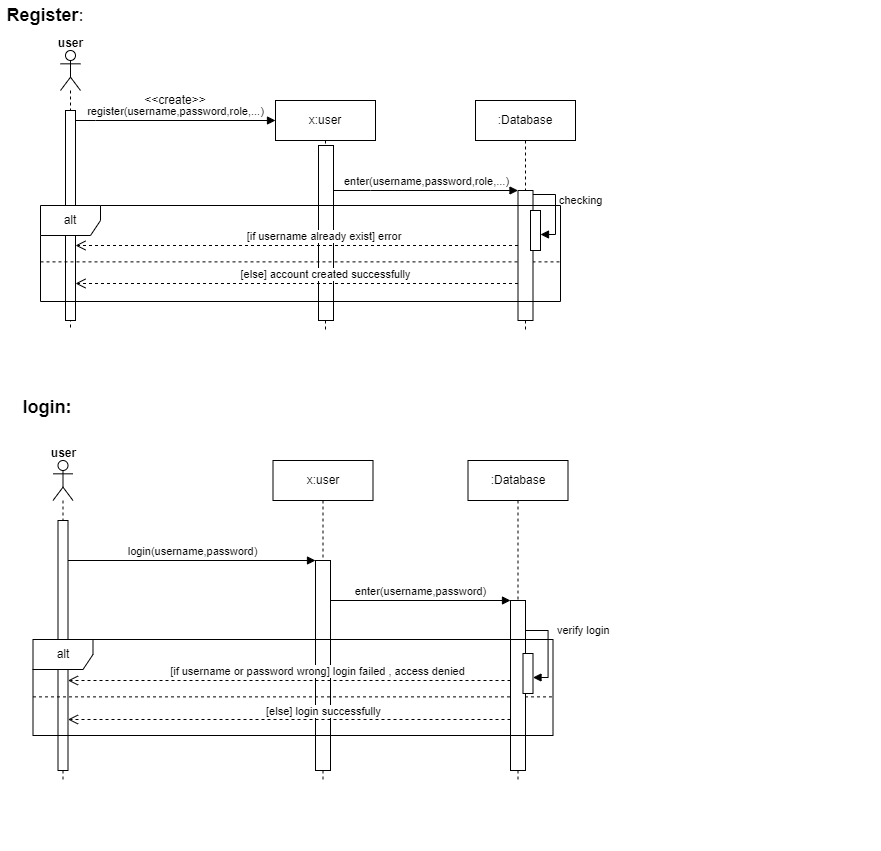
context User :: new\_order(id , buyer\_id,status)

post : product\_quantity = product\_quantity-1

context new\_order :: confirm\_order(order\_id)

pre : adding(flatNo, buildingNo, street, city, country, phone, email, creditCardHolderName,creditCardType,creditCardNum,expMonth,expYear,cvv,orderId,buyerId)

1. **Sequence Diagram**

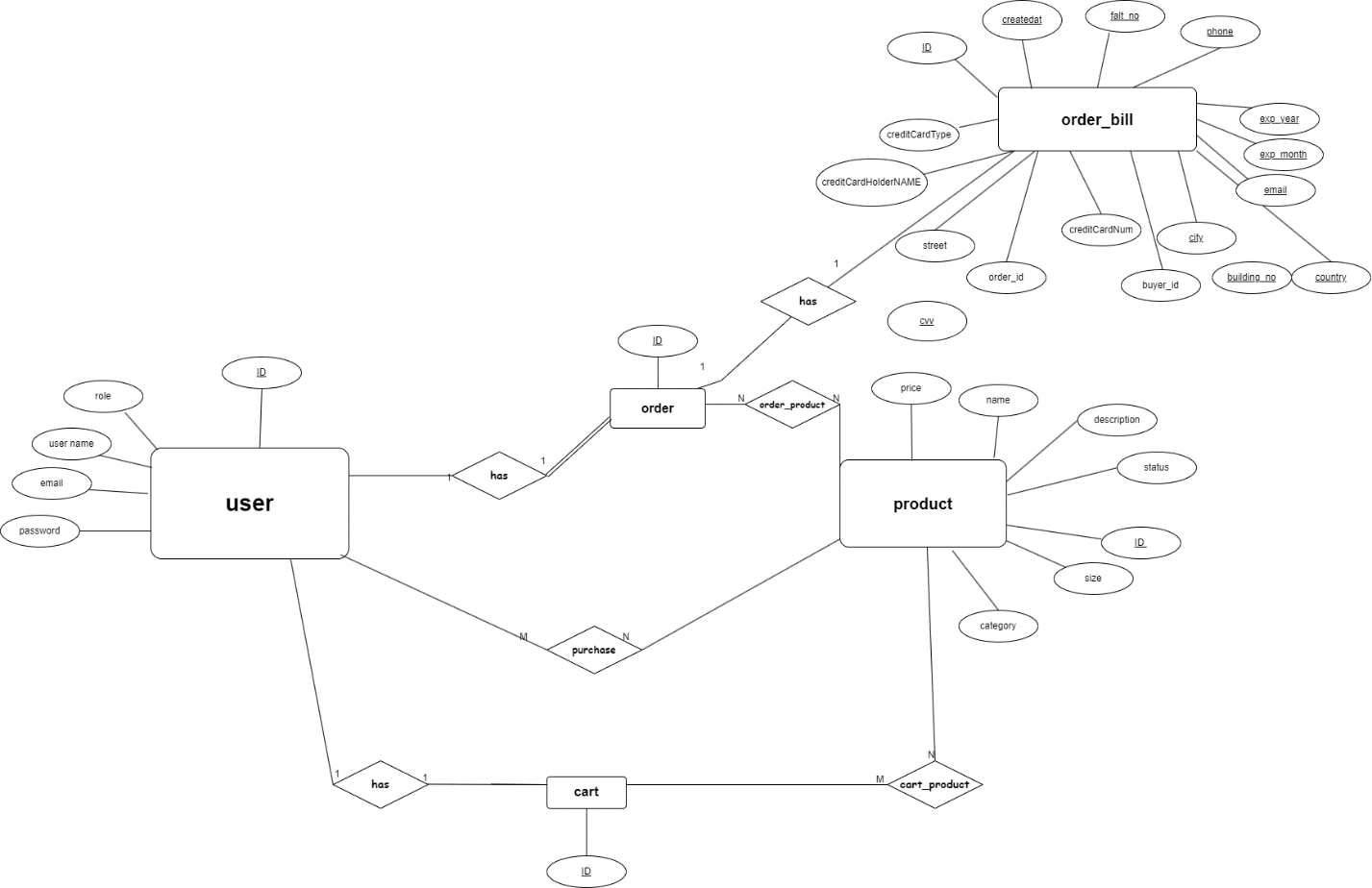
****

**A picture containing text, diagram, technical drawing, parallel

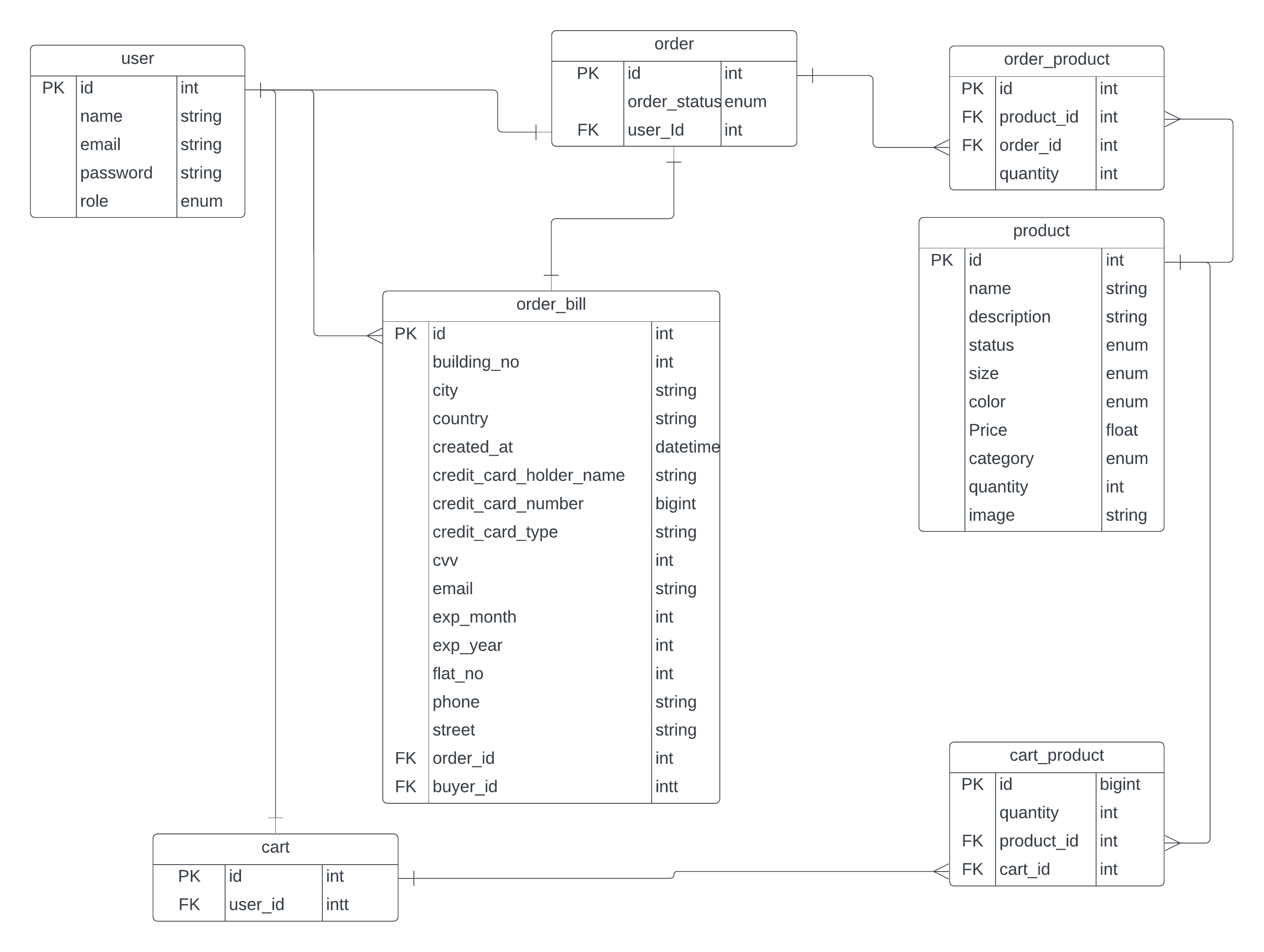
Description automatically generated**

1. **Database Specification**

**ERD**

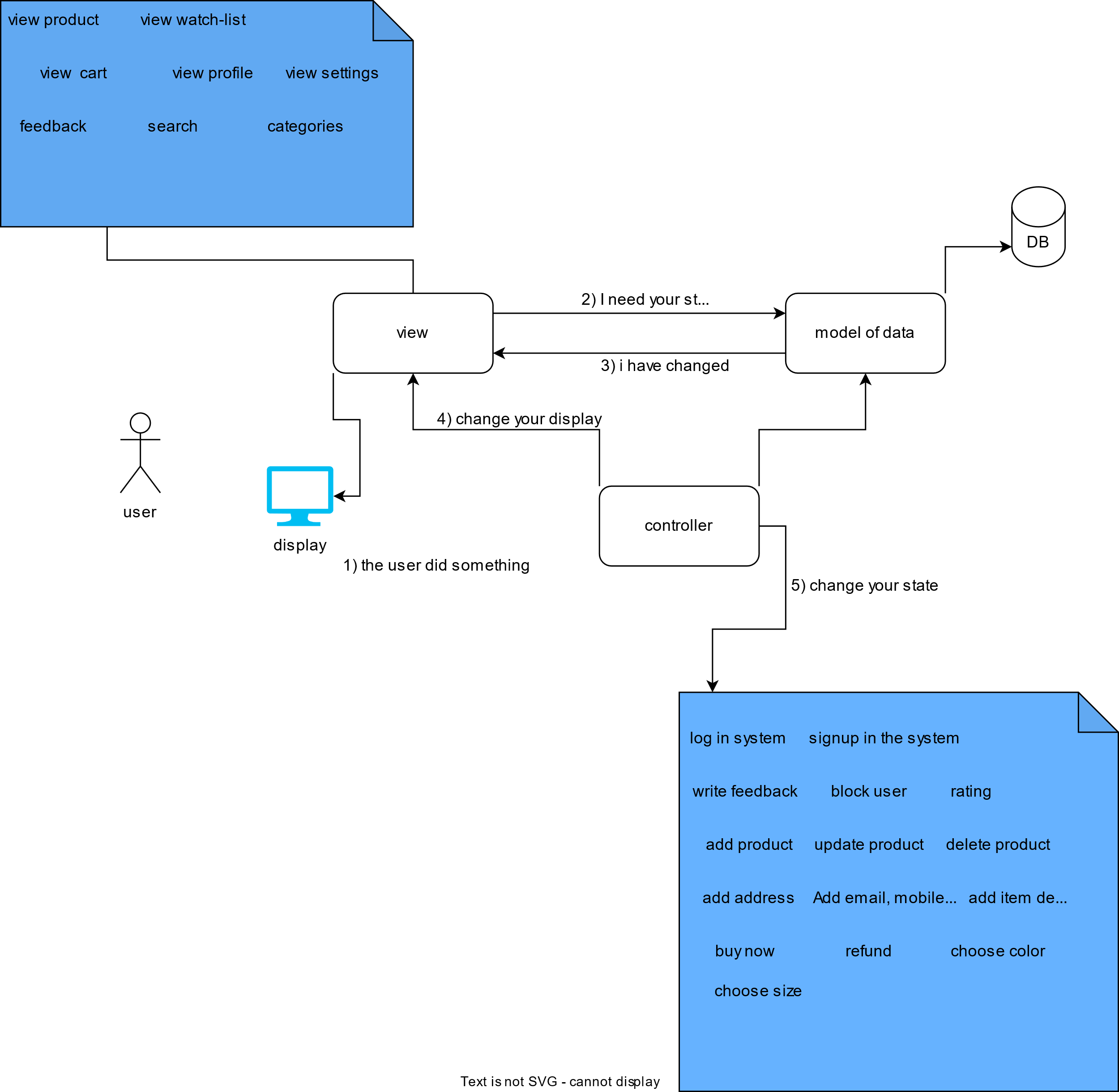
****

**Tables**

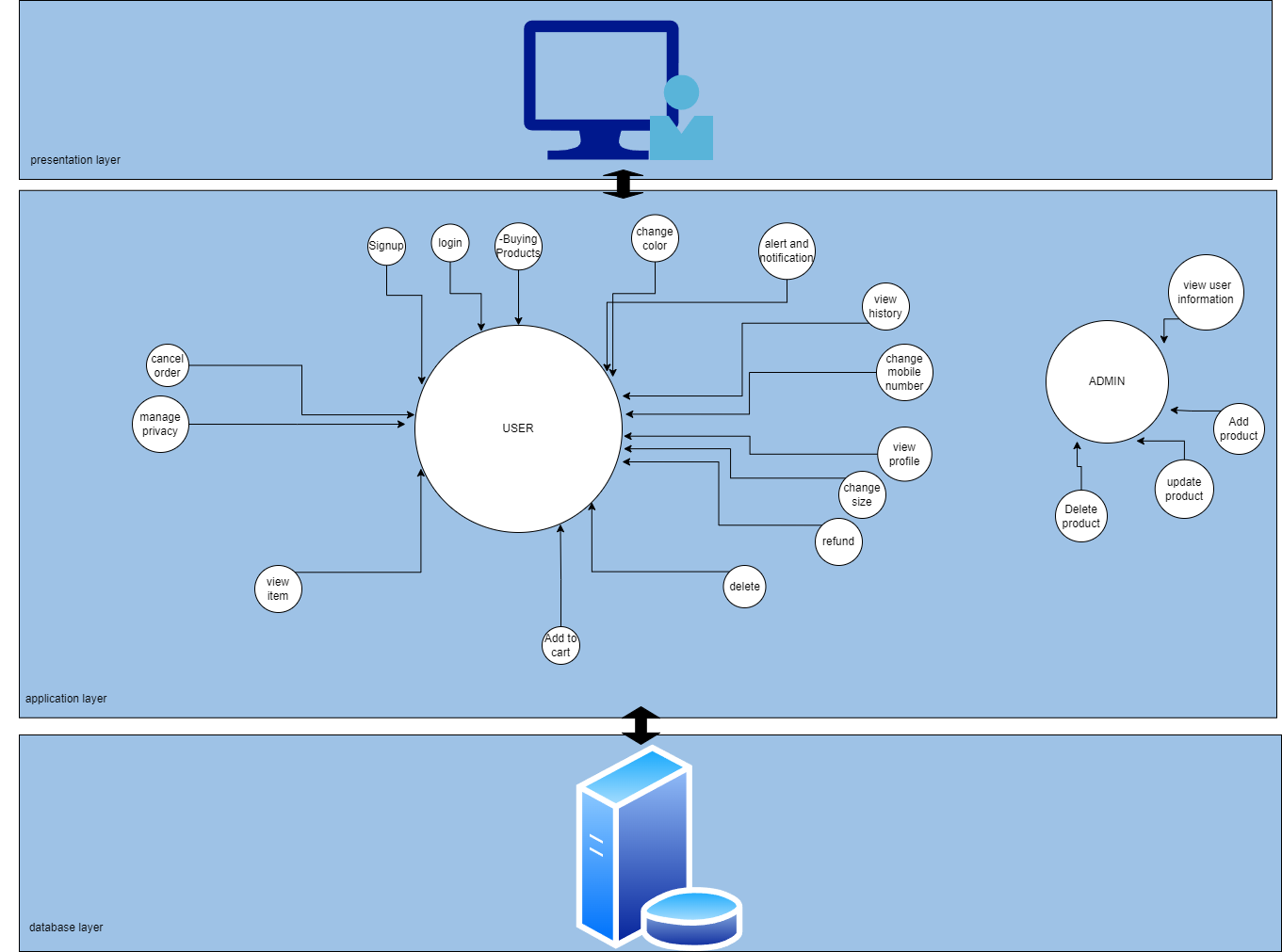
****

1. **System Architecture**

**MVC**

****

**Layer**

****