**S.W.2 project**

**Project Name: E-commerce**

**Our software project offers a user-friendly platform where you can log in, buy products, receive offers, filter, and search for products, and easily contact us. Simplify your product experience with our convenient and accessible solution.**

**-------------------------------------------------------------------------------------------------------------**

**Team Information:**

**ID NAME**

1. 20210286 حسن ابراهيم عبد المعز
2. 20210506 عبد الرحمن سعيد فؤاد
3. 20210209 آيه محمد عبد الرازق
4. 20210208 آيه محمد حسن
5. 20210224 بسمله رمضان محمود
6. 20210346 ريم سلامه عبدالسلام

**Functional REQ**

1. The application should allow the user to log in by entering the name and the password.
2. The application should allow the user to search for any item they want.
   1. The user chooses from the collection types.
   2. **If the product that the user chooses from the collection is not found in the collection, he will find (Sorry no products match your filter search ... Clear the filter and try again.)**

**3.** The application should allow the user to buy.

* 1. the user will add the product he wants to the cart and make an order.

1. The application shall allow the user to choose any category when buying.
2. the application should allow the user to log out.

-------------------------------------------------------------------------------------------------------------

Non-functional REQ

**Look and feel.**

1. The application shall use calm colors.

2. The application should use some animation

**Usability and humanity.**

**3.** The application shall be easy for all ages 12+.

4. The application shall be easy for most people without training.

5. The application shall be user-friendly and intuitive, with clear navigation and instructions. **Performance.**

6. the application should be able to buy or sell in 5 minutes for most of the users and older people in 10 minutes.

7. the application shall process the personal data that has been entered by the user in 30 seconds.

8. the application shall display the price after searching the bar code in 5 seconds.

9. the application shall be responsive and load quickly.

10. the application shall calculate the items in the cart in 1 second.

12. the application shall handle up to 5 users simultaneously.

13. the application shall be reliable and available all the time.

14. The application shall appear with a list of similar items that are searched by the user in 15 seconds.

15. The application shall save any transactions made by the user in 5 seconds.

16. the application shall be easy to maintain.

17. the application shall work with all operating systems.

18. The application shall be compatible with different devices, including smartphones, tablets, and desktop computers.

19. The application shall be scalable to accommodate a growing user base and increasing inventory.

20. the language used shall be polite, formal, and simple

21. the application shall not display any racist symbols or words.

22. the application shall respect the Buying and selling policy.

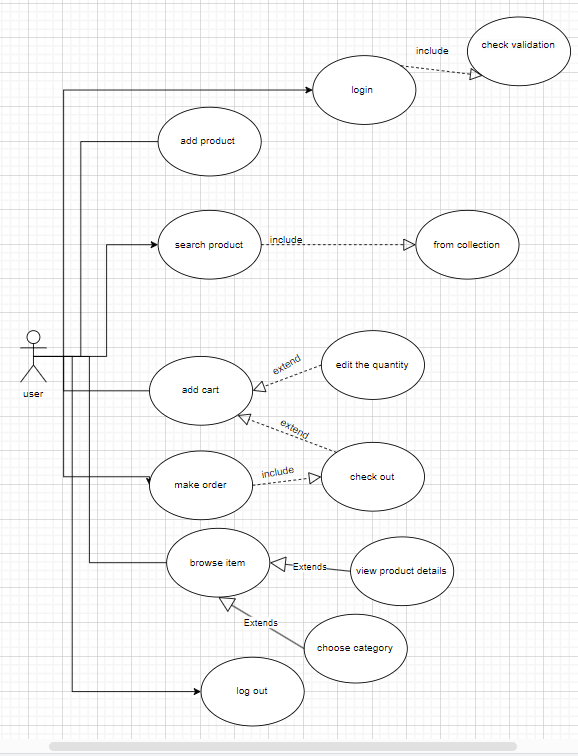
24. The application shall comply with data protection laws and regulations.

25. The application shall ensure that the data cannot be accessed modified or deleted by anyone but the user itself and the admin.

26. The application shall protect the user from any threats or attacks.

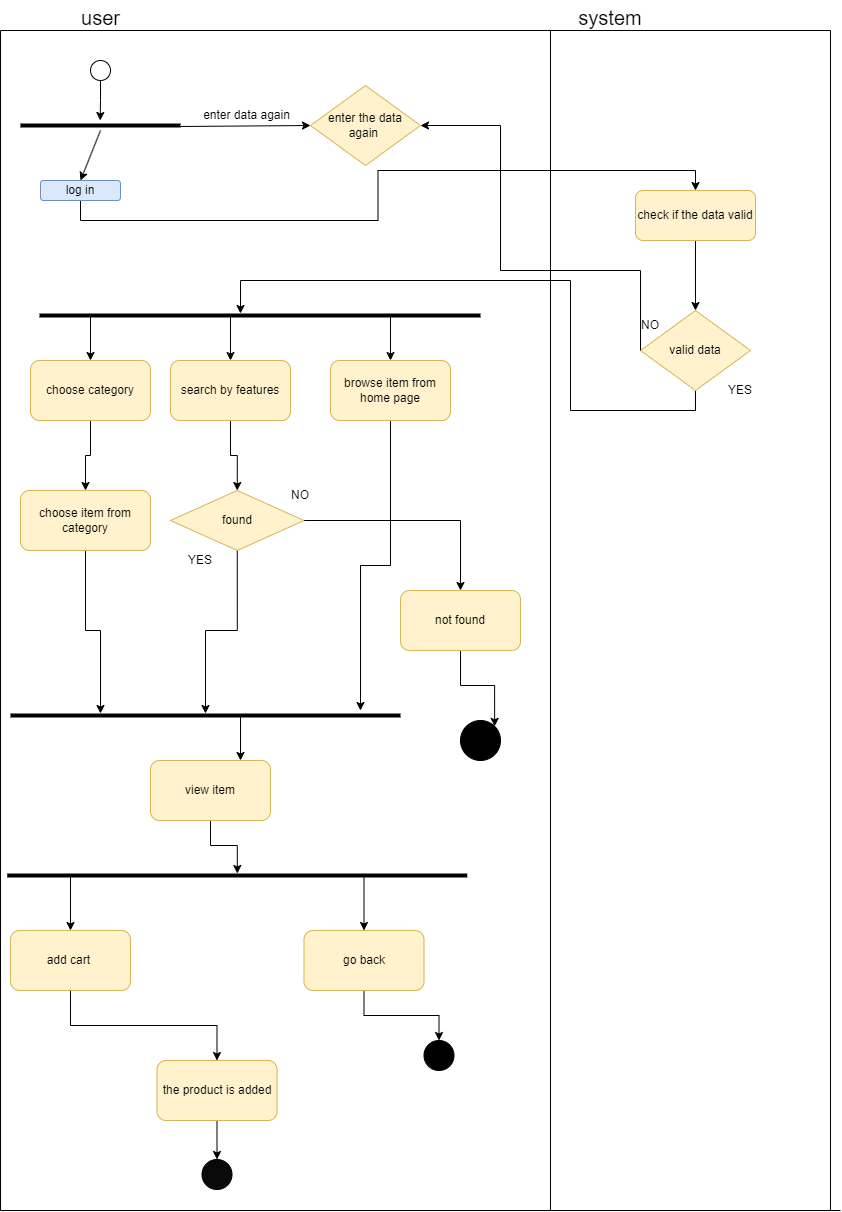
**Diagrams**

1. **use case diagram.**

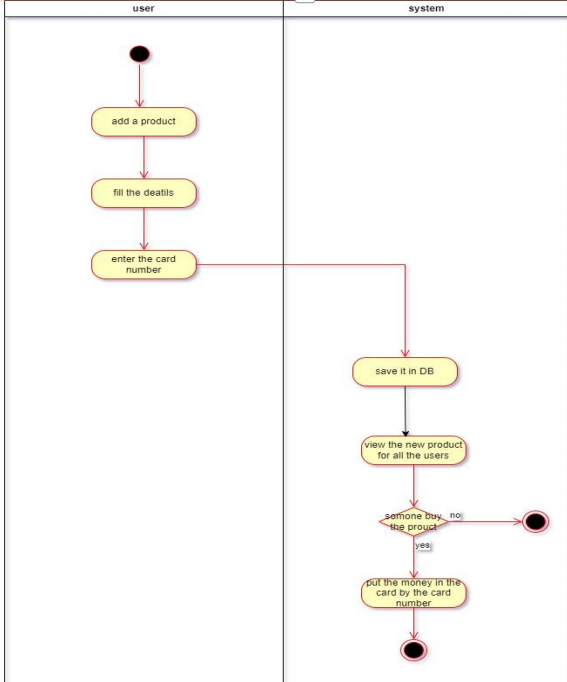
****

**2)Activity Diagram.**

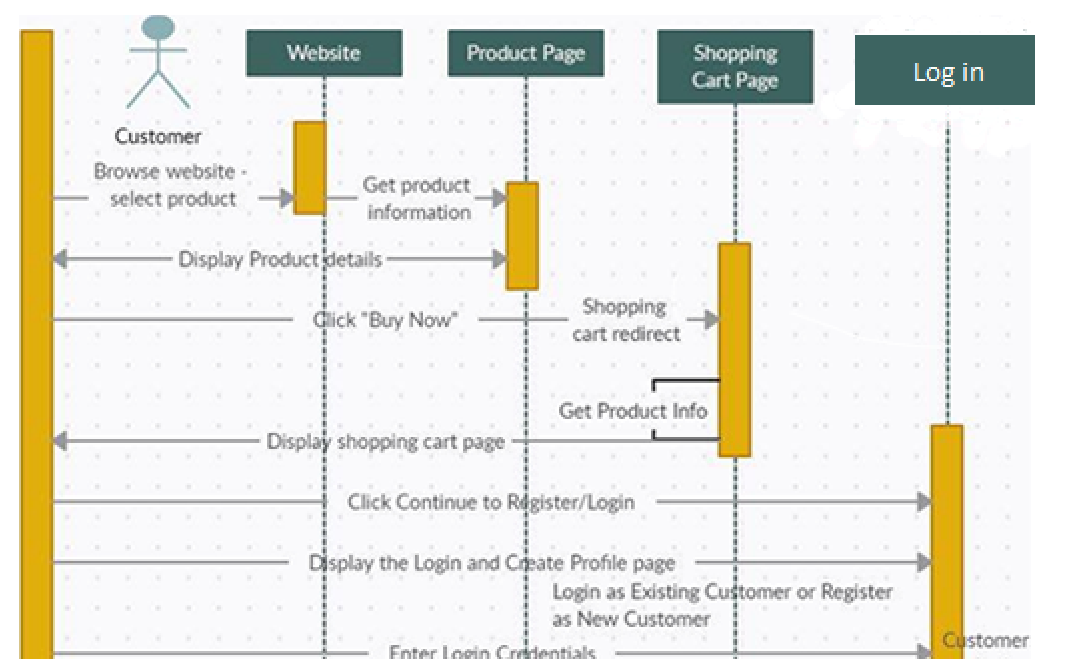
**1)**

****

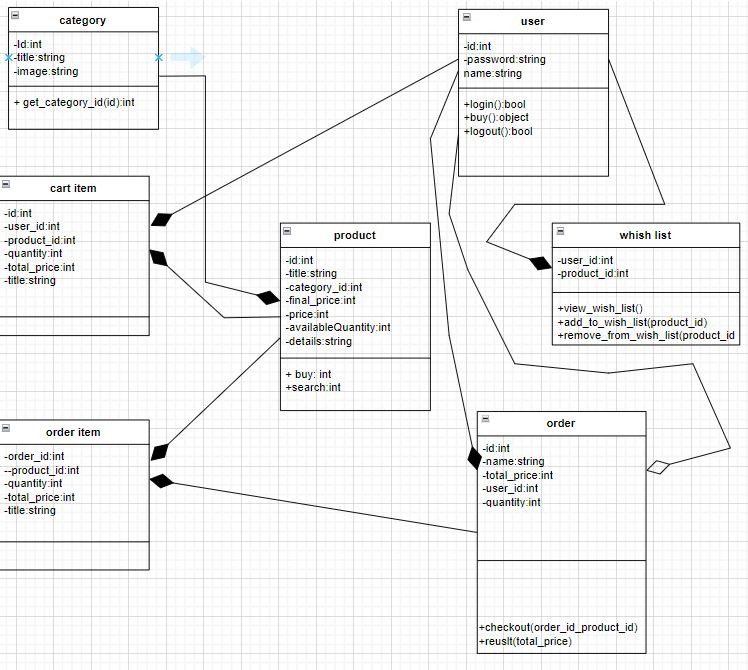
**2)**

****

**3)Sequence diagram**

****

**4)Class diagram**

****

**5)OCL**

1) context Product

inv: self.availableQuantity >= 0

explanation

 the available quantity of a product should always be greater than or equal to zero.

2) context User

inv: self.password.matches('.\*[A-Z].\*') and self.password.matches('.\*[a-z].\*') and self.password.matches('.\*[0-9].\*')

and self.password.matches('.\*[@,#,$,%,&].\*')

explanation

the password contains all letters even if they are lower or upper case also contains all numbers and other characters.

3)

context order item

inv: self.quantity >= 1

explanation

the minimum of the order item should be at least one.

4) context Order

inv: self.total = self.items->sum(price \* quantity)

explanation

it calculates the total cost of an order by summing the products of the price and quantity of each item in the order.

5) context Product

inv: self.price >= 0

explanation

the prices of the products can’t be negative.

6) context user

inv: not self.name.isEmpty() and not self.password.isEmpty()

explanation

you can’t login without writing the name and the password.

7)

context wishlist:: remove\_from\_wish\_list(product\_id)

pre: wishlist ->includes(product\_id)

explanation

We can't remove any product from the wish list if it is empty so with this OCL we check if the wish list is empty or not before the condition.

8)

context Order:: result(total\_price)

post: result = self.order->sum(price \* quantity)

explanation

After I choose the order, he is giving me the total price which equals the price multiply the quantity.

9)context user :: logout()

Pre: self.login =true

explanation

You can't log out in case if you are not logged in.

10) context product ::buy()

pre: product.available quantity > 0

explanation

he can't buy any product if the product available equals zero.

11)context product ::search()

pre: product.available quantity > 0

explanation

he can't search for any product if the product available equals zero.

12) context user :: buy()

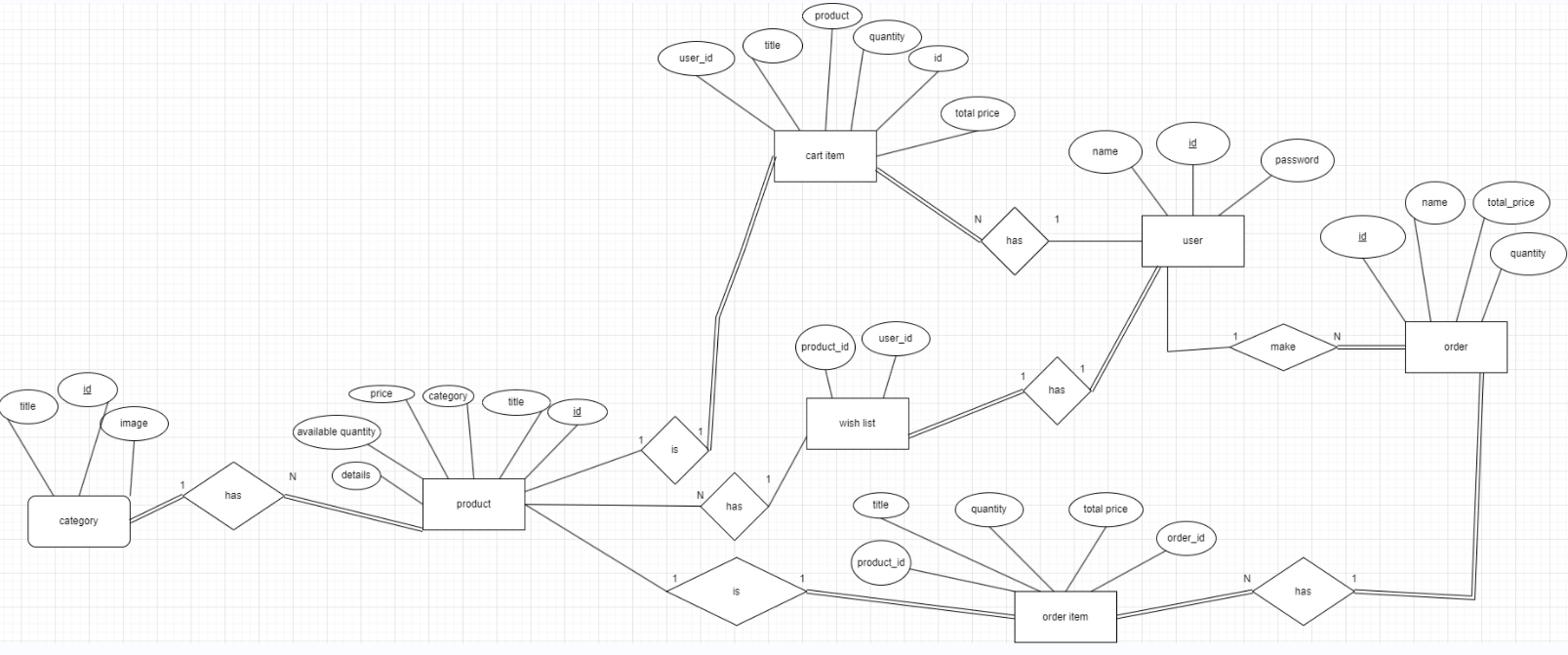
Pre: self.login =true

explanation

He can't buy any product if he is not logged in.

----------------------------------------------------------------------------

**6)ERD**



----------------------------------------------------------------------------