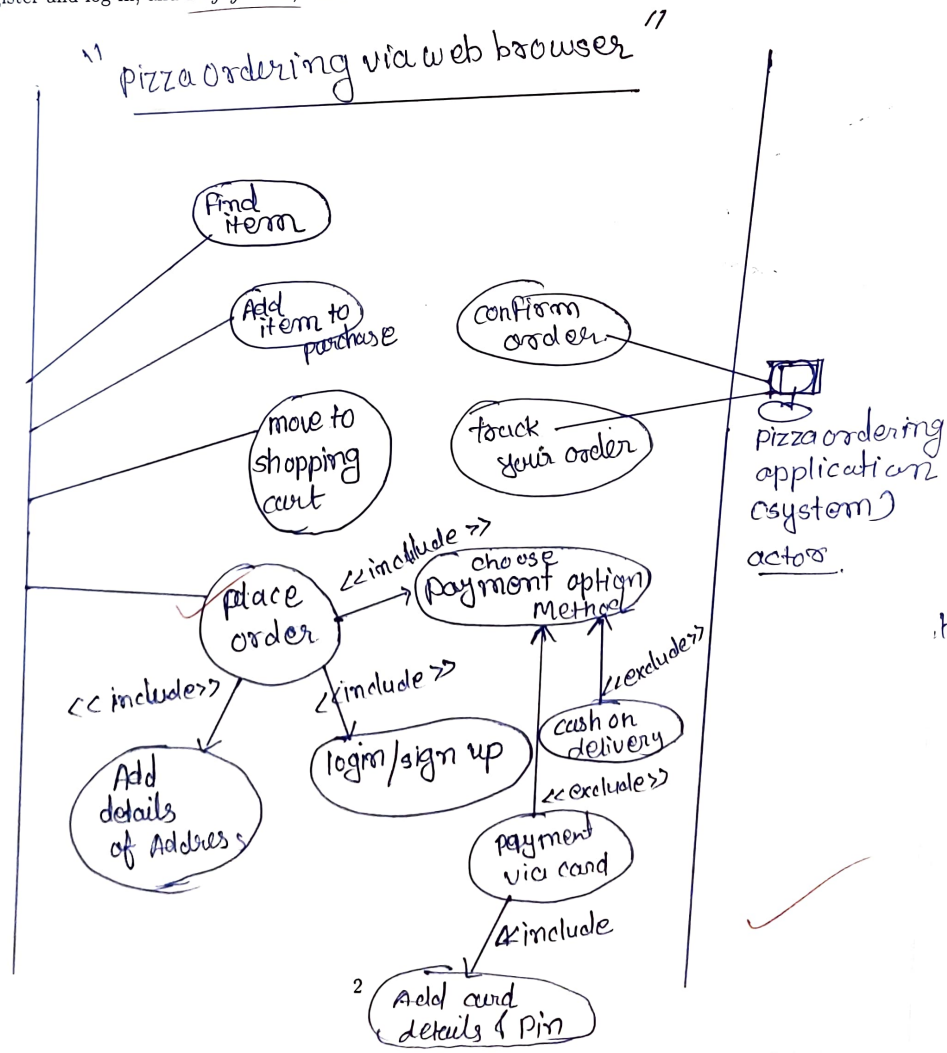
NAME: Ruchika Amin STUDENT NUMBER: 202101158 Page

1. The Pizza Ordering System

The Pizza Ordering System allows the user of a web browser to order pizza for home delivery. To place an order, a shopper searches to find items to purchase, adds items one at a time to a shopping cart, and possibly searches again for more items. When all items have been chosen, the shopper provides a delivery address. If not paying with cash, the shopper also provides credit card information. The system has an option for shoppers to register with the pizza shop. They can then save their name and address information, so that they do not have to enter this information every time that they place an order.

Develop a use case diagram, for a use case for placing an order, *PlaceOrder*. The use case should show a relationship to two previously specified use cases, *IdentifyCustomer*, which allows a user to register and log in, and *PaybyCredit*, which models credit card payments.



2. The following Use Case Description was written during the design of a card-based payment system: (10)

Use Case:

PayForProductWithCard

Actors:

User, Bank, Cashier

Trigger:

User inserts Card

Precondition:

The Cashier has taken the product from the User and entered the price of the product into the System.

Basic Flow:

1. The System reads information off of the Card
2. The System prompts the User to enter the type of Card (presenting the options "Credit" and "Debit" and "Cancel").
3. The System displays the amount of the payment (in dollars and cents).
4. The System prompts the User to approve the payment (presenting the options "Yes" and "No").
5. If the User approves the payment a transaction is sent to the Bank and the Cashier is notified to give the product to the User.

Extensions:

5a. If the User does not approve the payment, the Cashier is notified to keep the product and the System is reset.

Identify all of the problems in this use case description and write an improved version.

⇒ Use Case : ShoppingViaCard / BuyWithCard

Actors : User, Bank, Cashier.
↳ should be stackholder.

User & Cashier → Main Actors

Trigger : User inserts Card
User Approves for payment or not

Pre Condition : The cashier has taken all products that customer wants to buy and scan that all codes & total theire prices and add into system.

Basic Flow:

① The customer pick up the things that he/she wants to buy and give them to the cashier.

② The system reads information of the product like price & code & total them.

③ The system prompts the user to enter the type of card (credit / debit / cancel)

④ The system ask to user to ~~the~~ type the 'PIN' of their card.

⑤ And if the user approves the payment then transaction done & cashier have to give product to the customer.

5.a do not approve for payment then product will keep by the cashier.

5.b PIN of the card ~~will~~ may be entered wrong by the customer in that case also payment will not be done

2. customer will not get the product.

→ So the main change is to Add 'PIN' criteria for the customer to approve the payment for the product which is not given in question.

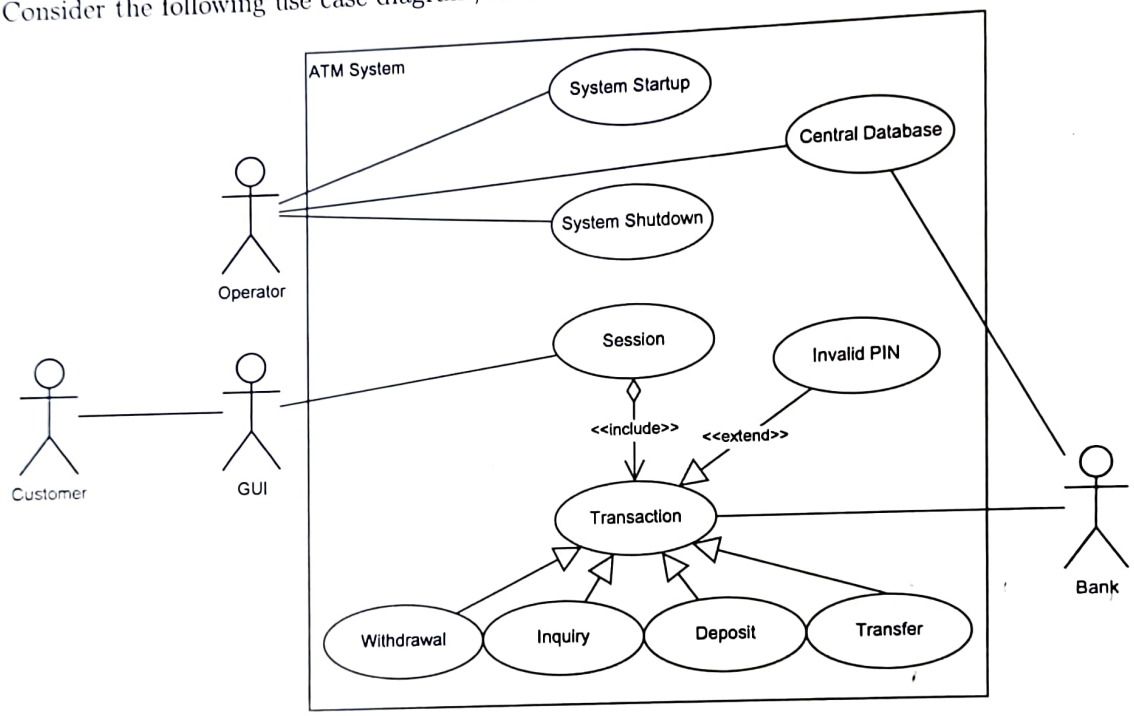
⟹ So the Improved System allows to add 'PIN' to the customer for their privacy and safe payment.

Extesions are ~~imp~~ incomplete.

~~2~~ for point ~~2~~ 4, 2

post conditions?.

3. Consider the following use case diagram, and mark whether the statements are *True* or *False*.



- According to the diagram, Transaction is an abstract superclass for Withdrawal, Inquiry, Deposit, Transfer and Invalid PIN. **[True/~~False~~]** True
- The relation between Invalid PIN and Transaction does not conform to the UML standard. [True/~~False~~] True
- The use case should clarify in what direction data is transferred to and from the Central Database. **[True/~~False~~]** True
- The Central Database should be moved outside the ATM System box, but the connections should be kept. **[~~True~~/False]** False
- The relation between the Customer and the GUI is not permitted in UML use case diagram syntax. **[True/~~False~~]** True
- The relations connecting the Operator, GUI and Bank to the ATM System are missing the arrows. **[~~True~~/False]** False

4. User Stories

- We are building an application for a business that sells products such as books, movies, music, and greeting cards. Assume a physical store. Your Product Owner has a story: *As a customer, I want to buy a product so that I can enjoy using it!* This story is a huge epic. The team needs to work with the product owner to split it. Also, specify the user stories representing the non-functional requirements of the applications. (10)
- Problem: Of the two user stories below, which was better written? Explain your answer, citing two specific reasons one is better than the other. (4)

Title: Rails Project Description: The system should be developed using Ruby on Rails, so that it will be less costly to develop and maintain. Estimate: 120 days	**Title: Manage Ads** **Description: As a system administrator, I want to be able to manage ads, so that I can remove expired and erroneous ads.** **Estimate: 2 days**

→ 'As a Customer, I want to buy a product so that I can enjoy using it!'

→ Above one is the given in question itself. Here we are ask to do the better version of it. So first we write out that what User want & need.

→ User story : As a _____, } who
I want _____, } what
So that _____. } why

Front of the project & Middle of the project there is always → User ←

→ Non functional Requirements :-

- ~~Safety~~ | security
- ~~Cost~~
- Privacy } → because of we are building an application for physical store. So there is obvious purchasing stuff and therefore there must be money transaction so privacy, security is More Important.

→ In the given description, user story says that user want something that can give him/her enjoyment.

→ Now we are looking for that "something" means "what ?"

→ An application for a business that sells

- books
- movies
- music
- greeting Cards

} the products which are offered by the app to the users.

→ Now for this we can write below user stories :-

First one is particular one :-

① As a Book-lover, (who has passion for reading book)
I want to buy new collection of books,
So that I can enjoy my free time to spend it on my hobby.

→ The second one is for general purpose :-

② As a Artist, (creative one)
I want to buy something innovative or interesting
So that I can explore my interests more.

→ this one is for whose have much interest in exploring new things such as books, movies, music, greeting cards that provided by this app.

(3) As a daughter,
I want to give greeting card to my parents
so that they can be more happy on their anniversary.

Better User story is second :- Manage Ads

As a system administrator, who?
I want to be able to manage ads, so that I can
remove expired & erroneous ads. what? why?

Estimate : 2 days.

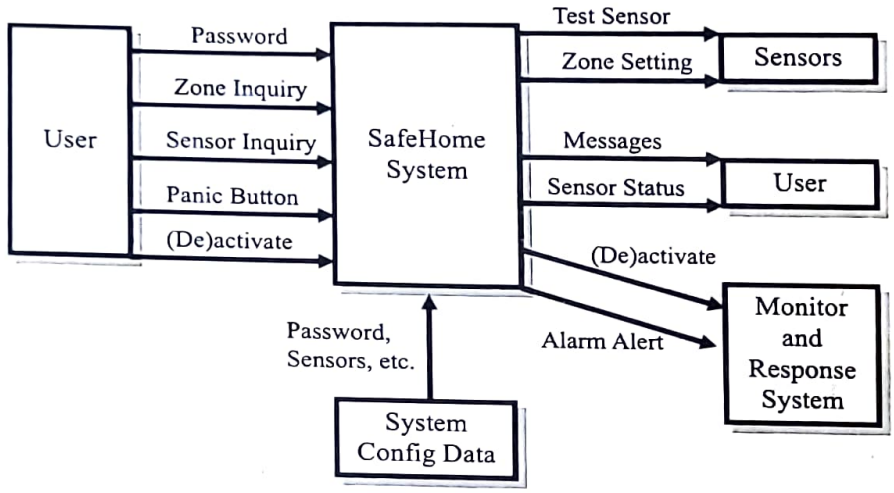
⟹ Here is the proper format of the how to ~~writing~~ user story.
The first ~~one~~ user story gives just normal
sentence which shows something can be done for
low cost / maintainance. But their is nothing described
in proper way of As a who? I want what?
and so that why?

4

⟹ Second user story is better for its proper
explanation and that we can understand that
"who" is going to be a system administrator
(who is managing ads) so "what" is going to be
a "manage ads" & why? → "why" is going to
be a remove expired & erroneous ads.

⟹ Second user story is better ~~for its~~ because
it contains all user's story aspects (who, what, why)

5. Compute the function point (FP) value for a safe-home functionality with the following information domain characteristics: (10)



Assume that weights are average and external complexity adjustment values are not important. *Weighting factor for average complexity is: (user inputs - 4; user outputs - 5; user inquiries - 4; files - 10; external interfaces - 7)*

Function point (FP) = UUSP * $\left(0.65 + 0.01 * \sum_{i=1}^{m} F_i\right)$

UUSP = Unadjusted Use case size point

For UCP = use case point

UCP = UUCP * TCF * EF → Environmental factor (8)

↳ unadjusted use case point

tecnical factor (1 to 13)

UUCP = UAW + UUCW

↓ Unadjusted actor weight

↳ unadjusted use case weight

UAW = (no. of simple actor × 1) +
(no of Avg. actor × 2) +
(no of complex actor × 3)

UUCW = (no. of simple use case × 5) +
(no. of Avg. use case × 10) +
(no. of complex use case × 15)

TCF : Technical factor

$TCF = 0.6 + \frac{\Sigma TF}{100}$ → total factor → Σ assign value * factor

EF = environmental factor

$EF : 1.4 + \left(-0.03 * \frac{\Sigma EF}{100}\right)$

Now for FP → $\frac{FOP}{PROD}$

UCP = UUCP * TCF * EF

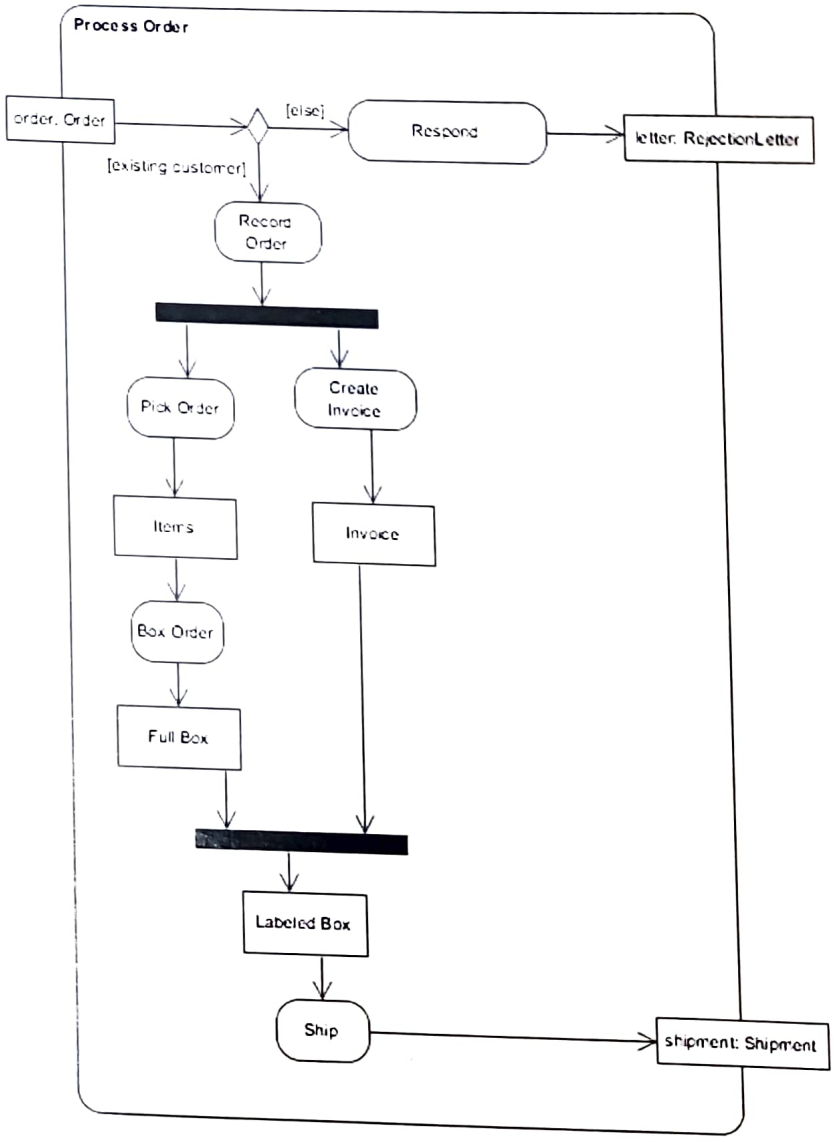
Ans ⇒ $\underbrace{4 \times 1}_{\text{user input}} + \underbrace{5 \times 1}_{\text{user output}} + \underbrace{4 \times 2}_{\text{user inquiries}} + \underbrace{10 \times 2}_{\text{Files}} + \underbrace{7 \times 3}_{EI}$

= 4 + 5 + 8 + 20 + 21

= 9 + 28 + 21

= 58 FP

6. Consider the following UML Activity Diagram.



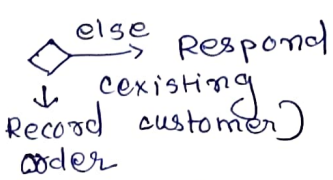
1 • Identify all of the activities in this diagram.
2 • Identify all of the object/data nodes in this diagram.
3 • Identify all of the actions in this diagram.
4 • Identify all of the decision nodes in this diagram.
5 • Identify all of the fork nodes in this diagram.
6 • Identify all of the join nodes in this diagram.
7 • Identify a control flow in this diagram.
8 • Identify a data flow in this diagram.
9 • Can "Pick Order" and "Create Invoice" occur at the same time?
10 • Can "Record Order" and "Ship" occur at the same time?

① Activities → Respond, Record order, pick order, create invoice, box order, ship

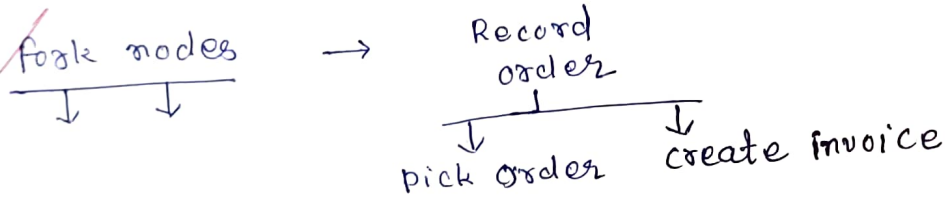
② object/data nodes → order:order, letter: Rejection letter, items, invoice, Full Box, labeled box, shipment:shipment

③ Actions → Respond, Record order, pick order, create invoice, box order, ship

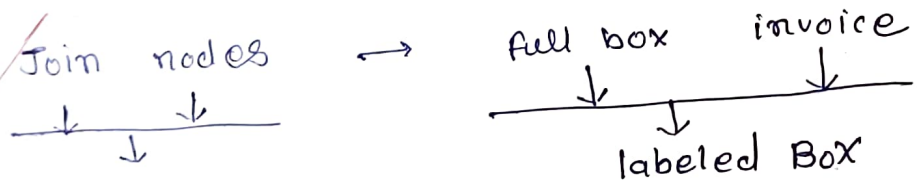
④ Decision node →



⑤ fork nodes →



⑥ Join nodes →



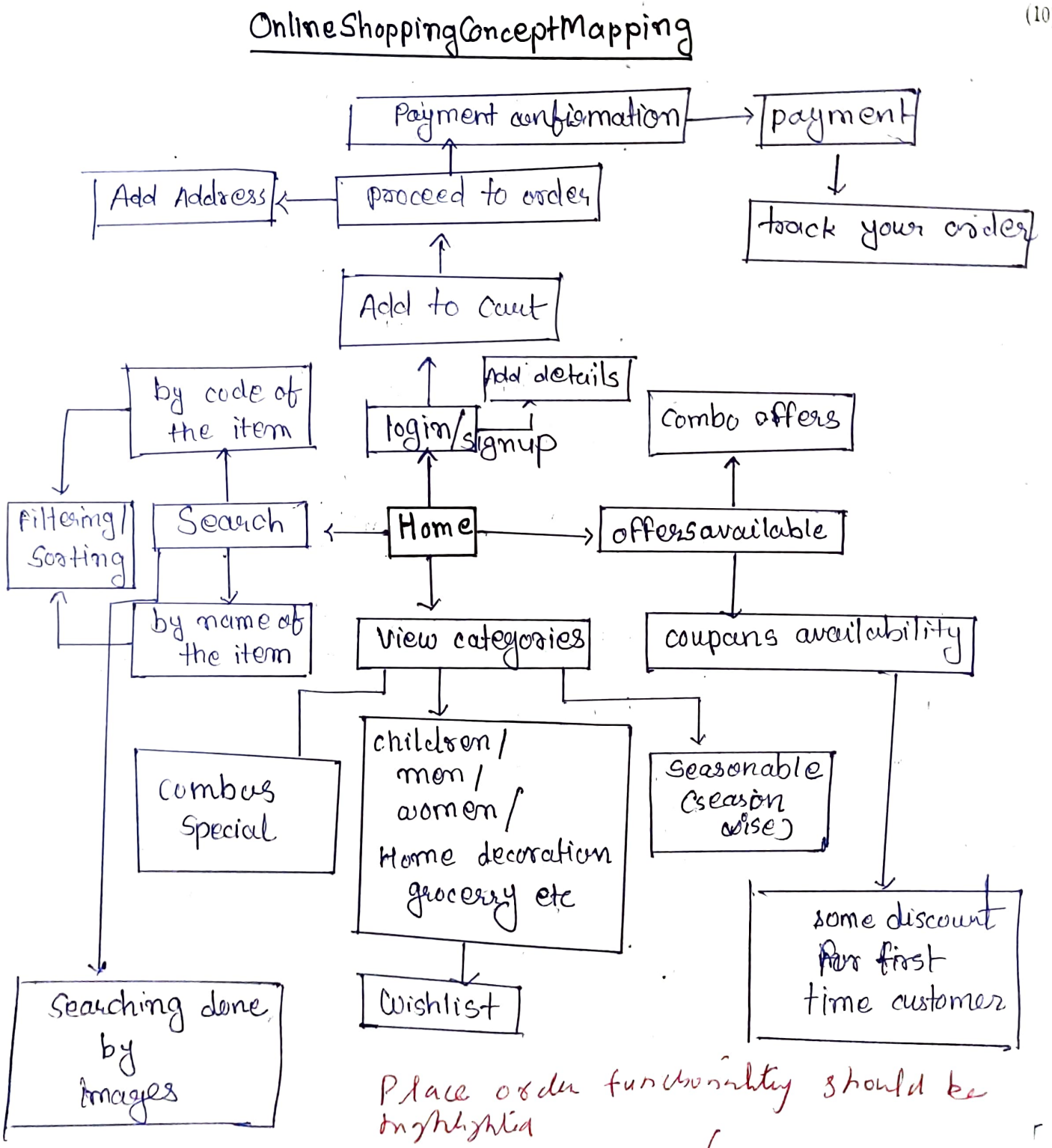
⑦ control flow → customer → Respond → letter.
↓
Record order
↓
pick order or create invoice
↓
Confirm order → ship

⑧ data flow → customer sign up Respond → letter
login ↓
pick order or create invoice
↓
payment → ship

⑨ Yes, occur at same time

⑩ NO, not necessary that Record order & ship occur at same time

7. Draw concept map for the "Place Order" functionality of the e-commerce for "Amazon/Flipkart etc." (10)



Place order functionality should be highlighted

6