IT313 Software Engineering (Autumn 2023-24)

Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT)

First-In Semester Examination, September 2023

INSTRUCTIONS:

- There are 8 double sided pages (16 printed pages). Ensure that you have all the pages.
- Answer **all questions**, writing clearly in the space provided.
- Proper justification of answers, wherever asked, is mandatory. If not stated marks will not be given.
- Write your name and student number **clearly** at the top of each page. If you do not follow this you will get zero right away.
- You have **1 hour and 30 minutes** to complete the examination.
- If you are making an assumption, clearly state the assumption and also reason for making that assumption.
- Marks for each question are indicated in brackets at right. You may use point form for your answers, but make sure the points are clear and unambiguous. I am more interested in your thought process.
- *Do not ask for sheets (supplementary/full) for rough work. Last two pages are reserved for rough work only.*
- Your answers should be on the question paper in the space provided.

FOR MARKER'S USE ONLY

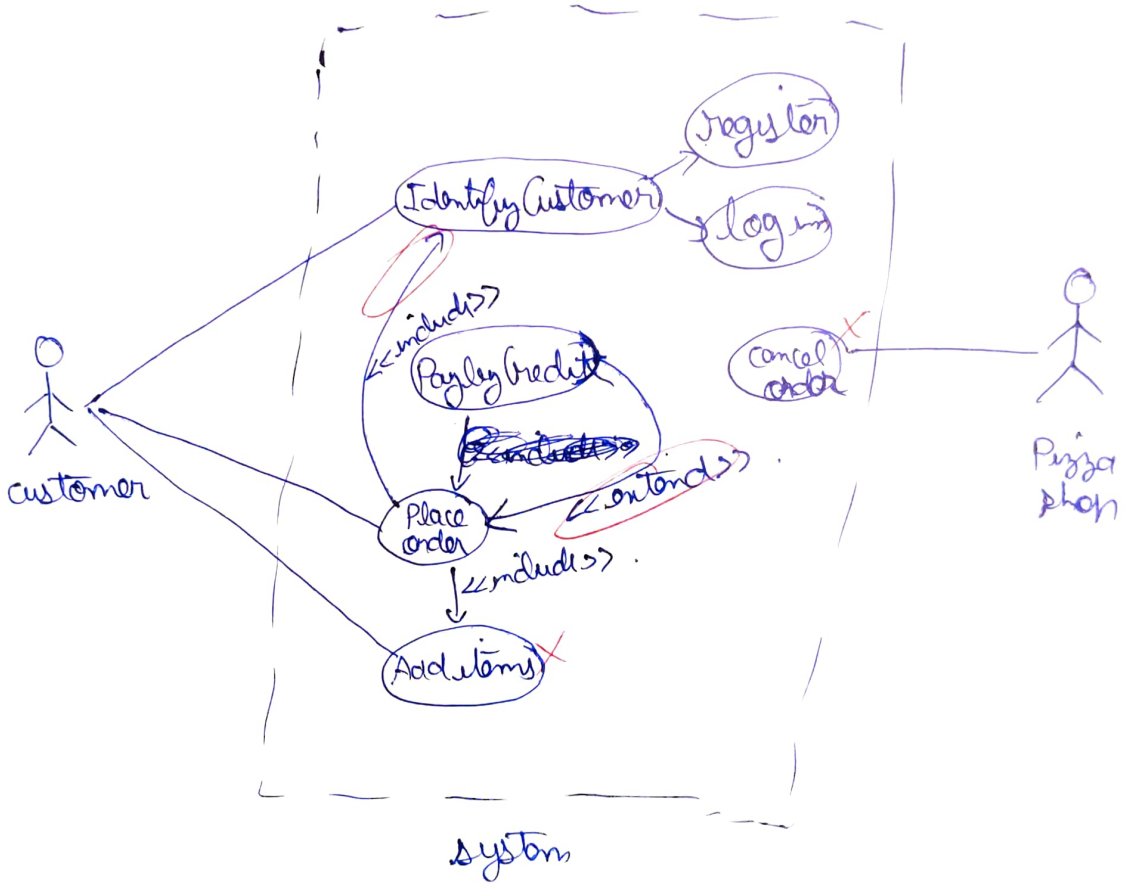
Question	Possible	Received
1	10	7
2	10	~~8~~ 10
3	06	3
4	14	8
5	10	0
6	10	7.5
7	10	5
TOTAL	70	40.5

NAME: Yuvraj Chaudhari

1. The Pizza Ordering System

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

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



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.

1) If the system can't read from the card it should through error
2) If the card is ~~stolen~~ injured or ~~to~~ reported lost the system ~~shows~~ should display the same.
3) PIN is not being entered after selecting type of card.
4) If the payment is approved the system should show a successfull message.

NAME:

N

Extensions continued -

4.a) If the pin is not approved it should show error.

5.a) If user selects 'No' it should stop the payment

2.a) If user selects 'cancel' it should cancel transaction

Post condition :-

The cashier gives the product to the User.

4

Improved Version:

Use Case:
Pay for Product With Card.

Actors:
User, Bank, Cashier

Trigger:
User inserts Card

Preconditions:
Cashier has taken the product from user and entered the price of product into the system.

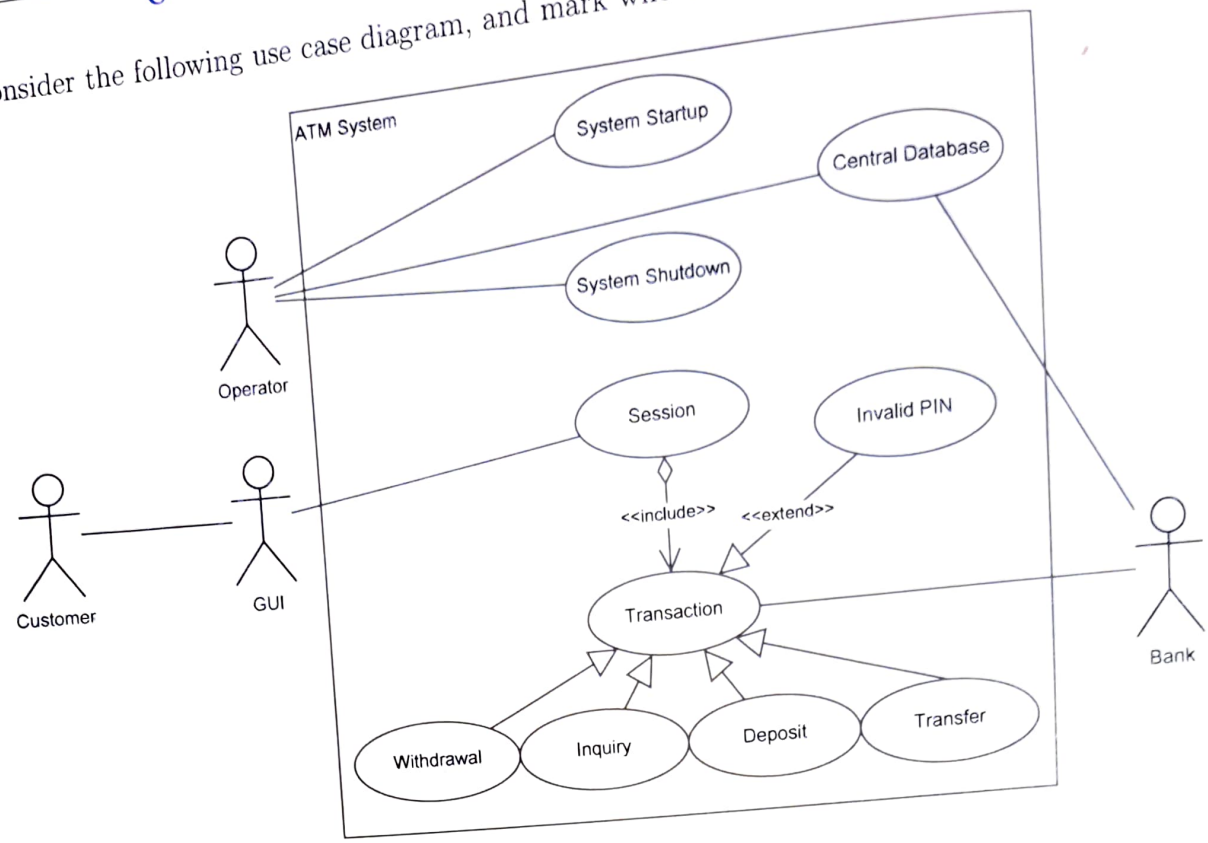
Basic Flow:-
1. ~~If the~~ System reads information off the Card.
2. System Prompts the user to select type of card or cancel.
3. System displays the amount to be paid in dollar and cents
4. User enter his card PIN.
5. System prompts the user to approve payment by giving options for (YES) or (NO)
6. User approves the payment and transaction is sent to the cashier organization Bank and Cashier is notified to give the Product.

Extensions:
1.a) If the system can't read the card it should eject the card and show the error
1.b) If the card is expired or lost show the ~~error~~ message

continued.

NAME:

3. Consider the following use case diagram, and mark whether the state



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

3

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

4.1)

→ Stories

1) As a customer, I want to buy a book so that I can read it.

2) As a customer, I want to buy a movie so that I can watch it.

3) As a customer, I want to buy music so that I can listen it.

4) As a customer, I want to buy greetings cards so that I can send them.

5) As a customer, I want the latest book, movie or music.

6) As a Product owner, I want my customers to select a product from lowest to highest price order so that he can see all available options in his budget.

7) As a product owner, I want to sell product to customers that are virus free so they trust the sh

→ Non-functional Requirement.

8) As a product owner, I want my products gallo to be as large as ~~poo~~ possible so to keep retentio

→ Non-functional Requirement.

9) As a customer, I want to buy the product through credit card so that I don't have to carry cash all time

→ Non-functional requirement

6

4.2) I think the latter written user story is Manage Ads

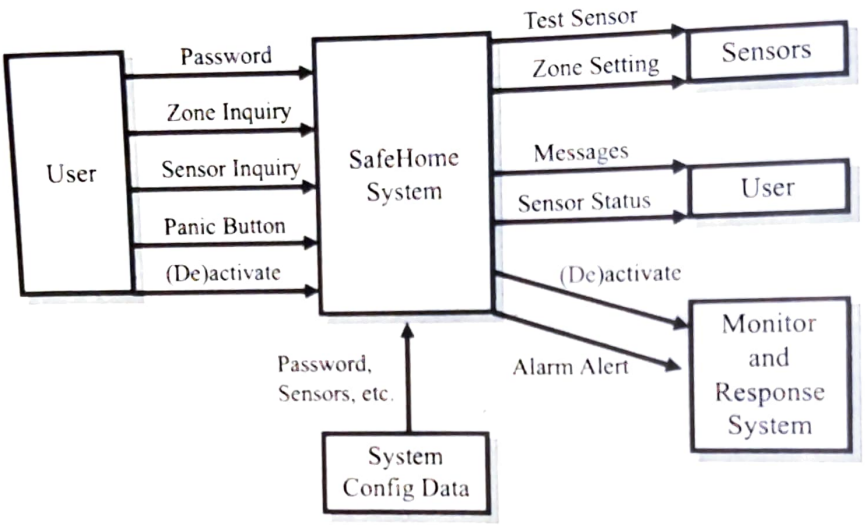
→ It has the temple [Who... What... Why...] of user stories which the Rails project doesn't have.

→ In rails project the stories can be splitted further to other stories.

reasons?

(2)

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



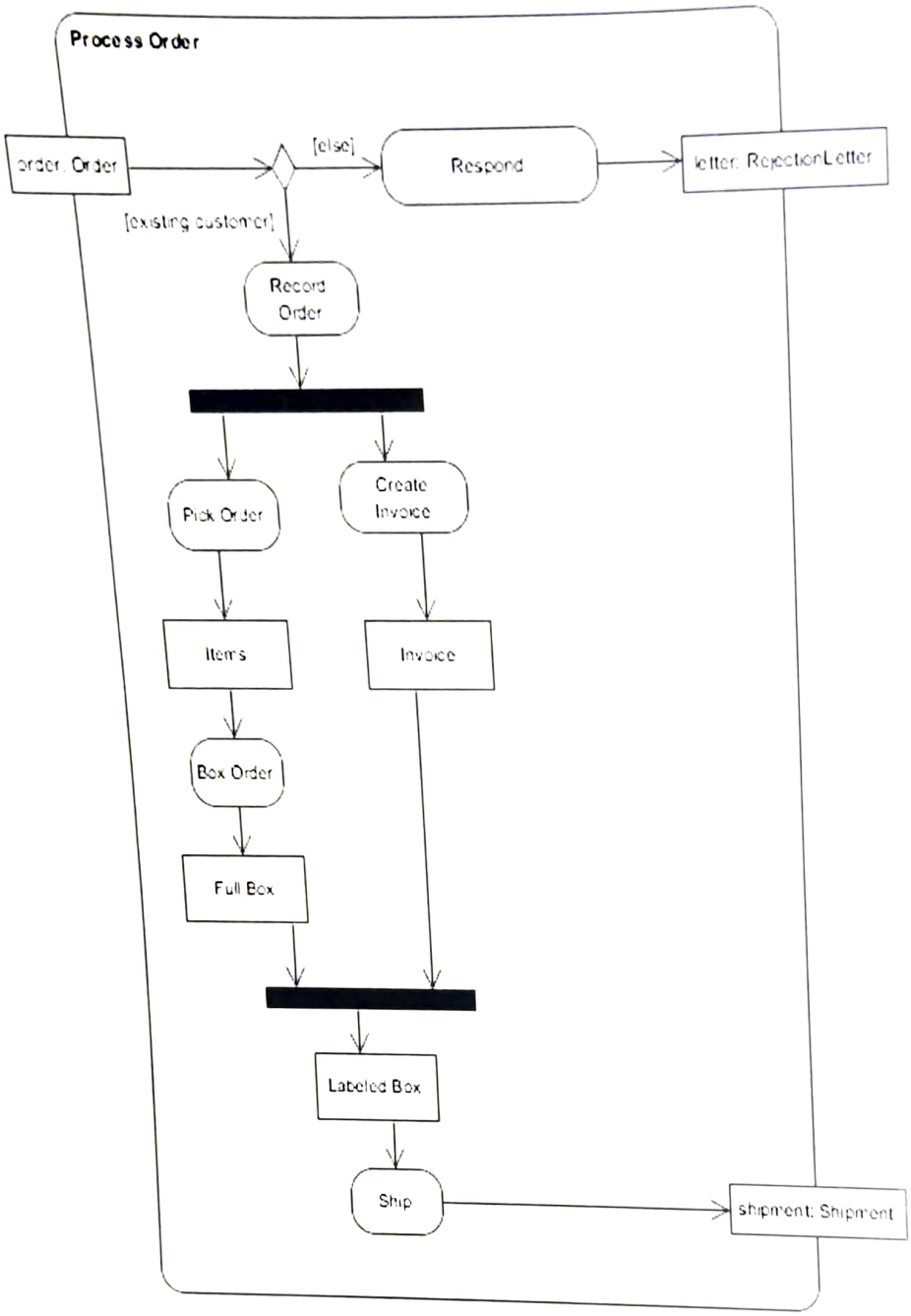
Assume that weights are average and external complexity adjustment values are not important. *Weighting factor for average complexity is: (user inputs - 4; user outputs - 5; user inquiries - 4; files - 10; external interfaces - 7)*

$\Rightarrow$ FP = (4x3) + (5x6) + (2x4) + 7 + 10

= 12 + 30 + 8 + 17

= 67

onsider the following UML Activity Diagram.



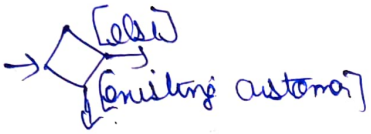
- Identify all of the activities in this diagram.
- Identify all of the object/data nodes in this diagram.
- Identify all of the actions in this diagram.
- Identify all of the decision nodes in this diagram.
- Identify all of the fork nodes in this diagram.
- Identify all of the join nodes in this diagram.
- Identify a control flow in this diagram.
- Identify a data flow in this diagram.
- Can "Pick Order" and "Create Invoice" occur at the same time?
- Can "Record Order" and "Ship" occur at the same time?

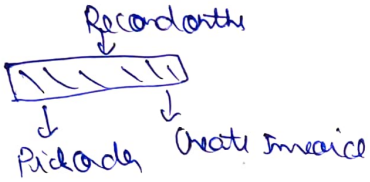
Yuning STUDENT NUMBER: 20210482

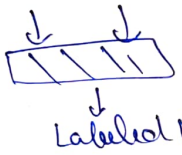
1) Order, Respond, Reject Letter, Record Order, Pick Order, Items, BoxOrder, Full Box, Create Invoice, Invoice, LabeledBox, Ship, Shipment

2) Rejection Letter, Order, Items, FullBox, Invoice, LabelledBox, Shipment

3) Respond, Recordorder, PickOrder, Boxorder, Creat Invoice Ship.

4) 

5) 

6) 

7) PickOrder → Items

8) Items → Boxorder

9) ~~No~~, Yes

10) No, because they are inside the fork and join

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