Advanced Software Engineering
Problem Set-1 (PS-1)

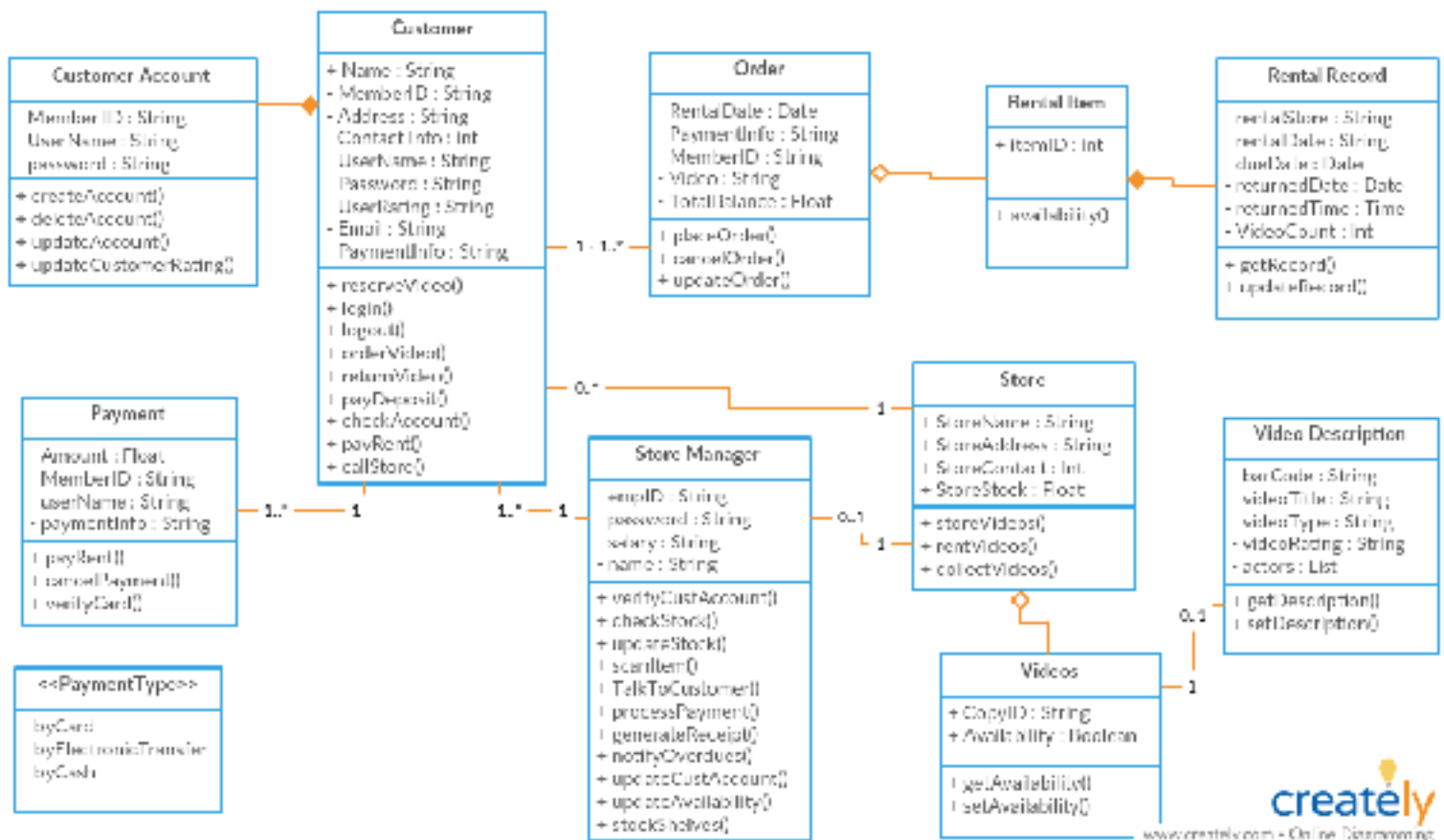**Name: Sneha Mishra**
Class ID: 21

Rent Video Use Case
A customer wishes to rent a video that is picked from the store's shelves or that have been previously reserved. Provided the customer has a non-delinquent account, the tape is rented out once the payment has been received. If the tape is not returned in a timely fashion, an overdue notice is mailed to the customer.
A customer asks an employee about video availability (including a reserved video) or picks one or more videos from the shelves. The videos and the membership card are scanned and any delinquent or overdue details are brought to the employee's attention. If the customer does not have a delinquent rating, then he/she can rent up to a maximum of eight videos. However, if the rating of the customer is "unreliable" then a deposit of one rental period for each video is requested. Once the amount payable is received, the stock is updated and the videos are given to the customer together with a receipt. The customer may pay by cash, credit card, or electronic transfer. Each rental record stores (under the customer's account) the check-out and due dates for each video together with the identification of the employee. A separate rental record is created for each rented video. The use case will generate an overdue notice to the customer if a video has not been returned within two days of the due date, and a second notice after another two days (and at that time the customer is noted as "delinquent").
1.Draw a class diagram for the video store, and be sure to label all associations with appropriate multiplicities.
2.Describe the difference between aggregation and composition relations using the video store.
3.Draw a sequence diagram for renting a movie from a video store.
4.Draw a state diagram that describes the lifetime of a video tape in a video store.
5.Draw an activity diagram that describes renting a movie from a video store them.

1. Class Diagram of a Video Store:

Customer Account

Member ID : String
User Name : String
password : String

+ createAccount()
+ deleteAccount()
+ updateAccount()
+ updateCustomerRating()

Customer

+ Name : String
- MemberID : String
- Address : String
  ContactInfo : int
  UserName : String
  Password : String
  UserRating : String
- Email : String
  PaymentInfo : String

+ reserveVideo()
+ login()
+ logout()
+ orderVideo()
+ returnVideo()
+ payDeposit()
+ checkAccount()
+ payRent()
+ callStore()

Order

RentalDate : Date
PaymentInfo : String
MemberID : String
- Video : String
- TotalBalance : Float

+ placeOrder()
+ cancelOrder()
+ updateOrder()

Rental Item

+ ItemID : int

+ availability()

Rental Record

rentalStore : String
rentalDate : String
dueDate : Date
- returnedDate : Date
- returnedTime : Time
- VideoCount : int

+ getRecord()
+ updateRecord()

Payment

Amount : Float
MemberID : String
userName : String
- paymentInfo : String

+ payRent()
+ cancelPayment()
+ verifyCard()

Store

+ StoreName : String
+ StoreAddress : String
+ StoreContact : int.
+ StoreStock : Float

+ storeVideos()
+ rentVideos()
+ collectVideos()

Video Description

barCode : String
videoTitle : String
videoType : String
- videoRating : String
- actors : List

+ getDescription()
+ setDescription()

Store Manager

+ empID : String
  password : String
  salary : String
- name : String

+ verifyCustAccount()
+ checkStock()
+ updateStock()
+ scanItem()
+ TalkToCustomer()
+ processPayment()
+ generateReceipt()
- notifyOverdues()
+ updateCustAccount()
+ updateAvailability()
+ stockShelves()

<<PaymentType>>

byCard
byElectronicTransfer
byCash

Videos

+ CopyID : String
+ Availability : Boolean

+ getAvailability()
+ setAvailability()

2. Differences between aggregation and composition relations using the video store are:
Aggregation -> It is a special kind of directed Association, also called as **"has-a"** relation. If an object of class A <u>is a part of</u> an object of class B, then there is no condition such that the part of object must exist, both objects can exist independently. Few examples for aggregation related to the video store example are :
A) Store - Video : Videos are stored in the store, but the lifetime of the Video object is independent of the Store object. Even if the store object is destroyed, the videos object still exist.
B) Store - Manager/Employee : A manager/ employee still exist even if the store is destroyed. The lifetimes of the two objects are independent.
C) Order - Rental Item : Same is the case with the objects of order and rental item, their lifetimes are independent and the objects exist even if the composite (order) object is destroyed.

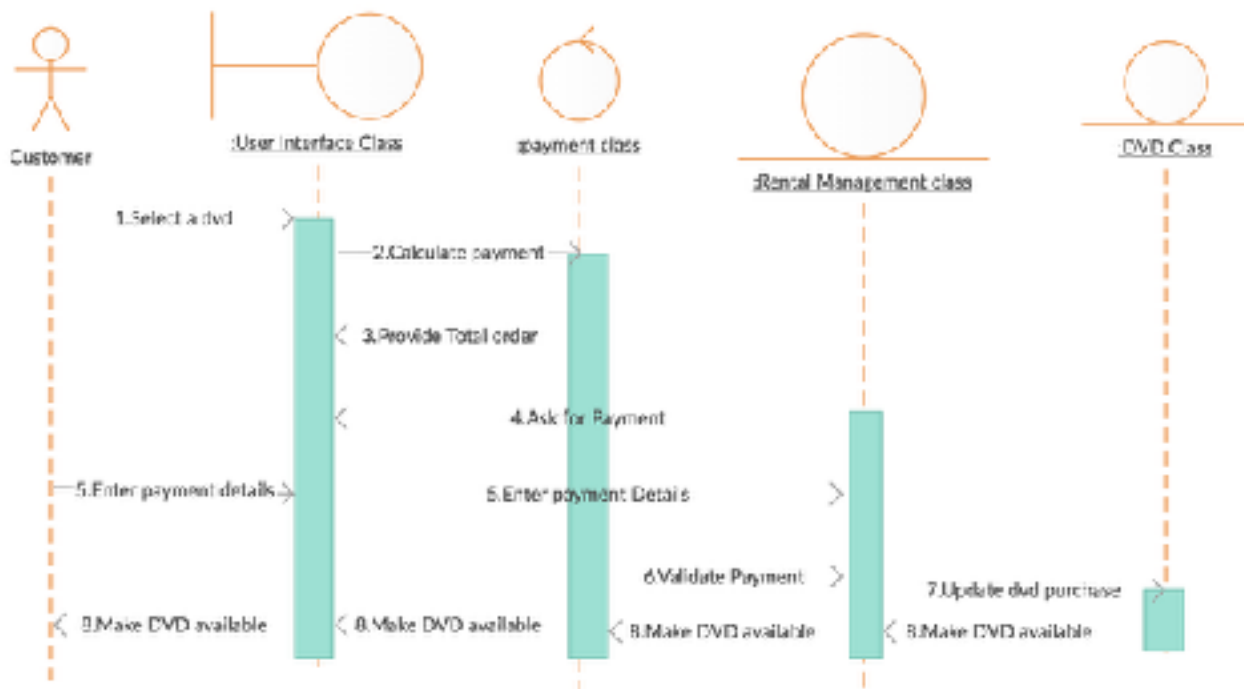Represented as - A hollow diamond at the has's end (Store).

Composition -> It is a stronger form of Aggregation, also called as **"part-whole"** relation. This relation is more restrictive since the lifetime of the part is dependent on the whole, the composed objects cannot exist without the other object. The composite manages the creation & destruction of its parts. Few examples related to the video store example are :
A) Rental item - Rental record : the rental record cannot exist unless there is a rented item. The object of a rental record is destroyed with the destruction of the rental item object.
B) Video - Video Description : the description of a video cannot exist unless there is a video. Also the video object once destroyed, destroys the video description object.
C) Customer - Account : The Account object cannot exist unless we have a customer object. The lifetime of the account object is dependent on the lifetime of the customer object.
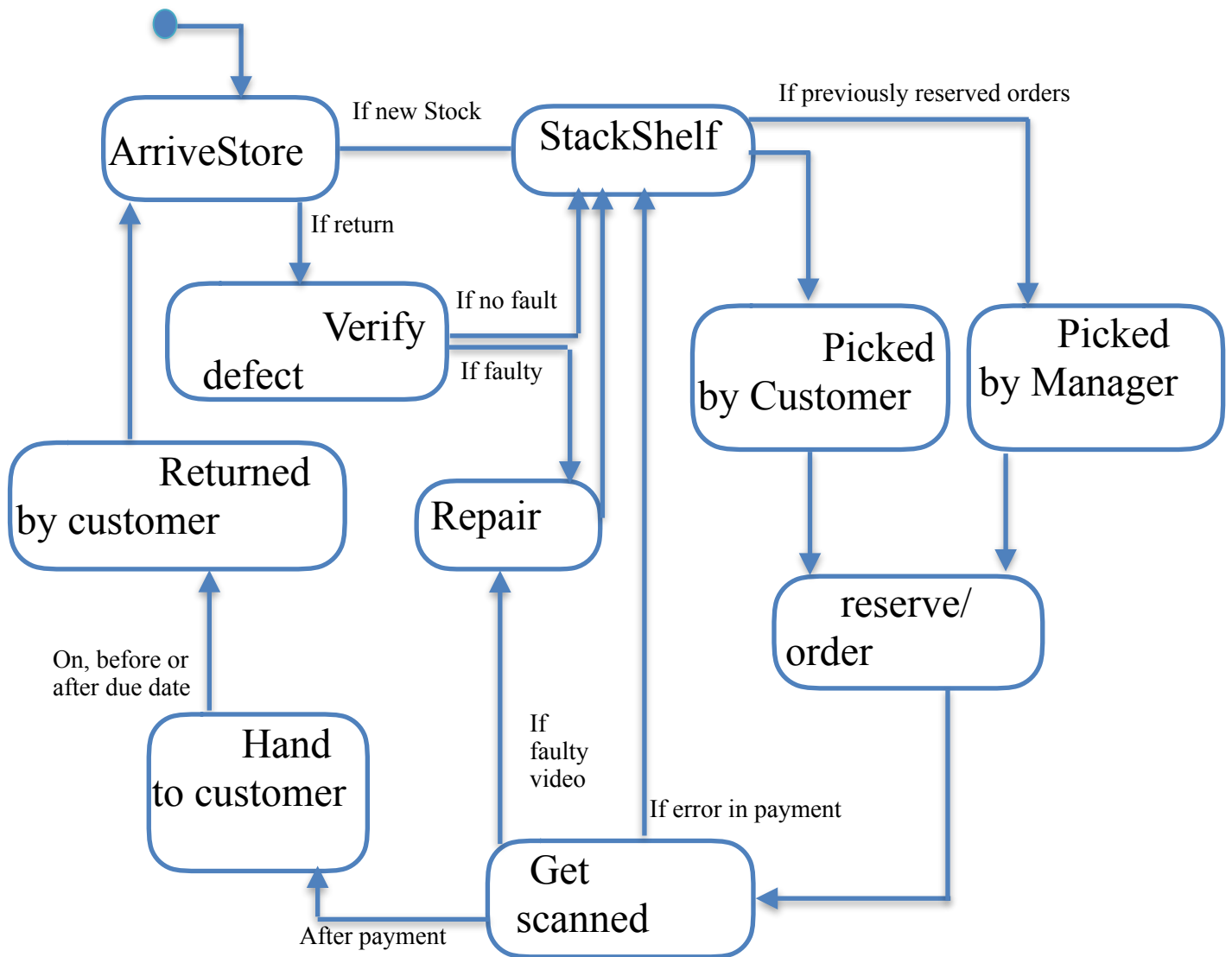
Represented as - A solid diamond at the whole's end (Rental Item).

3. Sequence diagram for renting a movie from a video store:

Customer	:User Interface Class	payment class	:Rental Management class	DVD Class
1.Select a dvd				
	2.Calculate payment			
	3.Provide Total order			
	4.Ask for Payment			
5.Enter payment details	5.Enter payment Details			
		6.Validate Payment	7.Update dvd purchase	
9.Make DVD available	8.Make DVD available	8.Make DVD available	8.Make DVD available	

4. State diagram that describes the lifetime of a video tape in a video store:

ArriveStore

If new Stock

StackShelf

If previously reserved orders

If return

Verify defect

If no fault

If faulty

Returned by customer

Repair

Picked by Customer

Picked by Manager

On, before or after due date

Hand to customer

If faulty video

reserve/ order

If error in payment

After payment

Get scanned

5. Activity diagram that describes renting a movie from a video store