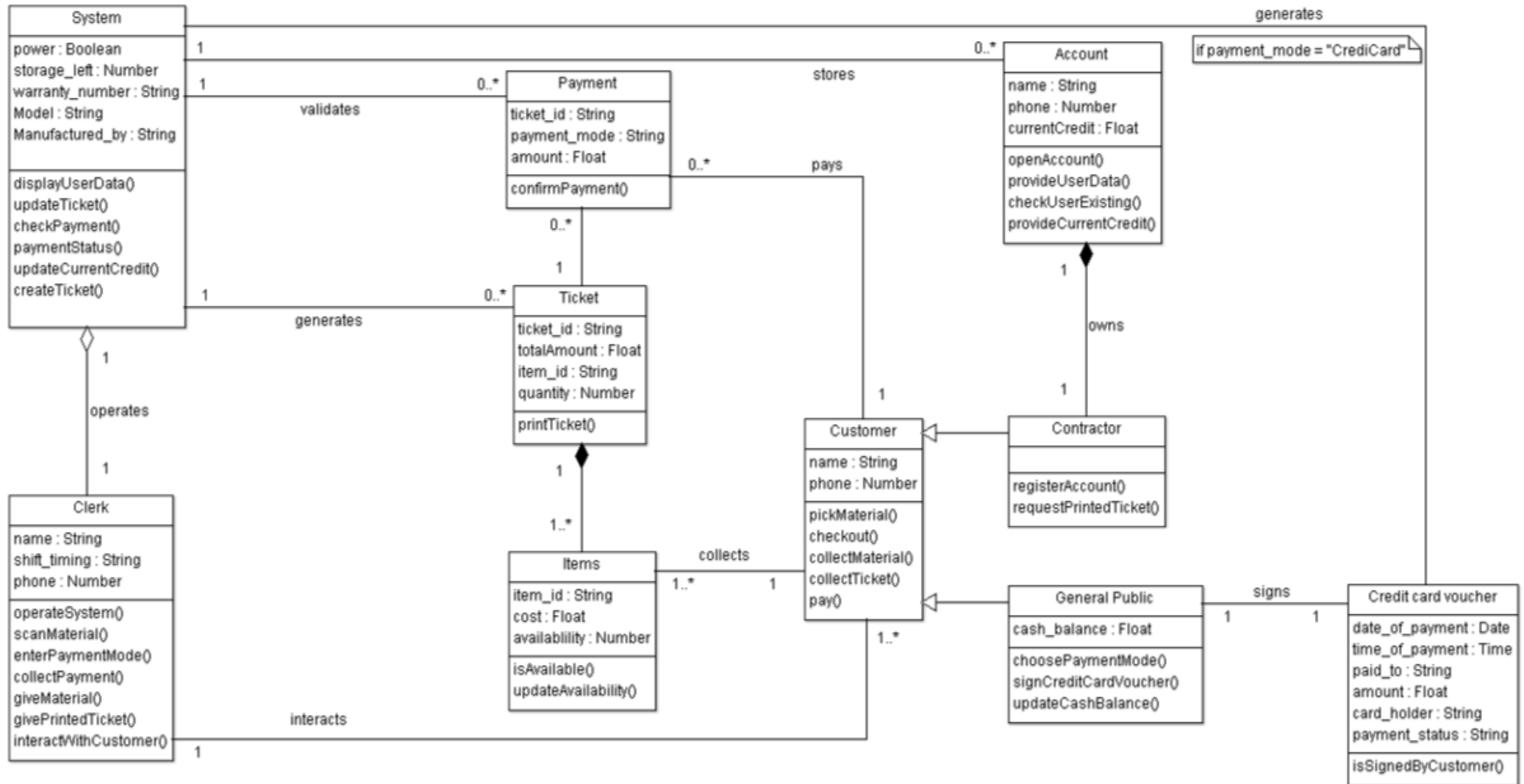**SPOT EXERCISE**

i. Refine the class diagram establishing relationships like association, aggregation, composition, class hierarchies (sub class, super class) for the given scenario.

Quality Building Supply has two kinds of customers: contractors and the general public. Sales to each are slightly different. A contractor buys materials by taking them to the checkout desk for contractors. The clerk enters the contractor's name into the system. The system displays the contractor's information, including current credit standing. The clerk then opens up a new ticket (sale) for the contractor. Next, the clerk scans in each item to be purchased. The system finds the price of the item and adds the item to the ticket. At the end of the purchase, the clerk indicates the end of the sale. The system compares the total amount against the contractor's current credit limit and, if it is acceptable, finalizes the sale. The system creates an electronic ticket for the items, and the contractor's credit limit is reduced by the amount of the sale. Some contractors like to keep a record of their purchases, so they request that ticket details be printed. Others aren't interested in a printout. A sale to the general public is simply entered into the cash register, and a paper ticket is printed as the items are identified. Payment can be by cash, check, or credit card. The clerk must enter the type of payment to ensure that the cash register balances at the end of the shift. For credit card payments, the system prints a credit card voucher that the customer must sign.

Class diagram with conceptual class hierarchies, association, aggregation & composition relationships:

System

power : Boolean
storage_left : Number
warranty_number : String
Model : String
Manufactured_by : String

displayUserData()
updateTicket()
checkPayment()
paymentStatus()
updateCurrentCredit()
createTicket()

Payment

ticket_id : String
payment_mode : String
amount : Float

confirmPayment()

Account

name : String
phone : Number
currentCredit : Float

openAccount()
provideUserData()
checkUserExisting()
provideCurrentCredit()

if payment_mode = "CrediCard"

generates

stores

validates

pays

Ticket

ticket_id : String
totalAmount : Float
item_id : String
quantity : Number

printTicket()

generates

owns

Clerk

name : String
shift_timing : String
phone : Number

operateSystem()
scanMaterial()
enterPaymentMode()
collectPayment()
giveMaterial()
givePrintedTicket()
interactWithCustomer()

operates

Items

item_id : String
cost : Float
availablility : Number

isAvailable()
updateAvailability()

collects

interacts

Customer

name : String
phone : Number

pickMaterial()
checkout()
collectMaterial()
collectTicket()
pay()

Contractor

registerAccount()
requestPrintedTicket()

General Public

cash_balance : Float

choosePaymentMode()
signCreditCardVoucher()
updateCashBalance()

signs

Credit card voucher

date_of_payment : Date
time_of_payment : Time
paid_to : String
amount : Float
card_holder : String
payment_status : String

isSignedByCustomer()

Conceptual class hierarchies:

Super Class	Sub Class
Customer	Contractor
	General Public

Association relationship:

Class-1	Role	Class-2
System	stores	Account
System	validates	Payment
System	generates	Ticket
System	generates	Credit card voucher
Clerk	interacts	Customer
Customer	pays	Payment
Customer	collects	Item
General Public	signs	Credit card voucher

Aggregation relationship:

Class of group elements	Class of groups
Clerk	System

Composition relationship:

Class of components	Class of composites
Item	Ticket
Contractor	Account