Software Modeling Dev with UML
CSP- 586
Assignment #5

Chen Xu A20377739

1. All artifacts

Assignment 1:

1. List of actors

1. Store manager
2. Customer
3. Salesman

2. List of use cases

1. The store manager can Add products
2. The store manager can Delete products
3. The store manager can Update products
4. The store manager can purchase the new and pre-owned different game consoles and accessories
5. The store manager can purchase the new and pre-owned games
6. The store manager can purchase the new and pre-owned tablets
7. Store manager can offer special deals:
    1. store special-discounts
    2. manufacturer rebates
8. The customer can pre-order products
9. The customer can trade-in products

10. The customer can place order online

11. The customer can check online order status

12. The customer can cancel Online order

13. The customer can pay in cash, check, or credit card

14. The customer can enroll (or cancel) Power Member: In order to receive 5% discount for every item purchased for an annual fee of $100

15. The customer can rent console with following lease plans:

    1. Daily rental (for example renting the console for 2 days)
    2. Monthly rental (for example rent the XBOX ONE console for 2 months with rental $20/month)
    3. Yearly rental (for example rent the XBOX ONE console for $100/year)

16. The customer can choose one of the following options when buying a new console:

    1. Buy the new console with no replacement
    2. Buy the new console with 1-year replacement for

50% fee of the console retail price

    3. Buy the new console with lifetime replacement for

65% fee of the console retail price

17. Customer can order Console models:

    1. Microsoft

a. XBOX One

b. XBOX 360

2. Sony

a. PS3

b. PS4

3. Nintendo

a. Wii

b. WiiU

18. Customer can order console accessories

19. Customer can order games:

1. Electronic Arts

2. Activision

3. Take-Two Interactive

20. Salesman can create Customer accounts

21. Salesman can Create customers order

22. Salesman can Cancel customers order

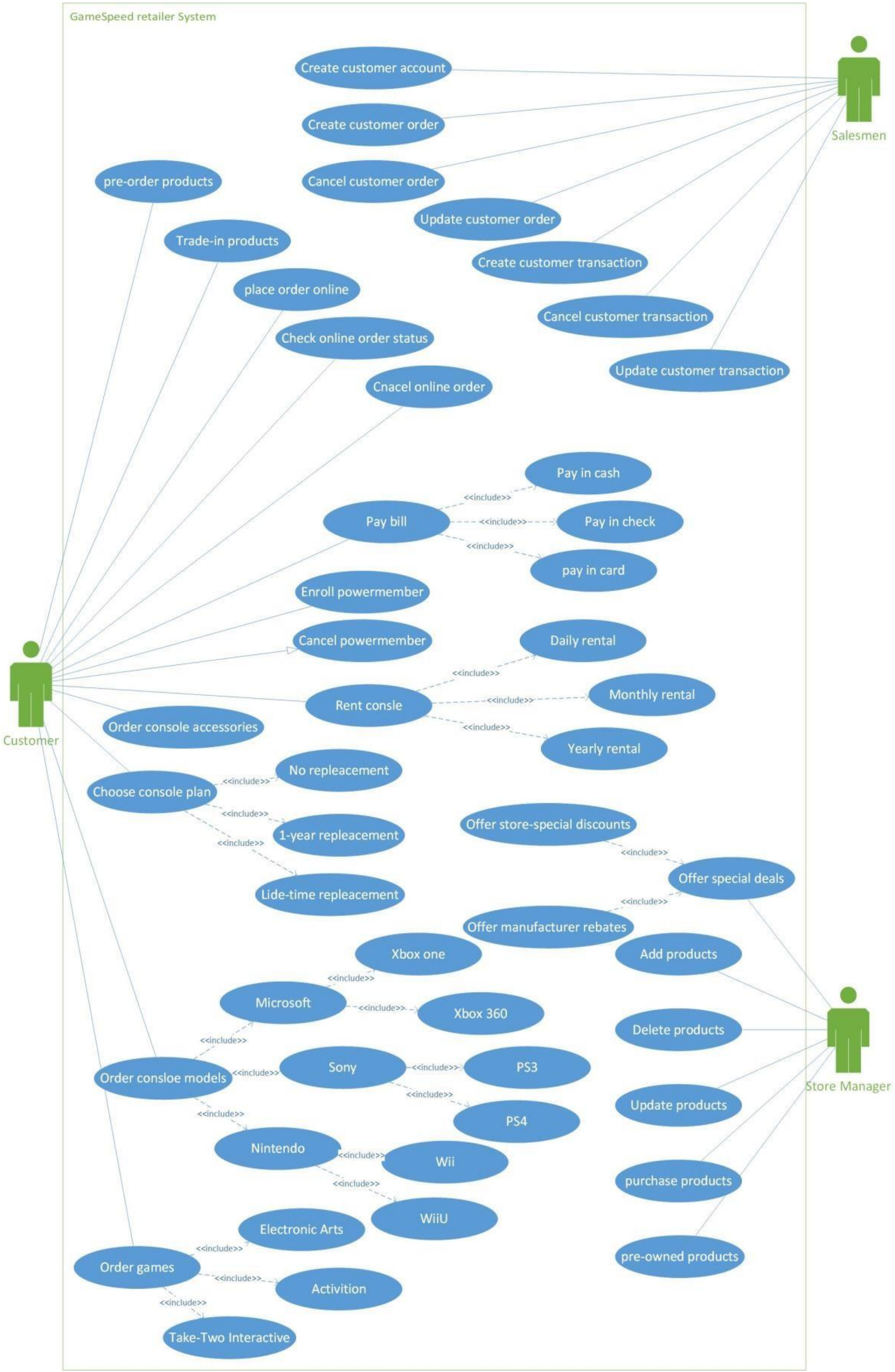
23. Salesman can Update customers order

24. Salesman can Create customers transaction

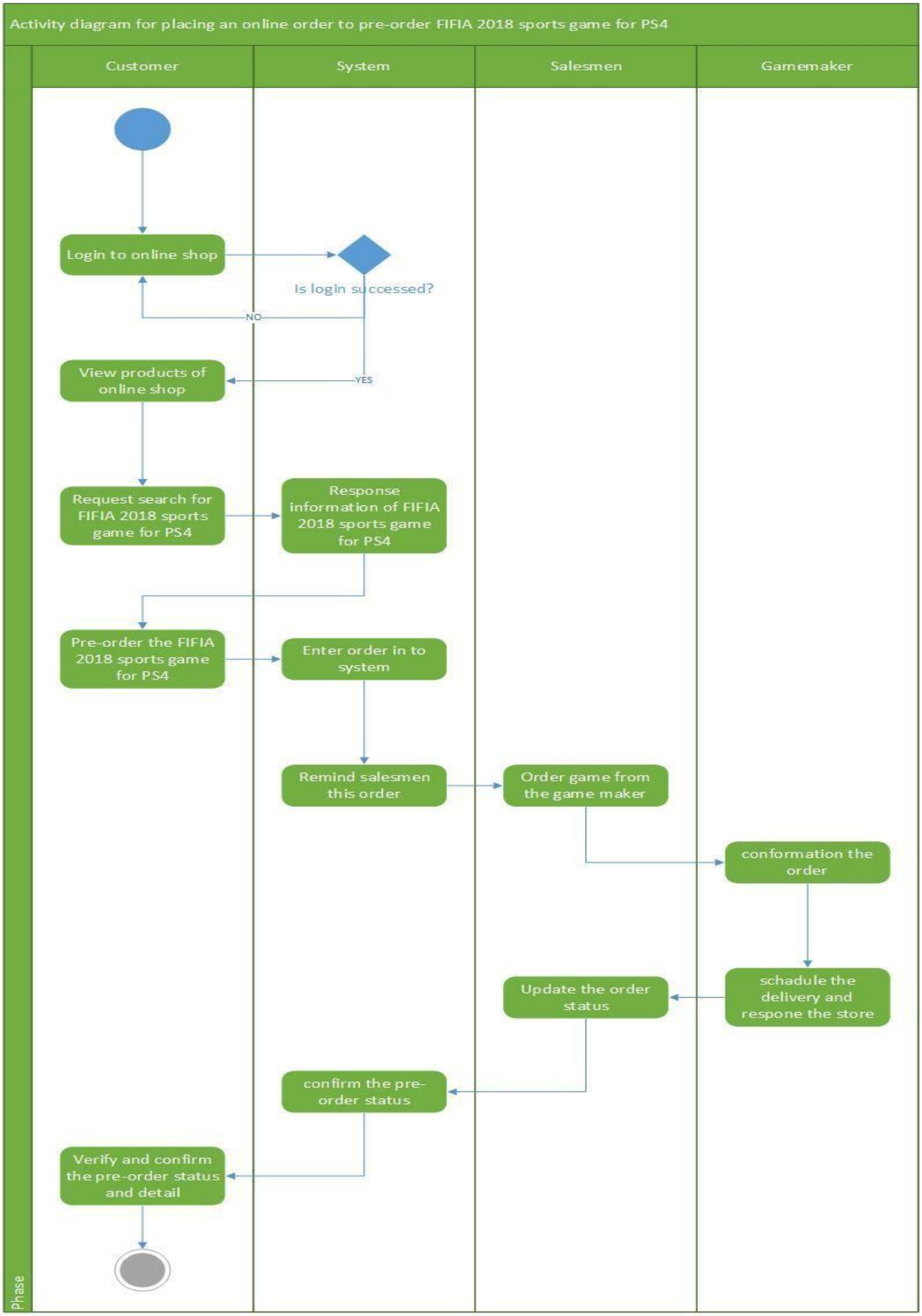
25. Salesman can Cancel customers transaction

26. Salesman can Update customers transaction

3. UML Use-Case Diagram



4. Activity diagram


Activity diagram for placing an online order to pre-order FIFIA 2018 sports game for PS4
Customer
System
Salesmen
Gamemaker
Phase
Login to online shop
Is login successed?
NO
YES
View products of online shop
Request search for FIFIA 2018 sports game for PS4
Response information of FIFIA 2018 sports game for PS4
Pre-order the FIFIA 2018 sports game for PS4
Enter order in to system
Remind salesmen this order
Order game from the game maker
conformation the order
schadule the delivery and respone the store
Update the order status
confirm the pre-order status
Verify and confirm the pre-order status and detail

5. Fully dressed format Use-case

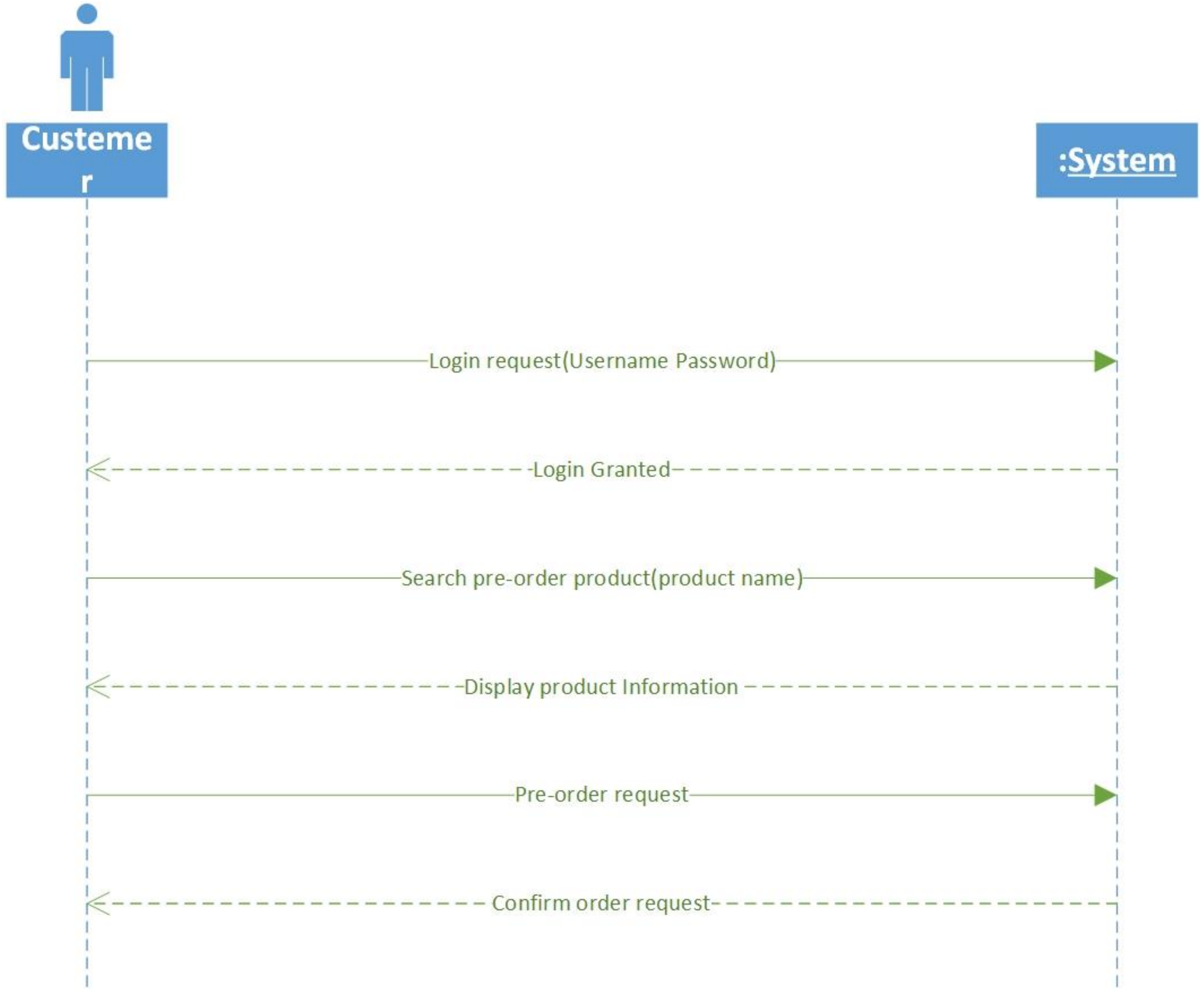
Use case: Place an online order

Use case name	Place an online order	
Scenario	Customer place an order online	
Triggering Event	Customer want to purchase product online	
Brief Description	Customer view the products online and choose products to order. Customer send a request to order products. Then verify and confirm the online order.	
Actors	Customer	
Stakeholders	Salesmen, Store manager	
Preconditions	1. Customer must exist 2. Customer profile must be valid	
Postconditions	1. The product must exist 2. The order must be associated with customer	
Flow of Events	Actor	System
	1. Customer login 2. Customer view products 3. Customer request to order	1.1 System confirm the login 2.1 System response the information of products 3.1 System confirms the order
Exception Conditions	1. Customer unable to login 2. Product not found 3. Website not found	

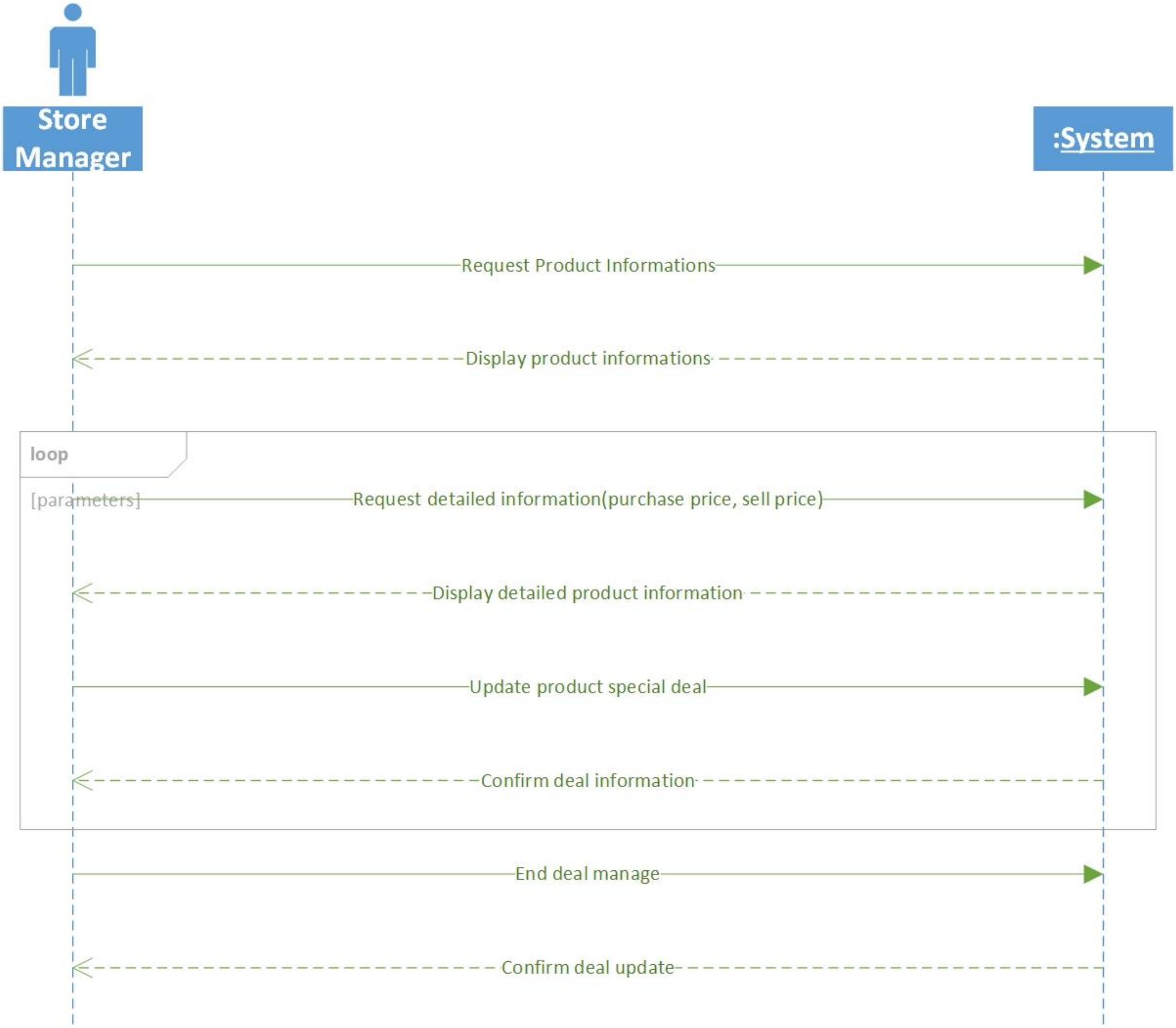
Assignment 2:

1. System Sequence Diagrams

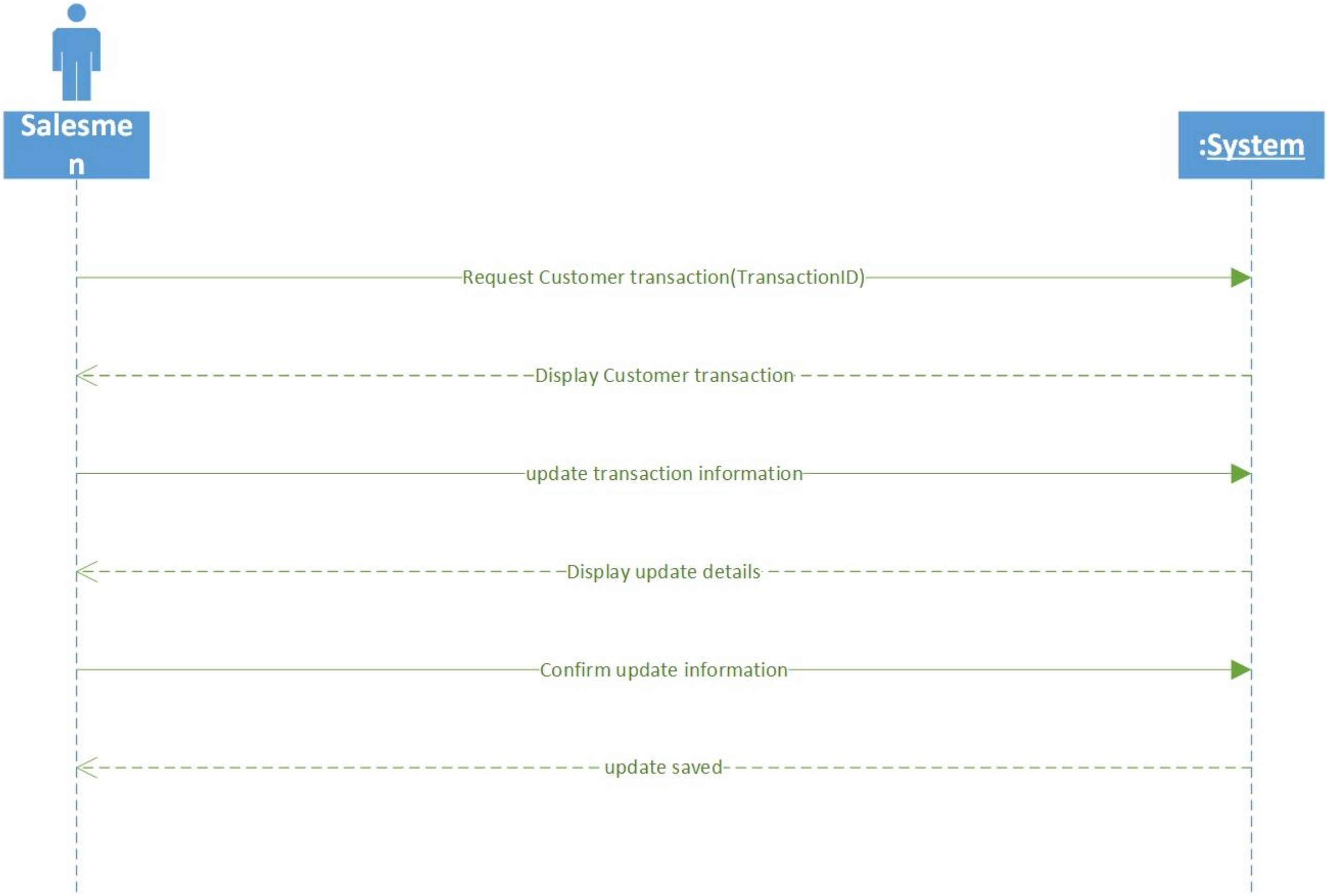
1. Customer Pre-order product online


Custeme r
:System
Login request(Username Password)
Login Granted
Search pre-order product(product name)
Display product Information
Pre-order request
Confirm order request

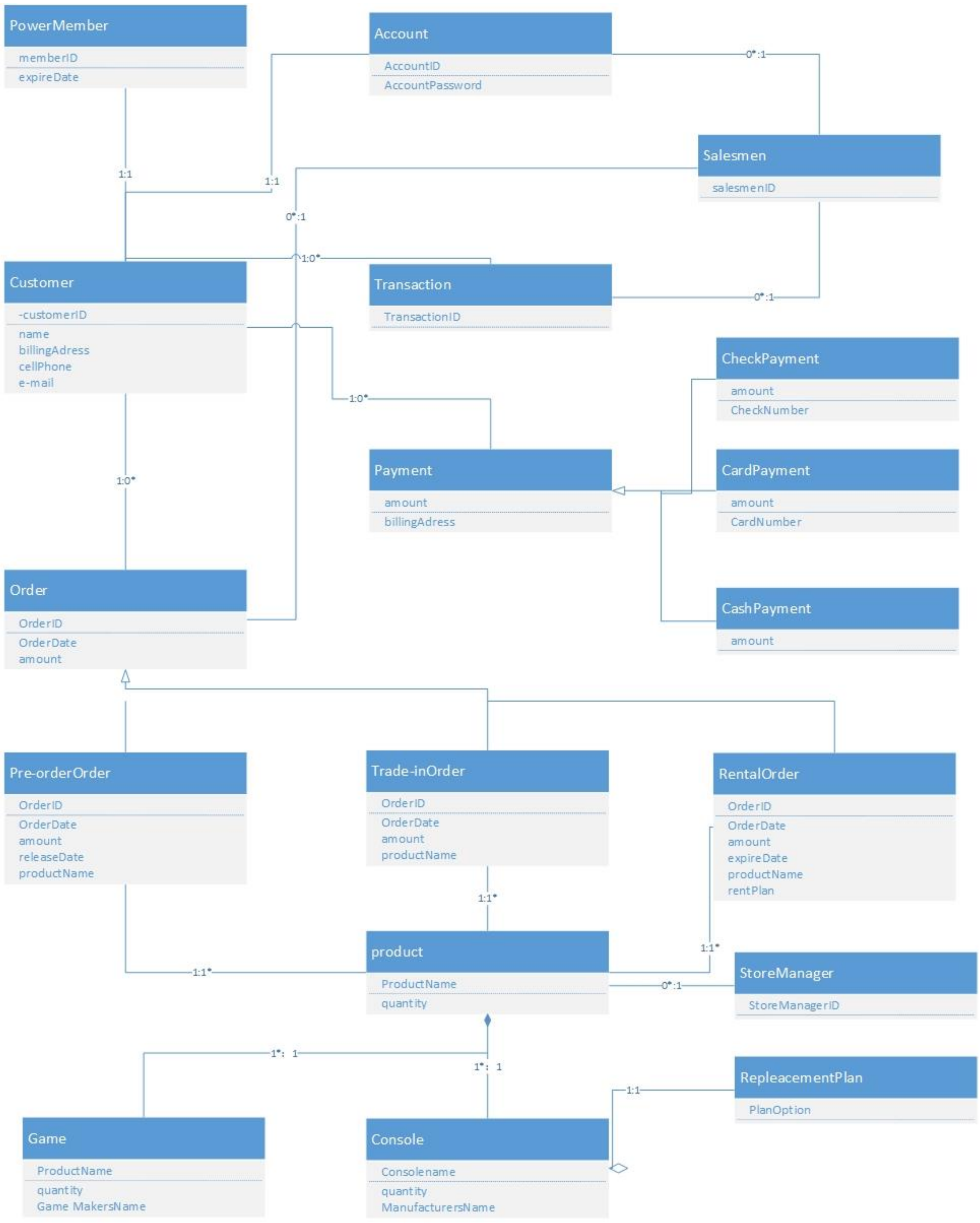
2. Store manager offer special deals


Store Manager
:System
Request Product Informations
Display product informations
loop
[parameters]
Request detailed information(purchase price, sell price)
Display detailed product information
Update product special deal
Confirm deal information
End deal manage
Confirm deal update

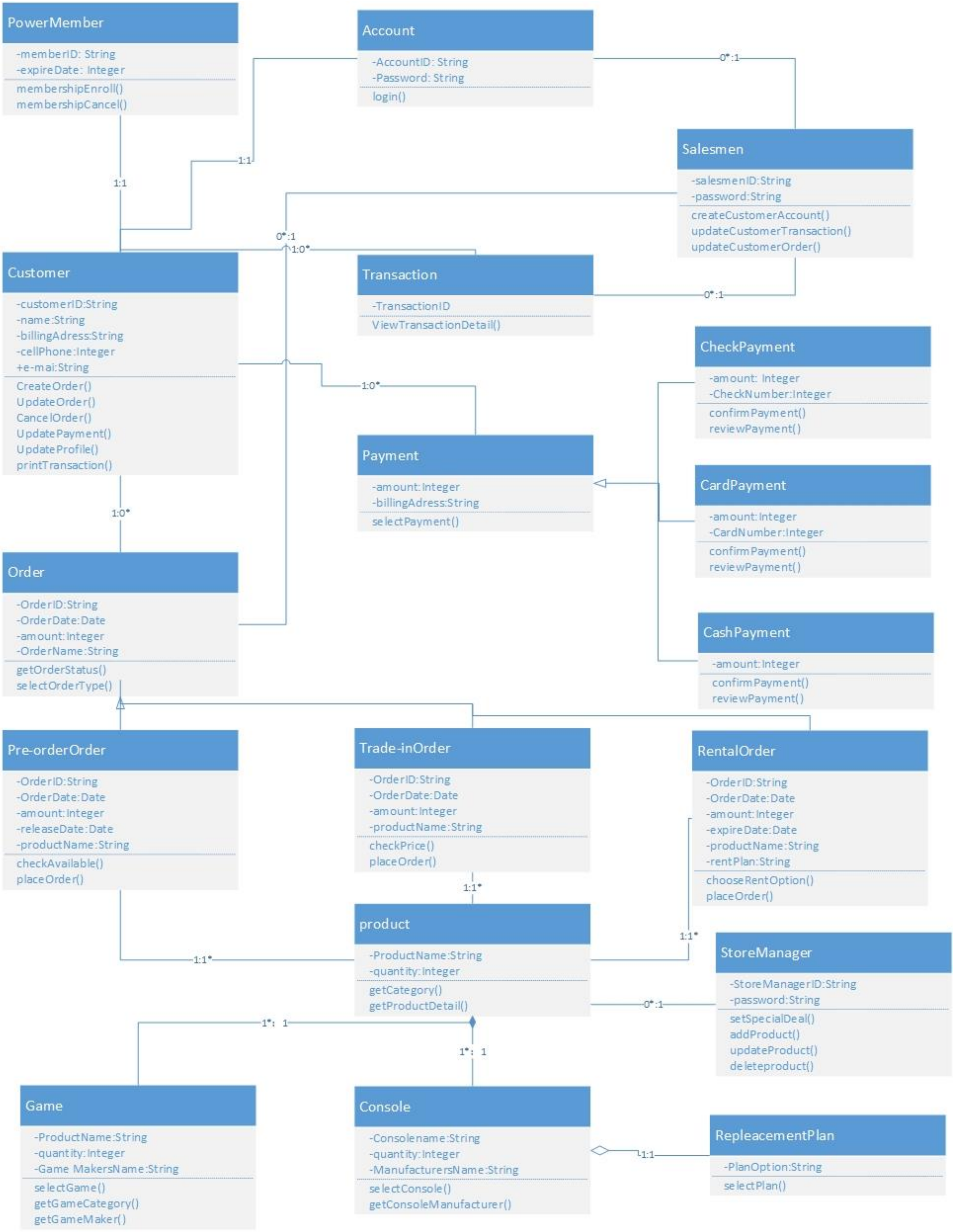
3. Salesmen update customer transaction

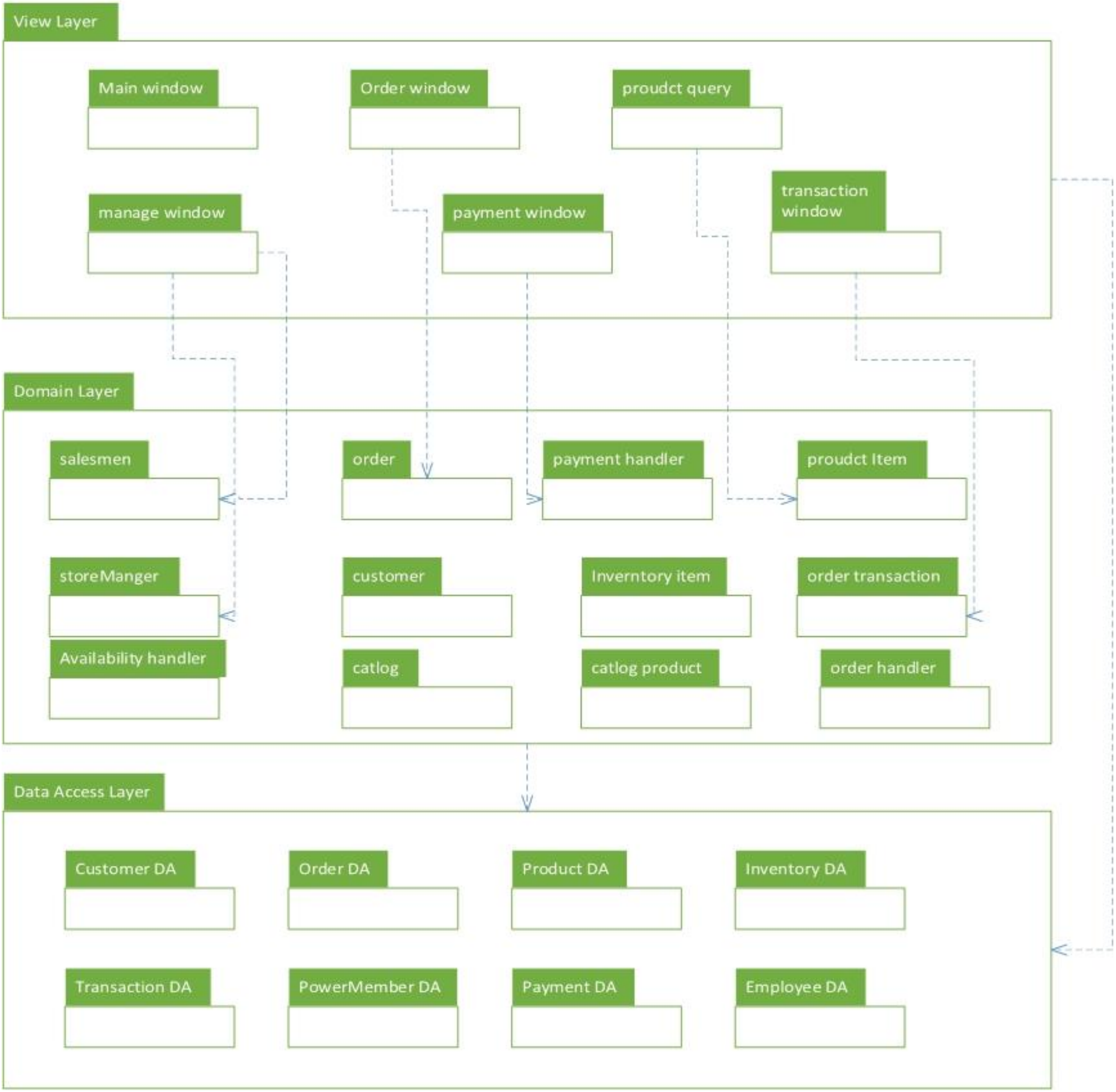

Salesmen
:System
Request Customer transaction(TransactionID)
Display Customer transaction
update transaction information
Display update details
Confirm update information
update saved

2. The Class diagram for analysis model



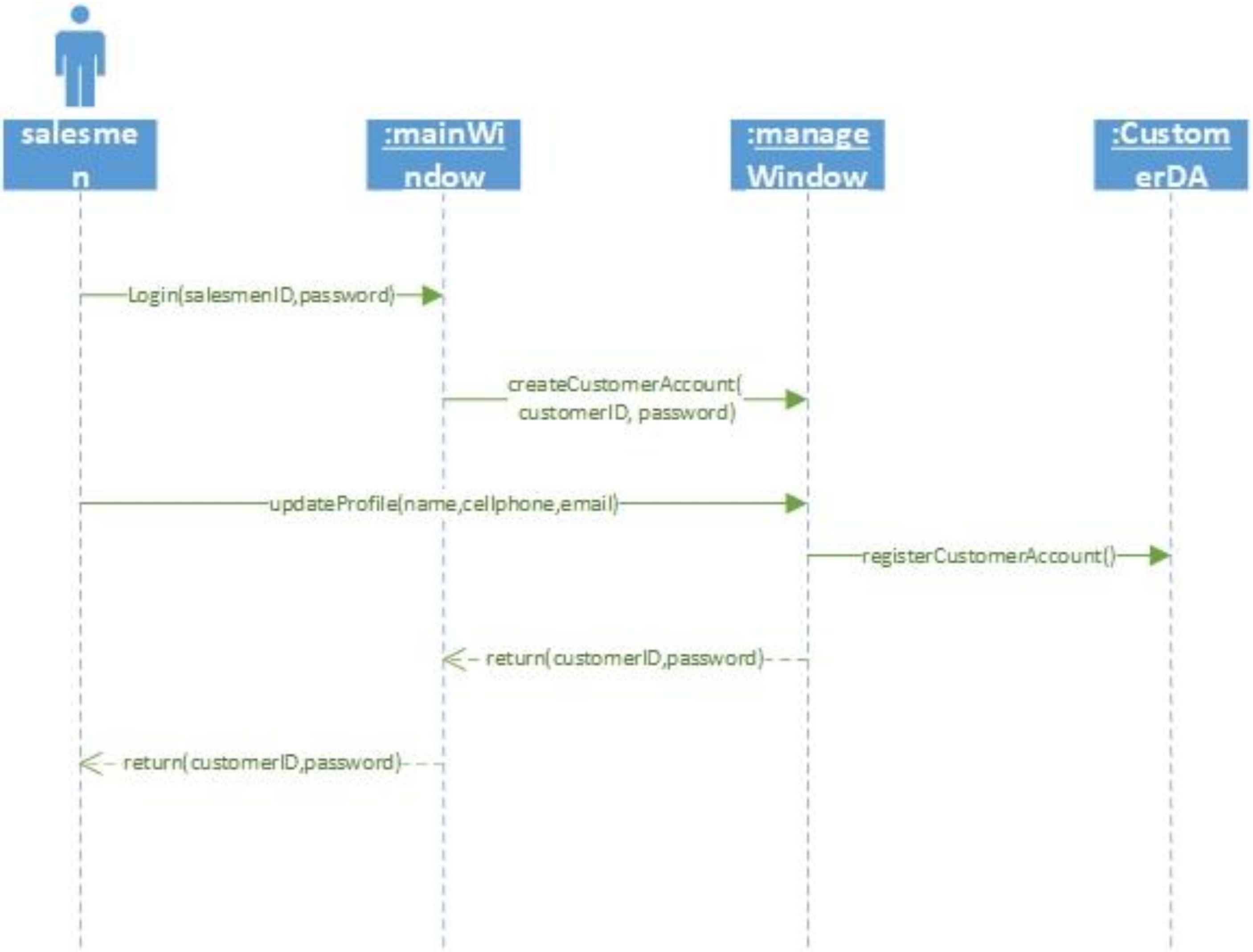
3. The Class diagram and package diagram for design model



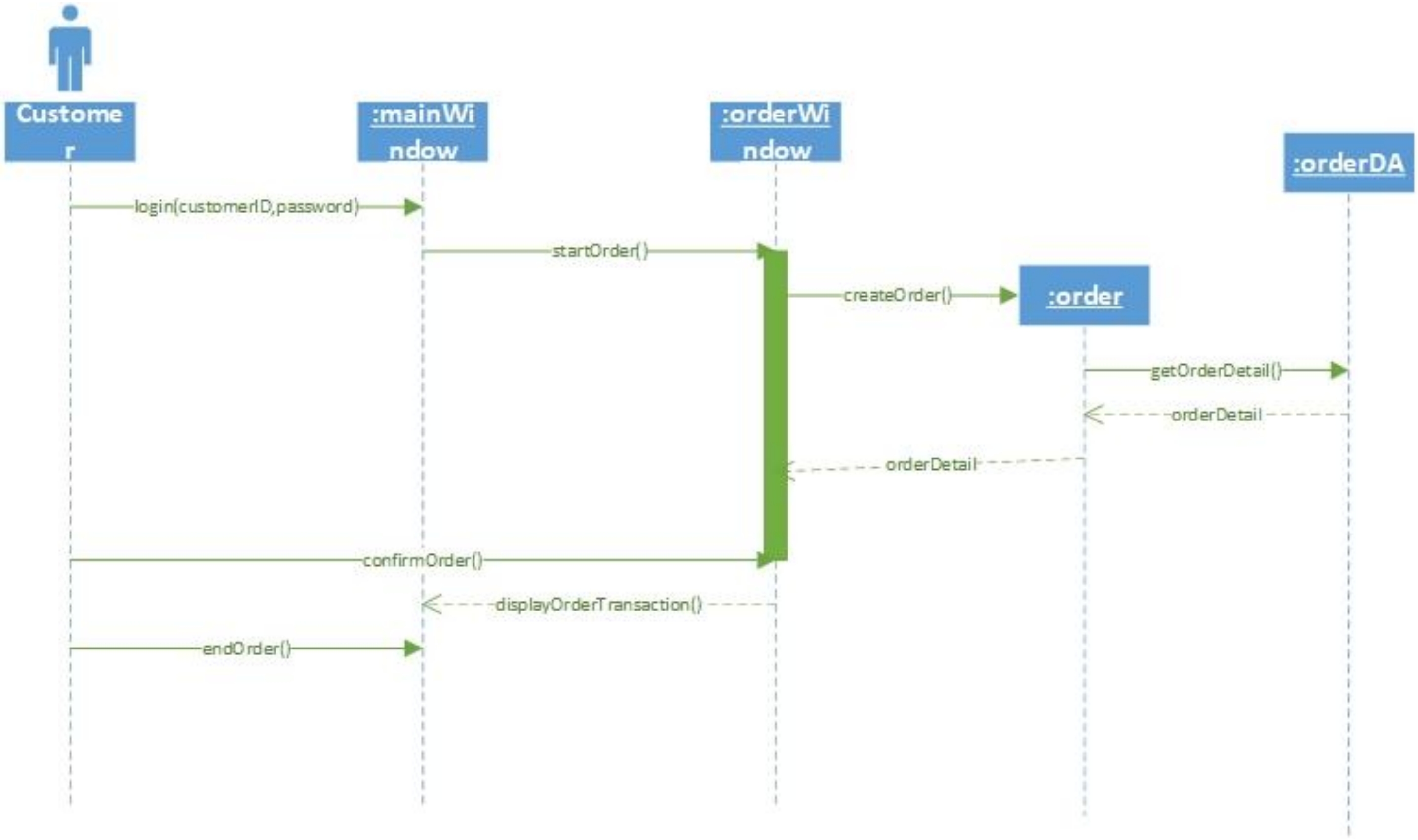

View Layer
Main window
Order window
proudct query
manage window
payment window
transaction window
Domain Layer
salesmen
order
payment handler
proudct Item
storeManger
customer
Inverntory item
order transaction
Availability handler
catlog
catlog product
order handler
Data Access Layer
Customer DA
Order DA
Product DA
Inventory DA
Transaction DA
PowerMember DA
Payment DA
Employee DA

4. Sequence interaction diagrams

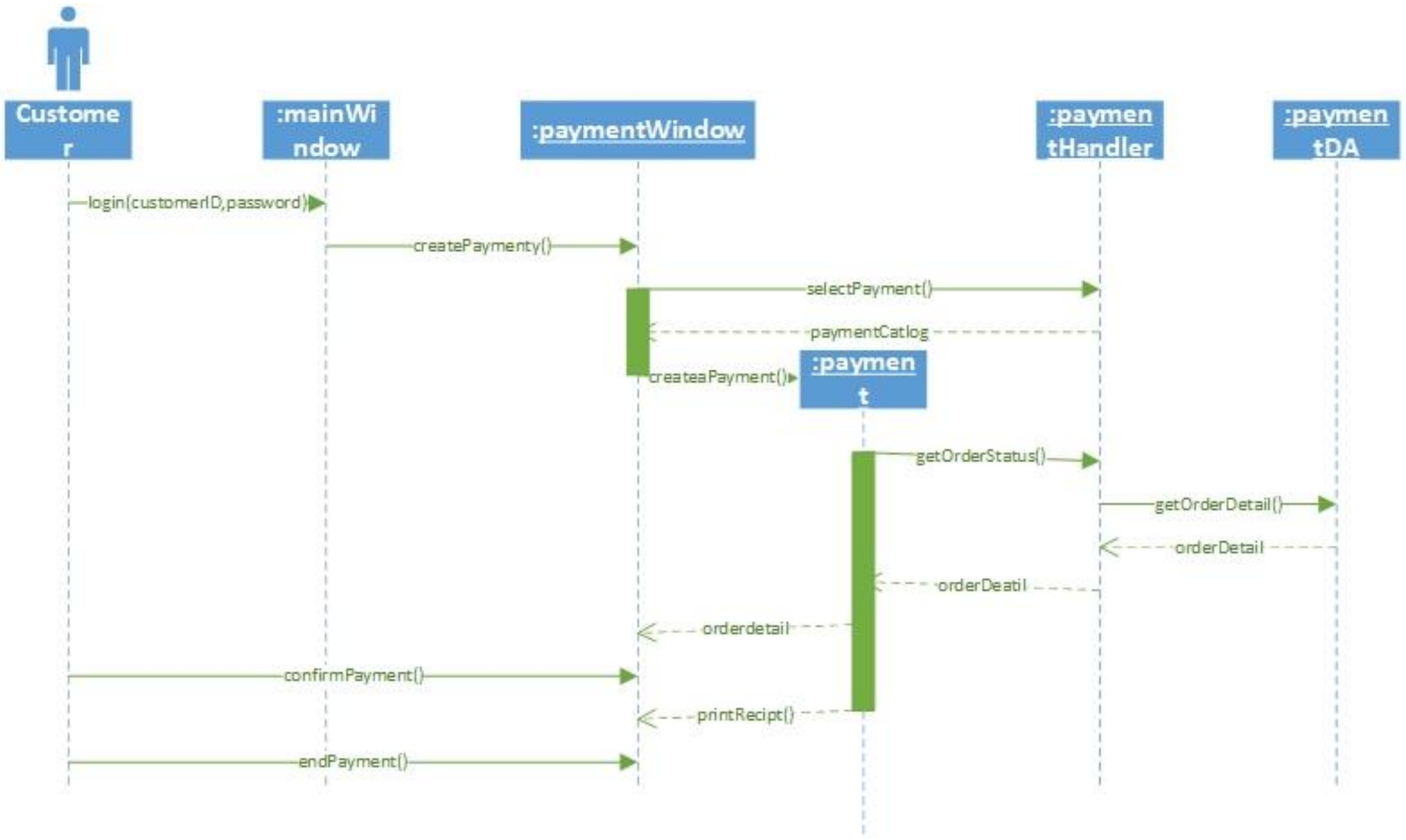
1. Salesmen create new customer account


salesmen
:mainWindow
:manageWindow
:CustomerDA
Login(salesmenID,password)
createCustomerAccount(customerID, password)
updateProfile(name,cellphone,email)
registerCustomerAccount()
return(customerID,password)
return(customerID,password)

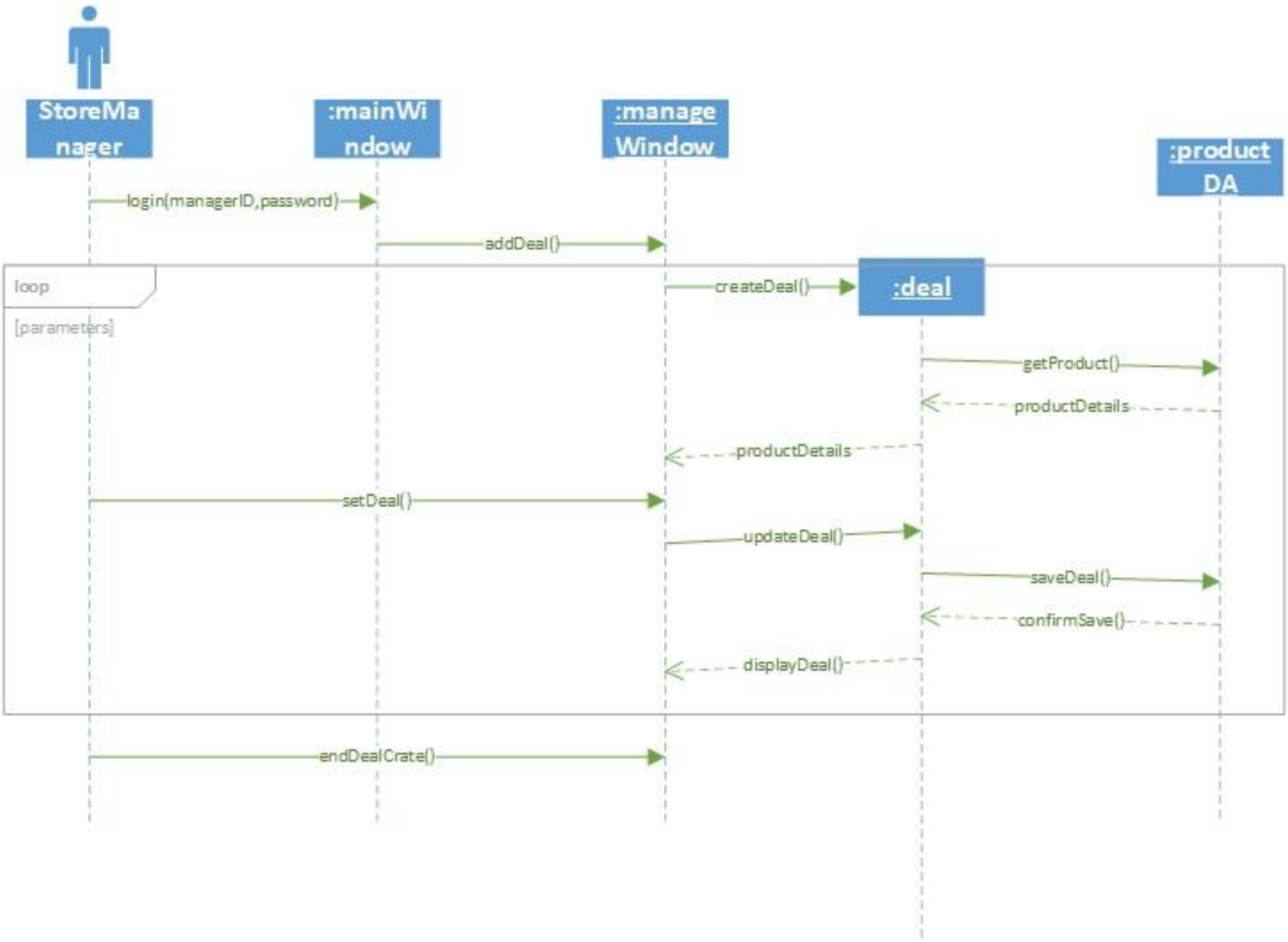
2. Customer pre-order product



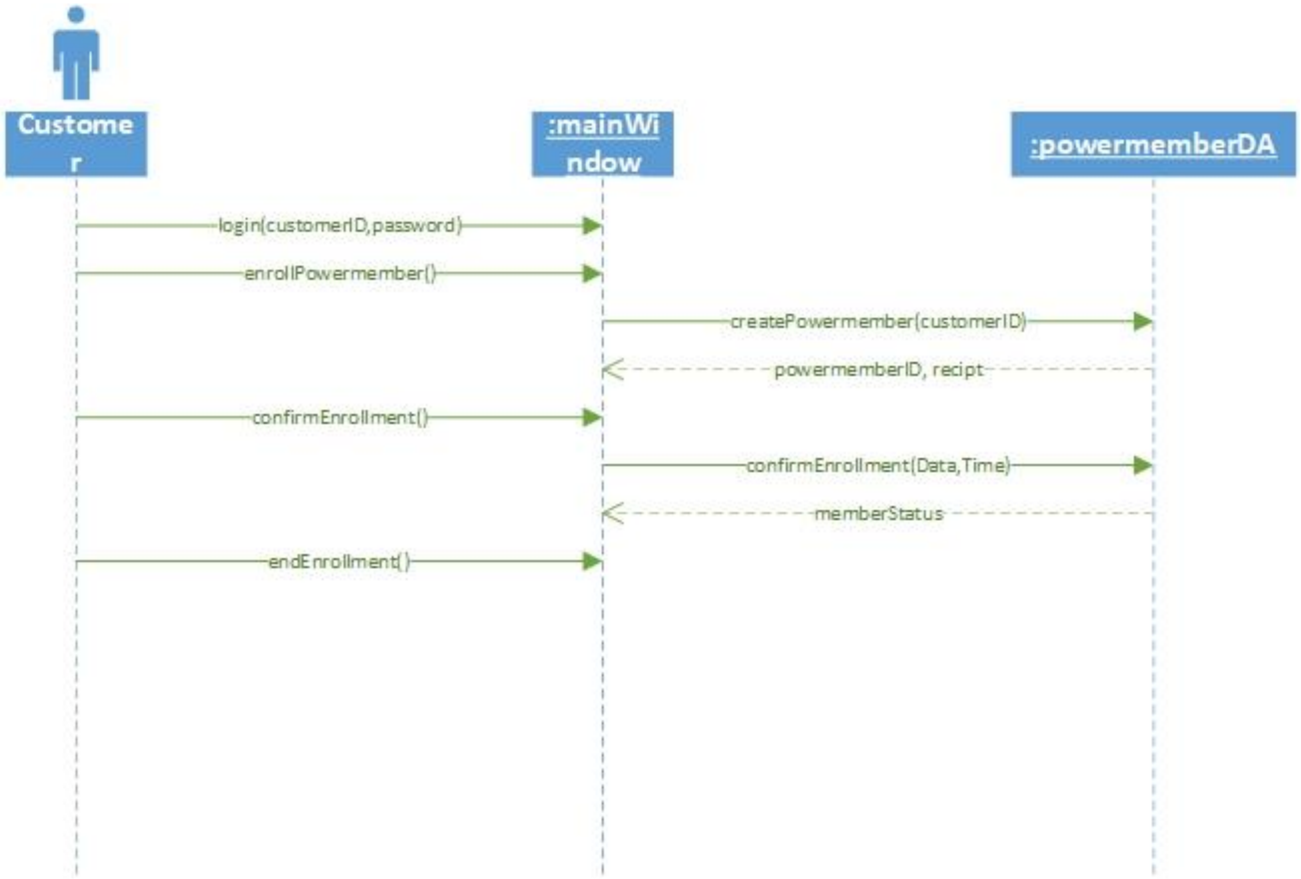
3. Make payment



4. Add special deal



5. Enroll power member

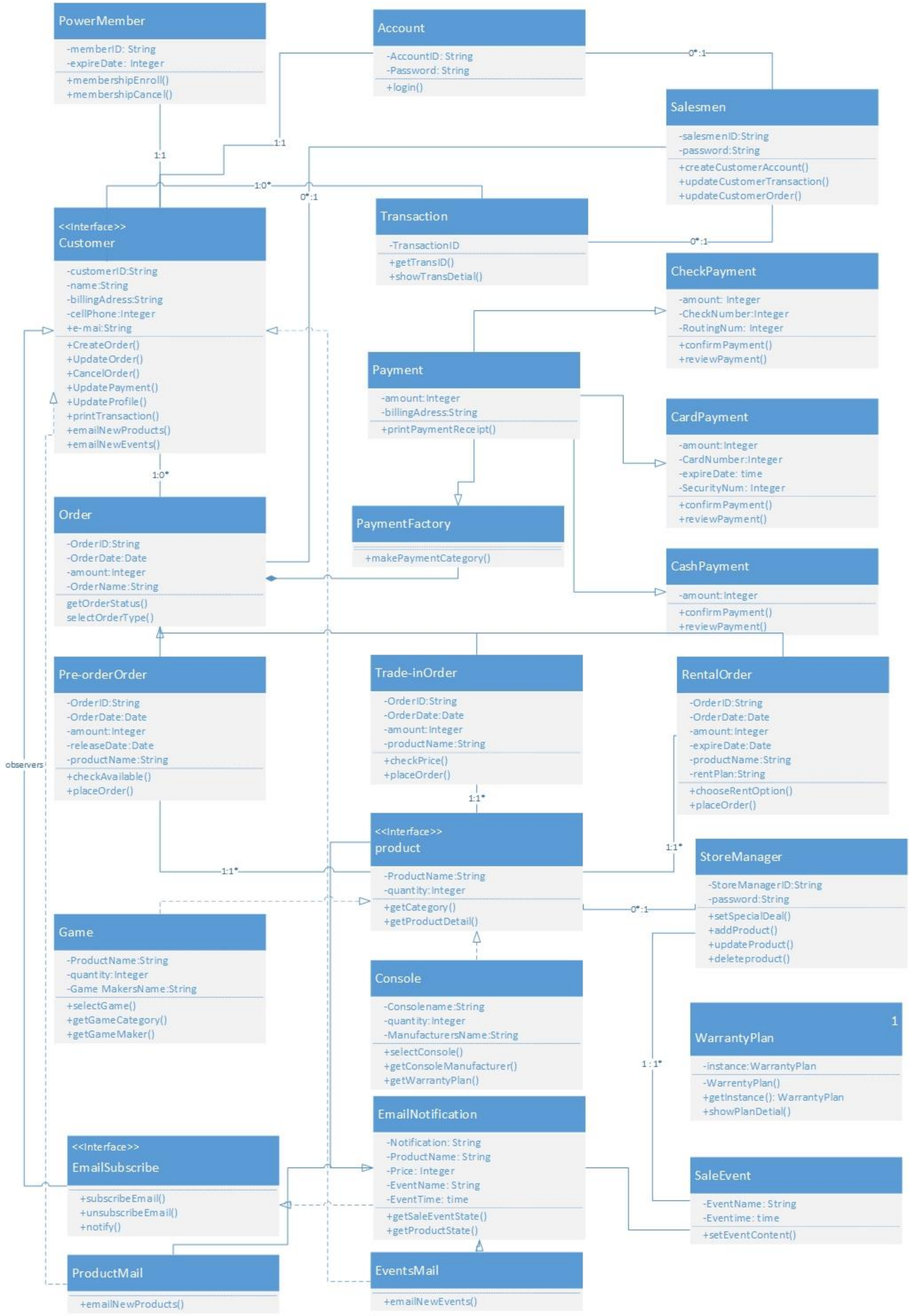


Assignment 3:

1. Complete list of classes

- Customer
- PowerMember
- Account
- Salesman
- Transaction
- Payment
- CheckPayment
- CardPayment
- CashPayment
- Order
- Pre-orderOrder
- Trade-inOrder
- RentalOrder
- StoreManager
- Product
- Game
- Console
- WarrantyPlan
- EmailNotification
- SaleEvent
- EmailSubscribe
- ProductMail
- EventMail

2. Complete UML Design Model/class diag

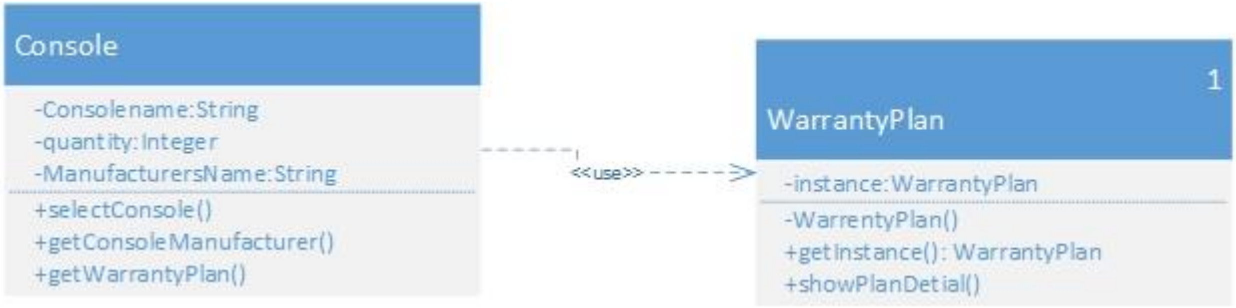


3. List of the Design pattern(s)

1) Singleton Design Pattern
2) Factory Method Design Pattern
3) Observer Design Pattern

4. Documentation of used design patterns

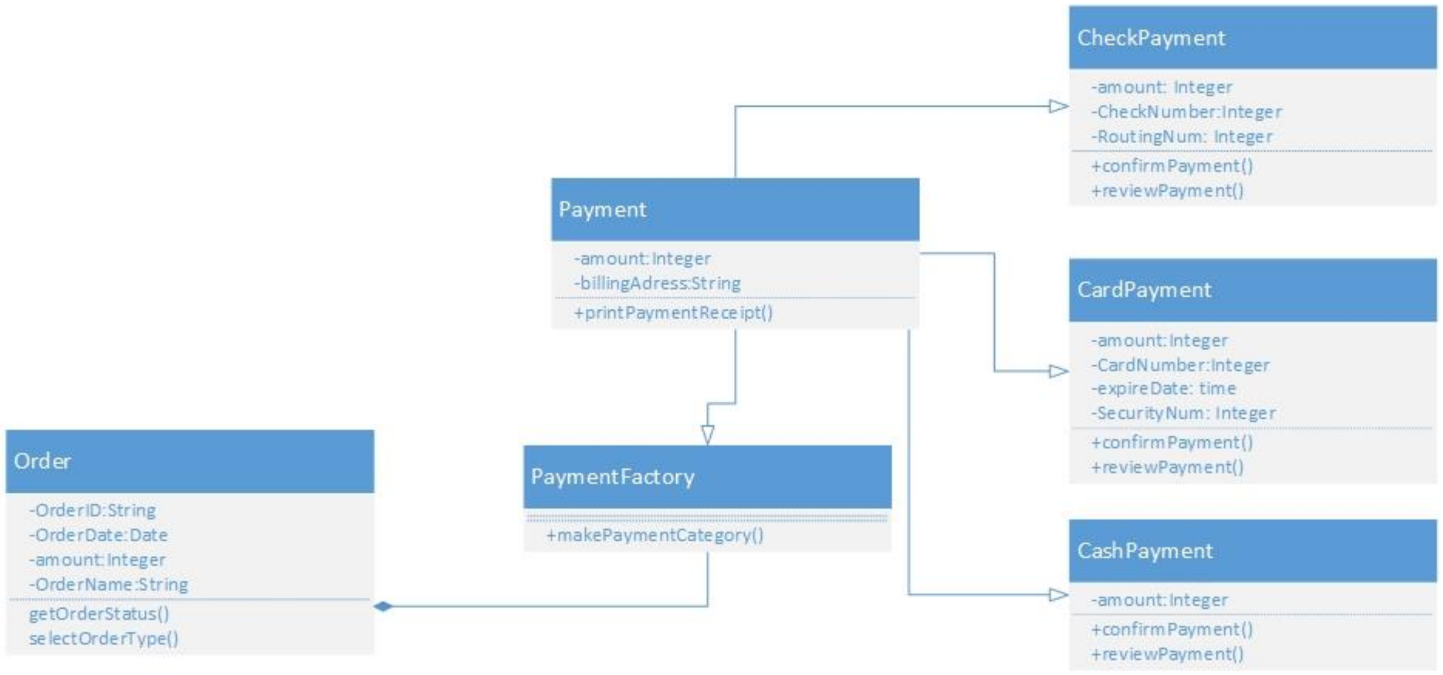
1) Singleton Design Pattern: Implement select console Warranty plan:



I. Use singleton Pattern to make every console use the instance that instantiated by WarrantyPlan Class.

II. WarrantyPlan has Instance in the Class and Console can get it by using getInstance() method.

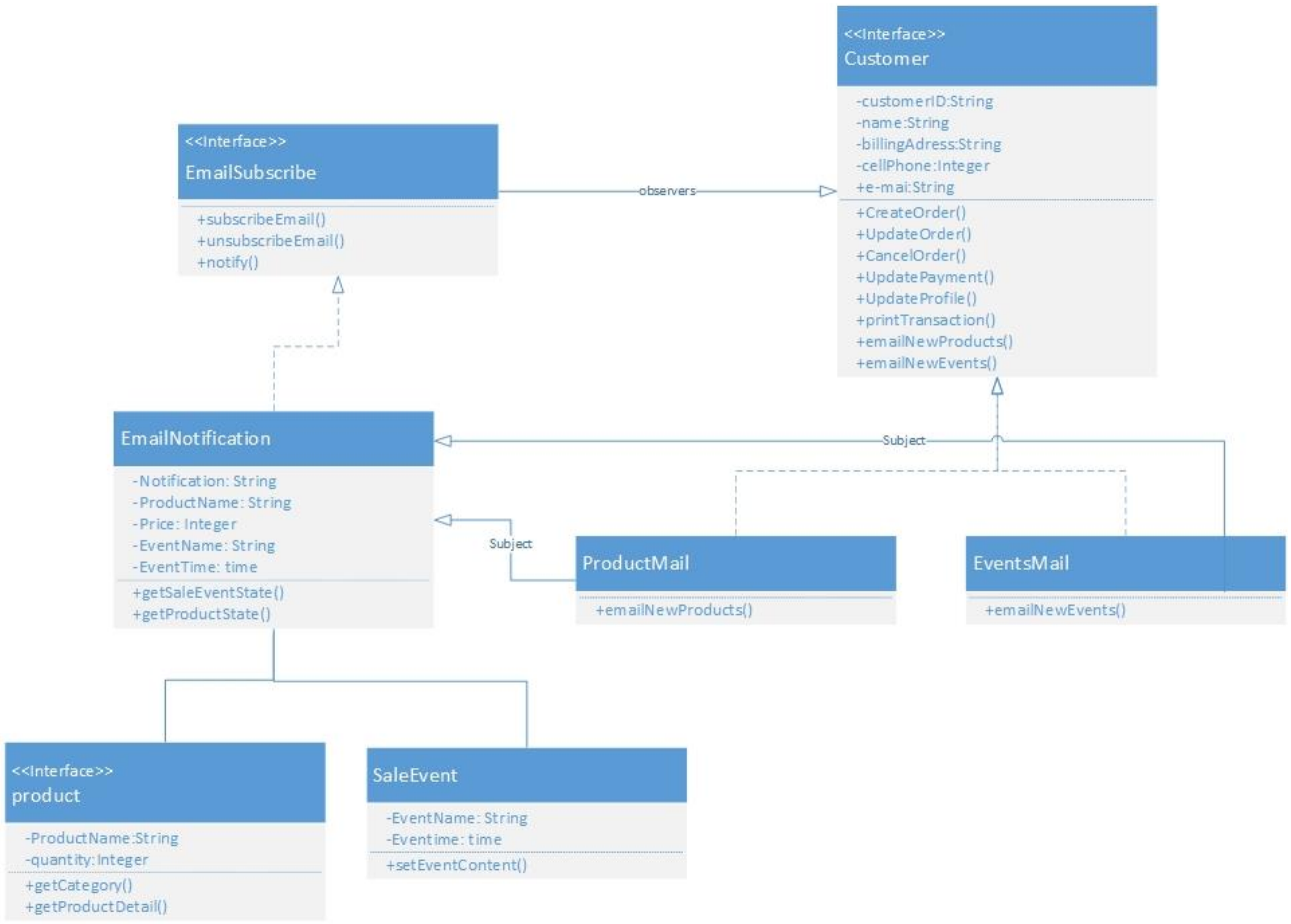
2) Factory Method Design Pattern: Implement Order make payment:



I. Order get payment instance through PaymentFactory Class and use the payment category of the Order to instance the payment in right way.

II. CheckPayment , CardPayment and CashPayment Class are the different category of payment that inherit from the Payment Class.

3) Observer Design Pattern: Implement email subscribtion:



I. the customer will subscribe or unsubscribe to get the notification of new products and events .

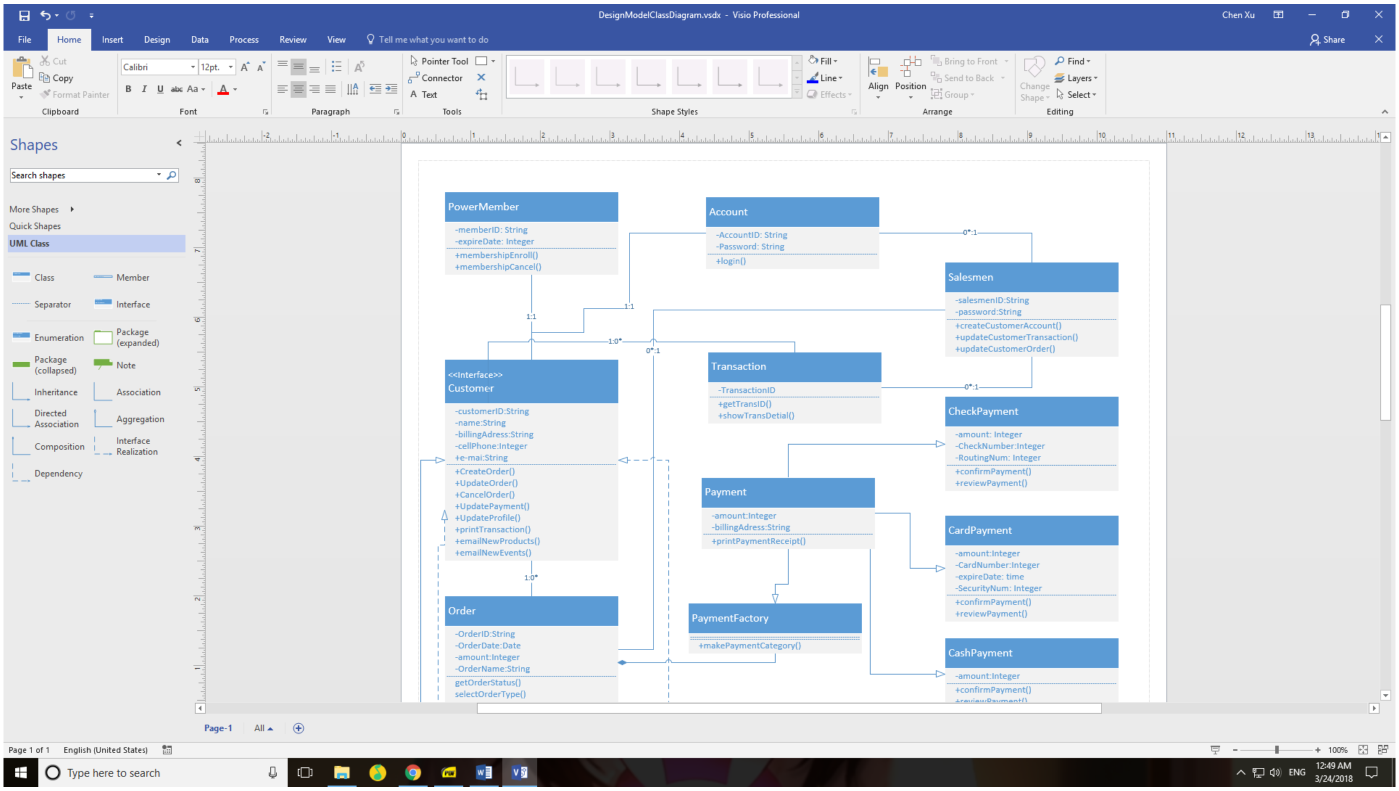
II. The Customer is the observer interface, it can use ProductMail and EventsMail Class to get new product and event notification.

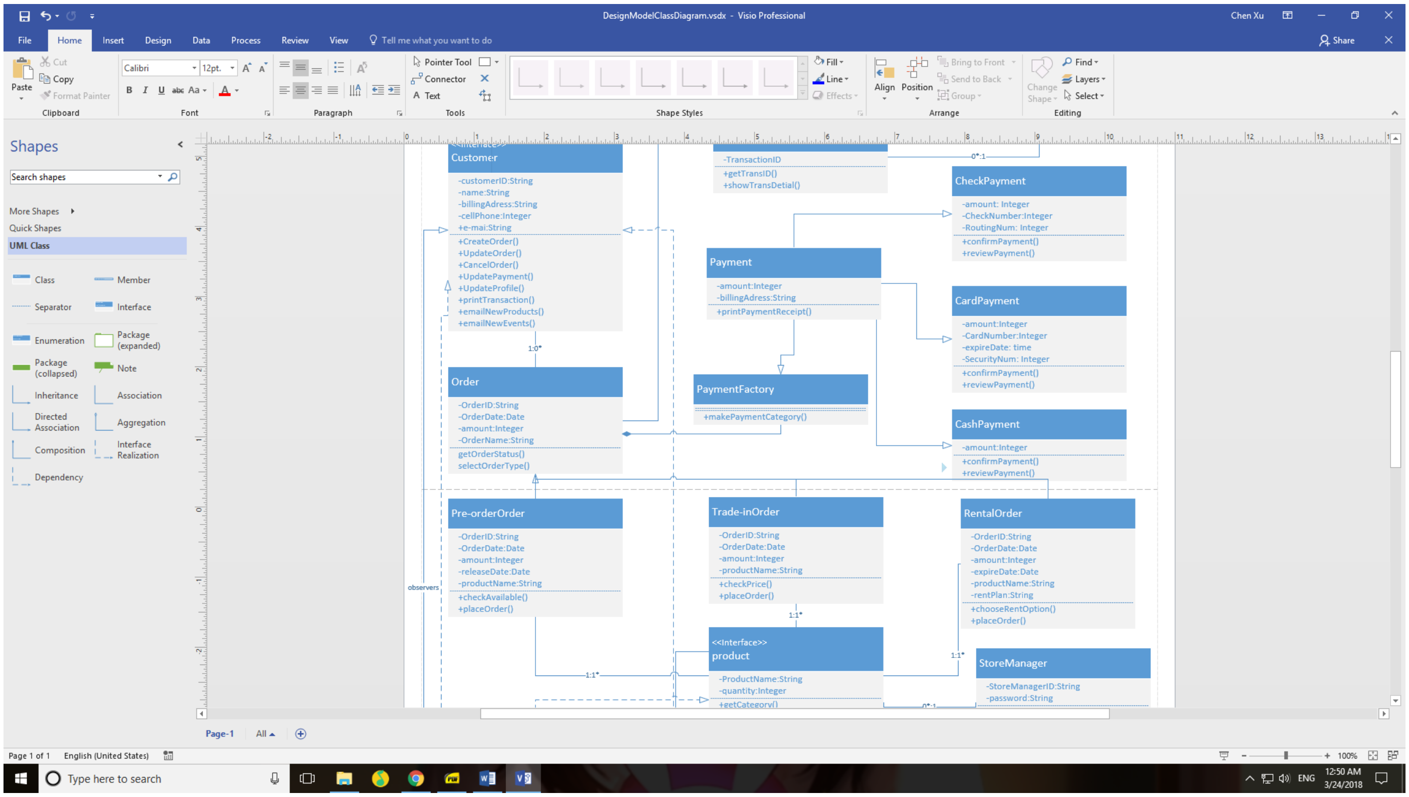
III. The EmailNotification Class implement the subject interface EmailSubscribe.

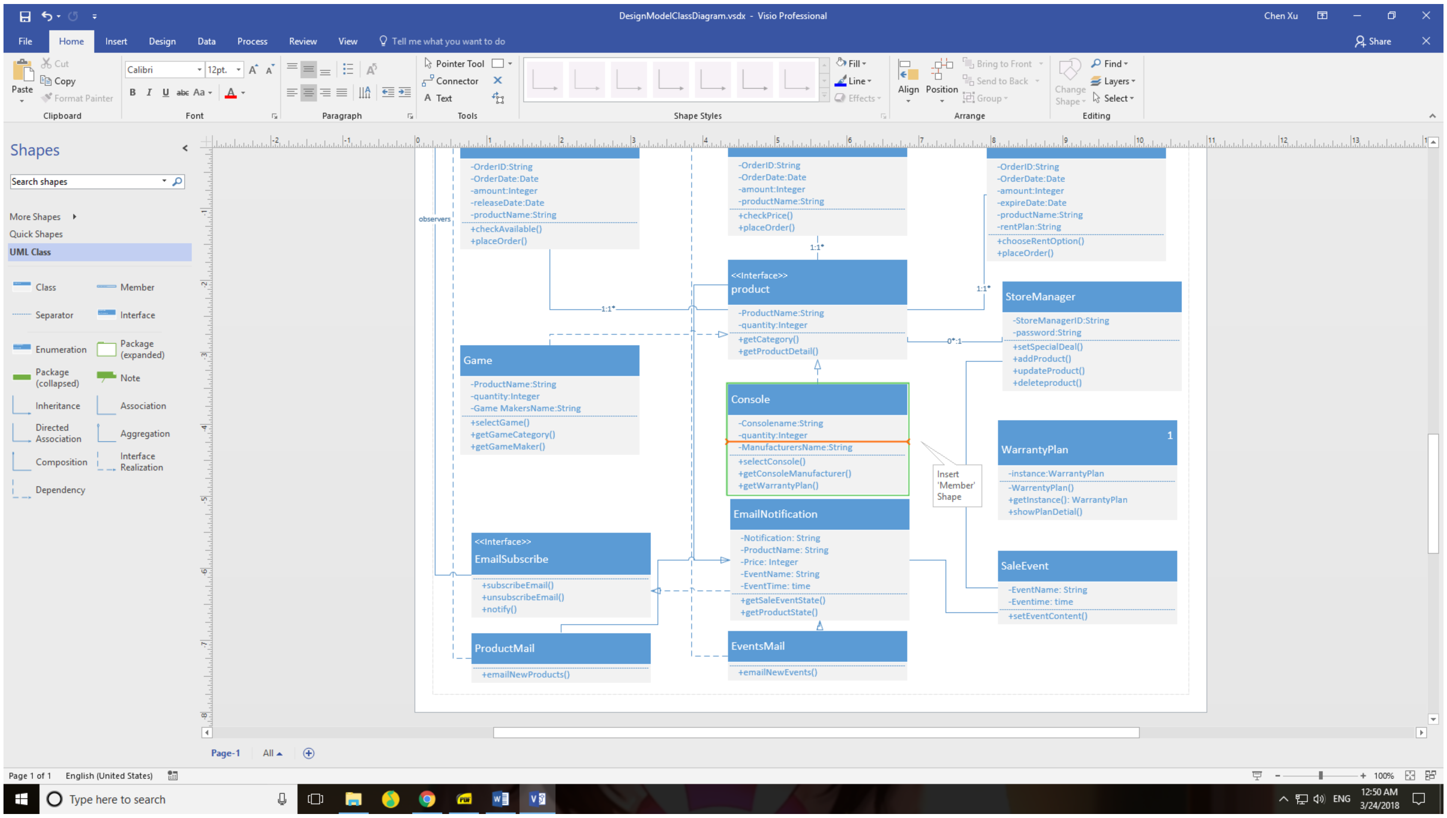
IV. ProductMail and EventMail Class are subclass of EmailNotification Class.

V. EmailNotification Class can get state of information from the product and SaleEvent Class.

5. Capture design model class diagram(s)

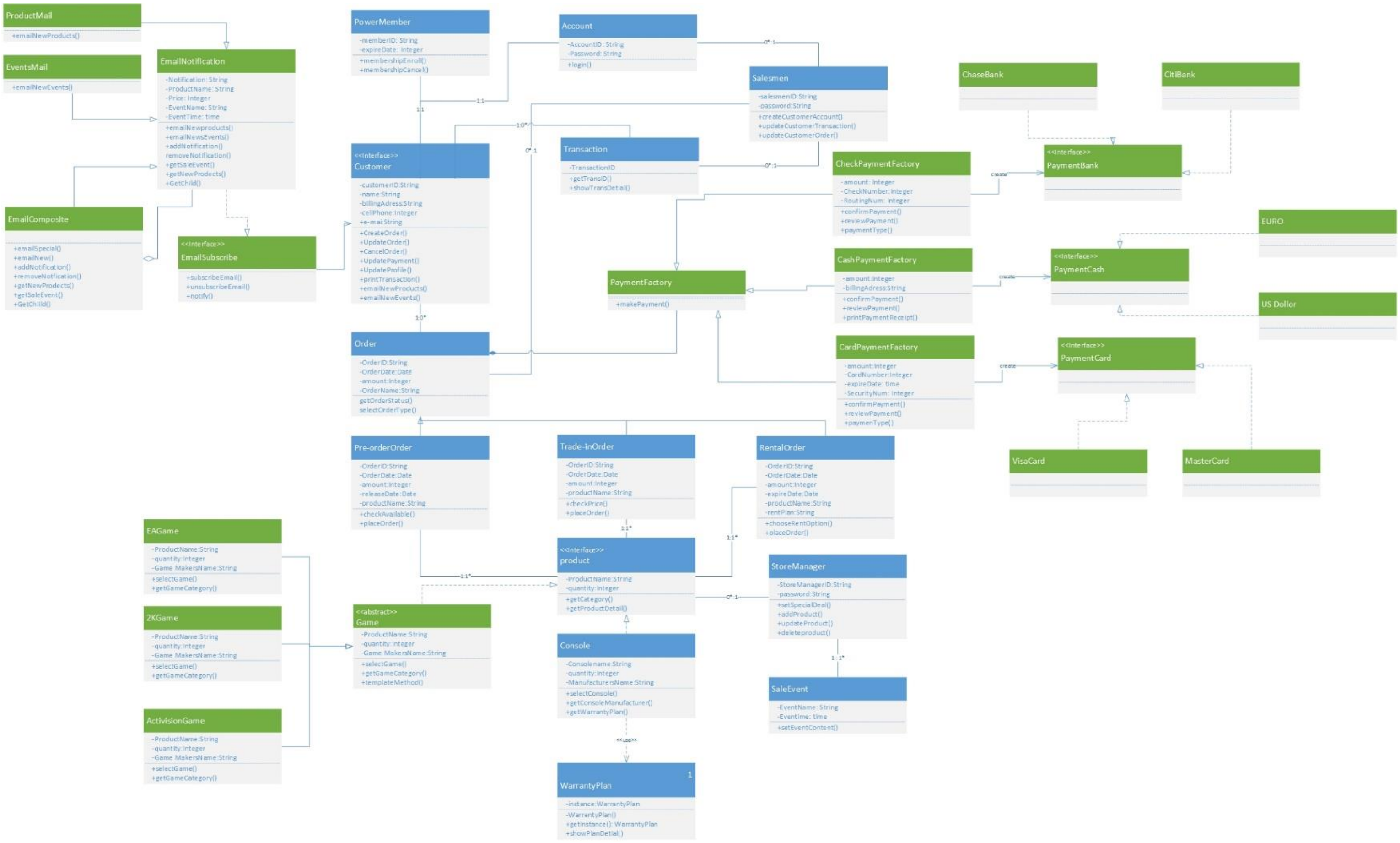





DesignModelClassDiagram.vsdx - Visio Professional
Chen Xu
File
Home
Insert
Design
Data
Process
Review
View
Tell me what you want to do
Share
Paste
Cut
Copy
Format Painter
Clipboard
Calibri
12pt.
Font
Paragraph
Pointer Tool
Connector
Text
Tools
Shape Styles
Fill
Line
Effects
Align
Position
Bring to Front
Send to Back
Group
Arrange
Change Shape
Find
Layers
Select
Editing
Shapes
Search shapes
More Shapes
Quick Shapes
UML Class
Class
Member
Separator
Interface
Enumeration
Package (expanded)
Package (collapsed)
Note
Inheritance
Association
Directed Association
Aggregation
Composition
Interface Realization
Dependency
-OrderID:String
-OrderDate:Date
-amount:Integer
-releaseDate:Date
-productName:String
+checkAvailable()
+placeOrder()
observers
-OrderID:String
-OrderDate:Date
-amount:Integer
-productName:String
+checkPrice()
+placeOrder()
-OrderID:String
-OrderDate:Date
-amount:Integer
-expireDate:Date
-productName:String
-rentPlan:String
+chooseRentOption()
+placeOrder()
1:1*
1:1*
1:1*
<<Interface>>
product
-ProductName:String
-quantity:Integer
+getCategory()
+getProductDetail()
0*:1
StoreManager
-StoreManagerID:String
-password:String
+setSpecialDeal()
+addProduct()
+updateProduct()
+deleteproduct()
Game
-ProductName:String
-quantity:Integer
-Game MakersName:String
+selectGame()
+getGameCategory()
+getGameMaker()
Console
-Consolename:String
-quantity:Integer
-ManufacturersName:String
+selectConsole()
+getConsoleManufacturer()
+getWarrantyPlan()
Insert 'Member' Shape
1
WarrantyPlan
-instance:WarrantyPlan
-WarrentyPlan()
+getInstance(): WarrantyPlan
+showPlanDetial()
EmailNotification
-Notification: String
-ProductName: String
-Price: Integer
-EventName: String
-EventTime: time
+getSaleEventState()
+getProductState()
<<Interface>>
EmailSubscribe
+subscribeEmail()
+unsubscribeEmail()
+notify()
SaleEvent
-EventName: String
-Eventime: time
+setEventContent()
ProductMail
+emailNewProducts()
EventsMail
+emailNewEvents()
Page-1
All
Page 1 of 1
English (United States)
100%
Type here to search
ENG
12:50 AM
3/24/2018

Assignment 4:

1. Design Model class diagram



2. List of the Design pattern(s)

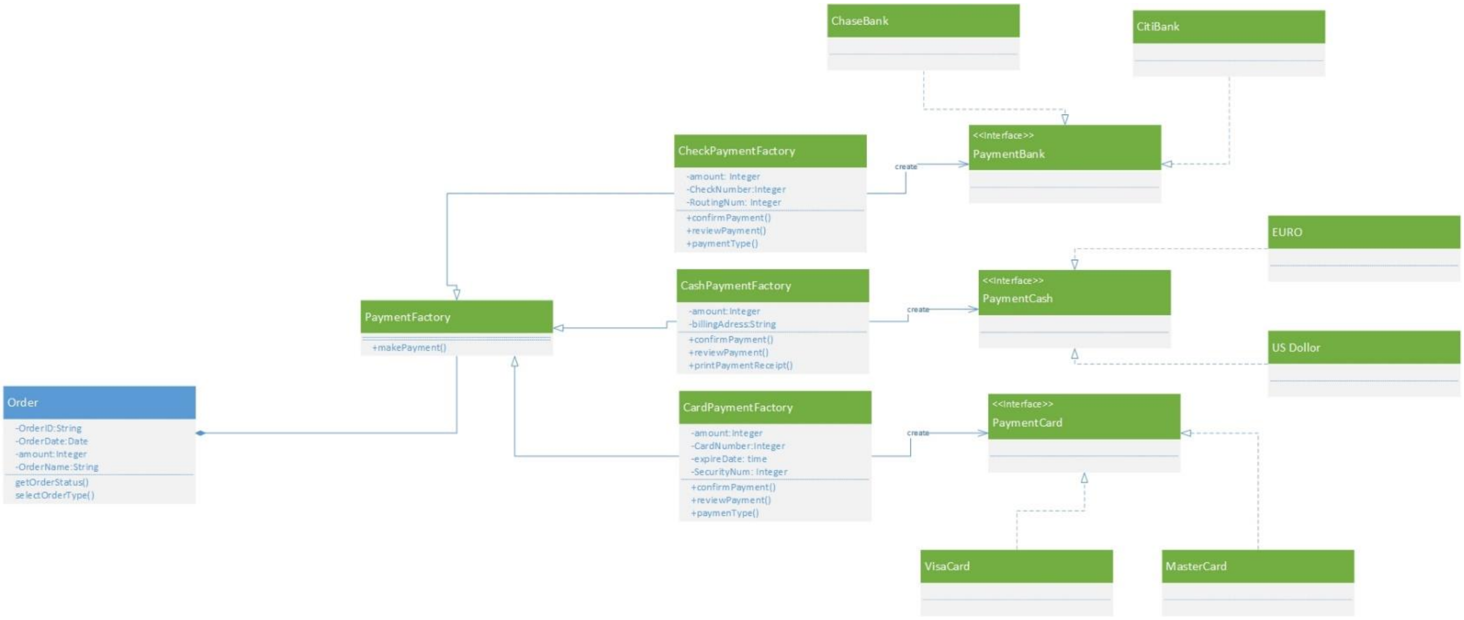
1) Abstract Factory Design Pattern

2) Composite Design Pattern

3) Template Method Design Pattern

3. Documentation of used design patterns

1) Abstract Factory Design Pattern: Implement make payment.

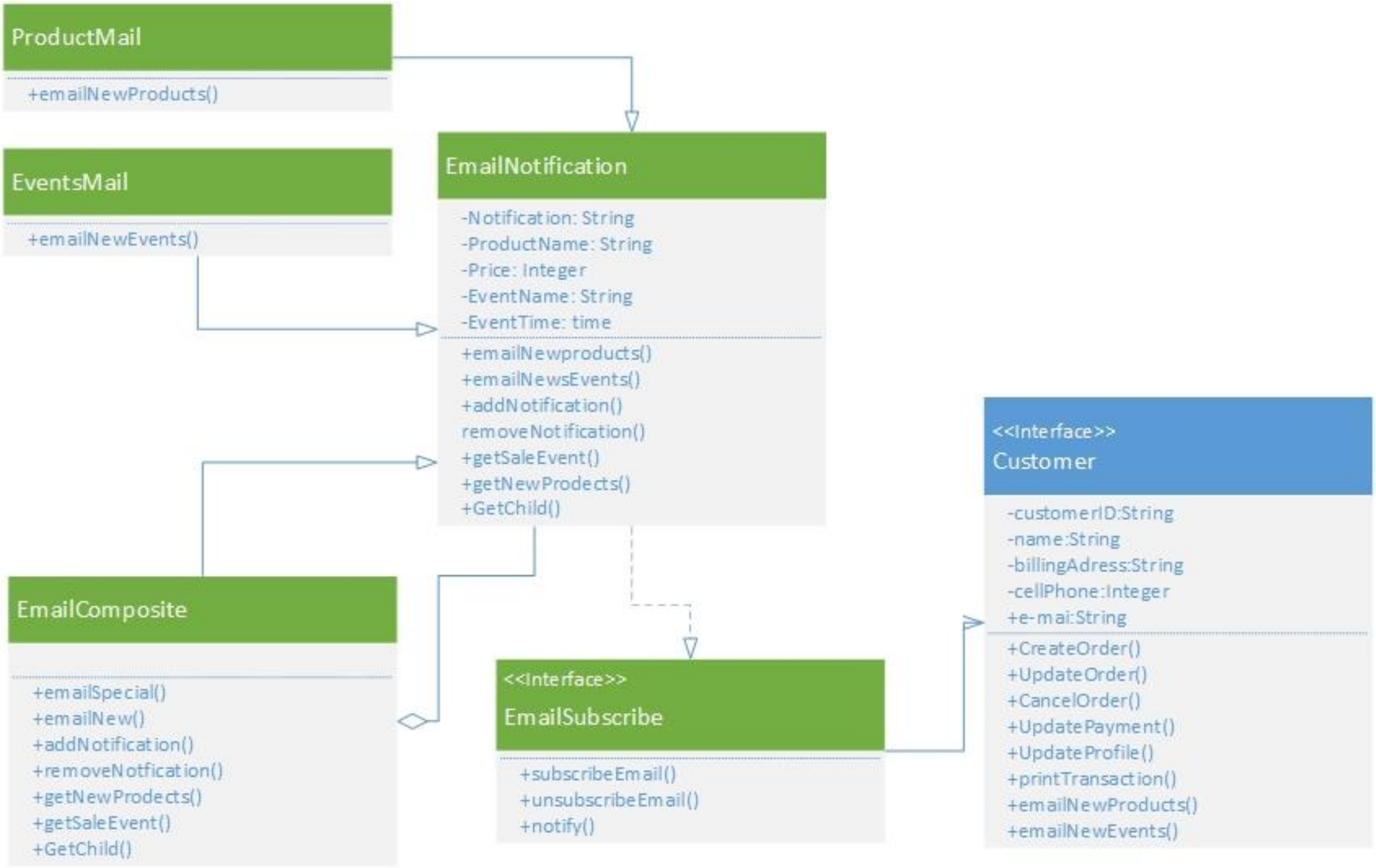


I. Use an abstract PaymentFactory Class to choose and create payment from individual factories.

II. CardPaymentFactory, CashPaymentFactory and CardPaymentFactory these concrete subClasses create a family of payments for each type.

III. PaymenBank, PaymentCash and PaymentCard are the interface of each payment types create parallel sets of their payment families.

2) Composite Design Pattern: Implement Email subscibptions.



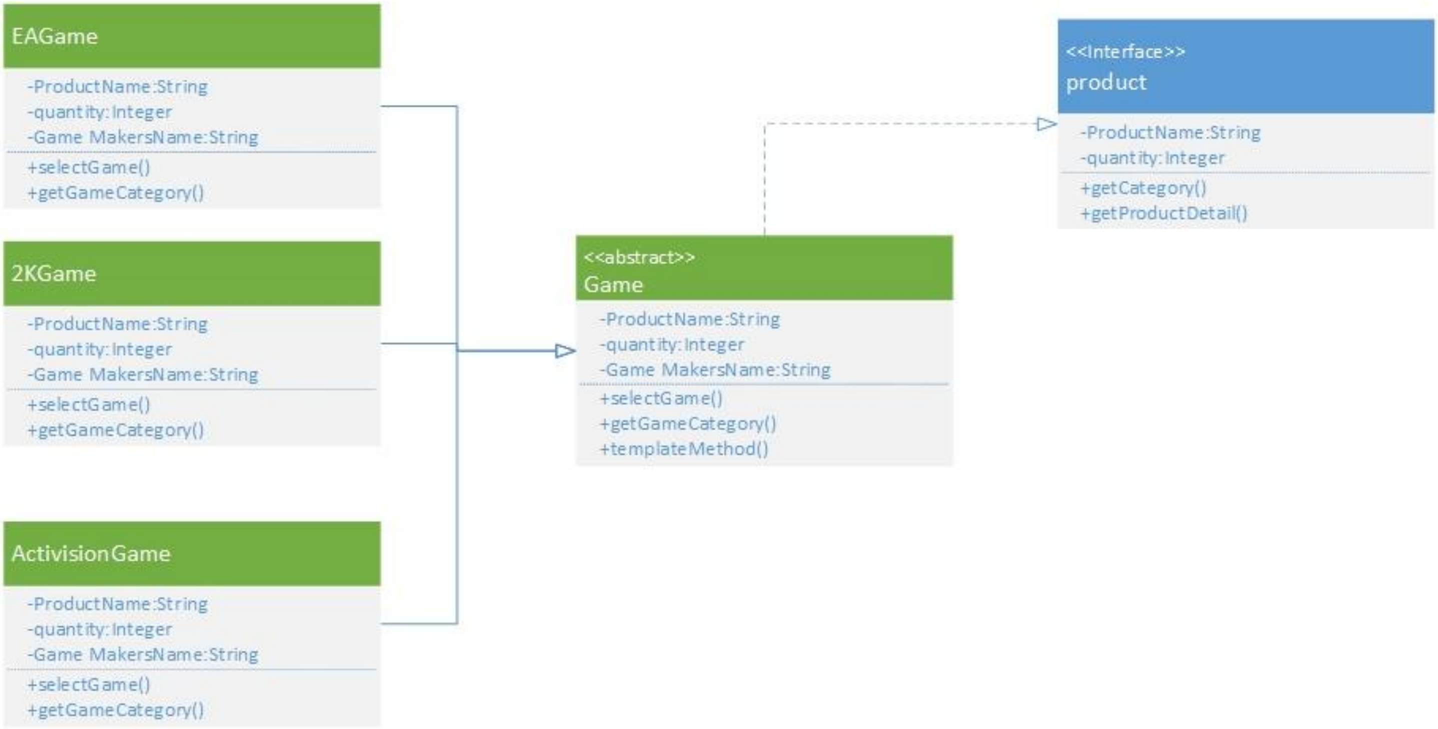
I. Use the EmailSubscibe Class as interface to manipulate the objects in the composition

II. Use EmailComposite Class to define the behavior of the components having children and to store child components. It implements the child related operations.

III. EmailNotification Class is the is the abstraction for all components, including EmailComposite Class. It declares the interface for objects in the composition.

IV. ProductMail and EventMail are the leaf Classes, they are the elemnts to help implement the composition.

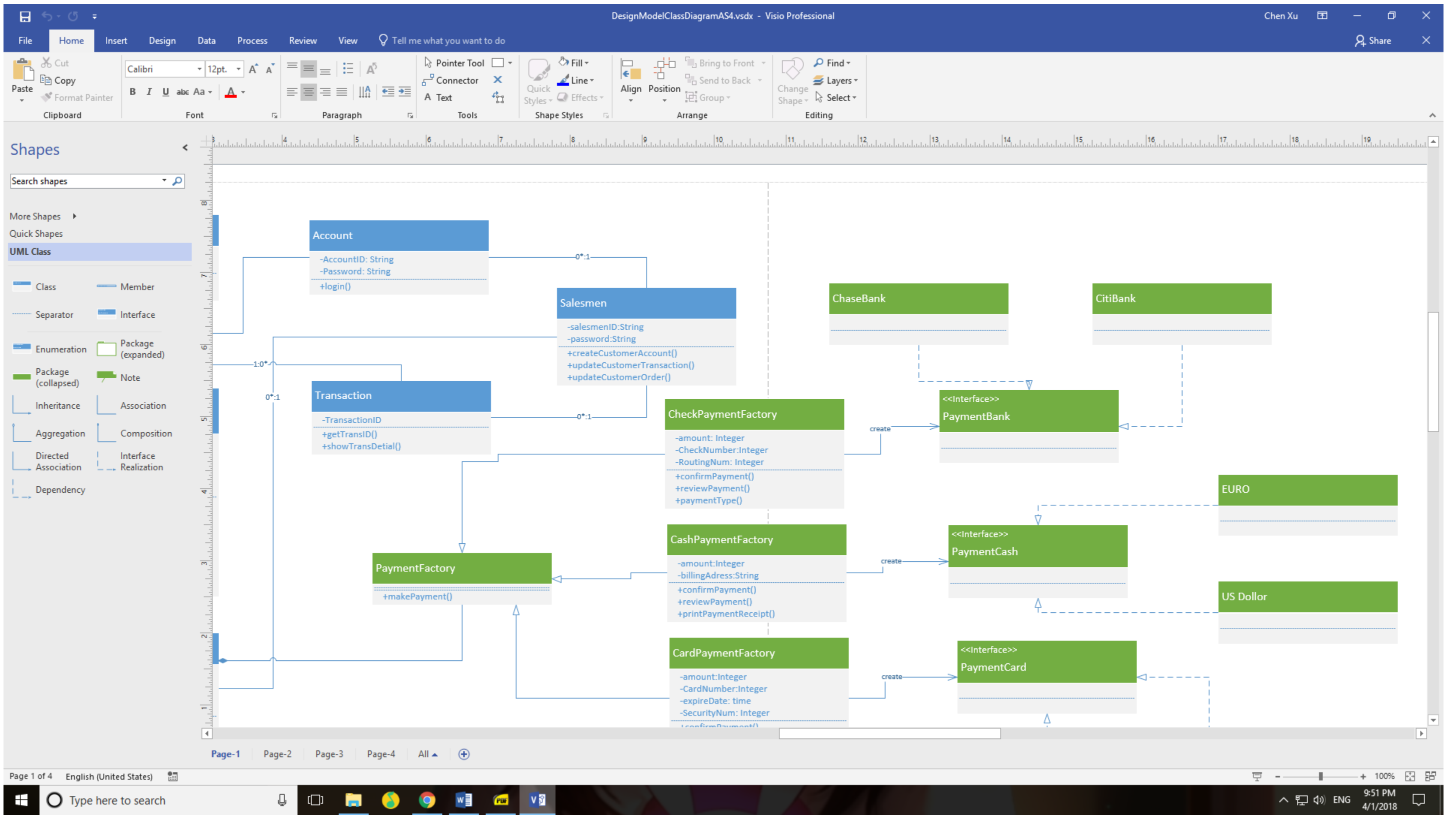
3) Template Method Design Pattern

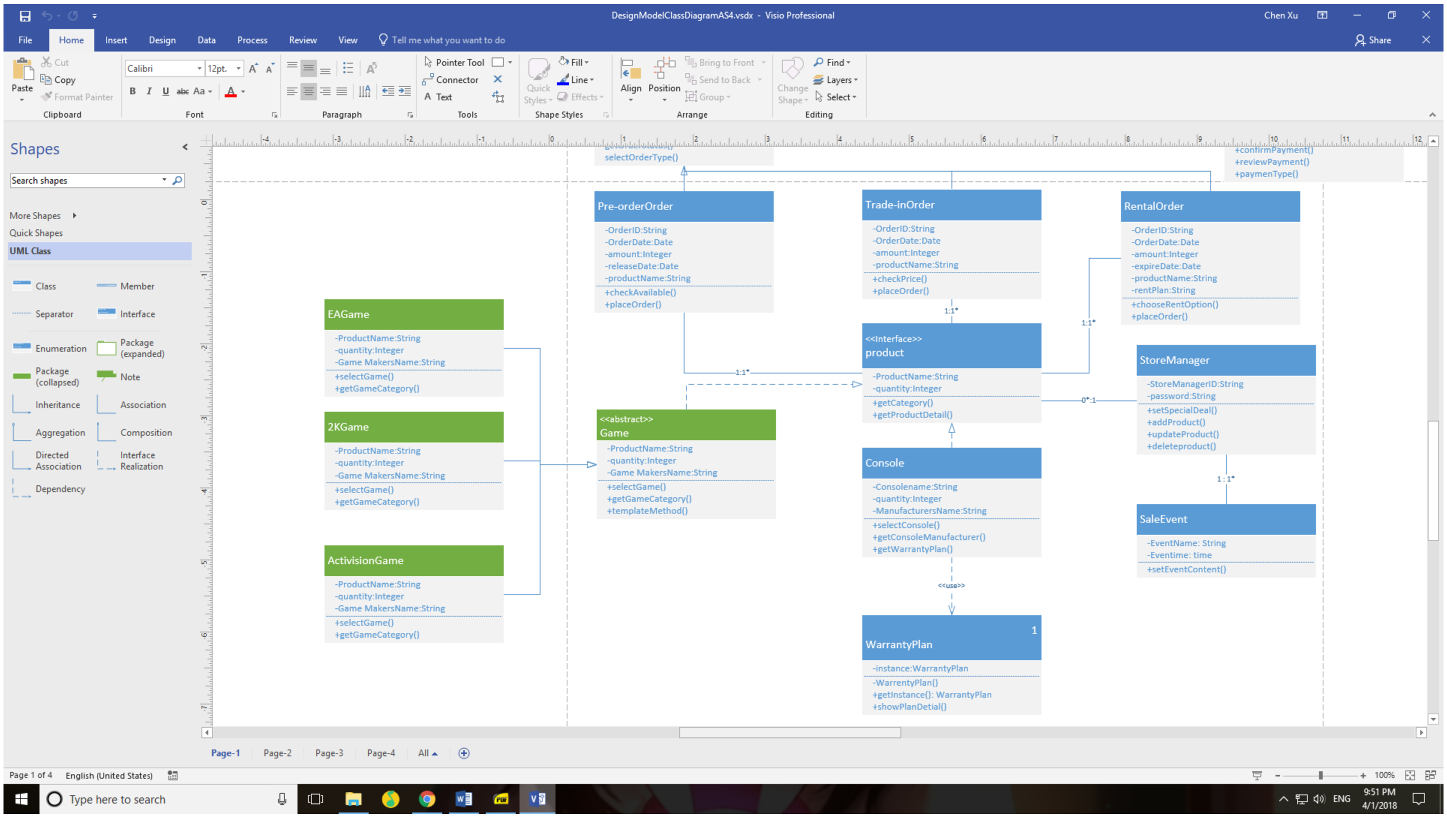


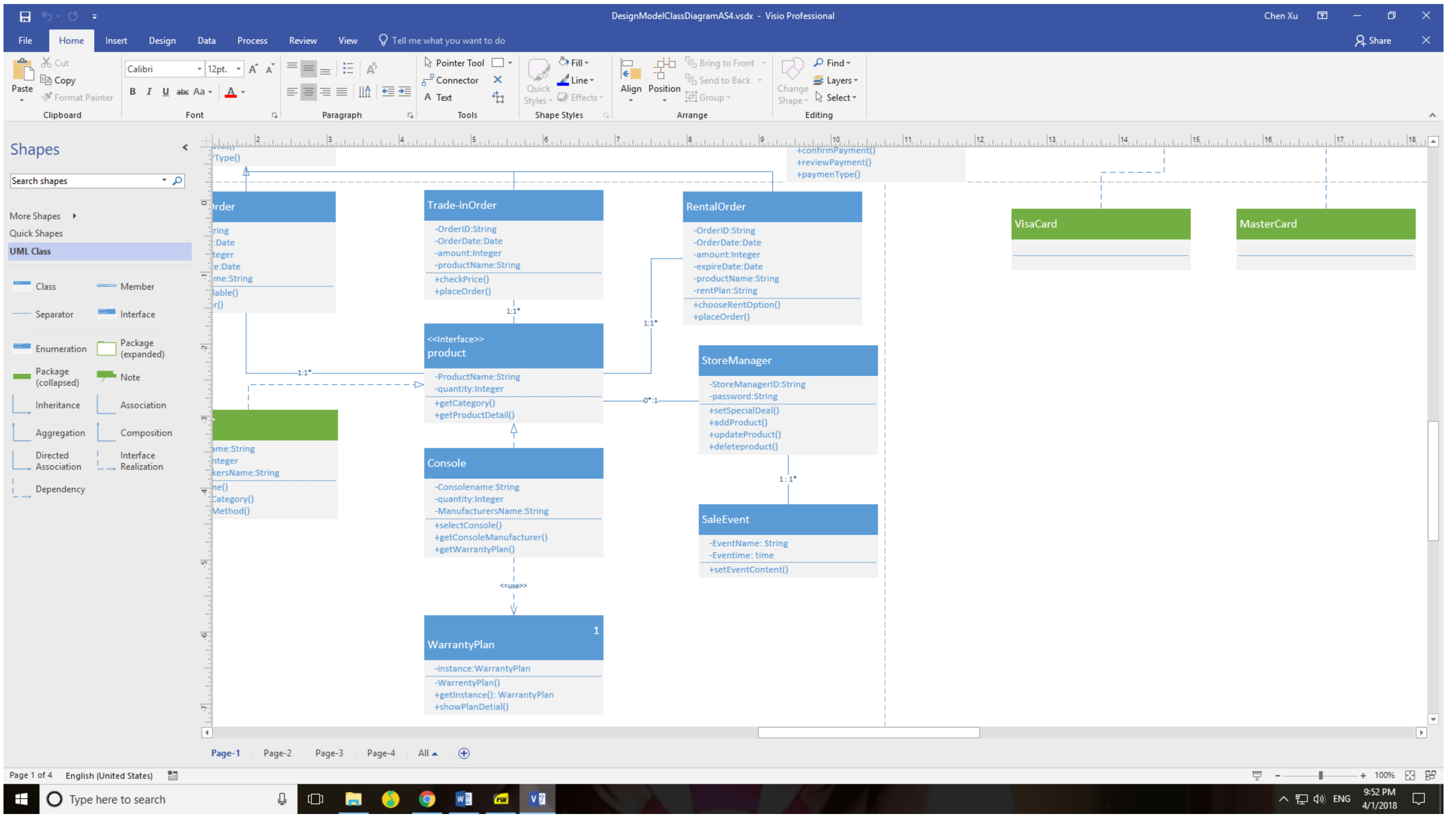
I. the Game Class defines a templateMethod() operation that defines the template of a behavior by implementing the invariant parts to each subClasses.

II. EAGame, 2KGame and ActivisionGame are subclasses that have defer part . They help template class to instantiated different category instances.

4. Capture design model class diagram(s)

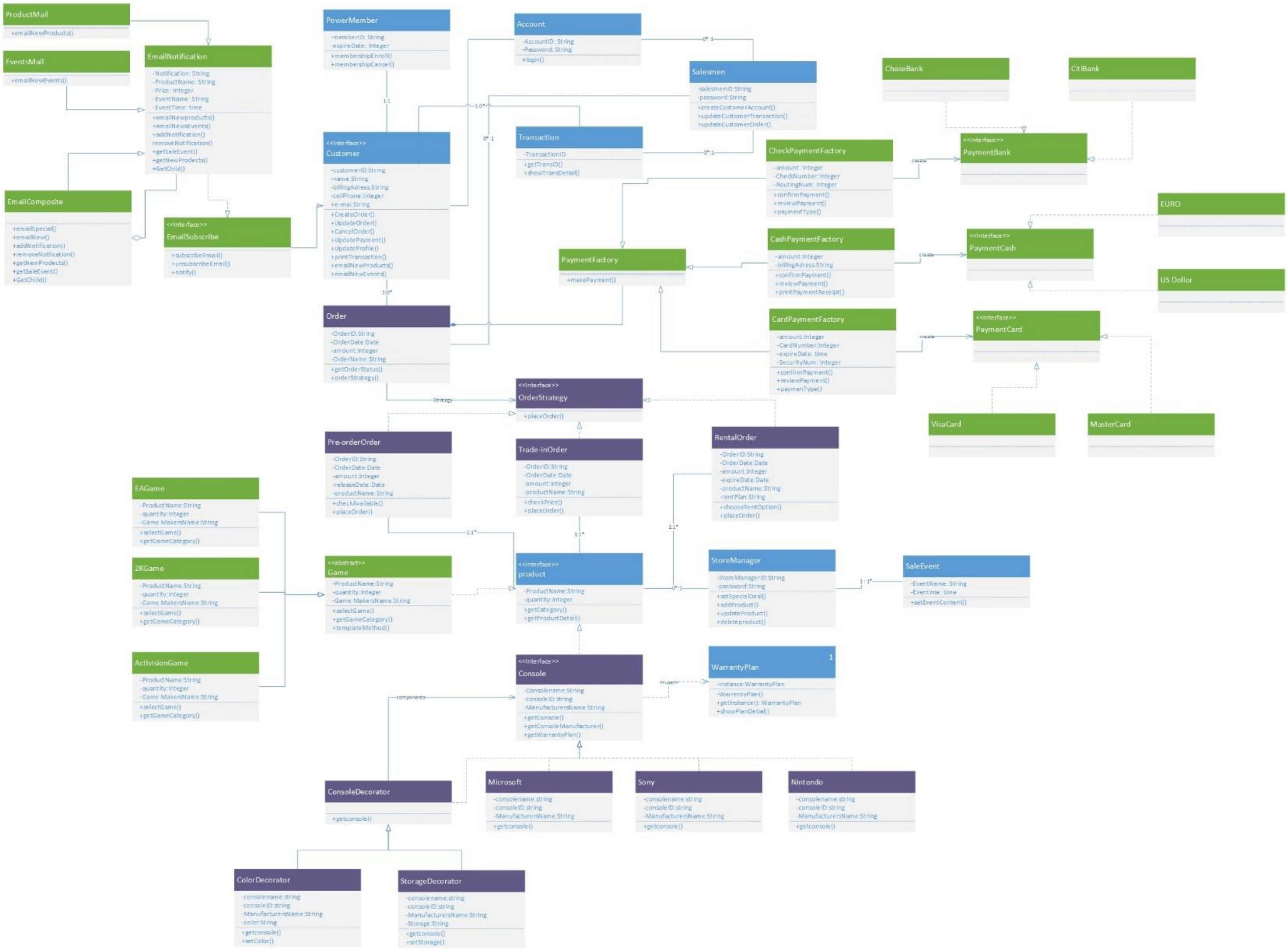





DesignModelClassDiagramAS4.vsdx - Visio Professional
Chen Xu
File
Home
Insert
Design
Data
Process
Review
View
Tell me what you want to do
Share
Paste
Cut
Copy
Format Painter
Clipboard
Calibri
12pt.
Font
Paragraph
Pointer Tool
Connector
Text
Tools
Quick Styles
Fill
Line
Effects
Shape Styles
Align
Position
Bring to Front
Send to Back
Group
Arrange
Change Shape
Find
Layers
Select
Editing
Shapes
Search shapes
More Shapes
Quick Shapes
UML Class
Class
Member
Separator
Interface
Enumeration
Package (expanded)
Package (collapsed)
Note
Inheritance
Association
Aggregation
Composition
Directed Association
Interface Realization
Dependency
Type()
rder
ring
:Date
teger
e:Date
me:String
lable()
r()
Trade-inOrder
-OrderID:String
-OrderDate:Date
-amount:Integer
-productName:String
+checkPrice()
+placeOrder()
+confirmPayment()
+reviewPayment()
+paymenType()
RentalOrder
-OrderID:String
-OrderDate:Date
-amount:Integer
-expireDate:Date
-productName:String
-rentPlan:String
+chooseRentOption()
+placeOrder()
VisaCard
MasterCard
1:1*
1:1*
1:1*
<<Interface>>
product
-ProductName:String
-quantity:Integer
+getCategory()
+getProductDetail()
StoreManager
-StoreManagerID:String
-password:String
+setSpecialDeal()
+addProduct()
+updateProduct()
+deleteproduct()
0*:1
1 : 1*
ame:String
nteger
kersName:String
ne()
Category()
Method()
Console
-Consolename:String
-quantity:Integer
-ManufacturersName:String
+selectConsole()
+getConsoleManufacturer()
+getWarrantyPlan()
SaleEvent
-EventName: String
-Eventime: time
+setEventContent()
<<use>>
1
WarrantyPlan
-instance:WarrantyPlan
-WarrentyPlan()
+getInstance(): WarrantyPlan
+showPlanDetial()
Page-1
Page-2
Page-3
Page-4
All
Page 1 of 4
English (United States)
100%
Type here to search
ENG
9:52 PM
4/1/2018

Assignment 5:

1. Design Model class diagram



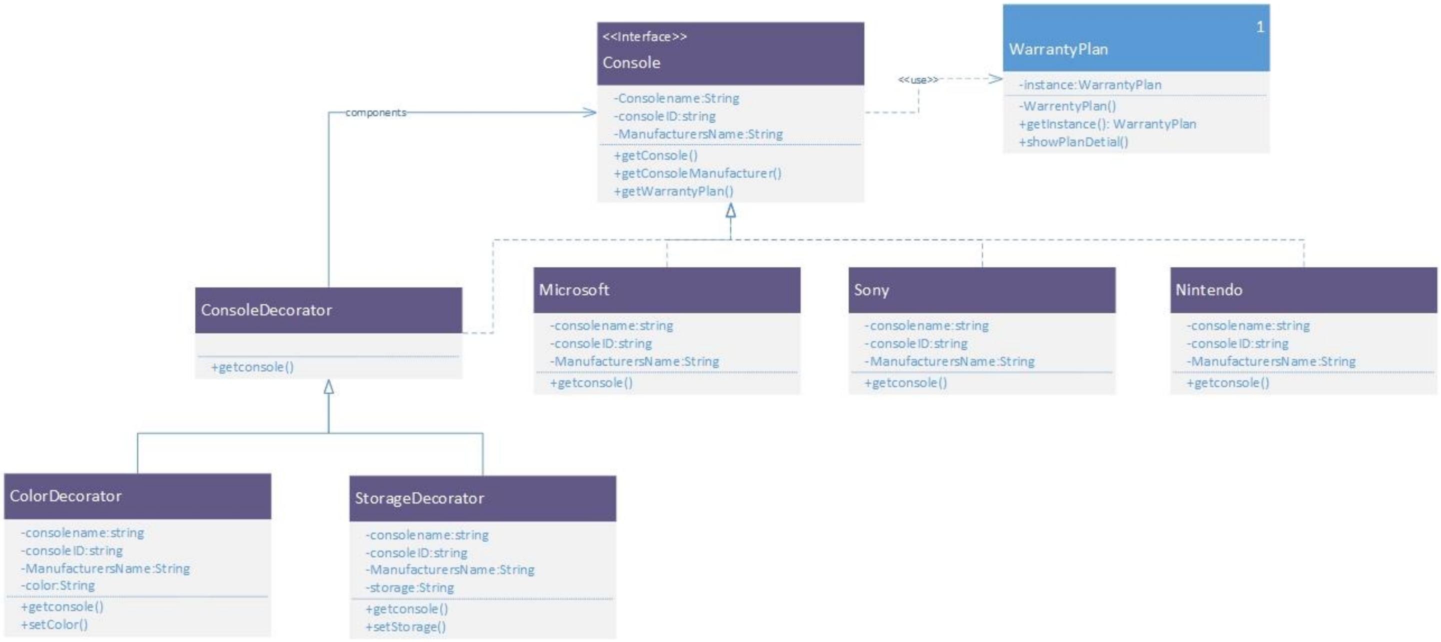
2. List of the Design pattern(s)

1) Decorator Design Pattern

2) Strategy Design Pattern

3. Documentation of used design patterns

1) Decorator Design Pattern: Implement console selection.

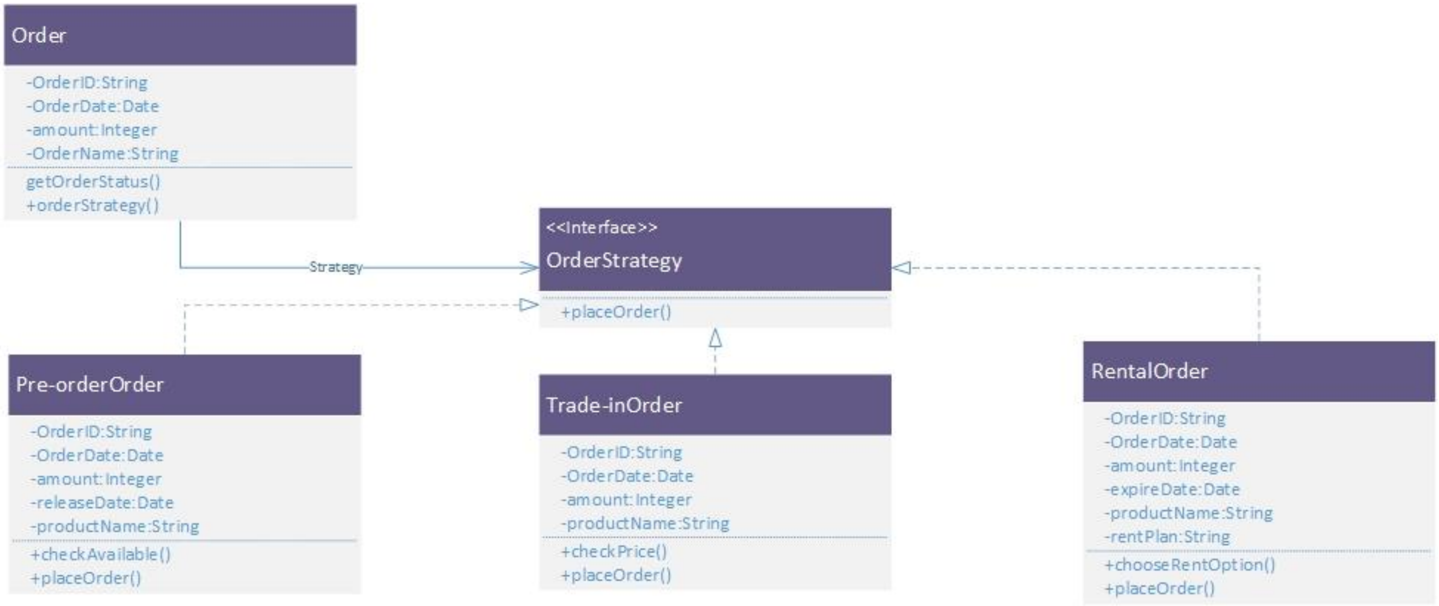


I. The abstract Class consoleDecorator maintains the reference of the Component and forwards all requests to it.

II. Subclasses colorDecortaor and StorageDecorator implement additional behavior (select color and storage) that should be added to the Component.

III. The Client object works through consoleDecorator objects to extend the functionality of a Component(Microsoft,Sony,Nintendo) object.

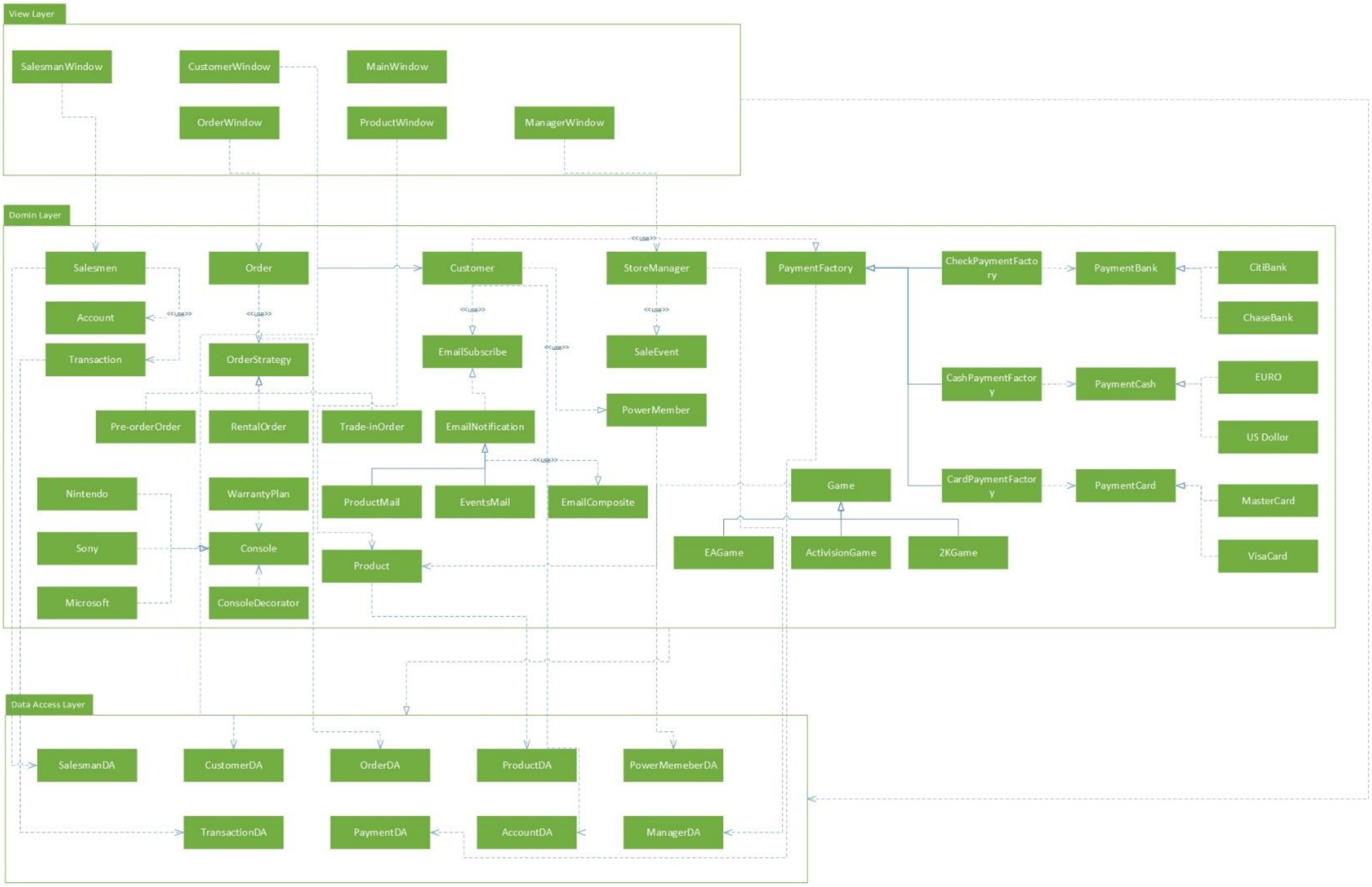
2) Strategy Design Pattern: Implement place order.



I. the Order class doesn't implement placeOrder() directly. Instead, Order Class refers to the OrderStrategy interface for performing an algorithm.

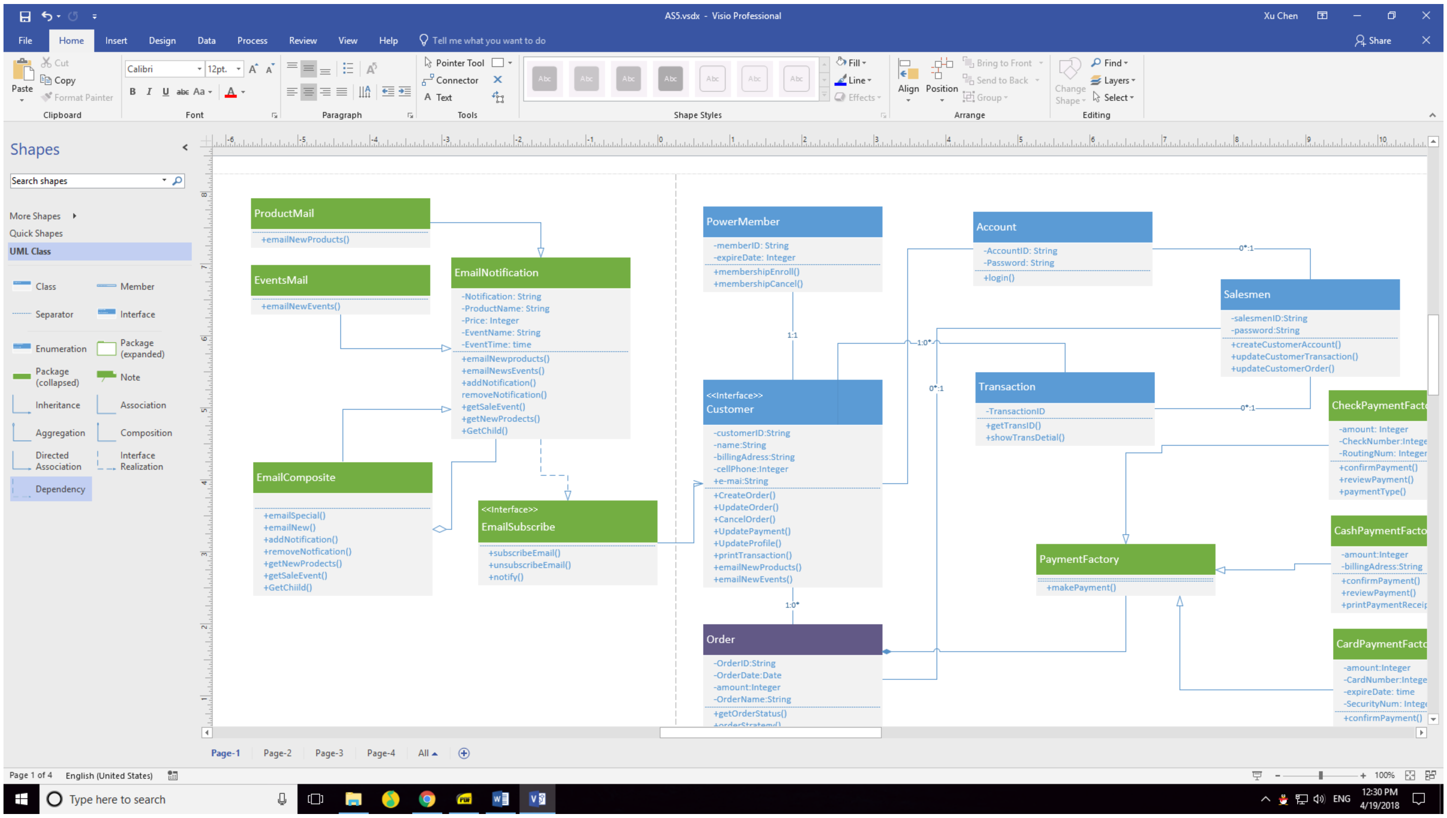
II. The Pre-orderOrder Trader-inOrder and RentalOrder classes implement the Strategy interface, which implement the placeOrder algorithm.

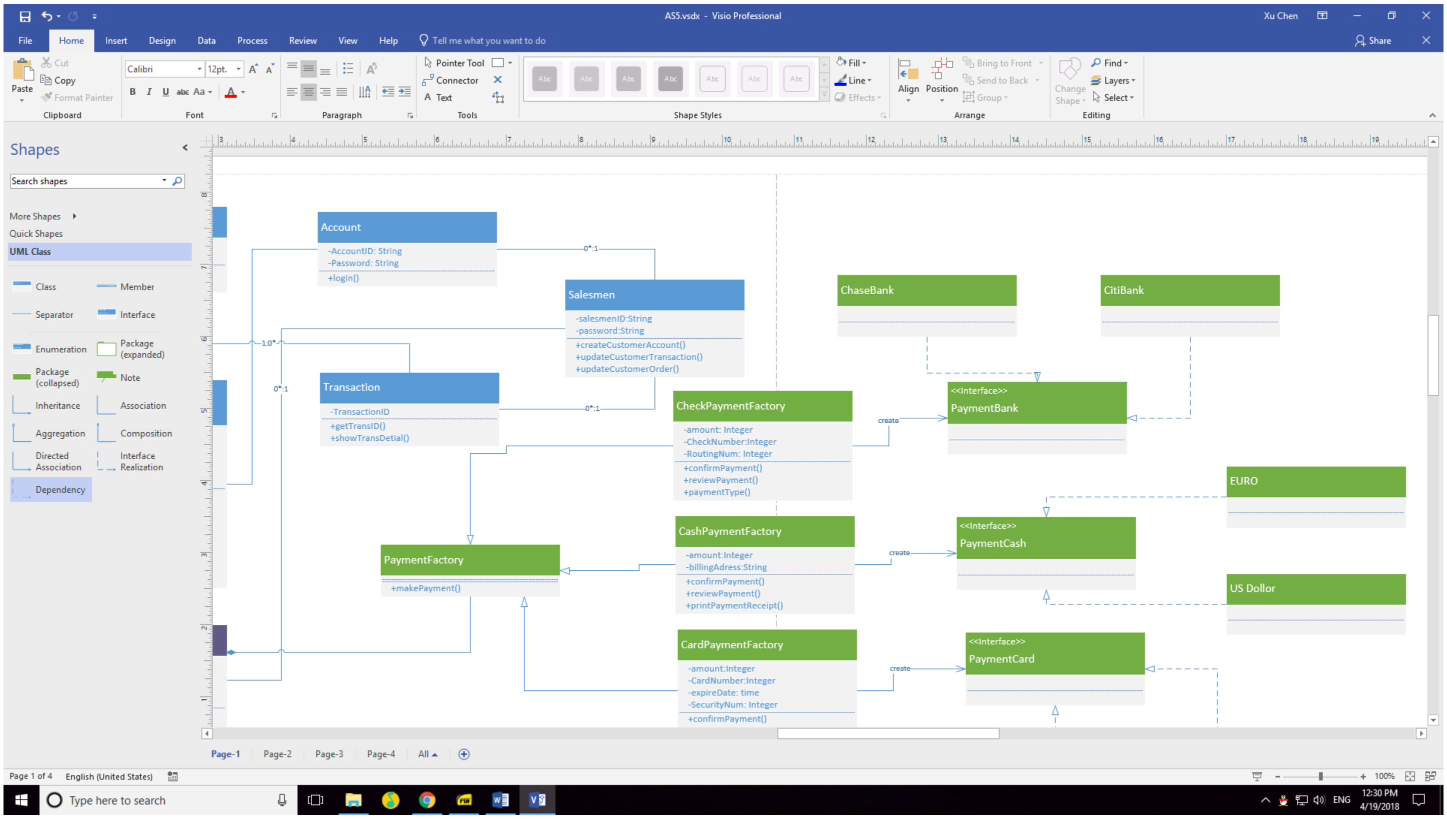
2. MVC Architectural Pattern

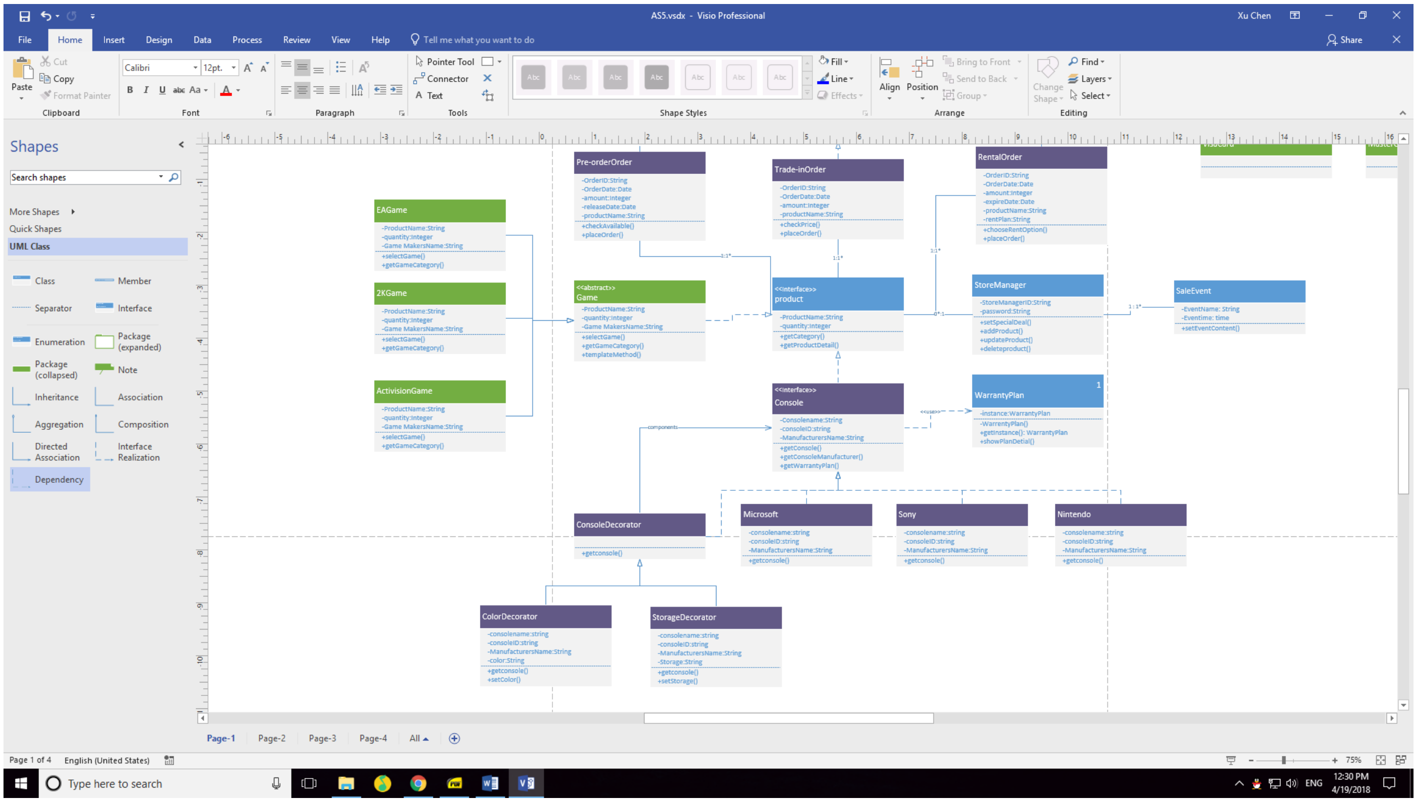


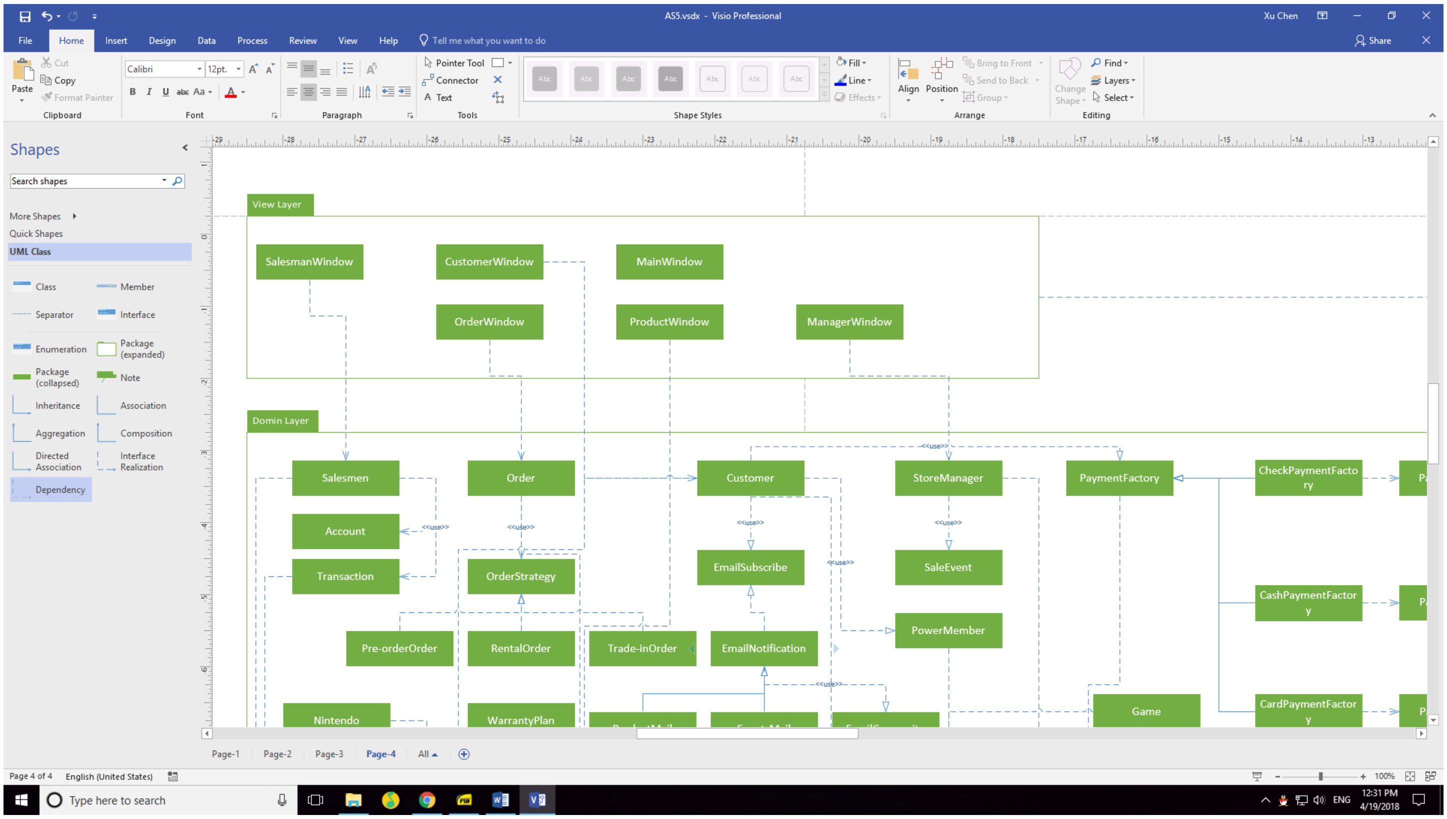
3. Capture design model class diagram(s) and

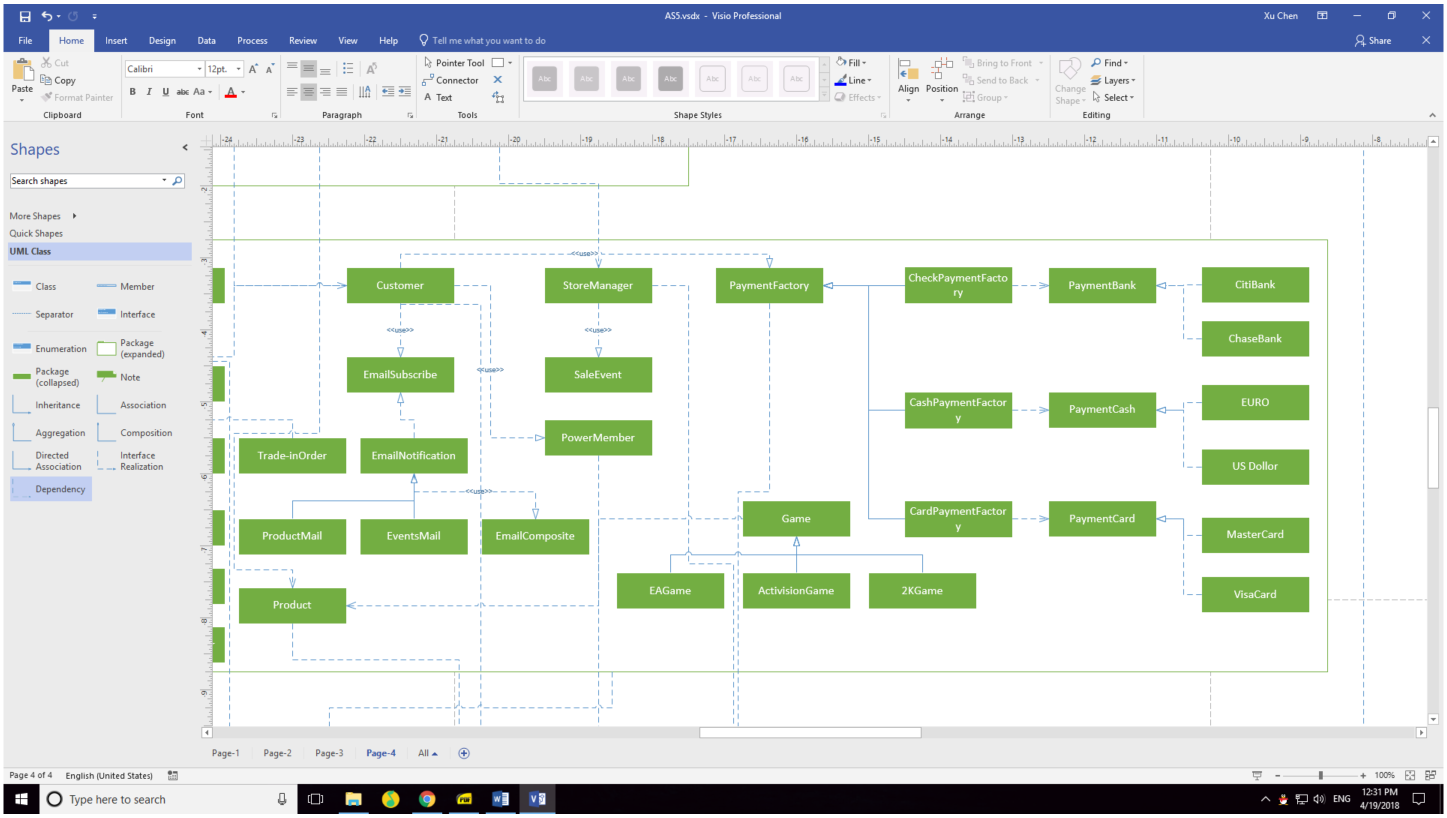
MVC Architectural Pattern

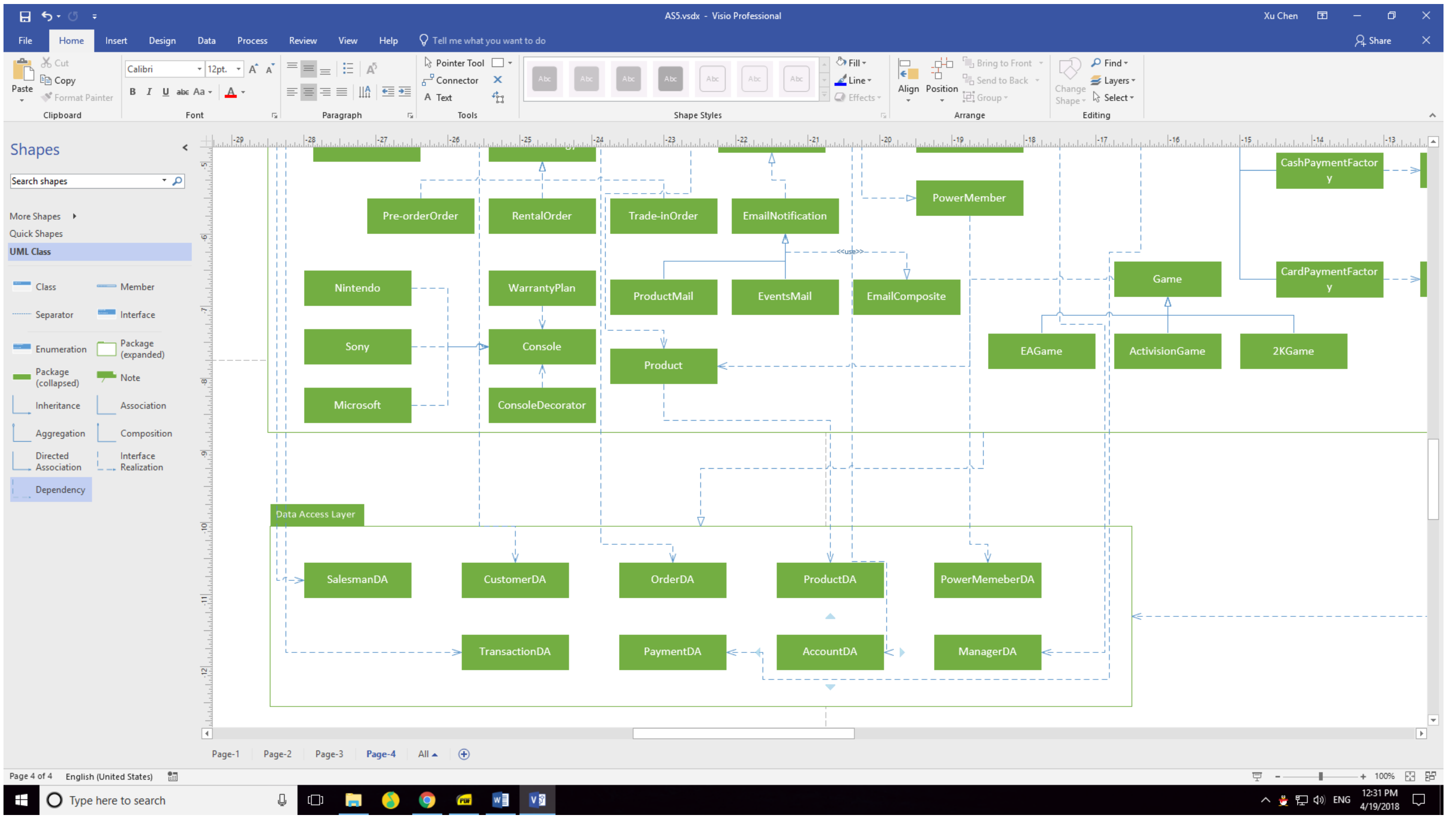

A5.vsdx - Visio Professional
Xu Chen
File
Home
Insert
Design
Data
Process
Review
View
Help
Tell me what you want to do
Share
Shapes
Search shapes
More Shapes
Quick Shapes
UML Class
Class
Member
Separator
Interface
Enumeration
Package (expanded)
Package (collapsed)
Note
Inheritance
Association
Aggregation
Composition
Directed Association
Interface Realization
Dependency
ProductMail
+emailNewProducts()
EventsMail
+emailNewEvents()
EmailNotification
-Notification: String
-ProductName: String
-Price: Integer
-EventName: String
-EventTime: time
+emailNewproducts()
+emailNewsEvents()
+addNotification()
removeNotification()
+getSaleEvent()
+getNewProdects()
+GetChild()
EmailComposite
+emailSpecial()
+emailNew()
+addNotification()
+removeNotfication()
+getNewProdects()
+getSaleEvent()
+GetChiild()
<<Interface>>
EmailSubscribe
+subscribeEmail()
+unsubscribeEmail()
+notify()
PowerMember
-memberID: String
-expireDate: Integer
+membershipEnroll()
+membershipCancel()
1:1
<<Interface>>
Customer
-customerID:String
-name:String
-billingAdress:String
-cellPhone:Integer
+e-mai:String
+CreateOrder()
+UpdateOrder()
+CancelOrder()
+UpdatePayment()
+UpdateProfile()
+printTransaction()
+emailNewProducts()
+emailNewEvents()
1:0*
Order
-OrderID:String
-OrderDate:Date
-amount:Integer
-OrderName:String
+getOrderStatus()
+orderStratemy()
Account
-AccountID: String
-Password: String
+login()
0*:1
Salesmen
-salesmenID:String
-password:String
+createCustomerAccount()
+updateCustomerTransaction()
+updateCustomerOrder()
1:0*
0*:1
Transaction
-TransactionID
+getTransID()
+showTransDetial()
0*:1
CheckPaymentFact
-amount: Integer
-CheckNumber:Intege
-RoutingNum: Integer
+confirmPayment()
+reviewPayment()
+paymentType()
CashPaymentFacto
-amount:Integer
-billingAdress:String
+confirmPayment()
+reviewPayment()
+printPaymentReceip
PaymentFactory
+makePayment()
CardPaymentFacto
-amount:Integer
-CardNumber:Intege
-expireDate: time
-SecurityNum: Intege
+confirmPayment()
Page-1
Page-2
Page-3
Page-4
All
Page 1 of 4
English (United States)
100%
Type here to search
ENG
12:30 PM
4/19/2018


A5.vsdx - Visio Professional
Xu Chen
File
Home
Insert
Design
Data
Process
Review
View
Help
Tell me what you want to do
Share
Shapes
Search shapes
More Shapes
Quick Shapes
UML Class
Class
Member
Separator
Interface
Enumeration
Package (expanded)
Package (collapsed)
Note
Inheritance
Association
Aggregation
Composition
Directed Association
Interface Realization
Dependency
Account
-AccountID: String
-Password: String
+login()
0*:1
Salesmen
-salesmenID:String
-password:String
+createCustomerAccount()
+updateCustomerTransaction()
+updateCustomerOrder()
1:0*
0*:1
Transaction
-TransactionID
+getTransID()
+showTransDetial()
0*:1
PaymentFactory
+makePayment()
CheckPaymentFactory
-amount: Integer
-CheckNumber:Integer
-RoutingNum: Integer
+confirmPayment()
+reviewPayment()
+paymentType()
CashPaymentFactory
-amount:Integer
-billingAdress:String
+confirmPayment()
+reviewPayment()
+printPaymentReceipt()
CardPaymentFactory
-amount:Integer
-CardNumber:Integer
-expireDate: time
-SecurityNum: Integer
+confirmPayment()
ChaseBank
CitiBank
create
<<Interface>>
PaymentBank
EURO
create
<<Interface>>
PaymentCash
US Dollor
create
<<Interface>>
PaymentCard
Page-1
Page-2
Page-3
Page-4
All
Page 1 of 4
English (United States)
100%
Type here to search
ENG
12:30 PM
4/19/2018


AS5.vsdx - Visio Professional
Xu Chen
File
Home
Insert
Design
Data
Process
Review
View
Help
Tell me what you want to do
Share
Paste
Cut
Copy
Format Painter
Clipboard
Calibri
12pt.
Font
Paragraph
Pointer Tool
Connector
Text
Tools
Shape Styles
Fill
Line
Effects
Align
Position
Bring to Front
Send to Back
Group
Arrange
Change Shape
Find
Layers
Select
Editing
Shapes
Search shapes
More Shapes
Quick Shapes
UML Class
Class
Member
Separator
Interface
Enumeration
Package (expanded)
Package (collapsed)
Note
Inheritance
Association
Aggregation
Composition
Directed Association
Interface Realization
Dependency
EAGame
-ProductName:String
-quantity:Integer
-Game MakersName:String
+selectGame()
+getGameCategory()
2KGame
-ProductName:String
-quantity:Integer
-Game MakersName:String
+selectGame()
+getGameCategory()
ActivisionGame
-ProductName:String
-quantity:Integer
-Game MakersName:String
+selectGame()
+getGameCategory()
Pre-orderOrder
-OrderID:String
-OrderDate:Date
-amount:Integer
-releaseDate:Date
-productName:String
+checkAvailable()
+placeOrder()
<<abstract>>
Game
-ProductName:String
-quantity:Integer
-Game MakersName:String
+selectGame()
+getGameCategory()
+templateMethod()
Trade-inOrder
-OrderID:String
-OrderDate:Date
-amount:Integer
-productName:String
+checkPrice()
+placeOrder()
RentalOrder
-OrderID:String
-OrderDate:Date
-amount:Integer
-expireDate:Date
-productName:String
-rentPlan:String
+chooseRentOption()
+placeOrder()
1:1*
1:1*
1:1*
<<Interface>>
product
-ProductName:String
-quantity:Integer
+getCategory()
+getProductDetail()
0*:1
StoreManager
-StoreManagerID:String
-password:String
+setSpecialDeal()
+addProduct()
+updateProduct()
+deleteproduct()
1 : 1*
SaleEvent
-EventName: String
-Eventime: time
+setEventContent()
<<Interface>>
Console
-Consolename:String
-consoleID:string
-ManufacturersName:String
+getConsole()
+getConsoleManufacturer()
+getWarrantyPlan()
<<use>>
1
WarrantyPlan
-instance:WarrantyPlan
-WarrentyPlan()
+getInstance(): WarrantyPlan
+showPlanDetial()
components
ConsoleDecorator
+getconsole()
Microsoft
-consolename:string
-consoleID:string
-ManufacturersName:String
+getconsole()
Sony
-consolename:string
-consoleID:string
-ManufacturersName:String
+getconsole()
Nintendo
-consolename:string
-consoleID:string
-ManufacturersName:String
+getconsole()
ColorDecorator
-consolename:string
-consoleID:string
-ManufacturersName:String
-color:String
+getconsole()
+setColor()
StorageDecorator
-consolename:string
-consoleID:string
-ManufacturersName:String
-Storage:String
+getconsole()
+setStorage()
Page-1
Page-2
Page-3
Page-4
All
Page 1 of 4
English (United States)
75%
Type here to search
ENG
12:30 PM
4/19/2018


AS5.vsdx - Visio Professional
Xu Chen
File
Home
Insert
Design
Data
Process
Review
View
Help
Tell me what you want to do
Share
Shapes
Search shapes
More Shapes
Quick Shapes
UML Class
Class
Member
Separator
Interface
Enumeration
Package (expanded)
Package (collapsed)
Note
Inheritance
Association
Aggregation
Composition
Directed Association
Interface Realization
Dependency
View Layer
SalesmanWindow
CustomerWindow
MainWindow
OrderWindow
ProductWindow
ManagerWindow
Domin Layer
<<use>>
Salesmen
Order
Customer
StoreManager
PaymentFactory
CheckPaymentFactory
Account
<<use>>
<<use>>
<<use>>
<<use>>
Transaction
OrderStrategy
EmailSubscribe
<<use>>
SaleEvent
CashPaymentFactory
Pre-orderOrder
RentalOrder
Trade-inOrder
EmailNotification
PowerMember
<<use>>
Nintendo
WarrantyPlan
Game
CardPaymentFactory
Page-1
Page-2
Page-3
Page-4
All
Page 4 of 4
English (United States)
100%
Type here to search
ENG
12:31 PM
4/19/2018


A5.vsdx - Visio Professional
Xu Chen
File
Home
Insert
Design
Data
Process
Review
View
Help
Tell me what you want to do
Share
Shapes
Search shapes
More Shapes
Quick Shapes
UML Class
Class
Member
Separator
Interface
Enumeration
Package (expanded)
Package (collapsed)
Note
Inheritance
Association
Aggregation
Composition
Directed Association
Interface Realization
Dependency
Customer
StoreManager
PaymentFactory
CheckPaymentFactory
PaymentBank
CitiBank
ChaseBank
EmailSubscribe
SaleEvent
PowerMember
CashPaymentFactory
PaymentCash
EURO
US Dollor
Trade-inOrder
EmailNotification
ProductMail
EventsMail
EmailComposite
Game
CardPaymentFactory
PaymentCard
MasterCard
VisaCard
EAGame
ActivisionGame
2KGame
Product
<<use>>
Page-1
Page-2
Page-3
Page-4
All
Page 4 of 4
English (United States)
100%
Type here to search
12:31 PM
4/19/2018


A5.vsdx - Visio Professional
Xu Chen
Shapes
UML Class
Pre-orderOrder
RentalOrder
Trade-inOrder
EmailNotification
PowerMember
CashPaymentFactory
Nintendo
WarrantyPlan
ProductMail
EventsMail
EmailComposite
Game
CardPaymentFactory
Sony
Console
EAGame
ActivisionGame
2KGame
Product
Microsoft
ConsoleDecorator
<<use>>
Data Access Layer
SalesmanDA
CustomerDA
OrderDA
ProductDA
PowerMemeberDA
TransactionDA
PaymentDA
AccountDA
ManagerDA
Page-1
Page-2
Page-3
Page-4
All
Page 4 of 4
English (United States)
100%
Type here to search
12:31 PM
4/19/2018