

Software Modeling Dev with UML  
CSP- 586  
Assignment #3

*Sivasenthil Namachivayan*  
*A20391478*

## Table of Contents

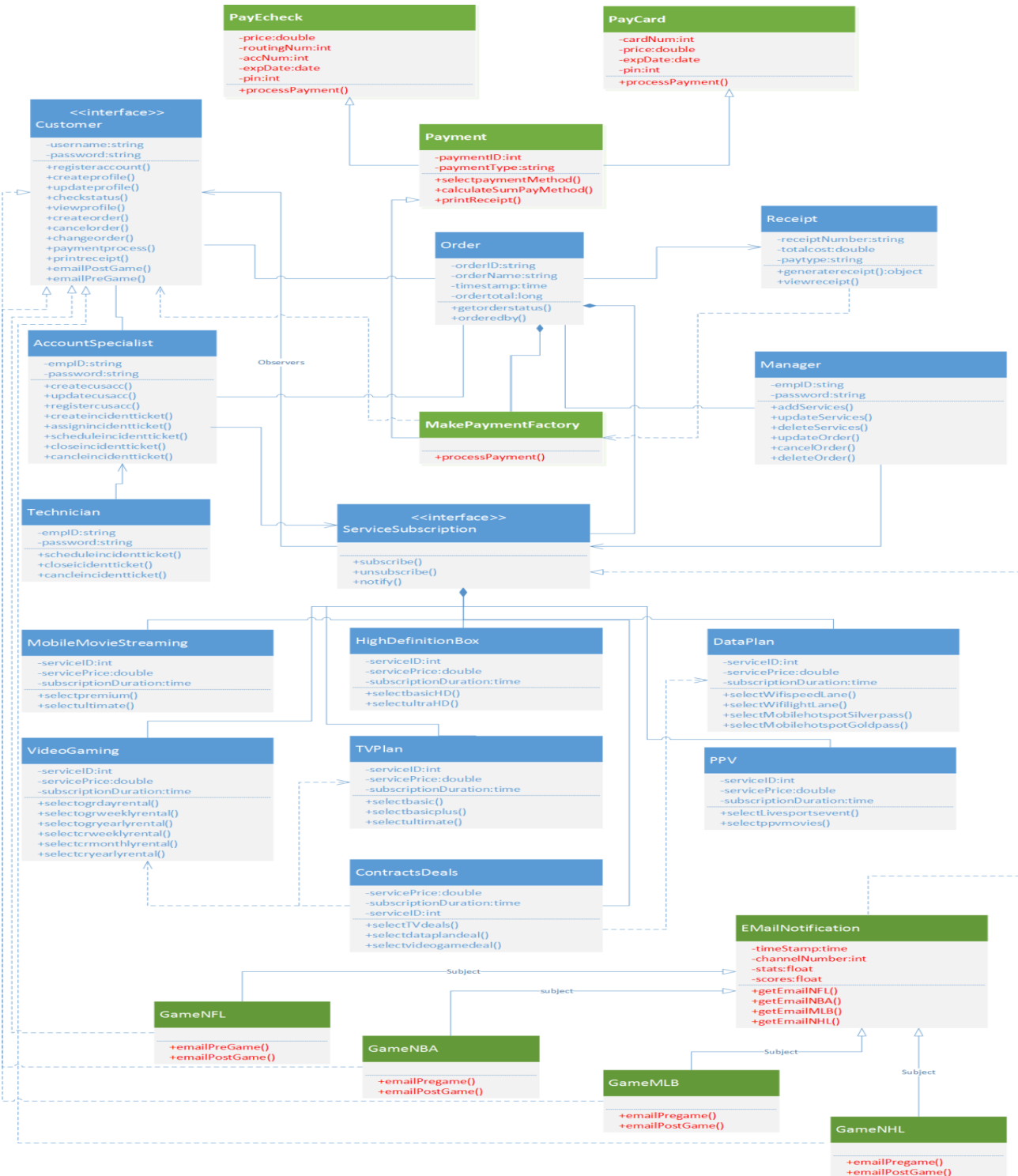
Deliverable 1.....	3
Complete List of Classes in the design.....	3
Deliverable 2.....	4
UML Design Model/ Class Diagram.....	4
Deliverable 3.....	5
List of Design Patterns Used.....	5
Deliverable 4.....	5
Documentation of Design Pattern used in the Design Class Diagram.....	5
Deliverable 5.....	7
Design Model Class Diagram using (Shift+PrtScr).....	7

## **Deliverable 1**

### **Complete List of Classes in the design**

- 1) Customer
- 2) AccountSpecialist
- 3) Technician
- 4) Manager
- 5) Order
- 6) MakePaymentFactory
- 7) Payment
- 8) PayEcheck
- 9) PayCard
- 10) Receipt
- 11) ServiceSubscription
- 12) MobileMovieStreaming
- 13) HighDefinitionBox
- 14) DataPlan
- 15) VideoGaming
- 16) TVPlan
- 17) PPV
- 18) ContractDeals
- 19) EmailNotification
- 20) GameNFL
- 21) GameNBA
- 22) GameMLB
- 23) GameNHL

## UML Design Model/ Class Diagram



## Deliverable 3

### List of Design Patterns Used

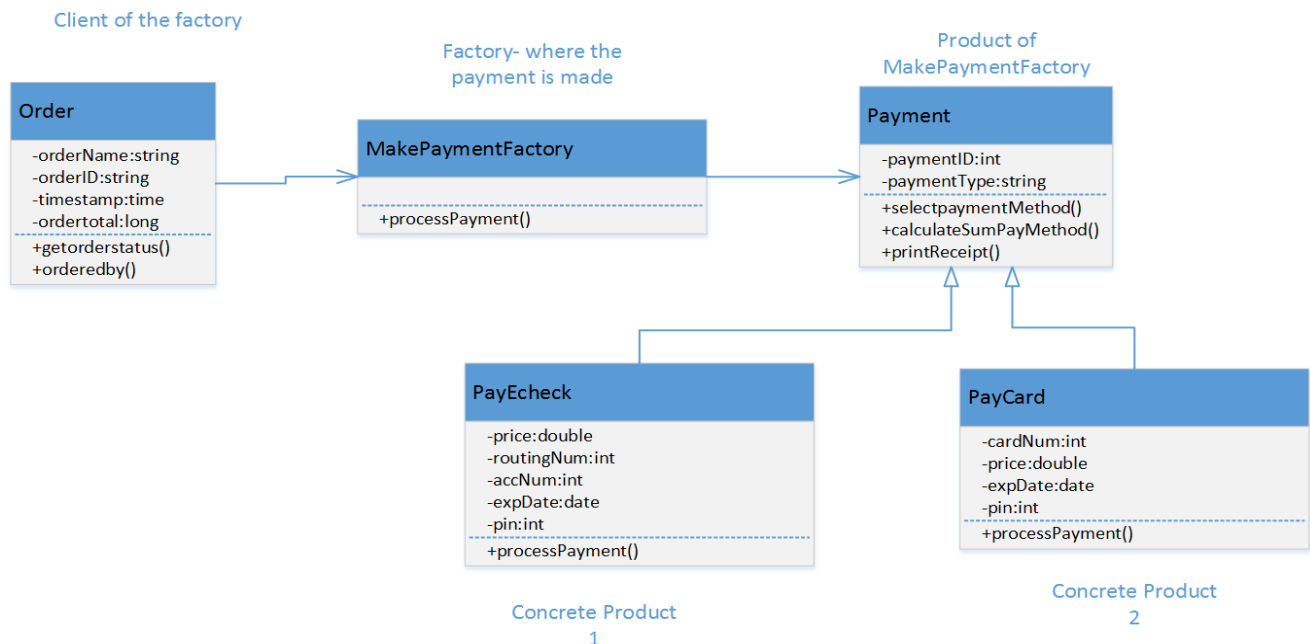
Factory Method Design Pattern

Observer Design Pattern

## Deliverable 4

### Documentation of Design Pattern used in the Design Class Diagram.

#### Factory Method Design Pattern- Implemented at Payment Process



#### Class Order

This is the client of the factory. Order now goes through the *MakePaymentFactory* Class to get the instance of Payment

#### Class MakePaymentFactory

This class is declared with the *processPayment* method statically. This is the factory where the customer process payment. This is the only class referring to the concrete class *Payment*

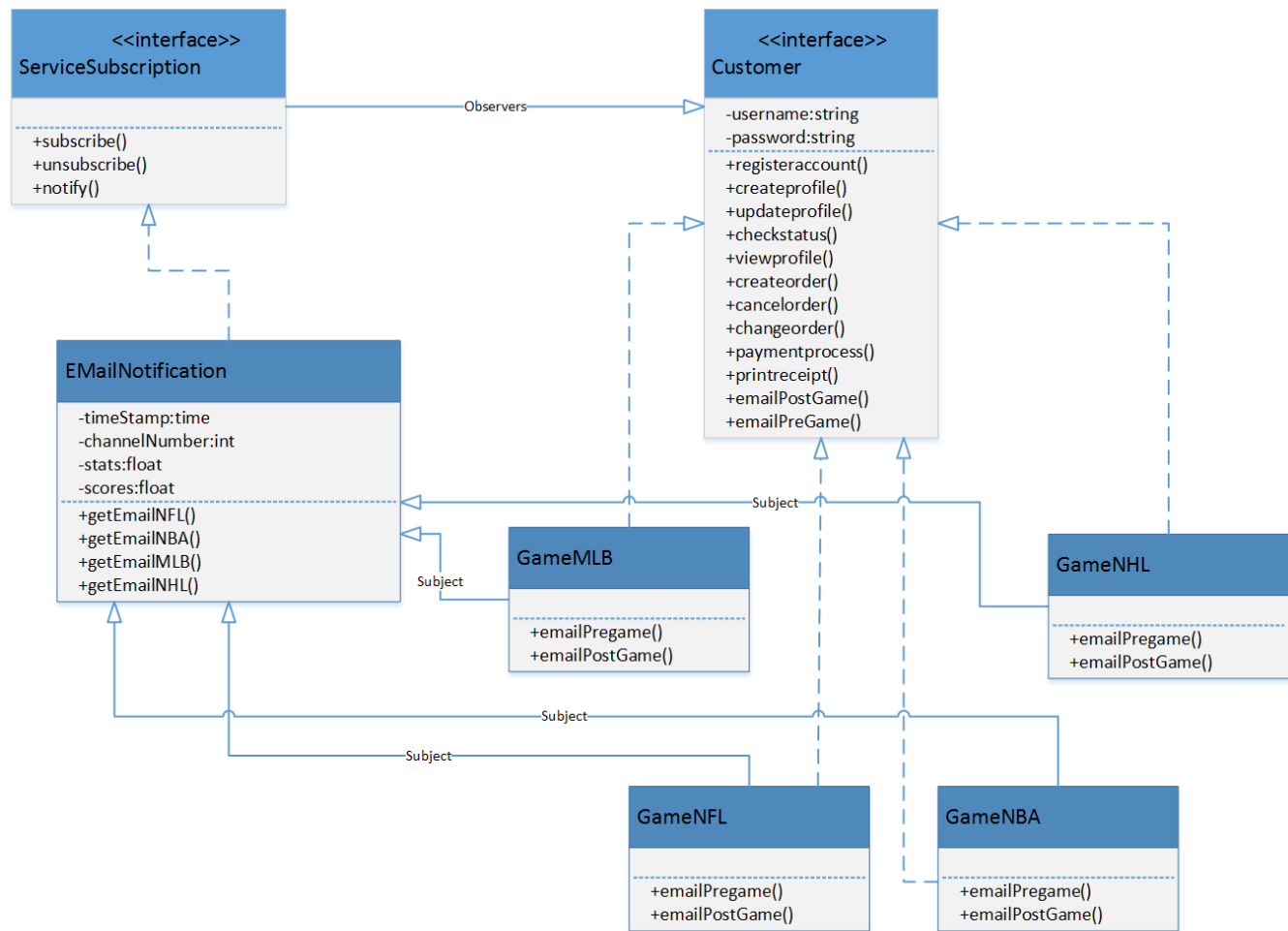
#### Class Payment

This is the product of the factory *MakePaymentFactory*. Here the Payment is defined as an abstract class

#### Classes PayEcheck & PayCard

These are the concrete products. Each Product needs to extend the abstract class *Payment* and be concrete. In this way it can be created by the factory and will be provided to the client when ever needed.

## Observer Design Pattern- Implemented at Email Notification



### Class ServiceSubscription

This is the subject interface in which the observers will subscribe or unsubscribe to get the notification of various games. This is common for all the other services provided by the vendor and if the Email notification is subscribed the customer will get the Email pre and post games at free of cost where the checkout process will have \$0 in the cart.

### Class Customer

The Customer is the observer, where he can subscribe or unsubscribe for an Email notification.

### Class EmailNotification

This class implements the subject interface *ServiceSubscription*.

### Classes GameMLB, GameNHL, GameNFL and GameNBA

These are the sub classes of the super class *EmailNotification*.

## Deliverable 5

### Design Model Class Diagram using (Shift+PrtScr)

