PROGRAMMING TECHNIQUE 2

SECJ1023-04

PROBLEM ANALYSIS AND DESIGN

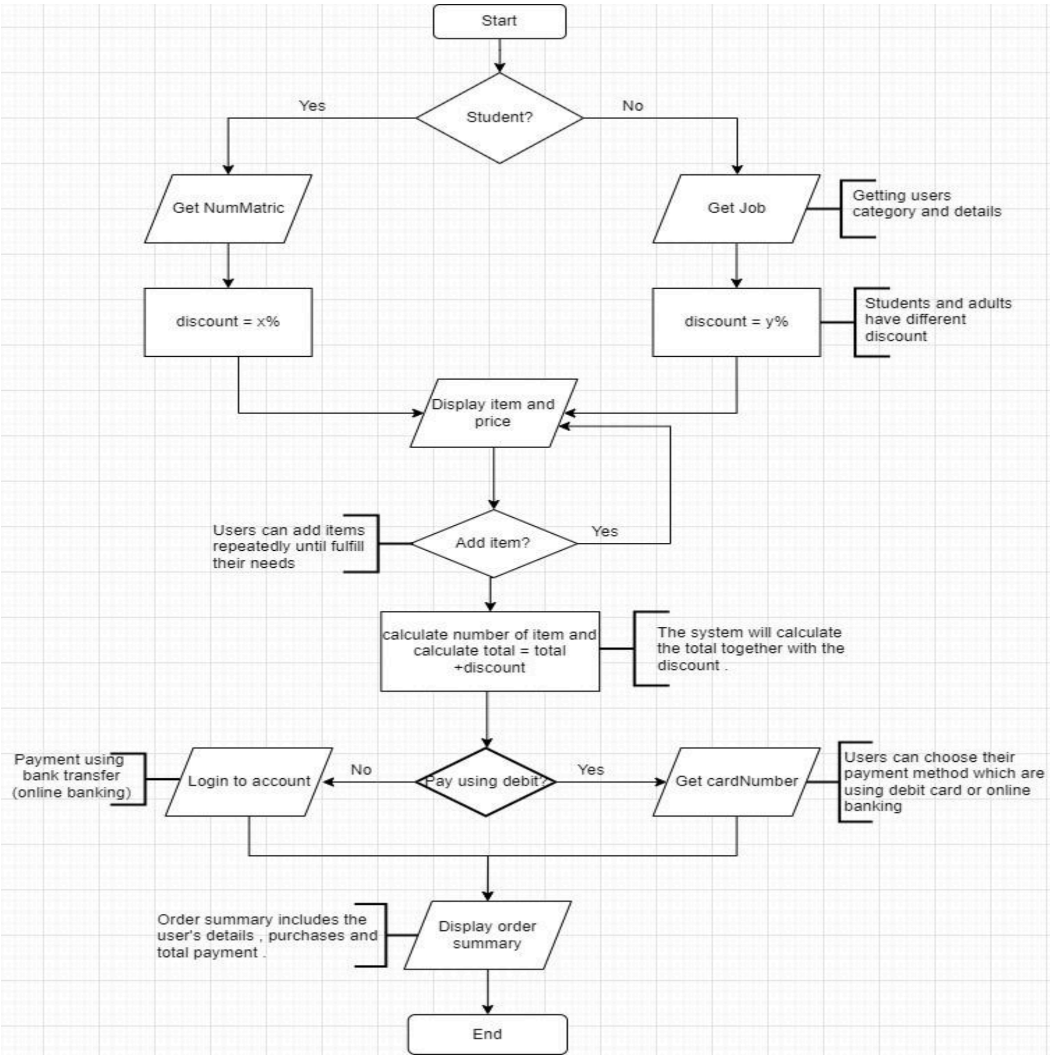
Lecturer:

Ms. Lizawati binti Mi Yusuf

Group Members:

No	Name	Matric No
1	MUHAMAD AMSYAR BIN IBRAHIM	A21EC0058
2	MUHAMAD FAIZ BIN ABDUL MUTALIB	A21EC0059
3	NORAIN BINTI MOHD SULAIMAN	A21EC0106

Section A : Flow Chart


Start
Student?
Yes
No
Get NumMatric
Get Job
Getting users category and details
discount = x%
discount = y%
Students and adults have different discount
Display item and price
Users can add items repeatedly until fulfill their needs
Add item?
Yes
calculate number of item and calculate total = total +discount
The system will calculate the total together with the discount .
Payment using bank transfer (online banking)
Login to account
No
Pay using debit?
Yes
Get cardNumber
Users can choose their payment method which are using debit card or online banking
Order summary includes the user's details , purchases and total payment .
Display order summary
End

First and foremost , the user has to insert his personal details which are his category whether a student or an adult . If he is a student , he has to enter his matric number , else(adult) he needs to enter his current job .Next, there will be a display that display current item that is available in store for purchase.Then, after user finish adding the item in their e-cart, the system will calculate the total price for their purchase. If the user has completed their purchase, the system will display an initial detail of the item purchased by the user to proceed with their next process which is payment. Users can decide whether to use a debit card or bank transfer to make their payment. After the payment is done, users will see their order details generated by the system. Finally, users can exit the system after all the process is done.

Section B : Problem Analysis

Classes :

- **Customer** - string contact , string email , Customer (string , string , string)
- **Student** - string NumMatric , Student (string , string , string)
- **Adult** - string Job, Job (string , string , string)
- **Name** - string name , void setName , string getName
- **Item** - float Price , string ItemName , int Quantity
- **Payment** - float totalAmount , bool orderStatus , virtual getPayment()
- **Credit@Debit** - string CardNumber , string expiredDate , int CVV
- **BankTransfer** - string BankID , string BankName

Classes that involve an association relationship are between class Customer and class Item. This is because class Customer will have one to many relationships with class Item which means that class Customer can access many class Item . In a simple way , it allows a customer to buy more than one item from the system .

On top of that , our system also implements a composition relationship which is between class Name and class Customer . Class Customer is the whole object which contains the part (class Name) objects . They are both dependent on each other as if the class Customer is destroyed , the class Name also will be destroyed .

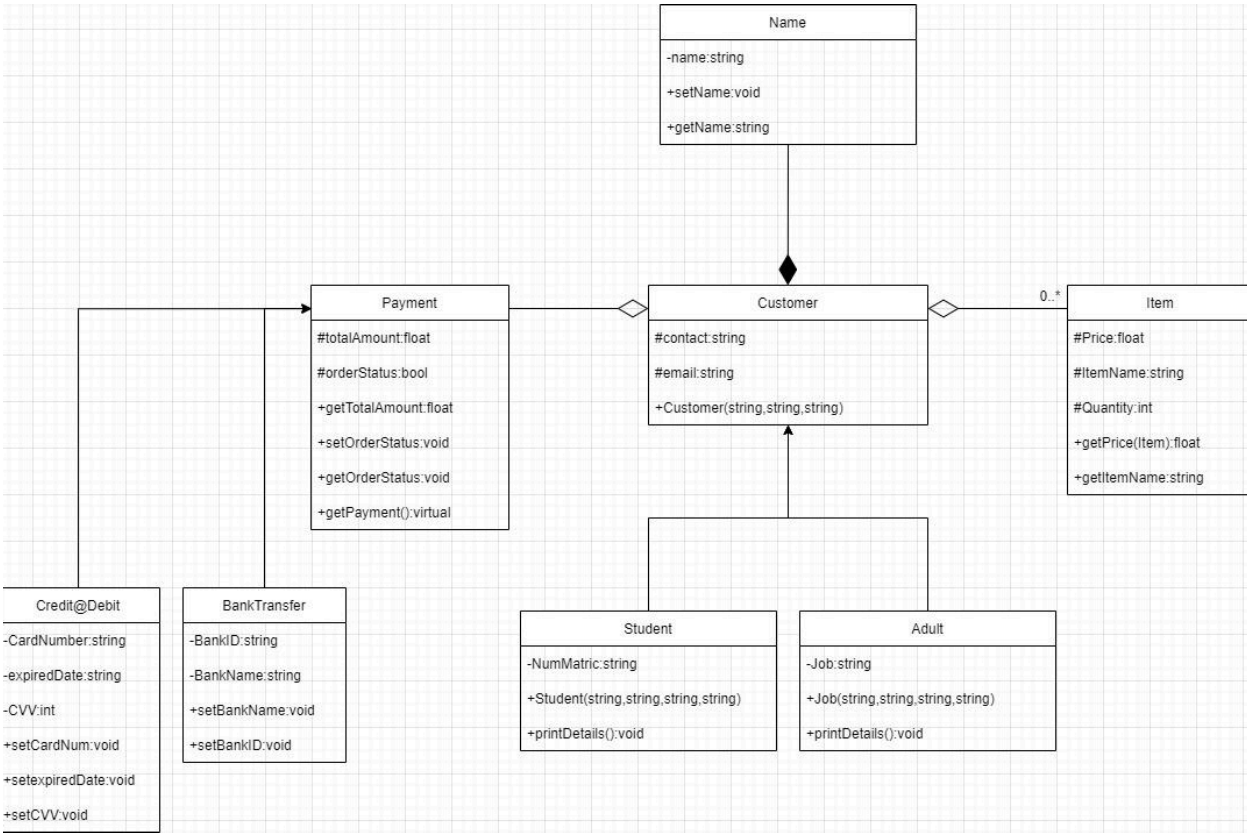
Aggregation relationships are also significantly implemented in our system which is between class Customer and class Payment . The class Customer contains pointers to the class Payment and both of the classes are independent of each other as if the class Payment is destroyed , the class Customer can still exist and vice versa .

Classes that involve inheritance relationships are between class Customer, class Student and class Adult. Class Customer will be the base class while class Student and class Adult will be the derived class. This is because the common attributes for every customer is they have contact information such as email address and phone number.That is why class Customer will be the base class.While for both class Student and class Adult, they have specialized attributes that are only specialized for them to use. For instance, class Student will have a member named numMatric that indicates matric number for a student in university while adults do not have it. And for adults, they have specialized members called job because students do not have job. This is why class Student and class Adult will be the derived class for class Customer.

Other than that, class Payment, class BankTransfer and class Credit@Debit also involve inheritance relationships. Class Payment will be the derived class while class Credit@Debit and class BankTransfer will be the derived class. This is because class Payment has a member that is common between those two derived classes which is totalAmount. While for class Credit@Debit, it has a member named CardNumber that is only specialized for that class and non other classes like class BankTransfer can have that member. While class BankTransfer has members named BankID and BankName that is only for class BankTransfer. This shows that class Payment, class BankTransfer and class Credit@Debit involves inheritance relationship.

Section C : Class Diagram

UML CLASS DIAGRAM



Name
-name:string
+setName:void
+getName:string
Payment
#totalAmount:float
#orderStatus:bool
+getTotalAmount:float
+setOrderStatus:void
+getOrderStatus:void
+getPayment():virtual
Customer
#contact:string
#email:string
+Customer(string,string,string)
0..*
Item
#Price:float
#ItemName:string
#Quantity:int
+getPrice(Item):float
+getItemName:string
Credit@Debit
-CardNumber:string
-expiredDate:string
-CVV:int
+setCardNum:void
+setexpiredDate:void
+setCVV:void
BankTransfer
-BankID:string
-BankName:string
+setBankName:void
+setBankID:void
Student
-NumMatric:string
+Student(string,string,string,string)
+printDetails():void
Adult
-Job:string
+Job(string,string,string,string)
+printDetails():void