

International Study Centre

*Report Title

by

*Author's Name: Rupesh Kumar Gupta

For: Iyalla john Alamina

Date: 24\08\2020

Course: Software design

Details

Assessment Criteria

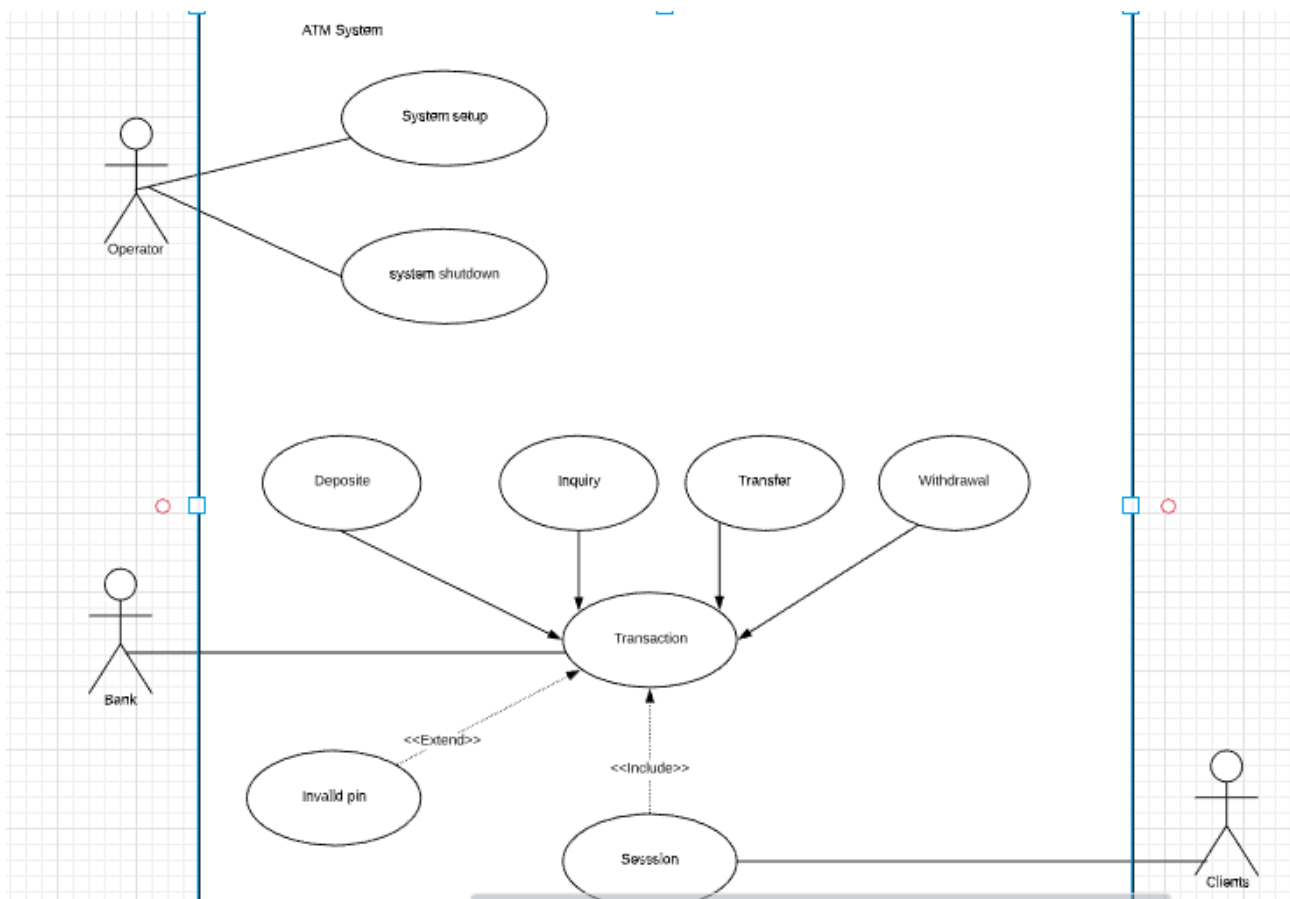
The student will be required to choose one out of four system specifications or any system of interest and must be discussed with tutor. To meet the marking criteria for the course work all systems must have extensive well written documentation of the following elements:

1. Use of object oriented analysis, design and implementation of classes obtained
2. Use of use case analysis to determine system requirements
3. Use of activity diagrams to model system behaviour and specific system outcomes.
4. Use of sequence diagram to demonstrate implementation system component interaction
5. Use of a class relationship diagram that reflects outputs of the use case, activity and sequence diagram
6. Use of wire frame models and/or package diagrams
7. A text user interface should be implemented for the system using the curses library
8. The use of low-level algorithms, programs, functions, pseudocode and flowcharts.
9. Unit tests should be written for a minimum of 5 functions within the program.
10. Working functional and appropriately commented program that demonstrates a direct implementation of the system design.

3. Automated Teller Machine (ATM) simulation

Given 3 trials, the user is able to see his balance by entering a four-digit pin that must NEVER be displayed on screen but masked by the Asterix(*) character. A list of pins stored on the file system must be loaded to verify the pin. The user should be able to withdraw funds below asset limit and cannot redraw more than 3 times within a space of minutes. The balance from each pin should be determined each time the program is run and is a random value between 0 and 100. The user should also be allowed to change his pin at any time which should be reflected in the file system.

Use case diagram of ATM machine:-

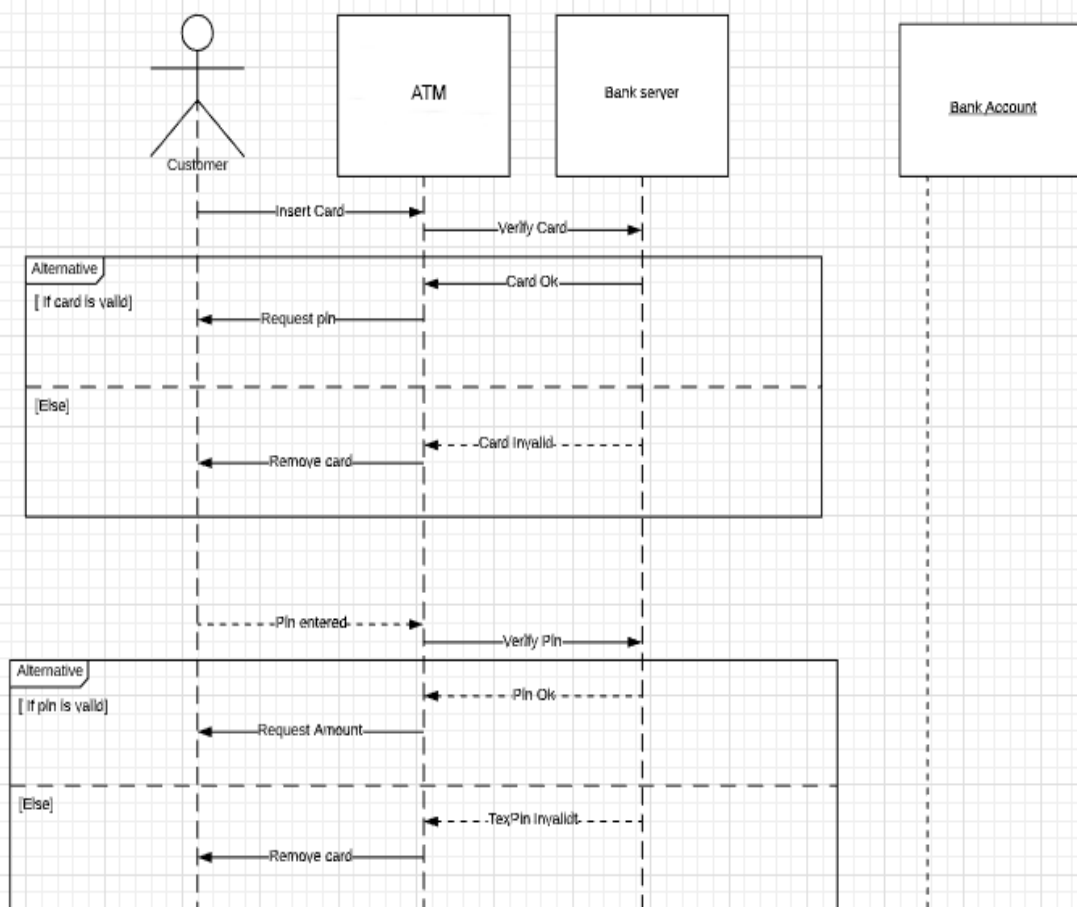


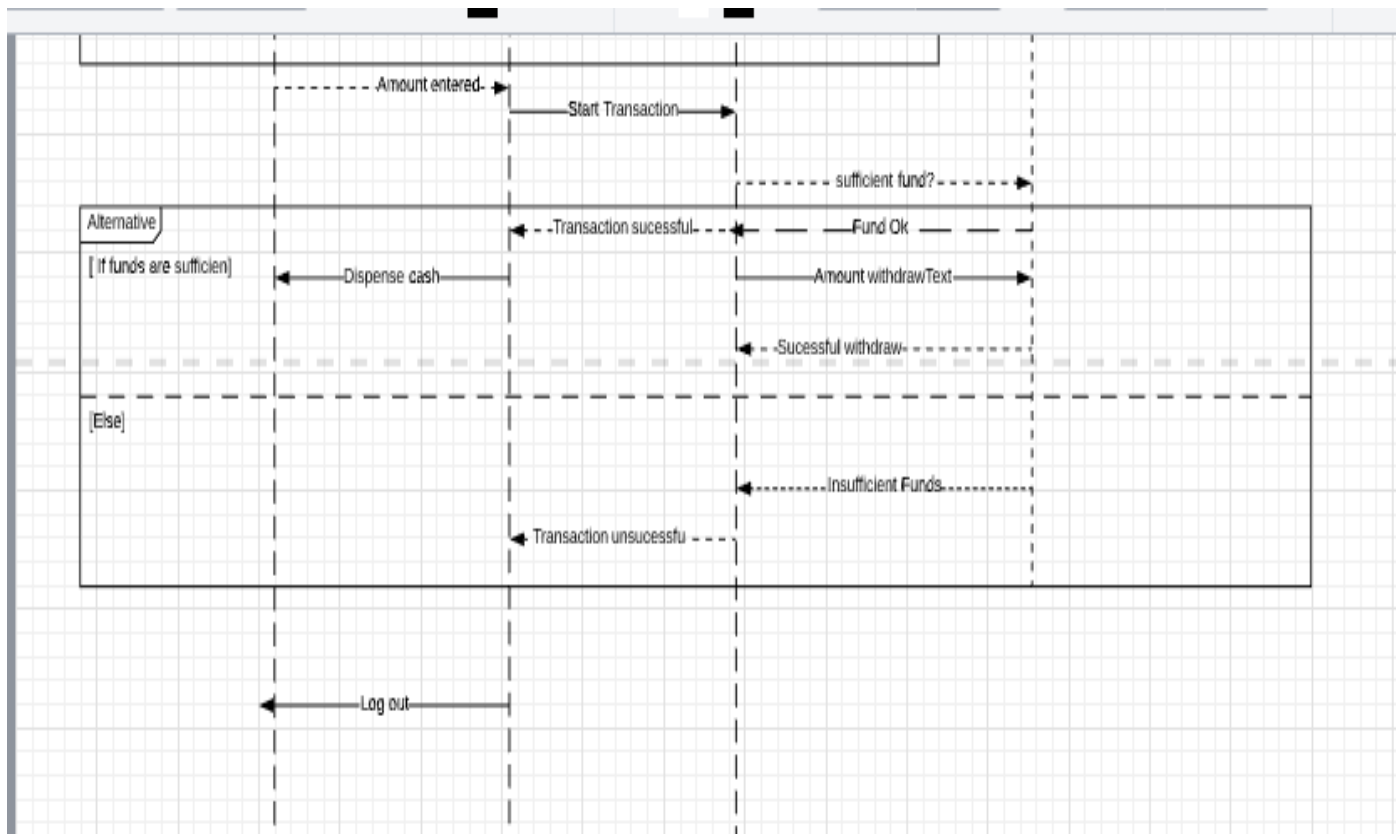
Description:-

In this given ATM framework, there three entertainer bank, customers and usable. In employable, it provides order to arrangement and shutdown being used case for provide guidance to activity for client. What's more, bank save the exchange like deposit, withdrawal, move and request. On the off chance that client utilize substantial pin number it gives right data and gives result positive. On the off chance that utilization off-base or invalid number it doesn't give any subtleties. And furthermore it keeps all the record of banking exchange. In any case, customers right off the bat check their detail by utilizing meeting use case in << include>> into exchange for keeping their records. In an equivalent manner, when customer attempt to check balance they should utilize substantial pin number through << extend>>. Which gives right data to client.

Sequence

diagram



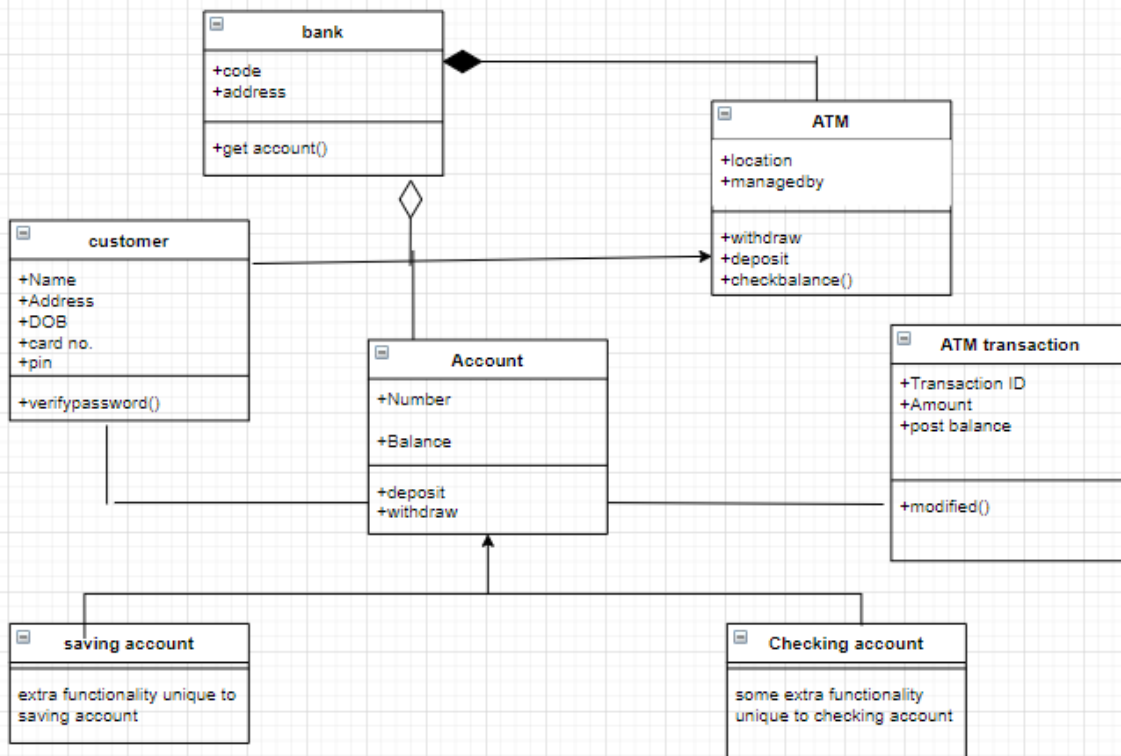


Description:-

In given sequence diagram of ATM machine, firstly customer request insert card to ATM and bank server verify card. If card is verified then it takes request pin number ATM. When pin number is correct then bank system approved the card ok. After that suggest to remove the card. And use the pin entered and again it verified that pin. If pin number is right then take request amount. And suggest remove the card. In a same way, If funds are sufficient ATM machine dispense cash and bank server shows transaction successful then bank account authorized fund ok and give amount withdrawal and successful withdrawal. In case user use incorrect pin number it gives wrong transaction unsuccessful and no give any information. And at final when user take right detail and bank server suggest for logout the ATM system.

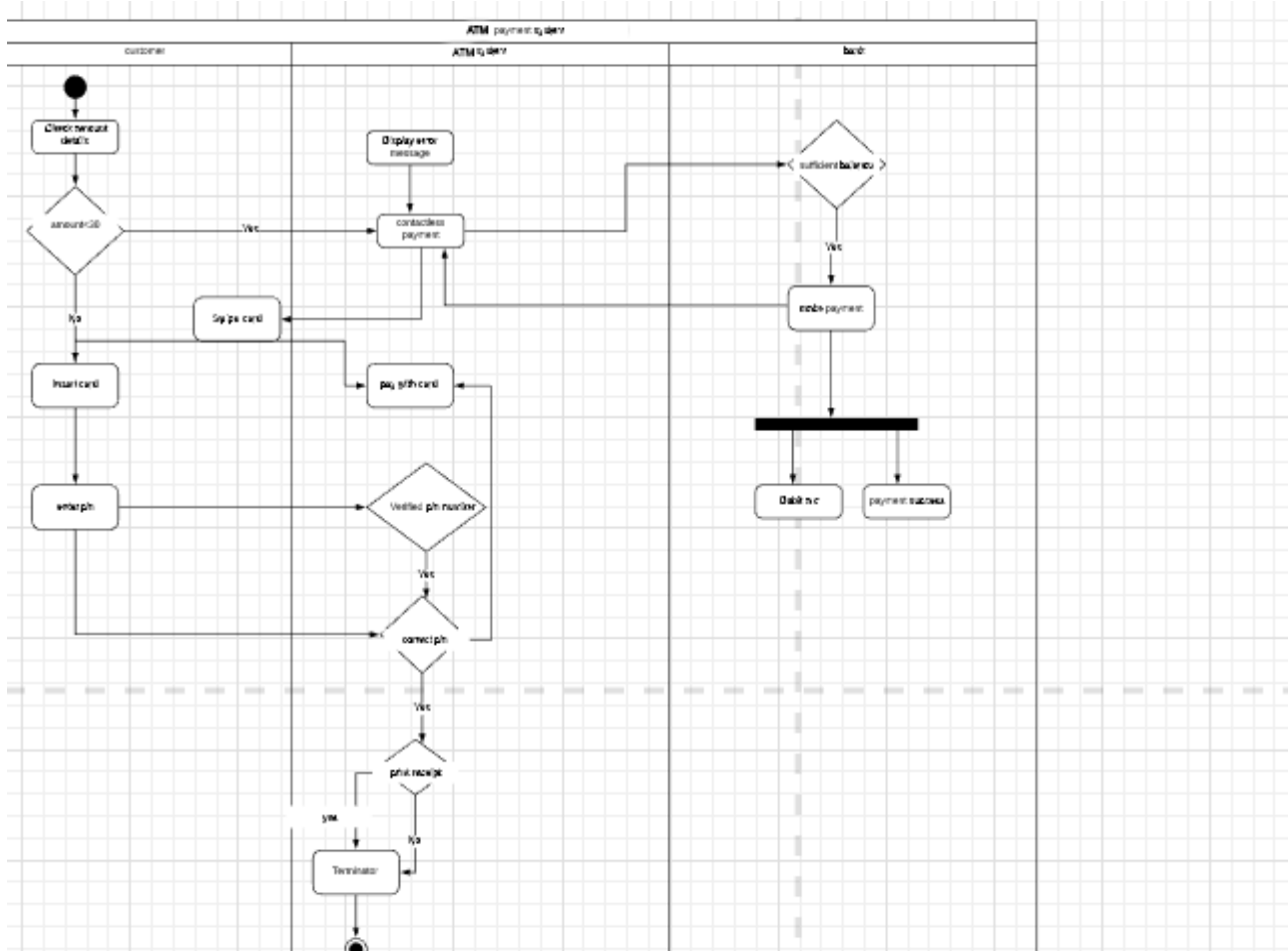
Class diagram of ATM system (Machine):-

UML class diagram of ATM system



In given class diagram of ATM machine there are five super class such as: bank, ATM, Customer, Account and ATM transaction. In bank super class it includes two attributes and one method which all are in positive signs. And in ATM class it has two attributes and three methods which includes all in positives. In customer class it carried name, address, DOB, card number and pin as an attributes and verified password as a methods. ATM transaction class include transaction ID, Amount and post balance in attributes and in methods it keeps only one modified class. As well as account super class it divided into two sub classes like saving account and checking account In saving account it has functionally unique to saving and checking account it functionally unique to checking account.

Activity diagram of ATM system (machine):-



C++ programming coding for ATM

```

#include<iostream>
using namespace std;
int main()
{

int pin;
for (int i=0;i<3;i++)
{cout <<"Enter Pin\n";
cin>>pin;

if (pin==9988)
{cout<<"Right Password!!\n";

double balance = 1000;
double withdraw, deposit;

int n;

```

```

cout<<"\n";
cout<<" Greetings BY HALIFAX BANK\n";

cout<<"Select option :\n";
cout<<"\n";
cout<<"1. Balance enquiry\n"
<<"2) Cash Withdraw \n"
<<"3) Cash Deposit \n"
<<"4) End \n"
<<"\n"
<<"Enter Option:";
cin>>n;
switch(n)
{
case 1:
cout<<"\n Check Amount\n";
cout.setf(ios::fixed);
cout.setf(ios::showpoint);
cout.precision(2);
cout<<"\n Your current amount is $"<<balance<<endl;
break;
case 2:
cout<<"\n Cash Withdrawl\n";
cout<<"Enter the Amount: $";
cin>>withdraw;
balance = balance - withdraw;
cout.setf(ios::fixed);
cout.setf(ios::showpoint);
cout.precision(2);
cout<<"Withdrawl amount is $"<<withdraw<<endl;
cout<<"Remainig amount is $"<<balance<<endl;
continue;
case 3:
cout<<"\n DEPOSIT\n";
cout<<"PLease Enter the amount: $";
cin>>deposit;
balance = balance + deposit;
cout.setf(ios::fixed);
cout.setf(ios::showpoint);
cout.precision(2);
cout<<"You have deposited $"<<deposit<<endl;
cout<<"Your new balance is $"<<balance<<endl;
continue;
case 4:
cout<<"\n EXIT \n";
break;

```

```
default:
cout<<"\n Wrong input \n";
}
break;
}
else
cout<<"Wrong Password\nPlease try again\n";}

return 0;
}
```