**CREDIT CARD PROCESSING SYSTEM**

**Ex: No:**

**Date:**

**AIM:**

To Design, Implement and Test the Credit Card Processing System described in the problem statement .

**PROBLEM STATEMENT:**

The given problem statement is about the processing of credit card. The Credit card processing system which is use to purchase an item for any shop mall. And it is used to maintain the limitation of credit card balance and current transaction process could be update via credit card machine.

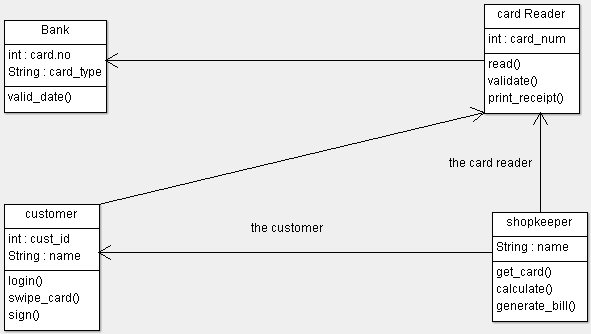
The customer should be select the item to be purchased from using credit card payment. Then the vendor should give a bill for the purchased item. The customer should give his card to swap and request a kind of transaction. After processing the transaction, the CREDIT CARD MACHINE should give the balance print statement or it can also be as a receipt. The credit card processing machine described sequentially through steps.

* Customer should select the item form the shop.
* Vendors make the bill for the selected item.
* Customer gives the credit card to vendor for swapping the card.
* The Required amount transaction is done by the card reader.
* Vendor will issue the balance statement to the respected customer.
* Customer put the signature in the receipt and return to the vendor.

**USECASE DIAGRAM:**

****

**CLASS DIAGRAM:**

****

**SEQUENCE DIAGRAM:**

****

**ACTIVITY DIAGRAM:**

****

**STATE CHART DIAGRAM:**



**COMPONENT DIAGRAM:**

****

**DEPLOYMENT DIAGRAM:**



**IMPLEMENTATION & TESTING**

**Banker.java**

package credit\_card;

/\*\*\* \* @author kavin \*/

public class Banker {

int bid;

String b\_name;

String b\_branch;

int limit;

int credit(int sal)

{

int li=(sal\*30)/100;

limit=li;

return limit;

}

int verify(int pur)

{

if(limit>pur)

{

limit=limit-pur;

return pur;

}

else

return -1;

}

}

**Retailer.java:**

package credit\_card;

/\*\*

\*

\* @author kavin

\*/

public class Retailer extends Banker {

int balence;

int go;

void validate(int pur,int l)

{

limit=l;go=l;

if(verify(pur)!=-1)

{

System.out.println("Transaction Successfull");

System.out.println("Updated Balance:"+limit);

balence=limit;

}

else

{

System.out.println("Insufficient Balance");

}

}

int update()

{

return balence;

}

}

**Credit\_Card.java**

package credit\_card;

import java.util.Scanner;

/\*\* \* \* @author kavin \*/

public class Credit\_Card extends Banker{

public static String name[]=new String[5];public static int sal[]=new int[5];public static String loc[]=new String[5];public static int limt[]=new int[5];

public static void menu()

{

System.out.println("Choose the below options:");

System.out.println("1.Create");

System.out.println("2.Purchase");

System.out.println("3.Exit");

}

public static void main(String[] args) {

System.out.println(“\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Credit Card Processing System\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”);

Scanner s=new Scanner(System.in);

int a=1;

int card[]=new int[5] ;int i=0;

card[0]=4490; card[1]=7645; card[2]=7455; card[3]=7664;card[4]=7766;

while(a==1)

{

menu();

int ch=s.nextInt();

switch(ch)

{

case 1:

{

System.out.println("Enter Name:");

name[i]=s.next();

System.out.println("Enter Location:");

loc[i]=s.next();

System.out.println("Enter the Salary:");

sal[i]=s.nextInt();

Banker ak=new Banker();

System.out.println("Your Credit card no.:"+card[i]);

limt[i]=ak.credit(sal[i]);

System.out.println("The Limit per Month :Rs."+ limt[i] +" (30% of Salary)");

i++;

break;

}

case 2:

{

System.out.println("Enter Total purchase value:");

int amt=s.nextInt();

System.out.println("Enter the card no.:");

int ca=s.nextInt();int g=0;int q=0;

for(q=0;q<5;q++)

{

if(ca==card[q])

{

g=1;

break;

}

}

if(g==1 )

{

if(amt<limt[q])

{

Retailer ak= new Retailer();

ak.validate(amt,limt[q]);

limt[q]=ak.update();

}

else

System.out.println("Insufficint credit");

}

else

{

System.out.println("Invaild Card Number...");

}

break;

}

case 3:

{

a=44;

System.out.println("ThankYou...");

break;

}

default:

System.out.println("Invalid Option.Try Again...");

}

}

}

}

**Output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Credit Card Processing System\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Choose the below options:

1.Create

2.Purchase

3.Exit

1

Enter Name:

Ram

Enter Location:

Egmore

Enter the Salary:

25000

Your Credit card no.:4490

The Limit per Month :Rs.7500 (30% of Salary)

Choose the below options:

1.Create

2.Purchase

3.Exit

1

Enter Name:

Gokul

Enter Location:

Chennai

Enter the Salary:

40000

Your Credit card no.:7645

The Limit per Month :Rs.12000 (30% of Salary)

Choose the below options:

1.Create

2.Purchase

3.Exit

2

Enter Total purchase value:

2390

Enter the card no.:

4490

Transaction Successfull

Updated Balance:5110

Choose the below options:

1.Create

2.Purchase

3.Exit

2

Enter Total purchase value:

4583

Enter the card no.:

7645

Transaction Successfull

Updated Balance:7417

Choose the below options:

1.Create

2.Purchase

3.Exit

3

ThankYou...

**TEST REPORT 1**

**Product : Credit Card Processing System**

**Use Case :credit card creating**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case / Action To Perform** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1. 1 | After Entering the Name, Location,Salary. | Displays the credit limit for customer per month | The card no. and credit limit is stored. | Pass |

**TEST REPORT 2**

**Product : Credit Card Processing System**

**Use Case :credit card swiping**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case / Action To Perform** | **Expected Result** | **Actual Result** | **Pass/Fail** |
|  | Swiping card at store after entering the purchase | Validates if the credit limit is not exceeded. | The Purchase is made Successful. | Pass |
|  | After purchasing the items | The credit card balance is printed. | The balance of the Credit Card is updated further Transaction | Pass |