

**Ex.No:**

# **ATM SYSTEMS**

**Date:**

## **1. Problem analysis and project planning**

### **1.1 Introduction**

Banking is one of the common and day to day attribute of life. Nowadays it is totally different from that existed a few years ago banking has become completely computerized new facilities such as credit cards, debit cards & ATM has been introduced. ATM is automatic teller machine which is basically used to withdraw money from an account.

### **1.2 Objectives**

The objective of this software is similar to ATM software installed in ATM center. It should first validate the pin in the ATM card. Then the type of transaction is enquired and the information from the customer is validated. If it is a withdrawal the amount is asked. After the money is delivered the transaction just made is updated in the database where the customer's information is stored.

### **1.3 Scope**

The scope of the project is to design an ATM system that will help in completely automatic banking this software is going to be designed for withdrawal and deposit of money and register the transaction in the database where the customer's information is stored.

### **1.4 Problem Statement**

ATM is another type of banking where the most frequently type of transaction made is withdrawal. A user may withdraw as much as many amount as he wants until his account holds a sum greater than his withdrawal amount. ATM is completely automated and there is no necessity of the ATM center being placed at the bank itself. It can be placed in the shopping malls, airports, railway stations etc.

This ATM system can use any kind of interface. But it should be user friendly and not confusing. Help manuals should be provided in case any customer has problem working with the software.

The system will retain information on the entire customer who has necessity rights to access the service. It will contain the balance amount in the account, rate of interest, any special allowance for that customer and most of all pin number of the customer. The ATM system should be compatible with any kind of database such as MS-ACCESS, DB2, ORACLE, SQL, SERVER etc. the emphasis here is on consistency.

Some customer could have availed some special offers on his ATM cards. So this must be taken care of and the appropriate data should be dealt with.

The ATM should provide easy access to the data for the customer. It should also have a highly secure interface so that one can take money one behalf of others. So the security is one of the main aspects in ATM.

## **2. Problem statement(Use case)analysis**

### **2.1 Identified use cases**

#### **i. Login:**

Here the user enters the card and the inputs his password to enter into the main form. If the password is incorrect, the system will display an error message.

#### **ii. Transaction:**

This is the important part of the ATM system, where there are two types of transaction-withdrawal and deposit. While withdrawing the user specifies the amount and may request for the printed output also.

#### **iii. Maintaining Customer Information:**

Here the administrator plays an important role, whose work is to add customer, delete customer account, update customer account, etc.

## 2.2 Identified Actors

### i Administrator:

Administrator plays an important role. He is the system designer. All the updating works is done by him only like adding, deleting customer accounts.

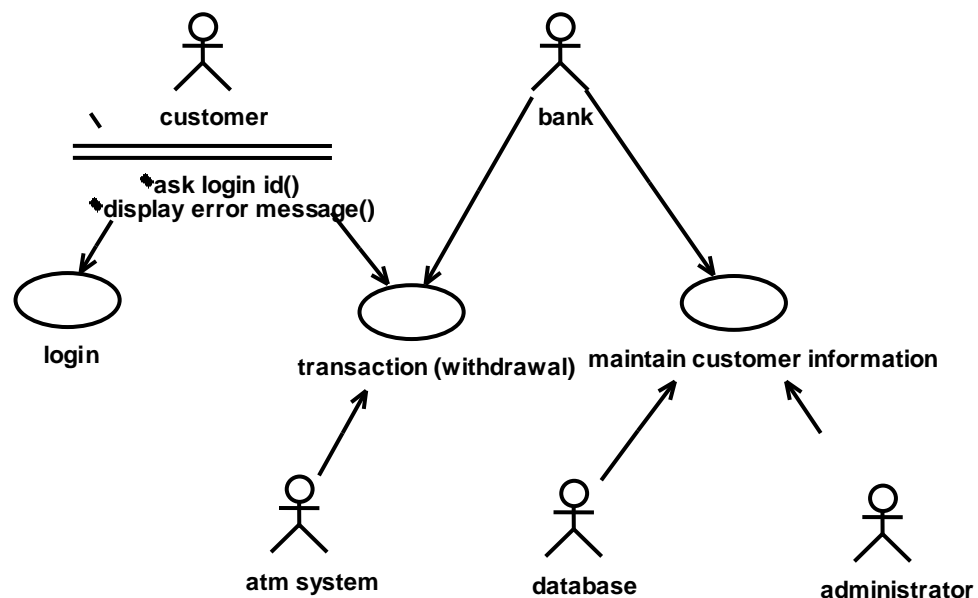
### ii Database:

All the transaction works-withdrawal and deposit are updated in the database.

### iii Customer:

He is the external user the ATM system for taking money and depositing money also.

## 2.3 Use Case Diagram



### **3. Design of ATM system**

#### **3.1 Design Documentation**

##### **1. Login**

###### **1.1 Brief description:**

This use case describes how the user logs into the System.

###### **1.2 Flow of events:**

###### **1.2.1 Basic flow:**

This use case starts with the actor wishes to log in to the ATM System.

1. The system requests the user to enter the name and PIN.
2. The actor enters the name and PIN.
3. The system validates the name and the PIN and logs the user into the system.

###### **1.2.2 Alternative flow:**

1. If the user enters the wrong name and the PIN then the system displays an error message.
2. The actor can either return to the basic flow or cancel login at which point use case ends.

###### **1.3 Pre conditions:**

None

###### **1.4 Post conditions:**

User will perform corresponding transaction.

##### **2. Transaction**

###### **2.1 Brief description:**

This describes the transaction that the user is doing.

## **2.2 Flow of events:**

### **2.2.1 Basic flow:**

This use case starts after the user has logged on to the system.

1. The system requests the user to enter the type of transaction of either withdrawal or deposit and asks for customer information.
2. The actor enters the type of transaction and the customer information.
3. The system displays the corresponding transaction screen.

### **2.2.2 Alternative flow:**

If the customer enters any wrong information then the system displays an error message.

## **2.3 Pre Condition:**

The user logs on to the system.

## **2.4 Post Condition:**

Based on the transaction he gets the transaction screen.

## **3. Maintain Information about Customer**

### **3.1 Brief description:**

This describes how administrator takes care of customer information.

## **3.2 Flow of events:**

### **3.2.1 Basic flow:**

This use case starts after the administrator has logged into the system.

1. The system asks the administrator whether he wants to add or delete customer information.
2. The administrator then enters the type of maintenance.

### **3.2.2 Alternative flow:**

None

### **3.3 Pre Condition:**

The administrator logs on to the system before this use case begin.

### **3.4 Post Condition:**

Administrator gets the corresponding maintenance screen according to his choice.

### **3.2.1.1 Adding Customer**

#### **3.2.1.1.1 Basic flow:**

1. This use case starts when the administrator has chosen to add customer's information.
2. The system asks the administrator to enter customer information.
3. The administrator enters the customer information.
4. The system displays the updated information.

#### **3.2.1.1.2 Alternative flow:**

If the administrator enters any wrong information the system displays an error message.

### **3.2.1.2 Deleting Customer**

#### **3.2.1.2.1 Basic flow:**

1. This use case starts when the administrator has chosen to delete an existing customer from the system.
2. The system asks the administrator to enter the customer information.
3. Administrator enters the corresponding user information.
4. The system then displays updated results.

### **3.2.1.2.2 Alternative flow:**

If the administrator has entered any wrong information then the system displays administrator error message.

### **3.2.1.3 Updating an existing Customer account**

#### **3.2.1.3.1 Basic flow:**

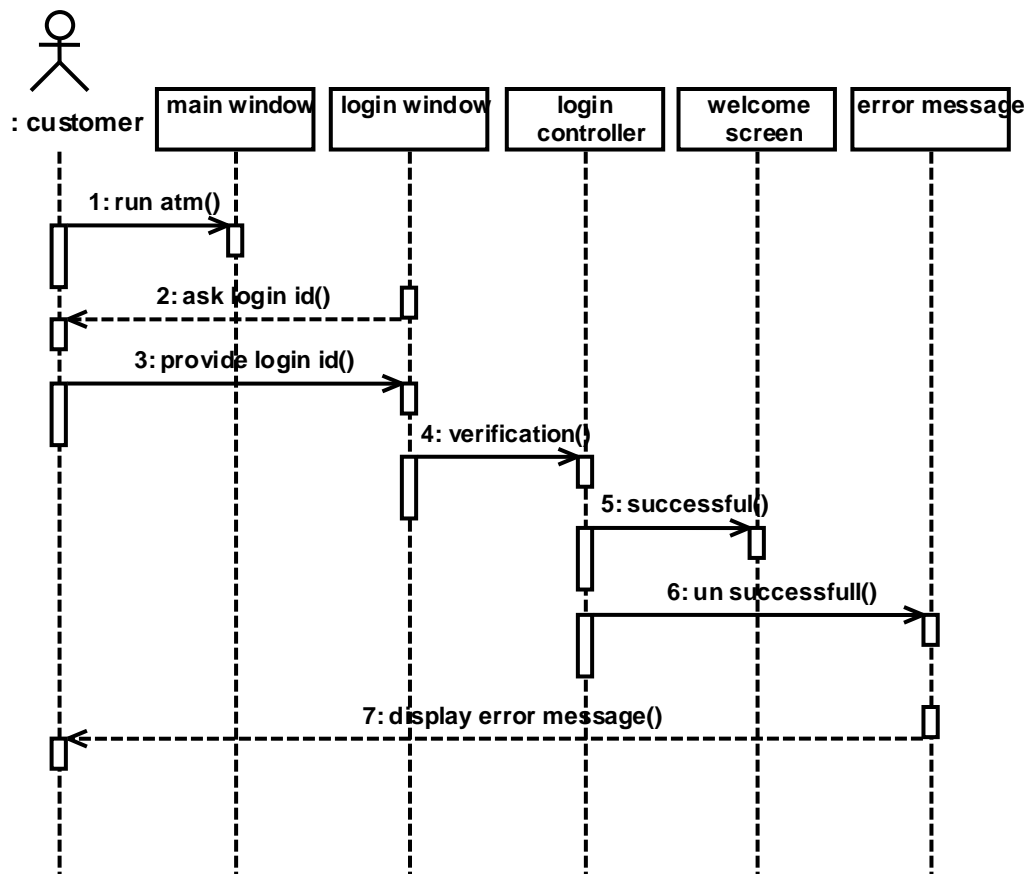
1. This use case starts when the administrator has chosen to update the customer's information.
2. The system asks the administrator to enter the customer information.
3. The administrator enters the customer information.
4. The system displays the updated information.

#### **3.2.1.3.2 Alternative flow:**

If the administrator has entered any wrong information then the system displays administrator error message.

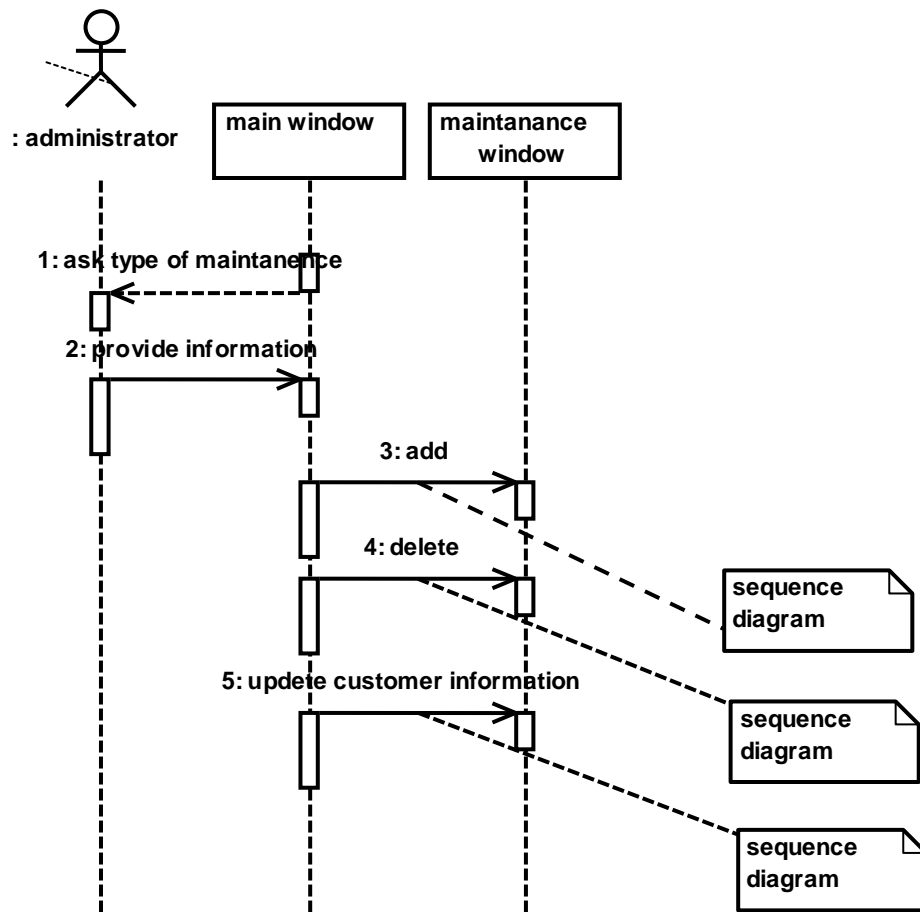
# SEQUENCE DIAGRAM

## 1. Login:

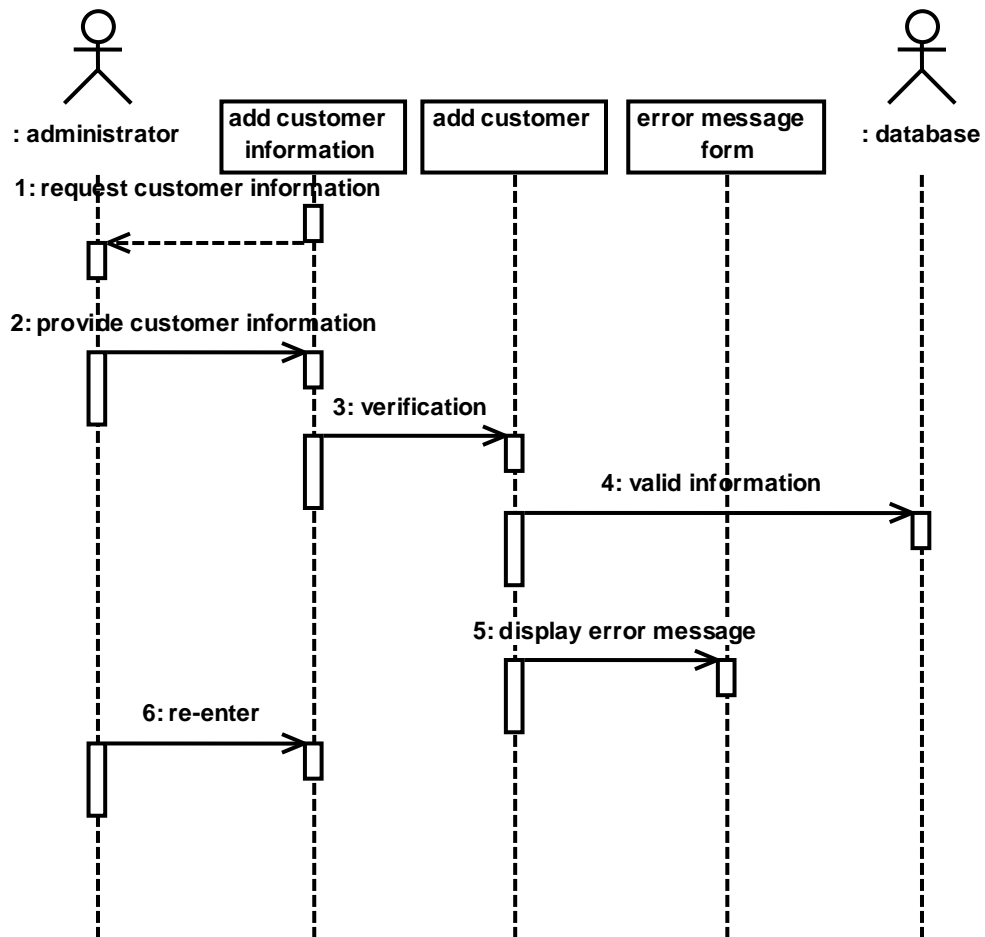




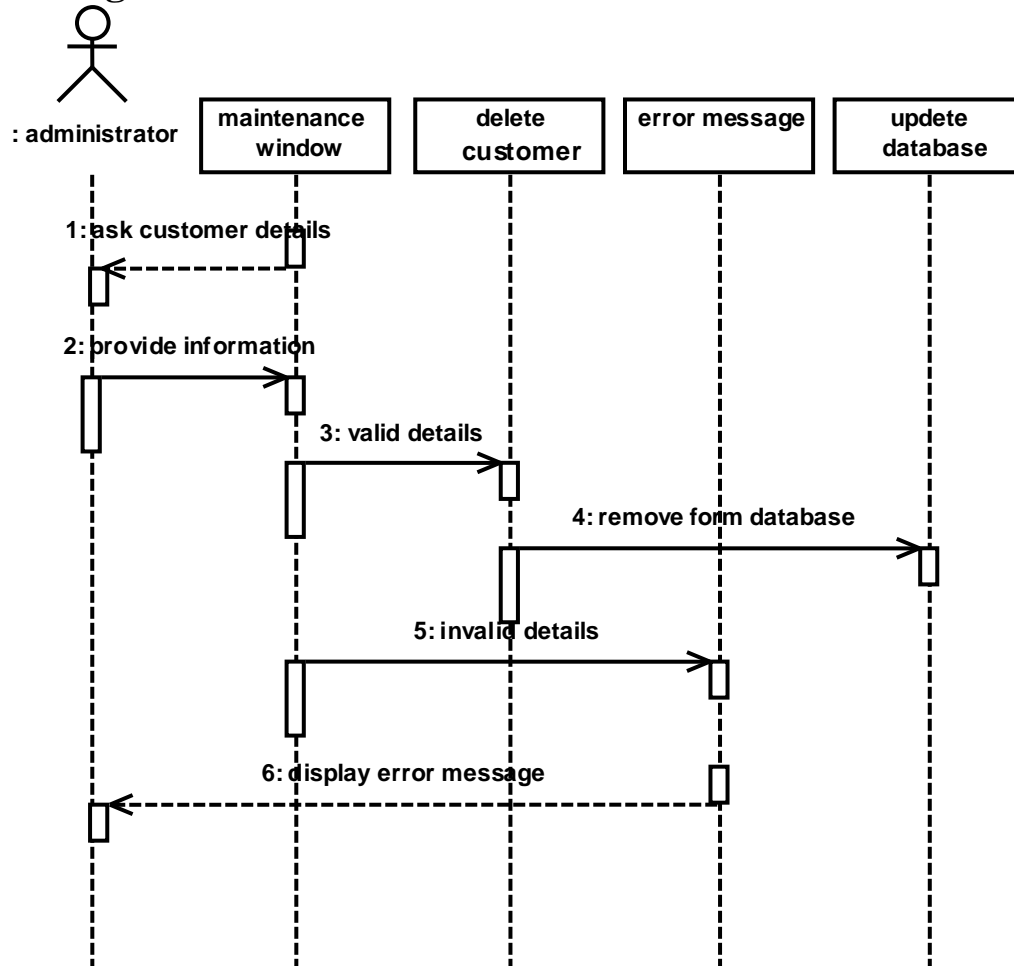
## 2. Maintenance:



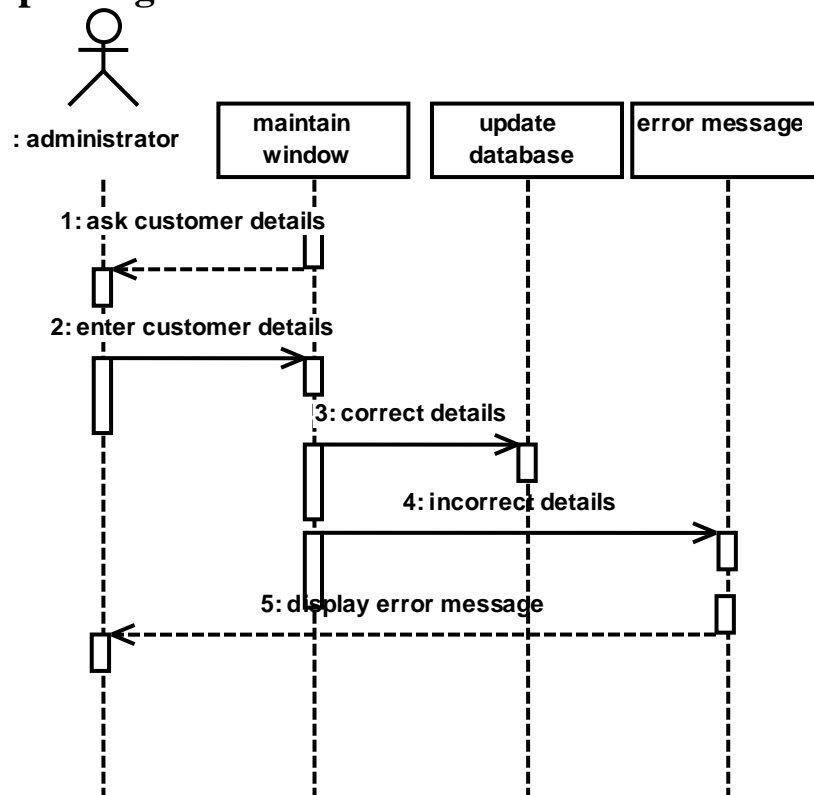
### 3. Adding customer:



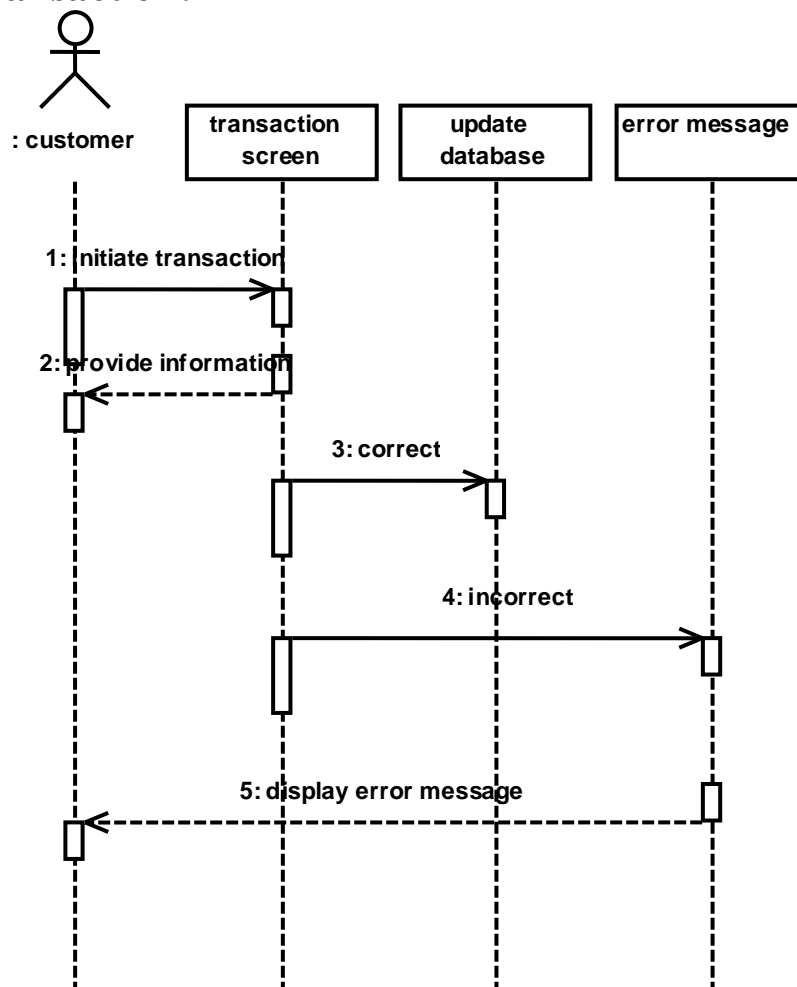
#### 4. Deleting customer:



#### 5. Updating customer:

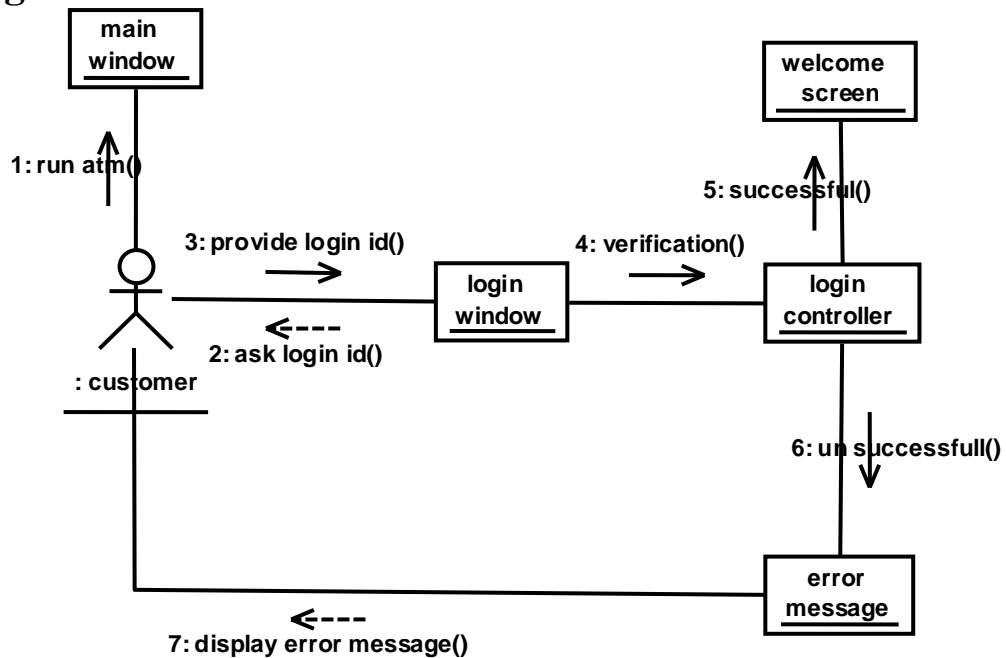


## 6. Transaction:

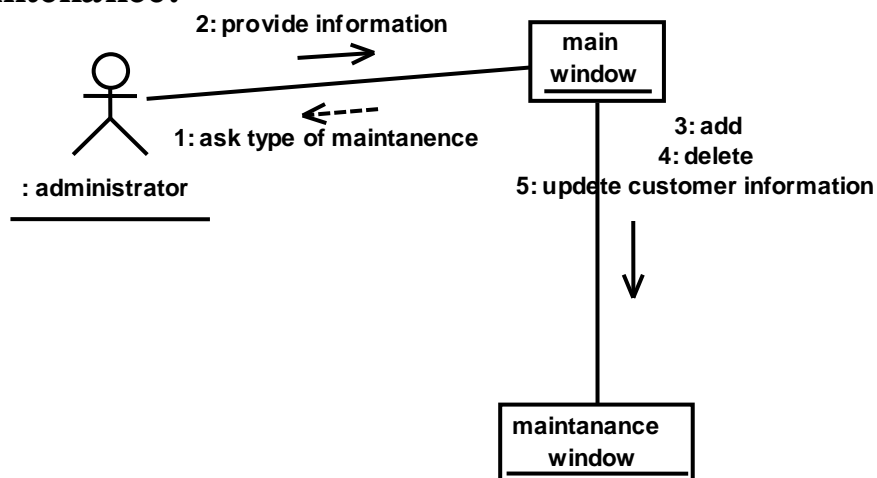


## COLLABORATION DIAGRAM:

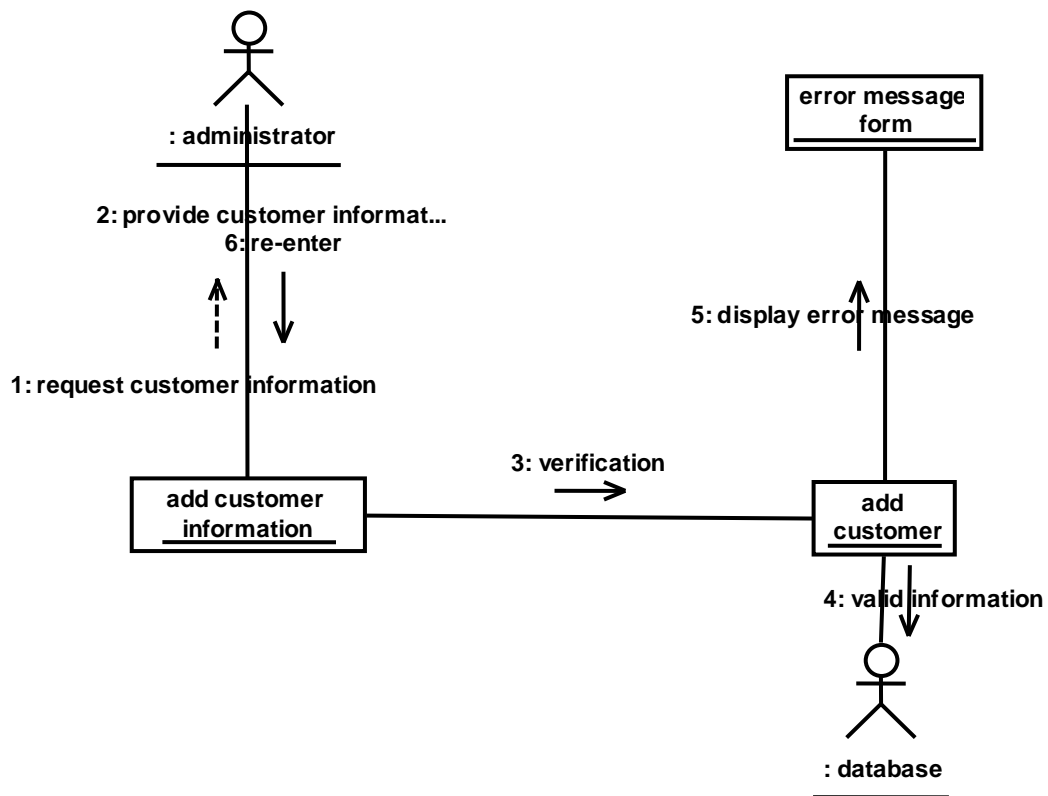
### 1. Login:



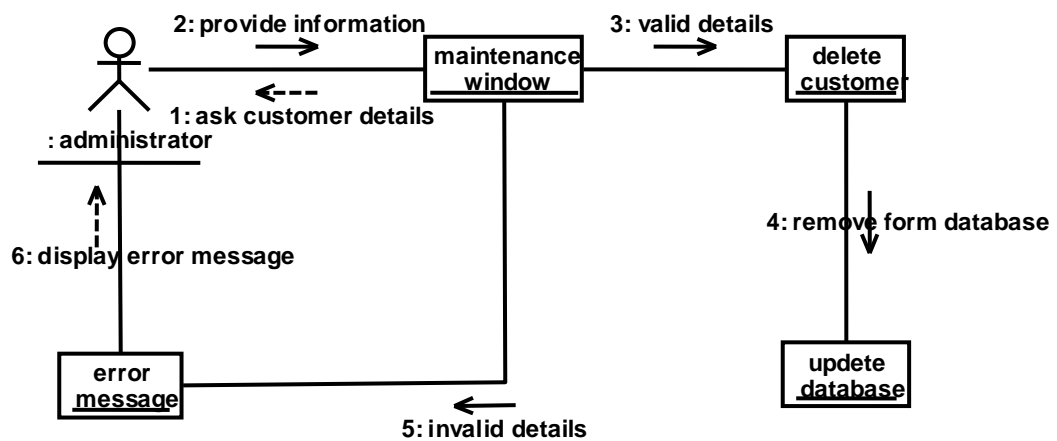
### 2. Maintenance:



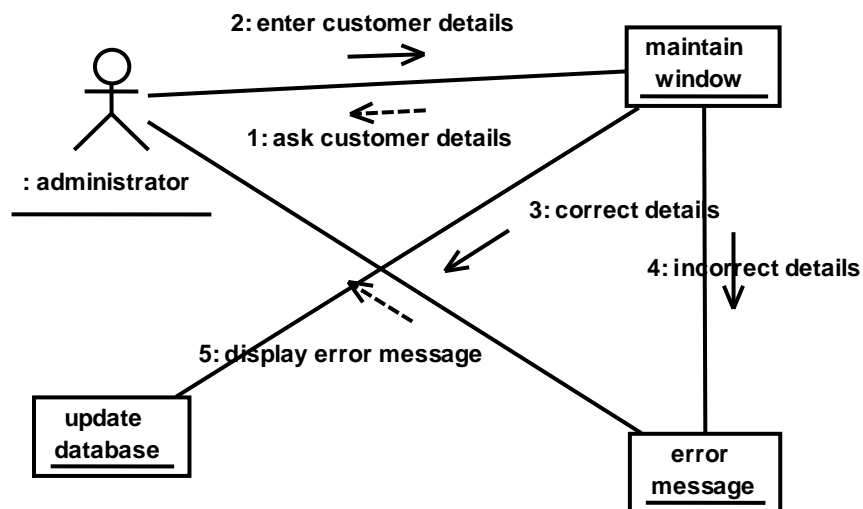
### 3. Adding customer:



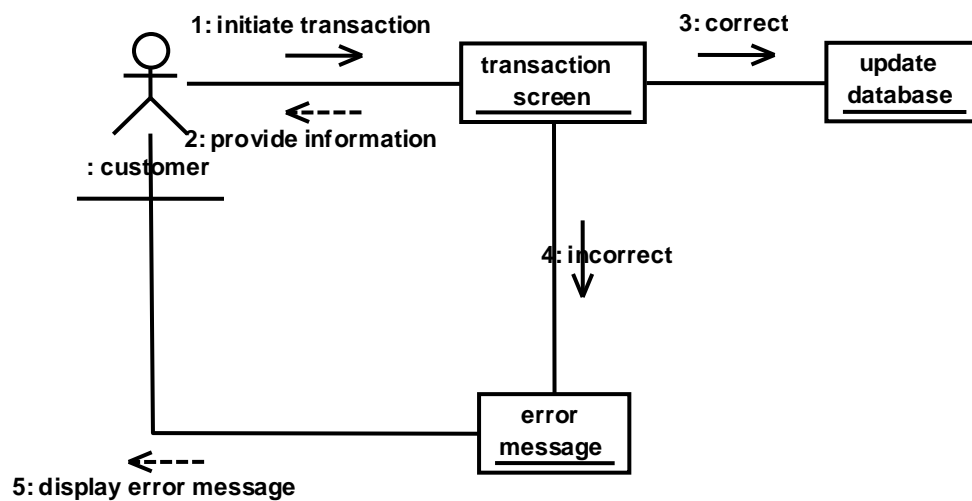
### 4. Deleting customer:



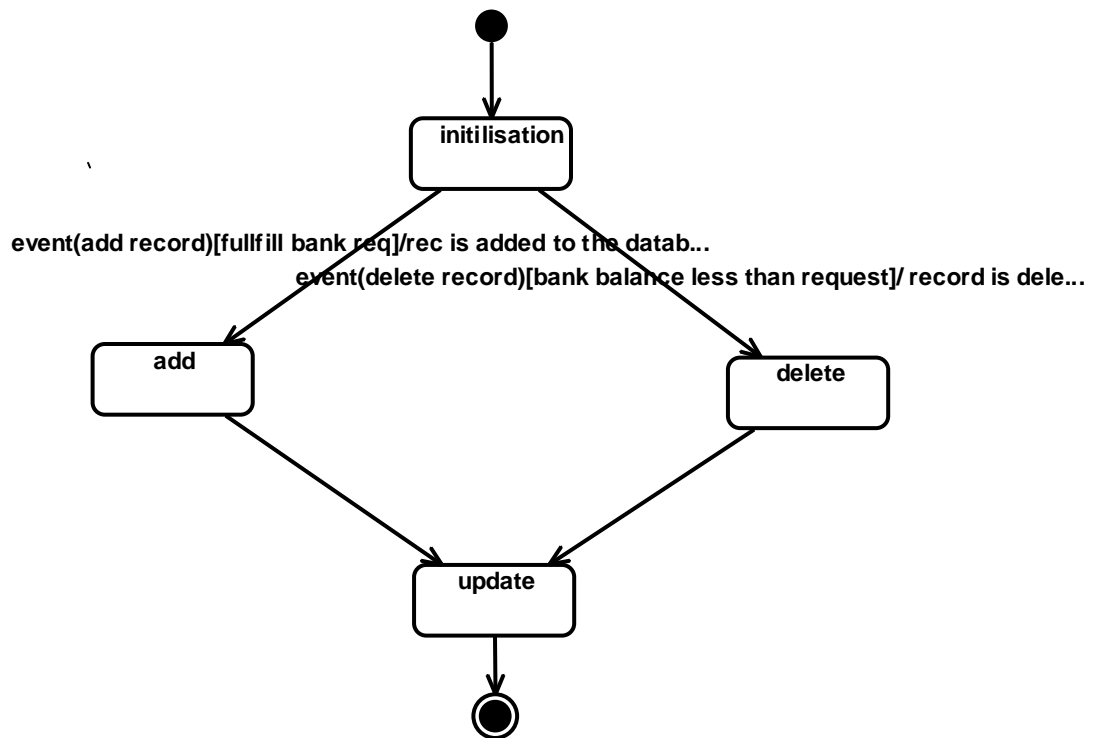
## 5. Updating customer:



## 6. Transaction:



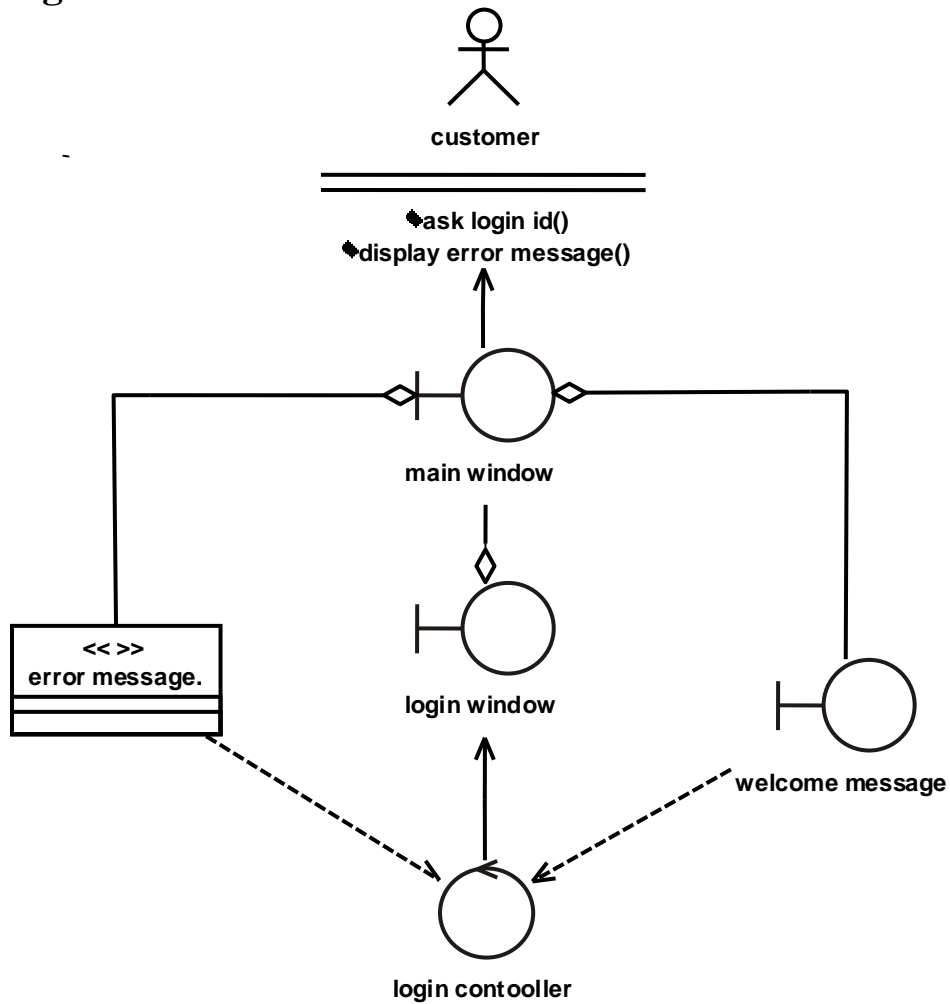
## STATE CHART DIAGRAM



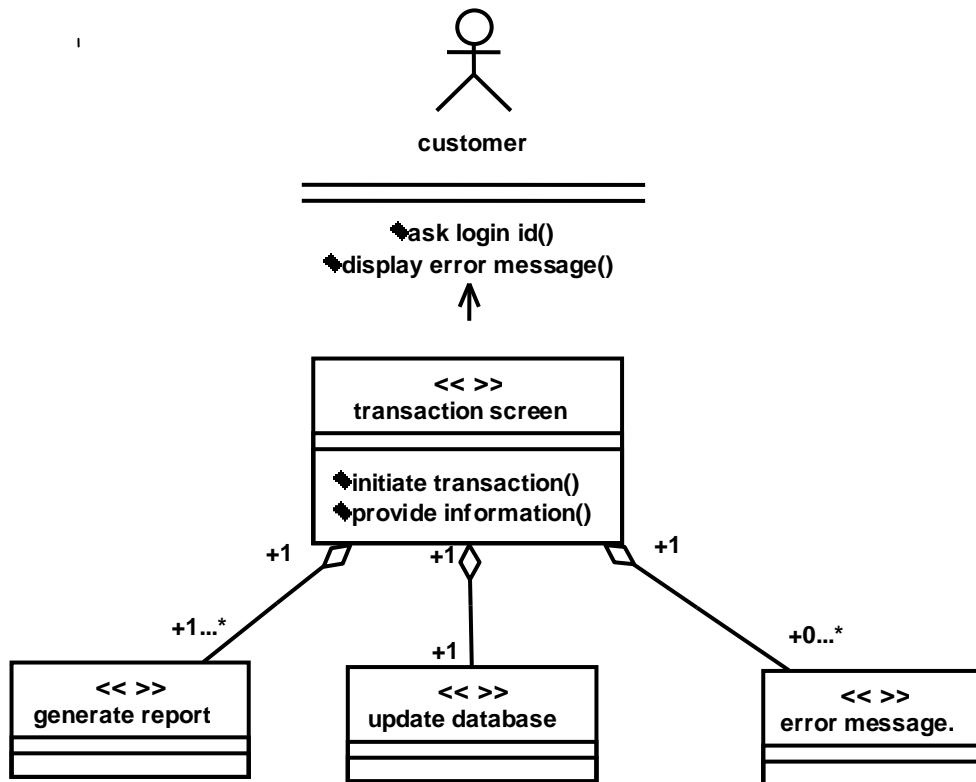


## CLASS DIAGRAM

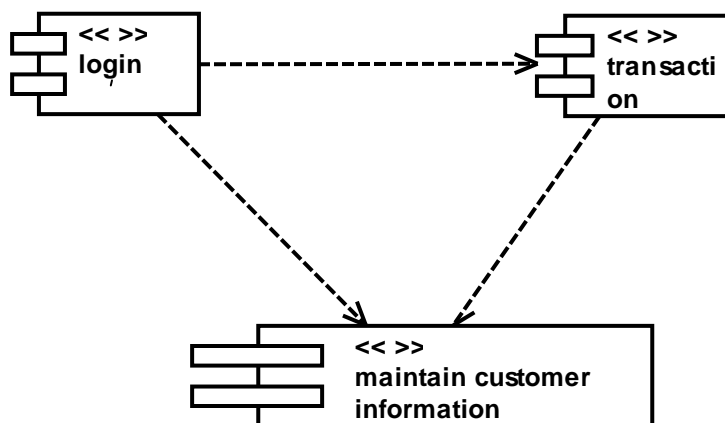
### 1.Login



## 2. Transaction:



## COMPONENT DIAGRAM



## **SOURCE CODE:**

### **1. Login**

Option Explicit

Public NewProperty As login\_window

Public NewProperty2 As welcome\_message

Public NewProperty3 As customer

Public NewProperty4 As error\_message

### **2. Transaction**

Option Explicit

Public As error\_message

Public NewProperty As customer

Public Sub initiate\_transaction()

End Sub

Public Sub provide\_information()

End Sub