

# **SOFTWARE ARCHITECTURE LAB**

## **LAB REPORT**

**PREPARED BY**

**NAME: RUBI SAHU**

**ROLL NO: 218CS3324**

**MTECH 1<sup>ND</sup> SEM**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,  
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA-769008  
ODISHA**

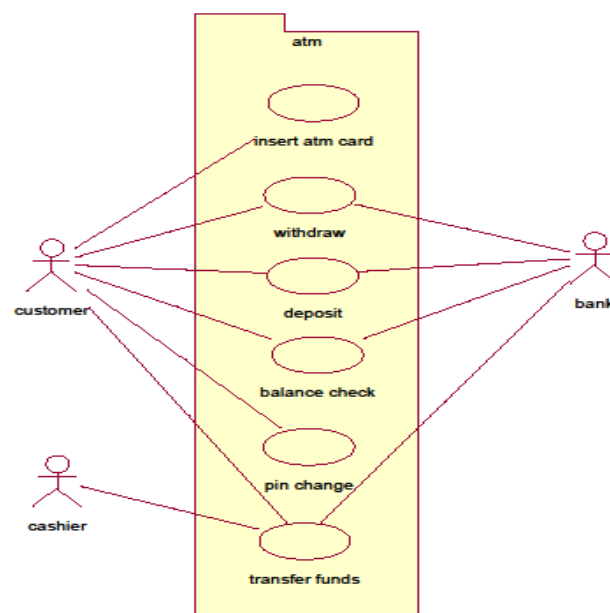
# TABLE OF CONTENTS

| <u>S.NO.</u> | <u>NAME OF THE TOPICS</u>                  | <u>PAGE NO.</u> |
|--------------|--|-----------------|
| <b>1.</b>    | <b>UML DIAGRAMS(RATIONAL ROSE)</b>         | 3               |
|              | 1.1UAE CASE DIAGRAM                        | 3               |
|              | 1.2ACTIVITY DIAGRAM                        | 4-7             |
|              | 1.3 STATE CHART DIAGRAM                    | 8-11            |
|              | 1.4 SEQUENCE DIAGRAM                       | 12-15           |
|              | 1.5 COLLABORATION DIAGRAM                  | 16-19           |
|              | 1.6CLASS DIAGRAM                           | 21              |
|              | 1.7OBJECT DIAGRAM                          | 22              |
|              | 1.8COMPONENT DIAGRAM                       | 23              |
| <b>2</b>     | <b>UML DIAGRAMS(RSA)</b>                   | 23-26           |
| <b>3</b>     | <b>ATM ARCHITECTURE USING ARCH_STUDIO</b>  | 27              |
| <b>4</b>     | <b>ATM ARCHITECTURE USING ACME_STUDIO</b>  | 27              |
| <b>5</b>     | <b>ATM ARCHITECTURE DIAGRAM USING LTSA</b> | 28-29           |
| <b>6</b>     | <b>PETRINET DIAGRAMS</b>                   | 29-31           |
|              | 6.1ATM                                     | 29              |
|              | 6.2PRIME NUMBER                            | 30              |
|              | 6.3PALINDROME OF NUMBER                    | 30              |
|              | 6.4FACTORIAL OF NUMBER                     | 31              |
| <b>7</b>     | <b>ATM Z-SPECIFICATION</b>                 | 31-34           |
| <b>8</b>     | <b>ATM PROJECT(SNAPSHORTS)</b>             | 35-37           |

# 1 UML DIAGRAMS(RATIONAL ROSE)

## 1.1 Use case

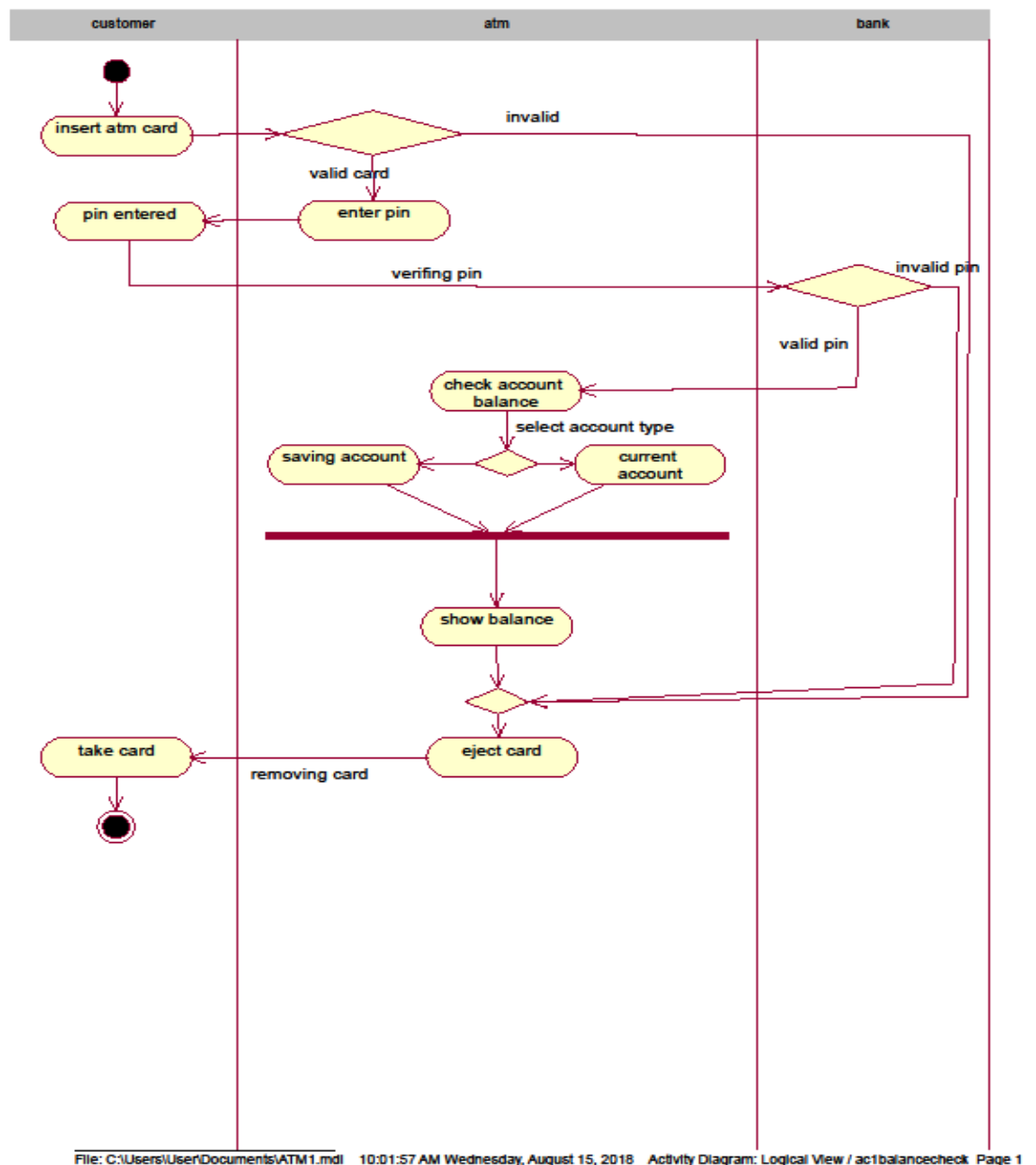
- A **use case diagram** is a graphic depiction of the interactions among the elements of a system.
  - A **use case** is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system to achieve a goal. The actor **can** be a human or other external system.
- Use case diagram for ATM machine



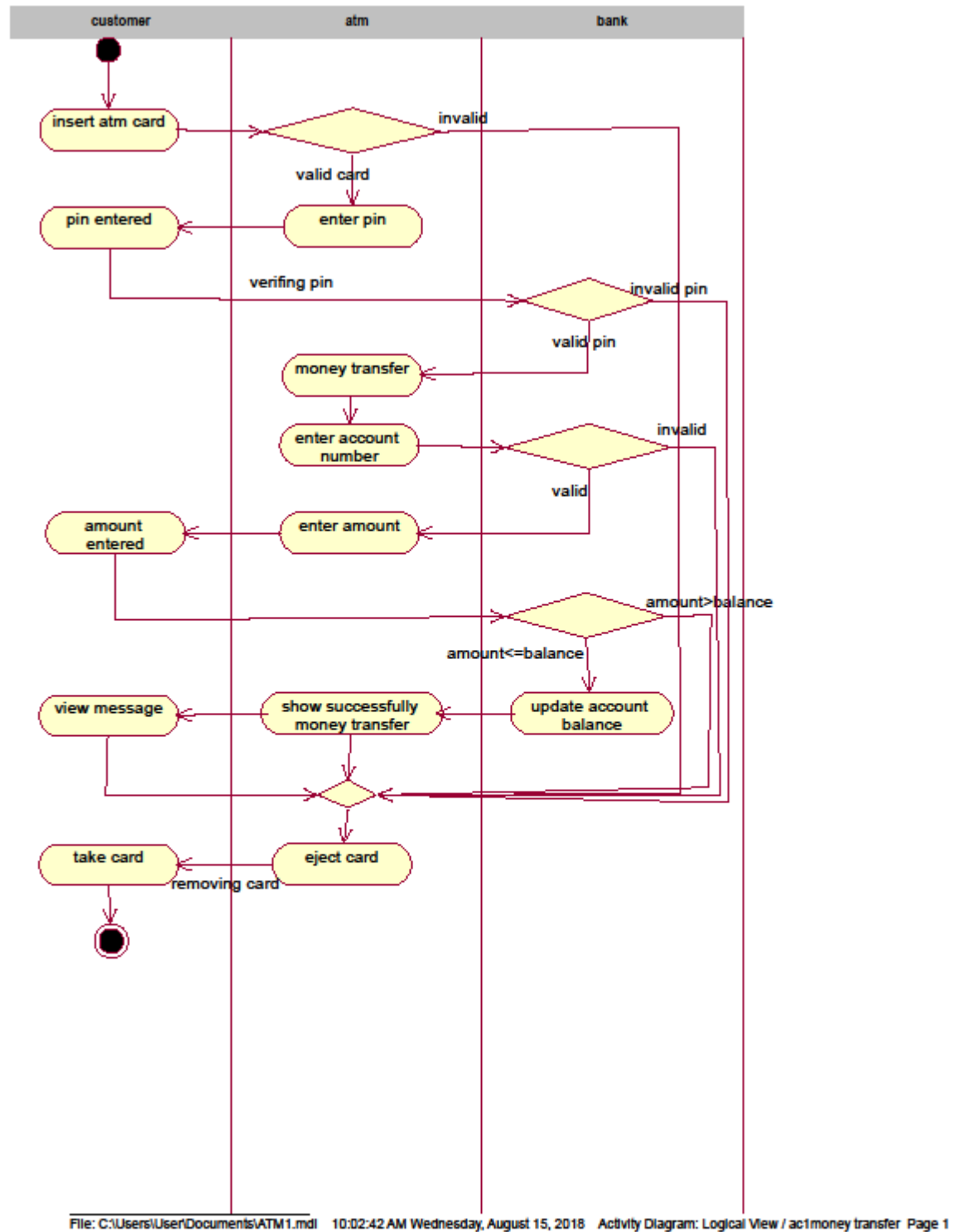
## 1.1 Activity diagram

**Activity diagram** is another important **diagram** in UML to describe the dynamic aspects of the system. **Activity diagram** is basically a flowchart to represent the flow from one **activity** to another **activity**.

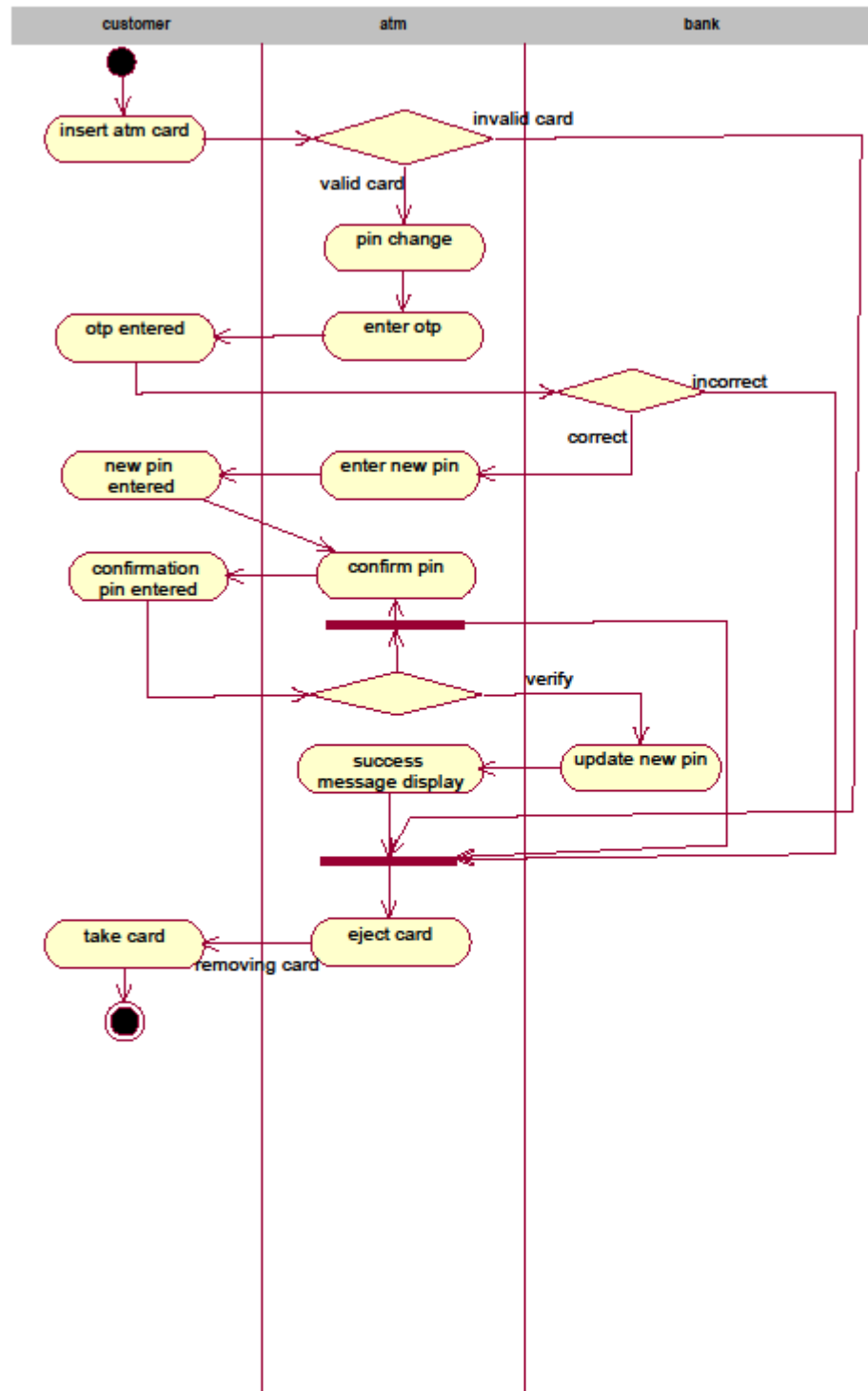
### a) Balance check activity diagram



### b) Money transfer activity diagram

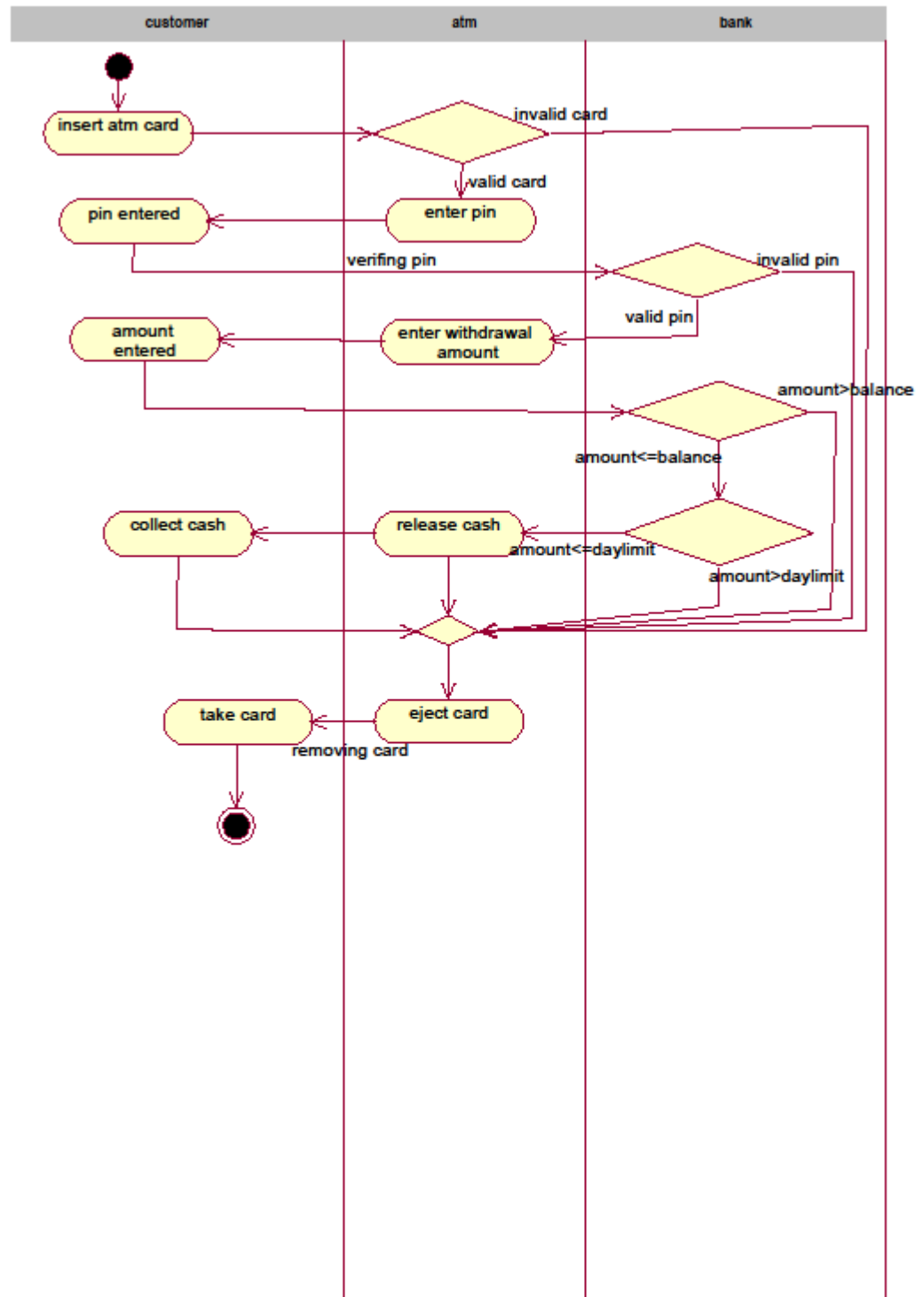


### c) Pin change activity diagram



File: C:\Users\User\Documents\ATM1.mdl 10:03:08 AM Wednesday, August 15, 2018 Activity Diagram: Logical View / ac1pinchange1 Page 1

#### d) Withdraw activity diagram

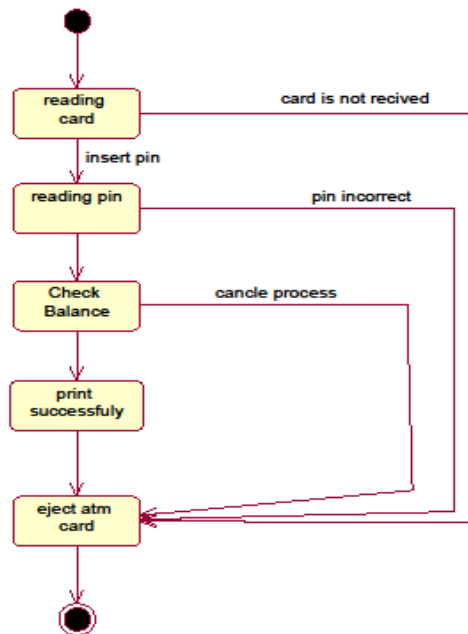


File: C:\Users\User\Documents\ATM1.mdl 10:03:23 AM Wednesday, August 15, 2018 Activity Diagram: Logical View / ac1withdrawal Page 1

### 1.3 State diagram

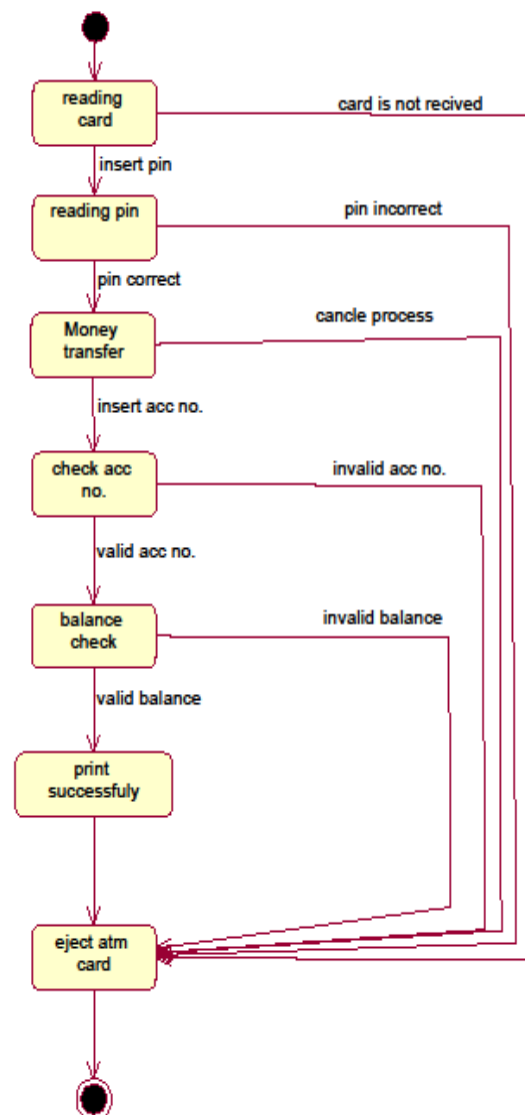
A **state diagram** is a **diagram** used in computer science to describe the behavior of a system considering all the possible **states** of an object when an event occurs. This behavior is represented and analyzed in a series of events that occur in one or more possible **states**.

### a) Balance check state diadram

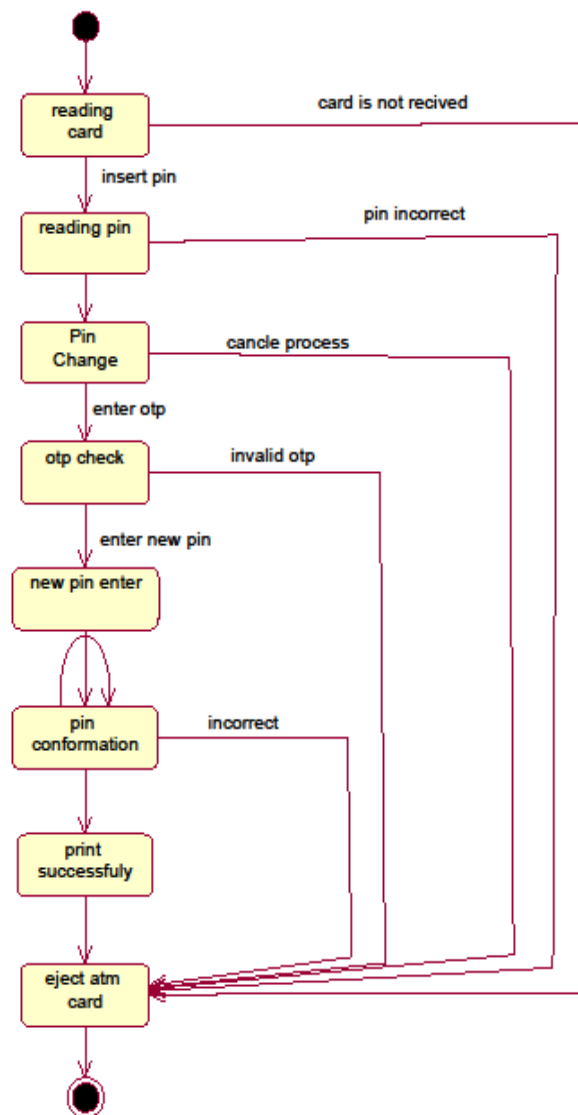


### b) Money transefor state diagram

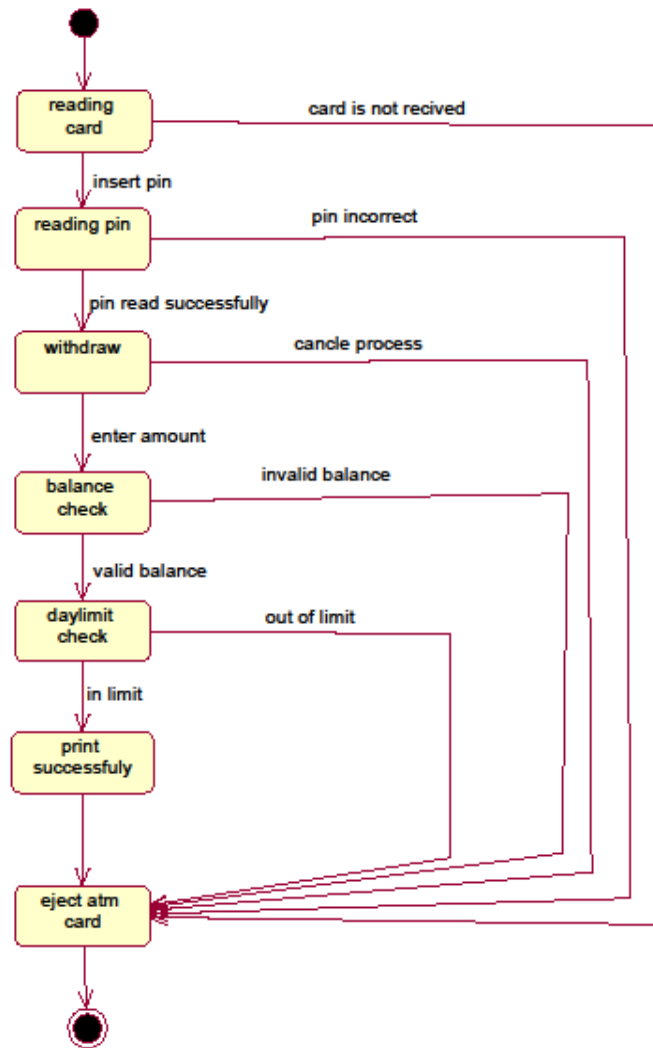




### c) Pin change state diagram



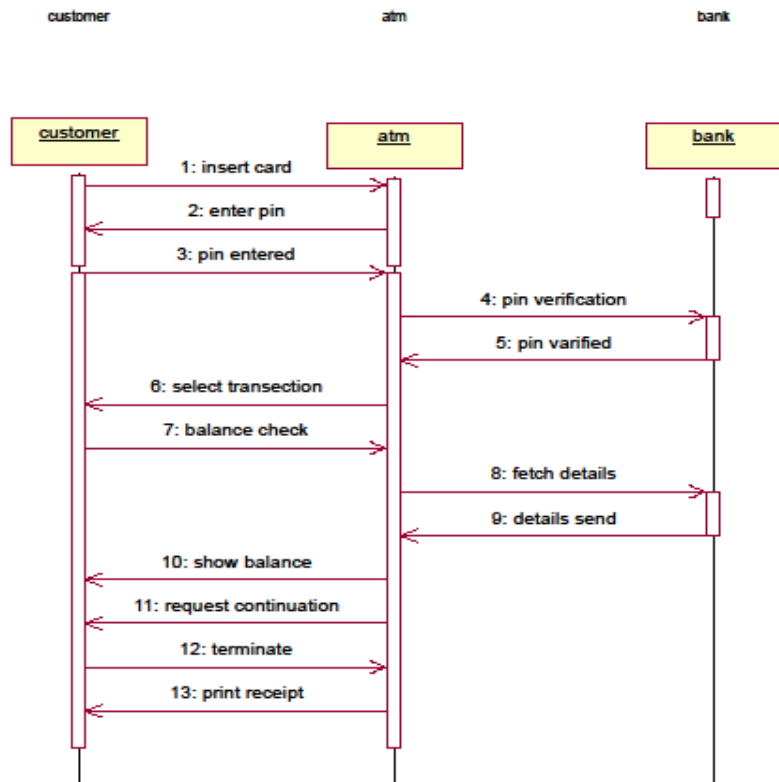
#### d) Withdram state diagram



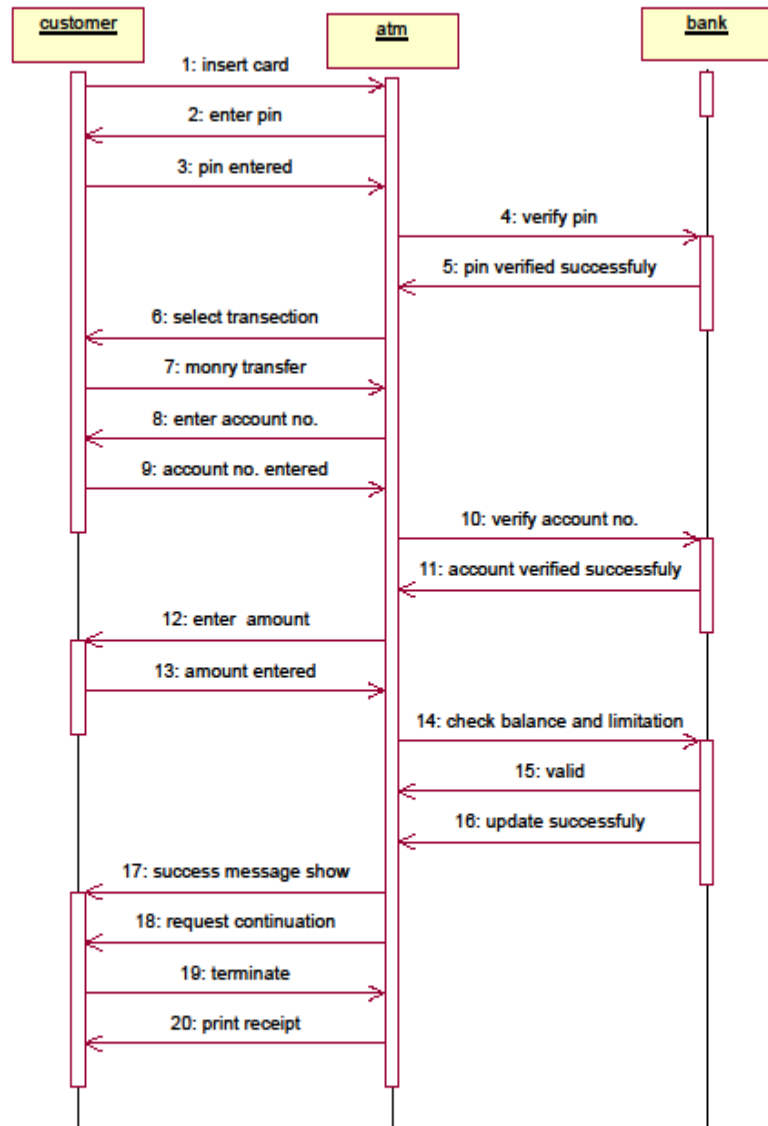
## 1.4Sequence diagram

A **sequence diagram** shows object interactions arranged in time **sequence**. It depicts the objects and classes involved in the scenario and the **sequence** of messages exchanged between the objects needed to carry out the functionality of the scenario.

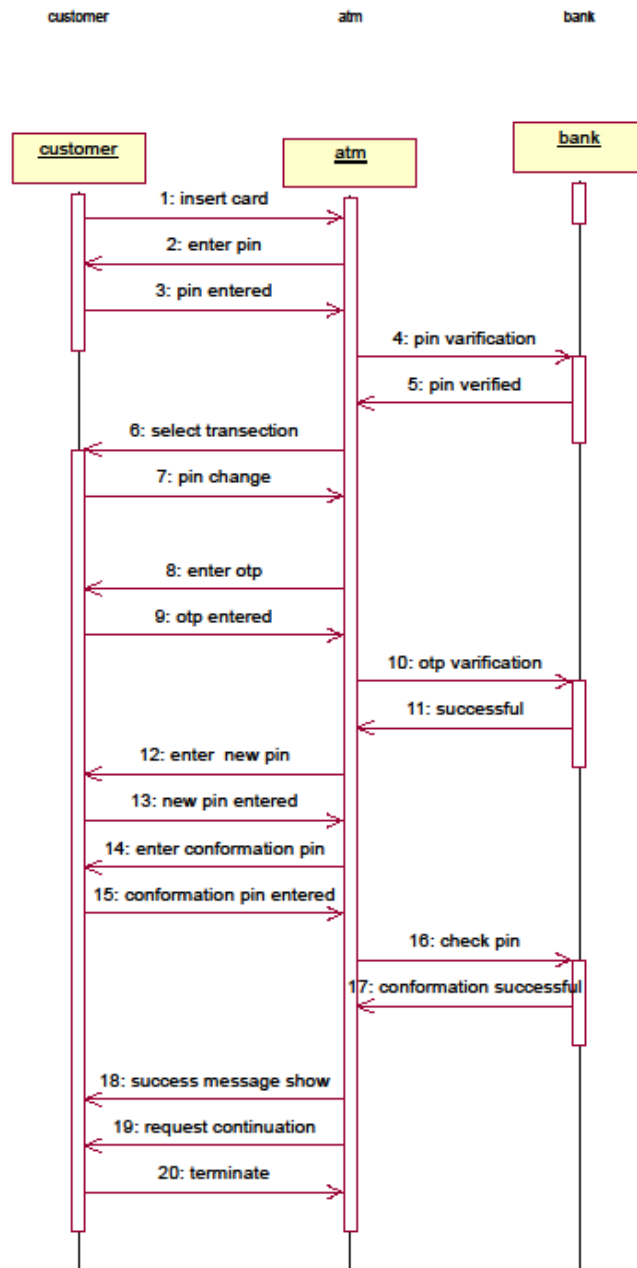
### a) Balance check sequence diagram



### b) Money transfer sequence diagram

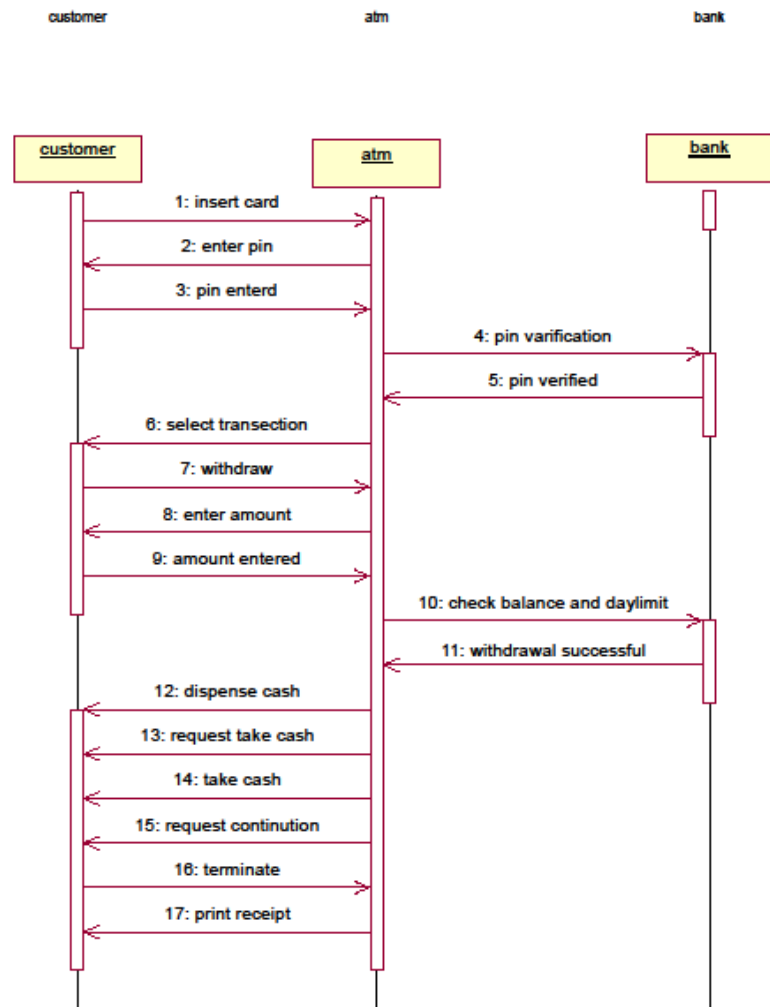


### c) Pin change sequence diagram



File: C:\Users\User\Documents\ATM1.mdl 9:58:20 AM Wednesday, August 15, 2018 Sequence Diagram: Logical View / se1pinchange Page 1

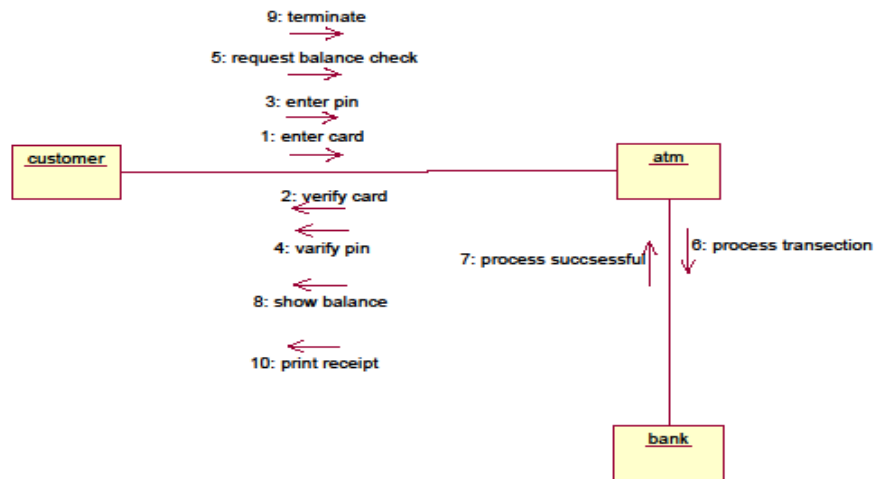
#### d) Withdraw sequence diagram



File: C:\Users\User\Documents\ATM1.mdl 9:58:39 AM Wednesday, August 15, 2018 Sequence Diagram: Logical View / se1w\thdrawal Page 1

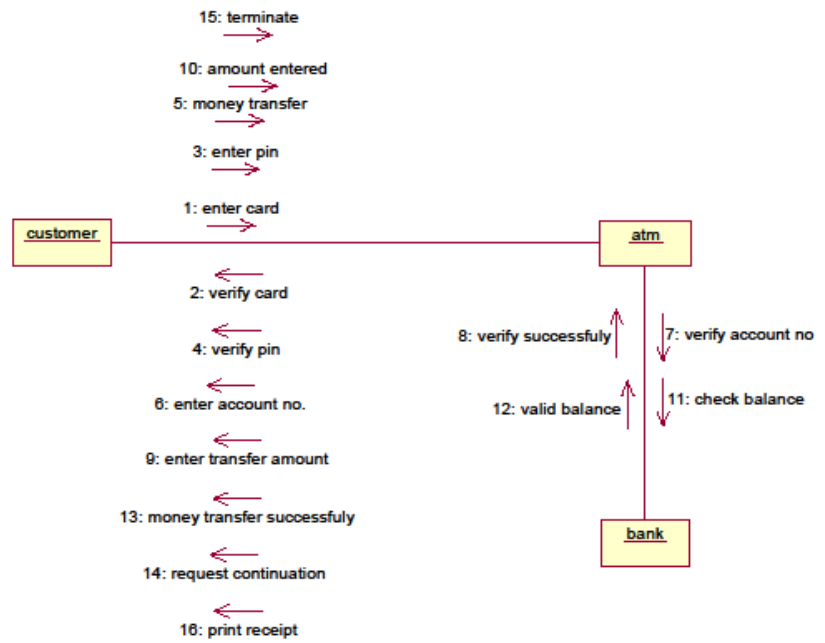
## 1.5 Collaboration diagram

### a) Balance check collaboration diagram

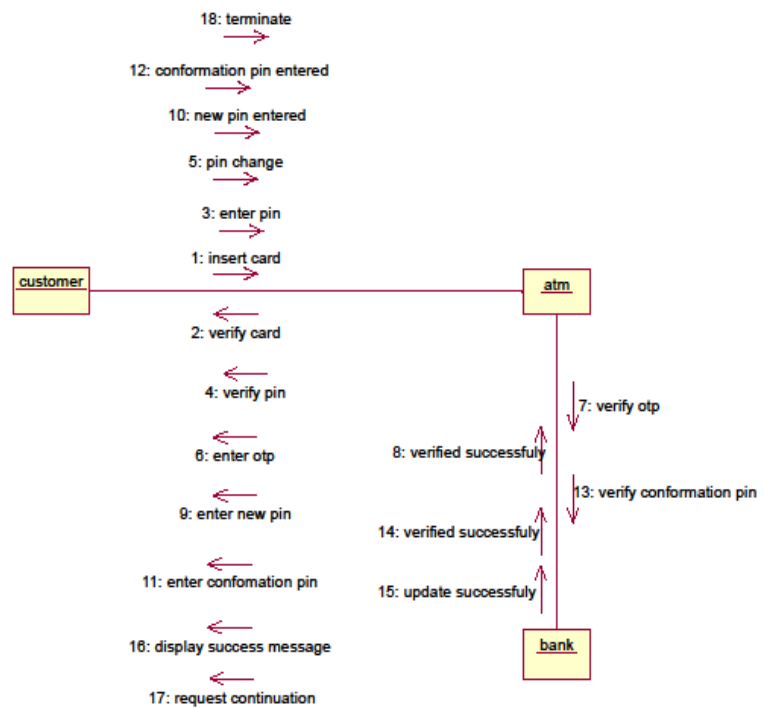


## b) Money transaction collaboration diagram



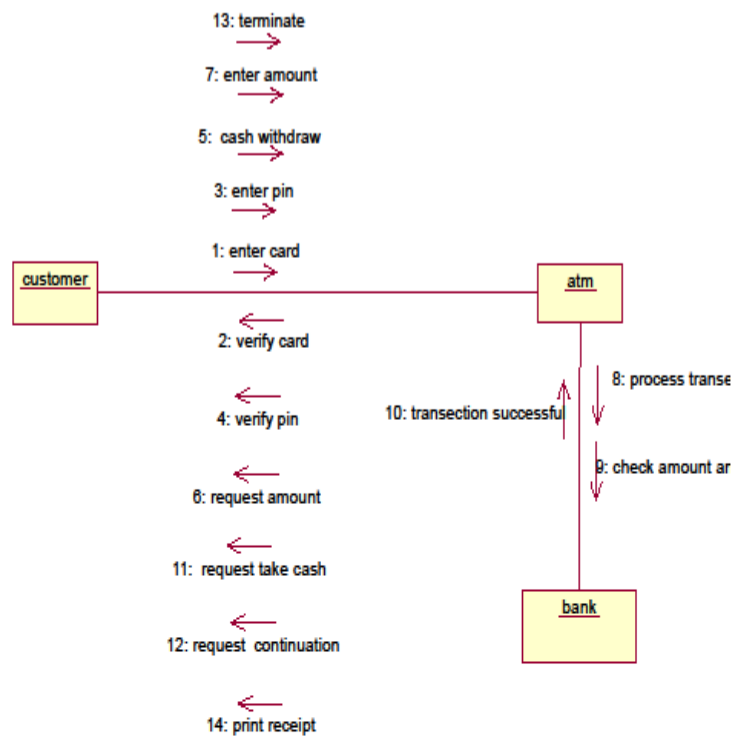


### c) Pin change collaboration diagram



File: C:\Users\User\Documents\ATM1.mdl 9:56:17 AM Wednesday, August 15, 2018 Collaboration Diagram: Logical View / co1pin change Page 1

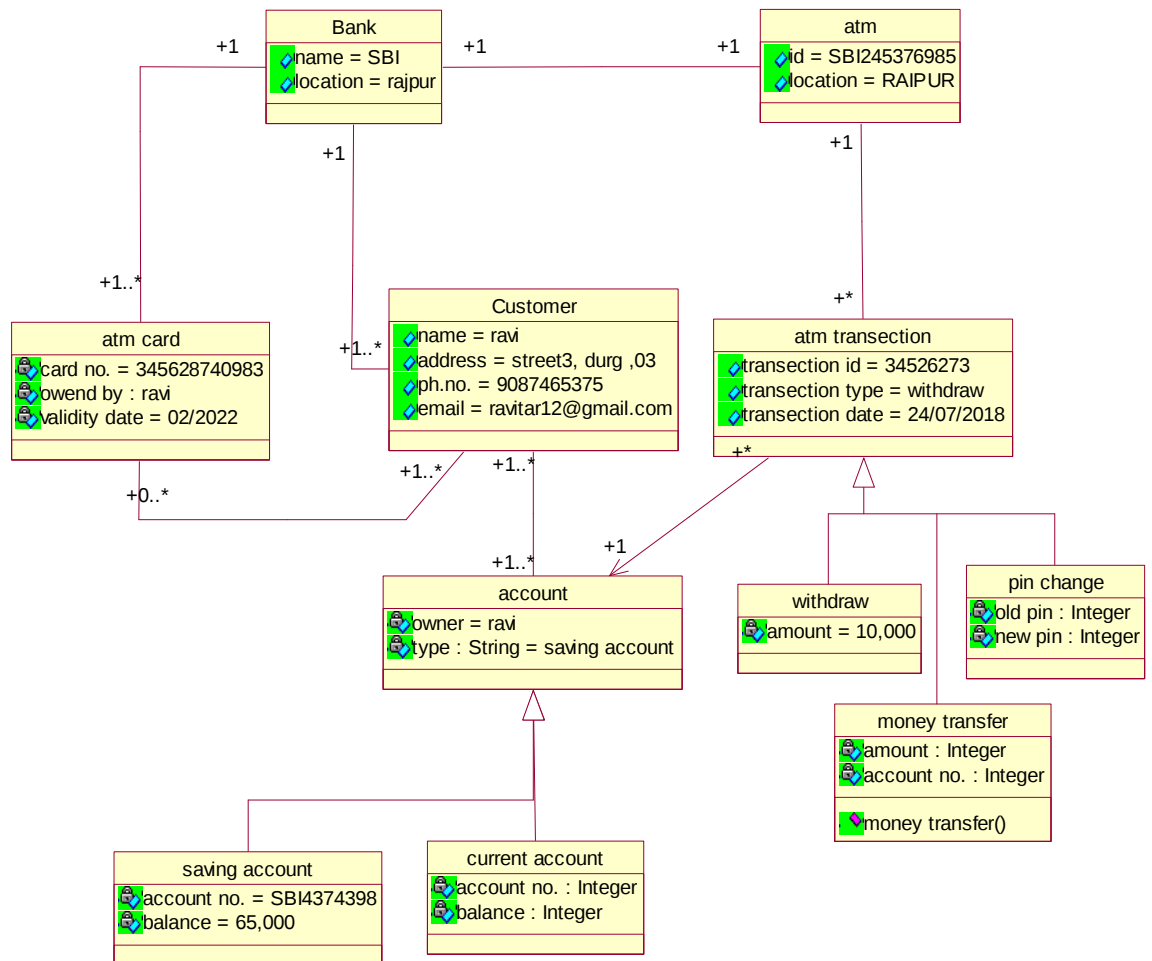
#### d) Withdraw collaboration diagram



## 1.6 Class diagram

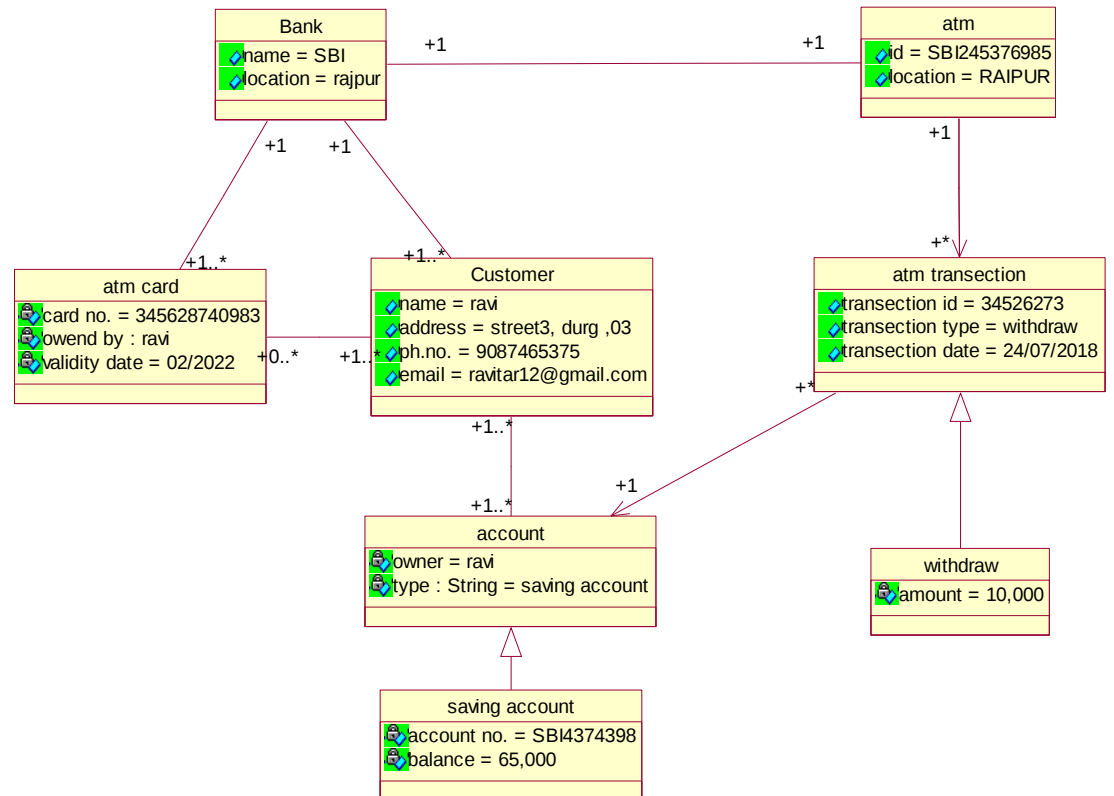
A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's *classes*, their attributes, operations (or methods), and the relationships among objects.

## Class diagram for atm machine



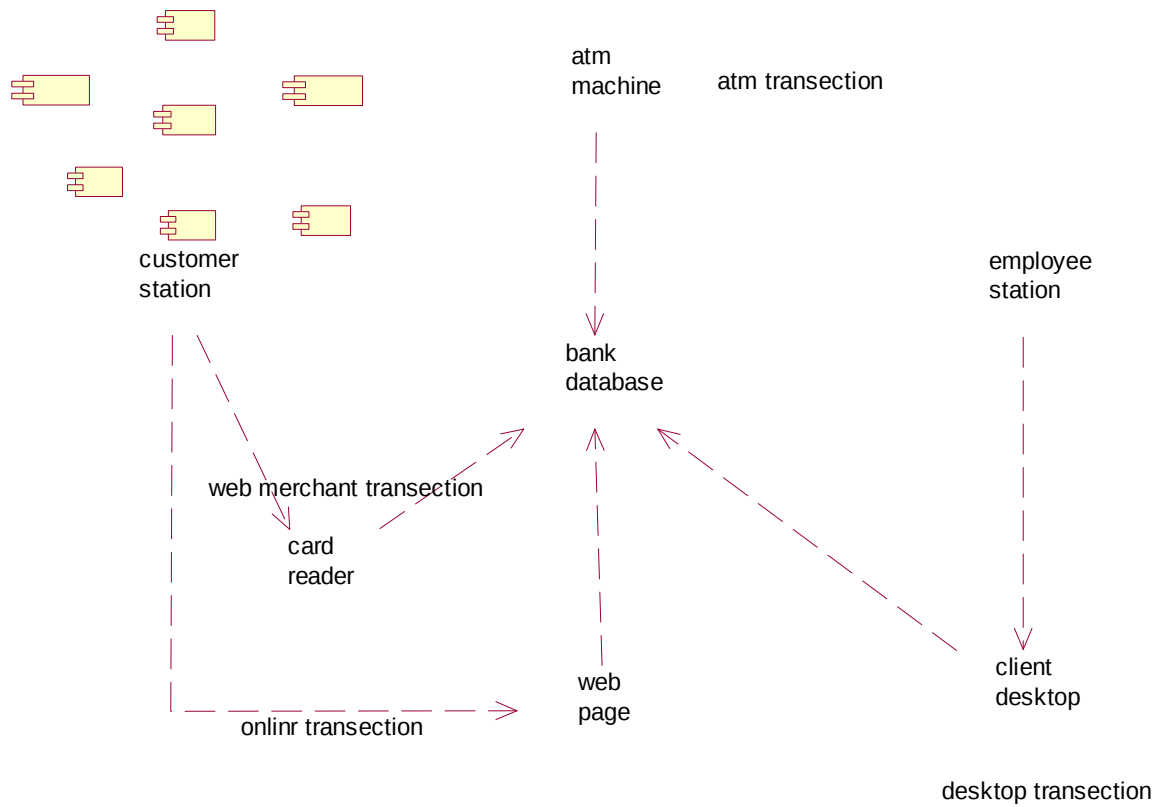
## 1.7 Object diagram

An **object diagram** is a graph of instances, including **objects** and data values. A static **object diagram** is an instance of a class **diagram**; it shows a snapshot of the detailed state of a system at a point in time.



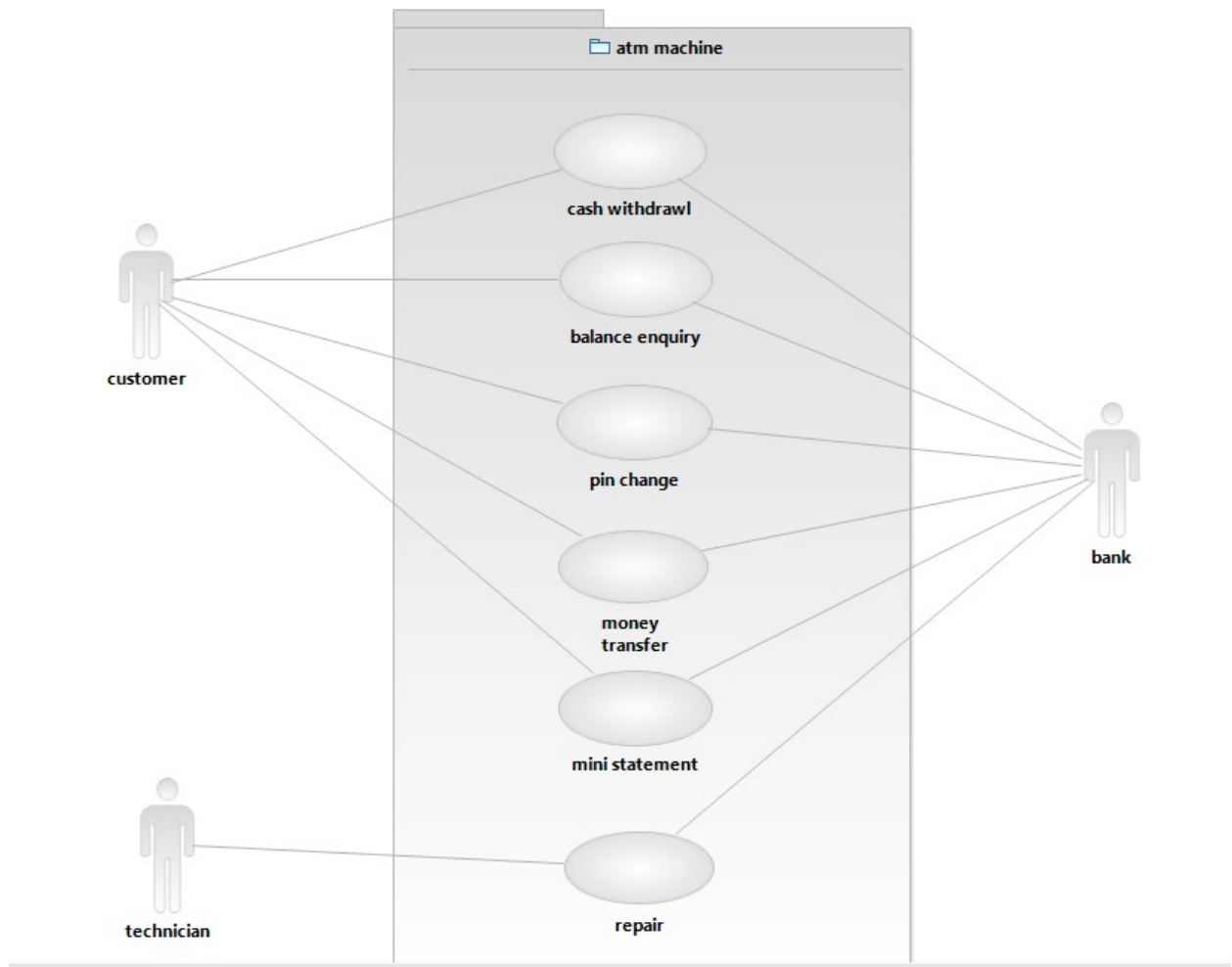
## 1.8 Component diagram

In Unified Modeling Language (UML), a **component diagram** depicts how **components** are wired together to form larger **components** or software systems. They are used to illustrate the structure of arbitrarily complex systems.



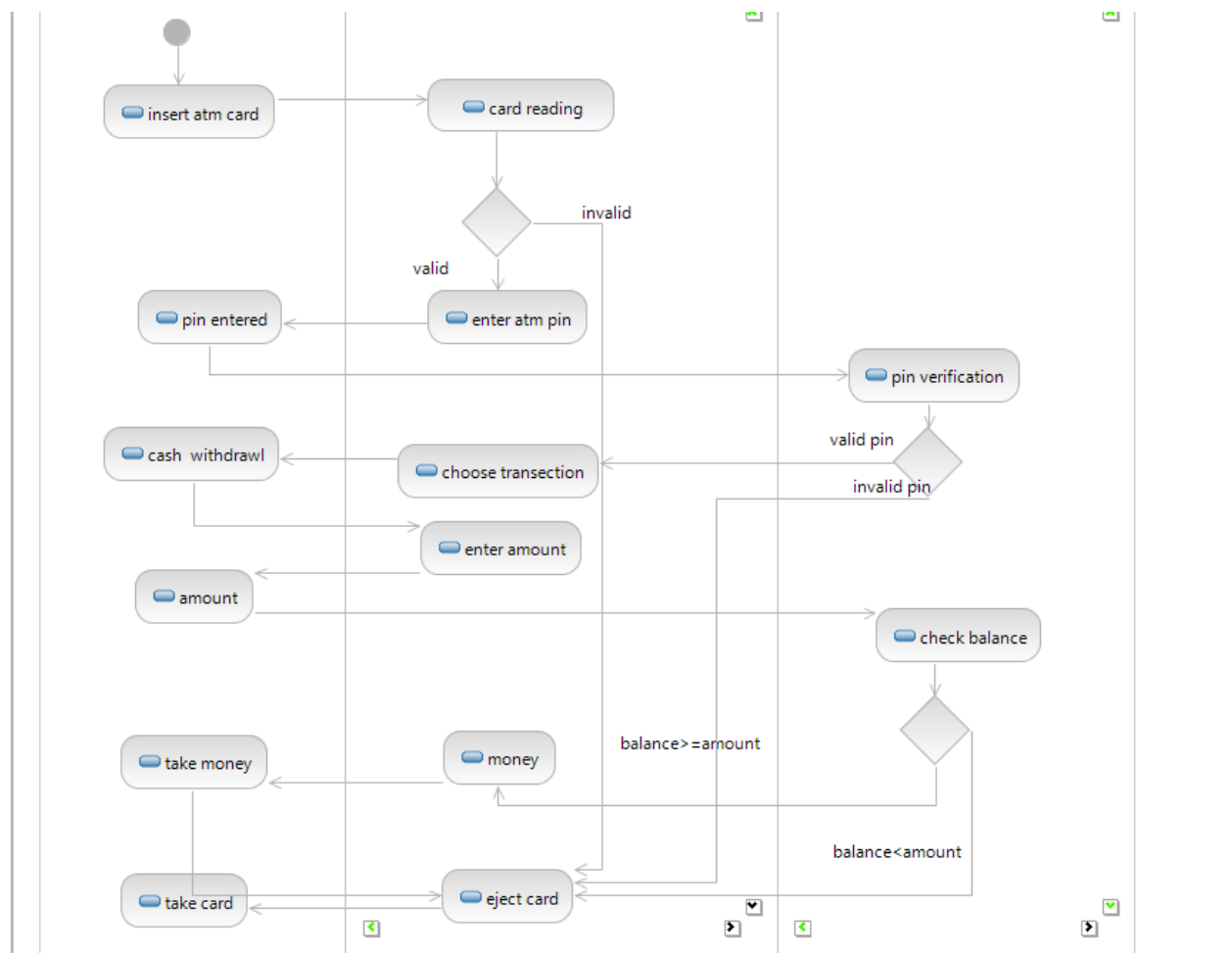
## 2 UML DIAGRAMS(RATIONAL ROSE)

### 1. Use case



## 2. Activity diagram.

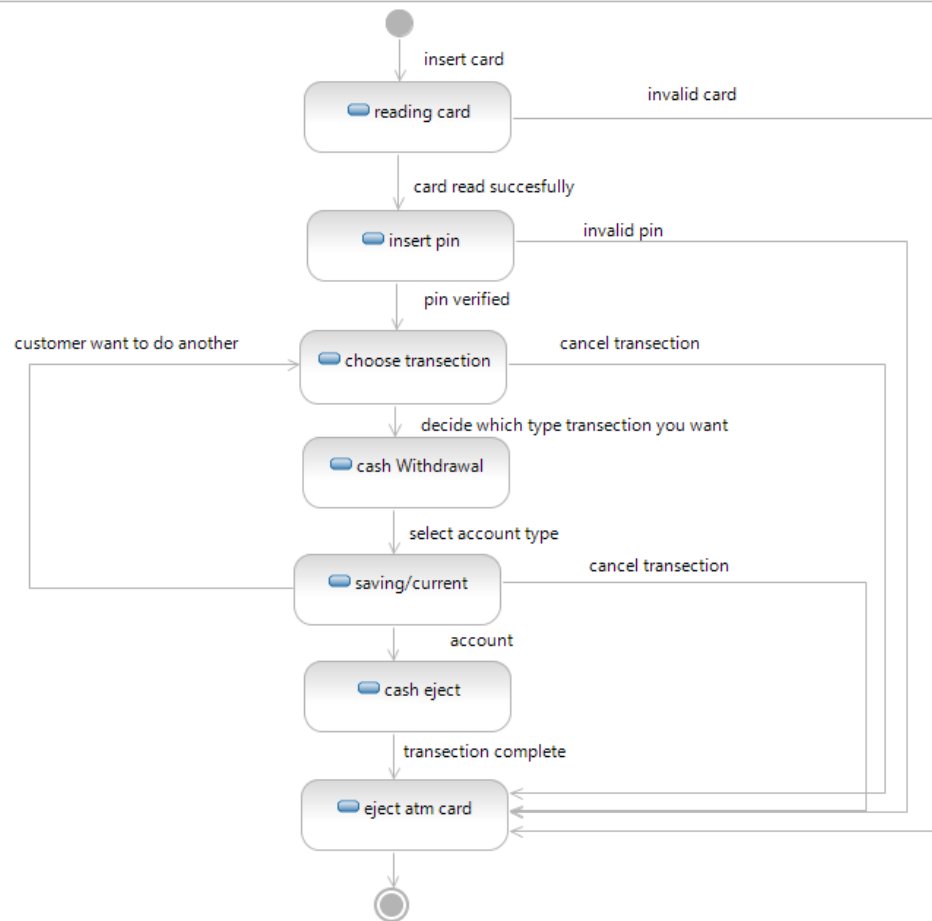
### 2.1 cash withdraw diagram



### 3 State diagram

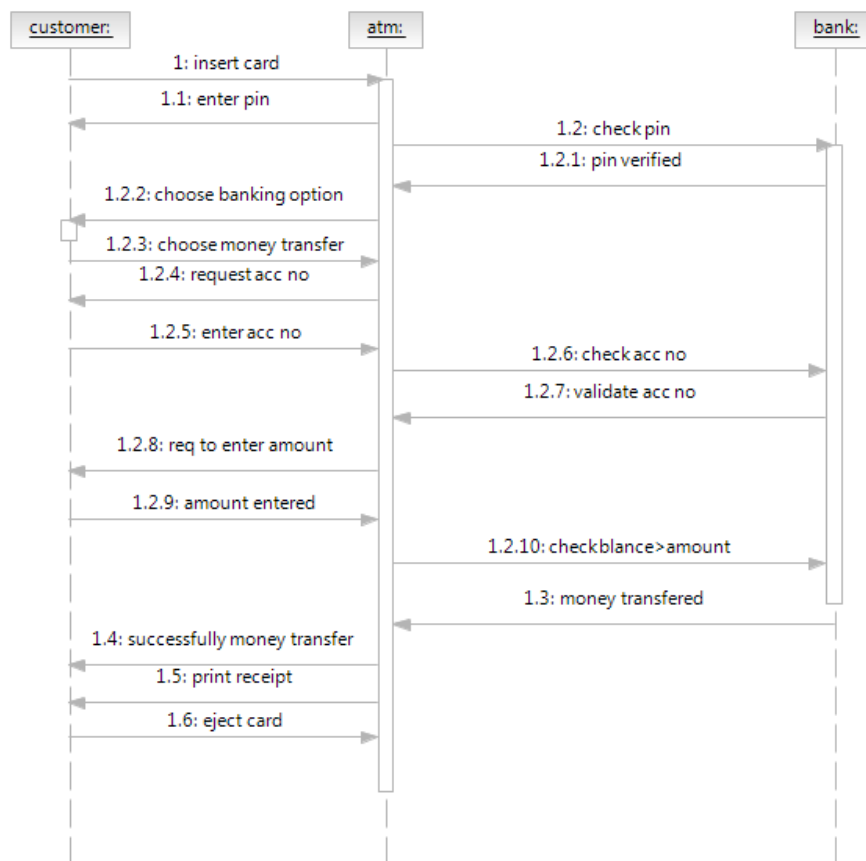
#### 3.1cash withdraw diagram





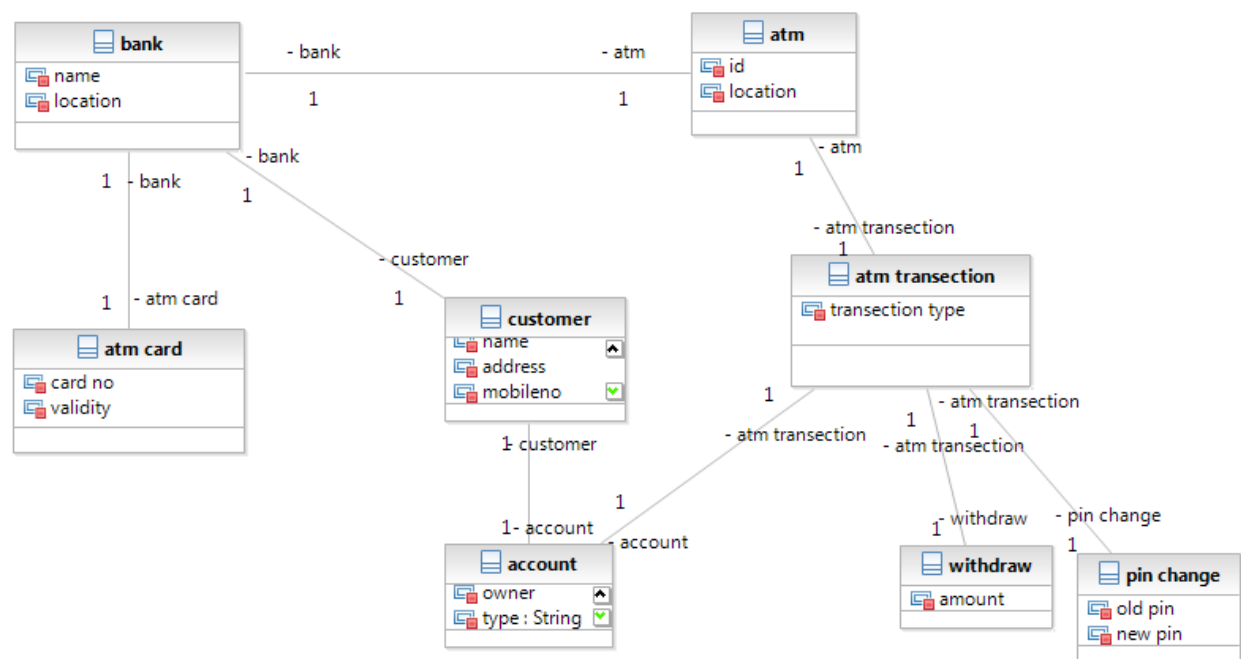
## 4 Sequence diagram

### 4.1 Money transfer diagram

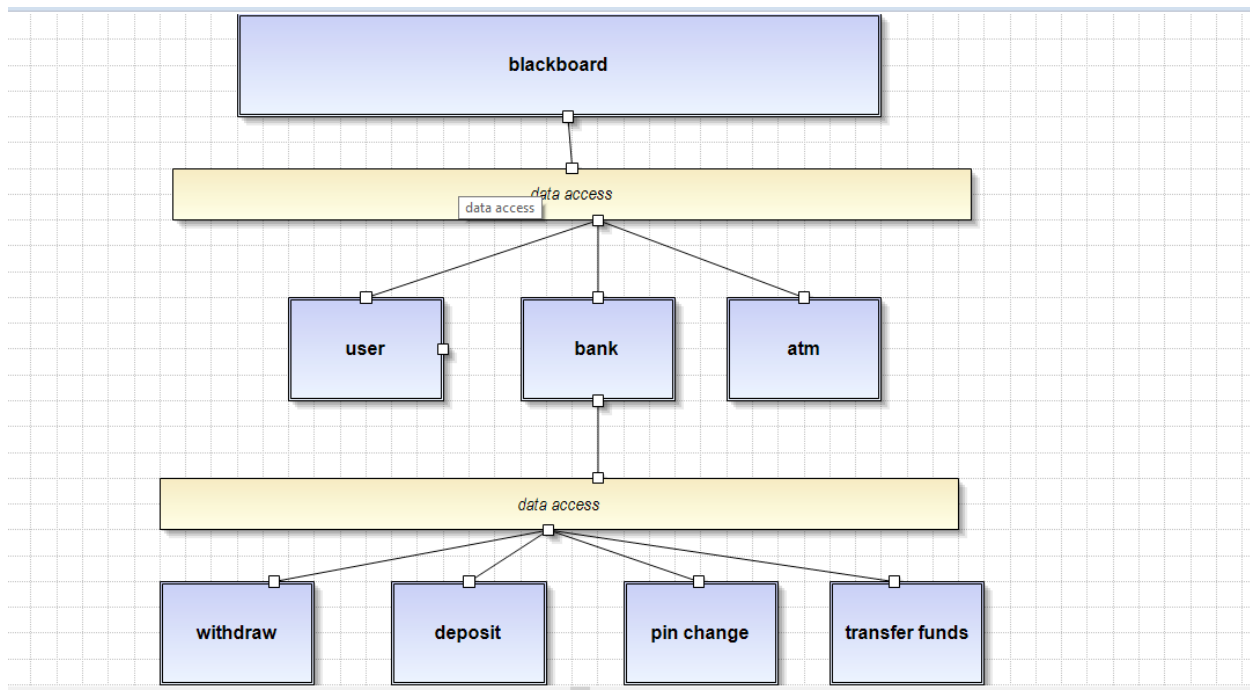


## 5 Class diagram

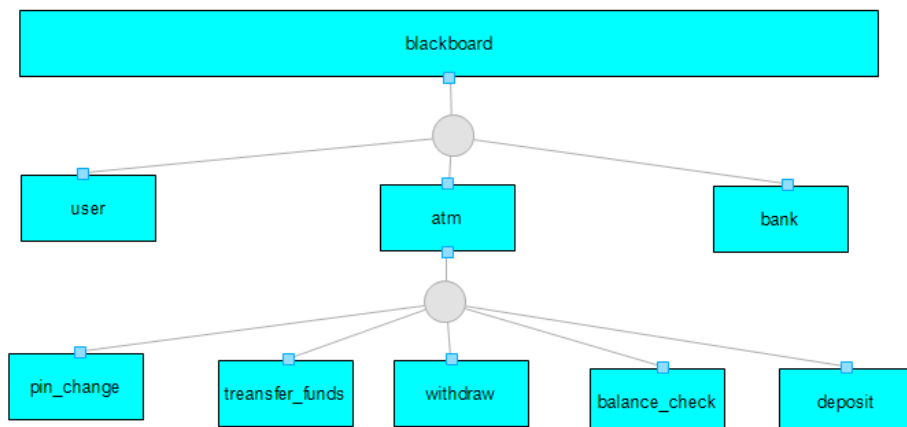
Class diagram for atm machine



## 4 ATM ARCHITECTURE USING ARCH\_STUDIO

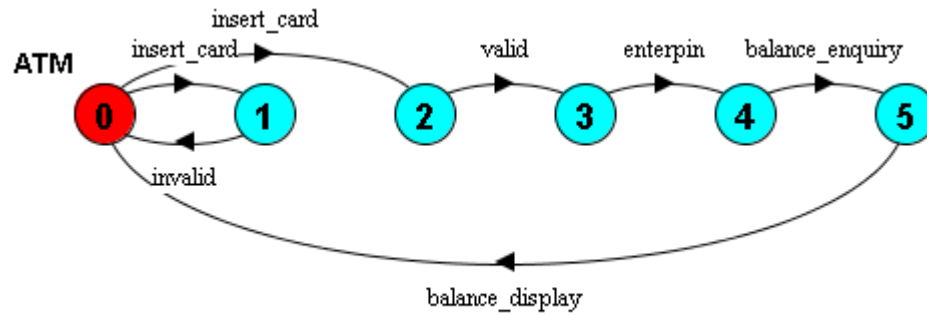


## 4 ATM ARCHITECTURE USING ACME STUDIO



## 5 ATM TRANSITION DIAGRAM USING LTSA

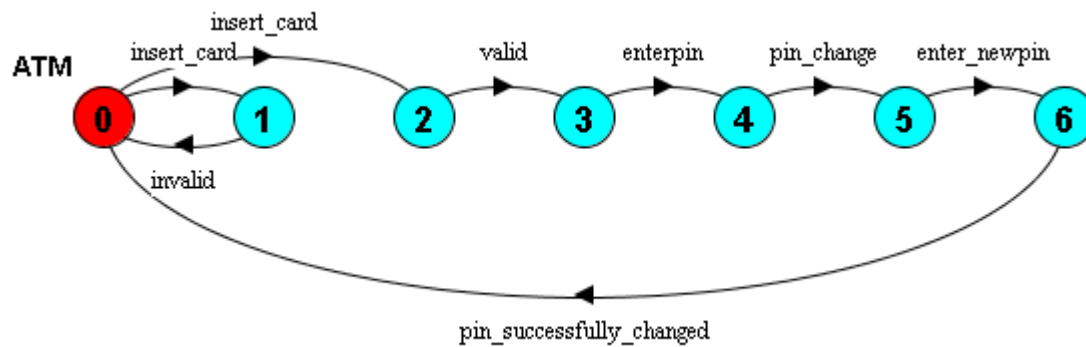
### 4.1 balance check



### Code:

```
ATM =      (insert_card -> valid -> enterpin -> balance_enquiry -> balance_display -> ATM
            |insert_card -> invalid -> ATM
            ).
menu RUN = {insert_card}
```

### 4.2 pin change



### Code:

ATM = (insert\_card -> valid -> enterpin -> pin\_change -> enter\_newpin -> pin\_successfully\_changed -> ATM

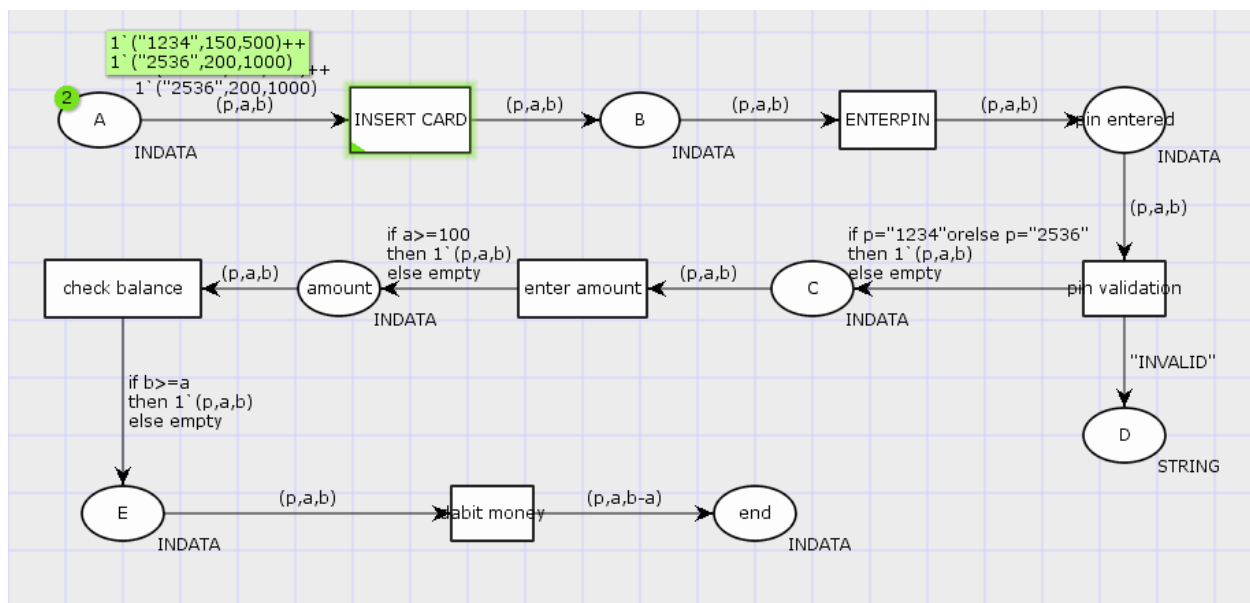
|insert\_card -> invalid -> ATM

).

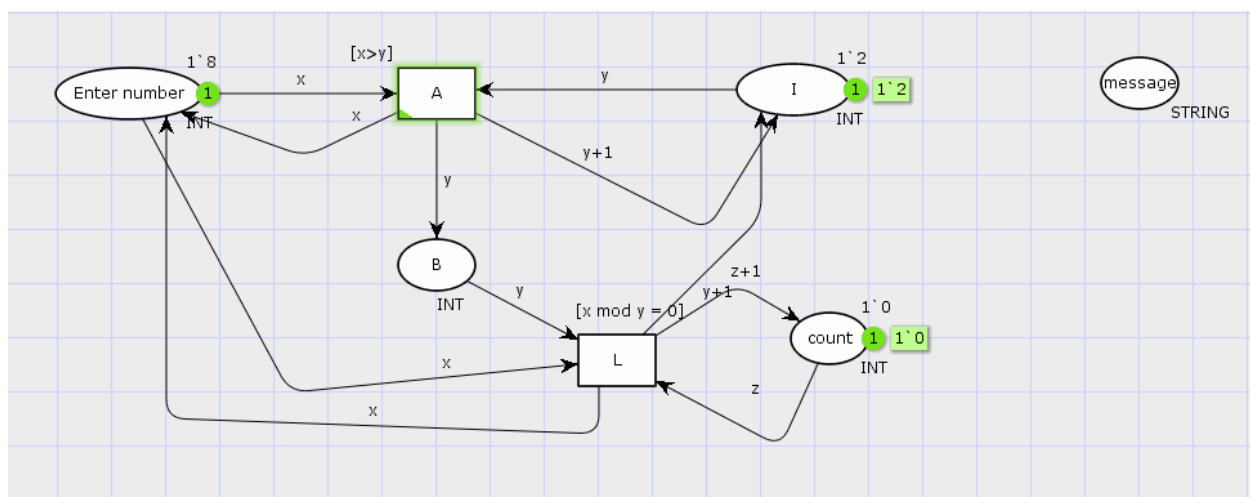
menu RUN = {insert\_card}

## 6 PETRINET DIAGRAM

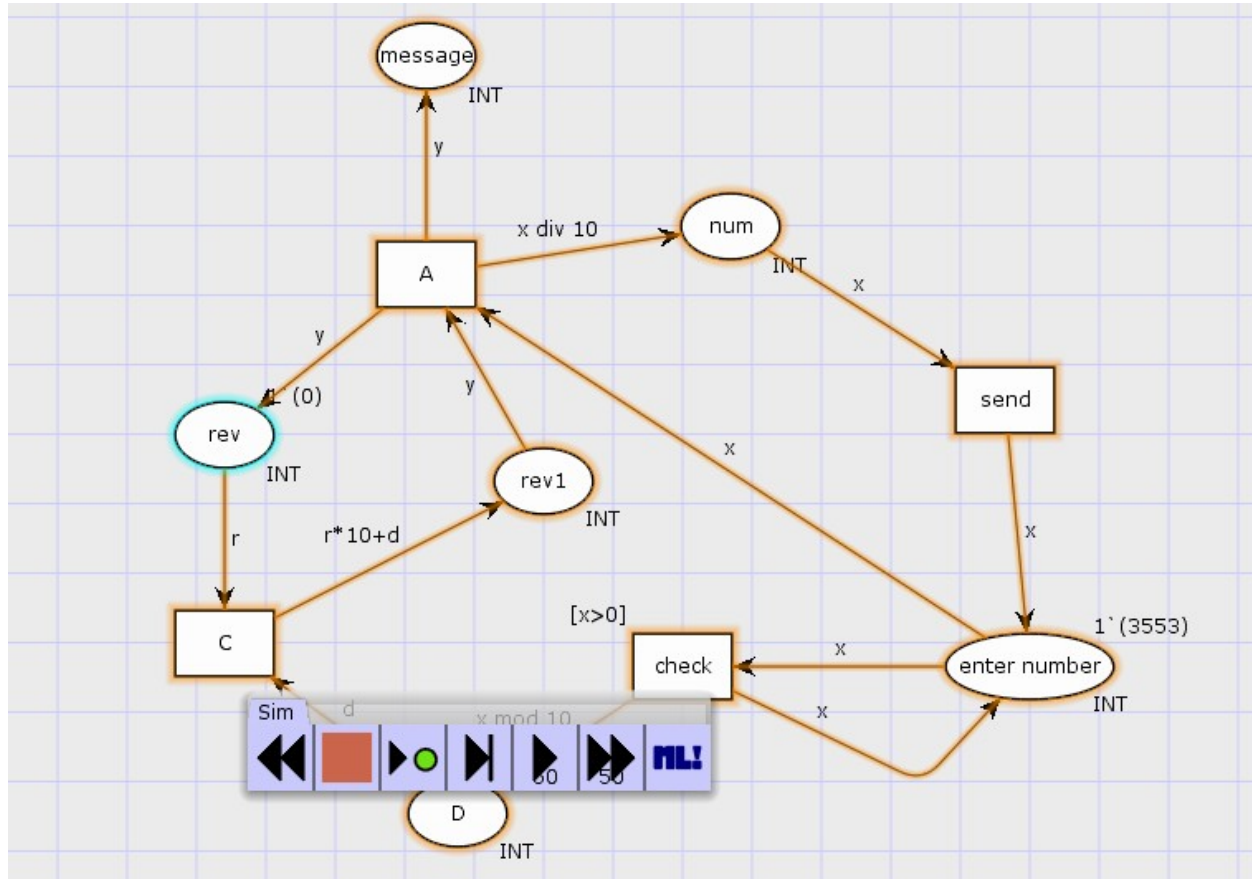
### 6.1 ATM



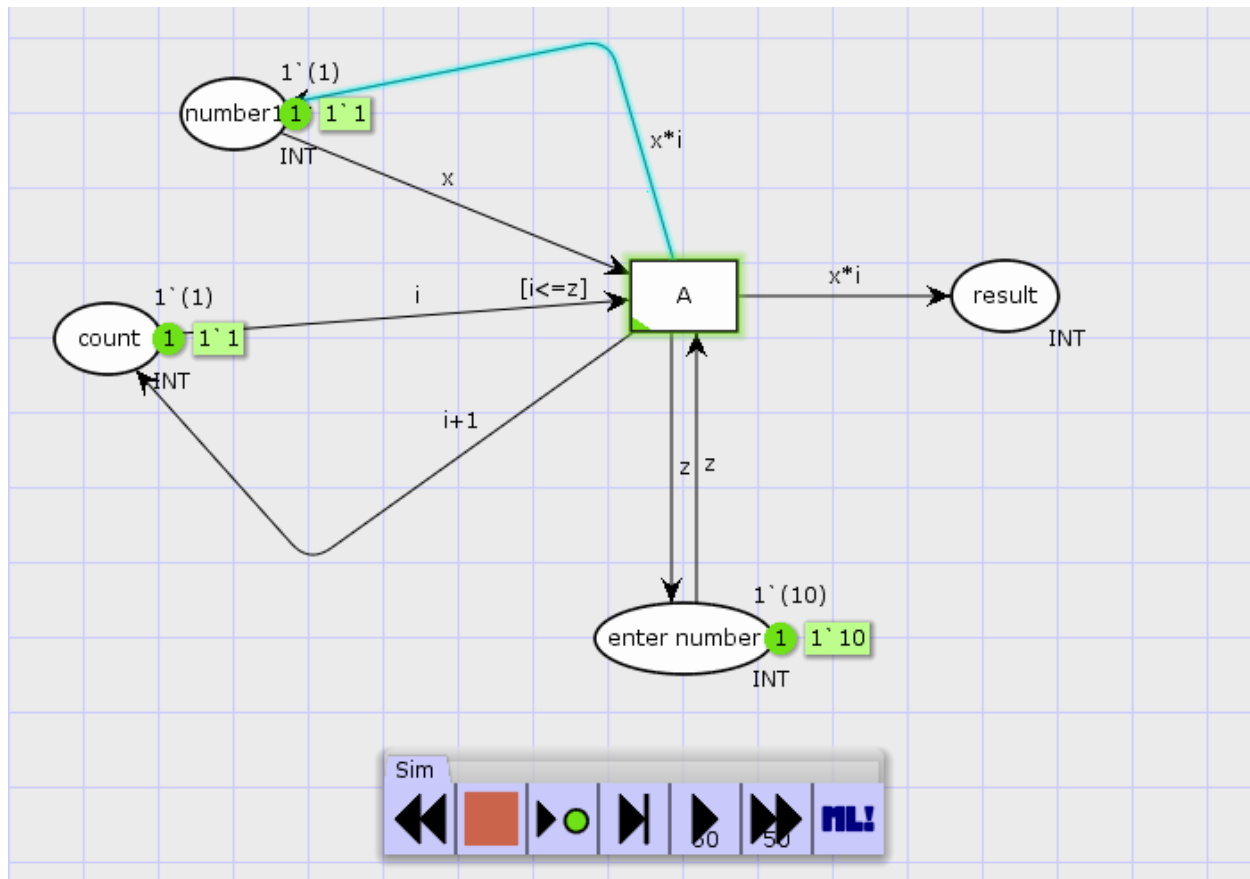
### 6.2 PRIME NUMBER



### 6.3 PALINDROME OF A NUMBER



### 6.4 FACTORIAL OF A NUMBER



## 7 ATM Z-SPECIFICATION

### Code:

```

\documentclass{article}
\usepackage{zed-csp}
\begin{document}
\begin{zed}
CARD ::= cardNo | acctNo | issuingBank | valid
\end{zed}
\begin{zed}
NAME ::= custName | bankName
\end{zed}

```

When any particular operation perform in ATM then related response generated by ATM system.

```

\begin{zed}
ATMResponse ::= opSuccess | opFailed | Nil
\end{zed}

```

For any operation like balance inquiry, cash withdraw it is required to know the status of ATM machine and sufficient balance.

```

\begin{zed}
STATUS ::= available | busy | idle
\end{zed}

```

```

\begin{zed}
RECEIPT ::= receipt
\end{zed}
\begin{zed}
DATE ::= issueDate | expiryDate | todayDate
\end{zed}
ATM system generate following possible types of error messages:
\begin{zed}
ErrorMessage ::= invalidePinNo | invalideCard \\\ | insufficientBalance | invalideAmount
\end{zed}
Amount should also declared for cash withdraw operation. Because every bank provides certain
restriction on minimum amount and maximum amount. The axiomatic definition for some
important constraints. It is required to declared some variables globally.
\begin{axdef}
minAmount: \nat \\\
maxAmount: \nat \\\
withdrawAmount: \nat \\\
accountBalance: \nat \\\
pinNo: \nat \\\
maxTran: \nat \\\
moneyInMachine: \nat
\where
withdrawAmount \leq maxAmount
\end{axdef}
\begin{schema}{Bank}
bankName: NAME \\\
card: CARD \\\
has: NAME \fun CARD \\\
balance: \nat \\\
todayDate: DATE
\end{schema}
\begin{schema}{ATM}
balance: \nat \\\
maxAmount: \nat \\\
todayDate: DATE
\end{schema}
\begin{schema}{CardReader} \\\
card?: CARD \\\
date: DATE \\\
status: STATUS \\\
message!: ErrorMessage \\\
\where
status = busy \\\
date = expiryDate \implies message! = invalideCard
\end{schema}
Schema have two parts first is declaration part and second is predicate part. In the declaration
part of schema CardReader, first variable card? represent the input by using ? symbol and
variable message? represent for output using ! symbol.
\begin{schema}{BalanceEnquiry} \\\
\Xi ATM \\\
\Xi Bank \\\
response! : ATMResponse \\\
accountBalance : \nat \\\

```



```

receipt! : RECEIPT \\
status : STATUS
\where
status = busy \\
moneyInMachine' = moneyInMachine \\
accountBalance' = accountBalance \\
response! = opSuccess \\
receipt!.amount = accountBalance \\
status' = idle \\
\end{schema}

```

In BalanceEnquiry schema  $\$Xi\$$  ATM and  $\$Xi\$$  Bank denotes that the state of schemas of ATM and Bank will not change after completing this operation. moneyInMachine' and accountBalance' represent next state of moneyInMachine and accountBalance by using ' operator.

```

\begin{schema}{CashWithdraw}\\
\Delta ATM \\
\Delta Bank \\
acctNo? : CARD \\
m? : \nat \\
accountBalance : \nat \\
response! : ATMResponse \\
receipt! : RECEIPT \\
status : STATUS
\where
status = busy \\
balance' = balance \oplus acct? \\ \fun balance - m? \\
response! = opSuccess \\
receipt!.amount = m? \\
status' = Idle
\end{schema}

```

In schema CashWithdraw  $\Delta$  ATM and  $\Delta$  Bank represent that after cash withdraw operation the state of ATM and the state of Bank both will change. The operator  $\oplus$  used for overwrite operation.

```

\begin{schema}{FundTransfer}\\
\Delta ATM \\
\Delta Bank \\
acct1?, acct2? : ACCOUNT \\
m? : \nat \\
balance : \nat \\
response! : ATMResponse \\
receipt! : RECEIPT \\
status : STATUS
\where
status = busy \\
balance \geq m? \\
accountBalance' = accountBalance \oplus \\ \fun accountBalance - m? \\
accountBalance' = accountBalance \oplus \\ \fun accountBalance + m? \\
response! = opSuccess \\
receipt!.amount = m? \\
status' = Idle
\end{schema}
\end{document}

```

## Output:

$$CARD ::= cardNo \mid acctNo \mid issuingBank \mid valid$$
$$NAME ::= custName \mid bankName$$

When any particular operation perform in ATM then related response generated by ATM system.

$$ATMResponse ::= opSuccess \mid opFailed \mid Nil$$

For any operation like balance inquiry, cash withdraw it is required to know the status of ATM machine and sufficient balance.

$$STATUS ::= available \mid busy \mid idle$$
$$RECEIPT ::= receipt$$
$$DATE ::= issueDate \mid expiryDate \mid todayDate$$

ATM system generate following possible types of error messages:

$$ERRORMessage ::= invalidePinNo \mid invalideCard \\ \mid insufficientBalance \mid invalideAmount$$

Amount should also declared for cash withdraw operation. Because every bank provides certain restriction on minimum amount and maximum amount. The axiomatic definition for some important constraints. It is required to declared some variables globally.

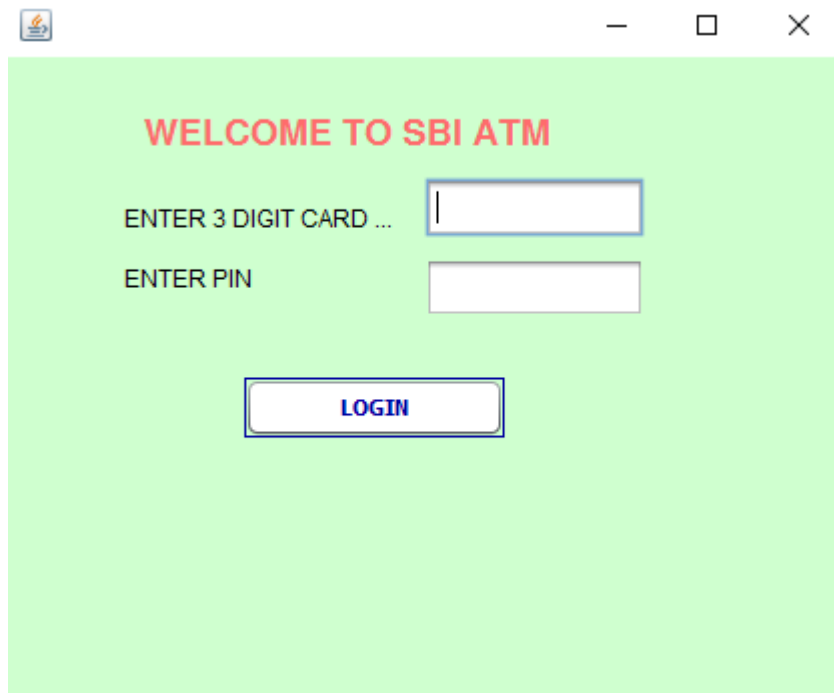
|  |
|--|
| $\begin{array}{l} minAmount : \mathbb{N} \\ maxAmount : \mathbb{N} \\ withdrawAmount : \mathbb{N} \\ accountBalance : \mathbb{N} \\ pinNo : \mathbb{N} \\ maxTran : \mathbb{N} \\ moneyInMachine : \mathbb{N} \\ \hline withdrawAmount \leq maxAmount \end{array}$ |
|--|

*Bank*

|  |
|--|
| $\begin{array}{l} bankName : NAME \\ card : CARD \\ has : NAME \rightarrow CARD \\ balance : \mathbb{N} \\ todayDate : DATE \end{array}$ |
|--|

## 8 ATM PROJECT (SNAPSHOT)

1)login page



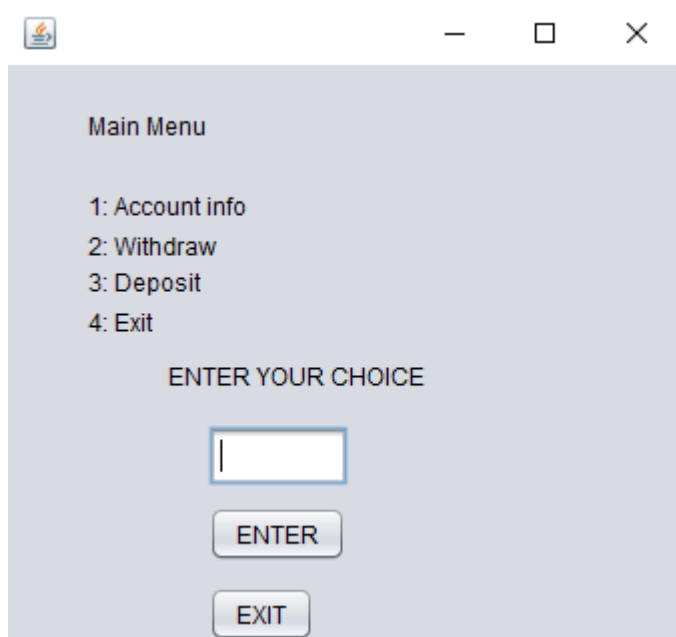
A screenshot of a web browser window displaying the SBI ATM login page. The page has a light green background. At the top, it says "WELCOME TO SBI ATM" in red. Below this, there are two input fields: "ENTER 3 DIGIT CARD ..." and "ENTER PIN". A "LOGIN" button is centered below the input fields. The browser window has a standard title bar with a minimize, maximize, and close button.

WELCOME TO SBI ATM

ENTER 3 DIGIT CARD ...

ENTER PIN

2)main menu



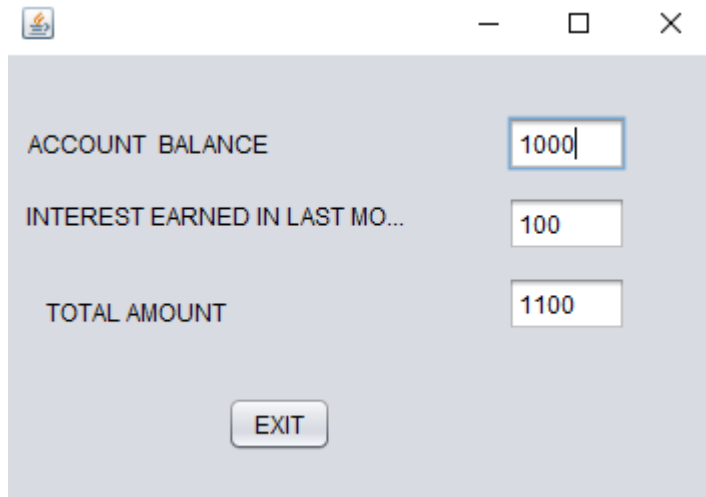
A screenshot of a web browser window displaying the SBI ATM main menu. The page has a light gray background. It lists four options: "1: Account info", "2: Withdraw", "3: Deposit", and "4: Exit". Below the list, it says "ENTER YOUR CHOICE". There is an input field for the choice, and two buttons labeled "ENTER" and "EXIT". The browser window has a standard title bar with a minimize, maximize, and close button.

Main Menu

1: Account info  
2: Withdraw  
3: Deposit  
4: Exit

ENTER YOUR CHOICE

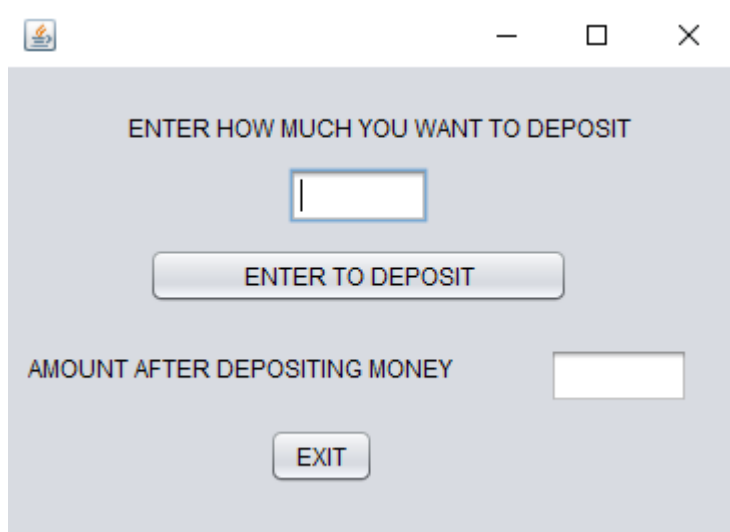
### 3)balance check page



A window titled "Balance check page" with a standard Windows-style title bar (minimize, maximize, close buttons). The window has a light gray background. It contains three labels on the left and three corresponding text input fields on the right. The first label is "ACCOUNT BALANCE" and its input field contains "1000". The second label is "INTEREST EARNED IN LAST MO..." and its input field contains "100". The third label is "TOTAL AMOUNT" and its input field contains "1100". Below these fields is a single "EXIT" button.

|                               |                                   |
|-------------------------------|-----------------------------------|
| ACCOUNT BALANCE               | <input type="text" value="1000"/> |
| INTEREST EARNED IN LAST MO... | <input type="text" value="100"/>  |
| TOTAL AMOUNT                  | <input type="text" value="1100"/> |

### 4)deposit page

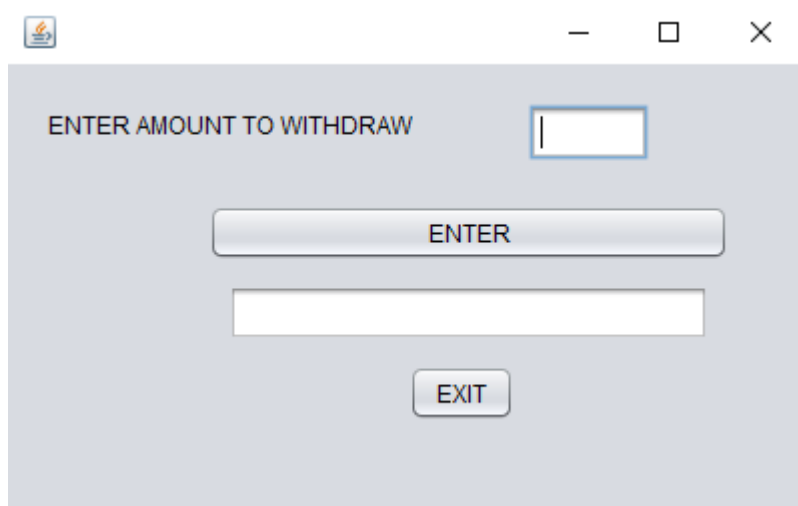


A window titled "Deposit page" with a standard Windows-style title bar (minimize, maximize, close buttons). The window has a light gray background. It contains a label "ENTER HOW MUCH YOU WANT TO DEPOSIT" above a text input field. Below the input field is a button labeled "ENTER TO DEPOSIT". At the bottom, there is a label "AMOUNT AFTER DEPOSITING MONEY" next to an empty text input field, and an "EXIT" button below it.

ENTER HOW MUCH YOU WANT TO DEPOSIT

AMOUNT AFTER DEPOSITING MONEY

## 5) withdraw page



A screenshot of a graphical user interface window titled "withdraw page". The window has a light gray background and a standard title bar with a minimize button, a maximize button, and a close button. The main content area contains the following elements:

- A label "ENTER AMOUNT TO WITHDRAW" in a dark blue font, positioned to the left of a small text input field.
- A small text input field with a blue border and a vertical cursor, containing a single vertical line.
- A wide, light gray button with rounded corners and a subtle gradient, labeled "ENTER" in a dark blue font.
- A wide, empty text input field with a light gray background and a thin border.
- A small, light gray button with rounded corners and a subtle gradient, labeled "EXIT" in a dark blue font.