1. Introduction:

Object oriented analysis is the procedure of identifying software engineering requirements and specifications which develops a set of graphical system models during the development life cycle of the software. UML stands for Unified Modeling Language. The UML tool set includes diagrams that allow us to visualize the construction of an object oriented system. They include use case diagrams, activity diagrams, sequence diagrams, communication diagrams, class diagrams, and state chart diagrams. UML is fundamentally based on an object-oriented technique known as use case modeling. Different scenarios are created for each different set of conditions of a use case.

2. UML Diagram:

Unified Modeling Language (UML) provides a standardized set of tools to document the analysis and design of a software system. The UML diagram serves as a complete design that requires the actual implementation of the system or software. The main components of UML are things, relationships, and diagrams. Diagrams are related to one another. Structural things are most common; they include classes, interfaces, use cases, and many other elements that provide a way to create models. Group things are used to define boundaries. Annotational things permit the analyst to add notes to the diagrams. Behavioral things describe how things work. Use case diagram is one of the behavioral things.

2.1 Use Case Diagram:

Use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen(how). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior. Use cases represent only the functional requirements of a system. Other requirements such as business rules, quality of service requirements, and implementation constraints must be represented separately, again, with other UML diagrams. Following are the common notations used in a use case diagram:

Use-case: These diagrams are typically developed in the early stage of development and people often apply use case modeling for the following purposes:

- Specify the context of a system
- Capture the requirements of a system

Actor: It is an entity which interacts with a use case (system function). It is similar to the concept of user, but a user can play different roles. Each Actor must be linked to a use case, while some use cases may not be linked to actors.

Communication Link: The participation of an actor in a use case is shown by connecting an actor to a use case by a solid link.

Boundary of system: The system boundary is potentially the entire system.

The Use Case diagrams of different sections of Dutch Bangla Bank Ltd. ,Noapara Branch is shown below:

i) General Banking:

Opening Account:

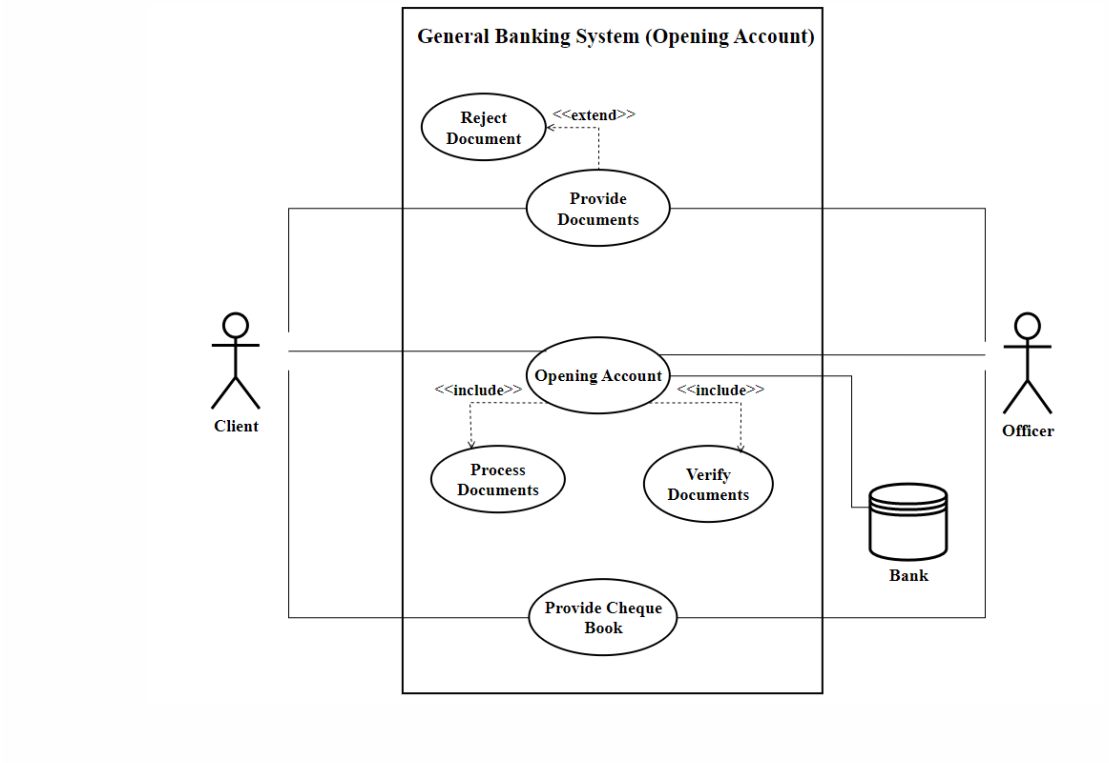

Figure 1: Use Case Diagram for General Banking (Opening Account)

Three actors and six use cases are in the opening account system diagram. The actors are the Client, Bank, and Officer. The use cases are: Provide Documents, Reject Document, Opening Account, Verify Document, Process Document, and Provide Cheque Book. At first, the Client provides the necessary documents to open a bank account. The officer checks these documents. If the information is not appropriate, the documents will be rejected. For opening an account, the documents must be verified and processed. After completing all the formalities, the client's information will be saved in the bank server and the client got a chequebook from the officer.

Closing Account:

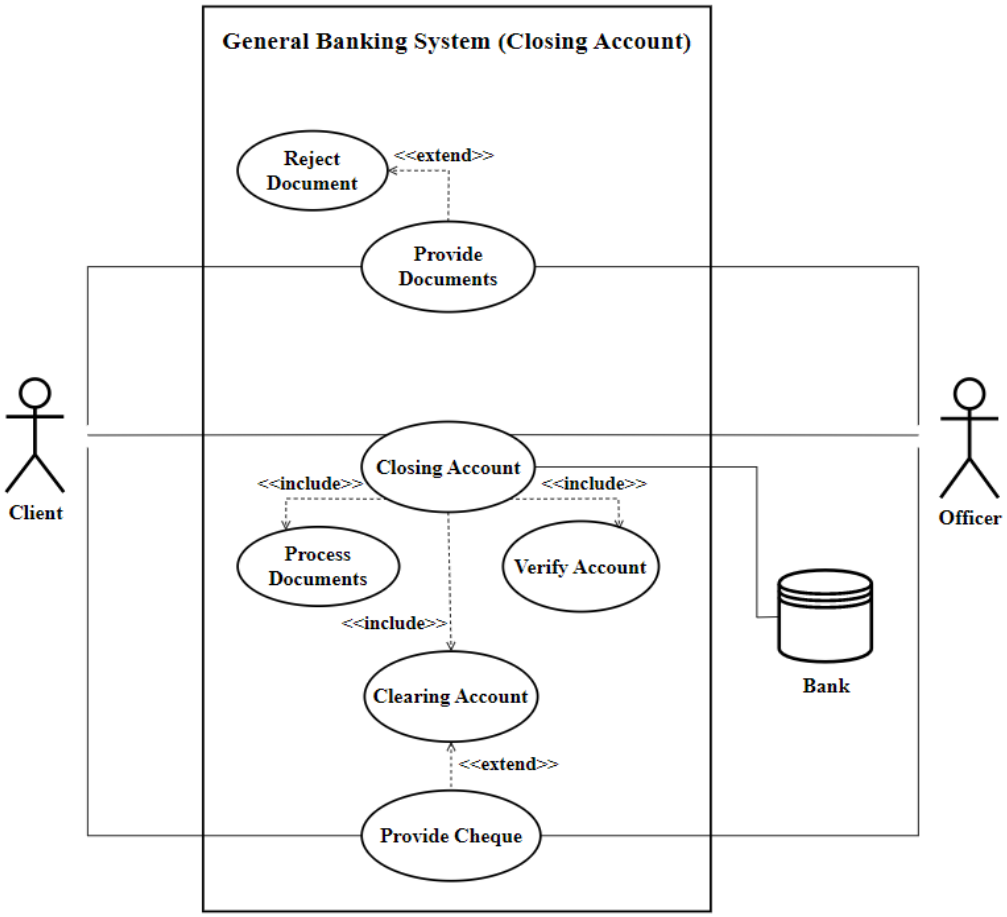

Figure 2 : Use Case Diagram for General Banking (Closing Account)

The Opening Account system diagram has there are three actors and seven use cases. The actors are the Client, Bank, and Officer. The use cases are: Provide Documents, Reject Document, Closing Account, Verify Account, Process Documents, Clearing Account, and Provide Cheque Book. At first, the Client provides the necessary documents to close the account. Then Officer

checks these documents. If there is any wrong, the documents will be rejected. For closing an account, the documents must be verified and processed. Clearing account is also a must. Because if there is any cash left in the account, the officer will provide a cheque to the client. So he/she can withdraw that cash from the cash section.

ii) Cash

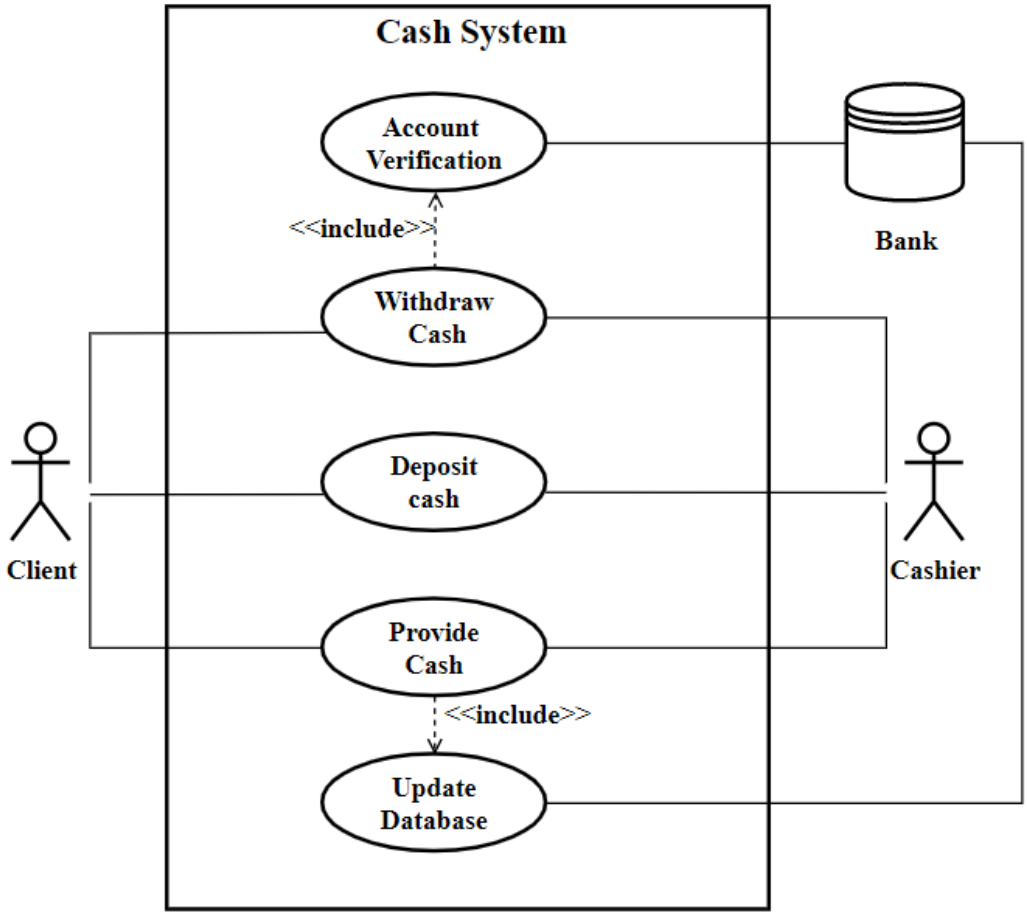

Figure 3: Use Case Diagram for Cash section

In the Cash system diagram, there are three actors and five use cases. The actors are the Client, Bank, and Cashier. The use cases are: Deposit Cash, Withdraw Cash, Account Verification, Provide Cash, and Update Database. If a client wants to withdraw cash, the bank must verify the account. Then the cashier provides cash to the client. At the same time, the bank database also updates. If the client wants to deposit some cash, he/she should provide it to the cashier in the cash section

iii) Loan

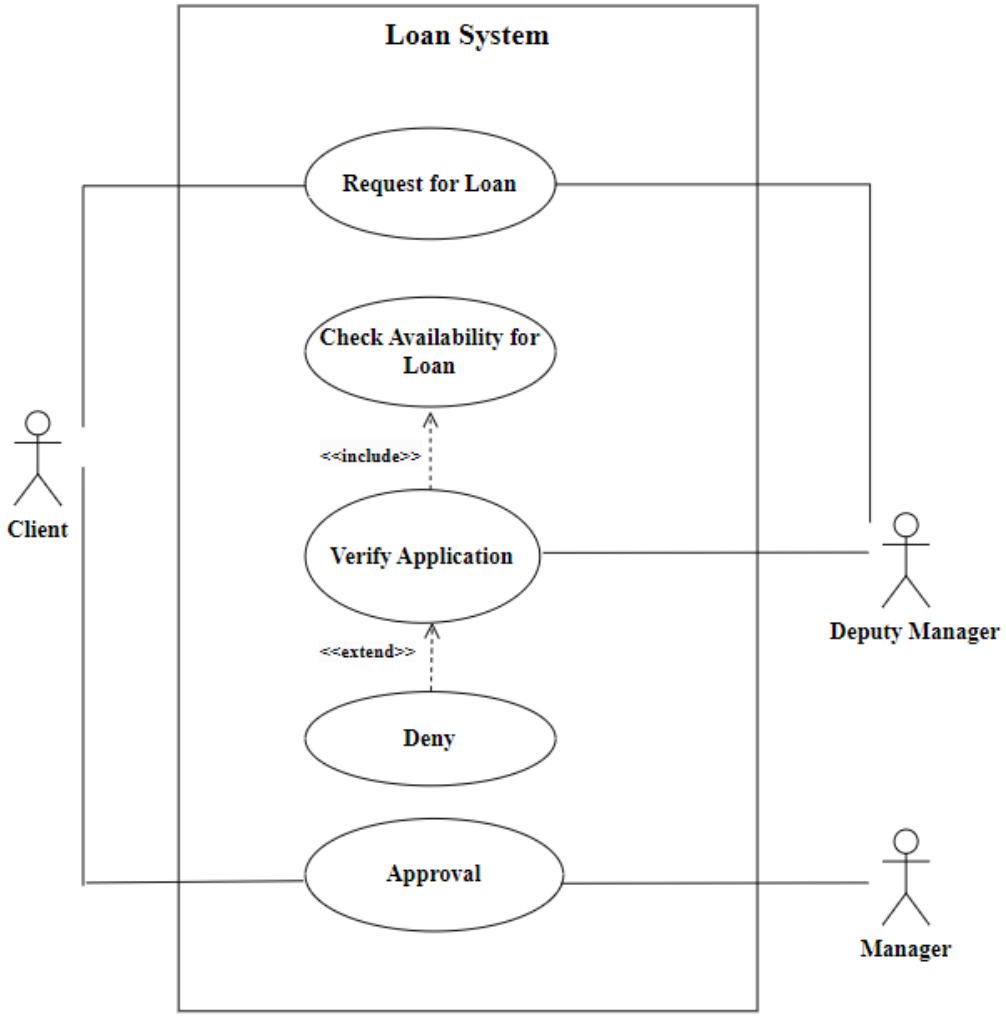

Figure 4: Use Case Diagram for Loan system

In loan system diagram, there are three actors and five use cases. The actors are Client, Deputy Manager, and Manager. Use cases are Request for Loan, Verify Application, Approval, Check Availability for Loan, and Deny. Here, "Verify Application" and "Check Availability for Loan" have include relationship, "Verify Application" and "Deny" have extended relationship. At first, Deputy Manager receives the loan request from the client. Then the Deputy Manager verifies the application by checking the availability of the loan for that client. If the loan is not available to that client, they will deny the request. If all information about the client is appropriate, then the Manager approves the loan.

iv) Remittance:

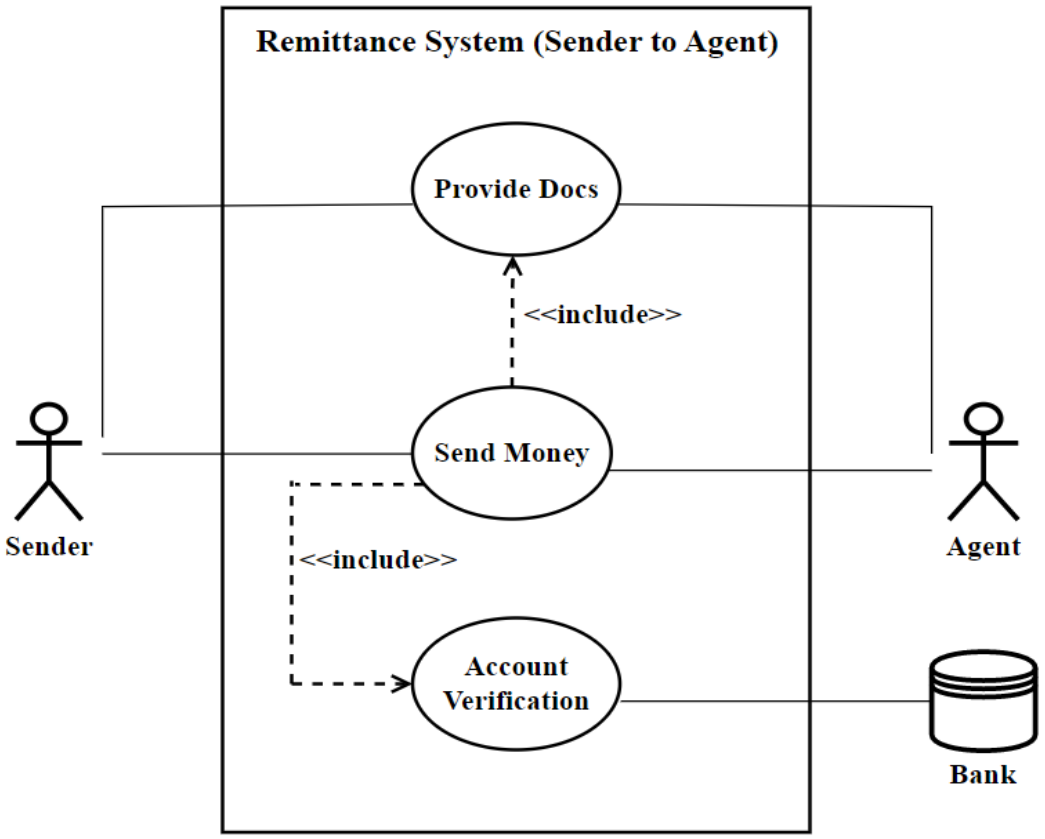

Figure 5: Use Case Diagram for Remittance (Sender to Agent)

There are three actors and three use cases in the Remittance system diagram (Sender to Agent). The actors are Sender, Agent, and Server. Use cases are Provide Documents, Send Money and Account Verification. The sender needs to provide money with the necessary documents first to the Agent. For sending money, the bank server must verify his/her account.

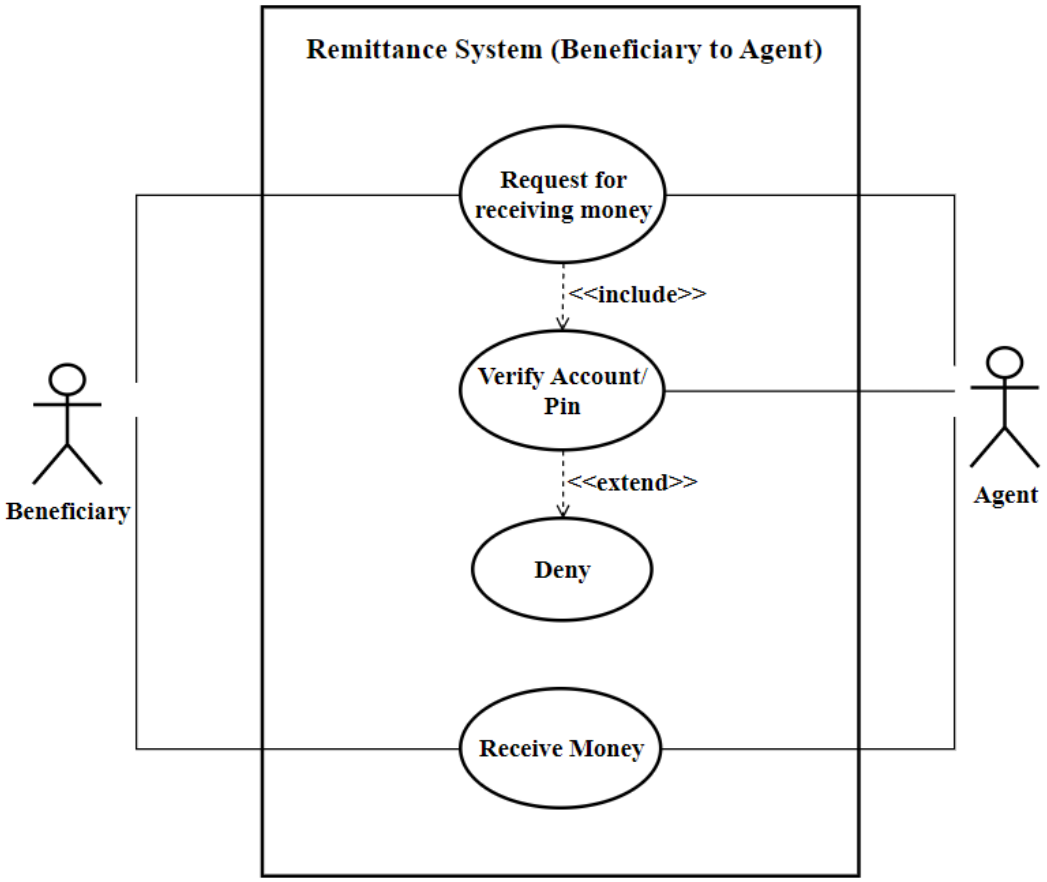

Figure 6: Use Case Diagram for Remittance (Beneficiary to Agent)

In the Remittance system diagram (Beneficiary to Agent), there are two actors and four use cases. The actors are Agent and Beneficiary. Use cases are Request for receiving money, Verifying Account/Pin, Deny, and Receive Money. At first, when a beneficiary wants to withdraw money, he/she must need to verify the account or pin by the agent. Then the beneficiary receives the cash from the Agent. If there is any error, the agent will deny the request.

v) ATM:

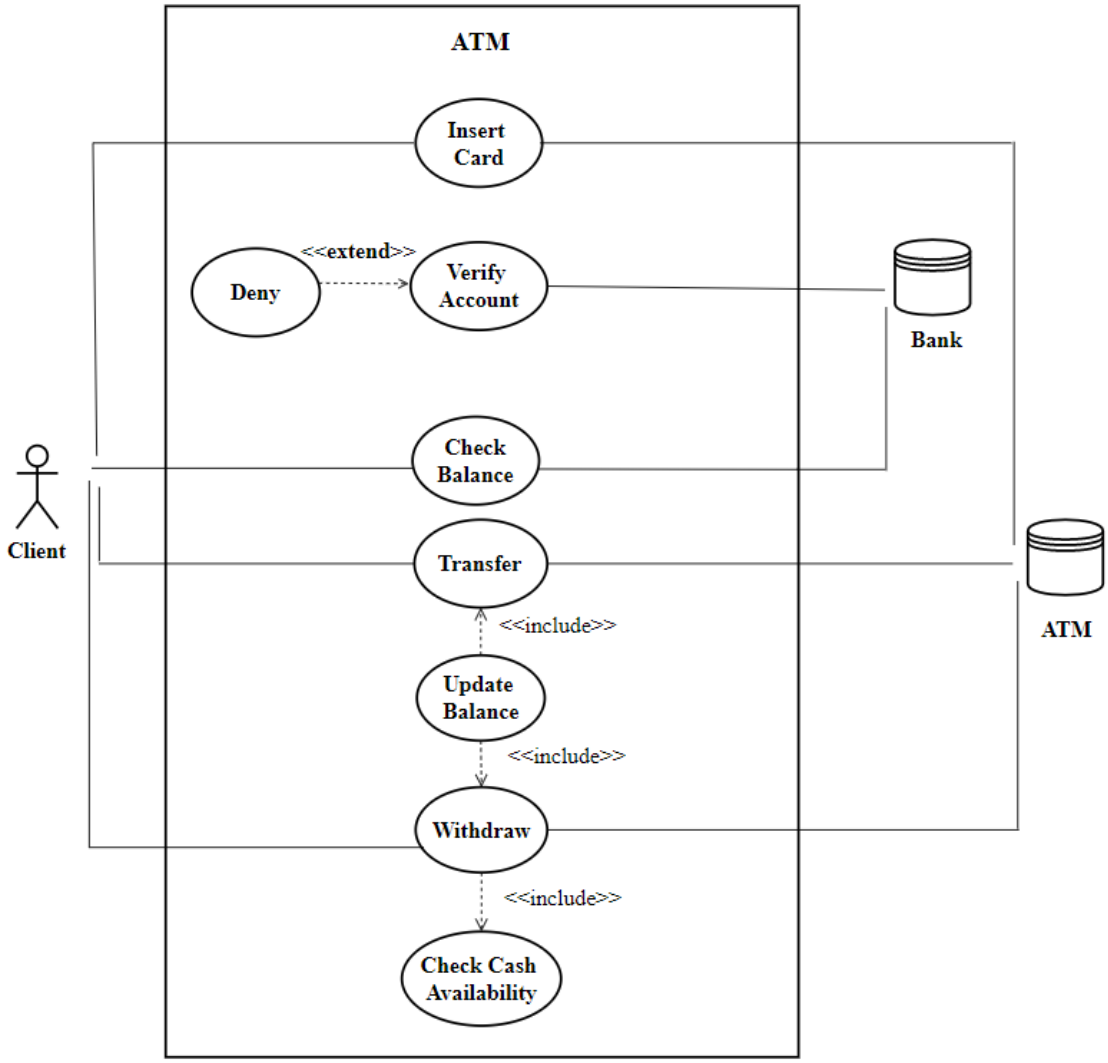

Figure 7: Use Case Diagram for ATM System

In the ATM system diagram, there are three actors and eight use cases. The actors are Client, ATM and Bank. Use cases are Insert Card, Verify Account, Deny, Check Balance, Transfer, Update Balance, Withdraw, Check Cash Availability. At the very first, the client should insert the card into ATM. Then the bank verifies the account. For any error, it gave the card back to the user. The client can check the balance from the bank database. If the client transfers or withdraws money, the balance will be updated. For withdrawing money, ATM must check cash availability.

3. Conclusion:

As we can see, UML design is very important to describe the objects & information of a system and also to show the communication with its users. In this report, we designed the associated

systems of Dutch Bangla Bank Ltd. Noapara Branch using the Use Case diagrams which help us to visualize the bank systems and to implement it logically. We can get a good idea on how the users and the system are interacting with each other, what kind of actions they are performing etc. So UML is very important to understand a system thoroughly.