**CS-255 System Analysis And Design**

**Module 6 Assignment**

**Interpreting UML Diagrams**

**Submitted By:**

HAMNA KHALID

ID: 2902671

[hamna.khalid@snhu.edu](mailto:hamna.khalid@snhu.edu)

**Submitted To:**

PROFESSOR PARUL HIRPARA

p.hirpara@snhu.edu

Southern New Hampshire University

Thursday, October 09, 2025

**Interpreting UML Diagrams**

Unified Modeling Language (UML) diagrams play a critical role in representing and communicating how a system behaves, interacts, and processes information. For this assignment, I analyzed an **ATM cash withdrawal use case** using both a **UML Activity Diagram** and a **UML Sequence Diagram**. These two diagrams depict the same system from different perspectives; one focusing on the flow of activities (the “how”) and the other emphasizing the communication between system components (the “who” and “when”).  
By interpreting the diagrams, identifying deficiencies, and reconstructing an improved version, I gained a stronger understanding of how UML visualizations ensure system logic is both user-friendly and technically accurate.

# **Diagram Interpretation**

The given diagrams describe the **“Withdraw Cash”** use case for an **Automated Teller Machine (ATM)**.

The process begins when a **user inserts their card**, followed by the **ATM prompting for a PIN**. Once entered, the ATM communicates with the **bank system** to verify credentials. Upon successful authentication, the user specifies the desired withdrawal amount. The ATM sends this request to the bank, which checks the account balance and confirms or denies the transaction. If sufficient funds exist, the ATM **dispenses cash**, **prints a receipt**, and **ends the session**.

If authentication fails or funds are insufficient, the system displays an error message and terminates the transaction. Information flows bidirectionally between the user, ATM interface, and bank server, illustrating the real-time interaction and control structure of a typical ATM withdrawal process.

# **Design Analysis**

While the current design captures the core transaction sequence, several logical and functional issues reduce its completeness and real-world accuracy. The following deficiencies were identified, along with potential solutions to enhance system behavior.

## **Lack Of Card Retention For Failed PIN Attempts**

* **Issue**

The current diagram ends after an invalid PIN, without indicating what happens after multiple failed attempts.

* **Improvement**

Add a counter for failed PIN entries. After three unsuccessful tries, the ATM should **retain the card** and end the session to prevent unauthorized access.

## **No Option To Cancel A Transaction**

* **Issue**

Users cannot cancel a transaction once initiated.

* **Improvement**

Introduce a **“Cancel Transaction”** path at any major step (PIN entry, amount selection, confirmation). When chosen, the ATM should safely return the card and terminate the session.

## **Missing Bank Response For Insufficient Balance**

* **Issue**

The diagram briefly mentions “Amount Not Available” but lacks explicit feedback from the bank system.

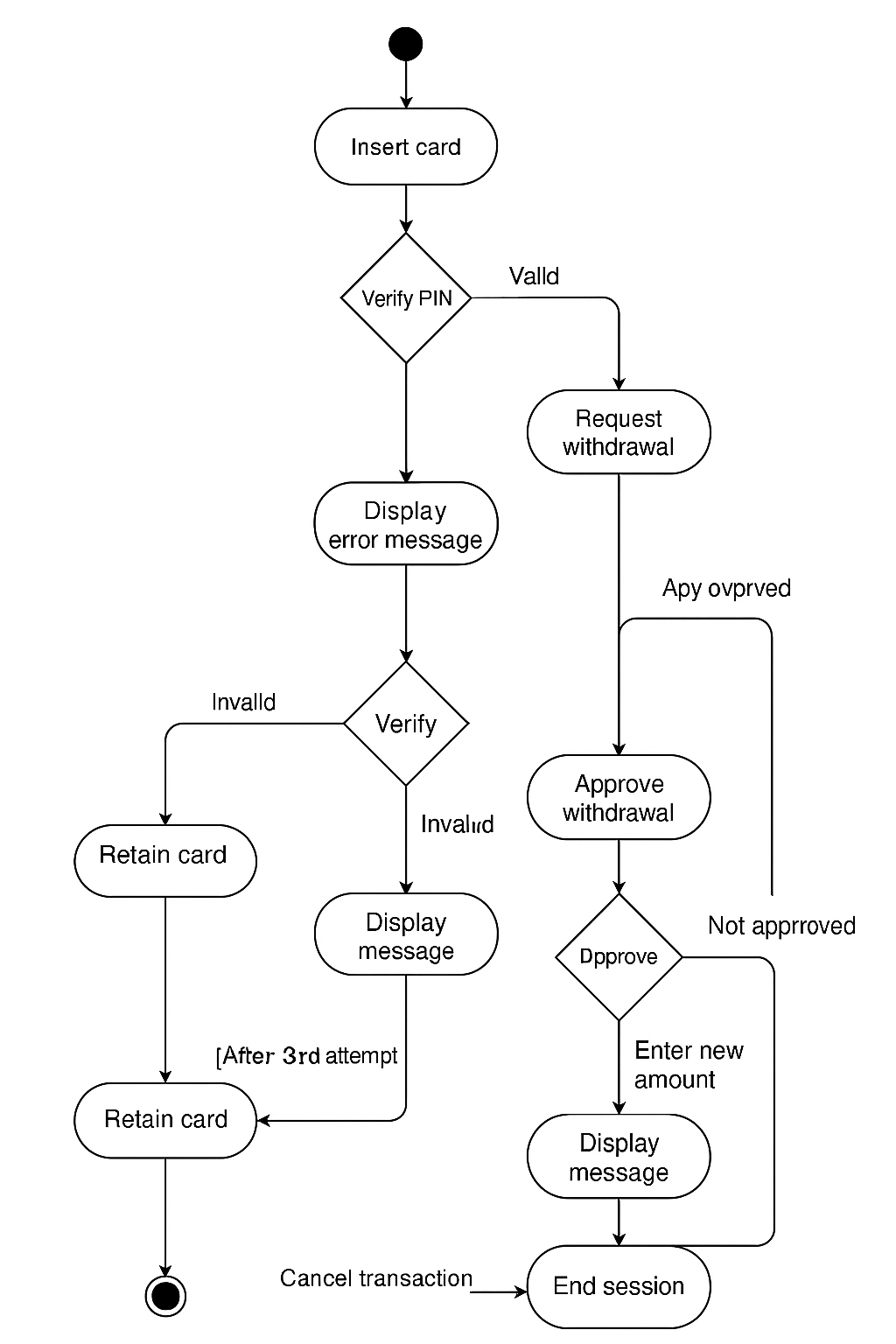
* **Improvement**

Model a **“Reject Transaction”** message from the bank to the ATM, prompting the user to either enter a new amount or cancel the request.

By addressing these areas, the diagram becomes more secure, user-oriented, and functionally realistic.

# **Reconstructed UML Activity Diagram**

The revised **UML Activity Diagram** improves upon the original design by adding missing decision logic, clearer process flow, and robust error handling. Below is a summary of the reconstructed structure:



**Fig 1: UML Activity Diagram**

## **Improved Flow Description**

1. **Insert Card → Enter PIN.**
2. **Verify PIN:**
   * If valid → Proceed to withdrawal.
   * If invalid → Retry up to three times.
   * After 3rd failure → Retain card → End session.
3. **Enter Amount → Send request to Bank.**
4. **Bank verifies funds:**
   * If sufficient → Approve withdrawal.
   * If insufficient → Display message → Prompt for new amount or cancel.
5. **Dispense Cash → Print Receipt → End Session.**
6. **Cancel Transaction (Optional Path):** Can occur at any stage; returns the card and ends the session.

This reconstruction ensures proper **security, flexibility, and user experience**, aligning the diagram with real-world ATM standards.

# **Technical Considerations**

To ensure reliability and maintainability, the following technical features are essential to the ATM system’s design:

* **Role-Based Access Control (RBAC)**

Each actor (user, bank server, admin) must have restricted permissions to ensure secure operation.

* **Error Logging and Recovery**

Every failed transaction or system fault should generate logs for troubleshooting.

* **Data Encryption**

All communication between the ATM and the bank server must use SSL/TLS encryption.

* **Timeout Management**

Inactivity for a specified duration should automatically cancel the transaction and return the card.

# **Conclusion**

Both the UML Activity and Sequence Diagrams are powerful tools for communicating system behavior. The original ATM diagrams successfully illustrate basic functionality but lacked the refinements needed for realistic implementation. By addressing deficiencies, such as missing error handling and user cancellation, the improved design not only enhances **security** and **usability** but also reflects professional-grade modeling standards.  
This exercise strengthened my understanding of how visual modeling bridges the gap between **conceptual design** and **technical development**, helping teams build systems that are reliable, safe, and user-friendly.

# **Acronyms And Full Form**

* **ATM –** Automated Teller Machine
* **UML –** Unified Modeling Language
* **RBAC –** Role-Based Access Control
* **SSL –** Secure Sockets Layer
* **TLS –** Transport Layer Security

# **References**

* Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2020). *Systems Analysis and Design in a Changing World* (8th ed.). Cengage Learning.
* Southern New Hampshire University. (2025). *CS-255 Module Six Assignment: UML Activity and Sequence Diagrams*. SNHU.