# CS 255 Module 6-3 Interpreting UML Diagrams

Tyler Gordon

1. Choose *one* of the deficiencies that you identified. Then **reconstruct a UML activity *or* sequence diagram** to improve functionality. Use the CASE tool Lucidchart to reconstruct the diagram. Be sure to use **proper UML notation and diagram flow**.

In both UML case diagrams, they both show the options a user of a bank has at an atm. The Activity Diagram shows this in a unidirectional format from “Verifying Pin” to “print Receipt” and ending their transaction. This type of depiction is simple and easy to understand because it doesn’t show a communication between the user and the atm as a conversation. This lack of conversation I feel is easier for coders to understand and break down what tasks the atm needs to be implemented. With this said, the sequence diagram is more conversational in it’s implementation of the sequence of events that take place at an ATM. In this diagram the user has a back and forth dialogue with the user as shown by the arrows. For instance, when the user wants withdraw money, they follow the following sequence of events(User process; ATM process; Bank process):

Enter Card -> Ask for PIN -> enter PIN -> Verify PIN -> Pin Valid -> Ask for amount -> Enter Amount -> Dispense Cash -> **END**

In both diagrams an input from the user is received, the PIN, which is the piece of information moving back and forth between the user and the bank . The ATM pushes this to the bank system for verification before proceeding to ask for the amount to be withdrawn from the account. In this current design a few steps are overlooked:

1. The card must be taken back by the user.
2. The account should also have a view balance and deposit option for a fully functional ATM.
3. Sequence diagram is missing the above as well as the generate and print receipt.

A reconstructed diagram would have the following flow:

A screenshot of a computer screen

Description automatically generated