Megan Mitchell
Professor Eppenger
CS 255 System Analysis and Design
11 October 2025

CS 255 Module Six Assignment: UML Activity and Sequence Diagrams

For the purpose of this assignment, I studied the UML activity and sequence diagrams for an ATM system and focused specifically on the use case that dealt with the withdrawal of cash. In my diagram, the main flow starts when the user inserts their PIN and the system verifies it. If the PIN is valid, the user can move on to selecting their transaction type, in this case, a withdrawal. As soon as the user enters incorrect PIN information, the ATM displays an error message and gives them a second chance to enter the correct PIN. The system checks to see if there are sufficient funds in the account when you select "Withdraw Cash.". If there aren't enough funds in the account, then the system displays an "Insufficient Funds" message and the user can decide whether to cancel the transaction or try another one. A receipt is printed and the ATM ends the session if the funds are available and the cash is dispensed. It also contains smaller interactions such as session timeouts, decision points for receipt printing, and proper markers to indicate both the beginning and the end of the process in order to show the complete flow of the process.

It came to my attention while reviewing and redesigning the diagram that there were a few areas that could be improved. I think the first problem with the original version was that it did not clearly show the user how to retry entering their PIN after a failed attempt. The ATM would not work properly without that loop because it would lock out users too quickly, which is not how it works in real life. To make it more accurate, I have added the option to retry the path in my version. I noticed a second issue with the earlier diagram, which was that it didn't account for actual session timeouts, for example, when someone walks away or does not respond after a long period of time. As a way of making the process more realistic, the "Session Timeout" activity was included. Lastly, I added a decision node for whether or not the user wants a printed receipt. This might seem small, but it’s something that happens at almost every ATM and makes the flow more user-friendly.

To fix the issues I found, I reconstructed the UML activity diagram in Lucidchart using proper UML notation. My version now includes all decision points, attached flow lines, and labeled branches such as “Valid PIN,” “Invalid PIN,” “Funds Available,” and “Funds Insufficient.” It also includes clear transitions for displaying messages, printing receipts, and ending the session. By including paths for both errors and normal behavior, the diagram now shows how the ATM interacts with users in a complete and logical way. I made sure that every line is connected properly, that decision diamonds are used where choices are made, and that all

arrows lead to appropriate next steps or an endpoint. This improved layout makes the system flow easier to follow and much closer to how an actual ATM operates.