Lawrence Artl

CS-255-21EW2

Assignment 6-3

December 04, 2021

Interpreting UML Diagrams

1. Write a short, paragraph-length response **interpreting the diagrams**. Be sure to address the following questions:
   1. What use case is being described by these diagrams?
   2. What interactions are involved in this use case? What information is being passed back and forth?
      1. In both of these diagrams, the use case being described is that of a bank customer attempting to withdraw funds from their account via an ATM. In this use case, the transaction is started by the customer interacting with the ATM; it is assumed that this interaction is initiated by the customer inserting their card into the machine. Once the card is inserted, the customer is prompted for their pin, which is then verified. Once the pin is verified, the ATM asks the customer for the amount they wish to withdraw; the customer allegedly inputs this amount on a keypad on the ATM. Cash is then dispensed. Once the cash is dispensed, a receipt is generated and printed for the customer. In each diagram, a customer actor is interacting directly with the ATM. In the second diagram, the actor of “Bank” is added as a third party within the system for pin verification.
2. These diagrams describe one basic design for this use case. **Analyze the design** by identifying *at least two* deficiencies in the logic or functionality of the current design. Think about ways that you could address these deficiencies to improve the logic or functionality of the design.
   1. First, in both diagrams there does not seem to be any logic included that would allow a customer to input an incorrect pin and retry. In the first diagram, an incorrect pin results in immediate system termination, while in the second diagram there is only logic for correct pin input. This is more of a convenience process; the customer could easily reinsert their card and try a new pin. However, this verification process is also missing in the next part of the diagram in which the customer inputs an amount of cash to withdraw. In the first diagram, incorrect input of funds results in a receipt being printed and system exit, with no option to retry. In the second diagram, this verification is missing altogether. This may seem like a simple oversight, except pin verification is present, meaning this was a blatant omission of a very important part of the banking process! Adding both a “pin retry” process as well as a verification of funds and “funds retry” would result in better security of funds as well as convenience to the customer.
   2. As an aside, the convenience factor may not seem very important in the grand scheme of things; however, this could result in less time spent at the ATM. If anyone has ever sat in the drive through at the bank while the person in front of them seemingly fumbles with what most would describe as the simplest of processes available to the average banking customer (interacting with an ATM), then they would understand the frustration. Making the ATM interactions somewhat easier could alleviate this frustration, making the banking experience better overall.
3. 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**.

|  | 1. For this section, I chose to modify the first diagram. The modification I chose to implement was a “retry” option for both the pin entry and the “funds-to-withdraw” process. I additionally added a logic pathway for termination of the transaction early (at pin input) lest a customer (or bad-actor, more likely) attempts to retry inputting the pin of a card an infinite number of times. This would allow the transaction to cancel itself after a certain number of incorrect attempts. |
| --- | --- |