**1. Discover ambiguities or omissions in the following statement of requirements for part of a ticket-issuing system.**

***An automated ticket issuing system sells rail tickets. Users select their destination and input card details and a personal identification number. The rail ticket is issued and their card account is charged. When the user presses the start button, a menu display of potential destinations is activated, along with a message to the user to select the destination. Once a destination is selected, users are required to input the card details. Its validity is checked and the user is then requested to input the personal identification. When the card transaction is validated, the ticket is issued.***

Customers and other stakeholders often have a difficult time communicating a complete set of system requirements to development engineers (Sommerville, 2011). Having read the above statement of requirements for the ticket-issuing system, there are several ambiguities and omissions that need to be clarified in order to ensure that the system project will not be at risk for failure.

The first ambiguity is whether the automated ticket issuing system is web portal or a ticketing kiosk machine at a railway station. This will make a difference in how the ticket is issued later. It can be issued as a printable pdf ticket in the case of a web portal or it can be an actual ticket that is printed and/or punched via the kiosk machine.

Regarding the destination selection, there are often transfers between legs of a long railway journey. Will the system account for transfers or layovers? More specifically, if the destination selection can be for transfers as well as non-stop routes, then will separate tickets be issued for each leg of the journey or will one ticket accommodate multiple stops?

Another point for clarification is the fares. Some railways charge more during busy rush hour and they may sell discounted tickets at non-peak hours. Many railways also have different fares for travel in coach, business, and first class. Lastly, will the system accommodate round trip tickets and vary their pricing accordingly?

The requirements document will be more complete once all the issues mentioned above have been clarified.

***2.* For thesame scenario, write the set of non-functional requirements.**

Non-functional requirements encompass the “constraints on the services or functions offered by the system” (Sommerville, 2011). They are contrasted with the functional requirements which dictate how the system should work. Ironically, the non-functional requirements may be more critical than the functional requirements. Here are some examples of non-functional requirements for the automated ticketing system described above.

The first consideration is that of system reliability. The system will not work well if power to the kiosks is intermittent or if the website portal for the the automated ticketing system is down. In order for the system to work, the kiosks need consistent power and/or the website needs to be reliable.

Next, the card/paper storage and toner levels are a consideration. If the kiosk cannot issue the ticket because the toner levels are low or the ticket card repository is empty, the automated ticketing system will not work. Similarly if there is a paper jam or if the credit card reader is broken, the system will not issue a ticket.

Finally, imagine a situation in which the trains are out of service due to inclement weather, accidents, or some other reason. Under these circumstances, we might not want the customer to be able to make a ticket purchase. Even if the system itself is operational, the system will not be “working” if the customers are not able to use the tickets they buy.

All of the above mentioned issues can cause an automatic ticketing system to fail. Yet, none of these issues have anything to do with how the ticketing software itself functions. This is the hallmark of non-functioning requirements that still need to be met for the system to work (Sommerville, 2011).

**3. Using your knowledge of an ATM, develop a set of Use Cases that could serve as a basis for understanding the requirements of an ATM.**

Use-cases are a technique in requirements engineering where the actors in an interaction are identified and the interaction itself is described (Somerville, 2011). A complete set of use cases ought to cover all of the possible interactions with the system in question.

The first actor is the account holder. This person interacts with the ATM for several reasons: to view their account balance, to deposit funds, or to withdraw funds. These people often acquire a transaction receipt or slip at the end of their transaction.

Another actor is the bank personnel. This person empties and restocks the ATM. This person will restock the ATM with currency which account holders may withdraw. The bank personnel may also restock the paper and ink for the receipts/slips. He or she may also restock the deposit envelopes. This person will collect any deposits that have been made to the ATM. The bank personnel will also generate monthly reports that will include all of the account holder’s ATM transactions.

Lastly, security personnel and police will be able to access the ATM’s security videos in the case of vandalism or theft.

ATM system requirements can be discovered by breaking down all of the interactions with an ATM by user. By creating these use-cases we have identified the various actors and their interactions with an ATM.



**4. Draw the sequence diagram for the same system (ATM) and explain why it might be necessary to have both Activity Diagram and Sequence Diagrams when modeling the behaviour of a system**.

Sequence diagrams show the sequence of events for a particular system (Sommerville, 2011). Specifically, these diagrams show the interactions between various actors, the system itself, and even components of the system. This can add some detail to the use-cases since the order of events can be understood. Here is the sequence diagram for the ATM use-cases described above.



When modeling the behaviour of an ATM system like the one mentioned above, it might be useful to not only have a sequence diagram, but to also include an activity diagram. An activity diagram shows the various actions that are involved in data processing or any process (Sommerville, 2011). This can be helpful when a business system is primarily data driven as activity diagrams show the sequence of actions involved in either processing data that has been input into the system or in generating data output. In our ATM example, this would be particularly helpful in understanding the processes for generating the monthly reports and/or the transaction slips.

**References**

Sommerville, I. (2011) "Chapter 4 - Requirements Engineering: Lecture 1-3.” Software Engineering, Ed. 9. Retrieved from<http://my.uopeople.edu/mod/resource/view.php?id=57336>

Sommerville, I. (2011) "Chapter 5 - System Modeling: Lecture 1-2.” Software Engineering, Ed. 9. Retrieved from<http://my.uopeople.edu/mod/resource/view.php?id=57337>