Requirements Engineering – Q&A

1. Software system requirements are often classified as functional or non-functional requirements. Explain and compare them.

Software system requirements are often classified as functional or non-functional requirements.

1. Functional requirements: The functional requirements for a system describe what the system should do. These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

2. Non-functional requirements: Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users. These are constraints on the services or functions offered by the system. The may relate to emergent system properties such as reliability, response time. And memory use. They may also define constraints on the system implementation, such as the capabilities of I/O devices or the data representations used in interfaces with other systems. Non-functional requirements often apply to the system as a whole rather than individual system features or services.

In reality, the distinction between different types of requirements is not as clear-cut as these simple definitions suggest. A user requirement concerned with security, such as a statement limiting access to authorized users, may appear to be a non-functional requirement. However, when developed in more detail, this requirement may generate other requirements that are clearly functional, such as the need to include user authentication facilities in the system.

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

An automated ticket machine sells rail tickets. Users select their destination and input a credit card and a personal identification number. The rail ticket is issued and their credit card account 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 a destination and the type of ticket required. Once a destination has been selected, the ticket price is displayed and customers are asked to input their credit card. Its validity is checked and the user is then asked to input their personal identifier (PIN). When the credit transaction has been validated, the ticket is issued.

Ambiguities and omissions include:
1. Can a customer buy several tickets for the same destination together or must they be bought one at a time?
2. Can customers cancel a request if a mistake has been made?
3. How should the system respond if an invalid card is input?

4. What happens if customers try to put their card in before selecting a destination (as they would in ATM machines)?

5. Must the user press the start button again if they wish to buy another ticket to a different destination?
6. Should the system only sell tickets between the station where the machine is situated and direct connections or should it include all possible destinations?

5. Ideally, the functional requirements specification of a system should be both complete and consistent. Do you think it is possible to achieve requirements consistency and completeness for large and complex software systems?

Ideally, the functional requirements specification of a system should be both complete and consistent. Completeness means that all services and information required by the user should be defined. Consistency means that requirements should not be contradictory.
Practically, it is not possible to achieve requirements consistency and completeness for large and complex software systems due to the following reasons.

- It is easy to make mistakes and omissions when writing specifications for large, complex systems.
- Large systems have many stakeholders, with different backgrounds and expectations. Stakeholders are likely to have different—and often inconsistent— needs. These inconsistencies may not be obvious when the requirements are originally specified, and the inconsistent requirements may only be discovered after deeper analysis or during system development.

6. Using your knowledge of how an ATM is used, develop a set of use cases that could serve as a basis for understanding the requirements for an ATM system.

There are a variety of different types of ATM so, obviously, there is not a definitive set of use cases that could be produced. However, I would expect to see use cases covering the principal functions such as withdraw cash, display balance, print statement, change PIN and deposit cash. The use case description should describe the actors involved, the inputs and outputs, normal operation and exceptions.

Withdraw cash:
Actors: Customer, ATM, Accounting system
Inputs: Customer's card, PIN, Bank Account details

Outputs: Customer's card, Receipt, Bank account details
Normal operation: The customer inputs his/her card into the machine.
He/she s promoted for a PIN which is entered on the keypad. If correct, he/she is presented with a menu of options. The Withdraw cash option is selected. The customer is promoted with a request for the amount of cash required and inputs the amount. If there are sufficient funds in his/her account, the cash is dispensed, a receipt if printed and the account balance is updated. Before the cash is dispensed, the card is returned to the customer who is prompted by the machine to take their card.
Exception: Invalid card. Card is retained by machine; Customer advised to seek advice.
Incorrect PIN. Customer is request to rekey PIN. If incorrect after 3 attempts, card is retained by machine and customer advised to seek advice.
Insufficient balance Transaction terminated. Card returned to customer.

Display balance:
Actors: Customer, ATM, Accounting system
Inputs: Customer's card, PIN, Bank Account details

Outputs: Customer’s card
Normal operation: The customer authenticates using card and PIN as in Withdraw cash and selects the Display Balance option. The current balance of their account is displayed on the screen. The card is returned to the customer.
Exception: Invalid card. As in Withdraw cash
Incorrect PIN. As in Withdraw cash

Print statement:
Actors: Customer, ATM, Accounting system
Inputs: Customer’s card, PIN, Bank Account details
Outputs: Customer’s card, Printed statement
Normal operation: The customer authenticates using card and PIN as in Withdraw cash and selects the Print statement option. The last five customer.
Exception: Invalid card. As in Withdraw cash
Incorrect PIN. As in Withdraw cash

Change PIN:
Actors: Customer, ATM
Inputs: Customer’s card, PIN
Outputs: Customer’s card
Normal operation: The customer authenticates as in Withdraw cash and selects the Change PIN option. He/she is prompted twice to input the new PIN. The PINS input should be the same. The customer’s PIN is encrypted and stored on the card. Card returned to customer.
Exception: Invalid card. As in Withdraw cash.
Incorrect PIN. As in Withdraw cash.
PINS do not match. The customer is invited to repeat the process to reset his/her PIN.

Deposit cash:
Actors: Customer, ATM, Accounting system
Inputs: Customer’s card, PIN, Bank Account details, Cash to be deposited
Outputs: Customer’s card, Receipt
Normal operation: The customer authenticates as in Withdraw cash and selects the Deposit option. The customer is promoted with a request for the amount of cash to be deposited and inputs the amount. He or she is then issued with a deposit envelope in which they should put the cash then return it to the machine. The customer’s account balance is updated with the amount deposited but this is marked as uncleared funds and is not cleared until checked. A receipt is issued and the customer’s card is returned.
Exception: Invalid card. As in Withdraw cash.
Incorrect PIN. As in Withdraw cash.
No cash deposited within 1 minute of envelope being issued. Transaction terminated. Card returned to customer.