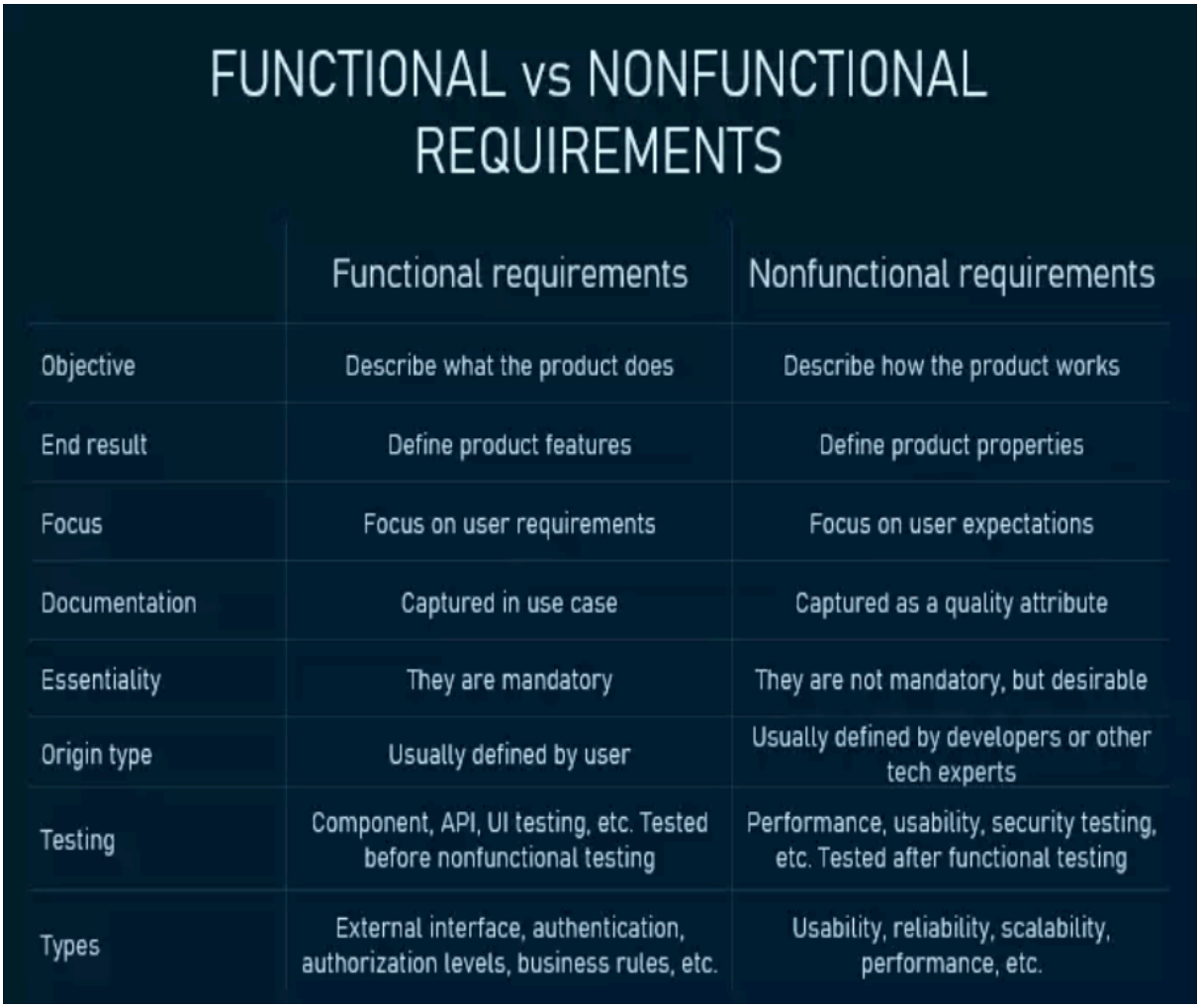
Experiment – 8: Define the functional and non-functional requirements of the system to be automated by using Use cases and document in SRS document.

Functional Requirements?

In software engineering, a functional requirement defines a system or its component. It describes the functions software must perform. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Non-Functional Requirements ?

A non-functional requirement defines the quality attribute of a software system. They represent a set of standards used to judge the specific operation of a system. A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.



Functional requirements can be classified according to different criteria. For example, we can group them on the basis of the *functions* a given feature must perform in the end product. Of course, they would differ depending on the product being developed, but for the sake of an example, the types of functional requirements might be

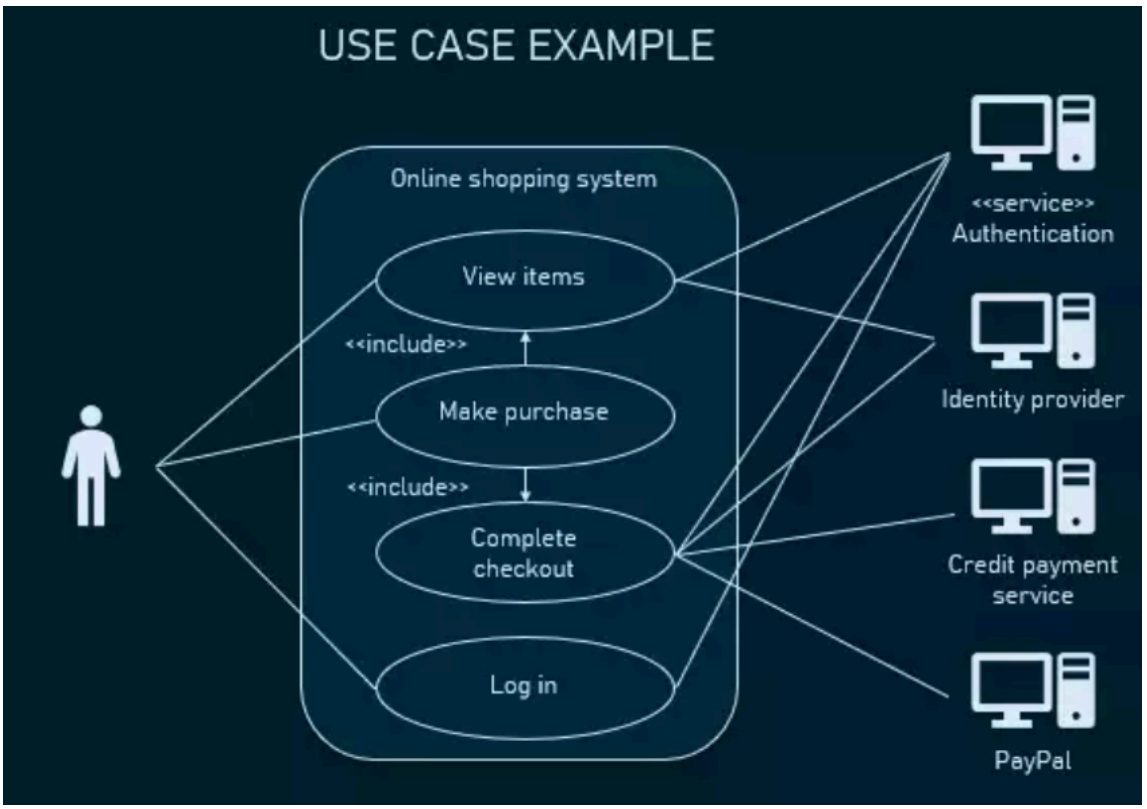
- Authentication
- Authorization levels
- Compliance to laws or regulations
- External interfaces
- Transactions processing
- Reporting
- Business rules, etc

A **use case diagram** doesn't contain a lot of details. It shows a high-level overview of the relationships between actors, different use cases, and the system.

The use case diagram includes the following main elements:

- **Use cases.** Usually drawn with ovals, use cases represent different interaction scenarios that actors might have with the system (*log in, make a purchase, view items,* etc.).
- **System boundaries.** Boundaries are outlined by the box that groups various use cases in a system.
- **Actors.** These are the figures that depict external users (people or systems) that interact with the system.
- **Associations.** Associations are drawn with lines showing different types of relationships between actors and use cases.

Example:



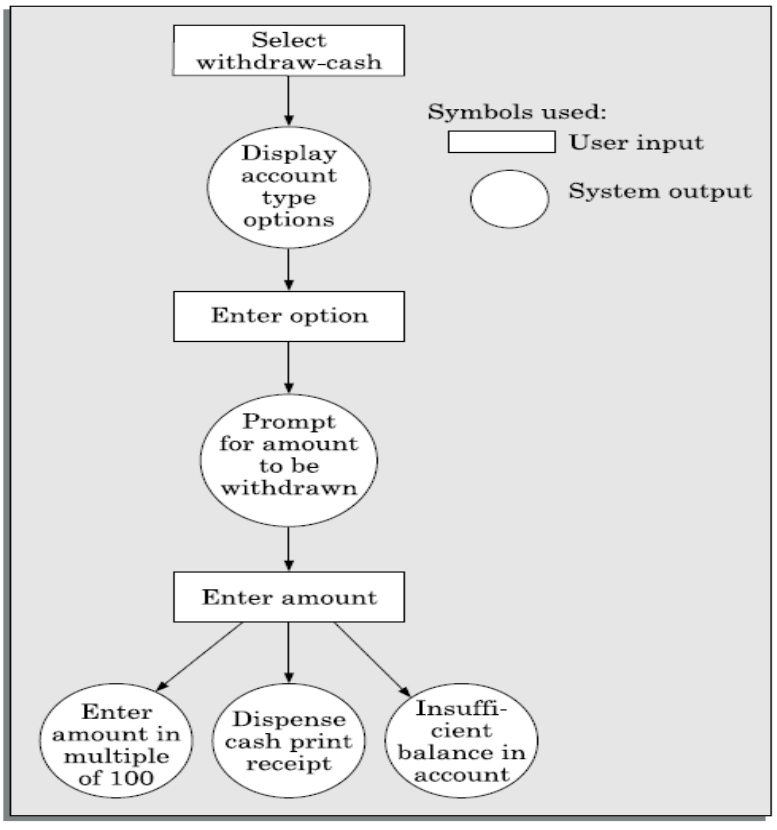
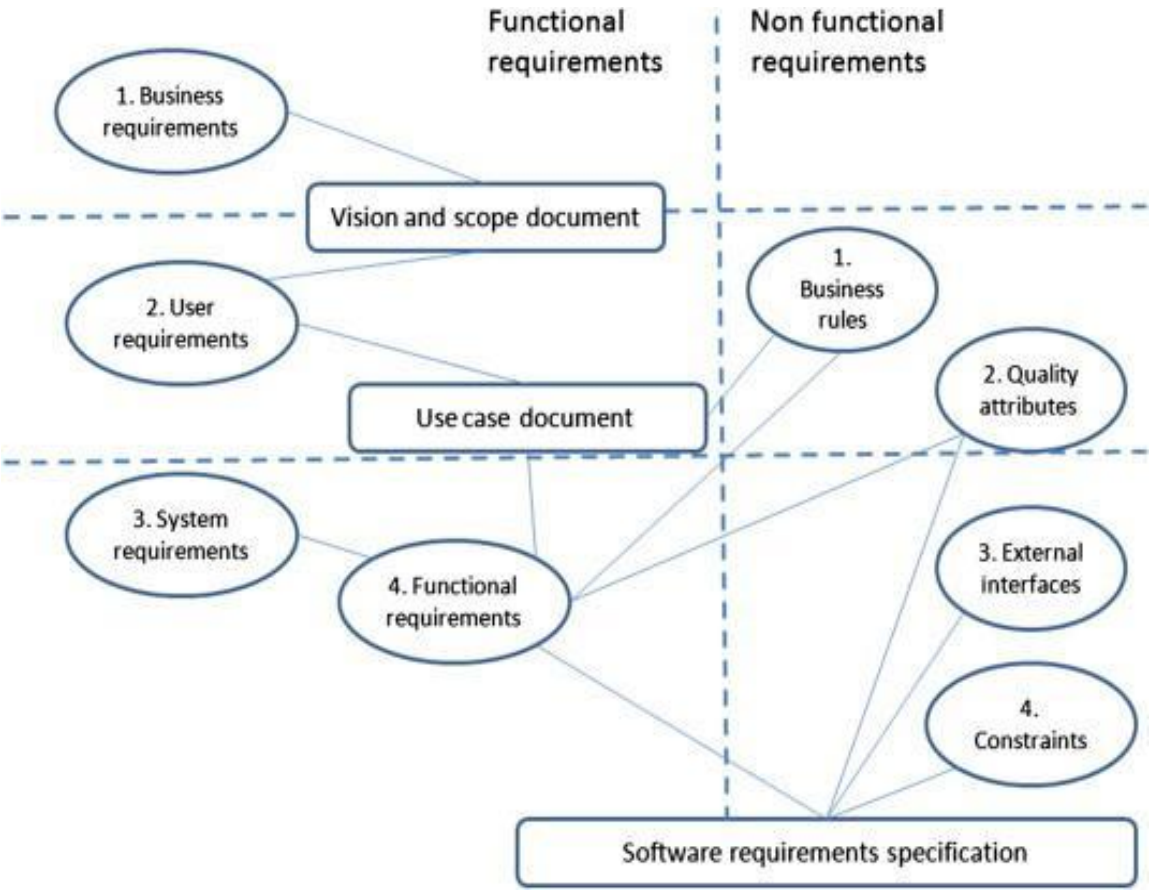


FIGURE User and system interactions in high-level functional requirement.

Non-functional requirements: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.
They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility



EXAMPLE: 1 **Withdraw cash from ATM**

R.1: Withdraw cash

Description: The withdraw-cash function first determines the type of account that the user has and the account number from which the user wishes to withdraw cash. It checks the balance to determine whether the requested amount is available in the account. If enough balance is available, it outputs the required cash, otherwise it generates an error message.

R.1.1 : Select withdraw amount option

Input: "Withdraw amount" option selected

Output: User prompted to enter the account type

R.1.2 : Select account type

Input: User selects option from any one of the following—savings/checking/deposit.

Output: Prompt to enter amount

R.1.3 : Get required amount

Input: Amount to be withdrawn in integer values greater than 100 and less than 10,000 in multiples of 100.

Output: The requested cash and printed transaction statement.

Processing: The amount is debited from the user's account if sufficient balance is available, otherwise an error message displayed.

EXAMPLE: 2 **Search book availability in library**

R.1: Search book

Description Once the user selects the search option, he would be asked to enter the keywords. The system would search the book in the book list based on the key words entered. After making the search, the system should output the details of all books whose title or author name match any of the key words entered. The book details to be displayed include: title, author name, publisher name, year of publication, ISBN number, catalog number, and the location in the library.

R.1.1 : Select search option

Input: "Search" option

Output: User prompted to enter the key words

R.1.2 : Search and display

Input: Key words

Output: Details of all books whose title or author name matches any of the key words entered by the user. The book details displayed would include—title of the book, author name, ISBN number, catalog number, year of publication, number of copies available, and the location in the library.

Processing: Search the book list based on the key words:

R.2: Renew book

Description: When the "renew" option is selected, the user is asked to enter his membership number and password. After password validation, the list of the books borrowed by him are displayed. The user can renew any of his borrowed books by indicating them. A requested book cannot be renewed if it is reserved by another user. In this case, an error message would be displayed.

R.2.1 : Select renew option

State: The user has logged in and the main menu has been displayed.

Input: "Renew" option selection.

Output: Prompt message to the user to enter his membership number and password.

R.2.2 : Login

State: The renew option has been selected.

Input: Membership number and password.

Output: List of the books borrowed by the user is displayed, and user is prompted to select the books to be renewed, if the password is valid. If the password is invalid, the user is asked to re-enter the password.

Processing: Password validation, search the books issued to the user from the borrower's list and display.

Next function: R.2.3 if password is valid and R.2.2 if password is invalid.

R.2.3 : Renew selected books

Input: User choice for books to be renewed out of the books borrowed by him.

Output: Confirmation of the books successfully renewed and apology message for the books that could not be renewed.

Processing: Check if anyone has reserved any of the requested books. Renew the books selected by the user in the borrower's list, if no one has reserved those books.

Non-functional requirements

N.1: Database: A data base management system that is available free of cost in the public domain should be used.

N.2: Platform: Both Windows and UNIX versions of the software need to be developed.

N.3: Web-support: It should be possible to invoke the query book functionality from any place by using a web browser.

Observation: Since there are many functional requirements, the requirements have been organised into four sections: Manage own books, manage friends, manage borrowed books, and manage statistics. Now each section has less than 7 functional requirements. This would not only enhance the readability of the document, but would also help in design