EXPERIMENT - 2

Software Engineering Lab

Aim

Do requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 2

Aim:

Do requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system.

Theory:

Software Requirement Specification (SRS) is a document that describes the requirements of a computer system from the user's point of view. An SRS document specifies: The required behavior of a system in terms of: input data, required processing, output data, operational scenarios and interfaces. The attributes of a system including: performance, security, maintainability, reliability, availability, safety requirements and design constraints.

A well-designed, well-written SRS accomplishes four major goals:

- It provides feedback to the customer. An SRS is the customer's assurance that the development organization understands the issues or problems to be solved and the software behavior necessary to address those problems. Therefore, the SRS should be written in natural language (versus a formal language, explained later in this article), in an unambiguous manner that may also include charts, tables, data flow diagrams, decision tables, and so on.
- It decomposes the problem into component parts. The simple act of writing down software requirements in a well-designed format organizes information, places borders around the problem, solidifies ideas, and helps break down the problem into its component parts in an orderly fashion.
- It serves as an input to the design specification. As mentioned previously, the SRS serves as the parent document to subsequent documents, such as the software design specification and statement of work. Therefore, the SRS must contain sufficient detail in the functional system requirements so that a design solution can be devised.
- It serves as a product validation check. The SRS also serves as the parent document for testing and validation strategies that will be applied to the requirements for verification.

SRSs are typically developed during the first stages of "Requirements Development," which is the initial product development phase in which information is gathered about what requirements are needed--and not. This information-gathering stage can include onsite visits, questionnaires, surveys, interviews, and perhaps a return-on-investment (ROI) analysis or needs analysis of the customer or client's current business environment. The actual specification, then, is written after the requirements have been gathered and analyzed.

Performance Instruction:

Use IEEE SRS Document Template for drafting SRS for your system.

Content

1. Introduction
    1.1. Purpose
    1.2. Scope
    1.3. Definitions, Acronyms, and Abbreviations
    1.4. References
    1.5. Overview
2. Overall Description
    2.1. Product Perspective
        2.1.1. System Interfaces
        2.1.2. User Interfaces
        2.1.3. Hardware Interfaces
        2.1.4. Software Interfaces
        2.1.5. Communication Interfaces
        2.1.6. Memory Constraints
        2.1.7. Operations
        2.1.8. Site Adaptation Requirements
    2.2. Product Functions
    2.3. User Characteristics
    2.4. Constraints
    2.5. Assumptions and dependencies
    2.6. Apportioning of requirements
3. Specific Requirements

B2C System SRS

1. INTRODUCTION

This document aims at defining the overall software requirements for ‘BUYER TO CONSUMER SYSTEM’. Efforts have been made to define the requirements exhaustively and accurately. The final product will be having only features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties involved in developing/testing/implementing/ using this product. In case it is required to have some additional features a formal change request will need to be raised and subsequently a new release of this document and/or product will be produced.

1.1 PURPOSE

This specification document describes the capabilities that will be provided by the software application ‘BUYER TO CONSUMER SYSTEM’. It also states the various required constraints by which the system will abide. The intended audiences for this document are the development team, testing team and end users of the product.

1.2 SCOPE

The software product 'BUYER TO CONSUMER SYSTEM ' is an application that will be used for buying and delivering products put up by a local seller or business, by a customer. The application will manage the information about products put up by a seller on our website, which can be bought by an interested customer and delivered by an available delivery man . The seller can fill up an online request with all the details for their parcel, get the parcel picked up and get the cost of the delivery instantly when the pickup person arrives and a tracking ID is shared with both the seller and buyer to check the status of delivery, which is updated by the delivery agent. A communication system is also implemented for communication between each party. This application will greatly simplify and speed up the delivery and management of the delivery and product process of this system.

1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

The following abbreviation has been used throughout this document

B2C : Buyer to Consumer

1.4 REFERENCES

(i) GitHub Link - https://github.com/syedareehaquasar/B2C-System

(ii) Deployment Link

1.5 OVERVIEW

The rest of this SRS document describes the various system requirements, interfaces, features and functionalities in detail.

2.OVERALL DESCRIPTION

This would be a B2C System with 3 interfaces: an interface of an individual seller or small business, an interface of a customer, and an efficient delivery system.

Current B2B systems don't support intermediary help for product delivery or transfer so we would provide aid for the same. This would also create revenue and employee opportunities

We aim at automating many operations performed at delivering the item to the customer. In many b2c services that already exist trade is carried out entirely at personal levels, with the risks of leaking of personal information and product not reaching the customer.

This can be solved if the seller uses one of the partner courier services. The seller can fill up an online request with all the details for their parcel, get the parcel picked up at home get the cost of the delivery instantly when the pickup person arrives and a tracking ID is shared with both the seller and customer. A feedback system can also be implemented on both the sender and receiver end.

2.1 PRODUCT PERSPECTIVE

The application will be a browser-based, self-contained and independent software product.

2.1.1 SYSTEM INTERFACES

None

2.1.2 USER INTERFACES

The application will have a user-friendly and web based interface.

Following users will interact with the application:

- Buyer
- Seller
- Deliverer

Following screens will be provided:

(i) A signup screen for registering as a buyer or seller, a Login screen for entering the username, password and logging in as buyer, seller or a delivery agent) Access to different screens will be based upon the role of the user.

(ii) For buyers, there will be a screen which will allow them to view the products put up by the seller. This screen consists of features like viewing the product, adding them to cart, tracking an ordered product. There will be a payment system implemented for buyers so that they can pay for their order at the time of checkout.

(iii) For Sellers, there will be a screen that will allow them to add products, manage them and remove them. There will be an interface that will allow the seller to assign a delivery agent of his choice.

(iv) For the delivery agent , there will be a screen where the orders which are assigned to him, are available. The information about the order, address and contact information of seller and buyer is also shared with the delivery agent . The delivery agent can generate a tracking ID which is shared with both the buyer and seller and the delivery agent can update the status of the shipping from his side.

(v) There will be a communication system between each party which will allow us to resolve issues during phases of order and delivery.

(vi) There will be a payment portal that supports many forms of payment like UPI, credit card etc. Cash on delivery option is also provided.

2.1.3 HARDWARE INTERFACES

1. Access to the internet is required to access the API Interfaces.
2. A computer or mobile device
3. Screen resolution of at least 800x600-required for proper and complete viewing of the screen. Higher resolution would not be a problem.

2.1.4 SOFTWARE INTERFACES

1. Any window -based operating system.
2. Any Browser to access the website based application.
3. Visual Studio Code —for coding /developing the software.
4. Libraries and programming languages such as react, nodejs, bootstrap, JavaScript and few others.

Software mentioned in pts. 3. and 4. above will be required only for development of the application. The final application will be packaged as an independent setup program that will be delivered to the client and can be accessed via any web browser.

2.1.5 COMMUNICATION INTERFACES

The B2C system shall use HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

2.1.6 MEMORY CONSTRAINTS

For running the program no memory is needed.

At least 64 MB RAM and 4 GB space on the hard disk will be required for development of the program.

2.1.7 OPERATIONS

This product release will not cover any automated housekeeping aspects of the database. The DB used is mongoDB. The DB at the seller side will be responsible for adding, modifying or deleting product data local to their shop. Database backup and recovery will be handled by the DBA(mongoDB server).

2.1.8 SITE ADAPTATION REQUIREMENTS

The terminals at the client site will have to support the hardware and software interface specified in the above section.

2.2 PRODUCT FUNCTIONS

The system will allow access only to registered users with specific roles. Depending upon the user's role he/she will be able to access only specific modules of the system.

A summary of the major functions that the software will perform:

(i) **A SIGNUP facility** for registering buyer and seller

(ii) **A LOGIN facility** for logging in as a buyer, customer or as delivery agent.

(iii) **Users (with role Seller)** will be able to add/modify/delete products on their respective shops which can be viewed by Buyer. For a given product in a shop, the amount of stock available, price of the product is set by the seller.

(iv) **Users (with role Seller)** will be able to assign the delivery agent to an ordered product by the buyer.

(v) **Users (with role Buyer)** will be able to view different shops of sellers and the products that each shop is offering. Buyer can select the amount of any product they want to buy and add them to cart.

(vi) **Users (with role Buyer)** will be able to pay for their order when checking out their cart via two modes available : Online Payment and Cash on Delivery. Another Interface is available for Online Payment, where payment can be done using various methods.

(vii) **Users (with role Delivery Agent)** will be able to get assigned to the delivery of a product by the seller. They get access to the details required for delivery.

(viii) **Users (with Delivery Agent)** will be able to update the status of delivery to either : Delivering or Delivered.

(ix) **Users (with role Buyer, Seller and Delivery Agent)** will be able to communicate with each other.

2.3 USER CHARACTERISTICS

(i) Educational Level : At least a graduate should be comfortable with English.

(ii) Experience : Should be well informed about the working of an e-commerce website and interacting with it.

(iii) Technical expertise : Should be comfortable with general purpose applications of computers.

2.4 CONSTRAINTS

1. Since the DBMS being used in MongoDB standard/ free version, it will be able to store records upto a certain limit very large data records can’t be stored for a longer duration of time.
2. Realtime online payments are not possible yet as integrated razorpay test credentials have to change to a real time account to be able to have live payments.

3. Admin of B2C System will have to implement a security policy to safeguard the product to be delivered and related information from being accessed by unauthorized users (by means of gaining access to the backend database).

2.5 ASSUMPTIONS AND DEPENDENCIES

The deliverer will approach the seller to be able to sign up as a deliverer and the seller has to authenticate the person before giving them access to be a deliverer.

2.6 APPORTIONING OF REQUIREMENTS

OTP verification of deliverers everytime they sign up as no credentials are associated with a deliverer.

3.SPECIFIC REQUIREMENTS

This section contains the software requirements to a level of detail sufficient to enable designers to design the system, and testers to test that system.

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 USER INTERFACES

The following screens will be provided:

SignUp Screen :

It will allow buyers and sellers to register to the website. Signup Screen consists of details such as Full Name, email , phone number , shop name, shop address, shop description etc. There is no Signup screen for the delivery agent , as they are registered and assigned by the seller.

Login Screen:

It will allow already registered users to access different screens based upon the user's role. Various fields available on this screen will be:

Role: Buyer or Seller

Phone Number: which acts as a user ID for a given user.

Password: Alphanumeric string which gives user access to their account

Role: Delivery Agent

Name: Registered name of the deliverer.

Phone No.: Registered for no, of the deliverer.

Buyer Screen:

This screen will be accessible only to users with role Buyer. It will allow the user to browse through different shops of different sellers, and view their products. In each shop there is contact information available about the seller with which the buyer can contact the seller.

Product Screen:

They are accessed on viewing a product from a shop. It includes description of product, its price, options like choosing the amount of the particular product buyer wants to order and an add to cart option

Cart Screen:

This screen will be accessible only to users with role Buyer. All the products that the buyer adds to cart while browsing the items in shops are present here. An item present here can be removed too. This screen leads to two options for payment : Pay online , or Cash on Delivery.

Payment Screen :

The user is given 2 options either online payment or Cash on delivery. On selecting the Online Payment Option. It has a razorpay payment interface, which has several payment modes available, like UPI, debit card, credit card etc. In the UPI mode, the buyer can scan a QR generated via their mobile device and pay for the order.

Post-Payment Screen:

This screen comes on after successfully paying. It shows OrderID, paymentID. An order request is also sent to the seller after the buyer makes an order. There are other options on the screen that can be used to track the orders made by the buyer and check their status.

Seller Screen :

This screen will be accessible only to users with role Seller. It has options that allow sellers to post new products in their shop, to remove an existing product, update the stock of an existing product, check orders made by the buyers and assign a delivery agent to make these orders.

Add Product Screen :

This screen will be accessible only to users with role Seller. It is used by sellers to add a new product to their shop. To add a product, the seller has to enter details about it like, Product Name, Product Description, Price, Amount , Product Picture etc. After successfully adding a product , it can be viewed by the buyer in their interface.

Seller Order Screen :

This screen will be accessible only to users with role Seller. In this screen , sellers can see all the orders that are made for their products by the buyers, and accordingly they register and assign a delivery agent of their preference by adding the delivery agent's phone number and name. In this screen , the status of delivery can also be checked.

Buyer Order Screen :

This screen will be accessible only to users with role Buyer. In this screen , buyers can see all the orders that they have made. Here, they can check the status of delivery of their order, and contact the seller and delivery agent too.

Delivery Agent Screen :

This screen will be accessible only to users with role Delivery Agent. A Delivery Agent is registered by a Seller and an order is assigned to them for delivery by that Seller only. In their Interface they can see the assigned orders that are to be delivered. They can see the details like the details of Buyer, Seller and their contact information for delivery purposes. They can also update the status of the delivery which is reflected in both buyer and seller interface.

3.1.2. HARDWARE INTERFACES

As stated in section 2.1.3.

3.1.3 SOFTWARE INTERFACES

As stated in section 2.1.4.

3.1.4 COMMUNICATIONS INTERFACES

As stated in section 2.1.5.

3.2 SOFTWARE PRODUCT FEATURES

3.2.1 USER ACCOUNTS INFORMATION MAINTENANCE

Description : The system will maintain information about various users who will be able to access the system. The following information would be maintained that is common between all roles :

The following information is maintained for each role :

For Buyer : Name, Email, Phone No., Address, Password

For Seller : Name, Email, Phone No., Shop Name, Shop Address, Password

For Delivery Agent : Name, Phone No.

Validity Checks:

Only the database will be authorized to access the User Accounts Information Maintenance module. Any changes made by the user such as change of password will be reflected in the database

- Phone Number cannot be blank.
- Phone Number should be unique for every user.
- Email cannot be blank.
- Passwords cannot be blank.
- Address cannot be blank.
- Shop Name cannot be blank

Sequencing Information:

A User Account for a particular user has to be created in order for the system to be accessible to that user. In the website, the user has to be registered first using the sign up option , then they can log in anytime after that.

Error Handling/ Response to Abnormal Situations:

If any of the above validations/ sequencing flow does not hold true, appropriate error messages. will be prompted for users to do the needful.

3.2.2 PRODUCT INFORMATION MAINTENANCE

Description:

The system database will maintain information about the products that various sellers put up in their shop. The following information would be maintained for each product :

Product Name, Product Description, Product Picture, Product' Stock Available, Seller of the Product

The system database will allow creation/modification/deletion of new/existing products.

Validity Checks:

Only system databases will be authorized to access and maintain the Product information Maintenance module. Only Seller's can make changes to their respective products that will be reflected in the database.

- Product Name cannot be blank.
- Product Picture cannot be blank.
- Product Description cannot be blank.
- If Product's Stock becomes 0, it will show Out of Stock.
- Product Price cannot be zero.

Sequencing Information:

Product Info will have to be entered into the system database by the Seller using the interface provided before any they can be reflected on the webpage.

Error Handling/ Response to Abnormal Situations:

If any of the above validations/ sequencing flow does not hold true, appropriate error messages will be prompted for users to do the needful.

3.2.3 DELIVERY SYSTEM MAINTENANCE

Description :

The system database will maintain information like buyer’s address and phone number, seller’s address and phone number and the status of the product, whether it is delivered or yet to be delivered, it is updated by the delivery agent, stored in the database and shown to buyer and seller when they check the order status. The following information would be maintained for each delivery :

- Order Details.
- Order of the Status.
- The system will allow updates of status of the delivery by the delivery agent for a particular order.

Validity Checks:

- Only system databases will be authorized to access and maintain the Product information Maintenance module. Only users with the role Delivery Agent assigned the particular order can update the delivery status.
- Order Status can be changed to Delivered or shipped.

Sequencing Information:

Order will have to be made by the Buyer and Delivery agent will have to be assigned by the seller to that order before any delivery agent can access this module.

Error Handling/ Response to Abnormal Situations:

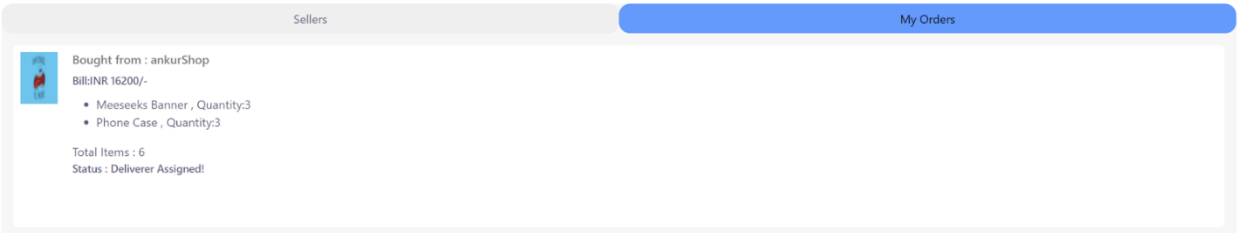
If any of the above validations/ sequencing flow does not hold true, appropriate error messages. and status will be prompted for users to do the needful.

3.2.4 ORDER GENERATION AND CONFIRMATION

Description :

The system will generate order receipt with a unique id along with the payment status and details to verify and confirm a particular order. The system database with be maintained and verified using razorpay APIs and Razorpay Dashboard.

ORDERS WILL HAVE THE FOLLOWING FORMAT :



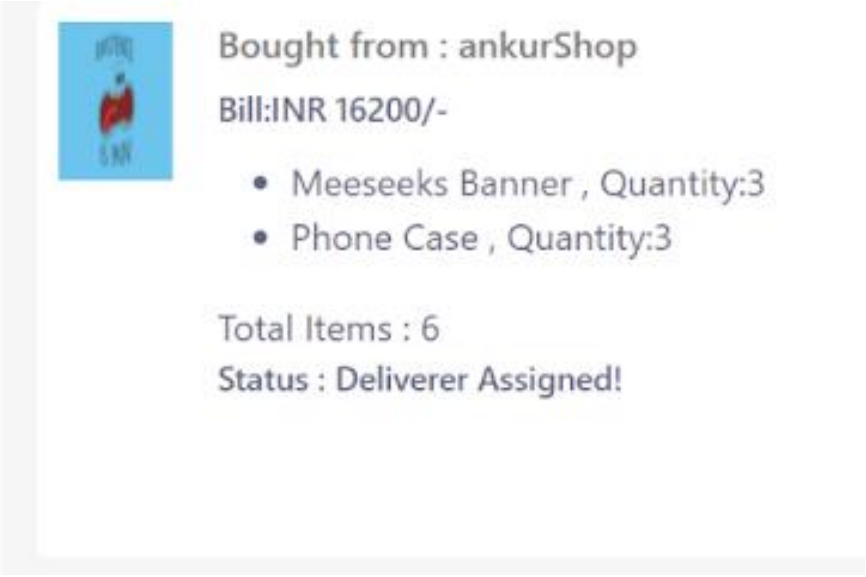
Bought from : ankurShop

Bill:INR 16200/-

- Meeseeks Banner , Quantity:3
- Phone Case , Quantity:3

Total Items : 6

Status : Deliverer Assigned!

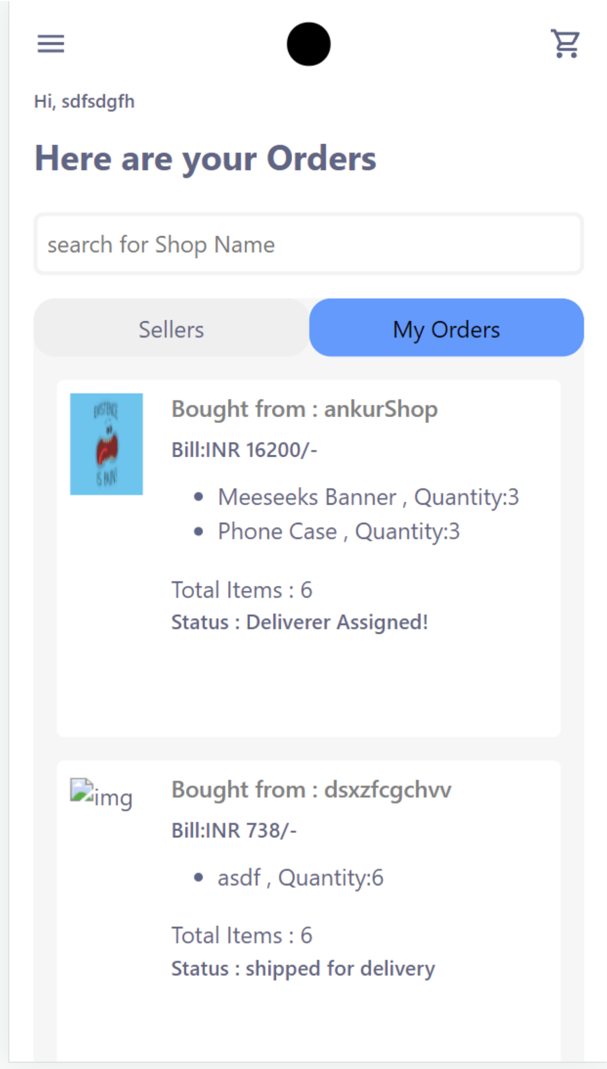

Bought from : ankurShop

Bill:INR 16200/-

- Meeseeks Banner , Quantity:3
- Phone Case , Quantity:3

Total Items : 6

Status : Deliverer Assigned!



Hi, sdfsdgfh

Here are your Orders

Bought from : ankurShop

Bill:INR 16200/-

- Meeseeks Banner , Quantity:3
- Phone Case , Quantity:3

Total Items : 6

Status : Deliverer Assigned!

Bought from : dsxzfcgchvv

Bill:INR 738/-

- asdf , Quantity:6

Total Items : 6

Status : shipped for delivery

Validity Checks:

Users who have ordered the product, seller of the product and assigned deliverer will be able to access this module.

Sequencing Information:

Order for a particular Buyer can be generated by the system only after the payment is done by the buyer and verified by the system using Razorpay API. OrderID and a receipt will be generated after that.

Error Handling/ Response to Abnormal Situations:

If any of the above validations/ sequencing flow does not hold true, appropriate error messages will be prompted for users to do the needful.

3.3 PERFORMANCE REQUIREMENTS

None

3.4 DESIGN CONSTRAINTS

For designing Figma was used and the system is device responsive but still has to work on improving the user Interface.

3.5 SOFTWARE SYSTEM ATTRIBUTES

3.5.1 SECURITY

The application will be password protected. Users will have to enter a valid phone number, password and role in order to access the application.

3.5.2 MAINTAINABILITY

The application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual modules when the need arises.

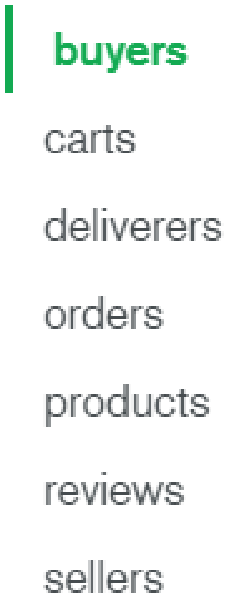
3.5.3 PORTABILITY

The application will be easily portable on any windows-based system that has a Web Browser.

3.6 LOGICAL DATABASE REQUIREMENTS

The following information will be placed in the database :

- User Information according to their Role (Buyer, Seller, Deliverer)
- Order Details
- Product details
- Cart Details
- Reviews



buyers

carts

deliverers

orders

products

reviews

sellers

3.7 OTHER REQUIREMENTS

Clients should have a working phone no. along with an email.

Conclusion

The SRS was written successfully by following the template described above.

Viva Questions

1. What are the objectives of requirement analysis?

Ans.

The purpose of the Requirements Analysis Phase is to transform the needs and high-level requirements specified in earlier phases into unambiguous (measurable and testable), traceable, complete, consistent, and stakeholder-approved requirements.

2. Define different types of requirements?

Ans.

According to IEEE standard 729, a requirement is defined as follows:

- A condition or capability needed by a user to solve a problem or achieve an objective
- A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents
- A documented representation of a condition or capability as in 1 and 2.

A software requirement can be of 3 types:

- Functional requirements
- Non-functional requirements
- Domain requirements

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

For example, in a hospital management system, a doctor should be able to retrieve the information of his patients. Each high-level functional requirement may involve several interactions or dialogues between the system and the outside world. In order to accurately describe the functional requirements, all scenarios must be enumerated.

There are many ways of expressing functional requirements e.g., natural language, a structured or formatted language with no rigorous syntax and formal specification language with proper syntax.

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

NFR's are classified into following types:

- Interface constraints
- Performance constraints: response time, security, storage space, etc.
- Operating constraints
- Life cycle constraints: mantainability, portability, etc.
- Economic constraints

The process of specifying non-functional requirements requires the knowledge of the functionality of the system, as well as the knowledge of the context within which the system will operate.

Domain requirements: Domain requirements are the requirements which are characteristic of a particular category or domain of projects. The basic functions that a system of a specific domain must necessarily exhibit come under this category. For instance, in an academic software that maintains records of a school or college, the functionality of being able to access the list of faculty and list of students of each grade is a domain requirement. These requirements are therefore identified from that domain model and are not user specific.

3. Outline structure of SRS Document?

Ans.

1. Introduction

- **(i) Purpose of this document**
- **(ii) Scope of this document**
- **(iii) Overview**

2. General description
3. Functional Requirements
4. Interface Requirements
5. Performance Requirements
6. Design Constraints
7. Non-Functional Attributes
8. Preliminary Schedule and Budget
9. Appendices

Outline used for this SRS:

1. Introduction
    - 1.1. Purpose
    - 1.2. Scope
    - 1.3. Definitions, Acronyms, and Abbreviations
    - 1.4. References
    - 1.5. Overview
2. Overall Description
    - 2.1. Product Perspective
        - 2.1.1. System Interfaces

3.2.4.4. Error Handling/ Response to Abnormal Situations

3.3. Performance Requirements

3.4. Design Constraints

3.5. Software System Attributes

3.5.1. Security

3.5.2. Maintainability

3.5.3. Portability

3.6. Logical Database Requirements

3.7. Other Requirements

4. What are benefits of writing SRS document?

Ans.

- An SRS establishes the basis for agreement between the customer and the supplier on what the software product will perform.
- An SRS provides a reference for validation of the final product/software.
- A high-quality SRS is a prerequisite to high-quality product/software.
- A high-quality SRS reduces the development cost.

5. Define Functional and non-functional requirements?

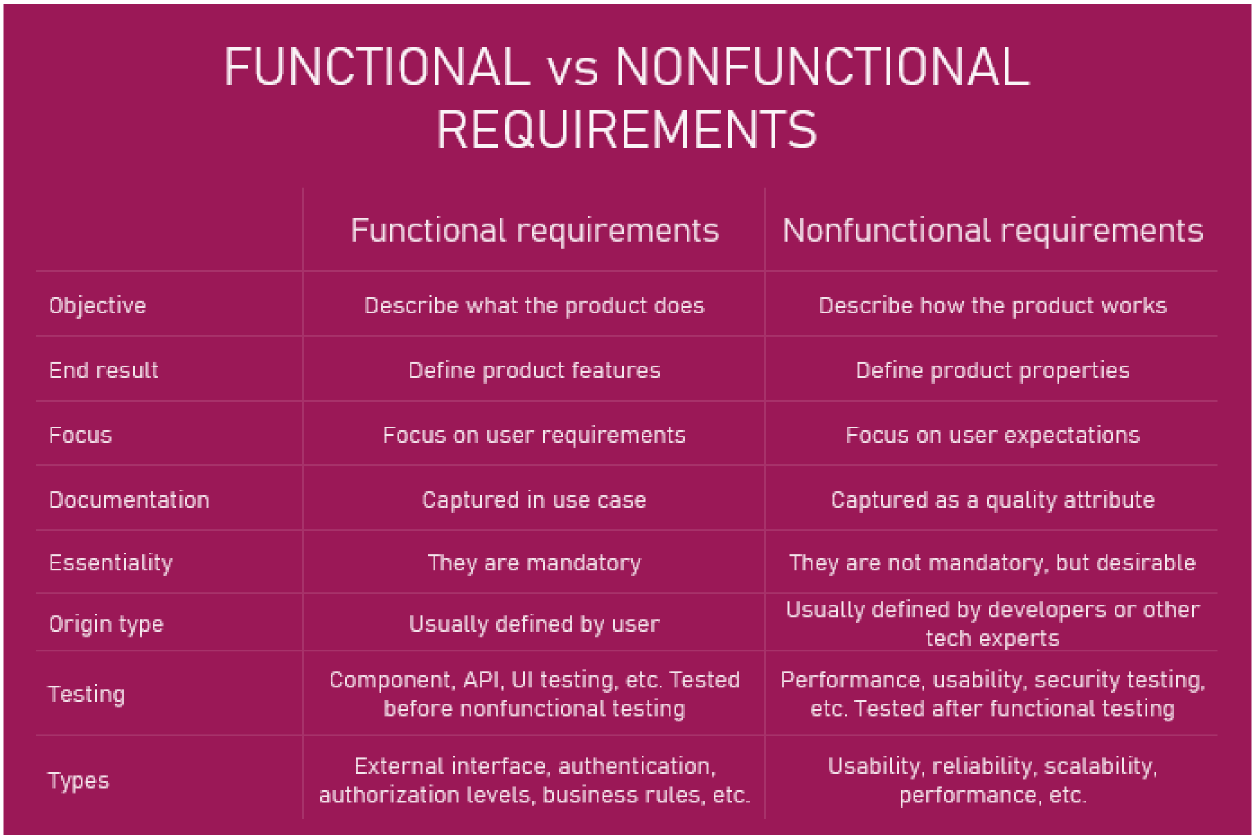
Ans.

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

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



FUNCTIONAL vs NONFUNCTIONAL REQUIREMENTS

	Functional requirements	Nonfunctional requirements
Objective	Describe what the product does	Describe how the product works
End result	Define product features	Define product properties
Focus	Focus on user requirements	Focus on user expectations
Documentation	Captured in use case	Captured as a quality attribute
Essentiality	They are mandatory	They are not mandatory, but desirable
Origin type	Usually defined by user	Usually defined by developers or other tech experts
Testing	Component, API, UI testing, etc. Tested before nonfunctional testing	Performance, usability, security testing, etc. Tested after functional testing
Types	External interface, authentication, authorization levels, business rules, etc.	Usability, reliability, scalability, performance, etc.