

Ali Haider
14259
Final Term Assignment
Course: SRS
Instructor: Mam Aasma Khan
Date: June 24, 2020



- **Attempt all questions.**
- **Marks will be given as per the DEPTH of the answer, not LENGTH.**

Question No: 01

(10)

Define requirements and define what the system (take example of any system) is required to do and what are the features and constraints under which it operates.

ANSWER#1:

Requirements:

The **requirements** themselves are the descriptions of the system services and constraints that are generated during the requirements engineering process. Requirements may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification. As much as possible, requirements should describe **what** the system should do, but **not how** it should do it.

Example: (Online Shopping System(OSS))

– **Purpose:**

The Online Shopping System (OSS) for electronics item shop web application is intended to provide complete solutions for vendors as well as customers through a single get way using the internet. It will enable vendors to setup online shops, customer to browse through the shop and purchase them online without having to visit the shop physically. The administration module will enable a system administrator to approve and reject requests for new shops and maintain various lists of shop category.

– **Scope:**

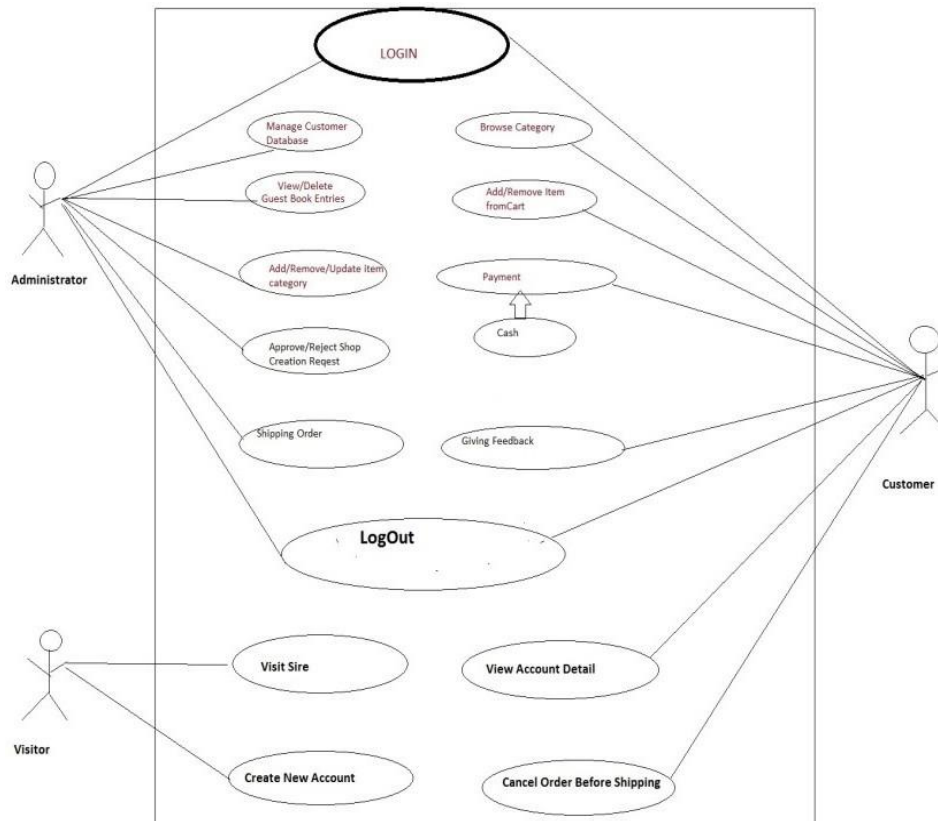
This system allows the customer's to maintain their cart for add or remove the product over the internet

– **Product Perspective:**

This product aimed toward a person who don't want to visit the shop as he might don't get time for that or might not interested in visiting there and dealing with lot of formalities.

– **Product Functions:**

OSS should support this use case:



Functional Requirements:

This section provides requirement overview of the system.

Registration

If customer wants to buy the product then he/she must be registered, unregistered user can't go to the shopping cart.

Login

Customer logs in to the system by entering valid user id and password for the shopping.

Changes to Cart

Changes to cart means the customer after login or registration can make order or cancel order of the product from the shopping cart.

Payment

In this system we are dealing the mode of payment by Cash. We will extend this to credit card, debit card etc in the future.

Logout

After ordering or surfing for the product customer has to logout.

Report Generation

After ordering for the product, the system will send one copy of the bill to the customer's Email-address and another one for the system data base.

Non-Functional Requirements:

Following Non-Functional Requirements will be there in the insurance to the internet:

- Secure access to consumer's confidential data.
- 24X7 availability.
- Better component design to get better performance at peak time.
- Flexible service based architecture will be highly desirable for future extension. Non-Functional Requirements define system properties and constraints.

Various other Non-Functional Requirements are:

- Security
- Reliability
- Maintainability
- Portability
- Extensibility
- Reusability
- Compatibility
- Resource Utilization

Performance Requirements:

In order to maintain an acceptable speed at maximum number of uploads allowed from a particular customer as any number of users can access to the system at any time. Also the connections to the servers will be based on the attributes of the user like his location and server will be working 24X7 times.

Question No: 02

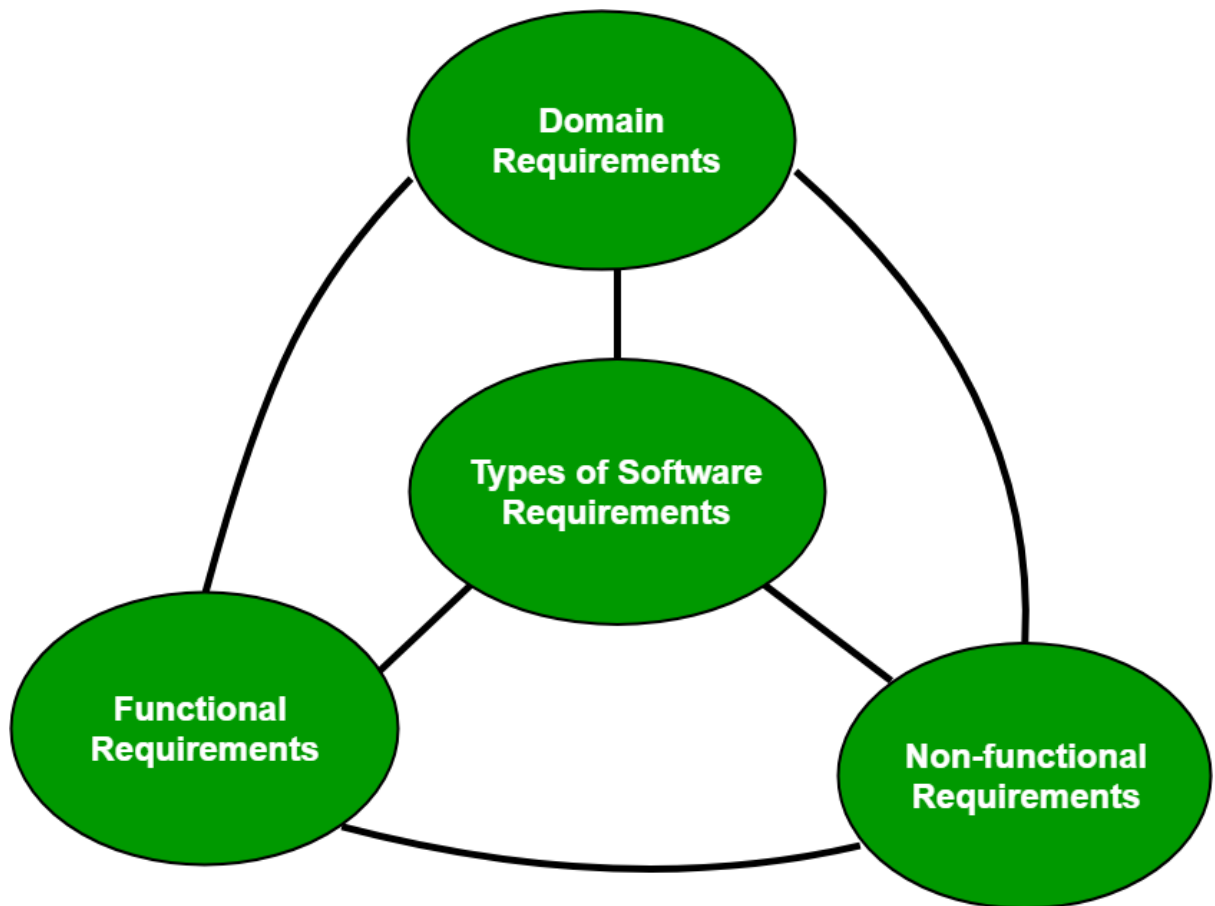
(10)

Explain software requirements types.

ANSWER#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

- Functional user requirements may be high-level statements of what the system should do; functional system requirements should describe the system services in details.
- Describe functionality or system services.
- Depend on the type of software, expected users and the type of system where the software is used.

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.

Non-functional requirements:

☞ Product requirements:

- Requirements which specify that the delivered product must behave in a particular way, *e.g.* execution speed, reliability *etc.*

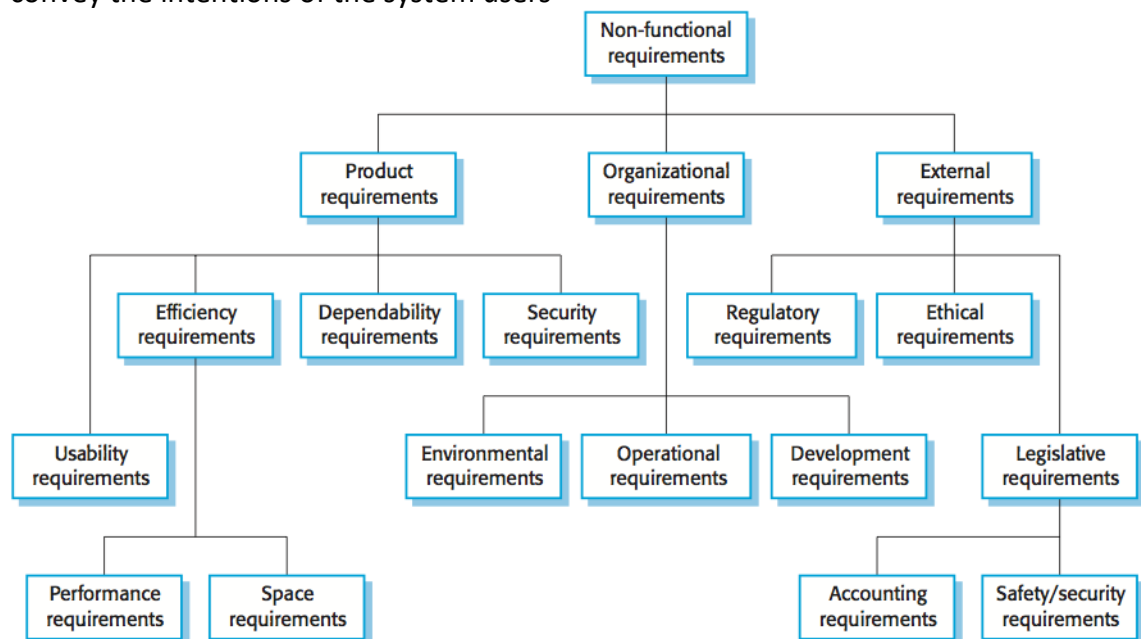
☞ Organisational requirements:

- Requirements which are a consequence of organisational policies and procedures, *e.g.* process standards used, implementation requirements *etc.*

☞ External requirements:

- Requirements which arise from factors which are external to the system and its development process, *e.g.* interoperability requirements, legislative requirements *etc.*

Non-functional requirements may be very difficult to state precisely and imprecise requirements may be difficult to verify. If they are stated as a **goal** (a general intention of the user such as ease of use), they should be rewritten as a **verifiable** non-functional requirement (a statement using some **quantifiable metric** that can be objectively tested). Goals are helpful to developers as they convey the intentions of the system users



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.

- Describe system characteristics and features that reflect the domain
- May be new functional requirements, constraints on existing requirements or may define specific computations
- If domain requirements are not satisfied, the system may be unworkable.

Question No: 03

(10)

State difference between system requirement engineering and software requirement engineering.

System Requirement Engineering:

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems. ... System requirements are also known as minimum system requirements

System requirements are all of the requirements at the system level that describe the functions which the system as a whole should fulfill to satisfy the stakeholder needs and requirements, and are expressed in an appropriate combination of textual statements, views, and non-functional requirements; the latter expressing the levels of safety, security, reliability, etc., that will be necessary.

System requirements play major roles in systems engineering, as they:

- Form the basis of system architecture and design activities.
- Form the basis of system integration and verification activities.
- Act as reference for validation and stakeholder acceptance.
- Provide a means of communication between the various technical staff that interact throughout the project.

Software Requirement Engineering:

Software specification or requirements engineering is the process of understanding and defining what services are required and identifying the constraints on these services. Requirements engineering processes ensures your software will meet the user expectations, and ending up with a high-quality software.

Software Requirement Engineering is perhaps the most difficult, most error-prone and most communication intensive software development. It can be successful only through an effective customer-developer partnership. It is needed to know what the users really need.

There are a number of requirements elicitation methods. Few of them are listed below –

- Interviews
- Brainstorming Sessions
- Facilitated Application Specification Technique (FAST)
- Quality Function Deployment (QFD)
- Use Case Approach

Question No 04:

(10)

Give five reasons why requirements negotiation is needed in software engineering.

ANSWER#4:

Negotiation:

- is a decision-making process.
- involves interaction and interdependency.
- implies two parties with conflicts

Reasons For Negotiation:

- Negotiation helps to encourage communication and therefore improve understanding of the requirements.
- Negotiation helps to reveal conflict, solution exploration, and collaborative resolution.
- Negotiation helps to improve the agreement level among the stakeholders.
- The objectives of customers, users, or developers have to be reconciled to develop mutually acceptable agreements
- Find if a better solution is possible for one party, without causing loss to others
- By Negotiation it ensure stakeholders' commitment over time
- In Negotiation Risks associated with each requirement are identified and analyzed
- Rough guesses of development effort are made and used to assess the impact of each requirement on project cost and delivery time
- Using an iterative approach, requirements are eliminated, combined and/or modified so that each party achieves some measure of satisfaction

Question No 05:

(10)

Identify the **actors** and the **objects** in the following scenario to register a patient in a hospital management system and draw a **use case diagram**:

The administrator enters the patient's name, address, date of birth and emergency contact details into the system. If the patient has only public health insurance, the

administrator enters the patient's medicare number, and the system verifies this with government health database. If the patient also has private health insurance, then the administrator enters also the patient's private health insurance details, and the system verifies these details with the private health insurance system. When these details are verified as correct, the system saves the patient's details and confirms the registration.

ANSWER#5:

Actors:

Administrator, Government Health Database (external systems),
Private Health InsuranceSystem (external systems)

Object:

Patients, Administrator, Address, Emergency Contact Number, Public Health Insurance, Private Health Insurance, Registration.

Use Case Diagram:

