

Name : Aman ullah
Id# : 14303
Program: BS(SE) 05

Question No: 01

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:

Requirements:

1. A condition or capacity needed by a user to solve a problem or achieve an objective.
2. 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.
3. A documented representation of a condition or capacity as in (1) or (2).

Define what the system is required to do and the constraints under which it is required to operate

- The system shall maintain records of all library materials including books, serials, newspapers and magazines, video and audio tapes, reports, collections of transparencies, computer disks and CD-ROMs.

- The system shall allow users to search for an item by title, author, or by ISBN.
- The system's user interface shall be implemented using a World-Wide-Web browser.
- The system shall support at least 20 transactions per second.
- The system facilities which are available to public users shall be demonstrable in 10 minutes or less.

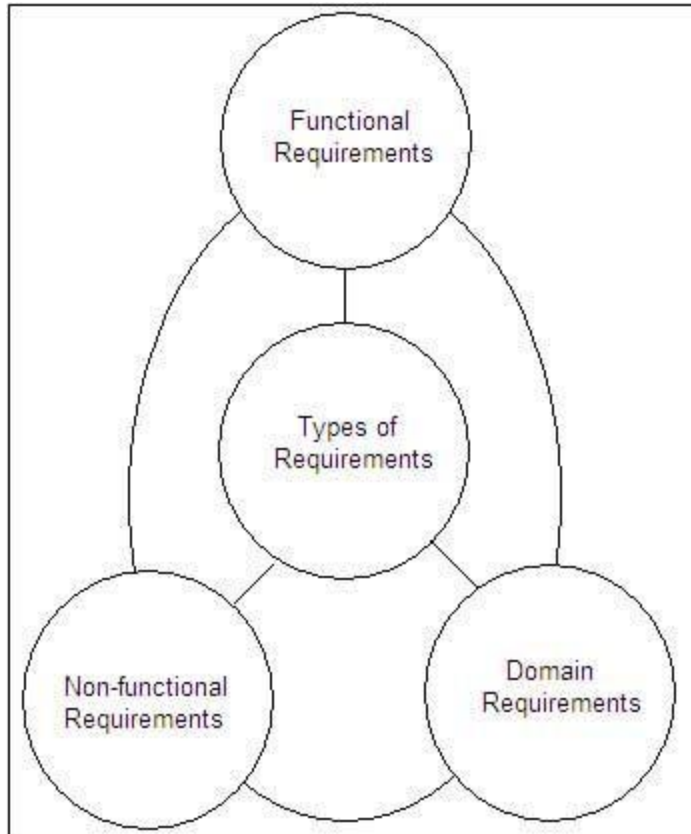
Question No: 02

Explain software requirements types.

Answer:

Types of Software Requirements

Requirements help to understand the behavior of a system, which is described by various tasks of the system. For example, some of the tasks of a system are to provide a response to input values, determine the state of data objects, and so on. Note that requirements are considered prior to the development of the software. The requirements, which are commonly considered, are classified into three categories, namely, functional requirements, non-functional requirements, and domain requirements.



Functional requirements:

IEEE defines functional requirements as 'a function that a system or component must be able to perform.' These requirements describe the interaction of software with its environment and specify the inputs, outputs, external interfaces, and the functions that should be included in the software. Also, the services provided by functional requirements specify the procedure by which the software should react to particular inputs or behave in particular situations.

Non-functional requirement:

Non-functional requirements should be accomplished in software to make it perform efficiently. For example, if an aeroplane is unable to fulfill reliability requirements, it is not approved for safe operation. Similarly, if a real time control

system is ineffective in accomplishing non-functional requirements, the control functions cannot operate correctly.

The description of different types of non-functional requirements is listed below.

1. **Product requirements:** These requirements specify how software product performs. Product requirements comprise the following.
2. **Efficiency requirements:** Describe the extent to which the software makes optimal use of resources, the speed with which the system executes, and the memory it consumes for its operation. For example, the system should be able to operate at least three times faster than the existing system.
3. **Reliability requirements:** Describe the acceptable failure rate of the software. For example, the software should be able to operate even if a hazard occurs.
4. **Portability requirements:** Describe the ease with which the software can be transferred from one platform to another. For example, it should be easy to port the software to a different operating system without the need to redesign the entire software.
5. **Usability requirements:** Describe the ease with which users are able to operate the software. For example, the software should be able to provide access to functionality with fewer keystrokes and mouse clicks.
6. **Organizational requirements:** These requirements are derived from the policies and procedures of an organization. Organizational requirements comprise the following.
7. **Delivery requirements:** Specify when the software and its documentation are to be delivered to the user.
8. **Implementation requirements:** Describe requirements such as programming language and design method.
9. **Standards requirements:** Describe the process standards to be used during software development. For example, the software should be developed using standards specified by the ISO and IEEE standards.
10. **External requirements:** These requirements include all the requirements that affect the software or its development process externally. External requirements comprise the following.
11. **Interoperability requirements:** Define the way in which different computer based systems will interact with each other in one or more organizations.
12. **Ethical requirements:** Specify the rules and regulations of the software so that they are acceptable to users.

domain requirements:

Requirements which are derived from the application domain of the system instead from the needs of the users are known as **domain requirements**. These requirements may be new functional requirements or specify a method to perform some particular computations. In addition, these requirements include any constraint that may be present in the existing functional requirements. As domain requirements reflect the fundamentals of the application domain, it is important to understand these requirements. Also, if these requirements are not fulfilled, it may be difficult to make the system work as desired.

Question No: 03

State difference between system requirement engineering and software requirement engineering.

Answer:

System requirements engineering:

The requirements for the system as a whole are established and written to be understandable to all stakeholders.

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 requirements engineering:

More detailed system requirements are derived for the system Software.

Software requirements engineering refers to the first phase, before any of the actual designing, coding, testing, or maintenance takes place. The goal is to create an important early document and process in the software design. Often referred to as software requirements specification, or SRS, it determines what software is produced. It is basically the gathering of information of a customer's or potential customer/target audience's requirements for a system, before any actual design takes place.

Question No 04:

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

Answer:

Reasons:

- 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.
- Negotiation helps to develop stakeholders' satisfaction.
- Negotiation helps to improve requirements quality.

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:

Actors:

Administrator

Government Health Database

Private Health Insurance System

(The last two are external systems)

Objects:

- Patient
- Administrator
- Address
- EmergencyContact
- PublicHealthInsurance
- PrivateHealthInsurance
- Registration

Assumptions:

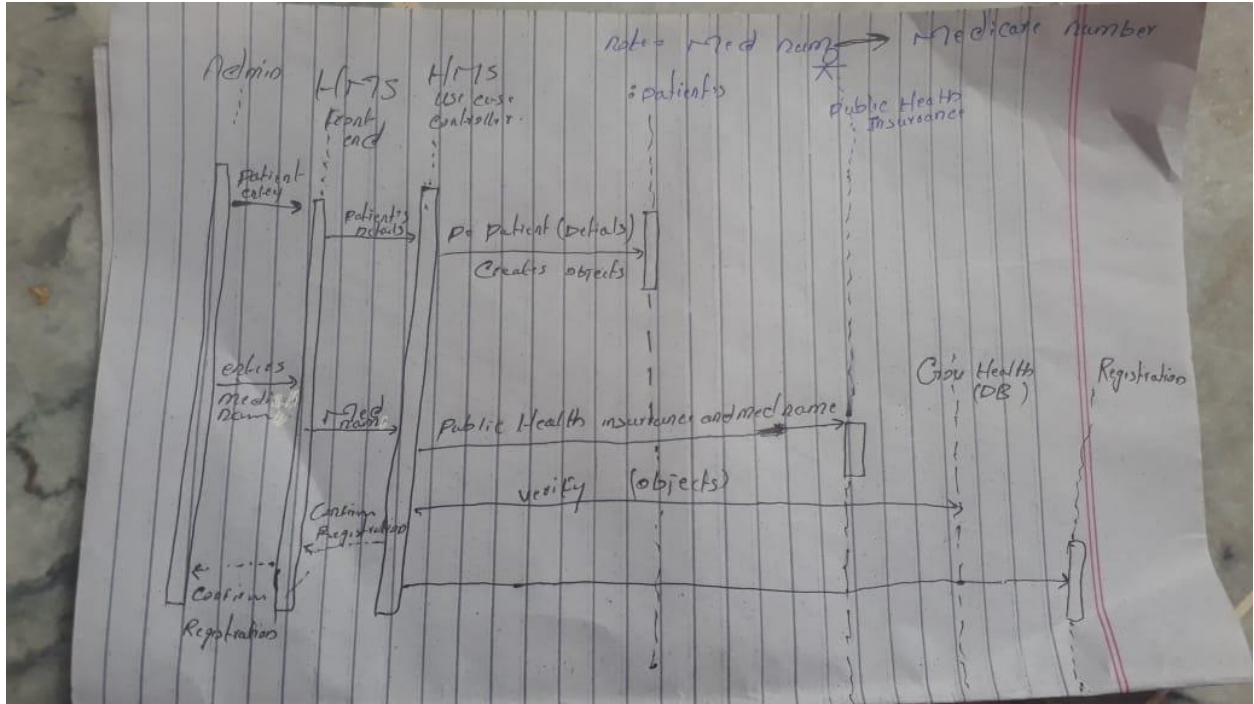
* Address contains street number, street name, suburb, and postcode and so is large enough to be an object.

* Other nouns identified that are not considered to be objects include: patient's name and date of birth. They are considered to be stored as primitive types or strings and are attributes of other objects.

* EmergencyContact is assumed to contain a contact person's name and phone number (and most likely the address and relationship to the patient) and so should be an object rather than an attribute.

* the noun "details" appears a few times. It is not considered to be an object as the word refers to the attributes of some objects.

USE CASE DAIGRAM:



THE ENDS