gradeup

# Software Requirements Part-1

# Software Requirements:

**Content:**

1. **Functional and Non - Functional Requirement**
2. **Use Case**
3. **Requirement Analysis and Review**
4. **SRS**

## Functional and Non-Functional Requirements:

It define the basic system behavior. They are system that does or must not do, and can be thought of in terms of how the system responds to inputs. It define that behavior and include calculations, data input, and business processes.

They are features that allow the system to function as it was intended. Another way, the functional requirements are not met, the system will not work. They are product features and focus on user interface.

**Nonfunctional Requirements:** It define system attributes such as security, reliability, performance, maintainability,and usability. It provide constraints or limitations on the design of the system across the different backlogs. They ensure the effectiveness of the entire system.

## Difference between Functional and Non Functional Requirements:




Gradeup UGC NET Super Subscription
Access to all Structured Courses & Test Series

| FUNCTIONAL REQUIREMENT | NON FUNTIONAL REQUIREMENT |
|---|---|
| It defines a system or its component. | It defines the quality attribute of a system. |
| It specifies “What should the software system do?” | It places constraints on “How should the software system fulfill the functional requirements?” |
| It is specified by User. | It is specified by technical peoples e.g. Architect, Technical leaders and software developers. |
| It is mandatory. | It is not mandatory. |
| It helps user to verify the functionality of the software. | It helps user to verify the performance of the software. |
| It is easy to define. | It is more difficult to define. |
| **Example :** A Verification email is sent to user whenever he/she registers for the first time on some software system. | **Example**: Site should load in 3 seconds when the number of continuous users are > 10000 |

**Eliciting Requirements:**Itis practice of researching and discovering the requirements of a system from users, customers, and other stakeholders.The practice is referred to as "**requirement gathering**".

They cannot be collected from the customer, as they would be showed by the name requirements gathering. It is non-trivial because user can never be sure you get all requirements from the user and customer by just asking them what the system should do or not dopractices include interviews, questionnaires, user observation.

**Developing Use Cases:**It is a list of actions or event defining the interactions between a role and a system to achieve a goal.

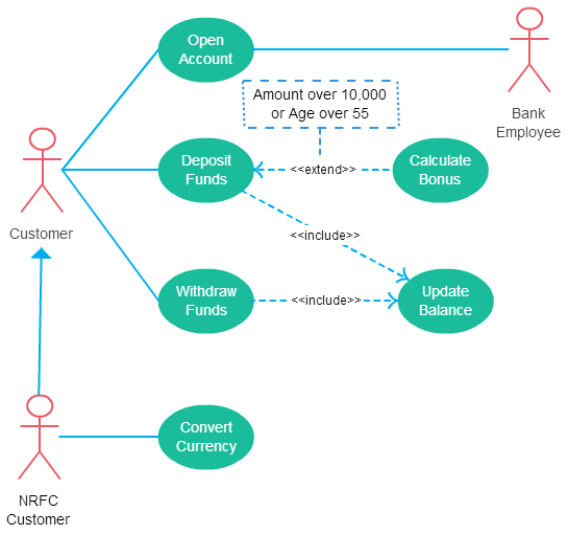
1. **Actor**, which is the user, which can be a single person or a group of people, interacting with a process
2. **System**, is the process that's required to reach the final outcome




Gradeup UGC NET Super Subscription
Access to all Structured Courses & Test Series

3. **Goal**, which is the successful user outcome
4. **Stakeholders**, which are those who have an interest in how the system turns out, even if they aren't direct users
5. **Preconditions**, which are things that must be true before a use case is run
6. **Triggers**, which are events that occur for a use case to begin

**Example**




**Requirement Analysis And Modelling:**It is the most important skill for a business analyst. The success of any software project depends on this task. Itinvolves multiple tasks.

**Requirements Analysis and Modelling Techniques:**

These (RAM) techniques are practiced to conduct this activity. Some of these techniques are :

- Structured RAM with DFD and ER diagrams
- RAM with UML diagrams
- Custom Technique




Gradeup UGC NET Super Subscription
Access to all Structured Courses & Test Series

**Requirements Review:**It is a manual process that involves people from both client and contractor organisations. They check the requirements document and omissions.

It can be used throughout software development for quality assurance and data collection. It is a set of people to find errors and point out other matters of concern in the requirement specification of system. It involves the author of requirement documents, someone who understands the client, a person of the design team, and the person responsible for maintaining the requirement document. It is good practice to include few people not directly involved with product development.

**Software Requirement And Specification (SRS) Document:** Itcan prevent software project failure.

It develops the basis for document between customers and contractors or suppliers on how the software product should function. Software requirements specification(SRS) is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should provide a realistic basis for estimating product costs, risks, and schedules.

SRS provides an overview of the entire SRS with purpose, scope, definitions, acronyms, and overview of the SRS. The goal of document is to analyze and give in-depth insight of the complete software system by defining the problem statement in detail.

**Qualities of SRS:**

- Correct
- Unambiguous
- Complete




Gradeup UGC NET Super Subscription
Access to all Structured Courses & Test Series

- Consistent
- Ranked for importance and/or stability
- Verifiable
- Modifiable
- Traceable


**Correctness:**

It is used to ensure the correctness of requirements. It is correct if it covers all the requirements that are actually expected from the system.

**Completeness:**

It indicates each sense of completion including the number of all the pages, resolving the to be determined parts to as much extent as possible as well as covering all the functional and non-functional requirements properly.

**Consistency:**

They are said to be consistent if there are no any other conflicts between any group of requirements. Examples of conflict include differences in terms used at separate places, logical conflicts like time period of report generation, etc.

**Unambiguousness:**

It is said to be unambiguous if all the requirements stated only 1 interpretation. Ways to prevent, it include the use of modern techniques like ER diagrams, proper reviews etc.



Gradeup UGC NET Super Subscription
Access to all Structured Courses & Test Series

**Ranking for importance and stability:**

There should a criterion to classify the requirements as less or more important or more specifically as desirable or required. An identifier mark can be used with each requirement to show its rank or stability.

**Modifiability:**

It should be made as updatation as possible and should be capable of easily accepting changes to the system to few extent. Modifications should be properly indexed and cross-referenced.

**Verifiability:**

It is verifiable when there is exists a particular technique to quantifiable measure the extent to which every requirement is met by the system. For example, a requirement state that the system must be user-friendly is not verifiable and listing such requirements should be avoided.

**Traceability:**

One should be able to trace a requirement to design component and then to code segment in the program. One should be able to find a requirement to the corresponding test cases.

**Design Independence:**

There should be option to select from couple of design alternatives for the end system. The SRS should not include any impose details.

**Testability:**

It should be written in such a way that it is easy to generate test cases and test plans from the document.




Gradeup UGC NET Super Subscription
Access to all Structured Courses & Test Series

gradeup SUPER


# Gradeup UGC NET
## Super Superscription

**Features:**

1. 7+ Structured Courses for UGC NET Exam
2. 200+ Mock Tests for UGC NET & MHSET Exams
3. Separate Batches in Hindi & English
4. Mock Tests are available in Hindi & English
5. Available on Mobile & Desktop

---

Gradeup Super Subscription, Enroll Now

gradeup.co