



**S. B. JAIN INSTITUTE OF TECHNOLOGY,
MANAGEMENT & RESEARCH, NAGPUR.**

Session 2023-24

LAB MANUAL

Software Engineering & Project Management

Lab (PCCCS504P)

Year: 3rd

Semester: 5th

Name of Student:

Roll No.:

Semester/Year:

Academic Session:

Date of Performance:

Date of Submission:

Hardware and Software Requirement

Hardware Requirement

- Processor : Dual Core
- RAM : 1GB
- Hard Disk Drive : > 80 GB

Software Requirement

- Operating System – Windows 2007 and Ubuntu ·

Package used – gcc/g++

- IDE - StarUml

Institute Vision:

Emerge as a leading Institute for developing competent and creative Professionals

Institute Mission:

1. Providing Quality Infrastructure and experienced faculty for academic excellence
 2. Inculcating skills, knowledge and opportunities for competency and creativity 3.
- Aligning with Industries for knowledge sharing, research and development

Department Vision:

To become a centre for quality education in the field of computer science & engineering and to create competent professionals.

Department Mission:

- To provide academic ambience and latest software tools to prepare competent Software Engineers with strong theoretical and practical knowledge.
- To foster professionalism and strong work ethics in students for the betterment of Society.
- To provide adequate infrastructure as well as experienced & skilled faculty members. ● To encourage the spirit of entrepreneurship and adaptability in our students in view of the ever-changing scenario of the Software Industry.

Course Outcomes:

On successful completion of this course, students will be able to

1. **Understand:** Interpret and determine the role of the operating system to deal with computer hardware by using its fundamental concepts.
2. **Apply:** Utilize the concept of process management, synchronization and memory management in designing operating systems to resolve different issues.
3. **Apply:** Apply various disk scheduling algorithms, system security and files by identifying the problem to find the solution.
4. **Analyze:** Analyze different process scheduling, process synchronization problems and different memory management techniques under different situations to identify the optimal solution. (improve system performance.)
5. **Analyze:** Examine the scenario using knowledge of deadlock, system security, files and disk scheduling algorithms to solve the real-world problems.

GENERAL INSTRUCTIONS FOR STUDENTS

DO'S

- Students should enter into the Laboratory with prior permission.
- Students should come in proper uniforms.
- Students should come with Practical note book to the laboratory.
- Students should maintain silence inside the laboratory.
- After completing the laboratory exercise, make sure to shut down the system and arrange chairs properly.

DONT'S

- Students bringing the bags inside the laboratory.
- Students using mobile phones inside the laboratory.
- Students using the computers in an improper way.
- Students scribbling on the desk and mishandling the chairs.
- Students making noise inside the laboratory.

List of Problem Statement

Sr.No.	Problem Statement	Sr.No	Problem Statement
1	E-commerce Website	23	Recipe Database
2	Customer Relationship Management (CRM) System	24	Online Learning Platform
3	Online Banking System	25	Legal Case Management System
4	Movie Database System	26	Art Gallery Management System
5	Gym Management System	27	Pet Adoption Platform
6	Restaurant Management System	28	Fitness Tracking Application
7	Property Management System	29	Home Automation System
8	Travel Booking System	30	Event Ticketing System
9	Inventory Tracking System	31	Mobile Health Monitoring App
10	Online Auction Platform	32	Legal Document Management System
11	Music Streaming Service	33	Fleet Management System
12	Healthcare Information System	34	Home Rental Platform
13	Social Networking Platform	35	Recipe Sharing Platform
14	Event Management System	36	Online Gaming Platform
15	Supply Chain Management System	37	Hotel Review Platform
16	Parking Management System	38	Carpooling Application

Software Engineering & Quality Assurance Lab (PCCCS504P)

17	Tourism Information System	39	Music Band Management System
18	Job Portal	40	Document Collaboration Platform
19	School Management System	42	Volunteer Management System
20	Real Estate Listings	42	Language Learning Application
21	Warehouse Management System	43	Community Forum
22	Project Management Tool	44	Personal Finance Tracker

45	Weather Forecasting System	48	Medical Research Database
46	Ticket Reservation System for Museums	49	Online Marketplace for Handmade Crafts
47	Fashion E-commerce Platform	50	Community Library Database

List of Practicals

List of Experiments		
Sr. No.	Aim of Experiment	CO Mapped
Pre-Laboratory Session: Demonstration & Installation of various tools required for implementation of Software Engineering & Quality Assurance practical.		
1	Identify the problem definition and requirement for the given problem statement.	CO2
2	Design the Use Case diagram and capture the Use Case Scenarios for the given problem definition.	CO4
3	Create the E-R Model from the given problem definition.	CO4
4	Analyze the Domain classes and design Class diagram for the given problem definition.	CO3, CO4
5	Design the UML Sequence diagram for the given problem definition.	CO4
6	Construct the State Chart diagram for the given problem definition.	CO4
7	Construct the Activity diagram for the given problem definition.	CO4
8	Design the Data Flow Diagram (DFD) for the given problem definition.	CO4
9	Estimation of Project Metrics.	CO3
10	Estimation of Test Coverage Metrics and Structural Complexity.	CO3
11	Identify and design the Test Suites for the given problem definition.	CO2, CO4
Post Laboratory Session: Construct SRS (Software Requirement Specification) for the given problem statement.		



**S. B. JAIN INSTITUTE OF TECHNOLOGY,
MANAGEMENT & RESEARCH, NAGPUR.**

Practical No. 0(PRE-LAB)

AIM: Identification and Installation of various tools required for implementation of Software Engineering and Quality Assurance Practical.

Name of Student	Shrutika Pradeep Bagdi
Roll No	CS22130
Semester/Year	5 th /3 rd
Academic Session	2024-2025
Date of Performance	
Date of Submission	

AIM: - Identification and Installation of various tools required for implementation of Software Engineering and Quality Assurance.

OBJECTIVE/EXPECTED LEARNING OUTCOME:

- To impart state-of-the-art knowledge on Software Engineering and UML in an interactive manner through the Web
 - Present case studies to demonstrate the practical applications of different concepts •
- Provide a scope to the students where they can solve small, real-life problems

THEORY:

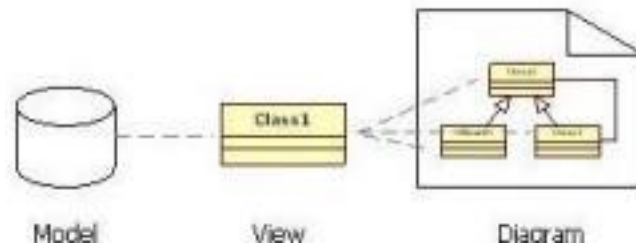
StarUML is a software engineering tool for system modelling using the Unified Modeling Language, as well as Systems Modeling Language, and classical modeling notations. It is published by MKLabs and is available on Windows, Linux and MacOS.

StarUML is an open-source software modeling tool that supports the UML (Unified Modeling Language) framework for system and software modeling. It is based on UML version 1.4, provides eleven different types of diagrams and it accepts UML 2.0 notation. It actively supports the MDA (Model Driven Architecture) approach by supporting the UML profile concept and allowing to generate code for multiple languages.

Features:

When you start a new project, StarUML proposes which approach you want to use: 4+1 (Krutchen), Rational, UML components (from Cheesman and Daniels book), default or empty. Depending on the approach, profiles and/or frameworks may be included and loaded. If you don't follow a specific approach, the "empty" choice could be used. Although a project can be managed as one file, it may be convenient to divide it into many units and manage them separately if many developers are working on it together.

StarUML makes a clear conceptual distinction between models, views and diagrams. A Model is an element that contains information for a software model. A View is a visual expression of the information contained in a model, and a Diagram is a collection of view elements that represent the user's specific design thoughts.



StarUML is built as a modular and open tool. It provides frameworks for extending the functionality of the tool. It is designed to allow access to all functions of the model/meta model and tool through COM Automation, and it provides extension of menu and option items. Also, users can create their own approaches and frameworks according to their methodologies. The tool can also be integrated with any external tools.

StarUML supports the following diagram types

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- Statechart Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram
- Composite Structure Diagram

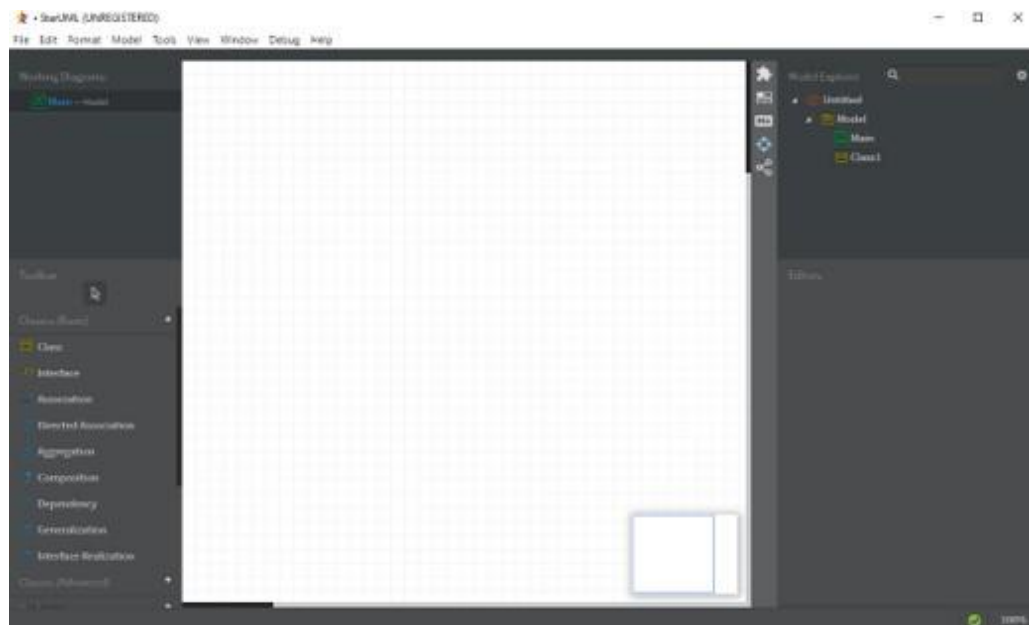
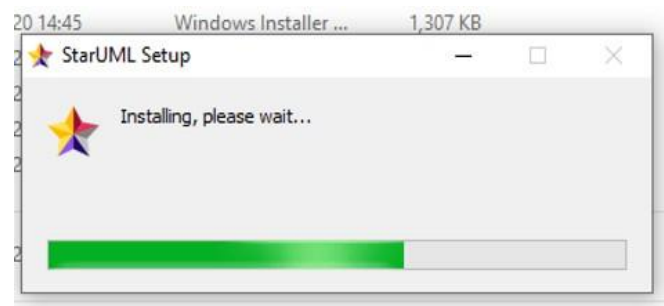
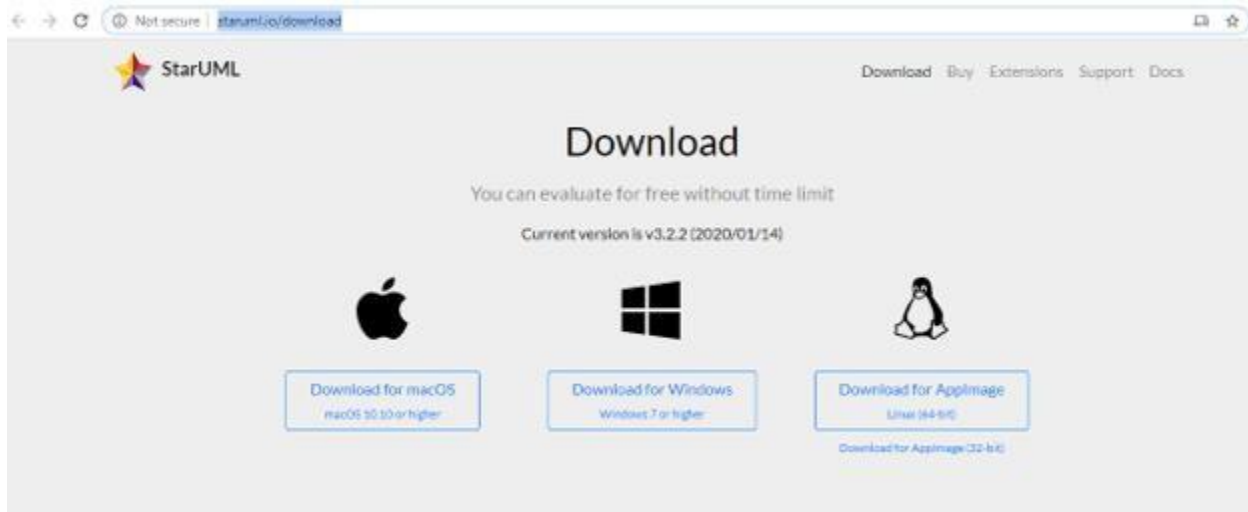
The user interface is intuitive. On the upper right side, a window allows to rapidly navigate between all the content of a project, adopting either a model or a diagram view. Multiple diagrams can be open at the same time and tabs allow switching rapidly between views. The lower right window allows to document the current diagram, either with plain text or attaching an external document. During diagram editing, "wizards" are located around the object that give you the quick shortcuts to main associated tasks with your current operation, like adding an attribute when you create a class for instance. A right-click on the mouse brings the full set of operations at your disposal.

StarUML has also a model verification feature. You can export diagram in different formats (jpg, bmp, wmf). It also supports a patterns approach and import of Rational Rose files.

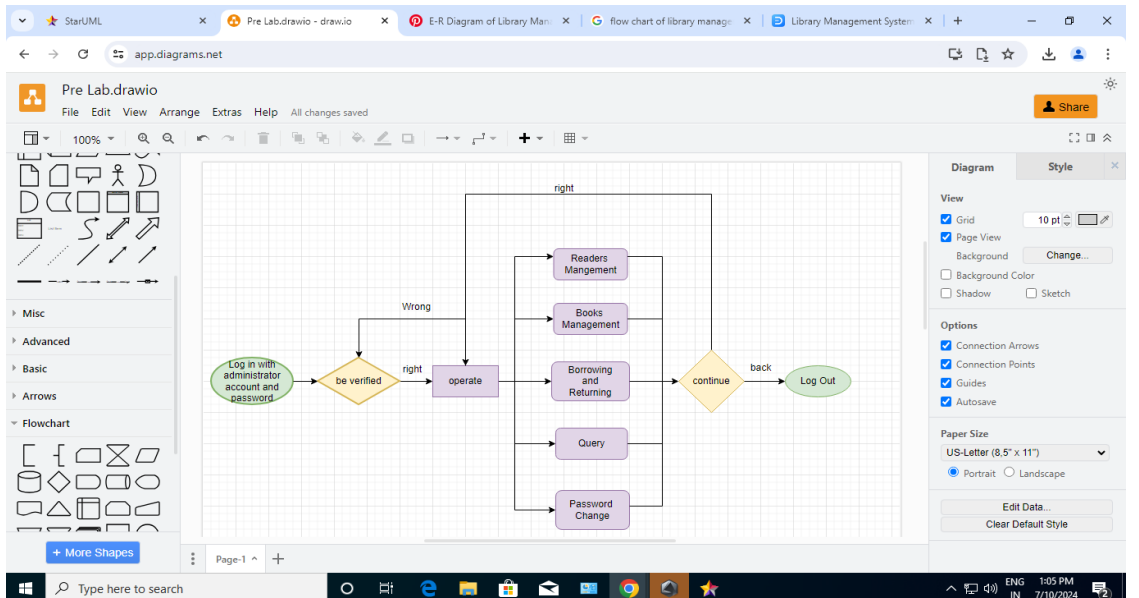
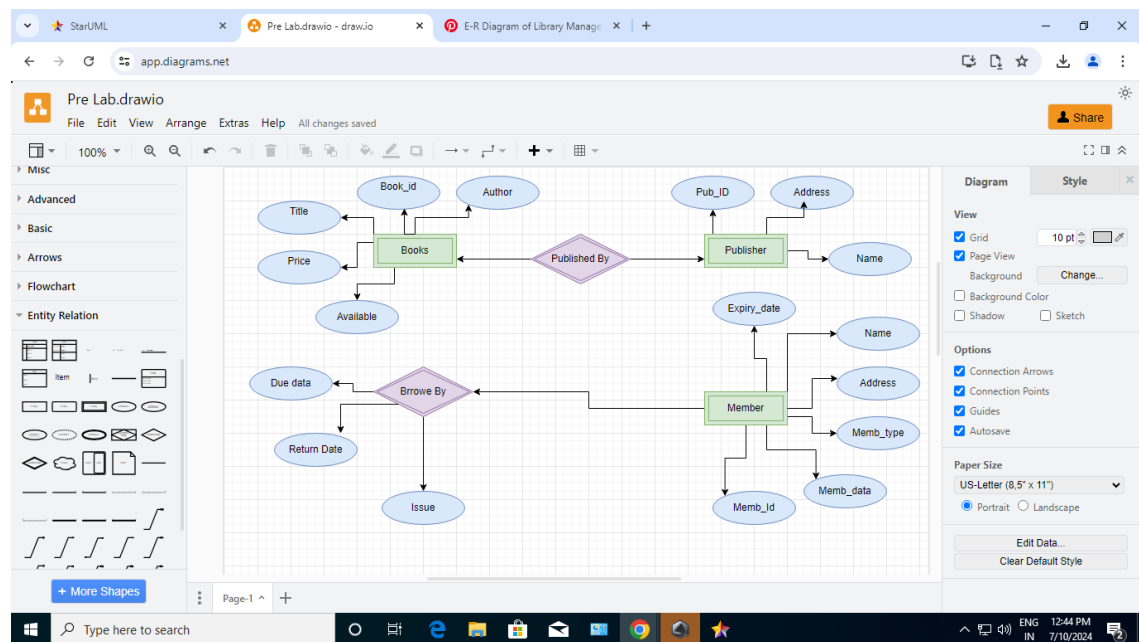
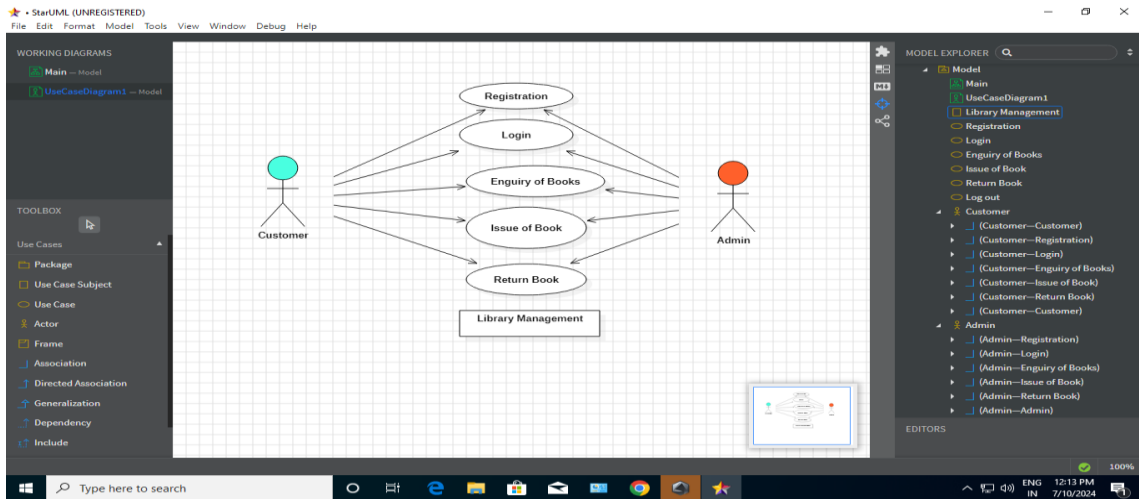
StarUML Generator is platform module to generate various artifacts (like as Microsoft Word, Excel, PowerPoint, and Text-based artifacts) by templates depending on UML model elements in

StarUML. The users can define their own templates and can apply many different kinds of templates to the same UML model, so the users can get various artifacts automatically, easily and fast. The tool supports code generation and reverse engineering for Java, C# and C++.

Installation Steps :



Software Engineering & Quality Assurance Lab (PCCCS504P)



CONCLUSION:

StarUML has many powerful features and is certainly more than a "simple" diagramming tool. With its support of MDA (Model Driven Architecture), it is more aimed at people using UML in an intensive way and with some code generations objectives than for simply drawing diagrams to document requirements. However, using StarUML just as a diagramming tool work fine, especially on Windows as the tool is built with Delphi and might execute faster than the Java based tools.

DISCUSSION QUESTIONS

Is StarUml Open Source?

What is purpose of StarUml?

How many diagrams are there in StarUml?

Explain model in StarUML.

Explain the features of StarUml.

REFERENCE:

- 1) <https://staruml.io/faq>
- 2) <https://www.educba.com/staruml/>
- 3) <https://sourceforge.net/projects/staruml/>



**S. B. JAIN INSTITUTE OF TECHNOLOGY,
MANAGEMENT & RESEARCH, NAGPUR.**

Practical No. 1

Aim: Identify the problem definition and requirement for a given problem statement.

Name of Student	
Roll No	
Semester/Year	
Academic Session	
Date of Performance	
Date of Submission	

AIM: Identify the problem definition and requirement for a given problem statement.

OBJECTIVE/EXPECTED LEARNING OUTCOME:

- Identify ambiguities, inconsistencies and incompleteness from a requirements specification
- Identify and state functional requirements
- Identify and state non-functional requirements

HARDWARE AND SOFTWARE REQUIREMENTS:

Hardware Requirement

- Processor : Dual Core
- RAM : 1GB
- Hard Disk Drive : > 80 GB

Software Requirement

- Operating System – Windows

THEORY:

Requirement's identification is the first step of any software development project. Until the requirements of a client have been clearly identified, and verified, no other task (design, coding, testing) could begin. Usually, business analysts having domain knowledge on the subject matter discuss with clients and decide what features are to be implemented.

Sommerville defines "requirement" as a specification of what should be implemented. Requirements specify how the target system should behave. It specifies what to do, but not how to do. Requirement's engineering refers to the process of understanding what a customer expects from the system to be developed, and to document them in a standard and easily readable and understandable format. This documentation will serve as reference for the subsequent design, implementation and verification of the system.

It is necessary and important that before we start planning, design and implementation of the software system for our client, we are clear about its requirements. If we don't have a clear vision

of what is to be developed and what all features are expected, there would be serious problems, and customer dissatisfaction as well

Characteristics of Requirements:

Requirements gathered for any new system to be developed should exhibit the following three properties:

- **Unambiguity:** There should not be any ambiguity what a system to be developed should do. For example, consider you are developing a web application for your client. The client requires that enough number of people should be able to access the application simultaneously. What's the "enough number of people"? That could mean 10 to you, but, perhaps, 100 to the client. There's an ambiguity.
- **Consistency:** To illustrate this, consider the automation of a nuclear plant. Suppose one of the clients say that if the radiation level inside the plant exceeds R1, all reactors should be shut down. However, another person from the client side suggests that the threshold radiation level should be R2. Thus, there is an inconsistency between the two end users regarding what they consider as threshold level of radiation.
- **Completeness:** A particular requirement for a system should specify what the system should do and also what it should not. For example, consider a software to be developed for ATM. If a customer enters an amount greater than the maximum permissible withdrawal amount, the ATM should display an error message, and it should not dispense any cash.

Categorization of Requirement:

Based on the target audience or subject matter, requirements can be classified into different types, as stated below:

- **User requirements:** They are written in natural language so that both customers can verify their requirements have been correctly identified
- **System requirements:** They are written involving technical terms and/or specifications, and are meant for the development or testing teams

Requirements can be classified into two groups based on what they describe: · **Functional requirements (FRs):** These describe the functionality of a system -- how a system should react to a particular set of inputs and what should be the corresponding output.

- **Non-functional requirements (NFRs):** They are not directly related what functionalities are expected from the system. However, NFRs could typically define how the system should behave under certain situations. For example, a NFR could say that the system should work with 128MB RAM. Under such condition, a NFR could be more critical than a FR.

Non-functional requirements could be further classified into different types like:

- **Product requirements:** For example, a specification that the web application should use only plain HTML, and no frames

- **Performance requirements:** For example, the system should remain available 24x7 .

Organizational requirements: The development process should comply to SEI CMM level

OBSERVATION:

1) Problem Definition :

[illegible]

2) Functional & Non-functional Requirement

Sr.No.	Functional Requirement
1)	
2)	
3)	
4)	

5)	
----	--

Sr.No.	Non-Functional Requirement
1)	
2)	
3)	

4)	
5)	

CONCLUSION:

DISCUSSION QUESTIONS:

What is Requirement?

Explain Problem Definition of given task.

What is Functional Requirement?

What is Non-Functional Requirement?

Differentiate between Functional & Non-Functional Requirement?

REFERENCE:

- <http://vlabs.iitkgp.ernet.in/se/1/>
- <https://sites.google.com/view/ait-se/Home/practicals>
- <https://www.javatpoint.com/software-requirement-specifications>