Unit III
Requirement Engineering

IC705 Software Engineering

Objective

The objective of this chapter is to introduce software requirements and to explain the processes involved in discovering and documenting these requirements.

Requirements

The requirements for a system are the descriptions of the services that a system should provide and the constraints on its operation.

These requirements reflect the needs of customers for a system that serves a certain purpose such as controlling a device, placing an order, or finding information. The process of finding out, analyzing, documenting and checking these services and constraints is called requirements engineering (RE)

Client → wants a software

Requirements establish { Brief outline

Sprint

↳ Requirements are not completely clear

Contract Publish

{I want this software to be developed}

Tender

Auction

{ 150,000 | 200,000 | — Rs

Approach 2.5 months

Approach 3 months

↑ domain experience

Ideas

↓

Interested companies will place a bid

↓

finalize the system

↓

Organization

Requirement specifications

Design

Android app to control M.W. oven

- ~~user input~~
- time
- temp.
- output
- Notification

Requirement Engineering

Requirement Engineering

- **System Definition** activities are conducted to create and describe in detail a **System of interest** (SoI) to satisfy an identified need.
- The activities are grouped and described as generic processes, which consist of system requirements definition, system architecture definition, system design definition and system analysis.
- The architecture definition of the system may include the development of related logical architecture models and physical architecture models.
- During and/or at the end of any iteration, gap analysis is performed to ensure that all system requirements have been mapped to the architecture and design.

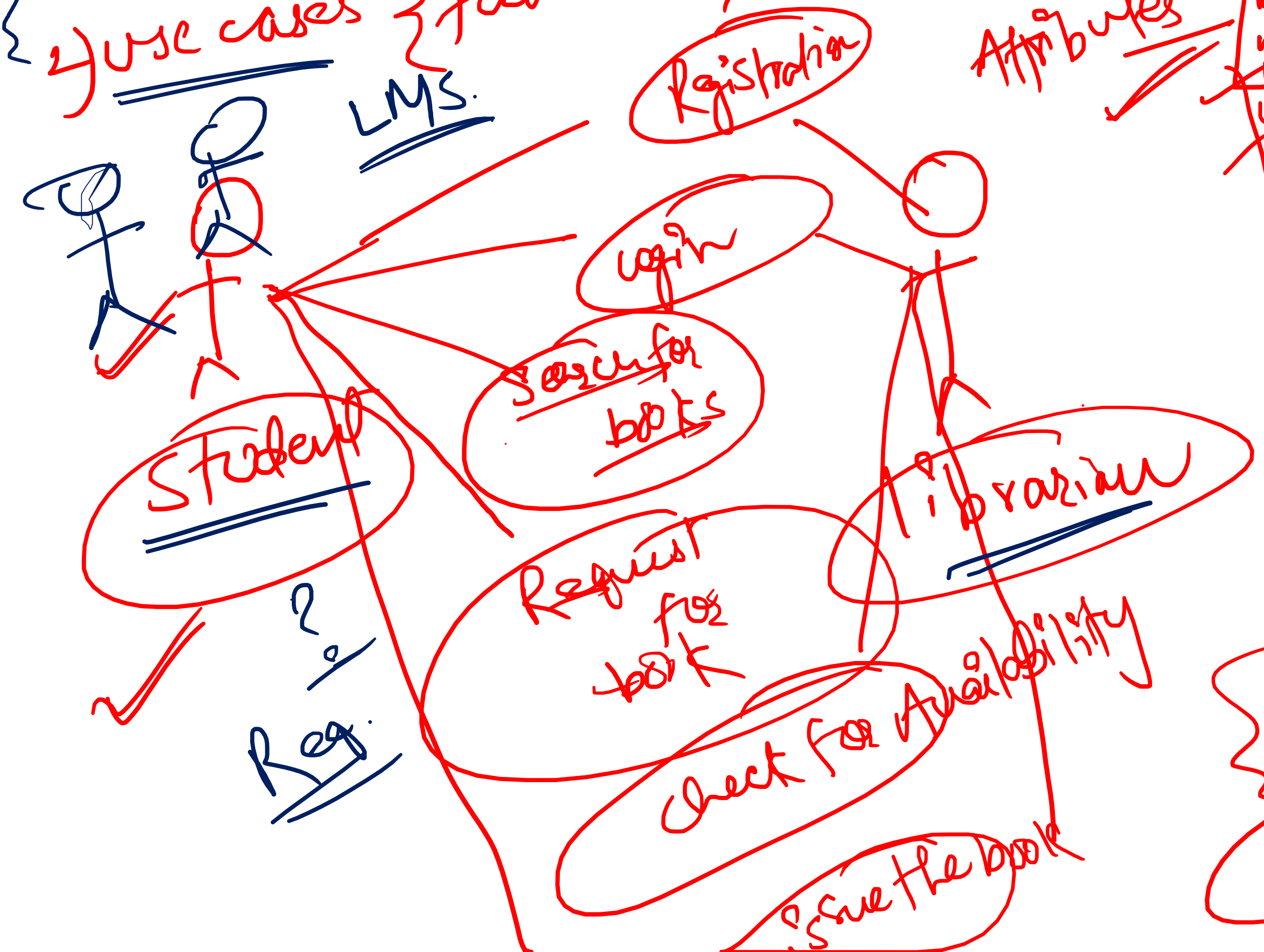
Library Management System

Database — Navathe (4) Elmasri and Forth (5)

Use-case Diagram

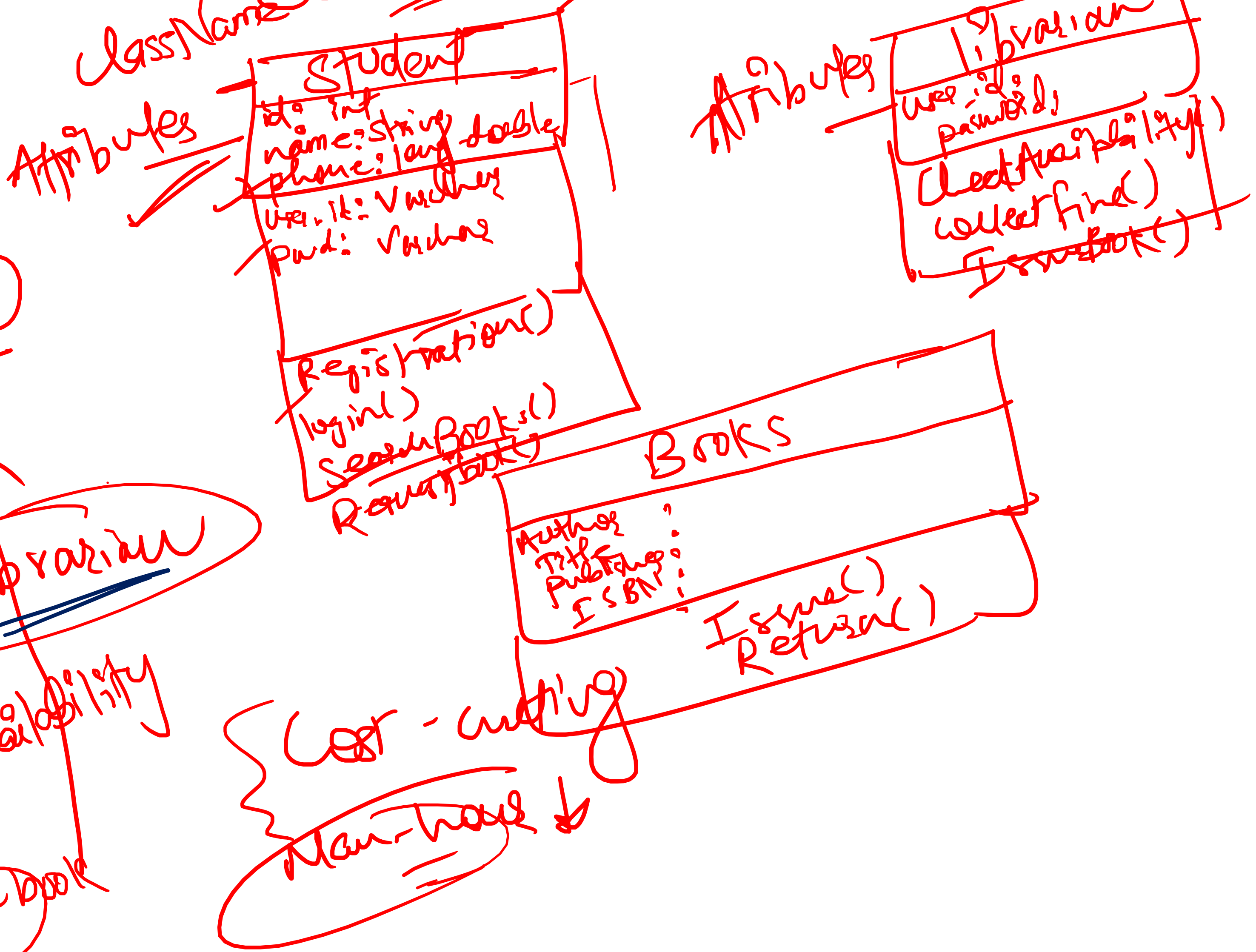
1) Actors {Student, librarian} Brief
2) Use cases {functions}

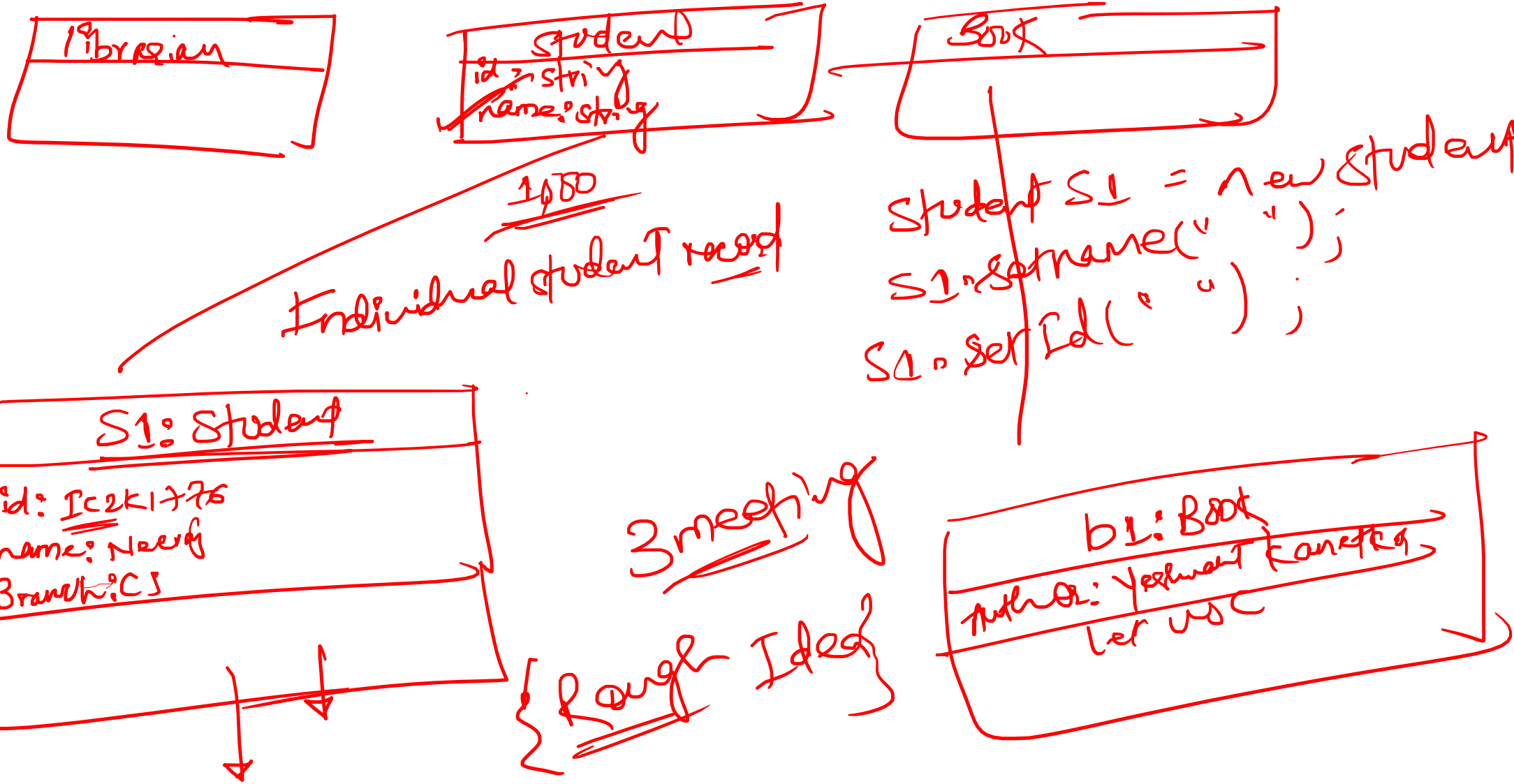
LMS.



Class Diagram

Object

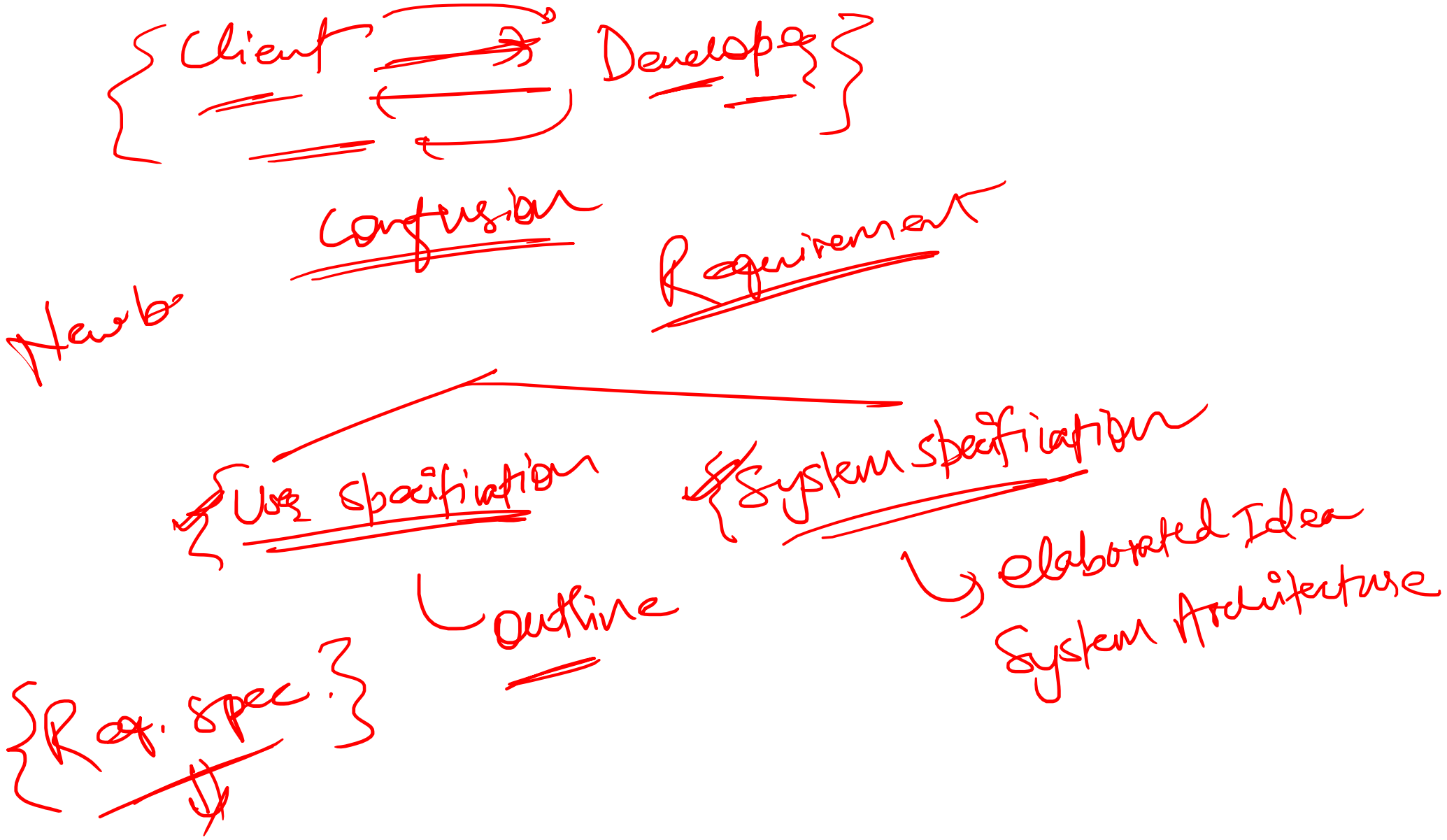



Librarian
Student
id: String
name: String
Book
1000
Individual student record
Student S1 = new Student()
S1.setName(" ");
S1.setId(" ");
S1: Student
id: IC2K1775
name: Neeraj
Branch: CS
3 meaning
{Rough Idea}
b1: Book
Author: Yashwant Kanetkar
Let us C

Requirement Engineering

Ian Sommerville

- *If a company wishes to let a contract for a large software development project, it must define its needs in a sufficiently abstract way that a solution is not predefined.*
- *The requirements must be written so that several contractors can bid for the contract, offering, perhaps, different ways of meeting the client organization's needs.*
- *Once a contract has been awarded, the contractor must write a system definition for the client in more detail so that the client understands and can validate what the software will do.*
- *Both of these documents may be called the requirements document for the system*


Client
Develop
Confusion
Requirement
Newb
User specification
System specification
Outline
elaborated Idea
System Architecture
Req. spec

Requirement Engineering

Some of the problems that arise during the requirements engineering process are a result of failing to make a clear separation between these different levels of description. I distinguish between them by using the term user requirements to mean the high-level abstract requirements and system requirements to mean the detailed description of what the system should do.

Example : Health Care Management System

User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc.) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

Requirement Engineering

- User requirements are statements, in a natural language plus diagrams, of what services the system is expected to provide to system users and the constraints under which it must operate. The user requirements may vary from broad statements of the system features required to detailed, precise descriptions of the system functionality.

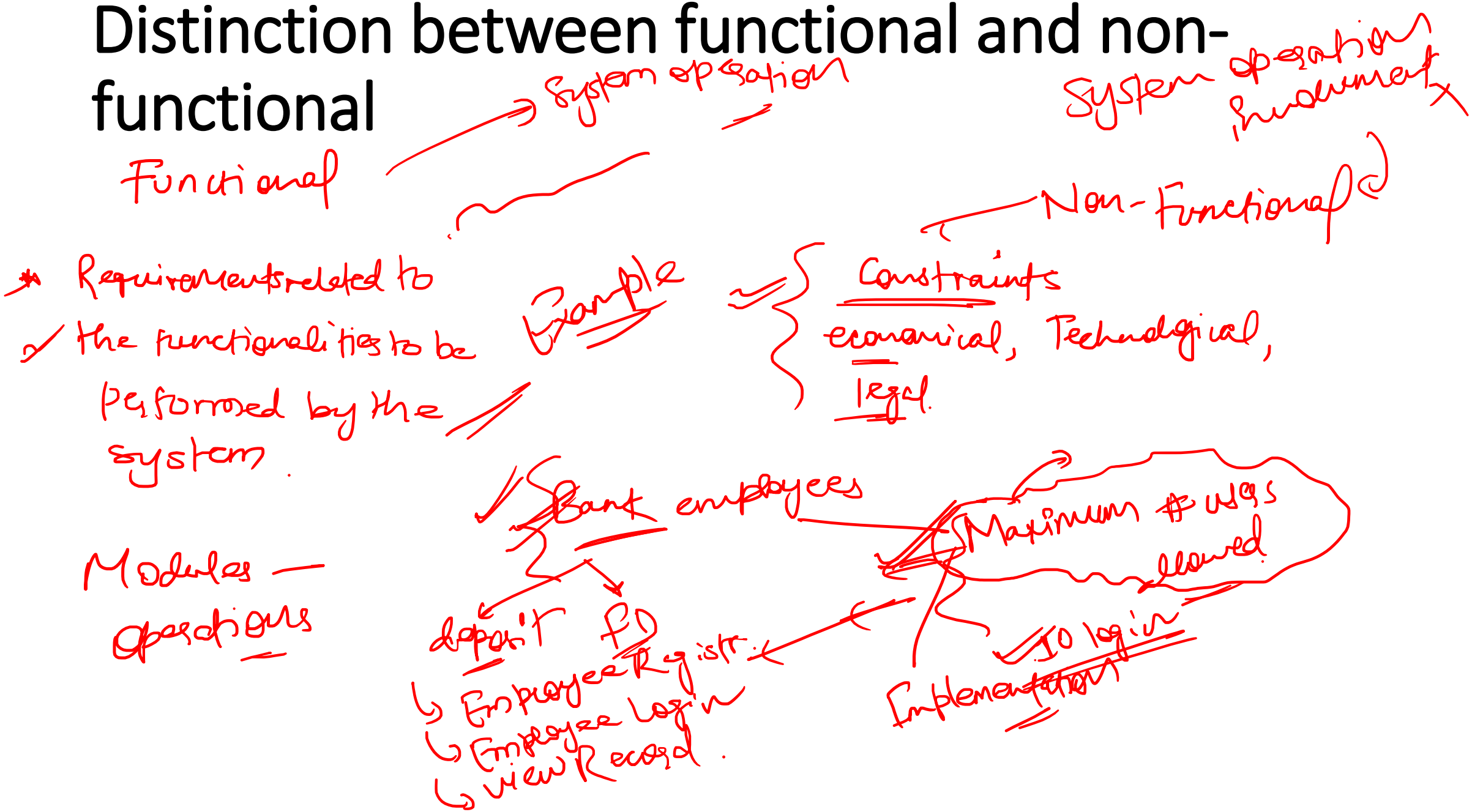
- System requirements are more detailed descriptions of the software system's functions, services, and operational constraints. The system requirements document (sometimes called a functional specification) should define exactly what is to be implemented. It may be part of the contract between the system buyer and the software developers

Functional and Non-Functional Requirements

Functional requirements - These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

Non-Functional Requirements - These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole rather than individual system features or services.

Distinction between functional and non-functional



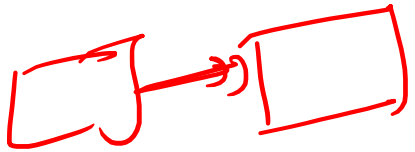
Distinction between functional and non-functional

- In reality, the distinction between different types of requirements is not as clear cut as these simple definitions suggest. A user requirement concerned with security, such as a statement limiting access to authorized users, may appear to be a nonfunctional requirement.
- However, when developed in more detail, this requirement may generate other requirements that are clearly functional, such as the need to include user authentication facilities in the system.
- This shows that requirements are not independent and that one requirement often generates or constrains other requirements.
- The system requirements therefore do not just specify the services or the features of the system that are required; they also specify the necessary functionality to ensure that these services/features are delivered effectively.

Example : Functional Requirement

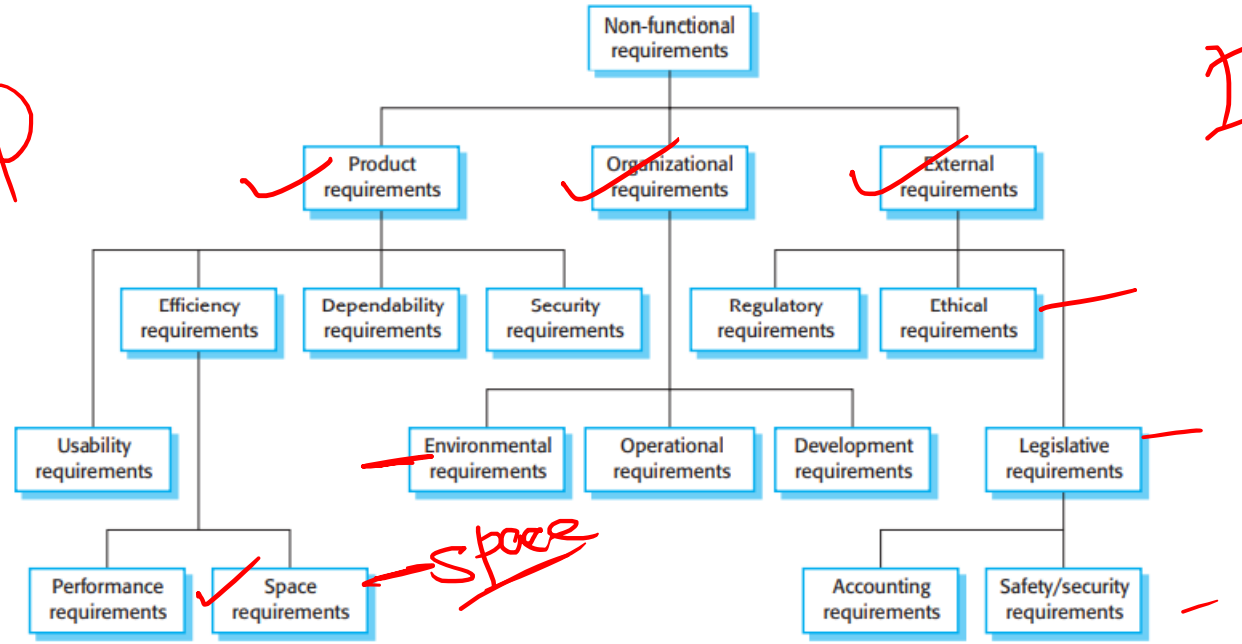
Requirements for the Mentcare system, used to maintain information about patients receiving treatment for mental health problems:

1. A user shall be able to search the appointments lists for all clinics.
2. 2. The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
3. 3. Each staff member using the system shall be uniquely identified by his or her eight-digit employee number



Non-Functional Requirement

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users. These non-functional requirements usually specify or constrain characteristics of the system as a whole.



Non-Functional Requirements

Nonfunctional requirements arise through user needs because of budget constraints, organizational policies, the need for interoperability with other software or hardware systems, or external factors such as safety regulations or privacy legislation.

Example – Mentcare

Medical staff shall be able to use all the system functions after two hours of training. After this training, the average number of errors made by experienced users shall not exceed two per hour of system use.

Requirement Specification

Requirements specification is the process of writing down the user and system requirements in a requirements document. Ideally, the user and system requirements should be clear, unambiguous, easy to understand, complete, and consistent. In practice, this is almost impossible to achieve.

User requirements are almost always written in natural language supplemented by appropriate diagrams and tables in the requirements document. System requirements may also be written in natural language, but other notations based on forms, graphical, or mathematical system models can also be used.

Requirement Specification

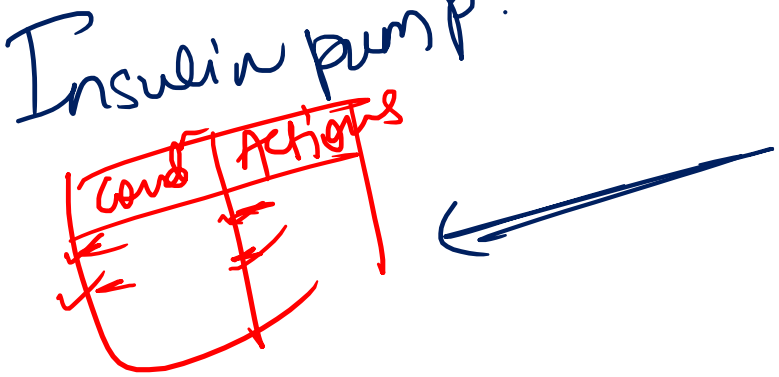
Notation	Description
Natural language sentences	The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system. UML (unified modeling language) use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't understand a formal specification. They cannot check that it represents what they want, and they are reluctant to accept it as a system contract. (I discuss this approach, in Chapter 10, which covers system dependability.)

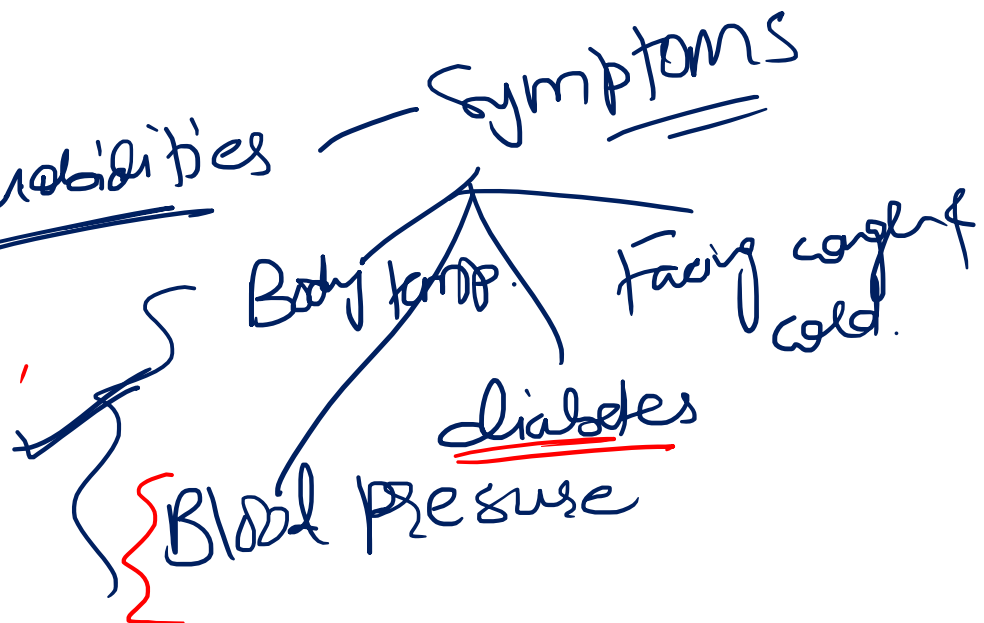
Requirement Specification Using Natural Language

3.2 The system shall measure the blood sugar and deliver insulin, if required, every 10 minutes. (*Changes in blood sugar are relatively slow, so more frequent measurement is unnecessary; less frequent measurement could lead to unnecessarily high sugar levels.*)

3.6 The system shall run a self-test routine every minute with the conditions to be tested and the associated actions defined in Table 1. (*A self-test routine can discover hardware and software problems and alert the user to the fact the normal operation may be impossible.*)

Example requirements for the insulin pump software system





Requirement Specification Using Structured Natural Language

When a standard format is used for specifying functional requirements, the following information should be included:

1. A description of the function or entity being specified.
2. A description of its inputs and the origin of these inputs.
3. A description of its outputs and the destination of these outputs.
4. Information about the information needed for the computation or other entities in the system that are required (the “requires” part).
5. A description of the action to be taken.
6. If a functional approach is used, a precondition setting out what must be true before the function is called, and a postcondition specifying what is true after the function is called.
7. A description of the side effects (if any) of the operation.

Example

Structured Natural Language

Insulin Pump/Control Software/SRS/3.3.2	
Function	Compute insulin dose: Safe sugar level.
Description	Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.
Inputs	Current sugar reading (r2), the previous two readings (r0 and r1).
Source	Current sugar reading from sensor. Other readings from memory.
Outputs	CompDose—the dose in insulin to be delivered.
Destination	Main control loop.
Action:	CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered. (see Figure 4.14)
Requires	Two previous readings so that the rate of change of sugar level can be computed.
Precondition	The insulin reservoir contains at least the maximum allowed single dose of insulin.
Postcondition	r0 is replaced by r1 then r1 is replaced by r2.
Side effects	None.

Area of circle
Program to area of circle
Function: computing the area of circle
Description: computes the area of circle provided the reading
Input: radius of circle : π
Source: User input through keyboard.

1) E - Ticketing system — Flight, Railway
2) Exam control.

E-Ticketing System

Online Ticket Booking System

Function: Facilitates online Ticket Booking

Description: It's the use make bookings provided the date of travel.
# of passengers

Input: Source: Destination: date of travel: # passengers
INJN

Output: Confirmed Reservation Ticket.