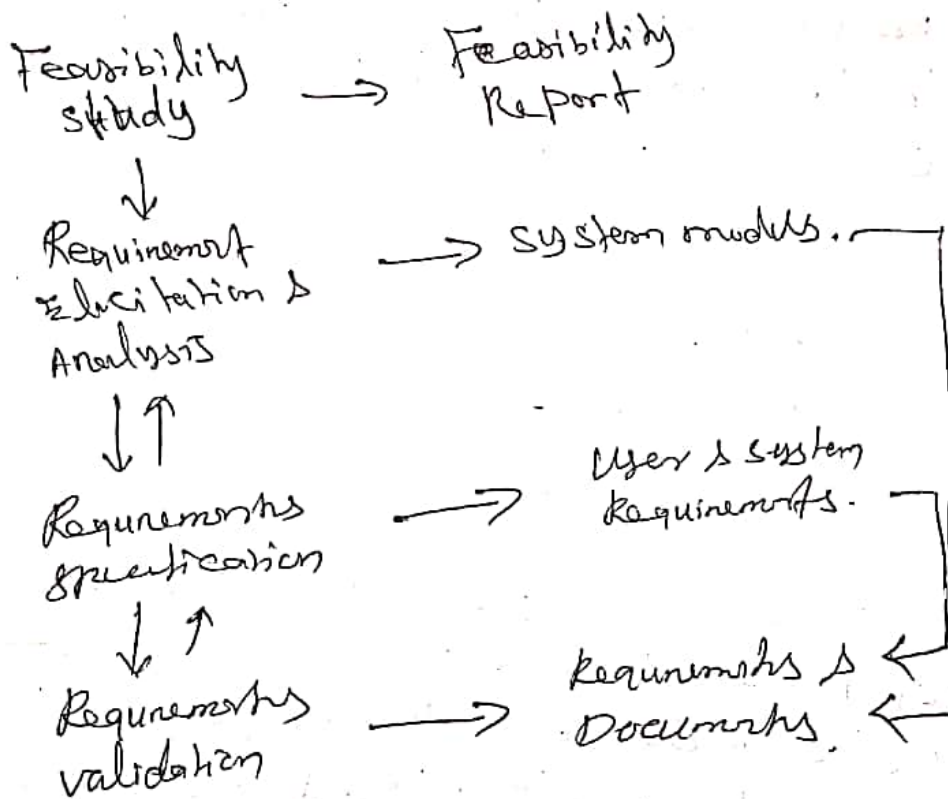


# 1. Requirement engineering process

SE  
U-2

The primary objective of requirement engineering process is to design and maintain a document related to system requirements.

The principle requirements engineering activities and their corresponding relationships are shown in the following fig.



## Feasibility Study:

It is the first phase of requirement engineering process, it specifies whether the system is capable of satisfying the business requirements.

## Requirements Elicitation and Analysis.

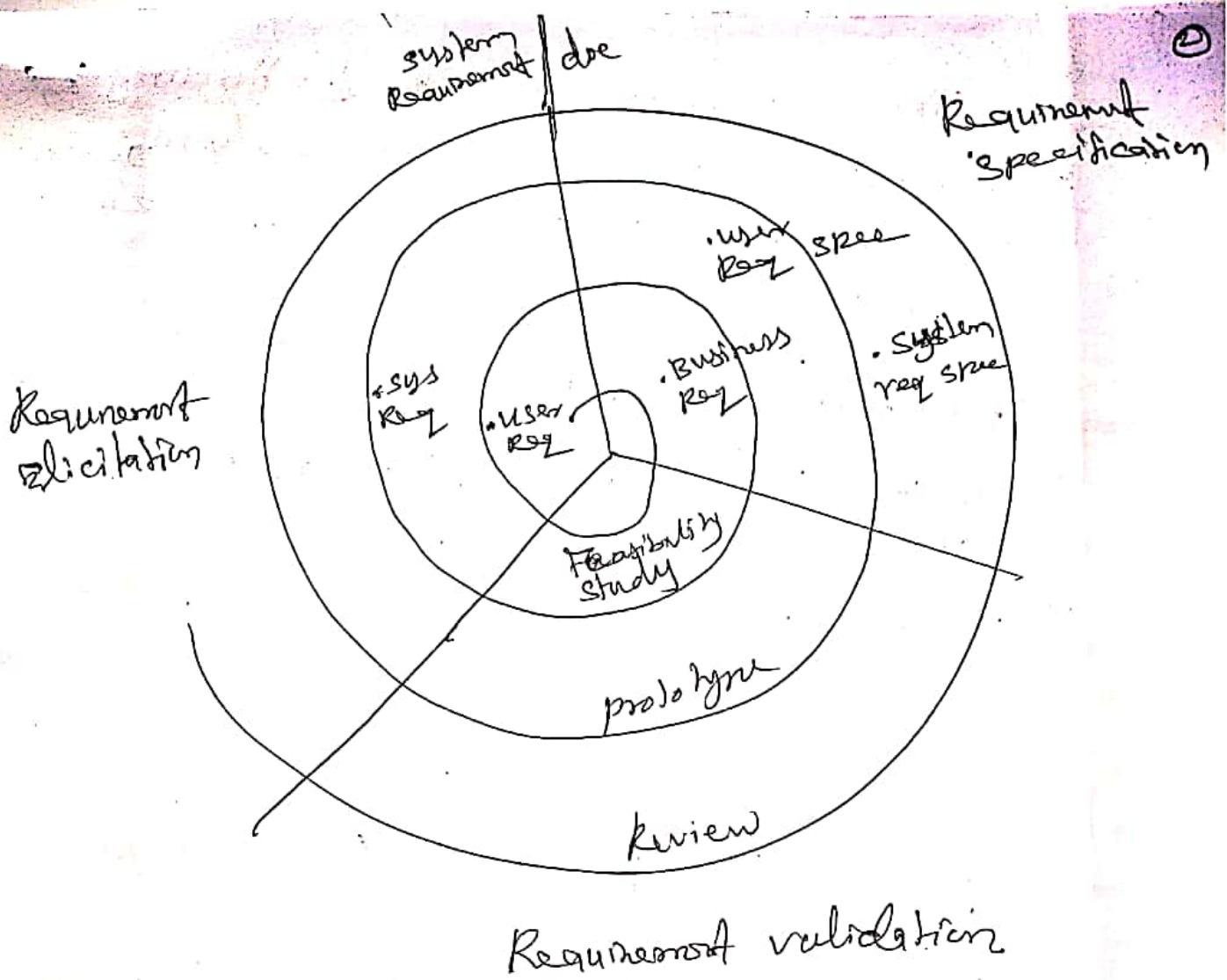
It is the second phase of requirement engineering process. In this phase the developers interact with customer and discuss various issues that might occur in the process of system development.

## Requirements Specification:

The main objective is to properly state the requirements. From the knowledge gained during problem analysis, the redundant information and the knowledge generated from analysis is properly organized and described in the requirements specification activity.

- \* Discovery
- \* Documentation.
- \* Checking.

The core of all processes will be the determination of business, functional and non functional specifications. Hence the developers usually pay higher attention at this ~~stage~~ stage, while other intermediate process becomes lesser to the final phase user review.



②. Explain in detail about Functional and non Functional Requirements.

Functional Requirements:-

It must describe the functionality of system.

The functionality of any system depends on the type of software being developed and the end users who will use it.

The scenario that suits the functionality of a system depends on the purpose of developing the software, for example consider online pizza ordering systems.

- i) It must be easy to browse
- ii) The user must be provide with a unique id after ordering pizza
- iii) The user must be capable of selecting the right pizza
- iv) The system can provide facilities to the user where they can choose their favorite ingredients in the pizza
- v) The system can provide the facility to user where they can pay credit/debit cards in a more ~~safe~~ secured manner.
- vi) The system should provide the facility to users where they can choose their place of delivery.

\* Completeness

\* Consistency

\* Non-Functional Requirements:

non functional requirements are used to address the issue like availability, reliability,

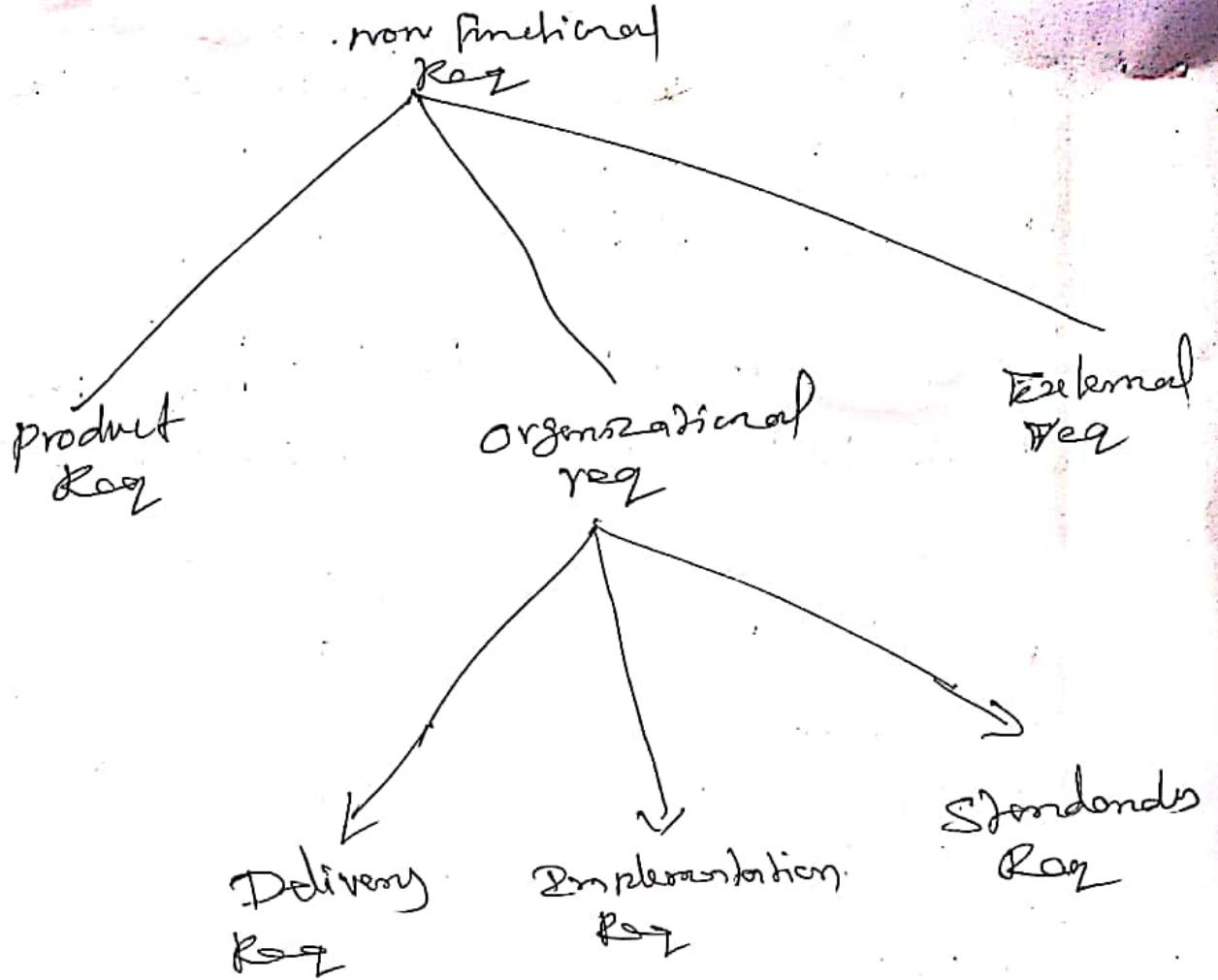
Performance, security, and various other related issues. They do not correspond to the system requirements. ③

- a) Pizza must have home-made appearance
- b) The package of the order must be attractive

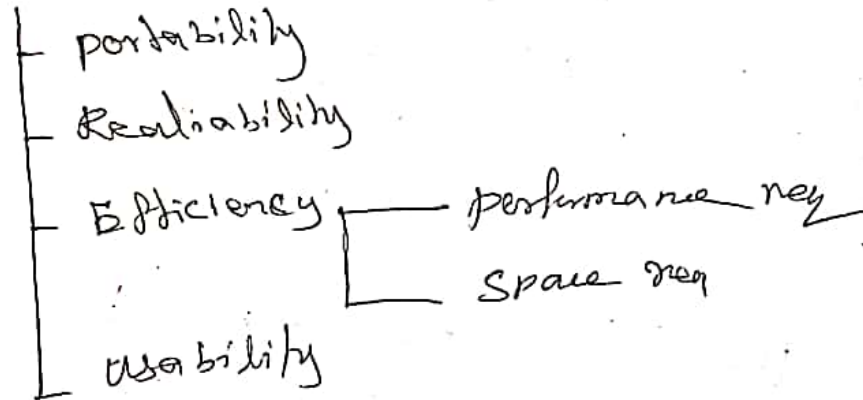
They deal with end product, the system which has to be developed along with the manufacturing process. The reasons behind the outcome of non functional requirements are listed below.

- 1) Budget constraints
- 2) Inadequate user needs
- 3) If there is requirements of one system to be operable with other software/hardware
- 4) Due to strong privacy & safety matters.
- 5) Strong organizational policies etc . . .

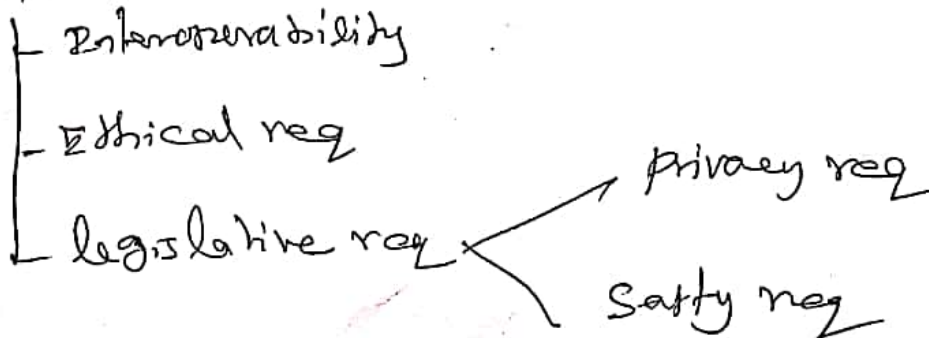
The following hierarchical tree depicting various types of non functional requirements.



Product Req :-



External req



③ Discuss in detail about user requirements and System Requirements, Also Differentiate between them. ④

### User Requirements:-

User Requirements are general statements written in natural language describing about what the system is expected to do.

They can also be expressed by using interaction diagrams in some cases. User requirements are usually read by non-technical users. So the user requirements document should only specify external behavioral aspects of the system ~~is~~ instead of internal implementation details as it may confuse the users.

As user requirements are expressed in natural language, certain problems can arise while understanding the requirement specification.

#### 1. Requirements Amalgamation.

This requirement arises when several requirements are grouped together and specified as ~~single~~ single requirements.

#### 2. Lack of clarity.

When the requirements are not specified in a precise, and unambiguous way, then clarity is said to be lacking.

2. Requirement Confusion!  
This problem usually arises when the developer does not specify clear distinctions b/w F & NF requirements.

### System Requirements:

System requirements is the initial step in the system designing process. They are the extension of user requirements they provides detailed description regarding software implementation.

The specification present in this requirements should be complete and consistent. These requirements describe the operational and behavioural aspect of the system.

The reason to specify design detail in system requirements are as follows,

1. The design details should be considered if the system is distributed in nature.
2. The system architecture should be to check whether the system is satisfying the non functional requirements.
3. The developers specify requirements in technical lang which become difficulty for user understanding
4. The requirements can<sup>not</sup> be included under different topics as side headings.
5. The developers use certain terms for specific situations. These terms may be mean different in different concepts. As a result the end users may fail to understand the actual requirements.



4. What is requirement elicitation? Briefly describe the various activities performed in requirements elicitation phase with an example of a watch system that set time & alarm.

The follows requirements to be discovered for setting time and alarm in a watch.

1. The watch must be user friendly for setting and changing time
2. It should contain simple and minimal interface with light support for visibility
3. It must provide alarm facility with a finite set of alert tones.
4. It must include an option of selecting either 24 hrs or 12 hrs time format
5. The display of time must include second counter along with DD MM YYYY.
6. The watch must have minimum 5 years of guarantee

Requirement classification and organizational functional requirements.

1. Time settings
2. Easy interface
3. Alarm facility
4. Time format
5. Date & time display
6. Alert tones

Non Functional requirements:

1. Light support
2. minimum 5 yrs guarantee
3. water resistant
4. Shock resistant.

Requirements prioritization and negotiation.

Functional requirements.

1. Easy interface
2. Date & time display
3. Time format option
4. Time setting
5. Alarm setting.

Non Functional requirements.

1. light support
2. water resistant
3. shock resistant
4. minimum 2 yrs guarantee

Requirement specification.

1. The watch must have an easy interface for performing actions like setting the date, time & alarm.
2. It must have options for selecting time format
3. The light support must be good enough to make the time, date & digits clearly visible
4. The watch must be water and shock resistant
5. It must have minimum 2 yrs guarantee

5. Discuss in detail about petri nets.

6

Petri nets,

Petri nets are graphical representation used for analysing and modeling a system. They were developed by Carl Adam Petri. The graphical representation is simple and is suitable for describing system status and dynamic behavior.

They are mainly used for simulating and validating systems.

$$PN = \{ P, T, A, w, M_0 \}$$

P - Finite set of places

T - Finite set of transitions

A - Finite set of arcs

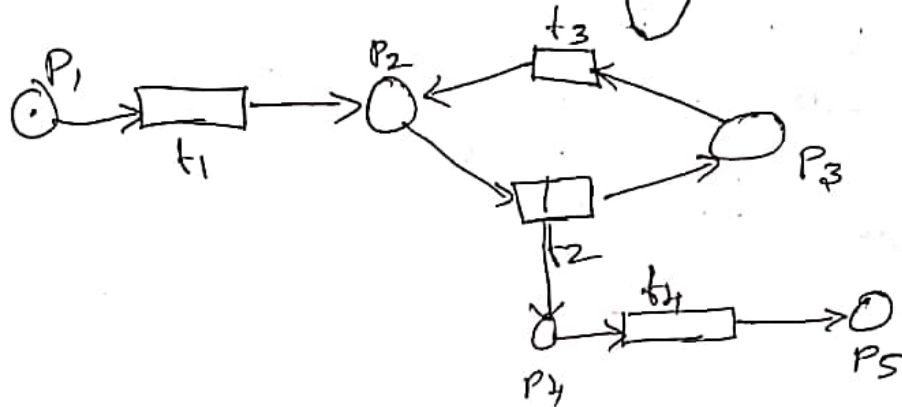
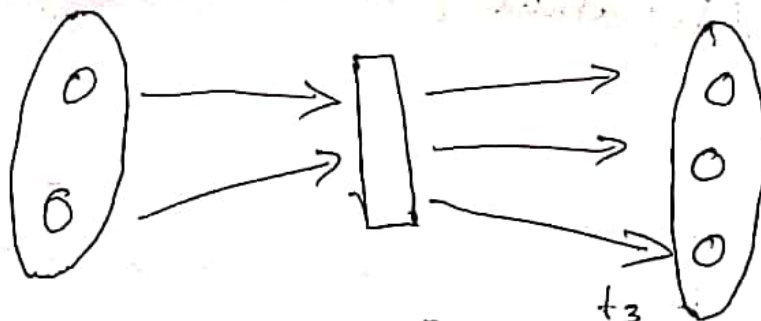
w - weight function

M<sub>0</sub> - initial marking

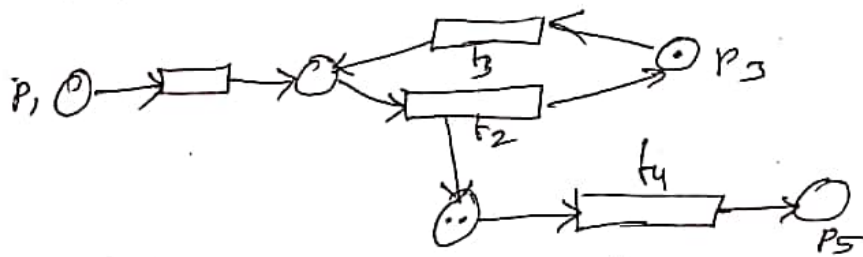
Place and transitions are disjoint sets, P ∩ T is an empty set.

$$P \cap T = \{ \}$$

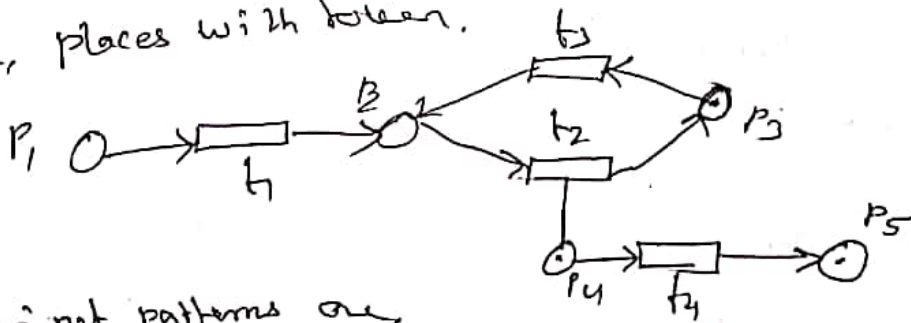
1. Place, Transition, Flow relation, Token
- \* Place denote the state/condition which needs to be met before the action is to be performed.
- \* Transition, the action to be performed in the system.
- \* Flow relation denotes the flow between states/ places and transitions.
- \* Token represents the active state in the system, it can be denoted / represented by a dot in place / state.



The above Petri net consists of 5 places and 4 transitions  
 i.e.  $P_1, P_2, P_3, P_4, P_5$  &  $t_1, t_2, t_3, t_4$



Finally, after firing of transition  $t_4$ , token will be moved from  $P_4$  to  $P_5$  and places  $P_3, P_4$  &  $P_5$  are said to be enabled  
 i.e. places with token.



Petri net patterns are

1. Simple sequential pattern
2. Branch pattern
3. Concurrent pattern
4. Merge pattern

6. Explain in detail the various components of SRS (7)

Software Requirement specification document is the final outcome that comes after problem analysis. But both problem analysis and SRS are carried out simultaneously due to which activities to the other.

The formal modeling performed in problem analysis is not treated as SRS as it lays stress on the problem structure than the external behaviour. The things like modeling user interfaces, minor issues, performance, design constraints, recovery etc... On the other hand, it is difficult for some structures to be translated into external behaviour specification due to their limited use.

Object diagram during OO Analysis.

The knowledge about the system that is obtained as an outcome of analysis activity and is passed to the specification activity.

Characteristic of an SRS.

There are several requirements and properties to be satisfied by the SRS to satisfy the basic goals of the purpose of system. The desirable characteristics are,

Correct, Complete, Unambiguous, Verifiable, Consistent, Stability, modifiable/convertible, Traceable,

Components of an SRS,

It is very difficult to achieve completeness of specification and its verification. There are certain guidelines that help in specifying the complete requirements.

## 1. Functional requirements.

It specifies the operations to be performed on i/p data, output that should be obtained and the relationship between i/p & o/p data of the system.

The operations specified by the functional requirements should include the parameters, equations & logical operations.

## 2. Performance requirements.

It specifies the constraints related to the performance of the system. The performance requirements are two types,

- \* static requirements
- \* dynamic requirements.

## 3. Design Constraints.

It recognize and specify all those factors that impose restrictions on the designer while designing a SW.

- a) Standards Compliance
- b) Hardware Limitations
- c) Fault Tolerance and Reliability.
- d) Security.

## 4. External Interface Requirements.

The hardware interface requirement must specify the communication b/w a software product and hardware component.

The features of hardware along with memory restrictions are needed to be specified while executing a software product on the hardware.

7. Consider an online railway reservation system, which allows the users to select route, book/cancel tickets using net banking/credit/debit cards.

Use-case Scenario.

The actions and their responsibilities involving one Actor - User

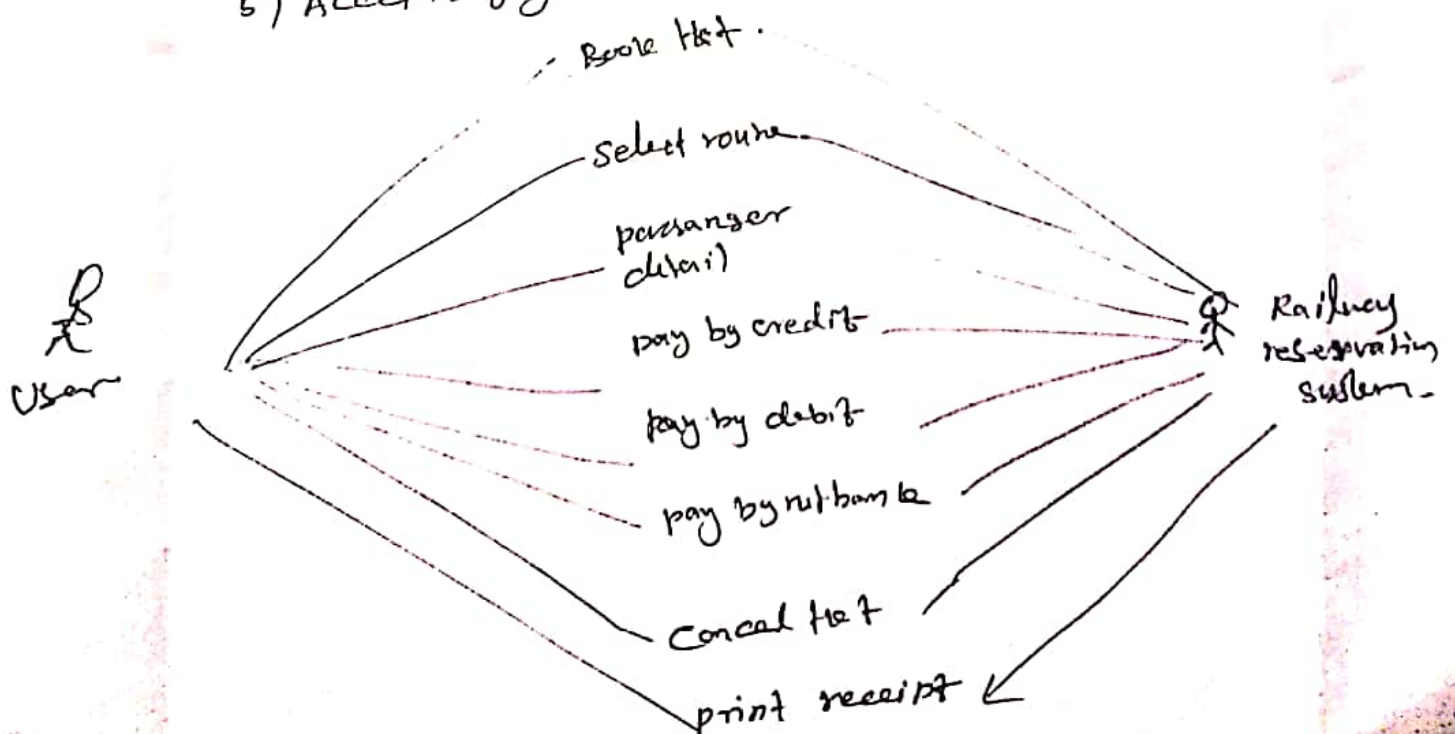
Responsibility:

- 1) book tickets (request)
- 2) select route (source & destination)
- 3) provides passenger details
- 4) pay fare through credit/debit/net banking
- 5) Cancels ticket (request)

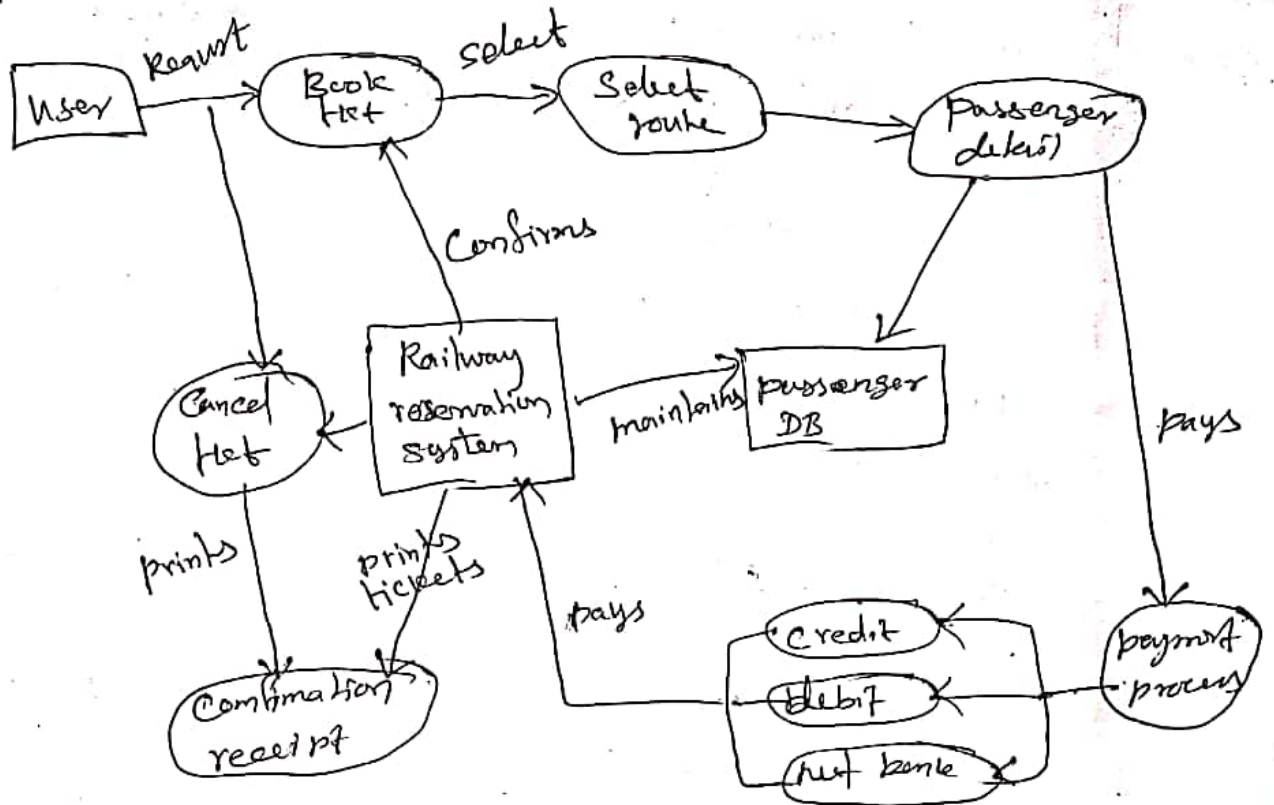
Actor - Railway reservation system software.

Responsibilities.

- 1) Books ticket (confirmation)
- 2) maintains the history of passenger in the database
- 3) Confirms payments and prints receipts
- 4) Cancels tickets (confirmation)
- 5) Accepts payments through net banking/debit/credit.



# DFD



processes.

- 1) Book ticket (request)
- 2) Selects route
- 3) provides passengers details
- 4) Pays fare through debit/credit/net bank
- 5) Cancels ticket (request).