

What is software?
Hardware vs Software

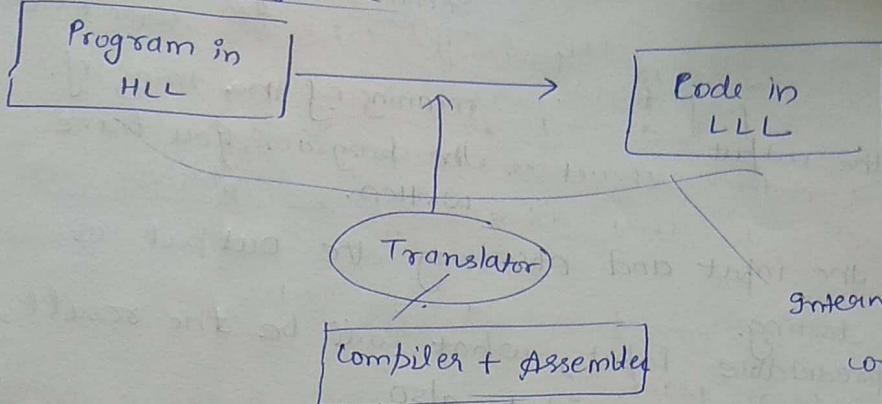
Software Engineering.

class-1
21/1/2018

HLL

LLL

Assembly
Language



Intermediate program that connects HLL to 0's and 1's.

Set of instruction is called program.
Collection of programs is called software.

Translators are written in programs. They are softwares.

How HLL is created.

Form words from the character set,

Apply some rule on it. → grammar in HLL
form sentences.

Software

Engineering

What is?

When you are following systematic steps along with some present knowledge and solve application of our knowledge

some ~~we~~ problem.

Before developing software

1.) First understand the problem, (Ex - what is factorial?)

2.) Write down steps needed to perform.

(Ex - Algo)

3.) Convert it into program.

4) Test it.

Syntax - rules (compiler check)

Semantic → (Compiler check to
↓
some extent)

See the output.

meaning of the thing of
the program, you have
written,

Giving the input and observing the output is
called testing.

For unpredictable input, what will be the result.

→ test for the wrong input also.

* Step by step development of software is called
Software Engineering.

We also write a document for software,
↓
very important.

Most important part of software development is
human resource.

Team of people who can understand the problem
and need of it which robots can not.

We always say that software development not
software manufacturing.

Set of requirements are different for every solution.

→ Software development lifecycle.

lifecycle of a software.

Software process model
in which order we follow steps

Risk in Software Development

bugs, crash when we are using.
but when we made it, there is also a risk.

SRS → document.

Software Project Management

Planning and Schedule

Function Oriented and Object Oriented Software

Testing

α version, β version

Coding X

Software Reliability

↳ of some unexpected input,
it should not crash
it should do something.

can be achieved to some extent (not 100%)

Quality Assurance

User experience

Books -

Standard - Pressman.

Prof. Sangeeta.

Sangeeta

Book Sabrawal,

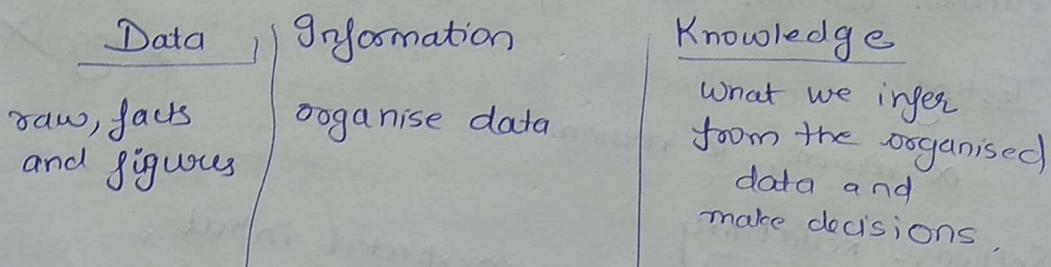
Book.

Software is a set of

class-2

(21/1/2018)

- 1) Instructions that when executed provide desired output and performance.
- 2) It is a data structure that enable the program to manipulate the information.
- 3) It is a document that describes the operation and use of program.



→ Store the data in an organised manner, that will become data structure.

instruction and data structure and document

software,

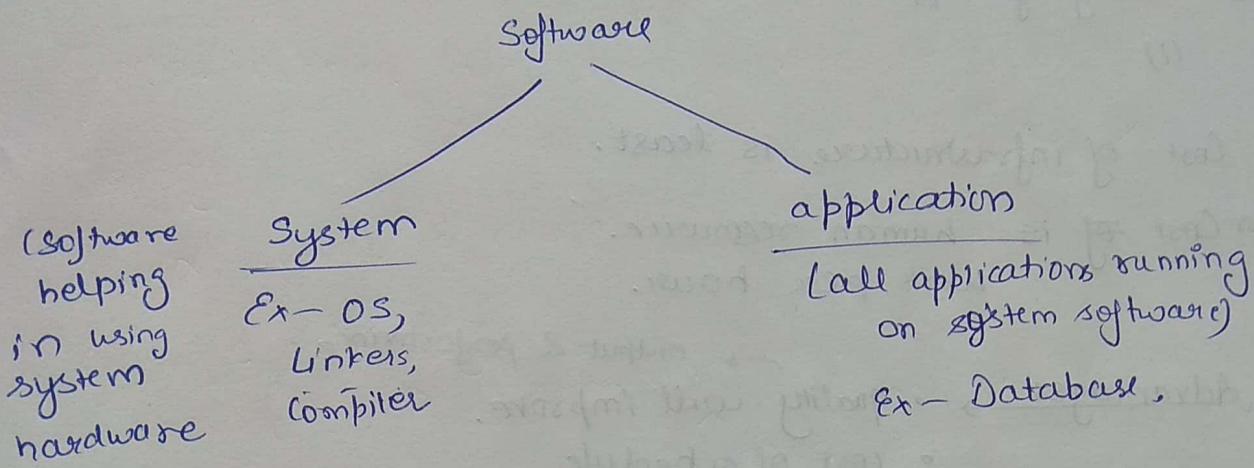
➤ Software Characteristics

- * It is not manufactured, but developed
 - developed
 - wear out
 - custom built
- * Software never breaks, it ~~also~~ always wear out.
- * Specific requirements are there for each software.
(Ex - for library, it is different from bank.)

Software Components → Reusability

→ Program. lang. → vocabulary
→ grammar
→ well formed rules.

Reusability → One software can be in different ways.



real-time software
(time unconstrained)

Ex - online booking
satellite comm.

Embedded software
that are embed in different machines.
Ex - in washing machine

Software related to AI, personal computer,

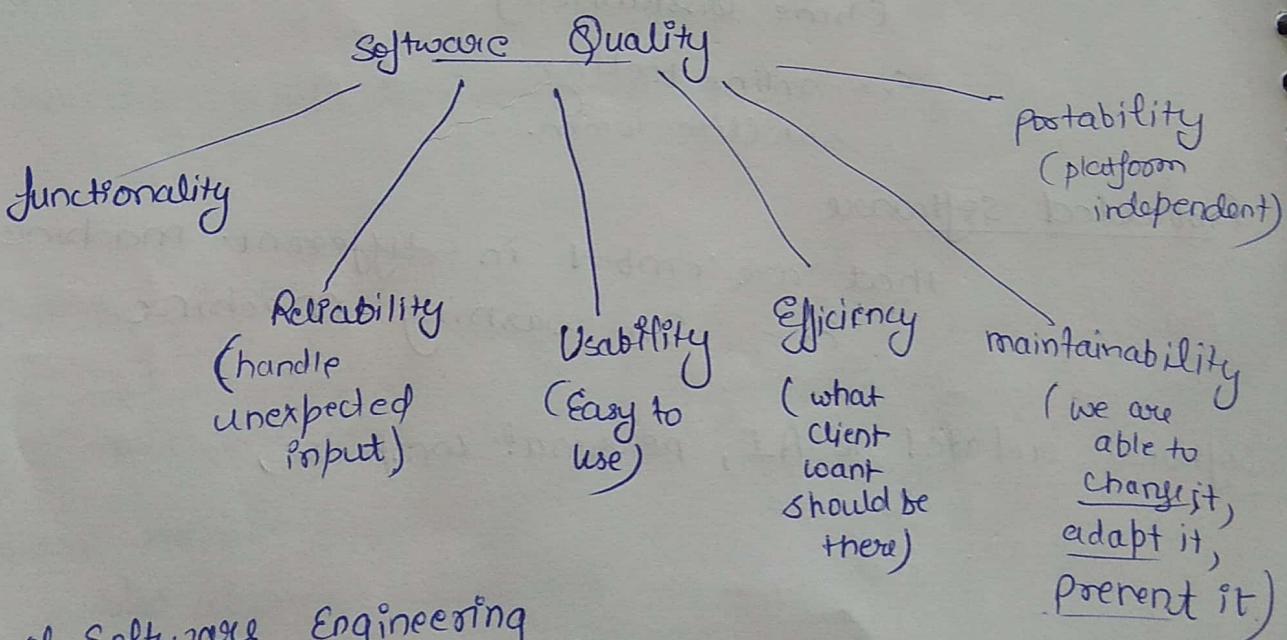
Software Engineering by IEEE

- Q1) 1) It is application of systematic discipline quantifiable approach to the development, operation and maintenance of software i.e. application of engineering to software.
- 2) Study of approaches mentioned in (1)

Cost of infrastructure is least.

Major Cost of is human resource work per hour.

- Advantages → output & performance
- quality will improve,
 - cost of schedule
 - complete tasks within given time & cost.



Need of Software Engineering

come when we have to deal with large software

Q-1) There is a disagreement about the terminology of software engineering. Read about it. Think it.

Q-2) Why do we study Software Engineering?

Q-3) What measurements will you take in a project to measure productivity?

(Velocity) speed of getting the work done

(Because the algo in solving a problem)

③ Program Simplementation

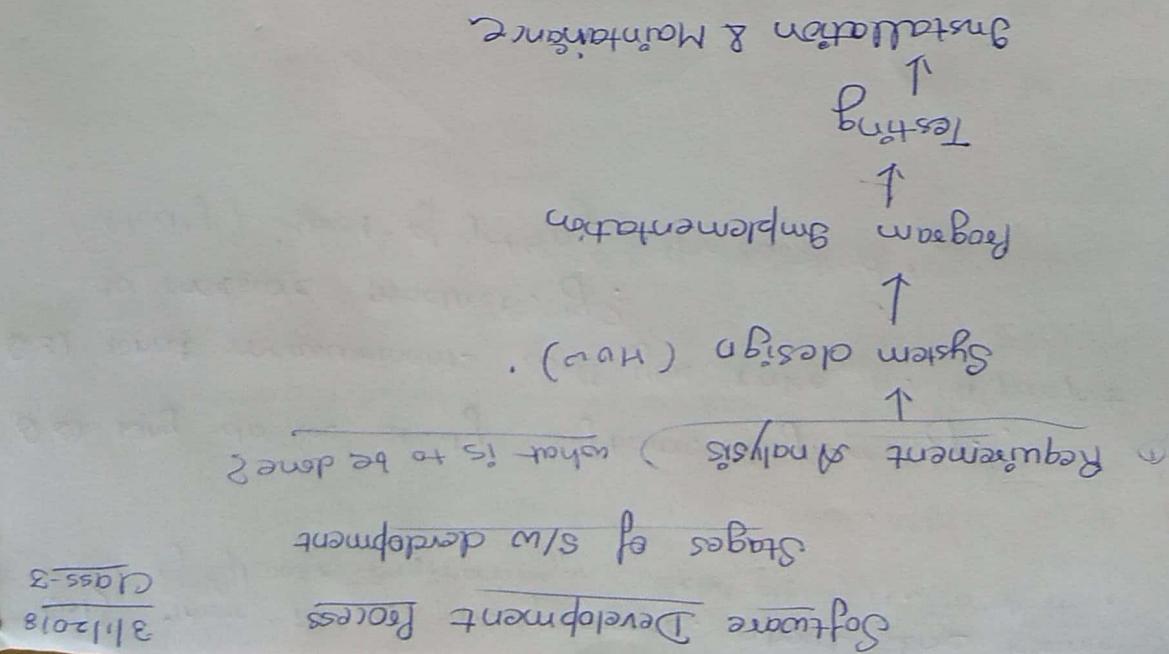
→ which language choose.
→ coding part.

How will you implement the part,
data organised, what interface.
Data structures, how data manipulated,

② System design
Ex - Educational, Scientific domain, health etc.
problem associated and whether it is solving.

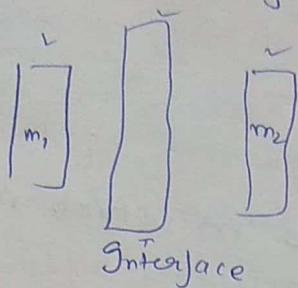
problem domain

Life cycle of the software
conceptualization of idea till the software becomes obsolete.



④ Testing

test the unit as a whole along with the interface and separately also.

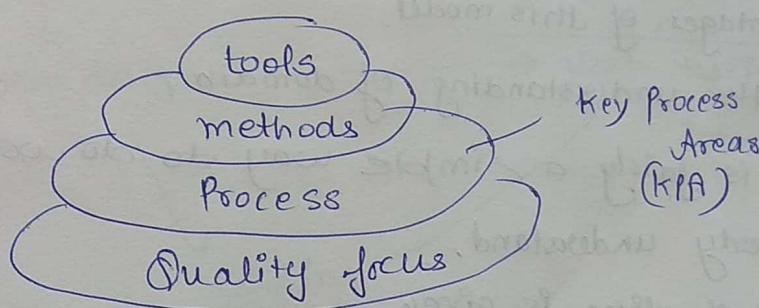


all test separately
and also as
a unit.

⑤ Installation & Maintenance

small changes made. if any errors from users

Software Engineering Principles.



KPA

KPA forms the basis of management control of the software project and establishes context in which technical methods are applied, work products like models, documents, data, reports etc are produced, milestones are established, quality is ensured, and change is properly managed.

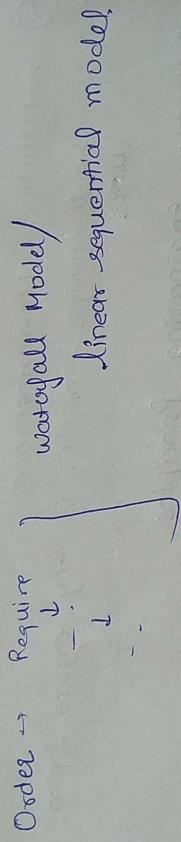
Methods

provides the technical tools on building a software system includes task as requirement analysis, coding, testing and maintenance,

- tools provides the automated and semi-automated support for process and methods.

Process
is a set of activities.

Software process Models
Order of activities in which it is conducted.



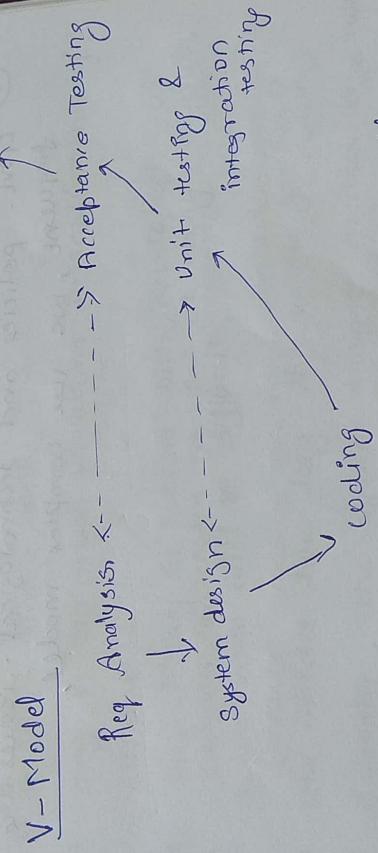
Advantages of this model.

- ① Better understanding of domain
- ② It is fairly a simple way to do work.
- ③ Properly understand.
- ④ enough time is given.
- ⑤ Documentation is made very easily. (+ and -)

Disadvantage

- ① Changing something will be very difficult.
- ② Probability of error propagation is very high.
- ③ To perform it in reality is very difficult. Different team is there for each phase.
In this model, one team will work, and other four team will remain idle.
- ④ No overlapping of phases lead to wastage.
- ⑤ lot of human resources,

- ⑤ When one team is waiting for other is called blocking & stale.
- ⑥ Error can not be changed.
- ⑦ Rigid system for doing work.
- ⑧ There is no scope of customer feedback at any stage. Therefore, customer may not like the product.
- ⑨ Expressing the requirements fully in the beginning is difficult.



Acceptance testing → You can review it yourself
 → ask customer
 → include a third party to check it.

Advantage same as waterfall
 (Difficulty in ~~same~~)

Disadvantages Overlapping is still not there.

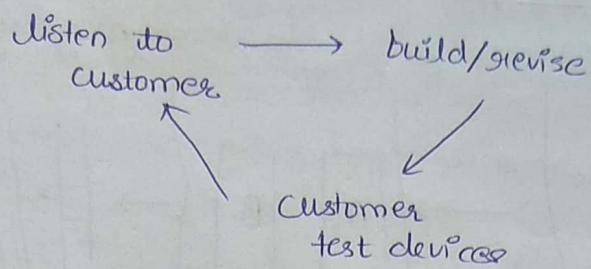
→ How to decide which model to use?

① size of software:
very large scale need complicated model,

- ② If requirements are given be forehand then waterfall model used else if requirement not given, don't use waterfall
- ③ Not sure of technologies to be used, use complex model.
- ④ Govt. policies and technological issues are frequent, we use complex model,

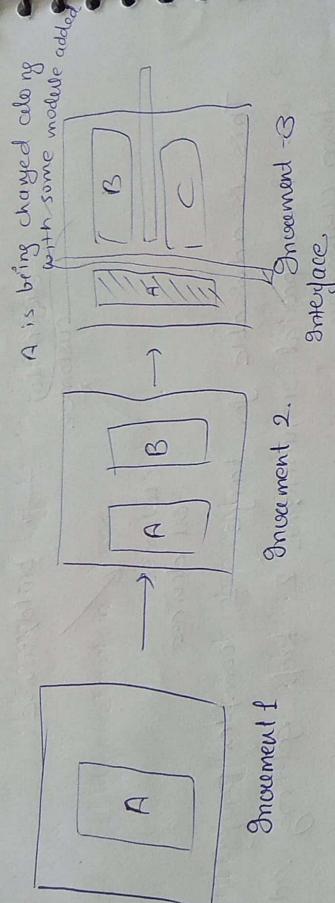
Prototyping Models

class-4
9/12/2018



- This is advisable for large & complex projects.
- we know partial requirements & keep getting advice from customers.
- time consuming, costly.
- may be too much variations in requirements of customers
- has liberty to interact with developer, customer can change his views at every iteration.
- documentation will be difficult.
- who decides how long this process goes on?
- Generally, the system analyst decides the number of iterations,
3-4 times prototyping is done
- if developer gets ~~the~~ feedback in nascent iterations,
he's likely to make minor fixes / and ship it.
sometimes customer may be in a hurry (and ask
for little fixes only)
- Generation ~~base~~ is there, feedback there, but -ve
is you want it to be done quickly & comprising
the quality of software quickly.

Incremental model



Increment 1.

Increment 2.
Interface

Increment 3.

Linear seq. model

+

Prototyping

Say word processing doc.

* Basic word processing functions (A)

* In next increment, you put in some advanced features (B) say foot.

- * In 3rd increment, improving first module & ~~the~~ segregating & making partitions in complete set of requirements & developing the fragments incrementally.
- * Developing ~~the~~ Interface takes less time
- testing interface independent & in combination

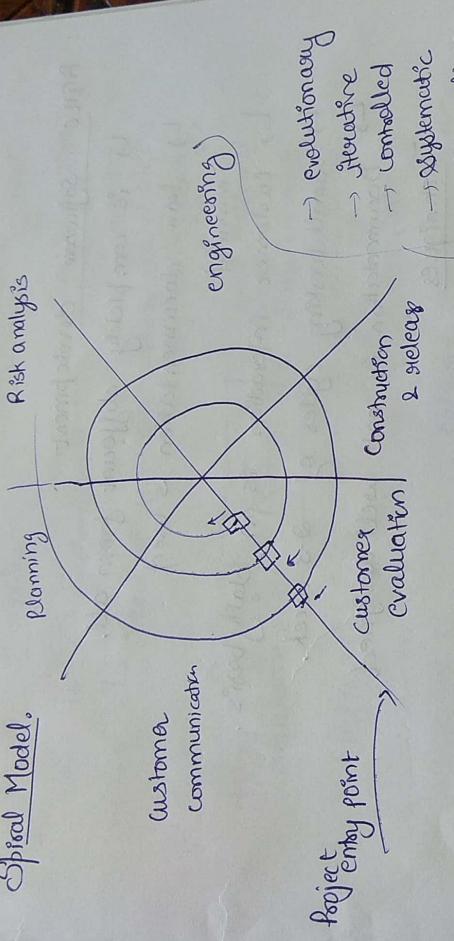
Advantages

- easy
 - customer feedback, improvement of modules (users and inform. about bugs)
 - we need smaller teams (less biased, less cost).
-
- ### Disadv.
- documentation
 - all requirements need to be available in advance
 - It can't start with partial data

Dell b/w error and bug.

When we code, we get problems called error
When software running, & discovering bugs.
bug identification is a big business.

Spiral Model.



Risk Analysis.

Risk → feature or factor that poses threat to the successful completion of software.
change in software, cost/time, govt policies, requirements.

Spiral, → used in large projects.

Software of library → No risk analysis.

- * unpredictability of requirements.
- * may be in future, technology becomes better
- * spiral has all features → evolution

Rapid Action Development Model

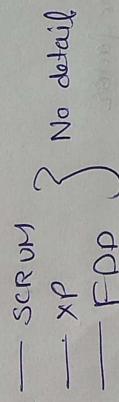
we implement or try to develop software in 60-90 days.

Major constraint is time,
(not in detail).

Agile Software development

- ↳ is completely different from all.
- ↳ less documentation is used or minimal
- ↳ customer interaction is on daily basis.
- ↳ small working cycles of 2-3 weeks.
- ↳ Documentation is not well designed

Techniques



Only viable / advisable for small scale software,
for bigger software, we need to use some sequencing.

Q. - Domain - Banking → (governmental)
library, (Waterfall)

which model we used?

There is no one ~~one~~ answer,

Always justify your answer.

Req

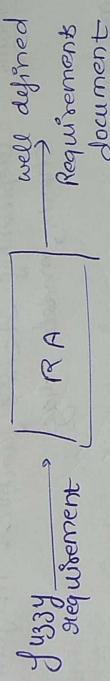
- Q) Find two problem domain where app implementation of special model will be appropriate, ? and why?
 # Defence, Aeronautics.

Class-5
16/12/2013

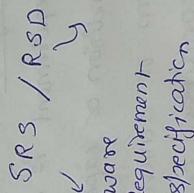
Requirements Analysis.

What is requirement?

- ↳ Need of the customers/stakeholders,
 - any third party,



(informal)



Output of RA

- 1) results in specification of software, operational characteristics such as functional data and behaviour, indicate software interface with other system elements and establish constraint that software must meet.
- 2) also provides software designer with a representation of information function and behaviour, that can be translated to data, architecture, interface and component level designs.

It is divided into five areas of efforts :-

- 1) Problem recognition
- 2) Evaluation & Synthesis.
- 3) Modeling
- 4) Specification (writing overall demands)
- 5) Review.

All these help in finding out what to be done?

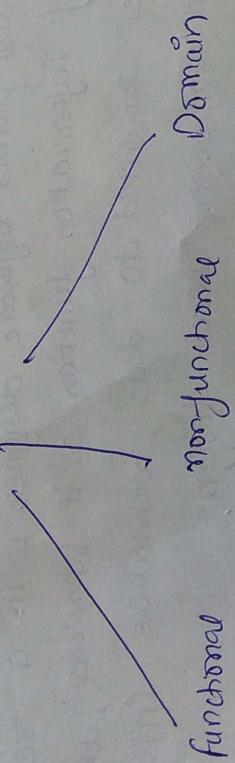
Modeling :- representing data & functions in form of
charts, models, diag.

Formal definition

A requirement is defined as :-

- 1) A condition or capability needed by user to solve a problem or achieve an objective.
- 2) It is 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 document.
- 3) A documented representation of a condition or capability as described in ① and ②.

Requirement



functional → Some for executes & manipulation in data. Ability to pay fees?

- ↳ Efficiency
 - ↳ reliable
 - ↳ secure
 - time constraint
 - system shouldn't crash.
- Non-functional
- ↳ we can not completely set them
 - ↳ we can not completely measure them.
 - ↳ every system needs those requirements beside the core functional requirement.

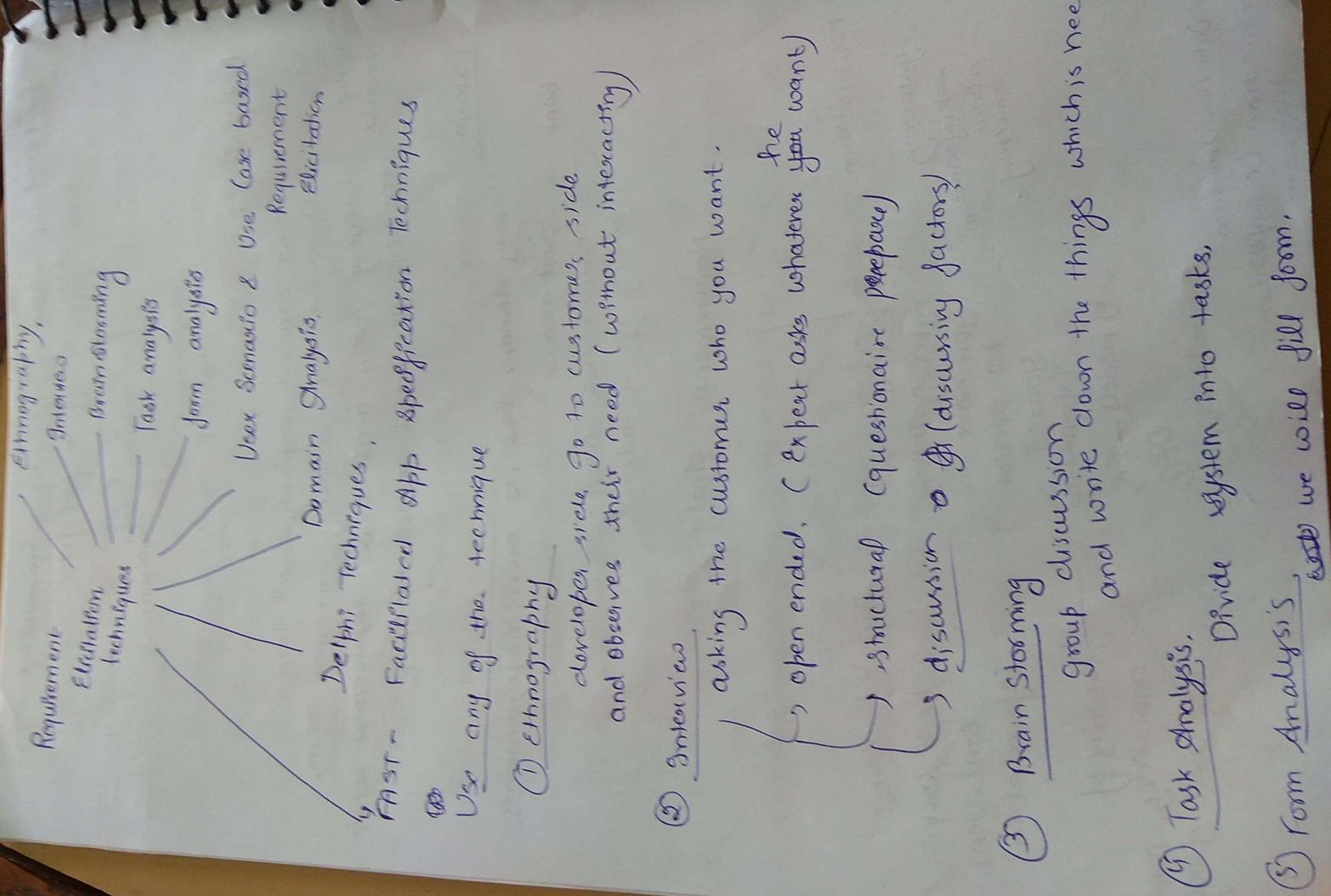
Domain → Every domain has its own requirements.

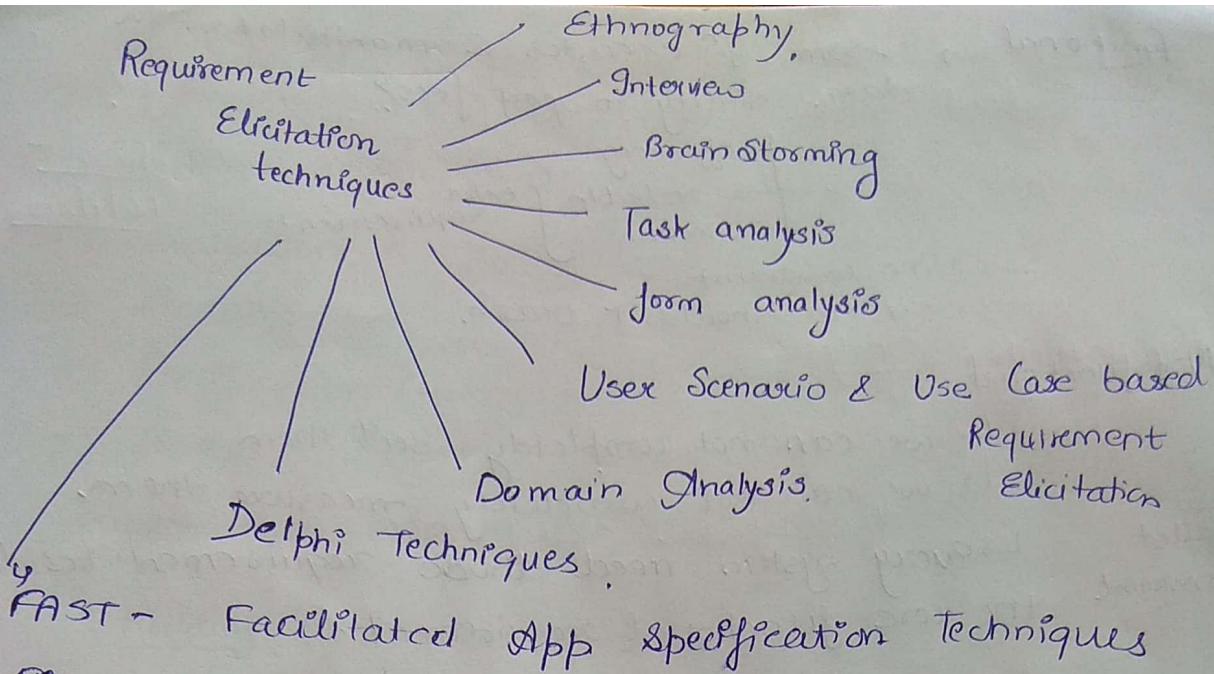
- Bank → create account
- keep his/her details
 - faculty of withdraw.

Requirements Analysis

- Req. elicitation → gathering information about something.
- Req. Specification → after gathering information you have to present your information.
- Req. Validation → check / review consistency, manual (no check of only)
- ↳ checking of consistency, manual (no check of only)
- by models → EER, DFD, FDD

DFD → Data flow Diagram
FDD → Function Decomposition
EER → Entity Relationship





Use any of the technique

① Ethnography

developer side go to customer side
and observes their need (without interacting)

② Interview

asking the customer who you want.

- open ended. (Expert asks whatever ~~he~~ you want)
- structural (questionnaire prepared)
- discussion (discussing factors)

③ Brain Storming

group discussion
and write down the things which is needed

④ Task analysis,

Divide system into tasks,

⑤ Form Analysis

~~and~~ we will fill form.

All forms which are part of any system can form the base of form analysis, base

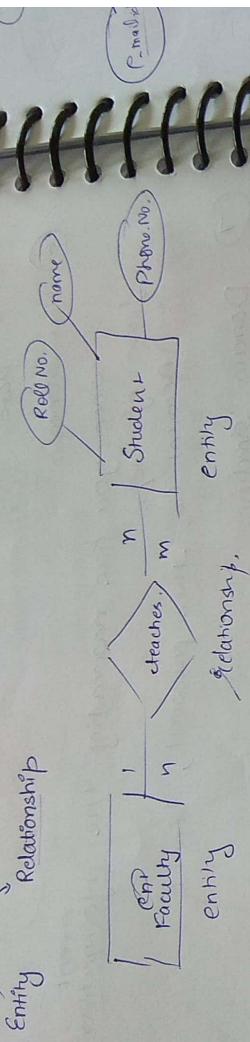
⑥ User scenario
when you are navigating the information in form of steps
→ Account No., then password, limit - - .

⑦ Domain analysis
Old legacy system, or requirement legacy.

find out

(Delphi & FAST) ?
/ (find out yourself).
In this,
manual export,
discuss
needs,
requirements,

ER Diagram



Attributes of Student are Roll no., name, phone no.

Cardinality → no. of participating entities

Primary key.

Tuples can be uniquely identified,
Atomic, complex attributes.

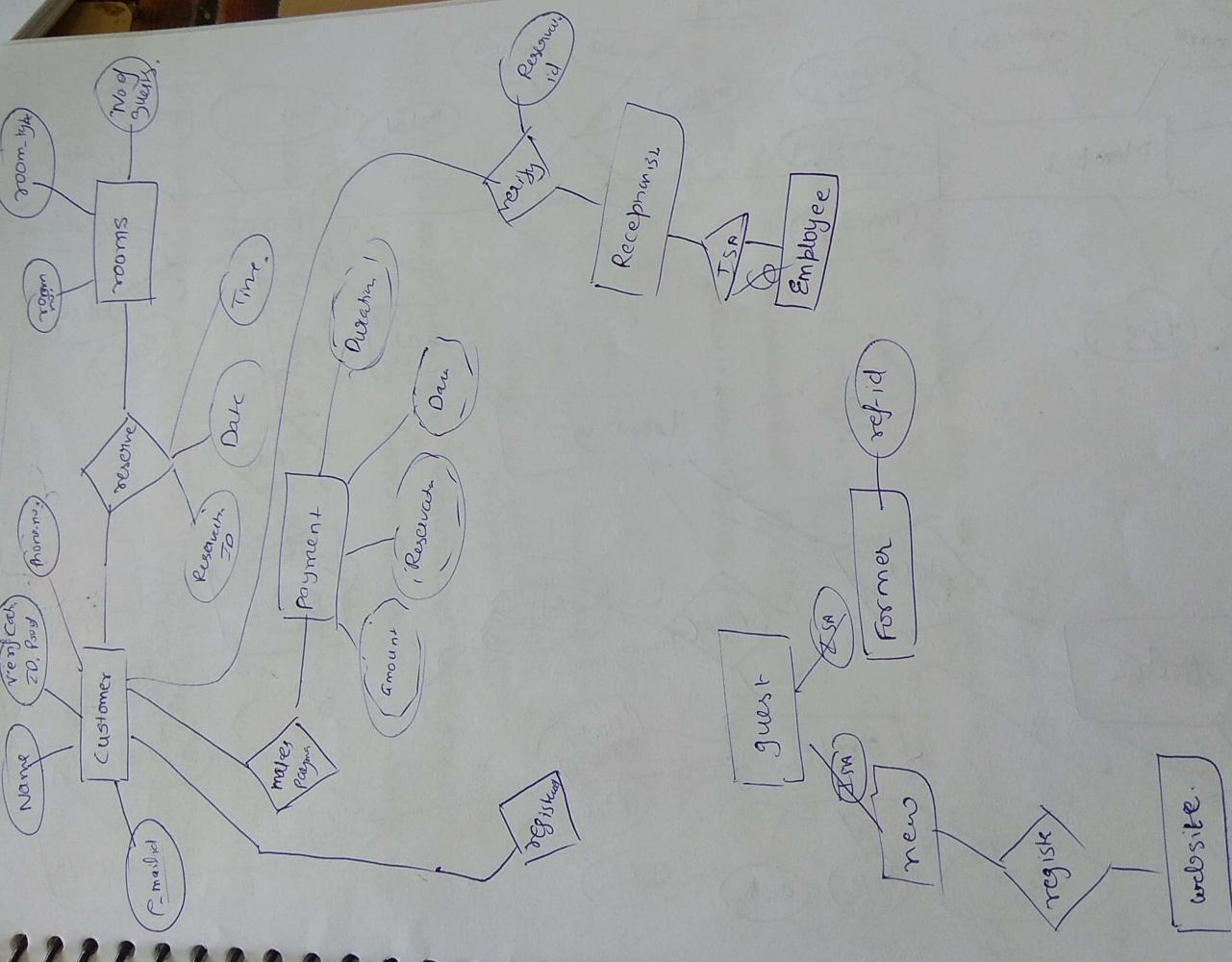
Normalization

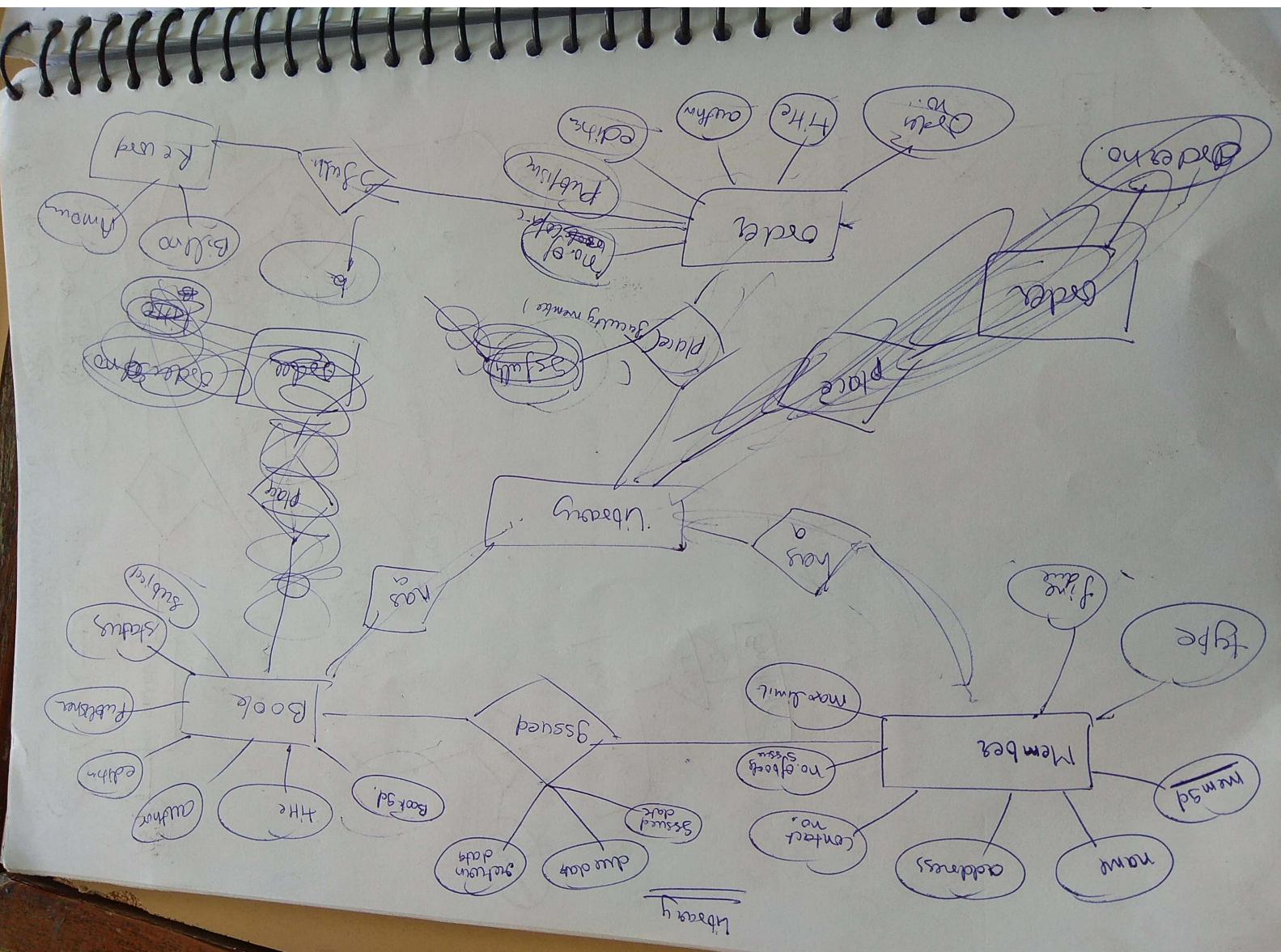
Relationship \Rightarrow unary, binary, ternary.

Draw ER diagram

Minimum four concepts to be used.
Submit assignment both,

Concept of generalization and specialization





23/11/18
class-8,9

→ Certain challenges faced during
Requirement Elicitation techniques,

- (1) changing requirements,
one meeting - something
another meeting - other
 - (2) One stakeholder may not express properly. Transfer of wrong info.
 - (3) Interpretation at receiver side may be wrong.
- (i) Understanding ^{scale} large and complex system requirement
is different.
- (i) Undefined system boundary.
 - (ii) Customer and stakeholders are not clear about their
needs.
 - (iii) ^{multiple stakeholders} conflicting requirements are there.
 - (iv) changing requirement is also there.
 - (v) partitioning the system suitably to reduce
complexity.
 - (vi) validating and tracing requirements, forward as well as
backward traceability
 - (vii) identifying critical requirements,
 - (viii) identifying part to be determined
solving to be solved
 - (ix) requirements
 - (x) proper documentation and meeting time and
budget constraint,

10

Organization of SRS.

IEEE 830
format

1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms, Abbreviations
- 1.4 References. → (Legacy systems. to understand current domain)
(it should be referenced.)
- 1.5 Overview

2. Overall Description

- 2.1 Product Perspective
- 2.2 Product functions
- 2.3 User Characters
- 2.4 Constraints.
- 2.5 Assumptions & Dependencies.
- 2.6 Apportioning Requirements.

3. Specific Requirements.

- collaborate some from third party
- 3.1 External Interface
 - 3.2 Functional Requirements.
 - 3.3 Performance Requirements.
 - 3.4 Design Constraints.

SRS

Software Requirement Specification Documentation.

There are also other formats also.

IEEE/EIA
12207, 1-1997,
Clause 6.2.2,

Exercise

You have to find atleast one format of writing SRS.

Industry Standards

ISO

is a company
follows a certain
set of activities and
basic standards to
provide something,
involves a lot of
documentations.

CMM

Capability

Maturity
Model

Standard for
software industry

↳ gives a set of
activities a company,
industry, developer
has to follow, while
developing a software

Ex- Infosys,

CMM has levels -

- 1.) level 1 — Ad hoc
- 2.) level 2 — Repeatable
- 3.) level 3 — Defined
- 4.) level 4 — Managed
- 5.) level 5 — Optimizing

KPA

↳ key process Area

If a company follows all the levels of CMM,
then it has given level 5 certification.

KPA for Level 1 → no KPA is defined

- Level 2 →
- ① Project Planning
 - ② Requirement Management
 - ③ Software Project Tracking and Oversight.

- (4) Sub Contract Management
- (5) Software Quality Assurance
- (6) Configuration Management.

Level 3 → ① Organization Process Focus

- ② Training Program
- ③ Organization ^{process} Definition
- ④ Integrated Software Management.
- ⑤ Software Product Engineering.
- ⑥ Peer reviews.

Level 4 → ① Qualitative process management

- ② Software Quality management

Level 5 → ① Defect Prevention

- ② Technology Change Management.
- ③ Process Change Management

Q, Is a software company eligible for ISO or CMM to both?

Which

Any Indian company which has both?

HCL is both.

Characteristics of SRS

24/11/18
class-10

- ① Correctness
- ② Completeness
- ③ ~~less~~ consistency
- ④ Unambiguousness
- ⑤ Ranking for importance
- ⑥ Modifiability
- ⑦ Verifiability
- ⑧ Traceability
- ⑨ Design independent
- ⑩ Understandable by customer

- ① What you are writing should be correct meaning, correct figures, correct result to the output.
- ② whatever you are writing should be left incomplete, make complete sense,
- ③ There should be consistency in different views,
- ④ clear cut single meaning.
- ⑤ we have to prioritize the requirements, those requirements which has to be done must, must give more priority, → make ranking of requirements.
- ⑥ Minor change is required,
SRS needs change

⑦ Testing part / manual review of document

⑧ You should be able to start do a point and end to a point,

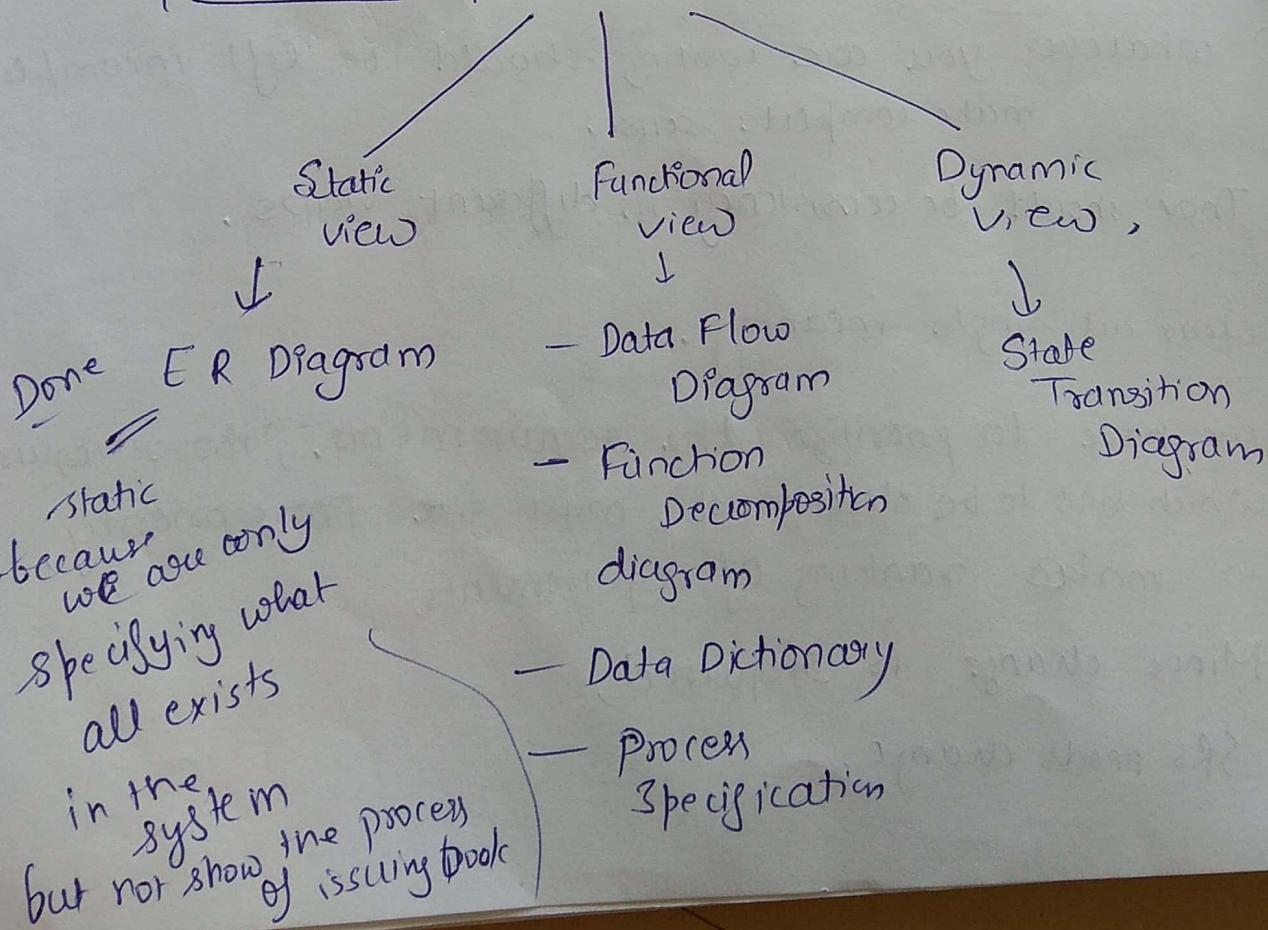
→ there is a link.
Write function (cooperating & transmitting data to traceability)

⑨ Whatever we are writing should not be language dependant.

We will be writing requirements, on the basis of requirements only, not language base.

⑩ Customer is not even in picture,
if doc is used as contract, then it should be understandable by customer.

Requirements Specification



functional view

we are describing the actual functioning of the system,

① Dataflow

flow of one function to another function is shown through dataflow.

For ex- issuing a book ~~can be~~ can be shown through this,

It is a time consuming diagram.

② function decomposition

→ able to show all the functions.

→ we can hierarchy of show all the modules of system.

We are decomposing the bigger function into small activities or function.

③ Data Dictionary

Meta Data → Data about Data.

Write in natural language that student is composed of name, Rollno, age etc.

④ Process specification

Describe all the process in natural language

Dynamic view

State diagram,

we ~~are~~ identified set of activities happening, event happening and ~~are~~ identifying the change happen to a state after performing an event.

Data Flow Diagram.

Class 11 & 12
30/11/18

In E.R. diagram, we can not express the process.

level 0 DFD
(context diagram)

level I DFD

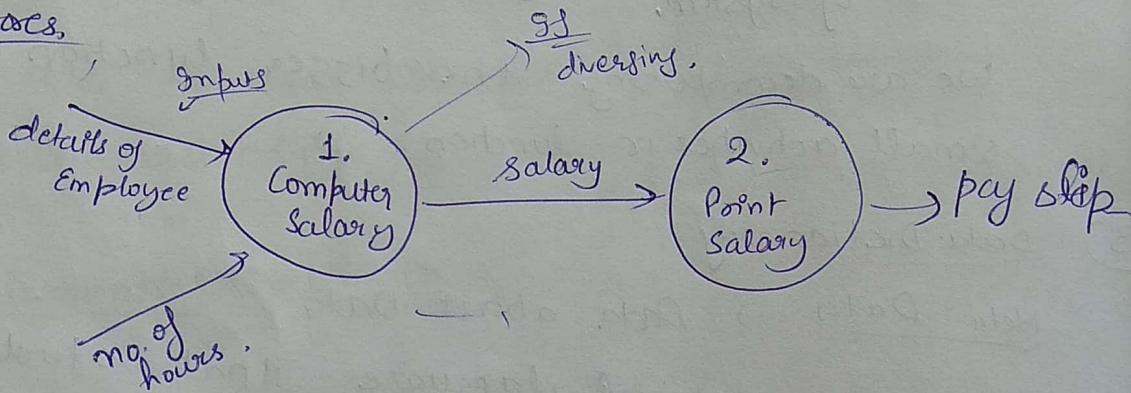
levels ~~of~~ representation, → there are different levels of every abstract.

→ Process.

→ External Agent/Terminator,

→ Data flow,

→ Stores,



Converging

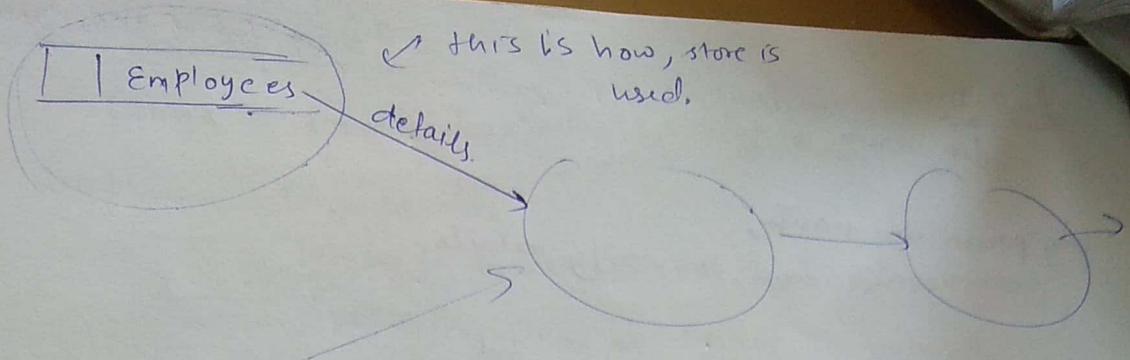
→ set of entities, will become the stores of Data Flow

Diagram

Stores

Employees

employees.

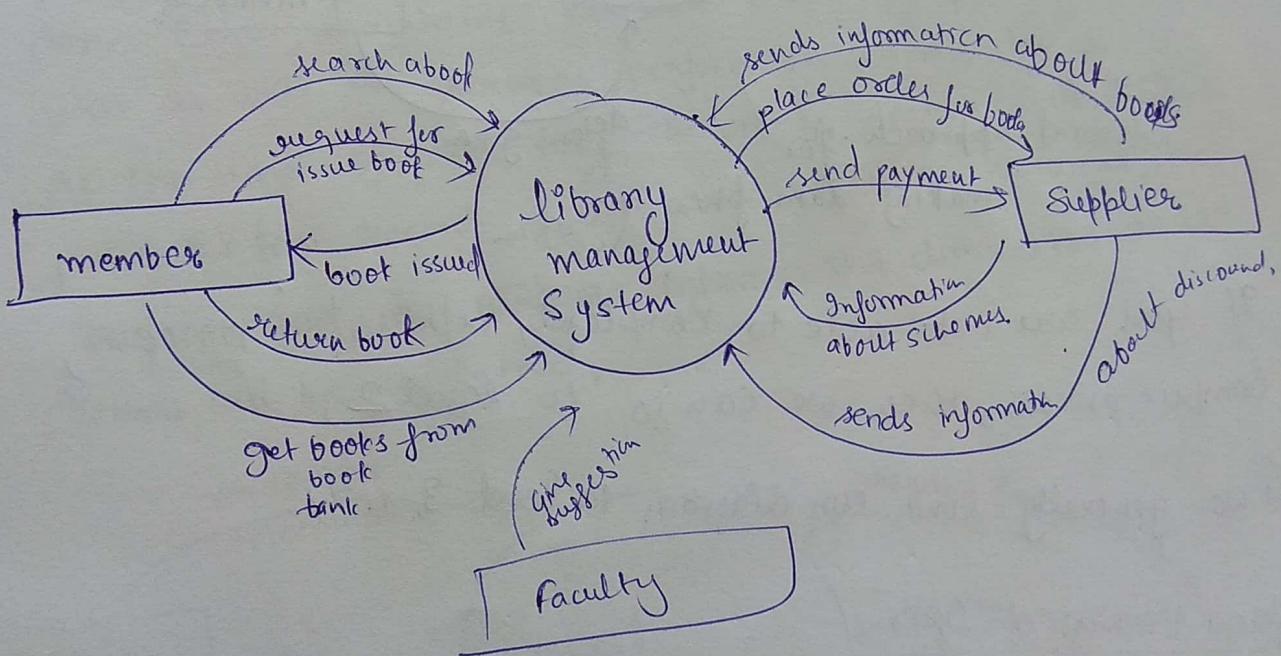


Ex Bank External Agent or Terminator.

Bank.

Context Diagram Level 0 DFD.

When we display only one process, and all external agents, then Context Diagram.

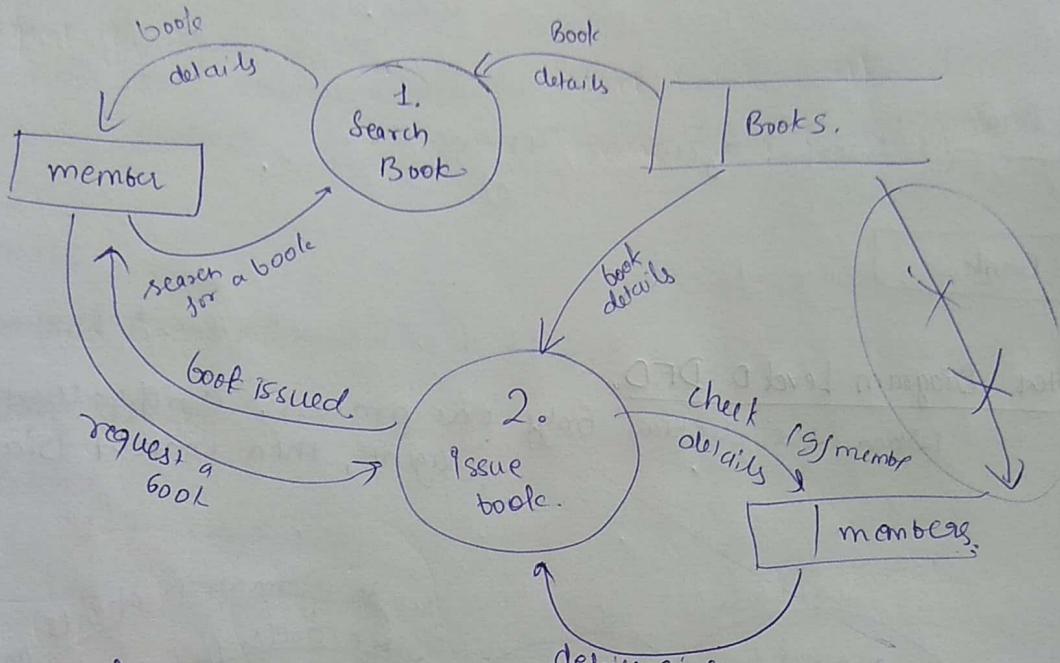


Advantages → please draw a boundary for the whole software

Level -1

for search book

Any person can search,
No need to verify details.



Classical approach for
making data flow diagrams

If you are not able to completely express a process,
Complex process, then we can go to level 2,
→ We generally limit our diagram to level 3, or 4

Different Version of DFD
Event participation approach for DFD

What will be an event?

Change in the data happening.

Event — Flow Oriented

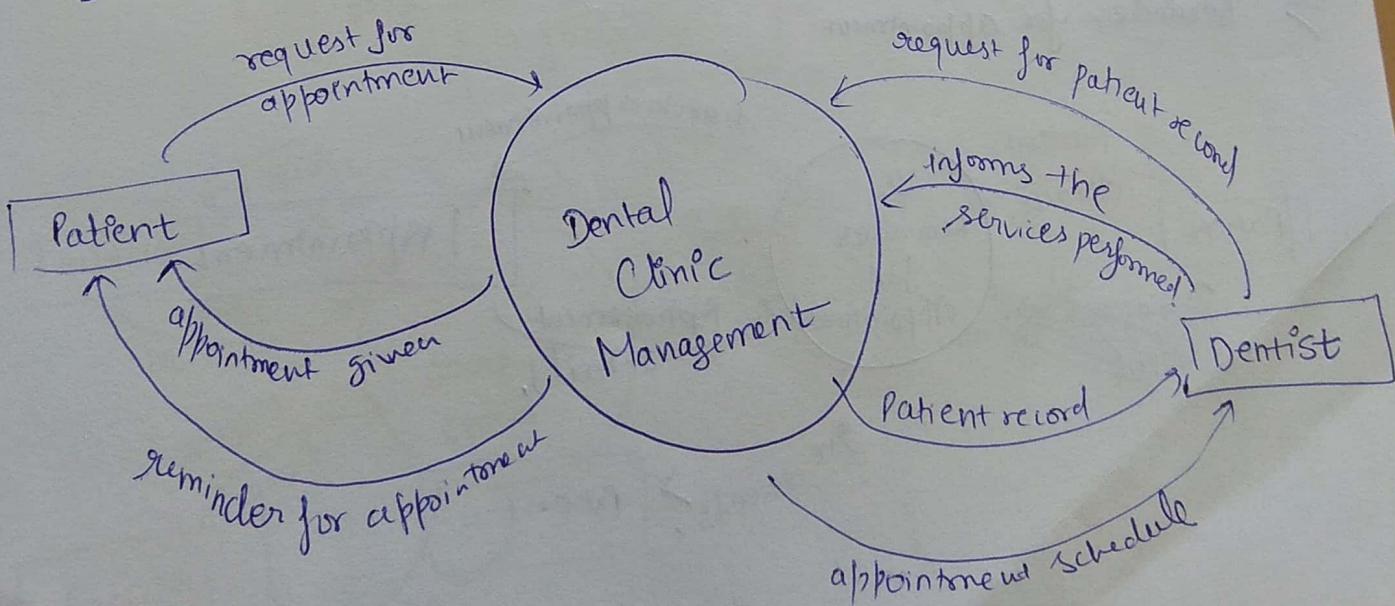
— Temporal (associated with time). Generate report weekly.
Ex System. (condition, set system automatically do some work)

Clinical Management System.

Problem System

A Clinic is in the business of providing dental services. A number of doctors are on the roles of clinic. Patient can take appointment on phone or personally, for a particular and a particular service.

Clinic sends reminders to patient regarding the appointment. Patient performs the visit to get the services done. Clinic maintains complete record of patient's last visit as well as services performed on him by doctor. At the time of performing services, doctor asked for patient's last record, if any and whatever services he performs ~~and~~, he ~~performs~~ informs the clinic, so that records can be maintained.



Event List.

1) Request for appointment

Event name / no.

dataflow

Patient details (IN)

Appointment (OUT)

detail.

Reminder sent (OUT)

2) Reminder for appointment

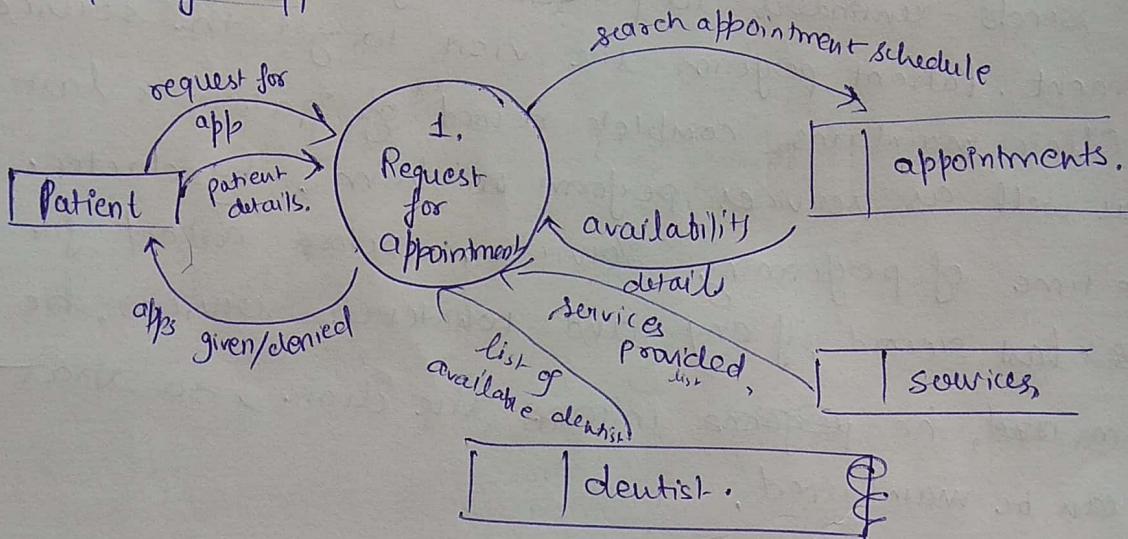
3) Request for Patient rewards.

Patient Details (IN)

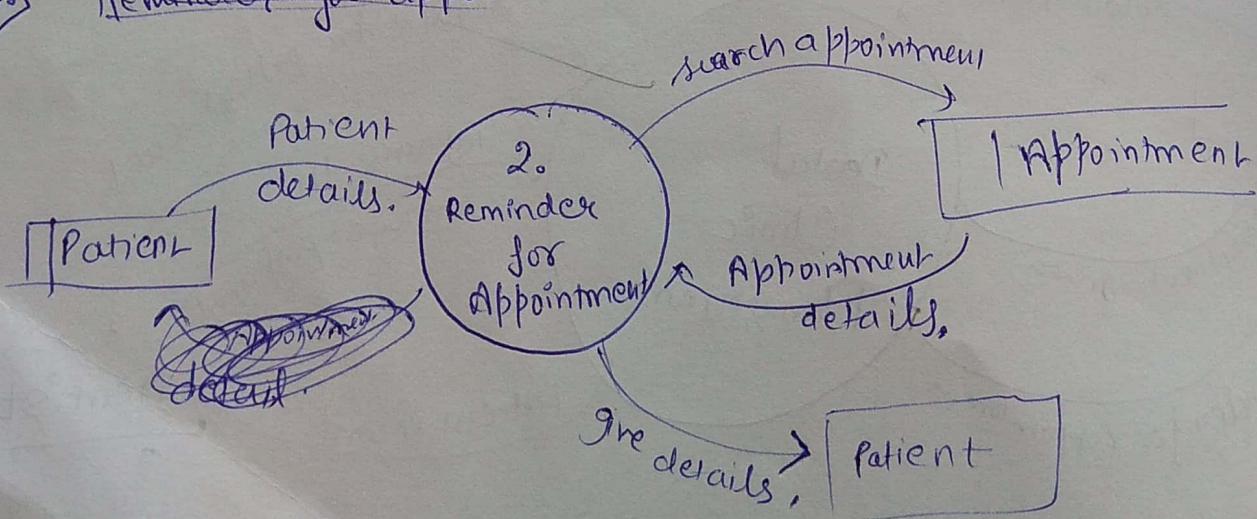
Previous record details (OUT)

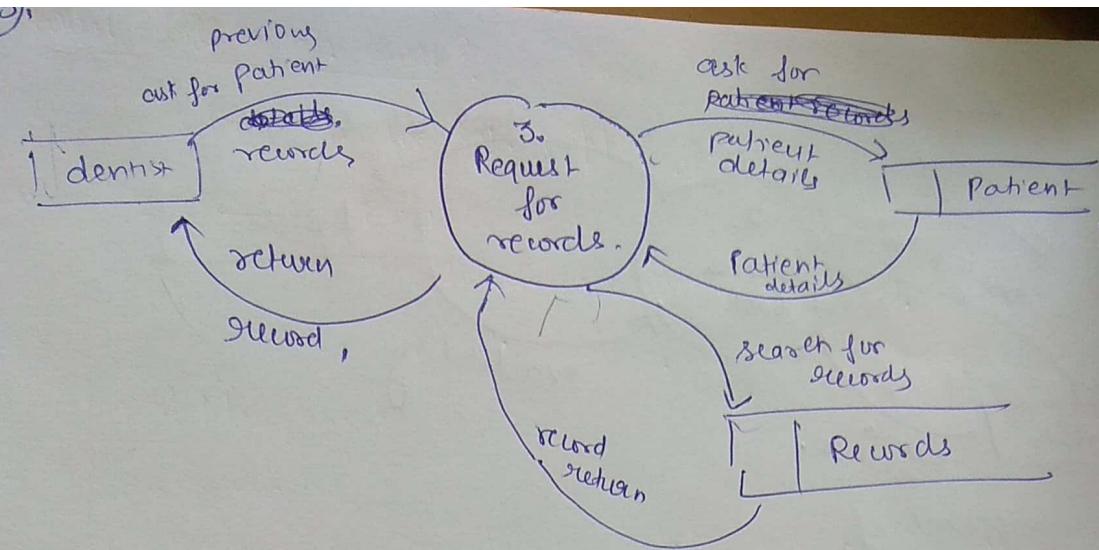
Data flow Diagram.

1.) Request for app.

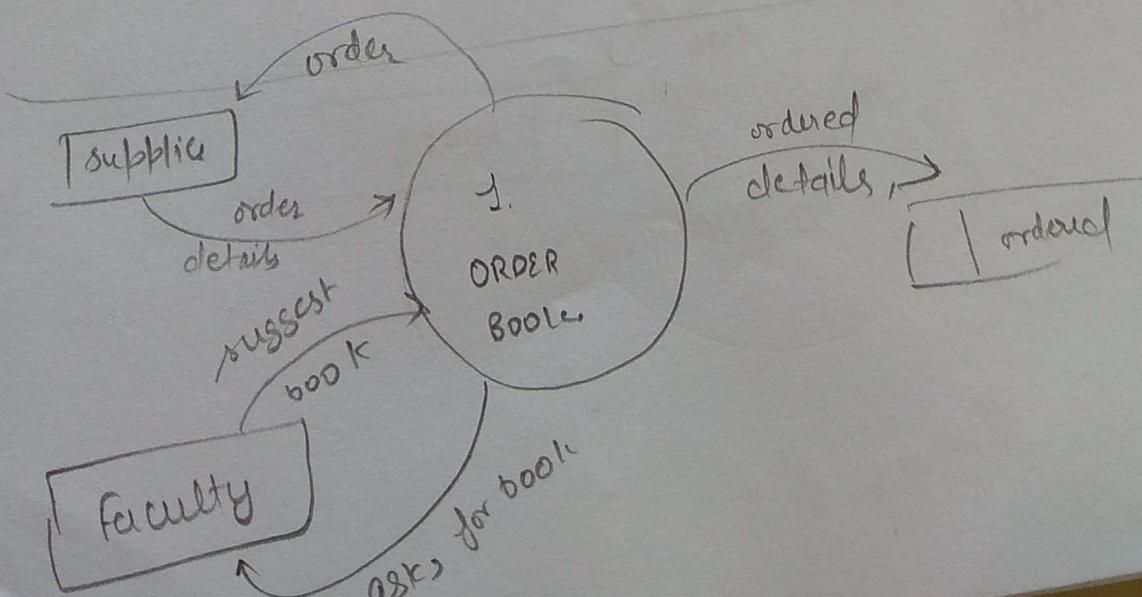
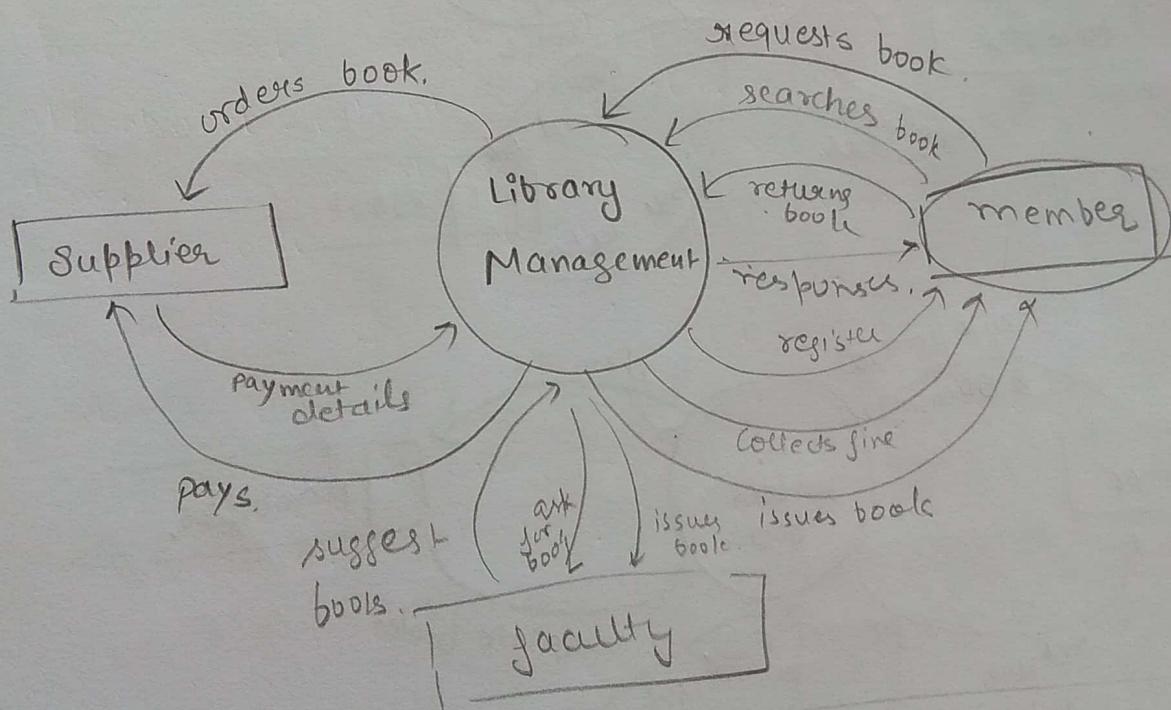


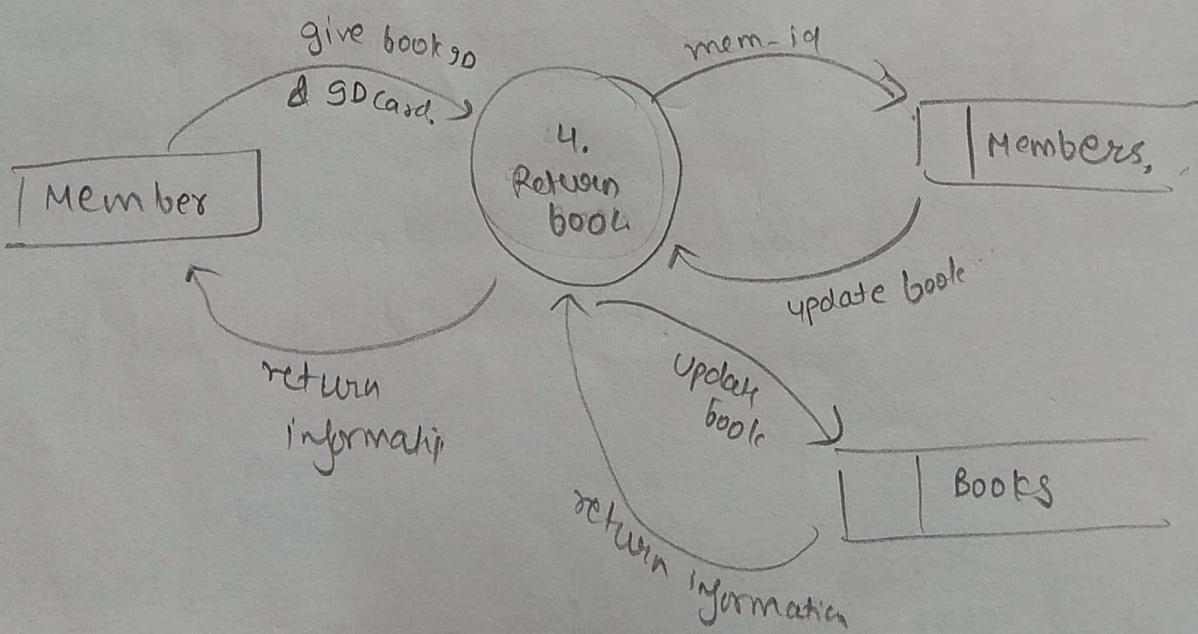
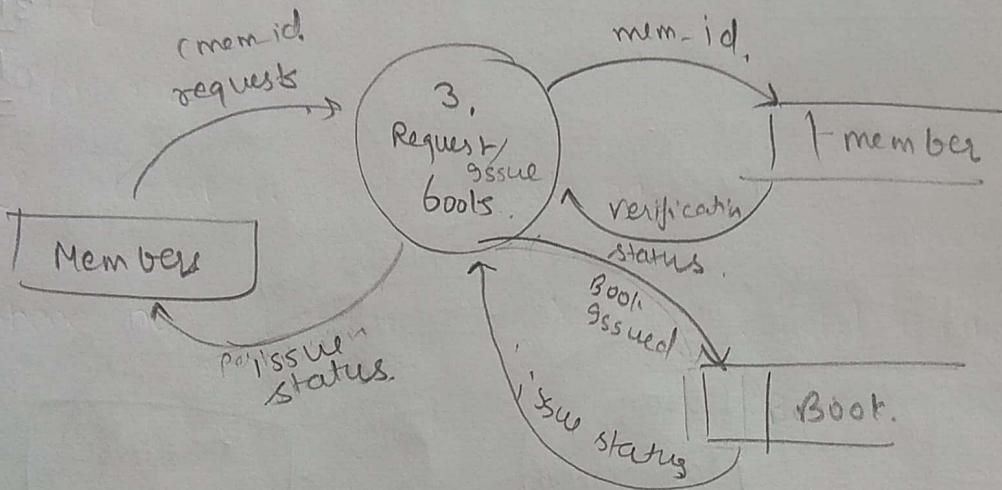
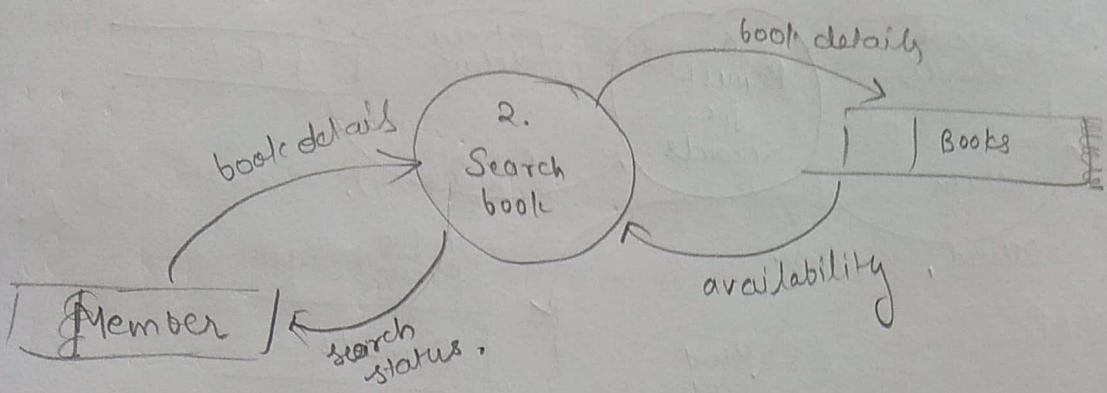
2) Reminder for appointment





31/1/18
class-13





Data Dictionary.

Halisa
class 14

function

Convention used

** , // comments.

[] One of several choices,

@ key field of a store

/ choice

= is composed of

+ and

() optional

n { } m atleast n and atmost m.

E-

telephone no = (country code) + (area code) + number

Country code = { digit } 3

digit = [0, 1, 2, ..., 9]

area code = { digit } 5

number = ~~customer id~~ { digit } 7 { digit } 8.

customers = { customer } m

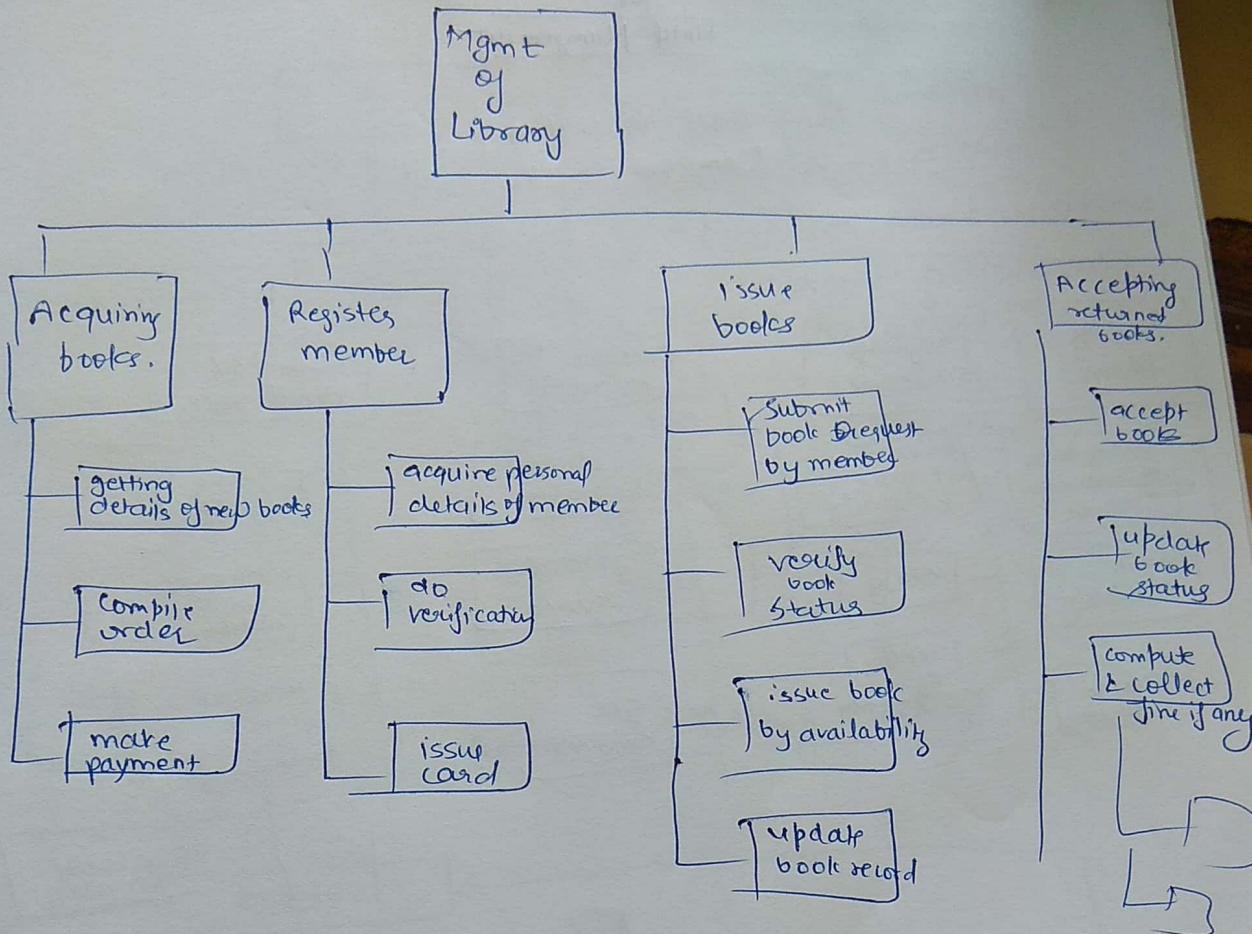
customer = @customer id + customer name + address
+ telephone no.

entity Customer is composed of customer id, and customer name
and address and telephone no.

Customer is a store that is composed of 1-m customers

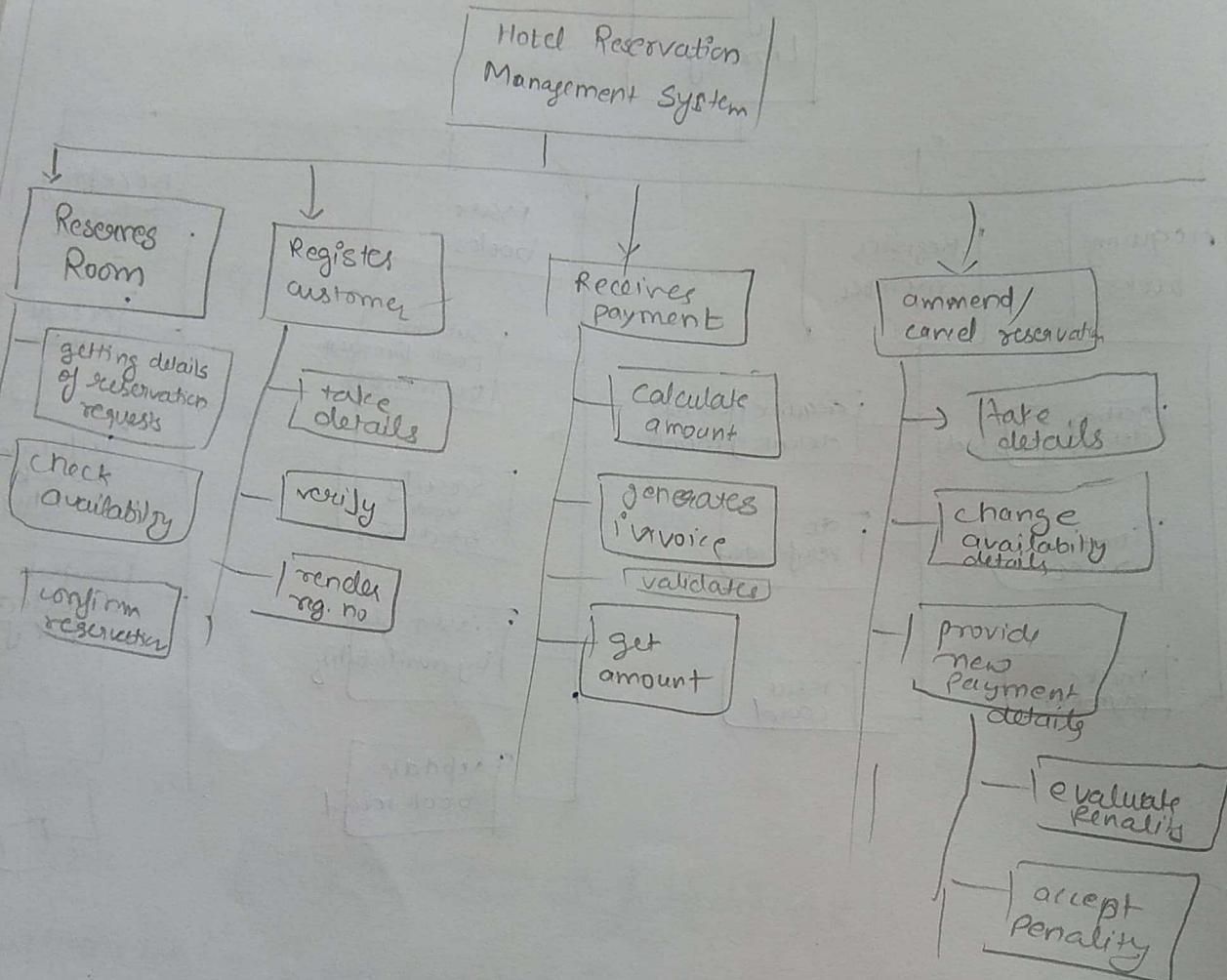
Data dictionary for address.

Function Decomposition Diagram



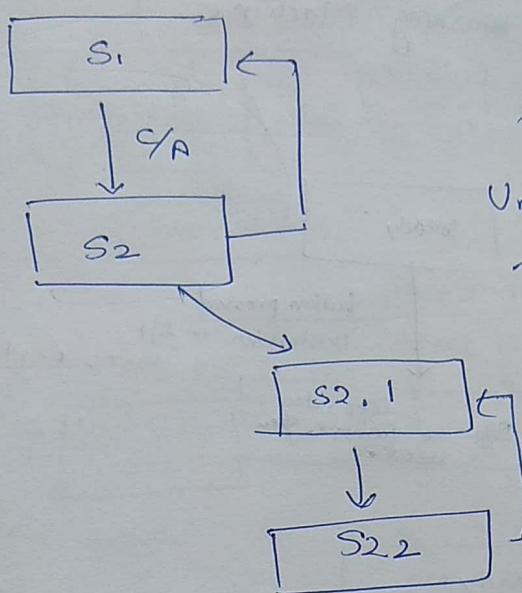
Function Decomposition Diagram

for
Hotel Management



State Transition Diagrams

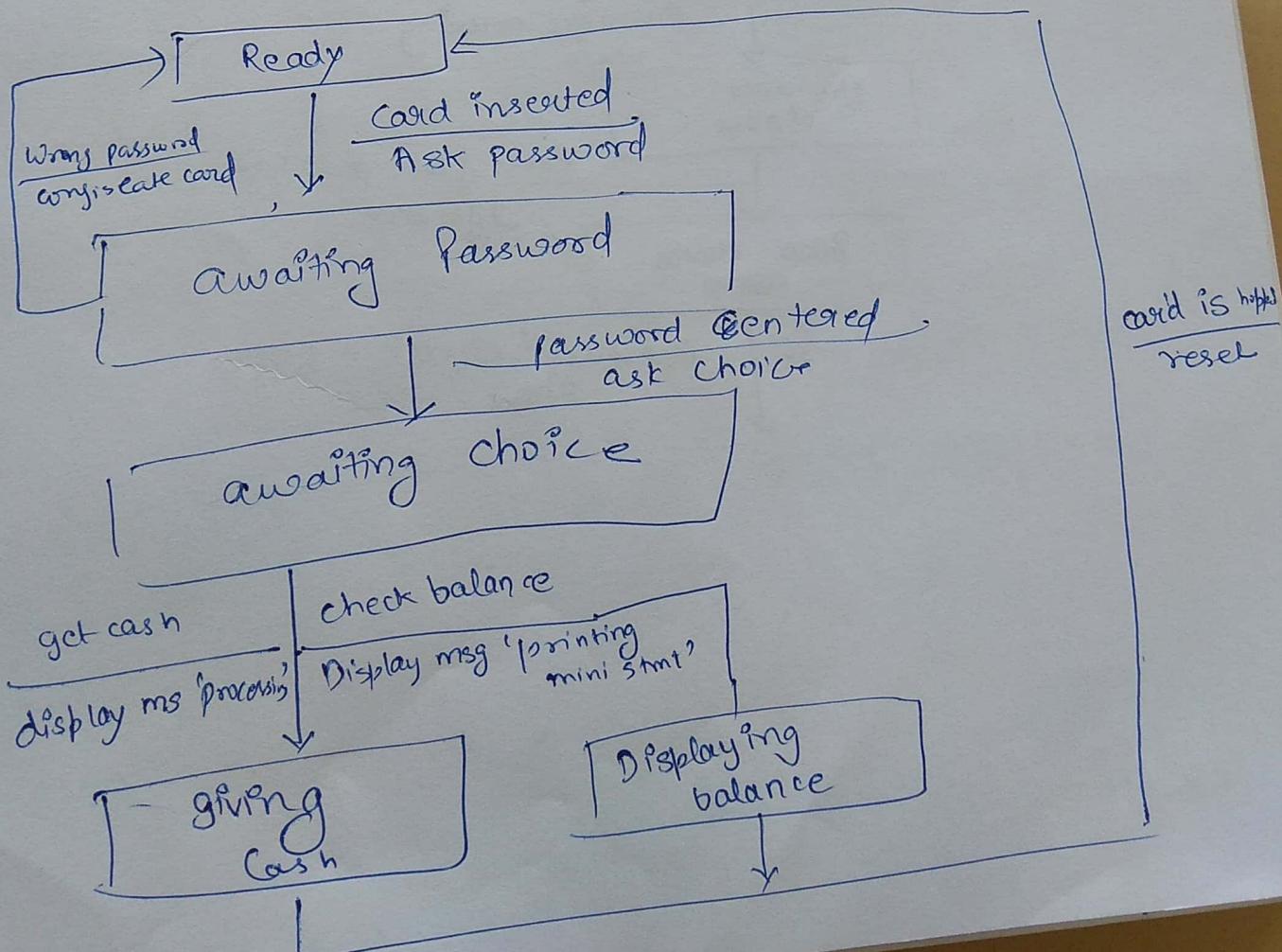
13/2/18
Class-15, 16



C → condition
A → action
Under some conditions,
some actions are
performed, such
that its state
changes.

Taking Example
ATM

Automatic Telling Machine

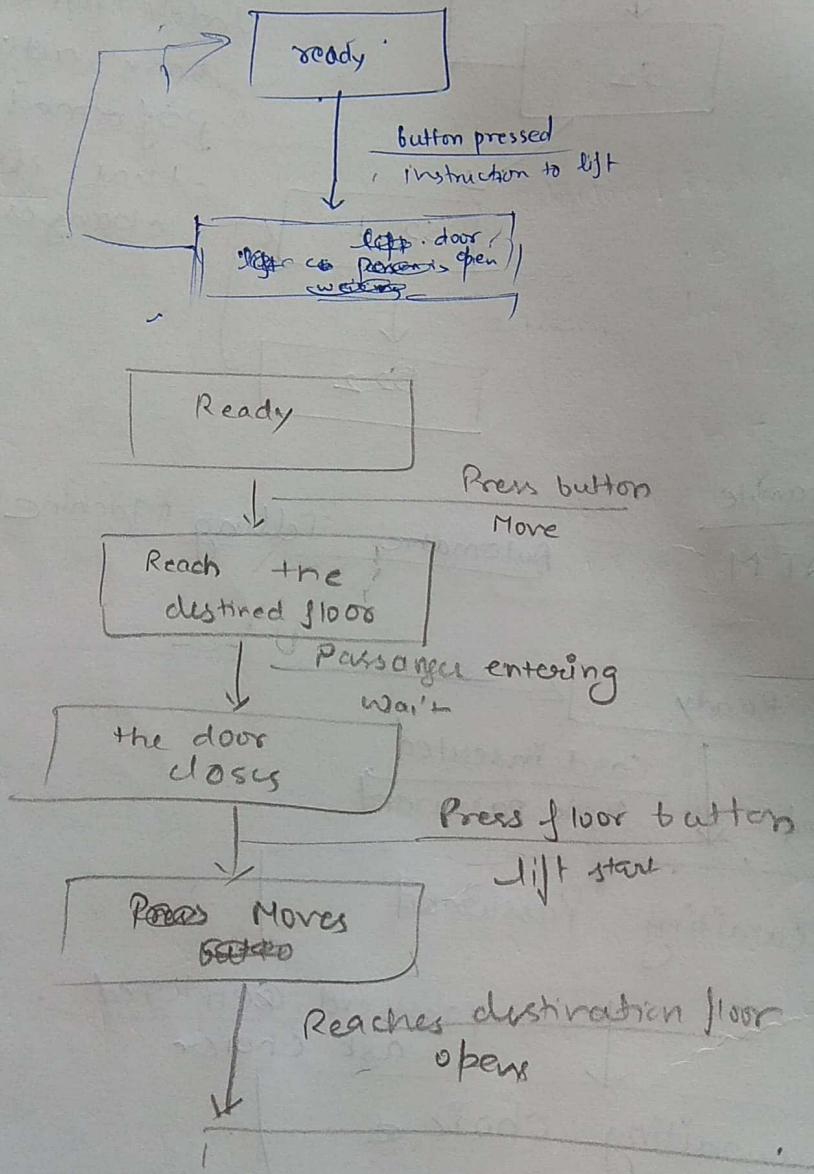


Make state transition diagram for

① Working of a Lift

② Working of Washing Machine

Microwave

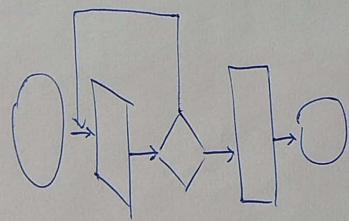


Microwave

Process Specifications. / Logic Modelling.

- Eliza Chat
 - Structured English
 - Pre and Post Condition
 - Nassi-Shneiderman Diagrams / Charts
 - Decision Table
 - Decision Tree

Flow chart



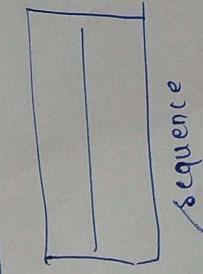
Stylised English

- C = a+b
- Using some predefined
variables, set.
- Set a to 5
- Set a to 6
- add a and b
- Store resultant in C
- Set, Store, Repeat until
do, while, if else.

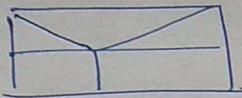
Pre and Post condition

before executing ~~as~~ a function , there are certain things
need to be fulfilled and also after executing the function.

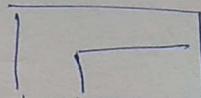
Nassi-Shneiderman Diagrams / Chart



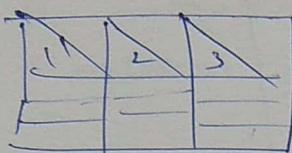
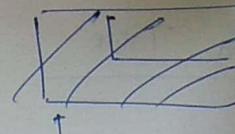
do until loop.



conditional statement

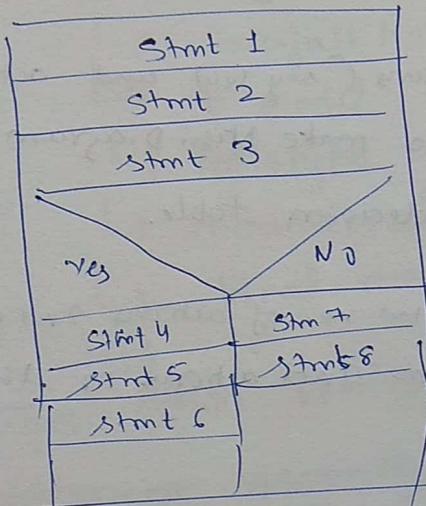


Do while
loop



case stmt.

Ex:-



set cash-sale to 0

set sale-status to 'cash'

Set total-sale to 0

Set credit-sale to n

Ready-Daily-sale, sale-status, EOF

$$\text{Total-sale} = \text{Total-sale} + \text{Daily-sale}$$

Sale-status = "cash"

Yes

$$\text{cash-sale} = \text{cash-sale} + \text{Daily-sale}$$

No

$$\text{Credit-sale} = \text{Credit-sale} + \text{Daily-sale}$$

Return Total-sale, cash-sale, credit-sale

Until EOF = Yes,

Decision Table

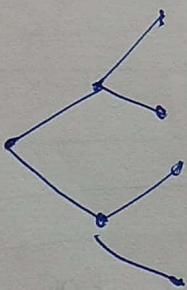
	Rules		
conditions	✓	✓	
Action			
Action			

where there are large conditions (say 100) and actions (50)
then it become difficult to make Nassi Diagram.

Therefore we can make decision table.

Decision Tree (when we have limited no. of actions and decisions)

when we represent conditions and actions in this format, then decision tree



One form
of
decision table

Example in pics in group

Syllabus

- Introductory concepts of
 - What is software?
 - characteristic of software?
- What is software Eng..
 - Definition & Purpose
- Software Development Life Cycle
- Process Models
 - ↳ waterfall Model
 - ↳ V-model
 - ↳ prototype model
 - ↳ incremental model
 - ↳ spiral model
- Rapid Action Development
- Component Assembly Model
- Capability Maturity Model
- Diff. b/w CMM & ISO.

Requirements Engineering & Requirement Analysis

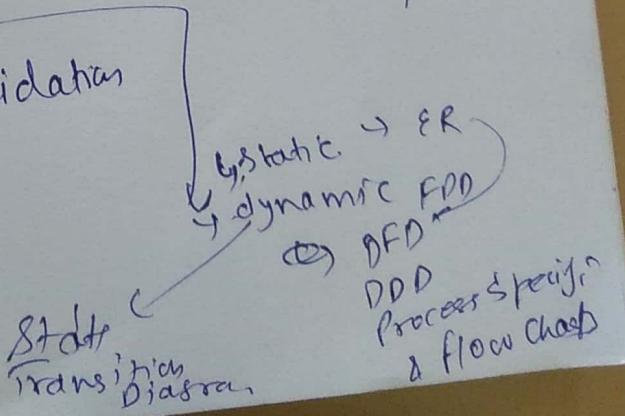
- ↳ Definition & Purpose

Requirement Engineering ~~as process~~ process (Definition & Purpose)
involving:

- ↳ Elicitation
- ↳ Specification
- ↳ requirement validation

Techniques, problem faced during Elicitation process.

ALSO



sets pass for 82.4 month ↗

It difficult to show how many should be given to you

if \overleftarrow{SRS} the form should be aware you