10-09-24

Define System Analysis & Design.

Its a Process of planning & developing high quality and new business information system which Combines It People & data to support business requirement

Differentiate between Prof's Hoffer's method & Prof. Kendall's method of (SDLC) with respective diagram.

SDLC is a conceptual model or systematic approa which includes polices and Procedure for develo System throughout their life cycle. Prof Hoffe & Kendall has different method for it. Her are,

For Prof Hoffer: (Waterfall Diagram)

Phases in SDLC

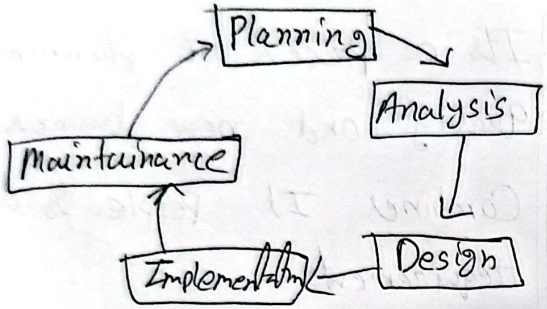
I) Planning

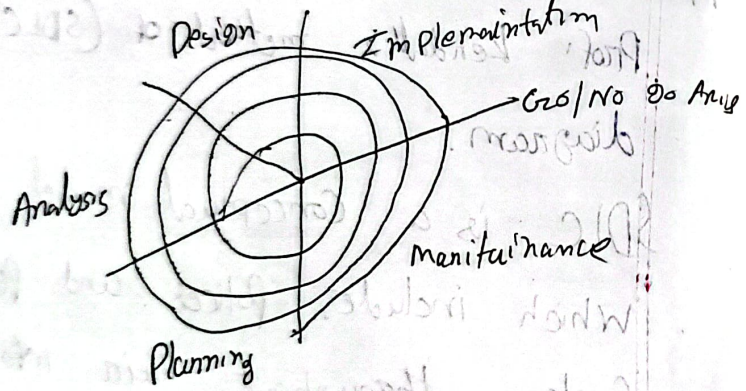
II) Analysis

III) Design

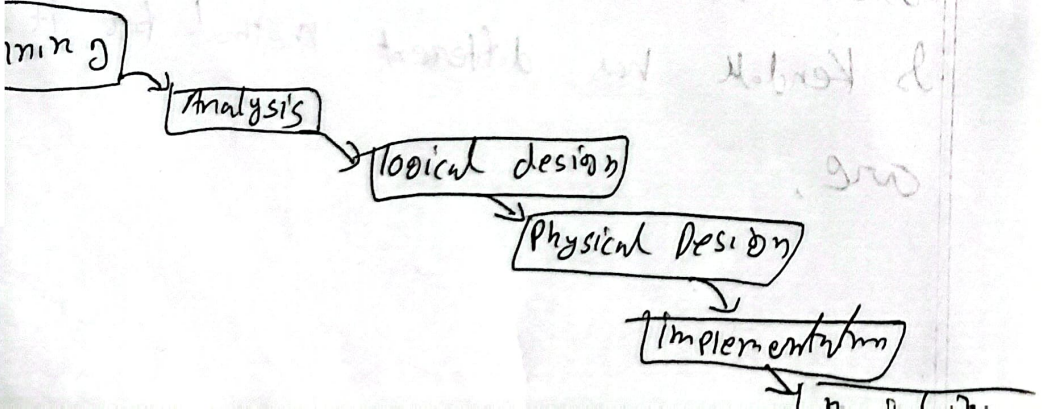
IV) Implementation

V) Maintainance.





terfall:



I) Planning: Define the Problem & the objectives of the system.

II) Analysis: its a detailed investigation on operation relationship among functional units of the system. Study of system requirements

III) Design: Create a blueprint of the system that m identified requirements.

Logical Design:

Physical Design:

IV) Implementation: The design is transform into Coding activities.

Maintainance: Systemicly repair and improve over the time

For Prof Kendall: (Iterative method)

Traditional methodology for maintaining and replacing information systems is emphasize iteration & prototyping with the user with user feedback loops.

Phases in SDLC:

① IDentyfing the Problem; in first phase of SDLC is analyst is concerned with correctly identifying the problem, opportunity and objective. This stage is critical to the success of the rest of the project

(i) Identifying Problem
(ii) Determine requirements
(iii) System Analysis
(iv) Design
(v) Coding
(vi) Testing
(vii) Implementation
→ Parallel

II) Determining the requirements: in this Phase analyst determines what information users need to perform their Job. People

The analyst need to know.:

i) who: one involved
ii) what: business activity
iii) where
iv) when (timing)
v) how
vi) why, the busine current systems

People involved in this Phase:

Analysts & users typically operation manager

III) Systems Analysis: Analysts prepares a Proposal that summarizes what has bee about the users, usablity and usefulnes Systems. Provide cost - benifit analyses and makes recommendations on what shou

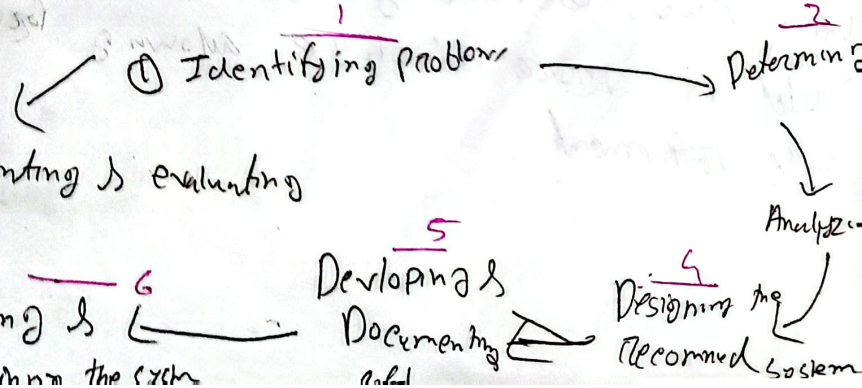
done

iv) Designing Systems: The system analyst uses information collected earlier to accomplish the design of the information system. Design Phase is for users to help them accurately enter data so that data going to info system are correct

v) Developing & documenting Software: Analysts works with programmer to develop any original software that is needed. In this phase analyst works with users to develop effective software.

vi) Testing and Maintaining the System: before the Software is publish it must be tested. it's less costier than catch a problem before it's publish, Maintainance and documentation begins in this Phase and it's carried out throughout the life of IS

vii) Implementing & Evaluating the system: Last ph of the system.. it involves towing user. to han System.. & Analyst needs to plan for smoo Conversation from the old system to the new



Key Differences:

1) Hoffers method: follows a step by step waterfall approach, while Kendall's method emphasize iterations and prototyping with user feedback loops.

2) Hoffer's Process assumes requirements are fully understood at the beginning, whereas Kendall approach allows for flexibility and adjustments throughout the development process.

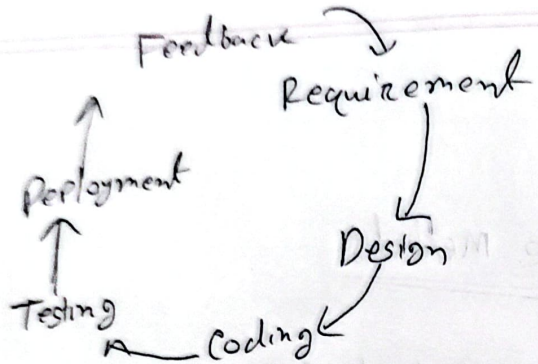
The waterfall method is better suited for project with clear, unchanging requirement while iterative model is more adaptable allowing for ongoing and refinement.

Agile method:

This method focuses's on delivering small & iterative increme releases of a project/product in collaboration with stakeholders, allowing for frequent feedba and adjustment. Agile emphasizes adaptibility Customer collaboration and continuous improvem rather than following a rigid sequential Process like the traditional waterfall mode

Phases Agile methods:

I) Requirement
II) Design
III) Coding
IV) Testing
V) Deployment
VI) Feedback



i) Requirement: Collaboration takes place with stakeholders to understand the Project

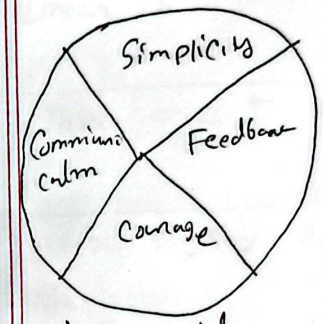
ii) Design: translated into actionable task breaking them into smaller chunks. In this step developer develop visual representation of solution

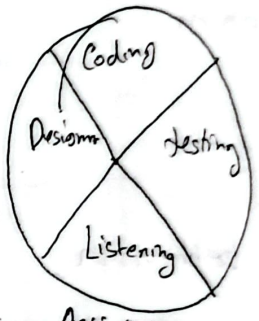
) Coding: iterative cycle known as ~~Chunks~~ Sprints. focuses on delivering small functional increments of the Project.

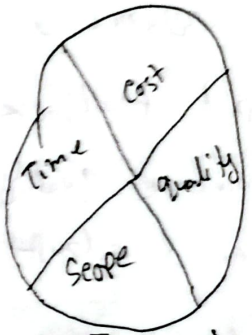
testing: part of iteration. ensure quality and functionality of software.

5) Deployment: involves releasing the software.

6) Feedback: Crucial for continuous improvement in Agile.







14 Principle of Agile model:

I) Satisfy the customer

II) Embrace change:

III) Continue to developing functional software incrementally

IV) Encourage customers & analyst work together

V) Trust motivated individual.

VI) Promote face to face conversation

VII) Concentrate on getting software to work

VIII) Continuous, regular & sustainable development

IX) Adopt Agility

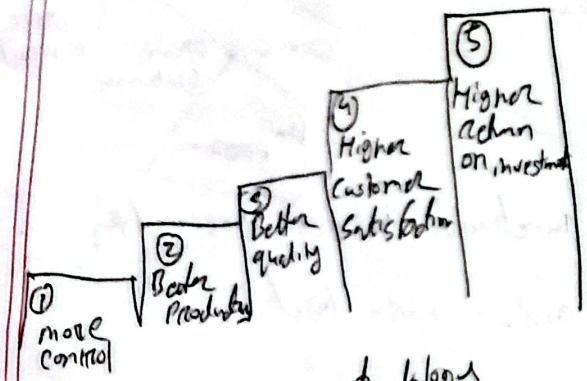
X) Self organizing team

XI) Rapid feedback

XII) Encourage quality.

XIII) Review & adjust behaviour.

XIV) Adopt Simplicity.



Agile methodologys.

Another

	Prof Hoffer	Proff Kendall
I)	Prof Hoff method is based on waterfall method	I) Kendall method is based iterative or incremental mode
II)	Linear & sequential approach	II) This approach is iterati & flexible
III)	There are 5 phases	III) There are 7 phase in method
IV)	Hoffer's method is less adaptable	IV) embraces change & refinement
V)	involves feedback mainly at the end of the development cycle.	V) This method incorporate feedback throughout the
VI)	carries higher risk	VI) mitigates risks
VII)	relies on detailed documentation before moving forward	VII) adaptive documents evolves with the project

Lecture : 2

Project : A Project is an interrelated set of of activi
that has a starting and a ending Point and result
in a unique product

Project management : its a scientific way of Plan
implementing monitoring & controlling various aspects
of Project like time, money, cost & manpower & oth
resources.

Opert : Programme Evaluation & Review Tech

its a management tool to schedule, organize &
coordinate tasks within Project

i) Determine tasks, ~~to, to, th~~
ii) network Diagram
iii) Critical Path
iv) Estimated time & variance for each tasks
v) ~~Standar~~ Overall Project vari
vi) SD of Project variance
vii) Find Z score

Post Analysis :

$$t_e = \frac{t_o + 4t_m + t_p}{6}$$

t_o = optimistic time
t_m = most likely time
t_p = Passimistic time.
t_e = Expected time

$$\text{Variance, } \sigma^2 = \left[\frac{t_p - t_o}{6}\right]^2$$

Variance is the difference between optimistic & Passimistic time from the mean time

Critical path: its a path conneting first initial node to the very last terminal node. (Longest duration)
(সব থেকে বড় যেই connection Path টাই নিবে।)

Overall Project এর variance বের করতে হলে Critical Path এ যেই node গুলো আছে তা যোগ করতে হবে

3) Standard deviation of variance is:

$$SD = \sqrt{\text{Project variance}}$$

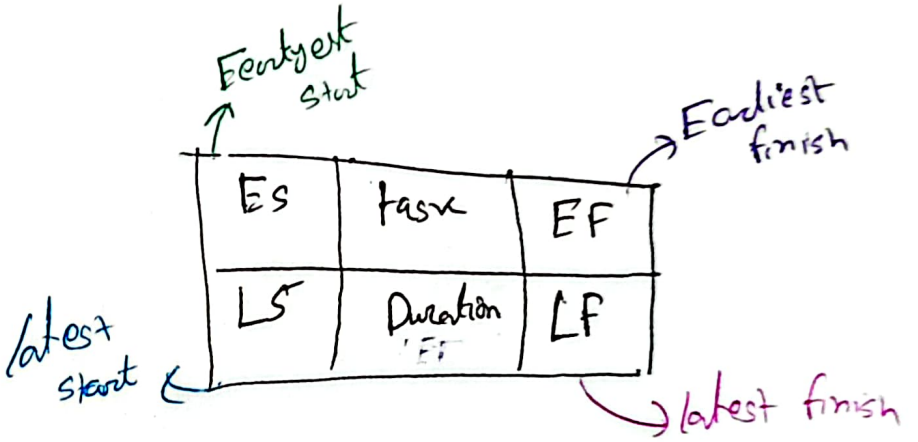
4) For Z- value:

$$Z = \left(\text{Due Date} - \text{Expected time of Completion}\right) \div$$

Alternative method

Slack: Critical Path is the longest path in the network. Each node falls under critical path has zero or negative float.

Slack is the amount of time an activity can be de



For the first node slack EF == LF

For 1st node Earliest start (ES) always 0

1) EF = ES + Duration — Forward Pass

LS = LF − Duration — Backward Pass

Slack

Late finish − Early

Late start − Early S

2) For next node ES is Previous nodes EF

For Forward Pass: (For ES & EF)

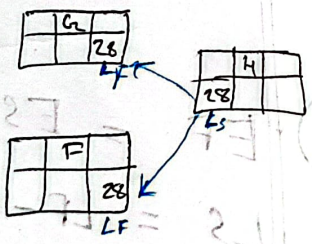
যদি multiple Previous node থাকে তাহলে multiple Earliest finish (EF) হবে যায় তখন বড় EF টা নিবে next node এর ES (earliest start)

For Backward Pass:

1) First node and the last node হবে EF = = LF

2) late start হবে late finish এ

3) LS = LF - Duration



4) যদি multiple Prev node থাকে then for backward minimum Laste Start এর next node এর LF এ নিবে

Slack যেটা Critical Path বের করে.

যেগুলোর Slack = 0 সেগুলো Critical Path এর member

~~LS - EF~~ = = 0

LS - ES = = 0
LF - FF = = 0

} member of critical Path

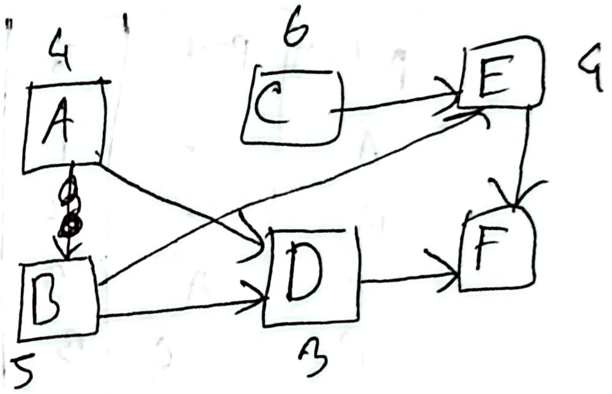
(See A CT 1)

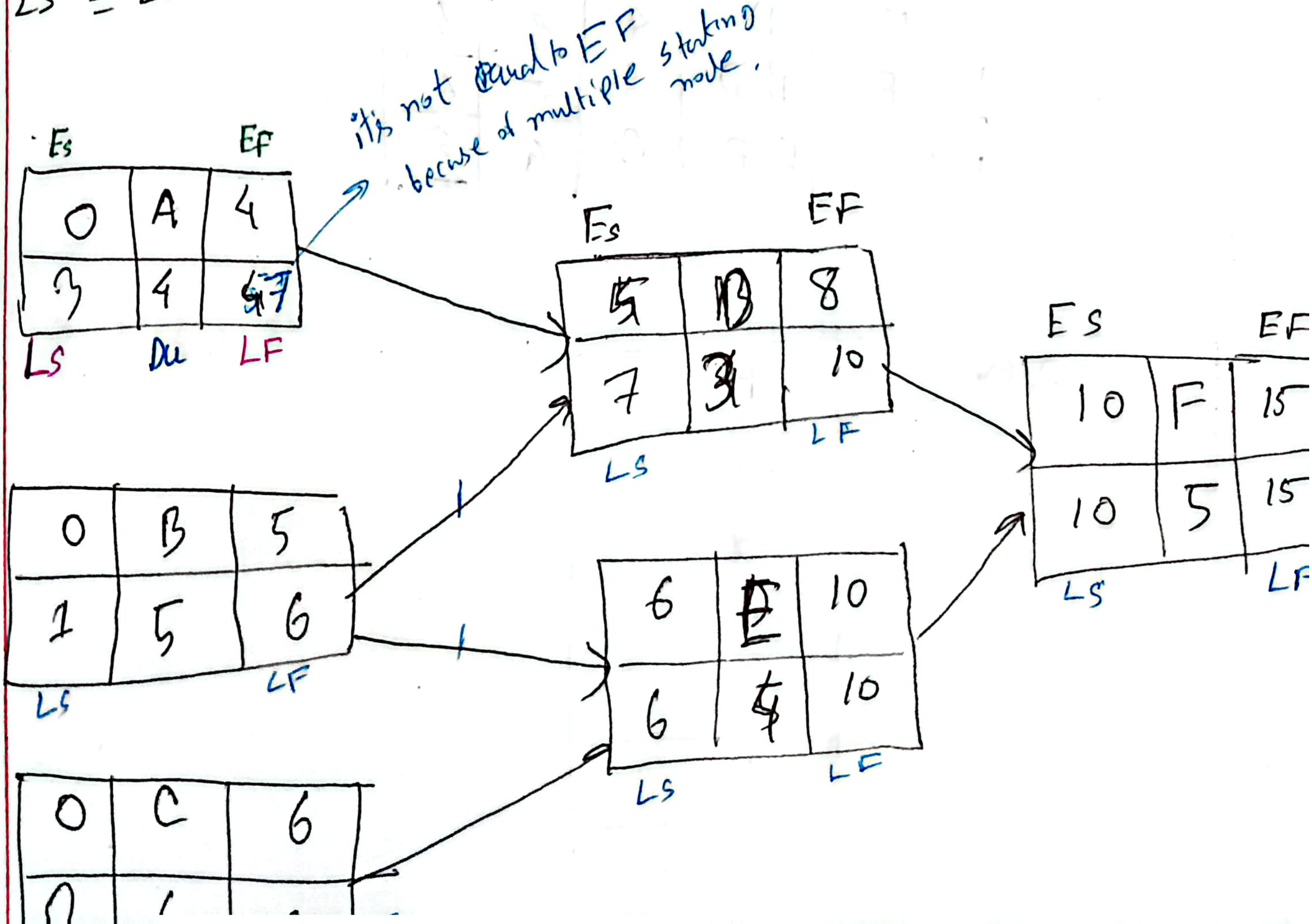
এইখানে A, B, C এর কারো ওপর Dependend না, এজন্য এরা হল Starting node

For these 2: Es = 0

Task	Duration	Depend
A	4	-
B	5	-
C	6	-
D	3	A, B
E	4	B, C
F	5	D, E

- EF = Es + Du
- For next node EF → ES
- Forward pass এ Bigger one
- For last's 1st LF == EF
- For backward Ls যাবে LF এ (minimum)
- LS = LF - Du





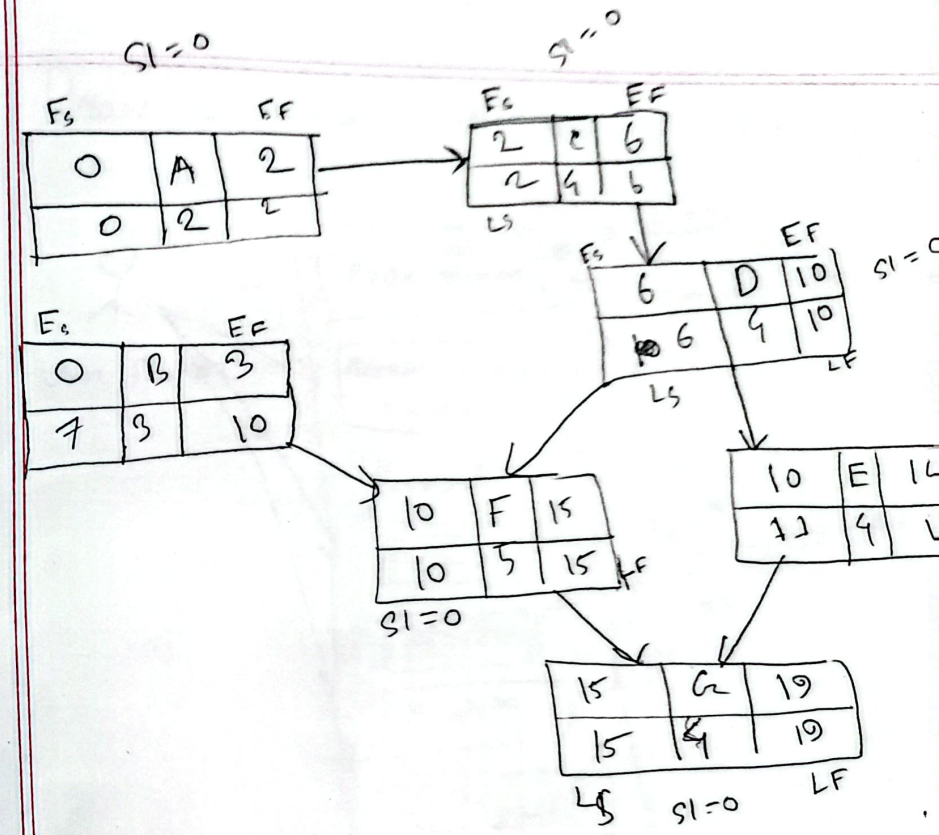
Section e

$$Et = \frac{t_o + 4t_m + t_p}{6}$$

Activity	Pred	T_o	T_m	T_p
A	-	1	2	3
B	-	2	3	4
C	A	3	4	5
D	C	3	4	5
E	D	4	3	8
F	B, D	4	5	6
G	E, F	3	4	5

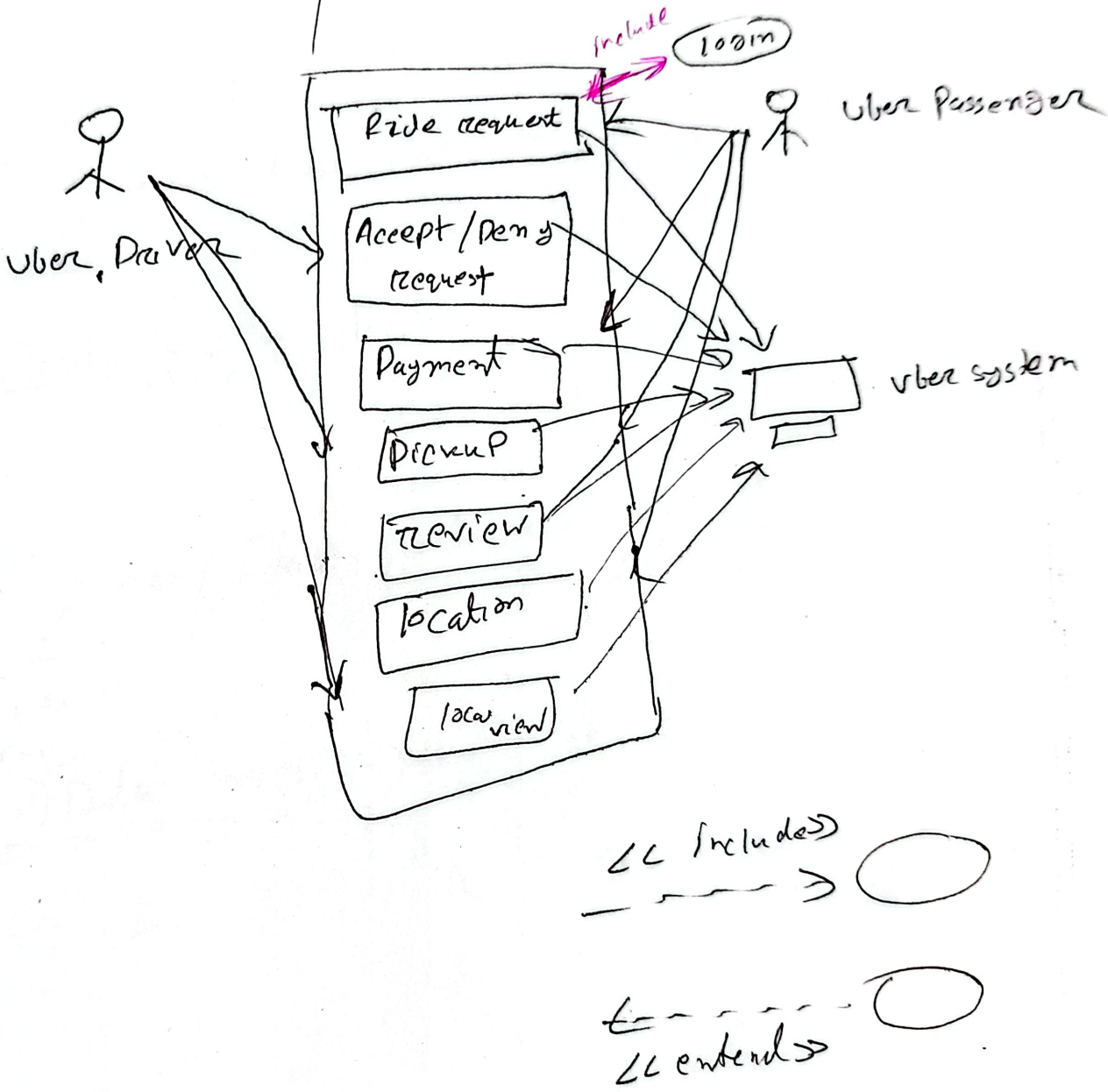
Activity	Et
A	2
B	3
C	4
D	4
E	4
F	5
G	4

$LS = LF - Du$



Critical Path = A → C → D → F →

Draw a Usecase diagram of Uber system.

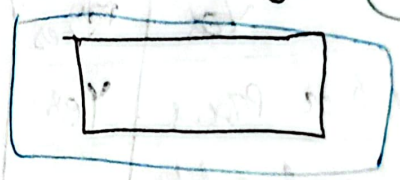


Data Flow Diagram
DFD

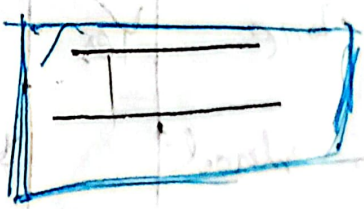
DFD shows how data moves through information System

(I) Data flow : depicting movement of data

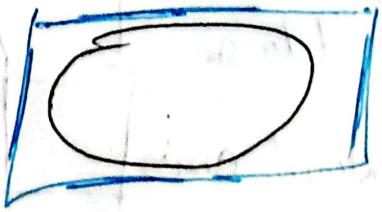
(II) Source / sink : external entity.



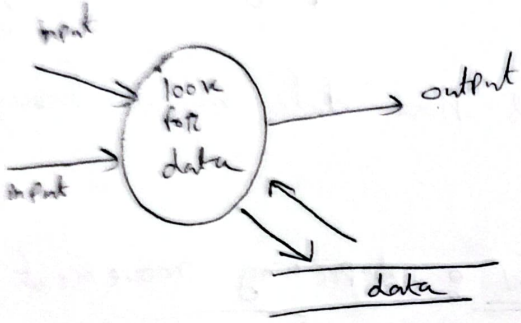
(III) Data Store : data at rest. labels should be noun phrases.



(IV) Process : actions performed on data



Data store Example



DFD Rules

	Yes	No
(I) A Process to another Process	Yes	
(II) Process to external entity	Yes	
(III) Process to data store	Yes	
(IV) External entity to external		✓
(V) External entity to data store		✓
(VI) Data store to data store		✓
(VII) Store to External entity		No

(1) Each Process must have at least 1 input/output

(2) Each data store must have at least 1 data flow in & out

(3) Stored data in a system must go through Process.

(4) All Process in DFD goto another Process OR data store

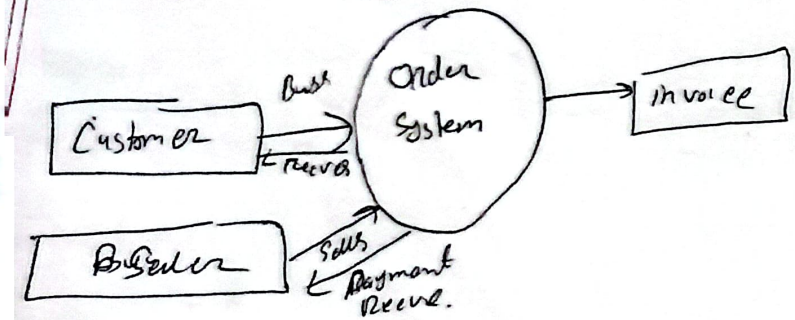
DFD vs ERD

	DFD	ERD
(I)	Data flow diagram	Entity relationship Diagram
(II)	Objective is to represent Process & data flow	Objective is to represent data entity & relationship between
(III)		

DFD levels & layers

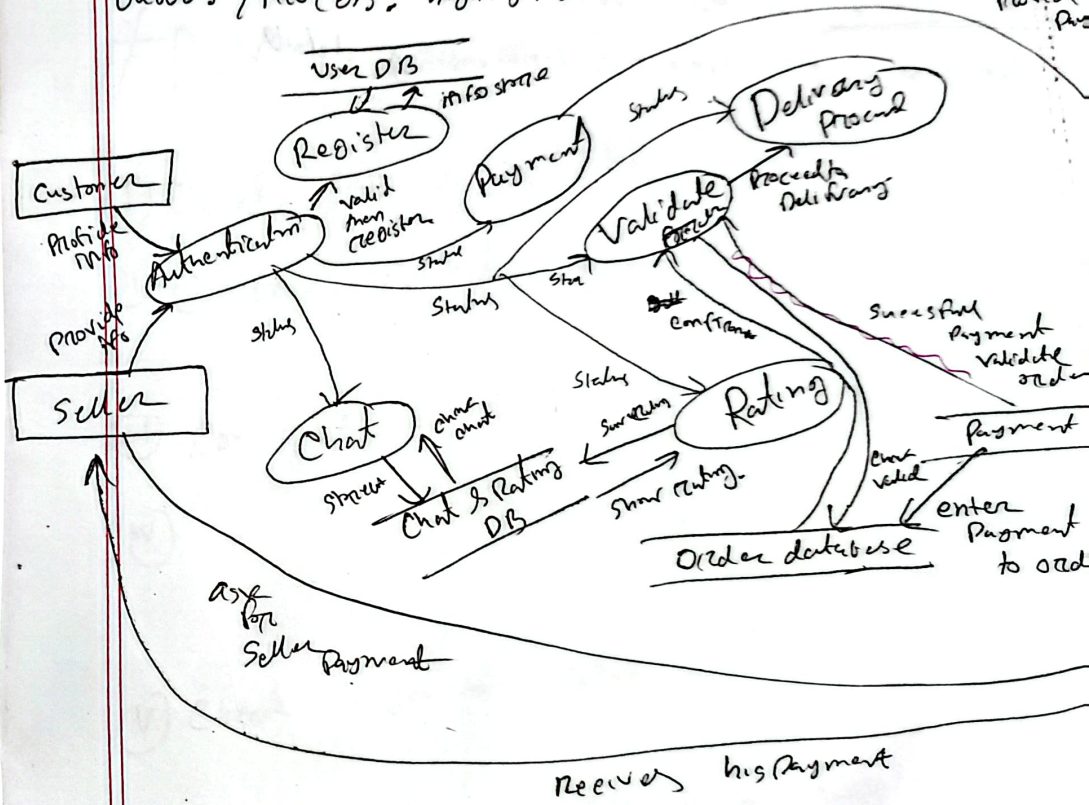
DFD levels are number 0, 1, 2 & occasionally go to even 3 or beyond

DFD 0: is a Context diagram. designed to be an abstraction view showing system as a single Process with relation with external entities. Represent entire system as a single bubble. with I/O we Dont use storage here.



(6,-7) Proces

DFD 1: Context diagram is decomposed into multi bubbles / Proces. highlights the main function.



DFD 2: 1 step deeper than level 1. Used to Plan / record specific / necessary details.

Developing / Creating Dataflow Diagrams:

(I) Create list of activities

(II) Construct Content level / Level 0 DFD

(III) Construct Level 1 DFD

(IV) " Level 2 "

(V) Check against rules of DFD

DFD Naming guidelines

1) External entity → Noun
2) Data flow → Names of data
3) Process → verb Phrase
4) Data sto
→ No

Economic Feasibility

Dimensions of feasibility: Operational, Technical, Economy, Schedule.

Present value (PV): refers to the current value today of an amount of money of income, to be received at a particular future date.

$$P_V = \frac{F_V}{(1+R)^n}$$

F_V = Future value/cash value at a time

R, i = rate of interest

n = time in year.

Net Present value (NPV):

$$NPV = \left[\frac{R_1}{(1+i)^n} + \frac{R_2}{(1+i)^n} + \frac{R_3}{(1+i)^n} \cdots\right]$$

i = interest rate.

n = time in year.

R = assumed cash flows of the investment in ith period.

It is the difference between the present value of expected cash flows and the present value of invested over a period of time.

NPV Decision Rule

- **NPV > 0:** The money earned on the investment is more than money investment, so it's a good investment.
- **NPV = 0:** The money earned on investment is equal to the money invested. Therefore no difference between cash inflows & outflows.
- **NPV < 0:** The money earned on investment is less than money invested. So it's not a fruitful investment.

Return On Investment (ROI):

$$ROI = \frac{\text{Total benefit} - \text{Total cost}}{\text{Total cost}} \times 100$$

Break even point (BEP):

of which total revenu
total cost are equ
this point there is

① BEP = fixed cost ÷ gross profit margin

② BEP = $\frac{\text{yearly NPV} - \text{Cumulative NP}}{\text{yearly NPV}}$

Production > BEP → Profitable (accumulate)

Production < BEP → loss in busin

of the investment from its final value. the dividing this new number by the cost of the investment and finally multiplying it by 100

Fall 23

Given,

investment = 250000

n = 2

Growth in Profit = 200000

$$Roi = \frac{250000 - 200000}{200000} \times 100\%$$

Spring 23

Total fees = 500000 + 150000 + 200000

$$Roi = \frac{1000000 - 850000}{850000} \times 100\%$$

Fall 22

Total Fees = 50000

again Roi for investor:

$$Roi = \frac{1000000 - 500000}{500000} \times 100$$

Fall 22:

total fees = 50000 + 100000 + 150

= 300000

$$Roi = \frac{500000 - 300000}{300000} \times 100$$

= 6.6 x100

=

Activity Diagram

(I) → Start → Starting Point / initial state.

(II) : Activity / action

(III) : control flow, edge

(IV) : object flow, control edge.

(V) : Final node, used to mark end of all control flows within activity.

(VI) : flow final node., end of a single control flow

(i) : merge node

( ) : Decision node.

(VII) : Fork, a flow that may branch into 2 or more Parallel flows.

(VIII) : merge : 2 inputs that merge into one out

(IX) Signal Sending : Sending a signal to an accepting act

(X) : note, comment

Steps to Draw Activity.

Step-1: identify initial & final state.

Step-2: identify intermediate Activities needed

Step-3: identify the conditions or constraints.

Step-4: Draw the diagram with appropriate notation.

Ac diagram for order management

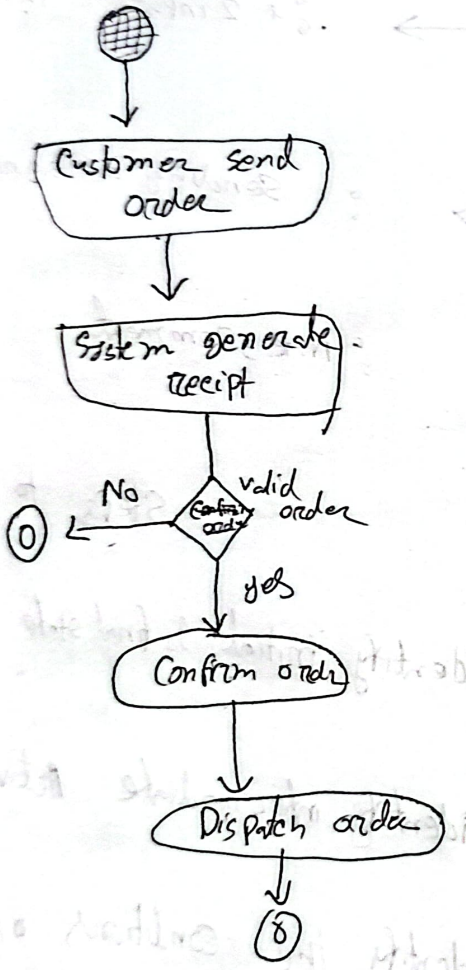
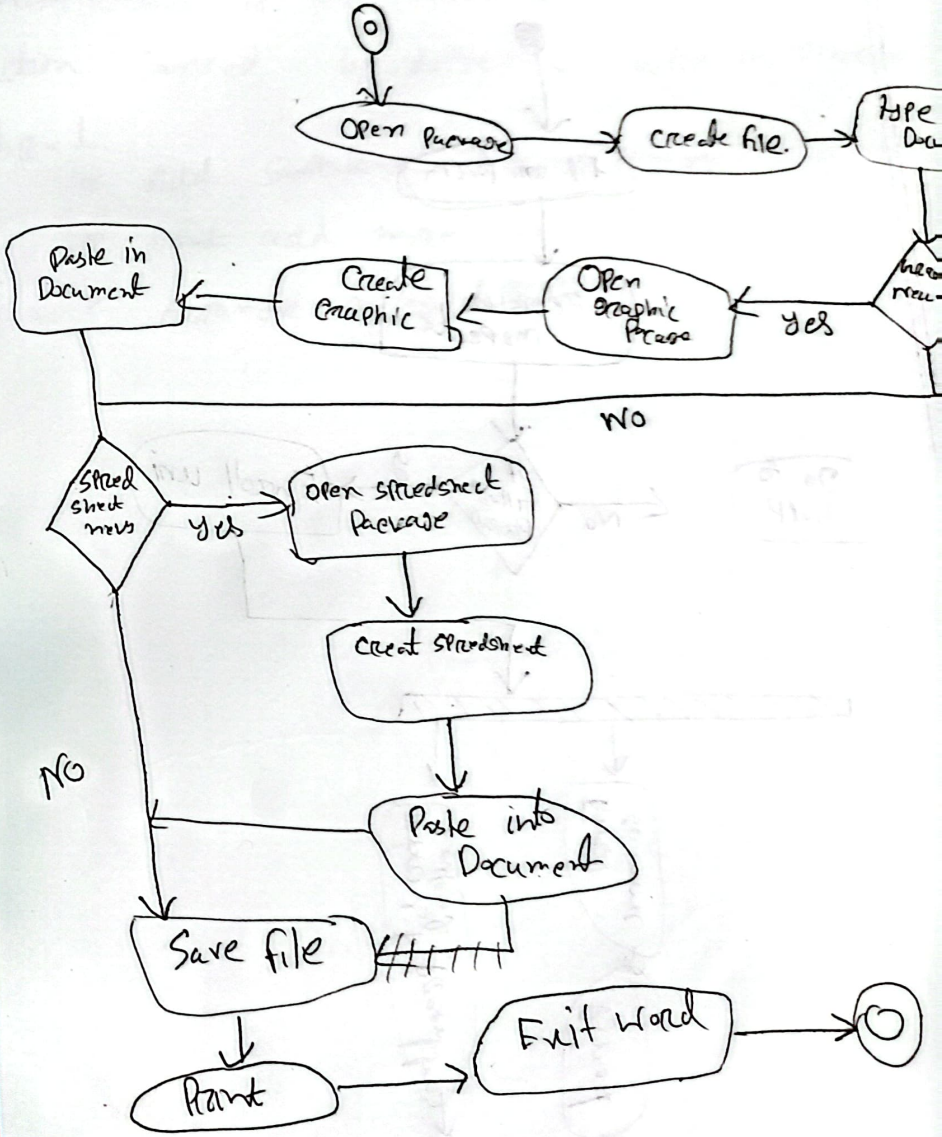
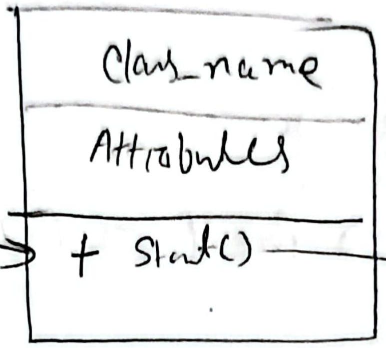


Diagram for word Processor



Class Diagram



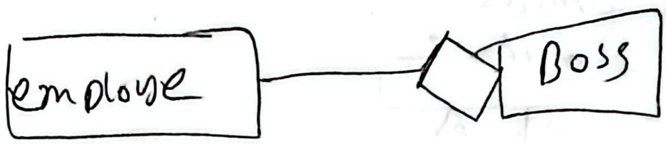
Private Seeding → (+ Start()) ; (+ Start()) → methods

+ → Public
− → Private
→ Protected
~ → Default

(I) **Association** : ——— it means both class are directly connected.

Directed Association ——→ Connected but not directly. with some other way

(II) **Aggregation** :



"Whole Part Relation". Boss এর under এ employee

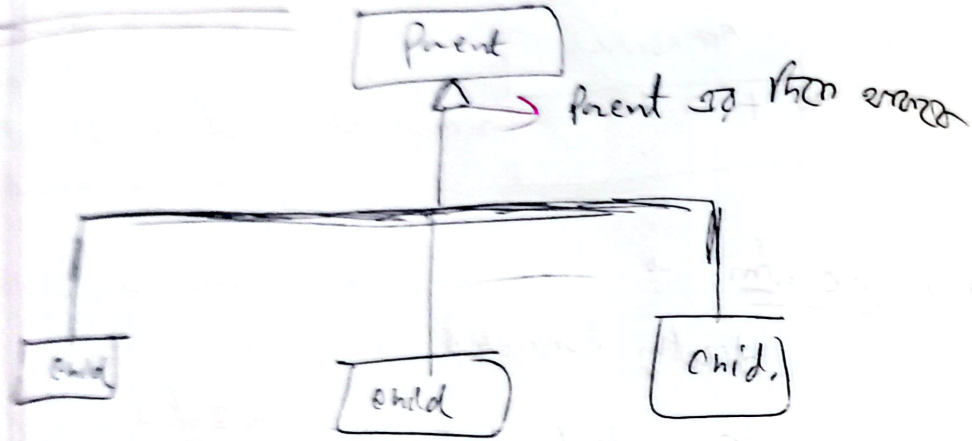
Boss না থাকলে employee থাকে but employee independently ...

(III) **Composition** : its also a "a whole part relation" but more stronger

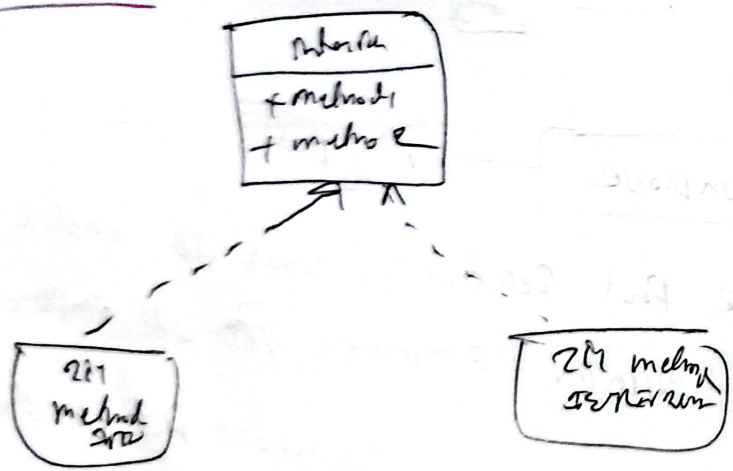
Contact book

~~Company~~ না থাকলে ~~ceo employe~~ কোনো contact কিছুই নেই, independently ও না

④ Inheritance:



⑤ Interface:



⑥ Dependency: One class Depends on another

but not as strong as Aggregation and Compo

Person relys to access book information
But book Doesn't care about Person

⑦ Usage Dependency

Client depend on Supplier to Pe

Certain task

Discuss the Function of swimlanes of activity Diagram

Swimlane in an activity diagram divide the workflow into section to clarify responsibilities of different entities..

Each swimlane represents specific actor / dept that perform task. It visually clarifies is highlight over which entity is responsible for each task

(11) Inspect how an activity Diagram can be used to model business requirement.

AD are used to visualize workflows / steps in a business Process. it Describe the sequence from one activity to another

Steps: Steps of AD

(1) Start & end points of the ~~Process~~ Process

(11) Identify key activities tasks, decision within Process.

(4) Assign swimlane to indicate who performs what

(5) Represent decision points with branches for alternate path

Uses

(1) Use simple tokens to understand business eg.

(2) Identify workflow & bottleneck in process

(3) Provide a blueprint for automation of task

Major notations of activity diagram.

(1) (2) Starting point / initial node : ●

(3) Activity / action : ▭

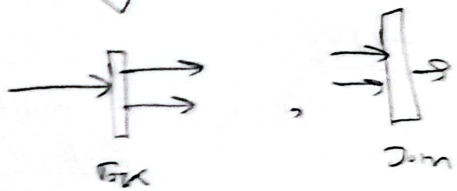
(4) Control flow : →

(5) Final / ending node : ◎

(5) Merge node combine multiple flows into one

(6) Decision node :

(7) Fork / Join node :



(8) Swimlane

Write the purpose of Usecase Diagram.

A usecase diagram visually represents the interactions between users and a system.

(I) To represent system functionality from perspetive.

(II) To define the interaction between user

(III) To communicate requirements to stak

(IV) Outline different actors & their roles in with the system.

Analyze rules & requirement of DFD.

Rules:

(I) Process must have at least one input & one output

(II) Each data store must have at least 1 data flow in & out

(III) Stored data in a system must go through process

(IV) All process in DFD go to another process or data store.

(V) Data store cannot transfer data directly to another data store.

(VI) External entities should interact with process not with data store.

Requirements:

(I) Identify all external entities

(II) Define all process, data store, data flows.

(III) Use a top-down approach, Level-0

What is Notation in DFD & Describe major Not

& Describe the roles of drawing with DFD rul

5, 20

Example: DFD is a graphical representation of the movement of data between external entities, process and data stores with

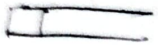
DFD Notations are symbols that are used t represent Components of a DFD

Process: transforms input data to output

emp; order fo

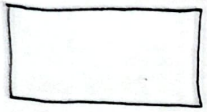
Dataflow: arrow shows Data movement. A path → Data move one Part of me to another

Data store: data at rest



eg: customer database

External / Source: Source of data



eg: customer

Roles of DFD:

(I) Analyze & model the flow of data in system
(II) Identify system boundaries Process & data Sources
(III) Improve systems data handling

Diagram Example:

What are the common error in Drawing D

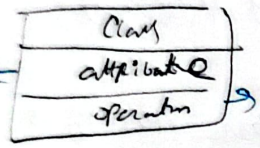
(I) Missing Processes:
(II) Improper naming → Vague name for Processes data flows.
(III) Data store to Data store → Data should not Directly flow betwee
(IV) missing Arrow: Omitting directional arro data flow
(V) Too many levels → overcomplecating the with unnecessary de

Advantage of DFD

(I) Used as a part of system documentation file
(II) Explains the logic behind the data flow

Describe the major notation/components of class Diagram

Class: Represented as rectangles with 3 parts name, attributes, operation



Also known as the Property or field represent the data members of the class

known as function or methods represent the behaviour or functions of the class

Relationship:

① Association : Simple line connects 2 class

② Aggregation : A hollow Diamond pointing to whole class

③ Composition : A filled diamond pointing to the whole class.

④ Inheritance : A solid line with a hollow triangle pointing to the Parent class

⑤ Dependency : Dashed arrow pointing to Dependent class.

Visibility:

Public = +
Private = −
Protected = #

(CD) is a Static structure Diagram t
Describes the structure of the syste
by showing system class, attributes, & meth
and relationship among object.
A class is a blueprint for an o

Categorize relationship of class diagram wit example;

	Relationship	Description	Example
①	Association	Simple link between class	Student ↔ Cou
②	Aggregation	A "whole part" relation (weak)	Department ◇—
③	Composition	A "stronger" "whole/part"	Car ◆— Engine
④	Dependency	One class depend on another	Order → Paym
⑤	Inheritance	is-a relation	Dog → Animal

Difference between Activity and sequence Diagram.

	Activity	Sequence
(I)	Describes work flows of activities (Process)	Shows object Interactions overtime
(II)	Focuses on operation & logic flow	Focuses on time based operation
(III)	Represent activities Decision	Lifelines, messages, activation
(IV)	System/Business Process	Interaction between Objects

DFD Usecase, class

	use Case	Class	DFD
(1)	Describe System functionality from user perspective	Shows static structure of system	model flow within system
(2)	focuses on user role	Focuses on class, attribute.	Focus on Process
(3)	between actor & usecase	Association, inheritance	Flow be Process &
(4)	Strict frame	rectang	Circle

Explain the purpose of using Sequence Diagram.

A sequence diagram represents the interaction between objects in a time sequential order.

Purpose:

(I) To show flow message over time.
(II) Describe how objects collaborate to perform function.
(III) Identify the lifelines of objects and their interaction.
(IV) to visualize the order of method calls & response between components.

(V) Easy to maintain
(VI) Can easily updated
(VII) Allows reverse and forward engineering

12.16

Major Notation of sequence Diagram

(I) Actors : Stick figures representing user.

(II) Objects : Show system components, class/object

(III) Lifeline : Dashed lines shows object lifetime

(IV) message :
1) Synchronous (solid arrow) → (Sender waits for the response of receiver)
2) Asynchronous → (Doesn't need Respons)
3) Return message ←- - -
4) Create message - - →

(V) Activision Bar : thin rectangle along lifeline
Object execution

(VI) Destroy (X)

Compare types of Arrow in sequence Diagram

	Synchronous	Asynchronous	Return	Create	Delete
①	→	→	<---	--->	X
②	Sender waits for response before continue	Sender continue without wait	Return of control or data back to sender	message to create new obj	termination or deletion of an Object

Drawback of sequence dg

(I) Can be complex when too many lifeline in

(II) if order of message change incorrect result

(III) each msg needs to be represent different msg

(IV) types of message decide types of sequence in diagram

Define Return on investment

it use it's a use to evaluate the profitability of efficiency of an investment. measures gain or according to the investment

$$RoI = \frac{\text{net Profit} - \text{cost of investment}}{\text{Cost of investment}}$$

How is RoI Beneficial for a Business?

(I) helps management allocate resources effe

(II) Assist investors in Comparing high and low va investment

(III) Aids in evaluating potential gains from va Opportunities

(IV) Supports organizations in cost analysis and some risks

Notations for Usecase Diagram

(I) **Actor:** Actors are individuals involved with Syst
these are human or other external Sys

(II) **Usecase:** usecase describes how actors uses
a system to accomplish a particular goal.

(III) **Relationship:** Relationship between actor & usecas

(IV) **System Boundary:** Define system of interest in
to the world around it

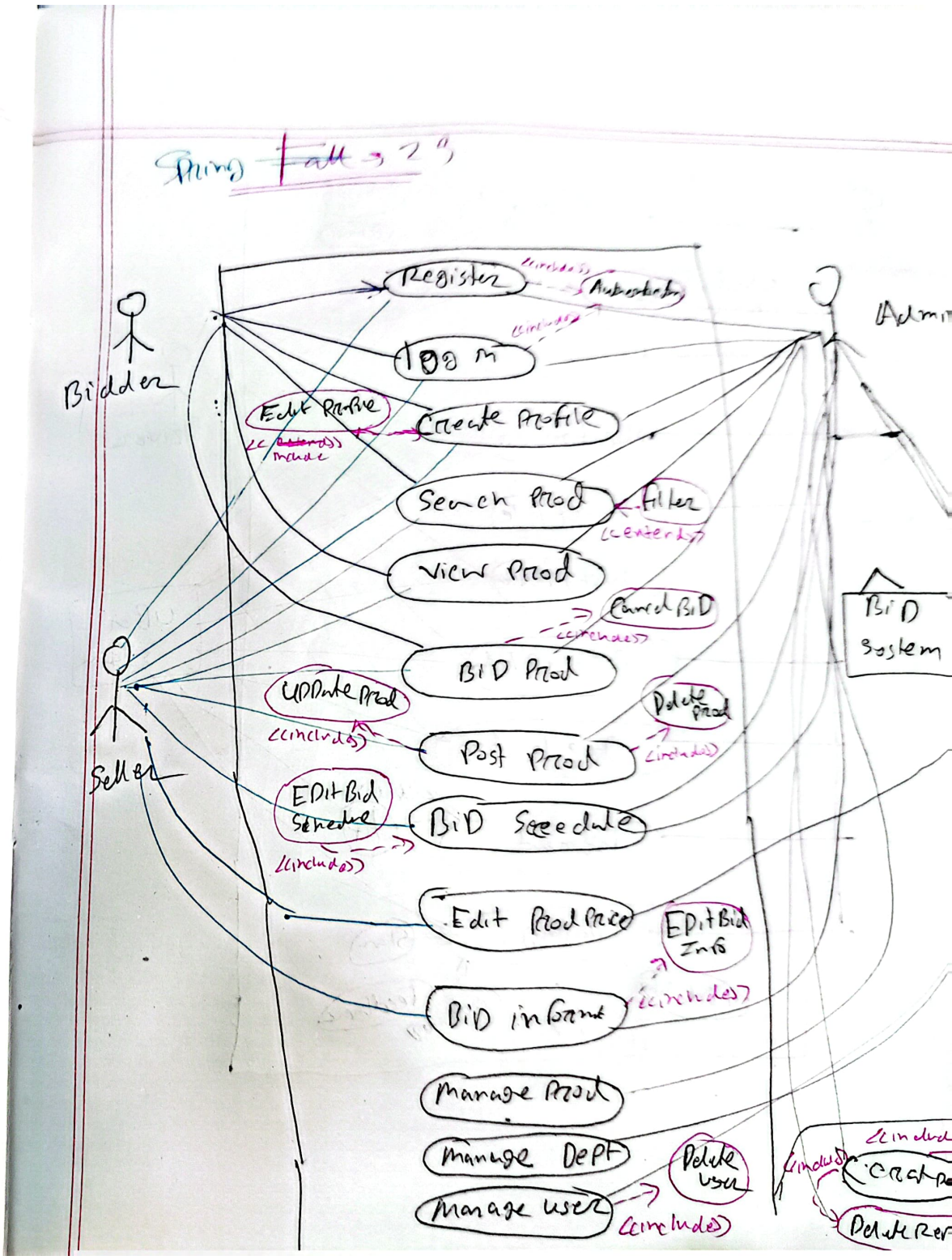
Benifits of Usecase Diagram

(I) Powerful technique for the Proces of develop
and documantation of functional requirement

(II) Easy to understand & excellent way to communicat
Customer

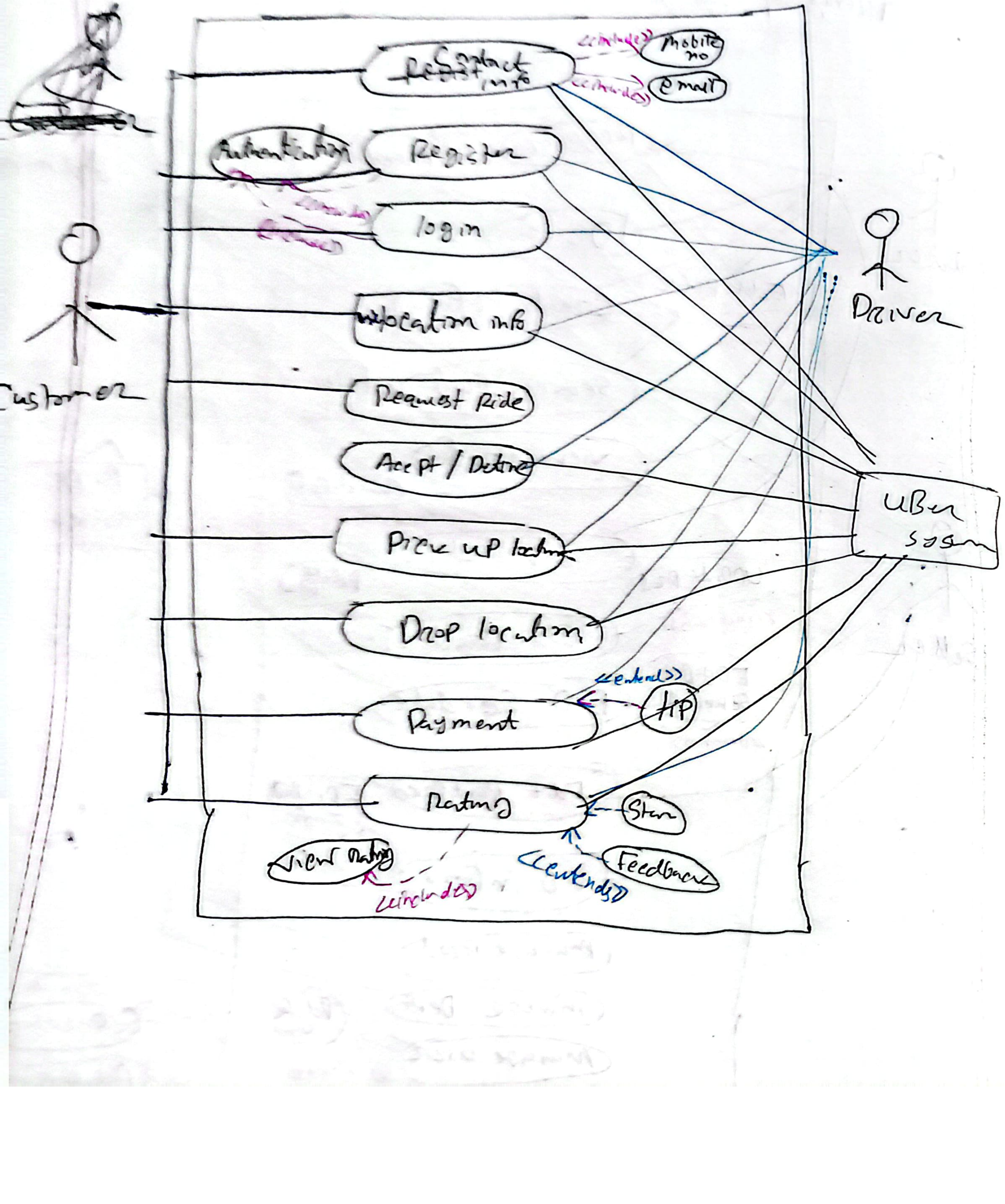
(iii) help to manage complexity

(iv) Involves collaboration with multiple object

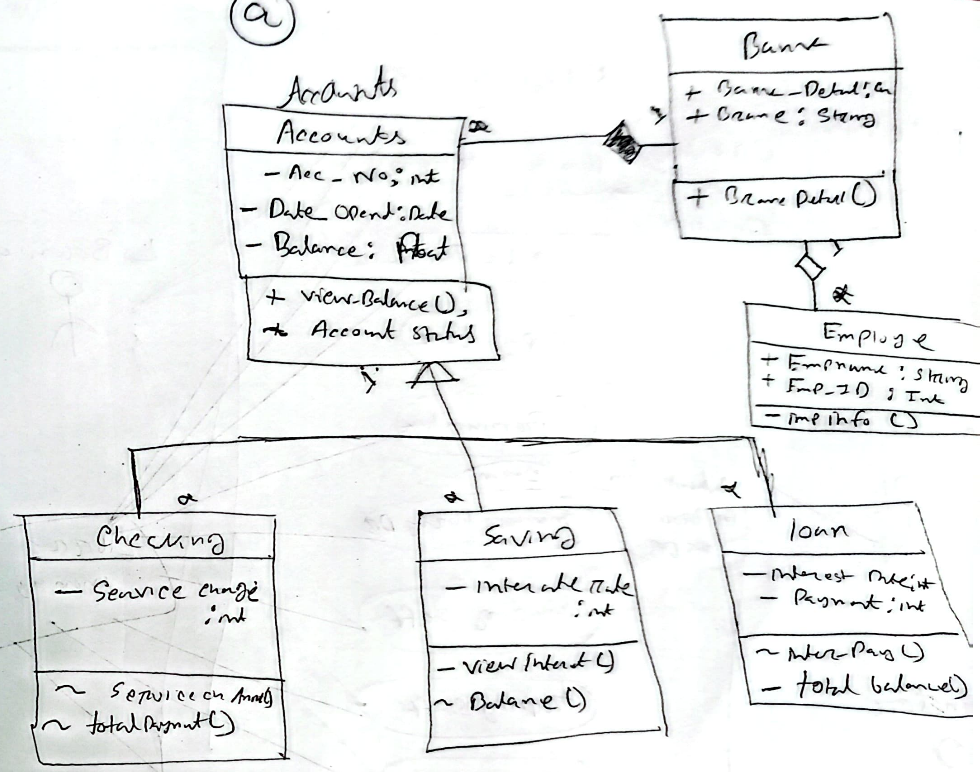


1 Account

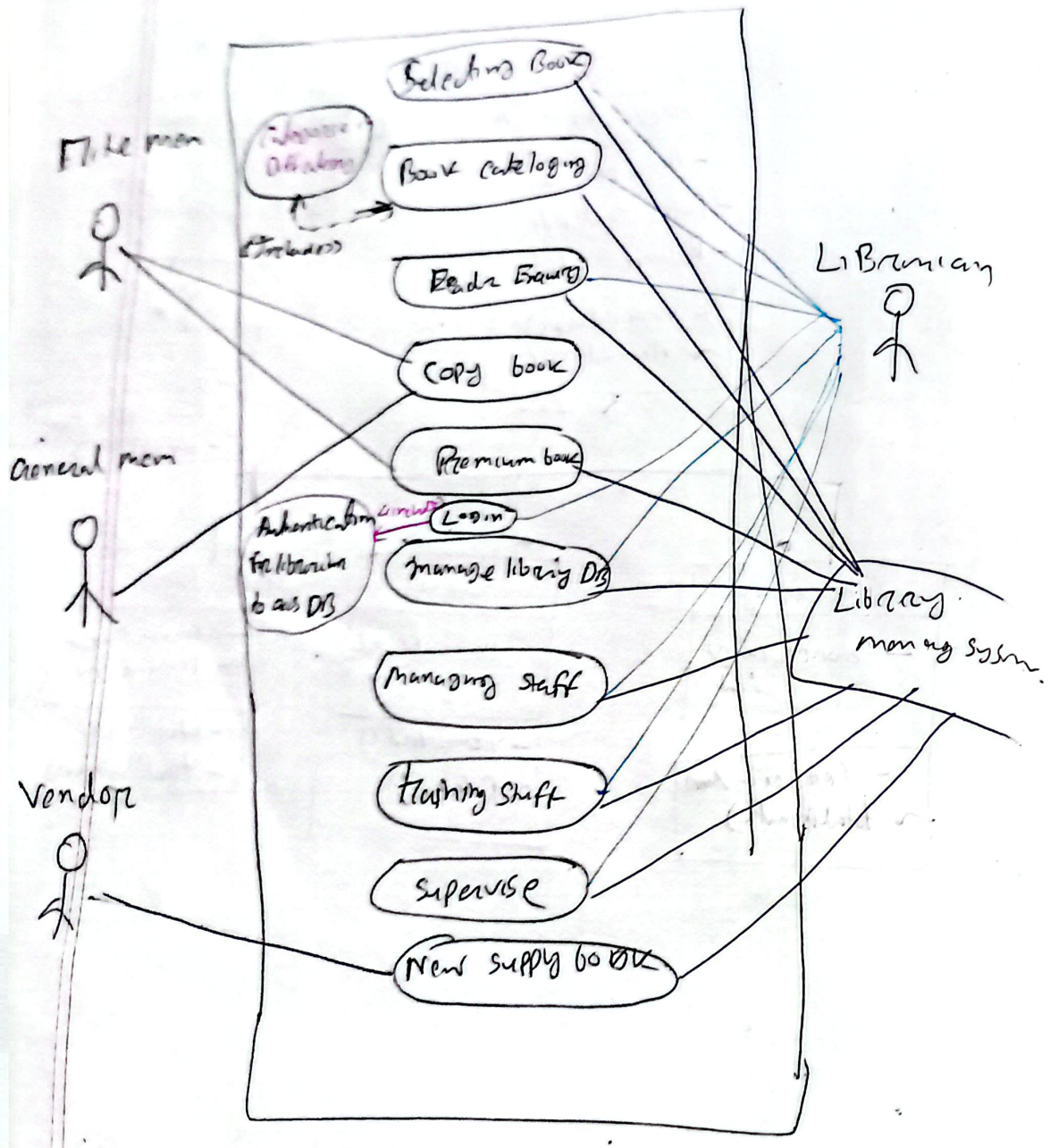
5)


Customer
Driver
Contact info
Mobile no
email
Register
Authentication
login
location info
Request Ride
Acept / Decline
Pick up location
Drop location
Payment
Tip
Rating
Star
Feedback
View Rating
<<extends>>
<<includes>>
Uber system

(a)


Accounts
Accounts
- Acc_No; int
- Date Opend: Date
- Balance: float
+ view-Balance(),
+ Account status
Bank
+ Bank_Detail; ...
+ Name; String
+ Bank Detail()
Employee
+ Empname; String
+ Emp_ID; Int
- Emp info ()
Checking
- Service charge; int
~ Service on Amount()
~ total Payment()
Saving
- Interest rate; int
- View Interest()
~ Balance()
loan
- Interest Amount
- Payment; int
~ Inter-Pay()
- total balance()

Spring →22



Fall →22