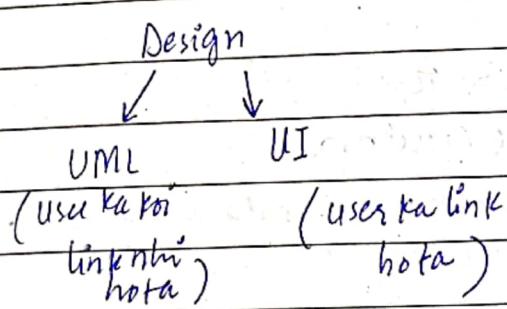


Amna nibaee
2022-1695

Software Design & Architecture

Date 41 Feb 2022.



design :-
eg:- (Map)

① Design important isliye hota ok
graphical representation se kaam
karna asaan hojata hai or error
ki chances kam hote.

② Conceptual Solution
(Object Oriented Parikram karte)

③ Customer Oriented hojata ok
program efficient hojega.

From Design to implementation:

Analysis

- > (problem ko investigate karte)
- > (yahan se ek problem statement ajeti)
- > Object identify kre or using domain knowledge.
- > domain info k through analyse karte.

Design

- (logical sol)
- > Analysis ko design krenge.
- > Clear hojaengi chezen

eg: class diagram.

Construction code.

- > design ko code mein convert kardete hain.
- > Using any Programming language.

Bright

Date _____

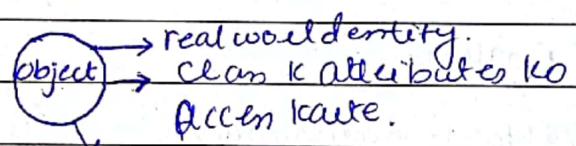
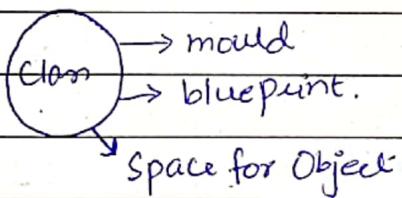
Procedural

- Top down.
- function Oriented
- Changes ko accomodate karna asan nahi.
- Repetition
- Carried out using structured design and structure, Analysis!

OOP

- Bottom Top
- Object Creation.
- Changes ko accomodate karna is easier in OOP.
- Inheritance & repetition kam hojati hai.
-

Class And Objects:



OOP Pillars:

- ① Inheritance (is-a relationship)
- ② Abstraction (relevant data, security & privacy purpose)
- ③ Polymorphism. (run time (overriding) & compiletime (overloading))
- ④ Encapsulation. (Data hiding).

Bright

Date _____

Software Development LifeCycle:-

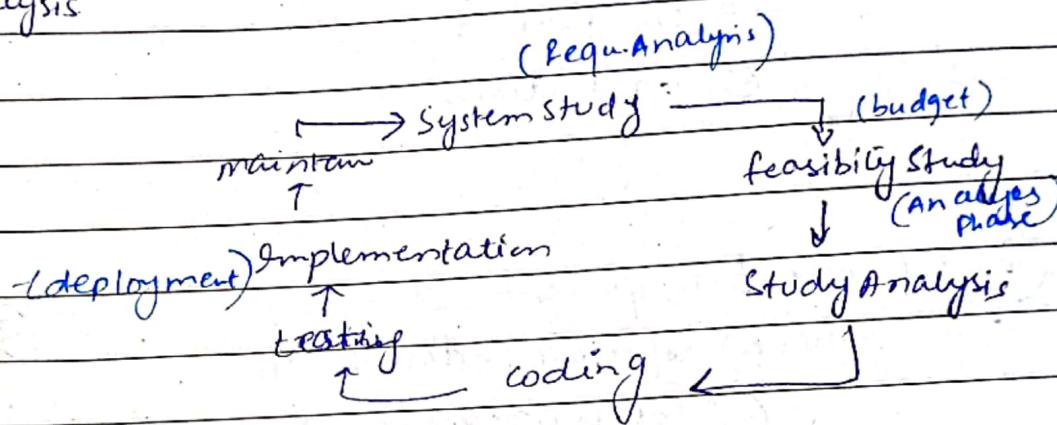
1) Requirement Analysis

2) Design

3) Coding

4) Testing

5) Maintenance.

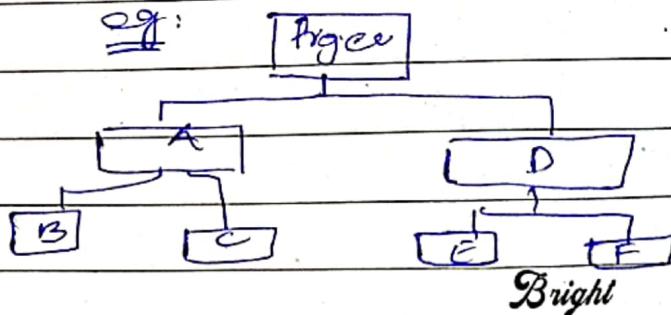


① feasibility study :- market value laya hogi.
acceptance milgi yani.
-- budget.

② System Analysis :
→ plan driven approach
→ agile approach
(System mein konsi approach use hogi)

③ Testing →
Support in Operations ka department sare mästle dekhta.

④ Structured Approach :- functionalities ko refine karte or
Step by step kalm karte.



Software Process Activities:-

Date _____

(1) Software Specifications:-

- ① Requirements to define & detect.
- ② Constraints.
- ③ Limitations.

(3) Software Validation

- ④ User acceptance.
- Verification mein msla agaya tu Valide se karte hain unko!

(2) Software Development :-

- ① Software to design krite.
- ② Software program karte.

(2) Software Verification

- ⑤ Developers apni taraf se dedekta hai apna best len requirement collection ya mein msla ho ya communication gap kia waja se msla ho!

(4) DevOps → Continuous integration.

Development + Operation team ke
Gth bethti.

→ take quality improved ho.

(4) Software Evaluation:-

.. User Acceptance

Bright

Date 4th Feb 2022.

Software Environments :-

① Development → Local machines / Virtual machine server.

VSP → virtual server provider.

② Test →

③ Staging → version controlling (developer environment mein live ho jata hai)
→ customer developer environment mein
④ dummy data par kaam kar rhe hote aks dekh skta.

⑤ Pre-Production → customer k environment ki virtual copy bnai or usko test kiya or bugs ko set karte.

⑥ Production → live Data par kaam kar rhe hote hain is stage par. ⑦ live transitions aati → live data par aapka system errors dena shuru krolet hai.

⑧ Mirror → virtual server bnaya, bugs ko identify kiya, log set kara. or phr database production par lejate.

Roles involved: B, D.M, P.M, T.M, C.M & P.T.

Bright

DS → developer

DM → development manager

PM → project manager

Tm → Test manager

Cm → configuration manager

DT → deployment team / implementation Team

System Development Life Cycle:

1. System Environment

System Study :

feasibility study :

System analysis.

System design

Coding

Testing

Implementation

Maintaining

① Development: This could be ~~both~~

developer's machine or any server jis se
login kare uska kam aur the hor / end user
PC.

② Test: development or Testing parallel

horahii hoti hain ek version pc develop.
horahii hoti or Sth Sth testing.

Tm responsible hota hai iske liye (testing)

Peg & hisab se testing karte.

③ functional testing (related to
non Performance)
reliability

④ non-functional : (related to features)

⑤ B testing:

users se testing karwate.

⑥ B versions korebase karte or
test karwate.

⑦ Mirror: Mirror ki raene ke usi wqt

jib ap production ka debug
karne the hota ho, production kya hi detta

mirror bnana hota hai, mirror ka

environment hm client's site par hi create

kahte hain. limited time mein debug
karna hota hai.

Agar limited time

mein nhi hota, two teams hoti hain

hamare pas CM & DT ki wo script hain

hain & ese limited time kar dehn debug

kr skte. Mirror ki transition phr production

par hoti hain. (Copy of Production)

mirror
Bright

⑧ Staging: live application ki

versioning, requirement changes ko

accommodate karna asan hota.

Staging usually 3 stages hain

hoti hai cause its expensive to

create 3 P stages of same program.

→ but if you are client centric

tw ap staging kartे,

has stage pr system (Pma)

deploy horaha hota.

⑨ total 3 stages hain... age or

bnani hain tw stage 4, 5, 6..

1, 2, 3 ko override krdengi.

yahan B testing nhii hoti.

⑩ Pre Production:

hardware or software ki

duplicates kare apne envir.

mein bna kar unpe kam

kaaste / errors dekh skete.

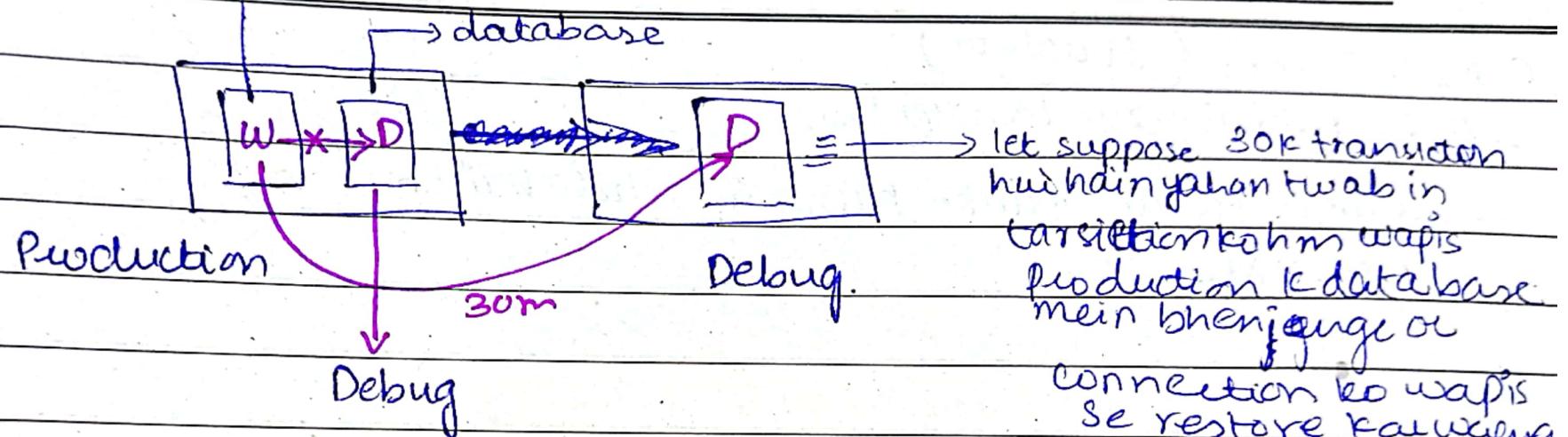
⑪ Production: live data

need data.

Mirror:

webserver

Date _____



Date _____

① Agile Process: (Iterative)

-- Use interaction 2yada hota hai.

-- changes to accommodate kaena asan hota hai :p

e.g.: XP.

Waterfall:

① backtracking allow nhi hoti.

② Plan driven hai Ikn

③ Detail Analyses ki zarurat hai
two plan driven is fine.

④ time consuming

⑤ Cost of change could be expensive (Expensive)

⑥ dubara sephase start karna parta
hai,

⑦ Changes of Accommodate kaena asan nahi
ek bar project complete hojaega uske bd
changes hangege, "Change of Scope".

Shortcomings:

⑧ Uske bd kisi ne kaha t backtracking allow karden (esa keli)

⑨ Resources zaya hote. (Project At a time kisi ek team ke
par hoga Ikn baki team faing
hongi).

Bright

Software Development Models :-

Date _____

Incremental :- Working project deliver ho ja hai ha cycle not necessary
not necessary

Iterative :- K bad./ meaning full throughput.
-- srs --

he cycle k bd hm user se interact karte.
Iterative >> Incremental.

Incremental :-

Increments mein development ho ja hi hoti
hai. yahan bhi user interaction zyada hota ha

Agile Manifesto:

Requirement >> > Requirement (2)
(slides).

Scrum: (Agile methodology).

more toward the planning of project

- ① daily stand-up ② Sprint Planning ③ retrospective
- ④ feedback.

→ Sprint k end mein jo prototype niklega wo throw away
nhi hogा. (real prototyping).

→ Possible hai k sprint mein developer k saath customer
beta ho.

→ hr sprint ka result user se satisfy krwate. casual puchte
etc.

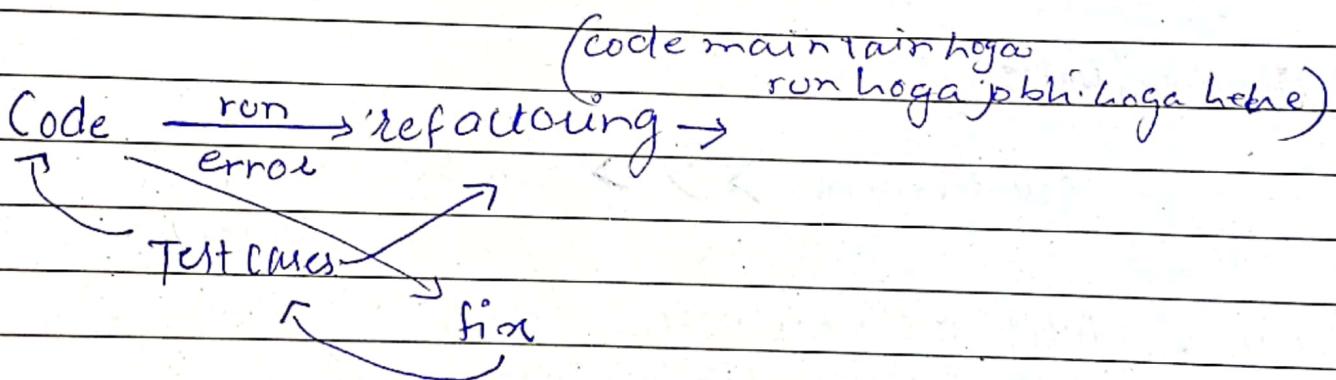
Bright

Date _____

XP :

more towards the development of the project.

- ① Pair Programming.
- ② Onsite customer.
- ③ Iteration mein hi hota hai kaam.
- ④ TDD → Test driven development.
 - requirements se test cases nikalte hain.
 - test cases pehle se define honge.
- ⑤ Its more toward quality of product and error handling.



- ⑥ Iterative and evolutionary Development model.

Page (14 -)

Evolutionary:

Ex) Project → ATM → requirements gathered (4 weeks ki ek iteration hogi.) → Use Cases →

upfront analyses → iterations start hone se pehle hi requirements
↳ Req. to analyze Bright challenge -

Date _____

Ques 02

reh functionalities ko choose karenge (jo max. business value return krega) jo hm first iteration mein netven kaerenge.

Iteration: design Phase(i)
→ designing → coding

Last week mein ek phase aega jisk code freeze kehte usmein hm saare teams apna kaam rok deti hain wo testing ki taraf chale jate hn agr kisi ek team ka kam reh gya tw iteration ka time nhi brhaenge usko backlog mein daal denge or bad mein refine kaerenge

backlog maintenance:

Jo reh jati cheerzen unhen

backlog main daal dete.

Risk driven Approach :- Evolutionary / UP.

① descope : Agr iteration 1 mein koi req. reh gai hai tw usko descope karte size nhi brha iteration ko next iteration main req. ^{comp} karte uske bad baqi kaam karte

Bright

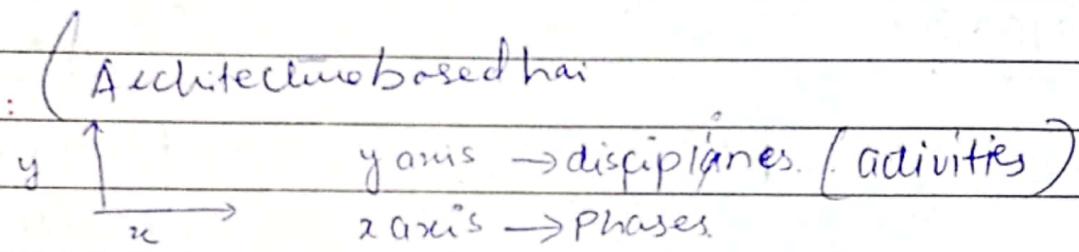
Date _____

Backlog:

Iteration stage	login ✓	} yahan iteration	Product & save features backlog mein hote. Com-phagai
	code ✓		
	design ✓		
	testing	} → testing ko backlog mein daalde hain.	

Stakeholders & other meetings haur hain wo bhi within the iteration hoti hain.

Unified Process: (Architecture based hai)



- ① Risk driven
 - ② Client driven
 - ③ Evolutionary approach pe kam kar rha hota hu client ka interaction syada hota
- AK iska architecture.
Pehle hi bjata hui.

Phases:

- ④ Inception:- hm decide karte hain k jo project hmare pass agaya hain wo hmare pass worthwhile hai bhi ya nahi.
- ⑤ vision → scope → business value → target Audience → uniqueness.
- ⑥ feasible [Generic product
- ⑦ G [Customize product

Bright

Date _____

① Buy OR Build → scratch se start karega & buy kilen ya
naya staff hire karen. q k customer ko
mha nhi kar skte.

② Rough Estimates of cost and time:

rough estimation deni hogi k. kitne tk ho jaga project

Flex

R. Estimation : 50,000

Actual iske qareeb qareeb hona chahye.

Artifacts: (Basically output) [Modeling k aur domain model Output
hoga]

③ Business Case or Vision Artifact
(Documents waaja bnege).

④ High level use case diagram.
→ 10% use case brijat hain.

Evolutionary

Incremental

UP

Bright

UP.

Date _____

(2)

① Elaboration

$E_1, E_2, I_1, I_2, \dots \rightarrow$ workshops.
Iterations.

Jo saa evolutionary model smjhia hai elaboration mein aara

① Requirement Specification

→ requirement ki workshop reakte hain.

② Elaboration mein hm 7 phases se hote hue ja raho hote hain.

③ Requirement gathering + designing (16% of implementation)

④ Conceptual image bnti, ~~new~~ working product nahi hota.

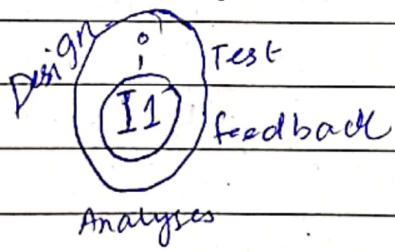
⑤ Phase ke end mein requirements stable hojati hain.

⑥ Architecture design hojata.

⑦ 90% requirement gather hojati hain is phase mein.

⑧ Usecases ko define krdia. / detailed use cases.

⑨ code



Artifacts:

① Domain Model

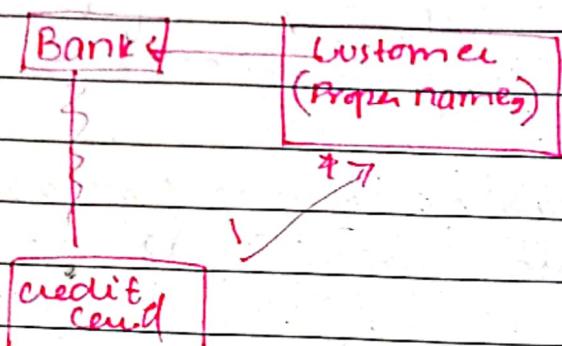
② Descriptive use.

⑩ Vision Box → technique for gathering requirement gathering).

Bright

Atm System:-

Date _____



Construction:

- ① design ko implement karne ke note hain is phase mein.

Transition:

- ① ~~des~~ deployment
- ② integration.

Architect design bhi base expenties kahte hain cause complex hain.

Pros:-

Cons:-

Bright

Date _____

Domain Model :

- ⑤ (Association Relationship)
- ⑥ Conceptual Model - b/w Rechte hain Domain Modeling to
- ⑦ Identify Entities.
- ⑧ Define relationship b/w entities.
- ⑨ nouns to identify krlo.

Scenario:

Valid classes:

- ① Fast
- ② Reward
- ③ Students
- ④ Courses.
 - syllabus(attribute)
 - Semester (attribute)
 - Exam
- ⑤ transcript:
 - grade
 - backlog
- ⑥ Admin
- ⑦ instructor.
- ⑧ Schedule.

Bright

Date _____

Semester ki class banegi aur us
se courses/subject ki class
Associate hogi.

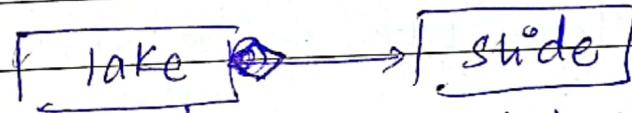
Dependency Relationship :- (object as a parameter pass hota hai)

dependent class ka parameter se submit

dotted line hai.

⊕ Association mein has a relationship hota hai.

Aggregation → 2 classes hain unke daemayan hai.
The hollow diamond is next to the whole.



Yehaega tb bhi slide ke class hogi.

Composition → ownership wali bt hogati hai.



Yehai tu B bhi jaegi.



Bright

⑥ 8:37
⑦ Friday
⑧ 25' feb' 2022
Date

Understanding Agile mindset

Kin overheads ko ignore karke rapidly software create
ki jasake direct working software miljae.

Mindset → Rapid development.

Principle & value → Working style. (Agile)

Practices → XP, Scrum.

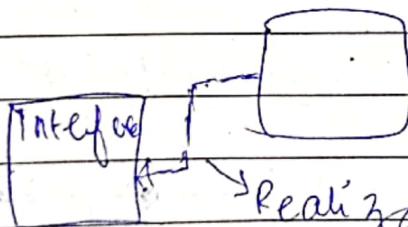
→ These models are framework.

theories,

mindset, philosophy, manifesto mein difference Pta
nonavalable.

Interface:-

- It is a blue print of class.
- abstract class declare hain or usmein koi method
declare kardete hain phr usko baghi classes orein
use karten



Realization. annekhead interface ki taaf
hoga.

Java main inheritance ko support n karta.

Bright

Date _____

class child extends parent A (parent B) → erroe in java.

③ ~~multiple programming~~

multi inheritance is not allowed.

isliye java mein ham interface key word use karte, through
2 interfaces.

e.g:

Class Child implements Parent A, Parent B.

UML ki domain mein interface ko Specifier bhi kہلتے
hain.

Aggregation:

ek class doosri class ke functionalities ko
use karegi. / ownership nahi hoti hai.

Aggregation and As Composition us same code hiske liye
Association ka subtype shahn.

Association

inheritance dependency Aggr. composition.

Association has a specified direction.

Bright

Date _____

com

Composition mein object const & and & bnega jo tc bar
bar call hoga. ek class ka object bna hwo sth
zame bnega doosra.

Public class

```
{ A() { B b=new B(); } }
```

Aggregation:

Aggregation mein object method & andar
bar raha hoga hai wo zame nhi hota bar bar
call ho jb tk hm call nhi karte call nhi hota wo.

class A{

```
void func() {  
    B b=new B(); } }
```

Parameterized class:

Parameterized dan kohm java mein o csharp mein generic
& concept se hata.

```
Class LL<T> { T value; T extense }  
main() { Linker LL list=new LL(); // curser  
Linker LL<int> list=new LL<int>(); // right  
process }
```

Bright

Date _____

↑ binary relationship.

Class ko bind karte/ link karte

type bind class hogi, int is bind and Square class
is also bind.

use define
(enumeration) →

Ye array ki tarah use hota, array create karke har variable
ki different data type deserve.

Agile → (rapid)

To develop and deploy rapidly. (want to change)

Use Case Diagram :-

① Brief

② Casual

③ Fully

④ Tools for gathering functional requirements.

⑤ Pictorial representation :-

⑥ By UCD you get to know the context of your

⑦ jahan deploy hoga, user kon hai kesa hai ye SB context mein
data

⑧ User or system ka interaction dikha rha hota.

⑨ non technical bude ko bhi smjhi aijata hai iski functions
Provide kar Jayati hai.

⑩ Use case Should be non technical.

Bright

Inception ke phase mein architecture bijata h Date

① Use Case se Architecture kya chalta hai?

② A links ③ boundaries ④ functionalities
se pta lgjata.

Actor: human bhi hostka os technology bhi.

③

type of Actor

1) Primary → jinko koi goal achieve hoatha hoga eg: environment / male

2) Secondary / Supporting :- Admin

3) Off Stage → koi kya contribution nahi hota :P interested in
behavior no contribution, high power low
interest
(director)

Use Case name:

[Verb + noun]

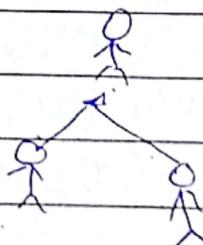
Relationships:

Association → Simple connectors b/w actor and User.

Include → must

extord → (can); Optional

↑ between actors - is Possible & P



Bright

4

Date.

Use Case Level:

i) Brief Use Case :-

brief Use Case :-
we will define the functionalities of ~~the~~ Scenario. ^{given} in sentences.

eg: Kehl's auto Algo-haha. (main flow)

3) Casual Use Case :-

multiple paragraph has alternate lines; alternative bni define
kaen ge. (main flow+alternatives)

③ Fully Dressed Use Case :- / detailed / descriptive / narrative.

Same aspects to analyse for each note.

- ① UseCase.Name → starts over a verb eg place Order

① Primary Actor → Customer } Stakeholder

① Secondary Actor → JO Primary Actor (System to Service)

① PreConditions → login, menu must be accessible - food must be selected.

⑦ Post Condition → Confirm your Order : proceed to checkout

⑥ Proceed to payment.

) Triggers → Confirm.

Main Success Scenarios → Main flow without alternatives

Alternative Paths → Underplace uli hua etc.

Exception → is anything that leads to not achieving the use case goal.

Bright

Date _____

Quality : e.g: Application high performance low latency / downtime
or low user friendly / Interactive System.

Exception se pehle pehle sara case achieve ho jata hai itn exception par
nhi hota ;

Write a narrative Use Case for booking a flight using UACAir app:-

Use Case Name : Book a flight

Primary Actor : Passenger/ customer.

Stakeholders : Passenger.

Secondary Actor : -

Pre Condition : Passenger must be logged-in.

Post Condition :

① Select domestic travel route.

② Ask passenger no. of tickets.

③ On pressing ok, the schedule along confirm button is shown.

④ Passenger chooses the desired flight and enters passenger details.

Alternative flow:

① If it is round trip passenger select arrival and departure date.

② If return date < departure date, then throw error.

Exception : -

Quality ;

Trigges : Select domestic flight.

Revision:-

//allows bidirectional data flow

class Student {

Course A; //Association

Public void AddCourse(Course A) //dependency

{

Course A = new(); //function //Aggregation

}

Student()

{ courseA = obj new Course();

A.addcourse(); //composition.

}

main()

{ Student st = new student();

}

① dependency: Object creation hui hota hoti.

② Aggregation : dependency + obj creation.

③ Association : Object creation nahi hui hain.

④ Composition : Object creation hote hain.

Bright

Revision:-

//allows bidirectional / Date :- P

class Student {

Course A; // Association

Public void AddCourse(Course A) // dependency

{

} Course A = new(); // function // Aggregation

Student()

{ courseA = null; new Course();

A.addcourse(); // composition

}

Main()

{ Student st = new student();

}

① dependency: Object creation hui hoabhi hoti.

② Aggregation : dependency + obj creation.

③ Association : Object creation nahi hoti hai.

④ Composition : Object creation hoti hai.

Bright

ECB (Robustness Analysis)

Date _____

- ① Jacobson's objectory method (Ivar Jacobson)
- ② Entity Control boundary ()
- ③ Robustness (requirement & design gap)
- ④ Jacobson's diagram.

Analysis or design k darmayan gaps hote smooth flow
nhii hota gap ko ishatam karne ki liye Robustness diagram
use karne se class diagram perfections se nhii braskte. us
gap ko fill karne ki liye ham ECB use karte.

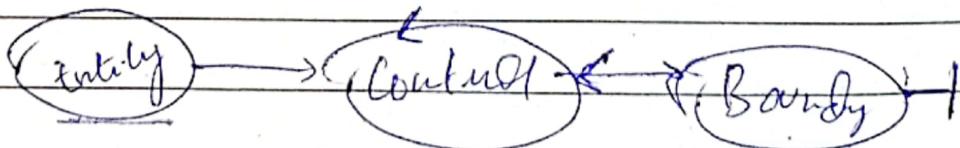
② 3 types of class:

- ① Entities → **entity obj** jo kisi bhi physical objects ko represent karte hain.
- ② Controls → **Boundary obj** object rep. transfer of information.
- ③ And boundaries. → **control obj** System ka interface hoga, Actors ko link/kinteract with system se.

MVC is practical implementation of ECB.

④ Controller

entity or boundary ek door se direct class ke access direct
acess nhii karsakte controller ke through hota.



Bright