Assignment - 02

Kriti Sahni
2301010236
B Tech CSE

Ans 1 : Scenario :

(1) (a) Using LOC

- I/O (25 modules) : 120 LOC → 25 × 120 = 3000 LOC
- User interface (10 UIs) : 200 LOC → 10 × 200 = 2000 LOC
- Internal files (8 files) : 100 LOC → 8 × 100 = 800 LOC

Estimated LOC = 3000 + 2000 + 800
= 5800 LOC

(b) function Point

- External Input (EI) : 4
- External Output (EO) : 5
- External Enquiries (EQ) = 4
- Internal logical files : 7
- External interface files : 5.

→ I/O → 25 × 4 = 100
User → 10 × 5 = 50
External → 8 × 5 = 40

Unadjusted FP = 100 + 50 + 40 + 40 = 190 function Point.

2. Halsted Metrics

$n_1 = 15$ (distinct operators)
$n_2 = 25$ (distinct operands)
$N_1 = 80$ (total operators)
$N_2 = 100$ (total operands)

Vocabulary = $n = n_1 + n_2 = 15 + 25 = 40$
Program length = $N = N_1 + N_2 = 80 + 100 = 180$.
Volume = $V = N \times \log_2(n) = 180 \times \log_2 40$
$= \log_2(40) \approx 5.32$
$V = 180 \times 5.32$
$= 957.6$

Effort = $E = V \times (n_1/2) \times (N_2/n_2)$
$(n_1/2) = 7.5$ $(N_2/n_2) = 100/25 = 4$

$E = 957.6 \times 7.5 \times 4 = 28728.$

Vocab = 40 Programming = 180 Val = 95.8

Effort = 28728.

3. COCOMO (organic)

Effort (PM) = $a \times (KLOC)^b$

Time (TDEV) = $C \times (\text{effort})^d$

$a = 2.4$ $b = 1.05$ $C = 2.5$ $d = 0.38.$

KLOC = 5800 / 1000 = 5.8.

Effort $\approx 2.4 \times 6.15 = 14.76$ PM.

Time = $2.5 \times (14.76)^{0.38}$

$= 14.76^{0.38} = 2.54.$

1PM $\approx$ \$10000 $\longrightarrow$ \$147600

Ans. 2 Design Strategy Analysis.

1. Top Down Vs Bottom Down Approach

→ Top - Down Approach

- Start with system approach as whole.
- Break into subsystem.
- Further refine.

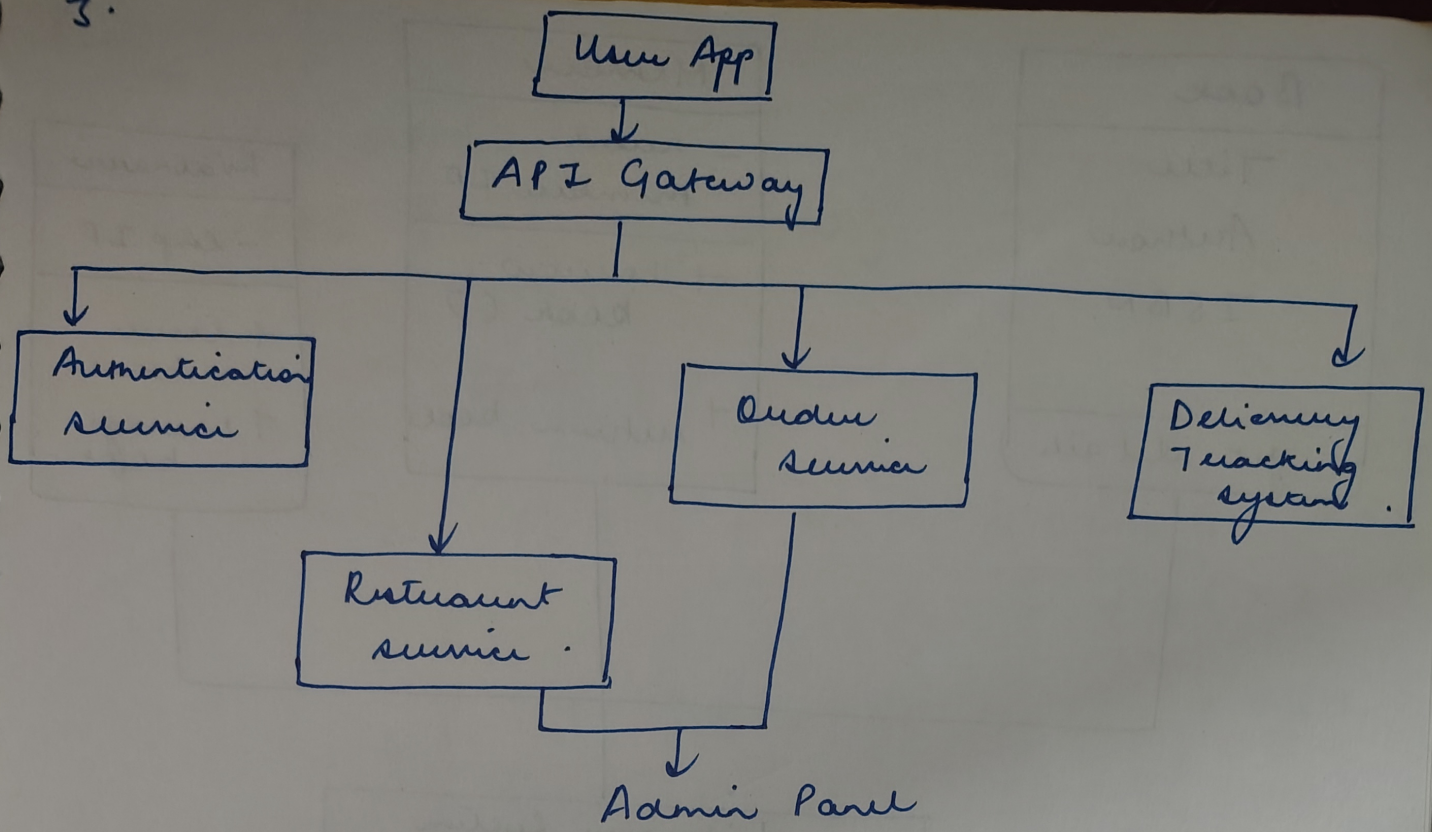
→ Bottom - Up Approach

- Start from small reusable modules.
- Combine into larger components
- Integrate upwards.

2. Design Decision

- cohesion → each module handle a single repository.
- coupling - Keep modules loosed.
- Abstraction - Expose only essential interfaces.

3.



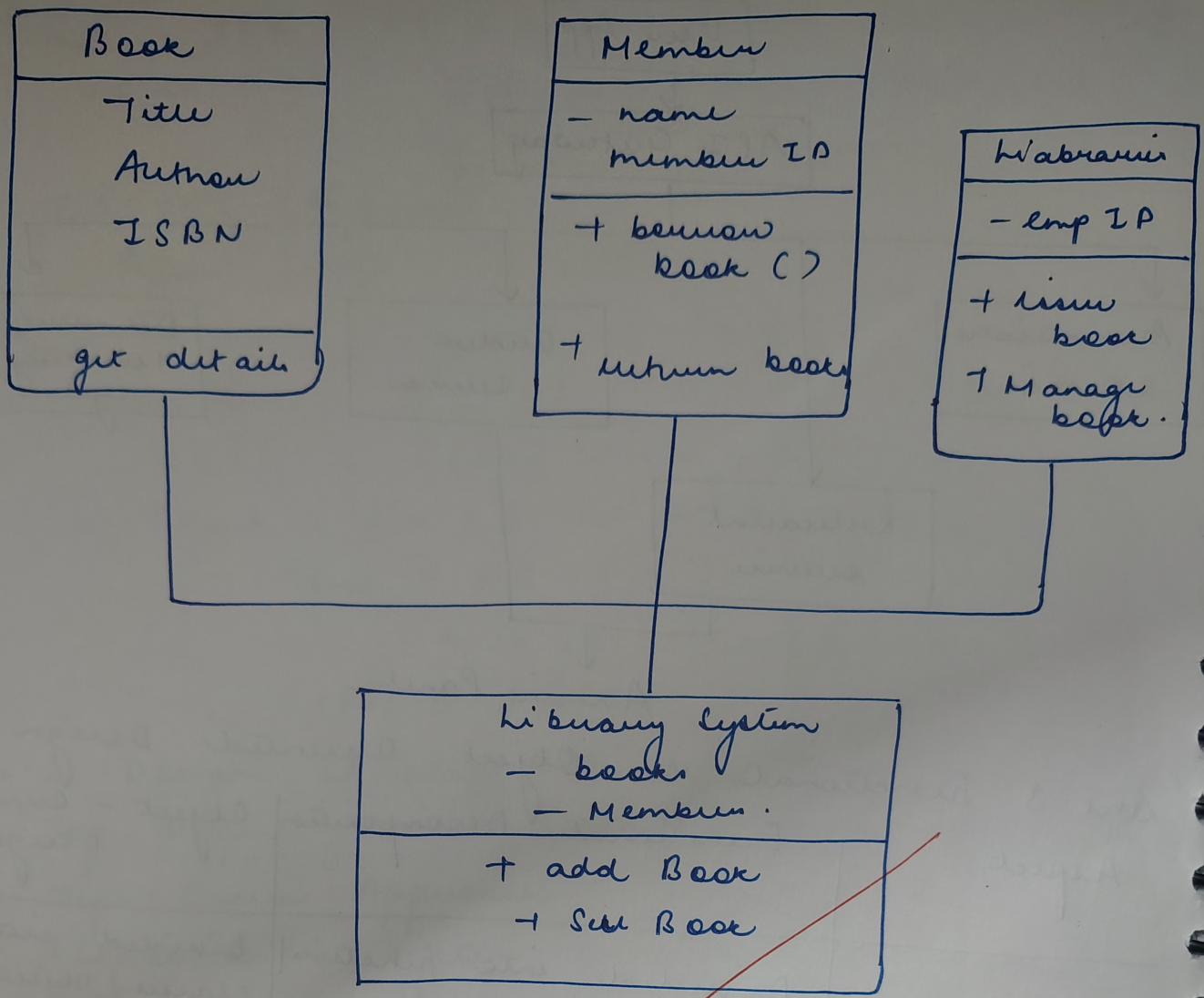
Ans. 3 Functional vs Object Oriented Design.

Aspect	Functional Decomposition	Object - Oriented Design
Structure	Divided into function/ procedure.	Divided into class / object
Data Handling	Data is global / shared passed b/w function	Data encapsulated inside object with attributes & method.
Extensibility	Harder to add new features	Easier.
Reusability	Limited: code is highly coupled	High, supports inheritance.

2. UML class Diagram (OOD Model)

↓

Object Oriented Diagram



3. Best Design for scalability

→ Object oriented Design is better because:

- Supports encapsulation
- Encourages low coupling & high cohesion.
- Provides inheritance and polymorphism
- Better suited for large, evolving system compared to rigid functional decomposition.

Mon 7/10