August

Week	31	32	33	34	35
Monday	1	8	15	22	29
Tuesday	2	9	16	23	30
Wednesday	3	10	17	24	31
Thursday	4	11	18	25	
Friday	5	12	19	26	
Saturday	6	13	20	27	
Sunday	7	14	21	28	

MODULE 3

2016 July (205-161) Saturday 23

8.00
8.30
9.00
9.30
10.00
10.30
11.00
11.30
12.00
12.30
13.00
13.30
14.00
14.30
15.00
15.30

INTERACTION DIAGRAMS -

It is used to describe diff types of interactions among different elements in the model. It is a part of dynamic behavior of the system

there are 2 types of interaction diagrams :

1) SEQUENCE DIAGRAM -
2) COLLABORATION DIAGRAM -

- Purpose of Interaction Diagrams :
  - To capture the dynamic behavior of a system
  - To describe the message flow in the system
  - To describe the structural organization of the objects
  - To describe interaction among objects.

SEQUENCE DIAGRAM - it is an interaction diagram that details how operations are carried out, what messages are sent, and when. They are organised according to time, and time progress and we move down the page. The objects involved in operation are listed from left to right according to when they take part in the message sequence.

SUNDAY 24

COLLABORATION DIAGRAMS - they describe interactions among classes and associations. These interactions are modeled as exchanges of msges b/w classes through their associations. It has foll. components :

- Class roles : which represent roles that object may play within the interaction
- Association roles : which represents roles that links may play within the interaction.
- Message flows - which represents msges sent b/w objects via links.

Meetings | Things To Do | ✓ Important Calls | ✓

Scanned by CamScanner

25 Week 30
July
Monday

SEQUENCE DIAGRAMS	COLLABORATION DIAGRAMS
1) represents the sequence of msgs that are flowing from one object to another	represent the structural organisation of the system & the msgs that are sent & recieved.
2) it is used when time sequence is main focus.	it is used when object organization is main focus.
3) They are better suited of analysis activities.	they are better suited for depicting simpler interactions of the smaller no. of objects.

LOGICAL ARCHITECTURE —

it is large scale organisation of software classes into packages, sub-systems and layers.

Layers is a very coarse-grained grouping of classes, packages, or subsystems that has cohesive responsibility for a major aspect of system.

- Typical layers in an Object Oriented system include:
  - User Interface
  - Application Logic or Domain objects: contains software objects representing domain concepts that fulfill application requirements. eg: calculating total sale
  - Technical Services: consists of general purpose objects and subsystems that provide supporting technical services. eg: interfacing with database.

Morning

✓ Things To Do ✓ Important Calls

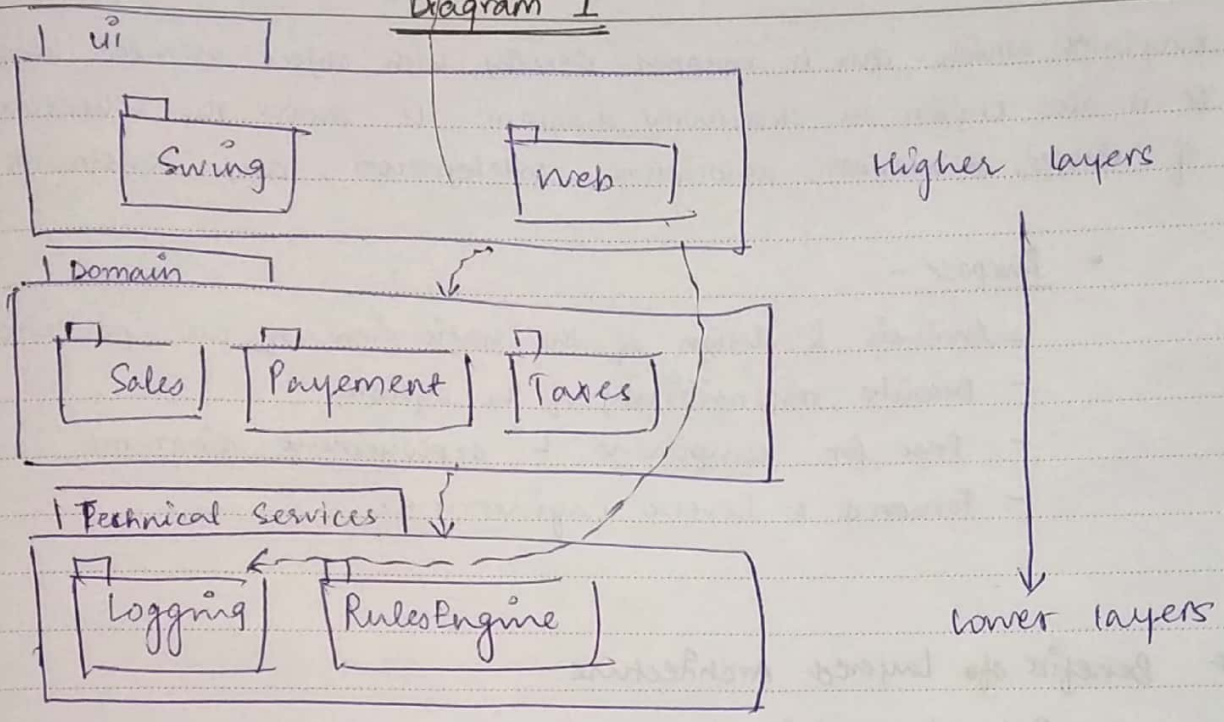
Scanned by CamScanner

August

Week	31	32	33	34	35
Monday	1	8	15	22	29
Tuesday	2	9	16	23	30
Wednesday	3	10	17	24	31
Thursday	4	11	18	25	
Friday	5	12	19	26	
Saturday	6	13	20	27	
Sunday	7	14	21	28	

2016
July
(208-158) Tuesday **26**

Diagram 1



8.30
9.00
9.30
10.00
10.30
11.00
11.30
12.00
12.30
13.00
13.30
14.00
14.30 # **Package Diagrams** –

they are often used to illustrate the logical architecture of the system. A layer can be modeled as UML package. In UML, a package can group anything: classes, other packages, use cases, etc.

15.00
15.30
16.00
16.30
17.00
17.30 eg Diagram 1 shows dependency b/w packages using a dashed arrow line
18.00

Evening # **CLASS DIAGRAMS** –

it describes the attributes and operations of a class and also the constraints imposed on the system. They are widely used in modeling object oriented systems because they are the only UML

Meetings	✓	Things To Do	✓	Important Calls	✓
	☐		☐		☐
	☐		☐		☐
	☐		☐		☐

AUG SEP OCT

Scanned by CamScanner

diagrams which can be mapped directly with object-oriented languages. It is also known as structural diagram. It shows the collection of classes, ~~of~~ interfaces, associations, collaborations, and constraints.

8.00
8.30
9.00
9.30
10.00
10.30
11.00
11.30
12.00
12.30
13.00
13.30
14.00
14.30
15.00
15.30
16.00
16.30
17.00
17.30
18.00

- **Purpose** -
  = Analysis & design of the static view of an application
  - Describe responsibility of the system
  - Base for component & deployement diagrams
  - Forward & Reverse Engineering.

Benefits of Layered Architecture -

- Reduced coupling & dependencies
- Improved cohesion
- Increased clarity
- Related complexities are encapsulated
- Some layers are distributed
- Increased potential for reuse.

Evening

Meetings

✓ Things To Do
☐
☐

✓ Important Calls
☐
☐

Scanned by CamScanner