

2/2022

	7	14	21	28
Monday	-			
Tuesday	1	8	15	22
Wednesday	2	9	16	23
Thursday	3	10	17	24
Friday	4	11	18	25
Saturday	5	12	19	26
Sunday	6	13	20	27

02.week

8/357

EXTRA

① let response = '{ "name": "John", "age": 30, "city": "New York"}';

let parsedResponse = JSON.parse(response);

E

```

  name: "John"
  age: 30,
  city: "New York"
}
```

JSON format

3

- ② • Props are passed as attributes to component and are accessible within that component as an object

~~Props are objects to be passed to a function~~

14

15

16

17

18

19

20

*

→ get user from → user/User hook

based on
Sign-in

onish,
horsh
horsh
Pabel

JANUARY
Saturday

08

it's not
true

- Geogt location Se response aage, Phir uska Main me JSON data me hold kya
and ye same chik Main database me store karwa di

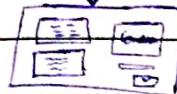
09

JANUARY
Sunday

9/356

[ab mere ID unique ho]

• not try using 'id' as
\$ we have primary ID & db - ID
• get the full JSON output of
that particular ID.



Monday	31	3	10	17	24	1/2022
Tuesday	-	4	11	18	25	
Wednesday	-	5	12	19	26	
Thursday	-	6	13	20	27	
Friday	-	7	14	21	28	
Saturday	1	8	15	22	29	
Sunday	2	9	16	23	30	

to switch to
next page we use
'next' tag as well

(on submit = --)
Create history

► Alternative options for clock Authentication

- 8 • Auth0
- Okta
- Firebase Authentication

► Alternative options for Shaders

- 10 • Material UI, is the famous one but selecting Shaders over Material UI depends on
Project requirement, and Shaders is more customizable and flexible.

► PostgreSQL is a powerful, open source relational database management system (RDBMS)

12 • There are others as - MySQL, SQLite

13 \$ also NoSQL Database

14 • like MongoDB, but I already used NoSQL database
in my StudyNation Project, that is why I use ^{any} Relational DBMS like PostgreSQL.

15 As I used drizzle (ORM) there are alternative methods like TypeORM but
16 according to the project efficiency & convenient, I choose to go with drizzleORM.

17 • also separate of

table is easy

So, I will go

drizzleORM

object relational
mapping?

18

19

20

- If you visit '<https://example.com/about>', a route in the application handles this URL and action the 'About us' page

- Middleware can check if a user is logged in before allowing them to access a protected route. If not logged in, it might redirect them to the login page.

1

JANUARY
Tuesday

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

11/354

30 03. week

Problem 1: So, while handling authentication part, Clark says that :-

8 By default, cloak middleware will not protect any routes. All routes are public and you must opt-in to protection for routes.

⁹ ▷ I handled that two sections separately, → in ~~influence~~ Section

```
const isProtectedRoute = createRouteMatcher((  
  10    ' / dashboard(*)',  
  11    ' / form(*)'  
  12  ));
```

just write whatever you want to
Protect because it is
in an array

we check

the route is Protected or not

\$ export it accordingly and check easily whether the user is signed in or not)

As we create dashboard route in the app folder, and whatever we write ^{it} is protected

And again, for signing it will showing some other domain

- I read some documentation on slack website itself, there I add some files and codes under the folder (auth)  ↳ To having my own domain for sign in

So that, next JS will not consider it as route!!

Environment Variable ..

1. *Leucosia* *strobli* (Herrich-Schäffer) *1850*
2. *Leucosia* *strobli* *var.* *lutea* (Herrich-Schäffer) *1850*

Want more
of him? P
want more
of him? P
want more
of him? P

Digitized by srujanika@gmail.com

In the land of the blind, the one-eyed man is stoned to death. – Joan D. Vinge

	2/2022				
	-	7	14	21	28
Monday	1	8	15	22	-
Tuesday	2	9	16	23	-
Wednesday	3	10	17	24	-
Thursday	4	11	18	25	-
Friday	5	12	19	26	-
Saturday	6	13	20	27	-
Sunday					

2/2022

- add components in app as -components
So, that Next.js will not treat it as Route

JANUARY

Wednesday

12

12/353

03. week

- ▷ Next.js is one of the best react framework for the web application. [It provide the variety of features, the most important is folder based routing]

9) shoder library is one of the popular third css library (Indirectly use button, textarea etc)

¹⁰ ▷ I use cleark authentication , as it is most popular authentication ~~service~~ service .
• Mainly most i use it as it is also free & easy to use .

¹¹ (we are having option to enable whatever the authentication service you want to add)

• (also, we can check all the users who sign-in) ↳ like-beagle/linked/fetchd/microsoft

• For Sign-in Page, I used hyperlink → Just for the URL of 1st Page

• ▷ For logs, I use `log ipsum`

done table ki

(reforming society)

- For Backend of our application, we use PostgreSQL to store all of our data, but for data we need to write the query for an alternative I use drizzle ORM

- The following
- Exact changes

- which stands for object relational mapping and which helps us to insert | update | delete the record from the PostgreSQL without writing the query

- As we waits the Selma and then I make a call then it fetch the data from PostgreSQL & then driver will return the response to the application.

19 ► For braini AI, I use Input prompt according to the user's input and send this to braini AI.

```
const InputPrompt = "Job Position: " + JobPosition + ", Job Description: " + JobDescription + ".\n* const result = await ChatSession.sendMessage(InputPrompt)
```

Concert resultat = await ChatSession.sendMessage (Input Prompt)

6 
text
coffee
face 9

while we're learning how
we can code in extra
about it as well.

front right to open:

Convert into JSON format

A man can stand anything except a succession of ordinary days. — Johann Wolfgang Von Goethe

13

JANUARY
Thursday

Monday	31	10	17	24	1/2022
Tuesday	-	4	11	18	
Wednesday	-	5	12	19	
Thursday	-	6	13	20	
Friday	-	7	14	21	
Saturday	1	8	15	22	
Sunday	2	9	16	23	

13/352

(partial version)

03. week

- 4 **▶ npx create-next-app@latest** • npm run dev
- 5 **▶ for static installation**
- 6 **▶ npx shaden-ui@latest init**
- 7 **▶ automatically add component folder (e.g.- like button / texture) first v via Keenig will add like add hate toggle)**
- 8 **▶ npx shaden-ui@latest add button**
- 9 **▶ npx shaden-ui@latest add display**
- 10 **▶ npx shaden-ui@latest add input**
- 11 **▶ npx shaden-ui@latest add textures**
- 12 **▶ for dark installation**
- 13 **▶ npm install @clerk/nextjs** → here we get Clerk Public & Secret key, which needs to be stored in .env.local file
 - Create middleware.js bypass the code as Clerk says
 - Pop up with < Clerk Providers > ... </providers>
 - all code
- 14 **▶ for database**
- 15 **▶ npm i drizzle-orm @mantineui/experiments -D drizzle-kit**
- 16 **▶ for push**
- 17 **▶ npm run db:push**
- 18 **▶ npm run db:studio**
- 19 **▶ for intelligent search API**
- 20 **▶ npm install uid**
- 21 **▶ for adding toast**
- 22 **▶ npm i moment**
- 23 **▶ for collapse**
- 24 **▶ npm i react-motion**
- 25 **▶ npm i react-hook-speech-to-text**
- 26 **▶ npm i react-hook-speech-to-text**

Flattery, is like cologne water, to be smelt of, not swallowed. — Josh Billings

2/2022

	-	7	14	21	28
Monday	-	1	8	15	22
Tuesday	-	2	9	16	23
Wednesday	-	3	10	17	24
Thursday	-	4	11	18	25
Friday	-	5	12	19	26
Saturday	-	6	13	20	27
Sunday	-				

AI Mock Interviews

~ 7 Page (Enough)

[Nextjs, React, Drizzle, ORM, Gemini AI, Clerk]

JANUARY
Friday

14

03. week

React, Nextjs, Gemini AI

→ Speech Synthesizer → TTS51
 → Natural language processing → NLP51
 → Text-to-speech → TTS51
 → Speech-to-text → STT51

- Authentication Service with the help of Clerk, to Just login with the account & then you will
- 9 Jump on the Dashboard,
- where you have option to add a new mockup Interview and you can also see
- 10 the all the previous mockup interviews.
- ↳ After clicking on the new mockup interview, you need to give couple
- 11 of information like what is the jobRole / Job Position which you want to take
- an interview, & Write the Tech Stack and Years of experience to
- 12 take the interview according to that.

After all this, As we will click on the start interview, my AI model will generate the questions for you

- After this, you will get the new page asking for microphone enabling option
- 14 After that you can start your Interview, after that you actually jump on
- the actual interview page where they will going to ask 5 questions (which is
- 15 completely AI based), also you have the option to listen the actual question

16 Also there will be option to record the answer, based on whatever I record

17 And after the success of recording the answers, with the help of

18 PostgreSQL and ORM, we are going to save user answers in our database

19 After the interview, it will give me the feedback of the interviewer.

20 In feedback, it will give Rating, your answer, correct answer and then the feedback for each question you answered.

Also, on the home page you will be having Previous Mock Interview feedback & option to give the Re-Interview.

3/2022

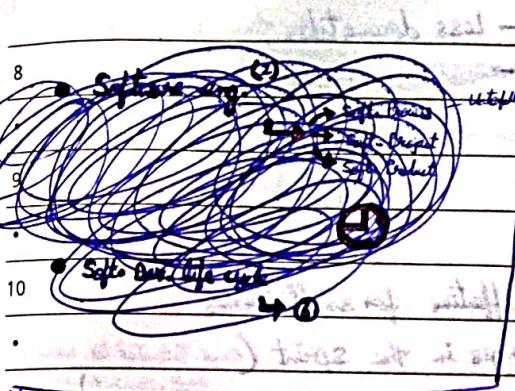
	-	7	14	21	28
Monday	-	1	8	15	22
Tuesday	1	8	15	22	29
Wednesday	2	9	16	23	30
Thursday	3	10	17	24	31
Friday	4	11	18	25	-
Saturday	5	12	19	26	-
Sunday	6	13	20	27	-

07. week

39/326

- Reg. - Understanding b/w customer & supplier
- Specification - what software must do

08

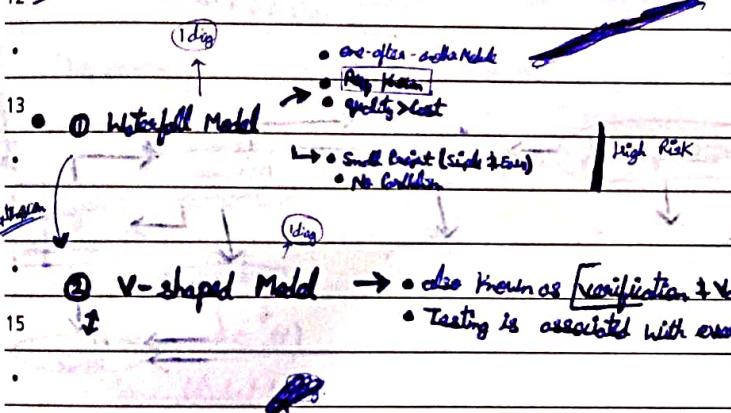
FEBRUARY
Tuesday

Waterfall Model - suitable for large system, stable requirements

Validation → Agile Model

Code & fix

- informal product idea
- work in random order
- No Plan



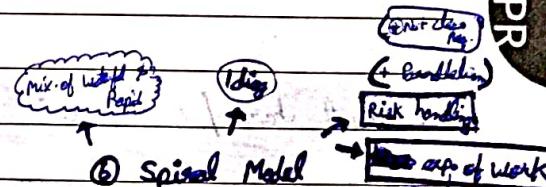
High Risk

Learn late

High Risk

- adv → Small Projects (mini-Projects)
- adv → anyone can use it / low expertise / individual idea to support
- dis → No flexibility / No visibility / Not possible for large product

③ Prototype Model → • customer not clear with Idea
• Throw away model



④ Incremental Model

1. Objective determination & identity set
2. Identify & resolve Risk
3. Develop next version product
4. Review & Plan for next phase

④ Incremental Model → • module by module working
• Early Release Product Demand
• easy to test & debug

⑤ Evolutionary Model → Iterative + Incremental
• feedback loop
• short release cycle

LARGE
changes

- custom. req. ✓
- Risk analysis ✓
- It supports changing environment
- Large Product

⑦ Rapid application development (RAD)

- Quick Prototyping
- Parallel development

Time and Cost
constraint

Rapid Prototyping

Life must be measured by thought and action, not by time. — John Lubbock, Conductor



07

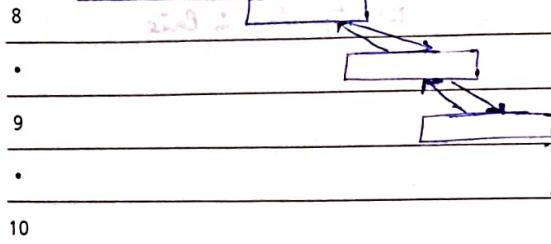
FEBRUARY
Monday

38/327

	2/2022		
	Monday	Tuesday	Wednesday
Monday	-	7	14
Tuesday	1	8	15
Wednesday	2	9	16
Thursday	3	10	17
Friday	4	11	18
Saturday	5	12	19
Sunday	6	13	20

07. week

① Waterfall :-



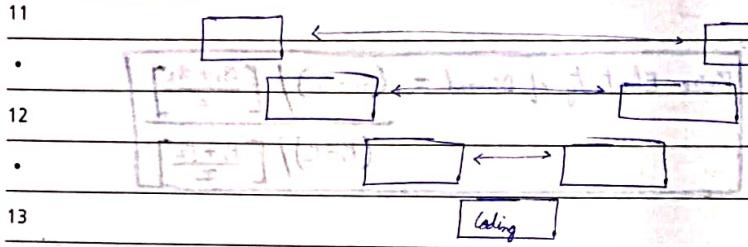
Adv:-

- ① Easy to understand
- ② Simple
- ③ Easy to use

Dig:-

- ① Must have all req.
- ② Integration issue big long at the end.
- ③ clients only can see only after the light is steady (No live-view)

② V-Shaped :-



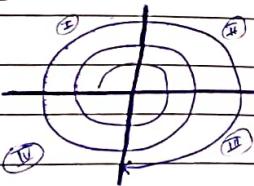
Adv:-

- ① Easy to use
- ② No bubble of testing all at last

Dig:-

- ① Not certain Risk analysis only
- ② Not certain concurrent events
- ③ Not really easy to handle dynamic changes in requirements

③ Spiral :-



adv → • slowly add increment functionality

dig → • Reg. good Planning + Design

Today is your day! Your mountain is waiting. So get on your way. — Dr. Seuss

12

13

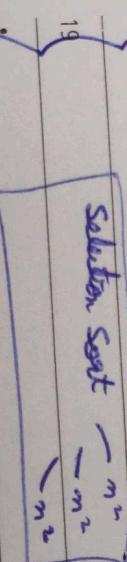
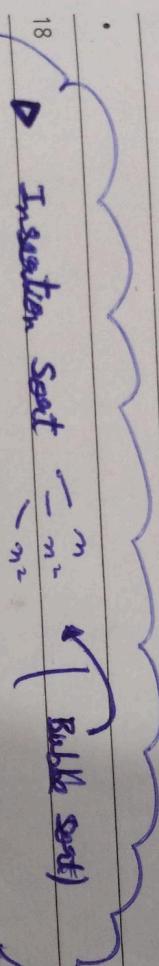
14

15

16

17

- $T.C \rightarrow O(1) > O(n \log n) > O(n) > O(n \log n) > O(n^2) > O(n^3) > O(2^n) > O(n!)$



20 ▶ merge sort $\sim n \log n$

21 ▶ quick sort $\sim n \log n$
 $\sim n^2$

Success is going from failure to failure without a loss of enthusiasm. — Winston Churchill

APR

MAY

JUN

• T.C \rightarrow $O(1) > O(n \log n) > O(n) > O(n \log n) > O(n^2) > O(n^3) > O(2^n) > O(n!)$

► Insertion Sort = n^2 | Bubble sort)

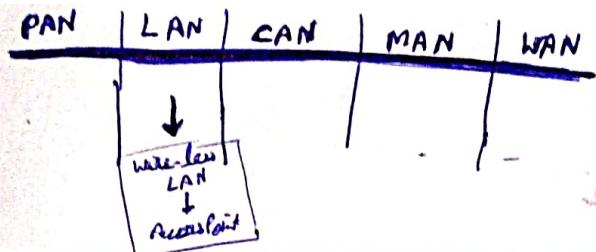
Bubble Sort

Selection Sort $\sim n^2$

► Merge sort $\Theta(n \log n)$

► Quick Sort $\xrightarrow{n \log n}$
 $\xrightarrow{n \log n}$

Success is going from failure to failure without a loss of enthusiasm. – Winston Churchill



MARCH
Wednesday

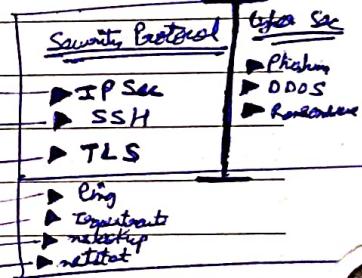
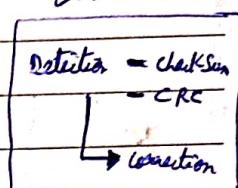
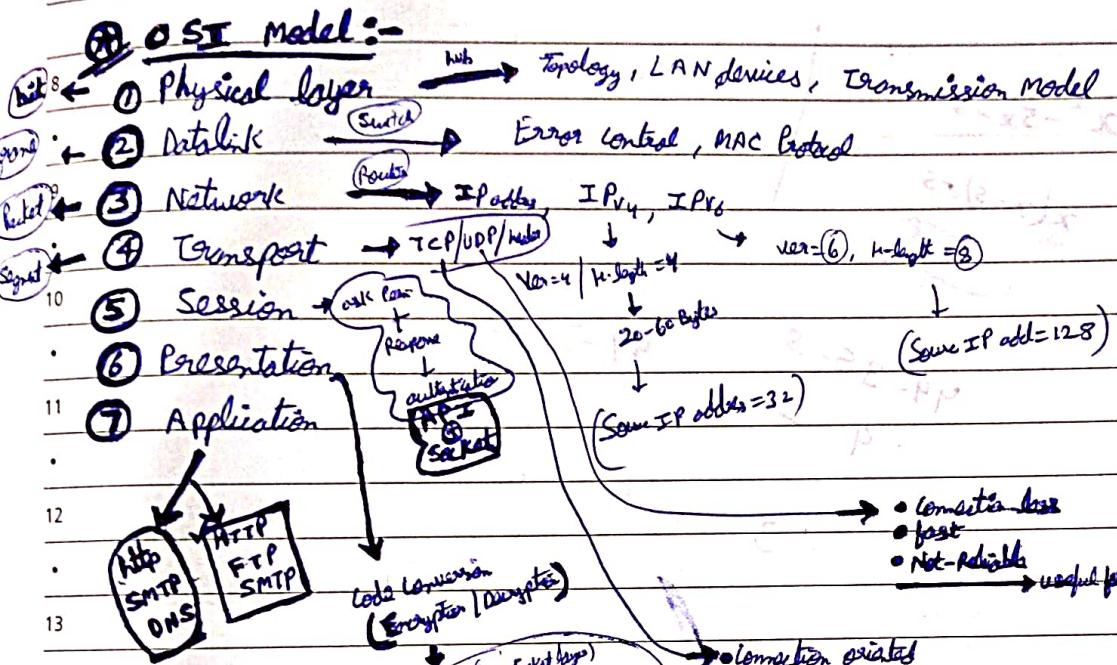
10

12. week

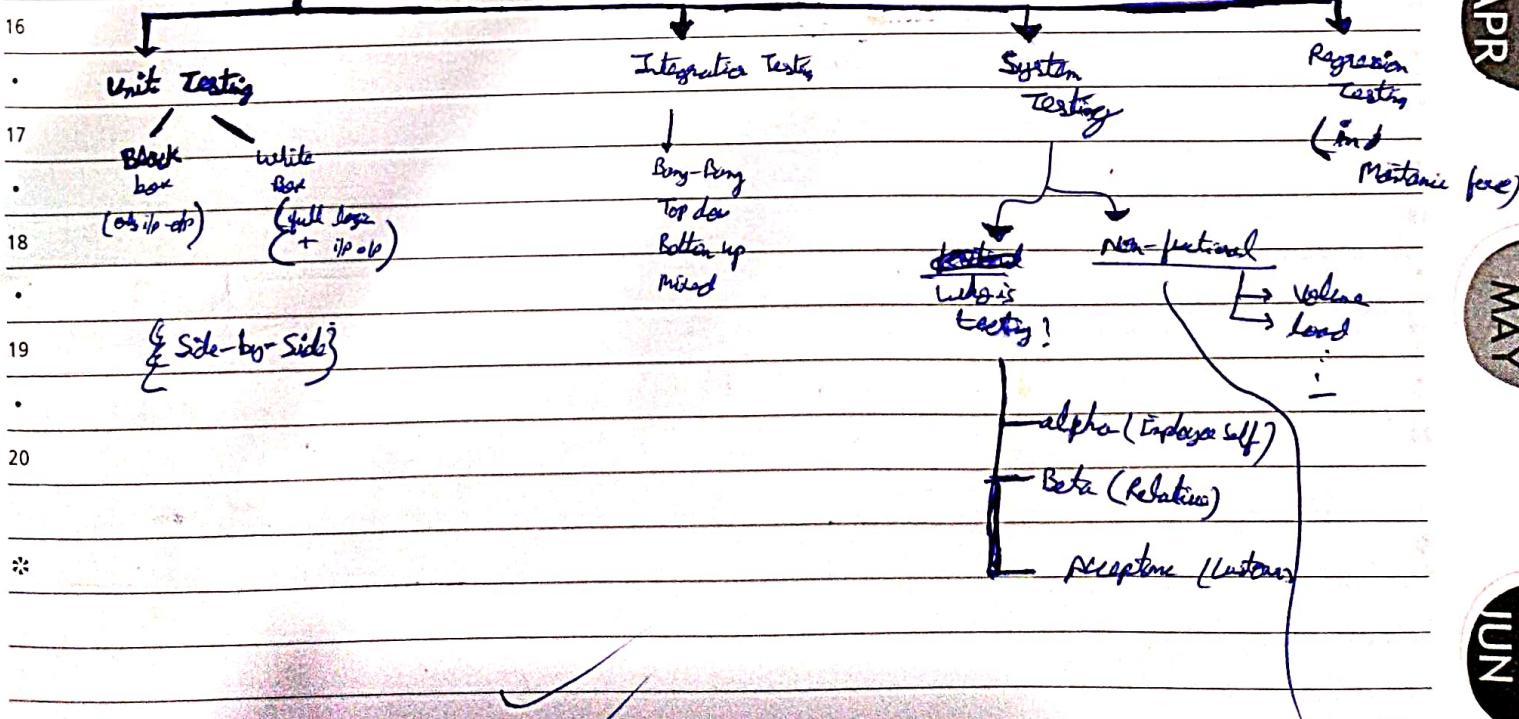
75/290

(DNS - UDP)

HTTP/S - TCP

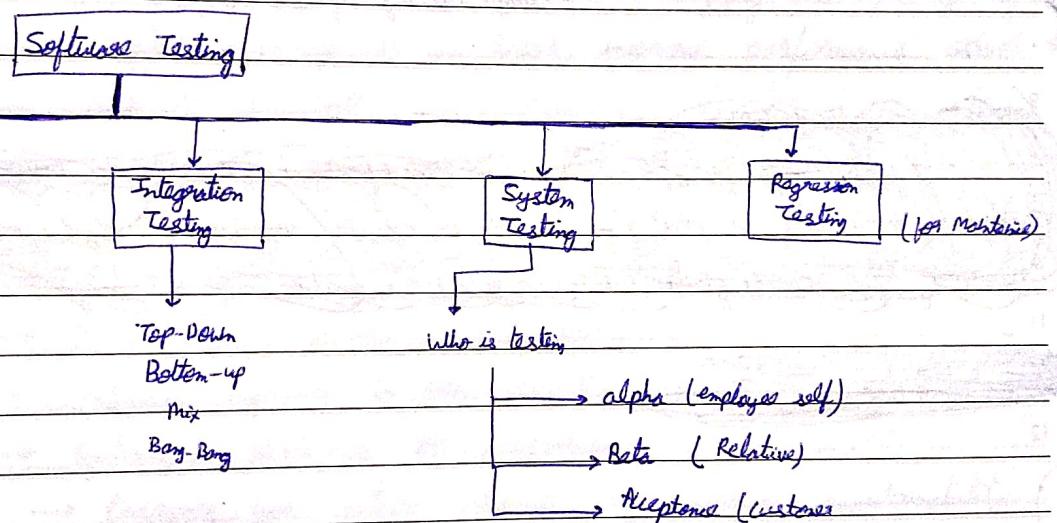


Software Testing



If we do not find anything very pleasant, at least we shall find something new. – Voltaire

JUN



11/2022

	7	14	21	28
Monday	-	8	15	22
Tuesday	1	9	16	23
Wednesday	2	10	17	24
Thursday	3	11	18	-
Friday	4	12	19	-
Saturday	5	13	20	27
Sunday	6	-	-	-

OCTOBER
Thursday

06

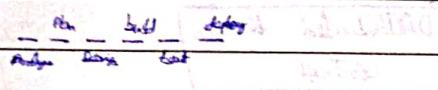
279/86

41.week

- organizations were used
- Software methodology

Waterfall Model

Iterative → one-after-one stage



disadvantages - take lot of time to change designs. Plus if we find [if it is market something in the testing phase then it is hard to do the same code again]

Starting & check at which Phase the error occurs

"Not good for dynamic changes by client.
As maybe client says some other"

Agile Methodology

continuously Review

design, code, build, test



for multiple iteration

iteration

but the operation team
as dev team agile/change early/quickly
There is some problem in operation, it does not work in production

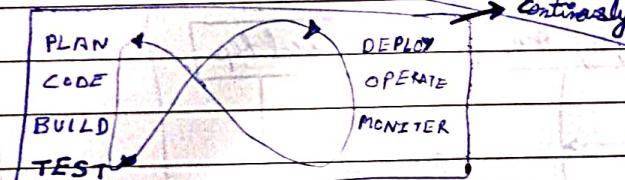
Dev team

Operation

(Conflict occurs)

- Devops is a Software dev strategy which bridges the gap b/w the Dev side and opx side of the company.

it's a related methodology



application used as

like git/JIRA

Stages of Devops :-

- ▷ Version control → maintaining different versions of code
- ▷ Continuous Integration eg Jenkins → which commit causes the error/time, so I can go back
- ▷ Continuous Delivery → e.g. Selenium (for End user testing) or JUnit
- ▷ Continuous Deployment → Not a good practice

(Config management + automation)

as we want some more testing,
so we don't want this as automated

Do it manually
using docker on Puppet tools

The distance is nothing; it's only the first step that is difficult. — Marquise du Deffand

- ▷ Once it is deployed, it is monitored using Nagios or Splunk.

Cloud failback & team.

Puppet works on master-slave architecture

UJ

OCTOBER
Wednesday

27/8/7

Monday	10	11
Tuesday	11	12
Wednesday	12	13
Thursday	13	14
Friday	14	15
Saturday	15	16
Sunday	16	17

41 weeks

8

9

10

Config - Management
Tool



11

12

13

14

15

16

Software Testing

18

19

20

With that UI
you can check your
childhood books.

Jenkins gives
online 360° view
of Software life cycle

With auto code,
it will build
complete
and testing
Iteration testing

Nature gave us one tongue and two ears so we could hear twice as much as we speak. — Epictetus

Which Phase it is right now,

Testing is done or not?

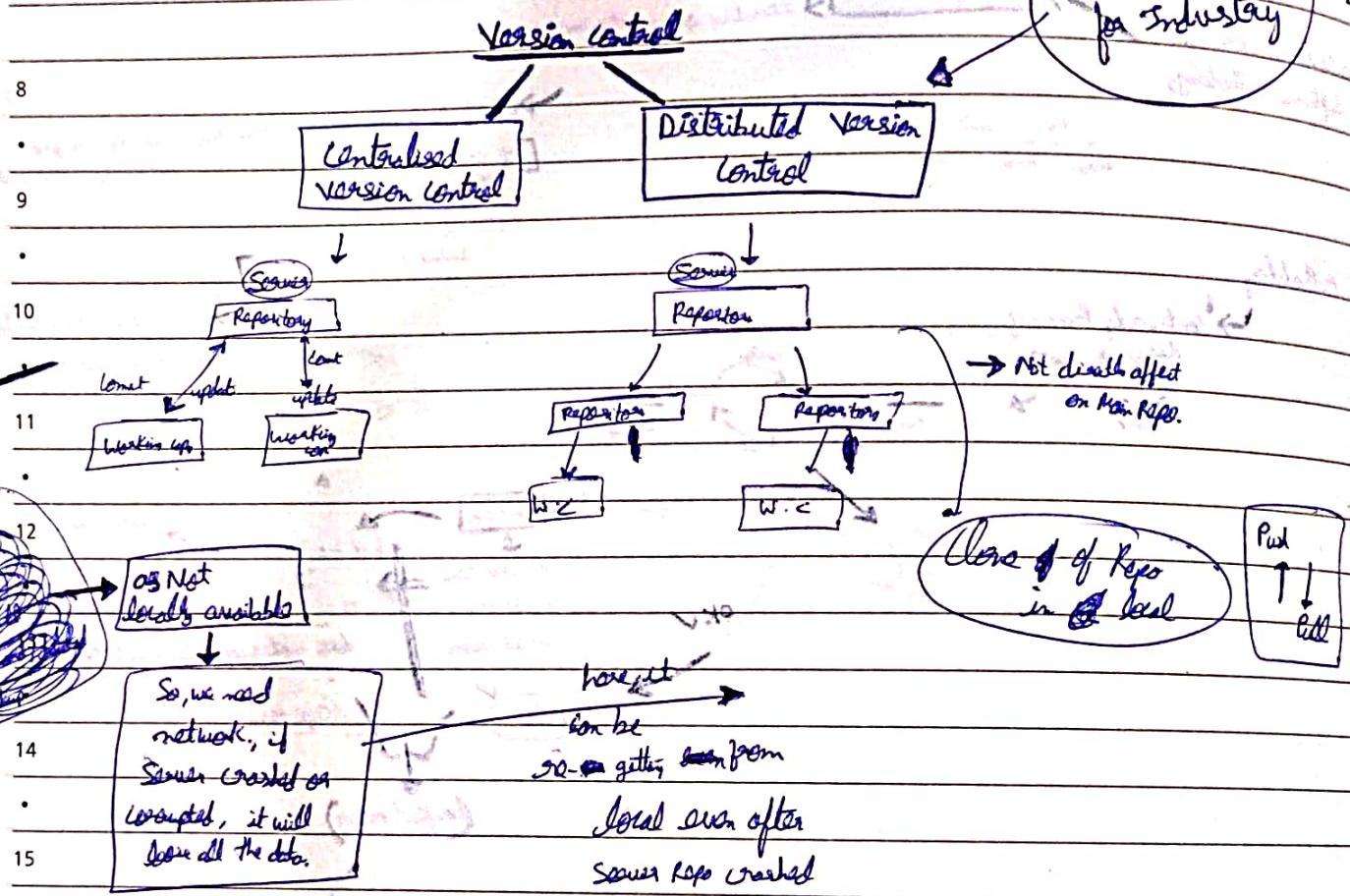
07

OCTOBER
Friday

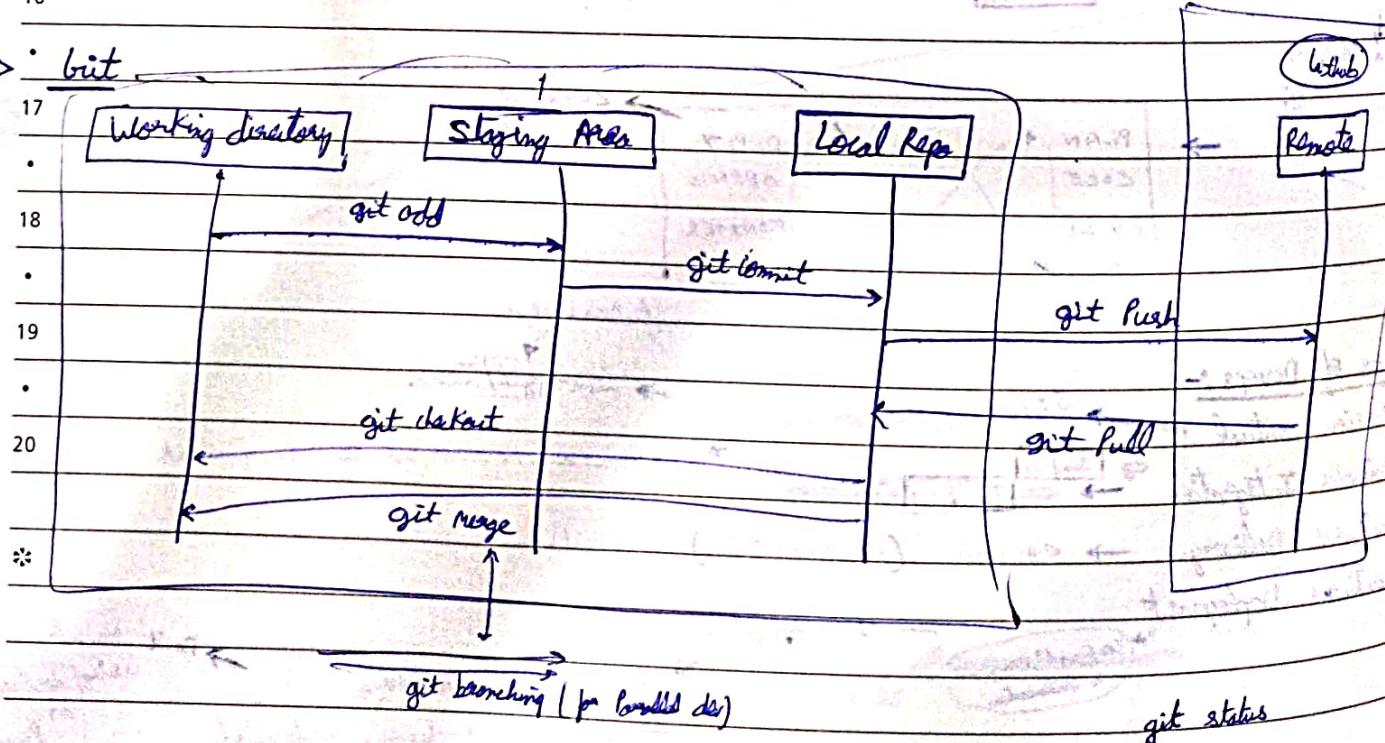
280/85

Monday	31	10	11
Tuesday	-	4	11
Wednesday	-	5	12
Thursday	-	6	13
Friday	-	7	14
Saturday	1	8	15
Sunday	2	9	16

Better
for Industry



git



► Devops pipeline & automated workflow → software development so better deployment take less time

	11/2022	12/2022	1/2023	2/2023
Monday	-	7	14	21
Tuesday	1	8	15	22
Wednesday	2	9	16	23
Thursday	3	10	17	24
Friday	4	11	18	25
Saturday	5	12	19	26
Sunday	6	13	20	27

OCTOBER
Saturday

08

41. week

281/84

- 'Containers are lightweight alternatives to Virtual Machine'

→ containerization tool, to isolate
→ Docker are OS-independent from
host OS

- Kubernetes, a powerful container orchestration tool,

↳ It automatically scales containers up & down based on demand

↳ It handles load balancing

* like AWS

like game Pokemon used Kubernetes to handle millions of real-time players.

- With Git, Docker, Jenkins, Kubernetes & Nagios, I helped UST to build a robust automated, and efficient Devops pipeline. Deployment times shrank from days to minutes, downtime reduced significantly, and the team focused more on innovation than fixing broken code.

Docker Image → blueprint
for
Container

Docker → self gradable → Docker Swarm

but Kubernetes More popular

Container
run
isolate

► Docker Workflow:-

- ① Dockerfile base
- ② Docker Image base
- ③ Container run
- ④ Container stop
- ⑤ Docker hub / private no store base

- Docker build
- Docker run
- Docker Stop

VM → whole OS.
Docker → only application

— Docker Push

You'll never achieve real success unless you like what you're doing. - Dale Carnegie

NOV

DEC

- In Design Round, there is no right or wrong answer, It is discussion Round!!
- You just have to explain the diagram & take feedback for add/remove acc. to Requirement

01

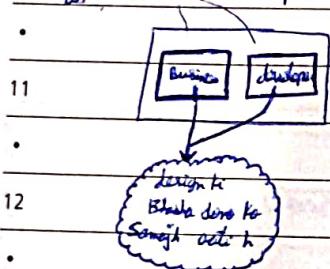
NOVEMBER
Tuesday

#1

305/60

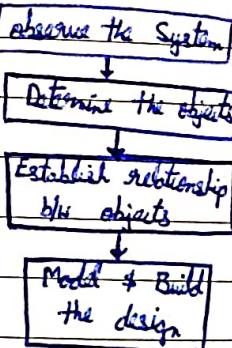
- LLD → In LLD, the focus is more on designing each component in detail such as what classes are needed, what abstractions to use, how object creation shall happen, how data flows between different objects.

- To make create Karne ke liye, the only first thing we need is **Requirement Specification**
- ab se deno ke agar main banta hoga to design / flowchart ka form me kta hoga, aur developer kya apna according data design ki → cir (Requirement) ab main iste according code kardunga



- kal ka Main toon chha ke chha gya, to koi participant karta hoga to requirement find out karta hoga
Requirement find out karta hoga
Participants ke bich na relationship find out karta hoga aur last me Task find karta hoga, phir uska implement karta hoga.

► Object oriented Analysis & Design (OOAD)

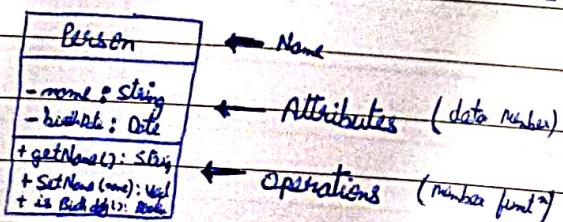


► UML (Unified Modeling Language)

jab bhi UML ki boat karne, tumhare dinay me visualization conchaliye caligt me ye behut sare tools provide karte h, Jis tools se ek system ko visualize kar pa sakte ho, ki kaise use ne diff-diff entities represent karte h

- UML is composed of three main building blocks : things, relationships, and diagrams

① Class



12/2022

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

NOVEMBER
Wednesday

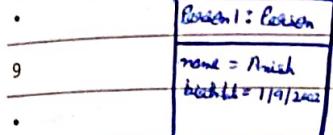
02

45.week

306/59

(2) object

8



10

(3) Interface

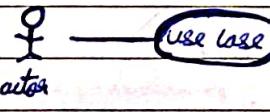
Show as



Student Application ← Name

(4) Use Case

→ Isse 'Car' ka use case kya h? → Likhna, Ph. fight Karo, Kisi Ki Khoon Karna



(5) Actor

→ Isse I Banking System me I driver,
Yu Parki Karo Wala → actor

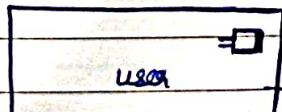
→ Collekti which me Primary actor → main bank

Secondary actor → To TA (debt solving)

To log in

Primary actors kisi karta h

(6) Component



ab user ke ander bh

baat Sare task ho sakte h

(7) Node

Physical aspect of a system
(hardware & component)

None

An undefined problem has an infinite number of solutions. — Robert A. Humphrey

DEC

03

NOVEMBER
Thursday

307/58

Monday	7	14	21	11/28
Tuesday	8	15	22	
Wednesday	9	16	23	
Thursday	10	17	24	
Friday	11	18	25	
Saturday	12	19	26	
Sunday	13	20	27	

⑧ Activity diagram

→ Such activities
flowchart h

8

⑨ Interaction diagram

9

10

11

12

13

14

15

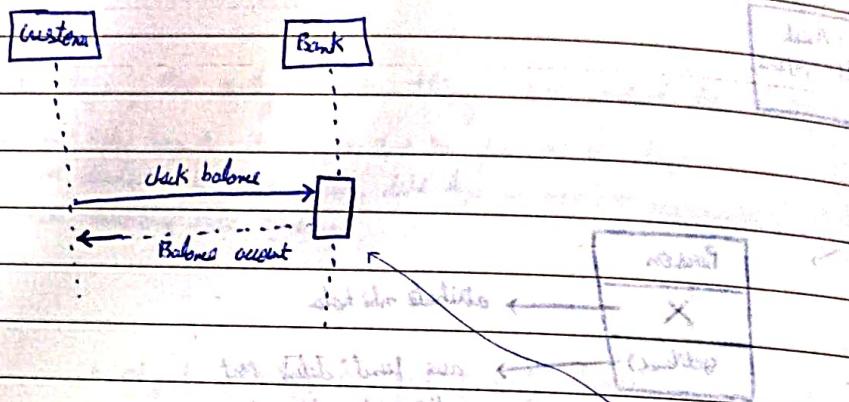
16

17

18

19

20



▷ Types of UML

→ structural and behavioural diagram

Static structure

(To pic h, with shapes like and tk)
(see class - data member fix kardig)

dynamic behavior

(like move method X)

• Most commonly used :-

① class diagram

② use-case diagram

③ Sequence diagram

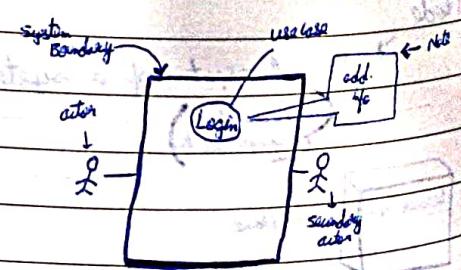
④ Activity diagram

⑤ Use-case diagram

→ It describes the specifications of users and their possible interactions with the system. These possible interactions are called use cases.

Components of use-case :-

- ▷ Actor : They interact with system
- Primary : responsible for initiating use case
- Secondary : responsible for assisting the primary actors
- ▷ use case : a single function
- ▷ Package : group of different elements
- ▷ Note : used to add additional information



Friendship make prosperity more shining and lessens adversity by dividing and sharing it. – Cicero

12/2022

	5	12	19	26
Monday	-	6	13	20
Tuesday	-	7	14	21
Wednesday	1	8	15	22
Thursday	2	9	16	23
Friday	3	10	17	24
Saturday	4	11	18	25
Sunday				-

45.week

► Relationships in Use-case diagram

- association
- generalization
- Include
- Extend

(Interactions)

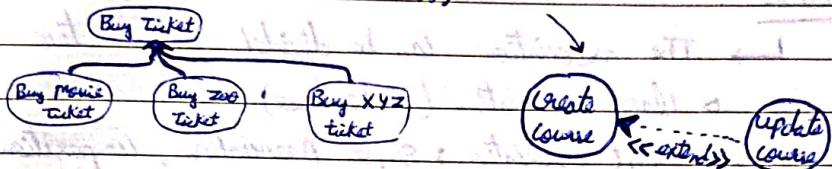
→ *We have to login*
We have some team

→ *add on ticket*
auto login (necessary which is auto login)

NOVEMBER
Friday

04

308/57

*Transfer funds**<< include >>**Sufficient funds*

10. Benefits of Use-case diagrams :-

- It shows what each part of the system does and what it aims to achieve.
- It outlines what the system includes and its limits.

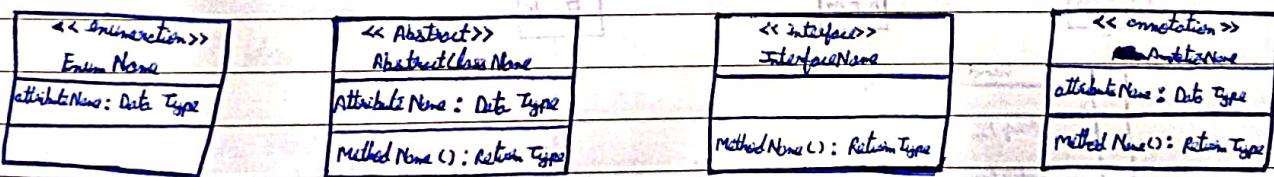
12. Class Diagram :-

13. Popular notations in the class diagram

14. Class Notation

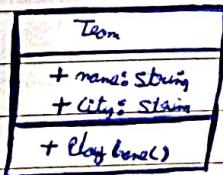
15. Inheritance, abstract class, and enumeration

16. Access modifiers



19. Symbol :-

- + → Public
- → Private
- # → Protected



Worry does not empty tomorrow of its sorrow; it empties today of its strength. – Corrie Ten Boom

DEC

05

NOVEMBER
Saturday

309/56

Monday	7	14	21	28	11/2022
Tuesday	8	15	22	29	
Wednesday	9	16	23	30	
Thursday	10	17	24	31	
Friday	11	18	25		
Saturday	12	19	26		
Sunday	13	20	27		

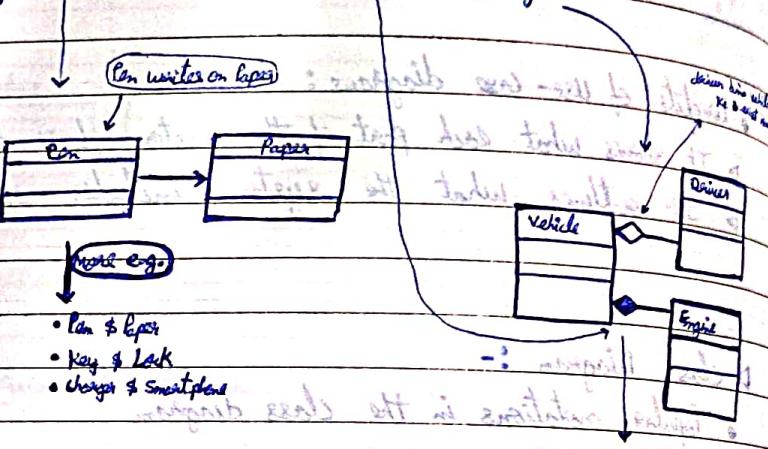
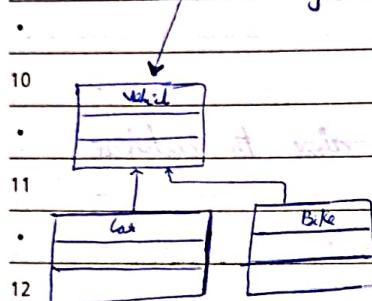
45. week

• Association

→ The association can be divided into two categories :-

▷ Class association (Inheritance)

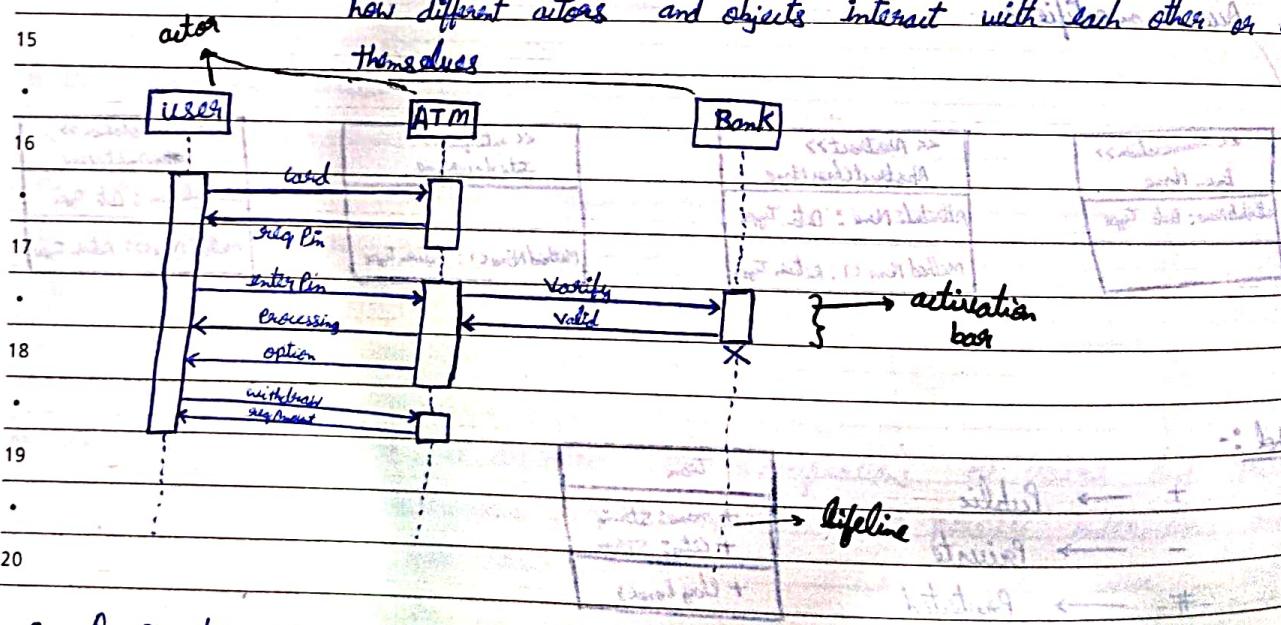
▷ object association : Simple Association, composition & Aggregation.



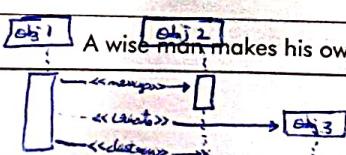
- Pen & Paper
- Key & Lock
- charger & smartphone

#2

▷ Sequence Diagram - It is a form of communication diagram that illustrates how different actors and objects interact with each other or by themselves.



- A synchronous message is when the sender sends a message and has to wait until it gets a reply before it can do anything else.
- An asynchronous message is when the sender sends a message but doesn't have to wait for an answer back.



A wise man makes his own decisions, an ignorant man follows the public opinion. - Chinese proverb

(create
destroy
message
function
always update to succeed)
note 1

lost message
found message

12/2022

	5	12	19	26
Monday	-	6	13	20
Tuesday	-	7	14	21
Wednesday	-	8	15	22
Thursday	1	9	16	23
Friday	2	10	17	24
Saturday	3	11	18	25
Sunday	4	-	-	-

NOVEMBER
Sunday

06

310/55

45.week

► How to draw a Sequence Diagram?

- 8 • Identify the use-case
- Identify the actors and objects
- 9 • Identify the order of actions
- Create the diagram

10 • [Notable uses Kuch sentences, in Sequence Diagram]

11 • Fragment frame:

- ↳ alt — (4-alternatives)
- 12 • opt — (4)
- loop
- ref

13 • [Notable uses Kuch sentences, in Sequence Diagram]

14 • Activity Diagram

15 • [Notable uses Kuch sentences, in Sequence Diagram]

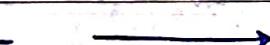
Start Node —



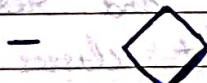
Action state —



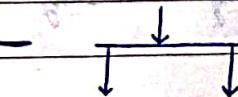
Control flow —



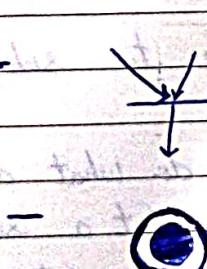
Decision Node —



Join —



End state —



► Steps to draw activity diagram?

- Determine the actions of the system
- Finding the flow of each activity
- Create the diagram

Good code: Quality code :-

- ① Readable code
- ② Extensible
- ③ flexible
- ④ Stable → exception handling
- ⑤ Reusability
- ⑥ Security → (less no less abstraction use less logic)
- ⑦ low overheads

SUNDAY		MONDAY		TUESDAY		WEDNESDAY		THURSDAY		FRIDAY		SATURDAY	
1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28
29	30	31	1	2	3	4	5	6	7	8	9	10	11
11/25/22		46. Week											

07

NOVEMBER
Monday

#3 311/54

SOLID principles

8

→ Purpose :-

- To make the code easier to read and understand, thus spend less time figuring out what it does and more time actually developing the solution.

(we class has ext
team test)
to

① Single Responsibility Principle (SRP)

- A class should have one, and only one reason to change. This means that a class should only have one job or responsibility.
- A class should only be responsible for one thing.
- Following SRP makes your code more modular, easier to understand, maintain, and extend. It helps in isolating functionalities, making debugging and testing more straightforward.

② Open-Closed Principle (OCP)

- An entity should be open for extension but closed for modification. This means you should be able to add new functionality without changing the existing code.
- Extend functionality by adding new code instead of changing existing code.
- OCP encourages a more stable and resilient codebase. It promotes the use of interfaces and abstract classes to allow for behaviors to be extended without modifying existing code.

③ Liskov Substitution Principle (LSP)

- Any derived class should be able to substitute its parent class without the consumer knowing it.
- Child class should be able to do what a parent class can do.
- The goal of LSP is to ensure that a subclass can stand in for its superclass. This principle helps in maintaining the correctness of the program when objects of a superclass are replaced with objects of a subclass.

12/2022

	5	12	19	26
Monday	-	6	13	20
Tuesday	-	7	14	21
Wednesday	-	8	15	22
Thursday	1	9	16	23
Friday	2	10	17	24
Saturday	3	11	18	25
Sunday	4	12	19	-

46.week

NOVEMBER
Tuesday

08

312/53

④ Interface Segregation Principle

- The interface segregation principle (ISP) is a design principle that does not recommend having methods that an interface would not use and require.
- To comply with the interface segregation principle (ISP), it's important to design interfaces that are tailored to specific client needs instead of creating broad, all-purpose interfaces.
- Do not build one big interface make smaller and specific ones.

LSP → ask you to make sure all child class have some behaviour as parent class.

ISP → create a diff interface for diff responsibility, don't group unrelated behaviour in one interface.

⑤ Dependency Inversion principle

- Never depend on everything concrete, only depend on abstraction.
- High level module should not depend on low level module. They should depend on abstraction.
- Able to change an implementation easily without altering the high level code.
- By adhering to DIP, you can create systems that are resilient to change, as modifications to concrete implementations do not affect high-level modules.

(Ex: Purple School Inst. to Notebooks depend on them to note book/bulk Note v chit Jangla Web attach.)

pm (Base Base) E
3

Lab Base Class
in Sparrow
says
Kathy to
Chit Jangla

DEC

09

NOVEMBER
Wednesday

long of four

313/52

GOF

► Categorisation :-

- 8 • **creational** - object creation like creation & how which
- **Structural** - here classes which are structured &
- **Behavioral**

9 **CREATIONAL**

① Factory Method Pattern

10

factory of factory

object factory

method

↳ It provides an **Interface** for object creation but delegating the actual information of objects to **subclasses**.

② Abstract Factory

11

↳ It solves the problem of creating family of related products.

12

13

↳ defining an interface to create families of dependent objects

14 without concrete classes

15

16

17

18

19

20

hum ke
to user ke bta

donge ke seat use hoga
but kyun sa seat use hoga
use ke belongs

• abstract
• user side

abstract layer



► Builder Pattern

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

12/2022

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

NOVEMBER
Thursday

10

314/51

46.week

Singleton

→ It enforces → Main kisi class ka single hi object create hone dunga

#7 Prototype Pattern

→ It focuses on new object

→ by copying → existing object

→ It focuses on creating new object

→ from existing object using prototype concept

→

#8

Structural Design Pattern:

→ Take class having new functionality add it to system without modifying the entire system.

→ focuses on adding a new functionality without any need to modify entire system.

14 Decorator Pattern

→ It allows a user to add new functionality to an existing object without altering its structure.

15

Adapter Pattern

→ Adapter is a structural design pattern that allows incompatible interface b/w classes to work together without modifying their source code.

#9

Proxy Pattern

→ authority to represent someone else

19

Facade Pattern

→ Facade is a structural design pattern that provides a simplified interface to a library, a framework, or any other complex set of classes.

20 (Notes)
order se hoga
aur
kahan se smile

The outward behaviour of a man is at once the sign and proof of the inner state. – Gandhi

DEC

11

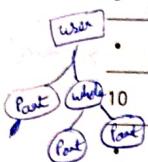
NOVEMBER
Friday

315/50

#10

Bridge Pattern

- 8 ↳ The bridge design Pattern allows you to separate the abstraction from implementation. It is structural design pattern.
- 9



Composite Pattern

- 10 ↳ made up of various elements
- 11 ↳ The composite pattern is partitioning design pattern. It describes a group of objects that are treated the same way as a single instance of the same type of object
- 12

Flyweight Pattern

- 13 ↳ The flyweight software design pattern refers to an object that minimizes memory usage by sharing some of its data with other similar objects
- 14

Behavioral

#11

► chain of responsibility

- 15 ► command pattern

16

#12

► Mediator Pattern

- 17 ► observer pattern

18

#13

► Iterator Pattern

- 19 ► visitor pattern

20

*

To be what we are and to become what we are capable of becoming, is the only end of life. – Baruch Spinoza

Monday	7	14	21	28	11/25/22
Tuesday	8	15	22	29	
Wednesday	9	16	23	30	
Thursday	10	17	24	31	
Friday	11	18	25		
Saturday	12	19	26		
Sunday	13	20	27		

46. Week

12/2022

	5	12	19	26
Monday	-	6	13	20
Tuesday	-	7	14	21
Wednesday	-	8	15	22
Thursday	1	9	16	23
Friday	2	10	17	24
Saturday	3	11	18	25
Sunday	4			-

NOVEMBER
Saturday

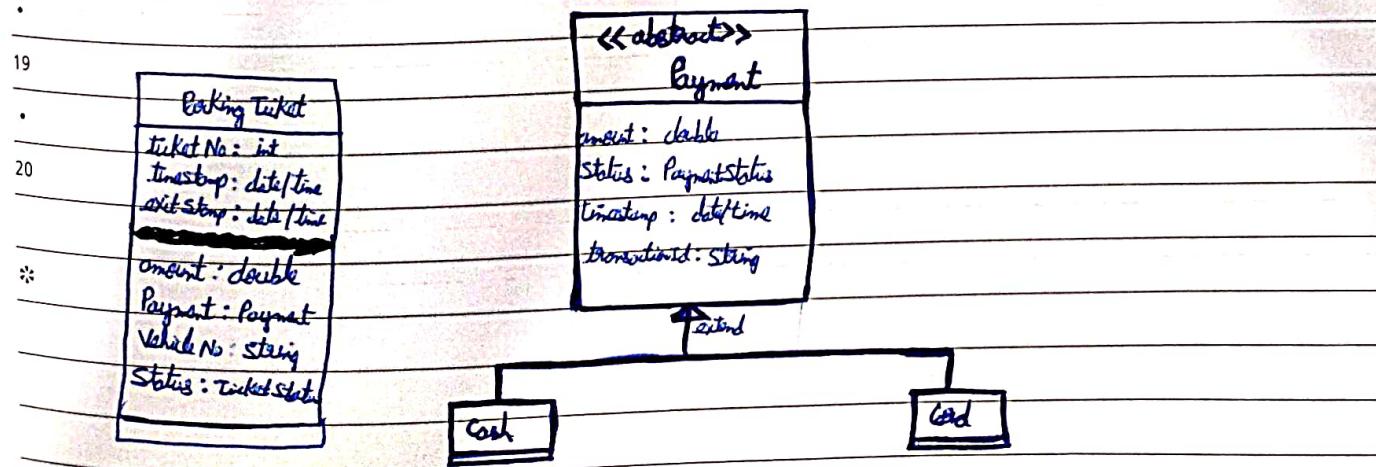
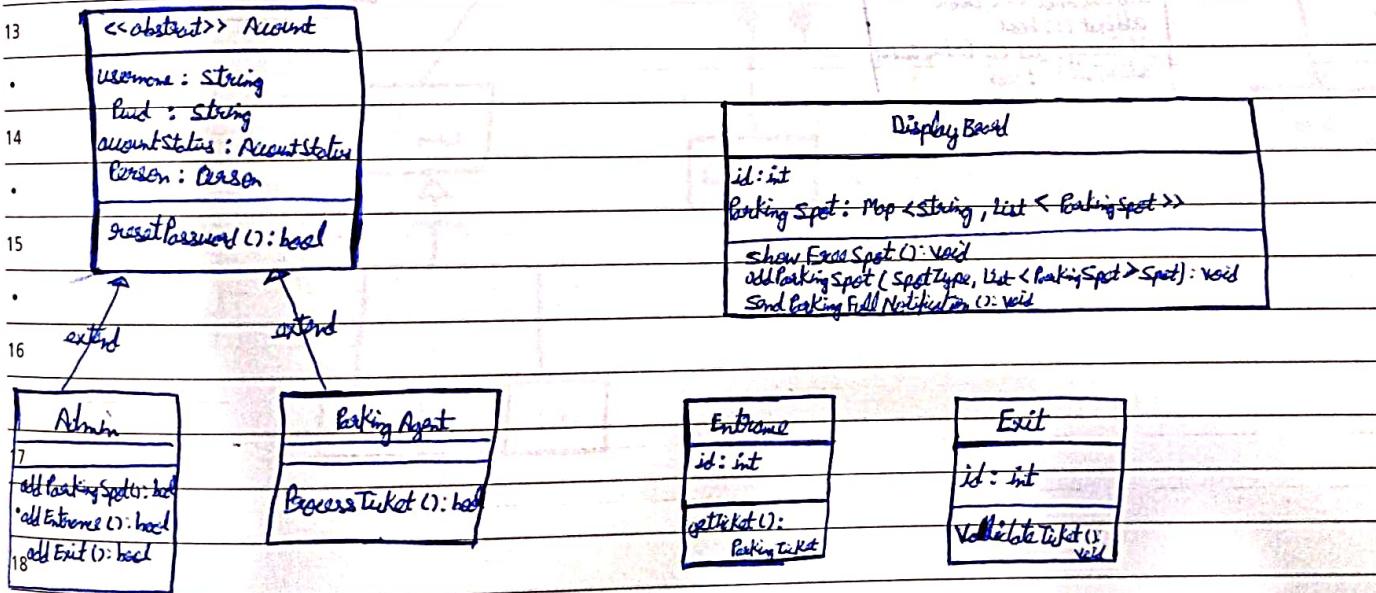
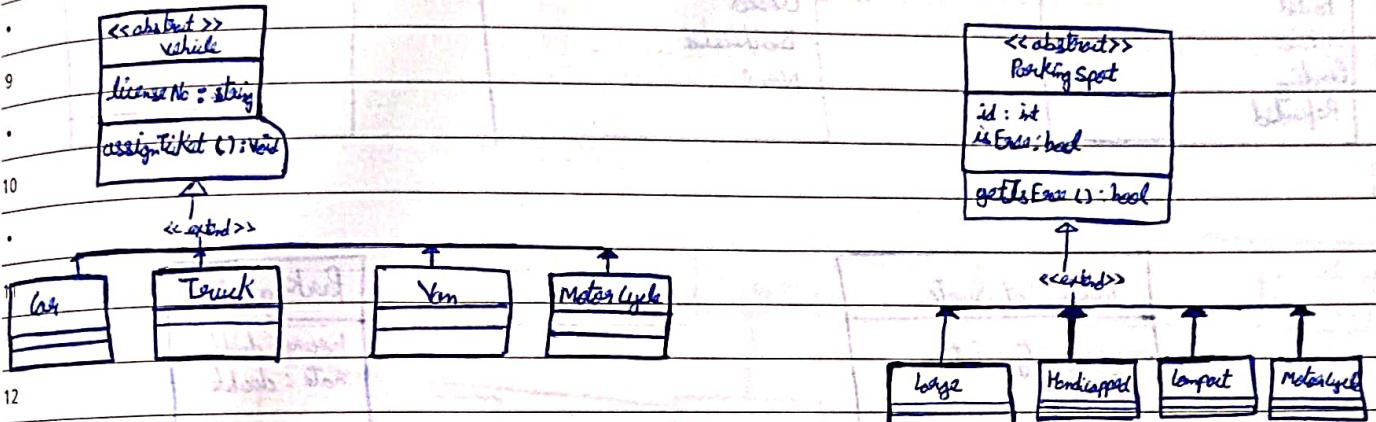
12

316/49

46.week

hence requirement details are hi 60-70% design have long backe Khat ka dinay se?

① PARKING LOT SYSTEM



A smile is worth a million, but it doesn't cost a cent. – Henry C. Mabry

DEC

13

NOVEMBER
Sunday

317/48

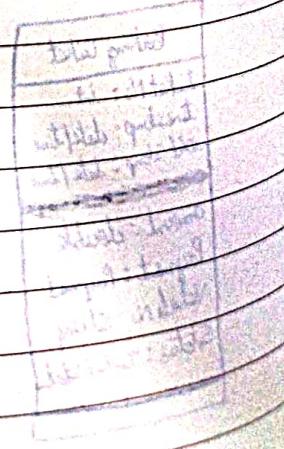
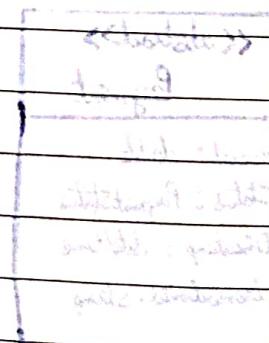
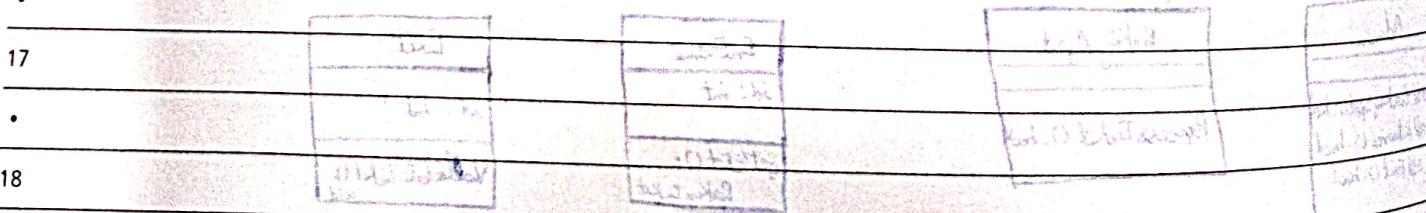
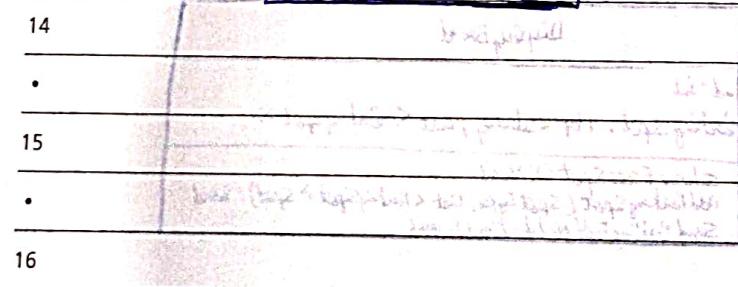
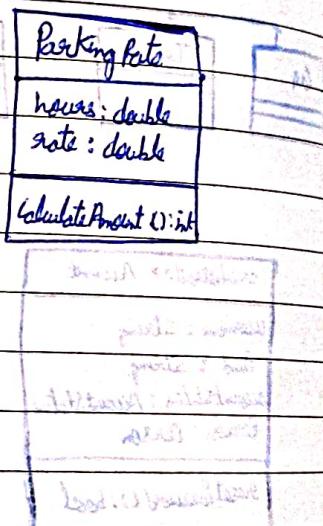
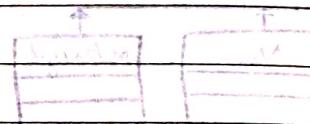
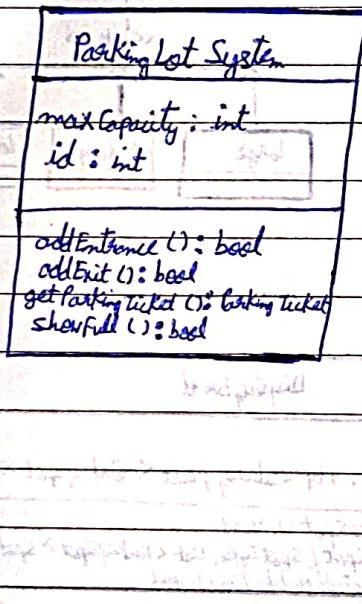
	<code><<enum>> PaymentStatus</code>
8	<code>Completed</code>
•	<code>Failed</code>
9	<code>UnPaid</code>
•	<code>Pending</code>
•	<code>Refunded</code>
10	

`<< enum >> AccountStatus`

Active
Closed
Blacklisted
None

46. week Kunstnerei Ticket status

<u>Parking Rate</u> hours: double rate: double	
<u>CalculateAmount()</u> : int	
<pre> double hours; double rate; double amount; amount = hours * rate; return amount;</pre>	



A fanatic is someone who can't change his mind and won't change the subject. – Winston Churchill

12/2022

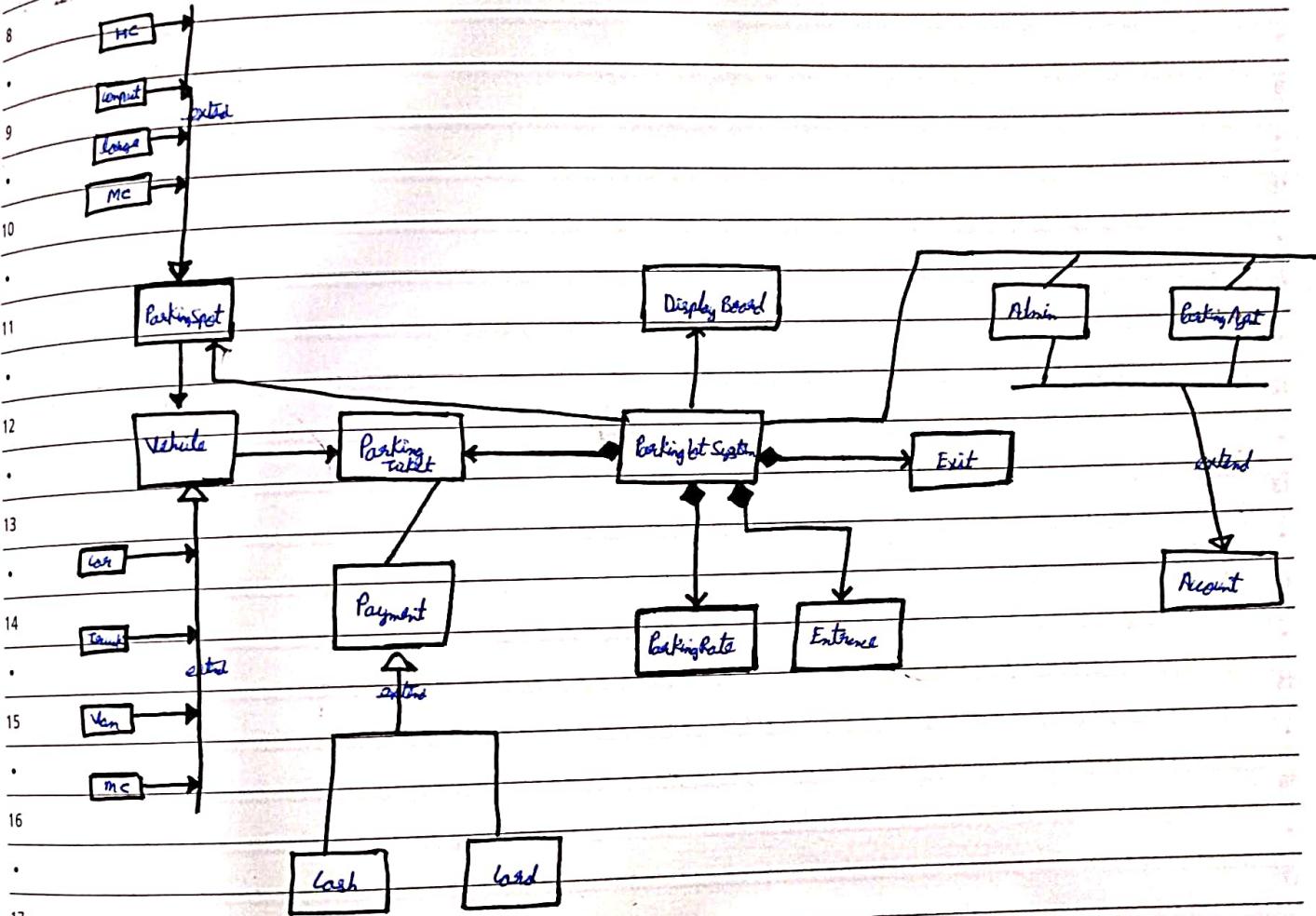
Monday	-	5	12	19	26
Tuesday	-	6	13	20	27
Wednesday	-	7	14	21	28
Thursday	1	8	15	22	29
Friday	2	9	16	23	30
Saturday	3	10	17	24	31
Sunday	4	11	18	25	-

47.week

NOVEMBER
Monday

14

318/47



Anyone who thinks he knows all the answers isn't up to date on the questions. – Frank Lawrence

DEC