

Agenda :

- ① Type of LLD Interviews
- ② How to approach LLD Interviews
- ③ Design a Pen Famous Amazon Problem.

LLD - 4 (Case Studies)
Tic Tac Toe
Parking lot
Book My Show
Splitwise

13-15 Sessions

- we will study about what all things to keep in Mind in your actual interview .
- How do you cover all the aspects that will be asked in LLD interview .

Type of LLD Interviews.

Theoretical	Design	Machine Coding Round
* Old MNCs, Banks ORACLE Mid size Non Tech Product	* Big MNCs, Amazon, walmart/PayPal	* Flipkart/Myntee/Rbreach Every startup
* Test Knowledge instead of Problem Solving Skills. ↓	* Case Study what will be the : interface, class, structure.	* Design + code
* OOP/Design Pattern <small>Whether you know or not</small>	* Problem Solving Round	* write end to end working for given set of requirements.
* 30-45 min	* 30-45 min	* upto 2.5 hours
* No code or at max, code for the question asked.	* 1 Question Asked. Code for at max few of the classes from your design	* 1 Question, good number of followups.

Design & Machine Coding Rounds

1 hr

Design

- Given a single line problem statement.

Design a Pen

Design a Fastag

Design a X.

- Gather requirements

You will have to think
about the requirements.

Discuss with interviewer ~~XX~~

Class diagrams ~~XX~~

Schema design

Write code for some

of the classes/interfaces

Machine Coding Round

- Detailed documented problem statement
- Clarify requirements.

Class diagrams ~~XX~~

Schema design

Complete end to end working code, Satisfy all the requirements.

You can miss few of the req^{nts}, but the major req should be working in code.
80-90% could be passed

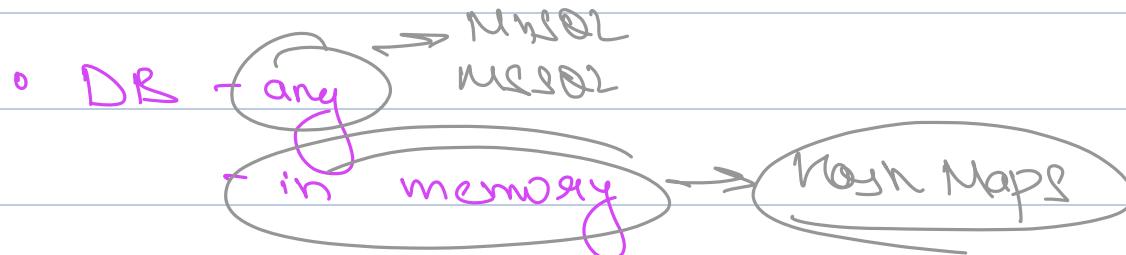
Could be NOT on IDE

Don't rely on IDE suggestions

- Follow ups

- Code on your own machine
- Tech Stack - nothing to be concerned about.

they have already short-listed your resume.



25 - 30 min

10

Req gathering / Clarifying Questions / Class diagrams

5 - 10

10

1 hr - 1.5 hr

Code



15 - 30 mins

evaluation / follow-up

At Scaler

- 1 line Problem Statement
 - Gather requirement
 - Classify requirement.
 - Class Diagrams ~~class~~
 - Schema Design
 - End to End working code.
- Discussion and come up by requirement owner(s).

This covers both Design and Machine Coding Rounds.

Design a Pen

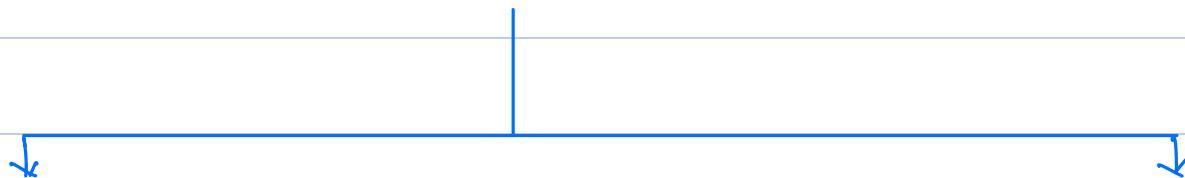
?



Step 0 : Get an Overview

- Align your thoughts with the thoughts of interviewer
- Hint about requirements

Overview



You already know
about the system.

Pen ↗

when you don't know
about the system

Splitwise X
(You may not know)
about this

Confirming and Aligning

Is it a normal day to day use Pen OR an Apple Pen of the topic.

It is OK to be unaware of the topic.

what you know ↴

align the direction ↴

more
so

Ask the interviewer to

give you an overview.

Dont

Assume

~~★ You have to be a good~~ ~~anyway~~

Listener and be Focused!!

In the case of Pen too...

Questions you can ask :

① what exactly you want me to design.
- Entities Classes / Interfaces and Class Diagrams
Design a Bird / Pen

- Real world S/w design system

Controllers / Services in LLD 1

② Do I have to persist the data?
use DB

use an actual Database OR keep in memory.

③ How is the user going to interact?

- API \rightarrow Spring APIs

- Command Line \rightarrow inputs from terminal

- Hard Code \rightarrow Hardcoded input in the
Main.java

② and ③ indicate you have to design S/w system

Class Diagrams are mandatory.

- Step 0 is used to classify your thoughts, And your thinking is not misaligned with the interviewer's thoughts
- It is fine to get an overview.

5 - 10 mins

Step 1 : Gather and Classify req^m

(and the most obvious)

- Suggest some rational ideas and ask if we should support these features.
- Suggest requirements you will be able to execute. Don't suggest tricky ones.
- 5 - 8 core features / Requirements.

Visualize the system. (Outside to Inside)

How will you use the system. (User Journey)

• Jot down every point you listen to. 15

• If this is the feature, how can it change in future?
Keep future nuances in mind.

Boolean → Enums.

login by email → login by phoneNo/Social
in future

payment → refund → same account
on wallet.

• Simple Edge cases / future scopes.

Pen: Come up with 5-8 features : Entity Design 16

attributes

classes

behaviours

Inheritance

enums

parts/components.

How far can you think!

Gathering requirements and Clarifying Questions

① what is a pen?

Abstract Class

Any physical entity that can have a write behaviour is a pen.

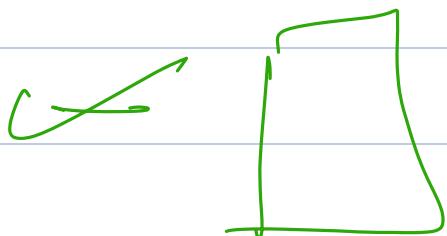
(The main use case / behaviour of a pen)

Do we support Markers, Sketch Pen, Apple Pencil

scribbly - This will be very scribbly. If they say yes, it will be trouble.

② Supports different kinds of Pen:

- Gel Pen
- Ball Pen
- Fountain Pen
- Use And Throw Pen



③ Gel and ball pen have a refill.

nib

Others don't have a refill.

Refill will have an ink.

Ink Type

{ enum class }

④ Different inks can have different colors

colours

String attribute

I: Is it good to associate colour with pen, or colour with ink?

⑤ For every Pen, we need to store brand, name and price

enum OpenCloseType

Cap
Relax
Rotate

⑥ Pens can be either closed via Cap or via Click buttons. Rotate could be another possibility in future.

22

⑦ You can ask: Often refills have different nib sizes. Do we have to store radius of nibs.

23

⑧ Fountain Pen itself has a nib and ink but no refill. Not having a nib on refill.

future: Can work in Spain

behaviours:

24

① Pen should be able to write.

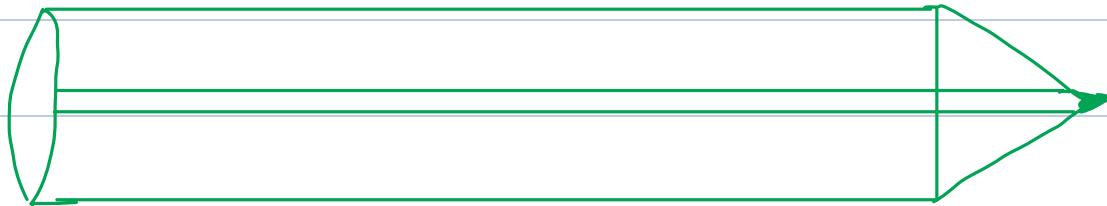
② we can close/open the pen.

③ we should be able to refill pens.

25

Class Diagrams

① Visualizing the requirements.



Going from outside to inside
to figure out entities.

② By identifying the nouns in the requirements.

This approach will be taken for Parking lot

Book My Show
Splitwise

This is the general suggested approach.

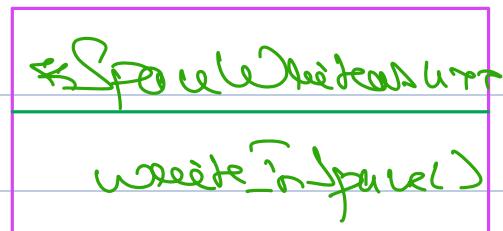
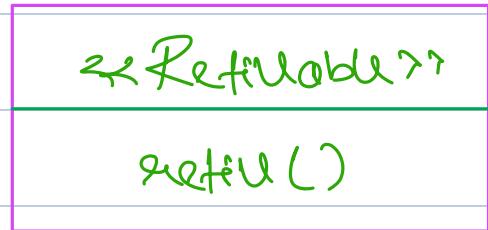
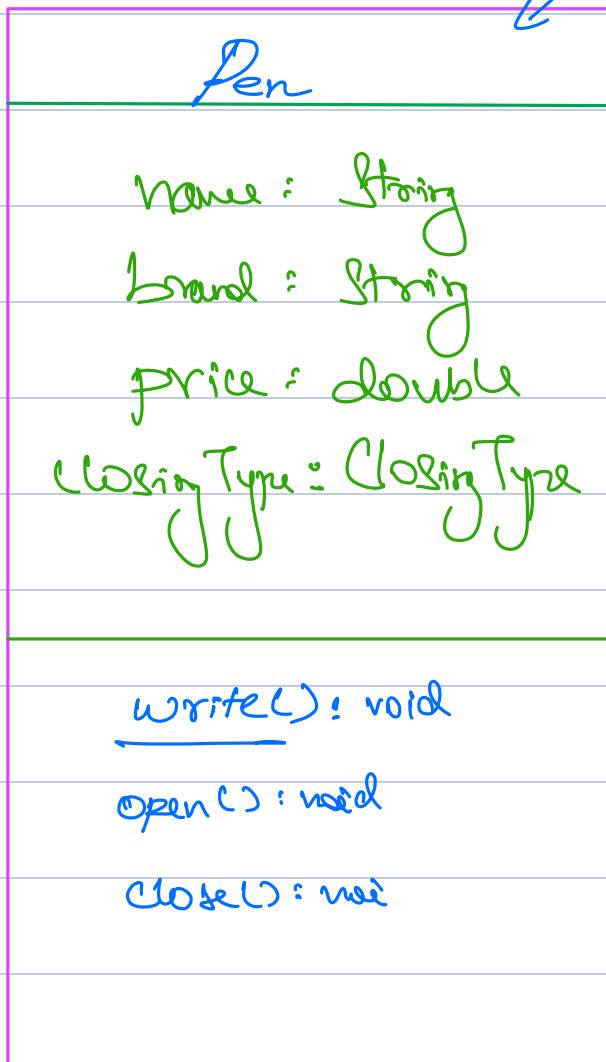
You have to come up with only entities.

- Classes ✓ Don't think about Services / Controllers.
- Interfaces ✗
- Enums ✓
- Factories (if needed)

The very first entity here? Pen

Is Pen a concrete class / Abstract class / Interface?

Abstract class



entails



GelPen

impl. refillable,

- refill

BallPen

impl. refillable,

- refill

FountainPen

- ink

- nib

UseAndThrowP

- refill

SpurPen

impl SpurWriteable

HeldSpur

impl SpurWriteable

- ink

- nib

writeInSpur

@Override
refill()

@Override
refill()

true or false

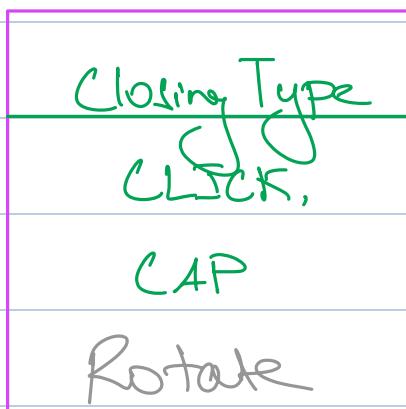
⑥ Is it OK to use boolean in isClick?

whenever you have a boolean, try to convert that to enum instead.

31

Employee

isPresent: boolean
isTerminated: 4



← Think about Future ~~changes~~

32

- You can have different interface

`<<Clickable>>` `<<Capable>>` `<<Rotatable>>`

`click()`

`on()`

`rotateClockwise()`

`off()`

`rotateAntiClockwise()`

If you want to have this as behaviour and not as attributes, \Rightarrow when combination is possible

- Here having an attribute makes more sense if it is always of one type. \rightarrow attribute
- Gel with CLICK & CAP etc... } interface
- Ball Pen with Click & Rotate. } interface

33

Visualize the Pen. Are there more Entities?

Always use Enums instead of Booleans.

Refill
ink
capacity
nib

Nib
radius

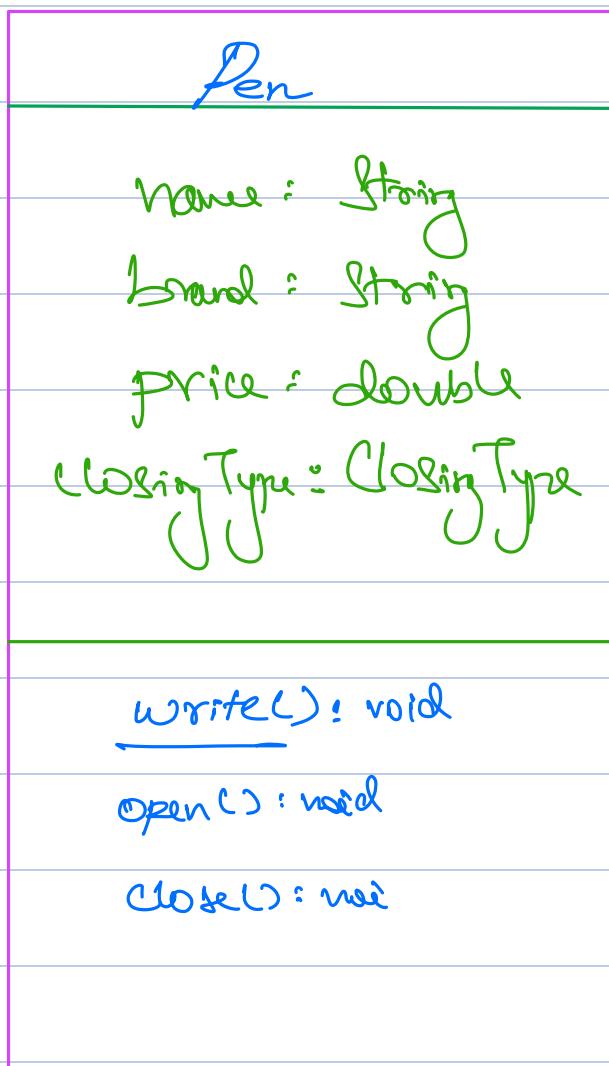
ink
color: string
inkType: InkType

InkType
Normal,
Sparkling,
Waterproof
Transparent

Follow up: what if all GelPens are not refillable.

A) Let GelPen extend or Pen only. Dont make it impl. Refillable

So child class of gel Pen can implement Refillable.



extends



- refill



refill()