Today's content

1. Types of LLD interviews
2. How to approach LLD interviews
3. Design a Pen.

Types of LLD interview.

1. Theoritical
2. Design
3. Machine coding

1. Theoritical

1. Service based / midtier / old tech companies.
2. Test your knowledge rather than testing how you apply your knowledge.
   → OOPS | Design Patterns / Java.

   Ex: Static keyword.

3. Timed [45-60 minutes]

   Ex: Singleton → Please write code → Handle concurrency.

2. Design [Product based companies]. [1 line statement] [1 hour]

Given

Q: Design a sdm similar to flipkart

What are they testing?

How do you apply the theory that you have learnt to real-time problems.

What is expected out of you?

1) 5-8 features that you should list down by taking an explicit confirmation from your interviewer.
2) class diagram, schema diagram.
3) A follow up question, how to handle that?

3. Machine coding [startups, Razorpay, Flipkart --] [2hr-3hr]

Q: Details of what you need to achieve?

What are they testing?

How do you apply the theory that you have learnt to real-time problems.

Also, implementation skills.

What is expected out of you?

1. Understand the requirements, Ask for clarifications
2. Class-diagram, schema dia

} ½ hr.

3. 1.5 hours you'll sit & write code.
4. Test what you've done.

& follow-up questions.

Design	Machine Coding
1. Single line case study,	1. Detailed requirements.
2. How you structure the code. How do you bring clarity for ambiguous system.	2. Clarity of requirement that you've got from the given doc.
3. Create a class diagram which will handle the above requirements frozen on? Schema diagram	3. Come with class-diagram.
4. Some classes/ interface / design - Patterns _ code on the paper.	4. Write code on your machine for all the requirements
5. Follow ups.	5. Follow ups.

Design a Pen

Step 0: Overview of system.

- You know something about the system
  - Explain a bit & make sure your understanding of the sim is correct.
- You do not know anything
  - Ask the interviewer to give an overview.

1. Do you want me to create

   (i) entities

   (ii) real world sw design (working code)

2. Do youive to persist data?

3. How user will interact with the system?

   [Rest API's, command line, hardcode].

Step 1: Gather and clarify requirements.

1. Start suggesting features and ask if he needs it/not.
2. (5-8) core features.
3. Try to visualise the sln & suggest some ideas.
4. Note down the pointers that you feel are needed going ahead.

   Think a little of what future enhancements can come in.

Whats your defn of pen?

1. A physical entity that writes is a pen.

   Yes

2. Should I support diff. types of pen?

   Yes;

   GelPen, BallPen, FountainPen.

3. GelPen, BallPen can have refills, FountainPen won't have.
4. Refills can ink,

5. Diff inks can have diff. colors.

6. Pens can have cap or click buttons.

Behaviours

Should all entities support this behaviour.

1. Pen should write → Yes
2. Refill the Pen → No for fountain pen
3. Open / close the pen → work via cap / button.

Class diagram

1. Visualising the sim / requirements.

TODAY

Go outside to inside to figure out the entities.

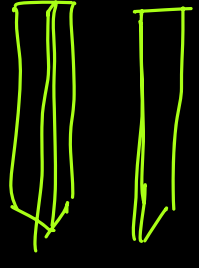
2. By finding nouns in your requirements.

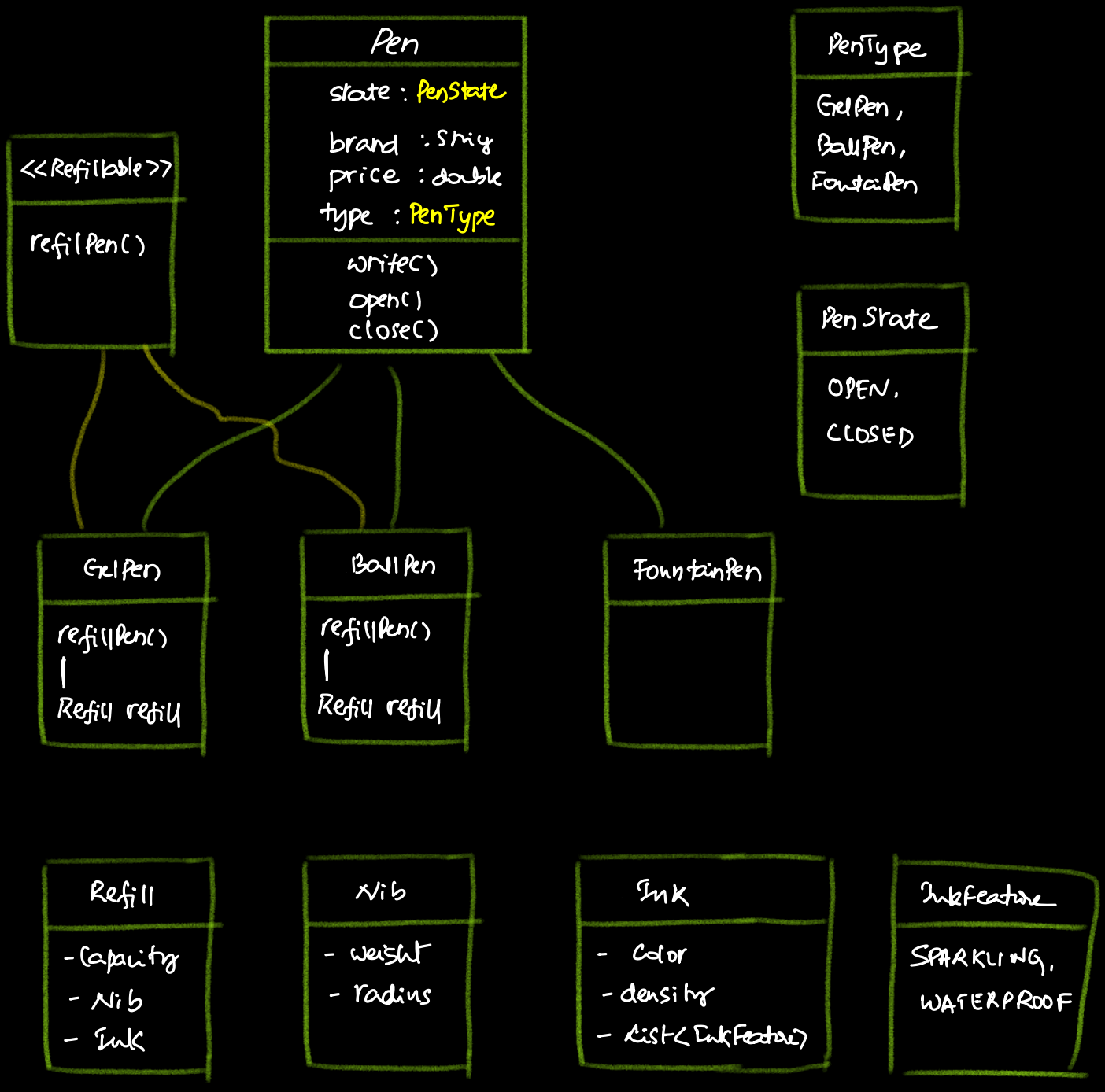
Parking Lot, BMS, Splitwise

① Visualisation

entities, interface, enums around your entities.

Show usage of any Design-Patterns you're planning to use.





Choose enum type over boolean.