

INTRODUCTION TO SYSTEM DESIGN

#1



24:46

Monday
12-05-2025
May

DAY-1

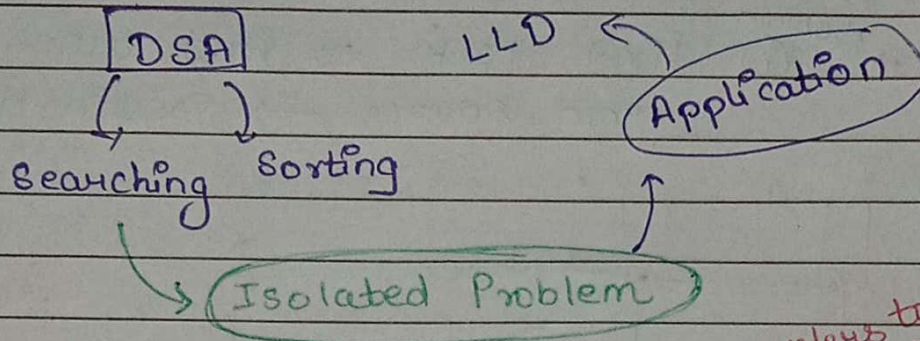
Lecture-1

Introduction To System Design

A little progress each day adds up to big result

Date _____
Page _____

→ What is LLD?



→ refers to isolated problem

→ In DSA, to search an element we use linear search or binary search

↓
Concept are used to solve that isolated problem

→ Like that array sort is an isolated problem and quick sort and merge sort are solution to that problem.

→ So using many DSA concept we build an application that is known as LLD.

Story to Understand LLD and link with DSA

Anurag

↳ DSA ✓
↳ LLD X

Mawrya

↳ DSA ✓
↳ LLD ✓

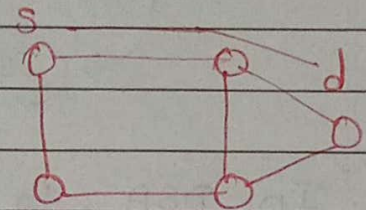
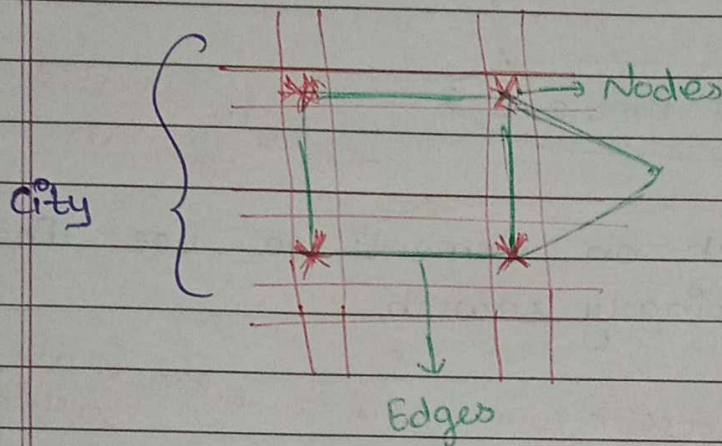
They both get placed at QuickRide Company which book ride like OLA, UBER.

Manager give task to both students to make application QuickRide.

Now Anurag don't have knowledge of LLD so he think only in DSA perspective

So Anurag think application build means Algorithm.

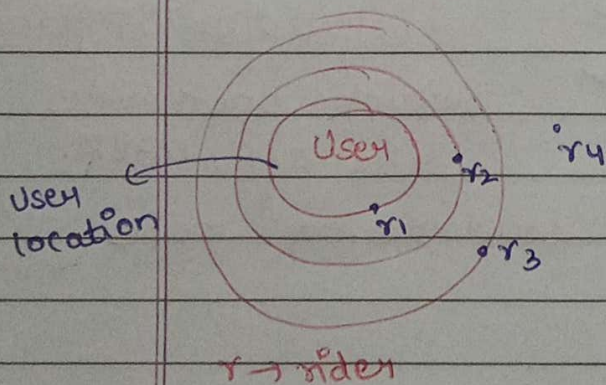
Anurag → ^{Problem} ① First task to ~~find~~ take user from source → Destination



→ Anurag assume that, interaction of roads as Nodes and path between them are edges so he make a map of a city using graph

→ And to find the shortest distance between source and destination he uses Dijkstra's algorithm

Problem-2 : first problem arises to assign a rider to a user before problem-1 (source → destination)



$r_1 \quad r_2 \quad r_3 \quad r_4$

Priority Queue

Min Heap

Max Heap

(Smallest element will be on top)

(Maximum element will be on top)

→ ~~Every user~~ He think what if he map priority queue around every user and put the riders in that Min-heap.

→ r1, r2, r3, r4 when we pop the element assign the closest rider to the user

After solving problem he go to Manager

Manager -

- ↳ You provide the Algorithm
- ↳ Where is the application

- ↳ Object / Entities
- ↳ Relationship
- ↳ Data Security
- ↳ Notification
- ↳ Payment gateway
- ↳ Millions of User Handle

Anurag doesn't solve this problem he directly jump to Algorithm

Mainya : first he think about the structure or LLD, DSA Come after that

→ Which are objects / Entities

- ↳ User
- ↳ Rider
- ↳ Location
- ↳ Notification
- ↳ Payments

→ Relationship b/w objects

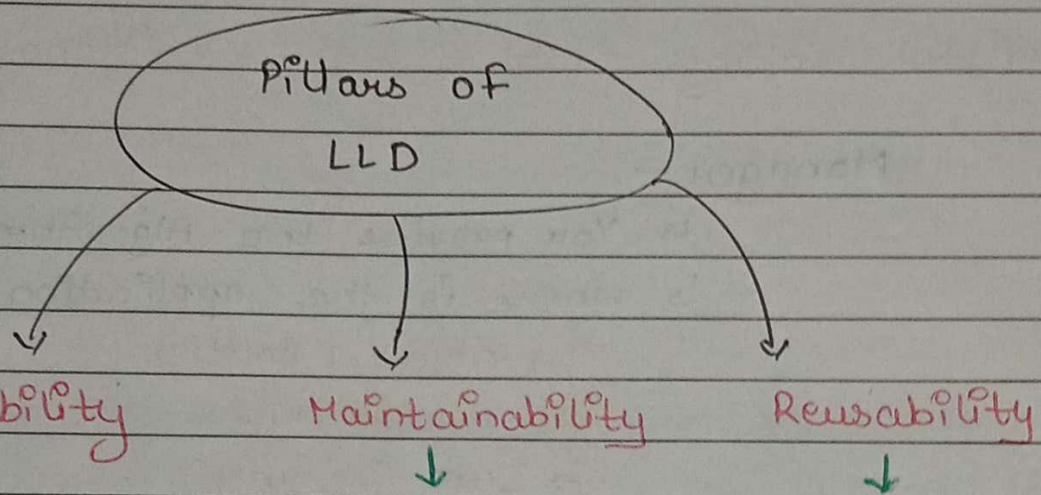
User ↔ Rider

→ Data Security - After Ride Completion Neither User require rider phone number nor rider require user phone number

→ Million of user handlation in application

After making structure or LLD then he go to DSA part.

→ Three Pillars of LLD



Scalability



→ When application use by millions of users then how it will be sustain
→ If we add new feature the how easily we can scale that

Maintainability



→ How code is maintain, If we add new feature then old feature should not give bug.
→ Code should be written like that new feature should be integrate easily
→ Your code should be easily debugable.

Reusability



→ Your code should be highly reusable
→ Notification and payment code should be integrate easily with other application
→ Plug and play Model

→ Tightly coupled

X Your code should not be tightly coupled

Rider - Mapping Algorithm can be used in zomato, Amazon Clone

→ What is Not LLD?

HLD: High level Design

We write negligible code because all focus is on architectural design

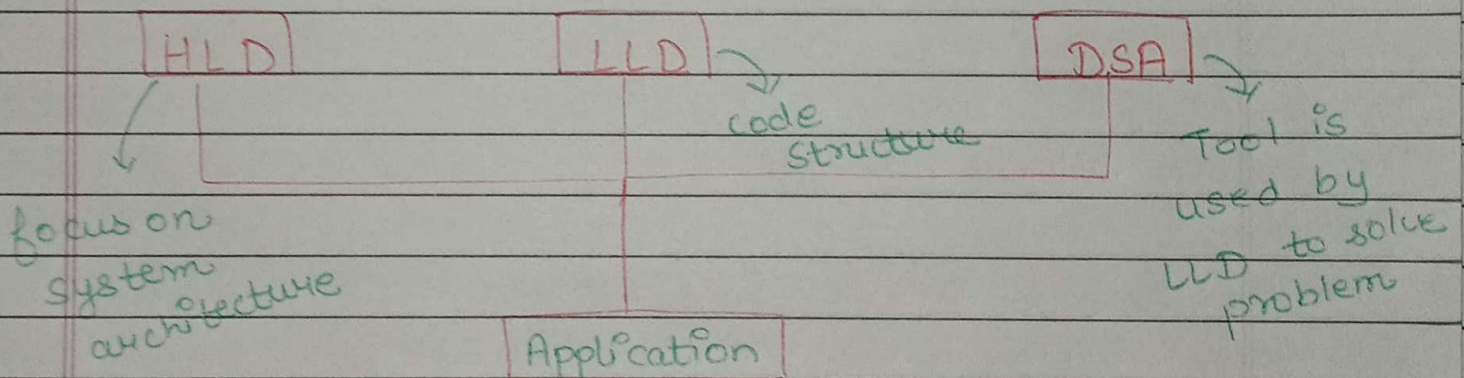
- Tech stack (Java / SpringBoot)
- DB (SQL / No-SQL)
- Server scale: When millions of users come then how server is going to scale
- Cost optimization

↓

Highly optimize

Users ↑ server ↑
Users ↓ server ↓

} To optimize the cost



HLD, LLD and DSA these all are used to build application / software

Note:

DSA is the brain of an Application then LLD is the skeleton.