

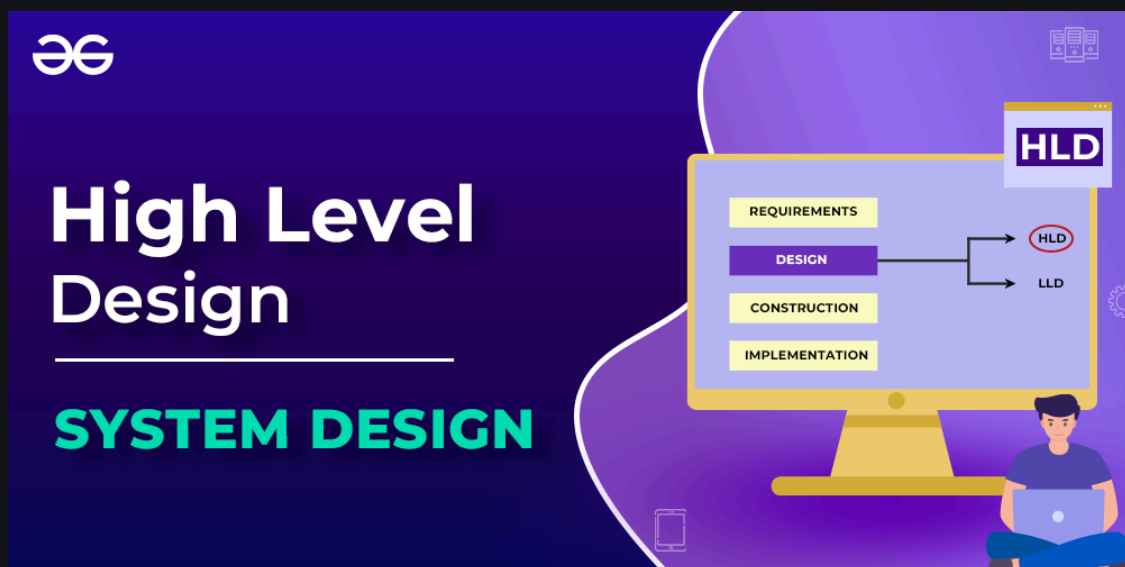
# What is High Level Design – Learn System Design

Last Updated : 15 Oct, 2024



In Developing scalable applications, proper planning, and organization play a significant role. High-level design plays an important role in this process by serving as the blueprint of the system's architecture. It provides a comprehensive view of how components interact and function together which will further help for detailed implementation.

*Understanding high-level design is very crucial for developers, architects, and project managers because it allows them to make sure that all stakeholders are aligned with the project objectives. High-level design is also known as macro-level design.*

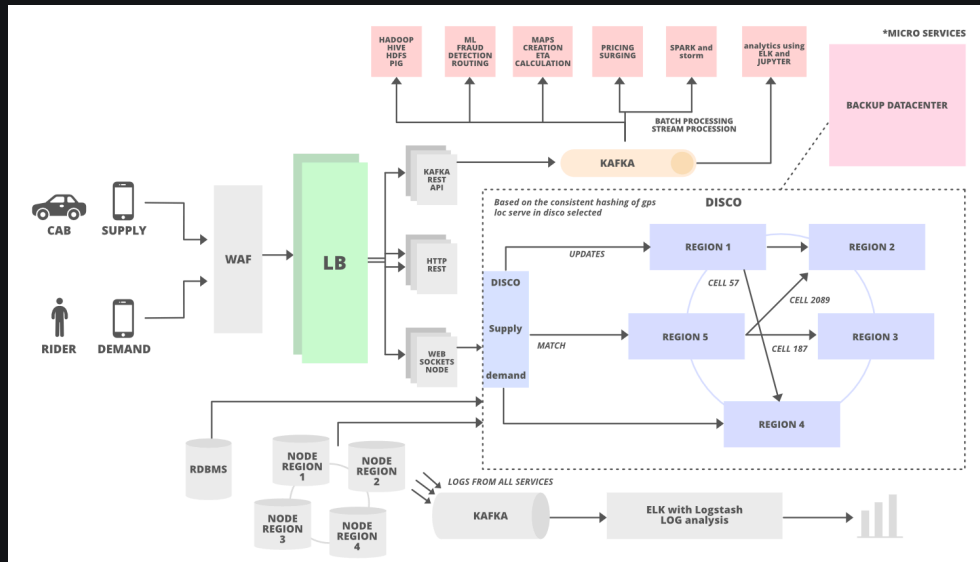


What is High-Level Design

## Table of Content

- [What is High-Level Design?](#)
- [Components of High-Level Design](#)
- [What is High-Level Design Document?](#)

High-Level Design is a initial step in development of applications where overall structure of a system is planned. High-Level design focuses mainly on how different components of the system work together without getting to know about internal coding and implementation. This helps everyone involving in the project to understand the goals and ensures good communication during development.



Components of High-Level Design

## Components of High-Level Design

Understanding the components of high-level design is very important for creating effective systems that meets user needs and technical requirements. Below are the main components of high-level design:

1. **System Architecture:** System architecture is an overview of the entire system which represents the structure and the relationships between various components. It helps to visually represent how different parts interact and function.
2. **Modules and Components:** High-Level design breaks down the systems into modules or components each with specific roles and responsibilities, and has a distinct function that contributes to entire system helping in developing an efficient system.
3. **Data Flow Diagrams (DFDs):** Data Flow Diagrams demonstrates the data movement within the system. They help to understand how information is processed and handled.

interfaces (APIs) and user interfaces necessary for seamless interaction between components.

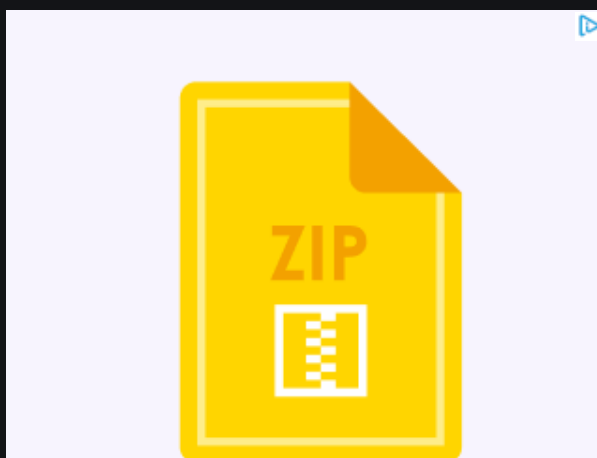
5. **Technology Stack:** The technology stack are various technologies and tools that will be used in the development of the system. This includes programming languages, frameworks, databases.
6. **Deployment Architecture:** It includes how the system will be hosted and accessed. It includes server configurations, cloud infrastructure, and network considerations.

## What is High-Level Design Document?

HLD document consists of data flows, flowcharts, and data structures to help developers in understanding and implement how the current system is being designed intentionally to function. This document is responsible for explaining the connections between system components and operations which depict the logic. The architecture design needed (for the system's functionality and flow) for each and every module of the system as per the functional requirements.

## Purpose and Characteristics of High-Level Design

The purpose of this High-Level Design (HLD) is to add the necessary detailed description to represent a suitable model. This is designed to help with operational requirements and will help to understand how the modules interact. Basically, HLD is a technical representation of functional requirements and the flow of information across components. Characteristics of High-Level Design include:





- A diagram representing each design aspect is included in the HLD (which is based on business requirements and anticipated results).
- Description of hardware, software interfaces, and also user interfaces.
- The workflow of the user's typical process is detailed, along with performance specifications.

## How HLD is different from LLD

High Level Design or HLD is a general system design where we do tradeoffs between different frameworks, components, and different databases and we choose the best considering what the business needs and how the system should work, both in terms of functional and non functional aspects.

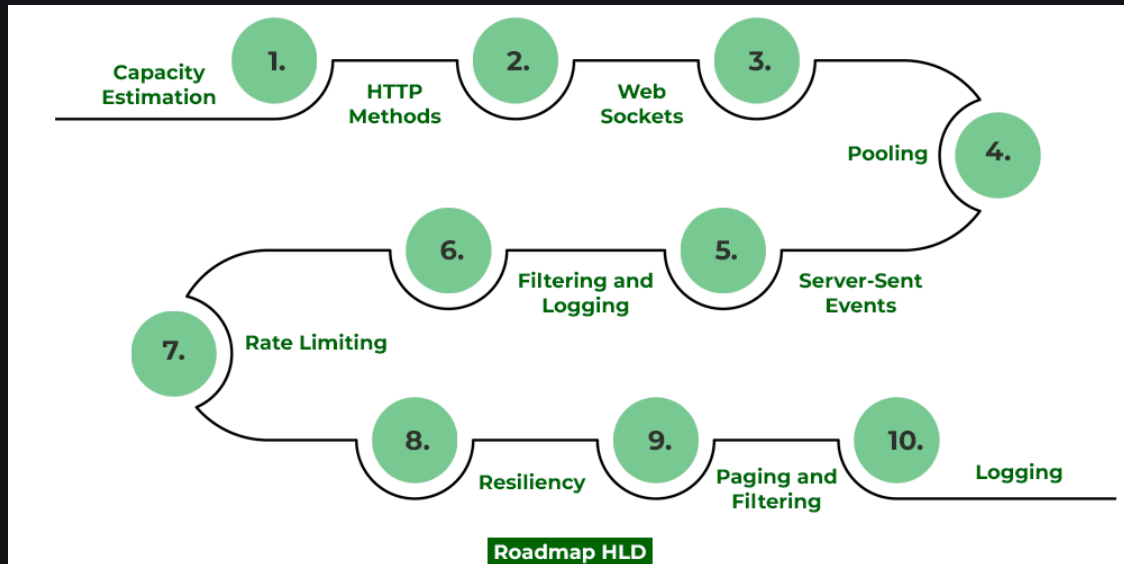
Whereas LLD(Low Level Design), translates the HLD into smaller and more specific details, It includes class diagrams, methods, data structures, and algorithms, focusing on how each part will be implemented. It is Primarily used by developers and technical teams.

*Further Read: [Differences between High Level Design\(HLD\) and Low Level Design\(LLD\)](#)ely detailed peculiar and specific code chunks snippets as per patterns are there while designing LLD.*

Let us now discuss the HLD roadmap after having an adequate understanding of HLD and how it is different from LLD which is shown below as an infographic:

## How To Design Scalable High-level Design (HLD)

Now in order to design any high-level system, certain terms are listed in a sequential manner so as to connect the dots in order to understand, Roadmap is very important for working professionals because these help us to get a complete binding understanding of how services in real-time are getting scaled at a high level.



*How To Design Scalable High-level Design (HLD) Systems*

## 1. Capacity Estimation

Capacity estimation in system design involves predicting the resources (such as processing power, memory, and bandwidth) required to meet the expected workload. It ensures that a system can handle current and future demands efficiently, helping in the proper allocation of resources and preventing performance bottlenecks.

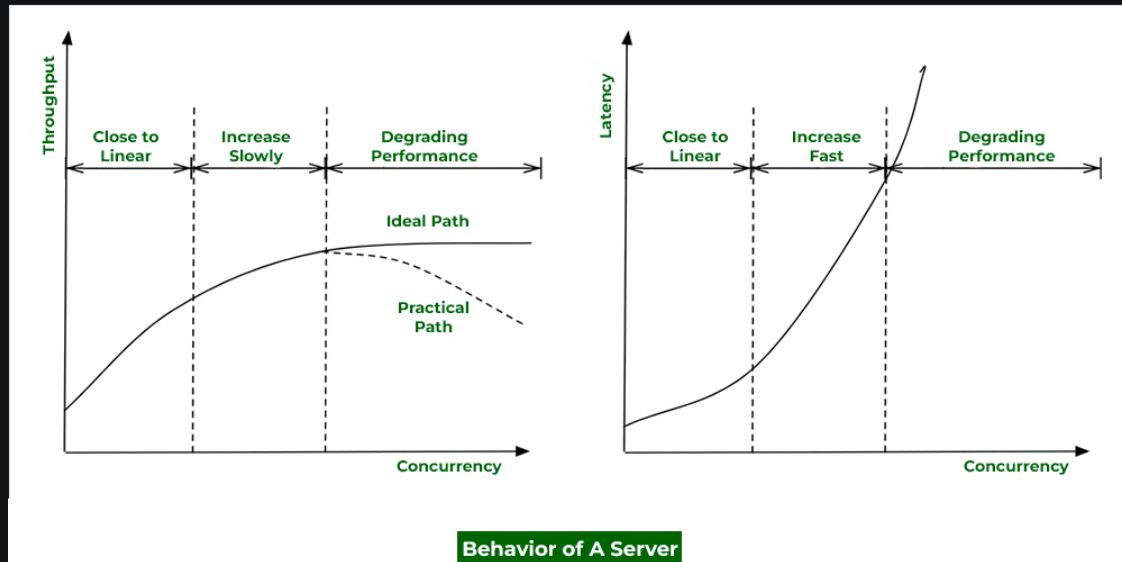




For example:

*Twitter which is recently in newsfeeds these days, here at high-level designing we need to make sure tweets of popular influencers are reaching out to millions of people so how we need to scale our system so service should not be hampered.*

We have also shown below the characteristics behavior of a server which is measured across throughput and latency within the system.



*Comparison of Throughput and Latency with Concurrency : Behavior Of A Server*

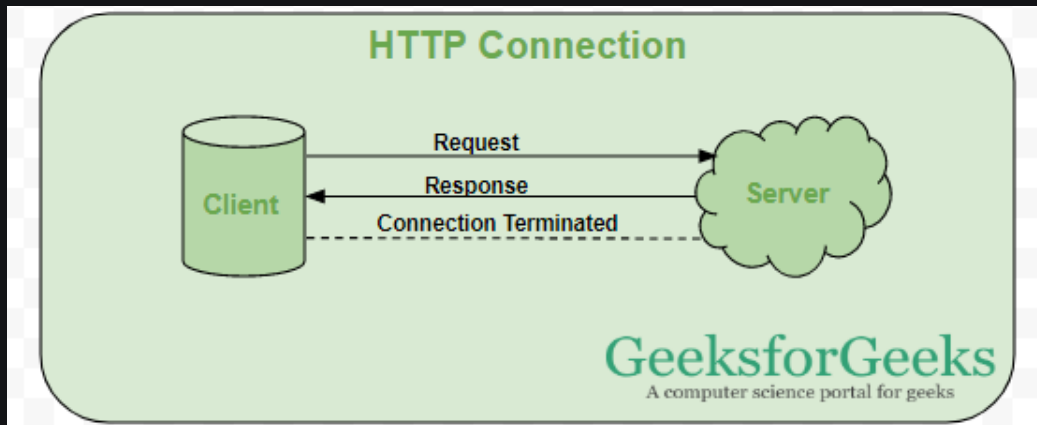
## 2. HTTP and HTTPS and their methods

HTTP(HyperText Transfer Protocol), is used to transfer the data over the web which enables the communication between clients and server that

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

HTTP which adds an extra layer of security through SSL/TLS encryption. Methods of HTTP include GET, PUT and POST.

***Tip:** These are also important as per interview perspective as constraint bounds in interviews in API over when to use what is very important.*



***Tip:** There are 2 ways to send data from server to client:*

- *Using Websockets*
- *Using Polling*

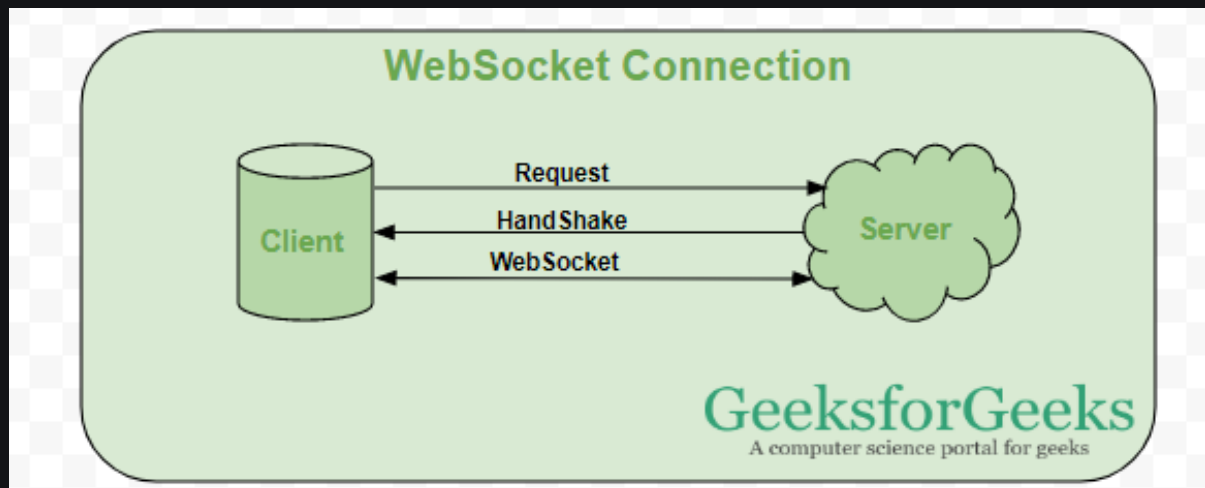
### 3. Web Sockets

There are two ways to send data from server to clients which are websockets and polling. In a [client-server architecture](#), we are sending a request to server and server sends it back and so in this way communication takes place. But in designing the system biggest problem we face is whether we can implement the client-server model because we can not wait until the server responds.

- Here Web Sockets play a crucial role that solving problems and enabling us to have real-time communication.
- WebSocket is a full-duplex protocol as it allows the application to send and receive data at the same time.



- They are very useful in Real-time web applications, gaming applications, chat applications.

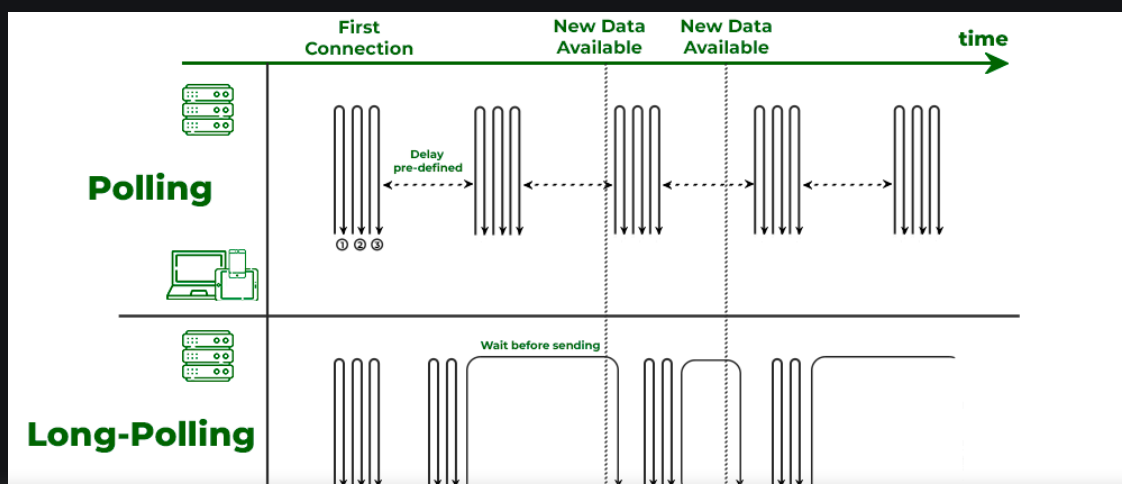


WebSocket Connection

#### 4. Polling

Another way of sending and receiving data from a server just likely as we do we doing above in the case of web sockets Polling. There are mainly 2 types of polling:

- **In short polling**, we sent the request. In short polling, the client sends a request to the server and the server immediately sends data at fixed intervals whether the data is updated or not.
- **In long polling**, the server waits for the client's request to respond and responds only and only if some new data is available or if some data is being updated.

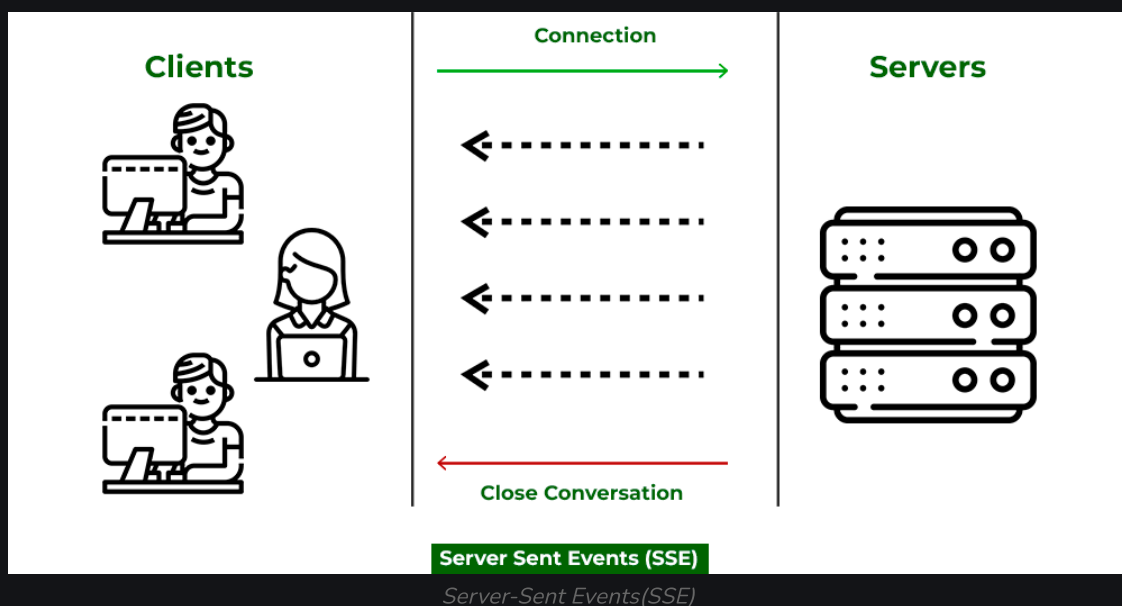




***Tip:** Long polling is preferred over short polling because lesser number of requests are sent in a system.*

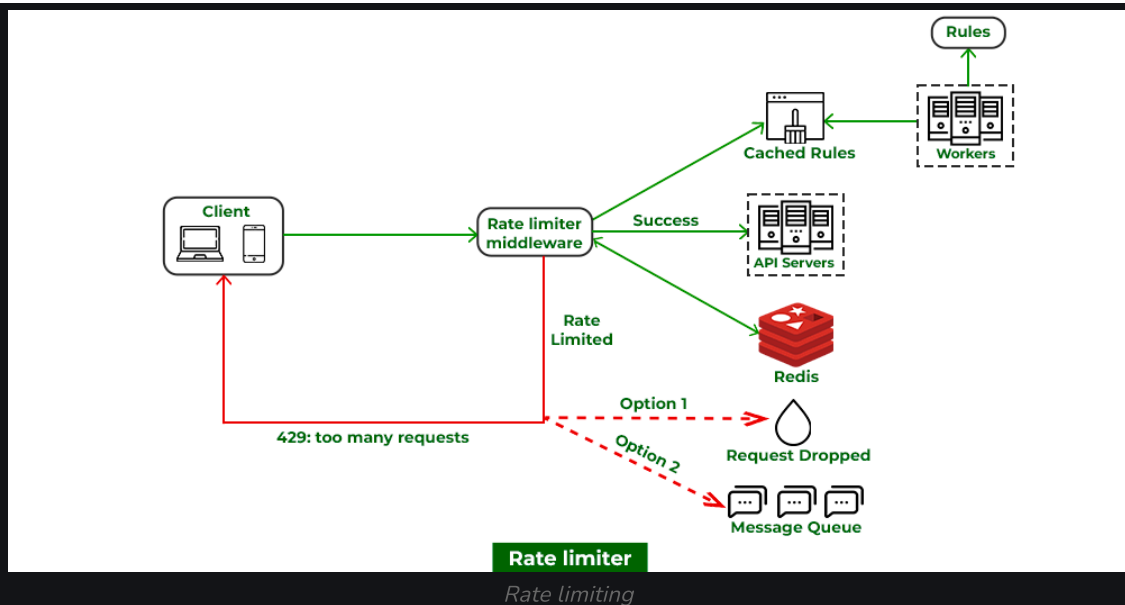
## 5. Server-Sent Events(SSE)

It is purposely built as one-way communication from servers to clients in specific design systems. SSE is a technology that helps to push real-time updates from the server to the clients over HTTP connection. SSE enables server to send data automatically as it becomes available. Example of SSE include real-time streaming.



## 6. Rate limiting

A rate limiter restricts the number of events that can be done in a timeframe by restricting the number of requests a sender can send in a given period of time. Here once the threshold limit is reached now further it blocks the incoming requests as can be seen below media as follows:



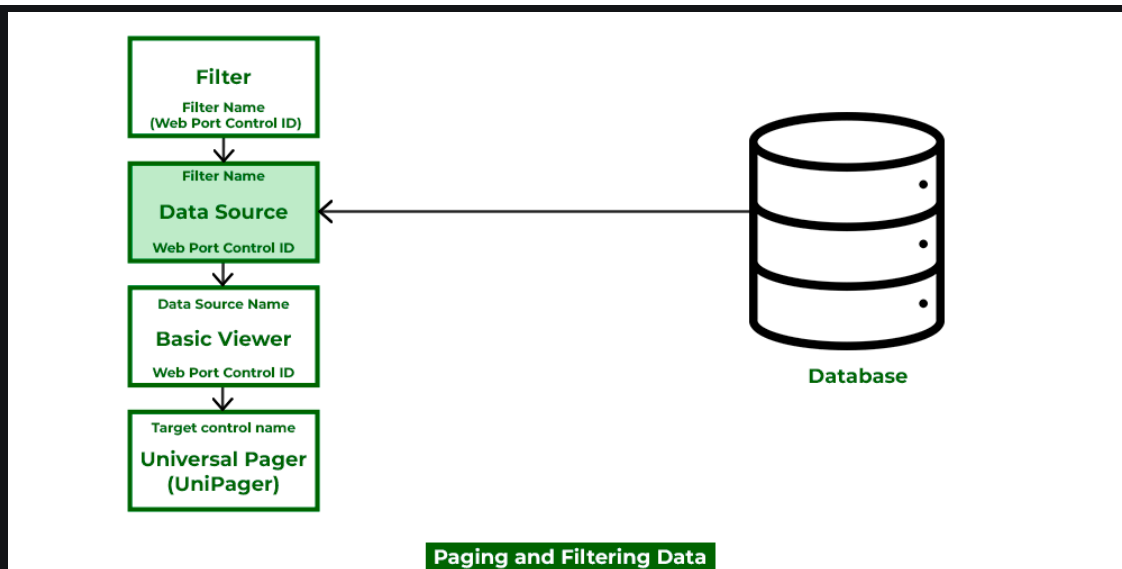
## 7. Resiliency

No matter how great a system we design there is always a chance of faults and failure which could be because of hardware issues or software issues (such as running low on memory) or there can be some human error. In such cases we need to provide resiliency through replication, redundancy and availability.

**Note:** System design interviews starts with open ended designing a specific system which later is bounded with constraints at multiple levels. In order to clear dealing with these bounds in layman language is known as resiliency via implementing common tradeoff in the system. Hence providing resiliency is very crucial in designing complex system designs and also in clearing interviews.

## 8. Paging

Paging in High-Level Design (HLD) refers to the method of dividing large datasets or content into smaller, manageable pages. This approach improves user experience by allowing users to load data incrementally, reducing initial load times and enhancing performance.



Paging

## 9. Logging

A log file records details of events occurring in a software application. The details may consist of microservices, transactions, service actions, or anything helpful to debug the flow of an event in the system. Logging is crucial to monitor the application's flow. This can also be useful for tracking the health and performance of a distributed system, as well as for debugging issues that may arise. There are several approaches to implementing distributed logging, including:

- Using a [centralized logging service](#)
- Using a [distributed logging system](#)
- Using a cloud-based logging service

Want to be a Software Architect or grow as a working professional? Knowing both Low and High-Level System Design is highly necessary. As such, our course fits you perfectly: [Mastering System Design: From Low-Level to High-Level Solutions](#). Get in-depth into **System Design** with hands-on projects and **Real-World Examples**. Learn how to come up with systems that are scalable, efficient, and robust—ones that impress. Ready to elevate your tech skills? Enrol now and build the future!

## 2. High level design:

- The expectation here is to identify major components of the system in agreement with the interviewer
- Draw the components and their interaction, in every step ask for feedback from the interviewer
- Ensure that initial design is small (fits into a page / half of the whiteboard)
- The interviewer could ask to detail out certain critical components.

Introduction to High Level Design

[Visit Course](#)



solan...



12



[Next Article](#) >

Horizontal and Vertical Scaling | System  
Design

## Similar Reads

### Difference between High Level Design(HLD) and Low Level Design(LLD)

System design involves creating both a High-Level Design (HLD), which is like a roadmap showing the overall plan, and a Low-Level Design (LLD), which is ...

🕒 4 min read

### HLD or High Level System Design of Apache Kafka Startup

Apache Kafka is a distributed data store optimized for ingesting and lower latency processing streaming data in real time. It can handle the constant...

🕒 7 min read

### Vending Machine: High Level System Design

Vending machines which are a common thing to be seen in shopping malls or metro stations, kind of operate by themselves. It's different from other system...

🕒 5 min read

Low-Level Design (LLD) is the detailed design process in the software development process that focuses on the implementation of individual...

🕒 5 min read

### What is the goal of High-Level Design(HLD)?

The goal of HLD is to provide a detailed overview of the system's structure, components, and interactions, setting the direction for the detailed design an...

🕒 2 min read

### What is High-Level Design Document?

A High-Level Design document is a technical document that outlines the overall architecture and design of a software system at a high level. It serves...

🕒 2 min read

### Google's Search Autocomplete High-Level Design(HLD)

Google Search Autocomplete is a feature that predicts and suggests search queries as users type into the search bar. As users begin typing a query,...

🕒 7 min read

### Top High-Level Design(HLD) Interview Questions 2024

High-Level Design (HLD) gives a big-picture view. It shows the main parts of a system and how they fit together. HLD acts as a map, focusing on modularity,...

🕒 9 min read

### How to Draw High Level Design Diagram?

System design is the process of designing the blueprint of the software system that guides the developer to how you have to build the software that meets...

🕒 13 min read

### High Latency vs Low Latency | System Design

In system design, latency refers to the time it takes for data to travel from one point in the system to another and back, essentially measuring the delay or la...

🕒 4 min read



Corporate & Communications Address:-  
A-143, 9th Floor, Sovereign Corporate  
Tower, Sector- 136, Noida, Uttar Pradesh  
(201305) | Registered Address:- K 061,  
Tower K, Gulshan Vivante Apartment,  
Sector 137, Noida, Gautam Buddh  
Nagar, Uttar Pradesh, 201305



## Company

- About Us
- Legal
- In Media
- Contact Us
- Advertise with us
- GFG Corporate Solution
- Placement Training Program
- GeeksforGeeks Community

## DSA

- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- Top 100 DSA Interview Problems
- DSA Roadmap by Sandeep Jain
- All Cheat Sheets

## Web Technologies

- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS

## Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial
- Tutorials Archive

## Data Science & ML

- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

## Python Tutorial

- Python Programming Examples
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

## Computer Science

Operating Systems  
Computer Network  
Database Management System  
Software Engineering  
Digital Logic Design  
Engineering Maths  
Software Development  
Software Testing

## System Design

High Level Design  
Low Level Design  
UML Diagrams  
Interview Guide  
Design Patterns  
OOAD  
System Design Bootcamp  
Interview Questions

## School Subjects

Mathematics  
Physics  
Chemistry  
Biology  
Social Science  
English Grammar  
Commerce  
World GK

## DevOps

Git  
Linux  
AWS  
Docker  
Kubernetes  
Azure  
GCP  
DevOps Roadmap

## Interview Preparation

Competitive Programming  
Top DS or Algo for CP  
Company-Wise Recruitment Process  
Company-Wise Preparation  
Aptitude Preparation  
Puzzles

## GeeksforGeeks Videos

DSA  
Python  
Java  
C++  
Web Development  
Data Science  
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved