

## Assignment-1

### 1) How does client server architecture works?

The Client-server architecture is a distributed application structure that divides tasks or workloads between the service requester called clients, and the providers of a resource or service, called servers. It refers to a system that hosts, delivers and manages most of the resources and services that the client requests. In this model, all requests and services are delivered over a network, and it is also referred to as the networking computing model or client-server model.

In the client-server architecture, when the client accepts the requested process and delivers the data packets requested back to the client. Examples: Email, WWW, etc.

Let's divide into the Client-server model and have look at how the Internet works through web browsers.

1. Client: The word Client, means to talk of a person or an organization using a particular service. Similarly, in the digital world, a Client is a computer (Host) i.e. capable of receiving the information or using a particular service from the services providers (Server). For example: when you open a website, your browser acts as the client.
2. Servers: It means a person or medium that serves something. A Server is a remote computer that provides information (data) or access to particular services. For example, when you visit a website, the web server sends the webpage to your browser.
3. Communication: The client and server communicates over a network (like in the internet). The client sends a request, the server processes it, and then the server sends back a response.

Let's look example of online banking. When you log in your bank's website or app.

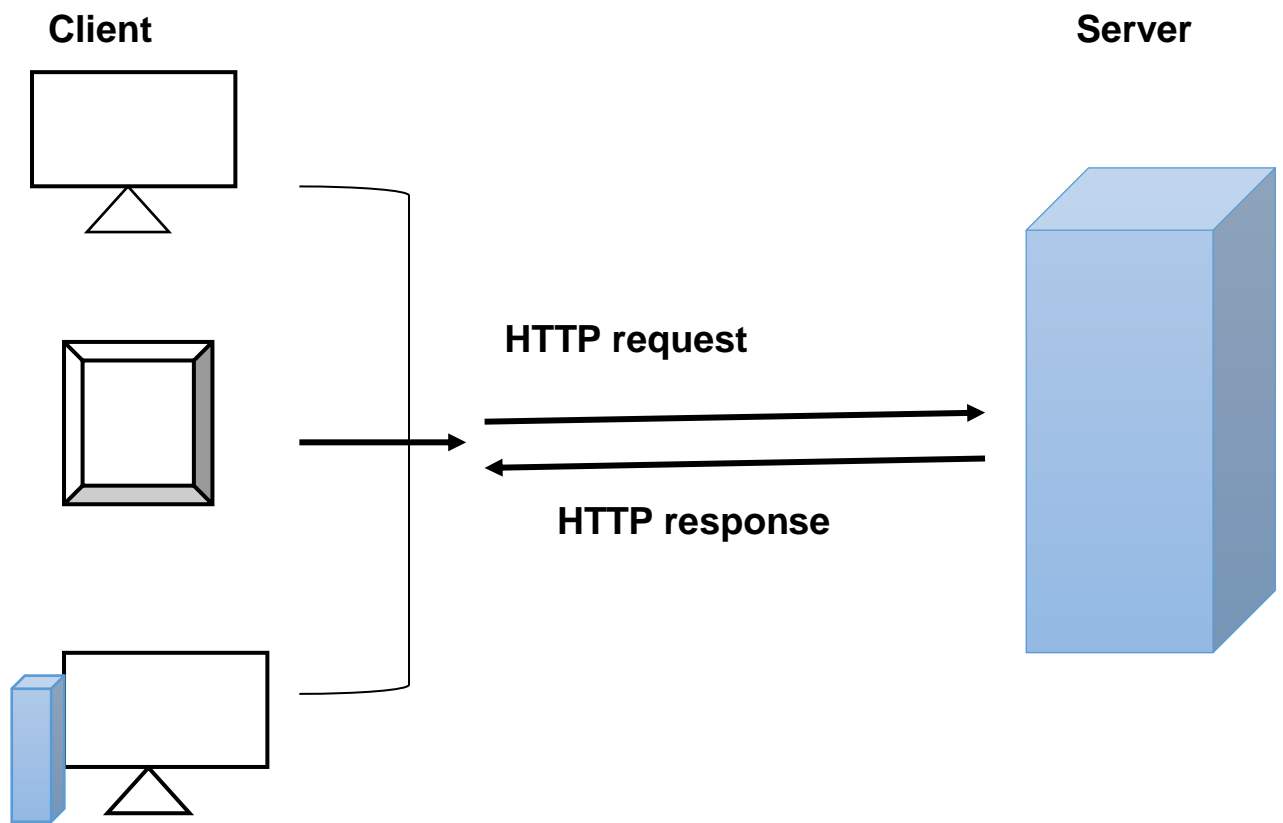
Client: You enter your login details and request your account balance. Your device sends this request to the bank's servers.

Server: The bank's server receive your request, verify your credentials, and fetch your account information from their database.

Communication: The server sends your account balance and transaction history back to your device.

Result: You can view your balance, transfer money or pay bills all because of the client-server model.

Client-server architecture is the foundation of how the internet and most modern applications work. It divides tasks between clients and servers, enabling seamless communication over a network.



**Figure: The Client-Server Architecture**

## 2) What are the factors to be considered when designing a software?

Designing a software is a multifaceted process that requires careful planning to ensure the final product is efficient, reliable and user-friendly. It involves creating a blueprint which outlines the architecture, components, interfaces and data for a system to meet specified requirements. It serves as a critical phase in software development, that bridges the gap between requirements analysis and implementation.

Here are the some factors to be considered while designing a software.

Some of them are described below:

- Requirement Analysis

Before initiating the design, it is essential to thoroughly understand both functional and non-functional requirements. Functional requirements define the specific behaviors or functions the system must perform such as data processing, user authentication. On the other hand, Non-functional requirement describe the system attributes like performance, security and usability.

- Software Architecture

The architecture serves as the foundational blueprint of the software which determines how the components interacts and integrate. Selecting an appropriate architecture style such as layered, client-server, or microservice. This help to profound implications on the system's scalability, maintainability and also in performance.

- User Interface (UI) and User Experience (UX) design

A well-crafted UI/UX is pivotal for user satisfaction and engagement. The design ensures that users can navigate the software effortlessly, leading to increased productivity and reduced error rates.

- Performance and Efficiency

The software performance directly impacts user satisfaction and operational efficiency. To ensure the system operates efficiently under expected load conditions, it is essential to optimize with algorithms and manage all the resources effectively.

- Security

In an era where data breaches and cyber threats are prevalent, incorporating security measures into the software design is negotiable. This encompasses by implementing authentication and authorization protocols to ensure that only authorized users can have access to specific functionalities.

- Maintainability and Scalability

With future growth and adaptability in mind, designing software ensures its relevance and functionality over time. This approach involves writing clean, modular code that allows for easy updates and modifications without disrupting the entire system. By utilizing load balancers, the system can handle increased user demand and data volume efficiently.

- Compatibility and platform Independence

Ensures that the software operates seamlessly across various platforms, devices and operating systems broadens its accessibility. This might involve using cross platform development frameworks to web standards.

- Testing and Debugging

To ensure that the software functions as intended and meets quality standards various testing methodologies should be included such as unit testing, integration testing, system testing and acceptance testing.

- Documentation

Comprehensive documentation serves as a vital resource for both developers and end-users. For developers, technical documentation provide insights into the system architecture, codebase and design decisions. Similarly, for end-users, user manuals and help guides offer instructions on how effectively utilize the software's features.

- Cost and Time Constraints

Identifying and focusing on the most critical features that deliver the highest value to users and stakeholders. Also allocating financial resources to various aspects of project such as development, testing and marketing.

### 3) Why do we really need network programming tools and platform?

Explain some of them.

Network programming tools and platforms are essential for developing, managing and optimizing networked applications and systems. They provide the necessary frameworks and utilities to facilitate communications between devices over a network for ensuring efficient data transmission, security and scalability.

We need network programming tools for the following purposes:

- Simplified communication: These tools offer abstractions and libraries which simplify the complexities of network protocols, enabling developers to focus on application logic.

- Enhanced security: Many tools come with built-in with security features such as encryption, authentication mechanisms which help in protecting data integrity.
- Scalability: The development of scalable applications helps to handle increased loads efficiently.

Some examples of Network programming tools and platforms are as follow:

1) Wireshark (free, open source)

It is the industry standard tool to collect and interpret with the traffic. It enables engineers to quickly get to the packet level of a problem. It also allows them to quickly determine if the issue is due to the network, server, service or client.

2) Nmap (free, open source)

A network scanning tool which has the capability to scan whole subnets and TCP port ranges, allowing engineers to spot problem devices and open sockets.

3) iPerf3(free, open source)

It is a tool that enables engineers to measure network throughput, packet loss and jitter. These measurements help to pinpoint whether the network is causing the performance problem or not.

4) Cisco Packet Tracker (free)

Cisco designed the Packet Tracker tool to help engineers simulate and test network environments before they are rolled out to the enterprise. By using these tools, as large environments can be built and tested without the need for expensive hardware.

#### 5) cURL (free, command-line, open source)

It allows network engineers to query URLs from the command line. If mobile device applications, IOT devices or other APIs are having connectivity issues, the cURL utility can help to test authentication, certificates to help troubleshoot the cause of the problem.