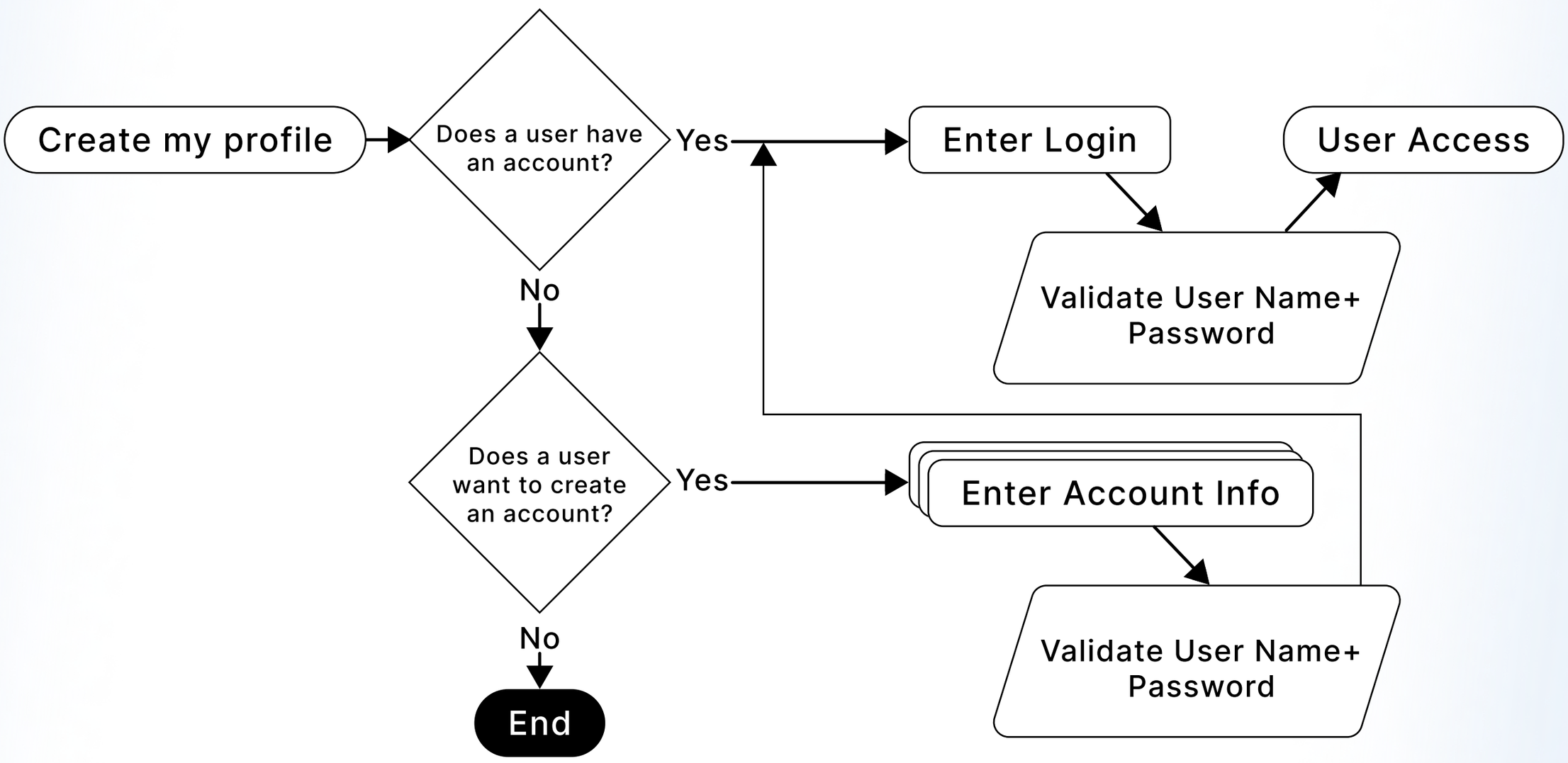
BOSSCODER ACADEMY

SYSTEM DESIGN

IN JUST 60 DAYS



BASIC → ADVANCED

DAY 1

What exactly is System Design?

📝 Topics to cover

- What is System Design and why is it important?
- Key components of a system: Clients, Servers, Databases, Caching, Load Balancing, Proxies.
- Horizontal vs. Vertical Scaling.

🗂️ Resource

- System design primer: Learn the basics of system design

❓ Questions

1. Explain why system design is crucial in software development.
2. What is the difference between horizontal and vertical scaling
3. Name three key components of a typical system architecture.

DAY 2-3

System Design Basics - Scalability

📝 Topics to cover

- Horizontal Scaling
- Vertical Scaling

🗂️ Resource

- System Design - Horizontal and Vertical Scaling - GeeksforGeeks

❓ Questions

1. Explain the main differences between horizontal and vertical scaling
2. Explain the terms "scaling out" and "scaling up" in the context of horizontal and vertical scaling
3. What are the main advantages and challenges of horizontal scaling?

4. What are the main advantages and challenges of vertical scaling?
5. Provide examples of applications or systems where horizontal scaling is a better fit.
6. Provide examples of applications or systems where vertical scaling is a better fit.
7. Is it possible to combine horizontal and vertical scaling strategies in a single system? Why or why not?

DAY 4-6

System Design Basics - Reliability and Availability

📝 Topics to cover

- Reliability
- Mean Time Between Failures (MTBF)
- Redundancy
- MTTR (Mean Time To Repair)
- High Availability Architecture
- Service Level Agreements (SLA)

🗂️ Resource

- System Reliability & Availability Calculations – BMC Software | Blogs

Questions

1. What does "reliability" mean in the context of system design, and why is it essential?
2. Define the terms MTBF (Mean Time Between Failures) and MTTR (Mean Time To Repair). How are they related to system reliability?
3. Explain the concept of redundancy. How does redundancy contribute to improved system reliability?
4. Define "availability" in system design. Why is high availability important?
5. Explain the concept of planned downtime and unplanned downtime. How do they impact system availability?
6. Why is monitoring important for maintaining system availability?

DAY 6-9

Basics of Networking

📋 Topics to cover

- OSI Model
- TCP/IP Protocol Stack

🗂️ Resource

- Layers of OSI Model - GeeksforGeeks
- OSI Model Explained | OSI Animation | Open System Interconnection Model | OSI 7 layers | TechTerms
- TCP/IP Model - GeeksforGeeks
- TCP IP Model Explained | TCP IP Model Animation | TCP IP Protocol Suite | TCP IP Layers | TechTerms

Questions

1. What does OSI stand for, and what is the OSI Model used for in networking?
2. List the primary functions of all the seven layers of the OSI Model.
3. Which layer of the OSI Model is responsible for routing and logical addressing?
4. Provide an example of a protocol or technology associated with each layer of the OSI Model.
5. Explain the concept of encapsulation in the context of the OSI Model.
6. How does the TCP/IP Protocol Stack compare to the OSI Model in terms of layers?
7. Which protocols operate at the Transport Layer of the TCP/IP stack?
8. Explain the differences between TCP (Transmission Control Protocol) and UDP (User Datagram Protocol)

DAY 10-12

IP Address Routing

📋 Topics to cover

- IP Addresses
- Subnetting
- Routing

🗂️ Resource

- What is an IP Address? | Definition from TechTarget
- What is IP Routing? - GeeksforGeeks
- Introduction To Subnetting - GeeksforGeeks

Questions

1. What is an IP address, and what is its primary purpose in networking?
2. Differentiate between IPv4 and IPv6 addressing schemes.
3. Explain the difference between a public IP address and a private IP address.
4. Why is subnetting used in networking, and what problem does it address?
5. Explain the concept of a subnet mask in subnetting
6. What is routing in networking, and why is it necessary for data transmission?
7. How does a router determine the optimal path for forwarding data packets?
8. Differentiate between static routing and dynamic routing protocols.

DAY 13-15

DNS and CDN

📋 Topics to cover

- Domain Name System (DNS)
- DNS Resolution
- Content Delivery Network (CDN)
- Content Replication and Caching

🗂️ Resource

- What is DNS? | How DNS works | Cloudflare
- How a DNS Server (Domain Name System) works.
- What Is A CDN? How Does It Work?
- What is a content delivery network (CDN)? | How do CDNs work? | Cloudflare

Questions

1. What is a Content Delivery Network (CDN), and how does it improve website performance?
2. Explain how a CDN minimizes latency for end users.
3. Describe the concept of edge servers in a CDN.
4. How does a CDN handle caching and content replication?
5. Give an example of a situation where a CDN would be especially beneficial.
6. What is the Domain Name System (DNS), and what is its primary function?
7. Explain the difference between a domain name and an IP address.
8. Describe the process of DNS resolution when a user enters a URL in a browser.

DAY 16-18

Load Balancing

📋 Topics to cover

- Active-passive
- Active-active
- Layer 4 load balancing
- Layer 7 load balancing
- Horizontal scaling

🗂️ Resource

- Load Balancers in System Design
- Load Balancer - System Design Interview Question - GeeksforGeeks
- What is Load Balancing? ⚖️

Questions

1. What types of information can layer 4 load balancers use to distribute traffic?
2. When would you choose layer 4 load balancing over layer 7 load balancing?
3. Define layer 7 load balancing and its role in system architecture.
4. Discuss the challenges associated with maintaining data consistency in an active-active setup.
5. How does an active-passive setup ensure minimal downtime during failures?

DAY 19-20

Consistency

📋 Topics to cover

- Weak consistency
- Eventual consistency
- Strong consistency

📁 Resource

- Idea of Consistency patterns in System Design
- Data Consistency and Tradeoffs in Distributed Systems

❓ Questions

1. How does weak consistency allow for relaxed synchronization among distributed nodes?
2. How does weak consistency impact the trade-offs between availability and consistency?
3. Describe the concept of convergence in an eventually consistent system.
4. Explain how distributed transactions and two-phase commit protocols relate to strong consistency.

DAY 21-23

CAP Theorem

📋 Topics to cover

- Consistency
- Availability
- Partition Tolerance
- CP
- AP

🗂️ Resource

- System design fundamentals: What is the CAP theorem?
- The CAP Theorem in DBMS - GeeksforGeeks

Questions

1. Explain the trade-offs among consistency, availability, and partition tolerance according to the CAP theorem.
2. Give an example of a real-world situation where partition tolerance is necessary.
3. What kind of systems prioritize consistency and partition tolerance over availability?
4. What trade-offs might CP systems face during network partitions or failures?
5. What kind of systems prioritize availability and partition tolerance over strong consistency?
6. How does an AP system balance read and write operations during normal operation and partitions?

DAY 24-26

Microservices and Proxy Servers

📝 Topics to cover

- Microservices and their Communication
- Service Resilience and Fault Tolerance
- Forward and Reverse Proxies
- Load Balancing with Proxy Servers

🗂️ Resource

- Microservices Design Guide 👨‍🏫. Everyone has heard about Microservices... | by Thilina Ashen Gamage | Platform Engineer | Medium
- Pattern: Microservice Architecture
- https://www.enjoyalgorithms.com/blog/proxies-in-system-design
- System Design, Chapter 6: Proxies

Questions

1. Explain the communication patterns between microservices. How do synchronous and asynchronous communication differ?
2. Describe how you would handle authentication and authorization in a microservices ecosystem.
3. Compare and contrast a forward proxy and a reverse proxy. When would you use each?
4. Describe a scenario where you might use a load balancer in conjunction with a reverse proxy.
5. Explain how a proxy server can improve performance through caching. What are the trade-offs?

DAY 27-29

Storage

📋 Topics to cover

- Block storage
- File storage
- Object storage
- Redundant Disk Arrays (RAID)
- Hadoop Distributed File System (HDFS)

🗂️ Resource

1. System Design — Storage. Storage concepts and considerations in... | by Larry | Peng Yang | Computer Science Fundamentals | Medium
2. System Design: Storage - DEV Community

Questions

1. How does block-level data access provide flexibility in managing storage resources?
2. How does file storage differ from block storage in terms of data access methods?
3. Explain how file locks work in a shared file system and their importance.
4. What advantages does object storage offer for handling large-scale unstructured data?
5. Provide examples of use cases where object storage is particularly beneficial.
6. What are the common RAID levels, and how do they offer different levels of redundancy and performance?
7. Discuss the trade-offs involved in selecting a specific RAID level for a given use case.
8. Give an example of a scenario where HDFS is well-suited, such as big data processing.

DAY 30-32

Message Queues, File Systems

📝 Topics to cover

- Role of Message Queues
- Kafka
- RabbitMQ
- Google File System(GFS)
- Hadoop Distributed File System (HDFS)

🗂️ Resource

1. System Design — Message Queues. Concepts and considerations for Message... | by Larry | Peng Yang | Computer Science Fundamentals | Medium
2. Message Queues in System Design
3. What is a Message Queue and Where is it used?
4. Apache Kafka - Introduction | Tutorialspoint
5. RabbitMQ vs Kafka - Difference Between Message Queue Systems - AWS

Questions

1. Explain how message queues can help in decoupling components and achieving asynchronous communication.
2. Describe a use case where message queues are crucial for maintaining data integrity.
3. How does Kafka ensure fault tolerance and data durability?
4. How do Kafka and RabbitMQ differ from each other?
5. How does GFS achieve fault tolerance and high availability for large-scale storage?
6. Define the Hadoop Distributed File System (HDFS) and its significance in big data processing.

DAY 33-35

Basics of Databases

📋 Topics to cover

- Relational databases
- Non-relational databases

🗂️ Resource

1. System Design — SQL vs NoSQL. Concepts and considerations for SQL and... | by Larry | Peng Yang | Computer Science Fundamentals | Medium
2. How To Choose The Right Database?
3. System Design Interview Prep: SQL vs NoSQL Databases - Exponent
4. Databases: system design interview concepts (2 of 9)

Questions

1. Explain the concept of normalization in the context of relational databases. What are the benefits and drawbacks of normalization?
2. Compare the differences between INNER JOIN, LEFT JOIN, and RIGHT JOIN operations in SQL.
3. Explain the trade-offs between using a relational database and a NoSQL database for a specific use case.
4. How does MongoDB differ from traditional relational databases in terms of data storage and querying?
5. Discuss the trade-offs between using a document-oriented database like MongoDB and a column-family database like Cassandra.
6. What are the benefits of denormalization in a document-oriented database like MongoDB?

DAY 36-38

Advanced Database Concepts

📝 Topics to cover

- ACID Properties
- Database sharding and partitioning
- Database indexing
- Concurrency Control

🗂️ Resource

1. ACID Properties in DBMS - GeeksforGeeks
2. Database Sharding vs. Partitioning: What's the Difference?
3. System Design — Sharding / Data Partitioning | by Larry | Peng Yang | Computer Science Fundamentals | Medium
4. An in-depth look at Database Indexing
5. Concurrency Control in DBMS - GeeksforGeeks

Questions

1. Explain the ACID properties in the context of database transactions. How do they ensure data integrity?
2. Discuss the trade-offs between ensuring strong ACID properties and achieving high performance in a database system.
3. Define database sharding and partitioning. How do they contribute to scalability?
4. What are the key factors to consider when selecting a sharding key for a database table?
5. Compare B-tree indexes and hash indexes. When would you choose one over the other?
6. What is a deadlock? Provide an example and describe strategies to prevent or resolve deadlocks.
7. Explain the concept of a database lock. How can locks be used to manage concurrent access?

DAY 38-40

Caching

📋 Topics to cover

- Different Types of Caching
- Cache Invalidation
- Cache Consistency
- Client side and Server side caching

🗂️ Resource

1. Caching - System Design Concept For Beginners - GeeksforGeeks
2. Introduction to Caching and Caching Strategies
3. System Design — Caching. Concepts and considerations for Caching... | by Larry | Peng Yang | Computer Science Fundamentals | Medium

Questions

1. Explain the difference between in-memory caching and content delivery networks (CDNs).
2. What is object caching, and in what scenarios would you use it?
3. Compare the FIFO (First In, First Out) and LFU (Least Frequently Used) cache replacement policies.
4. Explain a common approach to handle cache invalidation when data is updated.
5. What is cache consistency in distributed systems? How does it relate to strong consistency and eventual consistency?
6. Describe a scenario where dynamic data caching would be beneficial. How would you handle cache expiration for this type of data?
7. Discuss the trade-offs involved in using caching, including storage overhead and cache invalidation challenges.

DAY 41-43

API Contracts

📋 Topics to cover

- Request and Response Formats
- Request Parameters and Query Strings
- Response Data Structures
- Authentication and Authorization
- Rate Limiting and Throttling

🗂️ Resource

1. What is an API and how do you design it? 🗒️✅
2. APIs - Exponent

Questions

1. What is the significance of defining request and response formats in an API contract?
2. How would you handle versioning of request and response formats to ensure backward compatibility with older API clients?
3. Describe the purpose of request parameters and query strings in API requests. How do they affect the way clients interact with an API?
4. What role does OAuth play in enabling secure authentication and authorization in modern APIs?
5. Describe a scenario where rate limiting and throttling would be crucial to maintaining a reliable and secure API.

DAY 44-46

Containerization

📋 Topics to cover

- Containerization vs Virtualization
- Docker
- Kubernetes

📁 Resource

1. Containerization using Docker - GeeksforGeeks
2. Kubernetes - Concept of Containers - GeeksforGeeks
3. Containerization vs. Virtualization: 7 Technical Differences | Trianz
4. Containerization Explained

Questions

1. Describe a scenario where you would prefer to use containers over virtual machines.
2. Discuss the isolation levels provided by containers and virtual machines.
3. What is Docker, and how does it simplify the process of packaging and deploying applications?
4. What are Docker registries, and why are they important in the Docker ecosystem?
5. Explain the concept of a Kubernetes pod and its significance.
6. What is the purpose of a Kubernetes cluster, and how does it contribute to high availability?

DAY 46-48

Database Schema Design

📋 Topics to cover

- Selection of Database Schema Type
- Data Modelling
- Normalization and Denormalization
- Data Relationships

🗂️ Resource

1. Designing your Database Schema. Decide whether using a star or... | by Chloe Lubin | Towards Data Science
2. How to design database for a project
3. How to Design a Database Schema, With Examples | Zuar

Questions

1. Explain the concept of normalization and its advantages. When might you consider denormalization?
2. Compare and contrast one-to-one, one-to-many, and many-to-many relationships in database design.
3. Describe the purpose of primary keys and foreign keys in a database schema.
4. In a shopping website how do you handle the database schema design to accommodate varying product types, quantities, and orders?
5. How can you build and optimize the database schema for a social media platform to handle millions of user-generated posts and comments while ensuring quick retrieval?

DAY 48-50

Capacity Estimations

📋 Topics to cover

- Daily Active Users (DAU) and Monthly Active Users (MAU)
- Requests Per Second (RPS)
- Load and Traffic Patterns
- Storage Estimation
- Network Bandwidth Calculation
- Back of the Envelope Estimation

🗂️ Resource

1. Capacity Estimation in Systems Design | by Jeyabalaji Subramanian | Medium
2. Back-of-the-envelope Estimation

Questions

1. In a mobile app development project, how would you estimate the expected growth in DAU and MAU over the next six months?
2. How can you estimate the peak RPS for an e-commerce website during a major sale event?
3. In a real-life project involving video conferencing, how would you calculate the required network bandwidth to ensure smooth video and audio transmission for a specific number of participants?
4. Consider a messaging app. Estimate the RPS during a new year's countdown when users send messages simultaneously. Provide example inputs. Example Inputs: Total users = 100,000, Messages per user = 10, Countdown duration = 1 minute.
5. Imagine you're designing a fitness tracking app. How would you estimate the DAU and MAU for this app?

DAY 51-56

Study Architecture of Popular Systems

📝 Topics to cover

- Understand real-world systems to grasp practical application of system design principles.
- Examine how popular systems handle massive user loads without compromising performance
- Explore how data is organized, stored, and retrieved efficiently
- Study how load is distributed across multiple servers to avoid bottlenecks.
- Understand how microservices architecture contributes to flexibility and maintainability

📁 Resource

Masterclasses on System Design

DAY 57-60

Mock Interviews and Final Review

📝 Topics to cover

- Solve system design scenarios to hone problem-solving skills.
- Receive feedback on approach, communication, and technical depth.
- Review fundamental concepts and design patterns.
- Analyze your answers from the interviewer's viewpoint.
- Be aware of recent trends in system design.

BOSSCODER ACADEMY

WHY BOSSCODER?

750+ Alumni placed at Top Product-based companies.

More than **136% hike** for every **2 out of 3** working professional.

Average package of **24LPA.**

The syllabus is most **up-to-date** and the list of problems provided covers **all important topics.**

Lavanya
Meta

Course is very well **structured** and **streamlined** to crack any **MAANG** company

Rahul .
Google

EXPLORE MORE