Learning Roadmap: System Design

Duration: 12 Weeks | **Level:** Intermediate | **Weekly Effort:** 21 hours

Prerequisites

- Basic understanding of programming (any language)
- Familiarity with Data Structures and Algorithms
- Basic knowledge of computer networks and operating systems
- Understanding of web basics (HTTP, client-server model)

Week 1: Fundamentals of Distributed Systems & Core Concepts

Effort: 3 hours/day

■ Topics

- What is System Design?
- Key Principles: Scalability, Availability, Reliability, Efficiency, Maintainability
- CAP Theorem
- ACID vs. BASE Properties

- Latency vs. Throughput

- Introduction to Distributed Systems

■ YouTube Resources

Video	Channel	Duration	Link
System Design Interview - What is it? (Beginner)	Gaurav Sen	15 mins	Watch
CAP Theorem Explained	Hussein Nasser	20 mins	Watch
Scalability, Availability, Reliability, Maintainability in System Design	ByteByteGo	18 mins	Watch

■ Books

Title	Author	Focus
Designing Data-Intensive Applications	Martin Kleppmann	Foundational concepts of data systems, reliability, scalability, maintainability

■ Goals

- [ ] Understand the core definitions and principles of System Design.

- [ ] Differentiate between CAP Theorem, ACID, and BASE properties.

- [ ] Grasp the importance of scalability, availability, and reliability.

■■ Projects

Conceptual Design: URL Shortener: Design a basic URL shortener service conceptually, focusing on how it would store URLs and redirect requests.

Week 2: Core Components & Network Fundamentals

Effort: 3 hours/day

■ Topics

- Load Balancers (L4 vs. L7, types)
- Proxies (Forward, Reverse)
- Caching (CDN, Application-level, Database-level)
- DNS and Domain Management
- HTTP/HTTPS fundamentals

■ YouTube Resources

Video	Channel	Duration	Link
Load Balancer Explained	Tech Dummies	17 mins	Watch
What is a Proxy Server?	Simplilearn	10 mins	Watch
Caching in System Design	System Design Interview	25 mins	Watch

■ Books

Title	Author	Focus
System Design Interview – An insider's guide	Alex Xu	Practical examples and patterns for common system design components

■ Goals

- [ ] Understand the role and types of Load Balancers.

- [ ] Differentiate between various caching strategies.

- [ ] Explain how DNS works and its importance in distributed systems.

■■ Projects

Enhance URL Shortener with Load Balancing & Caching: Expand the conceptual design of the URL shortener to include a load balancer for distributing traffic and a caching layer for popular URLs.

Week 3: Data Storage: Databases & Scaling

Effort: 3 hours/day

■ Topics

- SQL vs. NoSQL Databases (Relational, Key-Value, Document, Columnar, Graph)

- Database Sharding (Horizontal Partitioning)

- Database Replication (Master-Slave, Master-Master)

- Indexing strategies

- Distributed File Systems (HDFS, S3 concepts)

■ YouTube Resources

Video	Channel	Duration	Link
SQL vs NoSQL Databases	Fireship	7 mins	Watch
Database Sharding Explained	ByteByteGo	12 mins	Watch
Database Replication Explained	Hussein Nasser	28 mins	Watch

■ Books

Title	Author	Focus
Database System Concepts	Silberschatz, Korth, Sudarshan	Comprehensive understanding of database internals, indexing, and transaction management

■ Goals

- [ ] Choose appropriate database types for different use cases.
- [ ] Understand techniques for scaling databases (sharding, replication).
- [ ] Explain how indexing improves query performance.

■■ Projects

Design a Distributed Key-Value Store: Design a simplified distributed key-value store like Amazon DynamoDB or Apache Cassandra. Focus on data distribution and replication.

Week 4: Asynchronous Processing & Messaging Systems

Effort: 3 hours/day

■ Topics

- Message Queues (Kafka, RabbitMQ concepts)

- Publish-Subscribe (Pub/Sub) Model

- Event-Driven Architecture

- Idempotency

- Background Jobs and Task Queues

■ YouTube Resources

Video	Channel	Duration	Link
Message Queues Explained	ByteByteGo	10 mins	Watch
Kafka Explained	Academind	18 mins	Watch
Event-Driven Architecture Explained	AWS	15 mins	Watch

■ Books

Title	Author	Focus
Building Microservices	Sam Newman	Designing and implementing microservices, including communication patterns like messaging

■ Goals

- [ ] Understand the benefits of asynchronous communication.

- [ ] Differentiate between various messaging patterns (queues, pub/sub).

- [ ] Design systems that handle background tasks and reliable event processing.

■■ Projects

User Notification Service with Message Queue: Design a system for sending notifications (email, SMS) to users. Integrate a message queue to handle notification requests asynchronously.

Week 5: Microservices & API Design

Effort: 3 hours/day

■ Topics

- Monolithic vs. Microservices Architecture
- API Gateway
- Service Discovery
- RESTful API Design Principles
- Introduction to GraphQL
- Inter-service communication (RPC, REST)

■ YouTube Resources

Video	Channel	Duration	Link
Microservices Explained	Google Cloud Tech	12 mins	Watch
REST API Design Best Practices	freeCodeCamp.org	20 mins	Watch
What is an API Gateway?	Kong	8 mins	Watch

■ Books

Title	Author	Focus
Microservices Patterns	Chris Richardson	Detailed patterns for building and deploying microservices architectures

■ Goals

- [ ] Decide when to use microservices vs. monolithic architecture.

- [ ] Design robust and maintainable RESTful APIs.

- [ ] Understand the role of API Gateways and Service Discovery.

■■ Projects

Design a Simplified E-commerce Platform (Microservices): Design an e-commerce platform with distinct services for products, orders, and users. Focus on service boundaries and inter-service communication.

Week 6: Security, Monitoring & Observability

Effort: 3 hours/day

■ Topics

- Authentication (OAuth, JWT)

- Authorization (RBAC)

- Rate Limiting & Throttling
- DDoS Protection
- Monitoring (Metrics, Alerts, Dashboards)
- Logging (ELK stack concepts)
- Tracing (Distributed Tracing, OpenTelemetry concepts)

■ YouTube Resources

Video	Channel	Duration	Link
OAuth 2.0 and OpenID Connect (in plain English)	OktaDev	10 mins	Watch
Rate Limiting Explained	ByteByteGo	11 mins	Watch
Observability vs. Monitoring	Grafana Labs	7 mins	Watch

■ Books

Title	Author	Focus
The Phoenix Project	Gene Kim, Kevin Behr, George Spafford	Understanding DevOps, IT operations, and the importance of monitoring and flow in IT

■ Goals

- [ ] Implement secure authentication and authorization mechanisms.
- [ ] Design systems with rate limiting and DDoS protection.
- [ ] Set up comprehensive monitoring, logging, and tracing for distributed applications.

■■ Projects

Add Security & Monitoring to E-commerce Platform: Integrate authentication (e.g., JWT) and rate limiting into the e-commerce platform. Outline a monitoring strategy for key metrics.

Week 7: Advanced Scalability & Performance Patterns

Effort: 3 hours/day

■ Topics

- Concurrency vs. Parallelism
- Distributed Transactions (2PC, Saga Pattern concepts)
- Circuit Breaker Pattern
- Bulkhead Pattern
- Backpressure
- Leader Election Algorithms (Raft/Paxos concepts)

■ YouTube Resources

Video	Channel	Duration	Link
Circuit Breaker Pattern Explained	ByteByteGo	9 mins	Watch
Distributed Transactions: Two-Phase Commit	Hussein Nasser	25 mins	Watch

What is Backpressure?	NDC Conferences	15 mins	Watch

■ Books

Title	Author	Focus
Release It! Design and Deploy Production-Ready Software	Michael T. Nygard	Resilience patterns like circuit breakers, bulkheads, and stability strategies

■ Goals

- [ ] Apply patterns for fault tolerance and resilience in distributed systems.
- [ ] Understand the complexities of distributed transactions.
- [ ] Design systems that can gracefully handle high load and failures.

■■ Projects

Design a Resilient Payment Gateway: Design a payment gateway that integrates with multiple external payment providers. Focus on resilience patterns like Circuit Breaker and Saga for handling failures and distributed transactions.

Week 8: Real-time Systems & Streaming

Effort: 3 hours/day

■ Topics

- WebSockets vs. Server-Sent Events (SSE)

- Long Polling

- Real-time Analytics

- Stream Processing (Apache Flink, Spark Streaming concepts)

- Change Data Capture (CDC)

■ YouTube Resources

Video	Channel	Duration	Link
WebSockets in 100 Seconds	Fireship	2 mins	Watch
What is Stream Processing?	Confluent	5 mins	Watch
Building Real-time Applications	Google Cloud Tech	15 mins	Watch

■ Books

Title	Author	Focus
Kafka: The Definitive Guide	Gwen Shapira, Neha Narkhede, Todd Palino	Deep dive into Kafka for building real-time data pipelines and streaming applications

■ Goals

- [ ] Choose appropriate technologies for real-time communication.

- [ ] Design systems for processing data streams in real-time.

- [ ] Understand use cases for real-time analytics and data updates.

■■ Projects

Design a Real-time Collaborative Editor (like Google Docs): Design a system that allows multiple users to edit a document simultaneously in real-time.

Week 9: System Design Interview Prep: Common Problems (Part 1)

Effort: 3 hours/day

■ Topics

- Designing a Twitter/Facebook News Feed
- Designing a URL Shortener (full scale)
- Designing a Distributed Cache
- Designing a Chat Application
- System Design Interview Frameworks (clarifying requirements, estimations, deep dive)

■ YouTube Resources

Video	Channel	Duration	Link
Design Twitter (System Design Interview)	Gaurav Sen	40 mins	Watch
Design a URL Shortener (System Design Interview)	Tech Dummies	35 mins	Watch
System Design Interview Prep - The Framework	Exponent	15 mins	Watch

■ Books

Title	Author	Focus
System Design Interview – An insider's guide Vol. 2	Alex Xu	More advanced and complex system design interview questions with detailed solutions

■ Goals

- [ ] Apply learned concepts to solve common system design interview problems.

- [ ] Practice breaking down complex problems into manageable components.

- [ ] Articulate trade-offs and design choices effectively.

■■ Projects

Design a Twitter Timeline/News Feed: Design the backend system for a Twitter-like timeline, focusing on fan-out strategies (push vs. pull) and data storage.

Week 10: System Design Interview Prep: Complex Problems (Part 2)

Effort: 3 hours/day

■ Topics

- Designing a Distributed Web Crawler

- Designing a Global Ride-Sharing Service (Uber/Lyft)

- Designing a Recommendation System

- Designing a Type-Ahead Suggestion System

- Estimating scale, storage, and bandwidth requirements

■ YouTube Resources

Video	Channel	Duration	Link
Design Uber (System Design Interview)	Gaurav Sen	45 mins	Watch
Design a Recommendation System	System Design Interview	30 mins	Watch
Design a Distributed Web Crawler	ByteByteGo	18 mins	Watch

■ Books

Title	Author	Focus
Grokking the System Design Interview	Design Gurus	Problem-solving techniques and detailed solutions for common system design interview questions

■ Goals

- [ ] Solve more complex and open-ended system design problems.

- [ ] Improve ability to make reasonable estimations for system resources.

- [ ] Refine communication skills for explaining design choices and trade-offs.

■■ Projects

Design a Distributed Web Crawler: Design a system to crawl websites, extract links, and store pages. Focus on parallelism, politeness, and handling duplicate URLs.

Week 11: Cloud-Native & DevOps Concepts

Effort: 3 hours/day

■ Topics

- Containerization (Docker)
- Container Orchestration (Kubernetes basics)
- Serverless Computing (AWS Lambda, Azure Functions concepts)
- Continuous Integration/Continuous Deployment (CI/CD)
- Infrastructure as Code (Terraform, CloudFormation concepts)

■ YouTube Resources

Video	Channel	Duration	Link
Docker in 100 Seconds	Fireship	2 mins	Watch
Kubernetes Explained	IBM Technology	11 mins	Watch
CI/CD Explained	Google Cloud Tech	10 mins	Watch

■ Books

Title	Author	Focus
The DevOps Handbook	Gene Kim, Jez Humble, Patrick Debois, John Willis	Principles and practices for implementing DevOps, including CI/CD and infrastructure automation

■ Goals

- [ ] Understand the benefits of containerization and orchestration.

- [ ] Explore serverless architectures and their use cases.

- [ ] Grasp the importance of CI/CD and Infrastructure as Code for modern deployments.

■■ Projects

Deploy a Microservice with Docker & Basic CI/CD: Design a simple REST API microservice and outline how it would be containerized with Docker and deployed using a basic CI/CD pipeline.

Week 12: Review & Advanced Topics/Future Trends

Effort: 3 hours/day

■ Topics

- Consensus Algorithms (Raft, Paxos - deeper dive)

- Blockchain basics (Distributed Ledger Technology)

- Edge Computing concepts

- Quantum Computing implications (brief overview)

- Review of all major system design concepts

■ YouTube Resources

Video	Channel	Duration	Link
Raft Consensus Algorithm Explained	The Secret Life of Data	15 mins	Watch
Blockchain Explained	Simply Explained	8 mins	Watch
What is Edge Computing?	IBM Technology	7 mins	Watch

■ Books

Title	Author	Focus
Distributed Systems: Concepts and Design	George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair	Comprehensive academic treatment of distributed systems, including advanced topics like consensus

■ Goals

- [ ] Solidify understanding of all fundamental and advanced system design concepts.

- [ ] Gain exposure to cutting-edge technologies and future trends.

- [ ] Be able to articulate a complete system design from requirements to deployment considerations.

■■ Projects

Conceptual Design: Decentralized Application (DApp) / Blockchain Component: Design a high-level architecture for a simple decentralized application or a component that leverages blockchain principles (e.g., a simple voting system or a distributed ledger for asset tracking).

Skills Acquired

- Ability to design scalable, available, and reliable distributed systems

- Proficiency in selecting appropriate data storage solutions (SQL, NoSQL, distributed file systems)

- Understanding of caching strategies and load balancing techniques
- Expertise in asynchronous communication patterns and messaging systems
- Knowledge of microservices architecture and API design best practices
- Skills in incorporating security (AuthN/AuthZ, Rate Limiting) and observability (Monitoring, Logging, Tracing)
- Familiarity with advanced resilience patterns (Circuit Breaker, Saga)
- Capability to design real-time systems using WebSockets and stream processing
- Strong problem-solving skills for system design interviews
- Basic understanding of cloud-native technologies (Docker, Kubernetes, Serverless) and DevOps principles

Next Steps

- Practice more system design interview questions on platforms like LeetCode, InterviewBit, or dedicated system design courses.
- Deep dive into a specific cloud provider (AWS, Azure, GCP) and get hands-on experience deploying designed systems.
- Contribute to open-source projects or build personal projects applying system design principles.
- Explore specialized areas like Big Data engineering, Machine Learning system design, or SRE (Site Reliability Engineering).