System Design Group Workbook

20-Minute Architecture Design Exercise

Objective

In 20 minutes, your group will design a scalable system from requirements to architecture and trade-offs. You may choose any domain (e-commerce, fintech, healthcare, AI assistant, etc.).

Activity Flow (20 Minutes)

- 3 minutes — Choose system and define requirements
- 5 minutes — Design high-level architecture
- 4 minutes — Define scalability and database strategy
- 4 minutes — Communication, CAP, and trade-offs
- 4 minutes — Optional: Extend with GenAI layer

PART 1: Select Your System (3 Minutes)

Group Name:

Chosen Use Case:

(E-commerce app, Food delivery, Video streaming, Social media, Banking app, EdTech platform, AI assistant, etc.)

Target Users:

Primary Business Goal:

PART 2: Requirements Definition (5 Minutes)

A. Functional Requirements

What must the system do? List at least five.

1.
2.
3.
4.
5.

B. Non-Functional Requirements

Choose what matters most:

- High availability
- Low latency
- High throughput
- Strong consistency
- Eventual consistency
- Fault tolerance
- Global scalability

Add two custom requirements:

1.
2.

C. Prioritization

Which is more important for your system?

Performance or Consistency?

Availability or Accuracy?

PART 3: Scale Estimation (3 Minutes)

Estimate realistically.

Daily active users: _____
Peak concurrent users: _____
Requests per second (peak): _____
Data stored per year: _____
Expected growth rate: _____

PART 4: High-Level Architecture (5 Minutes)

Draw your system architecture below:

User → _____ → _____ → _____ → Database → Response

Components You Plan to Use

- Load balancer
- API gateway
- Application servers
- Cache (Redis or similar)

- SQL database
- NoSQL database
- CDN
- Message queue
- Microservices
- Monolith

Architecture Style

Monolithic or Microservices?

Justification:

PART 5: CAP Theorem Decision (2 Minutes)

You can choose only two:

- Consistency
- Availability
- Partition tolerance

Chosen combination:

Why?

PART 6: Scalability Strategy (2 Minutes)

Scaling Approach

Vertical scaling or Horizontal scaling?

Where will you scale first?

Caching Strategy

What will you cache?

Eviction strategy (LRU, TTL, LFU, other):

PART 7: Database Strategy (2 Minutes)

SQL, NoSQL, or Polyglot persistence?

Why?

Will you use sharding?
Yes / No

If yes, what is the sharding key?

PART 8: Communication Design (2 Minutes)

Choose one:

- Synchronous (REST/gRPC)
- Asynchronous (Message queue/events)
- Hybrid

Why?

Will you use event streaming?

PART 9: GenAI Extension (Optional – 4 Minutes)

Now extend your system with an AI capability.

Example ideas:

- E-commerce → AI product advisor
- EdTech → AI tutor
- Banking → AI fraud detection assistant
- Healthcare → AI triage assistant

What problem will AI solve?

Will you use Retrieval-Augmented Generation (RAG)?
Yes / No

If yes:

- What documents will be embedded?

- Where will they be stored?

New Risks Introduced

Latency
Cost
Hallucination
Security
Token limits

Explain one key risk:

PART 10: Trade-Off Reflection (Final 3 Minutes)

Complete the sentence:

“Our system prioritizes _____ over _____ because…”

Biggest bottleneck in your design:

How would you improve this in version 2?

Final 1-Minute Presentation Structure

Each group must present:

1. The chosen system

2. Architecture type
3. One scaling decision
4. One major trade-off
5. AI extension (if added)