System Design Template Cheat Sheet

by vinu76jsr via cheatography.com/163274/cs/34196/

0. Non-functional Requirements (NFRs)

Security	Capacity
Compatibility (OS, etc)	reliability
Availability	Maintainable
Manageable	Scalability
Usability	Performance
Regulatory	Environmental

2. Estimations

Throughput (QPS, read/write)	Latency
R/W ration	Traffic Estimates - write(QPS, volume of data) and read (QPS volumen of data
Storage Estimates	Memory Estimates (Cache, what kind of data in it)
RAM	SSD or disk

Features and Expectation

Use case	Scenarios to not cover
who will use	How many will use
Usage patterns	

3. Design Goals

Latency / throughput requests

Consistency vs Availability

- weak / strong or eventual consistency
- failover / replication availability

4. High Level Design

APIs for read write scenario	DB schema
Basic algo	HLD for Read/write scenario

5. Deep Dive

Deep Dive
- scaling algo
- scaling components
- Availability, consistency and scale story of each component
- consistency and availability patterns
- Components
-DNS - CDN (push vs pull)
- LB (active passive, active active, L4 , L7)
- Reverse proxy
- App layers scaling (microservices, service discovery)
- DB (RDBMS, NoSQL)
- RDBMS (master-slave, Master, master, federarion, sharding, Denormalization, SQL tuning)
- NoSQL (KV< wide column, graph, document
- fast lookups
- RAM (redis, memcache
- AP - Cassandra, RIAK
- CP - HBase, Mongo
- Caches
- client, CDN, server, DB, applicaiton, query, object