**System Design Spring 2025**

**Assignment 1**

**Social network - VK (vk.com):**

**Why do we need this system.**

Platform that allows users to connect, share, and communicate, group discussions and events.

**Functional requirements.**

* Registration: sign up/log in
* Manage profile: users can manage personal information
* Content: users can create, like, comment on, and share posts
* Messaging system: provide private and group chats with multimedia support
* Search: enable users to search for other users, groups, or content
* Group management: support creation and management of groups.

**Business metrics.**

1. 50 million active users/day
2. 10-20 actions/session (loading feeds, sending message etc)

**Non-functional requirements.**

1. RPS

There are 50 million active users with 2 sessions/day, then:

RPS(read) = (50\*10^6\*20\*2)/(86400)=24.000 RPS – may scale to 100.000 for peak periods

RPS(write) = 2.000RPS - may scale to 15.000 RPS for peak periods

1. Storage

10 uploads/day – 1 user => 10\*50 million = 500 million/day – uploads/day

**Let’s** assume 1Mb – avarage file size of upload(photo, post etc) => Storage growth per day = 500 million \* 1 Mb = 500 Tb. Yealy growth ~1Pb accounting archival + 3x replication.

1. Network traffic and latency.

Average session ~ 10 Mb

Data traffic = 50 million \* 10 Mb = 500Tb/day

Let’s take peak period – 10Tb/s

Latancy: ~200ms – standart

1. 128–512 GB RAM and 32–64 cores per server
2. Number of connections/servers.

Assume 100 million concurrent users at peak. Each server handle ~2000 active connections => Servers need = 100 million/ 2000 = 50.000

**What is more and less important. (with CAP theorem)**

**AP. Since it must remain operational even when parts of the network fail and it cannot afford downtime (e.g. massive global level base)  
C – not critical if user will see on other user’s post** 541 like, instead 546…

**Social network - linkedin (linkedin.com)**

**Why do we need this system.**

It is a professional networking platform enabling users to connect, share, and explore career opportunities.

**Functional requirements.**

* Registration: sign up/log in
* Manage profile: users can manage personal information (education, experience)
* Job search and recruitment: support job postings, applications, and recruiter tools
* Content: users can create, like, comment on, and share posts
* Messaging system: provide private chats
* Search: advanced filters for profiles, jobs, and content

**Business metrics.**

Support 950+ million users globally

DAU = ~570 million users/day

**Non-functional requirements.**

1. RPS

Daily Reads =570 \* 10^6 users \* 20 reads/session \* 2 sessions = 22.8 billion reads/day

RPS(read) = **264.000** RPS

Peak RPS = RPS \* 3 ~ 800.000 RPS(peak)

RPS(write) - let’s take 10% from RPS(read)

1. Storage

Annual User Data (profile photo, posts etc) = 950 million \* 10MB = 9.5TB/year.

Job postings = 10 million/year, each ~1mb

1. Network traffic.

Assume 50% of user actions involve multimedia =>

Daily traffic = 0.5 \* 23 billion \* 5 = 57 Pb/day

Peak bandwidth = 661 Gbps

1. RAM and CPU

Let’s assume that LinkedIn use in-memory caching for feeds, search result etc

And assume that 10Kb/session – 1 user

Daily cache data = 570 million \* 2 sessions \* 10Kb = 12 Tb/day

With replication: 34 Tb.

Let’s take 256 GB => We need +- 200 servers, запас + 34Tb/256Gb

Estimate 64 cores/server to handle RPS. Each server handles 1,000–2,000 RPS; for peak traffic:

Servers required (Peak) ​= 792,000/2000​ ~ 400servers.

**What is more and less important. (with CAP theorem)**

Same as in Vkontakte case

**Video hosting / streaming - YouTube (youtube.com)**

**Why do we need this system.**

YouTube is a video-sharing platform that allows users to upload, share, and consume video content.

**Functional requirements.**

1. Video uploading and management (edit video content with metadata (titles, tags, descriptions))
2. Content streaming: support adaptive video streaming for smooth playback and varying bandwidths
3. Search/discovery
4. Manage user accounts (profile, subscriptions, histories)
5. Comments and like videos

**Business metrics.**

1. DAU ~ 1.5 billion, + 100 billion/year
2. 1/1000 - uploader (1,5 million uploader/day)
3. 1 video in 2 days(avg) (750.000 videos/day)
4. Visitors spend an average of 20 min/day (to watch)

**Non-functional requirements.**

1. Capacity

20 mins -> 500 Mb (avg. video) -> 750.000 \* 500Mb => 400 Tb/day

Total storage (yearly) = 365 \* 3 \* 400 Tb = **400 Pb**

SSD

1. Should be available -> 99% - uptime
2. Hold 500 millions connections concurrently
3. Network Traffic and Latency:

Latancy: < **200ms** – for search result

< **500ms** – for video start

Bandwidth:

1 min ~ 20 Mb => Let’s assume 1/3 daily user watch concurrently => 500 millions user \* (1/3)Mb/s = **170 Tbps**

**What is more and less important. (with CAP theorem)**

**Like in Vk and Linkedin, PA – is important, S – not, since slight delays in sync comments or likes - acceptable**

**URL shortener – bitly (bitly.com)**

**Why do we need this system?**

It is designed to make long, complex URLs shorter and easier to share

**Functional metrics.**

1. URL Shortening (hashing)
2. Redirection (302)
3. Link management
4. Custom short urls
5. Analytics and tracking (clicks, locations)
6. Link management

**Business metrics.**

1. Uptime – 99.99%
2. 10-20% growth/year
3. 1 billion active links globally (+ 5 million/day)

**Non-functional requirements.**

1. RPS

Daily reads = 1 billion \* 10 clicks = 10 billion reads/day

~ **120.000 RPS**

RPS(write) ~ 58 – do not count

1. Storage

Let’s take 1Kb per link then

Total storage for URLs = 1 billion \* 1Kb = **1 Tb (+2 Tb/year)**

Analytics storage = 1 billion URLs \* 10Kb = **10 Tb (+ 20Tb/year)**

1. Network traffic and latency

Assume 10 Kb per redirection, then:

10 billion read/day \* 10 Kb -> 100 Tb/day

Bandwidth ~ 11.6 Gbps

**Handle 1 Tbps**

**Latancy < 50 ms - redirection**

**What is more and less important? (with CAP theorem)**

P – critical, since it is global service.

A – important, since URL redirection is time-sensitive

C – low priority, since slight delays in analytics updates or propagation of newly created links are acceptable.

**Student portal – wsp (wsp.kbtu.kz)**

**Why do we need this system?**

WSP is essential for managing academic and administrative tasks within a university

**Functional requirements.**

A. Authorization

B. Registration on disciplines (student)

C. View schedule (student / teacher )

D. Requests (student / teacher)

E. Journal (student / teacher )

F. Roles ( students, teacher )

**Business metrics.**

A. Users (students / teachers). ~5400, +600 users/year

B. Few minutes per week;

C. High peak load, low in Q99 (2000)

**Non-functional requirements.**

A. < 200ms; ok for registration ~500 ms

B. Consistency in data, atomicity in operations (registration / requests)

C. 250 GB x3 HDD

D. 16 GB RAM for server

E. 2000 Connections

F. 8 vCPU

**What is more and less important? (with CAP theorem)**

C and A – is important, since we need immediate accurate updates for key data

P – not, since we based in one place

**Social media – Instagram**

**Why do we need this system.**

It is a social media platform designed to enable users to share visual content, connect with others, and engage in digital storytelling

**Functional requirements.**

1. Account management (registration, profile management)
2. Photo and video uploading
3. Feed and content consumption
4. Search and discovery
5. Messaging

**Business metrics.**

1. DAU ~ 1 billion
2. Average session length (user): ~30 minutes/day

**Non-functional requirements.**

1. Feed loading – < 100 ms

Feed upload – < 2000 ms

1. Uptime – 99.99%
2. Estimated 600 PB/year (5 Mb \* 365 \* 10^8 + some growth from new user) growth for multimedia content with 3x replication

**What is more and less important? (with CAP theorem)**

P and A – is important, since (1) Instagram is worldwide social network and (2) user itteraction should be available all time

C – not important, since does not matter if some one will see 81 like, instead 83

**Messenger – telegram**

**Why do we need this system.**

It is a messaging platform that provides secure, fast, and scalable communication

**Functional requirements.**

1. Account management (registration, auth, profile management)
2. Text messaging and multimedia sharing (user can upload file up to 2 Gb)
3. Group chats and channels
4. Voice and video calls
5. End-to-end encryption
6. Search and discovery

**Business metrics.**

1. DAU - 500 million
2. Average Messages (user): ~ 50 messages/day

**Non-functional requirements.**

1. < 200 ms – for text message
2. HDD: 10 Eb(with 3x replication, because we get data only when sent sync data on smartphones)
3. CPU: 32–64
4. RAM: 128–512 GB

**What is more and less important. (with CAP theorem)**

P – is critical, telegram operates globally

A – is high, users expect instant messaging and delivery, even during high traffic

C – is low, slight inconsistencies (message appearing on one device slightly later than another) are tolerable

**Banking – Halyk Bank**

**Why do we need this system.**

Halyk Bank’s digital system is essential for managing and delivering banking services to individual and corporate clients.

**Functional requirements.**

1. User authentication and authorization
2. Account management:
3. open all types of accounts like savings, fixed
4. deposit, etc.

* customers can open all types of accounts like savings, fixed deposit, etc
* view their transaction history and their available balance
* they will be able to transfer funds between their own accounts or to others accounts

1. open all types of accounts like savings, fixed
2. deposit
3. open all types of accounts like savings, fixed
4. deposit
5. open all types of accounts like savings, fixed
6. deposit
7. open all types of accounts like savings, fixed
8. deposit, etc.
9. open all types of accounts like savings, fixed
10. deposit, etc.
11. open all types of accounts like savings, fixed
12. deposit, etc.
13. open all types of accounts like savings, fixed
14. deposit, etc.
15. open all types of accounts like savings, fixed
16. deposit, etc.
17. open all types of accounts like savings, fixed
18. deposit, etc.
19. open all types of accounts like savings, fixed
20. deposit, etc.

**Business metrics.**

**Non-functional requirements.**

**What is more and less important. (with CAP theorem)**

**Banking - Kaspi Bank**

**Why do we need this system.**

**Functional requirements.**

**Business metrics.**

**Non-functional requirements.**

**What is more and less important. (with CAP theorem)**

**Taxi – XXX (analog Yandex Taxi)**

**Why do we need this system.**

Connects passengers with drivers in real-time, providing convenient and efficient means of transportation

**Functional requirements.**

1. User account management
2. Ride booking and matching

Allow users to book rides by providing pickup and drop-off locations.

Match riders with the nearest available driver.

1. Real-time tracking (GPS-based tracking of drivers and trips)
2. Dynamic pricing
3. Payment

**Business metrics.**

1. DAU = 5 million
2. Peak demand during rush hours, weekends, and special events
3. Geographic Distribution - Kazakhstan

**Non-functional requirements.**

1. **Uptime – 99.99%**
2. **Low latency (≤ 1000 ms) for driver matching and route calculations**
3. **Real-time GPS tracking with updates ≤ 100 ms**

**What is more and less important. (with CAP theorem)**

**P – is not critical, since it work only in Kazakhstan**

**A – is critical**

**C – is important**