EXPERIMENT NO: 4

Student Name: Devjot Singh   **UID:** 23BCS10864

Branch: BE-CSE   **Section/Group:** KRG-2A

Semester: 6th   **Date of Performance:** 03/02/2026

Subject Name: System Design   **Subject Code:** 23CSH-314

Aim

To design a scalable OTT (Over-The-Top) platform similar to Netflix or Amazon Prime that allows users to register, subscribe to plans, search and stream movies and TV shows in multiple resolutions, and receive personalized recommendations by identifying functional and non-functional requirements and defining system APIs.

Objectives

- Understand the architecture of a large-scale OTT video streaming platform.
- Identify functional requirements such as user onboarding, subscriptions, search, and video playback.
- Identify non-functional requirements including scalability, availability, and low latency streaming.
- Analyze CAP theorem trade-offs in OTT platforms.
- Design RESTful APIs for subscription management and video streaming.

Procedure

1. Studied real-world OTT platforms such as Netflix and Amazon Prime.
2. Identified core entities including Client, Subscription, Video, Metadata, and Recommendation Engine.
3. Analyzed user flow from registration to video playback.
4. Collected functional and non-functional requirements.
5. Designed APIs for user onboarding, subscription handling, searching, and video streaming.
6. Analyzed scalability requirements for millions of concurrent users.
7. Studied availability and consistency trade-offs for payments and video playback.

Functional Requirements

1. Clients should be able to create an account on the OTT platform.
2. After successful login, clients should be able to choose and manage subscription plans.
3. Clients should be able to search for movies or TV shows by title or name.
4. Clients should be able to stream videos in multiple resolutions (480p, 720p, 1080p, 4K).
5. Clients should receive recommendations for movies and TV shows (Optional).

Core Entities of the System

- Client

- Subscription
- Video
- Video Metadata (Thumbnails, Description)
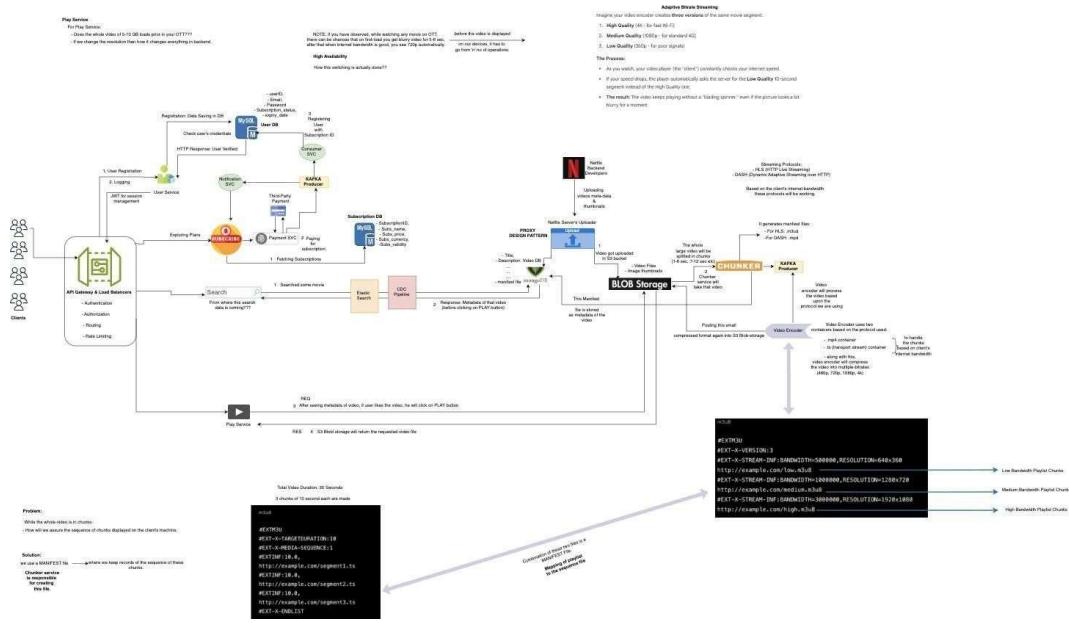- Recommendation Engine

API Design

1. **User Registration API**: `POST /user/register`
2. **User Login API**: `POST /user/login`
3. **Update User Profile API**: `PUT /user/update`

4. **Get Subscription Plans API**: `GET /get-subscription-plans`
5. **Subscribe to Plan API**: `POST /subscription`

6. **Search Video API**: `GET /search?q={movie_name}`
7. **Get Video Metadata API**: `GET /video/{video_id}`
8. **Play Video API**: `GET /play/{video_id}`

Non-Functional Requirements

1. Scalability to support 200–300 million users with approximately 20,000 videos.
2. High availability for video streaming services.
3. Availability over consistency for video playback, and strong consistency for payments and subscriptions.
4. Streaming latency between 50–80 ms with minimal buffering.
5. Horizontally scalable architecture using CDN and distributed systems.

High Level Design (HLD)

The system consists of Client Applications, API Gateway, User Service, Subscription Service, Search Service, Video Streaming Service, Recommendation Engine, CDN, Cache, Message Queue, and Distributed Databases. Video content is stored in object storage and delivered through CDN for low latency streaming.

Outcome

- Designed a scalable OTT video streaming platform.
- Identified functional and non-functional requirements.
- Designed RESTful APIs for user onboarding, subscriptions, search, and video playback.
- Understood scalability and CAP theorem trade-offs in OTT systems.