SRS DOCUMENT

For

Video Streaming Application

Team :

Mohammed Adnan Zaki
Naga Saketh V
Nandan N
Mohammed Zulqarnain

Table Of Contents :

1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to build an online video streaming platform with more emphasis on the quality of the service.

1.2 INTENDED AUDIENCE

This project is restricted within the college premises. This has been implemented under the guidance of college professors

1.3 PROJECT SCOPE

The project's scope involves creating a video streaming service ensuring smooth playback on various devices. It includes optimizing buffering time for different internet speeds, auto-adjusting video quality, maintaining consistent video and audio, responsive playback on diverse screens, handling internet connection loss, and appropriate response to corrupt video files. The goal is to deliver a user-friendly streaming service meeting these requirements.

1.4 REFERENCES

- https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

A centralized streaming service database system stores the following information.

- **Video and its Related Data:**
  It includes the BLOB content of the video along with its metadata such as date of upload, number of streams, likes & comments, size & quality.

- **User Accounts Information:**
  It includes the users' credentials, contact information for emergencies, list of videos watched, liked, uploaded, commented on and basic details like date of registration, last visit date.

2.2 PRODUCT FEATURES

- Check the compatibility with the browser and the device.
- Video buffering optimization based on internet speed
- Adaptive video quality adjustment during streaming
- Ensured high video quality without distortion or pixelation
- Ensure uninterrupted audio playback, preserving the clarity and continuity of the video's audio track throughout the streaming process, even in conditions of fluctuating internet speeds.
- Responsive video playback adapting to screen sizes
- Dynamic handling of video buffering in case of internet loss
- Exception raised by the streaming player when the user tries to stream a corrupt video file.

2.3 OPERATING ENVIRONMENT

The operating environment for the streaming service is as listed below:

- Centralised database
- Client/server system
- Database: Supabase
- Platform: HTML&CSS/Python/Django

2.4 DEPENDENCIES

1. Functional Dependency - Internet Connectivity:
   - Dependency: Availability of a stable and reliable internet connection.
   - Description: The streaming service functionality depends on users having access to a stable and consistent internet connection to stream video content without interruptions.

2. Functional Dependency - Modern Web Browser Compatibility:
   - Dependency: Availability of a modern web browser compatible with Django, HTML, and CSS.
   - Description: Users need a modern web browser that supports Django, HTML, and CSS to access and view the video streaming service seamlessly.

3. Functional Dependency - Adequate System Resources:
   - Dependency: Availability of a system with sufficient RAM and processing power to handle streaming content.
   - Description: The system must meet minimum hardware requirements, including adequate RAM and processing power, to ensure smooth playback and display of the streaming content.

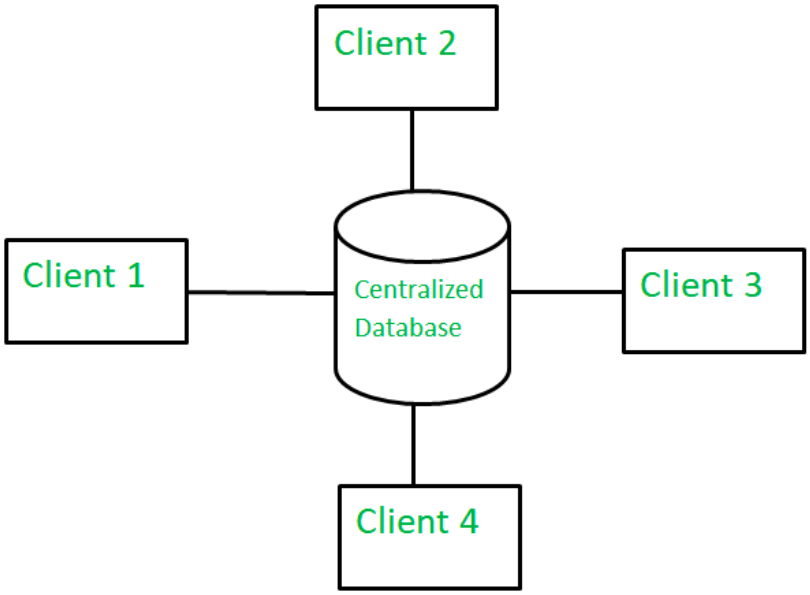
3. SYSTEM FEATURES

3.1 Description

Users can register for an account or login to existing ones. They can search for a particular video and interact with the videos such as viewing and liking. They can adjust the quality of the data stream to their liking or allow the service to handle it.

3.2 Functional Requirements

CENTRALISED DATABASE: A centralized database, represents a data management model where all the data is stored in a single, unified repository. In this configuration, a singular application interacts with and manages all its data from one central database location, as depicted in the diagram below.



CLIENT/SERVER SYSTEM: The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACES

- Front-end software: HTML & CSS
- Back-end software: Python, Django
- Database – Supabase

4.2 COMMUNICATION INTERFACES

- A browser that supports HTML5, CSS3, Javascript, HTTPS and WebRTC.

5. NONFUNCTIONAL REQUIREMENTS

5.1 Safety REQUIREMENTS

If there is damage to a video file on the database due to catastrophic failure, such as a disk crash, the recovery method restores the file from a copy of the file that was saved to a different server during the time of upload.

5.2 Software Quality Attributes

- Reliability:
  - Ensure the system's stability, availability, and robustness, minimizing downtime and service disruptions during video streaming.

- Usability:
  - Design an intuitive user interface, enabling easy navigation and providing clear feedback to users regarding the streaming process and controls.

- Maintainability:
  - Design the codebase in a modular and well-structured manner, making it easy to maintain, update, and extend in the future.

- Compatibility:
  - Ensure compatibility across various devices, browsers, and operating systems to reach a broader audience.

- Adaptability:
  - Make the system adaptable to different network conditions, adjusting video quality and streaming parameters accordingly.

- Fault Tolerance:
  - Design the system to gracefully handle errors and unexpected failures,providing appropriate error messages to users.

- Data Integrity:
  - Ensure the integrity of video files and user data, preventing corruption or unauthorised modifications.