Software Design Document

OTT Platform

Version 1.0

Printed: October 24th, 2023

Developer Team:

Adithya T G, Amara Sai Prasad,
Aaryan Shetty, Adithya Mahesh

Supervisor: Prof. M S Anand

PES University

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to define the specifications and requirements for the development of the Over-The-Top (OTT) platform. It serves as a comprehensive guide for the project stakeholders, including developers, designers and project managers to understand the goals, functionalities and constraints associated with the OTT platform.

1.2 Scope

The scope of this project encompasses the design, development, and deployment of a fully functional OTT platform. This platform will provide users with access to a vast library of video content, including movies, TV shows, and documentaries. It will feature personalised recommendations, user profiles, high-definition streaming, and multi-device compatibility. The platform will be accessible through both web and mobile applications, catering to a diverse audience across various demographics and regions.

1.3 Definitions, Acronyms and Abbreviations

Acronym	Meaning
OTT	Over-The-Top
UI	User Interface
API	Application Programming Interface
CDN	Content Delivery Network
GPU	Graphical Processing Unit
SSD	Solid State Drive
2FA	Two- Factor Authentication

1.4 References

- Industry standards for video encoding, streaming protocols and content delivery.
- Netflix Documentation
- AWS (Amazon Web Services) documentation for cloud-based server infrastructure.
- Payment gateway API documentation for secure payment processing.
- Regulatory authorities' guidelines for compliance reporting.
- External content providers' licensing agreements for content integration.
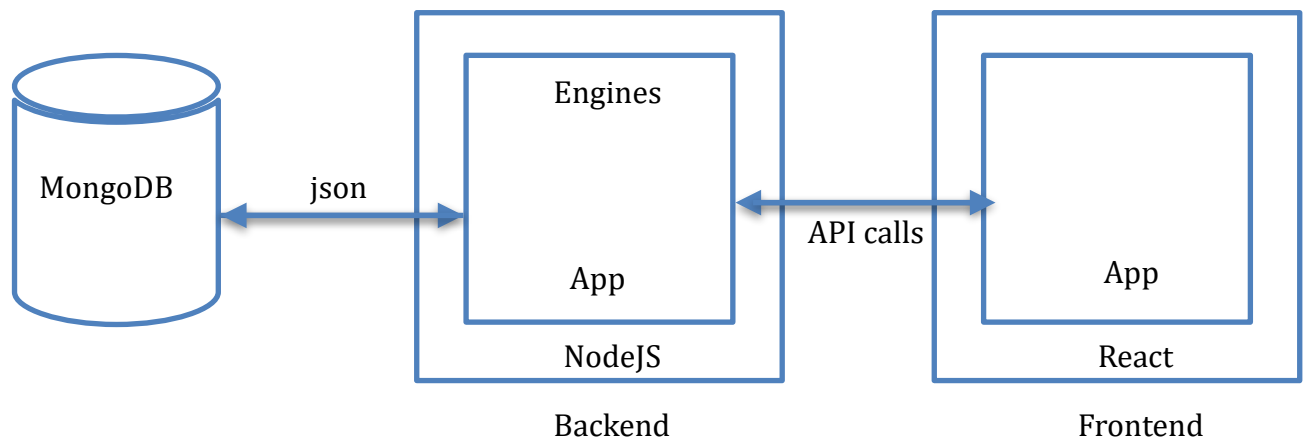
2. System Overview

Figure 1

Figure 1 above represents the architectural structure we have chosen for the development of OTT platform.

Backend:
Database Management System: This acts like a digital library, storing and organising all the content. MongoDB, a NoSQL database, is chosen for its flexibility and scalability, ensuring efficient management of diverse content types.
User Data Management: Storing user accounts and their preferences require robust solutions. MongoDB's flexible schema accommodates varying user data structures, ensuring secure and accessible storage.
Framework: Used NodeJS as our framework as it provides fast, efficient and tight coupling between the client and the server amongst other additional features. NodeJS is also highly scalable which will allow the system to grow further and accommodate a wider range of customers.
Recommendation Engine: Algorithms for content suggestions are the magic sauce. They use complex computations, often powered by Python, to analyse user behaviour and provide tailored recommendations.
Analytics Platform: To understand user habits, we can use an analytics system. Tools like Google Analytics help decode user interactions, guiding improvements.

Frontend:
The user-facing side of the platform is developed using technologies such as React or Angular, which ensure an interactive and seamless user experience.
APIs
To connect with external devices and applications, APIs (Application Programming Interfaces) are vital. These APIs enable devices like smart TVs and gaming consoles to interact with the platform. APIs are created using languages like NodeJs for efficient communication.

Architecture:
Scalable Database: MongoDB, a NoSQL database, ensures smooth scaling as the user base grows. Its flexible structure accommodates evolving data needs. Cloud-based solutions provided by AWS.
Reliable Servers: Backend operations rely on servers, powered by NodeJs, ensuring quick responses and seamless interactions.
Content Delivery Network (CDN): CDNs are used for fast content delivery. Plan to use Amazon CloudFront to cache content in multiple servers globally, reducing loading times for users.
Security Layers: Security tools like SSL certificates ensure encrypted data transfer, keeping user data safe.

Mobile Application Integration:
Expanding to mobile devices involves creating a mobile app that communicates with the backend. React Native enable us to build applications compatible with both Android and iOS, utilising the existing backend infrastructure.

3. System Components

3.1 Decomposition Description
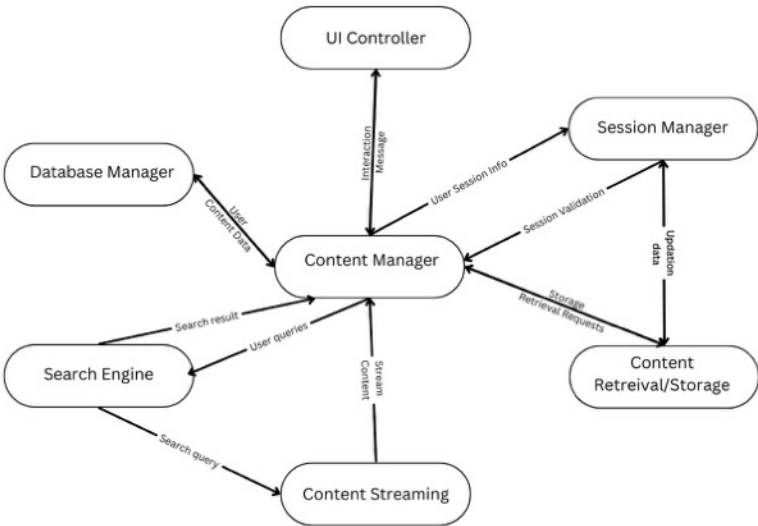
Top Down Details

FIGURE 2

In Figure 2, user interactions are at the core, initiating connections between essential components, taken care by Content Manager. The User Interaction module serves as the starting point, linking to the UI Controller for intuitive user experiences. The Session Manager ensures secure authentication, while the Database Manager facilitates content retrieval and storage. Content interactions are managed by the Content Retrieval/Storage component, connecting to Content Streaming for quality streaming and playback controls. User profiles inform the Recommendation Engine for personalised suggestions, and the Search Engine enables efficient content discovery. These connections create a seamless ecosystem, allowing users easy access to tailored content.
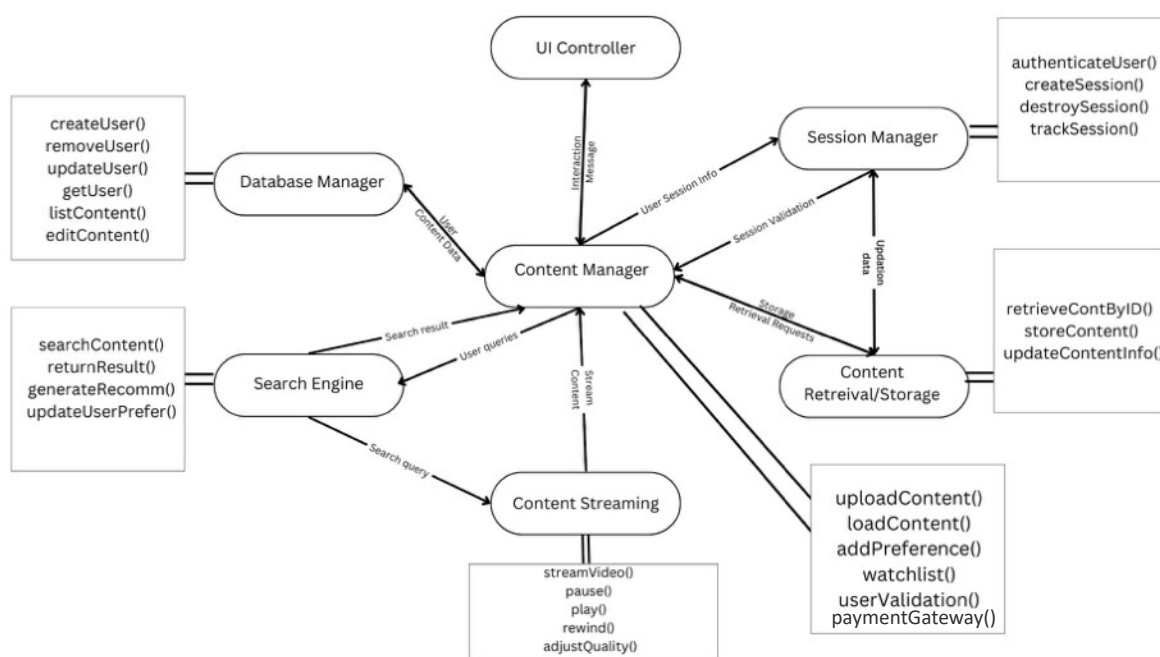
3.2 Dependency Description

FIGURE 3

Figure 3 above represents the component diagram of the OTT platform and how each module is dependent on another for its functionality. Double lines are used to show functions that are attached to modules. user interactions initiate a series of processes crucial for the functioning of the OTT platform. The Content Manager with UI Controller acts as the bridge between users and the system, interpreting user commands and rendering the interface accordingly. Upon user authentication by the Session Manager, the Database Manager comes into play. This component is responsible for managing user profiles and trip details, enabling functionalities such as adding new users, retrieving user information, and storing trip data securely. Simultaneously, the Content Retrieval/Storage component ensures swift access to stored content, utilising caching techniques for efficiency.

When a user requests video content, the Content Streaming module takes charge, ensuring smooth playback and allowing users to adjust streaming quality based on their preferences. Meanwhile, the Recommendation Engine analyses user behaviour, generating personalised content suggestions. If a user seeks specific content, the Search Engine swiftly provides relevant results, enhancing user experience.

These components work in harmony, ensuring that user inputs are translated into meaningful actions. The OTT platform's functionality, from user authentication to content streaming and personalised recommendations, is made possible through the seamless interaction of these interconnected components.

3.3 Interface Description
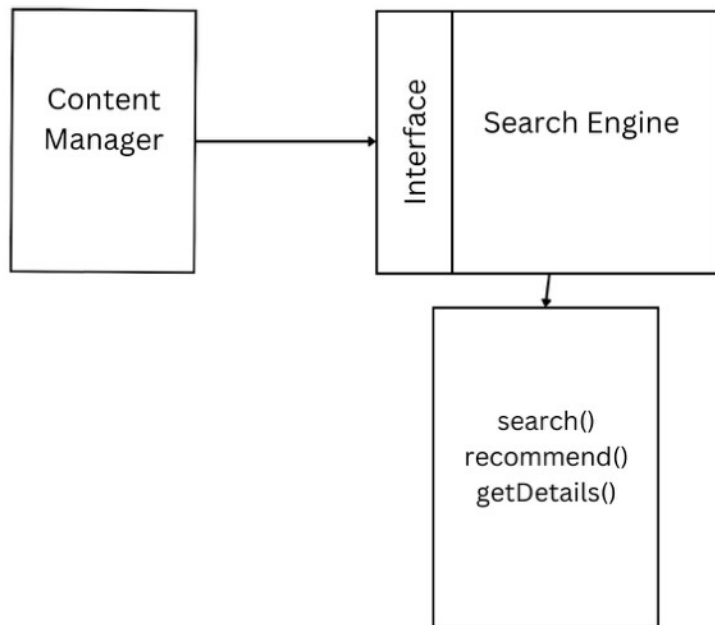
3.3.1 Content Manager to Session Manager
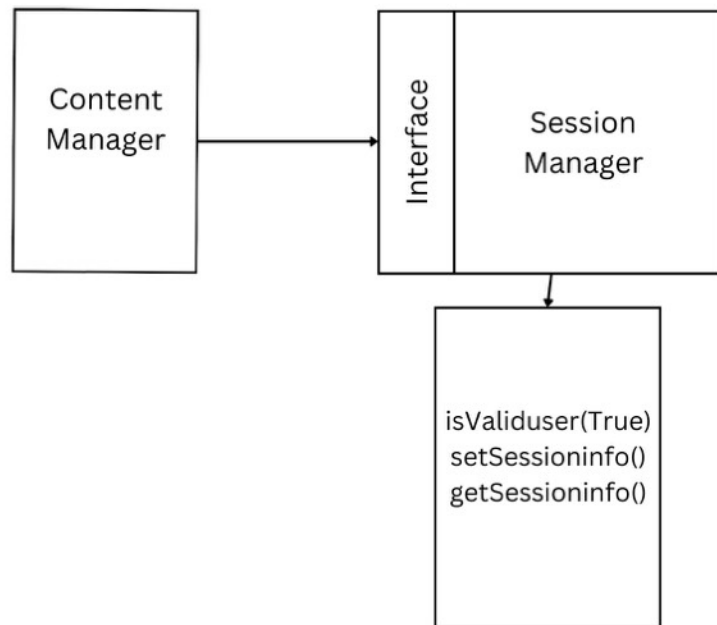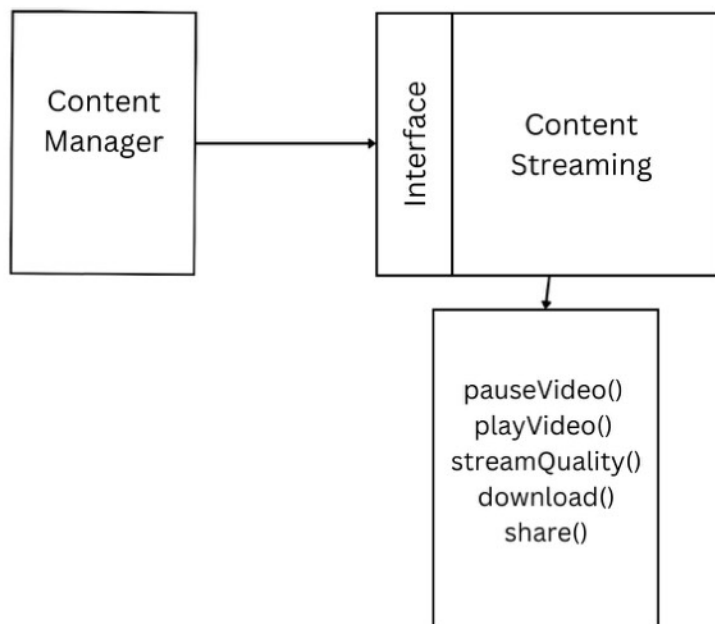

3.3.2 Content Manager to Database Manager
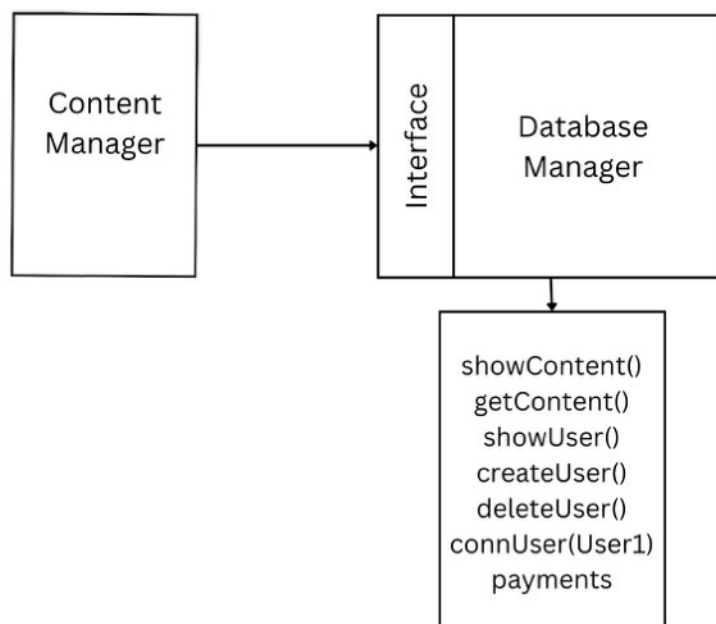
3.3.3 Content Manager to Search Engine


3.3.4 Content Manager to Content Streaming

3.4 User Interfaces (GUI)

The login form (Figure 1) includes fields for entering the username or email and password, alongside a "Remember me" toggle for user convenience. The login page serves as the gateway to the OTT platform, allowing users to securely access their accounts. Through this page, users can log in to their accounts to upload new content, manage their profiles, view their watchlist and access personalised recommendations.
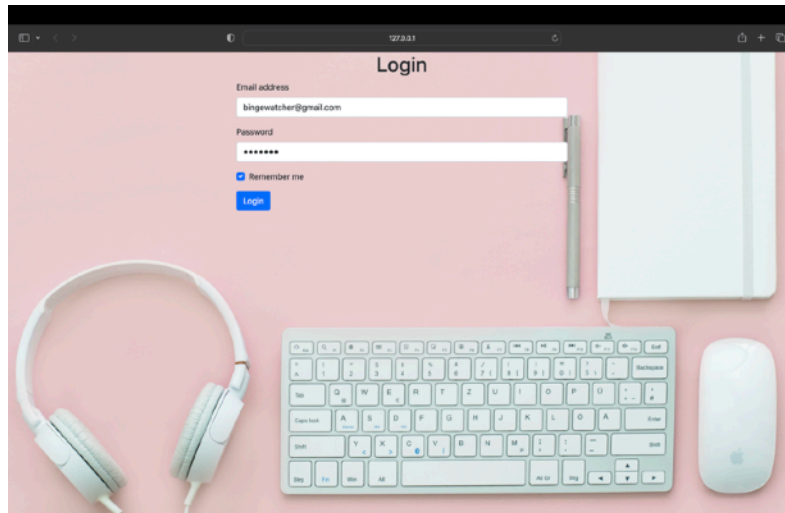
Figure 1 - Login Page

Figure 2 displays the landing page of our OTT platform welcomes visitors with a visually engaging and intuitive interface, designed to captivate and inform. It serves as the digital storefront, offering a glimpse into the diverse world of entertainment we provide. On this page, users can explore different tabs in our landing page enticing them to delve deeper into our platform. Through carefully curated visuals and concise descriptions, the landing page invites visitors to discover a vast array of entertainment options, enticing them to sign up and embark on a cinematic journey tailored to their tastes.
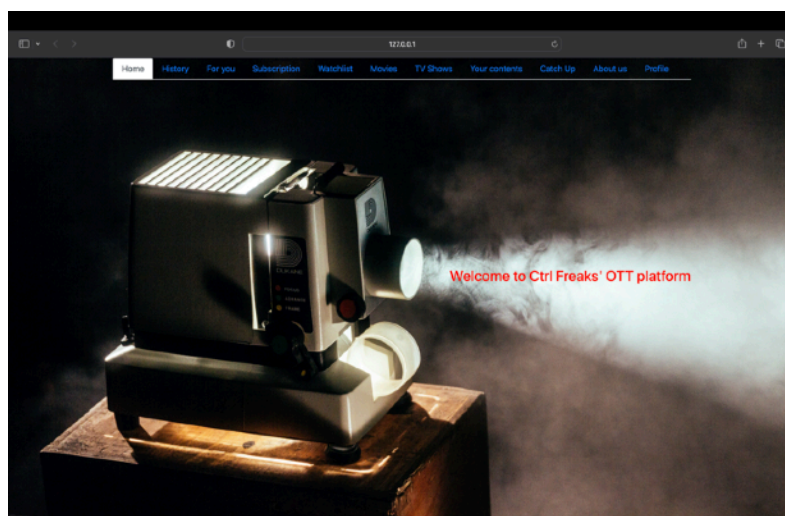
Figure 2 - Landing Page

Figure 3 displays the movies page on our OTT platform is a cinematic paradise for movie enthusiasts. With an extensive collection of films spanning various genres, eras, and cultures, users can dive into an immersive movie-watching experience. Navigating the movies page, users encounter a user-friendly interface that allows them to filter movies based on genres, release years, or even specific actors and directors. Each movie is accompanied by a detailed synopsis, cast and crew information, as well as user ratings and reviews, empowering viewers to make informed choices. Whether users are in the mood for timeless classics, pulse-pounding action, heartfelt dramas, or laugh-out-loud comedies, the movies page offers a vast selection, ensuring there's something for every cinematic palate. Users can seamlessly stream their chosen movies in high definition, bringing the magic of the big screen directly to their devices.
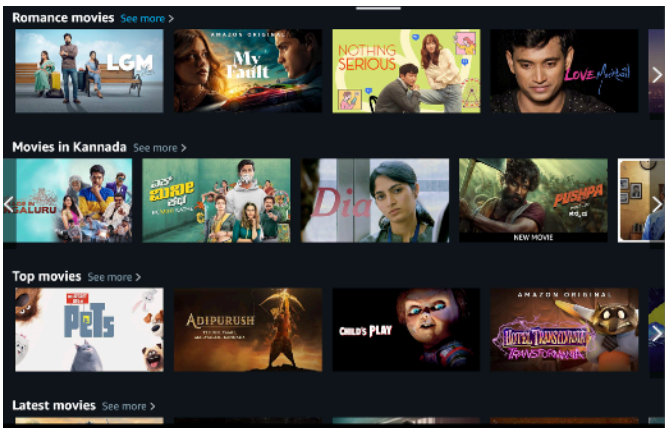

Figure 3 - Movies tab

Figure 4 shows the profile tab is your central control hub on our platform, offering quick access to vital features. Here, you can easily manage your subscription plan, ensuring uninterrupted access to our premium content. Track your payment history effortlessly, staying in the loop about your financial interactions with us. Additionally, personalise your content preferences, making your streaming experience uniquely yours. It's your go-to place for a hassle-free, customised journey on our platform.
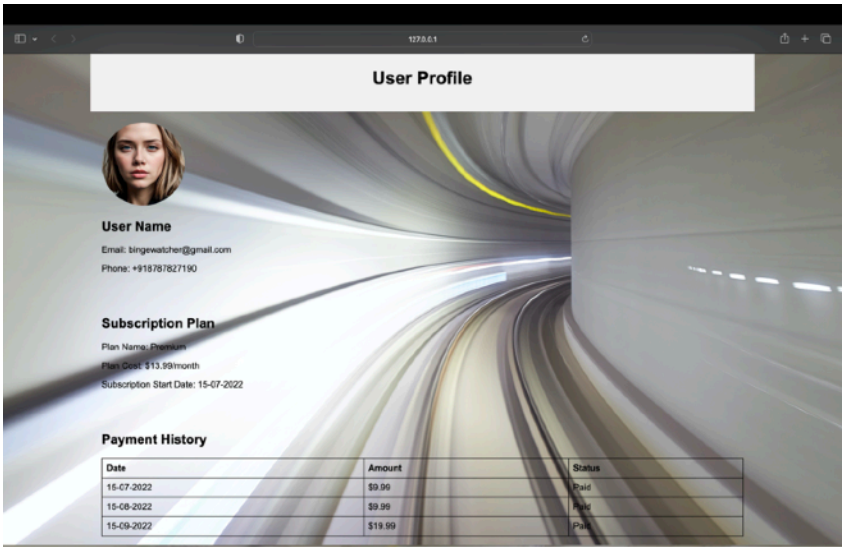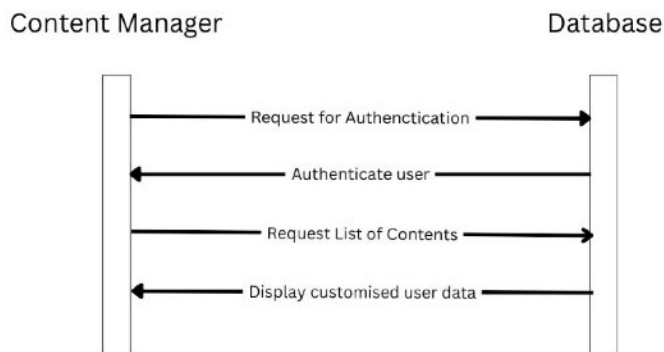

Figure 4 - Profile tab

4. Detailed Design

4.1 Module Detailed Design

4.1.1 Get User Authentication

Sequence Diagram
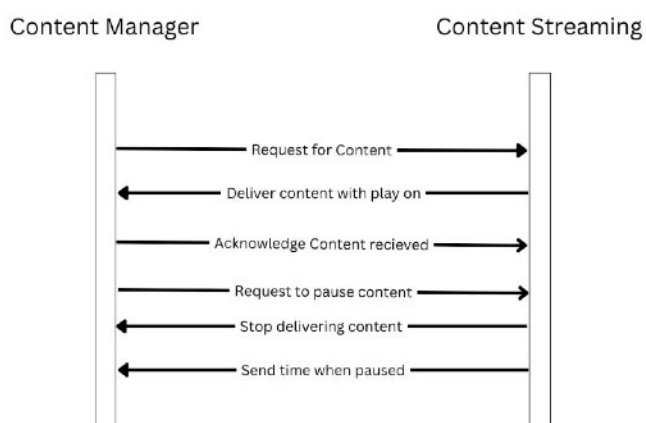

Pseudocode

ContentManager initiates the authentication process by calling
LoginComponent.AuthenticateUser(username, password).
LoginComponent authenticates the user by validating the credentials using Database Manager.
If authentication is successful, LoginComponent fetches user data using
DatabaseManager.FetchUserData(username).
LoginComponent returns the user data to ContentManager.
ContentManager displays the user data to the user. If authentication fails, an error message is
displayed.

4.1.2 Stream Data

Sequence Diagram

Pseudocode

Content Manager receives the request and communicates with Content Streaming to obtain video details and streaming options.
Content Streaming responds to Content Manager with the video data and available streaming quality options.
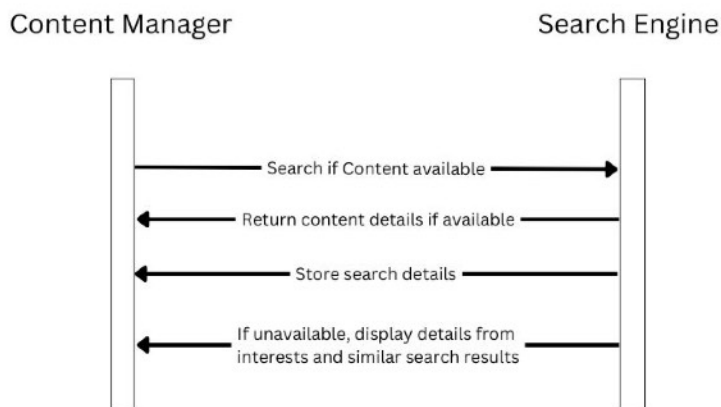Content Manager displays the video player on the user interface along with the streaming options.
User selects the desired streaming quality.
ContentManager communicates the selected streaming quality to ContentStreaming.
ContentStreaming starts streaming the video to the user in the selected quality.

4.1.3 Search Results

Sequence Diagram

Pseudocode

Content Manager sends a request for content recommendations to the Search Engine.
Search Engine receives the request and processes it.
Search Engine accesses user preferences and watch history from the Content Manager.
Search Engine analyses user data to generate personalised content recommendations.
Search Engine sends the list of recommended content back to the Content Manager.
Content Manager receives the recommendations from the Search Engine.
Content Manager displays the personalised content recommendations to the user if searched result in unavailable.

4.2 Data Detailed Design

1) User Data:

Attributes: User ID, Username, Email, Password (encrypted), Subscription Plan, Payment History, Watchlist, Watch history.
Description: User data includes information about registered users. It is stored securely with encrypted passwords. User preferences, such as watchlist and watch history, are associated with each user profile.

2) Content Data:

Attributes: Content ID, Title, Description, Genre, Release Date, Duration, Quality Levels, Streaming URLs, Thumbnails, Language, Age Rating.
Description: Content data represents movies, TV shows, and other media available on the platform. Each content item has unique identifiers, descriptions, and associated metadata, enabling users to search and access specific content.

3) Session Data:

Attributes: Session ID, User ID, Login Time, IP Address, Device Information.
Description: Session data tracks user sessions, including login times, devices used, and IP addresses. It ensures secure access to the platform, allowing users to stay logged in during their visit.

4) Payment Data:

Attributes: Transaction ID, User ID, Payment Method, Amount, Timestamp, Status.
Description: Payment data records user transactions, including subscription payments and content purchases. It includes details about payment methods, amounts, and transaction statuses for billing and auditing purposes.

5) Recommendation Data:

Attributes: User ID, Content IDs (Recommended), Timestamp, Recommendation Source.
Description: Recommendation data stores information about content suggestions made to users. It includes the user ID, content IDs recommended, the source of the recommendation, and timestamps. This data aids in enhancing user engagement.

6) Search Index:

Attributes: Keywords, Content IDs.
Description: The search index is a structured database containing keywords mapped to relevant content IDs. It facilitates efficient content search and retrieval, enhancing user experience by providing accurate search results.

4.3 RTM

Requirement ID	Requirement Description	Design Component	Test case ID
1.1.3.2	Track user access to the platform	Session Manager	2.3
1.2.2.1	Login	Profile Tab & Database Manager	3.7 3.8
1.2.2.3	Adding additional account	Profile Tab & Database Manager	4.0 4.2 4.3
1.3.1.2	Streaming Data	Content Streaming	5.7 5.8 5.9 5.91 5.92
1.4.2.1	Watchlist	Content Storage	6.1 6.3 6.8
1.5.1.1	Payments & Subscription	Profile Tab & Database Manager	7.0 7.1 7.2 7.3 7.4 7.5
1.6.4.5	Edit personal information	Database Manager	8.1 8.2