Design Document - Movie Streaming Platform System

1. Introduction

1.1 Purpose
This document provides a detailed system design for the Movie Streaming Platform. It outlines the architecture, database design, API structure, security considerations, and scalability measures.

1.2 Scope
The system will allow users to browse movies, watch trailers, add movies to their watchlist, and leave reviews. Users can also have premium memberships that grant access to download content. The platform will support web and mobile applications, ensuring a seamless experience across different devices.

1.3 Target Audience
- Software developers
- System architects
- Database administrators
- Quality assurance engineers
- Product managers

2. System Overview

2.1 High-Level Architecture
The system follows a three-tier architecture:
- **Presentation Layer**:Web and mobile applications (React.js / Vue.js for web, React Native for mobile).
- **Business Logic Layer**: RESTful API service built with **Node.js / Express.js** or **Django / Flask**.
- **Data Layer**: **MySQL** relational database for structured data storage.

Technology Stack:
- Frontend: React.js / Vue.js (Web), React Native (Mobile)
- Backend: Node.js / Express.js or Django / Flask
- Database: MySQL
- Cloud Deployment: AWS / Google Cloud / Azure

3. System Components

3.1 User Management Module
- User registration and authentication.
- Role-based access control(Standard & Premium users).
- Password reset functionality.

3.2 Movie Management Module
- Add, update, and delete movie records.
- Store metadata like title, release year, genre, and duration.
- Keep track of view count and ratings.

3.3 Watchlist Module
- Users can add or remove movies from their personal watchlist.
- Retrieve user-specific watchlists.

3.4 Review System
- Users can rate and leave reviews on movies.
- Reviews are stored and displayed with timestamps.

3.5 Actor And Director Management

- Associate actors and directors with movies.
- Store actor and director details.

3.6 Premium Membership System
- Users can upgrade to premium.
- Exclusive access to premium movies and features.

4. Database Design

The system will have the following key tables:

Users Table

Column	Type	Description
UserId	INT	Unique user identifier
Username	VARCHAR(50)	Unique username
Email	VARCHAR(100)	Unique email adress
Password	VARCHAR(100)	Hashed password
JoinDate	DATETIME	User registration date
IsPremium	BOOLEAN	Premium membership status

Genres Table

Column	Type	Description
GenreID	INT	Unique genre identifier
GenreName	VARCHAR(50)	Genre name

Movies Table

Column	Type	Description
MovieID	INT	Unique movie identifier
Title	VARCHAR(255)	Movie title
ReleaseYear	SMALLINT(4)	Year of release
Duration	SMALLINT	Movie duration in minutes
Rating	DECIMAL(3,1)	Average user rating
ViewCount	INT	Number of views

MovieGenre Table

Column	Type	Description
MovieID	INT	Foreign key from Movies table
GenreID	INT	Foreign key from Genres table

Reviews Table

Column	Type	Decription
ReviewID	INT	Unique review identifier
UserID	INT	Foreign key from Users table
MovieID	INT	Foreign key from Movies table
Rating	DECIMAL(3,1)	User rating
Comment	TEXT	Review text
ReviewDate	DATETIME	Date of review submission

Watchlist Table

Column	Type	Description
WatchlistID	INT	Unique watchlist identifier
UserID	INT	Foreign key from Users table
MovieID	INT	Foreign key from Movies table

CastMember Table

Column	Type	Description
CastID	INT	Unique cast member identifier
Name	VARCHAR(255)	Name of the actor or director
Role	ENUM('Director', 'Actor')	Role in the movie
BirthDate	DATE	Date of birth
Nationality	VARCHAR(100)	Nationality of the actor or director

MovieCast Table

Column	Type	Description
MovieID	INT	Foreign key from Movies table
ActorID	INT	Foreign key from CastMember table

5. Conclusion

This document provides a structured system design for the Movie Streaming Platform, detailing the architecture, database, API structure, and security measures. The system will be developed iteratively following Agile methodologies to ensure continuous improvement and scalability.