EZTechMovie Logo
Image of EZTechMovie Company Logo  
INT499: Information Technology Capstone Project

# Miguel Kanto

# Software Requirement Specification (SRS)

StreamList – Collaborative Streaming Service

Zachary Harrison, Armando Perez, Gizelle Samuels, Charles Taggart

3/3/2025

## Table of Contents

1. Introduction
   1. Product Scope
   2. Product Value
   3. Intended Audience
   4. Intended Use
   5. General description
2. Functional Requirements
3. External Interface Requirements
   1. User Interface Requirements
   2. Hardware Interface Requirements
   3. Software Interfaces Requirements
   4. Communication Interface Requirements
4. Non-Functional Requirements
   1. Security
   2. Capacity
   3. Compatibility
   4. Reliability
   5. Scalability
   6. Maintainability
   7. Usability
   8. Other non-functional Requirements

## **Introduction**

EZTechMovie is a fictional technology-driven company specializing in media and entertainment solutions. The company aims to expand the **StreamList** application from an individual movie list tracker into a monthly subscription-based streaming aggregator. This Software Requirements Specification (SRS) document outlines the key requirements for the development of **StreamList 2.0**, detailing its objectives, user expectations, and technical specifications.

**1.1 Product Scope** StreamList 2.0 will provide a comprehensive streaming solution that allows users to:

* Search for a movie across multiple streaming platforms (Netflix, Hulu, Prime Video, etc.).
* Compare pricing and availability on different platforms.
* Stream content directly through the app (if supported by the service).
* Save personalized watchlists that can be shared with friends or family.
* Integrate recommendations based on viewing habits.
* Offer a premium subscription for advanced features such as ad-free browsing and offline watchlists.

**1.2 Product Value** StreamList 2.0 offers a cutting-edge, all-in-one streaming aggregator that stands out from other apps in the industry due to its superior functionality, user-friendly design, and a set of advanced features that prioritize personalization, collaboration, and convenience.

**Key Advantages of StreamList 2.0:**

1. **Multi-Platform Search and Comparison**  
   StreamList 2.0 allows users to search for movies across various streaming platforms such as Netflix, Hulu, and Prime Video. Users can also compare pricing and availability, making it a one-stop-shop for discovering what to watch without switching between apps.
2. **In-App Streaming Integration**  
   Unlike traditional aggregators that merely link to external platforms, StreamList 2.0 allows users to stream content directly within the app (where supported), offering an unmatched level of convenience.
3. **Collaborative Features**  
   StreamList 2.0 enables users to create and share personalized watchlists with friends and family. The collaborative feature allows group voting and recommendations, fostering social engagement and enhancing the content discovery experience.
4. **AI-Powered Recommendations**  
   StreamList 2.0 leverages an AI-based recommendation system that suggests movies and shows based on user preferences and viewing habits, ensuring relevant content discovery with every interaction.
5. **Premium Subscription for Advanced Features**  
   StreamList 2.0 offers a premium subscription that unlocks an ad-free experience, offline watchlists, and priority customer support, providing a more refined and flexible streaming experience.

**How These Features Enhance User Satisfaction and Promote Long-Term Retention:**

* **Convenience**: The multi-platform search and in-app streaming features save users time and effort, providing a hassle-free experience.
* **Social Engagement**: Collaborative watchlists and group voting features create a sense of community, encouraging regular app usage and interaction with friends and family.
* **Personalized Content**: AI-based recommendations keep users engaged by offering relevant content tailored to their viewing habits, making content discovery enjoyable and rewarding.
* **Refined Experience**: The premium subscription’s ad-free experience and offline capabilities enhance user satisfaction, making it more appealing to users who want an uninterrupted streaming experience.

StreamList 2.0 is designed to deliver superior performance, ease of use, and a unique blend of social and personalized features, positioning it as a top contender in the streaming aggregator market.

**1.3 Intended audience** StreamList 2.0 is intended for a wide range of users who enjoy streaming movies and TV shows. These include:

* **General Consumers**: People looking for a streamlined, one-stop platform to search for and watch content across multiple streaming services.
* **Families and Friends**: Groups who want to share watchlists and discuss what to watch.
* **Film Enthusiasts**: Users interested in keeping track of available movies across platforms.
* **Subscription Managers**: Users who want to optimize their subscription plans by comparing content availability across different services.

**1.4 Intended use** StreamList 2.0 enables users to:

1. Create an account and select their preferred streaming services.
2. Search for movies and check availability across multiple platforms.
3. Stream content from integrated services or be redirected to external platforms.
4. Add movies to personal watchlists and share them with others.
5. Receive personalized recommendations based on past viewing history.
6. Subscribe to premium features for an ad-free experience and offline watchlists.

**1.5 General Description** StreamList 2.0 will function as a **centralized streaming aggregator** that offers:

* A search engine to find movies across multiple platforms.
* A collaborative watchlist system where users can add, remove, and vote on movies.
* A subscription model that unlocks premium features.
* AI-based recommendations powered by user preferences.

## 2. Functional Requirements

## Search Functionality: Ability to find a movie across different streaming services.

## Watchlist Management: Users can create, update, and share watchlists.

## Streaming Integration: Users can watch content via built-in media players or external redirects.

## Subscription Handling: Supports free and premium accounts with access restrictions.

## Social Features: Allow users to share recommendations and rate movies.

## Data Analytics: Track viewing preferences to improve recommendations.

## 3. External Interface Requirements

**3.1 User Interface Requirements**

* **Modern UI Design**: A responsive layout that adapts to various devices.
* **Dark Mode & Accessibility Options**: Ensure comfort for all users, including those with visual impairments.
* **Search Bar**: For quick movie discovery.
* **Filter Options**: Genre, rating, streaming platform, etc., to refine searches.

**3.2 Hardware Interface Requirements**

* **Desktop**: Windows, macOS, Linux.
* **Mobile**: iOS, Android.
* **Streaming Devices**: Fire TV, Roku, Smart TVs (future updates).

**3.3 Software Interface Requirements**

* **Frontend**: React.js (UI framework).
* **Backend**: Node.js with Express.js (API).
* **Database**: MongoDB or Firebase for storing watchlists.
* **Third-Party APIs**: Integration with streaming services (Netflix, Hulu, etc.).

**3.4 Communication Interface Requirements**

* **Email Notifications**: For new releases and updates to watchlists.
* **Push Notifications**: For mobile users.
* **Embedded Chat System**: For group recommendations (future feature).

## 4. Non-Functional Requirements

**4.1 Security**

* **OAuth2 Authentication**: Via Google, Apple, or email.
* **End-to-End Encryption**: Ensures the security of user data.
* **Compliance**: GDPR & CCPA.

**4.2 Capacity**

* **Initial Storage**: 1GB per user for metadata and watchlists.
* **Scalability**: Cloud-based auto-scaling (AWS or Firebase).

**4.3 Compatibility**

* **Browser Compatibility**: Chrome, Firefox, Edge, Safari.
* **System Requirements**: 4GB RAM, 1.5GHz processor.

**4.4 Reliability**

* **Uptime**: 99.9% uptime guarantee via cloud-based hosting.
* **Automatic Backups**: Every 24 hours.

**4.5 Scalability**

* Optimized for 1 million concurrent users.
* API request limit: 500 requests per user per hour.

**4.6 Maintainability**

* **CI/CD Pipelines**: For frequent updates.
* **Automated Testing**: Using Jest and Cypress.
* **Bug-Tracking System**: Via GitHub Issues.

**4.7 Usability**

* **Simple UI**: Easy-to-use interface.
* **Voice Search Integration**: For accessibility.
* **Minimal Onboarding**: Easy to get started for new users.

**4.8 Other Non-Functional Requirements**

* **Legal Compliance**: DMCA-compliant API access.
* **AI-Based Recommendation Engine**: For personalized content discovery.
* **Offline Mode**: Allows users to save watchlists for offline access.

## References

Amazon Web Services. (n.d.). AWS Auto Scaling. Retrieved February 3, 2025, from <https://aws.amazon.com/autoscaling/>

Fowler, M. (2018). Continuous integration: Improving software quality and reducing risk. Addison-Wesley Professional.

Google Developers. (n.d.). OAuth 2.0 for Google APIs. Retrieved February 3, 2025, from <https://developers.google.com/identity/protocols/oauth2>

Krug, S. (2014). Don't make me think, revisited: A common sense approach to web usability (3rd ed.). New Riders.

Mozilla Developer Network. (n.d.). Web storage API. Retrieved February 3, 2025, from <https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API>

Nielsen, J. (1994). Usability engineering. Morgan Kaufmann.

Open Web Application Security Project (OWASP). (2023). OWASP top ten security vulnerabilities. Retrieved from <https://owasp.org/www-project-top-ten/>

Roy, S. (2021). Node.js design patterns (3rd ed.). Packt Publishing.

Tanenbaum, A. S., & Bos, H. (2021). Modern operating systems (5th ed.). Pearson.

Woods, D. (2019). APIs: A strategy guide. O’Reilly Media.