**Software Requirements Specification**

**for**

**Video-on-Demand Platform (Meta Stream)**

**Version 1.0 approved**

**Prepared by**

**Chitturi Sai Suman – 1602-18-733-097**

**Praneeth Kapila – 1602-18-733-116**

**Vasavi College of Engineering**

**30-09-2020**

1. **Introduction**

**1.1 Purpose**

The purpose of this document is to present a detailed description of the Open-Source Web Application, Video-on-Demand Platform “Meta Stream”. It will explain the purpose and features of the application, the interfaces, what the application will do and the constraints under which it must operate. This document is intended for users of the application and potential developers.

**1.2 Document Conventions**

This Document was created based on the IEEE template for System Requirement Specification Documents.

**1.3 Intended Audience and Reading Suggestions**

* Typical Users, who are looking for some Entertainment.
* Programmers who are interested in working on the project by further developing it or fix existing bugs.

**1.4 Product Scope**

Meta Stream is a Video-on-Demand Platform that enables people to stream High-Quality content. Users can download the content and add Videos to their Watch-list. Users can also choose from wide range of high-quality Stock videos of different Genre.

**1.5 References**

[1]. <https://www.geeksforgeeks.org/web-technology/html-css/>

[2]. <https://www.w3schools.com/>

[3]. <https://www.w3.org/>

**2. Overall Description**

**2.1 Product Perspective**

Meta Stream was developed for anyone who is interested in watching High-Quality Videos. It is an open source project, and it has a very active developer team to support it and provide feedback to users. It offers cross-platform support.

**2.2 Product Functions**

**2.2.1 Admin Functions**

**2.2.1.1 Login**

|  |  |
| --- | --- |
| **Use Case:** | Login |
| **Primary Actor:** | Admin |
| **Goal in Context:** | Managing the Web – Application |
| **Scenario:** | 1. The system prompts the admin for their username and password.  2. The admin enters username and password.  3. The system gets password registered to the username.  4. The system verifies the password and sets the admin’s authorization.  5. The admin is given access to the application to perform their job. |

**2.2.1.2 Management Selection**

|  |  |
| --- | --- |
| **Use Case:** | Management Selection |
| **Primary Actor:** | Admin |
| **Goal in Context:** | Accessing to sub-menu |
| **Scenario:** | 1. Admin selects between stream management and user management. |

**2.2.1.3 Stream Management**

|  |  |
| --- | --- |
| **Use Case:** | Stream Management |
| **Primary Actor:** | Admin |
| **Goal in Context:** | Changing access rights of a stream |
| **Scenario:** | 1. The admin enters stream management interface.  2. The admin selects the stream that he/she wants to manage.  3. The admin changes the access rights of the stream on the menu.  4. New access rights of the stream are set. |

**2.2.1.4 Add Stream**

|  |  |
| --- | --- |
| **Use Case:** | Add Stream |
| **Primary Actor:** | Admin |
| **Goal in Context:** | Adding new stream to Media Server |
| **Scenario:** | 1. Admin clicks Add button in stream Management Window.  2. Window asks for the configuration settings of stream.  3. Admin enters the configuration settings.  4. Admin clicks OK button.  5. Stream is added to Stream List. |

**2.2.1.5 Remove Stream**

|  |  |
| --- | --- |
| **Use Case:** | Remove Stream |
| **Primary Actor:** | Admin |
| **Goal in Context:** | Removing stream from Media Server |
| **Scenario:** | 1. Admin chooses a stream from Stream list.  2. Admin clicks Remove button.  3. Stream is removed from Stream List |

**2.2.1.6 User Management**

|  |  |
| --- | --- |
| **Use Case:** | User Management |
| **Primary Actor:** | Admin |
| **Goal in Context:** | Changing access rights of a user |
| **Scenario:** | 1. The admin enters user management interface.  2. The admin selects the user that he/she wants to manage from the user list.  3. The admin changes the access rights of the user.  4. New access rights of the user are set. |

**2.2.1.7 Add User**

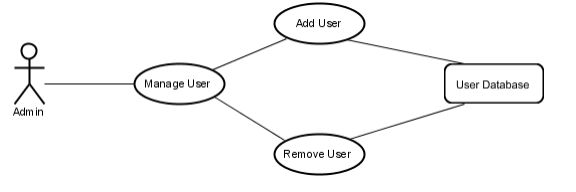
|  |  |
| --- | --- |
| **Use Case:** | Add User |
| **Primary Actor:** | Admin |
| **Goal in Context:** | Adding new user to Relational Database |
| **Scenario:** | 1. Admin clicks Add button in User Management Window.  2. Admin enters the user details.  3. Admin clicks OK button.  4. User is added to database. |

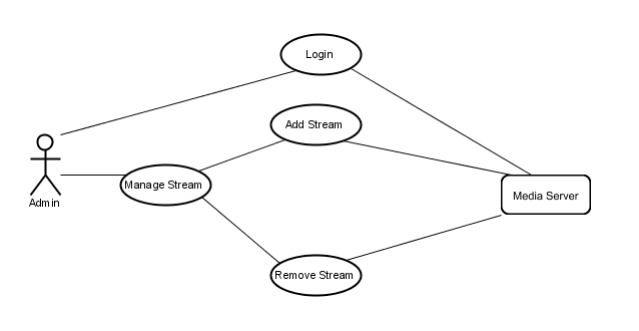
**2.2.1.8 Remove User**

|  |  |
| --- | --- |
| **Use Case:** | Remove User |
| **Primary Actor:** | Admin |
| **Goal in Context** | Removing user from Relational Database |
| **Scenario:** | 1. Admin chooses a user from User list. 2. Admin clicks Remove button.  3. User is removed from User List |

**2.2.1.9 Use-Case Diagrams**

In these diagrams, ‘Add Stream’ and ‘Remove Stream’ use-cases are specific operations of ‘Manage Stream’. Similarly, ‘Add User’ and ‘Remove User’ use-cases are specific operations of ‘Manage User’.





**2.2.2 User Functions**

**2.2.2.1 Log in**

|  |  |
| --- | --- |
| **Use Case:** | Login |
| **Primary Actor:** | User |
| **Goal in Context:** | Accessing to the system |
| **Scenario:** | 1. The application prompts the user for their username and password.  2. The user enters username and password.  3. The application gets password registered to the username  4. The application verifies the password and sets the users authorization.  5. The user is given access to the application to perform their job. |

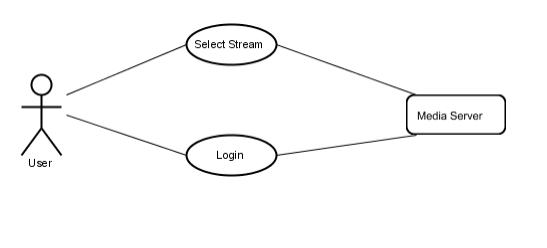
**2.2.2.2 Stream Selection**

|  |  |
| --- | --- |
| **Use Case:** | Stream selection |
| **Primary Actor:** | User |
| **Goal in Context:** | Selecting a stream to access |
| **Scenario:** | 1. User logs into the application.  2. After login screen, user is provided a list of streams that she/he can use.  3. User selects a stream and data flow starts. |

**2.2.2.3 Media Player Functions**

|  |  |
| --- | --- |
| **Use Case:** | Media player functions |
| **Primary Actor:** | User |
| **Goal in Context:** | Controlling playback |
| **Scenario-1:** | User presses play button to start data flow and play it. |
| **Scenario-2:** | User presses pause button to pause the stream at the current frame. |
| **Scenario-3** | User adjusts volume level from the volume bar. |
| **Scenario-4** | User adjusts Resolution of video dynamically during playback. |

**2.2.2.4 Use-Case Diagrams**



**2.2.3 System Functions**

**2.2.3.1 Load Check**

|  |  |
| --- | --- |
| **Use Case:** | Load Check |
| **Primary Actor:** | System Agent |
| **Goal in Context:** | Checking the load on web servers |
| **Scenario:** | 1. Waits for a pre-determined time interval.  2. Checks the current load on each reverse proxy web server.  3. If a webserver serves more than a predetermined number of clients, an idle web server is activated. |

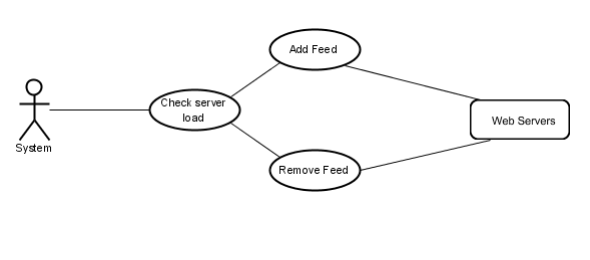
**2.2.3.2. Add Feed**

|  |  |
| --- | --- |
| **Use Case:** | Add Feed |
| **Primary Actor:** | System Agent |
| **Goal in Context:** | Activating an idle web server |
| **Scenario:** | 1. Agent decides that a new web server is needed. 2. An idle web server is activated, and it starts to stream data to clients. |

**2.2.3.3 Remove Feed**

|  |  |
| --- | --- |
| Use Case: | Remove Feed |
| Primary Actor: | System Agent |
| Goal in Context | Stopping a web server |
| Scenario: | 1. Agent decides that less active web servers are enough.  2. One web server stops streaming and its clients are distributed among other active servers. |

**2.2.3.4 Use-Case Diagrams**



**2.3 User Classes and Characteristics**

* Typical Users, who are looking for some Entertainment.
* Programmers who are interested in working on the project by further developing it or fix existing bugs.

**2.4 Operating Environment**

* Operating System
  1. Windows
  2. Linux
* Application Server
  1. Apache Tomcat v9.0
* Media Server
  1. Google Cloud
* Database
  1. MySQL

**2.5 Design and Implementation Constraints**

The Application uses modular design where every feature is wrapped into a separate module and the modules depend on each other through well-written APIs. There are several APIs available to make plugin development easy.

**2.5.1 Frontend**

The Frontend is implemented using HTML, CSS and Java Script.

**2.5.2 Backend**

**2.5.2.1 Backend Application**

The Backend consists of server-side application designed using Java/Python that serves requests of clients.

**2.5.2.2 Database**

Database used is MySQL. It is used for storing User Information and Content Information.

**2.5.2.3 Server**

Server used is Apache Tomcat v9.0

**2.5.2.4 Storage**

The application uses Google Cloud Storage to store media.

**2.6 User Documentation**

Online help and FAQs are provided in the User Documentation

**2.7 Assumptions and Dependencies**

* Media streams are working fine.
* Streams and server have enough bandwidth.
* System has low latency.
* Video processing and networking libraries are available on the system.

**3. External Interface Requirements**

**3.1.1 Admin Interfaces**

**3.1.1.1 Login Window**

Admins should login to the application before using application management operations. This login window is the first thing the users face when they attempt to access the application. They shall be able to continue using the application upon entering a valid username and password.

**3.1.1.2 Management Selection Window**

Admins see management options which are Stream Management and User Management in this window.

**3.1.1.3 Stream Management Window**

This window lists available Stream feeds of the application. Admins can open dialogues for adding Streams to the application and removing them from the application using buttons on this window.

**3.1.1.4 Add Stream Window**

This window prompts admins to enter configuration options of target stream. After entering correct settings stream will be added to available stream feeds list.

**3.1.1.5 Remove Stream Window**

This window prompts admins to approve removing target stream.

**3.1.1.6 User Management Window**

Admins should be able to list all users with their access levels in this window. There should be options to adjust users’ access levels.

**3.1.1.7 Add User Window**

Admins should be able to add new users in this window.

**3.1.1.8 Remove User Window**

Admins should be able to confirm or cancel user removal operation.

**3.1.2 User Interfaces**

**3.1.2.1 Login Window**

Users should login to the application before listing the available Stream feeds and viewing them.

**3.1.2.2 Stream Selection Window**

Users display the Stream selection window after successful login. They see the list of available Stream feeds that they can view in this window.

**3.1.2.3 Media Player Window**

Users see media player window when they want to view a Stream feed. This window has a media player with play/pause, sound volume buttons.

**3.2 Hardware Interfaces**

* **Streaming in 1080P FHD**

Processor: **2.4Ghz** (dual core) or **3.5Ghz** (single Core) processor.

Graphics: **Nvidia/ATi** having bare minimum **256MB Video RAM** and **core clock 600Mhz.**

* **Streaming in 2160P UHD**

Processor: **2.6Ghz** (dual core) or **4.0Ghz** (single Core) processor.

Graphics: **Nvidia/ATi** having bare minimum 512**MB Video RAM** and **core clock 1.0Ghz.**

**3.3 Software Interfaces**

* + JavaScript must be enabled for browser playback.
  + Popup and Ad-blockers can inhibit playback.
  + Flash: 15 or higher.
  + Browser: Internet Explorer 9.0 or later, Mozilla Firefox 17 or later, Google Chrome 36 or later, Safari 5.0 or later.

**3.4 Communication Interfaces**

[Our Application uses both streaming protocols and HTTP-based protocols](https://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=84496). Streaming protocols like [Real-Time Messaging Protocol (RTMP)](https://www.wowza.com/blog/rtmp-streaming-real-time-messaging-protocol) enable speedy video delivery using dedicated streaming servers, whereas HTTP-based protocols rely on regular web servers to optimize the viewing experience and quickly scale.

**4. System Features**

This section demonstrates Meta Stream’s most prominent features and explains how they can be used.

**4.1 Personalized Recommendations**

Based on previously watched videos, videos of the similar Genre are recommended to the users.

**4.2 Enhance Search Results**

Users can filter search results based on several attributes such as Most Popular, Upload date, Rating etc.

**4.3 User Reviews**

Users can rate and review the videos they watch. These reviews are publicly visible.

**4.4 Downloadable Content**

Videos can be downloaded to the User’s Local Machine in their preferred Resolution.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

The Application requires a system with at least a 2400 MHz CPU and 256 MB of VRAM and core clock of 600Mhz. Performance depends on the quality of Video and as a result, the system requirements for videos with higher quality are more demanding.

**5.2 Safety Requirements**

To ensure that no one of Meta Stream’s users loses any data while using it (due to a crash or a bug of some kind) the developer team updates the Application regularly. There is a bug tracker available where users can report any bugs, they have encountered so that the developers can fix it in the next release.

**5.3 Security Requirements**

Meta Stream does not have any security requirements and thus any type of user can use it without any additional privileges.

**5.4 Software Quality Attributes**

Due to its well designed and easy to use interface it can be used by anyone without much skill. However, users must already have a basic knowledge of using a Web Browser.

**5.5 Business Rules**

Sensitive information such as Name and Email will be private and will not be shared to 3rd party advertising companies. The Application is designed in such a way that the User Activity is not accessible by anyone (including the admin).

**6. Other Requirements**

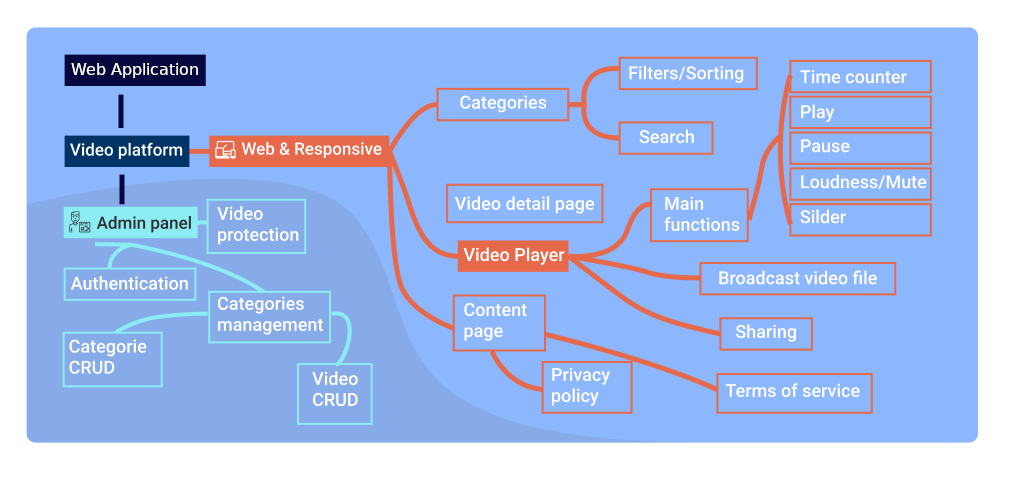
In order to get Seamless Streaming Experience, users require high-speed internet with a speed of 20 Mbps. As the videos are of high quality, they occupy significant storage space. So, before downloading them, users shall make sure there is enough local storage space.

**Appendix A: Glossary**

* HTTP: Hypertext Transfer Protocol
* GHz: Giga Hertz
* MHz: Mega Hertz
* FHD: Full High-Definition
* UHD: Ultra High-Definition
* MB: Mega Byte
* Mbps: Mega bit per second
* RAM: Random access memory
* VRAM: Video RAM

**Appendix B: Analysis Models**

* + **Data Flow Diagram**



**Appendix C: To Be Determined List**

1. We intend to add “Continue Watching” feature if possible.
2. We are looking to collaborate with Content Creators for paid content.
3. We are looking for adding Payment Integration to the Application.