**Software requirements specification**

**Introduction**

*Definition about red5 server:*

- Red5 Server is an open source, open source Flash RTMP server that includes HLS, WebSockets and RTSP, written in digital JAVA. It is supported completely free to users.

- This is an effective online solution used by famous companies around the world such as Amazon, Facebook. An online multitasking, multi-user server that connects easily and optimally.

*Features red5 server:*

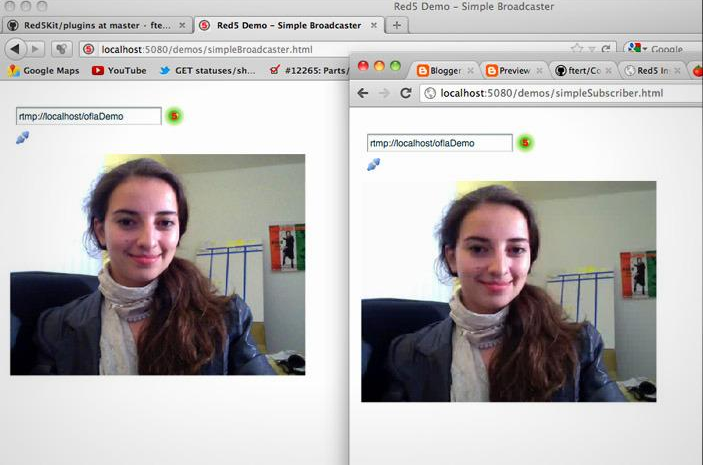
+ Transfer and download movies / videos / photos in 2 main formats: FLV, MP3.

+ Record the information streams transmitted from the client machine. However, it only supports FLV format.

+ Share data with objects operating on the same server.

+ Publish data including: audio, video during streaming, Streaming.

Remote data connection, sync with all members in a meeting / conference.



## System Requirements

The functional requirements define the basic functions of the system and how it behaves given certain circumstances. This section specifies the results each component should have, given the inputs from the users. After this requirements are complete, use cases can be detailed.

### Server

1. The main interface will communicate with both agents through a web services interface.
2. The main interface shall have a database to store all the clients and the video contents information.
3. Every communication starts in the client or the admin.
4. The main interface is responsible to control and monitor the streaming server.
5. The streaming server must be a stand-alone application and provide feedback to the main interface; this information shall only be accessible to the administrator.
6. The streaming server must support RTSP along with trick play functionality.
7. Every multimedia stream must use Transport Stream container.
8. Every multimedia file must be codified with H264 codec.
9. All the videos must be uploaded and stored in the server’s computer; the appropriate format should be created by the main interface in case the video file is not in according to the video file specification.
10. Every video file must have a unique identifier.
11. A TCP/IP connection will be used to the main interface and the streaming server’s communication.
12. To describe the web services a WSDL file must be used.
13. To provide the web services a SOAP web server must be implemented.

### WebAdmin

1. This application shall use a web based interface.
2. All video files must be uploaded through this interface.
3. An authentication process must be implemented.
4. The interface shall provide appropriate accessibility to all the content information and streaming server’s feedback .
5. The administrator must be able to collect and serialize all video information properly, and upload it to the server’s interface.
6. The video information must be collected from IMDB.
7. To connect to the server the web service’s API will be used.

### Client

1. An authentication process must be implemented.
2. All video content available must be displayed in the client interface.
3. When requested by the user the video information must be visualized, including the associated image.
4. The user must be able to play and pause the video stream including the ability to seek, fast-forward and reverse-play the stream.
5. To access to the server’s information and stream URL the web service’s API will be used.
6. To play the multimedia stream a RTP/RTSP must be used in the client interface.

**Security requirements**

**Encryption**

First and foremost, it needs to be mentioned that WebRTC streams are always encrypted.

Encryption is a way of scrambling data so that only authorized parties can understand the information. In technical terms, it is the process of converting plaintext to ciphertext. In simpler terms, encryption takes readable data and alters it so that it appears random. This requires the use of two encryption keys; one public and one private. Those keys are a set of mathematical values that both the sender and the recipient of an encrypted message can decipher. Encryption needs to be random to prevent unauthorized users from accessing the data, but predictable for the authorized parties receiving the information so that it can be used correctly.

Since WebRTC works directly in the browser, this means that the encryption process is also performed in the browser with no additional configurations required. Furthermore, WebRTC does not need to download any additional plugins. This further increases security as it eliminates the concern of third party software and potential side effects such as data tracking or viruses. Plugins are also another potential security risk as they are an additional connection that could be exploited.

WebRTC security enables AES (Advanced Encryption Standard) based protection. As such, this eliminates the risk of using third parties or leveraging a DIY platform to manage all the functions related to authenticating devices, and authorizing users. Instead, WebRTC uses the video transport protocol SRTP (Secure Realtime Protocol) to send and receive encrypted content over the three channels WebRTC devotes to video, audio, and data.

Exchanges of the keys used by SRTP to encrypt and decrypt content are managed through a version of the IETF’s TLS known as DTLS (Datagram Transport Layer Security), which is used with UDP (User Datagram Protocol) connectivity, the ultra-low latency packet transmission protocol employed by WebRTC. While we describe using UDP since that’s the typical setup using WebRTC, it should be noted that the same process can be done over TCP. All of this happens automatically with instantiation of a WebRTC stream. This will be covered in more detail later on.

Furthermore, the same WebRTC security architecture will be replicated no matter what hosting provider is used. The ability to support cross-cloud solutions increases flexibility. It also enables the establishment of the same security features in different regions since WebRTC security implementation is standard.

Encryption ensures that the data sent between a broadcaster and subscriber cannot be read. The next sections will cover how the connection is established in the first place.

## Use Cases

This section refers to all the use cases the users will have in their respective interface. A use-case diagram is available for all interfaces. Each table represents a more detailed description of each use case.

### Client Interface

|  |  |
| --- | --- |
|  | Content Access |
| Actor | Client |
| Include | Login, Play Content, Select Content, List Available Contents |
| Description | The client can access to a variety of contents which includes playing the media stream. |

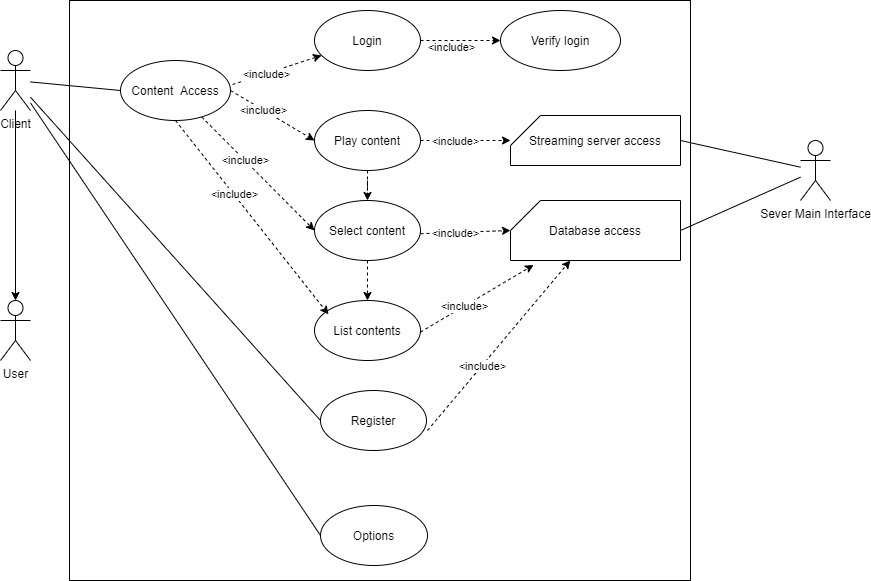
Content Access

|  |  |
| --- | --- |
|  | Register |
| Actor | Client |
| Include | Database Access |
| Description | The client needs to register before being able to access all the content. |

Register

|  |  |
| --- | --- |
|  | Options |
| Actor | Client |
| Description | The client can setup configuration options like viewing parameters. |

Options



Client Interface

|  |  |
| --- | --- |
|  | Login |
| Actor | Client |
| Include | Verify Login |
| Description | The client needs to authenticate to the server. |

Login

|  |  |
| --- | --- |
|  | Verify Login |
| Actor | Main Interface |
| Description | The main interface needs to verify the client’s login. |

Verify Login

|  |  |
| --- | --- |
|  | Play Content |
| Actor | Client |
| Include | Streaming Server Access |
| Dependency | Select Content |
| Description | The client requests to play a video stream. |

Play Content

|  |  |
| --- | --- |
|  | Select Content |
| Actor | Client |
| Include | Database Access |
| Dependency | List Available Contents |
| Description | The client selects content from the content list provided, to view more detailed information and to play the video stream. |

Select Content

|  |  |
| --- | --- |
|  | List Available Contents |
| Actor | Client |
| Include | Database Access |
| Description | The client can list all available contents provided by the VoD service. |

List Available Contents

### Admin Interface

|  |  |
| --- | --- |
|  | Content Management |
| Actor | Admin |
| Include | Login, MediaSession Information, MediaSession Management,  Video Upload, Manage Users, Manage Content information |
| Description | The admin has access to a variety of administration tools. |

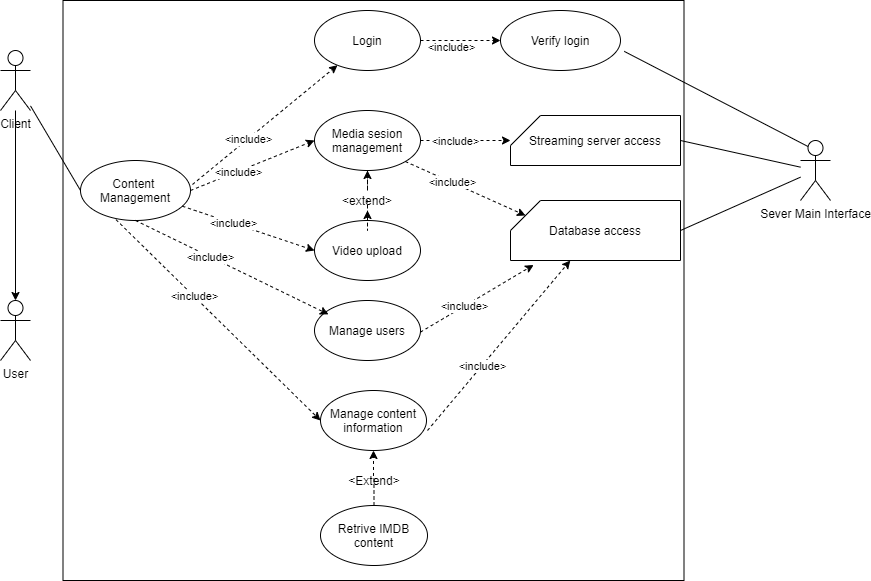
Content Management

|  |  |
| --- | --- |
|  | Login |
| Actor | Admin |
| Include | Verify Login |
| Description | The admin needs to authenticate to the server. |

Login

|  |  |
| --- | --- |
|  | Verify Login |
| Actor | Main Interface |
| Description | The main interface needs to verify the admin’s login. |

Verify Login



Admin Interface

|  |  |
| --- | --- |
|  | MediaSession Management |
| Actor | Admin |
| Include | Streaming Server Access |
| Description | The admin can manage the Media Sessions, like load a video file to the media server and add it to the database. The admin can also access the Media Session’s information. |

Media Session Management

|  |  |
| --- | --- |
|  | Upload Video |
| Actor | Admin |
| Extends | Media Session Management |
| Description | The admin can upload a file to the server. |

Upload Video

|  |  |
| --- | --- |
|  | Manage Users |
| Actor | Admin |
| Include | Database Access |
| Description | The admin can access and manage the User’s information. |

Manage Users

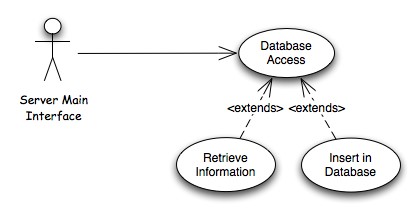
|  |  |
| --- | --- |
|  | Manage Content Information |
| Actor | Admin |
| Include | Database Access |
| Description | The admin can manage the content information. |

Manage Content Information

|  |  |
| --- | --- |
|  | Retrieve IMDB content |
| Actor | Admin |
| Include | Manage Content Information |
| Description | The admin can retrieve information from the Internet Movie Database Site. |

IMDB Information

### Database Access



Database Access

|  |  |
| --- | --- |
|  | Database Access |
| Actor | Main Interface |
| Description | The main interface accesses the database to retrieve or insert information. |

Database Access

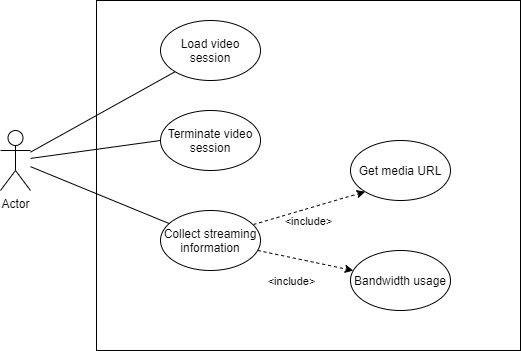
|  |  |
| --- | --- |
|  | Retrieve Information |
| Actor | Main Interface |
| Extends | Database Access |
| Description | The main interface accesses the database to query information. |

Retrieve Information

|  |  |
| --- | --- |
|  | Insert Information |
| Actor | Main Interface |
| Extends | Database Access |
| Description | The main interface accesses the database to insert information. |

Insert Information

### Streaming Server Access



Streaming Server Access

|  |  |
| --- | --- |
|  | Load Video Session |
| Actor | Main Interface |
| Description | The server’s main interface can load a video stream into the server’s Media Session. |

Load Video Session

|  |  |
| --- | --- |
|  | Terminate Video Session |
| Actor | Main Interface |
| Description | The server’s main interface can terminate a video stream in the server’s Media Session. |

Terminate Video Session

*Database Modelling*

|  |  |
| --- | --- |
|  | Collect Streaming information |
| Actor | Main Interface |
| Include | Get Media URL, Bandwidth Usage |
| Description | The server’s main interface can collect several information from the Media Server. |

Collect Streaming information

|  |  |
| --- | --- |
|  | Get Media URL |
| Actor | Main Interface |
| Description | Access the media URL. |

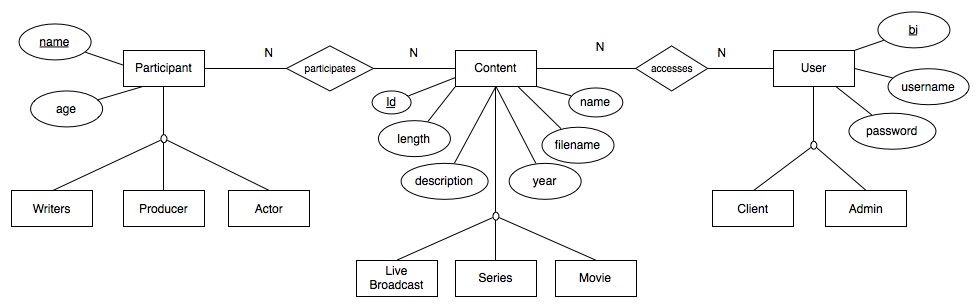
Get Media URL

|  |  |
| --- | --- |
|  | Bandwidth Usage |
| Actor | Main Interface |
| Description | Access the Media Server’s information and statistics. |

Bandwidth Usage

## Database Modelling

The most important thing the webserver will have is the media content. This content can include media streams like movies, series and live broadcast (i.e. Television). In respect to the content information we have several participants and these can be actors, producers, writers and can participate in several media content. A user can be both the client or the admin and has access to the content.



Main Interface’s Database

* Content (id, filename, name, description, year, length)
* Participant(name, age)
* User (bi, username, nome , password, email)

Associations:

* participates (Participant, Content)
* accesses (User, Content)

## Interface Specification

### Web Services API

Web services have shifted the way enterprises conduct their business nowadays. Internet it’s not just a collection of pages but a collection of services that interoperate through the Internet. This part discusses the VoD API as an interface for external services to communicate and use these web services.

User Methods:

* user.authRequest – authentication request from the user.
* user.getContentList – retrieve the list of available contents from the server.
* user.getContent – retrieve the specified content information.
* user.getMediaURL – retrieve the media URL to play the video stream.

Client Methods:

* client.createClient – register a new client.

Admin Methods:

* admin.addContent – add content information to the server’s database.
* admin.editContent – edit existing content information.
* admin.deleteContent – delete existing content information from the server’s database.
* admin.loadMediaSession – load a video stream into the server’s Media Session.
* admin.terminateMediaSession – terminate a video stream from the server’s Media Session.
* admin.getStatistics – get statistical information from the Media Server.
* admin.getUsers – get Users information form the Server.
* admin.editUsers – edit Users information.

## 