

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Software Requirements Specification

Nuno Mota

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Supervisor: Maria Teresa Andrade Prof.

Second Supervisor: André Mendes Eng.

March 6, 2011

Contents

1	Software Requirements Specification	1
1.1	System Architecture	1
1.1.1	Server	1
1.1.2	WebAdmin	2
1.1.3	Client	3
1.2	System Requirements	3
1.2.1	Server	4
1.2.2	WebAdmin	4
1.2.3	Client	5
1.3	Use Cases	5
1.3.1	Client Interface	5
1.3.2	Admin Interface	7
1.3.3	Database Access	9
1.3.4	Streaming Server Access	10
1.4	Database Modelling	11
1.5	Interface Specification	12
1.5.1	Web Services API	12

List of Figures

1.1	Server's Diagram	2
1.2	WebAdmin's Diagram	2
1.3	Client's Diagram	3
1.4	Client Interface	6
1.5	Admin Interface	8
1.6	Database Access	9
1.7	Streaming Server Access	10
1.8	Main Interface's Database	11

List of Tables

1.1	Content Access	5
1.2	Register	5
1.3	Options	5
1.4	Login	6
1.5	Verify Login	6
1.6	Play Content	6
1.7	Select Content	7
1.8	List Available Contents	7
1.9	Content Management	7
1.10	Login	7
1.11	Verify Login	7
1.12	Media Session Management	8
1.13	Upload Video	8
1.14	Manage Users	9
1.15	Manage Content Information	9
1.16	IMDB Information	9
1.17	Database Access	9
1.18	Retrieve Information	10
1.19	Insert Information	10
1.20	Load Video Session	10
1.21	Terminate Video Session	10
1.22	Collect Streaming information	11
1.23	Get Media URL	11
1.24	Bandwidth Usage	11

Chapter 1

Software Requirements Specification

This chapter will describe how the system will be implemented. It includes an overview of the system architecture in section 1.1, where it is explained in simple words the architecture and components of all the system. The system is to be composed of three major applications: the server's main interface, the webadmin interface, to control the server and the client application. All these applications must run in a Linux environment and use open-source software for development.

All the system requirements are defined in section 1.2 and then the captured use cases in section 1.3. The database present in the server's main interface is described in section 1.4. One important topic of this service is the Web Services Application Programming Interface (API) defined in section 1.5, created to provide a set of rules to use the services and resources that this application will offer. This API will have three relevant groups, the administration and the client interface and the generic user. A WSDL will be used to describe this set of rules of our web service.

1.1 System Architecture

1.1.1 Server

The server will be responsible to address all the user agents and to manage the streaming server. To make all this possible, a main interface is responsible to interconnect all the parts of the system as shown in figure 1.1. Among this parts we can include: a web services interface to communicate with both agents, administrator and client; a database to store all the client's information and video contents; a streaming server responsible to stream the video content to the clients interface through the appropriate channels.

Today, the most used approach in this kind of environment is the content delivery network (CDN). This CDNs have several servers spread around the network close to the costumers to provide a better streaming solution. Meaning that if the use of the server starts to increase, another

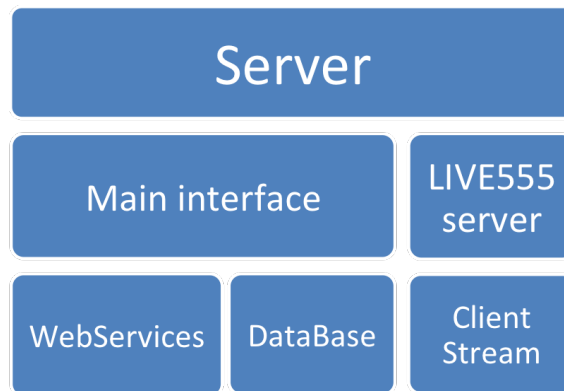


Figure 1.1: Server's Diagram

computer and network connection may be setup to provide sufficient bandwidth for a reliable service.

1.1.2 WebAdmin

To customize and configure the servers behavior and working requisites, a web interface shall be created. This environment consists of a webpage designed with HTML5, CSS3 and a PHP framework to implement all the needed tools. The interface will communicate with the server via a web services API and it's important to highlight that this type of dimensioning will give us a decentralized solution. With the admin interface, video files and contents which can be retrieved from the appropriate websites, will be uploaded to the servers local storage or database respectively. The video's information must be retrieved from official websites like IMDB, this information is required to be serialized by the admin's interface and then uploaded to the servers DataBase. An overview of the WebAdmin interface can be seen in figure 1.2

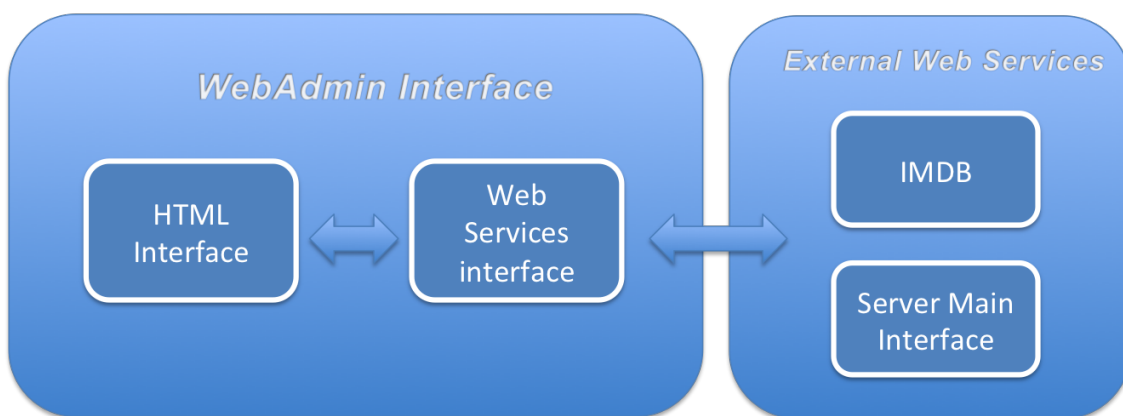


Figure 1.2: WebAdmin's Diagram

1.1.3 Client

The Client's only concern may be to properly play the video stream, but this application should use the state-of-art tools to create the best multimedia experience and usability. One of the programming tools ideal for this part of the project is the JavaFX platform because of its capability to bring a feature-rich application. Every communication will start in the client. After a successful login to the services, he will be able to search for the content available in the server. When it chooses a movie/series, all the respective information will be shown, and if it's eligible he can play the content. A client diagram can be seen in figure 1.3.

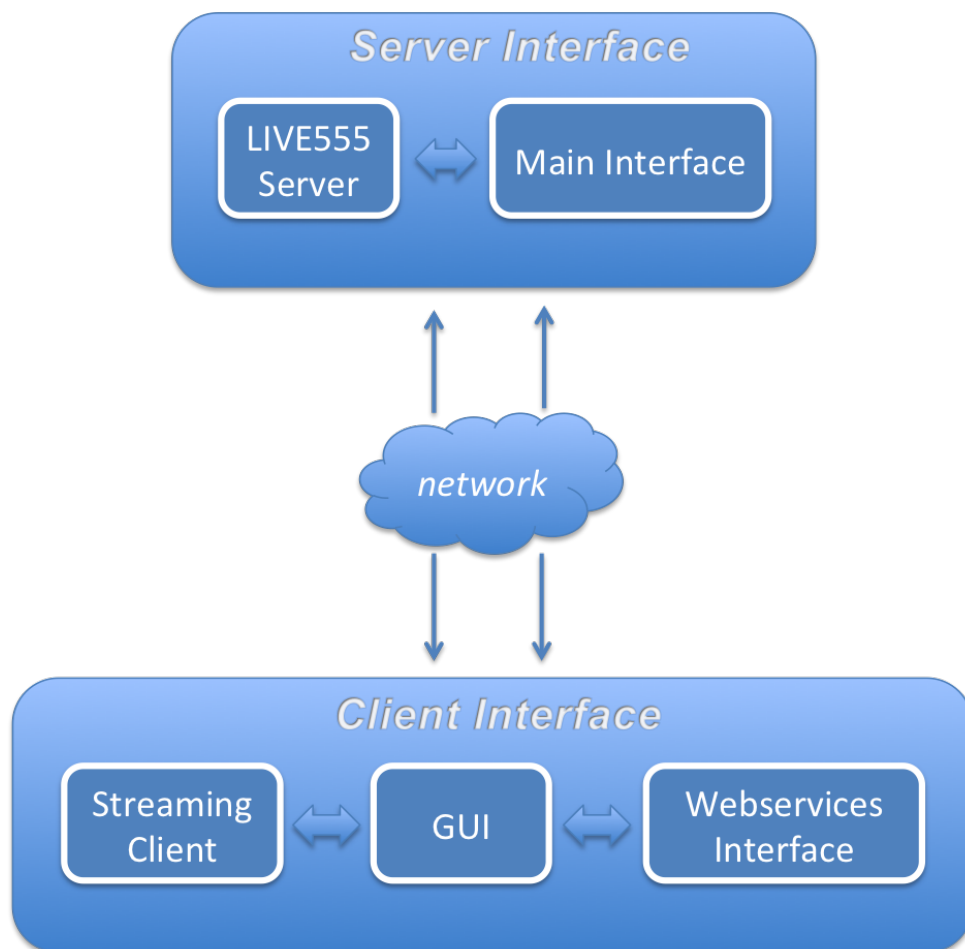


Figure 1.3: Client's Diagram

1.2 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.

1.2.1 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.

1.2.2 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.

1.2.3 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.

1.3 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.

1.3.1 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.

Table 1.1: Content Access

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

Table 1.2: Register

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

Table 1.3: Options

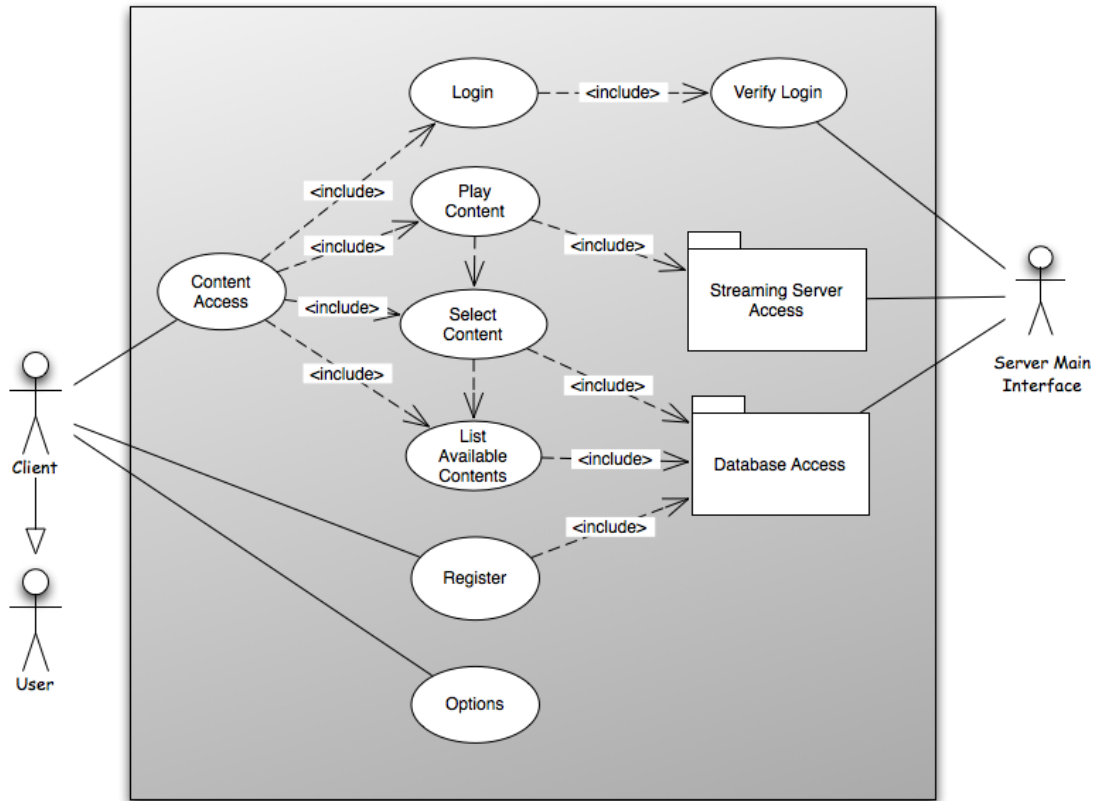


Figure 1.4: Client Interface

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

Table 1.4: Login

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

Table 1.5: Verify Login

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

Table 1.6: 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.

Table 1.7: Select Content

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

Table 1.8: List Available Contents

1.3.2 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.

Table 1.9: Content Management

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

Table 1.10: Login

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

Table 1.11: Verify Login

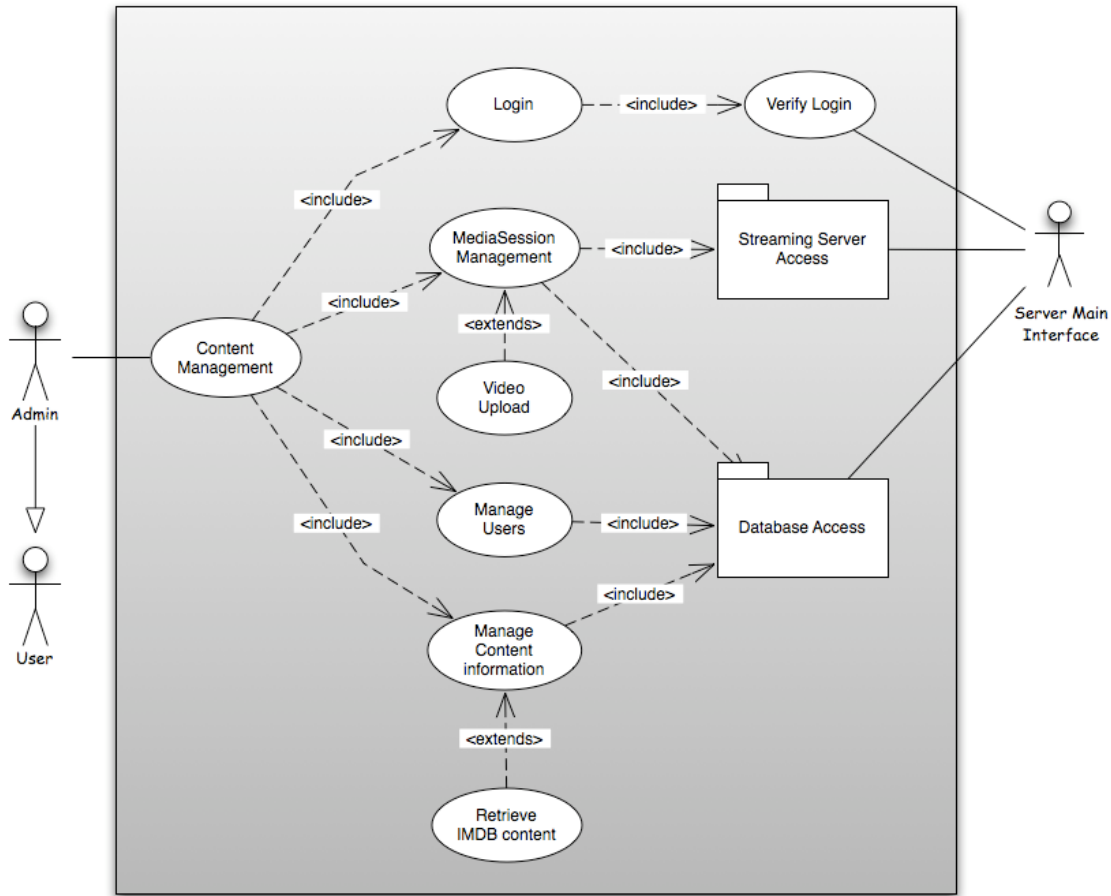


Figure 1.5: Admin Interface

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.

Table 1.12: Media Session Management

Upload Video	
Actor	Admin
Extends	MediaSession Management
Description	The admin can upload a file to the server.

Table 1.13: Upload Video

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

Table 1.14: Manage Users

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

Table 1.15: Manage Content Information

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

Table 1.16: IMDB Information

1.3.3 Database Access

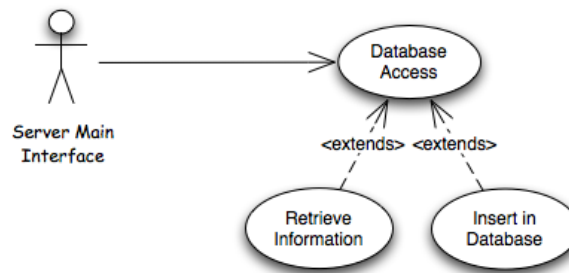


Figure 1.6: Database Access

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

Table 1.17: Database Access

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

Table 1.18: Retrieve Information

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

Table 1.19: Insert Information

1.3.4 Streaming Server Access

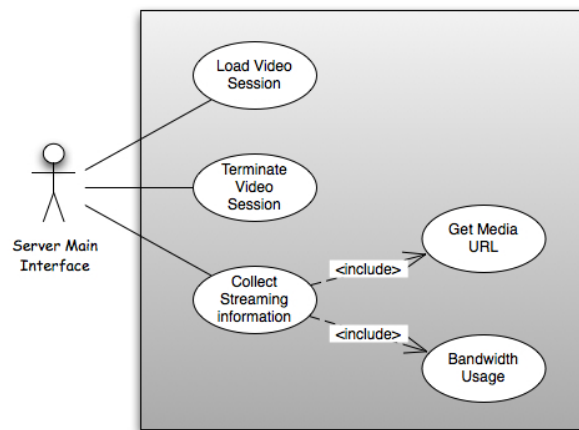


Figure 1.7: Streaming Server Access

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

Table 1.20: 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.

Table 1.21: Terminate Video Session

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.

Table 1.22: Collect Streaming information

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

Table 1.23: Get Media URL

Bandwidth Usage	
Actor	Main Interface
Description	Access the Media Server's information and statistics.

Table 1.24: Bandwidth Usage

1.4 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.

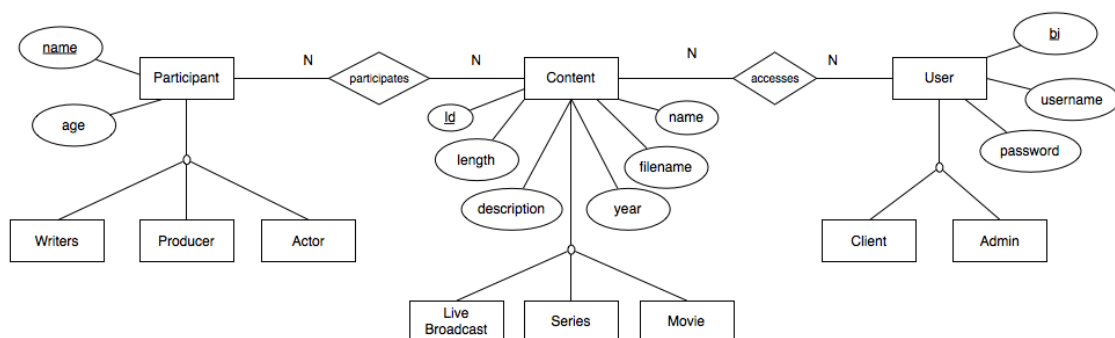


Figure 1.8: Main Interface's Database

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

Associations:

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

1.5 Interface Specification

1.5.1 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.