San Jose State University
SJSU ScholarWorks

Master's Projects | Master's Theses and Graduate Research

2009

Video Conferencing Tool

Sapna Blesson
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the Computer Sciences Commons

Recommended Citation
Blesson, Sapna, "Video Conferencing Tool" (2009). *Master's Projects*. 135.
https://scholarworks.sjsu.edu/etd_projects/135

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

VIDEO CONFERENCING TOOL

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment of the Requirements for the

Degree

Master of Science

By

Sapna Blesson

May 2009

© 2009

Sapna Blesson

ALL RIGHTS RESERVED

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

_____

Dr. Christopher Pollett

_____

Dr. Mark Stamp

_____

Dr. Sin Min Lee

APPROVED FOR THE UNIVERSITY

_____

ABSTRACT

Video Conferencing Tool (VCT) is a web-based video chat application that allows users anywhere in the world to join real-time streaming video chat rooms. This product is similar to social networking sites that allow web-based video conferencing. The main advantage of VCT compared to existing tools is that it is easy to use and does not require users to download and set up additional hardware. Since this product is a browser-based solution, it allows users from multiple platforms like Windows, Linux, or Mac to join a chat room. My VCT allows users to create new public or private chat rooms or enter into existing chat rooms with the click of a button. VCT allows users to share their live audio and video to all users in the chat room. It also allows users to see the list of attendees in the chat room. VCT users can invite their friends to join video chat rooms by sending a link to their email. Friends can click the link and directly enter chat room without creating an account in VCT. The users also have the option of sending video messages to other users. Adobe Flash Media Server is used as the back end for developing this web site.

ACKNOWLEDGEMENT

I would like to thank Dr. Christopher Pollett for his guidance.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

Over the last decade, the demand for video conferencing has increased enormously. Video conferencing is a software technology that allows group of people anywhere in the world to meet in a virtual room by simply connecting to internet using their personal computer and sharing live webcam. Video conferencing has its application in online distance learning, virtual meeting rooms in business, and social networking sites. Online distance learning allows students and teachers to meet in a virtual class room without the need to waste time and money on commuting. A virtual meeting room has significant benefits in the global market where the employees anywhere in the globe can share their ideas without needing to physically present in the company. A social networking site allows friends and families to connect and get in touch with their loved ones. Video conferencing in social networking sites allows friends and families to come together in a chat room and share their voice and video.

Some of the existing video conferencing products are “iChat”, ” ooVoo”, Stickam, Adobe Acrobat Connect Pro, etc. “iChat” is video chat software from Apple and the latest versions of Mac Operating System comes with built-in “iChat” plug-in. Problem with “iChat” is that it is platform dependent. Windows Operating System users need to install AOL instant messenger and set up the software. “ooVoo” is a social networking video conferencing application that allows users to do video

conferencing, video messaging, text messaging. The problem with “ooVoo” is that you need to download and install “ooVoo” software and also currently “ooVoo” has support only in the Windows platform. Another web conferencing application is Stickam. Stickam is a social networking site that allows users to upload their pictures and videos, to broadcast their live videos, and to do video conferencing. Problem with Stickam is the lack of privacy. Adobe Acrobat Connect Pro is a commercial video conferencing product from Adobe and it is used in business for conducting virtual meetings and in Universities for distant learning. It allows remote desktop sharing, slide sharing, and web conferencing. It is expensive software.

The problem with most of the commercial video conferencing systems is that, they have complex user interface (UI), platform dependent, high cost and require additional software installation. My goal of this project is to build a video conferencing social networking site that allows users anywhere in the world to enter virtual chat rooms. My VCT is easy to use and does not require additional software installation. The users can see list of public and private chat rooms along with user count after successful login. They can enter into chat rooms with the click of a button. Private chat rooms require users to enter password. The users can share their live audio and video to the chat rooms and they can also see the list of attendees in chat room. The users can also send email invitations to their friends to join a chat room. The friends can click on the link and enter

video chat rooms without even creating an account in the site. The users can also send video messages to their friends by sending a link to their email. Friends can click on the link and play the video message.

The main technology used in this project is Adobe Flex. Flex is an Adobe product used for the development and deployment of cross platform, rich Internet applications based on Adobe Flash platform. This project requires Adobe Flash Media Server which will work as a hub. Adobe Flash Media Server (FMS) is real time streaming media server and support client-server architecture.

The report is organized into the following Chapters. The Chapter 1 gives the introduction and Chapter 2 explains the various technologies used for implementing the project. The Chapter 3 explains VCT design and Chapter 4 gives the key concepts in FMS used in this project. The Chapter 5 explains the implementation of the project. Finally, Chapter 6 shows the testing results and Chapter 7 has the conclusion.

2. Technology Used

Adobe Flash Media Server and Adobe Flex are the main technologies used for developing video conferencing in VCT. The following are the technologies used for creating Video Conferencing Tool:

XHTML

The Extensible Hypertext Markup Language, or XHTML, is a markup language that follows same syntax as HTML, but also conforms to XML syntax. The front-end of the video conferencing web site is created using XHTML. VCT Login page and Sign-up page are created using XHTML. Shockwave Flash (SWF) file used for VCT video conferencing is embedded into XHTML page.

CSS

Cascading Style Sheets (CSS) are used to style the front-end of the VCT Login page and Sign-up page.

JavaScript

JavaScript is used to validate the user input in VCT Sign-up page and Login page before submitting the data to the back-end server-side scripts. Error messages are displayed if the required data field in the form input is left empty or if the format of input data is incorrect.

PHP

Hypertext Preprocessor (PHP) is a server-side scripting language used for producing dynamic and interactive web pages. PHP scripts are used in VCT to generate dynamic web pages. PHP script adds new user records to the back-end MySQL database during VCT sign-up. It is also used for authenticating VCT users during login. PHP script is also used as the back-end scripts for fetching VCT user accounts from MySQL database to the Flex application. VCT email functionality is created using PHP scripts.

MySQL

MySQL is a relational database management system available under GNU license. MySQL is used as the backend database for storing VCT user account.

Flex

Adobe Flex is used in VCT to client-side applications that interact with Flash Media Server (FMS). The front-end of the VCT dashboard page and video chat room page is created using Flex. Adobe Flex is used to create Rich Internet Applications. Flex applications are compiled into SWF file and embedded into XHTML page. Flex is a combination of MXML and ActionScript. MXML is an XML based language and is used to create the layout of user interface of the application. ActionScript is a scripting language based on ECMAScript language and is used to script the user interface of the application.

ActionScript

ActionScript version 3.0 is used in this project for creating interactivity to the Flex application. It is an object oriented programming language for developing Flash. ActionScript supports Video and Camera classes to process and control the webcam in VCT. The server-side ActionScript is similar to ActionScript 1.0 and it allows users to develop flexible client-server applications using Adobe Flash Media Server. Server-side ActionScript accepts or reject connections from Flash client. It allows VCT users to create new chat rooms or join the existing chat rooms. The server can send and receive data to and from the connected VCT

clients using the sever-side ActionScript code. Remote procedure calls are used to send request between clients and server.

Apache HTTP Web Server

Apache Web Server is used for hosting the VCT web application. VCT uses Apache HTTP Web Server to serve dynamic web pages and to host all the XHTML pages, PHP scripts, SWF files etc. The Web Server communicates with the web browser over Hyper Text Transfer Protocol (HTTP). The web browser requests the web pages to the Web Server and the Web Server processes the request and sends the result back to the web browser. It supports client-server architecture. Apache Web Server is available for free download in the Apache web site.

Adobe Flash Media Server

Adobe Flash Media Server (FMS) is used as the back-end server for developing VCT video conferencing. FMS is also used in VCT for recording and play back of video messages. Adobe Flash Media Server, formerly known as Macromedia Flash Communication Server, is a real-time streaming media server that supports one-way streaming for live video broadcasting and multi-way streaming for web conferencing, online gaming, interactive polling etc. Adobe Flash Media Development Server that supports multi-way streaming is freely available in the Adobe website and it supports up to 10 simultaneous connections at a time. All Flash Media Server applications are registered within the “applications” folder of the server. The figure below shows the application folder structure.

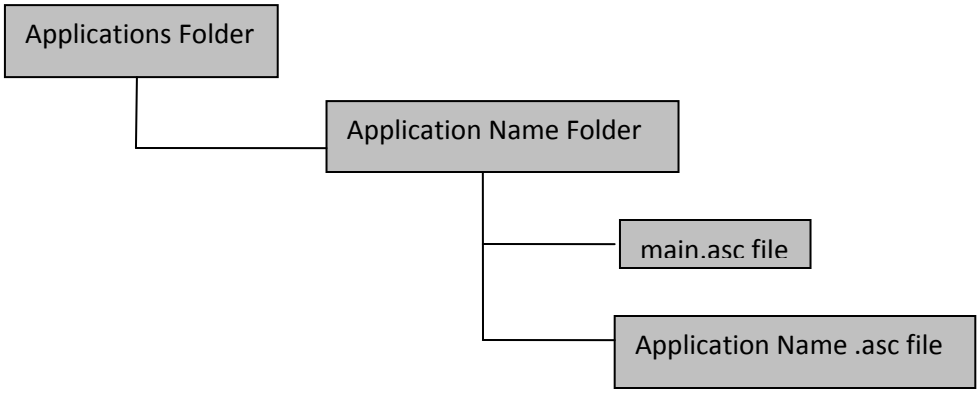


Figure 1: Application Folder Structure

The registered application is a folder inside the applications folder. Server-side script file is placed inside the application name folder. The server-side script can be named as “main.asc” file or name of the application “.asc” file.

3. VCT High-level Design

VCT design consists of three main components: Web Server, web browser with Flash player plug-in, and Flash Media Server. VCT design supports client-server architecture. VCT design consists of two main parts; the part that interact with Web Server and the part that interact with Flash Media Server. The figure below shows the client-server architecture of Web Server and web browser; Flash player and Flash Media Server.

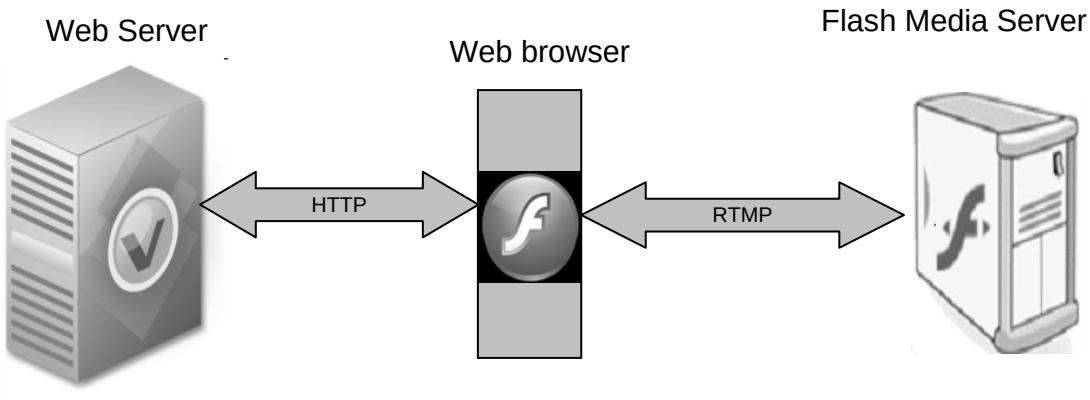


Figure 2: VCT Design

Web Server and web browser supports client-server architecture. Web browser and Web Server communicate over HTTP protocol. VCT web application is hosted in the Web Server and the web browser client request web pages to the Web Server. Web Server processes the request and sends response back to the web browser.

Most of the web browsers have Flash player plug-in installed in it. Adobe Flash Media Server has client-server architecture in which Adobe Flash player is the client and Adobe Flash Media Server is the server. The client-side Flex applications are compiled into SWF file and embedded into XHTML. The XHTML HTML file is then hosted into the web server. The Web Server sends SWF over HTTP to the Flash player. The Flash player plays the SWF file and establishes connection to the Flash Media Server using Real Time Messaging Protocol (RTMP). RTMP is a reliable TCP/IP protocol for streaming and data services. The successful connection with Flash Media Server allows data to stream between client and server in real time.

4. Key Concepts of Flash Media Server used in VCT

VCT Instances

VCT uses public and private chat room instances for video conferencing. Instances allows different group of people to access same application without interacting with other group. In this project, each chat room is connected to individual room instances and users in each chat room are completely unaware of users in the other chat rooms. The Flash Media Server applications can have multiple instances. The figure below shows instance structure.

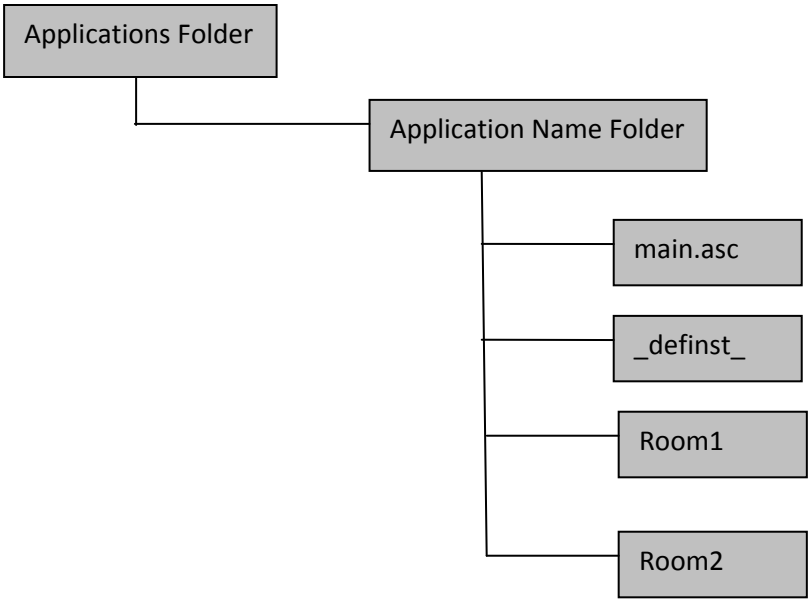


Figure 3: Multiple instances of VCT

By default, the application is connected to the default instance named "_definst_". In the above figure, the application is connected to three instances and they all use the same server-side script file "main.asc".

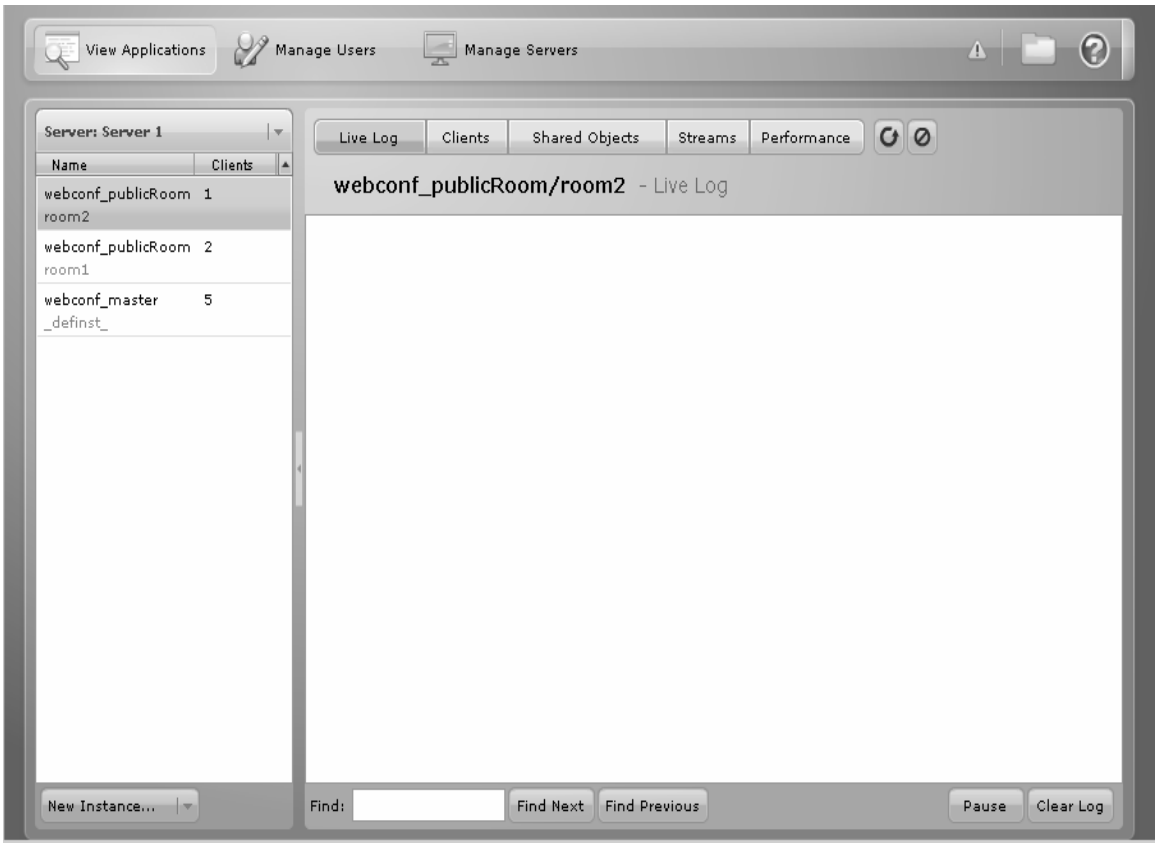


Figure 4: Active VCT Instances in FMS Console

Active VCT instances are listed in the administration console of the Flash Media Server. In the above figure, “room1” and “room2” instances of “webconf_publicRoom” application are active. Two clients are connected to “room1” instance and one client is connected to “room2” instance. Multiple instances can interact with each other by establishing instance to instance connection in the server. Instance to instance connection is referred as proxied network connection.

VCT Shared Objects

VCT uses remote shared object for storing public chat room list and private chat room list. It also uses remote shared objects for storing the list of users in individual chat rooms in VCT.VCT private chat room passwords are also stored in shared object. Shared objects are one of the powerful features of Adobe Flash Media Server. Remote shared objects are managed and stored by the server and are used for storing, messaging, and synchronizing data. Updates made to the remote shared object are received in real time to all connected clients.

VCT uses persistent shared objects for storing chat room names and users list. Remote shared objects can be either persistent or non persistent. Data stored in the persistent shared object is saved even after the application dies or reloads or reboots, but for non persistent shared object, data is lost.

VCT chat room list shared object is a proxied shared object. A proxied shared object is a remote shared object that is shared between Flash Media Server applications or between instances of same application. In VCT, master instance owns the chat room list shared object and all other room instances connect to the master instance to use the shared object.

If an application uses a shared object, a folder named “sharedobjects” is automatically created inside the application folder of FMS. The figure below

shows shared objects directory structure.

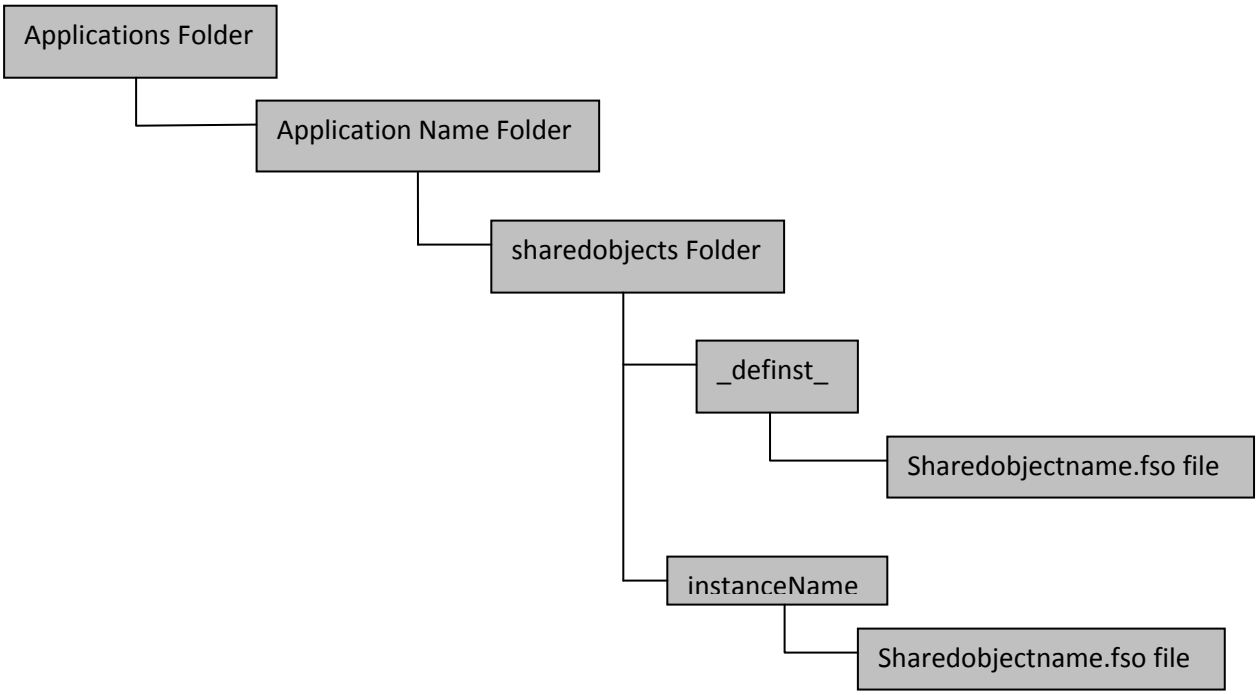


Figure 5: VCT remote shared objects storage

If the application is connected to the default instance, shared object is stored inside the sharedobjects/_definst_ folder; otherwise it is saved in sharedobject/instanceName folder. Remote shared objects are stored in a file with ".fso" extension.

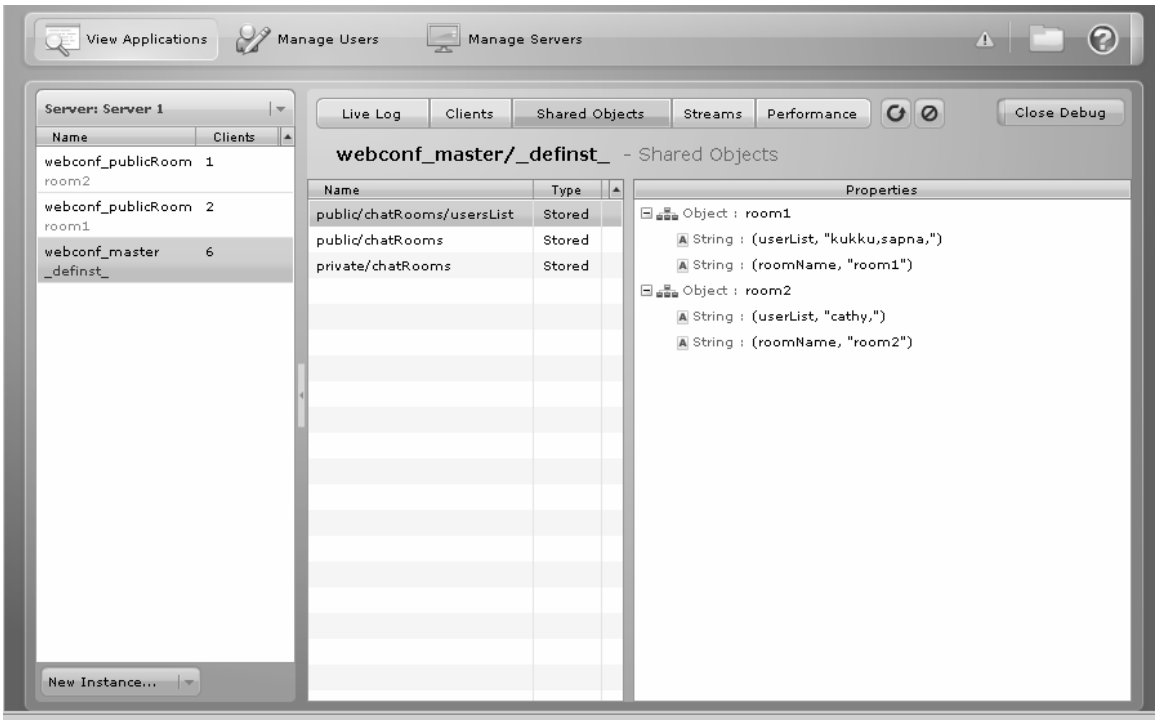


Figure 6: VCT shared objects in FMS console

We can see the data stored in the remote shared object using Flash Media Server's Administration Console. The above figure shows the name, type and data stored in "public/chatRooms/usersList" shared object. Type of persistent shared object is indicated as "Stored" and non persistent shared object is indicated as "Temp".

VCT Streams

VCT transmits audio and video streams between Flash client and FMS for live video conferencing. Real time transmission of audio, video and data between client and server is called **streaming**. The FMS server allows users to publish or play audio and video streams. VCT users publish live webcam stream using Flash Player to the FMS server and other users then play the live stream. In order to stream webcam, the stream name should be unique for each user.

Therefore username appended with room name is used as the unique stream name for user streams in individual VCT chat rooms. Not all streams are saved in server, only the streams published for recording are saved in server.

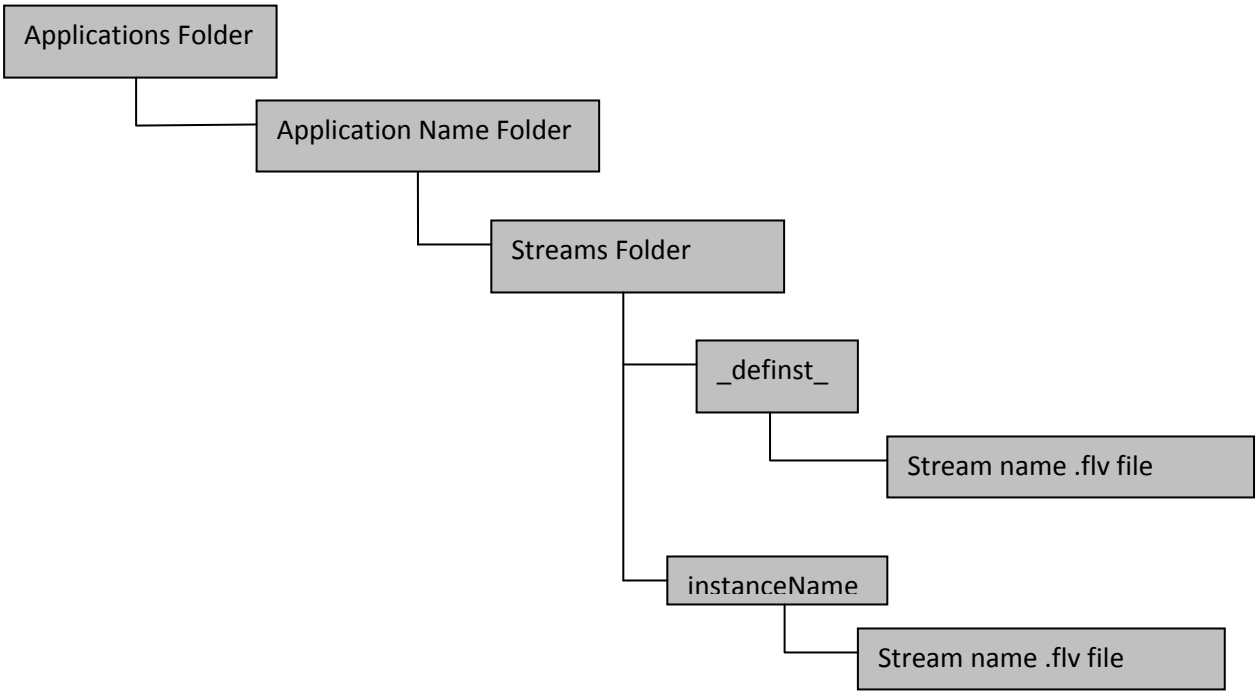
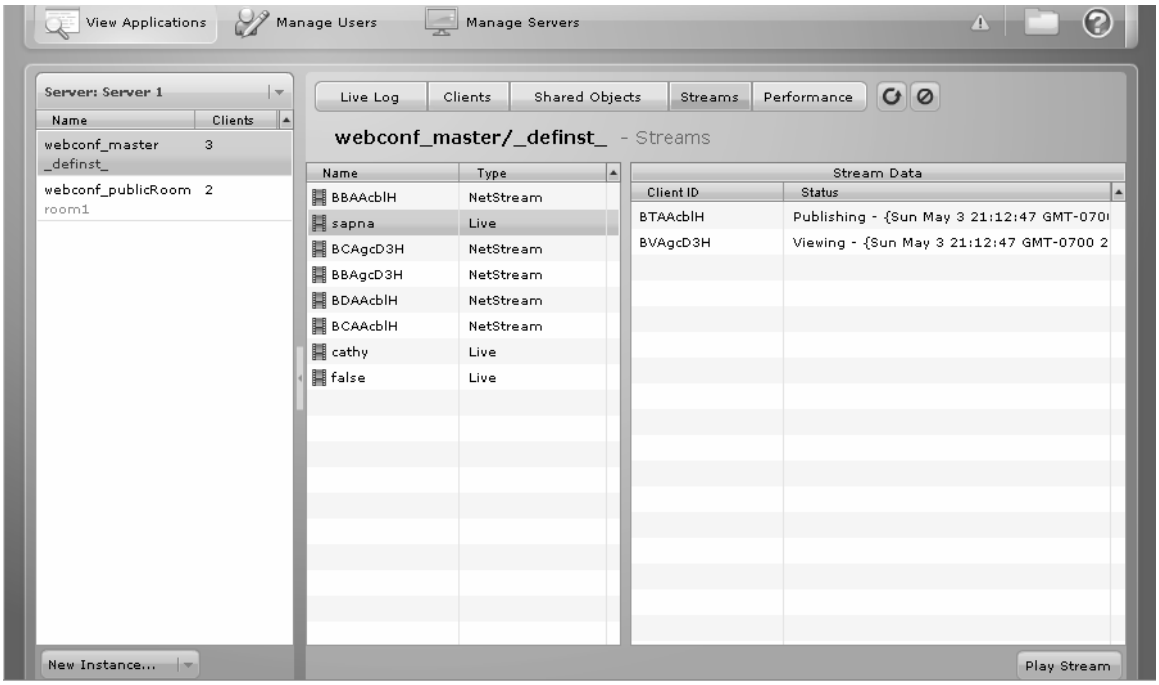


Figure 7: VCT steams storage

Streams are stored as a folder named “streams” inside the corresponding instance folder. If streams connected to default instance, the streams are stored in the folder “streams/_defInst_/”. Otherwise streams are stored in the corresponding instance name inside streams folder. Folders are created automatically by the server. Video streams are stored as Flash Video (flv) files. The figure below shows the active VCT video streams.



. Figure 8: Viewing live streams in FMS Console

We can see the list of active publishing and playing streams from administration console of Flash Media Server. The above figure shows the client ID of the publishing client and playing client of the live stream named “sapna”.

5. Implementation

The figure below shows VCT high-level page flow diagram.

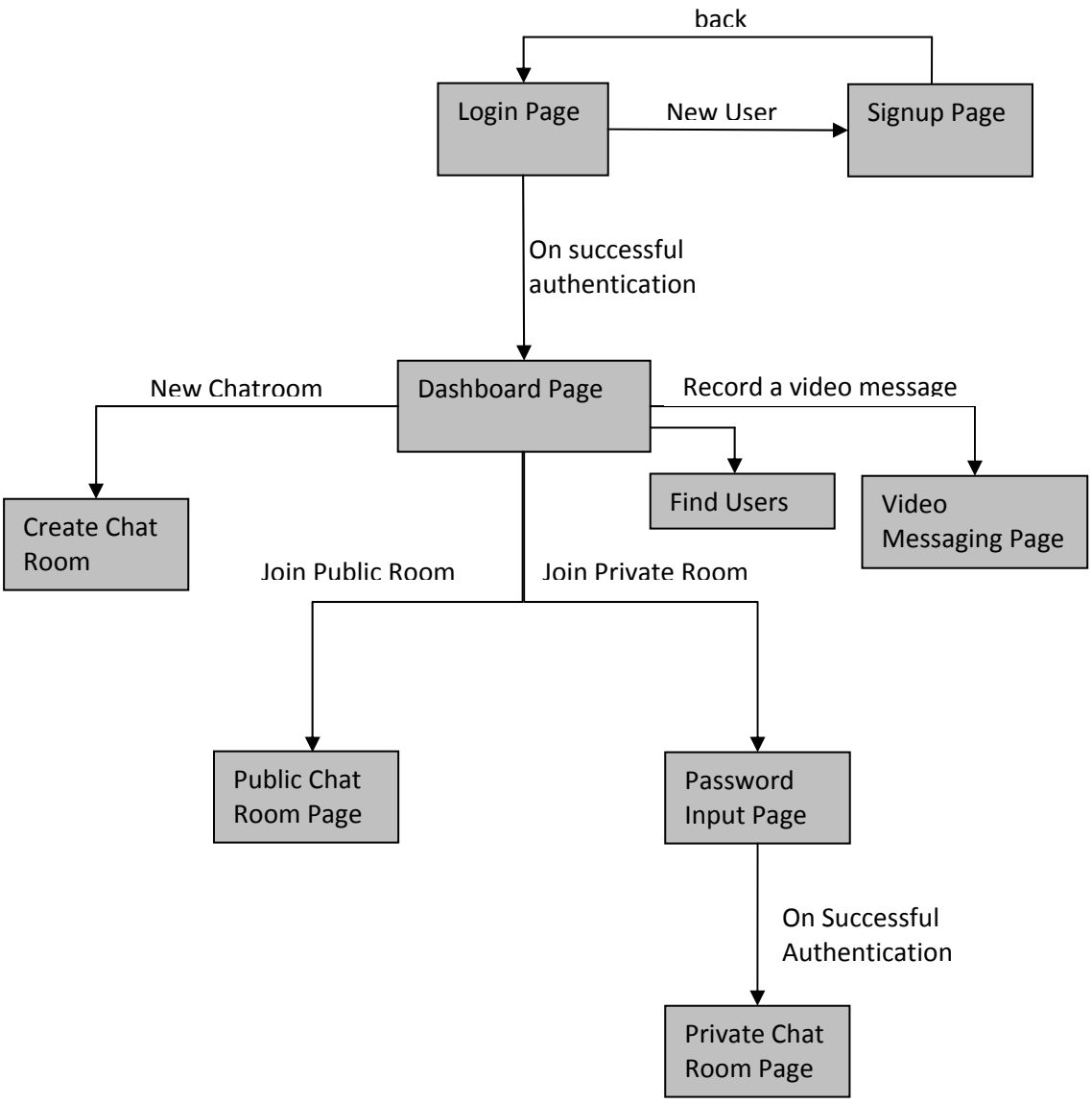


Figure 9: High level page flow diagram

VCT Login Page

Every user who wants to access VCT features must have an account in the website and an account is identified by its unique user name and password. The figure below shows the implementation of Login Page.

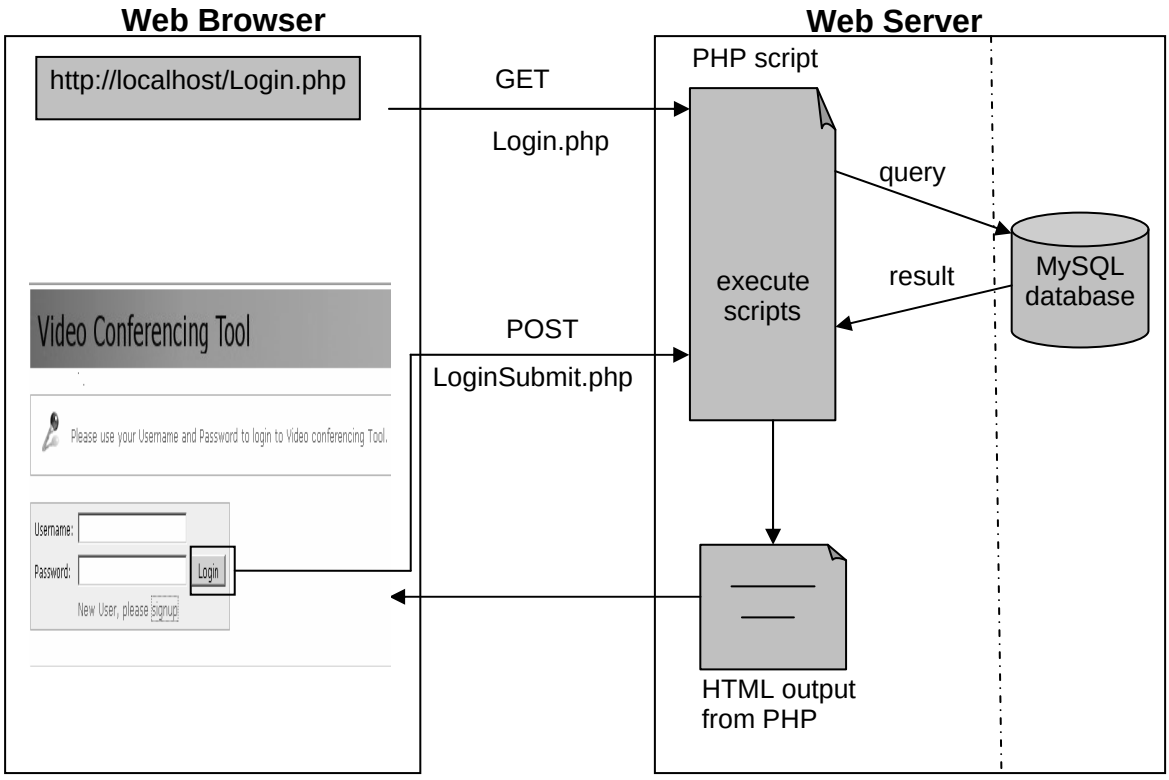


Figure 10: Implementation of Login Page

The figure shows the client-server architecture of web browser and Apache Web Server. The users who want to access the VCT home page types the Uniform Resource Locator (URL) “http://localhost/Login.php” in the web browser. The web browser client makes HTTP GET request to Apache Web Server. The Apache Web Server receives the “Login.php” request and executes the PHP script. The PHP script then outputs dynamic XHTML page and the Web Server sends the

result back to the web browser. The web browser then displays the Login Page of VCT. PHP scripts are used to generate dynamic XHTML output.

VCT Login Page provides an interface for the users to enter user name and password. JavaScript will validate the user input before submitting the data to the back-end PHP script. JavaScript is executed in the web browser and the Web Server is completely unaware of the JavaScript in the generated XHTML page. When user enters the username and password and clicks the “Login” button, the web browser makes HTTP POST request to the Web Server. The Web Server then processes the “LoginSubmit.php” script and authenticates users of VCT.

Authenticating VCT users in login

PHP scripts will authenticate the users by interacting with MySQL database. It first establishes a connection to MySQL database using mysql_connect() command and then selects the database using mysql_select_db() command and queries the data base using mysql_query() command. Each row in the result set can be accessed using mysql_fetch_array() command. The following sample code shows PHP scripts authenticating user accounts from MySQL database.

```
mysql_connect($db_hostname, $db_username ,$db_password);
mysql_select_db($db_name);
$query ="SELECT username FROM $table_name  WHERE
        username='$username' AND password='$password'";
$query_result=mysql_query($query) or
die('Error, could not execute the select query');
$row = mysql_fetch_array($query_result);
if($row)
header('Location: Dashboard.php');
mysql_close($conn);
```

If the above query returns a row, it means user name and password which user entered have a matching record in the database and the user is successfully authenticated. If there is no matching record, it will show an error message. Connection to database is closed by using mysql_close() command. If the user is successfully authenticated, the “LoginSubmit.php” script loads the “Dashboard.php” script.

VCT Sign-up Page

New users can create account in the website by clicking the “Signup” link of the home page. Signup link will redirect to signup page where users can fill the new user registration form.

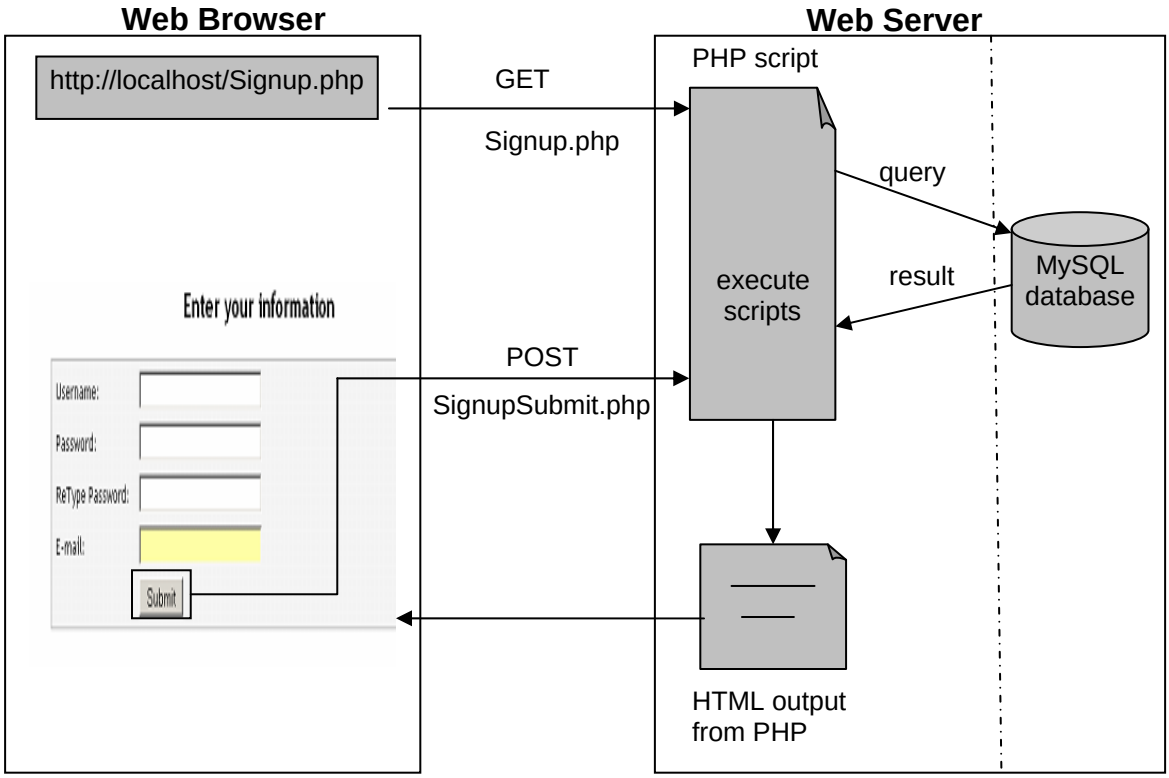


Figure 11: Implementation of VCT Sign-up Page

The web browser client makes HTTP GET request to Apache Web Server. The Apache Web Server receives the “Signup.php” request and executes the PHP script. The PHP script then outputs dynamic XHTML page and the Web Server sends the result back to the web browser. The web browser displays the Sign-up Page of VCT.

Sign-up Page of VCT allows users to enter new user account in the website. When user clicks the “Submit” button, JavaScript will validate the user input and after successful validation HTTP POST request is send by the web browser to the Web Server. The “SignupSubmit.php” script is executed by the Apache Web

Server which makes a connection to the MySQL data base and adds a new user record in the database table. . PHP script will add a new user record into the MySQL database using mysql_query() command. If the signup is successful, success message is displayed and redirect to login page; otherwise error message is displayed.

VCT Dashboard Page

The dashboard page provides majority of the functionalities of the VCT to the user. Users can create new public or private chat rooms, or join the existing chat rooms, find the list of users of the website, and record video messages from dashboard. The “Dashboard.php” outputs the dashboard web page and the Web Server sends it to the web browser. The SWF file is embedded in the generated XHML.

Embedding SWF into dashboard page

Client -side Flex application is compiled into SWF file and embedded into XHTML page. The <object> tag and <embed> tag are used to insert SWF into XHTML. The following sample code shows embedding SWF file into XHTML.

```
//Embedding SWF in XHTML
<object id="VideoChatTool" width="550" height="400"
classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://fpdownload.macromedia.com/get/flashplayer/current/swflash.cab">
        <param name="src” value="videochat.swf?userName=$username"/>
        <embed name="VideoChatTool” src="videochat.swf?userName=$username" width="550"
        height="400" pluginspage="http://www.adobe.com/go/getflashplayer” />
 </object>
```

Accessing query string in Flex

The username is passed as the query string parameter of the SWF file. This enables flash client to access username inside the SWF file.

```
//Accessing query string in Flex:

username = Application.application.parameters.userName;
```

VCT creating a new connection with Adobe Flash Media Server

When VCT users land in the dashboard, connection to “MasterInstance” of FMS application is made. The Flash client establishes connection to the FMS server using NetConnection object. Client to server connection is established by connect() method of the NetConnection object. The RTMP address is passed as the URI of the connect request. The server accepts or rejects the client connection inside the “application.onConnect” method. The client gets the status of the connection request inside the NetStatus event handler. URI of the NetConnection has the following format *[rtmp:///]host[:port]/appName[/instanceName]*. Server to sever connection is established in the same way as client to server connection using NetConnection object. The only difference is that you cannot pass relative URI for server to server connection request.

```
// Absolute and relative URI

nc.connect(“rtmp://localhost/appName”)        //absolute URI

nc.connect(“rtmp:/appName”)                   //relative URI
```

VCT Find Users Page

The list of users who has an account in the VCT website is listed in this page. It also shows the status of online or offline users by displaying online users with blue color icon and offline users with red color icon. The user details are fetched from the MySQL database to the Flex display list using back-end PHP script.

Accessing user account data from MySQL database in Flex

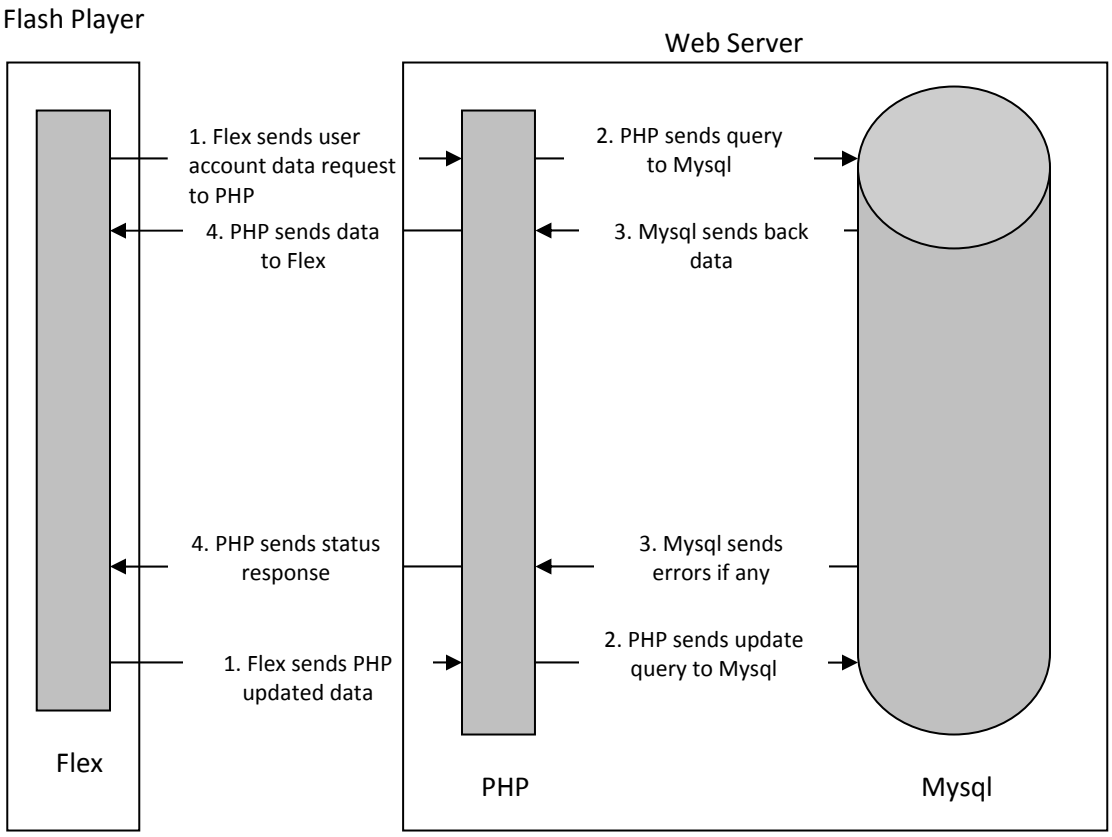


Figure 12: Flex application accessing data from MySQL database

VCT can access the user account data stored in the MySQL database using back-end PHP script. The SWF file containing the Flex application is played by

the Flash player in the web browser. The Flash player sends HTTP request to the Web Server for accessing the VCT user records. The SWF request the PHP script “Findusers.php” in the Web Server. The Web Server executes the PHP script and connects to the database and queries the database. Database sends the VCT user records back to the PHP script which then encode the data into XML format and send back to Flash client. The following sample code shows how Flex application access user accounts from MySQL database.

```
// Accessing VCT user data from MySQL database to Flex application
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
creationComplete="userDataService.send()" xmlns="*">
<mx:Script>
        private function userDataResultHandler(evt:ResultEvent):void
        {
                var xmlList:XMLList = XML(evt.result).UserAccountTable;
                xmlListColl = new XMLListCollection(xmlList);
        }
</mx:Script>

<mx:HTTPService id="userDataService" url=" Findusers.php"
        resultFormat="e4x"     result="userDataResultHandler(event);" />
<mx:TileList id="tileList" name="UserList"
                dataProvider="{xmlListColl}" height="100%" width="100%"
                itemRenderer=”CustomItemRenderer”
                 maxColumns="10" rowHeight="125" columnWidth="125" />
</mx:Application>
```

The Application object’s creationComple attribute is set to the send method of HttpService object which means the application after loading the user interface calls the send method. We set the url attribute of HttpService object to the PHP script. HttpService object sends HTTP request to the specified URL. The XML result is displayed in the TileList component. The “CustomItemRenderer” is a

MXML custom component which displays the user name and icon in the TileList component.

The “UserAccountTable” in the database stores the VCT user records. The table has a field named “isLoggedIn”. Its value is set to “true” when user login to VCT and set back to “false” when user logout. The back-end PHP script checks the “isLoggedIn” flag and loads the corresponding image which enables VCT to show online or offline status.

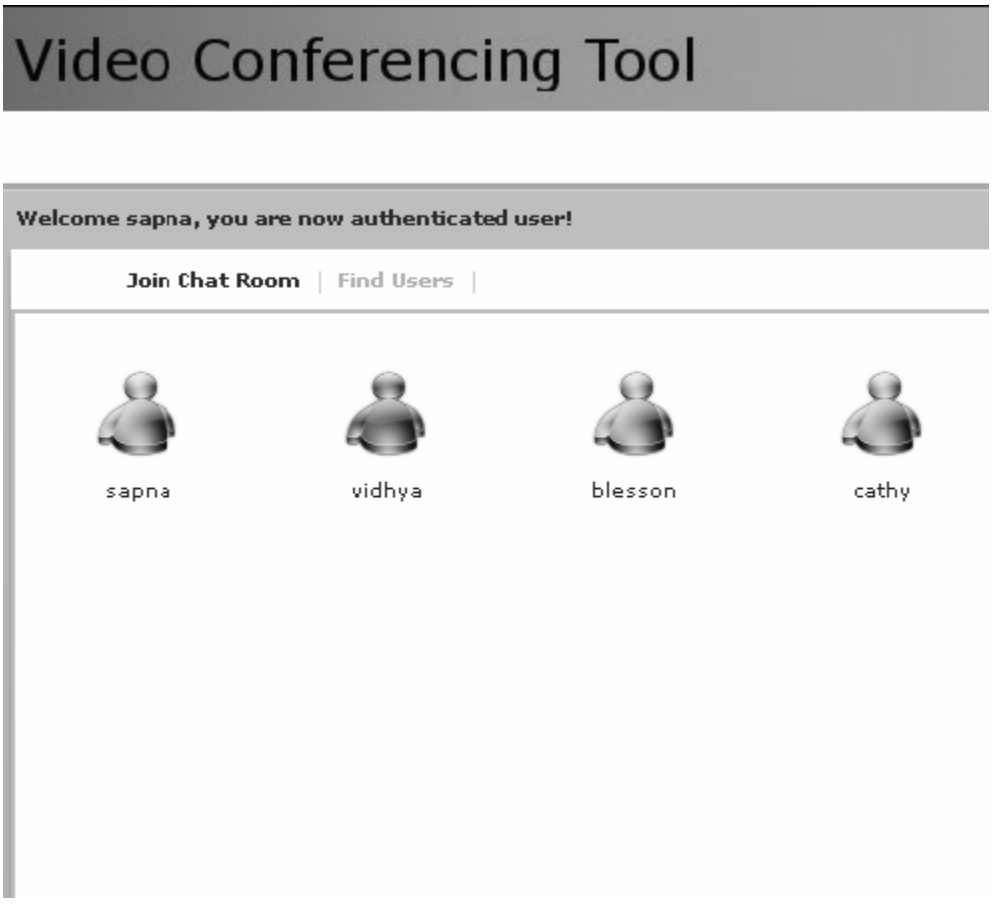
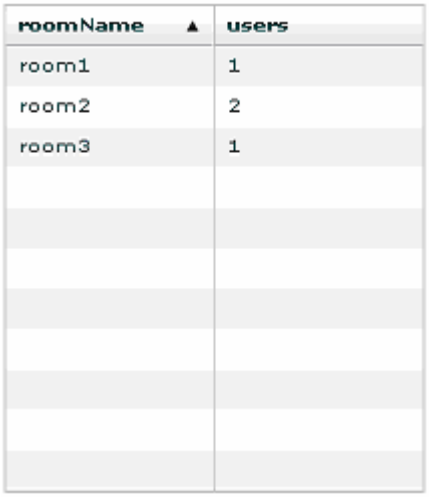


Figure 13: VCT Find Users Page

The above figure shows the list of users in VCT and their online or offline status.

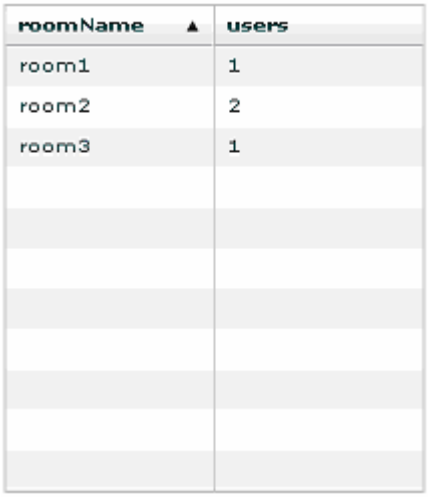
VCT public and private chat room list

The dashboard page displays a list of public and private chat rooms along with the users count. VCT stores list of chat rooms with users count in remote public and private chat room list shared object owned by the master instance of the FMS application. In-order to display list of private and public chat room list, the dashboard page first makes a connection to the master instance.



roomName ▲	users
room1	1
room2	2
room3	1

Figure 14: Public chat room list



roomName ▲	users
room1	1
room2	2
room3	1

Figure 15: Private chat room list

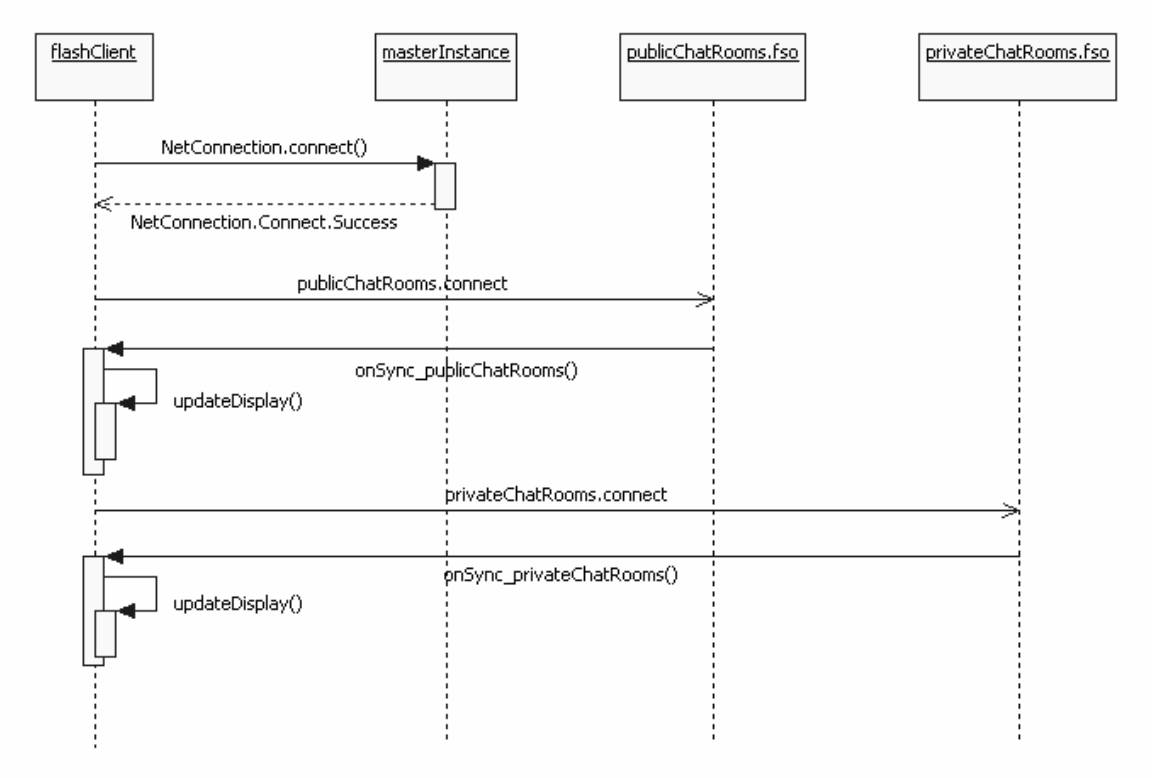


Figure 16: Sequence diagram for displaying chat room list

The above figure shows that the dashboard page after establishing successful connection to the master instance of the FMS application connects to the VCT public chat room list and private chat room list remote shared object owned by the master instance. Flash client establishes connection to the master instance of the server using NetConnection object. If the connection succeeds, client initializes shared object, adds event listener to the shared object and connect to the remote shared object. The event listener of the shared object fires synchronization event if shared object is updated or connected for the first time.

The VCT uses synchronization handler of the shared objects to get real time update of the shared object. Flash client connects to the remote shared object using SharedObject.getRemote() command.

```
//Using remote shared object
//client-side code
publicChatRoomsSo = SharedObject.getRemote(“public/chatRooms”,nc.uri,true);
publicChatRoomSo.connect(nc);
publicChatRoomsSo.addEventListenerr(SyncEvent.SYNC,syncHandler);
privateChatRoomsSo = SharedObject.getRemote(“private/chatRooms”,nc.uri,true);
privateChatRoomSo.connect(nc);
privateChatRoomsSo.addEventListenerr(SyncEvent.SYNC,syncHandler);
//server-side code
application.publicChatRoomSO = SharedObject.get("public/chatRooms",true);
application.privateChatRoomSO = SharedObject.get("private/chatRooms",true);
```

The first parameter of SharedObject.getRemote command is the name of the shared object and the second parameter is the URI of the NetConection object. The third parameter is the persistence of shared object and a “true” value indicates persistent shared object. Shared object should be connected to the NetConection object before using it. Updates made to the shared object are accessed when we attach shared object to the synchronization event handler. “clear” value of “SyncEvent” shows that client successfully connected to the remote shared object. “success” indicates that client successfully changed the

shared object. “change” indicates server or some other client successfully updated the shared object. Shared object is created on the server using SharedObject.get() function.

VCT Create Chat Room

VCT users can create private or public chat rooms from dash board page. Private chat room is created by entering chat room name and room password whereas public chat room is created by entering only chat room name.

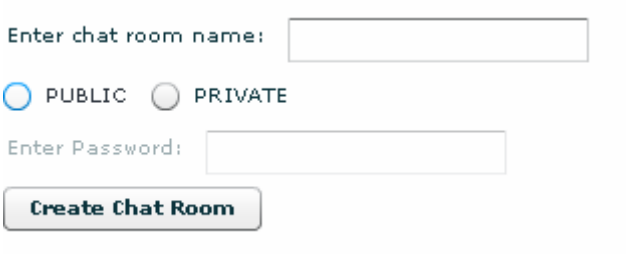


Figure 17: User interface of VCT create chat room

The above figure shows the user interface of creating a new chat room. If the user selects “private” radio button, the password input text box is activated and users can enter room password. When user clicks the “Create Chat Room” button remote method call to the FMS is made and corresponding shared object slots are updated.

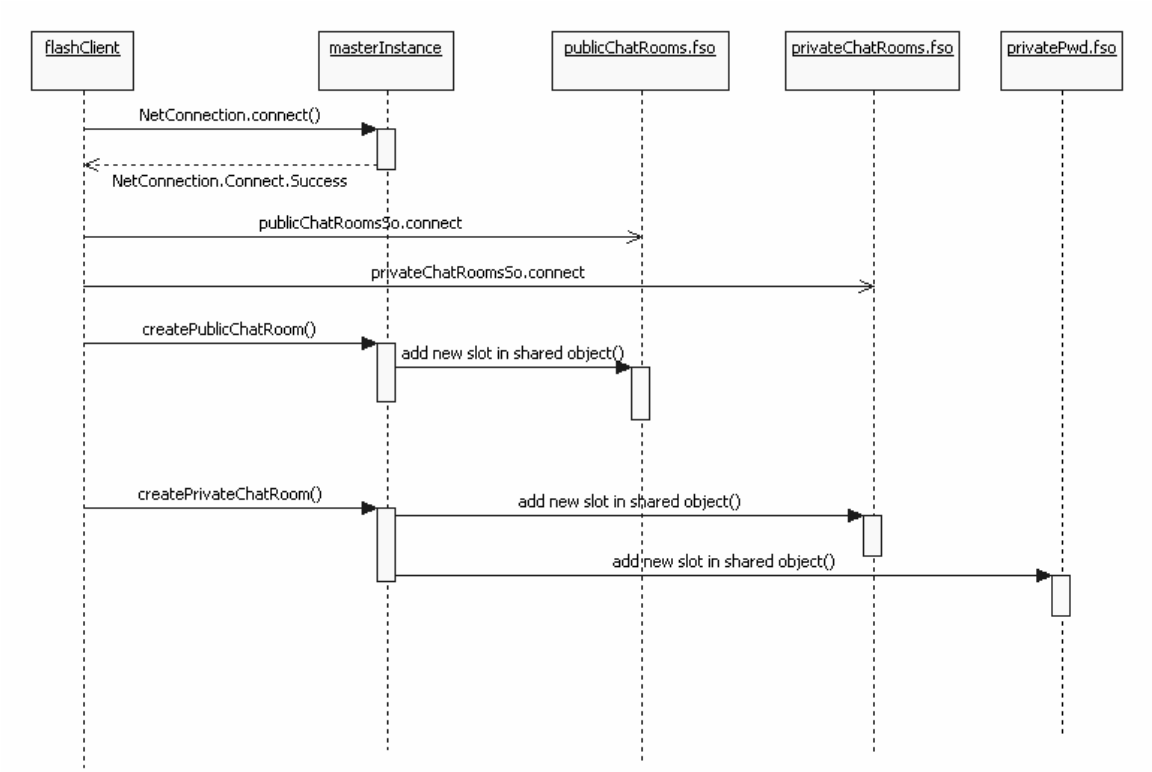


Figure 18: Sequence diagram of creating new chat room in VCT

The above sequence diagram shows the steps involved in VCT to create new pubic or private chat room. Flash client connect to the master instance of the FMS application before creating a new private or public chat room. It then connects to the remote public and private chat room list shared object. Private password list shared object is created in the FMS server for storing VCT private room passwords. When user clicks the “Create Chat Room” button of VCT, remote call to the server-side function is made. The Flash client calls server-side methods using NetConnection object’s call() method and corresponding client object invokes the call. The function name and function parameters are passed as the arguments of the call method. The code below shows how the Flash client

calls the “CreatePrivateChatRoom” method on the server. The chat room name and the room password are passed as the parameters of the function call.

```
// Flash client calling server-side “CreatePrivateChatRoom” method
nc.call(“CreatePrivateChatRoom”,null,”roomName”,”roomPassword”);
```

The “CreatePublicChatRoom” remote method call creates a new entry in the public chat room list shared object. The “CreatePrivateChatRoom” remote method call creates a new entry in the private chat room list shared object and private chat room password list shared object.

VCT Enter Password Page

When user double clicks on private chat room name in the list, enter password page is displayed. If password authentication succeeds, user enters in private chat room.

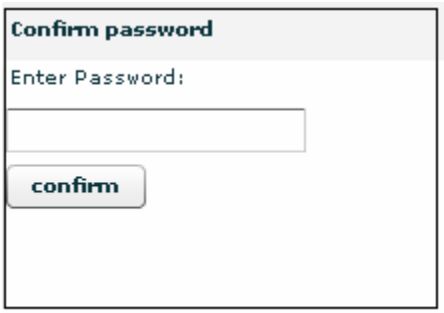


Figure 19: Enter password user interface

When VCT user enters the private chat room password and clicks the confirm button, remote method call to the master instance of the server application is made and it checks the entered password with the value stored in the private chat room password list shared object. If it matches private chat room page is loaded.

VCT Private Chat Room Page

When private chat room password authentication succeeds, the user will land in the private chat room page.

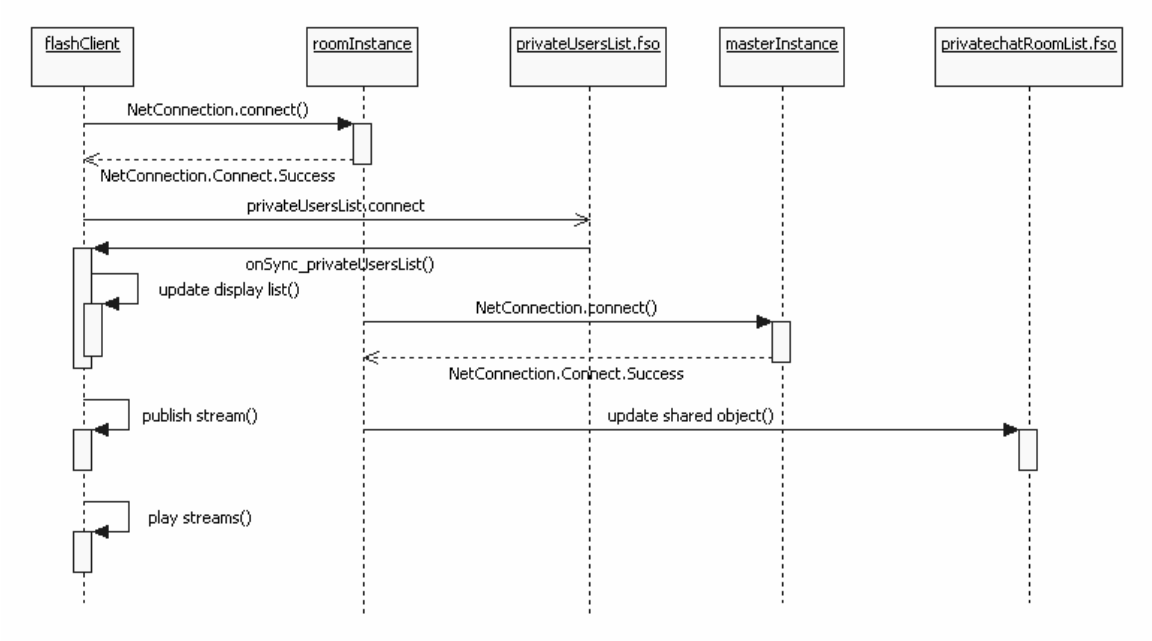


Figure 20: Sequence diagram of VCT private chat room

When VCT users enter the private chat room, flash client establishes connection to individual private room instance of the application. Individual room instances own its users list shared object. Room instances also connect to master instance

and update the private chat room list shared object. When a new user joins or leaves the chat room, room instances make remote method call to master instance and corresponding shared object is updated. The updates made to the shared object are automatically reflected in client from SYNC.EVENT of shared object. The users list shared object stored in individual room instances is used as the unique stream names for publishing and playing video streams. VCT uses user name appended with chat room name as the unique stream name for publishing and playing streams to individual chat room.

Publishing and playing streams

Flash client uses NetStream object to publish video and audio streams or subscribe to a published stream and receive data. The code below shows how to publish stream.

```
//publishing stream
publish_ns= new NetStream( nc );
video = new Video();
microphone = Microphone.getMicrophone();
camera = Camera.getCamera();
publish_ns.attachAudio( microphone );
video.attachCamera( camera );
publish_ns.attachCamera( camera );
publish_ns.publish(userName);
```

In order to publish a stream, we need to create NetStream object and connect it to NetConnection object. The Camera class captures video stream from user's webcam and Microphone class captures audio stream from user's microphone. Microphone object is attached to the NetStream using ns.attachAudio() method and camera object is attached to NetStream using ns.attachCamera() method. To view camera in the display, we need to attach camera to the video object using video.attachCamera() method. After attaching audio and video streams to the NetStream object, we can call NetStream.publish() method.

To subscribe to a published live stream, we need to use play() method of NetStream object. To view live stream in the display attachNetStream() method of video object is used. The code below shows how to subscribe to a stream.

```
//subscribing to a published stream
subscribe_ns=  new NetStream( nc );
video.attachNetStream( ns );
subscribe_ns.play(userName);
```

Multiple streams can be created for a single NetConnection object. Individual streams can either be published or played to or from the server, but the flow of data within the stream is only in single direction.

Figure 21: Private chat room page

VCT chat room page displays multiple videos of connected users in the chat room. List of attendees in the chat room is also displayed in the chat room page. Chat room page is opened as a new window from the dash board page. Users can invite new members to join chat room by sending the chat room link to their email. The friends can join the chat room by clicking the link from email. Friends can join chat room without creating new user account in the VCT web site.

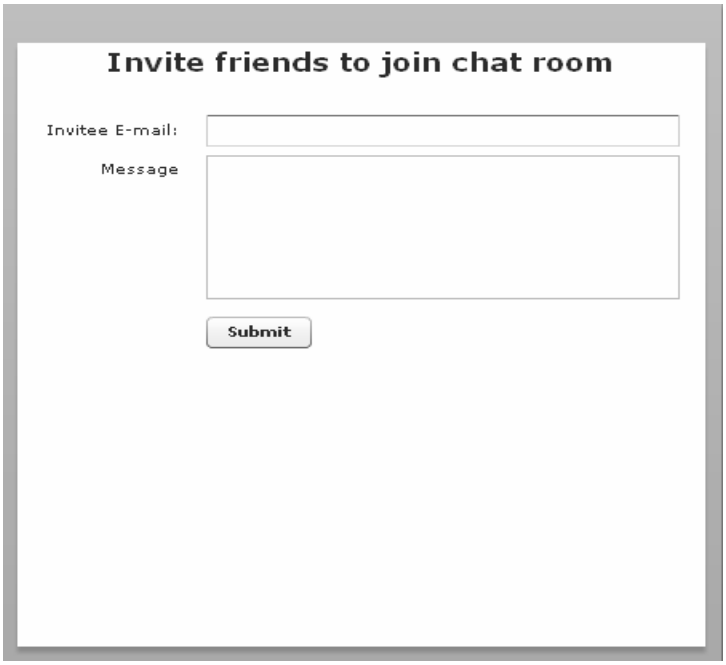


Figure 22: User interface for sending email invitation

VCT Public Chat Room Page

When VCT users enter the public chat room, flash client establishes connection to individual public room instance of the application. Individual room instances own its users list shared object. Room instances also connect to master instance and update the public chat room list shared object. When a new user joins or leaves the chat room, room instances make remote method call to master instance and corresponding shared object is updated. The updates made to the shared object are automatically reflected in client from SYNC.EVENT of shared object. The users list shared object stored in individual room instances is used as the unique stream names for publishing and playing video streams. VCT uses user name appended with chat room name as the unique stream name for

publishing and playing streams to individual chat room. The figure below shows the sequence diagram of joining public chat room.

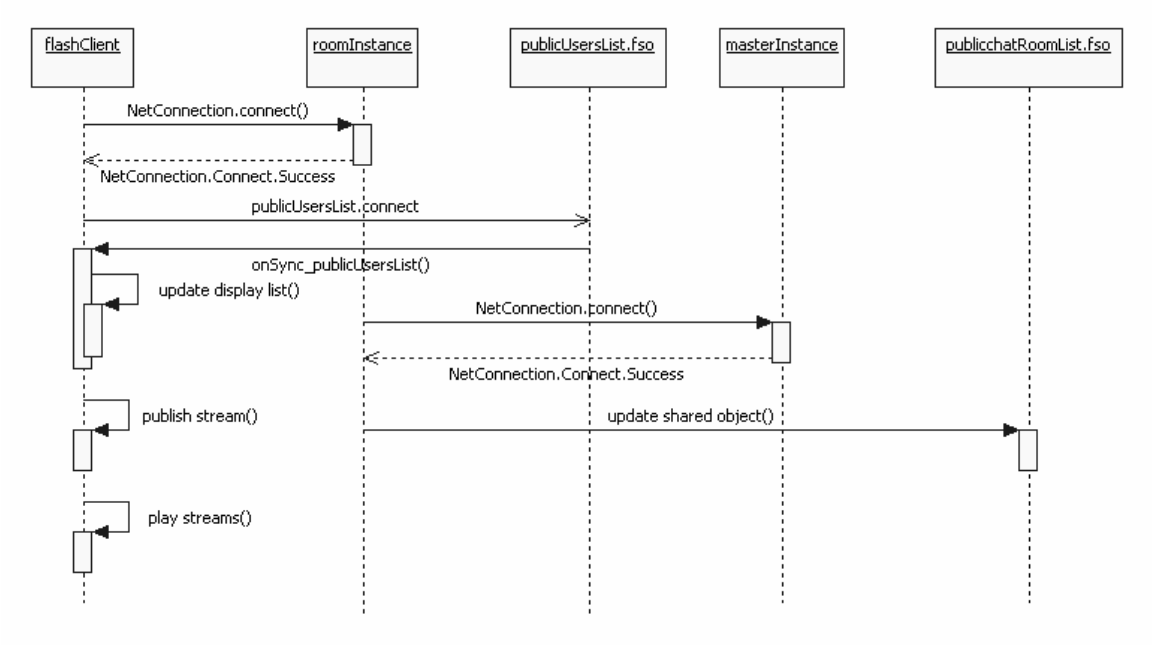


Figure 23: Sequence diagram of VCT public chat room

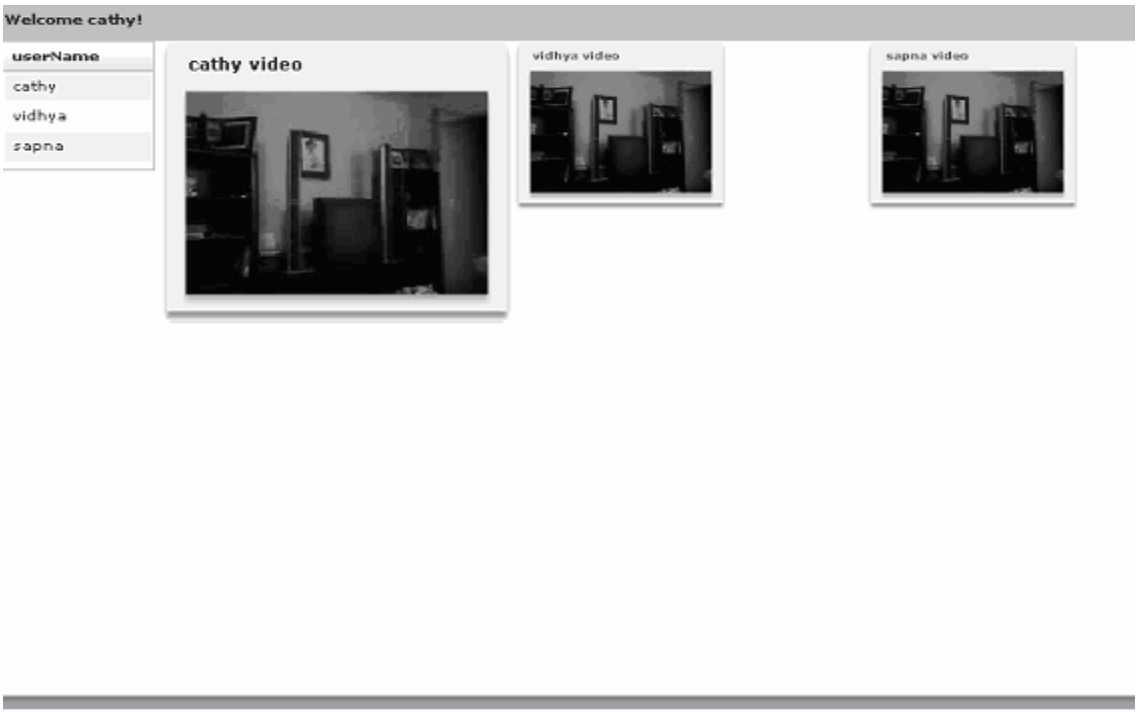


Figure 24: User interface of VCT public chat room

Record Video Message

VCT provides the mechanism to record video messages and send the video messages to others in an email. The interface allows the user to click on a user icon which will open a new window to record the video message. Once the user completes the recording, user can click on “Send Video Message” button which will send a link to the video message. The second user who receives the email can click on the link to view the video message.

Flash client can publish streams for recording using NetStream.publish() method. All video messages are stored in the server in form of flv file. The name of the flv files is a combination of user name and timestamp to make the recorded

```
//Recording stream

NetStream.publish (“StreamName”,”record”);
```

message unique. This also allows the system to retain the video messages for ever and these messages are never overwritten.

Emails are sending with the help of an SMTP server. Once the user clicks on “Send email message”, it will trigger PHP scripts which in turn call an API in SMTP server to send the video message. Users can also send video messages to users who doesn’t user account in the system.

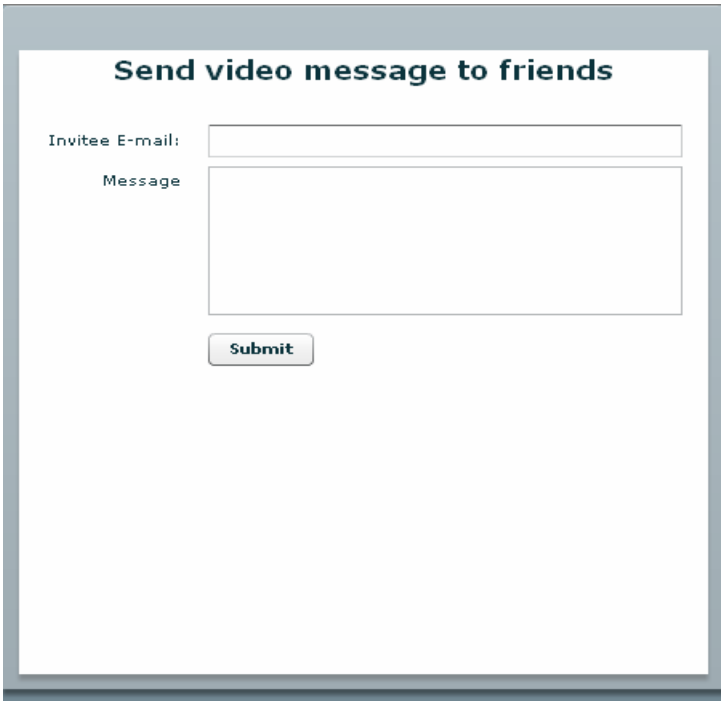


Figure 25: VCT user interface for sending video message

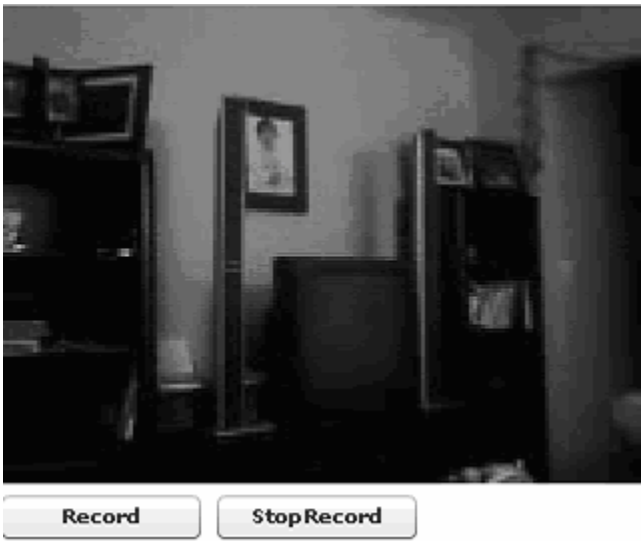


Figure 26: VCT user interface for recording video message

Figure 27: VCT user interface for playing video message

6. Testing

Performance Testing

I developed my Video Conferencing Tool using Adobe Flash Media Development Server. It is available for free download in the Adobe website and it supports up to ten simultaneous connections. Each client-to-server, instance-to-instance, and server-to-server connection is counted as individual client connection by the Flash Media Server. I tested my VCT with four users joining a chat room from a remote system. For every login request, Flash client makes a connection to the master instance of the application and every time a user joins a chat room and new connection is made to the room instance. In addition to that, room instance make a connection to master instance in-order to update the room status. If four users login 4 connections are made to master instance of the application and if all 4 users join a chat room, 4 connections to the room instance is made. Also room instance makes a connection to master instance of the application in-order to update the status of each room. Therefore a total of 9 simultaneous connections are made to FMS. Since license limit permit maximum of ten simultaneous connections, my VCT supports only up to four user videos in a chat room. The figure below is a graph showing the relationship between users sharing their videos in chat room versus the server bandwidth. The graph shows that as the number of user videos increases, the bandwidth requirement increases quadratically.

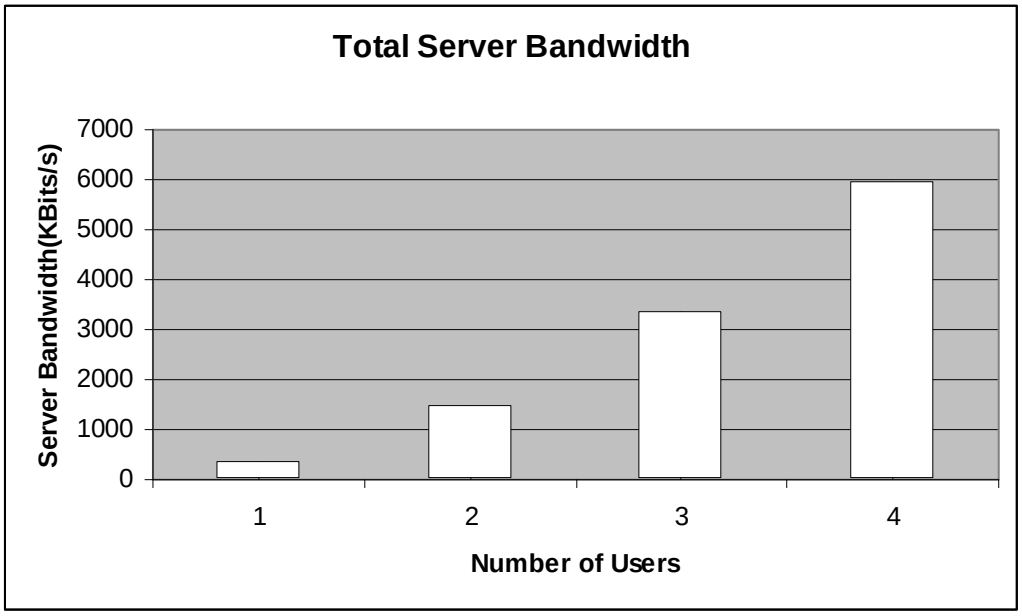


Figure 28: Bandwidth graph

Video conferencing is a many to many application in which each user will broadcast their own video and audio streams and receives the streams of others. The user publishes the video and audio streams and subscribes to other users published streams. The number of streams will increase exponentially with number of users in the chat room. For example if there are four users joining a video conference room and each of the user publishes their streams and subscribes to three other streams, it makes a total of sixteen streams.

The formula for calculating server bandwidth (BW server) = (P*N)*S, where ‘P’ is the number of publishers, ‘N’ is the number of subscribers and ‘S’ is the stream encoded at constant kilo bits per second(kbps). The formula for calculating client bandwidth (BW client) = (P*S), where ‘P’ is the number of publishers and ‘S’ is

the stream encoded at constant kbps. The server bandwidth rises quadratically with number of users in a chat room and client bandwidth rises linearly with number of users in a chat room. Also the server bandwidth drops if we limit the number of users in a chat room to two instead of four.

Usability Testing

I had tested VCT among different age groups and I got mixed results. In the age group 4-15, they couldn't learn to use VCT of their own. I need to guide them to use VCT. But once I walked through the steps, they got easily acquainted with the tool and they started loving the new tool.

In the age group 16-50, they learned the tool very fast. They found it very easy to use and learned the VCT themselves. They suggested enhancements to the tool to make it more user-friendly and more secure. Some of the suggested enhancements are richer user interface, enhanced authentication methods etc

In the age group 50 – 65, they found it little difficult to create user accounts and logging into the system. Even though I guided them multiple times, it was difficult for them to remember the steps. But they really liked the method of joining a chat room without logging in. As I described earlier, VCT provides a mechanism to send the link to the chat room in an email and the user just need to click the link to join the chat room. In this method, users don't need any expertise to join a chat room. The graph below shows usability rating versus age.

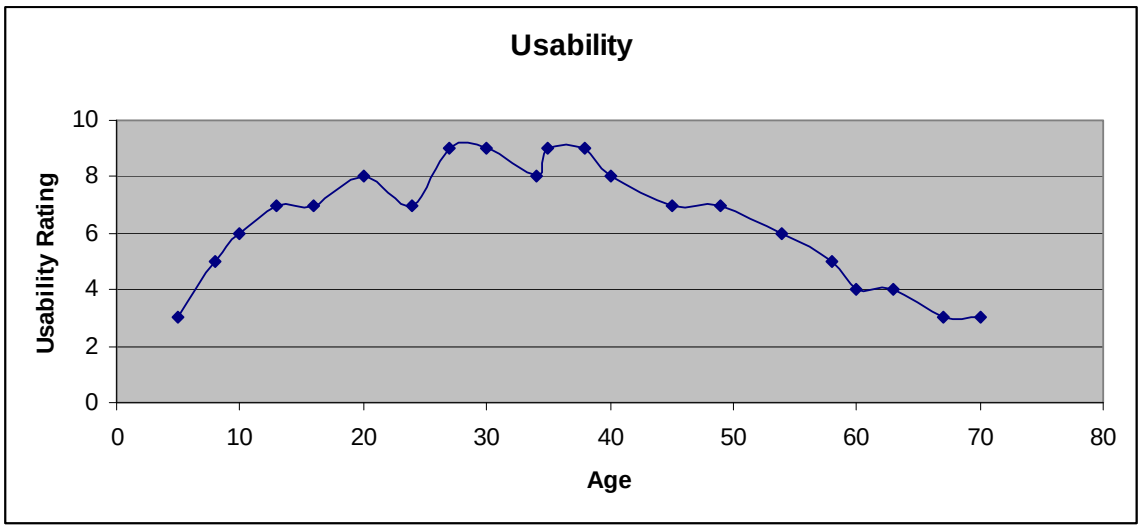


Figure 29: Usability testing graph

7. Conclusion

Video conferencing has its application in almost every field including education, business, entertainment, etc. My main goal in this VCT project was to develop an online video conferencing tool and I accomplished this goal by creating a server and client capable of multiple chat rooms with multiple users sharing their live video.

Adobe Flex, and Adobe Flash Media Server used in this project are some of the latest cutting edge technology in this field. During the initial phases of developing this project I faced several challenges in understanding some of the concepts in Adobe Flash Media Server like shared objects, multiple instances and streams. Lack of documentation was a big hurdle in understanding these technologies. I couldn’t find a good user guide that explains all these concepts in depth.

From this project, I learned a lot in building a Video Conferencing tool using Adobe Flash Media server. I hope this report will help others as a user guide for developing new tools with Adobe Flash Media server.

8. References

Lesser, B., Guilizzoni, G.,Lott J, Reinhardt R. & Watkins J. (2005*). Programming Flash Communication Server.* California: O'Reilly Media Inc.

Lott, J. (2003*). ActionScript Cookbook*. O'Reilly Media Inc.

Towes, K. (2002). *Macromedia® Flash™ Communication Server MX*. Peachpit Press.

Sanders, W.B. (2008).*Learning Flash Media Server 3.* O'Reilly Media Inc.

Noble, J. & Anderson, T. (2008). *Flex 3 Cookbook*. O'Reilly Media Inc.

Webster, S., McLeod, A. (2004). *Developing Rich Clients with Macromedia Flex*. Macromedia Press.

Hock. C. (2007). Calculating your Bandwidth and Software License Needs of Flash Media Server 2. Retrieved from Adobe

Flex Live Documents. Retrieved June 16, 2008 from Adobe: Official Site Web site: http://livedocs.adobe.com/

Flex Development Centre. Retrieved January 30, 2008 from Adobe: Official Site Web site: http://www.adobe.com/devnet/flex/

Flash Media Server Development Centre. Retrieved January 30, 2008*.* from Adobe Official Web site: http://www.adobe.com/devnet/flashmediaserver/

Flash Media Server Tutorials. Retrieved January 30, 2008*.* from Web site http://fmsguru.com/tutorials.cfm