|  |  |
| --- | --- |
|  | AShare CoNNECT  Software Design Document  Version <1.0>  Steven Prescott  2016-11-28 |
|  |  |
|  |  |



**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **People** |
| <28/Nov/16> | <1.0> | First Draft | Steve Prescott |
| <30/Nov/16> | <2.0> | Second Draft | Steve Prescott |
|  |  |  |  |

Table of Contents

Introduction

Purpose…………………………………………………………………………………….Page 3

Scope………………………………………………………………………………………Page 3

Overview…………………………………………………………………………………..Page 4

Glossary……………………………………………………………………………………Page 5

Use Cases………………………………………………………………………………….Page 6

Use Case Diagrams………………………………………………………………………...Page 7

Use Cases………………………………………………………………………………Page 8-17

Client/Server Model……………………………………………………………............Page 18-23

Class Diagram…………………………………………………………………………..Page

Software Design Document

1. **Introduction**
   1. **Purpose**

The purpose of this document is to explain the functionality of the AShare Connect video conferencing software.

* 1. **Scope**

This document covers the use case functionality of the AShare Connect view conferencing software for both host clients, including use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information.

1.5 Overview

The Software Design Document is divided into 11 sections with various subsections. The sections of the Software Design Document are:

1 Introduction

2 Glossary

3 Use Cases

4 Design Overview

1. Glossary:

Client- Client and Host

Host- A person that is give the lecture has chat, sharing desktop and audio control.

Participant-A person that is in the meeting listen to the lecture and can chat.

1. Use Cases

* 1. Use Case List

1.Join meeting

2. Chat

3. Leave meeting

4. Create Meeting

5. Entering the meeting

6. Start Meeting

7. End Meeting

8. Share Desktop

9. Create

10. Delete

3.2 Use Case Diagrams

Client

Host



|  |
| --- |
| User Case Name: Join Meeting |
| Primary Actors: Client, Host |
| Brief Description: To enter room and place name in the in the chat box text area |
| Procedure: This is where the participant and the host can enter a room by a hyperlink. The host must enter their password to use the controls. Both type their names in the textbox. Client goes to the connection site. They sign in their name in the textbox provide. Push the enter button and enter the meeting area. |
| Expected Outcomes: Client can enter the room |
| Errors: Name is not on the list. Client enter the room more than once. |

3.3 Use Cases

|  |
| --- |
| User Case Name: Chat |
| Primary Actor: Host, Client |
| Brief Description: An area on the side of the meeting area where people can communicate  with the host or each other during the meeting |
| Procedure: Host or Client type in the bottom textbox and click on the enter key. |
| Expected Outcomes: After the enter key is pushed the message goes to the textbox above so everyone can see it. |
| Errors: Once button pushed no message sent. During typing no letters are being type into the box. |

|  |
| --- |
| User Case Name: Leave meeting |
| Primary Actor: Client |
| Brief Description: Once the client click on the exit button they leave the meeting. |
| Procedure: Click on the web page exit button. |
| Expected Outcomes: Once the client click on the exit button they leave the meeting. |
| Errors: Client is unable to leave the page. |

|  |
| --- |
| User Case Name: Create meeting |
| Primary Actor: Host |
| Brief Description: Process of creating a meeting for the participants. |
| Procedure:  Log in as Administrator User.  Create a public meeting and set it to private.  Call Permissions: Add Host, Presenter, and Guest.  Create a Url for the meeting room. |
| Expected Outcomes: The meeting is created properly. Client can enter. Client is sent to the right Url. |
| Errors: The meeting isn’t created properly. Client can’t enter. Client is sent to the wrong Url. |

|  |
| --- |
| User Case Name: Entering the meeting |
| Primary Actor: Host |
| Brief Description: Host enter their meeting |
| Procedure:  From connect central click my meetings to open the desired meeting.  Click on the meeting Url that you received in your email.  Type the login username and password  Enter room or just type in Url and enter user name and password to enter room. |
| Expected Outcomes: After the user name and password was entered Host should be able to enter room. |
| Errors: After the user name and password was entered Host should be able to enter room. |

|  |
| --- |
| User Case Name: Start Meeting |
| Primary Actor: Host |
| Brief Description: Host starts the meeting |
| Procedure: Click on the start meeting button. Create an account or use existing account.  Type Url, login and Password |
| Expected Outcomes: Meeting starts as predicted. |
| Errors: Url not correct, meeting doesn’t start or password not accepted. |

|  |
| --- |
| User Case Name: End Meeting |
| Primary Actor: Host |
| Brief Description: Host end the meeting |
| Procedure: Click on the end meeting button to end the session. |
| Expected Outcomes: The session ends |
| Errors: Meeting doesn’t end when button is clicked. |

|  |
| --- |
| User Case Name: Share Desktop |
| Primary Actor: Host |
| Brief Description: Share the Host desktop in a meeting |
| Procedure: Select screen resolution  Star screen sharing button to begin  Prepare desktop for session |
| Expected Outcomes: Screen is shared to the participants |
| Errors: Screen isn’t shared |

|  |
| --- |
| User Case Name: Create |
| Actor: Host |
| Brief Description: Add new names to the meetings that can join. |
| Procedure: Adding Groups and Users can be done manually or import using a CVS file. |
| Expected Outcomes: New groups and users are added. |
| Errors: System won’t add groups and users. |

|  |
| --- |
| User Case Name: Delete |
| Actor: Host |
| Brief Description: Delete new names to the meetings that can’t join. |
| Procedure: Deleting Groups and Users can be done manually or import using a CVS file. |
| Expected Outcomes: New groups and users are deleted. |
| Errors: System won’t add groups and users. |

Design Overview 4.1

Introduction

The Design Overview is section to introduce and give a brief overview of the design. The System Architecture is a way to give the overall view of a system and to place it into context with external systems. Design is based on the Client/Server Model. Design will have scalability. Design will be easy to upgrade.

Architecture:

The architecture that we choose for this project was the Client/Server architecture because it meets our requirements very well. I came to this design because it would be easy to maintain while being reliable. Also, it would mean that we could hold large amounts of data which is required in video conferencing to handle video streaming so many people could be on at the same time.

Client/Server Model Diagram

Server

Systems Management

Client

Internet

Dashboard

Web Browser

Meeting

Password Account Diagram

Server

Account

Creation

Database

Internet

Web Browser

(Host)

Client

Add or Delete Name or Group Account

Account

Creation

Database

Dashboard

Server

Internet

Client

Web Browser

Meeting Account

Database

Dashboard

Meeting

Server

Internet

Web Browser

Client

Share Desktop Diagram

Server

Dashboard

Meeting

Internet

Web Browser

(Host)

Client

Audio Diagram

Server

Meeting

Dashboard

Internet

Web Browser

(Host)

Client

Remote Client Class Diagrams

|  |
| --- |
| Class: Connect |
| Attribute: |
| Socket:(Ip,Tcp) |
| Method: |
| Print(messge) |

|  |
| --- |
| Class: Screen Sender |
| Attributes: |
| Rectangle |
| Method: |
| CreateCaptureScreen |

|  |
| --- |
| Class: Client Connection |
| Attribute: |
| Socket: Client  ClientHandler:(Client,Destop) |
| Method: |
| Print () |

|  |
| --- |
| Class: Mouse Events |
| Attribute: |
| Mouse |
| Method: |
| MouseMove()  MousePressed()  MouseReleased() |

.

|  |
| --- |
| Class: Key Events |
| Attribute: |
| Key |
| Method: |
| KeyPress()  KeyRelease() |

**Connection Class–** this class connects the Client to the Server and tells the Client that you are connected to the Server

**Client Connection** **Class**-this class connects the Server to the Client and tell the Server that it is connected the Client.

**Mouse Events Class-** this class is used for the Host to be able to control mouse events.

**Key Events Class-** this class is used for the Host to be able to control key events.

**Screen Sender Class-** this class is used to capture the screen on the server and send it periodically to the user.

|  |
| --- |
| Class: ChatComunications |
| Attribute: |
| String: Message |
| Method: |
| set MessageType()  get message() |

|  |
| --- |
| Class: Chat Endpoint |
| Attributes: |
| Websocket  Chat pkg |
| Method: |
| ChatEndpoint() |

**Chat Communication Class-** this is a class that allows the Server and client to chat back and for.

**Chat Endpoint Class-** this class is used to disconnect from the chat room for both the Client and Server