**Interim report**

Project title:

**Remotely controlled advertisement unit**

Group members:

Arbaz Ahmed (59008)

Muhammad Ammar(59051)

Mohammad Hamza (58998)

Monis Azhar (59485)

**[S.M.Ammar Ali (59051)]**

**Start**

**Overview:**

This project is a remote desktop app. Client and server correspondence is at the introduce of organization provisioning over the Internet. Client sends request to multiple servers with the help of middle ware. Middle ware control multiple servers dynamically and servers usually wait to accept invitation from client side with the help of middle ware, maintaining connectivity or handshaking. Client connect to servers and provide ads list to play on servers. Sockets is introduce for basic communication between client and servers. Clients can run ads priority base remotely on servers.

Design of our project is complete of both side’s client and server until now. Back end implementation will complete in coming weeks. Project is divided into three parts client, serverand middleware. We will submit project on three releases on Github. Work is assign into group members given below:

* Ammar and Arbaz working on client application.
* Monis working on middleware application.
* Hamza working on Server application.

**Analysis:**

It is about inspection of user interface according to tools we will use that how server contain media list on the server and played it according to time duration. How client create , update and upload media files and send it remotely to the servers. And how the video files will update and upload file option will be show on client interface the backend code and message oriented or remote procedures were used as a middleware layer.

**End**

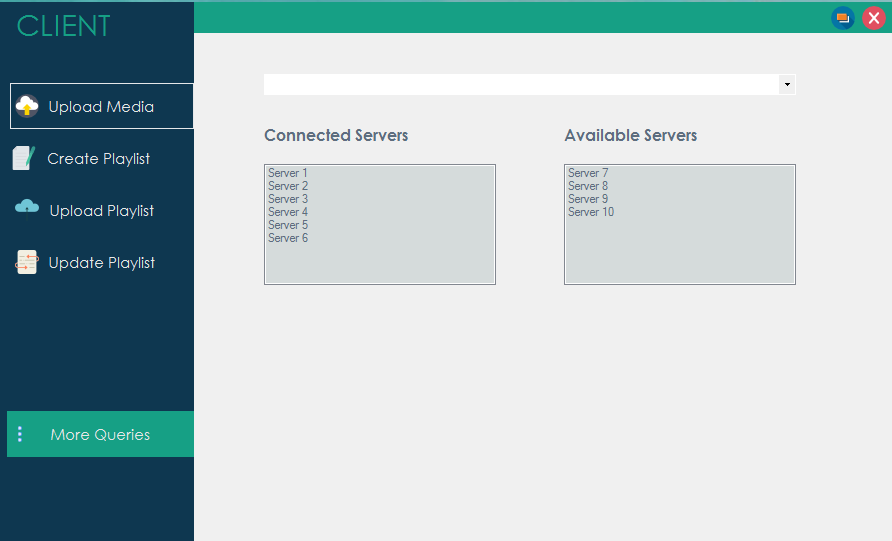
**[S.M.Ammar Ali (59051)]**

**[Arbaz Ahmed (59008)]**

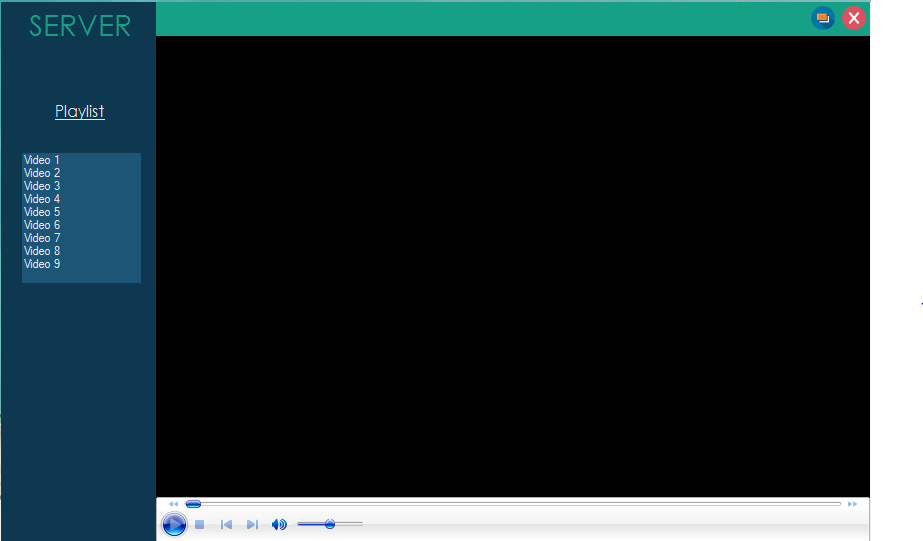
**Start**

**Features:**

* Client can upload Media
* Can create playlist
* Can upload playlist
* Can update playlist
* Client will show connected servers.
* Client will also show available servers.



* Server shows Playlist created, uploaded and updated from client side.
* Streaming ads on server from playlist with respect to time duration.



**Planning:**

We will using Tcp client and Tcp server for middle ware so it will work as distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

A server works as playlist streamer for several clients, that will handle playlist requests. Our client interface is connects to the server to provide playlist. The client provide a user interface that allows users to carry out actions. It forwards these requests to the server, which carries out the action and returns a response.

We will also setup FTP server on middle ware for the transfer of computer files between a client and server on a computer network. FTP is built on a client server model architecture using separate control and data connections between the client and the server.

**End**

**[Arbaz Ahmed(59008)]**

**[Muhammad Hamza (58998)]**

**Start**

**Overview:**

servers and clients are connted through middleware layer. all connected servers though middleware will be displayed to client. client will select server to send messages.

client select the type of message eg. create\_video\_playlist, upload\_video\_playlist, file\_upload etc. client can also select the video time to run and what video may play next etc.

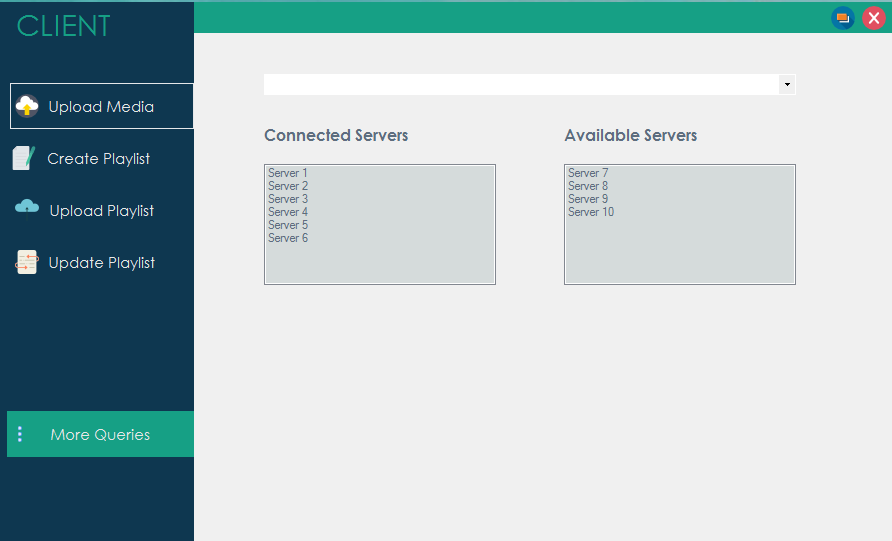
**Requirement gathering:**

Requirement gathering phase of the project describes "what" features in this project must have like ads were streaming on multiple servers by client and client can create, update & upload playlist. Client also display connected servers and available servers on its interface. Whereas server will play media list which is remotely get from client.

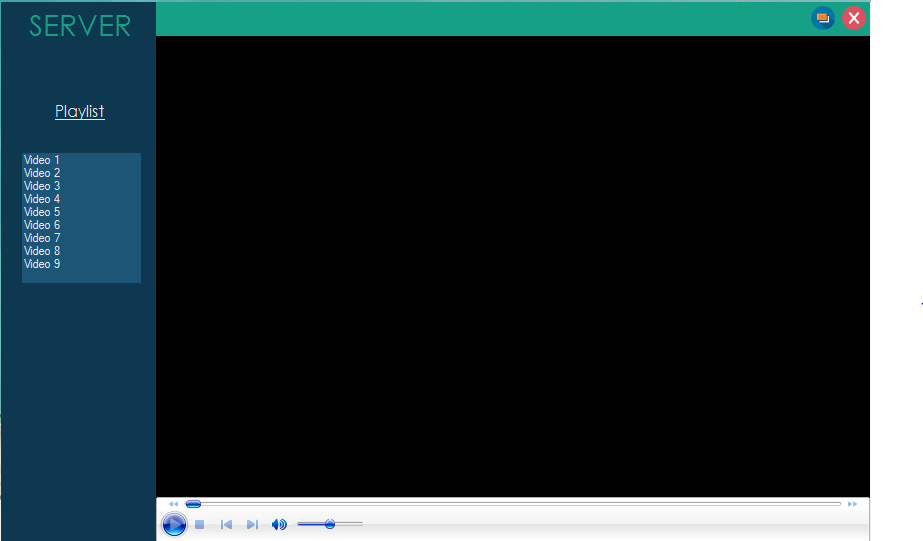
**Design:**

The design phase of the project contain the graphical user interface or view of the project and its events.

* **Client:**

****

* **Server:**

****

**[Muhammad Hamza (59889)]**

**End**

**[Arbaz Ahmed (59008)]**

**Start**

**Working:**

Server side code take concern with the client request, Code running on the server machine enable the server to response. Tcp / ip server client model were used to make reliable connection in between server and client and for middleware we were message oriented or remote procedures. Handshaking were maintain to verify the acknowledgement of message transfer.

**End**

**[Arbaz Ahmed (59008)]**