**User Manual**

***Compiling and Installation:***

Standard GStreamer and GTK+ packages were used for implementation. Version 0.10.30 of Gstreamer was used.

***Server:***

Once the server has started then it can be left alone. Nothing else is done with the server unless some feedback is required. Then just observe the server terminal for session monitoring.

***Client:***

Connecting

Once the client opens you will have a UI with video controls and stream settings. To connect to the server you have to click the Play button which will connect to the server and start streaming the video.

Playback

Once the video has begun playing you can affect the playback in different manners. There is a Pause button which will pause the stream until Play is selected again. There is a Rewind button which will start Rewinding the video. And there is a Forward button which will start Forwarding the video.

There is also a choice of resolution. You can decide whether you want to have a 240p resolution or a 480p resolution from a drop down menu.

Stream Control

There are a few options that allow you to control the flow of the stream. The first is the ACTIVE and PASSIVE modes. ACTIVE mode will play video at 25 frames per second or 15 frames per second depending on you bandwidth and play audio at 8 kHz. PASSIVE mode will just play video without audio at 10 frames per second.

You can also control the bandwidth. You type in the bandwidth available to you (in B/s) and click update bandwidth. This tells the server how much bandwidth you have available to you which affects how many frames per second you receive in ACTIVE mode.

**Development**

***Server***:

The server was what created the first half of the pipeline. It had the video source and encoded and then sent it out in the stream using udp. It had a resources.txt file that allowed to know how much bandwidth it has and that allows it to allocate resources correctly to multiple clients. It also reads in how much bandwidth a client has to figure out how much frames it should send per second in ACTIVE mode. In PASSIVE mode it will just send a constant 10 frames per second. It also has rate control in the pipeline to take care of dropped frames.

***Client:***

The client completes the second half of the pipeline. It receives the stream, decodes it and plays it on the UI. It tells the server how much resources it has available, what resolution to send, and whether it wants the stream in ACTIVE or PASSIVE mode.