Statement of Purpose:
The purpose of this application is to allow handicap students access to the same media as their peers regardless of their disability. This application will allow students to access content from Apple.com Trailers and YouTube.com's most popular videos through a constrained one button click and hold interface. This application will allow users to start, stop and skip videos by clicking a configurable button.

Project Requirements:
• Single key interface
• User configurable key
• Simplified user actions (start, stop and next)
• Supervisor configurable content restrictions (violence, language, etc.)
• Access most popular movie trailers from Apple.com
• Access most popular content from YouTube.com

Project Architecture:
The architecture of this project is divided into two different major parts, the UI and the content management.

The UI is very simple and consist of three main states, initialization, loading and playing. The initialization state is where user’s will select the key they wish to use through the course of their session. The initialization state will also allow Supervisors to configure age based content restrictions on the content the user can browse.

The loading state where he UI will display animated geometric patterns so that the user is entertained while the app searches for new content and caches it for uninterrupted viewing. The loading state is encountered once after initialization and before the next video is started. The user cannot skip this state, as it is required for the video to properly load.

The playing state is very simple in that it presents the VideoDisplay object for media playback. The playing state is where the user is given control of the interaction, they are allowed to start, stop and skip to the next video using the key they’ve configured.

The second major part of this project is the content management; this refers to all of the code actually interfacing with providers. Each provider will have class that will poll for content information and build a play list that the user can watch. The code in the loading state that actually interfaces with these classes after they have been initialized will be able randomly select videos from different content providers ensuring a very wide variety of videos.