In short order I will begin development of my class project. In order to facilitate this endeavor I will first need to breakdown what it is I will be doing. This project consists of the design and development of an application. For my specific project I have decided to create my own bare bones video sharing and streaming service. By the end of this document I will have defined exactly what I mean by a video sharing and streaming service. As well as, identified the purposed functionality and required resources that will make up the completed application. In order to lay these out in a clear and concise manner I will present them in terms of the MVC (Model, View, Controler) architecture.

But; before we can talk about the models, views, or controllers; we first need to get a broad level idea of what will be required. Because the entire basis of the application revolves around sharing content, it will obviously require users. Users need to be able to upload videos, search for other user's videos, comment on both their own and other user's videos, and users should be able to rate each others videos. Now that we have a broad level idea of what we are doing, we can explore the application in detail.

We now come to the models. We will of course need several models to complete this application. First of course, is the users. Users will consist of a User ID, a profile name, an optionally secret email address, a secret password, and a short “about me” block of text. Users will have a one to many relationship with their videos. Videos will have a Video ID, a title, a date uploaded, a video length, a keywords section (to facilitate search queries), a brief description of the videos contents, and a rating field that reflects other users rated score of the video. Users will also have a one to many relationship with their posts. Posts will include the profile name of the poster, a hidden relational field that ties it to a specific video, a second optional hidden relational field that will tie it to another users post (as a reply), a date/time posted, the contents of the post, an upvote tracker, and a downvote tracker. Further, users will have a one to many relationship with their video ratings. Each rating will have an associated Video ID, User ID, and a numerical rating.

We then come to the views. I will only discuss views briefly because many aspects of the views are likely to change throughout the development process. The views and their aspects that are likely to remain mostly static are: The homepage, which will consist of a logo/banner, a search bar, a navigation pane, and a feed of the most recently uploaded videos. The user view, which will serve as a users profile page and contain their profile name, “about me” description, a search bar, a navigation pane, and a library of their uploaded videos. And finally, the video view, which will consist of the the video occupying the majority of the window space, a search bar, navigation pane, a mini feed for related videos, a rating bar, and comments section.

Lastly we come to the controllers. Users, during account creation will need to provide and verify an email. The confirmation email process should be automatic. Search functionality will be facilitated by utilization of keywords, partial or complete titles, or partial or complete profile names. Sorting functionality should exist on both the user's view and the video search's view. Users should be able to sort by relevance (where applicable), date uploaded, or highest or lowest rating. There are of course a plethora of controllers that could be listed here, but this section of this document simply endeavor to lay a baseline for the functionality required for successful implementation of this proposed application.

At this point, I have defined and scaffolded my intended application for this project and explained the basic functionality it should provide. This document gives a simple and concise idea of what each of my models should look like, the basic layout of the application's presentation has been defined, and rough explanation of the application's functionality has been constructed. With this Document as a guide post, I have a clearly laid out path to follow and will be able to maintain course along my development journey.