This document serves as a project overview for William Powers backtrader V2 software. The scope defined here will help to organize the entirety of the project and clear up confusion in the relationship and tasks of the many moving parts.

The three questions that this document seeks to answer are as follows:

1. What will the data feed to the operate class look like?
2. How will incrementation take place throughout the whole program?
3. How will strategy class communicate back to operate when an order needs to be made?

To begin the software as a whole consists of three main components, data feeds, strategy, and operate. Operate will act as the main driver of the program, it will receive a incrementing data feed from the data feed and in turn feed that to a strategy class. If during this time the strategy class returns some kind of order based on its personal algorithm, the operate class is in charge of all the specific details in executing that order. As of right now td ameritrade seems to be the best place to make that happen.

Operate must be able to control the incrementing process flow, whether that be pulling from a historical feed or receiving from a live data feed. Possibility of creating two similar operate classes with the same ability to connect to strategy but have slightly different ways it pulling data as it does not seem that both the live and historical ways will be able to be accessed in the same way.

The strategy classes will be an assortment of different programmed trading algorithms with all the same input and output commands. They will be fed data piece by piece (the same as operate) and determine what to do with it from there. In some way, strategy needs to be able to send order commands to operate cleanly and with out stopping either of their process flows.

The data feed classes will be broken down into two components as well, the first being pulling live data feeds, and the second being historical data feeds. Live data feeds will be pulled from whatever source whether it be td amertirade or similar, and then instantly fed into operate in dictionary format. Historical will pull historical pricing from a various sites such as Bloomberg or yahoo finance or similar and store it an efficient manner to be pulled from impersonating a live data feed.