The purpose of the program, wherein the primary controller is found in the UserInterface.java class, is to obtain live market quotes from the "Investor's Exchange" free API and make notional trades depending on an indicator called the stochastic oscillator. There are three main elements comprising this project: the user interface (UI), a class built to obtain the stochastic oscillator (a specific indicator) of a given quote, and an API caller (retrieves live quotes). The user interface (UserInterface.java) has several options: a "Refresh" button, filters, a search box, and a "Display Trades" button. The "Display Trades" button checks the ETFs available and decides whether or not to trade and displays the results. The filters represent additional capability that could be added to the program, such as being able to trade commodities and bonds. The "Refresh" button notionally updates a quote if one is highlighted. The search box allows one to search for a specific ETF.The StochasticOscillator.java class has a singular function: determine the result of an equation designed to retrieve the stochastic oscillator for a given quote. It does this via a method called "calcStochasticOscillation. The API caller has several classes: URLTranslator.java, WebsiteDataReader.java, Quote.java, and AutoTrader.java. The purpose of the URLTranslator is to stitch together the various URL components required for a free, live quote by passing the ETF (Exchange Traded Fund in question, a.k.a. the stock ticker) to request the "high", "low", "open", and "close" prices. This class accomplishes this via two methods: "setSandbox" and "convert". "setSandbox" allows one to test the class by using a specialized test call to the API, and "convert" uses the ETF given to generate the appropriate URL. An instance of this class is created in the WebsiteDataReader.java class, whose purpose is to use this URL to physically connect to the website and retrieve the required data. It does this with another two methods: "getData" and "readConnection". "readConnection" merely connects to the URL given by the URLTranslator and returns the text received. "getData" uses this result by passing it to the Quote via creation of one. Additionally, it sets the ETF variable in the same quote. Quotes (Quote.java) represent information on an ETF at a particular time frame. This class includes a high, low, open, and close price; the ETF it represents, a "k" variable that represents the stochastic oscillator, and a String called "csvText" that stores comma delimited information on the quote. The Quote has several methods, mostly getters and setters, but of interest are the constructor (takes a String of output text given by the WebsiteDataReader and converts its information into the variables of the class, as well as sets the results given by the StochasticOscillator.java class to "k") and the overridden "toString," which prints the Quote's ETF and high, low, open, and close variables. The AutoTrader.java class utilizes the WebsiteDataReader to retrieve these quotes, which it stores in ArrayLists organized by the ETF they belong to. This class also stores the thresholds for which an acceptable trade can be made (based on the stochastic oscillator) as doubles organized by ETF. It is written to support repetitive trading on its own accord in the main method, which utilizes the "decideTrade" method. This method merely returns whether or not the stochastic oscillator is higher than the tolerance as defined earlier, and then notionally makes a trade based on this result. The last method in the AutoTrader class, "saveData," merely creates a csv file and prints the trades made, organized by ETF.