My primary goal for this capstone course is to pursue a project that is academically stimulating and interesting to me. To satisfy my first constraint, it needs to be something that I have little direct experience in. Secondly, I have to *want* to do this project. I have dabbled with building a trading algorithm in the past, but nothing to this scale or my planned degree of precision. I was greatly interested in the topic, but it was unfortunately shelved due to a lack of available time. I would like to put in a considerable amount of time into the development of this project, and ideally extend its lifetime far beyond the duration of these senior design classes.

Several courses that I have taken will assist with my development and soft skills while working on this project. I have broken this project out into three different areas in my head, and there are a few courses that can assist with each: low level technical challenges, architectural design, and creating requirements. CS2028C Data Structures, CS2071 Discrete Computational Structures, and CS4071 Design and Analysis of Algorithms have given a lot of knowledge about low level problem solving and taught how to develop efficient algorithms while taking things like time and space complexity, graph theory, set theory, and memory management into account. Next, the class that will primarily help with architectural design is CS5170 The Theory of Formal Languages and Automata. This class taught to think in terms of states, rather than just one line of code jumping to the next. This is a bit more abstract than the classes listed before, but it will provide a great deal of assistance when designing higher level processes. Lastly, some other supplementary courses that will assist with requirements are ENGL4092 Technical and Scientific Writing, EECE3093C Software Engineering, and STAT2037 Probability and Statistics. Technical writing and software engineering have taught how to create and keep documents around a project, as well as how to communicate with stakeholders. Software engineering also showed me how to formally develop requirements, which will be very useful early in the project's development. Statistics will be of use here due to the subject matter of financial data.

As far as co-op experience, I have held two different positions; I worked in both as a software engineer. My first position was at Medpace, which I accredit much of my technical ability, development strategy, and professional behaviors. While first starting, adjusting to more professional behaviors and practices was something that I needed to work on, and my former supervisor was a great help in this regard. We would consistently have meetings, and occasionally would go over things I could do or say differently to either be more efficient with my words or be more respectful in ways I hadn't thought of. I also had a very diverse array of projects to maintain and develop features for in areas like ASP.NET CORE Web Apps and APIs, Azure Active Directory Authentication, and UI development. My current position is with Triage Partners, where I do largely the same type of development. I am using the full Microsoft suite, just like Medpace, and the position is quite similar in most aspects except scale of the company – Triage Partners is much smaller. My most significant takeaway from my work so far is that I've been able to apply much of my knowledge from the aforementioned courses to really fine tune my development skills. By nature of the position, many of the requirement-related courses I listed are not applied as much, but concepts from algorithms, automata, and data structures are heavily used throughout my day-to-day. Since both positions used similar technical stacks, I plan to apply the skills I have learned onto this project, primarily using the full Microsoft suite for development.

I have been involved in the stock market for quite a while now, and I have always found the underlying math very interesting and learned quite a lot about it on my own time. I always look for opportunities to integrate my passion of computer science into my other hobbies (chess bot, wordle bot, sudoku bot). Generally, this greatly expands my interest in them as I develop a greater understanding of the underlying mechanics, theory, and concepts. This project is no different for me. I have several algorithms, some common and some that I created myself, that I will initially use as the base for requirements and design approach. Most of these are made up of more abstract ideas that I can work towards, which I call base indicators.

It should be relatively easy to verify that each of these base indicators work after the foundation of the project is built out. Math can be done by hand to determine if they are displaying correct results over time. Since I am working alone, my accomplishments are dictated by my advisors and myself. I may never "finish" this project, but for the purposes of this course my main goal is to host a platform that can sustain a profitable (depending on the market of course) algorithm. I feel that backtesting an algorithm against historical data is enough to show capability if live data does not work as well. Another key point of success is that this needs to be able to read live data as granular as a single trade. Potentially hundreds of thousands of trades per second will require a lot of thought being put into optimization.