

Table of Contents

[**1.0 Introduction 3**](#_8joljz8ewbtk)

[1.1 Goals and Objectives 3](#_k2v16jwjtlfa)

[1.2 Statement Of Scope 3](#_qbrhu6s7hfwl)

[1.3 Software Context 4](#_b8mkeok2n2bw)

[1.4 Major Constraints 4](#_rgire51sxpyp)

[**2.0 Product Design 5**](#_tcl8rdz8xucs)

[2.1 Overview 5](#_c8wsn2e21d8v)

[2.2 User Interface 5](#_malf2r884qui)

[**3.0 Architectural Design 7**](#_v3atbmpsb0se)

[3.1 Algorithm 7](#_r5tj1wq1fqqg)

[3.2 API Usage 8](#_1ljjrmhi4wak)

[**4.0 Restrictions, Limitations, and Constraints 10**](#_vzedgwv4tv3k)

[4.1 Restrictions 10](#_n18j72hu1o4d)

[4.2 Limitations 10](#_z8r6y8sa7bzd)

[4.3 Constraints 10](#_6yo85eik8jdu)

[**5.0 Testing 11**](#_v5f6q8ptcp48)

[**6.0 Revision Log 12**](#_1wqxzm7xphu1)

# 

# 1.0 Introduction

Code Craft has been given a task to create an application for computerized stock trading, as well as create our algorithm that will be used in the application. This document will outline all the requirements for this task.

## 1.1 Goals and Objectives

The goal of this project is to create an application that will help the user participate in stock trading. The algorithm that the team will create will be based on Trend Trading. Within the application, there will be three sections: Data Acquisitions, Strategy, and Trading Platform.

## 1.2 Statement Of Scope

Code Craft has been given a synopsis of the tasks they must complete to create an application for computerized stock trading:

* Creating an algorithm based off on Trend Trading
* Gathering data acquired from a real-time or near real-time streaming stock ticker feed, an RSS news feed, an Artificial Intelligence Large Language Model (e.g. ChatGPT), etc.
* Alpaca - our main Internet Trading Platform used to simulate a real trading platform)

The software project is being undertaken to develop a stock trading application based upon an algorithm using data acquired from an real-time or near real-time streaming stock ticker feed, an RSS news feed, an Artificial Intelligence Large Language Model (e.g. ChatGPT), etc. Our application will be executed any day from 9:30 am to 4:00 pm. Using Alpaca as our Internet Trading Platform, Code Craft will be funded $100,000 to use for our stock trading application. Project deliverables include documentation, individual and group timesheets and contributions, meeting minutes, and the software application itself. The group, Code Craft, will review and approve final documentation and application status before publishing. This project will not include option trading, and will only take a long or short position in a stock.

## 1.3 Software Context

| **Software** | **Description** | **Type** |
| --- | --- | --- |
| Google Drive | Cloud-based storage service that allows the team to store and collaborate on documents. | File Sharing |
| GitHub | Web-based version control repository to keep track of code versions. | Version Control |
| Python | The Programming language that the application will be created using. | Language |
| Jira | Allows tasks to be created and assigned to team members to be completed for each iteration. | Application |

## 1.4 Major Constraints

Implementation: Learning how to properly invest and sell stocks as well as learning the different stock trading platforms (Investopedia, E\*TRADE, Alpaca, etc.).

# 2.0 Product Design

## 2.1 Overview

The product being developed will be designed stemming from our requirements for the development process (See the Requirements Specification document for the requirements in detail). This document serves as the foundation of the development process focusing on key features and functional aspects needed for final delivery. The design of the project will consist of these main areas: automate stock buying, trading history report, and a bot algorithm that executes automatic stock purchasing and selling using trend trading and day trading strategies. The product developed will send purchase data to Alpaca and pull selling data from Alpaca. All stock purchasing and selling will be performed automatically during the hours of the open market. In order for the stock purchases and selling to be done automatically, the bot will be following a specific algorithm that will be stemming from a trend trading and day trading strategy. Since the bot will be adhering to a trend strategy it allows for the user to have trust in the fact that their investments are being guided by many different principles in market analysis. Ultimately, the product being developed aims to help users navigate through the stock market successfully.

## 2.2 User Interface

Code Craft must be able to develop a user-friendly interface. The said interface should allow users to be able to easily understand and learn from the stock trading platform by viewing automated stock purchases and sales. Considering that the buying and selling process for the stocks will be automated, the need for user interaction is limited. Thus, creating a user-friendly environment. Inspired by the design of Investopedia, our platform will encompass several pivotal areas that will be crucial to its functionality and the user's experience. These areas include the following:

**Bot Algorithm**

The automated trading feature of the platform will be powered by an advanced bot algorithm. This algorithm will be based on trend trading and day trading strategies. (For more details see section 3.1)

**Automatic Stock Purchasing/Selling**

During market hours, the platform will enable automated stock purchases and sales, removing the need for user participation. The trading process will allow for a great user experience while also serving as a guide for making investment decisions. ~~All in all,~~  That said, the overall design of our platform serves to empower its users with tools and resources needed inorder to be able to navigate through the stock market successfully.

**Trading History Report**

To show the stock purchases that were sold, Code Craft will incorporate a history report. This history report displays all stock purchases that were filled or canceled. This will serve as verification that our bot algorithm accurately sold users’ stock purchase without them having to interact with the program.

# 

# 3.0 Architectural Design

## 3.1 Algorithm

The algorithm and strategy being used automatically performs stock purchases and sells the stock based on trend analysis. The automated bot will be pulling data from Alpaca, ensuring access to real-time market information. Our algorithm will meticulously analyze the trends of stocks, with a particular focus on the nuanced fluctuations in overall stock price and percentage changes. Specifically, the algorithm will closely analyze the stocks of industry giants such as Apple, Tesla, Amazon, Google, META, and Microsoft leveraging their market influence and historical data for informed decision-making.

The system's operating parameters will coincide with the opening and closing hours of the stock market, from 9:30 am to 4:00 pm. The user must activate the bot at market opening and the bot will continue until the closing of the market where the bot will sell the users’ purchased stocks automatically. Once the user activates the bot it will automatically make purchases from Apple, Tesla, Amazon, Google, META, and Microsoft. During trading hours, the algorithm will continuously monitor the fluctuation of stock prices, dynamically adjusting its strategies to capitalize on emerging opportunities and mitigate risks. When evaluating when to sell stock, the algorithm will assess the overall percentage increase in price as a key indicator. If the price of the stock rises by 5% the stock will be sold. This tactic is put in place to ensure profit from short term price movements in the stocks. After each stock is sold, the sold price will be stored as the current price for the said stock. From here the algorithm will be able to monitor and assess the updated current price and make sales based on the new set price to ensure profit. Also, the algorithm will assess the overall percentage decrease in price. If the price of the stock decreases by 3% from then the stock will be sold. The algorithm will also sell each stock before the stock market closes automatically. This assessment is put in place to ensure that the user does not lose a large amount of money. This proactive approach ensures timely profit-taking and maximizes returns on investment.

Overall, the algorithm is meticulously designed to optimize trading outcomes for its users, leveraging advanced trend analysis techniques and real-time data to navigate the dynamic landscape of the stock market effectively. By harnessing the power of automation and strategic decision-making, the algorithm empowers users to make informed investment decisions and achieve their financial objectives.

## 3.2 API Usage

Alpaca is a modern, commission-free trading platform that enables users to buy and sell stocks, ETFs, and other assets. It's intended to be user-friendly and accessible to both new and seasoned traders. Alpaca provides an API that developers may utilize to create their trading algorithms and apps, making it popular among algorithmic traders and fintech businesses. Furthermore, Alpaca offers a variety of tools and information to assist traders to assess the market and make educated judgments. Our team has leveraged Alpaca’s API to develop a user interface for our algorithm to interact with.

When using Alpaca the user needs API keys. This enables a gateway from the bot to the API and serves as a username and password for the individual user. Once a user logs on they receive their API key. The bot uses the API keys to enter the Alpaca API ~~to see~~ in order for the user to have access and view their whole account. Once the user account is entered and they view their account, they have the ability to see how much money they have and what stocks they have purchased.

Also, as the user logs on they will be able to view our pop-up interface which has the capability of automating the buying and selling process for a chosen stock. On this screen the users first must select which stock they’d like to buy. The choice of buying a stock will show the company’s ticker symbol. After the choice of company is made using the company’s ticker symbol, the bot will then buy 1 share of that stock as a market order and trade it at the end of the day unless conditions are met. The other half of the stock selection screen shows the user’s sold stocks history. Thus, the API’s are used to send the buying data after a company is chosen and to grab the user’s data to display their selling history.

# 4.0 Restrictions, Limitations, and Constraints

## 4.1 Restrictions

This section will identify rules that were given by project requirements

* Options trading is not allowed
* Application must be automatic (no user interaction)

## 4.2 Limitations

This section will identify the scope of our software and its behavioral restraints

* The application must only be open from 9:30 am to 4:00 pm

## 4.3 Constraints

This section will identify any setbacks we presume we’ll face

* Final Delivery Date - The product is scheduled to be complete by May 1st, 2024 which limits the time Code Craft has for completion. Thus, some components may be incomplete or lack substantial quality
* Experience - A few team members are unfamiliar with stock trading which could limit the speed of our application production

# 

# 5.0 Testing

The following are the testing methods that are specified further in the testing document.

* Unit Testing
* Integration testing
* System Testing

# 6.0 Revision Log

| **Revision** | **By** | **Date** | **Description** |
| --- | --- | --- | --- |
| 1.0 | Code Craft | 1/25/2024 | Design Specifications Document for Iteration 1 |
| 2.0 | Code Craft | 2/15/2024 | Design Specifications Document for Iteration 2 |
| 3.0 | Code Craft | 3/7/2024 | Design Specifications Document for Iteration 3 |
| 4.0 | Code Craft | 3/28/2024 | Design Specifications Document for Iteration 4 |
| 5.0 | Code Craft | 4/23/2024 | Design Specifications Document for Iteration 5 |