

Table Of Contents

[**1.0 Introduction 2**](#_ibak86w53n7f)

[1.1 Goals And Objectives 2](#_vlyc8avfj5l1)

[1.2 Statement Of Scope 2](#_qbrhu6s7hfwl)

[1.3 Software Context 3](#_b8mkeok2n2bw)

[1.4 Major Constraints 3](#_rgire51sxpyp)

[**2.0 Usage Scenario 4**](#_s3dml0y46oor)

[2.1 Overview 4](#_2dj4su1dm4ff)

[2.2 Special Usage Considerations 4](#_ed7l6f1ewjft)

[**3.0 Design Model and Description 5**](#_izuie6dgiyfz)

[3.1 UI 5](#_gxakvnifmoc7)

[3.2 API Usage 5](#_ulqusj6tp2oy)

[3.3 Algorithm 6](#_3brt6xi6l6fd)

[**4.0 Functional Model and Description 7**](#_w1dwmemszcpn)

[4.1 Description of Stock Trading 7](#_1j0129yo8scd)

[4.2 Description of Algorithm 7](#_uf43vut1hn4y)

[4.3 Description of Application 7](#_xin9oyg1cy08)

[**5.0 Restrictions, Limitations, and Constraints 9**](#_sv1bi5gom1f7)

[**6.0 Validation Criteria 10**](#_10twm78fj40g)

[6.1 Classes of Test 10](#_8qwpmo3bgjtx)

[**7.0 Revision Log 11**](#_10qwa3e0vjfb)

# 1.0 Introduction

Code Craft has been given a task to create an application for computerized stock trading, as well as create our own 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, near real-time streaming stock ticker feed, an RSS news feed, or 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 (Alpaca, Investopedia, E\*TRADE, etc.).

# 2.0 Usage Scenario

The purpose of this project is to create a working computerized stock trading application. Stock trading is how investors can buy and sell stocks on the stock market. The stock market is open from 9:30 am - 4:00 pm, so the application will need to function during that time.

## 2.1 Overview

Alpaca is a current, 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. Just like Alpaca, our application will have the ability to make and cancel trades with companies automatically using a Trend Trading algorithm. Users will have the application choose which companies on the market to invest in.

## 2.2 Special Usage Considerations

Although the project will mainly use Alpaca as its main trading platform, users will have the ability to use this bot on different interfaces like E\*TRADE or other Internet Stock Trading platforms.

# 3.0 Design Model and Description

## 3.1 UI

The design model for the UI will be based on Alpaca. On this platform, individuals will be able to learn many different mechanisms for buying and selling securities without enduring any risk. When joining users will first be given $100,000 to invest freely and the user will be able to buy and sell based on Trend Trading strategies. Also through this platform, users will be able to have access to their account value, have the ability to run under trading algorithm control, have access to their performance history, have access to research capabilities on stocks, and have access to learning tools.

## 3.2 API Usage

The main interface for gaining access to and obtaining pertinent information, market trends, and financial data from Alpaca's extensive database will be the Alpaca API. The bot's functionality will be based on utilizing this API to run queries, examine information, and give customers up-to-date information about stock prices, market trends, and investing techniques. The algorithm will be based on market circumstances, spot possible investment opportunities, and provide well-informed recommendations using the data from the API. The Requirements Document should also specify the security procedures and authentication methods to provide authorized and safe access to Alpaca’s API while abiding by any particular restrictions and rules specified by the platform. The Alpaca API is mostly designed in the REST paradigm. It provides a simple and powerful way to interface with their services. In addition to the REST API, which allows for synchronous communication, their API includes an asynchronous event API built on WebSocket and SSE, or Server-Sent Events. With numerous sorts of events occurring in the financial markets, such as orders filling depending on market movement or cash settling after a period, this event-driven API provides immediate updates, improving consumer experiences.

## 3.3 Algorithm

The algorithm will be based on Trend Trading. Our trading bot has a trend trading algorithm that is a system made to navigate through financial markets. The algorithm starts with real-time market monitoring via the Alpaca API and uses several technical indicators, including moving averages, RSI, and MACD, to identify trends and possible entry and exit locations. An essential component is evaluating market volatility and modifying the bot's risk tolerance correspondingly to maintain responsible risk management. The algorithm uses time-sensitive algorithms to optimize trade execution and generates buy or sell signals based on predetermined criteria. Parameter modification is guided by extensive backtesting, which verifies the algorithm's performance using previous data. Integrating risk-reduction strategies, like take-profit and stop-loss thresholds, guarantees ethical trading. With protections against slippage and timely orders, the Alpaca API is used for real-time transaction execution. Constant observation makes it possible to adjust to shifting market circumstances, and adherence to Alpaca’s trading regulations and terms of usage is crucial. To reduce the risks involved with day trading, extensive testing in simulated environments is recommended before live deployment.

# 4.0 Functional Model and Description

## 4.1 Description of Stock Trading

Stock Trading is when someone, known as an investor, buys and sells shares in companies that are publicly traded. This buying and selling is done through the stock market. Prices of the shares change constantly, the trick is to buy your shares at a low price and sell them at a high price.

## 4.2 Description of Algorithm

The algorithm will be based on Trend Trading. Trend Trading is looking at past and current data to determine if the stock is on a downward or upward trend. The algorithm will utilize the API’s ability to connect to Alpaca. Alpaca shows a trend chart that shows the prices of the stock from different time periods (today, 1 month ago, 1 year ago, etc.). The algorithm will look at if the stock has a negative or positive number next to the stock name. For example, if the number next to the stock name is negative, then do not sell.

Today, the stock market opens up at 9:30am. Once the market opens the user will start the bot. The bot will be able to make both purchases and as well as selling the stocks of: Amazon(AMZN), Tesla(TSLA), Google(GOOG), META(META), Microsoft(MSFT) and Apple(AAPL). Throughout the day the bot will continuously make purchases and as well as selling stocks.

## 4.3 Description of Application

There will be three parts of the application, data acquisitions, strategy, and trading platform. Data Acquisitions: This gives the application the ability to access data in real-time or near real-time from various Internet resources. This can come from real-time news feeds, streaming stock ticker data, RSS feeds, or Artificial Intelligence Larger Language Models.

Strategy: This is the strategy that will be used to make buy and sell decisions.

Trading Platform: This gives the application access to various trading platforms to execute stock trades. For example, Alpaca, E\*TRADE, etc.

# 5.0 Restrictions, Limitations, and Constraints

* The application needs to work for all trading platforms, even though Alpaca is the main platform that will be used, compatibility for all platforms should be tested as well.
* Time: Most members of the team are not familiar with creating an API in Python as well as stock trading. As well as creating an algorithm to go with the application.
* Experience: Due to the unfamiliarity in Python, APIs, and stock trading, this will result with most of our time in and out of meetings researching these topics to better understand the project.

# 6.0 Validation Criteria

## 6.1 Classes of Test

The following methods are ways that Code Craft will test and will be explained in more detail in the Test Plan document:

* Unit Testing
* Integration Testing
* System Testing

# 7.0 Revision Log

| Version | Date | Document |
| --- | --- | --- |
| 1.0 | 1/25/2024 | Requirements Document Version 1.0 |
| 2.0 | 2/15/2024 | Requirements Document Version 2.0 |
| 3.0 | 3/7/2024 | Requirements Document Version 3.0 |
| 4.0 | 3/28/2024 | Requirements Document Version 4.0 |
| 5.0 | 4/23/2024 | Requirements Document Version 5.0 |