**Requirements Document**

****

**Table Of Contents**

[**1.0 Introduction 2**](#_1fob9te)

[1.1 Goals And Objectives 2](#_3znysh7)

[1.2 Statement Of Scope 2](#_2et92p0)

[1.3 Software Context 3](#_tyjcwt)

[1.4 Major Constraints 3](#_3dy6vkm)

[**2.0 Usage Scenario 4**](#_1t3h5sf)

[2.1 Overview 4](#_4d34og8)

[2.2 Special Usage Considerations 4](#_2s8eyo1)

[**3.0 Design Model and Description 5**](#_17dp8vu)

[3.1 UI 5](#_3rdcrjn)

[3.2 API Usage 5](#_26in1rg)

[3.3 Algorithm 6](https://docs.google.com/document/d/1zyBBGcVYmJ33HMKra9wcBICrrjdDFBD9/edit#heading=h.lnxbz9)

[**4.0 Functional Model and Description 7**](#_35nkun2)

[4.1 Description of Stock Trading 7](#_1ksv4uv)

[4.2 Description of Algorithm 7](https://docs.google.com/document/d/1zyBBGcVYmJ33HMKra9wcBICrrjdDFBD9/edit#heading=h.44sinio)

[4.3 Description of Application 7](#_2jxsxqh)

[**5.0 Restrictions, Limitations, and Constraints 9**](#_z337ya)

[**6.0 Validation Criteria 10**](#_3j2qqm3)

[6.1 Classes of Test 10](#_1y810tw)

[**7.0 Revision Log 11**](https://docs.google.com/document/d/1zyBBGcVYmJ33HMKra9wcBICrrjdDFBD9/edit#heading=h.4i7ojhp)

# **1.0 Introduction**

Our team, Profit Prophets, has been tasked with the completion, improvement, and amendment of the Code Craft project to create a computerized automatic stock trading bot. This document will outline all of our software design requirements and everything we plan to implement as we take on this task.

## 1.1 Goals And Objectives

Our group goal is to create a bot that advises understanding of the stock trading world. The bot should run on a trading strategy algorithm to inform the user about the current stock and whether they should trade it, buy it, or sell it. The bot will work as an extension of the Alpaca Application Programming Interface (API).

## 1.2 Statement Of Scope

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

* A Trend Trading Algorithm
* A Range Trading Algorithm
* Data acquired from Alpaca
* Alpaca (an Internet Trading Platform used to simulate a real trading platform)

The software project is being undertaken to develop a stock trading application based on an algorithm using data acquired from the Trend Trading strategy and Range Trading strategy. Our application will be executed any day from 9:30 am to 4:00 pm. Using Alpaca as our Internet Trading Platform, Profit Prophets 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. Our group, Profit Prophets, will review and approve the 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 |
| Alpaca | Paper trading software that allows us to generate API keys for our bot | Application |
| Visual Studio Code | Visual Studio Code is a source code editor that we decided to make edits to the existing code | Code Editor |

## 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 UI will follow Alpaca's platform design, offering users a risk-free environment to learn and practice trading. When users sign up, they’ll receive $100,000 in virtual funds, allowing them to trade using Trend Trading strategies.

Users will have full access to key features, including real-time account balance tracking, the ability to automate trades with trading algorithms, and a detailed performance history to monitor progress. The platform will also offer research tools to help users analyze stocks and make informed decisions. Additionally, it will include educational resources to teach users about trading strategies and market trends, making it a comprehensive learning tool for both beginners and experienced traders.

## 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.

**3.3 Algorithm**

The algorithm will be based on both Trend and Range Strategy. Trend strategy is a fast-paced day trading strategy that involves quickly buying and selling shares of highly liquid securities in order to profit from small fluctuations in price, but we will be looking at trend reports throughout the day. Range trading is a trading strategy that allows you to buy and sell stocks over a short period of time. The algorithm will utilize the API’s ability to connect to real-time stock updates.

When talking about creating an algorithm you need to understand the direction in which the stock’s value moves. The value is typically shown next to the stock name. This value is a key indicator for decision-making because the algorithm is designed to determine whether it is positive or negative. The algorithm is told not to suggest selling the stock if the value is negative. This strategy is used in financial markets where a negative number normally means there is a decline in the price. Selling under these conditions might not be the best plan and could result in major losses. Using this algorithm, the bot should be able to run smoothly and throughout the day review the portfolio to see if it hits that mark of when it needs to sell the stock or need to buy more stocks to gain a bigger profit.

# **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.

## 4.3 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. The application must include an installation script that simplifies the installation process for end users. Once installed, the application should be executable with a single click on the installed file.

# **5.0 Restrictions, Limitations, and Constraints**

* The application must be compatible with all trading platforms, even though Alpaca will be the primary platform used. Comprehensive testing for compatibility across platforms is essential. However, completing all tasks before the scheduled delivery date may lead to certain elements being incomplete or rushed due to time constraints
* **Budget:** The trading bot should accommodate various budget levels, including trading capital exceeding $100,000, and accurately display these values.
* **Time:** Most team members have limited experience in Python, API development, and stock trading, which may extend the timeline for building the application and its underlying algorithms.
* **Legal:** The bot must comply with trading regulations in each country where it operates.
* **Experience:** Due to the team's unfamiliarity with Python, APIs, and stock trading, significant time will be spent both in meetings and researching these topics to gain a better understanding of the project

# **6.0 Validation Criteria**

## 6.1 Classes of Test

The following methods are ways that Profit Prophets will test:

* Unit Testing
  + This method tests individual parts of the software to make sure each one works on its own. Since the product will have several different programs, unit testing ensures that each part functions correctly by itself.
* Integration Testing
  + After unit testing, integration testing checks how different components work together. It helps identify issues that arise when combining parts of the software.
* System Testing
  + System testing examines the entire product to make sure it meets all requirements. It checks both the functional and non-functional aspects to ensure the software is ready for delivery.

# **7.0 Revision Log**

| Revision | By | Date | Description |
| --- | --- | --- | --- |
| 1.0 | Profit Prophets | 09/09/2024 | Project Plan for Iteration 1. |
| 2.0 | Profit Prophets | 10/01/2024 | Project Plan for Iteration 2. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |