Design Specifications



Table of Contents

[**1.0 Introduction 2**](#_heading=h.30j0zll)

[1.1 Goals and Objectives 2](#_heading=h.1fob9te)

[1.2 Statement Of Scope 2](#_heading=h.3znysh7)

[1.3 Software Context 3](#_heading=h.2et92p0)

[1.4 Major Constraints 3](#_heading=h.tyjcwt)

[**2.0 Product Design 4**](#_heading=h.3dy6vkm)

[2.1 Overview 4](#_heading=h.1t3h5sf)

[2.2 User Interface 4](#_heading=h.4d34og8)

[**3.0 Architectural Design 6**](#_heading=h.17dp8vu)

[3.1 Algorithm 6](#_heading=h.3rdcrjn)

[3.2 API Usage 7](#_heading=h.26in1rg)

[3.3 Error Handling 8](#_heading=h.v67jxh6dejzq)

[**4.0 Restrictions, Limitations, and Constraints 9**](#_heading=h.lnxbz9)

[4.1 Restrictions 9](#_heading=h.35nkun2)

[4.2 Limitations 9](#_heading=h.1ksv4uv)

[4.3 Constraints 9](#_heading=h.44sinio)

[**5.0 Testing 10**](#_heading=h.z337ya)

[**6.0 Revision Log 11**](#_heading=h.3j2qqm3)

# 

# 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

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

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

* Creating an algorithm based 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, 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. The group, Profit Prophets, 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

Profit Prophets aims to develop a user-friendly interface. This 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 is currently based on trend trading and day trading strategies. Profit Prophet plans to implement a new trading strategy to complete this project, range trading. (For more details see section 3.1)

* Trend Trading: Identifies the direction of price movement
* Day Trading: Buys assets within the business day and sells to close financial positions before market close.
* Range Trading: Buys and sells stock while the market is within user-specified support and resistance limits.

**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. 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, Profit Prophets 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.

* Database
* Alpaca log

# 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 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 Error Handling

There are multiple reasons why the connection to the API may be temporarily lost including network issues or high traffic. Our bot must be prepared to handle and properly report such situations. We plan to integrate the following error handling, recovery, and user notification mechanisms to better ensure software reliability.

**Connectivity Errors and Retries**

We want to implement a retry function that, in the case of an API connection failure, will repeatedly attempt to reconnect to the API in intervals until the maximum amount of allowed retries has been reached. At that point, if the bot is unable to reconnect with the API, it will notify the user of the connectivity issue and halt stock trading operations until the issue is resolved.

**Expired or Invalid API Keys**

Our stock trading bot uses API keys and secret keys provided by Alpaca to be able to connect to the Alpaca API and essentially log the user into the system. This process, however, is currently dependent on the user correctly inserting a new API key into the bot to begin each trading session. Moving forward, Profit Prophets will implement a mechanism that will handle situations where the key entered into the bot does not match Alpaca’s current API key. This could be either because the key was inputted incorrectly, or because the API key, and therefore the trading session, has expired. We want our system to halt all stock trading operations when an API key or secret key becomes expired or invalid, and to notify the user that they must update their Alpaca API credentials to continue trading. This of course will require that the API credentials are checked regularly as the user is trading.

**Market Anomalies**

A stock market anomaly is a circumstance or event where the stock market deviates from the normal patterns and behaviors that are recognized and outlined by established financial theories. There are a multitude of reasons why these anomalies might occur, including seasonal trends in behavior, circuit breakers, or violated regulations. Our algorithm needs to account for these drastic changes in market conditions to best protect our user’s investments and capital. We want to implement a maximum loss system that will cause the bot to halt trading operations and notify the user that their cumulative loss has exceeded the allowed amount. This will prevent the bot from continuing to trade in volatile situations or market conditions and cost the user large profits. The maximum loss amount will be a specified percentage of the user’s stock portfolio. When this amount in loss is exceeded, a fail-safe will be triggered and the bot will halt operations.

**Error Reporting and Logging**

We want to create a centralized error logging location that will document the nature, dates, and times of any error that causes our bot to shut down and cease operations. The user will receive alert messages real-time when the system recognizes an error as well as our development team. When stock trading operations are halted due to an error, we would like our bot to enter a read-only state, where the user still has access to their account, trading portfolio, and market information.

# 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 December 7th, 2024 which limits the time Profit Prophets has for completion. Thus, some components may be incomplete or lack substantial quality
* Experience - The majority of our 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 used to ensure quality and reliability of our product. Please refer to our Testing Document for further detail on each methodology and results of tests conducted.

* Unit Testing
  + Involves testing separate, isolated components of the software
  + Ensures that each piece works properly independently
* Integration testing
  + Involves multiple previously tested components being combined and testing the interface between them
  + Ensures that all relevant parts of the software work properly together
* System Testing
  + Tests the completed software as a whole
  + Ensures the final product meet all stated requirements and specifications

# **6.0 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. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |