# SYSTEM REQUIREMENTS SPECIFICATION

## VERSION 2.0

# [ALCHEMETRY TRADING BOT]

Composed by: Eric Martin and Max Bar-on

[04/10/2022]

# Contents

# Introduction

In this document we will be providing details pertaining to the Alchemetry Trading Bot. This will include our companies' goals and objectives, background, stakeholders, product requirements, use-cases, and scope.

## Project Background

Trading has become avidly popular among younger generations in the past few years due to the socioeconomic effects of the Covid-19 pandemic paired with social media-based collaborations. After seeing the real-world effects of the Reddit community WallStreetBets in conjunction with young investors willing to take high risks, we saw the creation of a new generation of interest within the trading sector. Couple this with the boom in crypto trading, there is a wealth of investors ready for a new way to strategically invest their money in both the traditional and crypto sectors.

We at Alchemetry plan on bringing a new and innovative trading bot app that is attractive to the new generation of investors by creating an algorithmic trading bot that is user friendly, has levels of customization not currently seen in competitor products, can be run with or without an exchange API, and is packaged in a beautiful application.

## Goals & Objectives

Our objective at Alchemetry is to create an algorithmic trading bot that can trade both traditional and crypto assets with long term/ short term and low/high risk trading strategies. Alongside the traditional quantitative trading methodologies our product can scrape daily news articles for each asset and perform sentiment analysis utilizing advanced machine learning financial language processing models. To further customize the product, we plan on providing the user the ability to select Subreddits of interest to utilize in the sentiment analysis. Being as most exchanges require advanced paid accounts to utilize their API's we have also implemented alert features that can send messages to the user when a trade signal is triggered so that the user can assess the information and make trades manually.

Our goal is to get our product into the hands of the new generation to help provide new and innovative ways to invest while being transparent about the risks associated with algorithmic trading. Robinhood got traditional trading into the hands of millions, and we plan on getting quantitative trading knowledge and capabilities into the hands of those same people.

With these combined features, we at Alchemetry plan on providing a unique product that will in the hands of the plethora of new investors entering the market.

## Stakeholders

Our product's primary stakeholders include its developers, financers, and end users. The product owner and developers are actively involved with all of the stages of development. The financial investors have some oversight during the product development; however, they only take an active role within our demo meetings that occur at the end of each sprint. Other stakeholders include the platforms that we are partnering with, such as CoinBase and Robinhood. We will develop a protocol for profit sharing and communication with these platforms, after which, they will only have a passive role in our development process,

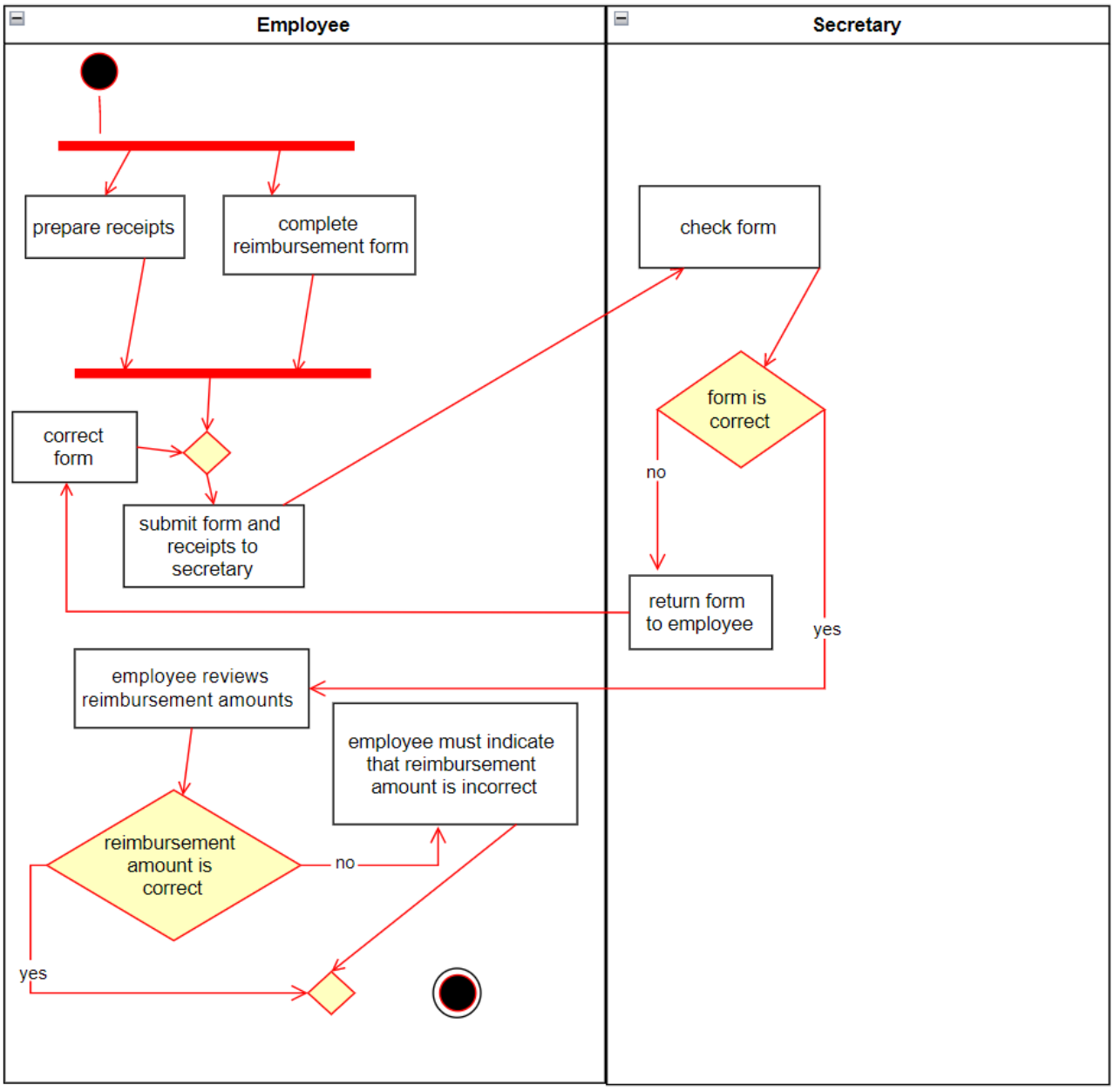
| Stakeholder | Type & Level of Interest | Level of Involvement |
|---|---|---|
| Application Users | Medium level of interest due to profits earned from successful trades on the application. | Low involvement - from consumer feedback on the application during beta testing. |
| Exchange Platforms | Low level of interest from profit sharing from API calls. | Passive involvement with final product, exchange contributes minimal work. |
| Financial Investors | High level of interest in profits obtained from stocks held in the company. | Medium involvement- want to be as passive as possible while still maximizing profits |
| Company Owners | High level of interest in the overall success of the company. | High involvement- Owners are involved with project from start to finish. |

## Current Situation and Problems/Opportunities

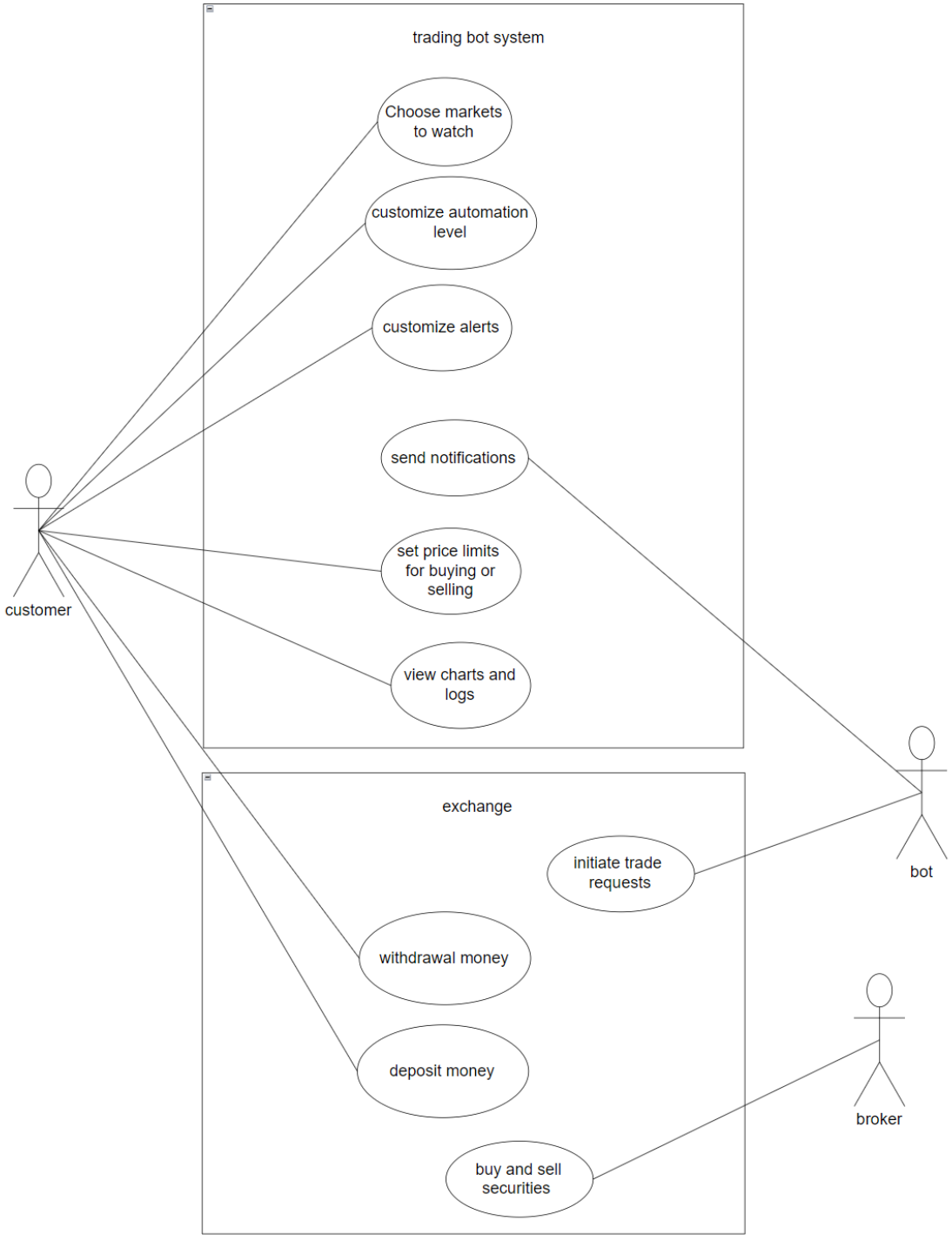
The main problem that many traders face is that they are unable to sit in front of a screen all day to make trades. Additionally, there is always a chance that a trader is not watching a certain security and ends up missing out on a trade. Markets tend to be extremely volatile on small-time scales; the large number of factors affecting the market makes it almost impossible to predict price shifts of a particular security. The main goal of our product is to make these predictions on behalf of the traders that use our system

# Statement of Scope & Requirements Overview

# Activity Diagram



# Use Case Diagram



| Actors | |
|---|---|
| **Actor Name** | **Definition / Description** |
| User | Interact with app, provide feedback |
| Bot | Executes functionality on host machines, responds and delivers API calls to clients and external platforms |
| Exchange platform | Receive API calls and executes orders |
| | |

| Market Analysis System | | | |
|---|---|---|---|
| **Use Case ID** | **Priority** | **Use Case Name** | **Description** |
| 1 | high | Select markets of interest | Users can choose which markets they want the system to analyze and keep track of. |
| 2 | low | customize automation level | Users can select which trading level the bot is set to from low to high. |
| 3 | low | customize alert notifications | Users can choose how frequently the system sends notifications about developments in the markets that it is watching. |
| 4 | medium | send notifications | Users receive notifications from the bot letting them know of the details pertaining to the market. |
| 5 | high | Set lows and highs for buying and selling prices | Users can limit when the system makes trade API requests to an exchange. They can set a maximum price for purchases and minimum price for sales. |
| 6 | medium | Review data | Users can select to review charts and logs. |

| Exchange Platform | | | |
|---|---|---|---|
| Use Case ID | Priority | Use Case Name | Description |
| 7 | high | Initiate trade requests | Receive trade request from the market analysis system. |
| 8 | high | Withdraw money | User withdraws money from account. |
| 9 | high | Deposit money | Users deposit money in an account managed by the exchange which is used to purchase securities on their behalf. |
| 10 | high | Buy and sell securities | Exchange platform executes the trade request and buys or sells securities. |

# Elaborated (Detailed) Use Cases

<table>
<tr><th>Use Case Name:</th><th>Choose Markets to Watch</th></tr>
<tr><td>Scenario:</td><td>Customer logs in and opens the “Add Markets” menu, scrolls through the menu until they find a market of interest, presses the “Watch” button under it, then fills out a popup form with options to customize the way that the system will use and interpret market information.</td></tr>
<tr><td>Triggering Event:</td><td>The Customer is initially setting up their account or wants to deepen the pool of markets that the system tracks for them.</td></tr>
<tr><td>Brief Description:</td><td>The “Add Markets” interface shows market summaries of various securities including stocks, bonds, cryptocurrencies, physical currencies, and options. The Customer can select any of the markets that are available in the menu to add to their personal watch list and can add or remove them at any time. The Customer will have access to in-depth analyses of the markets in their watch list and can be continuously updated with information about them.</td></tr>
<tr><td>Actors:</td><td>Customer</td></tr>
<tr><td>Related Use Cases:</td><td>Customize Alerts</td></tr>
<tr><td>Assumptions:</td><td>Once a Customer has been authenticated, there is an “Add Markets” button and a watch list that are easily available on the homepage of the user interface.</td></tr>
<tr><td>Stakeholders:</td><td>Customer</td></tr>
<tr><td>Preconditions:</td><td>Customer is accessing the system interface through a client application and has been logged in and authenticated.</td></tr>
<tr><td>Postconditions:</td><td>Markets have been added to or removed from the Customer’s personal watchlist.</td></tr>
<tr><td>Flow of Activities:</td><td>
<table>
<tr><th>Actor:</th><th>System:</th></tr>
<tr><td>1. Customer enters authentication information into client application</td><td>1.1. The system authenticates the Customer and the server sends their account information to the client<br>1.2. System displays the Customer’s watch list and the “Add Markets” menu on the client application</td></tr>
<tr><td>2. Customer presses the “Add Markets” button</td><td>2.1. System opens market menu</td></tr>
</table>
</td></tr>
</table>

| | | |
|---|---|---|
| | 3. Customer presses the “Watch” button under a market of interest | |
| | 4. Customer fills out customization form | 3.1. System displays form with customization options |
| | | 4.1. System records the ID of the market and all of the customization options in a client variable then sends them to the server<br>4.2. Server creates entry for the new market in the Customer’s watchlist record in the system database. |
| | 5. Customer either closes the menu or continues adding markets to their watch list (loop back to step 3) | |
| | | 5.1. System displays the changes to the Customer’s watchlist on the user interface |
| Exception Conditions: | 1.1. Customer’s authentication information is invalid<br>1.2. Customer’s client application fails<br>4.2. Customer loses internet connectivity | |

| Use Case Name: | Customize Automation Level |
|---|---|
| Scenario: | Customer logs in and presses the “Customize” button under one of the markets in their watch list. This opens a popup form that the user can fill out to modify the system automation level for that particular security. |
| Triggering Event: | Customer adds a new market to the watch list and sets the initial automation level or wants to update the automation level of a market that is already in the watchlist. |
| Brief Description: | Every market in the Customer’s watch list has an automation level that controls what the system can do without their permission. When the Customer sets the automation level on a market to “full”, the system will initiate trades on their behalf. When the Customer sets the automation level to “none”, the system won’t make any trades for that security. On the “some” automation level, the system can make trades but it will ask for the Customer’s permission before initiating one. |
| Actors: | Customer |
| Related Use Cases: | Customize Alerts, Set Price Limits for Buying or Selling |
| Assumptions: | The Customer has markets in their watchlist and is connected to the internet |
| Stakeholders: | Customer |

<table>
<tr><td>Preconditions:</td><td>Customer is accessing the system interface through a client application and has been logged in and authenticated.</td></tr>
<tr><td>Postconditions:</td><td>The automation level for one of the markets in the Customer’s watchlist has been modified.</td></tr>
<tr><td>Flow of Activities:</td><td>
<table>
<tr><td>Actor:</td><td>System:</td></tr>
<tr><td>1. Customer scrolls through the market watch list and presses the “Customize” button under a market of interest.<br><br>2. Customer sets the automation level in the pop-up form and presses the “submit” button on the form</td><td>1.1. System fetches the automation level for that market from the database<br>1.2. System opens a pop-up form and displays the current settings for the selected market<br><br>2.1. Client sends the updated automation level to the server and the server updates it in the database<br>2.2. The system modifies its behavior based on the updated automation level</td></tr>
</table>
</td></tr>
<tr><td>Exception conditions</td><td>1.1 Customer has not been authenticated<br>1.2 Client application is not connected to the internet</td></tr>
</table>

| Use Case Name: | Customize Alerts |
|---|---|
| Scenario: | Customer logs in and presses the “Customize” button under one of the markets in their watch list. This opens a popup form that the user can fill out to modify the alert settings for that particular security. |
| Triggering Event: | Customer adds a new market to the watch list and sets the initial alert settings or wants to update the alert settings of a market that is already in the watchlist. |
| Brief Description: | The system sends alerts about price behavior and analysis reports for the markets in the Customer’s watchlist. Every market has associated alert settings that can be modified by the Customer. These settings allow the Customer to choose which of the many alerts that the system generates about a market actually get sent to them. |
| Actors: | Customer |
| Related Use Cases: | Customize Automation Level, Choose Markets to Watch, Send Notifications |

<table>
<tr><td>Assumptions:</td><td>The Customer has markets in their watchlist and is connected to the internet</td></tr>
<tr><td>Stakeholders:</td><td>Customer</td></tr>
<tr><td>Preconditions:</td><td>Customer is accessing the system interface through a client application and has been logged in and authenticated.</td></tr>
<tr><td>Postconditions:</td><td>The alert settings for one of the markets in the Customer’s watchlist has been modified.</td></tr>
<tr><td>Flow of Activities:</td><td>
<table>
<tr><td>Actor:</td><td>System:</td></tr>
<tr><td>1. Customer scrolls through the market watch list and presses the “Customize” button under a market of interest.<br><br>2. Customer selects or de-selects which alerts will be sent to the customer about the market then presses the “submit” button at the bottom of the form.</td><td>1.1. System fetches the automation level for that market from the database<br>1.2. System opens a pop-up form and displays the current settings for the selected market<br><br>2.1. Client application sends the updated alert settings to the server and the server updates it in the database<br>2.2. The system adds or removes the Customer from alert lists based on the updated settings</td></tr>
</table>
</td></tr>
<tr><td>Exception conditions</td><td>1.1 Customer has not been authenticated<br>1.2 Client application is not connected to the internet</td></tr>
</table>

| **Use Case Name:** | **Send Notifications** |
|---|---|

| | |
|---|---|
| Scenario: | |
| Triggering Event: | |
| Brief Description: | |
| Actors: | Bot, Customer |
| Related Use Cases: | Customize Alerts |
| Assumptions: | The Customer has markets in their watchlist. |
| Stakeholders: | Customer, Bot |
| Preconditions: | Customer is connected to the internet. |
| Postconditions: | |
| Flow of Activities: | <table><tr><td>Actor:</td><td>System:</td></tr></table> |

| Use Case Name: | Set Price Limits for Buying or Selling |
|---|---|
| Scenario: | Customer logs in and presses the "Customize" button under one of the markets in their watch list. This opens a popup form for changing the settings for that market. If the automation level is set to "some" or "full", there will be two inputs for setting the purchase and sale price limits. . |
| Triggering Event: | Customer adds a new market to the watch list and sets the initial transaction price limits or wants to update the transaction price limits of a market that is already in the watchlist. |
| Brief Description: | When a Customer has indicated that they would like the system to make automated trading decisions for a security, the system must conform to price limitations that are set by the Customer. Every market in the watchlist that has automated trading permissions has an associated purchase price maximum and sale price minimum. The bot won't make a sale for a security when the market price is below the Customer-set price minimum and it won't purchase when the price is above the maximum. |
| Actors: | Customer |
| Related Use Cases: | Customize Automation Level |

<table>
<tr><td>Assumptions:</td><td>The Customer has enabled automated trading for some markets in their watchlist.</td></tr>
<tr><td>Stakeholders:</td><td>Customer</td></tr>
<tr><td>Preconditions:</td><td>Customer is accessing the system interface through a client application and has been logged in and authenticated.</td></tr>
<tr><td>Postconditions:</td><td>The maximum purchase price or minimum sale price for one of the markets in the Customer’s watchlist has been modified.</td></tr>
<tr><td>Flow of Activities:</td><td>
<table>
<tr><td>Actor:</td><td>System:</td></tr>
<tr><td>1. Customer scrolls through the market watch list and presses the “Customize” button under a market of interest.<br><br>2. Customer enters a number in one of the text inputs for sale or purchase price limits then presses the “submit” button at the bottom of the form.</td><td>1.1. System fetches the sale and purchase price limits for that market from the database<br>1.2. System opens a pop-up form and displays the current settings for the selected market<br><br>2.1. Client application sends the updated alert settings to the server and the server updates it in the database<br>2.2. The system modifies the constraints for the trade action triggers for the security based on the Customer’s inputs.</td></tr>
</table>
</td></tr>
<tr><td>Exception conditions</td><td>1.1 Customer has not been authenticated; the selected market doesn’t have automated trading enabled<br>1.2 Client application is not connected to the internet</td></tr>
</table>

| **Use Case Name:** | **View Charts and Logs** |
|---|---|

<table>
<tr><td>Scenario:</td><td>Customer logs in and presses the “View More” button under one of the markets in their watch list. This opens a dropdown full of graphical representations of the market price history, future predictions, and information summaries.</td></tr>
<tr><td>Triggering Event:</td><td>Customer looks at the expanded information view of a market when selecting it from the “Add Markets” menu or is looking closer at one of the markets in their watchlist..</td></tr>
<tr><td>Brief Description:</td><td>The system creates, updates, and stores extensive amounts of information about all of the markets that it makes available for customers to watch. This information can be requested by Customers through the user interface and displayed on their client information. This allows them to gain a more in-depth perspective about the markets they are watching.</td></tr>
<tr><td>Actors:</td><td>Customer</td></tr>
<tr><td>Related Use Cases:</td><td>Send Notifications, Choose Markets to Watch</td></tr>
<tr><td>Assumptions:</td><td>The Customer has been authenticated.</td></tr>
<tr><td>Stakeholders:</td><td>Customer</td></tr>
<tr><td>Preconditions:</td><td>Customer is accessing the system interface through a client application and has been logged in and authenticated.</td></tr>
<tr><td>Postconditions:</td><td>Historical information about one of the markets is displayed on the Customer’s client application interface.</td></tr>
<tr><td>Flow of Activities:</td><td>
<table>
<tr><td>Actor:</td><td>System:</td></tr>
<tr><td>1. Customer scrolls through the market watch list and presses the “View More” button under a market of interest.<br><br>2. The Customer scrolls through the dropdown menu and looks at the charts and logs that have been pulled up</td><td>1.1. Client application sends query to the database for the graphics records matching the name of the security<br>1.2. After fetching the graphics, the system displays them in a dropdown menu for the Customer</td></tr>
</table>
</td></tr>
</table>

| | |
|---|---|
| exceptions: | 1.1 the Customer is disconnected from the internet |

| Use Case Name: | Initiate Trade Requests |
|---|---|
| Scenario: | |
| Triggering Event: | |
| Brief Description: | |
| Actors: | Bot |
| Related Use Cases: | Customize Automation Level |
| Assumptions: | The Customer has markets in their watchlist with automation level set to “full” or “some”. |
| Stakeholders: | Bot, Exchange Firm |
| Preconditions: | The customer has an account with the exchange and the system has access to their authentication information. |
| Postconditions: | The customer’s asset holdings and exchange account balance have been changed. |
| Flow of Activities: | Actor: \| System: |

| Use Case Name: | Withdrawal Money |
|---|---|
| Scenario: | The Customer wants to use the money in their exchange account to buy something that isn’t available on the exchange. They log in to the exchange system and make a transfer request to move money from the exchange account to their personal checking account. |
| Triggering Event: | The Customer wants to use the money in their exchange account to buy something that isn’t available on the exchange. |
| Brief Description: | The Customer can withdraw money from their exchange account by transferring to another bank account that offers checking services. |
| Actors: | Customer |
| Related Use Cases: | Deposit Money |

<table>
<tr><td>Assumptions:</td><td>Customer has an account with the exchange and a personal bank account.</td></tr>
<tr><td>Stakeholders:</td><td>Customer, Exchange Firm</td></tr>
<tr><td>Preconditions:</td><td>The Customer has been logged in and authenticated by the exchange.</td></tr>
<tr><td>Postconditions:</td><td>The customer’s exchange account balance has decreased.</td></tr>
<tr><td>Flow of Activities:</td><td>
<table>
<tr><td>Actor:</td><td>System:</td></tr>
<tr><td>1. Customer uses exchange interface to make a “money transfer request”<br><br>2. Customer enters their exchange account routing number, their personal bank routing number, and the amount of money to transfer, and presses “submit”</td><td>1.1 exchange system queries Customer for the source and destination account routing numbers and the transfer amount<br><br>2.1 the bank and exchange accounts update their balance information for the Customer.</td></tr>
</table>
</td></tr>
<tr><td>Exceptions:</td><td>1.1. The Customer doesn’t have an account with the exchange<br>2.1. The Customer doesn’t have sufficient funds or the routing number for their bank account can’t be found.</td></tr>
</table>

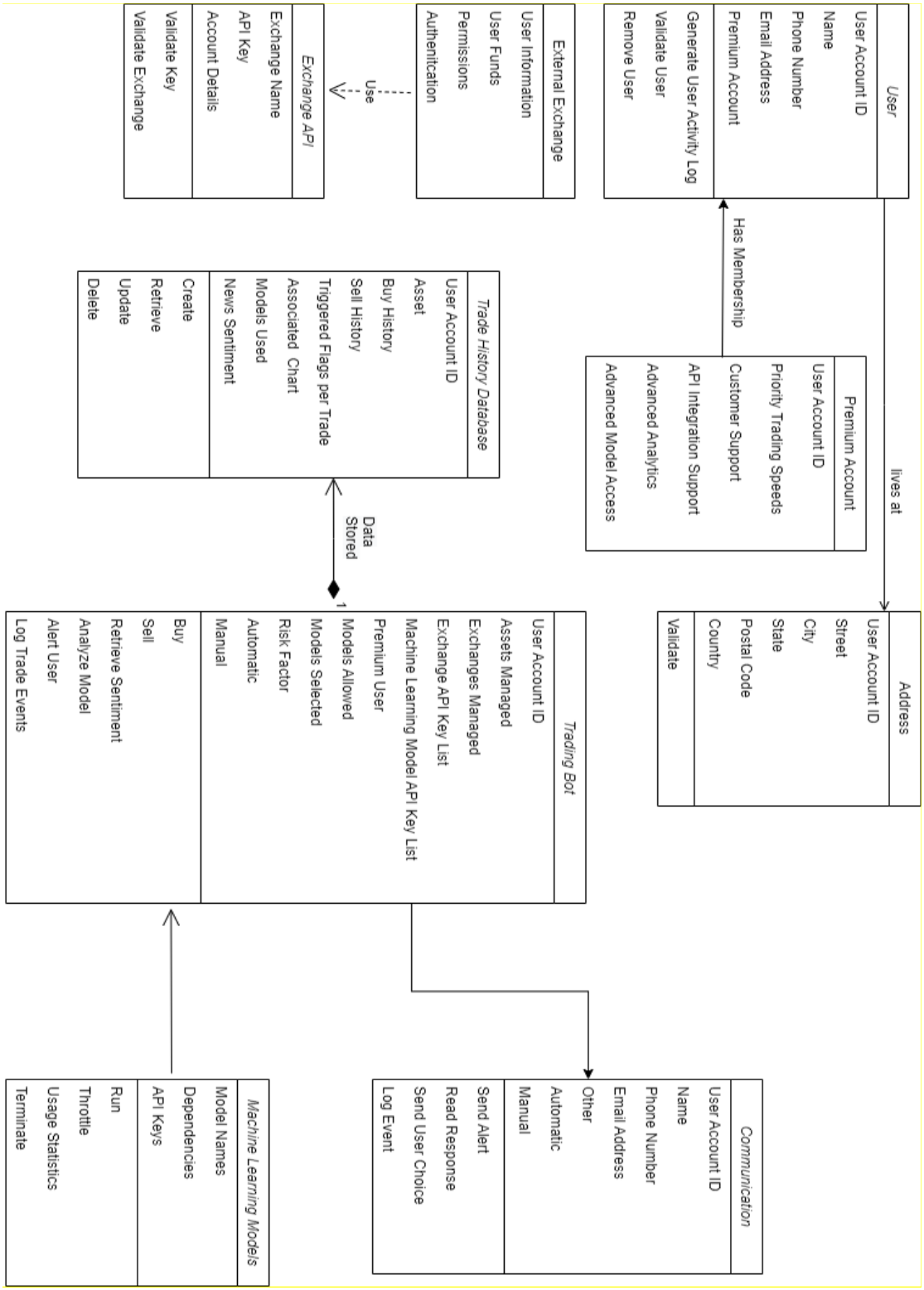
| **Use Case Name:** | **Deposit Money** |
|---|---|
| Scenario: | The Customer wants to invest money in financial securities. They log in to their bank and make a transfer request to move money from the personal bank account to their exchange account. |
| Triggering Event: | The Customer wants to invest money in the securities available on the exchange. |
| Brief Description: | The Customer can use a bank account to move money into their exchange account. |
| Actors: | Customer |

<table>
<tr><td>Related Use Cases:</td><td>Withdrawal Money</td></tr>
<tr><td>Assumptions:</td><td>Customer has an account with the exchange and a personal bank account.</td></tr>
<tr><td>Stakeholders:</td><td>Customer, Exchange Firm</td></tr>
<tr><td>Preconditions:</td><td>The Customer has been logged in and authenticated by the bank..</td></tr>
<tr><td>Postconditions:</td><td>The customer’s exchange account balance has increased.</td></tr>
<tr><td>Flow of Activities:</td><td>
<table>
<tr><td>Actor:</td><td>System:</td></tr>
<tr><td>1. Customer uses bank interface to make a “money transfer request”<br><br>2. Customer enters their exchange account routing number, their personal bank routing number, and the amount of money to transfer, and presses “submit”</td><td>1.1 bank system queries Customer for the source and destination account routing numbers and the transfer amount<br><br>2.1 the bank and exchange accounts update their balance information for the Customer.</td></tr>
</table>
</td></tr>
<tr><td>Exceptions:</td><td>1.1. The Customer doesn’t have an account with the bank<br>2.1. The Customer doesn’t have sufficient funds or the routing number for their exchange account can’t be found.</td></tr>
</table>

| **Use Case Name:** | **Buy and Sell Securities** |
|---|---|
| Scenario: | |
| Triggering Event: | |
| Brief Description: | |
| Actors: | Exchange Platform Broker |
| Related Use Cases: | Initiate Trade Requests |

<table>
<tr><td>Assumptions:</td><td>Customer has an account with the exchange and either owns securities under the account or has money in the account.</td></tr>
<tr><td>Stakeholders:</td><td>Customer, Exchange Firm, Exchange Platform Broker</td></tr>
<tr><td>Preconditions:</td><td>The customer or the bot has made a request to the broker to make a transaction.</td></tr>
<tr><td>Postconditions:</td><td>The customer’s asset holdings and exchange account balance have been changed.</td></tr>
<tr><td>Flow of Activities:</td><td><table><tr><td>Actor:</td><td>System:</td></tr></table></td></tr>
</table>

# Class Diagram



| Class | Class Description |
|---|---|
| Trading Bot | Automates buying and selling of assets, acts as go-between for user and exchange |
| Exchange API | Allows bot to communicate with external exchanges |
| Communication | Keeps user updated about all transactions related to account and allows user to verify trades |
| Machine Learning Models | Manages machine learning models, dependencies, API keys, and usage |
| Trade History Database | Database that keeps track of information about user's trading history |
| User | Keeps user information and data |
| Premium Account | Contains premium user information and options |
| Security Settings | User security and safety settings |
| Address | User address information |
| External Exchange | Contains User information needed to utilize API, access funds, and view funds |

| Class | Attribute | Attribute Description |
|---|---|---|
| **Trading Bot** | Assets Managed | The various stock and crypto tickers that the bot will manage |
| | Exchanges Managed | List of Exchanges bot will interact with |
| | Exchange API Key | Key Value pair of API Keys for associated exchanges |
| | Machine Learning Model API Key | Key Value pair of ML model and associated API keys |
| | Premium User | Boolean storing user membership |
| | ML Models Allowed | List of ML models bot can access |
| | ML Models Selected | List of current models being utilized |
| | Risk Factor | User set risk value for bot to follow |
| | Automatic | Boolean signifying automatic bot trading functionality |
| | Manual | Boolean signifying bot will not execute trades automatically |
| **Exchange API** | Exchange Name | Holds name of exchange |
| | API Key | Holds API key information |
| | Account Details | Holds Account authentication details |
| **Communication** | Name | User's name |
| | Phone Number | List of user phone numbers |
| | Email Address | List of Email addresses |
| | Other | Application communication vectors |
| | Automatic | Automatically trade but still communicate |
| | Manual | Require user permission to execute trade |

| | | |
|---|---|---|
| **ML Models** | Model Names | List of model names |
| | Dependencies | List of dependencies that are needed for each Model |
| | API Keys | List of API keys for each model |
| **External Exchange** | User Information | User data required to utilize exchange |
| | User Funds | Amount of funds held in the user's account |
| | Permissions | User and exchange defined permission |
| | Authentication | Authentication token |
| **Trade History Database** | User Account ID | Primary Key |
| | Asset | Asset column |
| | Buy History | Buy order history |
| | Sell History | Sell order history |
| | Triggered Flags | Flags that triggered the trade to execute |
| | Associate chart | Chart showing trade |
| | Models Used | Models used in decision |
| | News Sentiment | News sentiment on asset during trade |
| **Premium Account** | User account ID | Primary key |
| | Priority Trading Speeds | Set trading bot speeds based on payment |
| | Customer Support | Access to customer support Boolean |
| | API Integration Support | Access to API integration assistance Boolean |
| | Advanced Analytics | Access to advanced analytics Boolean |
| | Advanced Model Access | Access to advanced trading models Boolean |
| **Address** | User Account ID | Primary Key |
| | Street | User permanent address |
| | City | User city |
| | State | User state |
| | Postal Code | User zip code |
| | Country | User country |
| **User** | User Account ID | Primary User Identification Key |
| | Name | User's Name |
| | Phone Number | User Phone Number |
| | Email Address | User Email Address |
| | Premium Account | Boolean – Is user a premium member? |

# Supplementary Specification

| Usability Requirements & Constraints | | |
|---|---|---|
| Requirement / Constraint Name | Priority | Requirement/Constraint Descriptions |
| Able to be used on both IOS and Android Devices | High | This application needs to be able to be used across all of the most popular mobile device operating systems, most notably Apple's IOS and Google's Android. Additionally, the application must be able to be updated to be used on the latest versions of these operating systems. |
| Accessible on both Wi-Fi and Mobile Connection | High | Because this is designed to only be used on mobile devices, this application should work regardless of if the user is connected to the internet or using data provided by their cell provider. |
| Simple UI | Medium | The user should have no doubts about the trades that they are making, and that starts with the user interface. The UI should be simple, but still display all of the information required to make informed trades. |

| Reliability Requirements & Constraints | | |
|---|---|---|
| Requirement / Constraint Name | Priority | Requirement/Constraint Descriptions |
| Available during all hours of trading | High | The application should be available Monday through Friday, 9:30-4:00 EST. Additionally, the bot should be able to account for government holidays and any halts within the market. The bot should also be available after the hours stated above for crypto market applications. |
| Be able to execute trades rapidly | High | Trades should be executed without delays in order to make precision market moves. If a delay were to occur, the order would be canceled |

| Performance Requirements & Constraints | | |
|---|---|---|
| Requirement / Constraint Name | Priority | Requirement/Constraint Descriptions |
| Machine Learning GPU Server | Medium | In order to run machine learning model algorithms at fast speeds the company will need to have either physical access or cloud access to GPU servers to handle the computational needs. |
| Cloud Hosting Provider | High | The application needs to be hosted on a secure and trustworthy cloud provider like AWS to ensure that uptime is as high as possible, and storage is not an issue. |

| Security Requirements & Constraints | | |
|---|---|---|
| Requirement / Constraint Name | Priority | Requirement/Constraint Descriptions |
| User Data Security | High | All user data shall be encrypted. User exchange login information will not be stored on the servers. All withdrawals will require MFA. |

| | | |
|---|---|---|
| Satisfy Secure Socket Layer (SSL) security certification | High | System needs to meet SSL requirements/standards to secure data transaction between the client side and the server side. This protects from perpetrators on the network trying to obtain valuable information from either side. |
| OWASP | High | Company shall ensure the OWASP top 10 vulnerabilities are covered as best as possible as trading exchanges are targets for attack. |

| **Design Requirements & Constraints** | | |
|---|---|---|
| Requirement / Constraint Name | Priority | Requirement/Constraint Descriptions |
| Interfacing with SQL Server | Medium | All trade history will be stored on the backend database servers. This data must be always accessible by the client. |
| User Agreements | High | All users must be aware that there are risks involved in trading stocks and cryptocurrencies and that utilizing a trading bot does not guarantee success. |

| **Implementation Requirements & Constraints** | | |
|---|---|---|
| Requirement / Constraint Name | Priority | Requirement/Constraint Descriptions |
| Dependency Management | Medium | The machine learning models require specific software packages and updates may be required for security and performance issues. Keeping up to date with dependency updates will be important to maintain model performance |
| Front End Application Method Execution | High | In order to secure our application, all logic performed client side shall also be performed server side in order to prevent man in the middle attacks and secure our system. |

| **Interface Requirements & Constraints** | | |
|---|---|---|
| Requirement / Constraint Name | Priority | Requirement/Constraint Descriptions |
| Interoperability | Medium | Our application will be trading through various exchange API vectors. This means we will need to keep up with API changes as well as work towards implementing other exchange API's to increase our market share. |
| Financial Audit Reports | High | All data shall be logged in the correct ISO and financial formats in order to allow for smooth and fast auditing. |

| **Physical Requirements & Constraints** | | |
|---|---|---|
| Requirement / Constraint Name | Priority | Requirement/Constraint Descriptions |
| Data Storage Backups | High | All data shall be routinely backed up in multiple locations with one location being disconnected from the network. Data shall be ensured safe for up to 7 years per regulations. |
| Employee Computers | High | All employees shall have access to computers, monitors, keyboards, and mice in order to perform their jobs effectively. |

| | | |
|---|---|---|
| Off-Site Servers | High | Servers must be always available with the option to upscale when needed during DDOS attacks and high influx trading times. |
| Network Infrastructure | High | Consistent uptime is critical for a trading platform. When the internet is down, clients can lose valuable money. |

| **Supportability Requirements & Constraints** | | |
|---|---|---|
| Requirement / Constraint Name | Priority | Requirement/Constraint Descriptions |
| System Maintainability | High | This system relies on many external API's and machine learning models. In order for this application to have minimal downtime we must be proactive to ensure all updates do not interfere with our current system by implementing parallel testing prior to merging into our working application. |
| Security Updates | High | In order to ensure our product is safe we shall implement all critical updates required to meet our security standards. With this said, we must ensure the updates do not interfere with our current systems. |

# Conclusion

Our objective was to create an algorithmic trading bot that can trade both traditional and crypto assets with long term/ short term and low/high risk trading strategies. We have provided a detailed systems analysis and design plant that outlines all the pertinent steps needed in order for the client to reach this goal. The stakeholders asked for the both the backend design of their product and elaborate use cases for a front-end design. We have designed a robust application that allows for a trading bot to perform traditional quantitative trading methodologies, utilize advanced machine learning algorithms, and scrape daily news articles to perform sentiment analysis utilizing advanced machine learning financial language processing models. To further customize the product, we have provided the user the ability to select a risk profile, which machine learning models they are wanting to utilize, which trading algorithms they would like to use, and whether they want to have the bot trade fully automatically or require user verification prior to a trade.

Our goal is to allow your company to get this product into the hands of the new generation to help provide new and innovative ways to invest while being transparent about the risks associated with algorithmic trading. Robinhood got traditional trading into the hands of millions, and we have developed this plan to help Alchemetry Trading Bot get quantitative trading knowledge and capabilities into the hands of those same people. After the first test of the system is implemented and key stakeholders have given their feedback, we recommend reviewing the project objectives and requirements to deliver a solution aligned to all parties involved.

# Contact Information

**Eric Martin**
(303)358-5851
Ebmartin@colostate.edu