AlgoStock's Stock Analysis Services

Final Assignment

12/7/2020

Table of Contents

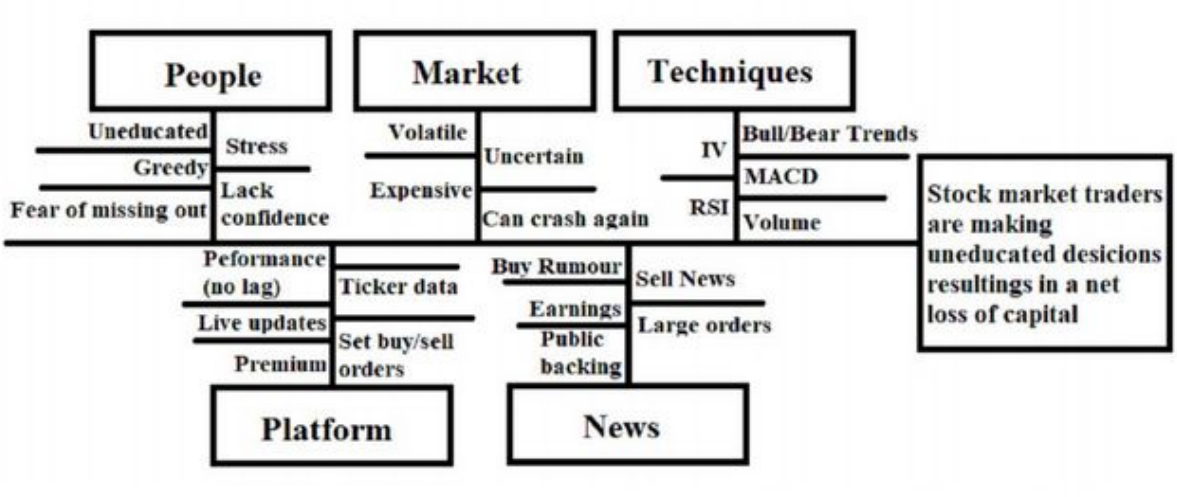
Executive Summary

AlgoStock will take advantage of the stock market to create wealth and get new traders to become long term investors that will try to make a profitable return on their investment. The algorithm of the application will take into account the volatility of a stock, the high and lows, and current market conditions to output a performance summary and decide if the stock will hold for the long haul. AlgoStock has the unique advantage of being cheap and effective, this creates enough of an opportunity for our members to pay for our service and still make profits on the stock market. In addition, our GUI is beginner friendly and easy to use for new traders. We are also looking to expand our project with continued support from the trading community, by adding in new, earning reports, professional analyst opinions, and more. It is common that new traders are lost without using third party platforms, our application will give them an edge over the ever expanding stock market. We have ambitions to continuously update the service in order to generate more interest in our product and develop it to our full expectations. Not only do we want to follow our own business model, but we want to make it our mission to follow the needs of our customers. We cherish their ideas on how we can make our product better and we promise to keep it updated based upon the feedback we receive. This is reflected in our cheap price point for first year users, as we believe that they should be given a chance to adapt to our way of trading and provide us with the direction they believe we should move the company in. AlgoStock appreciates all of the support it can get and we will continue to provide customers with the best trading experience we can offer.

Overview of the System

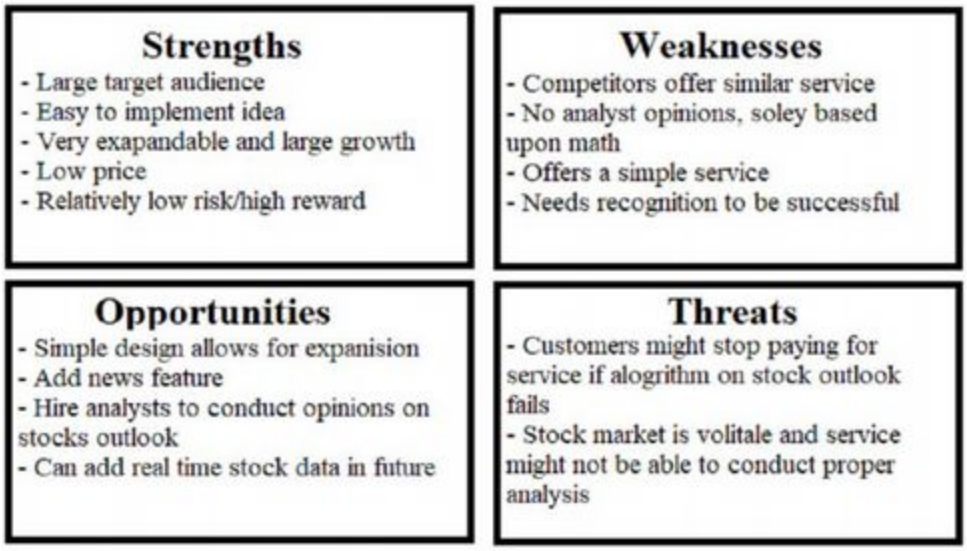
Our platform will allow you to take advantage of the stock market to create additional wealth. The algorithm used will take into account the volatility of a stock, the highs and lows, and current market conditions to output a performance summary and decide if it's one to hold for the long haul. This will be a service greatly utilized by long term investors who are trying to make a profitable return on their investment. The price package we present is reflective of the money that can be made utilizing our service and that is not to mention we are going to be using only the best servers. Not only is our service going to be helpful for finding out the outlook of stocks, but our GUI is beginner friendly and easy to use for any new trader. We are looking to expand our project with continued support from the trading community, by adding in news, earning reports, professional analyst opinions, and more. To accommodate young investors, we have decided to reduce our official price of $600 yearly to $500 for the first year of our service being run. Many traders are lost without using third party platforms to give them an edge over the ever expanding market, so let us do that hard work for you and ensure you that you will never have to worry about technical analysis again. Only 1% of traders are profitable, let us make a change to the statistics and create a new generation of smart traders.

Problem/Opportunity Statement:



The fishbone diagram above represents the five main factors of unsuccessful traders. Newer traders will often run into one or more of these factors on a daily basis that will create a poor experience. The problem lies deeply in the individual's lack of knowledge. Our project is setting out to provide an outlet that can offer support for the main goal, which is becoming a profitable stock market trade. People are less likely to invest in a market that they are constantly losing money on, so providing this service will increase the overall prosperity of traders and the market on a large scale.

SWOT Diagram for AlgoStock



System Components

User Stories

1. A stock analyst wants to have access to adding his opinion onto specific tickers to give his own perspective. WHO: stock analyst WHAT: wants to add opinions on certain stocks WHY: give unique perspectives on stock to generate additional value and insight

    a. Functional: Our service should provide additional information about stocks that can give a different view other than our algorithms to determine profitability.

    b. Non-Functional: Service should present reliable information.

2. The sales specialist wants access to regional sales data to find a target audience to focus on. WHO: sales specialist WHAT: understand regional sale data WHY: identify primary

target audience

a. Functional: AlgoStock should be able to identify a primary target audience to understand our demographics and increase sales within that group.

b. Non-Functional: Increase usability via target audience to appease their demographic needs, example: younger traders are more likely to want short term high risk profits vs older traders wanting to have limited risk.

3. Owner wants access to sales information to determine the success and profitability of AlgoStock. WHO: Owner WHAT: wants access to sales information WHY: determine success of AlgoStock, results will help understand future plans

a. Functional: Determine success of AlgoStock based on profits, this will keep the plan on track, or understand if changes are needed.

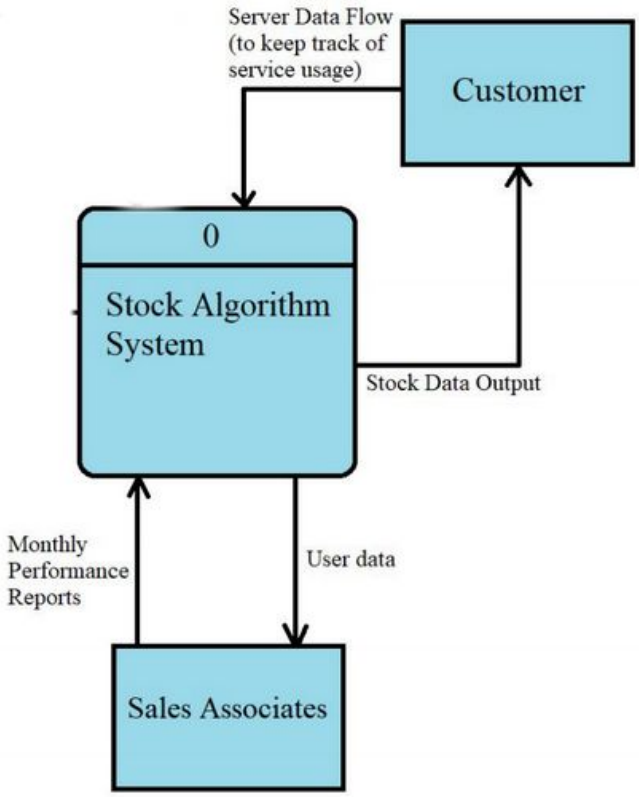
b. Non-Functional: Increase reliability of AlgoStock, as if the company is successful there is no need to change the plan.

4. A manager wants access to a planning schedule in order to find out if it's on track. WHO: manager WHAT: wants access to planning schedule WHY: stay on task's track and adapt to the situation if needed.

a. Functional: AlgoStock has a strict schedule to follow, if the criteria is not met, changes might need to be considered by management.

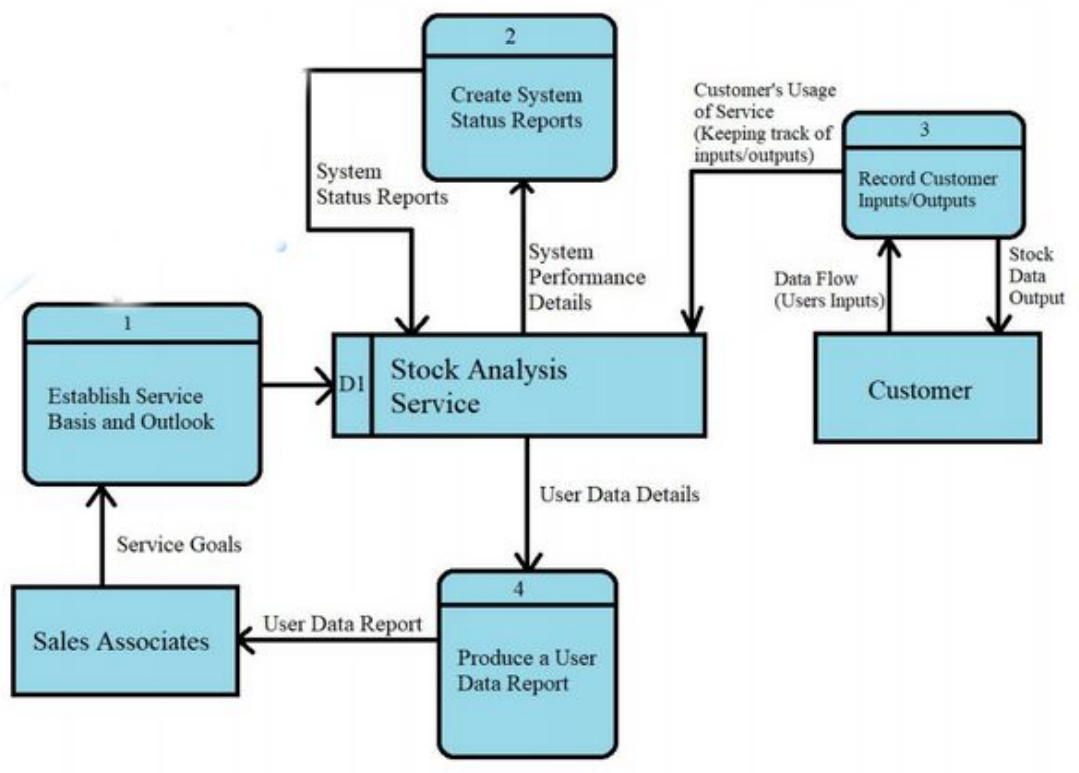
b. Non-Functional: Following schedule will increase reliability of AlgoStock.

Context Diagram



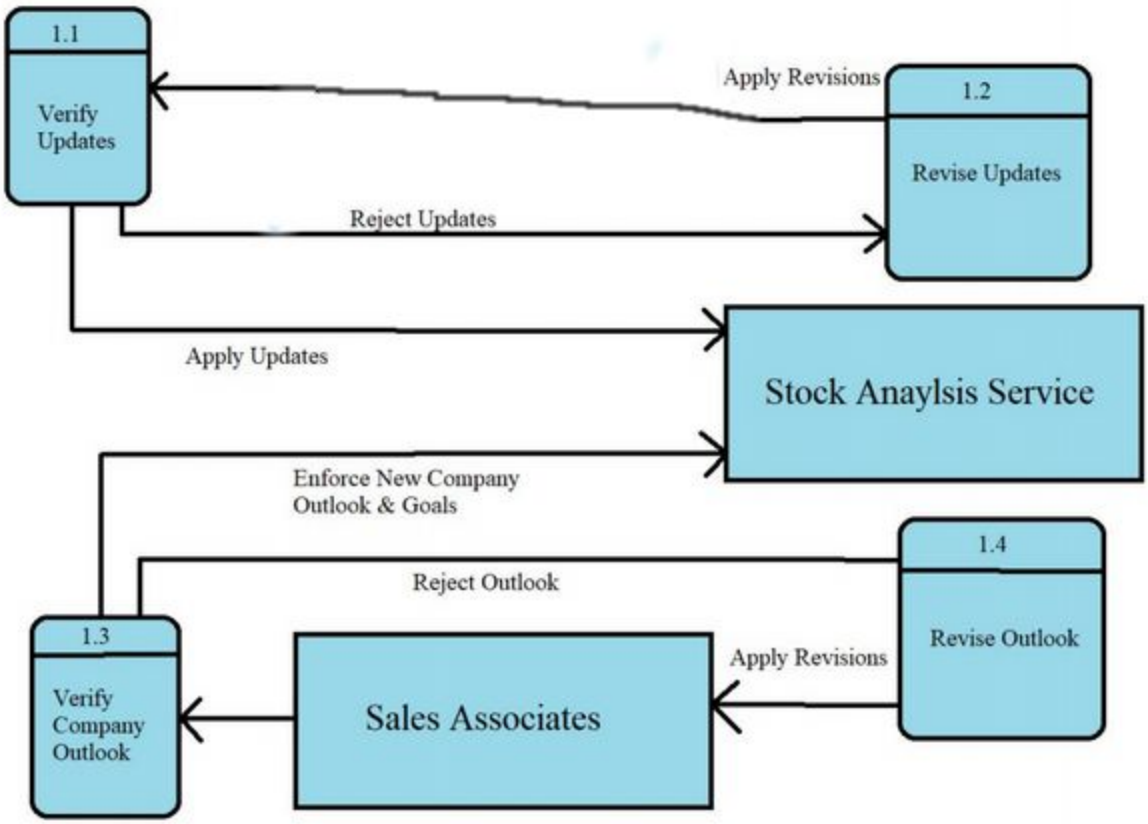
This is the context diagram of the stock algorithm system. The customer is going to be recorded with any inputs he makes and our system will display back outputs. The sales associate will be collecting user data in order to make reports on the monthly performance of the system

DFD0

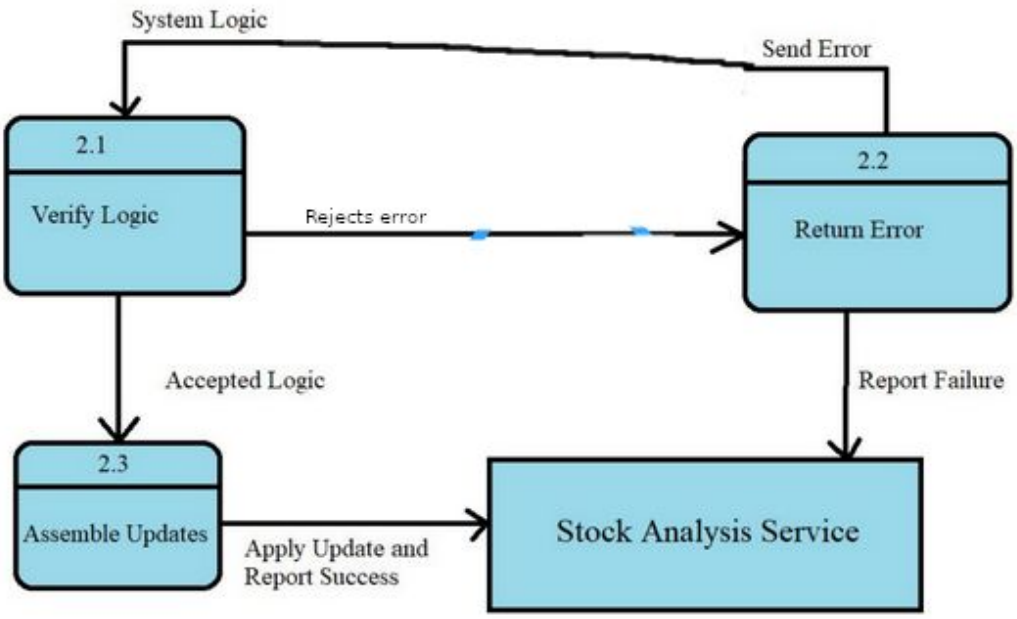


The DFD0 diagram will present the system in a more detailed perspective. It's seen that the system will start off with establishing outlooks and a foundation based on the outlooks of the sales associates and coders. The customer is going to be recorded in terms of his inputs and outputs based on what he or she is doing with our service. This could be tracking the most inputted stock to gain a better understanding of what is popular among our customers to target their wants better. This information will then be used by the sales associate to produce a user data report in which it will summarize the finding and create a way to make capital gains on our knowledge of the users we have and seek to get.

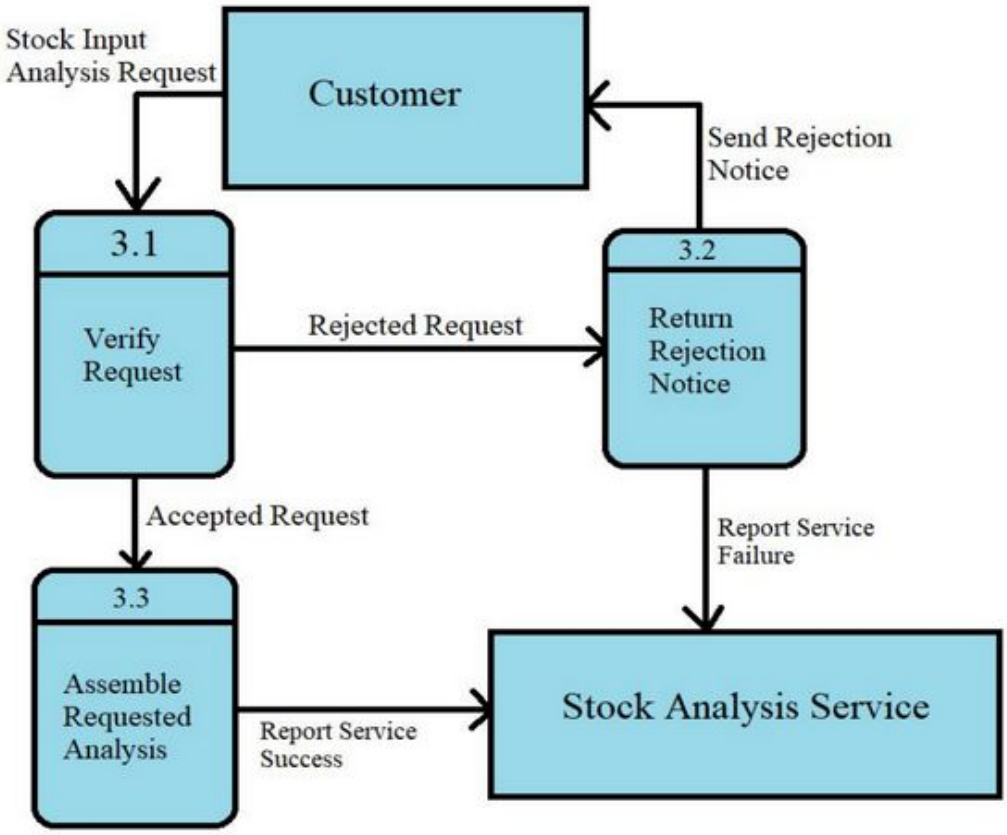
Establish Service Base & Outlook (Process 1)



Create System Status Report (Process 2)



Record Customer Inputs/Outputs Leveling (Process 3)



Record Customer Inputs/Outputs Leveling (Process 3)

Input data flows:

Stock, Verify Request, Stock Analysis Service

Output data flows:

Service Success, Rejection Notice

For each Stock:

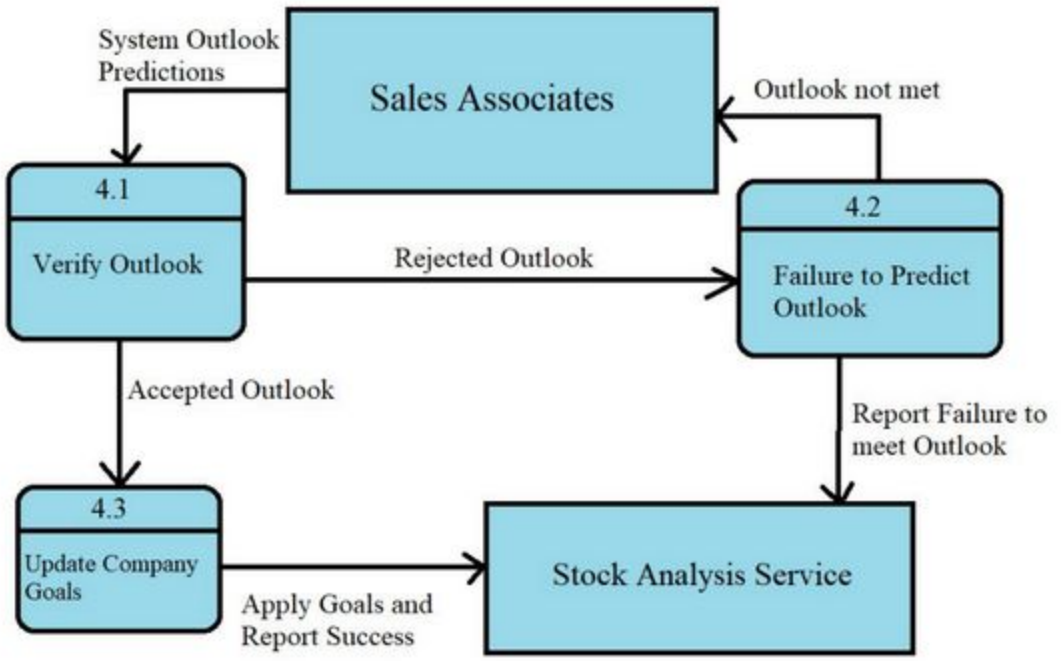
If Customer Analysis Request = y and Stock Analysis Service = OK

Output Service Success

Else

Output Rejection Notice

Produce a User Data Report (Process 4)



Data Dictionary Report:

External Entities:

Customer and Sales Associates.

Functions/Processes & Data Flows:

Establish service outlook and basis:

Sales associate will contribute to this process that allows the organization to

firmly grasp where it stands at a financial and technical level moving forward.

Create system service reports:

This is a process that involves the service displaying performance details that will be

passed to the developers so they can assess the situation and update the current system.

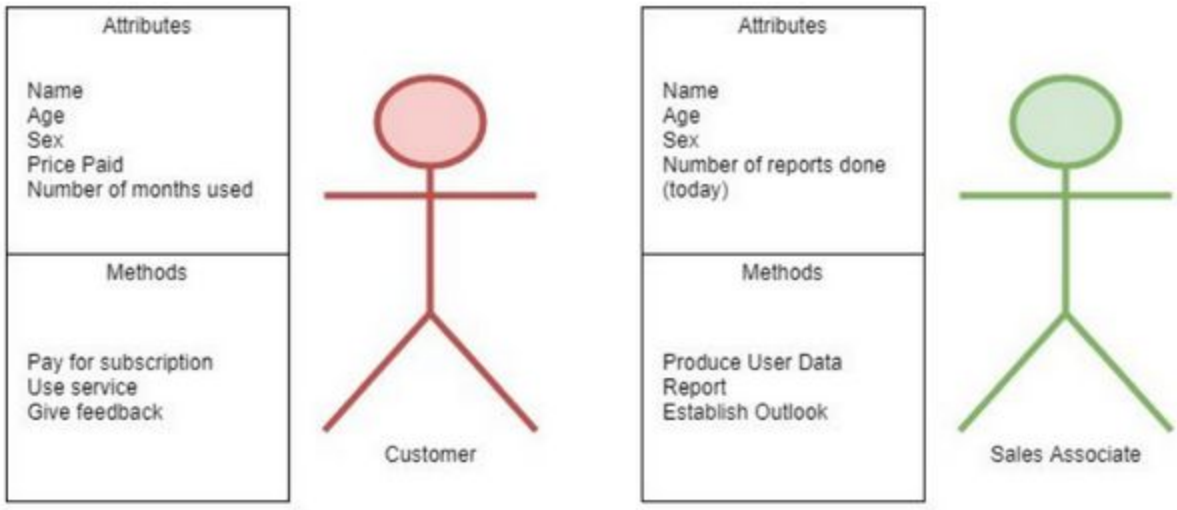
Record customer inputs/outputs:

This is a customer based process that will take all of the inputs and outputs that a customer has with our service to monitor the usage of our stock analysis product. This could be useful to track the popular stocks which we can further add professional analysis to.

Produce a user report:

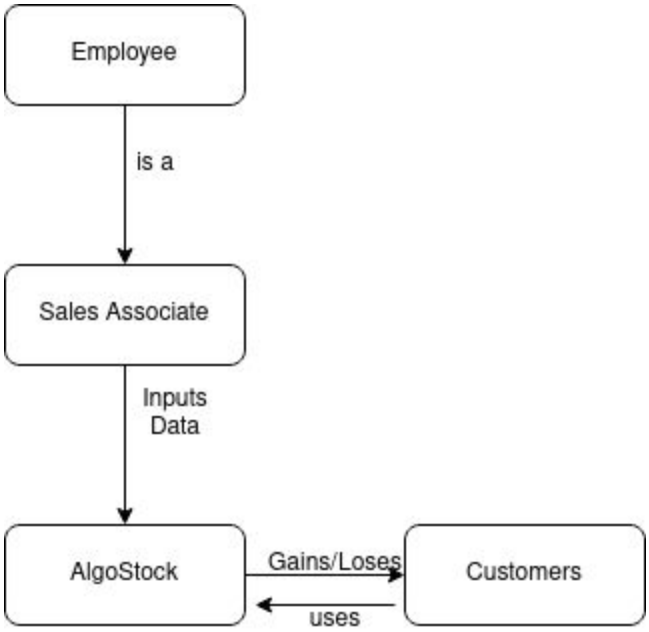
This will be the last process in our DFD0, which will consist of taking the user data from the service and putting it into a report to be passed onto sales associates who can use the information to make a proper analysis on what is going well and what is not.

Objects using Attributes and Methods



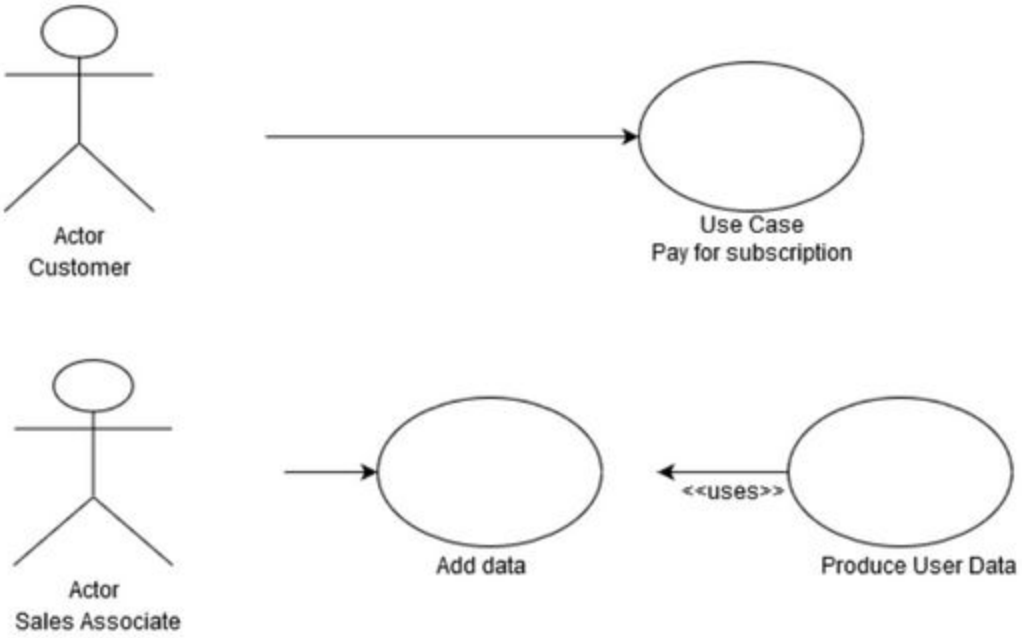
These objects are created based upon the Algo-Stock information system we have created. The methods are details derived from the IS and we have included specifics. The attributes are basic traits that we feel can help identify what is important.

Object Relationship Diagram



The object relationship diagram provides a visual representation of the interaction between our objects and their processes. The objects all derive from Employee and continue to follow down their respective chains. These chains present the process and show the ways in which they will work together

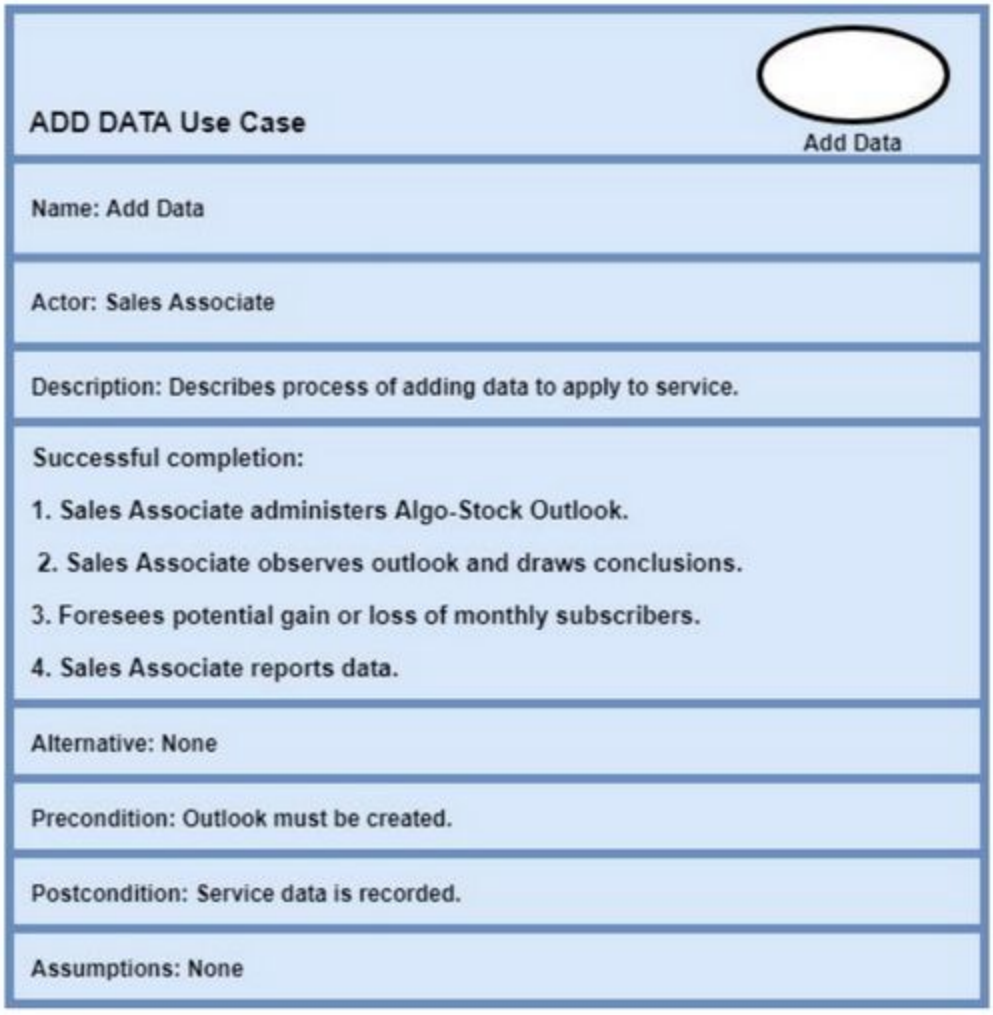
Define Actors (external entities) and Use Case



A user/customer can pay for premium content(subscription).

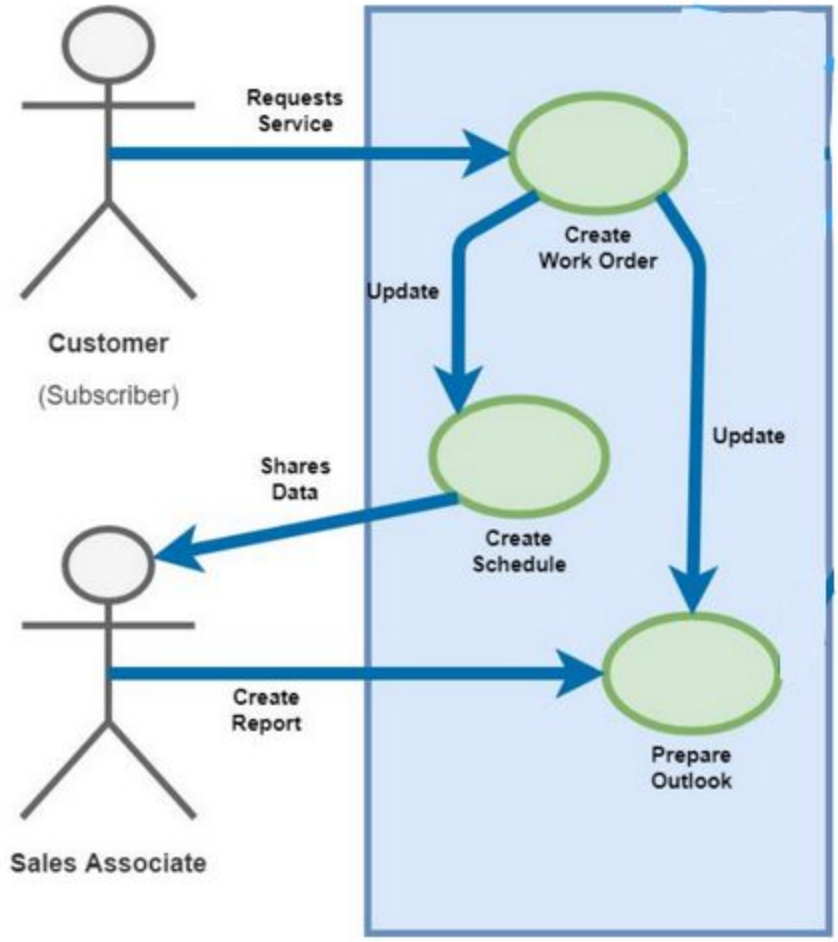
A sales associate can add a user and it can update data of the user if necessary.

Model at least one Use Case and Description



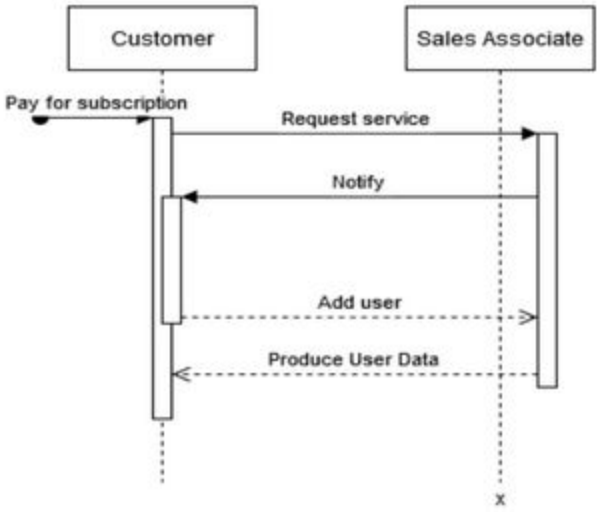
These Use Case descriptions will present a table that can be used to identify what must be done in a process that an actor possesses. For example, the Sales Associate will input data to the AlgoStock system in order to determine the amount of customers and success of the application.

Create Use Case Diagram for your IS

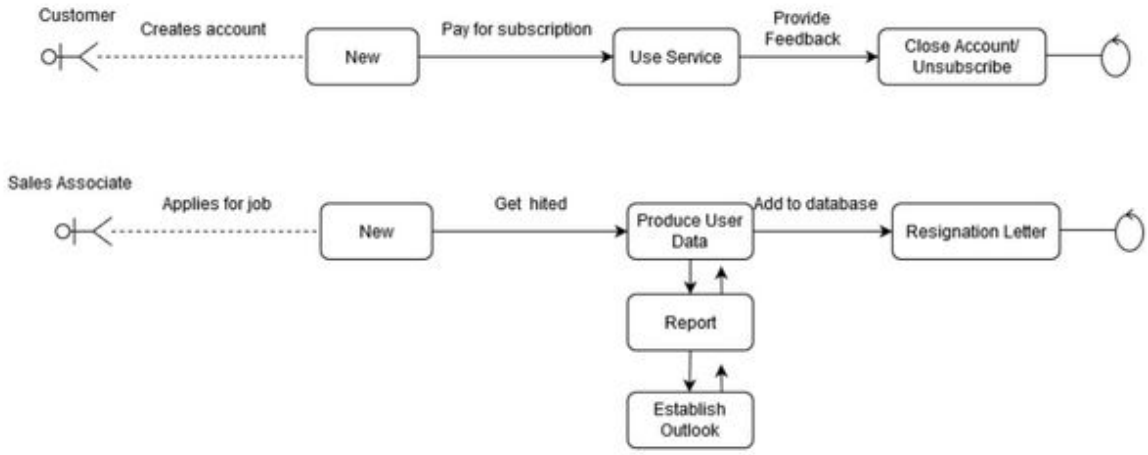


This Use Case diagram will present a visual representation of processes that need to be done by the system and identifies those outside of it too.

Provide Sequence Diagram for at least 1 Use Case



Provide State Transition Diagrams for at least 2 Objects. Identify your Objects and define the states you are modeling for each diagram



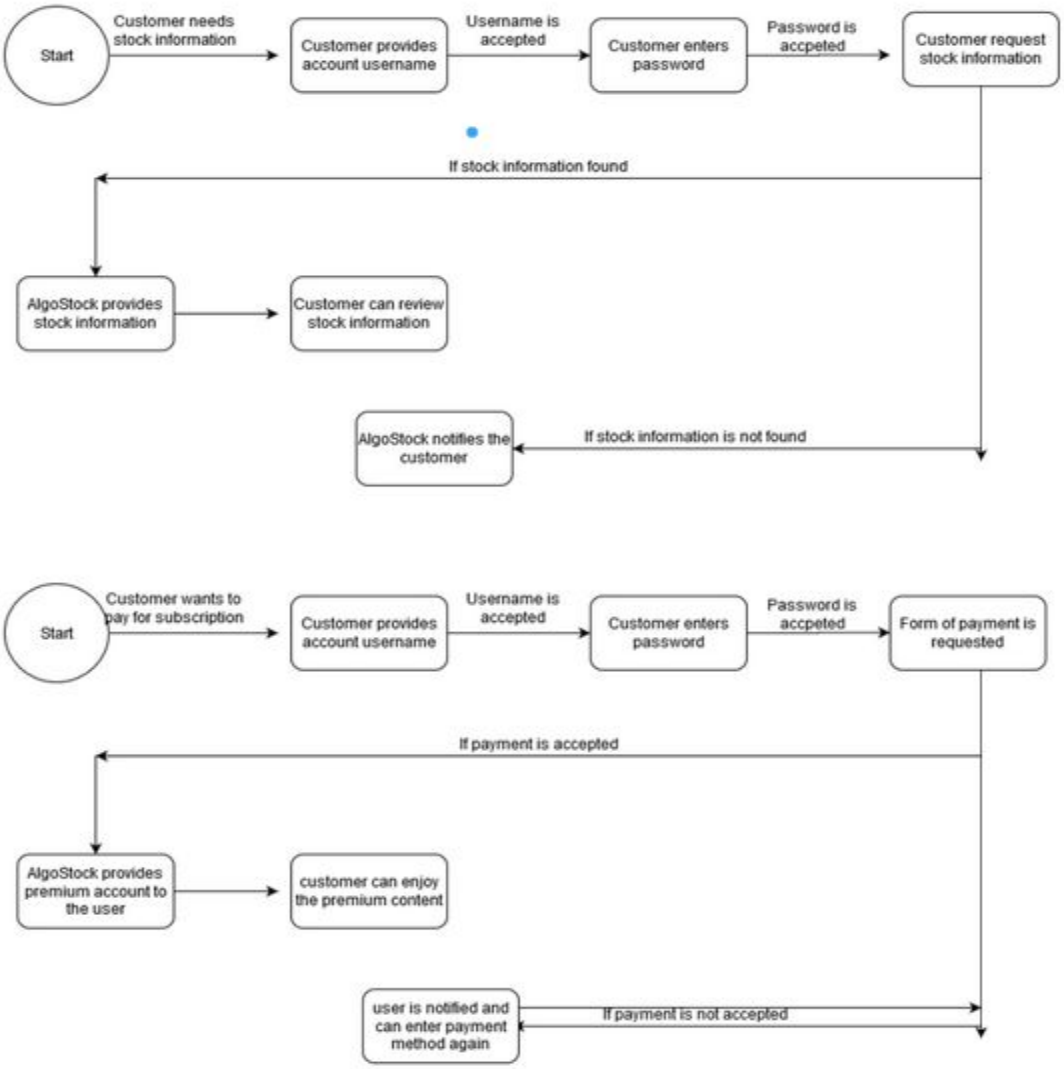
First object: Customer -> A customer can create a new account, pay for the service, provide feedback about the service and eventually unsubscribe or close the account.

Second object: Sales Associate -> A sales associate can apply for the position, once the

if an application is received he can get hired. He needs to do the tasks(methods) for his position,

and eventually he can resign this position.

Provide an Activity Diagram showing actions and events for at least two tasks



Action one: Stock information -> It shows an activity for stock information in the app. The user enters the username and password. Once log-in information is verified and accepted, the user requests stock information. If the stock is in the database, the user receives the

information requested, else AlgoStock notifies the user that stock requested is not in the Database.

Action two: Premium subscription -> It shows an activity regarding premium subscription. User provides log-in information. Once log-in information is verified and accepted, the the user needs to provide a payment method. If payment is accepted, AlgoStock provides premium content to the user, else the user needs to provide a new payment method until it is accepted.

User Interface and User Documentation

Home Page User Interface

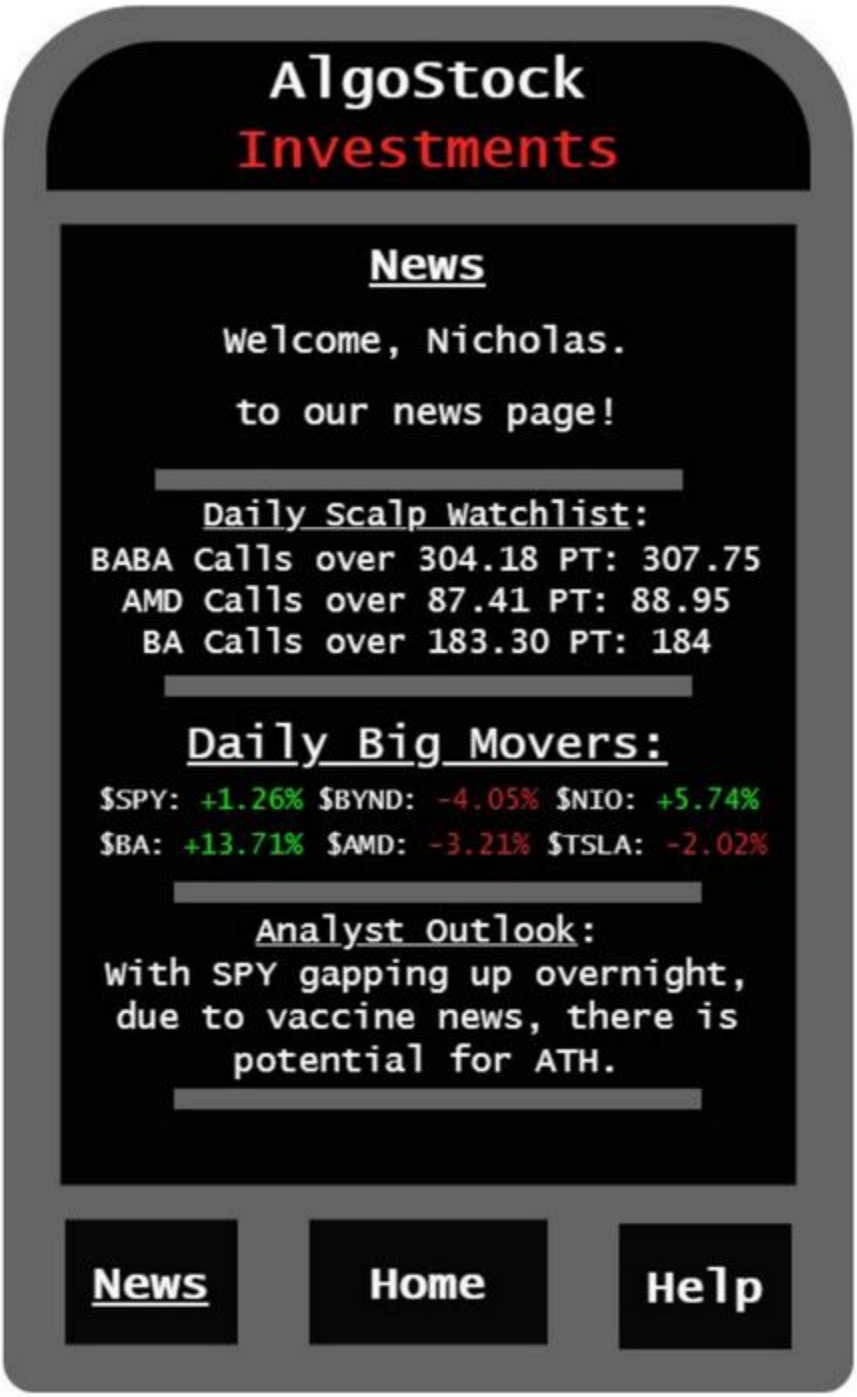
This a prototype of our investment advice application. The details might be confusing to new

traders, so we explain everything in the help section that will be presented next.

Help Page User Interface:

This is our help page that can introduce new traders to the indicators we use.

News Page User Interface



This is our news page that will be actively updated by our analysts.

Comparing UI Features with Guidelines:

I noticed that my UI has met quite a few of the guidelines. I have a common kept color scheme and design style in all of my mock ups. I labeled all of my buttons and created labeled functions of the application. I have a help page that explains what the expected parameters response is. I even included an area where a user can contact us directly for any additional information.

Describe Intention/Logic of UI:

On our “Home” page we are showing off the main selling point of our service. First this is the area where a use can see if they have an “Active” or “Inactive” subscription. Next a user can enter a stock ticker, we left the default input as “$AAPL” to present the user an example of what

We are looking for input. This part does not do anything other than display what the user inputs, just for clarification if they screenshot the page to use later. Next under “Daily Indicators” we have a whole bunch of parameters that are discussed in the “Help” tab. We essentially are asking for inputs that will help the code to present a reasonable output under the “Profitability Prediction”. The “Profitability Prediction” is going to output a response of either bullish or bearish movement based on the inputs. An example would be, if the user says MACD is green, RSI is undersold, Volume is high, and its above VWAP, then we would output a bullish prediction, as these indicators are going to indicate an upward trend. The “Help” page will be dedicated to answering emails from users and explaining what the parameters are looking to be imputed with. Finally, the “News” tab is going to be updated completely by our

analysts, we will display a morning watchlist that can present levels to enter trades on for a quick scalp. Next at the end of the day we will update which stocks moved the most, or moved with great significance in the context of its price action. Finally, we will release a nightly outlook, which will allow our users to see what our analysts think the market will act like. Everything on our “News” tab is actually accurate information for the data of 11/09/2020, and if it was a real

application, these watchlist levels could have been used, and the moves are accurately updated at closing time.

3 Most Important Guidelines:

These are the three guidelines I felt were most important in my design of the UI. The most important would be to “Maximize Graphical Effectiveness”, as I believe this is what makes the most attractive application to users. If you look at a stock trading application like Robinhood, it has horrible features and would be insanely ineffective for real traders to use, but they have a ton of people using it solely on its pleasing graphics and simplistic effectiveness. I think the next the most important would be to “Think Like a User”. Thinking like a user made me put extra work in the “Help” page, as you got to understand not everyone will know exactly what an experienced traders would know. I want to expand it even more in the future and add a guide or

include a video tutorial on using our service to limit the amount of emails we get from new users. Finally the next most important would be to “Provide Users with Help and Feedback”. I really want to be connected with my users, this can be shown in our quick ability to answer questions via email, and I would like to implement a live chat feature too. I think adding these

would make our product a lot more user friendly than it already is and being mindful of the struggle through feedback can help us to develop updates that accustom to user requests.

User Experience Features:

I have an email where users can respond with feedback and questions about our platform. I believe that this will help us develop free quality of life updates. I am confident that we can learn from our users, but I do want to add more of a way that we can connect to users, maybe by adding a survey that they can answer, or leave a review.

Online Documentation

FAQs

Q: How much will a yearly subscription cost?

A: We offer a discount of $500 for a year of access, every year after will be the standard $600 price point.

Q: Where can I contact support staff?

A: Email us at our support staff email: help@algostock.com. We plan to offer live chat support and a phone number in the future.

Q: What resources will help me become a better trader?

A: We have our main AlgoStock algorithm that will present you with information to make an educated decision on whether a stock is profitable or not, but we also include a news section that grants you daily news to our thoughts on stocks.

Menus

If you think our application is lacking/needs more options, send us at email at help@algostock.com

Picklists

Registration / Login page

The application will ask you for your username and password to verify your membership. If you don't have one, you can register

News

It will provide a section that our team has analyzed and possible trades you can make to get a profit of the current market. The news is updated each day, keep an eye on it.

Home

Provides information about your subscription status, If the status is inactive, you can select inactive and it will take you to our website to make the necessary payments. You can enter the stock you are looking to analyze, remember always to enter the abbreviation of the stock you are looking for. Once the stock is analyzed it will provide you with predictions of the stock.

Help

It provides our contact information if you need more help, also it covers so terminology/abbreviations we will use to analyze the stock.

Explanations

Our company feels as if it is our goal to provide the people with a trustworthy service to produce good quality trades. We have dedicated our time to making sure that this platform can provide just that and more. We encourage you to provide us with feedback so we can ensure that you have the best possible experience.

Exceptions

We have exceptions in place for those who input incorrect commands. This will include logging in and incorrect data entry.

Errors

Q: The app cannot analyze my stock, what can I do?

A: Send us an email about the specific stock, we will try to implement it.

Q: The app is closing when I try to analyze a stock, what can I do?

A: Update the application, or download the application again. If it is still not working, check if your phone is updated.

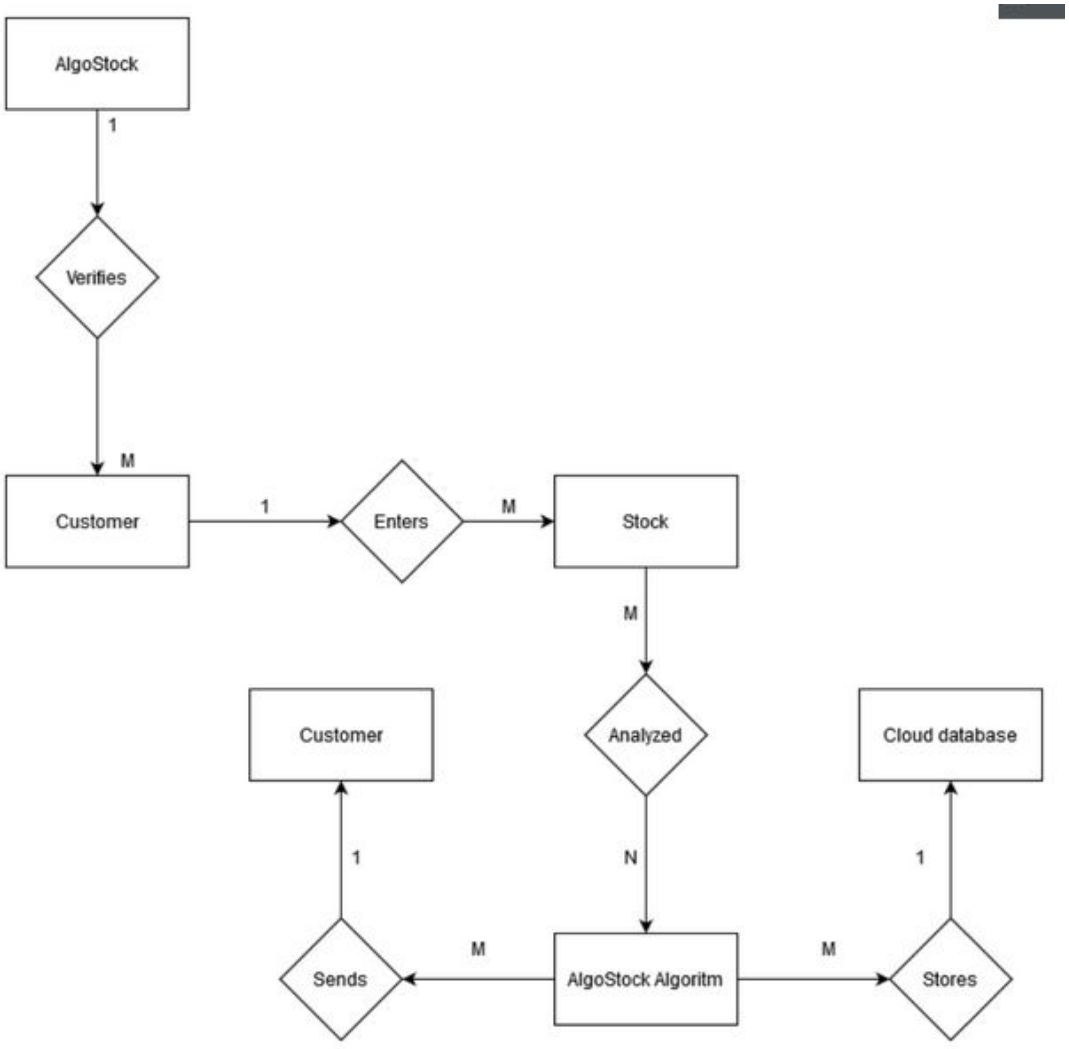
Q: I forgot my username/password, what can I do?

A: Select help on the menu, select I forgot my username/password. Once you provide your email information and security questions, an email will be sent to you with a temporary password and username information.

Help

Email-us at **help@algostock.com**

Entity Relationship Diagram



AlgoStock verifies customer information/credentials. Once the customer is verified, the customer can enter the stock he wants to be analyzed. Our algorithm will send the information to the user and it will also store the analyzed information in the cloud database.

System Environment

Transaction Volume

According to our cost-basis analysis, we would like to project a range of 50 users minimum to 100 user maximum to avoid paying an unnecessary amount of maintenance costs that could hinder performance. The range provided gives enough leeway that if we did happen to reach that 100+ users mark we could use the additional profits to purchase more size in our servers.

Storage

The way we have chosen to utilize our storage capacity is by implementing a single storage database that can hold up to 100 users on our current server. If we feel the need to have to expand our storage capacity we will use the “distributed database management system” to add additional storage in a new location. The pros of this method would be reducing the amount of traffic on our original servers, but it will cost more. We believe that we will have the sufficient funds if we reach 100+ users to begin with, so cost should not be of too much concern. This method could also prove useful if users are reporting large amounts of lag throughout peak activity times.

Response Times

Our goal to reduce response times is to collect data automatically from the client side and establish and record it on our servers. This will allow us to access the information we seek straight from our own records, without having to wait to connect to a client for information.

Security

We plan on opting into an internal hosting method for our servers. This will come with an additional cost, but we believe that security should be a guaranteed right to the consumer, and

it is reflected in the price point they are paying. If we need to establish an additional global server it will be for connectivity only, and not host any private client information, as we will host all client based records in our internal hosting server. To give our users even better access to higher priority of security, we plan to enable two-factor authentication on every subscriber account.

Hardware

AlgoStock plans on having a server farm located directly in the headquarters. This will allow us to have more control over the day to day operations of maintenance. We plan on adding additional servers via the cloud method, but the most important information will always be stored on our local servers for protection and integrity of our users.

Software

AlgoStock wants to retain a horizontal/vertical system method. This includes having inventory systems and a means to reach requirements. We of course will implement typical enterprise applications, such as payroll and processing systems. This will boost the efficiency of our company and let our IT department contribute to more pressing tasks.

Implementation Requirements

User Training Plan

Sales associates will receive training to make sure they understand how this application will run in our business operations and how the user can be involved and get a great experience. Coders will provide training to our sales associates. Based on the size of our company, we think coders will provide a mix of general background knowledge and detailed information so the sales associates can have the right answers. Tutorials will be our method to train our employees,

because it is interactive and provides a lot of information to the employees. Coders will use PowerPoint and make sure to provide examples with most of the slides to answer all the questions a sales associate might have. Training sessions will be general. Our sales associates team will be relatively small, it will be more efficient to train at the company's location. Additionally, we will upload our training sessions online, the employees can review them if they need to increase their knowledge about our product. After, we will run a simulation that will give our team a better understanding of what we are doing right and where we need to improve. It will give our company valuable experience and build confidence for everyone involved with the new application. We will give a PDF instruction to every new member of our premium plan, as well have a video in our youtube channel addressing all the information a user might need to start their journey with our application.

Test Plan

We plan to opt-into a Test Plan that will revolve around the creation of betas that will be updated according to user review. This will all be done prior to release of our application to the public. The major concerns and results we are looking for revolve around how well made our User Interface is and how effective our product is found to be. We want to take into consideration ideas that users provide with us to make the application have as many useful features as possible. We plan to continue creating application betas after the official launch, this can help push us in the direction that our users are looking for.

Start-up Processing and Data Acquisition

We plan on implementing numerous promotional events that advertise our service to the public prior to our release. We want to allow users to test our software and get a feel for what

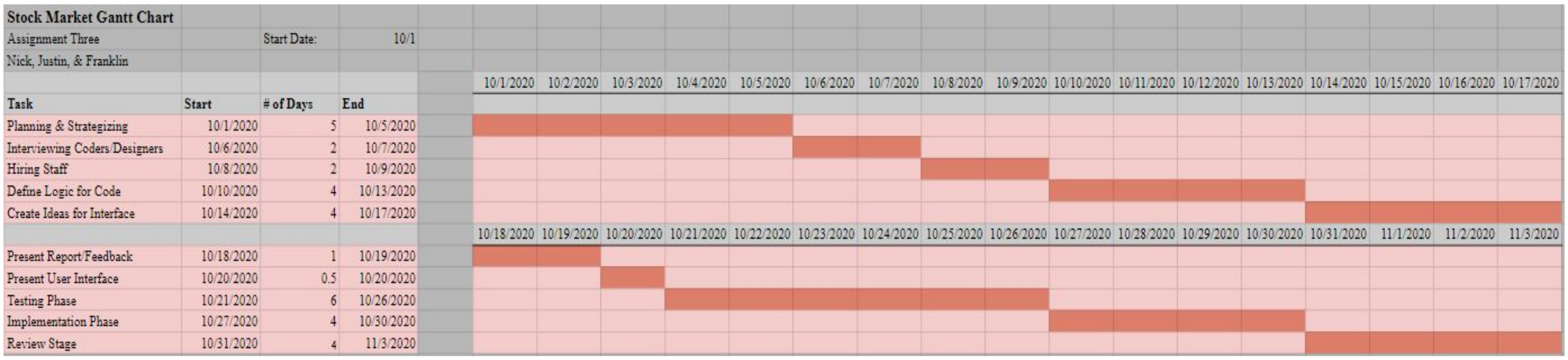
we have to offer, this can help generate traction for the official release and ties into gaining additional data. When we release our application we will ask our users if we can collect their data and store it in our databases. We can use this data to track our audience and understand better who is using our application, this can then be sold to individual advertisers for some additional wealth if we feel like it is necessary.

Time and Cost Estimates

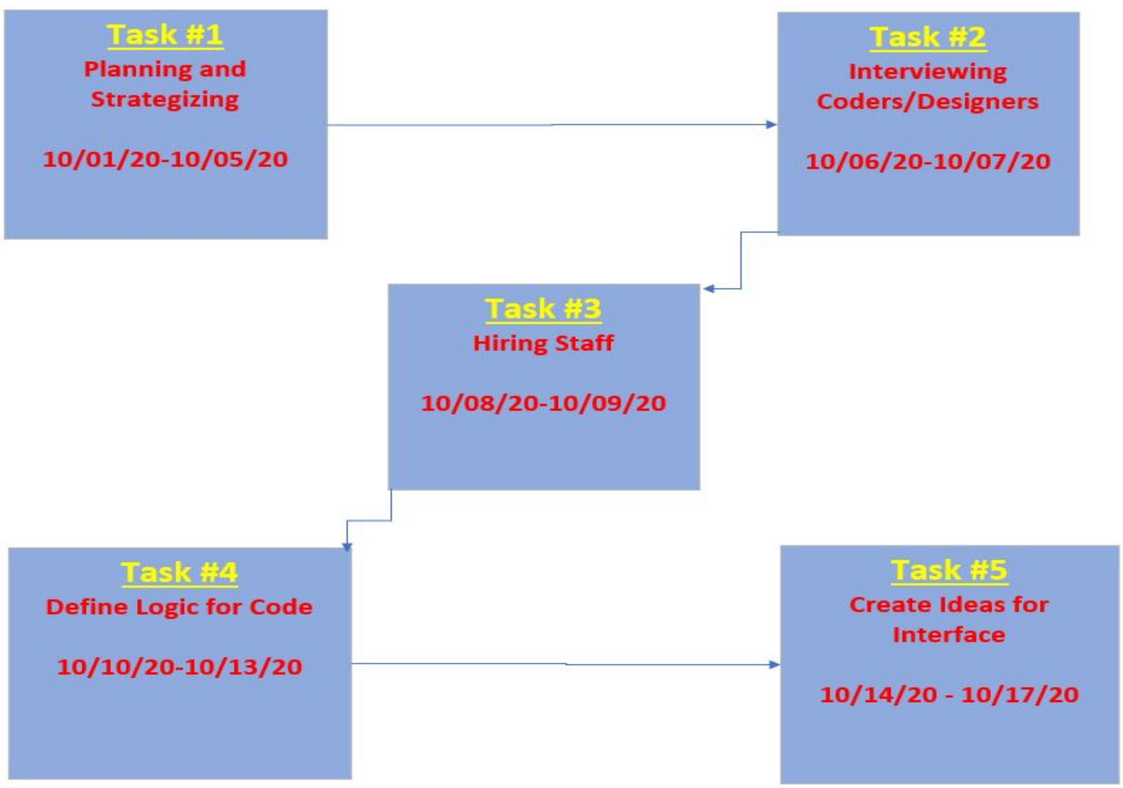
STOCK MARKET PROJECT					
	YEAR	COSTS	CUMULATIVE COSTS	BENEFITS	CUMULATIVE BENEFITS
	0	$11,665.60	$11,665.60	$12,500	$12,500
	1	$6,582.80	$17,748.40	$30,000	$42,500
	2	$6,582.80	$24,331.20	$60,000	$102,500
	3	$6,582.80	$30,914	$120,000	$222,500
				ROI:	Best Case: 719.7%

Cost-Benefit Analysis

As seen above, we have provided a payment plan that includes a huge ROI of 719.7% of our initial investment. This will allow us to make enough money to fund additional services and expand upon our current operations if we please to do so. We can also have enough additional funding at the end of our first year to create additional globalized servers if we reach our maximum capacity of 100 active subscribers. On the following page we will present how we plan on going about our endeavor on a Gantt and Pert chart.



Stock Market Gantt Chart			
Assignment Three		Start Date:	10/1
Nick, Justin, & Franklin			
Task	**Start**	**# of Days**	**End**
Planning & Strategizing	10/1/2020	5	10/5/2020
Interviewing Coders/Designers	10/6/2020	2	10/7/2020
Hiring Staff	10/8/2020	2	10/9/2020
Define Logic for Code	10/10/2020	4	10/13/2020
Create Ideas for Interface	10/14/2020	4	10/17/2020
Present Report/Feedback	10/18/2020	1	10/19/2020
Present User Interface	10/20/2020	0.5	10/20/2020
Testing Phase	10/21/2020	6	10/26/2020
Implementation Phase	10/27/2020	4	10/30/2020
Review Stage	10/31/2020	4	11/3/2020



Risk Management Plan

ID	Risk and Consequence	Probability	Impact	Priority	Mitigation Response
1	Define logic for code- Release Schedule	2	3	6	Hire more experienced engineers
2	Hiring Staff - Budget	1	2	2	Reduce staff
3	Create Ideas for Interface - Resourcing	2	4	8	Readjust team's goal
4	Testing Phase - Release Schedule	1	3	3	Implement better design
5	Implementation Phase - Release Schedule	2	5	10	Choose the right server
6	Faulty Code - Loss of users	2	3	6	Longer testing

Identify risk in your project

Our scope is defined towards the problem that researchers find 99% of stock market traders are not profitable. This can be taken advantage of by utilizing a service targeted at the 99% who are not profitable and providing them with ways to make money. We have a huge target market here and as long as the substance of the service we provide is good enough, people will buy it with the hopes of creating revenue. The problem will occur in our scope via poor quality service. Stakeholders would probably invest in our shares if we present them with great quarterly earnings, our industry would most likely be something around financials mixed with

tech, which is a great area to be in as these are popular sectors. The big risk in stakeholders might be how they are taking a risk on investing in a company who is giving people investing advice, so our service would have to portray us as a company who will offer a high ROI. Our budget is set fairly low, so we have the option to add to it as needed. The risk involved will come from paying a programmer or graphic designer we believe is qualified, but then underperforms causing us to have to spend additional revenue. The schedule we presented is going to be risky in the sense that the tasks could theoretically take longer if we find errors. I think risks that could come about heavily from external factors is if the market becomes super volatile to the point where it is unreadable by our service.

Analyze the Risks

Analyzing the qualitative risk involved can be done via a risk matrix:

			Chance of Occurrence (1-5)			
		1 (unlikely)	2	3	4	5 (very likely)
	5 (worst)	No customers	Lack of users	Expensive staff		Volatile Market
	4	Budget too high	No investors			
Consequences (1-5)	3		Too expensive	Server Issues	Late on schedule	
	2		Code too hard		Coding errors	
	1 (minor)	Poor service	Ugly interface			
		Green to Red measures how risky an outcome will be				

Analyzing the quantitative risk is done via a what-if analysis:

Biggest Risks	Percent Chance	Cost	Severity (1-5)
Coding errors	15%	$2,000	4
Late on schedule	25%	$500	2
Server issues	20%	$200	1

Risk Response Plan

ID 1. Define logic for code, if the algorithm has a runtime of O(n), and it is the 4 day we are working on the algorithm, the team most likely will fall behind the release schedule. To reduce or eliminate the risk we might have to hire experienced engineers, or in the worse case scenario we might have to hire one or two more engineers until the logic code is done and the runtime of the algorithm is O(1).

ID 2. Overstaff, if we hired more employees that needed, we would simply announce to the staff that we might not need as many employees as we thought as it would affect our budget. To reduce or eliminate the risk of overstaffing we have to come with perfect or near perfect planning and strategizing for the whole project, that means we might need to hire an experienced project manager to help us with it.

ID 3. Create ideas for the interface, if the team doesn't agree with the right idea for the interface of the app and days are running out. To eliminate or reduce the risk of lack of decision, the CEO of the company will have the final say and he might decide we need to readjust our goals for the project.

ID 4. Testing phase, unit testing gives the team an early feedback of the code. If the early feedback is saying our code isn't working. To eliminate or reduce the risk of poor testing, we need to reevaluate our ideas and come up with a better design that can be broken down into discrete units, this means we need a main function that knows how to run the tests, check the results, and print a report.

ID 5. Implementation phase, probably one of the most important tasks because it will affect the release schedule. To reduce or eliminate the risk we need to find the right servers that assure

us that is going to work with our code. We might need to delegate one person to find the right server so the implementation phase goes smoothly.

ID 6. Faulty Code, if our code isn't completely right and reads the stocks wrong in any way, it can lead to the loss of money of users. If our users are losing money, the ratings go down and it is a hole we could never get out of. A way to reduce the chance of this going wrong is having a longer testing phase. If we test our code more, having it deal with more situations, we can tweak any problems we see before release.

Conclusion

AlgoStock has predicted an average of $12k spent in our first year of development to create the foundation of our project, including graphic design of the application, coding, infrastructure, advertising, and all maintenance/server costs. This is our target price, but we have funds set aside if our budget needs to be increased. We then find that our subsequent years will only cost $7k at maximum, yielding our organization a huge ROI. We have set dates for release by the end of the year. We currently have a system that tries to correctly predict future stock trends and

movements, based on common indicators. This will prove useful to our customers who can also see what our analysts have to say about the stock market, which will be updated daily. Management will have to ensure that we keep on top of regular maintenance and upkeep the performance of our servers.

Implementation

AlgoStock will implement a strong plan that can keep everyone in touch with what is going on daily and what we expect at the end of the week, every week. We plan to require a report from

our management at the end of each day, as we want to record hardware and software performances, in addition we will also request monthly server updates to see how much traffic we can support. We will need a security login system that can be implemented by coders and we need to have enough storage space to save all of the information being logged daily. We plan to use that storage for server rollbacks if we need it. We are requiring our system to be implemented with users that test our software without prior knowledge to see how they behave and use our product. This will be our first stage of implementing the project, as we plan to adjust based on the results. Then we will revise and collect the initial data to provide to our coders to update the product. We want to ensure that all users are able to cohesively understand the product and what they are getting, so we offer many forms of ways to learn about how to trade and utilize our service to the fullest extent.