



YoloTrader

Contributors:

- Owen Murphy
- Prince Kalu



[Abstract]

YoloTrader is a stock analysis service targeted to retail Investors. It offers stock price tracking, current stock related news articles, stock organization, and houses artificial intelligence framework that analyzes stocks on user request. YoloTrader supports all stocks listed on NYSE and NASDAQ, as well as their market indexes.



[Story]

Investing is a worthwhile expenditure, however it's also very risky. For someone to make money in the stock market, someone else has to lose money. Before the digital revolution, the financial institutions controlled the stock markets, but as the stock market digitized, that power balance began to shift. A new term was coined, retail traders, retail traders are people like you and me, normal people looking to invest their hard earned money in the right places.

Once retail traders came onto the scene, the market was disrupted, and the institutions began to lose control of the market. But this new shift in power didn't last long, suddenly a new player entered the stock market. Robots, armed with proven stock trading algorithms, complex AI Frameworks, and the ability to react to and predict the market faster than a human can hit the buy/sell button. Today, roughly 75% of trades in the stock market are executed by automated trading programs. Programs that are only accessible to big institutions like JP Morgan, Morgan Stanley, Merrill Lynch, etc...

Once again, the power balance has shifted, and now another revolution is needed. YoloTrader was made specifically for that revolution, to once again democratize the stock market.



[User Manual]

-View the most volatile stocks of the day: To access a list of the day's most volatile stocks, view the home page. The volatile stocks will be categorized into two separate lists, their difference being their overall trend(up or down). You can click on any of these tiles, and the stock will be opened in another window.

-View a stock & its quote data : To view a stock, you must first open the stock. This can be done by typing the company name or ticker into the search bar, or clicking on a stock tile from the homepage.

-Catch up on news: To catch up on stock related news, you can either see general news at the bottom of the home page, or you can view a stock and scroll to the bottom of the stock page, where only news related to that particular stock will be displayed.



[User Manual]

Analyze a stock : To get an analysis on a stock, you must first view the stock. The analysis will be displayed onto the stock page. An analysis of Buy/Calls means the AI believes the stock will move at least +5%, from the current price at the time of prediction, within the next two weeks. An analysis of Sell/Puts means the AI believes the stock will move at least -5%, from the current price at the time of prediction, within the next two weeks. An analysis of Buy/Hold means the AI believes the stock will not move more than 5% in either direction, within the next two weeks. Now the AI is not aware of news, it trades solely off technicals & market sentiment, that said, if a news even comes out, always check with the bot again, as it may change its prediction if the news is strong enough to change market sentiment.



[User Manual]

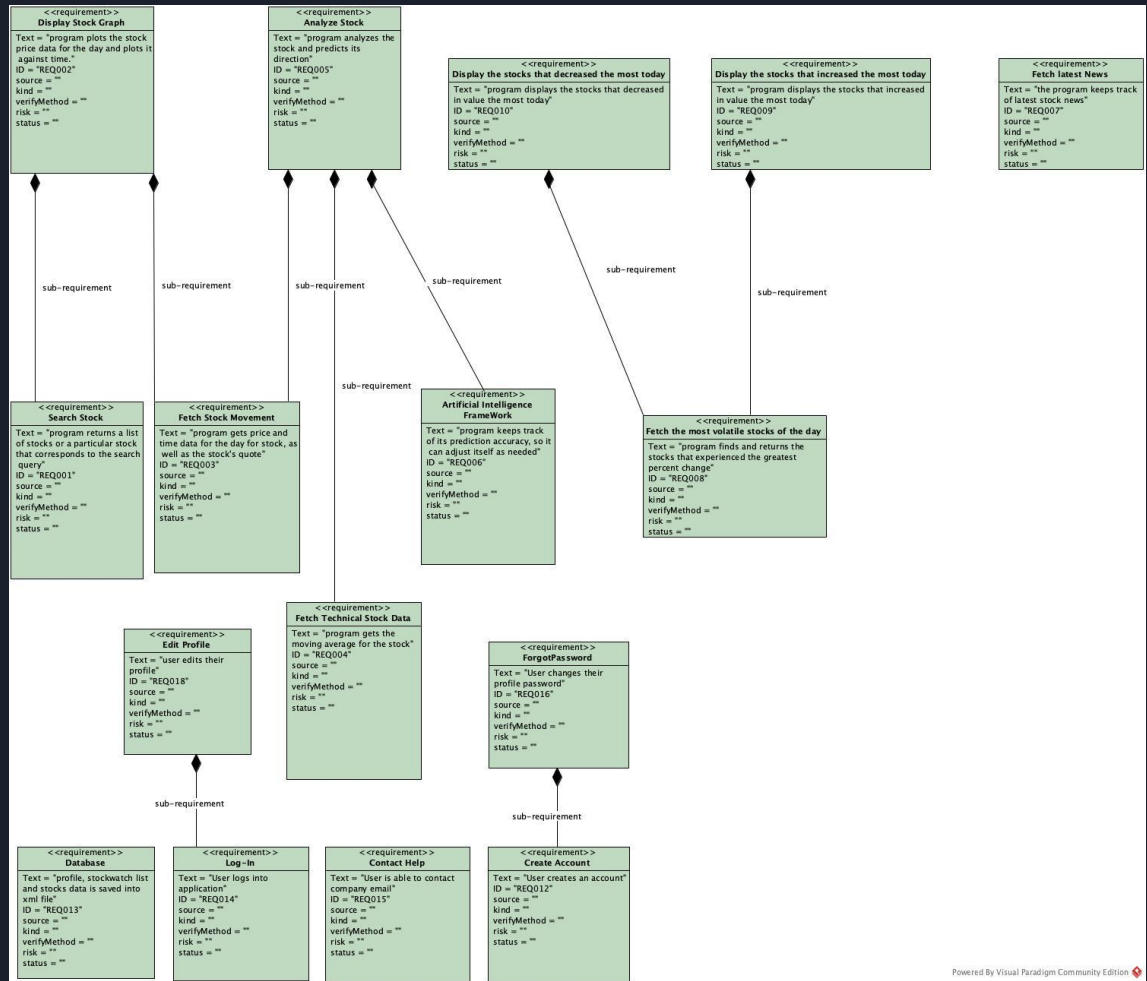
Forgot Password:

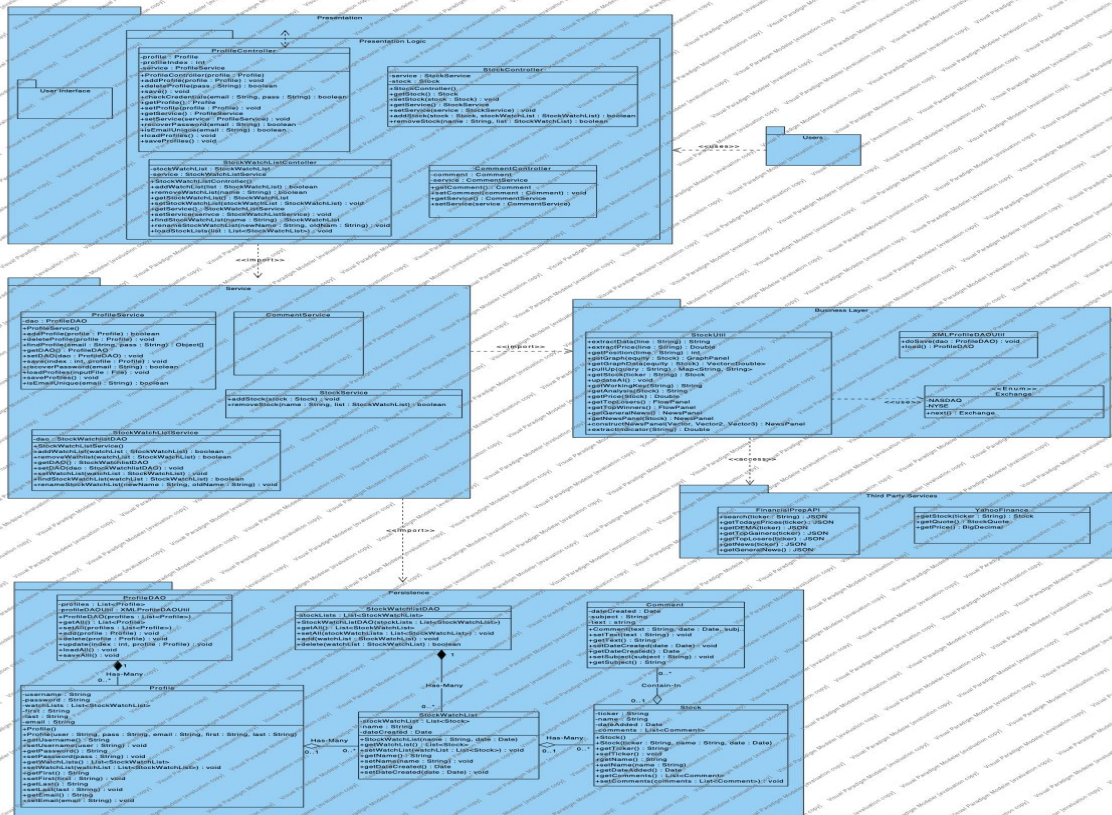
- This allows a user to reset their password given their associated email (if exists)
- Click the “help” button in either the log-in or create account form
- Click Forgot Password button and input email used to create account
- If email exists, an email is sent to it with the new password

Change Password:

- Got Profile Page
- Click “Change Password” button near bottom of page
- Input old password
- Input new password, and again.
- If old password is valid and new passwords match, password is changed

[Requirements]

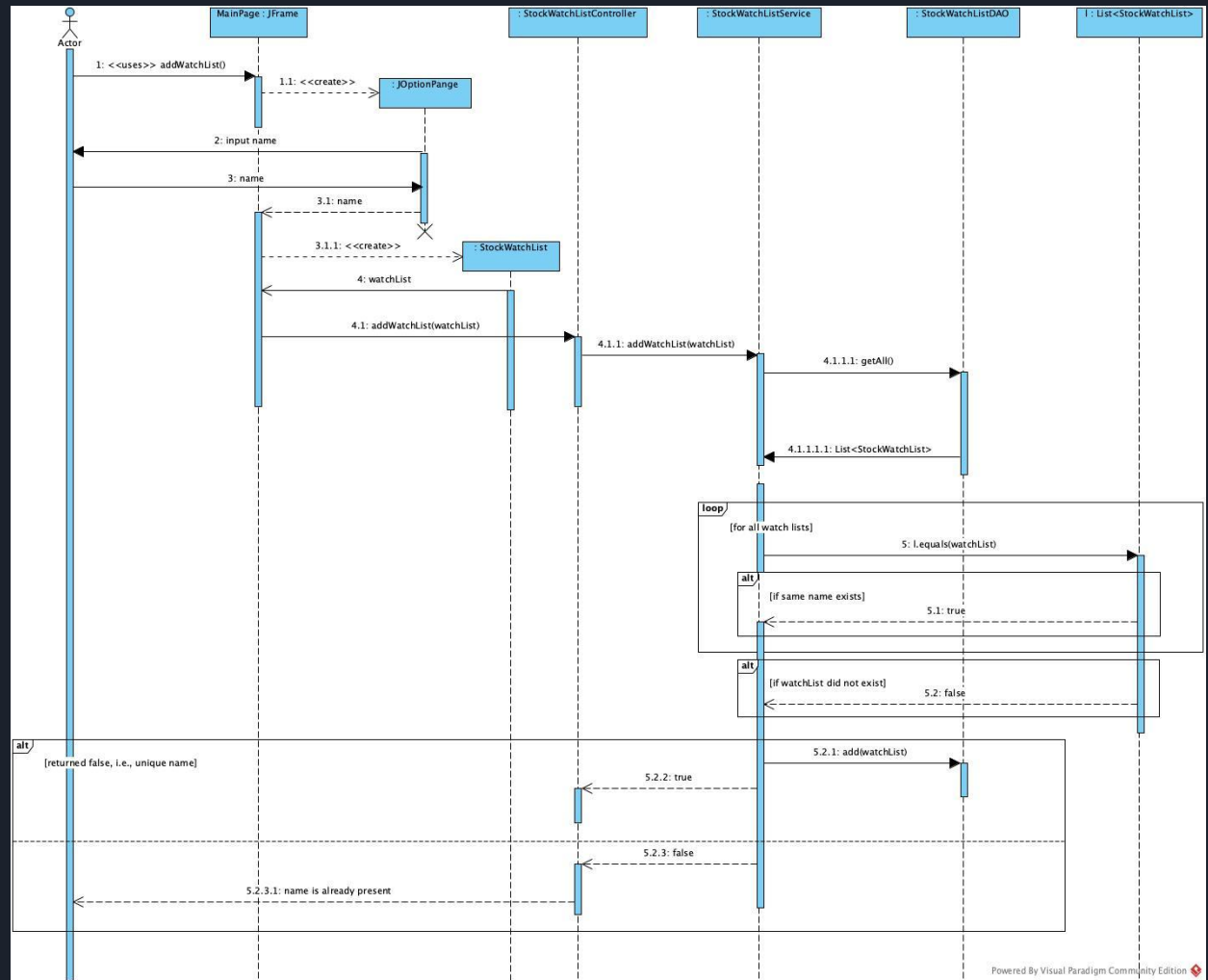




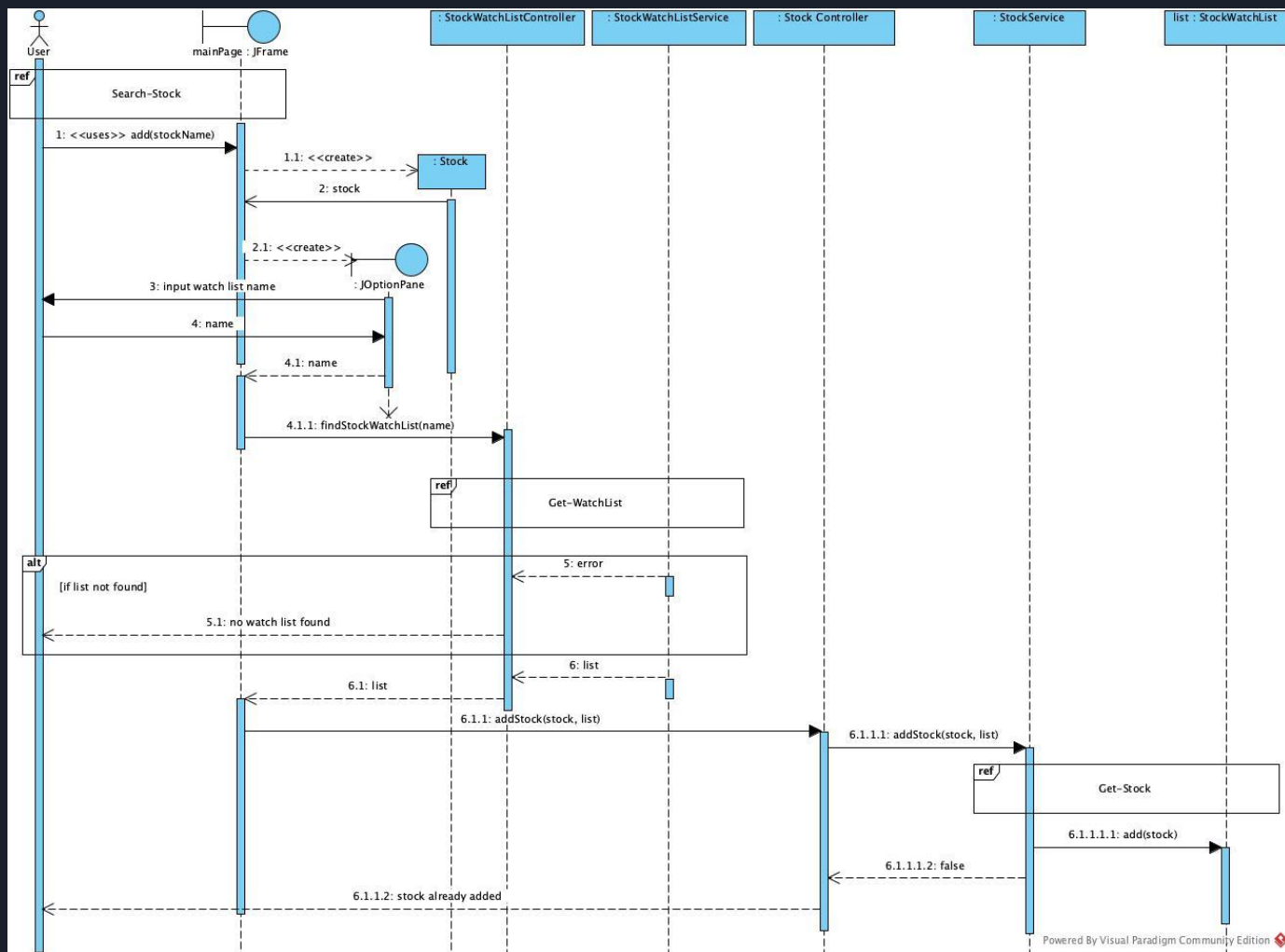


[Sequence Diagrams]

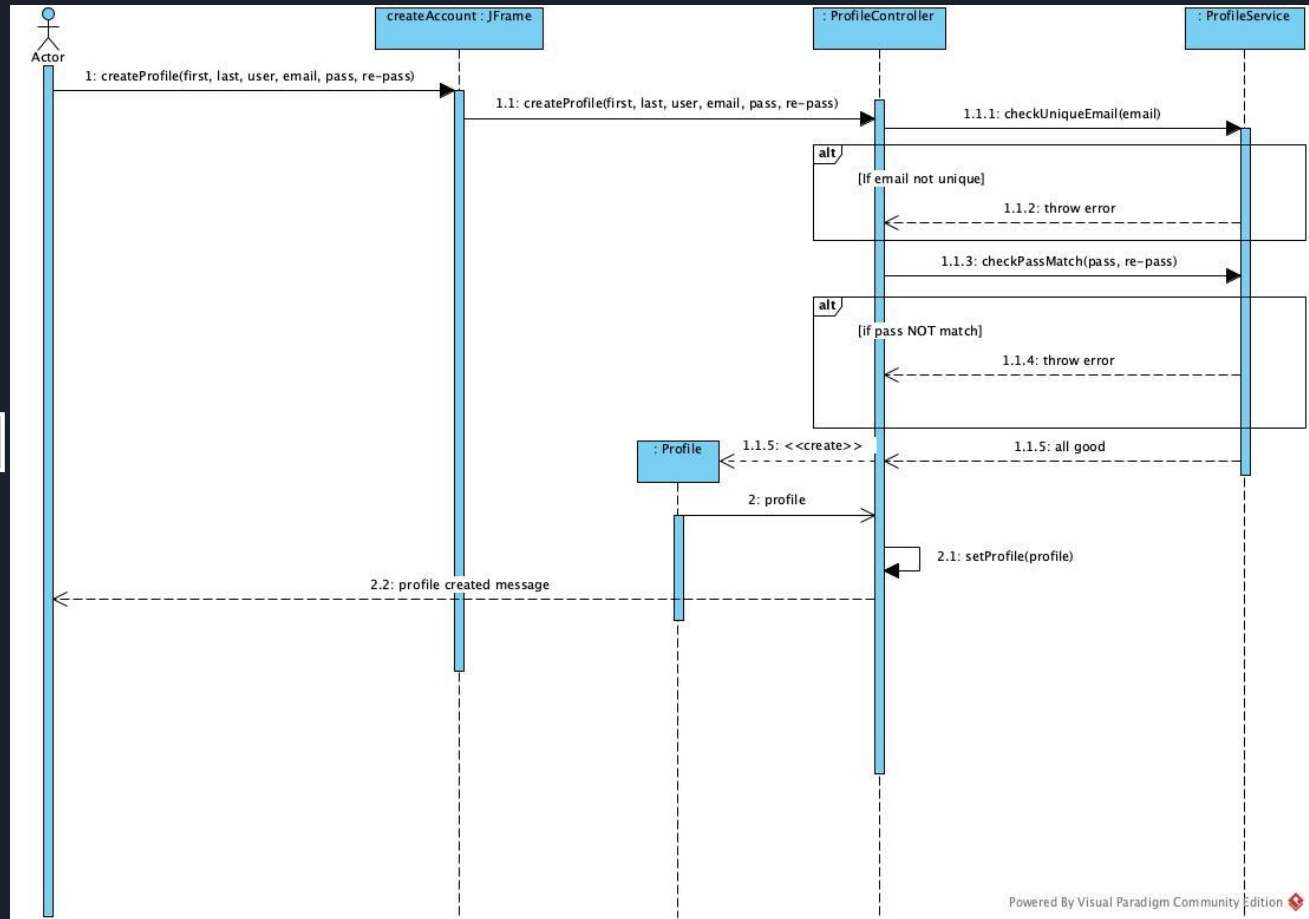
[Add-WatchList]



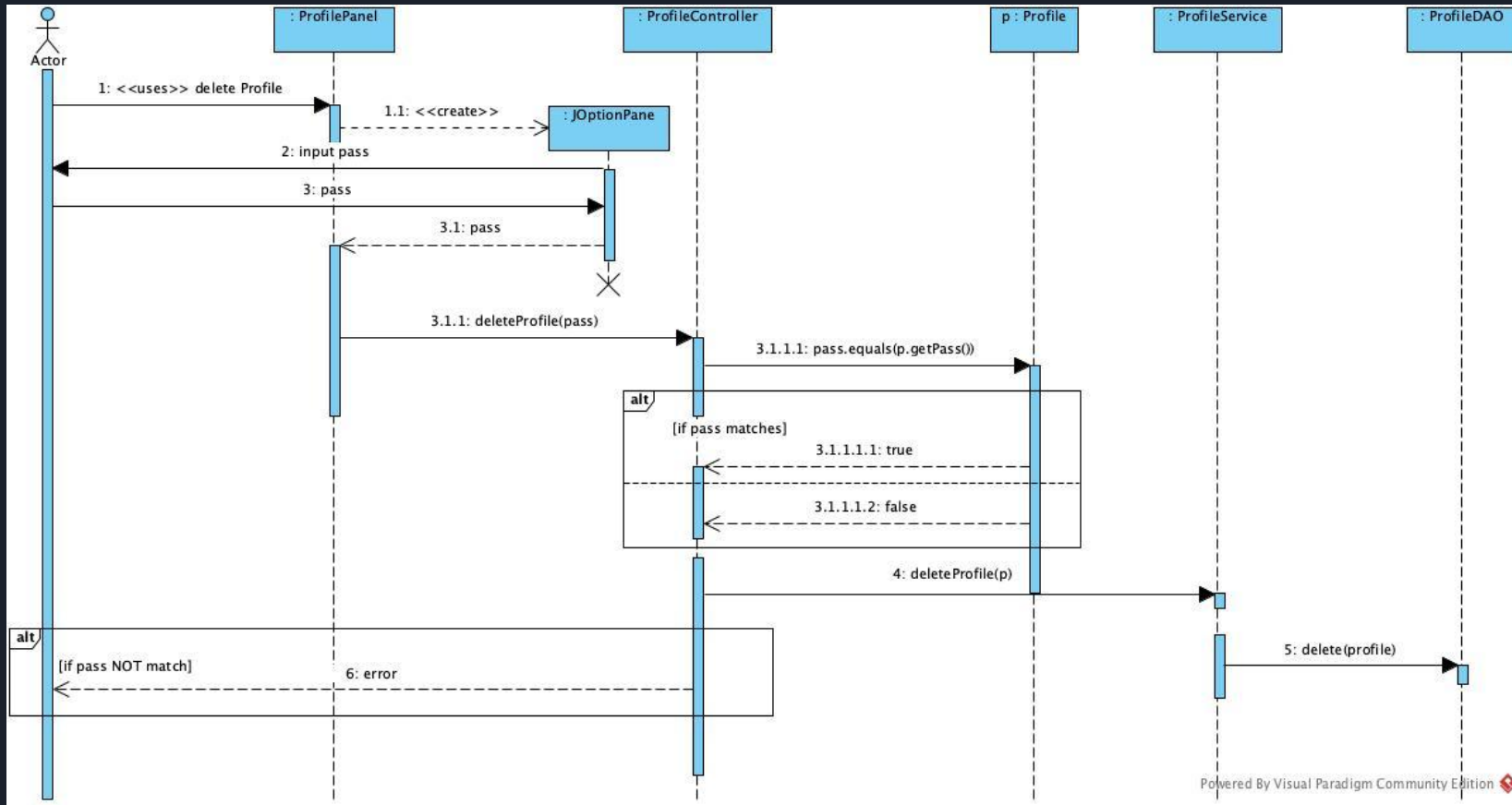
[Add-Stock]



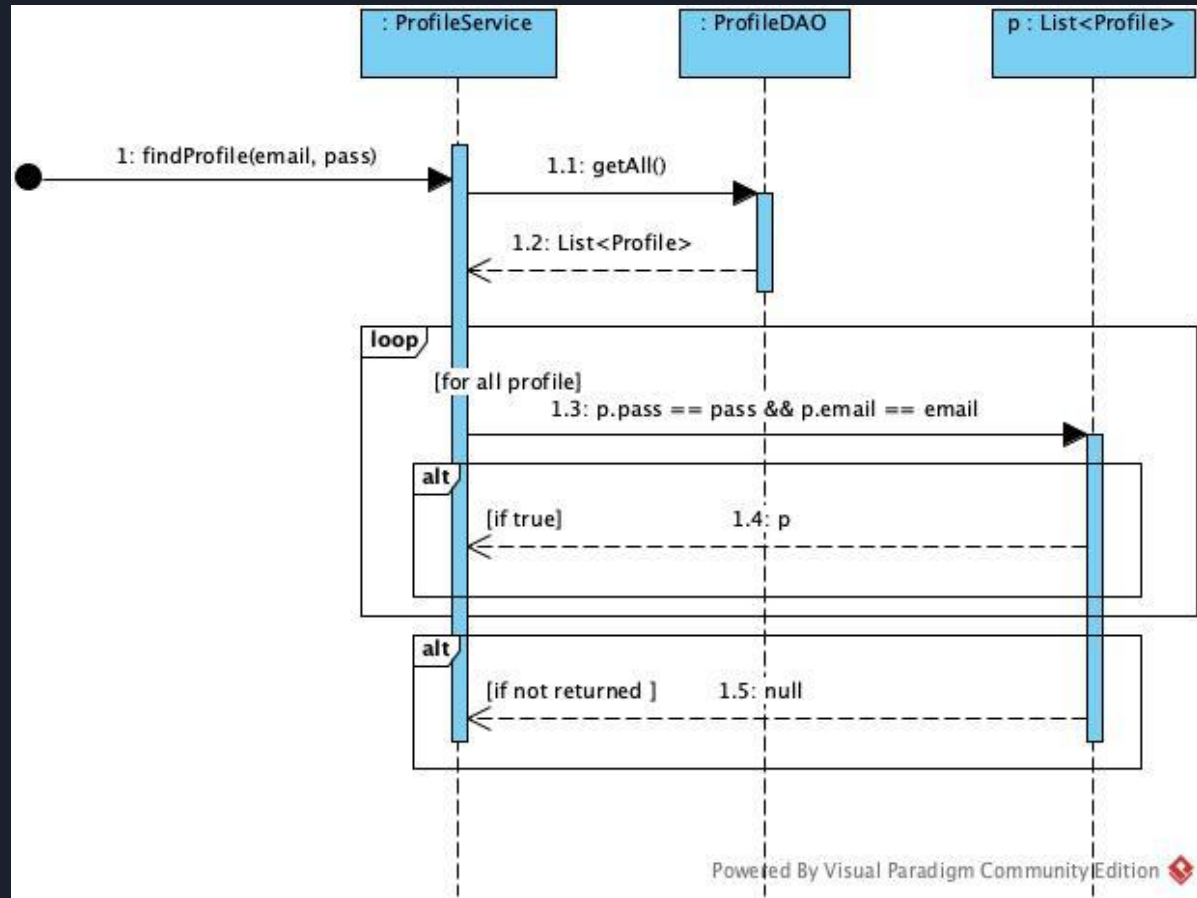
[Create-Account]



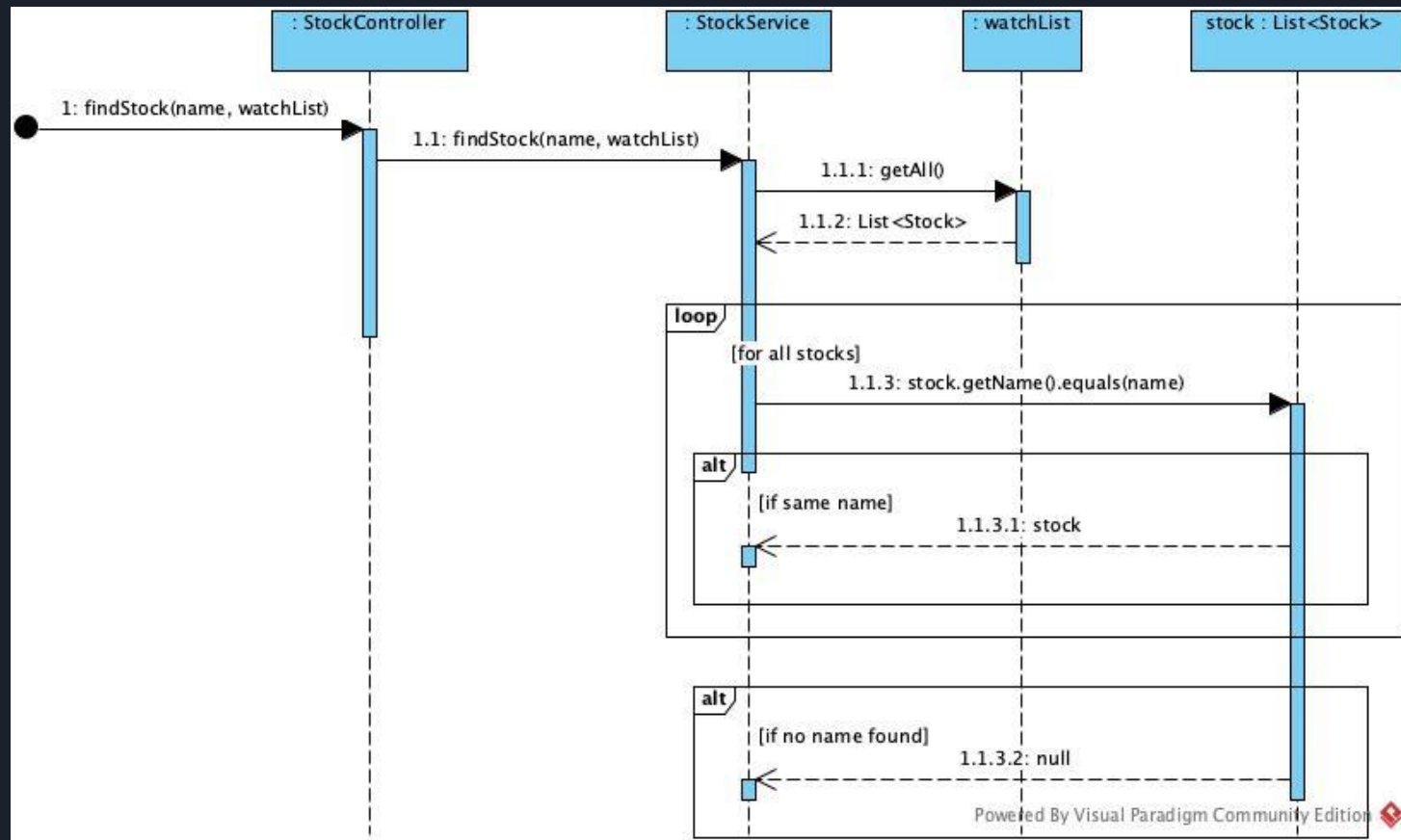
[Delete-Profile]



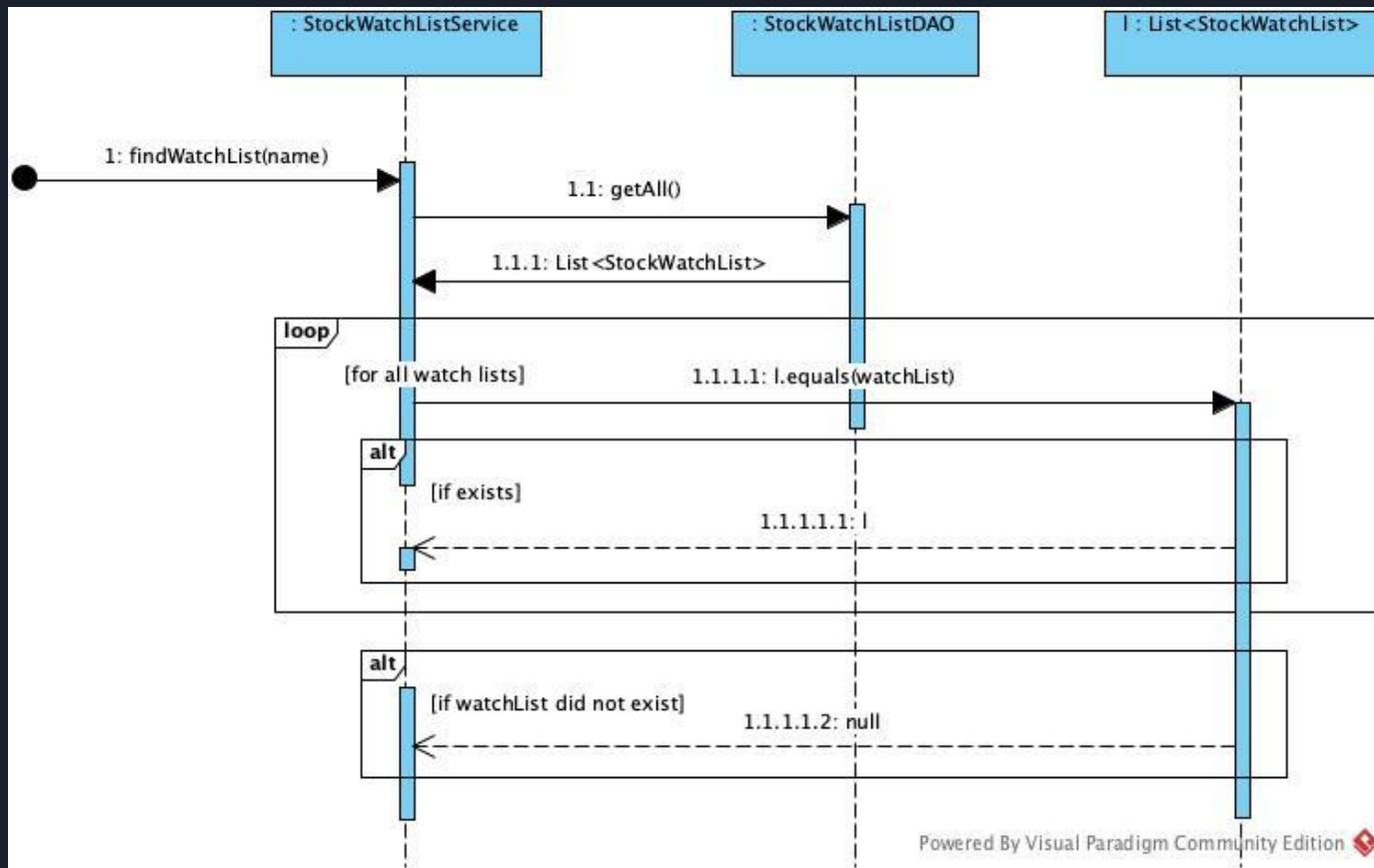
[Get-Profile]



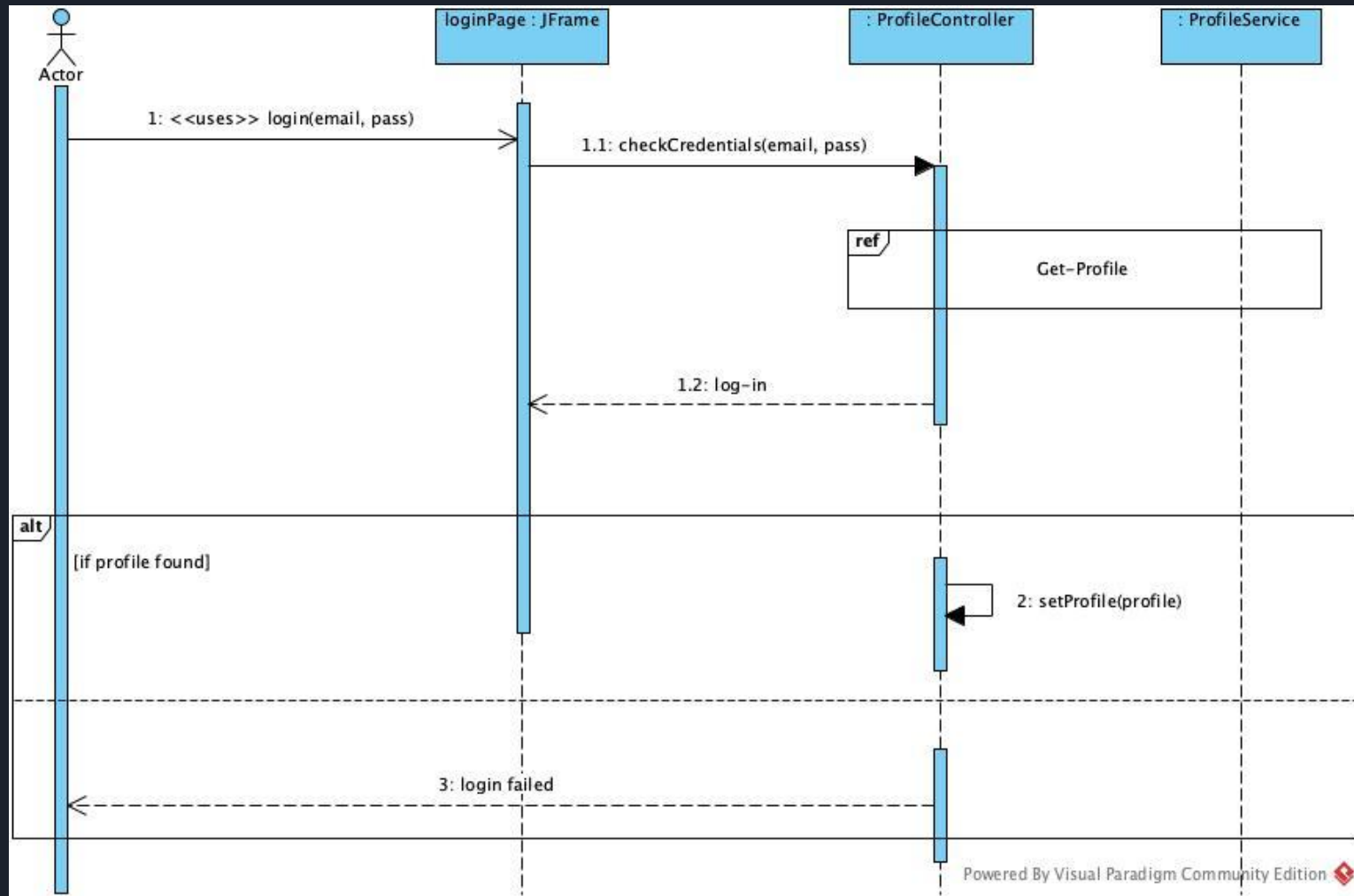
[Get-Stock]



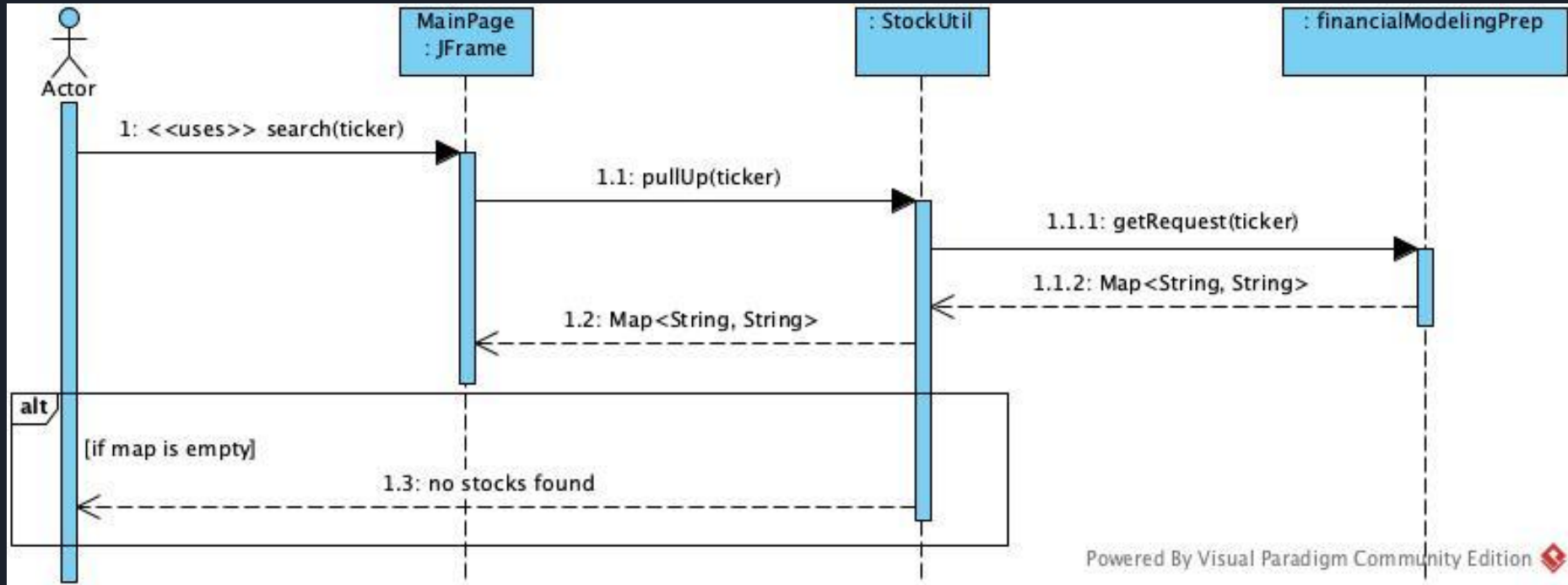
[Get-WatchList]



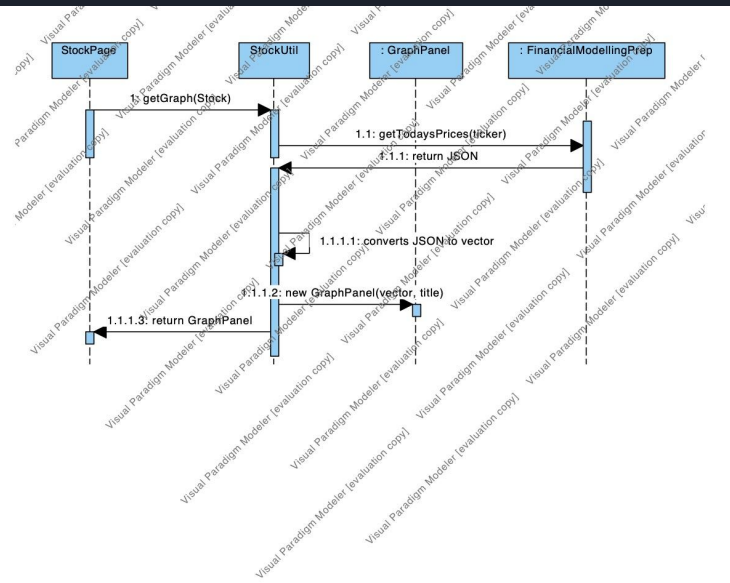
[Log-In]



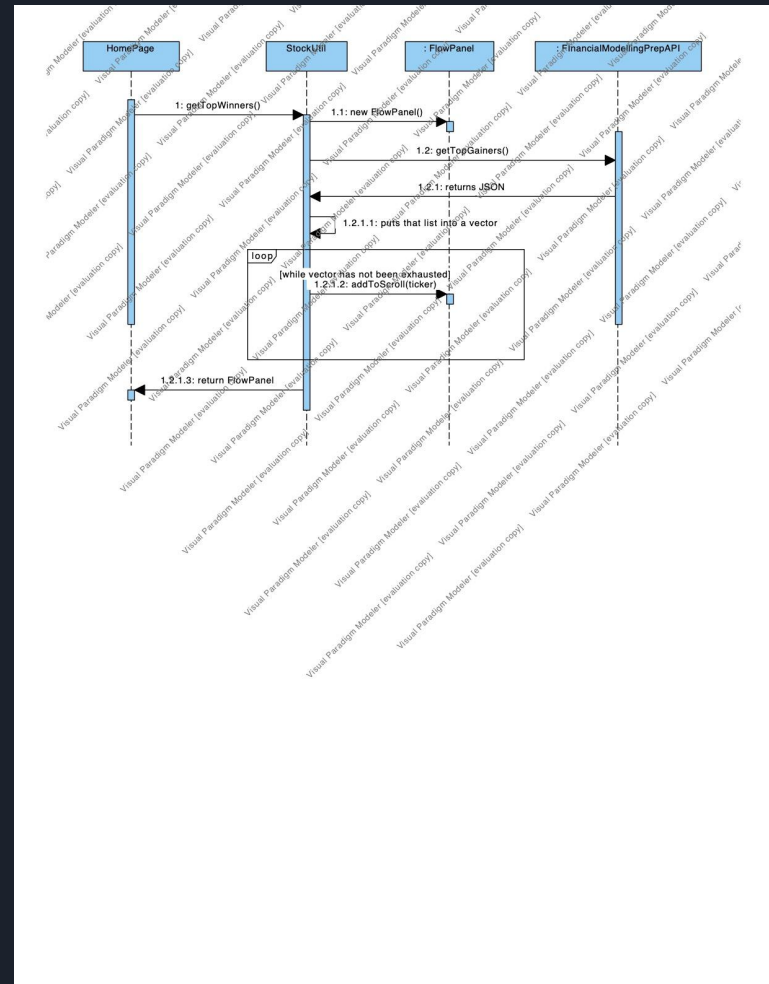
[Search-Stock]



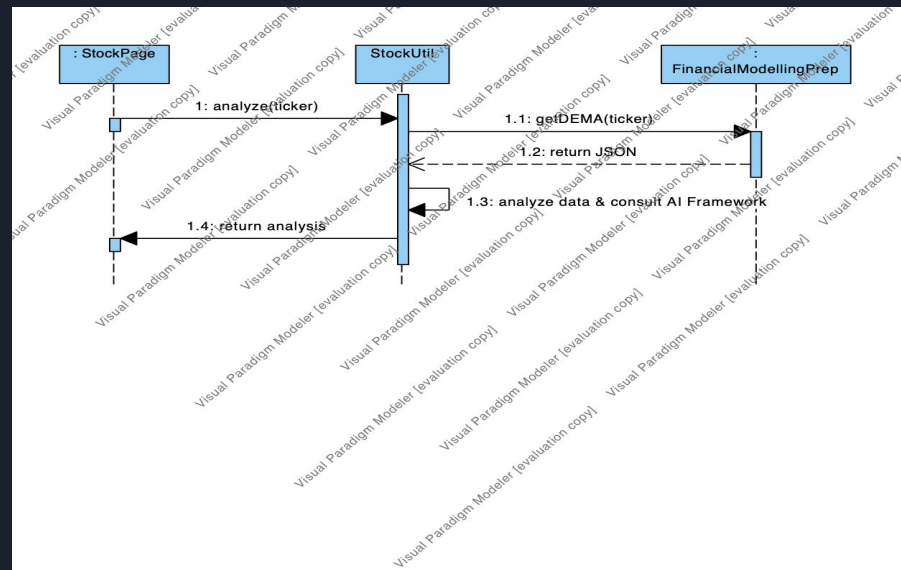
[getGraph]



[getTopWinners]



[AnalyzeStock]





[Evaluation/Testing]

With respect to evaluation and testing, our program is bug free. The only issues we have concern performance. The AI framework takes a couple seconds to analyze the stock upon the application starting and when the user clicks on a stock to open it. The analysis must be completed before a stock opens which results in a small wait time for a stock to open. There are several workarounds for this. One would be to launch the stock page without the analysis, and allow the user to click on a button to show it, should they desire to see it. That way the user knows the AI framework is working to produce an analysis. This fix wouldn't improve performance, but rather improve user experience. Another method would be to have a separate thread for getting the stock's analysis. This allows the AI framework to get the analysis in the background while the stock page is being opened.



[Time Log]

TIMECARD

OWEN MURPHY

DATE	HOURS	DESCRIPTION
ITERATION I		
09-17	1	Created Schedule
09-19	4	Refined Schedule, contributed to Requirements list, designed Wireframes, updated web pages, setup business email
9-21	1	Created Use Case
9-23	0.5	Updated webpage by centering headers
9-26	1	Finished 3rd Use Case, created SSD's, System Operations, Operations Contract
9-28	2	Domain Model, Iteration I
9-29	6	Refining Domain Model, Use Case Diagram, SSD, requirements
9-29	5	Learning git commands, repositories, etc., schedule, use cases, compiling project
SUBTOTAL:	20.5	
ITERATION II		
10/20	1.5	Developed pie chart
10/21	1	Created Sequence Diagrams, added link to bar meter, & updated project schedule
10/22	2	Created more Sequence Diagrams, Design Model
10/23	1	Revised Design Model, Started Google Slide
10/24	1	Messed with Stockwits API, looking into getting search function working
10/25	.5	Team Meeting
10/26	1.5	Sign In Window created
10/28	2.5	Fixing diagrams, finished window class prototype, created prez
10/29	2.5	Updated project, prez ready, updating design model
10/30	5	Updated Design Model, updated sequence Diagrams
SUBTOTAL:	18.5	
ITERATION III		
10/31	1	Finished initial design for window login, etc.
11/06	4.5	Started UI design, swing, etc.

11/07	4.5	Finished WestPanel look, added north panel title, committed to GitHub, started working on profile section UI
11/08	2	Continue Stock & Profile UI, added README.md to UI directory
11/09	3	Continued working on Profile, fixed relative link in .md file under the "UI" directory in GitHub, Team meeting
11/10	3	Added a progress bar to index.html, debugged and redesigned the profile panel, Add actions to the Stocks panel, experimented with the JLabel as the profile picture
11/11	1	Continued on the about and help sections
11/12	2.5	Refurbished the login & create account forms
11/13	3	Made Mods to Design Model, initialized model class, DAO class/interface (s), took image capture of prototype so no need to run to see what it looks like
11/14	4	Revised Design Model and added action to "search" button, continued revising design model, wrote comments, started controller & service classes
11/15	1.5	Form controller, service, DAO
11/16	2	Created mapping for database using xml file
11/17	2	Configuring database still
11/18	2.5	Working on logging in, creating account, adding stock watch list
11/19	3	Team meeting, integrated Prince's backend into Prototype master
11/20	1	Added functionality
11/21	4.5	Added add stock, forgot password, reconfigured UI, revised initialization of panels, added logout
11/22	2.5	Sorting out bugs for load and save xml file, delete, add, documentation
11/23	4.5	Debugging, finished, add/delete stock watch list, forgot pass functionality, UI effects
11/24	6	Added onto enum, enum.next(), reconfigured for-loop inside StockUtil to avoid hard code, updating design model, implemented deleteAccount, researching how to add a right click, added in right-click on items in list
11/25	1	Refurbished code to make it cleaner
11/26	1	Added "delete", "rename", functionality to application
11/27	4	Testing Profile in model package, finished Email validation, revised design model, revised create account, login, addWatchList SDs, added comments, added documentation
11/28	2	Colors on UI, fixed GitHub problem
11/29	3	Worked on StockPage
11/30	4	Worked on StockPage, collapsing code, reorganizing code, design

12/01	6	Finished StockPage, documentation, revising design model, finished, Add-Stock, Add-Stock-WatchList SDs
SUBTOTAL:	79	

TOTAL: 20.5 + 18.5 + 79 = 118

TIMECARD

PRINCE KALU		
DATE	HOURS	DESCRIPTION
ITERATION I		
9/20	3	Learning Backend Development & finding the best environment to create on.
9/21	1.5	Begun backend development.
9/21	2	Worked on sketches for iteration one & begun making a slideshow powerpoint.
9/26 - 9/27	7	Added detail to the front end of the home page, connected the front end to back end, researched API's, tested several API's for responsiveness and accuracy & learned JavaScript.
9/28-9/29	3	Iteration one SSD's and Ticket System, helping with domain model, updating GitHub, as well as finishing the slideshow.
SUBTOTAL:	16.5	
ITERATION II		
10/25	.5	Team Meeting
10/25-10/28	3	Java classes
10/25-10/28	1	Updating github
10/25-10/27	2	Researching a new IDE to use
10/28	1.5	Diagrams
10/29	1	Updated project, prez ready
10/30	3	Updated Design Model, updated sequence Diagrams
SUBTOTAL:	12	
ITERATION III		
11/1-11/28	27	Backend Development [StockUtil.java]
11/13-11/30	18	Middleware Development [FlowPanel.java, NewsPanel.java, GraphPanel.java]
11/23-11/30	5	Helping Owen with UI by designing the HomePage [HomeSection.java]
11/25-11/30	23	Base Analysis Algorithm [StockUtil.java]
11/26-11/30	28	Machine Learning Framework [StockUtil.java]
12/01	5	Finalizing project
SUBTOTAL:	106	

TOTAL: 16.5 + 12 + 106 = 134.5



[Conclusion]

This project was a very good learning experience, with respect to programming, analysis and design, and team collaboration. It taught us about ourselves and how much we can handle and sacrifice. In addition, we learned the importance of continuous testing and documentation throughout the software engineering lifecycle.

Overall, We put a lot of time and effort into this project, because at the end of the day what's most important is that we can beat our fists against our chests knowing we made a good service, and we put our sweat and energy into it.



[GitHub Analysis]



[Commits]

- Owen ~ 100
- Prince ~ 60

Total Commits ~ 150 commits



[Additions/Deletions]

Additions

- Owen ~ 10k (mostly UI)
- Prince ~ 6k

Deletions

- Owen ~ 5k
- Prince ~ 3k

NOTE: there are more than a total of 24k additions and deletions due to Logger issues



[Issue Tracking]

- Open 4
- Closed 10

Link to Issue Tracking -> [here](#)



[Packages]

Iteration 1 -> [here](#)

Iteration 2 -> [here](#)

Iteration 3 -> <https://github.com/olmurphy/YoloTrader.github.io/tree/master/YoloTrader/Iteration3>



[Link to Git Repo]

YoloTrader Project host -> [Prince and Owen](#)