King Saud University

College Of Computer and Information Sciences

Information Systems Department

TADAWUL TECH TRADER

Name	ID	Email
Abdulmohsen Alqahtani	442103210	442103210@student.ksu.edu.sa
Osama Alharbi	442103126	442103126@student.ksu.edu.sa
Raed Alamjed	442102093	442102093@student.ksu.edu.sa

Supervised by: Dr. Mohammed alnuem

Table of Contents

Table Of Figures

Table Of Tables

Abstract

Stock markets play a crucial role in the economy, serving as a platform for companies to raise capital and investors to participate in the growth and profitability of these companies. The Saudi Stock Exchange (TASI) is no exception, contributing significantly to the development of the Saudi Arabian economy. However, navigating the complexities of the stock market and making informed trading decisions can be challenging, particularly for individual investors who may lack the time, expertise, or resources to effectively analyze market trends and identify profitable opportunities.

Automated trading systems have emerged as a powerful tool to address these challenges, leveraging advanced algorithms and computational power to analyze vast amounts of market data, identify patterns, and execute trades with speed and accuracy. These systems have the potential to level the playing field for individual investors, providing them with access to sophisticated trading strategies and tools that were previously available only to institutional investors.

"Tadawul Tech Trader" is a cutting-edge automated trading system designed specifically for the Saudi Stock Exchange (TASI). By harnessing the power of artificial intelligence (AI) and machine learning algorithms, the system analyzes market trends, price patterns, and other relevant data to identify profitable trading opportunities. The platform incorporates risk management strategies to minimize potential losses and protect clients' investment portfolios, while offering customization options to cater to individual investment goals and risk tolerance.

The primary function of "Tadawul Tech Trader" is to predict stock prices and automatically execute trades based on generated insights. Additionally, the system offers valuable guidance to users by identifying trending stocks and providing recommendations on whether a particular stock is worth buying or selling at a given time. Efficiently execute trades, and provide personalized investment suggestions, "Tadawul Tech Trader" aims to empower both novice and experienced investors, offering a competitive edge in the Saudi stock market.

This report delves into the potential of "Tadawul Tech Trader" to revolutionize the way investors approach trading in the Saudi Stock Exchange, exploring how the system's advanced capabilities and user-centric features can contribute to improved efficiency, risk management, and potential profitability. By combining cutting-edge technology with a deep understanding of the Saudi market's unique characteristics, "Tadawul Tech Trader" represents a significant step forward in the realm of automated trading, ultimately contributing to the overall development and sophistication of the Saudi Arabian economy.

Chapter one

Introduction

The Saudi Stock Exchange (TASI) is the largest stock market in the Middle East and has played a pivotal role in the economic development of Saudi Arabia. With a market capitalization of over $3.2 trillion as of 2024, TASI provides a platform for Saudi companies to raise capital and for investors to participate in the growth of the Saudi economy.

However, trading on the stock market can be a complex and time-consuming endeavor, particularly for individual investors who may not have the same resources or expertise as institutional investors. Analyzing vast amounts of market data, identifying trends, and making trading decisions can be overwhelming for the average retail investor.

This is where automated trading systems like "Tadawul Tech Trader" come in. By leveraging cutting-edge technologies such as artificial intelligence (AI) and machine learning (ML), these systems can process massive volumes of data, spot patterns and generate trading signals in real-time. Automated trading has the potential to democratize access to sophisticated trading strategies and level the playing field between retail and institutional investors.

What sets our project apart is that it has been developed specifically for the Saudi stock market, taking into account the unique characteristics, dynamics and regulations of TASI. Rather than being a generic, one-size-fits-all solution, our system is tailored to navigate the specific challenges and opportunities of the Saudi market.

The core of "Tadawul Tech Trader" is a powerful algorithmic trading engine driven by advanced machine learning models. These models are rigorously trained on extensive historical TASI data, allowing them to recognize complex patterns and predict future price movements with a high degree of accuracy. By continuously analyzing a wide range of data points relevant to the Saudi market and economy, the system is able to generate reliable 'buy', 'sell', or 'hold' signals that are optimized for the local context.

But our solution goes beyond just generating trading signals. The system is designed to be a comprehensive tool that empowers users to make more informed investment decisions. By identifying trending stocks, providing in-depth market analysis and offering personalized portfolio recommendations, "Tadawul Tech Trader" aims to be a one-stop solution for navigating the Saudi stock market.

Risk mitigation is a core focus of our system's design. "Tadawul Tech Trader" employs robust risk management strategies to protect clients' investments and limit potential downside. The platform allows users to set custom risk tolerances and investment parameters, ensuring that the system's trades align with each individual's unique financial goals and risk appetite.

Looking ahead, our vision is to directly integrate "Tadawul Tech Trader" with the TASI trading platform, enabling real-time data analysis and trade execution. This will allow users to fully capitalize on the speed and efficiency advantages of automated trading.

In the following sections, we will take a deep dive into the technical architecture of "Tadawul Tech Trader", exploring how our AI algorithms, risk management protocols, and user-centric design come together to create a powerful tool for the Saudi market. We will also examine the broader implications and potential impacts of our system on the Saudi financial ecosystem.

Problem statement

The Saudi Stock Exchange (TASI) presents a unique set of challenges for investors and traders seeking to capitalize on the market's opportunities. One of the primary issues is the lack of accessible and reliable tools for automating trading strategies tailored specifically to the Saudi market. Many existing trading platforms and algorithms are designed for more developed markets and may not take into account the distinct characteristics and dynamics of the Saudi stock market, such as the significant influence of oil prices on the economy and stock market performance.

Investors in the Saudi market often face barriers to entry, including limited access to market data, inadequate technical analysis tools, and a lack of understanding of the market's unique behavior. This can lead to suboptimal trading decisions, missed opportunities, and increased exposure to risk. The time-consuming nature of manual trading and the need for constant market monitoring can also be overwhelming, particularly for novice investors or those with limited resources.

The Saudi stock market is influenced by various factors, including oil prices, government policies, market regulations, investor sentiment, and cultural dynamics. These factors create unique challenges and opportunities that may not be adequately addressed by generic trading systems. The absence of a trading solution that takes these specific characteristics into account can limit the effectiveness and profitability of automated trading strategies in the Saudi market.

"Tadawul Tech Trader" aims to address these problems by providing a comprehensive, user-friendly platform that empowers investors to customize and test automated trading strategies tailored specifically to the Saudi stock market. By leveraging advanced machine learning techniques and a rich dataset of historical TASI data.

The platform seeks to democratize access to sophisticated trading tools and insights, allowing both novice and experienced investors to benefit from the power of automated trading. By offering a simulated environment for backtesting and optimization, "Tadawul Tech Trader" aims to build confidence in the system's performance and help users make informed decisions based on data-driven insights and enhancing the potential for profitability in the Saudi stock market.

Objectives and motivations

Objectives:

- Build and train advanced machine learning models capable of analyzing historical TASI data, identifying patterns and trends, and generating accurate and timely trading signals.

- Implement a comprehensive risk management framework to minimize potential losses and protect clients' investment portfolios, while optimizing returns.

- Provide a user-friendly interface that allows clients to customize and backtest trading strategies based on their specific investment goals, risk tolerance, and market preferences.

- Offer valuable insights and recommendations to users by identifying trending stocks and providing data-driven suggestions on buy/sell decisions.

- Continuously monitor and adapt to changes in the Saudi stock market, ensuring the system remains effective and relevant in the face of evolving market conditions.

- Integrate with real-time market data and execute trades seamlessly through an API, once available, to enable clients to fully leverage the system's capabilities in live trading scenarios.

Motivations:

- Bridging the knowledge gap: Empowering investors with tools and insights to make informed trading decisions in the unique Saudi market.
- Harnessing AI potential: Leveraging advanced technologies to revolutionize trading approaches and generate superior returns.
- Promoting financial inclusion: Enabling a wider range of individuals to participate in and benefit from the Saudi stock market.
- Driving innovation: Pushing boundaries and setting new standards for automated trading in the Saudi financial landscape.
- Supporting economic diversification: Contributing to the development financial sector in line with Saudi Arabia's economic goals.
- Building trust in automated trading: Fostering confidence through a transparent, reliable, and user-friendly platform that prioritizes client interests.

Target Audience

"Tadawul Tech Trader" designed to cater to a diverse range of users within the Saudi stock market, including:

Main User:

Individual Investors: The platform empowers individual investors looking to enhance their trading performance and diversify their portfolios. By providing access to advanced trading tools, market insights, and automated trading strategies, we enable individual investors to make informed decisions and potentially improve their investment outcomes.

Potential User:

Brokers and Trading Platforms: Our solution can be seamlessly integrated with existing platforms offered by brokers and financial institutions in Saudi Arabia, such as Al Rajhi Capital, Al Ahli Capital, and other prominent players in the market. By offering our solution as a value-added service or product, these platforms can enhance their offerings and provide their users with advanced trading tools and automated strategies. The integration can help brokers and trading platforms differentiate themselves in the competitive market, attract new clients, and retain existing ones by delivering cutting-edge trading solutions.

Existing Solutions

In the fast-paced world of finance, automated trading systems have become increasingly popular, offering investors powerful tools to navigate the complex stock market. Three notable players in this field are NinjaTrader, MetaTrader, and the Saudi-based startup Auto Trading.

NinjaTrader and MetaTrader are well-established platforms that cater to professional traders, offering a wide range of advanced features and customization options. These platforms enable the integration of AI and machine learning techniques into automated trading strategies, allowing for the development of sophisticated and adaptable systems. However, their extensive functionality and complex user interfaces may be overwhelming for novice investors.

On the other hand, our project aims to provide an accessible and user-friendly automated trading solution tailored specifically for the Saudi stock market. Unlike NinjaTrader and MetaTrader, which do not directly support the Saudi market, our platform is designed to cater to the unique characteristics and requirements of this specific market. By focusing on the needs of a broader range of investors, from beginners to experienced traders, we strive to make automated trading more approachable and efficient for participants in the Saudi stock market.

Auto Trading, a Saudi-based startup, has emerged as a potential competitor in the automated trading landscape. While they claim to offer deep learning software and automated trading tools for the Saudi capital markets, limited information is available regarding their specific services and capabilities due to their unresponsiveness to inquiries. This lack of transparency makes it challenging to assess their suitability for the Saudi market and compare them directly to our project.

Figure 1. MetaTrader

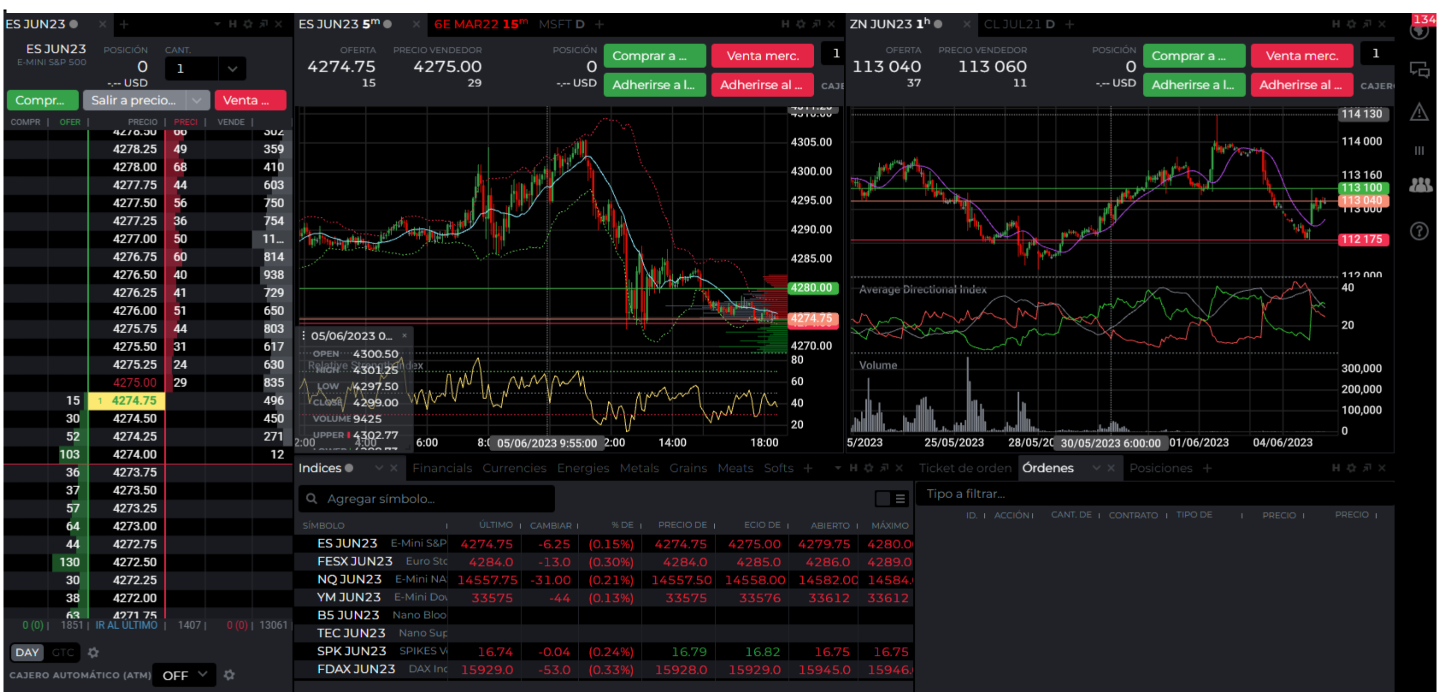

Figure 2. NinjaTrader

Gantt chart

Task Mode	Task Name	Duration	Start	Finish	Predecessors
	Planing	30 days	Thu 2/8/24	Wed 3/20/24	
	Search and Define ideas	15 days	Thu 2/8/24	Wed 2/28/24	
	Describe Ideas	4 days	Fri 3/1/24	Wed 3/6/24	
	Define objectives	5 days	Thu 3/7/24	Wed 3/13/24	3
	Define Problems and motivations	6 days	Wed 3/13/24	Wed 3/20/24	
	Modeling	25 days	Wed 3/20/24	Tue 4/23/24	
	Data Acquistion	15 days	Wed 3/20/24	Tue 4/9/24	
	Trial operation and test model	11 days	Tue 4/9/24	Tue 4/23/24	
	Analysis and designing	10 days	Tue 4/23/24	Mon 5/6/24	
	Define functional and non functional requirements	2 days	Tue 4/23/24	Wed 4/24/24	
	Draw use case diagram	1 day	Wed 4/24/24	Wed 4/24/24	
	write use case description	1 day	Thu 4/25/24	Thu 4/25/24	
	Draw activity diagram	2 days	Fri 4/26/24	Mon 4/29/24	11,12
	Draw ER diagram and relational schema	2 days	Tue 4/30/24	Wed 5/1/24	
	Class digram	2 days	Thu 5/2/24	Fri 5/3/24	
	Draw Sequence diagram	1 day	Mon 5/6/24	Mon 5/6/24	15
	Discussion	4 days	Mon 5/6/24	Thu 5/9/24	
	Reprot	3 days	Mon 5/6/24	Wed 5/8/24	
	Presentation	1 day	Wed 5/8/24	Wed 5/8/24	

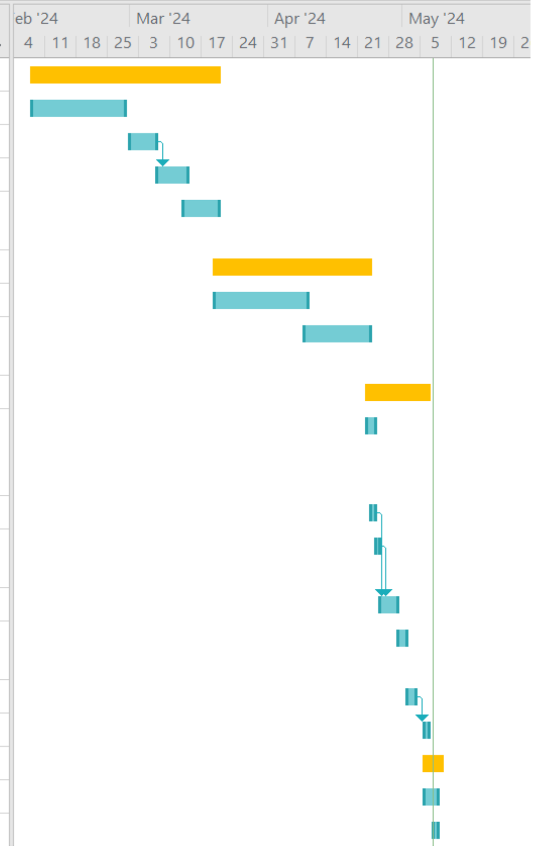

Figure 3. Gantt chart

Methodology

We implemented Waterfall approach is a traditional project management methodology where progress flows steadily downwards, like a waterfall, through distinct phases: requirements, design, implementation, testing, deployment, and maintenance. Each phase relies on the completion of the previous one, making it less adaptable to changes compared to agile methods. Despite its linear nature, Waterfall is valued for its clear structure and documentation, making it suitable for projects with well-defined requirements and minimal changes expected.

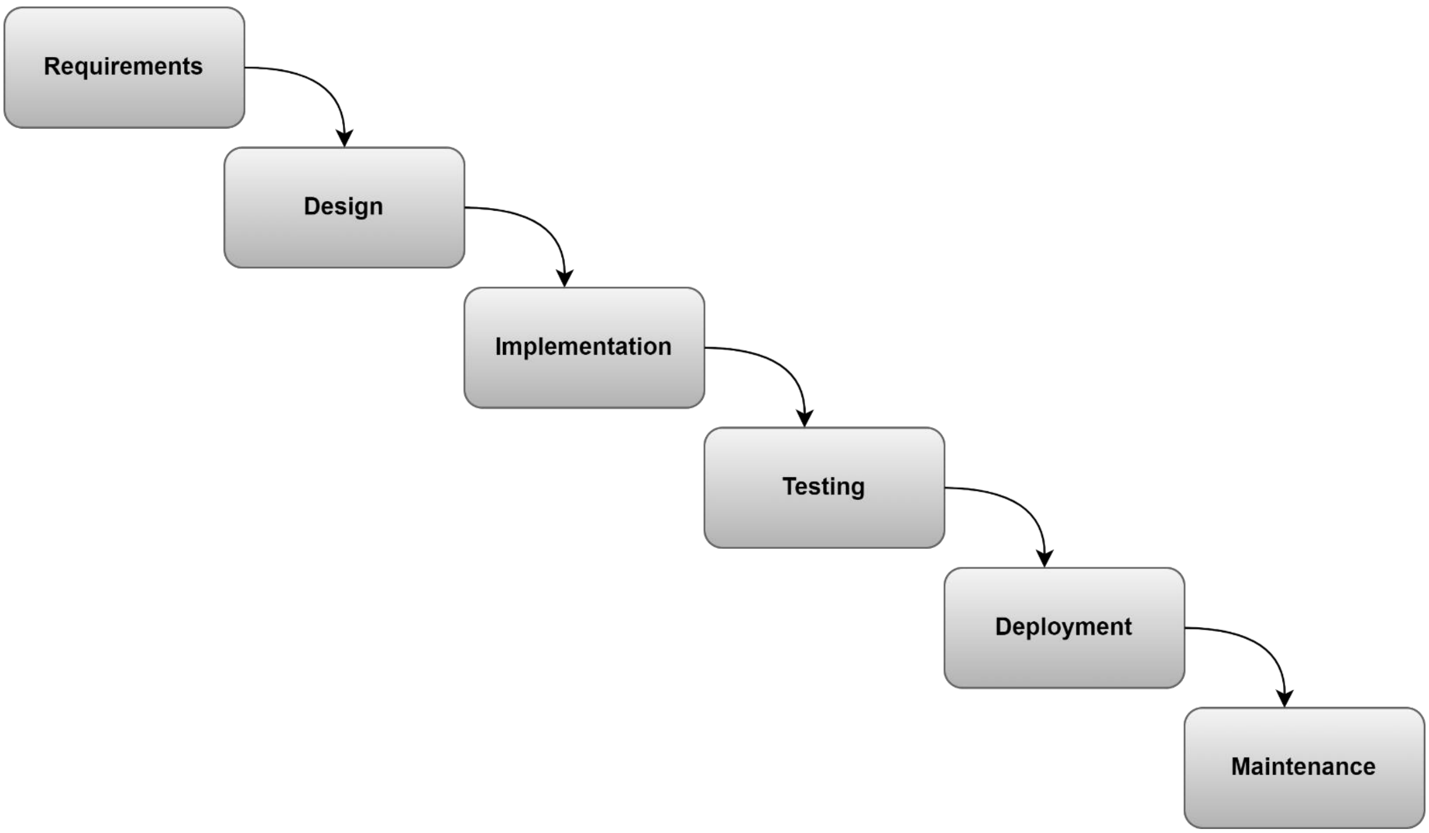

Figure 4. Waterfall

We implemented Waterfall approach because it is particularly suited to our project due to its clear structure and defined stages, which align well with our need for meticulous planning and execution.

Chapter two

Data acquisition and Modeling

Initial proposal

The proposed system architecture for our system integrates various components to enable automated trading in the Saudi stock market. The system relies on historical stock price data, historical oil prices, and historical market news as input data sources. This data undergoes analysis and feature engineering to extract meaningful patterns and indicators.

The processed data is then stored in a database, which serves as a central repository for the machine learning and trend prediction module. This module utilizes advanced algorithms to identify market trends, predict future stock prices.

The decision-making algorithm takes the output from the machine learning and trend prediction module and combines it with predefined rules and risk management strategies to determine the optimal trading actions. This algorithm considers factors such as market conditions, risk tolerance, and investment goals to generate final trading signals, these final signals are then passed to the automation component, which executes the transactions based on the signal.

Throughout the process, the system continuously monitors and analyzes market data, adapting its strategies and decisions based on the latest information.

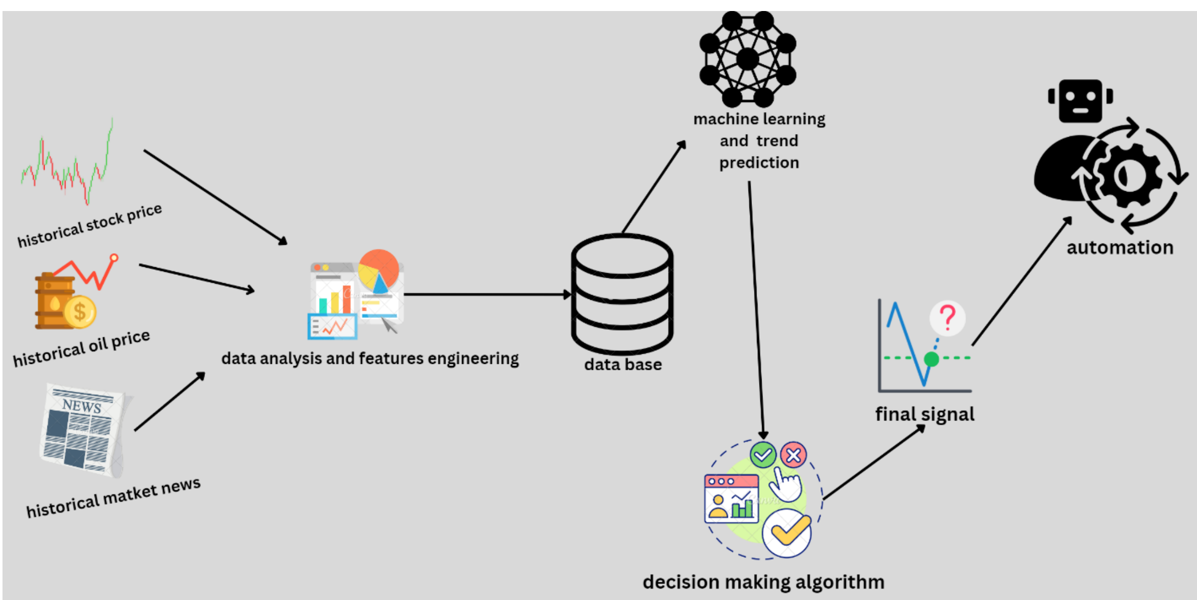

Figure 5. Initial proposal

Data Acquisition

One of the most critical aspects of developing an automated trading system is the acquisition of high-quality and comprehensive data. To ensure the accuracy and reliability of the automated trading decisions, we have sourced data from multiple APIs and platforms, each offering unique insights into the Saudi stock market.

- **Tadawul eReference Data:** Tadawul, the Saudi Stock Exchange, provides a valuable resource in the form of historical stock equity data spanning from 2001 to 2024. This extensive dataset allows us to perform robust backtesting and train sophisticated machine learning models capable of identifying long-term trends and patterns. However, integrating and preprocessing this data to ensure consistency and compatibility with other data sources presents a challenge that requires careful data mapping and normalization, we will use this dataset.

- **Saudi Exchange Stocks Tadawul API:** To obtain real-time information on all companies listed on the Saudi Stock Exchange (TASI), we utilize the Saudi Exchange Stocks Tadawul API. This API provides daily data in a structured format, including essential details such as stock symbol, price, change percentage, volume, and transactions. While the API offers valuable insights, handling the large volume of data and ensuring efficient data retrieval and storage processes is a significant challenge. Implementing robust error handling and data validation mechanisms is crucial to maintain data integrity throughout the system.

- **Yahoo Finance API:** Integrating real-time stock prices is essential for making informed trading decisions. The Yahoo Finance API allows us to retrieve up-to-date information for specific companies, providing a wide range of data points such as price, volume, averages, dividend information, and financial metrics. However, managing API rate limits and handling potential data discrepancies or inconsistencies between different data sources requires careful consideration and implementation of appropriate data harmonization techniques.

- **Argaam Data APIs and Marketaux API for Market News:** To incorporate market sentiment analysis into our system, we leverage the Argaam Data APIs and Marketaux API, which provide access to real-time and historical news articles related to the Saudi stock market. These APIs offer article details such as title, description, keywords, link, language, and published date. Instead of developing our own natural language processing models to analyze the news data, we employ a powerful artificial intelligence language model, such as GPT (Generative Pre-trained Transformer), through an API integration. This approach allows us to leverage state-of-the-art language understanding and generation capabilities to classify and extract meaningful insights from the news articles. By utilizing a pre-trained language model, we can efficiently process the unstructured text data and obtain accurate sentiment analysis and classification results without the need for extensive in-house model development and training.

DATASET:

: Sample of historical Saudi stock exchange data. Stock prices are reported in Saudi Riyal (SAR).

Company name	Ticker ID	Date	Open price	Highest price	Lowest price	Close price	Volume
Saudi Reinsurance Co.	8200	2010/02/10	12.9	12.9	12.85	12.9	5191406
Saudi Reinsurance Co.	8200	2010/02/13	12.9	12.96	12.79	12.96	4714687
Saudi Reinsurance Co.	8200	2010/02/14	12.9	12.96	12.85	12.9	3533019
Alinma Bank	1150	2010/02/10	9.41	9.45	9.37	9.45	401010899
Alinma Bank	1150	2010/02/13	9.41	9.48	9.41	9.45	393167756
Alinma Bank	1150	2010/02/14	9.48	9.56	9.45	9.56	484667229
Saudi Kayan Petrochemical Co.	2350	2010/02/10	17.85	17.85	17.5	17.6	112031139
Saudi Kayan Petrochemical Co.	2350	2010/02/13	17.55	17.9	17.55	17.9	80609654
Saudi Kayan Petrochemical Co.	2350	2010/02/14	17.95	18.1	17.85	18.05	154690207
Saudi Public Transport Co.	4040	2010/02/10	8.25	8.3	8.25	8.25	2867685
Saudi Public Transport Co.	4040	2010/02/13	8.25	8.35	8.25	8.3	8897551
Saudi Public Transport Co.	4040	2010/02/14	8.3	8.35	8.3	8.3	4419915

Data Integration and Preprocessing

Integrating data from multiple sources is a complex task that requires careful data mapping and normalization to ensure consistency and compatibility across the entire dataset. We employ various preprocessing techniques, such as data cleaning, handling missing values, and feature engineering, to prepare the data for analysis and model training. Establishing a robust and scalable data pipeline is essential to handle the volume and variety of data sources effectively.

The set of 17 learning features.

Feature	Description	Definition
Open	Open price	The first traded price at the start trading period
Highest	Highest price	Maximum price traded during the trading period
Lowest	Lowest price	Minimum price traded during the trading period
Change	The difference between the closing price of the current period and the closing price of the previous period.	Change = (Close - Close_lag_1)
%Change	The percentage change in price from the previous period to the current period	$\%Change = \left(\frac{(\square)(\text{Close} - \text{Close\_lag\_1})}{Close\_lag_{-1} * 100}\right)$
Volume	*The total number of shares*	*Sum of the volume of trades during the trading period*
Rolling mean	*The moving average of the closing price over a specified number of periods*	*Mean(Close, n), where n is the number of periods*
Rolling Std	*The rolling standard deviation of the closing price over a specified number of periods*	*Rolling Std = Std(Close, n), where n is the number of periods*
Rolling Min	*The rolling minimum of the closing price over a specified number of periods*	*RollingMin = Min(Close, n), where n is the number of periods*
Rolling Max	*The rolling maximum of the closing price over a specified number of periods*	*Max(Close, n), where n is the number of periods*
MACD	*A trend-following momentum indicator that shows the relationship between two moving averages of a security's price*	*MACD = EMA(12) - EMA(26), where EMA stands for Exponential Moving Averag*
MACDSignal	*The signal line for the MACD indicator, often a 9-day EMA of the MACD*	*EMA(MACD, 9), where EMA stands for Exponential Moving Average*
RSI	*A momentum oscillator that measures the speed and change of price movements*	*RSI = 100 - [100 / (1 + RS)], where RS is the average of x days' up closes / average of x days' down close*
BB_(Upper,Middle ,Lower)	*These are lines plotted two standard deviations away from a simple moving average (SMA) of a security's price*	*BB_Upper = SMA + 2 * StdDev* *BB_Middle = SMA,* *BB_Lower = SMA - 2 * StdDev*
Close_lag_n	*this is lagged version of the closing price, indicating the closing price of the previous days (1, 2, 3, 5, and 7 days ago, respectively), where n is the number of days*	

Modeling

In the pursuit of developing an accurate and efficient automated stock trading system, we explored various machine learning models to analyze and predict stock prices effectively. This section delves into the four models investigated in our initial testing: Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Random Forest Regressor, and Support Vector Machine (SVM). By evaluating the performance and suitability of these models, we aim to identify the most promising approach for our dataset and research objectives.

Initial Model Testing

- **Long Short-Term Memory (LSTM):** We began our exploration with the LSTM model, a variant of recurrent neural networks renowned for its ability to capture long-term dependencies in time series data. The LSTM architecture consisted of two LSTM layers with 128 and 64 units, respectively, followed by two dense layers. The model was trained using a set of feature columns, including percentage change, rolling mean, MACD, and RSI, which encapsulate key technical indicators and market dynamics.

  Upon evaluation, the LSTM model achieved a Mean Squared Error (MSE) of 0.0001, and R-squared (R2) value of 0.9136 . These metrics indicate the model's ability to learn and predict stock price patterns with reasonable accuracy.

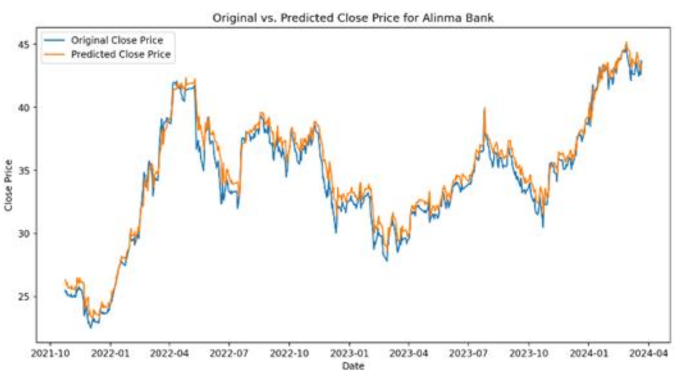

Figure 7. Alinma Bank LSTM

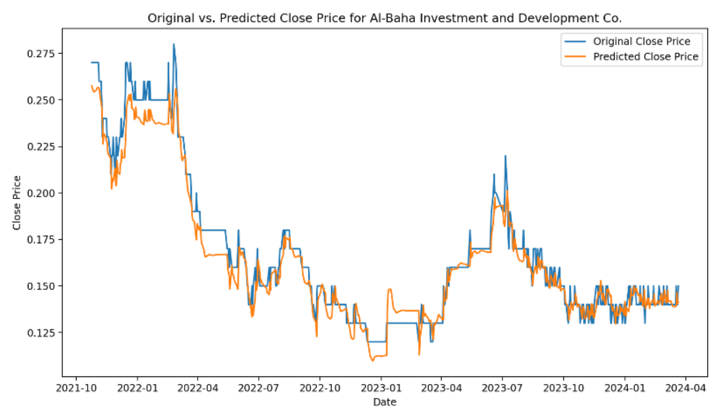

Figure 6. Al-Baha Co LSTM

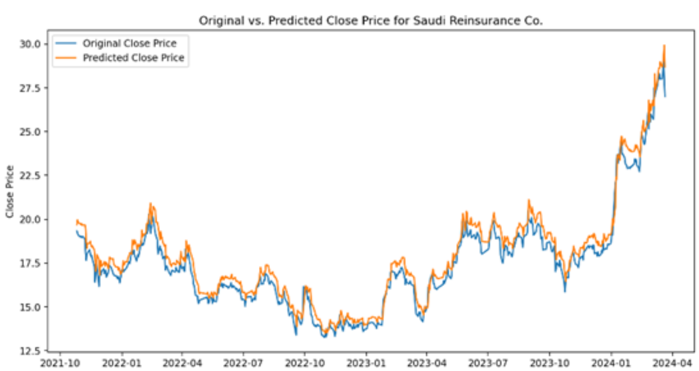

Figure 9. Saudi Reisurance Co LSTM

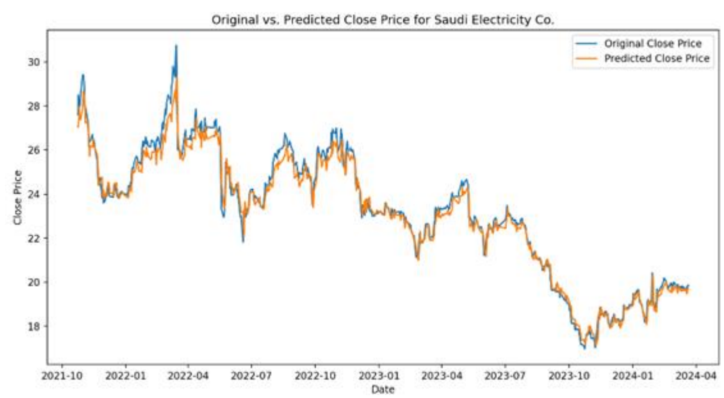

Figure 8. Saudi Electricty Co LSTM

- **Recurrent Neural Network (RNN):** As an alternative approach, we investigated the RNN model, another type of recurrent neural network. The RNN architecture comprised two SimpleRNN layers with 128 and 64 units, respectively, followed by two dense layers. The input features for the RNN model mirrored those used in the LSTM model to ensure a fair comparison.

  The RNN model yielded an MSE of 3.3439 , and R2 value of 0.8662,. While the RNN model demonstrated predictive capabilities, its performance slightly lagged behind the LSTM model, potentially due to its limited ability to capture long-term dependencies effectively.

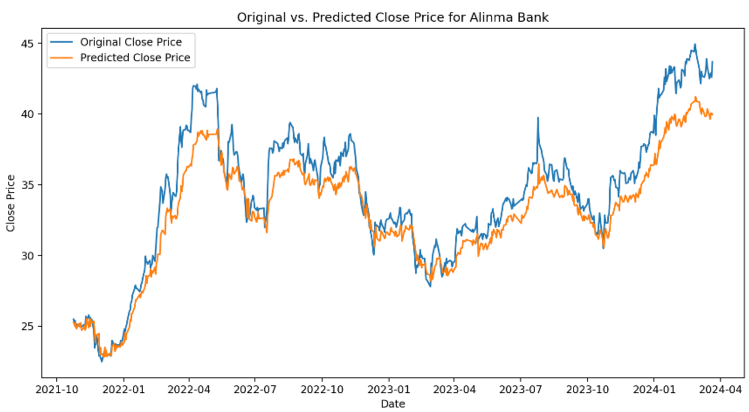

Figure 11. Alinma Bank RNN

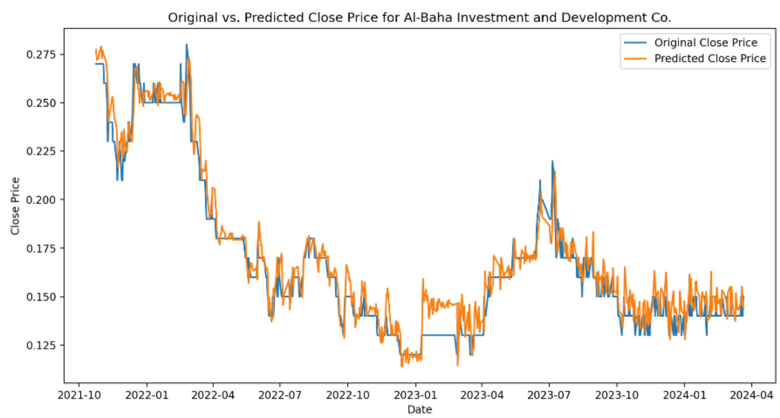

Figure 10. Al-Baha Co RNN

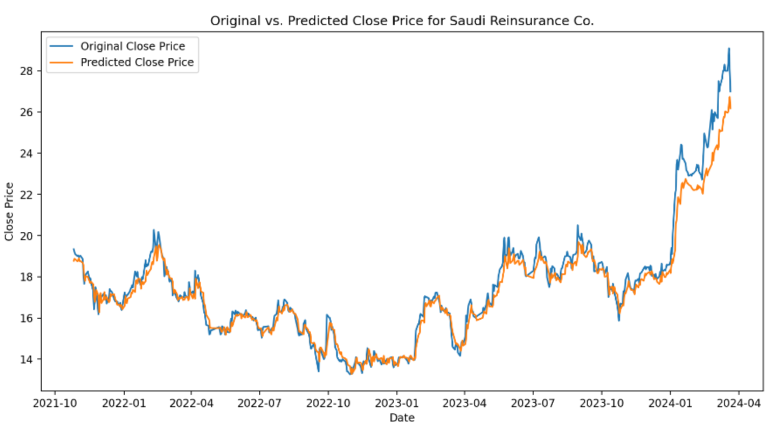

Figure 13. Saudi Reinsurance Co RNN

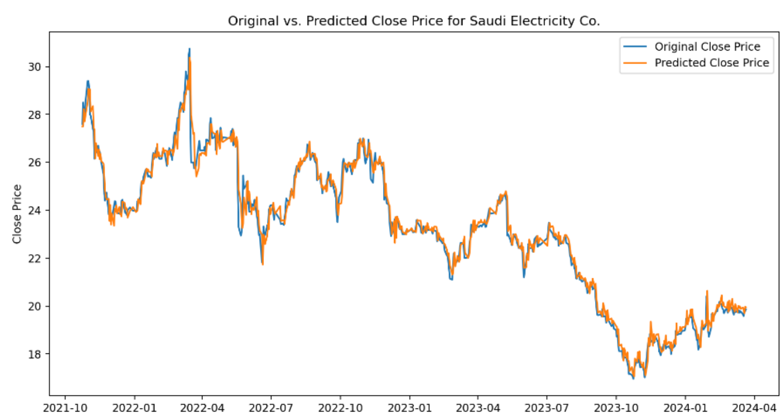

Figure 12. Saudi Electricity RNN

- **Random Forest Regressor:** Shifting our focus to a different modeling paradigm, we employed the Random Forest Regressor, an ensemble learning method that combines multiple decision trees to make robust predictions. The Random Forest model was constructed with 150 estimators, balancing model complexity and computational efficiency.

  The Random Forest Regressor achieved an MSE of 4.9750, and R2 value of 0.3415.
  Although the model captured the general price trend, it exhibited limitations in capturing intricate.

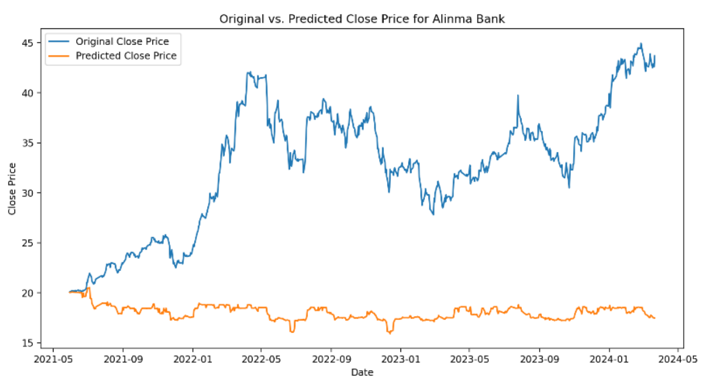

Figure 15. Alinma Bank RF

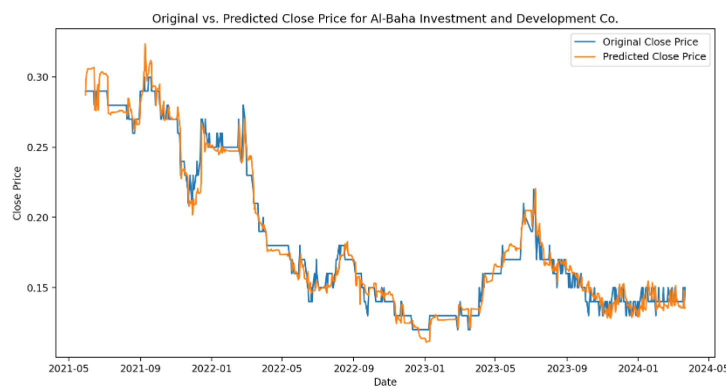

Figure 14. Al-Baha Co RF

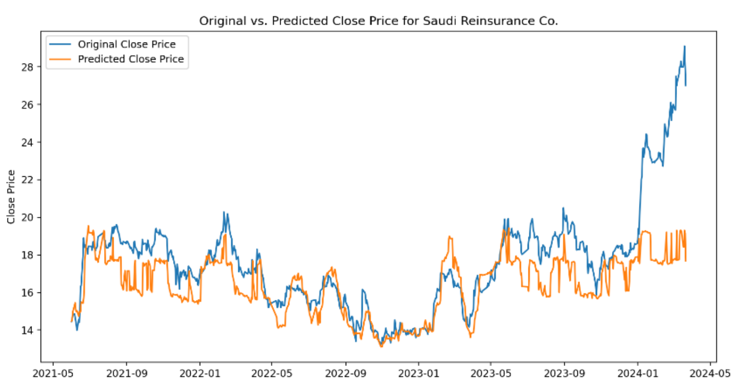

Figure 17. Saudi Reinsurance Co RF

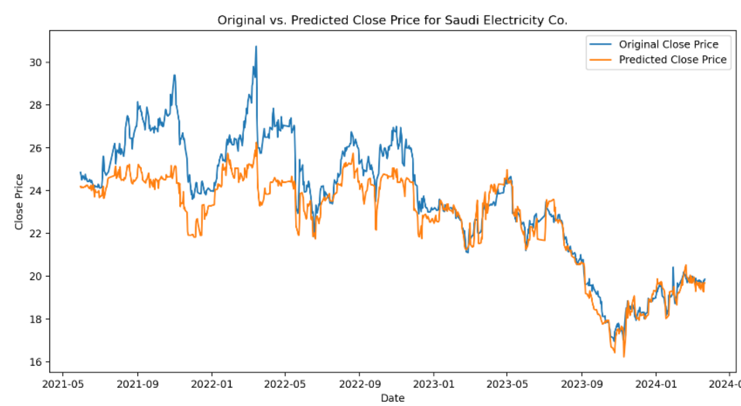

Figure 16. Saudi Electricity Co RF

- **Support Vector Machine (SVM):** We also explored the Support Vector Machine (SVM) model, specifically the Support Vector Regression (SVR) variant, for predicting stock prices. SVR is known for its ability to handle non-linear relationships and provide robust predictions.

  Upon evaluation, the SVM model achieved an MSE of 0.3519, and R2 value of 0.7618.

Figure 19. Alinma Bank SVM

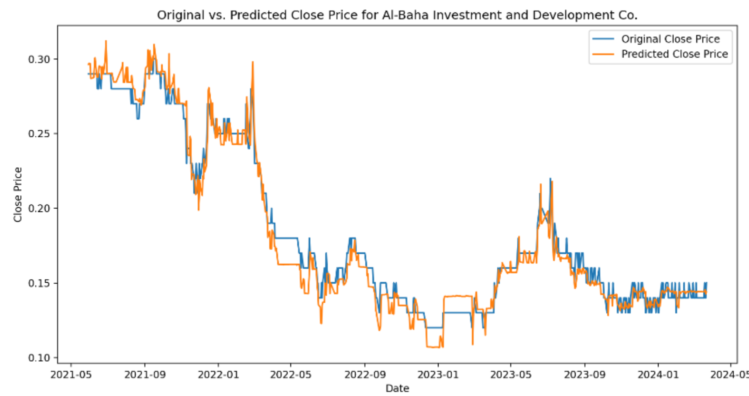

Figure 18. Al-Baha Co SVM

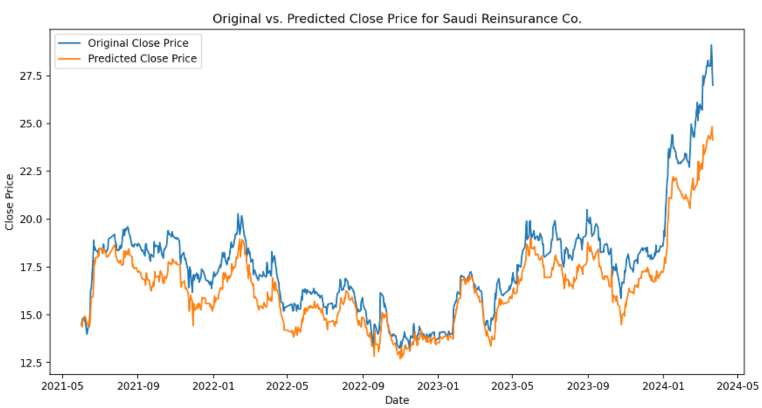

Figure 21. Saudi Reinsurance Co SVM

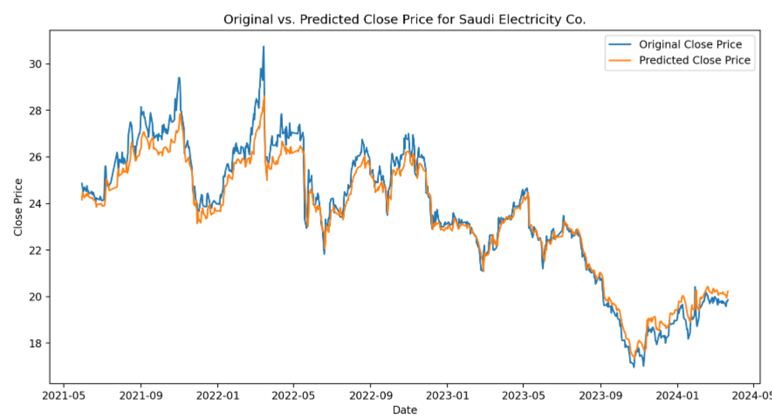

Figure 20. Saudi Electricity Co SVM

Model Selection and Conclusion

Based on the initial testing results, we concluded that the LSTM model emerges as the most promising approach for our dataset. Its superior and consistent performance across different stocks, as evidenced by the lowest MSE, highest R2, and highest accuracy, can be attributed to its ability to capture long-term dependencies and effectively handle the temporal nature of stock price data.

The RNN model, while demonstrating comparable performance to the LSTM, also exhibited relatively stable results across various stocks. Although it slightly underperformed compared to the LSTM, the RNN model's ability to capture dependencies in the data makes it a viable alternative for our stock price prediction task.

The SVM and Random Forest models showed varying performance depending on the specific stock being analyzed. For some stocks, these models achieved relatively good results, indicating their potential to capture certain patterns and relationships in the data. However, for other stocks, their performance was subpar compared to the LSTM and RNN models.

The inconsistent performance of the SVM and Random Forest models across different stocks suggests that they may be more sensitive to the specific characteristics and dynamics of each stock. The SVM model's ability to handle non-linear relationships and the Random Forest model's ensemble nature could potentially explain their good performance on certain stocks. However, their varying results indicate that they may not be as robust and generalizable as the LSTM and RNN models for our overall dataset.

It is important to note that the performance of the SVM and Random Forest models should not be entirely disregarded. While they may not be the top performers across all stocks, their successful application in specific instances highlights the potential for further exploration and improvement. Future research could investigate the factors contributing to their success on certain stocks and explore ways to enhance their performance and generalization ability.

Comparative Analysis of LSTM and RNN Performance

An interesting observation from our experiments was that the RNN model exhibited better performance compared to the LSTM model for stocks with high volatility. This finding prompted us to investigate the underlying reasons behind this behavior.

Several factors could contribute to the RNN's superior performance in highly volatile stocks.

Firstly, RNNs, with their simpler architecture, may be more sensitive to short-term patterns and fluctuations, which are prevalent in highly volatile stocks. The gating mechanism in LSTMs, designed to capture long-term dependencies, may inadvertently filter out some of these short-term patterns, leading to slightly inferior performance.

Additionally, the complexity of LSTMs, with their greater number of parameters, may make them more prone to overfitting, especially when dealing with highly volatile data. Overfitting occurs when a model learns to fit the noise and idiosyncrasies of the training data too closely, resulting in poor generalization to unseen data.

Furthermore, the effectiveness of LSTMs in capturing long-term dependencies relies on the availability of sufficient training data. If the dataset for highly volatile stocks is relatively small or lacks adequate historical data, the LSTM model may struggle to learn the short-term patterns effectively. In such cases, the simpler RNN architecture may be more robust and able to capture the essential patterns with limited data.

It is important to note that these explanations are hypothetical and would require further investigation and empirical validation. The performance of different models can vary depending on the specific dataset, preprocessing techniques, and model architectures used.

Future Directions and Recommendations While the LSTM model has shown promising results, it is essential to acknowledge that there is room for further improvement. We recommend the following future directions to enhance the robustness and accuracy of our automated stock trading system:

1. **Ensemble Modeling:** We plan to explore the combination of RNN and LSTM models, and potentially other models, using ensemble techniques such as voting or stacking. Ensemble modeling can leverage the strengths of different models and provide more robust and accurate predictions.

2. **Hyperparameter Tuning:** Fine-tuning the hyperparameters of the LSTM and RNN models, such as the number of layers, units, and learning rate, could potentially improve their performance and generalization ability.

3. **Feature Engineering:** Exploring advanced feature engineering techniques, such as sentiment analysis, may provide additional valuable inputs to the models and improve their predictive capabilities.

In short, this section demonstrates the potential of machine learning techniques, particularly LSTM and RNN models, in developing an automated stock trading system. By combining multiple models, optimizing hyperparameters, and continuously adapting our approach, we can work towards building a robust and profitable trading system.

Risk Management Strategies

Effective risk management is a crucial aspect of our system to ensure the protection of clients' investments and the long-term sustainability of the system. The following initial risk management strategies are being considered for implementation:

1. **Position Sizing Strategies:**
   - Fixed Ratio Position Sizing: Allocating a fixed percentage of the portfolio to each trade.
   - Percent Risk Position Sizing: Sizing positions based on a predetermined maximum risk per trade.
2. **Stop-Loss Strategies:**
   - Fixed Stop-Loss: Setting a predetermined price level to exit a losing trade.
   - Trailing Stop-Loss: Dynamically adjusting the stop-loss level as the trade moves favorably.
3. **Portfolio Diversification Techniques:**
   - Sector Diversification: Diversifying across various industry sectors to mitigate sector-specific risks.
4. **Risk Monitoring and Alerting:**
   - Market Monitoring: Continuously monitoring market conditions, news, and events that may impact trades.
5. **Backtesting and Simulations:**
   - Historical Backtesting: Testing trading strategies on historical market data to evaluate performance and risks.
6. **Continuous Improvement:**
   - Model Validation and Recalibration: Regularly validating and recalibrating the system's models and algorithms.
   - Strategy Performance Evaluation: Continuously evaluating the performance of trading strategies and making necessary adjustments.
   - Ongoing Research and Development: Investing in research and development to incorporate the latest advancements in risk management.

These initial risk management strategies will be further refined and optimized based on extensive backtesting, simulations, and ongoing monitoring of market conditions and system performance. The goal is to implement a comprehensive and adaptive risk management framework that balances risk mitigation with the pursuit of potential returns, ensuring the long-term success and reliability of our system.

Chapter three

Data analysis and design

Functional requirements

- **Registration:** The system enables users and administrators to create an account, log in, and for the users they can edit their profiles.
- **Select Account:** The system enables admins to select user accounts and manage it by deleting or editing accounts.
- **Manage Portfolio:** The system enables users and administrators to create and delete portfolios, and for administrators they can change portfolio balance.
- **View Reports:** The users can view the report of their bots transactions.
- **View Stocks:** The system enables users to show their list of stocks.
- **Make Transaction:** The system allows the bot to buy and sell stocks.
- **View Profiles:** The system allows users to view their profiles.
- **Customize Bot:** The system allows users to customize their bots by selecting trading strategies, desired stocks, and risk level etc.
- **Get Suggestions:** The user can ask about a specific stock and the bot will generate suggestions for him.
- **Predict Price:** The bot can predict price using machine learning models.
- **Backtesting:** The user can test his bot in a simulation environment.

Non-functional requirements

- ***Availability:*** The system is available 24/7.
- ***Usability:*** The system shall be easy, ease of use and user friendly.
- ***Reliability:*** the system should be reliable and be able to perform the required functions correctly and consistently over time.
- ***Responsiveness:*** The system should perform the orders in real time with minimal latency.
- ***Security:*** The system should protect User's information from unauthorized access.

Use Case Diagram

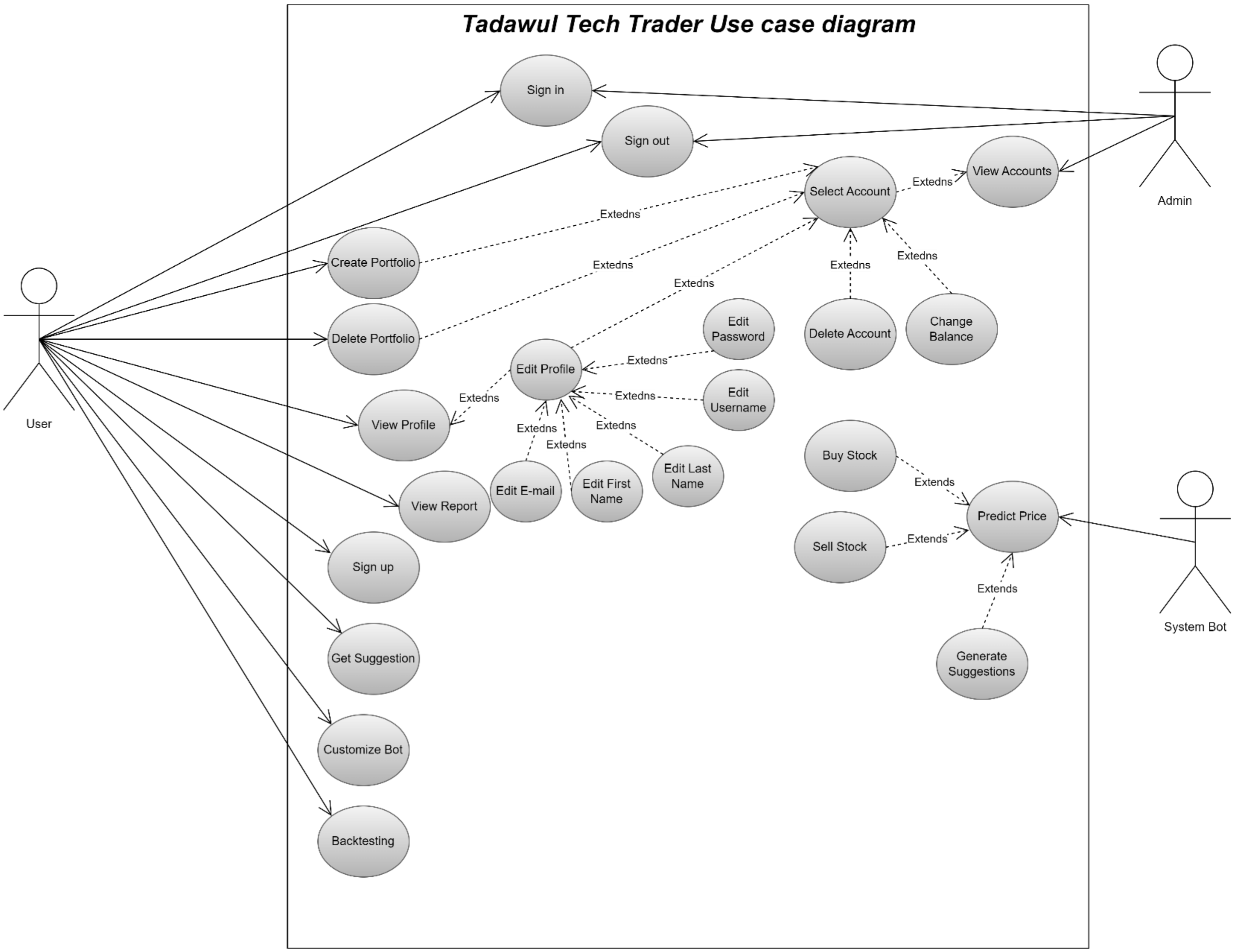

Figure 22. Use Case Diagram

Use Case Description

Use Case Name	Get suggestions	
Scenario	User enters selected stock to get a suggestion	
Triggering Event	User requests a stock suggestion.	
Brief description	Get suggestion from system bot to a selected stock.	
Actor	User	
Preconditions	User select a stock to predict	
Postconditions	User receives suggestion	
Flow of activities	Actor	System
	1. User clicks on get suggestion button. 2. User will fill in a stock name	1.1 System display box to fill in stock name. 2.1 System will send the stock name to the model. 2.2 System will receive the suggestion and show it to the user
Exceptions		

Table 1. Get Suggestions

Use Case Name	Sign in	
Scenario	User/Admin enters credentials to log in his account	
Triggering Event	User/Admin wants to enter his account	
Brief description	User/Admin will have access to his account.	
Actor	User, Admin	
Preconditions	User/Admin enter the web page	
Postconditions	User/Admin will be in his account	
Flow of activities	Actor	System
	1. User/Admin Enters website 2. User/Admin fills required credentials	1.1 System shows required credentials 2.1 System will check if credentials match an account 2.2 System gives access if credentials match
Exceptions		

Table 2. Sign in

Use Case Name	View Report	
Scenario	User wants a report about a portfolio	
Triggering Event	User wants a	
Brief description	User ask System to generate a report about a portfolio	
Actor	User	
Preconditions	Selected portfolio	
Postconditions	Report generated	
Flow of activities	Actor	System
	1. User enters a portfolio and clicks view report button	1.1 System generate a report for the selected portfolio. 1.2 return the report to the User
Exceptions		

Table 3. View Report

Use Case Name	Customize Bot	
Scenario	User wants to change bot settings	
Triggering Event	User customize trading bot	
Brief description	User change bot settings and trading strategies	
Actor	User	
Preconditions	user has the necessary permissions and credentials to customize bot settings.	
Postconditions	Bot settings updated	
Flow of activities	Actor	System
	1. User click on customize button 2. change to the wanted settings then click save 3. confirm new settings	1.1 display all bot settings 2.1 shows new settings with confirmation massage 3.1 update settings
Exceptions		

Table 4. Customize Bot

<table>
<tr><td>Use Case Name</td><td colspan="2">Sign up</td></tr>
<tr><td>Scenario</td><td colspan="2">The user wants to make a new account.</td></tr>
<tr><td>Triggering Event</td><td colspan="2">-</td></tr>
<tr><td>Brief description</td><td colspan="2">Sign up to website by inserting full name, username, email, and password</td></tr>
<tr><td>Actor</td><td colspan="2">User</td></tr>
<tr><td>Preconditions</td><td colspan="2">The user must have a new email.</td></tr>
<tr><td>Postconditions</td><td colspan="2">A new account registered to the database.</td></tr>
<tr><td rowspan="2">Flow of activities</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Enter his details.
2. Confirm email.</td><td>1.1 Check if email and username doesn’t exist already.
1.2 Validate information.
1.3 Send a confirmation email.
2.1 Create account.
2.2 Open account.</td></tr>
<tr><td>Exceptions</td><td colspan="2">1.1 Account already exists.
1.2 Information not valid.</td></tr>
</table>

Table 5. Sign up

Use Case Name	Predict Price	
Scenario	Bot wants to Predict a price	
Triggering Event	Bot scheduled prediction process	
Brief description	Bot sends stock details to system for price prediction and signal	
Actor	Bot	
Preconditions	Created portfolio	
Postconditions	Price Predicted	
Flow of activities	Actor	System
	1. Send stock name and symbol	1.1 predict stock price 1.2 generate decision signal 1.3 send predicted price and signal
Exceptions		

Table 6. Predict Price

Use Case Name	Create Portfolio	
Scenario	User wants to create a portfolio	
Triggering Event		
Brief description	User enters a set of stocks that he wants the bot to track and trade and set a balance to the bot	
Actor	User	
Preconditions	User must login an account	
Postconditions	Portfolio created	
Flow of activities	Actor	System
	1. Click on create portfolio button 2. select a time and set a balance for portfolio	1.1 ask user for needed information 2.1 create a portfolio with selected
Exceptions		

Table 7. Create Portfolio

Use Case Name	Change Balance	
Scenario	Admin wants to change portfolio balance	
Triggering Event	User have technical issues	
Brief description	Admin changes balance available to the bot in portfolio	
Actor	Admin	
Preconditions	Selected a portfolio	
Postconditions	Balance updated	
Flow of activities	Actor	System
	1. Admin select a portfolio and press update balance Button 2. confirm the new balance	1.1 System shows old and new balance and asks for confirmation 2.1 update portfolio balance
Exceptions		

Table 8. Change Balance

Use Case Name	Buy stock	
Scenario	Bot wants to buy a stock	
Triggering Event	Buy signal received	
Brief description	Bot buys a stock	
Actor	Bot	
Preconditions	Buy signal received	
Postconditions	Bought stock	
Flow of activities	Actor	System
	1. Send stock name, symbol, and quantity	1.1 Send a buy order with received information
Exceptions		

Table 9. Buy Stock

Use Case Name	Sell stock	
Scenario	Bot wants to Sell a stock	
Triggering Event	Buy signal received	
Brief description	Bot buys a stock	
Actor	Bot	
Preconditions	Sell signal received	
Postconditions	Sold stock	
Flow of activities	Actor	System
	1. Send stock name, symbol, and quantity	1.1 Send a sell order with received information
Exceptions		

Table 10. Sell Stock

Use Case Name	Backtesting	
Scenario	Test Bot performance	
Triggering Event	User press Backtesting button	
Brief description	User test bot performance in simulated environment before live trading	
Actor	User	
Preconditions	Portfolio must be created	
Postconditions	Backtest report generated	
Flow of activities	Actor	System
	1. User enters Bot id and period of time	1.1 run a test on the period selected 1.2 generate a report of the performance 1.3 display a report for User
Exceptions		

Table 11. Backtesting

Activity Diagram

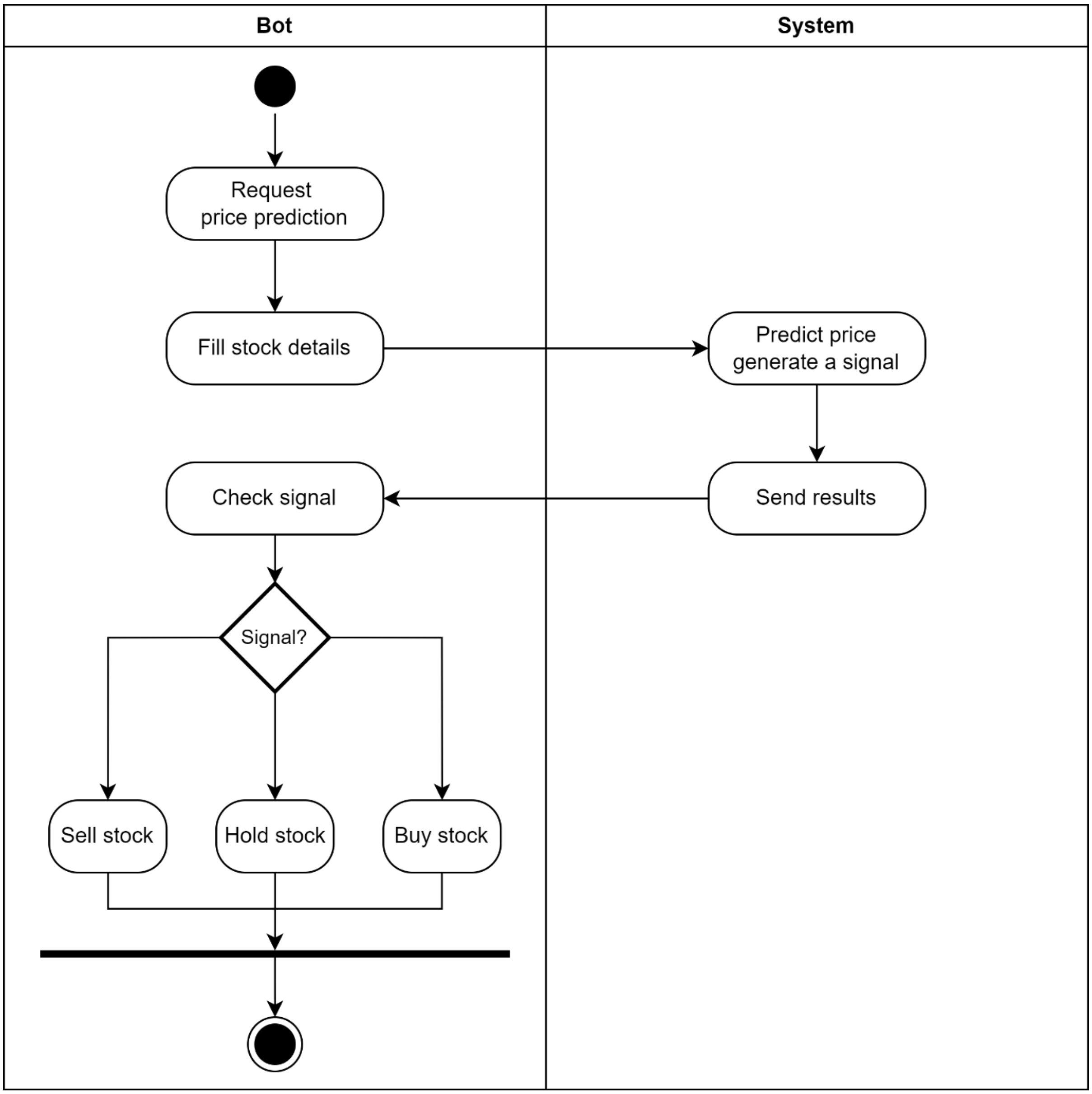

Figure 23. Predict price

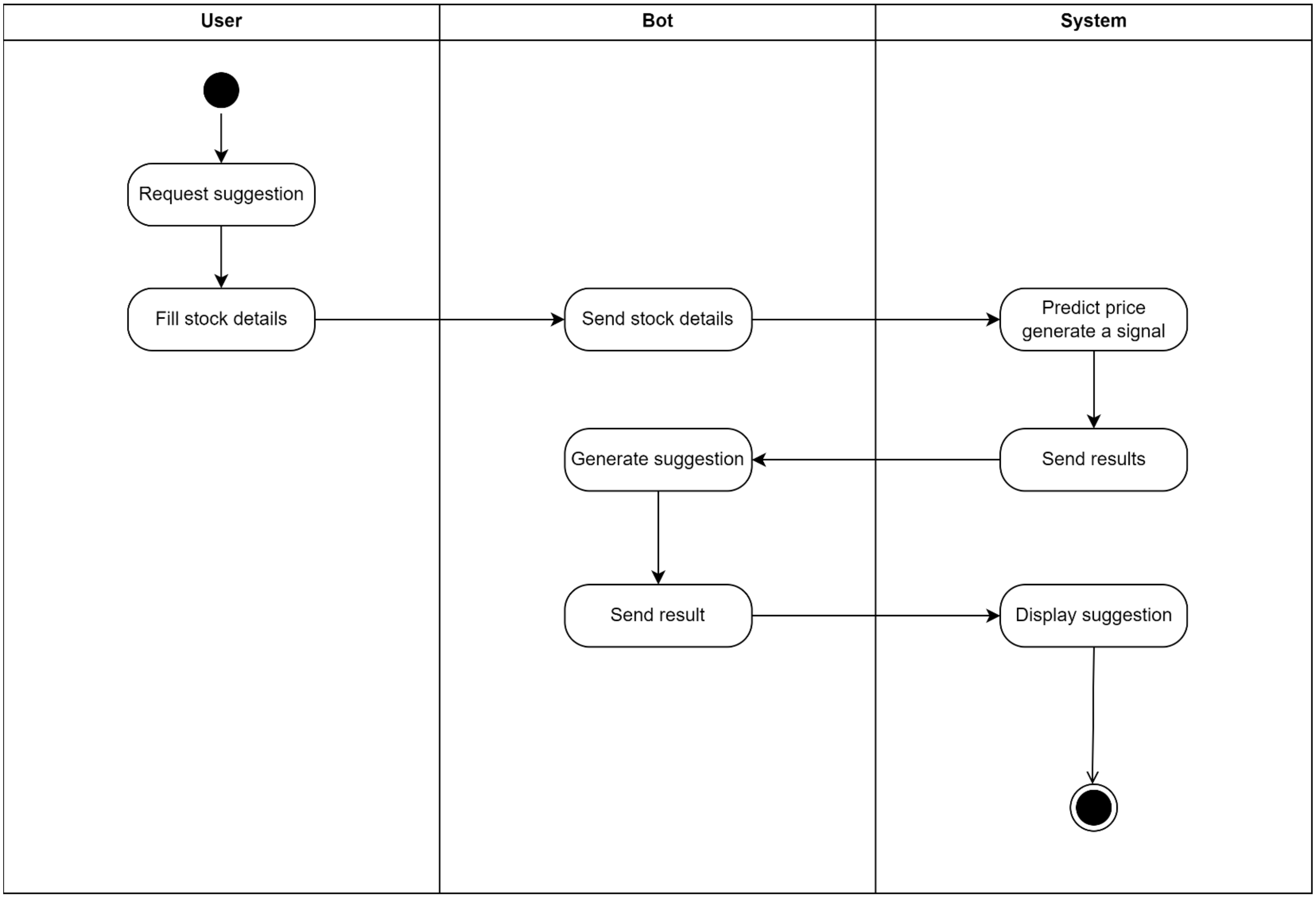

Figure 24. Get suggestions

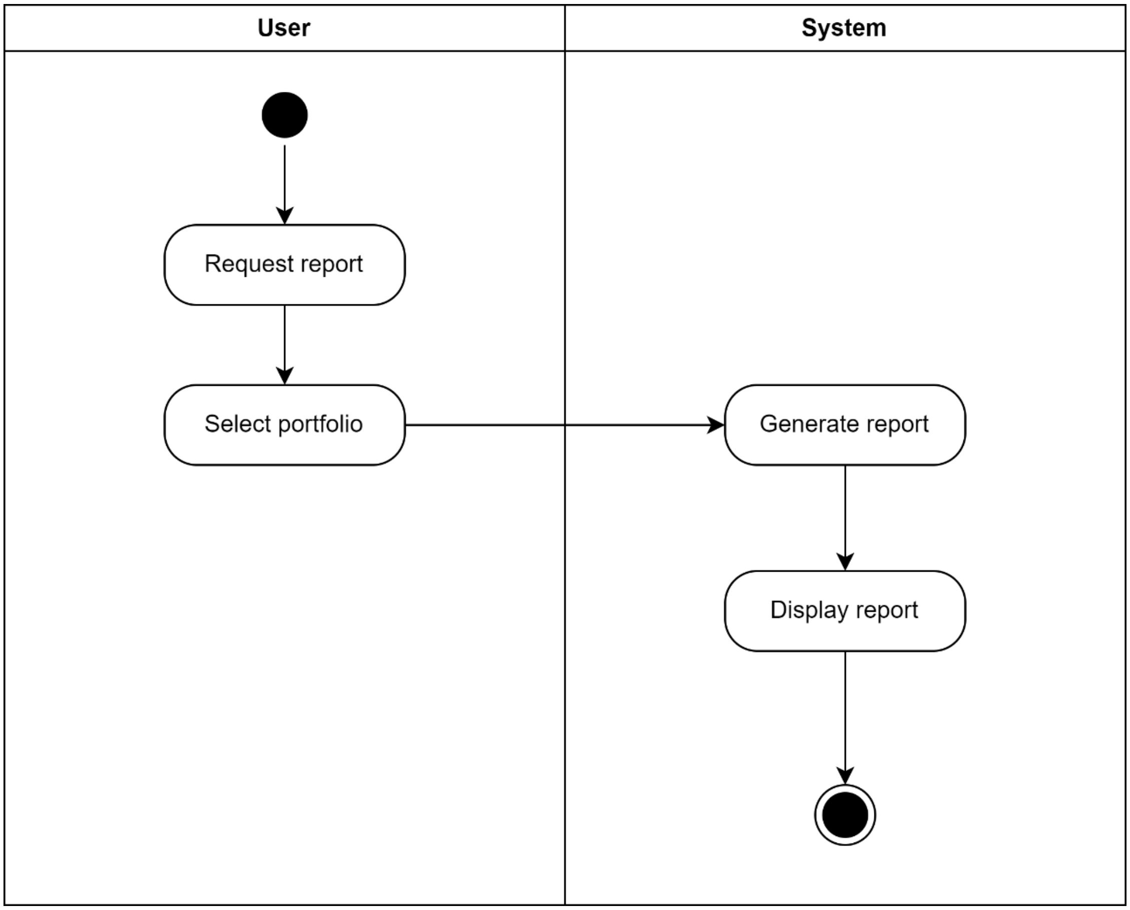

Figure 25. View report

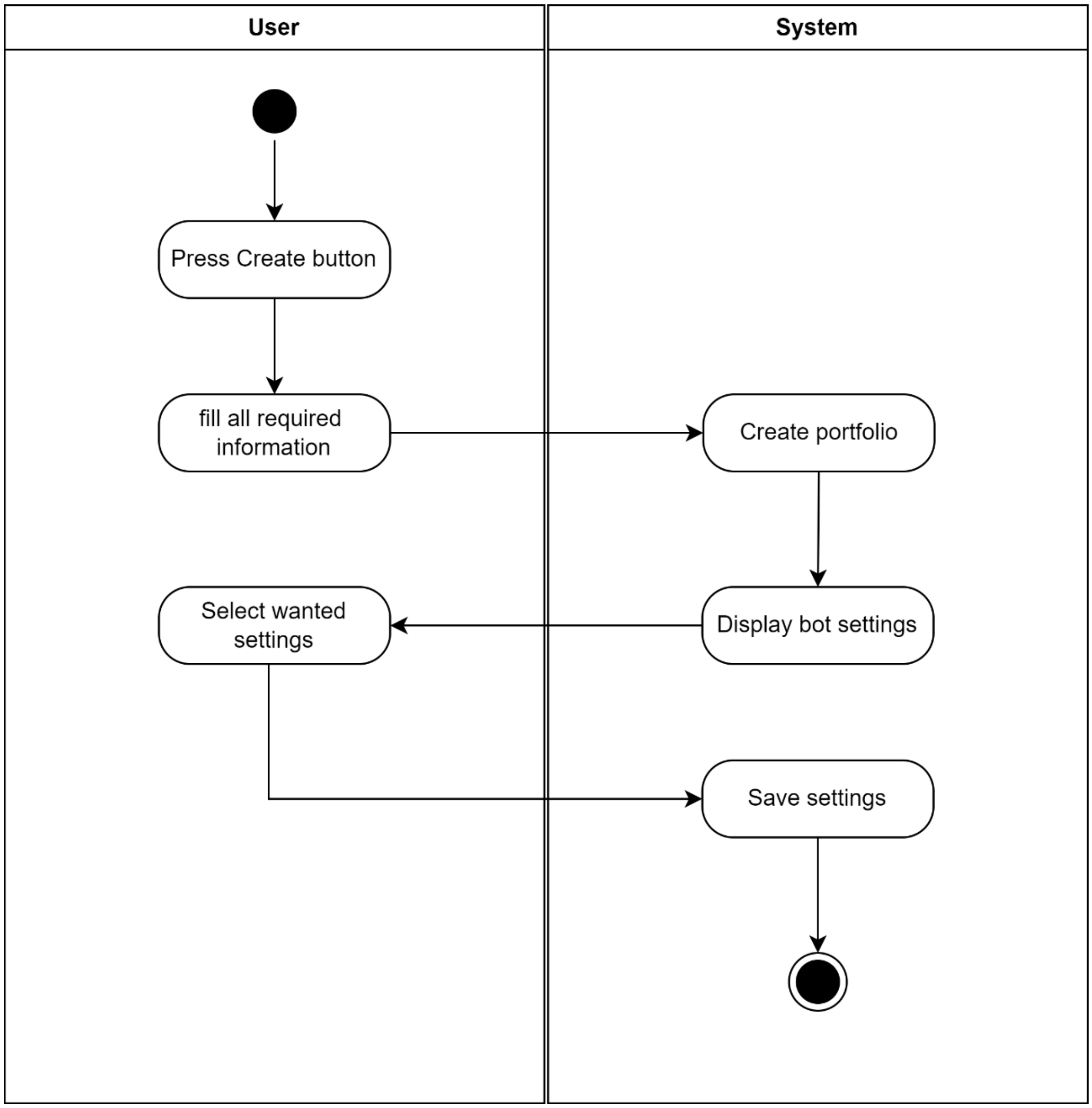

Figure 26. Create portfolio

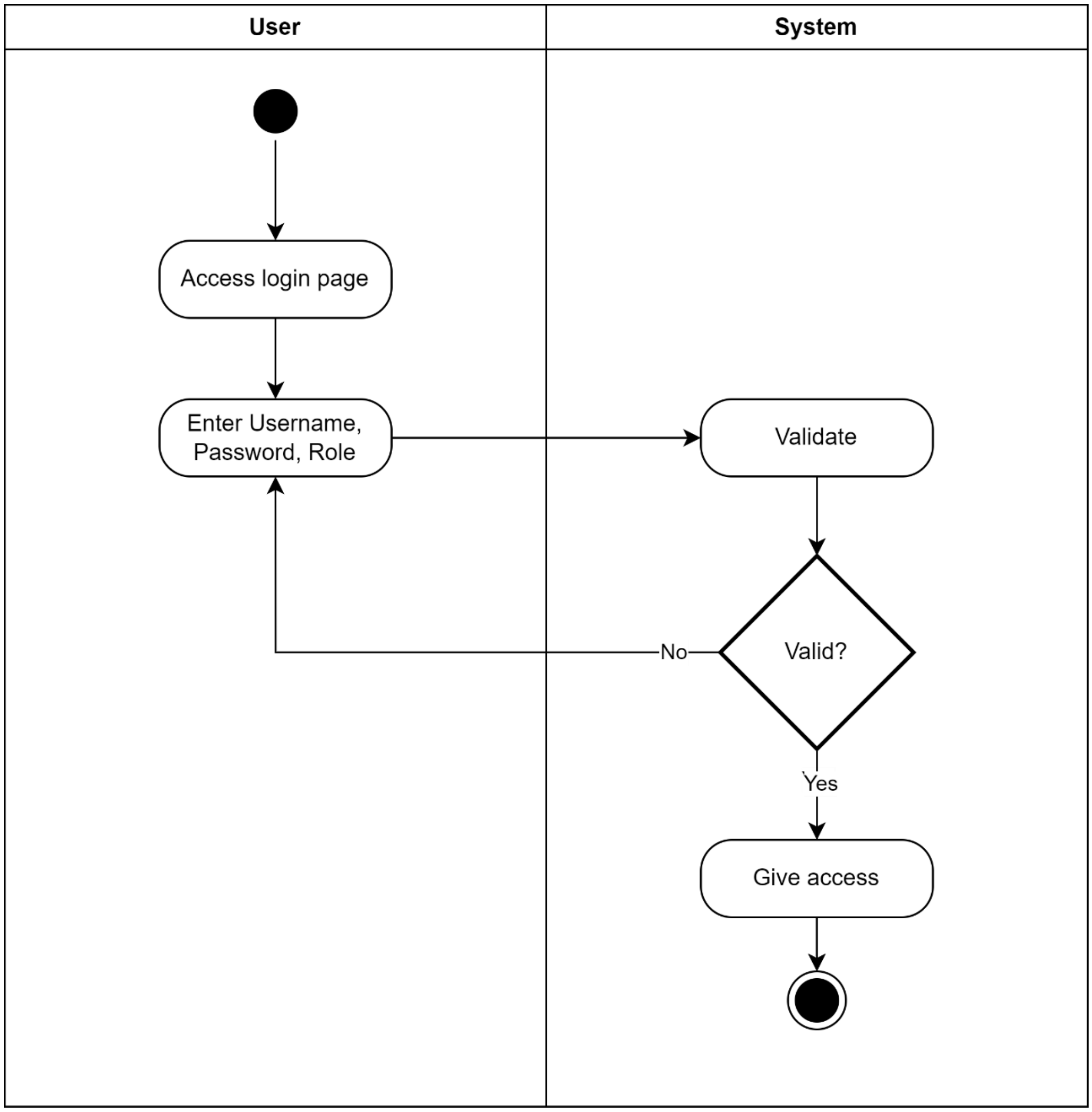

Figure 27. Sign in

Class Diagram

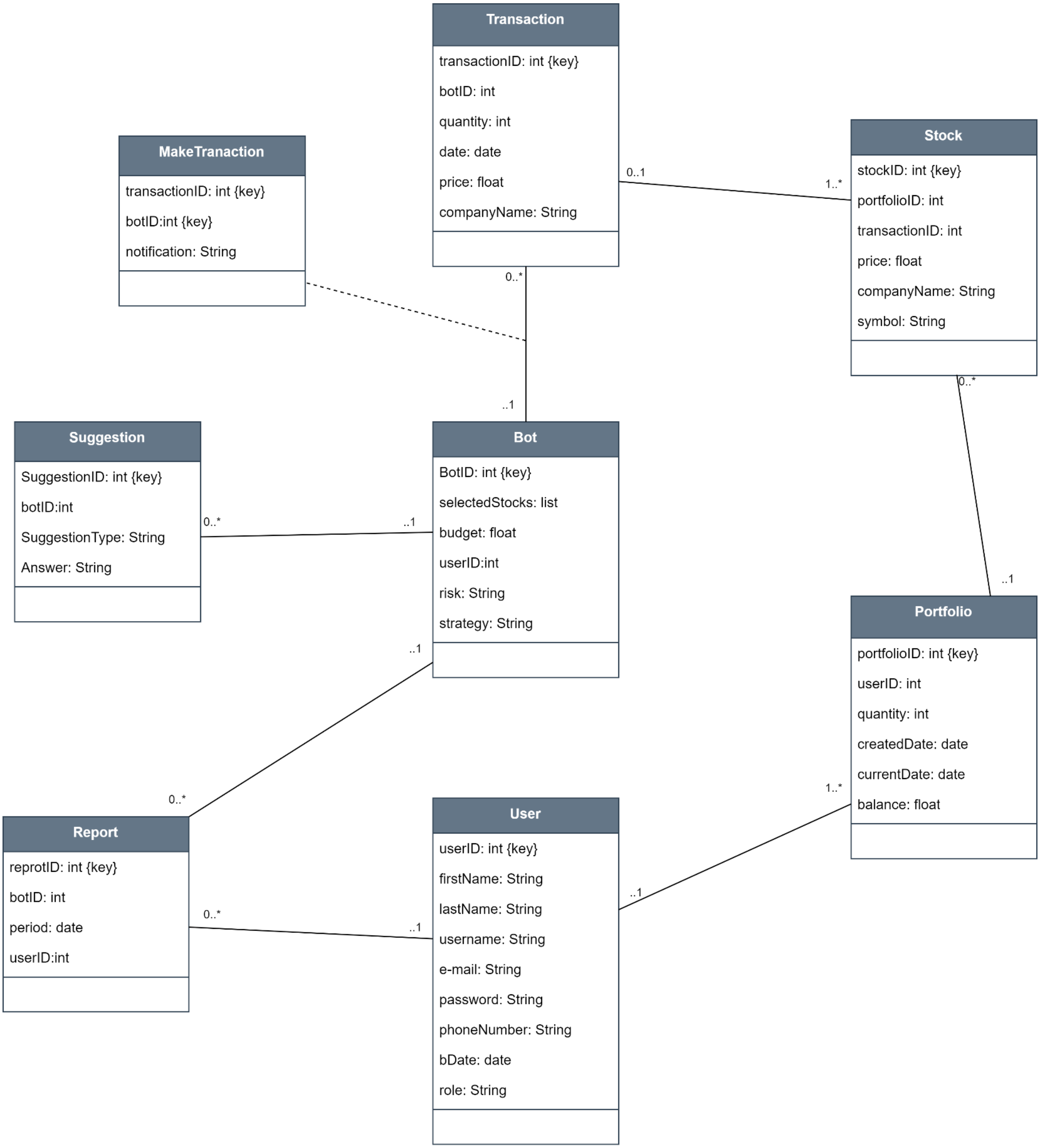

Figure 28. Class Diagram

ER Diagram

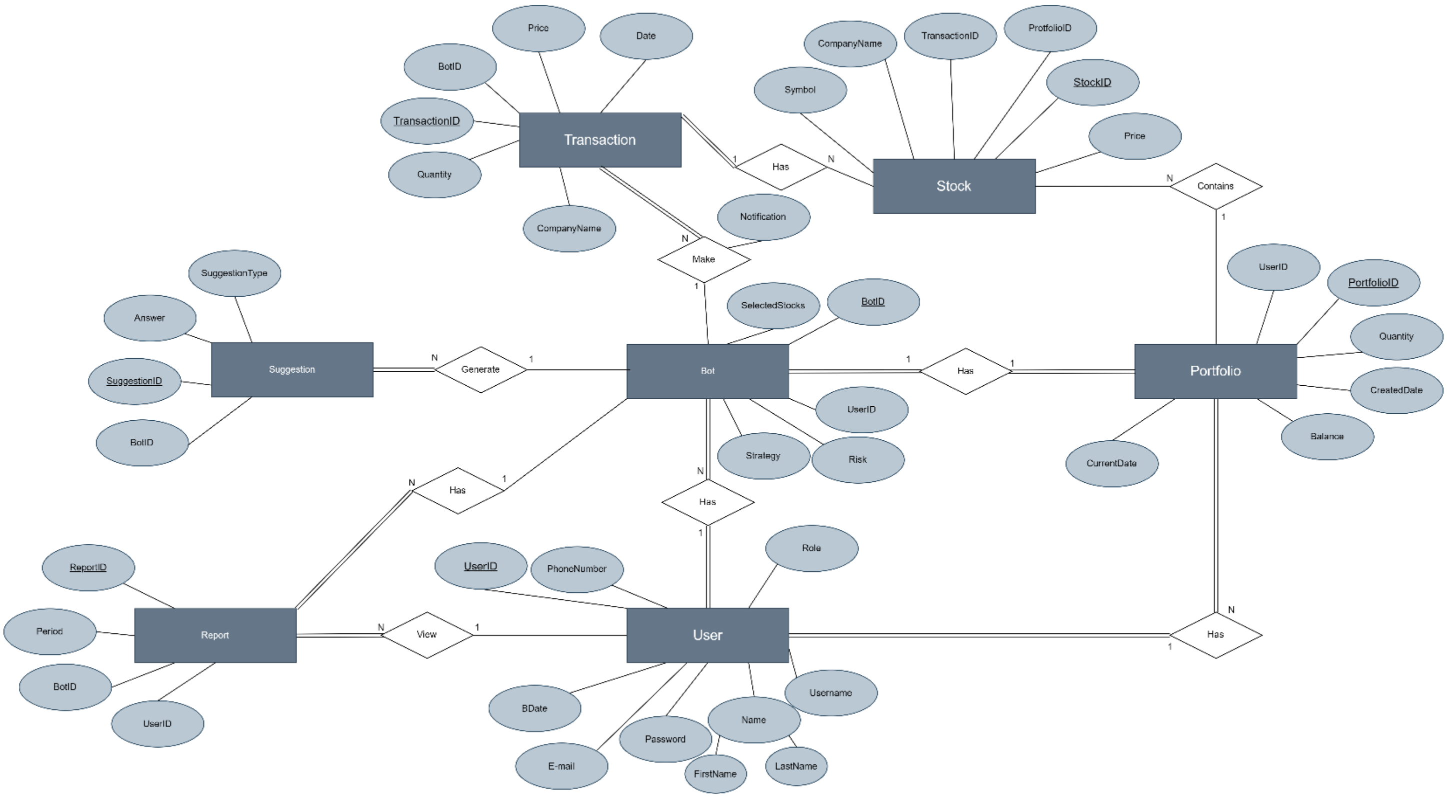

Figure 29. ER Diagram

Relational Schema

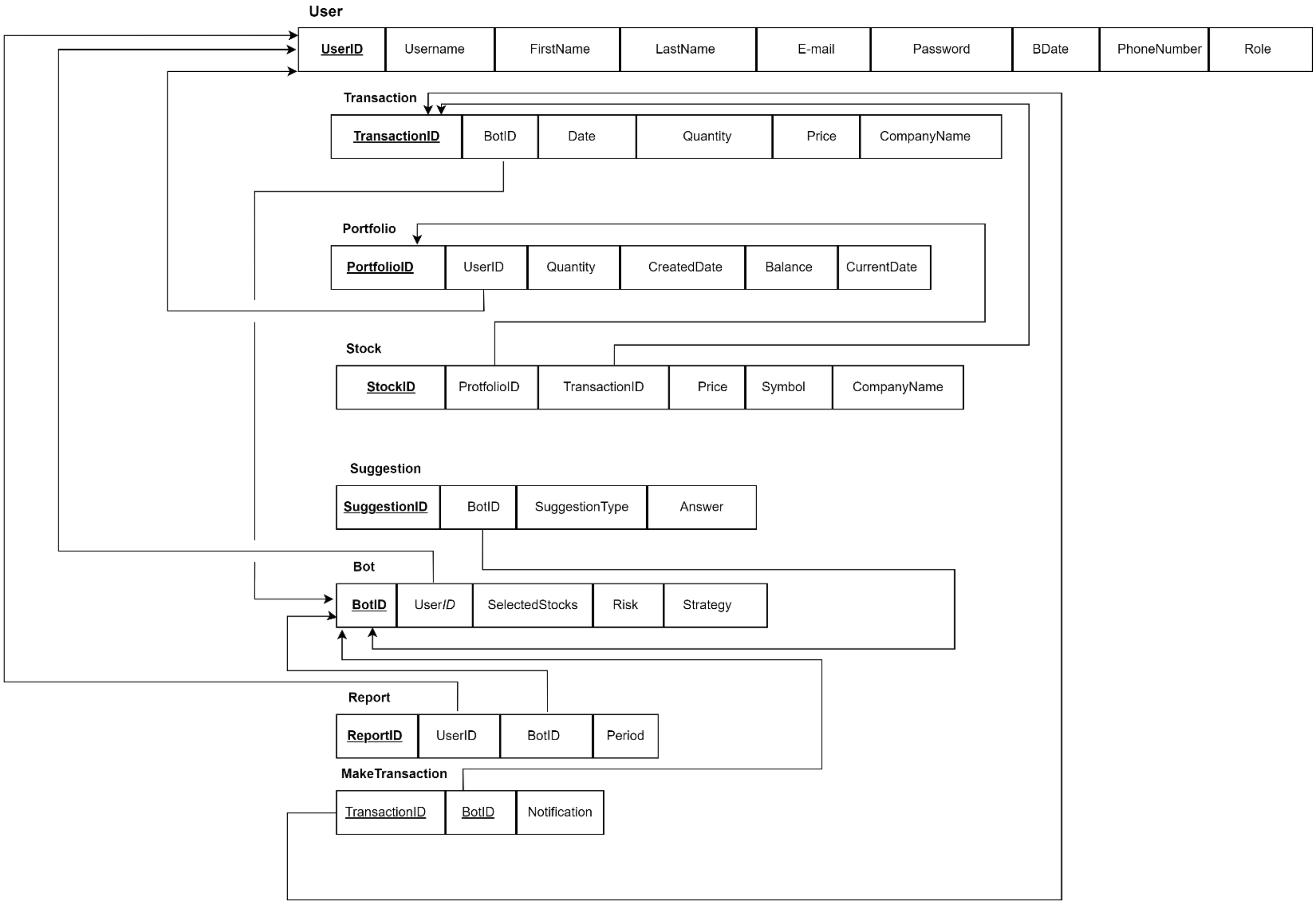

Figure 30. Relational Schema

First-Cut Class Diagram

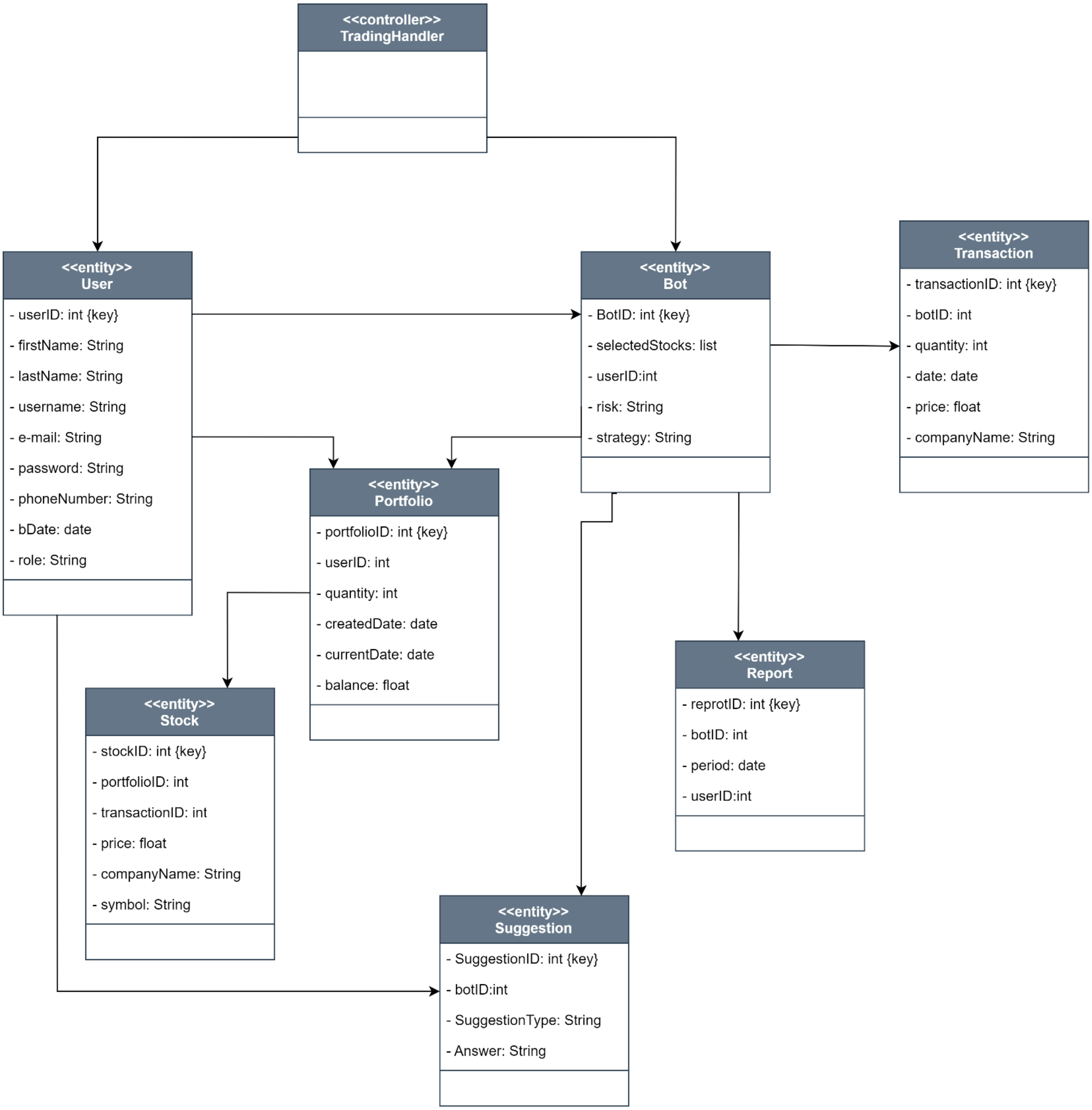

Figure 31. First-Cut Class Diagram

Sequence Diagram

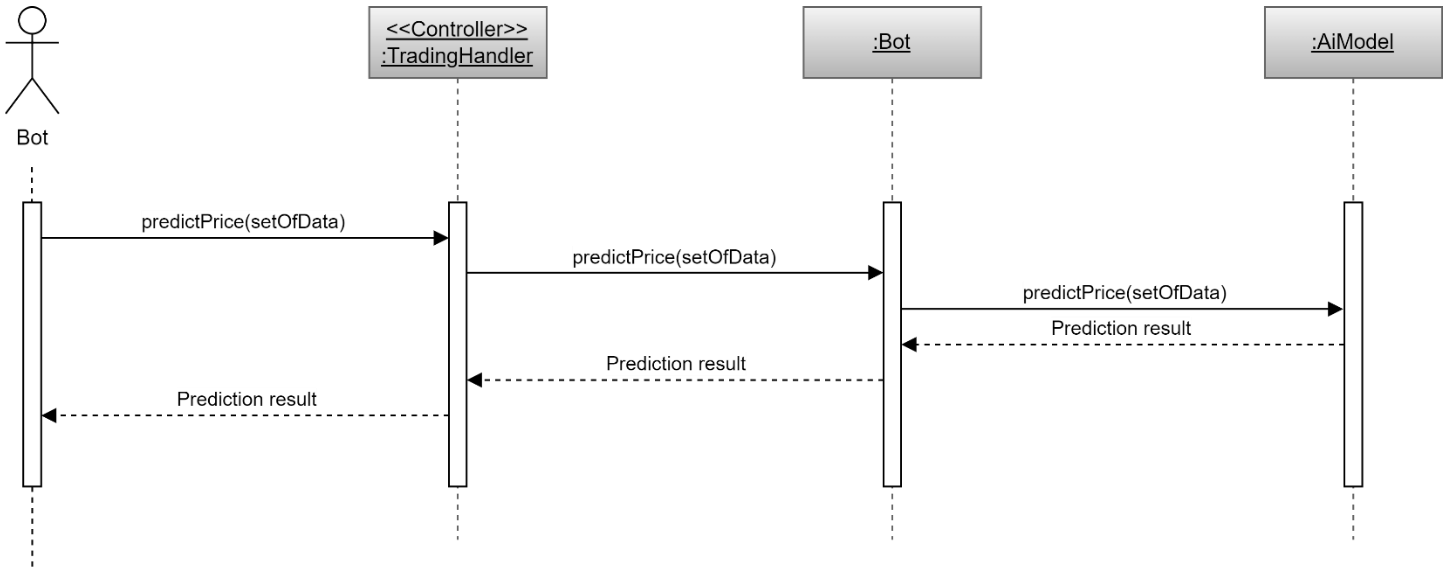

Figure 32. Predict Price Sequence Diagram

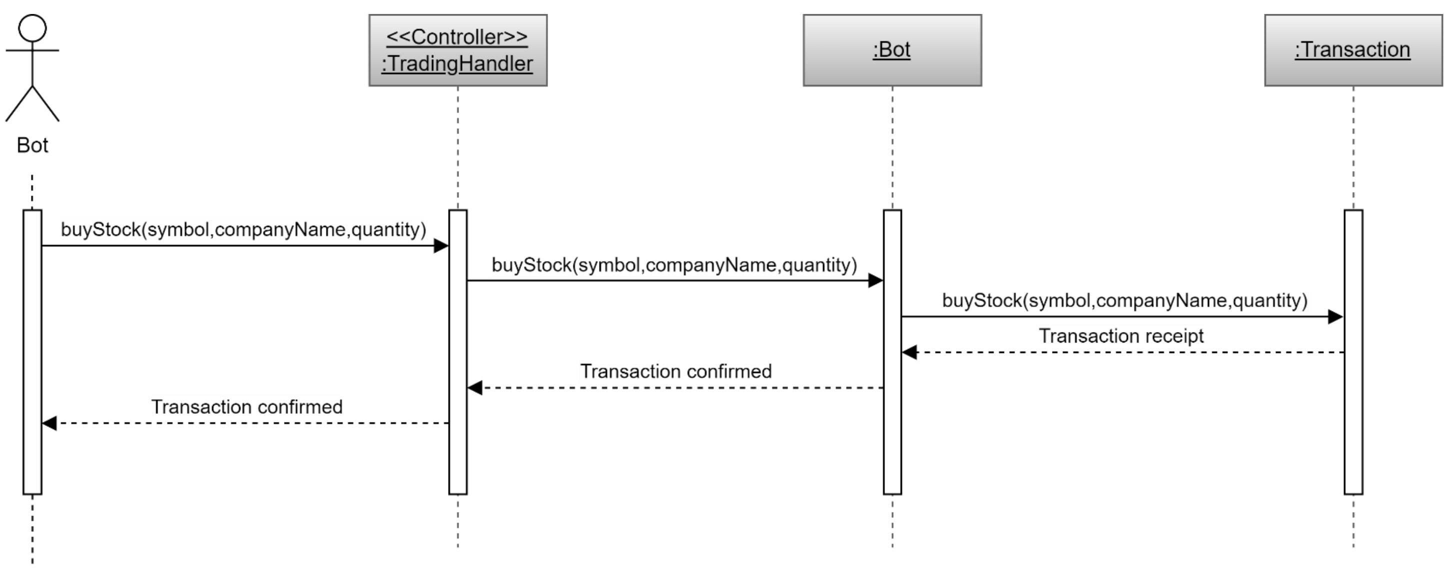

Figure 33. Buy stock Sequence Diagram

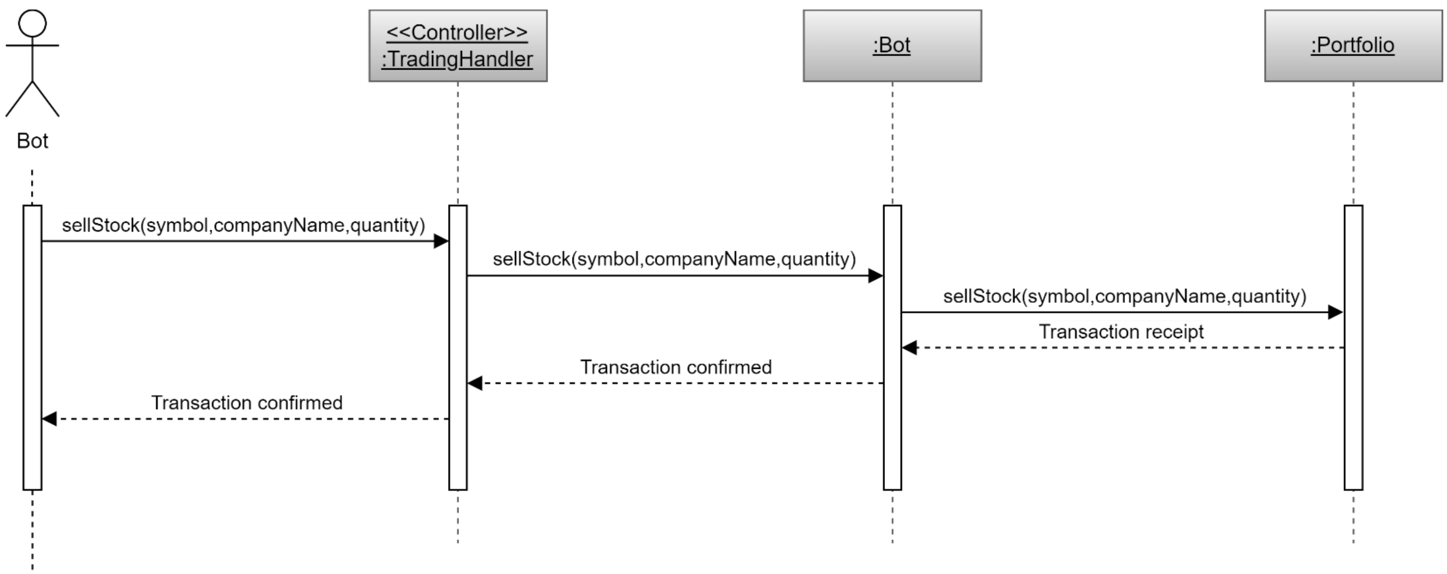

Figure 34. Sell stock Sequence Diagram

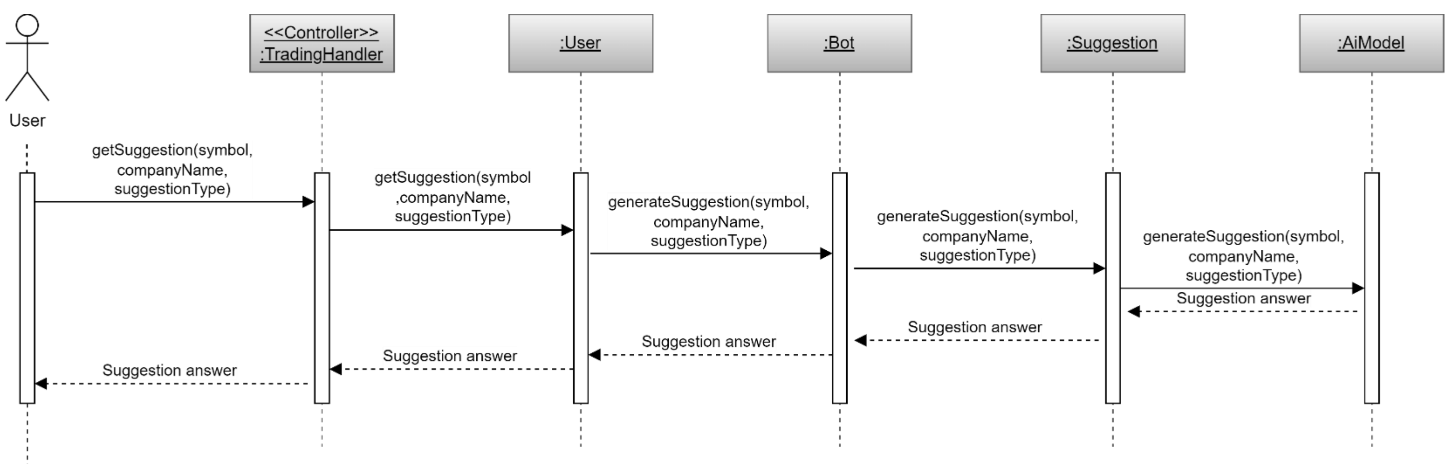

Figure 35. Get Suggestion Sequence Diagram

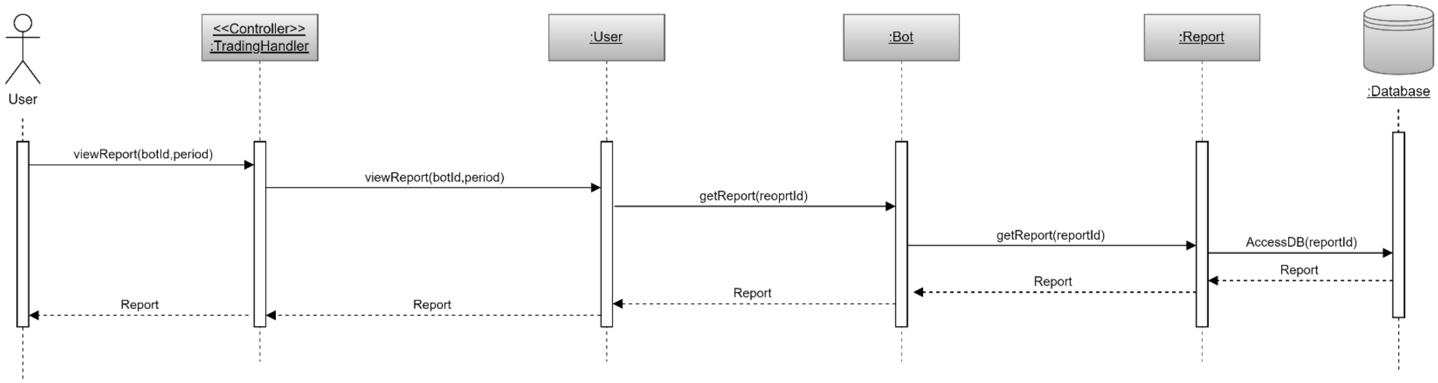

Figure 36. View Report Sequence Diagram

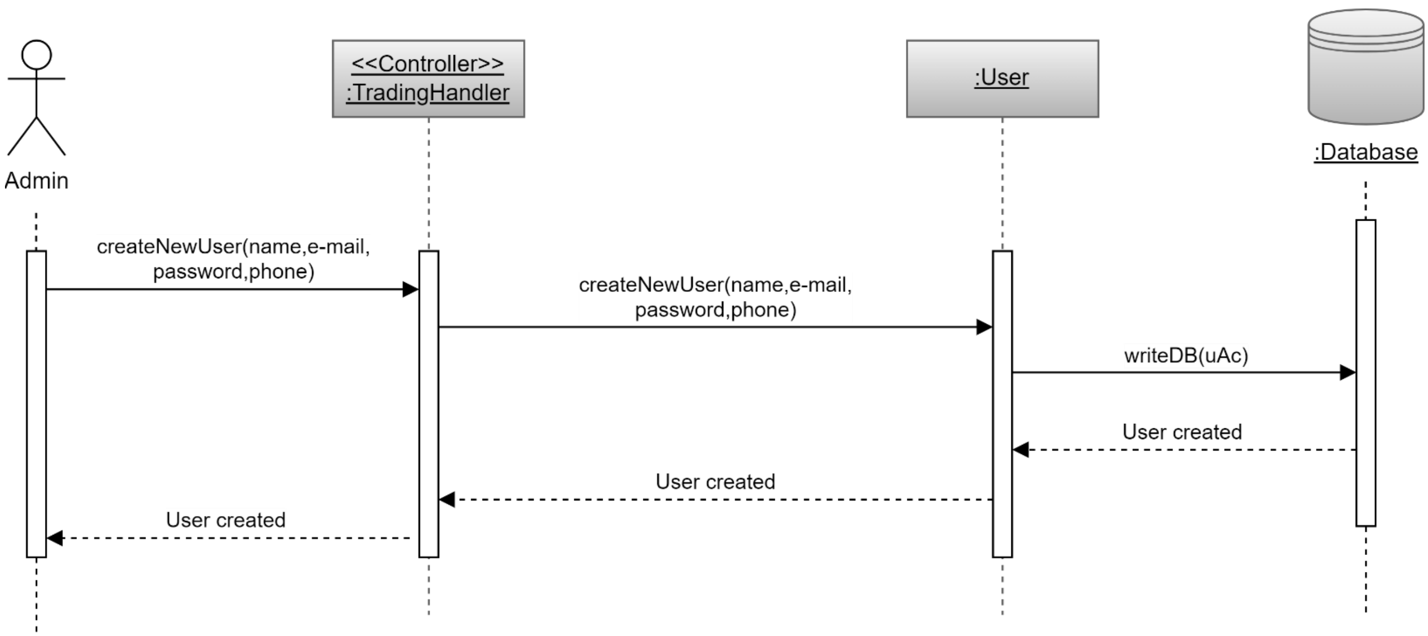

Figure 37. Create new user Sequence Diagram

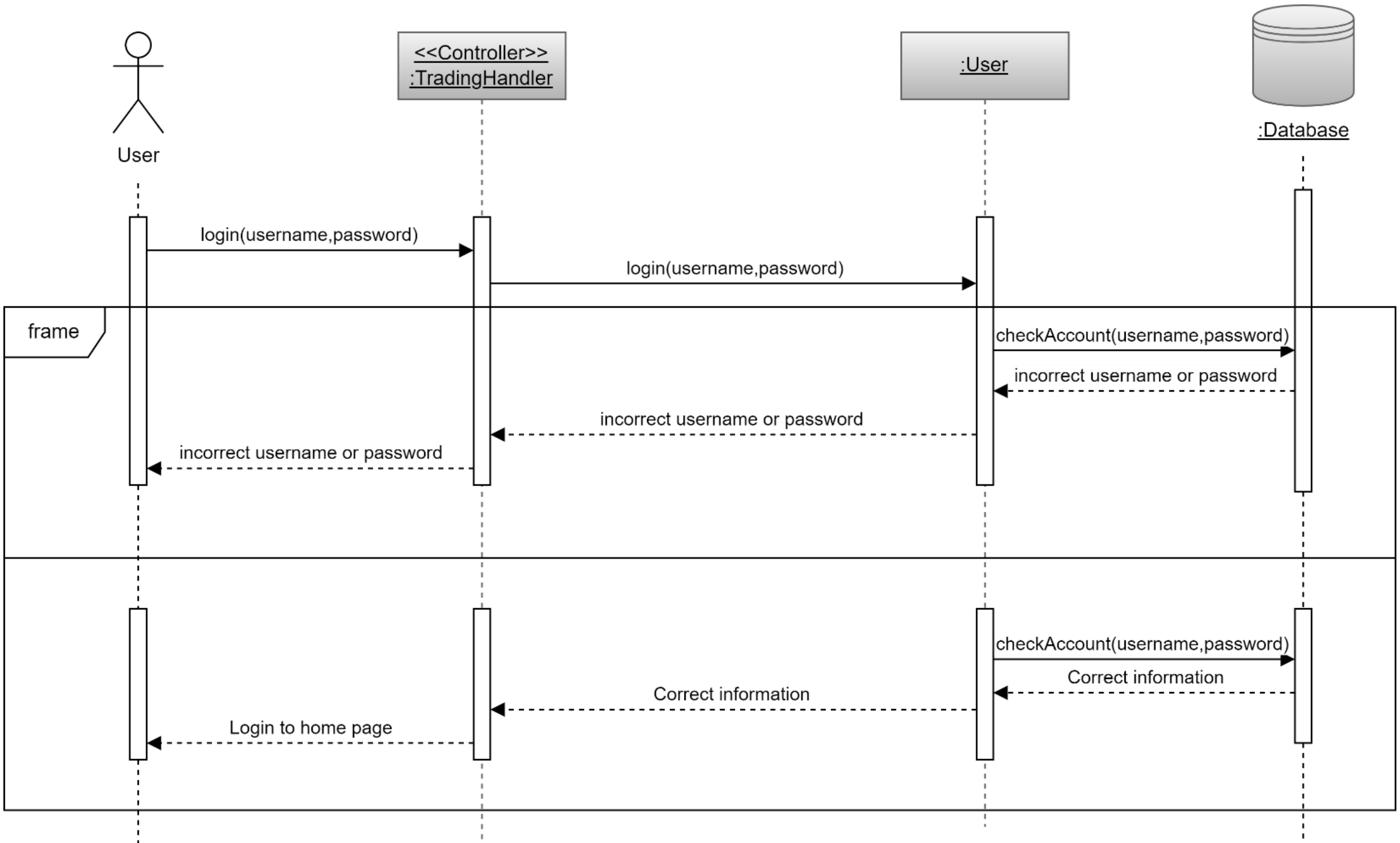

Figure 38. Login Sequence Diagram

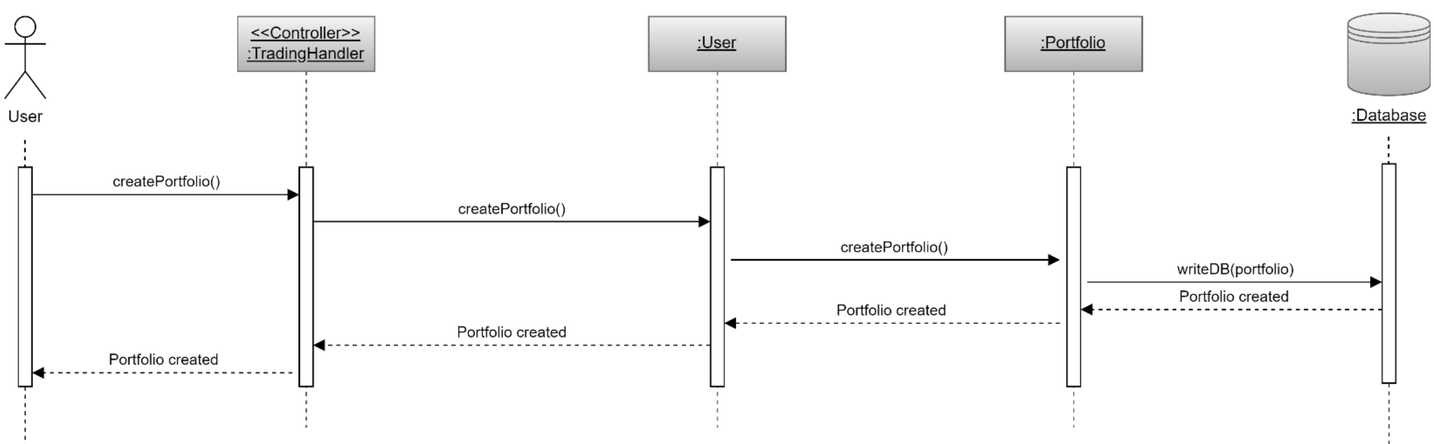

Figure 39. Create portfolio Sequence Diagram

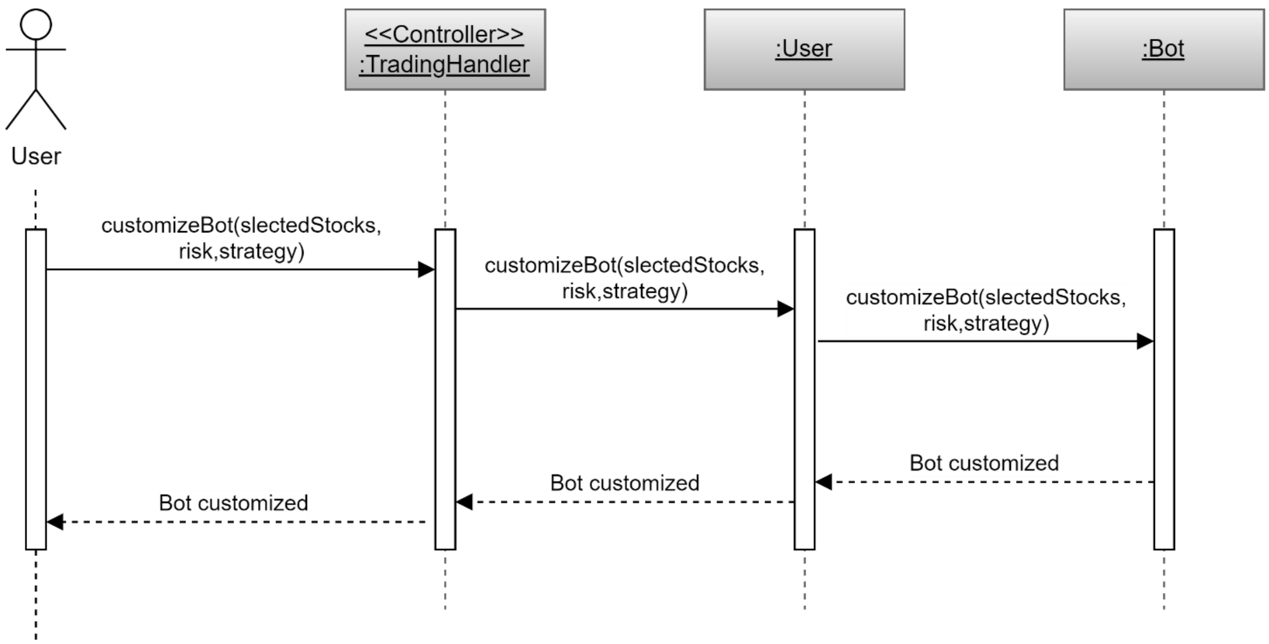

Figure 40. Customize Bot Sequence Diagram

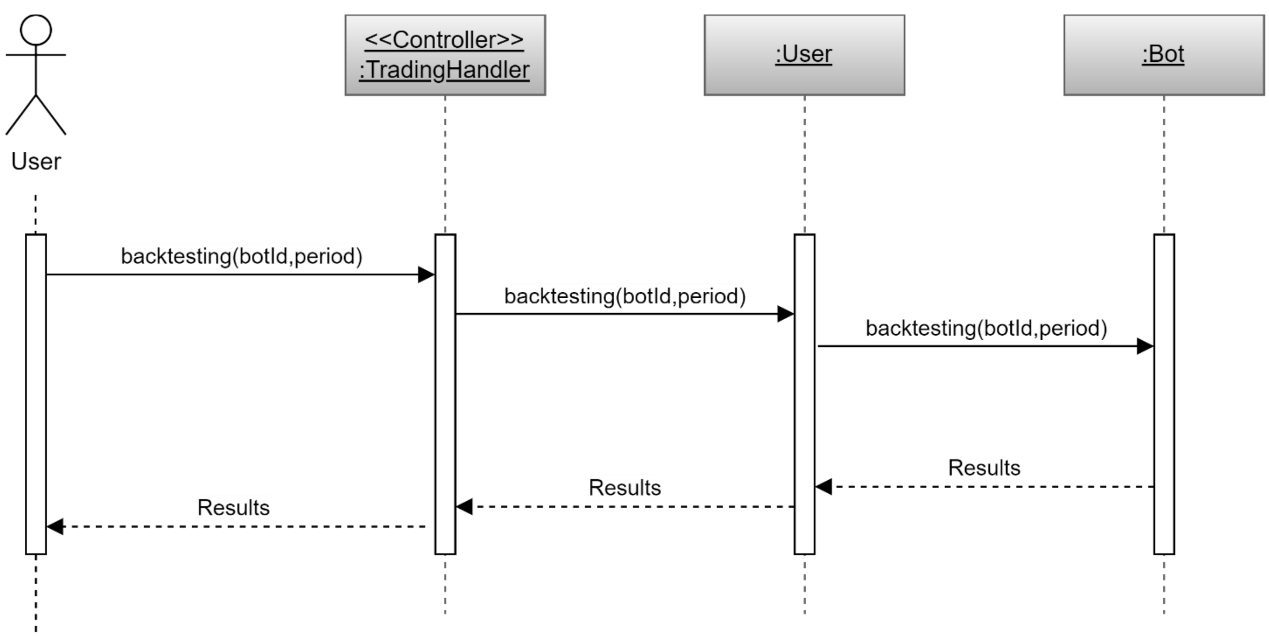

Figure 41. Backtesting Sequence Diagram

Final-Cut Class Diagram

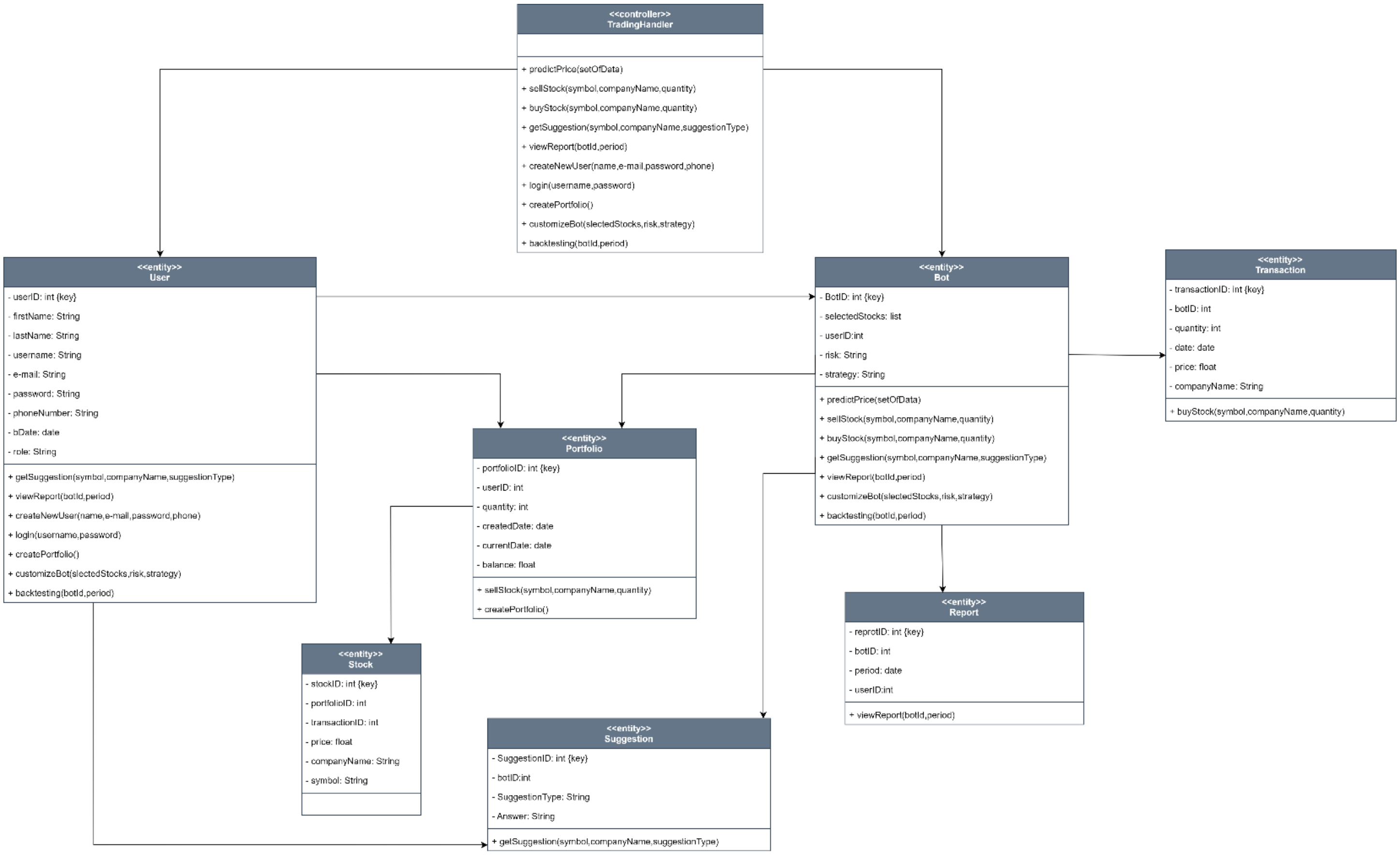

Figure 42. Final-Cut Class Diagram

Conclusion

Impact

Economic impact: The automated trading system could improve the efficiency and attractiveness of the Saudi stock market, and attract more investors from both inside and outside the country.

Social impact: By giving more people access to advanced trading tools and information, the platform could help close the gap between big and small investors. This could allow more Saudis to take part in and benefit from the stock market.

Technological impact: Building this new solution could encourage innovation and growth in the Saudi fintech industry, leading to more use of advanced technologies like AI and machine learning. This could make Saudi Arabia a leader in financial technology in the region and bring in top talent and investment.

Regulatory impact: Bringing in automated trading systems may require changes to current financial rules and monitoring systems in Saudi Arabia. Policy makers will need to balance encouraging innovation with keeping the market stable and protecting investors.

Challenges

Lack of publicly available information: One of the main challenges we encountered during the development of our automated trading system was the scarcity of detailed, publicly available information about the specific regulatory requirements and compliance obligations for automated trading in the Saudi stock market. This lack of transparency made it difficult to ensure that our system was designed in full compliance with all applicable regulations.

Model performance: Developing machine learning models that consistently generate accurate predictions and trading signals was a significant challenge. We had to experiment with various algorithms, feature engineering techniques, and model hyperparameters to find the optimal configuration that could adapt to changing market conditions and deliver reliable performance.

Data gathering: During the development of our automated trading system, we faced a significant challenge in acquiring sufficient historical news data specific to the Saudi stock market. Although we found a paid dataset with a good structure, the volume of news articles was much lower than we hoped for. This limited the amount of data available for training our machine learning models, potentially impacting the performance and accuracy of our potential news-based models. The lack of comprehensive, affordable, and easily accessible market-specific news data remained a concern throughout the project.

References

- Saudi Exchange
- Yahoo finance
- Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2016) 7th edition. System Analysis and Design in a Changing World.
- Elmasri, R., & Navathe, S. B. (2015) 7th edition. Fundamentals of Database Systems.
- Research papers:

  [1] D. Cao, B. H. Falk, and G. Tsoukalas, "Automated Market Makers and the Value of Adaptive Fees," Working Paper, Oct. 2023.

  [2] W. Addy, O. Odeyemi, A. Ajayi-Nifise, T. Falaiye et al., "Algorithmic Trading and AI: A Review of Strategies and Market Impact," World Journal of Advanced Engineering Technology and Sciences, vol. 11, no. 1, Feb. 2024.

  [3] S. Pawar et al., "Algorithmic Trading System," Conference Paper, Apr. 2024.

  [4] G. Sharma, S. Vidalis, P. Mankar, S. Kumar, and N. Anand, "Automated Passive Income from Stock Market Using Machine Learning and Big Data Analytics with Security Aspects," Journal For Global Academic & Scientific Research, vol. 2, no. 4, pp. 18–33, 2023.

  [5] D. Cliff, "Deep Learning can Replicate Adaptive Traders in a Limit-Order-Book Financial Market," Preprint, Nov. 2018.

  [6] H. A. Abdou, A. A. Elamer et al., "The Impact of Oil and Global Markets on Saudi Stock Market Predictability: A Machine Learning Approach," Energy Economics, Feb. 2024.

  [7] M. Alsulmi and N. Al-Shahrani, "Machine Learning-Based Decision-Making for Stock Trading: Case Study for Automated Trading in Saudi Stock Exchange," Research Article, Sep. 2022.

  [8] D. H. Bailey, "Advances in Financial Machine Learning," Book.

[9] M. Alsulmi, "Reducing Manual Effort to Label Stock Market Data by Applying a Metaheuristic Search: A Case Study From the Saudi Stock Market," Supported by the Research Center of College of Computer and Information Sciences, Deanship of Scientific Research, King Saud University, Saudi Arabia.

[10] R. Zada, "The Relationship Between Trading Volume, Stock Returns and Volatility in the Saudi Stock Market," Article, Sep. 2021.

[11] F. Alturki and A. M. Aldughaiyem, "Trading Saudi Stock Market Shares using Multivariate Recurrent Neural Network with a Long Short-term Memory Layer," International Journal of Advanced Computer Science and Applications, Sep. 2020.

Copyright Form

Project Title: Tadawul Tech Trader

Term of the Project: (1st / 2nd)__________Academic Year 1445

We the team (Team Members Names):

	Member Name	Member ID #	Member Signature
1	Abdulmohsen Alqahtani	442103210	
2	Osama Alharbi	442103126	
3	Raed Almajed	442102093	

Hereby assign our copyright of this report and of the corresponding executive summary to the Information System (IS) Department of King Saud University. We also hereby agree that the report or demo from our oral presentations is becoming full property of the IS Department. The publication of this report does not constitute approval by King Saud University, the IS Department or its faculty members of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.