



YourStock

Buy the rumor,
sell the news



كلية الحاسوب والذكاء الاصطناعي
Faculty of Computers & Artificial Intelligence



HELWAN UNIVERSITY
Faculty of Computers and Artificial Intelligence
Information System Department

[Your Stock]

A graduation project dissertation by:

[Ahmed El Sayed Mahmoud El Sayed (201900019)]

[Omar Ahmed Ibrahim Hamed (201900498)]

[Omar Adel Abdel Halim Mohamed (201900512)]

[Mohamed Magdy Mohamed Mohamed (201900724)]

[Abdel Rahman Ezzat Mohamed Salem (201900422)]

[Abdel Rahman Hussein Mahmoud Mohamed (201900413)]

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in
Computers & Artificial Intelligence, at the Information System Department, the Faculty of
Computers & Artificial Intelligence, Helwan University

Supervised by:

[Dr. Laila Abdel Hamid]

June 2023

Acknowledgements

This work would not have been possible without the effort that my team put in this project we worked so hard to get this done. I am grateful to all of those with whom I have had the pleasure to work during this and other related Projects . Each of the members of my team has provided me extensive personal and professional guidance and taught me a great deal about both technical solutions and life in general. I would especially like to thank Dr. Laila Abdel Hamid, As my teacher and mentor. I would like to thank my friend who has experience in this field, he helped me a lot with answering my questions.

Nobody has been more important to me in the pursuit of this project than the members of my family. I would like to thank my parents, whose love and guidance are with me in whatever I pursue. They are the ultimate role models.

Table of Content

Abstract	8
Chapter 1: Introduction.....	9
1.1 Overview	10
1.2 Objectives.....	10
1.3 Purpose.....	11
1.4 Scope.....	11
1.5 General Constraints.....	12
Chapter 2: Project “Planning and analysis”	13
2.1 Project planning:.....	14
2.1.1 Feasibility Study:	14
2.1.2 Estimated Cost:	15
2.1.3 Gantt Chart:	17
2.2 Analysis and Limitation of existing system:	18
2.3 Need for the new system:	18
2.4 Analysis of the new system:.....	18
2.4.1 User requirements:	18
2.4.2 System Requirements.....	19
2.4.3 Domain Requirements:	19
2.4.4 Functional Requirements:	19
2.4.5 Non-Functional Requirements:	20
2.5 Advantages of the new system:.....	22
2.6 Risk and Risk Managements:	22
Chapter 3: Software Design:.....	27
3.1 Design of Class Diagram:	28
3.2 Use case diagram:	29
3.4 sequence diagram:	33

3.5 activity diagram:	35
Chapter 4: Implementation:	36
4.1 Software tools.....	37
4.2 Software architecture.....	38
4.3 Software user interface (Screenshots).....	40
4.4 Code examples (Screenshots).....	46
4.4 Pseudocode, Flowchart or workflow.....	50
Chapter 5: Testing:	51
5.1 Unit Testing:	52
5.2 Integrated testing:	53
5.3 Additional Testing:	54
Chapter 6: Results and Discussion:	55
6.1 Results	56
6.1.1 Expected result:	56
6.1.2 Actual results:	56
6.2 Discussion:	57
Chapter 7: Conclusion:	58
Chapter 8: Future work:	60
Bibliography:	62

Lists of Tables

Table 1 (Functional requirements).....	19
Table 2 (Non Functional requirements).....	20
Table 3 (Use case scenarios).....	29
Table 4 (Unit test case).....	52

Lists of Figures

Figure 1 (Estimated Cost).....	16
Figure 2 (Gantt Chart).....	17
Figure 3 (Class diagram)	28
Figure 4 (Log in& Sign up use case diagram)	29
Figure 5 (Home page use case diagram)	30
Figure 6 (Use case diagram)	32
Figure 7 (Sequence diagram)	33
Figure 8 (Activity diagram)	35
Figure 9 (Software architecture)	39
Figure 10 (Firebase architecture)	39
Figure 11 (Machine Learning Model Diagram)	39
Figure 12 (User Interface (Screenshots))	40
Figure 13 (Code Example Screenshots)	46

Abstract

Financial prediction, especially stock market prediction, has been one of the most attractive topics for researchers and investors over the last decade. The project aims to provide retail investors with a third-party investment mobile application to navigate through the stock market. This is achieved through the use of machine learning and mobile web technologies. Several stock price prediction approaches and models are developed with neural networks, simple linear regressions, Gradient boosting and deep-learning (LSTM). The project serves as a foundation for democratizing machine learning technologies to the general public in the context of discovering investment opportunities. It paves the way for extending and testing out new models and developing AutoML in the financial context in the future.

Chapter 1: Introduction

In this chapter we are going to discuss and see the overview of the project and know more about its scope and limitations and objectives and constraints

1.1 Overview:

Retail investors spend a lot of time finding investment opportunities. Wealthier investors could seek professional financial advisory services, but for typical retail investors, the costs are prohibitive. Thus, retail investors have to figure out the market themselves and make informed decisions on their own. This makes investment very stressful in modern societies. This project aims to provide stock price predictions based on the latest machine learning technologies to all retail investors include features such as real-time news alerts, market analysis, and customizable portfolios, which can help users stay up-to-date on the latest market trends and adjust their investment strategies accordingly and also provide them with the latest news related to the stock market. A mobile web application is developed to provide predictions in an intuitive way.

1.2 Objectives:

- ❖ The ultimate goal of our application is to serve retail investors as a third party investment tool that uses machine learning to help them navigate in the fast-changing stock market.
- ❖ The project aims to support the user with real time streaming data gives him information about stock prices and charts.
- ❖ The project aims to introduce and democratize the latest machine learning technologies for retail investors.
- ❖ The project aim to support the user with the latest news related to the stock market to keep him up to date.
- ❖ No prediction is 100% accurate. Therefore, the upper bound and lower bound of the stock prices will be displayed to illustrate the trading range the investors should be looking at.
- ❖ This application serves as a supplementary quantitative tool for investors to see the market at a different perspective with the help of technology.[1.3](#)

1.3 Purpose:

- Application aims to serve retail investors as a third party investment tool.

- Application aims to facilitate the use of stock applications and giving the user accurate information and be trust worthy.
- Application aims to support the user with the information about any stock that he needs.

1.4 Scope:

scope refers to the boundaries and limitations of a software development project. It defines what the software will and will not do , and outlines the specific features and functionality that will be included in the final product.

The software scope is typically defined at the beginning of the project, during the planning and requirements gathering phase. This helps to ensure that all stakeholders have a clear understanding of what the software will deliver, and helps to prevent scope creep - the tendency for the project to expand beyond its original boundaries.

Project goals: serving retail investors as a third party investment tool that uses machine learning to help them navigate in the fast-changing stock market and support them with up to date information.

Project deliverables: Main Features:

- User will be able to view prices and details of any stock.
- User will see charts of any stock.
- User can search stocks by name or ticker.
- User can see news related to the stocks.
- User can add stocks in his watchlist (Stocks that he wants to be in touch with).
- User can filter stocks (between their high, low prices , volume ,etc....).
- User can send feedbacks about his personal opinion in the application (pros and cons).

Project constraints: market danger, due to the total systematic risk afflicting the financial markets, investors may suffer losses. A prime illustration of increased market risk is stock market crashes. Although it cannot be totally eradicated, market risk can be protected.

Business risk, the most frequent risk facing investors who buy individual equities is a company-specific risk. Investors risk losing their money if the firm they invested in is unable to generate sufficient sales or profits. The market value of a corporation might also decrease due to subpar operational performance.

User trust, due no prediction is 100% accurate. Therefore, the upper bound and lower bound of the stock prices will be displayed to illustrate the trading range the investors should be looking at.

Project Exclusions: Demo account the user can make to practice investing with our prediction model but due to shortage of time we excluded it from the project.

1.5 General constraints:

- Predicting the market is challenging because the future is inherently unpredictable.
- The accuracy and reliability of stock market data can vary widely, and may be subject to errors, omissions, or biases. This can impact the accuracy of any predictions that are based on that data.
- Short-term traders are typically better served by waiting for confirmation that a reversal is at hand, rather than trying to predict a reversal will happen in the future.
- Viewing price action as a series of waves is an alternative to predicting future price moves.
- Stock market prediction applications may require significant computing power and processing capabilities, and may be subject to limitations related to hardware or software compatibility.
- Establishing significant points to buy and sell should be based on what price is actually doing, rather than what we expect it to do.
- Stock market prediction applications may be subject to user behavior, such as herd behavior or emotional decision-making, which can impact the accuracy of any predictions that are based on user input or feedback.

Chapter 2: Project “Planning and analysis”

In this chapter we are going to discuss how we plan the project and show
The steps and the instructions that we have followed to plan the application.

2.1 Project planning:

2.1.1 Feasibility Study:

1.Financial Feasibility

The system will follow the freeware software standards and no cost will be charged from the potential customers. Gathering stock information and livestreaming will be done with no cost. Bug fixes and maintaining tasks will have no associated cost. there will be many benefits for the customers. Especially with the technical rating feature that will help them taking decisions whether it's buy or sell. From these it's clear that the project is financially feasible.

2.Techical Feasibility

Each of the technology used in the application is freely available and the technical skills required are manageable. Time limitations of the project development and the ease of implementing using these technologies are synchronized from these it's clear that the project is technically feasible.

3.Resource and Time Feasibility

Resource feasibility resources that are required for the project includes, Programming device (Laptop), Programming tools (freely available), Backup tools Programming individuals so it's clear that the project has the required resource feasibility.

4.Risk Feasibility

Risk feasibility can be discussed under several contexts. Risk of the prediction of the stock prices that the system supports the user with it because there is unexpected disasters that could happen like corona pandemic that affect the stock market plus there is no prediction with 100% accuracy ,that might be a

misleading sometimes. The user will should know that any of these decisions will take under his own risk.

5.Operational Feasibility

In case of usability and operation app should allow users Flexible navigation between pages of stocks easily. It's easy to operate. In case of maintainability app issues fixed as soon as possible from these it's clear that the project is operationally feasible.

2.1.2 Estimated Cost:

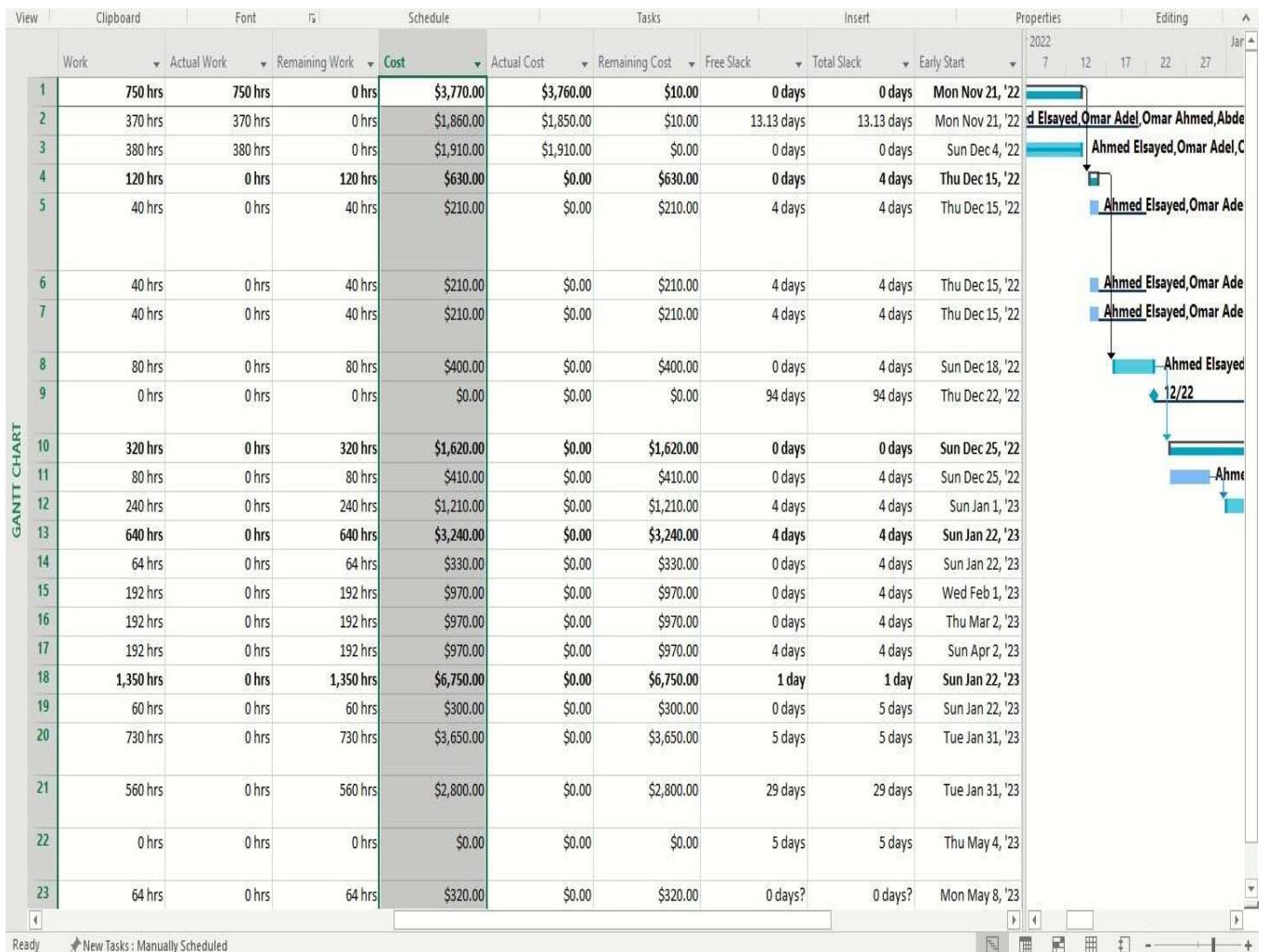
The estimated cost of developing a stock market prediction application can vary widely depending on a number of factors, including the complexity of the application, the development team's experience and expertise, and the amount of time and resources required to complete the project.

it is difficult to provide a specific estimate for the cost of developing a stock market prediction application. However, depending on the complexity of the application and the resources required to develop it. It is important to carefully consider all of the factors that may impact the cost of developing the application, and to work with a development team that has expertise in this area to ensure that the project is completed on time and within budget.

Here are some factors that can impact the estimated cost of developing a stock market prediction application:

- Features and functionality: The more features and functionality the application includes, the more time and resources will be required to develop it.
- Data sources: The cost of accessing and using stock market data sources can vary widely depending on the data provider and the frequency of updates.
- Technical requirements: The cost of developing an application can be impacted by technical requirements such as hardware and software compatibility, user interface design, and security features.

- Development team: The cost of hiring a development team can vary depending on the experience and expertise of the team members, as well as their location and other factors.
- Testing and quality assurance: The cost of testing and quality assurance can be significant, especially for an application that requires real-time data.
- Maintenance and support: The cost of maintaining and supporting the application over time should also be taken into account.



2.1.2 Gantt Chart:



2.2 Analysis and Limitation of existing system:

- Compatibility: The application works on android not IOS.
- Performance: The application may not perform optimally on low-end hardware or slow internet connections.
- Maintenance: The application may require regular maintenance or updates to function properly.

2.3 Need for the new system:

- Outdated technology: The existing system may be based on outdated technology that is no longer supported.
- User needs: The existing system does not meet the needs of its users, such as lacking certain features or a user-friendly interface.

2.4 Analysis of the new system

2.4.1 User requirements:

- The system should be easy and simple to operate.
- The software should provide the functionality that the user needs to perform their tasks effectively and efficiently
- The software should have a user interface that is easy to use and understand, and that meets the needs of the user
- The software should perform well, with minimal lag or delay, and should be able to handle large amounts of data or complex tasks.
- The software should be reliable, with minimal errors or crashes, and should be able to recover quickly from any issues that arise.
- The software should be secure, with appropriate measures in place to protect user data and prevent unauthorized access avoid data breaches.
- The software should be compatible with the user's hardware and software environment
- The software should be supported and maintained over time, with regular updates and bug fixes as needed

- The user will be able to keep track to the stock market at any time.

2.4.2 System Requirements:

Operating system requirements: system requires Android & iOS system

Storage requirements : system requires a minimum amount of 1GB storage space

Network requirements : system requires a connection with the internet to access.

2.4.3 Domain Requirements:

- User must have to make an account to use the application.
- Multiple users must be able to use the application simultaneously without crashing.
- The application need to provide a user-friendly interface that meets the needs of its users, such as ease of use or accessibility features.
- The application must update capabilities for future models.

2.4.4 Functional Requirements:

Register	<ul style="list-style-type: none"> • If the user doesn't have an account , he must register. • User have to register with essential details like (First Name, Last Name, Email, Password, confirm password, phone number, Address).
Login	If user is already registered so he can login by using his username and password.
User stories	<ul style="list-style-type: none"> • User need to see the prices of the stocks in estimate time and the prediction of the application to this stocks. • User can mark stocks as his favorite and add them to watchlist to be in touch with the changes. • User need to update profile information.

Use cases	<ul style="list-style-type: none"> User check the app from time to time to be updated to the changes of the prices, news. User know the prediction of the model to any stock.
Used technology	<ul style="list-style-type: none"> Flutter TensorFlow Python Firebase
Additional functionality	<ul style="list-style-type: none"> Search engine for the user to facilitate the search on different stocks. User can be updated to the changes of the prices to his watchlist stock. Filter stocks to minimizing the results to the user to see what he needs.
Integration with other systems	<ul style="list-style-type: none"> The system able to integrate with other systems like accounting software (User can sign in with google account).

2.4.5 Non- Functional Requirements:

Usability	<ul style="list-style-type: none"> User interface must be simple to make users familiar with it. Application should allow users Flexible navigation between pages of stocks easily.
Reliability	<ul style="list-style-type: none"> The system must be up and running in 24 hours. Stocks and user information must be up to date. Application should perform well in low latency conditions shows stocks with the lowest delay possible.
Availability	<ul style="list-style-type: none"> System functionalities must always be available. Monitoring and alerting such as using automated systems to detect and alert technical staff to potential problems,

	can help ensure that issues are detected and addressed quickly.
Security	<ul style="list-style-type: none"> The Application has detailed protocols for authentication accounts. The Application doesn't use sensitive information about the user.
Performance	<ul style="list-style-type: none"> Optimizing code and algorithms can help improve performance by reducing processing time and improving memory usage. Caching data can help improve performance by reducing the time required to retrieve data. Content loading speed is priority, user doesn't want to wait too much time.
Scalability	<ul style="list-style-type: none"> System should be capable enough to handle almost hundreds of users simultaneously without affecting its performance.
Accessibility	<ul style="list-style-type: none"> Use the simplest language possible for instructions, prompts and outputs. Provide accessible training and support materials. Ensure that the user interface and task flow are similar across different functions. System functionalities must always be available.
Maintainability	<ul style="list-style-type: none"> Designing for maintainability designing products or systems with ease of maintenance in mind. Conducting regular maintenance such as inspections and preventative maintenance, can help identify and address problems before they become significant issues. Application issues fixed as soon as possible.

2.5 Advantages of the new system:

- Better user experience: The new system has a more intuitive user interface and provide better user support (User can send his feedback about the application).
- Cost savings: User doesn't need subscription to use specific services It will be free stock market prediction app.
- Improved security: The new system have better security features leading to improved data protection and reduced risk.

2.6 Risk and Risk Managements:

Operational Risk: operational risks include:

- Technology risk - The risk of loss resulting from technology failures, such as system crashes or cyber attacks.
- Human error risk - The risk of loss resulting from mistakes or errors made by employees, such as data entry errors or incorrect transactions.
- Fraud risk - The risk of loss resulting from fraudulent activities, such as embezzlement or money laundering.
- Legal and regulatory risk - The risk of loss resulting from non-compliance with laws and regulations, such as fines or legal fees.
- Business continuity risk - The risk of loss resulting from disruptions to business operations, such as natural disasters or power outages.

What to do to avoid operational risk or at least reduce it?

- Use technology and automation - Implementing technology solutions, such as automated systems and artificial intelligence (AI) tools, can help reduce operational risk by improving accuracy, efficiency, and consistency.
- Develop and implement robust internal controls - This includes policies, procedures, and processes to manage operational risk. Businesses should

regularly review and update their internal controls to ensure they are effective and up-to-date.

- Conduct regular risk assessments - Regularly assess potential operational risks and the likelihood and potential impact of these risks. This can help identify areas for improvement and inform risk management strategies.
- Implement effective training and communication programs - Employees should be trained on how to identify and manage operational risks, as well as the importance of internal controls and risk management procedures.
- Effective communication can help ensure that everyone in the organization is aware of operational risks and how to manage them.
- Have backup plans and disaster recovery procedures - Businesses should have backup plans and procedures in place in case of disruptions, such as natural disasters or technology failures. This can help ensure business continuity and minimize the impact of operational losses.

Schedule Risk: schedule risk refers to the risk that a project or activity will not be completed on time or within the expected timeframe. It is the risk that arises from uncertainties in the duration of project activities, the availability of resources, and the interdependencies between tasks.

Schedule risk can have a significant impact on a project's success, particularly for time-sensitive projects. It can result in delays, missed deadlines, increased costs, and a negative impact on project stakeholders.

What to do to avoid schedule risk or at least reduce it?

- Develop a detailed project plan - Developing a detailed project plan that includes task dependencies, timelines, and resource requirements can help identify potential scheduling risks and inform risk management strategies.
- Conduct a schedule risk analysis - Conducting a schedule risk analysis can help identify potential schedule risks and their likelihood and potential impact. This can inform risk management strategies and help manage stakeholder expectations.

- Use project management software - Project management software can help track project progress, identify potential scheduling risks, and manage project resources effectively.
- Monitor and control project progress - Regularly monitoring and controlling project progress can help identify potential scheduling risks and take corrective action to address them before they become significant issues.
- Build in contingency plans - Building in contingency plans, such as additional resources or alternative approaches, can help mitigate schedule risks and minimize the impact of any delays.

Technical risk: technical risk refers to the risk of loss resulting from technical failures or problems in the development, implementation, or operation of a product or system. It is the risk of loss resulting from technical issues, including hardware, software, and infrastructure problems, that can lead to project delays, increased costs, and decreased quality.

Technical risks include:

- Hardware failure - The risk of loss resulting from hardware failures, such as server crashes or equipment malfunctions.
- Software bugs - The risk of loss resulting from software bugs or errors, which can lead to project delays, increased costs, and decreased quality.
- Data security breaches - The risk of loss resulting from data security breaches, which can lead to loss of sensitive data, reputational damage, and legal and regulatory penalties.
- Interoperability issues - The risk of loss resulting from interoperability issues between different systems or technologies.
- Technology obsolescence - The risk of loss resulting from the obsolescence of technology, which can lead to increased costs, decreased efficiency, and decreased quality.

What to do to avoid operational risk or at least reduce it?

- Conducting a technical risk assessment - Conducting a technical risk assessment can help identify potential technical risks and their potential impact on the project.
- Developing a contingency plan - Developing a contingency plan can help mitigate the impact of technical risks by outlining alternative approaches or solutions.
- Implementing quality control processes - Implementing quality control processes, such as code reviews and testing, can help identify and address technical risks early in the development process.
- Regularly updating hardware and software - Regularly updating hardware and software can help prevent technical risks resulting from outdated technology.
- Providing training and support - Providing training and support to employees can help reduce the risk of technical problems resulting from user error or lack of knowledge.

Information risk: Information risk refers to the risk of loss or damage to sensitive or confidential information or giving the user wrong information about the prices of the stocks. It is the risk that arises from the potential exposure, theft, or unauthorized access to sensitive information, such as personal data, financial information, or trade secrets.

Information risk can have a significant impact on an organization's financial performance and reputation, particularly in industries such as finance, healthcare, and technology. It can result in legal and regulatory penalties, loss of customer trust, and damage to brand reputation.

What to do to avoid Information risk or at least reduce it?

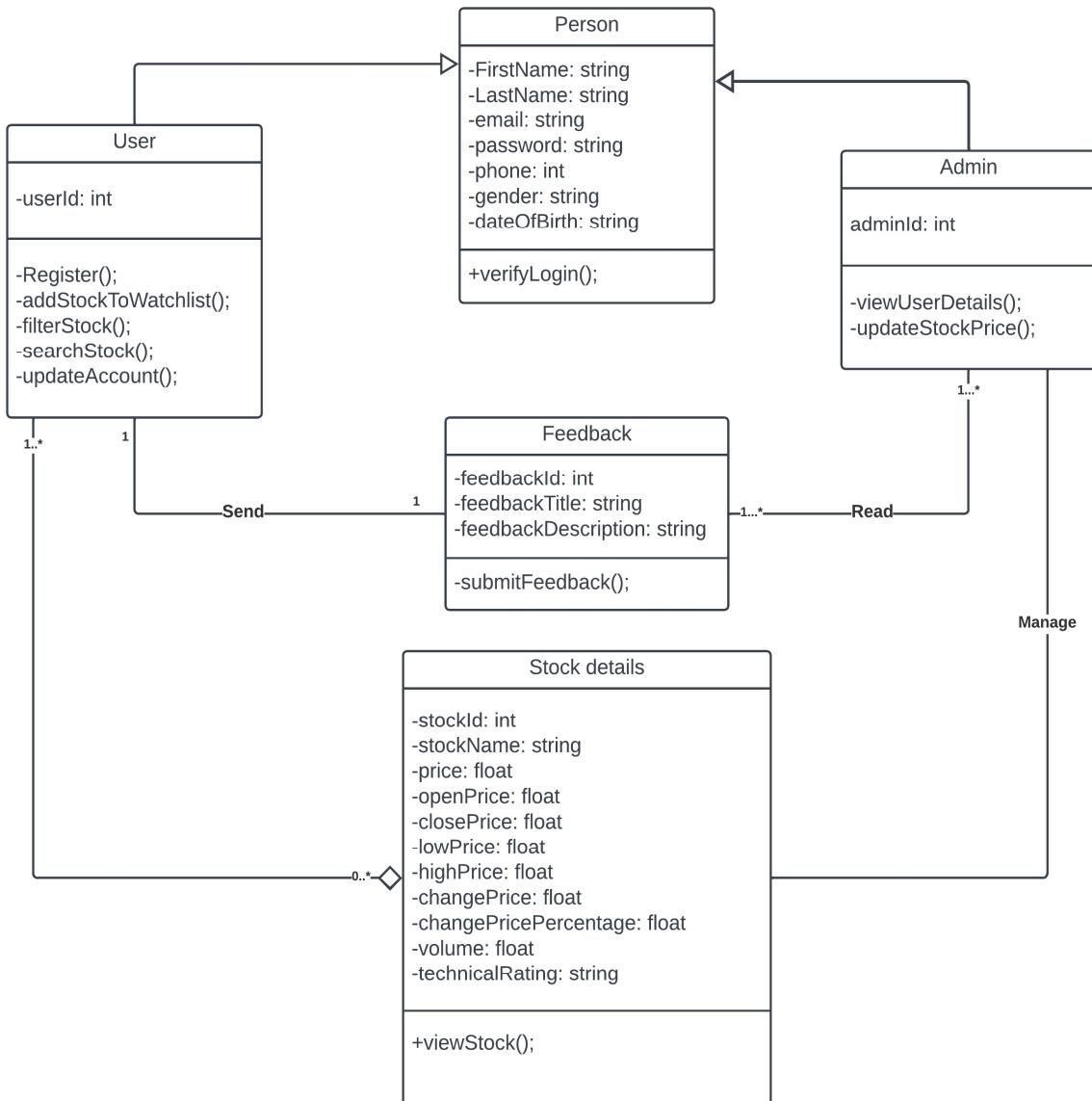
- Conducting a risk assessment - Conducting a risk assessment can help identify potential information risks and their potential impact on the organization.
- Developing an information security policy - Developing an information security policy that outlines procedures and guidelines for protecting sensitive information can help reduce the risk of information loss or exposure.

- Implementing access controls - Implementing access controls, such as user authentication and role-based access, can help prevent unauthorized access to sensitive information.
- Regularly updating software and systems - Regularly updating software and systems can help prevent information risk resulting from outdated technology.
- Providing employee training and awareness - Providing training and awareness to employees can help reduce the risk of information loss or exposure resulting from user error or lack of knowledge.
- Monitoring and auditing - Regularly monitoring and auditing information systems, networks, and processes can help detect potential information risks and take corrective action before they become significant issues.

Chapter 3: Software Design

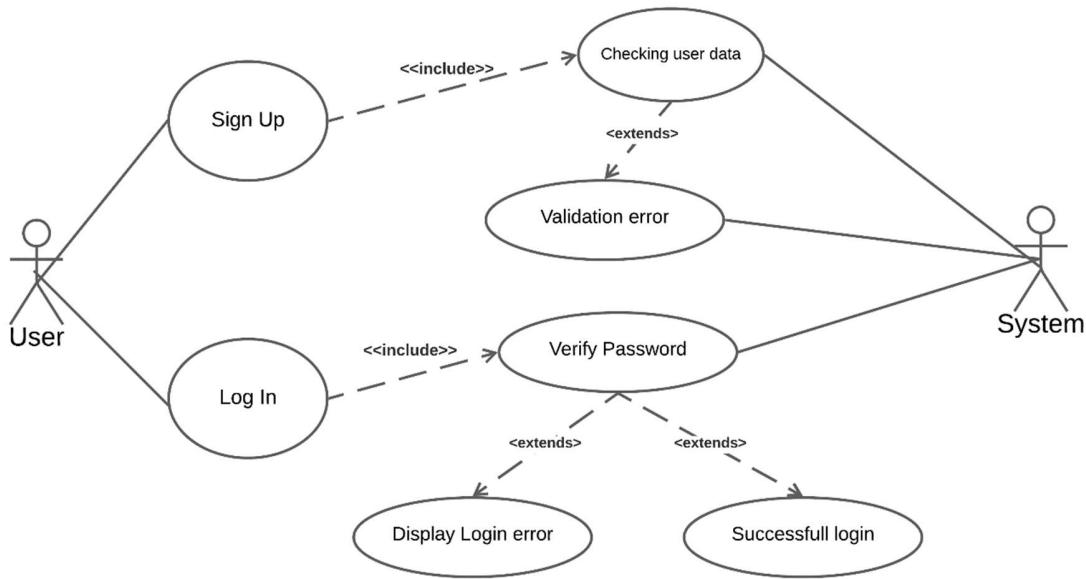
In this chapter we're going to discuss and go deeper in our application design and present its diagrams

3.1 Design of Class Diagram:



3.2 Use case diagram:

1. Signup and login use case and scenario:

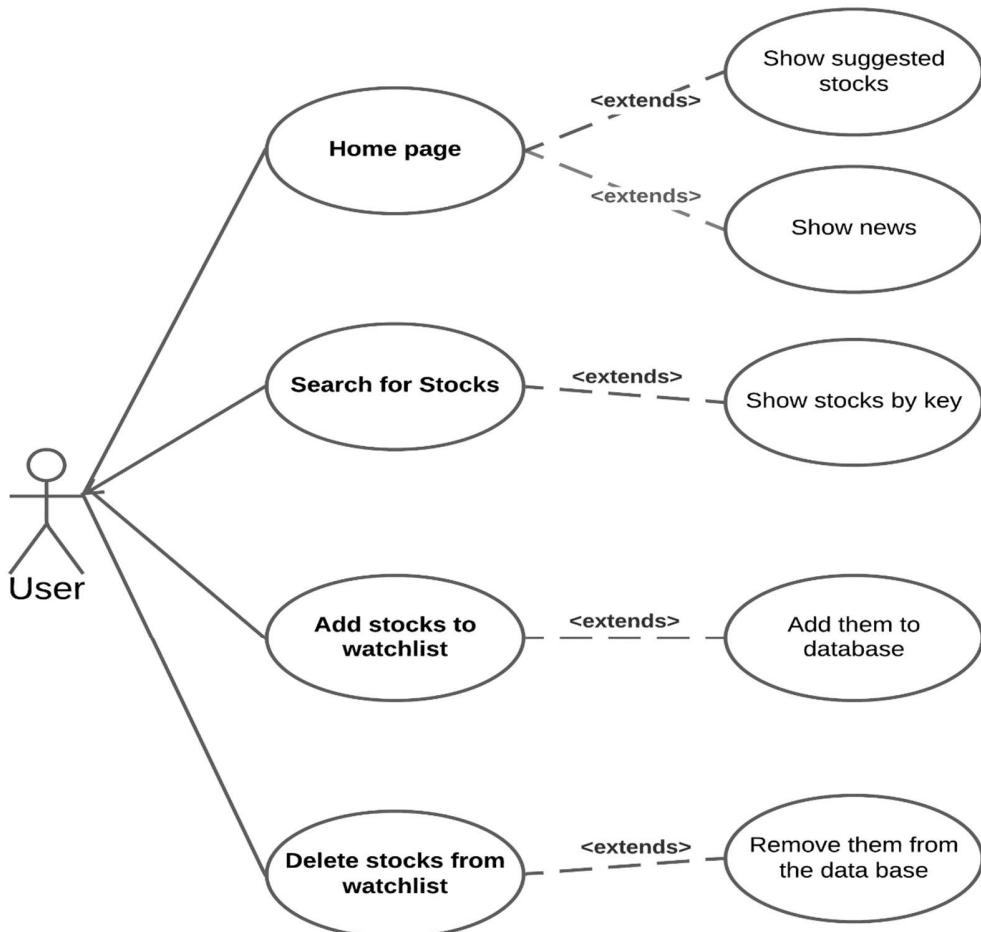


Use case name	Sign up
Actors	User, System
Pre-Conditions	If the user will sign up, his email must be unique and there is no other user used it before.
Post-Conditions	-If the registration was successful, the user is now registered into the system. -If not, the system state remains unchanged and sends error message
Flow of details	<ul style="list-style-type: none"> when the user sign up he must enter : (email, password) the system check the validation of each field, if there is no validation error so ,the user will successfully signed up.
Notes	Email address must be unique. Password must contain only letters and numbers.

Use case name	Log in
Actors	User, System
Pre-Conditions	if the user will login, he must be signed up before.

Post-Conditions	If the logging in was successful, the user is now logged in into the system and the home page will be opened. If not, the system state remains unchanged and sends error message
Flow of details	<ul style="list-style-type: none"> • If the user will log in he must enter his email & password • The system check user email and password
Notes	Email must be registered in the database. Password must be the same as the registered password.

2. Home page case scenario:

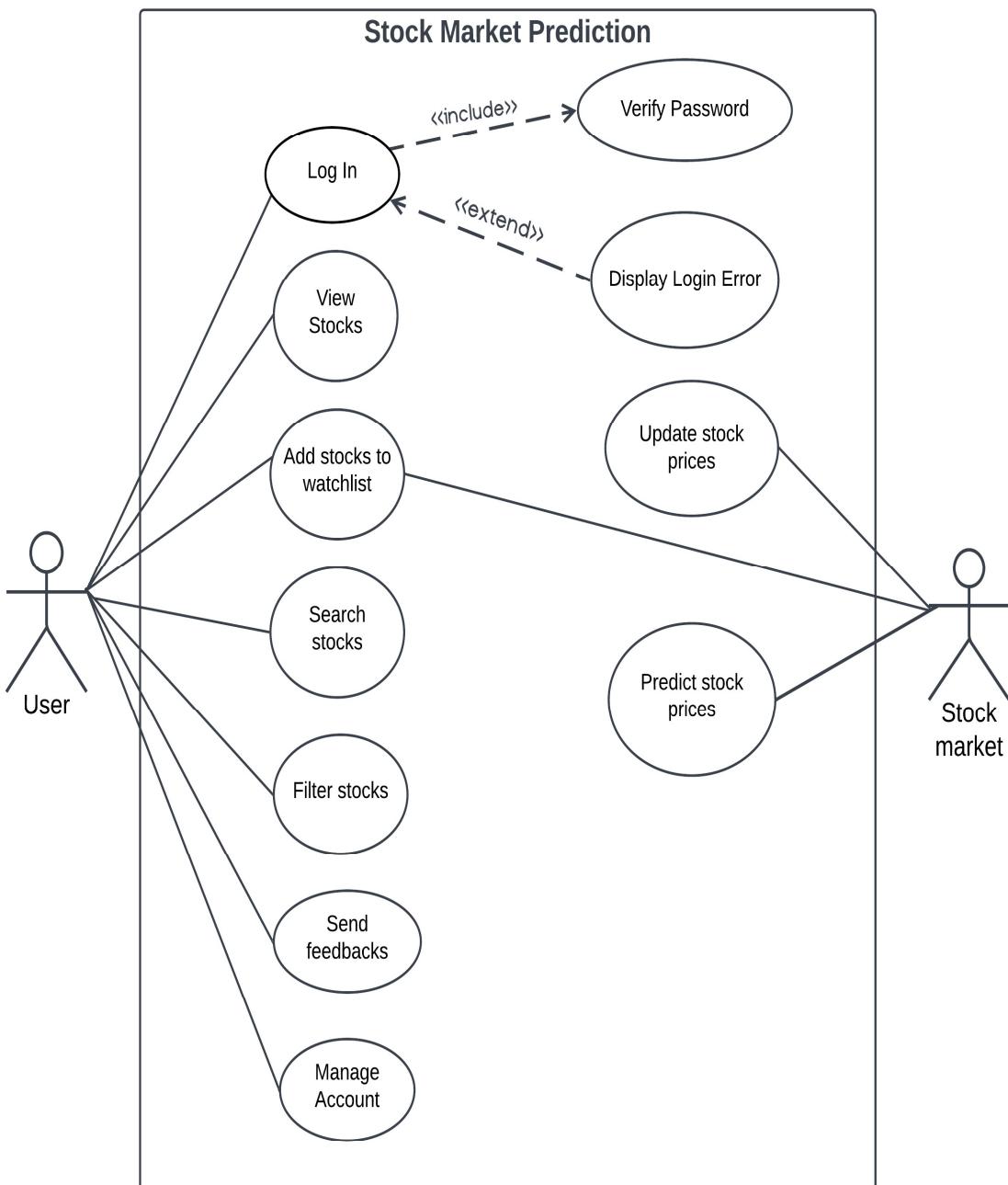


Use case name	Home page
Actors	User , Suggested stocks
Pre Conditions	User must be logged in to view suggested stocks.
Post Conditions	None
Flow of events	User can scroll up and down and click on stocks to see details information.

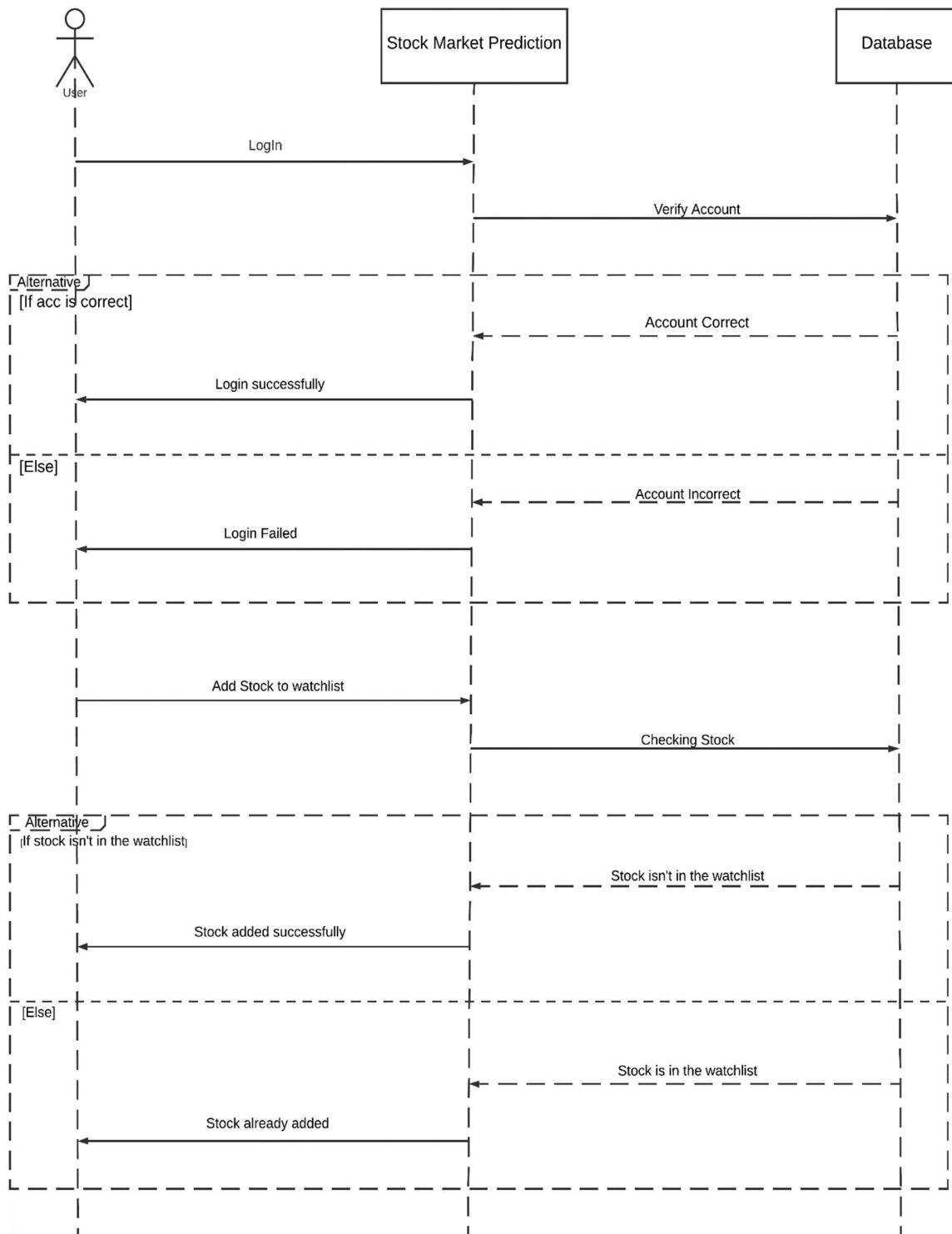
Use case name	News Page
Actors	User , News
Pre Conditions	User must be logged in to view news.
Post Conditions	None
Flow of events	User can scroll up and down to see the news related to the stocks he is watching.

Use case name	Watchlist
Actors	User, Watchlist
Pre Conditions	User must be logged in to add or remove stocks from the watchlist
Post Conditions	User can't add stock already exists in the watchlist.
Flow of events	<ul style="list-style-type: none"> • User can see his favorite stocks in the watchlist. • Add new stocks to the watchlist. • Remove stocks from the watchlist.

3.Full view of the application:

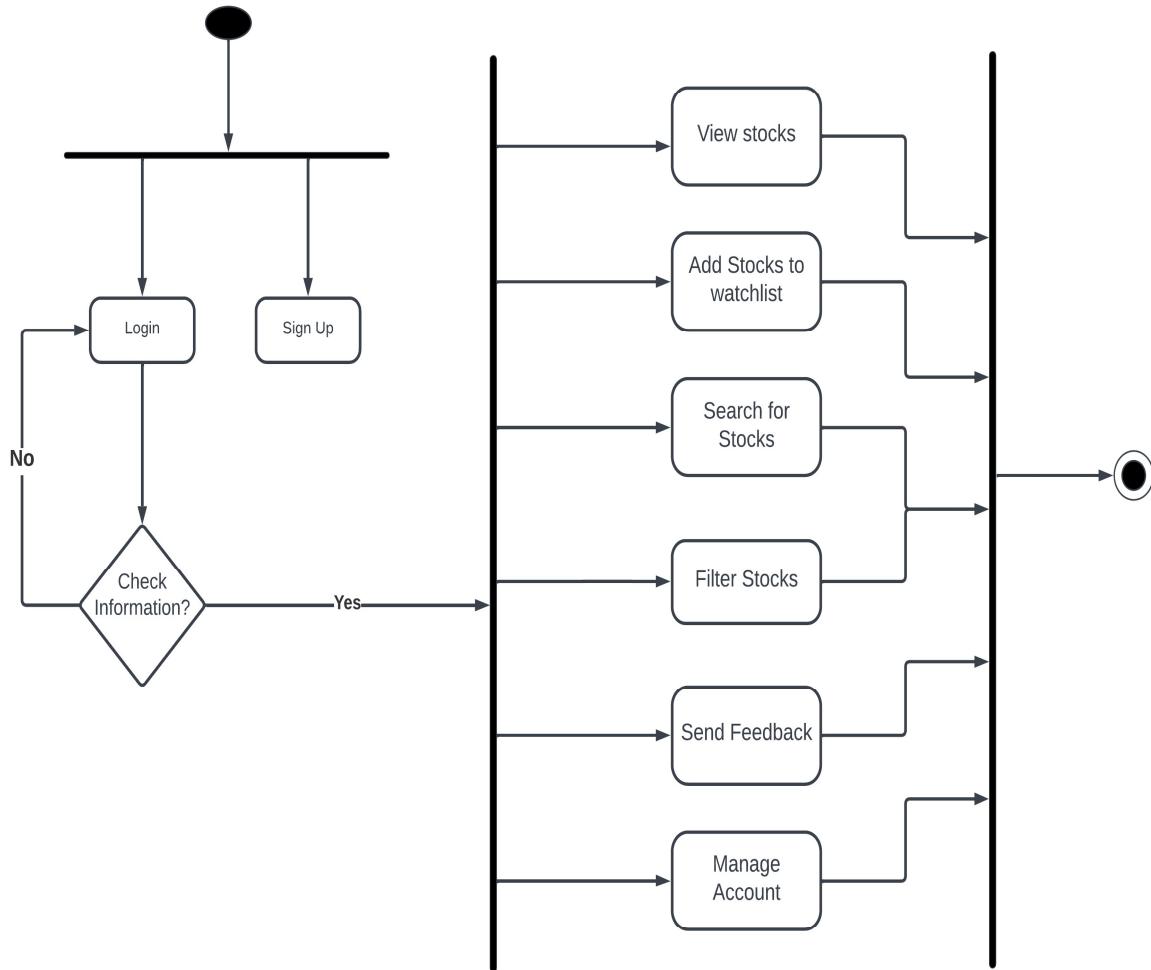


3.4 sequence diagram:





3.5 activity diagram:



Chapter 4: Implementation:

In this chapter we're going to discuss and go deeper in our implementation and present its code and the algorithms used to build it.

4.1 Software Tools:

Flutter : Flutter is an open-source mobile application development framework created by Google. It allows developers to build high-performance, cross-platform mobile applications for both Android and iOS using a single codebase.

Flutter uses the Dart programming language, which was also developed by Google, and provides a rich set of pre-built widgets and tools to help developers create visually appealing and responsive mobile applications. The framework also includes a hot reload feature that allows developers to make changes to their code and see the results in real-time, making the development process faster and more efficient.

Python: Python is a popular high-level programming language known for its simplicity, ease of use, and versatility. It was first released in 1991 by Guido van Rossum and has since become one of the most widely used programming languages in the world.

Python is also an object-oriented language, which means that it allows developers to define and manipulate objects with properties and methods. This makes it well-suited for tasks such as building complex data structures and creating reusable code used in a wide range of applications, from web development and data science to artificial intelligence and machine learning.

TensorFlow: TensorFlow is a popular high-level programming language known for its simplicity, ease of use, and versatility. It was first released in 1991 by Guido van Rossum and has since become one of the most widely used programming languages in the world.

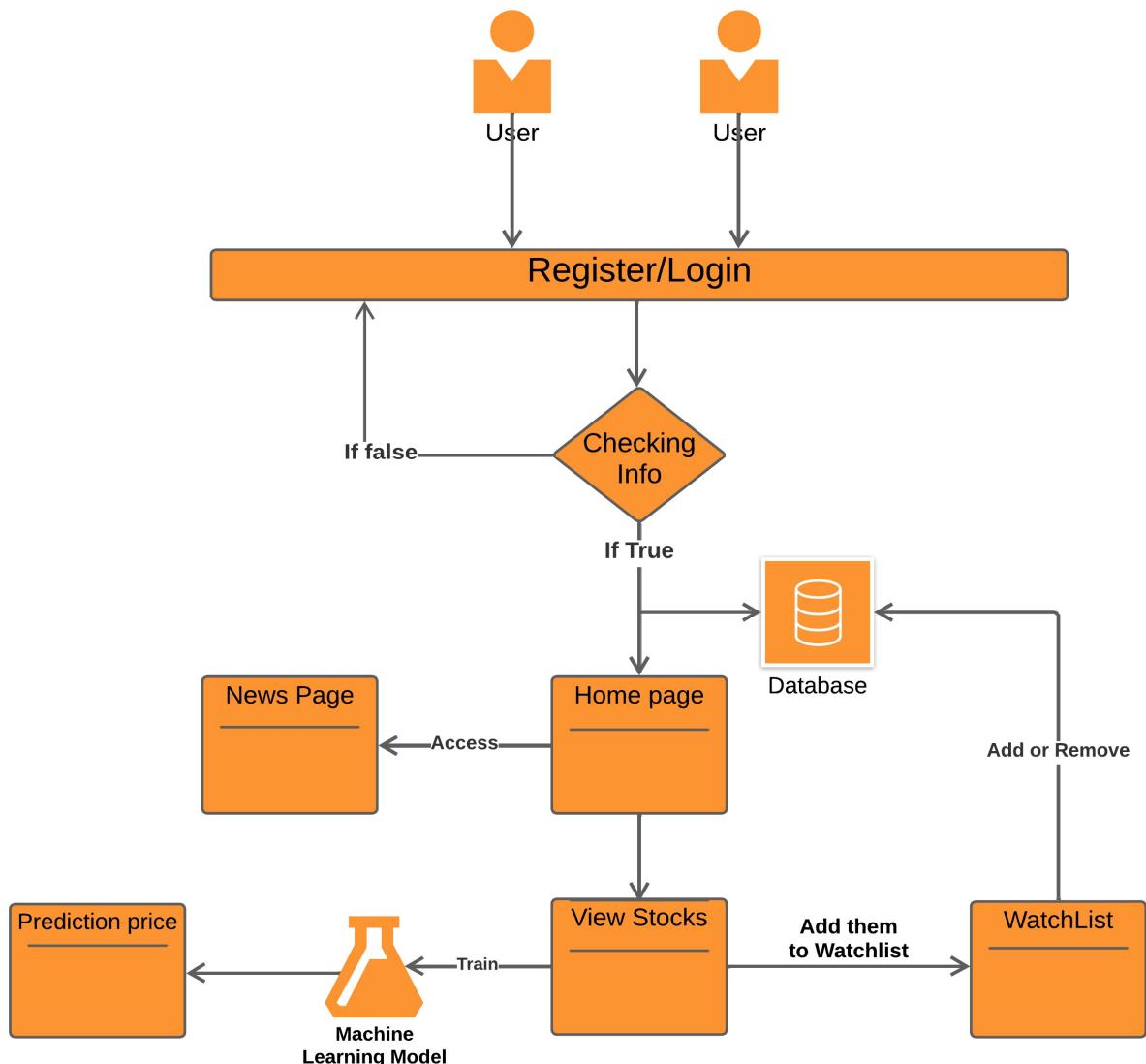
It's provides a wide range of pre-built functions and modules for building ML models, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs). Additionally, TensorFlow includes a high-level API called Keras, which allows developers to build and train ML models with a simple and intuitive interface.

Firebase: Firebase is a mobile and web application development platform, which was acquired by Google in 2014. It provides developers with a suite of

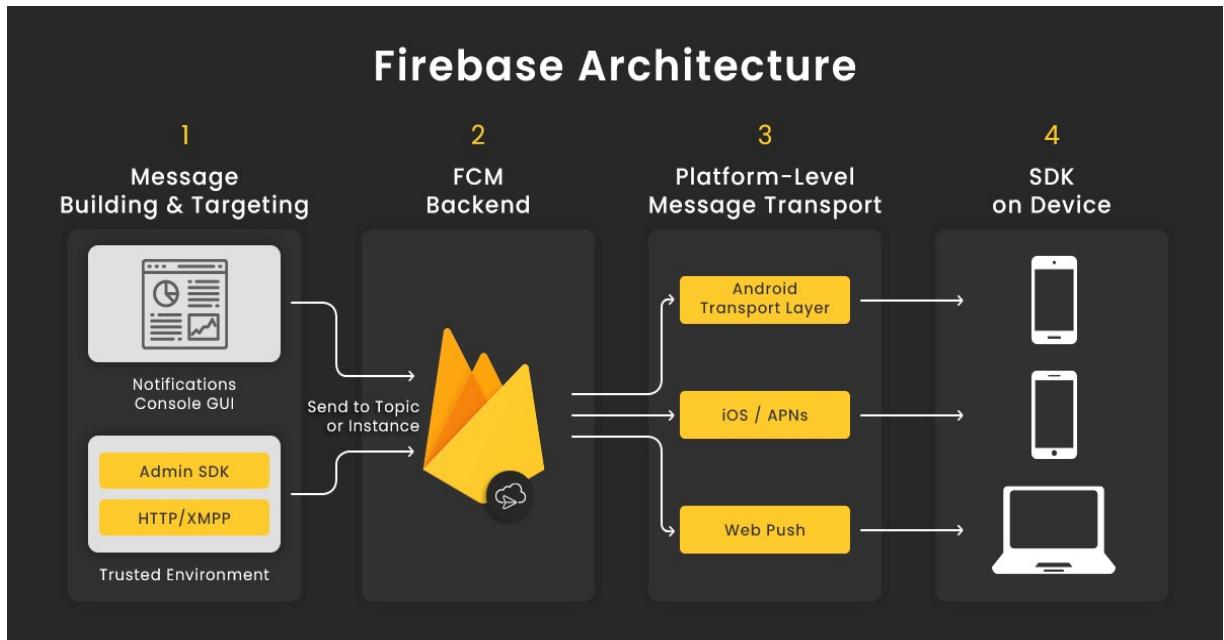
tools and services to help them build high-quality apps, improve user engagement, and grow their business.

its real-time database, which allows developers to build apps that update in real-time without needing to write complex server-side code. Firebase also provides a variety of authentication options, including email/password, Google, Facebook, and Twitter, which makes it easy for developers to add authentication to their apps.

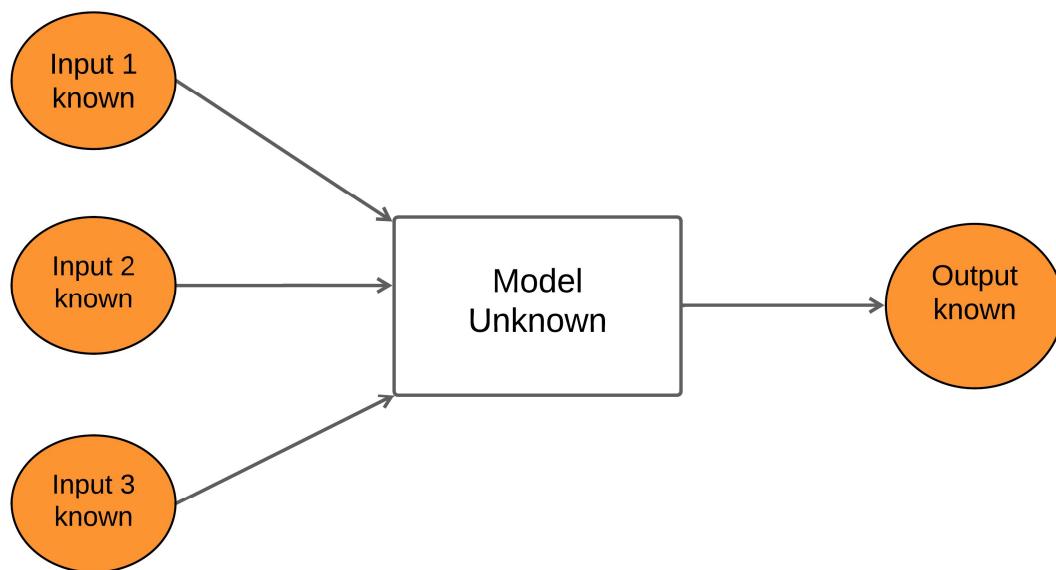
4.2 Software architecture:



Firebase architecture:

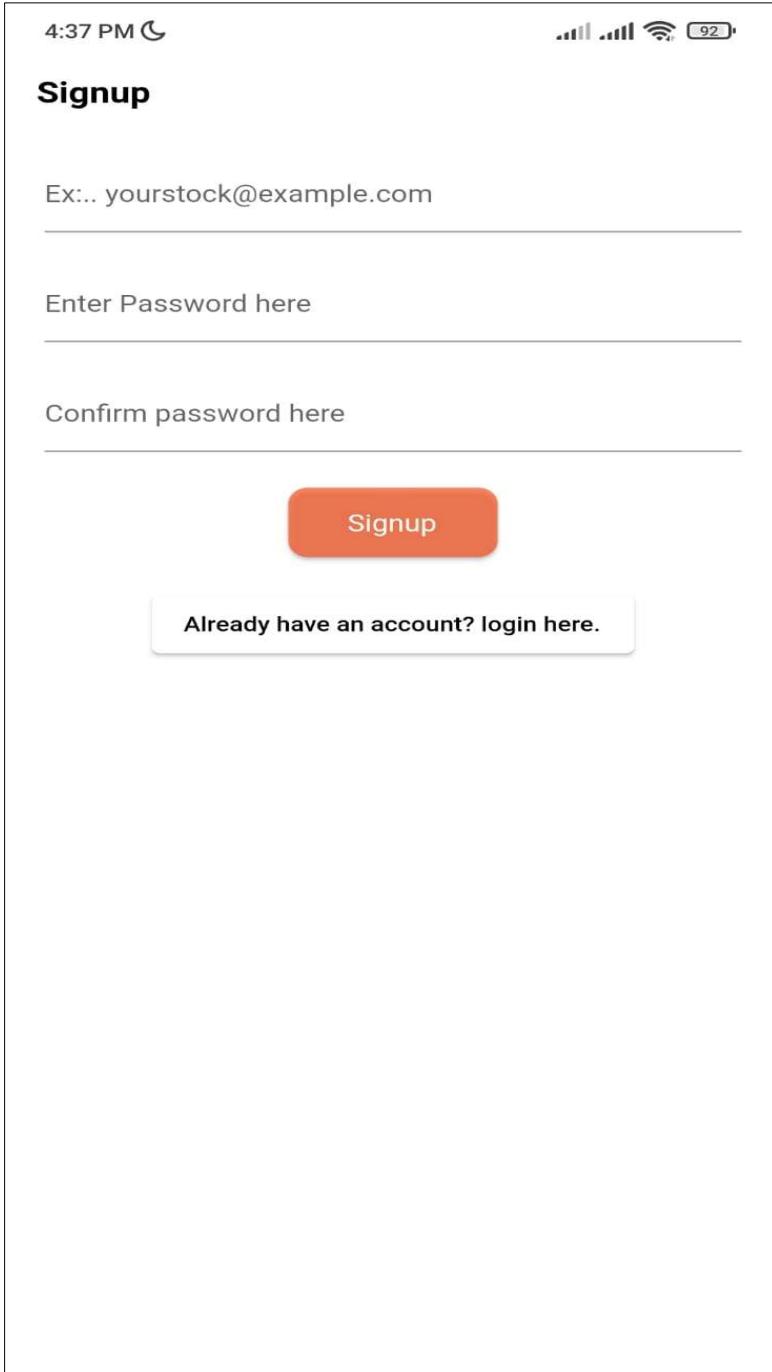


Machine Learning Model Diagram:



4.3 User Interface (Screenshots):

Sign up page:



4:37 PM ⌂

Signup

Ex... yourstock@example.com

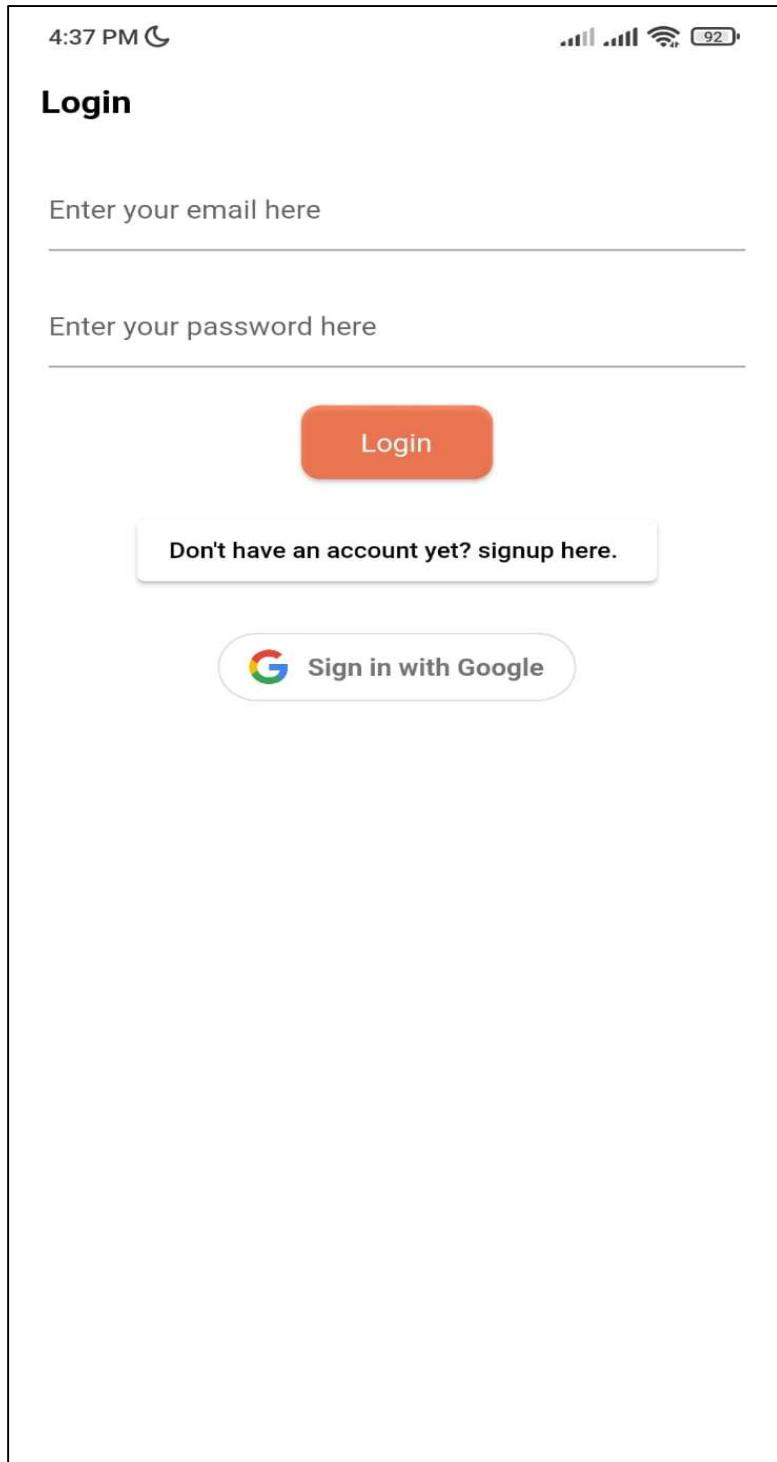
Enter Password here

Confirm password here

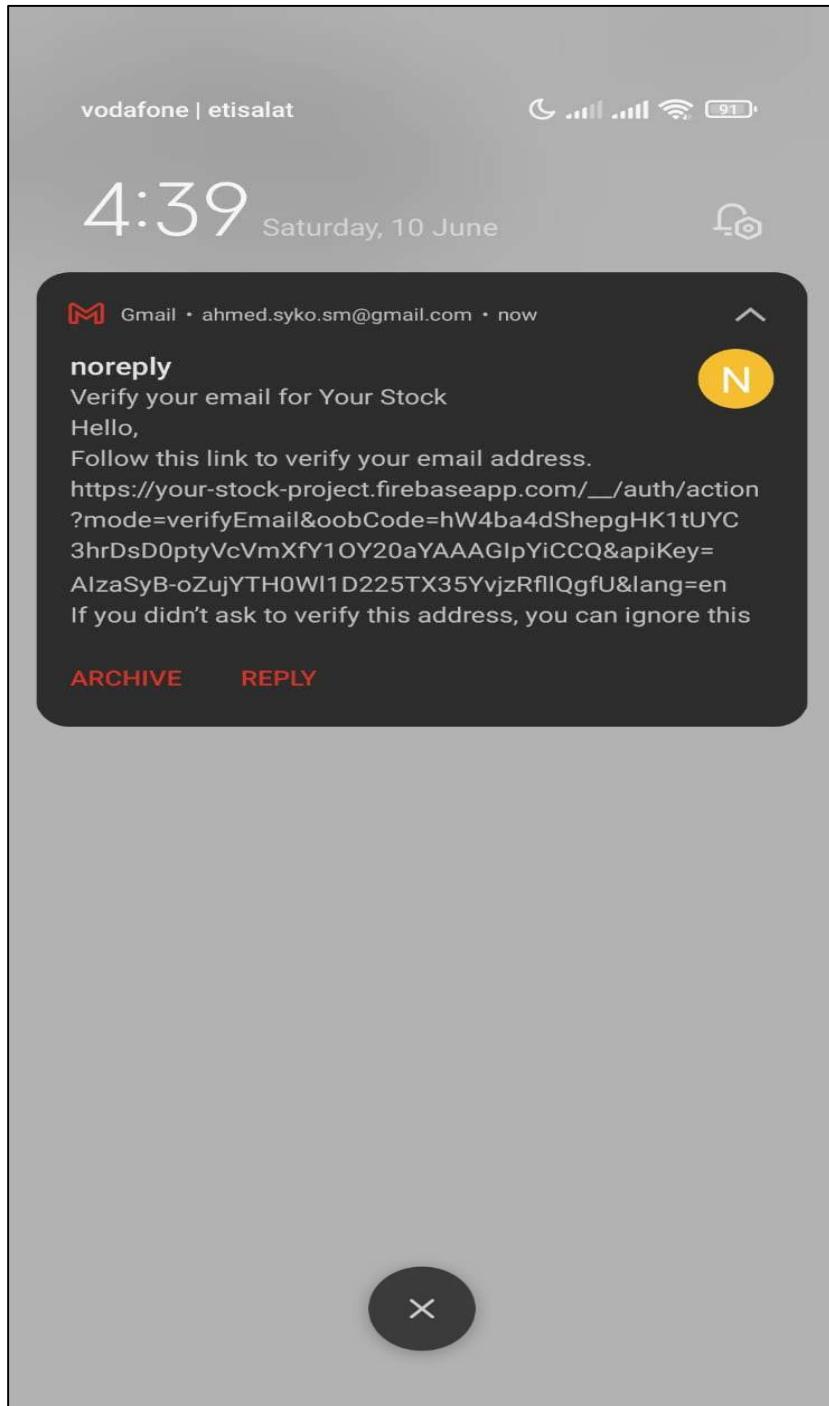
Signup

Already have an account? [login here.](#)

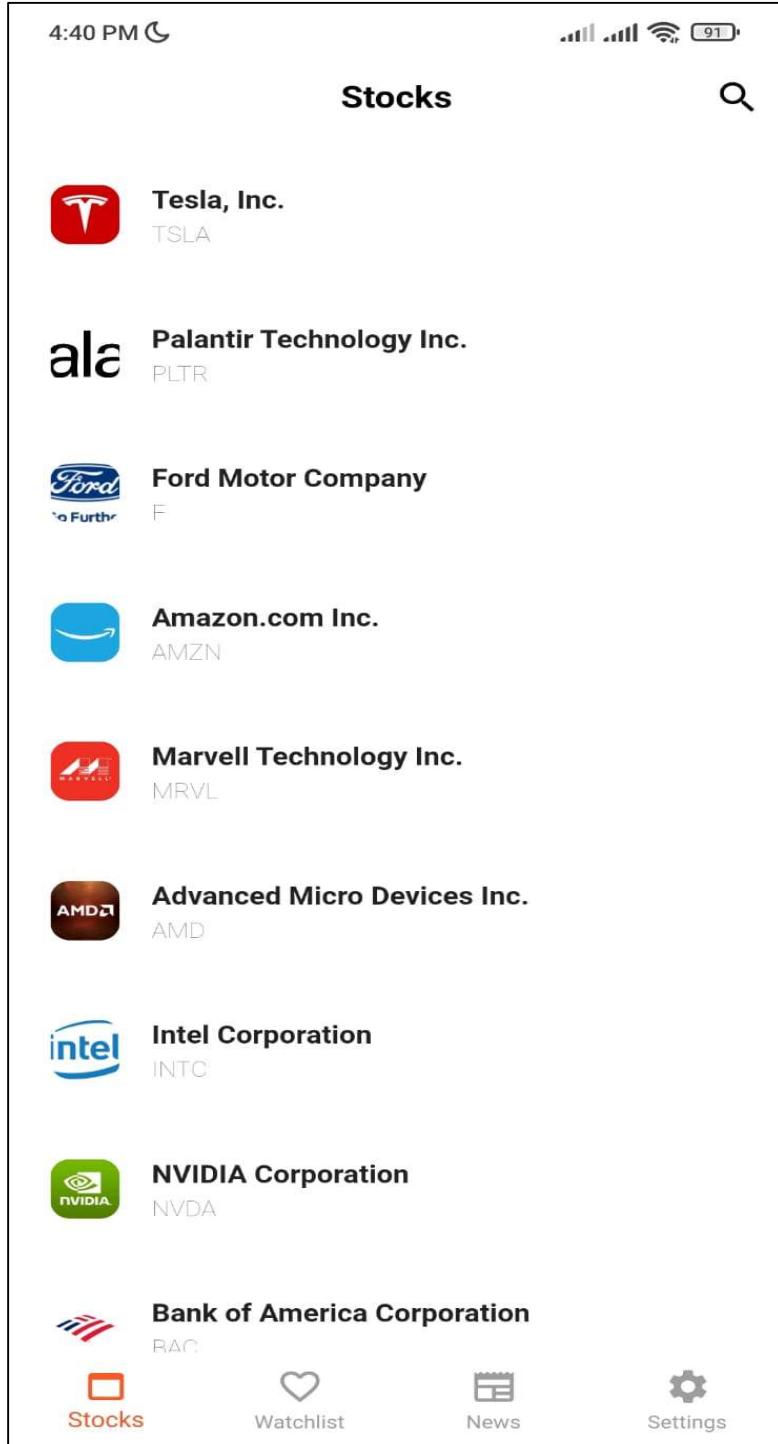
Log In Page:



Verification (Gmail sending an e-mail to verify your account):



Home page (Suggested stocks):

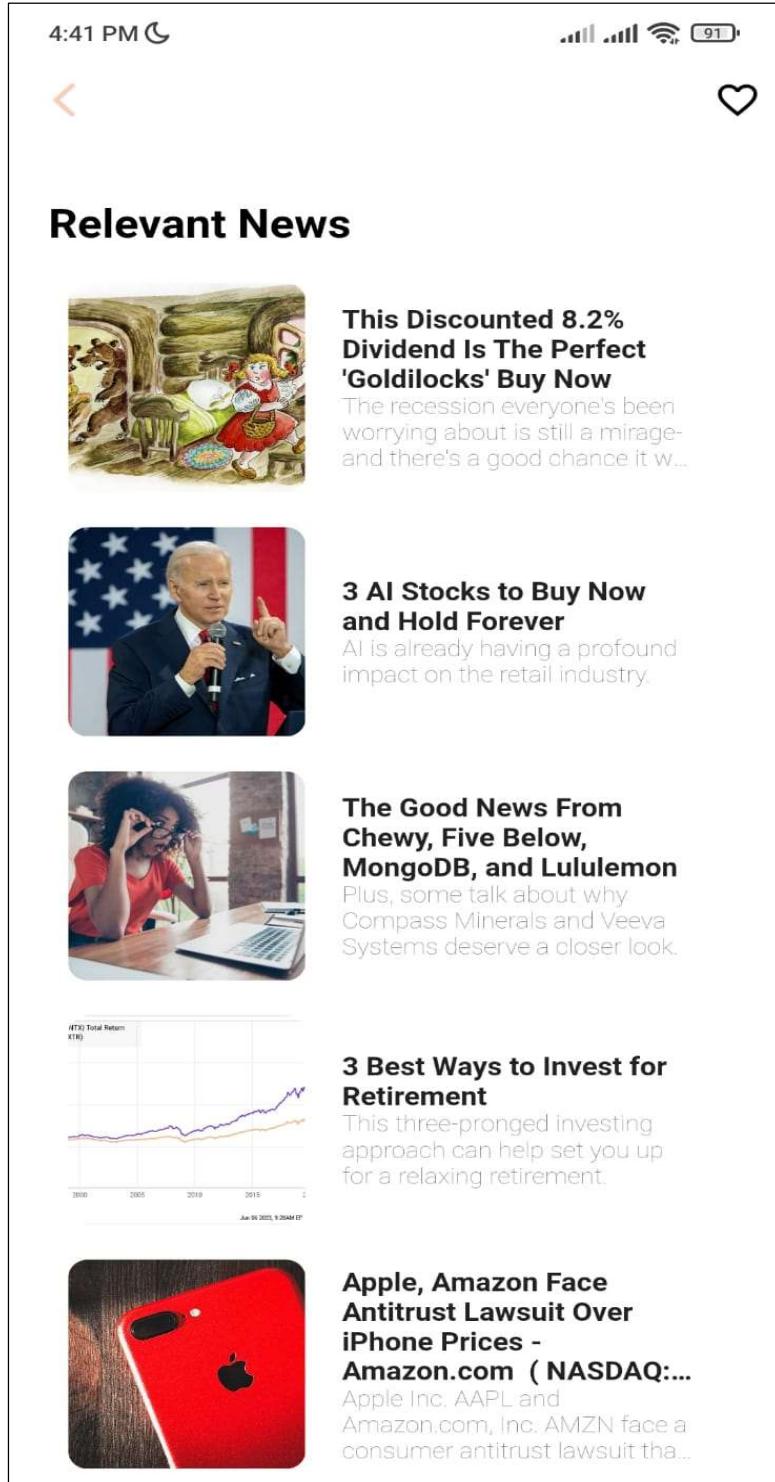


The screenshot shows the YourStock mobile application interface. At the top, it displays the time as 4:40 PM and battery level at 91%. The title "Stocks" is centered above a search bar with a magnifying glass icon. Below the search bar, there is a list of suggested stocks, each with a small company logo and the stock symbol.

Stock Logo	Company Name	Stock Symbol
Tesla, Inc. (TSLA)	Tesla, Inc.	TSLA
Palantir Technology Inc. (PLTR)	Palantir Technology Inc.	PLTR
Ford Motor Company (F)	Ford Motor Company	F
Amazon.com Inc. (AMZN)	Amazon.com Inc.	AMZN
Marvell Technology Inc. (MRVL)	Marvell Technology Inc.	MRVL
Advanced Micro Devices Inc. (AMD)	Advanced Micro Devices Inc.	AMD
Intel Corporation (INTC)	Intel Corporation	INTC
NVIDIA Corporation (NVDA)	NVIDIA Corporation	NVDA
Bank of America Corporation (BAC)	Bank of America Corporation	BAC

At the bottom of the screen, there are four navigation icons: "Stocks" (orange square), "Watchlist" (heart icon), "News" (newspaper icon), and "Settings" (gear icon).

News Page:



4:41 PM                 

Relevant News

This Discounted 8.2% Dividend Is The Perfect 'Goldilocks' Buy Now

The recession everyone's been worrying about is still a mirage—and there's a good chance it w...

3 AI Stocks to Buy Now and Hold Forever

AI is already having a profound impact on the retail industry.

The Good News From Chewy, Five Below, MongoDB, and Lululemon

Plus, some talk about why Compass Minerals and Veeva Systems deserve a closer look.

HTF Total Return (%)

Year	Total Return (%)
2000	~10
2005	~15
2010	~20
2015	~25
2020	~35
2021	~40

Jun 06 2021, 9:30AM ET

3 Best Ways to Invest for Retirement

This three-pronged investing approach can help set you up for a relaxing retirement.

Apple, Amazon Face Antitrust Lawsuit Over iPhone Prices - Amazon.com (NASDAQ:...

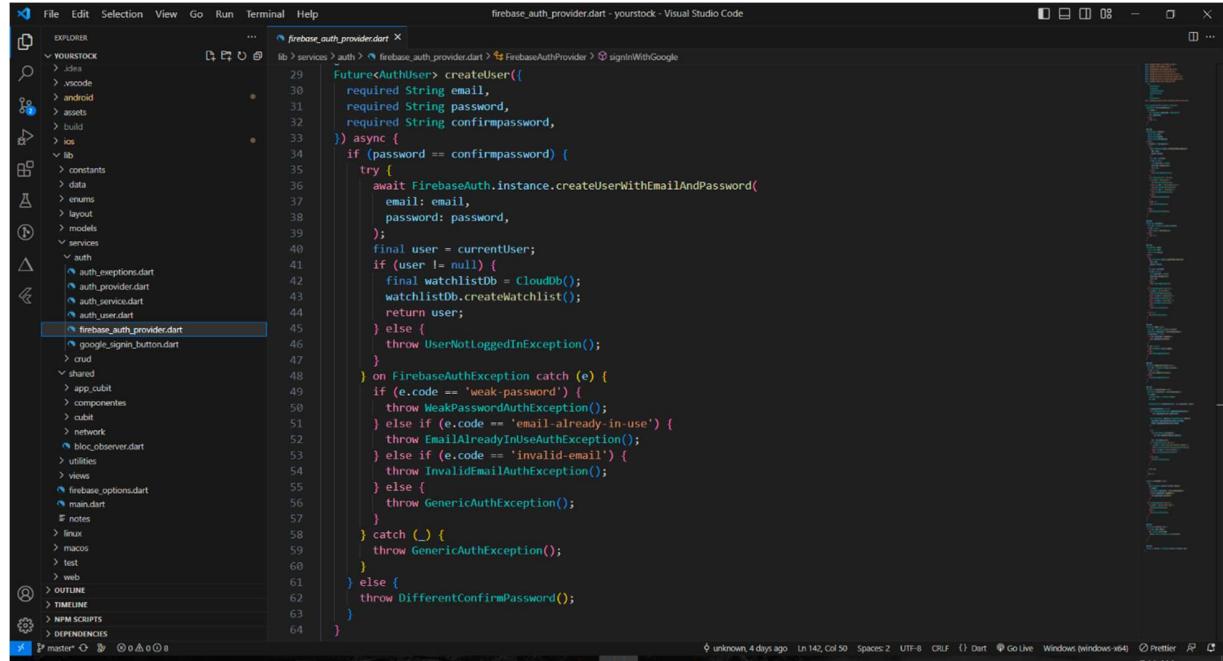
Apple Inc. AAPL and Amazon.com, Inc. AMZN face a consumer antitrust lawsuit tha...

Stock view:



4.4 Code Example Screenshots:

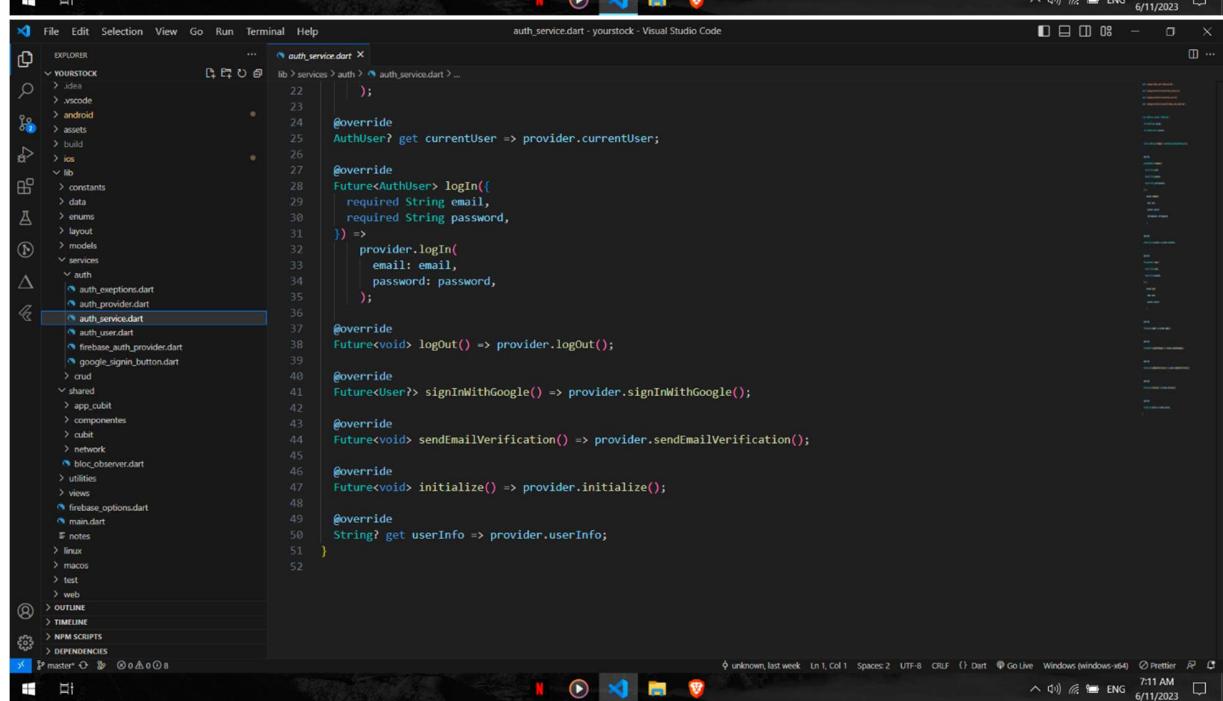
User Register & login:



```

firebase_auth_provider.dart
29 Future<AuthUser> createUser({
30   required String email,
31   required String password,
32   required String confirmPassword,
33 }) async {
34   if (password == confirmPassword) {
35     try {
36       await FirebaseAuth.instance.createUserWithEmailAndPassword(
37         email: email,
38         password: password,
39       );
39       final user = currentUser;
40       if (user != null) {
41         final watchlistDb = CloudDb();
42         watchlistDb.createWatchlist();
43         return user;
44       } else {
45         throw UserNotLoggedInException();
46       }
47     } on FirebaseAuthException catch (e) {
48       if (e.code == 'weak-password') {
49         throw WeakPasswordAuthException();
50       } else if (e.code == 'email-already-in-use') {
51         throw EmailAlreadyInUseAuthException();
52       } else if (e.code == 'invalid-email') {
53         throw InvalidEmailAuthException();
54       } else {
55         throw GenericAuthException();
56       }
57     } catch (_) {
58       throw GenericAuthException();
59     }
60   } else {
61     throw DifferentConfirmPassword();
62   }
63 }

```

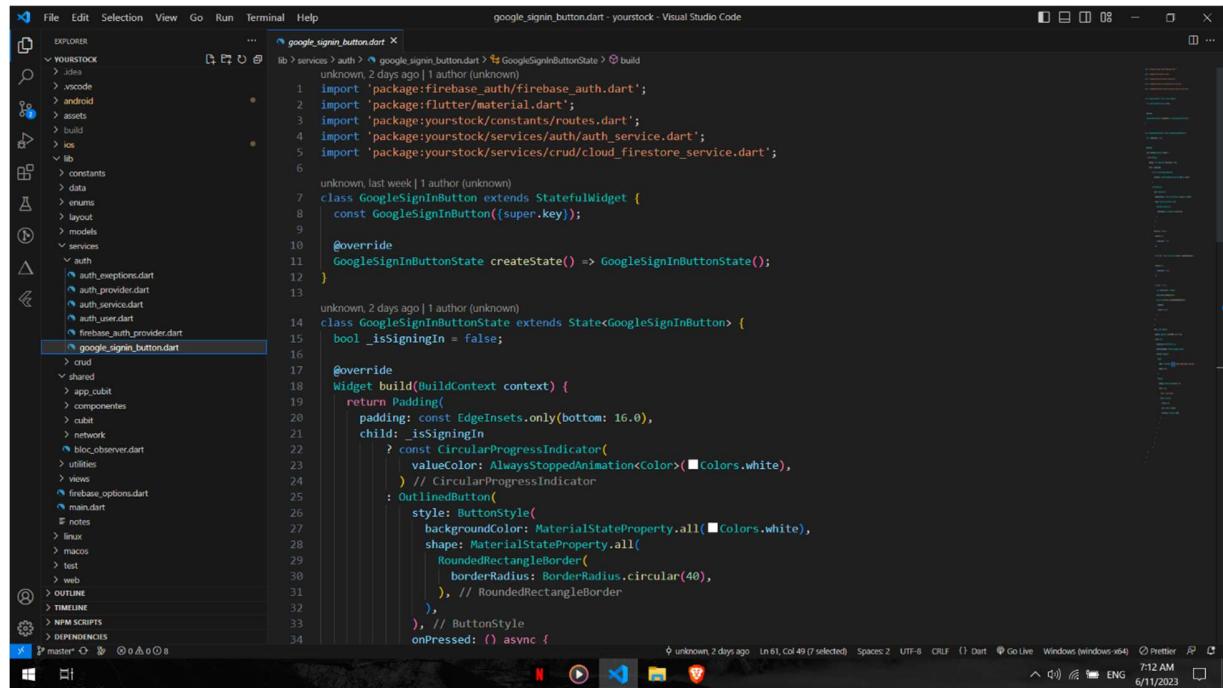


```

auth_service.dart
22 );
23
24 @override
25 AuthUser? get currentUser => provider.currentUser;
26
27 @override
28 Future<AuthUser> logIn({
29   required String email,
30   required String password,
31 }) =>
32   provider.logIn(
33     email: email,
34     password: password,
35   );
36
37 @override
38 Future<void> logOut() => provider.logOut();
39
40 @override
41 Future<User?> signInWithGoogle() => provider.signInWithGoogle();
42
43 @override
44 Future<void> sendEmailVerification() => provider.sendEmailVerification();
45
46 @override
47 Future<void> initialize() => provider.initialize();
48
49 @override
50 String? get userInfo => provider.userInfo;
51

```

Log In with Google:



```

File Edit Selection View Go Run Terminal Help
... google_signin_button.dart ...
EXPLORER lib > services > auth > google_signin_button.dart > build
1 import 'package:firebase_auth/firebase_auth.dart';
2 import 'package:flutter/material.dart';
3 import 'package:yourstock/constants/routes.dart';
4 import 'package:yourstock/services/auth/auth_service.dart';
5 import 'package:yourstock/services/crud/cloud_firestore_service.dart';
6
7 unknown, 2 days ago | 1 author (unknown)
8 class GoogleSignInButton extends StatefulWidget {
9   const GoogleSignInButton({super.key});
10
11   @override
12     GoogleSignInButtonState createState() => GoogleSignInButtonState();
13 }
14
15 unknown, 2 days ago | 1 author (unknown)
16 class GoogleSignInButtonState extends State<GoogleSignInButton> {
17   bool _isSigningIn = false;
18
19   @override
20     Widget build(BuildContext context) {
21       return Padding(
22         padding: const EdgeInsets.only(bottom: 16.0),
23         child: _isSigningIn
24           ? const CircularProgressIndicator(
25             valueColor: AlwaysStoppedAnimation<Color>(Colors.white),
26           ) // CircularProgressIndicator
27           : OutlinedButton(
28             style: ButtonStyle(
29               backgroundColor: MaterialStateProperty.all(Colors.white),
30               shape: MaterialStateProperty.all(
31                 RoundedRectangleBorder(
32                   borderRadius: BorderRadius.circular(40),
33                 ), // RoundedRectangleBorder
34               ),
35             ),
36             onPressed: () async {
37               ...
38             }
39           );
40       }
41     }
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160

```

File Edit Selection View Go Run Terminal Help

firestore_auth_provider.dart - yourstock - Visual Studio Code

EXPLORER lib > services > auth > firebase_auth_provider.dart > build > signInWithGoogle

```

@override
Future<void> sendEmailVerification() async {
  final user = FirebaseAuth.instance.currentUser;
  if (user != null) {
    await user.sendEmailVerification();
  } else {
    throw UserNotLoggedInException();
  }
}

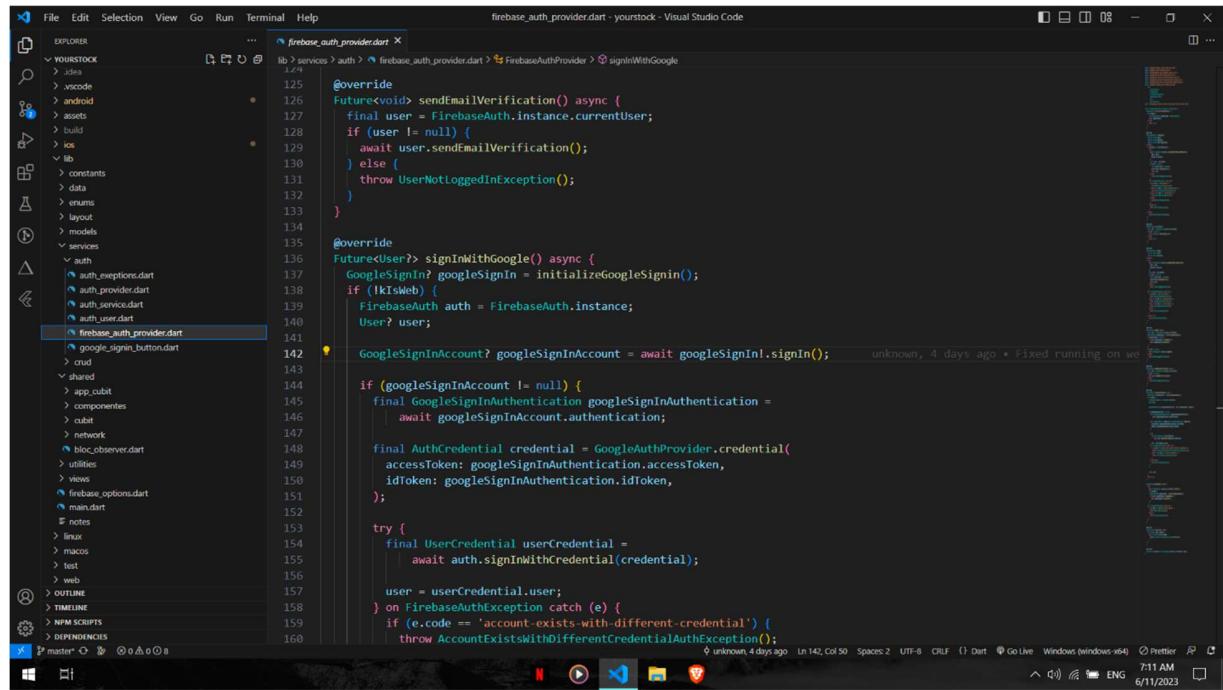
@Override
Future<User> signInWithGoogle() async {
  GoogleSignIn? googleSignIn = initializeGoogleSignIn();
  if (!kIsWeb) {
    FirebaseAuth auth = FirebaseAuth.instance;
    User? user;
    GoogleSignInAccount? googleSignInAccount = await googleSignIn.signIn();
    if (googleSignInAccount != null) {
      final GoogleSignInAuthentication googleSignInAuthentication =
          await googleSignInAccount.authentication;
      final AuthCredential credential = GoogleAuthProvider.credential(
        accessToken: googleSignInAuthentication.accessToken,
        idToken: googleSignInAuthentication.idToken,
      );
      try {
        final UserCredential userCredential =
            await auth.signInWithCredential(credential);
        user = userCredential.user;
      } on FirebaseAuthException catch (e) {
        if (e.code == 'account-exists-with-different-credential') {
          throw AccountExistsWithDifferentCredentialAuthException();
        }
      }
    }
  }
}

```

File Edit Selection View Go Run Terminal Help

firestore_auth_provider.dart - yourstock - Visual Studio Code

EXPLORER lib > services > auth > firebase_auth_provider.dart > build > signInWithGoogle



```

@override
Future<void> sendEmailVerification() async {
  final user = FirebaseAuth.instance.currentUser;
  if (user != null) {
    await user.sendEmailVerification();
  } else {
    throw UserNotLoggedInException();
  }
}

@Override
Future<User> signInWithGoogle() async {
  GoogleSignIn? googleSignIn = initializeGoogleSignIn();
  if (!kIsWeb) {
    FirebaseAuth auth = FirebaseAuth.instance;
    User? user;
    GoogleSignInAccount? googleSignInAccount = await googleSignIn.signIn();
    if (googleSignInAccount != null) {
      final GoogleSignInAuthentication googleSignInAuthentication =
          await googleSignInAccount.authentication;
      final AuthCredential credential = GoogleAuthProvider.credential(
        accessToken: googleSignInAuthentication.accessToken,
        idToken: googleSignInAuthentication.idToken,
      );
      try {
        final UserCredential userCredential =
            await auth.signInWithCredential(credential);
        user = userCredential.user;
      } on FirebaseAuthException catch (e) {
        if (e.code == 'account-exists-with-different-credential') {
          throw AccountExistsWithDifferentCredentialAuthException();
        }
      }
    }
  }
}

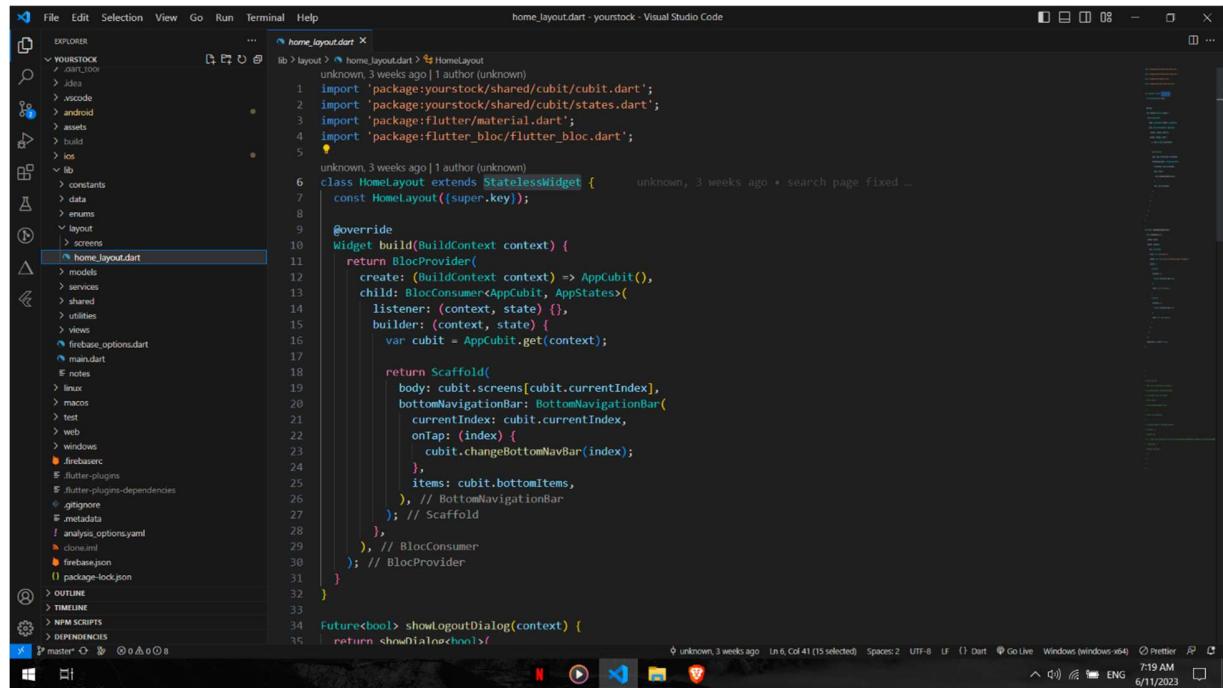
```

File Edit Selection View Go Run Terminal Help

firestore_auth_provider.dart - yourstock - Visual Studio Code

EXPLORER lib > services > auth > firebase_auth_provider.dart > build > signInWithGoogle

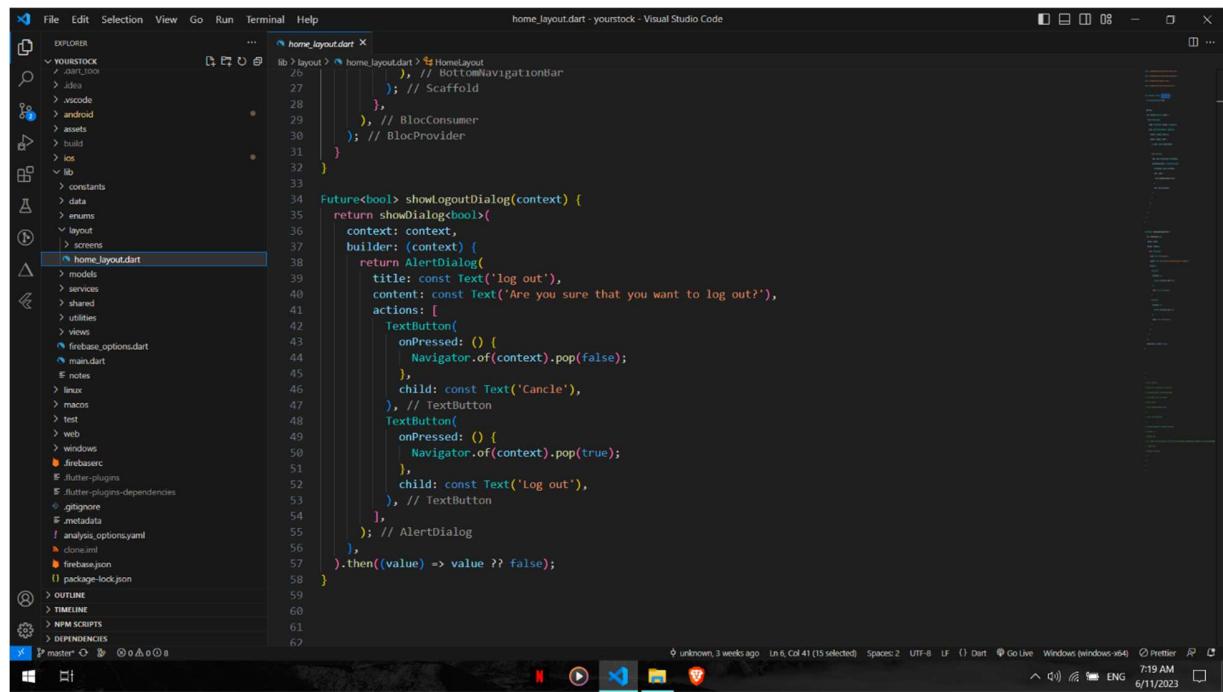
Home Page:



```

File Edit Selection View Go Run Terminal Help
... home_layout.dart ...
lib > layout > home_layout.dart > HomeLayout
unknown, 3 weeks ago | 1 author (unknown)
1 import 'package:yourstock/shared/cubit/cubits.dart';
2 import 'package:yourstock/shared/cubit/states.dart';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_bloc/flutter_bloc.dart';
5
6 unknown, 3 weeks ago | 1 author (unknown)
7 class HomeLayout extends StatelessWidget {
8     const HomeLayout({super.key});
9
10 @override
11     Widget build(BuildContext context) {
12         return BlocProvider(
13             create: (BuildContext context) => AppCubit(),
14             child: BlocConsumer<AppCubit, AppState>(
15                 listener: (context, state) {},
16                 builder: (context, state) {
17                     var cubit = AppCubit.get(context);
18
19                     return Scaffold(
20                         body: cubit.screens[cubit.currentIndex],
21                         bottomNavigationBar: BottomNavigationBar(
22                             currentIndex: cubit.currentIndex,
23                             onTap: (index) {
24                                 cubit.changeBottomNavBar(index);
25                             },
26                             items: cubit.bottomItems,
27                         ), // BottomNavigationBar
28                     ); // Scaffold
29                 },
30             ); // BlocConsumer
31         );
32     }
33
34 Future<bool> showLogoutDialog(BuildContext context) {
35     return showDialog<bool>(
36         context: context,
37         builder: (context) {
38             return AlertDialog(
39                 title: const Text('Log out'),
40                 content: const Text('Are you sure that you want to log out?'),
41                 actions: [
42                     TextButton(
43                         onPressed: () {
44                             Navigator.of(context).pop(false);
45                         },
46                         child: const Text('Cancel'),
47                     ), // TextButton
48                     TextButton(
49                         onPressed: () {
50                             Navigator.of(context).pop(true);
51                         },
52                         child: const Text('Log out'),
53                     ), // TextButton
54                 ],
55             ); // AlertDialog
56         },
57     ).then((value) => value ?? false);
58 }
59
60
61
62

```

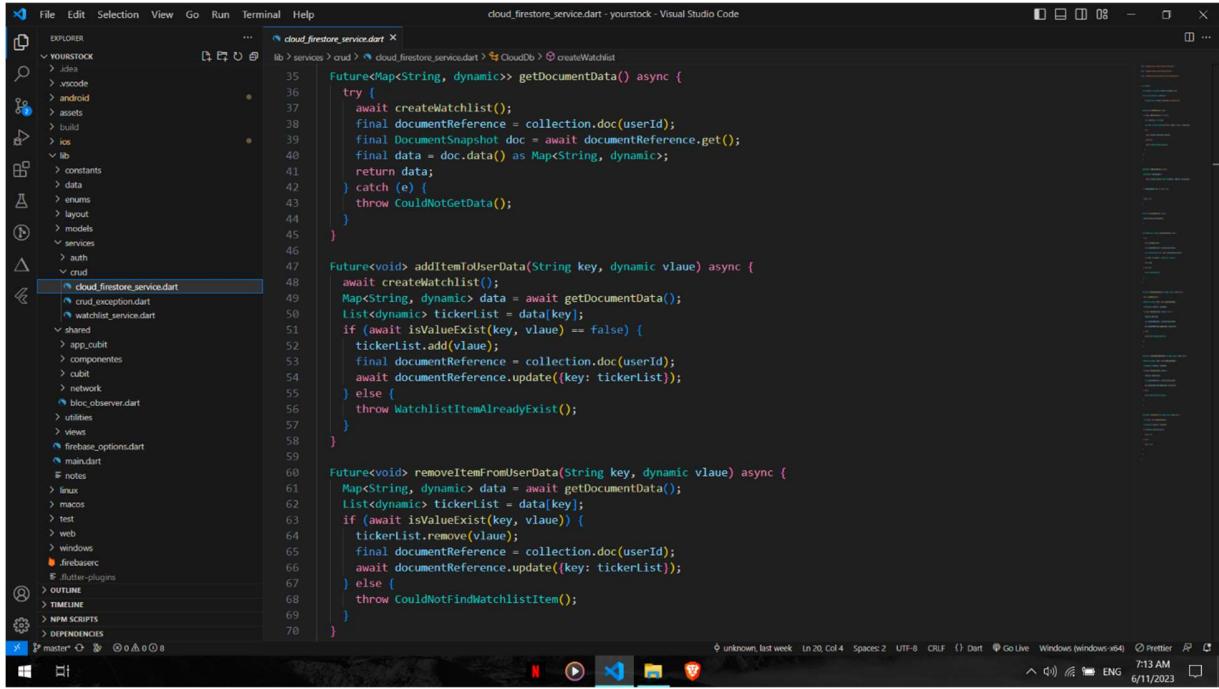


```

File Edit Selection View Go Run Terminal Help
... home_layout.dart ...
lib > layout > home_layout.dart > HomeLayout
unknown, 3 weeks ago | 1 author (unknown)
1 import 'package:yourstock/shared/cubit/cubits.dart';
2 import 'package:yourstock/shared/cubit/states.dart';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_bloc/flutter_bloc.dart';
5
6 unknown, 3 weeks ago | 1 author (unknown)
7 class HomeLayout extends StatelessWidget {
8     const HomeLayout({super.key});
9
10 @override
11     Widget build(BuildContext context) {
12         return BlocProvider(
13             create: (BuildContext context) => AppCubit(),
14             child: BlocConsumer<AppCubit, AppState>(
15                 listener: (context, state) {},
16                 builder: (context, state) {
17                     var cubit = AppCubit.get(context);
18
19                     return Scaffold(
20                         body: cubit.screens[cubit.currentIndex],
21                         bottomNavigationBar: BottomNavigationBar(
22                             currentIndex: cubit.currentIndex,
23                             onTap: (index) {
24                                 cubit.changeBottomNavBar(index);
25                             },
26                             items: cubit.bottomItems,
27                         ), // BottomNavigationBar
28                     ); // Scaffold
29                 },
30             ); // BlocConsumer
31         );
32     }
33
34 Future<bool> showLogoutDialog(BuildContext context) {
35     return showDialog<bool>(
36         context: context,
37         builder: (context) {
38             return AlertDialog(
39                 title: const Text('Log out'),
40                 content: const Text('Are you sure that you want to log out?'),
41                 actions: [
42                     TextButton(
43                         onPressed: () {
44                             Navigator.of(context).pop(false);
45                         },
46                         child: const Text('Cancel'),
47                     ), // TextButton
48                     TextButton(
49                         onPressed: () {
50                             Navigator.of(context).pop(true);
51                         },
52                         child: const Text('Log out'),
53                     ), // TextButton
54                 ],
55             ); // AlertDialog
56         },
57     ).then((value) => value ?? false);
58 }
59
60
61
62

```

Add & remove stocks from Watchlist:



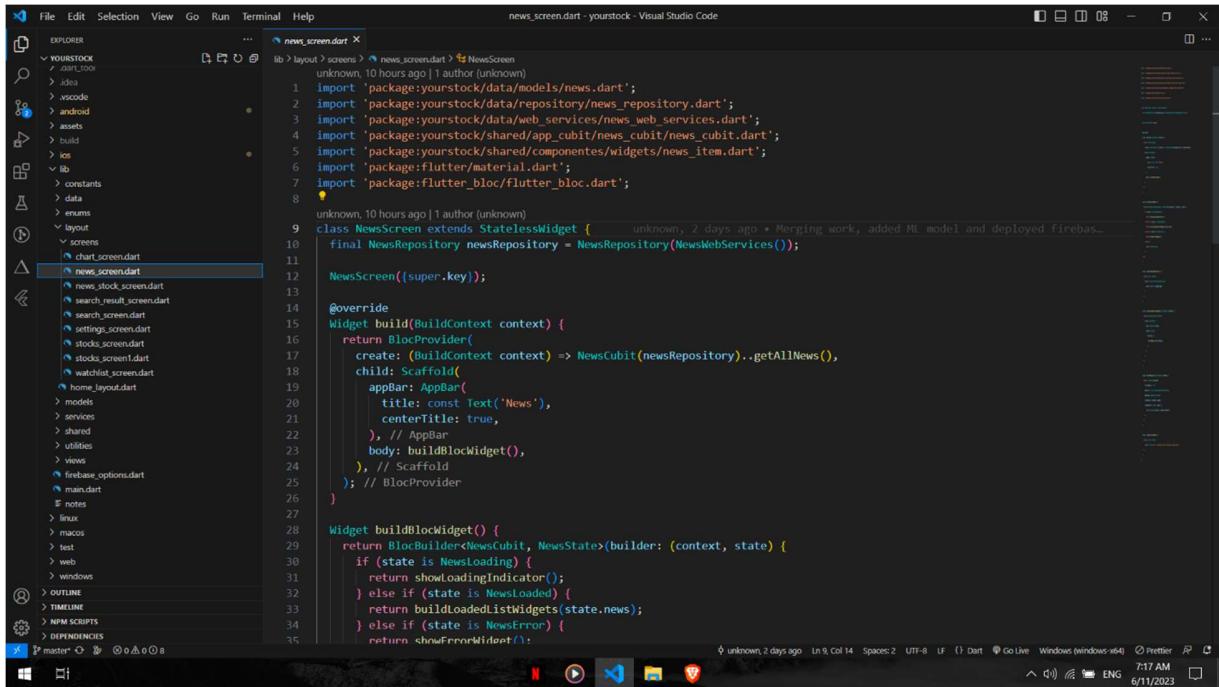
```

File Edit Selection View Go Run Terminal Help
... cloud_firestore_service.dart ...
lib> services> crud> cloud_firestore.service.dart > CloudDb > createWatchlist
35 Future<Map<String, dynamic>> getDocumentData() async {
36   try {
37     await createWatchlist();
38     final documentReference = collection.doc(userId);
39     final DocumentSnapshot doc = await documentReference.get();
40     final data = doc.data() as Map<String, dynamic>;
41     return data;
42   } catch (e) {
43     throw CouldNotGetData();
44   }
45 }
46
47 Future<void> addNewItemToUserData(String key, dynamic value) async {
48   await createWatchlist();
49   Map<String, dynamic> data = await getDocumentData();
50   List<dynamic> tickerList = data[key];
51   if (await isValueExist(key, value) == false) {
52     tickerList.add(value);
53     final documentReference = collection.doc(userId);
54     await documentReference.update({key: tickerList});
55   } else {
56     throw WatchlistItemAlreadyExist();
57   }
58 }
59
60 Future<void> removeItemFromUserData(String key, dynamic value) async {
61   Map<String, dynamic> data = await getDocumentData();
62   List<dynamic> tickerList = data[key];
63   if (await isValueExist(key, value)) {
64     tickerList.remove(value);
65     final documentReference = collection.doc(userId);
66     await documentReference.update({key: tickerList});
67   } else {
68     throw CouldNotFindWatchlistItem();
69   }
70 }

```

unknown, last week | ln 20 Col 4 | Spaces: 2 | UTF-8 | CR LF | Dart | Go Live | Windows (Windows-x64) | Prettier | ENG | 7:13 AM | 6/11/2023

News Page:

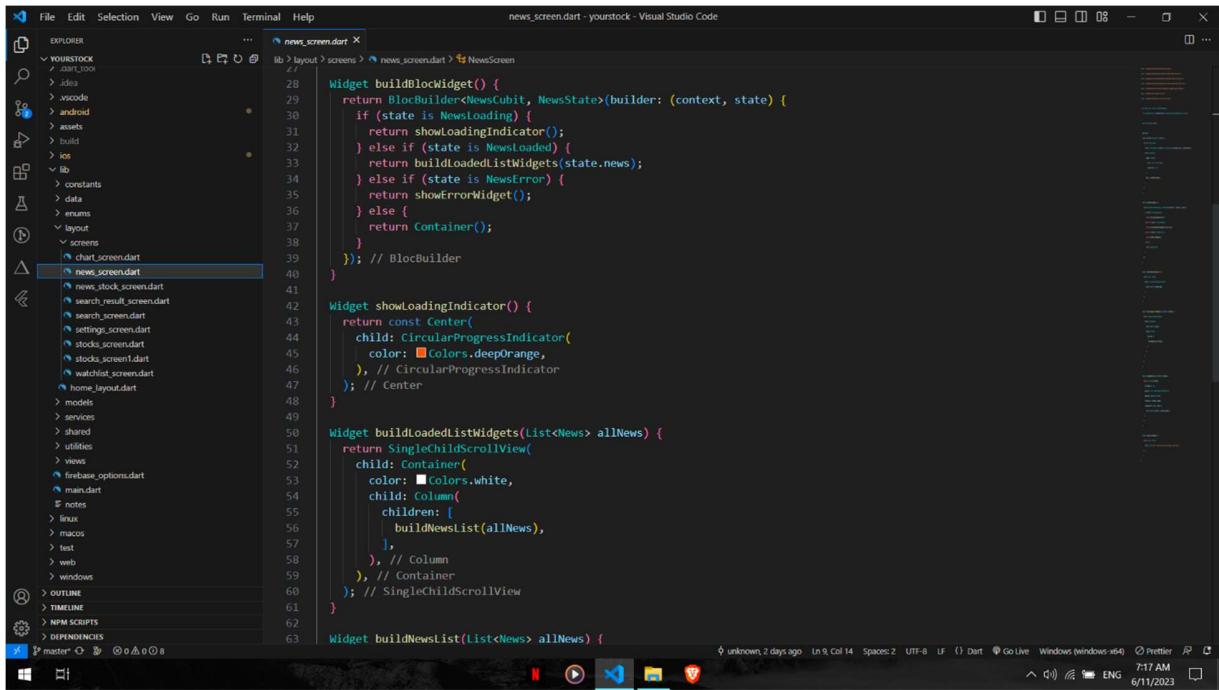


```

File Edit Selection View Go Run Terminal Help
... news_screen.dart ...
lib> layout> screens> news_screen.dart > NewsScreen
unknown, 10 hours ago | 1 author (unknown)
1 import 'package:yourstock/data/models/news.dart';
2 import 'package:yourstock/data/repository/news_repository.dart';
3 import 'package:yourstock/data/web_services/news_web_services.dart';
4 import 'package:yourstock/shared/app_cubit/news_cubit/news_cubit.dart';
5 import 'package:yourstock/shared/components/widgets/news_item.dart';
6 import 'package:flutter/material.dart';
7 import 'package:flutter_bloc/flutter_bloc.dart';
8
9
10 unknown, 10 hours ago | 1 author (unknown)
11 class NewsScreen extends StatelessWidget {
12   final NewsRepository newsRepository = NewsRepository(NewsWebServices());
13
14   NewsScreen({super.key});
15
16   @override
17   Widget build(BuildContext context) {
18     return BlocProvider(
19       create: (BuildContext context) => NewsCubit(newsRepository)..getAllNews(),
20       child: Scaffold(
21         appBar: AppBar(
22           title: const Text('News'),
23           centerTitle: true,
24         ), // AppBar
25         body: buildBlocWidget(),
26       ); // Scaffold
27     );
28
29     Widget buildBlocWidget() {
30       return BlocBuilder<NewsCubit, NewsState>(builder: (context, state) {
31         if (state is NewsLoading) {
32           return showLoadingIndicator();
33         } else if (state is NewsLoaded) {
34           return buildLoadedListWidgets(state.news);
35         } else if (state is NewsError) {
36           return showErrorMessage();
37         }
38       });
39     }
40   }
41 }

```

unknown, 2 days ago | ln 9 Col 14 | Spaces: 2 | UTF-8 | CR LF | Dart | Go Live | Windows (Windows-x64) | Prettier | ENG | 7:17 AM | 6/11/2023



```

File Edit Selection View Go Run Terminal Help
... news_screen.dart - yourstock - Visual Studio Code
EXPLORER lib > layout > screens > news_screen.dart NewsScreen
news_screen.dart
news_screen.dart
news_stock_screen.dart
search_result_screen.dart
search_screen.dart
settings_screen.dart
stocks_screen.dart
stocks_screen.dart
watchlist_screen.dart
home_layout.dart
models
services
shared
utilities
views
firebase_options.dart
main.dart
notes
linux
macos
test
web
windows
OUTLINE
TIMELINE
NPM SCRIPTS
DEPENDENCIES
news_screen.dart
lib > layout > screens > news_screen.dart NewsScreen
Widget buildBlocWidget() {
  if (state is Newsloading) {
    return showLoadingIndicator();
  } else if (state is Newsloaded) {
    return buildLoadedListWidgets(state.news);
  } else if (state is Newserror) {
    return showErrorWidget();
  } else {
    return Container();
  }
}; // BlocBuilder
Widget showLoadingIndicator() {
  return const Center(
    child: CircularProgressIndicator(
      color: Colors.deepOrange,
    ), // CircularProgressIndicator
  ); // Center
}
Widget buildLoadedListWidgets(List<News> allNews) {
  return SingleChildScrollView(
    child: Container(
      color: Colors.white,
      child: Column(
        children: [
          buildNewsList(allNews),
        ],
      ), // Column
    ), // Container
  ); // SingleChildScrollView
}
Widget buildNewsList(List<News> allNews) {
}

```

unknown, 2 days ago Ln 9, Col 14 Spaces: 2 UTF-8 ⚡ Dart ⚡ Go Live ⚡ Windows (Windows x64) ⚡ Prettier ⚡ 7:17 AM 6/11/2023

4.5 Pseudocode, Flowchart or workflow

1. User makes a new account.
2. User login to the application.
3. Server checks if the username and password matches or not.
4. User see suggested stocks in the home page.
5. User can search for stocks by using the name or the ticker.
6. User can see details about any stock.
7. User can see the news related to the stocks.
8. User can add stocks to his watchlist.
9. User can see the predicted price by the Machine Learning model of any stock.
10. User can delete his/her account and all the data on it.

Chapter 5: Testing:

In this chapter we're going to discuss and go deeper in our application testing and present the types of testing to be used and test cases we examined our application through.

5.1 Unit Testing:

Including unit testing in software documentation is important for ensuring the quality of the software, and can help developers and testers understand how to design and execute unit tests effectively.

Purpose : The purpose of unit testing is to validate the functionality of individual code units, identify bugs early, ensure code integrity, support refactoring, and facilitate faster and more efficient development processes. Unit testing is an essential part of the software development process that helps you ensure the high quality of your product: it allows developers to check the performance of each unit and prevent possible problems in advance. Throughout the product development lifecycle, unit tests save your time and money and help make code quality better.

Unit test cases:

Login	<ul style="list-style-type: none"> • Check response for valid e-mail & password. • Check response for invalid e-mail & password. • Check response for empty e-mail & password and click login.
Register	<ul style="list-style-type: none"> • Check response for e-mail that exists. • Check response for e-mail that doesn't exist. • Check response for password that doesn't match the confirm password box. • Check response for empty e-mail & password and click sign up.
Watch list	<ul style="list-style-type: none"> • Check if clicking on like button will put the stock in the watchlist. • Check if clicking on delete button will remove the stock from the watchlist.
Search	<ul style="list-style-type: none"> • Check if search find results by name or ticker.
View stock	<ul style="list-style-type: none"> • Check if the news related to the ticker. • Check if price , charts details related to the ticker.

Each test case specifies the inputs to the function, the expected output, and uses the expect function to compare the actual output of the function to the expected output. If the actual output does not match the expected output, the test case fails.

Summary: Benefits of unit testing is:

- Improve Quality and Performance. Unit testing can improve the quality of your codebase, making it more maintainable and less error-prone.
- Find Bugs.
- Makes Debugging Easier.
- Reduces Software Complexity.

5.2 Integrated testing:

Integrated testing is a type of software testing in which the different units, modules or components of a software application are tested as a combined entity.

Purpose : The purpose of integrated testing is to ensure that the individual modules or components of the system work together correctly, and that the system as a whole meets its functional and non-functional requirements.

Integrated testing can be performed at different levels of the software system:

System testing: This type of integrated testing verifies the interactions between the different subsystems or modules of the system. It is typically performed by software testers and can be manual or automated.

Acceptance testing: This type of integrated testing verifies that the software system meets the requirements and expectations of the end-users or stakeholders. It is typically performed by end-users or stakeholders and can be manual or automated.

There is different techniques can perform the Integrated testing:

Top-down testing: This technique starts with testing the higher-level modules or subsystems of the system and gradually moves down to the lower-level

modules or components. This approach is useful for identifying high-level issues early in the testing process.

Bottom-up testing: This technique starts with testing the lower-level modules or components of the system and gradually moves up to the higher-level modules or subsystems. This approach is useful for identifying low-level issues early in the testing process.

Hybrid testing: This technique combines both top-down and bottom-up testing approaches to verify the interactions between different modules or components of the system.

We used the Top-down testing to identify high-level issues early.

5.3 Additional Testing:

Additional testing is a type of software testing that goes beyond traditional functional and non-functional testing to explore other aspects of the software system.

Security testing: That testing focuses on identifying vulnerabilities and weaknesses in the software system that could be exploited by attackers. The Application will ask the user to verify his account to make sure that the account will be secured.

Usability testing: Usability testing is a type of testing that evaluates the user interface and user experience of the software system. The app is easy to use and nothing is not understandable.

Compatibility testing: Compatibility testing is a type of testing that verifies that the software system works correctly on different hardware, software, and network configurations. The App is compatible it works on Android and iOS.

Additional testing is important for ensuring the quality and reliability of the software system, and can help to identify issues that traditional functional and non-functional testing might miss.

Chapter 6: Results and Discussion

In this chapter we're going to find out the results of the project whether they're achieved or not and also the differences between the expected results and the actual ones.

6.1 Results:

6.1.1 Expected result:

1. User's record and data are secured.
2. Server checks if the username and password matches or not.
3. Server checks the verified e-mail if it's matches or not
4. User must verify the account by checking his gmail account.
5. User can have a demo account that can trade with virtual money.
6. User can access to the stocks and see it's details easily.
7. User can search for the stock by using the name or the ticker.
8. User can see the news related to the stocks
9. User can add stocks to his/her watchlist.
10. User can filter the stocks by (open price, close price, volume,...etc)
11. User can see the predicted price by the Machine Learning model of any stock.
12. User can delete his/her account and all the data on it

6.1.2 Actual results:

1. User's record and data are secured.
2. Server checks if the username and password matches or not.
3. Server checks the verified e-mail if it's matches or not.
4. User must verify the account by checking his gmail account.
5. User doesn't have a demo account that can trade with virtual money.
6. User can access to the stocks and see it's details easily.
7. User can search for the stock by using the name or the ticker.
8. User can see the news related to the stocks.
9. User can add stocks to his/her watchlist.
10. User can not filter the stocks by (open price, close price, volume,...etc)
11. User can see the predicted price by the Machine Learning model of any stock.
12. User can delete his/her account and all the data on it.

6.2 Discussion:

As a conclusion for the previous points, we have managed to meet most of the expectations we planned for except for some points such as let the user have a demo account to trade with virtual money and make the user filter the stocks.

Due to shortage of time, we canceled the feature of making demo account to the user (that's not the purpose of the application) and the ability of user can filter stocks need much requests to the Api that costs a big budget we can't afford.

Chapter 7: Conclusion:

In this chapter we sum up the whole thing and summarize
the things mentioned before.

Conclusion:

After all this work the project is finally done , our goal was providing retail investors and beginners as a third party investment tool that uses machine learning to help them take a decision in buying/selling their stocks, and we did our best to choose the best model that supports a high accurate result with low time consuming, also providing the user with charts and news that will illustrate the current stock market situation.

The app is under development and by good resources we could turn this project to one of the best stock applications by adding more features and servers , Improving Project planning and quality , Communicating With the customer listening to his feedback , Using intuitive time and expense technology and Discussing lessons learned in the middle of a project.

Chapter 8: Future work:

In this chapter we will propose what are the future plans and some developments we have concerning our application.

➤ Here is our plan for the coming versions of YourStock :

- User can create a demo account to trade in stocks using virtual money that will help the user learning and taking experience in the stock market.
- Application will make filter options to the user (open price ,close price , volume,...etc) that will facilitate searching the stocks to the user.
- The Application will add Multiple features in the stock chart such as (Indicators , Annotations , Comparisons , Customization)
- The Application using the machine learning model will give the user recommended stocks that has a high chance of increase in it's price.
- The application will send a notification to the user if one of his/her stock changes it's price.

• Bibliography

- Faculty of Computers and Artificial Intelligence Helwan university
- <https://www.tradingview.com/>
- <https://www.geeksforgeeks.org/>
- <https://www.alphavantage.co/>
- https://lucid.app/documents#/documents?folder_id=recent
- <https://www.tensorflow.org/>
- <https://www.investopedia.com/>
- <https://www.google.com/finance/?hl=en>
- <https://www.investing.com/economic-calendar/>
- <https://www.alphavantage.co/>