Nostradamus (Predicting Stock Price)

*Group Name: YA

1st CheongRok Yoon
Information Systems
Hanyang University
Seoul, South Korea
jashealer@naver.com

2nd ZiJian AN
Information Systems
Hanyang University/ESILV
Seoul, South Korea
azj535041051@hanyang.ac.kr

***Abstract*—Stocks are a "game" of probability. No matter what decision to buy or sell, there is no 100 percentage correct or incorrect one. The potential of the stock market is amazing, and the risks are also great. When does the customer buy and when to sell. Its profit and loss depend on the customer's decision. We will develop an application software that can predict the stock market to help customers make decisions. This article focuses on how to design and implement applications through software engineering methods. The app is equipped with some features and deep learning technology.**

***Index Terms*—stock prediction, tensorflow, Android OS, deep learning, data analysis**

I. Introduction

The stock market has always been a persistent topic. Currency and stocks affect people differently. Some people become rich overnight due to inventory, while others become poor overnight. "Veterans" who have experience in stock trading can use their experience to buy and sell at an appropriate price. This is an ability that requires years of experience. According to the survey, 80percentage of new immigrants will be eliminated within three months to one year. In this process, most people are the result of blind follow and gambling psychology. Generally, political factors, economic factors, corporate factors, and time series factors affect stock prices. Political factors include international and domestic political situations, relevant laws and policies issued by the government and other factors. There are many economic factors that affect stock prices, which are mainly affected by the growth rate of gross domestic product (GDP). If the GDP growth rate is large, it will usually stimulate stock investment and increase the stock price. Conversely, if GDP growth is slow or negative, the stock price will fall. The company factor mainly depends on the company's profitability. Generally, the stock price and the company's profit change simultaneously. Finally, time series factors affect stock prices. Time series factors include past stock price and trading volume. These factors make it possible to analyze market psychology and trend. Analyzing political, economic, corporate factors is called "fundamental analysis". Fundamental analysts study anything that can affect the security's value, from macroeconomic factors such as the state of the economy and industry conditions to microeconomic factors like the effectiveness of the company's management. Analyzing Time series factors is called "technical analysis". Technical analysis examines and predicts price movements in the financial markets, by using historical price charts and market statistics. It is based on the idea that if a trader can identify previous market patterns, they can form a fairly accurate prediction of future price trajectories. But for new people entering the stock market for the first time, they need to learn a lot of relevant knowledge and cultivate insights. At this time, there is a software Nostradamus that can help new people make decisions. What is Nostradamus? Nostradamus is an application that predicts the stock market and helps customers make investment decisions. This app will use deep learning technology in order to do technical analysis. It aims to make it possible for people to save time to study technical analysis. Customers can find the stocks they want to invest through Nostradamus and make good judgments based on the data analysis obtained.

TABLE I
Role Assignments

Role	***Name***	***Task description***
User	CheongRok Yoon	the people who want to invest stock and want statistical help.
Customer	ZiJian AN	Special personnel is required to supervise and communicate, communicate problems promptly.Clarify requirements, development environment, and development time. Determine the development cycle and time,delivered within a certain time.
Software Developer	CheongRok Yoon	Understand all the characteristics of the stock market and basic economic feature. Manipulate data for use in deep learning Provide customers with as useful information as possible for decision making.
Development Manager	ZiJian AN	Communicate customer ideas and requirements with developers. Ensure that the development team can work as efficiently as possible and complete the delivery within a certain time.Regularly test, maintain, optimize, and improve product performance with developers.

II. Requirements

A. Mobile application

1) Run: This application will run on Android OS.

B. Login screen

1) Login
   a) The login screen is the first page that users can see when they enter our application. If the username and password match our database, the user can log in to our application to use our application.
2) Sign-up
   a) New users who use this software for the first time need to register their own account. Users can choose to use their favorite numbers or meaningful numbers as their ID. User can choose to use their favorite numbers or meaningful numbers as their ID.
3) ID/PW Search
   a) When users forget their ID or password, this function can find their ID or password after the verification process.

C. Browse

1) Search
   a) When the user logs in his ID, he jumps to the search function page. Users can use this function to find the stock information they care about.

D. Main Page

1) Chart
   a) The stock chart consists of opening price, closing price, the highest and lowest prices. Users can check the recent stock prices at a glance through chart.
2) Performance
   a) The performance button simulate the app using historical data. This shows how successful the program has been in price forecasting. User can see how reliable the app is through this.
3) Result
   a) This screen shows the prediction result of Nostradamus. It consists of prediction price, probability and volatility. User can use this result to make investment decision.

E. Bookmarks

1) Bookmarks
   a) What bookmarks do: Bookmarks are the records saved when browsing stocks so that they can be viewed at any time in the future. Each user can add the data he cares about to the bookmark.

III. Development environment

A. Choice of software development platform

1) Windows10
   - We will going to use Windows10 OS platform. We will be using Windows 10. The first big reason is that the api provided by HTS (Home Trading System) of a Korean securities company is made window-friendly. Also, later we need data from Korean stocks to do deep learning. HTS provides high-quality data necessary for learning. So We chose windows. In addition, we are going to use Android.
2) Programming Language
   a) Python 3.6
      - Python helps developers be productive and confident about the software they're building. Benefits that make Python the best fit for machine learning and AI-based projects include simplicity and consistency, access to great data analysis libraries(NUMPY, SCIPY, PANDAS, SEABORN) and frameworks for AI and machine learning (TensorFlow, Keras, Scikit-learn). In addition, Python has wide community to study. There are Python forums and an active exchange of experience related to machine learning solutions. For any task, the chance is pretty high that someone else out there has dealt with the same problem. This is why our team chooses Python3.6.
   b) R4.0
      - R is generally applied when need to analyze and manipulate data for statistical purposes. R has packages such as Gmodels, Class, Tm, and RODBC that are commonly used for building machine learning projects. These packages allow developers to implement machine learning algorithms. This language can give you in-depth statistical analysis whether you're handling data from an IoT device or analyzing financial models. What's more, if your task requires high-quality graphs and charts, you may want to use R. With ggplot2, ggvis, googleVis, Shiny, rCharts, and other packages, R's capabilities are greatly extended, helping you turn visuals into interactive web app.
   c) Android Studio
      - Android Studio is built-in based on Intellij IDEA, which can provide the shortest possible turnaround time for coding and running workflows. With the Apply Changes feature of Android Studio, you can push code and resource changes to the running application without restarting the application (in some cases, you don't need to restart the current

Activity). When you want to deploy and test small incremental changes while maintaining the current state of the device, this flexibility helps you control the scope of application restarts. Android Studio includes project and code templates, allowing you to easily add existing features such as navigation drawer and ViewPager. You can start with code templates, or even right-click an API in the editor and select Find Sample Code to search for examples. In addition, you can also import a full-featured application from GitHub directly on the "Create Project" screen. Android Studio fully supports the modification of C/C++ project files, enabling you to quickly build JNI components in your application. IDE provides syntax highlighting and reconstruction functions for C/C++, and also provides a debugger based on LLDB, which allows you to debug Java and C/C++ code at the same time. The build tool can also execute CMake and ndk-build scripts without any modification, and then add shared objects to the APK.

3) Cost Estimation

TABLE II
COST ESTIMATION

Category	*Name and Version*	*Cost*
Operating System	Windows10 1909	£119.99
IDE	Android Studio version-4.0.2	0
Communication	KakaoTalk	0
Documentation	Overleaf	0
Programming Language Back end	Python R	0
Programming Language Front end	Java	0

4) Information about development environment
   a) Computer
      i) Window PC
         - processor: AMD Ryzen7 1700 Eight-Core Processor
         - Ram: 24GB
         - OS: Windows 10 Pro
      ii) HP OMEN 15-ax016TX
         - Processor: Intel Core i5, 2.3GHz Quad-core
         - Ram: 4GB 1600MHz DDR4
         - OS: Windows 10 home edition

B. Software in use

1) Android Studio
   - Android Studio is an official integrated development environment (IDE) based on IntelliJ IDEA and suitable for developing Android applications. In addition to IntelliJ's powerful code editor and developer tools, Android Studio also provides more features that can improve the efficiency of Android application compilation.

2) Python
   - Python is a cross-platform computer programming language. It is a high-level scripting language that combines interpretation, compilation, interactivity and object-oriented. It was originally designed to write automated scripts (shell). With the continuous update of the version and the addition of new language features, the more it is used for the development of independent and large-scale projects.

3) Java
   - Java is an object-oriented programming language. The Java language has two characteristics: powerful and easy to use. As a representative of the static object-oriented programming language, the Java language implements the object-oriented theory very well, so that programmers can execute complex programming with elegant thinking. Java has the following characteristics: simplicity, object-oriented, distributed, robustness, security, platform independence and portability, multi-threading and dynamics. Java can write desktop applications, web applications, distributed systems and embedded system applications.

4) Jupyter
   - The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

5) R
   - R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. Polls, data mining surveys, and studies of scholarly literature databases show substantial increases in popularity; as of September 2020, R ranks 9th in the TIOBE index, a measure of popularity of programming languages

C. Task distribution

1) Task distribution

TABLE III
TASK DISTRIBUTION

Name	*Task Distribution*
CheongRok Yoon	Back-end (data analysis)
ZiJian AN	Front-end (display screen)

IV. SPECIFICATIONS

A. Login Screen

1) Login

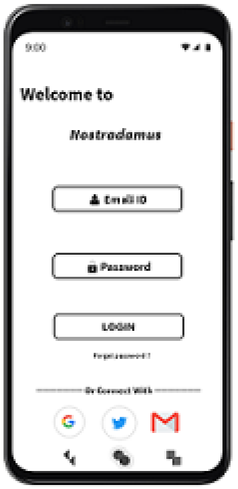


Fig. 1. Login

This is the first screen where the user logs into the application login page. Users who enter our APP for the first time must register before they can enter. If the user logs in successfully, it will be verified by the internal identification system and then jump to the search page. When the user login ID or password is entered incorrectly, "wrong id or wrong password" is displayed. At this time, the user needs to input again. If the user enters the wrong password more than 5 times, it will be locked. Users can reset their passwords through email verification.

2) Sign-up
When a user registers for an account, the user must use a unique name as his user name. If the name used by the user matches the name in the database, "duplicate" will be displayed and a new name will be required. The second step requires the user's email account as the ID when logging in. The third step when setting a password, the user must set a password within 15 digits. These include uppercase letters, lowercase letters, numbers and special symbols. Users need to click "I agree to this clause" to register their account. If users have Google, email and Twitter accounts, they can connect to our APP to create an account.

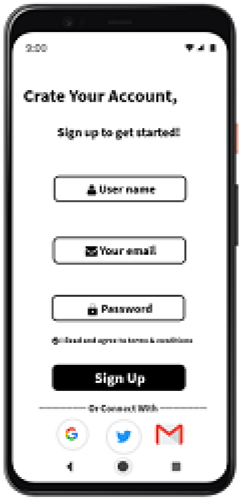


Fig. 2. Sing-up

3) ID/PW Search

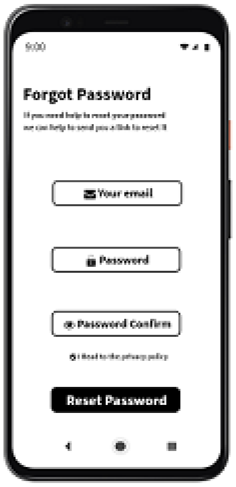


Fig. 3. ID/PW

If the user forgets his password, he can verify and reset his new password through the email during registration. The user cannot use the old password in the past to reset the new password. If there is no matching object in the database to reset the new password, the password is considered successful.

B. Browse

1) Search

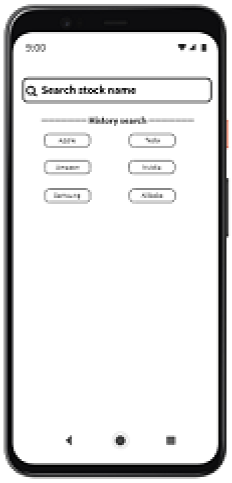


Fig. 4. Search

The search dialog is an interface component controlled by the Android system. After being activated by the user, the search dialog will be displayed at the top of the activity. The Android system controls all events in the search dialog. When a user submits a query, the system will pass the query to the Activity you specify to process the search. The dialog box can also provide search suggestions while the user enters content. Searchable Activity is an activity in your application that performs a search based on a query string and displays search results. When the user performs a search in the search dialog or search widget, the system will start the searchable activity and pass the search query in the Intent to it through the ACTIONSEARCH operation. The searchable activity retrieves the query from the QUERY extra of the intent, then searches the data and displays the results.

C. Main Page

1) chart
In the stock market, charts allow people to see the stock market situation at a glance. So we will always have a stock chart on the main page. We provide customer with chart like above figure1.1. On the bar chart there are four values: open price, high price, low price, and closing price to indicate additional trends for that trading day. To draw a candle stick bar, open, high, low, and close price are required, but the DataFrame provided by HTS already has a column composed of the corresponding data. And We will use the data we receive to represent the moving average line. The moving average(MVA) line makes the chart easier to see by removing noise. Also, the moving average line can inform people of mid- to long-term trends. The formula for the moving average(MVA) line is as follows. We use “KIUM Open API +” to acquire price data and draw real time chart through matplotlib library.

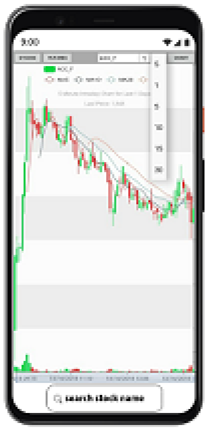
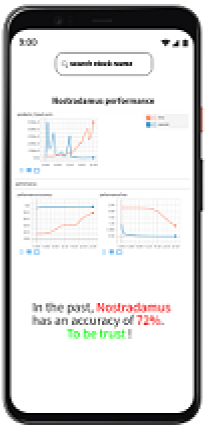


Fig. 5. Chart

2) Performance measurement

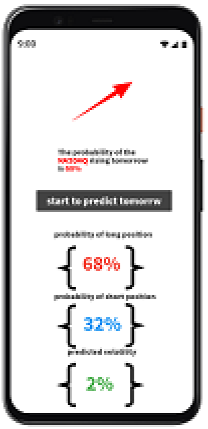


In the past, Nostradamus has an accuracy of 72%. To be trust !

Fig. 6. tensorboard

The performance measurement button simulates the app to measure accuracy. This page shows the accuracy of how well Nostradamus predicted stock price up and down(long position, short position). We create testsets by using past data obtained from HTS and simulate Nostradamus through testsets. To show performance in graph, we use TensorBoard. TensorBoard is TensorFlow’s visualization toolkit, enabling you to track metrics like loss and accuracy, visualize the model graph, view histograms of weights, biases. We will continue to update the performance table, and we expect that this page will give our customers more confidence.

3) Result page


9:00

The probability of the NASDAQ rising tomorrow is 68%

start to predict tomorrw

probability of long position

68%

probability of short position

32%

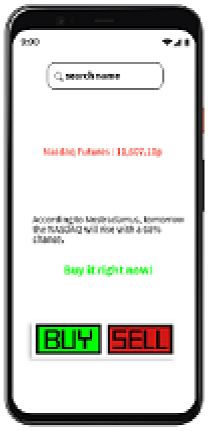
predicted volatility

2%

Fig. 7. Result page

In stock market, there are three positions: a long position, where you can benefit when stock prices rise, and a short position, where you can benefit when stock prices fall. In addition, there is a composite position that uses an option to make money based on volatility regardless of the fluctuations in the stock price. Therefore, Nostradamus outputs three predicted long and short positions and values of volatility to help customers make investment decisions. To get these three outputs, we will build an LSTM (Long Short Term Memory) model using TensorFlow. LSTM is a type of recurrent neural network (RNN). The RNN is suitable for learning time series data because it considers not only the current input value but also the input value that has been previously input.

4) Buy and Sell button
According to Nostradamus' signals, customers can buy and sell through mock investment account. It is test whether the signal is correct. We linked KOSPI 200 futures data through "KIUM Open API +" and implemented long position through buy button and short position through sell button. Stop loss refers to setting the maximum amount of loss that can be incurred in order to prevent serious damage if the purchased item is falling below the purchase amount due to a continuous decline, and if it is below that amount, selling immediately. This is a function introduced by Nostradamus to minimize loss in case of prediction failure. We look forward to providing customers with opportunities to experience Nostradamus through these features.


9:00

search name

Nasdaq Futures | [illegible]

According to Nostradamus, tomorrow the NASDAQ will rise with a 68% chance.

Buy it right now!

Fig. 8. Buy and Sell button

V. ARCHITECTURE DIESIGN AND IMPLEMWNTATION

A. Overall architecture

We have two modules that make up our application. The first part is "Android studio", which is our front end. It mainly responds to the client part (user) and stores the user's information in the database. This will help our entire application to contain several functions and designs. The second part is that our back-end mainly responds to the database, using the Firebsae database. The back-end uploads the analyzed data to the database. The part of the back-end requests "stock information" from the source server. Then analyze the requested data.

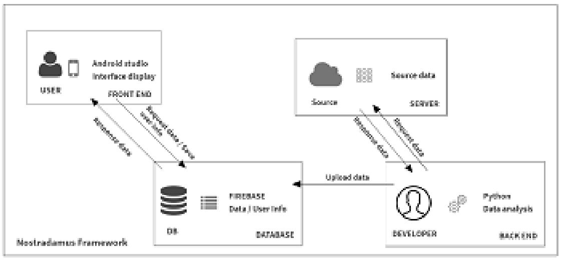


Fig. 9.

B. Directory organization

1) Task distribution

TABLE IV
DIRECTORY ORGANIZATION

Directory	***File name***	***Module name***
AndroidStudioProjects/nostradamus/app/src/main/java/com/example/nostradamus	MainActivity.java Password.java ProfileActivity.java RegistrationActivity.java SecondActivity.java ThridActivity.java UserProfile.java	Android Studio
AndroidStudioProjects/nostradamus/app/src/main/res/layout	Activity_main.xml Activity_password.xml Activity_profile.xml Activity_registration.xml Activity_second.xml Activity_thrid.xml	Android Studio
AndroidStudioProjects/nostradamus/app/src/main/res/menu	Activity_menu.xml	Android Studio
AndroidStudioProjects/nostradamus/app/src/main/res/drawable	ic_launcher_background.xml ic_launcher_foreground.xml ic_search_black_24dp.xml	Android Studio
AndroidStudioProjects/nostradamus/app/src/main/res/values	colors.xml string.xml Styles.xml	Android Studio
Backend	.Kiwoom .deeplearing .database	backend
Backend/main_kiwoom/ui	__init__.py	backend
Backend/main_kiwoom/kiwoom	kiwoom.py	backend
Backend/main_kiwoom/dataset_down	Dataset_down.py Dataset.csv	backend
Backend/main_kiwoom/config	errorCode.txt KiwoomType.txt	backend
Backend/main_deeplearing/Stock_Predict	Stock_LSTM.py	backend
Backend/main_deeplearing/DataProcessing	DataProcess.py Dataset.csv	backend
Backend/main_deeplearing/Result	Result.json	backend
Backend/database	Database.py	backend

C. Module 1: Front end

1) Purpose: We use the integrated development environment Android studio to develop our product-Nostradamus. Android studio is the official integrated development environment for Android application development. It provides a basic layout and uses the Java programming language to help us develop applications easily. The purpose of the software front end is to provide users with our visualization capabilities. Users can request their needs through the front-end, and can pass their requests to the back-end without knowing the internal service structure.

2) Functionality: Android studio is an environment based on all aspects of application development. And allows you to design applications using the features provided. It also provides various SDK tools to develop additional functions in the application. You can add some dependencies to connect to other programs or servers (Firebase). We build the front end through Android studio and display it on the user's screen. Users can operate through visual images.

3) location of source code: AndroidStudioProjects/nostradamus

4) class component There are several components in the Android studio.

a) MainActivity.java: This is the first picture the user sees. The user can enter the account and password to log in. If you enter it 5 times, you will lose the login permission (implemented by setEnabled(false)). When the user is a new user, click the register button to jump to the registration page to register (implemented by Intent()). If you forget the password, click on the forgotten password to jump (implemented by Intent()).

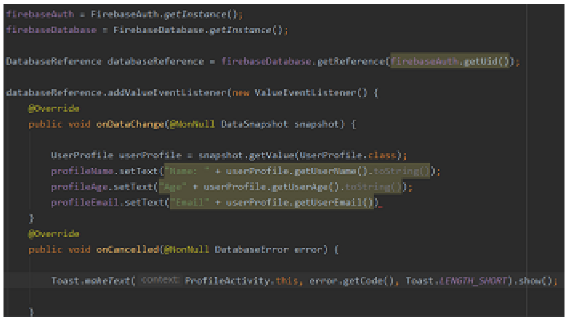
b) RegistrationActivity.java: This is the registration interface. Users need to enter relevant information to register. For example, email, name, password and age (implemented by name = userName.getText().toString()). If the user fills in the blank information, it shows that more information is needed.(implemented by (name.isEmpty()).

c) Password.java: This is the password recovery interface. When the user forgets the password and needs to retrieve the password. The user only needs to enter the email address when registering the account. Then log in to your mailbox, there will be a password change email, where you can reset your password. implemented by firebaseAuth.sendPasswordResetEmail(userEmail). addOnCompleteListener(new OnCompleteListener<Void>().

d) SecondActivity.java: This is the Second interface for successful login. At the top of this page, users can click on Profile to view their personal information (implemented by Intent()). There is a search function at the bottom of the page. Here users can find what they want. Then click the NEXT button to jump to the next page (next.setOnClickListener). If the user wants to log out, click the LOGOUT button (logout.setOnClickListener).

e) Profile.java: This page is the profile interface. Use this page to see your personal information.

The implementation is shown in the figure below.

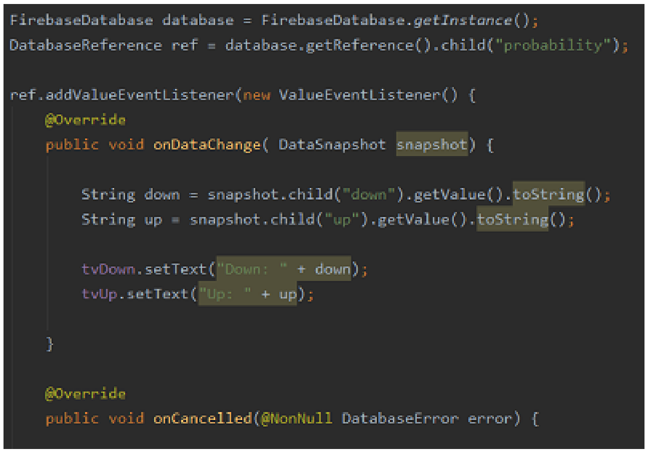


```
firebaseAuth = FirebaseAuth.getInstance();
firebaseDatabase = FirebaseDatabase.getInstance();

DatabaseReference databaseReference = firebaseDatabase.getReference(firebaseAuth.getUid());

databaseReference.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {

        UserProfile userProfile = snapshot.getValue(UserProfile.class);
        profileName.setText("Name: " + userProfile.getUserName().toString());
        profileAge.setText("Age" + userProfile.getUserAge().toString());
        profileEmail.setText("Email" + userProfile.getUserEmail())
    }
    @Override
    public void onCancelled(@NonNull DatabaseError error) {

        Toast.makeText( context: ProfileActivity.this, error.getCode(), Toast.LENGTH_SHORT).show();

    }
```

Fig. 10.

f) ThridActivity.java: Jump to this page by clicking the search function in the second interface. This page displays the data found through the search function.

The implementation is shown in the figure below.



```
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference ref = database.getReference().child("probability");

ref.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange( DataSnapshot snapshot) {

        String down = snapshot.child("down").getValue().toString();
        String up = snapshot.child("up").getValue().toString();

        tvDown.setText("Down: " + down);
        tvUp.setText("Up: " + up);

    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
```

Fig. 11.

D. Module 2: Back end

1) Purpose: The purpose of backend is delivering stock forecast information to the frontend. To send information to the front-end, there are four procedures in detail. The first is to receive stock information through the securities firm API. Receive account information and stock information using KIWOOM OPEN API. Second, the data received should be handled appropriately for deep learning. Training data should be processed in chronological order to use LSTM. Third, we have to make a model. LSTM was chosen as the model for learning. LSTM is doing very well in processing continuous data. The LSTM model was made using Keras of Tensorflow. Finally, predictive information and personal account information through the model are forwarded to the frontend through the database. We used the Firebase database to deliver the information.

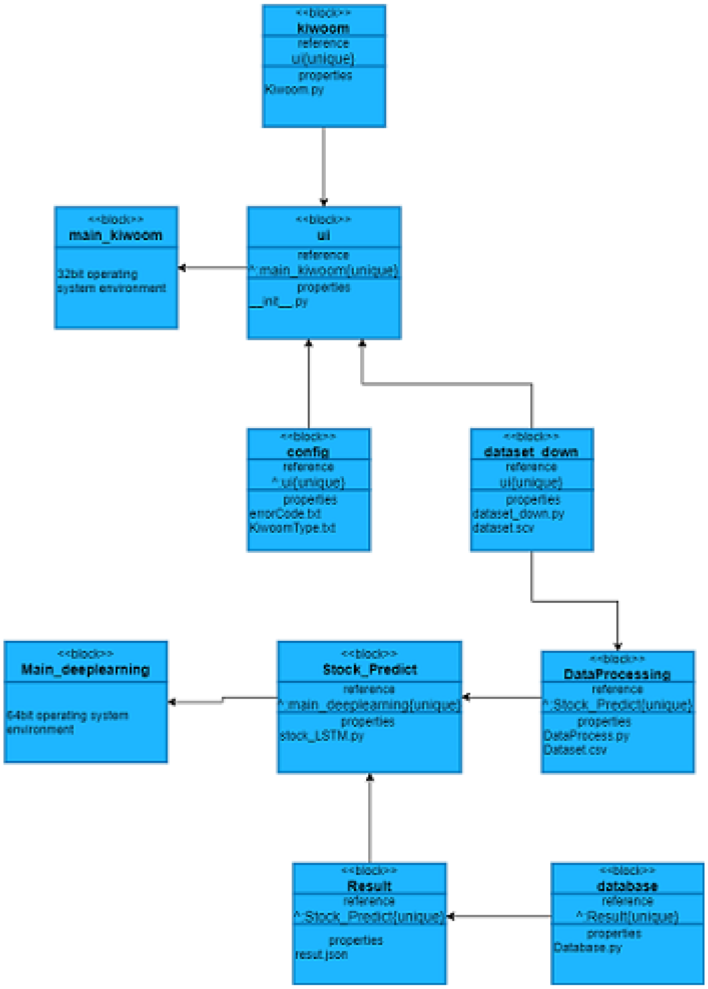


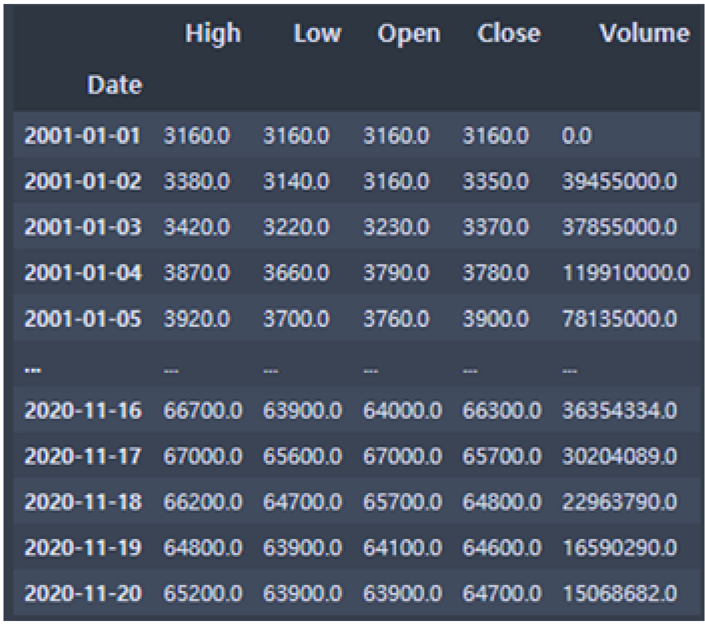
Fig. 12. Backend directory organization

2) Functionality: TensorFlow is an open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level Keras API, which makes getting started with TensorFlow and machine learning easy.

3) Location of Source Code: /pythonproject /Nostradamusbackend

4) Class component: There are several components in the project.

a) Datasetdown.py:



	High	Low	Open	Close	Volume
Date					
2001-01-01	3160.0	3160.0	3160.0	3160.0	0.0
2001-01-02	3380.0	3140.0	3160.0	3350.0	39455000.0
2001-01-03	3420.0	3220.0	3230.0	3370.0	37855000.0
2001-01-04	3870.0	3660.0	3790.0	3780.0	119910000.0
2001-01-05	3920.0	3700.0	3760.0	3900.0	78135000.0
...	...	...	...	...	...
2020-11-16	66700.0	63900.0	64000.0	66300.0	36354334.0
2020-11-17	67000.0	65600.0	67000.0	65700.0	30204089.0
2020-11-18	66200.0	64700.0	65700.0	64800.0	22963790.0
2020-11-19	64800.0	63900.0	64100.0	64600.0	16590290.0
2020-11-20	65200.0	63900.0	63900.0	64700.0	15068682.0

Fig. 13.

Datasetdwon class plays the role of getting data from securities firms. We went through the stock selection process for deep learning as follows. First, among the 200 stocks that represent Kospi200, the top 10 stocks in market capitalization were selected. And to predict whether each stock will rise or fall tomorrow, we collected daily data from the listing date to the present day. Daily data consist of open price, high price, low price, close price, and trading volume for each day. In addition, the adjusted stock price was selected to obtain accurate stock prices excluding factors that affect stock prices such as dividends, capital decrease, and capital increase. The stock data we selected were obtained through Kiwoom API +.

b) DataProcess.py: DataProcess Class does three things. The first is to extract learning data from M period in a time series. The second is to do one hot encoding. The third is to divide the dataset into training set(80) and test set(20).

First, we have extracted five column data of open price, close price, high price, low price, and trading volume of M days (blue bar) from the start date (t) and close data of M+1 day (yellow bar), which we intend to predict. It then extracted data from M days that lasted one day (t+1) from the start date and extracted close data from M+1 day. The above process was repeated over and over the entire dataset.

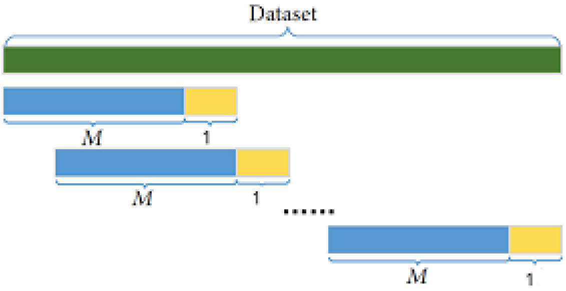


Fig. 14.

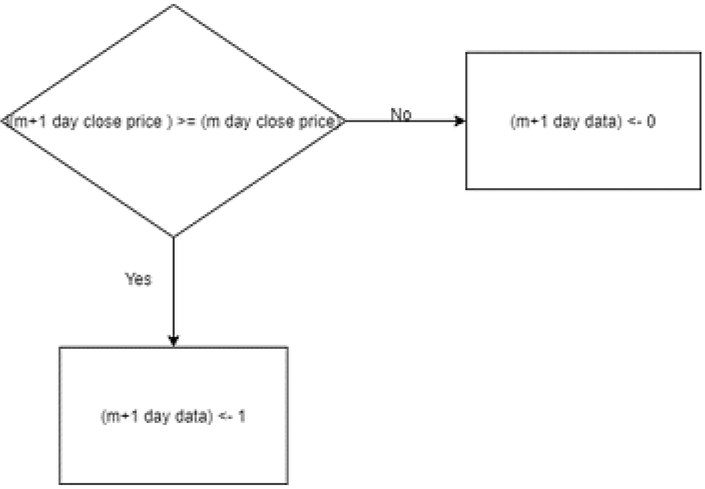


Fig. 15.

Second, we applied one-hot encoding to the M+1 (yellow bar) data that we wanted to predict to use the binarycrosentropy loss function. If the close price (t=M+1) rose the following day from the previous day's (t=M) close price, the M+1 data would be replaced with 1 instead of 0 otherwise. In other words, M+1 data was composed of two classes: 1 when rising and 0 when falling.

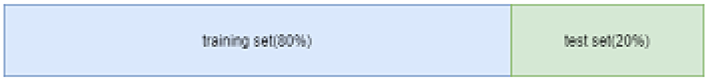


Fig. 16.

Third, We divided the dataset into training set(80) and test set(20).

c) Kiwoom.py: Kiwoom class serves to communicate with securities firms. Kiwoom open api adopted the oxy method. Therefore, we used the PyQt5 library to construct the overall class. Logging in to securities firms is essential and is designed to get stock event information and personal stock account information. When the class receives an event request from the front-end, it connects to the securities firm's server in real time, receives the information, and delivers the information back to the front-end.

d) Ui.py The ui class plays a role in making the Kiwoom class run throughout the stock market

e) StockLSTM.py:

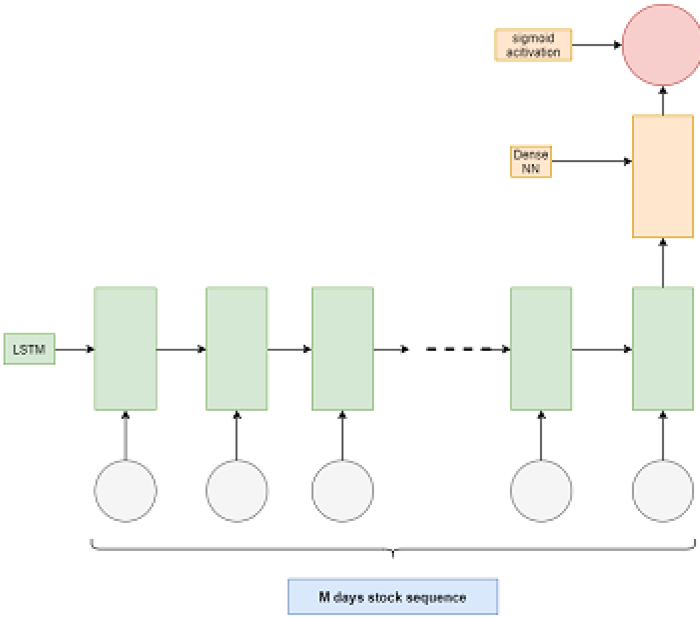


Fig. 17.

StockLSTM class uses the tensorflow library to learn, receiving processed data, learning, and storing values of parameters. They also work to calculate the predicted probability values. In learning data, we used LSTM optimized for continuous data. LSTM serves to hand over important information of past learning to the preceding nodes. The form of NN is Many - to - one. Because we receive as much learning data as M days and print only one output of probability whether it will rise or fall on M+1. In addition, the Binarycrossentropy will be used as a loss function, with only two cases of classes predicting whether the result will rise or fall. It Computes the cross-entropy loss between true labels and predicted labels when there are only two label classes (assumed to be 0 and 1). Finally, we selected the last activation function as sigmoid function to indicate output as probability.

f) Database.py:
Database class serves to deliver stock account information to the database. It also delivers probability values calculated from StockLSTM.py We used Firebase to upload those data to the server.

VI. Use Cases

A. Installation

This mobile application can be found in the "Google PlayStore". When users search for certain keywords (such as "stocks", "predictions"), our app will be recommended. Just click a button to install it on the user's mobile phone.

B. Starting the application

If the users clicks the 'Nostradamus' application button, Our application will display the login page

C. Login

The login page is the first page that users can see when they enter our application. Users must fill in their email and password on this page. If the username and password match our database, the user can log in to our application to use our function. But if the password is wrong, they will trigger a fault message. If the user is a new user. You need to register an account before you can log in and use the functions in our application. If the user already has an account. There is no need to log in, and you can directly enter the second page after logging in.

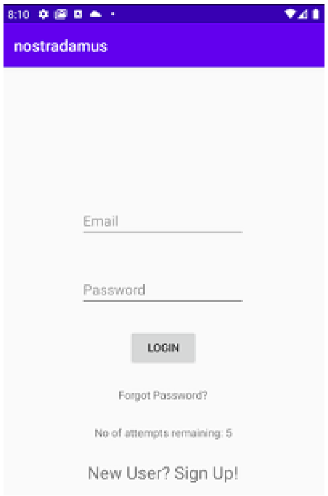


Fig. 18.

D. Register

Registration requires user information, such as email, name, password, and age. If the user enters empty. The user will be prompted "please fill in more details". The user has successfully registered, return to the login page. After users register, they can log in to our app. If they don't register in this case, they cannot use our app.

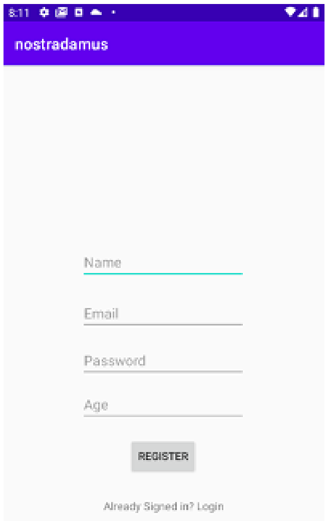


Fig. 19.

E. Forget Password

If the user forgets his password. You can click "Forget password" on the login page. Then, the user will enter the "Forgot Password" page. The user enters the email used for registration at the time. The mailbox will receive an email with a reset password. Click the password reset link, and users will be able to reset their own password.

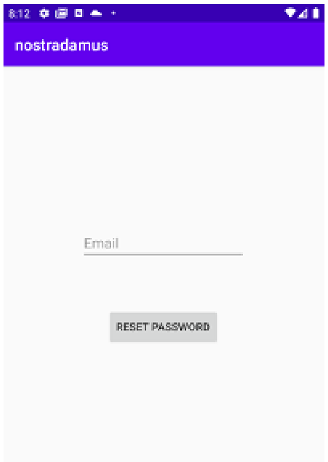
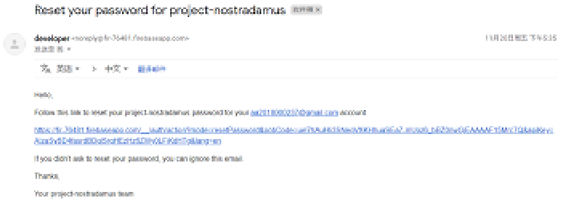


Fig. 20.



Reset your password for project-nostradamus

developer [illegible]

Hello,

Follow this link to reset your project-nostradamus password for your [illegible] account.

[illegible]

If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your project-nostradamus team

Fig. 21.

F. Second Page

When the user successfully logs in, enter the second page. This page has a profile function and a search function.

1) Profile

This page will display the user's personal information. When a user logs in, the user's information will be recorded in the database. When the user enters this page, the user data in the database is called.

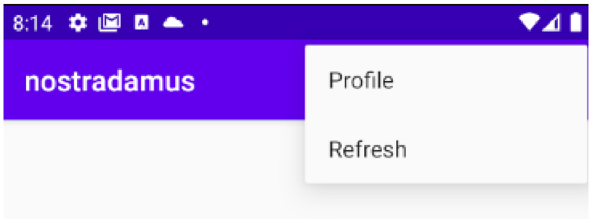
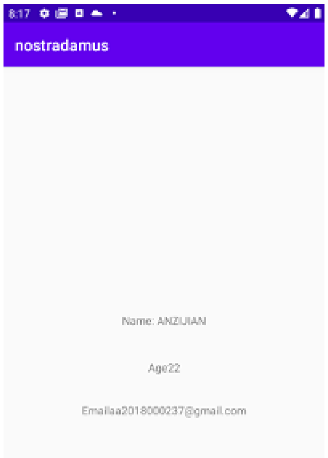


Fig. 22.



Name: ANZIJIAN

Age22

Emailaa2018000237@gmail.com

Fig. 23.

2) Search
   Users can find the stock information they want to search through the search function. Match the data in the database with the entered information, and the data will be displayed on the third page.

3) Logout
   When the user wants to exit the application, click the LOGOUT button.

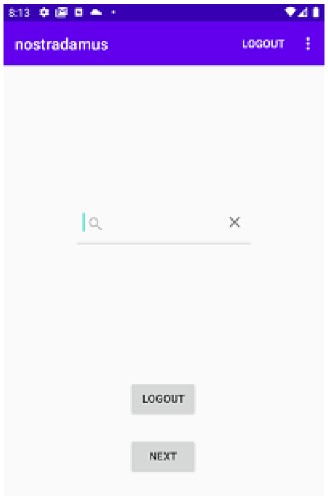
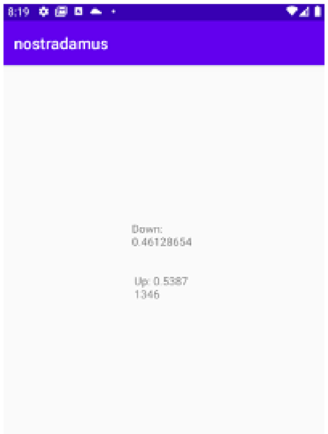


Fig. 24.

G. *Third Page*

The third page will show the results that users have found through the search function.



Down:
0.46128654

Up: 0.5387
1346

Fig. 25.

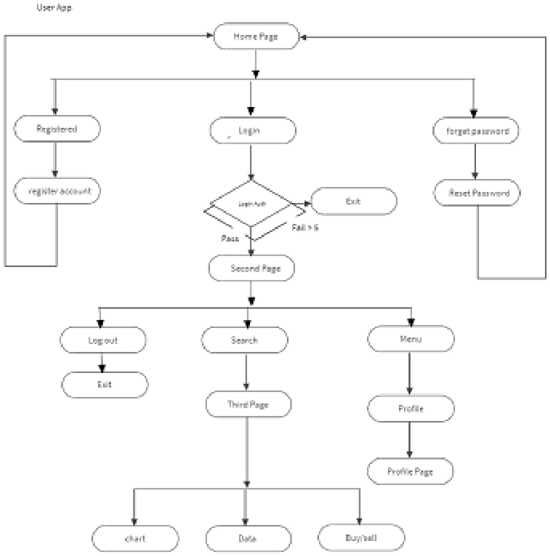


Fig. 26.

VII. DISCUSSION

A. *conclusion*

1) We have been working on this project for more than three months. We researched our ideas and looked for other examples of the design and implementation of our own applications that can be used to realize our ideas. Since this is the first time to personally implement the software, there are many shortcomings. Some very simple things become very complicated. In the search for stocks, I don't know how to refer to the search algorithm. I used a method that I think is quite stupid. I used the if else statement to complete. This made a redundant program. And it is very inefficient.

In the application, we have implemented login, forgot password, registration, personal information and search functions. But we also encountered parts that were not realized. For example, the chart function and the buy and sell functions are not implemented. I found some examples of chart and buy and sell on the Internet. Because I need to add some related dependencies, but my computer cannot recognize these dependencies. This led me to be unable to implement chart functions and buy and sell functions. This is the worst situation we have encountered.

2) In many LSTM examples, learning model is often done at stock prices. However, this leads to an autocorrelation error. Rather than the computer predicting the price of a stock, it simply makes an error that closely follows the previous day's price. So, my group trained in two labels, rising and falling. As a result, learning was difficult. Increasing the parameters did not change the result. From these results, we concluded that data for stock forecasting were lacking or that other key variables were missing. The performance is not perfect yet, but it was meant to be reached with these conclusions. This part will be modified later to improve performance.

We must study hard to understand unfamiliar technologies, but one of our biggest goals is to combine our own skills with technology to achieve our learning goals. We think we achieved our simple goal because we learned a lot how to use these programming techniques. In the future, because stocks are applicable in any country, only a few young people in our country pay attention to stocks. We hope that more people can easily access stocks and become interested in stocks through our stock forecasting application.