Software Engineering II
Web Application

Web Based Stock Forecasters

Final Report

Group #15

Boyu Ni

Jiajun Zhu

Ruoxi Geng

Sam Ramezanli

Xiaoheng Liu

Table of Contents

1. Introduction:

Today people are using many different methods to accumulate their wealth. Investing stock is one of the most popular ways to make more money. In order to maximize the benefits, the investors intend to predict the trend of Stocks so they can make better decision regarding purchasing Stock shares. Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on a financial exchange. The successful prediction of a stock's future price could yield significant profit for the stockholder. The aim of our project is to develop an application that runs continuously as a background process and periodically retrieves stock information, parses the received responses, and stores the extracted parameters into the local relational database and then analyze the data using the Machine Learning. Moreover, the web site will provide the prediction for the stock.

The aim of our project is to provide more information regarding possible future prices for stocks by predicting the future price value based on historic price. This will provide a guideline for investors to make decisions while investing in stock market. We intend to achieve this accomplishment by following some algorithm to give approximate stock price in form of long-term and short term predictions.

The user interface of the web site is friendly and simple. We have implemented complex and smart algorithms to perform price predictions. These algorithms run as a back-end task and compute the prediction values for the various stocks.

The main purpose of the project was to implement a system, which will make use of web-services and make the future stock prediction. In order to predict the stock’s future price, we implemented two algorithms one which is predicting future value estimation based on the Monte Carlo method, and the other which is using curve-fitting algorithm. In Addition, the database holds the updated Yahoo! Finance data, as well as the results generated by the predictors.

This report describes the overall structure, behaviors and interactions of various modules of our system. It also contains the details of the algorithm, prediction modules, databases and web services used in our project.

Project Goals:

- **Potential customers:**

Our system aims to provide the service for two clusters users. First is the user that just wants to get the real time data of stocks, we call this type of customer **Guest**. They can just use only the simple function of our system. The second is the user who has registered to our system and we call him the Client. In addition, they may be investors who trade the stock shares and they need more information about stocks, and stock predictions.

- **Data collection:**

Our system can extract the stock price from the internet and store in the local database. Our major data source is YAHOO FINANCE. We can extract the real time price and the historical price for a long period.

- **Stock price prediction:**

We can use the data we collected to do the stock price prediction, including short-term prediction and long-term prediction. To let the users can better arrange their investment. The system will also provide a basic recommendation to the users.

- **User-friendly interface:**

Our system provides a user-friendly interface to the user. We have very simple navigation bar on top of our interface, which can let the user to get all the functions in the system easily. Furthermore, we provide the charts for showing the change of the stock price. All these make our system friendly to the users.

- **Personalization setting:**

One of the highlights of our system is the personalization setting in the system. Our system has a function call “Favorite Stock”. The users can choose stocks that they have bought or stocks they want to buy. The system will extract the information of the favorite stocks, and show the prediction on the screen when the user clicks on the page of “Favorite Stock”.

2. Technology Background

2.1 Python

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming orprocedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Like other dynamic languages, Python is often used as a scripting language, but is also used in a wide range of non-scripting contexts. Using third-party tools, such as Py2exe or Pyinstaller, Python code can be packaged into standalone executable programs. Python interpreters are available for many operating systems.

2.1.1 Django --Python web development framework

Django is a free and open source web application framework, written in Python, which follows the model–view–controller architectural pattern. It is maintained by the Django Software Foundation, an independent organization established as a 501non-profit. Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

2.1.2 SQLite --- embedded database for Django web design

SQLite is a relational database management system contained in a C programminglibrary. In contrast to other database management systems, SQLite is not

a separate process that is accessed from the client application, but an integral part of it.It is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.

SQLite is a popular choice as embedded database for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems, among others. SQLite has many bindings to programming languages.

2.1.3 Pandas --- Legendary data analysis library in Python to make stock operations

What's pandas? Pandas is a open source BSD-licensed(Berkeley Software Distribution) library providing high-performance, easy-to-use data structure and data analysis tools for the Python programming language.

What can pandas do? Python has long been great for data munging and preparation, but less so for data analysis and modeling. *pandas* helps fill this gap, enabling you to carry out your entire data analysis workflow in Python without having to switch to a more domain specific language like R.

Combined with the excellent IPython toolkit and other libraries, the environment for doing data analysis in Python excels in performance, productivity, and the ability to collaborate.*pandas* does not implement significant modeling functionality outside of linear and panel regression; for this, look to statsmodels and scikit-learn. More work is still needed to make Python a first class statistical modeling environment, but we are well on our way toward that goal.

Some other libraries we use to assist pandas:

Sklearn: We can comprehend Sklearn as "Scientific Kit for Machine learning", there are many method for machine learning, like "Logistical Regression", "Linear Discriminant Analysis", "Quadratic Discriminant Analysis", etc,.

Numpy, matplotlib, datetime, math.

2.2 Stock prediction:

2.2.1 Long-term prediction: Stock market prediction has been an important issue in the field of finance and mathematics due to its potential financial gain. As a vast amount of capital is traded through the stock market, the stock market is seen as a peak investment outlet. In addition, stock market prediction brings with it the challenge of proving whether the financial market is predictable or not. With the advent of faster computers and vast information over the Internet, stock markets have become more accessible to either strategic investors or the general public. With advent of Machine Learning techniques, computers can "learn" various features and trends in data which can then be used to predict future values. The major focus of machine learning research is to extract information from data automatically, by computational and statistical methods. The Machine learning method we use here to do long-term prediction is "Quadratic Discriminant Analysis"

2.2.2 Short-term prediction: the same like the long-term prediction, the different thing is we only use very short-time data to do our prediction. We use Logistical Regression and Linear Discriminant Analysis to do our short-term prediction.

3. System Design

3.1 System requirements

The following Table 3.1 describes the functional requirements for the system.

Identifier	Priority	Description
REQ-1	5	The system shall be able to let the user search a stock.
REQ-2	5	The system shall be able to extract the data from web source like Yahoo Finance API.
REQ-3	5	The system shall be able to store the stock data in the local database.
REQ-4	5	The system shall be able to retrieve stock data from local database.
REQ-5	5	The system shall be able to do the prediction of the stock price, including short-term and long-term prediction.
REQ-6	4	The system shall be able to let user choose the favourite stock.
REQ-7	4	The system shall be able to let the user register a new account.
REQ-8	4	The system shall be able to let the user login/logout to the system.
REQ-9	3	The system shall be able to provide basic function to the guest.
REQ-10	3	The system shall be able to provide all functions to the client.
REQ-11	2	The system shall be able to provide a user-friendly interface to all users.
REQ-12	2	The system should be easy to use for all users.
REQ-13	4	The administrator should be able to access the user account data.

REQ-14	4	The administrator should be able to delete a user account.
REQ-15	4	The administrator should be able to edit the stock price database.
REQ-16	3	The administrator could add some related news to the website.
REQ-17	4	User shall be not allowed to access the accounts of other users'.
REQ-18	4	User shall be not allowed to modify any data in the database.

Table 3.1 Functional requirements

3.2 Use Case:

The following Table 3.2 describes the actors and goals of each of them and the use case these goals are represented by.

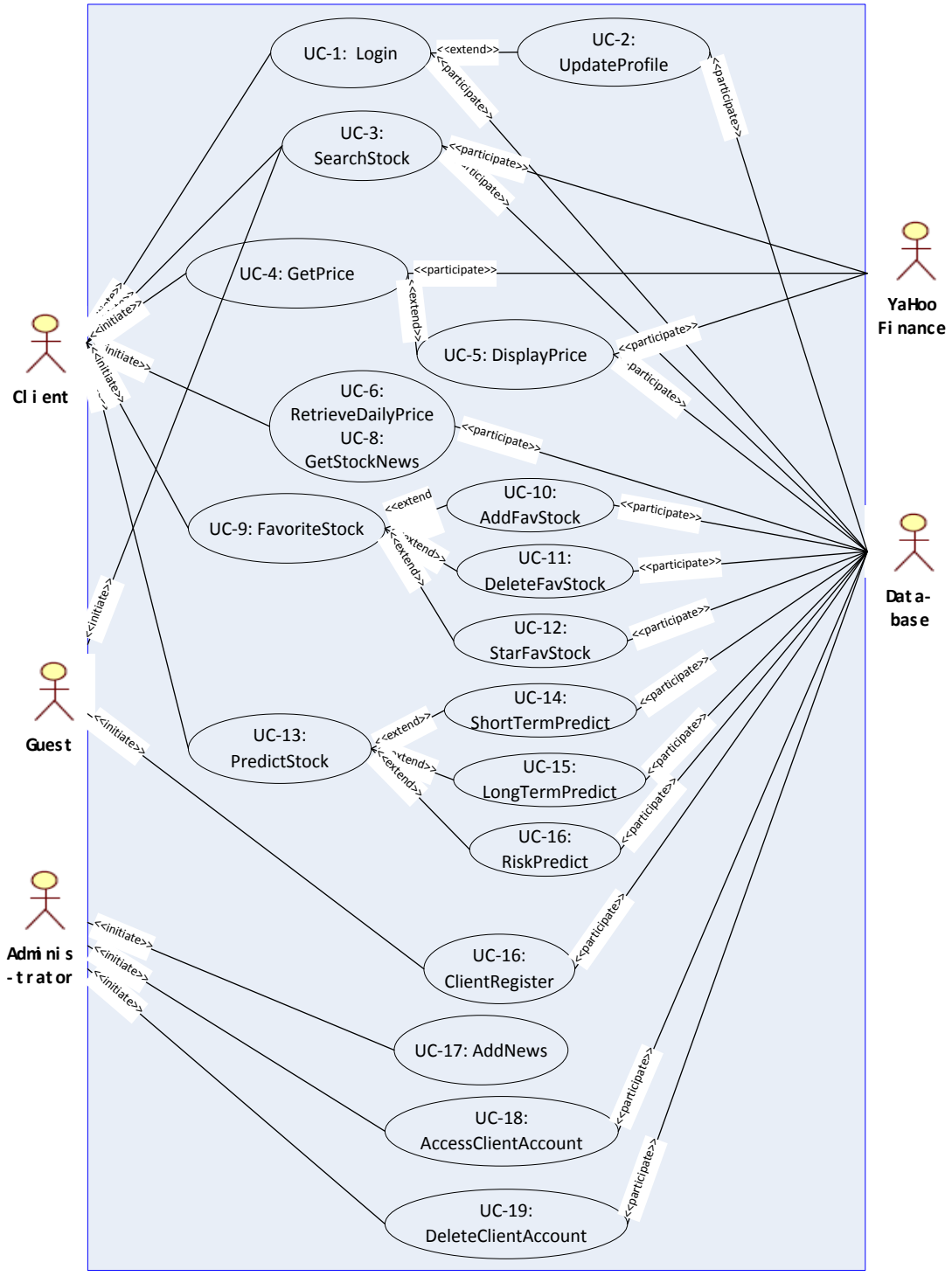
UC	Use Case Name	Actors	Goals	REQ
UC-1	Login	Client	the client can login the system using their username and password.	REQ8
UC-2	UpdateProfile	Client	The client can update their profile.	REQ6
UC-3	SearchStock	Client/ Guest	The users can search the stock by typing key words.	REQ1
UC-4	GetPrice	Client/ Guest	The users can extract the stock price via Yahoo Finance API, including real time and historical data.	REQ2
UC-5	StorePrice	Client	The client can store the price data to database.	REQ3

UC-6	RetriveDailyPrice	Client	The client can retrieve the daily price of the favorite stock from database.	REQ4
UC-7	DisplayPrice	Client	The client can let the system display the price of the stock.	REQ11
UC-8	GetStockNews	Client	The client can get important news about the stock.	REQ16
UC-9	FavoriteStock	Client	The client can keep an eye on some interesting stocks.	REQ6
UC-10	AddFavStock	Client	The client can add the favorite stock.	REQ6
UC-11	DeleteFavStock	Client	The client can delete the favorite stock.	REQ6
UC-12	StarFavStock	Client	The client can star the favorite stock and pay more attention on it.	REQ6
UC-13	PredictStock	Client	The client can be allowed to get the prediction of the stock.	REQ5
UC-14	ShortTermPredict	Client	The client can be allowed to get the short term prediction of the stock.	REQ5
UC-15	LongTermPredict	Client	The client can be allowed to get the long term prediction of the stock.	REQ5
UC-16	ClientRegister	Guest	The guest can register to become a client.	REQ7
UC-17	AddNews	Administrator	The Administrator can add the important news in the system.	REQ16
UC-18	AccessClientAccount	Administrator	The Administrator can access the client's account.	REQ13
UC-19	DeleteClientAcco	Admini	The Administrator can delete	REQ14

	unt	strator	the client’s account.	

Table 3.2 Use case

3.3 Use Case Diagram



3.4 Traceability Matrix

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17	U18	U19
R1			X																
R2				X															
R3					X														
R4						X													
R5													X	X	X				
R6		X							X	X	X	X							
R7																X			
R8	X																		
R9																			
R10																			
R11							X												
R12																			
R13																		X	
R14																			X
R15																			
R16								X									X		
R17																			
R18																			

3.5 Main Use Case Description

Use Case 1: Login

Initiating Actor: Client/Administrator

Actors Goal: The clients can login into the account

Participating Actor: Database

Pre-conditions: The Client/Administrator has not logged into the account

Post Condition: The Client/Administrator has logged into the account by providing the correct username and password

Success scenario:

-> 1. The Client/Administrator opens the login page and type the correct user name and password, then send the request

<- 2. The database compare the user name and password, send back the successful result

-> 3. The application will show the homepage of the client or administrator

Fail scenario:

-> 1. The Client/Administrator opens the login page and type the correct user name and password, then send the request

<- 2. The database compare the user name and password, send back the successful result

-> 3. The application will show the error information in the screen

Use Case 3: SearchStock

Initiating Actor: User

Actors Goal: The users can search the stock by typing key words.

Participating Actor: Yahoo Finance API

Pre-conditions: The application is open

Post Condition: The application will show the information about the stock in a new page if search successfully.

Success scenario:

-> 1. User type the key words of the stock and send the request to Yahoo Finance API

<- 2. Yahoo Finance API sends back the information

-> 3. The application will show the information in a new web page

Fail scenario:

-> 1. User type the key words of the stock and send the request to Yahoo Finance API

<- 2. Could not find the stock, Yahoo Finance API sends back the fail information

Use Case 4: GetPrice

Initiating Actor: User

Actors Goal: The users can extract the stock price via Yahoo Finance API, including real time and historical data.

Participating Actor: Yahoo Finance API, Database

Pre-conditions: The application is open

Post Condition: The application downloads the real time stock data or historical stock data, and stores in the database

Success scenario:

-> 1. User sends the request of downloading data to Yahoo Finance API

<- 2. Yahoo Finance API sends back the data and stores in the database

Fail scenario:

-> 1. User sends the request of downloading data to Yahoo Finance API

<- 2. The database can not store the data

Use Case 6: RetrieveDailyData

Initiating Actor: Client

Actors Goal: The clients can retrieve the daily price of the favorite stock from database.

Participating Actor: Database

Pre-conditions: The application is open and data are already stored in the database

Post Condition: The application retrieved the data from database

Success scenario:

-> 1. User sends the request of retrieving data to database

<- 2. Database sends back the data

Fail scenario:

-> 1. User sends the request of retrieving data to database

<- 2. The database can not send back the data

Use Case 13: PredictStock

Initiating Actor: Client

Actors Goal: The clients can get the prediction of selected stocks

Participating Actor: Database

Pre-conditions: The clients does not have any prediction of a particular stock

Post Condition: The clients get the prediction of a selected stock, including short-term and long-term prediction. They will also get a simple recommendation about the stock

Success scenario:

-> 1. User select a stock

<- 2. Database sends back the data of the selected stock

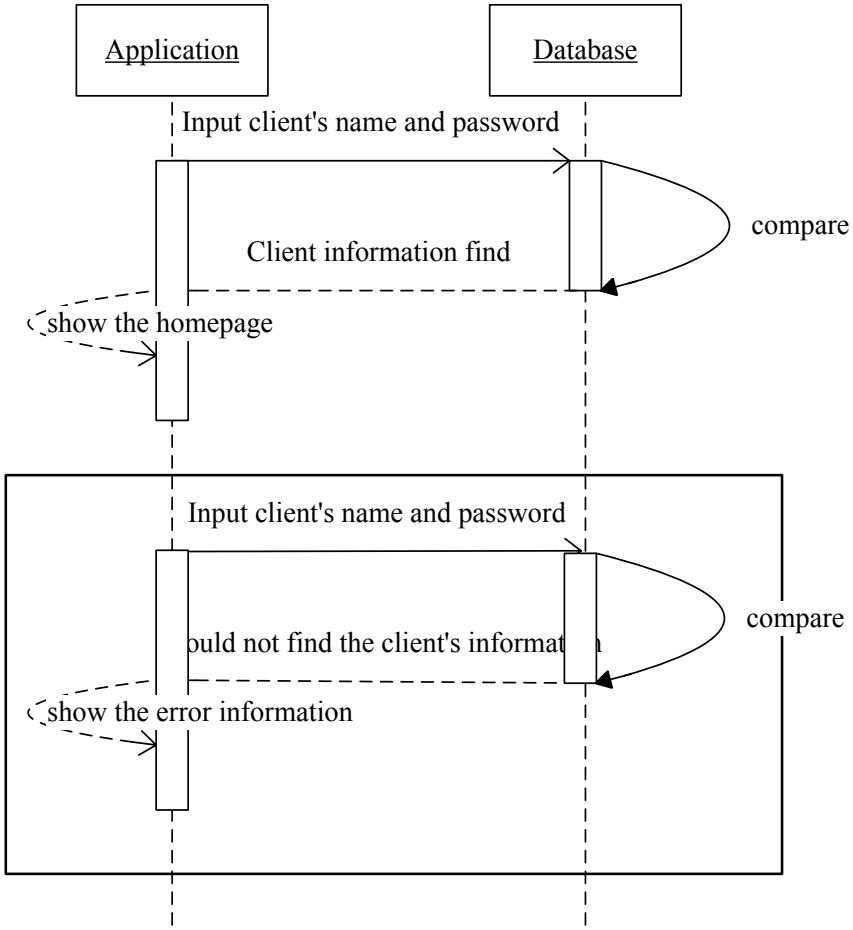
-> 3. System calculate the short-term and long-term prediction

<- 4. Send back the result and display in the page

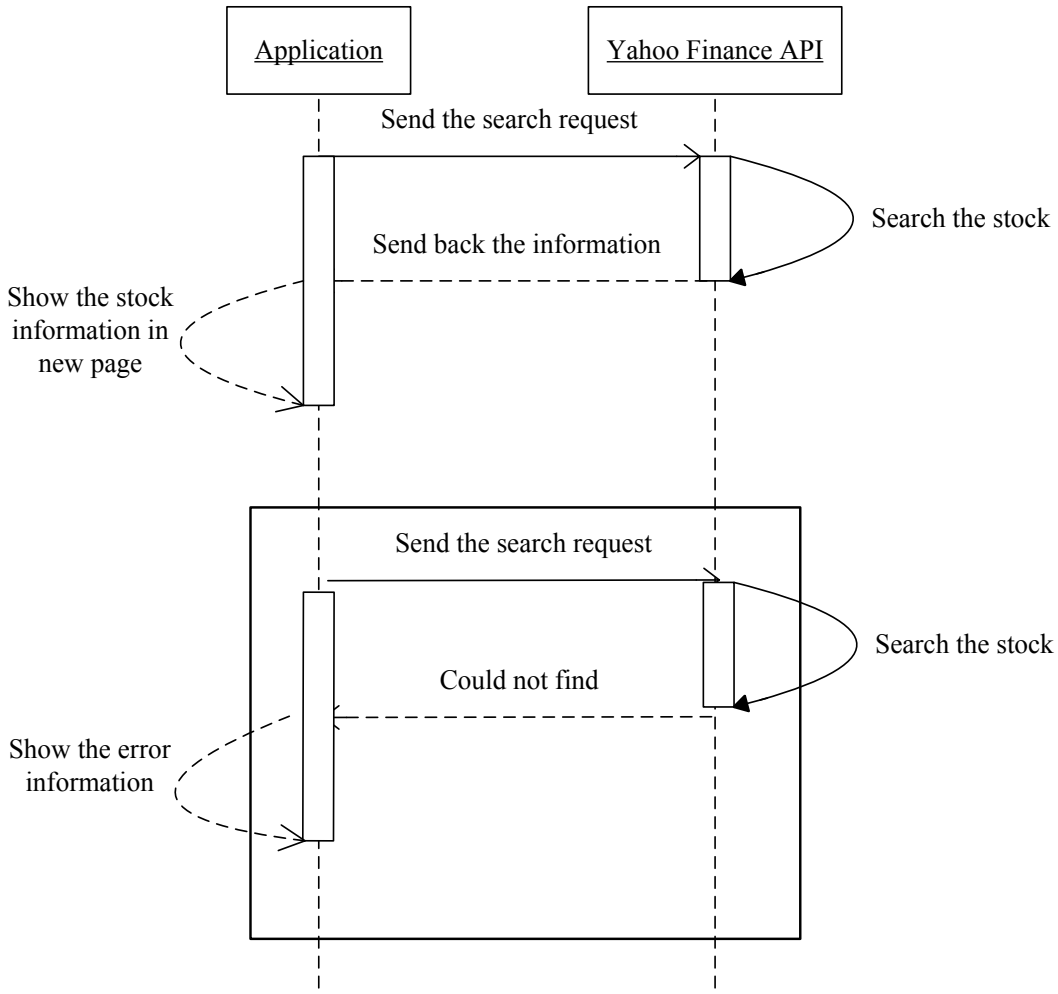
3.6

Sequence Diagram:

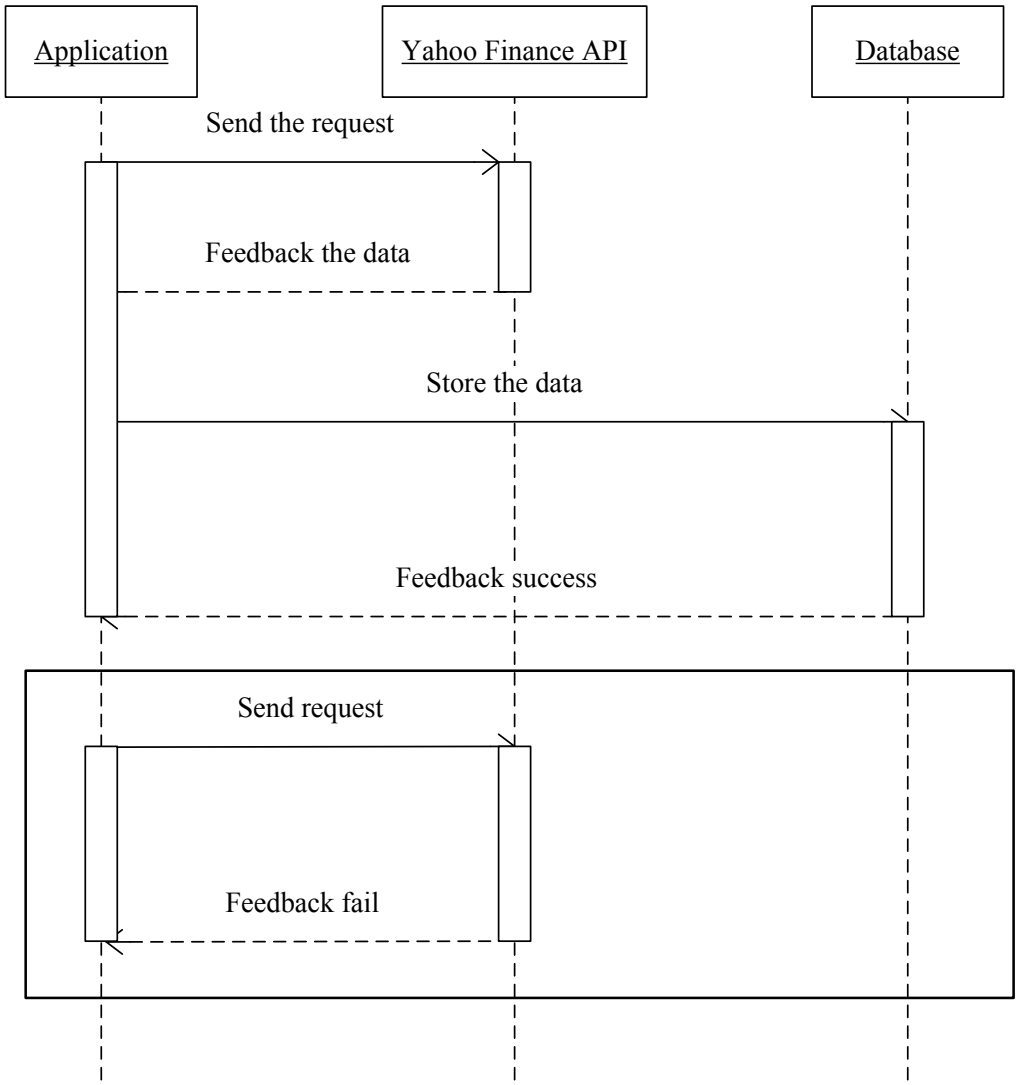
Use Case 1: Login



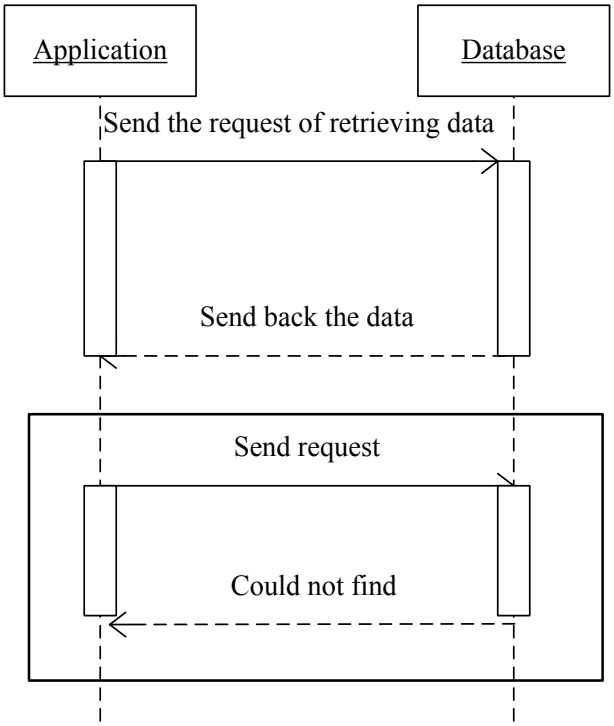
Use Case 3: SearchStock



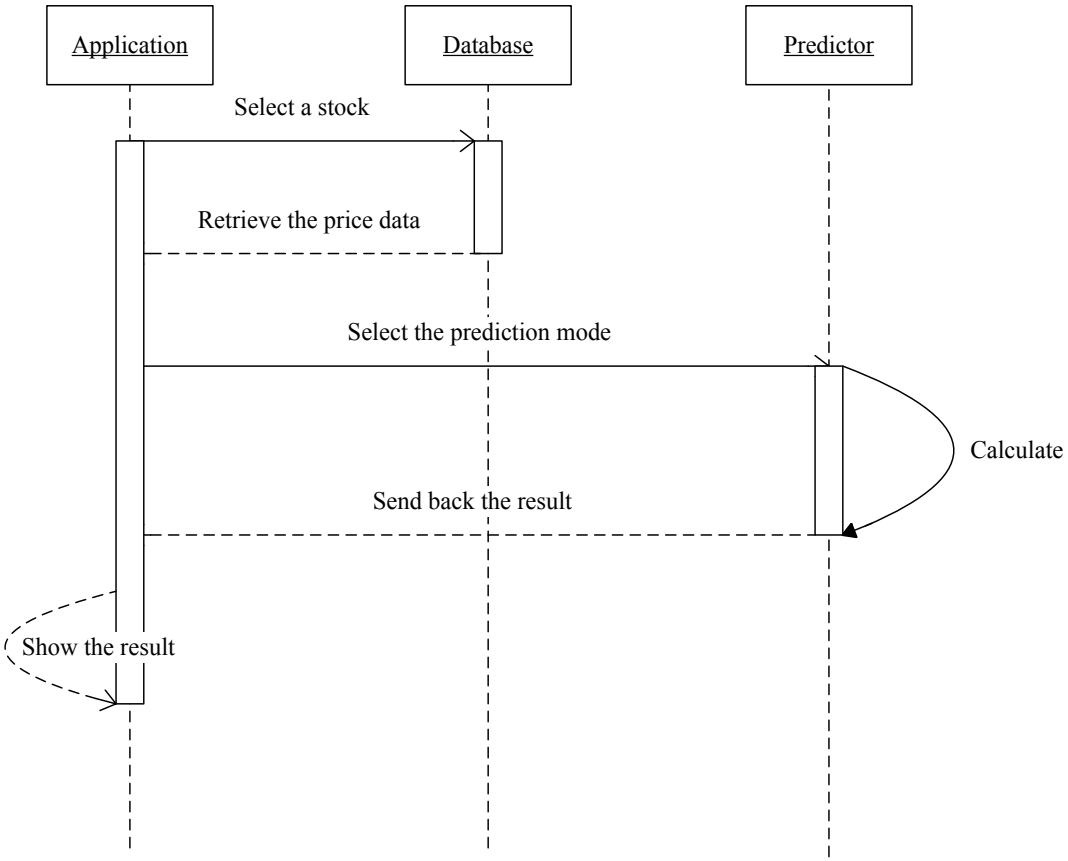
Use Case 4: GetPrice



Use Case 6: RetrieveDailyData



Use Case 13: PredictStock



3.7 Data collection

We use the python as our program language and SQLite3 as our database tool. The data were obtained from the Yahoo Finance API. We chose yahoo finance as it is free source of stock data and it is one of the most reliable sources. The stock data from Yahoo Finance are delayed by five minutes. After extracting the stock data, we will store the data in the local database, so we can use the data for the prediction. The Figu llustrates an f the ction s

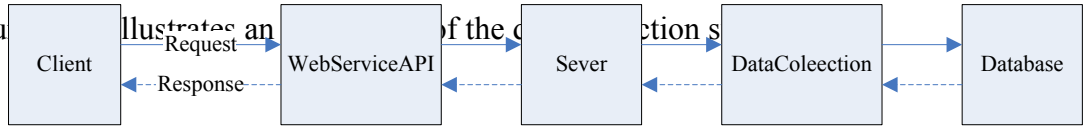


Figure 3.1 Block Diagram of the data collection system

4. Process for prediction

A detailed explanation of the field of statistical machine learning is beyond this article. In order to utilize techniques such as Logistic Regression, Linear Discriminant Analysis and Quadratic Discriminant Analysis we need to outline some basic concepts.

4.1 Supervised Learning Techniques

Supervised learning techniques involve a set of *known* tuples (*xi*,*yi*), $i \in \{1,...,N\}$, with *xi* representing the predictor variables (such as lagged stock market returns or volume traded) and *yi* representing the associated response/observation variables (such as the stock market return today). In this situation we are interested in *prediction*. Given future predictor variables we wish to estimate the responses from these predictors. This is in opposition to *inference* where we are more interested in the *relationship* between the variables.

4.2 Measuring Prediction Accuracy

The particular class of methods that we are interested in involves *binary classification*. That is, we will attempt to allocate the percentage return for a particular day into two buckets: "Up" or "Down". In a production forecaster we would be very concerned with the magnitude of this prediction and the deviations of the prediction from the actual value.

In such cases we can make use of the Mean-Squared Error, Mean Absolute Deviation and Root-Mean-Squared Error to provide an estimate of forecasting accuracy. The literature provides numerous other examples of forecasting accuracy measures.

In this instance we are only going to be concerned with the **hit rate**, which is simply the percentage of times that the forecaster achieved an accurate prediction (i.e. up when the day was up and vice versa). In later examples we will make use of a confusion matrix to determine the prediction performance on a class-by-class basis. In addition we will calculate the aforementioned values and incorporate them into our trading research process.

4.3 Forecasting Factors

A forecasting methodology is only as good as the factors chosen as predictors. There are a staggering number of potential factors to choose from when forecasting stock market index returns. In this article we are going to restrict the factors to time lags of the current percentage returns. This is not because they are the best predictors, rather it is because it is straightforward to demonstrate the process of forecasting on an easily obtained dataset.

Forecasting factor choice is extremely important, if not the most important, component of the forecaster. Even simple machine learning techniques will produce good results on well-chosen factors. Note that the converse is not often the case. "Throwing an algorithm at a problem" will usually lead to poor forecasting accuracy.

For this forecaster specifically, I have chosen the first and second time lags of the percentage returns as the predictors for the current stock market direction. This is a relatively arbitrary choice and there is plenty of scope for modification, for instance by adding in additional lags or the volume of shares traded. It is generally better to have fewer predictors in a model, although there are statistical tests available which can demonstrate the predictive capability of each factor.

4.4 Forecasting S&P500 with Logistic Regression, LDA and QDA

The S&P500 is a weighted index of the 500 largest publicly traded companies (by market capitalization) in the US stock market. It is often considered an equities "benchmark". Many derivative products exist in order to allow speculation or hedging on the index. In particular, theS&P500 E-Mini Index Futures Contract is an extremely liquid means of trading the index.

In this section we are going to use three *classifiers* to predict the direction of the closing price at day N based solely on price information known at day $N-1$. An upward directional move means that the closing price at N is higher than the price at $N-1$, while a downward move implies a closing price at N lower than at $N-1$.

If we can determine the direction of movement in a manner that significantly exceeds

a 50% hit rate, with low error and a good statistical significance, then we are on the road to forming a basic systematic trading strategy based on our forecasts. At this stage we're not concerned with the most up to date machine learning classification algorithms. Right now we're just introducing concepts and so we'll begin the discussion on forecasting with some elementary methods.

4.5 Logistic Regression

The first technique we will consider is logistic regression (LR). In our case we are going to use LR to measures the relationship between a binary categorical dependent variable ("Up" or "Down") and multiple independent continuous variables (the lagged percentage returns). The model provides the *probability* that a particular (following) day will be categorized as "Up" or "Down". In this implementation we have chosen to assign each day as "Up" if the probability exceeds 0.5. We could make use of a different threshold, but for simplicity I have chosen 0.5.

LR uses the logistic formula to model the probability of obtaining an "Up" day ($Y=U$) based on the lag factors ($L1$, $L2$):

$$p(Y=U|L1,L2)=e\beta 0+\beta 1L1+\beta 2L21+e\beta 0+\beta 1L1+\beta 2L2$$

The logistic function is used because it provides a probability between [0,1] for all values of $L1$and $L2$, unlike linear regression where negative probabilities can be generated in the same setting.

To fit the model (i.e. estimate the βi coefficients) the maximum likelihood method is used. Fortunately for us, the implementation of the fitting and prediction of the LR model is handled by the scikit-learn library.

4.6 Linear Discriminant Analysis(LDA)

The next technique used is Linear Discriminant Analysis (LDA). LDA differs from LR in because in LR we model $P(Y=U|L1,L2)$ as a conditional distribution of the response Y given the predictors Li, using a logistic function. In LDA the distribution of the Li variables are modeled separately, given Y, and $P(Y=U|L1,L2)$ is obtained via Bayes' Theorem.

Essentially, LDA results from assuming that predictors are drawn from a multivariate Gaussian distribution. After calculating estimates for the parameters of this distribution, the parameters can be input into Bayes' Theorem in order to make predictions about which class an observation belongs to.

4.7 Quadratic Discriminant Analysis(QDA)

Quadratic Discriminant Analysis (QDA) is closely related to LDA. The significant difference is that each class can now possess its own covariance matrix.

QDA generally performs better when the decision boundaries are non-linear. LDA generally performs better when there are fewer training observations (i.e. when needing to reduce variance). QDA, on the other hand, performs well when the training set is large (i.e. variance is of less concern). The use of one or the other ultimately comes down to the bias-variance trade-off.

As with LR and LDA, scikit-learn takes care of the QDA implementation so we only need to provide it with training/test data for parameter estimation and prediction.

4.8 Brief introduction of Python Implementation

We need to create a pandas DataFrame that contains the lagged percentage returns for a prior number of days (defaulting to five). create_lagged_serieswill take a stock symbol (as recognised by Yahoo Finance) and create a lagged DataFrame across the period specified:

```
def create_lagged_series(symbol, start_date, end_date, lags=5):
    ts = DataReader(symbol, "yahoo", start_date-datetime.timedelta(days=365), end_date)
     # Create the new lagged DataFrame
     tslag = pd.DataFrame(index=ts.index)
     tslag["Today"] = ts["Adj Close"]       tslag["Volume"] = ts["Volume"]
     # Create the shifted lag series of prior trading period close values
     for i in xrange(0,lags):
            tslag["Lag%s" % str(i+1)] = ts["Adj Close"].shift(i+1)
```

```
        # Create the returns DataFrame
    tsret = pd.DataFrame(index=tslag.index)
    tsret["Volume"]=tslag["Volume"]
    tsret["Today"]=tslag["Today"].pct_change()*100.0
        # If any of the values of percentage returns equal zero, set them to      # a small number
    (stops issues with QDA model in scikit-learn)
    for i,x in enumerate(tsret["Today"]):
        if (abs(x) < 0.0001):
          tsret["Today"][i] = 0.0001
       # Create the lagged percentage returns columns
    for i in xrange(0,lags):
        tsret["Lag%s" % str(i+1)] = tslag["Lag%s" % str(i+1)].pct_change()*100.0
    # Create the "Direction" column (+1 or -1) indicating an up/down day      t
    sret["Direction"] = np.sign(tsret["Today"])      tsret = tsret[tsret.index >= start_date]
```

The next helper function is designed to create a percentage hit_rate for each model, by eliminating duplicated code. It relies on the fact that the Logistic Regression, LDA and QDA objects have the same methods (fit and predict). The hit rate is output to the terminal:

```
def fit_model(name, model, X_train, y_train, X_test, pred):
    # Fit and predict the model on the training, and then test, data
    model.fit(X_train, y_train)      pred[name] = model.predict(X_test)
    # Create a series with 1 being correct direction, 0 being wrong      # and then calcul
ate the hit rate based on the actual direction
    pred["%s_Correct" % name] = (1.0+pred[name]*pred["Actual"])/2.0
    hit_rate = np.mean(pred["%s_Correct" % name])      print "%s: %.3f" % (name, hit_rat
e)
```

Finally, we tie it together with a __main__ function. In this instance we're going to attempt to forecast the US stock market direction in 2005, using returns data from 2001 to 2004:

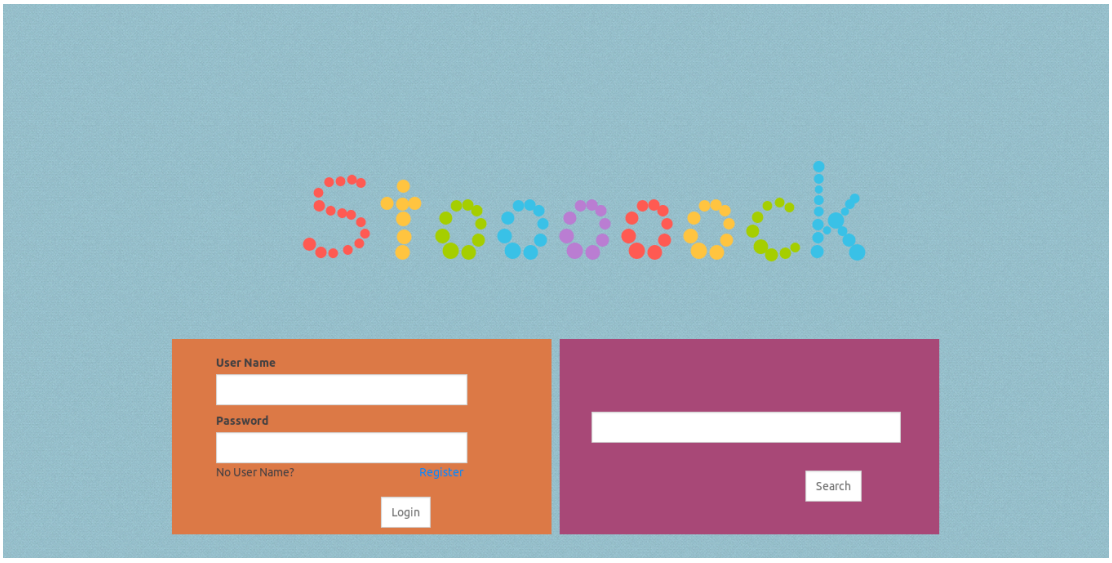
```
if __name__ == "__main__":
# Create a lagged series of the S&P500 US stock market index
        Snpret=create_lagged_series("^GSPC",datetime.datetime(2001,1,10),
        datetime.datetime(2005,12,31), lags=5)
 # Use the prior two days of returns as predictor values, with direction as the response
        X = snpret[["Lag1","Lag2"]]       y = snpret["Direction"]
 # The test data is split into two parts: Before and after 1st Jan 2005.
        start_test = datetime.datetime(2005,1,1)
# Create training and test sets
        X_train = X[X.index < start_test]
        X_test = X[X.index >= start_test]
        y_train = y[y.index < start_test]
        y_test = y[y.index >= start_test]
# Create prediction DataFrame       pred =
        pd.DataFrame(index=y_test.index)       pred["Actual"] = y_test
# Create and fit the three models
        print "Hit Rates:"
        models = [("LR", LogisticRegression()), ("LDA", LDA()), ("QDA", QDA())]
        For m in models:
                fit_model(m[0], m[1], X_train, y_train, X_test, pred)
```

5. User Interface and Functions Description

Our system contains many functions, and we will divide them into 3 parts: search, client part and administrator part. Also, we will show our interface when we introduce the function.

When we click into our website, we will see a very sharp welcome animation:

After the flash, we will see the welcome page for all the users, including the guests and clients:



Then we can start using the function provided by our system. First, we want to introduce the purple frame in the right--search function.

Search

Our program allow the all the users to search the stock by themselves, and the data searched by the clients(not guests) can be added in the database. The application send the search request to Yahoo API, so that Yahoo API can feedback the data which the user had searched. Also the data that user searched before can be stored directly in the database and the database can reply whether the data has been stored successfully or not.

The figure bellow is what the guests get after searching a stock:

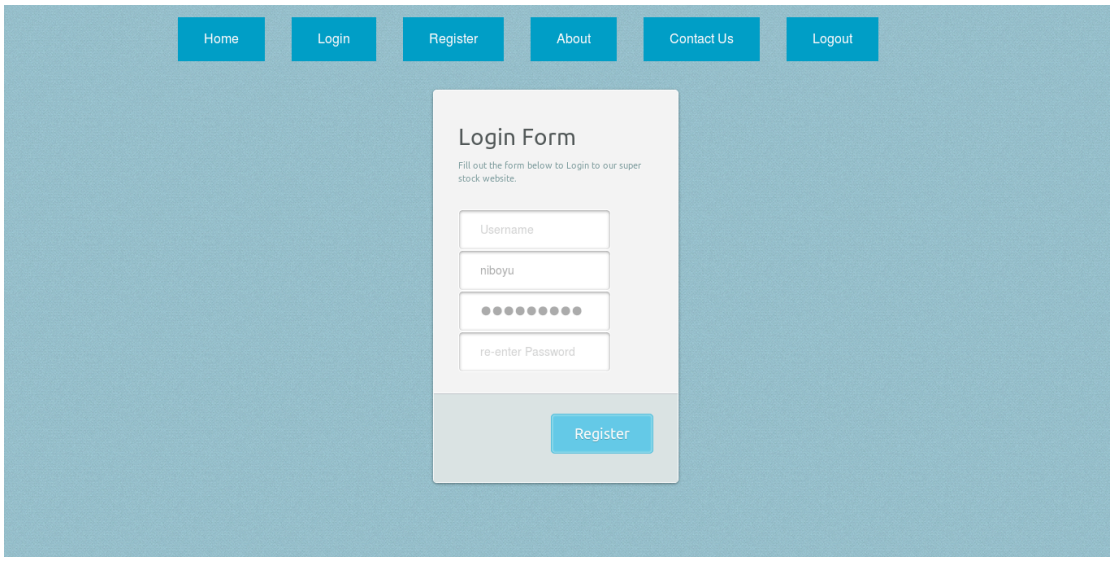
Then, we will introduce you the client part--which will provide you more functions in our system.

Client:

For the client function, our system implements not only the basic function to the clients like *Register, Login, Search Stock, Query* and *Display History Chart* from the basic requirements, but also the extended functions like *Favorite Stock* and *Stock News*.

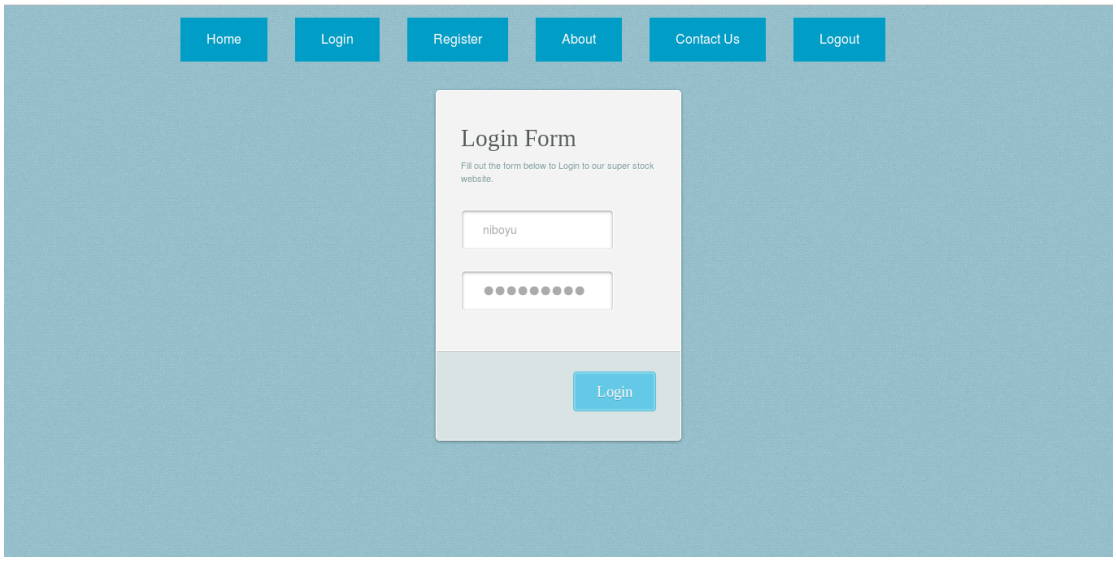
1. Register

As the guests can only use a very basic function in our system, we encourage the guests to register and be a client in our system. As a client, they can use all powerful functions in the system.

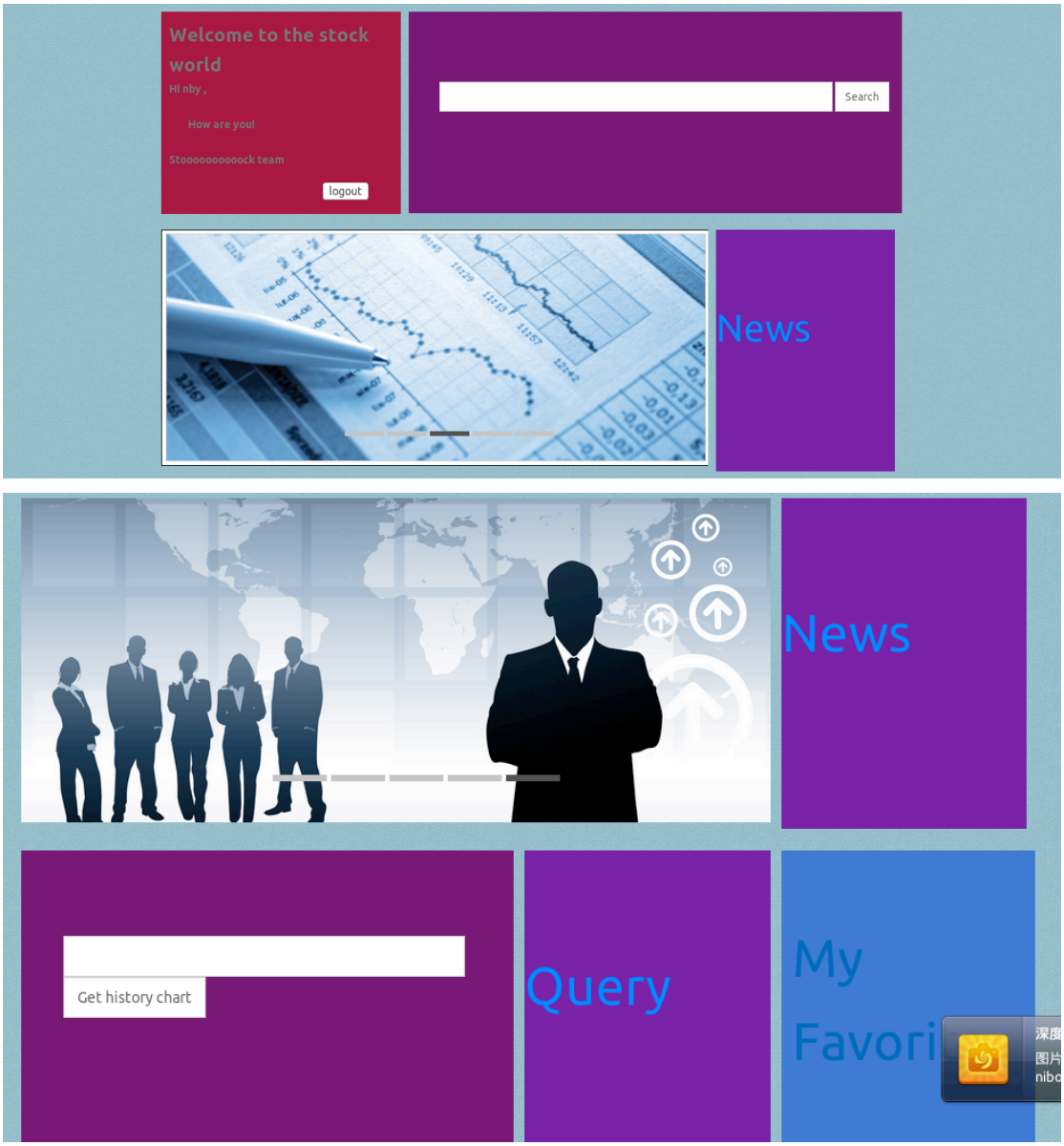


2. Login

When you become a client in our system, we recommend you to log in your account every time you open the website. After you log in the account, the website will jump to your home page, which contains all the powerful functions. You can access these functions just click the tag, it is easy.



After login the system, the website will turn to the home page for the client--a very cool home page.



The clients can access many functions in their home page, such as search the stock, get the news of the stocks, get history chart, query and my favorite stocks.

3. Search stock

The first basic function is searching the stock you are interested. The result is the same as the search function part introduced just now.

4. Query

The query function is a basic function in the project requirement. It implements the functions bellow:

1) Show the list of all companies in the database along with their latest stock price.

2) Get the highest stock price of Google in the last ten days (Of course, you can select other stock you are interested in.)

3) Average stock price of Microsoft in the latest one year (Of course, you can select other stock you are interested in.)

4) Lowest stock price for each company in the latest one year

5) List the ids of companies along with their name who have the average stock price lesser than the lowest of Google in the latest one year

---

the list of all companies in the database along with there altest price

The highest price of Google in the last 10 days : 534.81

Average stock price of Microsoft in latest one year: 35.751468254

Lowest price:

Google: 516.18

Microsoft: 31.15

Apple: 393.78

Amazon: 252.55

Yahoo: 24.07

IDs:4

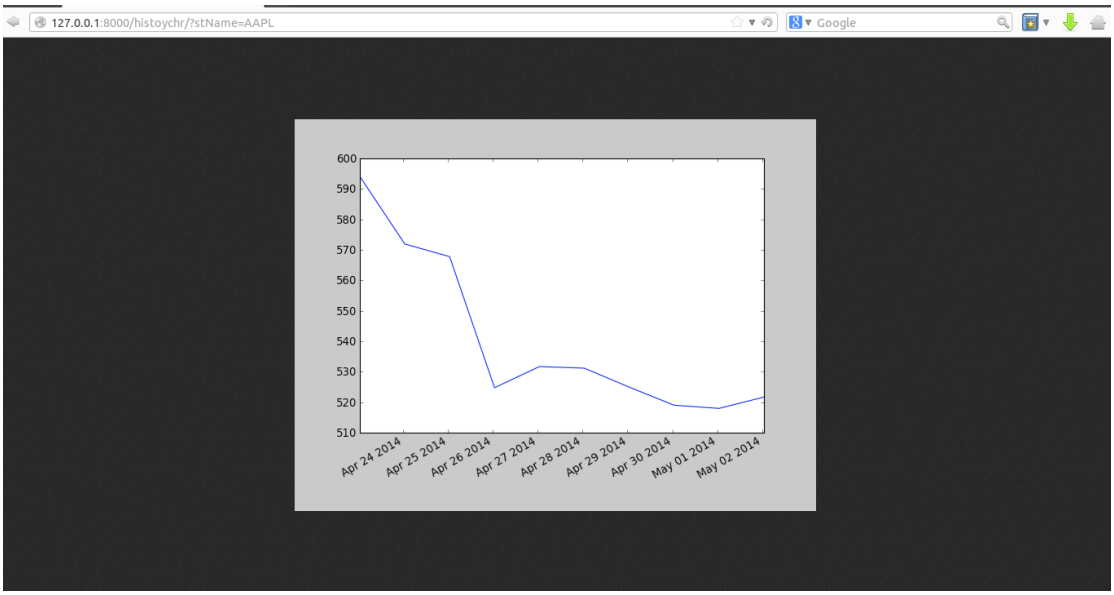
IDs:5

IDs:2

IDs:3

5. Display history chart

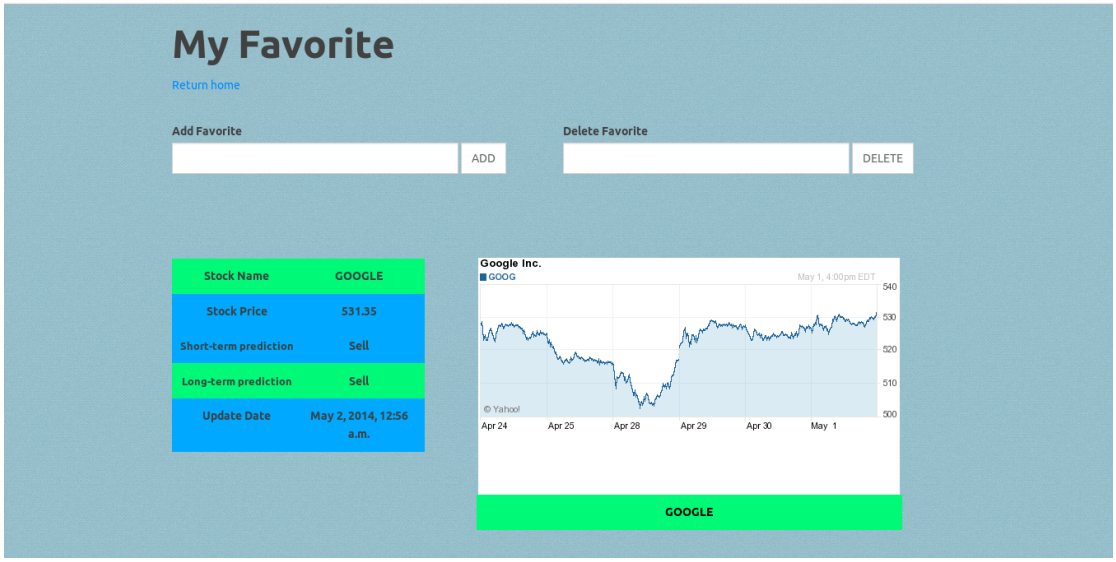
The clients can input a stock and get the history chart for a time period. The time interval of the figure is preset and can be changed by clients. The value interval can be automatically adjusted by the data. Also the chart in our system can be refreshed automatically which means when a new data comes, the chart will add the data in the chart automatically, without refreshing the page.



6. Favorite stock

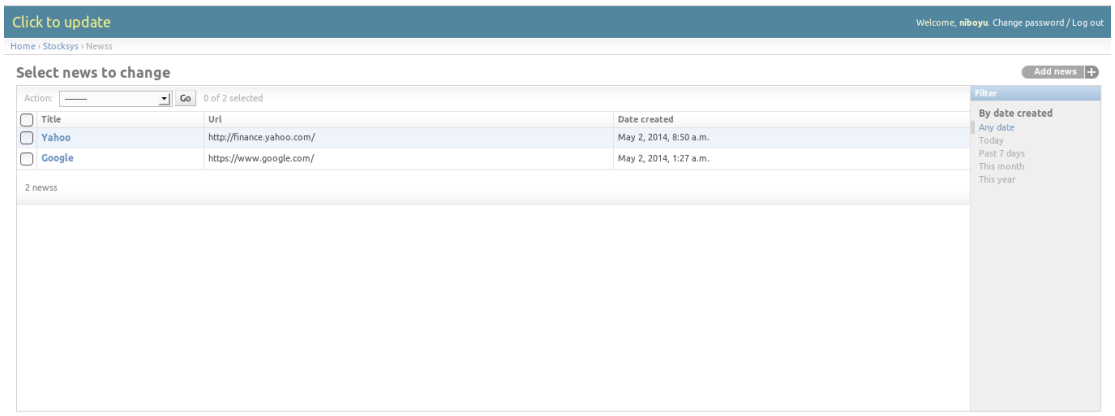
One of our creative functions is the Favorite stock which is a very useful function for all the clients. When the client click into the Favorite stock page, the information of the favorite stocks will display automatically. It is very convenient for the clients to check the stock they have bought or they want to buy.

The information will display in the page include the name, the current price, the short-term prediction, the long-term prediction and a technical analysing chart. The clients can add or delete a favorite stock in the bottom of the page. Everything is so easy.



7. Stock news

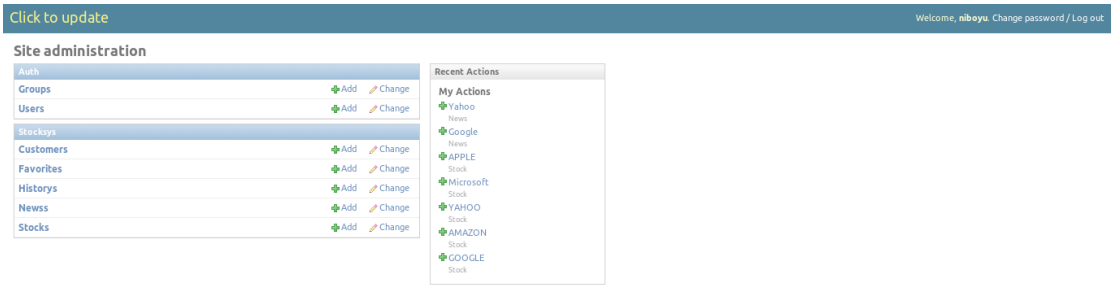
For providing more information to the clients, the administrator will put the important news, which may influence the stock market, in the News page. The clients can use the news as reference, combining the short-term and long-term prediction from our system, to make a better decision for their investments.



Administrator/Super user:

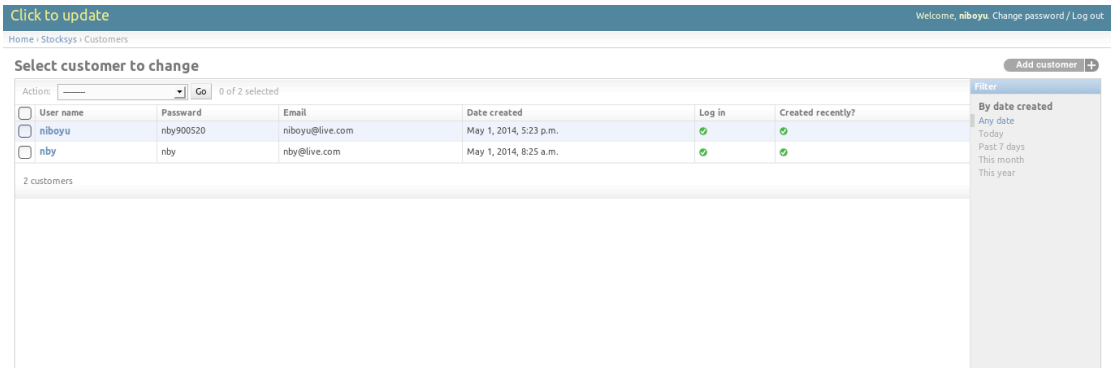
The administrators are the people who maintenance the system and protect the website can run in a long time steadily, update the information and website content timely.

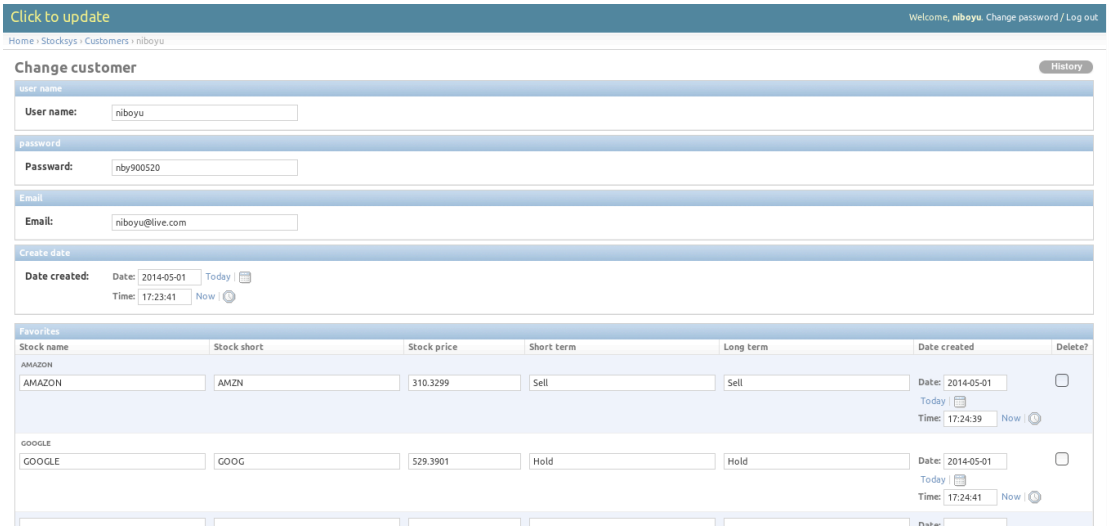
The part of administrator in our system includes four sessions: client management, stock management, news management and data update.



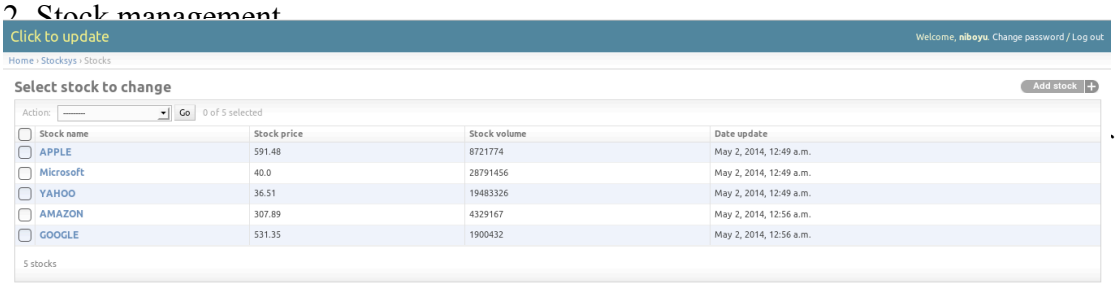
1. Client management

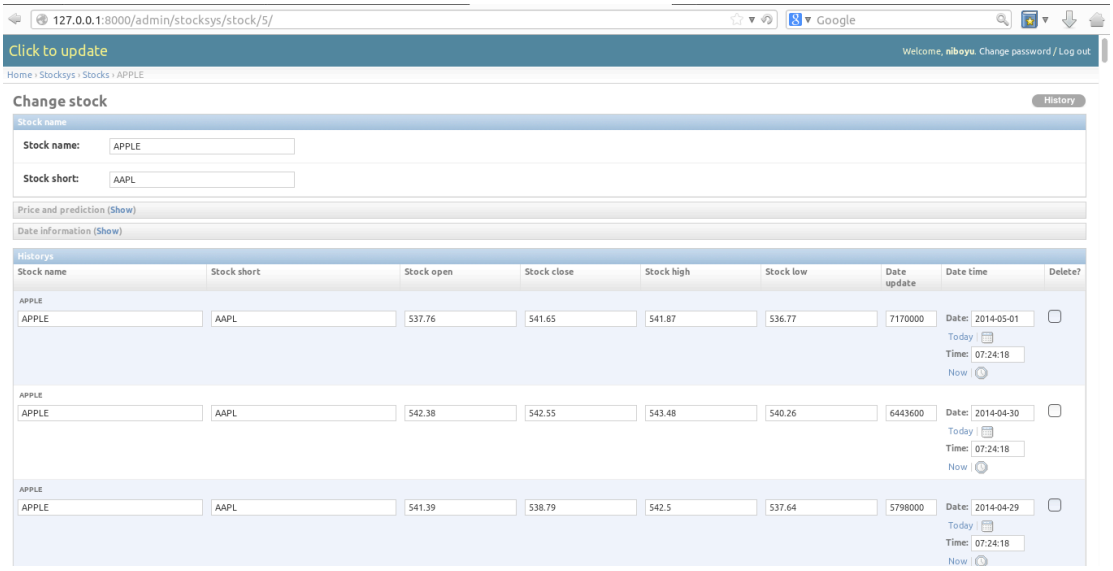
In the client management, the administrator can edit the profile of the client by adding, deleting or editing the information of all the clients. The administrator can also edit the profile of clients' favorite stock data.





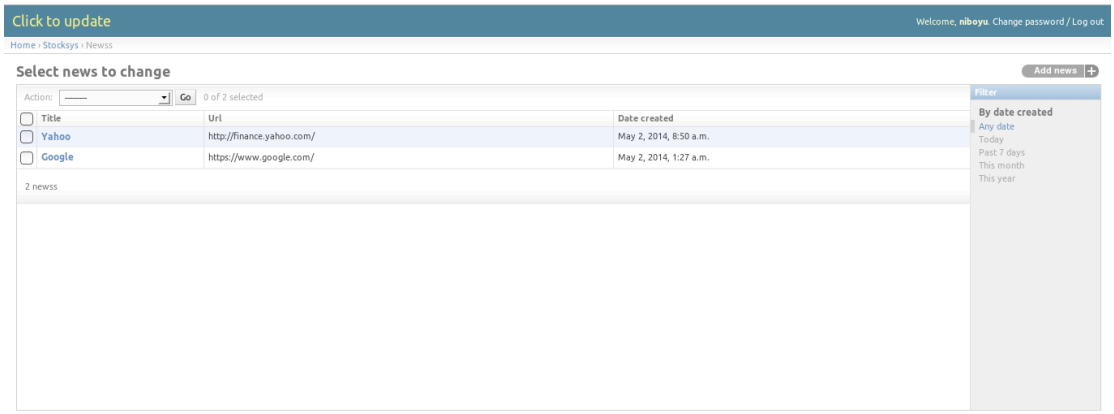
2. Stock management





3. News management

The administrator can add or delete the referenced news here.



4. Data update

Our system can update all the stock data by just clicking one button, it is easy for the administrators who don't know about python or PHP. It is convenient to maintenance the website.

6. Future Work:

- **Improving Notification System:** User will be allowed to use notification system to be notified regarding major changes in their favorite stocks. These notifications can be sent through email and text message for cell-phone and the user will be able to choose which one he/she likes to use.
- **Optimize Prediction Strategy:** There are many machine learning tools that we can benefit from in order to have more accurate result. Currently we are using two of them. Some examples that can be used are: Artificial Neural Network (ANN) and Support Vector Machines (SVM).
- **Mobile and Tablet Application:** Use of cell-phones and tablets for doing daily tasks can't be denied and our advanced stock website can also be expanded to a mobile application were users can access their favorite data even on their phones.
- **Advanced Data Search:** The application can be improved so similar stocks to what user have searched and favorite before, be shown for him/her in the suggestion page. For example if the user keep looking for Apple, Google and other Tech companies, the suggestion list will include Yahoo, Amazon, etc.

7. References:

[1] http://www.quantstart.com/articles/Forecasting-Financial-Time-Series-Part-1
[2] http://pandas.pydata.org/
[3] http://en.wikipedia.org/wiki/Technical_Analysis_Software_(Finance)
[4] http://www.django.org
[5] http://www.python.org
[6] The django book
[7] http://www.codecademy.com/
[8] http://www.w3schools.com/
[9] http://en.wikipedia.org/wiki/Python
[10] http://en.wikipedia.org/wiki/Django
[11] http://en.wikipedia.org/wiki/Html
[12] http://en.wikipedia.org/wikir/JavaScript