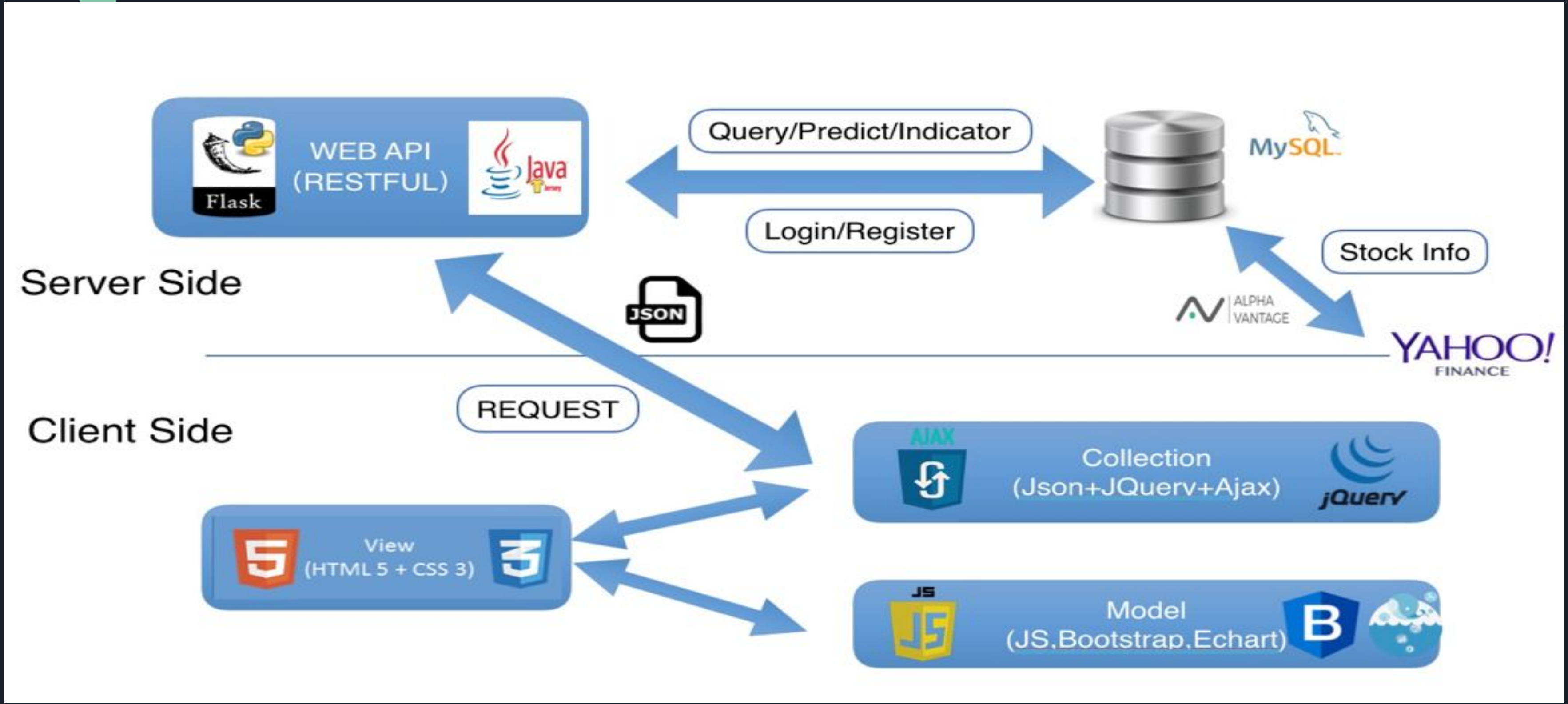
Web Based Stock Forecaster

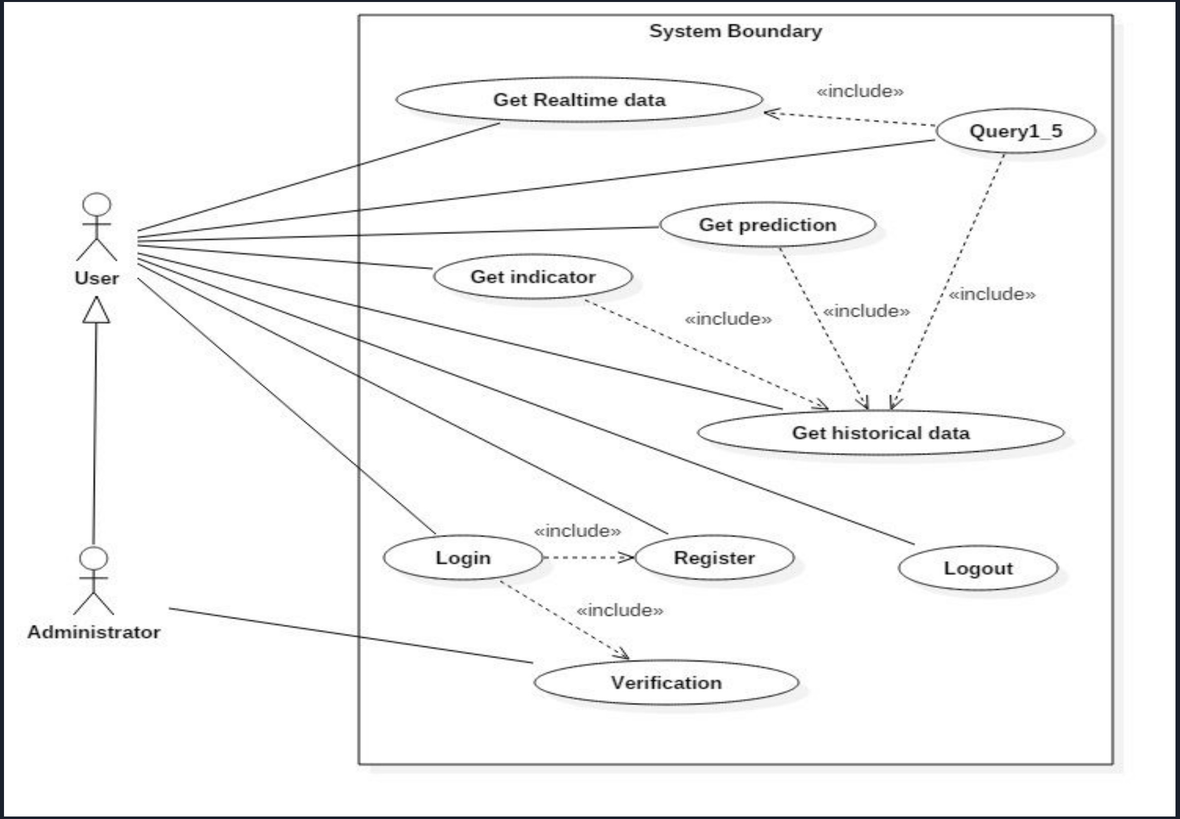
Department of Electrical and Computer Engineering

Ze liu
Tong Wu
Xinyu Lyu
Xinwei Zhao
Jay Borkar

Architecture Diagram



Use Case


System Boundary
Get Realtime data
«include»
Query1_5
Get prediction
User
Get indicator
«include»
«include»
«include»
Get historical data
«include»
Login
Register
Logout
Administrator
«include»
Verification

Programming Technologies

FrontEnd:

HTML;

JavaScript;

Bootstrap;

BackEnd:

RESTful;

JAX-RS;

Jersey;

Tomcat;

Flask;

MySQL;

Algorithms:

SVM;

LSTM;

ANN;

Bayesian Curve Fitting;

ROC, OBV, MACD;

Parse and save stock price;

Special Features

User’s Log in and Register:

We allow user to register an individual account. The accounts and passwords are stored in Database.

All logins will be checked in Database in case of duplicate accounts or incorrect passwords.

Favorite Companies:

Users can save their favorite companies by adding them in Favorite. We maintain a seperate table for each user in database.

Web Sources

- Used Alpha Vantage API to collect our raw data from Yahoo finance website for the prediction.
- We used ten companies stock, we have randomly chosen these famous companies: [“GOOG”,”AABA”,”AMZN”,”OPK”,”FB”,”TWTR”,”NFLX”,”TSLA”,”BABA”,”SPLK”]
- For each stock, we stored the latest day of real-time data and one year of Historical data.
- For each stock, the historical data contains Date, Open, High, Low, Close, Volume data. The real-time data contains the same headers and time slice is 1 minute.
- We stored data in MySQL database.
- There are two tables for each stock one for Historical data and other for Real-time data in our database.

Prediction Strategy

Long Term:

- **ANN**: Total 4 layers. Read the data from MySQL and split it into 5 sections as input. Used the last one of each section to train. we will train it for 500 times (epochs=500)
- LSTM: Composed of a memory cell, an input gate, an output gate forget gate. It can avoids the long-term dependencies. It remembers information for periods of time。

Short Term:

- **SVM:** Use create_engine class to read date and close price from MySQL. Then use the function svr to train the (date, price) and use the predict (Next Day) to get the value. kernel='rbf' (Radial Basis Function), C=1.0, cache size = 200
- **Bayesian Curve Fitting**：Use create_engine class to read the date from down to up automatically and save them as list. Put the list with parameter m = 9 (order of polynomial) into the BCF class and it will return an object. The getMx( ) of the object will return the value.

$$m(x) = \beta\phi(x)^{\mathrm{T}}\mathbf{S}\sum_{n=1}^{N}\phi(x_n)t_n$$

Indicators: On Balance Volume (OBV), Rate of Change (ROC), Moving Average Convergence Divergence (MACD)

Web Service

API: RESTful -- URL + HTTP

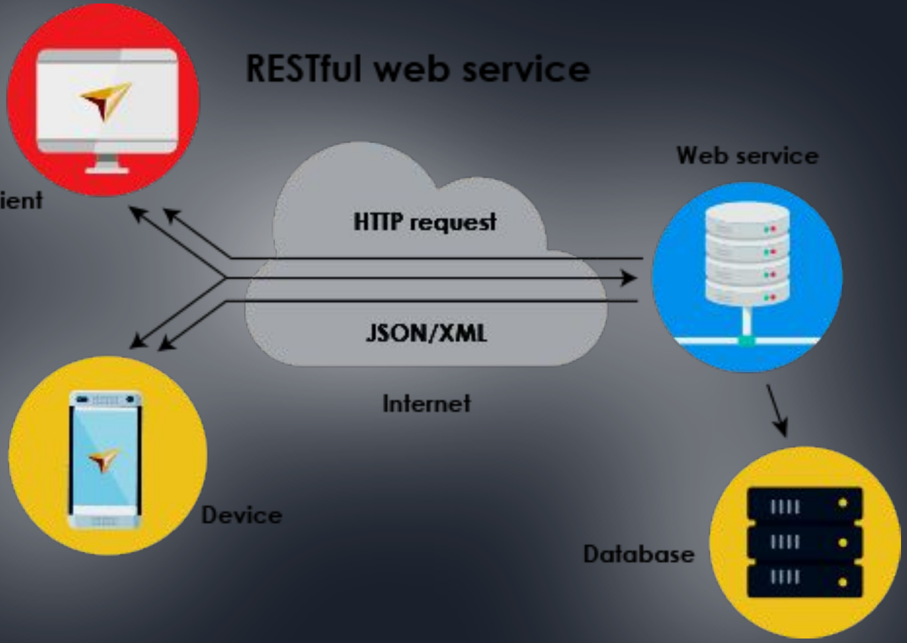
Framework & Languages:

- Flask & Python:

  User signup and login; Indicators.

- Jersey & Tomcat & Java:

  Query stock data; Price prediction.



Results

- Register
- Login and logout
- Get historical and realtime stock price
- Get the prediction of these stocks through ANN, LSTM, SVM and Bayesian curve fitting
- Show indicators such as OBV, ROC and MACD
- Implement the function of query 1 to 5

Please refer to the demo for specific implementation.

Thank you~