

## **Project Report**

1. Please list out changes in directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).

We carefully followed the original proposal, so in the end, we successfully designed a website that provides both the prediction for the stock market with the weighted growth rate on the homepage and the relationship between companies and people's sentiments toward them with the stored procedure. Our weighted growth rate is generated by linear regression, and our stored procedure outputs the most frequent sentiment word relating to the company. Users can register and follow the companies, yet they would not be able to receive updates about the stock or companies. Thus, the only slight change to our proposal is the function to send the related company or stock updates to users who follow the company.

2. Discuss what you think your application achieved or failed to achieve regarding its usefulness.

It achieved the main features of usefulness which is to predict the stocks growth rate based on the Google trend data of different words searched with the company names. It allows users to follow the companies they care about and checks the updates on predictions each day easily.

3. Discuss if you changed the schema or source of the data for your application

For the source of data, we still scrape the most recent day emotions data from Google Trend by searching keywords "company name" + "emotional word". Now we'd like to add the source of companies data, Nasdaq, which includes stocks' symbols, names, last sales, net changes, change rates, and market capitalization. We selected American stocks that have market capitalization greater than \$2 billion, downloaded the updated csv files everyday, and uploaded them to the database.

4. Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?

We add a table Purchase and a relation UserPurchases to keep track of the transactions of stocks for each user. Based on the purchase, we are able to produce a user rank, and see which users hold the highest total worth of stock. Besides that, we closely followed our original design of tables and implemented them successfully.

5. Discuss what functionalities you added or removed. Why?

We add two functions.

- (1) User Rank: find the top 15 users holding the highest total worth of all stock under the current price. This function is meaningful because users are able to compare their total stock asset with others. Note that this rank will not lead to any private problem because it only shows the userid, which is a series of randomly generated numbers.
- (2) Company Rank: find the top 15 companies whose stocks were sold for the highest total values, and find out how many followers these companies have. This function is meaningful because it displays the most popular companies for user references, especially for new investors.

6. Explain how you think your advanced database programs complement your application.

We designed a trigger that updates the number of followers of the company after the user clicks the “Follow” or “Unfollow” button of that company on our website. This trigger is built on the table “Companies Followed” which tracks which user is following which companies. Any insert or delete operation on the table would trigger it. This trigger complements our application because the change also appears in our second advanced query website, which demonstrates the top 15 companies whose stocks were sold for the highest total values and how many followers these companies have. The number of followers would better inform our users on the popularity of the companies and help them make better investment decisions. We also designed a stored procedure that outputs the companies with their weighted growth rate and their emotion word with the highest frequency of searches on Google Trends. The stored procedure is the foundation of our stock price prediction model, and it shows the users how we quantify the emotional words and how we connect emotional words with stock prices.

7. Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.

- login/logout

The challenge for user login and logout: it would be a problem if we only check user ids and passwords without saving info about the users. After searching online, we found that users can be stored in session when they login, and the session would be popped out when they logout. Also, if we

successfully find a user in session, we would create a global variable with g, to store the userid info and apply it to all user-specific functions.

- Stored procedures

Our stored procedure outputs the companies with their weighted growth rate and people's most frequently used sentiment word towards the company. It has been difficult to figure out the advanced query to calculate the weighted growth rate and weighted frequency of the sentiment word. We have weighted the growth rate and frequency of the emotion words by date, so it was hard to subquery the rate on each date and then calculate them altogether.

- Triggers

Our trigger would update the number of followers the companies have when users follow or unfollow the companies. The most difficult part is to connect the front end (the follow and unfollow button) with our backend database. When a user follows a new company, the status update function would insert a new row into the table CompaniesFollowed, and that would trigger the update on the table Company. We also need to pass the arguments of userid, companyid and follow status.

- Predicted GR

We used pytrend for getting the Google trend data, then we will create training and predict set with these data and stock prices from yfinance as labels, then we used linear regression from sklearn to create the model and predict.

8. Are there other things that changed comparing the final application with the original proposal?

We don't think there are any other changes from the original proposal. We tried our best to follow our plan and carry out all the implementations of the functions we proposed.

9. Describe future work that you think, other than the interface, that the application can improve on.

We can put the application on GCP, including functions that update and predict stock prices and growth rate as scheduled functions so that it would update at a time in each day automatically. In addition, we could update the Google trend data on our database so that we can decrease the number of requests to google to avoid being banned.

10. Describe the final division of labor and how well you managed teamwork.

Ziyang was responsible for building the login and logout functions of our websites, and keeping track of the stock prices data.

Erin implemented the stored procedure and the search function.

Allen built the machine learning model that gives the stock price prediction and designed the user interface for our application.

Yiqing wrote the trigger, and connected the login function with the database.

We all contributed to designing the subqueries and basic CRUD functions. The overall work is distributed evenly, and we have constant and efficient communication with each other all the time. If someone in our team bumped into some trouble, there is always help available from other team members.