SKANNER

Functional Specification

Dublin City University

CA4000

Author	Mateusz Koltun
ID	11366981
Course	Computer Applications and Software Engineering
Year	4th
Date	November 30th, 2018

Table of Contents

1. Introduction....4
1.1 Purpose....4
1.2 Overview....4
1.3 Glossary....5
2. General Description....5
2.1 Product Functions....5
2.2 User Characteristics and Objectives....8
2.3 Operational Scenarios....9
2.3.1 Running a query....9
2.3.2 Saving a query....10
2.3.2 Loading a query....10
2.4 Constraints....11
3. Functional Requirements....12
3.1 Accurate stock movement prediction....12
3.1.1 Description....12
3.1.2 Criticality....12
3.1.3 Technical Issues....12
3.1.4 Dependencies with other requirements....12
3.2 Sentiment Analysis of articles and blog posts....13
3.2.1 Description....13
3.2.2 Criticality....13
3.2.3 Technical Issues....13
3.2.4 Dependencies with other requirements....13
3.3 Machine learning and model training....13
3.3.1 Description....13
3.3.2 Criticality....14
3.3.3 Technical Issues....14
3.3.4 Dependencies with other requirements....14
3.4 Articles and blogs interface....14
3.4.1 Description....14
3.4.2 Criticality....14
3.4.3 Technical Issues....14
3.4.4 Dependencies with other requirements....14
3.5 Finance and stock data interface....15
3.5.1 Description....15
3.5.2 Criticality....15

3.5.3 Technical Issues....15
3.5.4 Dependencies with other requirements....15
3.6 User interface....15
3.6.1 Description....15
3.6.2 Criticality....15
3.6.3 Technical Issues....15
3.6.4 Dependencies with other requirements....16
3.7 Save and Load query meta data....16
3.7.1 Description....16
3.7.2 Criticality....16
3.7.3 Technical Issues....16
3.7.4 Dependencies with other requirements....16
4. System Architecture....17
5. High-Level Design....18
6. Preliminary Schedule....19
7. Appendices....19

1. INTRODUCTION

1.1 Purpose

The Purpose of this document is to give an outline of the fourth-year project - Skanner. This functional specification is used to give an overview of how the final product operates on the technical and logical level. It describes the development of the application as well as technologies used to create it.

The document summarizes the research and ideas gathered during the design stage. Information contained further, define the architecture and functionality of Skanner. All features and functions are clearly defined and illustrated to provide a better understanding of the software. Below, you will also find outlined challenges and constraints which will be encountered during the development, as well as function and software requirements which must be met for the product to work correctly and fulfil its purpose.

1.2 Overview

The ability to predict the movement of stock market is considered an important part in investing. People had been trying to predict share's price since the beginning of the stock market [1]. However, the task is not as simple and easy as it may look at first. The movement of stock in the short term is determined by multiple of factors and hundreds of variables. The price of the stock tends to go up, when demand for it increases and it goes down if it decreases. The demand increases, when there are more people interested in buying the stock than there are who wants to sell their shares. Some investors, if not the majority, make their investment decisions based on the condition of the particular market place, the company's earnings, their business model and actions taken by them, as well as their competitors. One of the best sources of this information are blogs, news, articles, social media posts and podcasts. It's safe to make an assumption that investors potentailly will purchase particular stock if they were to find out the company, who owns the stock recently has made great business decisions and has taken a lead in their market. With that said, it can be assummed that articles and blogs essentially influence indirectly the stock market and its movement. This here is a hypothesis which drives this project's nature.

Skanner is a software application, which intends to assist a user with making the decision whether to invest or not to invest in the particular stock. The program is designed to help with short term investments. It focuses on predicting share price behavior within next one to three days. The application bases its decision by analyzing news articles and compare the results against the historical share price. Firstly, Skanner learns how the same type of articles and their sentiment had the indirect impact on the stock price in the past. Then, the gained knowledge is applied to the current articles and the final result is derived out of four possible outcomes. The user finds out whether the investment is recommended or not recommended with four levels of emphasis:

- highly not recommended,
- not recommended,
- recommended,
- highly recommended.

There are two essential components of Skanner, which would directly affect how well the prediction works. The first one is the sentiment analysis software and the model used to score articles. The other one is a component responsible for machine learning and a model which are used to find correlation between article analysis and stock's past trends. The behavior of both will have a significant effect on the final result and probably are the most important functions in this system.

1.3 Glossary

Skanner	Proposed software application - this project.
NLP	Natural Language Processing

2. General Description

2.1 Product Functions

The concept of Skanner is to provide an automated solution to evaluate given market place using articles and blogs. Once the user enters required parameters, the

stock ticker symbol and tag tokens, which are used to retrieve articles, the application starts its main procedure. Firstly, it validates if given stock exists and see how much of a historical data is available to download. There could be a case, where the particular stock is fairly new in the market, therefore Skanner may have to reduce the training dataset to minimal. In the worst case scenario prediction may not be able to happen at all, because there will be not enough sufficient data to analyze, therefore the program would exit, notifying the user about the constraint.

In the other hand, once stock data is retrieved, Skanner starts to fetch all type of articles and blogs by querying news feed API with previously specified tags. The articles then are divided into groups based on the published date. Each group by default has a span of three days. The date ranges can be specified by the user prior starting the procedure, this way different type of queries can be used for more in depth analysis.

The next stage of execution is where the magic starts to happen and deep analysis takes over. Skanner, using open source Stanford CoreNLP software, begins the sentiment analysis on all articles. Modified or custom model must be used for this task rather than the one provided by Stanford, because articles are usually written in more objective manner than comments or reviews, which makes the original model not a good fit for this purpose.

At this point, Skanner has obtained share price historical data, related to the query articles and had evaluated them by analyzing their sentiment and assigning them an appropriate score. Now, the application presents its findings in the form of graphs and tables. The user has the ability to review the sentiment analysis results and modify them accordingly to assign articles a new score which they think would fit better. Ideally, there would be a mechanism in Skanner, so it can learn from the user's input to improve its sentiment analysis model and algorithm. Once the user is happy with sentiment scores, they can proceed to the next step and start the learning procedure. The learning procedure allows Skanner to analyze the obtained data and train its prediction model to fit this in particular case only. In other words, model used for prediction is trained each time a new query/execution is ran. This strategy is used to obtain more accurate results for the given query. Even though this means that the execution takes longer, it is necessary, because correlation between stock price and articles may vary from query to query. Skanner uses only two variables for its prediction, the sentiment scores and the price of the share at certain dates. However,

if this might not be enough and predictions are not going to be accurate, next variables will have to be introduced into the equation. Potential candidate is a number of ‘mentions’ of the company on social media (e.g. Twitter). Adding an extra attribute may improve accuracy of prediction as long as there is correlation between the two but it also adds complexity to the models and training algorithm.

Skanner trains its prediction model on a portion of the created dataset and uses the rest of it for validation. The training procedure repeats itself by default number of times, unless specified otherwise. At each of these iterations the training approach slightly changes until the validation completes with the acceptable result or until the numbers of attempts runs out. The portion of the dataset used for the training and the validation can be specified by the user, the same way as the number of attempts Skanner takes to train its prediction model.

At the very end, the program takes the most recent articles and applies the last successful model to derive the final answer to the question whether to invest in the stock or not. The answer is obtained from the combination of few things - how likely the share’s price may rise and potentially by how much. So, for example, if Skanner concludes that the stock has 75% chance to rise by 10% the final answer would be classified as highly recommended. On the other hand, if there is 50% chance for the stock to remain at the same price within the next days, the final result would read “not recommended”.

Skanner allows the user to save all results into a file and store it on the hard disk. The same file then can be used to load data back into the program which allows the user to continue their work. Graphs and tables can also be exported into .PNG and .CSV files respectively.

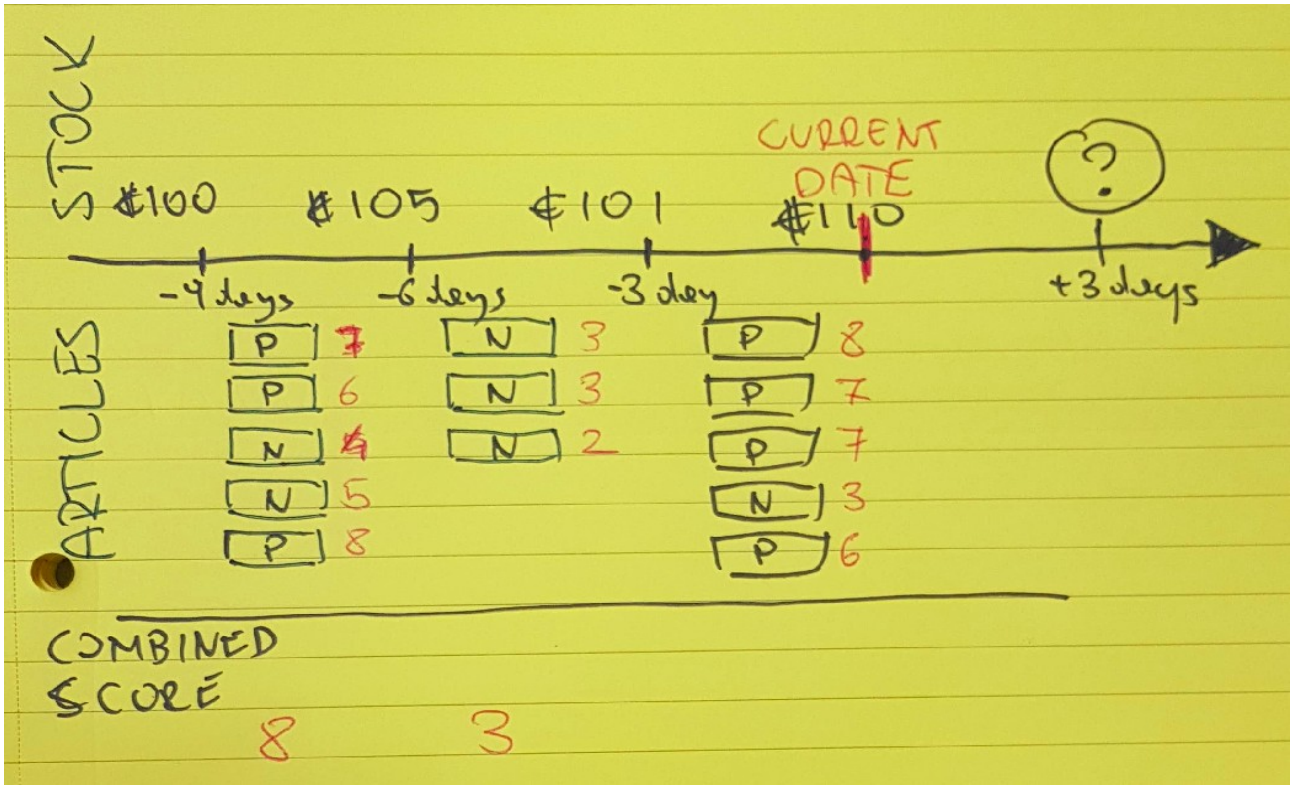


Figure 1. Logical representation of data.

On the figure 1 is shown how logically sentiment analysis results and stock data are going to modeled. This representation is going to be fed to machine learning module to analyze and train the model.

2.2 User Characteristics and Objectives

Skanner is targeted towards any demographic group. From young amateurs, through skilled professionals, to non-technical elderly. Its target audience is general public, but mostly people who work with stock market and its investments. This could be a broker, a stakeholder or an investor, who work in an office and trade shares professionally for living. On the other hand, a user could be a simple man who saved a bit of capital and wants to buy their first share. With that said, users' expertise with software systems may vary and this fact needs to be taken into the account. This is why, installation process of Skanner should be simple and straight forward, so users with different technical skills could follow and complete it successfully. Navigating through and using Skanner shouldn't be complicated either. Its design should accommodate users with minimal technical knowledge, therefore low expertise with software systems should be expected.

Even though users may vary a lot and come from different backgrounds, they still have one thing in common – their goal. Their goal is always going to be similar, to find out if a short-term investment in a particular stock will pay off or not. User wants to know the answer to that question so it helps them making the final decision whether to invest. The answer doesn't have be given at the instant, but it must be derived in a timely fashion. Users would expect to have it within an hour, after all - time is money. Depending on specified properties, analysis may take significant amount of time. Therefore, it would be useful to provide an estimate query time to a user prior executing the query so it's possible to modify properties if required. For the same reason, it's important a user be able to see the progress of a running query. Skanner should display the progress made in a form of a bar or any other similar indicator. There should also be information provided about currently undergoing step of the execution, elapsed time and estimate completion. Users would also expect to have an ability to save their work and findings, so they are able to continue their work from where they left. Although, the above cannot be omitted during the development, the most important thing for a user is to understand how the final recommendation been derived. Graphical representation of the analysis needs to be shown to users at the end of execution, so validation can be performed by themselves if desired. After all, there could be high stakes involved in the related investment, and extra verification for sure won't harm anyone.

2.3 Operational Scenarios

2.3.1 Running a query

Title	Running a query
Preconditions	1. Skanner built/installed successfully. 2. The program started.
Flow of events	1. User enters stock ticker symbol into the form. 2. User enters tags used to fetch articles. 3. User initiates the query. 4. Skanner pulls historic stock data. 5. Skanner fetches relevant articles and blogs. 6. User initiate articles analysis. 7. Skanner analyzes articles and assigns them a sentiment score. 8. Skanner displays results of analysis.

	9. User initiates prediction procedure. 10. Skanner trains the model. 11. Skanner presents results in a form of graphs and tables. 11. Skanner derives its final answer.
Extensions	2a. User adjusts advanced properties of the query. • Estimate query runtime updates. 3a. Stock symbol cannot be found. • User is notified about the problem. • The process must be repeated. 4a. Not enough stock data found. • The exception is thrown. • Query must be adjusted. 5a. Number of articles returned is not sufficient. • The exception is thrown. • Query must be adjusted. 8a. User modifies the sentiment score for articles. 10a. Skanner fails to train the model. • User is notified about the reason.
Post conditions	Skanner provides the final result to the user.
Requirements	Internet access

2.3.2 Saving a query

Title	Saving a query
Preconditions	Query initiliazed
Flow of events	1. User clicks the **SAVE** icon. 2. User specifies the location and confirms the procedure. 3. Skanner saves the file on the drive.
Extensions	3a. Skanner fails to save to a file. • Not enough space found on the disk. • File permission exception.
Post conditions	Query meta data saved to a file.
Requirements	Enough space on the hard disk.

2.3.2 Loading a query

Title	Loading a query
Preconditions	Query saved.

Flow of events	1. User clicks the **LOAD** icon. 2. User selects the file to load. 3. Skanner loads meta data to the program.
Extensions	2a. Skanner cannot load or find the file. • File permission exception is thrown. • File not found. • File format not recognized.
Post conditions	Query data loaded to the program and ready to use.
Requirements	Read permissions on the save file.

2.4 Constraints

There are few ways in which performance or more likely the effectiveness of Skanner could be impacted. Performance in this case is not a big deal, because users don't mind waiting a bit longer to get the answer, as long as it is reliable and accurate. In other words, users are willing to sacrifice performance over effectiveness, so prediction can take longer to be derived, but at the end it needs to be precise.

So, an inaccurate sentiment analysis of articles can be the factor, which affects the effectiveness of Skanner. If our scoring mechanism of news and blogs is off, then our assumptions and logic is not going to apply. Analysis must be somewhat accurate because it's a fundament of the whole software.

Another constraint in the system is the fact that Skanner treats all the fetched news and blogs the same way. The source of news articles doesn't provide information about how many times was the article read over time or how many clicks it gathered. These numbers could be really useful during scoring, because it would introduce another variable for machine learning – an importance value of articles. Without it, Skanner considers an article which hasn't been read by anyone the same way as the most popular article of this scope. This is something that needs to be taken into consideration during development of prediction component.

As mentioned before, query runtime is something that cannot be forgotten. Since Skanner trains its model for every query, runtime can become problematic very quickly, especially if query properties are changed to accommodate more data. Let's say a user increases the range of historical data to be pulled. So, instead of looking into last three months of articles and finance data, users decide to obtain a year worth of data. On top of that, they reduce the span of grouping articles by date, to periods of one day instead of three days. This type of query definitely takes longer to execute, because the dataset is a lot bigger and there are more coupled groups to be analyzed.

A potential solution to this problem would be introducing caching mechanism, which essentially would save already computed once model into a file or perhaps on the cloud so it can be used in the future and additional computation can be avoided. Another viable workaround would be moving the computational part of the software onto the cloud, where more resources are available. This approach however, would require additional costs which would introduce another constraint.

When it comes to hardware and software constraints, Skanner needs to be able to run on any computer therefore any required dependencies must be shipped together with the application. It should be able to run on most operating systems, this includes Mac OS, Linux and Windows.

3. FUNCTIONAL REQUIREMENTS

3.1 Accurate stock movement prediction

3.1.1 Description

The main purpose of Skanner is to predict **short term** behaviour of a particular stock. The application must be able to determine accurately if the investment in the particular stock is recommended and will pay off or not.

3.1.2 Criticality

This requirement is basically the reason why the application is created in the first place. Not meeting this requirement doesn't necessarily mean the project fails entirely. It can mean a lot of things, even if other components are implemented properly the initial hypothesis could be wrong. What if news articles and blogs do **NOT** influence the stock price movement at all?

3.1.3 Technical Issues

As described above in the constraint section 2.4, there are few factors that may affect this functionality. It comes down to implementing properly directly linked components which makes up this software.

3.1.4 Dependencies with other requirements

This requirement is linked directly to the sentiment analysis and the machine learning components. First one is for scoring articles and the other one is used to derive the prediction on constructed dataset, both listed below.

3.2 Sentiment Analysis of articles and blog posts

3.2.1 Description

Sentiment analysis is the process of computationally identifying and categorizing opinions expressed in piece of text, especially in order to determine whether the writer's attitude towards a particular topic or product is positive, negative or neutral.

3.2.2 Criticality

This requirement is considered to be a fundament on which Skanner mostly bases its prediction, therefore it is classified as essential for the system to work.

3.2.3 Technical Issues

The most common application of sentiment analysis in industry is to derive people's opinions about products and services. Whether these are reviews of movies or comments on blogs, sentiment analysis thrives to classify author's opinion. As difficult as this sound, there are available models out there to do it in somewhat accurate way. This is possible, because most reviews contain the opinion about the topic, on the other hand news articles are commonly written in the objective manner, making sentiment analysis nearly impossible using the same model.

If the sentiment of articles is evaluated inaccurately then the dataset won't represent ideal scenarios and representation of reality will be fake and based on "lies", which at the end will make the predictions not justified.

3.2.4 Dependencies with other requirements

Directly depending on the component which fetches articles and blogs posts.

3.3 Machine learning and model training

3.3.1 Description

Even though this functionality is what derives the final answer, it's listed in the third place, because it directly depends on other functions. This part of software applies machine learning techniques to train the model used by Skanner to predict stocks price. It uses constructed dataset from historical stock data and results of sentiment analysis to compute possible outcomes and chooses the most probable one.

3.3.2 Criticality

This requirement is also classified as essential for the overall system to work. It's the component which will take the most time to implement and it will take days of testing and evaluation to optimise it and tweak to the point where prediction will be reliable.

3.3.3 Technical Issues

The model training will become tricky, because Skanner might have to iterate through different sets of configurations and algorithms until it finds the most successful combination. The process will become even more complex if additional attributes are going to be introduced to the equation.

3.3.4 Dependencies with other requirements

This functionality doesn't have direct dependencies, but it itself is dependent on other requirements listed above.

3.4 Articles and blogs interface

3.4.1 Description

This interface is responsible for pulling articles and related readings to the query specified by a user. Its purpose is to feed the system with data which needs to be analyzed and processes particular way so it can be used later on by other components.

3.4.2 Criticality

As everything before, this functionality is also classified as essential because it provides resources which are used to compute the final answer.

3.4.3 Technical Issues

There shouldn't be any issues with this component. There are APIs out there provided by services which handle and distributes news articles and blog posts. Implementing such a client won't give any problems.

3.4.4 Dependencies with other requirements

The only dependency for this requirement is the internet access and of course functional external service which is used as source.

3.5 Finance and stock data interface

3.5.1 Description

Very similar to the previous one, but this interface is responsible for retrieving historical data about the stock from the query.

3.5.2 Criticality

Also classified as crucial, can't predict stock market behavior without any stock market data.

3.5.3 Technical Issues

There are multiple clients available to connect to many financial services which are capable of providing the required data about stocks. There should be no technical issues implementing this in Skanner.

3.5.4 Dependencies with other requirements

The only dependency for this requirement is the internet access and of course functional external service which is used as source.

3.6 User interface

3.6.1 Description

It is also crucial for the program to be intuitive, easily adopted and not repelling to a user. A great solution to this problem can be found in Shneiderman's "Eight Golden Rules of Interface Design", which should be strictly applied while designing the user interface [4].

3.6.2 Criticality

This requirement is important, but only to some extent if taking the above functionality into consideration.

3.6.3 Technical Issues

Since Skanner is a standalone desktop application, the user interface can be tricky to develop programmatically using Java Swing library or something similar. There is definitely a learning curve here which makes application of the Eight Golden Rules of Interface Design somewhat uncomfortable.

3.6.4 Dependencies with other requirements

None.

3.7 Save and Load query meta data

3.7.1 Description

The purpose of this functionality is to give a user an ability to save their results to a file at any point. The same file then can be used by the user to load query results back to Skanner so they can continue their unfinished work.

3.7.2 Criticality

This requirement is more for convenience than anything else. It's a feature that would be appreciated, but won't affect the core functionality of the software if it's not implemented.

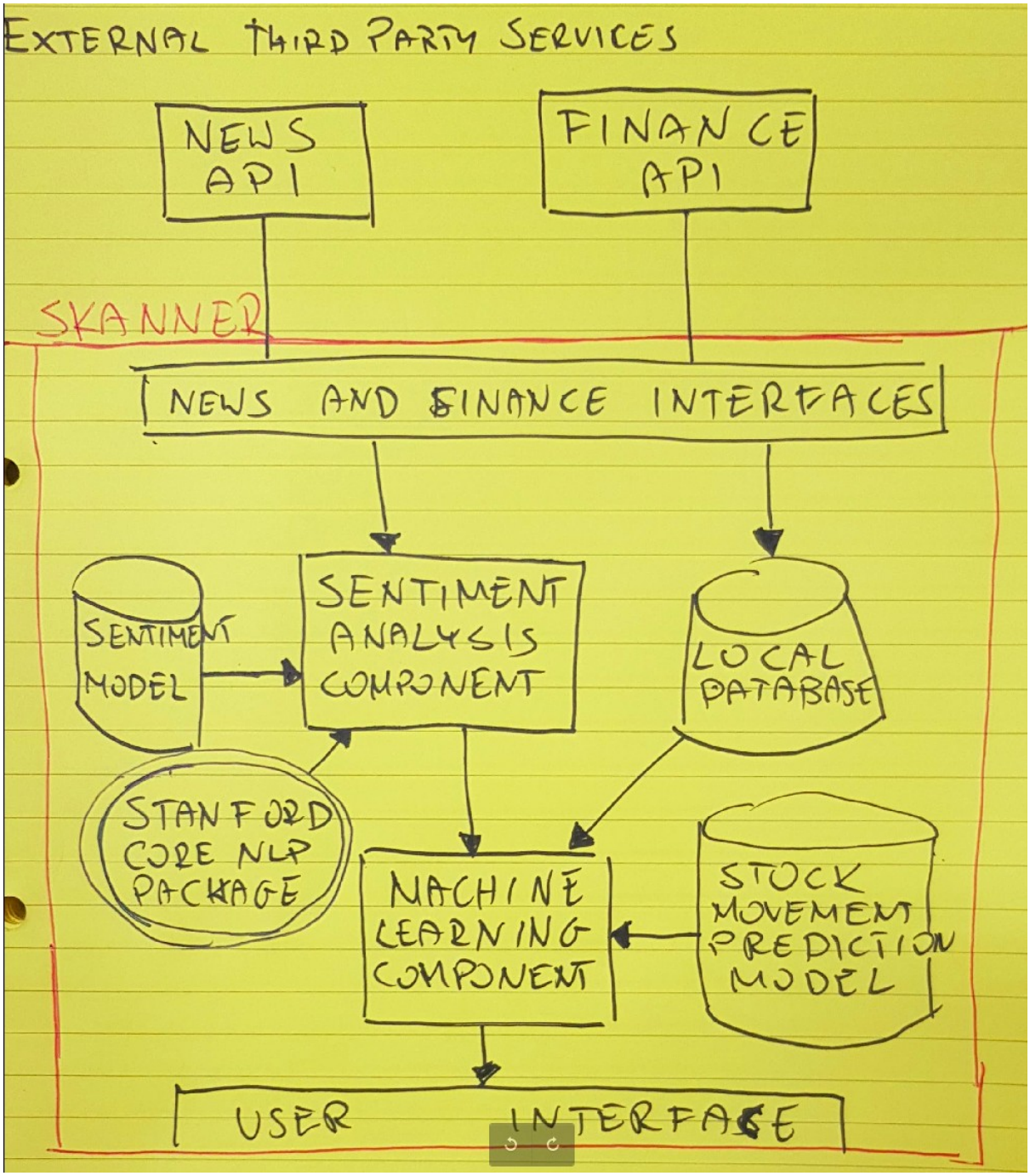
3.7.3 Technical Issues

The issue which can arise with this requirement is in the process of summarizing the whole query result. Rather than saving all data into a file, which wouldn't be the best practice, the meta data needs to be used instead. This is where it becomes tricky, all articles, sentiment analysis, historical data and the model itself have to be condensed to relevant information and meta data generated, which only then is saved into a file.

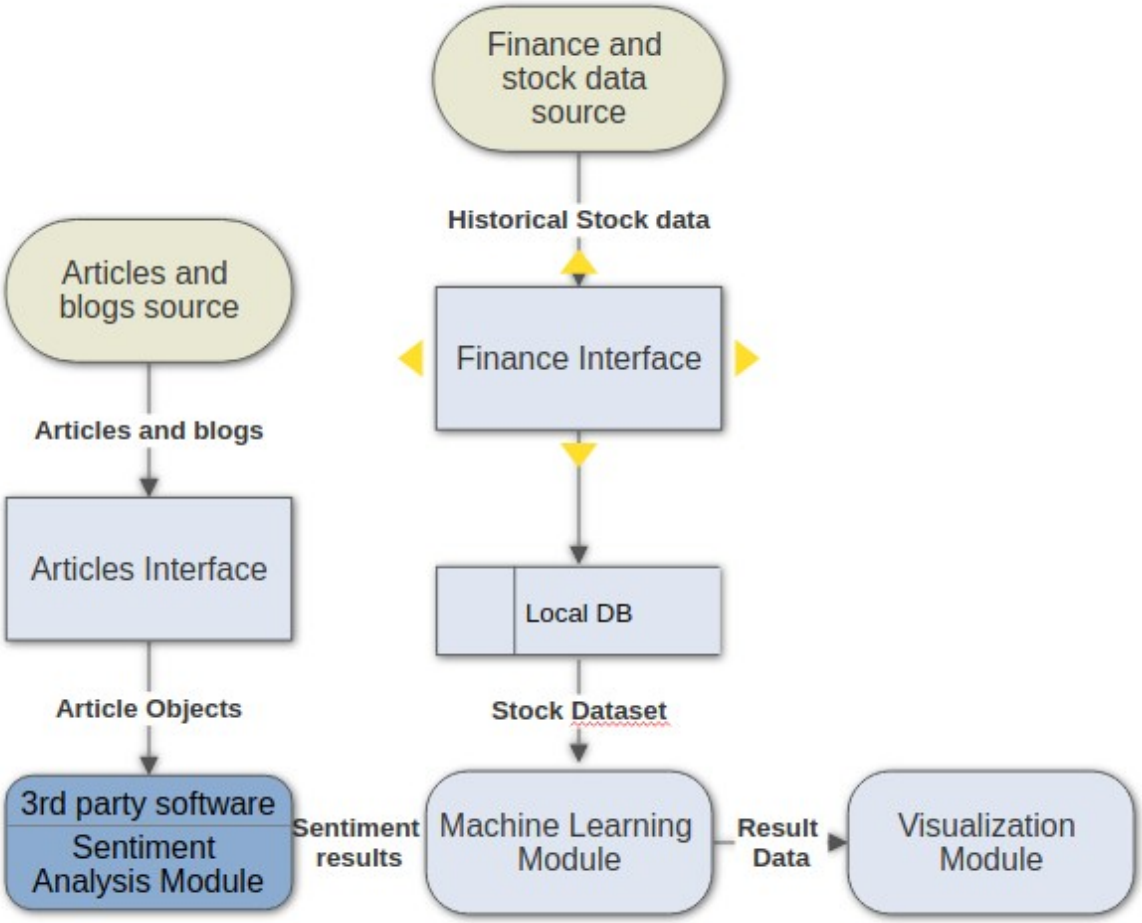
3.7.4 Dependencies with other requirements

None.

4. System Architecture

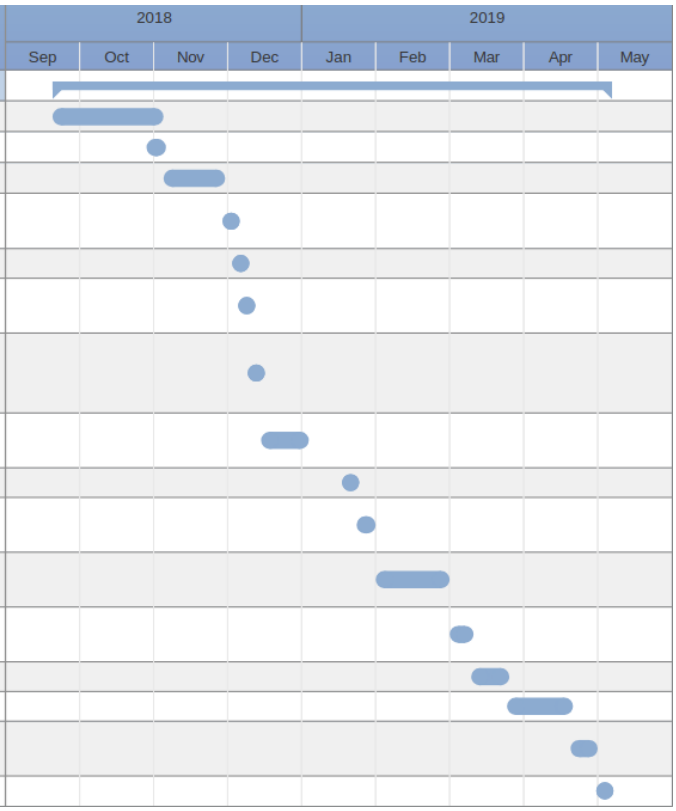


5. High-Level Design



6. Preliminary Schedule

	Task		Start	End	Duration	%
	Skanner - Fourth Year Project		9/20/18	5/6/19	157	
1	Identify the project idea		9/20/18	11/4/18	31	
2	Project proposal		10/29/18	11/5/18	6	
3	Functional specification		11/5/18	11/29/18	18	
4	Researching optimal sentiment analysis software and model		11/29/18	12/5/18	5	
5	Implement news feed interface		12/5/18	12/7/18	3	
6	Implement finance / stock interface		12/7/18	12/10/18	2	
7	Identify methods of scoring the articles based on their sentiment analysis		12/10/18	12/15/18	5	
8	Implement sentiment analysis component		12/15/18	1/3/19	12	
9	Identify data classification		1/18/19	1/24/19	4	
10	Construct model for machine learning		1/24/19	1/31/19	6	
11	Implement machine learning component		2/1/19	2/28/19	19	
12	Implementing save and load modules		3/1/19	3/10/19	6	
13	Integrating all modules		3/10/19	3/25/19	11	
14	Testing, validation and bug fixes		3/25/19	4/20/19	20	
15	Writing technical and user manuals		4/20/19	4/30/19	7	
16	Create a video walkthrough		5/1/19	5/6/19	4	



7. Appendices

1. **Don't try to predict the stock market. Focus on these three factors while investing instead,** ET CONTRIBUTORS, 2018-05-07 11:48 AM IST
   - https://economictimes.indiatimes.com/wealth/invest/dont-try-to-predict-the-stock-market-focus-on-these-three-factors-while-investing-instead/articleshow/64038318.cms
2. **Four ways to predict market performance,** Tristan Yates, 2018-08-04 10:23 AM EDT
   - https://www.investopedia.com/articles/07/mean_reversion_martingale.asp
3. **Stanford CoreNLP – Natural Language Processing software**
   - https://stanfordnlp.github.io/CoreNLP/
4. **Shneiderman's "Eight Golden Rules of Interface Design"**
   - https://www.designprinciplesftw.com/collections/shneidermans-eight-golden-rules-of-interface-design