CA326 - Finance App- Functional Specification

Brian Quilty(18373856), Sylwia Bielinska (19374573)
Date finished ; 10/12/2021

Table Of Contents

1.Introduction

1.1 - Overview

Our project is based on a finance app that will keep track of a database of the five-hundred most popular stock options available. The user can then select a stock which will be the stock that the system will keep track of. The phone will connect to the cloud (e.g. an aws account) which will scrape information about the price of the stock and the news surrounding the stock in the background and store it into a database (we will use MySQL for this). The app will send notifications if the price fluctuates too much based on a set of rules from existing data or if there is any relevant news surrounding the stock.

The user can also view all the stocks from the database and view vital information about each stock, such as the price, price over a certain period, a rating depending on hard coded rules to determine if the stock is doing good or not, and also any relevant news articles. There will also be a news section for any relevant news articles about the selected stocks. This news section will give a brief synopsis about the news for the stock, and also lead the user to the website for the stock.

The user will receive notifications if there are any applicable updates surrounding the stock. So, if the app determines that there is a fluctuation in price that meets the hard coded rules, then it will send a push notification to the user. The user can then swipe on the notification or open it, and the app will open and lead them to the relevant update. This information will also be displayed on the app, so if the user is using the app, then a relevant notification will appear on the app, or if the user opens the app, then the notification will also appear.

1.2 Business Context

The product will be used in the financial sector for organizations involved in the stock market.

2. General Description

2.1 Product / System Functions

User functionality

The main functions of the app will be listed below. The functionality of the app might shift over the course of doing the project, but for now the following list is a good representation of the expected functionality of the app.

- User will download the app and create an account on the android phone that will be specifically linked to the account.
- Login with their credentials used when creating the account. App connects to the cloud and the database of stocks.

- View the database of all the available stocks.
- View stock information (new article, price etc.) for individual stock.
- Select a stock to keep updated on from the list of stocks.
- App sends notifications to the phone when there is a new update (fluctuating price, news article).
- List of Rising stocks, based on database information from hard coded rules and or news articles.
- List of heavily fluctuating stocks.
- Links to relevant stock news
- User profile
- Remove stock from list of selected stocks.
- Logging out.

Cloud and database

2.2 User characteristics and Objectives

This application is mostly suited for people over the age of 18 and in some cases over the age of 21. This is because most brokerages require the person to be over 18 and sometimes over the age of 21. The user community that will predominantly be targeted will be people involved in the financial sector and more specifically people involved in keeping up to date with any stocks that they have bought before.

The users should have some sort of experience with mobile apps before, and know how to download an app, as this is the way in which they will access this app. As a result of this being a mobile application it saves the user time when checking the prices for a stock. Notifications and a seamless interface where simple clicks under their selected stocks can give detailed information about a stock rather than having to seam through various information online. Our app should accommodate people who have little to no experience with using an app before.

2.3 Operational Scenarios

2.3.1 User Creates account

When the user first opens the app, and they have not yet created an account, they will be met with a UI for creating an account. The UI will consist of fields for entering an email address, a password and a user-name.

2.3.2 User logs in

Upon entering an app where an account has been created, the user will be met with a log-in UI. This UI has one field for either the user's email address, or the user's username. Alongside this the user has to enter their password if they want to login to their account. If the details entered are correct, then the user can enter the app. However, if this is not the case then the user will be met with a warning that the details entered were not right, thereafter the user can re-enter their details.

2.3.3 View/Select from a database of stocks.

Once the user enters the app from there, they are presented with a list of the stocks from the database. This can also be accessed by a UI/UX system at the bottom of the app. If they click on a stock, they will be relayed with information regarding its price history, news articles and an estimated rating for the stock. From this screen they can then go on to select the stock that they want to be updated on.

2.3.4 View Relevant News articles

The user can select a news article button on the bottom of the UI for the app. This button will move the user to an interface where any relevant new articles for the database of stocks will be displayed. Once they select it a brief synopsis will be displayed and a link to the relevant news article will guide the user to that news article.

2.3.5 View rising stocks

Again, a button for rising stocks will be displayed at the bottom of the UI for the app. This button will lead the user to an interface where the system will determine the stocks with the most potential based on a rating. This rating can be sorted by different parameters, (e.g., high to low).

2.3.6 User sent notifications

If the system gets an indication that stock that the user has selected has fluctuated in price then the User will be sent a notification directly to their phone. Similarly, if a stock that the user has selected has a news article written about it then a notification will be sent directly to the user's phone. When a user receives a notification, they can click on it and the notification will directly link to the app and the information that the notification sent.

2.3.7 Log out/ Delete Account

The user can delete their account or log out of their account if they wish to. If they delete their account any relevant information associated with that account will also be deleted. If their account is logged out, the user will be brought back to the login screen.

2.4 Constraints

2.4.1-time constraints

We are in a limited time-frame as this project has to be completed by the 4th of march. That gives us around three months for this project to be fully completed and functional. Due to this it is vital to ensure that the fundamental parts of the project are implemented first, while other less notable features can be implemented afterwards.

2.4.2 Operating System Constraint

The mobile operating system is a significant constraint that will be evident throughout making the app. The app will only be developed on the android operating system and not on the apple operating system, meaning some users might not be able to access the app as a result.

2.4.3 Database memory constraints

We will be using MySQL. There is a limit on the amount of memory that you can store, as a consequence of this there will be a limitation on how the number of users allowed.

3. Functional Requirements

3.1 User creates account

Description;

This is the first functionality of the app. It will be the first thing that the user is presented with when they first install the app. There will be three fields which the user must fulfill in order to create an account and proceed to the app. These three fields consist of an email address, a username and a password. The email address and username must be unique to the user, so the data will be checked in the database in the backend to make sure that this requirement succeeds. If this test does not pass, then a warning trigger will be evoked and the user will be prompted to re-enter new details. The password also has to meet certain conditions, such as being of certain length as well as having a number. This is to help add security to the users' passwords. Their email address will have to be verifiable, therefore a confirmation link will be sent to the selected user's email address, from there a link will enable the account and link it to the database with their information.

Criticality;

This function is extremely critical, as without it the user will not be able to enter the app and therefore not be able to attain any benefits from any other functions inside the app. Security is vitally important so having an account for each individual user ensures that any data is linked to their account. Email-addresses and the enforcement of strong passwords help tie the concept of security and engrain it into our app.

Technical Issues;

Checking against already existing user details in the database could create a problem as there could be a vast amount of user details to check.

Dependencies;

App will not be able to be accessed without this function being successful.

3.2 User logs into their account

This process occurs when a user has already created an account and wants to log into their existing account and link up their data. They will either use their account username or their email address in the first field. The next field will contain their password, which will be tied to either their email/or account. Both of these fields will be checked to values in the database. If the values entered succeed, then the user will enter the app. Otherwise they will be prompted to re-enter their details.

Criticality;

Similarly, to creating an account this functionality is vital. Without logging in the user cannot use the app.

Technical Issues;

Checking the details against the database could cause an issue to arise, as there will be a variety of users inside the database.

Dependencies;

App will not be accessible without the user executing this function.

3.3 User Views database of stocks

Description;

This is the first UI that the user will be presented with after they log in to their account. This interface will consist of all of the stocks within the database, information such as the stock name, the stock price and percentage increase/decrease will be displayed one after the other. Options to sort a stock by price and name will also be available for the user to help them navigate through stocks. This UI will be scrollable, helping the user shift through all the individual stocks.

Criticality;

This is one of the most important parts of the app. The app is fundamentally about stocks, so therefore this feature is critical to the app. This feature must update when a stock's price updates, thus it is continuously updating over time.

Technical issues;

The database will only be able to store a finite amount of stock information as MySQL sets a limit.

Dependencies;

Users must be logged in to access this feature.

3.4 User searches for an individual stock

Description;

When a user is in the UI for the database of stocks, there will be a search field at the top of this UI, where a user can enter the name for a stock and then use the search button to search for that stock. The backend of the app will then deal with this request. The search will shift through each stock in the database until it finds a matching request. Any stock name that matches the string input from the search field will then be displayed in the results section. A minimum of three letters will have to be entered to increase the optimality of this feature. Any search that does not match a stock’s name, will be returned with a message that there are no matching results.

Criticality;

This is not of utmost importance for the app, but it is a very useful feature to help sift through a vast number of stocks. It increases the time efficiency for a user when looking for a specific stock, as without it they would have to spend a large amount of time scrolling through stocks.

Technical Issues;

Checking the input against the database might be strenuous on the app.

Dependencies;

User clicks the search field at the top of the stock database.

3.5 User clicks on a stock

Description;

When a user clicks on an individual stock inside the database, they will be met with information regarding the stock. Stock information like the name of the stock, the price of the stock, and price over the course over certain periods will be displayed for the user to look through. Alongside this a rating of the app will be displayed, and any relevant news articles will also be shown.

Criticality;

This function is crucial as the user needs this information to keep up to date with the stocks they want to view. Providing this information is the information that the app wants to show to its user database.

Dependencies;

User clicks a stock that is in the stock database.

3.6 User Selects a stock

Description;

User selects a stock where they will be constantly updated on. Once a user clicks on a stock, there will be an option to select the stock. When a user selects a stock, they will be constantly updated with information regarding the stock, e.g., if the price of the stock fluctuates, or if there are any news articles surrounding the stock.

Criticality;

This function is highly important as the whole point of the app is to keep the user up to date in real time about the state of their stock, so they can make decisions swiftly.

Technical Issues;

Users might only be able to select a certain number of stocks, so the app is not overloaded with tasks.

Dependencies;

User clicks on the select a stock UI.

3.7 User receives notification (stock update)

Description;

The user will receive a notification from their phone, if there is a price fluctuation in one of their selected stocks. Alongside this if there is a brand-new news article on one of their stocks then a notification will also be sent to the user's phone. A notification will have a pristine UI (including the message for the notification) that will show up on the phone's home screen. Once the user opens the notification, the app will open as well, leading the app to what the notification was trying to convey to them. For example, if there is a new news article on the user's selected stock, then the app will open and show the news article.

Criticality;

This function is especially important to let the user know as fast as possible if there is an update to their stock so they can make any necessary adjustments(buy/sell) to their stock. The whole purpose of the app is to make sure that the user gets real-time updates to their stock, so adding this feature is paramount to implementing this into our app.

Dependencies;

User clicks on the notification sent to their phone.

Technical Issues;

Make sure the user is not constantly overflowed with notifications.

3.8 User checks news articles for stocks.

Description;

There will be a UI at the bottom of the app which will lead the user to all the news articles for the stocks. Once the user clicks the UI, they will be presented with any relevant news article for the stocks. This will be over a certain time period(e.g. a day). This time period will be tried and tested to see what yields the best results.

Criticality;

This function is not extremely critical to the app, but it is a good insight for the users to check the overall state of stocks, and to keep up to date with stock news. If there is enough time remaining then this function will be implemented.

Technical Issues;

Limit on the number of news articles per day, so the app is not overflowed.

Dependencies;

User clicks on a symbol in the UI at the bottom of the app.

3.9 User clicks on news articles.

Description;

This function coincides with the previous function. So therefore, if the previous function is fully implemented then it is safe to assume that this function will also be fully implemented as well. Once a user finds a news article that it wants to read, then the app will give a brief synopsis, and a link leading the user to the news article.

Criticality;

This function is of the same criticality of the previous function. So, if the previous function is implemented then this function will also be implemented.

Dependencies;

User clicks on the check news article button.

3.10 User checks updates for selected stocks

Description;

A UI is necessary to make sure that the user can keep up to date with their selected stocks. To apply this to our app, we will create an individual UI that can be accessed at the bottom of the app through a symbol in a UI. This will lead the user to a UI where information about the updates for their selected stocks will be displayed.

Criticality;

This function is critical to the app, even though notifications about updates will also be sent to the user's phone. A concise UI showing all the updates to the stocks will be a good for the user to keep up to date with their stocks

Technical Issues;

Rules in the program to make sure that there will only be a certain amount of updates a day. Example, will only send updates if price fluctuates within a limit.

Dependencies;

Users must have selected stocks before.

3.11 User deselects a stock

Description;

In this function the user can deselect a stock that they have already selected and want to be updated on. The result of this is that the user will no longer be sent updates about the state of the stock (the price and the news for the stock). The user can do this when they click on that specific stock, they will be presented with a button to deselect the stock.

Criticality;

This is highly critical as if the user cannot deselect a stock, they might end up with a vast number of stocks where they will be updated on. This can overload the app, and if this happens then a user cannot select any new stocks, therefore destroying the app. As a result of this the user will have to uninstall and reinstall their app.

Technical Issues;

Rules in the program to make sure that there will only be a certain amount of updates a day. Example, will only send updates if price fluctuates within a limit.

Dependencies;

User must have first selected a stock.

3.12 User logs out

Description;

This allows the user to log out of their created account if they wish to. If they decide that they want to log out, they can log back in at any time if they wish to and their account will remain unaffected.

Criticality;

This is important as if a user loses their device, or if they want to use a different device, they can still afford of the functionality of the app given their user information.

Technical Issues;

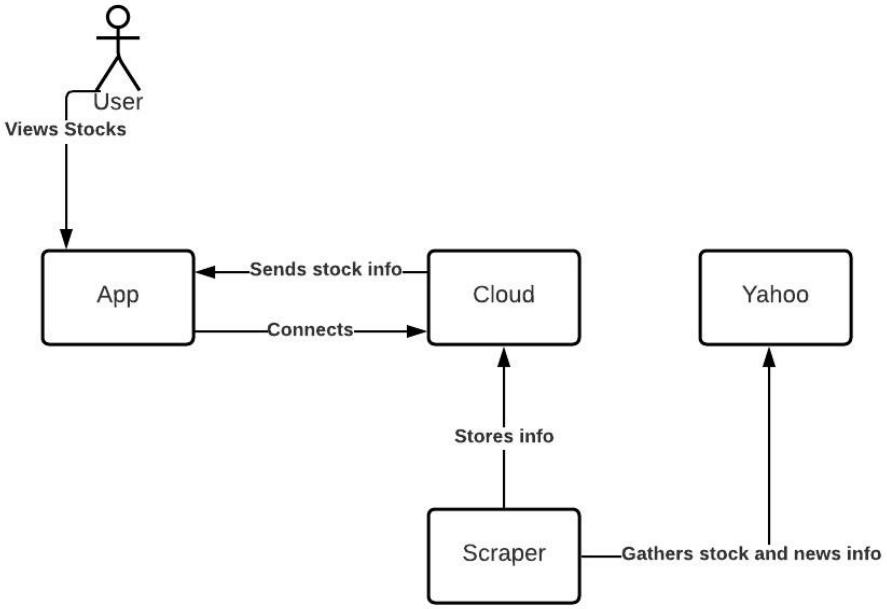
The app must make sure that it does not function (send updates etc.) if the user has logged out.

Dependencies;

Users must be logged in first.

4. System Architecture

4.1 System Architecture Diagram

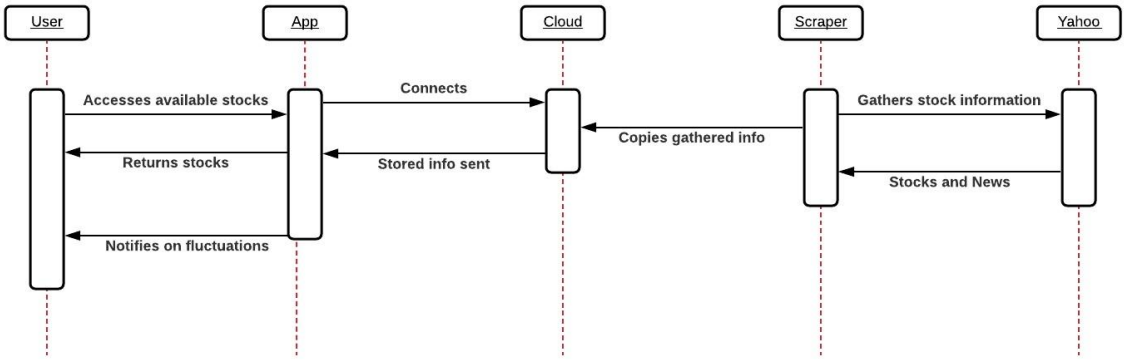


4.2 Description

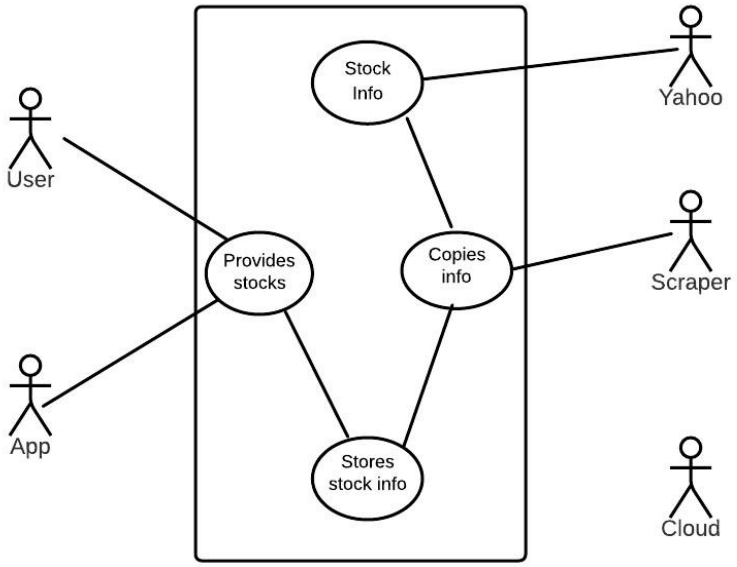
The current system architecture diagram shows the interaction between each system and tool used. The user, through using the app, will be able to find all the different stocks that are available and be able to choose which stocks they want to follow. This will then allow for the app to send the user notifications if any large fluctuations occur with a stock that is being followed by the user. In order to get information on different stocks, the app connects to the cloud. The cloud stores stock information that has been gathered from Yahoo using a scraper. The scraper copies the stock information from Yahoo to the cloud database from which the app will have access to the stocks.

5. High-Level Design

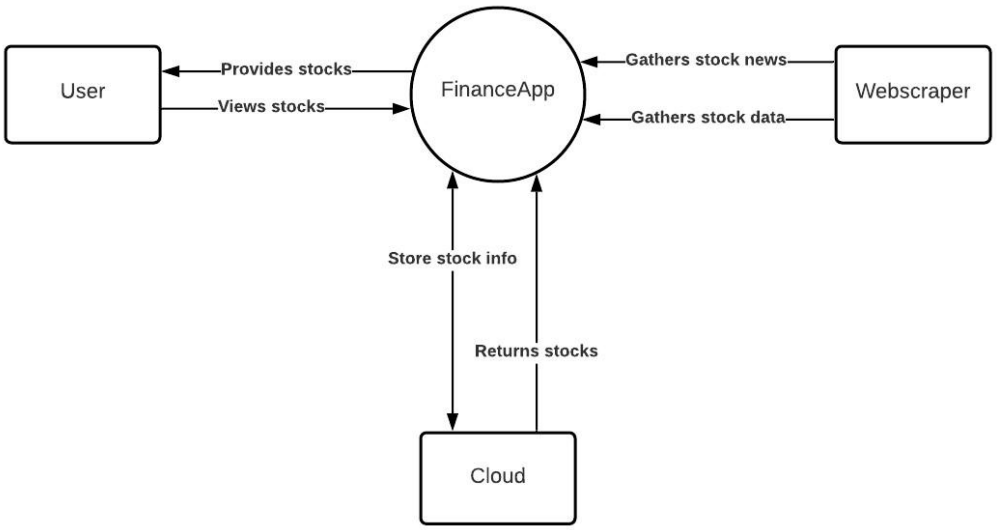
5.1 Sequence Diagram



5.2 Use Case Diagram



5.3 Context Diagram

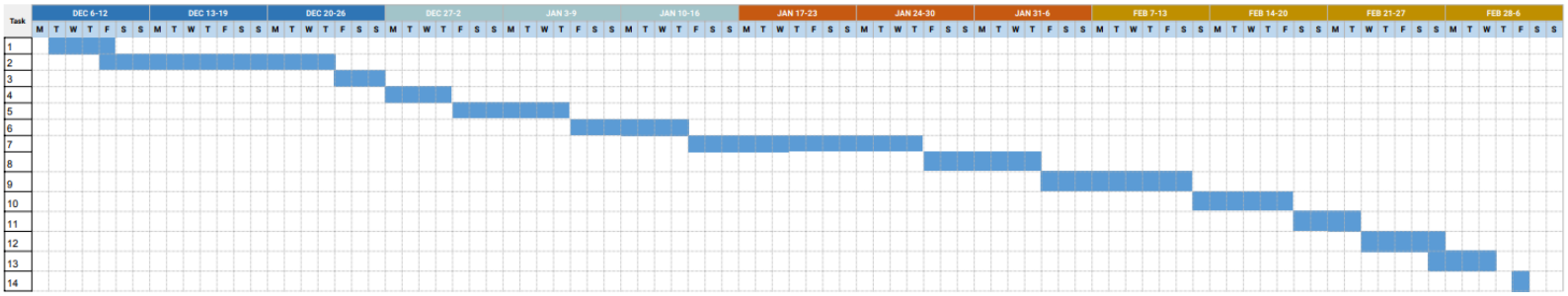


6. Preliminary Schedule

6.1 Schedule

Task	Duration	Start	Finish
1. Functional Specification	4 days	7/12/21	10/12/21
2. Research Period	14 days	10/12/21	24/12/21
3. Christmas Break	3 days	24/12/21	27/12/21
4. UI Prototype Design	4 days	27/12/21	30/12/21
5. Backend Code Skeleton	7 days	31/12/21	6/01/22
6. UI Login Implementation	7 days	7/01/22	13/01/22
7. Backend Stock Implementation	14 days	14/01/22	27/01/22
8. UI Stocks Library Implementation	7 days	28/01/22	3/02/22
9. Backend Final implementation	9 days	4/02/22	12/02/22
10. UI Final Implementation	6 days	13/02/22	18/02/22
11. Testing and Debugging	4 days	19/02/22	22/02/22
12. Video Walkthrough	5 days	23/02/22	27/02/22
13. Documentation	4 days	27/02/22	2/03/22
14. Final Submission		4/03/22	4/03/22

6.2 Gantt Chart



7. Appendices

- https://www.mysql.com
- https://aws.amazon.com

- https://gitlab.computing.dcu.ie/quiltyb2/2022-ca326-quiltyb2-bielins2 (view updates)
- https://www.codecademy.com (learn html, css, javascript etc.)