A PERSONAL MONEY MANAGEMENT APPLICATION WITH TAGS AND TWITTER INTEGRATION

By

Cheuk Him Calvin Lo

Undergraduate Honours Thesis

Faculty of Science (Computer Science)

University of Ontario Institute of Technology

Supervisor: Dr. Ken Pu

Abstract

A Personal Money Management Application Using Tags with Twitter Integration

Cheuk Him Calvin Lo

Undergraduate Honours Thesis

Faculty of Science (Computer Science)

University of Ontario Institute of Technology

2017

The aim of this thesis is to develop the idea of using tags in the area of personal money tracking, and to build an android application that demonstrated the features, visualizations along with twitter integration. This thesis first talk about the procedure of interface design the data structures. Then, it examines the implementation of the android application. The technology and libraries used in this thesis will be discussed. In conclusion, the thesis used the concept of tags to build a mobile application, with the aim of improve the user experience of managing spending. With Twitter integration, this thesis hopes to provide a social media platform where user can share and follow other users. Further modification to the application are opened to enhance the performance.

Content

Chapter 1

Introduction

Money management is a good habit for every individual. With money management, people will have a better control of their money. However, keep track of your spending can be boring and tedious, but posting a tweet to twitter is interesting. According to Internet Live Stats, there are around 6,000 tweets are posted on Twitter on average per second. It means 350,000 tweets per minute, 500 million tweets per day and around 200 billion tweets per year. [1] The statistics revealed the significant use of twitter. Compare to structured data entry, a single message is more convenient and easy to use. Mobile money management applications have been widely used in today's society. However, nearly all of those applications' approach are using a traditional form to input the data. On the contrary, the usage of tags become more popular nowadays. Hence, this thesis creates a personal money management application using the concept of "Tag". Furthermore, people love share their life and follow other people story. There is no doubt that social networking has a big influence in today's society. From elementary school students to retired people, social media is in their life. This thesis will discuss and develop the connection between money management and social media.

1.1 Problem description

Using a single message will enhance the user experience and make money management become more enjoyable. How to use a message to keep track of your spending? This thesis will discuss about the move from structured data entry to messaging data entry. To make a message to have

the ability to store data, the application applies the concept of tagging. The general idea of tagging is using a label to connect to some information. In this program, the label will use to separate the message into a list of tags while each of the tag is some information related to the transactions. User will be able to select a tag through a simple and user-friendly method and enter the information for that tag. For the social media aspect, a connection between the app itself and online social network provider is needed. With this integration, user can easily share their spending story and follow other user.

1.2 Overview

Transaction is the main data used in this thesis. Every single message user inputted will parse to a transaction data and store in a local database. Tag is another important data this application required. There are three types of tag used - General, location, category. Each transaction will contained a list of tags that entered with that transaction. Since user can follow other user, a user database is constructed where store the information of the local user and all following user. We will take about the structure of different data later in this paper. The application builds on Android platform. Android Studio used as the main development environment. It provided very useful libraries that used in this application. The application used different method to illustrate the data - a listview that list all corresponding transaction, a tag cloud that provide a visualization of the tag used by the user. With MPAndroidChart [2], the program can display the data with Pie Chart and Line Chart. It offers a different view of the data.

This application allows user to tweet their transaction to Twitter and follow other user's spending from Twitter. Fabric is a library that connect twitter and android. With this library, user can connect their personal account to the application and have the ability to post and download tweets from Twitter. All the implementations and technologies used in this application will be

discuss more in the later part.

Chapter 2

Background

The first general idea of this thesis to create an application that is useful and bring convenience to everyone's daily life. Everyone's life involved money and keep track of your spending should be interesting. However, using traditional form stopped many people to keep a record of their expend especially teenagers. In the culture of teens, money management is an old school and boring things. Therefore, a new way of money management will be a great proposal to explore. To move away from structured data entry, a single message with multiple tags is a solution. Moreover, social media integration can increase the interest of money tracking. With the ability of sharing and following, the experience of money management will be completely new.

2.1 Social Media Culture

The use of social media had increased dramatically in recent year. There are more and more different types of social media appear in the market. Social Media can divide into three areas - communication, socialization and change. [4] Technology gives communication a new definition. Online communication can break the barriers of space and time, increase the speed, minimize the uncomforted with traditional communication. [5] Although online communication can affect our daily life in a bad way, make good use of it can definitely improve the quality of social life. You can add friends in facebook to see their post. In Twitter, you can follow other user to see what they are doing. Instagram allows you to share your life with a photo. It helps

you and your friends get connected or even get the world connected. With social media, you can see what is the world happening right now and give you the most updated information. When there are somethings happening in the other half the world, the newest details you can see is always from the news publisher's social media account. It can even predict the future of some events. [3] "Socialization, the process whereby an individual learns to adjust to a group (or society) and behave in a manner approved by the group (or society)." [6] Social media have the power to help individuals building better attitudes and behaviours that accepted by the society. [5] Imagine when you are following other people's spending, you can see other people personality on their transactions. We cannot avoid the change made by social media no matter it is good or bad. It depends on how people use social media. There are always benefit from using social media correctly.

2.1.1 Tag culture

#blacklivesmatter is a hashtag appear in Twitter that raise a lot of awareness from the public. From a hashtag on Twitter to a chapter-based national organization, you can see the power of a hashtag. Hashtag was the "Children's word of the year" declared by the Oxford University Press. [8] You can see hashtag anywhere no matter in online world or real world. So, what is a hashtag? Hashtag contains a hash(#) label and a simple text. Hashtagging has become more and more popular and a cultural phenomenon that grow with the growth of social media. [9] Tagging also bring convenience to our life. You can categorize your data with a tag and search informations by a tag. Tag is a simple idea that help people group and classify things. Using tag in a single message allows the system to parse the data automatically and store it in databases. Tag can also visualize as a tag cloud, which provide a visual representation of your data.

2.1.2 Following

Following celebrities is common in online world. You do not need to physically follow a people but by clicking a button, you can see the entire story posted by celebrities. The following feature also create more online celebrities like video blogger and travel blogger. People love to know what happening in the world right now in a simple way. Following different media or people is the answer. It can give you updates and notifications of what you are interested. With a single screen, you can see all the things you followed.

2.2 Messaging vs. Structured data entry

There are two types of data input - messaging and structured data entry. Both types have their own advantages. However, messaging data entry provides a more easy and fun way to input data especially in the area of social media. It is hard to replace structured data entry completely. For example, some legal documents still required a tridiagonal form as the method of data input. Messaging already used widely in the online culture and can be used in more area.

2.2.1 Single message

The idea of using a single message is the user only need to enter a message and the system will automatically parse the data. It is a type of free text data entry. To make it works, it will apply the concept of tag discussed in 2.1.1. A single message can have multiple tags. With different types of tag, the system can categorize the message based on the type. Social media use a single message to input the data. You type your post in a sentence from one textbox. You don't want to input your thought in a form layout that ask you your mood, what you're thinking, the location and the people with you. Doing it in a single message is more fast and convenient. In addition, using mobile to post message or store data is the trend recently. However, the limitation of a

mobile phone limit the way we type data. Furthermore, people enter data into a form usually required two hands and need to select the wanted category. Messaging makes the things easier. It allows single hand input even thumb only since user only need to enter the message itself without clicking different area of the screen. Single messaging with tags and background parsing is the way to input data today and the foreseen future.

2.2.2 Traditional spreadsheet

Traditional spreadsheet is a structured data entry using a form layout. This type of data entry commonly use in formal setting. For example, government documents and job applications. It provides a clear and organized form for user to input data. The form provided the category that can store it the corresponding column in a database easily. Using a form to input data, you can expect the data types and required some category as mandatory. The entire category is predefined and display as a form. Structured data entry can constrain data entry behavior, improve data quality, readability. [10]

2.2.3 Comparison

Messaging provide a simple and convenient method for data input while structured data entry has an organized layout to enter information. Structured data entry will be more applicable when dealing with complex data. In money management, most of the approach is using a form to input all the information such as title, category and amount. It is a clear method to enter data but also a tedious way. The data types required in money management is little. Therefore, messaging with tags can be a good alternative method. Messaging has a flexible and unsorted structure while structured has a fixed and sorted structure. We can input multiple tag in a single message in a random order and information will display as a text. In addition, share a text is easier than share a form of data. You can easily share your

transaction to any social media without any reformatting. Compare to structured data entry, messaging gives a more flexible, unsorted, sharable, social network ready method to do data input.

2.3 Twitter platform

This application used Twitter as the social media platform. According to the twitter official website [11], the mission of Twitter is "To give everyone the power to create and share ideas and information instantly, without barriers." The number of user has increased remarkably, since it first started in 2006. The main feature of Twitter is posting a text that publish into the public. A text can be everything from having a coffee to official statement. Twitter also have the well-developed hashtag databases. The hashtag changed the world in different area - business, politics, breaking news, sport, celebrity, activism, arts and law. The culture of tagging can be easily seen in the Twitter world. When there are some hot issues around the world, a lot of related post and hashtag can be found online. With the huge amount of active user, the information will spread out in a notably quick time. Twitter use a simple idea to catch the eye of the different user. Unlike Facebook, Twitter only has one simple feature - tweeting with tags.

2.3.1 Tweets

Tweets is the name of the message sent in Twitter. The maximum words in a single tweet is 140 words. It makes the message become more clear and organized. The limited length brings the uniqueness of a tweet. There are different types of tweet. Regular tweet is a plain text. Image tweet is up to four photos with a message. Video tweet is simply a tweet with a maximum 30 seconds video. One of the biggest attribute of a tweet is hashtagging. Multiple tweet can added to a tweet with hash (#) label. Moreover, location and media link can also put to a tweet. User

can also share other people's tweets while this process called retweet. Tweets is generally a posting posted to twitter or the action of posting a post to Twitter.

2.4 Mobile UI

Since the application built on a mobile device, the interface should be user and touch friendly. "Material Design is a unified system that combines theory, resources, and tools for crafting digital experiences." [14] Material design has a design guideline that assist developer to build an application, which can examine the best user experience. It combines the classic principles and the modern technology. [15] The design is developed by Google and aimed to provide a unified user experience. [15] Most of the new android application followed the material design guideline to create the layout. Material color scheme is part of the material design that provide the predefined color set. Using the material color scheme can provide a simple and prepossessing design. Material icon is another package in the material design guideline. It provide multiple open source icons, which have a simple design and understable design. User experience is an important consideration when designing a mobile application. The material design gives a unique standard to all applications. With this standard, user can expect the same design pattern with different applications. As an android application, this program will follow the material design. Using the color scheme and icon package, it hope to increase the overall experience and amusement when the customer using this application.

Chapter 3

Design and User Experience

A success application require a high quality design. The design of this application desired to produce a user and mobile friendly interface, which can increase the general experience when using the application. A simple design lead to a simple usage. One of the key feature of this application is provide simple method to do finance management. Therefore, a simple design will assist to achieve this target. Compare to complex design, a simple design is more focused on the main function and easier to utilize. With the material design guideline discussed in 2.4, you can expect a unified design across different applications. In addition, a simple design is scientifically easier for human to process. [17] It require less usage to process the information from our eyes and brain. It can provide a directly way for the user to find what they need from the application. This program can divided into 4 main areas for the designing purpose - data, mobile interface, visualization and social media.

3.1 Data

A personal finance management application based on the user data. The user enter their transaction to the application and store in a database. All the data demanded a well-organized and developed design to increase the simplicity of usage and optimize the performance. A good design of the attributes in each data type can help the developer to process the information easily. Keep all the data in a clear and readable structure is the focus when designing the data in this application. Furthermore, the data should be as simple as possible. Any unnecessary or duplicated attributes should be removed or combined depended on the real usage. The three

major data types developed in this application are user, transaction, tag. All three types defined as a Java class.

3.1.1 User

When the user choose to view the local device data or any following user's data, a user need to be selected. With social media integration, user can follow other user from Twitter and download the data. It allows multiple users in the same device. "Local" user is the owner of the physically devices. All the inputted data will attached to local user type by default. "Following" user means a single Twitter account, that user had followed. There is a "All" user as well which included the local user and all following user. The user will determine what data will be displayed on the screen. The main attributes of user are the id, display name, username, profile image and a since timestamp. The id is simply the unique primary key. The display name is the screen name that user chosen or the Twitter display name if it is a following user. The username is a unique string from Twitter while local user will use "local" as the username. When followed a user from Twitter, it will download the profile image and resize it. Then, a copy of image will store in the device. When user download or refresh the data from Twitter, it will generate a current timestamp. The timestamp allows the user to download the data after a specific time. The following code is a sample code on selecting data from database based on the current selected user.

```
if (user.equals("*")) {
            selectQuery = "SELECT  * FROM " + TABLE_NAME;
} else {
            selectQuery = "SELECT  * FROM " + TABLE_NAME + " WHERE " + KEY_USER + "='" +
user + "'";
}
```

User can select a specific user to generate the visualization. The solution to store the selected across the application is using a shared preference. Shared Preference allows the application to

store a simple data that can access throughout the whole program. When user selected a specific user account, it will updated the shared preference. With this function, the system can know what data should be selected and displayed.

3.1.2 Transaction

Transaction is the message itself that user entered. After parsing (we will discuss it in 3.1.4), the single message will save as class with multiple attributes. Table 3.1 showed the summary of all attributes.

Title	**Type**	**Description**
ID	INTEGER	The primary key.
Message	TEXT	The original message user entered.
Timestamp	TEXT	The timestamp when user enter the message in "dd/MM/yyyy HH:mm:ss" format.
Year	INTEGER	Year extracted from the timestamp.
Month	INTEGER	Month extracted from the timestamp.
Day	INTEGER	Day extracted from the timestamp.
Amount	REAL	The total amount of this message.
User	TEXT	The user who own this message.
General	TEXT	All general tags as a string.
Location	TEXT	All location tags as a string.
Category	TEXT	All category tags as a string.
Color	INTEGER	The color code user selected for this message.

Table 3.1

INTEGER is real number while REAL is decimal number. To avoid parsing the timestamp and extract the information every time the system query the database, Year, Month and Day stored in the database as an individual row when first added.

3.1.3 Tag

This application built on the concept of tagging. A tag is simply a single unicode follow by a text. A tag is the combination of a label and text. All tag labels used in this application are a unicode. The reason of using a unicode as the tag label is the uniqueness. Twitter used hashtag(#) and the at(@) label to do tagging in their system. As a separate app from Twitter, the system should not retrieve the data that is not created or required by the application. Using a unicode can be unique with other social media. There are 4 types of predefined tag that user can choose to use. The first one is the general tag. The usage of general tag is all-purpose. It can be everything that user want to tag with a transaction or any thoughts that user having while spending the money. Second, location tag used for storing the place where the transaction took place. Third, category tag used to classify the whole transaction. This tag allow the user to group their transaction. The last one is amount tag and it is a special type of tag. The message can have multiple tags included the amount tag but the amount tag will sum up when doing the parsing. A real life example will be when users want to write one transaction with more than one item, they can enter more than one amount and the system will do the calculation for them.

3.1.4 Parsing

Since the data entry method is a single message, the system required to parse the message into a structured data. There are two type of parsing - application and Twitter. The difference between application message and Twitter is Twitter's message have a timestamp and application hashtag. When the system download the tweets from Twitter, it will first search and download the tweets that have the ending of the application name. Other than downloading all data, this approach will save the storage space and make the data simpler. The first step of parsing is to create a empty transaction class. The parsing process will be done by Java regular expressions. It will

split the string based on the tag label. If it is an amount label, it will sum of the amount. All other tag will store directly into the database. After reading all the tags from the splitted list, it will add this transaction to the databases.

```
format = "(?=" + GENERAL_ICON + "|" + LOCATION_ICON + "|"+ "\\" + DOLLAR_ICON + "|" +
CATEGORY_ICON + ")";
String parsedTags[] =  message.split(format);
```

There are some steps need to finish before the main parsing process. After received the tweet from Twitter, system will remove the application hashtag and find the timestamp label. Then, the timestamp will added to the transaction class other than using the default setting which is apply the current timestamp. After this process, it will remove the timestamp from the message and able to start the parsing as described above.

3.2 Mobile interface

Mobile interface is the front end where the user interact. User would not able to see how the back end worked. The only thing they can play with is the interface, which means they will spend all the time with the interface. It proved the importance of a good mobile interface. Without a good mobile interface, user may find a hard time to explore the application and have a bad impression. Therefore, the mobile interface should be designed well to attract user and provide a easy to use environment.

3.2.1 Layout

A mobile device is mostly used in the touch mode where user use their finger to navigate the application. Therefore, a good layout need to maximize the benefits of touchscreen and swipe feature. Instead of creating an individual activity for each view, tab layout is chosen to be the

main layout of this mobile application. With tab layout, user can easily swipe between different views. It used fragment as the view for different tab. Fragment can be seen as the sub-class from the main activity. It can access and communicate with the parent activity. Most of the layout in this application used an action bar, which can provide the information of the current view. Title will set to the username when selected a specific a user and subtitle will set to the total amount of transaction or tags depend on the view. To select a specific month and year to display, a month and year picker created. Using a picker is more user friendly than typing the actual text and increase the speed. The action bar also implemented the search view function. Searching tag directly from the action bar means the user can search with one touch. Another main layout is the drawer. Drawer opened by swipe or touch the menu icon. The drawer work as the navigation menu that provide a layout to move between function. User can choose the user from the drawer and changed the current view to the selected user by a simple swipe and touch. The target of the layout design is to provide a simple but well functional mobile interface.

3.2.2 Tag Toolbar

As mentioned in 3.1.3, a tag started with a unicode. To enter a unicode with the actual code is definitely unrealistic and inconvenient. The solution is building a customized a toolbar to improve the keyboard functionality.

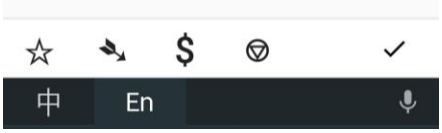

Figure 3.1 a screenshot of a tag toolbar.

Tag toolbar is a box of button with a tag label. The function of it is to let user directly use the toolbar to enter the tag. There is a problem need to deal with when using a button to input unicode. The first thing is the position problem. The system need to track the position of cursor

before editing the message with a button. In addition, it need to return the correct position after inputted the unicode. Tag toolbar will be an accessibility tool to input tag label.

3.3 Visualization

The transaction data need to be visualize in a clear and engaging way. An application is a two-way program. The design of the input process had already discussed. The output is the way user see the data. Designing the visualization is a major challenge of this thesis. It required multiple techniques to generate different views. In addition, an attractive illustration is difficult to define but is a key aspect to this application. In this thesis, we examined and developed four kind of visualization for the data. A listview to show the message, a tag cloud to illustrate the tag, a pie chart to display the portion of tag and a lie chart to picture the trend. For any visualization, user can choose a specific user account, month and year as the target.

3.3.1 Listview

This is the main and first view which user will see when they open the applications. Expanded list view used to represent the transaction data. An expanded list view contains two elements - parent holder and child holder. The parent holder is the collapsed view (Figure 3.2) while the child holer is the part show when the list item expanded (Figure 3.3).

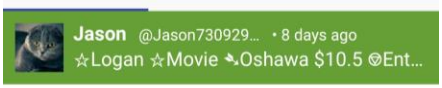

Figure 3.2 collapsed view

Figure 3.3 Expanded view

The parent holder will show as a social media structure, which has the profile image, display name, user name and the timestamps of the transaction. In addition, it will display the original message in a single line. When the user click the list item, it will expand and show the child holder. The expanded view will expand the message from single lines into multiple lines if applicable. It will display the corresponding tag as well. The reason of using an expanded list view is the flexibility. The height of each item can be auto adjusted by the displayed content. To give a user-friendly experience, the expanding and collapsing process is simple by a single click and have the capability to cooperate the main view. Compare to a general list view, the feature of expandable can applied to show more information. Compare to swipe layout, expandable list view change the view by a click action while swipe layout used swipe. Using swipe can be overridden by the tab layout since it already has a swipe feature that applied to the whole screen. In addition, swipe layout cannot will have a predefined size and it is fixed. Therefore, expandable list view is the best suitable visualization to display the transactions data.

3.3.2 Tag Cloud

Tag cloud is a visual depiction of user-generated tags attached to the content. [16] To show what user had typed in, tag cloud is the solution. Tag cloud display all tags from a particular user, month and year in a grid layout. It is easier to define the number of row and column with the staggered grid layout manager. It has gaps at edges between each box. When a user entered a message with a amount tag, all the other tags will attach to the amount as well. The size of each tag box determined by the amount of that tag. A bigger box and font size will create for a tag with higher amount. The formula of calculating the font size modified by the formula. `tag_font_size = ( (( tgs – sgc) * (lfs – sfs)) / (lc-sc) ) + sfs` where sc is the smallest amount, tgc is the total count of the tag, lc is the largest amount, sfs is the smallest

font size and lfs is the largest font size. [18] Tag Cloud is a simple design that allow user to navigate all the tags they used before.

Figure 3.5 Sample Tag Cloud

3.3.3 Pie Chart

A pie chart illustrate the portion of the tag with the material color scheme. It provides another visualization of the tag's amount. Compare to tag cloud in the previous section (3.3.2), a pie chart display the amount with a more straightforward design. The size of each portion of the tag depended on the amount. It also allow the user to select of tag they want to see. Figure 3.6 showed a sample pie chart with five tags. The limitation of pie chart is the fixed size since every tag need to be fit a circle. If there are a lot of tag, the tags with small amount will not be seen easily. The pie chart gives a general picture of the tag's usage.

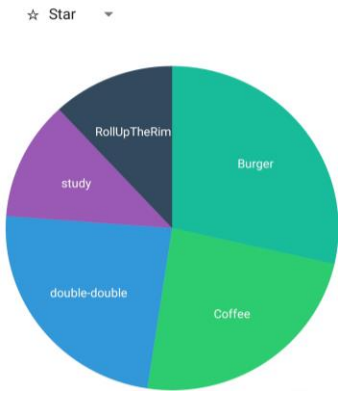

Figure 3.6 Sample Pie Chart

3.3.4 Line Chart

Line chart used to show the usage of a tag within a particular period. Trending is one of the key things user want to have. They may want to see the timeline of their spending within a tag. User can select the specific period and generalize the line chart. The design is to provide a easy layout that user can view their transaction based on time. Therefore, they can see the timeline from a whole year to a single month. With the zoom feature, user can even select a weekly view. Figure 3.7 showed a sample line chart of a whole year trend. The benefit of a line chart is you can see the overall usage of a specific tag based on the time. It aims to help the user to do a better management of their spending and give them a picture on what period in the year that they have a higher spending habit`.

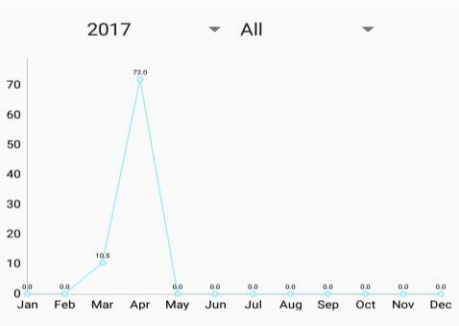

Figure 3.7 Sample Trend

3.4 Social Media

Twitter is the social media platform used in this project. To integrate with twitter, the program need to able to connect to the Twitter server for the purpose of uploading and downloading data. To design the social media integration, the focus is how to provide a simplest way to let the user access their twitter account. The integration based on two parts - Sharing and following. When user typed a message, what is the best approach to post to twitter? On the other hand, how to download data from Twitter server is the easiest way and require less actions?

3.4.1 Share

The function behind the sharing feature is uploading data to the Twitter server and publish it. The design of this is to allow user to choose when and what to share. The easier way is to post automatically right after user saved the message to the local databases. However, for the reason of flexibility, the application should allow the user to choose to auto upload or not. Therefore, user can check the features of auto upload to true or false based on their behaviour in the setting menu. If the user chose not to auto upload, they can manually click a twitter icon to upload the message to Twitter from the input screen. After clicked the twitter icon, a native Twitter import screen will be shown. A native twitter screen means the official Twitter android application or the web page if no Twitter installed. The message will directly copy to the native Twitter application with all the append information. In this stage, user can easily can the tweet button to publish the message. The visibility of the twitter icon depended on the the boolean variable of auto upload. It will provide a more clean design to the user who are using the auto upload features. The auto upload feature is simplest way to do sharing while the native Twitter application sharing method provide alternative way that gives more flexibility.

3.4.2 Follow

To follow a user from Twitter, the program will search from Twitter and download the relatively data. The application will provide an individual activity that allow user to search the twitter account. From this activity, user can easily search the twitter user in the action search view and a preview of the data will be displayed. The application will only download and display the tweets that have the application hashtag happened. All other tweets will be considered as irrelevant and ignored. The user now can decide to follow this user or not by clicking a button. After followed a twitter user, user can select the following twitter user from the navigation menu and get the

visualization. To refresh and update the data, the system will delete all the previous data and downloading the new one. This approach covered all situation but not efficient. However, if the application only download the data since the last refresh, the old data that had been deleted from Twitter will still show in the application. Swipe Refresh Layout gives a swipe up and refresh function to the user that is easy to use.

Chapter 4

Technology Stack and Implementation

In this chapter, we will discuss about the major technologies used in this thesis and the process of implementing different libraries. There are many open source application program interfaces (APIs) online and available to use right away. Those APIs helped the developer to build the program easier because to implement an API is faster than building the function from scratch. However, the huge number of APIs led to the fragmentation problem. There are different version of build tools of Android and the technology keep changing every day. There are new versions of different software coming out in a short time. Therefore, choosing the best and suitable APIs is the first thing developer consider. To make sure the APIs work and collaborate with the rest of the program correctly, we required to do multiple testing. Technologies used in this thesis will be tested and function correctly with some testing.

4.1 Android

This application used Android to demonstrate all the features and provide an interface for the user to explore the program. Android, a most common mobile operating system in the world

right now. Android application based on Java and provided different libraries to use. The flexibility of an Android application is high. It allows the developer to build an application that can used in most of the mobile devices.

4.1.1 Android Studio

Android studio is the official develop environment to build an Android application. The interface of Android studio is user friendly and updated to the new official Android version. It means all the newest features of Android can be developed in Android studio easily with the build in integration. The first thing to do when developing this application is building a draft user interface. With Android studio, layout can be done easily by drag and drop. In addition, developer can use XML codes to define the layout. The basic layout type is enough for most of the application while it allows further modification to fit the special requirement. All the layout and interface used in this application is from the original Android studio layout libraries. Android developed in Java. Therefore, all the programming of this thesis is using Java. Android studio has a good environment of building Java program with the rich autocomplete action and shortcuts. To start an android application in Android studio, a manifest file is needed. It will declare all the application main outlines and permission. For example, the permission of accessing the internet to connect to twitter can be defined in here. Manifest also provided the overall structure of the application with all the activities information. Each application also have a gradle section that declared all the system dependencies and libraries implementation. With the android studio file directory structure, all the data can be stored clearly in different folder. All the icon files stored in an individual folder, which the rest of the program can access easily. In addition, the values such as the string and color scheme will store in a single xml file. It keep

the program clean and gives more flexibility to update in the future. Overall, Android studio made the process of developing an application more easy.

4.2 Libraries

Library is some code that already built by some developer and open to other developer to use and implement. Using different library can increase the functionality of the application. The main challenge and work of using libraries is how to implement it correctly. All the libraries works well independently but the implementation to your own program can cause problem. Therefore, to make good use of a library, developer need to understand the library and able to apply it correctly. In addition, modifications of a library to fit the application's needs is possible and common. The application's need can be different from the original purpose. However, most of the libraries are customizable to fit different needs. We will discuss some major libraries used in this application to achieve the specific feature.

4.2.1 Activity, Context and Intent

An android application works with activity. According to the official android documentation, activities are the fundamental building blocks on the Android platform. [20] It is the entry point for a user's interaction and navigation. This main four activities used in this application are main view activity, input activity, twitter explore activity and individual Activity. To switch between activities, we need intent. Intent is the operation between activities and communication channel between applications. [22] To launch an activity from another activity, intent is used.

```
Intent intent = new Intent(getApplicationContext(), FormActivity.class);
startActivityForResult(intent, SAVING_DATA);
```

An intent can pass and receive data. The example of intent usage is starting the form activity and retrieve the result data. A form activity ask the user to enter the message and it will return all the required data to the calling activity, which is the main activity. Context is an interface to global information about an application environment. [21] Context used widely in this application to access different data. [22] Activity class will contain a context and application context, which allow the activity to get the information. However, some general classes in this application do not below to an activity. Therefore, the best practice of creating a class is passing a context argument. It will allows the user to use the context to retrieve the required information across the application. Activity, intent, context are the foundation of this android program. They are the basis of the all the functionality and features.

4.2.2 Adapter

Most of the views used in this application displayed more than one data. Therefore, an adapter is used to show a group of information. "An Adapter object acts as a bridge between an AdapterView and the underlying data for that view." [23] Also, it requires modifying the adapter to fit the requirement. MainExpandableListViewAdapter is the array adapter to provide the data for the main list view (3.3.1). This adapter will query the required data from the database and return them to the list view. Then, the listview will have the data to display. Additionally, the expandable feature require another modification to the adapter which this the parent and child holder. Since the information shows in the collapsed view and expanded view is different. To handle this feature, we need to customize the getChildview() and getGroupView() functions from the adapter that will implement the correct information. A SolventRecyclerViewAdapter is used in the tag cloud (3.3.2). It will also query the data from the database. In addition, the calculation of the font size and color setting happened here. An adapter provides access to the

data items [23] while a modificated adapter implemented all the special features and generate the expected view.

4.2.3 Auto suggestion

Autosuggestion provide a better user experience when entering the data. When user typed in tag label, it will automatically provide a drop down list of the suggestions. Android provided a MultiAutoCompleteTextView class that allows multiple suggestions in a single text view. It will consider text and provide new suggestion by a tokenizer. The default tokenizer is a comma, which means when user typed a comma, it will consider it later input as a new string and do the auto complete. However, it will be less user friendly if user need to type a comma every time. The solution is to implement a custom tokenizer and put the token automatically. The program used a space tokenizer to replace the default comma tokenizer. In addition, the tag toolbar (3.2.2) will handle the space generation. It will make decision on when and where to put a space. The general case is put a space in front of the tag label when user entered a tag. It will also handle the special case. First, if the tag label is the first character, it will not need to enter a space. Second, if there is already a space, it will not prepend a space. The provided suggestion is the unique tag list from the database and it will use an ArrayAdapter to retrieve the data. The threshold will determine when to show the drop down list. For example, a threshold 1 will show the list when any single character inputted. This technology applied to the input form and the search view.

```
tv_search.setTokenizer(new SpaceTokenizer());
String tagList[] = tagDBHelper.getTagsStringList("*");
ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1,
tagList);
tv_search.setAdapter(adapter);
tv_search.setThreshold(1);
```

4.2.3 Search View

To enable the search feature, we need to implement a search view. The simplest approach is to make use of the action bar and override it to a search box. First, we need to create an invisible text box in the action bar and will be shown when the search button had been clicked.

```
tv_search.setVisibility(View.VISIBLE);
tv_search.requestFocus();
imm.showSoftInput(tv_search, InputMethodManager.SHOW_IMPLICIT);
tv_search.setOnEditorActionListener(new TextView.OnEditorActionListener() {
  @Override
  public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
    if (actionId == EditorInfo.IME_ACTION_SEARCH) { … }
  }
```

The system will handle the focus and the visibility of the soft keyboard. After all the setup, the user can enter the data with the tag toolbar. Finally, the searching process will start by clicking the search button in the keyboard. To implement the action of it, we need to create OnEditorActionListener. It will handle the search button and start the searching process.

4.2.5 Asynctask and Listener

Asynctask used to download data from Twitter and a listener will handle the action after the task is finished. TweetsListener is the interface used in this application and it has a function to handle the tweet-completed action. LoadTweets is a public class extends AsyncTask that will take the context and a TweetListener. It will execute the call with the specified argument and download the data from Twitter. After the downloading process finished, it will pass the result to the TweetsListener where the called activity can access the result. The other usage of AsyncTask and Listener is download the profile image since Twitter will only provide a URL of the image. With AsyncTask and Listener, the program can access the internet to retrieve relevant information and handle the result.

4.2.6 Android MP Chart

Android MP Chart is a third party library that provide a simple way to generate chart. [2] This program used the library to generate the pie chart and line chart. First, need to retrieve the information for database and transform it the chart data entry. A chart data entry has a label and the amount. After setting up the data, we can convert the entries to the corresponding chart type data set. To create a chart, we will use the data set. The library is completely flexible. For pie chart, we can enable the spinning features, a hole in the middle and the color. For line chart, we can set up the range and the space between each label. In addition, the feature of zooming allow user to get a detail view. All customization can easily set by using a function to set.

4.2.7 Keyboard Visibility Event

The tag toolbar (3.3.2) connected to the soft keyboard where it need to handle the keyboard visibility. The solution is using the KeyboardVisbiliyEvent libraries created by yshrsmz. [19]

```
KeyboardVisibilityEvent.setEventListener(this, new KeyboardVisibilityEventListener() {
            @Override
            public void onVisibilityChanged(boolean isOpen) { … }
}
```

It is a third party library and allow creating a KeyboardVisibilityEvent class to handle what to do when the keyboard is visible and invisible. The function onVisibilityChanged will trigger every time the visibility of the soft keyboard changed. Therefore, we can show the tag toolbar if the boolean value is true or hide it when the variable is false.

4.3 Twitter

Twitter is an individual and separate application from Android. It is difficult to connect to twitter directly from code because of the permission and security aspect. However, Implementation with Twitter can be easy with a third party library. Using a library to integrate,

it will apply the security to our own application. In addition, it provided multiple class for developer to choose what the best fit to the application is. Twitter has many features, however, not all of the features needed in this application. Therefore, we need to decide what method should be used to fulfill the requirements.

4.3.1 REST APIs

REST API is the official twitter development library. According to Twitter, REST APIs provide "programmatic access to read and write Twitter data. Create a new Tweet, read user profile and follower data, and more." [24] The technology behind REST APIs is using a URL to connect to the server. Every features of Twitter can access by an URL. Twitter provided a documentation on how to configure the URL to get the information we need. It is good idea to use URL to access the server, however, it is also more complex to use. Different third party library provided a easier to access the REST APIs and more user friendly to integrate with Android.

4.3.2 Fabric - Twitter Kit

We used Twitter Kit from the Fabric library. "Twitter Kit is designed to make interacting with Twitter- whether it's logging someone in, displaying Tweets or Tweeting from your app- seamless and efficient." [25] The three-basis class of Twitter Kit is TwitterCore, TweetUi and TweetComposer. TwitterCore provided the method to connect to Twitter. It will use the native Twitter application to do the connection and grant all the required permission. An active twitter session will be generated and allow the rest of the program to use. TwitterCore also provided a StatusServices class that can access the some of the REST APIs method. Instead of using a URL, StatusServices converted the URL to a method where the developer need to pass to relative arguments. Then, it will create a CallBack class, which can understand as a class that contain the

URL and ready to execute. When executing, it will run the URL and do the corresponding task. The result will also be returned to the class when applicable. TweetUi is the class that provided a visualization of a tweet. A Tweet contains different information such as name, message and the profile image. Therefore, TweetUi will illustrate the Tweet and provide a preview of the Tweet in this application. TweetComposer gives the ability to use the native Twitter application to post a tweet. Although, StatusServices with REST APIs can post a tweet, TweetComposer provided another option to tweet. Using TweetComposer will provide a native Twitter interface to post a tweet because it linked to the Twitter application. With the original design, user may find it safer to use. In addition, it provided all features that Twitter provided when entering a new Tweet. For example, attach a location and access the databases when entered a hashtag. This application provided both approaches to user to choose when sharing their transition to Twitter.

4.4 Databases

In this section, we will talk about how to store and load from databases. The default local database language of Android is SQLite. SQLite provided a simple structure to store different type of data and retrieve information from databases. An Android SQLiteOpenHelper class provided a useful interface to interact with the SQLite database. The create, update, delete process can be completed within a single function. There are three major types of data used to store in this application, INTEGER for real number, REAL for decimal number and TEXT for a string. To insert a row to the databases, we need to use a ContentValues. ContentValues is Android class that allow multiple data with label in a single class. SQLite is able to insert a ContentValues to the databases with the matching label between the database's table and the ContentValues. After defined a SQLite format query, we can use the query to read the database

by a cursor. rawQuery() is the function that used to apply a SQLite query and return a result as a cursor. A cursor works as a combination of pointer and table. It can go to different row and column to get the information.

```
String s = cursor.getString(cursor.getColumnIndex(KEY_MESSAGE));
```

By using the column label, we can retrieve the needed column from the result in a single method. Avoiding data leak is an important security issue. Therefore, closing the database and the corresponding open helper every time after used is a good practice. SQLite in Android provided an easy way to store and query data. With the build in SQLite open helper class, you can access the databases from other class or activity in a safe way.

Chapter 5

Conclusion

In conclusion, this thesis created a personal Android finance management application. With the Android's libraries, different features can be implemented successfully. It using the tagging and messaging approaches to enter data that moves away from structured data entry. Comparing to structured data entry, tagging is a new trend to enter data, which is more flexible and interesting. Using tags to keep track of your spending is a new idea of finance management. The technology culture keeping changing every day and some of the daily routines can be changed to improve the efficiency. With the high demand of use of mobile devices, a more mobile friendly way of entering data should be considered. Using mobile phone is a very different experience of computer of handwriting. Using tag and messaging input is the solution. This application also

provided different visualization of the data. User can easily see the spending in a pie chart or in a line chart. With the concept of tag, user can also illustrate their tag in a tag cloud format. The expandable list view allow the user to see their transaction along with the tags. Social media is part of our life. The application created in this thesis allow user to share and follow with the twitter integration. Finally, the aim of this thesis to create a new experience of personal money management.

5.1 Lesson Learned

The first thing I learnt in this thesis is the process of planning and generating ideas. The ideas need to be organized well. To create a good application, the planning is an important part. Designing all the features is one of the biggest thing I learnt in this thesis. A user friendly application required a good design. An application that customer will keep using need to be simple to use. Therefore, I learnt different approach of creating a user-friendly design. I compare different layout and design and try to find the best. I learnt how to set up a good layout that will satisfy the entire requirement. Android development is the main part of the thesis. Although the programming language of Android Java which I already have experience with it, there are still many areas for me the explore. Android provided a lot of libraries and specific class. Learning and implementing different techniques to the android app is the key things of this thesis. Xml design, android activity class, SQL database, deal with the user input, generate visualization to twitter integration are some of the sample techniques used. Finally, the social media culture is one of the many things I learnt in this thesis. Twitter is a social media platform that used by the majority. The progress of working on this thesis taught me many things from planning, theory, design, mobile application development to social media culture research.

5.2 Features and limitations

The key feature of this application the way of input, which use a single message and tag to input data and do money management. The limitation is user still need to physically type in the data with their hand. User can able search through the databases and return the wanted information. The search view allow the user directly enter the searching query. In addition, the application provided different layout that allow the user select and generate the visualization easily. However, the visualization can be limited by the mobile devices hardware. To create a better visualization and in a faster time, the hardware need to be supported. Finally, the feature of sharing and follow with Twitter gives user a way to enjoy the social media with this app. Nevertheless, Twitter is individual company and have their own limitations. Using an existing social media platform can benefit from the existing user databases but the integration and features will be limited.

5.3 Future development

First, the application can improve by optimizing the performance. Storing and loading the data used the most memory. It will read all data when generate the view. The performance is okay when dealing with small amount of data but the speed will decrease with more data. Therefore, a partially loading data approach can be implemented to improve the performance. In this stage, the program can only query with only one tag at each time. It means user can only view the information about one tag only right now. To improve this, we can use a multiple argument query to allow getting data with more than one tags. Better visualization can improve the user experience. Using online databases will be another feature that can be implemented. It allows following user directly from online database with any social media integration. The application

can be a standalone social media platform. Additionally, it can save some local space of the device. The program predefine the type of tags, which fixed the data type.

If user can customize the tag they want, add or delete the tag they will use, a more flexible experiment will be built. Therefore, open to user-defined tag is another area for future development. Finally, testing and debugging is a necessary step before publishing the application. A beta run need to be started for checking the functionality. The application should fixed all known bugs and operation functionally to maximize the ability. Any further modification and suggestions to the application are opened to improve the performance.

Bibliography

[1] Twitter Usage Statistics. (n.d.). Retrieved March 27, 2017, from http://www.internetlivestats.com/twitter-statistics/#sources

[2] P. (2017, March 24). PhilJay/MPAndroidChart. Retrieved March 28, 2017, from https://github.com/PhilJay/MPAndroidChart

[3] Asur, S., & Huberman, B. A. (2010). Predicting the Future with Social Media. 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. doi:10.1109/wi-iat.2010.63

[4] Stanford, K. H. (2016, August 18). Social Media Culture. Retrieved April 12, 2017, from http://www.business2community.com/brandviews/growing-social-media/social-media-culture-01635249#g5OtykFcyHEJG96y.97

[5] Tardanico, S. (2014, April 15). Is Social Media Sabotaging Real Communication? Retrieved April 12, 2017, from https://www.forbes.com/sites/susantardanico/2012/04/30/is-social-media-sabotaging-real-communication/#783267962b62

[6] The Editors of Encyclopædia Britannica. (2005, May 12). Socialization. Retrieved April 12, 2017, from https://www.britannica.com/topic/socialization

[7] About the Black Lives Matter Network. (n.d.). Retrieved April 12, 2017, from http://blacklivesmatter.com/about/

[8] Sillito, D. (2015, May 28). Hashtag is 'children's word of year' Retrieved April 12, 2017, from http://www.bbc.com/news/entertainment-arts-32902170

[9] Knight, S. (n.d.). SOCIAL TOOL AND CULTURAL PHENOMENON: WHAT'S IN A HASHTAG? Retrieved April 12, 2017, from https://www.zestdigital.com/social-tool-and-cultural-phenomenon-whats-in-a-hashtag/

[10] Structured data entry. (n.d.). Retrieved April 13, 2017, from http://clinfowiki.org/wiki/index.php/Structured_data_entry

[11] Twitter. (n.d.). Retrieved April 13, 2017, from https://about.twitter.com/company

[12] Lee, D. (2013, November 07). How Twitter changed the world, hashtag-by-hashtag. Retrieved April 13, 2017, from http://www.bbc.com/news/technology-24802766

[13] Nations, D. (n.d.). What Is Exactly Is a Tweet, Really? Retrieved April 13, 2017, from https://www.lifewire.com/what-is-a-tweet-3486211

[14] Material Design. (n.d.). Retrieved April 13, 2017, from https://material.io/

[15] Materialize. (n.d.). Retrieved April 13, 2017, from http://materializecss.com/about.html

[16] Oxford Dictionary Tag Cloud. (n.d.). Retrieved April 13, 2017, from https://en.oxforddictionaries.com/definition/tag_cloud

[17] Jones, S. (2014, November 18). The Psychological Benefits of Simple Design | SEJ. Retrieved April 13, 2017, from https://www.searchenginejournal.com/psychological-benefits-simple-design-less-really/118906/

[18] Gichevski, A. (n.d.). What is a Tag Cloud and How to Calculate it by Formula. Retrieved April 14, 2017, from http://agichevski.com/2014/05/14/what-is-a-tag-cloud-and-how-to-calculate-it-by-formula/

[19] Y. (2017, April 03). Yshrsmz/KeyboardVisibilityEvent. Retrieved April 14, 2017, from https://github.com/yshrsmz/KeyboardVisibilityEvent

[20] Activities. (n.d.). Retrieved April 14, 2017, from https://developer.android.com/guide/components/activities/index.html

[21] Context. (n.d.). Retrieved April 14, 2017, from https://developer.android.com/reference/android/content/Context.html

[22] Intent. (n.d.). Retrieved April 14, 2017, from https://developer.android.com/reference/android/content/Intent.html

[23] Adapter. (n.d.). Retrieved April 14, 2017, from https://developer.android.com/reference/android/widget/Adapter.html

[24] REST APIs — Twitter Developers. (n.d.). Retrieved April 15, 2017, from https://dev.twitter.com/rest/public/

[25] Twitter Kit. (n.d.). Retrieved April 15, 2017, from https://docs.fabric.io/android/twitter/