



澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Outstanding Academic Papers by Students

學生優秀作品



University of Macau

Faculty of Science and Technology



澳門大學

UNIVERSIDADE DE MACAU

UNIVERSITY OF MACAU

Manage Your Mobile Time

by

Ho Sin Nga, Student No: DB027016

Final Project Report submitted in partial fulfillment
of the requirements of the Degree of
Bachelor of Science in Software Engineering

Project Supervisor

Dr. Zhuang Yan, Sofia

08 October 2014

DECLARATION

I sincerely declare that:

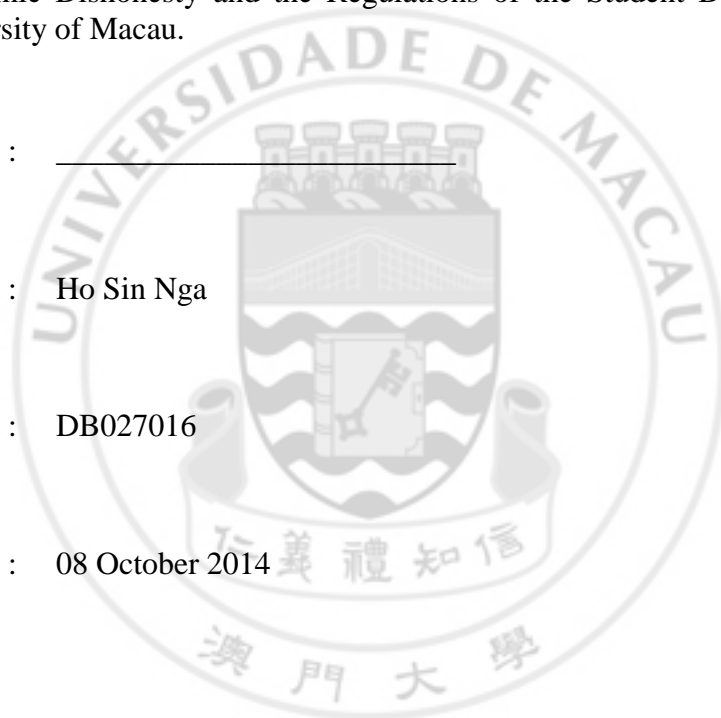
1. I and my teammates are the sole authors of this report,
2. All the information contained in this report is certain and correct to the best of my knowledge,
3. I declare that the thesis here submitted is original except for the source materials explicitly acknowledged and that this thesis or parts of this thesis have not been previously submitted for the same degree or for a different degree, and
4. I also acknowledge that I am aware of the Rules on Handling Student Academic Dishonesty and the Regulations of the Student Discipline of the University of Macau.

Signature : _____

Name : Ho Sin Nga

Student No. : DB027016

Date : 08 October 2014



ACKNOWLEDGEMENTS

I would like to express my utmost gratitude to UM for providing the opportunity to carry out a project as a partial fulfillment of the requirement for the degree of Bachelor of Science.

Throughout this project, I was very fortunate to receive the guidance and encouragement from my supervisor Sofia. She gave us a lot of suggestions on implementing the project and instruct us to think more from the user's point of view. I've learnt many in this project and in her instruction. I express my sincere thanks to her.

Secondarily, I want to thank the Accreditation of PPRD Project Competition of Computer Science. They give us a lot of suggestions how to present our work better.

Finally, I want to thank my partner Ao Ieong Kit Ha, Sue. We worked closely during the project, and we fight together in this project. This project can't finish without her cooperation.



ABSTRACT

Out of the world's estimated 7 billion people, 6 billion have access to mobile phones. The average Smartphone user tends to spend around two hours a day using their gadget, according to a study. However, time management is important for improving productivity; people may be interested in finding out exactly how much time they have spent on their Smartphone daily.

In particular, people may like to know how much time they have used in the various activities per day. It is including browsing the Internet, checking social networks, listening to music, playing games, making calls, text messaging, checking/writing emails, reading books, watching TV/films, taking photographs, etc. Statistics about time spent on a particular activity can be more specific (for example, people spend different time on different games). The project aims in helping people to know more about their time spent on their smartphone per day, thus opt for better time management.

To achieve the project goal – to help users to manage their time, we implement an application for android OS to track the time usage of the mobile phone. Since IOS can't provide enough permission for us to implement, so the project, we only focus on implementing the app in android OS.

Since my partner and me have not implemented app yet. So we do a lot of study and research of implementing app and understanding android OS at the beginning.

In this report, it is mainly talked about the Reminder function and the Data function of the application. Besides, including our understanding of developing the app. My partner Ao Ieong Kit Ha will introduce the main function in this app.

TABLE OF CONTENTS

CHAPTER 1.	INTRODUCTION.....	1
1.1	Background	1
1.2	Objective	1
1.3	Work flow	1
1.4	Work Scheduling	1
1.5	Task Allocation	1
1.6	Environment	1
CHAPTER 2.	LITERATURE SURVEY/RELATED WORK	6
2.1	Related application	6
2.2	Eclipse	6
2.3	Android SDK	6
2.4	ADT plugin.....	7
2.5	AchartEngine.....	7
CHAPTER 3.	FUNCTIONAL SPECIFICATION	8
CHAPTER 4.	SOFTWARE DESIGN SPECIFICATION	9
CHAPTER 5.	IMPLEMENTATION NARRATIVE AND DESCRIPTION	18
5.1	Programming language	18
5.2	Software environment	18
5.3	Implementation narrative	18
5.3.1	Design	錯誤! 尚未定義書籤。
5.3.2	Difficulty	25
CHAPTER 6.	SYSTEM QUALITY	錯誤! 尚未定義書籤。
CHAPTER 7.	EVALUATION.....	26
CHAPTER 8.	SUMMARY.....	28

CHAPTER 9. APPENDIX.....錯誤! 尚未定義書籤。



LIST OF FIGURES

Figure 1: Data Flow Diagram	10
Figure 2: Flow Chart for the web data usage function.....	12
Figure 3: Flow Chart for the reminder function.....	13
Figure 4: Flow Chart for the reset app usage function.....	14
Figure 5: Flow Chart for the reset web data usage function	15
Figure 6: Flow Chart for the reset reminder (rule) function	15
Figure 7: AppdataUsage and Totaldatausage relationship and detail	16
Figure 8: InstalledApp and RuleInfo relationship and detail.....	17
Figure 9: ResetData detail.....	17
Figure 10: OpenTimesTable detail	17
Figure 11: manage your mobile time application size	26
Figure 12: App Timer Mini 2 application size.....	26
Figure 13: Android System web data count funtion	27

LIST OF TABLES

Table 1: Work Scheduling	3
Table 2: Task allocation.....	5
Table 3: function Description	8



CHAPTER 1. INTRODUCTION

1.1 Background

Nowadays, Most of the people have at least one smartphone in their pocket. Smartphone is so useful to let people are more and more relied on it. People use it to check mails, browse the Internet, play games, take photos, listen to music, etc. If we spend too much time day by day, it will cause neck pain, short sight. Our app can show the usage of each application in your smartphone; also you can set up a reminder to alert you to put down your phone [4].

1.2 Objective

The objective of this project is to manage user's time to use the Smartphone so that user doesn't become a morbid. In the project, we develop an application to monitor the status of the mobile usage. It shows the time of each application, which in the user's mobile. We hope the user can reduce the time on the phone through this application [4].

1.3 Work flow

We list the steps of the project. There are eight parts:

- The decision of the requirement
- The APP's design for the function
- The APP's design of the interface
- Implementation of function
- Implementation of interface
- Testing
- Modification
- Report

1.4 Work Scheduling

Table 1 shows the workload in the given period.

1.5 Task Allocation

Table 2 shows the task allocation.

1.6 Environment

Smart phone's operating system mainly divided into two types: android and IOS. Since we need a permit to access to the system to get the other application's data and the system's resources, IOS cannot provide such big permission that we only focus on the mobile in the Android OS. In the project, we have used Java, SQLite and XML. All collected data we stored in SQLite. The interface of the application was written in

XML. The activity and service between the application and the database was written in Java [4].



Table 1: Work Scheduling

Date	Work
23/09/2013 to 31/10/2013	Search the related application to see the availability in both IOS and android. Report the progress to our supervisor.
01/11/2013 to 15/11/2013	Study how to develop an application for android. Study how to connect SQLite to the application.
16/11/2013 to 30/11/2013	Discussion for the functions of the application. Report the progress to our supervisor.
01/12/2013 to 15/12/2013	Confirm the four main functions and design the interface for the application. Report the progress to our supervisor.
16/12/2013 to 31/01/2014	Find related information of the android system. Study how to get the system information.
01/02/2014 to 15/03/2014	Implement the main functions: Count time, chart, and reminder. Meeting with our supervisor.
15/03/2014 to 20/03/2014	Design and create database table to save the record. Implement the main functions: 3G/WIFI. Meeting with our supervisor.
20/03/2014 to 04/04/2014	Improve the functions Count time & Chart. Integrate the record in database to generate the chart.
05/04/2014 to 15/04/2014	Improve the function 3G/WIFI.
16/04/2014 to 07/05/2014	Redesign the reminder function.

	<ul style="list-style-type: none">• Add settings for each application.• Add settings for reminder mode. Draft the report.
08/05/2014 to 14/05/2014	Improve the whole application. <ul style="list-style-type: none">• Interface.• Add a sorting feature to count time.• Add custom setting.
15/05/2014 to 22/05/2014	Test the functions and find bugs for one week.
22/05/2014 to 28/05/2014	Fix all the bugs in the testing we found.
28/05/2014 to 05/06/2014	Write the final report

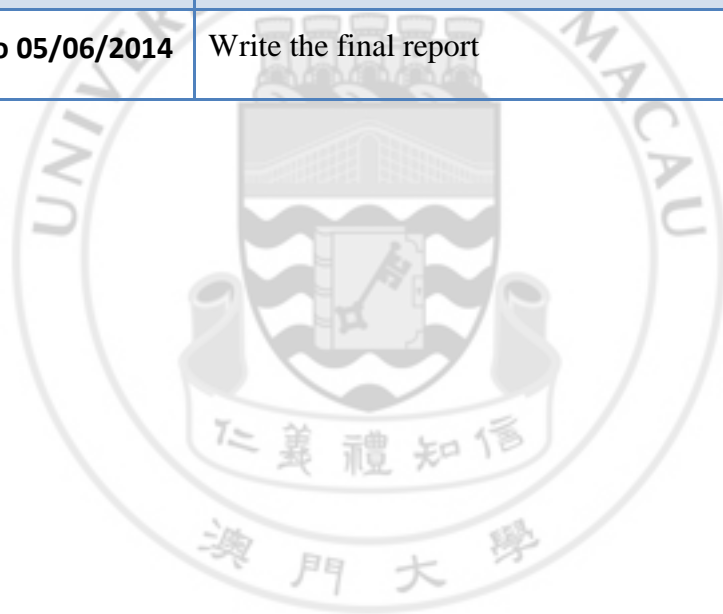


Table 2: Task allocation

Task	Responsible person
Project study	Sue & Veronica
System study	Sue & Veronica
Getting data	Sue
Creating database	Sue & Veronica
Create reminder & 3G/WIFI functions	Veronica
Integrate all the functions	Sue & Veronica
Testing	Sue & Veronica

CHAPTER 2. LITERATURE SURVEY/RELATED WORK

At the beginning of the project, we focus on search some applications that are related to our project. We hope that we can find any clue in both IOS and android. Actually, we get many related applications in android. Android is a good platform, which is a fully open-source operating system. However, we can't find any applications that are related to our project in IOS. We only can find some applications that need to be installed on a jailbreak iPhone. We start to search why the iPhone doesn't have such application to manage mobile time. The answer is that iPhone blocks some permissions to protect the phone and all the developer's application so that no one can get such information to implement this kind of application only if the phone has been jailbreak. A jailbreak iPhone relieves all permissions, but it makes the phone become unstable. We don't expect the user jailbreak the phone for our application. So we decide to give up implementing the application in IOS, only implement an application that in Android OS.

2.1 Related application

In the store, there are some related applications. They can count the time for your usage also, but some have different addition functions.

In the beginning, we spend a period of time to search them in the word 'time', 'manage' etc. However, we can't find any related applications. We realized that maybe such an application's name didn't have those words. We started using another word 'count'; 'track', and we get a little prospect. After the research process, we found a fact those were those applications have not been widely used.

Although we are familiar with Java's environment, we still need to study about the android system and specification. To develop an application, we need to design interface with XML.

2.2 Eclipse

Eclipse are the tools for Java developers use to create Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn, EGit and others. In this project, we use Eclipse for Java EE. Eclipse can install many plugins. In order to implement our project, we need to install the Android SDK plugin and ADT plugin in Eclipse.

2.3 Android SDK

The Android SDK provides us the API libraries and developer tools necessary to build, test, and debug apps for Android. API (Application Programming Index) is a group of libraries for the Android developers to develop android application. Android SDK also provides a set of the virtual system for us to debug. This is the core tool to implement the application.

2.4 Java SE Development Kit (JDK)

Java SE Development Kit (JDK) is for Java Developers. It includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.

2.5 ADT plugin

Android Development Tools (ADT) is a plug-in for the Eclipse IDE that is designed to give you a powerful, integrated environment in which to build Android applications.

ADT extends the capabilities of Eclipse to let you quickly set up new android projects, create an application UI, add packages based on the Android Framework API, debug your applications using the Android SDK tools, and even export signed (or unsigned) .Ape files in order to distribute your application.

2.6 AchartEngine

It is user friendly that we use a chart to present our data. AChartEngine is a charting library for Android applications. It currently supports the following chart types:

- Line chart
- Area chart
- Scatter chart
- Time chart
- Bar chart
- Pie chart
- Bubble chart
- Doughnut chart
- Range (high-low) bar chart
- Dial chart / gauge
- Combined (any combination of line, cubic line, scatter, bar, range bar, bubble) chart
- Cubic line chart

It is user friendly that we use chart to present our data.

2.7 GenyMotion

GenyMotion is one kind of android emulator. We can use it toe simulate a wifi environment or low battery environment. It help us so much that it emulator those phone we can't test in a real phone.

CHAPTER 3. FUNCTIONAL SPECIFICATION

In our application, there are four main functions (see the detail in Table 3). They are called App Usage, App Chart, reminder and 3G/WIFI. They are shown in the top of the app using by tabs, we were using four different tabs to divide it.

Table 3: function Description

Function	Description
App Usage [4]	It used to count the time of each application's usage in the smart phone.
App Chart [4]	The chart shows the application usage in the diagram by using the App Usage data.
3G/WIFI	It used to show the data usage of 3G/WIFI. It can show the detail of each application's data usage.
Reminder	The reminder function is optional. User can set the amount of the time of using which application. When the user uses the application over the limit, the reminder will warn the user.

The four main functions (see the detail in Table 3) is the mainly task of our project. Our application mainly focuses on the installed application in the smartphone. In addition to showing the using time of the application, our application will also show the information of those installed applications. That information includes the application installed time and the application size.

The App Usage tab shows the count time of each application usage time in the Smartphone. The count time can sort by application name or sort by application usage time. User can choose the application to view more information [4].

The App Chart's tab shows a pie chart. It is according to the data that shows in the App Usage function to generate. We choose pie chart because it is clear to show what user's most used application is. We hope the chart can be as a warning for user when they see in first sight. User also can click on the application pie to show the usage percentage [4].

The 3G/WIFI's tab used to show the web data usage of the Smartphone. In this tab, it has a pie chart, the pie chart only used to show the data usage proportion of Wi-Fi and 3G. On the other hand, it in addition shows the data usage of each application on Wi-Fi, on 3G or all data usage.

The reminder's tab is similar to an alarm. It alerts users when the application is used over the limit, which is set by the users themselves. User can add a rule to create a reminder of an application.

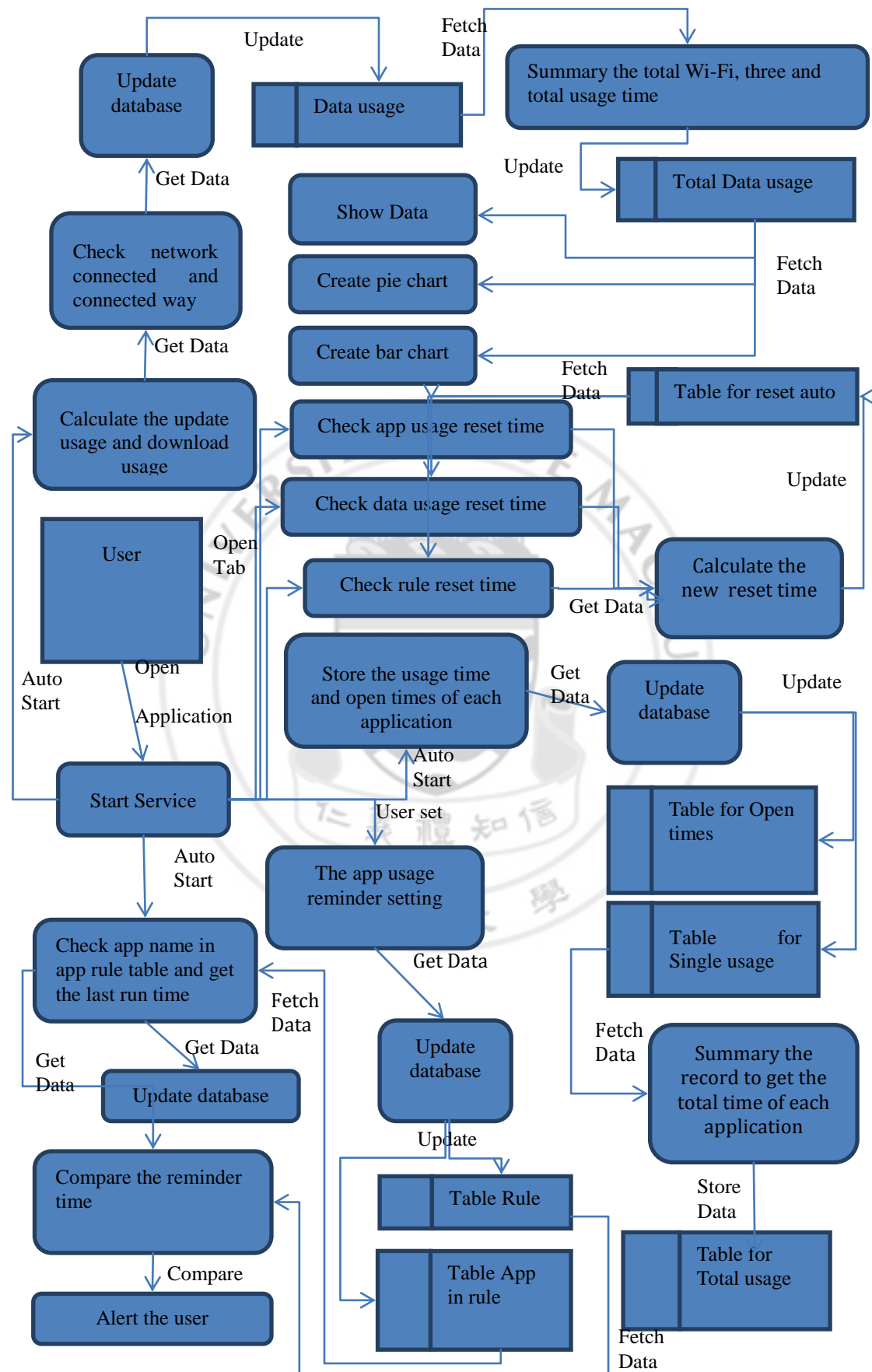
CHAPTER 4. SOFTWARE DESIGN SPECIFICATION

Our application can't operate without a database. We store the important data in a database. It is because our function has relied on the data in the database. The following Data Flow Diagram is focused on the activity flow but doesn't contain all tables which on our database. It is because other tables will show by my partner's report (see Figure 1).

In the Data Flow Diagram, the functions of our application are core function. Our application has back-end service; this back end service would count the app usage time, web data usage, open times, check reminder, reset time and monitoring the battery.



Figure 1: Data Flow Diagram



4.1 Flow Chart

In this session, I am going to introduce the flow chart of the function of our application which I handle. In the following part, I will introduce 3G/Wifi usage, reminder and data reset check.

First, I am going to introduce the count of 3G/ WI-Fi usage (see Figure 2). Since the service is started, for the 3G/ WI-Fi usage function. It will check is there the network is connected at the first. If the networked, our application will get all the running application of the Smartphone. If there have been running applications, it will check is it the first time opens our application after reboot or power on the Smartphone? If it is the initial time opens our application, the start up counter, else get the last 3G/ WI-Fi data usage, which is stored in the database. After that, the function is trying to get the sent and received data usage of that running application by the function provided by Google developers [2]. Since some of the Smartphone didn't support this function, the function would check the data usage by getting the file on Smartphone, which store the web data usage if the function returns less or equal to zero.

After getting the total data usage of that running application, the function of web data usage would check the connecting way in the next. If it is connected by WI-Fi, if connecting has at this moment changed, the result of the data usage data will update by type in 3G, if connected way didn't change, the result of the data usage data will insert or update by type in WIFI. If it is connected with 3G, if the check that the connecting just changed, the result of data usage data will update by the type in WI-Fi, if the connected way didn't change, the result of data usage data will insert or update by type in 3G. Repeat these steps every second then update the data immediately, this function will stop when the service stop.

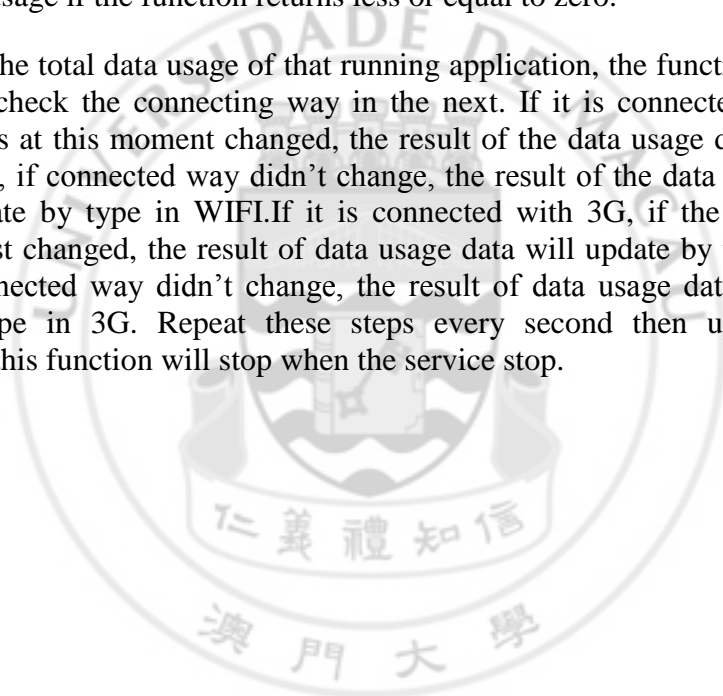
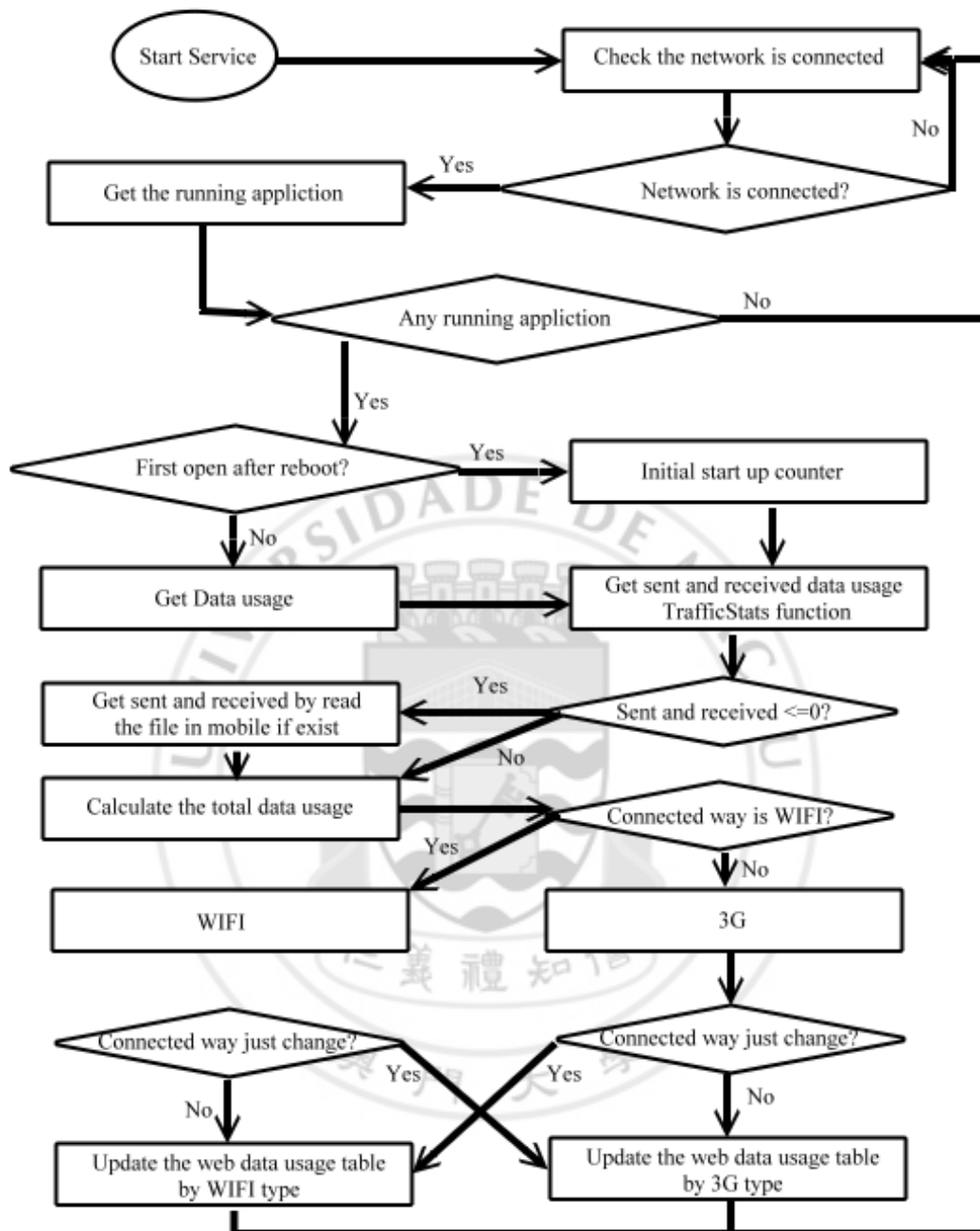


Figure 2: Flow Chart for the web data usage function

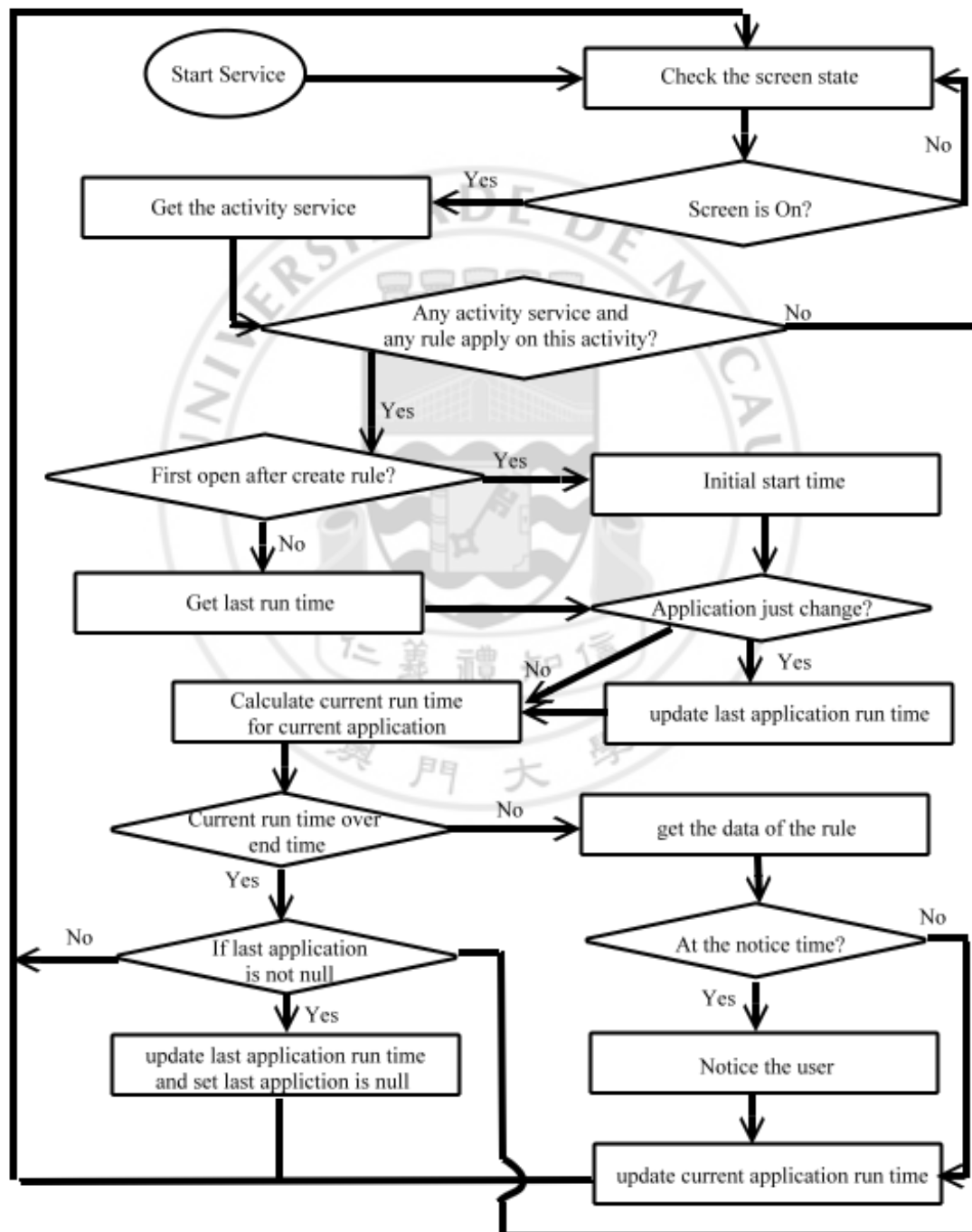


Secondarily, the reminder function is one of the main functions of our application (see Figure

3.). It also starts with the service. Initial, it will get all activity service applications when the screen is on. A check is there any rule applies to the activity application and gets the data if there have any rule apply to this application. Get the application last start time, if it is started at the first time, initial it. If the activity application just changes, update the run time of the last application, then calculate the current active time. Check if the application is not over the end notice time, then checks if it at the noticeable time

me, if in the noticeable time, notice the user by the noticeable way, the user chooses. If the application is just over the end time, update the run time for the last application and initial the runtime. If there is no rule applies on this activity application or if the screen is off, if the last activity application is not null, update the last activity application run time and initially, the run time. Repeat these steps every second then update the data immediately, this function will not stop when the service is an activity.

Figure 3: Flow Chart for the reminder function



Finally, the reset data function help user to reset the data by the user set. This function should use the data which user input. Those data are input when the user enters our application. The reset data function need to check is their data it relies on app usage. Yet there are some data it relies on app usage (see Figure 4.), it will get the data reset time and the present time. If the present daytime is the time needs to reset the app usage data, delete the app usage data and update a new reset time. In the result of web data usage (see Figure 5.), it is similar as the result of the app usage; the difference is the reset table of web data usage is the total web data usage table. The new result is based on the data, which set by the user. For the rule reset function (see Figure 6.). First of all, the function should check is there any rule reset data in the database or not. If there is some data that is relied on a rule, get the rule number of this rule and check if it's the time to reset the rule, if it is the reset time, reset the data and update new reset time. This function would repeat every hour. This function will not stop when the service is an activity.

Figure 4: Flow Chart for the reset app usage function

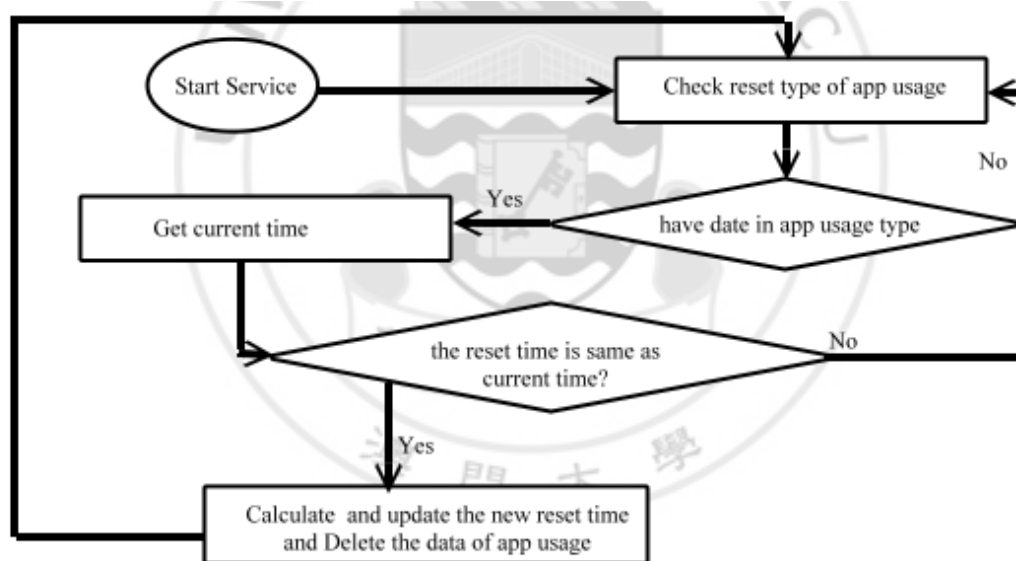


Figure 5: Flow Chart for the reset web data usage function

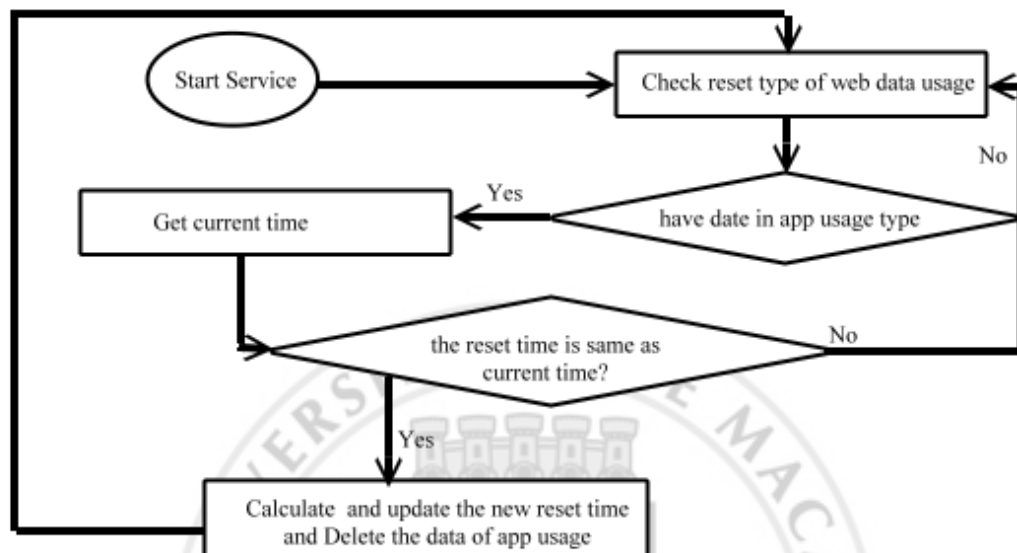
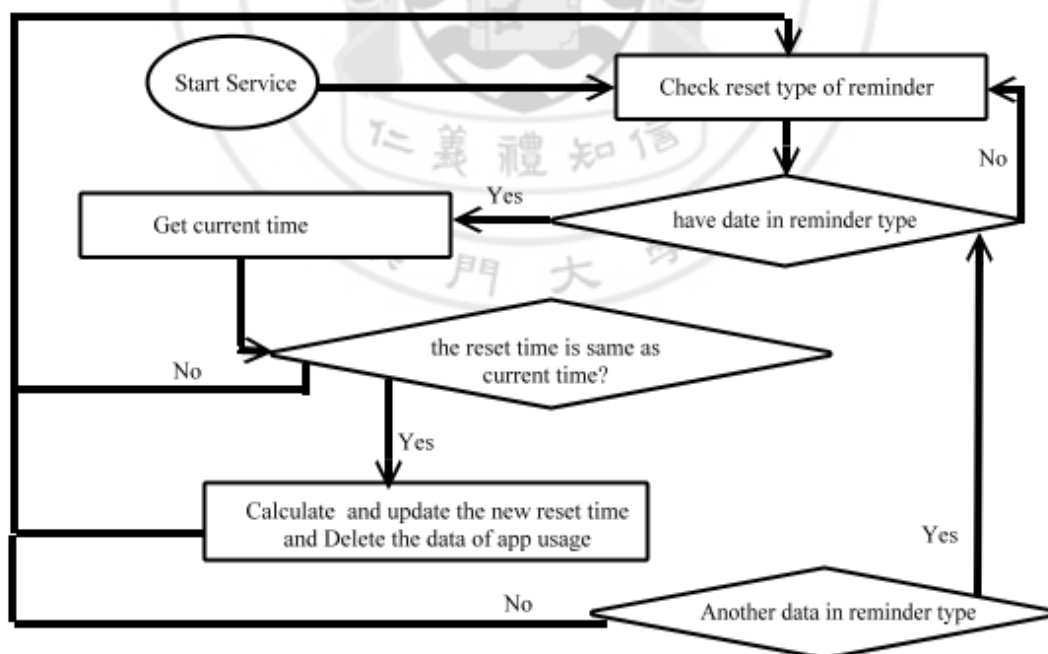


Figure 6: Flow Chart for the reset reminder (rule) function



4.2 Database Design

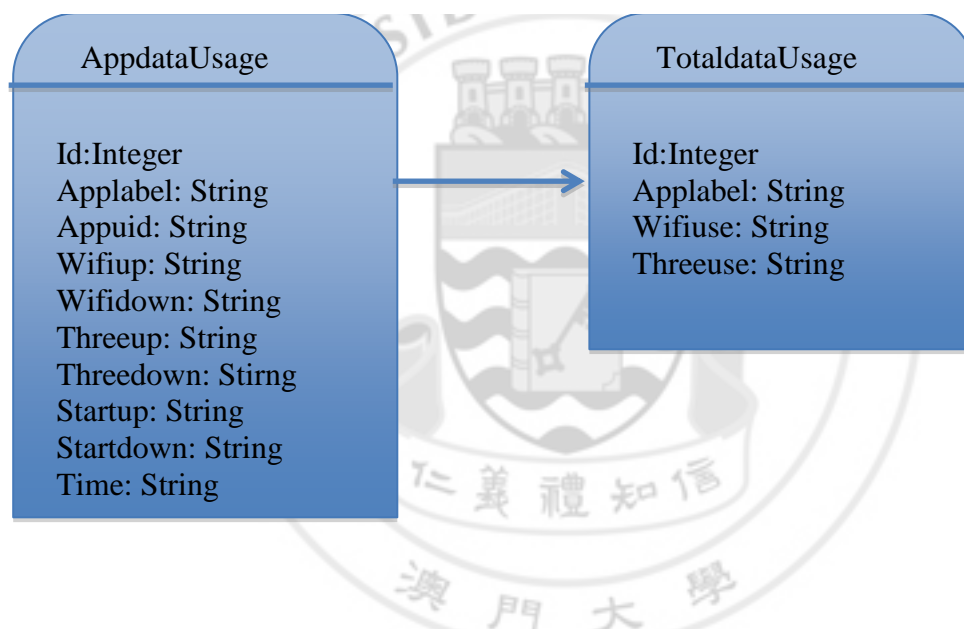
In this final year project, it is using SQLite to our database. SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional

SQL database engine [3]. it is an embedded database, we no need to create a server to access the database. This database is a small size and not different to create.

In the web data usage function, I have use two table. One of the table is to store all web data usage by time. It can contain all useful data for all related function, shch as the nearest 7 days data usage information. The second table is store the sum of a day of web data usage. It can use to calculate the total data usage, the web data usage chart and the web data information list.

Appdatausage is to store the current data usage of each applicaton. It contains the applabel, app uid and the wifi, 3G send and received usage and the time of the web data usage time. The total data usage table is calculate all wifi usage, 3G usage and total usage in one day. TotaldataUsage table update when AppdataUsage have data (see Figure 7.),.

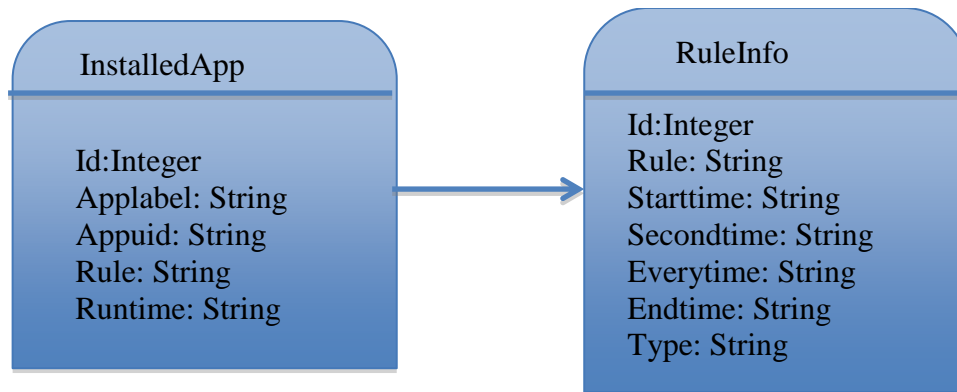
Figure 7: AppdataUsage and Totaldatausage relationship and detail



In reminder function, I also use two table to store the data. One table is to store the application data which have set in reminder function. It contain all application which have been set and store the run time of the application. The other table is store the reminder rule data, it use to check the application reminder time.

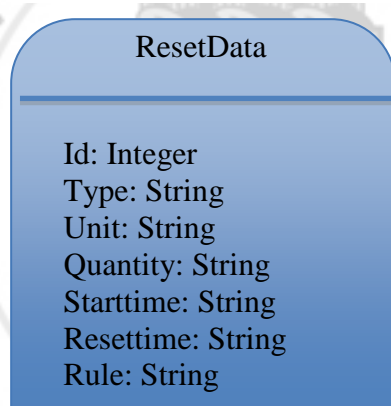
InstalledApp is to store the application data which set in reminder rule funtcion. It contain the applabel, the rule id and the rule id that is set on this application and it also contain the application run time that the application already run after generate this rule. RuleInfo table is to store the data of user set this rule. It contain the rule id, first warning time, second warning time, again warning time and end warning time and it also contain the warning way. (see Figure 8.)

Figure 8: InstalledApp and RuleInfo relationship and detail



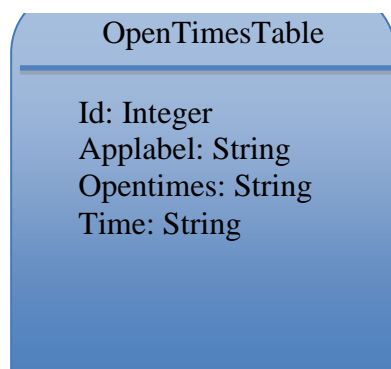
reset function, I have create a table named ResetData, it contain all type of reset. We can check the different type of reset by check this table type is exist or not. The type of reset of app is "Rank", the type of reset data usage is "Data" and the type of reset rule is "Rule". In reset rule data, it also contain the rule number to determine which data is for which reminder rule. The table include the reset type, the reset time unit which store the data like "day", "week" or "month". Quantity , the data start time, the reset time and the rule id. (see Figure 9.)

Figure 9: ResetData detail



On addition function it to calculate the open times, which store the application open times named as "OpenTimesTable". It stores the application label, the open times and the date of open the application. it summary all open time of one application by the open date. (See Figure 10.)

Figure 10: OpenTimesTable detail



CHAPTER 5. IMPLEMENTATION NARRATIVE AND DESCRIPTION

Since I am in charge three main parts in our final year project. It is 3G/ Wi-Fi data usage, Reminder function and reset. So the follow will talk about this two function and one additional function this is count open times function.

5.1 Programming language

Since the programming language of android application development, So we use java to implement our final year project.

5.2 Software environment

Our application needs runs on android 3.0 platforms or above. We suppose the phone have 1GB RAM to 2GB RAM to have a good performance.

5.3 Implementation narrative

5.3.1 Implementation

I am going to show you the way to implement the following function step by step.

5.3.1.1 3G/ Wi-Fi data usage

Since in this function, we need to get the activity applications a first. At the beginning, I try to get those applications by get the permission which needs a network, but we find that will use a lot of resources, and it will make our application in a large delay. So I change the way to get the activity application like follow:

```
public String[][] getInfo2(){
    PackageManager pm = this.getPackageManager();
    ActivityManager am = (ActivityManager)
this.getSystemService(Context.ACTIVITY_SERVICE);
    List<ActivityManager.RunningAppProcessInfo> app =
am.getRunningAppProcesses();
    String[][] map =new String[2][app.size()];

    for (int j = 0; j <app.size(); j++) {

        name=app.get(j).processName;
        ApplicationInfo a1 = null;
        try {

            a1 = pm.getApplicationInfo(name, 0);
        } catch (NameNotFoundException e) {
            // TODO Auto-generated catch block
        }

        if(name!=null && name !="" && a1!=null){
            map[0][j] =String.valueOf(pm.getApplicationLabel(a1));
            map[1][j] = app.get(j).uid+"";
        }
    }

    return map;
}
```

after that, I return this activity function and check the network is connected or not by following:

```
ConnectivityManager mConnectivity = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
TelephonyManager mTelephony = (TelephonyManager)
getBaseContext().getSystemService(Context.TELEPHONY_SERVICE);
if (mConnectivity!=null) {
```

and it will get this application web data usage by its uid by using the function:

```
TrafficStats. getUidTxBytes(uid);
TrafficStats. getUidRxBytes(uid);
```

But some of the smartphone doesn't support this function, so we need to get the file data which in the mobile file.

```
public long getusagesent(Integer uid){
    File dir = new File("/proc/uid_stat/");
    String[] children = dir.list();
    String textSent = "0";
    if(children!=null){
        if(!Arrays.asList(children).contains(String.valueOf(uid))){
            return 0L;
        }
        File uidFileDir = new File("/proc/uid_stat/"+uid);
        File uidActualFileSent = new File(uidFileDir,"tcp_snd");

        try {
            BufferedReader brSent = new BufferedReader(new
            FileReader(uidActualFileSent));
            String sentLine;
            if ((sentLine = brSent.readLine()) != null) {
                textSent = sentLine;
            }
        }
        catch (IOException e) {
        }
    }
    return Long.valueOf(textSent).longValue();
}
```

If the TrafficStats. getUidTxBytes(uid) Return less or equal to zero. Our application will check web data usage by get the file data.

Since we only can get one application total sent or received data usage on those function. So i need to check it every second. I use current data usage minus start data usage then if current is 3G, then minus WIFI total data usage and last data usage store in 3G total usage. Else use current data usage minus start data usage, then minus 3G total data usage and last data usage store in WIFI total usage. It show as following:

```
tup = gup - startgup +
DatabaseHelper.getInstance(RankingService.this).getThreeUseupintime(appname,
String.valueOf(currenttimemillis));

tdown = gdown - startgdown +
DatabaseHelper.getInstance(RankingService.this).getThreeUsedownintime(appname,
String.valueOf(currenttimemillis));
```

Then update the data in database.

Show up the 3G/ Wi-Fi data usage function. I was using pie chart and list view to show the information. I do get the Wi-Fi and 3G data usage, then check if there are no Wi-Fi and 3G data use before. If can show a different pie chart. For showing the list

of all application that has using web data. I have change the unit of the data for check which unit is most fix on it. First get the data of each application, then try to divide it by KB,MB,GB then get the unit, it show as follow :

```
private ArrayList<HashMap<String, Object>> getDatalistthree(){
    ArrayList<HashMap<String, Object>> datalist = new
    ArrayList<HashMap<String, Object>>();
    Cursor crlist =
    DatabaseHelper.getInstance(getActivity()).fetch4threeg();
    if(crlist.getCount()>0){
        crlist.moveToFirst();
        for(int j=0; j<crlist.getCount();j++ ){
            if(Long.valueOf(crlist.getString(2))>0){
                HashMap<String, Object> tempdatalist = new
                HashMap<String, Object>();
                tempdatalist.put("applabel", crlist.getString(1));
                long ununit = Long.valueOf(crlist.getString(2));
                if(ununit>0){
                    long unit;
                    String stringunit;
                    if(ununit<=0){
                        unit = 0;
                        stringunit = " B";
                    }
                    else{
                        if(ununit/GIGA>=1){
                            unit = ununit/GIGA;
                            stringunit = " GB";
                        }
                        else if(ununit/MEGA>=1){
                            unit = ununit/MEGA;
                            stringunit = " MB";
                        }
                        else if(ununit/KILO>=1){
                            unit = ununit/KILO;
                            stringunit = " KB";
                        }
                        else{
                            unit = ununit;
                            stringunit = " B";
                        }
                    }
                }
                String united = String.valueOf(unit)+stringunit;
                tempdatalist.put("totaluse", united);
                datalist.add(tempdatalist);
            }
            crlist.moveToNext();
        }
    }
    return datalist;
}
```

I have set an adapter to listen the data is the click, then it would intent to an activity to show more detail such as Wi-Fi sent and received usage; 3G sent, and received usage and total sent and received usage. In the activity have a button it can check more detail that showed the last seven days web data usage. In this activity, I have been using a bar chart and list view to show these seven days data. First, I get these last seven days data and store in a two-dimensional string array, and then add those data in some category series which for Wi-Fi data, 3G data and total data. I have created a button to let the bar chat can show the data by user choice:

```
if(onchoose==1){
    serieswifiup = new CategorySeries("Wifi Upload");
    for (int i = 0; i < dcount; i++) {
        serieswifiup.add(dateymmdd[i], wifiupy[i]);
    }
}
```

```

    }

    serieswifidown = new CategorySeries("Wifi Download");
    for (int i = 0; i < dcount; i++) {
        serieswifidown.add(dateymmdd[i], wifidowny[i]);
    }

```

Then add the category series to a series dataset and put the series database to series renderer, the set the series renderer. One series renderer can have more than one series dataset. Series dataset is stored on set of data, such as Wi-Fi up or Wi-Fi down, the implement as follows:

```

XYMultipleSeriesDataset dataset = new XYMultipleSeriesDataset();
if (onchoose == 1) {
    dataset.addSeries(serieswifiup.toXYSeries());
    dataset.addSeries(serieswifidown.toXYSeries());
}

```

```

XYMultipleSeriesRenderer rendererBarDatasetbar = new
XYMultipleSeriesRenderer();

```

```

ChartFactory.getBarChartView(this, dataset, rendererBarDatasetbar,
Type.DEFAULT);

```

5.3.1.2 Reminder function

For the Reminder function, first get the activity application. Check is there any rule on the activity application. If there have activity application, get the last running time by follow:

```

//get the last runtime
if (dh.check5byuid(u1)) {
    Cursor abc = dh.get5runtime(u1);
    if (abc.moveToLast()) {
        uidruntime = abc.getString(1);
    }
}
//if no
last runtime, Initial
if (uidruntime == null || uidruntime == "") {
    uidruntime = "0";
}

```

then if it just change the application, update the last application runtime by following:

```

//if not the same application
if (!tempuid.equals(u1)) {

    uidendtime = System.currentTimeMillis();
    Log.d("====>", String.valueOf((uidstarttime)/1000));
    Log.d("====>", String.valueOf((uidendtime)/1000));
}

```

```

        dh.update5runtime(tempuid, String.valueOf(((uidendtime-
uidstarttime)/1000)+Long.valueOf(uidruntime)));

        uidstarttime = System.currentTimeMillis();

        tempuid = u1;

    }

```

then get the current run time and the calculate the end warning time as follow:

```

nowuidruntime = Long.valueOf(uidruntime)+((System.currentTimeMillis()-
uidstarttime)/1000);

= dh.get6byrule(remcr.getString(4));

    if(rulecr.moveToFirst()){

        if(rulecr.getString(5)!=null){

            endfuctime = Long.valueOf(rulecr.getString(5))*60;

        }else if(rulecr.getString(4)!=null && rulecr.getString(5)==null){

            endfuctime = 0;

        }else if(rulecr.getString(3)!=null && rulecr.getString(5)==null
&&rulecr.getString(4)==null){

            endfuctime = Long.valueOf(rulecr.getString(3))*60+Long.valueOf(rulecr.getString(2))*60;

        }else if(rulecr.getString(5)==null && rulecr.getString(4)==null &&
rulecr.getString(3)==null){

            endfuctime = Long.valueOf(rulecr.getString(2))*60;

```

If the application runs time not over the end warning time, then check is it the warning time and update the data, it shows as following:

```

    if( !uidcheck.equals(tempuid) &&endfuctime < nowuidruntime && endfuctime!=0){

        dialog(2);

        uidcheck = u1;
    }else{

        uidcheck = u1;

    }

    if(endfuctime >= nowuidruntime || endfuctime ==0 ){

        if(endfuctime == nowuidruntime&&endfuctime!=0){

            atendtime=1;

        }

        if(!u2.equals(u1)){

            startcount =
String.valueOf(Integer.parseInt(rulecr.getString(2))*60);

            if(rulecr.getString(3)!=null&&rulecr.getString(3)!=""){

                secondcount =
String.valueOf(Integer.parseInt(rulecr.getString(3))*60);

            }else{

                secondcount = "0";

```

```

    }

    if(rulecr.getString(4)!=null&&rulecr.getString(4)!="") {
        everycount
String.valueOf(Integer.parseInt(rulecr.getString(4))*60);
    }else{
        everycount = "0";
    }

    if(rulecr.getString(5)!=null&&rulecr.getString(5)!="") {
        endcount
String.valueOf(Integer.parseInt(rulecr.getString(5))*60);
    }else{
        endcount = "0";
    }

    noticetype = Integer.parseInt(rulecr.getString(6));
    u2=u1;
}
}

```

If the application is not in the reminder rule data or the screen is off and the last activity application is not null, update last activity application run time and set last activity application is null, it shows as following:

```

else if(tempuid!="") {
    uidendtime = System.currentTimeMillis();

    dh.update5runtime(tempuid, String.valueOf(nowuidruntime));
    tempuid="";
    nowuidruntime = 0;
    uidstarttime = System.currentTimeMillis();
}
}

```

To implement this function on the interface, the data can show in two ways. One way is shown by application; it will list all the application, which had to apply on reminder. The other way showed the data by rule; user can see the rule detail on this page that included the reminder time, and those applications apply to this rule.

To add a reminder, I chose to use a dialog to help user to input data; such as text dialog, input dialog, radio and date dialog, we get the date that user input then puts it in the edit text helping user add a rule. The following is one of the text dialogs:

```

public void showtextdays() {
    final EditText editTextbyDay = new EditText(addrule.this);
    editTextbyDay.setInputType(InputType.TYPE_CLASS_NUMBER);
    AlertDialog.Builder dialog = new
AlertDialog.Builder(addrule.this);
    dialog.setTitle("Setting");
    dialog.setMessage("Please input how many days reset the data
once: ");
    dialog.setView(editTextbyDay);
    dialog.setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
        @Override

```



```

        public void onClick(DialogInterface dialog, int
which) {
        // TODO Auto-generated method stub

        if(editTextbyDay.getText().toString().length()>0){

            if(editTextbyDay.getText().toString()!=null&&editTextbyDay.getText().toStri
ng()!=""){

                long
starttime=System.currentTimeMillis();
                Calendar c=Calendar.getInstance();
                c.setTimeInMillis(starttime);
                int daytem = c.get(Calendar.DAY_OF_MONTH);

                c.set(Calendar.DAY_OF_MONTH,
daytem+Integer.parseInt(editTextbyDay.getText().toString()));
                long resettime =c.getTimeInMillis();

                System.out.println(DateFormat.format("EEEE, MMMM d, yyyy ", new
Date(starttime)));

                System.out.println(DateFormat.format("EEEE, MMMM d, yyyy ", new
Date(resettime)));

                dh.Insert7rule("rule", "days",
editTextbyDay.getText().toString(), String.valueOf(starttime),
String.valueOf(resettime),thisruleno);
                dropdown.setSelection(0);

                countevery.setText(editTextbyDay.getText().toString());
                //showTextStartTimeDialog();
            }
        }
    });
    dialog.show();
}

```

To choose the application to the reminder rule. First, I get all the install application that user installed and show in a list. If the user choices one application on the list, this application will have the rule number as temp, when the user submitted this reminder rule. All the application which rule number is temp will change to the current rule number, else all the applications, which rule number is temp will delete.

5.3.1.3 Reset function

For the Reset function, Delete App Usage data and Delete web data usage data is similar. First, we need to check it there any reset rule in the data and if there have the reset app usage type or reset web data usage type existed, it will get the current time and compare with the reset time, if on the same day, delete data, then update the reset time:

```

        long resettime=Long.valueOf(crresetrank.getString(5));
        Calendar c=Calendar.getInstance();
        c.setTimeInMillis(resettime);

        if(curc.get(Calendar.YEAR)==c.get(Calendar.YEAR)&&curc.get(Calendar.MONTH)==c.g
et(Calendar.MONTH)&&curc.get(Calendar.DAY_OF_MONTH)==c.get(Calendar.DAY_OF_MONT
H)){
            long starttime = resettime;
            if(crresetrank.getString(2)=="days"){
                int daytem = c.get(Calendar.DAY_OF_MONTH);
                c.set(Calendar.DAY_OF_MONTH,
daytem+Integer.parseInt(crresetrank.getString(3)));
                resettime =c.getTimeInMillis();
                dh.update7startresetrank("days",String.valueOf(starttime),
String.valueOf(resettime));
            }else if(crresetrank.getString(2)=="weeks"){

```

```

        int daytem = c.get(Calendar.WEEK_OF_MONTH);
        c.set(Calendar.WEEK_OF_MONTH,
            daytem+Integer.parseInt(crresetrank.getString(3)));
        resettime =c.getTimeInMillis();
        dh.update7startresetrank("weeks",String.valueOf(starttime),
            String.valueOf(resettime));
    }else if(crresetrank.getString(2)=="months"){
        int daytem = c.get(Calendar.MONTH);
        c.set(Calendar.MONTH,
            daytem+Integer.parseInt(crresetrank.getString(3)));
        resettime =c.getTimeInMillis();
        dh.update7startresetrank("months",String.valueOf(starttime),
            String.valueOf(resettime));
    }
}

```

For reset the reminder, the different is to check all the type that is “rule” and search every rule’s reset time. If in the reset time, recount the rule application open time and update the reset time.

5.3.1.4 Opentimes additional function

For the function to calculate, the open time is based on the app usage time function, when the function got the using application just change, then the function will insert or update the open times.

5.3.2Difficulty

In the project, one of the difficulties is there may have a different method to show some data in different android version. It is hard to test all the mobile phone can be successful at show some data. Such as my smartphone, my smartphone is galaxy note 2, some of the galaxy note two users can get the web data usage in TrafficStats functions, but it is not worked on my phone, but my phone android version is over level 12 which the android developers website show it is support.

Second difficulty is we didn’t know any information on building an application at the beginning. So we use a long time one it.

Third difficulty is about web data usage count, I find that those function or file data are started on every time the phone reboot and the data not divide by Wi-Fi or 3G, show I need to think a way to count the data start after the application installed, get Wi-Fi and 3G data by those data and not reset the data after reboot the smartphone.

Finally, difficulty is the interface problem when the smartphone rotate, I try to create a scroll to solve that, but the result not so good. Finally, I create a different layout for the rotate.

CHAPTER 6. EVALUATION

6.1 Testing

Since my partner and me also using android phone, show we would use our phone to evaluation. And we find some older version phone to test our application. It also work on those phone. And our supervisor suggests us upload our application to play store for the user to test it. We have get some error report form the user who download our application, Play Store have the error report function help us to find our problem in our project. It also can get some suggestions form the user in anywhere. it help us so much.

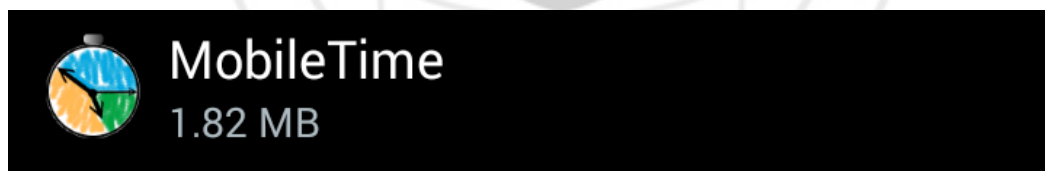
Since there have different kind of phone are using android system. But my partner and me also use Samsung, so we need to find some users who are using other king of smartphone such as SONY, LG etc.

6.2 Comparison

Since we find there is less application that is about counting the app usage. We have find a application it named “App Timer Mini 2”. This application about ranking the app usage.

Since our application have store the data in a long time, but the size of our application on 1.82MB (see Figure 11.)

Figure 11: manage your mobile time application size



For “App Timer Mini 2”, since I didn’t set any data, the size of the application already 2.29MB (see Figure 12.)

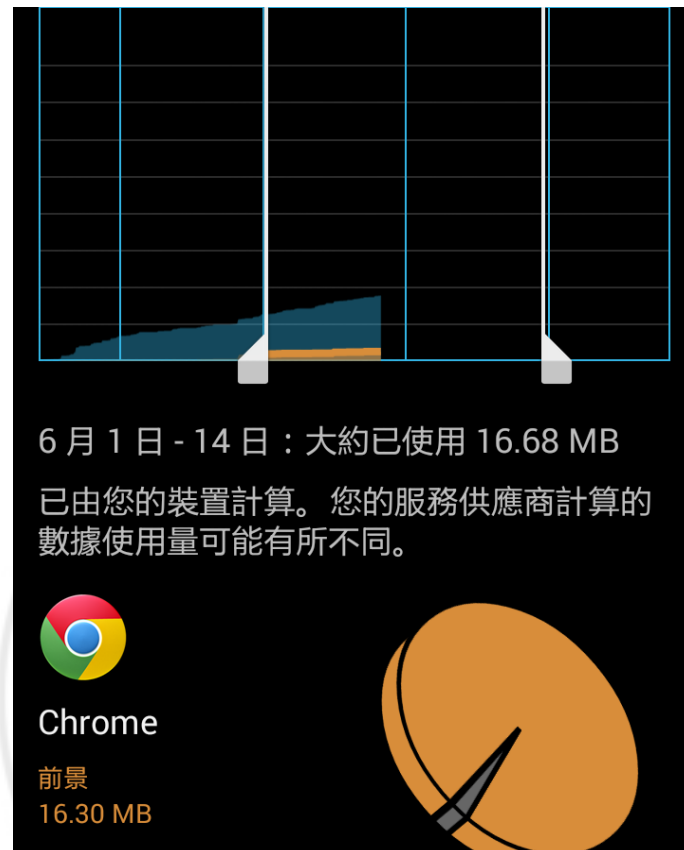
Figure 12: App Timer Mini 2 application size



For “App Timer Mini 2”, it only have the function of ranking, but our application not only have ranking app usage function, but also web data usage, chcek open time and low battery wraning.

For the Data usage function, although android system also have the function to see the data usage. But the data usage on android system only for 3G data usage and it can see the detail usage of the application, such as upload and download usage(see Figure 13.).

Figure 13: Android System web data count funtion



CHAPTER 7. SUMMARY

7.1 Future work

We already developed an application which can count the application usage, count the web data usage and reminder function of application usage. However, our application only had an English version. For the future work, we would like to develop the application usage time that can count with actual usage. For example, the application can show the detail time that the user used the Web browser to watch movie, search information or entertainment, etc. Another example, such as "WeChat," the application can verify that user is communicating with his friend or watching the social network information. We have tried to find the solution of this function on the Internet and books, but we fail to find the solution. So on the future work, we would like to find out the solution. Secondly, we would like to create more Language versions of our application.

7.2 Conclusion

Smartphone is a great invention of the world, but it also brings some harm when people are too dependent on Smartphone, when people use their phone in a long time, the user may get anxiety, depression, withdrawal, delinquent behavior and difficulty maintaining attention and concentration. Our application is provided to people of all ages who wanted to away these harms. So we start to design an application that can help people to solve this problem.

The overall time of this final year project is quite hurriedly. At the beginning, we didn't know any knowledge on develops an application. So we need to do a lot of research to find which mobile operating system is suitable for our application. In this short time, we are successful in creating an application. The application has the basic function to manage the mobile usage time. However, if we have more time, we would like to improve our application to more function, such as create a classification to all user-installed applications and other additional function that can help to manage the mobile usage time.

In this final year project, one of the difficulties is that there have many kinds of Smartphone are using an android operate system in different versions. Different version may support a function in a different way, such as data usage, in android official website; it shows that the function which can get the total data usage an application only supports the android version which is higher than version 3.1, if the android version is less than 3.1, we should implement it in another way.

After this project, we have been successful in the development a mobile application that can manage the usage time. We have joined the PPRD Project Competition of Computer Science, and the judges agree with our project. We are successful at implementing an application that can run on most of the Android phone. We have found the idea of develop an application and understanding the technique in develop the android operating systems. In summary, we already achieved the original goal of this final year project. During developing the application, Dr. Zhuang Yan, Sofia

helped us to think more ideas and give us some comment; it helps us to find out our deficiency.



CHAPTER 8. REFERENCES

[1] Mail Online, 'To have and to hold: We now spend more time looking at our PHONE than with our partner'. [Online]. Available: <http://www.dailymail.co.uk/sciencetech/article-2333261/We-spend-MORE-time-phones-partner.html>

[2] Android Developers, 'TrafficStats'. [Online]. Available: [http://developer.android.com/reference/android/net/TrafficStats.html#getIdUdpTxBytes\(int\)](http://developer.android.com/reference/android/net/TrafficStats.html#getIdUdpTxBytes(int))

[3] SQLite, "SQLite Home Page", [Online], Available: <http://www.sqlite.org/>

[4] Ao leong Kit Ha, my partner

