

# Bachelor thesis

*Major Subject*

**Exercise Monitor on E-healthcare System**

**Xiaojie Liang**  
**xili1100@student.miun.se**



**Mittuniversitetet**  
MID SWEDEN UNIVERSITY

## Abstract

With the rapid development of terminal hardware, mobile users and network environments, the scale of the mobile Internet appears to be catching up with the desktop Internet. Positioning System, phone sensors and applications which are able to be used anywhere have caused the arrival and increase in usage of the smart phone, and the world has become mobile-first. Within medical fields, the smart phone is also regarded as an acceptable assistant for nurses and patients as based on the advantages of mobile Internet. It is possible for healthcare personnel to assign tasks for the patients from a different place if there is a computer connected to the Internet. In addition, the smart phones is able to act as a nurse and remind the patients to complete their tasks. Accordingly, it has become a necessity to achieve this electronic healthcare system for hospitals. The objective of this research project is to summarize the procedures of this development, and release the relevant software based on an Android platform. Additionally, a corresponding website is to be designed so that healthcare personnel are able to sign in in order to deal with the tasks. The entire system has been tested in real scenarios, at this point only by the author but has still not been used by other organizations. The final part of the report involves the conclusions drawn and provides suggestions for further work about the whole project.

**Keywords:** mobile Internet, Android platform, electronic healthcare system.

## Acknowledgements

Many thanks to my tutor Professor Zhang under whose guidance, I was able to complete my final theme project. Particular thanks must go to her for allowing me to use a smart phone from university which made the development more efficient than it would have been if using an emulator. I would also like to offer my deep appreciation to Mr. Guo for cooperating with me throughout the whole project and finally enabling me to achieve it.

# Table of Contents

<b>Abstract.....</b>	<b>ii</b>
<b>Acknowledgements.....</b>	<b>iii</b>
<b>Terminology.....</b>	<b>vi</b>
<b>1 Introduction.....</b>	<b>1</b>
1.1 Background and problem motivation.....	1
1.2 Overall aim.....	2
1.3 Scope.....	2
1.4 Concrete and verifiable goals.....	2
1.5 Outline.....	3
1.6 Contributions.....	3
<b>2 Theory.....</b>	<b>5</b>
2.1 Mental illness and exercise.....	5
2.2 Privacy policy.....	5
2.3 Introduction Android 4.0.....	6
2.4 Struts/Hibernate/Spring.....	6
<b>3 Methodology.....</b>	<b>8</b>
3.1 Phone application model.....	8
3.2 Interactions between the participants.....	9
3.3 Observer mechanisms of the phone application.....	9
3.4 Java beans model.....	10
<b>4 Design.....</b>	<b>11</b>
4.1 Overall framework design.....	11
4.2 Phone client program design.....	11
4.2.1 GPS service implementation.....	12
4.2.2 Calendar service implementation.....	15
4.2.3 Pedometer service and TextToSpeech service implementation.....	17
4.3 Alarm design.....	18
4.4 Upload file design.....	21
4.5 Web client program design.....	22
4.5.1 Account Manager.....	23
4.5.2 Validation before query.....	25
4.5.3 Calendar.....	26
4.5.4 Synchronization.....	28
4.5.5 File Receiver.....	29
4.5.6 Consumption Display.....	29
<b>5 Results.....</b>	<b>31</b>
5.1 GPS calories collector.....	31
5.2 Upload file.....	32
5.3 Calendar.....	33
<b>6 Conclusions.....</b>	<b>34</b>

**References.....35**  
**Appendix A: Documentation of own developed program code.....36**  
    See the attachment “MyWeb\_US.war” and “CalendarOps\_US.rar”.. .36

# Terminology

## Abbreviations

API	Application Programming Interface
JSP	Java Server Pages
XML	Extensible Markup Language
GPS	Global Positioning System
MVC	Model-View-Controller
LAN	Local Area Network

## Mathematical notation

$\cos$	the trigonometric cosine of an angle
$\sin$	the trigonometric sine of an angle

# 1 Introduction

## 1.1 Background and problem motivation

In the past, mobile phones have only played a role as a call or message transceiver. However, with the rapid development of the mobile Internet, another popular mobile phone, which is called the smart phone, could also be used as a reminder, a music player or a game player. In relation to medical fields, a smart phone could be used as a reminder not only to be a time alarm, but to also be more intelligent. Based on clinical trials in hospitals, it is known that it is an effective measure in curing some mental disease if the patient maintains a daily exercise routine. In addition, it is also the case that some patients are keen to bring a phone because it is a rather fancy accessory. Consequently, a smart phone could take the place of a healthcare person to remind the patients to exercise according to the doctor's advice. The aim of this project is to offer the significant advantage of liberating healthcare personnel from their busy, repetitive schedules and the hope to relevant application for an e-healthcare system.

Nowadays, the smart phone has become increasingly popular and the three smart phone platforms available in the market are the iphone, android and the windows phone. Based on the open source platform of android, it is relatively easy to acquire the data from different phone sensors about the accelerated speed and different directions. In addition, it is also possible to obtain accurate geographic locations by means of satellites signals. Large amounts of data are also able to assist in the calculation of the energy consumption of an owner of such a phone over a certain period of time. According to the record of the calories consumed, the phone is able to call an intelligent alarm to remind the patient that the energy consumption did not meet the requirements set by their doctor. On the other hand, it is also possible for the doctor to specify a detailed schedule for his/her patients by signing in to a certain website by means of a computer. This then means that the calendar events will be synchronized to Android phone belonging to the same Google account.

## 1.2 Overall aim

The project's overall aim is to achieve an intelligent design for a smart phone reminder and a location separation of the participants. Two people in one group are involved in implementing this theme project. The author of this project will implement an intelligent reminder and collect the location data by means of a GPS sensor. The work of the second member of the group is to collect large amounts of data from the phone's gravity sensor and direction sensor, and then to analyze these data so as to draw a conclusion regarding the number of calories that have been consumed within a certain period of time. At the end, the final part of this work will involve the integration of the two parts into one complete application. On the healthcare person's side, a website should be designed to enable them to deal with the calendar events and to be able to look at the daily exercise report.

## 1.3 Scope

The focus of the project is on the realization of an intelligent reminder in relation to calendar events. Observations based on the results of a practical test, have shown that the reminder is distinguished by its strong applicability and high sensitivity. However, for more accurate data with regards to calories consumption, not only has a pedometer been chosen as an energy collector, but in addition, the energy collector has been improved by using GPS. In the future, it might be possible for this energy collector to be replaced by a more precise instrument. At the beginning of this project, Nike shoes were chosen to record the steps as they were found to be the most accurate. However, it was discovered that the chips in these shoes are only available for the iphone and the system for this report is based on Android, which is widely used worldwide and could thus be cheaper than the iphone. It was, however, felt that the conclusions drawn in relation to the energy collector for this project should be generally valid for a number of cases.

## 1.4 Concrete and verifiable goals

On the patient's side, the objective of the project are to respond to the following questions:

P1: How should the energy collector be implemented?



--There are two modes to collect the data, namely, pedometer mode and GPS mode. The application could use one or both of them to improve the accuracy of the energy data.

P2: How will the intelligence of the reminder be shown?

-- There are two modes to remind the patient to complete the task. The reminder will choose one of them according to historical energy consumption.

P3: How will the interaction between healthcare person and the patient be implemented?

--Through the Google Calendar, any changes of the calendar would be synchronized to the phone which belongs to the same Google account.

On the healthcare person's side, the objectives for the project are to respond to the following questions:

P1: How will the daily energy consumption be shown for the healthcare person?

-- In a moment, which has been set to be the upload time, the server will receive the energy records file. The healthcare person could sign in and see these files.

## 1.5 Outline

Chapter 2 describes the relevant theories including GPS, Google Calendar, smart phone sensor and a simple introduction to the website. Chapter 3 describes the methods used to solve the problems based on the Android platform. Chapter 4 describes the relative designs of client-side design and the website design. Chapter 5 describes the processes involved in testing the different modules. Chapter 6 gives the conclusion about the whole project.

## 1.6 Contributions

As mentioned previously, two people are involved in implementing this theme project. The main work of the author is to design the intelligent reminder, GPS data collector and healthcare person website while the second person's work is to design the pedometer so as to collect data for the energy calculation. The two parts are finally integrated

in order to provide a exercise monitor for the complete e-healthcare system.

In order to improve the appearance of the web page, a search was made for attractive and simple web pages online. From the Google official site [\[1\]](#), the Cascading Style Sheets (.css file) were downloaded and they were used to become part of the web page. Furthermore, FullCalendar[\[2\]](#) is one of the well-known calendar jQuery plugins that assists developers to quickly build calendar functionality in the foreground. With the assistance of FullCalendar, it is possible to create a powerful calendar such as the Google Calendar.

## 2 Theory

### 2.1 Mental illness and exercise

Walking and running have positive effects on the body, spirit and mind, especially on the way home from the work place. "Activity and exercise are very important for people living with mental illness. Individuals living with mental illness often have a higher risk for heart disease, and exercise can play a key part in a wellness plan. Activity and exercise are great ways to combat factors that are part of heart disease risk, stress, high blood pressure, weight gain and diabetes--all problems commonly found among people living with mental illness. Exercise plays a key part in elevating your mood and regulating sleep patterns." [3] Relevant researchers have demonstrated that exercise possesses antidepressant properties. Other research has shown that exercise can improve the functions of the brain. In the official website of "National Centre for Biotechnology Information", a paper is included to state "the relation of physical activity and exercise to mental health" [4]. It begins with "Mental disorders are of major public health significance. It has been claimed that vigorous physical activity has positive effects on mental health in both clinical and nonclinical populations. This paper reviews the evidence for this claim and provides recommendations for future studies." [4]

### 2.2 Privacy policy

Consider the features of this system; the system will inevitably interact with the user directly or indirectly. The following description states the measures taken by the users' personal information collection, use and protection policies, and these should be read carefully:

(1) Only the user can permit a healthcare person to monitor his/her movement, otherwise this phone application will not be installed in the user's phone.

(2) After the user agrees to install and run the phone application, the phone does not record any location information, but only records the calories consumption.

(3) Without the agreement of the user, the phone application will not upload any calories record files automatically.

(4) All the files uploaded to the website will only be visible to the specific healthcare person. The site will not disclose any personal information about users.

## 2.3 Introduction Android 4.0

“Android 4.0 (Ice Cream Sandwich) is the latest version of the Android platform for phones, tablets, and more. It builds on the things people love most about Android — easy multitasking, rich notifications, customizable home screens, resizable widgets, and deep interactivity — and adds powerful new ways of communicating and sharing.” [5] This new application will be developed based on the Android 4.0. But it should still be able to run on some Android phones whose version is less than Android 4.0.

“To help organize appointments and events, an updated Calendar app brings together personal, work, school, and social agendas. With user permission, other applications can contribute events to the calendar and manage reminders, for an integrated view across multiple calendar providers. The app is redesigned to let you manage events more easily. Calendars are color-coded and you can swipe left or right to change dates and pinch to zoom in or out agendas.” [5] Based on Google Calendar's synchronization technology, the phone's calendar and the server's calendar are always consistent. This offers an easy means of maintaining a unified calendar.

## 2.4 Struts/Hibernate/Spring

There are three open source frameworks which are rather popular in J2EE web development: Struts, Hibernate and Spring. With the integration of these three frameworks, it is possible to build a robust website which also involves simple maintenance.

Generally speaking, Struts is an open source framework used to develop web applications using an MVC (Model-View-Controller) design pattern based on Servlet, JSP and Java Beans. “The Apache Struts Project offers two major versions of the Struts framework. Struts 1 is recognized as the most popular web application framework for Java. The 1.x framework is mature, well-documented, and widely supported. Struts 1 is the

best choice for teams who value proven solutions to common problems. Struts 2 was originally known as WebWork 2. After working independently for several years, the WebWork and Struts communities joined forces to create Struts 2. The 2.x framework is the best choice for teams who value elegant solutions to difficult problems.”[\[6\]](#)

Hibernate is an open source framework used as a middleware which maps from Java classes to database tables. Through the encapsulation of Hibernate, it becomes possible to visit the database tables in a similar manner to process object-oriented. The convenience of using Hibernate is that “Hibernate takes care of the mapping from Java classes to database tables, and from Java data types to SQL data types. In addition, it provides data query and retrieval facilities. It can significantly reduce development time otherwise spent with manual data handling in SQL and JDBC.”[\[7\]](#)

Spring is an open source framework consisting of many widely-used modules. “The Spring Framework helps you build Java Applications faster because it allows you to focus on your business problem rather than the plumbing code that connects components and systems.”[\[8\]](#)

## 3 Methodology

### 3.1 Phone application model

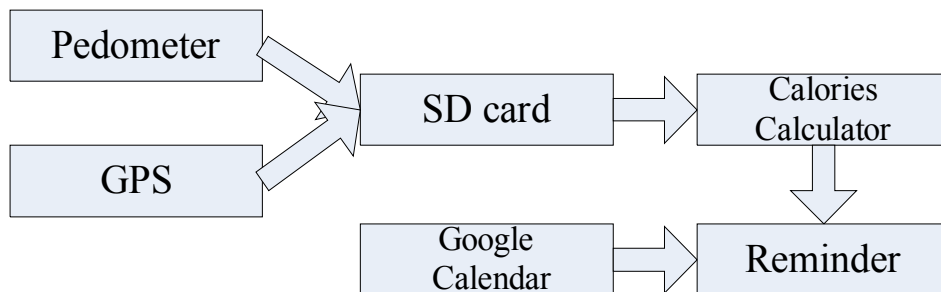


Figure 3.1 core model of the whole application

Referring to figure 3.1. This shows the core model of the whole phone application. As can be seen, a pedometer or GPS are used as the calories collectors. At any given time, only one collector is running. The collector, which is running, writes the calories consumption records into the files of the SD card during every time period. On the other hand, if there is an event starting in Google calendar, the reminder will invoke the calories calculator to assess the event.

“When developing a location-aware application for Android, you can utilize GPS and Android's Network Location Provider to acquire the user location. Although GPS is most accurate, it only works outdoors; it quickly consumes battery power, and doesn't return the location as quickly as users want.”[\[9\]](#) This is the reference from the Android Dev Guide, but it is not exact in relation to some high-quality smart phones. For example, the Samsung i9520 smart phone uses the technology of AGPS, which is the GPS with the net. When using AGPS, the phone will download the satellites' information from the Internet in advance. Then the phone will know which four satellites are within the search range, thus it is faster and less power is used than for a GPS chip which searches all the satellites which are in the search range.

### 3.2 Interactions between the participants

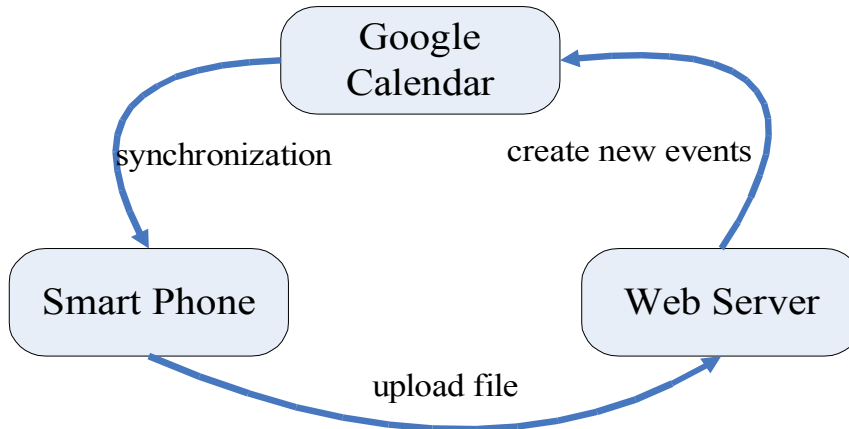


Figure 3.2 interaction model

Figure 3.2 shows how to achieve an information transmission between the participants. As mentioned previously, the objective of this project is to separate the healthcare person from the patient, and the smart phone will take the place of the healthcare person. With the powerful mobile synchronization technology of Google Service, all the changes of the calendar's events will respond to the user's smart phone at a very fast speed. Additionally, there it is necessary for the healthcare person to know about the exercise situation of his/her patients. Therefore, the smart phone application must upload the calories record file to the web server.

### 3.3 Observer mechanisms of the phone application

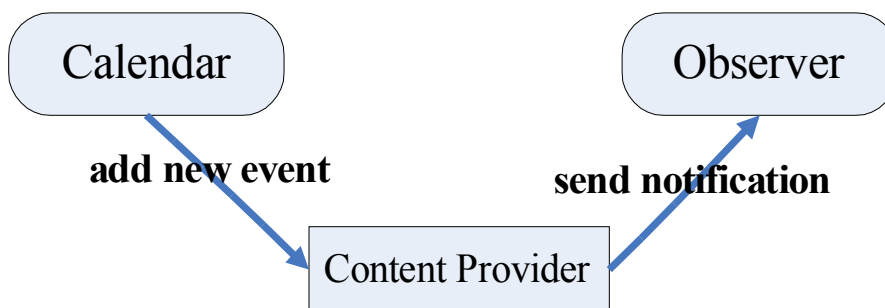


Figure 3.3 observer mechanisms

Figure 3.3 shows that the phone application is running independently and it is unable to read the data from calendar all the time, which is a waste of CPU. However, it is sometimes the case that the changes of the calendar events should respond to the application at a given time. Fortunately, based on the Android platform, an observer mechanism which is similar to a listener, could assist the phone

application to obtain a timely notification when the content provider changes. Referring to figure 3.3, when the calendar events changes, the calendar provider will send notifications to the observer who is listening for it. The type of trigger handler will assist in improving the performance of this application.

### 3.4 Java beans model

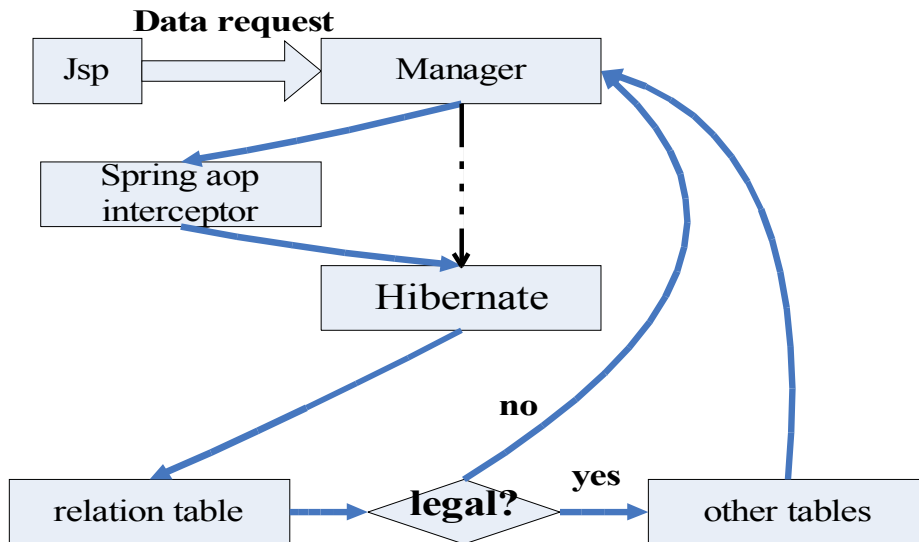


Figure 3.4 Java Beans Model

As can be seen in figure 3.4, when the Java server page requests the data from the Manager class, the Manager class attempts to obtain the data from the database through the Hibernate framework, the Spring framework interceptor will stop this behaviour and will invoke the validate method in order to check whether or not the access to the database is legal. If it is legal, then the Manager class is allowed to continue running. If it is not legal, then all access to other tables will be prohibited.



## 4 Design

### 4.1 Overall framework design

Figure 3.2 shows the basic implementation of the whole project. Supported by the instantaneous synchronization of Google Calendar, this project would establish two client programs for the phone and the web respectively. The phone application is able to read the calendar's events through a built-in Google Calendar, and the healthcare person could write the events to the Google Calendar through the web. A detailed exhibit is shown in figure 4.0 as follows:

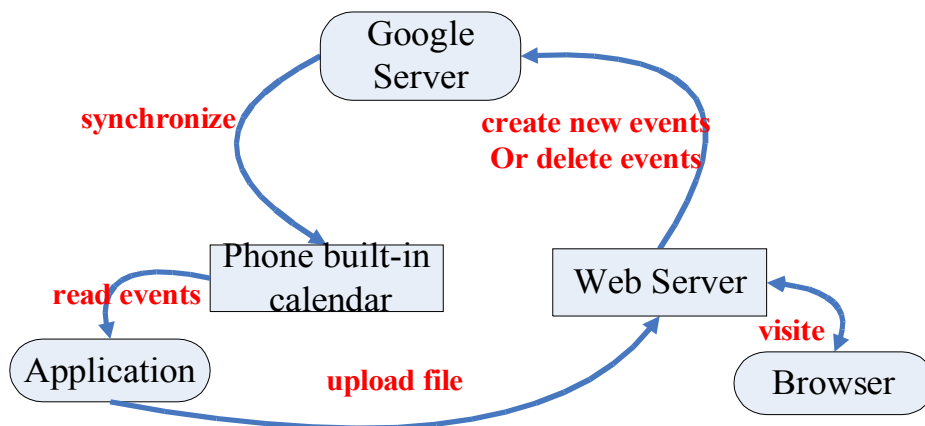


Figure 4.0 Entire System

### 4.2 Phone client program design

As mentioned above, the phone client program contains several services running in the background. Details about all the services are shown in table 4.1 as follows:

Table 4.1 Several Services

Service name	Function
GPS Service	acts as a calories collector
Pedometer Service	acts as a calories collector
Calendar Service	reads event from calendar and sets reminder
TextToSpeech Service	convents the remind text to a voice

#### 4.2.1 GPS service implementation

The calories collector determined the reminder's start-up time. In accordance with a normal life schedule, a man may wake up at 8 o'clock in the morning and go to sleep at 22 o'clock. However, from 18 o'clock to 22 o'clock, the man was always watching TV. Therefore the period from 8 o'clock to 18 o'clock is the best running time for the calories collector. Furthermore, two kinds of collectors have been designed to record the energy consumption. One being the pedometer collector which is the main part of the second person in the group's work.

The other is called the GPS collector, which actually only works outdoors. Although it may be felt that it works indoors on occasions, this is not the real GPS data. From the point of mature technology, it is possible to utilize GPS and Android's Network Location Provider to acquire the user location information. Many manufacturers have incorporated the network location provider into the GPS so that it is possible to obtain the location information even if the GPS is only used indoors. How is the GPS Service used to record the calories consumption at a certain time? See the exhibit in figure 4.2 as follows.

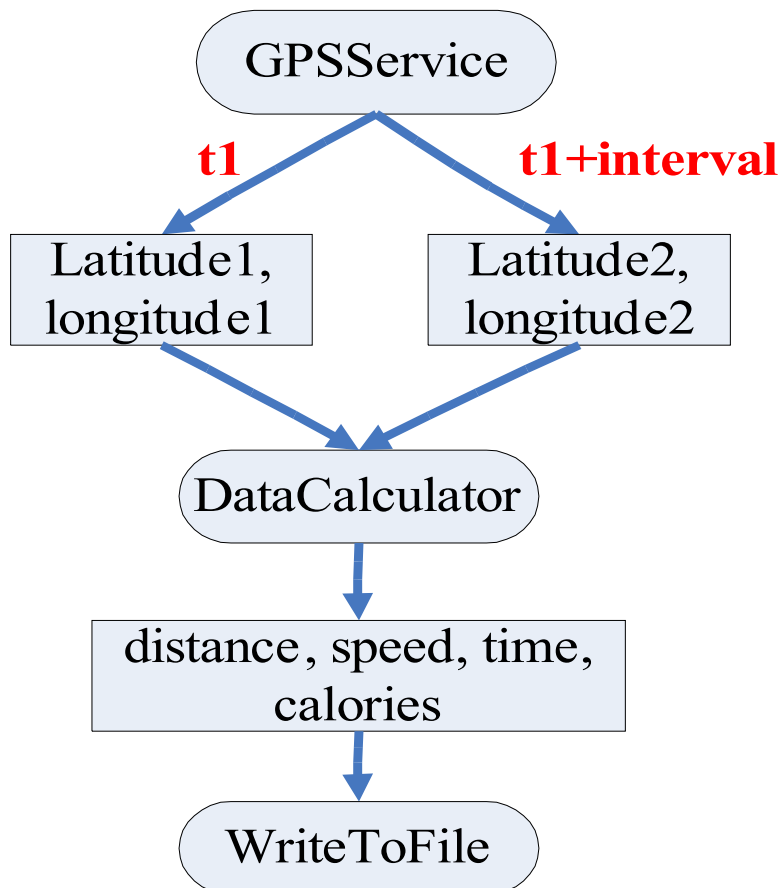


Figure 4.2 GPS Calculator

In "Google Maps Javascript API V3 Reference", it offers a utility function for computing geodesic distances between two specified locations. "By default, this distance is calculated given the default equatorial earth radius of 6378137 meters. The earth is approximated as a sphere; hence the distance could be off as much as 0.3%, especially in the polar extremes." [10] This function is required frequently and thus this API must be implemented locally.

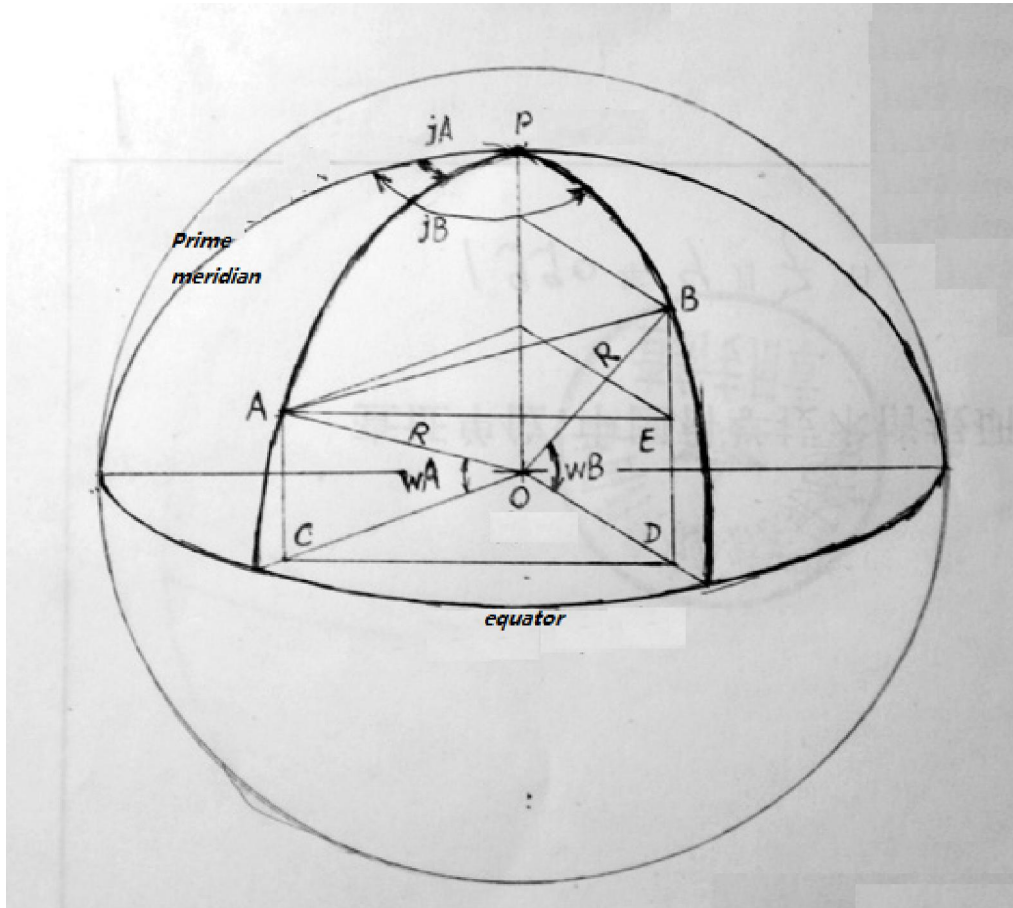


Figure 4.3 Calculate distance between two locations

Referring to figure 4.3, the arc length of AB can be calculated using Law of Cosines. Formula derived as follows:

$$A(\text{Lat1}, \text{Lng1}), B(\text{Lat2}, \text{Lng2})$$

$$wA = \text{Lat1}, wB = \text{Lat2}, jA = \text{Lng1}, jB = \text{Lng2}$$

$$AC = ED = R \cdot \sin(wA), OC = \cos(wA), BD = \sin(wB), OD = R \cdot \cos(wB)$$

$$BE = R(\sin(wB) - \sin(wA)), AB^2 = AE^2 + BE^2$$

$$(\overline{AB})^2 = R^2 + R^2 + 2 * R^2 * \cos(\angle AOB)$$

$$\widehat{AB} = \frac{\angle AOB}{180^\circ} * \pi$$

The formula is only applied to that the location with east latitude and north longitude which is fortunately suitable for a Swedish geographic location. After the value of speed has been obtained, another formula can be used to calculate the caloric expenditure during physical activity.

Energy Expenditure Formula:

$$(\text{kcal per min}) = 0.0175 * \text{METS} * \text{body weight (kg)}$$

Example: A person weighing 75kg, walking at a speed of 4mph (4METS) will use approximately  $0.0175 * 4 * 75 = 5.25$  kcal per min

METS is the abbreviation of metabolic equivalents. Different values are shown in table 4.4.

Table 4.4 METS values

METS	Type of exercise
2.56	Walking 1.88mph
2.5	Walking 3.00mph
6.0	Walking 4.00mph
9.8	Running 6.00mph
11.0	Running 7.00mph

Finally, the calories collector must record the segmented consumption into a file. The file's standard format appears as follows.

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<root>
  <energy start="xx" end="xx">xxx</energy>
  <energy start="xx" end="xx">xxx</energy>
</root>
```

#### 4.2.2 Calendar service implementation

When the program starts, the Calendar Service would read out the most recent event from the built-in Google Calendar, set up a series of intelligent alarms and respond quickly to the change in the calendar. As long as the current event starts, it is not allowed to be modified and this service will ignore other change in the calendar. After the current event ends, this service will read out the most recent event again.

The content of the event contains the title, the start time, the end time and the description. The standard of calories consumption and the reminder word will be recorded in the field of description. By default, the description is a type of string, but for correct reading, the format of the description must appears as shown in the following text:

```
<root>

  <energy>integer</energy>

  <description>text</description>

</root>
```

Base on an Android platform, this offers a loader called *CursorLoader* that queries the *ContentResolver* and returns a *Cursor*. *ContentResolver* provides access for applications to the content model. Additionally, a *Cursor* is an interface, which provides random read-write access to the result set returned by a database query. How should the *CursorLoader* be used? See its constructor in Figure 4.5, Table 4.6 and Table 4.7.

```
public CursorLoader (Context context, Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder)
```

Figure 4.5 the constructor of *CursorLoader*

Table 4.6 User's manual of query

<i>Parameters</i>	
context	A interface to global information about an application environment, get it by using "getApplicationContext()"
uri	The URI, using the content:// scheme, for the content to retrieve.
projection	A list of which columns to return. Passing null will return all columns, which is inefficient.
selection	A filter declaring which rows to return, formatted as an SQL WHERE clause (excluding the WHERE itself). Passing null will return all rows for the given URI.
selectionArgs	You may include ?s in selection, which will be replaced by the values from selectionArgs, in the order that they appear in the selection. The values will be bound as String.
sortOrder	You may include ?s in selection, which will be replaced by the values from selectionArgs, in the order that they appear in the selection. The values will be bound as String.
<i>Returns</i>	
Cursor object	which is positioned before the first entry, or null

All the parameters must be provided. It is simple to provide the parameter of "Context". However, there is a minor difference between the usages of the SDK version, which is greater than API8. The SDK versions which are less than API8 use the Uri address of "content://calendar/events" to query the events, and the other versions use the Uri address of "content://com.android.calendar/events". Details are given in the following form.

Table 4.7 the instances of parameters

<i>parameter</i>	<i>instance</i>
context	getApplicationContext()
uri	>=API8: content://com.android.calendar/events <API8: content://calendar/events
projection	new String[] { "title", "dtstart", "dtend", "description"};
selection	new String("calendar_id = 1 AND dtstart > ?");
selectionArgs	String.valueOf(System.currentTimeMillis())
sortOrder	New String("dtstart DESC, dtend DESC");

Although the program has recorded the most recent event, it is possible to modify the current event as long as the event was not started. How can the program respond to the change of the current event? Fortunately, the Android SDK API has provided a design pattern which is similar to the Listener mechanism in JVM. As suggested by its name "*Observer*", an Observer is the interface to be implemented by objects that receive notification of updates on an observable object. Furthermore, an observer object should be instantiated which extends the abstract class of *ContentObserver*. To make it work, this object must also be registered to the application environment. The real effect is that the method of "*onChange(boolean selfChange)*" is called when a change occurs to the cursor that is being observed. Inside this method, any code can be written to make judgements and then make adjustments in relation to the change.

### 4.2.3 Pedometer service and TextToSpeech service implementation

Furthermore, little is stated in this case in relation to the Pedometer Service and TextToSpeech Service implementation as this is the work for the second member of the group. The Pedometer Service also acts as a calories collector and it works both indoors and outdoors. The TextToSpeech Service could also convert the text to a speech, and the Calendar Service would then call this TextToSpeech Speech to say something to remind the user to exercise.

### 4.3 Alarm design

As mentioned above, several features of the module of “Calendar Service” have been summarized . However, the vital part of the Calendar Service is in relation to the design concerning a series of alarms. When the Calendar Service reads out a new event, the start time is given, but, the question is how to start this alarm on time and how to tell the patient the result of his exercise during the executing time. The class of “*AlarmManager*” provides access to the system alarm services, which allows the developers to schedule their application to be run at some point in the future. Hence, it is possible to set up this event’s reminder in this way. The greatest advantage is that registered alarms are retained while the device is asleep and it could also wake the device up. With this advantage, this application code could be made to run at a specific time even if that application is asleep. Details are shown in Figure 4.8.

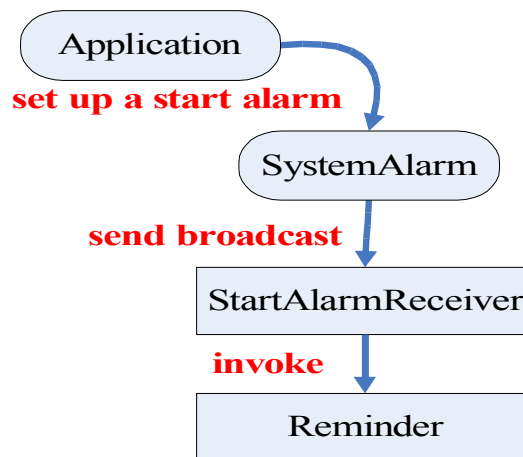


Figure 4.8 System Alarm

Referring to figure 4.8, how is it possible to implement the specified reminder when the “*StartAlarmReceiver*” starts to work? There are two modes to make this reminder become more intelligent. One is called default mode, when executing the first event, there is no previous reference thus the application choose the default mode to implement the reminder. Details about the default mode are shown as follows.



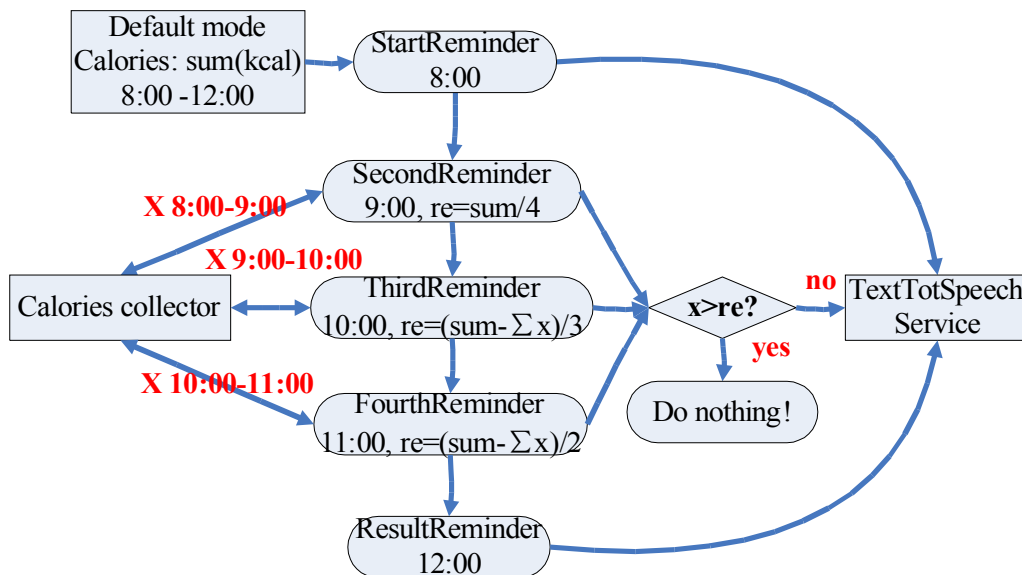


Figure 4.9 Default mode

Referring to figure 4.9, it can be seen, in default mode, that the application had built five alarms to evaluate different stages' result. The whole event has diverged into four stages according to the average interval. During the stage, if the calories consumption is less than the required consumption, the reminder will invoke the TextToSpeech Service in order for something to be said. Another reminder pattern is called habitual mode, which works depending on yesterday's calories records. Details are shown in the following.

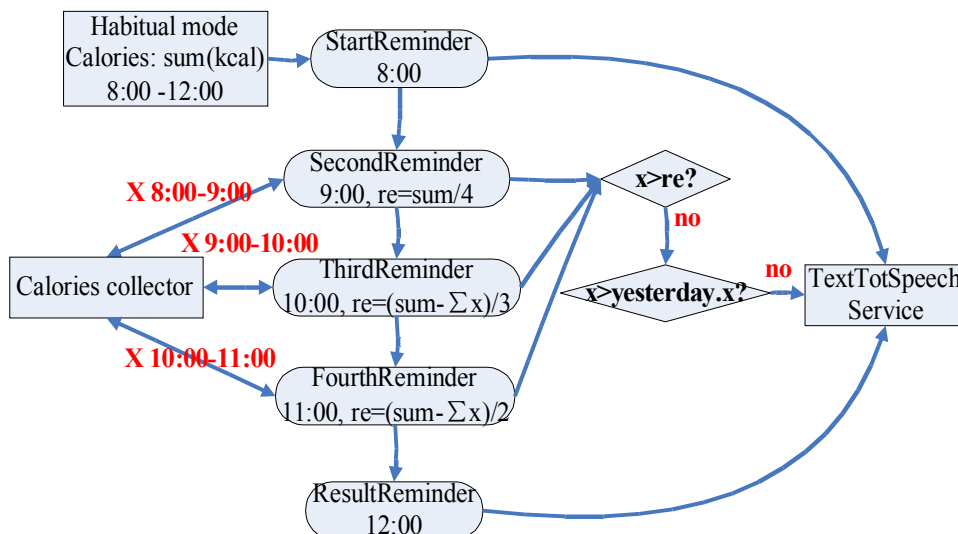


Figure 4.10 Habitual mode

Referring to figure 4.10, the difference associated with the default mode is visible. When the calories consumption is less than the required consumption at one stage, the reminder is not going to invoke the Text-

ToSpeech Service, but the reminder will continue to compare the consumption with yesterday's consumption during the same interval. Supposing that the consumption is at the minimum value, then the reminder will call the voice.

There is still a problem which is when to choose the habitual mode reminder. By default, when the event ends, the Calendar Service will write an evaluation report into a file. The format of the file has the following appearance.

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<root>
  <assignment start="1337203140000" end="1337289660000">
    <required>666.0</required>
    <actual>345.0</actual>
    <evaluate>fail</evaluate>
  </assignment>
</root>
```

When a new event starts, the reminder would firstly check whether there is an event with the same start and end time as for yesterday's evaluation report. If the historical event has reached the reference standard, then the reminder will choose the habitual mode.

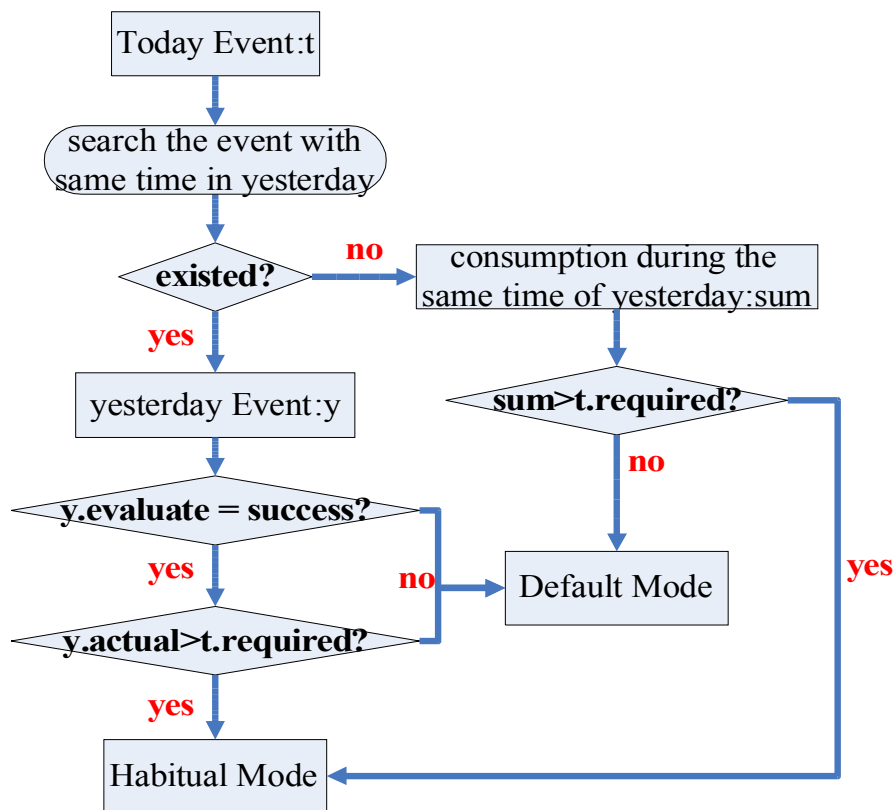


Figure 4.11 when to choose the habitual mode

## 4.4 Upload file design

The module of file upload could be implemented by means of different web technologies. The most commonly used approach is to benefit from the Apache projects. The Apache projects provide three options to realize this function. Furthermore, Struts 1 is selected as the framework as shown in the following.

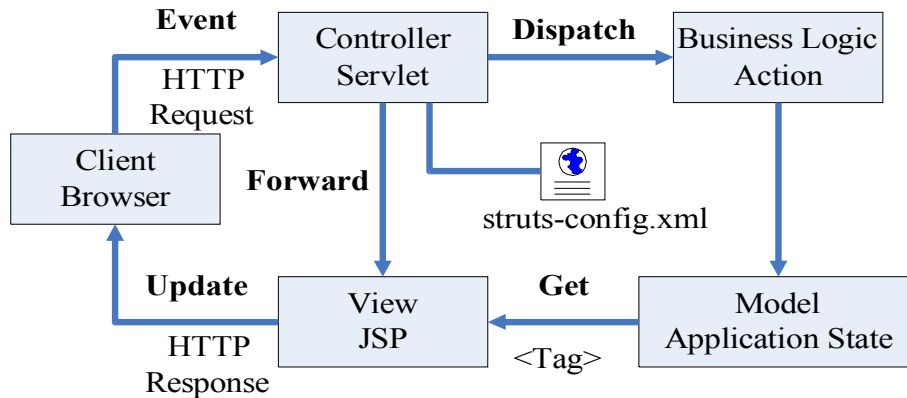


Figure 4.12 Struts flow chart

Referring to the figure 4.12, it is easy to build a Java page to receive the file as shown in figure 4.13.

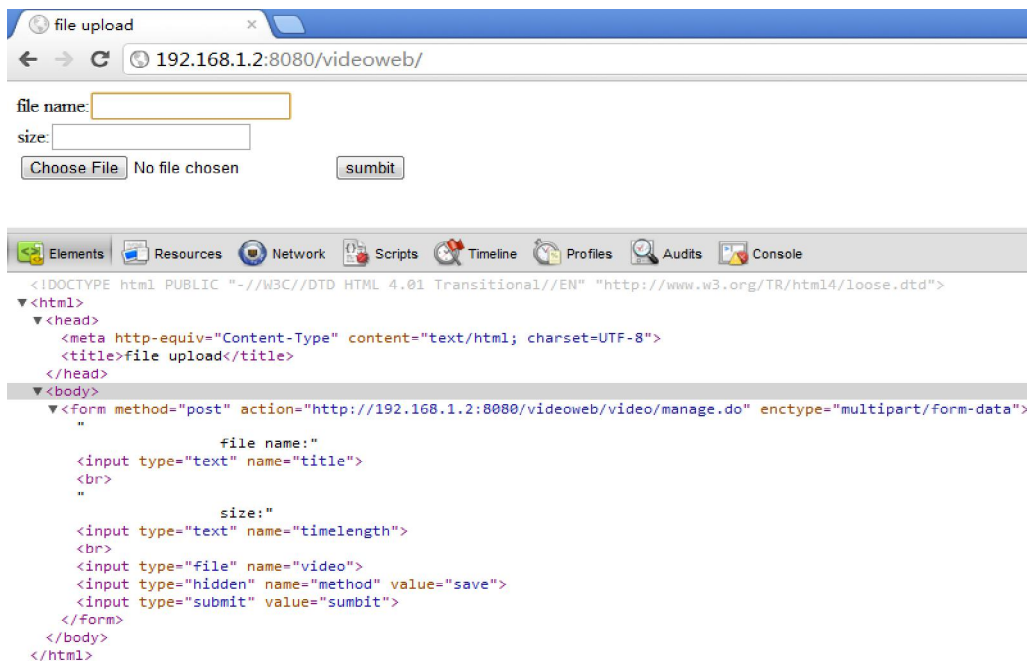
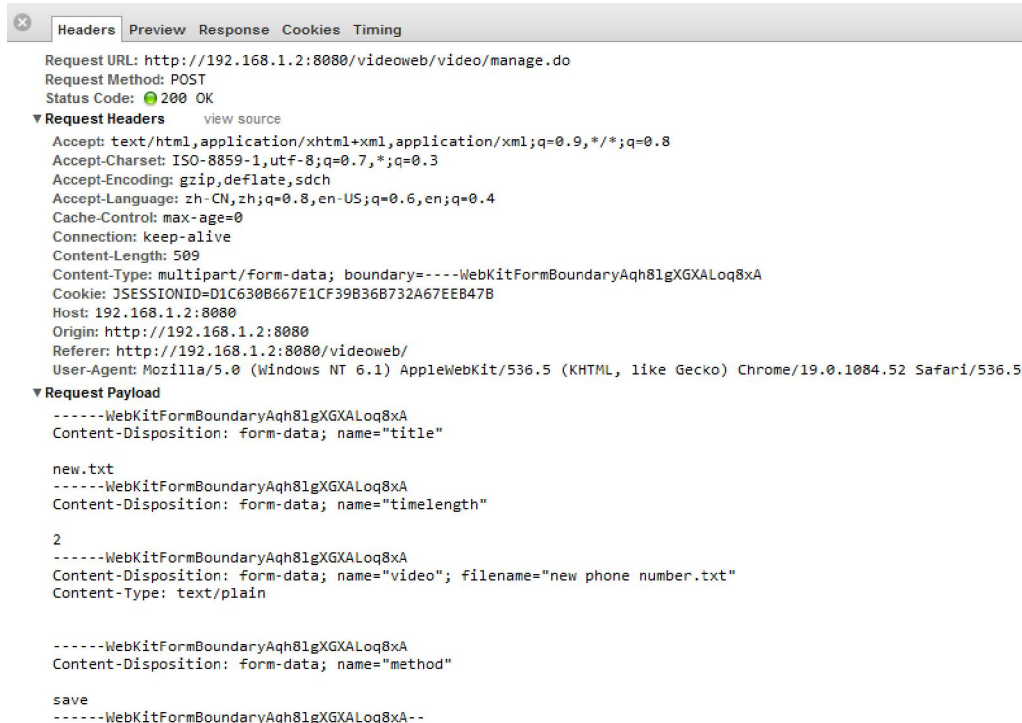


Figure 4.13 File Form

Obviously, using the tag of "`<input>`" in the html could upload the specific file. But in this case, a file is to be uploaded from the phone's

application not from the phone's browser. According to the http1.1 protocol, it is necessary to create an http request in the phone's application while uploading the file. Different fields of http request headers are shown in the following exhibit.



```
Request URL: http://192.168.1.2:8080/video/web/video/manage.do
Request Method: POST
Status Code: 200 OK

Request Headers
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Accept-Encoding: gzip,deflate,sdch
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.6,en;q=0.4
Cache-Control: max-age=0
Connection: keep-alive
Content-Length: 509
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryAqh81gXGXALoq8xA
Cookie: JSESSIONID=D1C630B667E1CF39B36B732A67EEB47B
Host: 192.168.1.2:8080
Origin: http://192.168.1.2:8080
Referer: http://192.168.1.2:8080/video/web/
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.52 Safari/536.5

Request Payload
----WebKitFormBoundaryAqh81gXGXALoq8xA
Content-Disposition: form-data; name="title"

new.txt
----WebKitFormBoundaryAqh81gXGXALoq8xA
Content-Disposition: form-data; name="timelength"

2
----WebKitFormBoundaryAqh81gXGXALoq8xA
Content-Disposition: form-data; name="video"; filename="new phone number.txt"
Content-Type: text/plain

----WebKitFormBoundaryAqh81gXGXALoq8xA
Content-Disposition: form-data; name="method"

save
----WebKitFormBoundaryAqh81gXGXALoq8xA--
```

Figure 4.14 http request headers

Referring to figure 4.14, the custom http request could be written according to the http protocol standards. The code implementation can be found at the Java file of *“cn.itcast.utils.SocketHttpRequester”*.

## 4.5 Web client program design

Based on the overall framework of the e-healthcare system, Google Calendar only plays a role as a transfer server. Why is the Google Calendar not operated directly but another web server is built to access it? Signing in to the Google's official website and enjoying the Google Calendar service would achieve exactly the same effect. However, it also brings some other problems. These problems are list below.

- 1) The doctor could access the patient's personal account which affects the privacy protection.
- 2) The Google Calendar's event editor contains limited fields which affects the expansibility.

- 3) The Google Calendar cannot provide some special functional modules such as receiving the file from the phone and show the energy consumption conclusion for the healthcare persons.

Considering the above these problems, it would be necessary to establish an integrated website. The architecture of this website consists of six different functional modules.

Table 4.15 the form of six modules

<i>Module</i>	<i>Function</i>
Account Manager	Login, logout, and offer the parameter of Google account connection
Validation before query	A query will be released after the successful authentication
Calendar	To show all the events
Synchronization	Maintain consistency of the event data
File Receiver	To receive the file uploaded from the phone
Consumption Display	To show the file content according to the date

#### 4.5.1 Account Manager

Each healthcare person, with a unique account, could sign in to the website through a browser and execute within his/her permission. This web client does not open to the public as in a forum, so that a register module is not provided. When there is the need to add a new user, contact must be made to the website administrator for a new account. Furthermore, the user group is small and the user's information does not often change, thus the website chooses an xml file in which to store the user's information. The format is as shown in figure 4.16.

```
<?xml version='1.0' encoding='UTF-8'?>
<root>
<user>
<name>Mark</name>
<passwd>123</passwd>
<email>chinesejie@qq.com</email>
</user>
</root>
```

Figure 4.16 users.xml

It is quite difficult for a non-art designer to write attractive web pages. In this case, it was finally necessary to download UI files of Google's official website in order to enhance the effect of the screen. The web page for reference is shown in figure 4.17 below.

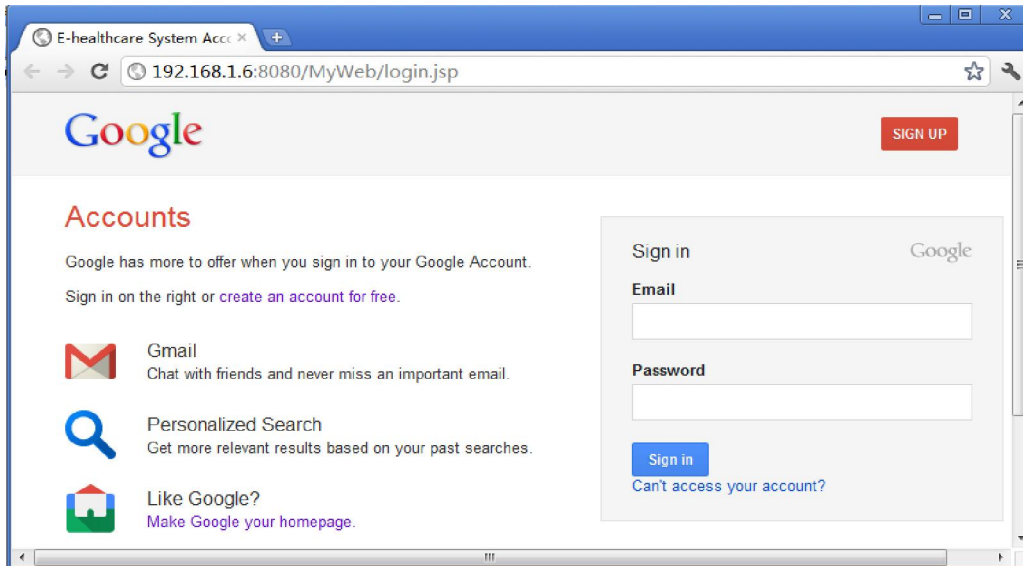


Figure 4.17 Home Page

Apart from login and logout service, the account manager also provides access to the Google Account service. All the patients' information were stored in an xml file. The format of the file is shown in figure 4.18.

```
<?xml version='1.0' encoding='UTF-8'?>
<root>
<account>
  <name>Service Liang</name>
  <gmail>service4liang@gmail.com</gmail>
  <password>020118114</password>
</account>
</root>
```

Figure 4.18 accounts.xml

With the advantages of the JDOM parser, it is possible to read and write XML data without using the original ecological JDK API which is complex and memory-consumptive. For example, the process to obtain the value of the tag “<gmail>”, is similar to that in figure 4.19:

```
public String getValue() {  
  
    SAXBuilder builder = new SAXBuilder();  
  
    Document doc = builder.build(new FileInputStream("xx.xml"));  
  
    Element root = doc.getRootElement();  
  
    List<Element> items=(List<Element>) XPath.selectNodes(root,  
    "//account");  
  
    for (Iterator<Element> ite = items.iterator(); ite.hasNext();) {  
  
        Element ele = ite.next();  
  
        return ele.getChild("gmail").getText();  
  
    }  
}
```

Figure 4.19 Code of getting specific value

#### 4.5.2 Validation before query

AOP is the abbreviation of “Aspect-oriented programming” which is a programming paradigm commonly used as a design pattern. It aims at improving the modularity by allowing the developers only pay close attention to the programming at one point or the crosscutting structure. The design idea of AOP is often used for logging, such as the Log4J of the Apache Foundation and the Joss-logging of Red Hat Inc. Which have both benefited from this idea. Based on the idea of aspect programming, it is possible to validate all the queries, but not, to affect the normal structure of the top logical design. The AOP could be also thought of as an interceptor when using it in the validation framework. Details are shown as follows.

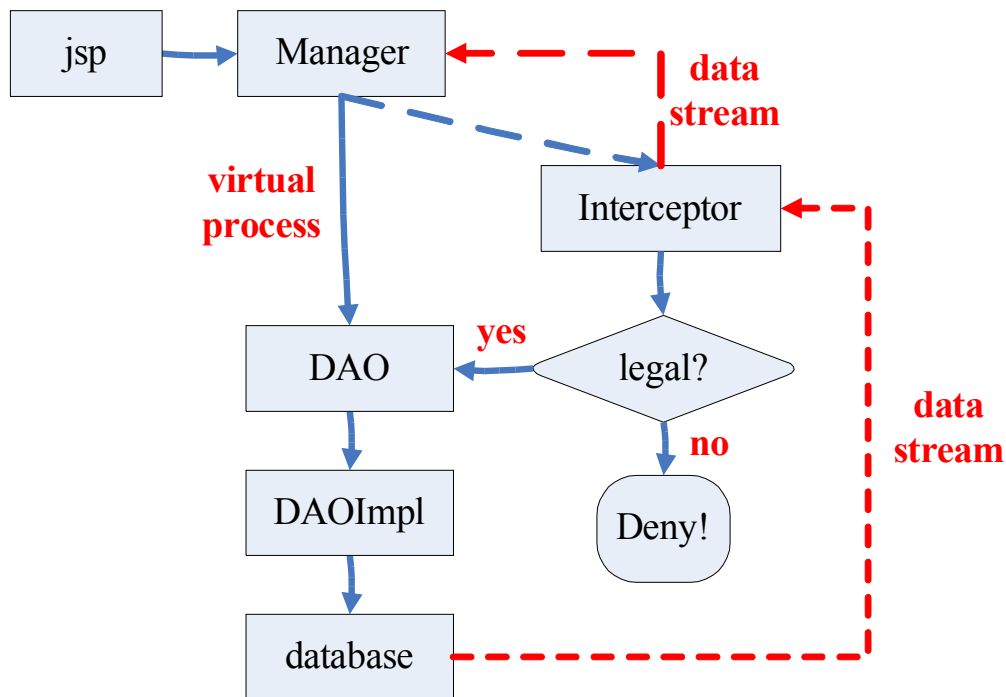


Figure 4.20 AOP diagram

Spring framework supports the AOP design pattern that has been integrated with the widely-used aspect-oriented extension called AspectJ. Additionally, using annotation to the program makes the code appear to be simpler.

### 4.5.3 Calendar

The calendar design is the largest proportion of the whole website. The calendar module includes the database design on events and participants, the display of the events and participants and the operation of the events.

Table4.21 doctor\_table

Field	Type	Null	Extra
<u>Id</u>	int(11)	No	Pk & auto_increment
username	varchar(30)	No	
firstname	varchar(20)	No	
lastname	varchar(20)	No	



Table 4.22 patient\_table

Field	Type	Null	Extra
<b><u>Id</u></b>	int(11)	No	auto_increment
<b>gmail</b>	varchar(30)	No	
<b>firstname</b>	varchar(20)	No	
<b>lastname</b>	varchar(20)	No	

Table 4.23 d\_p\_table

Field	Type	Null	Extra
<b><u>doctor id</u></b>	int(11)	No	FK(doctor_table, id)
<b><u>patient id</u></b>	int(11)	No	FK(patient_table, id)

Table 4.24 event\_table

Field	Type	Null	Default	Extra
<b><u>Id</u></b>	int(11)	No		auto_increment
<b>title</b>	varchar(255)	No		
<b>startTime</b>	datetime	No		
<b>endTime</b>	datetime	No		
<b>place</b>	varchar(255)	Yes	<i>NULL</i>	
<b>content</b>	text	No		
<b>remind</b>	int(11)	No		
<b>memo</b>	text	Yes	<i>NULL</i>	
<b>energy</b>	int(11)	No		

The form of the event\_table is merely a sample shown here. In the actual development, the web had built one table for one patient. The namespace of the patient's table consists of the Google account name with the suffix of "\_event", because the Google account is unique, so the name of table is also unique.

When visiting the database, another framework has been found—Hibernate. Through the encapsulation by Hibernate, it is possible to obtain the data, such as accessing a class, with the assistance of the Hibernate encapsulation.

FullCalendar is a jQuery calendar plugin, which uses Ajax to obtain each month's calendar events and can be configured to use their own calendar event source, such as obtained from the Google Calendar event. In order to realize the function of the events' show, it is necessary

to obtain the data from the local database. FullCalendar plugin should be configured to use the correct source through the servlet.

Firstly, a servlet class is created and then it is, configured to the web.xml. This is shown below:

```
<servlet>
  <servlet-name>JSONAction</servlet-name>
  <servlet-class>com.liang.action.JSONAction</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>JSONAction</servlet-name>
  <url-pattern>/json</url-pattern>
</servlet-mapping>
```

Figure 4.25 Servlet configuration

Therefore, any request using the url address “/json” could visit the servlet class “com.liang.action.JSONAction”, this class will obtain the data from the database and return the data to the request.

On the other side, the JQuery should build a servlet similar to the following:

```
events:function(start, end, callback) {
  $.ajax( {
    type : "get",
    url : "json",
    dataType : "json",
    success : function(data) {
      //get the data
    }
  });
};
}
```

#### 4.5.4 Synchronization

The Google Calendar could catch the changes quickly and make all the events belong to the same account thus remaining the consistency. This organization's event database should also maintain the consistency with the Google Calendar. Google provides access for client applications to view and to update calendar events in the form of Google Data API feeds. With the powerful Google Calendar Data API, this web could create new events, edit or delete existing events, and query for existed events.

However, the question then involves how to know that an event existed in the Google Calendar. The unique id is written into the title of the event in order to solve this problem. When it is necessary to delete the old event, the application will search the event whose title contains the unique id, and then delete the event that have been found.

#### 4.5.5 File Receiver

The display on the daily consumption depends on the file uploaded by the client phone. Generally, uploading a large file which is greater than 2Mb should choose socket programming, which supports the resumption of transmission. Http transmission is stateless and the transport scale is limited by the server's hardware configuration. Generally, http transmission only supports a file size of less than 2 Mb. It was observed on the file feature, that even if the whole day's records were written inside; the file size is still small. The most important consideration is that the socket server is independent of the web server; it is cost-consumptive to maintain two servers. Finally, the http pattern is selected to build the module for the file receiver.

#### 4.5.6 Consumption Display

When the file uploaded from the phone is received, the web could display the daily consumption to the healthcare person.

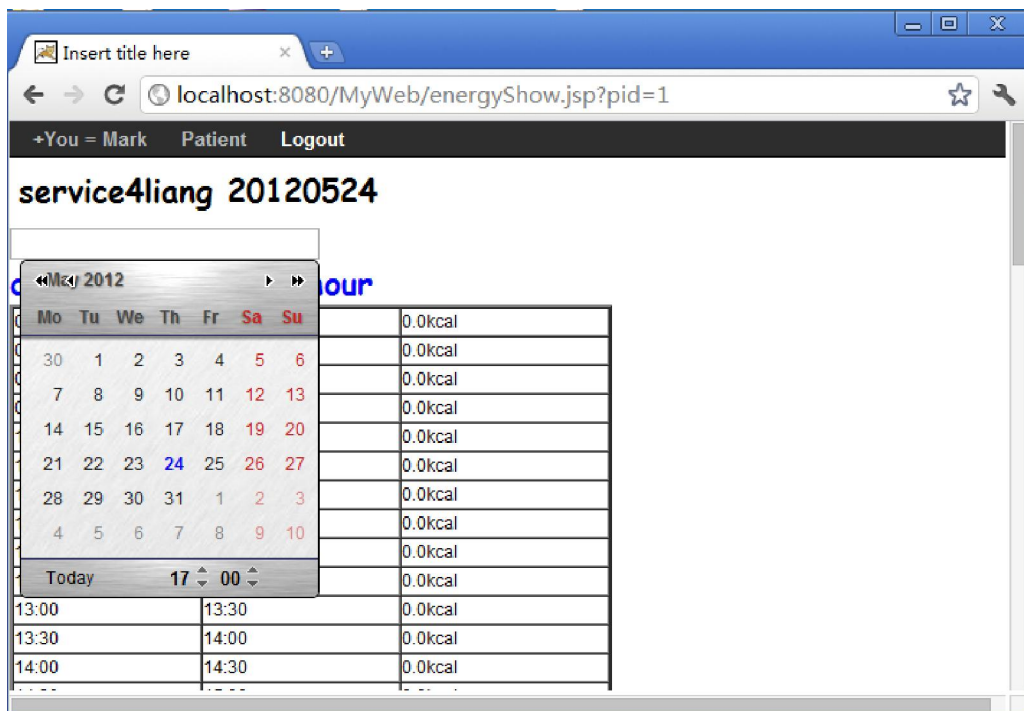


Figure 4.26 the daily consumption pages

Referring to the Figure 4.26; a date was selected and the corresponding conclusion will be shown in a form. In order to improve the user's experience, the newest JavaScript Calendar, JSCal2[11] was used as a tool in order to easily select the date.

## 5 Results

### 5.1 GPS calories collector

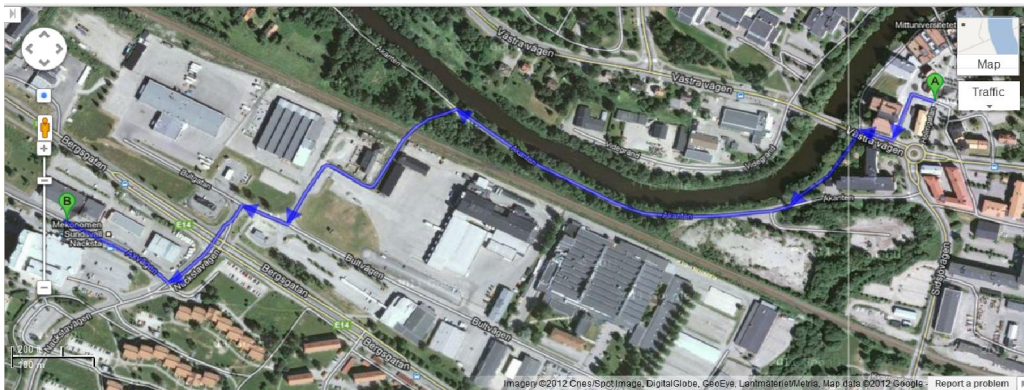


Figure 5.0 Google Map (Sundsvall Nacksta ICA - Campus )

Refer to the Google Map above, the tester walks from the Nacksta ICA supermarket to the campus which takes nearly 20 minutes. The data is extracted in a form shown below.

Table 5.1 location records (only for test, not recorded in the application)

	Time	Lat	Lng
1	May 24 13:10:28	62.390114343725145	17.259278828278184
2	May 24 13:11:34	62.39046265371144	17.260527899488807
3	May 24 13:12:40	62.390748099423945	17.26
4	May 24 13:13:44	62.391197411343455	17.263594921678305
5	May 24 13:14:49	62.39166222978383	17.26581646129489
6	May 24 13:15:51	62.3917374573648	17.266081580892205
7	May 24 13:16:56	62.392155546694994	17.268011178821325
8	May 24 13:18:01	62.39213434047997	17.269227895885706
9	May 24 13:19:09	62.393415346741676	17.27113176137209
10	May 24 13:20:17	62.39320039283484	17.272433303296566
11	May 24 13:21:47	62.39270200487226	17.275233613327146
12	May 24 13:22:52	62.39255666267127	17.27557006292045
13	May 24 13:23:57	62.392487595789135	17.279988918453455
14	May 24 13:25:03	62.39293208811432	17.28068989701569
15	May 24 13:26:09	62.39304180722684	17.28110228665173
16	May 24 13:27:15	62.39318899344653	17.28263173252344

Table 5.2 distance, time duration, speed between two points

duration	Length of time	distance	speed/ mph	kcal unit	sum
1-2	65s/1.083min	143.86m	4.95	10.89	11.80
2-3	65s/ 1.083min	187.24m	6.44	13.78	14.92
3-4	63s/1.05min	163.77m	5.81	12.86	13.50
4-5	65s/1.083min	252.12m	8.67	16.14	17.48
5-6	62s/1.03min	30.546m	1.10	3.36	3.472
6-7	64s/1.067min	219.39m	7.67	15.09	16.1
7-8	64s/1.066min	135.47m	4.73	10.89	11.62
8-9	68s/1.13min	251.83m	8.28	15.48	17.55
9-10	68s/1.13min	146.67m	4.82	10.89	12.35
10-11	89s/1.48min	316.09m	7.94	15.49	22.97
11-12	64s/1.06min	40.37m	1.41	3.36	3.58
12-13	65s/1.083min	491.97	7.56	16.93	32.6
13-14	66s/1.1min	91.24m	3.09	5.23	5.76
14-15	66s/1.1min	47.42m	1.60	3.36	3.69
15-16	65s/1.083min	170.86m	5.88	12.86	13.93

The table 5.2 shows that the calorie consumption each minute. It should be noted that the GPS calories collector works accurately outdoors and, indoors, although it still works it is of a greatly inferior quality. Thus another calories collector was created for use both indoors and outdoors. The pedometer collector would be introduced in the paper of the second member of the group.

## 5.2 Upload file

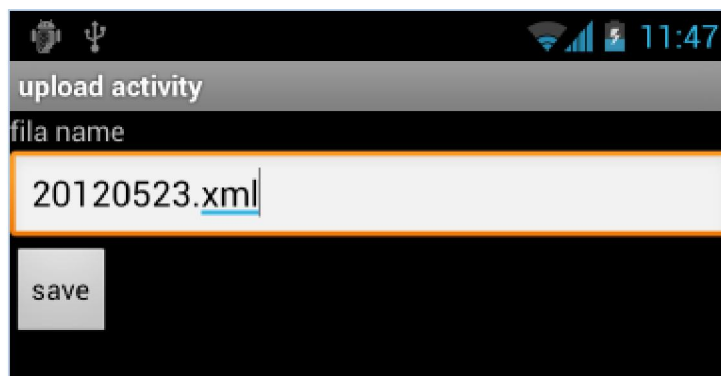


Figure 5.3 Upload file panel

The application only uploads the file after the user agrees. The file name is composed of the current date and the file is stored in the SD card.

### 5.3 Calendar

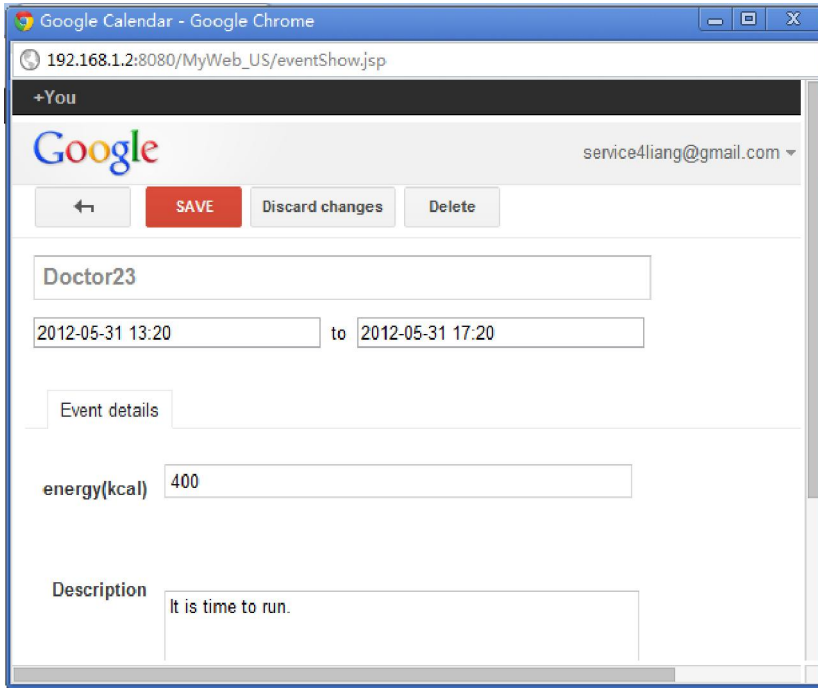
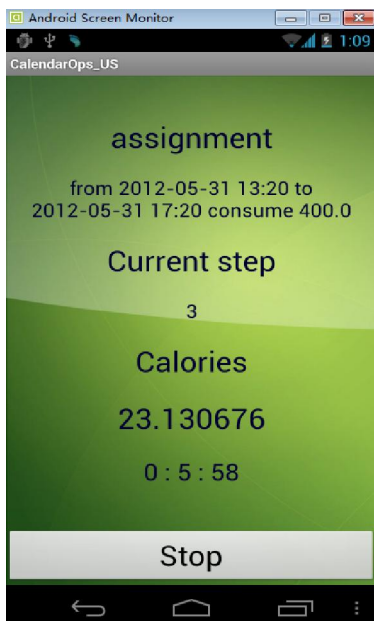


Figure 5.4 Calendar Event

Referring to figure 5.4, a new event was created on the computer browser. The application was then run using the phone, as shown below, so that it is possible to determine that the time and the required calories consumption are the same.



## 6 Conclusions

The goals of this project have been achieved. The e-healthcare system has been realized in a clear and complete way. Firstly, the phone application is able to remind the patient to perform exercise according to the calendar. Secondly, the system also offers a website for the healthcare person to operate the calendar and to read the daily exercise report. In real scenarios, there are still no organizations which are using this E-healthcare System and the system has only been tested by an internal University basis. The web server which is deployed for the healthcare personnel is the most costly, but they would be able to sign in to the website in order to give the latest calendar anywhere and at any time, which means that the organization must hire a host on the internet to run this project. Additionally, some related technical staff are required to maintain the normal running of the host. For the remainder, the deployment on the phone client is rather simple and only requires to install the application after the user agrees to be monitored by the relevant healthcare personnel.

For the high-level and complex design, it might be necessary to add the back-stage management into the organization's website. Thus, when the organization adds new accounts to the website, it is convenient to operate online instead of requiring professional modification in relation to the configuration file of the project. Generally speaking, the GPS calories collector is most accurate and power-saving mode when it works outdoors, but if the person is not outdoors, then the pedometer calories collector will work instead of the GPS calories collector. However, it requires the user to touch the screen to switch between these two modes. An attempt has made to implement an automatic switching mode, but it finally failed due to fuzzy concepts on the GPS satellites data. Perhaps this humane setting can be achieved as knowledge in this field is increased.

For a developer who focuses on business logic, the UI design is very abstract and could be ignored. From an artistic perspective a great deal was learnt about UI design from the Google website, but all the business logic implementations were written by the author.



## References

- [1] Google Accounts  
<https://accounts.google.com/ServiceLogin> | 4<sup>th</sup> June 2012
- [2] FullCalendar - Full-sized Calendar jQuery Plugin  
<http://arshaw.com/fullcalendar/> | 4<sup>th</sup> June 2012
- [3] NAMI | Mental Illness and Exercise  
[http://www.nami.org/Content/NavigationMenu/Hearts\\_and\\_Minds/Exercise/Mental\\_Illness\\_and\\_Exercise.htm](http://www.nami.org/Content/NavigationMenu/Hearts_and_Minds/Exercise/Mental_Illness_and_Exercise.htm) | 4<sup>th</sup> June 2012
- [4] The relation of physical activity and exercise to mental health.  
<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1424736/?page=8>  
| 4<sup>th</sup> June 2012
- [5] Android - Introducing Ice Cream Sandwich  
<http://www.android.com/about/ice-cream-sandwich/>  
| 4<sup>th</sup> June 2012
- [6] Apache Struts  
<http://struts.apache.org/> | 4<sup>th</sup> June 2012
- [7] Hibernate - JBoss Community  
<http://www.hibernate.org/hibernate> | 4<sup>th</sup> June 2012
- [8] SpringSource.org  
<http://www.springsource.org/features> | Date: 4<sup>th</sup> June 2012
- [9] Obtaining User Location | Android Developers  
<http://developer.android.com/guide/topics/location/obtaininguser-location.html> | 4<sup>th</sup> June 2012
- [10] Google Maps JavaScript API V2 Reference  
<https://developers.google.com/maps/documentation/javascript/v2/reference> | 4<sup>th</sup> June 2012
- [11] The New Coolest DHTML Calendar  
<http://www.dynarch.com/projects/calendar/> | 4<sup>th</sup> June 2012

## **Appendix A: Documentation of own developed program code**

See the attachment "MyWeb\_US.war" and  
"CalendarOps\_US.rar".