ESTIMATE OF HUMAN'S DEHYDRATION LEVEL BASED ON EDA AND TRI-AXIAL DYNAMIC ACCELERATIONS

A Project

Presented to

The Faculty of the Graduate School

At the University of Missouri

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

Bo Cai

Dr. Yi Shang, Advisor

JANUARY2014

The undersigned, appointed by the dean of the Graduate School,
have examined the entitled

ESTIMATE OF HUMAN'S DEHYDRATION LEVEL BASED ON EDA AND TRI-AXIAL DYNAMIC ACCELERATIONS

Presented by Bo Cai

A candidate for the degree of

Master of Science

And hereby certify that, in their opinion, it is worthy of acceptance.

_____

Dr. Yi Shang

_____

Dr. Dong Xu

_____

Dr. Jianlin Cheng

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

If people just start to rehydrate their body when they feel thirsty, health experts consider it is already late. Advances in Exercise Physiology science, sensor and smart phone technology make an intelligent rehydration prompter possible. This project endeavors to use a combination of modern technology and techniques to produce a rehydration prompter that reliably reminds its user with proper timing to rehydrate and avoid losing further body water.

In this project, a smart phone application has two modes based on a user's physical state. Calm mode collects EDA and body temperature data from Q sensor. An experiment is designed to find the relationship between EDA and people's dehydration level. A web server was built to update data and show result of calm mode. Active mode collects data from the phone accelerometer and use these data to calculate calorie consumption. Each mode analyzes these data using exercise physiology science to predict a point at which the user needs to rehydrate. The program is self-contained, requiring no outside processing or expert interpretation, meaning that the solution as a whole can be easily integrated into one's schedule and utilized by the general populace without trouble.

1. Introduction

Every health expert will tell you that once you feel thirsty, it is already too late to rehydrate. If people don't rehydrate in time, it may cause concentration, fatigue and some other problems. If we have some professional equipment, we can take some tests to see if we need to rehydrate. But as a normal person in our daily life how can we know when to drink some water to keep our body performing well if we aren't thirsty?

As Smartphone become more common and their capabilities improve, their usefulness in a variety of application areas also improves. The portability of Smartphone makes them ideal for many tasks that in the past required specialized hardware. And also the sensor technology develops rapidly: it is more portable and accurate sensors are made for a variety of uses. To solve this rehydration problem we can construct an application for smart phones. In order to see if people need water, we need to divide a person's physical state into two parts first. One is the active state, which means the person is doing some sports or working out. Another is the calm state, which means the person is in a resting mode.

For the active state, since calories burned is related to water loss, we can use the smart phone's accelerometer sensor to get three coordinate axis accelerations which can be used to calculate calorie consumption. Besides that, in order to get more accurate results we also have to know some basic information about the person: weight, height and gender. This information has a great influence on calorie consumption. How can we evaluate the result? Since every running machine can

calculate calorie consumption, I compared the result of the system and the calorie consumption of a running machine. The error was within 3.3% each time.

For the calm state, I used the Q sensor to collected EDA data (skin conductance data). It can help to judge the skin's dryness level, which can reflect a person's dehydration level. Since there is no clear relationship between the EDA data and the person's dehydration level, I designed one experiment to find how EDA data can reflect dehydration level and based on the result I defined five levels of dehydration. For the evaluation part, since there is no ground truth I added a feedback part in this application after it shows the result to the user. The comparison between the result and feedback will be done in a webpage which people can also visit to view the comparative result.

2.Related Work

2.1 Affectiva Q Sensor

The Affectiva Q Sensor is wireless and wearable. It’s small and comfortable so participants can wear it without distraction. There are no messy gels to apply, no wires to tape, and nothing to configure. Simply strap on the Q and it turns on automatically and begins collecting data. Take it off, and it will turn off automatically after two minutes to conserve battery life.

Figure 2.1: Affectiva Q sensor

The Affectiva Q Sensor can measure electro dermal activity(EDA), which can capture the intensity of an individual’s current experience and their skin's dryness level. Besides that it can also measure skin temperature, which is also helpful to make a judgment of a person's dehydration level. The Affective Q sensor is Bluetooth accessible, which means we can get all the data we need in real time.

2.2 Accelerometer noise reduction

The accelerometer sensor measures the acceleration applied to the device, including the force of gravity. The sensor framework uses a standard three-axis coordinate system to express data values. For most sensors, the coordinate system is defined relative to the device's screen when the device is held in its default orientation (see figure 2). When the device is held in its default orientation, the X axis is horizontal and points to the right, the Y axis is vertical and points up, and the Z axis points toward the outside of the screen face. In this system, coordinates behind the screen have negative Z values

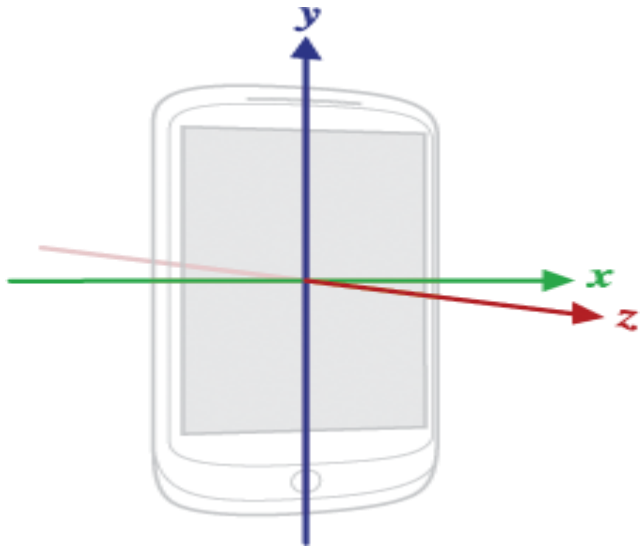

Figure 2.2: Coordinate System

Conceptually, an acceleration sensor determines the acceleration that is applied to a device (A_d) by measuring the forces that are applied to the sensor itself (F_s) using the following relationship:

$A_d = - \sum F_s / mass$

However, the force of gravity is always influencing the measured acceleration according to the following relationship:

$A_d = -g - \sum F / mass$

For this reason, when the device is sitting on a table (and not accelerating), the accelerometer reads a magnitude of g = 9.81 m/s^2. Similarly, when the device is in free fall and therefore rapidly accelerating toward the ground at 9.81 m/s^2, its accelerometer reads a magnitude of g = 0 m/s^2. Therefore, to measure the real acceleration of the device, the contribution of the force of gravity must be removed from the accelerometer data. This can be achieved by applying a high-pass filter. Conversely, a low-pass filter can be used to isolate the force of gravity. Here is the solution to remove the influence of accelerometer of gravity:

alpha is calculated as t / (t + dT),where t is the low-pass filter's time-constant and dT is the event delivery rate. Here the value of alpha is 0.8;

- Isolate the force of gravity with the low-pass filter.

```
gravity[0]= alpha * gravity[0]+(1- alpha)*event.values[0];
gravity[1]= alpha * gravity[1]+(1- alpha)*event.values[1];
gravity[2]= alpha * gravity[2]+(1- alpha)*event.values[2];
```

- Remove the gravity contribution with the high-pass filter.

```
linear_acceleration[0]=event.values[0]- gravity[0];
```

```
linear_acceleration[1]=event.values[1]- gravity[1];

linear_acceleration[2]=event.values[2]- gravity[2];
```

2.3 Energy Expenditure

2.3.1 Energy Expenditure Detection Based on Tri-Axial Accelerometer

The detection of physical activity energy expenditure(PAEE) based on tri-axial accelerometer was presented by ZHU Guozhong, WEI Caihong and PAN Min in 2011. They had done the comparative experiments between waist, knee and lip and found the best position to wear the device is determined to be on the waist. In their research they introduced an algorithm based GM(Geometric Mean).

GM=$\sqrt{(A_x^2 + A_y^2 + A_z^2)/3}$

A_x , A_y and A_z are acceleration of three coordinate axes after filtering.

In this system, I only consider one physical activity, which is running. Figure2.3 shows GM waveform of running with a speed of 10 km/h.

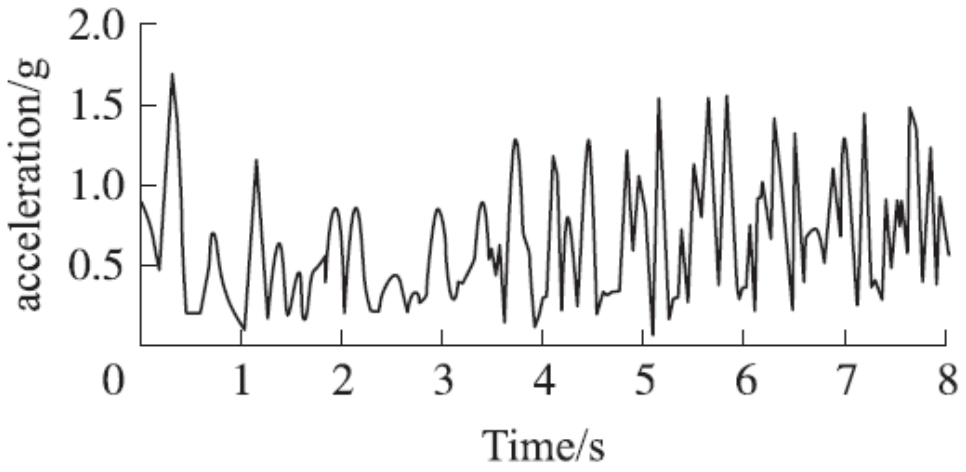

Figure 2.3: GM waveform of running with a speed of 10 km/h

Based on classical Newton's Mechanics' formula:

$E=F*S$

$F=\mu*mg$

$S=\frac{1}{2}\ at^2$ (uniformly accelerated motion)

In this system, the motion of running is variable accelerated motion, so we need to use calculus to calculate energy expenditure.

$E=\frac{1}{2}\mu*mg\int_{a_1}^{a_2} da \int_{t_1}^{t_2} tdt$

In the formula above, a is the value of GM, μ is another factor that will be determined by experiment.

2.3.2 Energy Expenditure Detection Based on Metabolism

Metabolism is the set of life-sustaining chemical transformations within the cells of living organisms. Metabolic rate determines how much energy people consume. There are two kinds of metabolic rates, one is Basal metabolic rate (BMR), which is the energy used by an organism at complete rest, and the one is relative metabolic rate (RMR), which is the ratio of energy expenditure of physical activity and BMR.

Basal metabolic rate is determined by a person's gender, age, weight and height.

Female: BMR = 655 + (9.6 x weight in kilos) + (1.8 x height in cm) - (4.7 x age in years)

Male: BMR = 66 + (13.7 x weight in kilos) + (5 x height in cm) - (6.8 x age in years)

Relative metabolic rate can be checked by the following form:

Activity	RMR
Running: 15 km/ h	18
Running: 10 km/h	10
Running: 5 km/ h	6
Walking: 2 km/ h	2.2
Basketball	6~11
Volleyball	6~8
Soccer	6~7
Table tennis	3.9
Swimming: 400 meter	27.7
Swimming: 1500 meter	21.9
Lifting	104
Skiing	3.3~6.5

Table 2.1: RMR value of different sports

In this System, we only consider running. For the experiment, running is based on a speed of 10 km/h.

Besides BMR and RMR, body surface area(BSA) is another important factor to energy expenditure. BSA is determined by gender, weight and height. Here are two formulas to calculate BSA for different genders:

Male: BSA = 0.00607 x height in cm + 0.0127 x weight in kilos - 0.0698 (m^2)

Female: BSA = 0.00568 x height in cm + 0.0126 x weight in kilos - 0.0461 (m^2)

Since we have all this information we can calculate energy expenditure by the following formula:

E=(RMR +1.2) x BMR x BSA x T (second)

3. Implementation

3.1System Architecture

The whole system includes two modes. Calm mode collects data from Affectiva and Q sensor then processes the data and sends it to a web server. On the server side, all this data will be stored in a database and also displayed in a webpage. Active mode collects data from the smart phone's accelerometer, processes the data and then returns the result on phone's screen.

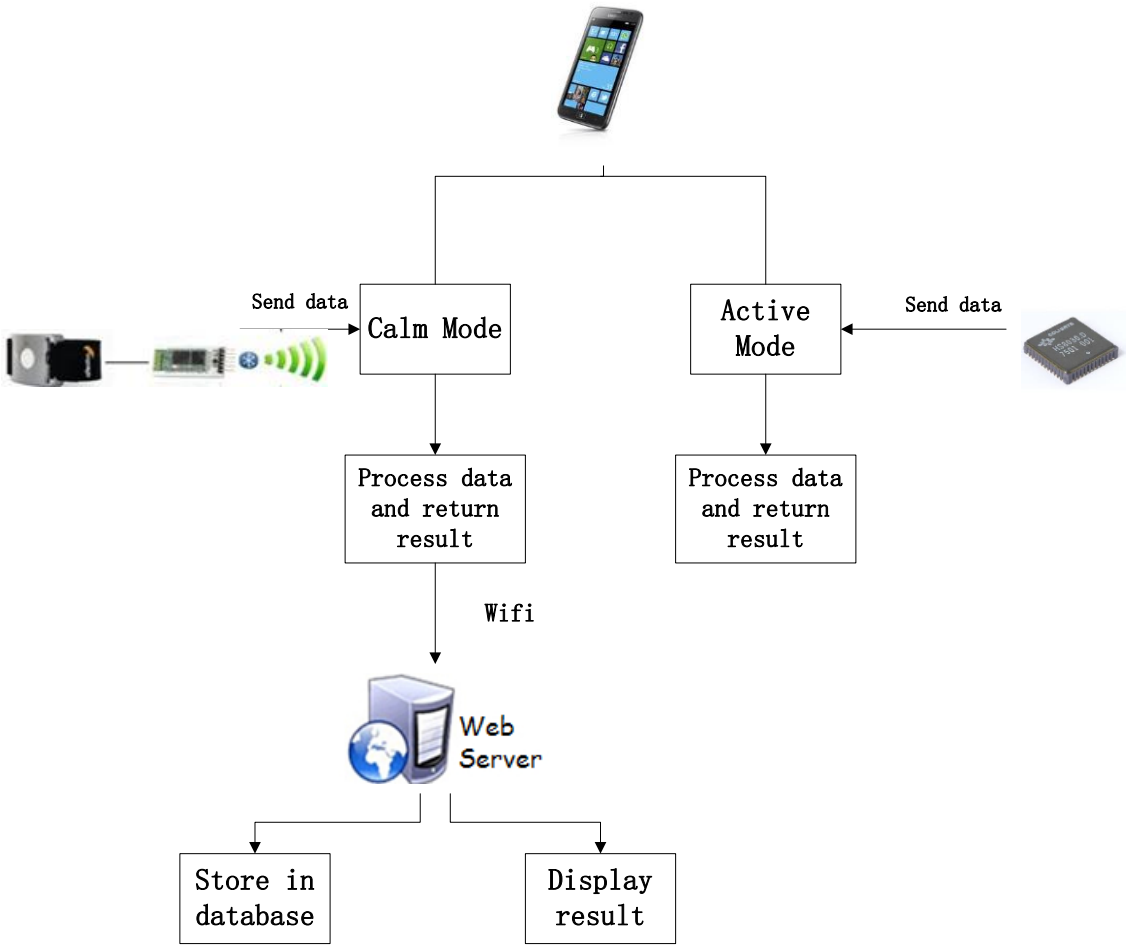

Figure 3.1: System Architecture

3.2calm mood implementation

3.2.1 Bluetooth connection

The Affective Q sensor is Bluetooth accessible and the Android platform also includes support for the Bluetooth network stack, which allows a device to wirelessly exchange data with other Bluetooth devices. The application framework provides access to the Bluetooth functionality through the Android Bluetooth APIs. These APIs let applications wirelessly connect to other Bluetooth devices, enabling point-to-point and multipoint wireless features.

Using the Bluetooth APIs, this application can perform the following:

- Scan for other Bluetooth devices
- Establish RFCOMM channels
- Connect to other devices through service discovery

In order to scan the Affectiva Q sensor using an android smart phone, we need to set up Bluetooth first, which includes two steps, one is to get the Bluetooth Adapter and another one is enable Bluetooth. After setting up Bluetooth, we can use the Bluetooth Adapter to find the Affectiva Q sensor either through device discovery or by querying the list of paired (bonded) devices. Device discovery is a scanning procedure that searches the local area for Bluetooth enabled devices and then requests some information about each one. After setting the Affectiva Q sensor to discoverable mood, this sensor will respond to the discovery request by sharing some information, such as the device name, class, and its unique MAC address. Using this information, an Android smart phone performing discovery can then choose to initiate a connection to the Affectiva Q sensor.

Since this connection is made for the first time, a pairing request is automatically presented to the user. When a device is paired, the basic information about that device (such as the device name, class, and MAC address) is saved and can be read using the Bluetooth APIs. Using the known MAC address for the Affectiva Q sensor, a connection can be initiated with it at any time without performing discovery (assuming the device is within range).

Since we paired two devices, we need to build the connection between them. In order to create a connection between your application on two devices, you must implement both the server-side and client-side mechanisms, because one device must open a server socket and the other one must initiate the connection (using the server device's MAC address to initiate a connection). The server and client are considered connected to each other when they each have a connected Bluetooth Socket on the same RFCOMM(Radio frequency communication, which is a simple set of transport protocols) channel. At this point, each device can obtain input and output streams and data transfer can begin.

In this application, the Affectiva Q sensor is considered as server-side, so that we don't need to implement and an Android smart phone is considered as client-side. In order to initialize a Bluetooth Socket to connect to the Affectiva Q sensor on client-side, we use the Bluetooth Device to get a Bluetooth Socket by calling createRfcommSocketToServiceRecord (UUID). UUID(**universally unique identifier**) is an identifier standard used in software construction, standardized by the Open Software Foundation (OSF) as part of the Distributed Computing Environment

(DCE).The UUID passed here must match the UUID used by the sensor. For the Bluetooth Socket, the UUID is 00001101-0000-1000-8000-00805F9B34FB.Then the system needs to perform an SDP lookup on the Affectiva Q sensor in order to match the UUID. After the sensor accepts the connection, it will share the RFCOMM channel to use during the connection and we can get the data from sensor through this channel.

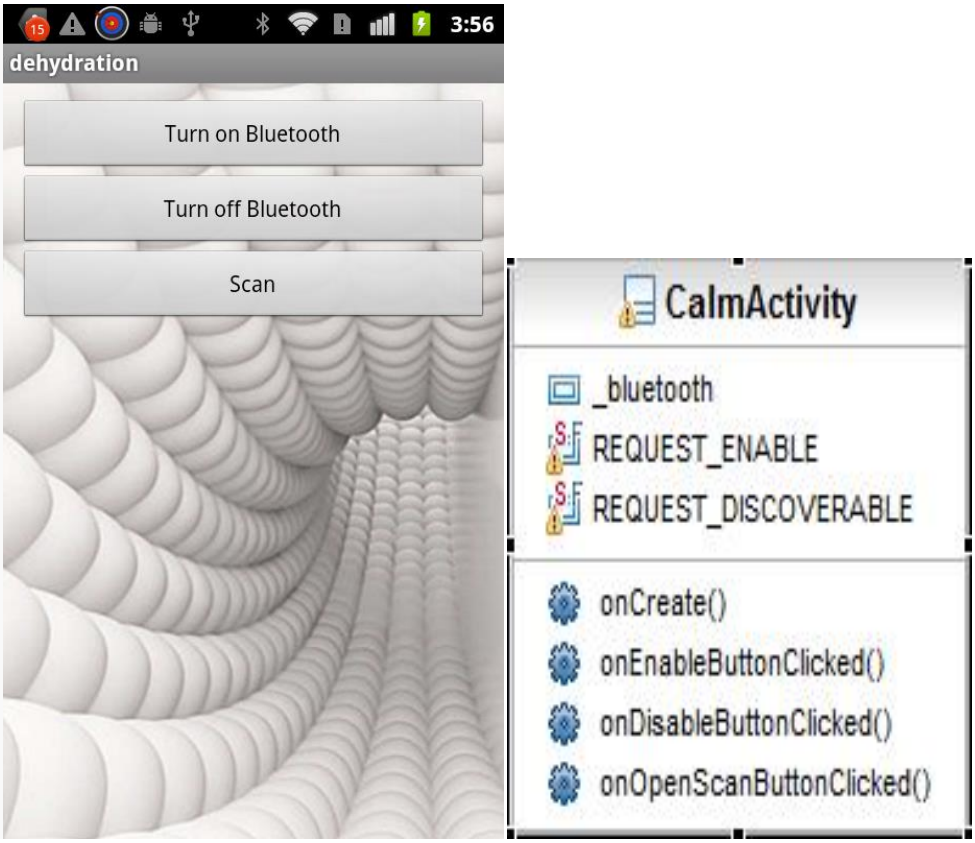

Figure 3.2: Setting screen

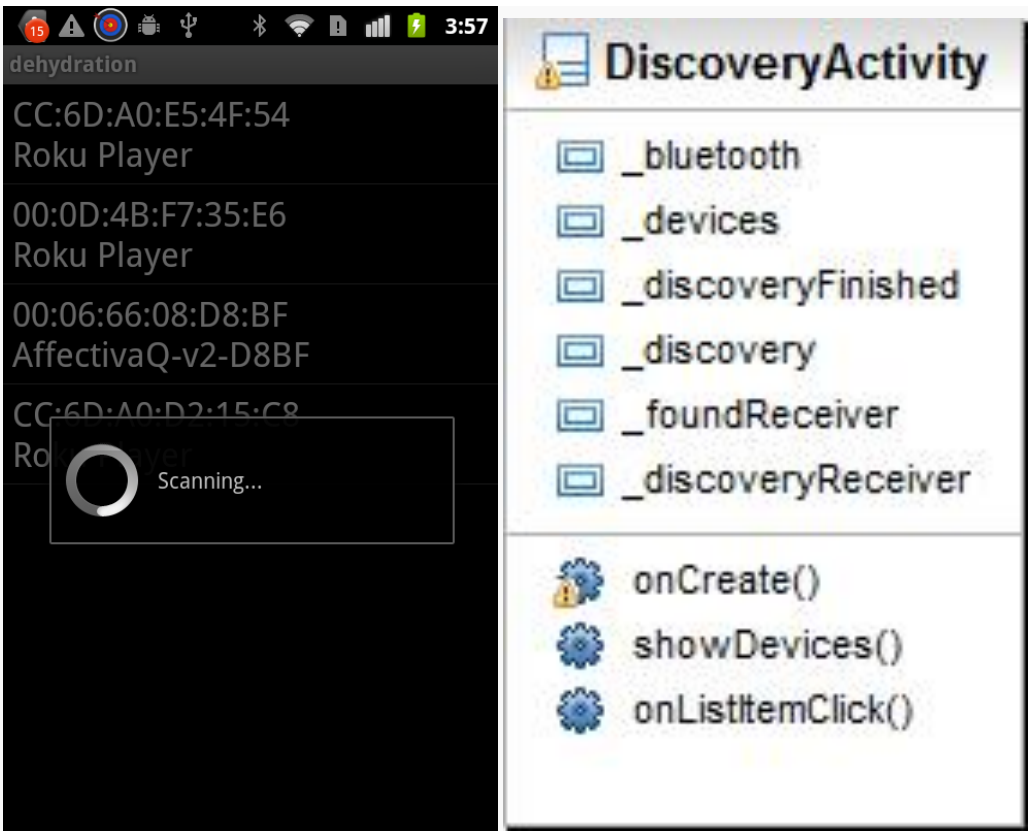

Figure 3.3: Scanning Device screen

After the user turns on Bluetooth for the smart phone and clicks the scan button, the application will start to scan for available devices and list them. From Figure 3.3 we can see that the third device is the Affectiva Q sensor which has a Mac address of 00:06:66:08:D8:BF.

3.2.2 dehydration level

We can receive three types of real time data from the Affectiva Q sensor: skin temperature, EDA(body conductance) and tri-axis acceleration. In this application, we only use skin temperature and EDA data.

Users wear the Affectiva Q senor on their wrist, the normal temperature of the

wrist should be around 29 degrees to 32 degrees. If the temperature collected in this application is higher than 34 degrees, then the user is probably running a fever, so the system will remind the user to drink some water by popping a toast window.

In order to find the relationship between EDA and a person's dehydration level, I designed an experiment with 10 participants. All of them were very thirsty at the beginning of the test and they were all considered to be hydropenic. I used the Affeciva Q sensor to collect their EDA data at this condition. Then I made them drink some water, but not very much to make sure they were still thirsty, but feel slightly better than the last stage. Then I collected their EDA data. I repeated the same procedure three times until they didn't feel thirsty at all. There were 50 samples from 10 people in this test. Each sample included five pieces of EDA data, which came from 5 steps of this test.

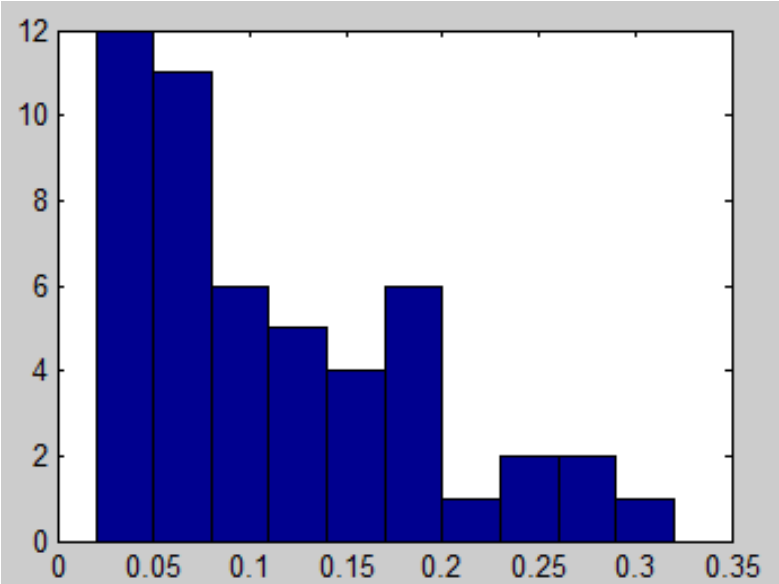

Figure 3.4: Histogram of EDA(dehydration level 1)

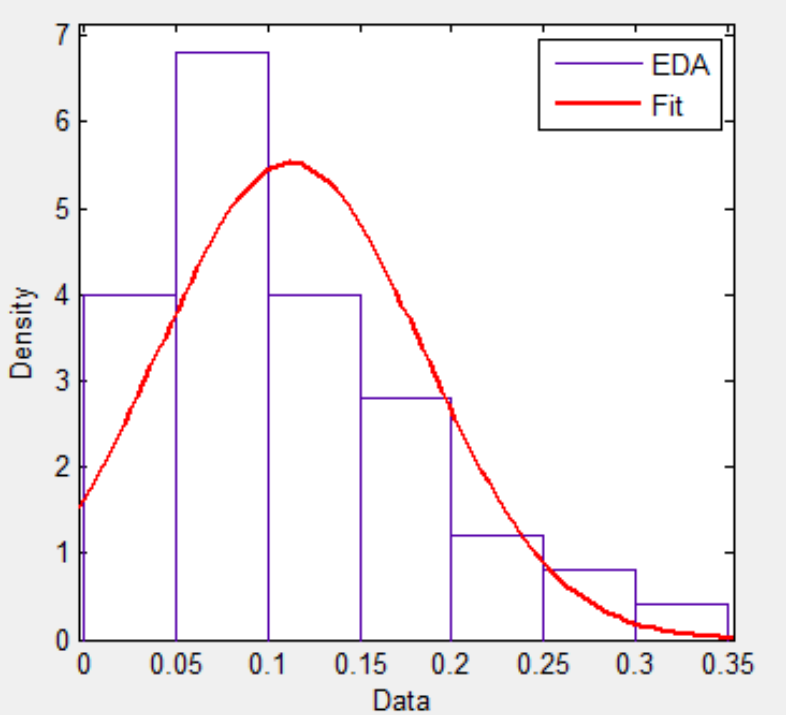

Figure 3.5:Distribution fitting(dehydration level 1)

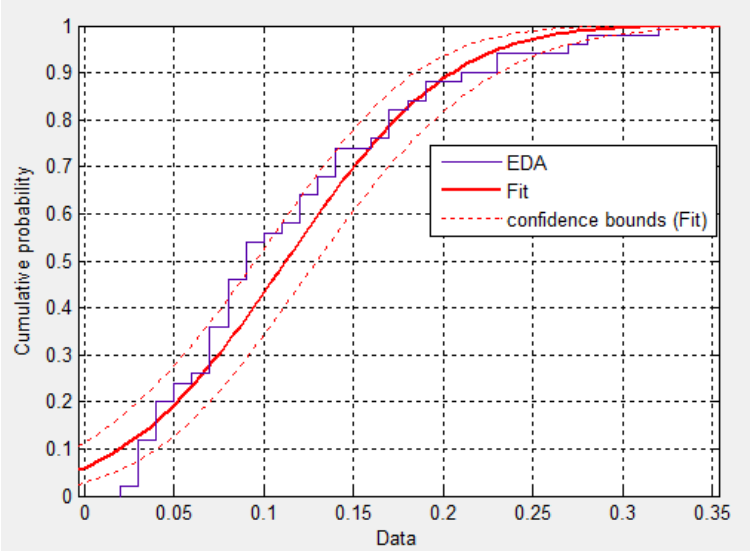

Figure 3.6: CDF display(dehydration level 1)

Figure 3.4 shows the original data of user's EDA data at first step of test, x-axis is the value of EDA and y-axis is the amount of samples in this range of EDA. Figure 3.5 is the distribution fitting of the data; I used normal distribution to fit. The peak is around 0.11 and 90% data are smaller than 0.25. Figure 3.6 is the Cumulative Distribution Function(CDF) display of the distribution. From this figure

we can see after 0.2 on x-axis, the curvature of the curve changed very little until the end. Based on this information, I chose the middle value between the peak value and 0.2,which is 0.155 as the upper limit of dehydration level 1.

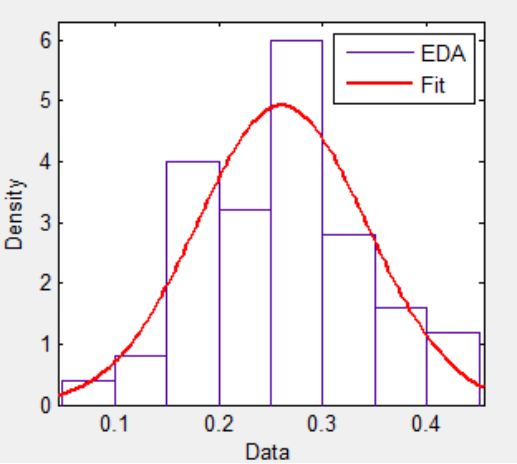

Figure 3.7:Distribution fitting figure (dehydration level 2)

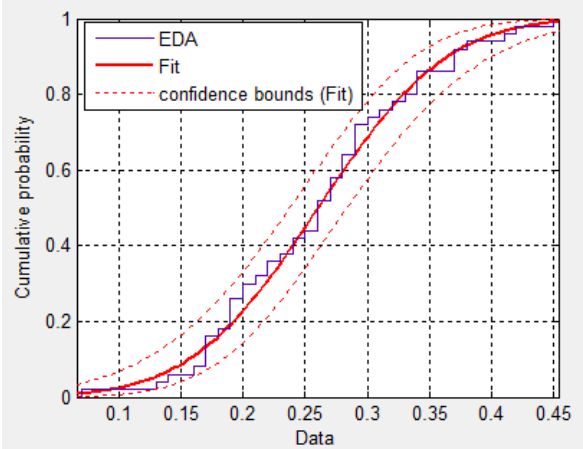

Figure 3.8 :CDF display(dehydration level 2)

These are the figures from step 2. I measured their EDA value 10 minutes after they drank some water. Based on these figures, I chose 0.320 as the upper limit of dehydration level 2.

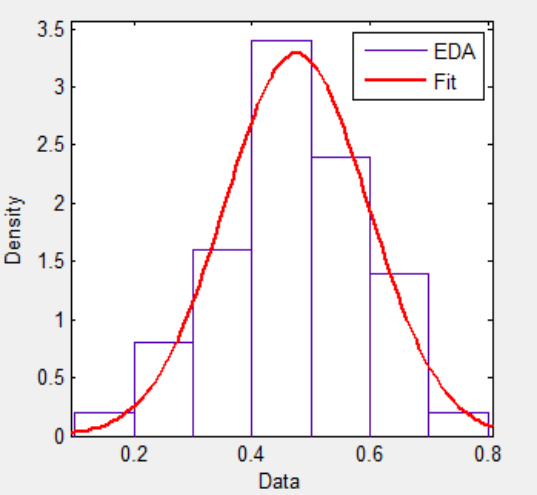

Figure 3.9: Distribution fitting figure (dehydration level 3)

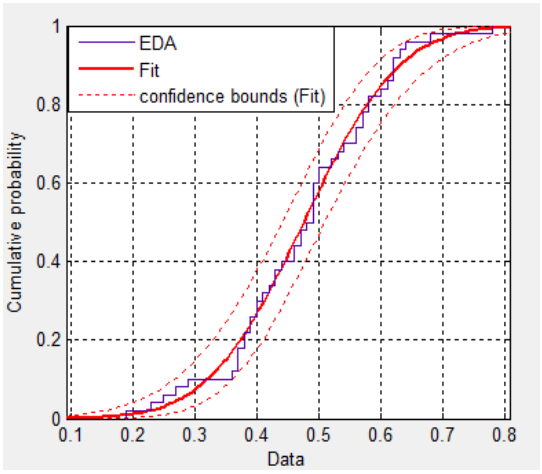

Figure 3.10 :CDF display(dehydration level3)

These are the figures from step 3. I choose 0.555 as the upper limit of dehydration level 3.

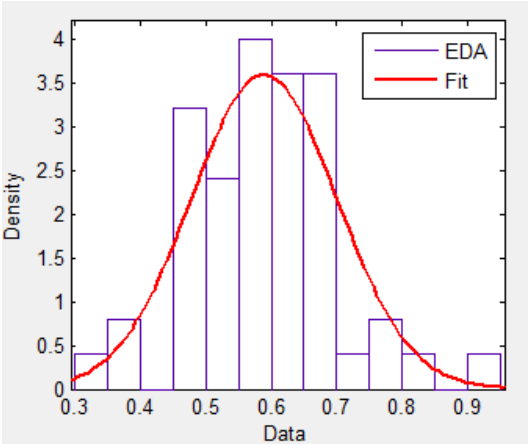

Figure 3.11: Distribution fitting figure (dehydration level 4)

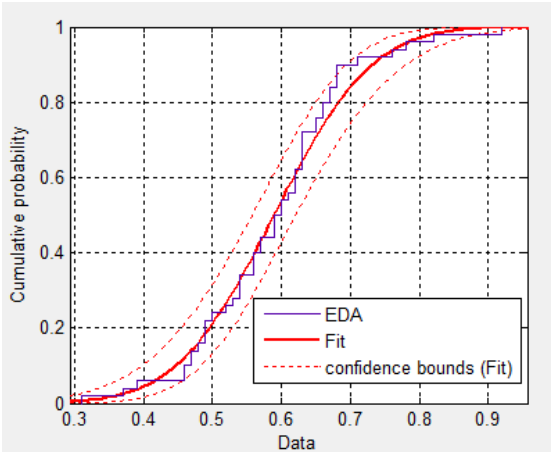

Figure 3.12:CDF display(dehydration level4)

These are the figures from step 4. I choose 0.660 as the upper limit of dehydration level 4.

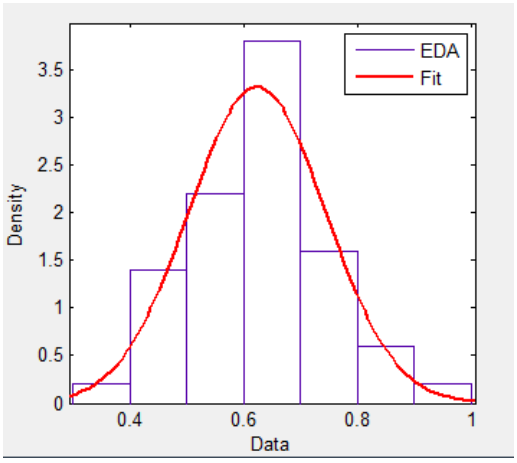

Figure 3.13: Distribution fitting figure (dehydration level 5)

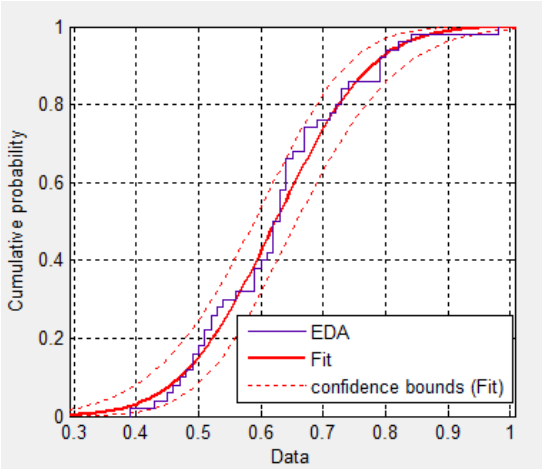

Figure 3.14 :CDF display(dehydration level5)

These are the figures from step 5. We can see the change of upper limit value is only 0.04 compared to the last level. That means that although people still drink some

more water, their EDA data don't change a lot. So the dehydration at level 5 could be the highest level which means people in this level do not need to rehydrate any more.

Here are the five dehydration levels:

Level 1: EDA from 0.0 to 0.155

Level 2: EDA from 0.156 to 0.320

Level 3: EDA from 0.321 to 0.555

Level 4: EDA from 0.556 to 0.660

Level 5: EDA above 0.661

3.2.3 User feedback

In order to see if this dehydration standard works well, after the system's toast window - which shows the user which dehydration level he was in, the user can type in which level he thought he was in based on his feeling.

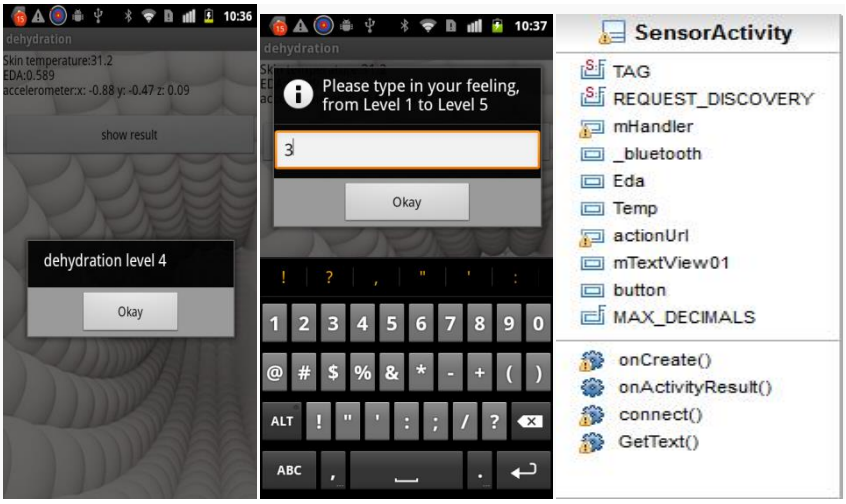

Figure 3.15:Sensor activity page

3.2.4 data storage and web server

After each test, the system will send all the information collected to a simple web server. This information includes: EDA data, skin temperature, measured dehydration level, the user's feedback and the timeline.

In order to send data to the web, we need to implement the sever side and the client side. The Android Smartphone is considered a client. I created an HTTP Client and an HTTP Post, then bundled the post parameters' pair with key and value. Before making an HTTP request, I encoded the post data in order to convert all the string data into a valid URL format. Finally, I executed an HTTP Post using the HTTP Client created before. For the server side, I wrote some PHP code to receive this data from the Smartphone and insert it into the database. If the measured value is equal to the feedback value, it means it is a matched result. If the measured value and feedback value are within 1 level distance, it means it is an approximate matched result. On the sever side, there is a webpage to show this data and keep updating the matched rate and approximate matched rate.

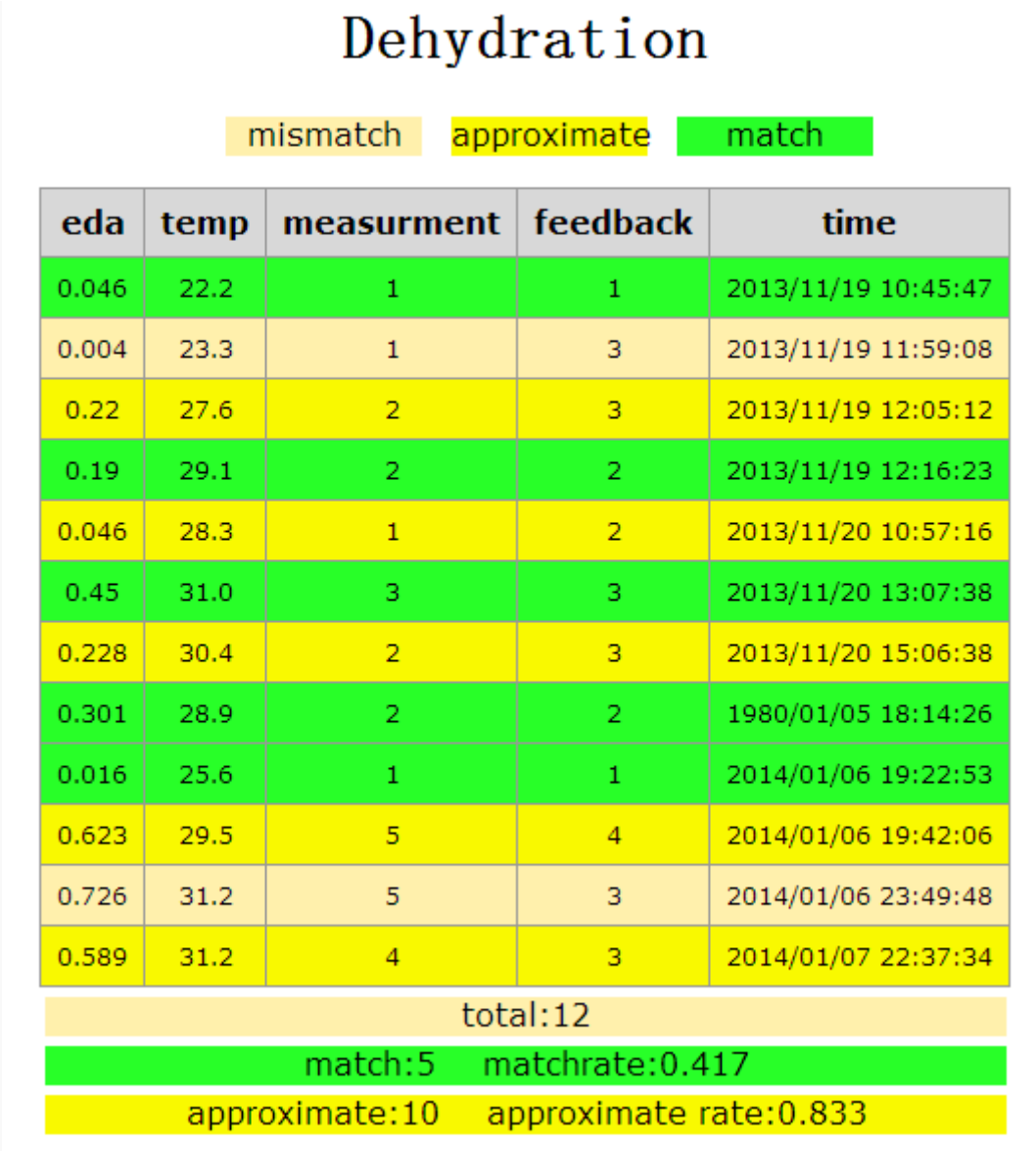

Figure 3.16: Webpage

If the test has a matched result, the background color of that test's information is green. If the test has an approximate matched result, the background color is yellow.

3.3 Active mode implementation

3.3.1 Information collection

The implementation of Active mode can be divided into two parts: one is user basic information collecting and another one is energy expenditure calculation. Basal Metabolic Rate is determined by people's gender, age, weight and height. This information needs to be provided by users, here is the activity page for collecting this data.

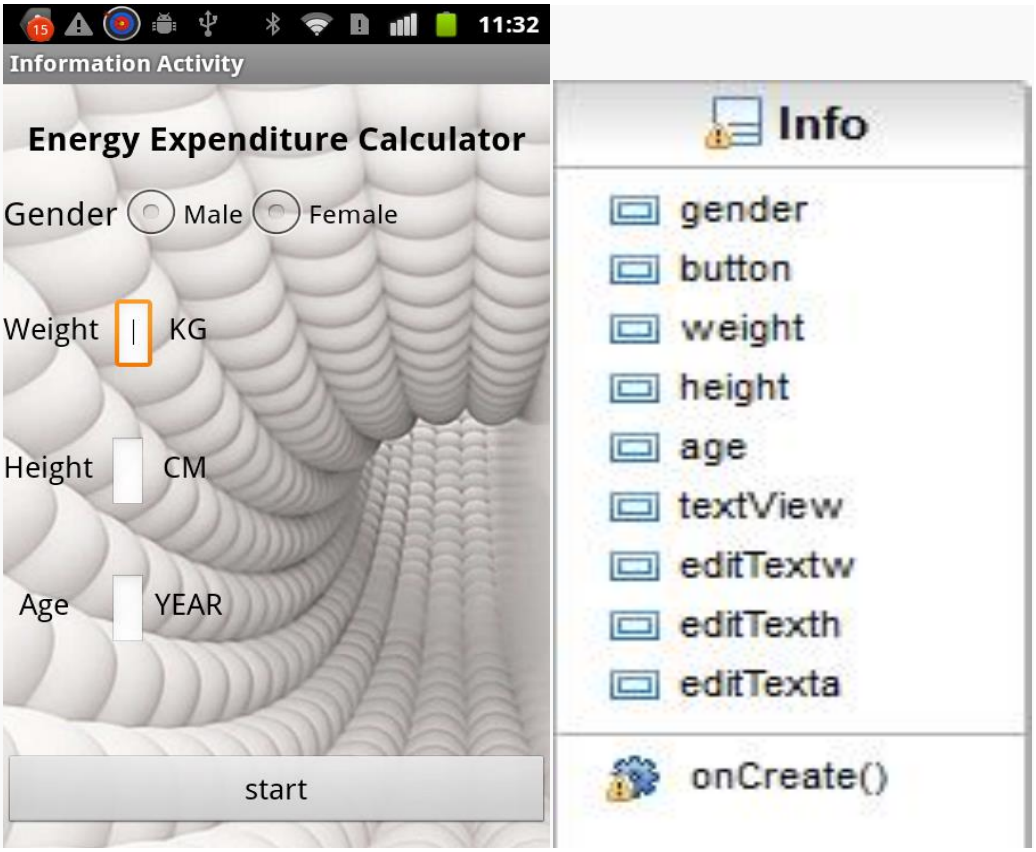

Figure 3.17:Information colleting page

After the user types in all this information, they can click the start button to go to the next step. This data will send to another activity which will do the calculation work. I used final class Bundle, which can insert a String value into the mapping and

replace any existing value for the given key to transfer the data.

3.3.2 Energy expenditure calculation

There are two algorithms that can calculate energy expenditure, one is based on tri-axis accelerations and another one is based on metabolism science.

3.3.2.1 Tri-axis approach

The basic idea of this approach is based on classical Newton's Mechanics.

$E=\frac{1}{2}\mu * mg\int_{a_1}^{a_2} da \int_{t_1}^{t_2} tdt$

In order to implement this calculus calculation, I need to divide time into very short periods. During this period of time, I considered the user's physical activity as a uniformly accelerated motion. Since the Smartphone's accelerometer is 100Hz, I divided 1 second into 100 periods of time. For each period, I used the following formula to calculate energy expenditure.

$\texttt{Calories} = \frac{1}{2}\mu * mg(a_2 - a_1) * ((\frac{1}{2}t_2^2 - \frac{1}{2}t_1^2) + v_1 * 0.01) / 4.18$

In this formula, $a_1 - a_0$ is equal to $\int_{a_1}^{a_2} da$, v_1 is the initial velocity of this period of time. Because the unit of kinetic energy is Joule, which is equal to 4.18 Kcal, we need to divide the value by 4.18 to compare with the result showed on a running machine, which uses the calorie as a counting unit.

3.3.2.2 Metabolism approach

The metabolic rate determines how much energy people consume. BMR, RMR BSA and T(second) decide energy expenditure.

E=(RMR +1.2) x BMR x BSA x T (second)

BMR, BSA and T are decided by the user's information but RMR is different from these activities. GM is the mean factor of RMR. If we only consider running, these two values are linearly dependent. Usually RMR is equal to five times GM, so the formula change to the following one:

E=(5*GM +1.2) x BMR x BSA x T (second)

3.3.4 Water loss based on energy expenditure

Water loss is determined by energy expenditure, BSA and RMR. The unit of water loss is milliliter.

Water loss = E x BSA x RMR / 10;

Although dehydration only affects performance in workouts lasting longer than an hour, hydration during workouts is highly recommended. The general rule of thumb for fluid consumption during runs is to take in 150 ml to 200 ml of fluid every 20 minutes. The test of this system is based on running with a speed of 10 km/h, once the user has lost 300 ml water, the system will remind the user to hydrate. Since there are two approaches to calculate the energy expenditure, I used the mean of these two

methods for the final result.

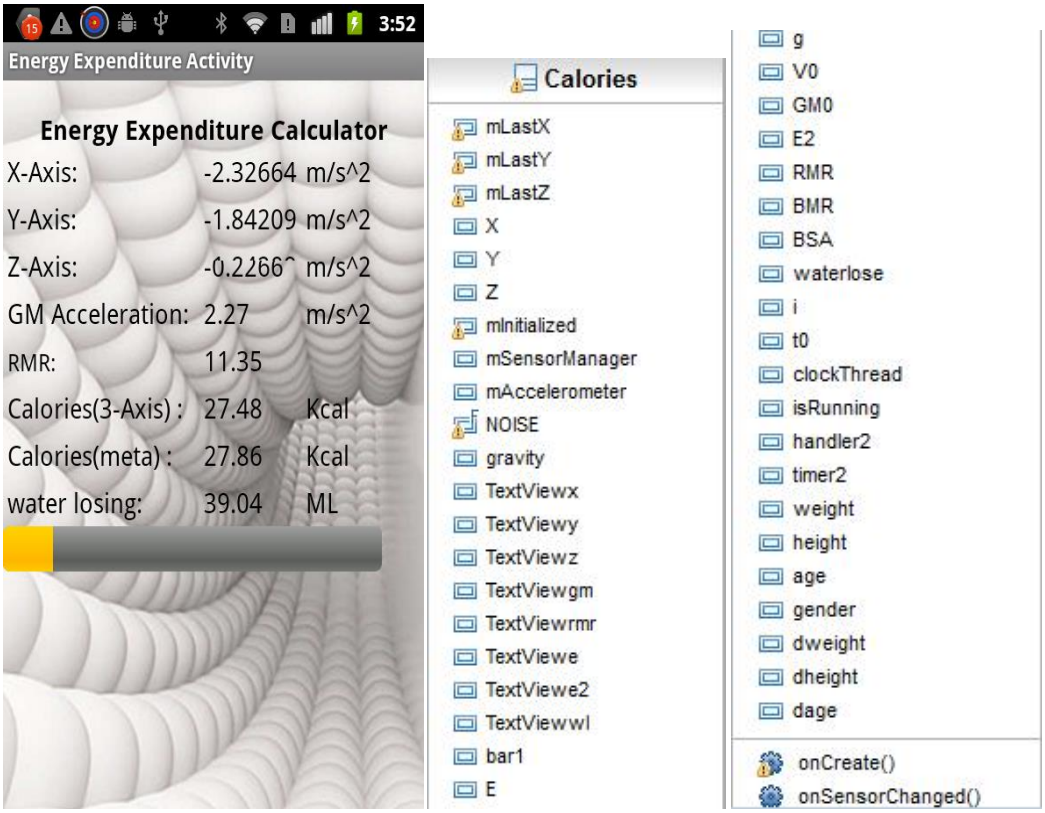

Figure 3.18:Energy Expenditure activity

This activity displays real time tri-axis accelerations, GM acceleration, RMR, calories burned and water loss amount. Tri-axis accelerations update every 10ms. Since GM and RMR are based on tri-acceleration data, they will also update every 10ms, which is a very short time and the user cannot check their GM and RMR clearly. I used a thread that will sleep 1 second at first then send a message to a handler, which is bound to the thread. From that point on, this handler will deliver messages and runnables to that message queue and execute them as they come out of the message queue. The handler we created will calculate the average GM and RMR

of one second and display them on that page. It will also calculate the energy expenditure in this second and update the total energy expenditure. The progress bar shows the amount of water loss more intuitively. The maximum value of this progress bar is 300, once the progress reaches the end means this user has already lost 300 ml water and then there will be a toast window to remind the user to hydrate.

3.3.3 Test

Three participants used this system and ran on treadmills that show calorie expended on their screen. Here are the participants' information:

	Gender	Weight (KG)	Height (CM)	Age (Year)	Speed (km/ h)
Participant A	Male	72	175	24	10
Participant B	Male	62	173	23	10
Participant C	Female	56	165	23	10

Table 3.1 Information of participants of Running test

After the participants start running, this system starts to calculate their energy expenditure and the amount of water loss. Once the amount of water loss reached 300 ml, they stop running and check the result both on the treadmill's screen and the phone screen. Each of participants took the same procedure five times.

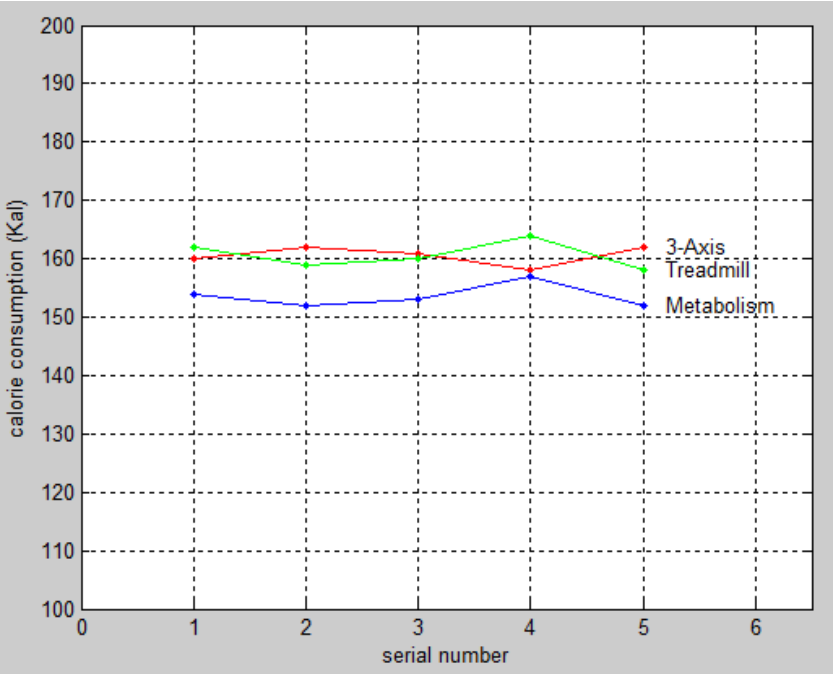

Figure 3.19: Data of Participant A

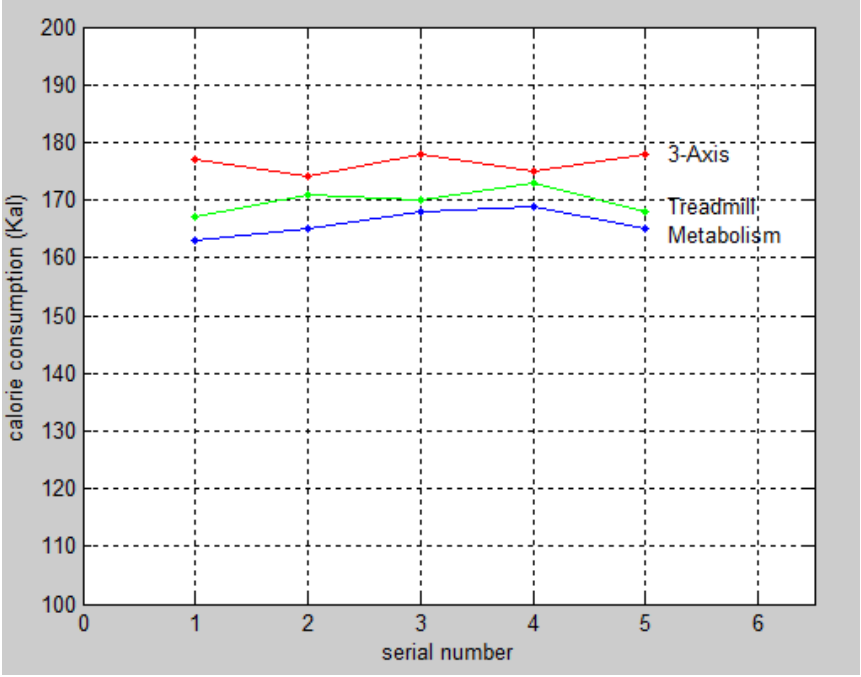

Figure 3.19: Data of Participant B

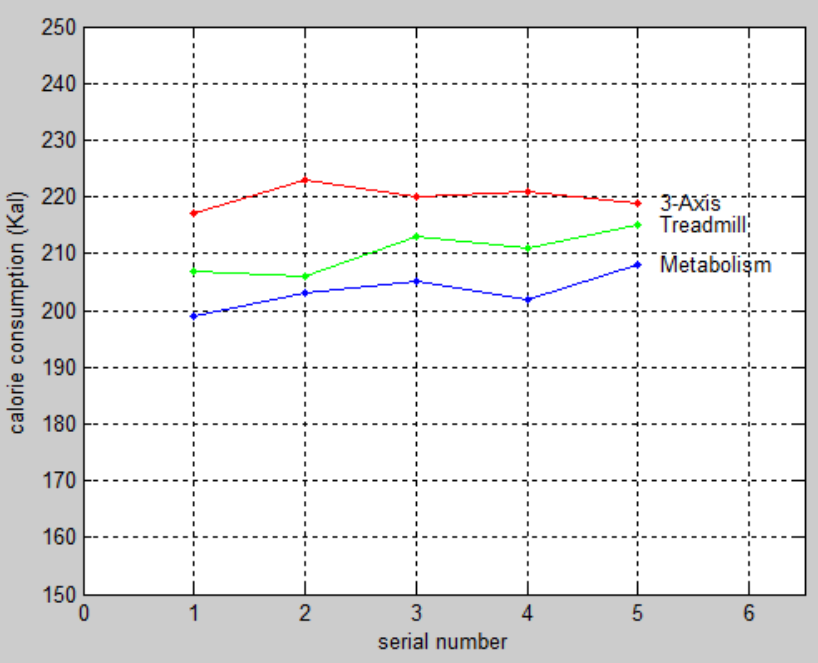

Figure 3.19: Data of Participant C

The data from treadmill, which is the green line in these figures, can be treated as the ground truth of energy expenditure during running. The results from both of the methods in this system are close to the reading of the treadmill. The average error of tri-axis acceleration approach is 3.0% and the average error of metabolism approach is 3.3%.

Recall those formulas I mentioned before:

$E=\frac{1}{2}\mu * mg\int_{a_1}^{a_2} da \int_{t_1}^{t_2} tdt$ (tri-axis) (1)

E=(RMR +1.2) x BMR x BSA x T (second) (Metabolism) (2)

Water loss=E x BSA x R/10 (3)

Since all the participants were running with the same speed means their GM is close to each other's. Participant C is a girl who is in smaller shape as compared to the other two participants and energy expenditure is positively correlated with weight in formula 1 and also positively correlated with BMR and BSA in formula 2. So in unit

time, participant C's energy expenditure is the smallest. Since the amount of water loss is positively correlated with BSA, in order to lose the same amount of water, participant C needs to expend more energy.

4 .Future work

This project has a large amount of possible future work both in calm mode and active mode. Some more work also can be done in web server side.

For calm mode, I just used EDA data and skin temperature as the two only factors to judge the dehydration level. With the Smartphone technology developing, more and more types of sensors are used in a Smartphone. The new generation Android Smartphone has an already installed temperature sensor, which can detect environment temperature. If the temperature sensor can be used in this system, the system can compare the environment temperature and skin temperature, which will definitely improve the accuracy. Some other sensors like humidity and light may also help to enhance the system's performance.

For active mode, all the tests and results are only based on running. Besides running, this system can also apply to other sports. More experiments and tests need to be done to find the relation between GM value and RMR value in other sports.

For the server side, this system only stores and displays calm mode data and the result. The active mode may also send data to the server. Besides that, since user's BMR and BSA are different, this system may also create personal data records for the user's convenience. On the Smartphone side, the user needs to create their own account when they use this system for very first time. And on the sever side, the user needs to log on to their account and then can check their historical data. All this work can be done to potentially enhance the system's performance and improve the user experience.

5 .Conclusion

This project builds an application on Android Smartphone, which can help a user to rehydrate in time. This application uses EDA and body temperature data that are collected from an Affectiva Q sensor to estimate the user's dehydration level. On the other side, this application also uses tri-axis acceleration data from the built-in accelerometer and the knowledge of metabolism science to calculate the amount of the user's water loss when they are running, and then reminds the user to rehydrate even when they have not realized their body is lacking water. This application is easy to use and both functions in this application allow for easy extension and maintenance.

Reference

1. ZHU Guozhong, WEI Caihong, PAN Min, "The Research of Energy Expenditure DetectionAlgorithm Based on Tri-Axial Acceleration Transducer", CHINESE JOURNAL OF SENSORS AND ACTUATORS, vol. 24, No. 3, Chapter 24, Aug. 2011.

2. Ceaser, Tyrone Gene, "The Estimation of Caloric Expenditure Using Three Triaxial Accelerometers. " PhD diss., University of Tennessee, 2012. http://trace.tennesse.edu/utk_graddiss/1514

3. Lawrence E. Armstrong PhD, FACSM, " Hydration Assessment Techniques ", Nutrition Reviews, Vol.63, Issue Supplement s1, pages S40-S54, June 2005

4. Björn Ekblom, Carol J. Greenleaf, John E. Greenleaf, Lars Hermansen, "Temperature Regulation during Exercise Dehydration in Man", Acta Physiologica Scandinavica, Vol.79, Issue 4, pages 475-483, Aug 1970.

5. BMR formula, http://www.bmi-calculator.net/bmr-calculator/bmr-formula.php

6. LIU Jidong, "Sports and Energy", available athttp://www.docin.com/p-594891400.html

7. Acceleration and Energy,http://www.artificial-gravity.com/Dissertation/3_4.htm

8. Affectiva, Q Sensor 2.0 Datasheet, 2012, http://www.affectiva.com/download/Affectiva_Q_User_Manual.pdf.

9. Android Sensor Overview, 2012**,** http://developer.android.com/guide/topics/sensors/sensors_overview.html

10. Bluetooth Connectivity,http://developer.android.com/guide/topics/connectivity/bluetooth.html

11. Intents and Intent Filters, available at http://developer.android.com/guide/components/intents-filters.html.

12. Android Interface Definition Language (AIDL), available at http://developer.android.com/guide/components/aidl.html

13. Android Handler,http://developer.android.com/reference/android/os/Handler.html

14. Android Http Request,
http://developer.android.com/reference/org/apache/http/HttpRequest.html