

Self-Orthotherapy program using Arduino and pressure sensor

Hakjun Kim
Information System in HYU
Email: canon3478@gmail.com

Minsoo Choi
Information System in HYU
Email: choiza03@gmail.com

Taejin Kim
Information System in HYU
Email: o___xo@naver.com

Group Name : 'Healpers'

Role	Name	Task description and etc.
User	Minsoo Choi	<ul style="list-style-type: none">- Test the application in various postures.- Require bugs to be fixed.- Demand the all of data about user.- Require additional functions.
Software Developer	Taejin Kim	<ul style="list-style-type: none">- Develop the application using Arduino and pressure sensor.- Fix bugs.- Develop additional functions according to users' needs.
Development Manager	Hakjun Kim	<ul style="list-style-type: none">- Monitor the progress of the project.- Make a overall plan about developing program and managing the budget- Reflect users' needs

Abstract—This project is about self-orthotherapy system using Arduino with a built-in bluetooth 4.0 communication chip and pressure sensor which communicates with an Android application. This program can definitely detect users' incorrect postures and can be used for long-term postural analysis and correction as saving users' postural data in database.

Keywords—Self-orthotherapy, Arduino, Pressure sensor, Android application

I. INTRODUCTION

In modern society, most office workers spend over 12 hours each day sitting on a chair. This immoderate sedentary lifestyles and incorrect posture can cause many severe health problems. For instance, incorrect posture can lead to turtle neck syndrome, scoliosis, chronic low back pain, cervical disc and herniated disc. In particular, crossing legs can twist the pelvis which induces pelvis inflammation and may change the shape of spine and trigger crooked legs, all of which are the reason for an asymmetrical

body. In other words, correct posture will be the most effective way to prevent the above diseases.

Therefore, maintaining a good posture in a forced sedentary lifestyle has a great significance to keep them healthy. Nevertheless, when office workers concentrate on working or loosen the tension, posture would be disturbed. So we would like to develop an application to help them correct their wrong postures.

The application can recognize the position by using Arduino and pressure sensor. The application will perform the correction of posture immediately by informing the users incorrect postures. This application is expected to greatly contribute to maintaining a healthy posture and lifestyle of the modern office workers.

II. REQUIREMENTS

A. Required hardware and software

- Arduino Uno R3 Board
- FSR 406(Pressure sensor)
- Connecting pin
- Bread Board
- HC 06(Bluetooth Module)
- USB cable
- Arduino Sketch(Using C)
- Android Studio(Using Java)
- MySQL

B. Requirements for Developers

- Making pressure recognizing cushion

First, we have to make proper pressure recognizing cushion to recognize users' posture. So, we are going to make cushion with pressure sensors fitted to the chair's size. Now, we have plan to use 'FSR 406(Pressure sensor)' that the size is approximately 38mm x 38mm. So, we are going to make cushion with four(2x2) pressure sensors or more to recognize users' posture properly.

- Connecting pressure sensor with Arduino Board

Second, we have to connect pressure sensor with Arduino Board to get pressure value. And we have plan to use 'Arduino Uno R3 board'. So, we are going to connect 'FSR pressure sensor' with 'Arduino Uno R3 board' and test whether we get proper pressure value when user changes the posture.

- Making orthotherapy algorithm using pressure value

If we succeeded in getting proper pressure value, we make our own orthotherapy algorithm. We will analyze pressure values and set cases to correct users' posture. For example, when the right side's pressure value is low we show alert message ("Your body is tilted to the left" or "Don't let your right leg crossed").

- Developing Front-end Android Application

After we make orthotherapy algorithm we have to develop android application. So, we will develop the application according to our UI/UX and architecture design. We have two parts in our front-end application. The first is the Self-Orthotherapy part and the second is showing analyzed result using users' personal data.

- Constituting Database for saving users' data

We should design database tables for saving users' data. And we make sign-up form for user to register in our application. And we get essential data such as ID, PW, Phone number and put it into the our database.

- Developing Server to manage users' data

We will provide analyzing service using users' personal data. So, we have to manage users' own Id and Password by server and save the users' specific data. For example, if the user got some alert messages or moves out the users' seat, we save that data and we will provide graph or chart by analyzing that data.

C. Requirements for Users

- Getting alert message when having bad posture

If user had bad posture, program recognizes that and shows the alert message to fix that posture. Bad posture means i) sit with one's legs crossed. ii) the waist is bended iii) the body is tilted to right or left too much.

- Approaching users' personal data

User needs authority to approach to own personal data. So, user should sign-up for our application. After that, user could check own personal data any time. For instance, how many times the user sit with right leg crossed today or how long the user left the seat during the work concentration time.

D. Requirements for Application

- Sign Up

This application uses the ID to the user by storing the time information of the server to be able to store and view the information, if not the own device. The user can easily sign up only your name, E-mail, if you ever forget your ID and Password can be easily found via E-mail.

- Log In / Out

When running this application, user has to log-in in order to record the time sitting on the cushion on a server. But, to shorten the log-in procedure, user can log-in easily after log- in once because the information of ID and password is saved. If the User wants to log-out, user can logs out in the Setting Page.

- Statistics

The application stores the users' data on the server. The application stores the users' data on the server. Then this application produces statistics

on a weekly, monthly, yearly. These statistics will help determine whether seeing how much work they have kept for themselves some attitude.

- Data Visualization

User allows to easily identify their own data through the application using the data visualization. The application will use the visualization techniques, such as Android Chrono-meter, Chart, VideoView.

- Guide to Good Posture

This application shows the right postures using some kind of videos and images and introduces stretching and exercises which are good to correct wrong posture.

- Suggest to Rest

If user overworks, this application will show the message likewise "Take a Break! Do stretch!".

- Push Alarm

If user keeps wrong posture over certain time, this application will warn to the user about that wrong posture by push alarm.

- Customize Setting

User can set the status of getting push alarm(On/Off) and the time when the user gets push alarm while the user keeps wrong posture. And the user could use "Initializing" function. "Initializing" means the user's own pressure values saved in the database (when user does 'normal' posture) are initialized.

- Provide Scores

Users can grasp how well they have sat for today as this application provides today's posture score. Users can also check whether their postures are improving or not.

- Body Balance Measurements

Users can measure their current body balance status by taking a photo of their upper body. If, users take picture by using this function, the program would offer the information about how much their body is twisted.

III. DEVELOPMENT ENVIRONMENT

A. Choice of software development platform

1. Which platform and why?

- Windows for android application developing

Our target is office workers who want to correct their sitting posture. For ortho-therapy, it is the most important and efficient to maintain correct posture intentionally. Considering this, developing mobile application is appropriate rather than either web or computer program operated in window OS because time with mobile phone is longer than time using computer. If there is only our posture-correcting cushion, users can notice that they sit in incorrect posture by push alarm at anytime and anywhere, even while watching TV and reading a book at home. And this data is saved in server automatically so they can check the statistics how long they were sitting and how many times they sat incorrectly. Because Android OS is being used much more than IOS, as the below Figure 3.1 is showing, we develop our application in Android application.

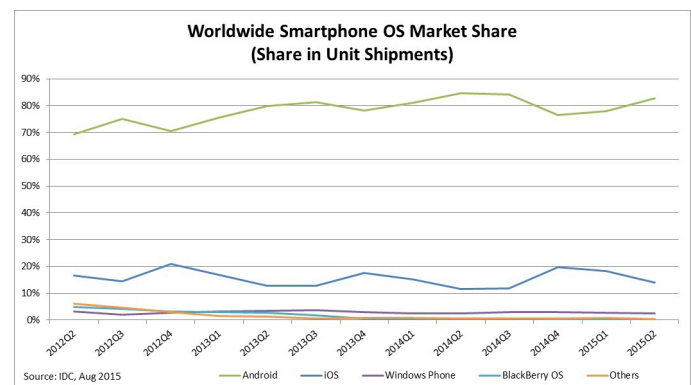


Fig. 1. Worldwide Smartphone OS Market Share

2. Which programming language and why?

- Java for android application developing

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. As of now, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired

by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them. And Java is the best suitable language to develop the application in android OS.

– C for IDE

We should do many tests to recognize various users' postures using pressure values getting from FSR-406 pressure sensors. So, we use the open-source Arduino Software (IDE) and the best proper language to use this program is C. The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board.

– PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose scripting language that is especially suited to web development. PHP is fast, flexible and pragmatic language. And PHP makes the user write dynamic web files because PHP follows many sentence forms used in C, Java, Perl etc. And there is excellent documentation we can refer to. <http://www.php.net/> has really good documentation with helpful user-contributed gotchas and code snippets. We can also find tons of tutorials, etc. around the web.

3. Provide a cost estimation for your built.

– Cost for Server :

1 year for free. And after 1 year, there will be additional prices. We predict maybe about 1,000 people will use our service, and DAU (Daily Activity User) will be 300 around. So we will use t1. micro instance (AWS), and its prices are about 30 dollars per month: So may be there will be additional 360 dollars per year.

– Cost for hardware

1) Arduino

The Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

2) Pressure Sensors

The model 406 FSR is a single-zone Force Sensing Resistor optimized for use in human touch control of electronic devices such as automotive electronics, medical systems, and in industrial and robotics applications. FSRs are two-wire devices. They are robust polymer thick film (PTF) sensors that exhibit a decrease in resistance with increase in force applied to the surface of the sensor. It has a 39.6mm square active area and is available in 4 connection options. Interlink Electronics FSR 400 series is part of the single zone Force Sensing Resistor family.

3) Bluetooth Module

HC-06 is a serial port module composed in SMD form which is possible to be replaced easily. It also uses the connection between the bluetooth Master module in either PC or mobile device and embedded system in substitution of the serial port.

Arduino receives the input values, which the four pressure sensors perceive, through the breadboard and sends those values to the Android Application via bluetooth communication with the mobile phone. The system flow is shown in Figure 3.2.

Therefore, we need 4 pressure sensors, Arduino, bluetooth module, bread board, USB cable and jump wire MM. The specific products' cost is shown in Table 3.1.

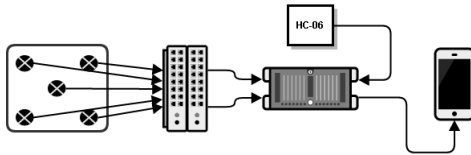


Fig. 2. System Flow

TABLE I
THE SPECIFIC PRODUCTS' COST

Products	Prices	Quantities	Costs
Pressure Sensor (FSR-406)	14,000	4	56,000
Arduino Uno (R3)	7,500	1	7,500
Bluetooth Module (HC-06)	6,000	1	6,000
Bread Board	4,500	1	4,500
USB Cable	500	1	500
Jump Wire MM	70	50	3500
Total			78,000

4. Provide clear information of your development environment. (e.g., version of software, OS version, your computer resources)

- Android develop
 - 1) Windows : Windows 10
 - 2) Android Studio 1.5.1 / Android 4.4.4 Kitkat
- Arduino develop
 - 3) Windows : Windows 10
 - 4) Arduino 1.6.8 (IDE)

5. Using any commercial cloud platform (e.g., Amazons EC2) is definitely a BONUS.

We will use Apache Web Server because it is the world's most widely used web server software. As of June 2013, Apache was estimated to serve 54.2 percent of all active website and 53.3 percent of the top servers across all domains. (The most important reason is we have already apache web server.)

- Apache Web Server has some features:
- easy and fast customizing using module.
 - It can handle many traffic easily.
 - It can control web server more delicately

- It is tested enough, so it is very stable.

B. Software in use

- Footlogger

3L-Labs, internal venture corporation, developed a smart wearable device 'FootLogger'. It is insoles with some pressure sensors which analyze users' steps. This device can be used for health care, sports and entertainment.

Healthcare

- Monitors recovery after surgery
- Assesses balance of diabetics
- Allows for monitoring of prescribed exercises
- Monitors safety and activity of senior citizens
- Activity tracking (calories, distance, time)
- Early prediction of dementia and spinal disease
- Early prediction of accidents from falling
- Monitors rehabilitation (stroke, paralysis)

Sports

- Can monitor distance traveled
- Monitors golf stance and gives coaching
- Marathon/jogging aid
- Out-toeing/in-toeing gait identification, provides gait coaching
- Kids' correct posture coaching
- Can be applied to every sport

Entertainment

- Daily log (standing, sitting, walking)
- Personality analysis
- Daily mood analysis
- Smartphone game input device

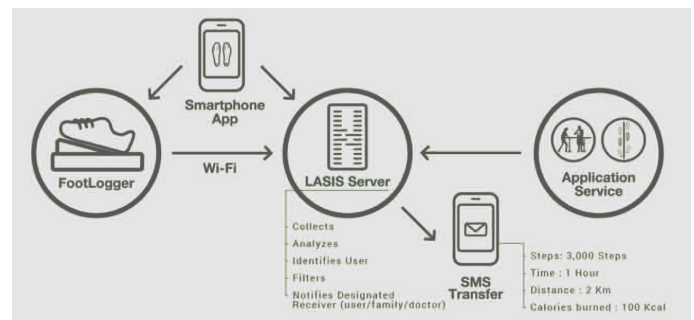


Fig. 3. Footlogger

C. Task distribution

TABLE II
TASK DISTRIBUTION

Role	Name	Task description and etc.
User	Minsoo Choi	<ul style="list-style-type: none"> – Test the application in various postures. – Require bugs to be fixed. – Demand the all of data about user. – Require additional functions.
Software Developer	Taejin Kim	<ul style="list-style-type: none"> – Develop the application using Arduino and pressure sensor. – Fix bugs. – Develop additional functions according to users' needs.
Development Manager	Hakjun Kim	<ul style="list-style-type: none"> – Monitor the progress of the project. – Make a overall plan about developing program and managing the budget – Reflect users' needs

value for recognizing person's changing posture. The dimensions of FSR-406 are shown in Figure 4.2

TABLE III
DIMENSIONS OF THE CHAIR AND CUSHION

Width of the chair	45cm
Height of the chair	42cm
Depth of the chair	47cm
Width of the cushion	30cm
Height of the cushion	30cm

IV. SPECIFICATIONS

A. Modeling for Specifications

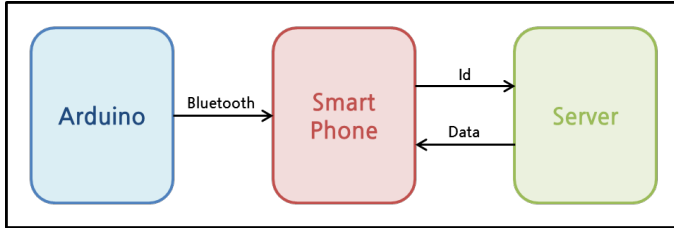


Fig. 4. Basic structure

Arduino receives some of the pressure value from the pressure sensor. And arduino sends the values to smart phone using HC-06. HC-06 is a Bluetooth Module. Then smart phone application determines posture of the user by analyzing the values. After that, smart phone calculates the time for holding the posture. The data of time will be sent to server, and saved at server. At the same time, the application will receive the user's whole data from server. Then, the application will display the data by data visualization.

B. Posture Recognition using pressure sensor

We made a kind of cushion for measuring pressure force. The dimensions of the chair and cushion are shown in Table 4.1. And we used FSR-406 pressure sensor model to get proper pressure

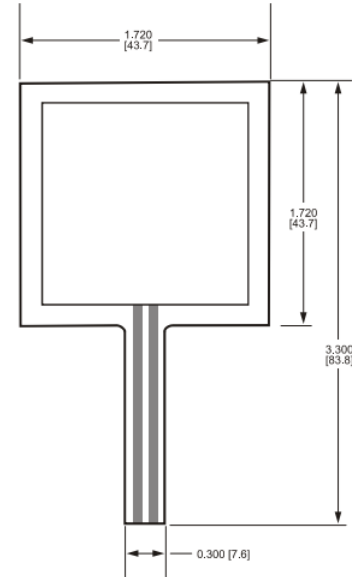


Fig. 5. Dimensions of FSR-406

The reason why we choose the FSR-406 is the specifications of FSR-406 are suitable to make the posture recognition cushion. This sensor is very thin and hence can be easily bent to fit, so a user can sit down on the seat comfortably. Since the sensor sheet is light and thin, it can be carried anywhere and used in various situations in daily life. And the active area is appropriate to get pressure value when person changes his or her posture. FSR-406 specifications are shown in Table4.2.

TABLE IV
FSR-406 SPECIFICATIONS

Thickness	0.018"(0.46mm)
Active area	38.1mm x 38.1mm
Connector	AMP Female connector
Minimum load	0.1N
Maximum load	10N

FSRs are two-wire devices with a resistance that depends on applied force. For specific application needs please contact Interlink Electronics support team. An integration guide is also available. For a simple force-to-voltage conversion, the FSR device is tied to a measuring resistor in a voltage divider configuration. This is shown in Figure 4.3. The output is described by the equation:

$$V_{out} = \frac{R_m V +}{(R_m + R_{FSR})}$$

In the shown configuration, the output voltage increases with increasing force. If R_{FSR} and R_m are swapped, the output swing will decrease with increasing force. These two output forms are mirror images about the line $V_{out} = (V+)/2$. The measuring resistor, R_m , is chosen to maximize the desired force sensitivity range and to limit current. Depending on the impedance requirements of the measuring circuit, the voltage divider could be followed by an on-amp. A family of FORCE vs. V_{out} curves is shown on the graph above for a standard FSR in a voltage divider configuration with various R_m resistors. A $(V+)$ of +5V was used for these examples.

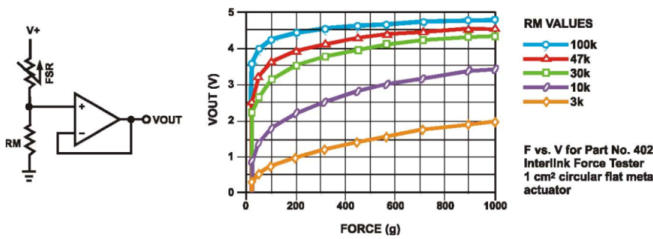


Fig. 6. Typical Schematic and force curve of FSR-406

To recognize difference of sitting posture, we have to make the cushion with proper number of pressure sensors. So, we arranged five pressure sensors to the

five points in the cushion. The five points are 'Left-front', 'Right-front', 'Center', 'Left-back', 'Right-back' and this classification is shown in Table 3. And the pressure values are transmitted to Arduino board through bread board. The pressure sensors' spot position is shown in Figure 4.4.

TABLE V
THE CLASSIFICATION OF PRESSURE SENSOR'S POSITION

(LF)	Left Front
(RF)	Right Front
(C)	Center
(LB)	Left Back
(RB)	Right Back

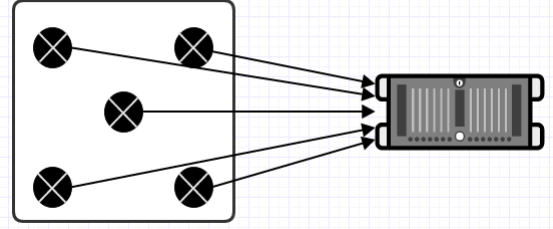


Fig. 7. Pressure sensors' spot position

C. Posture classification

In this part, we describe a way of classifying sitting postures. Posture classification involves three basic issues. One issue is person dependency. Different persons would obviously sit in different ways. Even the same person might sit on different parts of the seating face. One ways to resolve this problem is to normalize pressure distributions by standardizing different postures when workers sit in the chair. The classification of seven postures is shown in Table 4.4.

TABLE VI
THE CLASSIFICATION OF SEVEN POSTURES

(N)	Normal
(F)	Leaning Forward
(B)	Leaning Backward
(L)	Leaning Left
(LC)	Left leg crossed
(R)	Leaning Right
(RC)	Right leg crossed

Second issue is the time dependency. We have to push alarm to users when they are doing wrong postures. So, it is needed to decide when we do alarm function. We set time standard from 1 minute to 10 minutes. User could select time when user gets push alarms while user keeps wrong posture. And the last issue is weight dependency. The sensor values would increase linearly or non-linearly with an increase in weight. Therefore, it would be effective to divide the sensor values by the total value. And the each user's pressure value is different when user does 'Normal' posture. So, if user registers in our application, the user's initial pressure values are saved in the database and our algorithm uses each user's data to recognize posture. And we offer the function which initializes user's weight in the application.

D. Specification for front-end application pages

– Log-In Page

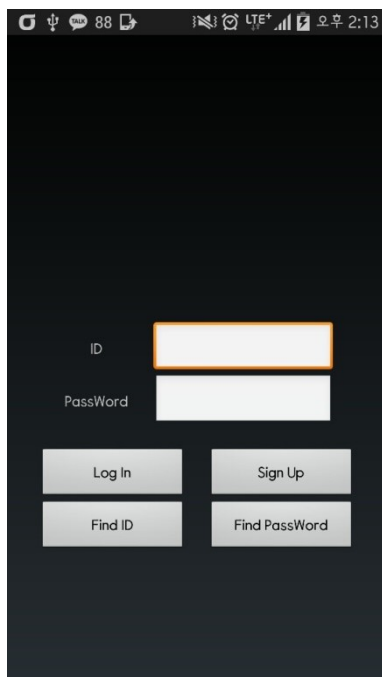


Fig. 8. Log-In Page

This page is the first page of the application. To import the posture data, you have to log in first. Your working posture records will be saved in your

server with your ID. If you do not have an ID, you can click the Sing Up button. Or, if you forgot your ID or password, you can find it using Find ID button or Find Password button. Once you log in, your ID and password will be automatically saved. Even when you log in via another phone, you can open your records with ease since your posture records are imported from your ID only.

– Sign-Up Page

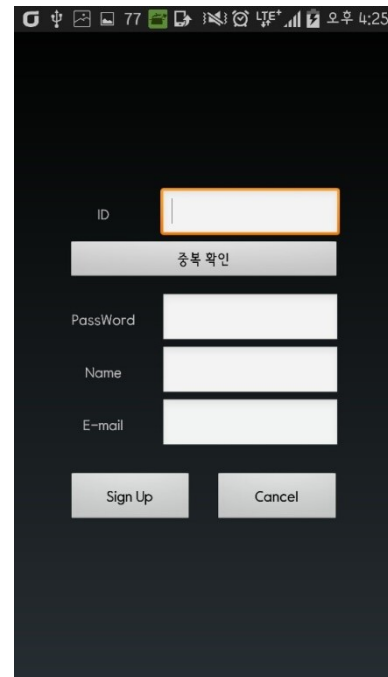


Fig. 9. Sign-Up Page

This page is where a user with no ID can create a new ID. Since ID will be the Primary Key of the User DB that will be saved in the server, it should not be overlapped. So, it is mandatory for you to double-check your ID. There is no limit in the length of the password and the password will be replaced with other letters, such as ****, as soon as you enter your password for the sake of absolute confidentiality.

Name and E-mail will be used for discerning between members and non-members. And, they will also be used for finding ID or Password in case you forgot it.

– Find ID Page

Once you enter your E-mail that you entered during Sign-up process, it tells your ID on the screen.

–Find Password Page

Once you enter your E-mail that you entered during Sign-up process, it sends your password to your E-mail.

–Now Page

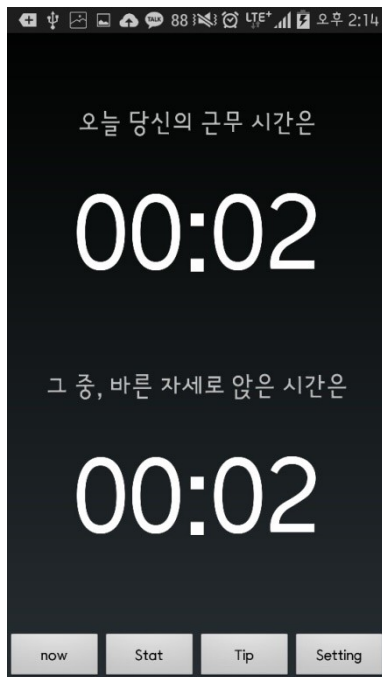


Fig. 10. Now Page

This page is the first page you see after you log in. There are 4 tabs in the tab bar below. 'Now' tab is the tab of the current page. If you click the 'Stat' tab, you can check your accumulated statistics on daily, weekly, monthly and yearly basis. 'Tip' tab is where you can find images and videos of stretches or exercises useful for correcting posture. The last 'Setting' tab is for setting few functions related to this application.

In this page, the app counts the number of hours you sit on the cushion that is composed of pressure sensor. This data will be exported on the screen in order for you to check your total working hour. (Upper Timer)

In specific, it counts the number of hours of your correct posture and bad posture separately. So, it is possible to check the number of hours of correct posture only. (Lower Timer)

The number of hours exported from this page will be saved in the server as well. Moreover, once the user maintains a bad posture for a long time, the user will receive a push notification that warns his/her long-time bad posture.

–Stat Page

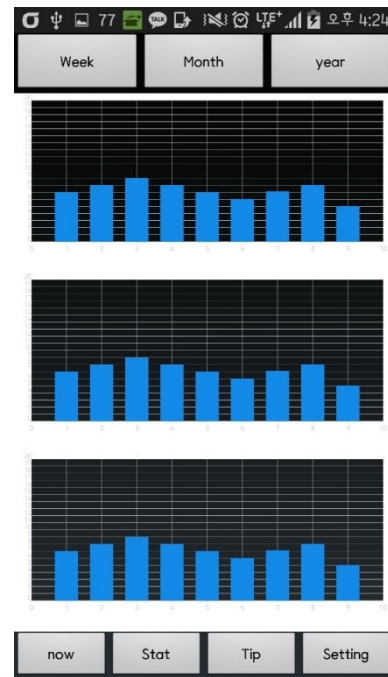


Fig. 11. Stat Page

This page shows up as you click the Stat tab below. It is divided into 3 big graphs and has three tabs at the top of the screen. The graphs of this page are composed of individual time data saved in the server. Each of the three graphs represents total working hour, the number of hours of correct posture and the bad posture. The x-axis of this graph changes according to the upper tab of the screen. The x-axis consists of 7 days in the 'Weekly' tab, 30 days in the 'Monthly' tab and 12 months in the 'Yearly' tab. The y-axis of every tab consists of numbers from 0 to 24, which refer to the number of hours.

The first graph will be presented as a Bar Chart that shows working hour of the days of the week

in the 'Weekly' tab, daily working hour in the 'Monthly' tab and monthly average working hour in the 'Yearly' tab. With the same principle, the second graph shows the number of hours of correct posture for each tab whereas the third graph shows the number of hours of the bad posture.

And user could see the score measuring by this program. The score means how well users have sat for today. The score is from 0 to 100 and 100 is the maximum score which means the users posture for today is perfect.

-Tip Page

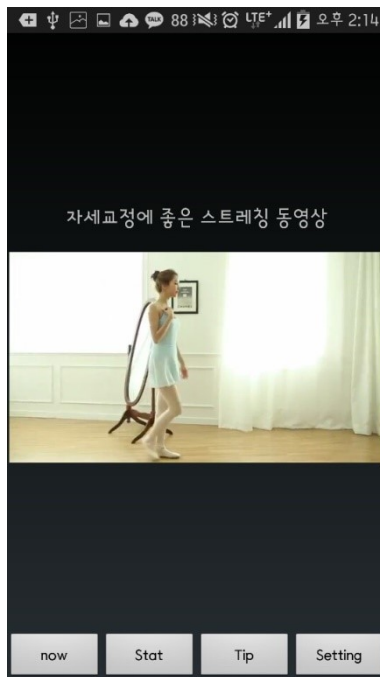


Fig. 12. Tip Page

When clicking the Tip tab below this page, this page will show up. It exports images and videos of stretches or exercises useful for correcting posture as a customized ListView. The function of playing or pausing the video takes the similar format with Youtube. The first click plays the video and the second pauses it.

-Setting Page

This page shows up when you click the 'Setting' tab below this page. The first function of this page, comprised of a ListView, is switching on or off the push notification. Essentially, if the user maintains

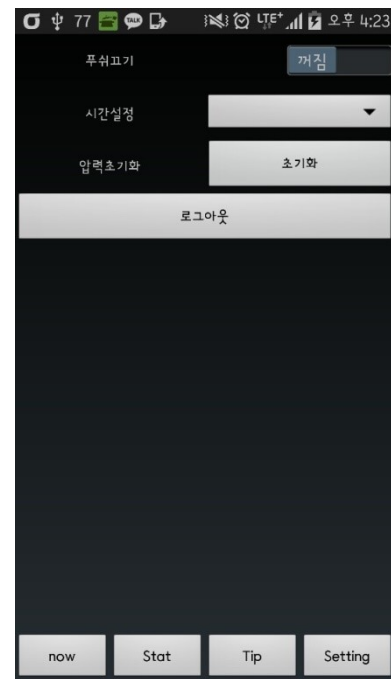


Fig. 13. Setting Page

a bad posture for a long time, the user will receive a push notification that warns his/her long-time bad posture. This function can be controlled by the user.

The second function is to set the time when user gets push alarms while user keeps wrong posture. It ranges from minimum 1 minute to maximum 10 minutes. For example, if you set your time to 2 minutes and you maintain a bad posture for 2 minutes, you will directly receive a notification.

The third function resets the pressure value that matches the users' body. According to individuals' weight and environment, the pressure value differs. Hence, it is necessary to save individuals' initial pressure value of the correct posture in the server. The last but not least function is log-out. It allows the user to log out from the app.

- Body Balance Page In this page users could take picture of their upper body sitting on the chair. And the program analyses the users body balance by comparing left and right location of shoulder, chest and waist. Then the program shows how much the users body is twisted using degree. For example if the user's left shoulder is tilted to the left about 30 degrees, the user could see the status of the body balance by green and red line(Green line is criteria line meaning good body balance

and the Red line shows how much the body is tilted.)and the message like "Your left shoulder is tilted to the left about 30 degrees" is shown.

V. ARCHITECTURE DESIGN AND IMPLEMENTATION

A. Overall architecture

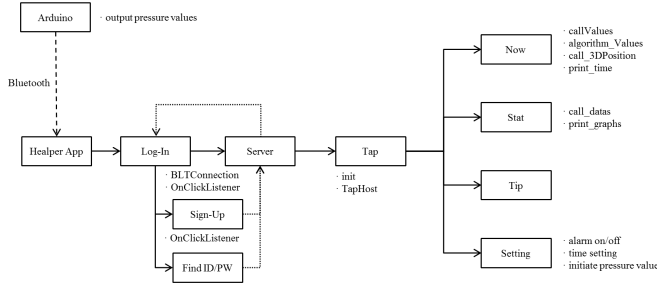


Fig. 14. overall architecture

B. Directory organization

Dir 1:

Healper_Proto/app/src/main/java/
com/example/lg/healper_proto

Dir 2:

Healper_Proto/app/src/main/res/
layout

TABLE VII
DIRECTORY ORGANIZATION

Directory	File names	Module name in use	Etc
Dir1	MainActivity.java	MainActivity class	
Dir1	NowActivity.java	NowActivity class	
Dir1	SettingActivity.java	SettingActivity class	
Dir1	SignUpActivity.java	SignUpActivity class	
Dir1	StatActivity.java	StatActivity class	
Dir1	TapActivity.java	StatActivity class	
Dir1	TipActivity.java	TipActivity class	
In Arduino	ValuesToApp.ino	setup, loop method	
Dir2	main_sub.xml		
Dir2	now_sub.xml		
Dir2	setting_sub.xml		
Dir2	signup_sub.xml		
Dir2	stat_sub.xml		
Dir2	tap_sub.xml		
Dir2	tip_sub.xml		

C. Code analysis

1. ValuesToApp.ino

– Purpose:

To send the pressure value that Arduino receive from the pressure sensor to the application through bluetooth,

– Functionality:

This code is embedded in the Arduino. The pressure sensor accepts the pressure value every second, and the pressure value is sent to the application through bluetooth.

– Location of source code:

In Arduino

– Class componenets:

a. Setup method

Setup method is used firstly when the arduino is executed. In this program, the speeds of serial communication and bluetooth communication are established in setup method. Variables and Pin value that is connected with the pressure sensors are declared and established.

b. Loop method

Loop method is used constantly while the arduino is executed. In this program, the functions receiving and sending the pressure value are established in loop method.

2. MainActivity

– Purpose:

To show main page and before the program is used in earnest, to set fundamental environment.

– Functionality:

This program uses ID and password functions to save each users individual data. For this, MainActivity provides log-in function that each users can log-in as their ID, sign-up function for users who start to use, find ID/password function for users who have forgotten theirs and bluetooth function for interworking specific cushion.

- Location of source code:

Healper_Proto/app/src/main/java/
com/example/lg/healper_proto/
MainActivity.java

- Class componenets:

a. BLTConnection method

Relationship between the pressure sensors and smartphone is necessary because this program receives users posture data from the pressure sensor. In on_Click method, BLTConnection method performs on/off functions and searching/connecting devices functions of bluetooth. Therefore, BLTConnection method is necessary in the first page, that is to say Log-In Page.

b. On_ClickListener

There are four On_ClickListner in MainActivity.

1) Log_()In

When users enter their ID and password and click Log-in Button, if the account is signed up for server, loading data that are associated with the account from server synchronize with moving to TapActivity. Unless the account is signed up for server, Sign-Up Page appears.

2) Sign_()Up

Users who start to use this program use Sign-up button to sign up. When this button is clicked, Sign-Up Page appears.

3) Find ID/ Find Password

When users have forgotten their ID or password, Find ID/Find Password button is used to move to Find ID/Password Page.

3. SignUpActivity

- Purpose:

To show Sign-Up Page to users who start to use this program. Users can sign-up in Sign-Up Page.

- Functionality:

This activity saves new users accounts information. In this activity, therefore, ID that is going to be the primary key goes through the duplication check and individual account information are saved in server.

- Location of source code:

Healper_Proto/app/src/main/java/
com/example/lg/healper_proto/
SignUpActivity.java

- Class componenets:

a. On_ClickListener

There are three On_ClickListner in SignUpActivity.

1) Redundancy_()Check

Individual account information reported in this page is going to be saved in server. Each users ID is going to be the primary key in server database. Therefore, the duplication check is necessary to check if an entered ID that tries to sign-up is same as other existing users ID. There are duplication check button for this.

2) Sign_()Up

Account information entered by users is saved in server and moves back to Log-In Page again.

3) Cancel

Cancel Sign-Up process moves back to Log-In Page again.

4. TapActivity

- Purpose:

To show tap page and consitute tap view which consist of Now, Stat, Tip, and Setting for customer.

- Functionality:

TapActivity is the body of this program. We build this program to make users be able to transfer freely using tab from a tap to another tap among four taps; now status, statistics, tips and setting page. TapActivity is used in this part to make Fragment Activity showing each page as fragment.

- Location of source code:

Healper_Proto/app/src/main/java/
com/example/lg/healper_proto/
TapActivity.java

- Class componenets:

a. init method

Init method in onCreate method, when this page is created, initialize all of taps to interwork each other.

b. TapHost

when four taps at the bottom of the page are clicked, Fragment is changed to an activity corresponding the tap. Switching page is convenient by using taps.

5. NowActivity

– Purpose:

NowActivity shows users current posture. It helps users grasp what posture and how long they have adopted.

– Functionality:

NowActivity analyzes the pressure value that comes from Arduino through bluetooth. It also helps users see their current postures in 3D and shows how long the users have adopted the postures and how long they have sat by using chronometer function.

– Location of source code:

```
Healper_Proto/app/src/main/java/  
com/example/lg/healper_proto/  
NowActivity.java
```

– Class componenets:

a. callValues method

callValues method in onCreateView saves the pressure values that are sent from Arduino through bluetooth in variables to input those values in algorithm_()Values.

b. algorithm_()Values method

Using the pressure values saved in variables by callValues method, it outputs the anatomy-computed current posture by input the variables in designed algorithm for analogizing the current posture.

c. call_()3Dposition method

call_()3Dposition method invokes 3D_()Position Graphic file corresponding current posture that is computed by algorithm_()Values and displays it in Now Page.

d. print_()time method

print_()time method displays users' total sitting time and the time users have adopted their current postures in Now Page.

6. StatActivity

–Purpose:

StatActivity makes statistical charts of users' posture data in server and displays three graphs arranged by the week, the month and the year.

– Functionality:

StatActivity loads users' data from server, makes statistical charts and by three graphs, helps users grasp their posture tendencies that they have adopted.

– Location of source code:

```
Healper_Proto/app/src/main/java/  
com/example/lg/healper_proto/  
StatActivity.java
```

– Class components:

a. call_()data method

call_()data method in onCreateView loads a recent year's data of users from server. That loaded data is saved in time_()array.

b. print_()graphs method

Using data that is saved in variables through call_()datas method data, makes graphs and displays them in Stat Page.

7. TipActivity

– Purpose:

To select and recommend orthotherapy videos that users probably need mostly after analyzing their postures.

– Functionality:

StatActivity analyzes today's posture statistics and recommend the orthotherapy exercise for the particularly bad posture in priority. On the basis of data of server, furthermore, it recommends two or

three exercises for the steadily bad postures.

– Location of source code:

```
Healper_Proto/app/src/main/java/  
com/example/lg/healper_proto/  
TipActivity.java
```

– Class components

8. SettingActivity

– Purpose:

To configure software setting and hardware setting to suit users' individual conditions.

– Functionality:

This program basically lets users know that they are maintaining bad posture by push alarm. However, it also has function that turns on/off push alarm function for the case that users do not want to be disturbed by push alarm. Or users can configure schedule time when they want to be alarmed. Because every user has different weights and balances, in addition, we added reset function in Setting Page in case of that basic pressure value of postures might need to be configured newly.

– Location of source code:

```
Healper_Proto/app/src/main/java/  
com/example/lg/healper_proto/  
SettingActivity.java
```

–Class components:

a. alarm on/off method

There is a on/off switch for the push alarm in Setting Page. As we established on/off function in OnClickListener of this switch, users can choose whether they turn on the push alarm or not.

b. time setting method

There is a list box in Setting Page. In this list box, there are ten options; from 1 minute to 10 minutes. After that time user selects, push alarm informs that users are maintaining the bad postures.

c. initiate pressure value method

Setting up a common pressure value is impossible since every users have different pressure values.

Therefore, we made it possible that every users configure their own pressures value. When initialization button in Setting Page is clicked, the average pressure value that has input for five seconds is saved in application and also in server. This program judges whether users' current postures are wrong or not comparing with the saved values.

VI. USE CASES

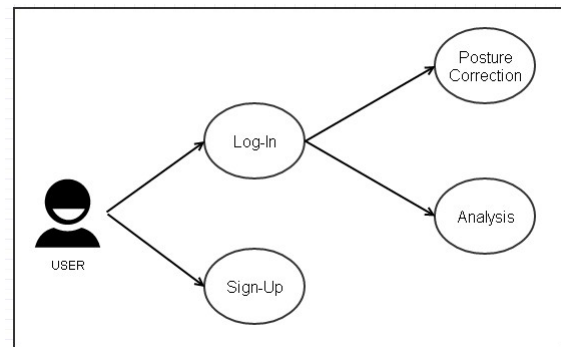


Fig. 15. Overall Usecase

The overall usecase is shown in the Figure 20. Our project's platform is Android-Mobile Application. So, first user should download our application and execute the program to do self-orthotherapy. And if user executed the program, a dialog window to get permission of bluetooth and connect with arduino would pop up. Then user will find the proper device and connect with the arduino and finally could see the first page of our application. After that if user has registered already, user could enter into the next functions like 'Posture Correction' or 'Analysis'. If not, user should sign-up and register users data into the program's database to use this program.

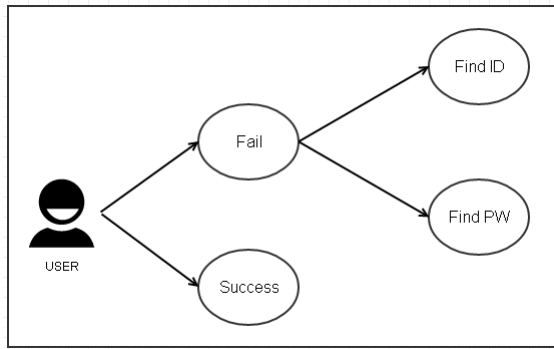


Fig. 16. Log-In Usecase

To use self-orthotherapy program, user should log-in to our system. But if user fails to do log-in, there are two ways of solving this problem. These are 'Find ID' and 'Find PW'. Log-In Usecase is shown in the Figure 21 and the Use Case's descriptions are written in the table x.

Table x : The description of Log-In Usecase
 Use Case Description Find ID There are two ways to solve log-in problem. The first is 'Find ID'. User could find the own ID by writing the name and E-mail compared to the database's data.

Find PW And the second is 'Find PW'. User could find the own password by writing the ID, and name compared to the database's data. It could meet the need of 'Log-In' requirement.

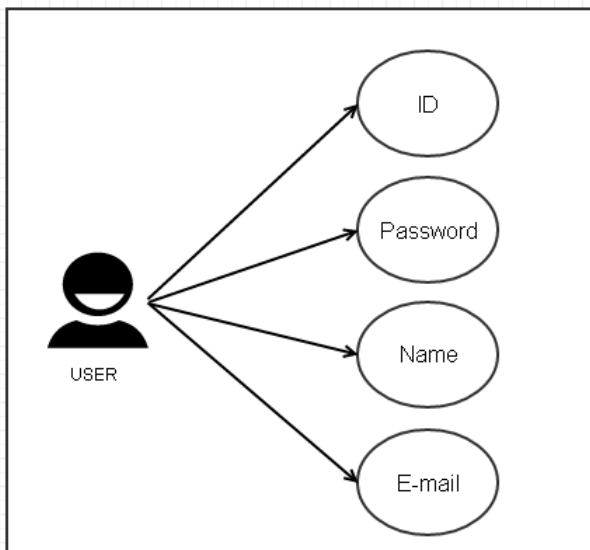


Fig. 17. Sign-up Usecase

If user had not registered in our program, user should do sign-up to use the program. There are

four components to register user's data into the database. It could meet the need of 'Sign-Up' requirement. Sign-Up Usecase is shown in the Figure 22 and the Use Case's descriptions are written in the table x.

Table x : The description of Sign-Up Usecase. Use Case Description ID User should make unique ID to sign-up. It should be made of english and numbers under 10 characters. The special letters can not be used to make ID. And, it is mandatory for you to double-check your ID. Password There is no limit in the length of the password and the password will be replaced with other letters, such as ****, as soon as you enter your password for the sake of absolute confidentiality. Name User should write own full name(first name + last name). And the english only could be used. E-mail User should write one e-mail address. User could select the portal's address like 'gmail.com' or 'naver.com' or write the address in person. After that the permission e-mail will be sent to the written e-mail address. So, it is not possible to do sign-up unless the user do not check the permission e-mail.

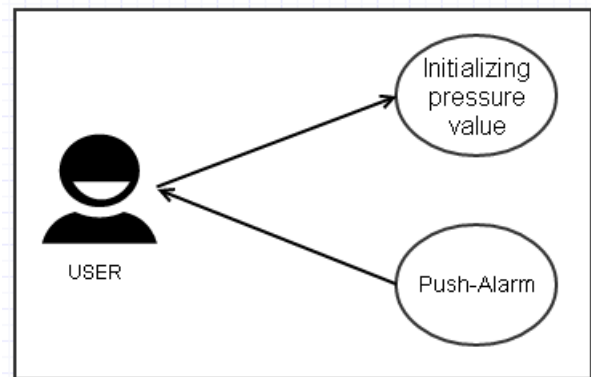


Fig. 18. Posture Correction Usecase

The main function of this program is to correct users bad posture when user sits down on the chair. So, setting the users own pressure value is mandatory. Because people have different physical size and weight. Initializing pressure value is the function to meet that requirement. After initializing pressure value, user could get alarm message when they keep bad posture. Posture Correction Usecase is shown in the Figure 23 and the Use Cases descriptions are written in the table x.

Table x : The description of Posture Correction

Usecase Use Case Description Initializing Pressure Value To get posture correction alarm, initializing pressure value is needed. Because our algorithm needs the fixed pressure value suited to the user. So, average of 5 times pressure values(when user sit down to the cushion, the pressure values are recognized.)is selected as the initial pressure value of user. Push-Alarm If user has wrong posture, program would recognize that status using posture correction algorithm. And the push alarm will be popped up to the users mobile phone like Do not make your leg crossed.

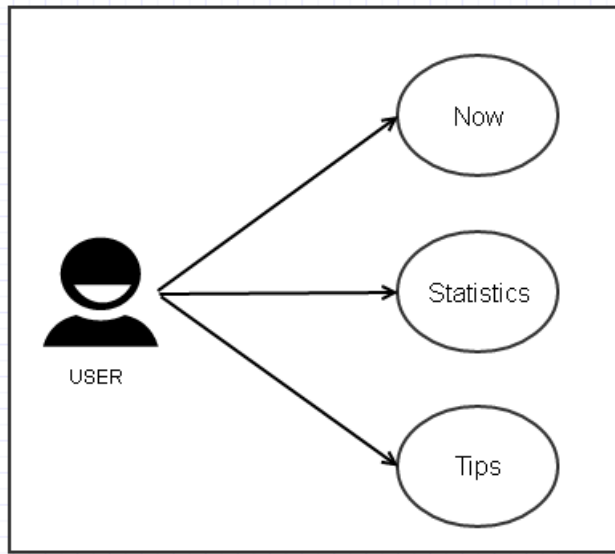


Fig. 19. Analysis Usecase

The another main function of this program is to offer analysis information using users posture data. And there are three sub functions in analysis part Now, Statisticsand Tips. Analysis Usecase is shown in the Figure 24 and the Use Cases descriptions are written in the table x.

Table x : The description of Analysis Usecase Use Case Description Now User could see the two type of time in the Now part. First is the the time of users actual working hour. And the second is the time how much user keeps good posture and bad posture. And user could see the 3D feature reflecting users present posture. So, it could help user recognize their present posture visually. Statistics User could see the three type of analysing graphical data such as Day, Week and Month. The users data is saved in the database and the program analyzes that data

and shows the graphical graph using that data. So, user could see the data distribution according to the selected period(User could select period by touching the upper tab). It meets the Statistics and Visualization requirements. Tips User could see the some videos about stretching good for posture correction. And some of exercises which are good for posture correction are recommended in this page.

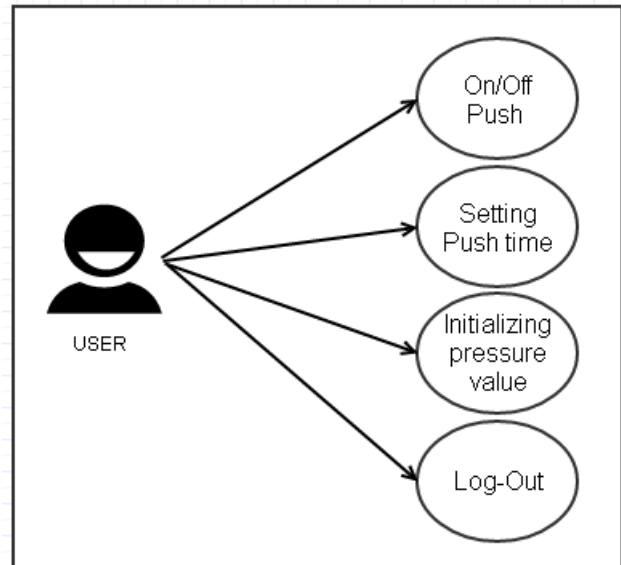


Fig. 20. Setting Usecase

User could do four functions in the Setting page. Those are On/Off Push, Setting Push time, Initial-izing pressure value and Log-Out. This functions could help user cutomize programs functions. Set-ting Usecase is shown in the Figure 24 and the Use Cases descriptions are written in the table x.

Table x : The description of Setting Usecase Use Case Description On/Off Push User could change the status of push alarm by On/Off. When user has urgent project meetint or needs to keep mobile phone silent, this function would be needed. Setting Push time User could set the time when user get the push alarm. The time has range from 1 to 10 minutes. For example, if user selects 2 minutes in this page(Default value is 1 minute)user would get push alarm when user keeps bad posture for 2 minutes. Initializing pressure value User could initialize pressure value in this page. The reason why user uses this function is that the actual user could be changed. So, when the actual user is touched

the program initializes the pressure value making average value getting five times pressure values. Log-Out User could do Log-Out in this page. This function is required when user do not want to use this program or take out his/her own data to the database.

.

VII. SOFTWARE INSTALLATION GUIDE

8.1 How to install?

Fig. 21. Searching at playsotre

First, you have to access the playstore(android) or App Store(iOS) and search [SMART BROCHURE].

Fig. 22. Access the application page

Second, click the [installation] button and download the [SMART BROCHURE] to your smartphone.

Fig. 23. Downloaded to your smartphone

Finally, you finish the installation.

Fig. 24. install using QRcode

Or, take this QR-code and access to the Playstore.

the deadline for each process is one of the most important components, in order not to delay the whole project. It was really a good experience for us and we feel that our ability and confidence has been improved a lot, so we would like to say thank you, professor. Additionally, (which you may already know about) we think it is better that doing usecase at the first step of the documentation because it would be better for us to draw the blueprint of our project documentation.

VIII. DISCUSSION

First of all, it was really hard for us to make the device communicate with beacons. Since we didn't have any SDK that works with our beacons, we had to develop the code from nothing. Therefore, we searched about it on the website, which gives developers android APIs that it supports. In the Bluetooth category, we were able to find the information about Bluetooth Low Energy(BLE), which was necessary for our project. We had to study about the threads in the android, how to use Bluetooth in our application, and make it possible to run the service on the background of the device. In addition, we tried the communication of mobile application and the server for the first time. There were just a few of problem in the language, but we had to study about the JSON type, which we used to send and receive the information about the exhibitions and brochures. Moreover, we had a good chance to use the new programs, 'filezilla' and 'pgadmin' we used for our server. Also, we had some trouble to use Latex and Github. We were very confused when professor told us that we have to use them. After a lot of tries, we were able to be used to use them. After getting used to it, we sincerely felt that it is really important to communicate with teammates in the project. From this class, we could deal with so many unfamiliar softwares and hardwares and learn how to go through the project from these process. Because there were a lot of our first tries, such as developing back-end server and linking it with the front-end applications. We also learned that keeping