Smart Shoebox
(Shoes care solution utilizing IoT concept)

Ko Byunghee, Kwon Gyuhyeok, Kim Junghyun, Shin Minki
Information System Department
College of Engineering
Hanyang University
Seoul, South Korea

Role	Name	Task and description etc
User	Kwon Gyuhyeok	Suggest the actual features for the Smart Shoebox users can feel comfortable and interesting to use
Customer	Shin Minki	Suggest the actual features for the Smart Shoebox customers can feel comfortable and interesting to use
Software developer	Kim Junghyun	Focusing on the Technical aspects of the Smart Shoebox while developing
Development manager	Ko Byunghee	Consider the service side of the Smart Shoebox while developing the system

TABLE I
ROLE ASSIGNMENT

***Abstract*—This document is about the realization of automatic remote control for shoesthrough IoT. We will make smart shoes cabinet that provides this kind of features with other different kind of functions.**

***Index Terms*—shoebox; shoes care; shoes rack; IoT;**

I. INTRODUCTION

Many people experience difficulty managing their own shoes in a decent and pleasant form. Especially for the people living alone, keeping shoes clean and sweet smelling becomes a tough task to manage. When it rains, shoes get wet and dirty. Can you imagine the smell and feel of the shoe? Even worse the smell starts from the entrance to the place where you will go to sleep. This is when the actual management features are required.

What if someone or something could take care of my shoes periodically and automatically. If the shoes could be managed regularly with the aspects of humidity, temperature, and sterilization, it will save money and also provide a pleasant day with a cozy footwear. To realize the concepts of taking care of our shoes, we will develop a shoebox which manages shoes condition by controlling humidity and temperature automatically and periodically.

We are going to use Arduino to support with humidity and temperature recognition by receiving inputs through switches or sensors. Internet of Things (IoT) is also on the base of the idea. The ability to control things (especially shoes in this case) through internet is the main concept we are trying to realize. We are looking forward to create an integrated service tool such as situation awareness, automatic computing, self-growing.

II. REQUIREMENT

A. Optimizing environment function

When we wear shoes, they easily become in a state of high temperature and humidity which causes the disgusting smell, which is also the best environment for bacteria to grow. As a result, there is a need to control the condition of the cabinet keeping the shoes. To provide an optimized environment automatically and also on user?s demand is the goal. (There is a need for defining optimized temperature and humidity)

1) Temperature/Humidity control through electric fan (automatic): The sensor receives temperature and humidity as inputs and provides an optimized environment as an output.

2) Temperature/Humidity control through ultraviolet lamp (automatic): The sensor receives temperature and humidity as inputs and provides an optimized temperature and humidity as output.

3) Drying feature (on demand): In case the user's shoes get wet by rain or other liquids the user can request for drying will operate (1), (2).

4) Sterilization function (on demand): In case the user feels the need for sterilization, user can request for this function, which operates (1), (2). This function (4) differs from (3) in degrees of intensity.

5) Deodorization function (on demand): In case the user feels the need for deodorization, user can request for this function, which triggers a deodorant to shoot out.

6) Deodorization function (automatic): The user can set regular intervals to trigger the deodorant to shoot out.

7) Intensity control feature: The user can choose the intensity level of (1), (2). Intensity is calculated as number between 1 to 5.

B. Management function

Different type of shoes requires different type of proper cares. The shoe rack needs to understand and recognize the shoes type and provide a proper management for the shoes. (Modeling : changing ambiguous information into actual concept.)

1) Shoe categorization function (bar-code scanning): Shoe categorization through capturing the barcode for the shoes.

2) Shoe categorization function (user input based): Shoe categorization through selected category of the user.

3) Shoe categorization function (captured image): Shoe categorization through captured images of the shoes.

4) Shoe categorization function (3D scanning): Shoe categorization through 3D scanning of the shoes.

5) Setting the proper management tool: After Shoe categorization, based on the shoes category, the shoe rack provides the proper setting. (There is a need for defining proper setting for each category) The proper setting is different in the aspect of the intensity from Optimization environment function.

C. Analysis function

To keep the user's shoes in high quality we can provide an analysis for the shoes the user own.

1) Absence of shoes analysis (Base information): We have decided to analyze the absence of shoes by sensor and use it as a base information for other analysis functions.

2) Durability analysis: Durability is set to decrease by the time the shoe has been put on increases.

3) Life prediction analysis: Based on the information of (1), we provide the expected life of the shoes.

4) Preference analysis (personal): Based on the information of (1) for one user, we provide the preference information of the shoes. More the user put on, more the preference increases.

5) Preference analysis (general): Based on the information of (1) for a number of users, we provide the preference information of the shoes for general aspect. Using this Big data, the user can know which shoes are popular nowadays.

6) Frequency analysis: Based on the information of (1), we provide the frequency information for the shoes.

7) Walking habit analysis (health care): Based on the information flatness of the shoes , we provide the information about walking habit of the users.

D. Recommendation function

Smart Shoes cabinet will provide recommendation information with percentages based on different kind of aspects. Of course the final choice is up to the user.

1) Recommendation based on weather forecast : With weather API, the proper type of shoes is recommended.

2) Recommendation based on the use of shoes: Recommending the shoes type which matches with the user?s activity.

3) Recommendation based on the color of shoes: Recommending the shoes color which balances with the users clothing color.

4) Notice of recommendation rate by color: Showing the recommendation rate by different colors. For example, if the shoes are recommended, a specific color will appear on the shoe rack or on the screen the user is looking at.

5) Notice of recommendation rate by percentage: Showing the recommendation rate by percentage. If the shoes are recommended strongly, the percentage will appear on the shoe rack or on the screen the user is looking at.

E. Notification function

Shoes easily get dirty, since when people do activities, shoes are the first thing that touches the ground. The shoes cabinet will provide notification for contamination of dirt or rainwater by checking on the weight difference.

1) Recognition of contamination by sensor: With the increased weight, notification is given for contamination.

2) Notification for contamination by message: After the recognition of contamination, the information is notified to the user through messages.

F. Networking / Remote control function (UI)

Without the function for internet control, it becomes nothing more than a drying machine. With this networking function on the base, the user is able to take care of the users shoes any time, anywhere. This is the most important feature we will concentrate on. Providing the IoT environment is the main goal.

1) Control function through web programming (main): With web based program, the user can interact with the smart shoe care software and other provided information.

2) Control function through mobile (sub): With mobile application, the user can interact with the smart shoe care software and other provided information.

3) Control function through embedded system (sub): With embedded system, the user can interact with the smart shoe care software and other provided information.

III. Development Environment

A. Choice of software development platform

1) Platform used for developing: package We will use both Windows and MAC OS . Since Windows is the most popular OS used worldwide and MAC OS is the second most popular OS leaving out all the other versions of Windows. We thought MAC OS X will become more popular. We also thought using other OS besides windows will mean a lot for us to use another environment to develop a software.

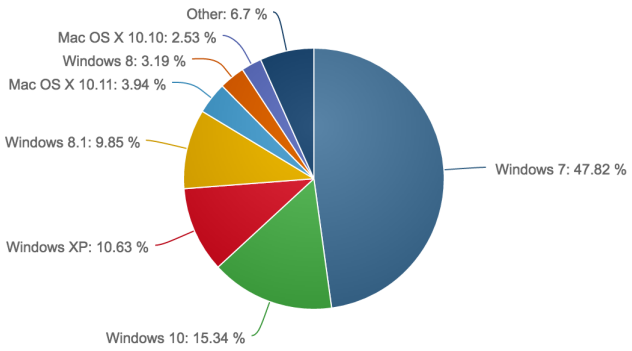

Fig. 1. Market share reports (January, 2016 to March, 2016)

Programming language	Reason
Arduino(hardware)	The main hardware part of our project is based on Arduino. The Smart Shoebox has functions to work provide behavioral motions such as recognizing the temperature and humidity of the shoebox, turning on the fan or infrared lamp as a result of it and so on.
MySQL(server side)	We need to have a database to save information about the shoes, users. To easily get and set and manage the information, we have decided to use a database management tool.
Ruby	We first thought of php for the work between the server and web side environment, since we have all learned php in another course. Though we thought it would be much better to learn a new language for this project. Ruby on rails was the interesting programming language in the aspect that it shortens and simplifies the code much more than the php.
HTML5 and CSS3(client side)	We have decided the user interface environment as a web-based structure. The functions of Smart Shoebox will be triggered and managed in the web.

TABLE II
PROGRAMMING LANGUAGE USED FOR DEVELOPING

2) Programming language used for developing: We are using Arduino, mySQL, Ruby and HTML. We are trying to provide a web service with arduino acting inside the Smart Shoebox. The frontend will be using html and css, while the backend will be using ruby and ruby on rails as a application framework. If we think of the server as a localhost, we might be using only ruby and ruby on rails for the server without MySQL.

Device	Price (won)
Arduino uno R3	7,500
Bread board	2,400
Wifi module(ESP8266)	9,000
Temperature Humidity sensor	3,000
Pressure Sensor	14,000
Fan(actuator)	4,000
Board	2,000
USB cable	500
jump wire	2,500
M-F wire	2,000
Resistance	200(5 per unit)
Small LED lamp	1,000
AA battery	1,200
transistor	500
TOTAL	49,800

TABLE III
COST ESTIMATION(HARDWARE)

Software	Task Description
Source Tree(v1.8.3)	Version control
Git(v2.8.1)	Project control
Github	Remote repository
Sublime Text3(3103)	Text editor
mockflow	Wireframe creation
Mac OS X El Capitan	Operating System
Windows 8 / 10	Operating System
Arduino(v1.6.8)	Text editor for Arduino
TOTAL	0

TABLE IV
COST ESTIMATION(SOFTWARE)

3) Cost estimation (Software / Hardware): TABLE III and TABLE IV

B. Software in use

We have researched to find out if there is any existing software or algorithm in use doing a similar task we are trying to provide. We were really surprised to find so much information related to our project. There was a lot of algorithms and systems during the research. The most interesting and related ones were the three below.

1) Temperature Humidity Control system: As anyone can think of the air conditioner or greenhouse there were already a lot of systems and devices doing the actual part of our project to control the temperature and humidity for the given environment. (For our home, or for growing plants in the optimized temperature and humidity, and so on.) Even there were a lot of information about making the Arduino actually work as we planned to.

Fig. 2. Advance Temperature Control (http://infusionva.com/)

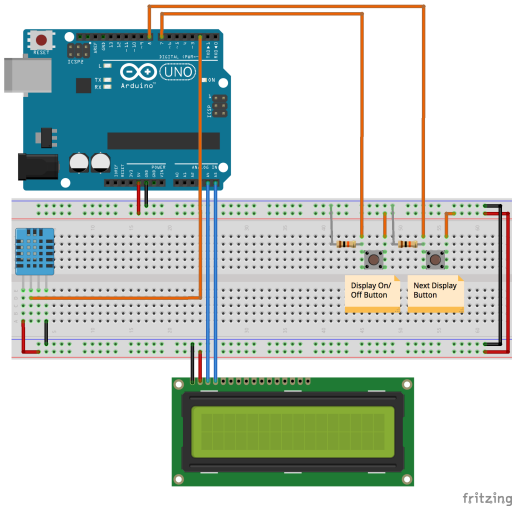

Fig. 3. Airconditioner automatic control through Arduino

2) Recommendation System : There is an extensive class of Web applications that involve predicting user responses to options. Such a facility is called a recommendation system. In Figure above we see an example utility matrix, representing users? ratings of movies on a 1?5 scale, with 5 the highest rating. Blanks represent the situation where the user has not rated the movie. The movie names are HP1, HP2, and HP3 for Harry Potter I, II, and III, TW for Twilight, and SW1, SW2, and SW3 for Star Wars episodes 1, 2, and 3. The users are represented by capital letters A through D. The goal of a recommendation system is to predict the blanks in the utility matrix. This recommendation system was in common with our project in the point that we will provide a recommendation information for the shoes with the weather APT and color of the shoes matching with the user?s clothes.

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Fig. 4. utilitymatrix

3) Classification Algorithm in datamining: Basic Principle (Inductive Learning Hypothesis): Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.

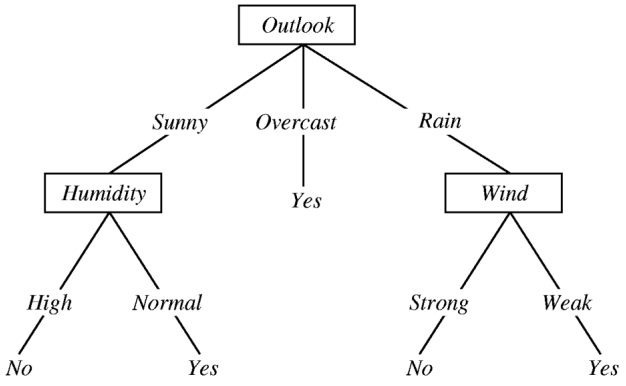

Fig. 5. decisiontree learning

This algorithm was in common with our project in the point that we will classify the shoes.

C. Task Distribution

We have decided to distribute the project in big parts to make each participants to be responsible for the assigned parts. Still every person has to know how the project is going on in a big picture while being responsible for the assigned part.

Name	Responsible Part
Kwon GyuHyeok	Arduino
Shin MinKi	MySQL
Kim JungHyun	Wireless fidelity control
Ko ByungHee	Ruby on rails

TABLE V
TASK DISTRIBUTION

IV. SPECIFICATION

The specification is mostly in pseudocode and additionally graphs and charts will be used to specify the requirements. Particularly, mockflow will be used to wireframe the user interface. The pseudocode is a little bit close to the programming languages we have already learned during other courses while disregarding the details of the grammar.

A. Optimizing environment function

1) Temperature/Humidity control through electric fan (automatic):

2) Temperature/Humidity control through ultraviolet lamp (automatic): The arduino sensor receives temperature and humidity as inputs and truns on fan and ultraviolet lamp when temperature drops below 15 Celsius degree or humidity is higher than

```
// Check temperature and humidity and optimize automatically
function check_temperature(){
    if (temperature < 15) {
        return false
    else
        return true
}
//When humidity is less than 40, it is better to have higher temperature

function check_humidity(){
    if (humidity > 40 )
        return false
    else
        return true
}

//실내 적정 습도가 50~60%이고 신발장은 40%이하가 적절하다고 합니다

function auto_control_temperature_humidity(){
    if(check_teperature() == true and check_humidity() == true){
        fan_off()
        lamp_off()
    }
    else{
        fan_on(MID)
        lamp_on(MID)
    }
}
```

3) Drying feature (on demand): In case the user?s shoes get wet by rain or other liquids the user can request for drying function. The function will turn on fan on maximum rate and intensity of lamp will be middle.

```
boolean check_demand_drying

function demand_control_temperatureHumidity(){
    if(check_demand_drying == false){
        fan_off()
        lamp_off()
    }
    else{
        fan_on(MAX)
        lamp_on(MID)
    }
}
```

4) Sterilization function (on demand): In case the user feels the need for sterilization, user can request for this function. The function will turn on fan on middle rate and intensity of lamp will be maximum.

```
boolean check_demand_sterilization

function demand_control_temperature_humidity(){
    if(check_demand_sterilization == false){
        fan_off()
        lamp_off()
    }
    else{
        fan_on(MID)
        lamp_on(MAX)
    }
}
```

5) Deodorization function (on demand): In case the user feels the need for deodorization, user can request for this function, which triggers a deodorant to shoot out. The Arduino will trigger the deodorant.

```
boolean check_demand_deodorization

function demand_control_deodorization(){
    if(check_demand_deodorization == true){
        deodorization_on()
        check_demand_deodorization = false
    }
}
```

6) Deodorization function (automatic): This function will provide the Smart Shoebox to trigger the deodorant to shoot out in one hour interval.

```
function auto_control_deodorization(){
    deotoranzation_on()
    delay(one hour) // one hour interval
}
```

7) Intensity control feature: The user can choose the intensity level of fan and lamp. Intensity is calculated as number between 1 to 3, which means maximum, middle, minimum.

```
fucntion fan_on(int a){
    if a==3 // turn fan MAX
    else if a==2 //  turn fan MID
    else // turn fan MIN
}

fucntion lamp_on(int a){
    if a==3 // turn lamp MAX
    else if a==2 // turn lamp MID
    else // turn lamp MIN
}
```

B. Management function

This management function is mostly about modeling and categorizing the shoes. Through the serial number scanning, captured image, 3D scanning and user input, the information about the user and shoes are managed in the database.

1) Shoe categorization function (bar-code scanning):

2) Shoe categorization function (user input based):

3) Shoe categorization function (captured image):

4) Shoe categorization function (3D scanning):

5) Setting the proper management tool: Modeling

The big arrow pointing to the database will be one of the ways of receiving the information of the shoes by the user. Like we mentioned at above part the ways will be the serial number scanning, analyzing the captured image, 3D scanning and analyzing user input.

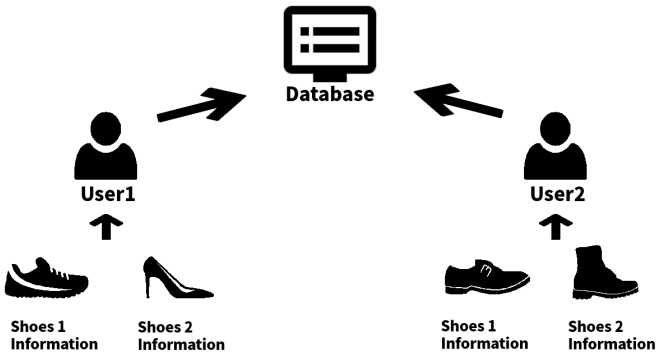

Fig. 6. Shoe information into the database

As we can see in Fig.7 the saved information of the shoes will be assigned to each model we will provide before. For each model the proper care solution will be provided. The modeling of each shoe into a number of category and setting the proper care solution will be the most important part to realize this function. Also at the same time it is the most ambiguous part before the design phase.

C. Analysis function

1) Absence of shoes analysis (Base information): We have decided to analyze the absence of shoes by weight sensor and use it as a base information for other analysis functions. The absence timer will be on and count the time of the shoes absence.

Fig. 7. Example of the shoes modeling

```
int absence_time

function check_absense(){
    if (weight_sensor == 0)
        absence_timer_on()
    else
        absence_timer_off()
}
```

2) Durability analysis: Durability is set to decrease by the time the shoe has been put on increases. We have decided the average shoes usage time as 12 hours a day and the durability changes while the absence time increases. Less than 2 weeks is considered to be a new one. Between 2 weeks and 4 weeks is considered to be not bad. Between 4 weeks and 8 weeks is considered to be washed. Between 8 weeks and 16 weeks is considered to be careful. More than 16 weeks the shoes is considered to be replaced.

```
// average time wearing shoes : 12 hours a day
string durability

function check_durability(){
    if absence_time < 2 weeks
        durability = new one
    else if 2 weeks < absence_time < 4 weeks
        durability = not bad
    else if 4 weeks < absence_time < 8 weeks
        durability = need an washing at least once
    else if 8 weeks < absence_time < 16 weeks
        durability = be careful
    else
        durability = recommend to buy new one
}
```

3) Life prediction analysis: Based on the information of absence time, we provide the expected life of the shoes. We subtract the absence time from the original life time.

```
int shoes_number // assigned number for each shoes
int life_time

function check_life_time(int shoes_number, int absence_time) {
    life_time = original_life_time(shoes_number) - absence_time
    return life_time
}
```

4) Preference analysis (personal):

5) Preference analysis (general): Based on the information of absence time for a number of users, we provide the preference information of the shoes in general aspect. Since we all store the information about the shoes of the user in the database this is possible in both personal and general aspect. Using this Big data, the user can know which shoes are popular nowadays.

```
int count_using_time = absence_time
int preference // server variable

function check_preference(){
    if count_using_time > 2 weeks {
        count_using_time = count_using_time - 2 weeks
        // one week use : preference + 1
        preference = preference + 1
    }
}
```

6) Frequency analysis: Based on the information of absence time, we provide the frequency information for the shoes in percentage. We devide the used days by the whole day since the user started to use the Smart Shoebox.

```
(6)  Frequency analysis

int use_date

function check_frequency(){
        return use_date / whole_date * 100
}
```

D. Recommendation function

Smart Shoes cabinet will provide recommendation information with percentages based on different kind of aspects. Of course the final choice is up to the user. The main recommendation standard is the weather. So we have decided to place a screen for the weather.

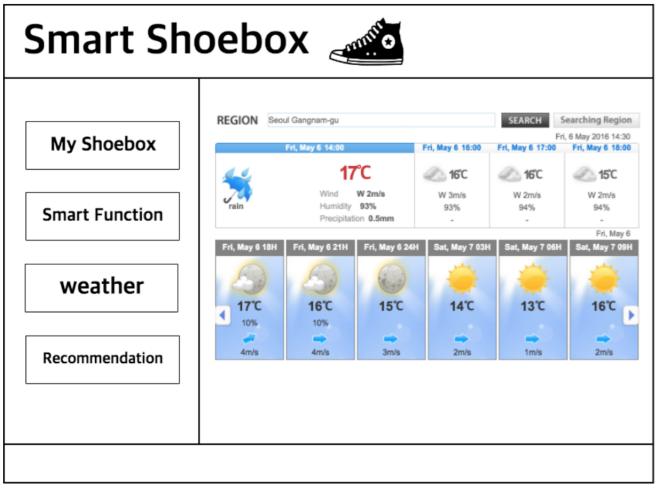

Fig. 8. Weather on the web screen

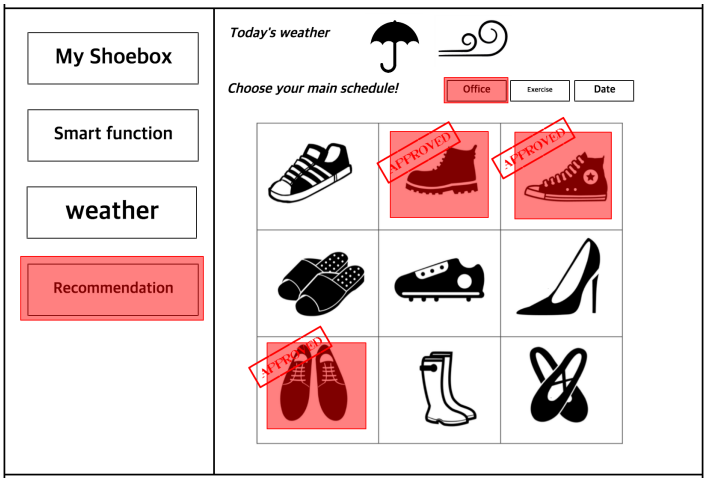

Fig. 9. Recommendation on the web screen

1) Recommendation based on weather forecast : With weather API, the proper type of shoes is recommended. We decided to classify weather in three cases. ('Sunny' and 'Cloudy' and 'Rainy or Snowy') We will restrict the recommendation choices by the weather becomes bad. The word proper (the proper shoes for the weather) will be represented in the shoes modeled information in the database.

```
string weather

fuction weather_forcast(String Weather){ // from weather API
    if(Weather='sunny'){
    return 1;
}
else if(Weather='cloudy'){
    return 2;
}
else if(Weather='rainy' or Weather='snowy'){
    return 3;
}
}
function recommend_weatherforcast(){
    switch(Weather_forcast()){
    case 1 : return shoes1
    case 2 : return shoes2
    case 3 : return shoes3  // sunny -> shoes1,2,3  cloudy -> shoes2,3  rainy -> shoes3
            break
    }
}
```

2) Recommendation based on the use of shoes: Recommending the proper shoes type which matches with the user?s activity. The word proper (the proper shoes for the activity of the user) will be represented in the shoes modeled information in the database.

```
string user_purpose

function recommend_user_purpose(){
    if user_purpose == exercise
        return shoe1
    else if user_purpose == office(formal)
        return shoe2
    else if user_purpose == (informal)
        return shoe3
}
```

3) Recommendation based on the color of shoes: Recommending the proper shoes color which balances with the users

clothing color. The word proper (the proper shoes for the clothes of the user) will be represented in the shoes modeled information in the database.

```
fuction recommned_color(class shoe){
    if shoe.color matches clothes.color
        return recommend
    else
        return not recommended
}
```

4) Notice of recommendation rate by color:

5) Notice of recommendation rate by percentage: Showing the recommendation rate by different colors. For example, if the shoes are recommended, a specific color will appear on the shoe rack or on the screen the user is looking at. Showing the recommendation rate by percentage. If the shoes are recommended strongly, the percentage will appear on the shoe rack or on the screen the user is looking at.

```
function notice_recommedation(class shoe) {
    if recomedation_rate > 80
        recommedation_color = green
        display recommedation_rate, recommedation_color
    else if 60 < recomedation_rate < 80
        recommedation_color = lightgreen
        display recommedation_rate, recommedation_color
    else if 40 < recomedation_rate < 60
        recommedation_color = orange
        display recommedation_rate, recommedation_color
    else if recomedation_rate < 40
        recommedation_color = red
        display recommedation_rate, recommedation_color
}
```

E. Notification function

1) Recognition of contamination by sensor:

2) Notification for contamination by message: With the increased weight, notification is given for contamination. The weight to be sensed is defined to be 20g. After the recognition of contamination, the information is notified to the user through messages on the screen.

```
function check_contamination(){
    if weight_sensor_now > weight_sensor_average + 20
        notification_message()
}
```

F. Networking / Remote control function (UI)

1) Control function through web programming (main):

2) Control function through mobile (sub):

3) Control function through embedded system (sub): To realize this function we will provide a user interface through mainly web and additionally mobile and a screen for the Smart Shoebox. The already mentioned functions before will be on the UI so that the user can interact with the system. The temperature and humidity will be provided on the screen the user will be seeing. Optimaization function, management function, analysis function, recommendation function, notification function will be on the main screen for users be able to use.

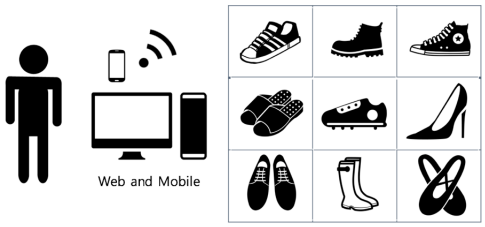

Fig. 10. Networking function through mobile and web

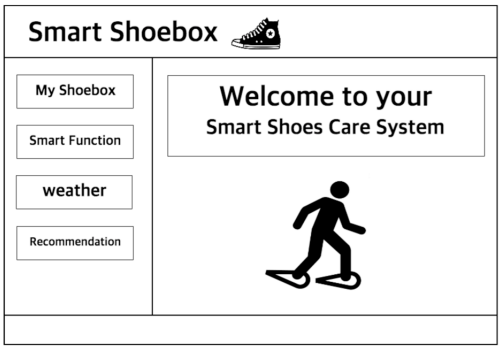

Fig. 11. Main page for Smart Shoebox

V. Architecture Design and Implementation

VI. Use Cases

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

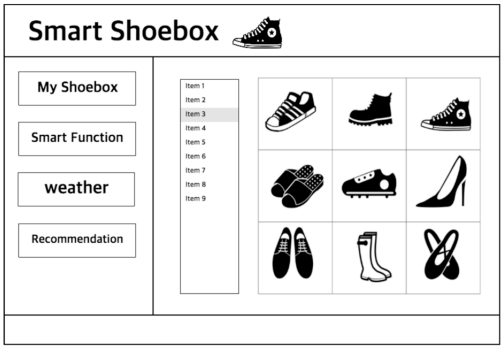

Fig. 12. Shoes status for Smart Shoebox