Hong Kong Institute of Vocational Education (Tsing Yi)
Department of Information and Communications Technology
Higher Diploma in Software Engineering

ITP4913M – System Development Project

Final Year Project – Final Report

Digital Home System

Digital Home System in IOT environment with the sensor control model building

Group Number: Group 23 (Johnny Group 4) Supervisor: Johnny Kwong Co-Maker: Jesse Shum

Student Name	Student ID Number	Signature
Au Chi Chung	150533022	
Kwok Yuk Lam	150032356	
Li Kai Kwong	150321559	
Yu Kwok Ho	150031347	

1. Abstract

In recent years, some accidents about the home safety in Hong Kong were happened due to lacking the awareness of the home safety. For example, the elderly faced the accidents such as the slip in a dark environment, fire disaster caused from cooking, and ease of slipping and falling from heights in public housing estates. Also, the accidents involving children fall injury and scald of their bodies are also common in Hong Kong because the parents are attempted to work and cannot take care of their elderly and children.

Moreover, there are few measures in the market to avoid relevant accidents, which are also uncomfortable for users in family use. For instance, most of the measures are expensive in the maintenance and installation that the cost cannot be afforded by the low-income families. With the progress of ever-changing technology, different digital devices could be used for home safety to avoid the home accidents.

Also, most of the smart home system in the market are also monotonous and focus on life-quality instead of a wide range of citizen. In addition, the system developments are also based on the simple use such as the monitoring and the control of switching the devices, there are no logical and secure features.

In view of this, the project aims to build a digital home system with multiple platforms for focusing on the element of “safety” and targeting on the user of “family” for preventing the occurrence of accidents. Regarding the system, there are two platforms - Android application and Website. The system can automate the processes, make some records and perform all operations related to the users, which can provide the direct status of the working environment and the direct information of the users.

2. Acknowledgement

ACKNOWLEDGEMENT

This project consumed huge amount of work, research and dedication. Still, implementation would not have been possible if we did not have a support of many individuals and organizations. Therefore we would like to extend our sincere gratitude to all of them.

First of all, we are thankful to the supplier, METAS for their logistical support. And then, we would like to express our special thanks of gratitude to our project supervisor Johnny Kwong as well as our project co-maker Jesse Shum who gave as the opportunity to do this project on the topic Digital Home System in IoT environment with the sensor control model building. Also, we are thankful for their financial support who providing necessary guidance concerning projects implementation.

Additionally, their support for provision of expertise, and technical support in the implementation are very important. Without their superior knowledge and experience, the Project would like in quality of outcomes, and thus their support has been essential.

Nevertheless, we express our gratitude toward our families and classmates for their kind co-operation and encouragement which help us in completion of this project.

Digital Home System
Final Year Project Group 23

Table of contents

3. Project description

3.1. Introduction

For the final stage of system development, the system has been the major change including the server development, database building, and the user-interface. But the project background of the scope will be stable at the details in initial ideas correspond with the solution of issues.

In the beginning of this report, the introduction introduces all the things in the proposal. And the problem is talking about the main problem which we are facing and analysis the existing product. Then, the solution is reducing the problem what we just talked.

Besides this, the functional requirement introduces all the function details of our application. Furthermore, the non-functional requirement is describing the hardware and the software which we are needed. Also, the project management is planning the project base on the system development life cycle.

For developing the application in web platform and smart device platform, the technical use is to introduce the requirement. Finally, the conclusion of the project is made.

3.2. Problems finding

- **Home safety and home accidents in Hong Kong**

Home accident in Hong Kong is a hidden crisis for the family. In recent years, home accident cases of the elderly and children have been rising, which is a serious issue and leads peoples to pay close attention.

According to the survey report about “Hong Kong’s top ten children home accident” by The Hong Kong Paediatric Foundation, pinched by the finger by the doors or windows and slipping in the bathroom are the most frequently occurring events. Moreover, the most accidents in Hong Kong occurred on Sunday afternoon. The report also explains that the parents have no enough time to take care of their children and cause the home accidents due to their busy business or housework. Also, the parents do not have a regular home-monitoring habit.

3.3. Proposed solution

- **Help reduce for home accidents**

In order to avoid the emergence of these home accidents, the digital home system will be applied. There are some sample features in the system such as avoiding children climb out of the house, preventing the children getting dangerous items (E.g. weapon, flammable liquids) and notifying the parents if a disaster is happened etc.

3.4. Comparison with similar product

Product Name	Savant APP by EAS System	Smart Living by HKT
Features & Sensors	• User tailor-made control scenes • Access home from anywhere with Savant Pro Remote • Support climate and lighting devices <u>Sensors</u> ▪ Lighting with on / off Switch ▪ Climate detect ▪ Home theater and entertainment ▪ Security Home monitoring	• User tailor-made control scenes • User can using application remote for turn on their air-con and cooled before they step in <u>Sensors</u> ▪ Lighting without on / off switch ▪ Curtains control ▪ Temperature control ▪ Security Home monitoring
Login method	None	Via to 3 different application (ULTI Home Control, Wiser2, and Control4)
Problem	Have no function about user management for user that it is lack of diversity for user control.	User needs to use different mobile application for control with the system.

▪ Table 1: Comparison with similar product of digital home system in Hong Kong

Those smart home systems are most popular in Hong Kong, and the comparison will compare the advantages and disadvantages of two systems for getting the benefit for our digital home system.

- Two of the smart home system “Savant APP by EAS System”, and “Smart Living by HKT” can access home from anywhere via the digital devices, and the software having the user tailor-made control scenes that user have a huge degrees of freedom and it can let user feels comfortable. But they focus the goal of the improvement of life-quality.

- For the first smart home system, “Savant APP by EAS System”, it have the comfortable element such as support tailor-made control scenes for user to setup their user property, and the system support climate and lighting devices that user can browse the weather via the application. Also, the system provides the security element as well as user can browse the status at home via security monitoring. But the disadvantage is that the application can only use in “Savant Pro Remote” which they provided, and it has no function about user management for user who has lack of diversity in user control. On the other hands, it may create the system security problem because of other one can browse the system easily.

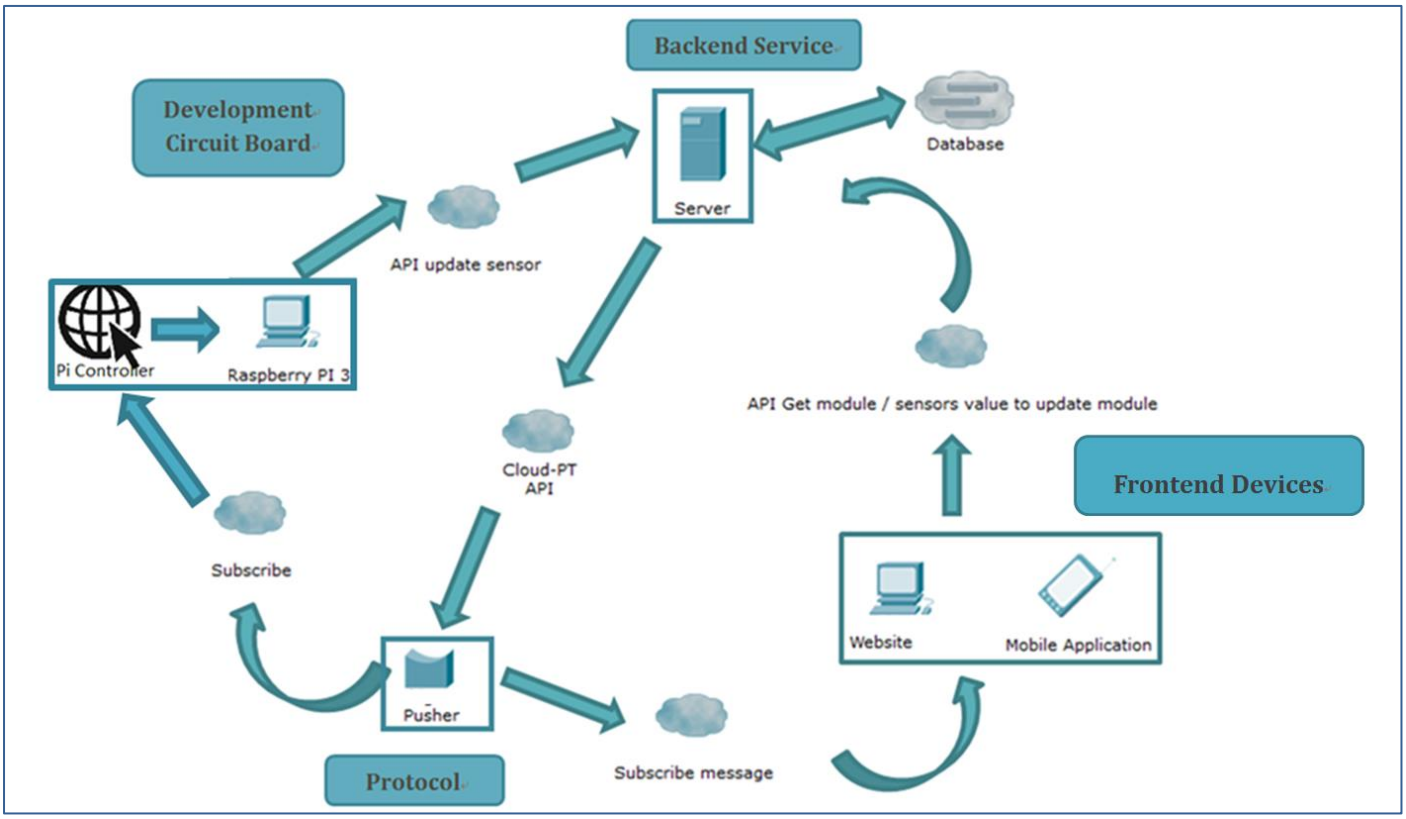
- For the second smart home system, “Smart Living by HKT”, the functions are most same with the first system and the target goal is also set on the improvement of life-quality. And the system can using smart phone devices via three different applications for taking control, the disadvantage is that user needs to use different mobile application for control with the system that they will feel trouble.

<u>Conclusion of the comparison</u>

1. The digital home system should better have some functions in management and control for users in the mobile application.
2. The digital home system should better have different platform for the user to control different devices. (Web platform for management and control, and Android platform for basic control)

4. System requirements

4.1. System architecture



- Figure 1: System architecture with digital home system IoT model

4.2. Architecture explanation

Backend service

- System backend server will connect with the database. It uses the PHP to do a slim framework for building up and REST FUL API, which can judge the input value whether it meets the criteria for turning on / off the module.

Protocol

- For convenience of make a real-time service, we choose using a “Pusher” protocol to provide related service. When any values are changed, the system will also send a message to the Pusher for notifying the client sides, including mobile application and website platform, to update the relevant data.

Development circuit board

- PI Controller is a platform for connecting the Raspberry PI and the server in different networks. When the server needs to update the sensors or actuators in the Raspberry PI, it is necessary to open the controller first for Raspberry PI to receive the request.

Frontend devices

- When website or the mobile devices which installed of the digital home system application that accepted the subscribe message from pusher protocol. Then those supporting device will send back the API Get module or sensors value request to the server for getting updated module.

4.3. Functional requirement

- Since the system is feasible, which means that all the monitoring settings are based on the users, there are some functions as samples to demonstrate our features as below.

4.3.1. Sensors

Functions	Function Description
- Partial Area (Living room) -	
1. Closing the windows automatically when it rains	**Purposes:** Some accidents of falling from heights in public housing estates are caused by some housework such as closing windows without window grilles when it rains. In view of this, if the windows are installed some sensors for detecting whether it rains and they are closed automatically, it can avoid and decrease the accident rate. At the same time, via the mobile application to turn on/off the windows, it can provide convenience for the users, especially the elderly. P.S. There is an infrared sensor for each window to check whether something exists, which can prevent some accidents such as pinched finger.
2. Detecting temperature and humidity of home location	**Purposes:** The temperature sensor and humidity sensor can coordinate the action of closing windows automatically. When a value is detected by the sensors and is out of a range, which is set by the users, the system will do something such as closing windows to reduce the environment's effect for users. For example, when the sensor detects a low temperature or high humidity of the home location, the windows will be closed. Also, in the mobile app and website, users can monitor the temperature and humidity of the home.
3. Checking the heart rate	**Purposes:** To provide a more safety for users, a heart rate detector will be offered to check their heart rate and the measurements record will be stored automatically for users to compare their physical conditions. Also, the mobile application will show the data for references.
4. Turning on/off the light automatically	**Purposes:** The elderly has poor eyesight that the dark and wet environment will make them ease of causing accidents. To reduce the opportunity of accidents, the system is equipped with motion sensors to detect the people.

	For the automatic lighting equipment, when the users are detected, the lights will be turned on automatically. Also, they can setup the time schedule per their request if they do not want to waste the electricity.
- Partial Area (Kitchen) -	
1. Turning off the electric range via mobile application to avoid forgetting it.	**Purposes:** Forgetting to close the electric range to cause the fire is a common home accident. If the users can turn off (excluding turn on) the electric range via the website or mobile application, the fire accidents could be reduced. When the sensor detects no person in kitchen and the electric range is open over a few minutes, which is according to users' settings, the system will send a warning alert to the users' mobiles and the buzzer will sound until the range is closed or someone exists in the kitchen. Moreover, when a fire is detected, the system will turn off the electric range automatically and turn on the sprinkler. At the same time, the camera will capture the situation and send it to the users' mobiles by email.
2. Turning on/off the drawer lock automatically	**Purposes:** In the kitchen, there are many weapons such as scissors and knife or some medicines. It is very dangerous if their children get them, which will cause an accident. For the automatic lock of drawer, the drawer lock will be opened automatically when the people is adult. As for another method, users can also control it via the website or mobile application.
- Partial Area (Bathroom) -	
1. Opening the exhaust fan automatically if the carbon monoxide is detected	**Purposes:** Leaking carbon monoxide from the gas water heaters is a crisis, If the people inhale the carbon monoxide during taking a bath, they may have difficulty in breathing or have an opportunity of death. When the carbon monoxide is detected, the system will turn on the exhaust fan automatically for air circulation.

- Table 2: Function requirement – Sensors

4.3.2. Mobile Application

Functions	Function Description
- Home Page -	
1. Displaying the actuators for control with push notification and the weather information	**Purposes:** There are some switches for users to control the actuators quickly and display a week of weather. The system will send a notification for users when having a change of actuator status and needing users to choose whether the actuator is open or not.
2. Using voice for function control	**Purposes:** The family members, especially the elderly, who may have motor impairment of the upper extremities. As a result, this feature could help them more easily for using the system.
- Heart Rate Page -	
1. Measuring heart rate	**Purposes:** This feature can make it easier to take care of family with elderly. For example, measuring the heart rate can provide real-time health record for the users.
2. Checking history record	**Purposes:** In addition, the users can also check their history records for comparison and reference.
- Weather Page -	
1. Displaying weather information	**Purposes:** The weather information could be offered and the users can easily pay attention to their clothes. Moreover, the charts of weather information such as temperature and humidity are displayed to users for reference.
- Location Page -	
1. Using the map function for getting other users' location (for root user only)	**Purposes:** This page would show the users' current location for who have installed the system application. If the user is an administrator (root account), the location of their family members, whom are willing to be monitored, will be showed. This feature can also make it easier to take care of their family members, especially the elderly, whom has Alzheimer's disease. Moreover, it could help the parents look for their lost elderly or children.

- Personal Page -	
1. Displaying personal information	**Purposes:** The personal page will basically display the personal information such as name, account type and email address.
- Settings Page -	
1. Handling the preference	**Purposes:** The Settings page will allow the user to edit their preference such as willingness of displaying their location, of receiving the push notification, etc.

- Table 3: Function requirement – Mobile Application

4.3.3. Website

Functions	Function Description
1. Displaying the actuators for control with notification and the weather information	**Purposes:** There are some switches for users to control the actuators quickly and display a week of weather. The system will send a notification for users when having a change of actuator status and needing users to choose whether the actuator is open or not.
2. Managing the actuators	**Purposes:** User can manage the actuators, grouped by rooms, such as updating actuators' setting. In addition, time range setting is also provided, which is a function for user to set up the actuators when to open and close.
3. Managing notification records	**Purposes:** All changed actuators' status, including users' action and sensors' control, are kept the records to the server. In this page, users can see the notification records for reference. Some filters such as date, actuator type will be provided.
4. Using the map function for getting other users' location (for root user only)	**Purposes:** This page would show the users' current location. If the user is an administrator (root account), the location of their family members, whom are willing to be monitored, will be showed. This feature can also make it easier to take care of their family members, especially the elderly, whom has Alzheimer's disease. Moreover, it could help the parents look for their lost elderly or children.
5. Setting function policy	**Purposes:** Function policy means that is a management of which sensors will control with actuators. User can create, update and delete the function policy per they request.
6. Displaying personal information	**Purposes:** In the profile page, there are some personal information such as name, account ID, account type and image.

- Table 4: Function requirement – Website

4.4. Non-functional requirement

4.4.1. Performance

Available time

- The digital home system should be available for use in 24 hours per day, which can allow users to control and manage the system every time and everywhere. Since the backend server is a core, it keeps operating with different system module such as the system circuit board, Raspberry PI 3 with the Arduino sensors.

Response time

- The system would be interacting to user within 10 seconds and process the data within 5 transactions per second when the mobile application or website are asking for access into system. When the user are using the mobile application for control sensors, the loading time of user action must try to deduct that it let system running smoothly. And the data result, reaction, and operating effect would affect the performance during the process data are transfer by backend server to different digital devices or circuit board.

4.4.2. Reliability

Data storage

- For the data storage and the data transfer in the backend server, we must ensure that the data in database and the data passing from system module are match. It is because it related to the safety and the sensors control at home. That all the data should be correct during the changes for user control.

System operation

- The mobile application and the website should process successfully per each request. Also, the system should be reserved the last control and keep the records for reference, that it can be more effective for maintaining data integrity.

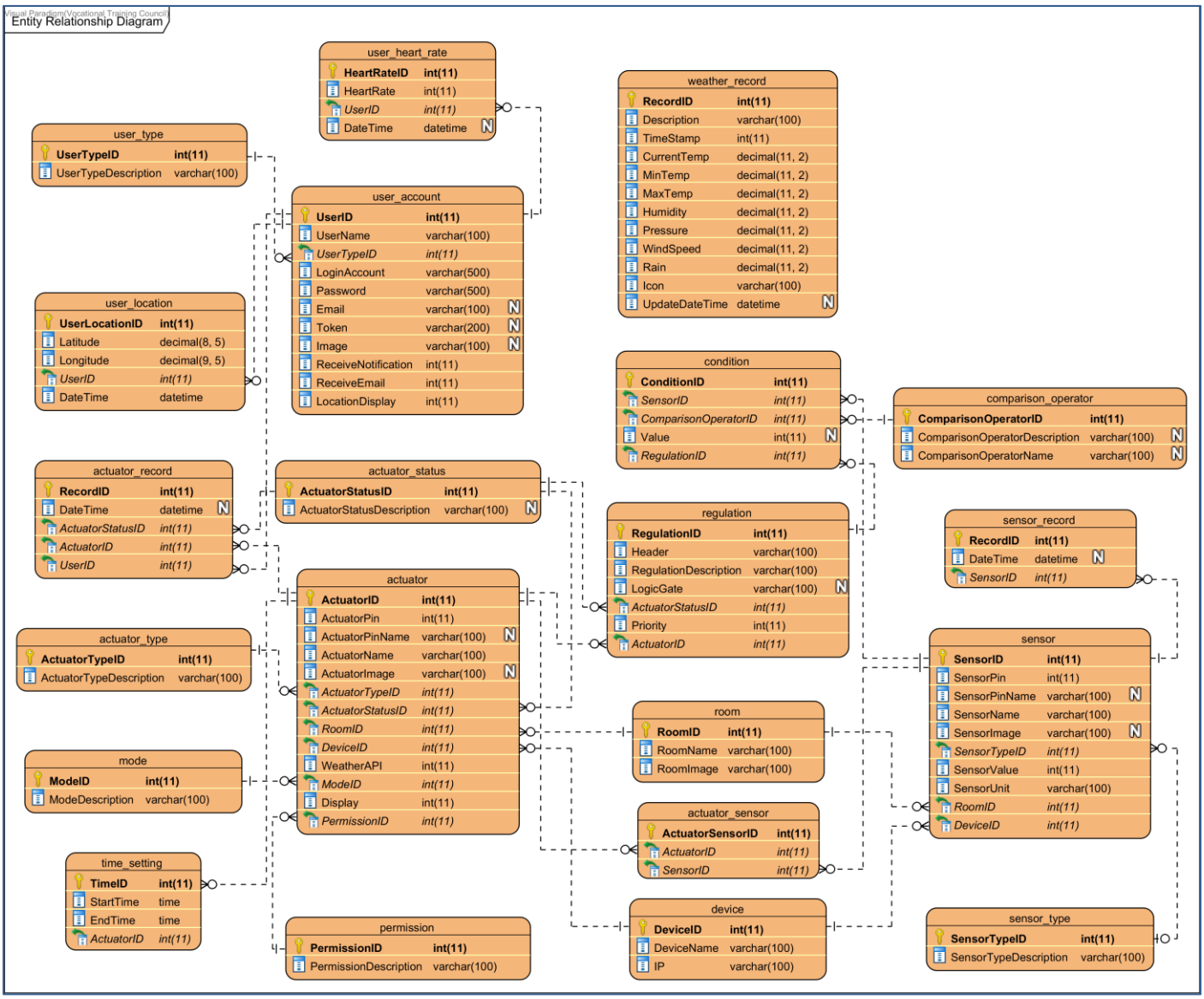
System security

- For the system security, we should ensure the user personal information and their account password have encryption in the system that it can avoid the data leakage when someone accesses the system.

5. System analysis

5.1. Data model

5.1.1. Entity relationship diagram



- Figure 2: Entity relation diagram for digital home system

5.1.2. Data dictionary

P.S. * Red color with star symbol = primary key * black color with star symbol = foreign key

Table - actuator

Name	Type	Length	Min	Max	Description
* ActuatorID	Integer	11	1	99999999999	Identity number for actuator
ActuatorPin	Integer	11	1	99999999999	Pin number of the actuator
ActuatorPinName	Alphanumeric	100	//	//	Pin name of the actuator
ActuatorName	Alphanumeric	100	//	//	Actuator name description
ActuatorImage	Alphanumeric	100	//	//	Short path of actuator image
* ActuatorTypeID	Integer	11	1	99999999999	Actuator type
* ActuatorStatusID	Integer	11	1	99999999999	Actuator status
* RoomID	Integer	11	1	99999999999	Room identity number
* DeviceID	Integer	11	1	99999999999	Device identity number
WeatherAPI	Integer	11	0	1	1: For weather control
* ModeID	Integer	11	1	99999999999	Mode number
Display	Integer	11	0	1	1: Display in the application and web for control
* PermissionID	Integer	11	1	99999999999	Control permission

- Table 5: Data dictionary – actuator table

Table - actuator_record

Name	Type	Length	Min	Max	Description
* RecordID	Integer	11	1	99999999999	Identify number of actuator record
DateTime	Datetime	//	//	//	Date time of the record
*ActuatorStatusID	Integer	11	1	99999999999	Actuator status identity number
*ActuatorID	Integer	11	1	99999999999	Actuator identity number
*UserID	Integer	11	1	99999999999	User identity number

- Table 6: Data dictionary – actuator_record table

Table - actuator_sensor

Name	Type	Length	Min	Max	Description
* ActuatorSensorID	Integer	11	1	99999999999	Actuator sensor identity number
* ActuatorID	Integer	11	1	99999999999	Actuator identity number
* SensorID	Integer	11	1	99999999999	Sensor identity number

- Table 7: Data dictionary – actuator_sensor table

Table - actuator_status

Name	Type	Length	Min	Max	Description
* ActuatorStatusID	Integer	11	1	99999999999	Actuator status identity number
ActuatorStatusDescription	Alphanumeric	100	//	//	Actuator status description

- Table 8: Data dictionary – actuator_status table

P.S. * Red color with star symbol = primary key * black color with star symbol = foreign key

P.S. * Red color with star symbol = primary key * black color with star symbol = foreign key

Table - actuator_type

Name	Type	Length	Min	Max	Description
* ActuatorTypeID	Integer	11	1	99999999999	Actuator type identity number
ActuatorTypeDescription	Alphanumeric	100	//	//	Actuator type description

- Table 9: Data dictionary – actuator_type table

Table - comparison_operator

Name	Type	Length	Min	Max	Description
* ComparisonOperatorID	Integer	11	1	99999999999	Operator identity number
ComparisonOperatorDescription	Alphanumeric	100	//	//	Operator description
ComparisonOperatorName	Alphanumeric	100	//	//	Operator name

- Table 10: Data dictionary – comparison_operator table

Table - condition

Name	Type	Length	Min	Max	Description
* ConditionID	Integer	11	1	99999999999	Condition identity number
* SensorID	Integer	11	1	99999999999	Sensor identity number
* ComparisonOperatorID	Integer	11	1	99999999999	Operator identity number
Value	Integer	11	0	99999999999	Sensor current value
* RegulationID	Integer	11	1	99999999999	Regulation identity number

- Table 11: Data dictionary – condition table

Table - device

Name	Type	Length	Min	Max	Description
* DeviceID	Integer	11	1	99999999999	System control device identity number
DeviceName	Alphanumeric	100	//	//	System control device name
IP	Alphanumeric	100	//	//	Internet protocol address

- Table 12: Data dictionary – device table

Table - mode

Name	Type	Length	Min	Max	Description
* ModeID	Integer	11	1	99999999999	Mode identity number
ModeDescription	Alphanumeric	100	//	//	Mode description

- Table 13: Data dictionary – mode table

Table - permission

Name	Type	Length	Min	Max	Description
* PermissionID	Integer	11	1	99999999999	Permission identity number
PermissionDescription	Alphanumeric	100	//	//	Permission description

- Table 14: Data dictionary – permission table

P.S. * Red color with star symbol = primary key * black color with star symbol = foreign key

P.S. * Red color with star symbol = primary key * black color with star symbol = foreign key

Table - regulation

Name	Type	Length	Min	Max	Description
* RegulationID	Integer	11	1	99999999999	Regulation identity number
Header	Alphanumeric	100	//	//	Header of regulation
RegulationDescriiption	Alphanumeric	100	//	//	Header description of regulation
LogicGate	Alphanumeric	100	//	//	Logic gate number
* ActuatorStatusID	Integer	11	1	99999999999	Actuator status identity number
Priority	Integer	11	1	99999999999	Priority number
* ActuatorID	Integer	11	1	99999999999	Actuator identity number

- Table 15: Data dictionary – regulation table

Table - room

Name	Type	Length	Min	Max	Description
* RoomID	Integer	11	1	99999999999	Room identity number
RoomName	Alphanumeric	100	//	//	Room name
RoomImage	Alphanumeric	100	//	//	Short path of room image

- Table 16: Data dictionary – room table

Table - sensor

Name	Type	Length	Min	Max	Description
* SensorID	Integer	11	1	99999999999	Sensor identity number
SensorPin	Integer	11	1	99999999999	Pin number in the sensor
SensorPinName	Alphanumeric	100	//	//	Pin name in the sensor
SensorName	Alphanumeric	100	//	//	Sensor name description
SensorImage	Alphanumeric	100	//	//	Short path of sensor image
* SensorTypeID	Integer	11	1	99999999999	Sensor type identity number
SensorValue	Integer	11	0	99999999999	Sensor current value
SensorUnit	Alphanumeric	100	//	//	Sensor unit (No unit : null)
* RoomID	Integer	11	1	99999999999	Room identity number
* DeviceID	Integer	11	1	99999999999	Control device identity number

- Table 17: Data dictionary – sensor table

Table - sensor_record

Name	Type	Length	Min	Max	Description
* RecordID	Integer	11	1	99999999999	Identify number of sensor record
DateTime	Datetime	//	//	//	Date time of the record
*SensorID	Integer	11	1	99999999999	Sensor identity number

- Table 18: Data dictionary – sensor_record table

P.S. * Red color with star symbol = primary key * black color with star symbol = foreign key

P.S. * Red color with star symbol = primary key * black color with star symbol = foreign key

Table - sensor_type

Name	Type	Length	Min	Max	Description
* SensorTypeID	Integer	11	1	99999999999	Sensor type identity number
SensorTypeDescription	Alphanumeric	100	//	//	Sensor type description

- Table 19: Data dictionary – sensor_type table

Table - time_setting

Name	Type	Length	Min	Max	Description
* TimeID	Integer	11	1	99999999999	Time identity number
StartTime	Time	//	//	//	Start time of setting
EndTime	Time	//	//	//	End time of setting
* ActuatorID	Integer	11	1	99999999999	Actuator identity number

- Table 20: Data dictionary – time_setting table

Table - user_account

Name	Type	Length	Min	Max	Description
* UserID	Integer	11	1	99999999999	User identity number
UserName	Alphanumeric	100	//	//	User name description
* UserTypeID	Integer	11	1	99999999999	User type identity number
LoginAccount	Alphanumeric	500	//	//	Login account name
Password	Alphanumeric	500	//	//	Login account password
Email	Alphanumeric	100	//	//	Login account email address
Token	Alphanumeric	200	//	//	Smart device token
Image	Alphanumeric	100	//	//	Short path of user image
ReceiveNotification	Integer	11	0	1	0: Not receive, 1: Receive
ReceiveEmail	Integer	11	0	1	0: Not receive, 1: Receive
LocationDisplay	Integer	11	0	1	0: Not display, 1: Display

- Table 21: Data dictionary – user_account table

Table - user_heart_rate

Name	Type	Length	Min	Max	Description
* HeartRateID	Integer	11	1	99999999999	Heart rate identity number
HeartRate	Integer	11	0	99999999999	Heart rate value
* UserID	Integer	11	1	99999999999	User identity number
DateTime	Datetime	//	//	//	Date time for record

- Table 22: Data dictionary – user_heart_rate table

P.S. * Red color with star symbol = primary key * black color with star symbol = foreign key

P.S. * Red color with star symbol = primary key * black color with star symbol = foreign key

Table - user_location

Name	Type	Length	Min	Max	Description
* UserLocationID	Integer	11	1	99999999999	User location identity number
Latitude	Decimal	8 , 5	00.00000	99.99999	Latitude value (location point x)
Longitude	Decimal	9 , 5	000.00000	999.99999	Longitude value (location point y)
* UserID	Integer	11	1	99999999999	User identity number
DateTime	Datetime	//	//	//	Date time for record

- Table 23: Data dictionary – user_location table

Table - user_type

Name	Type	Length	Min	Max	Description
* UserTypeID	Integer	11	1	99999999999	User type identity number
UserTypeDescription	Alphanumeric	100	//	//	User type description

- Table 24: Data dictionary – user_type table

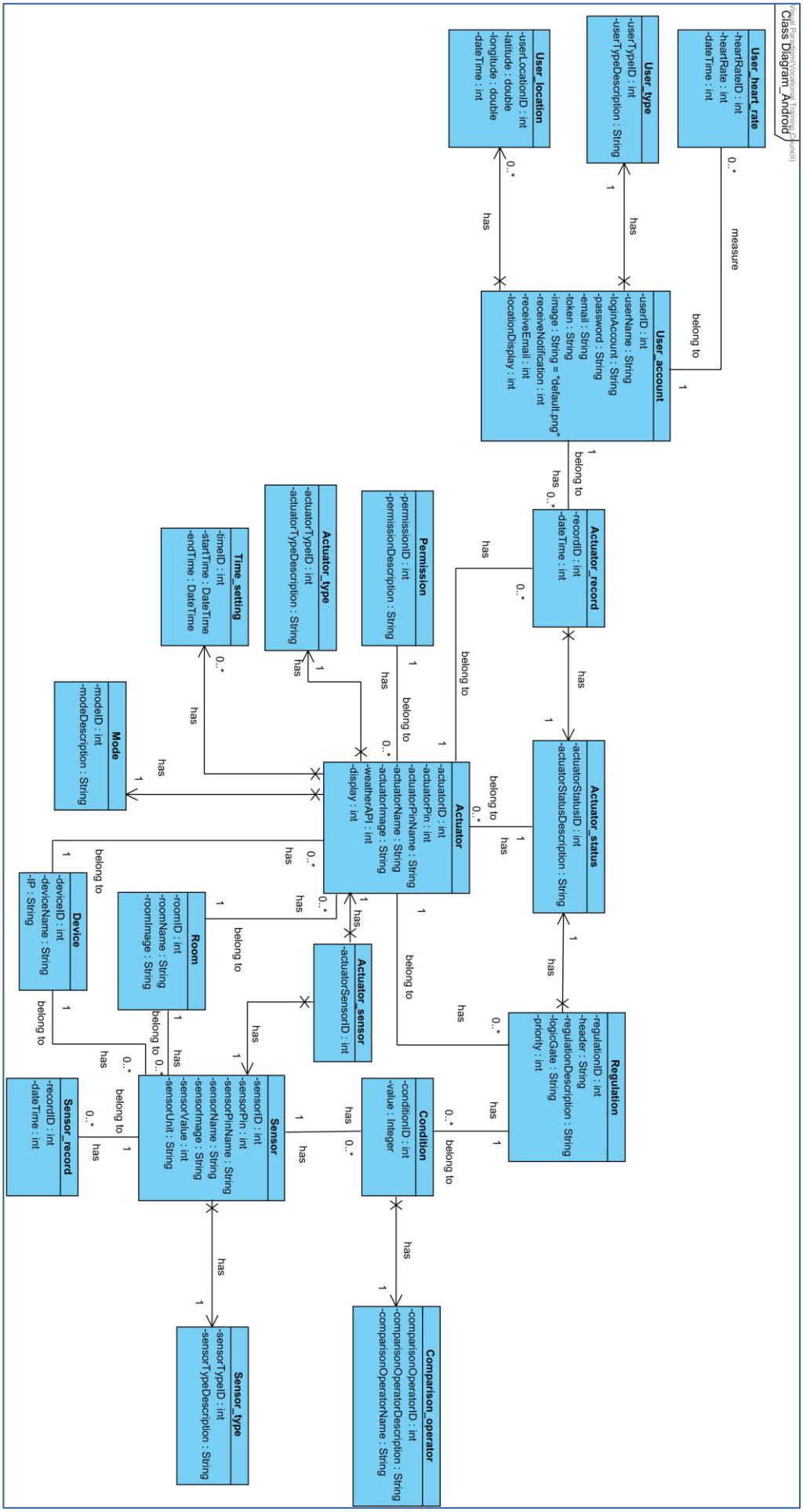
Table - weather_record

Name	Type	Length	Min	Max	Description
* RecordID	Integer	11	1	99999999999	Record identity number
Description	Alphanumeric	100	//	//	Weather record description
TimeStamp	Integer	11	0	99999999999	Time stamp for the record
CurrentTemp	Decimal	11 , 2	00000000.00	99999999.99	Current temperature in the record
MinTemp	Decimal	11 , 2	00000000.00	99999999.99	Minimum temperature in the record
MaxTemp	Decimal	11 , 2	00000000.00	99999999.99	Maximum temperature in the record
Humidity	Decimal	11 , 2	00000000.00	99999999.99	Current humidity in the record
Pressure	Decimal	11 , 2	00000000.00	99999999.99	Current pressure in the record
WindSpeed	Decimal	11 , 2	00000000.00	99999999.99	Current wind speed in the record
Rain	Decimal	11 , 2	00000000.00	99999999.99	Current rainfall in the record
Icon	Alphanumeric	100	//	//	Icon set in the record
UpdateDateTime	Datetime	//	//	//	Update date time for the record

- Table 25: Data dictionary – weather_record table

P.S. * Red color with star symbol = primary key * black color with star symbol = foreign key

5.1.3. Class diagram

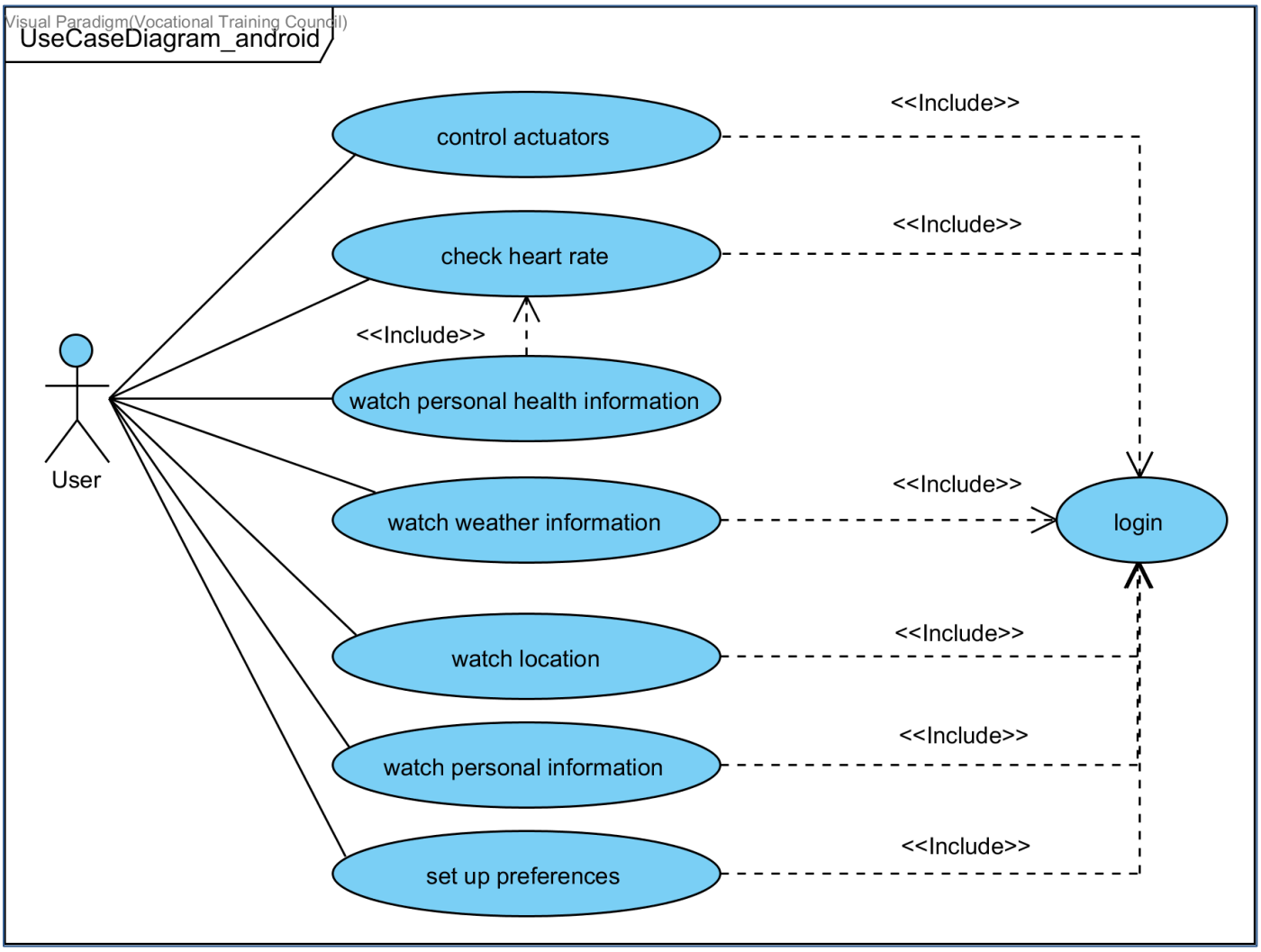


- Figure 3: Class diagram for digital home system

5.2. Functional model

5.2.1. Use case diagram

5.2.1.1. Use case diagram (Android application)



- Figure 4: Use case diagram – Android application

5.2.1.2. Actor description (Android application)

Actor	Definition
User (Common account)	The user can be a family member in a family, and who are able to control actuators via the Android application. When user login with their own personal account in mobile application, they can take the interactive operation with the system like browse his / her personal health information, check heart rate, browse weather information, browse weather information, browse user location, browse his / her personal information, and set up his / her preferences.

- Table 26: Actor description – Android application

5.2.1.3. Use case description (Android application)

Use case name :	**Login**
Use case ID :	UC-001
Primary actor :	User
Secondary actor :	N/A
Brief description :	User can sign in the mobile application by using the personal account.
Pre-conditions :	The people who have owned the system account.
Post-conditions :	The account id and password are verified as creatable.
Flow of events :	1. User enters username and password. 2. User clicks on the login button for asking sign in. 3. System verifies the information and gives respond.
Alternative flows and exceptions :	If the user account are not existing not are not valid, users are required to re-enter the username and password to re-login.
Priority :	High
Non-behavior requirements :	N/A

- Table 27: Use case description – Login (Android application)

Use case name :	**Control actuators**
Use case ID :	UC-002
Primary actor :	User
Secondary actor :	N/A
Brief description :	User can take interactive operation to control actuators via the mobile application.
Pre-conditions :	People who have owned the system account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User slides the screen to the proper location in main page. 3. User press the on / off switch to control actuators. 4. System gives respond for the actuators' setting.
Alternative flows and exceptions :	N/A.
Priority :	High
Non-behavior requirements :	N/A

- Table 28: Use case description – Control actuators (Android application)

Use case name :	**Watch personal health information**
Use case ID :	UC-003
Primary actor :	User
Secondary actor :	N/A
Brief description :	User can browse his / her health personal information via the mobile application.
Pre-conditions :	People who have signed in the system and have records of checked heart rate.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User slides the screen to the "Heart Rate" tab. 3. User selects the health history for browsing record. 4. System gives respond with the details for user's health record.
Alternative flows and exceptions :	N/A.
Priority :	High
Non-behavior requirements :	N/A

- Table 29: Use case description – Watch personal health information (Android application)

Use case name :	**Check heart rate**
Use case ID :	UC-004
Primary actor :	User
Secondary actor :	N/A
Brief description :	User can check heart rate via the mobile application.
Pre-conditions :	People who have owned the system account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User slides the screen to the “Heart Rate” tab. 3. User check heart rate function then contact with the heart rate sensor. 4. System gives details respond to the mobile application.
Alternative flows and exceptions :	If the heart rate sensor got no value, that user is required to restart the action of contact with the heart rate sensor by finger.
Priority :	High
Non-behavior requirements :	User must contact with the heart rate sensor accurately that heart rate sensor can get result effectively and accurately.

- Table 30: Use case description – Check heart rate (Android application)

Use case name :	**Watch weather information**
Use case ID :	UC-005
Primary actor :	User
Secondary actor :	N/A
Brief description :	User can browse weather information via the mobile application.
Pre-conditions :	People who have owned the system account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. System displays the weather information with temperature and humidity with the temperature sensors value in the house. 3. User slides the screen to the “Weather” tab. 4. System gives details respond with the weather information in the location when user living.
Alternative flows and exceptions :	N/A
Priority :	High
Non-behavior requirements :	N/A

- Table 31: Use case description – Watch weather information (Android application)

Use case name :	**Watch location**
Use case ID :	UC-006
Primary actor :	User
Secondary actor :	N/A
Brief description :	User can browse the location with other user who are installed the application.
Pre-conditions :	People who have owned the system account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User slides the screen to the “Location” tab. 3. System gives respond with current location and other users’ location.
Alternative flows and exceptions :	N/A
Priority :	High
Non-behavior requirements :	N/A

- Table 32: Use case description – Watch location (Android application)

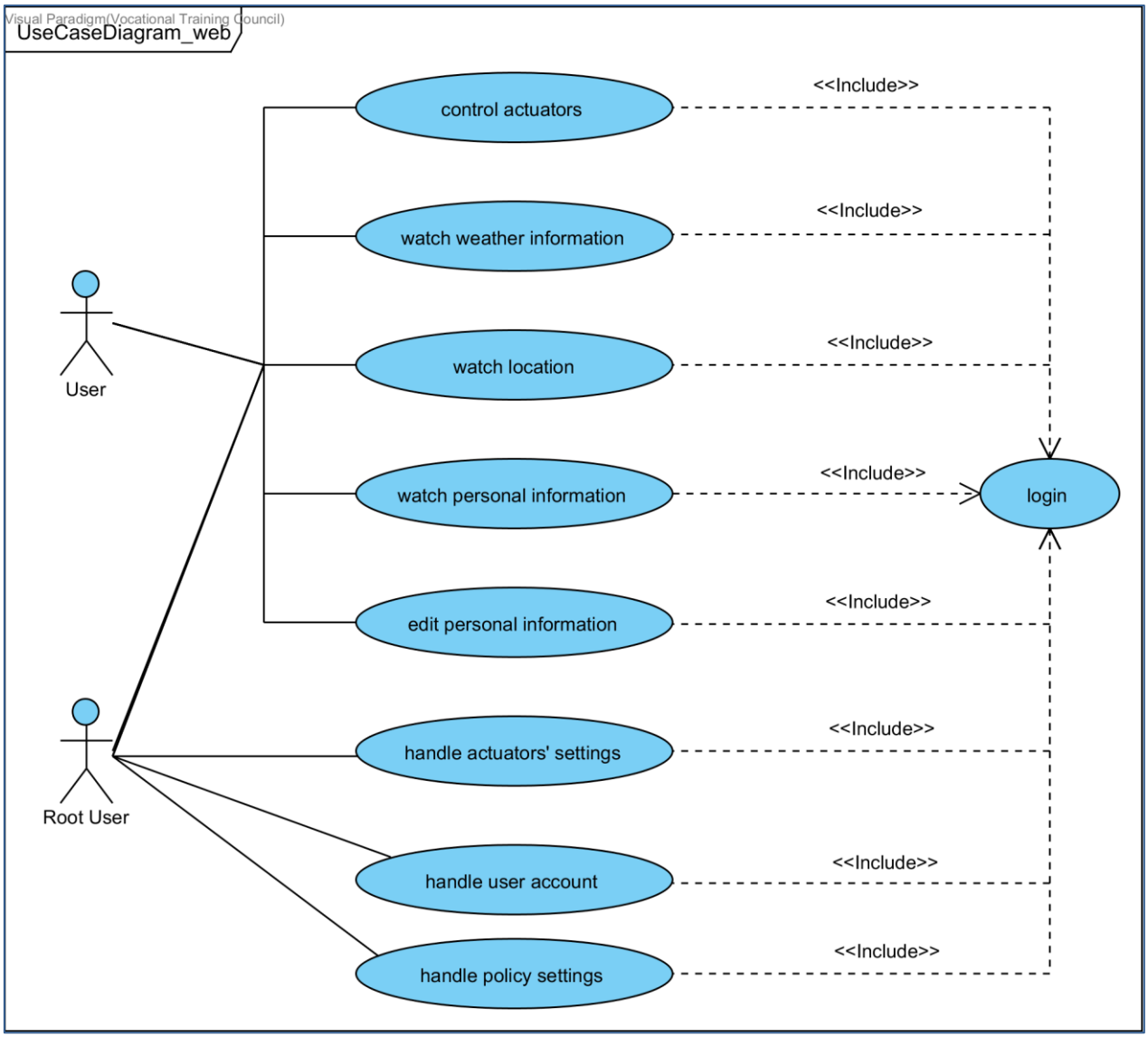
Use case name :	**Watch personal information**
Use case ID :	UC-007
Primary actor :	User
Secondary actor :	N/A
Brief description :	User can browse the personal account information via mobile application.
Pre-conditions :	People who have owned the system account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User slides the screen to the “Personal” tab. 3. System gives respond with the user account information.
Alternative flows and exceptions :	N/A
Priority :	High
Non-behavior requirements :	N/A

- Table 33: Use case description – Watch personal information (Android application)

Use case name :	**Set up preferences**
Use case ID :	UC-008
Primary actor :	User
Secondary actor :	N/A
Brief description :	User can set up preferences via mobile application.
Pre-conditions :	People who have owned the system account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User slides the screen to the preferences setting page 3. System gives respond of the preferences setting. 4. User selects the option of the preference setting. 5. System gives respond of the setting when completed action.
Alternative flows and exceptions :	N/A
Priority :	High
Non-behavior requirements :	N/A

- Table 34: Use case description – Watch personal information (Android application)

5.2.1.4. Use case diagram (Website)



- Figure 5: Use case diagram – Website

5.2.1.5. Actor description (Website)

Actor	Definition
User (Common account)	The user can be a family member in a family, and who are able to control actuators via the website. When user login with their own personal account in mobile application, they can take the interactive operation with the system like browse his / her personal health information, check heart rate, browse weather information, browse weather information, browse user location, browse his / her personal information, and set up his / her preferences.
Root User (Administrator account)	The root user look like a system administrator in a family that who have the additional user jurisdiction of handle actuators' settings in his / her house which installed the system and sensors, handle all user account in his / her family, and handle policy settings with whole system.

- Table 35: Actor description – Website

5.2.1.6. Use case description (Website)

Use case name :	**Login**
Use case ID :	UC-101
Primary actor :	User, Root User
Secondary actor :	N/A
Brief description :	User can sign in the system website by using the personal account.
Pre-conditions :	The people who have owned the system account.
Post-conditions :	The account id and password are verified as creatable.
Flow of events :	1. User enters username and password. 2. User clicks on the login button for asking sign in. 3. System verifies the information and gives respond.
Alternative flows and exceptions :	If the user account are not existing not are not valid, users are required to re-enter the username and password to re-login.
Priority :	High
Non-behavior requirements :	N/A

- Table 36: Use case description – Login (Website)

Use case name :	**Control actuators**
Use case ID :	UC-102
Primary actor :	User, Root User
Secondary actor :	N/A
Brief description :	User can take interactive operation to control actuators via the system website.
Pre-conditions :	People who have owned the system account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User press the on / off switch to control actuators. 3. System gives respond for the actuators' setting.
Alternative flows and exceptions :	N/A.
Priority :	High
Non-behavior requirements :	N/A

- Table 37: Use case description – Control actuators (Website)

Use case name :	**Watch weather information**
Use case ID :	UC-103
Primary actor :	User, Root User
Secondary actor :	N/A
Brief description :	User can browse weather information via the System website.
Pre-conditions :	People who have owned the system account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. System displays the weather information with temperature and humidity with the temperature sensors value in the house. 3. User slides the screen to the "Weather" tab. 4. System gives details respond with the weather information.
Alternative flows and exceptions :	N/A
Priority :	High
Non-behavior requirements :	N/A

- Table 38: Use case description – Watch weather information (Website)

Use case name :	**Watch location**
Use case ID :	UC-104
Primary actor :	User, Root User
Secondary actor :	N/A
Brief description :	User can browse the location with other user via the system website.
Pre-conditions :	People who have owned the system account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User move to the “Location” tab. 3. System gives respond with current location and other users’ location.
Alternative flows and exceptions :	N/A
Priority :	High
Non-behavior requirements :	N/A

- Table 39: Use case description – Watch location (Website)

Use case name :	**Watch personal information**
Use case ID :	UC-105
Primary actor :	User, Root User
Secondary actor :	N/A
Brief description :	User can browse the personal account information via system website.
Pre-conditions :	People who have owned the system account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User move to the “Personal” tab icon. 3. System gives respond with the user account information.
Alternative flows and exceptions :	N/A
Priority :	High
Non-behavior requirements :	N/A

- Table 40: Use case description – Watch personal information (Website)

Use case name :	**Edit personal information**
Use case ID :	UC-106
Primary actor :	User, Root User
Secondary actor :	N/A
Brief description :	User can edit the personal account information via system website.
Pre-conditions :	People who have owned the system account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User move to the “Personal” tab icon. 3. User enters the details of modification. 4. System gives respond with the user account information.
Alternative flows and exceptions :	If the data are not valid or missing, user is required to re-enter for the related details of modification for their personal information.
Priority :	High
Non-behavior requirements :	N/A

- Table 41: Use case description – Edit personal information (Website)

Use case name :	**Handle actuators' settings**
Use case ID :	UC-107
Primary actor :	Root User
Secondary actor :	N/A
Brief description :	Root user can modify for the actuators' settings in the house via system website.
Pre-conditions :	People who have root user account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User move to the area option button: "Living Room" or "Kitchen". 3. User modifies the on / off switch option or time range setting. 4. System gives respond with the actuators' setting.
Alternative flows and exceptions :	N/A
Priority :	High
Non-behavior requirements :	N/A

- Table 42: Use case description – Handle actuators settings (Website)

Use case name :	**Handle user account**
Use case ID :	UC-108
Primary actor :	Root User
Secondary actor :	N/A
Brief description :	Root user can modify for the user account in the system via system website.
Pre-conditions :	People who have root user account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User move to the user account management page. 3. User modifies the details of user account 4. System gives respond with the user account setting.
Alternative flows and exceptions :	If the data are not valid or missing, user is required to re-enter for the related details of modification for their personal information.
Priority :	High
Non-behavior requirements :	N/A

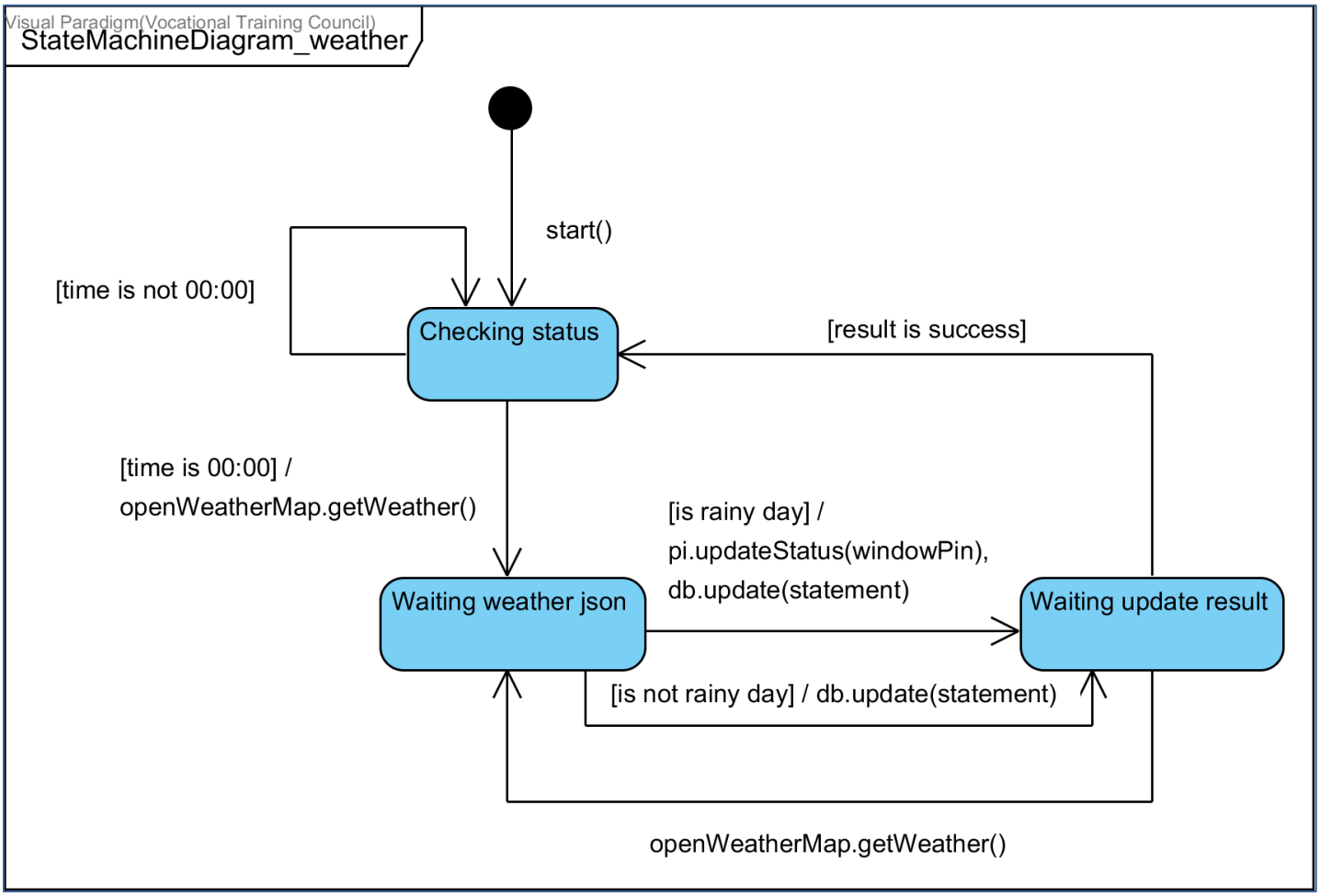
- Table 43: Use case description – Handle user account (Website)

Use case name :	**Handle policy settings**
Use case ID :	UC-109
Primary actor :	Root User
Secondary actor :	N/A
Brief description :	Root user can manage the policy settings in the system via system website.
Pre-conditions :	People who have root user account and who have signed in the system.
Post-conditions :	N/A.
Flow of events :	1. Include [Login]. 2. User move to the policy management page. 3. User creates / update / delete of policy for sensors control 4. System gives respond with the policy setting.
Alternative flows and exceptions :	N/A
Priority :	High
Non-behavior requirements :	N/A

- Table 44: Use case description – Handle policy settings (Website)

5.2.2. Dynamic model

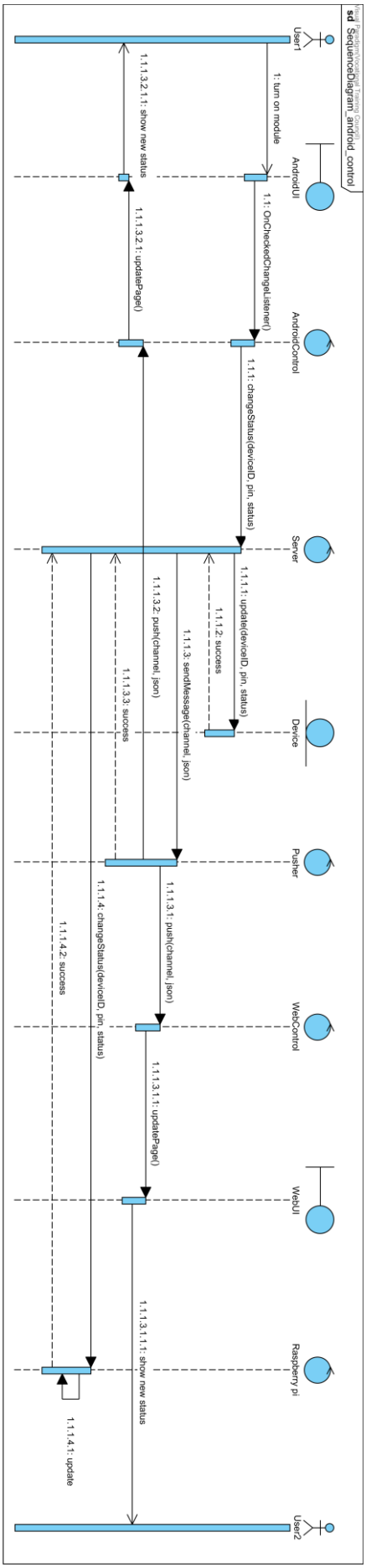
5.2.2.1. State machine diagram



- Figure 6: State machine diagram – weather

5.2.2.2. Sequence diagram

5.2.2.2.1. Android application



Description:

After a user switches on an actuator, the request will be sent to server. The server will update the actuator status in the database. At the same time, the server will push the request to the Pusher. Finally, Pusher will send the message typed of json to all clients side.

- Figure 7: Sequence diagram – android control

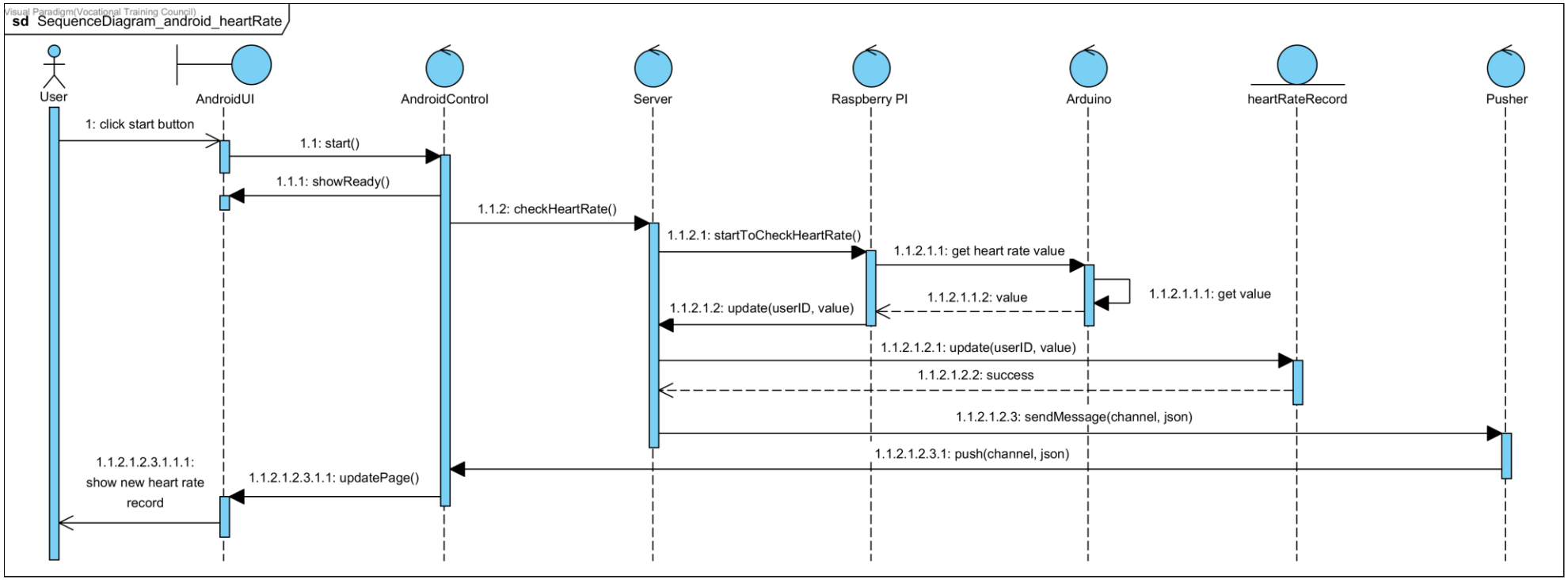
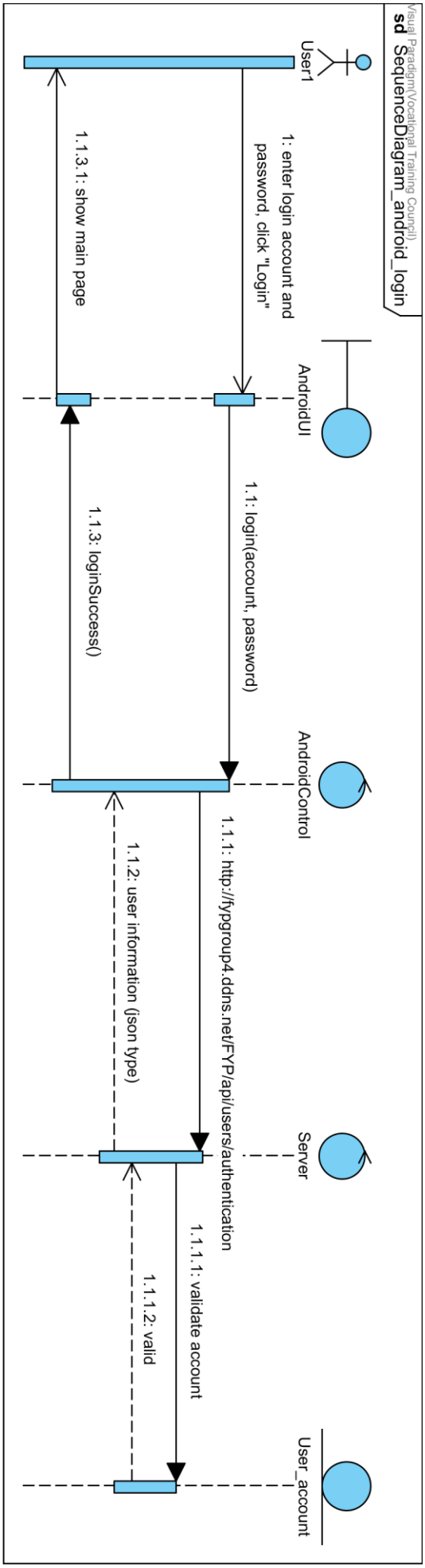


Figure 8: Sequence diagram – android heart rate

Description:

After a user click the start button, the request will be sent to the server. The server will send a start message to the raspberry PI for starting measure. After getting heart rate value from Arduino, the raspberry PI will send the value to the server via API, the server update the value to the database and push the message to the Pusher. Finally, Pusher will send the message to the client to show the value.

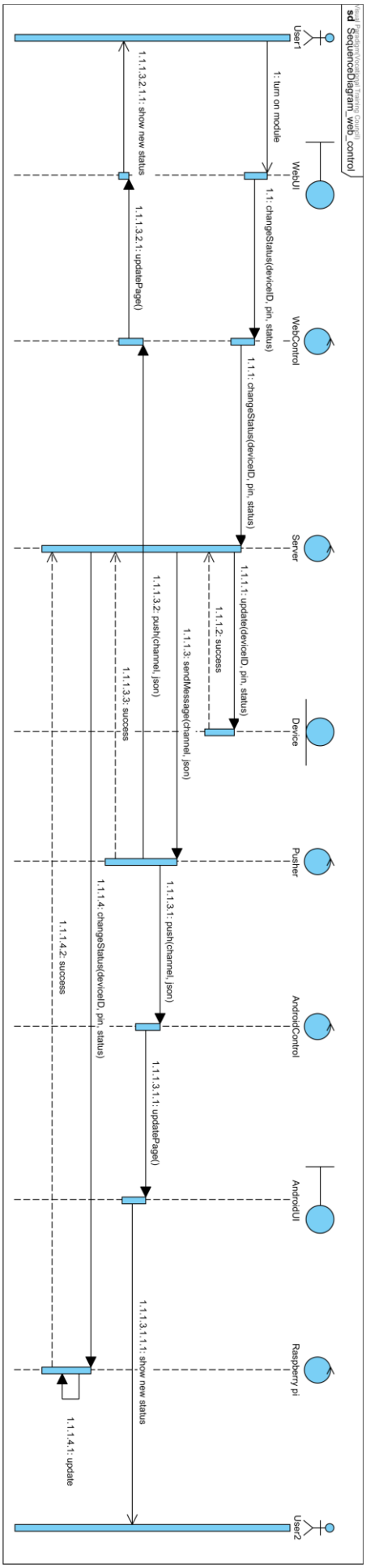


Description:

After a user enter login information and click the Login button, the request will be sent to the server via API. The server will validate the account. If valid, some personal data will be returned without password for display.

- Figure 9: Sequence diagram – android login

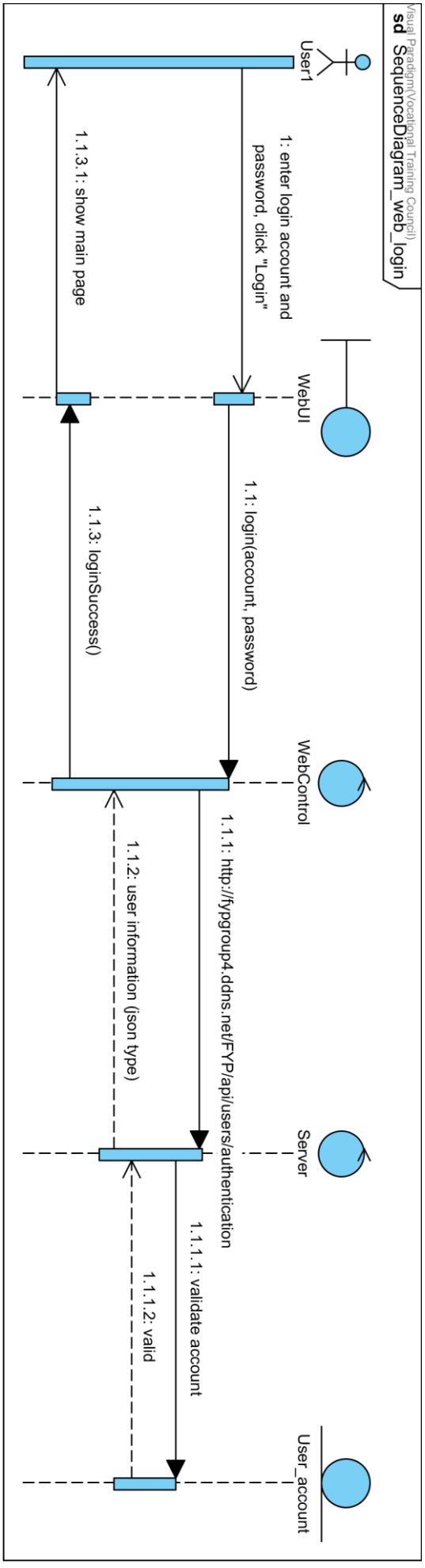
5.2.2.2.2. Website



Description:

After a user switches on an actuator, the request will be sent to server. The server will update the actuator status in the database. At the same time, the server will push the request to the Pusher. Finally, Pusher will send the message typed of json to all clients side.

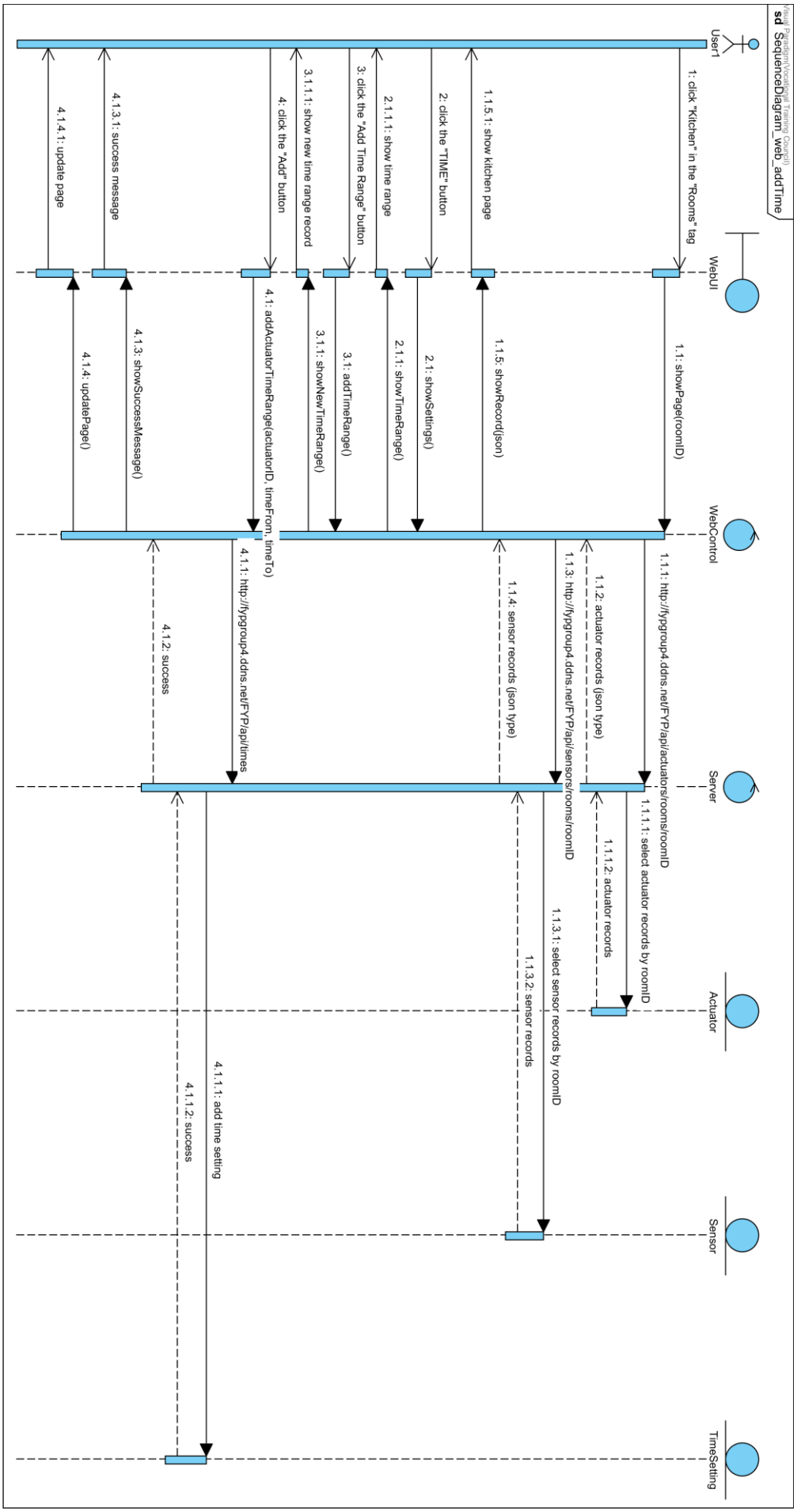
- Figure 10: Sequence diagram – web control



Description:

After a user enter login information and click the Login button, the request will be sent to the server via API. The server will validate the account. If valid, some personal data will be returned without password for display.

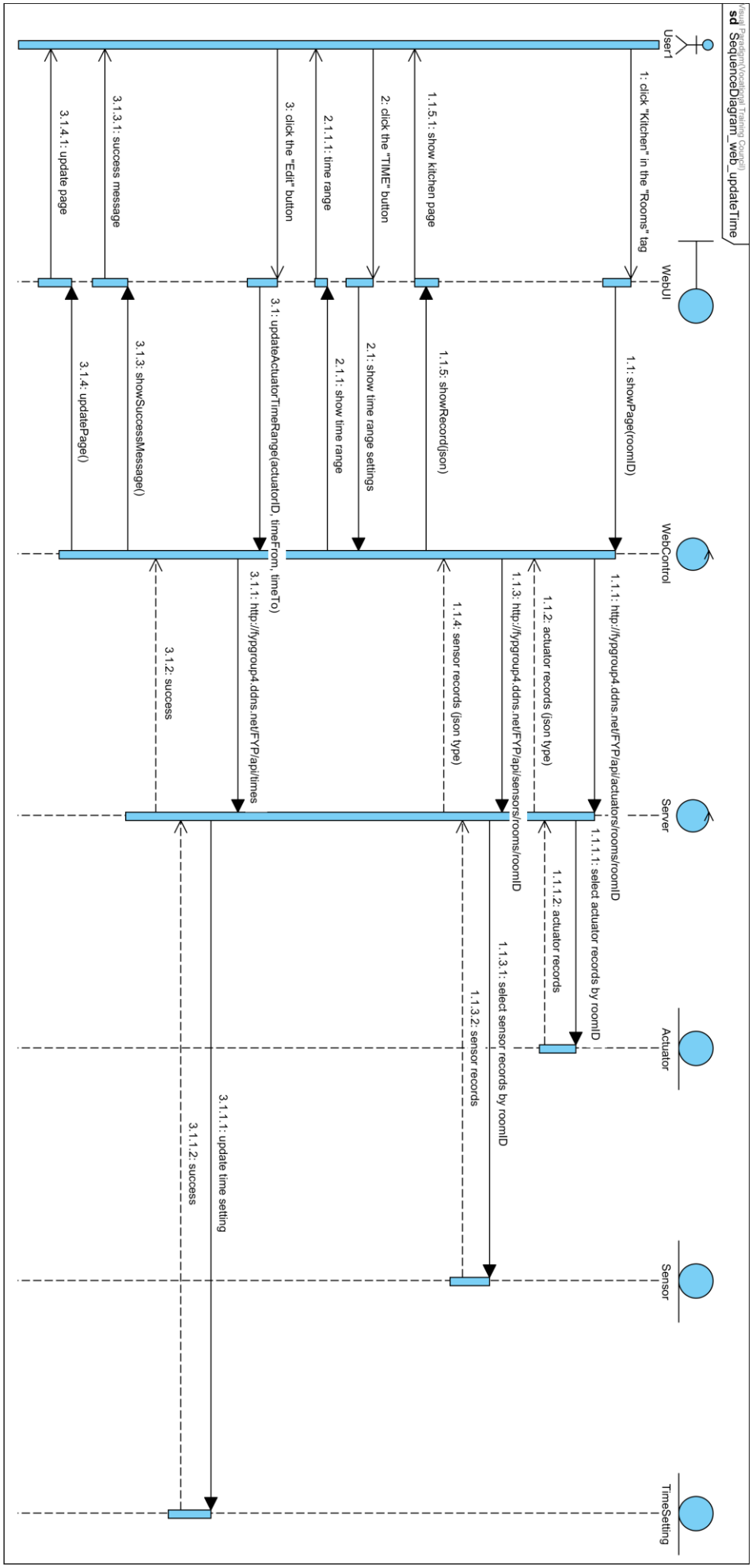
- Figure 11: Sequence diagram – web login



Description:

User can add a new time setting for actuator. When click the Add button after inputting the time, the WebControl will send the request to the server via API. The server will then add it to the database and return a related message to the user.

- Figure 12: Sequence diagram – web add time



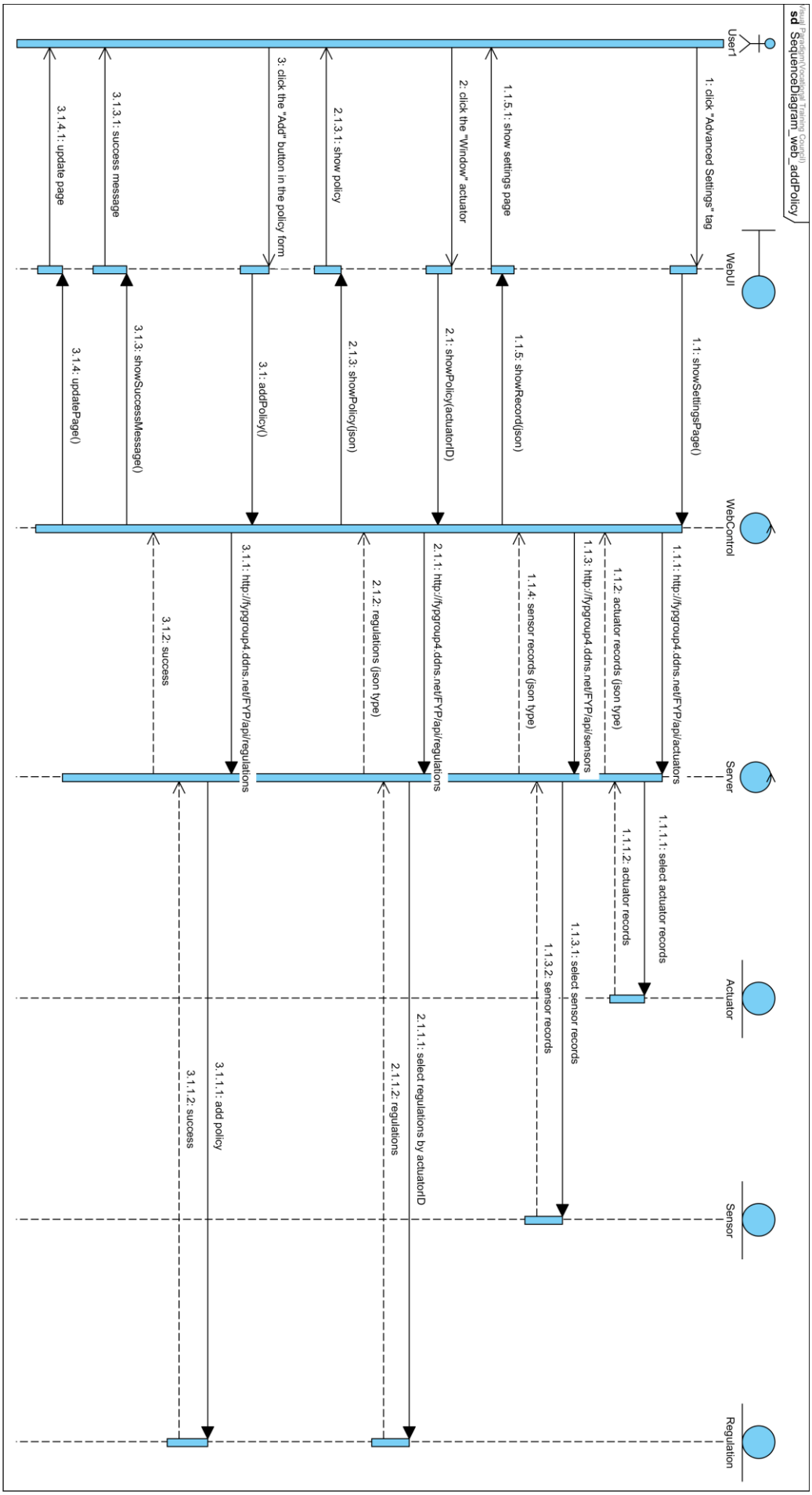
Description:

User can edit an existed time setting for actuator. When clicking the Edit button, the WebControl will send the update request to the server via API. The server will then update the actuator settings and return a related message to the user.

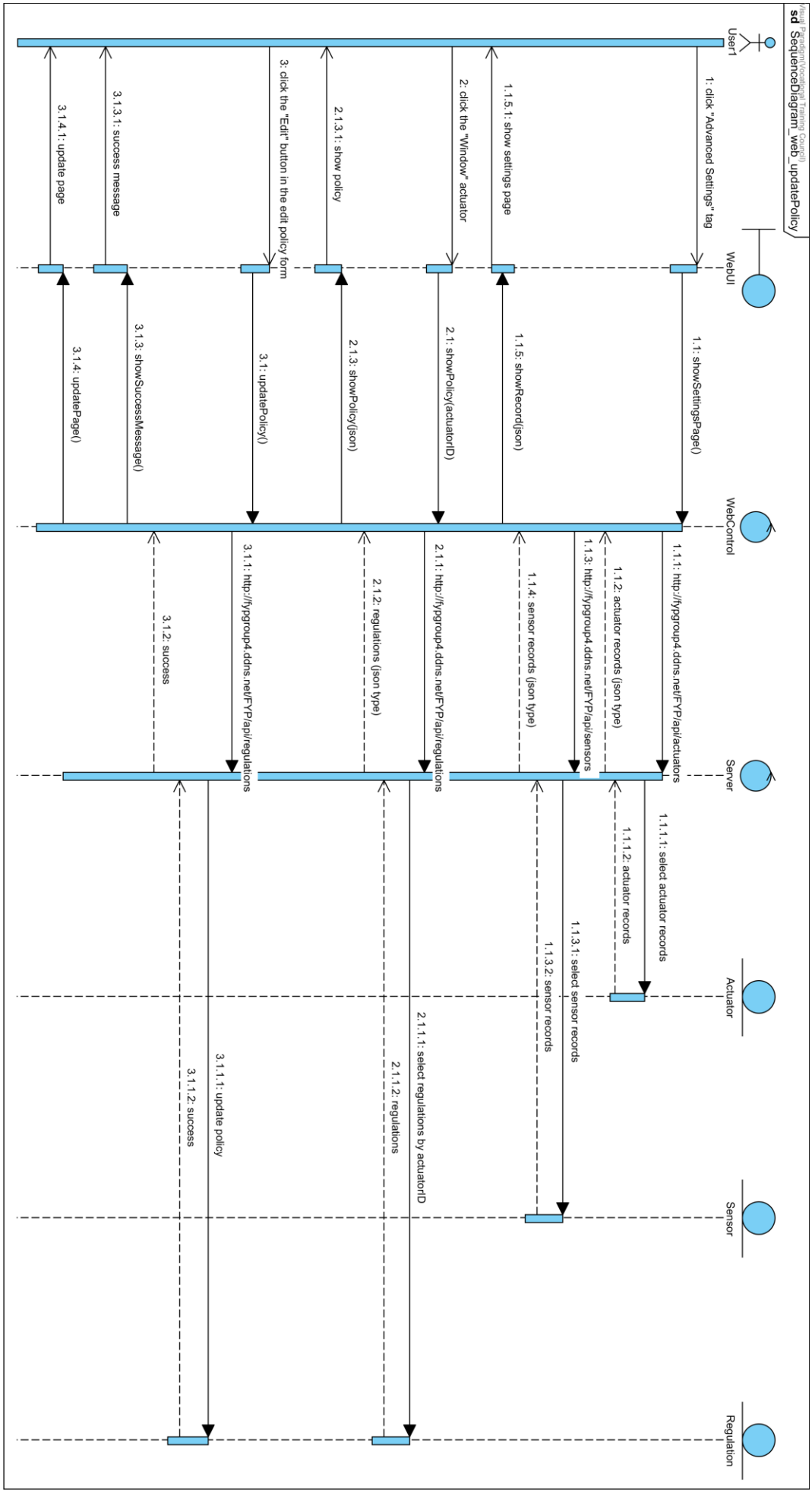
- Figure 13: Sequence diagram – web update time

Description:

User can add some new policies for connecting the sensors and actuators. When clicking the Add button, the WebControl will send the add policy request to the server via API. The server will then store it to the database and return a related message to the user.



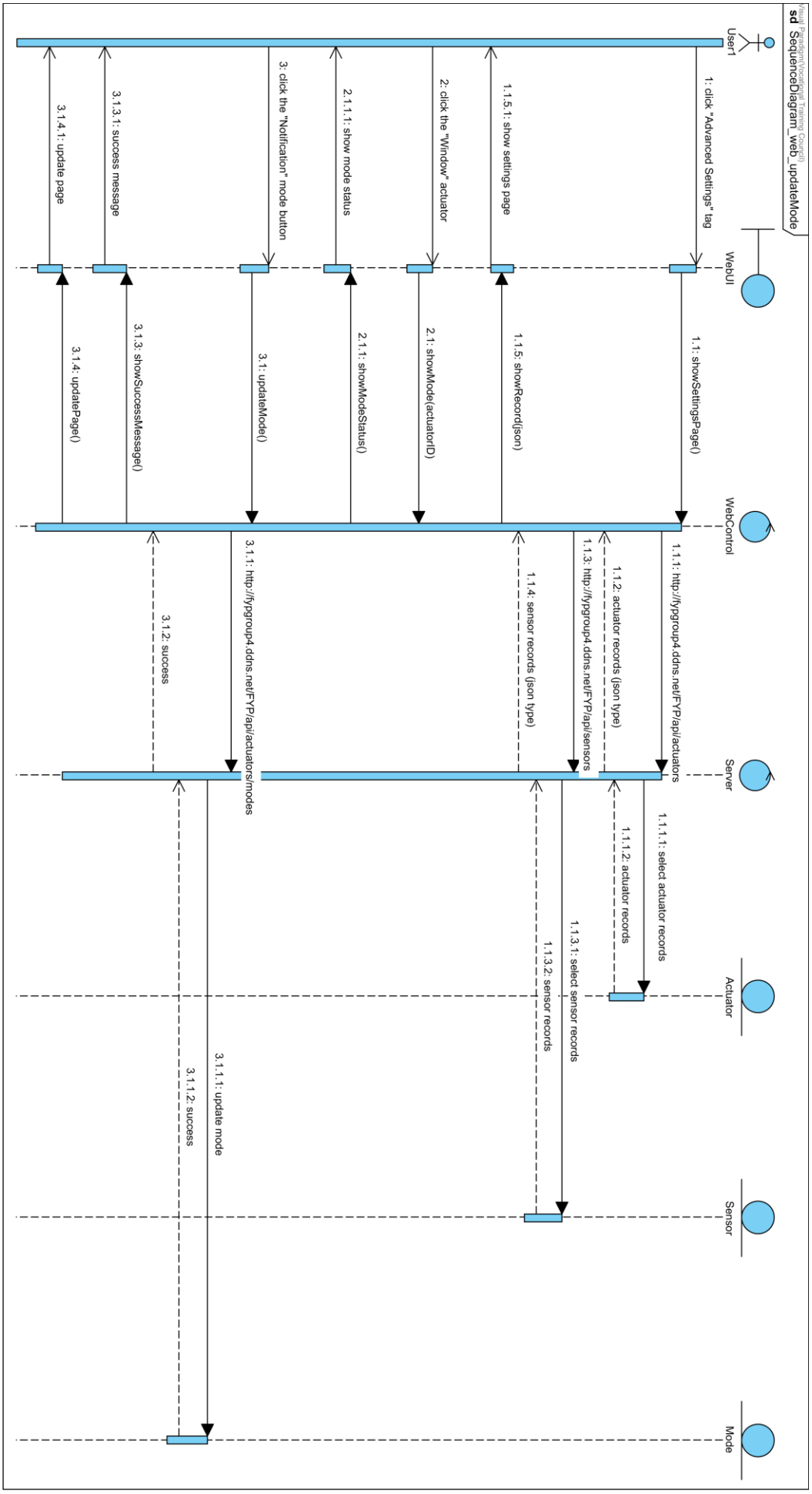
- Figure 14: Sequence diagram – web add policy



Description:

User can edit some existed policies. When clicking the Edit button, the WebControl will send the update request to the server via API. The server will then update it and return a related message to the user.

- Figure 15: Sequence diagram – web update policy



- Figure 16: Sequence diagram – web update mode

Description:

User can change the mode settings for actuator such as auto change performed by sensors, notifying user for user control, etc. When clicking the Notification Mode, the WebControl will send the update request to the server via API. The server will then update the mode and return a related message to the user.

5.3. Detailed design

5.3.1. System design

5.3.1.1. Software architectural design

1. Software requirement

1.1. Arduino Software (1.6.13)

- 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. And this software can be used with any Arduino board.

1.2. Android Studio

- Android Studio provides the fastest tools for building apps on every type of Android device. And it is a world-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow programmer to focus on building unique and high quality apps.

2. Required implementation language

2.1. Android Studio (Java and xml)

- Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form. XML is a popular format for sharing data on the internet. Websites that frequently update their content, such as news sites or blogs, often provide an XML feed so that external programs can keep abreast of content changes. Uploading and parsing XML data is a common task for network-connected apps. This lesson explains how to parse XML documents and use their data.

2.2. Swift (Objective-C)

- Apple laid the foundation for Swift by advancing our existing compiler, debugger, and framework infrastructure. We simplified memory management with Automatic Reference Counting (ARC). Our framework stack, built on the solid base of Foundation and Cocoa, has been modernized and standardized throughout. Objective-C itself has evolved to support blocks, collection literals, and modules, enabling framework adoption of modern language technologies without disruption. Thanks to this groundwork, we can now introduce a new language for the future of Apple software development.

2.3. Website – Content Management System (HTML5, CSS, JavaScript)

- Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

5.3.1.2. Hardware architectural design

1. Hardware requirement
    - **1.1.** Sensor and control panel
    - **1.2.** Sensor (METAS sensors and ARDUINO sensors)
    - **1.3.** Control panel (METAS Node One or ARDUINO UNO or raspberry PI)

2. Testing devices with three platform

    - Android device (Samsung Galaxy Note 2 LTE)
    - Computer (Microsoft Windows 7/8.1/10, Mac OS)
    - Web browser (Google Chrome, Firefox, Internet Explorer 11, Safari)

3. Future extension

    1. **Temperature adjustment**
        - To switch on or switch off the air conditioner via the smartphone, to pre-adjust the room temperature on the way.

    2. **Environment protection and energy saving**
        - Automatic adjustment of the indoor lighting and temperature with the outdoor intensity. And the action achieve energy saving and environmental protection effects.

    3. **Compatible with different working environment and scenes**
        - It can have an all-for-one control of various compatible and which has been connected to the intelligent system of lighting, curtains and air conditioning and ventilation equipment. Also with the property management system connection, easy to control, that it is easy to control in the office area.

1. Required sensors and devices

Product Name	 **Raspberry Pi 3 Model B** ▪ Figure 17: Required sensors and devices – Raspberry Pi
Features	Built on the latest Broadcom 2837 ARMv8 64bit processor, the new generation Raspberry Pi 3 Model B is faster and more powerful than its predecessors. With built-in wireless and Bluetooth connectivity, it becomes the ideal IoT ready solution.
Role	1. Accept server subscribe from pusher for controlling sensors and modules. 2. Send the API updated sensors value to the server.

▪ Table 45: Required sensors and devices – Raspberry Pi

Product Name	 **Arduino : Arduino Uno SMD R3** ▪ Figure 18: Required sensors and devices – Servos
Features	Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers.
Role	1. Use for take connection with heart rate sensor.

▪ Table 46: Required sensors and devices – Servos

Product Name	 **PWM Servos : FS90MG Micro Metal Gear Servo** ▪ Figure 19: Required sensors and devices – Servos
Features	Made by plastic body and metal gears that easy to do the switch action.
Role	1. Use for the door or windows to be a hinges.

▪ Table 47: Required sensors and devices – Servos

Product Name	 **Arduino Switches & Physical Input : FS90MG RobotGeek Pushbutton** ▪ Figure 20: Required sensors and devices – Switch
Features	The RobotGeek Pushbutton is a self-contained Arduino Button board that makes it easy to add a pushbutton to your Arduino Project.
Role	1. Use for the doorbell and sensors on / off switch.

▪ Table 48: Required sensors and devices – Switch

Product Name	 **Sensor Accessories: Basic Components Mixed Pack** ▪ Figure 21: Required sensors and devices – Sensor Accessories
Features	Resistors and LEDs - these components are found in almost every electronic device known to man. This pack is a perfect mix of these components, so that you'll be ready for all sorts of projects.
Role	1. Use for displaying the status of the sensors.

▪ Table 49: Required sensors and devices – Sensor Accessories

Product Name	 **Cameras & Vision : Raspberry Pi camera** ▪ Figure 22: Required sensors and devices – Camera
Features	Vision Processing is a lot of work - even low resolution camera can output lots of data, and parsing through that data can be a lot of work.
Role	1. Use for capture when fire detector detected a fire that remind for users.

▪ Table 50: Required sensors and devices – Camera

Product Name	 **Distance & Object Detection : Ultrasonic Range Measurement Module** ▪ Figure 23: Required sensors and devices – Distance detection
Features	This is a non-contact distance measurement module designed for use.
Role	1. Use for the human / object detector for safety use.

▪ Table 51: Required sensors and devices – Distance detection

Product Name	 **Distance & Object Detection : Phidgets IR Reflective Sensor 10cm** ▪ Figure 24: Required sensors and devices – Object detection
Features	This sensor can detect the presence of a highly reflective object at 10cm, for less reflective objects (like hands) the distance is about 5cm.
Role	2. Use for the automatically flushing in the toilet.

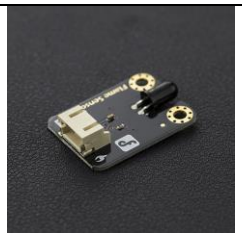
▪ Table 52: Required sensors and devices – Object detection

Product Name	 **Displays, Lights, Buzzers, Relays : RobotGeek Buzzer** ▪ Figure 25: Required sensors and devices – RobotGeek Buzzer
Features	The RobotGeek Buzzer is an easy way to add audio output to your Arduino. By simply sending a 5v signal to the RobotGeek Buzzer, the device will emit a tone. You can also use the Arduino Tone Function to send Different tones to the buzzer
Role	1. Issue alarm sound warning user with somethings else.

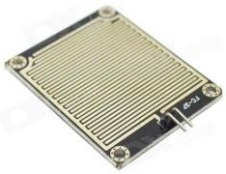
▪ Table 53: Required sensors and devices – RobotGeek Buzzer

Product Name	 **Gravity: DHT11 Temperature and Humidity Sensor** ▪ Figure 26: Required sensors and devices – Temperature sensor
Features	The DHT11 has a full range temperature compensation, low power consumption, long term stability and calibrated digital signal.
Role	1. Use for the temperature and humidity detect in the digital home.

▪ Table 54: Required sensors and devices – Temperature sensor

Product Name	 **Gravity: Analog Flame Sensor** ▪ Figure 27: Required sensors and devices – Analog flame sensor
Features	The Analog Flame Sensor can be used to detect fire or other light at wavelengths of 760nm ~ 1100nm.
Role	1. Use for the flame detect in the kitchen and the house.

▪ Table 55: Required sensors and devices – Analog flame sensor

Product Name	 **Gravity: Analog Rain Sensor** ▪ Figure 28: Required sensors and devices – Analog rain sensor
Features	The Analog Rain Sensor can be used to detect Rain rate one the board.
Role	1. Use for the rain detect outside for closing the windows.

▪ Table 56: Required sensors and devices – Analog rain sensor

Product Name	 **Gravity: Heart rate sensor** ▪ Figure 29: Required sensors and devices – Heart rate sensor
Features	A heart rate monitor allows one to measure one's heart rate in real time
Role	1. Use for checking user heart rate.

▪ Table 57: Required sensors and devices – Heart rate sensor

Product Name	 **Gravity: CO Gas Sensor Module Detector** ▪ Figure 30: Required sensors and devices – CO Gas Sensor
Features	Output voltage of carbon monoxide boosts along with the concentration of the measured gases increases
Role	1. Use for detect the Co Gas in the bathroom.

▪ Table 58: Required sensors and devices – CO Gas Sensor

Product Name	 **Motion Sensor: Parallax PIR Motion Sensor** ▪ Figure 31: Required sensors and devices – Motion sensor
Features	The PIR Sensor detects motion up to 20 feet away to detect changing patterns of passive infrared emitted by objects in its vicinity.
Role	1. Use for the automatically lighting in digital home system.

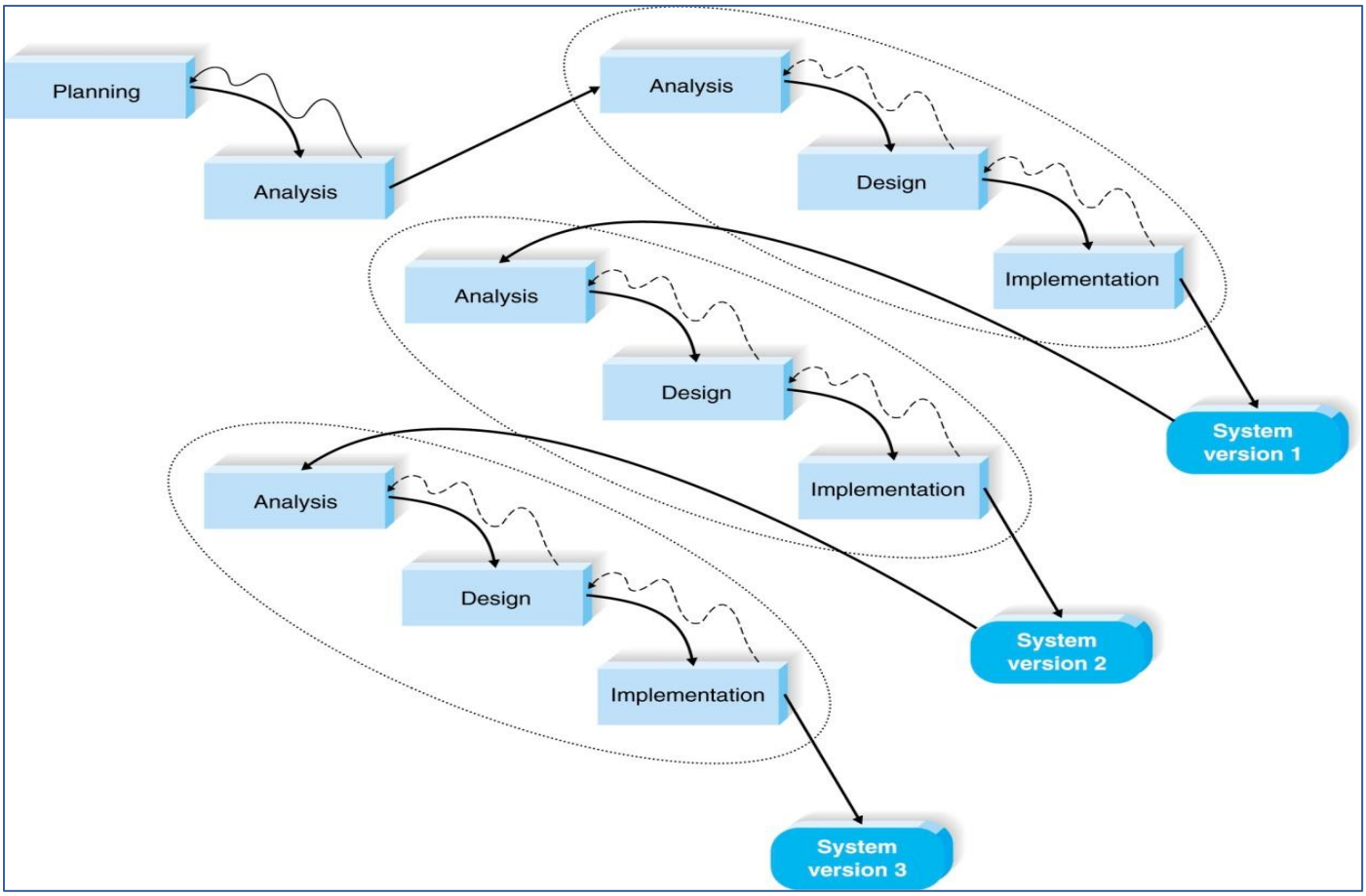
▪ Table 59: Required sensors and devices – Motion sensor

5.3.1.3. Procedural design

For the methodology or approach adopted for developing the digital home system, we decide to use "**Iterative Development**" and it will take evolution to "**System Prototyping**".

Iterative development is a way of breaking down the software development of a large application into smaller chunks. In iterative development, feature code is designed, developed and tested in repeated cycles. With each of iteration, additional features can be designed, developed and tested until there is a fully functional software application ready to be deployed to customers.

Iterative development is key practices in Agile development methodologies. The purpose of working iteratively is to allow more flexibility for changes but it may create any step of system analysis and system version which is unnecessary. In view of this case, we decide to take evolution to System Prototyping.



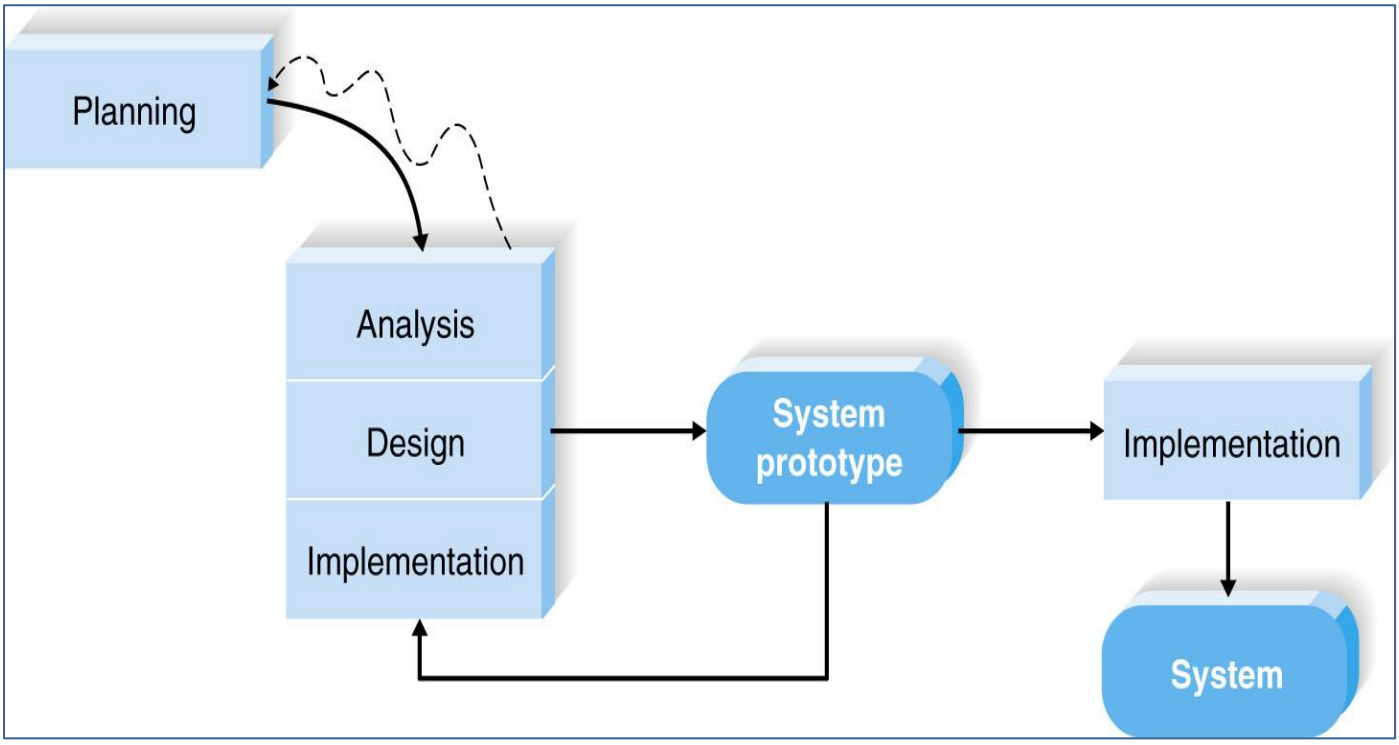
- Figure 32: Rapid Application Development – Iterative Development

System Prototyping is the activity of creating prototypes of software applications, and a prototype typically simulates only a few aspects of, and may be completely different from, the final product.

Prototyping has several benefits:

- The software designer and implementer can get valuable feedback from the users early in the project.
- The client and the contractor can compare if the software made matches the software specification, according to which the software program is built.
- It allows the software engineer some insight into the accuracy of initial project estimates and whether the deadlines and milestones proposed can be successfully met.

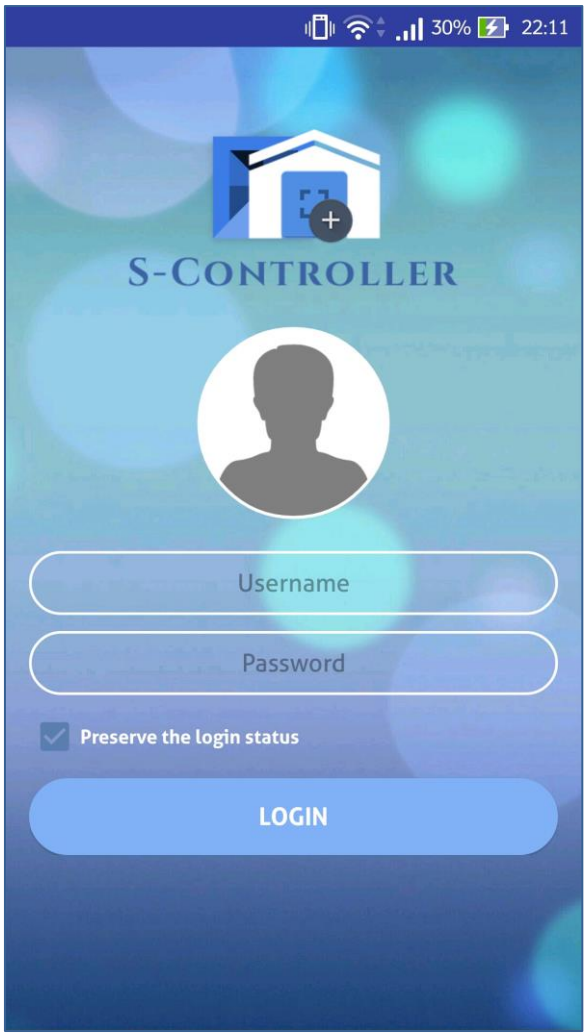
In view of the project plan for the digital home system, we may need to modify at any time that the system prototype should be already change. Also, System Prototyping is done well for the system with unclear user requirement and with short time schedule. That System Prototyping may be a good choice for the software process model.



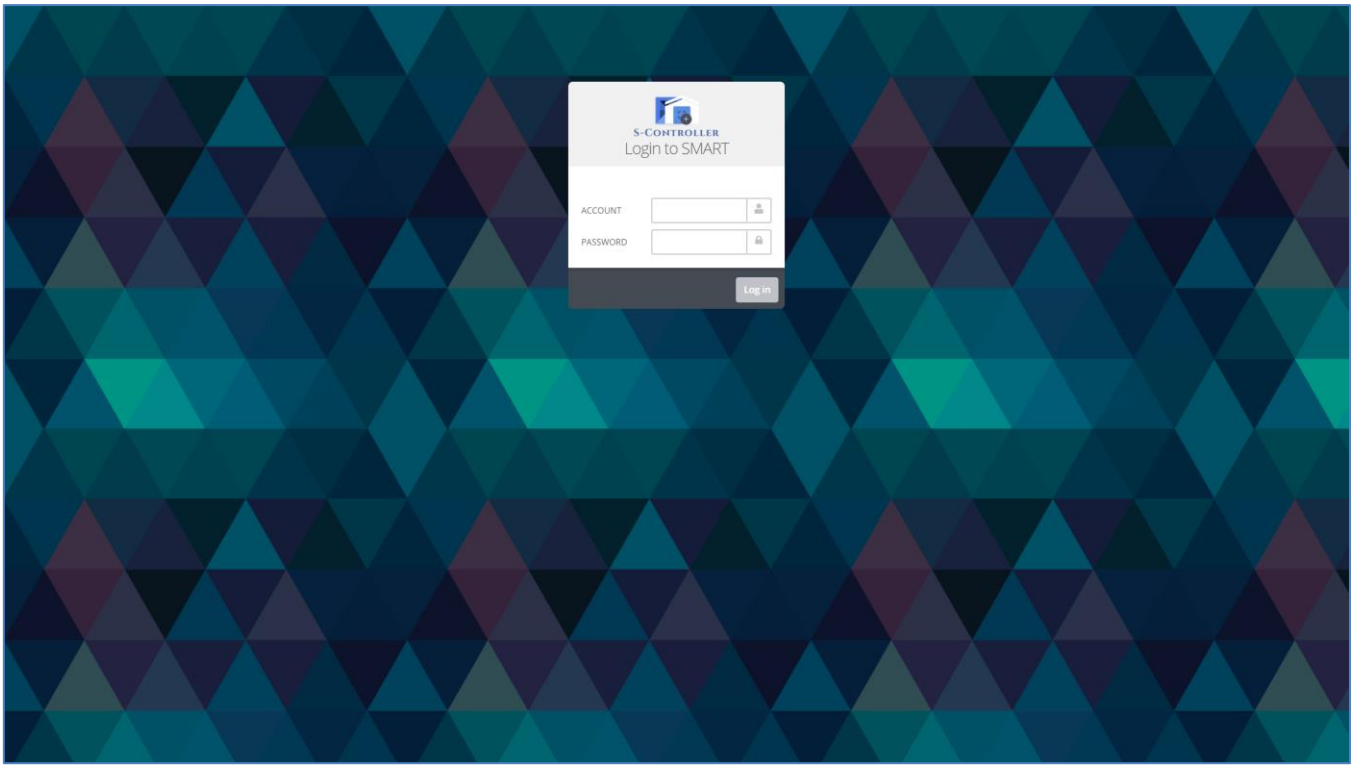
- Figure 33: Rapid Application Development – System Prototyping

5.3.1.4. User interface design

5.3.1.4.1. Login page

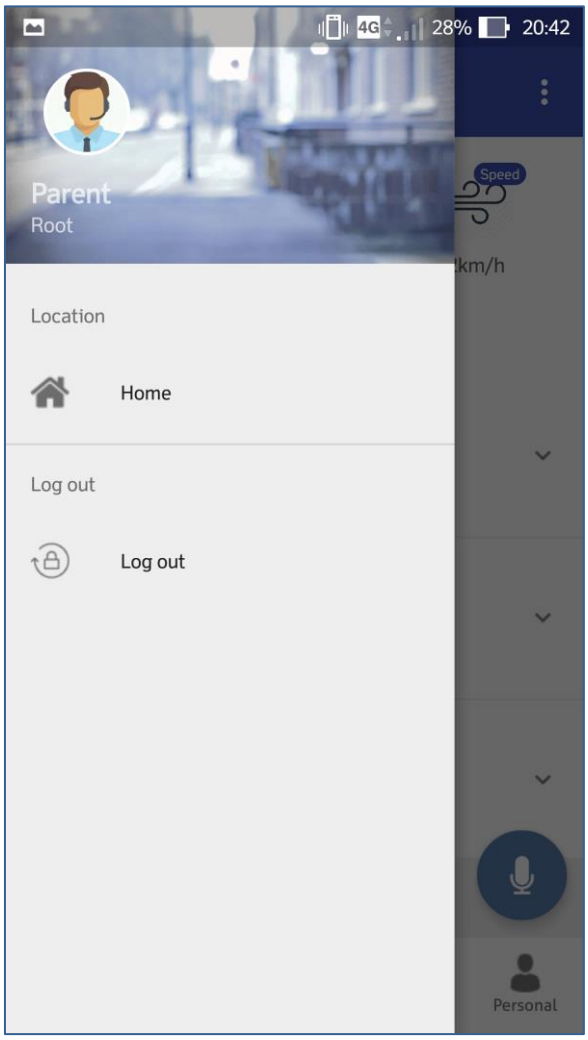


- Figure 34: User interface design – Login page (Android application)

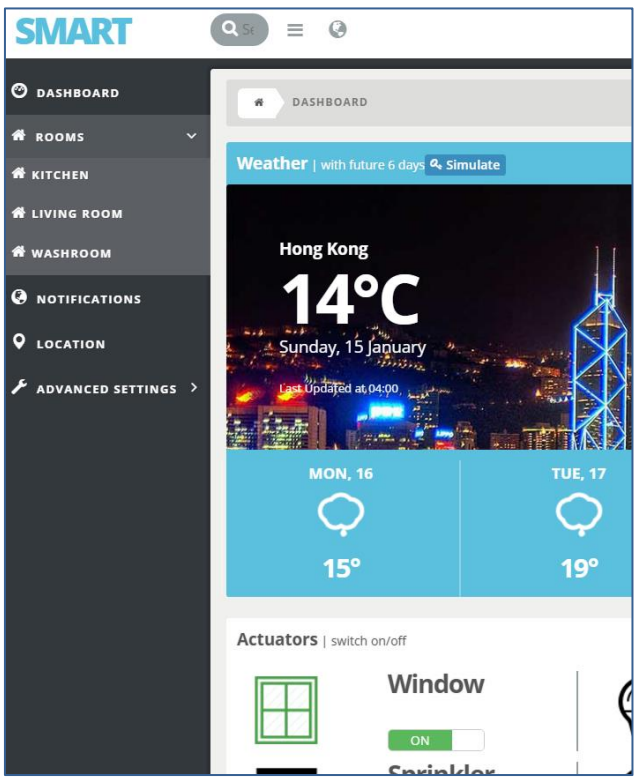


- Figure 35: User interface design – Login page (Website)

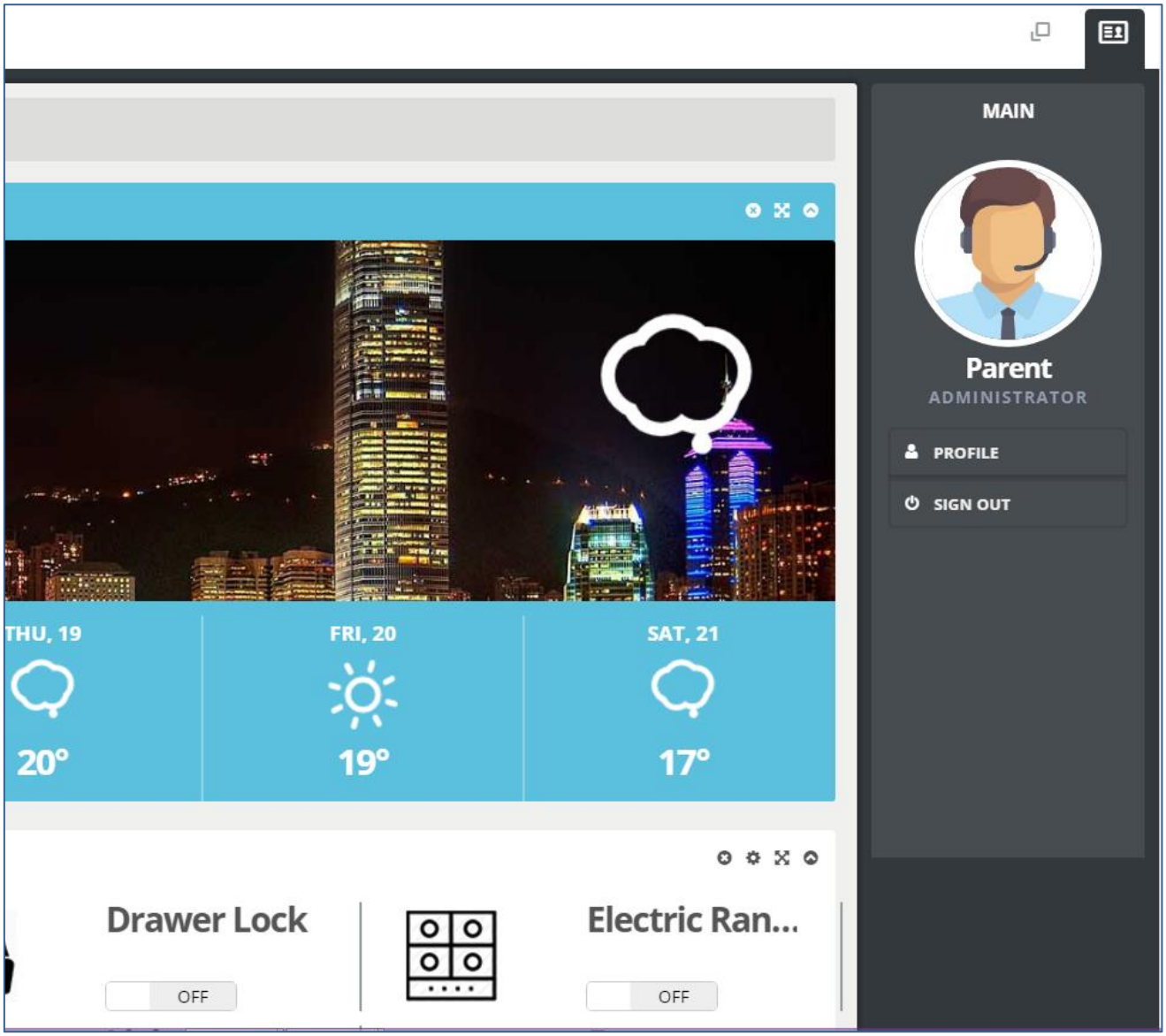
5.3.1.4.2. Navigation bar



- Figure 36: User interface design – Navigation Bar (Android application)

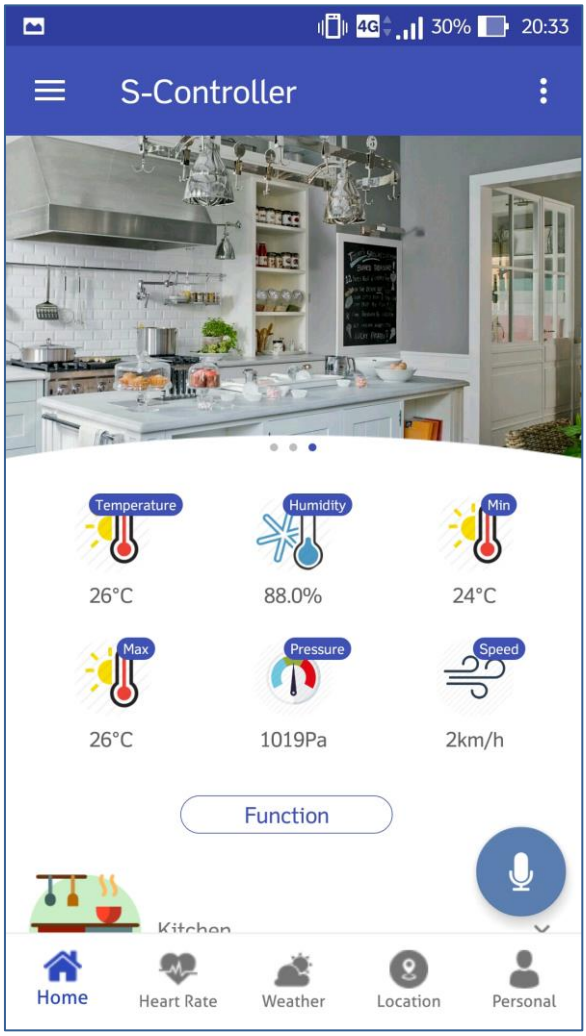


- Figure 37: User interface design – Navigation Bar (Website)

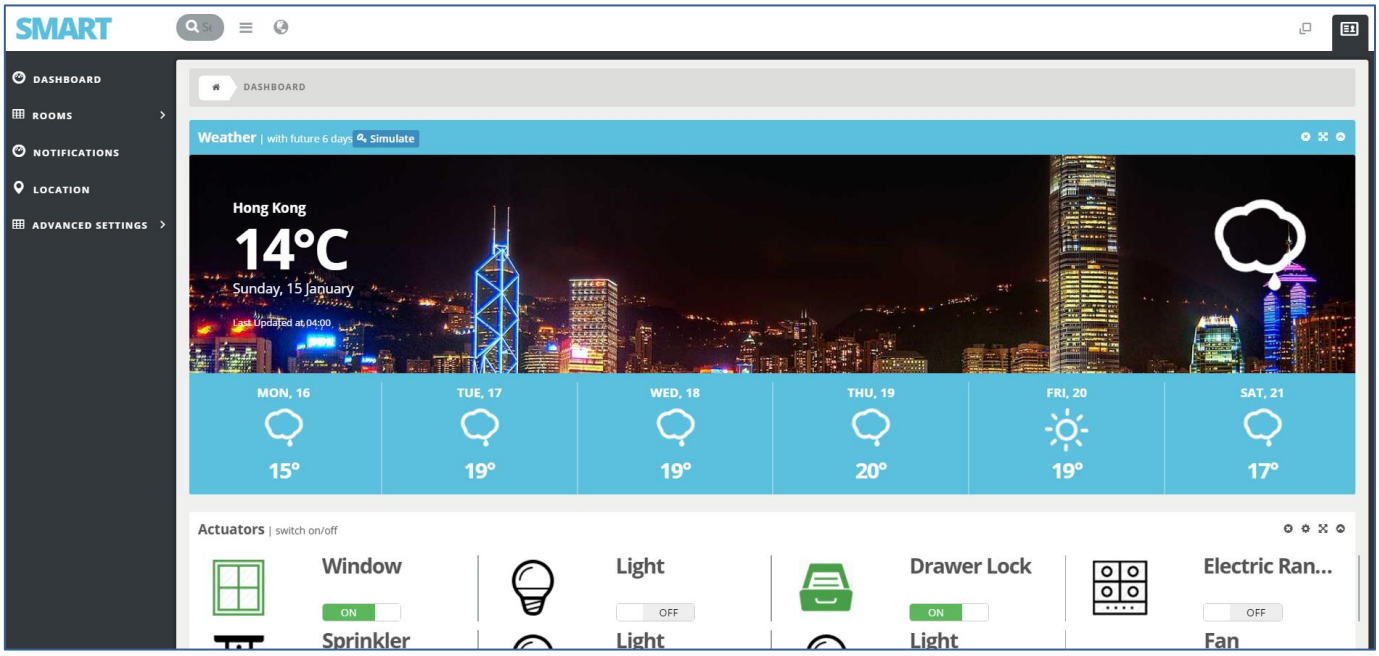


- Figure 38: User interface design – Profile Navigation bar (Website)

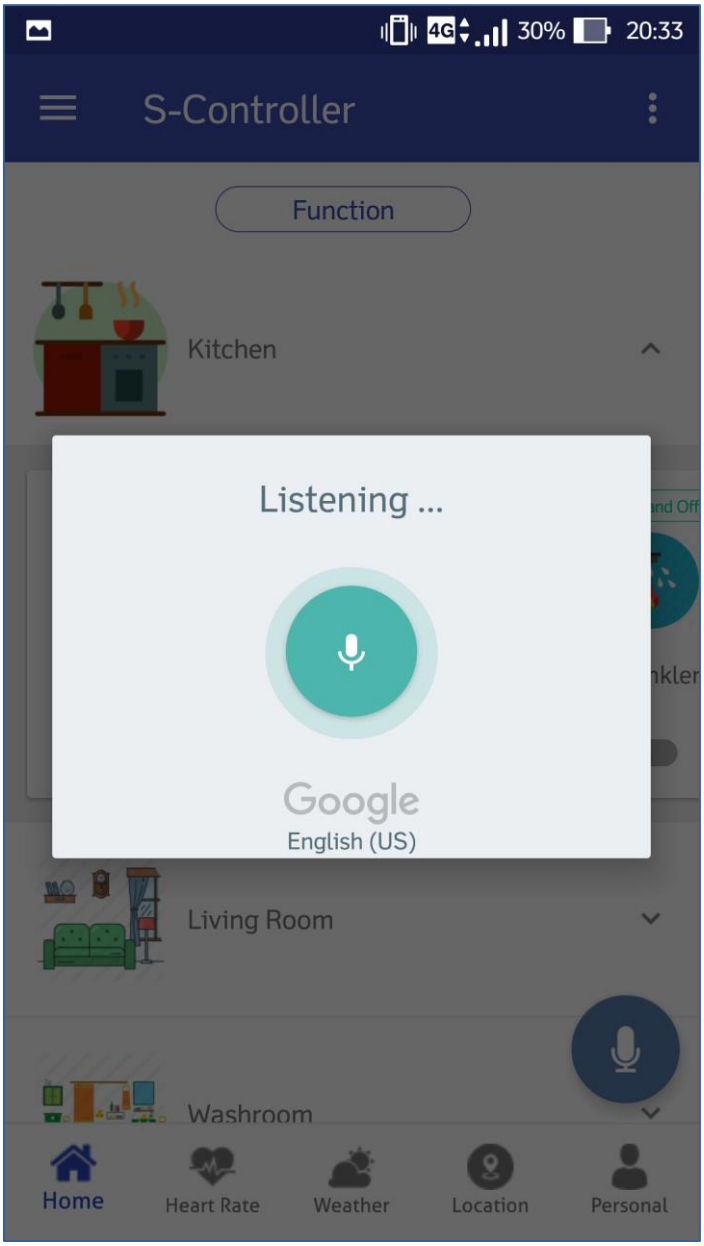
5.3.1.4.3. Main page



- Figure 39: User interface design – Main Page (Android application)

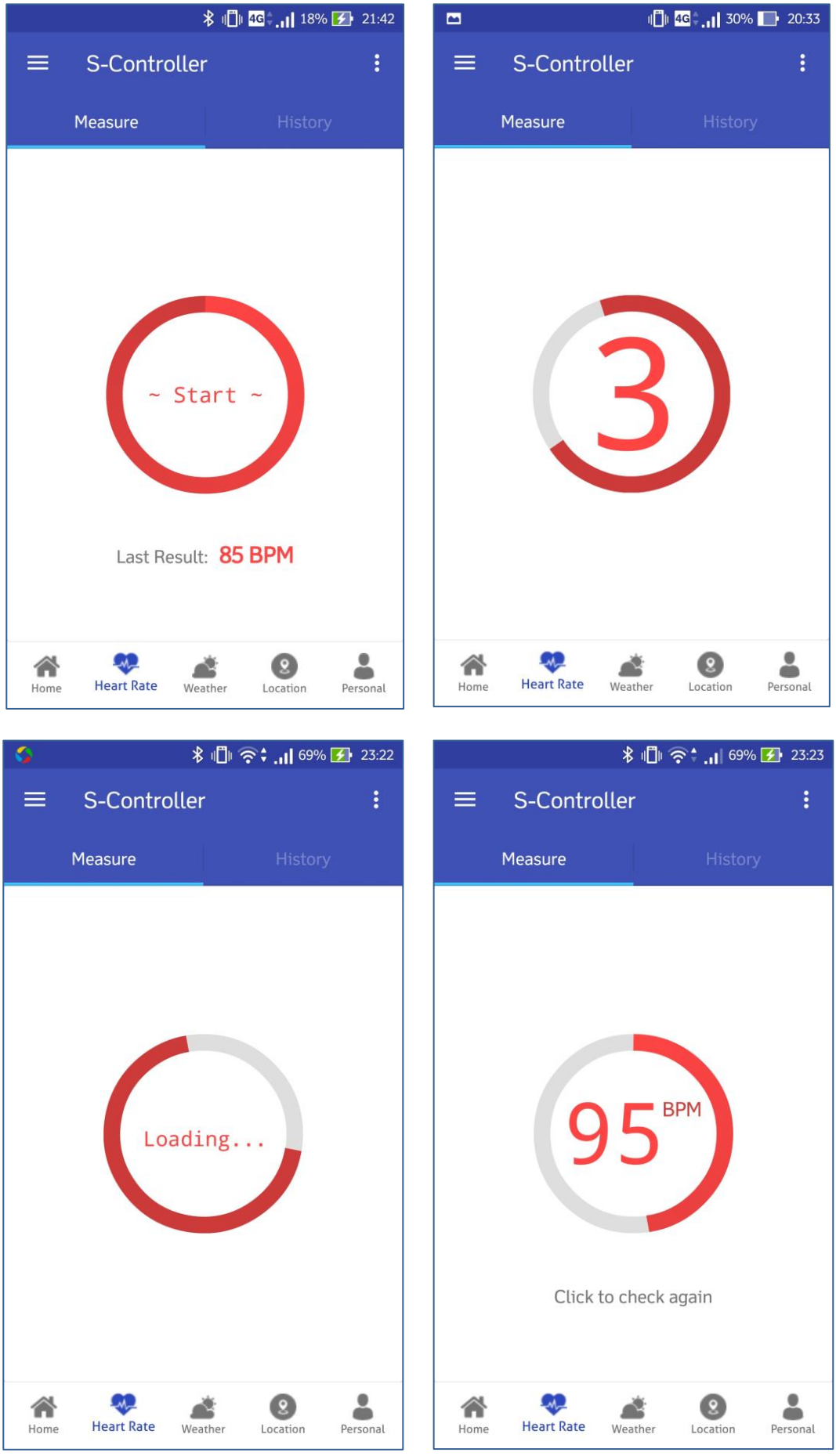


- Figure 40: User interface design – Main Page (Website)



- Figure 41: User interface design – Main Page with sound control actuators (Android application)

5.3.1.4.4. Heart rate page



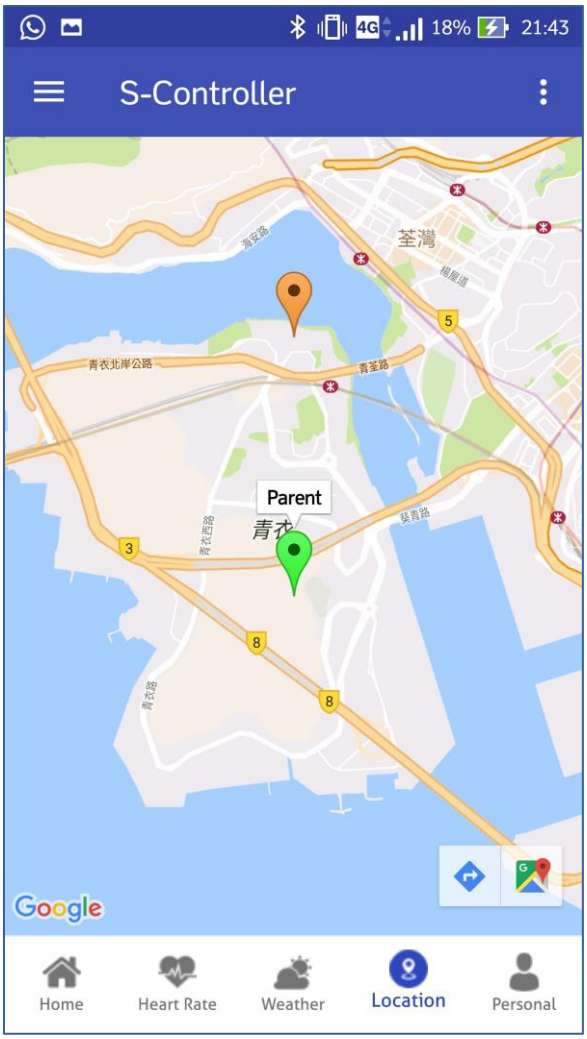
- Figure 42: User interface design – Heart rate page with measure (Android application)

- Figure 43: User interface design – Heart rate page with history (Android application)

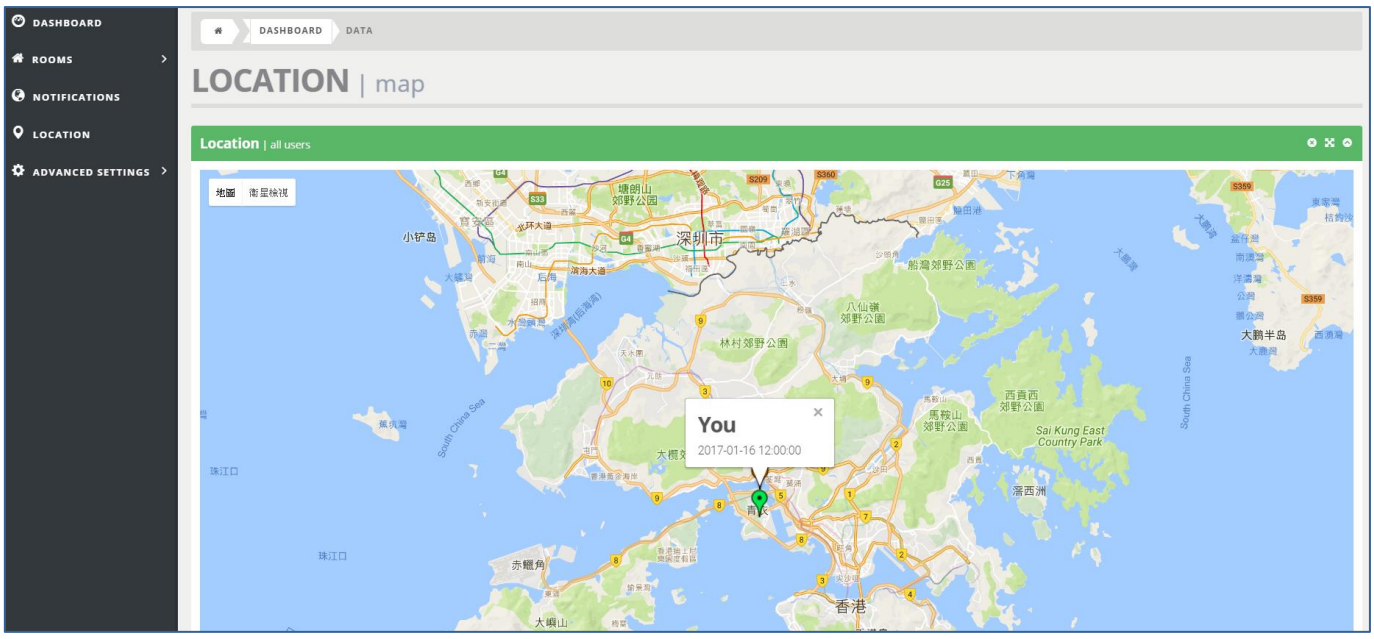
5.3.1.4.5. Weather page

- Figure 44: User interface design – Weather page (Android application)

5.3.1.4.6. Location page

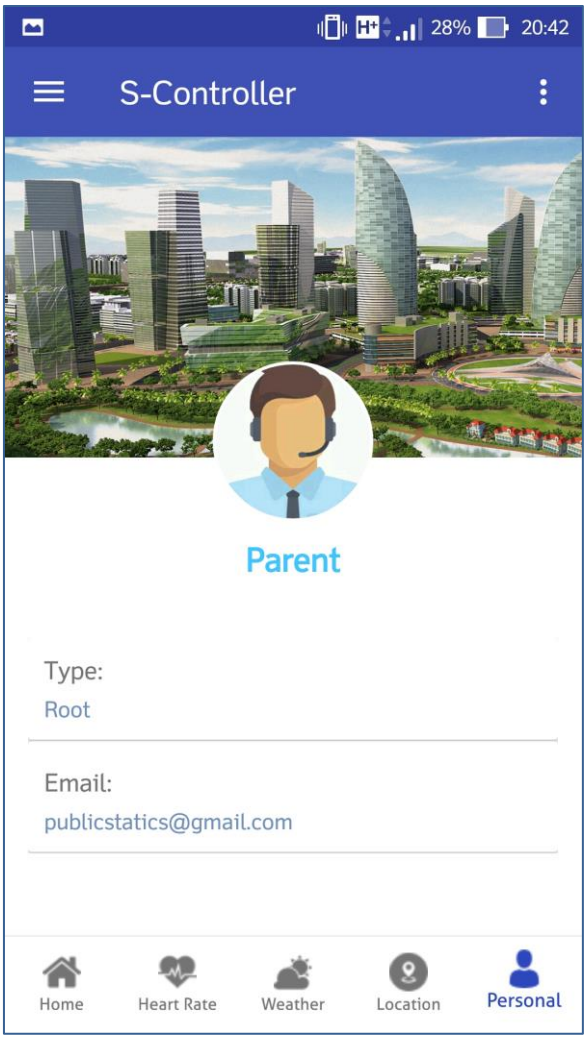


- Figure 45: User interface design – Location page (Android application)

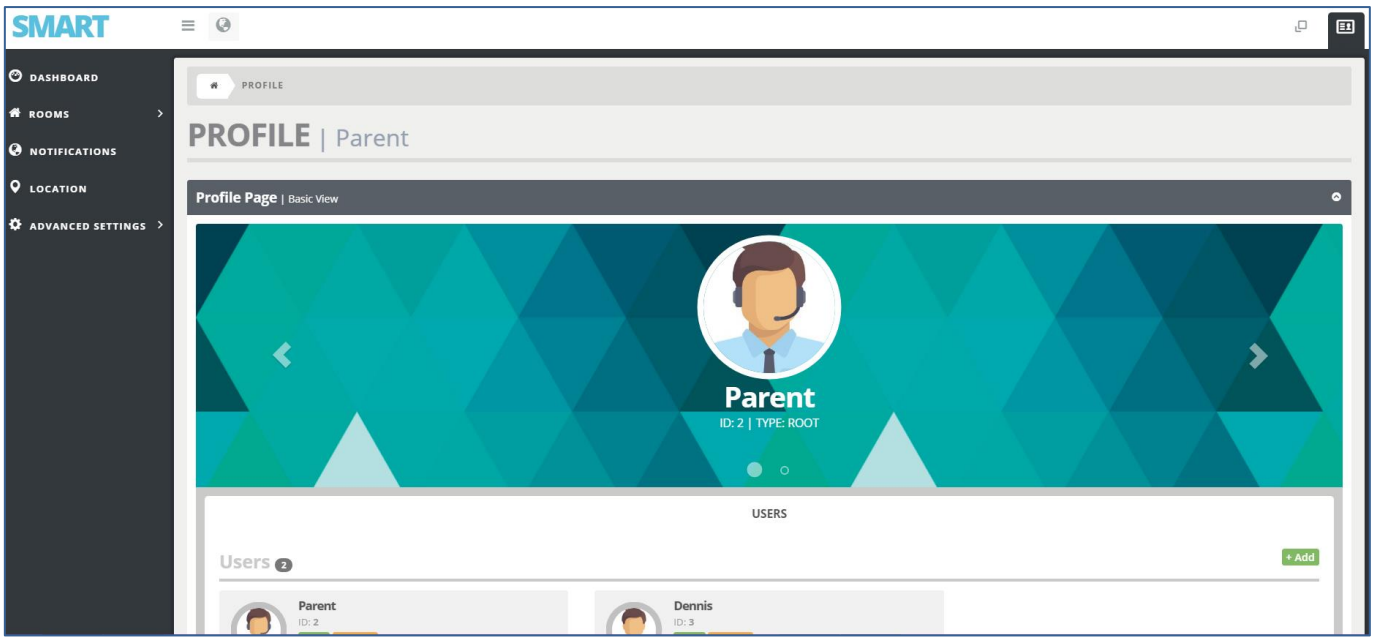


- Figure 46: User interface design – Location page (Website)

5.3.1.4.7. Personal page

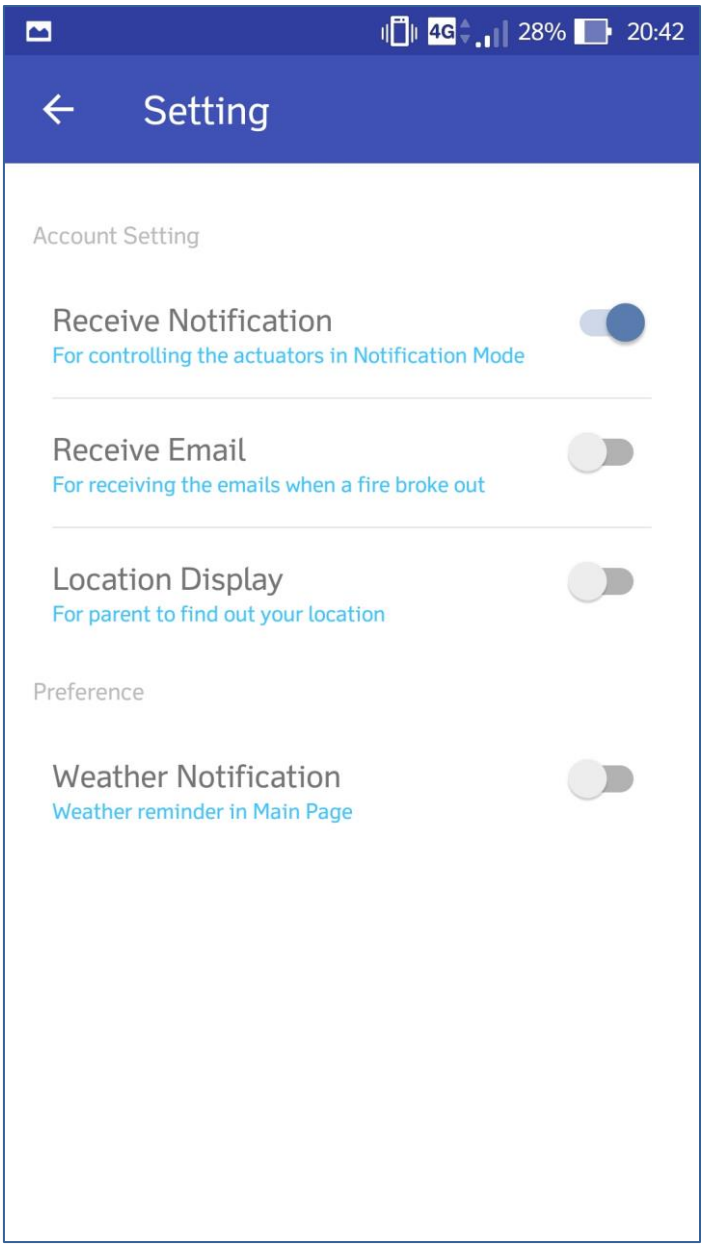


- Figure 47: User interface design – Personal page (Android application)

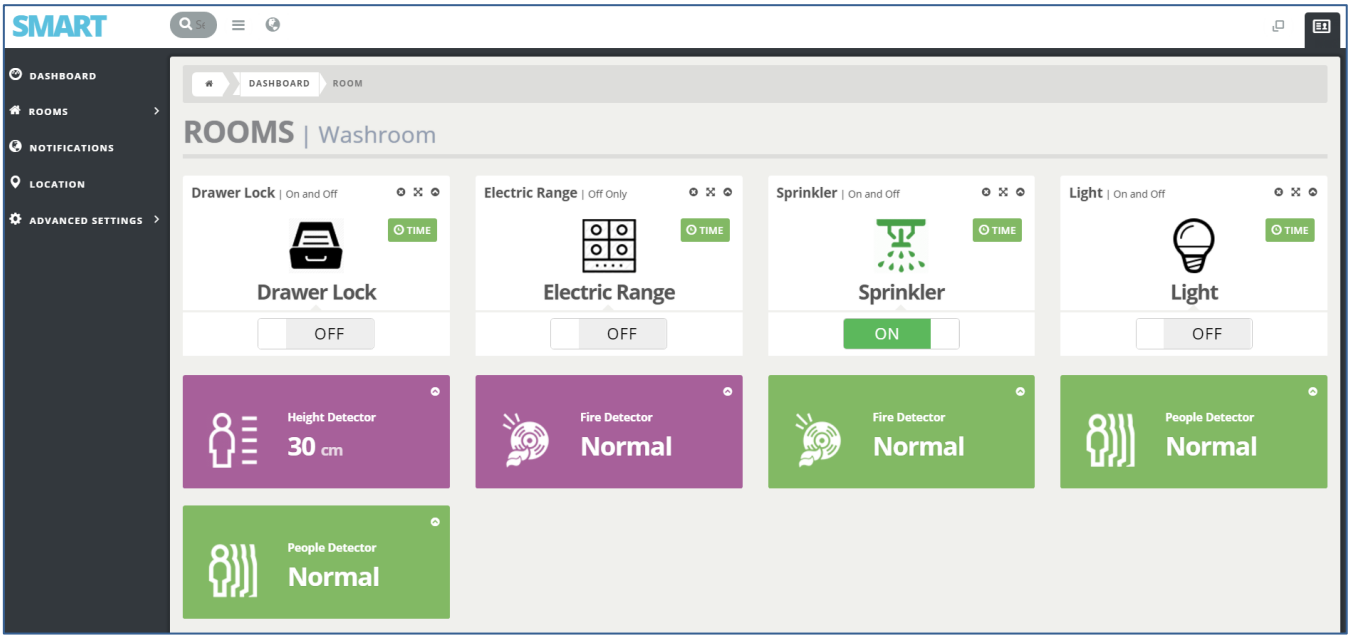


- Figure 48: User interface design – Personal page (Website)

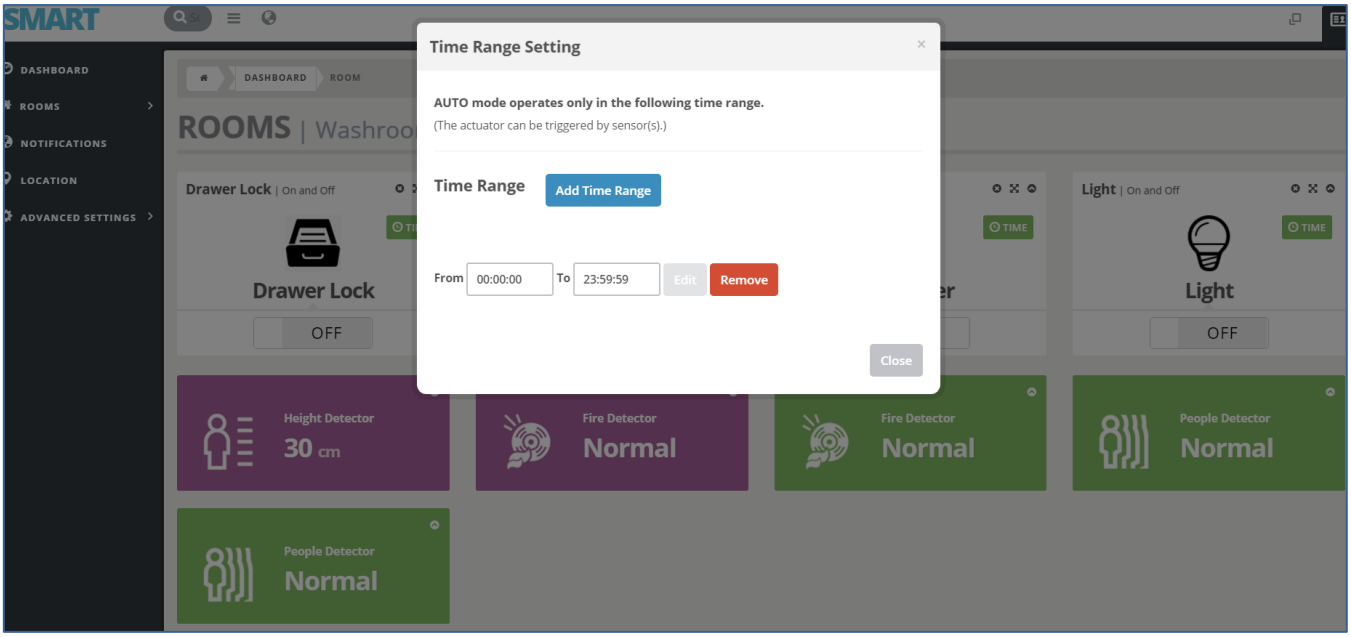
5.3.1.4.8. Setting page



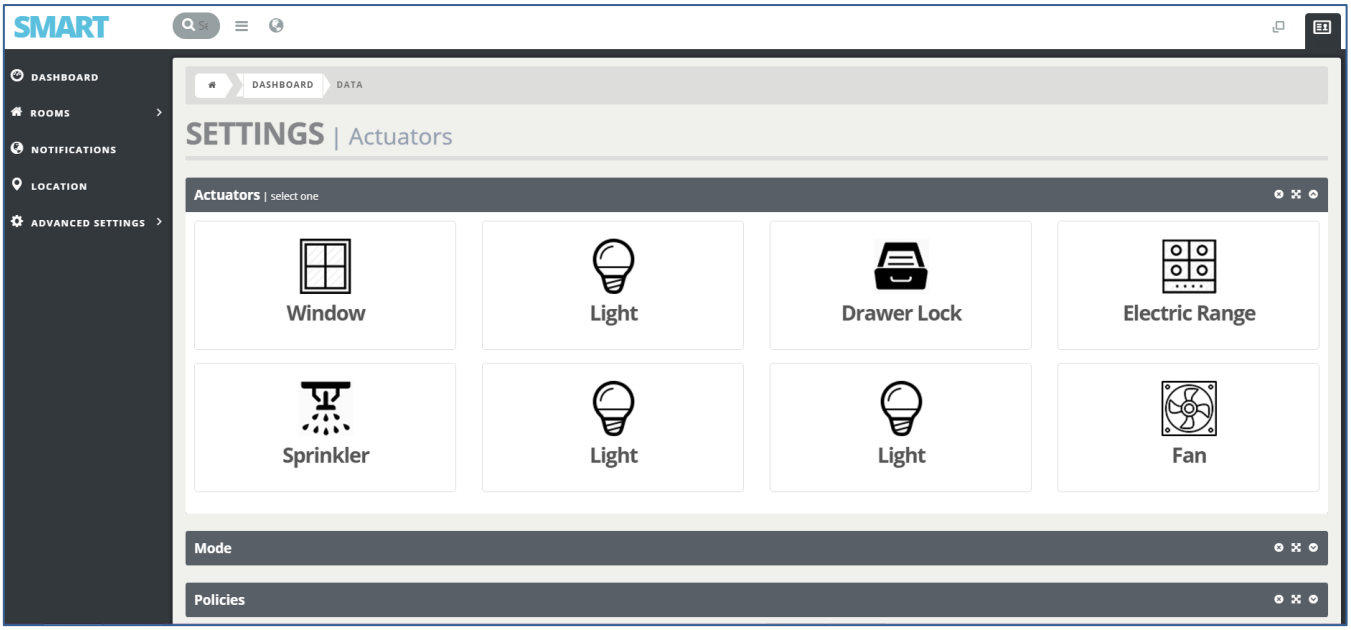
- Figure 49: User interface design – Setting page (Android application)

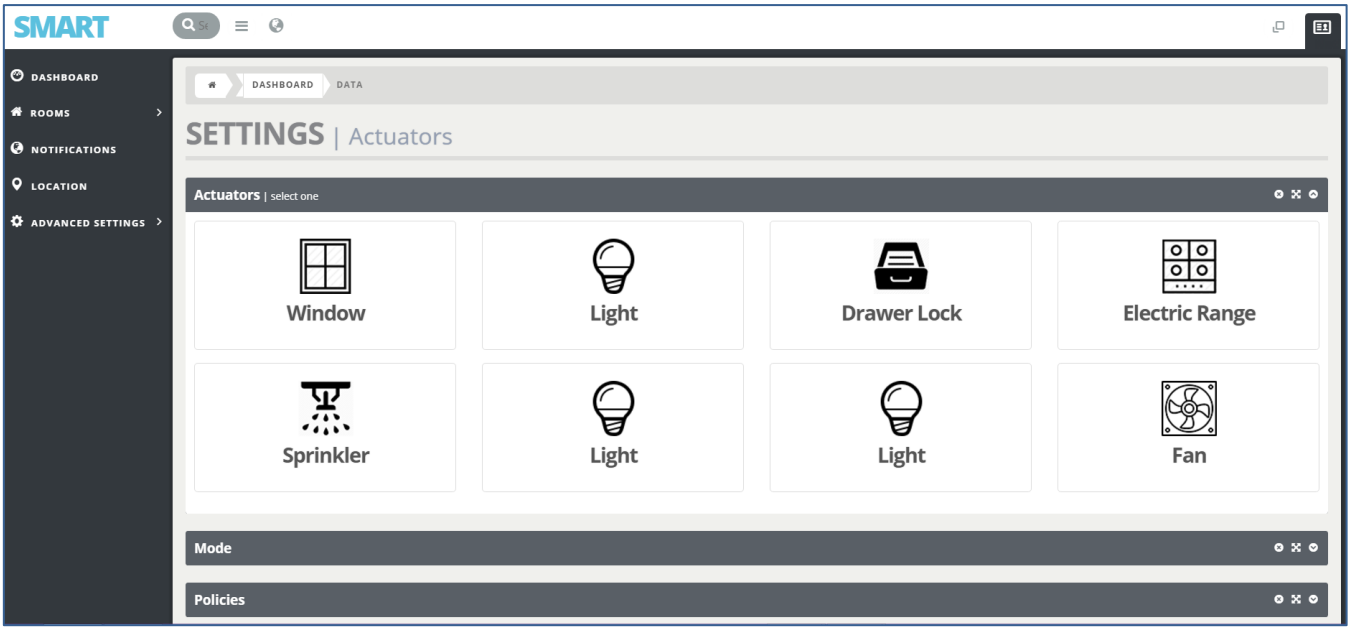


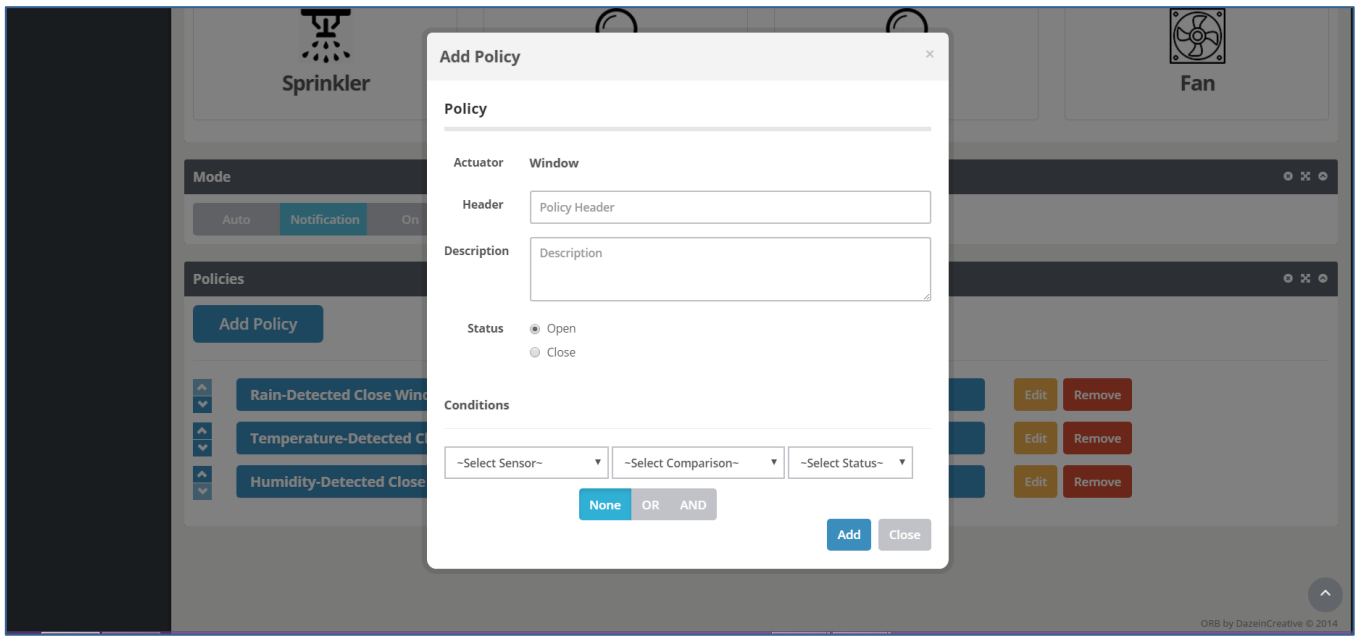
- Figure 50: User interface design – Room settings page (Website)



- Figure 51: User interface design – Room settings page detail information (Website)

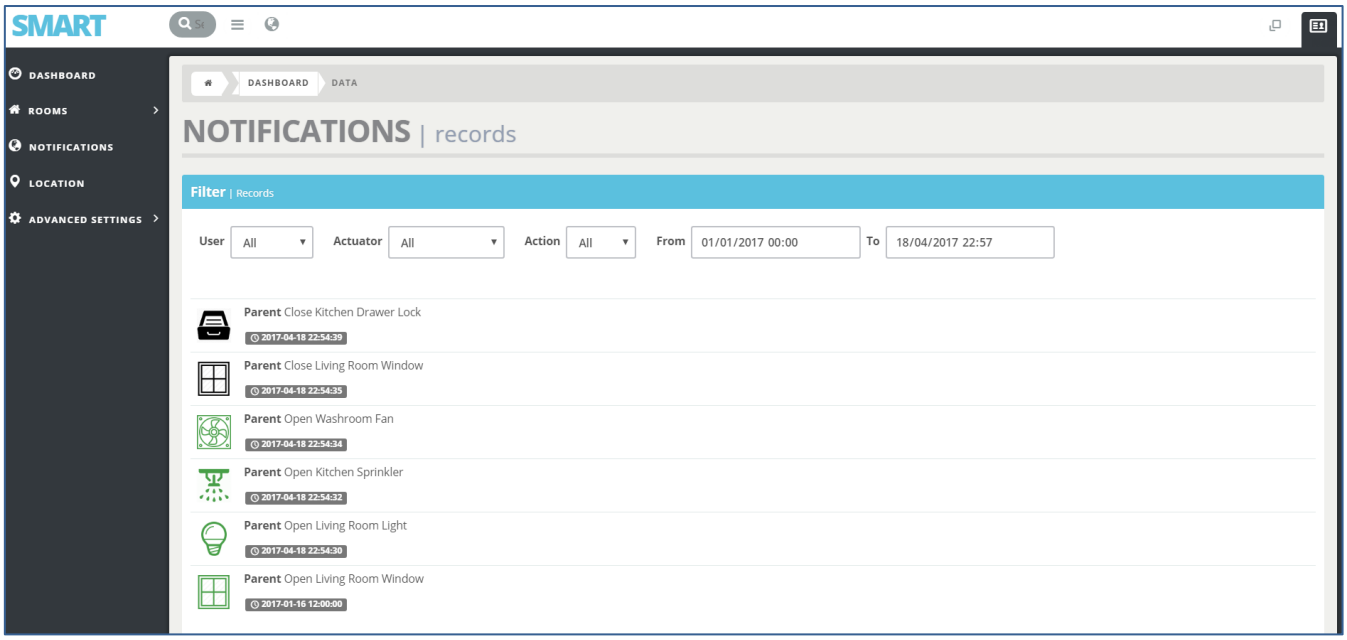




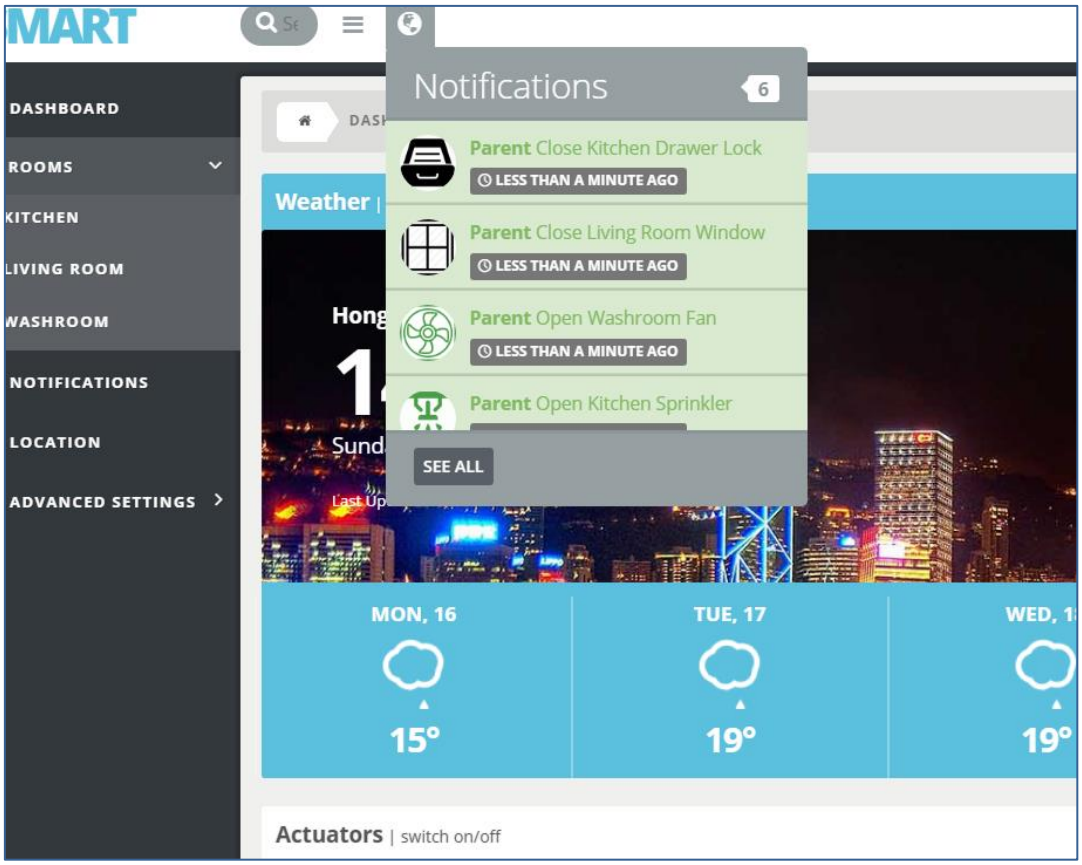


- Figure 52: User interface design – Setting Page (Website)

5.3.1.4.9. Notification page



- Figure 53: User interface design – Notification page (Website)

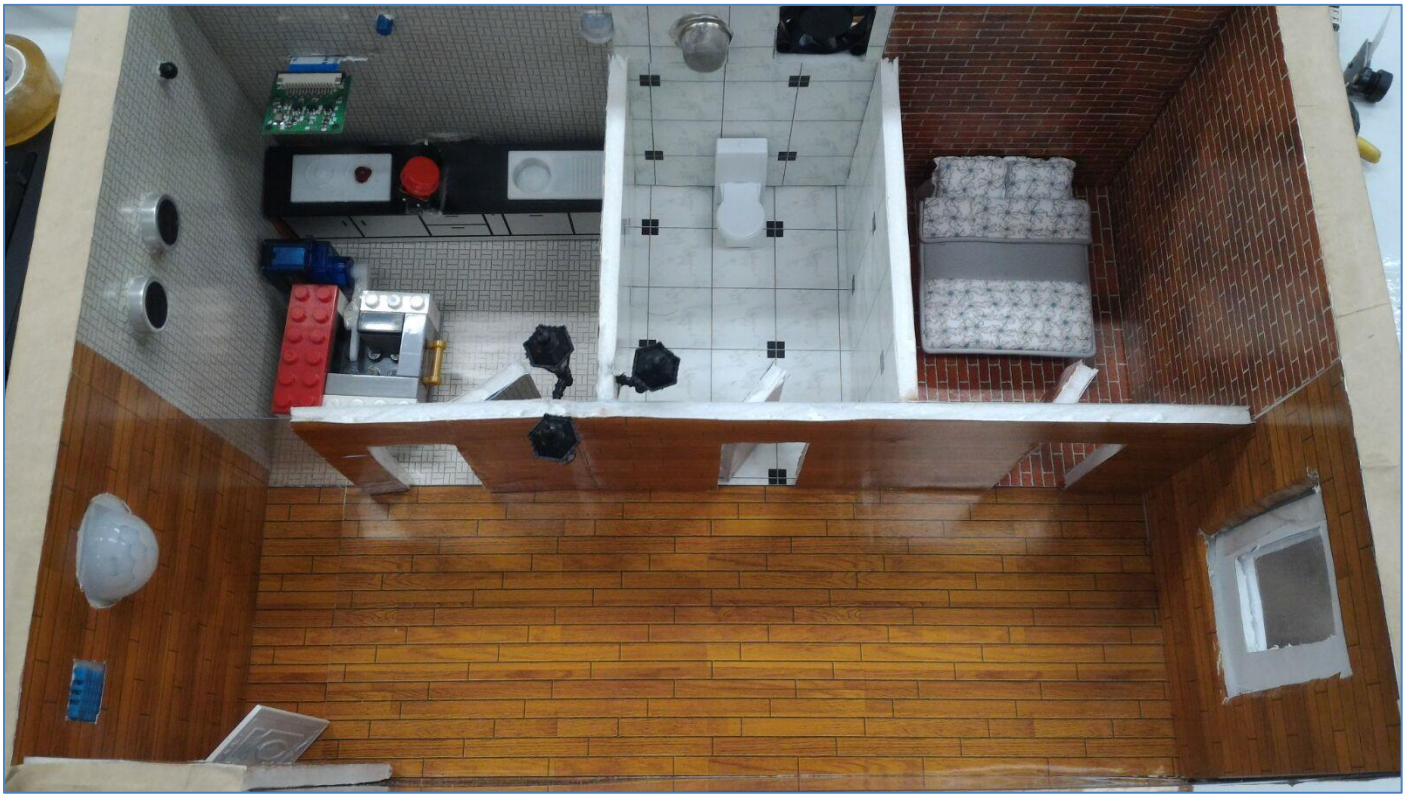


- Figure 54: User interface design – Notification page with detail information (Website)

5.4. System icon

- Figure 55: User interface design – System icon (Android application and Website)

5.5. System model building



- Figure 56: System model building

5.6. Implementation

5.6.1. Test plan

Test No.	Test	Description	Date
Android			
T1	Login	Enter different input	11-3-2017
T2	Control actuators	Control all actuators	11-3-2017
T3	Control actuators by sound	Control actuators to open or close by sound	11-3-2017
T4	Speech weather information	Click weather icon to speech by system	11-3-2017
T5	Measure heart rate and show history	Measure heart rate and show the heart rate records (including history)	11-3-2017
T6	Show weather information in detailed	Show weather information in detailed such as temperature, humidity and related charts	18-3-2017
T7	Show current location [and other users (for root accounts only)]	Show current location [and other users (for root accounts only)]	18-3-2017
T8	Show profile information	Show profile information such as name, account type and email	18-3-2017
T9	Change settings	Change users' preferences	18-3-2017
Test No.	**Test**	**Description**	**Date**
Web			
T10	Login	Enter different input	11-3-2017
T11	Control actuators	Control all actuators	11-3-2017
T12	Manage actuators	Manage actuators' settings such as time range	25-3-2017
T13	Push and receive notification	Push and receive notification for each actuator in Notification mode	25-3-2017
T14	Manage notification	Manage notification (filter)	25-3-2017
T15	Show current location [and other users (for root accounts only)]	Show current location [and other users (for root accounts only)]	25-3-2017
T16	Show profile information	Show profile information such as name, account id and other members	1-4-2017
T17	Manage users (for root accounts only)	Manage users such as adding, editing and deleting user	1-4-2017
T18	Show advanced settings	Show advanced settings such as mode and policy for each actuator	8-4-2017
T19	Change actuators' mode	Change actuators' mode	8-4-2017
T20	Manage policy settings	Manage policy settings such as adding, editing and deleting policy	15-4-2017

- Table 60: Implementation – Test plan

5.6.2. Test results

Test Input	Expect Result	Actual Result
Android		
T1: Login		
Input correct account password (✓Successful Case)	Login successfully and go to Main Page	✓Meet the exception
Input wrong account (✗Wrong Case)	Ask user for re-entering account and password	✓Meet the exception
T2: Control actuators		
Switch on / off all the actuators (✓Successful Case)	Perform related actions	✓Meet the exception
T3: Control actuators by sound		
Say "Turn on the window" (✓Successful Case)	Open the window and notification about update successfully	✓Meet the exception
Say "Open the window" (✓Successful Case)	Open the window and notification about update successfully	✓Meet the exception
Say "Turn off the light" (✗Wrong Case)	Notification about the system cannot find which light the user want to close	✓Meet the exception
Say "Turn off the kitchen light" (✓Successful Case)	Close the light in the kitchen and notification about update successfully	✓Meet the exception
Say "Kitchen light on" (✓Successful Case)	Open the light in the kitchen and notification about update successfully	✓Meet the exception
Say "Kitchen light" (✗Wrong Case)	Notification about the system cannot know user's action	✓Meet the exception
Say "Turn on the window", but the window has been opened (✗Wrong Case)	Notification about update failed	✓Meet the exception
T4: Speech weather information		
Click the temperature icon (✓Successful Case)	Speech about the temperature sentence	✓Meet the exception
Click the humidity icon (✓Successful Case)	Speech about the humidity sentence	✓Meet the exception
Click the maximum temperature icon (✓Successful Case)	Speech about the maximum temperature sentence	✓Meet the exception
Click the minimum temperature icon (✓Successful Case)	Speech about the minimum temperature sentence	✓Meet the exception
Click the pressure icon (✓Successful Case)	Speech about the pressure sentence	✓Meet the exception
Click the speed icon (✓Successful Case)	Speech about the wind speed sentence	✓Meet the exception
T5: Measure heart rate and show history		
Click "Start" (✓Successful Case)	Show detected heart rate and update the charts	✓Meet the exception
T6: Show weather information in detailed		
Click the weather page (✓Successful Case)	Show latest 7-days weather records	✓Meet the exception

T7: Show current location [and other users (for root accounts only)]		
Click location page with GPS (✓Successful Case)	Show current location	✓Meet the exception
Click location page without GPS (✓Successful Case)	Show the last location	✓Meet the exception
Click location page with GPS and is a root account (✓Successful Case)	Show current location and other users' location	✓Meet the exception
T8: Show profile information		
Click the personal page (✓Successful Case)	Show personal information	✓Meet the exception
T9: Change settings		
Switch on "Receive Notification" (✓Successful Case)	Update database and can receive the notification if changed actuator is in Notification Mode	✓Meet the exception
Switch on "Receive Email" (✓Successful Case)	Update database and can receive the email if a fire was detected	✓Meet the exception
Switch on "Location Display" (✓Successful Case)	Update database and the user is showed in the location page for root user to monitor	✓Meet the exception
Switch on "Weather Notification" (✓Successful Case)	Update database and the system will speech the weather information after the user logins	✓Meet the exception
Switch off "Weather Notification" (✓Successful Case)	Update database and the system will not speech the weather information after the user logins	✓Meet the exception

- Table 61: Implementation – Test results (Mobile application)

Test Input	Expect Result	Actual Result
Web		
T10: Login		
Input correct account password (✓Successful Case)	Login successfully and go to Dashboard Page	✓Meet the exception
Input wrong account (✗ Wrong Case)	Ask user for re-entering account and password	✓Meet the exception
T11: Control actuators		
Switch on / off all the actuators (✓Successful Case)	Perform related actions	✓Meet the exception
T12: Manage actuators		
Add new time range and set time from 18:00 to 20:00 for window actuator (✓Successful Case)	The window performs any actions during that time only	✓Meet the exception
Remove all time range for the light in the living room (✓Successful Case)	The light in the living room will close, even if the sensor detects movement.	✓Meet the exception
T13: Push and receive notification		
A sensor is activated and the linked actuator is in Notification mode (✓Successful Case)	The actuator will not perform any actions and the user will receive a notification for choosing action	✓Meet the exception
A sensor is activated and the linked actuator is in Auto mode (✓Successful Case)	The actuator will perform the related action automatically	✓Meet the exception
T14: Manage notification		
Click notification page (✓Successful Case)	Show latest 10 records	✓Meet the exception
Scroll the record to the bottom (✓Successful Case)	Load more 10 records	✓Meet the exception
T15: Show current location [and other users (for root accounts only)]		
Click location page with GPS permission (✓Successful Case)	Show current location	✓Meet the exception
Click location page without GPS permission (✓Successful Case)	Show the last location	✓Meet the exception
Click location page with GPS permission and is a root account (✓Successful Case)	Show current location and other users' location	✓Meet the exception
T16: Show profile information		
Click the "Profile" in the navigation (✓Successful Case)	Show personal information	✓Meet the exception
T17: Manage users (for root accounts only)		
Add a new user, enter related data and then click the "Add" button (✓Successful Case)	Show success message	✓Meet the exception

T18: Show advanced settings		
Click advanced settings page (✓Successful Case)	Show all actuators and their current mode and policies	✓Meet the exception
T19: Change actuators' mode		
Change the mode for window from Notification to Auto (✓Successful Case)	Update database	✓Meet the exception
T20: Manage policy settings		
Add a policy and the status is "close" for window when temperature is less than 15C (✓Successful Case)	The window will be closed automatically in Auto mode if the temperature is less than 15C	✓Meet the exception
Add a policy and the status is "close" for window when temperature is less than 15C or greater than 30C (✓Successful Case)	The window will be closed automatically in Auto mode if the temperature is less than 15C or greater than 30C	✓Meet the exception
Edit the above policy for changing the 15C to 20C (✓Successful Case)	The window will be closed automatically in Auto mode if the temperature is less than 20C or greater than 30C	✓Meet the exception

- Table 62: Implementation – Test results (Website)

5.7. Conclusions

At the end of the final stage, all of the system development and the system implementation in the project are already completed with scheduled. And the final report is illustrates all the details and information of the system development. Also, the system design has using different system diagrams to describe of the system architecture.

In the section in this report with system development, we have provided the problem findings and solution, main function and the direction of our digital home system. We also planning with different system architecture by using different system diagram, such as entity-relation diagram, use case diagram, and the sequence diagrams to specify the whole structure of our application. Those diagrams can provide a clear relationship between each user and each function effectively.

Furthermore, the data dictionary introduce about all the data in our database will store, and the design of our database a shown too. Then the system architecture talks about the devices, which we will use to run our application.

With processing the system development, we have considered with the procedural design and the user interface design. We hope that the digital home system with two platform of Android application and system website can be more humanity and user-friendly.

For the last of the report, we have provided the planning schedule, user guide, test script, and some of the program listings after the conclusion. And the section of the log sheet will explain the record of the meeting with our supervisor and supplier. Also, the user guide is a document which teaches the new user to use our application.

6. References

6.1. Source materials

1. Hong Kong Housing Society, “家居陷阱逐個捉”. Retrieved 4th July 2015. From PDF report: http://www.hkhselderly.com/tc/house/designing/74

2. Department of Health, “長者跌倒”. Retrieved November 2013. From PDF report: http://www.chp.gov.hk/files/pdf/ncd_watch_nov2013_chin.pdf

3. The Hong Kong Paediatric Foundation, “香港十大兒童家居意外高危陷阱調查報告”. From PDF report: https://www.hkupop.hku.hk/english/report/childSafety09/content/resources/ChildSafetySurvey.pdf

4. The Nethersole School of Nursing, “居家安老「零意外」”. From PDF report: http://www.cadenza.hk/training/pdf/ps/CTP005_cur4_ps1.pdf

5. TrossenRobotics.com, “Arduino Sensors”. From website: http://www.trossenrobotics.com/c/arduino-sensors.aspx

6. METAS, “METAS – Think outside the box”. From website: http://www.funmetas.com.hk/

7. HKT, “Smart Living”. From website: http://smartliving.hkt.com/eng/

8. Somfy, “Home automation: Somfy, connected house specialist”. From website: http://www.somfy.com.hk/cn-hk/products/home-automation/tahoma-box

9. CityGrow, “CityGrow E-Systems”. From website: http://citygrowsys.com/

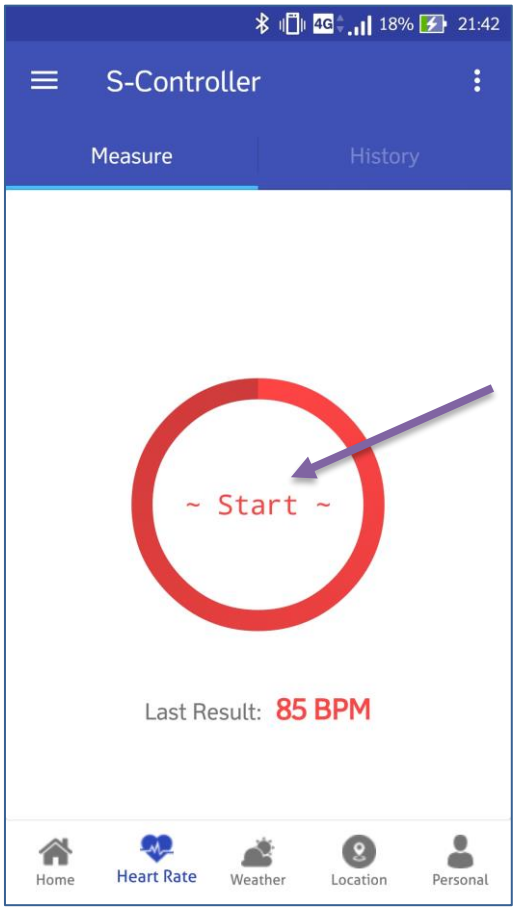
10. Smart Technology, “Home Automation”, From website: http://home.hit.no/~hansha/documents/home_automation/home_automation.htm

6.2. Appendices

6.2.1. User guide and installation guide

6.2.1.1. User guide (Android Application)

1. Measure heart rate



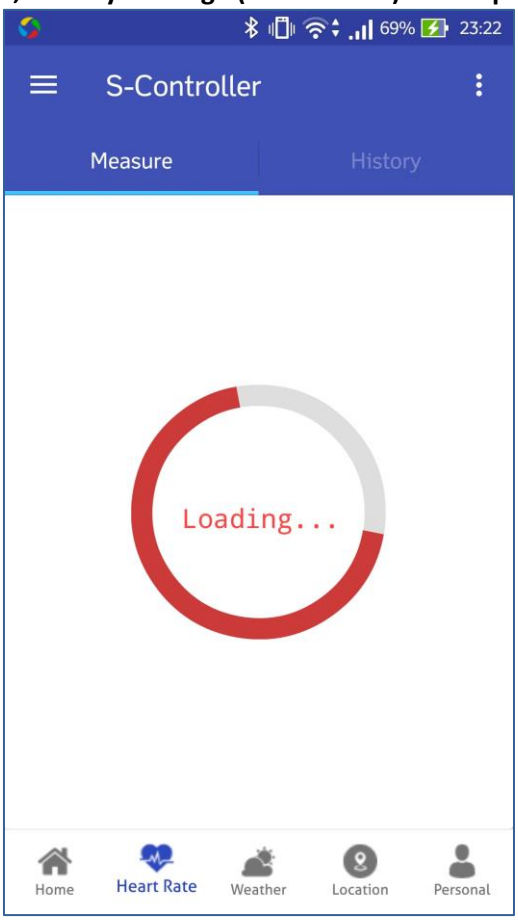
- Figure 57: User guide – Measure heart rate (Android application)

⇨ **Step 1**: **After clicking the "Heart Rate", you will go to that page.**

⇨ **Step 2:** **Put a finger to the heart rate detector and click the "Start" for measuring heart rate.**

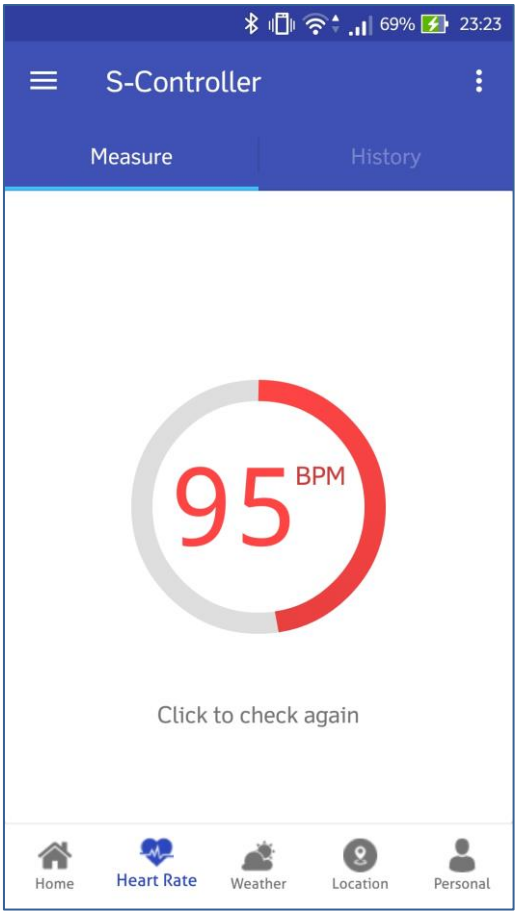
- Figure 58: User guide– Measure heart rate with count down (Android application)

⇨ **Step 3: After clicking the "Start", a ready message (3 ... 2 ... 1 ...) will display.**



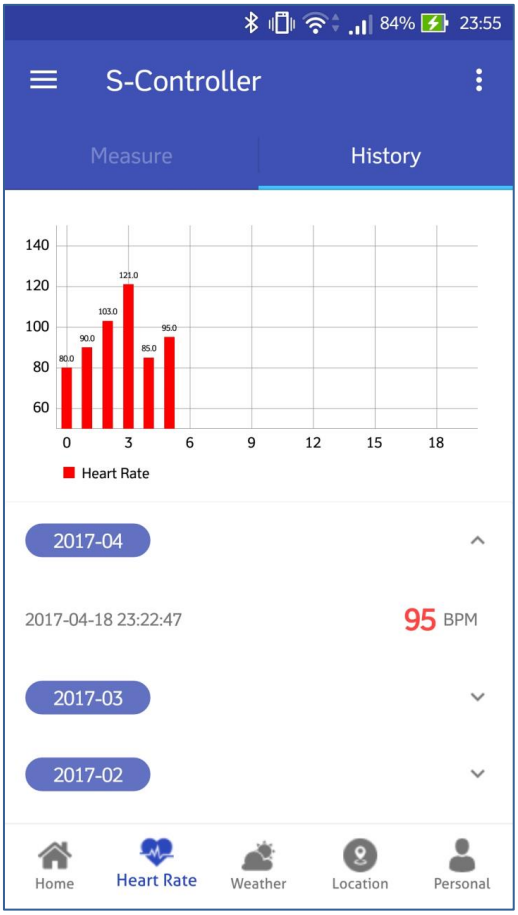
- Figure 59: User guide – Measure heart rate with loading (Android application)

- **Step 4: You will see the "Loading . . ." when the sensor is detecting.**



- Figure 60: User guide – Measure heart rate with result (Android application)

⇨ **Step 5: If success, you will see your heart rate and the record will be stored in the database automatically for reference and comparison.**

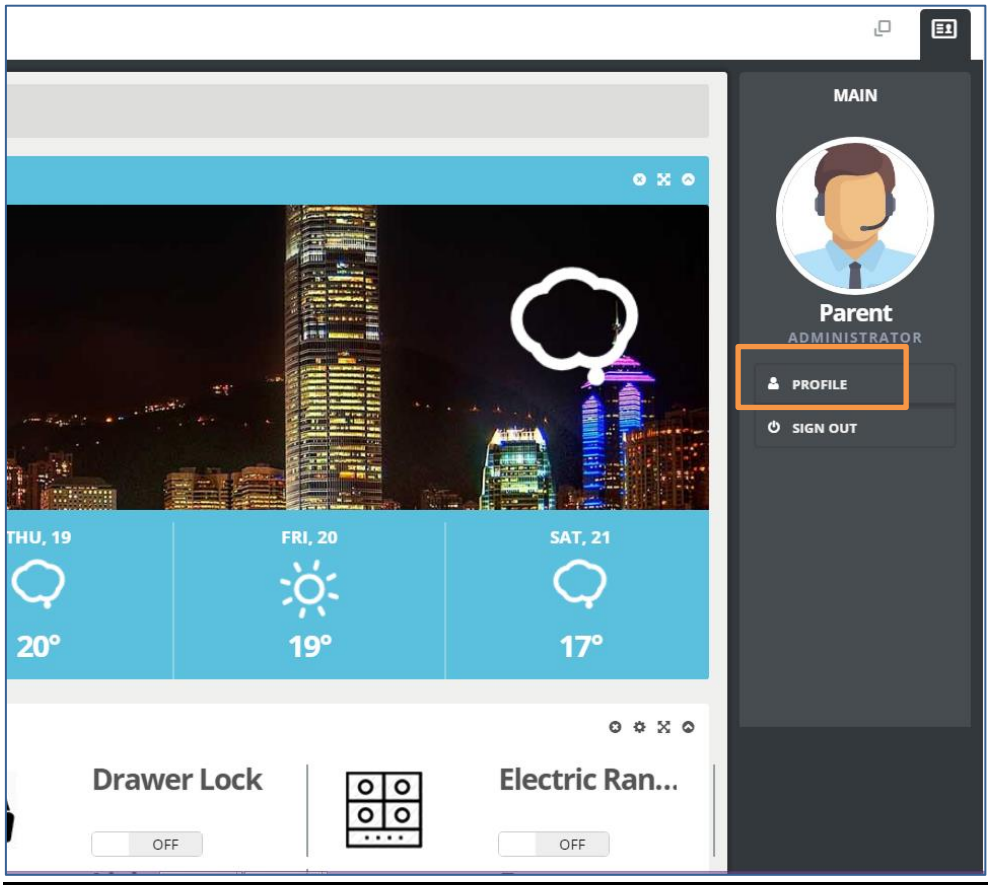


- Figure 61: User guide – Measure heart rate with history (Android application)

⇨ **Step 6: You can also select the "History" tab to compare your records (Show the latest 20 records only).**

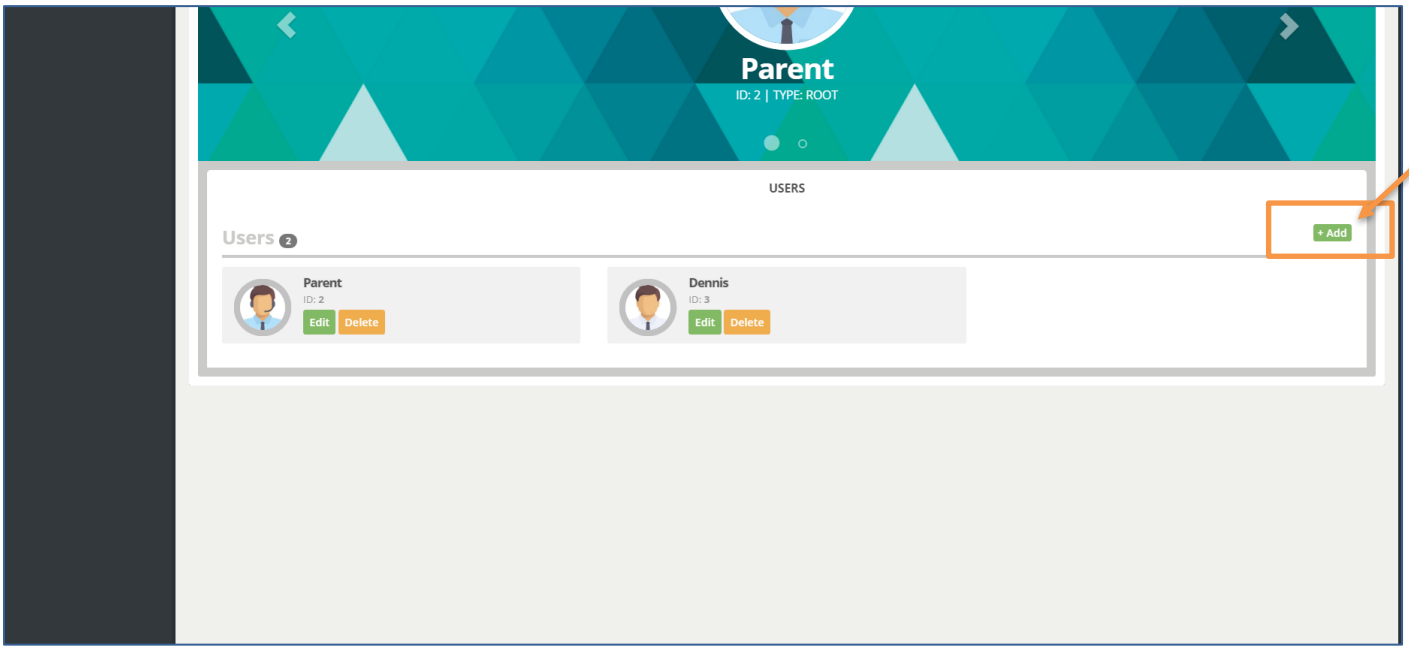
6.2.1.2. User guide (Website)

1. Add user (For root user account only)



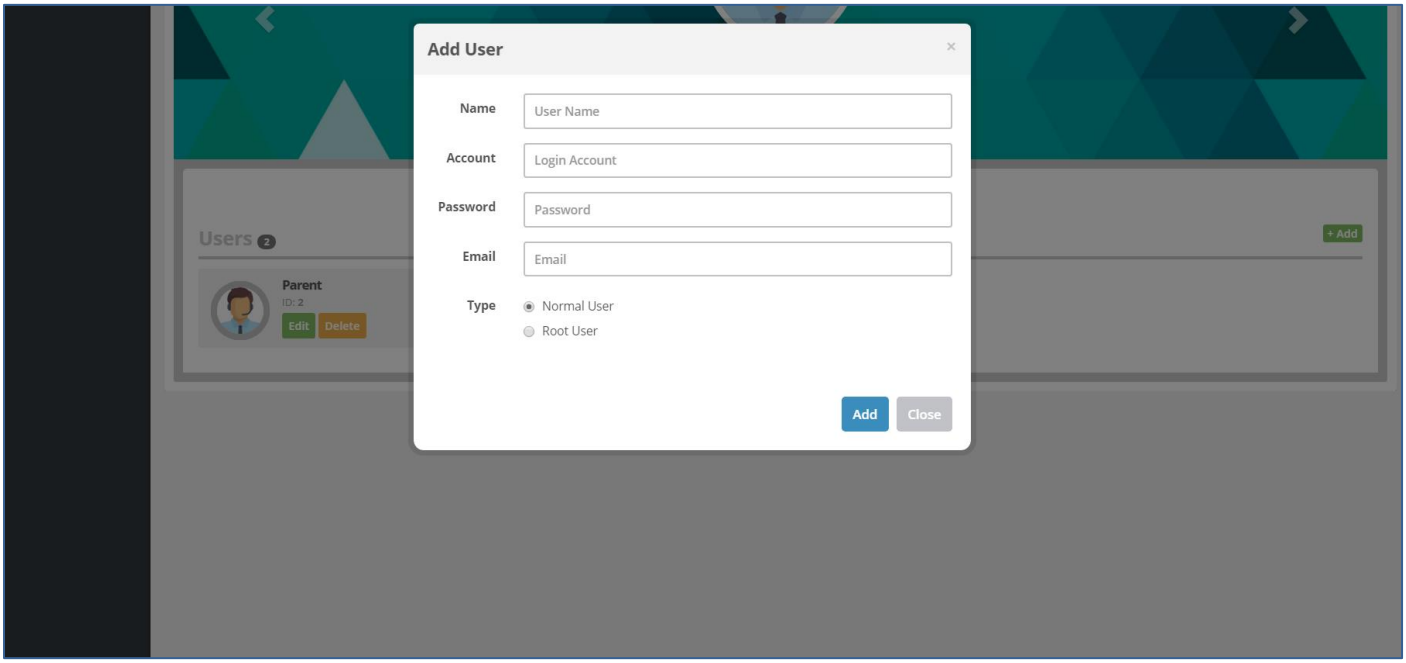
- Figure 62: User guide – Add user (Website)

- **Step 1: In the right-hand side, click the "Profile" button.**



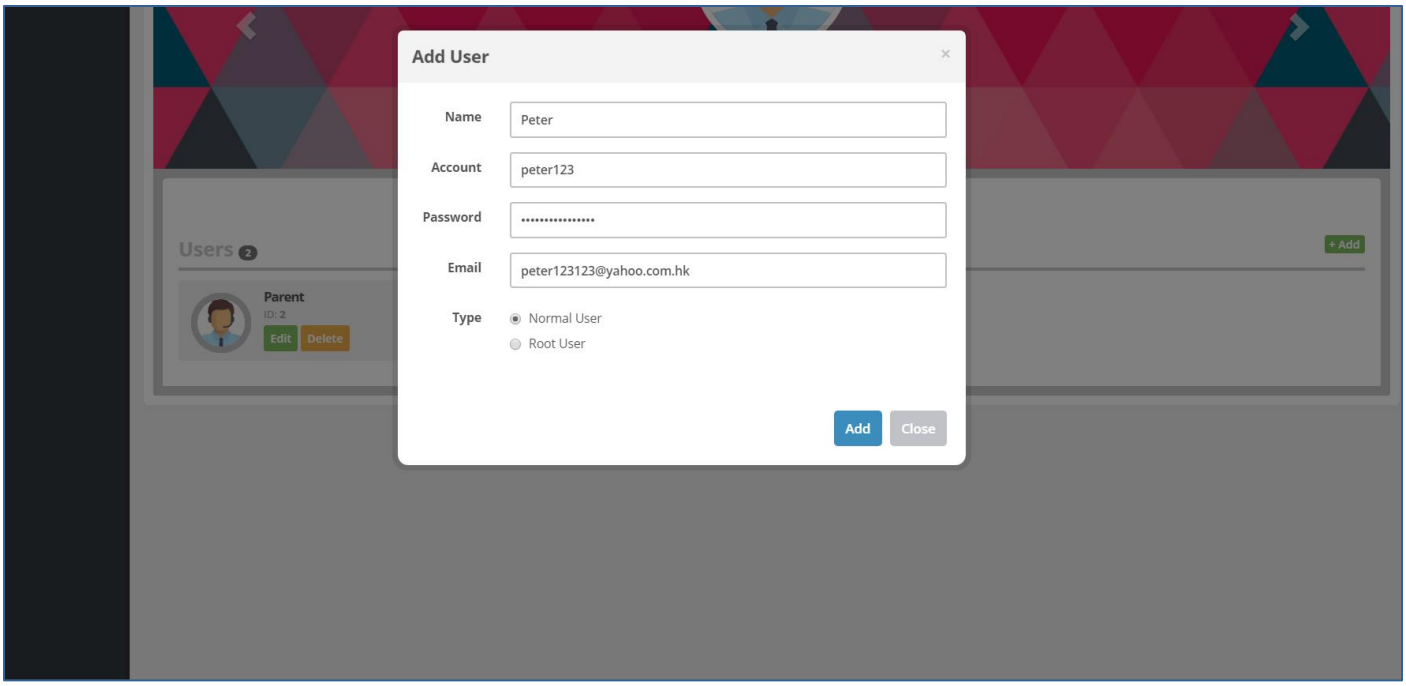
- Figure 63: User guide – Add user with step 2 (Website)

⇨ **Step 2: Click the "Add" button and a registration form will display.**



- Figure 64: User guide – Add user with step 3 (Website)

⇨ **Step 3: Enter all the user information and select the type of the account.**

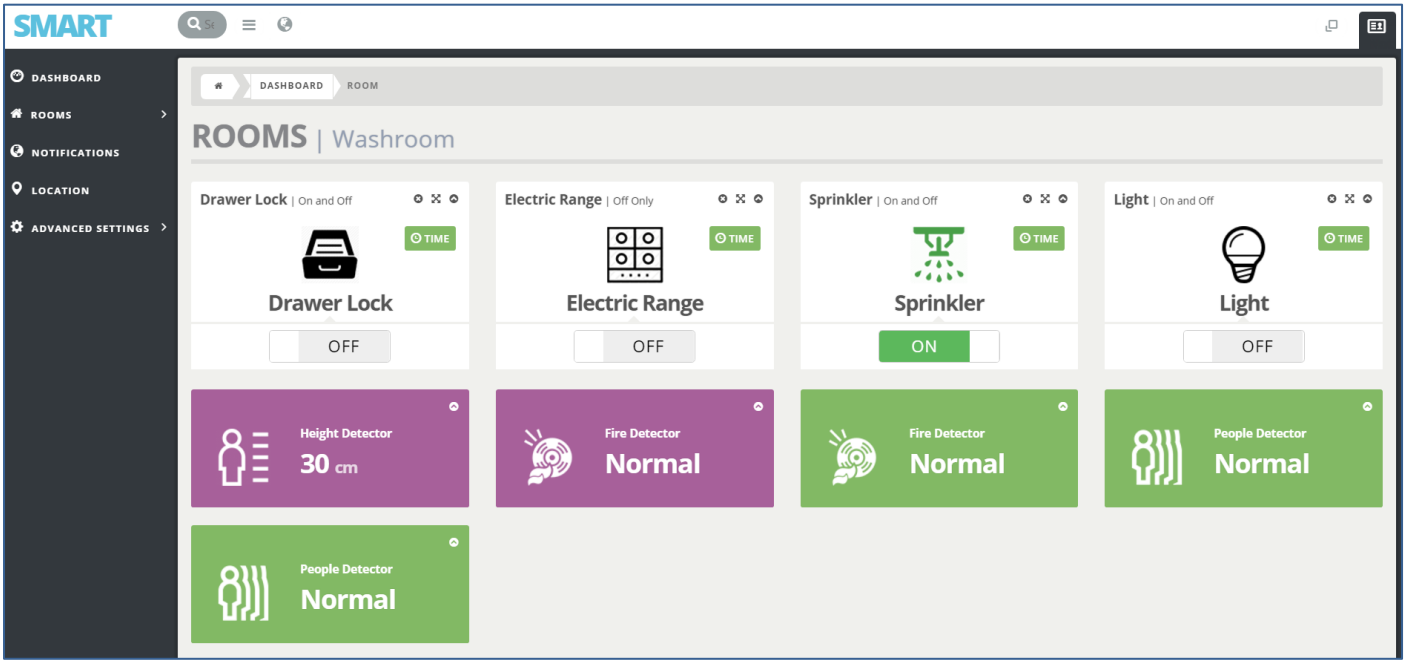


- Figure 65: User guide – Add user with step 4 (Website)

⇨ **Step 4: Click "Add" button.**

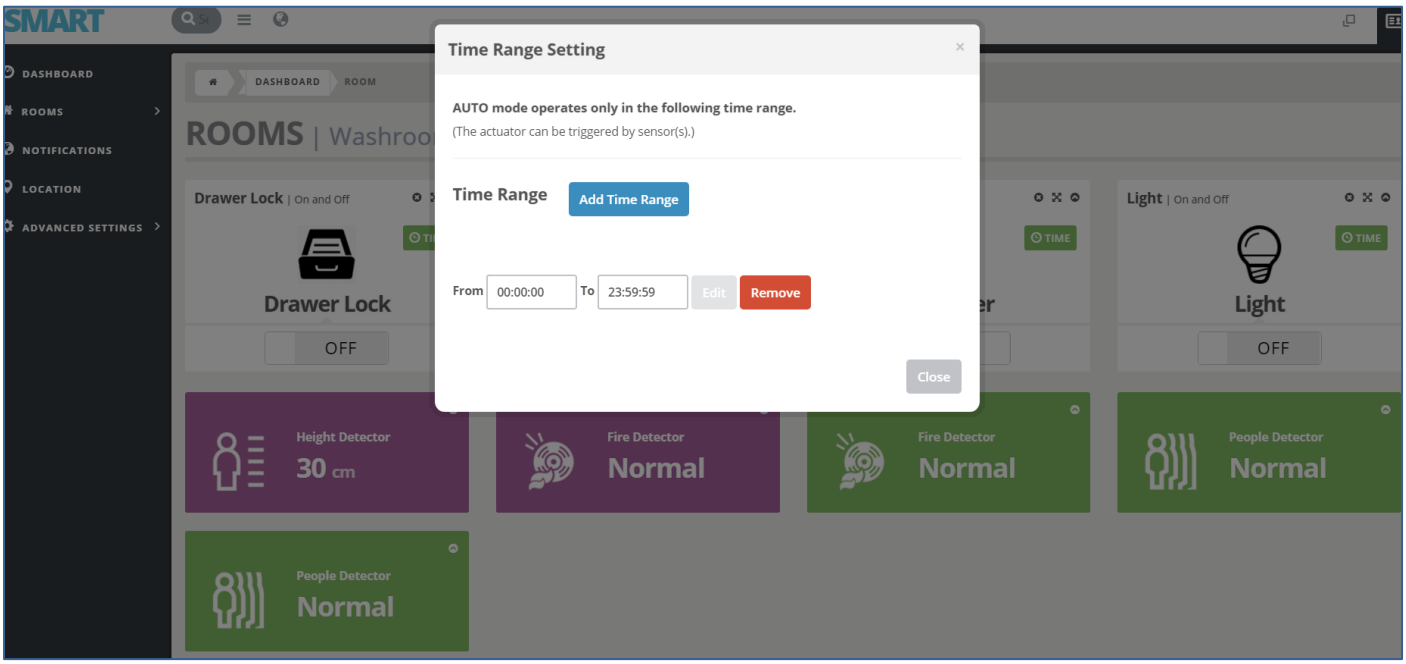
⇨ **Step 5: The account has been added and will be displayed in the "Users" area.**

7. Manage time range for an actuator



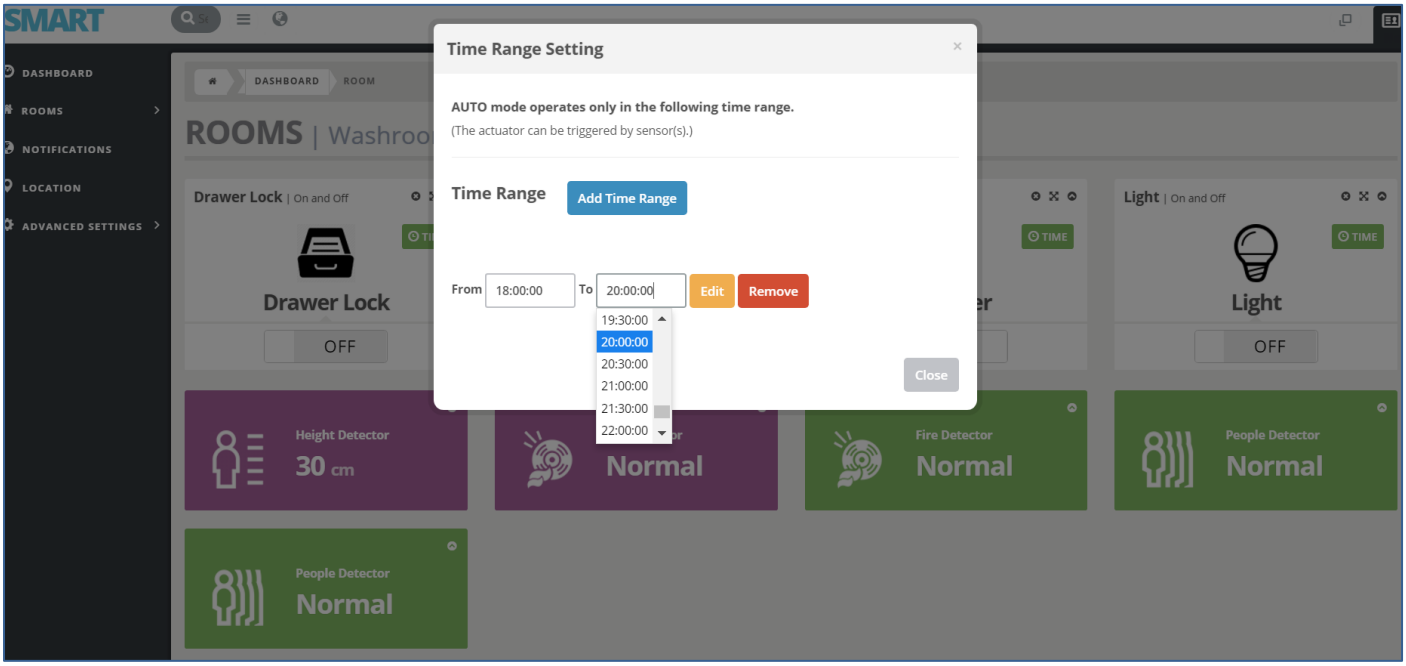
- Figure 66: User guide – Manage time range for an actuator (Website)

⇨ **Step 1: In the "Rooms" page, choose an actuator and click the "TIME" button.**



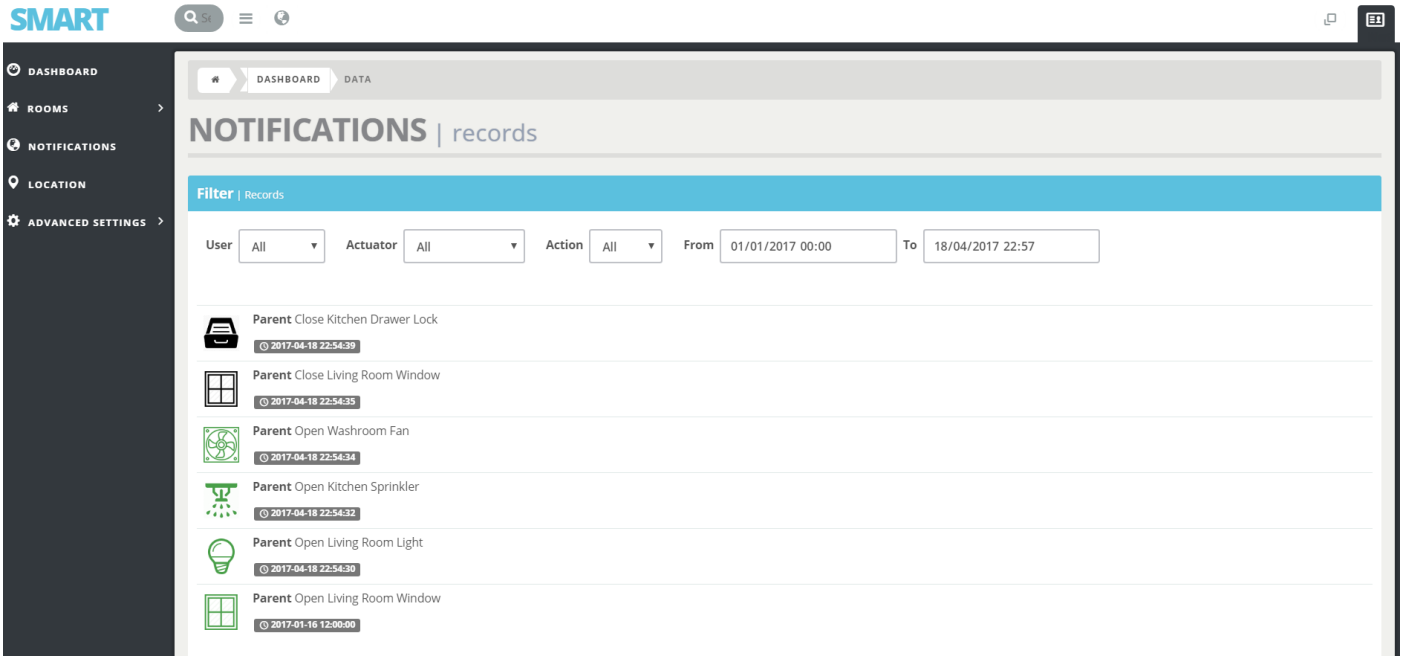
- Figure 67: User guide – Manage time range for an actuator with step 2 (Website)

⇨ **Step 2: You can add a new time range or change the period for editing an existed time range.**



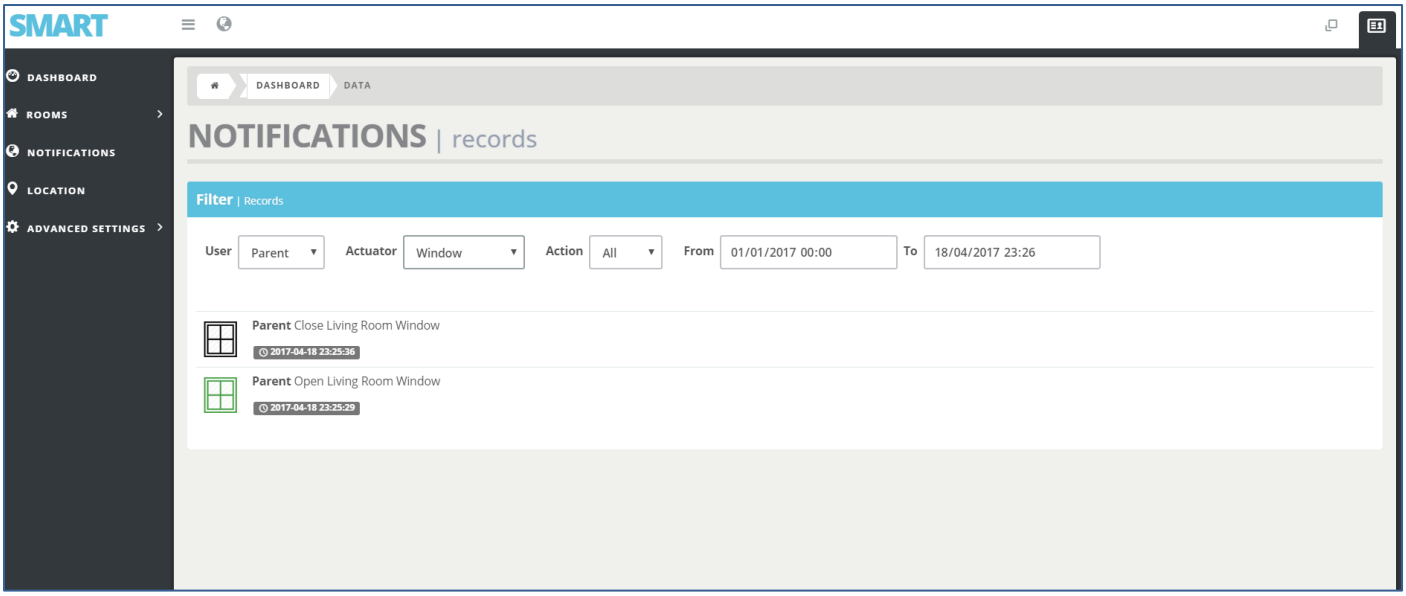
- Figure 68: User guide – Manage time range for an actuator with step 3 (Website)
- **Step 3: Click the "Edit" button for saving your new settings.**

8. Filter notification



- Figure 69: User guide – Filter notification (Website)

⇨ **Step 1: In the "Notifications" page, some history records will be displayed.**



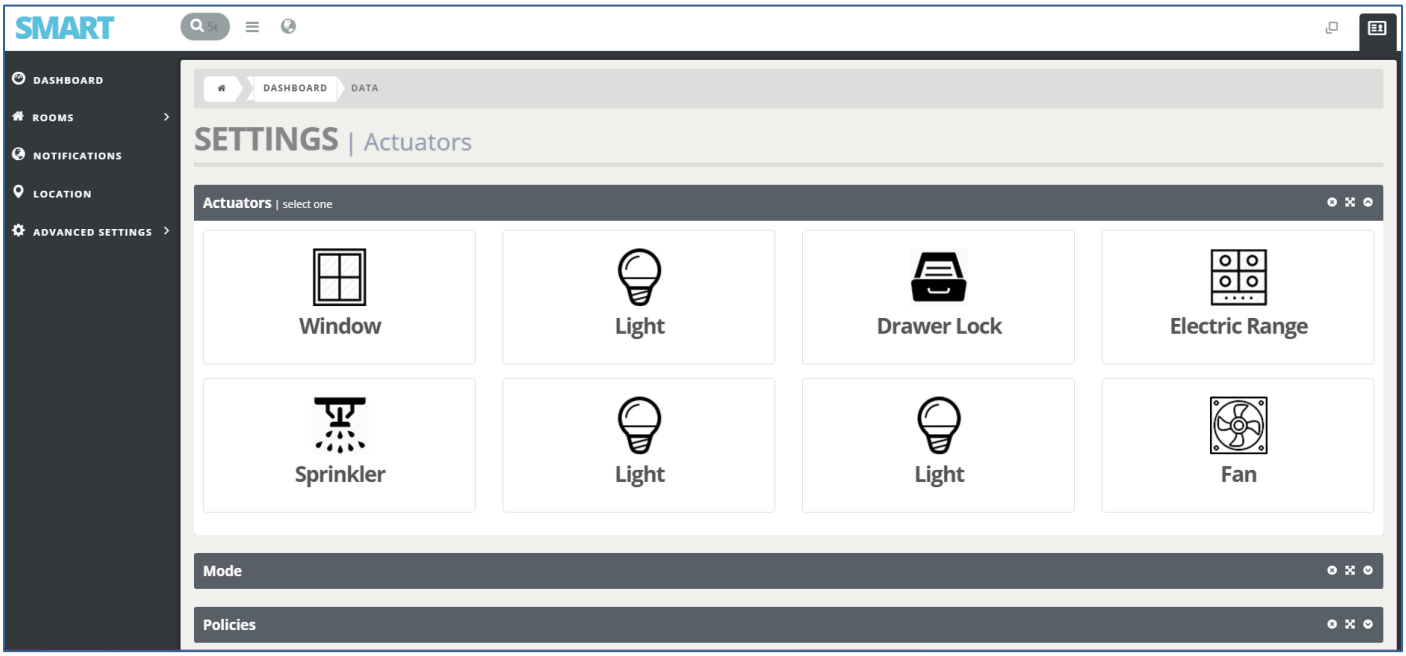
- Figure 70: User guide – Filter notification with step 2 (Website)

⇨ **Step 2: In order to find out records easily, some items for filter will be provided.**

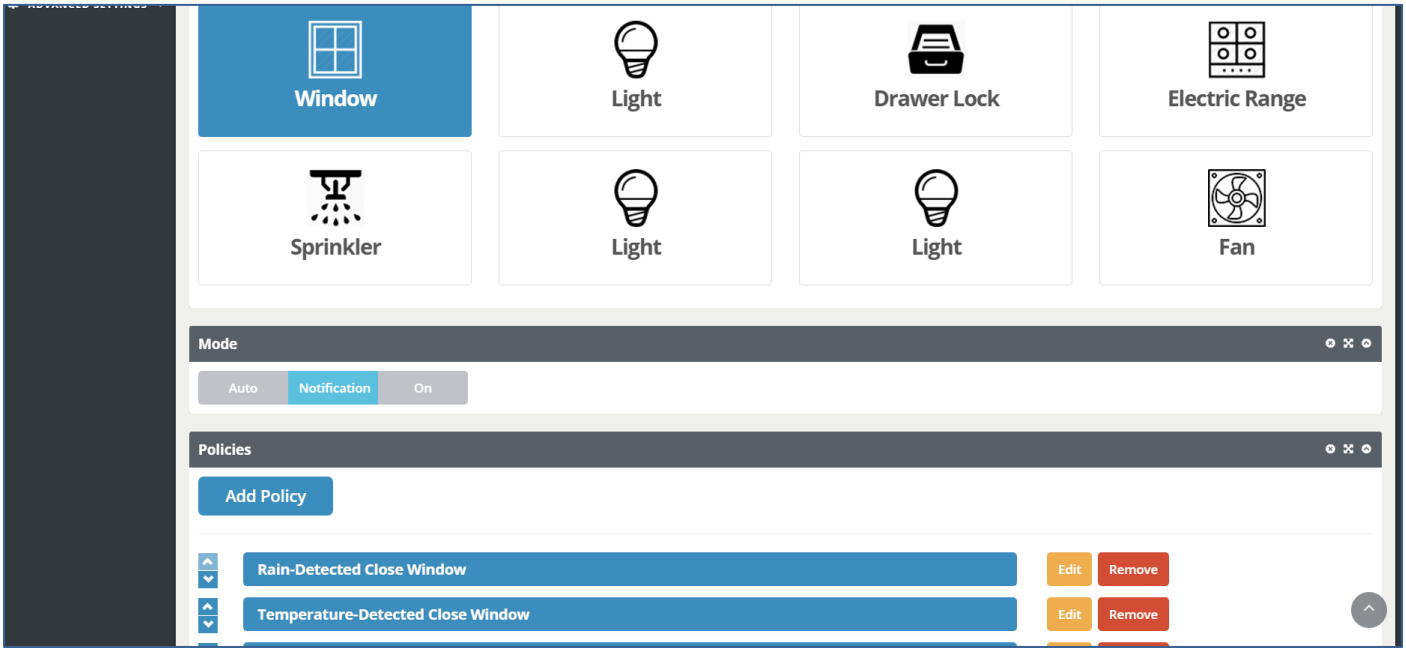
⇨ **Step 3: After selecting the filter items, the records will be updated immediately.**

⇨ **Step 4: In this case, some records of types of window based on parent will be displayed.**

9. Manage policy settings



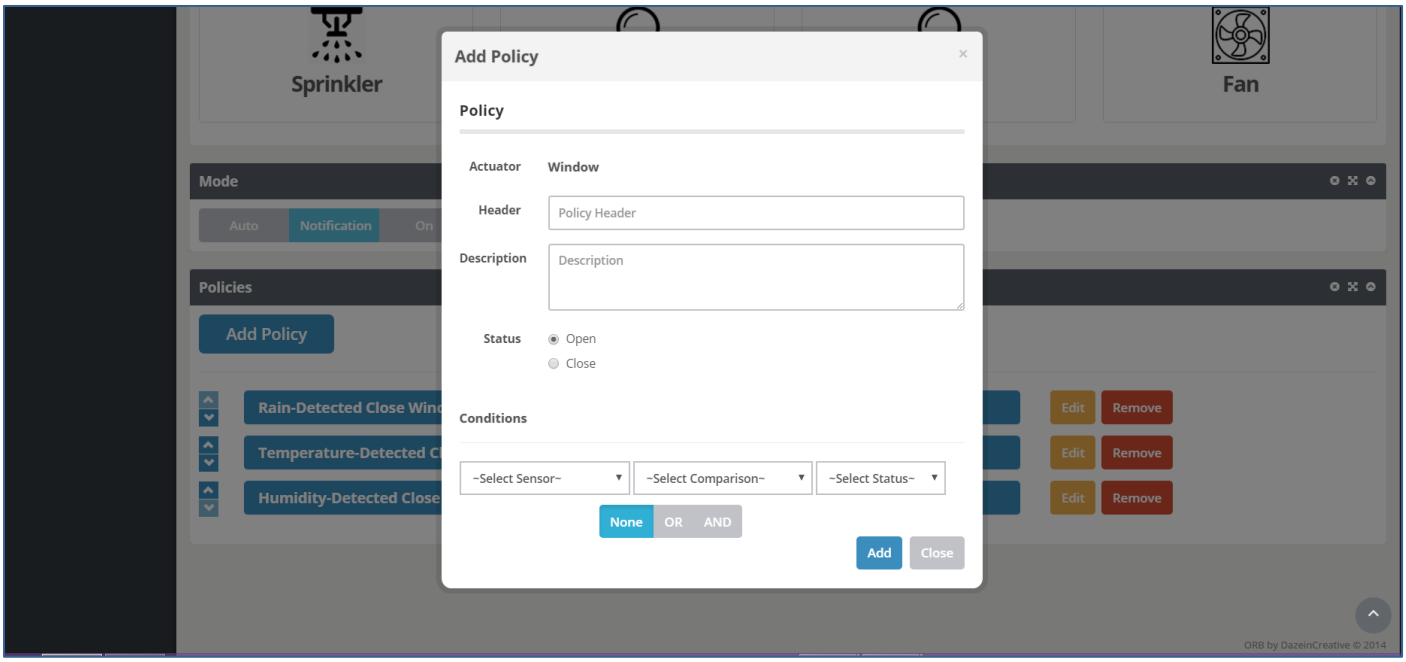
- Figure 71: User guide – Manage policy settings (Website)
- **Step 1: In the "Advanced Settings" page, some actuators are displayed for management.**



- Figure 72: User guide – Manage policy settings with step 2 (Website)

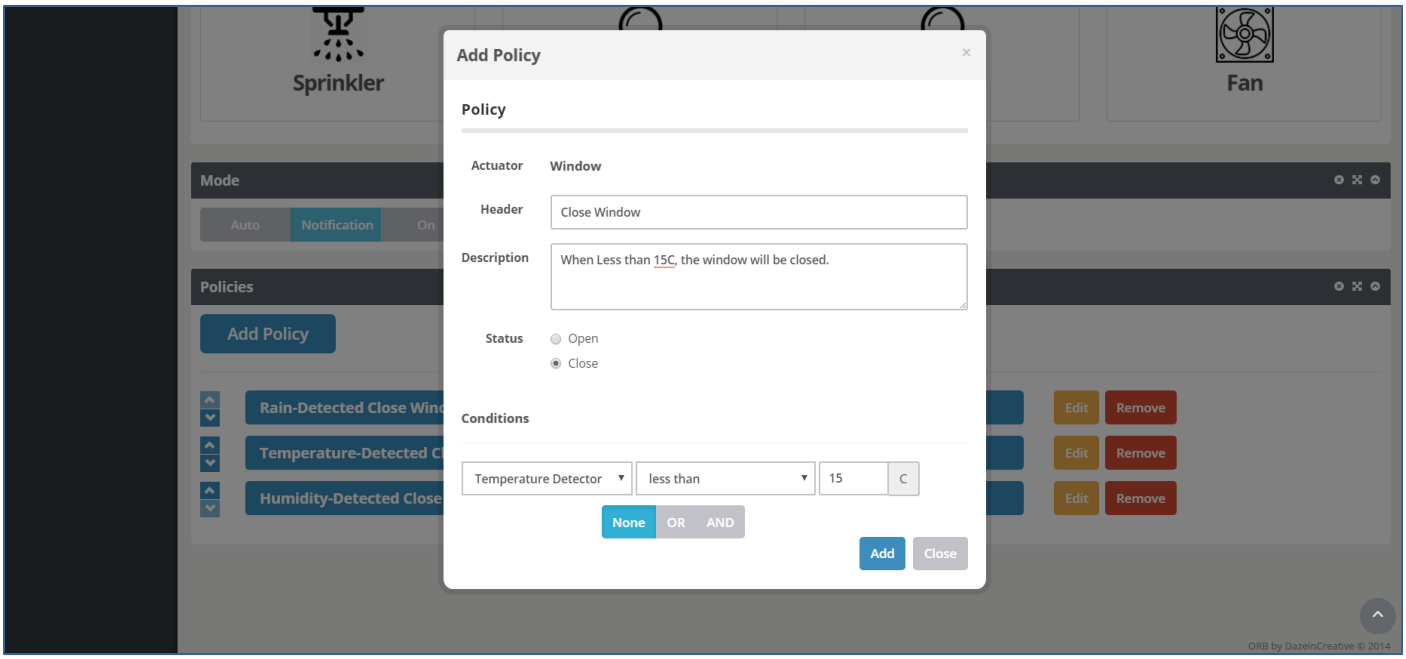
⇨ **Step 2: Click the "Window" actuator, its mode and policies will be displayed.**

⇨ **Step 3: Click the "Add Policy" button to add a new policy for this actuator.**

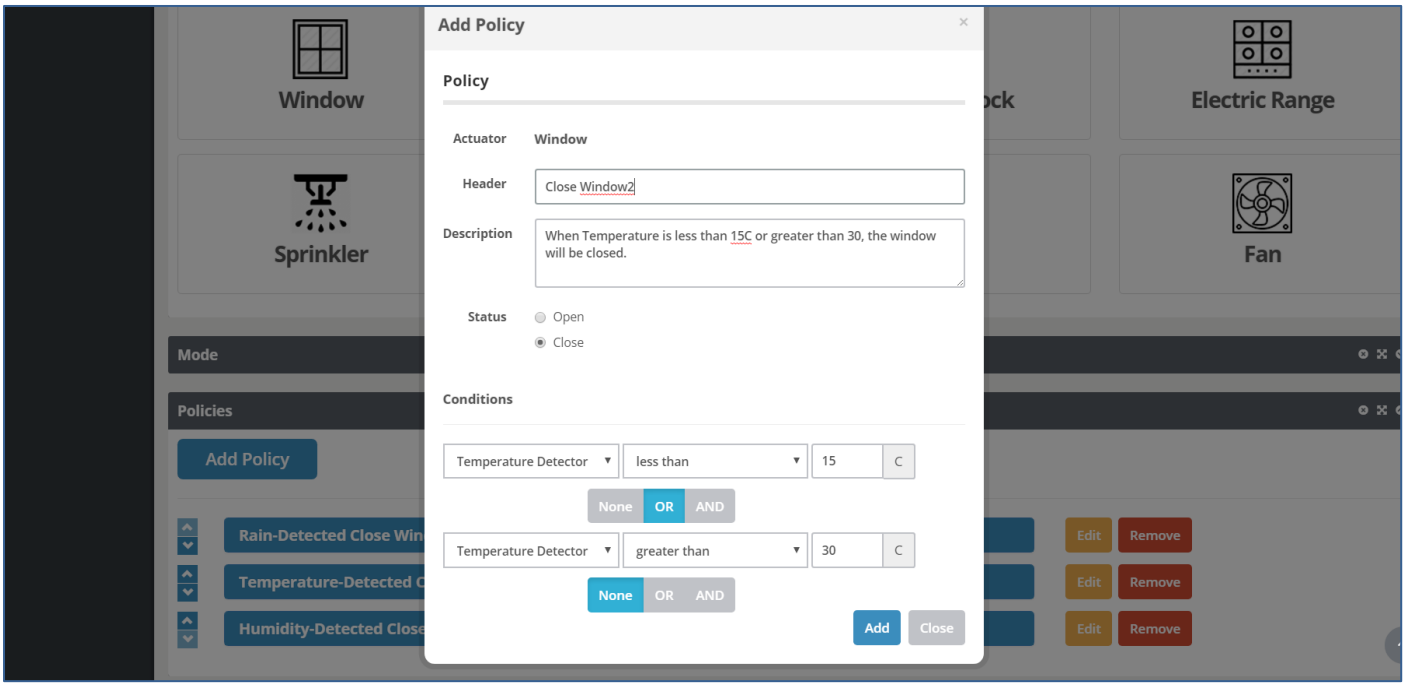


- Figure 73: User guide – Manage policy settings with step 3 (Website)
- **Step 4: Enter the policy information such as status and conditions**

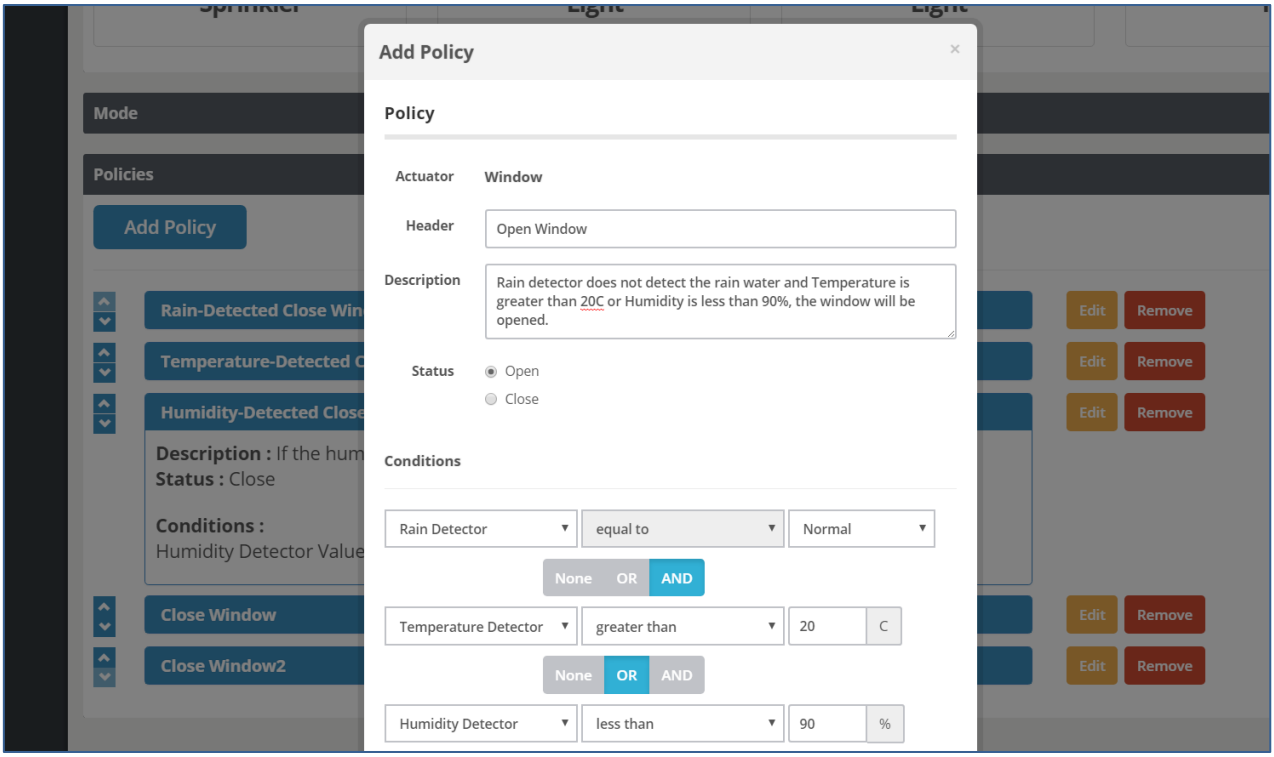
Some samples will be provided as follows:



- Figure 74: User guide – Sample 1 (Website)
- **When the temperature is less than 15C, the window will be closed automatically in Auto mode.**



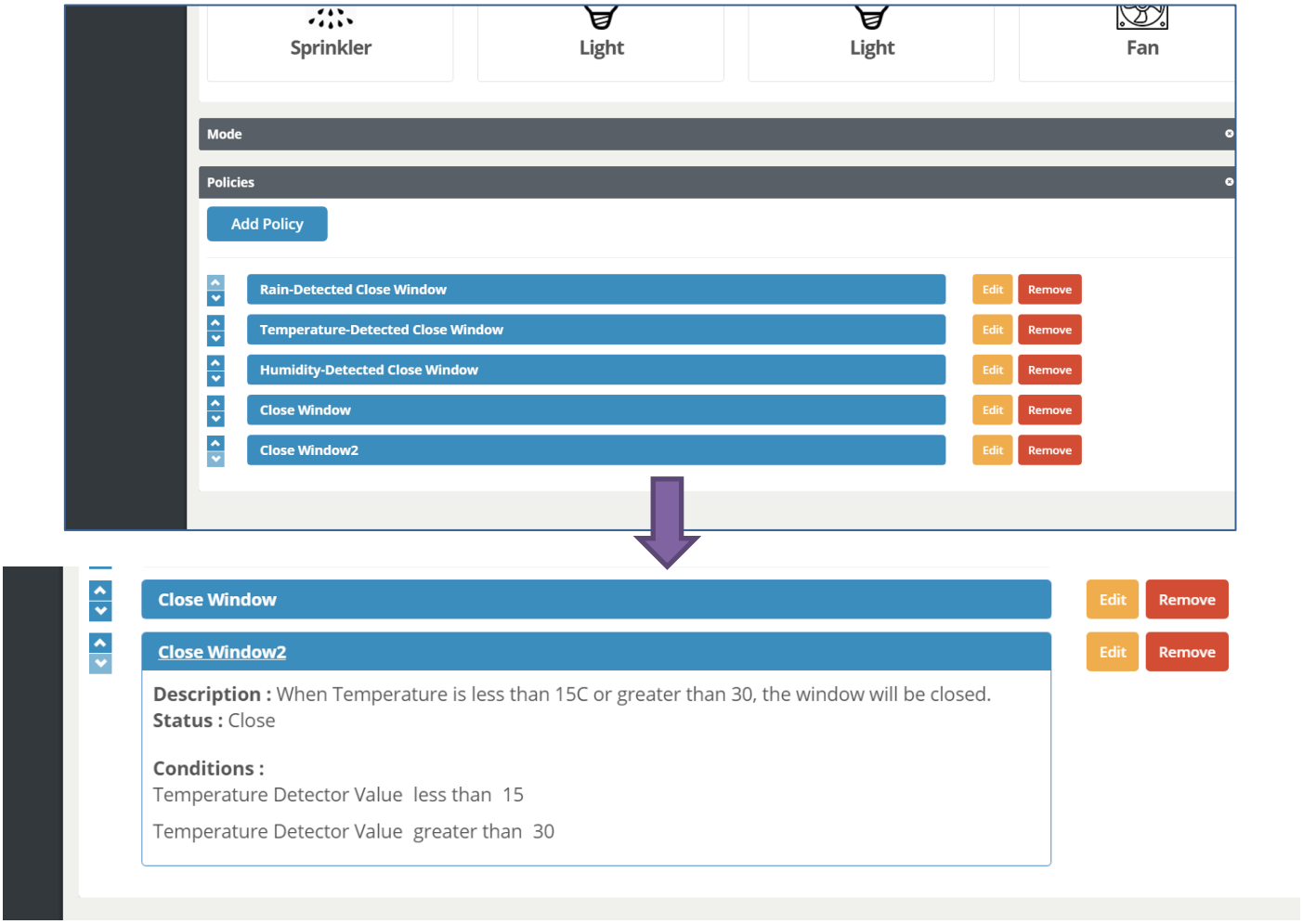
- Figure 75: User guide – Sample 2 (Website)
- **When the temperature is less than 15C or greater than 30C, the window will be closed automatically in Auto mode.**



- Figure 76: User guide – Sample 3 (Website)

⇨ **When the rain detector does not detect the rain water and the temperature is greater than 20C or the humidity is less than 90%, the window will be opened automatically in Auto mode.**

After setting the policy, the policy will be displayed and you can select them for detailed information.



- Figure 77: User guide – Sample 4 (Website)

6.3. Program listing

- **Control actuators**

  - **Handle function view**

```
public class HeaderFunctionView extends HeaderViewInterface<List<FunctionEntity>> {

   @BindView(R.id.expandableListView)
   ExpandableListView expandableListView;

   private String userID;
   private HeaderFunctionAdapter headerFunctionAdapter;

   public HeaderFunctionView(Activity context, String userID) {
      super(context);
      this.userID = userID;
   }

   private int previousGroup = -1;

   @Override
   protected void getView(List<FunctionEntity> functions, ListView listView) {
      View view = mInflate.inflate(R.layout.header_function_layout, listView, false);
      ButterKnife.bind(this, view);

      dealWithTheView(functions);
      listView.addHeaderView(view);
   }

   private void dealWithTheView(List<FunctionEntity> list) {
      headerFunctionAdapter = new HeaderFunctionAdapter(mActivity, list, userID);
      expandableListView.setAdapter(headerFunctionAdapter);

      expandableListView.setOnGroupExpandListener(new
MyExpandableListView.OnGroupExpandListener() {
         @Override
         public void onGroupExpand(int groupPosition) {
            if ((previousGroup != -1) && (groupPosition != previousGroup)) {
               expandableListView.collapseGroup(previousGroup);
            }
            previousGroup = groupPosition;
         }
      });
   }

   public void refresh() {
      headerFunctionAdapter.refresh();

      if (previousGroup != -1) {
         expandableListView.collapseGroup(previousGroup);
         expandableListView.expandGroup(previousGroup);
      }
   }
   public HeaderFunctionAdapter getHeaderFunctionAdapter() { return headerFunctionAdapter; }
}
```

- **Function adapter for handling parent ui**

```
public class HeaderFunctionAdapter extends BaseExpandableListAdapter {

   private Context context;
   private String userID;
   private List<FunctionEntity> functions = new ArrayList<FunctionEntity>();
   private ParentHolder parentHolder = null;
   private ChildHolder childHolder = null;
   private HeaderModuleAdapter headerModuleAdapter;

   public HeaderFunctionAdapter(Context context, List<FunctionEntity> functions, String userID) {
      this.context = context;
      this.functions = functions;
      this.userID = userID;
   }

   public void refresh() {notifyDataSetChanged();}

   public void setFunctions(List<FunctionEntity> functions) {this.functions = functions; }

   @Override
   public void registerDataSetObserver(DataSetObserver observer) {}

   @Override
   public void unregisterDataSetObserver(DataSetObserver observer) { }

   @Override
   public int getGroupCount() { return functions.size();}

   @Override
   public int getChildrenCount(int groupPosition) {return 1; }

   @Override
   public Object getGroup(int groupPosition) { return functions.get(groupPosition); }

   @Override
   public Object getChild(int groupPosition, int childPosition) { return functions.get(groupPosition); }

   @Override
   public long getGroupId(int groupPosition) { return groupPosition; }

   @Override
   public long getChildId(int groupPosition, int childPosition) {return childPosition; }

   @Override
   public boolean hasStableIds() {return true; }

   @Override
   public View getGroupView(int groupPosition, boolean isExpanded, View convertView, ViewGroup
parent) {
      FunctionEntity entity = (FunctionEntity) getGroup(groupPosition);

      if (convertView == null) {
         LayoutInflater mInflater = (LayoutInflater)
context.getSystemService(context.LAYOUT_INFLATER_SERVICE);
```

```
            convertView = mInflater.inflate(R.layout.fragment_home_item_parent, null);
            convertView.setHorizontalScrollBarEnabled(true);

            parentHolder = new ParentHolder();
            convertView.setTag(parentHolder);
        } else {
            parentHolder = (ParentHolder) convertView.getTag();
        }

        parentHolder.iv_image = (ImageView) convertView.findViewById(R.id.iv_image);
        ImageManager mImageManager = new ImageManager(context);
        mImageManager.loadDrawableImage(entity.getImage_url(), parentHolder.iv_image);

        parentHolder.tv_title = (TextView) convertView.findViewById(R.id.tv_title);
        parentHolder.tv_title.setText(entity.getTitle());

        parentHolder.tv_functionID = (TextView) convertView.findViewById(R.id.tv_functionID);
        parentHolder.tv_functionID.setText("Function ID：" + entity.getFunctionID());

        parentHolder.image_indicator = (ImageView) convertView.findViewById(R.id.image_indicator);
        if (isExpanded) {
            mImageManager.loadResImage(R.drawable.ic_keyboard_arrow_up_black_18dp,
parentHolder.image_indicator);
        } else {
            mImageManager.loadResImage(R.drawable.ic_keyboard_arrow_down_black_18dp,
parentHolder.image_indicator);
        }
        return convertView;
    }

    @Override
    public View getChildView(int groupPosition, int childPosition, boolean isLastChild, View convertView,
ViewGroup parent) {

        if (convertView == null) {
            LayoutInflater inflater = (LayoutInflater)
context.getSystemService(context.LAYOUT_INFLATER_SERVICE);
            convertView = inflater.inflate(R.layout.fragment_home_item_group_child, parent, false);
            childHolder = new ChildHolder();
            convertView.setTag(childHolder);
        } else {
            childHolder = (ChildHolder) convertView.getTag();
        }

        childHolder.modules = (RecyclerView) convertView.findViewById(R.id.modules);
        LinearLayoutManager layoutManager = new LinearLayoutManager(context,
LinearLayoutManager.HORIZONTAL, false);
        childHolder.modules.setLayoutManager(layoutManager);

        headerModuleAdapter = new HeaderModuleAdapter(context, functions.get(groupPosition).modules,
userID);
        childHolder.modules.setAdapter(headerModuleAdapter);
        return convertView;
    }
    @Override
    public boolean isChildSelectable(int groupPosition, int childPosition) { return false; }
```

```
    @Override
    public boolean areAllItemsEnabled() { return false; }

    @Override
    public boolean isEmpty() { return false; }

    @Override
    public void onGroupExpanded(int groupPosition) {}

    @Override
    public void onGroupCollapsed(int groupPosition) {}

    @Override
    public long getCombinedChildId(long groupId, long childId) { return 0; }

    @Override
    public long getCombinedGroupId(long groupId) { return 0; }

    public static class ChildHolder {static RecyclerView modules; }

    public static class ParentHolder {
        ImageView iv_image;
        TextView tv_title;
        TextView tv_functionID;
        ImageView image_indicator;
    }
}
```

- **Module adapter for handling child ui**

```
public class HeaderModuleAdapter extends RecyclerView.Adapter<HeaderModuleAdapter.ViewHolder> {

   private Context context;
   private List<ModuleEntity> modules = new ArrayList<ModuleEntity>();
   private ImageManager mImageManager;
   private String userID;

   public HeaderModuleAdapter(Context context, List<ModuleEntity> modules, String userID) {
      this.context = context;
      this.modules = modules;
      this.userID = userID;
      mImageManager = new ImageManager(context);
   }

   @Override
   public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
      LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
      View cardView = inflater.inflate(R.layout.fragment_home_item_child, null, false);
      ViewHolder viewHolder = new ViewHolder(cardView);
      viewHolder.iv_image = (ImageView) cardView.findViewById(R.id.iv_image);
      viewHolder.tvActuatorID = (TextView) cardView.findViewById(R.id.tvActuatorID);
      viewHolder.tvDeviceName = (TextView) cardView.findViewById(R.id.tvDeviceName);
      viewHolder.sw = (Switch) cardView.findViewById(R.id.sw);
      viewHolder.tag_group = (TagGroup) cardView.findViewById(R.id.tag_group);
      return viewHolder;
   }

   @Override
   public void onBindViewHolder(final ViewHolder holder, final int position) {
      mImageManager.loadDrawableImage(modules.get(position).getImage(), holder.iv_image);
      holder.tvActuatorID.setText(modules.get(position).getActuatorID() + "");
      holder.tvDeviceName.setText(modules.get(position).getDeviceName() + "");

      Switch sw = holder.sw;
      sw.setChecked(modules.get(position).isStatus());

      sw.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
         @Override
         public void onCheckedChanged(CompoundButton compoundButton, boolean status) {
            int iStatus = (status)? 1:2;

            //check permission
            if ((iStatus == 1 && ( modules.get(position).getPermissionID() == 1 ||
modules.get(position).getPermissionID() == 3))) {
               holder.sw.setChecked(false);
            } else if (iStatus == 2 && ( modules.get(position).getPermissionID() == 1 ||
modules.get(position).getPermissionID() == 2)) {
               holder.sw.setChecked(true);
            } else {
               //update actuator's status
               try {
                  MainActivity mainActivity = new MainActivity();
                  Connection conn = new ConnectionFromOkHttp3(mainActivity.getPath() + "actuators");
```

```
                conn.post("userID", userID);
                conn.post("actuatorID", modules.get(position).getActuatorID() + "");
                conn.post("status", iStatus + "");
                conn.send("PutRequest");
              } catch (TimeoutException e) {
                e.printStackTrace();
              }
            }
          }
        });

        holder.tag_group.setTags(modules.get(position).getPermissionDescription());
    }

    public void setModules(List<ModuleEntity> modules) {
        this.modules = modules;
        notifyDataSetChanged();
    }

    @Override
    public int getItemCount() {return modules.size();}

    @Override
    public void onAttachedToRecyclerView(RecyclerView recyclerView) {
        super.onAttachedToRecyclerView(recyclerView);
    }

    public static class ViewHolder extends RecyclerView.ViewHolder {

        ImageView iv_image;
        TextView tvActuatorID;
        TextView tvDeviceName;
        Switch sw;
        TagGroup tag_group;

        public ViewHolder(View itemView) {
            super(itemView);
            iv_image = (ImageView) itemView.findViewById(R.id.iv_image);
            tvActuatorID = (TextView) itemView.findViewById(R.id.tvActuatorID);
            tvDeviceName = (TextView) itemView.findViewById(R.id.tvDeviceName);
            sw = (Switch) itemView.findViewById(R.id.sw);
            tag_group = (TagGroup) itemView.findViewById(R.id.tag_group);
        }
    }
}
```

- **Part of HomePageFragment for receiving the Pusher message**

```
//build pusher
      Pusher pusher = new Pusher("1bc88481522c1ebbf138");

      Channel channel = pusher.subscribe("actuator");

      //when receiving message, do the action
      channel.bind("updateStatus", new SubscriptionEventListener() {
        @Override
        public void onEvent(String channelName, String eventName, String jsonData) {
          try {

headerFunctionView.getHeaderFunctionAdapter().setFunctions(ModelUtil.getFunctionData(mainActivity));
            headerFunctionView.refresh();
          } catch (Exception e) {
            e.printStackTrace();
          }
        }
      });

pusher.connect();
```

6.4. Original project plan

- The project plan is showing the main stages involved in the development of digital home system. And it include with estimated time to completion of each stage of project.

Planning Phrase

- The stage of understanding the scenario and combine the user requirements together, write a project proposal with problems and solutions.
  - Deliverables: Project Proposal [27 / 9 / 2016]

Analysis Phrase

- The stage of problems analysis and system analysis of the digital home system, and find out the suitable solution and the ways for the system development.
  - Deliverables: Initial Report [28 / 11 / 2016]

Design Phrase

- The stage of ideas with the details in digital home system, real model building, and the user interface design.
  - Deliverables: Interim Report [16 / 01 / 2017]

Implementation Phrase

- The stage of implementing the test of system website and the mobile application with the real model building.
  - Deliverables: Final Report & Digital home system [19 / 04 / 2017]

Details	SEP,2016	OCT,2016	NOV,2016	DEC,2016	JAN,2017	FEB,2017	MAR,2017	APR,2017
Project Proposal								
Initial Report								
Interim Report								
Prototype 1								
Presentation (Prototype 1)								
Prototype 2								
Testing (Prototype 2)								
Presentation (Prototype 2)								
Final Report								
Final testing								
Presentation(Final Version)								

- Table 63: Original project plan – Gantt chart

6.5. Log sheet

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 1)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 9th September 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
First Meeting Minutes Record – HKSTP Meeting ▪ First meeting with different company in HKSTP ▪ Topic discussion of digital home system ▪ To understanding of different scopes of final year project	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Information searching ▪ Research with different program IDE builder ▪ Think of the user interface design	Au Chi Chung

Supervisor's comment on student's work
▪ Information searching

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 2)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 20th September 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Second Meeting Minutes Record – IVE Supervisor Meeting ▪ Topic discussion of digital home system ▪ To understanding of digital home system of final year project	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
▪ Information searching ▪ Research with different program IDE builder ▪ Think of the user interface design	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Information searching ▪ Research with different program IDE builder ▪ Think of the user interface design	Au Chi Chung

Supervisor's comment on student's work
▪ Information searching ▪ Decide the topic of final year project and preparation of the topic of Digital Home

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 3)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 27th September 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Third Meeting Minutes Record – HKSTP METAS Meeting ▪ Submit project proposal ▪ In touch with different sensors in METAS and learn how to use of the METAS Node One	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
▪ Research with different program IDE builder	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Research with different program IDE builder ▪ User interface design ▪ System design ▪ API searching	Au Chi Chung

Supervisor's comment on student's work
N/A

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 4)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 4th October 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Fourth Meeting Minutes Record – IVE Supervisor Meeting ▪ Accept the sample model of the digital home figure ▪ Understanding of the initial system with function requirements	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
▪ User interface design ▪ System design ▪ API searching	Au Chi Chung
Plan of work before the next recording	Logged by
▪ User interface design ▪ System design ▪ API searching ▪ Learn on the sample model of the digital home figure	Au Chi Chung

Supervisor's comment on student's work
▪ Learn on the sample model of the digital home figure

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 5)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 11st October 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Fifth Meeting Minutes Record – IVE Supervisor Meeting ▪ Accept different ARDUNIO sensors for learning ▪ Understanding of the initial system with Entity Relationship Diagram (ERD)	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
▪ API searching	Au Chi Chung
Plan of work before the next recording	Logged by
▪ API searching ▪ Learn on the sample model of the digital home figure	Au Chi Chung

Supervisor's comment on student's work
▪ Learn different ARDUNIO sensors for learning

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 6)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 18th October 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Sixth Meeting Minutes Record – HKSTP METAS Meeting ▪ Accept SNAP practice with 2 figure (robot and mechanical arm with sensors) ▪ Accept 2 of METAS Node One Controller ▪ Accept 4 ARDUNIO sensors (flame sensor, infrared receiver and emission, laser emit) ▪ Accept the practice assignment with the provided devices (videos and user guide)	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	**Logged by**
▪ Complete the SNAP practice with 2 figure ▪ Learn how to use with the METAS Node One Controller ▪ Learn how to use with 4 ARDUNIO sensors ▪ Complete the practice assignment with provided devices	Au Chi Chung

Supervisor's comment on student's work
N/A

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 7)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 25th October 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Seventh Meeting Minutes Record – IVE Supervisor Meeting ▪ Discuss the use of any sensors and testing of Node One connection ▪ Prepare the practice assignment of robot and take a video record for robot control	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
▪ Complete the practice assignment with provided devices ▪ Complete the SNAP practice with 2 figure	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Complete the SNAP practice with 2 figure ▪ Complete the practice assignment with provided devices	Au Chi Chung

Supervisor's comment on student's work
N/A

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 8)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 8th November 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Eighth Meeting Minutes Record – IVE Supervisor Meeting ▪ Modification of the system function	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Modification of the system function	Au Chi Chung

Supervisor's comment on student's work
▪ Modification of the system function

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 9)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 15th November 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Ninth Meeting Minutes Record – IVE Supervisor Meeting ▪ The requirement work with build up a simple sensor environment (METAS Competition Set)	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Build up a simple sensor environment (METAS Competition Set)	Au Chi Chung

Supervisor's comment on student's work
▪ Build up a simple sensor environment (METAS Competition Set)

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 10)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 22rd November 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Tenth Meeting Minutes Record – IVE Supervisor Meeting ▪ To discuss of the initial report with any modification ▪ Check the process rate of the project	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Process with the system development	Au Chi Chung

Supervisor's comment on student's work
▪ Start to process with the system development

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 11)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 29th November 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Eleventh Meeting Minutes Record – IVE Supervisor Meeting ▪ To discuss of the system progress and the sensors value list ▪ Decide the main control panel to the system model	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Decide the main control panel to the system model	Au Chi Chung

Supervisor's comment on student's work
▪ Decide the main control panel to the system model

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 12)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 6th September 2016
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Twelfth Meeting Minutes Record – HKSTP METAS Meeting ▪ Instrument : - Two of Node One ver.1 with two charging wire - One of 8266 ARDUINO Wi-Fi module - Two of input / output ARDUINO sensor adapter - Two input / output extension wire - Distance / motion trigger - "METAS" OR gate - Video of Program IDE - Sensors testing result statement	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	Logged by
▪ To do List : - Provide the report in word document (Confirm all modules, type of sensors) - Take communication between all sensor / output and server - Add sensors API in the METAS Node One - Start to build the demo model as requested case	Au Chi Chung

Supervisor's comment on student's work
N/A

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 13)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 17th January 2017
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Thirteenth Meeting Minutes Record – IVE Supervisor Meeting ▪ To discuss of the system progress interim report	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	Logged by
▪ To discuss of the system progress interim report	Au Chi Chung

Supervisor's comment on student's work
▪ Feedback for the interim report

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 14)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 14th February 2017
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Fourteenth Meeting Minutes Record – IVE Supervisor Meeting ▪ To discuss of the suggested modification in the interim report ▪ To discuss of the suggested modification in the system	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	Logged by
▪ To discuss of the suggested modification in the system	Au Chi Chung

Supervisor's comment on student's work
▪ Feedback for the interim report and the system progress

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 15)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 21st February 2017
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Fifteenth Meeting Minutes Record – IVE Supervisor Meeting ▪ Follow the progress in the final year project	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
▪ Progress with the system development	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Follow the progress in the final year project	Au Chi Chung

Supervisor's comment on student's work
N/A

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 16)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 28th February 2017
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Sixteenth Meeting Minutes Record – HKSTP METAS Meeting ▪ Take a presentation to report the progress to METAS	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Follow the progress in the final year project	Au Chi Chung

Supervisor's comment on student's work
N/A

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 17)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 7th March 2017
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Seventeenth Meeting Minutes Record – IVE Supervisor Meeting ▪ Discuss the progress for the final year project	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Follow the progress in the final year project	Au Chi Chung

Supervisor's comment on student's work
▪ Feedback with progress for the final year project

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 18)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 21st March 2017
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Eighteenth Meeting Minutes Record – IVE Supervisor Meeting ▪ Discuss the progress for the final year project ▪ Take the new materials of the real digital home model	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	**Logged by**
▪ Follow the progress in the final year project ▪ Test of the new materials of the real digital home model	Au Chi Chung

Supervisor's comment on student's work
▪ Feedback with progress for the final year project ▪ Test of the new materials of the real digital home model

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 19)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 28th March 2017
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Nineteenth Meeting Minutes Record – IVE Supervisor Meeting ▪ Discuss the progress for the final year project ▪ Building with the real model	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	Logged by
▪ Follow the progress in the final year project ▪ Building with the real model	Au Chi Chung

Supervisor's comment on student's work
▪ Feedback with progress for the final year project ▪ Test of the new materials of the real digital home model ▪ Building with the real model

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 20)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 11st April 2017
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Twentieth Meeting Minutes Record – IVE Supervisor Meeting ▪ Discuss the progress for the final year project ▪ Building with the real model	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	**Logged by**
▪ Follow the progress in the final year project ▪ Building with the real model	Au Chi Chung

Supervisor's comment on student's work
▪ Feedback with progress for the final year project ▪ Feedback with the digital home system real model building

Student's Signature:	Supervisor Signature

Hong Kong Institute of Vocational Education

Discipline of Information Technology

Final Year Project - Log Sheet (No. 21)

Course/Class: IT114015 / SE2A Project ID: Group 23 (Johnny Group 4)
Supervisor: Johnny Kwong Date: 18th April 2017
Project Title: Digital Home System in IOT environment with sensor control model
Student Name(s): Au Chi Chung, Kwok Yuk Lam, Li Kai Kwong, Yu Kwok Ho

Work done & findings since last recording	Logged by
Twenty-first Meeting Minutes Record – IVE Supervisor Meeting ▪ Discuss the progress for the final year project ▪ Building with the real model	Au Chi Chung
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan	Logged by
N/A	Au Chi Chung
Plan of work before the next recording	**Logged by**
▪ Follow the progress in the final year project ▪ Building with the real model	Au Chi Chung

Supervisor's comment on student's work
▪ Feedback with progress for the final year project ▪ Feedback with the digital home system real model building

Student's Signature:	Supervisor Signature