# Hololens for Home Energy Management System


Peeranut Chindanonda
Thisada Sutthitham
Thitiwat Watanajaturaporn



Bachelor of Engineering in Software Engineering
International College
King Mongkut's Institute of Technology Ladkrabang
Academic Year 2016
KMITL-2017-IC-B-003-005

**Thesis - Academic Year 2016**

Bachelor of Engineering in Software Engineering

International College

King Mongkut's Institute of Technology Ladkrabang

**Title**: Hololens for Home Energy Management System


**Authors**

1. Mr. Peeranut Chindanonda Student ID: 56090016
2. Ms. Thisada Sutthitham Student ID: 56090030
3. Mr. Thitiwat Watanajaturaporn Student ID: 56090031


Approved for submission

V. Hirankitti

.....

(Assistant Professor Dr. Visit Hirankitti)

Advisor

Date 4/6/2017

# Abstract

In recent years, the world is moving toward green technologies, the technologies which can alleviate global warming. A smart home with advanced energy management is one of those technologies. Smart home systems are being controlled by a home gateway on which HEMS (home energy management systems) is operating. HEMS can be perceived as a centralized system which connects to every home appliance and sensor in the house to monitor and control appliances to achieve home energy efficiency. In this project, we have developed multiple user interfaces to connect augmented reality (Microsoft Hololens) device to HEMS. The first interface we have developed was a web application which runs on a web browser on the Microsoft Hololens. The web application provides users with an interface to control and monitor smart home devices. However, using the web interface is not a natural way to interact with a smart home. So, we then developed an augmented reality application, where we can interact with a virtual object as if we interacted with the real one. Therefore, we have made an AR application using C# and Unity as a 3D engine. The AR application is location-aware using sensor information from Microsoft Hololens. We use location awareness to display appliance's control panel next to the appliance. Both AR and web applications enable users to use speech command to interact with HEMS. The speech is translated into text using an API and then send to API.AI platform for processing user's intention. After we can identify user's intention, we then process what we have to do and which appliance the user wants to take action on with HEMS. Both AR and web can also update appliances data in a real-time manner using WebSocket.

# Acknowledgements

The project could not be finished unless there were advices from our advisor and other lecturers. First, we would like to give our deep gratitude to our advisor, Asst.Prof.Dr. Visit Hirankitti for his generous support on our project since the first day. We gained several knowledge, suggestions, comments, and remarkable experience from him.

Also, we would like to express our gratitude to Asst.Prof.Dr. Chaiwat Nuthong, Dr. Isara Anantavrasilp, Dr. Ukrit Watchareeruetai, and Dr. Natthapong Jungteerapanich who also proposed some helpful suggestions.

Lastly, we would like give thanks to our colleagues including friends for the support and help.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

Over past few years, Internet of things(IoT) was also made known and became widely popular. The concept of Internet of things has emerged according to a convergence of different technologies incorporated in together. All appliances are Internet-connected. Furthermore, they are installed with electronics, software, sensors, actuator, and network connectivity that permit these devices to gather and exchange data to the main system. This concept of IoT can lead to another effective concept by applying the concept IoT with home, which led to 'Smart Home' concept. There was a concept of smart home or home automation introduced for a couple of years. The concept of smart home became worldwide and well-known. Its purpose is to provide comforts with ease-of-use features to home's inhabitants. The home is controlled by a centralized computer to control a building's lighting, air-conditioning, and other home appliances via the building management system. Each appliance has its mechanical task to deal with events. However, the most essential part of a smart home is Home Energy Management System (HEMS) [2]. HEMS is kind of a software that operates the smart home by monitoring and controlling all electrical appliances in such a way to control the electricity usage in an efficient way. For example, arranging electricity usage during periods of low demand to pay cheaper rates [1].

Last year, there was an announcement of the launch of a new product from Microsoft. It was called Microsoft Hololens; a fully self-contained holographic computer running on Windows 10. It comes with the technology of augmented reality that can display holograms in your environment and provide a new experience - the experience of virtual reality and augmented reality holding together [4]. Your new world on Microsoft Hololens will be overlaid by augmented reality which makes you feel like they are part of your world. Figure 1-1 is an example of what user's eyes see when they look through Hololens. With an advent of Microsoft HoloLens, we would like to explore the possibilities on how this new paradigm of a computing device can replace the In-Home-Display to use with HEMS in the future.

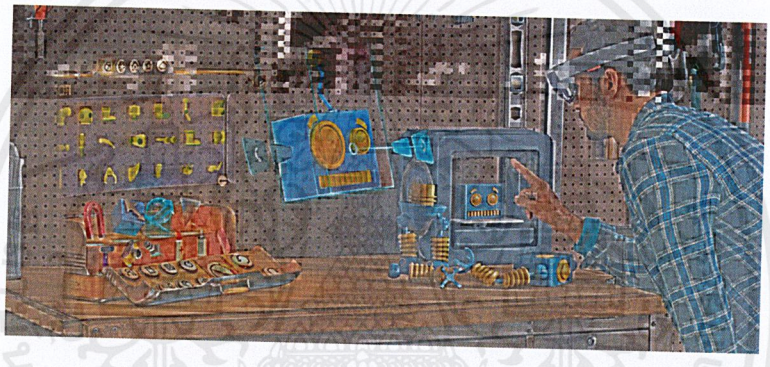


Figure 1-1: A new world of augmented reality with hololens [3]

In this thesis, we shall propose a use of Microsoft Hololens as a device of virtual interface to HEMS. This is an extension to the PEA Smart Home Project which was funded by Provincial Electricity Authority, shortly call PEA. The PEA Smart Home is managed by a home energy management system (HEMS) which aims at the energy consumption of the home. HEMS can also be accessed by using a mobile application on a mobile device, or so-called "In-Home-Display". In this thesis, we want to go beyond that by providing augmented reality interfaces to HEMS using Microsoft Hololens.

## 1.2 Problem Statement

Generally, in order to control and interact with a smart home, many solutions were proposed, e.g., smartphone application or web-based application. Users will have to open up the application from his device and monitor appliances from the device's screen.

However, accessing a smart home through a mobile application or web application may not be a natural way to interact with the home. If someone had difficulty in using such devices, they would not be able to manage a smart home. Also, user's experience would not be great.

The augmented reality in a smart home system is proposed. It is truly expected to serve better user's experiences with ease-of-use in the interaction between users and the smart home via the system.

We believe the easiest way to operate a smart home is by using an AR headset.

## 1.3 Objective

This project intends to develop an alternative approach of user interface by using Microsoft Hololens to let any smart home's users be able access, visualize and control the smart home in such a way that it is more natural and responsive than using a mobile application. Therefore, in order to indicate the completeness of the project, several goals have been set. The following list presents all the goals which are needed to be accomplished:

- To use Microsoft Hololens for home energy management system (HEMS).
- To review the performance and possibility on how Microsoft Hololens can be used as an interface device for HEMS.
- To present all required functionalities from HEMS to the Microsoft Hololens, for example, detailed information of appliances, real-time pricing data.

## 1.4 Scope of work

In this project, we focus on implementing augmented reality smart home application for Microsoft Hololens. Moreover, there will also be a smart home web application to interface with HEMS. The scope of this project can be listed as follows:

- To study a smart home system and home energy management system (HEMS).
- To develop a real-time Microsoft Hololens application to interface with HEMS.
- To develop smart home web application as an alternative way to interface with HEMS.

## 1.5 Thesis structure

This thesis contains eight chapters which are organized as follows.

- Chapter 1 Introduction - refers to the introduction, problem statement, objective, scope of work of the project, and structure of this thesis.
- Chapter 2 Related Works - describes the literature review and also shows concepts that are significantly relevant to this project.
- Chapter 3. Smart Home and Augmented Reality - which will explain the theory and other knowledge relevant to the project.
- Chapter 4 Requirement Analysis - Explains the requirements (functional requirements/non-functional requirements) of the system and includes use case diagrams with explanation, relevant system analysis diagrams (such as activity diagrams).
- Chapter 5 System Design - Explain overall architecture/structure of the software including relevant diagram i.e. class diagrams, sequence diagram.

- Chapter 6 System Development - Explain the development process, techniques, tools to be used in the project, and algorithms.
- Chapter 7 Experiment and Results contents are about an experimental setup, analysis, or results.
- Chapter 8 Conclusion- is the final chapter that talks about the conclusion, some suggestion, any further development, or improvements to the project.

# Chapter 2

# Related Works

There are many approaches for smart home with different kinds of interface, techniques and technologies. Some approaches that are relevant to our project are discussed as follows.

## 2.1 3D Web Interfaces for Smart Home

Wenshan Hu [9] proposed web-based 3D uses flash 3D engine to create virtual home in user's web browser. See Figure 2-1. Users are able to remotely control and monitor any appliances in their virtual home by clicking mouse or keyboard and control panel will appear.

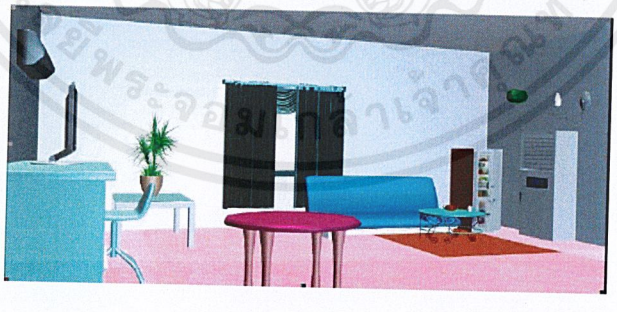


Figure 2-1: 3D room Displayed in web-based interface [9]

In our project, we also have Smart Home 3D model in our website but it can not control directly using 3D model like this approach. User needs to control in appliances page. However, When something changes, it will appear in real-time.

In the Table 2.1, we compare previous research that has 3D web interface with our project.

| **Project** | **Remote Control System Of Web-Based Smart Home** | **Our Project** |
|---|---|---|
| Tools for 3D web | flash 3D engine | WebVR/WebGL |
| Render in web | yes | yes |
| Room generated from | 3D model | Json file |
| Hardware for user | Smart phone or anything that can access web | Smart phone or anything that can access web |
| Control appliance from 3D model | Yes | No |
| See real time changes when control appliance | Yes | Yes |
| Import or export 3D model | No | Yes |

Table 2.1: Comparison of research paper in 3D Web Interfaces

## 2.2 Augmented Realities for Smart Home

Snehal Nargundi [7] proposed the mobile application that allows user to control on/off of the appliances. This application works when user hold mobile camera up to real appliance then virtual switch will pop up to mobile screen and user can control this switch.

In our project we also use AR technology but the different is we use AR in Hololens, and when we look at the real object you will see the panel for control and information instead of the virtual object.

In the Table 2.2, we compare previous research about augmented reality with our project. It is obvious that our project is more interactive and natural.

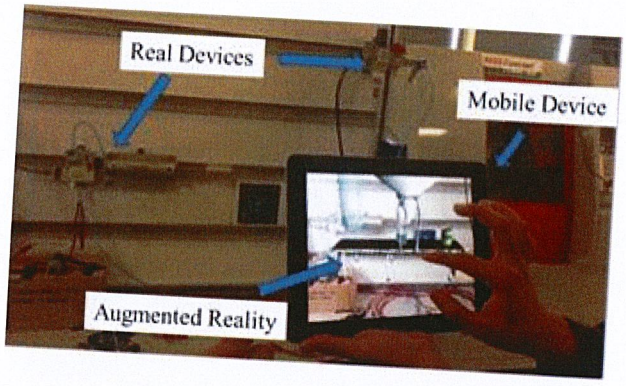



Figure 2-2: Overall system [7]

| **Project** | **A Review on Home Automation using Augmented Reality** | **Our project** |
|---|---|---|
| Idea | Control appliances by using virtual switch | Control appliances by using virtual panel or voice commands |
| Features | Set on/off | Set on/off, brightness, temperature, color |
| Hardware for user | Mobile device | Hololens |
| Technology | AR | AR |

Table 2.2: Comparison of research paper in VR

## 2.3 Smart Home Control

Pieter Simoens [6] proposed a smart home control with head-mounted sensors for vision and brain activity where the authors used a commercial Emotiv EEG neuro-headset to capture a command from brain activity and used the Google Glass for visual processing using OpenCV.

The main idea from this paper was to combine two technologies of brain activity from the Emotiv EEG neuro-headset and Google Glass to let users able to adjust an actuator by gazing at the object and simultaneously performing a cognitive action to operate the appliance.

In the Table 2.3, we draw a comparison between our project and Pieter Simoens' works.

| **Project** | **Smart Home Control with Head-Mounted Sensors for Vision and Brain Activity** | **Our project** |
|---|---|---|
| Idea | control appliances through user's gaze and brain activity | Control appliances by using virtual panels or voice commands |
| Ease-of-use | No, since it requires user's concentration for brain activity and hardware setup prior to using its system, e.g., hydrating the Emotiv EEG sensors with a saline solution | Yes, user just wears the Hololens headset, open the system, then prepare to use |
| Controlling features | user's gaze with brain activity | Voice command, gesture |
| Support User Interface | No | Yes |

Table 2.3: Comparison between research paper in controlling a smart home

# Chapter 3

# Smart Home and Augmented Reality

In this chapter, we will explain theories and other knowledges that have been used in this project for development of Microsoft Hololens for home energy management system (HEMS) which mainly are smart home concept, Home Energy Management System (HEMS), and augmented reality on Microsoft Hololens.

## 3.1 Smart Home

Smart home, also known as an automation home, is a future home equipped with a convergence of different technologies incorporated in together and connected wirelessly to a centralized computer called "Home Gateway" as shown in Figure 3-1. Moreover, the home automation aims to facilitate inhabitants with comfort, convenience, and high security. A smart home is managed by an intelligent software, such as HEMS, running on a home gateway. The software operates a smart home to achieve some purpose, such as home energy efficiency, security, and well-living. Also, the software aims to keep an eye on hardware components in the house [2]. However, a home energy management system or HEMS will be elaborated in the next section. The way how user controls the smart home is various, for example, web application, mobile application, or even voice command. However, with the

debut of Hololens, we would like to explore more possibility on how we can utilize this wearable smart glass with HEMS, which will be detailed in next section.

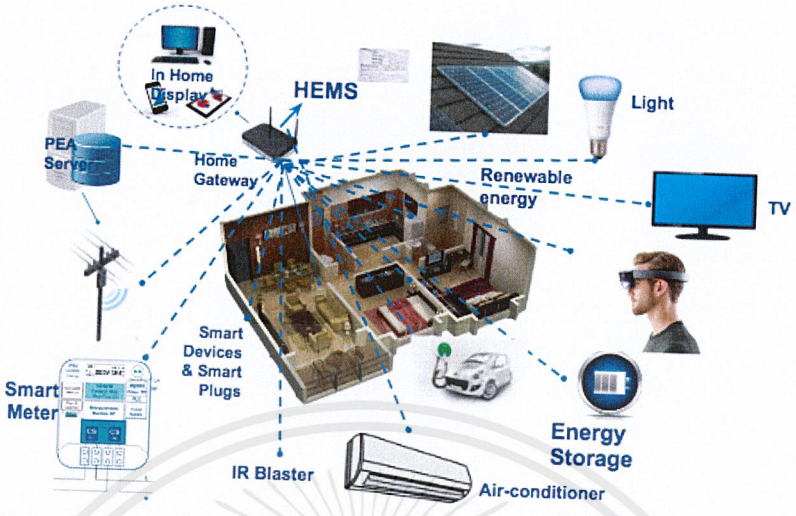



Figure 3-1: A smart home equipped with smart devices that are connected to home gateway.

## 3.2 Home Energy Management System

Home Energy Management System, or HEMS, is a software system to support green technology like home automation (smart home). Its concept is that it monitors and controls all its electrical devices in the smart home in such a way to optimize the electricity usage. To give a better understanding, here is an example scenario. A television is turned on automatically, as well as an air-conditioner is automatically turned on with the appropriate temperature to suit the outside weather and efficiency when a house's owner is coming to the home. When the house's owner gets in, he/she is able to control any appliance using a mobile application. However, no matter how beautiful the smart home feature providing to the users is, an energy consumption of the house, i.e. electricity consumption, is also concerned. Luckily, there is a concept of real-time pricing presented in the smart home. The real-time pricing, in plain text, is an information that gives to customers about the actual cost of electricity usage at any point of time. All real-

time data is from a smart meter, which takes care of sending and receiving info about electricity usage. Electricity prices change from hour-to-hour. The benefits users would get from real-time pricing is that users can adjust their electricity usage in an appropriate manner [1].

## 3.3 Augmented Reality for Microsoft Hololens

Microsoft Hololens is well known as a smart-glasses headset for displaying holographic scene. And it is run on Windows 10 operating system which could be said to be one of the first devices running on the Windows Holographic platform[5]. As for Microsoft Hololens features, it comes with 3 ways to enable users to interact with the content naturally: gaze, gesture, and voice [3].


Figure 3-2: Microsoft Hololens with 3 main features enabling user to interact with content and information in the most natural ways possible [3].

**Gaze** – Allows user to select holograms and to move the hologram. Whenever user his/her head, the cursor on the Hololens screen will follow and track relative position.

**Gesture** – Allows user to select and resize any items in the screen and drag and drop holograms in his/her world.

**Voice command** – Allows user to navigate, select, open, command, and control your application. It also provides a speech-to-text feature.

You can think of scenario like user tries to visualize a 3D home model user just designs and adjust the rotation by gesture, or user is on video call with his/her friend through Skype, or even user can see the updated news and to-do list from a wall by gazing at the wall as shown in the Figure 3-3.


Figure 3-3: An example of how user sees when heads on Hololens [8].

This smart-glasses headset is composed of a number of sensors, high technology visual perception, and custom holographic processing unit which make it go remarkably beyond than a screen. The Figure 3-4 illustrates on some important sensors on the Hololens.

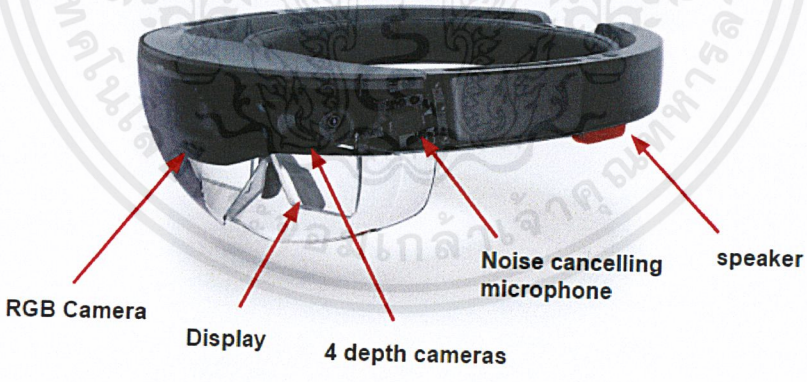



Figure 3-4: Some important sensors on the Hololens.

A description of the sensors in Figure 3-4 is explained below.

**RGB Camera** – To provide the three basic colors which are red, green, and blue.

**Display** – To display holographic scene on the Hololens screen.

**4 Depth cameras** –These cameras help in user's hand tracking and perform surface reconstruction for the displayed screen, where there are 2 cameras on the left and another 2 on the right.

**Noise cancelling microphone** – To receive a voice command from a user.

**Speaker** – To provide stereo sound to user.

# Chapter 4

# Requirement Analysis

## 4.1 Requirement for the application

### 4.1.1 Augmented Reality Interface for HEMS

**Functional requirement**

- User shall be able to view appliance information and control appliances from panels.
- Panel will stays at the same place nearby its appliances even after restarting the application.
- User shall be able to control appliances by using voice command
- When using voice command, current room and nearby appliances will also be sent to HEMS to process the action.
- Hololens send event to HEMS when user enter or leave some room.
- Virtual assistant will follow you wherever you go.
- When using every voice command, there will be a speech response.
- User shall be able to hear speech response from the direction of virtual assistant.

- User shall be able to hide or show panels and virtual assistant.

**Non-Functional requirement**

- The system recognizes the object using Vuforia.
- The system is implemented using unity, c#, Vuforia and API.AI.
- The system can run on Hololens.
- Voice commands supports only English language.
- Datas are updated real-time.

### 4.1.2 Electricity Company Server

**Functional requirement**

- User shall be able to change DR mode.
- User shall be able to broadcast price in real time.
- User shall be able to update flat rate price.
- User shall be able to update TOU price.
- User shall be able to view and set rule for each house.
- User shall be able to view list of household and can view house information.
- User shall be able to create news.

**Non-Functional requirement**

- The system is implemented using HTML, CSS, JS, Django, WebSocket and PostgreSQL.
- The web can be run on Microsoft Edge, Google Chrome, Apple Safari and Mozilla Firefox.
- The web is responsive.

### 4.1.3 Smart Home Web Application

**Functional requirement**

- User shall be able to view daily, weekly, monthly, yearly usage histories as charts.
- User shall be able to view date, time, temperature in Celsius and humidity in percentage.
- User shall be able to view voltage, battery, price per unit and energy mode of smart home.
- User shall be able to view daily, weekly, monthly, yearly saving histories as charts.
- User shall be able to view all appliances and can control each appliance.
- User shall be able to view smart home 3d model and can export or import in JSON format.
- User shall be able to view total KWH, power factor, total kvar, current and voltage histories as charts and user be able to download as excel, PDF and text file.
- User shall be able to view news from Electricity Company server.
- User shall be able to view rule simulation of smart home.
- User shall be able to view, add, edit or delete rule.
- User shall be able to view and select options to saving energy.
- User shall be able to view prediction of energy consumption.
- User shall be able to view, add, edit or delete preference.
- User shall be able to login or logout.

**Non-Functional requirement**

- The system is implemented using HTML, CSS, JS, React, Django, WebSocket and WebGL.
- The web can be run on Microsoft Edge, Google Chrome, Apple Safari and Mozilla Firefox.
- The web is responsive and real-time.
- The web is easy to use.

## 4.2 Use Case Diagram

This section will show user's functionalities in use-case diagrams, where there are 3 diagrams: Hololens application, electricity company server, and smart home web application. As for use case descriptions, readers can see more in Appendix A.

### 4.2.1 Electricity Company Server

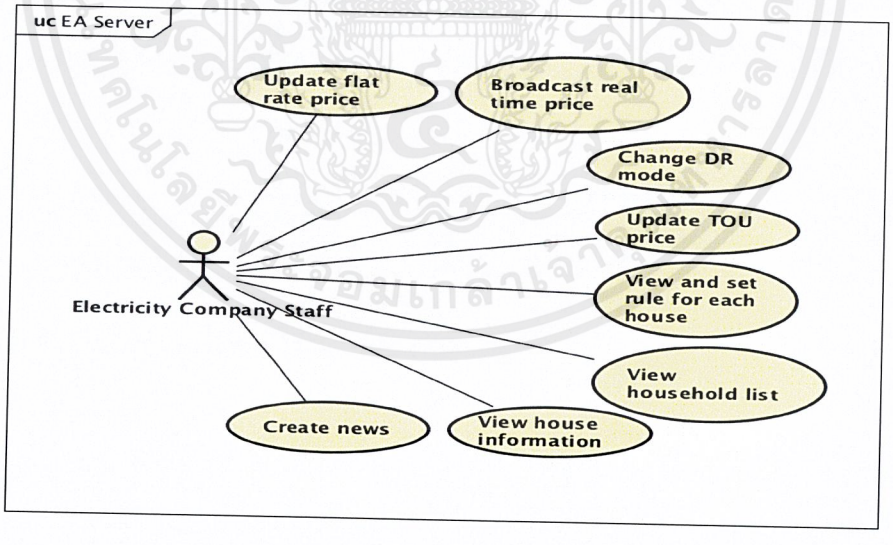



Figure 4-1: Electricity Company Server use case diagram

### 4.2.2 Hololens Application

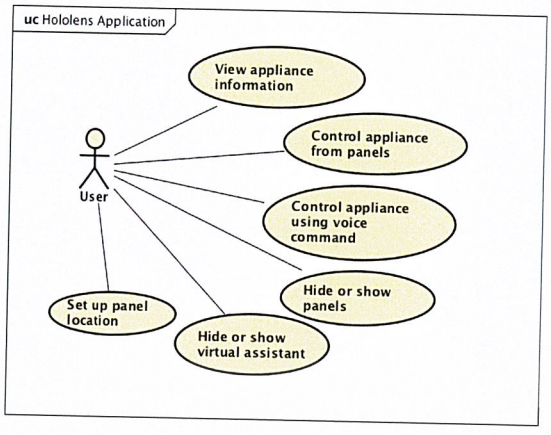



Figure 4-2: Hololens application use case diagram

### 4.2.3 Smart Home Web Application

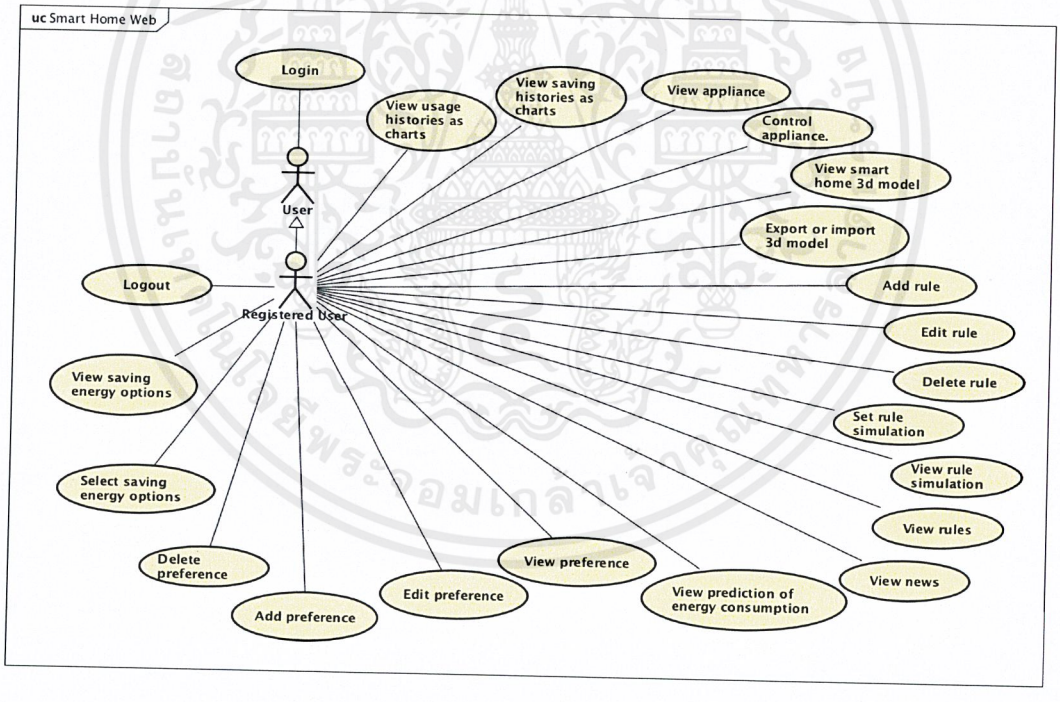



Figure 4-3: Smart home web application use case diagram

# Chapter 5

# System Design

This chapter provides an overview of Hololens for Home Energy Management System system design which is designed according to the requirements specification in the Chapter 4. The design depicts the system in a detailed manner ranged from the basis for implementation to the flow of system functions along with components that are used to construct the system.

## 5.1 System Architecture

The Figure 5-1 illustrates an overview of our system of the project to make readers clearly understand how the project works, and how each component in this system collaborate with one another. As the Figure 5-1 depicted, the system mainly consists of three components: a list of front-end applications as in the dashed-line box, HEMS as in centered-box , and a group of appliances. Basically, HEMS is an existing software which we utilize its essential features. Also, the list of red boxes in the dashed-line box is the application implemented by us. This is to make visually clear in each component. In this thesis, Home Energy Management System (HEMS) plays a key role in the whole system to make every other component achieves its own task. HEMS acts as the home gateway to receive necessary data from other front-end applications by communicating via RESTful protocol and WebSocket, while ZigBee, WiFi and Bluetooth are used for communication with

household appliances. And it responds those received data by displaying via front-end applications such as a smart home website or Hololens application. Moreover, HEMS can control and monitor those devices by, for example, turning on a light or setting rules for specific device what to do when it is timely night. However, all the above components as shown in the Figure 5-1 have their own system design, and we will go into details in the next section.

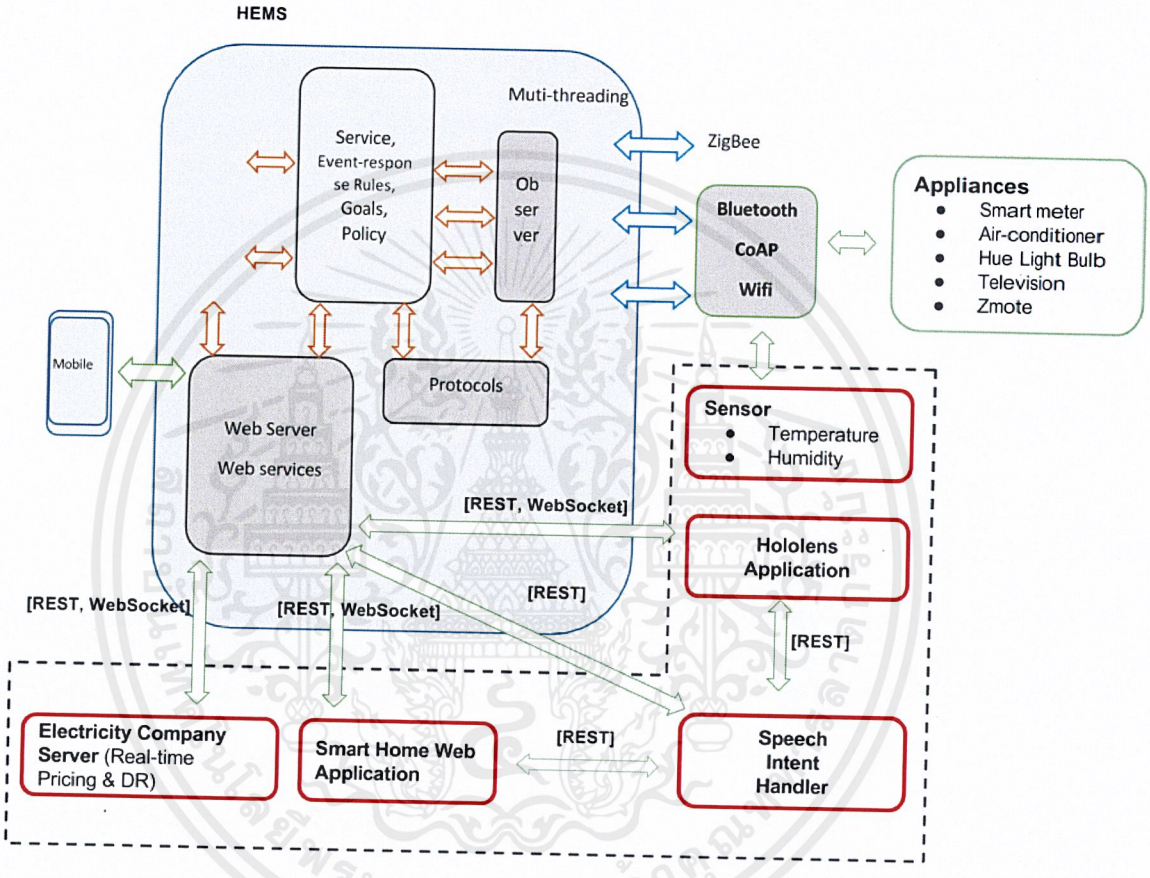



Figure 5-1: The system overview of the project

## 5.2 System Components

As Figure 5-1 shown in previous section, there are mainly six items in the system which will be elaborated in details in this section.

### 5.2.1 Home Energy Management System (HEMS)

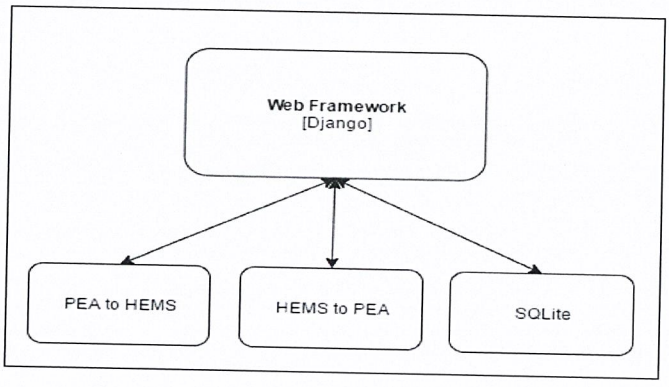



Figure 5-2: The system overview of the HEMS

According to the system's design overview in Figure 5-2, it refers to how inside HEMS is constructed. HEMS is a server-side that handles requests and response of other software. So, web framework and database are much required to achieve the server-side. Regarding the web framework, we use Django to implement as back-end, because it is one of the most common framework and its supportability for other technologies, such as REST API, web service, which are used to access resources on the server. Another part in HEMS is the SQLite. This relational database is used to store relevant data securely and to deal with workloads from other software applications' requests and responses. EA_Snooper or electricity company snooper is a module that sends home data to electricity company server. EA_Receiver is a module that listens for updates from electricity company server.

### 5.2.2 Hololens Application

The Figure 5-3 depicts a design of the Hololens application system design. In addition, this client-side application comprises of mainly four components included in the design to make the whole application run completely: Hololens application, API.AI, HEMS, and a group of devices. The augmented reality, speech recognition, location service and gesture control embedded in the Hololens are used to serve a better experience to users to feel and have fun with it. API.AI is another platform to perform the natural language processing. API.AI also provides

natural language voice web service for developers to easily build and test their conversation scenarios. So, we choose API.AI platform to support natural language understanding which helps our system to understand an intent of user's speech. By designing conversational scenario and corresponding action in API.AI for the smart home, it helps us to manage and analyze interactions with users easily. As for HEMS, it is a medium to receive a command from Hololens and data from appliance to send the command to home appliance or to respond back to Hololens application.

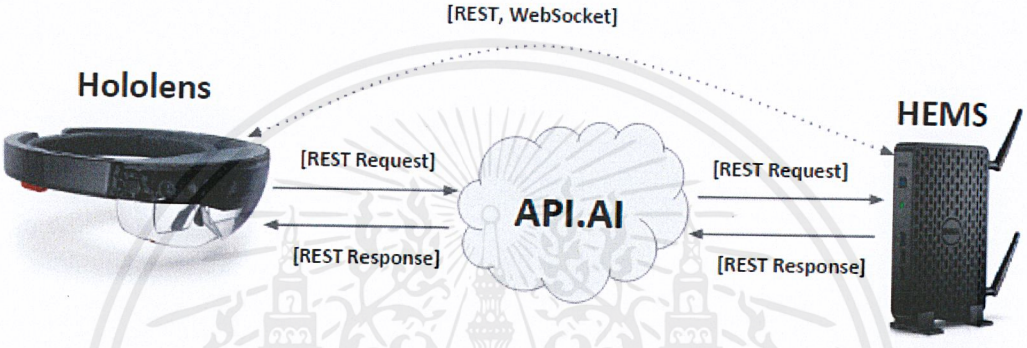



Figure 5-3: The system overview of the Hololens application

### 5.2.3 Smart Home Web Application

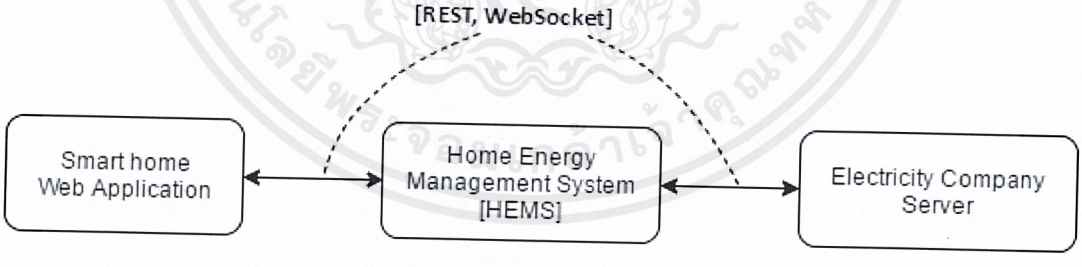



Figure 5-4: The system overview of the smart home web application

The smart home web application design, as shown in Figure 5-4, has almost the same design as Hololens application, except that this web application is no need additional instance in between itself and HEMS. So, it can communicate directly to HEMS by using REST and WebSocket protocol. Moreover, when there is any

news updates or notification from electricity company, i.e. new electricity rate, users can instantly receive the notification. In this front-end website, it contains several technology being employed, such as, real-time web technology, responsiveness, which will be described in detail how they are used in the application in the Chapter 6. Note that, basically, the site was created to serve as web-based application an also an alternative mode for Hololens' user interface depending on circumstances.

### 5.2.4 Electricity Company Server

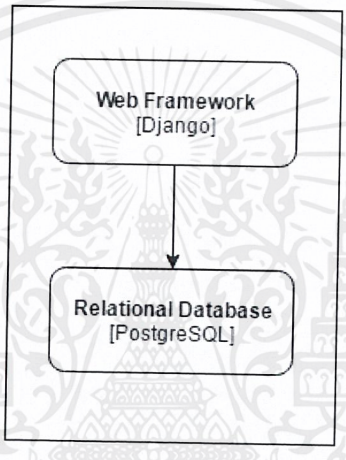



Figure 5-5: The system overview of the Electricity Company Server

The Electricity Company Server design, as depicted in Figure 5-5, has resemblance to HEMS design. It used PostgreSQL Object-relational database as to cope with numerous requests and responses. However, its task is different from HEMS. This server is under controlled of electricity company admins to keep updating all news and updates and send those information to HEMS.

## 5.3 Component Diagram

In this section, it shows all required components to create the whole system. The blue box color refers to a software system, while the sky-blue boxes in the dashed box correspond to a number of components to make up another software system.

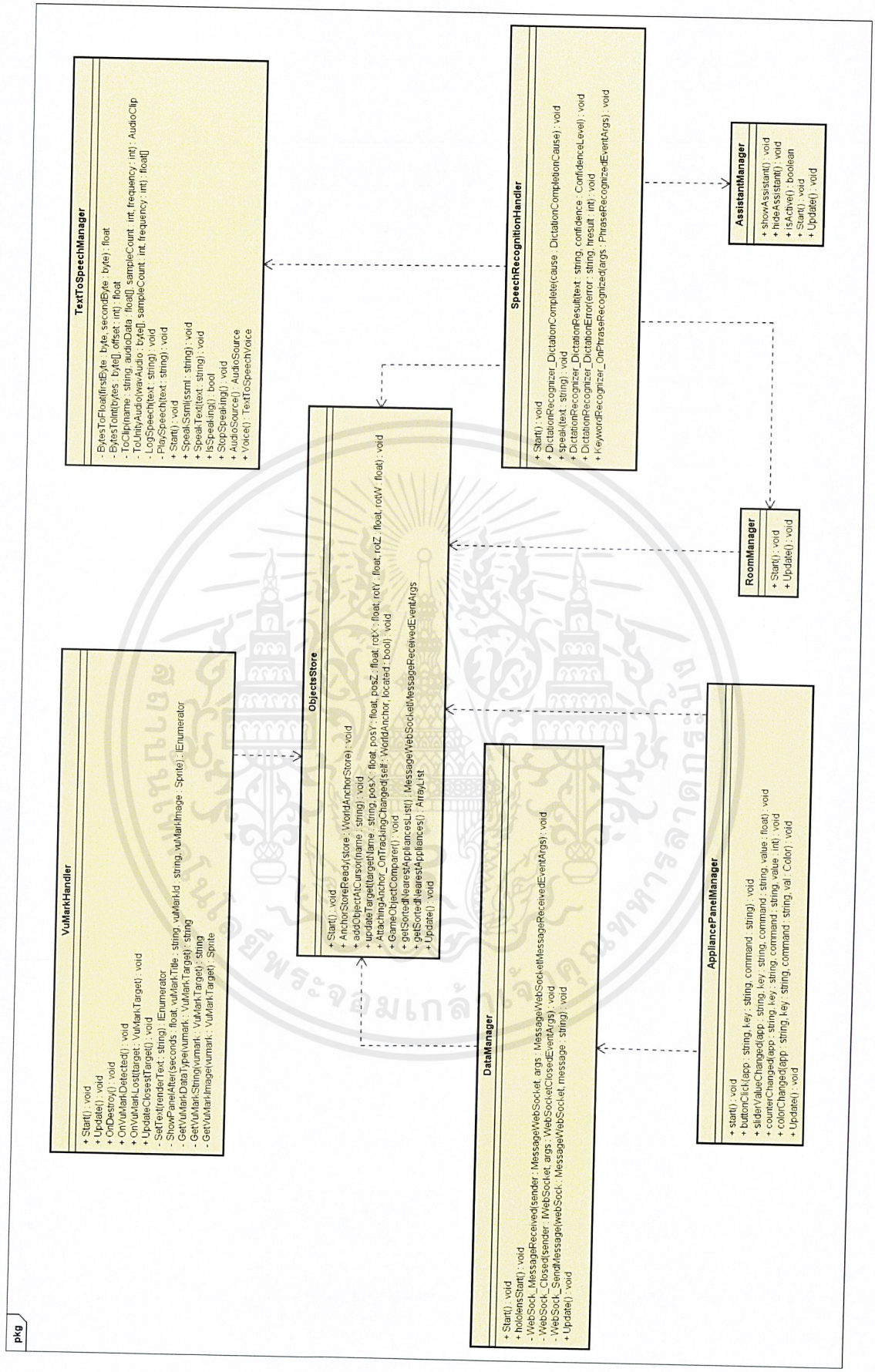



Figure 5-6: The component diagram

# Chapter 6

# System Development

This chapter amplifies the development processes of the project. Also, it includes techniques to be implemented for each component in the system including tools and resources that involve in the development process. However, This chapter is divided into four parts: Home Energy Management System (HEMS) Development, Smart Home Web Application Development, Electricity Company Server Development and Microsoft Hololens Application Development

## 6.1 Home Energy Management System (HEMS) Development

As said earlier in the Chapter 3, Home Energy Management System or HEMS is responsible for monitoring and controlling all electrical devices in a smart home in such a way to optimise the electricity usage. Also, HEMS, as mentioned in Section 5.2 in Chapter 5, characterizes itself as an server-side. With respect to HEMS in the development process, HEMS has undergone several changes in the web application because of new discovered obstacles and limitations. This section contains the detail of each implementation and development.

### 6.1.1 Real-time Web Application

HEMS real-time update allows user to receive up-to-date information of home appliances from the smart home application which helps users to follow all changes happened in their house or in Electricity Company, such as news or electricity rate. In order to carry out the real-time web application, WebSocket technology is taken to serve the application's requirement. In addition, there is a 3D house model created in the site via WebVR technology to support user's experience in real-time house appliances' status. For example, if an air-conditioner is turned on, user will see the air-conditioner with softly wind signifying the AC status, or a light-bulb with some color implying the bulb is turned on. However, the result of the house model will be drawn in next Chapter.

During the implementation, WebSocket protocol is installed on HEMS to run WebSocket service which make HEMS able to serve data in real-time. The process starts with a client sending a regular HTTP request to HEMS, where there is a specific header included in the request. This specific header will inform HEMS that the client wants to create a WebSocket connection. HEMS then agree on the header and will communicate this through the header in the response. This makes the initial HTTP connection get replaced by WebSocket connection Hence, when there is a new data in HEMS, WebSocket service will push the new data to clients whose a connection's header stays the same as initialized.

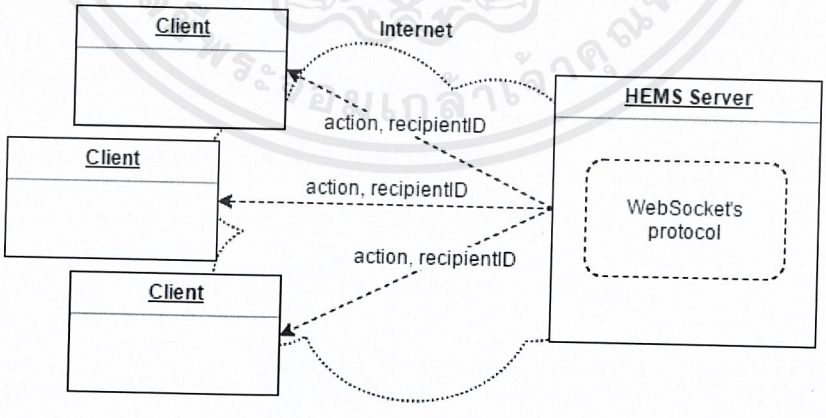



Figure 6-1: How WebSocket works with HEMS-connected clients

However, WebSocket only provides us a way to communicate via a socket, the use of WebSocket can be in many ways. We use Django Channel as a WebSocket handler, and it provides us a concept of groups or rooms. Groups or rooms is similar to publish and subscribe pattern. In Django Channel, you have to execute a code to subscribe or unsubscribe user to groups. After that, we can publish message to preferred group. We have implemented that if user sends a message to join or leave group, we will execute the code to do so. Certain changes inside the system will trigger the code that publish messages to a certain group. The Table 6.4, Table 6.1, Table 6.2 and Table 6.3 shows the JSON data format of WebSocket message used in HEMS system for each group.

| Name | Type |
|---|---|
| appliances | |
| consumption | float |
| id | int |
| name | string |
| current | float |
| sys_kVarf | float |
| sys_kVarr | float |
| sys_kWf | float |
| sys_kWr | float |
| volt | float |

Table 6.1: Group : meter.info

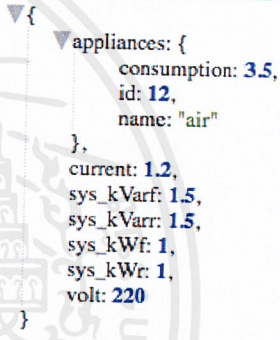



Figure 6-2: Example Group : meter.info

| Name | Type |
|---|---|
| body | string |

Table 6.2: Group : alert

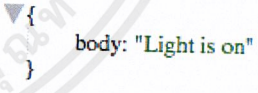



Figure 6-3: Example Group : alert

| Name | Type |
|---|---|
| parameter | string |
| value | int |

Table 6.3: Group : appliance.status

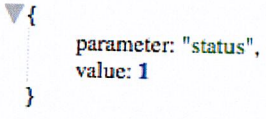



Figure 6-4: Example Group : appliance.status

| Name | Type |
|---|---|
| alt_inputs | |
| current | float |
| kW | int |
| volt | int |
| battery | |
| current | float |
| kW | int |
| volt | int |
| line_input | |
| current | float |
| kW | int |
| volt | int |
| output | |
| current | float |
| kW | int |
| volt | int |
| solar | |
| kW | int |
| use mode | |
| wind turbine | |
| kW | int |

Table 6.4: Group : inverter

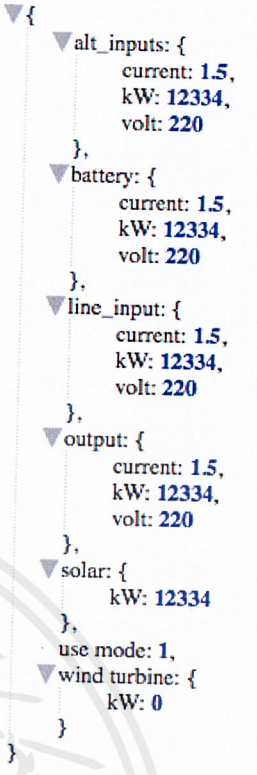



Figure 6-5: Example Group :inverter

### 6.1.2 HEMS Database Model

#### Overview

```
class Floor(models.Model):
    name = models.TextField()

class ApplianceTypes(models.Model):
    name = models.TextField()
    commands = models.TextField()
    on_read = models.TextField()

class Appliance(models.Model):
    name = models.TextField()
    app_type = models.IntegerField()
```

```
12      status = models.IntegerField()
13      room_id = models.IntegerField()
14      current_consumption = models.IntegerField()
15      current_state = models.IntegerField()
16      states = models.TextField()
17      data = models.TextField()
18      type = models.ForeignKey(ApplianceType)
19
20  class Room(models.Model):
21      name = models.TextField()
22      floor_id = models.IntegerField()
23      appliances = models.ManyToManyField(Appliance)
24
25  class Rule(models.Model):
26      conditions = models.TextField()
27      action = models.TextField()
28
29  class Preference(models.Model):
30      conditions = models.TextField()
31      prefer = models.TextField()
```

Listing 6.1: Implementation of HEMS ER Diagram in Python

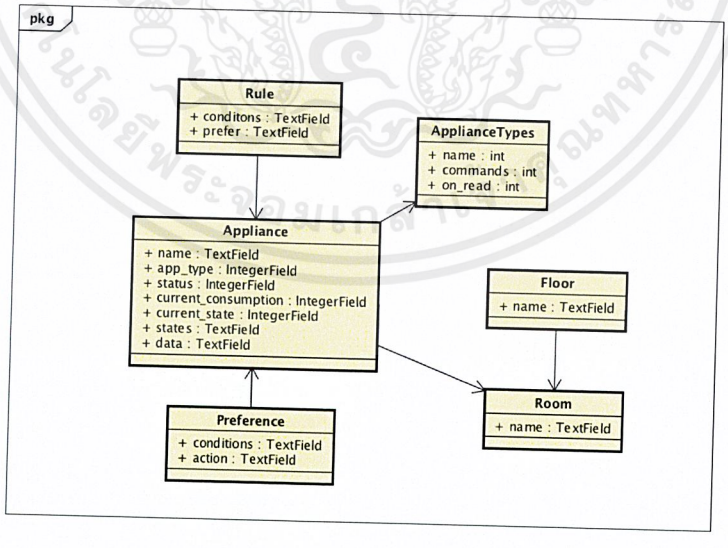



Figure 6-6: Database Model Class Diagram

Commands in appliance type stores list of attributes that an appliance type may have. It is represented in json format. The data consist of value, name and cmd for each attribute, if cmd is null, it means that that attribute is a read only attribute. The example of read-only attribute is consumption as you can not change the consumption of appliance directly. If the attribute is not read-only, it will contain more information such as minimum value, maximum value and value step. It also tells which type of control should be displayed in the UI such as toogle, slider, color picker or step. There is also data that tells which voice commands are mapped with which attribute and value to be changed to, when the voice command are matched.

```
[
    {
        "cmd":{
            "control_type":"toggle",
            "description":"Turn on or off {name}",
            "function":
                "if setTo == 1:
                    self.setData('last_command_result', self.emitZmote(
self.getData('setting.zmote_name'), self.getData('setting.
zmote_action_power_on')))
                else:
                    self.setData('last_command_result', self.emitZmote(
self.getData('setting.zmote_name'), self.getData('setting.
zmote_action_power_off'))).status = setTo.save()",
                "name":"set_On_Off",
                "parameters":[
                    {
                        "name":"setTo",
                        "type":"integer"
                    }
                ],
                "voice_commands":[
                    {
                        "cmd":"smarthome.device.switch.on",
```

```
21                          "value":"1"
22                      },
23                      {
24                          "cmd":"smarthome.device.switch.off",
25                          "value":"0"
26                      }
27                  ]
28          },
29          "max_value":1,
30          "min_value":0,
31          "name":"status",
32          "value":"value['status']",
33          "value_step":1
34      }
35  ]
```

Listing 6.2: Command's mapping

### 6.1.3 HEMS API

HEMS API is a RESTful web-service inside HEMS, and it is implemented to be the resources of other software to request. This resources are accessed by a common interface using HTTP Protocol for data communication.

As for the development of HEMS API, it needs to ensure that all required appliances have their own medium to communicate between them and HEMS, so that HEMS can request and respond to the appliances correctly, such as Bluetooth, WiFi, or ZigBee. So, whenever there is a request from other software, HEMS will receive the request and map the request with its methods. Then HEMS will call that methods to get data from the target appliance. Later on, the received data will be sent back as a response message to the caller.

However, before user can request HEMS to get data from the target, there should be a list of methods provided by RESTful web-service in order to function the request. There are several services available on the server for the application

to utilize. By using RESTful API, the request may vary by URL, HTTP Verb and data. The list of the services and their description are described in Table 6.5

Table 6.5: HEMS REST API

| **URL** | **HTTP Verb** | **Description** |
|---|---|---|
| api/preferences/ | GET | Return the current preferences |
| api/preferences/{id}/ | PUT | Update preference that has id={id} |
| api/preferences/ | POST | Create new preferences |
| api/preferences/{id}/ | DELETE | Delete the preference that has id={id} |
| api/rules/ | GET | Return all rules |
| api/rules/{id}// | PUT | Update rules that has id={id}/ |
| api/rules/ | POST | Create new rules |
| api/rules/{id}/ | DELETE | Delete rule that has id={id}/ |
| api/appliances/ | GET | Get a list of appliances |
| api/appliances/{id}/ | PUT | Update appliance that has id = {id} |
| api/appliances/ | POST | Create a new appliance |
| api/appliances/{id}/ | DELETE | Delete appliance that has id = {id} |
| api/appliances/{id}/info/ | GET | Get appliance more detailed info that has id = {id} |
| api/appliances/{id}/command | POST | Command appliance that has id = {id} |

## 6.2 Smart Home Web Application Development

### 6.2.1 Responsive Web Interface

The web application is implemented to provide an alternative choice for users to access. Since users are capable of accessing the web application through many ways, i.e. personal computer, tablet, or mobile phone. So, creating responsive web interface for all kinds of devices should be also become aware of. In order to make the web interface become responsive, several web development frameworks are taken up with. In this process, we chose the popular web design framework which was Bootstrap. Not only is it free, but it is also open-source with numerous themes for the beauty of components. In order to make screen responsive, grid system should be exploited. The grid system is a structure that helps in organizing and arranging page layout in a series of rows and columns. By applying the grid classes (e.g. .row, .col) with properly specific size of number (e.g. .col-xs-4 is set to 4 columns when screen is extra-small) in the web page, we can have all responsive web page which support all various screen size. For example, Figure 6-9 and Figure 6-7 depict different types of home page from two devices accessing the smart home web application, where the first and second Figure are personal computer screen and iPhone, respectively. While the Figure 6-8 is navigation bar of the website in iPhone.

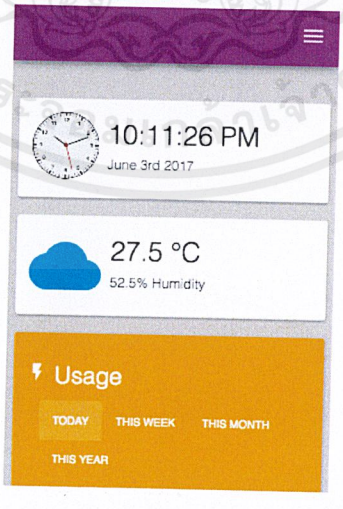



Figure 6-7: Home page from iPhone

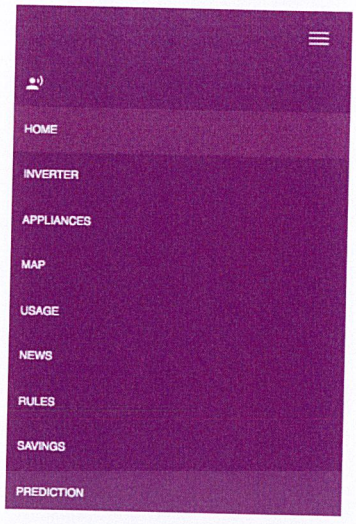



Figure 6-8: Navigation bar from iPhone

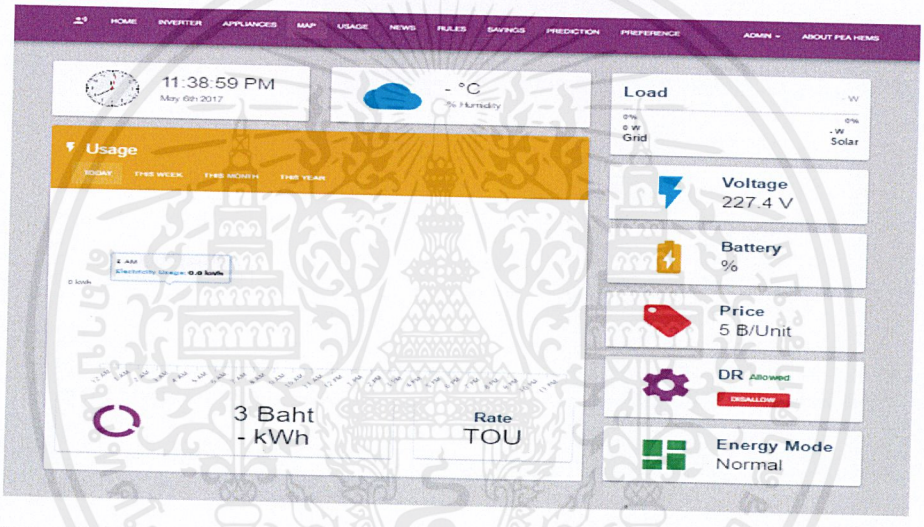



Figure 6-9: User interface of smart home web application from Personal Computer

## 6.3 Electricity Company Server Development

Electricity Company Server acts as the main server to keep notifying any updates towards other smart homes. In the development of Electricity Company Server, we have also created services for the server to let other servers, e.g., HEMS, use. The list of the services and their description are described in Table 6.6

Table 6.6: Electricity Company Server REST API

| **URL** | **HTTP Verb** | **Description** |
|---|---|---|
| api/news/ | GET | Return all news |
| api/news/{id}/ | PUT | Update news that has id={id} |
| api/news/ | POST | Create new news |
| api/news/{id}/ | DELETE | Delete news that has id={id} |
| api/households/ | GET | Get a list of households |
| api/households/{id}/ | PUT | Update household that has id = {id} |
| api/households/ | POST | Create a new household |
| api/households/{id}/ | DELETE | Delete household that has id = {id} |
| api/settings/ | GET | Get all settings |
| api/settings/{id}/ | PUT | Update setting that has id = {id} |
| api/settings/ | POST | Create a new setting |
| api/settings/{id}/ | DELETE | Delete setting that has id = {id} |
| api/settingsKv/ | GET | Get all settings as JSON class |

There is also a web page for Electricity Company to let admin set new information, and monitor current setting, such as current electricity price rate. The Figure 6-10, Figure 6-11, and Figure 6-12 show a home page, a home's list page, and news' page of the site, respectively.

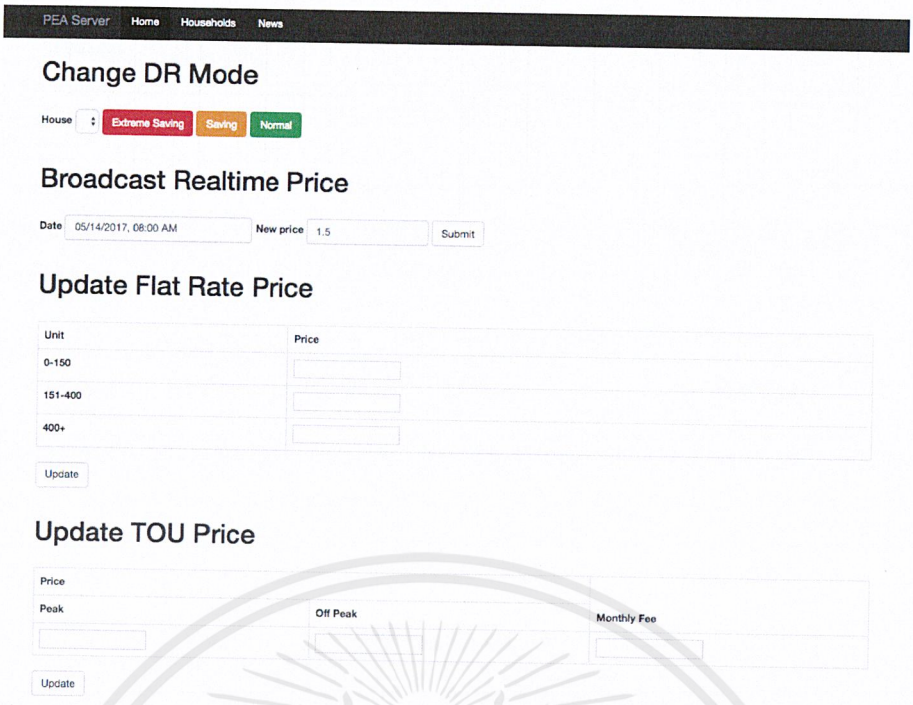



Figure 6-10: The home page of virtual-cities.net for Electricity Company Server

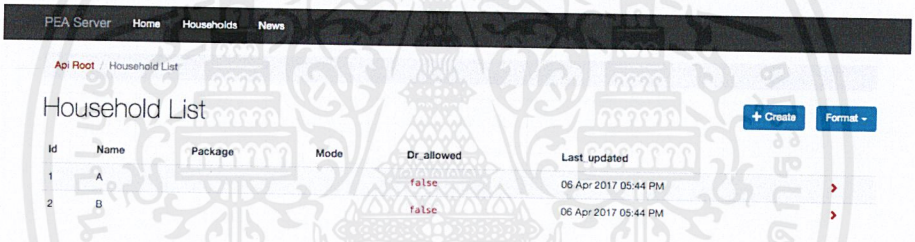



Figure 6-11: The households page of virtual-cities.net for Electricity Company Server

## 6.4 Microsoft Hololens Application Development

The Microsoft Hololens application is proposed to help residents who live in a smart home to have an access to the home as natural as possible by using their voice command to control or monitor home's appliances. It is important to let the Hololens get relevant data from home's appliances via HEMS, so that user is able to view those information. The development of the application utilizes a set of functions provided by HEMS to display on the Hololens' screen. During the application development, the app had undergone several problems and limitations regards to the Hololens' capability. In this section, it shows the detail of each

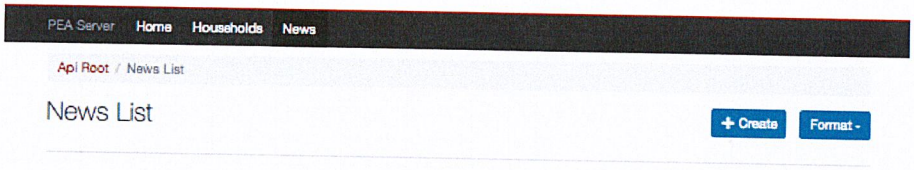



Figure 6-12: The news page of virtual-cities.net for Electricity Company Server

implementation to resolve the problems and limitations.

### 6.4.1 3D Modelling

The model of personal assistant is designed in Blender software and then exported to Unity. The Figure 6-13 is an example of designing 3D model in Unity.

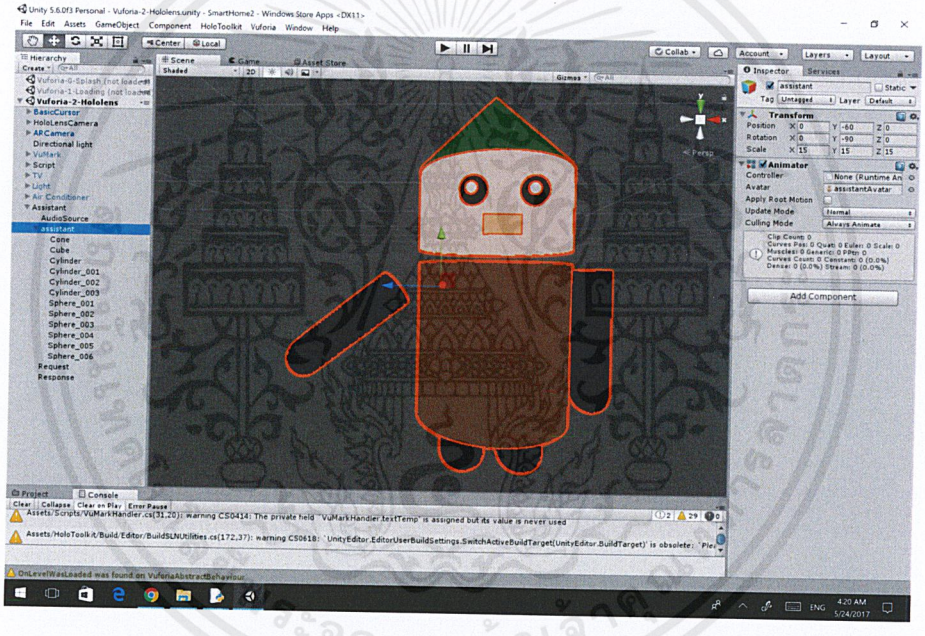


Figure 6-13: An example of designing 3D model in Unity

### 6.4.2 User Interface Design and Visualization

The design of appliance panels and other panels are designed in Adobe Illustrator and are exported as multiple images which are used to render in Unity.

### 6.4.3 Voice Interface

As for the implementation of voice command for the application, firstly we take in voice input from user's speech from Hololens or web application. For Hololens, we use Windows SDK to get text representation of user's speech. For web application, we use Webkit speech API to get text representation of user's speech. However, the user's speech cannot be understood easily by software and is difficult to grab the user's intent. So we use API.AI cloud platform to help in natural language processing. The API.AI offers combination of rule based matching and machine learning which we can take an advantage of. The text representation of user's speech will be sent to API.AI to be analyzed and identified the user's intent from the speech. In the API.AI, there will be a list of entities which extract parameter values from natural language inputs, and intents of user's voice command that will map between what the user says and what action should be taken. In the Figure 6-14 is a sample list of device entities that we created for appliance where the rectangle is the main entity of the appliance while the ovals are entity's synonyms. This allows user to speak the specific device with more than one type. For example, user can speak AC, air-con, or air-conditioner (in the oval shape) to refer to air-conditioner in user's home which can be justified as the AC device's entity in the rectangle. The interface we use to define device entities is shown in the Figure 6-15. The intent is like a set of rules defining what user can says to do certain things. Defining intent in API.AI is done in a web interface shown in Figure 6-16.

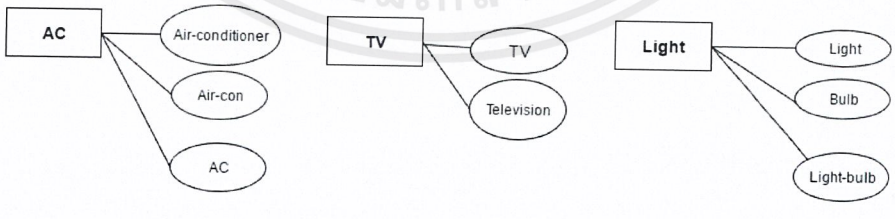



Figure 6-14: Example of device entities of home's appliances

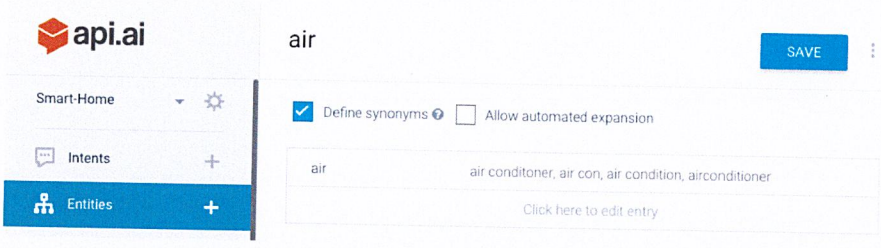



Figure 6-15: Defining device entities of home's appliances

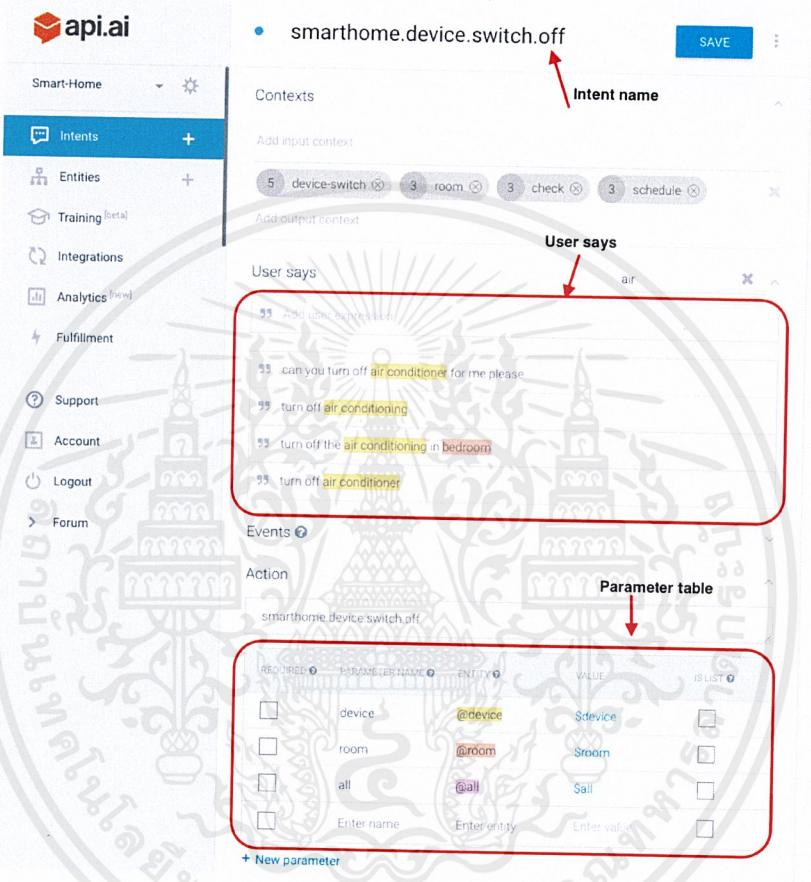



Figure 6-16: Defining user intents

When voice command is sent to API.AI as depicted in Figure 6-17, it will be parsed by machine learning by API.AI web service to query users' intents that is likely to be corresponded with the input. Once API.AI gets the user's intent, it will send the result to HEMS as JSON format as shown in the Figure 6-18. Later on, HEMS will extract the result into JSON format into its understandable result. Once HEMS knows the intent from the JSON file after extracting, it will prompt the target to behave in accordance with the user's intention. Note that if

the system could not recognize any device from user's speech, for example, user said only 'turn on', the system will automatically apply the user's intent command with the user's closest device by checking position of each appliance in a room.

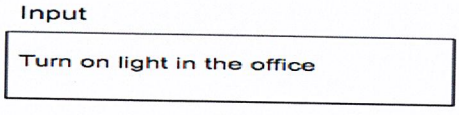



Figure 6-17: Example of input to API.AI

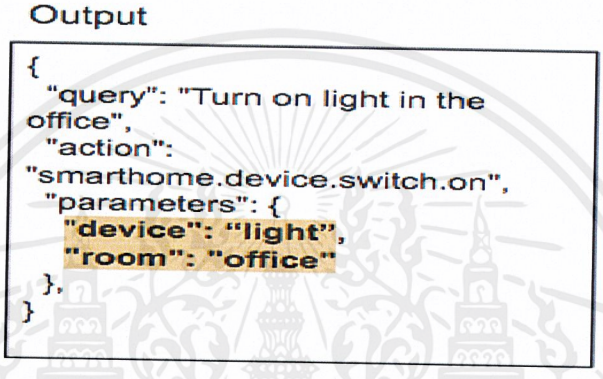



Figure 6-18: Example of output from API.AI

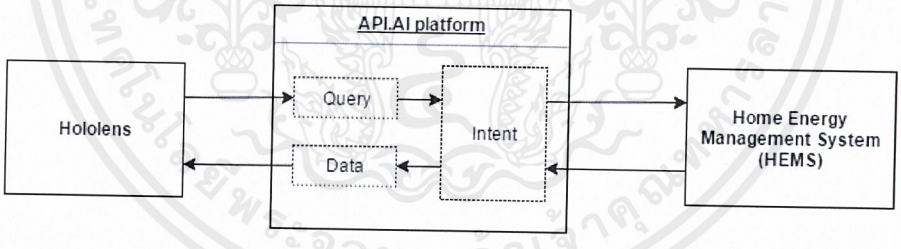



Figure 6-19: The overview of voice command in Hololens application for HEMS

In the Figure 6-19, it shows how the voice interface works where Hololens gets an input, then send the input to API.AI to query the user's intent. Later on, once the intent is identified, it is sent in JSON format to HEMS to have the target complied with the user's intent.

### 6.4.4 Location Mapping

As for the location mapping, we used VuMark- a new kind of bar code that is free for a customized design while encoding data and acting as an AR target at the same time- as a reference point of any appliances. The Figure 6-20 is some example of VuMark bar code. The reason we use VuMark is because we can design our own logo, and it can store various type of data. Basically, We use this bar code to map an appliance to identify the targeted object from user's gazing. Hence, every HEMS-connected device will have its own unique VuMark bar code ID with encoded data. Some example of VuMark in our project is represented in the Figure 6-21 and example of applying VuMark with home appliance is shown in Figure 6-22b. The data stored in VuMark is an array of characters. The first letter will specify wether the VuMark represent a wall or appliance's control panel. The following characters will represent the ID of the object.

Figure 6-20: Examples of VuMark bar code

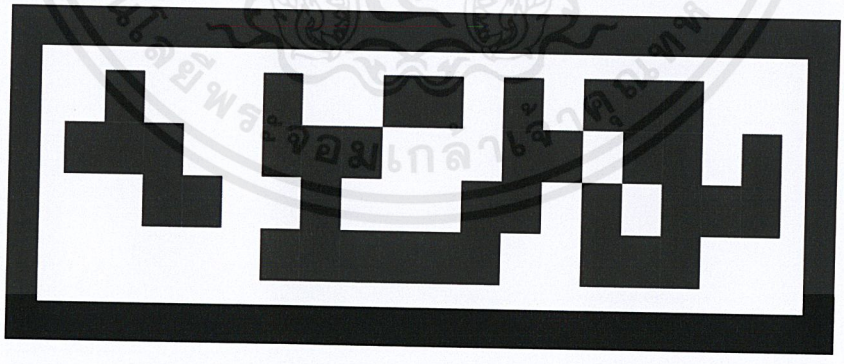
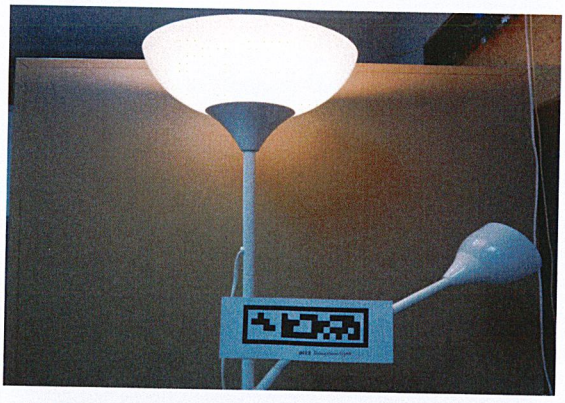


Figure 6-21: Examples of some VuMark designed for home appliance



(a) Examples of applying VuMark with a light

(b) Examples of applying VuMark with a television.

After we get the targeted object from user's gazing, we will save the position of the object with its coordinate, metadata from spatial mapping for further calculation, such as calculating the nearest object from user's position in a room. Then we can show appliance control panel where the VuMark are recognized.

# Chapter 7

# Experiments and Results

In this chapter, we go through our project experimentation of each functionality of the software we developed and display its result with some explanation.

## 7.1 Web Application

The web application works well on every function according to our requirement. Moreover, real-time request and response can be seen instantly when user issues a command to the web. Setting up a rule for a specific appliance also works. However, user-interface may not be easy enough for the user to use.

Figure 7-1 shows some screenshots of home page of the web application. The main page will be displayed once users has logged into the web. The main purpose of this page is to show all basic parameters concerning energy usage which users should firstly know. For example, it displays electricity usage in daily mode. Also, users can select other modes of electricity usage to display; such as weekly, monthly, or annually. See figure 7-1 for more details.

1. Electricity usage and its graph representation
2. Current local time, temperature and humidity in the house
3. Current load

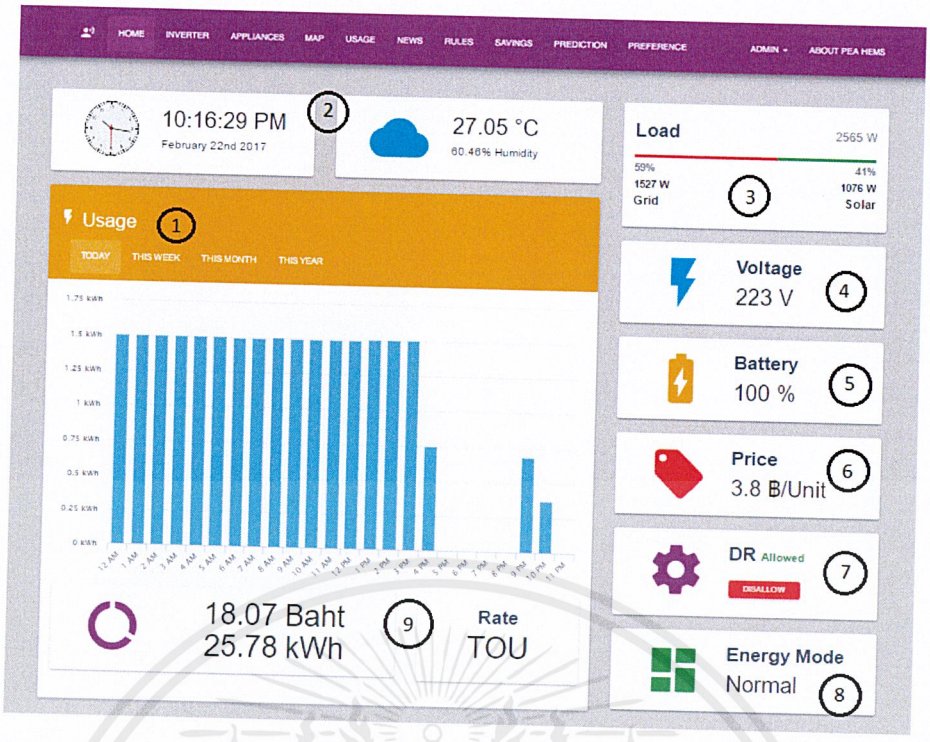
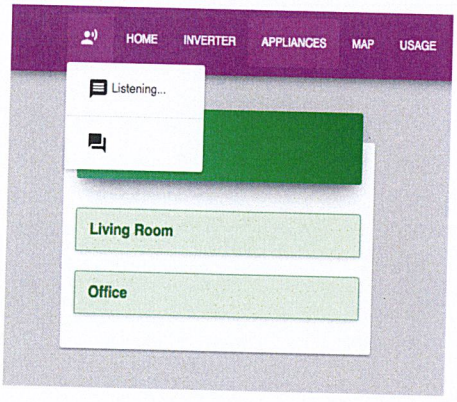



Figure 7-1: Home page

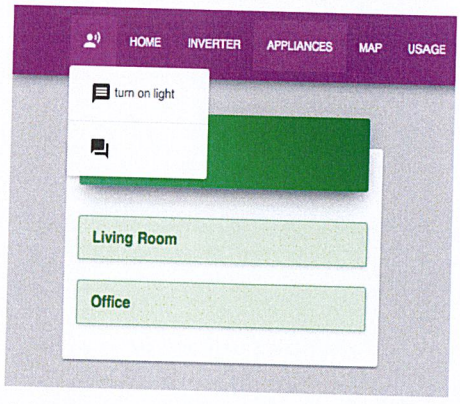
4. Current voltage
5. Current battery remaining
6. Current price rate per unit
7. Current DR mode
8. Current energy mode
9. Current total price and rate type

The web site also supports a voice command to let a user control house appliances using his/her voice command instead of controlling directly on the appliance itself. User will be prompted to speak a command after clicking voice command button as depicted in Figure ??.

For more screenshots on web application, see in Appendix B.




(a) User speaks a voice command to the web




(b) The system responds the user's action in both text and voice

Moreover, the smart home web application can be supplementary to Hololens in terms of control panel. This allows user to be able to control his/her smart home as an alternative. In Figure 7-3 is an example on how the web site can be used as a control panel on Hololens.

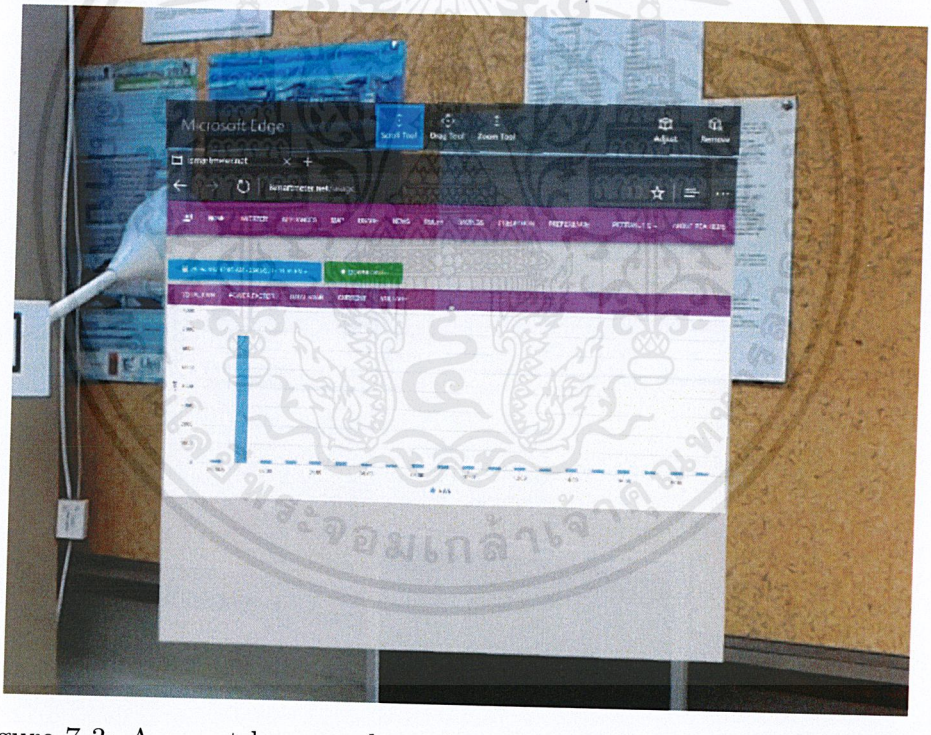



Figure 7-3: A smart home web application as control panel on Hololens

## 7.2 Microsoft Hololens

### 7.2.1 Voice Commands

Voice command is one of the main features to make a home become smart. Basically, user just speaks a command to let HEMS-connected device work, so that the target device behaves accordingly to the command. Some basic of voice commands are to turn-on and turn-off a specific light bulb. Or user can say change specific light color to yellow. After saying the command, the system will respond back whether or not the command is successfully completed, as illustrated in Figure 7-4.

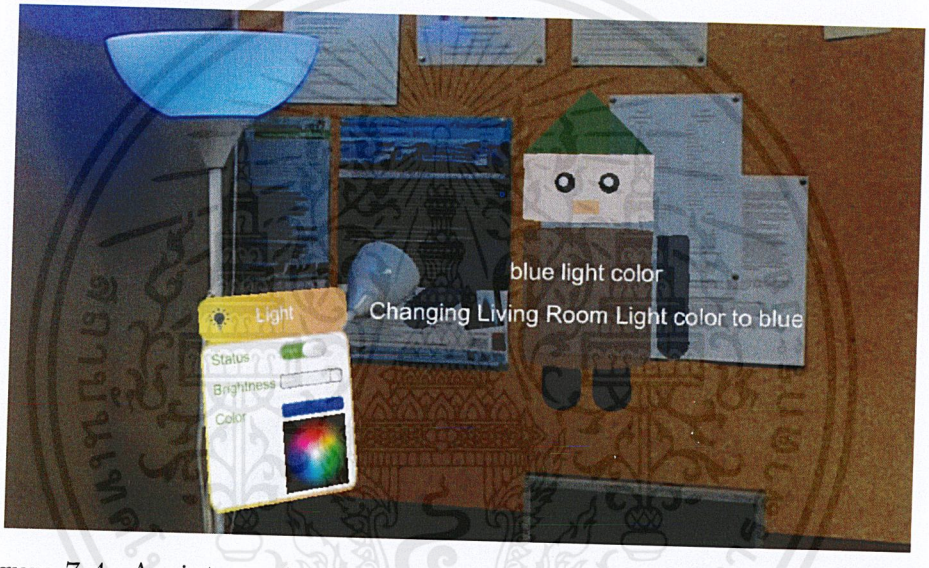



Figure 7-4: An interaction between what user said and system response

### 7.2.2 3D Interface

3D interface has a goal to display another step of user interface by relying on the technology of augmented reality. Since the Hololens is the main tool in this project which supports the AR technology. Hence, we merged the 3D modelling of objects into the screen of Hololens. This will give users another aspect of user interface sight. After applying the 3D modelling into the Hololens' screen, users shall be able to see the virtual object as if they were real. In this project, when

user gazes the HEMS-connected object, there will be a 3D setting panel to let user control the object directly by gesture or voice command as shown in Figure 7-6.

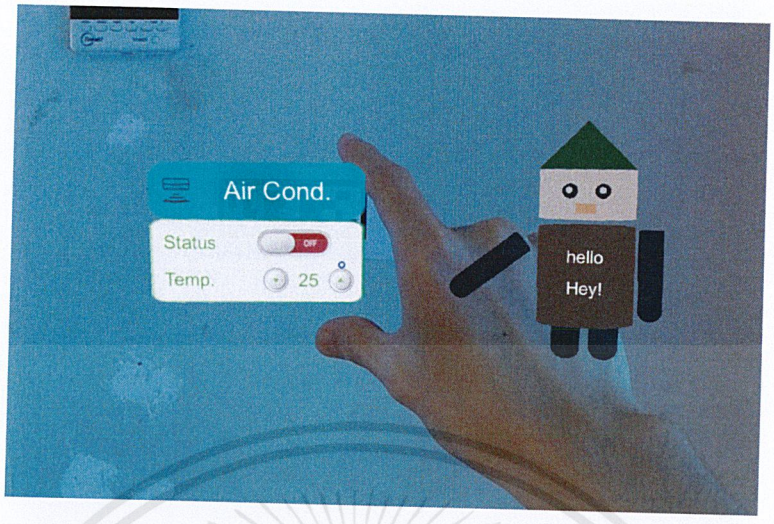



Figure 7-5: A panel is displayed at a VuMark position when user looks at the VuMark.

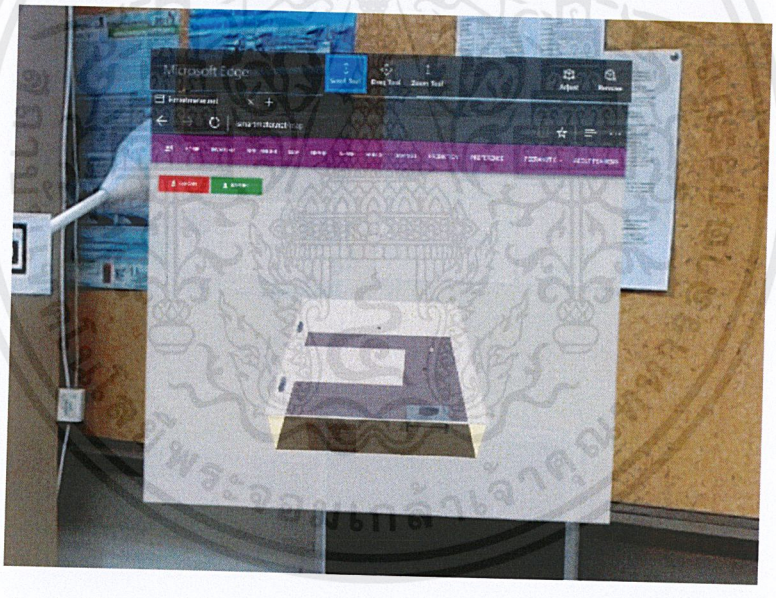


Figure 7-6: A 3D model from a smart home web application on Hololens

# Chapter 8

# Conclusion

## 8.1 Summary

In this project, we aim to create an application using Microsoft Hololens as a 3D user-interface for Home Energy Management System (HEMS), but we do not complete developing all the function due to the requirements. Some goals were refined to suit the time constraint and user's requirement with acceptable results. The application can provide some features to user for trials. For example, users shall be able to control HEMS-connected appliances using both their voice command or gesture. On the other hands, the application has some parts missing. That is users do not be able to view data in chart mode, for example.

As we have experimented Hololens for smart home system, we believe that augmented reality for smart home system can make our life easier because we can do things more natural. There are many parts that can be further improved including hardware and software aspect.

## 8.2 Problems and Obstacles

Since team members are new to almost all the field in this project, we have undergone several problems and obstacles due to new field of this project. A few of them are:

**Hardware Limitation** — Although Microsoft Hololens provides gestures input to interact with your holograms naturally, they are not significantly supportive for this project. Just only a few of gesture styles were given. Therefore, the development is limited to have users use their gesture naturally.

**Software Limitation** — Even though Microsoft Hololens has an augmented reality technology to display holographic scene, they do not support some kinds of user-interface rendering. At first, our team tried several ways to render web user-interface to Hololens, however, it can not due to its restriction.

**Natural Language Processing** — Specific type of voice commands are needed when users are about to command using their speech. Users are not able to speak any desired sentence if they want to access home appliances using voice command. Because, currently, the project does not support a variety of voice command to be the input, so some user might not be satisfied with the system requirements.

## 8.3 Lessons Learned

After all that we have been through in this project, there have been numerous lessons we have learned during the period of the project. Some of them are:

**Implementation on Augmented-Reality** — Not only did the holographic scene on Hololens we learned to apply with smart home, but also voice command provided by the Hololens that we had to go through.

**Technology on web-based application** — there are several technologies that support web application, for example, real-time web protocol that allows user to have an up-to-date information. Moreover, 3D modelling in web application is an incredible technology that allows user to experience a 3D object in 2D screen.

**Experience from project-based learning** — At the very beginning of this project, we had an opportunity working the real project where there were user's requirements given to you with time constraint, and you had to put your effort to reach the user's needs. That was a very challenging experience, because we could integrate our knowledge to apply with the real work.

**Do not expect what you planned** — During the process of working on the project, everything did not go as my team planed and expected to be. Many things has been changed because of some technical problems. So, Readiness and adaptability is what we needed.

## 8.4 Future Work

To improve an experience of using augmented reality technology in company with smart home system, there might need more study to support and reduce some limitation. For instance, user shall be able to use free-style gesture to control the hologram in natural manner, or user shall be able to view real-time data in graph mode. Additionally, personal assistant like JARVIS in the Iron Man Movie who will help you and notify you any events happen in your home, may become real in the project. Additionally, when user gazes at such device, the system should be able to identify the object type and show up related device information instantly on Hololens' screen. However, in the future, there will be more study which can be used to support the free-style gesture and lessen constraints that currently are problematic in this project.

# Bibliography


[1] Galvinpower. Frequently asked questions: Real-time pricing. http://galvinpower.org/power-consumers/act/real-time-illinois/faq.

[2] Visit Hirankitti. An agent framework for home energy management system. In *Proceedings of the World Congress on Engineering 2015*. WCE, 2015.

[3] Microsoft Hololens. Why hololens. https://www.microsoft.com/en-us/hololens/why-hololens.

[4] Microsoft. General frequently asked questions. https://www.microsoft.com/microsoft-hololens/en-us/faq.

[5] Microsoft. Microsoft hololens. https://www.microsoft.com/microsoft-hololens/en-us.

[6] Thomas Vervust Jan-Frederik Van Wijmeersch Tom Ingelbinck Tim Verbelen Maaike Op de Beeck Bart Dhoedt Pieter Simoens, Elias De Coninck. Smart home control with head-mounted sensors for vision and brain activity. *Proceedings of the fifth international workshop on Mobile cloud computing services*, pages 29–33, June 2014.

[7] Snehal Nargundi Pranav Bedekar. A review on home automation using augmented reality. *International Journal of Science and Research (IJSR)*, 5.

[8] Wearable.com. Microsoft announces hololens headset. http://www.wearables.com/microsoft-announces-hololens-headset/.

[9] Huanjin Wang Hongyi Li-Zhongcheng Lei Jing Wang When Yang, Wenshan Hu. Remote control system of web-based 3d smart homes. *International Journal Of Innovative Computing, Information and Control*, 10(2):741–751, 2014.

# Appendices

# Appendix A

# Use Case Diagram

## A.1 Hololens Application

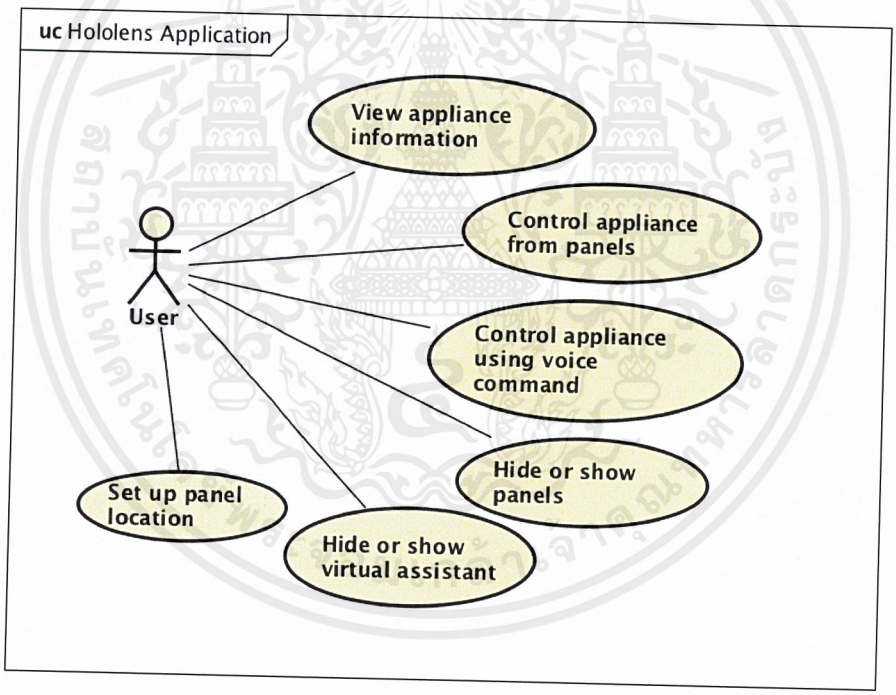



Figure A-1: Hololens application use case diagram

## A.2 Electricity Company Server

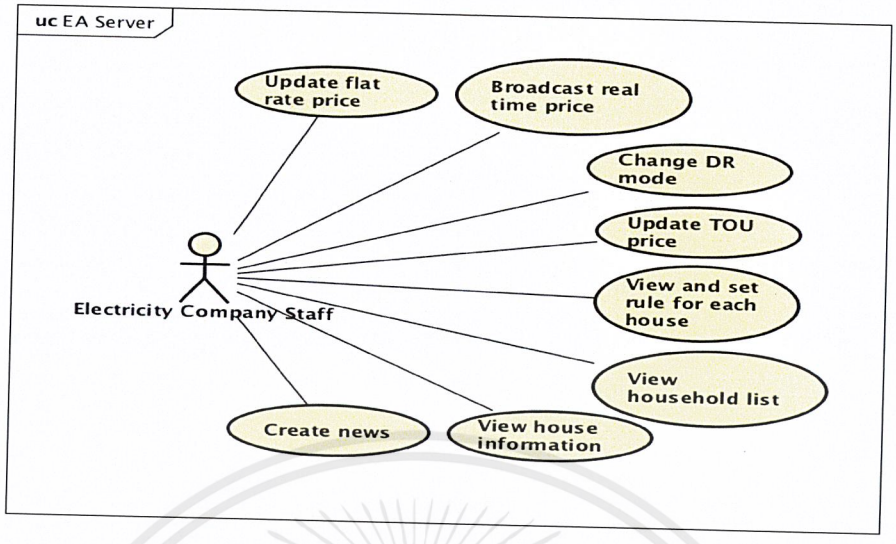



Figure A-2: Electricity Company Server use case diagram

## A.3 Smart Home Web Application

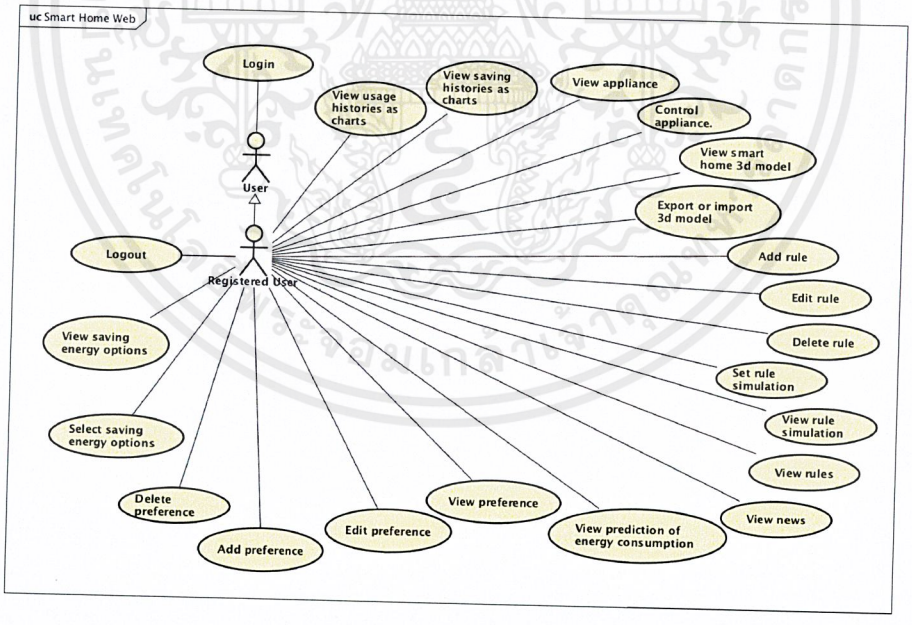



Figure A-3: Smart home web application use case diagram

## A.4 Use Case Description

This section describes about actions between an actor and the system of this application. Each use case will represent a primary actor, pre-condition and post-condition. Moreover, it presents a flow of the events for each action for more clearly understanding.

**Hololens Application**

| **Use Case** | **Set up panel location** |
|---|---|
| **Primary Actor** | User |
| **Brief Description** | Set up panel location by looking at VuMark. |
| **Pre-condition** | - |
| **Post-condition** | Panel are shown where it is set up. |
| **Flow of events** | 1) User looks at the VuMark. |
| | 2) System will remember the location. |

Table A.1: Set up panel location

| **Use Case** | **View appliance information** |
|---|---|
| **Primary Actor** | User |
| **Brief Description** | View appliance information such as status and usage. |
| **Pre-condition** | Set up panel location. |
| **Post-condition** | Panel are shown where it is set up. |
| **Flow of events** | 1) User looks at the panel. |
| | 2) System shows the necessary info.; such as status and usage. |
| | 3) User sees the detail of the appliance. |

Table A.2: View appliance informationn

| Use Case | **Control appliance from panels** |
|---|---|
| **Primary Actor** | User |
| **Brief Description** | Control appliance by click adjustable feature at the panel. |
| **Pre-condition** | Set up panel location. |
| **Post-condition** | Appliance change its state according to what user controls. |
| **Flow of events** | 1) User looks at the panel. |
| | 2) System shows the necessary info.; such as status and usage. |
| | 3) User clicks adjustable feature provided by the appliance. |
| | 4) User changes the status of the appliance. |
| | 5) System checks validation of the change. |
| | 6) System applies the change to that appliance. |
| | 7) System responds the user's change back to user. |

Table A.3: Control appliance from panels

| Use Case | **Control appliance using voice command** |
|---|---|
| **Primary Actor** | User |
| **Brief Description** | Control appliance by using voice command. |
| **Pre-condition** | Set up panel location. |
| **Post-condition** | Appliance change it's state according to what user control and the system should reply the result of recognizing the voice command. |
| **Flow of events** | 1) User speaks some command to the Hololens. |
| | 2) System recognizes the voice message and does analysis. |
| | 3) System applies the voice command by doing Adjust Appliance Status feature. |
| | 4) System responds back to the user by voice message. |

Table A.4: Control appliance using voice command

| Use Case | **Hide or show panel** |
|---|---|
| **Primary Actor** | User |
| **Brief Description** | Hide or show panel. |
| **Pre-condition** | Set up panel location. |
| **Post-condition** | Panel will be hidden or shown according to what user control. |
| **Flow of events** | 1) User looks at the panel. |
| | 2) User chooses to hide or show panel. |
| | 3) System applies change to panel. |

Table A.5: Hide or show panel

| Use Case | **Hide or show virtual assistant** |
|---|---|
| **Primary Actor** | User |
| **Brief Description** | Hide or show virtual assistant. |
| **Pre-condition** | - |
| **Post-condition** | Virtual assistant will be hidden or shown according to what user control. |
| **Flow of events** | 1) User looks at the virtual assistant. |
| | 2) User chooses to hide or show virtual assistant. |
| | 3) System applies change to virtual assistant. |

Table A.6: Hide or show virtual assistant

**Electricity Company Server**

| Use Case | **Update flat rate price** |
|---|---|
| **Primary Actor** | Electricity Company Staff |
| **Brief Description** | Update flat rate price to every household. |
| **Pre-condition** | - |
| **Post-condition** | Flat rate price is updated by Electricity Company staff. |
| **Flow of events** | 1) User keys the price for each range. |
| | 2) User clicks update button. |
| | 3) System gets price and updates price in real time. |

Table A.7: Update flat rate price

| Use Case | **Broadcast real time price** |
|---|---|
| **Primary Actor** | Electricity Company Staff |
| **Brief Description** | Send real time price to every household. |
| **Pre-condition** | - |
| **Post-condition** | Price is updated in every household. |
| **Flow of events** | 1) User chooses date. |
| | 2) User selects new price. |
| | 3) User clicks submit. |
| | 4) System gets date and new price. |
| | 5) System updates price in real time. |

Table A.8: Broadcast real time price

| Use Case | **Update TOU price** |
|---|---|
| **Primary Actor** | Electricity Company Staff |
| **Brief Description** | Update TOU price to every household. |
| **Pre-condition** | - |
| **Post-condition** | TOU price is updated by Electricity Company staff . |
| **Flow of events** | 1) User keys the price for peak and off peak. |
| | 2) User keys the monthly fee. |
| | 3) User clicks update button. |
| | 4) System gets price and monthly fee then updates price in real time. |

Table A.9: Update TOU price

| Use Case | **Change DR mode** |
|---|---|
| **Primary Actor** | Electricity Company Staff |
| **Brief Description** | Change DR mode from specific house. |
| **Pre-condition** | - |
| **Post-condition** | DR mode is changed. |
| **Flow of events** | 1) User chooses house from id. |
| | 2) User selects DR mode. |

Table A.10: Change DR mode

| Use Case | **View and set rule for each house** |
|---|---|
| **Primary Actor** | Electricity Company Staff |
| **Brief Description** | View and set rules from specific house. |
| **Pre-condition** | - |
| **Post-condition** | Rules are changed. |
| **Flow of events** | 1) User chooses house from id. |
| | 2) User clicks add button. |
| | 3) System shows add rule dialog. |
| | 4) User fills in all information then clicks add button. |
| | 5) System shows rule. |

Table A.11: View and set rule for each house

| Use Case | **View household list** |
|---|---|
| **Primary Actor** | Electricity Company Staff |
| **Brief Description** | View all household. |
| **Pre-condition** | - |
| **Post-condition** | Household list is shown. |
| **Flow of events** | 1) User chooses households menu. |
| | 2) System shows household list. |

Table A.12: View household list

| Use Case | **View house information** |
|---|---|
| **Primary Actor** | Electricity Company Staff |
| **Brief Description** | View specific household information. |
| **Pre-condition** | View household list. |
| **Post-condition** | Household information is shown. |
| **Flow of events** | 1) User chooses specific household. |
| | 2) System shows all information. |

Table A.13: View house information

| Use Case | Create news |
|---|---|
| Primary Actor | Electricity Company Staff |
| Brief Description | Create news for every household. |
| Pre-condition | - |
| Post-condition | News is created. |
| Flow of events | 1) User chooses news menu. |
| | 2) User clicks create button. |
| | 3) System shows create dialog. |
| | 4) User fills in all information then click save. |

Table A.14: Create news

**Smart Home Web Application**

| Use Case | Log in |
|---|---|
| Primary Actor | User |
| Brief Description | Log in to smart home web. |
| Pre-condition | - |
| Post-condition | User become registered user. |
| Flow of events | 1) User enters username and password. |
| | 2) User clicks login button. |
| | 3) System shows home page. |
| Alternative Flow | Condition : Username or password is invalid. |
| | 3a) System shows error message and asks user to log in again. |
| | 4) Go to step 1. |

Table A.15: Log in

| Use Case | **View usage histories as charts** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | View usage information daily, weekly, monthly or yearly. |
| **Pre-condition** | - |
| **Post-condition** | Usage information is shown. |
| **Flow of events** | 1) User clicks home page. |
| | 2) System shows usage information. |

Table A.16: View usage histories as charts

| Use Case | **View saving histories as charts** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | View saving information daily, weekly, monthly or yearly. |
| **Pre-condition** | - |
| **Post-condition** | Saving information is shown. |
| **Flow of events** | 1) User clicks inverter page. |
| | 2) System shows saving information. |

Table A.17: View saving histories as charts

| Use Case | View appliances |
|---|---|
| Primary Actor | Registered user |
| Brief Description | Viewing appliance list. |
| Pre-condition | - |
| Post-condition | Appliance list is shown. |
| Flow of events | 1) User clicks appliances page. |
| | 2) System shows appliance list. |
| | 3) User chooses specific appliance to view more information. |
| | 4) System shows appliance information. |

Table A.18: View appliances

| Use Case | Control appliances |
|---|---|
| Primary Actor | Registered user |
| Brief Description | Control appliance by clicking adjustable feature. |
| Pre-condition | View appliances |
| Post-condition | Appliance changes its state according to what user controls. |
| Flow of events | 1) User chooses appliance from the list. |
| | 2) System shows adjustable feature. |
| | 3) User clicks adjustable feature provided by the appliance. |
| | 4) User changes the status of the appliance. |
| | 5) User System checks validation of the change. |
| | 6) System applies the change to that appliance. |

Table A.19: Control appliances

| Use Case | **View smart home 3D model** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | Control appliance by click adjustable feature. |
| **Pre-condition** | Web browser should support webVR |
| **Post-condition** | Smart home 3D model is shown. |
| **Flow of events** | 1) User clicks map page. |
| | 2) System shows smart home 3D model. |

Table A.20: View smart home 3D model

| Use Case | **Export or import smart home 3D model** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | Export or import smart home 3D model in json file. |
| **Pre-condition** | - |
| **Post-condition** | Smart home 3D model file is downloaded or imported. |
| **Flow of events** | 1) User clicks map page. |
| | 2) User clicks export button. |
| | 2a) User clicks import button. |
| | 3) System shows dialog where to save file. |
| | 3a) System shows dialog what file do you want to open. |
| 4) User clicks save button. | |
| 4a) User clicks open button. | |

Table A.21: Export or import smart home 3D model

| Use Case | View rules |
|---|---|
| Primary Actor | Registered user |
| Brief Description | View all rules in smart home. |
| Pre-condition | - |
| Post-condition | Rules are shown. |
| Flow of events | 1) User clicks rule page. |
| | 2) System shows rules. |

Table A.22: View rules

| Use Case | Add rule |
|---|---|
| Primary Actor | Registered user |
| Brief Description | Add rule to smart home. |
| Pre-condition | - |
| Post-condition | Rule is added. |
| Flow of events | 1) User clicks rule page. |
| | 2) User clicks add button. |
| | 3) System shows add dialog. |
| | 4) User enters do rule and if rule then clicks add. |
| | 5) System shows added rule. |

Table A.23: Add rule

| Use Case | **Edit rule** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | Edit rule in smart home. |
| **Pre-condition** | - |
| **Post-condition** | Rule is edited. |
| **Flow of events** | 1) User clicks rule page. |
| | 2) User chooses rule from list. |
| | 3) User clicks edit button. |
| | 4) System shows edit dialog. |
| | 5) User edits rule then clicks update. |
| | 6) System shows edited rule. |

Table A.24: Edit rule

| Use Case | **Delete rule** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | Delete rule in smart home. |
| **Pre-condition** | - |
| **Post-condition** | Rule is deleted. |
| **Flow of events** | 1) User clicks rule page. |
| | 2) User chooses rule from list. |
| | 3) User clicks delete button. |
| | 4) System deletes rule. |

Table A.25: Delete rule

| Use Case | **View rule simulation** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | View rule simulation. |
| **Pre-condition** | - |
| **Post-condition** | Rule simulation is shown. |
| **Flow of events** | 1) User clicks rule page. |
| | 2) User clicks simulation result button. |
| | 3) System shows rule simulation. |

Table A.26: View rule simulation

| Use Case | **Set rule simulation** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | Setting rule simulation. |
| **Pre-condition** | - |
| **Post-condition** | Rule simulation is set. |
| **Flow of events** | 1) User clicks rule page. |
| | 2) User set simulation speed. |
| | 3) System applies the change. |

Table A.27: Set rule simulation

| Use Case | **View news** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | View news from Electricity Company server. |
| **Pre-condition** | - |
| **Post-condition** | News are shown. |
| **Flow of events** | 1) User clicks news page. |
| | 2) System shows all news. |

Table A.28: View news

| Use Case | **View prediction of energy consumption** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | View prediction of energy consumption in smart home. |
| **Pre-condition** | - |
| **Post-condition** | Prediction of energy consumption is shown. |
| **Flow of events** | 1) User clicks prediction page. |
| | 2) System shows prediction of energy consumption. |

Table A.29: View prediction of energy consumption

| Use Case | **View saving energy options** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | View saving energy options for each appliance. |
| **Pre-condition** | - |
| **Post-condition** | Saving energy options are shown. |
| **Flow of events** | 1) User clicks saving page. |
| | 2) System shows list of saving energy options. |

Table A.30: View saving energy options

| Use Case | **Select saving energy options** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | Select saving energy options for smart home. |
| **Pre-condition** | - |
| **Post-condition** | Saving energy options are selected |
| **Flow of events** | 1) User clicks saving page. |
| | 2) System shows list of saving energy options for each appliance. |
| | 3) User selects option. |

Table A.31: Select saving energy options

| Use Case | **View preferences** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | View all preference in smart home. |
| **Pre-condition** | - |
| **Post-condition** | Preferences are shown |
| **Flow of events** | 1) User clicks preference page. |
| | 2) System shows preferences. |

Table A.32: View preference

| Use Case | **Add preference** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | Add preference to smart home. |
| **Pre-condition** | - |
| **Post-condition** | Preference is added. |
| **Flow of events** | 1) User clicks preference page. |
| | 2) User clicks add button. |
| | 3) System shows add dialog. |
| | 4) User enters preference information then clicks add. |
| | 5) System shows added preference. |

Table A.33: Add preference

| Use Case | **Edit preference** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | Edit preference in smart home. |
| **Pre-condition** | - |
| **Post-condition** | Preference is added. |
| **Flow of events** | 1) User clicks preference page. |
| | 2) User chooses preference from list. |
| | 3) User clicks edit button. |
| | 4) System shows edit dialog. |
| | 5) User edits preference then clicks update. |
| | 6) System shows edited preference. |

Table A.34: Edit preference

| Use Case | **Delete preference** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | Delete preference in smart home. |
| **Pre-condition** | - |
| **Post-condition** | Preference is deleted. |
| **Flow of events** | 1) User clicks preference page. |
| | 2) User chooses preference from list. |
| | 3) User clicks delete. |
| | 4) System deletes preference. |

Table A.35: Delete preference

| Use Case | **Log out** |
|---|---|
| **Primary Actor** | Registered user |
| **Brief Description** | Log out from smart home web. |
| **Pre-condition** | - |
| **Post-condition** | Registered user become user. |
| **Flow of events** | 1) User clicks logout button. |
| | 2) System log user out. |

Table A.36: Log out

# Appendix B

# Screenshots of Web Application

## B.1 Inverter page

This page depicts an amount of electricity saving on that day, this week, this month, or this year. See Figure B-1.

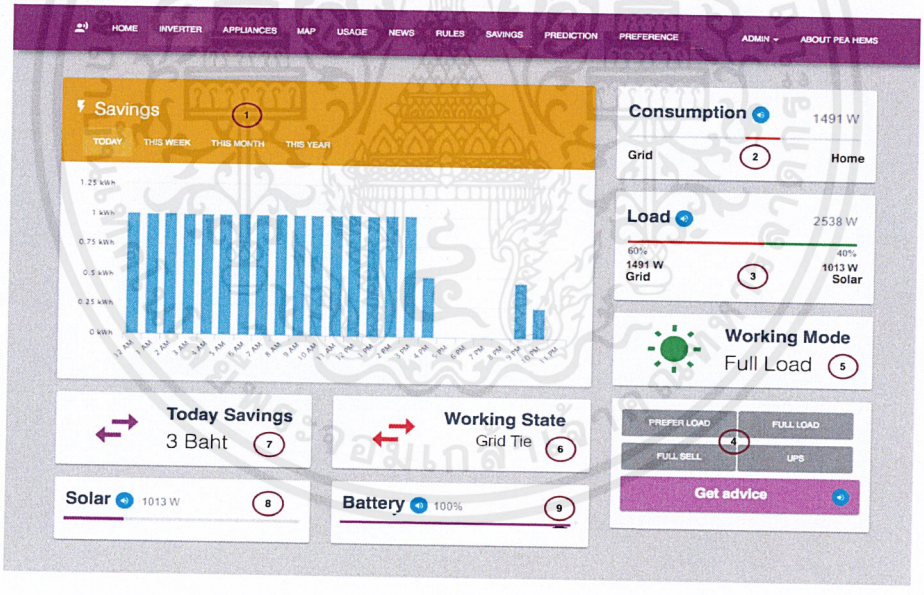



Figure B-1: Inverter page

1. Amount of electricity and graph saving
2. Current consumption
3. Current load

4. A list of working modes of inverter

5. Current working mode status

6. Current working state

7. Total saving on today

8. Solar energy generation

9. Current battery remaining

## B.2 Rules Page

This page acts like a house's controller by letting user simulate a rule to the house via this site, then all things in the rule will be run accordingly. For example of the rule, user prompts the system to turn a HUE-light on and off when these three condition hold: 1. If status of the HUE-light is 0, 2. The time should be in between 12.31 to 14.13, and 3. Mode is Normal,as shown in Figure B-2.

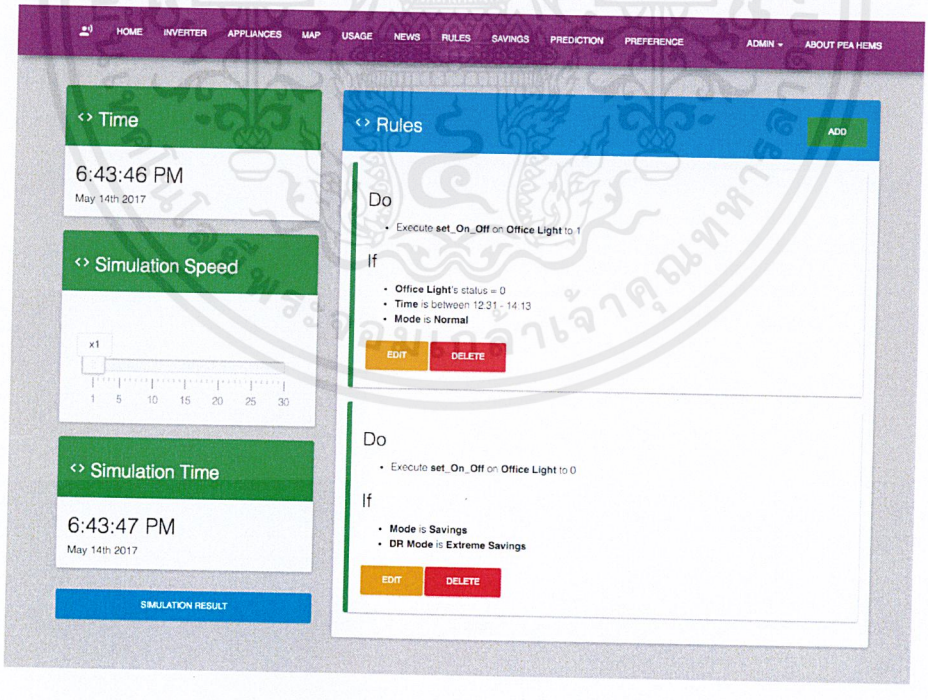



Figure B-2: Rule page

## B.3 Appliances page

This page shows a list of appliances connected to home gateway. Users can select any device to adjust parameters of the device. Figure B-3 shows a list of all devices separated by their location. For example, there is an air-conditioner and a bulb in a bedroom. When users click on the device icon, as shown in Figure B-4, the page will pop up the device details which allows users to adjust the value.

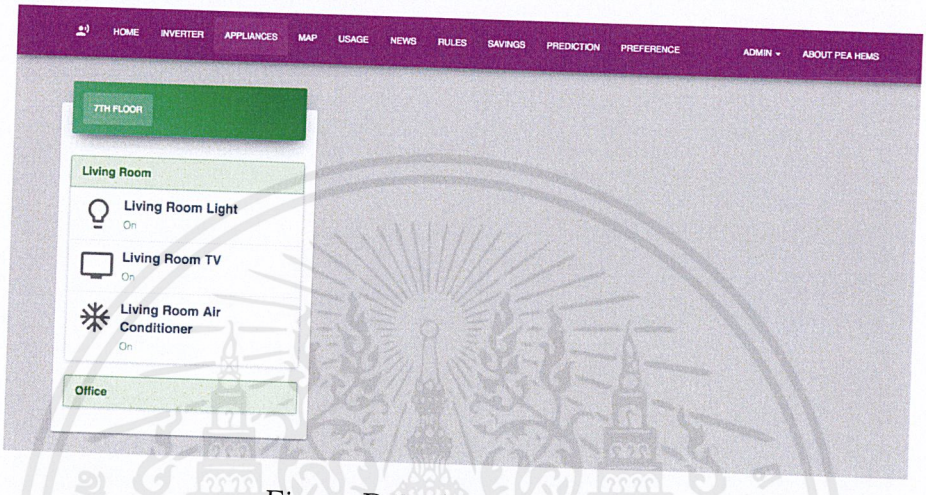



Figure B-3: Appliance page

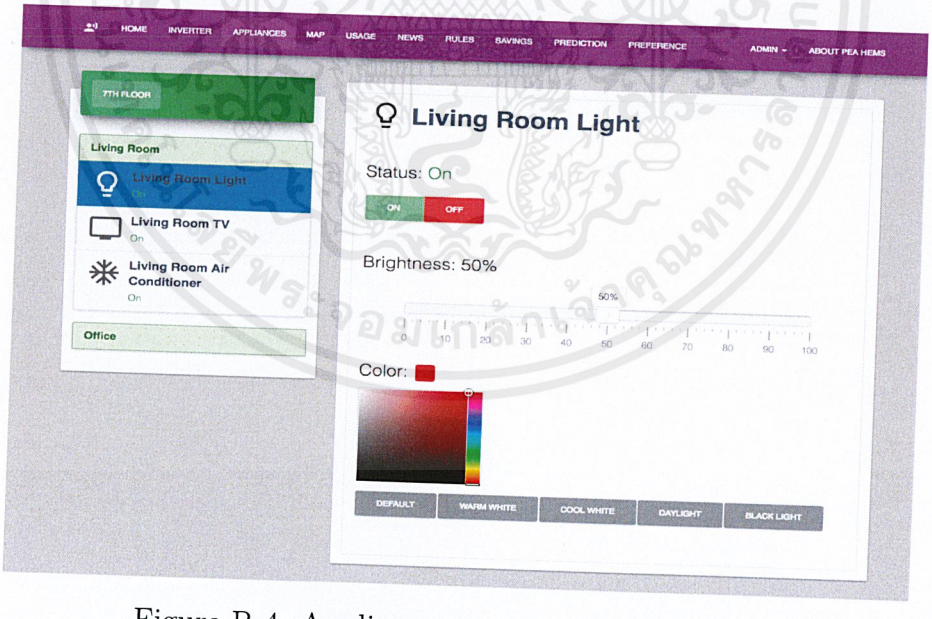



Figure B-4: Appliance page with device details

## B.4 Map page

This page illustrates a 3D house model. Moreover, each device in the model can show its status when that device status is triggered. For instance, as Figure B-5 shown, is a house model. When user changes light color, that light model will turn into another color and appears in the 3D house model in real-time as shown in Figure B-6

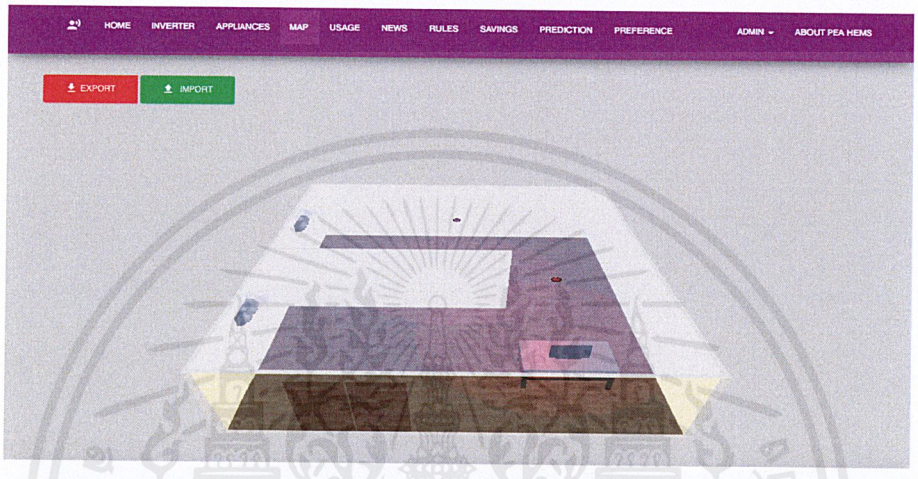



Figure B-5: Map Page


Figure B-6: Map Page

## B.5 Savings Page

This page shows total consumption, Max savings and saving options for each appliance. For example, current option of office light is dimming light to 100 % if user wants to save 15W, user needs to turn off the office light, as shown in Figure B-7 .

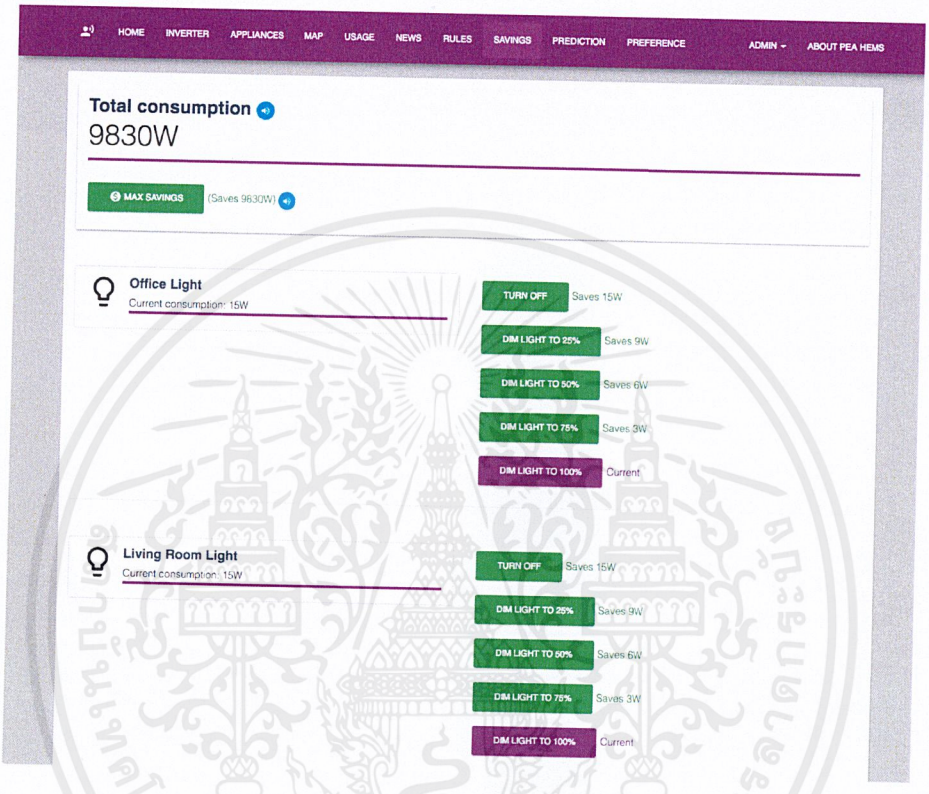



Figure B-7: Savings page

## B.6 Preference Page

This page user can add, edit or delete preference. For example when user sets an office light on between 8:00 to 10:00. In savings page between 8:00 to 10:00 office light will not shows option to turn off light because user set preference,as shown in Figure B-8.

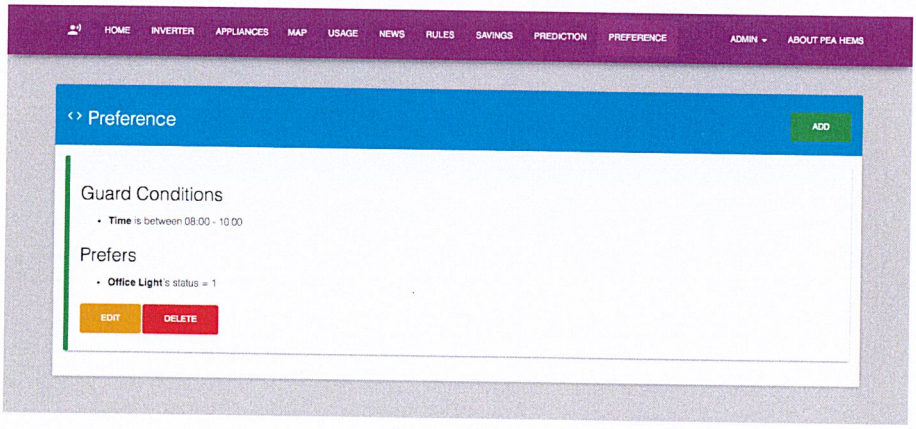



Figure B-8: Preference page

## B.7 News Page

When user comes to this page, user will see news and notification from Electricity Company.

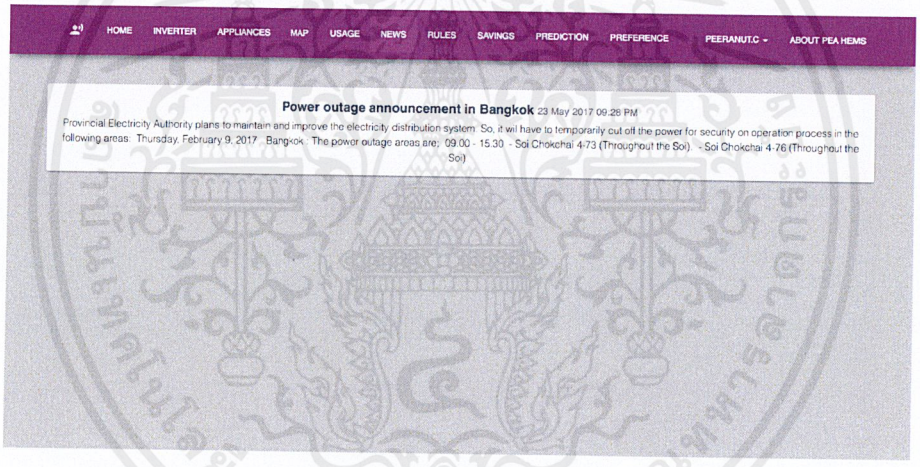



Figure B-9: News page
news