Smart Home Remote Control

Software Requirement Specification

2022.05.01.

Introduction to Software Engineering

TEAM 12 (스마트 리모컨)

Team Leader	조윤근
Team Member	금상인
Team Member	전종문
Team Member	이대희
Team Member	Stefan Hofmann

CONTENTS

LIST OF FIGURE

LIST OF TABLES

1. Introduction

1.1. Purpose

This document is a Software Requirements Specification (SRS) for providing 'Smart Remote Control' services. With this application users can control smart home devices from their smartphone. This service is designed and implemented by Team 12 of the Introduction to Software Engineering class of the spring semester 2022 at Sungkyunkwan University. The requirements for this are summarized, analyzed, and the system is designed and implemented based on the contents described. In this document, Team 12 designs and implements the functions of this service according to this specification. Additionally, professor, TAs, and team members of the Introduction to Software Engineering class can be the main readers. The purpose of this document is to outline and publish the Requirement Specification for a new smart home system controlled by smartphones. Unlike other smart home system our system is able to be controlled by every smartphone independent of the operating system. Furthermore with additional hardware is it possible to include non-smart devices (e.g. simple light bulb) into the smart home system and therefore extend the amount of devices you can control in your smart home.

1.2. Scope

The system is an application that will be able to control all devices in the house which have been added before. You can easily control the system with every smartphone which communicates with an control hub. The control hub will then send the commands to the connected devices. Furthermore you are able to configure certain macros or combinations which will control multiple devices by just selecting one option (e.g. "coming home" mode will turn on the lights as well as the heater or ac and play some music). The system will also support a wide range of different independent of the manufacturer. As another feature the system will use reinforcement learning techniques to learn from the behavior of users and will make recommendations from the gathered data.

1.3. Definitions, Acronyms, and Abbreviation

The following table explains the acronyms and abbreviations used in this document.

[Table 1] Table of acronyms and abbreviations

Acronyms& Abbreviations	Explanation
RAM	Random Access Memory
HDD	Hard Disk Drive
CPU	Central Processing Unit
SSD	Solid-State Drive
OS	Operating System
GUI	Graphical User Interface
API	Application Programming Interface
UI	User Interface
HTTP	Hypertext Transfer Protocol

The following table defines certain technical terms used in this document.

[Table 2] Table of terms and definitions

Terms	Definitions
User	Someone who uses a system
System administrator	Someone who quantify the keywords included in the reviews for each laptop and manage the system
Back-End	Application part that is not directly accessed by the user, such as the server and database
Front-End	The user interface, also known as the presentation layer of an application
Algorithm	A set of rules or procedures followed by a computer in problem-solving

Terms	Definitions
	operations
Client (user device)	A user device/user that connected to server
Server	A computer or computer program which manages access to a centralized resource or service in a network
Software	The programs and other operating information used by a computer
Network	For connect devices together so that they can share information. In this system, it usually means internet

1.5. Overview

The remainder of this Software Requirement Specification document includes two chapters and an appendix. The second chapter provides an overall description of the product's perspective, including multiple interfaces, system features, and system interactions with other systems. This chapter also introduces interactions with different types of stakeholders and systems. This chapter also discusses item details, system constraints, assumptions, and product dependencies. The third chapter provides detailed requirement specifications, including a description of the various system interfaces and software system characteristics. Various specification techniques are used to more accurately specify requirements for different users. It also displays many use cases and data dictionaries. All members contributed equally to the production of this project. We hope readers enjoy this document.

2.   Overall Description

2.1.   Product perspective

This System is designed for people who want to establish a smart home system. The systems goal is to support biggest variety of smart appliances on the market. Moreover it should encourage user to also integrate non-smart devices into their smart home. With cheap hardware it is possible to make ordinary devices smart. The system will be controlled by a smartphone

application in which the user can easily add new devices, create user-specific macros and control all your currently active appliances.

2.1.1. Market status

On the market is a huge variety of smart home systems available (e.g. Google home and Apple home). These systems often support only devices from their own company. For example apple TV can only be controlled by the Apple smart-home system. Smart devices which are supported by both smart home systems are rare. Especially households with different generations tend to use different platforms. While the young people prefer apple products the older generation is more likely to use Android based products. If a household uses devices from different manufacturer the members of the household can't control all devices and therefore can't use the full potential of a smart-home system.

In Addition the market of smart-home appliances compared to normal appliances is still very low. Almost 95 % off all appliances are non-smart. So there is a huge potential to also include these devices for people who want a smart-home system but don't want to buy new devices because their old devices aren't capable of being controlled by a smart-home system.

2.1.2. Overall structure
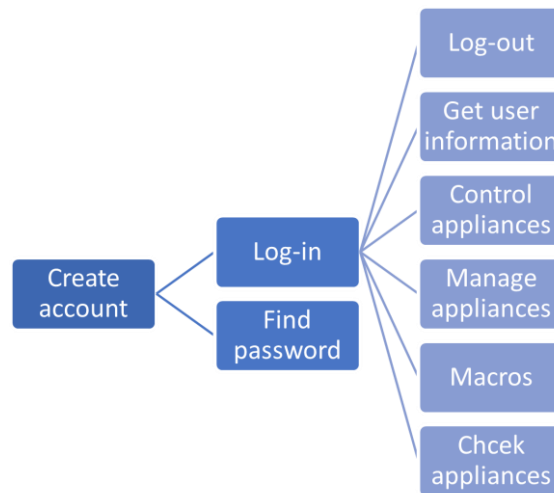
[Figure 1] System Structure

2.2. Product functions

2.2.1. Create user account, log-in, log-out, find password and get user information

The first thing an user has to do is downloading the application. After the user downloaded the application, the user needs an account for using the application functions. The first page of our application is the login page which includes the register button. The user can get to the register page with the register button and fill the required information fields to create their own account. Afterwards the user can log-in and enter to the system with their account.

If the user already logged-in he can log himself out.

If the user forgot his password he can push a button 'forgot password' and afterwards he can change his password via e-mail.

The user can request information about the registered appliances and the created macros after he is logged in.

2.2.2. Add/Delete appliances

On the main page the user can search for unregistered appliances. After this search, it is possible to use an automatic function to add appliances to the user's account. It is also possible to add appliances manually by using the ID of the device. Moreover it is also possible to deregister appliances from the registered appliances list.

2.2.3. Control appliances

It is possible to control appliances directly. The user has to select an appliance icon from the list and gets redirected to the control panel page. At the control page the user can use various options to control the selected device. There exist several control options like: 'Turn on/off', 'Turn up/down the volume', 'Change the channel' and 'set a timer'. Depending on the type of appliance different control options appear on the control page of an appliance (e.g. light only has the option 'Turn on/off').

2.2.4. Create and use Macros

To create a macro the user can use the function 'macro & combine'. After that the user selects an appliance and defines a sequence of controls for this appliance. He can add more appliances to this macro. If all the appliances are temporarily saved he can create the macro.

If the user created a macro it appears in the appliance control page. The User can select the icon and the previous defined appliances and the recorded controls will be executed.

2.2.5.   Check connectivity and check currently working appliances

The connectivity check and the currently working appliances check are executed periodically. The 'check connectivity' function removes appliances from the available appliances list if they are disconnected. The 'check currently working appliances' maintains a list with all the

currently working appliances.

2.3.   User Classes and Characteristics

2.3.1.   User

The user for this application can be any user who has downloaded the application from an App-Store (either use Android or iOS). Additionally the user wants to control at least one device in his home. It is also assumed that the user can read and understand English. Users should also be able to configure smart appliances in a way that can be connected to the smart home system.

2.3.3.   System Manager

The system manager in this application is limited to specialist of the system. They need to have sufficient knowledge of the system and can understand overall system circumstance. It is also assumed that they have competence to detect the system error or such kind of problems. And they should be able to deal with that error or problems. It is assumed that the system managers have ability to reflect the new system changes. To satisfy these things, the system administrator must be a computer engineering, network, or system-related major, or a person with equivalent knowledge and qualifications. And, of course, they must have a software ethics consciousness too.

2.4.   Operating Environment

2.4.1.   Hardware

The system is for Android or iOS mobile phones. The device needs at least 1 GB RAM and 1.0 GHz single processor.

2.4.2. Software

The system is for Android OS and iOS. For Android its version should be at least Android 10.0 (API 29). But the version Android 12 (API 32) is the best environment. For iOS its version should be at least iOS 13.0. But the version iOS 15.4 is the best environment.

2.5. Design and Implementation Constraints

The system will be designed and implemented by considering the following checklist. There are details of design and implementation directions.

- ✓ Showing the main page should not take more than 2 seconds.

- ✓ The system needs to consider the users' convenient and requirements.

- ✓ The system should run on mobile devices with least 1 GB RAM.

- ✓ The system requires at least 600 MB for installation and execution.

- ✓ Source code will be optimized properly to prevent memory waste.

- ✓ User device and server communicate with HTTP protocol.

- ✓ Consider both the system cost and its maintenance cost.

- ✓ basic English is required.

- ✓ Avoid using additional pays for royalty, using open-source software as much as possible.

- ✓ Develop with Windows 10 environment and Android Studio. And the build tools version is 29.0.3

- ✓ Develop with minimum Android version 10.0 (API 29)

- ✓ Develop with minimum iOS version 13.0

2.6. User Documentation

To assist end users to use the application and its service, some documentations are needed. First,

minimum hardware and software requirements. And User manuals are also needed, the contents of user manuals are application installation guide to help users that not familiar with the device, description of how to start the system and how to use different features of the system. The screenshots explaining main features of the system and example inputs and outputs are attached in the user manual too and the explanation video about this document contents is also provided. And last, information to contact the developer of the system are given.

2.7. Assumptions and Dependencies

The systems are designed and implemented based on Android and iOS devices. Thus, all function and contents are based on the Android OS as well as on iOS.
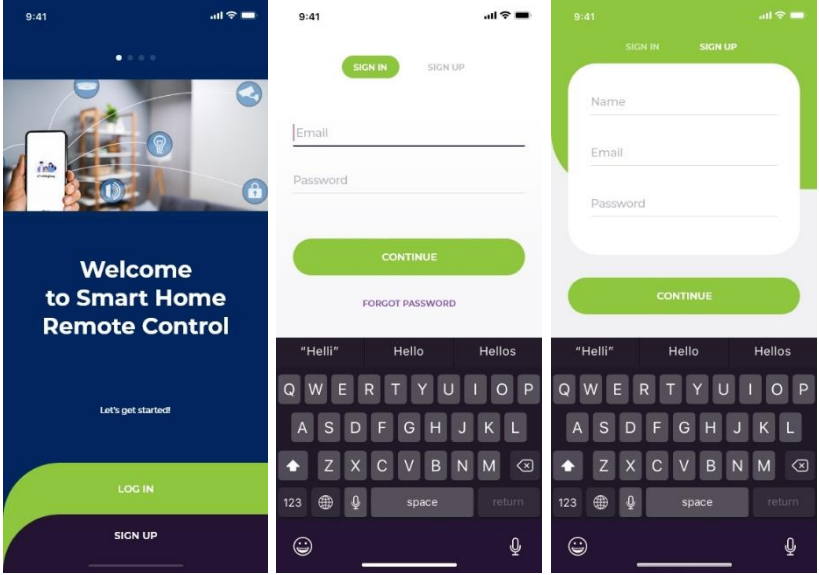
3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

[Table 3] Starting Interface
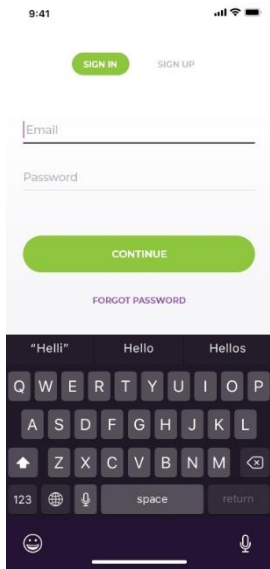
Name	Starting Interface
Purpose/Description	Users choose whether login or register to use this program.
Input source/ Output destination	Client/Server
Range/ Accuracy/ Margin of error	N/A
Unit	Screen
Time/ Velocity	N/A
Relationship with other input/outputs	N/A

Name	Starting Interface
Format and configuration of screen	 1. When "Log in" button is pressed, the device shows new page that user can type in "email" and "password" 2. When "sign up" button is pressed, the device shows new page that user can register.
Format and configuration of window	N/A
Data type	Button, Text, Image
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

[Table 4] Login Interface
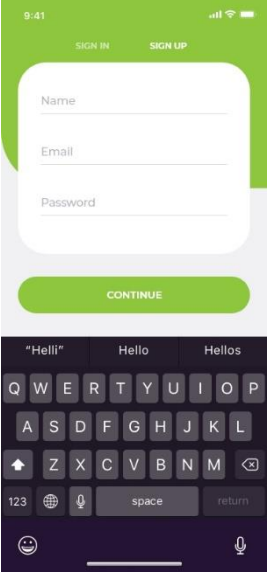
Name	Login Interface
Purpose/Description	Login to use this program.
Input source/ Output destination	Client/Server
Range/ Accuracy/ Margin of error	N/A
Unit	Screen

Name	Login Interface
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	 1. If the user type in legitimiate user information, then the device shows main page which offers user to register new appliances.
Format and configuration of window	N/A
Data type	Button, Text, Image
Instruction type	Instruction mapping according to the value of a button
Exit message	"welcome [USER]"

[Table 5] Sign Up Interface

Name	Sign Up Interface
Purpose/Description	Register to use this program.
Input source/ Output destination	Client/Server
Range/ Accuracy/	N/A

Margin of error	
Unit	Screen
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	1. If the user use the app for the first time, then the user is required to register his ID using their email address. 2. If the user type in their information with right format and press "continue" button, the user information will be saved in the database. If the user information is in wrong format, the app ask the user to type the information again in right format. 3. When the register is completed, the app shows the "Starting Interface".
Format and configuration of window	N/A
Data type	Query
Instruction type	Instruction mapping according to the value of a button code
Exit message	"성공적으로 가입되었습니다."
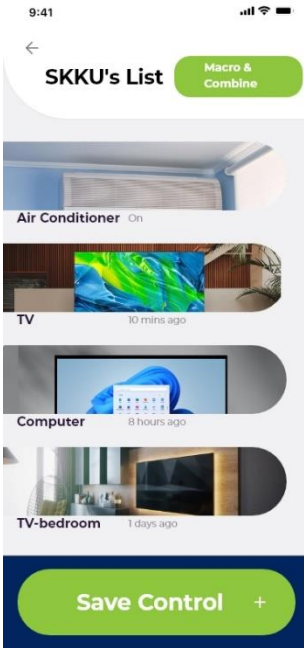
[Table 6] Main Interface

Name	Main Interface
Purpose/Description	Users transmit their instructions screen of the device, and check running appliances
Input source/ Output destination	User/ user device equipped with Android OS
Range/ Accuracy/ Margin of error	Range according to the number of buttons on the screen/ Accuracy according to the accuracy of touch from users/ Margin of error of touch sensitiveness
Unit	A click
Time/ Velocity	Asynchronous user input/ Instant execution of a user instruction
Relationship with other input/outputs	After receiving all the inputs, the user device transmits the input data to the server for processing the input data and request desired output data
Format and configuration of screen	1. On the top of the screen print (user id)'s Home 2. With the horizontal scroll tab, user can easily access to the

Name	Main Interface
	appliances that the user had controlled previously
	3. With "appliances list" user can access to all of the registered appliances.
	4. With "register appliance" user can register new appliances into the app.
Format and configuration of window	N/A
Data type	Screen
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

[Table 4] Remote control Interface

Name	Appliances List Interface
Purpose/Description	Show registered home appliances
Input source/ Output destination	Client/ Server
Range/ Accuracy/ Margin of error	N/A
Unit	Screen
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
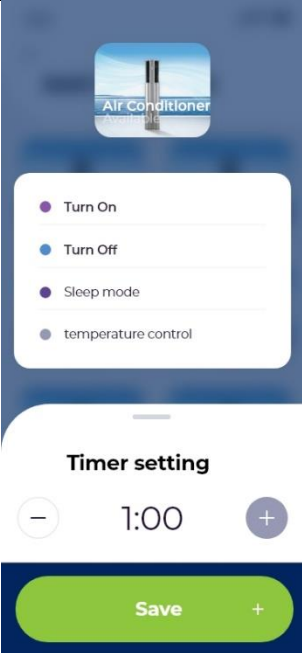
Name	Appliances List Interface
Format and configuration of screen	 1. User can access to the macro list that they had previously set up. 2. By pressing appliance images, the app shows "Remote Control Interface" of the appliance 3. By pressing "Macro & Combine" button, the app shows "Save Control Interface"
Format and configuration of window	N/A
Data type	Screen
Instruction type	Instruction mapped to the button
Exit message	N/A

[Table 5]Save Control Interface

Name	Save Control Interface(1)
Purpose/Description	Build/Save new macro controls
Input source/ Output destination	User/ Servers
Range/ Accuracy/ Margin of error	N/A
Unit	Page
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	1. When the user press the image of each appliances,

Name	Save Control Interface(1)
	the app shows "Save Control Interface(2)" of each appliances 2. The app remembers every interaction(such as pressing buttons) that happened in "Remote Control Interface" 3. When the user press the "Save to Control List" then the app shows "Check Saved Control Interface"
Format and configuration of window	N/A
Data type	Query
Instruction type	Instruction mapped according to the value of a button code
Exit message	"[동작] 등록 완료"

Name	Save Control Interface(2)
Purpose/Description	Save controls that the user want and save it to control list
Input source/ Output destination	User/ Servers
Range/ Accuracy/ Margin of error	N/A
Unit	Page
Time/ Velocity	N/A
Relationship with other input/outputs	N/A

Name	Save Control Interface(2)
Format and configuration of screen	 1. The user can choose actions for the appliances 2. The user can set time with "Timer setting" 3. The user can save actions with pressing the "Save" button and the data will be saved in "Check Saved Control Interface"
Format and configuration of window	N/A
Data type	Query
Instruction type	Instruction mapped according to the value of a button code
Exit message	"[동작] 등록 완료"

Name	Check Saved Control Interface
Purpose/Description	The user can check actions that will be saved in Control list, and actually can save
Input source/ Output destination	User/ Servers
Range/ Accuracy/ Margin of error	N/A
Unit	Page
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	 1. The user can see actions that they previously had

Name	Check Saved Control Interface
	added 2. The user can delete registered actions by tapping twice at the action 3. If the user press "Save to Control List" then the combination of action will be saved into one macro, and move the screen into "Appliance List Interface"
Format and configuration of window	N/A
Data type	Query
Instruction type	Instruction mapped according to the value of a button code
Exit message	"[매크로] 등록 완료"
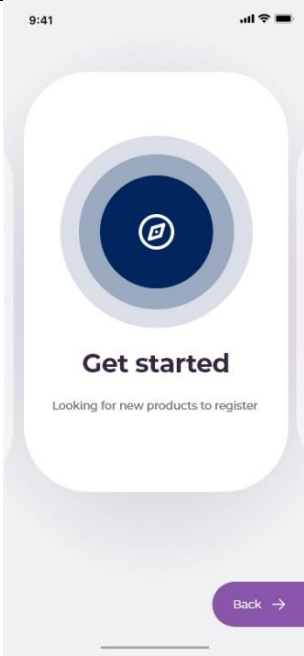
[Table 6] Remote Control Interface

Name	Remote Control Interface
Purpose/Description	가전제품을 조작하는 화면
Input source/ Output destination	Server / Client
Range/ Accuracy/ Margin of error	N/A
Unit	Screen
Time/ Velocity	N/A
Relationship with other input/outputs	N/A

Name	Remote Control Interface
Format and configuration of screen	 1. The user can on / off the button with "on/off" toggle 2. The app can show the user specific information about the appliances (ex. 12 volume / 13(ebs) channel) 3. The user can interact with appliances with "arrow buttons" 4. The "Timer" button shows pop up screen to set the timer
Format and configuration of window	N/A
Data type	Text, Image, Widget
Instruction type	N/A
Exit message	N/A
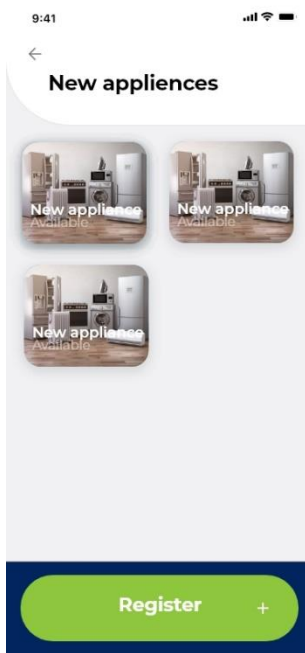
[Table 7] Device Register Interface

Name	Device Register Interface
Purpose/Description	Show appliances that is not yet registered in the app

Name	Device Register Interface
Input source/ Output destination	Client/ Server Server/Client
Range/ Accuracy/ Margin of error	N/A
Unit	Screen
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	 1. When the user press "Get Started" button, the app detect the unregistered appliances. 2. After the detect process is over, the app shows "Device Register List Interface" 3. With "back" button, the user can go back to previous screen
Format and configuration of window	N/A
Data type	Image, Text, Widget

Name	Device Register Interface
Instruction type	N/A
Exit message	N/A

[Table 8] Device Register List Interface

Name	Device Register List Interface
Purpose/Description	Show list of appliances that is not registered
Input source/ Output destination	Server/Client
Range/ Accuracy/ Margin of error	N/A
Unit	Screen
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	

Name	Device Register List Interface
	1. The user can find list of appliances that is not registered. 2. The user can add new appliances by clicking the image of the appliances. If the user press the image one more time, then the appliance will not be registered. 3. When "Register" button is clicked, chosen appliances will be added to appliance list.
Format and configuration of window	N/A
Data type	Image, Text, Widget, Query
Instruction type	N/A
Exit message	"제품 등록이 완료되었습니다."

3.1.2. Hardware Interfaces

[Table 9] Hardware Interface

Name	Applicable device for the system
Purpose/Description	Enable users to take advantage of the service provided by the system/Android OS Enabled Smart Phone. (At least Android 6.0)

3.1.3. Software Interfaces

[Table 10] Software Interface

Name	Firebase Real-time Database
Purpose/Description	Query input/output for managing multimedia/meta data

Name	Firebase Real-time Database
Input source/ Output destination	Host server/ User, User/Host server, User/User
Range/ Accuracy/ Margin of error	Depends on the performance of the Firebase
Unit	Query
Time/ Velocity	Instant reaction
Relationship with other input/outputs	Related to all inputs/outputs from server
Format and configuration of screen	N/A
Format and configuration of window	N/A
Data type	Query
Instruction type	Query statement
Exit message	N/A

3.1.4. Communication Interfaces

[Table 11] Communication Interface

Name	Client and Host
Purpose/Description	Each client requests the connection to the host, requesting list of appliances, list of macros, list of unregistered appliances. Host provides list of appliances, list of macros, list of unregistered appliances.
Input source/ Output destination	User/Host server
Unit	packet
Time/ Velocity	At least 10Mbps

Name	Client and Host
Relationship with other input/outputs	Related to all inputs/outputs from server
Format and configuration of screen	N/A
Format and configuration of window	N/A
Data type	Query
Instruction type	Query statement
Exit message	N/A

3.2. Functional Requirements

3.2.1. Use Case

<Account Management>

[Table 12] Use case of register an account

Use case name	Register an account
Actor	Unregistered user
Description	When an unregistered user tries to register in our app, this process will happen.

Normal Course	1. Every user except admin encounters a log-in page after executing the application.
	2. If a user does not have an account, the user will touch the 'sign up' button to register an account.
	3. The user is redirected to the register page.
	4. In the register page, the user should provide several information.
	    1. Name
	    2. Email address (This should be unique)
	    3. Password
	5. The system sends a verification code to the given email address to verify whether the email address belongs to the user.
	6. If the user correctly enters the verification code, the system creates an account for the user and redirect the user to the log in page.
	7. The system sends an email notifying that the user has been registered.

Pre-condition	The user is not registered to the system yet. The user enters correct information. The Email address given by the user belongs to the user.

Post-condition	An Email notifying that the user has been registered is sent to the user's Email.
	The following information should be encrypted and saved to system DB to manage users.
	    a. Name
	    b. Password
	    c. Email address

Assumptions	N/A

[Table 13] Use case of log-in

Use case name	Log-in
Actor	Registered user
Description	Log-in is the process that a registered user tries to log-in to the system.
Normal Course	1. A registered user wants to use the services of the application. 2. The user touches the 'log in' button to get to the log in page. 3. The user enters his/her Email address and password. 4. If the information is correct, the user can log-in successfully and use the system. 5. The system sends information about user's registered appliances and macros to the user.
Pre-condition	The user is already registered into the system.
Post-condition	The user should be connected to the server.
Assumptions	N/A

[Table 20] Use case of log-out

Use case name	Log-out
Actor	Registered user
Description	Log-out is the process that the user who has logged-in tries to get out of the system.

Normal Course	1. If the user wants to get out of the system, the user can touch a log-out button. This button is available in the user profile page. 2. The user touches the log-out button. 3. Then when the user opens the application next time, the user should log-in again. 4. If the user closes the application without logging-out, the system automatically log-out the user.
Pre-condition	The user is logged in the system.
Post-condition	The user has logged out.
Assumptions	The user might close the application without logging-out.

[Table 21] Use case of find password

Use case name	Find password
Actor	Registered user
Description	Find password is the process where a registered user tries to find a password for log-in.
Normal Course	1. A user tries to log-in, but he/she don't remember his/her password. 2. The user touches the 'forgot password' button in the log-in page and the user redirected to the 'find password' page. 3. If the user enters the correct email address, the system sends a verification code to the email address. 4. If the user enters the sent code correctly, the user is redirected to the 'change password' page. 5. The user enters a new password for his/her account. 6. If the entered password is strong enough, the system changes the user's password to the new one. 7. The user is redirected to the log-in page.
Pre-condition	The user knows his/her email address for log-in but don't remember his/her password.
Post-condition	The user has changed his/her password.
Assumptions	The user forgot his/her password.

\<Initial Action\>

[Table 22] Use case of get user information

Use case name	Get user information
Actor	Registered user
Description	User information about registered appliances and macros is retrieved from the system right after a user logs-in.
Normal Course	1. The user just logged in and request information about his/her account from the system. 2. The system searches its DB and sends related information to the user. 3. The user receives the information.
Pre-condition	The user is logged in.
Post-condition	The user is logged in. The user receives his/her information about registered appliances and macros.
Assumptions	The user logged in just now.

<Appliance management>

[Table 23] Use case of search unregistered appliances

Use case name	Search unregistered appliances
Actor	Registered user
Description	The user searches unregistered appliances around the controlling machine (e.g., mobile phone).
Normal Course	1. The user touches 'register appliance' button, and the user is redirected to the 'search appliance' page. 2. The user touches 'search' button to search unregistered appliances around the controlling machine. 3. The application shows appliances that can be registered.
Pre-condition	The user is logged in. There are unregistered appliances that can be connected nearby.
Post-condition	The user is logged in. The user can see appliances that can be registered.
Assumptions	N/A

[Table 24] Use case of register appliances(automatic)

Use case name	Register appliances(automatic)
Actor	Registered user
Description	The user can use this functionality to register appliances automatically.
Normal Course	1. The user touches icons of unregistered appliances that have already been found by 'Search unregistered appliances' functionality. 2. The selected appliances are registered to the user's account. 3. The information about newly added appliances is sent to the system.
Pre-condition	The user is logged in. The user has found some unregistered appliances nearby that can be connected.
Post-condition	The user is logged in. The selected appliances are registered to the user's account. The system updates information about the user in the DB.
Assumptions	N/A

[Table 25] Use case of register appliances(manual)

Use case name	Register appliances(manual)
Actor	Registered user
Description	The user can use this functionality to register appliances manually.
Normal Course	1. The user touches 'manual register' button in the register page. 2. The user enters ID of the target device. 3. If the ID is correct and the target device can be connected to the user, it is registered.
Pre-condition	The user is logged in. The ID of the target device is correct.
Post-condition	The user is logged in. The target appliance is registered to the user's account. The system updates information about the user in the DB.
Assumptions	N/A

[Table 26] Use case of deregister appliances

Use case name	Deregister appliances
Actor	Registered user
Description	The user can use this functionality to remove appliances from the appliance list.
Normal Course	1. The user touches 'deregister' button in the appliance list page. 2. The user selects appliances to remove from the list. 3. The selected appliances are deregistered, and the system updates user's information in the DB.
Pre-condition	The user is logged in. There are appliances registered to the user's account.
Post-condition	The user is logged in. The selected appliances are deregistered. The system updates user's information in the DB.
Assumptions	N/A

<Appliance Control>

[Table 27] Use case of directly control

Use case name	Directly control
Actor	Registered user
Description	The user can use this functionality to control registered appliances directly.
Normal Course	1. The user touches an appliance icon in the appliance list page. 2. The user is redirected to control panel page. 3. The user can use buttons in the page to control the target device, such as turning up/down the volume, turning on/off the appliance, and changing the channel.
Pre-condition	The user is logged in. There is an appliance connected to the user.
Post-condition	The user is logged in. The target appliance does what it is commanded to do.

Assumptions	The connection between the user and the target appliance should be maintained.

[Table 28] Use case of record a macro

Use case name	Record a macro
Actor	Registered user
Description	The user can use this functionality to record a sequence of behaviors that the user wants the appliances to do.
Normal Course	1. The user touches the 'macro & combine' button in the appliance list page. 2. The user is redirected to the appliance selection page. 3. The user selects an appliance. 4. The user is redirected to the appliance control page with a recording button. 5. The user records a sequence of actions the user wants this appliance to do. 6. The user can stop the recording and temporarily save the sequence. Temporarily saved sequences can be found in the 'saved control' page. The 'saved control' page can be accessed by touching the green icon in the bottom-left corner. 7. If the user wants to record more sequences, he/she can repeat 3-6. 8. The user touches the 'saved control' icon in the bottom-left corner. 9. The user is redirected to the 'saved control' page. 10. The user touches the 'save' button to make a macro. The macro consists of sequences of actions that the user recorded. 11. The system updates user's information in the DB.
Pre-condition	The user is logged in. There is an appliance connected to the user.
Post-condition	The user is logged in. The system updates user's information in the DB. A macro is created and added to the user's account.
Assumptions	The connection between the user and the target appliances should be maintained.

[Table 29] Use case of use a macro

Use case name	Use a macro
Actor	Registered user
Description	The user can use this functionality to use an already registered macro.

Normal Course	1. The user touches an icon of a macro in the appliance control page. 2. If the related appliances of the macro are connected, the macro starts. 3. If not, the user cannot start the macro getting an error message saying there is an unconnected appliance related to this macro. 4. The user can stop the macro by touching the macro icon again. 5. The user can start the macro again after it is over. 6. The user cannot start a macro that contains an appliance that is currently executing another macro.
Pre-condition	The user is logged in. There is an already created macro. Appliances related to the macro are connected to the user.
Post-condition	The user is logged in. A macro has finished its job.
Assumptions	The connection between the user and the target appliances should be maintained.

<Appliance Unit Action>

[Table 30] Use case of turn on/off

Use case name	Turn on/off
Actor	Registered user
Description	The user can use this functionality to turn on/off a connected appliance.
Normal Course	1. The user turns on/off the target appliance.
Pre-condition	The user is logged in. There is an appliance registered to the user.
Post-condition	The user is logged in. The target appliance turns on/off.
Assumptions	The connection between the user and the target appliance should be maintained.

[Table 31] Use case of turn up/down the volume

Use case name	Turn up/down the volume
Actor	Registered user
Description	The user can use this functionality to turn up/down the volume of a connected appliance.
Normal Course	1. The user turns up/down the volume of the target appliance.
Pre-condition	The user is logged in. There is an appliance registered to the user.
Post-condition	The user is logged in. The target appliance turns up/down the volume.
Assumptions	The target appliance has a volume controller. The connection between the user and the target appliance should be maintained.

[Table 32] Use case of change the channel

Use case name	Change the channel
Actor	Registered user
Description	The user can use this functionality to change the channel of a connected appliance.
Normal Course	1. The user changes the channel of the target appliance.
Pre-condition	The user is logged in. There is an appliance registered to the user.
Post-condition	The user is logged in. The target appliance changes its channel.
Assumptions	The target appliance has a channel controller. The connection between the user and the target appliance should be maintained.

[Table 33] Use case of set a timer

Use case name	Set a timer
Actor	Registered user
Description	The user can use this functionality to set a timer for a connected appliance
Normal Course	1. The user sets a timer for the target appliance. 2. When the timer is over, the target appliance automatically turns off.
Pre-condition	The user is logged in. There is an appliance registered to the user.
Post-condition	The user is logged in. A timer for the target appliance is set.
Assumptions	The connection between the user and the target appliance should be maintained.

<Regular Execution>

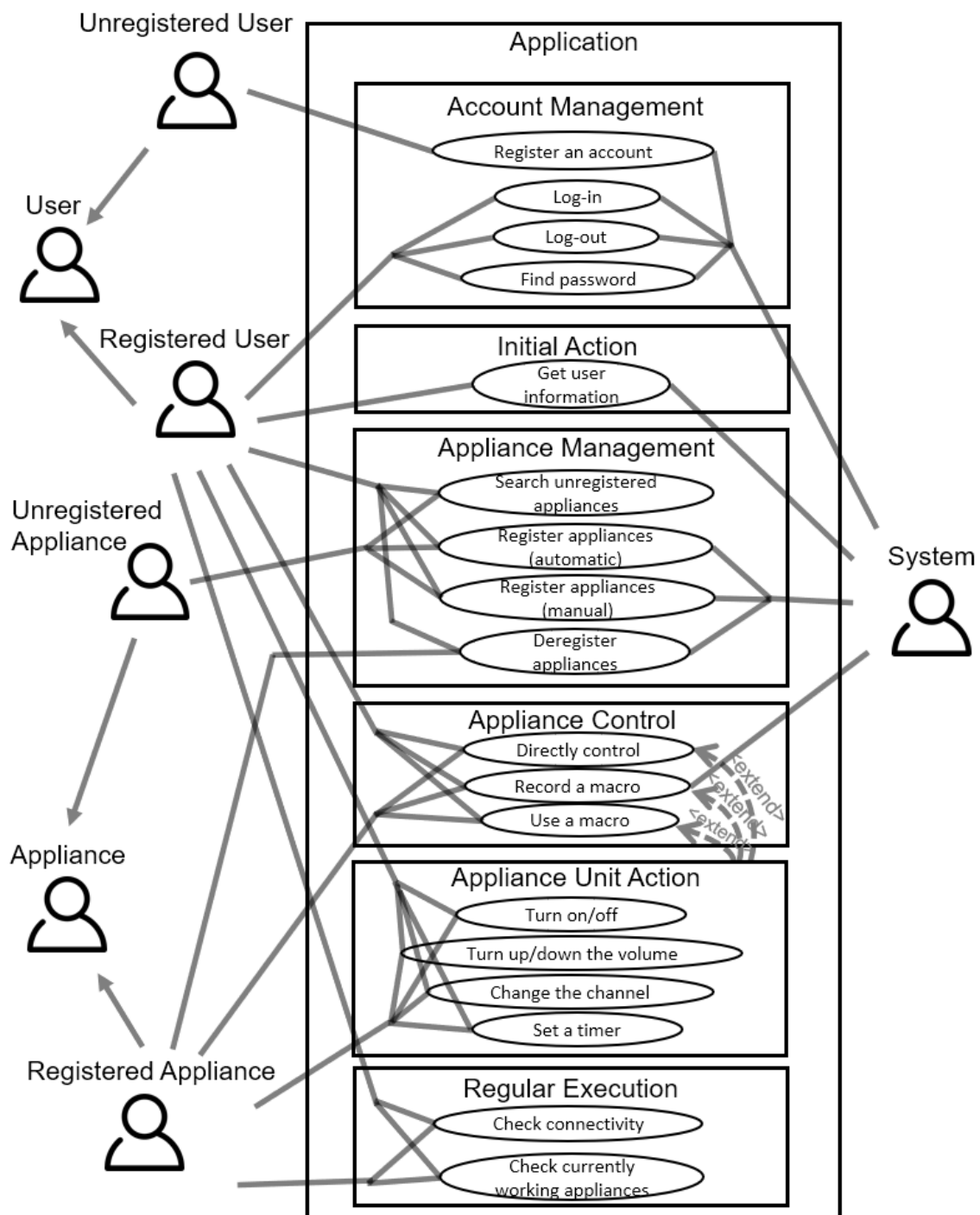[Table 34] Use case of check connectivity

Use case name	Check connectivity
Actor	Registered user
Description	This functionality is executed periodically to check that the registered appliances are connected.
Normal Course	1. This functionality is executed automatically to make sure that the appliance the user is going to use is connected. 2. Even if the user does nothing, this functionality is executed periodically. 3. This functionality is executed every time there is a change to the appliance list. 4. If a disconnected appliance is found, remove it from the available appliances list. Only appliances in the available appliance list can be used. 5. If a connected appliance is found, add it to the available appliances list.

Pre-condition	The user is logged in.
Post-condition	The user is logged in. If a disconnected appliance is found, it is removed from the available appliances list. If a connected appliance is found, it is added to the available appliances list. The available appliances list contains currently connected appliances.
Assumptions	N/A

[Table 35] Use case of check currently working appliances

Use case name	Check currently working appliances
Actor	Registered user
Description	This functionality is executed periodically to check which appliances are currently working
Normal Course	1. This functionality maintains a working appliances list. 2. If an appliance starts working, it is added to the list. 3. If an appliance has finished working, it is removed from the list. 4. A macro containing an appliance in the working appliances list cannot be executed.
Pre-condition	The user is logged in.
Post-condition	The user is logged in. The working appliances list contains currently working appliances.
Assumptions	N/A

3.2.2.   Use Case Diagram

[Figure 2] Use case diagram

3.3.   Performance Requirements

The following requirements are based on estimates and are subject to change as the application is complete.

3.3.1.   Static numerical requirement

 - The system only supports one concurrent user and after disconnecting, users can switch accounts and access.

 - The system is recommended to run on a mobile device with 4 GB of RAM and a 2.0 GHz single processor. And the system supports the latest version of Android 8.0 and above, iOS 13 and above.

3.3.2.   Dynamic numerical requirement

 - The system works well in environments with at least 300 concurrent users. And the system is built to manage at least 10000 members.

 - The mapping within the application must be performed within 5 seconds.

 - The basic function and connection of the application must be performed within 3 seconds.

3.4.   Logical Database Requirements

The system manages user information through a database called Firebase. The system stores user information and location information in a database. The database is managed so that it can have basic performance for processing information through the database.

3.5.   Design Constraints

The system must not contain components that are not covered under the license. The system must be accessible from various mobile devices using the Android and iOS operating system. Administrators should be able to access and manage the system through a web browser and administrator application. The system should be designed to use Firebase's database.

3.6.   Standards compliance

All programs in the system are written according to JAVA standards. Function and variable names in the program use camel notation and underscore notation apply to the database. System management tools follow the HTML5 standard.

3.7.   Software System Characteristics

This section describes non-functional requirements that are categorized as product requirements, organizational requirements, and external requirements. Software system characteristics are revealed through non-functional requirements.

3.7.1.   Product Requirements

The system must meet the following product requirements.

3.7.1.1.   Usability Requirement

The most important of the non-functional requirements, the system should be designed so that it can be easily used by non-experts. Users should be able to use the functions of the system without having to go through a separate manual. If a function is required to be described, it should be described as easily as possible. To satisfy the above, the system needs to have a

simple and easy user interface.

3.7.1.2. Performance Requirement

The algorithm for mapping aims to provide results to users within 5 seconds. This is the most time-consuming operation on the system, which can reduce usability.

3.7.1.3. Security Requirement

The user must be certified as a student at Sungkyunkwan University before using the application. User can use the system after receiving the certification, and for system management, user must obtain a separate certification for the administrator in order to have the authority for system management.

3.7.2. Organizational Requirements

This section covers a wide range of requirements regarding the policy.

3.7.2.1. Environmental Requirement

Location information and location details are taken from 'Naver Map' and used it in the system. Information is provided by the application based on the rich API provided by 'Naver Map'.

3.7.2.2. Operational Requirement

Users are identified through ID, and users are also identified through ID, but detailed ID, but detailed information of each other can be provided by agreement between users. User

information is managed in a database, and user information can be provided within 3 seconds, and location information can be provided within 5 seconds.

3.7.3.　External Requirements

This section covers requirements for external factors.

3.7.3.1.　Safety Requirement

The system must be designed so that users of external systems cannot access the system. In addition, safety must be provided so that data is not damaged by external shocks.
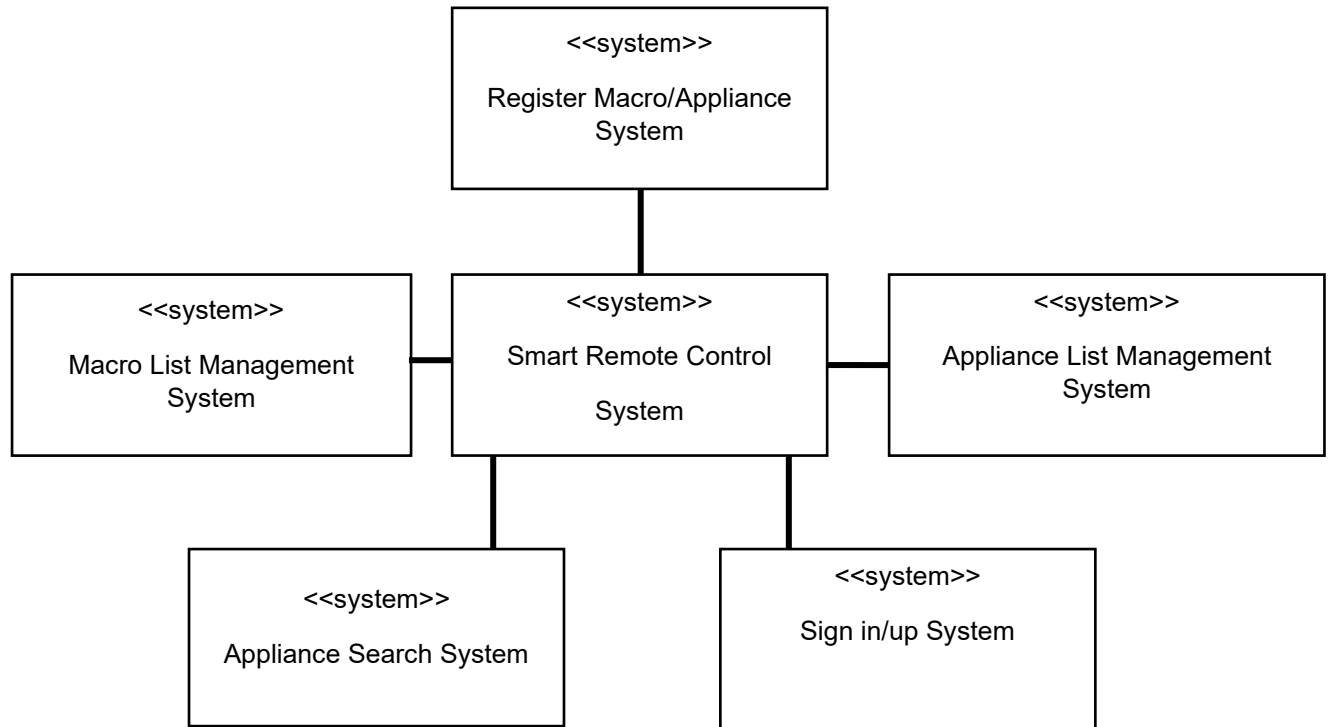
3.7.3.2.　Regulatory Requirement

Permission should be given from Naver to get location information through 'Naver Map'. Users' personal information must be legally protected and not infringed. The system should be developed in accordance with national privacy standards.

3.8.　Organizing the Specific Requirements

This section allows finding out specific system model for requirements. The system model uses a graphical notation based on UML (Unified Modeling Language) and tabular format.

3.8.1. Context Model

[Figure 3] Context Model