Advanced Software Engineering
Fall 2015

Final Specification for Smart Home Guard System

Instructor: Prof. Seok-won Lee

Prepared by :

<Team 3>

Byungwook Lee

Hyesun Lim

Sunghoon Byun

Dec 4, 2015

Contents

Figure & Table

Revision History

Doc. Name	Date	Reason For Changes	Version
Final	15th of Nov	Initial version of Final Document	V 0.1
Final	23th of Nov	Insert the contents of previous documents.	V 0.2
Final	27th of Nov	Created Test case	V 0.3
Final	3rd of Dec	Created Project summary	V 0.4
Final	4th of Dec	Finish the Final Report	V 1.0

Table 1. Revision History

1. Project Overview

The population that live alone is increasing rapidly. This is mainly because the twenties tend to be “Sampo generation” who gives up courtship, marriage and childbirth. According to the Korea National Statistical Office, the ratio of single household is 23.9 percent in 2012, compared to 9.0 percent in 1990. The most recent research, which is conducted in August 2015, shows 26.0 percent of single household ratio, and it looks to be 34.3 percent in 2035 [1].

There is the other statistics that conducted a research on single household of women from 25-year-old to 49-year-old living in Seoul. It is conducted by Seoul municipal office with Ministry of Gender Equality and Family. The characteristics of single family is the highest portion is in elder people above 60 ages, and the portion of women increased a little from 66.1 percent in 2010 to 69.0 percent in 2014 [1].

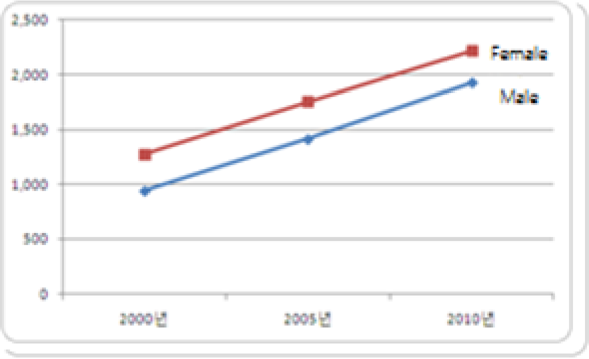


Figure 1. The rate of “Single household” is 23.9 percent in 2012, Rate of female is higher than male [1]

Private security guards said that the most unsafe areas are daycare center, institution for people with disabilities, and single-women household. They said there is need to strengthen social security system, because the crime for the weak is increasing. The majority of security guards said that the most urgent people are the women who live alone [2].

While the single household has been increased, consumption for them has been too increased. Also, a neologism was economically formed which is called “Solo Economy”. However, crimes which is targeting the single household has increased [3].

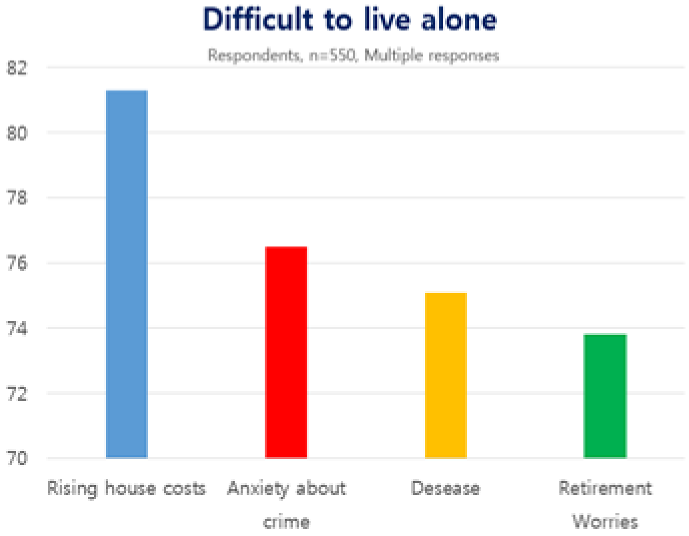


Figure 2. Seoul Foundation of Women & Family, Difficult to live alone. 2012 [3]

When asked a question 'Difficult to live alone', women responded that "Anxiety about crime, such as sexual assault" secondly high. That is, women who live alone are feeling uneasy. Due to the relentless news about female crime, they would anxious a lot. The crime for women who live alone is really happened much?

Figure 3. 45 people of 357 people among the female Single household in Seoul [3]

From the 20-year-old to the 50-year-old woman hit 12.6 percent of 357 respondents replied that the case went through the crime or injury. Also, they hit about 80 percent near the house (around the building (39 percent), inside building (29.3 percent) and in the home (19.5 percent)). This mean that about 13 women out of every 100 people living alone suffered crime. It was higher than previously thought. Now that most of the crimes occur near the house, even what women living alone sleep fitfully is not an idle fear. The most comfortable and safe home more than anything that you need to place surely that cannot shake the "anxiety" is ironic [3].

In spite of the implementation of "women assured home services" and "women assured courier service" currently in Seoul, which is not enough to be away from crime. Surged the single household and sending most of the day while the house is empty for a long time for the company, a crime aiming for the elderly and socially disadvantaged women by increasing. So, it requires the introduction of the institutional system to prevent this kind of crime.

1.1. Objective

Socially the "Single Household" has increased dramatically. It is necessary to prevent unexpected intrusion, that causes greater damage and to get away from crime for women and the elderly. For this reason, we propose a "Smart Home Guard System (SHGS)" that prevents the sneak thief and sex crimes targeting women. And also, this system makes user control the house conveniently while they are out of the home.

During absence, Smart Home Guard System prevent crime target by controlled like someone in the house. When did intrusion detection, the attacker cannot get out of the house by lock is operated. So, police can catch the crime scene in the meantime.

- Lock the front door and windows by TIZEN phone through the wireless [4]
- Turn on the TV and lighting as people at home
- If someone ring a doorbell, user can watch a video of front door and talk over TIZEN phone
- If intrusion is detected, lock the front door and windows, and report to the police

When user is at home, it doesn't detect the motion in house. When it detect external intrusion, it reports to the police and family for preventing the bigger crime.

- When it detect external intrusion, report the warning to the police and family

1.2. Scope

Figure 4 and Figure 5 represent the conceptual framework of the Smart Home Guard System and the scenario flowchart of the Smart Home Guard System, respectively. Our system consists of Raspberry Pi for main server and calculations, Arduino Uno for getting sensor values, and TIZEN Application for user's controller.

The Smart home Guard System is the core of the Home-IoT technologies which are sensing an activity of household, real time streaming for sending a video to user who has TIZEN phone and so on.

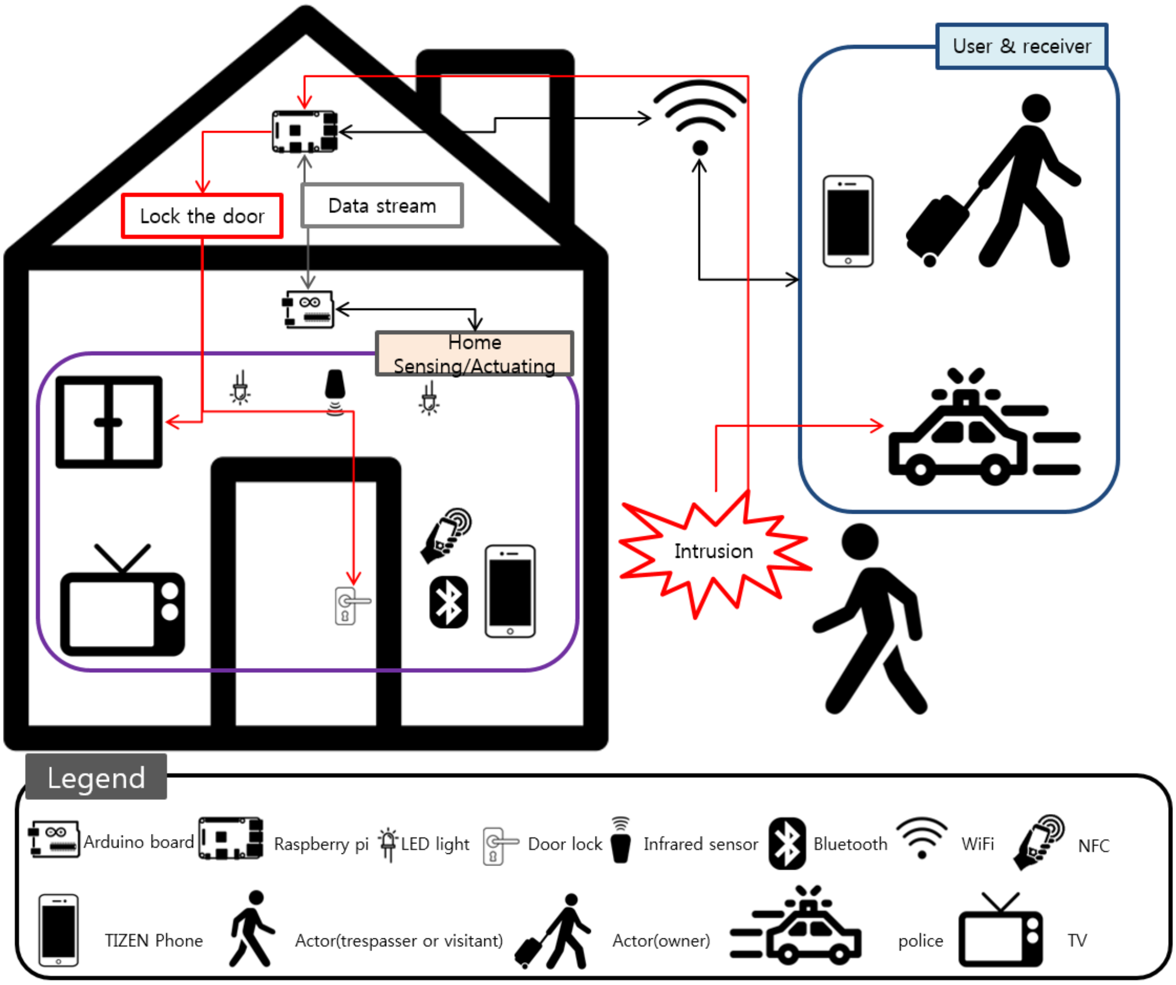


Figure 4. Conceptual Framework of the Smart Home Guard System

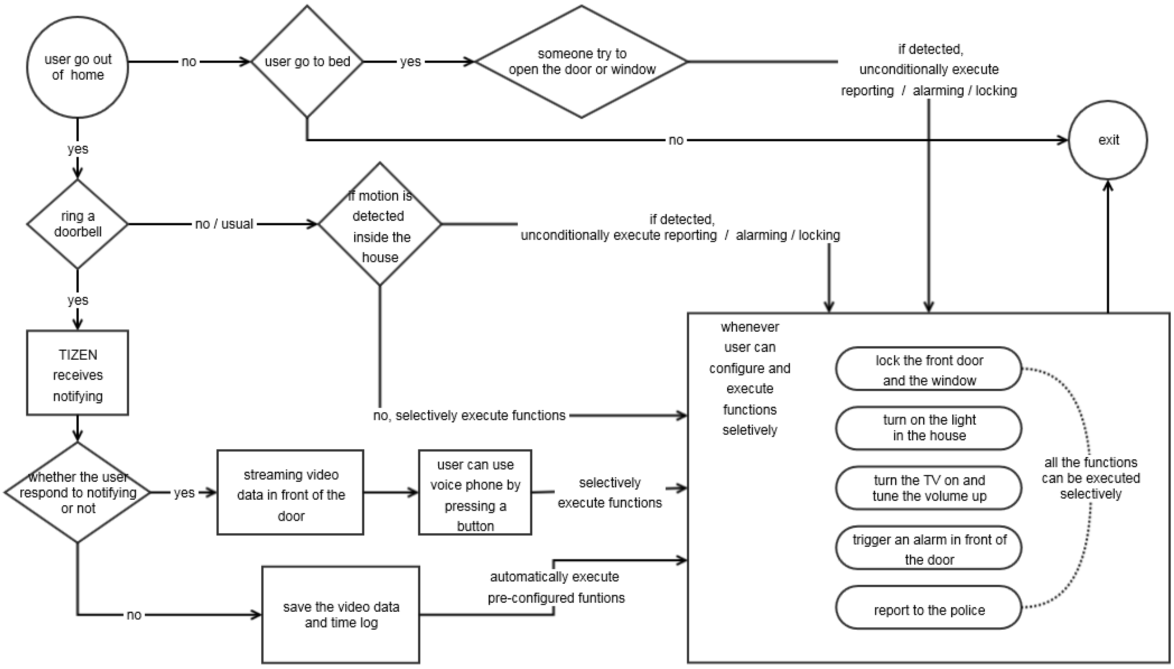

user go out of home
no
user go to bed
yes
someone try to open the door or window
if detected, unconditionally execute reporting / alarming / locking
no
exit
yes
ring a doorbell
no / usual
if motion is detected inside the house
if detected, unconditionally execute reporting / alarming / locking
yes
TIZEN receives notifying
no, selectively execute functions
whenever user can configure and execute functions seletively
lock the front door and the window
turn on the light in the house
turn the TV on and tune the volume up
trigger an alarm in front of the door
report to the police
all the functions can be executed selectively
whether the user respond to notifying or not
yes
streaming video data in front of the door
user can use voice phone by pressing a button
selectively execute functions
no
save the video data and time log
automatically execute pre-configured funtions

Figure 5. Scenario Flowchart of the Smart Home Guard System

2. Requirements of SHGS

This document represents only the top Goals of SHGS which is Functional Requirement Goal graph and Non-Functional Requirement Goal graph.

2.1. Supplementary Requirements

We derived the Functional Requirement Goal and Non-Functional Requirement Goal for Safety Home Guard System.

2.1.1. Goal-Graph of Smart Home Guard System

We represented the Sub-Goal and Task of implementing SHGS, based on the Goal derived from the above Stakeholder Requirement Goal.

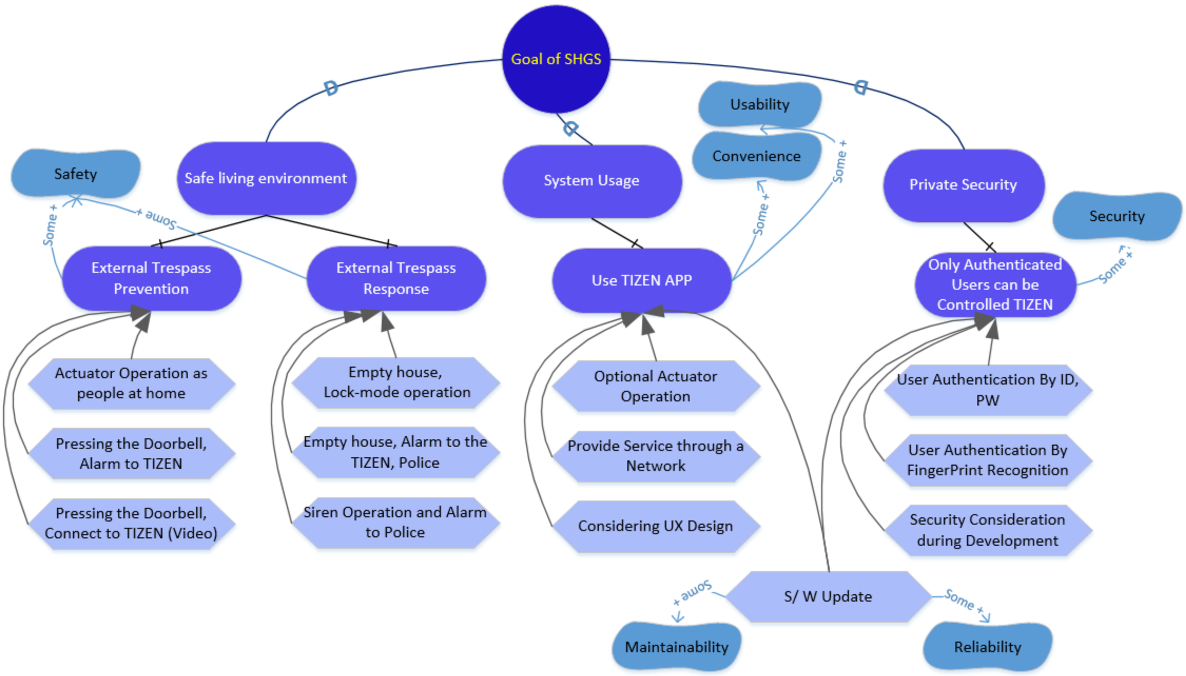

Goal of SHGS
D
D
D
Usability
Convenience
Safety
Safe living environment
System Usage
Private Security
Security
Some +
Some +
Some +
Some +
Some +
External Trespass Prevention
External Trespass Response
Use TIZEN APP
Only Authenticated Users can be Controlled TIZEN
Actuator Operation as people at home
Pressing the Doorbell, Alarm to TIZEN
Pressing the Doorbell, Connect to TIZEN (Video)
Empty house, Lock-mode operation
Empty house, Alarm to the TIZEN, Police
Siren Operation and Alarm to Police
Optional Actuator Operation
Provide Service through a Network
Considering UX Design
User Authentication By ID, PW
User Authentication By FingerPrint Recognition
Security Consideration during Development
Some +
S/ W Update
Some +
Maintainability
Reliability

Figure 6 Goal-graph of the Smart Home Guard System Requirements

2.1.2. Non-Functional Requirements

It arranged the relationship between the Non-Functional Requirement in SHGS. The system includes the properties of the Safety, Security, Performance, Convenience, Usability, Reliability, and Maintainability. We consider the interaction with the Functional Requirements and Non-Functional Requirements of the system.

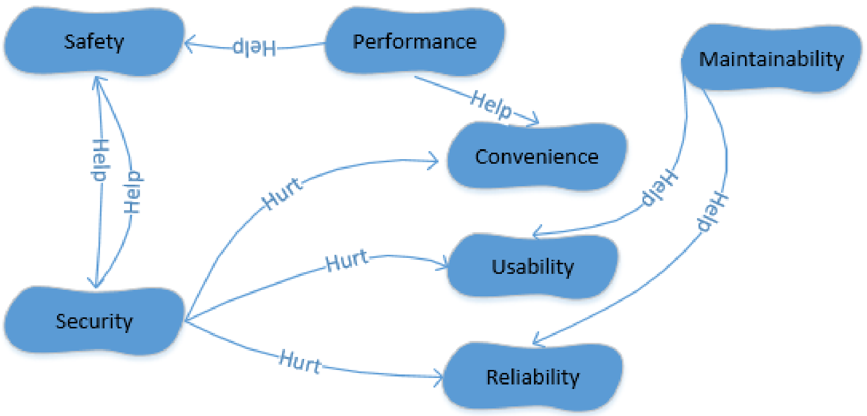


Figure 7. Non-Functional Requirements in System Requirements

2.2. Customer Requirements

2.2.1. Actor

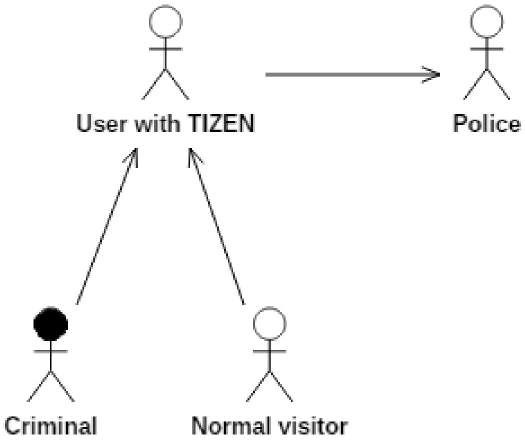


Figure 8. Identification of Actors of the Smart Home Guard System

- User with TIZEN : This is the user of the system. They have TIZEN device that controls the SHGS and contacts with the Police.
- Normal Visitor : Normal visitor is a trusted person based on user actor. They do not affect abnormal.
- Criminal : Criminal has an adverse effect on the SHGS.
- Police : Police can be a police station or a security office if user live in a small apartment. They receive user's request or request of the system.

2.2.2. Essential Use Case Model

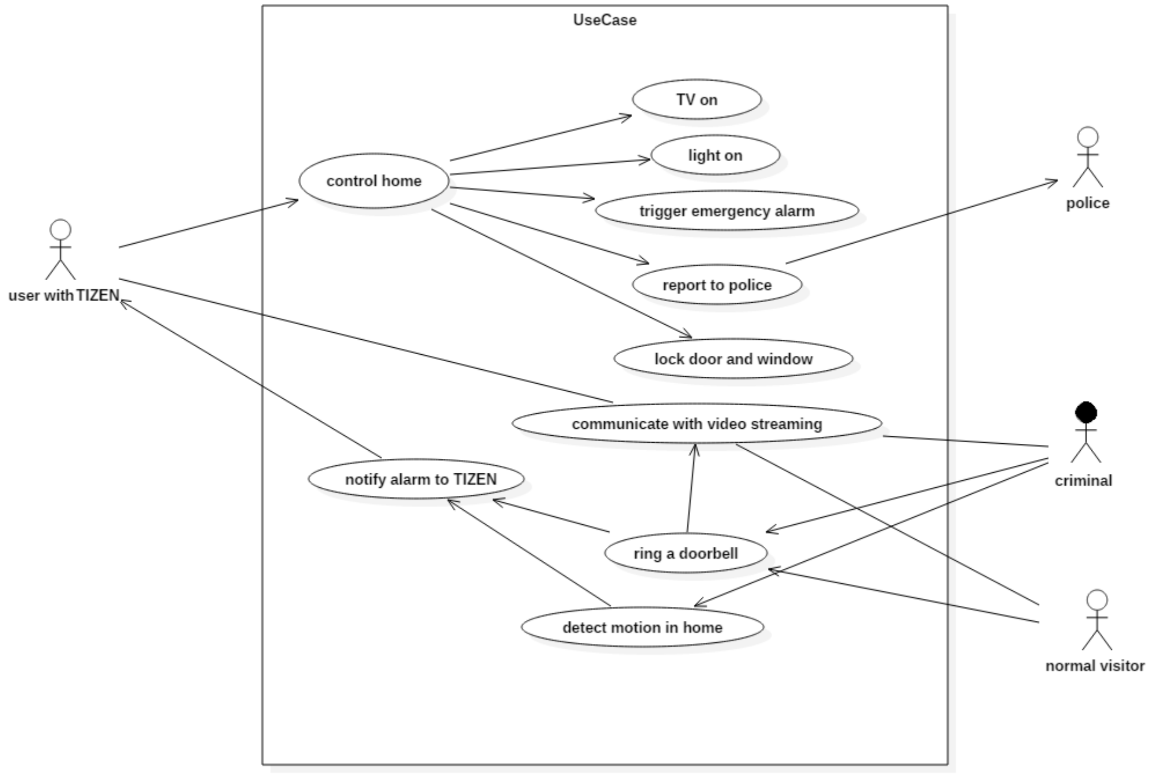


Figure 9. Essential Use Case Model of the Smart Home Guard System

Figure 9 shows the Essential Use Case Model of SHGS. For each design is described in Use Case 3.2 Domain analysis and design.

3. Architectural Design

This section is written for developers who implement the system to provide more detailed technical help. This document contains the design diagrams of each requirement and the whole architecture of the system.

3.1. Package Diagram

We consider three packages to create our SHGS. There are TIZEN system, Arduino system and Raspberry Pi system. TIZEN system control the SHGS to monitor the situation with which user is confronted. Arduino system has two factors which are sensing environment and actuating system. Lastly, Raspberry Pi system can process the control of TIZEN system or Arduino system. Additionally, we draw package diagram using StarUML [5].

3.1.1. TIZEN

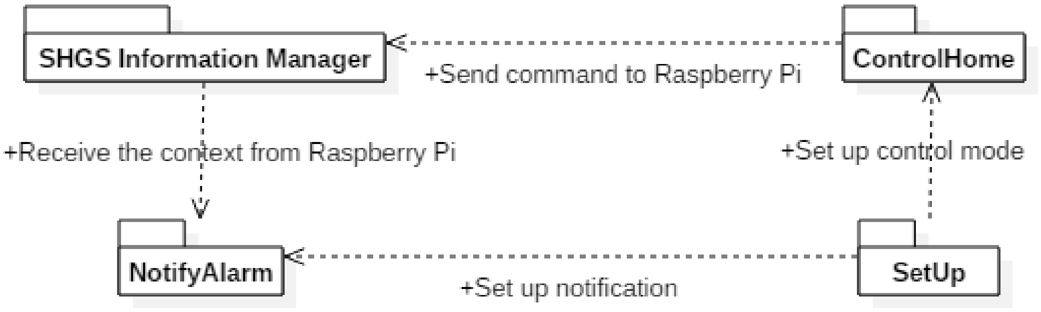


Figure 10. Package diagrams of TIZEN system

TIZEN system have 4 packages; each package performs below roles respectively.

- SHGS Information Manager: SHGS information manager sends processing information from Raspberry Pi system to TIZEN system for notification to user. Also, SHGS information manager receives user command for delivering to Raspberry Pi system.
- Control Home: Control home sends command information for manipulating the SHGS controlled by Raspberry Pi system. Also, Control home is set up by SetUp package for setting up control mode.

- Notify Alarm: Notify alarm receives the context from SHGS information manager, which is delivered from Raspberry Pi. Also, Notify alarm is set up by SetUp package for setting up notification information.
- Set Up: Set up controls the set up information for control home package and notify alarm package. In control set up, there are Lock the door, Turn on TV, Turn on light, and Sound Up. Also in notify alarm set up, there are safe mode window and danger mode window setting.

3.1.2. Arduino

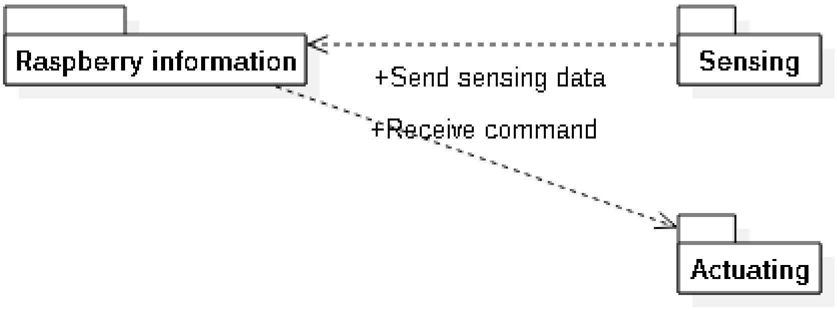


Figure 11. Package diagrams of Arduino system

Arduino system has 3 packages; each package performs below roles respectively.

- Raspberry information: Raspberry information receive the sensing data from sensors which are infrared sensor and switch sensor, and they do perceive inside motion and notify doorbell information, respectively. Also, Raspberry information sends the command from TIZEN to actuators.

- Sensing: Sensing sends the sensing data from sensors which are infrared sensor and switch sensor, and they do perceive inside motion and notify doorbell information, respectively.

- Actuating: Actuating receives command from Raspberry information in order to act right performance. There are LED light, Door lock (servo motor), and Piezo Buzzer.

3.1.3. Raspberry Pi

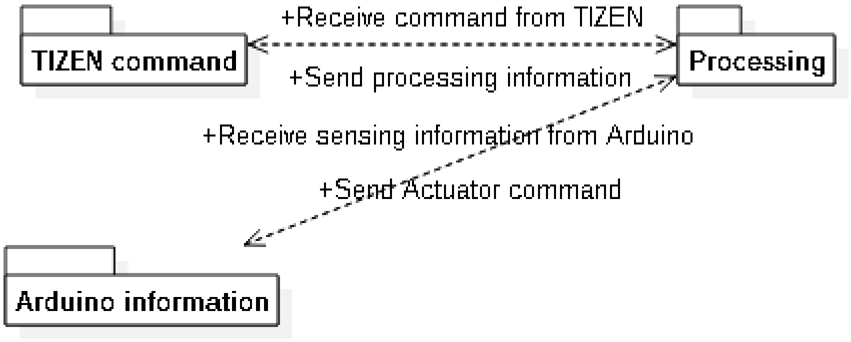

TIZEN command
+Receive command from TIZEN
+Send processing information
Processing
+Receive sensing information from Arduino
+Send Actuator command
Arduino information

Figure 12. Package diagrams of Raspberry Pi system

Raspberry Pi system has 3 packages; each package performs below roles respectively.

- Processing: Processing receives command mode from TIZEN system in order to set the SHGS what the user want, and sends the command to Arduino information for actuating the actuators.

- TIZEN command: TIZEN command receives processing information from processing package, it means that TIZEN system will be notified by Raspberry Pi system. Also, TIZEN command sends the command from TIZEN to processing package, it is about the user’s command, such as turning on TV, turning on the light, and so on.

- Arduino information: Arduino information receives actuator commands from processing package in order to perform the actuators. Also, Arduino information sends the sensing data to processing package, it will be processing in processing package and send the information to TIZEN system.

3.2. Domain Analysis and Design

In this section, we gave a very high-level view over our system. This section's objective is to provide the reader with a good understanding of the details of the logical architecture of our system. We want to explain the static structure of our system, using object-oriented design. We will explore each of our use cases, and we will elicit which classes are necessary. We will also provide a dynamic view over the requirement, so that to explain in what order each function should be called. We use UML class diagrams and UML sequence diagrams respectively to model the static and dynamic aspects of our design. We model using web tool, draw.io [6] and websequencediagrams.com [7].

3.2.1. Use Case with Class diagram, Sequence diagram

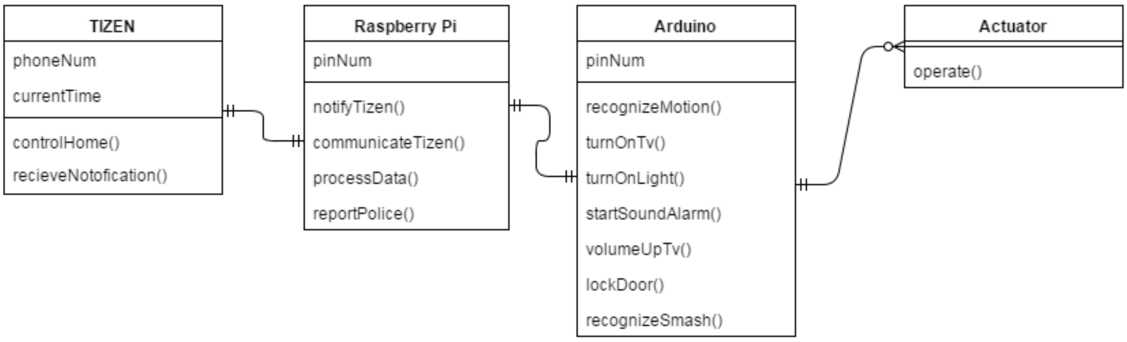
We draw the Class Diagram, Sequence Diagram based on use case which derived from Goal graph.

UC001: Controlling home in ordinary day

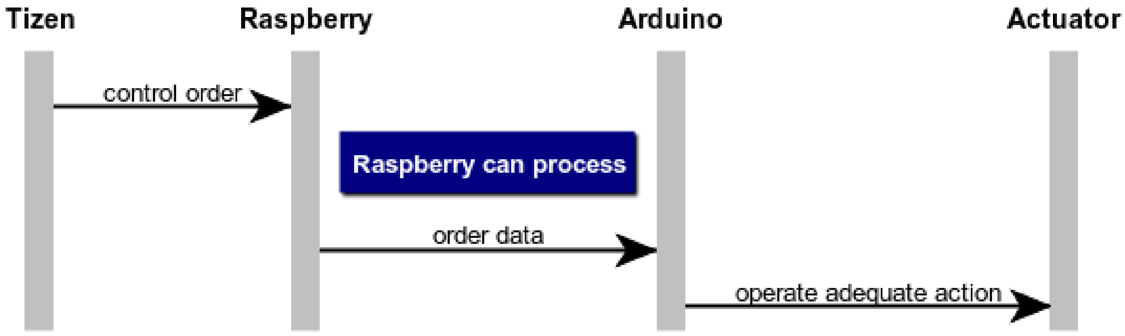
UC001 Class Description

1. User wants to pretend that there is someone at home.
2. User executes mobile application installed on TIZEN.
3. User turn on TV and volume up TV
4. User turn the light on.
5. If user want to change the status of the above, he can control them with TIZEN.

UC001 Class Diagram



UC001 Sequence Diagram

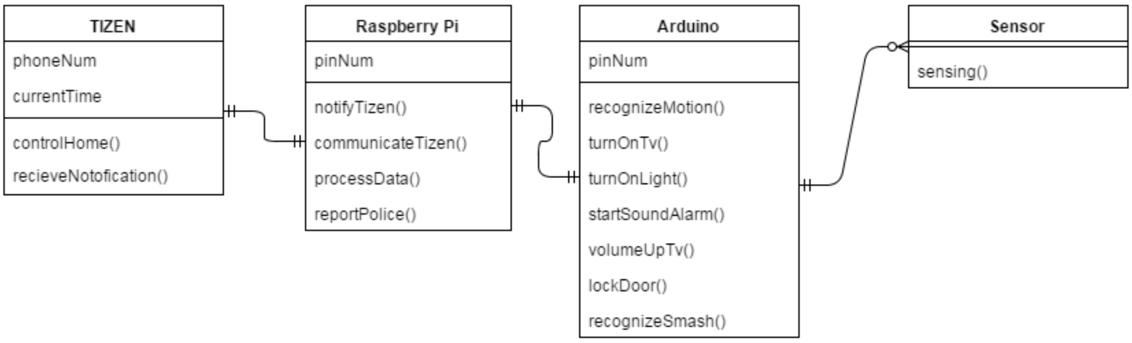


UC002: Communicating with visitors

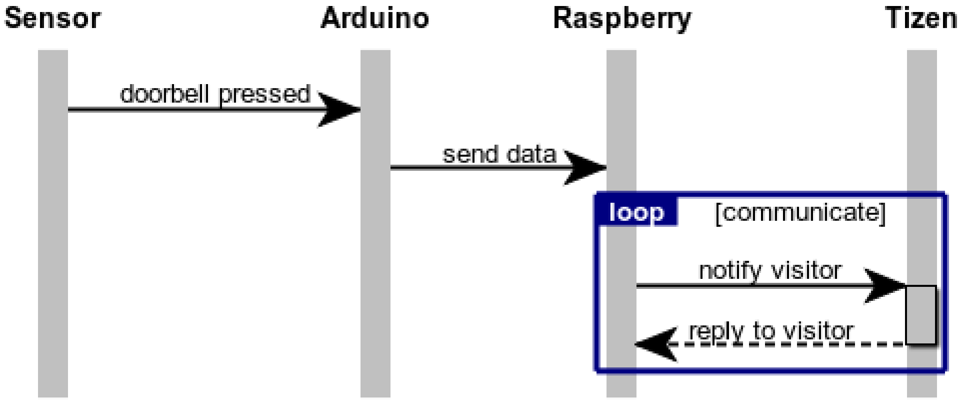
UC002 Description

1. User is carrying his phone.
2. Normal visitor visits user's home.
3. Visitor pressed a doorbell.
4. A notification alarm arrives on the user's phone.
5. User presses 'confirm' button on the phone's screen.
6. User confirm who the visitor is and converse with him.
7. And then, user can deal with the reason why the visitor has visited.

UC002 Class Diagram



UC002 Sequence Diagram

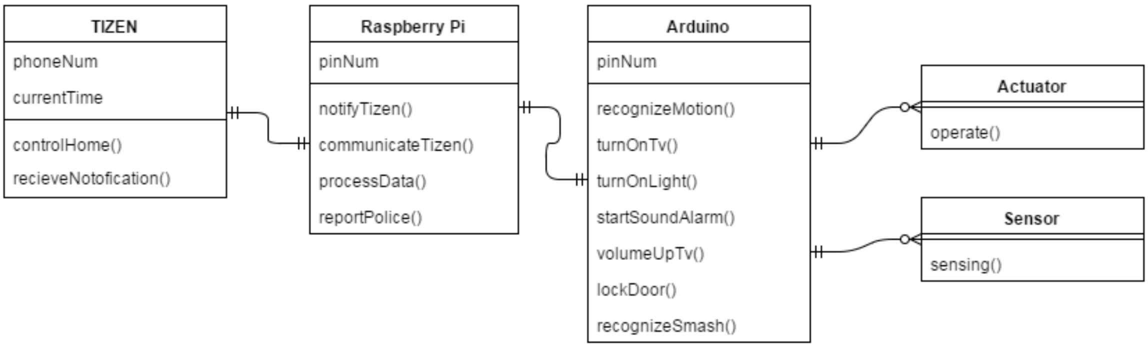


UC003: Replying to malicious visitor

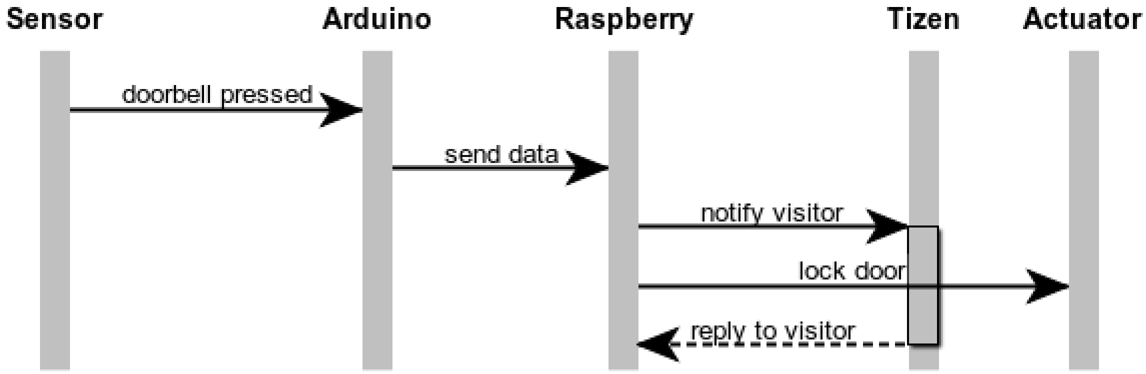
UC003 Description

1. User went out of the home at night.
2. User is carrying his phone.
3. A malicious visitor visits user's home.
4. Visitor pressed a doorbell.
5. A notification alarm arrives on the user's phone.
6. User presses 'confirm' button on the phone's screen.
7. User confirm the person who the visit home and notice that he is suspicious.
8. Turn on the TV and lighting as people at home

UC003 Class Diagram



UC003 Sequence Diagram

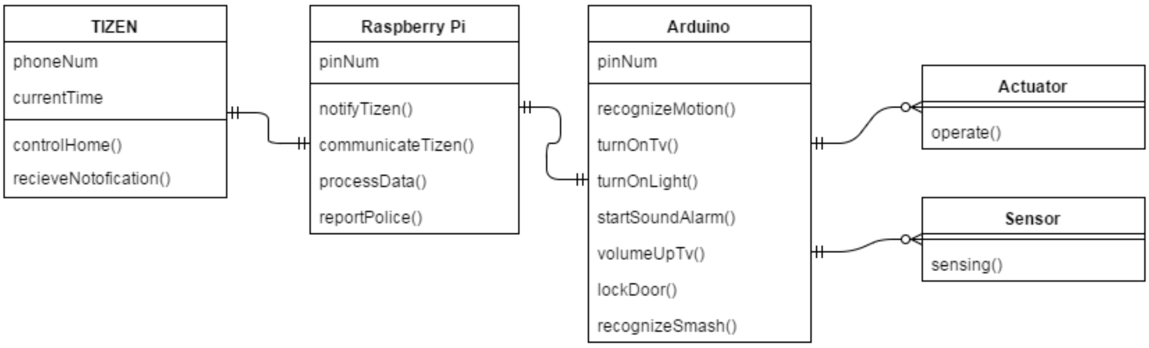


UC004: Automatically protect home

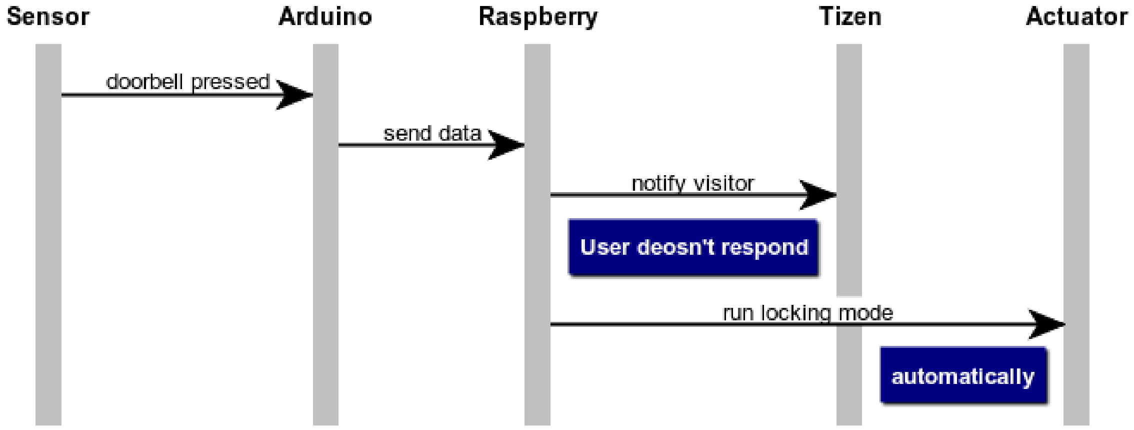
UC004 Description

1. User is not able to check his phone's notification alarm.
2. A malicious visitor visits user's home.
3. Visitor pressed a doorbell.
4. A notification alarm arrives on the user's phone.
5. User cannot confirm the notification.
6. The pre-configured functions are automatically executed.
7. The malicious visitor cannot enter into user's home.

UC004 Class Diagram



UC004 Sequence Diagram

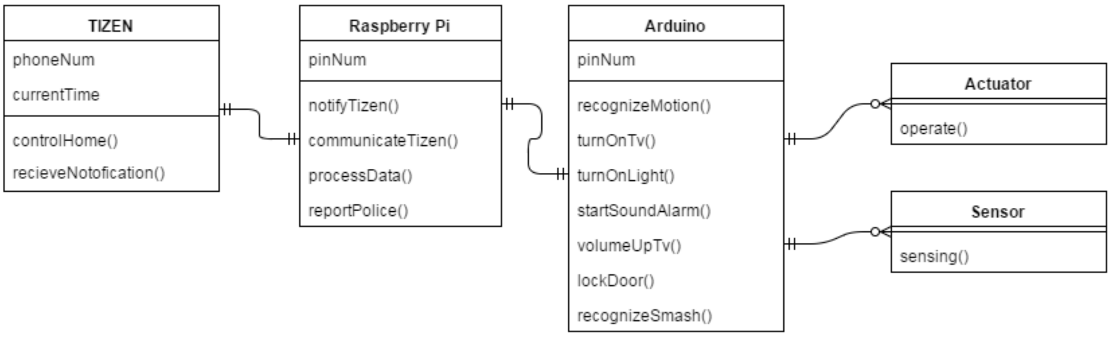


UC005: Automatically protect home

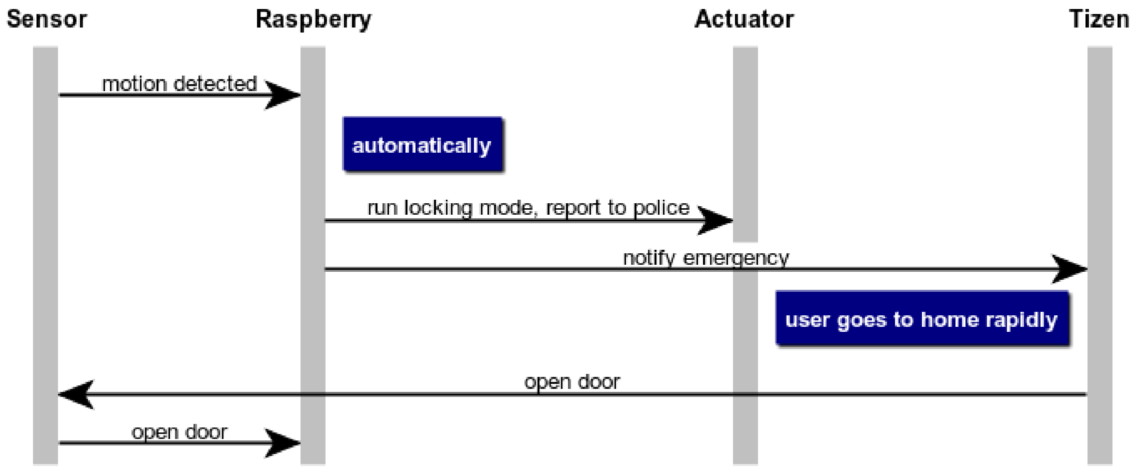
UC005 Description

1. While user is out of home, the window is opened.
2. User's phone receive an emergency alarm.
3. Alarm say that the motion is detected at home.
4. The police arrive at user's home.
5. The user arrive at home.
6. The police open the door with user's phone and enter into home.
7. The police arrests the thief.

UC005 Class Diagram



UC005 Sequence Diagram

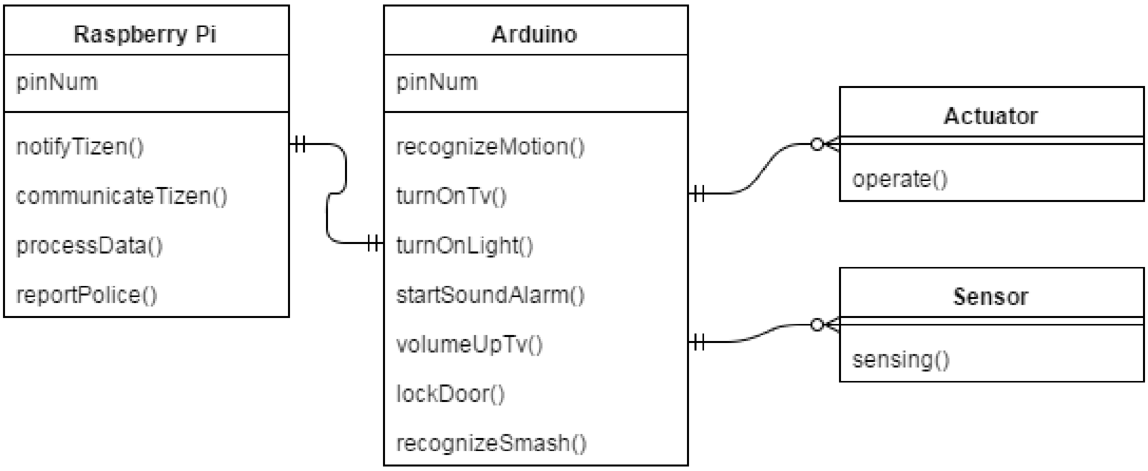


UC006: protecting home during sleep

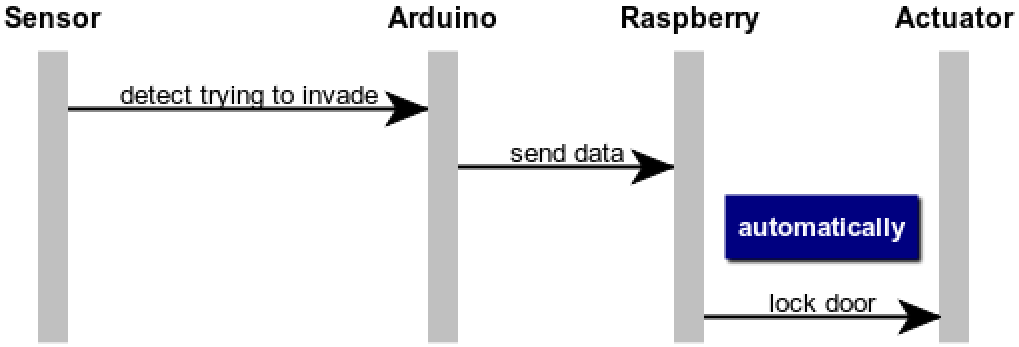
UC006 Description

1. User is sleeping.
2. A thief try to open the door forcibly.
3. The pre-configured functions are automatically executed.
4. Because of the system's functions, the thief cannot enter into use's home.

UC006 Class Diagram



UC006 Sequence Diagram



3.3. Layered Architecture

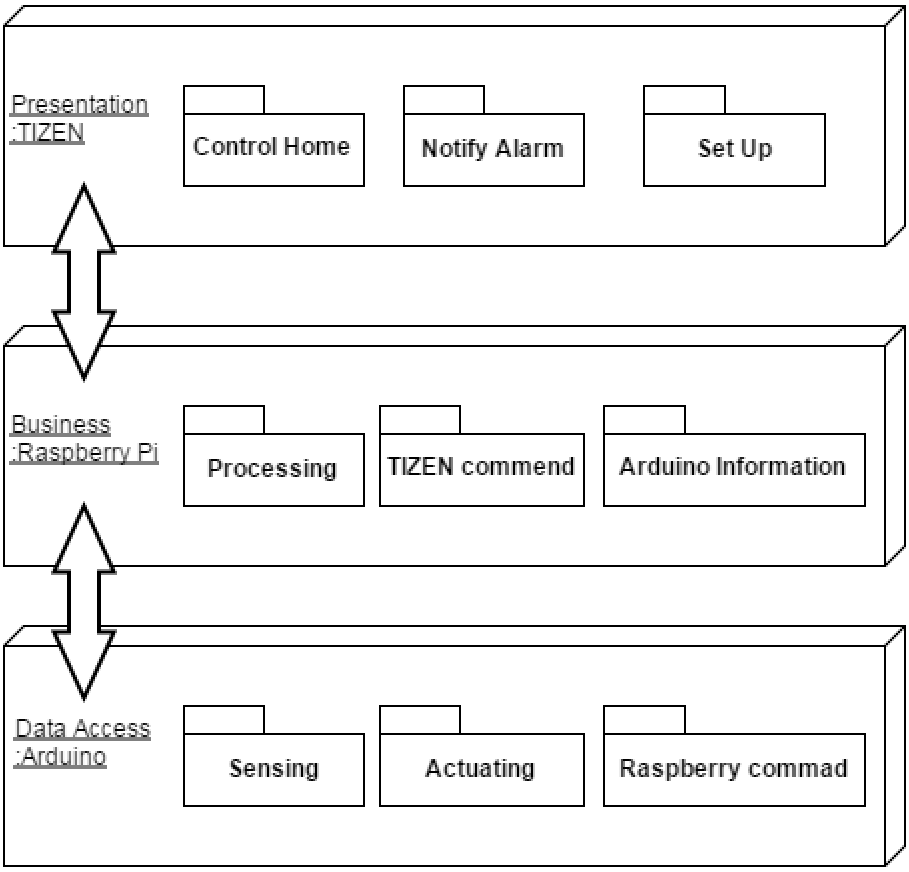


Figure 13. Layered Architecture of Smart Home Guard System

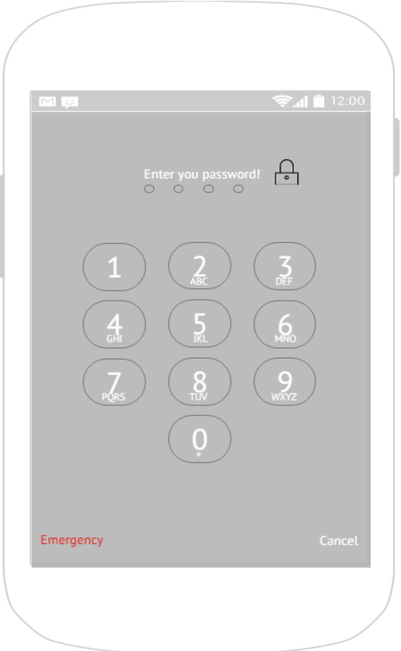
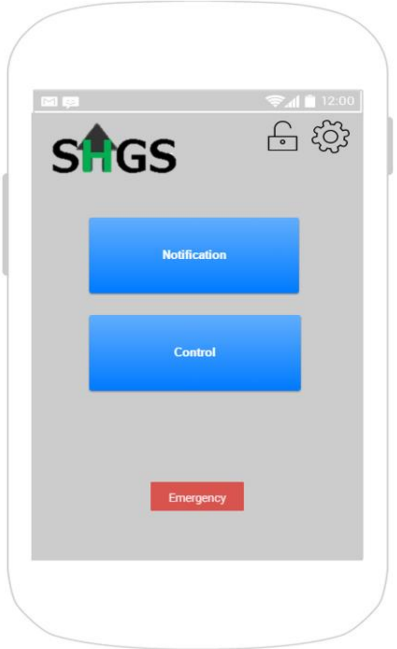
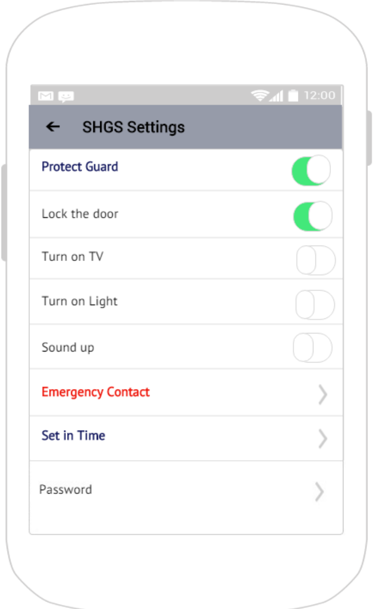
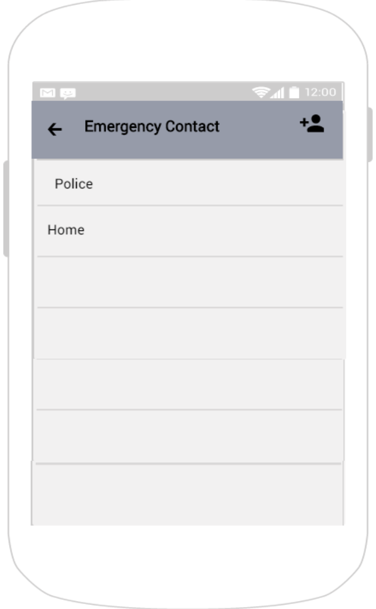
Figure 13 shows the layered architecture of SHGS. Each layer represents presentation, business and data access which are represented for TIZEN, Raspberry Pi, and Arduino, respectively. In each layer, packages are already explained before sections. Additionally, we draw package diagram using draw.io [6].

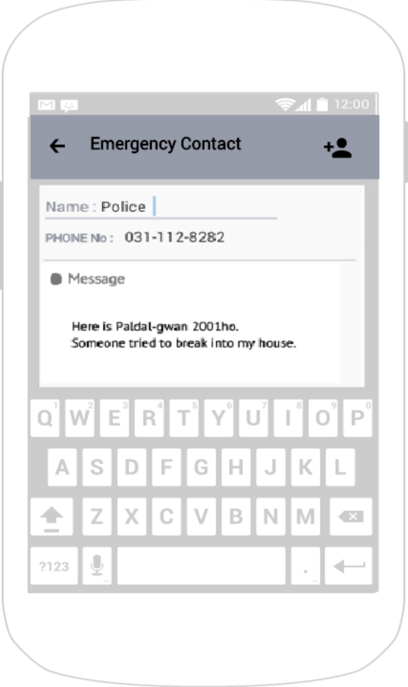
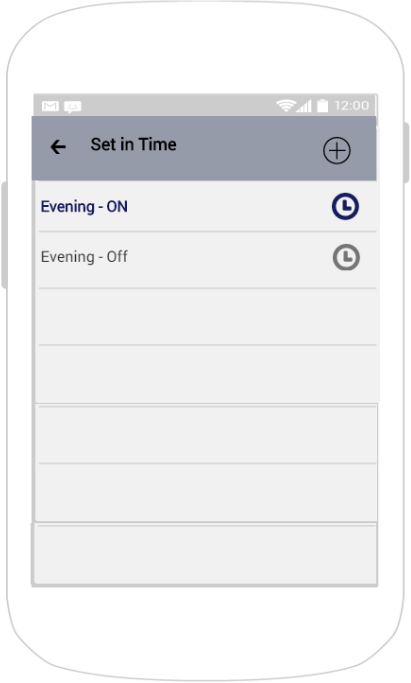
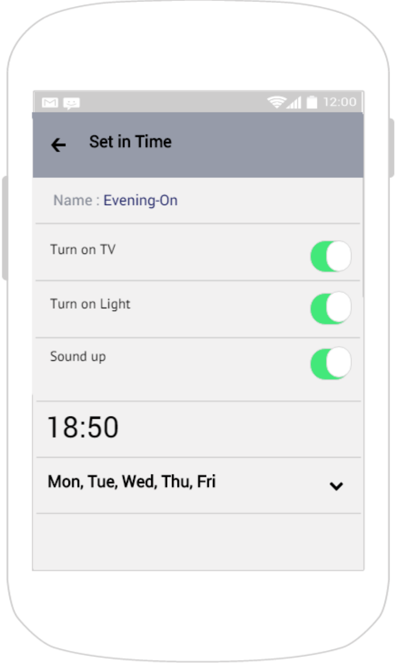
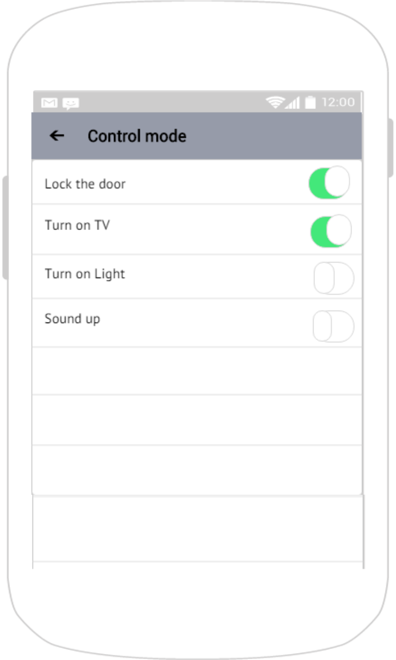
4. User Interface Design

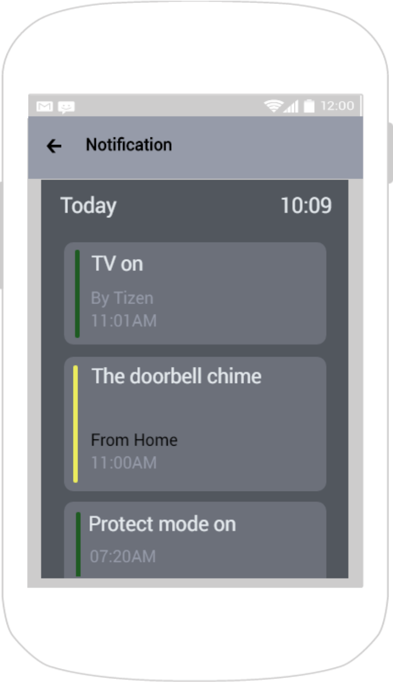
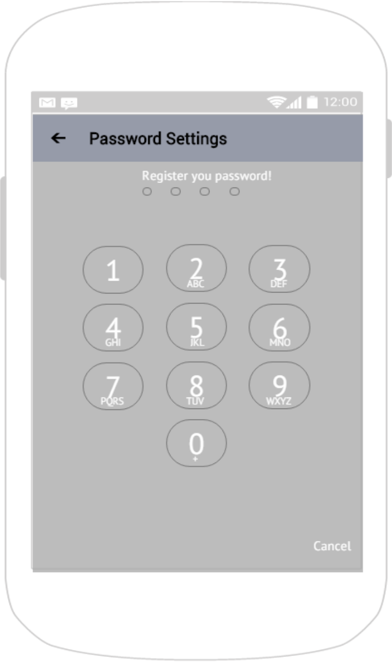
We were used UI mockup tool [8] for UI design. It divided the Main Menu, SHGS settings and sub-menu.

4.1. TIZEN UI

Table 2. TIZEN UI Design

Lock release by password input	Main Menu
 12:00 Enter you password! 1 2 ABC 3 DEF 4 GHI 5 JKL 6 MNO 7 PQRS 8 TUV 9 WXYZ 0 Emergency Cancel	 12:00 SHGS Notification Control Emergency
SHGS Settings	**Emergency Contact List**
 12:00 SHGS Settings Protect Guard Lock the door Turn on TV Turn on Light Sound up Emergency Contact Set in Time Password	 12:00 Emergency Contact Police Home

Emergency contact - Input	Set in Time List
 12:00 Q W E R T Y U I O P A S D F G H J K L Z X C V B N M ?123 Emergency Contact Name : Police PHONE No : 031-112-8282 Message Here is Paldal-gwan 2001ho. Someone tried to break into my house.	 Set in Time Evening - ON Evening - Off
Set in Time - Input	**Control**
 Set in Time Name : Evening-On Turn on TV Turn on Light Sound up 18:50 Mon, Tue, Wed, Thu, Fri	 Control mode Lock the door Turn on TV Turn on Light Sound up

Notification	Password Setting
 12:00 Notification Today 10:09 TV on By Tizen 11:01AM The doorbell chime From Home 11:00AM Protect mode on 07:20AM	 12:00 Password Settings Register you password! 1 2 ABC 3 DEF 4 GHI 5 JKL 6 MNO 7 PQRS 8 TUV 9 WXYZ 0 + Cancel

We will develop the TIZEN app to using UI shown here.

5. Circuit Diagram

In SHGS, we use various sensors and actuators. For smooth development at implementation phase, we made a circuit diagram that shows board and each of connected sensor and actuator. Additionally, we draw package diagram using 123d Circuits [9].

5.1. Arduino Circuit Diagram

Arduino use RGB LED(2,3,4 pin), Buzzer(5 pin) as TV function, LED(6 pin) as light function, Servo Motor(12 pin) as locking door function, Buzzer(10 pin) as warning function, pressure sensor(A0 pin) as detecting trespassing, PIR motion sensor(11pin) as moving detection.

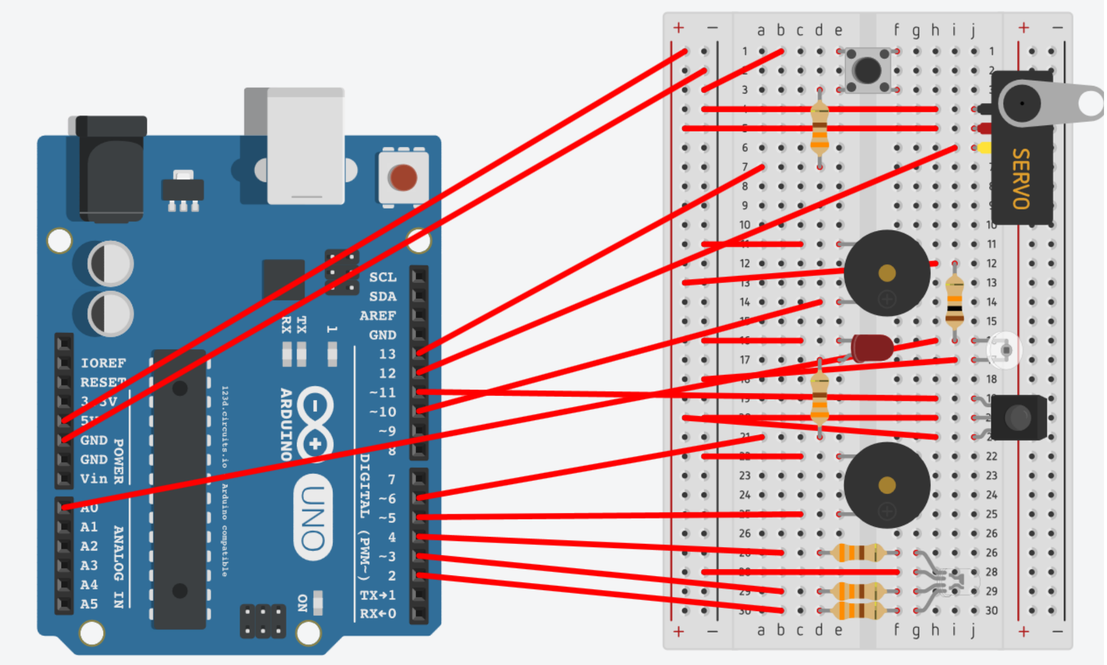


Figure 14. Arduino Circuit Diagram-Lab View

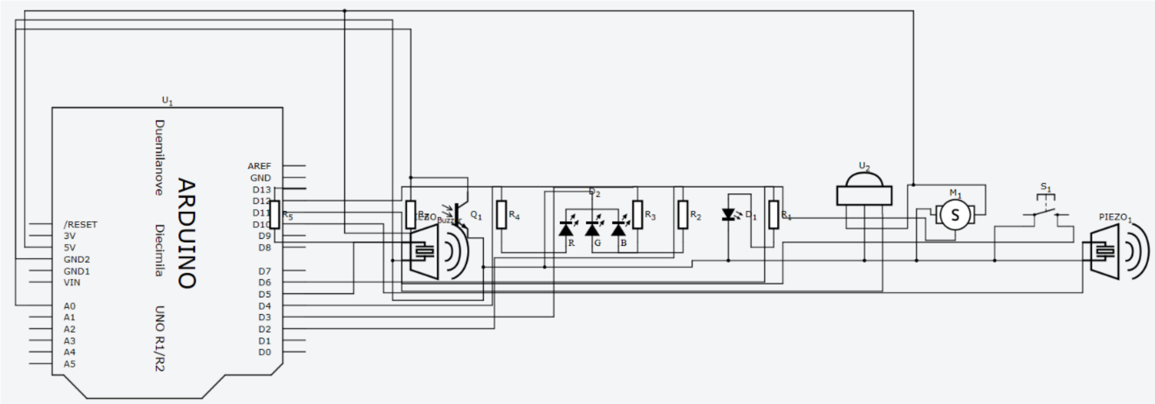


Figure 15. Arduino Circuit Diagram-Schematic View

5.2. Arduino and Raspberry Pi Connection Diagram

Board uses RX and TX to process data between Raspberry Pi and Arduino.

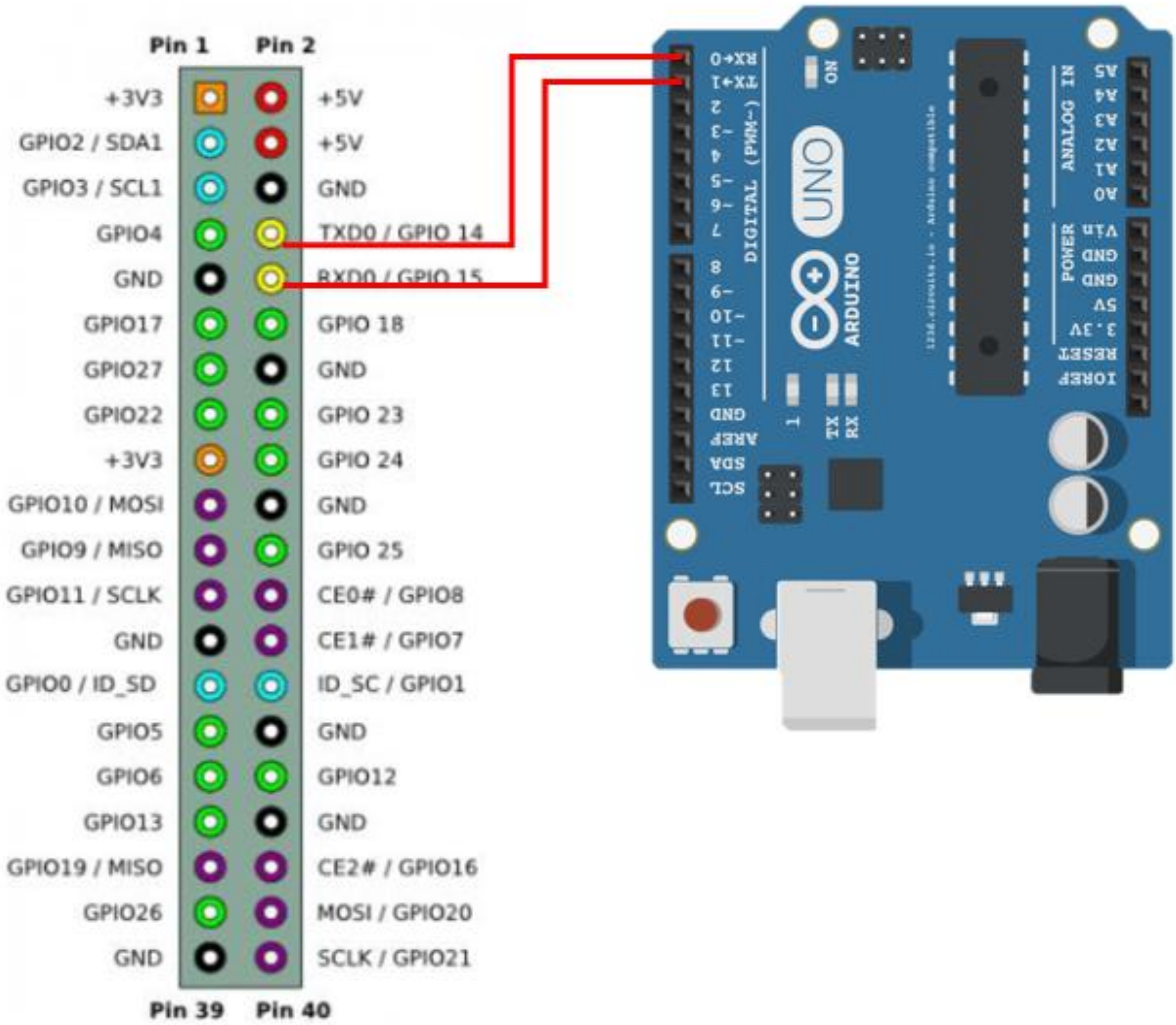


Figure 16. Arduino and Raspberry Pi2 Connection Diagram

6. Verification and Validation Results

In this section, we describe the validation process of the actual implementation from verification of the requirement of SHGS.

6.1. Testing Strategy

In section 6.2, we verify that SHGS developed as a requirement specification based on the checklist. Section 6.3, validate the developed system with unit testing and integration testing. SHGS are many sensors and actuators are connected to each Device. So, we proceed to unit testing for each module. After that, we tested interaction between each devices and integration function.

6.2. Verification of Requirements

We verify that SHGS has developed as correct requirement by the below Checklist.

6.2.1. Functional Requirements Checklist

It enumerates the crucial functions for SHGS.

Table 3. Functional Check List

Category	TIZEN	**Test ID**	FC_001	**Requirement ID**	TIZEN_Req
No	Scenario	Response (Y, N, N/A)	Tester	Description	Note
1	TIZEN should accept notification well		Byun		
2	TIZEN should have secure network communication function by using 3G	N/A	Byun	couldn't use 3G	
3	Only the authorized owner can TIZEN		Byun		
4	There should not be any delay of video streaming	Y	Lee		
Category	Raspberry Pi	**Test ID**	FC_002	**Requirement ID**	Rasp_Req
No	Scenario	Response (Y, N, N/A)	Tester	Description	Note
1	Raspberry should process all the data in a second	Y	Lee		

2	Raspberry should be running 24 hours(always)	Y	Lee		
3	Raspberry should run in a consecutive time more than 24 hours	Y	Lee		
Category	Arduino Uno	**Test ID**	FC-003	**Requirement ID**	Ardu_Req
No	Scenario	Response (Y, N, N/A)	Tester	Description	Note
1	Arduino should exactly detect the motion	Y	Lim		
2	There should not be any error that cause emergency alarm	Y	Lim		
3	Arduino should run in a consecutive time more than 24 hours	Y	Lim		

6.2.2. Qualitative Requirements Checklist

It enumerates the crucial qualities for SHGS.

Table 4.Qualitative Check List

Evaluation Item	**Item to Examine**	**Score**					**Note**
		5	**4**	**3**	**2**	**1**	
Understandability (5)	✓ Can readers of the document understand what the requirements mean?	O					
Redundancy (5)	✓ Is information unnecessarily repeated in the requirements document?	O					
Completeness (5)	✓ Does the checker know of any missing requirements or is there any information missing from individual requirement descriptions?	O					
Ambiguity (10)	✓ Are the requirements expressed using terms which are clearly define?	O					
	✓ Could readers from different backgrounds make different interpretations of the requirements?	O					
Consistency (10)	✓ Do the descriptions of different requirements include contradictions?		O				
	✓ Are there contradictions between individual requirements and overall system requirements?		O				
Organization (10)	✓ Is the document structured in a sensible way?	O					
	✓ Are the descriptions of requirements organized so that related requirements organized so that related requirements are grouped?	O					
Conformance to standards (10)	✓ Does the requirements document and individual requirements conform to defined standards?		O				
	✓ Are departures from the standards, justified?		O				

Traceability (5)	✓	Are requirements unambiguously identified, include links to related requirements and to the reasons why these requirements have been included?	O					
			Total Score :					**56/ 60**
Total Discussion :								

6.2.3. Use Case Checklist

Table 5.Use Case Check List

1. **The use-case name is meaningful and un-ambiguous**
[0] 1.1 Does the use case have a unique name? [0] 1.2 Is the name a verb + noun phrase (for example, Withdraw Cash)? [0] 1.3 Does the name accurately summarize the main goal of the use case? [0] 1.4 Is the name "actor independent"?
2. **The brief description clearly describes the primary goal of the use case**
[0] 2.1 Is it clear from the brief description what the main purpose of the use case is? [0] 2.2 Is the "observable result of value" obvious?
3. **Associated actors and information exchanged are clearly defined**
[0] 3.1 Is the use case associated with one or more actors? [0] 3.2 Is the primary, or initiating actor, defined? [0] 3.3 Is it clear who wishes to perform the use case? [0] 3.4 Is all information exchanged between the actor(s) and the system clearly specified? [0] 3.5 If a "time" actor is used, are you sure you did not miss an important actor and associated use cases (such as administrative or maintenance personnel that define schedule events)?
4. **Pre-conditions have been specified**
[0] 4.1 Does each pre-condition represent a tangible state of the system (for example, the Withdraw Cash use case for an automated teller machine has a precondition that the user has an account)?
5. **The Basic Flow and Alternate Flows are complete, correct and consistent**
[0] 5.1 Is it clear how the use case is started? [0] 5.2 Is the triggering event clearly described? [0] 5.3 Does the flow have a definite ending?

[X] 5.4 Does each step in the scenario contain the same level of abstraction?

[0] 5.5 Does each step in the scenario describe something that can actually happen and that the system can reasonably detect?

[0] 5.6 Does each step make progress towards the goal?

[0] 5.7 Are there any missing steps? Is it clear how to go from one step to the next? Does the sequence of communication between the actors and the use case conform to the user's expectations?

[0] 5.8 Does each step describe how the step helps the actor achieve their goal?

[0] 5.9 Is each step technology independent? Is it free of technical details, and design decisions?

[0] 5.10 Are the steps correctly numbered?

[0] 5.11 For each alternate flow is the condition(s) for initiation of the flow clearly defined?

[0] 5.12 For each alternate flow is it clear how the use case ends or where in the basic flow that the use case resumes?

6. Post-conditions have been specified

[0] 6.1 If "Minimal Guarantees" are present, do they always happen when the use case completes, regardless of success? (A Minimal Guarantee represents a condition that will be true when the use case ends, regardless of how it terminates.)

[0] 6.2 If "Success Guarantees" are present, do they always happen when the use case completes successfully? (A Success Guarantee represents a condition that will be true when the use case ends successfully, regardless of which path it takes.)

7. Applicable non-functional requirements have been captured

[0] 7.1 Are non-functional requirements (such as performance criteria) that are applicable to the use case captured in the use case?

[0] 7.2 Are these non-functional requirements applicable to many use cases? It they are, consider capturing them in the system-wide requirements specification to simplify maintenance.

6.3. Validation of Implementation

6.3.1. Unit Testing

TIZEN and Raspberry Pi connected by RX/TX and we check the operation to actuator by control command.

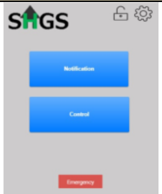
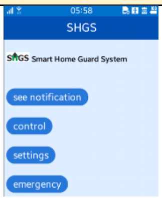
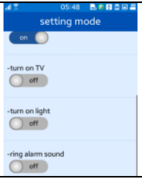
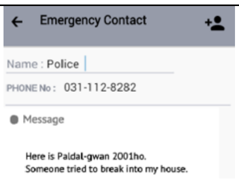
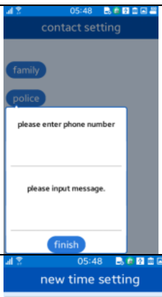
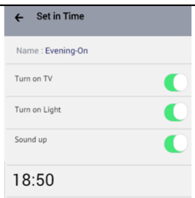
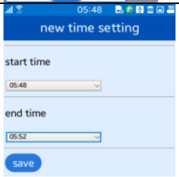
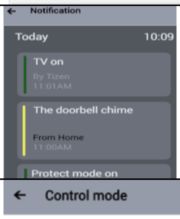
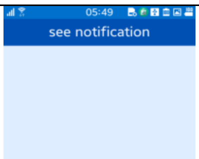
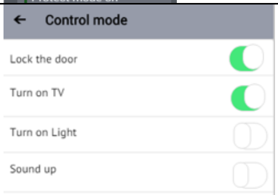
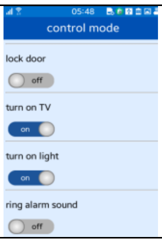
Table 6. Unit Testing - Test Case

Test Case ID	Action/ Unit	Expected Results	Actual Results	Pass/Fail
Arduino_1	TV operation - RGB LED, Buzzer	If "TV ON" command from the Raspberry Pi, RGB LED is on and sound is generated from the buzzer.	RGB LED is turned on and Piezo buzzer is play the melody.	PASS
Arduino _2	Light operation – LED	If "Light ON" command from the Raspberry Pi, LED is on.	LED is turned on.	PASS
Arduino _3	DoorLock operation - Servo Motor	If control the Doorlock, it will operate.	Doorlock is moving of 90 degrees by control.	PASS
Arduino _4	Warning operation - Buzzer	If "Warning" command from the Raspberry Pi, Beep occurs.	Beep occurred at the buzzer.	PASS
Arduino _5	Recognition of Pressure sensor	Detect the value, if press the pressure sensor (value>150, Intrusion attempts)	It detected press value, and beep occurs.	PASS
Arduino _6	Recognition of PIR motion sensor	Detect the motion by PIR Motion sensor, beep occurs.	It detected motion and beep occurs.	PASS
Arduino _7	Recognition of Doorbell – Button	Generating input value when press the Button.	Press the Button, Raspberry Pi is recognized the input value	PASS
Rasp_1	Camera module operation	Check the Video to the web page	Checked the video to the web page	PASS

6.3.2. UI Testing

We tested UI on TIZEN app. There was a change when applied to TIZEN. Because it was Change the control button the left and right on SHGS settings menu designed for and Android environment.

Table 7. UI Testing - Test Case

Test Case ID	UI	Expected Results	Actual Results	Pass/Fail
UI_1	Check the Main menu UI	 SHGS Notification Control Emergency	 05:58 SHGS SHGS Smart Home Guard System see notification control settings emergency	Pass
UI_2	Change the control button the left and right on SHGS settings menu.	Check the UI, if change the control button the left and right on SHGS settings menu.	 05:48 setting mode on -turn on TV off -turn on light off -ring alarm sound off	Pass
UI_3	Check the value you changed in UI_2	Make sure maintain the value you changed in UI_2		Fail
UI_4	Check the "Add emergency Contact list"	 Emergency Contact Name : Police PHONE No : 031-112-8282 Message Here is Paldal-gwan 2001ho. Someone tried to break into my house.	 05:48 contact setting family police please enter phone number please input message. finish	Fail
UI_5	Check the "Add time settings"	 Set in Time Name : Evening-On Turn on TV Turn on Light Sound up 18:50	 05:48 new time setting start time 05:48 end time 05:52 save	Fail
UI_6	Main Menu -> Notification	 Notification Today 10:09 TV on By Tizen 11:01AM The doorbell chime From Home 11:00AM Protect mode on	 05:49 see notification	Pass
UI_7	Main menu -> Control	 Control mode Lock the door Turn on TV Turn on Light Sound up	 05:48 control mode lock door off turn on TV on turn on light on ring alarm sound off	Pass

6.3.3. Integration Testing

We tested that operation in Arduino, Raspberry Pi by Control in TIZEN APP. And we tested that receiving normal value from the Arduino sensor at TIZEN app.

Table 8. Integration Testing - Test Case

Test Case ID	Function	Expected Results	Actual Results	Pass/Fail
Tizen_Set1	Set the password used to authenticate the app.	The password is set properly.	-	Fail
Tizen_Set2	It can run the app after enter the password.	Entered correct password then, it can approach the Main menu.	-	Fail
Tizen_Set3	Set the actuator operation and time in "set in time".	It operated as control set at the specified time.	-	Fail
Tizen_Set4	Enter the contact number on the "Emergency contact menu".	Contact number and the message is entered correctly.	-	Fail
Tizen_Set5	Press the Emergency button for 2 seconds or more.	The emergency message is sent to set up contact number.	-	Fail
Tizen_Set6	Lock the door from the control mode menu.	The door is locked by the servo motor rotation.	When user send the command 'lock the door', the servo motor rotate 180degree.	Pass
Tizen_Set7	Open the door from the control mode menu.	The door is opened by the servo motor rotation.	When user send the command 'lock the door', the servo motor rotate 180degree.	Pass
Tizen_Set8	Turn on the TV from the control mode menu.	RGB LED is turned on and Piezo buzzer is play the melody.	When user send the command 'turn on the TV', the RGB LED and Piezo buzzer is on.	Pass
Tizen_Set9	Turn off the TV from the control mode menu.	RGB LED is turned on and Piezo buzzer is play the melody.	-	Fail
Tizen_Set10	Turn on the Light from the control mode menu.	LED is turned on.	When user send the command 'turn on the	Pass

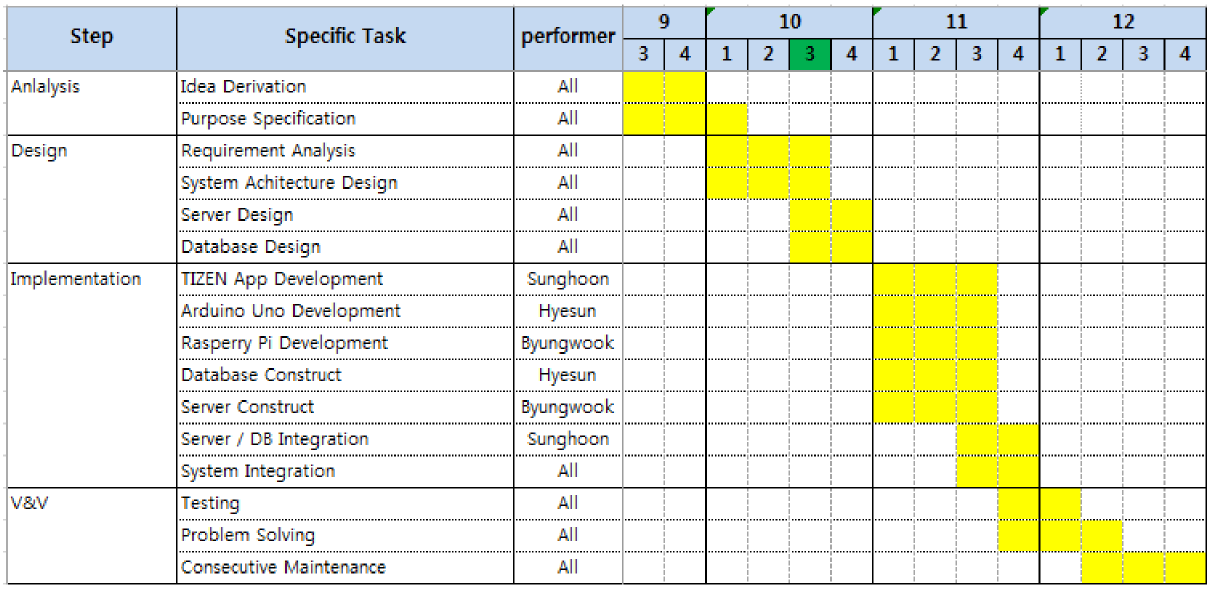
			light', the LED is on.	
Tizen_Set11	Turn off the Light from the control mode menu.	LED is turned off.	When user send the command 'turn on the light', the LED is off.	Pass
Tizen_Set12	Turn on the Sound from control mode menu.	Beep occurred at the buzzer.	-	Fail
Arduino _8	Press the pressure sensor on Arduino.	a. Check the "External intrusion" message in the Notification menu. b. The emergency message is sent to set up contact number.	When the pressure sensor detect, "danger" message is printed on the client.	a. Pass b. Fail
Arduino _9	Detected the movement to PIR Motion sensor.	a. Check the "External intrusion" message in the Notification menu. b. The emergency message is sent to set up contact number.	When the PIR motion sensor detect, "danger" message is printed on the client.	a. Pass b. Fail
Arduino _10	Press the doorbell.	Check the "Pressing the doorbell" message in the Notification menu.	When the switch is clicked, "someone come" message is printed on the client and also camera module is executed.	Pass
Rasp_2	Press the doorbell from Arduino, the camera module is operating.	a. Check the "Visitor" message in the Notification menu b. If it click the "OK" button, It connected Video on the Web Page.	When the switch is clicked, "someone come" message is printed on the client and also camera module is executed.	Pass

7. Project Summary and Conclusion

We had opportunity to learn new technologies that was TIZEN, Arduino, Raspberry-Pi, Node-js, Java script. We tried to follow the Software Engineering process.

We divided the work into respective task by using Work Breakdown Structure based on Gantt chart

Table 9. Work break Structure of SHGS Project



Step	Specific Task	performer	9		10				11				12			
			3	4	1	2	3	4	1	2	3	4	1	2	3	4
Anlalysis	Idea Derivation	All	🟨	🟨												
	Purpose Specification	All	🟨	🟨	🟨											
Design	Requirement Analysis	All			🟨	🟨	🟨									
	System Achitecture Design	All			🟨	🟨	🟨									
	Server Design	All					🟨	🟨								
	Database Design	All					🟨	🟨								
Implementation	TIZEN App Development	Sunghoon							🟨	🟨	🟨					
	Arduino Uno Development	Hyesun							🟨	🟨	🟨					
	Rasperry Pi Development	Byungwook							🟨	🟨	🟨					
	Database Construct	Hyesun							🟨	🟨	🟨					
	Server Construct	Byungwook							🟨	🟨	🟨					
	Server / DB Integration	Sunghoon									🟨	🟨				
	System Integration	All									🟨	🟨				
V&V	Testing	All										🟨	🟨			
	Problem Solving	All										🟨	🟨	🟨		
	Consecutive Maintenance	All												🟨	🟨	🟨

In Proposal phase, we have tried to propose a project to help people. Thus, we proposed SHGS which is Home Guard System for Single Household's safety and comfortable

In Requirement specification phase, we elicit the requirement from various stakeholders. We have designed our requirement with the UML. We represented by Goal graph, Scenario Flow Chart, Use Case Diagram, Use Case Description. And we verified requirement by Checklist, Traceability Table.

In Design phase, we designed interaction between each device, User Interface, Circuit Diagram, Class Architecture.

In Implementation Phase, we have met technical issues from TIZEN. We have no TIZEN development experience. In addition, we couldn't find reference material what we need. We have reduced the implementation of overlapped use case.

8. Reference

[1] STATISTICS KOREA, "Current situation of live alone," 9 2015. [Online]. Available: https://kostat.go.kr/portal/korea/kor_nw/2/1/index.board?bmode=read&aSeq=269194&pageNo=&rowNum=10&amSeq=&sTarget=&sTxt=. [Accessed 19 9 2015].

[2] S.-e. Kang, "Digital Times," Economic Daily, 8 9 2015. [Online]. Available: http://www.dt.co.kr/contents.html?article_no=2015090802109960800004. [Accessed 9 2015].

[3] STATISTICS KOREA, "Statistics Blog," 4 2015. [Online]. Available: http://blog.naver.com/hi_nso/220337287919. [Accessed 9 2015].

[4] M. Li and H.-J. Lin, "Design and Implementation of Smart Home Control Systems Based on Wireless Sensor Networks and Power Line Communications," in *Industrial Electronics, IEEE Transactions*, 2014.

[5] MKLab, Co., *StarUML,* MKLab, Co., 2015.

[6] JGraph Ltd, *draw.io,* JGraph Ltd.

[7] Hanov Solutions Inc., "Web sequence diagrams," [Online]. Available: https://www.websequencediagrams.com/#. [Accessed 11 2015].

[8] UXPin Sp., *UXPin,* UXPin Sp..

[9] 123d Circuits, "123d Circuits," [Online]. Available: https://123d.circuits.io/. [Accessed 11 2015].