Design Modeling Document
For the SafeHome Product Software

Prepared By

Team 3

Eut-deum Kim

Myung-kyung Lee

Ji-seong Gu

Da-bi Ahn

Issued at

2008. 4. 28.

Version 1.0.0

This document is inspected and approved by persons who signed below. Therefore, this document will be used as the basis for later process. This document can't be modified or supplemented without approval signature.

Signature :

Name : Moon-zoo Kim

Title : Client

Date : 2008. 4. 28.

Signature :

Name : Eut-deum Kim

Title : Team Member

Date : 2008. 4. 28.

Signature :

Name : Myung-kyung Lee

Title : Team Member

Date : 2008. 4. 28.

Signature :

Name : Ji-seong Gu

Title : Team Member

Date : 2008. 4. 28.

Signature :

Name : Da-bi Ahn

Title : Team Member

Date : 2008. 4. 28.

Signature :

Name :

Title :

Date :

Signature :

Name :

Title :

Date :

Document History

Issue	Date	Change Reason	Author
01	2008. 4. 18.	Building a rough draft Adding architecture description	Team 3
02	2008. 4. 20.	Adding a deployment diagram Adding class extracting description	Team 3
03	2008. 4. 23.	Modifying a deployment diagram Adding a class diagram Adding CRC cards Modifying classes	Team 3
04	2008. 4. 25.	Modifying CRC cards Modifying classes Adding state diagrams	Team 3
05	2008. 4. 26.	Modifying state diagrams Adding elaborated classes	Team 3
06	2008. 4. 27.	Modifying state diagrams Adding a software architecture diagram	Team 3
07	2008. 4. 28.	Final review and modification	Team 3

TABLE OF CONTENT

System Overview

1. External Components

- Before reporting our SafeHome Product software design modeling, we introduce our architecture design for understanding.
- The composition of our SafeHome Product design is like below.

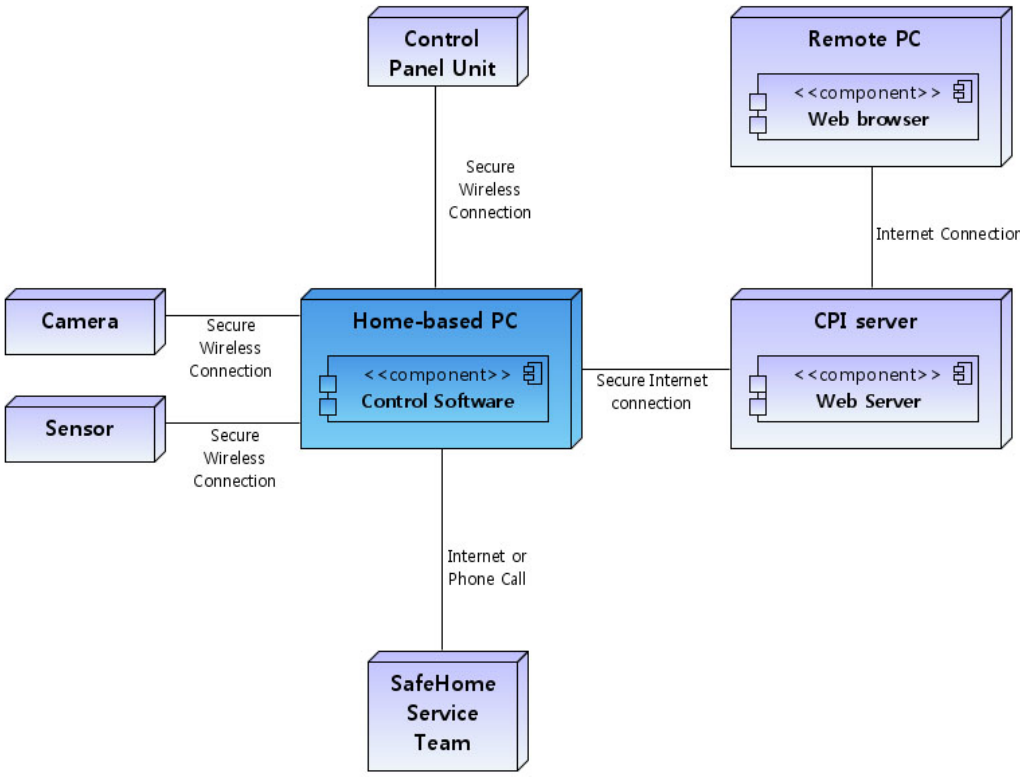


Figure 1-1. Deployment diagram for SafeHome Product

- SafeHome Product main system is composed of home-based PC, CPI server and Control Panel. Main control software operates in home-based PC.
- Control Panel, cameras and sensors are connected to home-based PC via wireless connection. When alarm condition encountered, home-based PC calls SafeHome Service team via phone. Home-based PC and CPI server are connected via secured Internet connection.
- There are two way to access SafeHome Product from external - via Control Panel and CPI server. The homeowner can use SafeHome security function via Control Panel at home or use all SafeHome functions via web browser of remote PC.

2. Software Architecture

2.1 Software Architecture Diagram

- As you see before, SafeHome Product is controlled by control software on home-based PC. All external accesses and hardware control are executed by control software.
- Internet account data(ID, passwords, permission) is stored in CPI server and other data is stored in home-based PC.
- From this feature, software architecture can be derived like below.

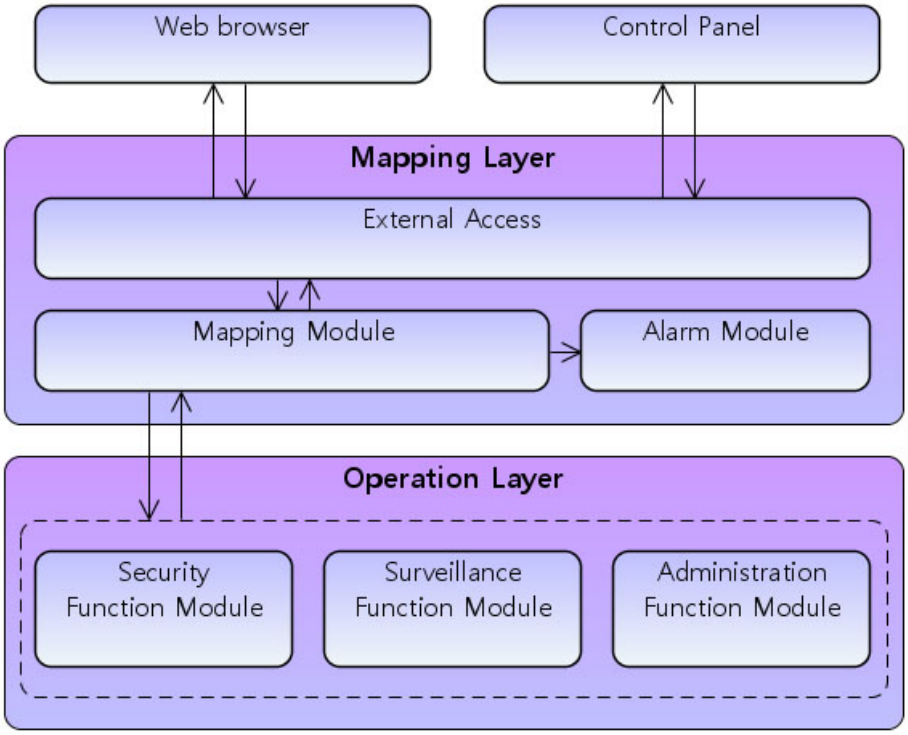


Figure 1-2. Software architecture diagram for SafeHome Product

- "Web browser" and "Control Panel" mean external interfaces. The homeowner can use SafeHome functions via Internet or Control Panel.
- When the interfaces send control signal to SafeHome Product software, mapping layer receive, decode it and map to appropriate function in operation layer. Mapping layer also includes alarm control module. The reason that alarm module is placed in mapping layer is that alarm condition can be encountered at any time and affects entire system, so it is reasonable that alarm module is included in mapping layer.

2.2 Advantage of this design

- Adding any new interface is much easier. New interface just has to have appropriate standard function interface.
- Adding any new function is easier. A developer just inserts new function in Operation Layer as class and modify mapping function in Mapping Layer.
- Because Mapping Layer encode and decode control signal and map into each function

module, each function module in operation layer can have low coupling, high cohesion.

- All external accesses must pass through Mapping Layer, so if encode/decode functions are well defined, they can guarantee the security of communications among system, interface, devices.

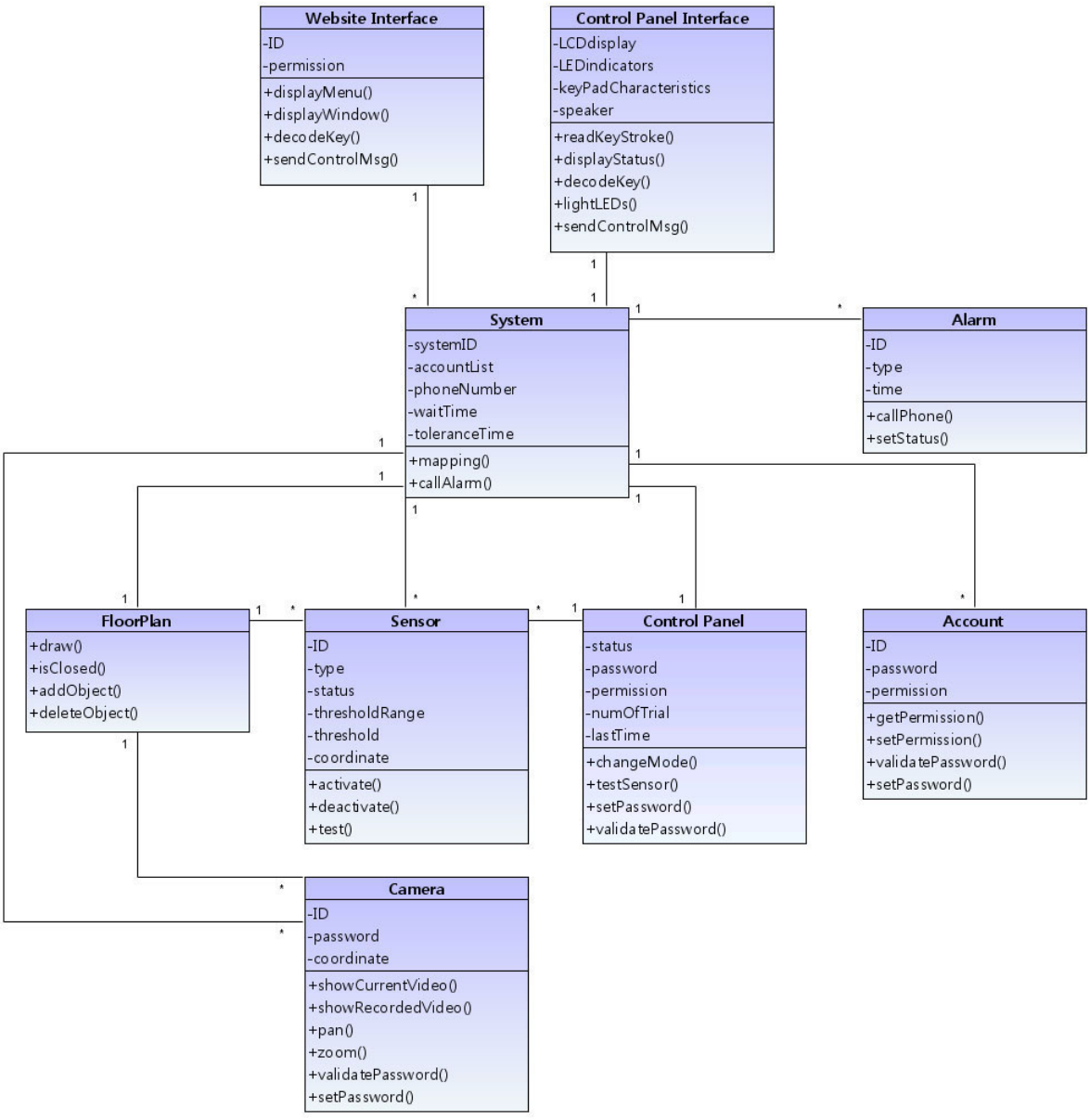
2.3 Disadvantage of this design

- Mapping Layer can be complex and have lower maintainability. But it seems to be conquered by well-designed mapping function like multi-level mapping.
- All external accesses like communication with interface, signals from sensors must pass through encode/decode function of Mapping Layer. It can consume more time and resources.

Class Design

1. Class Analysis

1.1 Overview - Class Analysis Diagram



- Our whole analysis class diagram is like this. Extracting steps of each class are described in next chapter.
- System and Alarm class act as Mapping Layer. System class can call the plural number of

Alarm class, so there are one-to-many relationship.

- Floor plan includes all diagrams of cameras and sensors, so there are one-to-many relationships.
- Control panel can control all sensors of the system, so there is one-to-many relationship.
- The plural number of the homeowner, guests can connect to system via web browser, so there is one-to-many relationship.
- Each system has only one control panel and floor plan, so there are one-to-one relationships.
- Each system can have many accounts, sensors and cameras, there are one-to-many relationships.

1.2 Class Extracting Description

1.2.1. Applied principles

- We classify all functionalities in three categories - surveillance, security, administration - and extract potential classes from each categories using principles - grammatical parsing, categorizing analysis classes into 7 types(External entities, Things, Occurrences or events, Roles, Organizational units, Places, Structures) and six selection characteristics(Retained information, Needed services, Multiple attributes, Common attributes, Common operations, and Essential requirements) - in textbook.

- After that, we discuss about which ones are more essential than other. Finally, we can pick out 7 classes : Floorplan, Alarm, Account, Camera, Control Panel, Sensor, and System.

- After finishing this work, we get to define operations and attributes of them. We find each classes' attributes referring to the first work done when we extract potential classes. We also do the work based on the following principles in text book. At first, we extract verbs from use-case descriptions. And the next step is that categorize all corresponding verbs to general broad categories of operations:

  (1) operations that manipulate data in some way
  (2) operations that perform a computation
  (3) operations that inquire about the state of an object
  (4) operations that monitor an object for the occurrence of a controlling event

1.2.2. Classes about Security Functions

1.2.2.1 Making Class - Grammatical Parse & Applying Six Selection Characteristics

Potential class	General classification	Characteristic number that applies
homeowner	role / external entity	reject (1, 2 fail)
guest	role / external entity	reject (1, 2 fail)
sensor	external entity	accept
control panel	external entity	accept
picture	interface	reject ⇒ function of interface
installation	occurrence	reject
system	thing	accept
floor plan	thing	accept
ID, type	attributes of sensor	reject ⇒ attribute of sensor
tolerance time	attribute of sensor	reject ⇒ attribute of sensor
master password	thing	reject ⇒ attribute of CP
secondary passwords	thing	reject ⇒ attribute of CP
telephone number	thing	reject ⇒ attribute of system
sensor event	occurrence	reject ⇒ function of sensor
audible alarm	external entity	accept ⇒ omitted in discussion
monitoring service	organizational unit or external entity	reject (1, 2 fail)

1.2.2.2 Making Operations - Grammatical Parse & classification

operation	operation type
activate sensor	(1) manipulate data in some way
deactivate sensor	(1) manipulate data in some way
test sensor	(3) inquire about the state of an object
validate control panel password	(2) perform a computation
change master password	(1) manipulate data in some way
add secondary password	(1) manipulate data in some way

1.2.3. Classes about Surveillance Functions

1.2.3.1 Making Classes - Grammatical Parse & Applying Six Selection Characteristics

Potential class	General classification	Characteristic number that applies
camera password	attribute of camera	reject ⇒ attribute of camera
camera	external entities	all apply
the system	things	all apply
internet ID	things	all apply
internet password	attribute of internet ID	reject ⇒ attribute of ID
alarm condition	occurrences or events	all apply
pan	attribute of camera	reject ⇒ attribute of camera
zoom	attribute of camera	reject ⇒ attribute of camera
website	places	all apply
thumbnail	organizational units	reject : 1 fails
current video	things	reject : 1, 5 fails
locked video	things	reject : 5, 6 fails
recorded video	things	reject : 4, 5 fails
camera icon	attribute of floorplan	reject ⇒ attribute of floorplan
floorplan	things	all apply
permission	attribute of internet ID	reject ⇒ attribute of internet ID
range	attribute of camera	reject ⇒ attribute of camera
guest	roles or external entities	reject : 1,2 fails
homeowner	roles or external entities	reject : 1,2 fails

1.2.3.2 Making Operations - Grammatical Parse & classification

operation	operation type
select camera	(1) manipulate data in some way
display current video	(1) manipulate data in some way
display thumbnail	(1) manipulate data in some way
zoom	(1) manipulate data in some way
pan	(1) manipulate data in some way
display recorded video	(1) manipulate data in some way
screening camera view	(1) manipulate data in some way
lock/unlock	(2) perform a computation
lock/unlock video	(2) perform a computation

1.2.4. Classes about Administration Functions

1.2.4.1 Making Classes - Grammatical Parse & Applying Six Selection Characteristics

Potential class	General classification	Characteristic number that applies
administrator	roles or external entities	reject : 1,2 fails
ID	things	all apply
permission	attribute of ID	reject ⇒ attribute of ID
Website	places	all apply
homeowner	roles or external entities	reject : 1,2 fails
administration	kind of permission	reject
system	things	all apply
menu	attribute of website	reject ⇒ attribute of website
window	attribute of website	reject ⇒ attribute of website
button	attribute of website	reject ⇒ attribute of website
alarm condition	events	all apply
Sensor	external entities	all apply
Threshold	attribute of sensor	reject ⇒ attribute of sensor
intrusion	kind of sensor	all apply
alarm	threshold for some of sensor	reject ⇒ attribute of sensor
away	threshold for some of sensor	reject ⇒ attribute of sensor
CO sensor	kind of sensor	all apply
smoke sensor	kind of sensor	all apply
motion-tracking sensor	kind of sensor	all apply
door sensor	kind of sensor	all apply
window sensor	kind of sensor	all apply
water level sensor	kind of sensor	all apply
floorplan	things	all apply
camera	external entities	all apply
grid	attribute of floorplan	reject ⇒ attribute of floorplan
wall	things	all apply
door	things	all apply
window	things	all apply
passwords	attribute of ID	reject ⇒ attribute of ID
guest	roles or external entities	reject : 1,2 fails

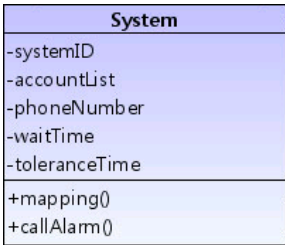
1.2.4.2 Making Operations - Grammatical Parse & classification

operation	operation type
validate IDs/passwords	(2) perform a computation
change mode	(1) manipulate data in some way
select menu	(1) manipulate data in some way
display window	(1) manipulate data in some way
store id's data	(1) manipulate data in some way
change permission	(1) manipulate data in some way
get permission data	(3) inquire about the state of an object
delete id's data	(1) manipulate data in some way
encounter alarm condition	(4) monitor an object for the occurrence of a controlling event
control threshold	(1) manipulate data in some way
store threshold value	(1) manipulate data in some way
draw floorplan	(1) manipulate data in some way
store floorplan	(1) manipulate data in some way
check closed	(2) perform a computation

2. CRC Index Cards

2.1 System Class

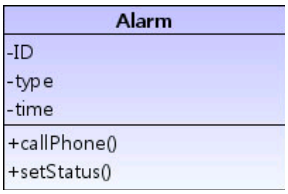
• System class receives and sends control message, maps the message into appropriate functions in Operation Layer and call alarm class when alarm condition encountered.



Class : System	
Description : This class provides mapping delivered message to fitting functions in other class.	
Responsibility :	**Collaborator :**
Defines system ID.	
Maps interface message to functions.	
Activates alarm.	Alarm, Website Interface, Control Panel Interface
Communicates with interface.	Website Interface, Control Panel Interface

2.2 Alarm Class

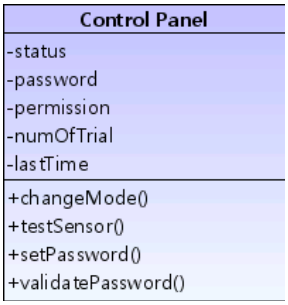
• When alarm condition encountered, System class calls Alarm class. Alarm class calls SafeHome Service team and the homeowner and change the specific sensor's status to "Alarm".



Class : Alarm	
Description : This class provides informations about what system should do when alarm condition encounters.	
Responsibility :	**Collaborator :**
Makes phone calls.	
Sets sensor's status.	Sensor

2.3 Control Panel Class

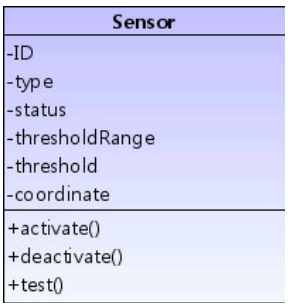
• Control Panel class takes part in the behavior of the Control Panel Interface. It validates/ manages entered password, changes status of the sensors, test sensors.



Class : Control Panel	
Description : This class store informations needed from using control panel's functionalities.	
Responsibility :	**Collaborator :**
Changes status of the sensor.	Sensor
Tests if the sensor is working.	Sensor
Sets or removes password.	
Validates the password.	

2.4 Sensor Class

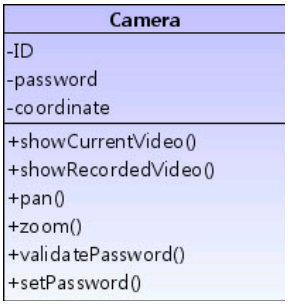
• Sensor class stores informations, manipulates sensors - activate, deactivate, test.



Class : Sensor	
Description : This class stores sensor informations and provides all of necessary functions which make sensors work properly.	
Responsibility :	**Collaborator :**
Defines sensor ID/type.	
Arms/Disarms sensors.	
Tests sensors.	
Changes threshold.	
Sends signal.	System
Manages sensor list.	

2.5 Camera Class

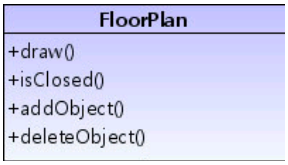
• Camera class manipulates cameras. It can record, show videos, pan/zoom cameras, set password on camera.



Class : Camera	
Description : This class stores camera informations and provides all of necessary functions which make cameras work properly.	
Responsibility :	**Collaborator :**
Defines camera ID.	
Shows current video.	
Shows recorded video.	
Pans/zooms.	
Changes camera passwords.	
Records video.	
Validates camera passwords	
Manages camera list.	

2.6 Floor Plan Class

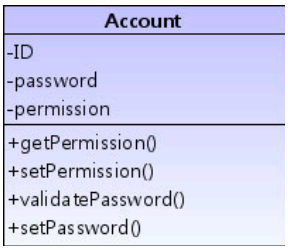
• Floor Plan class stores floor plan data, draws floor plan on the screen, provides the function to edit floor plan.



Class : FloorPlan	
Description : This class stores floorplan informations and provides all functions which make floorplan information change.	
Responsibility :	**Collaborator :**
Displays the floor plan.	
Displays position of sensors.	Sensor
Displays position of cameras.	Camera
Edits the floor plan.	Camera, Sensor

2.7 Account Class

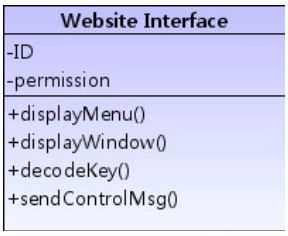
• Account class store account data and provide functions to validates or manages user ID/passwords and permissions.



Class : Account	
Description : This class store informations and functionalities needed to validates or manages user ID/passwords and permissions.	
Responsibility :	**Collaborator :**
Validates Internet Login ID/Password.	
Changes passwords.	
Changes permission.	
Sends permission information.	
Manages account list.	

2.8 Website Interface Class

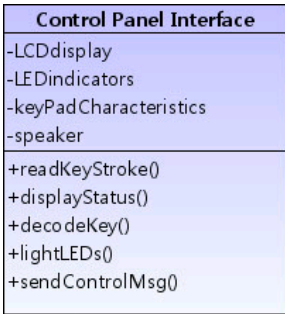
• Website Interface class composes web interface. It provides functions to draw appropriate windows, menus and connection to the control software in home-based PC.



Class : Website Interface	
Description : This class provides user interface via internet Webserver.	
Responsibility :	**Collaborator :**
Displays available menus.	Account
Communicates with system.	System
Displays function window.	

2.9 Control Panel Interface Class

• Control Panel Interface Class only provides user interface and connection methods.



Class : Control Panel Interface	
Description : This class provides user interface via Control Panel.	
Responsibility :	**Collaborator :**
Reads key stroke.	
Displays system status.	System
Communicates with system.	System
Gives an alarm.	System

3. State Chart Diagrams

3.1 Overall Description

As you see in **Software Architecture** paragraph, SafeHome Product software consists of several layers, and also as you see in **Overview - Class Analysis Diagram** paragraph, each layer is consists of several collaborated classes. Therefore in this chapter, state chart diagrams are shown in several functional modules.

The relationship among the analysis classes and the state diagrams are like below table.

Class	State Diagram	Note
System	External Access and Mapping / Alarm Module	
Alarm		
Control Panel	Security Function Module	
Sensor		collaborated with Administration
Camera	Surveillance Function Module	collaborated with Administration
Floor Plan	Administration Function Module	
Account		collaborated with Interfaces
Website Interface	Interfaces Module	
Control Panel Interface		

3.2 External Access and Mapping / Alarm Module

External Access and Mapping / Alarm Module consists of **System** class and **Alarm** class. Their state diagram is like below.

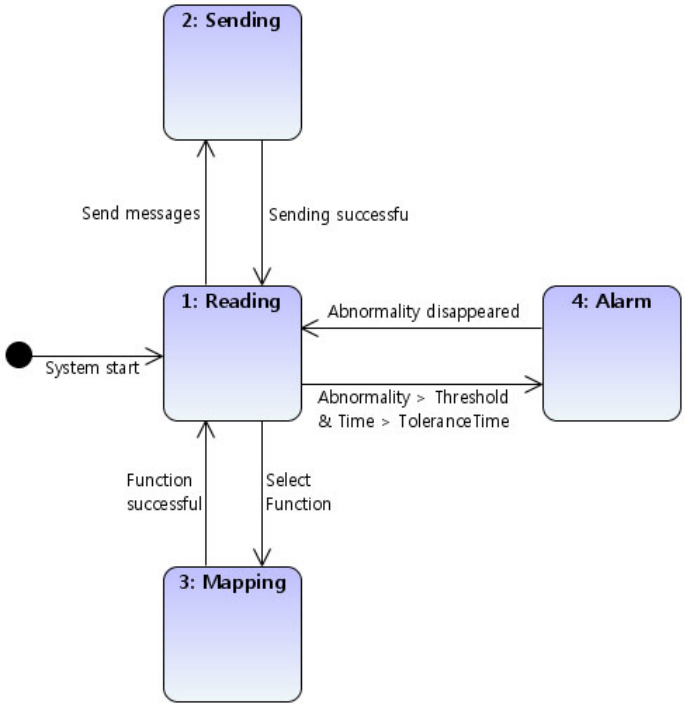


Figure 2-2. State diagram for external access and mapping / alarm module

3.2.1. Reading state

After the system started(control software executed), the system waits for signal from external access - web site, control panel, sensors. When the system receives messages, the system decodes them and the state is changed into appropriate state like below.

1. If the system receives messages from external interface, the state is changed into **3. Mapping** state.
2. If the system receives function result from operation layer, the state is changed into **2. Sending** state.
3. If the system receives "alarm" signal from sensor and its duration time is longer than tolerance time, the state is changed into **4. Alarm** state.

3.2.2. Sending state

In this state, the system encodes function result and send them to appropriate external interface. If the system sends messages to external interface successfully, the state is changed into **1. Reading** state.

3.2.3. Mapping state

In this state, the system maps messages into appropriate functions in operation layer. If the system sends messages to external interface successfully, the state is changed into **1. Reading** state.

3.2.4. Alarm state

In this state, the system calls the SafeHome Service Team and the homeowner and does services of **1. Reading** state simultaneously. If abnormality is terminated or the homeowner offs the system, the state is changed into **1. Reading** state.

3.3 Security Function Module

Security Function Module consists of **Control Panel** class and **Sensor** class. Their whole state diagram is like below. (This diagram is large, so inserted vertically. Sorry for your inconvenience) If you want to look Control Panel state diagram separately, refer to **Interface Module** paragraph.

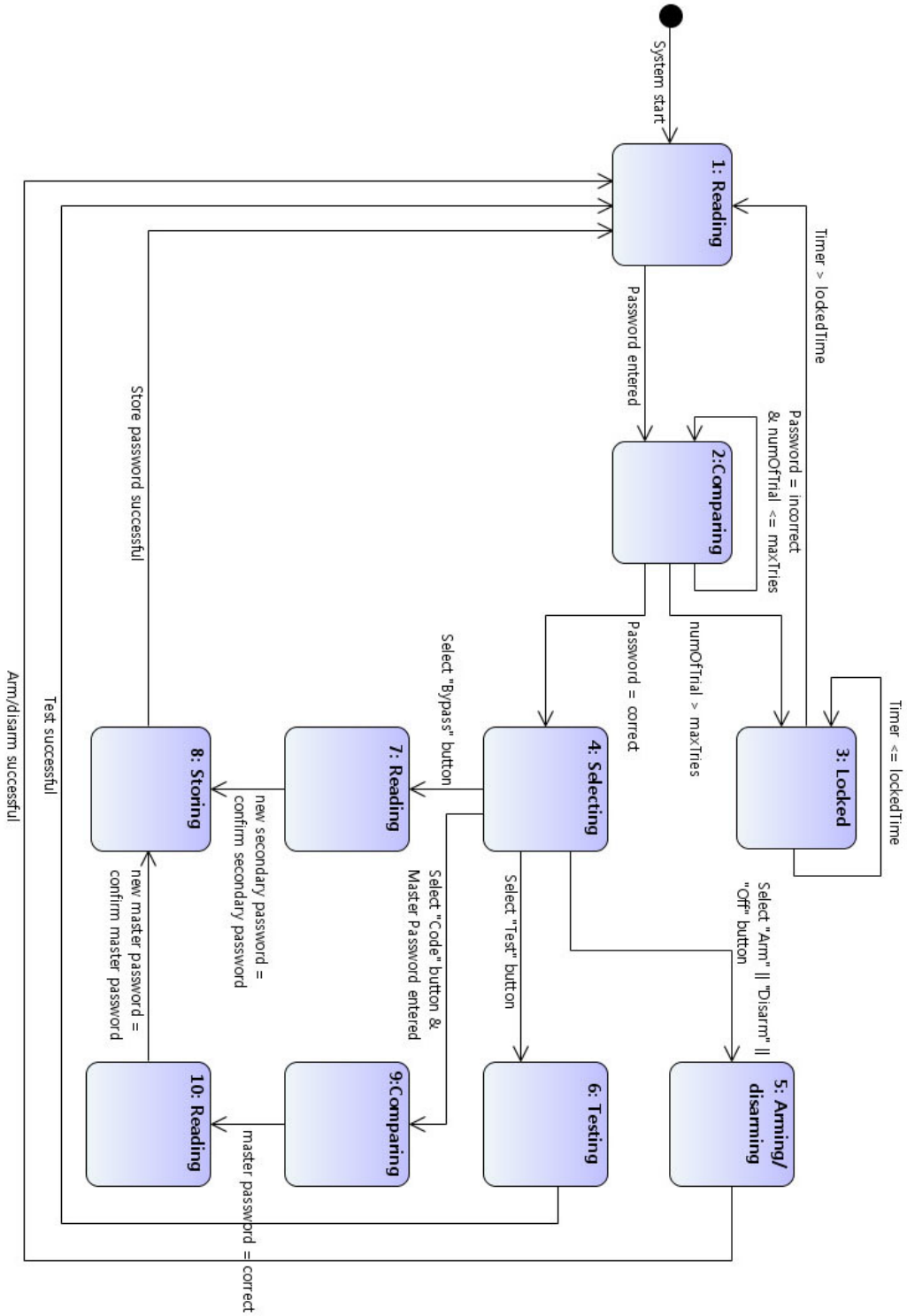


Figure 2-3. State diagram for security function

3.3.1. Reading state

After the system started, the control panel waits for key stroke. If the homeowner enters his/her password to the control panel with its key pad, the state is changed into **2. Comparing** state.

3.3.2. Comparing state

In this state, the control panel compares the entered password and password stored in the system.

1. If the entered password is incorrect and chance to reenter password remains, the homeowner can reenter password, and the state remains as **2. Comparing** state.
2. If the entered password is incorrect and chance to reenter password does NOT remain, the state is changed into **3. Locked** state.
3. If the entered password is correct, the state is changed into **4. Selecting** state.

3.3.3. Locked state

In this state, the control panel is locked and does NOT receive any input. The control panel records locked time with its timer and If timer dues for lockedTime stored in system, the state is changed into **1. Reading** state.

3.3.4. Selecting state

In this state, the control panel waits for selecting the function button that homeowner want to use.

1. If the homeowner selects "Arm" or "Disarm" or "Off" button, the state is changed into **5. Arming/Disarming** state.
2. If the homeowner selects "Test" button, the state is changed into **6. Testing** state.
3. If the homeowner selects "Bypass" button, the state is changed into **7. Reading** state.
4. If the homeowner selects "Code" button and his/her master password to the control panel with its key pad, the state is changed into **9. Comparing** state.

3.3.5. Arming/Disarming state

In this state, the control panel sends activate/deactivate signal to sensors through the system. The sensors receive signals and the state of the sensors are changed appropriately. (Refer to the sensor state diagram below) After arm/disarm sensors, the state is changed into **1. Reading** state.

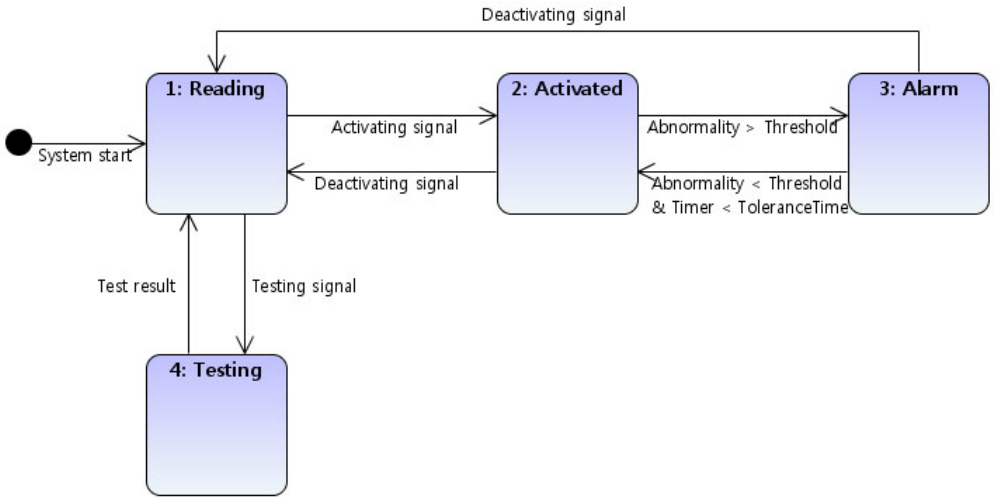


Figure 2-4. State diagram for sensor

3.3.6. Testing state

In this state, the control panel sends test signal to sensors through the system. The sensors receive signals and the state of the sensors are changed appropriately. (Refer to the sensor state diagram) After testing sensors, the state is changed into **1. Reading** state.

3.3.7. Reading state

In this state, the control panel waits for key stroke. If the homeowner enters new secondary password and confirming secondary password to the control panel with its key pad and new secondary password and confirming secondary password are same, the state is changed into **8. Storing** state.

3.3.8. Storing state

In this state, the control panel stores received password in the system. After storing received password, the state is changed into **1. Reading** state.

3.3.9. Comparing state

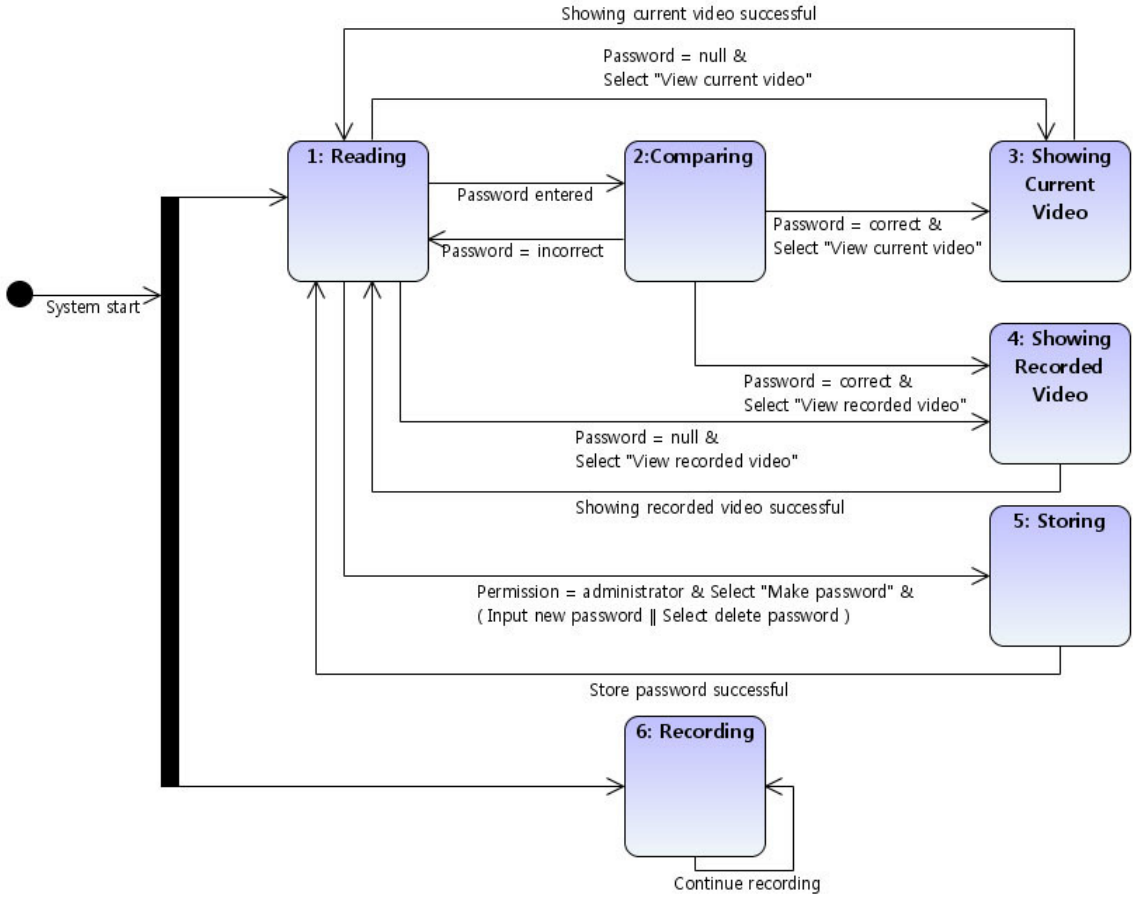
In this state, the control panel compares the entered password and master password stored in the system. If the entered password is correct, the state is changed into **10. Reading** state.

3.3.10. Reading state

In this state, the control panel waits for key stroke. If the homeowner enters new master password and confirming master password to the control panel with its key pad and new master password and confirming master password are same, the state is changed into **8. Storing** state.

3.4 Surveillance Function Module

Surveillance Function Module consists of **Camera** class. Its state diagram is like below.



3.4.1. Reading state

After the system started, the surveillance function module waits for control signal. If the module receives signals, the state is changed into appropriate state like below.

1. If the camera password exists and the homeowner enters camera password, the state is changed into as **2. Comparing** state.
2. If the camera password does NOT exist or the homeowner has administration permission, and the homeowner selects "View current video", the state is changed into as **3. Showing Current Video** state.
3. If the camera password does NOT exist or the homeowner has administration permission, and the homeowner selects "View recorded video", the state is changed into as **4. Showing Recorded Video** state.
4. If the homeowner has administration permission and the homeowner selects "Make password", and input new camera password or selects deleting password, the state is changed into as **5. Storing** state.

3.4.2. Comparing state

In this state, the surveillance function module compares the entered camera password and camera password stored in the system.

1. If the entered password is incorrect, the state is changed into **1. Reading** state.
2. If the entered password is correct and the homeowner selects "View current video", the state is changed into as **3. Showing Current Video** state.
3. If the entered password is correct and the homeowner selects "View recorded video", the state is changed into as **4. Showing Recorded Video** state.

3.4.3. Showing Current Video

In this state, the surveillance function module shows current video of selected camera. After the homeowner exists from this function, the state is changed into **1. Reading** state.

3.4.4. Showing Recorded Video

In this state, the surveillance function module shows recorded video of selected camera stored in the system. After the homeowner exists from this function, the state is changed into **1. Reading** state.

3.4.5. Storing state

In this state, the surveillance function module stores entered camera password or deletes stored camera password. After storing data is finished, the state is changed into **1. Reading** state.

3.4.6. Recording state

After the system started, the surveillance function module records videos continuously. So, the surveillance function is in recording state any time.

3.5 Administration Function Module

Administration Function Module consists of **Floor Plan** class and **Account** class, and has three functions - Internet ID/permission Management, Sensor Threshold Management, Floor Plan Management. As you can see, administration functions operate data value stored in the system and does not have any active operation. So they does not have any state to describe.

3.6 Interfaces Module

Surveillance Function Module consists of **Control Panel Interface** class and **Website Interface** class. Their state diagrams are like below. The state descriptions of Control Panel Interface is same as the state description 1-4 of **Security Function Module**, so they are omitted here.

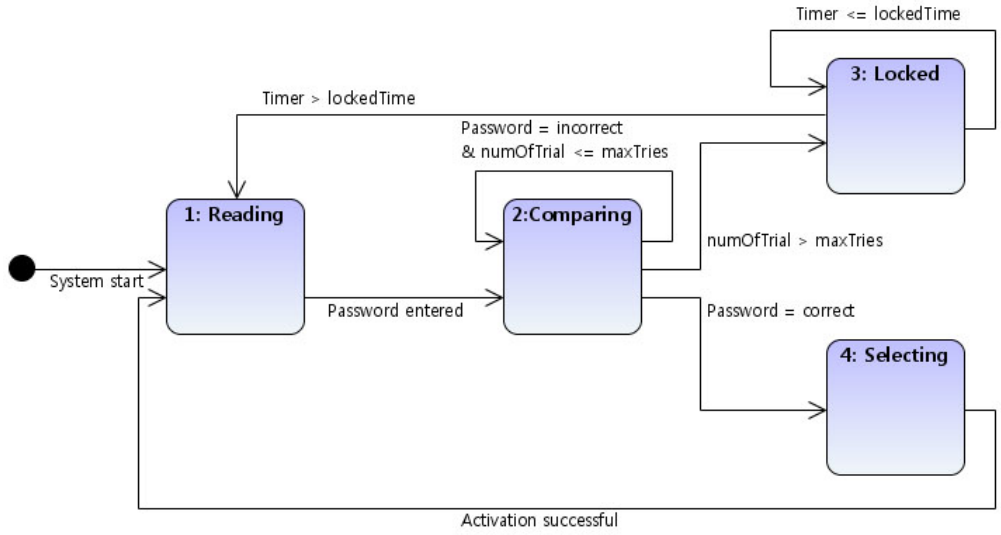


Figure 2-6. State diagram for control panel

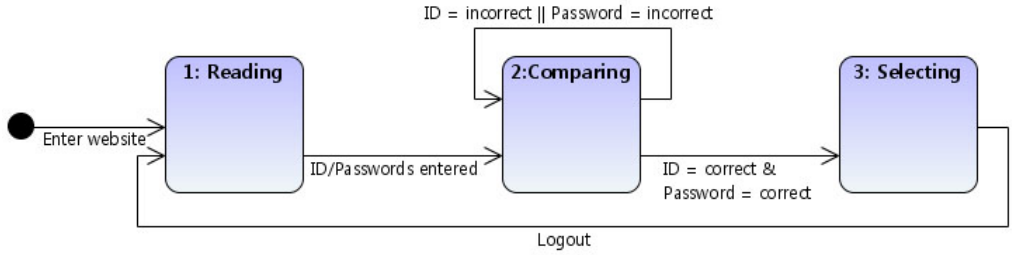


Figure 2-7. State diagram for web site

3.6.1. Reading state

After the homeowner entered SafeHome Product web site, the web site interface waits for any key stroke of the homeowner. If the homeowner enters his/her Internet ID and passwords, the state is changed into **2. Comparing** state.

3.6.2. Comparing state

In this state, the web site interface compares the entered Internet ID/passwords and Internet ID/passwords stored in the system.

1. If the entered Internet ID or passwords is incorrect, the homeowner can reenter Internet ID/passwords, and the state remains as **2. Comparing** state.
2. If the entered Internet ID and passwords are correct, the state is changed into **3. Selecting** state.

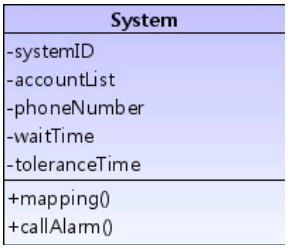
3.6.3. Selecting state

In this state, the web site interface waits for selecting the function button that homeowner want to use. If the homeowner selects "Logout" from the web site, the state is changed into **1. Reading** state.

4. Elaborated Classes

4.1 System Class

Analysis Class

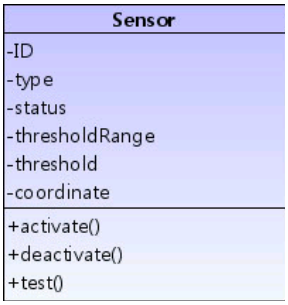


Elaborated Class

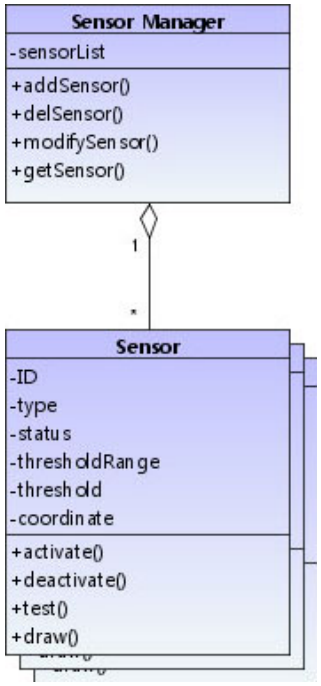
- Functions to encode, receive and send Control messages are added.
- To reduce complexity, mapping function is divided into two level. The first level maps messages to the three second level mapping functions. The second mapping functions call appropriate class and methods.

4.2 Sensor Class

Analysis Class



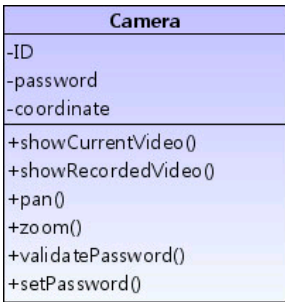
Elaborated Class



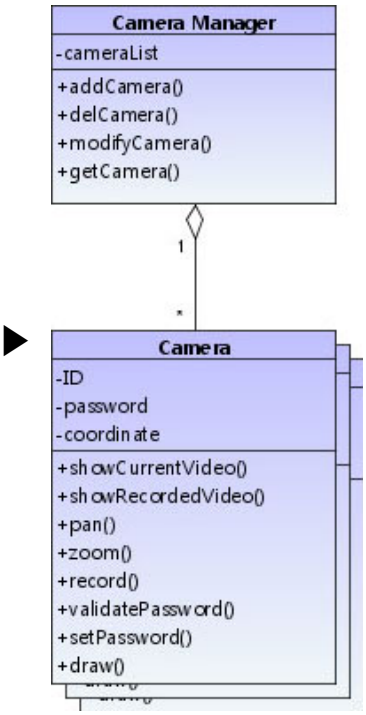
- There can be multiple sensors on the system, so to manage multiple sensors efficiently, Sensor Manager class is added.
- Sensor classes are associated with Sensor Manager class.
- Sensor Manager class has a sensor list attribute and operations to manipulate the list. The operations can be used when a sensor is replaced or added, or when needed to get the pointer to selected sensor class.
- To draw sensor icon in floor plan, draw() method is added to the sensor class.

4.3 Camera Class

Analysis Class



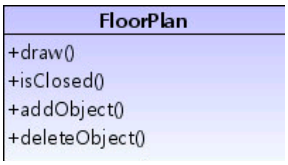
Elaborated Class



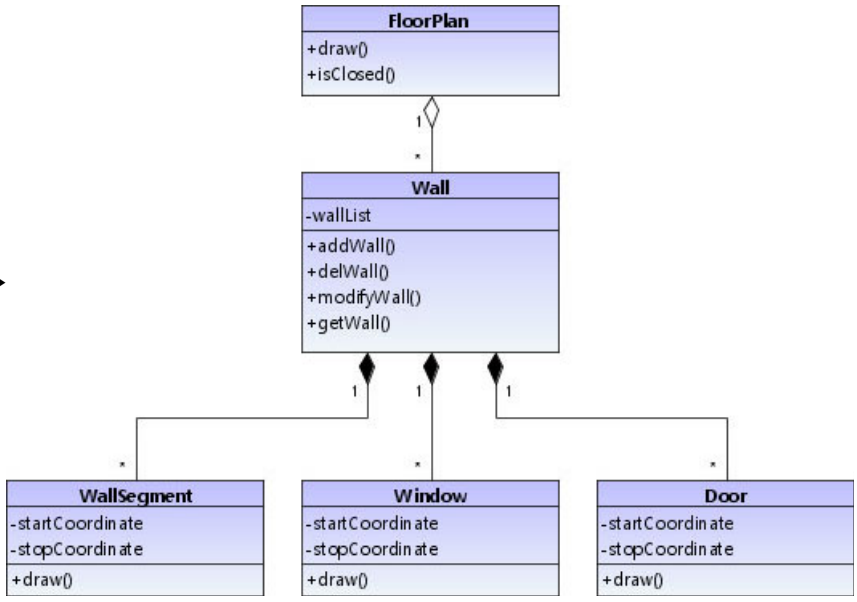
- There can be multiple cameras on the system, so to manage multiple sensors efficiently, Camera Manager class is added.
- Camera classes are associated with Camera Manager class.
- Camera Manger class has a Camera list attribute and operations to manipulate the list. The operations can be used when a camera is replaced or added, or when needed to get the pointer to selected camera class.
- To draw camera icon in floor plan, draw() method is added to the camera class.
- Camera recording method is added.

4.4 Floor Plan Class

Analysis Class



Elaborated Class

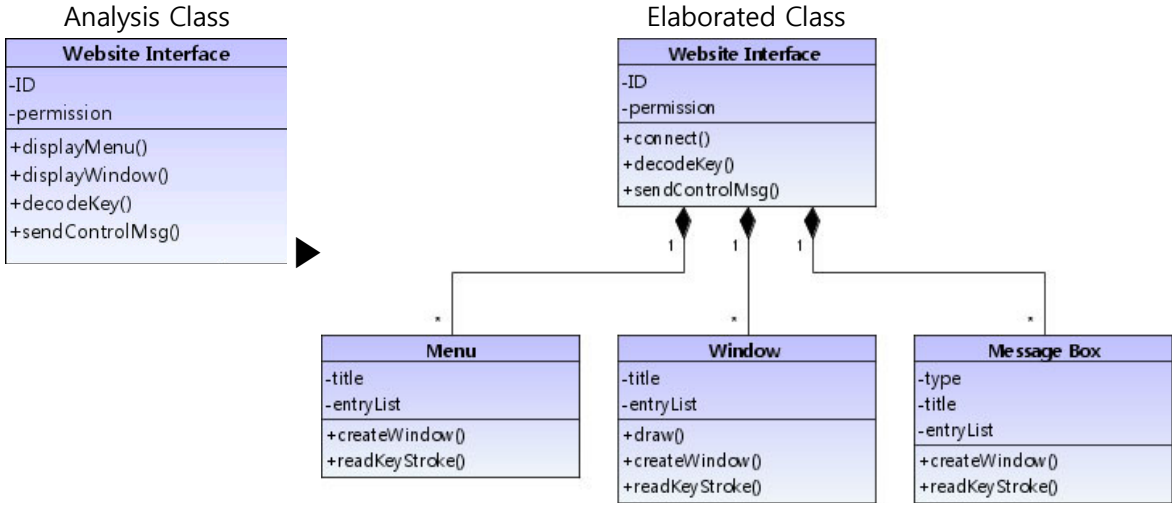


- Data classes are separated from manipulation class.
- Wall class has a wall list attribute. Wall segment, window, door data are stored in list.
- Wall class has operations to manipulate the list. The operations can be used when the floor

plan is edited.

- Wall Segment, Window, Door classes have start and end coordination attributes and draw() method. They are used to draw each entities on the screen.
- Because there can be plural entities, the entity classes have one-to-many composition relationship with Wall class.

4.5 Website Interface Class



- Connect() method to connect the system is added.
- Menu, Window, Message Box classes are separated from Website Interface class. Each entity classes are used to compose whole website interface. Because there can be plural entities, the entity classes have one-to-many composition relationship with Website Interface class.
- Each entity classes have title attribute to display window title, entry list attribute to store included entity, methods to draw each window and receive key or mouse events.
- Window class have draw() method to display contents of window.
- Message Box class have type attribute to store message type - alarm, declare, confirm and so on. Message Box class display appropriate box form due to type attribute.

- Other classes will be used untouched.

APPENDIX

A. Glossary

SafeHome Product
The product that consists of the SafeHome Product software, home-based PC, control panel, wireless box, sensors, cameras, accessibility to CPI servers and service team.

SafeHome Product software
The software that manages storing data, sensor and camera setting and external access.

SafeHome Product service team, service team
The division of the SafeHome company that provides the password-missing case disposal service, the sensor-detection disposal service and so on.

SafeHome Product Web site
The web site that operates SafeHome Products.

administration
The permission level that allows user to use all functions of the SafeHome Product software.

administrator
The user who has the administration permission.

camera
A optical device that records video data.

control panel
The interface located in the house of the homeowner and providing security mode changing functions.

CPI server
The server computer located in the SafeHome company and providing the web service.

guest
A person who doesn't have the administrator permission or isn't the homeowner.

home-based PC or PC
The computer located in the house of the homeowner, storing all data except web data and providing exceptional access gate.

homeowner
The user who has the house and purchases the SafeHome Product.

input box
A blank form that the system displays to receive data from the homeowner or guest.

internet login ID/password
A ID and passwords that needs to log in to SafeHome Web site.

master password
A password that needs to operate the control panel. Master password owner can make/delete secondary passwords.

message box
A box form that the system displays to inform the homeowner or the guest.

permission
The authority that needs to use some functions.

priority
The order that how much the user want to use.

secondary password
A password that needs to operate the control panel. Secondary password owner can not make/delete another secondary passwords.

sensor
A device that detects specified objects.

thumbnail
Snapshots provided by the SafeHome Product software to let users know the content of the video file.

threshold
A bounded value that doesn't make sensors to respond.

tolerance time
The time that the system waits after the first sensor response. After tolerance time, if there is the sensor reponse yet, the system makes phone call to SafeHome Product service team.

validation
The process that the system verifies a internet login ID and passwords, changes user state to log-in state and gives appropriate permission.

wireless box
The device located in the house of the homeowner and providing wireless transmission among the home-based PC, control panel, cameras and sensors.

B. Word Index

C. Picture & Table Index

D. Meeting Records

To be continued on the next page.

The 11th Meeting Record

Course	CS350 Software Engineering	**Instructor**	prof. Moon-zoo Kim
		Recorder	Eut-deum Kim
Time	April. 10th, 2008. p.m. 04:30~06:00	**Location**	Resting room in CS building
Subject	Drawing use-case diagrams		

Attendant	**Student Num**	**Name**	**Remark**
	20010090	Eut-deum Kim	
	20030364	Myung-kyung Lee	
	20060034	Ji-seong Gu	
	20060340	Da-bi Ahn	

Contents

◎ The Contents of the meeting

1. The Essential Activities

· Some similar detailed items are deleted and merged.

· Modify some pictures.

· We draw use-case diagrams.

① We pick out some important features from requirement specifications - Internet validation, Arm/Disarm System, Arm/Disarm sensors, Alarm encounter, Show current video, Show recorded video, Camera password management, Internet ID/permission management, Sensor threshold management, Floor plan management - and make them use-case entity.

② We make device and person taking part in the features as role.

· We write use-case descriptions each every use-case entities.

2. Next schedule

· We are scheduled to have next meeting on Friday.

※ We admit and agree with the contents of this meeting record.

Signature				
	Eut-deum Kim	_____	Myung-kyung Lee	_____
	Ji-seong Gu	_____	Da-bi Ahn	_____

The 12th Meeting Record

Course	CS350 Software Engineering	**Instructor**	prof. Moon-zoo Kim
		Recorder	Eut-deum Kim
Time	April. 11th, 2008. p.m. 04:30~06:00	**Location**	Resting room in CS building
Subject	Drawing use-case diagrams		

Attendant	**Student Num**	**Name**	**Remark**
	20010090	Eut-deum Kim	
	20030364	Myung-kyung Lee	
	20060034	Ji-seong Gu	
	20060340	Da-bi Ahn	

Contents

◎ The Contents of the meeting

1. The Essential Activities

· Some similar detailed items are deleted and merged.

· We add and modify some pictures in requirement specification in administration.

· We modify use-case diagrams.

① Internet validation entity is needed by all entity that can be accessed by remote PC - all entity except alarm encounter must "include" Internet validation entity.

② Alarm encounter can be activated during all entity except Internet validation executed when abnormality occurred. - all entity except Internet validation must "extend" Alarm encounter entity.

· We add use-case descriptions. Alarm encounter description was missed off.

· We add some pictures to use-case descriptions. Because requirement specification is followed use-case description, changing in requirement specification must be applied to use-case description.

2. Next schedule

· We spend too much time to modify requirement specification. So, we must finish it until next meeting.

· We are scheduled to have next meeting on Sunday.

※ We admit and agree with the contents of this meeting record.

Signature				
	Eut-deum Kim	_____	Myung-kyung Lee	_____
	Ji-seong Gu	_____	Da-bi Ahn	_____

The 13th Meeting Record

Course	CS350 Software Engineering	**Instructor**	prof. Moon-zoo Kim
		Recorder	Eut-deum Kim
Time	April. 13th, 2008. p.m. 06:00~09:00	**Location**	Resting room in CS building
Subject	Drawing swim-lane diagrams		

Attendant	**Student Num**	**Name**	**Remark**
	20010090	Eut-deum Kim	
	20030364	Myung-kyung Lee	
	20060034	Ji-seong Gu	
	20060340	Da-bi Ahn	

Contents

◎ The Contents of the meeting

1. The Essential Activities

· We add and modify some entities in the index page of revised requirement specification.

· After final review, we input final edition to DOORS.

· We write swim-lane diagrams and sequence diagrams by hand.

① As we wrote before, Internet validation entity is needed by all entity that can be accessed by remote PC. So, we draw common Internet validation part first.

② How can we apply Alarm encounter entity more actually? It can be derived at any time, when abnormality occurred.

③ Do we need to add state in sequence diagram? Our tool - visual paradigm doesn't support that feature - need to ask professor or TA.

④ What will be the subject of time-line? We use role-interface-object first, but does it OK?

2. Next schedule

· We are scheduled to have next meeting on Tuesday.

※ We admit and agree with the contents of this meeting record.

Signature				
	Eut-deum Kim	_____	Myung-kyung Lee	_____
	Ji-seong Gu	_____	Da-bi Ahn	_____

The 14th Meeting Record

Course	CS350 Software Engineering	**Instructor**	prof. Moon-zoo Kim
		Recorder	Ji-Seong Gu
Time	April. 15th, 2008. p.m. 04:30~11:30	**Location**	Resting room in CS building
Subject	Refining Diagrams and Report & Prefaring Presentation		

Attendant	**Student Num**	**Name**	**Remark**
	20010090	Eut-deum Kim	
	20030364	Myung-kyung Lee	
	20060034	Ji-seong Gu	
	20060340	Da-bi Ahn	

Contents

◎ The Contents of the meeting

1. The Essential Activities

· We make our hand-writing Swim-lane Diagrams and Sequence Diagrams be refined UML Diagrams using Visual Paradigm.

· We standardize our Swim-lane and Sequence diagrams.

· We review our Requirement specification and finally check if it reflects all changes made until now.

· We attach brief explanatory notes for Use-case Diagrams.

· We complete to draw up or modify our analysis-modeling report and requirement specification reports.

· We discuss what we should state in our presentation and make our presentation materials using Power Point.

2. Next schedule

· We have done a lot things in last week, so we decides to have next meeting after 3 days.(Starting next projects)

※ We admit and agree with the contents of this meeting record.

Signature				
	Eut-deum Kim	_____	Myung-kyung Lee	_____
	Ji-seong Gu	_____	Da-bi Ahn	_____

The 15th Meeting Record

Course	CS350 Software Engineering	**Instructor**	prof. Moon-zoo Kim
		Recorder	Ji-Seong Gu
Time	April. 18th, 2008. p.m. 04:30~06:00	**Location**	Resting room in CS building
Subject	Discussion on our data design architecture		

Attendant	**Student Num**	**Name**	**Remark**
	20010090	Eut-deum Kim	
	20030364	Myung-kyung Lee	
	20060034	Ji-seong Gu	
	20060340	Da-bi Ahn	

Contents

◎ The Contents of the meeting

1. The Essential Activities

· We first discuss which kinds of components are needed, but as times go by, our discussion is nearly about our implementation. So we decide to give priority to make our data design architecture to hold our top-down approaches.

· During our discussion, we conclude that Data Centered architecture or layered architecture are fitting to our SafeHome Product's data design architecture.

2. Next schedule

· Until 20th April, we decides to read our textbook and make candidate class using the methodology and principles in our textbook.

※ We admit and agree with the contents of this meeting record.

Signature				
	Eut-deum Kim	_____	Myung-kyung Lee	_____
	Ji-seong Gu	_____	Da-bi Ahn	_____

The 16th Meeting Record

Course	CS350 Software Engineering	**Instructor**	prof. Moon-zoo Kim
		Recorder	Ji-Seong Gu
Time	April. 20th, 2008. p.m. 06:00~08:30	**Location**	Resting room in CS building
Subject	Deriving classes and Drawing Deployment Diagram		

Attendant	**Student Num**	**Name**	**Remark**
	20010090	Eut-deum Kim	
	20030364	Myung-kyung Lee	
	20060034	Ji-seong Gu	
	20060340	Da-bi Ahn	

Contents

◎ The Contents of the meeting

1. The Essential Activities

· We first discuss about what should be made into class and what shouldn't based on each one's investigations and integrate all results into 9 classes.

· After deciding classes, we fill attributes and operations needed to execute all functionality defined in requirement specification.

· We draw Deployment Diagram to understand the structure of whole SafeHome Product system -Composition of Hardware-.

2. Next schedule

· Until 23rd April, we decides to make CRC Cards based on today's discussion, and think about the relation among classes.

※ We admit and agree with the contents of this meeting record.

Signature				
	Eut-deum Kim	_____	Myung-kyung Lee	_____
	Ji-seong Gu	_____	Da-bi Ahn	_____

The 17th Meeting Record

Course	CS350 Software Engineering	**Instructor**	prof. Moon-zoo Kim
		Recorder	Myung-kyung Lee
Time	April. 23th, 2008. p.m. 04:30~6:30	**Location**	Meeting room in CS building
Subject	Drawing class diagram and refining them with CRC cards		

Attendant	**Student Num**	**Name**	**Remark**
	20010090	Eut-deum Kim	
	20030364	Myung-kyung Lee	
	20060034	Ji-seong Gu	
	20060340	Da-bi Ahn	

Contents

◎ The Contents of the meeting

1. The Essential Activities

· We refined our former deployment diagram.

· We drew class diagram with prior class definitions.

· We made CRC cards based on preceding class design. We discussed whether our class definitions could support prior use-cases perfectly with these cards, and refine class definitions based on the discussion. We performed this step once again.

2. Next schedule

· We are scheduled to have next meeting on Friday.

※ We admit and agree with the contents of this meeting record.

Signature				
	Eut-deum Kim	_____	Myung-kyung Lee	_____
	Ji-seong Gu	_____	Da-bi Ahn	_____

The 18th Meeting Record

Course	CS350 Software Engineering	**Instructor**	prof. Moon-zoo Kim
		Recorder	Myung-kyung Lee
Time	April. 25th, 2008. p.m. 05:30~8:30	**Location**	Meeting room in CS building
Subject	Drawing state diagram		

	Student Num	**Name**	**Remark**
Attendant	20010090	Eut-deum Kim	
	20030364	Myung-kyung Lee	
	20060034	Ji-seong Gu	
	20060340	Da-bi Ahn	

Contents

◎ The Contents of the meeting

1. The Essential Activities

· Based on last discussion using CRC cards, we refined class definitions once again.

· We drew state diagrams for each class using Visual Paradigm.

2. Next schedule

· We complete analysis step. We will expand our previous analysis class definitions. From now, we must care about real implementation.

· We will have a meeting tomorrow.

※ We admit and agree with the contents of this meeting record.

Signature	Eut-deum Kim	_____	Myung-kyung Lee	_____
	Ji-seong Gu	_____	Da-bi Ahn	_____

The 19th Meeting Record

Course	CS350 Software Engineering	**Instructor**	prof. Moon-zoo Kim
		Recorder	Myung-kyung Lee
Time	April. 26th, 2008. p.m. 04:30~10:30	**Location**	Meeting room in CS building
Subject	State diagram and elaborated class		

	Student Num	**Name**	**Remark**
Attendant	20010090	Eut-deum Kim	
	20030364	Myung-kyung Lee	
	20060034	Ji-seong Gu	
	20060340	Da-bi Ahn	

Contents

◎ The Contents of the meeting

1. The Essential Activities

· Tomorrow, April 27th, there is going to be stoppage of power supply. We won't be able to make meeting or to do assigned work until tomorrow evening. So we made hard work today.

· We refined state diagrams.

· We refined our prior class definitions in the concrete and then made elaborated class diagram.

2. Next schedule

· From tomorrow, we will make overall document.

※ We admit and agree with the contents of this meeting record.

Signature	Eut-deum Kim	_____	Myung-kyung Lee	_____
	Ji-seong Gu	_____	Da-bi Ahn	_____

E. Who-did-what list

Here is our who-did-what list in student number order.

Eut-deum Kim	Wrote meeting records. Made document forms. Drew architecture diagrams and wrote their descriptions. Drew state diagrams and wrote their descriptions. Drew elaborated class diagrams.
Myung-kyung Lee	Wrote meeting records. Extracted analysis classes. Made CRC index cards. Took part in CRC verification. Wrote class descriptions.
Ji-seong Gu	Wrote meeting records. Extracted analysis classes. Made CRC index cards. Took part in CRC verification. Wrote class descriptions.
Da-bi Ahn	Extracted analysis classes. Made CRC index cards. Took part in CRC verification.