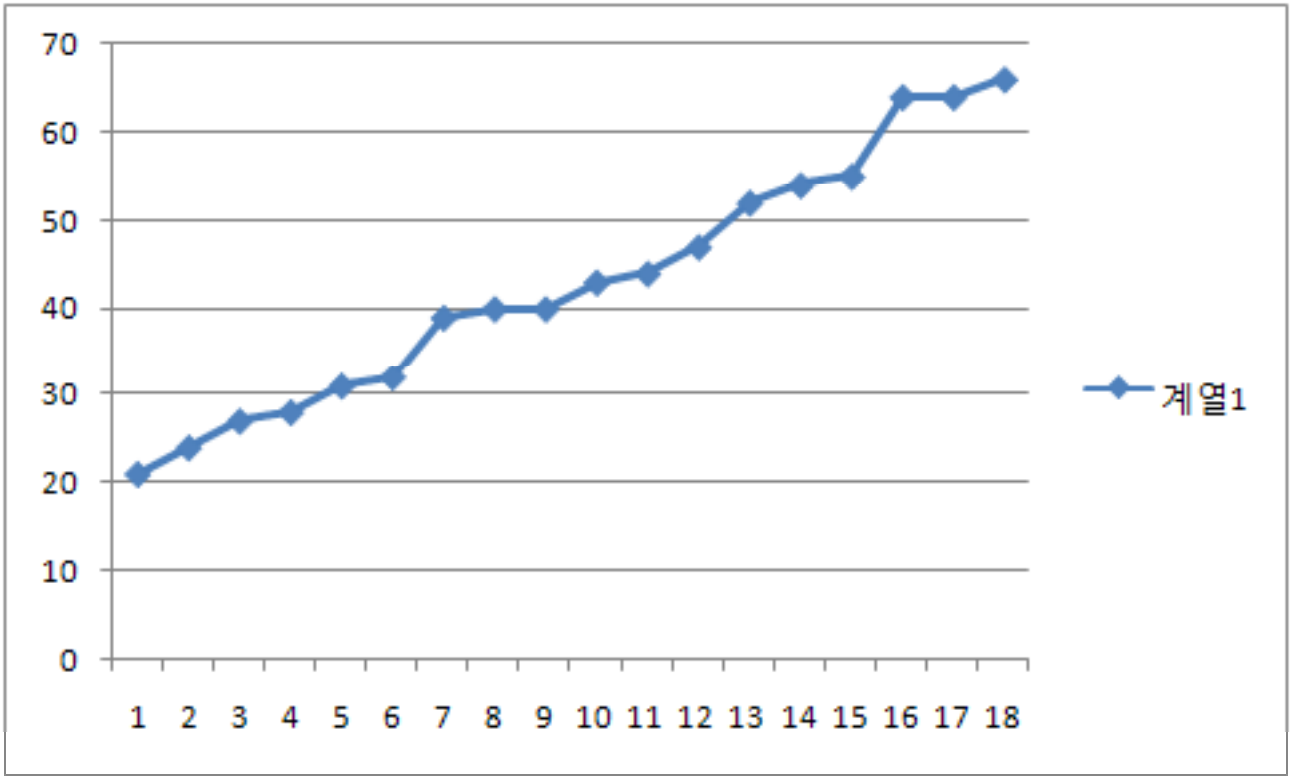
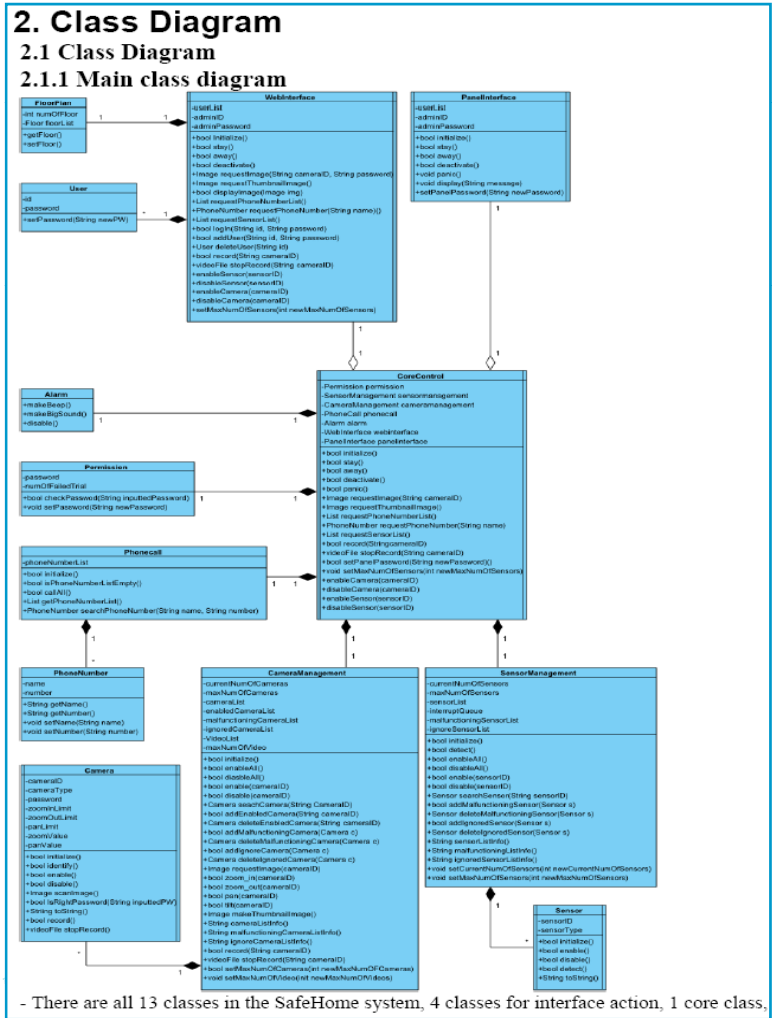
Review on Safehome Projects

Moonzoo Kim
CS Division of EECS Dept.
KAIST
moonzoo@cs.kaist.ac.kr
http://pswlab.kaist.ac.kr/courses/CS350-07

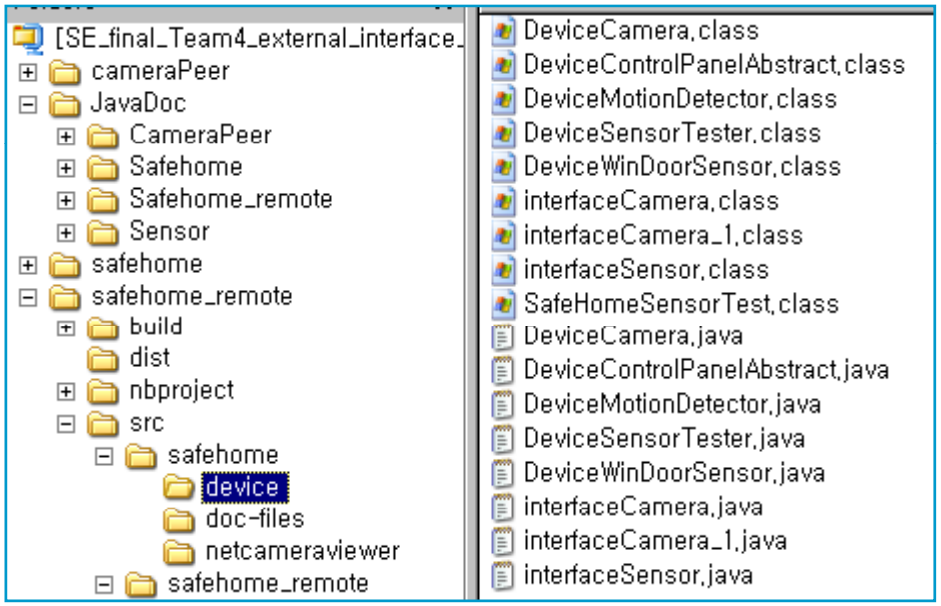


- Regarding Safehome project
  - Design documents do not match implementation well
  - Testing documents are poor
  - Project delivery deadline is crucial

Mismatch between Implementation and Design Documents



v.s.



Also, missing indexes and authorships

Implementation Document

4. Revision history of Implementation

When: 2008/05/31

What:

1. modify control Panel class in Maindemo.java.
   a. Add isLocked, isPower, isAway, isStay to store Status.
   b. Add variable for Socket Communication.
   c. Add ArraryList for collecting several Password.
   d. Modify Button1 ~ Button0, #,* method to Lock for each Status.
   e. Add codes for passing Password in Lock condition.
   f. Create function Void addDigit(int) which will receive and save Password into ArrayList one digit by one digit. If the Password becomes four digit, then s end it to Core.

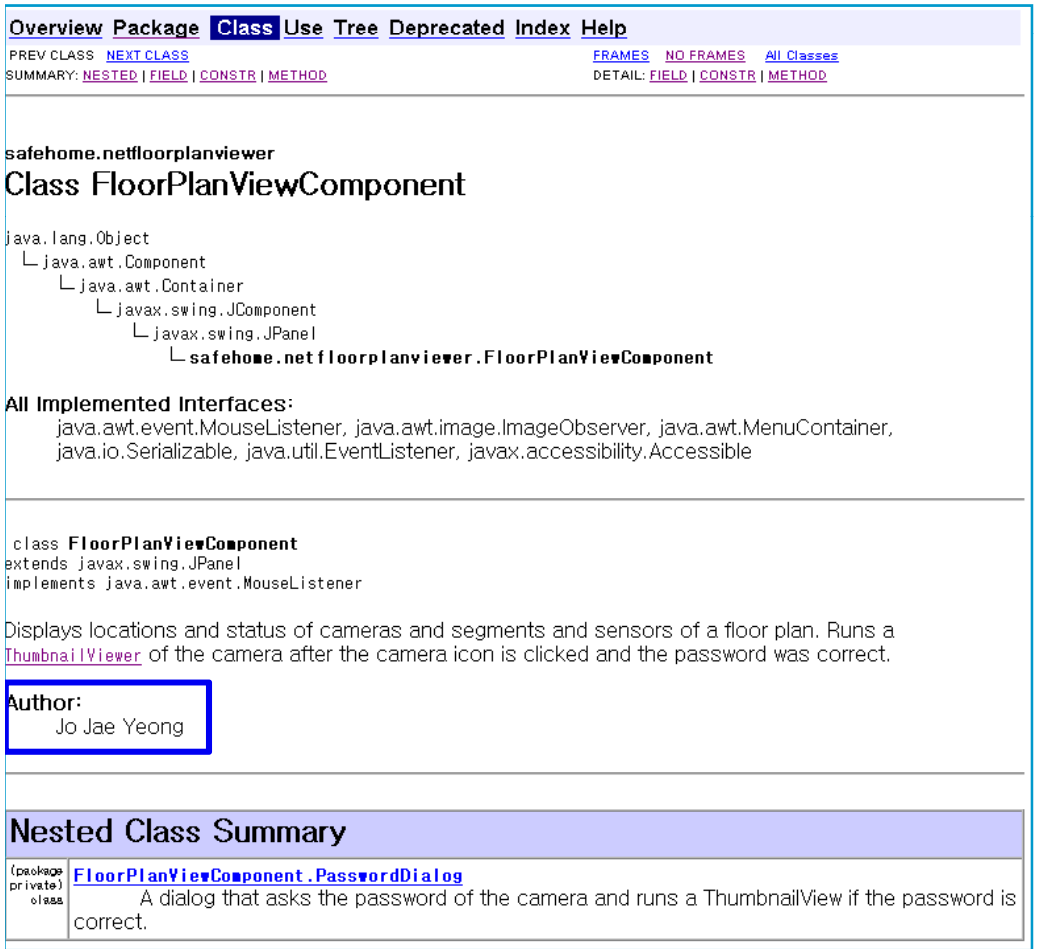
2. Create CPanelPeer.java and CPanelPacket based on the program Socket communic ation that is provided from TA.

3. Modify CPanelPeer
   a. CPanelPeer.java: This Class is for connecting Socket and for sending and rec eiving massages. Try Massage sending program after connecting Socket.
   b. Add variable Socket Comunication.

4. Add function sendMessageToCore and Core function getMessageFrom.

More information should be given
-Class name, function name, line #, etc.
-5W1H questions

Implementation Document: JavaDoc & Doxygen



Overview Package Class Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES All Classes
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

safehome.netfloorplanviewer

Class FloorPlanViewComponent

java.lang.Object
  └ java.awt.Component
    └ java.awt.Container
      └ javax.swing.JComponent
        └ javax.swing.JPanel
          └ **safehome.netfloorplanviewer.FloorPlanViewComponent**

All Implemented Interfaces:
java.awt.event.MouseListener, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.EventListener, javax.accessibility.Accessible

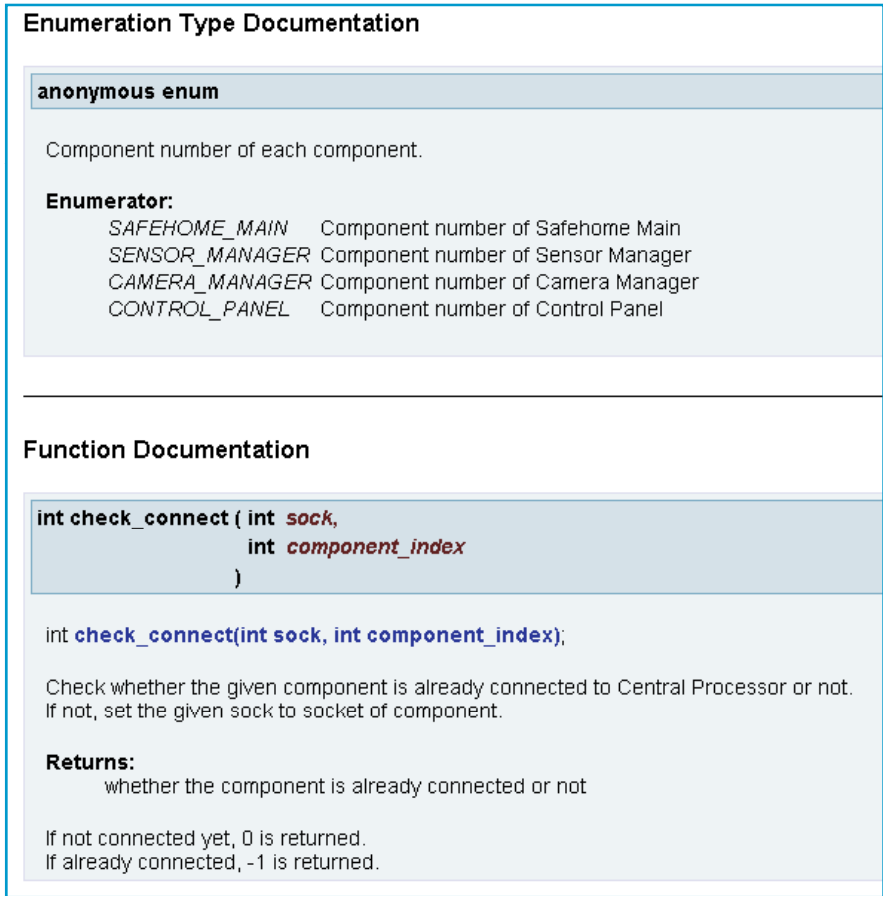
class **FloorPlanViewComponent**
extends javax.swing.JPanel
implements java.awt.event.MouseListener

Displays locations and status of cameras and segments and sensors of a floor plan. Runs a ThumbnailViewer of the camera after the camera icon is clicked and the password was correct.

Author:
Jo Jae Yeong

Nested Class Summary

(package private) class	**FloorPlanViewComponent.PasswordDialog** A dialog that asks the password of the camera and runs a ThumbnailView if the password is correct.



Enumeration Type Documentation

anonymous enum

Component number of each component.

Enumerator:

SAFEHOME_MAIN	Component number of Safehome Main
SENSOR_MANAGER	Component number of Sensor Manager
CAMERA_MANAGER	Component number of Camera Manager
CONTROL_PANEL	Component number of Control Panel

Function Documentation

int check_connect (int *sock*,
int *component_index*
)

int **check_connect(int sock, int component_index)**;

Check whether the given component is already connected to Central Processor or not.
If not, set the given sock to socket of component.

Returns:
whether the component is already connected or not

If not connected yet, 0 is returned.
If already connected, -1 is returned.

Testing Document (1/3)

3. Test Cases

3.1 Interface

1. Can user access the system by using control panel?

Theoretical Result

1. If user turns on the system, user can access the system by using control panel.
2. Control panel is connected to the system, so if user pushes "on" button on control panel after the system stared, power LED which is green on control panel is turned on.
3. Now control panel is ready to act several functions for user.

Testing Result

1. Starting with the SafeHome system, user must select control panel that is used by to access the system.

Good idea to include visual snapshots since many testing scenarios of Safehome are performed through GUI

Testing Document (2/3)

4.1.2 DeviceCamera

Test case name	DeviceCamera	Test case Description
Test case provided by	Soohyun Lee	To test whether functions in camera is working properly.
Tested by	Soohyun Lee	

Test	Input	sequence	output	Pass/Fail
1	Zoom in the camera	zoomIn()	Camera is zoomed in	P
2	Zoom out the camera	zoomOut()	Camera is zoomed out	P
3	Pan left	panLeft()	Camera is panned to the left	P
4	Pan Right	panRight()	Camera is panned to the right	P
Comments				

Test Pass Criteria	All tests are passed	Pass/Fail
Test Fail Criteria	One or more tests failed.	P

Good idea to describe test cases using a standard template

Important info. is missing
-Pre&post condition
-Description of how input is given and how output is checked

Testing Document (3/3)

3. Integration Test

We performed integration test in following order.

3.1 Primary Socket Communication

① Description: First trial for network communication by exchanging dummy message between primary server and two primary clients.

② Goal: Complete exchange dummy message between primary server and two primary clients.

③ Result:

- Type casting error. Watch out!

3.2 Camera Interaction

① Description: Camera and GUI camera part was completed. So check communication between camera and GUI camera part. Other parts are not completed.

② Goal: Exchange request from GUI camera part and image from Camera.

③ Result:

- Camera Packet error.
  - Because of different package name
- Type casting error.

Testing document is poor.
-Bug history is barely informative
-Concrete function name, line #, causes of bugs and how to fix it are missing