Comments on the SafeHome Project

- All teams did good job in general.
  - So you do not need to have a team interview with me
  - However, you could improve your requirement specification document in detail
- You should present your design on the class of May 17, which is also the deadline of the 2nd part of the SafeHome project
  - Demonstrate how good your design is based on the design principles
  - Also, explicitly show that your design can be traced back to the requirement specification and the analysis model

Ex 0. Think Carefully

- You have to think carefully to define right requirements of the target system which may not explicit in the source document (in our case SEPA)
  - Ex. You have to allow 5 min delay for activating/arming the sensors. Otherwise, a homeowner cannot leave his/her house; if a homeowner opens the door to leave, the SafeHome will raise alarm.
  - Ex. The SafeHome should not require password to activate “panic” function because “panic” button will be pressed in an emergency

Ex1. Organization of the Document

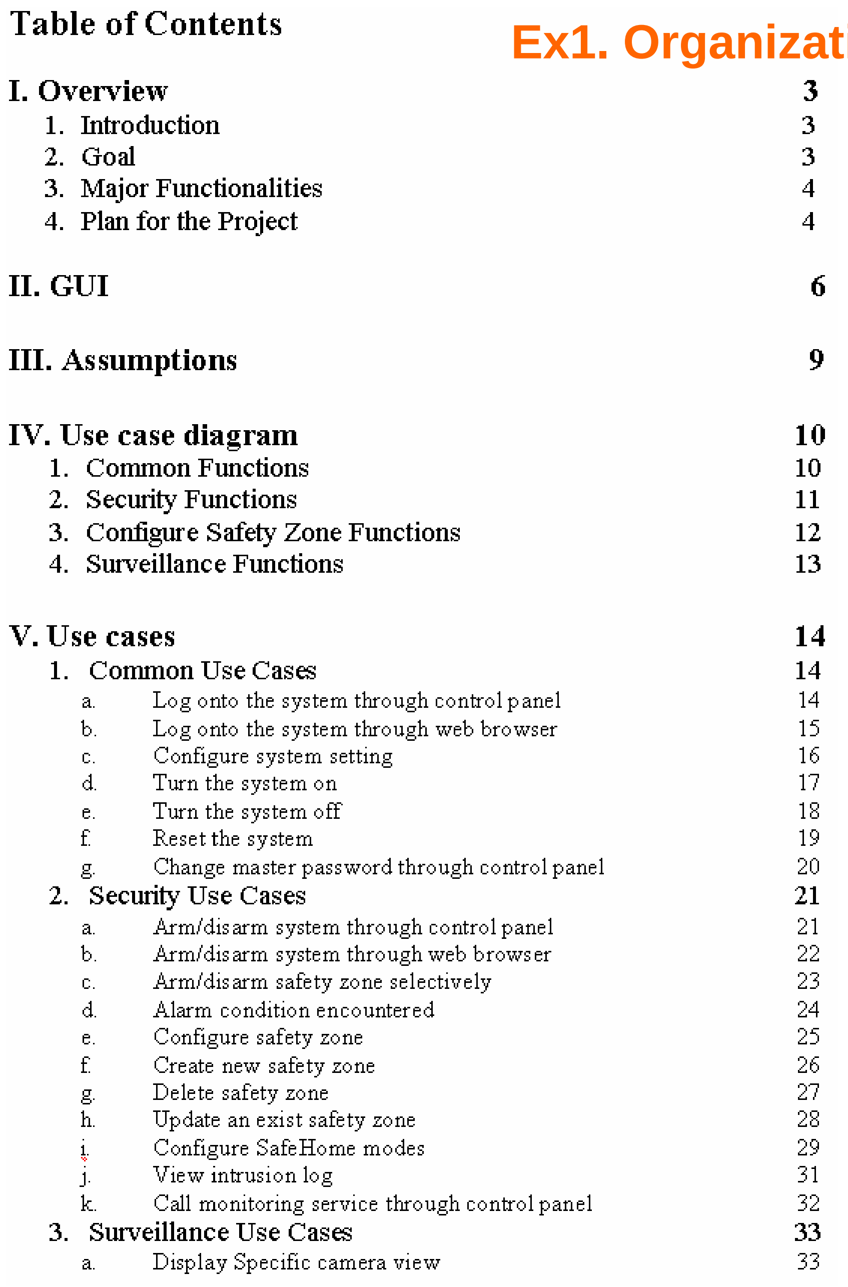


Table of Contents

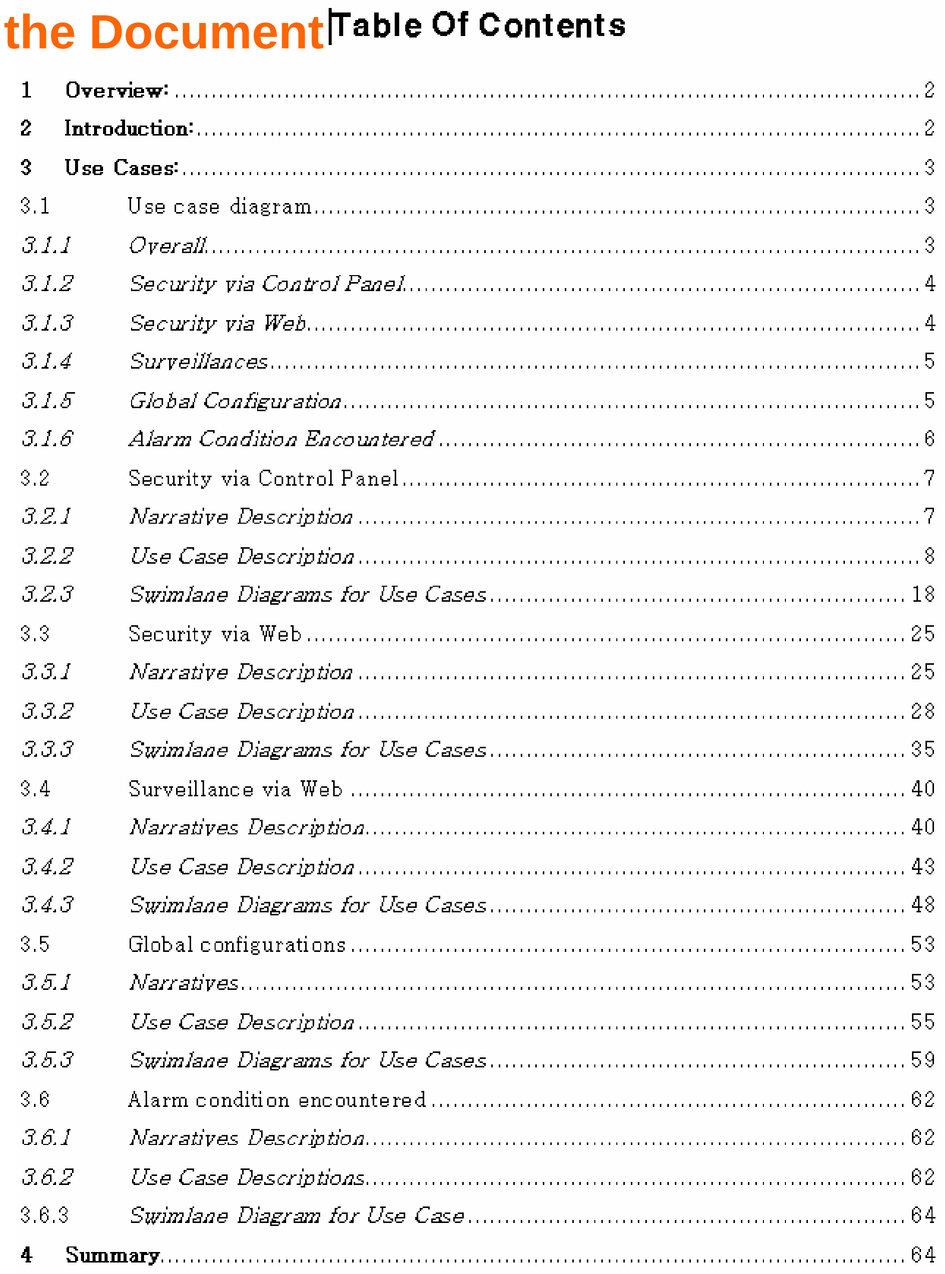


Table Of Contents

Ex2. Use Cases

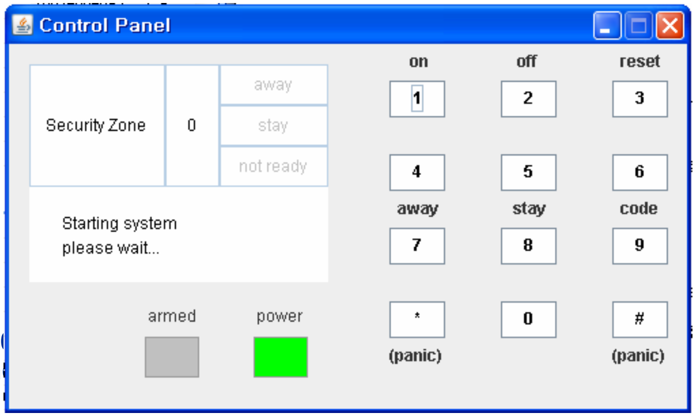
- Describe who wrote the use cases. Authorship is important
  - It helps you figure out whom you should talk to when you need to modify use cases
    - This rule applies for all types of document including code
- Exception # should be from success scenario
- Differentiate exceptions from different use-cases
- Open issues are important as well
  - Your requirement specification is always incomplete
  - Explicit description of what to be resolved in future helps detailed design

Ex 3. Cosmetics

- One of the most important goals for documents is to help reader to access necessary information easily. Cosmetics matters
  - Pay attention to English grammar!
    - Singular/plural
    - Articles
    - Pronoun
  - Right indentation
  - Be sure that texts in your diagram are large enough to be read
  - When you refer use-cases, add page #.
  - Hypertext link is recommended

Classes for HW Device Drivers

- A list of HW in the SafeHome system
  - Control panel
    - An abstract class **`SafeHomeControlPanel.class`**
    - You can extend it and fill out the callback methods for the buttons and use display APIs.



Starting system
please wait...

- Window/door sensors
  - **`WinDoorSensor.class`**
  - You can use it through the **`Sensor`** interface
    - See SEPA 331pg
- Motion sensors
  - **`MotionDetector.class`**
  - You can use it through the **`Sensor`** interface
- Cameras
  - **`Camera.class`**
  - You can instantiate the class and use it through **`InterfaceCamera`** interface

SafeHomeControlPanel.class

- Callback functions for the Control Panel's buttons
  - `abstract public void button1();`
  - `abstract public void button2();`
  - `abstract public void button3();`
  - `abstract public void button4();`
  - `abstract public void button5();`
  - `abstract public void button6();`
  - `abstract public void button7();`
  - `abstract public void button8();`
  - `abstract public void button9();`
  - `abstract public void buttonStar();`
  - `abstract public void button0();`
  - `abstract public void buttonSharp();`
- Control Panel's visual display
  - `public void setSecurityZoneNumber(int num)`
  - `public void setDisplayAway(boolean on)`
  - `public void setDisplayStay(boolean on)`
  - `public void setDisplayNotReady(boolean on)`
  - `public void setDisplayShortMessage1(String message)`
  - `Public void setDisplayShortMessage2(String message)`
  - `public void setArmedLED (boolean on)`
  - `public void setPoweredLED (boolean on)`

CameraInterface

```
public interface CameraInterface {
  // Note that recording/playback operations
  // are excluded for the sake of simplicity
public int getID();
   public void setID();
   public Object getView();
   public boolean panRight();
   public boolean panLeft();
   public boolean zoomIn();
   public boolean zoomOut();
}
```