Test Plan

# Home Alarm module: Home Alarm, and Sensors, by Chad Ingram

The plan is to test for expected output from the selected methods in the order of how the information/ feedback is created or generated. Initially, I will want to test the detector classes to make sure that the information that the classes provide is useful, honest, and fair.

**The Detector** classes have a member variable of type string for the location in which they are placed. This class also has a Boolean flag that acts as the alarm status. We will want to flip this flag for this sensor if the sensor is trigger. By default, this flag is set to false, meaning the sensor has not been tripped. There is a method to generate a random number to simulate the detector being “tripped.” I have a setter and a getter for both the location and the alarm status.

I would like to vary the randomness of the number generator up and down for whole module testing purposes, but on an individual testing basis, have the number entered be matched in a test less than or equal to 20% of the time.

Lastly, I would like to test the time that a detector takes to return the “checkStatus.”

Testing the **Detector**:

1. I expect the “get location” method to return the string that is entered.
2. I expect the “check status(get status)” method to return false(good!) at a high percentage of the time. But we do expect failure from time to time because we have a method that will feed into the “check status” method, so that will be tested as well.
3. I expect the “generate random num” method to return a random integer.
4. I expect the “generate random num” method to match an expected value 1 out of 5 times, but only to reign in the over simulation of the home alarm system.
5. I expect the time of the processes would be less than 1 second.

The **Home alarm** consists of sensor objects: CO2 detector, Motion Sensor, Smoke Sensor, and Window/Door Sensor. It implements a list to keep track of each type of sensor object that is created in the overarching structure of the home alarm. Then there is a printed list of actions when the system simulation is running. I want to test that the varying sensor object lists are being updated correctly. I will have one master list that tracks and logs all of the action in the home alarm, including the setting of the system mode by detector, that detector’s location, and the type of detector that is firing. And I have another list that will clear before the action is performed again, which will be more of an immediate report to the user.

Testing the **Home Alarm**:

1. I expect for “sensor objects” to be created with the proper data in them, and to be entered into the correct list according to their type.
2. I expect that the “ Check sensor”” methods will check it’s appropriate sensor for it is Boolean flag response(note: if “check status” return false then the system mode stays on safe, otherwise mitigate STATUS response. To “WARNING” or “EMERGENCY”)
3. I expect for “action Monitor” to run the sensor checks one after another(or at the same time).I will need to find a benchmark for how many times the “action monitor” needs to be called to change the status of the alarm system.
4. I expect for the “list of actions” to be populated after every “action monitor” method is called. So, we will test to see that the lists’ size in not zero.

\*\* I expect to add another level of abstraction called “**HOMEALARM**” to interact/interface with the monitor module. I currently need to figure out how to demo the system once it is set up. I want to send the “filtered list” up to the monitor, make the “list of actions” available for retrieval, send messages to the monitor to dispatch for appropriate help.

Testing for **HOMEALARM:**

1. I expect for the creation of a “Home Alarm” to , well, create a “Home Alarm,” and store it in a list of home alarms with a key of some sort.(id number, or user name)
2. I expect to simulate an emergency message to send to the monitor
3. I expect the ability to set the status of the “home alarm.” (for simulation and arming/disarming)
4. I expect the ability to pull the collection of the monitor report. (“list of actions”)