SMART HOME - SAFETY

Sahil Babel - 261315

Luca Pattavina - 262449

Gagana - 267359

Course: Software Engineering for Autonomous Systems
University of L'Aquila
A.Y. 2019-2020

Prof. Davide Di Ruscio

UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

Introduction

Home Safety measures can be classified into two types, external safety measures - such as protecting the home from thieves and external disasters, internal safety measures - such as fire accidents, gas leakage and water leakage. Autonomous systems can be used to automatically monitor and take measures to handle such accidents, and in our project we are focussing on building a Smart Home Internal Safety System using the technologies and methodology used to build efficient autonomous systems.

Objective

In this project, we are focusing on implementing the safety aspect in a smart home automation system. We are trying to detect and solve a few kinds of internal safety threats (unlike the external safety threats like thieves and intruders) that can occur in a basic home environment.

The possible safety threats that we are trying to solve in this project are :

1. **Gas Leakage or excess smoke in the house :**

   While one of the most commonly used resources in a home environment is the LPG gas, there are high chances for a gas leakage which can end in a harmful explosion. A lot of accidents have always been reported on this aspect. Also, excess smoke inhalation causes severe health issues. So, Through this project we try to prevent this scenario by implementing a proper gas/smoke leakage detection and explosion prevention system.

2. **Fire Accident in the house :**

   Fire is the most widespread cause of death by accident in a home scenario. Fire affects thousands of residents each year, resulting in injury and loss of life. Fire accidents can occur due to electric short circuits or even from a stove flame etc. Through this project, we try to prevent this scenario by implementing a fire detection and fire stopping system.

3. **Water Leakage in the house :**

   Water damage claims are on the rise due to faulty plumbing materials, failed connections, human error, and burst pipes. So, to avoid any property damage or wastage of water, through this project we try to implement a water leakage detection and handling system.

4. **Wastage of Resources :**

   LPG gas and water are resources that are not very efficiently used in a home environment. Sometimes we forget to turn off the stove knob and sometimes we forget to turn off the water taps as well. So, one of the goals of this project is to prevent wastage of gas and water as well.

Technology Specifications:

Our system will consist of the following devices,

1. **Gas/Smoke Sensors -** They are used in gas leakage detecting equipment in home and industry, are suitable for detecting LPG, natural gas , town gas, cooking fumes and cigarette smoke.

2. **Fire Detector Sensors -** A fire detector is a sensor designed to detect and respond to the presence of a flame or fire.

3. **Temperature Sensors -** Temperature sensors are used to help in fire detection, because fire is a combination of flame, high temperature and smoke.

4. **Flow liquid meter sensors -** Flow liquid meter sensor is one of the sensors that is used for this monitoring process. This sensor uses the Hall Effect sensor inside it to measure the water flow rate and is placed on a pipe that has a diameter equal to the diameter of the sensor.

Scenarios:

1. **Turning off the Stove knobs when it is just left on for a long time and Gas leakage** : Gas Sensor detects that a gas stove was left on for some period of time without using it and the percentage of gas is high in atmosphere of room then the control unit will instruct the actuator to close all the stove knob which are on (If the gas leakage was due to stove it will get avoided).

2. **Starting Home Fire Sprinklers when Fire detected at home** : If the sensed value of a fire detector is positive, it means that fire is detected at home and our control unit will instruct the home water sprinklers to turn on.

3. **Turning off the water taps when it is just left on for a long time** : If the flow liquid meter sensors sense that there is continuous water flow in the pipeline for some period of time then our control unit will instruct the actuator to switch off the tap to save water.

4. **Opening of windows when the temperature is high** : If the sensed values from the temperature sensor is high then the control unit will instruct the actuator to open all the windows of that room.

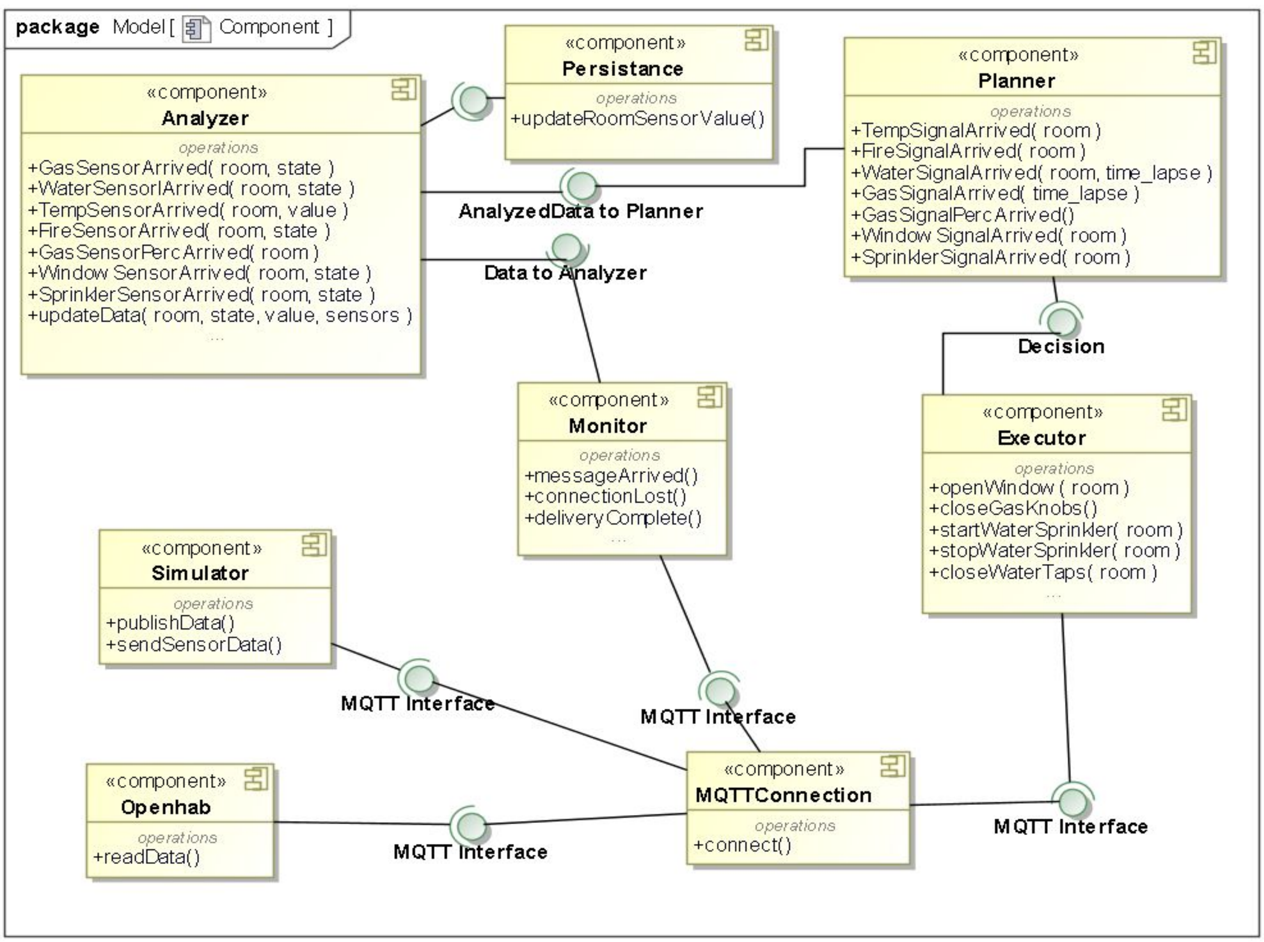
Diagrams

The following diagrams explain the functionality of operations and implementation of the system:

Component Diagram

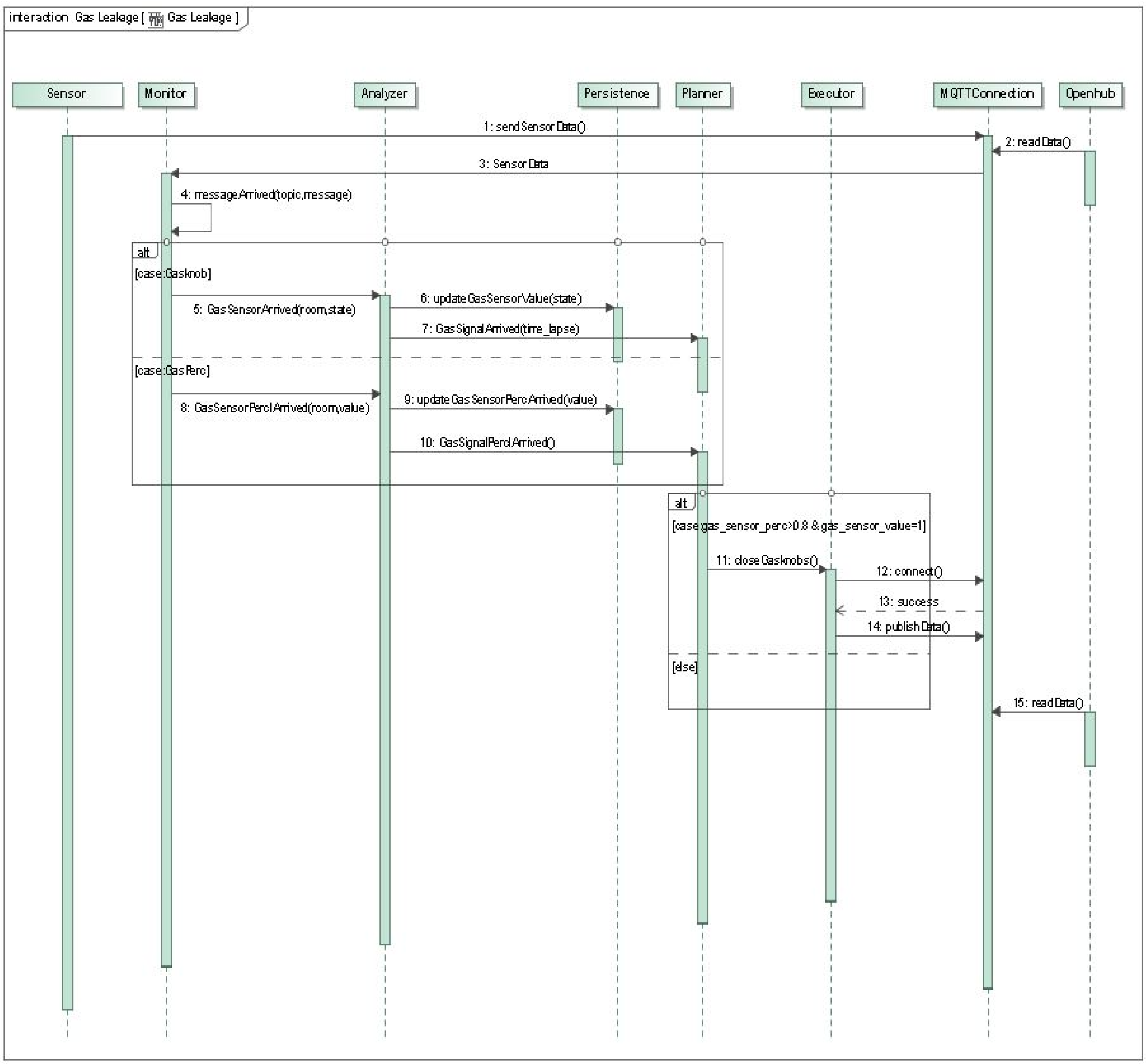
The proposed framework is based on the MAPE-K model. The model consists of five components that communicate with each other. we realise the (1) **monitoring** and (2) **analysis** as a function of the Monitor process.Monitoring of sensor data is a function of Monitoring process and Analyzing of sensor data is a function of the Analysis process while the **(3) Plan** and **(4) Execute** processes realise the system recovery by performing some necessary operations..

Another important component is 'OpenHAB'. From this component we're able to check the sensor status and values and also modify the state of some components. The 'MQTT Connection' component takes care of starting, managing and closing a connection MQTT.

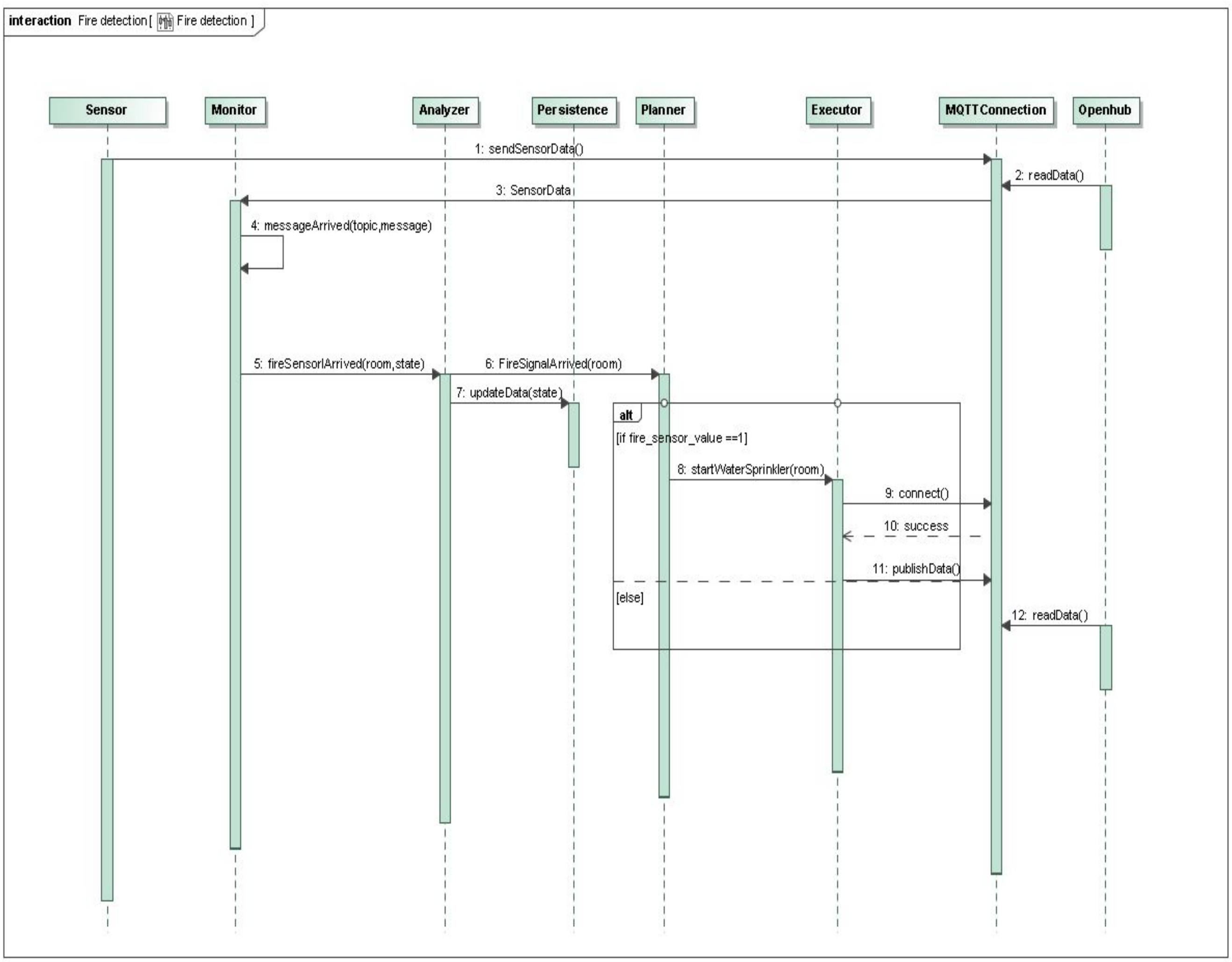


Sequence Diagrams

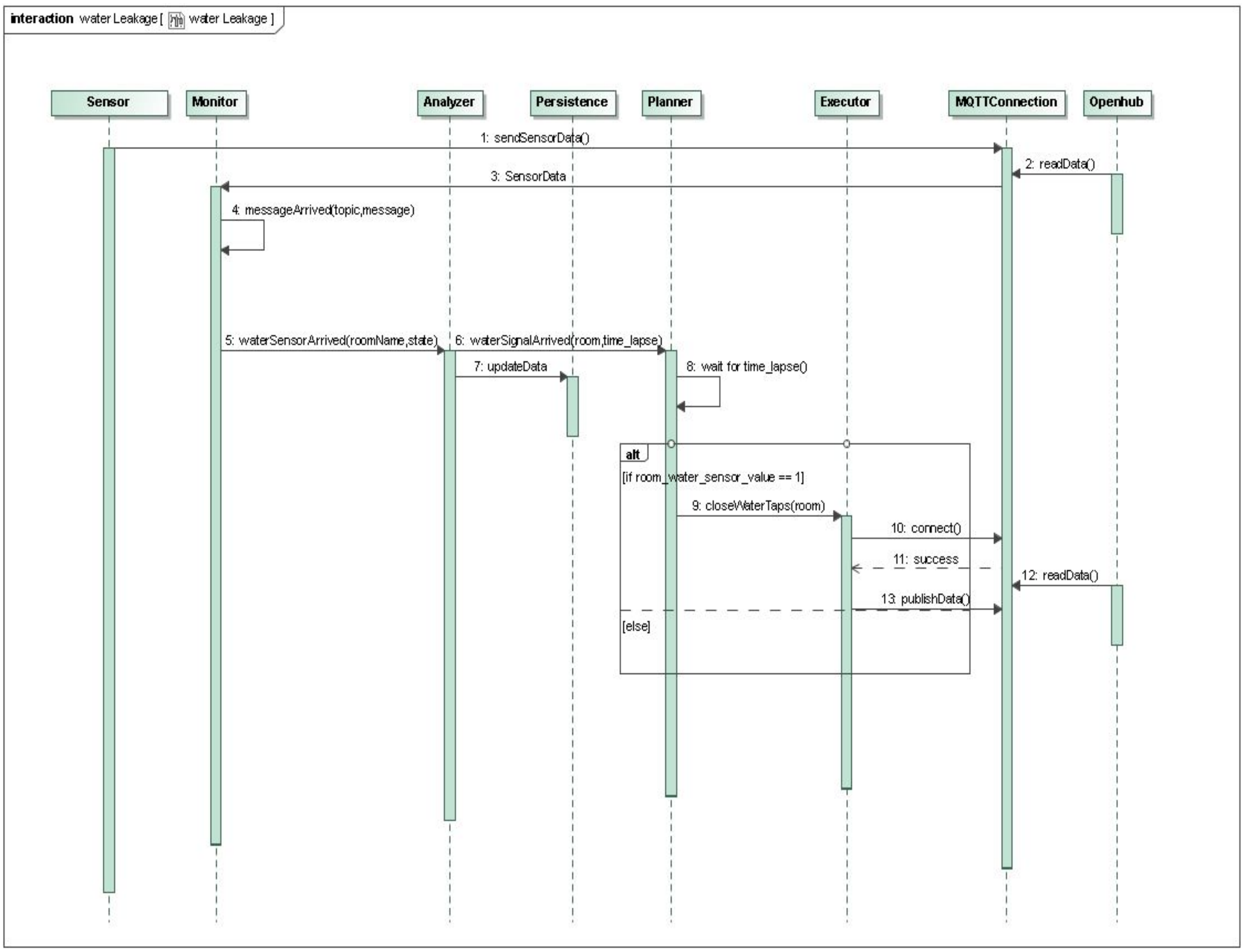
The following sequence diagram shows how our system works with the data given from the gas sensors. We have two types of gas sensor, one that gives the state of the gas knob (open or closed) and one that sends the quantity of the gas in the air. If the gas is open for more than a given threshold, in our case 10 seconds just as dimostration, it closes the gas. Also if the gas it's on and the gas percentage in the room's air reaches 0.8 ‰ the system will automatically close the gas knob in the kitchen.



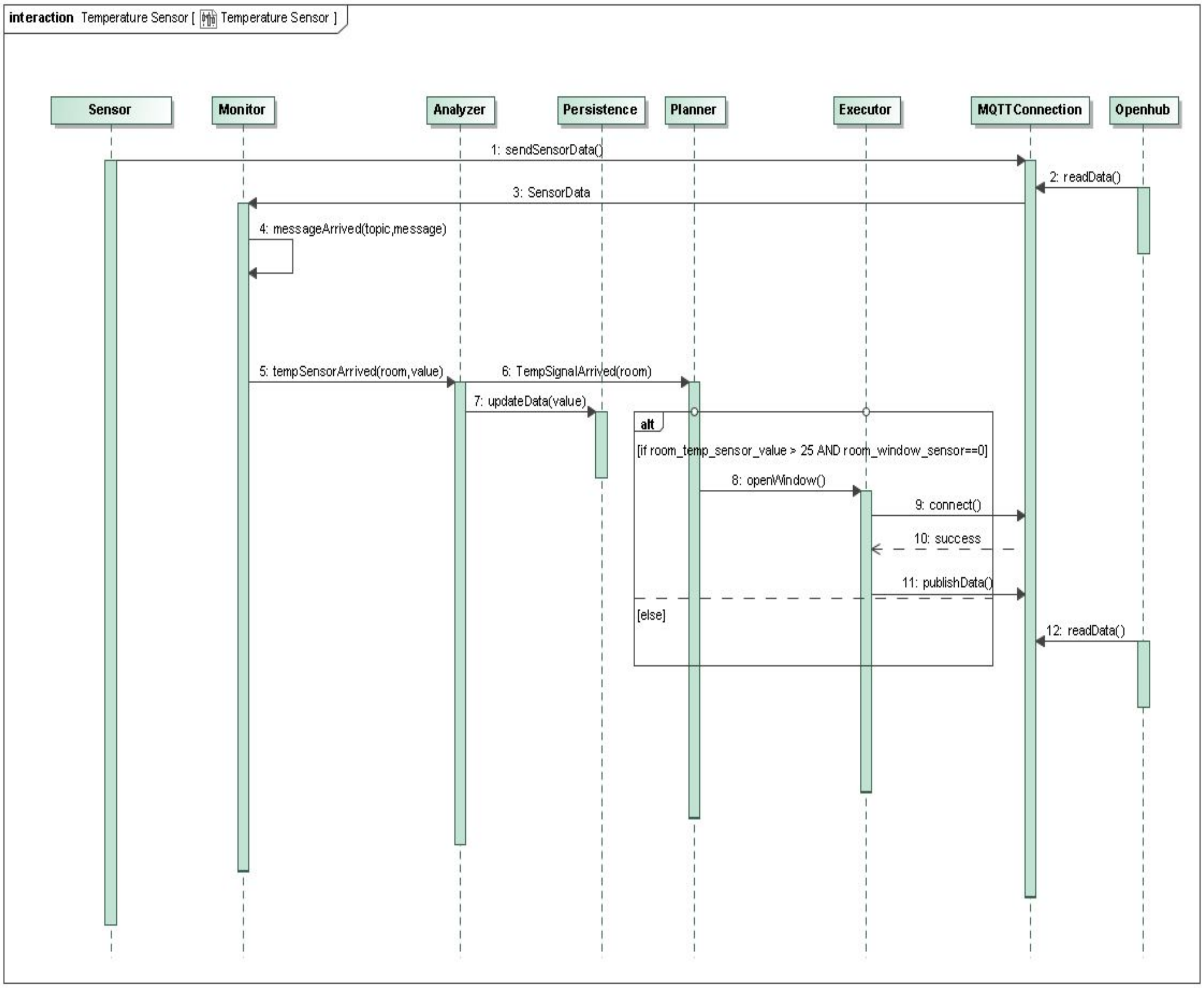
The following sequence diagram shows how our system works when fire is detected at home . Once the Fire signal is sensed, the system will automatically start the room water sprinklers to shut down the fire.
After Some time(i.e. 10 seconds in our project) water sprinklers will automatically stop to prevent flooding.



This sequence diagram shows how our system works when there is continuous flow of water. Based on a given time_lapse (hard coded in our system), once the water signal is sensed, it waits for this time lapse and, if the water is still on the system will automatically close the water in the specified room.



The following sequence shows how our system works when temperature is high. The MAPEk part of our system will read all Temperature sensor values, if it crossed the threshold (i.e. 25°C in our project) then (if the window it's not already opened) , the system will automatically open the windows of that room.



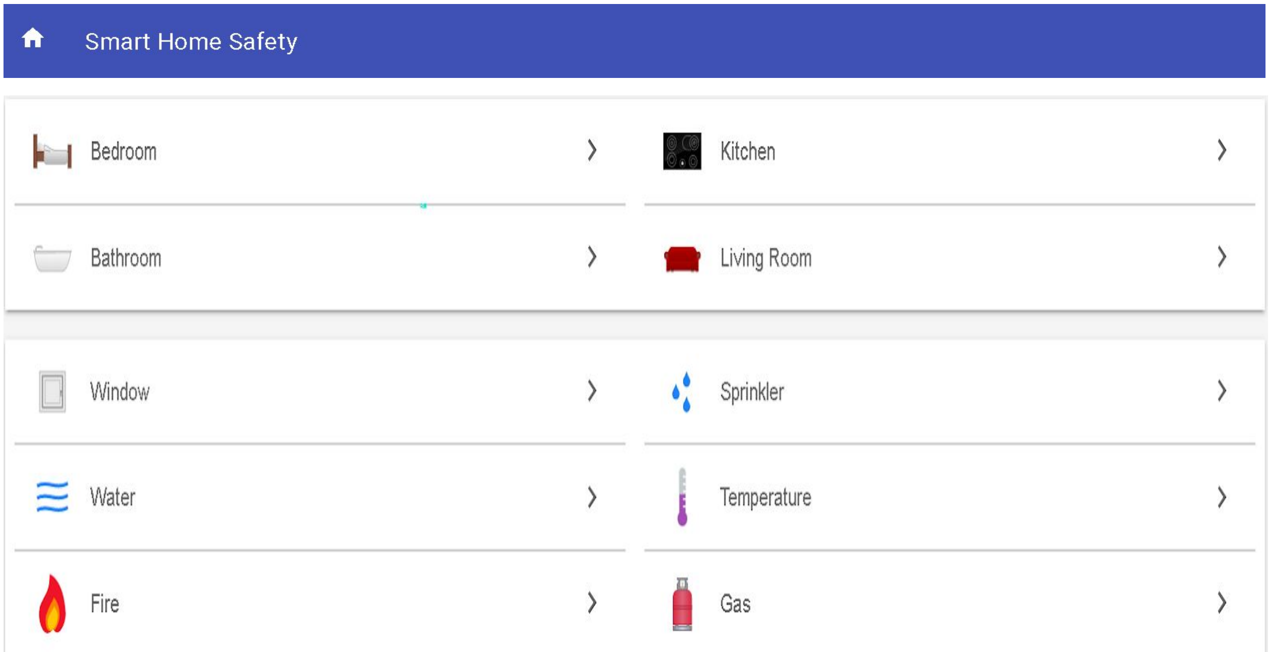
The Application

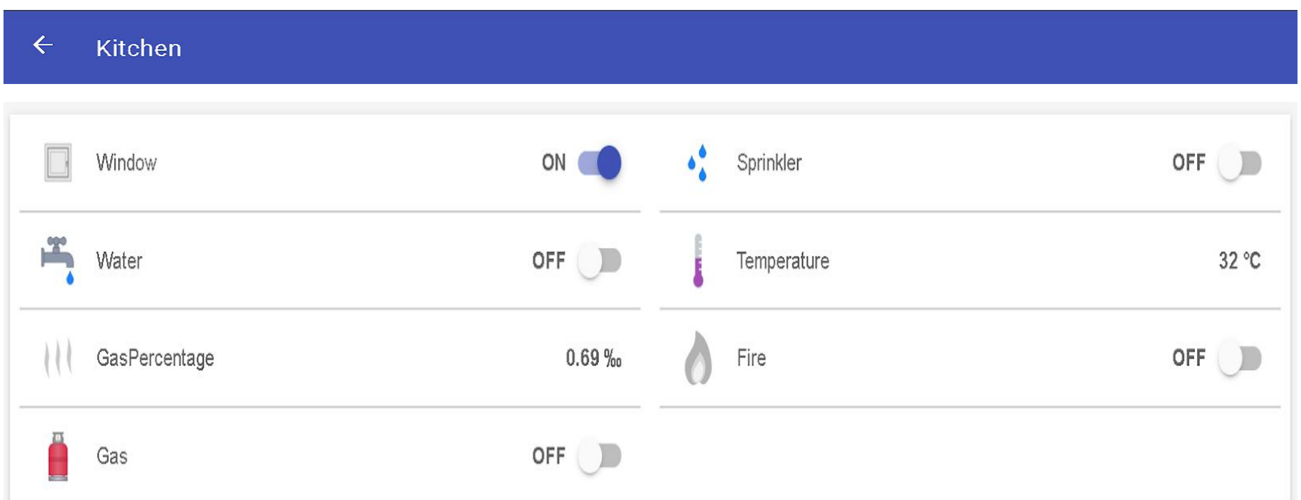
For the development of the application, as seen in the diagrams, we used three different applications, all linked by the MQTT Broker.

Starting from the simulation part we decided to use, after having different problems with the OSGi bundles, **Python**. The SensorSimulation.py sends every second the **temperature** value of the four rooms and also the **gas percentage** of the kitchen. Each sensor, described as a class in our project has two different methods to create and provide the message to send to the right MQTT topic. The **standard method**, given a base value, e.g. 20° for the temperature, raises or reduces this value by 0.1° with a 50/50 probability. The **"raised" method** raises the temperature of the sensor by 0.5° until it arrives at 32° just to see the system open automatically the window.

The main application, developed in **java**, contains the Monitor, Analyzer, Planner and Executor with a file used to keep track of the sensor values and their states. The functionality of this application is all described by the sequence diagrams in the above section.

For the UI part we decided to use **OpenHab**, deciding thought to program and set up his environment all by the **code** and not using the visual console to set up all the different components. We also decided during the development of the application, to use OpenHab also as a simulator, using the possibility to control the components to trigger different behaviors of the system, the one not simulated by the python code.


Smart Home Safety
Bedroom
Kitchen
Bathroom
Living Room
Window
Sprinkler
Water
Temperature
Fire
Gas


Kitchen
Window
ON
Sprinkler
OFF
Water
OFF
Temperature
32 °C
GasPercentage
0.69 ‰
Fire
OFF
Gas
OFF