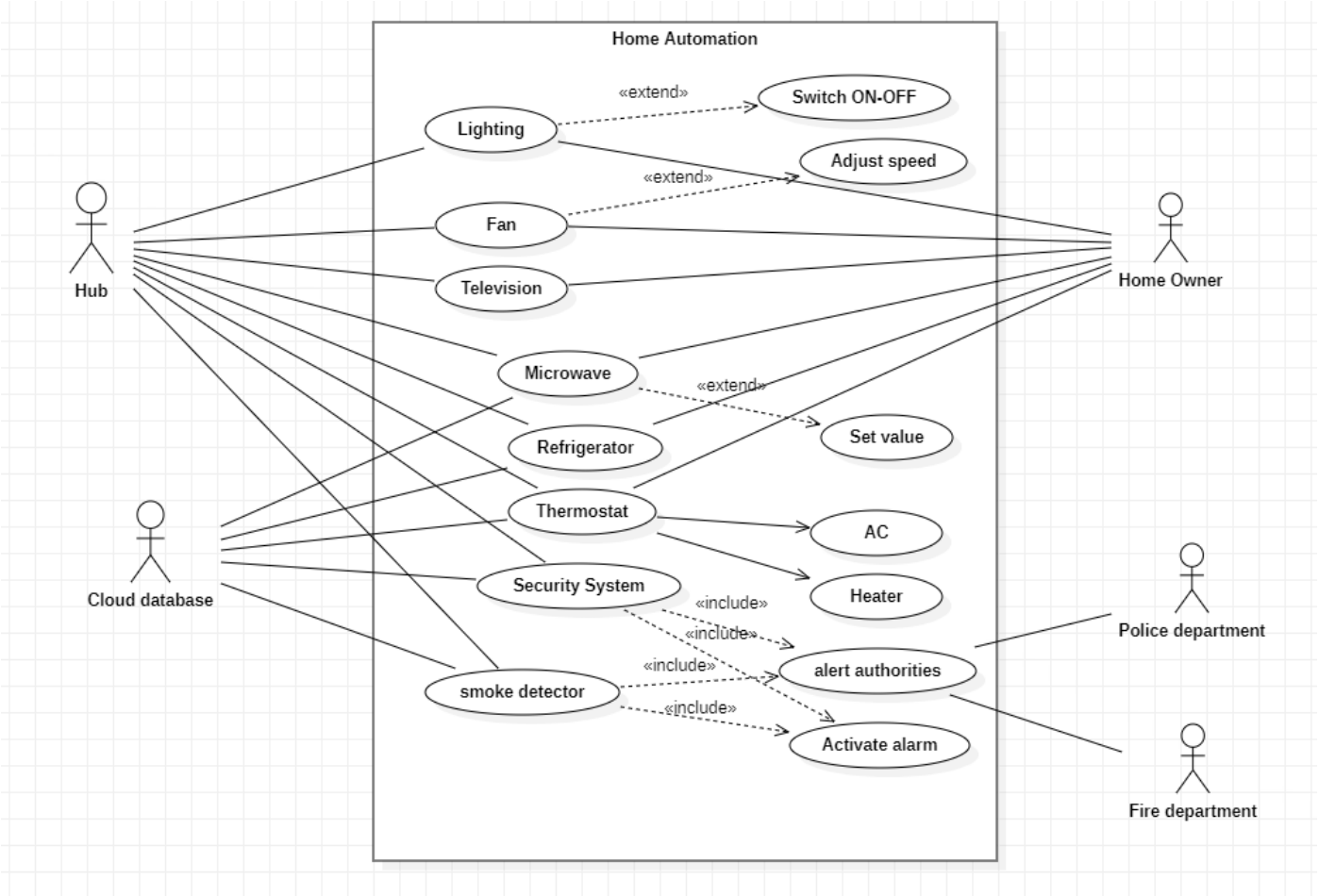
USE CASE DIAGRAM

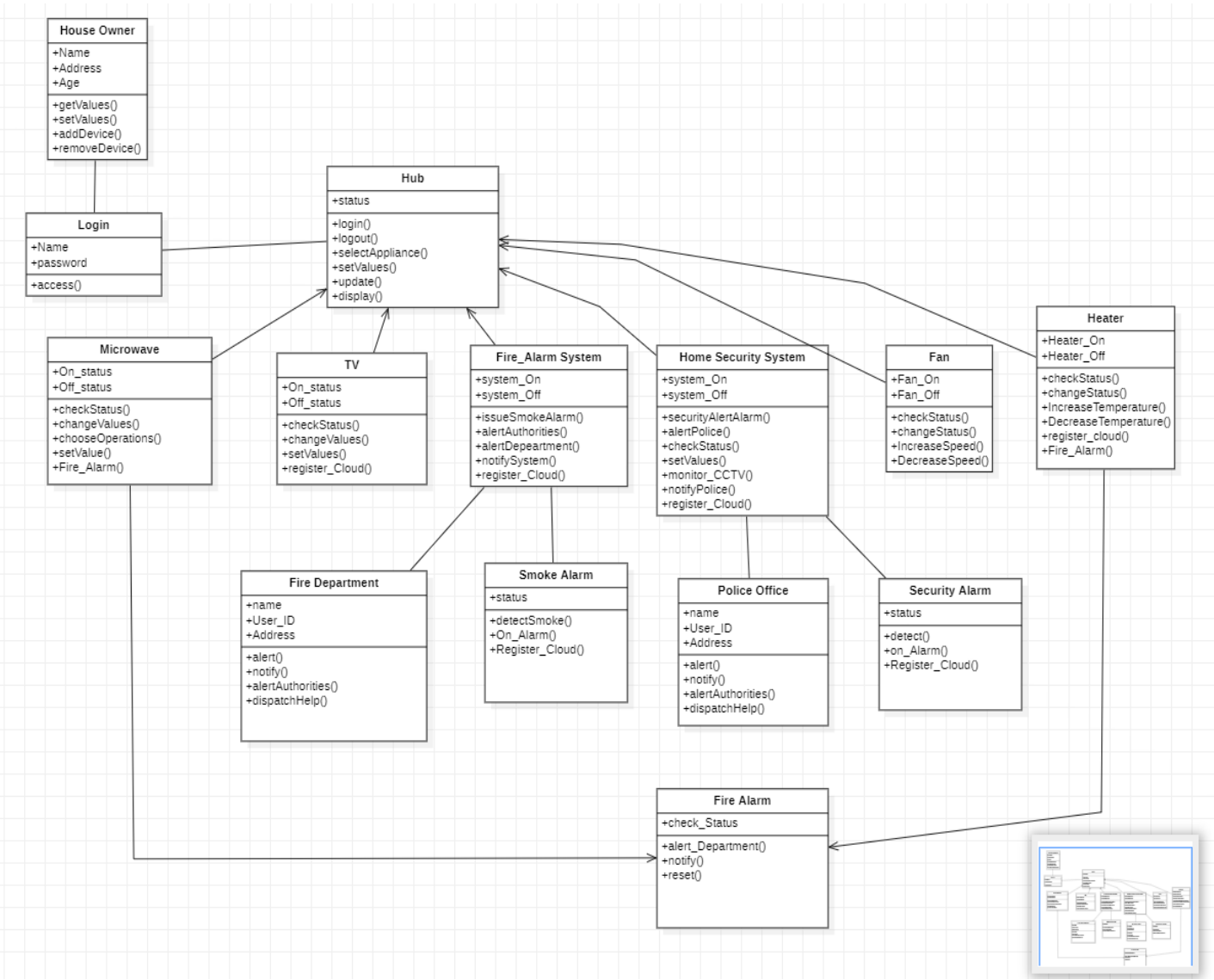
Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. The actors used in this use case diagram are Hub, Cloud database, Home Owner, Police department, Fire department and the use cases are Lighting, Fan, Television, Microwave, Refrigerator, Thermostat, Security System, smoke detector, Switch ON-OFF, Adjust speed, Set value, AC, Heater, alert authorities, Activate alarm.



CLASS DIAGRAM

A class diagram is a static diagram. It represents a static view of the program. The class diagram is used to visualise, describe, and document various aspects of the system and build executable code for a software application. You can even generate basic code sets from properly constructed class diagrams. The class diagram describes the attributes and operations of the class, as well as the constraints imposed on the system. Class diagrams are widely used in object-oriented modelling systems, as they are the only UML diagrams that can be directly mapped to object-oriented languages.

The classes used House Owner, Hub, Login, Microwave, TV, Fire_Alarm System, Home Security System, Fan, Heater, Fire Department, Smoke Alarm, Police Office, Security Alarm, Fire Alarm.

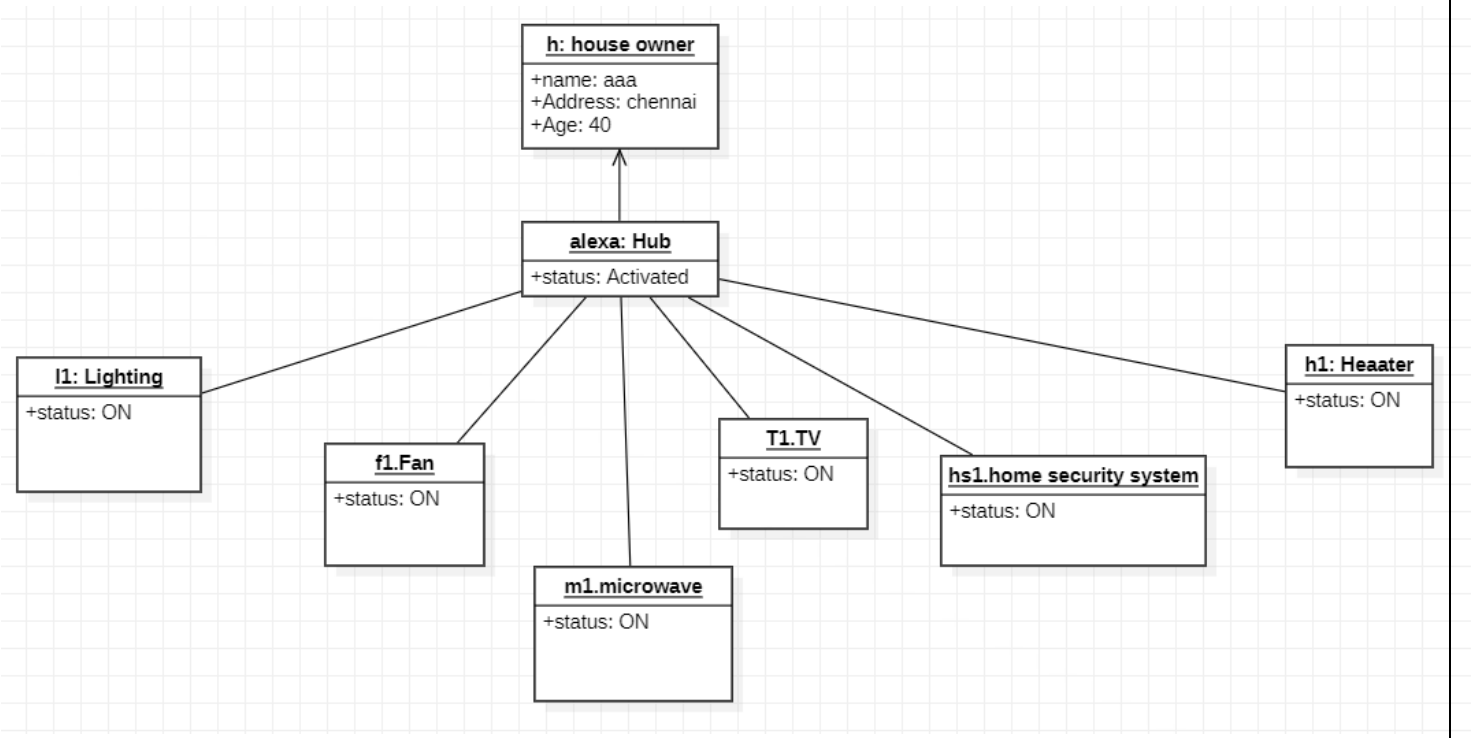


OBJECT DIAGRAM

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram.

The class house owner has the object h, the class hub has the object alexa, the class lighting has the object l1, the class fan has the object f1, the class TV has the object T1, the class home security system has the object hs1, the class Heater has the object h1and the class microwave has the object m1.

The functionality of home owner is to login with hub and get access to the appliances linked with hub.

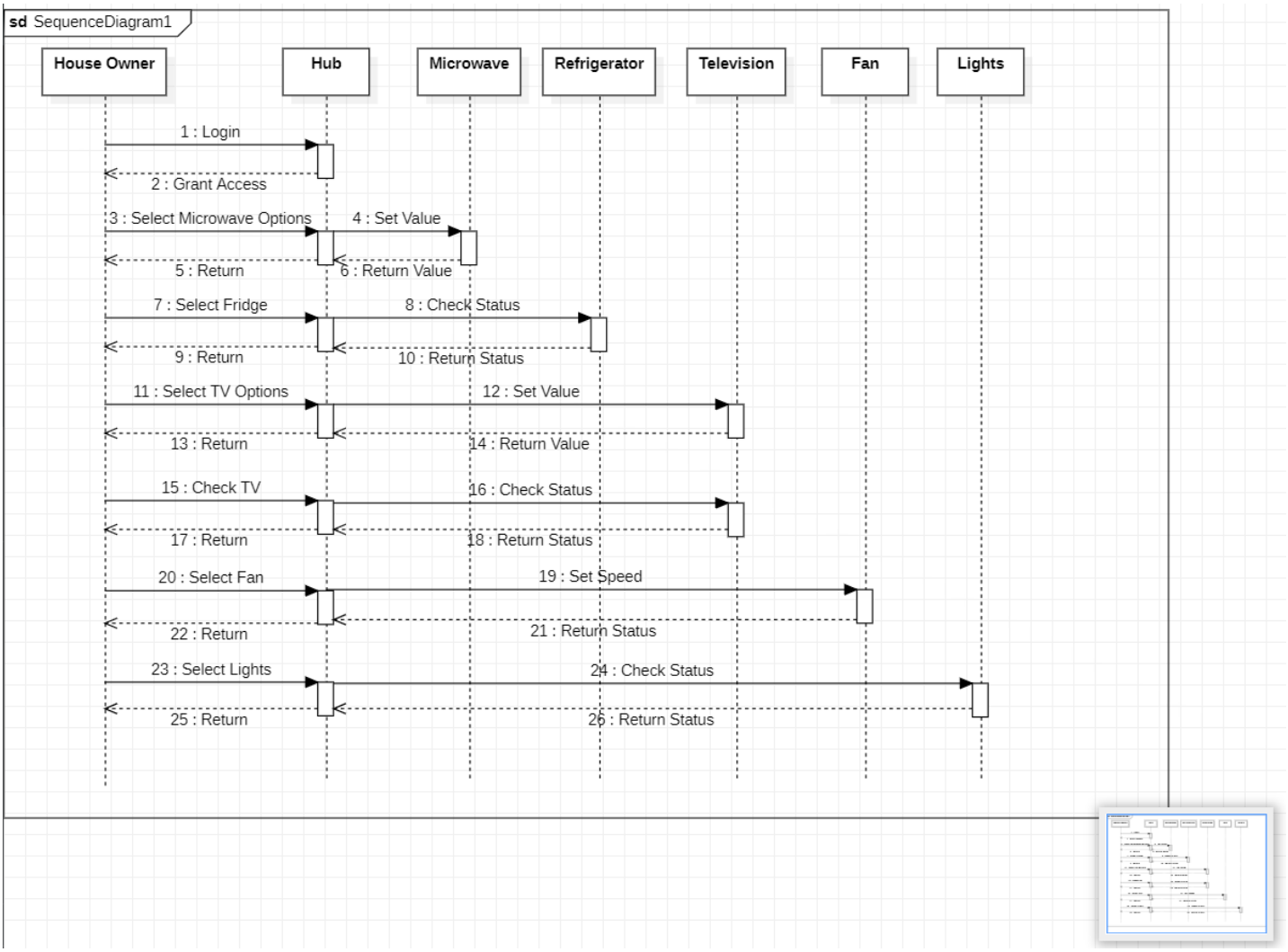


SEQUENCE DIAGRAM

The purpose of interaction diagrams is to visualize the interactive behaviour of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction.

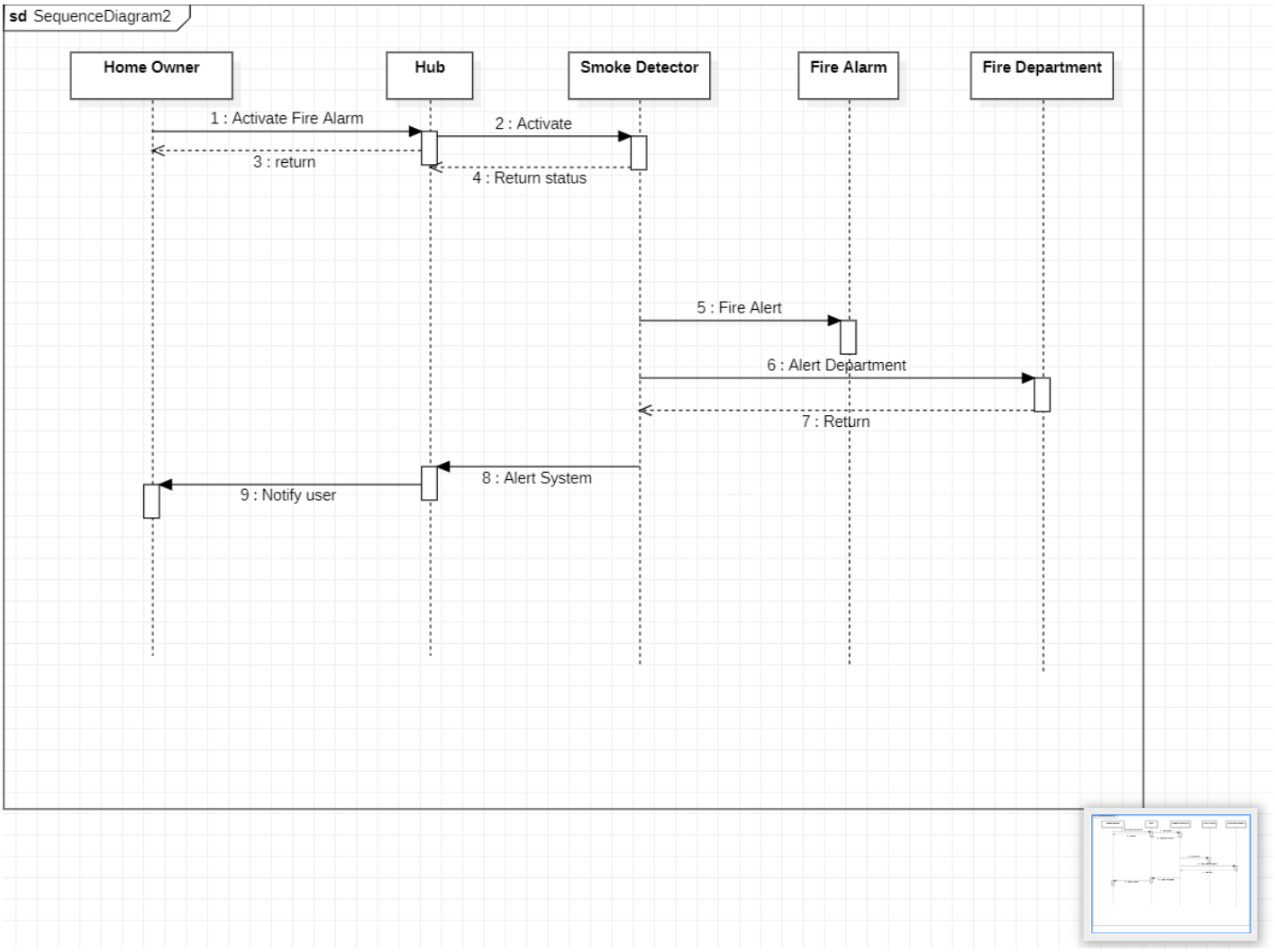
The following diagram shows the message sequence for visit the hub. It is important to understand the time sequence of message flows. The message flow is nothing but a method call of an object.

The first call is login(). The next call is select appliances(), and checks for its status. This process gets repeated for the remaining appliances.



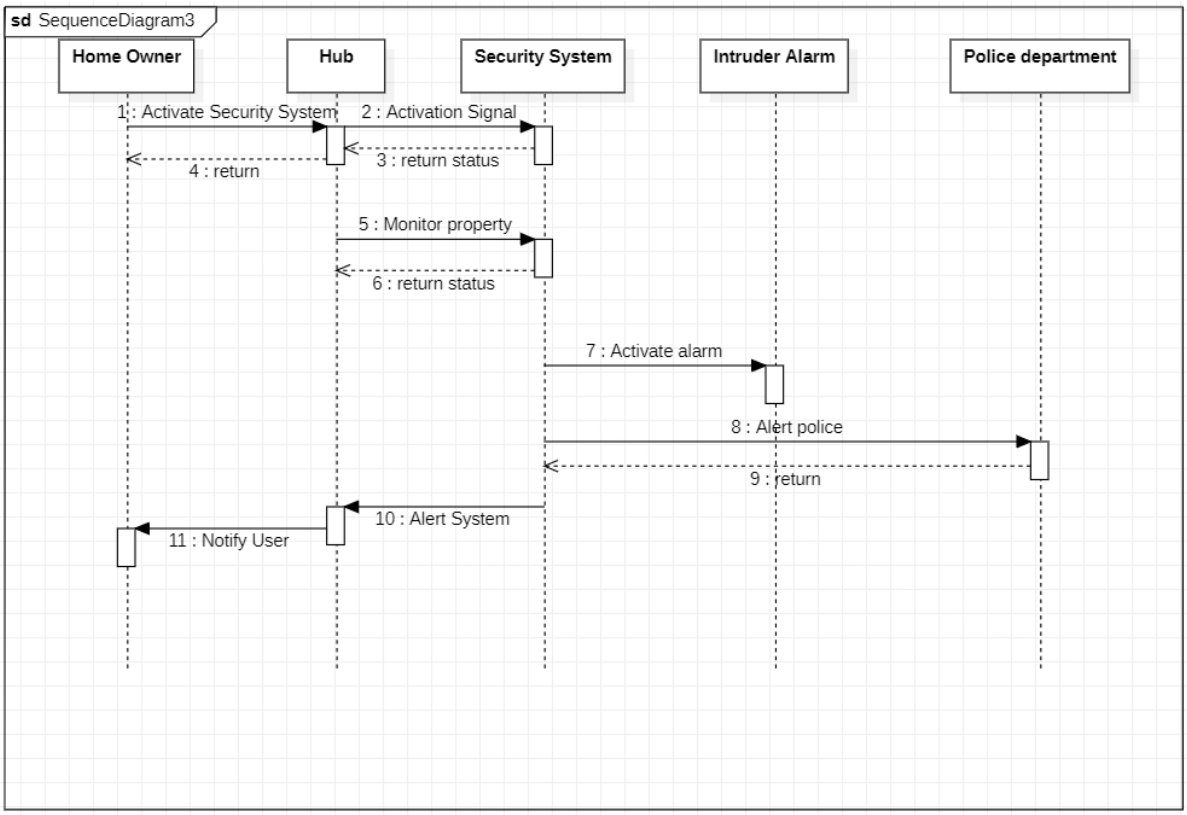
The following diagram shows the message sequence for visit the hub. It is important to understand the time sequence of message flows. The message flow is nothing but a method call of an object.

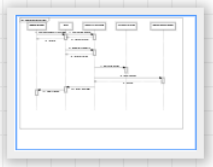
The first call is activate fire alarm(). The next call is activate alarm() in smoke detector, and returns its status. In case of fire, the alarm is activated and the details is sent to the fire department for the further actions.



The following diagram shows the message sequence for visit the hub. It is important to understand the time sequence of message flows. The message flow is nothing but a method call of an object.

The first call is activate() security system. The next call is activate signal() to security system, and returns its status. In case of unauthorized motion Intruder alarm gets activated() and the details are forwarded to the police department.



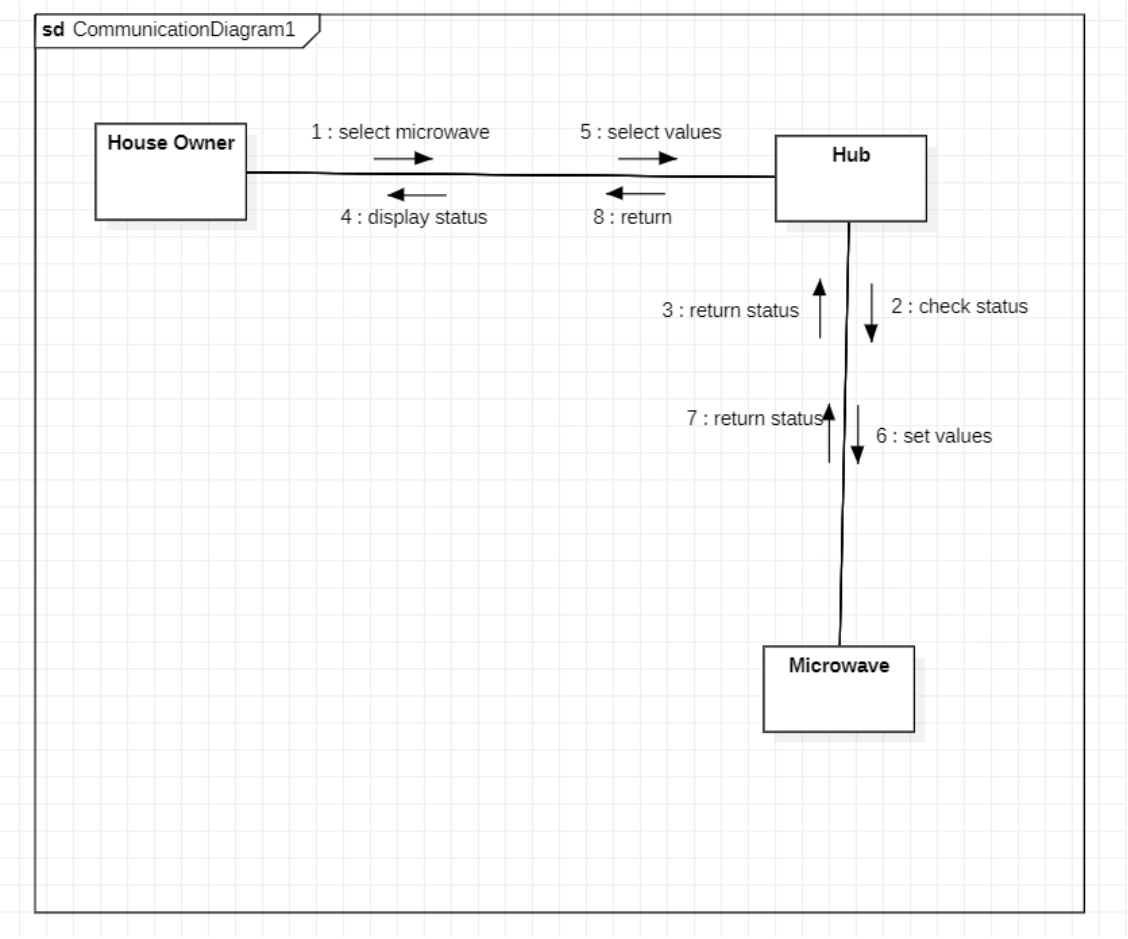


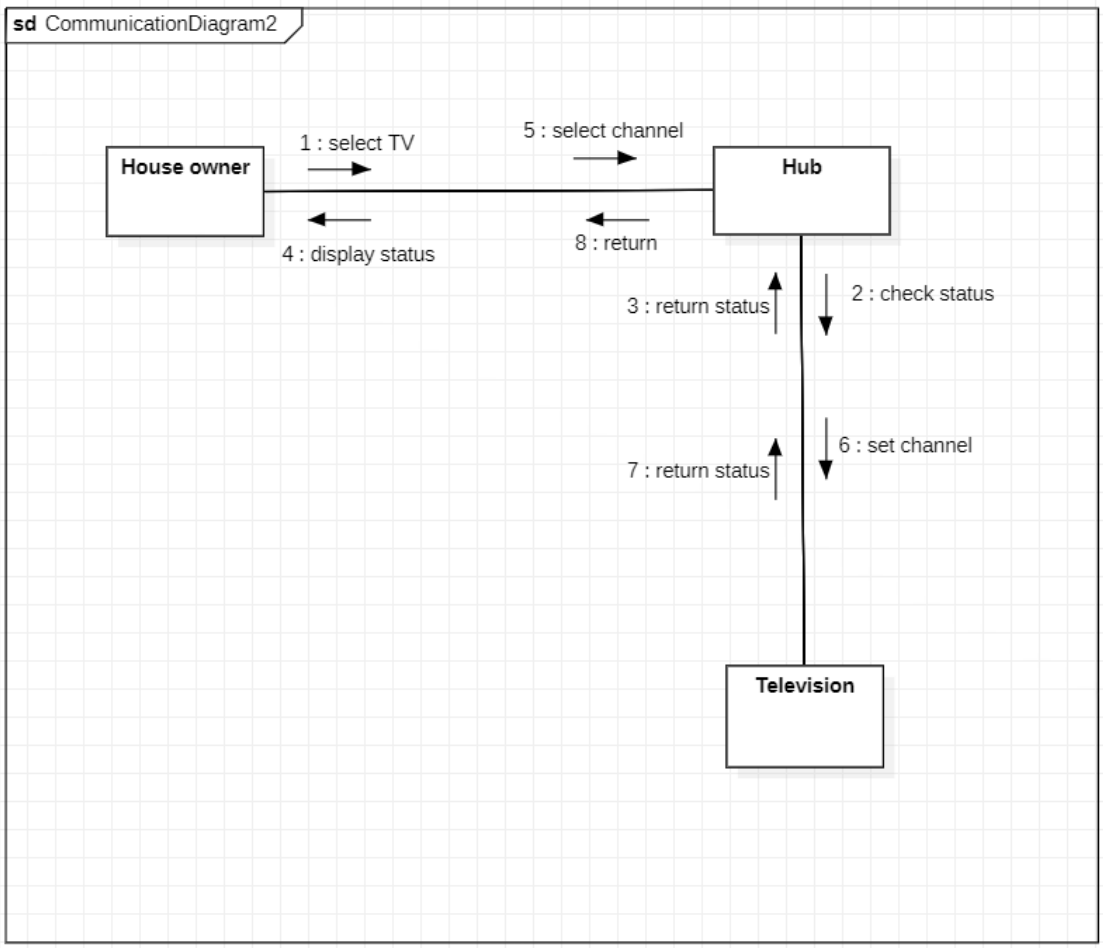
COLLABORATION DIAGRAM

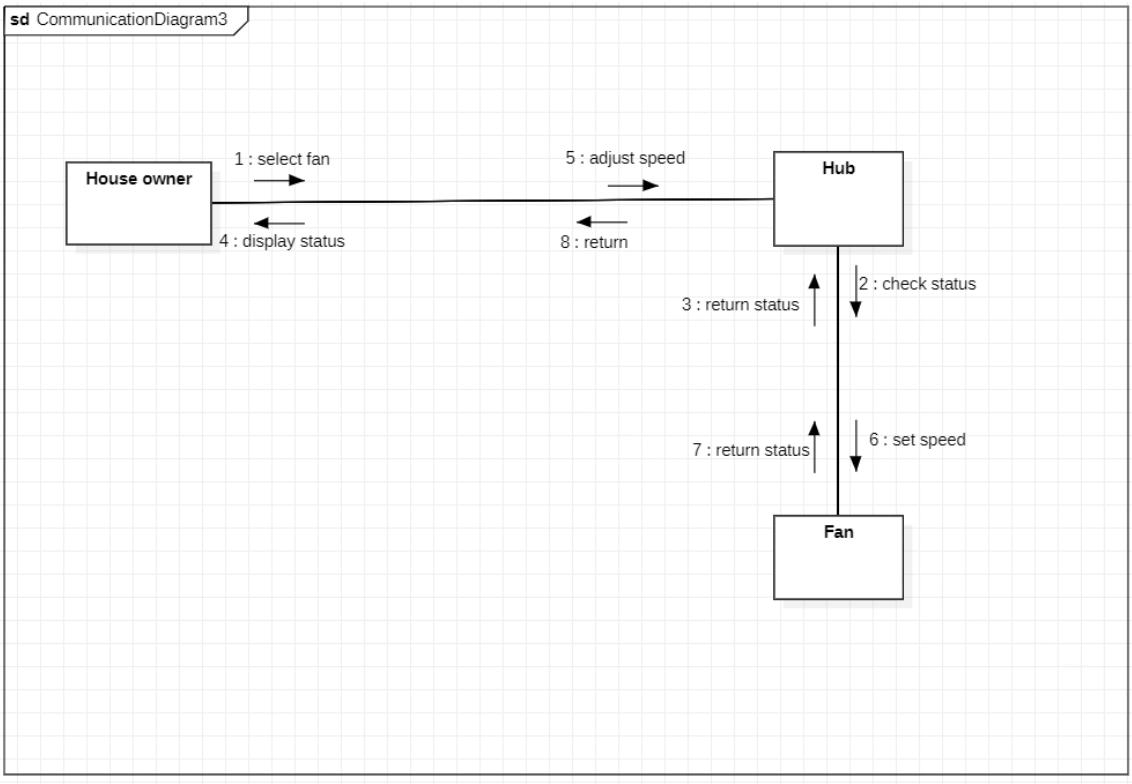
It shows the architecture of the object residing in the following diagram. In the collaboration diagram, the object is indicated by showing the flow of messages. The actor home owner will visit the hub to select the appliances and the flow of the message goes.

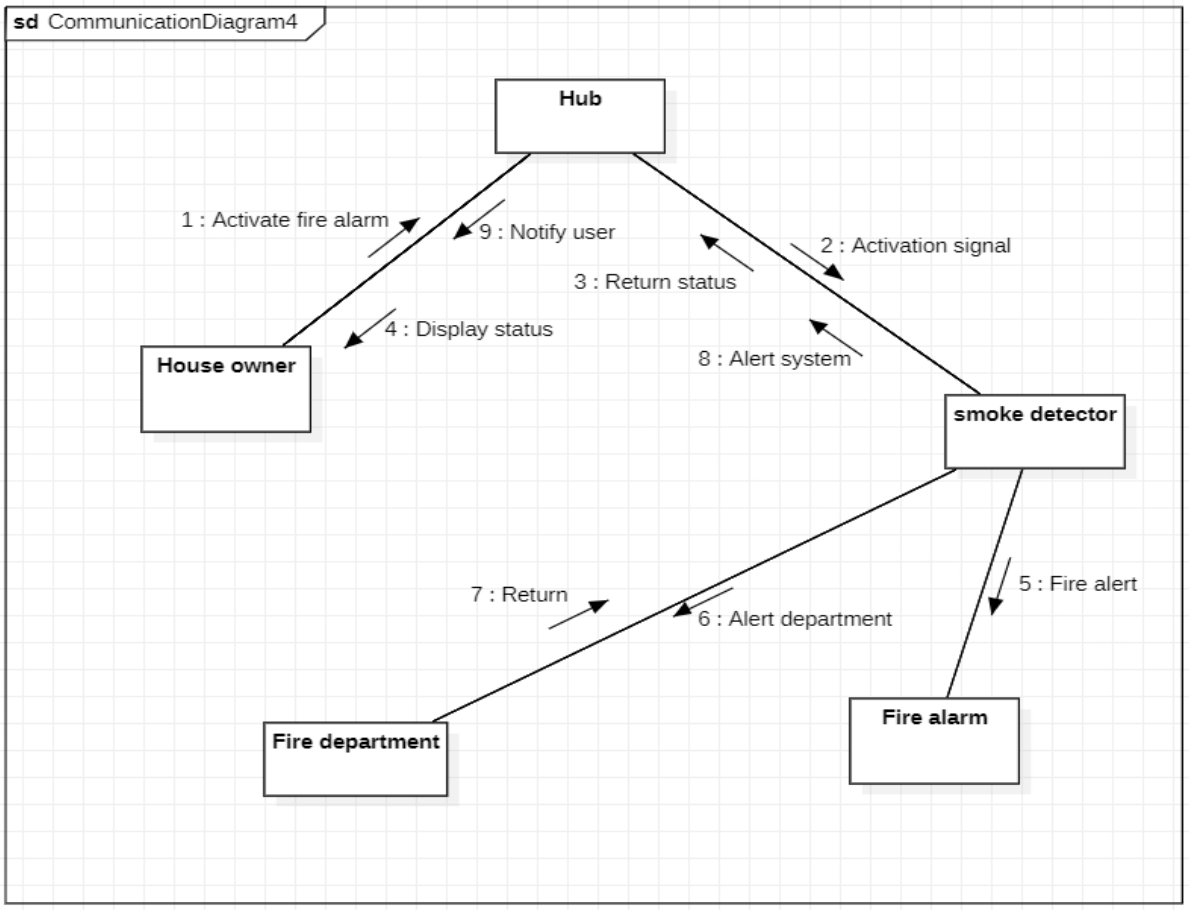
In case of fire and unauthorized motions the respective alarm beeps and the details were forwarded to the fire and police department.

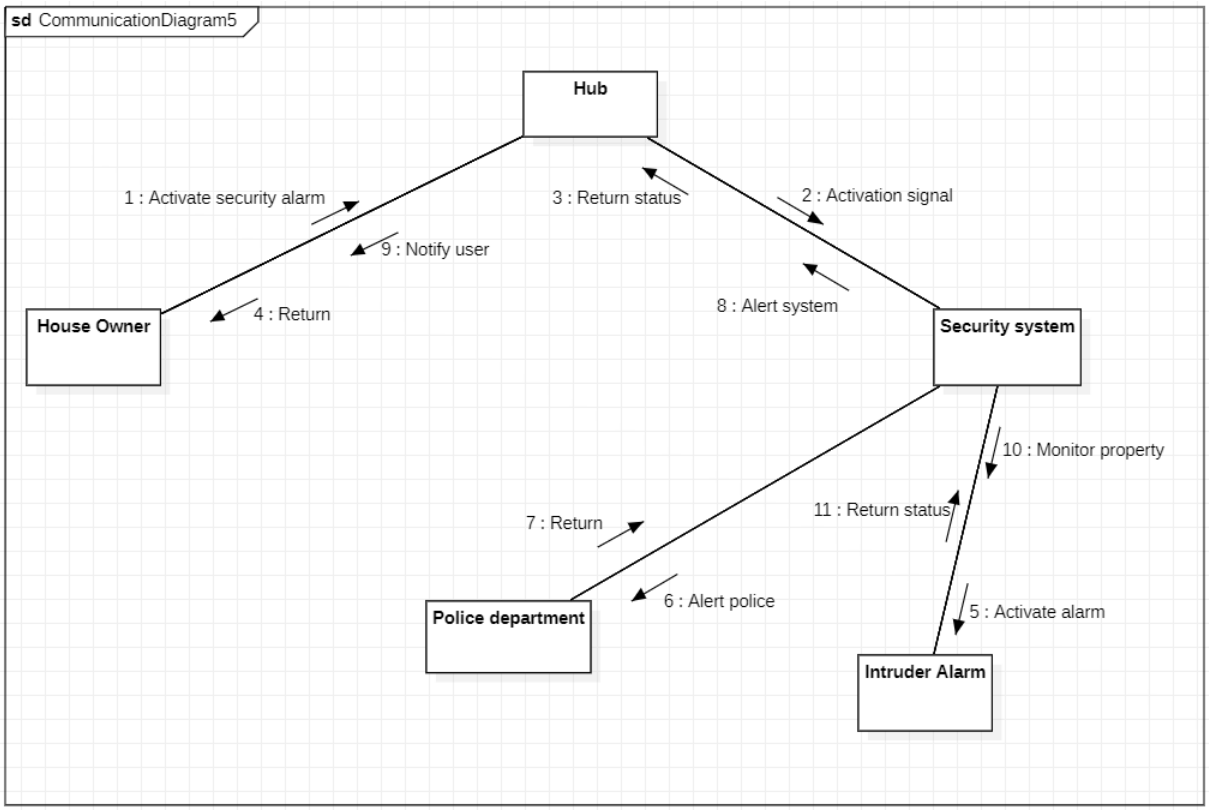
The collaboration of our project is shown below:




sd CommunicationDiagram2
House owner
Hub
Television
1 : select TV
5 : select channel
4 : display status
8 : return
3 : return status
2 : check status
7 : return status
6 : set channel


sd CommunicationDiagram3
House owner
Hub
Fan
1 : select fan
5 : adjust speed
4 : display status
8 : return
3 : return status
2 : check status
7 : return status
6 : set speed

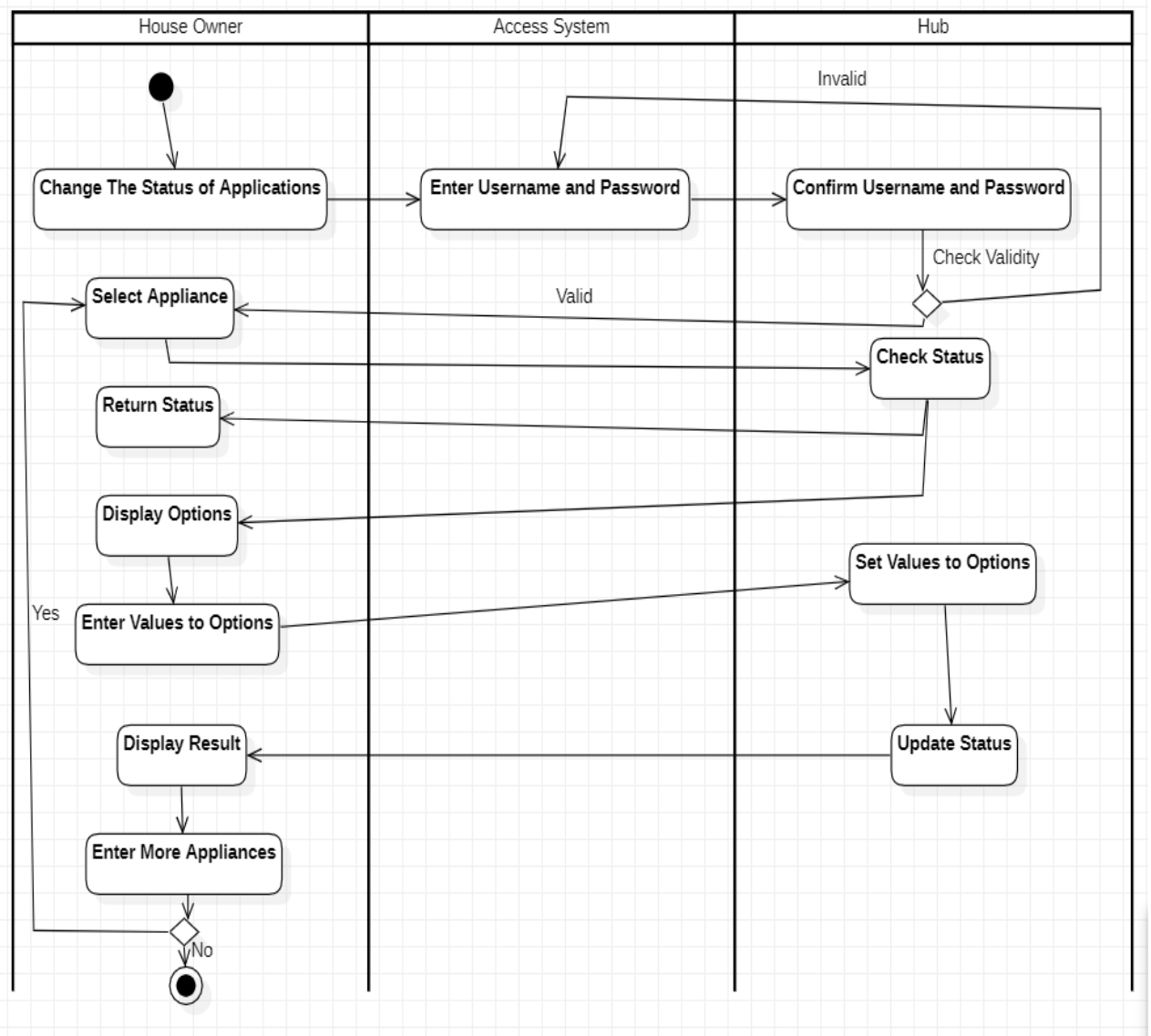

sd CommunicationDiagram4
Hub
1 : Activate fire alarm
9 : Notify user
2 : Activation signal
3 : Return status
4 : Display status
8 : Alert system
House owner
smoke detector
7 : Return
6 : Alert department
5 : Fire alert
Fire department
Fire alarm

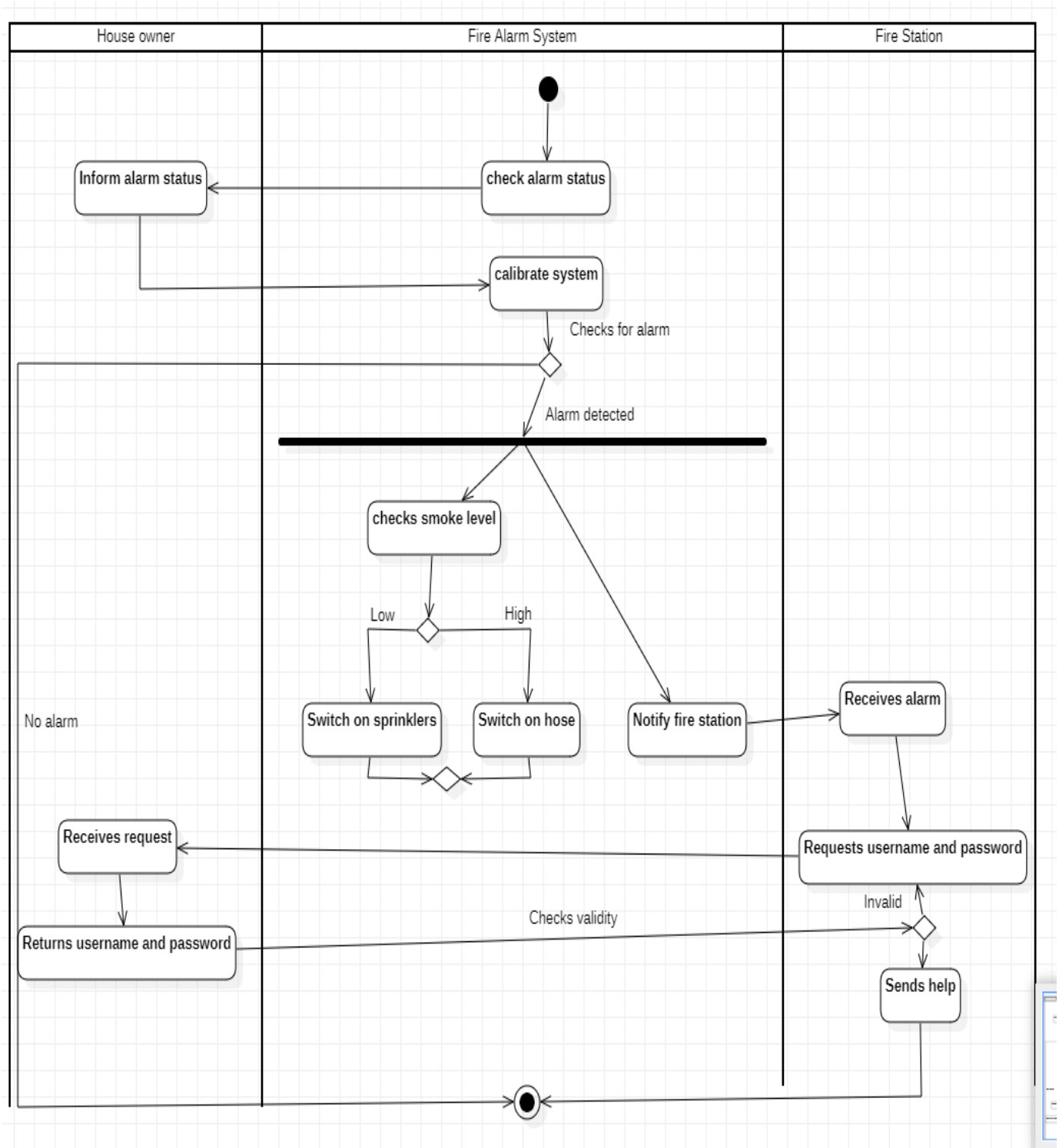

sd CommunicationDiagram5
Hub
1 : Activate security alarm
3 : Return status
2 : Activation signal
9 : Notify user
8 : Alert system
House Owner
4 : Return
Security system
10 : Monitor property
11 : Return status
7 : Return
6 : Alert police
5 : Activate alarm
Police department
Intruder Alarm

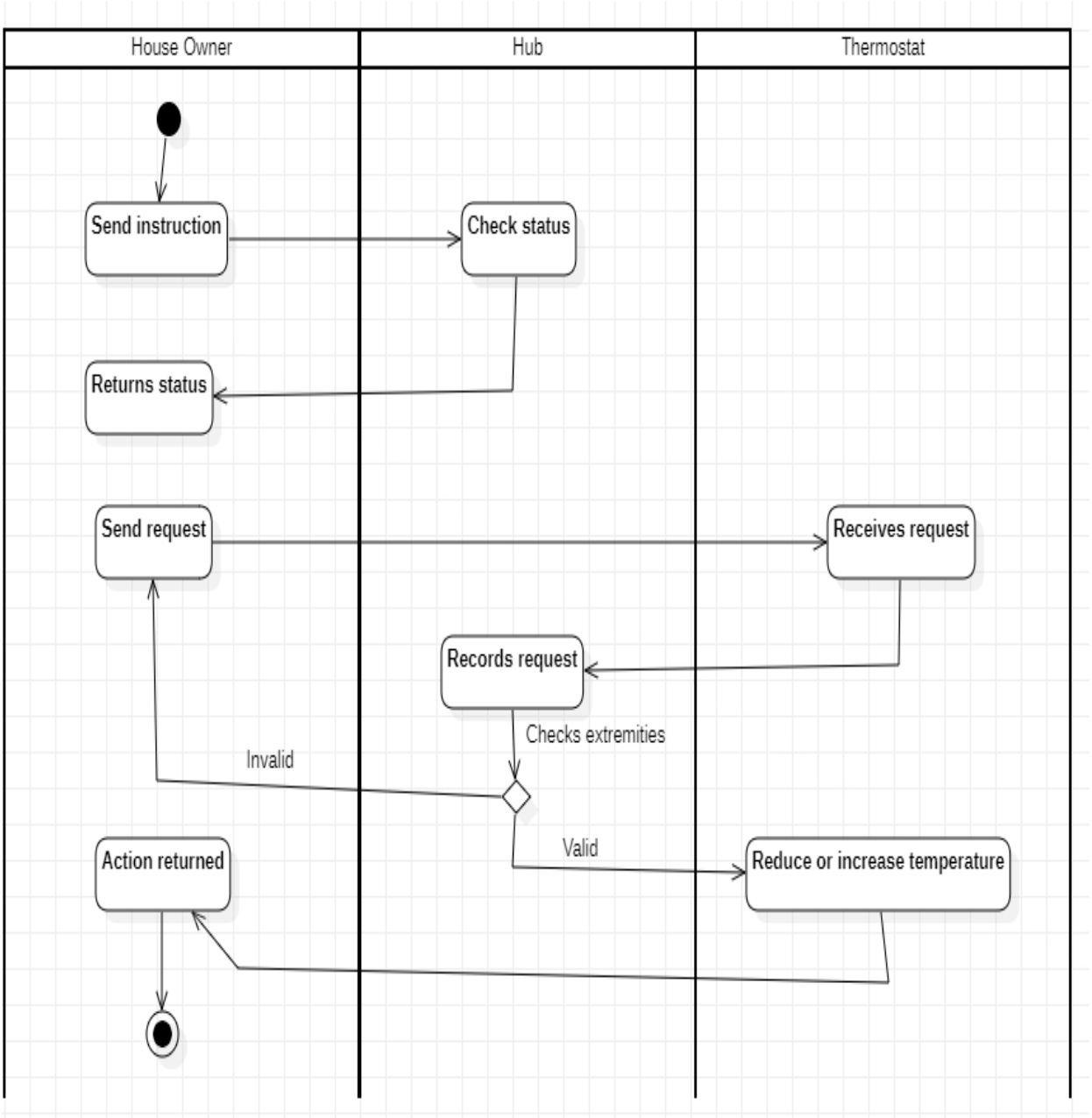
ACTIVITY DIAGRAM

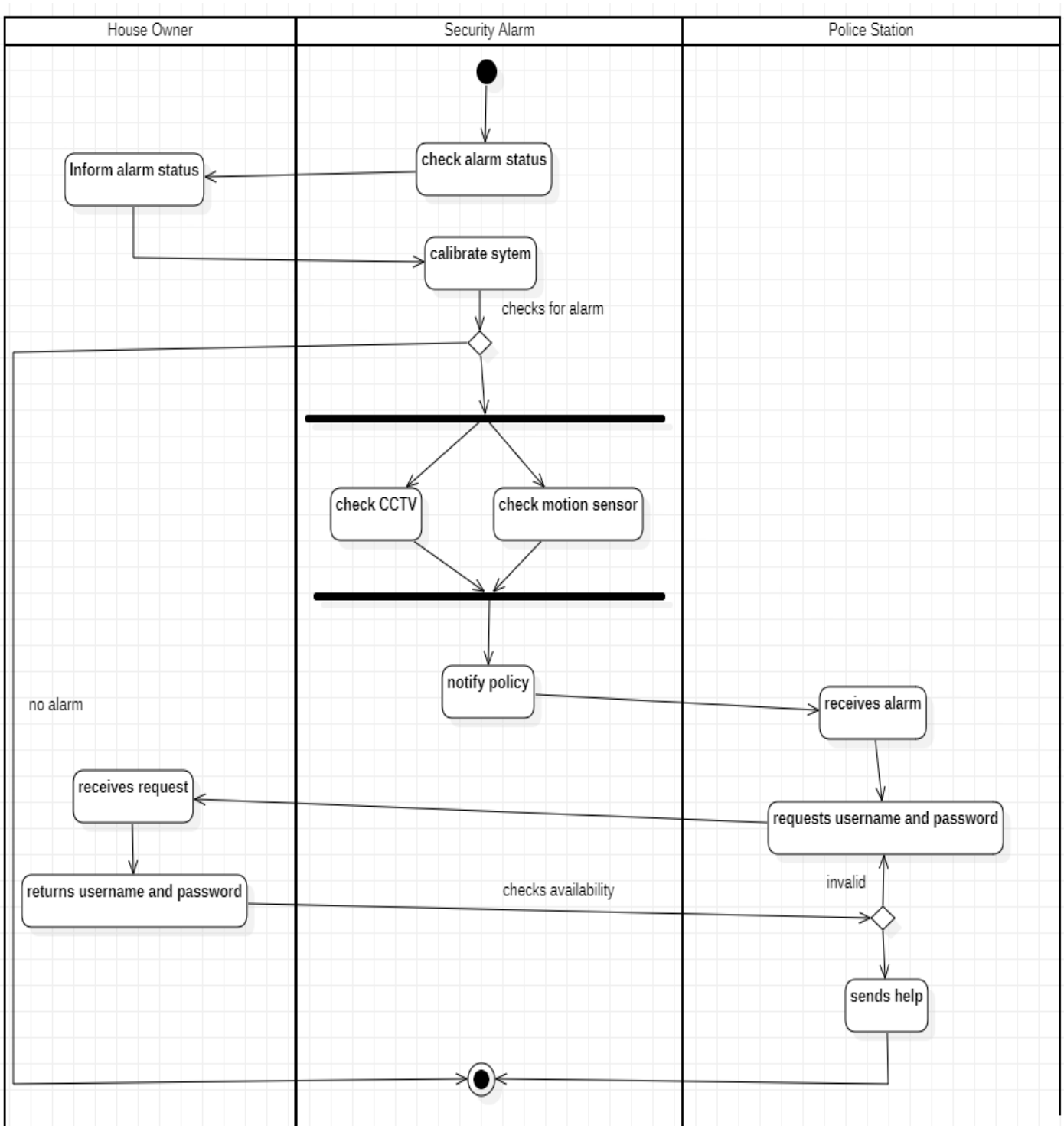
An activity diagram is a flowchart that represents the transition from one activity to another. The activity can be described as the operation of the system. The primary purpose of activity diagrams is to capture the dynamic behaviour of the system. It is also called an object-oriented flowchart. This UML diagram focuses on the execution and flow of system behaviour, not implementation. Activity diagrams consist of actions that include actions that are applied to behavioural modelling technology.

The activities performed by home owner, hub, fire alarm, fire station, security alarm and police department are shown below.




House owner
Fire Alarm System
Fire Station
Inform alarm status
check alarm status
calibrate system
Checks for alarm
Alarm detected
checks smoke level
Low
High
No alarm
Switch on sprinklers
Switch on hose
Notify fire station
Receives alarm
Receives request
Requests username and password
Invalid
Checks validity
Returns username and password
Sends help

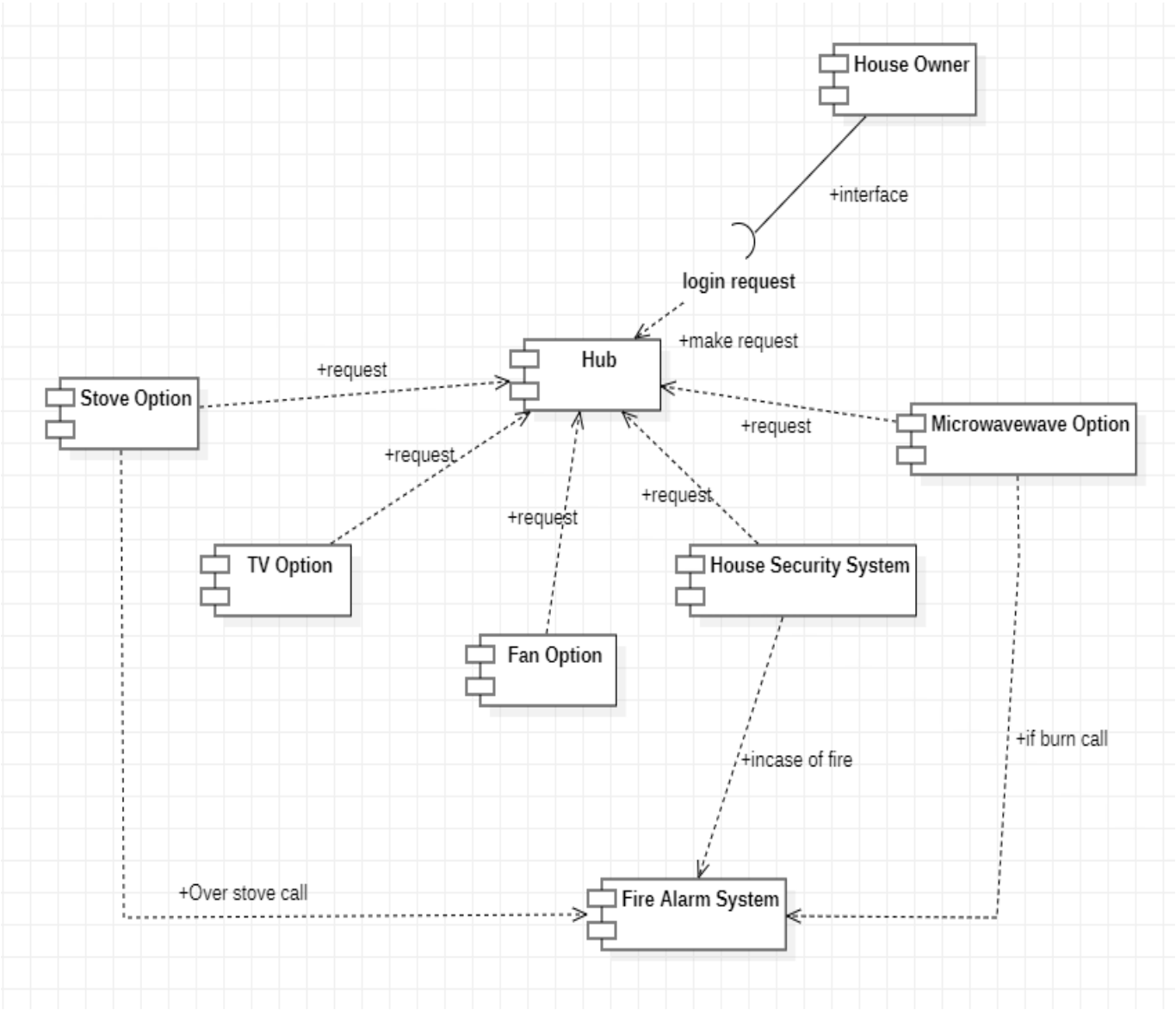

House Owner
Hub
Thermostat
Send instruction
Check status
Returns status
Send request
Receives request
Records request
Checks extremities
Invalid
Valid
Action returned
Reduce or increase temperature


House Owner
Security Alarm
Police Station
check alarm status
Inform alarm status
calibrate sytem
checks for alarm
check CCTV
check motion sensor
notify policy
no alarm
receives alarm
receives request
requests username and password
returns username and password
checks availability
invalid
sends help

COMPONENT DIAGRAM

Component diagrams are used to ensure a consistent transition from the representation level to actual system modules and program code. The graphical elements of this diagram are components, dependencies, and interfaces.

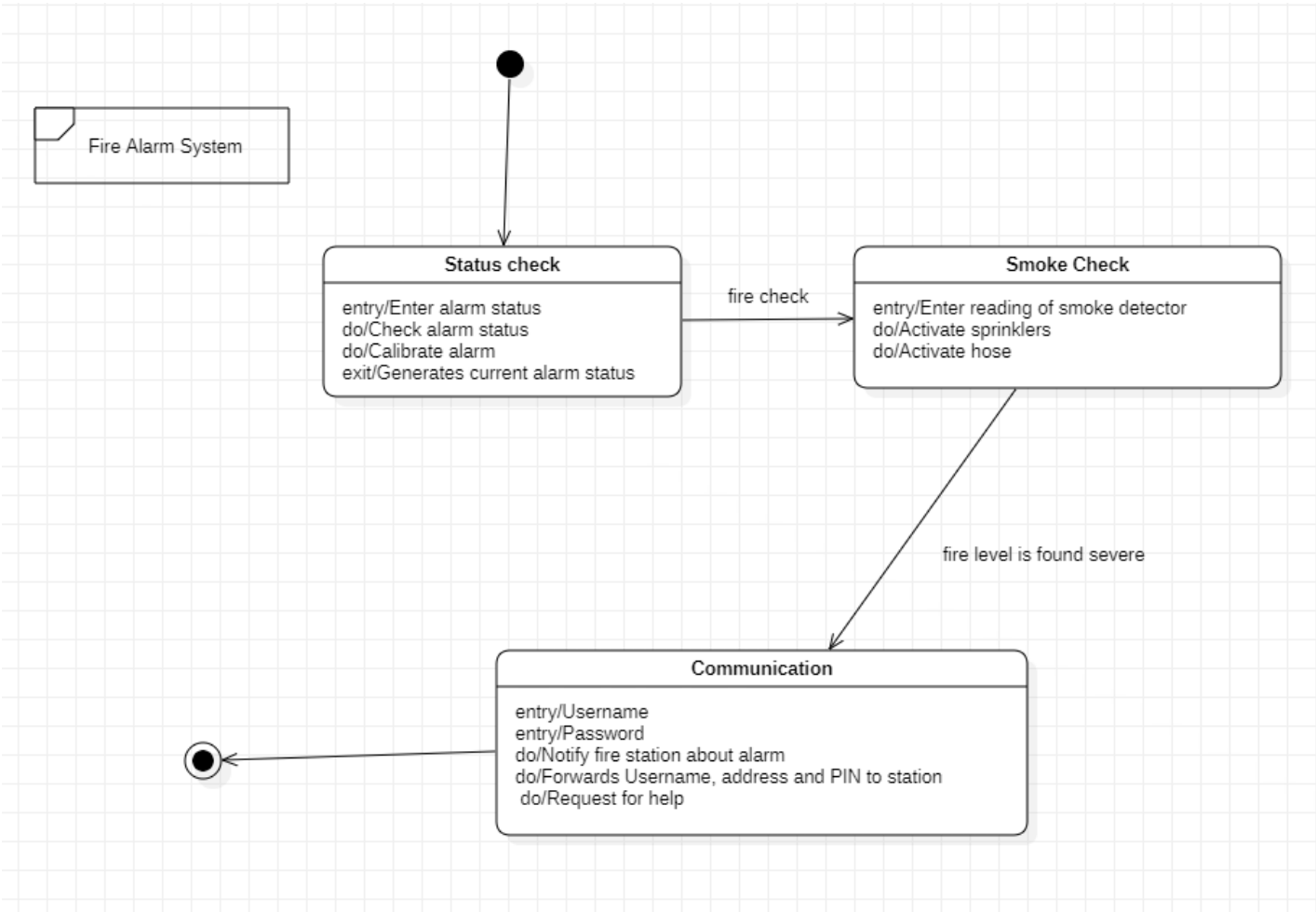
Home owner component gets access to the hub through login request interface and then a request is made to access the appliances associated with the hub. The fire alarm system and security system component waits for the hub's request to get activated.



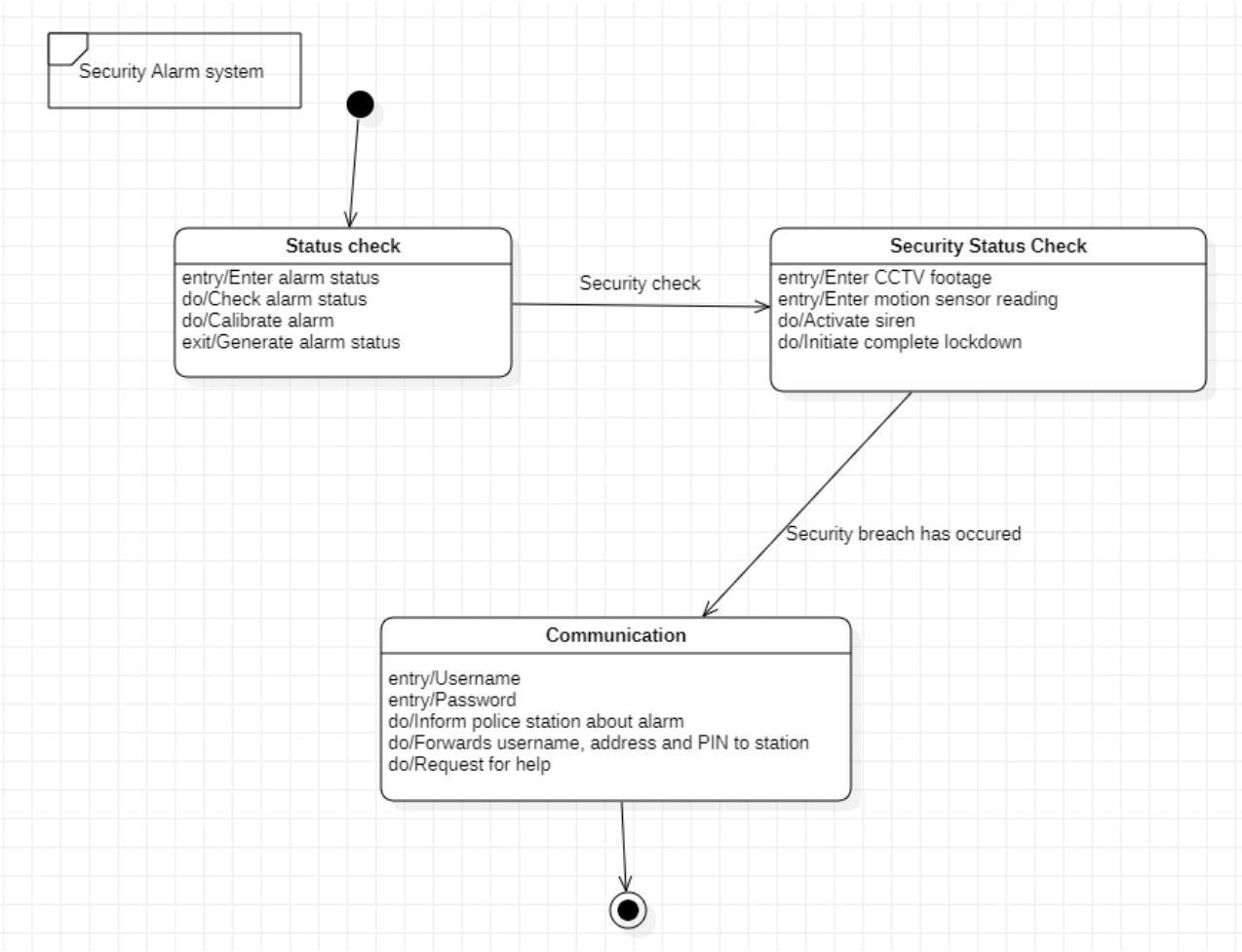
STATECHART DIAGRAM

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination.

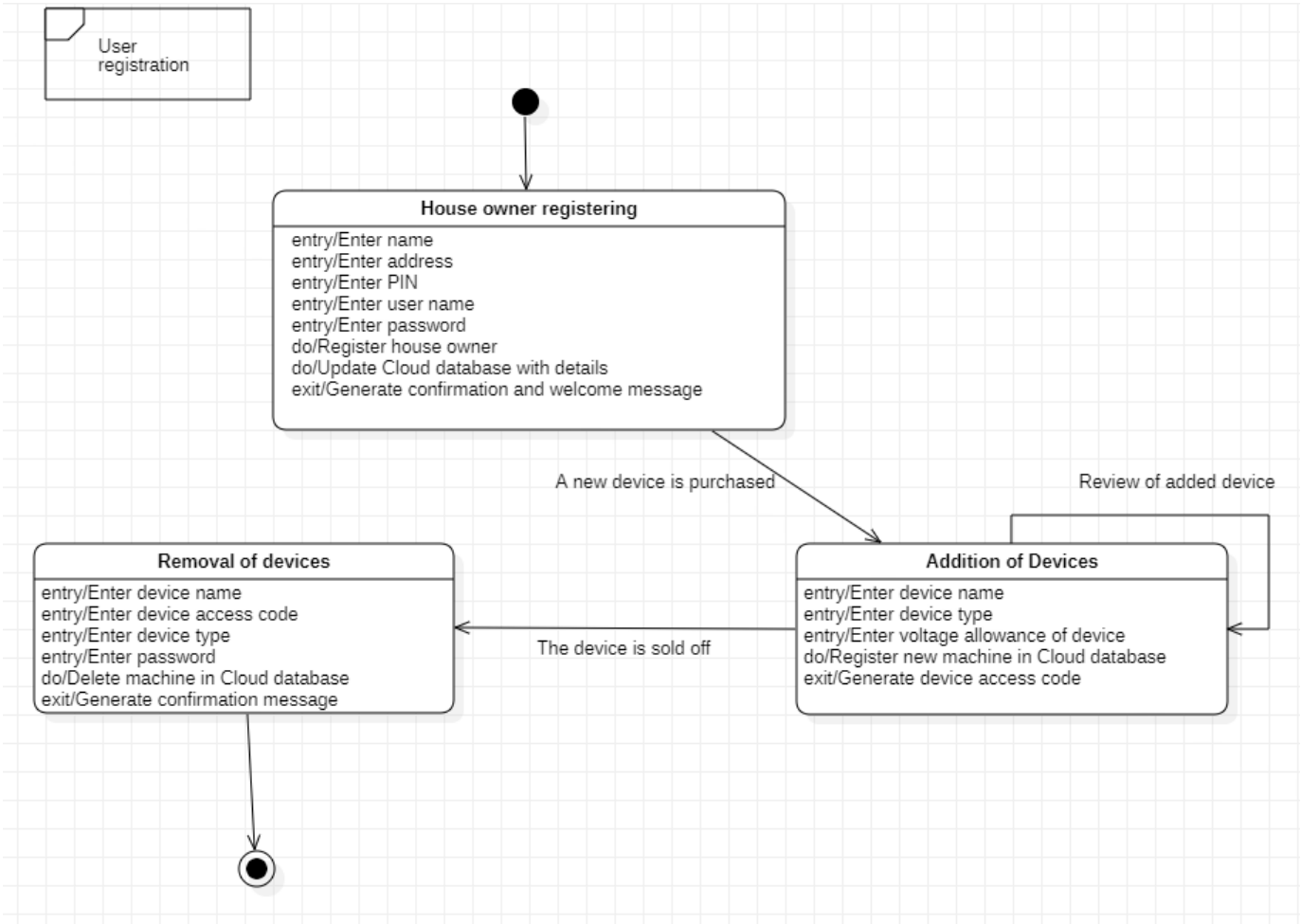
The system has fire alarm system which has status check state that checks with the smoke check state if that found severe then communication state is triggered.



The system has Security alarm system which has status check state that checks with the Security status check state if security breach has occurred then communication state is triggered.



The system has User registration which has Home Owner registering state that is a new device is purchased. The new addition of devices state is triggered and in case of removal of devices the removal of devices state is triggered.



DEPLOYMENT DIAGRAM

In UML, deployment diagrams model the physical architecture of a system. Deployment diagrams show the relationships between the software and hardware components in the system and the physical distribution of the processing.