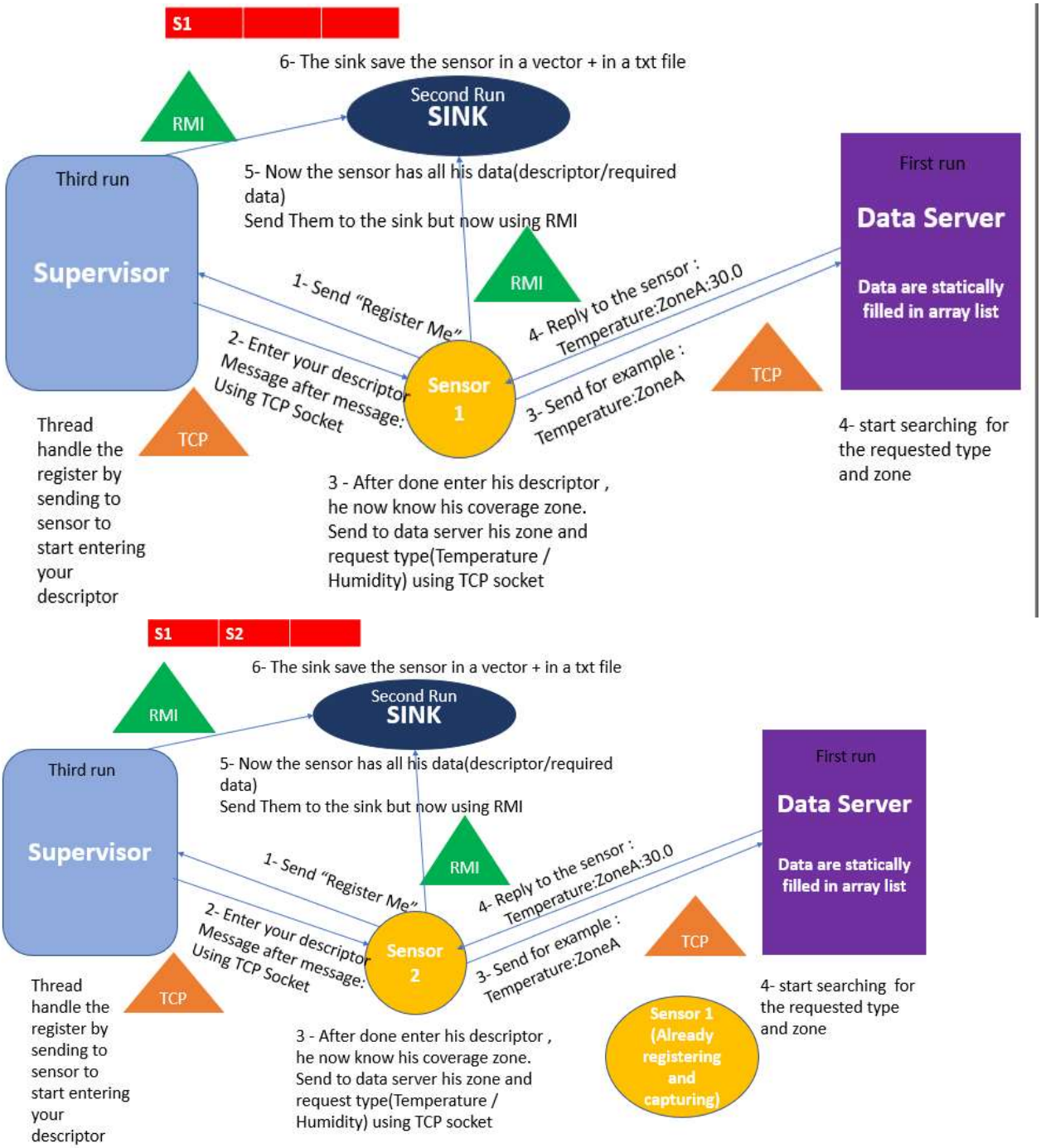
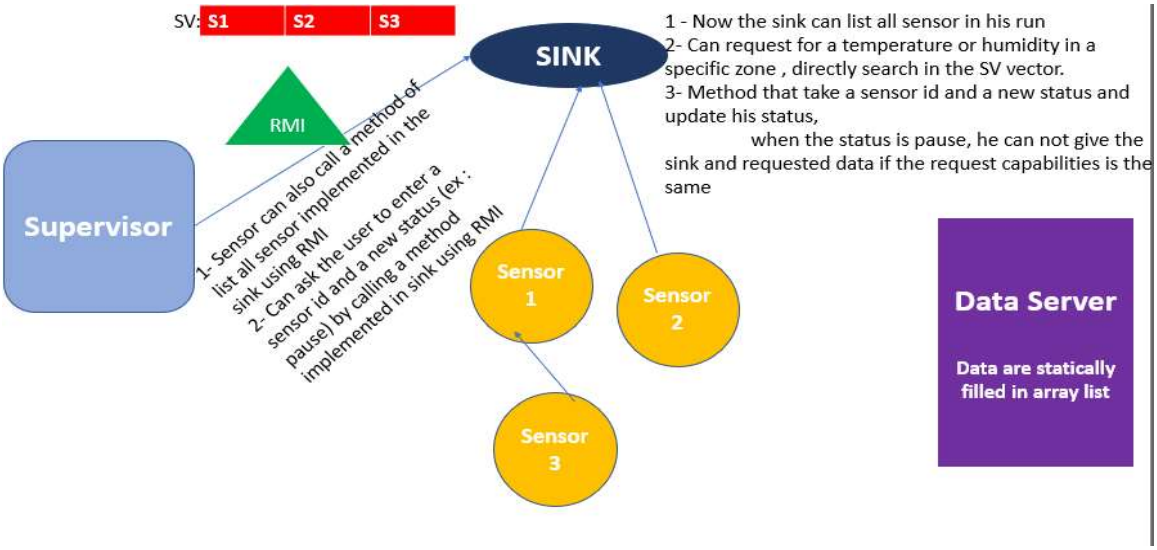
Distributed Applications Programming - Project
Networked media sensors simulator – DEV. Manual





Sink project:

[Sink, Sensor, ItakeSensor ,TakeSensor ,IConfigure , Configure,ThreadCP,Control Panel]

1-Sink (Main class):

- Class containing the main method
- Creating 2 Registrey and rebind 2 object of 2 implmented class.

2-Sensor:

Class Sensor (Serialiazble) , constructor-getters-setters-toString().

3-ItakeSensor : Interface extends Remote contain a method call TakeData()

4-takesensor:

- Class that implements ItakeSensor extends UnicastRemoteObject.

- Method: TakeData(): to receive data from a sensor, and put this string data in the file for track them and put them in vector when I need them , and call the control panel.

5-IConfigure : Interface extends Remote contain 2 method (ListSensors() / UpdateSensor(int id , String status))

6-Configure class : Class implements IConfigure extends UnicastRemoteObject that implements these 2 method

ListSensors : return all sensors and their parametre

UpdateSensor : Update status for a specific sensor(by his id)

7- ThreadCP : a thread that show the control panel when I need it.

7-ControlPanel: (Class for commands)

- Q:to exit the file after emptying them and exit the project
- 0: just write that the control panel will be disable, for preparing for a new sensor to come or for a request from supervisor(by a RMI call).
- 1:to list all available sensors by getting data from the file by the method getsens() (loop the vector of sensor and print all sensors)

- 2:to capture data by putting all sensors in a vector and ask for type and zone ,then he loop in the arraylist of data with conditions on the level(middle, base)and print the final result.

DataServer project:(Multithreading server)

[DataServer, database, HandleRequestThread, Humidity, Temperature]

1-DataServer main: That define a server socket , open a tcp connection between data server and the sensor and a thread is created to do jobs.

2-Database: Class containing the value of data that is available in our project.

(Contains 2 filled arraylist by temperature and humidity values)

3- HandleRequestThread: check the type and the zone and reply with the specific value to the sensor.

4-Humidity class: class with id, zone name and humidity parameters with constructor and getter, setter methods.

5-Temperature class: class with id, zone name and temperature parameters with constructor and getter, setter methods.

Sensor Project:

[Sensor main, Sensor, ItakeSensor]

1-Sensor main: send the message (“register me”) by tcp to the supervisor , then the supervisor accept the connection, and start send to the sensor to fill the id/parentid/requestype(Temperature or Humidity)/coverage zone and the status for now will be set to ‘Registring..’ , these are filled by scanner by the user in the sensor main but on parallel with supervisor(we will se in the code).

2-Sensor class: Class Sensor (Serialiazble) , constructor-getters-setters-toString().

3-ItakeSensor: interface implemented in the sink to take data of sensor.(for the RMI Call)

Supervisor Project:

[Supervisor main, Sensor, IConfigure,ControlPanel, HandleRequestRegister Thread]

1-Supervisor main: open a tcp connection between him and the sensor and running the control panel and a thread handle the connection with the sensor(MultiThreading Server) for ask him to start entring his descriptor(using TCP Socket).

2- Sensor class: Class Sensor (Serialiazble) , constructor-getters-setters-toString().

3- HandleRequestRegister: a thread that receive the message “register me” from a sensor and he send to sensor a message for enter his id , when the sensor finish taking his id , send to the supervisor to send me a new order(enter your parent id) and so on… .

4- IConfigure : Interface implemted in the sink exist here for the RMI call to call the method listsensors() or updatesensor(int id, String status)

5-ControlPanel: (Class for commands)

- Q :to exit the file after emptying them and exit the project
- 0 : just write that and the control panel will be disable, for preparing for a new sensor to come.
- 1 : to List all sensors registred in the sink (RMI)
- 2 : to pause a sensor(update his status to ‘Pause’(RMI))
- 3 :to ready a sensor(update his status to ‘Registred And Ready’(RMI))
- R : to reload the controlpanel if the lookup for the registert is faild (in case of runnig supervisor befor sink(handle by a try catch))