Sensors Task:

# Description

We have different IOT sensors (Temperature Sensor, Humidity Sensor, light sensor) For an environment. Environment is the place where sensors read the current value and alarm if any abnormal case. Environment has the Temperature, Humidity, Light details. Sensors may be active or inactive (inactive sensors will not work, neither read data nor affect abnormality state of the location). The user can configure and interact with the sensors. Also, user can adjust the environment values and set number of locations. Every location in the environment will have one sensor of every type.

# Details:

## Sensors:

We have 3 types of sensors:

* Humidity Sensor reads (Absolute humidity, Relative humidity)
* Light sensor reads (Radiometry and Luminous)
* Temperature sensor reads temperature.

Every sensor must be attached in a certain location.

Every sensor has a normal range (min and max) for every value it reads from environment. These ranges are configured per location by the user.

Every sensor reads the environment values every 1 sec. (for simulation use a generator for the values)

NOTE: All reads are float values

Location:

1. User sets the number of locations in the environment. And give it a name (code)
2. Location will have a state (normal or abnormal) according to these conditions:
   1. if the temperature and light are high ==> abnormal temperature
   2. if the humidity and temperature are high ==> abnormal humidity
   3. if the light out of range ==> abnormal light
   4. any low value ==> abnormal case
   5. all other cases are normal cases
3. The location state changes every 10 sec according to the readings of the sensor.

## User:

The user can:

1. Configure the system
   1. Initiate the environment and sensors’ locations
   2. Initiate the range values of every sensor
   3. Activate/deactivate the sensors

For Sensors:

1. update any Sensor’s normal range.
2. ask for the current sensor reading of the environment value and sensor’s normal range.
3. activate or deactivate the sensor

For environment

1. add location to the environment.
2. remove location from the environment.

GUI:

Use JavaFX to build the user interface.

Screens and its design have been left to your imagination.

At first run of the application, there is no environment configured. So, the user must configure it. Then at the second run the application supposed to load existing environment with its locations and sensors already have their values with the option of adding new location.

# NOTES:

1. Use spring data and hibernate to connect to the database.
2. You can choose the database (mysql, sqlite … etc)
3. In the database, you should store all the environment configuration and the abnormal readings with its messages and time.
4. you should develop a method which generates random number whenever called. This method will be considered as a simulation for the API to be called by the sensor to read from the physical device.
5. Use the Threads and OOP concepts in the implementation.
6. You have to stick to the [SOLID](https://en.wikipedia.org/wiki/SOLID) principles, Clean code and OOP concepts.