Class Quiz - 10

1.

1. **Waste Management** :

   Purpose: Optimization of routes of waste pickup trucks and giving sanitation workers insight into the actual fill level of various disposal units.

   Sensor Types: Proximity Sensors, GPS, Camera

   Algorithms: 1. GetBestRoutes() : displays the routes to follow on map for efficient garbage collection.
   2. GetFillLevel(): Get the garbage fill level of a container

2. **Smart Home :**

   Purpose: Managing home appliances, theft detection using automation of all its embedded technology. It defines a residence that has appliances, lighting, heating, air conditioning, TVs, computers, entertainment systems, big home appliances such as washers/dryers and refrigerators/freezers, security and camera systems capable of communicating with each other and being controlled remotely by a time schedule, phone, mobile or internet. All sensors are connected to a central hub controlled by the user using wall-mounted terminal or mobile unit connected to internet cloud services.

   Sensor Types: Camera, Temprature Sensor, Energy Consumption Sensor, Motion sensors

   Algorithms: 1.Cool_room(): Cools the room depending on the environment.
   2. MoodMusic(): Play a suitable song based on the environment and mood.
   3. TheftDetection(): Detects whether there is any theft happened or in process.

3. **Smart Hospital :**

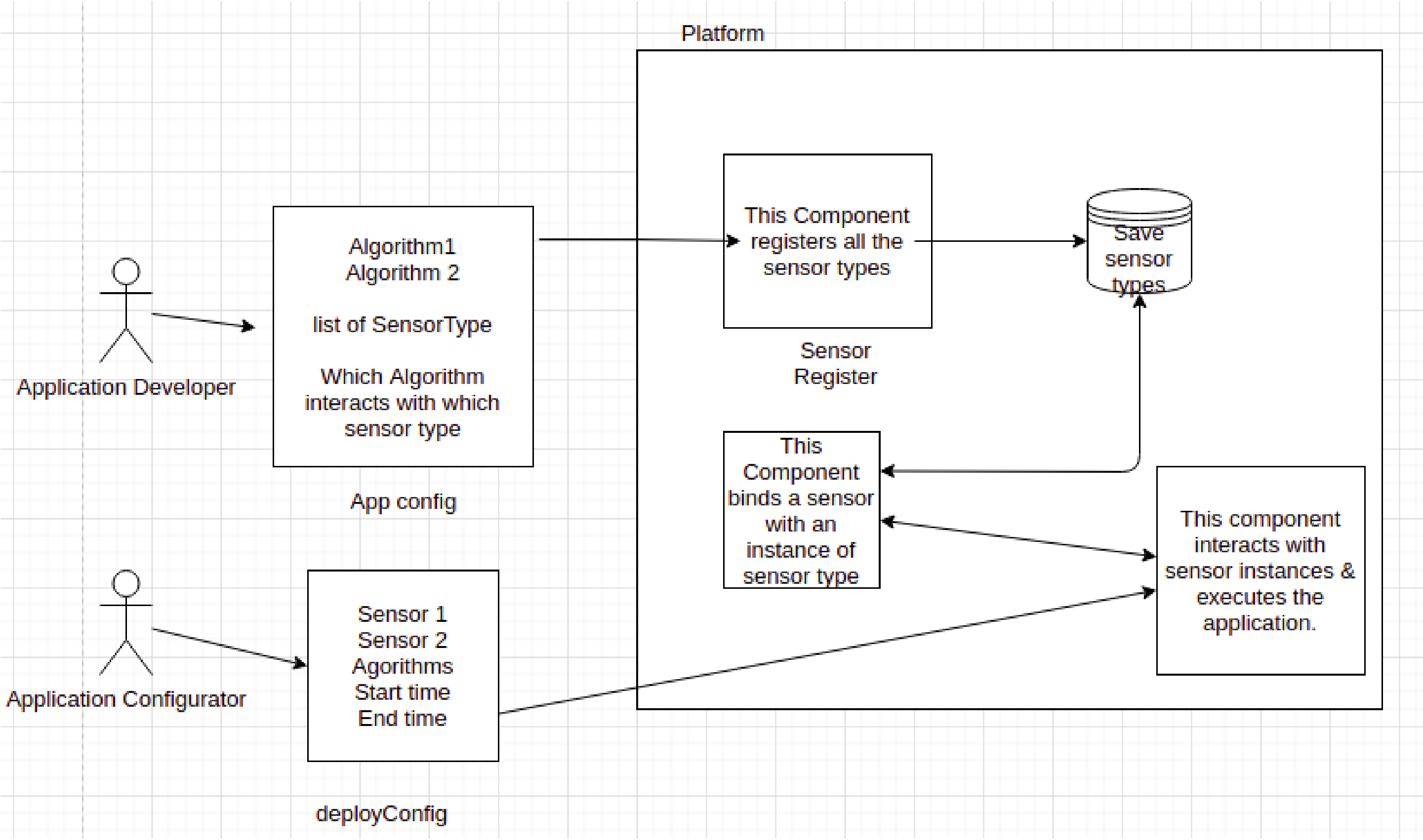
   Purpose: Connecting every patient's details like temprature, heart rate, pressure over a cloud distinguishing each room and wards. Thus to optimize the hospital staffs over the patients.

   Sensor Types: Temprature sensor, pulse sensor, camera, pressure sensor.

   Algorithms: 1. Get_patient_details(): gets a patients basic details like temprature , pressure and pulse rate.

2. Get_criticial(): Get the patient(s) which need some attention in a ward.

2.


Platform
Application Developer
Algorithm1
Algorithm 2
list of SensorType
Which Algorithm interacts with which sensor type
App config
This Component registers all the sensor types
Sensor Register
Save sensor types
This Component binds a sensor with an instance of sensor type
This component interacts with sensor instances & executes the application.
Application Configurator
Sensor 1
Sensor 2
Agorithms
Start time
End time
deployConfig

3.

Keys Elements:

1. AppConfig File: This file contains the algorithms, list of sensor types and which algorithm is interacting with which sensor types. This file is given by the Application developer to the platform for adding the sensor.

2. DeployConfig File: This file contains the sensor types, algorithms and scheduling informations. The purpose of this file is to register the sensor and deploy the algorithms on the platform server.

3. Sensor Register: This component registers the sensors.

4. Platform Manager: This is the interface through which the Application developer and Application configurator interacts. All the tasks and notifications are only performed through this for the applications.

4.

Application Developer:

- Define the sensor type.
- Write the algorithms which will be running on the above sensor type.
- Will provide the above data via a config file. (appConfig)

Application Configurator:

- Register the sensors(which sensor has what type → predefined by app developer) on which algorithms are supposed to run.
- Provide a config file (deployConfig) to the platform containing when to deploy the algorithms and on which sensors.

End User:

- Interacts with the application UI which is provided by application developers.
- Will receive notification from the action server.
- May trigger a particular action on a sensor.

5.

1) The Application developer will register any sensor to our platform with details including:

- sensor type (Above example)
- sensor id
- Algorithms and the respective sensorTypes on which it will run.

2) The algorithms will be saved in appRepo and sensorType will be saved in sensorRepo. Platform will generate an applicationID.

3) Application configurator will create a deployConfig file. Which contains what sensors to be registered and which algorithms are to be deployed with the binding of sensors. This config file will be saved corresponding to the applicationID.

4) Platform read the deployConfig and correspondingly deploy the application.

5) The application can interact with sensor using the action sensor.