SAN JOSÉ STATE UNIVERSITY

CMPE 281

Component Design Document - Deliverable#2

Submitted By:

Lakshmi Kameswari Maduri <lakshmikameswari.nishthala@sjsu.edu>
SJSU ID: 010751715

IOT Smart Street Data Manager

Objective

The main objective of this document is to present the work completed for the component ‘IOT Smart Street Data Manager’. This includes the database design, the API Use cases, UI Template design, class diagram and sequence diagrams to present a comprehensive understanding of the component.

BASIC COMPONENT DESIGN

Functional API Use Cases

The API Use Cases for a IOT Smart Street Data Manager include the following:

1. Add a Sensor
2. Delete a Sensor
3. Read Sensor Data
   a. View Sensor data
   b. Add Sensor reading
   c. Delete sensor reading
4. Check Node connectivity status in the cloud

The use cases can be further explained as follows:

1. Add sensor - Upon receiving request from the farmer, the IOT data manager adds a sensor to the farm
2. Delete sensor - Upon receiving request from the farmer, the IOT data manager deletes sensor from the farm
3. Read sensor data - The IOT data manager reads the sensor data and checks the sensor reading from time to time. He can add/record a new reading or delete a faulty sensor reading
4. The IOT data manager can also check the connectivity status of the nodes in the cloud. He performs health checks for the cluster nodes in the cloud and removes/repairs faulty nodes in the node network on cloud.

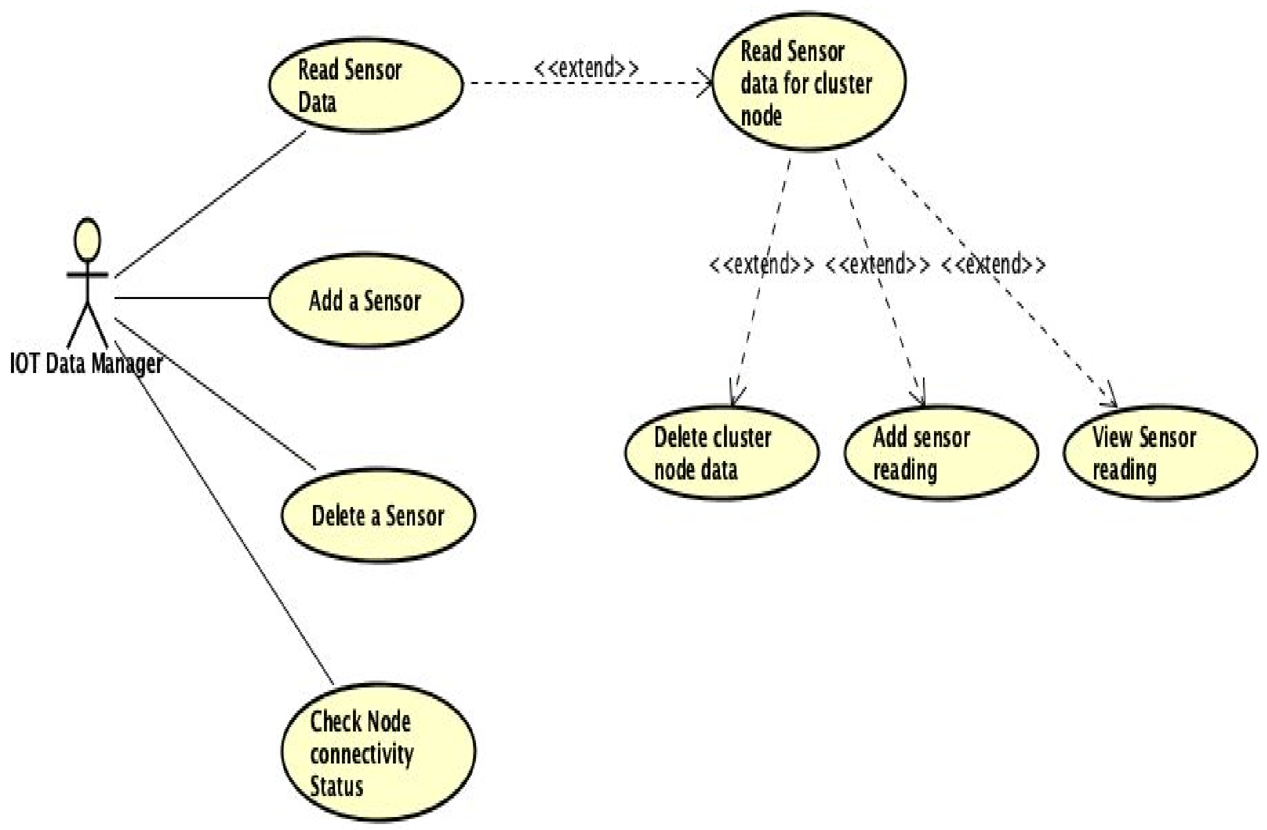


Figure 1: Use Case Diagram for IOT Smart Street Data Manager

Let us now discuss the various classes present in the class diagram for the IOT data manager:

1. Farmers - the farmers are the main users that the IOT data manager deals with. The farmer is the one who requests the IOT data manager to add/delete a node in his farm.
2. Sensor - the IOT data manager needs to keep track of the sensor readings for all the sensors in all the farms for which he is responsible. Depending on the

request from the farmer, the IOT data manager may deploy one of theses types of sensors in the farm:

   a. Humidity sensor - The humidity sensor forms a subclass of the sensor class. The humidity sensor keeps track of the humidity in the atmosphere in the farm.
   b. Moisture sensor - The moisture sensor forms a subclass of the sensor class. The moisture sensor keeps track of the moisture in the soil in the farm.
   c. Soil Nutrition sensor - The soil nutrition sensor forms a subclass of the sensor class. The soil nutrition sensor keeps track of the nutrient content of the soil in the farm.
   d. Temperature sensor - The temperature sensor forms a subclass of the sensor class. The temperature sensor keeps track of the temperature of the farm thereby ensuring that the optimal temperature in the farm is maintained for the growth of crops in the farm.

3. Smart Node - A group of similar sensors form the sensor node. The IOT data manager needs to keep track of the node connectivity status in the cloud
4. Cluster - A group of smart nodes forms a cluster and the IOT data manager needs to keep track of the connectivity of cluster nodes in the cloud

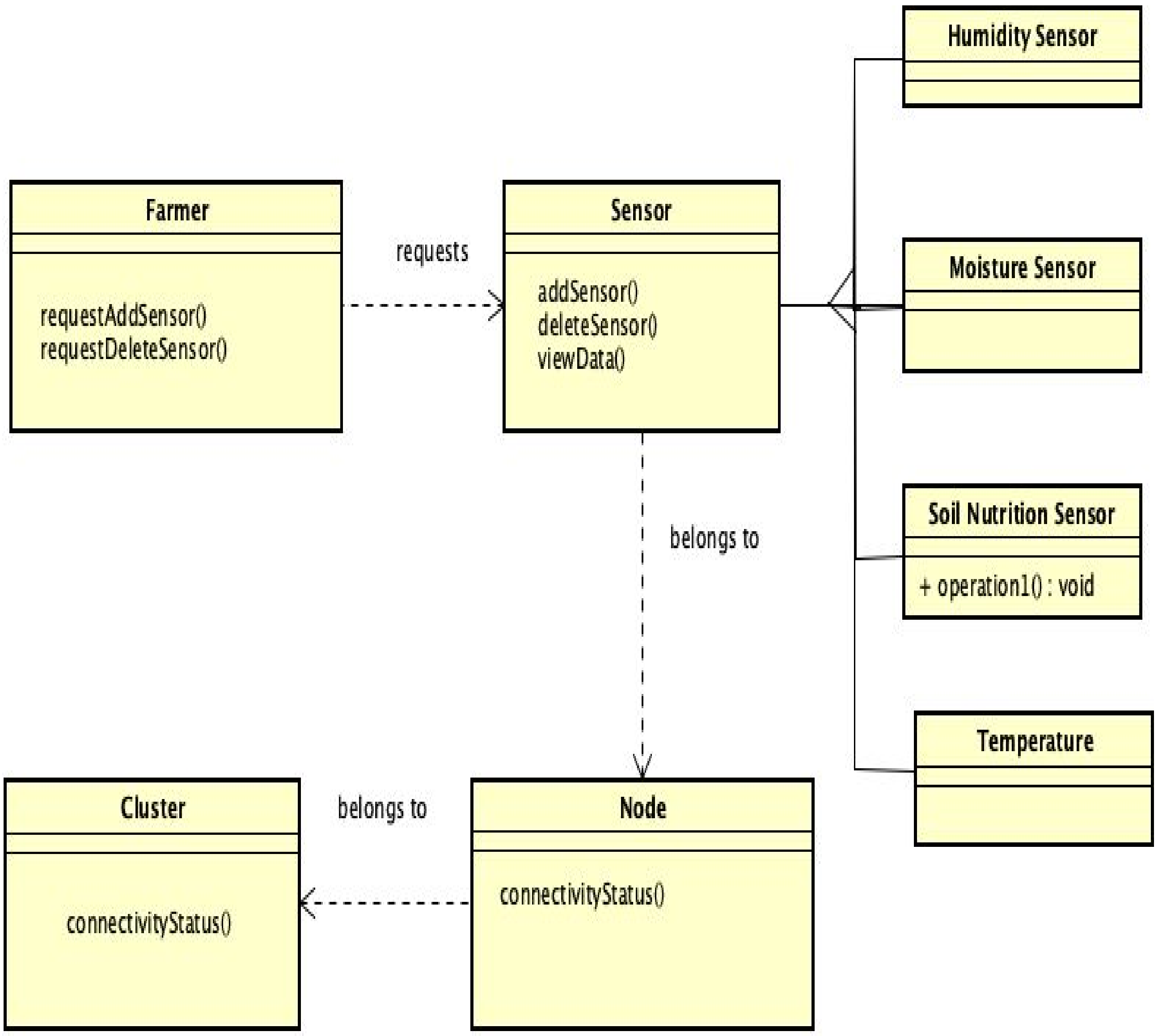


Figure 2: Class Diagram for the IOT Data Manager

The following is the sequence diagram for IOT data manager that depicts the sequence of messages exchanged between the different objects with respect to the IOT Smart Street Data Manager:

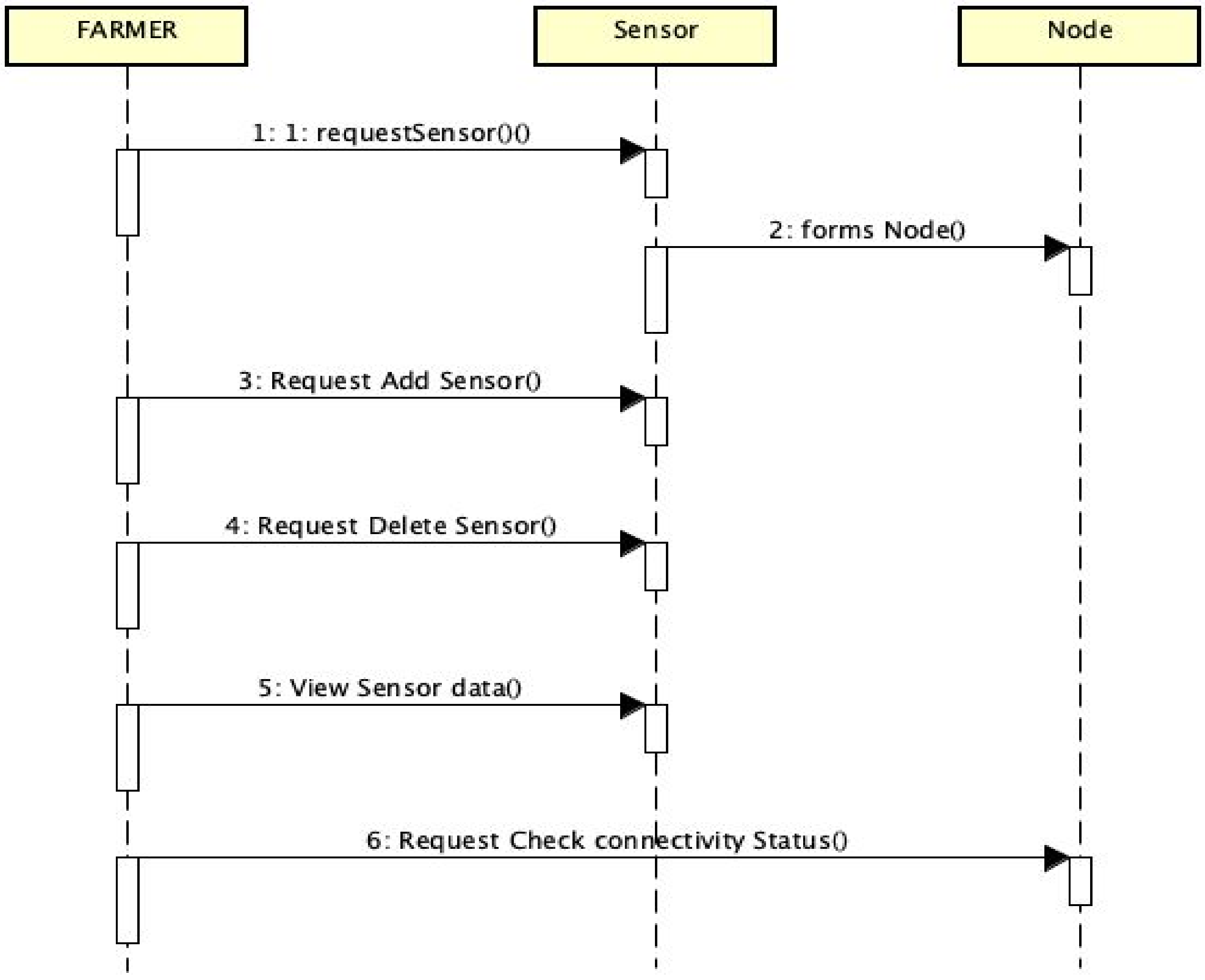


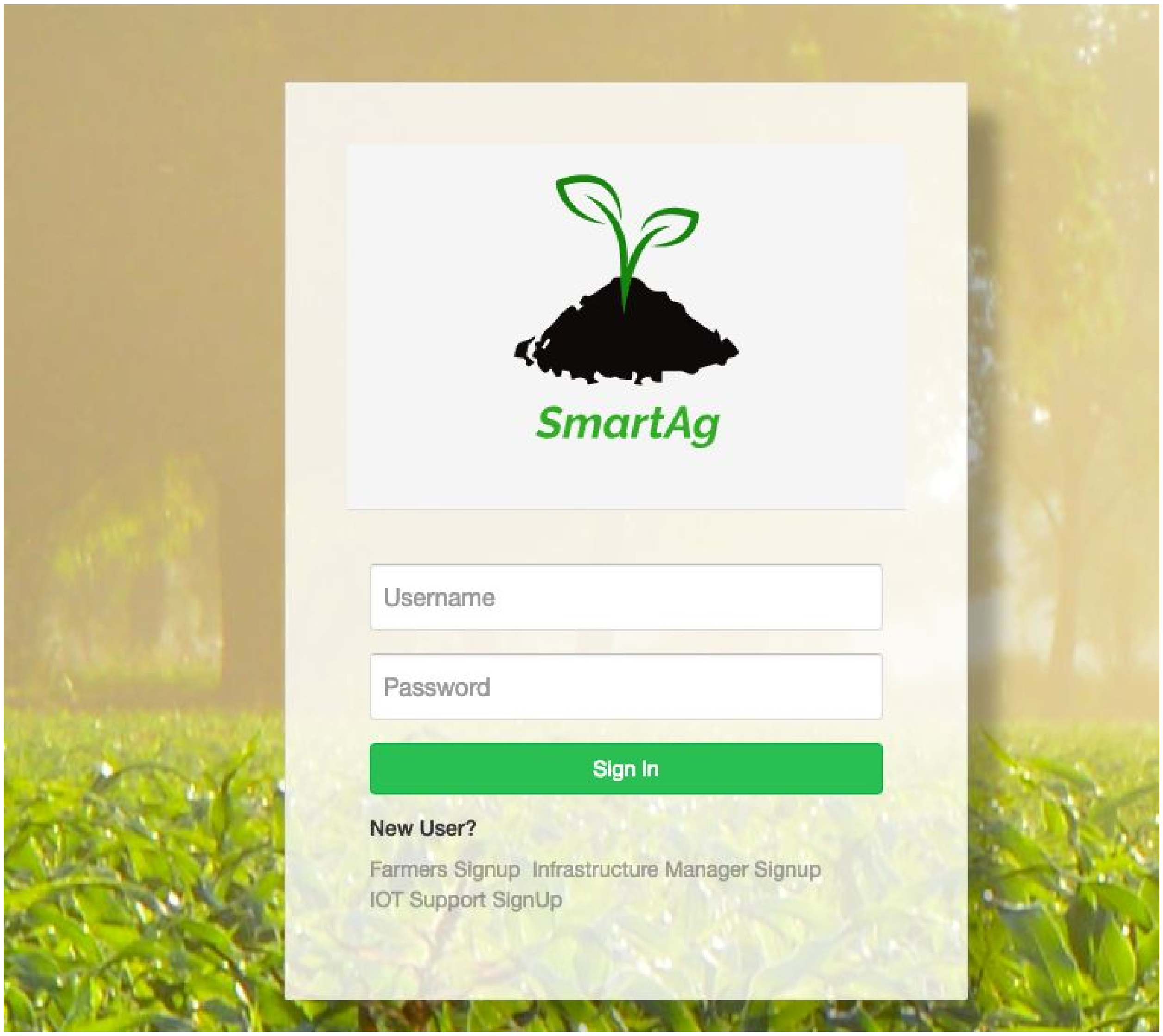
Figure 3: Sequence diagram for the IOT Data Manager

COMPONENT UI DESIGN

The following are the UI templates for the IOT Data Manager:

1. **Login** - The IOT data manager logs into the application using the login page
2. **IOT Dashboard -** This is the main page of the IOT data manager UI where the IOT data manager can see the overall view of the system in the various shown below:
    a. **List Of Framers -** The IOT data manager can view the list of farmers assigned to him. He can view details about the farmers here
    b. **Add a Sensor -** The farmer can request the IOT data manager to add a sensor of a particular type to his farm
    c. **Delete a Sensor -** The farmer can request the IOT data manager to delete a certain type of sensor from his farm
    d. **View Node connectivity status -** This page allows the IOT data manager to manage the nodes the cluster nodes in the cloud. He can manage the health of nodes deployed in the cloud using this page and check whether each node is connected or disconnected.

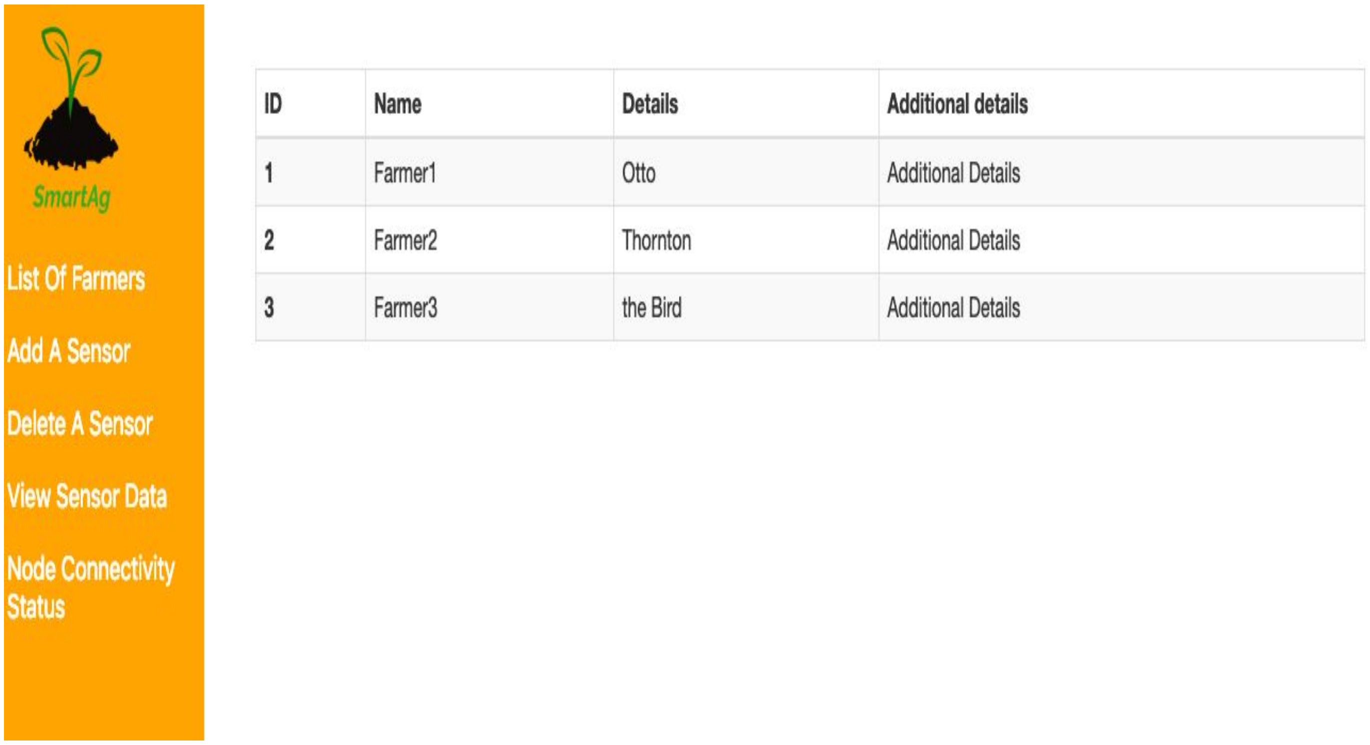
Login


SmartAg
Username
Password
Sign In
New User?
Farmers Signup
Infrastructure Manager Signup
IOT Support SignUp

IOT Dashboard:

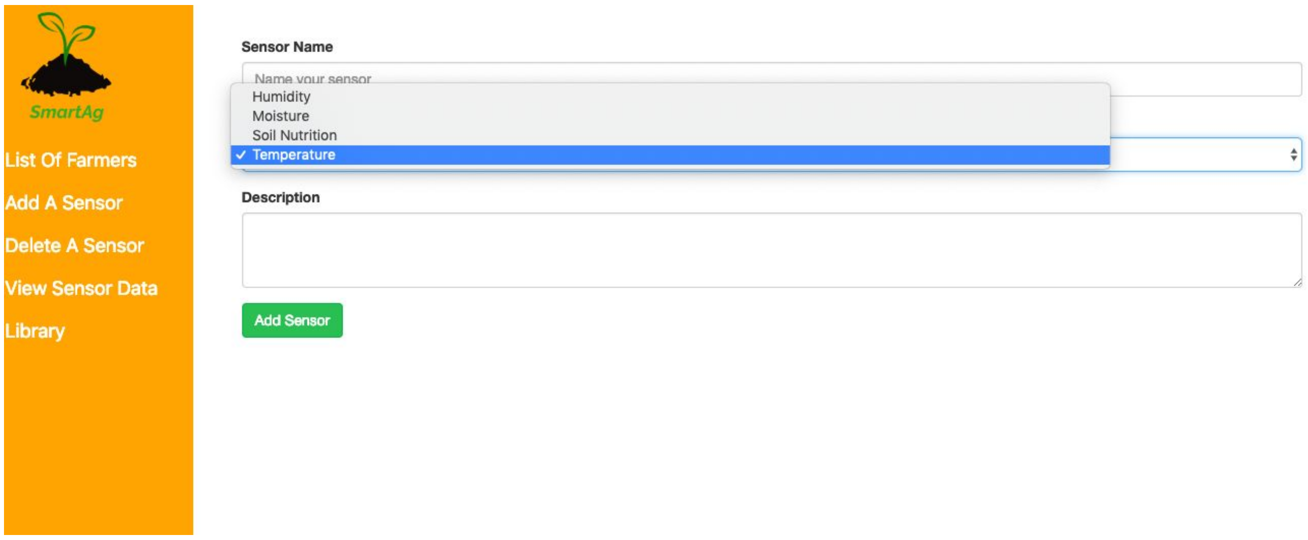
The IOT dashboard consists of the following pages:

List Of Farmers UI:



ID	Name	Details	Additional details
1	Farmer1	Otto	Additional Details
2	Farmer2	Thornton	Additional Details
3	Farmer3	the Bird	Additional Details

Add A Sensor UI**:**



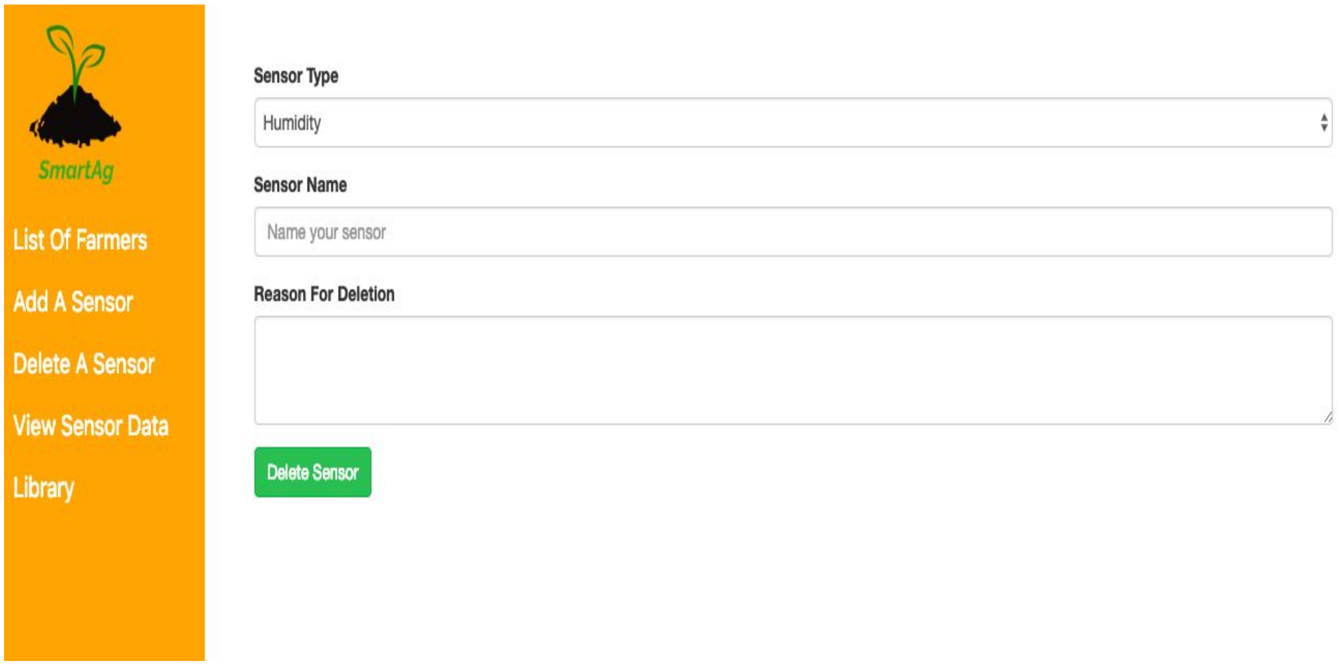
Sensor Name

Name your sensor

✓ Temperature

Description

Delete A Sensor UI:

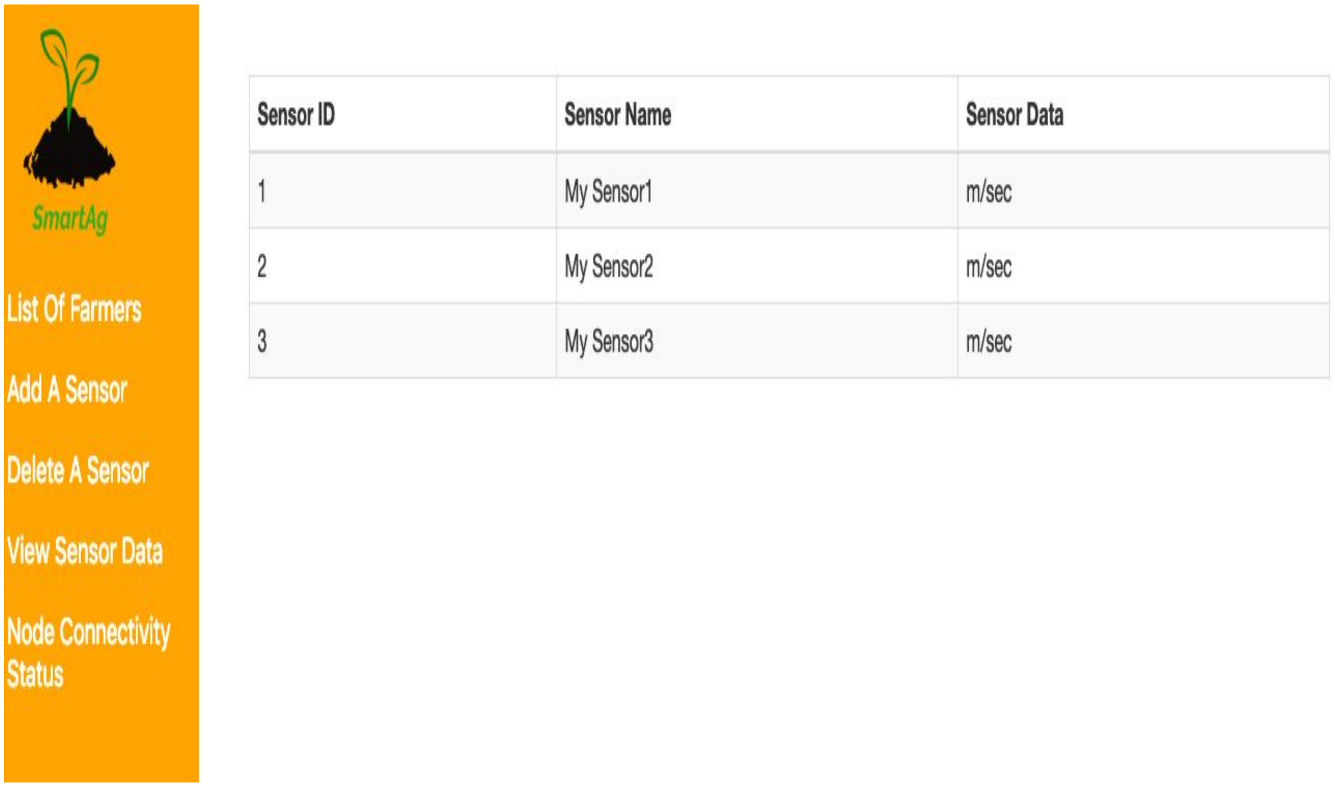


Sensor Type: Humidity

Sensor Name: Name your sensor

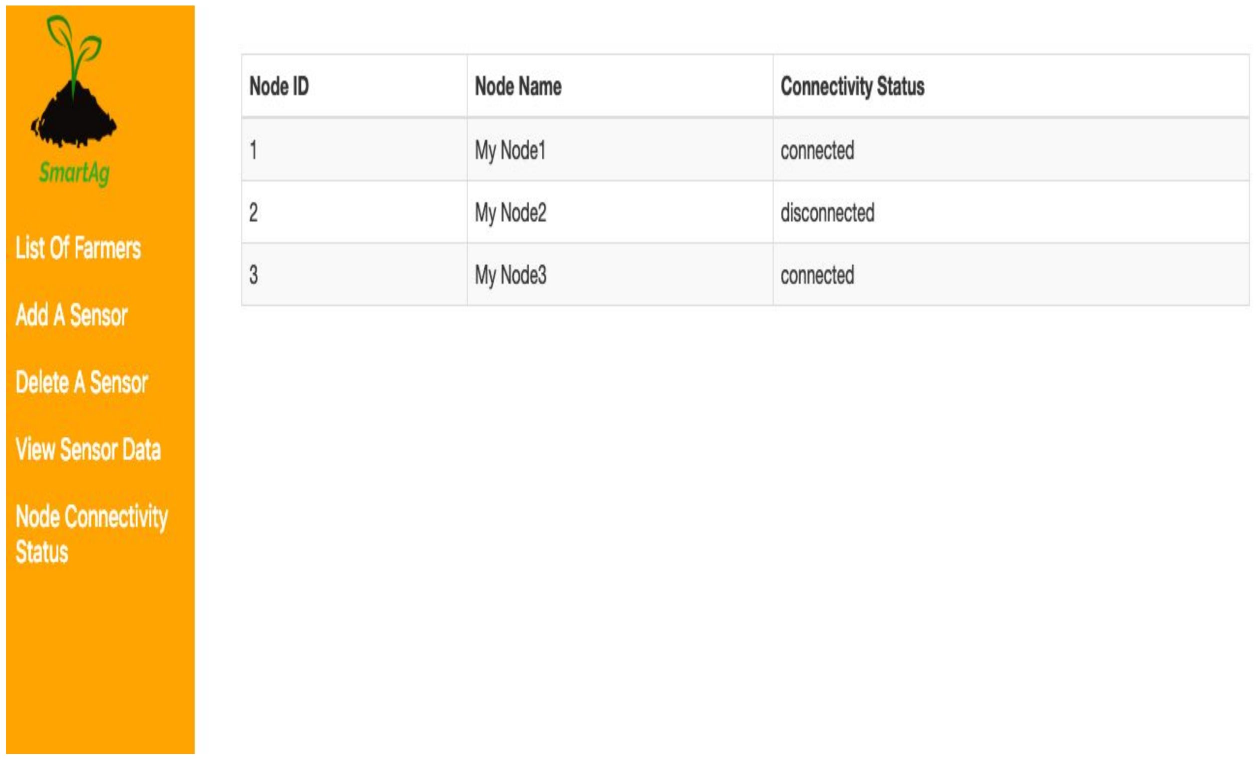
Reason For Deletion:

View Sensor Data UI:



Sensor ID	Sensor Name	Sensor Data
1	My Sensor1	m/sec
2	My Sensor2	m/sec
3	My Sensor3	m/sec

Node Connectivity Status UI:



Node ID	Node Name	Connectivity Status
1	My Node1	connected
2	My Node2	disconnected
3	My Node3	connected

<u>COMPONENT DATA DESIGN</u>

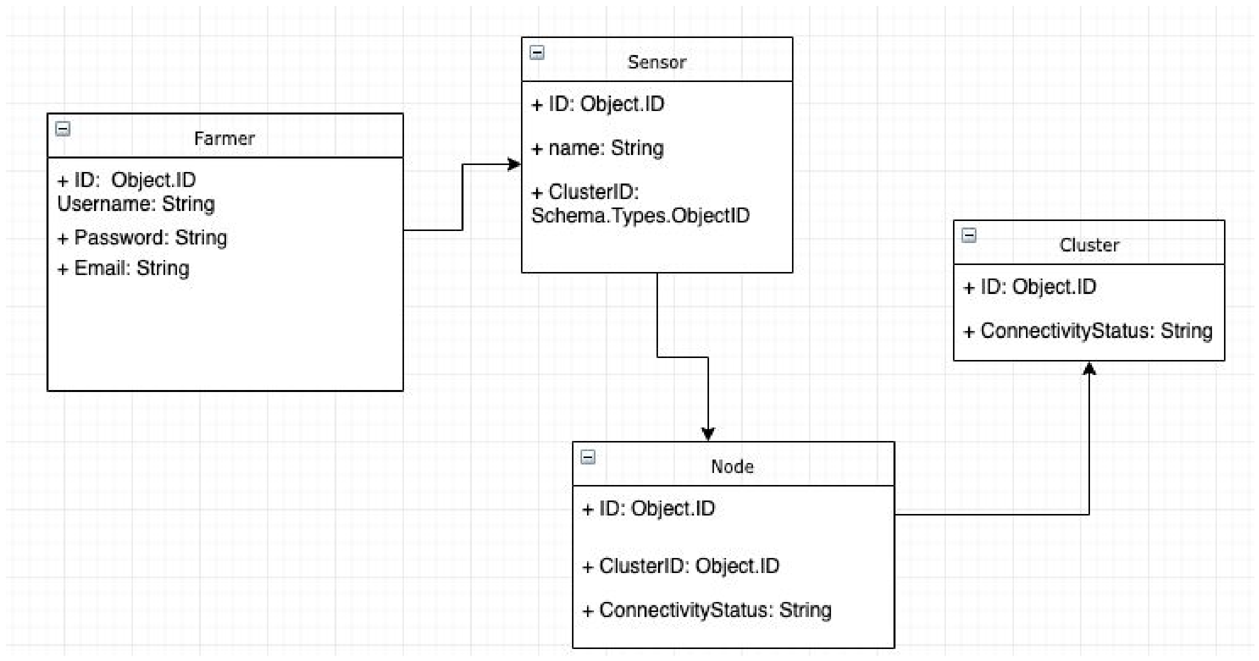
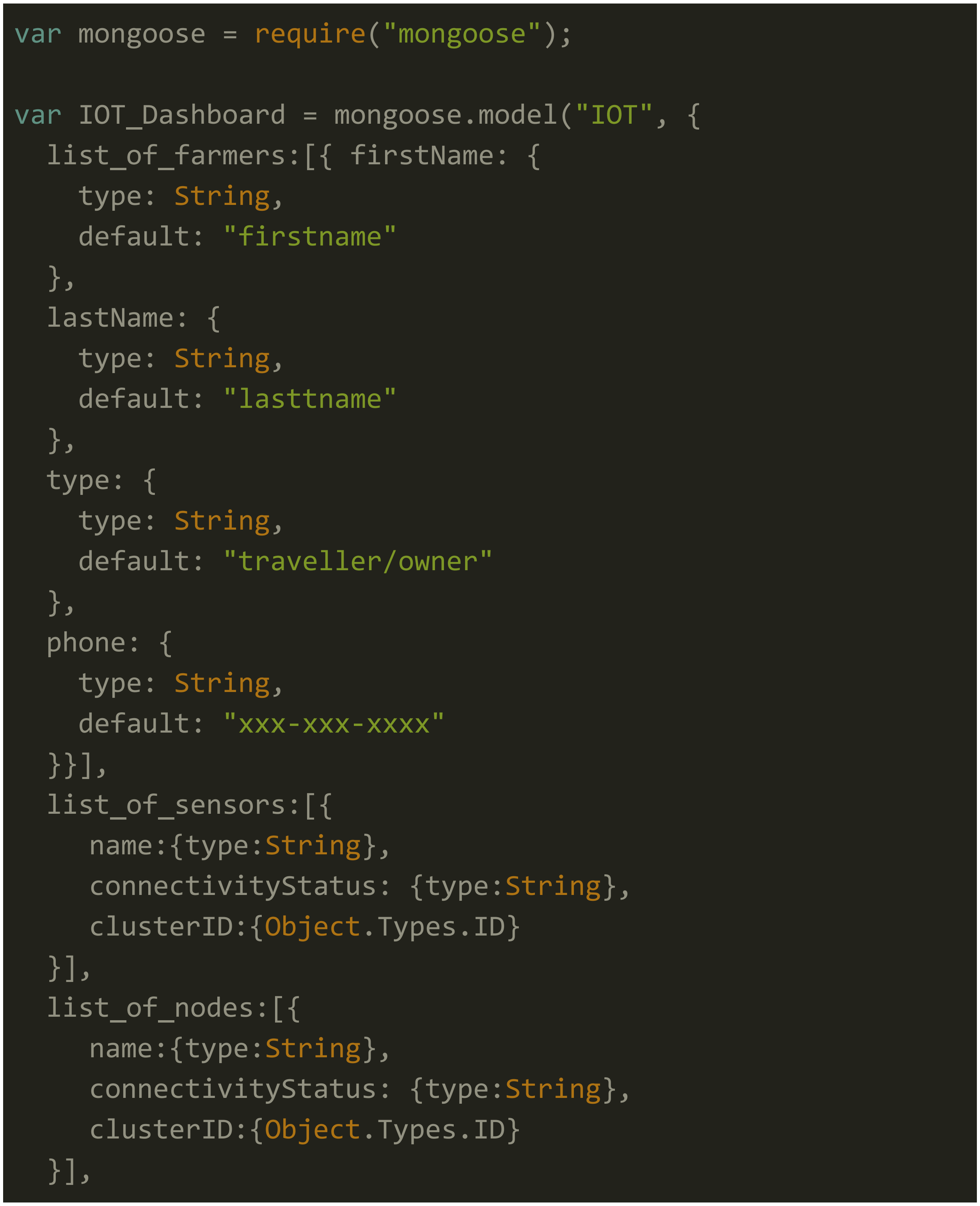


Figure 4: UI Design for the IOT Data Manager

The following is the screenshot showing the actual UI design in the Mongodb schema:



```
var mongoose = require("mongoose");

var IOT_Dashboard = mongoose.model("IOT", {
  list_of_farmers:[{ firstName: {
    type: String,
    default: "firstname"
  },
  lastName: {
    type: String,
    default: "lasttname"
  },
  type: {
    type: String,
    default: "traveller/owner"
  },
  phone: {
    type: String,
    default: "xxx-xxx-xxxx"
  }}],
  list_of_sensors:[{
     name:{type:String},
     connectivityStatus: {type:String},
     clusterID:{Object.Types.ID}
  }],
  list_of_nodes:[{
     name:{type:String},
     connectivityStatus: {type:String},
     clusterID:{Object.Types.ID}
  }],
  list_of_clusters:[
     name: {type: String},
     connectivityStatus:{type:String}
  ]

});

module.exports = { IOT_Dashboard };
```