CMPE 281: CLOUD TECHNOLOGIES

Spring 2016



Evaluation and Project Report

**Submitted To-**

Professor Dr. Jerry Gao

Department of Computer Engineering

**Submitted By-**

Venkatesh Ramanujan

Nikhil Dhamija

Neeraj Chaudhary

Rajiv Pullella

Table of Contents

[Introduction 3](#_Toc437347584)

[Problem Statement 3](#_Toc437347585)

[Document Scope 4](#_Toc437347586)

[Architecture Scope 4](#_Toc437347587)

[**Project Overview** 4](#_Toc437347588)

[Data Collection 4](#_Toc437347589)

[Sata Analytics 4](#_Toc437347590)

[Mobile Sensor Monitor with Dashboard 4](#_Toc437347591)

[Service Billing Module 5](#_Toc437347592)

[Load Balancing 5](#_Toc437347593)

[System Overview](#_Toc437347594) 6

[System Architecture overview](#_Toc437347595) 6

[Workflow](#_Toc437347596) 8

[Technicality](#_Toc437347597) 8

[Request Generation and Monitoring](#_Toc437347598) 8

[Algorithm for Load Balancing](#_Toc437347599) 8

[Service Billing Module](#_Toc437347601) 9

[Technologies](#_Toc437347602) 10

[Node JS](#_Toc437347603) 10

[MongoDB](#_Toc437347604) 11

[Others](#_Toc437347605) 11

[HTML, CSS, JavaScript, jQuery, Bootstrap, PHP 11](#_Toc437347606)

[High Charts 11](#_Toc437347607)

[Case Study](#_Toc437347608) 14

[Conclusion](#_Toc437347617) 14

[References](#_Toc437347618) 14

# Introduction

Cloud computing is the most recent significant advancement in processing. It is a paradigm which has computing resources that are accessible when required, and follows a pay by use method for services. Cloud computing resources are readily accessible at whatever point required and charges depend on the amount of utilization of resources by the user. When the services are turned off, shared cloud resources can be utilized by others when not utilized by that user. The cloud provides a lot of resources like storage, servers and applications. Cloud computing also provides real time monitoring and controlling of data and resources for both the users and providers.

Internet of Things (IoT) and Cloud Computing are rising technologies of the new era. They have become the center of majority of Information Technology industries these days. Both are closely related to each other such that IoT receives powerful computing tools from cloud computing and cloud obtains the best practicing channels using IoT. We are constantly surrounded by web of sensors that are interconnected with each other. But collecting data is just the beginning. We need to process the data collected so that intelligent decision can be made. We require the help of cloud which can help to store, process and analyze the data obtained from the sensors.

The increase in numbers of mobile users had led to the rise of mobile sensor cloud. Mobile cloud computing is data collection from mobile sensor cloud to collect and analyze data. The data processing and data storage occurs in the cloud, outside the mobile devices. The main objectives of mobile sensor cloud computing are:

1. Independence of platform.
2. Independence of device.
3. Multi-tenant.
4. Scalability.
5. Reduction of Maintenance cost.
6. Accessibility.

## Problem Statement

The agricultural industry has always been vulnerable to natural forces which are unpredictable and can never be calculated. The success of any cultivation has just been an option to hope for since hundreds of years. There has been very few or no means to measure the quality of soil or to analyze the soil ingredients and the capacity of soil to cultivate. The agriculture industry also faces challenges to engage and connect to diverse sensor networks. Big data sensing services can help in achieving a centralized and standard level of abstraction with low cost.

The main motivation for the project is to find a simple solution that can help in real time sensor data analysis to determine the probability of a successful batch of cultivation. By applying big data analysis on soil, we are able to determine many factors like pH, silt, temperature etc. which can in turn help to decide the brand of crop and measures to make the soil fertile. By using sensor data as the main base of knowledge we can have an easy and less complex method to provide accurate information to the user.

The aim of this project is to setup an Infrastructure as a Service (IaaS) for Mobile sensor cloud computing. This project involves setting up a mobile sensor cloud infrastructure environment, which supports provisioning, management, monitoring and billing services to support mobile cloud services. This setup must provide an on-demand infrastructure to support multi-tenant user access. Users can directly connect their sensors to our platform, store data and obtain valuable information from it.

## Document Scope

The document starts with the introduction to cloud based soil sensing as a service, the use of a universal platform for devices and the approach to build such platform, implementation and technologies used for development.

## Architecture Scope

The primary objective of the project is to give a base platform that would help to provide smart agriculture utilizing sensor cloud. By utilizing sensors we would be able to do precision agriculture, it not only ensures quality of agricultural products, but also helps in efficient utilization of resources used in agriculture.

The outcome of the project will result in a complete agricultural platform holding sensor cloud to sense the farm data, sensor cloud monitor and controller, sensor information store, sensor information manager and client interface with access to various components and dashboard.

## **Project Overview**

The components in this project can basically be divided into:

1. Data Collection Layer
2. Data Analytic Layer
3. Application Layer

### Data Collection Layer

Data collection layer is responsible for collection and transferring the data. The bottom most part of the layer are sensors which are the source for the data. The sensors are under the control of IoT node which is responsible for gathering the data. No filtering of data is done at this stage and the data is transferred as such to the IoT Hub. IoT Hub filters the data and sends them to the cloud storage. A given IoT hub controls ‘N’ number of IoT nodes in a given region. The number of IoT hubs required should be decided based on profiling of the farm lands.

### Data Analytic Layer

Data analytic layer is responsible for storing and analyzing the sensor data. It also acts as the interface between the Data collection layer and the user interface layer, thus abstracting the end user from the direct interaction with the sensors. Stream analytics acts as the pipe line that connects the event hub and the Mongo DB. Once the data reaches Mongo DB, the DB analyses the new dataset and makes decision regarding what needs to be done. For example, if the logic apps analyzes that the soil moisture is low, the decision will be to start the irrigation motor. The decision taken is then sent to the IoT hub in the data collection layer via event hub. Based on the event received the IoT hub performs the required action.

The data stored can also be retrieved based on the query generated due to the end user operation. This query is handled by the query engine (ADO .net based). Also the user can control, configure and monitor the end sensors using the web apps. Web apps sends the user requests regarding the sensor cloud in the form of events to the IoT hub via event hub. Based on the events received, the IoT controller performs the necessary action.

### Application layer

Application layer is the one which the user sees. It is provided with the dash board which updates the user regarding the latest proceedings as well as the result of the user initiated actions. For example, Dash board updates information like successful addition of the sensors, Bad sensor that has stopped sensing the data.

**Monitor section** is used to view the data that is being sensed. This section displays the latest datasets in the graphical form for the view of the user.

**Controller section** is used to control the sensor. For example, using the controller UI the end user can change the frequency with which data should be sent from the sensor and format of the data that is being extracted from the sensor.

**Configure section** allows the end user to configure the parameters that are needed for the data analysis such as threshold for the soil moisture.

**Onboarding section** allows the end user to install new sensors at a new location.

All the transaction from the UI reaches the end data collection layer as events through Event hubs.

## Service Billing Module

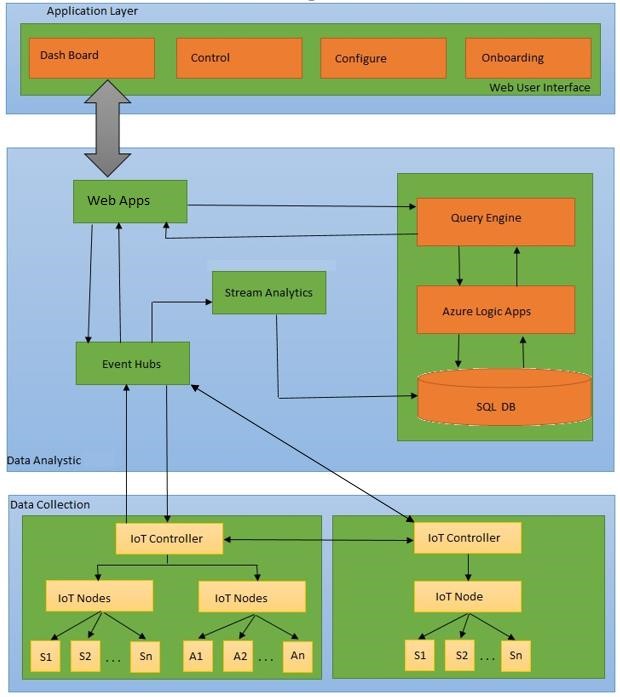
This component devices a business cost model that is implemented with billing metrics to calculate the costs for each user based on their resource request. The total service cost is calculated and generated as bill to the user.

## Load Balancing

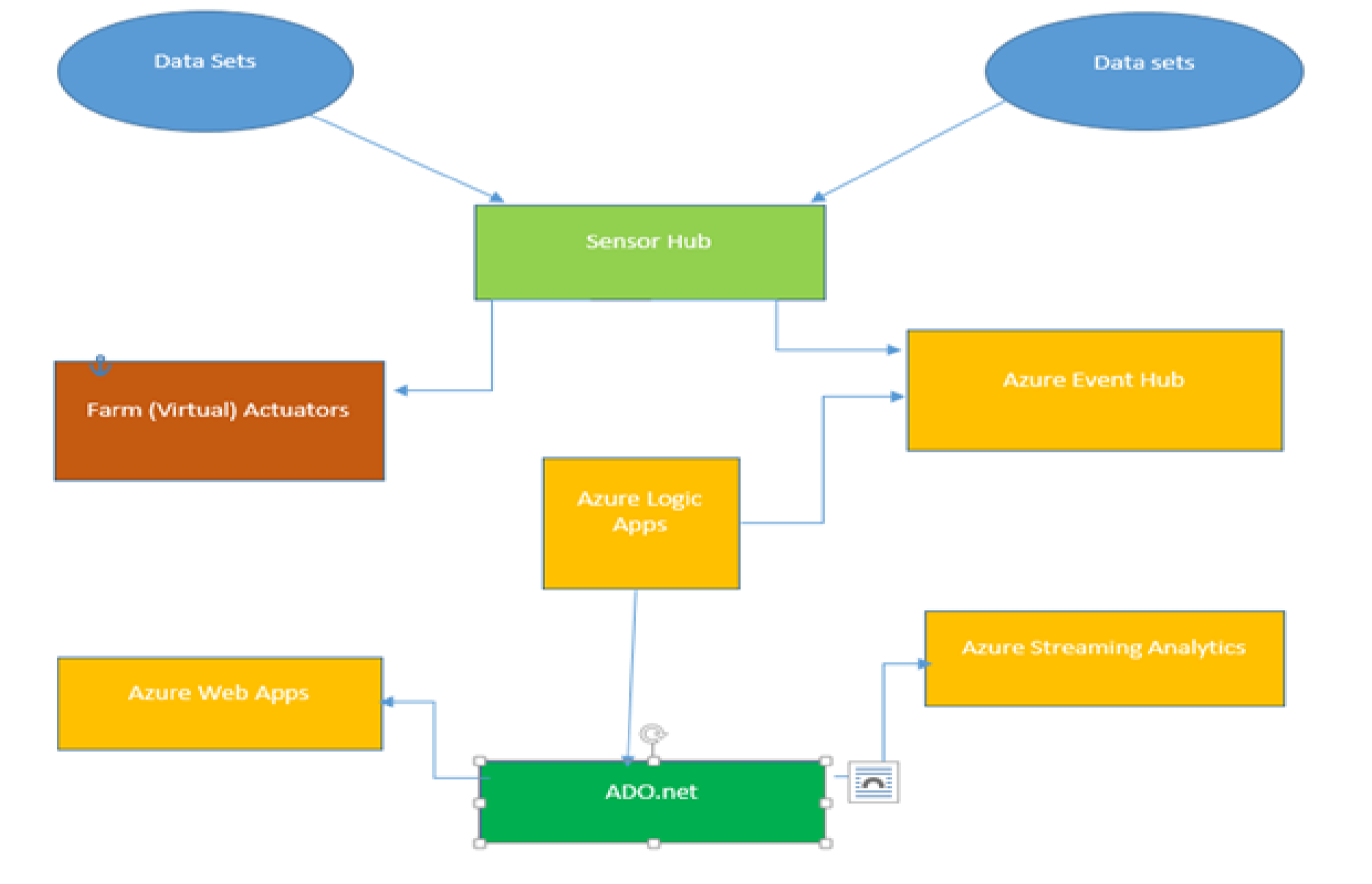
## System Overview

## System Architecture overview

The below illustration demonstrates a typical scenario of how an incoming request is processed. For multiple requests, the resource provisioning and management engine employs a load balancer algorithm to distribute the workload among multiple hosts spread across 2 different clouds.



*Figure 1: High Level System Architecture*



*Figure 2: High Level System Deployment diagram*

|  |  |
| --- | --- |
| **Service** | **Description** |
| Sensor Controller | This service is used by the end user to communicate with the sensors. |
| Agro Environmental Automation | Based on the retrieved datasets such as soil moisture and soil temperature agriculture activities such as switching on the irrigation motor, closing the rain shields are automated. |
| Sensor Cloud Monitor | This service can be used by the user to monitor the live status on the sensors located in a particular grid. |
| Sensor Configurator | This service by the user to configure the sensor. For example, the user can set the frequency with which the data should be sensed and what should be the format of the data sensed. |
| Sensor Onboarding | Sensor onboarding service can be used by the user for the addition of a new sensor into the existing sensor cloud. |
| Sensor Data Visualization | The data from the sensor over a period of time is presented in graphical form to the user. |
| Dashboard | Dash board contains the news about recent  activities that took place in the sensor cloud |

## Workflow

The workflow of the application can be derived from the above mentioned Architecture Diagram, we can understand the system flow as follows:

1. The user is logged in to his dashboard after he enters his credentials.
2. The can add or delete a sensor.
3. The user request is sent using GET through one of the node servers.
4. The server then adds or deletes the sensor for the user based on his request.
5. The user dashboard starts to display data of the newly added sensor.
6. The user can check all the subscribed sensor values in the dashboard via charts and graphs.

There are several interactions between the Internal APIs and External APIs. The HTTP GET method adds or deletes a sensor on the user’s request. The sensor data collected is in JSON format. This data is parsed and the respective values are stored in the database. All the sensor data is stored in the database and used for graphical representation. The entire log of sensor values is displayed separately in a table format.

# Technicality

## Request Generation and Monitoring

The application developed uses RESTful web services to interact with the system. They are listed below:

1. Add Sensor: It is a GET method to create a sensor in a specified location.
2. Delete Sensor: It is a GET method to delete a particular sensor. The user needs to enter sensor ID of the sensor and the sensor is deleted.
3. Enable Sensor: It is a GET method used to enable a sensor once the Sensor is added to the user.
4. Disable Sensor: It is a GET method that disables an already existing sensor.

## Developing Stream Analytics

Stream analytics is used to bridge the event hub to the data store. Stream analytics can be configured using the console. One end of the stream in connected to the output of the event hub and other end is connected to the Mongo DB.

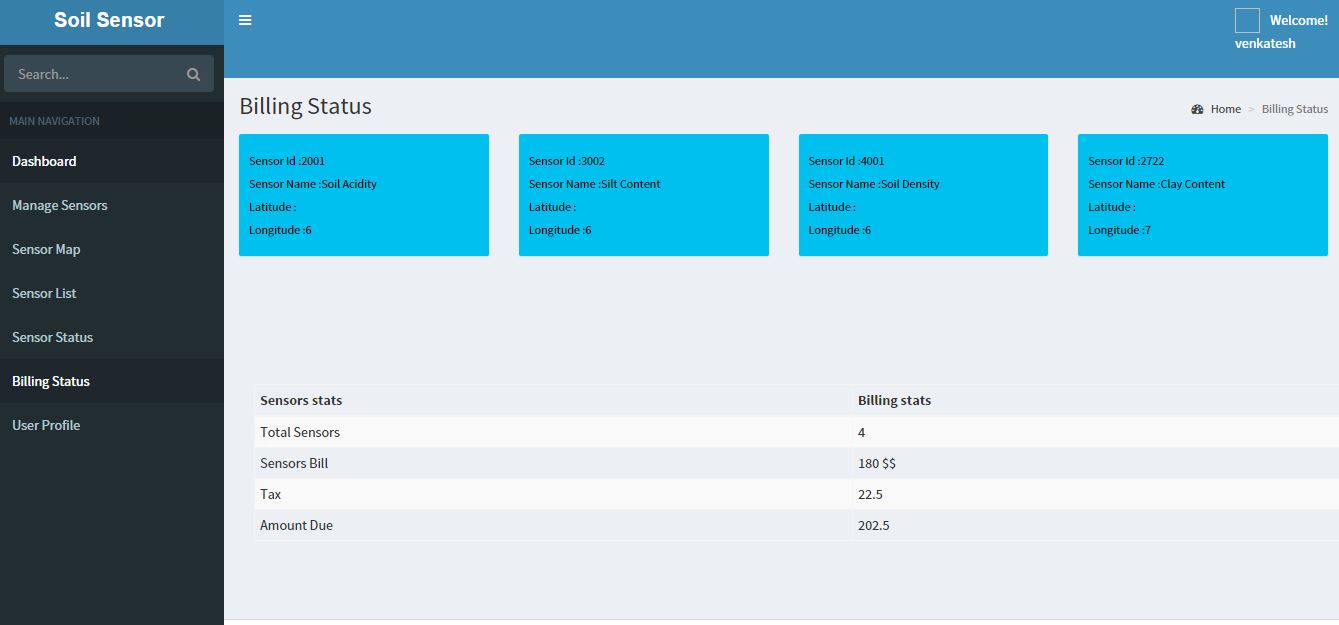
## Algorithm for Load Balancing

## Service Billing Module

The Billing model of cloud is *Pay-as-you-use* Model. Billing does not consider time as a factor as the data received from one sensor can be so huge that it can overload the server within minutes while some may send very little.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Customer | OS | RAM | CPU | No of Sensors | Sensor frequency | Data sent | Data Region | Price |
| Venkatesh | Windows | 8GB | 4 | 4 | 5 sec | 500 | Belgium | 202.5$ |

Request tracking per user:



# Technologies

## Node JS

**Node.js** is an open-source, cross-platform runtime environment for developing server-side Web applications. Although Node.js is not a JavaScript framework,many of its basic modules are written in JavaScript, and developers can write new modules in JavaScript. The runtime environment interprets JavaScript using Google's V8 JavaScript engine.

## MongoDB

MongoDB is a free and open-source cross-platform document-oriented database. Classified as a NoSQL database, MongoDB avoids the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster.

## Others

## HTML, CSS, JavaScript, jQuery, Bootstrap

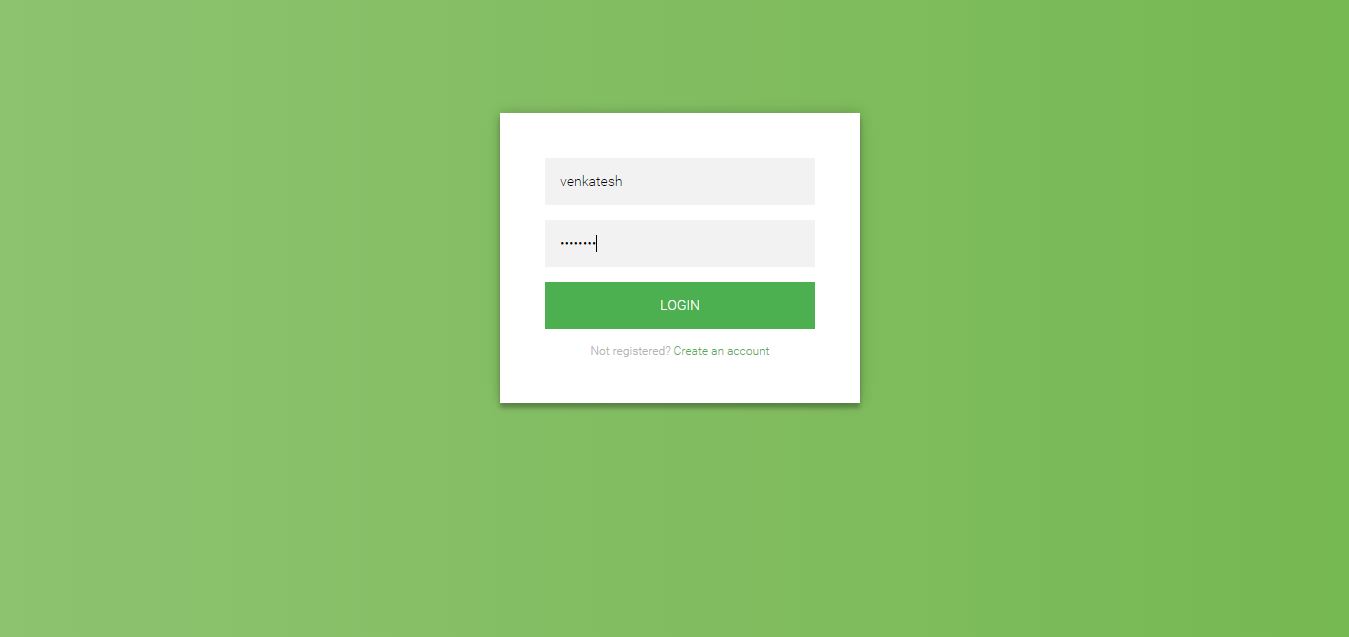
HTML5, CSS and Bootstrap were used to develop the front-end UI.

### High Charts

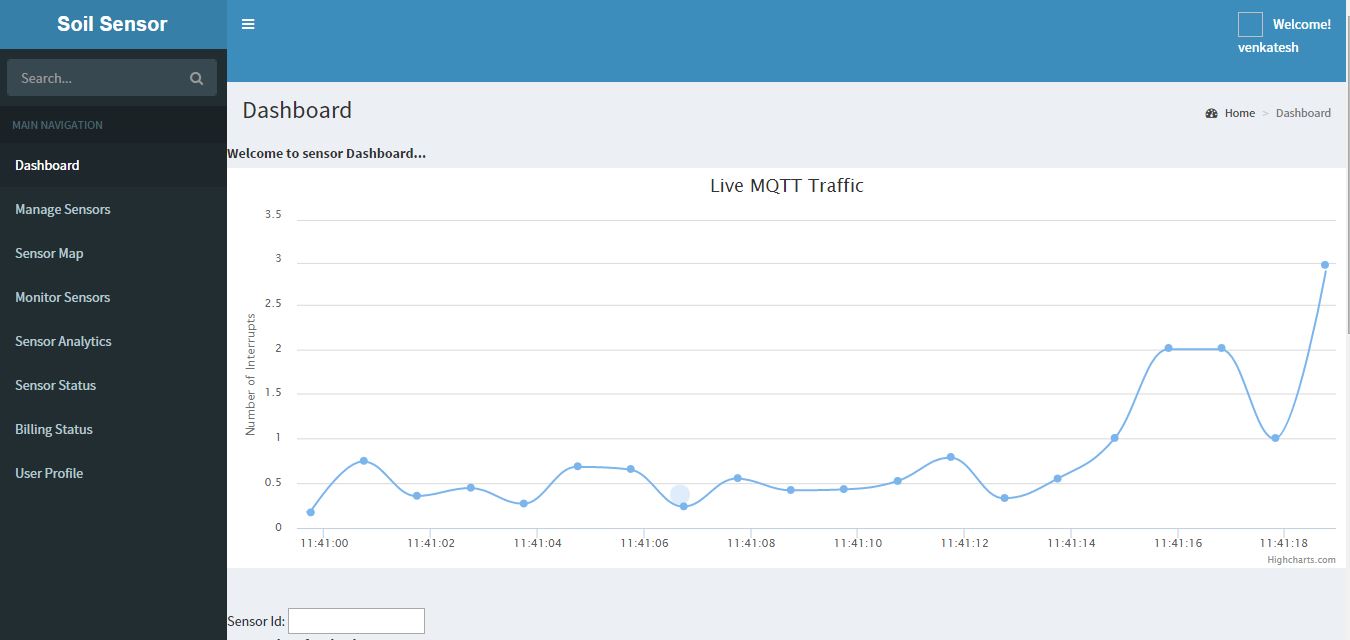
High charts were used for the purpose of dynamic updating of the graphs and displaying charts for cloud statistics.

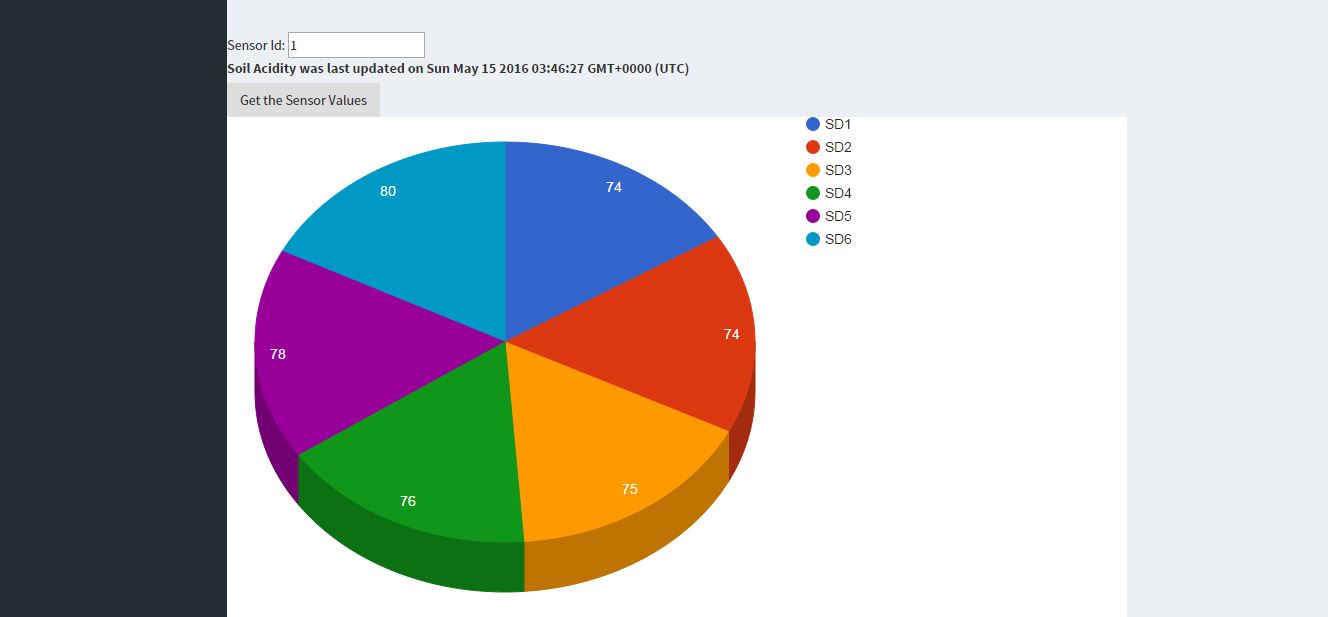
# Case Study

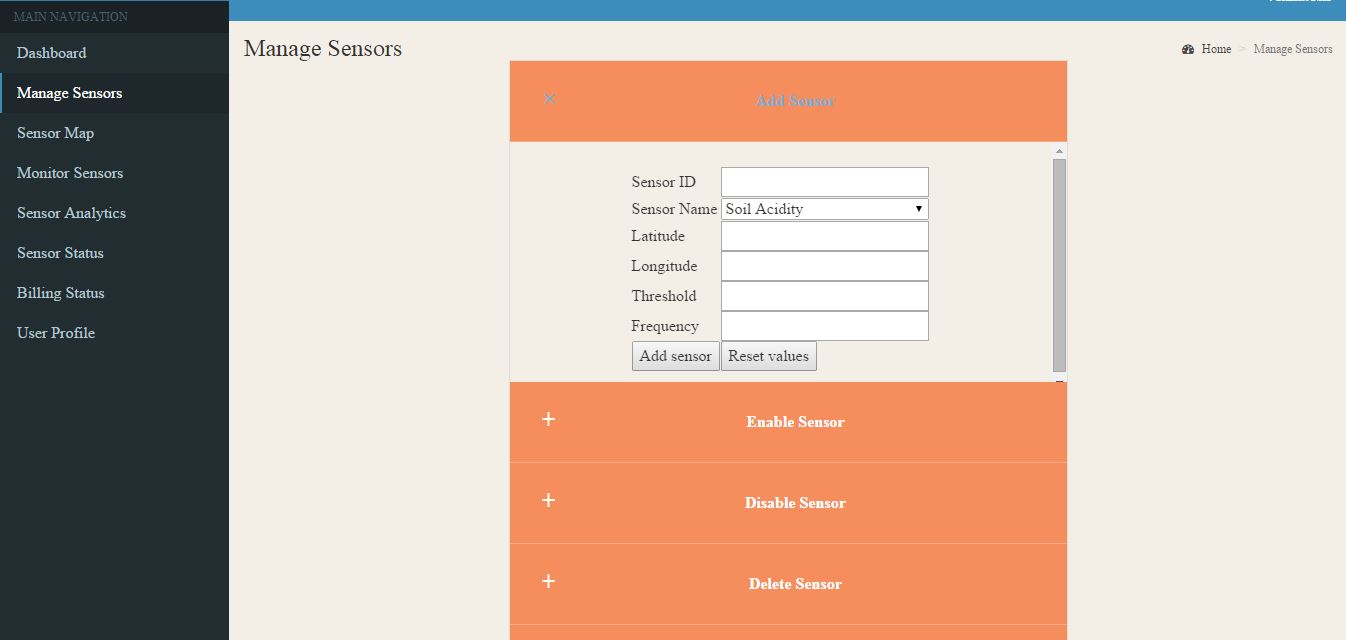
To implement our project, we login as a user and add sensors so that we can be able to view soil data of a particular location. User login page validates the user to redirect to his dashboard.



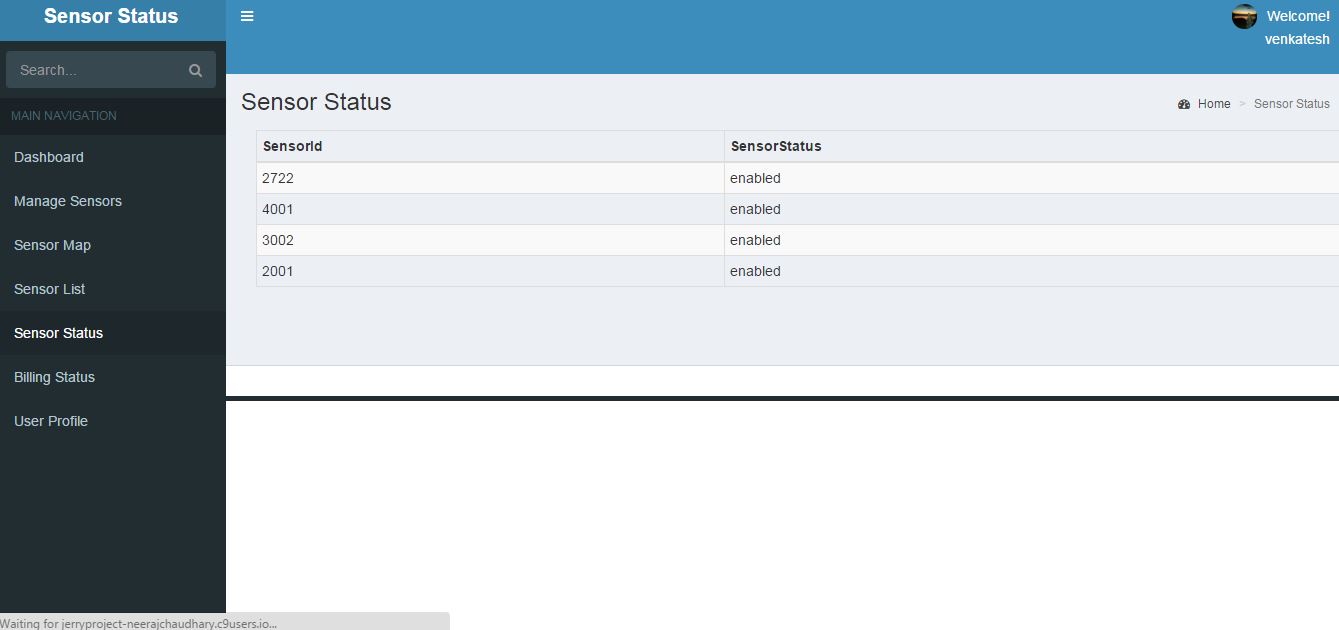
The user dashboard has the data analytics of his subscribed sensors.



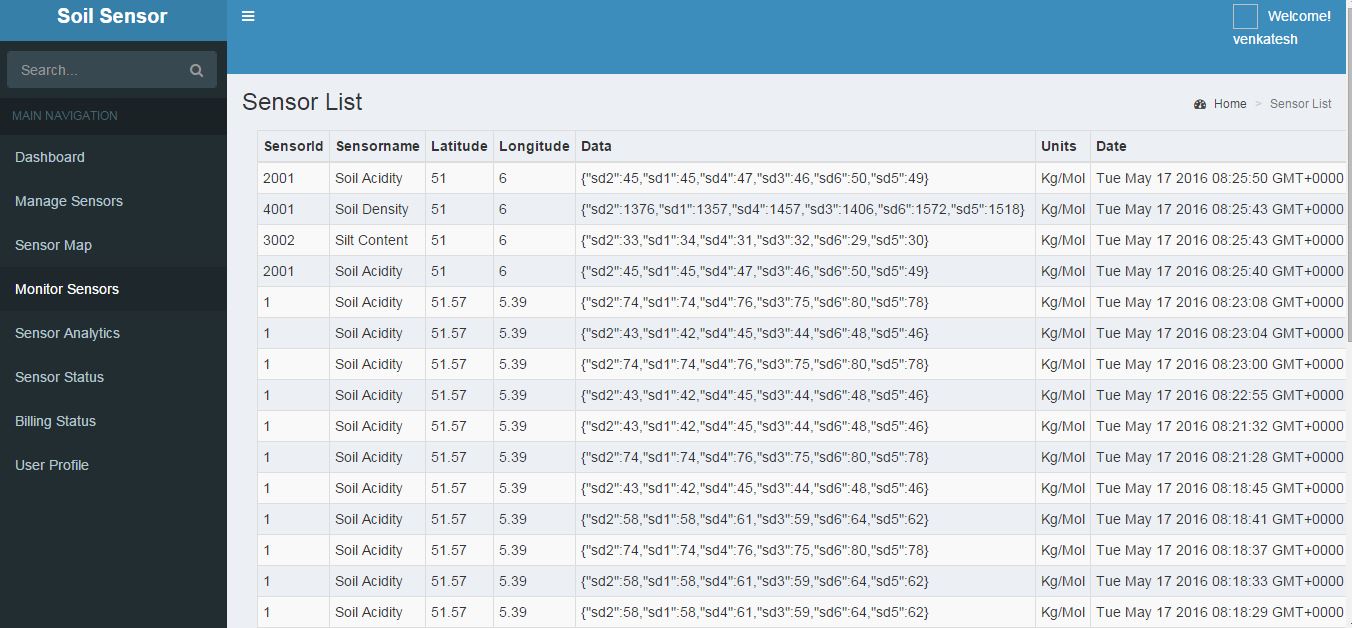




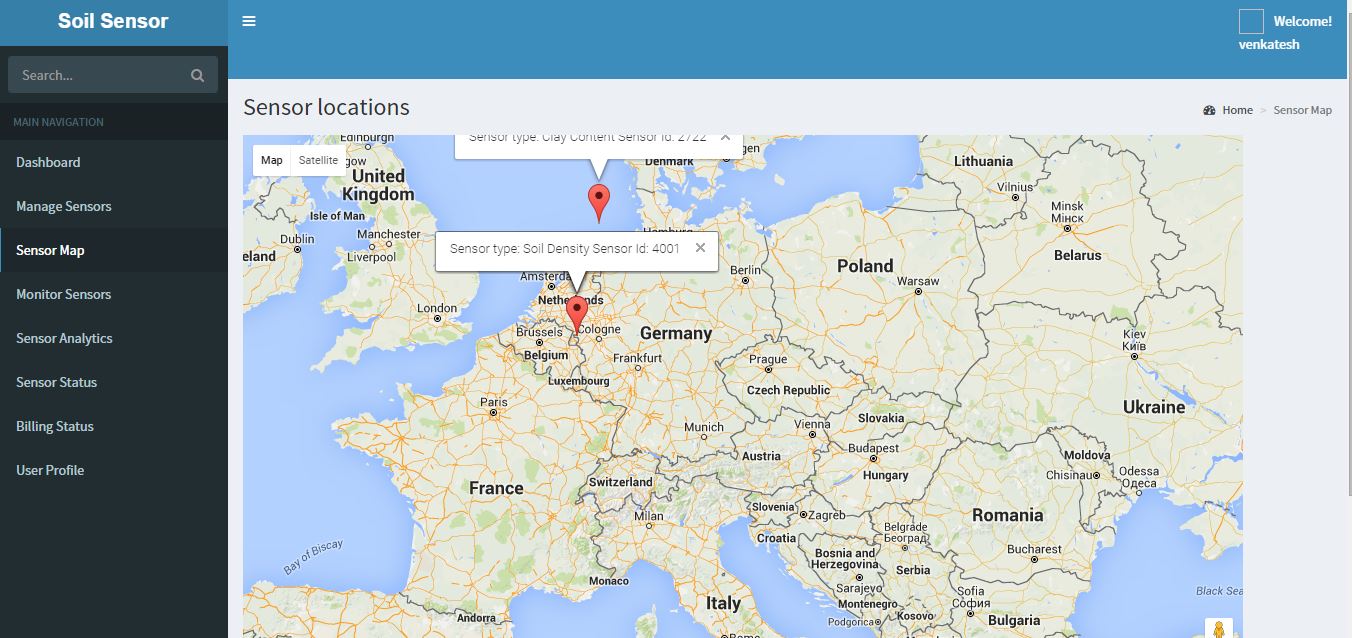
The user can add/delete sensors and enable/disable added sensors.



The added sensors status can be viewed under the sensor status tab.



The monitor sensors provides the list and details of all the available sensors and data log to the user. The entire log of sensor data is present in the Table list tab. It lists all the sensors of a particular user and the entire history of their values.



The map in the dashboard displays the various locations of the user subscribed sensors with a marker. The user inputs the latitude and longitude of the sensor location and the map displays it dynamically.

## C:\Users\SCS_USER\Desktop\9.JPG

The sensor analytics displays the sensor traffic data by means of an interactive graph.

# Conclusion

We have implemented an infrastructure for mobile sensor cloud. The user can add, delete, enable and disable sensor with just a click. The sensor values are generated from a virtualized sensor cloud. The data displayed is real time and accurate with respect to the location and sensor type. The billing model is very stable and hence would work for any types of users. The load balancer is implemented to acquaint large number of users without the server getting overloaded.

# References

1. Dr Jerry Gao, *Mobile Cloud*, lecture notes, CMPE-281 Cloud Technologies, San Jose State University
2. Gao, J.; Wei-Tek Tsai; Paul, R.; Xiaoying Bai; Uehara, T., "Mobile Testing-as-a-Service (MTaaS) -- Infrastructures, Issues, Solutions and Needs," High-Assurance Systems Engineering (HASE), 2014 IEEE 15th International Symposium on , vol., no., pp.158,167, 9-11 Jan. 2014