CMPE 281 - Cloud Technology

SAN JOSÉ STATE UNIVERSITY

AIR CONTROL SYSTEM

Team 23

Archana Ramalingam (SJSU ID: 010761114)

Clinton Rego (SJSU ID: 010757955)

Dan Lu (SJSU ID: 011028589)

Sreekar Medarametla (SJSU ID: 010807004)

Guided by
Dr. Jerry Gao

Fall 2016

Table of Contents

1. Introduction

Mobile sensor cloud is the novice research area emerged with the concepts of Mobile sensing and Cloud Computing. The communication among sensor nodes using Internet is often a challenging issue. Cloud Computing is a good solution for this. Sensor cloud collects data from physical mobile sensors and computes the large scale data. Cloud enables on-demand service, resource storage and virtualized instances to the end users. It also provides cost effective services which is an added advantage to users as many of them cannot afford a physical sensor for their small usage. Here the cloud provides the control to physical sensors on pay per use basis. Mobile Sensors are not fixed at a particular location; they have the capability to move around. This feature of the sensors supports the proposed cloud engine system. Cloud should have the capability to manage the mobility of such sensors so that the end users feel like they have continuous on-demand access of the physical sensors.

1.1 Motivation

Around half of the American population live in areas with air, which is at dangerous levels of pollution for breathing. Top five of the most polluted regions are in our California state [11]. Apart from this around 17.7 million people have asthma [12]. This is a critical issue that has to be given enough attention to. Also there are not many applications that tells the real-time level of pollution of a particular area. This kind of application can help people choose if an area is safe for them to travel or not and becomes indispensable for asthma patients.

This is the noble motivation behind our project. This project is about designing a air control system for detecting any accidental smoke due to any leakage, forecasting and analyzing weather conditions of the city using physical sensors deployed on the streets and the powerful mobile sensors inbuilt in today's smartphones or drones. Set of virtual sensors are set up which resemble physical sensors, these virtual sensors are dynamic and provide automatic provisioning of services. A user interface consisting of user dashboard is provisioned by sensor cloud for monitoring and administrating the virtual sensors like adding and deleting sensors etc.

1.2 Project Outcomes

The project can be divided into the following components:

Mobile Sensor management:

This service enables users to subscribe, modify or unsubscribe different available mobile sensors. The admin can add, delete or modify a sensor. This service acts as an interface between admin, users and mobile sensor clusters. The Connectivity and configuration is also taken care in this module.

Mobile Sensor Monitoring:
This module enables admin and user to monitor the user statistics, sensor availability and billing.

Mobile sensor Provisioning:
Mobile sensor network connection management, Connection service, Mobile sensor owner registration, Mobile sensor Connection and configuration service are included in this service.

Scalability support:
Load balancing in sensor clouds is a technique that distributes the excess dynamic local workload evenly across all the nodes. It is used for achieving a better service provisioning and resource utilization ratio, hence improving the overall performance of the system. In our project, incoming service request are coming from different users are received by the load balancer and then distributed to the virtual sensor cluster for the proper load distribution.

Dashboard User Interface:
User can add, delete and modify the sensor data using this web based user interface. A user can visualize the collected data, admin can use this interface to configure and control mobile sensor data.

2. Infrastructure Analysis and Design

2.1 System Architecture Design

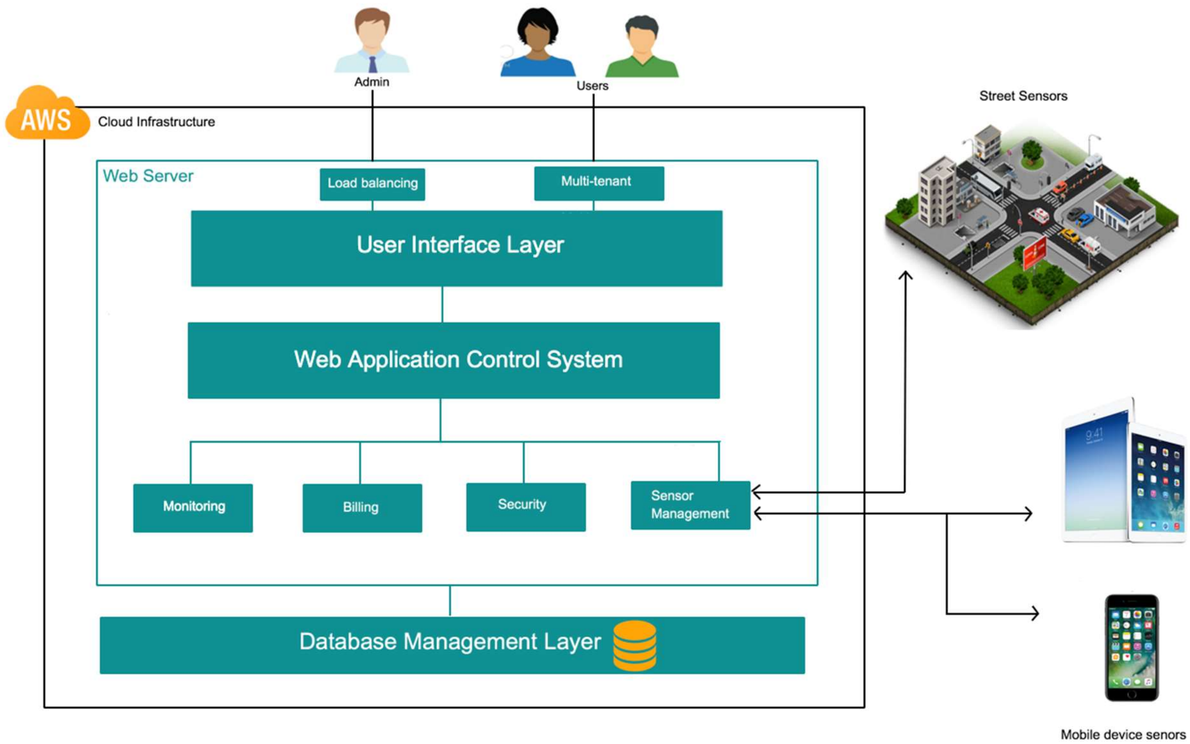

Figure 1: System Architecture Design

The architecture design of our project is as shown in Figure 1. The entire application is hosted on AWS EC2 server. The users and admin both interact with the UI layer. In order to handle heavy traffic and to serve many clients, there will be a load balancer and the system will support multi-tenant operations. The request will then be received by the core application layer to monitor, compute and respond back to the users. The sensor data will be collected from the sensors on the street as well as mobile sensors such phones, tablets etc. The application layer will also have a MySQL database to persist user, application and sensor information.

2.2 Deployment Infrastructure

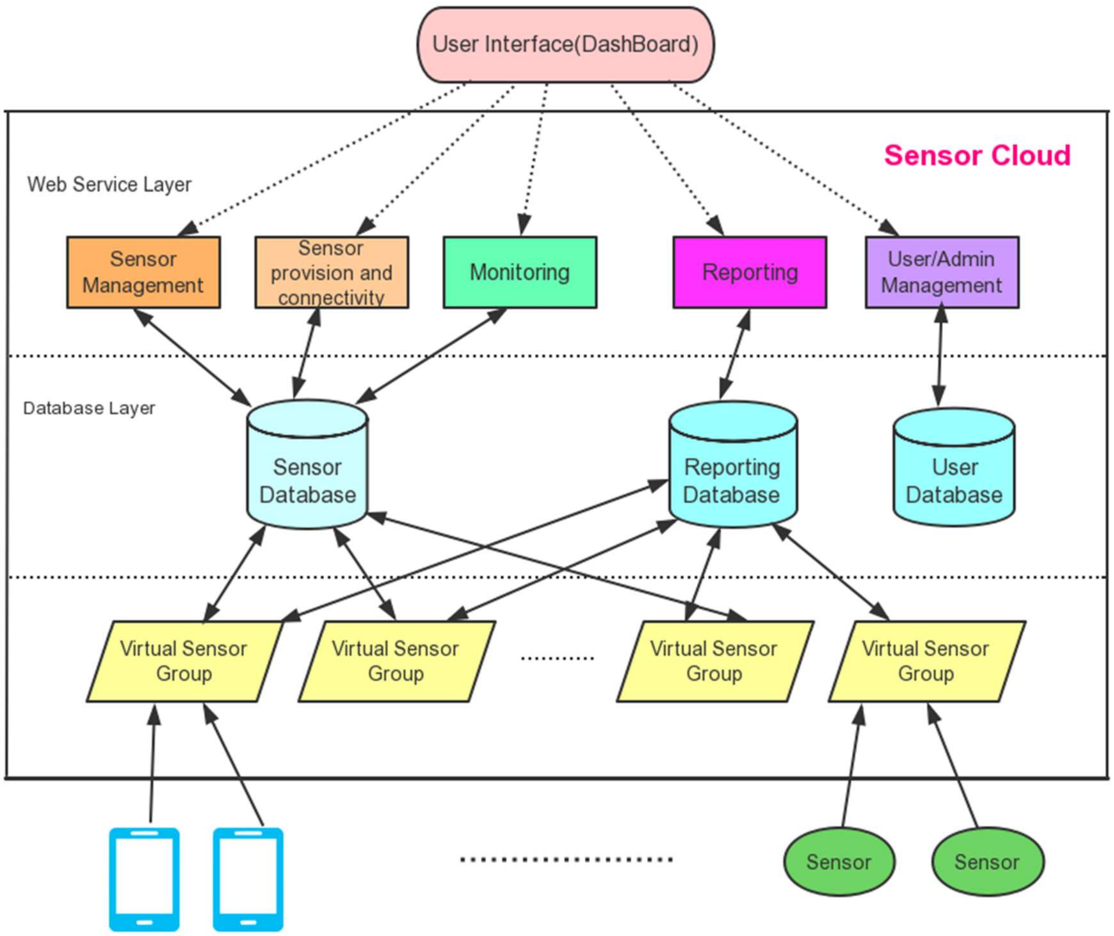

Figure 2: Deployment design diagram

The Figure 2 demonstrates the deployment flow path of the sensor cloud. User or administrator using sensor cloud service through dashboard user interface. Sensor management allow users to register, configure, and manage different types of sensor. Mobile virtual sensor resources are automatic on demand provision to users. Mobile sensor provision in charge of sensor cluster allocation, resource collection and data storage. Mobile sensor monitoring allows users to track the sensors' status, type and other detail information. Mobile reporting or we could call billing service provides the resource usage information and billing statement for users.

2.3 Component based UML Design

The main goal of the UML diagram shown below is to define each functional components of the Cloud system. The path taken by each action or task is also clearly shown. The main sub functions of each component is also briefed.

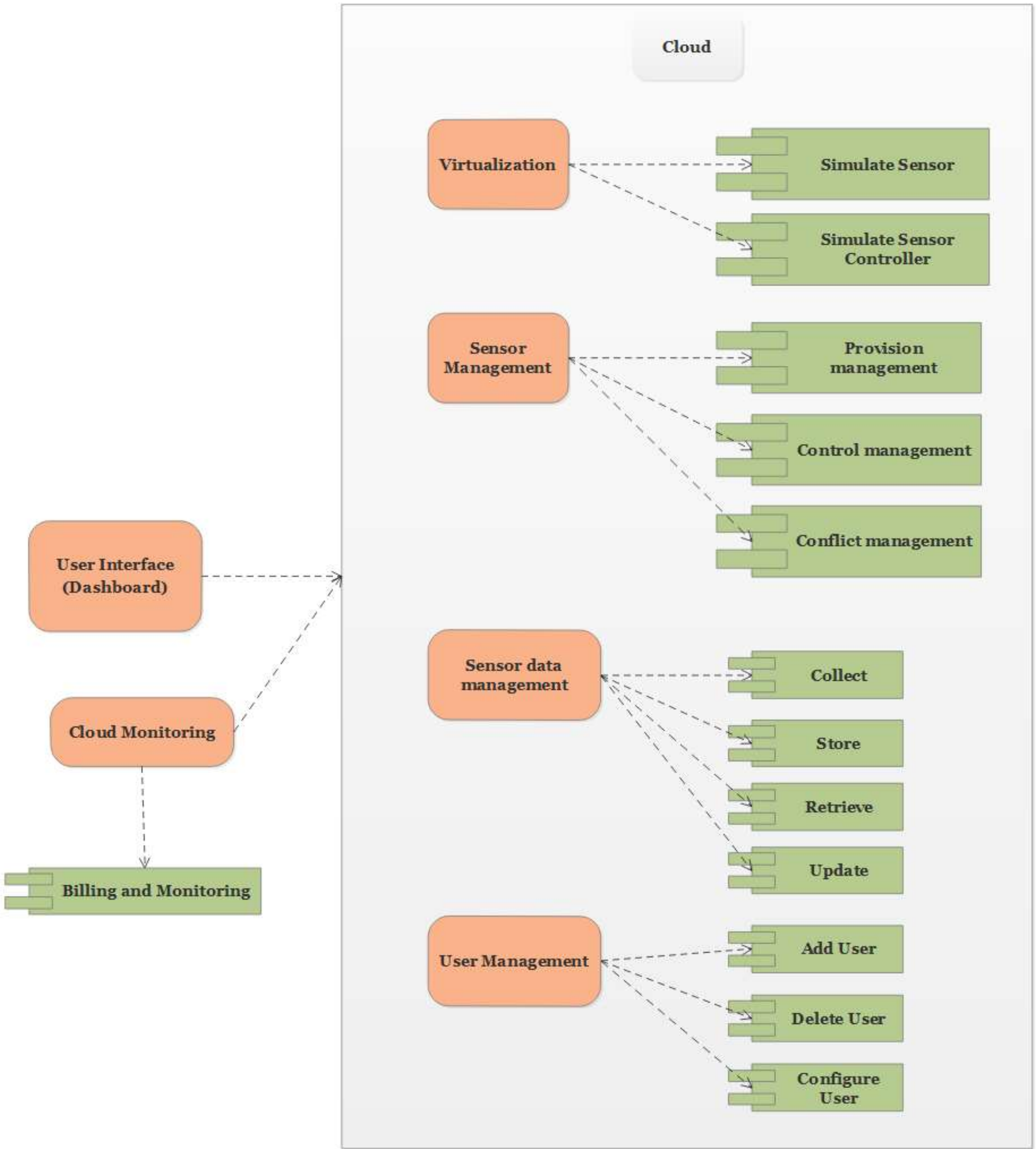

Figure 3: Component design diagram

2.4 Class Design

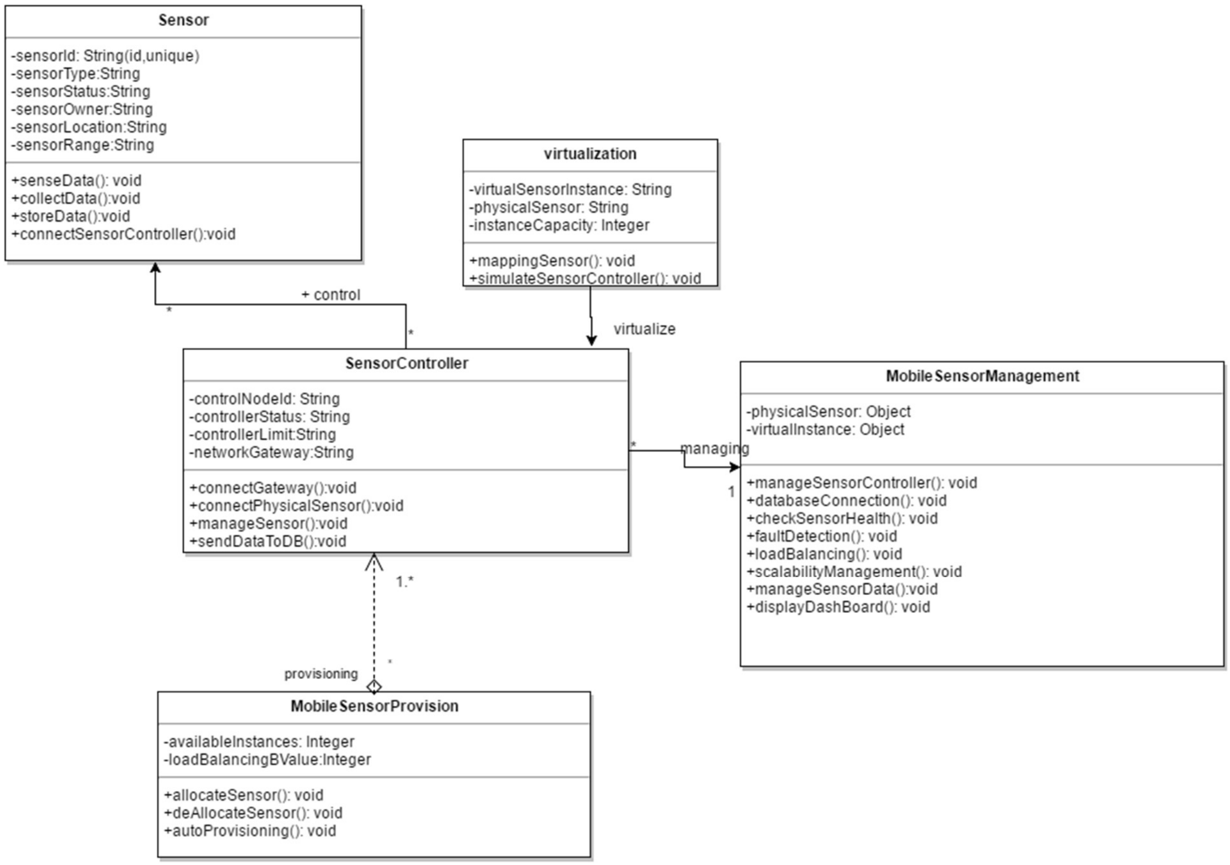

Figure 4: Class Design

3. Function Component Design

3.1 Mobile Sensor Management

The mobility feature of our sensors can be implemented by putting them on a drone. Since, we work on the cloud platform part of this application, we will stay focused on the same. From user perspective, Sensor cloud system is a multi-tenant platform which permits multiple user to register and request virtual sensor resources on demand, without impacting each other. User joining the sensor cloud first need to create account and establish authentication mechanism. After that, users are able to register and request sensor resources on the assigned logic space of the sensor cloud. When the user submits the registration request with the configuration, the sensor management system is able to prepare the required virtual sensor resource automatically. When the user wants to edit or remove certain sensors from the cloud, the sensor management system is able to update the association of those sensors upon request.

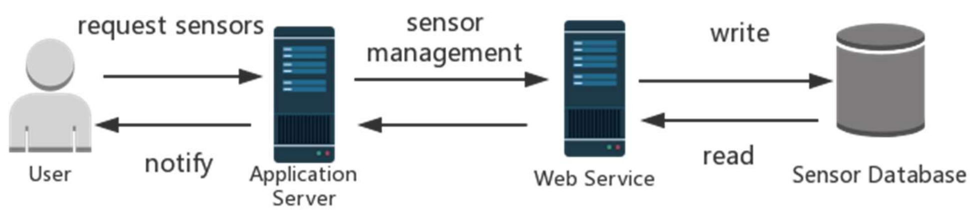

Figure 5: user request sensor workflow

3.2 Mobile Sensor Monitoring

The function of mobile sensor monitoring is to provide user or administrator the real-time information about sensor (for example: sensor status, location, type etc.) and resource usage (including CPU utilization, network usage, storage usage etc.). All that information is stored in the database and will be processed and displayed to users through the dashboard. In order to make the information easier to read, the comparison metrics and statistical model will be implemented before displaying to dashboard. Overall, the monitoring system consists of four parts:

- Sensor data collector
- Data processing and visualization
- Resource usage collector
- Resource usage reporter

3.3 Mobile Sensor Provision

Mobile sensor provision is the allocation of a cloud provider's sensor and computing resources to a user. When a user sends a request to the mobile sensor cloud, it must create the appropriate number of virtual sensor resources and allocate it to support them. The process is conducted in several different ways: advance provisioning, dynamic provisioning and user self-provisioning.

With advance provisioning, the cloud user contracts with the provider for services and the provider prepares the appropriate resources in advance of start of the service. The user is charged a flat fee or is billed on a monthly basis. Dynamic provisioning provides users a pay-per-use model by allocating resources when user request and removes them when user do not need. With user self-provisioning, the user purchases resources through a web form, creating a customer account and paying for resources with a credit card. The cloud provider's resources are available for customer use within hours, if not minutes. In our project, we will adopt advance provisioning.

3.4 Scalability Support

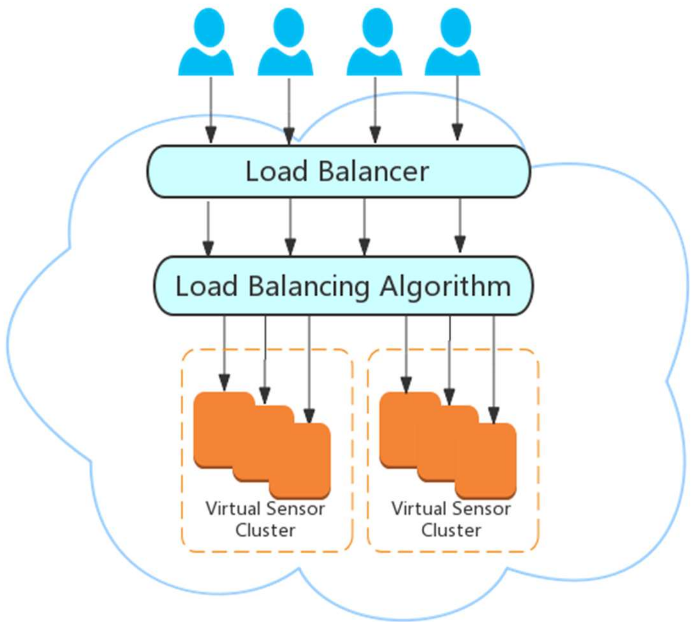

Figure 6: load balancing diagram

Load Balancer acts as a reverse proxy and distributes network or application traffic across a number of servers. In our project, we host our application on AWS. AWS provide a service named Elastic Load Balancing. ELB is a load balancing solution that automatically scales its request-handling capacity in response to incoming application traffic. It increases the availability of our application by configuring the load balancer to route traffic to EC2 instances. You can add and remove instances from your load balancer as your needs change, without disrupting the overall flow of requests to your application.

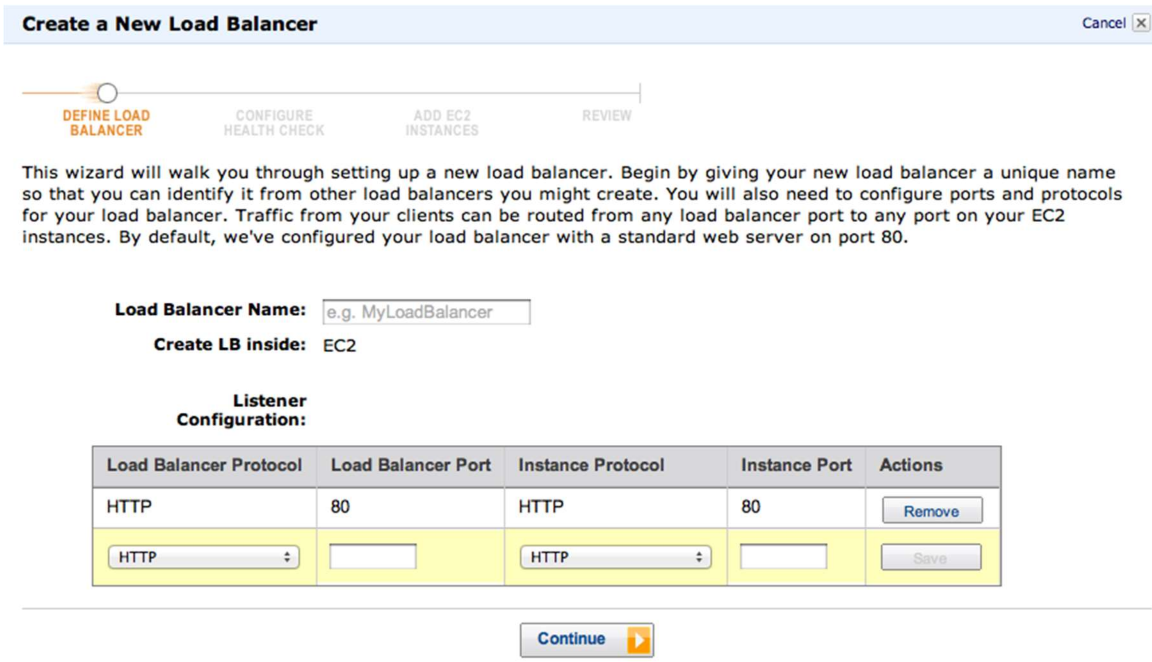

Figure 7: Load balancer configuration

For load balancing algorithm, we use a currently prevalent load balancing algorithm Round Robin algorithm in our project. It uses the round robin scheme for allocating user's service request. The virtual sensor resources are provisioned to the users on first-cum-first-serve basis and scheduled in time sharing manner. It means each virtual sensor cluster receive some traffic to avoid some server is heavily loaded or someone is lightly loaded.

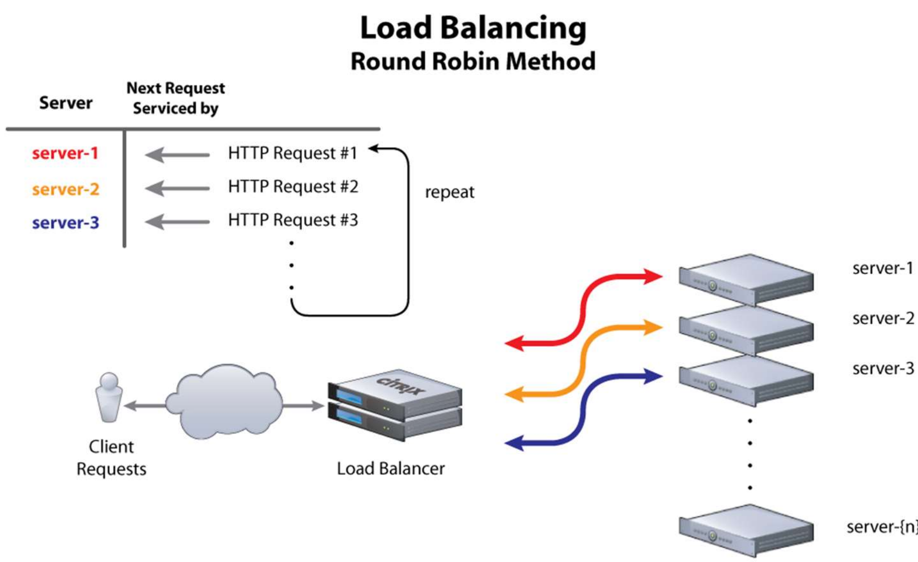

Figure 8: Round Robin load balancing algorithm

4. Implementation

4.1 Project Technologies

- Front end technology: Node.js, AngularJS, Bootstrap
- AWS infrastructure: EC2 Service, Elastic load balancer, CloudWatch
- Storage Service: MySQL

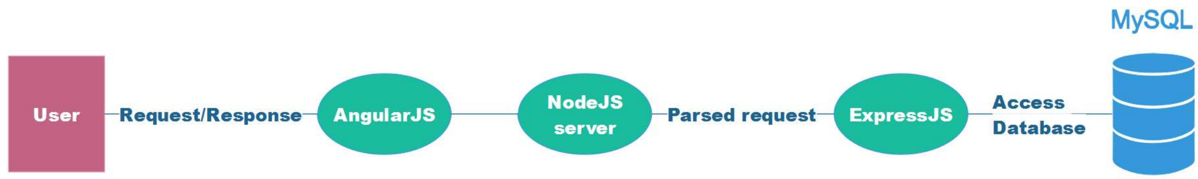

Figure 9: Application Layout

We are using Node.js to create server environment. AngularJS is used for front-end designing. Sensor data, user and admin data, billing details are all stored in MySQL database. Amazon's Elastic load balancer is used to provide the scalability and load balancing feature for our application.

4.2 Why AWS?

AWS has been the choice of cloud provider for various reasons as listed below:

- Flexibility to port application and data to other cloud providers
- Easy to use for beginners
- Cost-effective
- Reliable
- Secure
- Scalable

4.3 Screenshots

- User interface for user/admin register or login
- Mobile sensor cloud management
- Sensor Configuration
- Mobile sensor provision
- Mobile sensor monitoring

Figure 10: Home page 1

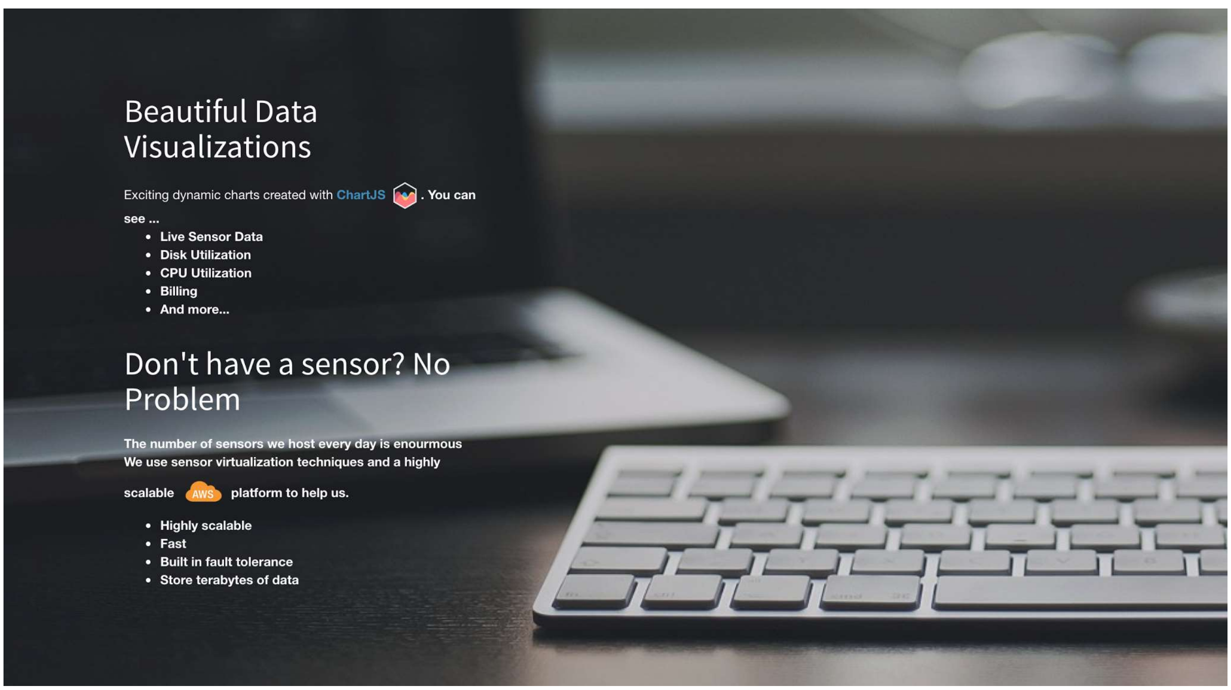

Figure 11: Home page 2

Figure 12: **Home page 3**

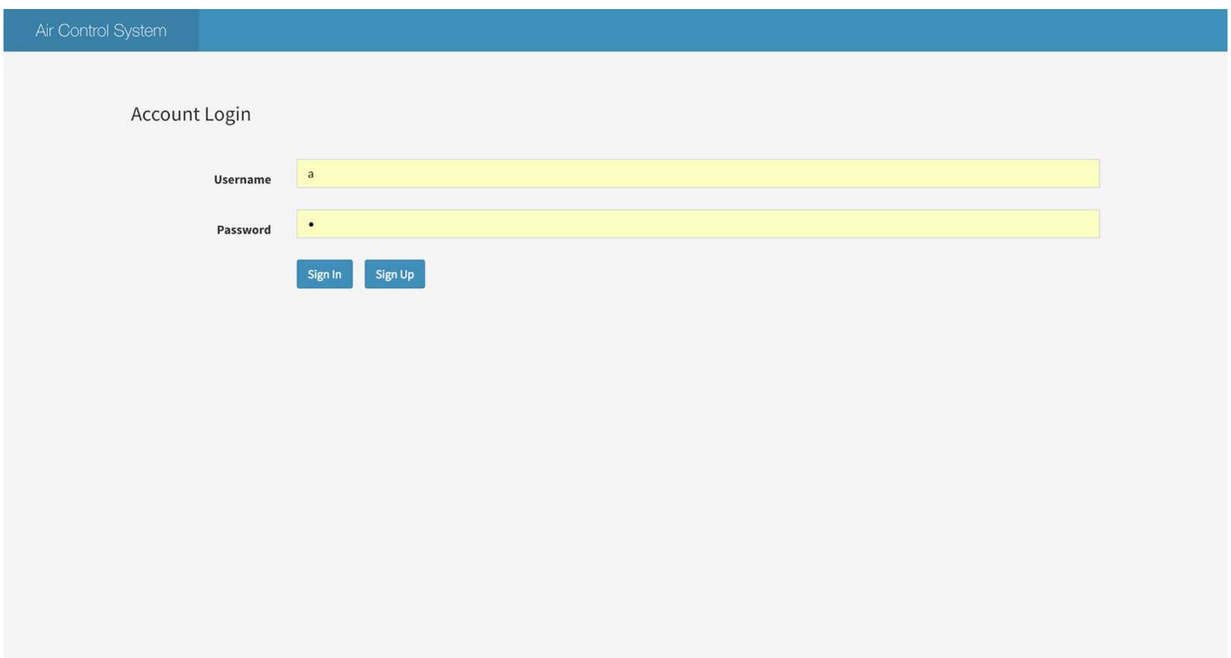

Figure 13: **Login page**

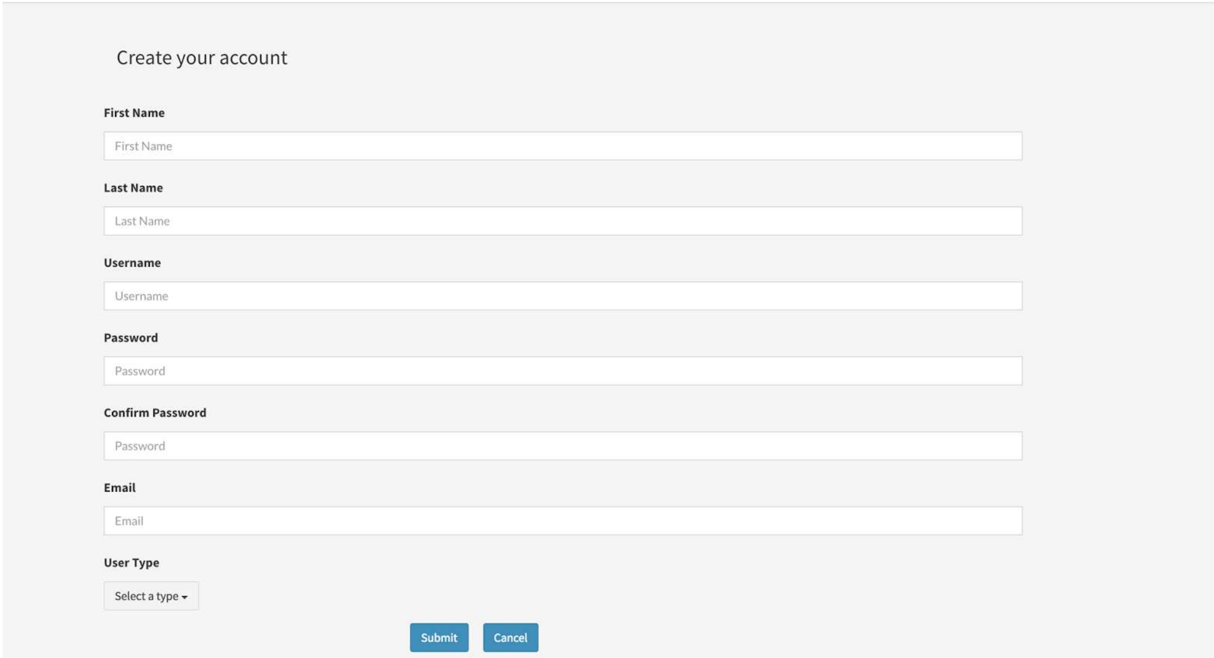

Figure 14: Registration page

Mobile sensor cloud management:

Figure 15: Admin Dashboard displaying sensor statistics

Sensor Configuration:

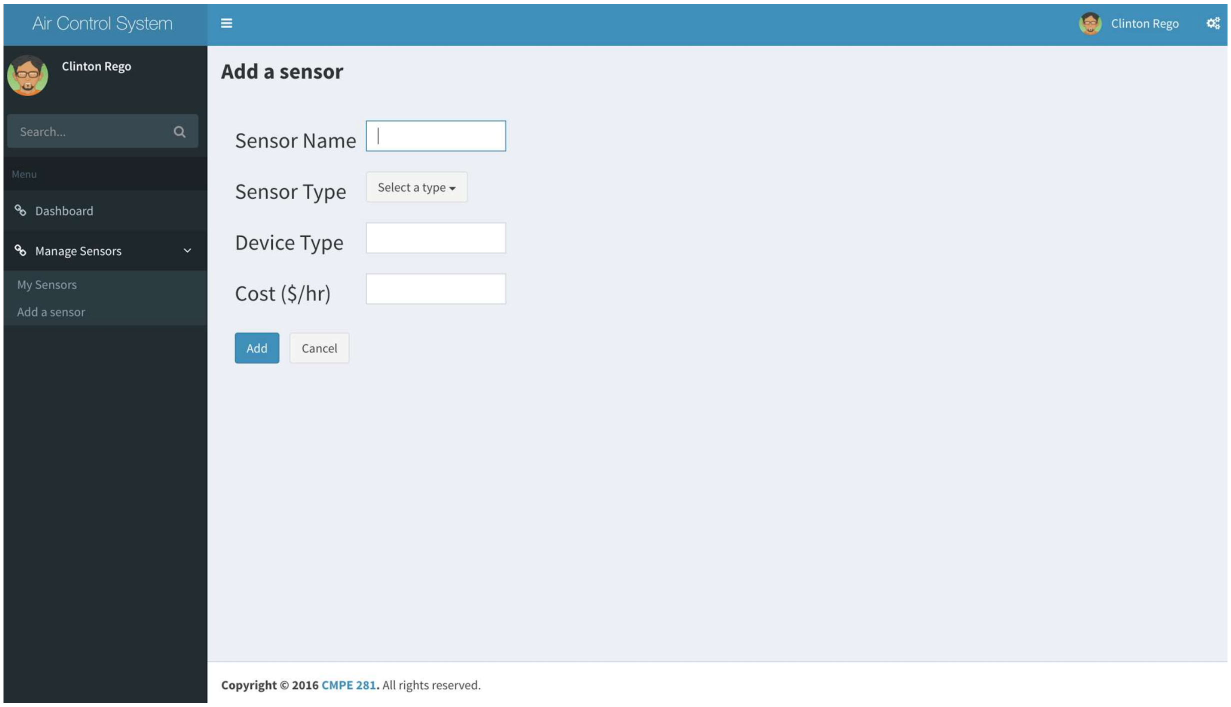

Figure 16: Configuration page for admin to add a new sensor

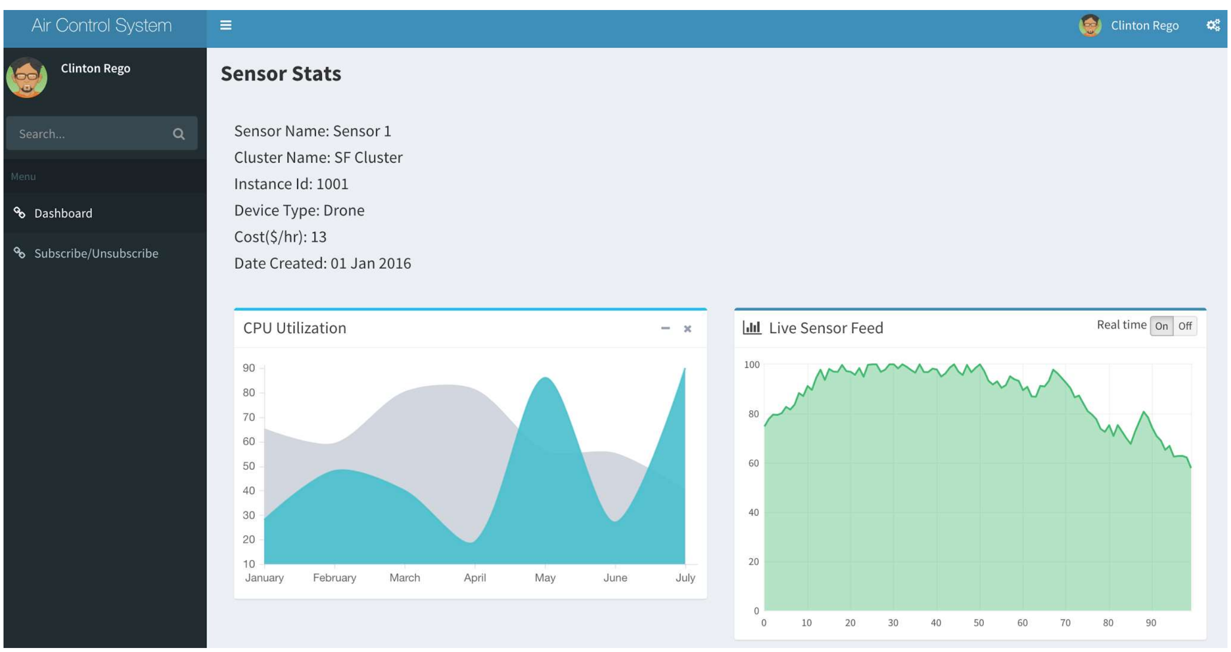

Figure 17: Admin can view the statistics of the sensors added by him

Sensor Management for Admin:

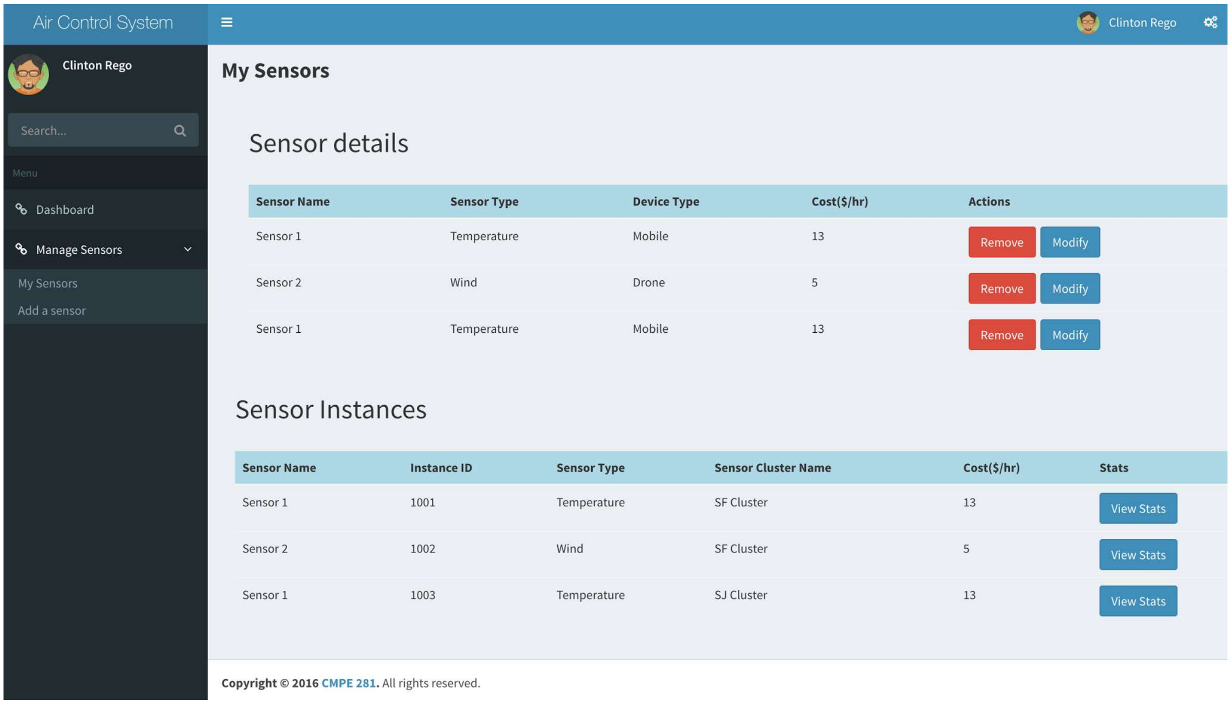

Figure 18: Admin can view and modify the sensors added

Sensor Management for User:

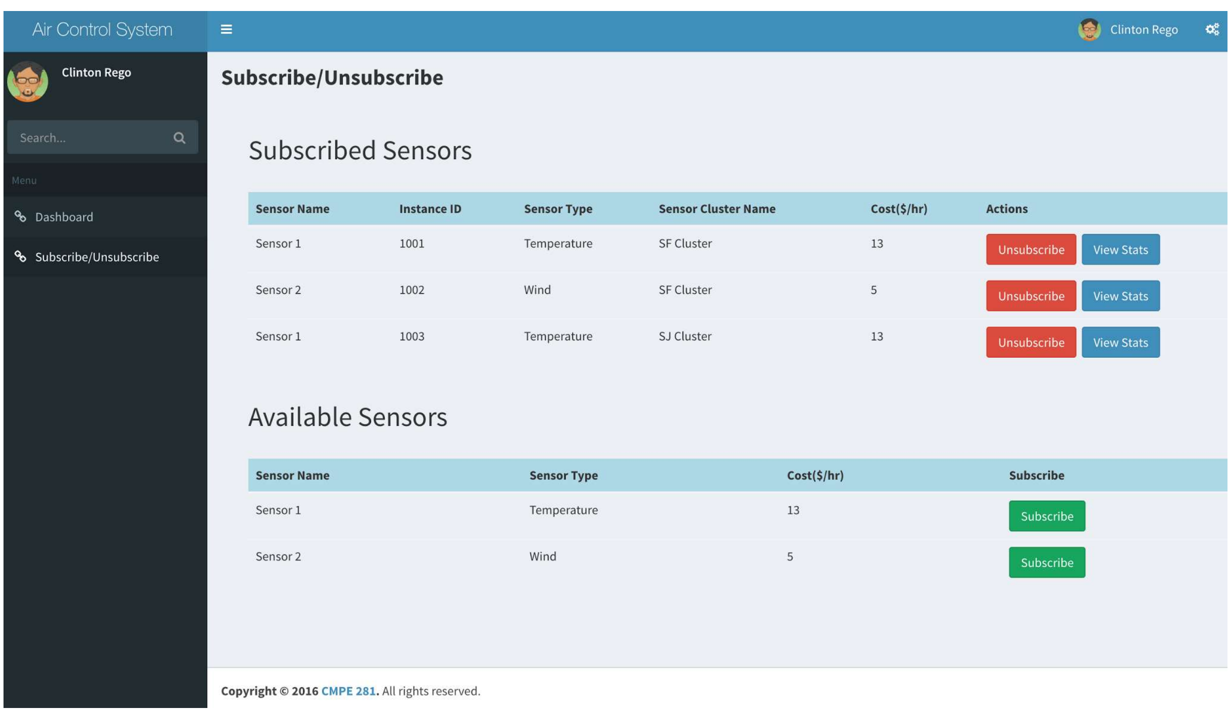

Figure 19: User can view and modify the subscribed sensors

Resource Usage Monitoring:

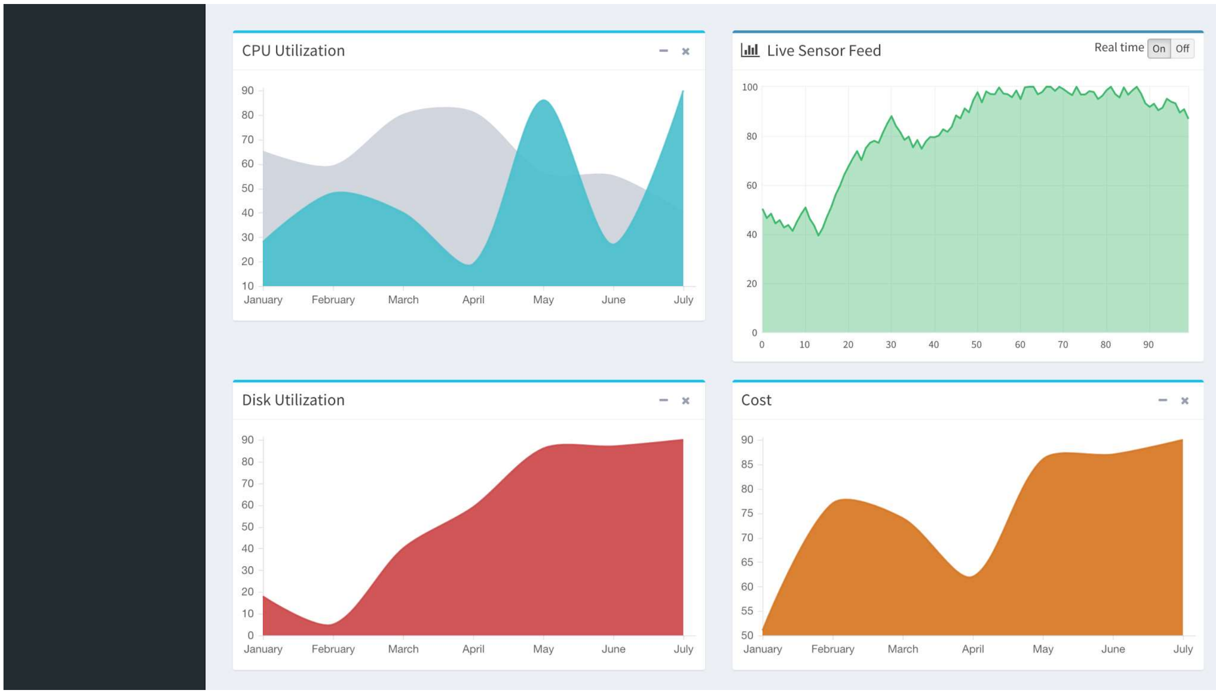

Figure 20: User can view amount of resources used

5. Database Design

5.1 ER Diagram

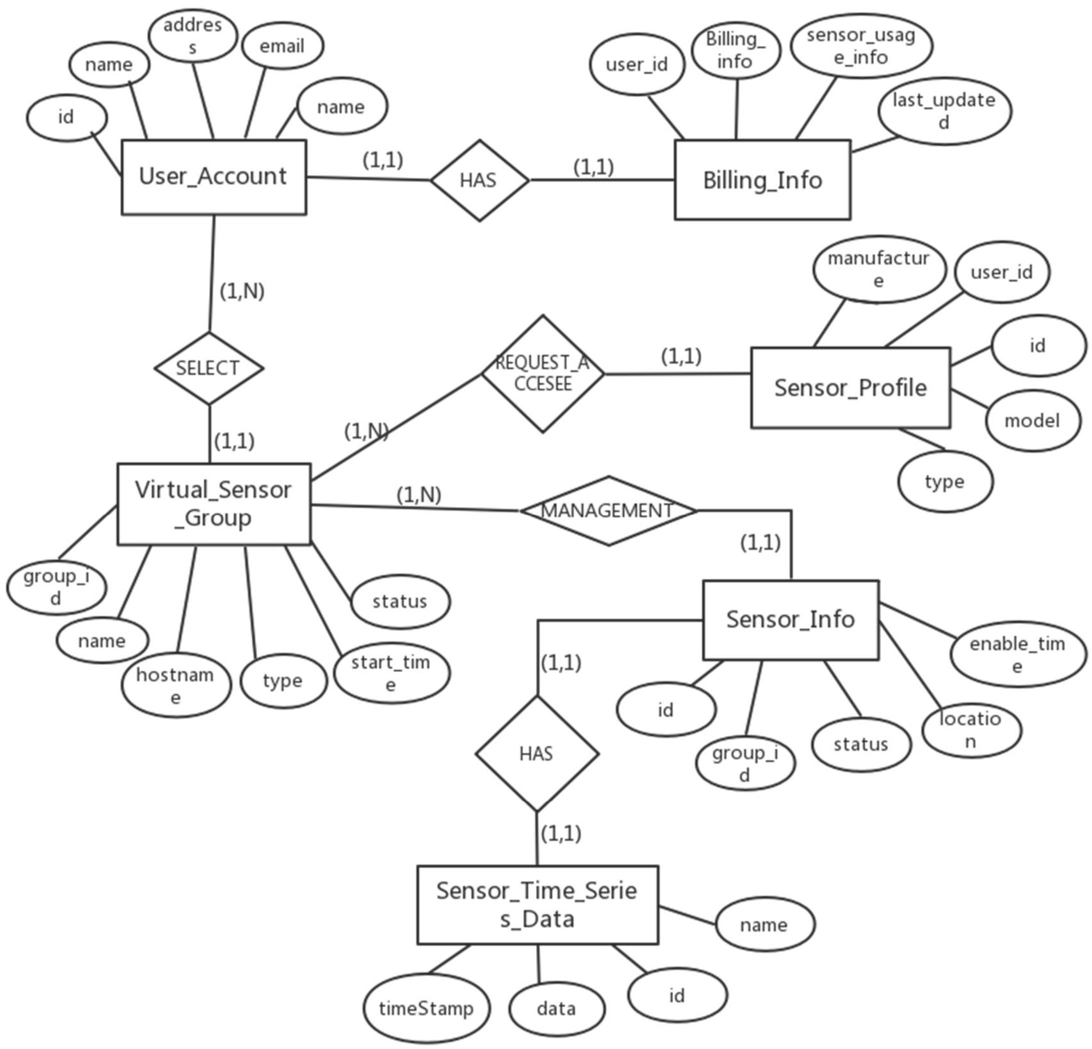

Figure 21: ER Diagram

5.2 Database Schema Design

Table Name: users

Column Name	Type	Description
First Name	varchar(40)	

Last Name	varchar(45)	
Username	varchar(45)	Primary Key
email	varchar(45)	
password	varchar(45)	
UserType	varchar(45)	

Table Name: sensors

Column Name	Type	Description
Name	varchar(45)	Primary Key
Type	varchar(45)	
DeviceType	varchar(45)	
Cost	float	
AdminUserName	varchar(45)	
start_time	float	

Table Name: sensorinstances

Column Name	Type	Description
Name	varchar(100)	
Instance ID	int(11)	Primary Key
Type	varchar(60)	
Cluster Name	varchar(70)	
Cost	float	
DateCreated	date	
UserName	varchar(100)	
AdminUserName	varchar(100)	
bill	int(11)	

6. Conclusion

The main goal of the project was to create a Mobile sensor cloud application that helps users to subscribe to and manage sensors, with a pay-as-you-go billing model. The application supports virtualization, provision, management, monitoring, billing and an interactive well-designed user interface for user dashboard. It runs on the AWS platform running efficiently with the help of the services provided by AWS. Load balancing has been implemented with AWS ELB, which uses round-robin algorithm to perform balancing.

References

[1] AWS Cloud products and services https://aws.amazon.com/products/

[2] C. Chang, S. N. Srirama and M. Liyanage, "A Service-Oriented Mobile Cloud Middleware Framework for Provisioning Mobile Sensing as a Service," Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on, Melbourne, VIC, 2015, pp. 124-131

[3] Misra, A. and Lim, L. "Optimizing Sensor Data Acquisition for Energy-Efficient Smartphone-based Continuous Event Processing". IEEE MDM, 2011

[4] N. Cao, S. Brahma and P. K. Varshney, "Towards cloud sensing enabled target localization," Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on, Monticello, IL, 2014, pp. 537-544

[5] H.-H. Tran and K.-J. Wong, "Mesh networking for seismic monitoring - the Sumatran cGPS array case study", 2009 IEEE Wireless Communications and Networking Conference, Apr. 2009.

[6] X. Sheng, X. Xiao, J. Tang and G. Xue, "Sensing as a service: A cloud computing system for mobile phone sensing," Sensors, 2012 IEEE, Taipei, 2012, pp. 1-4. doi: 10.1109/ICSENS.2012.6411516

[7] Mihui Kim, Mihir Asthana, Siddhartha Bhargava, Kartik Krishnan Iyyer, Rohan Tangadpalliwar, Jerry Gao, "Developing an On-Demand Cloud-Based Sensing-as-a-Service System for Internet of Things", Journal of Computer Networks and Communications, Volume 2016, Article ID 3292783.

[8] Ting Liu, Christopher M. Sadler, Pei Zhang, Margaret Martonosi, Implementing Software on Resource-Constrained Mobile Sensors: Experiences with Impala and ZebraNet, 2004.

[9] Madoka Yuriyama, Takayuki Kushida, "Sensor-Cloud Infrastructure Physical Sensor Management with Virtualized Sensors on Cloud Computing", 2010 13th International Conference on Network-Based Information Systems, Sept 2010.

[10] Kian-Lee Tan, "What's Next? Sensor Cloud?", Proceeding DMSN '10 Proceedings of the Seventh International Workshop on Data Management for Sensor Networks, pp. 1-1, ACM New York, 2011.

[11] http://www.cbsnews.com/news/air-pollution-dangerously-high-for-almost-half-of-us/

[12] http://www.cdc.gov/nchs/fastats/asthma.htm