CMPE 281-01 Cloud Technologies

Project System Design Document

Spring-2016

SAN JOSÉ STATE UNIVERSITY

Submitted to:

Dr. Jerry Gao

By

Aditya Chauhan(010816676)

Ashay Argal(010822513)

Divyakumar Patel(010819822)

Shalin Amin(010823826)

Table of Contents

Table of Figures

Acknowledgement

*We would like to thank **Dr. Jerry Gao** and his assistants **Khushali Desai** and **KoushikRam KunathurThikshaja** without whom this project would not have been completed. We thank our colleagues from San Jose state University who provided insight and expertise that greatly assisted the project.*

1. Introduction

The objective behind developing this application is to develop, implement and validate a mobile sensor cloud platform for managing and providing on demand sensor cloud service. Features of the application other than sensor controller management and provisioning are dashboard user interface, scalability management - load balancing and Sensor Monitoring.

1.1 Project Introduction

As connectivity, storage, and compute become more pervasive, The Internet of things (IoT) is becoming one of the most relevant trends in the history of the software engineering industry. In this IOT era sensors are deployed in various products and places in large number for gathering real time data. After gathering data, it is analyzed to provide insights to the clients. Research predicts that the worldwide Internet of Things market will grow to $1.7 trillion in 2020 with a compound annual growth rate (CAGR) of 16.9%, connecting 50 billion devices. The need is felt by industries to manage these devices and making maximum use of huge amount of data generated.

The project gathers data from different sources like sensors and sensor-networks to provide access to real-time and historic data with easy user interface. It will also provide facilities like profile management, role management, monitoring and billing.

Main functionalities that the application would consist are: -

1. **Provision and Management of Virtual Sensor**
   This functionality will allow user to create, manage and delete different virtual mobile sensors.
2. **Provision and Management of Sensor Controller**
   This functionality will allow user to manage various type of sensor controllers.
3. **Provision and Management of Scalability and Load Balancing**
   This functionality will enable a feature of load balancing and make application scalable so that it can add/remove number of sensors whenever needed.
4. **Virtualization of sensor clouds**
   This functionality will simulate a virtual sensor by providing weather sensing data as virtual sensor controller will be controlling sensors.
5. **Monitoring of Virtual Sensors**
   This will help user in monitoring the sensing data gathered by a virtual sensor.

6. **User Dashboard**
   User Dashboard will provide a way to users interact with application.

7. **Billing**
   Application follows pay as you go model. We have enabled metering service that gathers usage data of users.
8. **Admin Panel**
   By using this functionality admin will be able to login and monitor and manage different virtual sensors.

2. Deployment Architecture Design

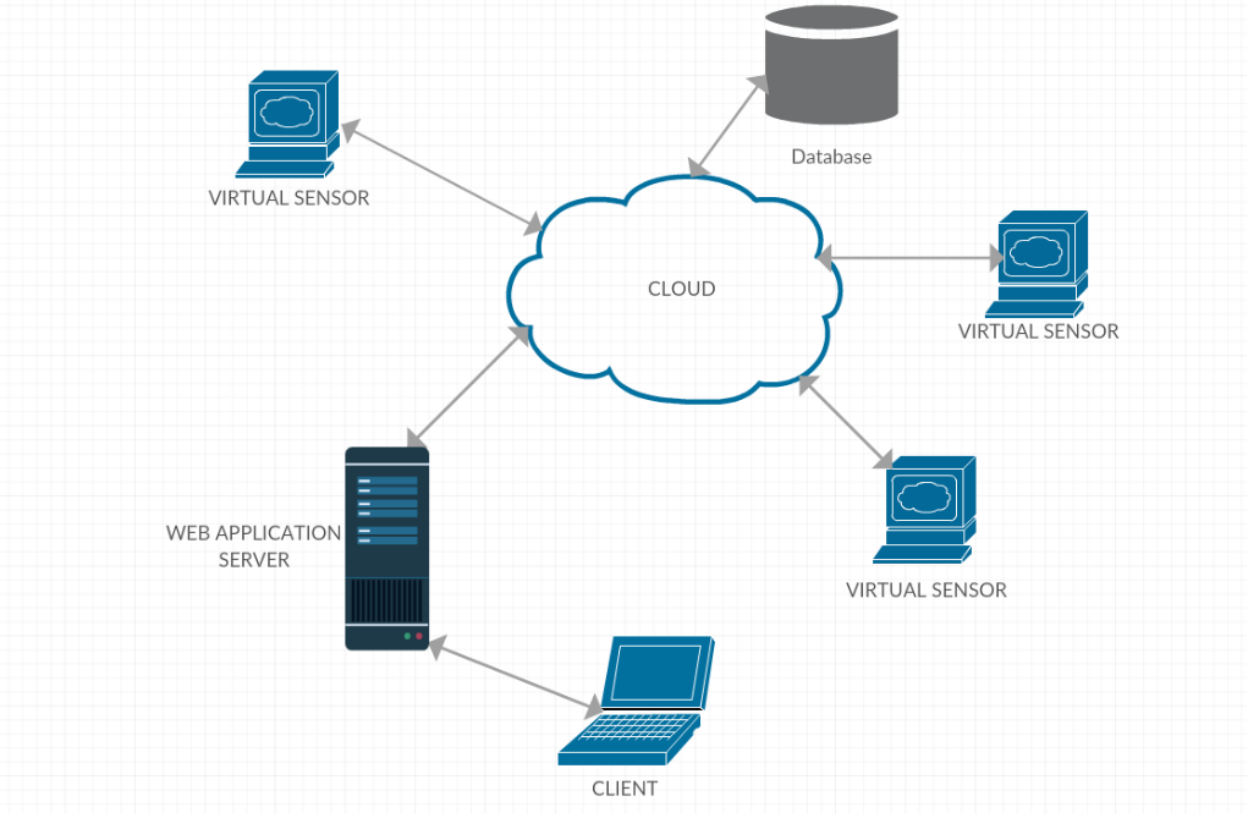
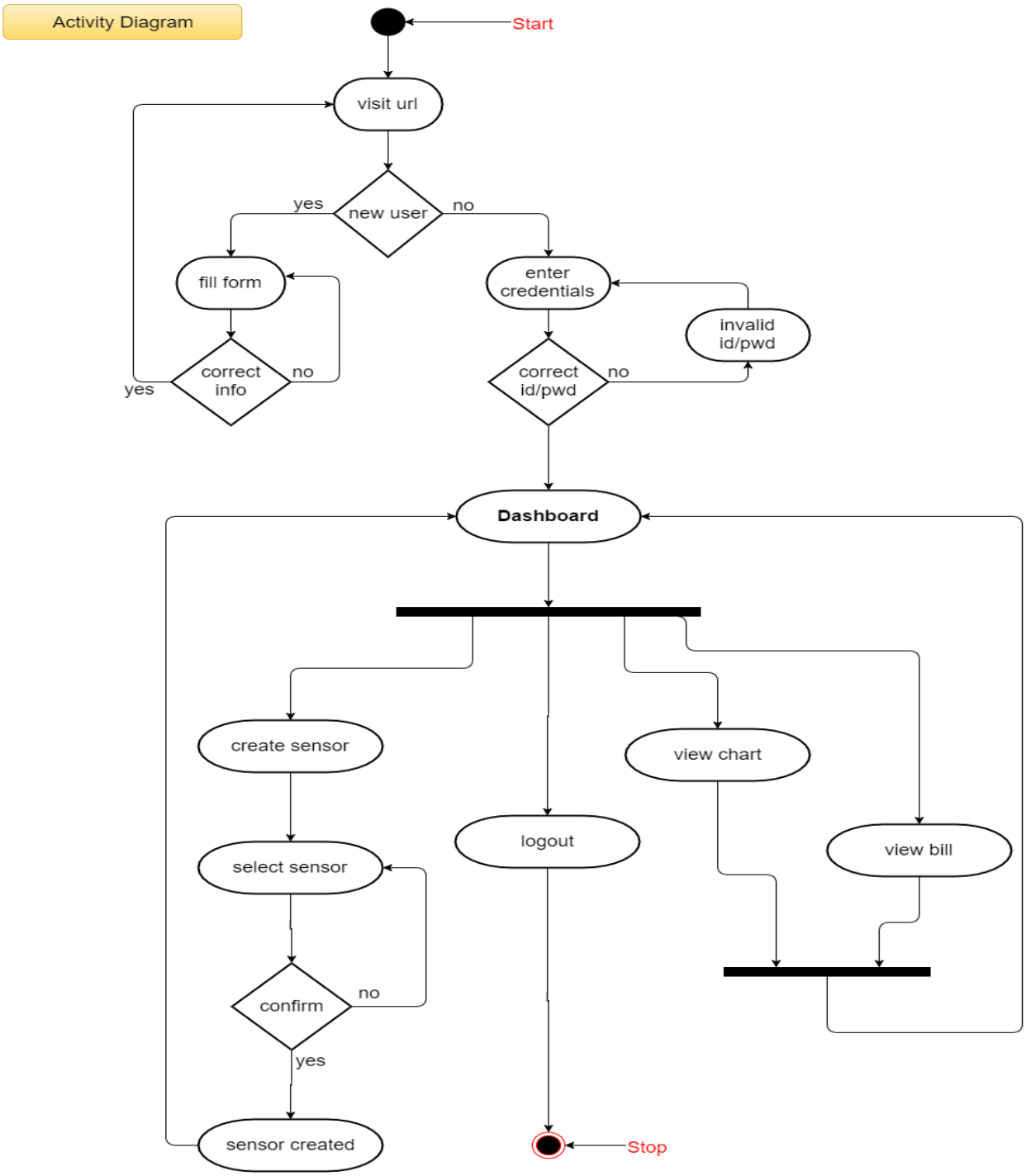
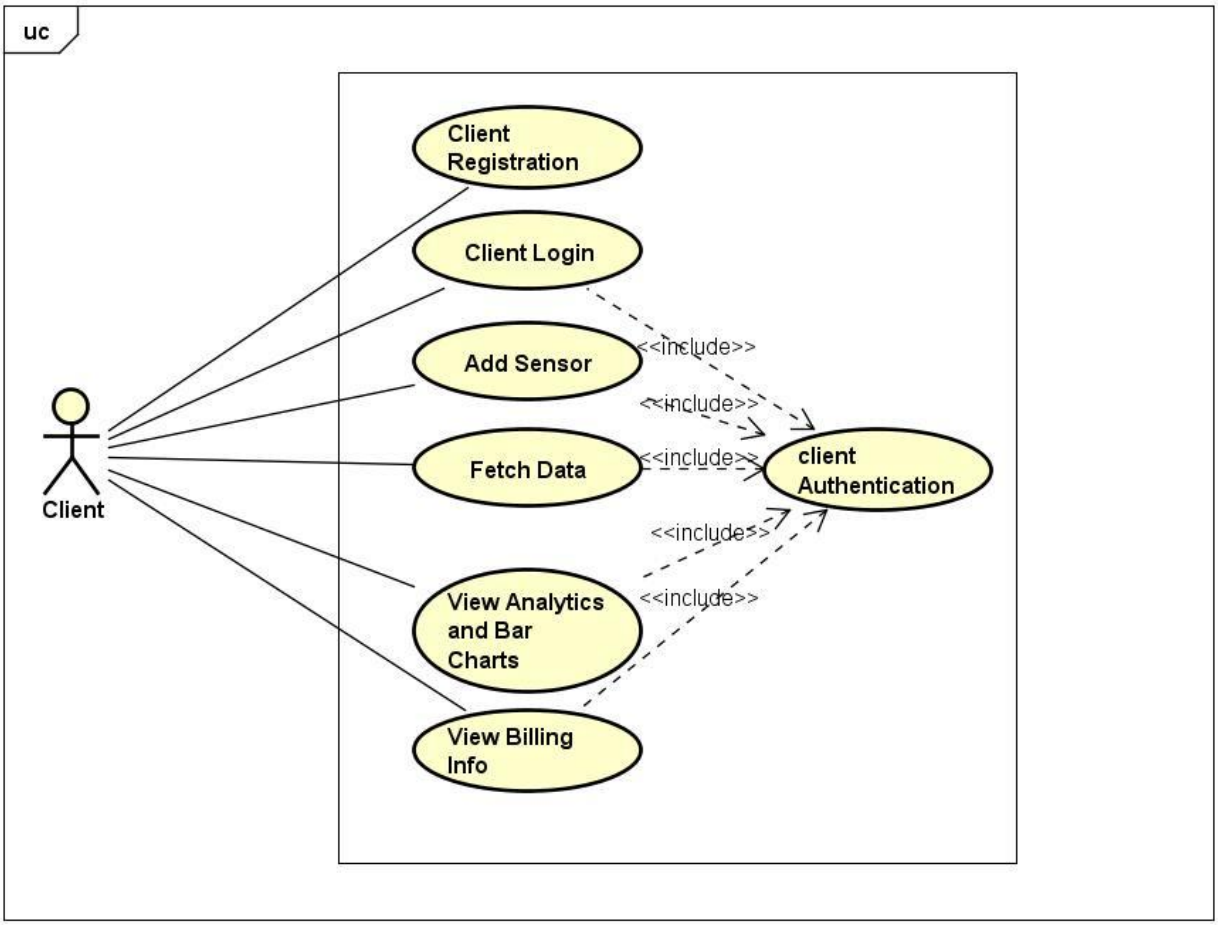


Figure 1 : Deployment Architecture

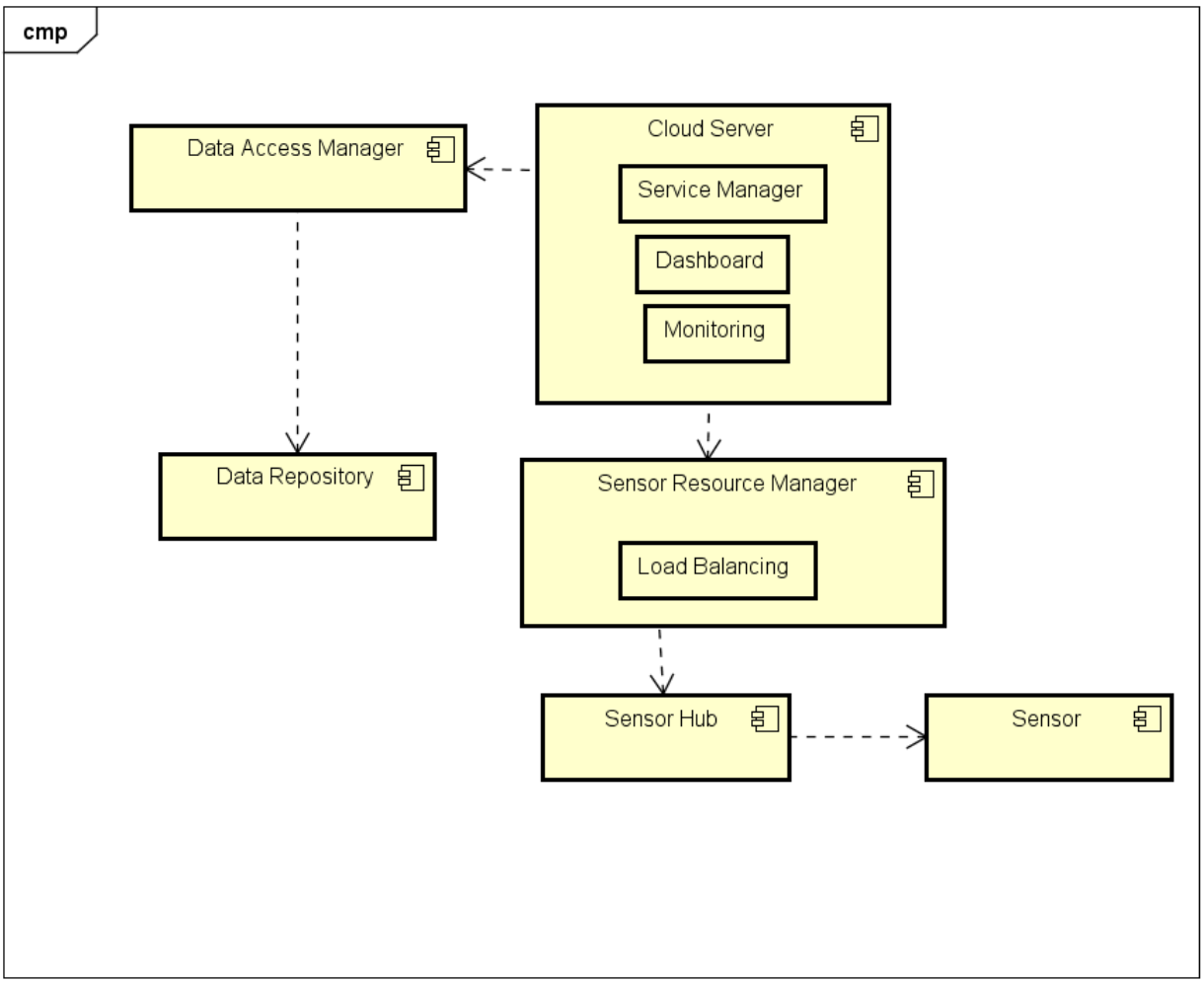
3. Activity Diagram



4. Use Case Diagram



5. Component Diagram



6. Application Architecture

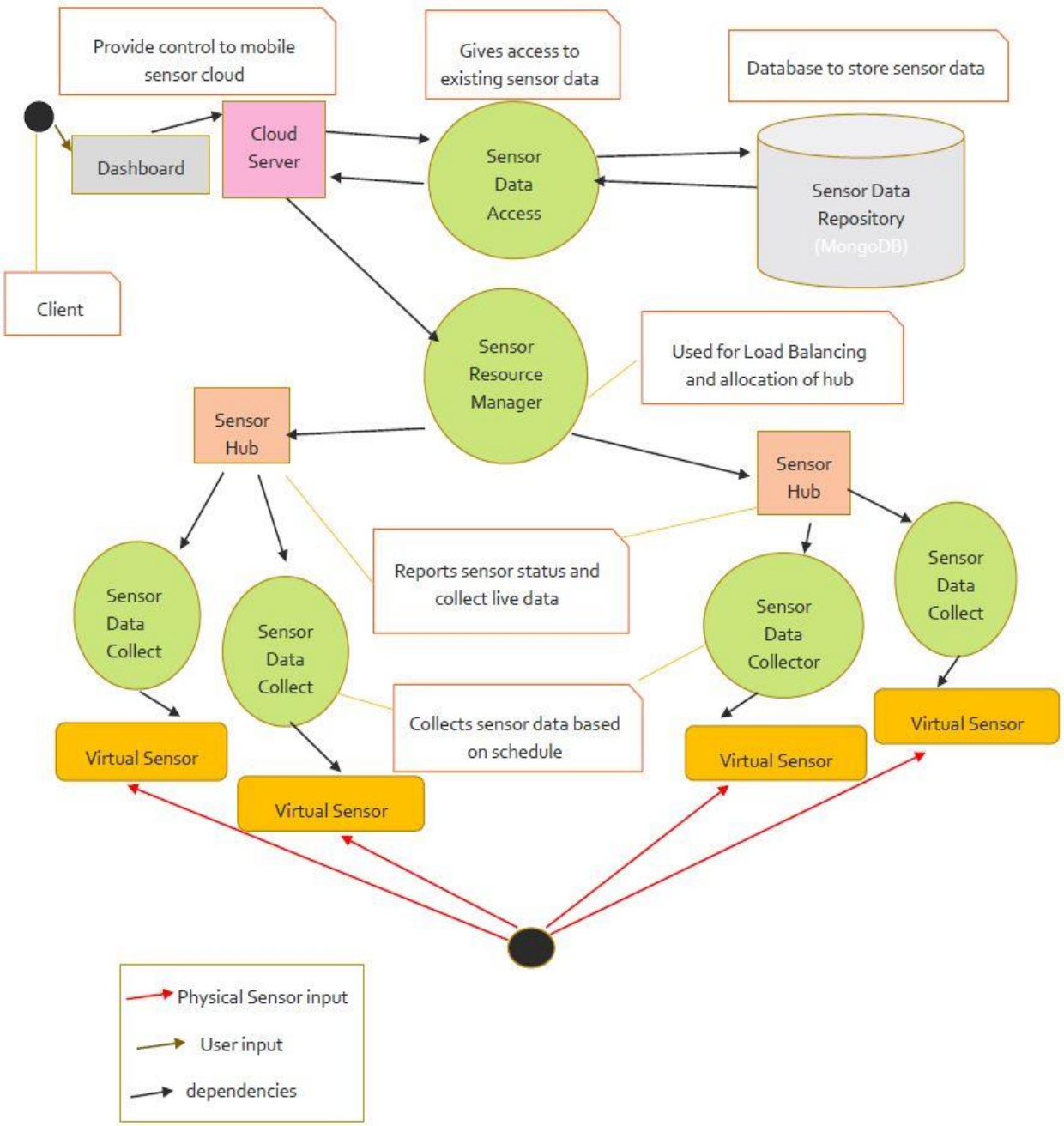


Figure 1 : Application Architecture

7. Load Balancing Algorithm

To manage the load on the server, effective load balancing is needed. The mechanism of dispersing the traffic across multiple hubs is Sensor Cloud load balancing. This mechanism lessens costs and maximizes availability of resources. Through an optimized load balancing, the resources can be better and easily be administered and can be better allocated and the scheduling can also be effective. This will help us handle the resources and will also reduce cost too.

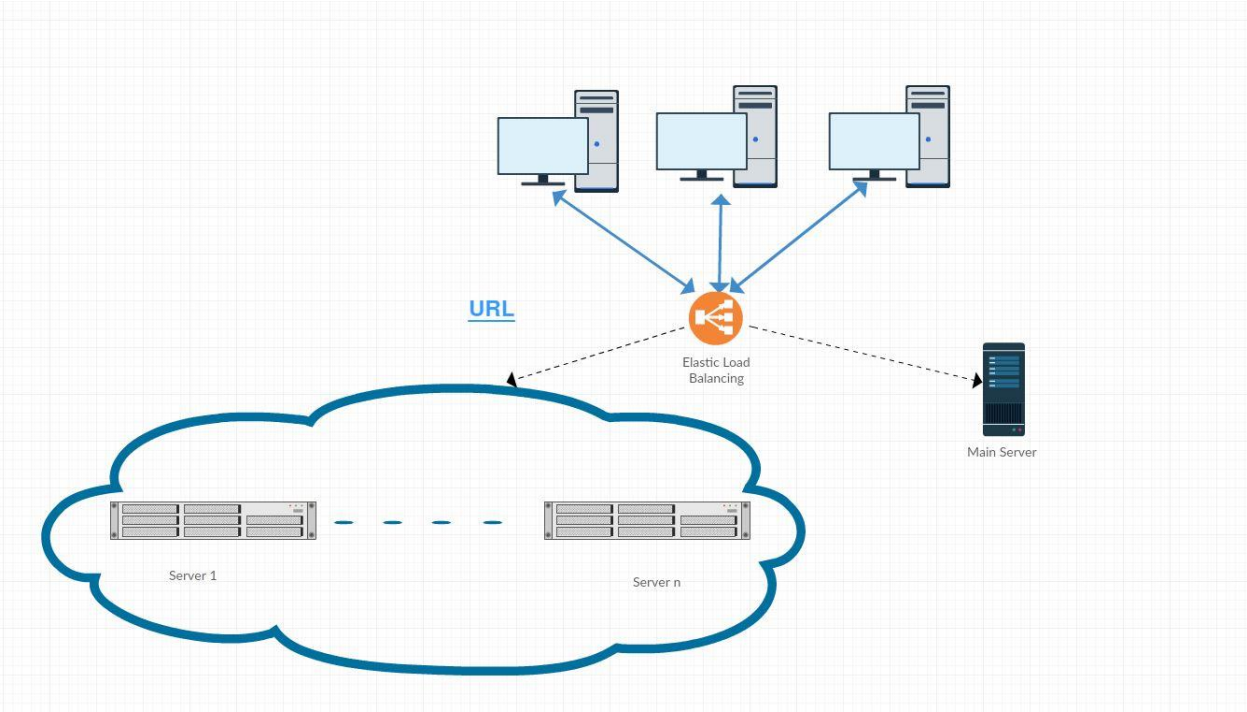


Figure 2 : Load Balancing Algorithm

8. Technology Selection

Figure 3 : Technologies Used

9. Strategy

Figure 3 : Strategy Architecture

9.1 Virtual Sensor:

This simulates a sensor and provides sensing data. This sensor takes the data from actual physical sensor and sends it to the user for the completion of the request

9.2 Billing:

User needs to pay for the sensors that he adds according to area. Utility billing is the universal concept that is used in this project.

9.3 Scalable:

Sensors can be scaled up and down as per the user need. The rescaling done is the product of itself. With this feature we can remarkably diminish the risks correlated with traffic overflow which causes server overflow.

9.4 On Demand:

User can add sensors and monitor them as he wants real-time or hourly. This service enables the provisioning of sensors, on demand whenever they are required instead of waiting for the subs.

10. Implementation Design

10.1 Signup Page

The user has to go to the sign-up page to register the account and enter all the locations. The fields are marked as required for all the mandatory information

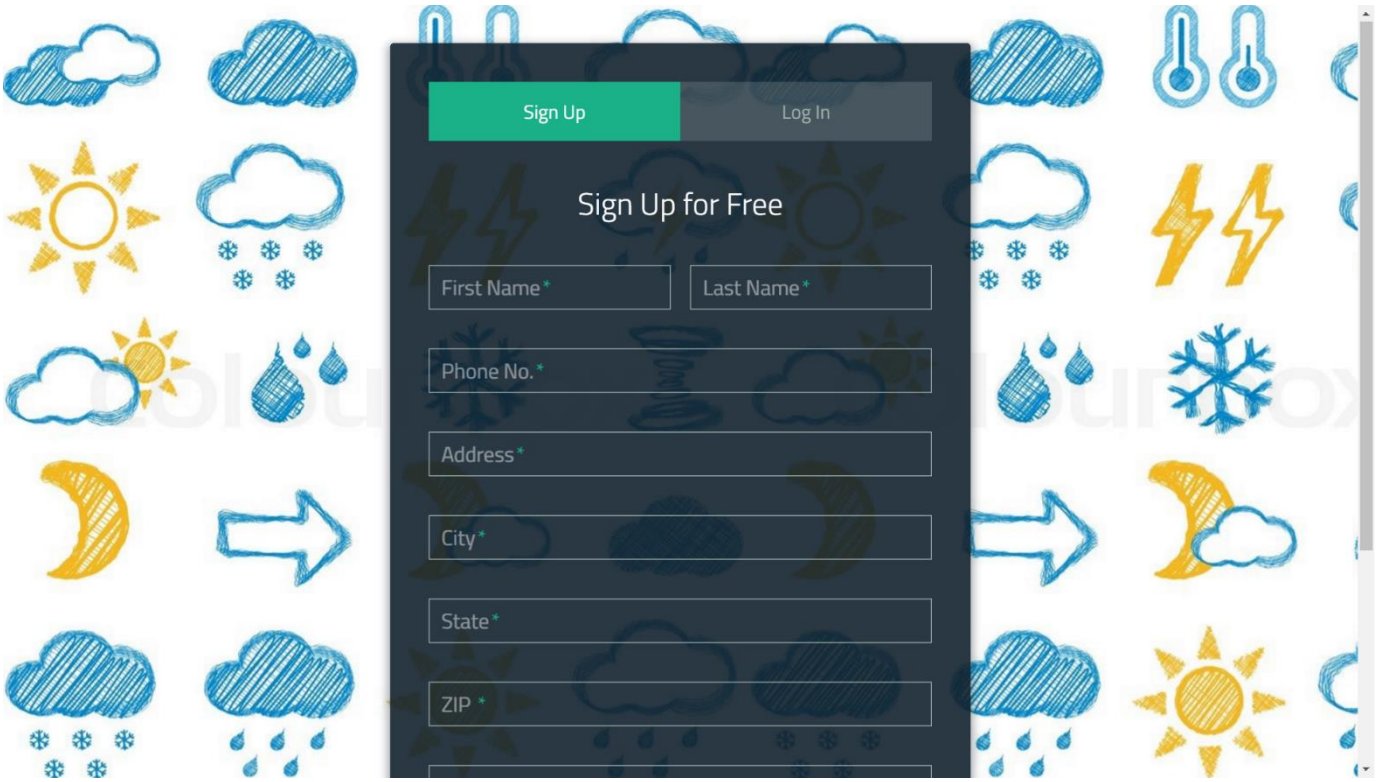


Figure 5 : Signup Page Screenshot

10.2 Login Page

The returning user must provide the email address and the password which was input at the time of signup.

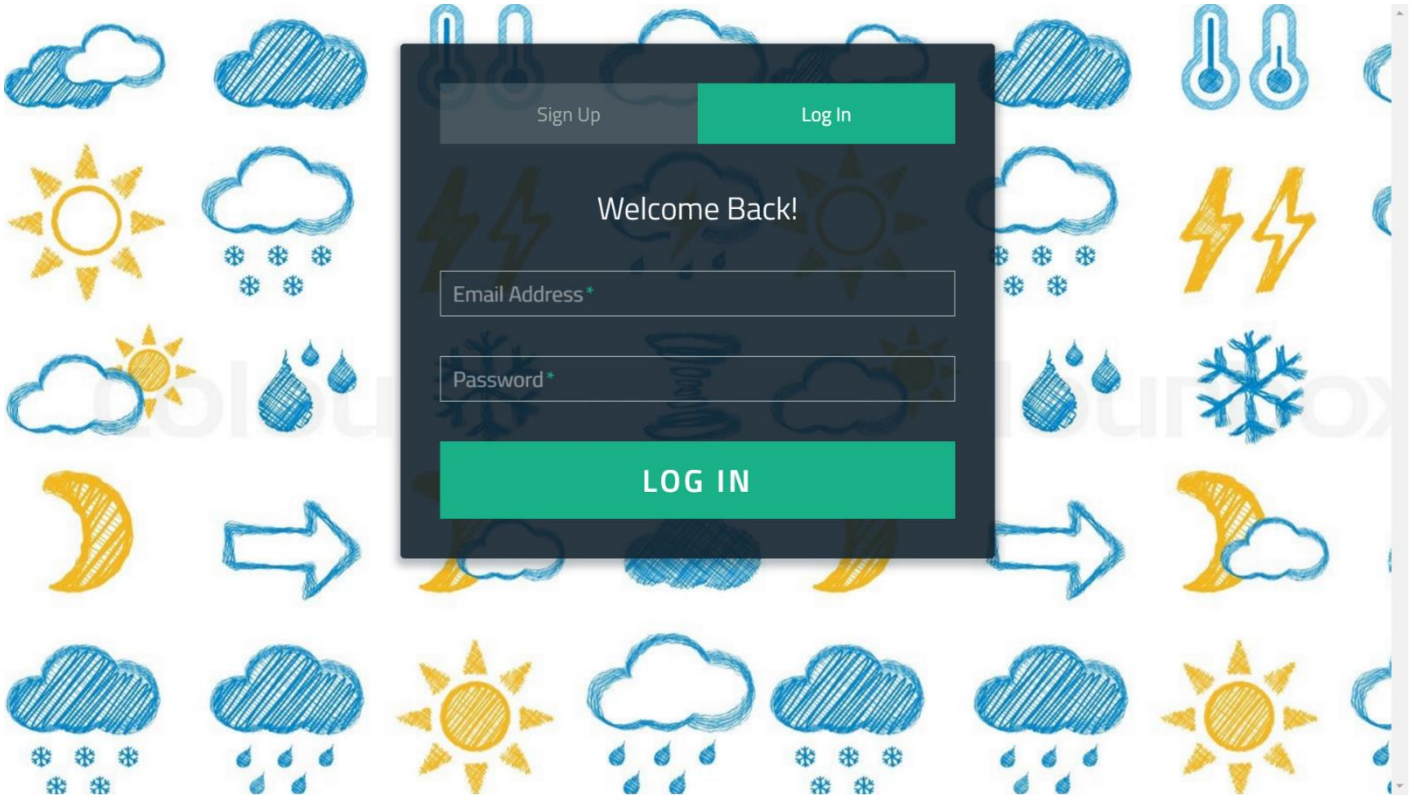


Figure 6 : Login Page Screenshot

10.3 Dashboard

The user has to login first. After the user is authenticated the user can add the sensors that he/she wants to see in his/her dashboard. The sensors he wants can be selected from the dashboard. The user can even view all the sensors that he has added and all that are deactivated and needs to be added.

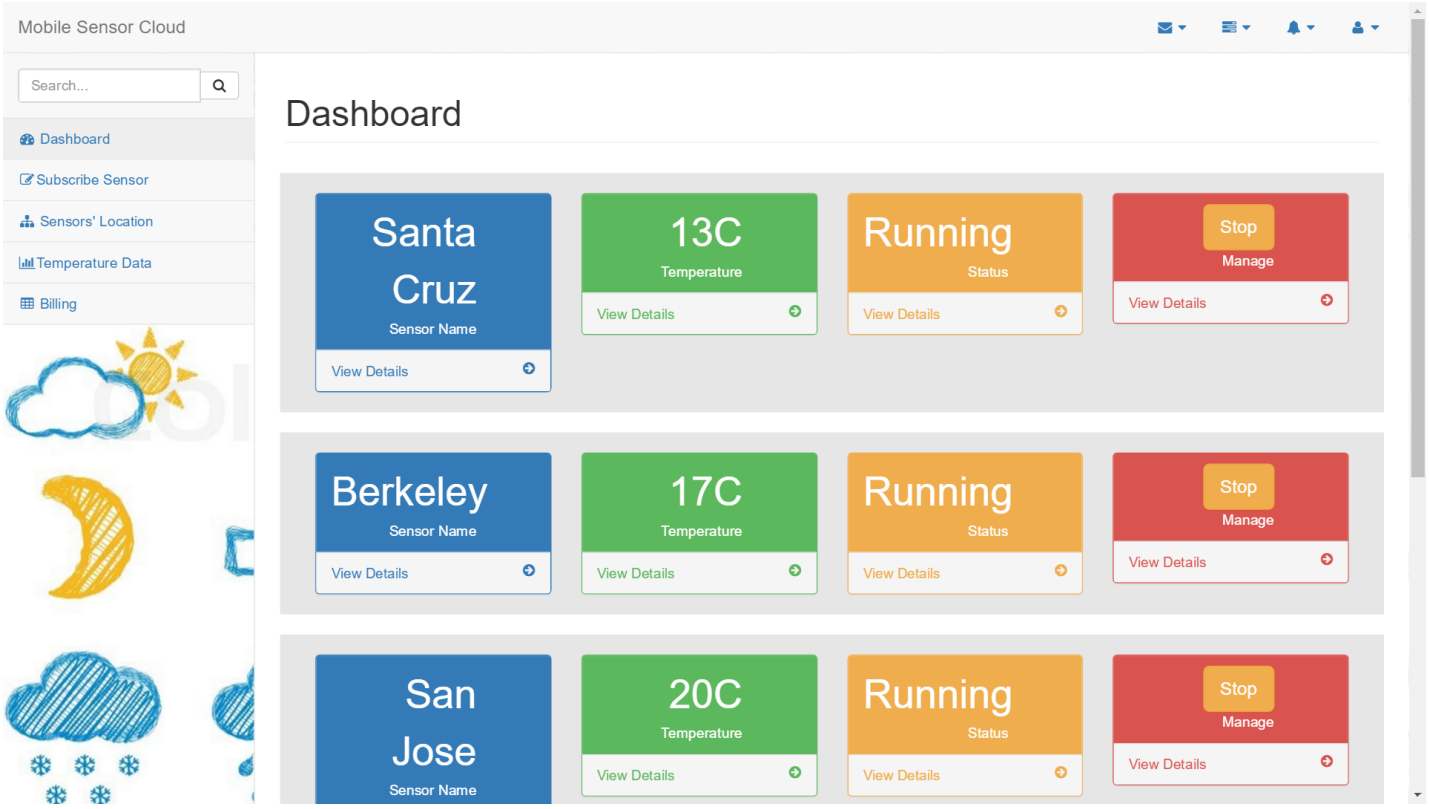


Figure 7 : Dashboard Screenshot

10.4 Temperature Data

User can view temperature graphs in this page. User can group graphs according to area and city also. User can also choose the cities that he wants to be displayed.

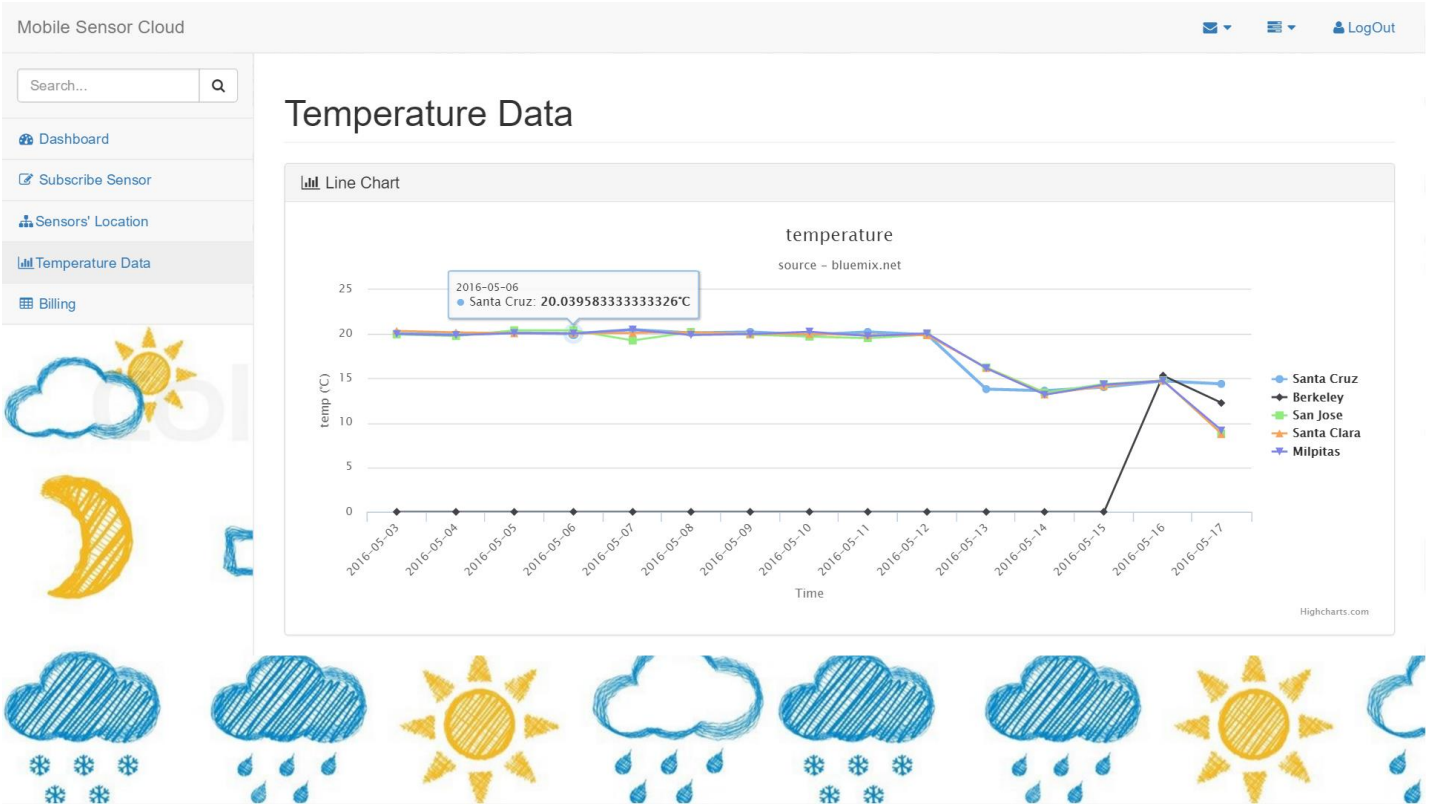


Figure 8 : Graphs Screenshot

10.5 Billing

In this the user can see all the payment data and estimate the cost of the sensor as well as the total cost.

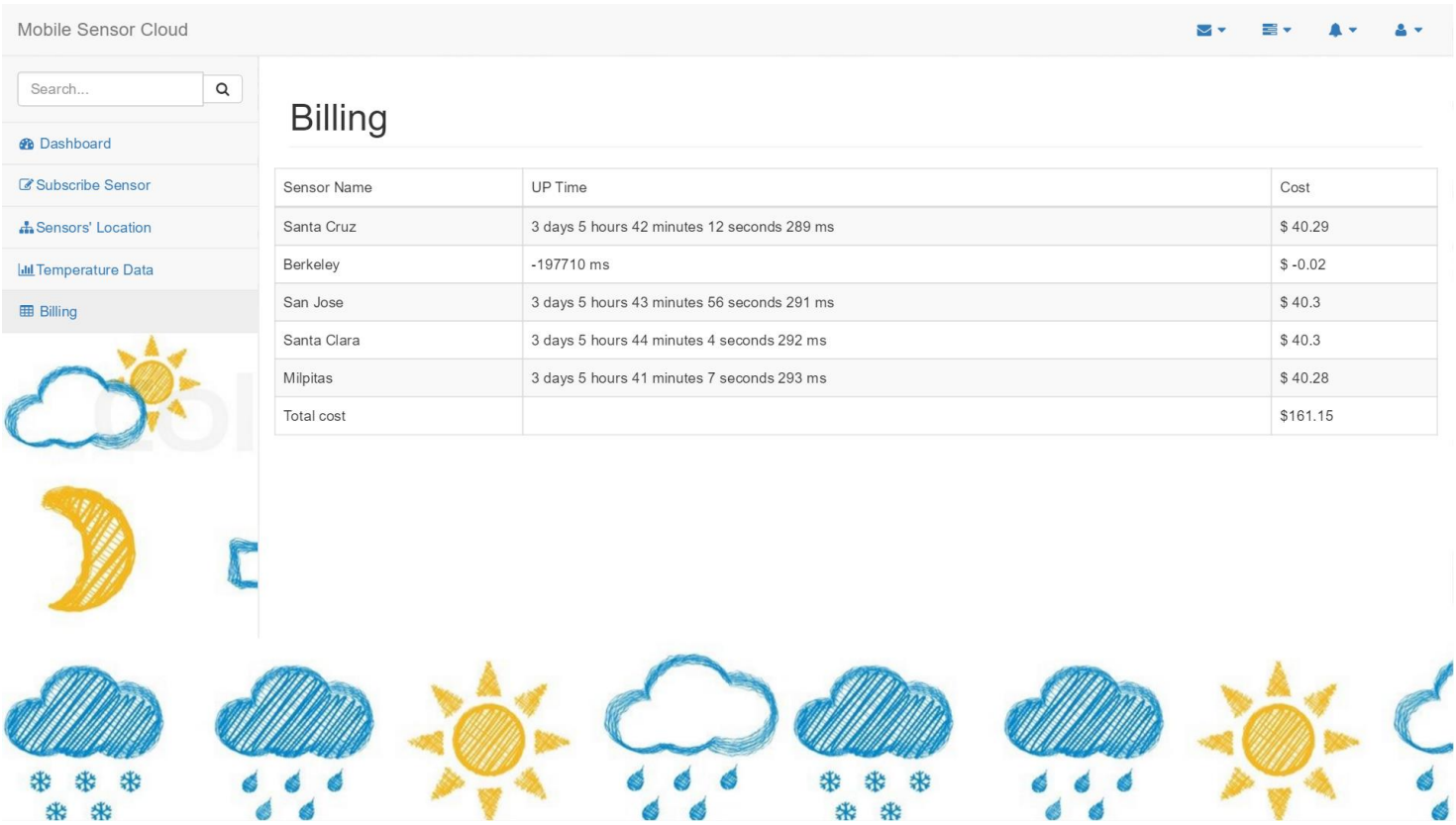


Figure 9 : Tables Screenshot

10.6 Maps

In this the user can see the sensors on the map and can create real time virtual sensor by selecting location on map. User can select the sensor and sensor with the area on maps would be visible.

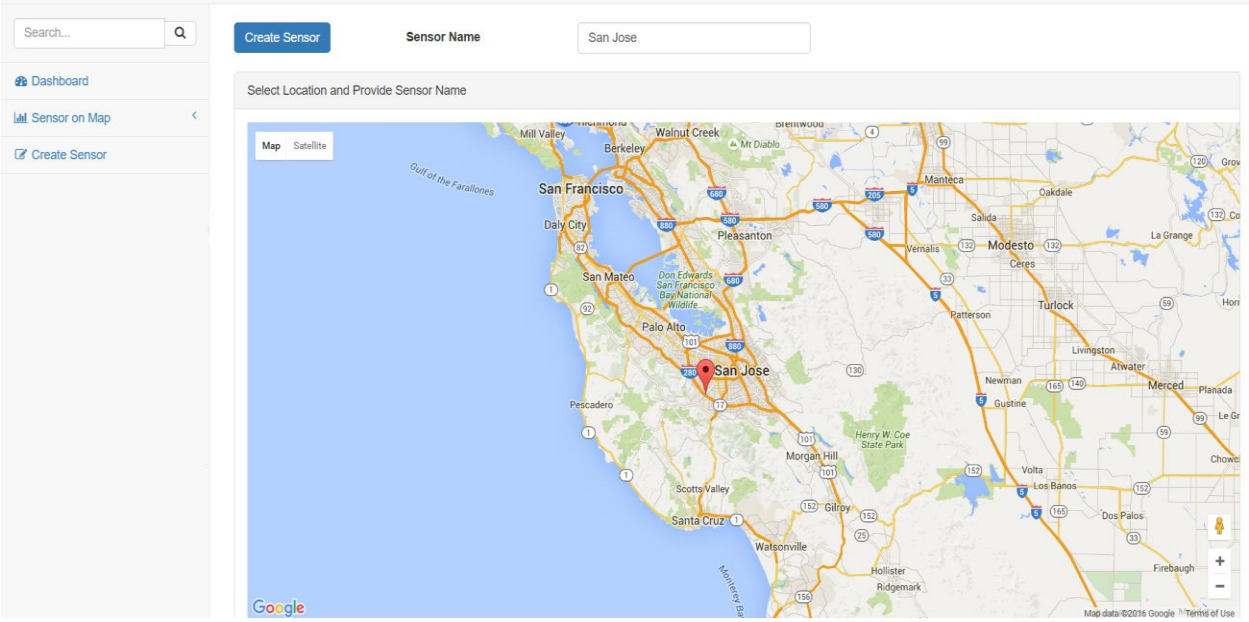


Figure10 : Maps Screenshot

10.7 Billing

User will be charged $1.00/minute and can add as many as he wants and will have to pay per sensor only.

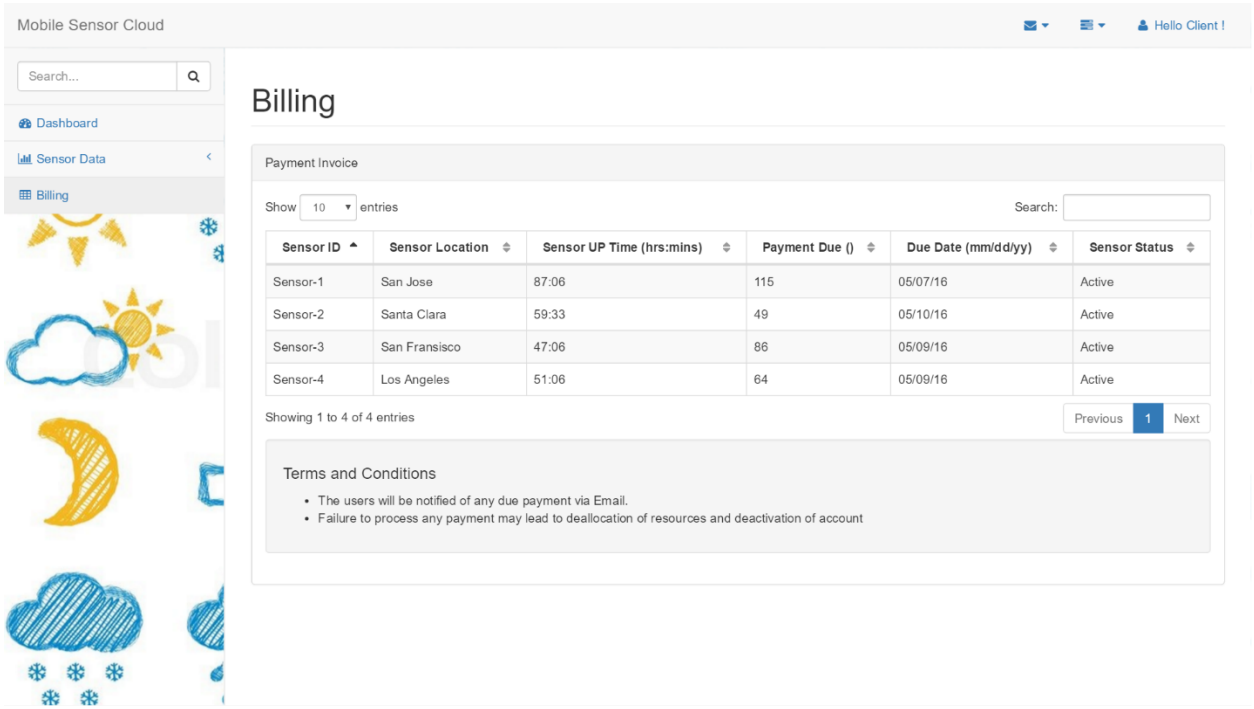


Figure11 : Billing Screenshot

10.8 Admin Panel

The Admin can login using his User Id and Password and can keep a check on all the sensors that are being added to be monitored. The admin can view Sensor Type whether it is temperature, humidity or pressure sensor, Admin can also see the Sensor Area, City, Owner Name and Owner Id. Along with all these Admin can see the health of all the sensors also on his screen.

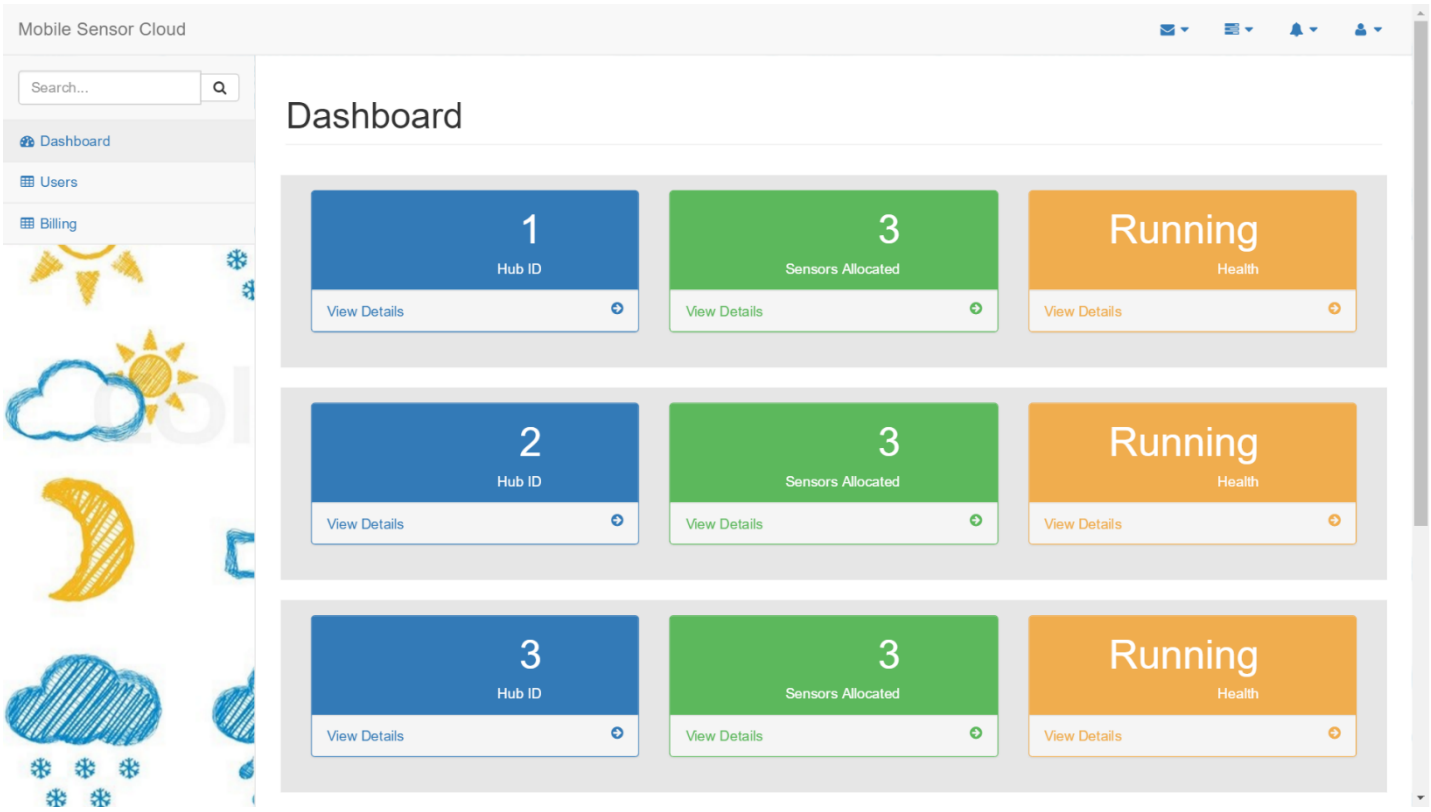


Figure12 : Admin Dashboard Screenshot

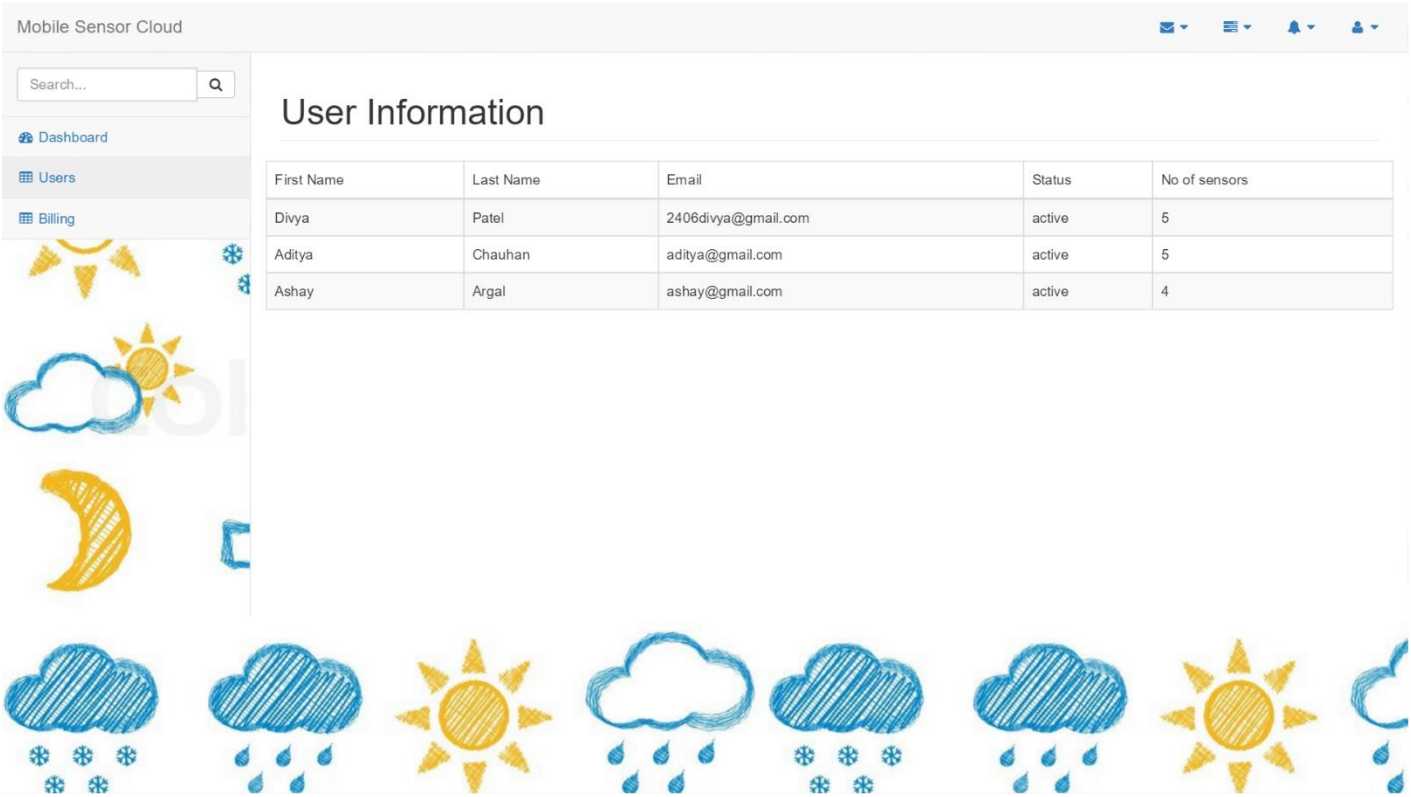
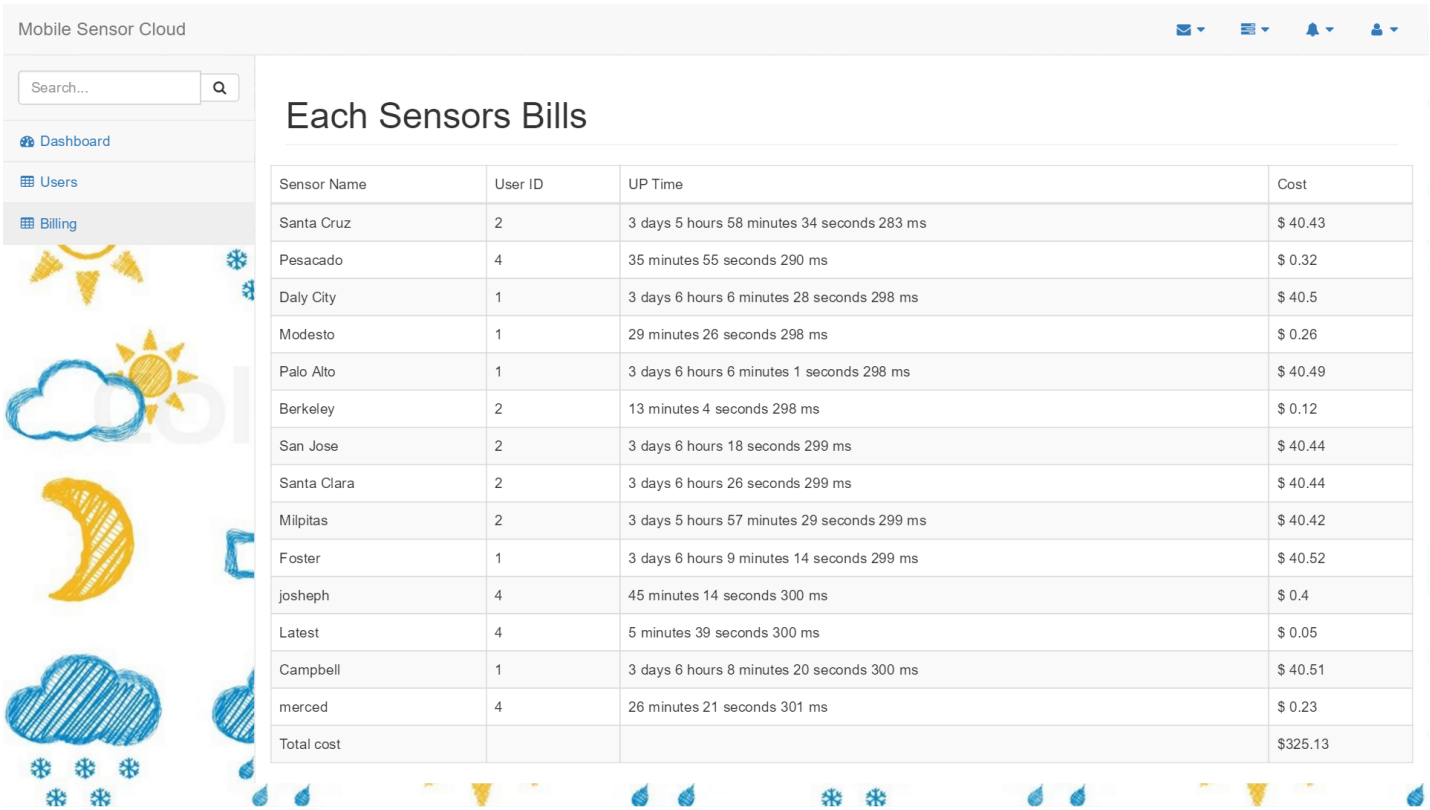


Figure13 : User Information Screenshot



Sensor Name	User ID	UP Time	Cost
Santa Cruz	2	3 days 5 hours 58 minutes 34 seconds 283 ms	$ 40.43
Pesacado	4	35 minutes 55 seconds 290 ms	$ 0.32
Daly City	1	3 days 6 hours 6 minutes 28 seconds 298 ms	$ 40.5
Modesto	1	29 minutes 26 seconds 298 ms	$ 0.26
Palo Alto	1	3 days 6 hours 6 minutes 1 seconds 298 ms	$ 40.49
Berkeley	2	13 minutes 4 seconds 298 ms	$ 0.12
San Jose	2	3 days 6 hours 18 seconds 299 ms	$ 40.44
Santa Clara	2	3 days 6 hours 26 seconds 299 ms	$ 40.44
Milpitas	2	3 days 5 hours 57 minutes 29 seconds 299 ms	$ 40.42
Foster	1	3 days 6 hours 9 minutes 14 seconds 299 ms	$ 40.52
josheph	4	45 minutes 14 seconds 300 ms	$ 0.4
Latest	4	5 minutes 39 seconds 300 ms	$ 0.05
Campbell	1	3 days 6 hours 8 minutes 20 seconds 300 ms	$ 40.51
merced	4	26 minutes 21 seconds 301 ms	$ 0.23
Total cost			$325.13

Figure14 : Sensor billing Screenshot

11. Virtual Sensors

This component works as a simulator of actual physical sensor residing on different location. This virtual sensor gets the data from actual physical sensor in appropriate format as required by the client. This component acts as a physical sensor which will provide all the sensor data, health data and status data and will be able to communicate as well with sensor hub through sensor data hub.

Input = physical sensor data.

Output = sensor data

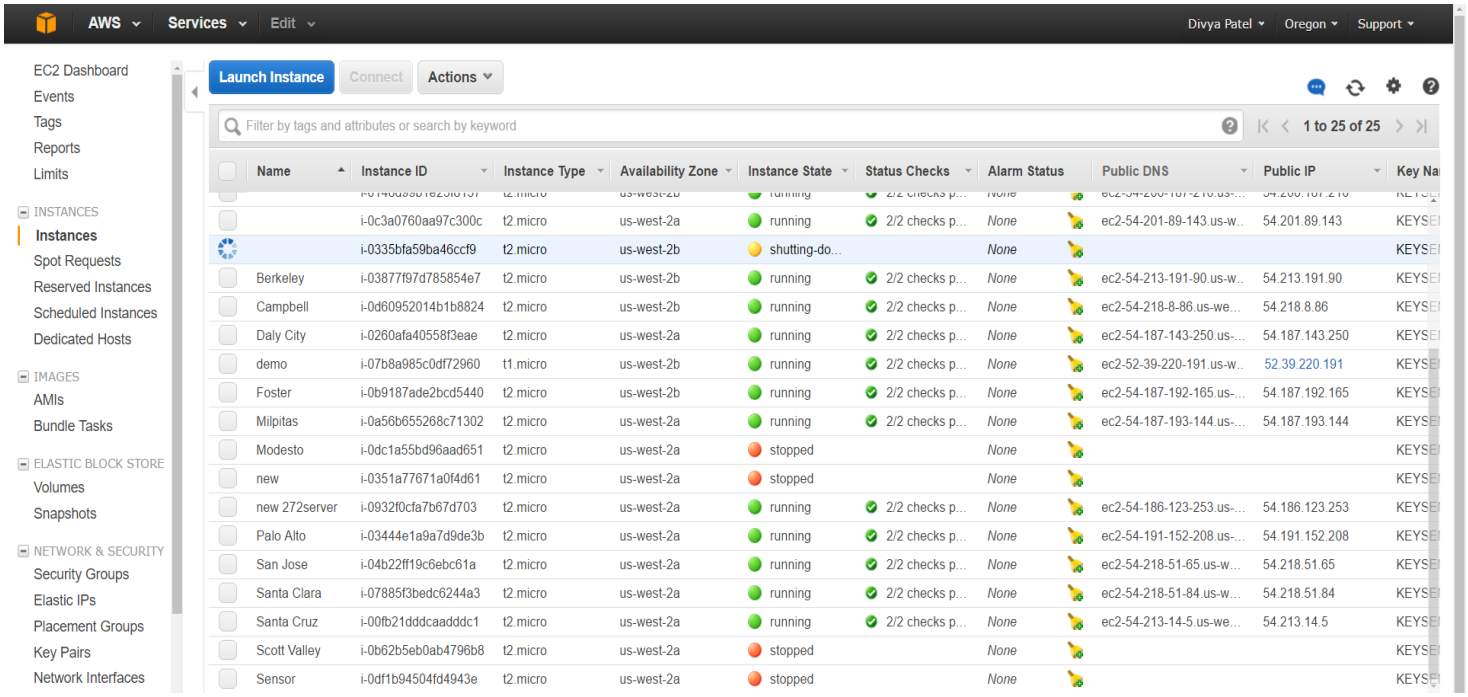


Figure 15 : Virtual Sensors website screenshot

12. Conclusion

User can use this web application to remain updated about weather data of any area across the world, which is done by providing virtual sensors to users which collects sensor data. In addition to that user will be able to see historical data to know the change in temperature, atmospheric pressure and humidity. Users are provided with functionality to add sensor that they want to monitor and remove sensor that they don't want to. Application is highly scalable and follows, pay as you go model.

13. References

- CmpE-281 lecture slides
- https://sjsu.instructure.com/courses/1162965/files/?preview=40037457
- http://en.wikipedia.org/
- https://aws.amazon.com/documentation/ec2/
- Big Data Sensing and Service: A Tutorial