SCALABLE HEALTHCARE SENSOR NETWORK

[1]Sumukh Ballal, [2] Dr. Rupesh Jaiswal
[1]Student, [2]Professor
[1]Department of Electronics and Telecommunication
[1]Pune Institute of Computer Technology, Pune, India

Abstract : The healthcare system is currently experiencing challenges of low quality of service, lack of easy accessibility and high costs of services. We have given a possible solution, to face this challenge. Here, we have implemented a healthcare management system, which is operated wirelessly. This system measures physiological parameters such as ECG, Body Temperature, Pulse Rate using a scalable sensor system, hosted on Cassandra Cluster on AWS. These parameters are relayed to a web application for medical practitioners to use. The display of the sensor data wirelessly reduces wired interventions, paper –based ambiguity and can improve health services provided with a data based approach.

Index Terms **– Sensor, Cassandra, IOT, Data Model, Network, Healthcare**

I. INTRODUCTION

According to the "2012 world health report 'no health without research': the endpoint needs t**o** go beyond publication outputs" as written in [1] , research in health needs to be conducted to better improve healthcare and this needs to go beyond just statistical research, a better end product is required and this needs to happen quick. A combination of software, medical research and medical practitioner input is required to achieve the end product, a cross collaboration among multiple domains of expertise. Our goal was to leverage software in a way, which benefitted both the patient as well as the medical practitioner with ease of usability as well as providing a way for the data of a patient to be available to a focal medical practitioner, so as to provide a better prognosis.

II. LITERATURE REVIEW

The implementation of the scalable sensor network for healthcare was done, keeping in mind scalability of the system to not one, but multiple hospitals across multiple use cases, while keeping speed and achieving ease of usability of the final application in mind. A combination of a robust wireless sensor network along with a scalable data model implemented in Cassandra allowed us to achieve this. Spark was used to achieve analytics and prediction. All of this data was relayed to our web application written in C# and using the ASP .NET framework.

Our research in the storage and retrieval of Sensor data as well as provide scalability pointed us to Cassandra as an IoT sensor data storage point since it can handle up to 2000 sensor's concurrently with the data being clustered with the sensor table as seen in [4]. Sensor data storage was needed, but also required was an extremely fast read time and concurrent overwrite, so as to relay the last know data of the patient from the Cassandra cluster to the web application with the new in-memory option as articulated in [5]. All the sensor data was maintained with security in mind, and we followed the tips as mentioned in [3] and this helped is intended to keep the patient data secure and should be disclosed only once a medical practitioner with authorized access wants to see this data to help with the patient prognosis.

A postgres Database was also needed for our web application and our data model was designed to better approach the problem of consistently changing sensor data values, coming from our Cassandra cluster sensor network storage. In Figure 1, a high level overview is seen with the sensor network relaying data to a sensor aggregator node which in turn relays it to a system like a Adruino, which sends it to the Cassandra cluster for further storage.

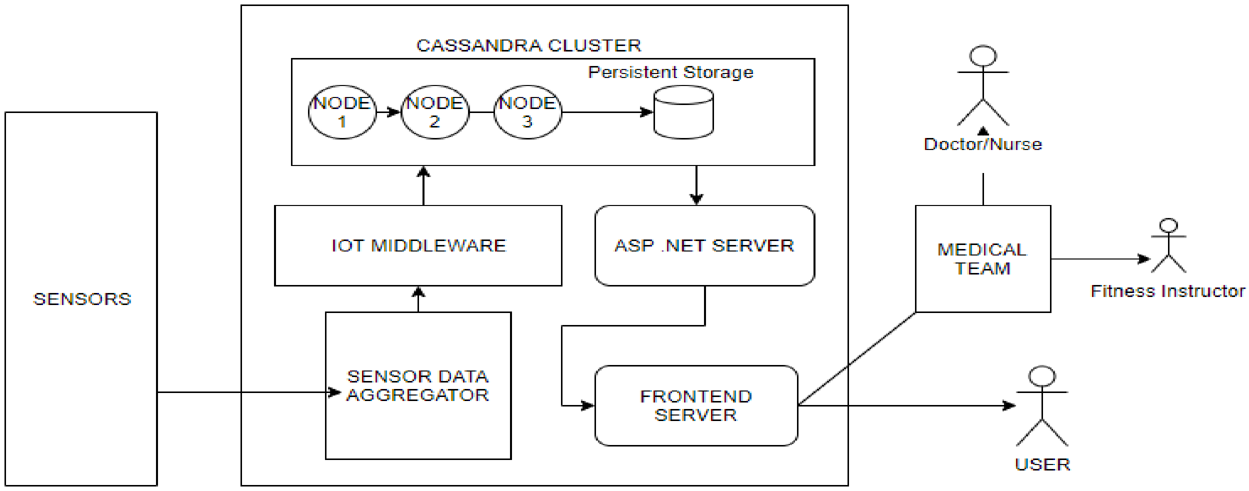

Figure 1: Architecture Diagram High Level Overview

III. PROPOSED WORK

For our testing and the lack of a free open source Simulation for a large scale sensor cluster a combination of 10 sensors which included Temperature sensors, ECG sensors and Pulse Sensor, each with its own UUID was used. These sensors were attached to a adruino to send the data serially with baud 9600. The data model was created, to include these sensor UUID's and use Clustered grouping to group the data as per the sensor, in our case 10 * 3 = 30 sensor groups, each group consisting of data ranging from [temp, pulse, ecg [timeseries]] raw data. This data was then manipulated on retrieval from our ASP .NET code and displayed on the website.

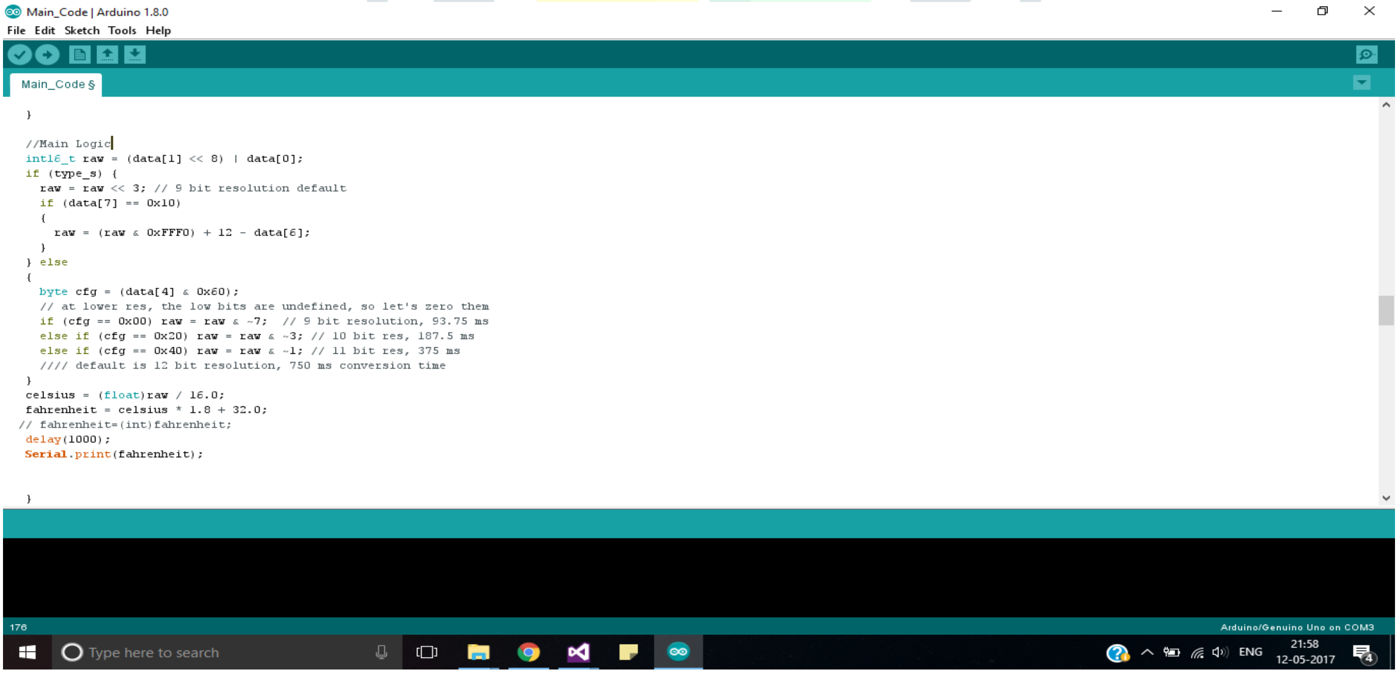

Figure 2: Temperature Sensor Code

The temperature sensor, records data serially which was then altered by our code and the raw hexadecimal values was changed to a decimal number which got the Celsius degree as seen in Figure 2, this was done by our adruino processor and for a simpler data credibility.

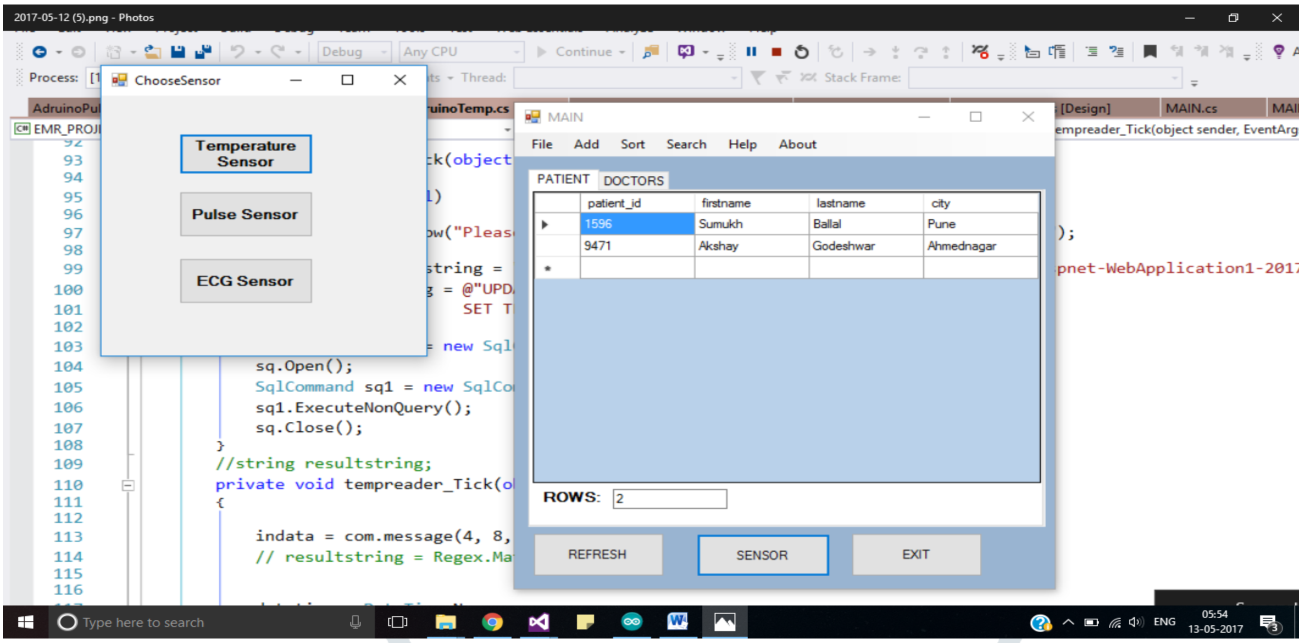

Figure 3: Sensor UI page

Once the sensor is hooked on to the patient, and the sensor name or UUID is provided, it is linked to the patient ID and the sensor data is then stored in our Cassandra data model, concurrently as seen in Figure 3, our postgres Database for our web application is updated with general patient information which is updated with a proprietary C# desktop application.

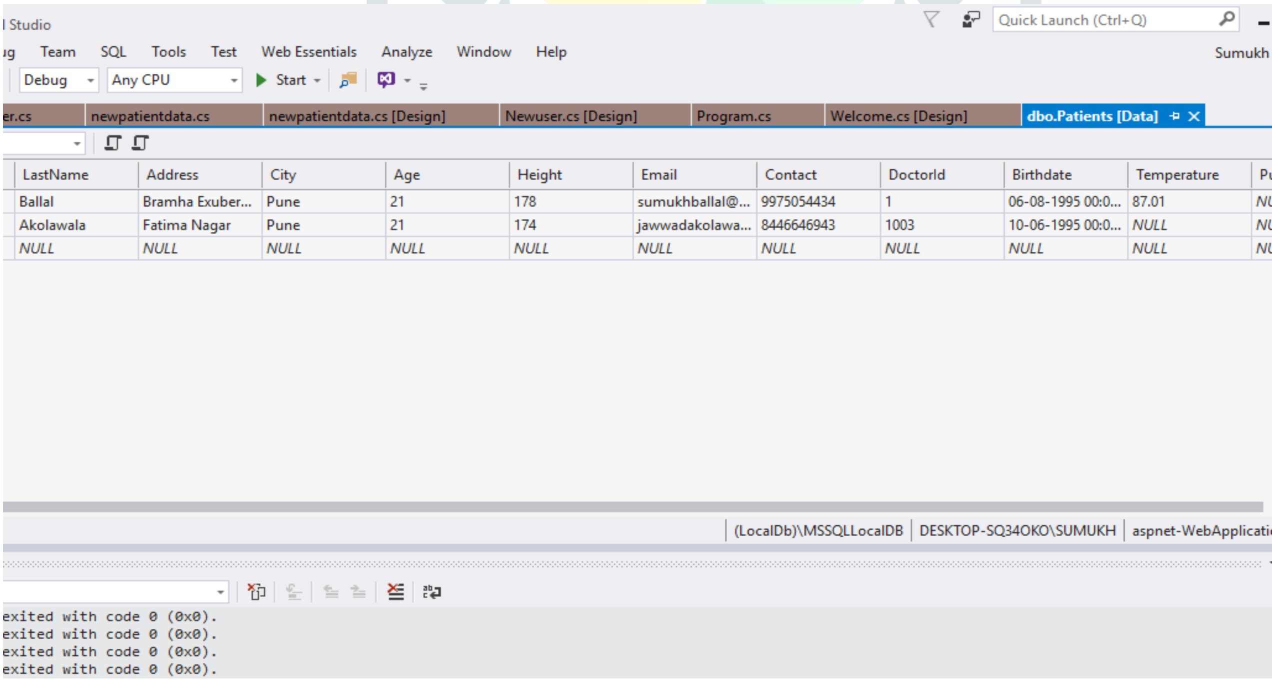

Figure 4: Database Patient System

Our application code, fetches the latest sensor value, for the required sensor, then relays that to the postgres database which is then used to display this on the web application. As seen in Figure 4, our data model was defined to include the last sensor value. This was done to prevent any null values when Cassandra provides us data, due to the consistently updating sensor values, and to provide a safe mechanism for our webpage to prevent consistent polling calls to the Cassandra cluster from our web application. This was a compromise in our system done solely to provide a smooth end user experience.

IV. RESULTS

From the results it can be seen, that the data from the sensors are stored in the Cassandra Database and it can be used by our application written in C# and supported by the ASP .NET framework as well as systems like Spark which can be used to further allow for analysis of that data. This data is displayed on our UI using our web application hosted on an IIS server.

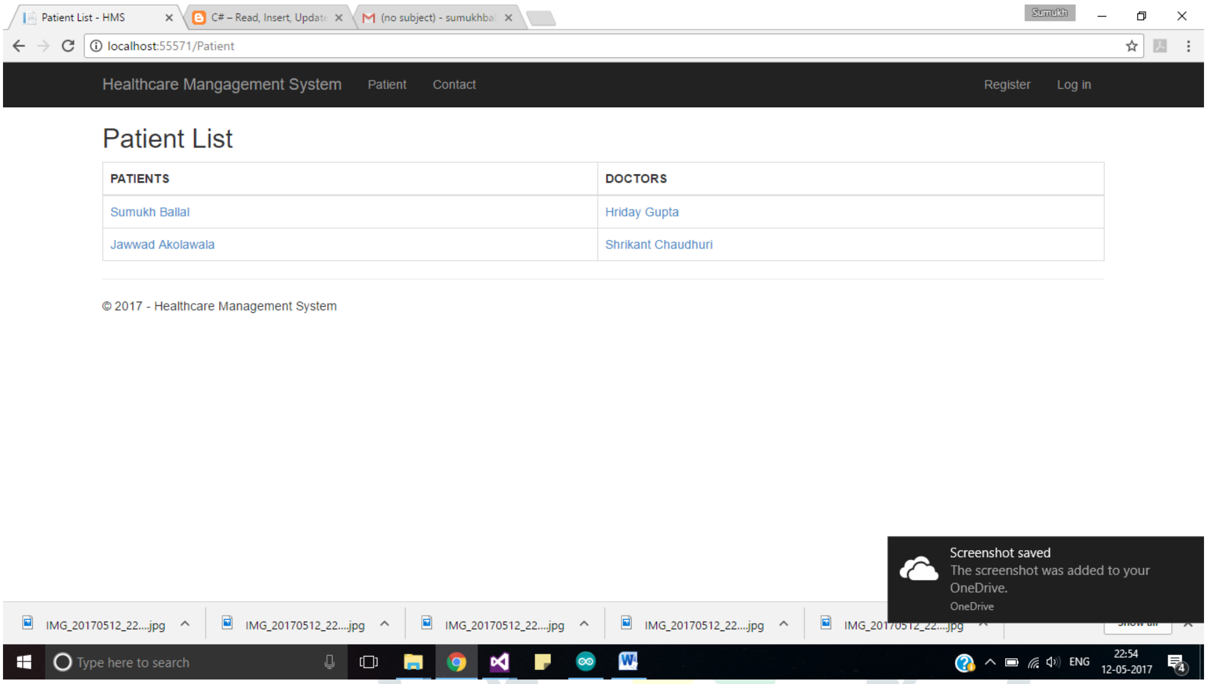

Figure 5: Web Application UI

In Figure 5, we see the test data created for the patient. When the medical practitioner clicks this patient, from the unique UUID, all information stored for that patient, referenced in our Cassandra database as well as our Application Database is displayed on the user interface.

V. CONCLUSION

The Healthcare Management System is responsible for monitoring the health of many numbers of patients in the same critical unit. The system proposed above is the cheapest system that can be used for this purpose. One of the important beneficiaries of the system is the doctors who can monitor the physical and medical conditions of their patients from any part of the world and thus he can give instructions to others to attend to the patient. Thus the proposed system is in other words an e-hospital system, where the doctor can attend to many patients at a time .Since it is a continuous monitoring system, the doctor would be able to give clear and best instructions within the least time.

VI. ACKNOWLEDGEMENT

I would like to express my gratitude towards my mentor, Dr. R. C. Jaiswal for guiding me through the research. His contribution in the providing the research guidance required for this was immense and his insight in the topic gave the paper what was needed.

VII. REFERENCES

[1] Zachariah R1, Reid T, Ford N, Van den Bergh R, Dahmane A, Khogali M, Delaunois P, Harries AD, “The 2012 world health report 'no health without research': the endpoint needs to go beyond publication outputs.”

[2] B.L. Titzer, D.K. Lee, J. Palsberg, "Avrora: scalable sensor network simulation with precise timing", IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.

[3] Qian Wang, Kui Ren, Shucheng Yu, Wenjing Lou, "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance", ACM Transactions on Sensor Networks (TOSN) TOSN Homepage archive, Volume 8 Issue 1, August 2011

[4] Lee Carman Ka Man1, Cheng Mei Na1 and Ng Chun Kit1 “IoT-based Asset Management System for Healthcare-related Industries”, International Journal of Engineering Business Management, 2015, 7:19

[5] DataStax Enterprise 4.0 In-Memory Option, White Paper, 2014-02

[6] Castellani, A., Dissegna, M., Bui, N. & Zorzi, M. (2012), “WebIoT: A web application framework for the internet of things”. In: Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE, pp. 202-207.

[7] Eben Hewitt and Jeff Carpenter, "Cassandra: The Definitive Guide: Distributed Data at Web Scale", O' Reilly 2nd Edition

[8] C. Y. KAN, "Cassandra Data Modeling and Analysis", Packt Publishing Ltd, December 2014 Edition

[9] Matthew MacDonald, "ASP.NET: The Complete Reference", McGraw-Hill/Osborne, 2002

[10] Ted Hills, "NoSQL and SQL Data Modeling: Bringing Together Data, Semantics, and Software", Technics Publication, 2016

[11] Benny P.L. Lo, Surapa Thiemjarus, Rachel King, Guang-Zhong Yang, "Body Sensor Network – a wireless sensor platform for pervasive healthcare monitoring", csis.pace.edu – 2005

[12] Raymond R.Bond, Dewar D.Finlay, Chris D.Nugent, George Moore, "A review of ECG storage formats", International Journal of Medical Informatics Volume 80, Issue 10, October 2011

[13] Pardeep Kumar, Hoon-Jae Lee, "Security Issues in Healthcare Applications Using Wireless Medical Sensor Networks: A Survey", Sensors 2012, 12, 55-91

[14] Jan Sipke van der Veen, Bram van der Waaij, Robert J. Meijer, "Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual", 2012 IEEE Fifth International Conference on Cloud Computing

[15] Yong-Shin Kang, Il-Ha Park, Jongtae Rhee, Yong-Han Lee, "MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data", IEEE Sensors Journal, Volume: 16 , Issue: 2 ,Jan.15, 2016