**SMART HOSPITAL SYSTEM**

PROJECT REPORT

Submitted for CAL in TARP

By

Kartik Sharma 16BCE1124

Sai Anil 16BCE1318

P. Akash 16BCE1373

Sai Susanth 16BCE1047

Vivek Reddy 16BCE1306

Sai Phanindra 16BCE1075

Madhusudhan Reddy 16BCE1316

Sreekanth Reddy 16BCE1362

Slot: TCC1

Name of Faculty: DR.M. Poongodi



(MARCH, 2019)

# CERTIFICATE

This is to certify that the Project work entitled “SMART HOSPITAL SYSTEM” that is being submitted by “POLAMREDDY VIVEK REDDY,

TEKURI SAI ANIL, CHOUTIPALLE SREEKANTH REDDY, SHAMBHAVI PARASHAR” for

CAL in B.Tech Large Scale Data Processing(CSE3025) is a record of bonafide work done under my supervision. The contents of this Project work have not been submitted for any other CAL course.

Place: Chennai

Date: 3rd April 2019

# ACKNOWLEDGEMENTS

Firstly, we would like to thank DR M. POONGODI, TARP (CSE3229) Winter Semester 2018-2019 for her guidance and her resources which made this PBL project possible. We would also like to thank our lab assistants for helping us whenever required. Finally, we would acknowledge our deemed to be university, Vellore Institute of Technology, Chennai Campus for providing us with the opportunity and facilities which ensured the project’s completion.

Kartik Sharma 16BCE1124

Sai Anil 16BCE1318

P. Akash 16BCE1373

Sai Susanth 16BCE1047

Vivek Reddy 16BCE1306

Sai Phanindra 16BCE1075

Madhusudhan Reddy 16BCE1316

Sreekanth Reddy 16BCE1362

# ABSTRACT

The main idea behind the Smart Hospital System is that it can help in automating the visitor's flow and help doctors take care of patients who are in an emergency or need. We have demonstrated Pneumonia detection with implementing Deep learning (ResNet algorithm) and also successfully demonstrated how can patient 's vitals be analysed and examined by Doctor 24\*7 round the clock. We have trained Tensor Flow serving model and implemented our image dataset on it. We have successfully implemented and demonstrated storage solution (CEPH Storage). We have used Kibana and Elastic Search for log reports generation. Using NodeJS we have created a dashboard. We have integrated all the above into a single working solution.

# INTRODUCTION

As per information provided by the Medical Council of India, there were a total 10,22,859 allopathic doctors registered with the state medical councils or Medical Council of India as on March 31 2017

There is a great need for making doctors availability for the patients needed whenever it is considered an emergency. We had brainstormed and moulded our thought process as to make the hospital system automated and we have demonstrated the same.

This project will find its relevance in creating sustainable eco-spaces in automation of Hospital management. The health sector is very important when a developing nation in considered. With adopting and implementing technologies that can bring the change we can expect development in leaps and bounds.

Objectives:

1. To make automation at ease to hospitals

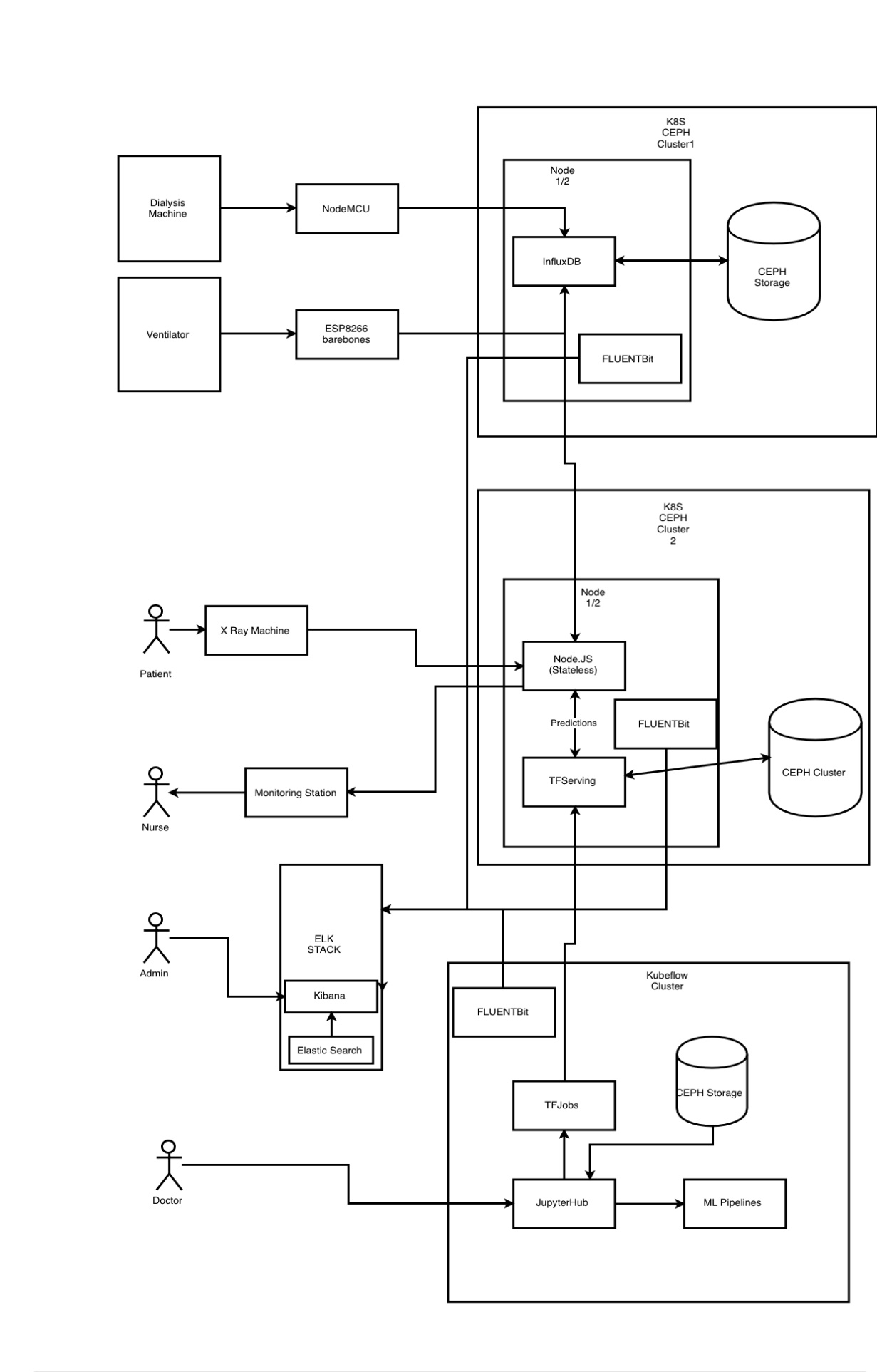
2. To make doctor availability to those in emergency need

3. To build a complete prototype that can be implemented.

4. To make understand people with little technical knowledge also entire operations in the project

# METHODOLOGY

Patient --> For patient the X-ray is taken and directly the results are predicted using the tuned Pneumonia model though the Nods.JS Webserver. The data format is DICOM which is standard for medical imaging files. The model is served using TF-Serving (Pneumonia RESNET 80 layers)  
Nurse --> Rather than going to each ventilator and dialysis machine, the nurse monitors the patient vitals using the dashboard which also shows any warnings like very high heartrate etc.  
Doctor --> Doctor can write ML code easily at scale, for this he will access the Jupyter Hub and tell the hardware details like CPU GPU Memory he need, then a notebook is provisioned to him to write the code with the storage attached.  
Admin --> The admin monitors the various cluster logs through the ELK Stack, basically using the Kibana visualization for the finding and any error or problems in the cluster as failure of any component is tough to track in distributed environment.  
The Dialysis machine is connected to the ESP8266 barebones for minimal hardware design and hence a cost efficient factor.  
The ventilator is connected to NodeMCU Development board which is basically for prototyping purposes.  
  
The standard communication port used if RS232.  
The storage is provisioned using CEPH RADOS Gateway, with a standard replication factor of 3x.  
The Wireless Medium used for communication is 802.11n  
The Protocols used are --> HTTP, REST, TCP, SSH



**NODEMCU**

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language.

**ESP8266**

The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by manufacturer Espressif Systems in Shanghai, China. The chip first came to the attention of western makers in August 2014 with the ESP-01 module, made by a third-party manufacturer Ai-Thinker.

**GRAFANA**

* Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored. Create, explore, and share dashboards with your team and foster a data driven culture.
* From heatmaps to histograms. Graphs to geomaps. Grafana has a plethora of visualization options to help you understand your data, beautifully.
* Seamlessly define alerts where it makes sense — while you’re in the data. Define thresholds visually, and get notified via Slack, PagerDuty, and more.
* Bring your data together to get better context. Grafana supports dozens of databases, natively. Mix them together in the same Dashboard.

**ELASTIC SEARCH**

* Elasticsearch is an open-source, distributed search and analytics engine built on Apache Lucene. Elasticsearch is a free, open source software.
* Elasticsearch is the most popular search engine, and is commonly used for log analytics, full-text search, security intelligence, business analytics and other applications.
* We can search and retrieve the document using the Elasticsearch API.

**PUTTY**

* PuTTY is a free and open-source terminal emulator, serial console and network file transfer application.
* It is used to connect Linux server from windows machine.
* With the help of IP address, username and password, we can access the shell of a server.

**KIBANA**

* It is an open-source visualization tool.
* Kibana lets you visualize your Elasticsearch data.
* Kibana gives you the freedom to select the way you give shape to your data.
* Some examples are bar graphs, pi charts, heatmaps.
* We can also perform advanced time series analysis on the Elastic search data with time series UIs.
* Some use cases that can be analyzed are logging, metrics, security analysis, site visits and searches etc.

# REFERENCES