International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 6, Issue 9, ISSN No. 2455-2143, Pages 296-301
Published Online January 2022 in IJEAST (http://www.ijeast.com)

IOT BASED PATIENT MANAGEMENT SYSYTEM

Bibikhuteja Pattewale, Mr.Mujamil Dakhani
Department of CNE
SECAB Institute of Engineering and Technology,
Vijaypur, Karnataka, India

Abstract— **Identification of patients while on medication plays a crucial role. It's observed on many instances that a mistaken identity leads to severe consequences causing irreversible losses in terms of health and resources. With the advent of contagious ailments such asCoVid-19 this challenge appears to be more intriguing. We have seen that many people were deprived of knowing correct status of health of their dear ones due to huge confusion in patient management system at our health centers. This project tries to address this anomaly by automating the patient identification and management by employing techniques of contactless tags worn around wrists of patients removing human errors that may be caused by wrong entries in existing patient management systems. It also provides timely health status to the attendant.**

Keywords—**IoT, Patient health management system, RFID.**

I. INTRODUCTION

Patient Management has been mostly manual even in the advent of technology. These systems are usually depends on manually entered or written data by staff on a system or patient records. This is error prone as a wrong entry leads to misidentification and may adversely affect the diagnosis and treatment.

During the pandemic we have witnessed trails of agonizing stories where misidentification of the patient driven the near and dear one's of the patients into a spiral of anxiety and fear.

Thus, it inspired the author to come up with a viable, robust and economical solution to this problem.

II. OBJECTIVE/MOTIVATION

The main objective of this work is to provide a viable, economic and robust framework to ensure that correct patient information is recorded, processed and acted upon. This work thrives to achieve the said objective by employing various technologies varying across the fields of IoT, database management and front end web server design.

Problem Identification:–

Most of the hospitals use a standalone data processing package or manual record keeping for storing patient data. Such system is error prone, difficult to manage and become hurdle in escalating the patient information to referred specialists.

Thus by employing the framework discussed in this paper, we could achieve a better patient management with least scope for human error and easy management of the patient data.

III. PROPOSED SYSTEM

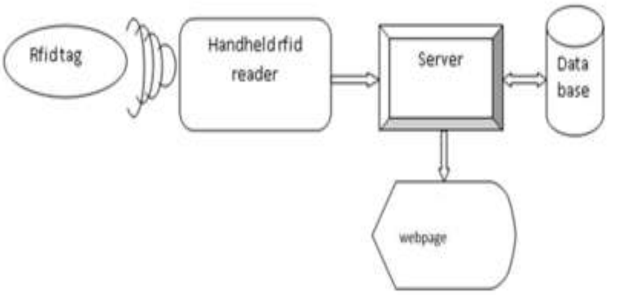


Fig 2.1: Proposed system architecture

The above system diagram represents the proposed system. Patients would wear a RFID wrist band; this work proposes to device a hand held r fid reader which would pass the information to the server which would allow the hospital staff to perform various operations such as patient registration, updating vital information, noting measurements, prescription of medicines, uploading lab reports, noting of case history etc.

The details are updated in a patient data base securely and can be retrieved as and when required. All the information would be available with a simple scan of RFID tag of patient with handheld device. The health updates would be set to a registered attendant through SMS/ E-mail. Case history can be easily when patient is referred to a specialist.

Web page would display the summary of case history with options to dig deeper into the details of patients such as test reports, medication admin is tered etc. This would be protected to ensure privacy of the patient, only an authorized user with permission can retrieve this information with his hand held device.

IV. SYSTEM REQUIREMENTS

The system requirements for the proposed framework can be classified in two categories hardware and software.
Hardware requirements dictate the sensor nodes to be used for various vital data collection, bare metal programming for reading of the identification tags and so on. Where, software requirements dictate the framework to process the data captured by hardware sensors and to provide a user interface for data interpretation.

i. Hardware Requirements

ESP32

The ESP32 is an adaptable System on a Chip (SoC) that can also be used as an all around valuable microcontroller with a huge wide plan of peripherals including WiFi and Bluetooth far off limits.
It is produced by Shanghai-based Espress if Systems, and expenses under $5. Albeit the ESP32 is a SoC, most clients won't begin by utilizing only the ESP32 chip itself.
One significant advantage to using this module instead of preparation without any planning is that it has adequately pre-stacked the low-level of device drivers, the far off show stacks for the Wi-Fi Bluetooth and BLE, and Free RTOS as the base OS.

Another major component module installed as ESP32 is what is more commonly referred to as the ESP32 Module Development. This is basically a board-mounted ESP32 module with additional equipment such as a voltage regulator and a consecutive USB IC.

It allows the direct relationship with work region PC that would have the option which is to be used to total, download, and runthe programs clearly on this module.

RFID Tags

A RFID name works by conveying and getting information through a receiving wire and a CPU — moreover to a great extent called an organized circuit or IC. The CPU on a RFID pursuer is created with whatever information the customer needs.
There are two main types of RFID labels: battery operated and latent. As the name suggests, RFID battery-powered tags contain a locally accessible battery as a power supply, but the RFID tag not installed does not work, claiming to operate using the power supply sent to the RFID fan. Battery-related RFID tags can be called RFID dynamic names.

Fig 4.1: RFID passive tags

Latent RFID labels are a substantially more affordable decision than dynamic RFID labels, and cost around 20 pennies each. This settles on them a mainstream decision for production network the executives, race following, document the board, and access control applications. While an uninvolved RFID tag doesn't need an immediate view to the RFID pursuer, it has a lot more limited read range than a functioning RFID tag. They are little in size, light weight, and can conceivably endure forever.

AD8232sensor

Fig 4.2: AD8232 heart sensor

AD8232 is a little, impeccable chip which is used to test the electrical action of the heart. This electrical movement can be outlined as an ECG. Electrocardiography is utilized to assist with diagnosing different conditions of the heart. So in this project, we will interface this ECG Sensor with Arduino and recognize the ECG signal in a fixed chart of plotter or in Processing IDE's.
This heart rate detector sensor is a sharp board utilizes to screen electrical action of the patient heart. These electrical improvements can also be drawn as an ECG and introduced as an immediate test. ECGs can be exceptionally befuddling; This Single Lead Heart Rate sensor runs nearly as an amp to assist with getting irrefutable sign from the time periods and QT with no issue.
AD8232 is a blog for enhancing the arranged ECG signal and other testing programs. It is dedicated to eliminate,

upgrade, and move little signals inside there cognition of uproarious circumstances, for instance, those made by and improvement or distant a node activity.

ii. Software Requirements

LAMP

LAMP is a Web Socket sub protocol enlisted at IANA, indicated to offer steered RPC and Pub Sub. Its plan objective is to give an open norm to delicate constant message trade between application parts and facilitate the production of approximately coupled models dependent on micro services. Along these lines, it is a reasonable undertaking administration transport (ESB), fit for creating responsive Web applications or to organized if ferent associated gadgets in the IoT.

LAMP requires a solid, requested, full-duplex message channel as a vehicle layer, and of course utilizes Web socket. Not with standing, executions can utilize different vehicles coordinating with these qualities and speak with LAMP over for example crude attachments Unix attachments or HTTP long survey.

Message serialization assumes numbers, strings and requested grouping types are accessible, and defaults to JSON as the most widely recognized arrangement offering these. Executions regularly give Message Pack as a quicker option in contrast to JSON, however at the expense of an extra reliance.

To distinguish far off methods and Pub Sub points without clashes, LAMP additionally needs an ID space permitting worldwide task and goal. Since the convention is Web local–Web Socket being the favored vehicle-URIs are utilized.

LAMP is architectured around customer correspondences, with a focal programming, the switch, dispatching messages between them.

As LAMP utilizes Websocket, associations can be enveloped by TLS for encryption. In any event, when full secrecy isn't set up, a few systems are carried out to disconnect segments and keep away from man-in-the-center assaults. Default executions guarantee that attempting to enlist a general lyen rolled methodology will fall flat.

Switches can characterize domains as managerial areas, and customers should indicate which domain they need to join upon association. When joined, the domain will go about as a namespace, forestalling customers associated with a domain from utilizing IDs characterized in another for RPC and Pub Sub. Domains likewise have authorizations connected and can restrict the customers to one subset of the REGISTER/CALL/Pub Sub activities accessible.

A few domains must be joined by validated customers, utilizing different confirmation strategies, for example, utilizing TLS authentication, treats or a straight forward ticket.

Ubidots

The fundamentals parts of IoT applications constrained by Ubidots are: Devices, Synthetic Variables Engine, Variables, Dashboards, and Events.In this section we will describe all of these parts related to Ubidots IoT Development and Deployment Platform and how we can all the promptly organize Ubidots Apps to best suite our application.

At the point when system contraptions, factors, are gathered, we can custom our App with a couple of layers with Ubidots Device Management to con template Ubidots insight plan and how we can use Apps, Organizations, and Users are feasibly connect to the data with the people who should use it.

An Ubidots' appliance is an effective prediction of an information resource or essentially, a resource taking sensor information and granting said information through a connection show to Ubidots' cloud. Take some break here for current firmware models and instructive exercises for accomplice your gadget to Ubidots.

MQTT Protocol

The "MQTT stands for Message Queue Telemetry Transport" is a light-weighted, open advising show that outfits resource constrained organization clients with an essential technique to proper telemetry information in low-move speed conditions. The show, which uses an appropriate/ purchase in correspondence configuration, is used for machine-to-machine (M2M) correspondence.

In spite of the fact that MQTT began as an exclusive convention used to speak with administrative control and information obtaining (SCADA) frameworks in the oil and gas industry, it has gotten well known in the keen gadget field and today is the main open source convention for associating web of things (IoT) and mechanical IoT (IIoT) gadgets.

The TT in MQTT addresses Telemetry Transport, the MQ is with respect to a thing called IBM MQ Albeit the illuminate for MQTT is in some cases given as Message Queuing Telemetry Transport, there is no message lining in MQTT correspondence.

MQTT customers incorporate distributers and supporters, terms that allude to whether the customer is distributing messages or bought in to get messages. These two capacities can be carried out in a similar MQTT customer.

Exactly when a device (or client) necessities to send data to a laborer (or vendor) it's everything except a distribute. Exactly when the movement is exchanged, it's everything except an upfront investment.

V. METHODOLOGY

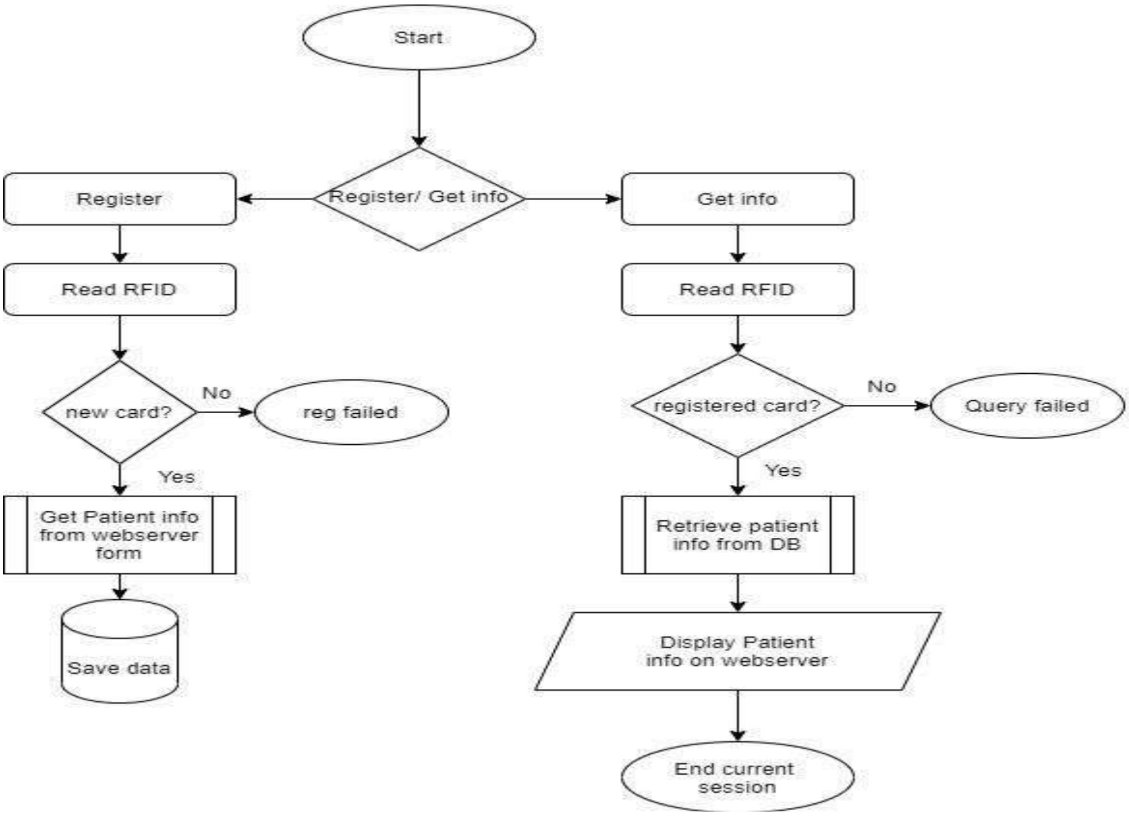


Fig 5.1: System flow diagram

The basic components of the system design include a Database, python code and Arduino code running in the backend and an apache web server in the front end for patient information system; whereas, the work uses a third party IoT vendor ubidots for publishing data from the medical sensors. The system flow chart for patient information storage and retrieval system is as shown in figure below.

The process begin with the authorized person logging in into his/her account and select whether to register a patient or retrieve/ update patient data. Once the appropriate option is selected the system will return a registration form in case of first option or it'll display relevant information of the patient in case of the latter option.

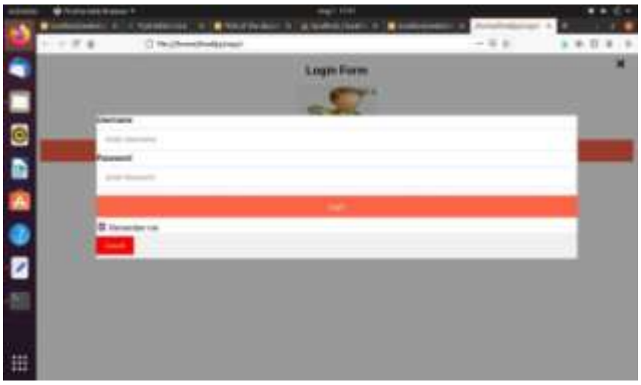
VI. RESULT AND CONCLUSION

The framework has been tested on a linux machine with basic web browsers such as Mozilla firefox, google chrome and safari.

Results were found to be satisfactory and framework tested to be functional.

Fig 6.1: User login

This is our user login form, to get access into the system first user have to login with their register username and password.



Login Form

Username

Password

Login

☑ Remember me

Cancel

Fig 6.2: Login popup

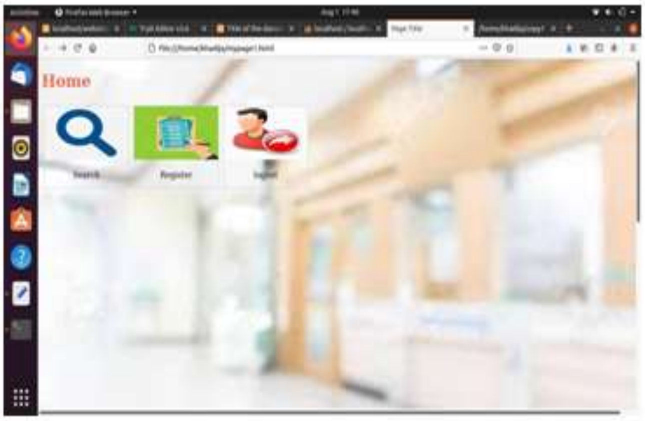
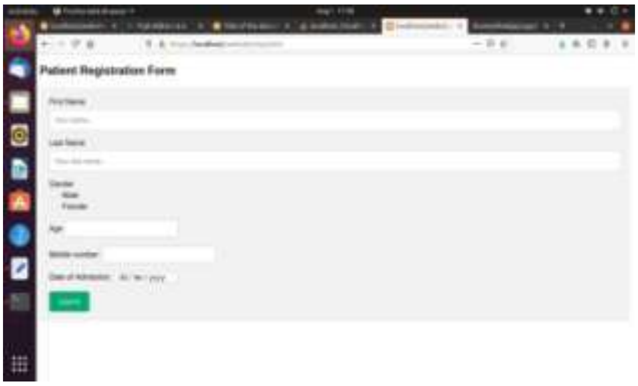


Fig 6.3: User Home page

This is our homepage of the system, in that we have 3 modules **search** for the patient details, register for the patient registration form and last logout for logging off from the system.



Patient Registration Form

First Name

Last Name

Gender
○ Male
○ Female

Age

Mobile number

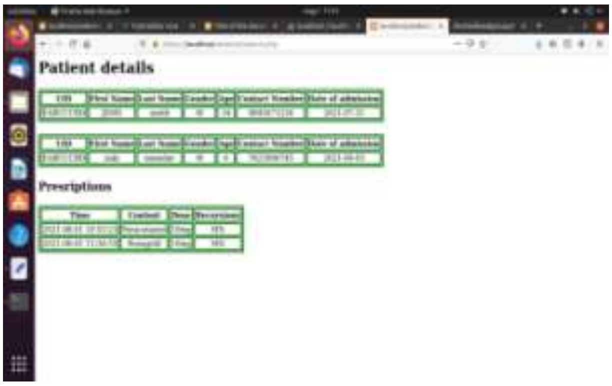
Date of Admission dd / mm / yyyy

Submit

Fig 6.4: Patient registration form

This is our patient registration form, which consist of some field attributes and a submit button. Fields are namely First name, Last name, Gender, Age, Mobile number and Date of Admission.

After entering all the field values user has to click on submit button to register the patient into the system, before the values are added in to the system the browser will wait for the scanning of the RFID card on the RFID reader, this RFID card will become the primary key for the patient to uniquely identification.



Patient details

UID	First Name	Last Name	Gender	Age	Contact Number	Date of admission
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

UID	First Name	Last Name	Gender	Age	Contact Number	Date of admission
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Prescriptions

Time	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

Fig 6.5: Patient details display page

This is our patient detail form,which shows the details of the patient which include their personal as well as medical details.

VII. MERITS OF THE SYSTEM

- Less prone to error due to misidentification of patients.
- Automated health updates to attendants.
- Easy access to case history when referred to specialists
- Secure data base ensures patient's privacy
- All data available at fingertips

VIII. REFERENCE

[1]. Haripriya, K., Aravind, C.M., Karthigayen, V., Ganesh, P.: Patient health monitoring system using IoT and cloud based processing. Indian J.Sci. Technol.9 (2016)

[2]. Abdullah, W.A.N.W., Yaakob, N., Elobaid, M.E., Warip, M.N.M., Yah,S.A.:Energy efficient remote healthcare monitoring using IoT: a review of trends and challenges.In:ICC2016,March22–23(2016)

[3]. Abasi Julius and Zhang Jian-Min, "IoT Based Patient Health Monitoring System UsingLabVIEW and Wireless Sensor Network", published on the International Journal of Science and Research(IJSR), ISSN(Online): 2319-7064,Volume 6,Issue 3,March2017

[4]. Framework for Monitoring Patients Health Using Sensors A Pravin, TPrem Jacob ,PAsha International Conference on Intelligent Data Communication Technologies and InternetofThings,1250-1254,2018.

[5]. IoT Based Remote Medical Diagnosis System Using Node MCU Fahim Faisal Syed Akhter Hossain 2019 13th International Conference on Software , Knonwledge, Information Management and Applications (SKIMA), 1-7,2019

[6]. “Node MCU v2,” Last accessed on February 1, 2019, at 1:20:00PM [Online]. Available: https:// www.seeedstudio.com/NodeMCU-v2-Lua-based-ESP8266-development-kit.html

[7]. “digital Read (),” Last accessed on February 1, 2019, at 01:45:00PM [Online]. Available: https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread

[8]. “analog Read (),” Last accessed on February 1, 2019, at 01:55:00PM [Online]. Available: https://www.arduino.cc/reference/en/language/functions/analog-io/analogread

[9]. “Arduino Nano,” Last accessed on February 1, 2019, at 1:10:00PM [Online]. Available: https:// www.arduino.cc/en/Guide/ArduinoNano.