Web Application for an IoT based Remote Health Monitoring System

Sachin Raja
Embedded Computing and Signal Processing Laboratory
Armour R&D
Illinois Institute of Technology Chicago, Illinois
Professors: Dr. Jafar Saniie

Abstract: In general, most of the people want to have an independent life and live on themselves, especially patients and elderly people. It is essential to monitor their vital signs, health and safety. Internet of Things (IoT) has provides better access to monitor the data remotely. If the person has had a fall at a remote location, it's a tedious task to find the patient. So, the location of a fall of a patient is also important along with the detection of a fall. Incorporating body sensors that are connected to the internet and the data from these sensors along with the location is saved on the database.
It is then analyzed and interpreted through a website. In this project we monitor the vital data of a person through a website created and provides a remote system for alerting and detecting dangerous falls of the patient to the physician. With the help of data analysis, we analyze the vital information of the patient along with the location of the patient and trigger an email to the physician so that proper actions can be taken at the right time.

Keywords: IoT, Web Application, fall detection, Health Monitoring

I. INTRODUCTION

Internet of Things in healthcare domain is gaining popularity as it serves all purposes and is used in many wellness devices. This helps monitor a person's health to an unimaginable extent. To address the challenges in monitoring a patient's health, providers in the healthcare domain have starting using Internet of Things to increase awareness of patient conditions and provide better insight to the physician. This new concept has allowed the amalgamation of new applications of ingenious/smart sensing devices which can be embedded into existing medical devices to improve the consciousness of the patient's wellbeing. This integration has allowed doctors to observe the patient vital signs at any location, using the data collected from the smart devices embedded the physicians can keep patients under observation from any part of the world through the internet and any point of time. This system provides more insight regarding patient vital signs and allows physicians and nurses to detect and treat deadly conditions even before they occur. Since, these smart devices operate remotely it is possible that a person's wellbeing can be tracked by the physician even if he is in at his home without the hassle of going to the hospital for a checkup. This allows patients to be treated at home or at their convenient place rather than the hospital.

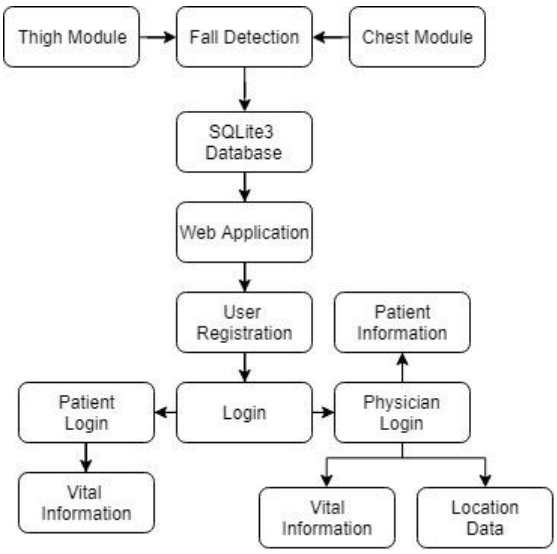
In-Home treatment has become a trend in the healthcare industry, and it is required for a patient that needs constant observation by a physician in-person which is expensive and burdens patients being treated. When the patients are not monitored regularly there may be moments that can lead to a point where a lethal health event may occur. So IoT provides a solution to this problem by providing the ability to monitor the patient solves then problem as here a trained professional physician receives alerts about deadly events.

It is declared by the World Health Organization that; Falls are the second leading cause of injury that are caused by unexpected events across the world. And among these senior adults of age greater than 65 years old suffer the greatest number of fatal falls. The project model developed here detects the fall that has occurred in the system and the vital data will be stored in the database and will be displayed in the system. The physician tagged to the patient will receive an alert stating that the person has fallen and also gives the location of the fall so that immediate action can be taken to give necessary treatment to the patient. This is very helpful as the physician can use the website anywhere in the world to monitor the patient's stats and also get the location of the patient when a severe event occurs. This system will allow the user to track their own wellbeing by providing with the feedback about their wellbeing and lifestyle. The fall detection system is calibrated to identify initial position of the user like standing, sitting, or lying down, later give a notification for the fall. Based on the fall readings an alert is sent to the physician along with the location of the fall. The system is designed to present data samples to the user and physician for analysis.

II. METHODS

1. System Overview and Hardware:

In the model created the is depicted in the below flowchart. Based on the data received through the thigh module and chest module the fall detection is interpreted and stored into a text file. This text file is processed and fed into the SQLite3 database. The web application displays the data from the SQLite3 database in the form of an interesting website using a raspberry pi as a webserver. The website is created using the python-flask microweb framework. SQLite3 as the backend database and HTML, CSS3, Bootstrap for the front-end GUI.



i. System Flowchart for the model

2. Fall Detection System:

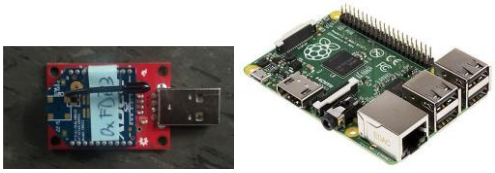
The fall detection system used here is a modified version of the fall detection system created and designed in the ECASP lab by Bruno Fernandes dos Santos and Eduardo Fonseca Carvalho. The system consists of a thigh module comprising of an accelerometer to detect the user's movement is connected to the user's thigh and a XBee module for communication, a chest module comprising of an accelerometer, a gyroscope to detect the user's position is connected to the chest of the patient and a XBee module for data communication. Both the modules are integrated with an Arduino pro mini that is used as a central co-ordinator for each unit. These Arduino pro minis operate on the Arduino IDE which uses C++ programming language to program the Arduino. The program used to detect a fall is written in C++. This system successfully not only detects a fall but also can detect a false fall and stores them on to a text file.

Positional data of the user is retrieved using two accelerometers and a gyroscope. The accelerometer in both the thigh and chest modules are triple axis accelerometers, and they detect motion in the all three axes directions i.e. x,y,z. The thigh module consists of only the accelerometer (adafruit ADXL345) and the chest module consists of both accelerometer and gyroscope combined chip model L3GD20/LSM303. These modules collect data in different ways but were integrated implementing a set of libraries provided by the chip manufacturer.

ii. Thigh module for the fall detection system: XBee, Arduino Pro mini, and an accelerometer(ADXL345)

iii. Chest Module for the fall detection system: XBee, Arduino Pro mini, and a gyroscope/accelerometer module (L3GD20/LSM303.)

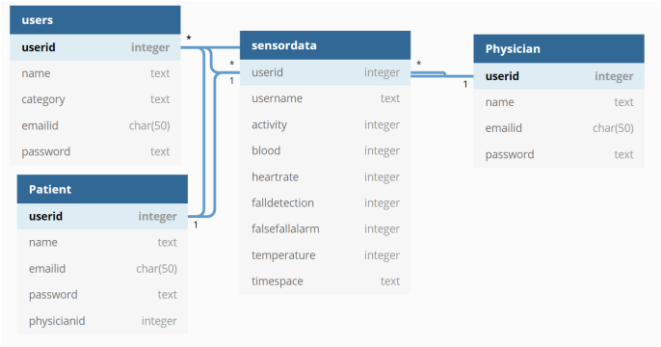
Both the chest and thigh modules are connected to a webserver to perform data analysis and storage. The webserver is hosted on a Raspberry Pi. The central XBee unit is receives the data that is transmitted by XBee in the thigh and chest modules in the system and the Raspberry Pi collects the received data and stores it into a relational database that is discussed in detail later for further analysis. The XBee unit communicates directly with the Raspberry Pi through a USB connection. Raspberry Pi 3B+ used has built-in wifi compatibility that is used to connect to the internet to host the webserver.



iv. The central XBee module on the left connected to the raspberry pi 3B+ on the right.

3. Database:

The database diagram given below is the relational database implemented on SQLite3 database. It is called a relational database because all the tables in the data base are connected to each other through a unique id. It consists of four tables: users table that contains all the users present in the application including the physicians, Patient table that contains all the patient information and also the physicians they are tagged to, sensordata table consists of all the data regarding the vital information of the patient, physician table that contains the physician information in the application. All these tables are connected using the userid as a common relation. Using the userid we can track all the information narrowed down to one user.



v. Database Diagram for the web application used

4. Web-Application:

For creating the web application, I have used python-flask microweb framework. It is a very light weight and powerful web framework and can be used in on devices with low processing power like the raspberry pi. The backend of the website is written in python. The frontend or the UI of the website is created using HTML, JavaScript, CSS3 and Bootstrap.

Flask is a microweb framework that is written in python and it does not need any specific tools or libraries. Flask is built on a range of open source packages, the most important of which is the WSGI application development library Werkzeug and the template engine Jinja. There is no database component in the Flask framework. In Flask we do not need to pay attention to the network operation when doing Flask development. The entrance of the Flask application is the encapsulated network request packet, and the exit is the network response. We only need to pay attention to this stage. A sample program in flask is explained below

1. Importing flask class:

The Flask class below is the core class of the Flask framework, which implements the WSGI application specification.

```
from flask import Flask
```

2. Creating a flask instance:

The first argument after the flask class declaration is the Flask constructor that specifies an import name. The Flask framework uses this name to locate static resources, templates, and error messages. The Flask instance is callable (with the call method) as seen below, which can be directly docked to the WSGI server

```
app = Flask(__name__)
```

3. Registration route:

Registering a route is to establish an association between a URL rule and a handler as shown below. The Flask framework relies on routing to complete the distribution of HTTP requests. The function in the route is called a view function, and its return value will be the body content of the HTTP response.

```
@route('/')
def index():
    return 'Hello, World!!'
```

4. Start the WSGI server:

Flask encapsulates a simple development WSGI server, we can start the server by calling run() when the main function is triggered as shown below:

```
if __name__ == '__main__':

    app.run(host='0.0.0.0',port=80)
```

Port 80 determines the port through which the data is transferred to the browser.

This gives the basic outline of how to host a webpage using python flask. It is a framework with a huge number of possibilities.

HTML (**H**yper **T**ext **M**arkup **L**anguage), Standard Generalized Markup Language application under. HTML is not a programming language , but a markup language that is essential for web page creation. " Hypertext " means that the page can contain images, links , and even non-text elements such as music and programs . The structure of the Hypertext Markup Language includes a " header" portion, and a "body" portion, where the "header" portion provides information about the web page and the "body" portion provides the specific content of the web page.

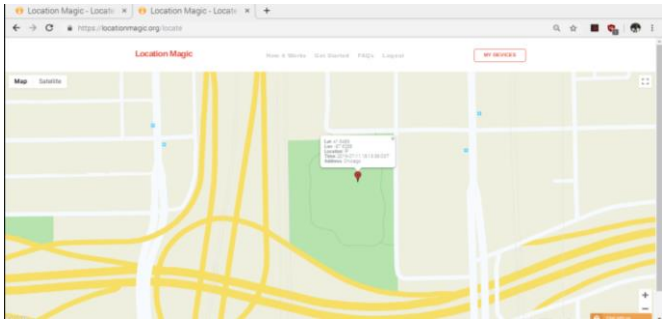
Cascading Style Sheets (CSS3) is a programming language used to represent file styles such as HTML (an application of the standard universal markup language) or XML (a subset of the standard universal markup language). CSS can not only statically modify web pages, but also dynamically format various elements of web pages in conjunction with various scripting languages. CSS provides pixel-level precision control over the layout of element locations in web pages, supports almost all font size styles, and has the ability to edit web page objects and model styles.

JavaScript is a web-based scripting language that has been widely used in web application development. It is often used to add a variety of dynamic functions to web pages to provide users with smoother and more beautiful browsing results. Usually JavaScript scripts implement their own functionality by embedding them in HTML.

Bootstrap is a front-end framework for rapid development of web applications and websites. Bootstrap is based on HTML, CSS, and JavaScript. Bootstrap provides a basic structure with a grid system, link styles, and background. This project I have used jumbotron UI content style to develop the entire website.

For getting the location of the device I have used a location API from unwired labs that provides the location of the patient a **GeoLocationAPI**.

- This **GeoLocation API** can locate any **IoT** and other connected devices **without a GPS**.
- Using the cellular and Wi-Fi network nearby the device is located based on the various access points available and the available access points are sent to unwired labs server.
- This data is being analyzed using large datasets of cell towers and Wi-Fi networks that are assisted by complex algorithms to send back the location of the device.
- The location of the raspberry pi is returned as shown below using the token generated for the device



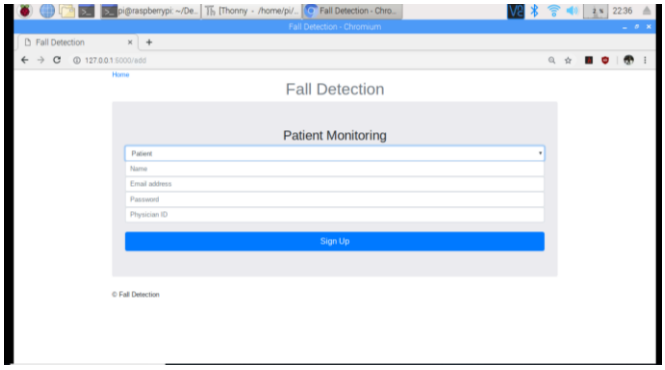
vi. The **API** returns a location as above providing the latitude and longitude values of the location.

III. RESULTS

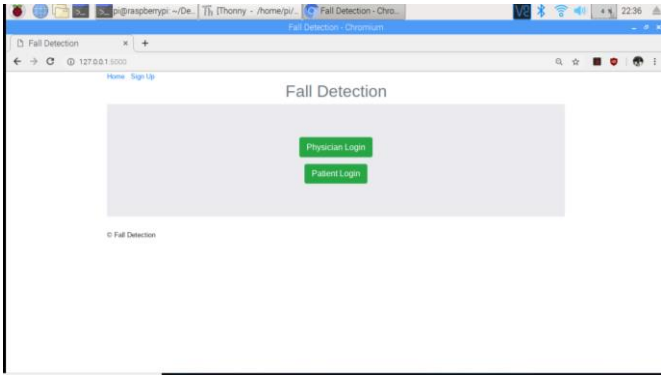
The C++ code that runs the hardware is executed on the raspberry pi and the positions of the patient are recorded as below.

```
pi@raspberrypi:~/Downloads $ g++ -o fall SachinRaja.cpp -lwiringPi
pi@raspberrypi:~/Downloads $ ./fall
successfully opened serial connection
684     2019-07-19 17:07:47     False Alarm
724     2019-07-19 17:07:50     Sitting
752     2019-07-19 17:07:53     Upright
783     2019-07-19 17:07:55     Lying down
794     2019-07-19 17:07:56     False Alarm
801     2019-07-19 17:07:56     Upright
811     2019-07-19 17:07:57     Sitting
817     2019-07-19 17:07:57     Upright
830     2019-07-19 17:07:58     False Alarm
837     2019-07-19 17:07:59     Sitting
843     2019-07-19 17:08:00     False Alarm
868     2019-07-19 17:08:02     Upright
919     2019-07-19 17:08:06     Sitting
934     2019-07-19 17:08:07     Upright
959     2019-07-19 17:08:09     Lying down
980     2019-07-19 17:08:10     Upright
996     2019-07-19 17:08:12     Sitting
1004    2019-07-19 17:08:12     Upright
1032    2019-07-19 17:08:15     False Alarm
1075    2019-07-19 17:08:18     Upright
1095    2019-07-19 17:08:20     User fell forward
1105    2019-07-19 17:08:21     False Alarm
Fall at 1095 marked as False Alarm
1119    2019-07-19 17:08:22     Upright
```

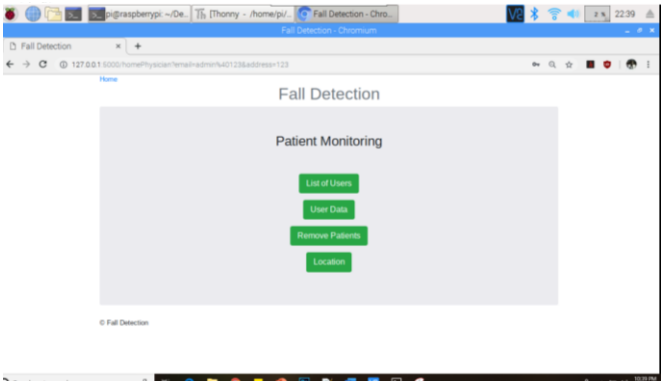
My website consists of various pages right form the user registration page to removing the user from the database.



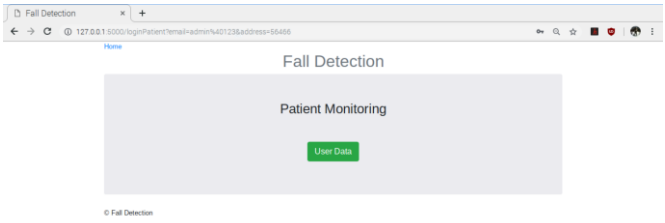
The **user signup page** where the user can sign up as a physician or a patient. The patient will be tagged to a physician always.



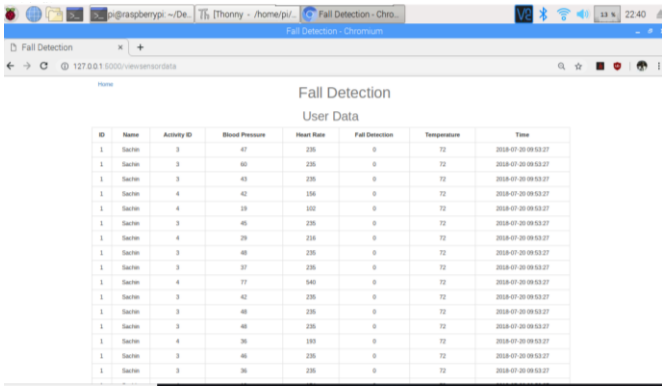
This is the **home page** of the website and based on the selected options the user can login



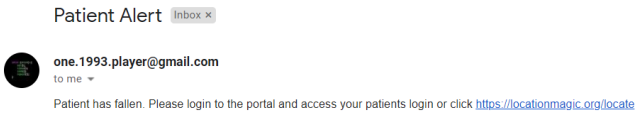
physician home page where he can view the location of the patients fall, patient's vital data, and can remove the user from the system.



Patient Home page where the patient can only view the user data to track himself



The **patient's vital data** is represented in the form of a table as shown above



The above screen shot Email Alert the physician gets when the patient has fallen.

IV. DISCUSSION

Implementations of such systems in the medical industry is very convenient and timely to monitor a person's health. Patients who are prone to falls in a hospital, homes would benefit a lot as they may require continuous monitoring that the physician cannot provide always.

The size of the system is bulky, there is need for miniaturization of the system. This is the next step in the future works of this project. The code used for the hardware was written in C++ and was running on the raspberry pi. The web server implementation was written in python. This causes two environments to be set up to run the system. So the C++ code is to be converted into python to a unibody code.

V. CONCLUSION

The system differentiates falls and false alarms and It uploads the sensor data to the SQLite3 database successfully. The elimination of GPS sensor reduces the cost.

The website is designed in such a way that there are two different views for both physician and patient.

This application has allowed the system to provide greater insight into the patient's well-being. The location of the fall is also displayed on the website providing a many possibilities for the physician to help the patient.

VI. FUTURE WORKS

This system can be implemented on a cloud-based infrastructure and which can improve the performance of the website.

We can use a TCP client app to fetch the location data from the cell phone itself.

This current fall website can be extended using various vital sensors like blood pressure, heart rate etc.

VII. ACKNOWLEDGEMENTS

I would like to thank the rest of the researchers at the Embedded Computing and Signal Processing (ECASP) Research Laboratory for their continuous and timely support. This project could not have come so far without their help. I would also like to thank the Armour R&D ACE Undergraduate Research Program for their support.

VIII. REFERNCES

1] W.-J. Yi, B. David Arnold, Andrew Mustea and J. Saniie, "IoT Remote Health Monitoring System," ECASP Lab, Chicago, 2018.

2] Kabalan Chaccour, Rony Darazi ; Amir Hajjam El Hassani ; Emmanuel Andrès: From Fall Detection to Fall Prevention: A Generic Classification of Fall-Related Systems, 2016.

3] World Health Organization, “Falls”, 16 January 2018, http://www.who.int/news-room/fact-sheets/detail/falls

4] W.-J. Yi and J. Saniie, "Patient Centered Real-Time Mobile Health Monitoring System," Scientific Research Publishing, Chicago, 2016.

5]For the location API: https://unwiredlabs.com/docs-html/index.html?python#endpoints

6] For flask: https://flask.palletsprojects.com/en/1.0.x/

7] For bootstrap: https://getbootstrap.com/docs/4.3/getting-started/introduction/