Instituto Superior Técnico

Mestrado Integrado em Engenharia
Electrotécnica e de Computadores

Wireless Mobile Networks

2nd Semester – 2019/2020

Intermediate Report

Detecting and reporting human falls

Diogo Silva 79462

June 7, 2020
Professor: António Grilo

1 Introduction

As the human body ages, it becomes weaker and susceptible to various threats. For higher risk groups, like elder people, when suffering a fall, it is important to be assisted as soon as possible. The faster the assistance the more likely it is to prevent more serious consequences. In order to decrease this delay, it is important to study and develop new and more efficient ways to predict this scenario. This project intends to provide a solution to provide assistance as soon as possible. In the solution adopted, data from an accelerometer will be continuously acquired and analyzed. In case a fall is detected, an alert signal will be sent to a mobile application. A camera will also be used to transmit live footage that will be accessible through a web browser.

2 Overall solution

Next will be described the different parts acting in the presented solution.

2.1 ESP8266 NodeMCU Development Board v1.0

The NodeMCU is an open source LUA base firmware developed for the ESP8266 and provides connection to the internet via WiFi. It's role in this project is to transmit to the server the occurrence of a fall through TCP/IP communications.

2.2 ESP32-CAM

As the ESP8266, this module firmware is produced by Espressif Systems. This MCU also has integrated WiFi and besides that, offers the possibility to connect a camera directly to the board. Here it is used to transmit *.jpg* files to the server when a fall occurs. This is done through TCP/IP communications. To be noted that in order to upload code to the ESP32, an FTDI programmer had to be used.

2.3 AltIMU-10 v3

This sensor from Pololu is an inertial measurement unit and altimeter with three integrated sensors: a L3GD20H (gyroscope), a LSM303D (accelerometer and magnetometer) and a LPS331AP (altimeter). For this project only the LSM303D was used. It provides digital 3-axis acceleration measurements with sensitivities in the ranges of ± 2 g to ± 12 g. This component was connected ESP8266 through I2C, where the measurements are analyzed in order to detect a possible fall. This is done by detecting two behaviours: the start of the fall and the impact. They are marked with a yellow and red zones and, respectively, in figure 4.

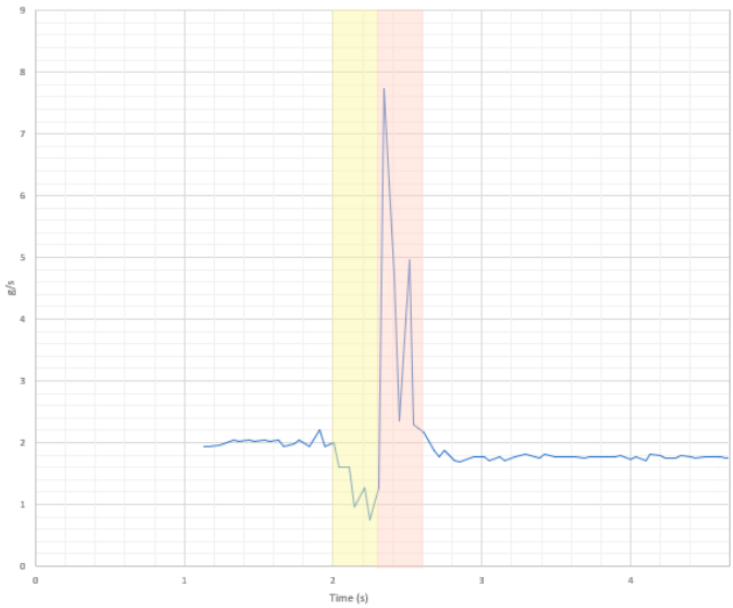


Figure 1: Acceleration change curves of the sum of the absolute value of the 3-axis readings during the process of falling

3 Sensors

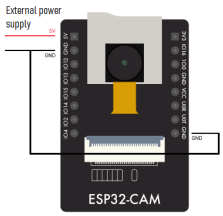
3.1 Hardware

Figure 2 shows the needed connections between the components.

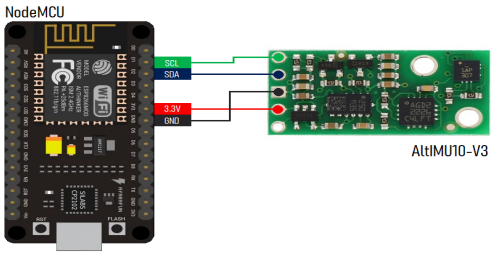
3.2 Software

In this project, an average is computed taking into account the sum of the absolute values for the three axis readings for the last five seconds. To identify the two stages referred in figure 4, the sum is compared to a lower and upper bounds that depend on the average. To identify the second stage as positive, after the start of the fall has been detected, a time gap is defined to consider the possible occurrence of the impact.

The ESP32 connects to the server every five seconds to check the patient status. If a fall occurs it immediately starts taking and sending photos. It sends the patient ID to a PHP script to create a directory, with the name equal to this ID, in the server where the footage will be saved and later accessed. The images are sent via HTTP Post requests to a PHP script.



(a) ESP32-CAM



(b) NodeMCU and AltIMU10-v3

Figure 2: Hardware connections

4 Cloud Server

All the back-end supporting the communications between the app, the MCU and the database can be found at http://web.tecnico.ulisboa.pt/ist178685/FallDetection. The back-end was mainly developed using PHP and MySQL. JavaScript and CSS was also used to produce a slideshow of the images uploaded to the server, making it look like a time-lapse.

4.1 MySQL database

Two MySQL tables were made to save valuable information regarding the entities involved in this project, the patient and the responsible, the first being the one who is monitored and the second the on who is monitoring.

id	name	surname	risk	isOK	responsible	camIP

Table 1: Patient

The variable *risk* can take the values L, M, H corresponding to low, medium and high respectively. *isOK* says whether the patient has suffered a fall or not, taking the value false in case he has. *responsbile* is a foreign key from the table responsible representing the ID of the person taking care of this patient, and

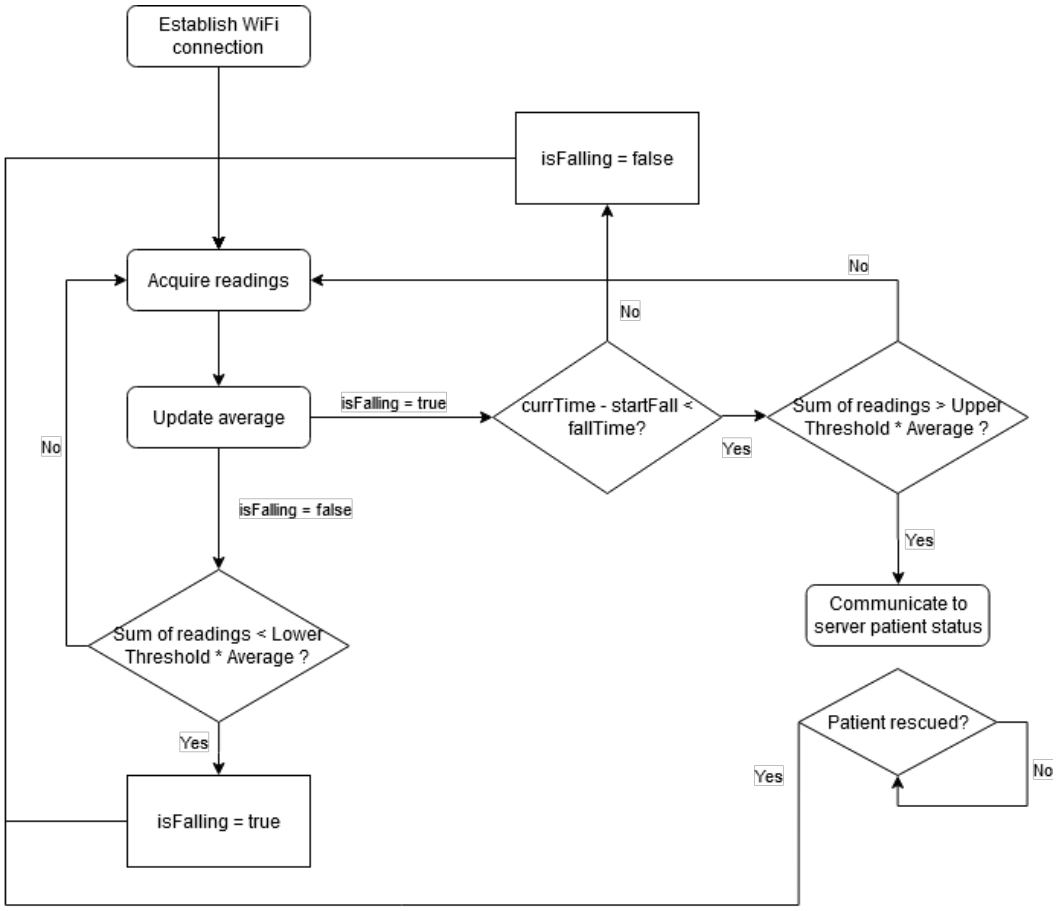


Figure 3: NodeMCU flowchart

lastly *camIP* will contain the IP address of the ESP32. It turned out not to be necessary.

id	name	surname	email	password	accessToken

Table 2: Responsible

The variable **accessToken** is the Firebase token generated by the mobile app when it is installed for the first time. It is necessary so the user can receive notifications when a patient falls.

4.2 PHP

The PHP scripts used are:

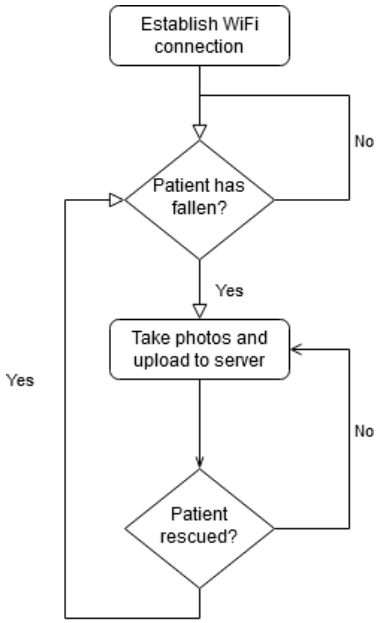


Figure 4: ESP32-CAM flowchart

- updatePatientStatus.php - when a fall is detected the NodeMCU sends a GET resquest with the patient ID in order to update the variable isOK to false. This will cause the responsible to receive a notification in his phone
- checkPatientStatus.php - both the NodeMCU and the ESP32 make GET request with the patient ID to verify the value of isOK
- newPatient.php - a responsible can add a new patient through the app by making a POST request
- createPhotoDir.php - the ESP32 makes a POST request to this script with the patient ID. A directory is created with the same name as the ID, if it doesnt't exist already and a "postPhotos.php" script is copied to that directory
- getPatients.php - to be used from the app, a GET request is sent with the responsible email and from there the script returns all the patients associated with it
- Login.php - the app sends a POST request with the email and password and the script returns the id and password associated with the email provided
- patientRescued.php - to be used within the app. A responsible can set the state of a patient isOK to true

- postPhotos.php - .jpg images are sent to this script via POST requests. The images are then saved in the same directory as the script with the time they arrived as name. The script also deletes any files in the directory with extension .jpg that are older than 1 minute
- showPhotos.php - this script includes JavaScript and CSS to produce a slideshow of the photos send by the ESP32
- Token.php - the app sends a POST request with the token and the ID to be saved in the database

5 App

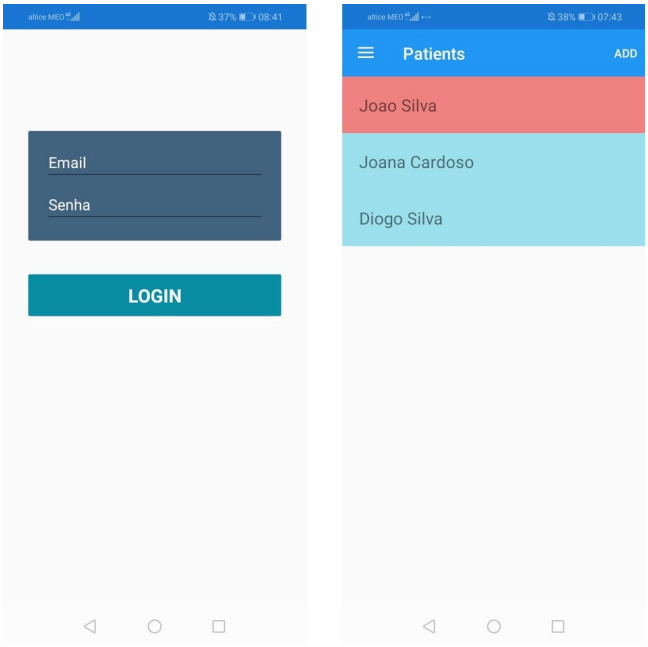
The mobile application was developed using the Xamarin Forms platform. This a tool that uses a .NET cross-platform UI enabling developers to deploy to several platforms with the same code. For this project the Nuget packages used were:

- Microsoft.Net.Http
- Newtonsoft.Json
- Xamarin.Firebase
- Xamarin.Forms
- Xamarin.GooglePlayServices.Base

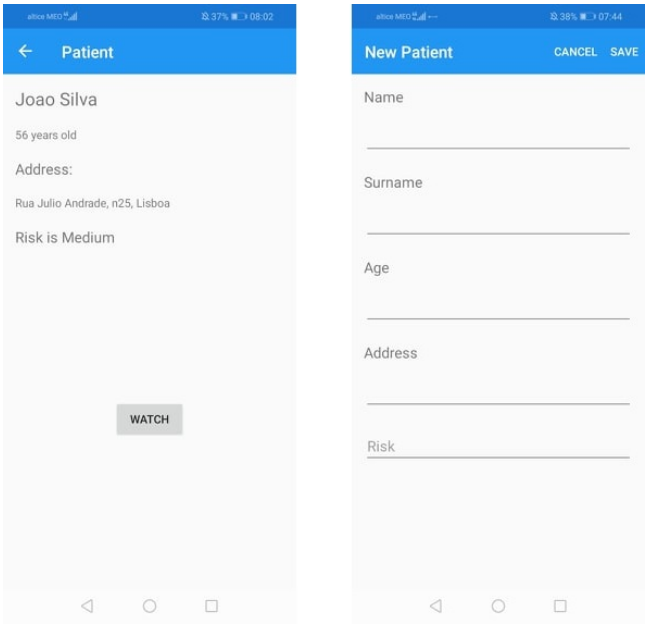
The app starts with a login page. The user credentials must already be in the database for him to be able to enter. After, he is presented with a list of the patients associated with him. If the patient presents a red color is because he suffered a fall and needs assistance. In this page there is also an Add button that allows the user to add a new patient to his list. By touching in one of the patients, the user is redirected to a new page containing all the patient details. Here he will have a button that when pressed, opens a browser in a link associated with the patients footage. This will only work if the patient has suffered a fall. If not a display alert will appear. If the patient needs assistance, there will also be another button is this page that should be pressed after the patient has been rescued.

6 Conclusion

The presented project offers a solution that can be used as basis for a more complete system. Most functionalities can be improved and some can be replaced by more efficient solutions. The parts of the project that are more alarming are the communication between the two MCU's and the need for live streaming video directly from the ESP32 to the application.



(a) Login page and patients list



(b) Patient detail page and add new patient page