



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

November 10, 2015

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 305/440W Design Specification for Fall Emergency Distress System

Dear Dr. Rawicz,

The attached document is the design specification for our product, the Fall Emergency Distress System (FEDS). FEDS will determine if a dangerous fall has occurred and will send a distress call to a 24/7 service system. In addition, FEDS includes an emergency button that allows the user to send a distress signal in case they require medical assistance.

This design specification document presents a high-level design for the proof-of-concept device of FEDS and reviews the specifications on each design scheme. Design provisions and speculation of FEDS are examined in high details for both its hardware and software components. Additionally, methods of testing the falling situations will be included in this document. This document will serve as a comprehensive reference during the development phases of our project.

4Life Technology Services consist of six highly motivated and talented fourth year engineering students: Yuvn Ng, Cyrus Chan, Benjamin Sia, Janet Mardjuki, Welson Yim and Daniel Lei. If there are any questions or concerns about our design specification document, please feel free to contact me by phone at 604-446-6738 or by email at cyrusc@sfu.ca.

Sincerely,

Cyrus Chan
CEO
4Life Technology Services

ENCLOSED: Design Specification for Fall Emergency Distress System



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

Fall Emergency Distress System

Design Specifications

Project Team: Cyrus Chi Chung Chan
 Yuvn Ng
 Benjamin Sia
 Janet Mardjuki
 Daniel Lei
 Welson Yim

Contact Person: Cyrus Chi Chung Chan
 cyrusc@sfu.ca

Submitted to: Andrew Rawicz
 Steve Whitmore
 School of Engineering Science
 Simon Fraser University

Issued: November 10, 2015

Revision Number: 1.10



Table of Contents

Executive Summary	iv
Glossary	v
List of Tables, Graphs and Figures	vi
1. Introduction	1
1.1 Scope	1
1.2 Intended Audience	1
2. General Overview.....	1
3. General System Design	2
4. Embedded System	3
5. Network Communication	3
6. Device Sensor	4
6.1 Proof-of-Concept Sensors.....	4
6.1.1 Accelerometer	4
6.1.2 GPS	6
6.2 Prototype Sensor	8
6.2.1 Gyroscope	10
6.2.2 Further Prototyping/Production Model	10
6.2.3 Heart Rate Sensor	10
7. Hardware User Interface	10
7.1 Status Light	10
7.2 Help Button	11
7.3 Cancel Button	11
7.4 On/Off Switch	11
8. Device Casing	11
9. Power Supply	12
10. Server Design	14
11. FEDS Device Application Client	15
12. Operator Web application Client	16
12.1 Client Status	17
12.2 Client Information Page	18
12.3 Further Design of the Operator Web Application	18
13. System Test Plan	18
13.1 Component Test	19
13.1.1 Hardware	19
13.1.2 Software	19
13.2 Integrated Test	19



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

14. Conclusion	21
15. References	23



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

Executive Summary

As we age, our bodies gradually become weaker and less responsive. Senior citizens tend to have difficulty controlling their body movements. At the beginning of the 20th century, it was estimated that there were at least 2.6 million fall related injuries in the US alone in which 10300 were fatal [1]. The graph in Figure 0.1 shows us the severity of this issue. It depicts an increase in death rate for individuals over the age of 65 who succumb to their fall injuries. Omar Aziz, a PhD graduate of SFU Engineering Science and a researcher in prevention of injury to the elderly, stated that the most common causes of falls involve an incorrect shift in body weight. In many cases, falls are unavoidable but we want medical assistance to arrive as soon as possible.

To solve these problems, our company is designing and developing a system called the Fall Emergency Distress System (FEDS), which responds to the falling motion of the user. When the user has a sudden downward acceleration, the system will detect this motion and will emit a buzzer alarm. This informs the user that a distress call will be sent to the service center if it is not deactivated within a certain period of time. The device will also include an emergency button that can be pressed anytime and it will notify the system of the user's need for medical assistance. When a distress call is received on our servers, the service team will locate the user through GPS and send the information to a medical response team.

This document outlines the details of the design specification and provide in depth calculation and analysis of the different aspects of the project. First, a general system design is describe to introduce our system functions as a whole. Next, we discuss on the sensors selection for the project followed by the hardware design. Lastly, the server and application design is reviewed in detail.

These specifications outline the detailed design of our proof-of-concept device. Access to this documentation will be provided to all members of 4Life Technology Services. Engineers are highly encouraged to refer to this documentation during the implementation of FEDS.

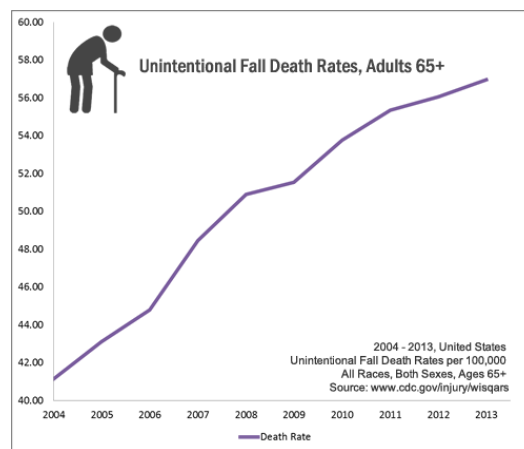


Figure 0.1: Death rate of unintentional falls for seniors age 65+ [2]



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

Glossary

API	Application Program Interface
EC2	Elastic Compute Cloud
DB	Database
FEDS	Fall Emergency Distress System
ID	Identification
GPIO	General Purpose Input Output
GPRS	General Packet Radio Services
GPS	Global Positioning System
GSM	Global System for Mobile Communication
HRPM	Heart Rate Per Minute
I2C	Inter-Integrated Circuit
I/O	Input/Output
LED	Light-emitting Diode
JS	JavaScript
JSON	JavaScript Object Notation
LiPo	Lithium Polymer
MSB	Most Significant Bit
NMEA	National Marine Electronics Association
OS	Operating System
POC	Proof-of-concept
RGB	Red Green Blue
SIM	Subscriber Identification Module
SSH	Secure Shell
UART	Universal asynchronous receiver/transmitter
USB	Universal Serial Bus
Wifi	Wireless Fidelity



List of Tables, Graphs and Figures

Figures:

Figure 0.1 (Death rate of unintentional falls for seniors age 65+)	iv
Figure 3.1 (FEDS' subnet system)	2
Figure 5.1 (EDIMAX EW-7811Un N150 USB 2.0 Wireless Nano Adapter)	3
Figure 5.2 (SIM808 GSM/GPRS/GPS Module (2015 ver.) for Raspberry Pi and Arduino)	3
Figure 6.1 (Accelerometer axis orientation)	5
Figure 6.2 (GPS breakout board connection)	7
Figure 6.3 (NMEA stream output for GPGGA).....	7
Figure 6.4 (Gyroscope rotation axis)	9
Figure 6.5 (Pseudo code to get the heart beat per second)	10
Figure 8.1 (Model of FEDS' case design)	12
Figure 9.1 (Schematic of Raspberry Pi Voltage Input Module)	13
Figure 9.2 (Aukey Amzdeal 9000mAh Portable Power Bank).....	14
Figure 9.3 (5V DC booster converter (Model: AiLi013))	14
Figure 10.1 (Client-server application framework)	15
Figure 12.1 (Client Status Interface of FEDS4LIFE system)	16
Figure 12.2 (Insert/edit Client Information Interface)	17

Equations:

Equation 6.1 (Accelerometer axis orientation)	5
Equation 6.2 (Formula to get magnitude of acceleration based on XYZ-axis output)	6
Equation 6.3 (Formula to get overall angular rotation)	9

Tables:

Table 6.1 (Data representation of x- axis for full-resolution with 8-g range output)	5
Table 6.2 (Interpretation of the output coming in the Raspberry)	8
Table 6.3 (Summary of the relationship between accelerometer and gyroscope in detecting fall)	9



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

1. Introduction

FEDS is a system for detecting falls, where a 4Life operator will be notified when a fall has occurred or if the help button has been pressed. This device is designed to detect a user's fall and to pinpoint their location regardless if the user is at home or outside. If the user falls or decides to call for help, the operator will take action and request assistance as soon as possible. This document contains the detailed specifications for FED's mock-up device and a less detailed specification for the future consumer production model. Along with the specifications, a detailed system test plan will also be included in this document

1.1 Scope

The document will focus largely on the specifications of different designs and functions. As such, it will list out the different components that will be used, as well as the functionality of each one. In addition, the document will provide explanations on why and how the components were chosen.

1.2 Intended Audience

This document is targeted towards all members of 4Life and any future engineers who will be involved in the project. This document is to be used as a guideline for these targeted users in order to create the mock-up device. As for the testing engineers, they are required to follow the system test plan accordingly to ensure that the device meets 4Life's intended standards.

2. General Overview

Fall Emergency Distress System (FEDS) is designed to provide an emergency response service when the user falls or decides to manually call for assistance. FEDS is a wearable device which is designed to be worn around the user's chest. The device will only be activated after the user registers the device and sets up their medical record. The medical record will then be uploaded with the specific device identification (ID) and then sent to the server's database. The following are two scenarios that would cause the device to send a distress response:

1. The first scenario occurs when the on-board sensors detect the fall of the user. When the user falls, a buzzer will sound at a low frequency for 30 seconds. If the user does not deactivate the buzzer within 30 seconds, the buzzer will sound at a higher frequency and the service operator will call the user. If the fall was a false alarm or if the fall was not serious, the user can push the cancel button on the device to cancel the countdown. However, if the user decides he or she is in need of assistance or has lost consciousness, the service operators will follow our designated procedures after 30 seconds. The designated procedures are the following:
 - a. The operator will first attempt to call the user's home/cell phone. If there is no answer, the operator will proceed to call the emergency contacts that the user has set up during device registration.



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

- b. If the emergency contacts cannot determine the well-being of the user, the operator would then call a medical response team. The user's location will be determined either by the WiFi connection or the GPS location of the device; the operator will give this information to the called emergency contacts and medical response team.
2. The second scenario occurs when the user is conscious and decides he or she needs emergency assistance. In this case, the user can press the help button on the device which will activate the distress signal. Once the distress signal is received, the operator will follow the same designated procedures mentioned above.

3. General System Design

The system is separated into two distinct parts: the embedded system and the server (shown in Figure 3.1).

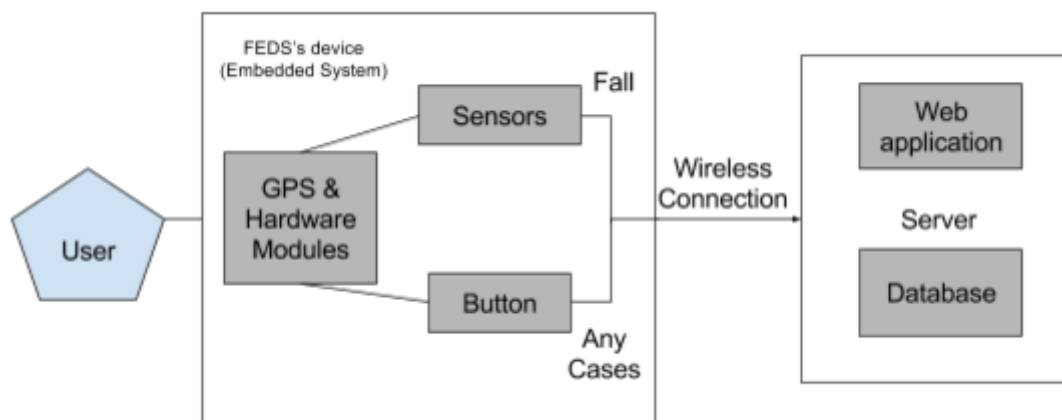


Figure 3.1: FEDS's subsystems

For the proof of concept (POC) device, the accelerometer is the driving sensor that will measure acceleration in the X, Y and Z axis. The accelerometer communicates with the microprocessor via the general purpose input/output (GPIO) pins. The microprocessor uses the data from the accelerometer and inserts it into an algorithm that has been rigorously tested by 4Life's team to determine if a fall has occurred. If the result of the fall is positive, the embedded system will activate the buzzer. While the buzzer is active, the user can press the cancel button which will reset the status of the device to its default state. Otherwise, if FEDS is left alone for 30 seconds, the device will execute the command to send a distress signal to the 4Life Technologies Services' server database through an internet connection.



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

The user will also have the option to press the distress call button at any time. Once the distress button is pressed, FEDS will execute the same command that collects the user's location and send a distress signal to the server database.

On the server side, the operator web application will constantly be updated by retrieving the client's information from the database to monitor if any client is in need of medical assistance.

4. Embedded System

For the POC model, a Raspberry Pi 2 B will be used as FEDS's digital signal processing unit. It is flexible for development purposes and contains most of the hardware components needed for development on-board. The SSH feature, which is included in the Linux operating system on the Raspberry Pi, allows the members of 4Life Technologies Services to wirelessly access the network connection for development processes. This feature is very useful as group members are not required to physically have the device in order to work on the project.

After the concept is proven with the Raspberry Pi, an Arduino Mini Pro will be used as FEDS's digital signal processing unit for the prototype version. The Arduino microcontroller will consume less power and have better I/O capabilities. Overall, after the prototype stage, FED's device will be even smaller and more efficient.

5. Network Communication

For our POC model, we are going to use the EDIMAX EW-7811Un Wireless Adapter (Figure 5.1). The advantage of this model is its small size, low cost, and it is WiFi certified [3]. It also complies with wireless IEEE 802.11b/g/n standards (except AC standard) and has a data rate up to 150Mbps. With these specifications, it will satisfy most WiFi network capabilities [4].



Figure 5.1: EDIMAX EW-7811Un N150 USB 2.0 Wireless nano Adapter

For future designs, the use of GSM from mobile networks services will be considered. This is because WiFi may have trouble accessing the network when the device is far from an access. By using Global System for Mobile Communication (GSM) or General Packet Radio Services (GPRS) networks, the user will not need prior knowledge about WiFi protocols or WiFi hotspots and passwords. The user can simply put in an active SIM card and connect with 4Life's network. One

possible replacement for the wireless Nano adapter is the SIM808 module. The SIM808 module includes the GPS and GPRS features but the cost is higher than the current GPS unit that FEDS is using (Figure 5.2). Once a GSM or GPRS module is implemented into the future design of FEDS, the WiFi feature will be removed.



Figure 5.2: SIM808 GSM/GPRS/GPS Module (2015 ver.) for Raspberry Pi and Arduino

6. Device Sensor

In the POC stage, an accelerometer sensor will be the source of verification for when a fall has occurred. The POC stage will also include a GPS module to determine the user's location at the moment a fall is detected. The accelerometer and GPS module will be implemented as part of the embedded system.

In the prototype stage, an additional gyroscope sensor will be included to measure the orientation of the user, further improving the accuracy of the fall detection algorithms. For the production stage, other sensors may be added to the production model. For example, a heart rate monitor could be included to collect data for hospitals research/monitoring purposes.

6.1 Proof-of-Concept Sensors

6.1.1 Accelerometer

The accelerometer is the component used to determine whether or not the user has fallen. The accelerometer of choice for the POC design is the Adafruit ADXL345 [5]. This accelerometer communicates with the Raspberry Pi through the I2C Bus. I2C Bus Protocol is a commonly used communication protocol for low speed peripherals to microprocessors. Because I2C is such a common protocol, the OS on the Raspberry Pi, Raspbian, already has the I2C driver pre-installed, allowing the Raspberry Pi to read through a virtual file.

When selecting an accelerometer to use, a couple of different choices were put into consideration. Initially, the ADXL335 [6] seemed to be a desirable choice because of its cheap cost and large amount of online support for this unit. However, the ADXL335 only measures a

range of $\pm 3g$. During our meeting with Omar, a PhD student who specializes in fall research, he suggested that the accelerometer should be able to detect a minimum range of $\pm 6g$. The next two accelerometers that were considered were the LIS331HH and the ADXL345. The LIS331HH has a measuring sensitivity level up to $\pm 24g$ while our ADXL345's was $\pm 16g$. Both were roughly the same price and satisfied the conditions needed for FEDS. However, we decided upon the ADXL345 in the end, as we have found that there was more online support for it.

The accelerometer has the following registers: DATA0, DATA1, DATAY0, DATAY1, DATAZ0, and DATAZ1. These registers continuously read the data which the microprocessor will use to calculate whether or not a fall has occurred. The XYZ orientation for the registers is illustrated in Figure 6.1 below.

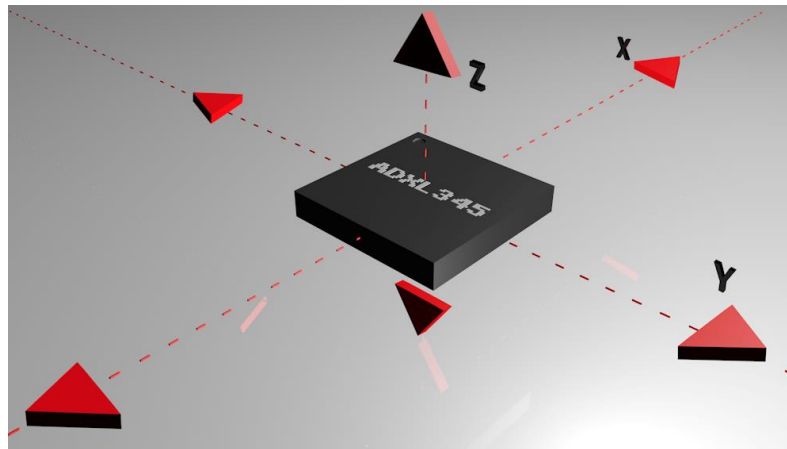


Figure 6.1: Accelerometer axis orientation

Data obtained from the accelerometer is stored in eight bit registers as signed binary numbers. The data from the registers will be taken in pairs as DATA_X, DATA_Y, and DATA_Z; only the first 12 most significant bits (MSB) will be used (D0 – D11). Looking at Table 6.1, the last three bits of DATA_{X0} will not be used as the data measured from the accelerometer with a range of 8g does not utilize them [7].

DATA _{X1} (1 byte)								DATA _{X0} (1 byte)							
sign	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Z	Z	Z

Table 6.1: Data representation of x- axis for full-resolution with 8-g range output

Each bits represents 6^n , so the formula to calculate the g-force output is shown in Equation 6.1:

$$g - force = \frac{(DATA_{x1} \& DATA_{x0}) \text{ srl } 3}{6^{12}} * 6g$$

Equation 6.1: Formula to get output in G



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

From experimentation, it was found that individually setting a threshold for each of the X, Y, and Z axis cannot properly detect a fall as there were too many false alarms from simple actions such as speed walking and sitting. In order to correct this, it was necessary to calculate the combination of each axis of acceleration and to set a threshold to that equation. The equation used is shown in Equation 6.2.

$$\text{Acceleration Magnitude } (A_m) = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

Equation 6.2: Formula to get magnitude of acceleration based on XYZ-axis output [8]

According to a study conducted by Chien-Hsien Liu, a Master student of Tatung University, 1.8g is a good threshold to set in order to determine the event of a fall [8]. However, in our experiments, we found that setting the threshold to 1.8g does not always distinguish whether a person is falling or sitting down, because the magnitude of an individual sitting down varies from person to person. To compensate for this, we determined that it was more accurate to request for the user to initialize the threshold themselves. This initialization process is done by creating a setup section where the user is required to stand up and sit down a couple times while A_m is being calculated. Using this data, the custom threshold is calculated and calibrated to the individual, decreasing the amount of false alarms detected by FEDS.

In regards to compatibility, the Raspberry Pi GPIO pins will drive all of the accelerometer's pins. The max current for each GPIO pin is 16mA per pin and the voltage is 3.3V. The accelerometer will not break from the GPIO's max current/voltage as 16mA at 3.3V is not high enough to cause any damage [9].

6.1.2 GPS

In order to locate the client after a fall, the FEDS system will use the Adafruit Ultimate GPS Breakout to get their current location. When searching for a GPS to use, there were a lot of different modules to choose from, such as the EM-506, Dexter industries GPS, and the RasPiGNSS. However, all of the mentioned GPS were either too big, too costly, or too power consumption heavy. The Ultimate GPS Breakout fits our application well as it has a low current draw (only 20mA), compact, and relatively inexpensive.

The Ultimate GPS Breakout interacts with the Raspberry Pi through the UART with the standard baud rate of 9600. Figure 6.2 displays the schematics of the breakout board and how the components communicate with one another.

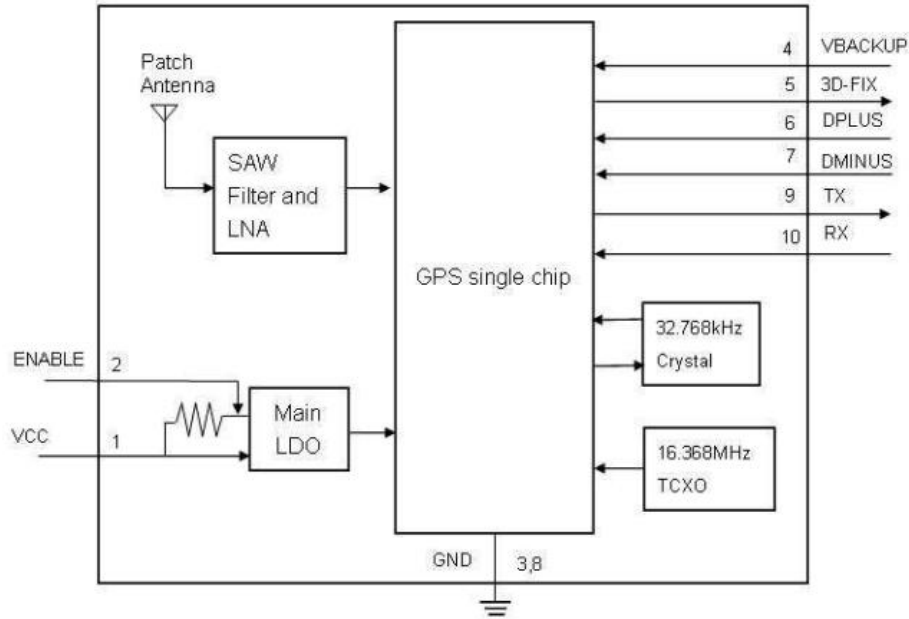


Figure 6.2: GPS breakout board connection

The GPS outputs a stream of position data from the NMEA output. The Raspberry would process this data and push its location to the MongoDB Server when a fall has been detected. Below is an example of the GPGGA stream [10] that will be collected from the GPS.

\$GPGGA,064951.000,2307.1256,N,12016.4438,E,1,8,0.95,39.9,M,17.8,M,,*65

Figure 6.3: NMEA stream output for GPGGA

The GPGGA stream can be decoded by the table in Table 6.2.



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

Table-2: GGA Data Format			
Name	Example	Units	Description
Message ID	\$GPGGA		GGA protocol header
UTC Time	064951.000		hhmmss.sss
Latitude	2307.1256		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12016.4438		dddmm.mmmm
E/W Indicator	E		E=east or W=west
Position Fix Indicator	1		See Table-3
Satellites Used	8		Range 0 to 14
HDOP	0.95		Horizontal Dilution of Precision
MSL Altitude	39.9	meters	Antenna Altitude above/below mean-sea level
Units	M	meters	Units of antenna altitude
Geoidal Separation	17.8	meters	
Units	M	meters	Units of geoid separation
Age of Diff. Corr.		second	Null fields when DGPS is not used
Checksum	*65		
<CR> <LF>			End of message termination

Table 6.2: Interpretation of the output coming in the Raspberry

Using Table 6.2, the data stream will be parsed and relevant data such as latitude, N/S Indicator, longitude, and E/W Indicator will first be converted into JSON format and then sent to 4Life's server when required. The server will use this data to map the client's position using Google Map's Geocoding Application Program Interface (API) [11] [12].

In a similar case to the accelerometer (as discussed previously), the breakout board is safe to use with the Raspberry Pi. The max voltage to drive the GPS pins is 3.3V and max current is 48mA; this only occurs during position acquisition.

6.2 Prototype Sensors

In order to improve the accuracy of our fall detector, FEDS will be using an additional gyroscope in the prototype stage if 4Life team decides to continue on the project. The gyroscope's function is to determine the user's orientation to the ground, which can help detect whether someone is sitting down or lying flat on the ground.

6.2.1 Gyroscope

In addition to the accelerometer, the use of a gyroscope will increase the accuracy of the fall detection algorithm. The gyroscope of choice will be the ST's L3G4200D and it is used to measure rotation and angular velocity. The reason why this gyroscope is chosen is because the L3G4200D and our ADXL345 (accelerometer) can be combined into a single module – the GY80 – which would save a lot of space [13]. In a similar case to the accelerometer as mentioned in Section 6.1.1, the gyroscope uses I2C Bus to interact with the Raspberry Pi and stores the XYZ

values in two registers as signed binary numbers (Figure 6.4).

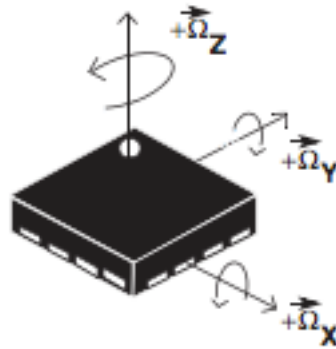


Figure 6.4: Gyroscope rotation axis

The formula in Equation 6.3 will be used to calculate the overall angular rotation on the user. If the overall angular rotation is higher than the threshold during a fall, the device will update the database to represent that a fall has occurred. Additionally, this formula can also be used to help detect and determine the user's orientation so we can make assumptions about the user's position: the user could be sitting down, meaning they will remain upright on a chair; the user has fallen, so it is highly probable that they will be horizontal to the floor.

$$\omega = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$$

Equation 6.3: Formula to get overall angular rotation

In order to increase the accuracy for detecting falls, the device will perform a binary AND operation with the calculated values from the accelerometer and the gyroscope. An operations table can be seen in Table 6.3.

	Accelerometer	Gyroscope	Output
1	True	False	False
2	False	True	False
3	True	True	True

Table 6.3: Summary of the relationship between accelerometer and gyroscope in detecting fall

Just like the other components, the gyroscope will be driven by the Raspberry Pi. Again, the voltage and current will be within a safe range for the gyroscope to function properly, as with the other components [14] [15].

6.2.2 Further prototyping/production model

Along with trying to improve the accuracy of fall detection, other sensors may be added to FEDS in the production stage to benefit not only the user, but hospitals and researchers as well. An example would be the addition of a heart rate monitor.

6.2.3 Heart Rate Sensor

The purpose of a heart rate monitor is to collect data for hospitals and researchers to study the heart rate before and after someone falls or to simply monitor the heart rate of a person throughout the day. The heart rate sensor will keep track of incoming heart pulses every minute to calculate the Heart Rate per Minute (HRPM). This incoming data will be collected and processed by the devices' programs and then sent to the database server. A specialized nurse or doctor may be able to monitor the changes in heart rate to infer the condition of the client and to notify emergency contacts of the situation. Figure 6.5 below presents the pseudo code in which the heart rate is sent to the server's database.

```
-- pseudo code to gather heart pulse
init heartPulseGpio

while pulseActive
    if oneSecPass then
        send(count)
        count=0
    end
    if thereIsPulse then
        count++
    end
    nanoSleep(50000)
end
```

Figure 6.5: Pseudo code to get the heart beat per second

Once again, similar to the other components inside the fall detection device, the compatible voltage and current output by the Raspberry Pi should not damage the sensor [16].

7. Hardware User Interface

This section goes over the interactive aspect of FEDS. The device contains a total of 4 major components which users can interact with or use to determine the status of the device. The 4 major components consist of a light source, help button, cancel button and on/off switch.

7.1 Status Light

The light source will be a RGB 4 pin LED. The LED will use 3 different colours to indicate the status of the device for the user. The following colors correspond with the mentioned status:

- Green light: The device is on and is monitoring the user's movement to sense for falls



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

- Yellow light: The device has detected a fall. When the status light changes to yellow, the buzzer will also buzz at a low frequency.
- Red light: The red light turns on 30 seconds after the yellow light has been activated and the user did not press the cancel button or if the help button is pressed. When the red light is on, it indicates the operator has been notified and response action is on its way.

There are many advantages of using an LED instead of other alternative light sources such as halogen and incandescent bulbs. Some of those reasons include: high energy efficiency, long lifetime, excellent color rendering (colors are not washed out), no warm up period (full brightness immediately), and it does not contain any hazardous chemicals [17].

7.2 Help Button

FEDS comes with a help button in the event when a conscious user decides he or she requires help right away. Pressing the help button bypasses the device's falling detection algorithms and immediately sends out the distress signal to the operator. Originally, 4Life's team was planning on making the help "button" as a switch. However, it was decided that a button was more suitable as it is easier to access when calling for help. Although the color of the button can be changed, it is suggested to be green coloured in order to distinguish it from the red coloured cancel button. Pressing the help button should set the status light to red.

7.3 Cancel Button

In case of a false alarm, the cancel button is used to stop the buzzer's alarm. If the cancel button is depressed before 30 seconds from the time the buzzer started buzzing, then the operator will not attempt to contact emergency services. If the button is not pressed 30 seconds after the buzzer has started buzzing, the operator will then follow the procedures mentioned in Section 2 of this document. It should be noted that aside from the chance of the battery running out, the cancel button should be the only way to stop the buzzer's alarm. Pressing the cancel button should also set the status light to green. It is recommended that the cancel button should be coloured red as people commonly associate "stop/cancel" as red [18].

7.4 On/Off Switch

The on/off switch is a simple switch to turn the device on/off. When the switch is turned on, the status light should turn green. When the switch is turned off, the status light should turn off. A switch was decided to be used as the on/off component because using a switch requires more precision than pressing a button. This way, the user will not accidentally turn the device on/off.

8. Device Casing

The casing is designed to hold the battery, Raspberry Pi and the sensors in different compartments of the case. The case has 2 sliders which the Raspberry and circuit board perfectly fits into to prevent parts from breaking from the shock damage when a person falls. These two sliders will also separate the wiring to avoid short circuiting the board. The casing is fillet on all 6 edges of the case to prevent it from injuring the user. Figure 8.1 shows an open view of the case with the Raspberry, battery and sensors inside.

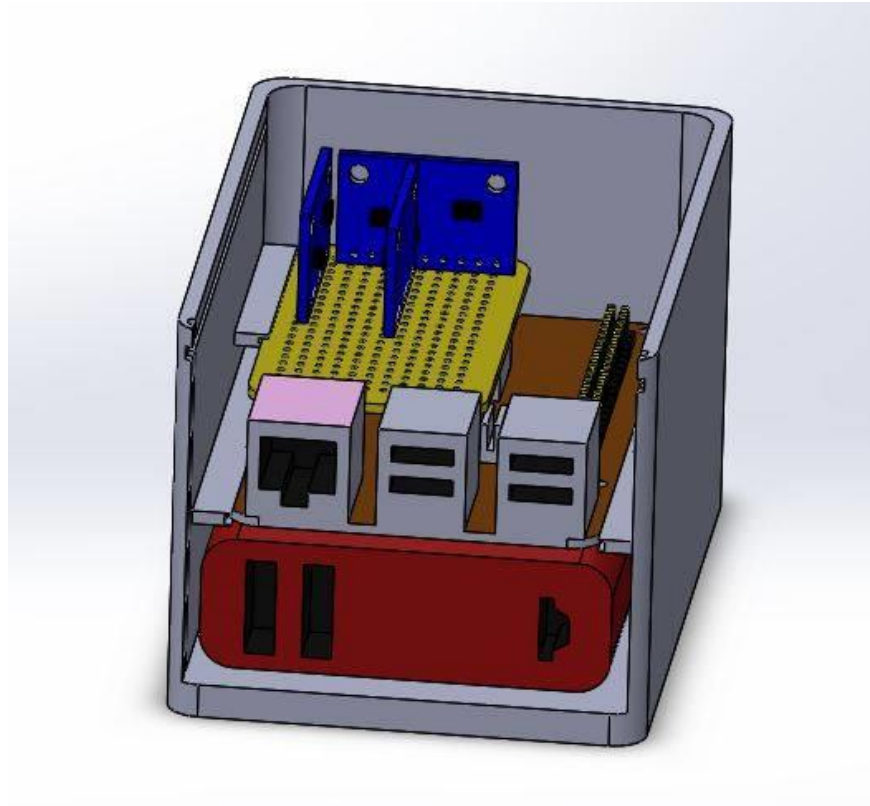


Figure 8.1: Model of FEDS' case design

9. Power Supply

When it comes to portable devices, power consumption is one of the most important aspects that can make or break any device. As an emergency detection device, the fall detector has to be running for extended periods of time. For our POC design, the Raspberry Pi board with 802.11n WiFi adapter will be used; it requires a 5V input voltage and 590mA (490mA for the Raspberry with 100% CPU load and 100mA for the Wi-Fi adapter) to maintain its operation [21][22]. The overall power consumption will increase as more sensors such as accelerometers, GPS, and gyroscopes are added. The POC design will use an external battery as the power supply because the Raspberry Pi uses a micro USB port to connect to its power source. (Figure 9.1)

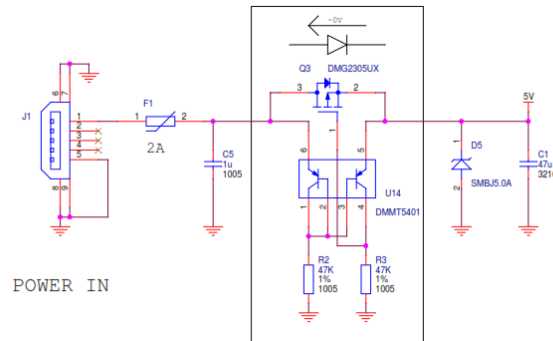


Figure 9.1: Schematic of Raspberry Pi Voltage Input Module [23]

Besides powering the Raspberry Pi and the Wi-Fi adapter, the only other devices that have heavy power consumption are the accelerometer, LED and GPS. The accelerometer requires 23 μ A in measurement mode. Since the GPS will only be on if the user is in emergency – meaning that it is not continuously turned on – the power consumption of the GPS can be ignored. The LED will use around 60mA [19]. By adding up the current consumption of Raspberry Pi and the three components, the total current consumption is only around 650mA. Research shows that an adult should have around 7-9 hours of sleep while seniors require around 8 [20]. Because of this reasoning, it was decided that the device should be able to run for around 14 hours a day. This leaves the 10 hours (24 hours – 14hours = 10hours) for ample charging time, which would be done while the user sleeps. In order for the device to run 14 hours, we need a battery that is approximately 9100mAh as shown in the equation below.

$$\text{Current Consumption} * \text{Hours of operation} = \text{Required Capacity of Battery}$$

$$650\text{mA} * 14 \text{ hours} = 9100 \text{ mAh}$$

With this information, the Aukey Amzdeal 9000mAh power bank, a lithium ion based battery, will be used as the external power supply as shown in Figure 9.2. The benefits of this battery include its low cost of \$17 CAD, relatively light weight of 259g for this capacity, and 5V output (perfect for the raspberry) with a possible 1300mA output current. Furthermore, it has a 4 LEDs to indicate the remaining charge [24].



Figure 9.2:AukeyAmzdeal 9000mAh Portable Power Bank

For the prototype stage, we may want to implement an internal LiPo battery inside our device. The advantage of the LiPo battery is that it is even lighter in comparison to the Lithium battery. However, with a LiPo battery, we may need to include a 5V DC booster converter (Figure 9.3) to maintain a 5V DC output and 2A current output, because some LiPo batteries are not able to output 5V DC.



Figure 9.3: 5V DC booster converter (Model: AiLi013)

10. Server Design

Figure 10.1 illustrates the client-server communication model diagram and the web server architecture. The application is hosted on Node JS server using Amazon EC2 (Elastic Compute Cloud) services. Between the server and FEDS's device application, they will communicate via HTTP Request/Response protocol of socket communication. At the same time, the operator's application will communicate with the server via AngularJS interface. The server will also parse the client's requests to ExpressJS where the MongoDB driver is utilized to fetch and return the requested information.

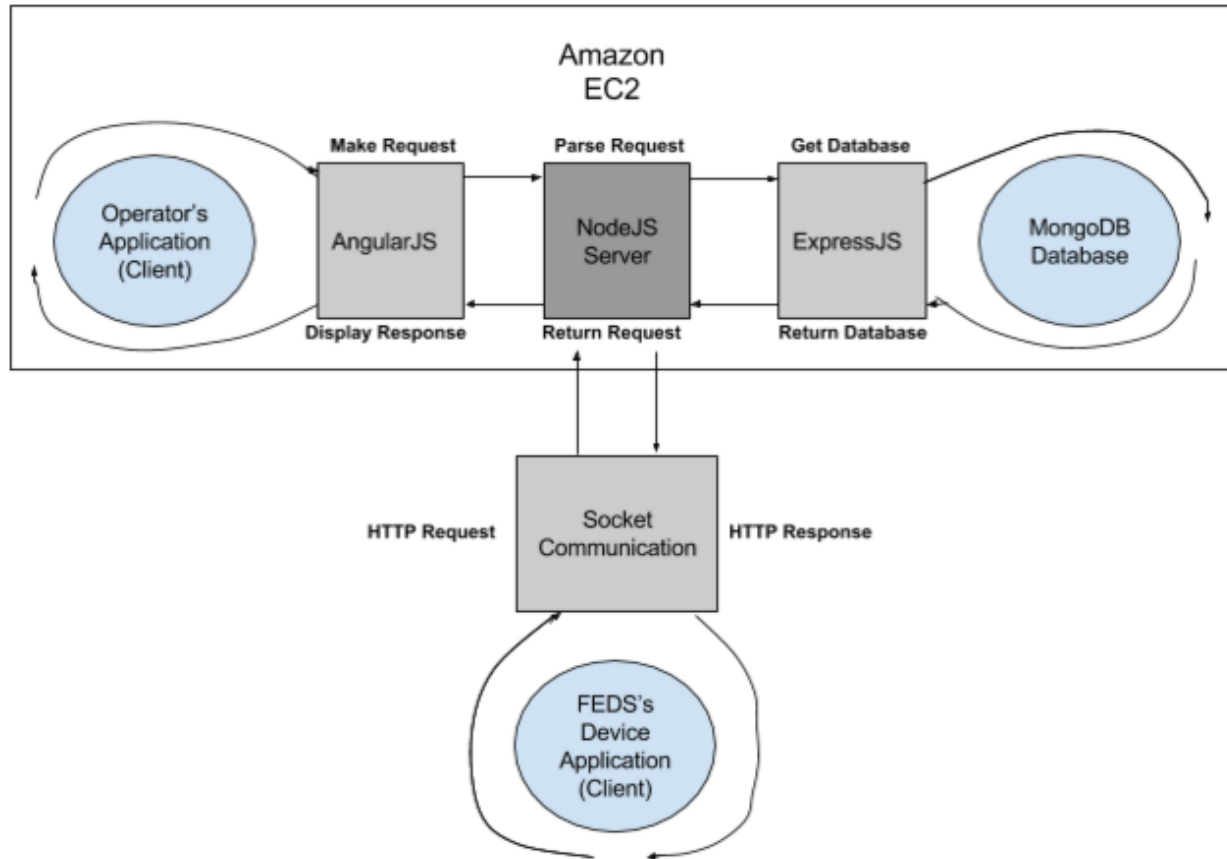


Figure 10.1: Client-server application framework

11. FEDS' Device Application Client

The FEDS's device application will be operating in the embedded system. This application will be responsible in collecting and processing the GPIO signals, then provide instructions for the next states of the device. At default state, FED's device application will set the LED colour to green, and will keep reading the accelerometer input from the GPIO to decide if its acceleration force has surpassed the threshold that has been configured. The application will also read the GPIO pin that is connected to the help button to see if the user decides to bypass the fall detection function and request immediate assistance.

If a fall is detected, the application will change to warning state and set the LED to yellow colour. At this state, the application will begin the 30 second countdown and check the GPIO input connected to the cancel button while activating the buzzer. After the countdown is completed, the application will change to emergency state and set the LED to red colour. For emergency procedures, the application will request the GPS module to collect the user's location. After that the user's device ID, status of the device and user's location will be sent to the server and the database will be updated.



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

Also, if the user requires help, he can press the help button at any time. After pressing the help button, the application will change to emergency state and set the LED to red colour immediately. Then, the application will follow the same emergency procedures as above.

If the cancel button is pressed during warning or emergency state, the application will deactivate the current state and set the device back to default state. It will also communicate with the server and the database will be updated.

12. Operator Web Application Client

The operator web application is hosted on <http://feds4life.com/public/index.html> (currently in BETA mode). The web application is separated into 2 parts: the Client Status page (shown in Figure 12.1) and Client Information page (shown in Figure 12.2).

FEDS4LIFE	Client Status	Client Add
-----------	---------------	------------

Client Status							
Device ID	Client Name	Contact Number	Location	Status	Medical History	Trigger Time	Emergency Number
1	Tang John	(604) 444-4444	null	Connect	High Blood Pressure	00:00	(604) 111-1111
2	Ho Chan	(604) 523-4135	null	Connect	Obesity	00:00	(604) 512-4124
3	Ang Kao	(778) 882-2223	null	Connect	Diabetes	00:00	(603) 333-2211
5	Bai Ci	(778) 332-2230	null	Disconnect	Diabetes	10:11	(778) 392-2222
7	Janet Mardjuki	(902) 333-2222	49.276883, -122.914845	Emergency	Obesity	03:20	(999) 112-2211
4	Ka Lee	(778) 719-1234	null	Disconnect	Osteogenesis Imperfecta	00:00	(604) 132-1234
9	Benjamin Sia	(333) 662-8261	null	Warning	Parkinson's Disease	00:10	(331) 122-1122
6	Holo Madoka	(604) 523-1241	null	Connect	None	00:00	(604) 242-3521

Copyright2015 © feds4life.com

Figure 12.1: Client Status Interface of FEDS4LIFE system



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

FEDS4LIFE Client Status Client Information

Client Information

Device ID	Client Name	Contact Number	Medical History	Emergency Number	Action
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add Contact"/> <input type="button" value="Update"/> <input type="button" value="Clear"/>
1	Tang John	(604) 444-4444	High Blood Pressure	(604) 111-1111	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
2	Ho Chan	(604) 523-4135	Obesity	(604) 512-4124	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
3	Ang Kao	(778) 882-2223	Diabetes	(603) 333-2211	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
5	Bai Ci	(778) 332-2230	Diabetes	(778) 392-2222	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
7	Janet Mardjuki	(902) 333-2222	Obesity	(999) 112-2211	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
4	Ka Lee	(778) 719-1234	Osteogenesis Imperfecta	(604) 132-1234	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
9	Benjamin Sia	(333) 662-8261	Parkinson's Disease	(331) 122-1122	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
6	Holo Madoka	(604) 523-1241	None	(604) 242-3521	<input type="button" value="Remove"/> <input type="button" value="Edit"/>

Copyright2015 © feds4life.com

Figure 12.2: Insert/edit Client Information Interface

12.1 Client Status

The client status page is used by the operator to keep track of the statuses of all the registered devices. On this webpage, the device ID, client name, contact name, location, status, medical history, trigger time and emergency number will be listed (Figure12.1). The parameters: device ID, client name, contact number, medical history, emergency number are used to clarify the user's information and is editable by the operator. The remaining columns include location, status, and trigger time; these will be refreshed based on the status of the device. There will be four possible statuses displayed on the Client Status Page: connect, disconnect, warning, or emergency. These statuses will change according to the following:

- 1. Status of device is connected**
Status changes to "Connect", location changes to "N/A", trigger time resets to "00:00"
- 2. Status of device is disconnected**
Status changes to "Disconnect", location does not change, trigger time does not change
- 3. Status of device is in warning mode**
Status changes to "Warning", location changes to "N/A", trigger time updates
- 4. Status of device is in emergency mode**
Status changes to "Emergency", location shows the current location of the user in latitude longitude format, trigger time updates



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

Once the device is in emergency mode, the operator will follow the procedures mentioned in Section 2 of this document.

12.2 Client Information page

Operators will use this page to insert, edit, or delete client information from the database. On this page, the device ID, client name, contact name, medical history, and emergency number will be included (Figure 11.2).

For adding new clients into the database, the operator will fill the information inside each text field. The text fields accept any type of input (strings, integers or symbols), which means the operator needs to keep the fields cleanly formatted themselves. After filling in the required information, the operator will click on “Add Contact” to store the information inside MongoDB. Afterwards, the client information page will automatically refresh and pull the new changes made from the database.

To edit the client information in the database, the operator will click on the “Edit” button which is located to the right of the “Remove” button. The system will then show a text field allowing the operator to change the data inside the field. After the changes are made, the operator needs to click on the “Update” button to update the database. Once again, the page will automatically refresh and pull the changes from the database. If the operator does not want to update the information after clicking on “Edit”, they have the option of clicking on the “Clear” button to clear out the text field.

For deleting the client information in the database, the operator simply clicks on the “Remove” button which is located beside the “Edit” button. After clicking on the “Remove” button, the client and their information will be removed from the database and the page will automatically refresh.

12.3 Further Design of the Operator Web Application

For the future design, the operator application should focus on increasing the security for the page. Currently, the web application is being hosted on Amazon’s cloud system, EC2, and anyone can access the operator application. To add a layer of security to the system, a login system will be implemented on the web application. Another improvement which will be made is by only allowing a certain data type within each of the client information fields. For example, instead of allowing a string to be set in the phone number field, only integers should be accepted.

13. System Test Plan

The test plan is divided into two major parts. The first part involves the testing of each



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

individual component. The second part focuses on the completed POC project which involves checking whether the device is able to properly detect a fall and if the operator application/database works accordingly.

13.1 Component Test

13.1.1 Hardware

The hardware test focuses on the individual components that are connected to the Raspberry Pi. This unit testing is necessary to ensure that each component is working properly based on FEDS requirements. Manual testing will be done on these components as described below.

Accelerometer:

To test the accelerometer, a script which utilizes the accelerometer to measure the force will display a measured value on the screen. The first test will confirm the functionality of the accelerometer. To do this, the accelerometer will be held with the x axis facing downward. The measured value on the x axis should be the force acted by gravity (mass of object*9.81). A second person will be monitoring the output to make sure the force is properly changing. A second and third test will be performed on the y and z axes.

GPS:

To test the GPS, a script such as gpsmon [26] will be executed. The device will be moved to different locations and the longitude and latitude will be displayed on the screen. To check if the location is correct, a map such as (<http://mygeoposition.com/>) will be used to compare the GPS' longitude and latitude to the individual's actual location.

Buttons:

To determine if the buttons are working properly, the button will be connected in series with a LED and resistor to the Raspberry's GPIO. A test script will be designed so that pressing a button will activate/deactivate the GPIO. The LED will then be used to determine if the button is working as intended as it should turn on and off when the button is pressed.

13.1.2 Software

MongoDB/Operator Application:

The web application will be tested by accessing the 4Life's web site, <http://feds4life.com/public/index.html> (currently in BETA). To test the operator's web application, the operator will attempt to add, edit and remove a client. The changes made should be immediately stored in MongoDB and displayed on the operator application upon a refresh. The data can also be checked with other software such as MongoChef [27].

13.2 Integrated Test



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

The integrated testing will only be conducted when the component tests are completed. These tests will focus on a couple of key ideas:

1. The device should be able to detect a fall.
2. The device should be able to push data to the server.
3. The device should be able to ignore daily activities such as walking and sitting down.
4. The device's hardware user interfaces, as mentioned in section 7 should function properly

When testing for falls, the tester will be falling onto a bed or a gym mat for safety precautions.

Test Case 1: Detecting a forward fall and checking status indicator

User Input: The user should fall forward towards a bed or mat

Expected Result: After the user falls forward, a low frequency buzz should start and the status indicator should turn yellow

Test Case 2: Detecting a sideways fall and checking status indicator

User Input: The user should fall sideways towards a bed or mat

Expected Result: After the user falls sideways, a low frequency buzz should start and the status indicator should turn yellow

Test Case 3: Detecting a backwards fall and checking status indicator

User Input: The user should fall backwards to a bed or mat

Expected Result: After the user falls backwards, a low frequency buzz should start and the status indicator should turn yellow

Test Case 4: Testing the cancel button and checking status indicator

User Input: The user should fall and activate the buzzer

Conditions: After the buzzer sounds, the user must press the button within 30 seconds

Expected Result: After pressing the cancel button the buzzer should stop and the status indicator should turn from yellow to green

Test Case 5: Testing the help button and checking status indicator

User Input: The user should press the help button when standing still

Expected Result: After pressing the help button, a high frequency buzz should start and the status indicator should turn from green to red

Test Case 6: Testing the on/off switch and checking status indicator

User Input: The user should turn the device on, wait a minute and then turn the turn off

Expected Result: After turning the device on, the status indicator should turn green.
When the user turns the device off, the status indicator light should turn off.

Test Case 7: Testing the distress signal, server connection and status indicator



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

User Input: The user should fall forward and not press the cancel button for 30 seconds after the low frequency buzzer starts buzzing

Expected Result: After 30 seconds, the warning buzzer should change from a low frequency buzz to a high frequency buzz. The status indicator light should turn from yellow to red and a distress signal shall be sent to the server.

Test Case 8: Testing the GPS/location searching

User Input: The user should fall forward and not press the cancel button for 30 seconds after the low frequency buzzer starts buzzing

Expected Result: After 30 seconds, the warning buzzer should change from a low frequency buzz to a high frequency buzz. The status indicator light should turn from yellow to red and the location of the user should be pushed to the server

Test Case 9: Testing for false alarms while walking

User Input: The user should walk normally for 1 minute

Expected Result: The device should not detect a fall and the status indicator remains green

Test Case 10: Testing for false alarms when the user sits down normally

User Input: The user should sit down on a chair in a “normal” fashion

Expected Result: The device should not detect a fall and the status indicator remains green

Test Case 11: Testing for false alarms when abruptly sitting down

User Input: The user should abruptly sit down on a chair

Expected Result: The device should not detect a fall and the status indicator remains green

Test Case 14: Testing for false alarms when lying down

User Input: The user should lie down on a bed in a “normal” fashion

Expected Result: The device should not detect a fall and the status indicator remains green

14. Conclusion

The functional specification of FEDs clearly defines the functionality and requirements of 4Life's FEDs. The requirements in each section are categorized under: A, proof of concept; B, prototype; and C, consumer production. FEDs will be built under the completion of each category beginning from A to B. The order allows the device to meet all underlying requirements as the final product is built up for commercial use.



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

To summarize, the FEDS is a fall detection system which will send an alert to an emergency monitoring team when a fall has occurred. The device works by using an accelerometer to sense a fall and the user's location is found through GPS or WiFi, which is sent to the server. The server will be hosted on Amazon's EC2 server and utilizes MongoDB to store the user's data. NodeJS, ExpressJS and AngularJS are used to allow communication between the FEDS device, database and operator application. With this device, we hope to prevent serious injuries that may occur from falls if left unattended and untreated. On top of that, we hope FEDS would provide peace of mind for the user to know that there is someone always looking out for them.



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

15. References

- [1] Injury Prevention, "The cost of fatal and non-fatal falls among older adults", 2015. [Online]. Available: <http://injuryprevention.bmj.com/content/12/5/290.short>. [Accessed:17- Oct- 2015].
- [2] [Cdc.gov, "Important Facts about Falls | Home and Recreational Safety | CDC Injury Center", 2015. [Online]. Available: <http://www.cdc.gov/homeandrecreationalsafety/falls/adultfalls.html>. [Accessed: 18-Oct- 2015].
- [3] Wi-Fi Alliance, "Certification", 2015. [Online]. Available: <http://www.wi-fi.org/certification>. [Accessed: 8-Nov-2015]
- [4] EDIMAX, "N150 Wi-Fi Nano USB Adapter, Ideal for Raspberry Pi", 2015. [Online]. Available: http://www.edimax.com/edimax/merchandise/merchandise_detail/data/edimax/global/wireless_adapters_n150/ew-7811un. [Accessed: 28Nov-2015]
- [5] adafruit, "ADXL345 - Triple-Axis Accelerometer (+-2g/4g/8g/16g) w/ I2C/SPI", 2015. [Online]. Available: <http://www.adafruit.com/products/1231>. [Accessed: 2-Nov-2015]
- [6] adafruit, "ADXL335 - 5V ready triple-axis accelerometer (+-3g analog out)", 2015. [Online]. Available: <http://www.adafruit.com/product/163>. [Accessed: 2Nov-2015]
- [7] "Data sheet", 2015. [Online] Available: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>. [Accessed: 2-Nov-2015]
- [8] Chien-Hsien Liu, "A Fall Detection System Using Accelerometer and Gyroscope Based on Smartphone", 2015. [Online]. Available: http://www.researchgate.net/publication/279533573_Fall_Detection_System_Using_Accelerometer_and_Gyroscope_Based_on_Smartphone. [Accessed: 2-Nov-2015]
- [9] Nathan Osman, "What is the maximum current the GPIO pins can output?", 2015. [Online]. Available: <http://raspberrypi.stackexchange.com/questions/9298/what-is-the-maximum-current-the-gpio-pins-can-output>. [Accessed: 3-Nov-2015]
- [10] "GPS - NMEA sentence information", 2015. [Online]. Available: <http://aprs.gids.nl/nmea/>. [Accessed: 3-Nov-2015]
- [11] GlobalTop Technology Inc., "FGPMMOPA6B GPS Module Data sheet", 2015. [Online]. Available: <http://www.adafruit.com/datasheets/PA6B-Datasheet-A07.pdf>. [Accessed: 4-Nov-2015]



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

- [12] Google Developers, “Google Maps Geocoding API”, 2015. [Online]. Available: <https://developers.google.com/maps/documentation/geocoding/intro>. [Accessed: 4-Nov-2015]
- [13] Fork Robotics, “Using the GY80 10DOF module with Arduino”, 2015. [Online]. Available: <http://www.forkrobotics.com/2013/06/using-the-gy80-10dof-module-with-arduino>. [Accessed: 4-Nov-2015]
- [14] Quoc T. Huynh, Uyen D. Nguyen, Lucia B. Irazabal, Nazanin Ghassemian, and Binh Q. Tran, “Optimization of an Accelerometer and Gyroscope-Based Fall Detection Algorithm”, 2015. [Online]. Available: <http://www.hindawi.com/journals/js/2015/452078/>. [Accessed: 5-Nov-2015]
- [15] ST, “L3G4200D Datasheet”, 2013. [Online] <http://www.forkrobotics.com/wp-content/uploads/2013/05/L3G4200D.pdf?189db0> [Accessed 3-Nov-2015]
- [16] Pulesensor.com, “Pulse Sensor Speaker Tutorial”, 2015. [Online] <http://pulesensor.com/pages/pulse-sensor-speaker-tutorial> [Accessed 2-Nov-2015]
- [17] Continental Inc., “Basic advantages of LED Light”, 2015. [Online] <http://www.continental-lighting.com/led-basics/advantages-disadvantages.php> [Accessed 3-Nov-2015]
- [18] APS, “Stop On Red! The Effects of Color May Lie Deep in Evolution...” June 7, 2011. [Online] <http://www.psychologicalscience.org/index.php/news/releases/stop-on-red-a-monkey-study-suggests-that-the-effects-of-color-lie-deep-in-evolution.html> [Accessed 7-Nov-2015]
- [19] Adafruit, “LED Current Draw”, 2015. [Online] <https://learn.adafruit.com/rgb-led-strips/current-draw> [Accessed 9-Nov-2015]
- [20] National Sleep Foundation, “How Much Sleep Does We Really Need?”, 2015. [Accessed: 10-Nov-2015]
- [21] Raspberrypi.org, “Power Supply”, 2015. [Online] <https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md> [Accessed: 7-Nov-2015]
- [22] GitHub, “Power Consumption”, 2015. *Online+. Available: <https://github.com/geerlingguy/raspberry-pi-dramble/wiki/Power-Consumption>. [Accessed: 19- Oct-2015]
- [23] Adafruit, “Raspberry Pi Power Supply”, 2015. [Online] <https://learn.adafruit.com/introducing-the-raspberry-pi-model-b-plus-plus-differences-vs-model-b/power-supply> [Accessed: 9-Nov-2015]



8888 University Drive,
Burnaby, BC, V5A 1S6
604-446-6738

[24] Aukey, "Aukey PB-N12 Dual Port USB External Battery Pack with AI Power", 2015. [Online]
<http://www.aukey.com/product/dual-port-usb-external-battery-pack-ai-power> [Accessed: 9-Nov-2015]

[26] Eric S. Raymond, "GPSPMON — Real-time GPS Packet Monitor and Control Utility", 2015
[Online]<http://www.catb.org/gpsd/gpsmon.html> [Accessed: 8-Nov-2015]

[27] 3T.io, "MongoChef for MongoDB 3.0", 2015. [Online]<http://3t.io/mongochef/> [Accessed 5-Nov-2015]