Applied Information Technology And Computer Science Vol. 4 No. 1 (2023) 327-345
© Universiti Tun Hussein Onn Malaysia Publisher's Office

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Design and Development of Go-Ambulance System based on IoT Technology

Yeong Mang Sen[1], Firkhan Ali Hamid Ali[1]*

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/aitcs.2023.04.01.020
Received 19 June 2022; Accepted 29 September 2022; Available online 30 June 2023

Abstract: Nowadays, the contact process to hospital and the time taken for ambulance to reach the car accident scene is very critical if the caller and location information received are inaccurate that cause the delay in the arrival of ambulance service and causes the loss of human life. The main objective of this system is to design an IoT-based system that can help the hospital and driver's parent to get the driver's information such as the driver's current location and phone number during a car accident occur. Based on waterfall methodology, this system is developed using Arduino IDE, using C programming language and ThingSpeak as the database. Sensor-embedded objects and devices connect to the Internet of Things (IoT) platform, which integrates the data from a variety of devices and to sending the SMS and GPS location. The IoT system is believed to help in saving lives and reducing time wastage.

Keywords: IoT, SMS, GPS

1. Introduction

An ambulance is a vehicle which is devoted to transfer of individuals suffering from any disease or that they are wounds. This meant that transport usually carries remedies and various devices and products used for the emergency healthcare. In general, ambulances are responsible for the transfer of people to and from hospitals or other care centers. Once they are not seemed to be making a visit and not making a trip, they have a tendency to stay and remain on a base waiting for a call to carry the emergency patients.

Emergencies such as road accidents and disease attacks which require the ambulance services. Before the emergency department sent an ambulance to the scene, there are some problems that often arise. Nowadays most of the cars are lack some safety and rescue systems such as GPS system and collision sensor to detect any car accident. When a car accident occurs, the callers are unable to provide correct scene information. It will cause the delay in the arrival of ambulance to the accident spot and causes the loss of human life. Besides that, if the passenger and driver are unconscious in a car accident, they are incapable of saving themselves and seeking help from the hospital.

*Corresponding author: firkhan@uthm.edu.my

The main objective of this system are to design an IoT-based system that can help the hospital and driver's parent to get the driver's information such as the driver's current location and phone number during a car accident occur, to develop a smart IoT-based system that can detect the vehicle's collision during car accident and to test the functionalities of the IoT-based system mainly concentrates on mainline functions, basic usability, accessibility and the error conditions. Scope of the projects are to help administrators from the hospital to be able to provide ambulance services to victims who need this service, driver able sending an alert message when facing an accident and the data obtained from the hospital will be stored in the system.

The expected result of this project is this system should solve all the problems stated in the problem statement. The main expected outcome of this system developed is to facilitate the process for drivers to call an ambulance by providing the location to the emergency department and help the emergency department make a decision. Secondly, the system can be developing with an Internet of Things (IoT) functionality system. Thirdly, the data of the system should be able store and record on the database system. This system should be able to implement an alternative medium for safety and healthy. This app is believed to help in saving lives and reducing time wastage.

Therefore, a smart Internet of Things system named GO-AMBULANCE with A Smart IoT-Based System is proposed to solve the problems mentioned above. This system was developed with the aim of accelerate the process of contacting and delivering ambulance services to the victims. In addition, this system contains other functions such as detecting collision of the car, provide current car location, and so on to the users of this system. The proposed of this system is able to solve the problem and can provide a comfortable life of the users.

2. Related Work

2.1 Automatic Ambulance System Using IoT

Automatic Ambulance System using IOT is an IOT system which developed from India. This system has three architecture of the proposed models such as ambulance module, cloud or Wi-Fi module and hospital module. In the ambulance module, this model and architecture consists of the Raspberry Pi with the sensor that will be used inside the ambulance. The sensors are used to measure the patient's body parameters. Heart rate, blood pressure, temperature, cholesterol levels and blood glucose are among of the parameters. A camera module is also included in this device, which captures the treatment image of the patient. After this ambulance module, the data is uploaded to the cloud.

Besides that, the cloud or Wi-Fi module consists of two cloud services which are Dropbox and ThingSpeak. The ThingSpeak cloud platform allows user to upload the graphical representation data and numerical data. Dropbox is used to upload the image. For the hospital module, the data uploaded to the cloud is monitored by the doctor in this module. The doctors can distinguish the severity of wounds by the red color detected images. A basic application can download the data from the cloud makes up of the hospital module. Doctors can prepare for urgent medical response after receiving the data from the cloud.

2.2 IoT Smart Ambulance System with Patient Health Monitoring

The IoT Smart Ambulance System with Patient Health Monitoring was developed from India. The progress of work in an IoT-based system is determined by three systems which are transmission, cloud and sensor work. First and foremost, the sensor network is the initial stage in patient monitoring and data collection. Next, the gateway system is a network of continuous connections between the cloud-based system and the sensors. The death rate of people per year, or per day, or per hour is a major concern throughout the world.

As a result, they propose a paradigm in which the patient can check his or her own ECG and heart beat and have the result transferred to the cloud promptly. After that, doctor will use these reports within a shorter time and these can save both patients and doctors' time. Sensors are providing the real-time data, so they don't have to wait for reports. For those living in rural regions, the model is particularly beneficial.

IoT allows data or patient reports to be sent to physicians with time and date using GSM/3G/4G technology. This proposed technique can be used to any sort of individual, whether or not they are suffering from an illness. As a result, they can check it on a frequent basis since people are more concerned with disease early prevention. All reports will be captured in real time, as well as live video recordings. IoT devices generate a significant amount of data and information. By recoding and collecting the patient monitoring data, these health care services are becoming better and cheaper.

2.3 IoT Based Smart Ambulance Monitoring System with Traffic Light Control

This article outlines a solution to the issues traffic jam of an ambulance and which can be solved by clearing the best path for an ambulance travelling. The nearest traffic control room will be notifying when there an ambulance is approaching. This article also indicated a health monitoring system in which the current health data of patients in ambulances are recorded and sent to hospitals before they arrival.

An android application is being developed for traffic control that can be utilized by both the control room and the ambulance. This application is functioning for the traffic control room and ambulance to monitor the traffic condition in a specific region. The ambulance driver can request a best path when there has a high density of traffic by using this application. The traffic lights can be control by traffic control room based on the ambulance's request and destination.

The parameter such as the patient's body temperature and heart beat rate are monitored using a lm35 temperature sensor and heart rate sensor respectively. These data are collected and sent to the hospital via GSM using a Raspberry Pi.

2.4 Comparison with the Existing System

This Table 1 shows a comparison between similar IoT system that have been selected for study, namely Automatic Ambulance System Using IoT, IoT Smart Ambulance System with Patient Health Monitoring, IoT Based Smart Ambulance Monitoring System with Traffic Light Control and Go-Ambulance with A Smart IoT-Based System. These IoT system were selected based on functions and features that have similarities to the Go-Ambulance with A Smart IoT-Based System. Among the functions that have been listed are GSM module, GPS module, vibration sensor, temperature sensor, heart rate sensor, rescue push button, camera module and database.

Based on the table comparison above, the Automatic Ambulance System Using IoT does not have GSM module, GPS module, vibration sensor and rescue push button. The IoT Smart Ambulance System with Patient Health Monitoring does not have vibration sensor and camera module. The IoT Based Smart Ambulance Monitoring System with Traffic Light Control does not have vibration sensor, rescue push button, camera module and database.

Table 1: Comparison Between Similar IoT System

Function/ System	Automatic Ambulance System Using IoT	IoT Smart Ambulance System with Patient Health Monitoring	IoT Based Smart Ambulance Monitoring System with Traffic Light Control	Go-Ambulance: A Smart IoT-Based System
GSM Module	Not Available	Available	Available	Available
GPS Module	Not Available	Available	Available	Available
Vibration Sensor	Not Available	Not Available	Not Available	Available
Temperature Sensor	Available	Available	Android	Not Available
Heart Rate Sensor	Available	Available	Available	Not Available
Rescue Push Button	Not Available	Available	Not Available	Available
Camera Module	Available	Not Available	Not Available	Not Available
Database	ThingSpeak, Dropbox	MySQL	Not Available	ThingSpeak

3. Methodology/Framework

For the purpose of developing our program and system, we have chosen the Waterfall Development Method to attain our purpose. The Waterfall Development Method works just as the name suggest, the SDLC (System Design Life Cycle) will go from one phase to another in a strictly set manner, without the chance to revert back to phase before the current phase. This method is very strict, and it does not allow any chance for redesigning and changing the developed system. This method suits us because the system we are developing has a lot of design example in before and the design for our current system has been determined. Not much change can be made about the program, and such, a strict yet simple methodology like this will work well.

3.1 Planning Pahse

In the planning phase, the contents of the proposal which are problem statement, objectives, project scope, expected result and project significance are discussed. The work plan of the developing the proposed system is modified in this phase to ensure that the proposed system is completed within the time frame. The Gantt chart of this project is to estimate the period of each phase for developing the proposed IoT system. In order to gather more data for the proposed system, research studies on similar information and knowledge from online resource and journal article on the relevant topic are undertaken. Comparison between the existing IoT systems and the proposed system is conducted to analyze the functions provided in each system and the limitation of the existing systems. This will help to improve the proposed system.

3.2 Requirement Analysis Phase

The second phase is to identify the project scope of the Go-Ambulance with a Smart IoT-Based System. At this phase, the established requirements are collected. The data of user and system requirements are collected through a survey in term of questionnaire. All the requirements will be collected, analyzed and documented for further use. This can ensure the proposed system meets the user's needs and avoid the lack of functionality of the proposed system.

Table 2: Functional Requirement for The System

Module	Description
Register	• All new user is required to register their account before login into the Go-Ambulance System. • Systems alert for any invalid input
Login	• User already have an account can login directly through this page. • User inputs valid ID and password • System alert for any invalid input
Reset Password	• User able to reset a new password after register a new account
Rescue Push Button	• A direct SOS message and call to the hospital for ambulance request
Vibration Sensor	• Detection the collision impact within a car accident
GSM	• Sending the message to the related person
GPS	• Used to determine position, time, and speed if you're travelling
Database Management	• Collected and store the data obtained from the sensor

3.3 Design Phase

In this phase, the requirements recorded in the requirement analysis phase are examined and system designing of Go-Ambulance with a Smart IoT-Based System was prepared. In terms of the user interface (UI) design, it is determined by considering the arrangement of the display the components such as functionality and contents display. The appearance of the system application should be easy to understand and user friendly. This phase can make sure that all requirements and everything related had be suitable for system. The use case diagrams, sequence diagram, activity diagram, class diagram and architecture diagram will be designed and created in the design phase.

3.3.1 System Architecture Diagram

Figure 1 shows the system architecture diagram of Go-Ambulance with a Smart IoT-Based System. The system architecture consists of several components which are vibration sensor, push button, WEMOS D1 R1 ESP8266, GPS NEO-6M and GSM SIM900A. ThingSpeak is used as a database and to store data and generate graphs.

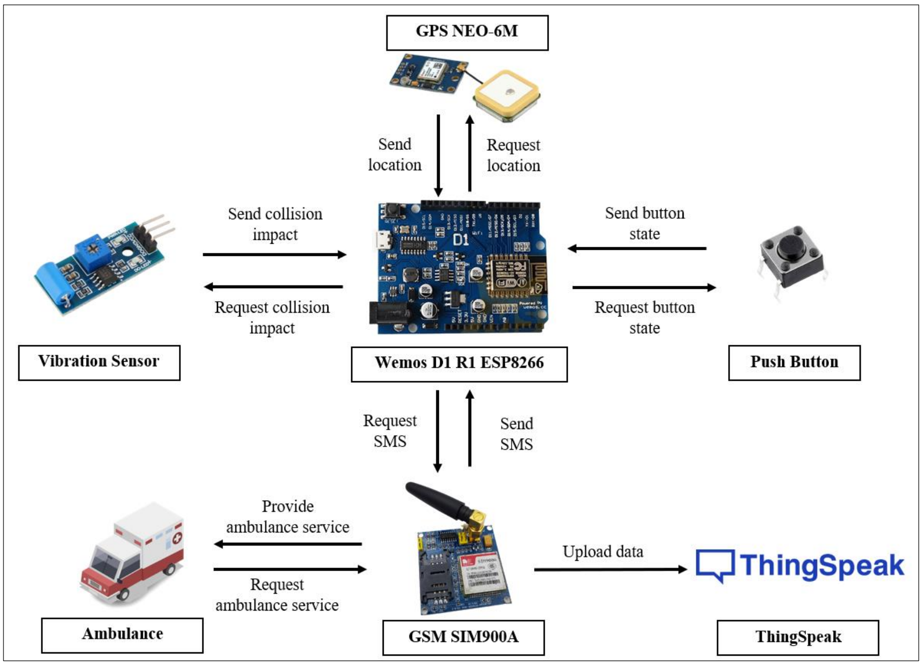

Figure 1: System Architecture Diagram

3.3.2 Use Case Diagram

Figure 2 shows the use case diagram for the Go-Ambulance with a smart IoT-based system. The figure presents the user, vibration sensor, push button, GSM, GPS and ThingSpeak that act as actor to make the preferred action in this system. There are total seven of the use cases for this system which includes register account, login, reset password, check vehicle log report, receive notification, ambulance service and view graph. The GSM, GPS module, vibration sensor and push button that embedded inside a vehicle will standby and get ready to detect the car accident. Data collected from the sensor will be sent to the Arduino Wemos D1 R1 ESP8266 board. The Arduino board will send it to the cloud by Wi-Fi and the cloud will alert the user's parent and hospital though the mobile phone. On the other hand, admin can view the report and maintain the system. After getting the notification, the hospital will provide the ambulance system to the incident.

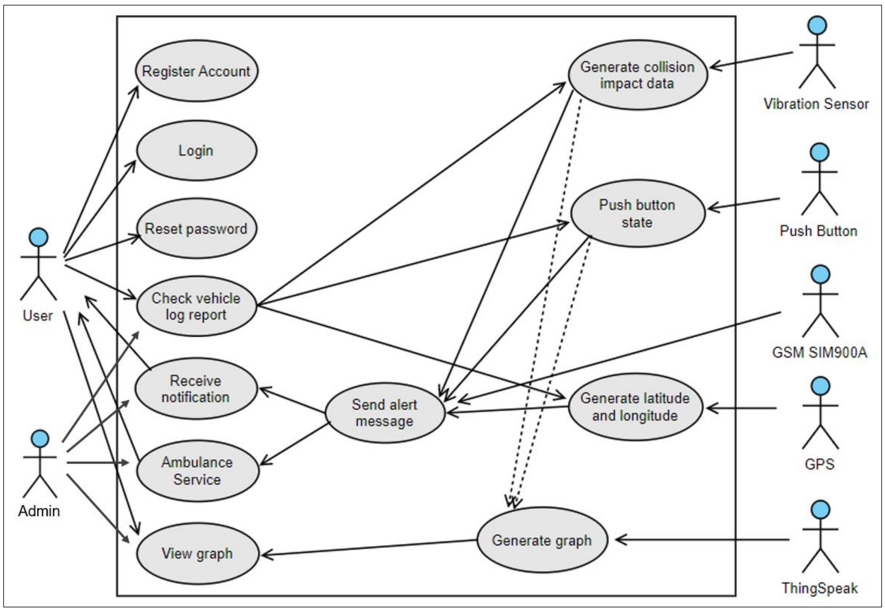

Figure 2: Use Case Diagram

3.3.3 Sequence Diagram

Figure 3 shows the sequence diagram for register account. User can register a new account with entering their username, email, and password. Then, the registration detail will be store and an account registration verification is performed to identify successful or failed account registration processes.

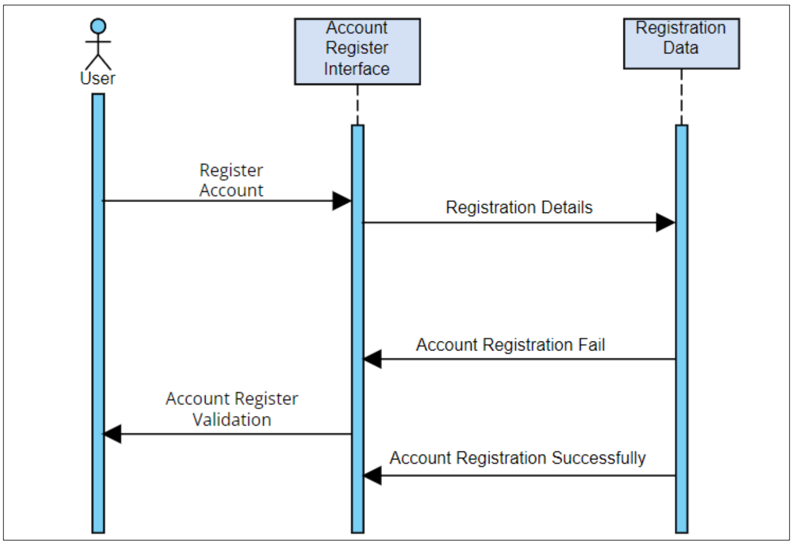

Figure 3: Sequence Diagram for Account Registration

Figure 4 shows a sequence diagram for the Login use case. Users can login using the email and password that they have registered. Next is the process of verifying user information in order to obtain permission to access the application.

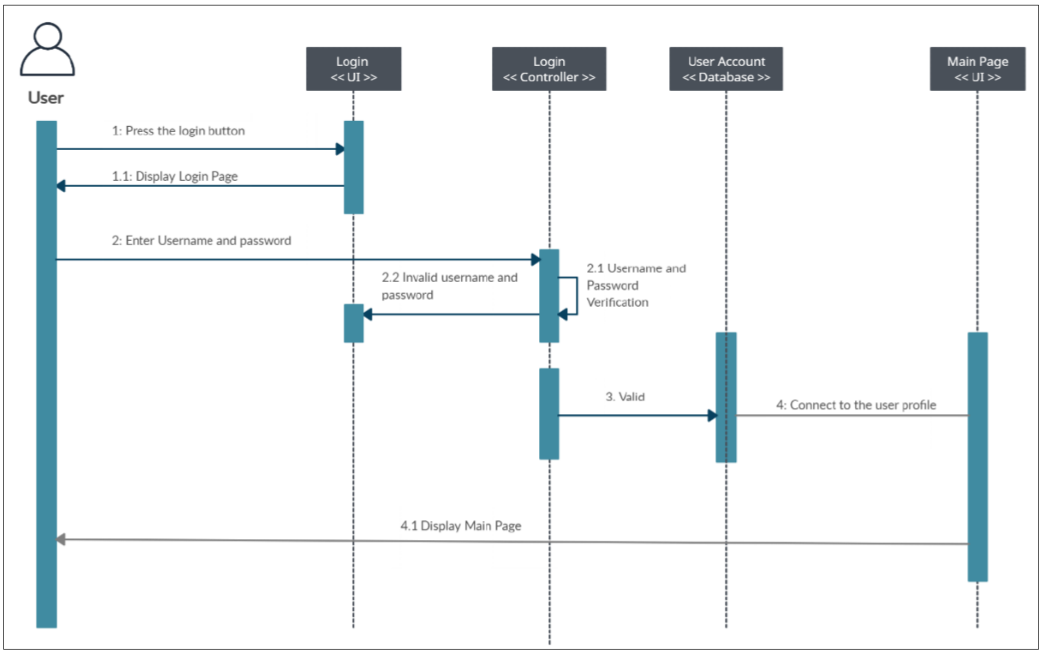

Figure 4: Sequence Diagram for Login Page

Figure 5 shows a sequence diagram for the reset password use case. Users can reset the password using the email that has been registered. A notification to reset the new password will be sent to the email registered in the account registration process.

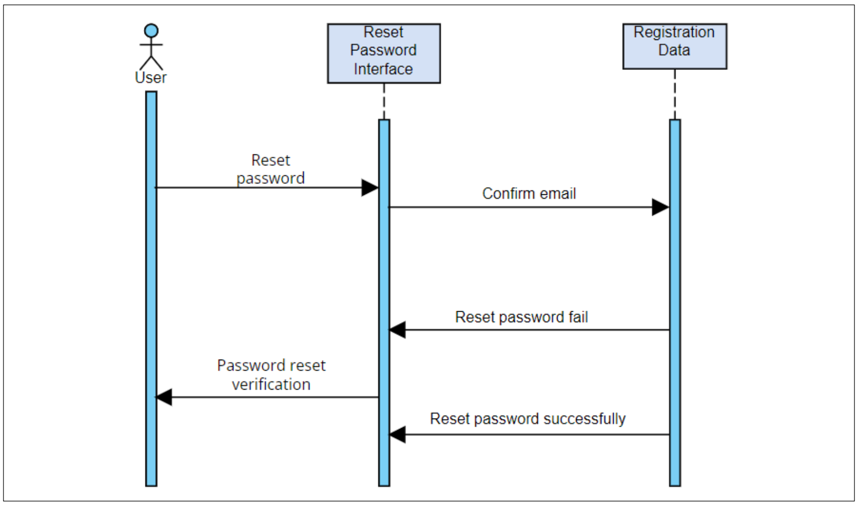

Figure 5: Sequence Diagram for Reset Password

Figure 6 shows a sequence diagram for the check vehicle log report. User can view the vehicle condition and car sensor reading from time to time.

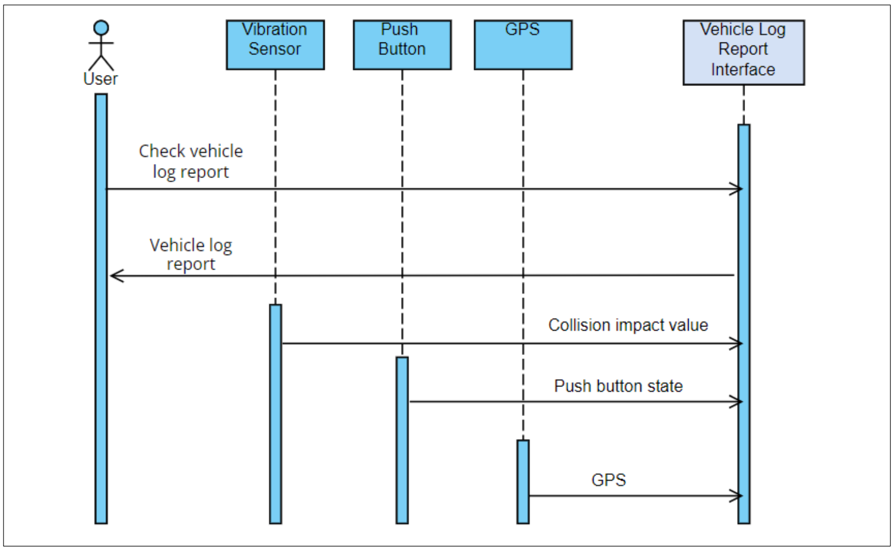

Figure 6: Sequence Diagram for Check Vehicle Log Report

Figure 7 shows a sequence diagram for the Receiving Notification use case. Users can receive notifications when the when the collision impact value is greater than threshold value or the push button is been pressed. The notification sent to the user with a message that contain GPS latitude and longitude by GSM module.

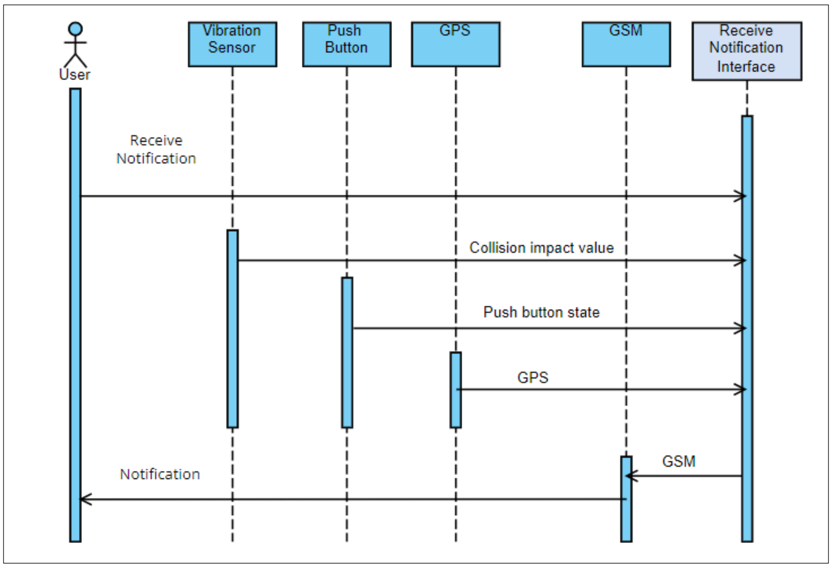

Figure 7: Sequence Diagram for Receive Notification

Figure 8 shows a sequence diagram for the Ambulance Service use case. Users can get an ambulance service when the when the collision impact value is greater than threshold value or the push button is been pressed. The notification sent to the hospital with a message that contain GPS latitude and longitude by GSM module.

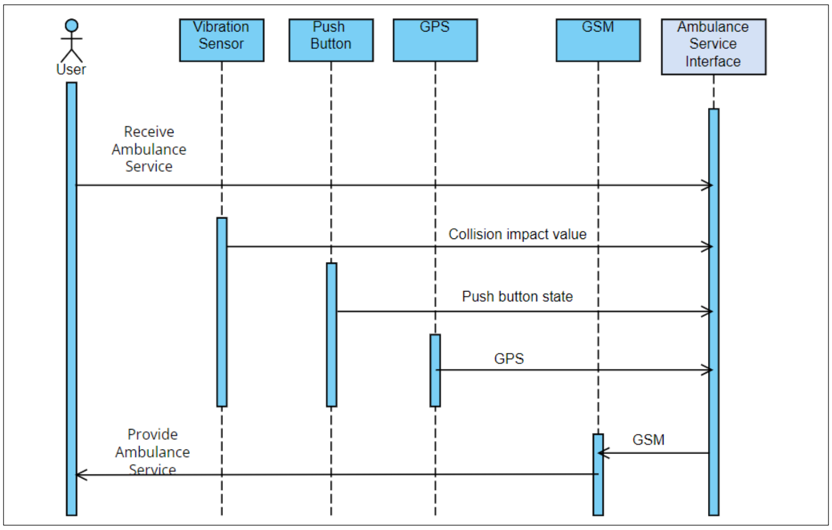

Figure 8: Sequence Diagram for Ambulance Service

Figure 9 shows a sequence diagram for the View Graph use case. Users can view collision impact graphs and push button state graphs via the View Graph function which generate by ThingSpeak. Data readings from the vibration sensor and the push button were used to generate the ThingSpeak graph.

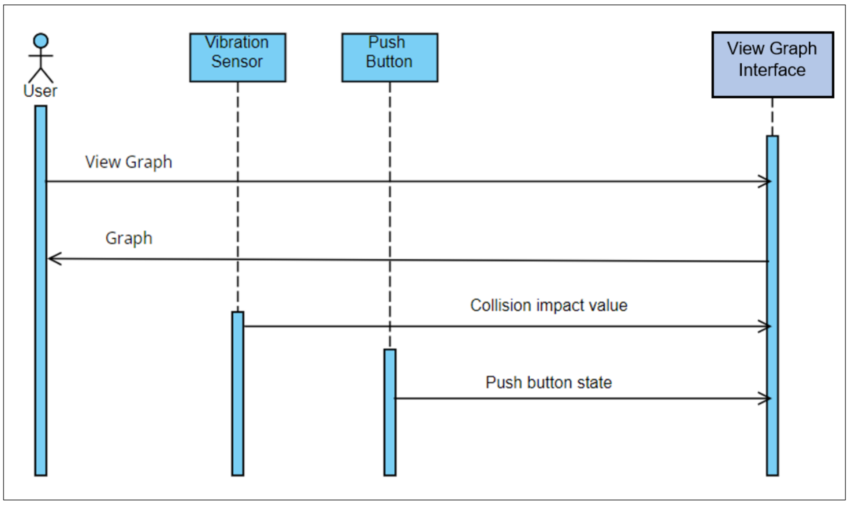

Figure 9: Sequence Diagram for View Graph

3.3.4 Activity Diagram

Figure 10 shows an activity diagram that can be performed by the user and admin. There are eight activities that can be done by the user and admin, which are register account, login, reset password, view graph, view collision impact, view button state, view ambulance and view patient. The first activity is the account register activity. New users need to register an account to access the application. Upon successful account registration, the user will login to the application. Once they successfully logged in, users will be able to access system functions as state above.

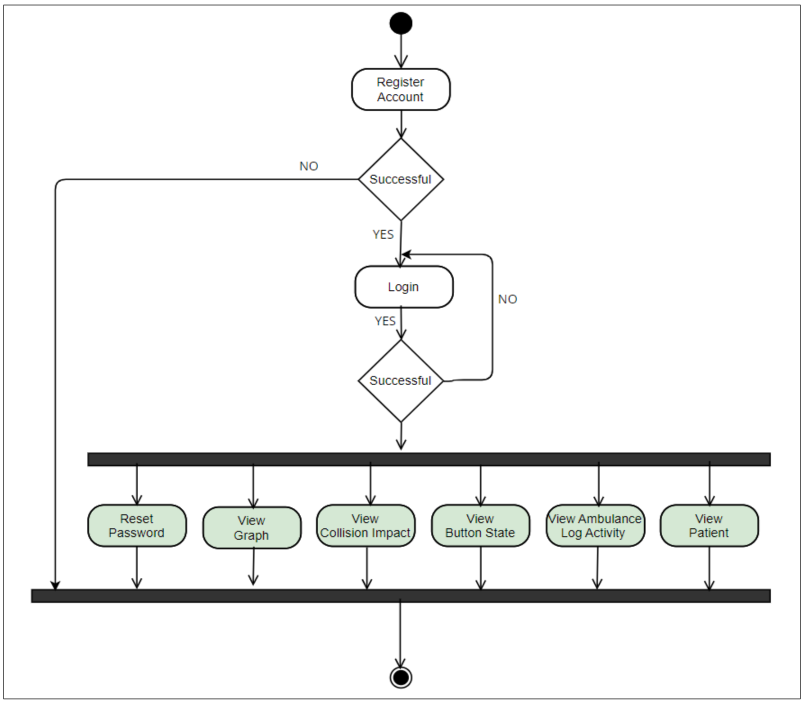

Figure 10: Activity Diagram of Admin and User

Figure 11 shows the activity diagram of the vibration sensor and push button. It starts with a vehicle engine started. All the sensors are prepared to get ready for each function. There are two sensor data which are vibration sensor and push button. If the vibration sensor produces a collision impact value that greater than the threshold value or the push button is pressed, the process of collecting the longitude and latitude from the GPS module is started. This data is combined and send in form of alert message to the user's parent and hospital by GSM module. The ambulance service is provided once they receive the alert message. All the data generate by the sensor has been collected and recorded to ThingSpeak.

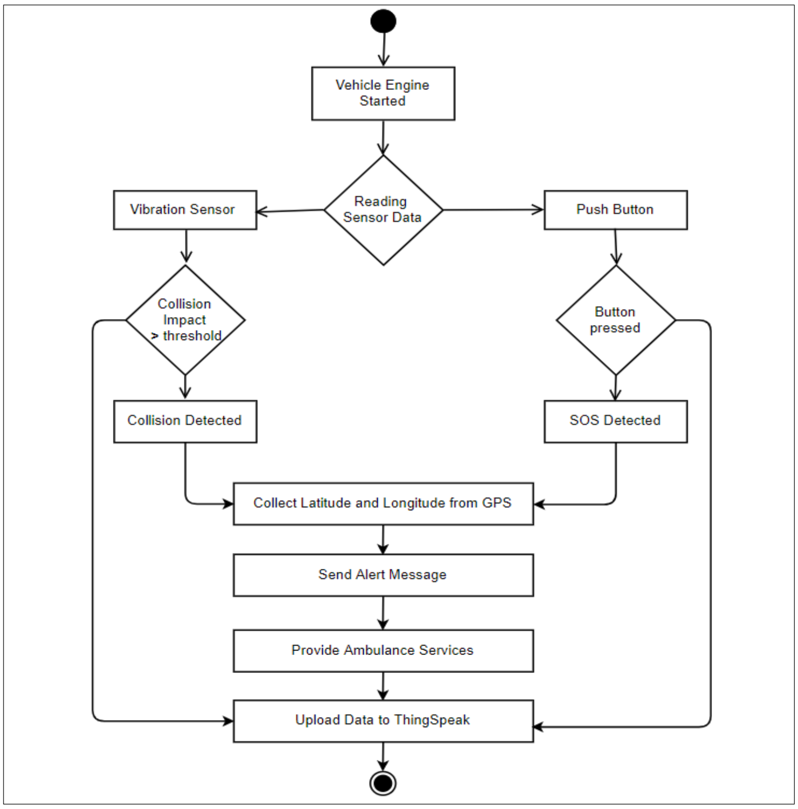

Figure 11: Activity Diagram of Vibration Sensor and Push Button

3.3.5 Class Diagram

A class diagram is a type of static structure diagram that describes a system by showing the system classes, properties design, operations (or methods), and relationships between objects. Figure 12 shows the inter-class relationships for the Go-Ambulance with a smart IoT-based system.

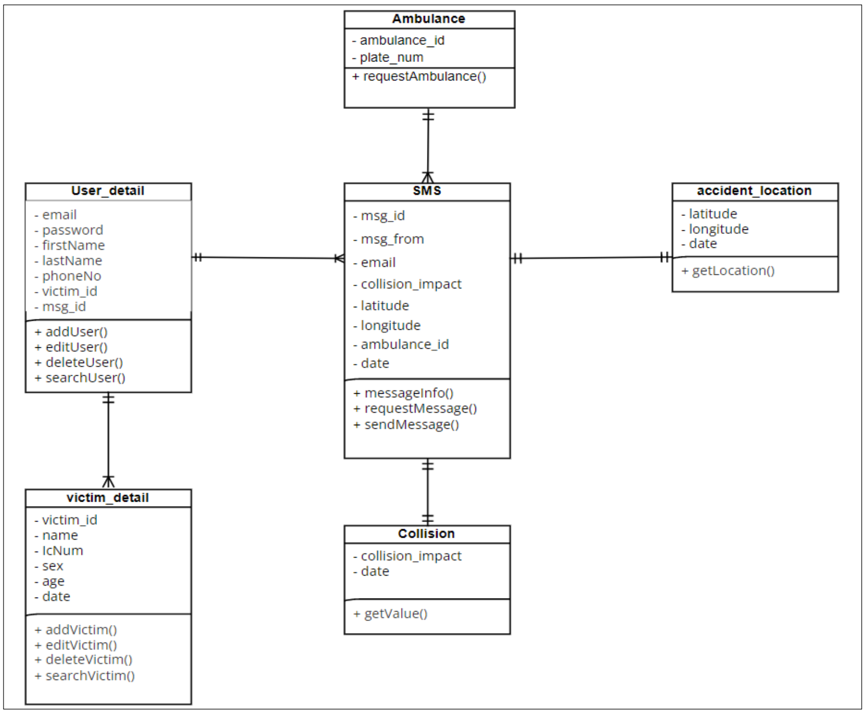

Figure 12: Class Diagram

3.4 Implementation Phase

Implementation is the phase that can execute our plan, model, design, algorithm and coding. All the software and hardware requirement need prepared before starting the implementation. For the Go-Ambulance with a Smart IoT-Based System is developed using Arduino IDE, Arduino Wemos R1 D1 ESP8266, ThinkSpeak, vibration sensor, GPS NEO-6M, GSM SIM900A, buzzer and push button. Figure 13 shows the schematic diagram of Go-Ambulance IoT system.

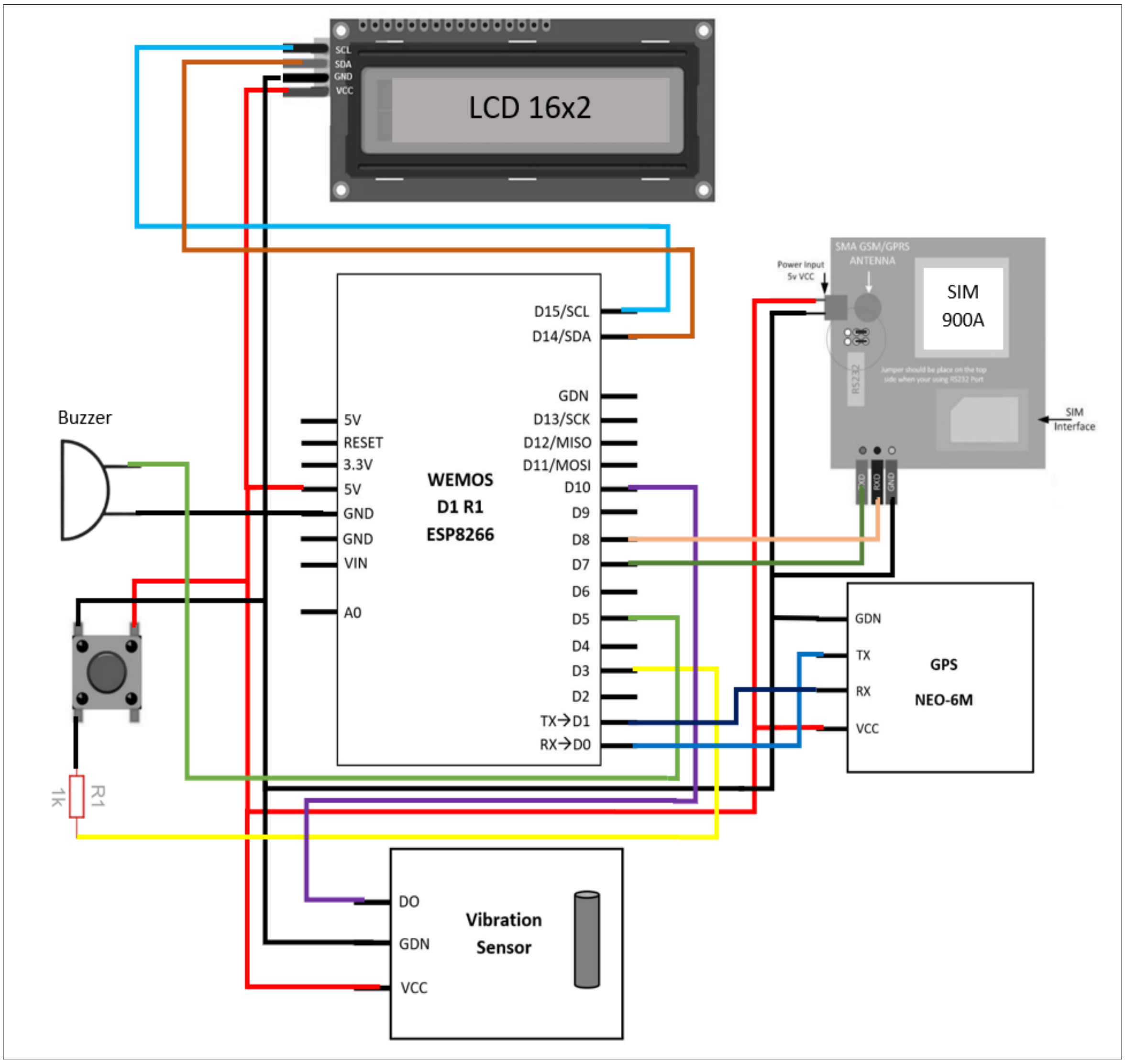

Figure 13: Schematic Diagram

3.5 Testing Phase

After the implementation of system application was completed, the IoT system will be evaluated and tested to seek whether the system have problem on its debugging and performances. Any bugs will be solved and fixed through this testing phase. The testing phase is important to ensure that all the tools will function well and provide the stability for the system which to determine the application will achieve the requirements of the project or not.

Table 3: Result of Test Plan

Test	Expected Result	Actual Result
Register account for new user	New user registration successful	Pass
Log in to a registered account	Users can login the account using the username and password that have registered	Pass
Reset password	The new password can be reset	Pass

Test	Expected Result	Actual Result
Knocking on vibration sensor	The sensor is to detect the collision impact when there has collision	Pass
User press on push button	System sent a SOS message to hospital	Pass
Click on the GPS address from the message	The current location of accident will display on the Google Map or Waze	Pass
Click on Report	Collision impact value and push button state of vehicle displayed in the graph form	Pass
Send notification	Send a notification when the vehicle's collision impact greater than threshold or push button is pressed	Pass

4. Results and Discussion

Go-Ambulance with a smart IoT-based system has successful developed and functioning according to the plan. In this section, further discussion about the implementation of this project.

4.1 Testing for Go-Ambulance System

Figure 14, Figure 15 and Figure 16 show the system experiments performed to test the functionality of each module. Among the experiments conducted were new user registration, new password reset, using new password to login, edit profile, view sensor graph and apply ambulance service. To register a new account, users need to enter their name, email, birthday, gender, IC number, phone number, address and profile picture, then click on "Register" button. Once the new account registration is successful, the user is allowed to access the system. To reset a new password, the user enters the old password and enter a new password where the old password is "1234", then click on 'Reset Password'. For report display, graphs are generated using ThingSpeak. Data from the sensor readings were used to generate two graphs, namely the collision impact of vehicle and the button state of push button. User also can view and the ambulance service of the hospital. Figure 17 shows the alert message send from the car accident's victim. In the message, hospital or victim's parent can view victim information and their current location via google map link that are provided in the message.

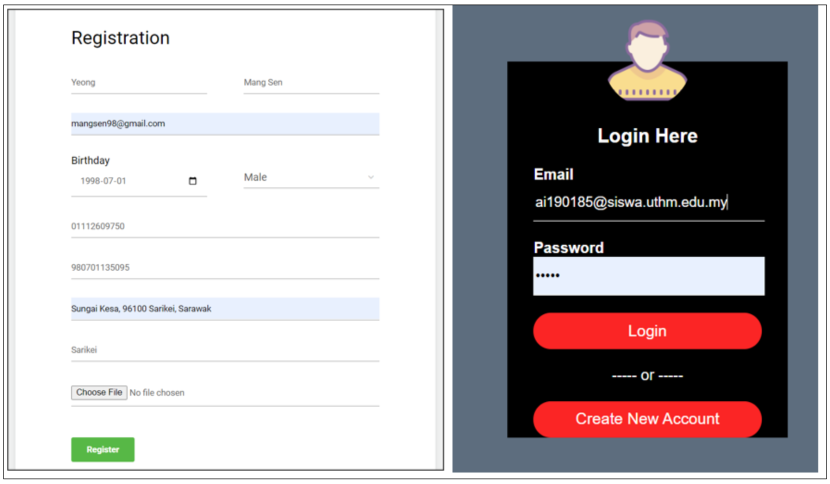

Figure 14: Testing for Account Registration and Login

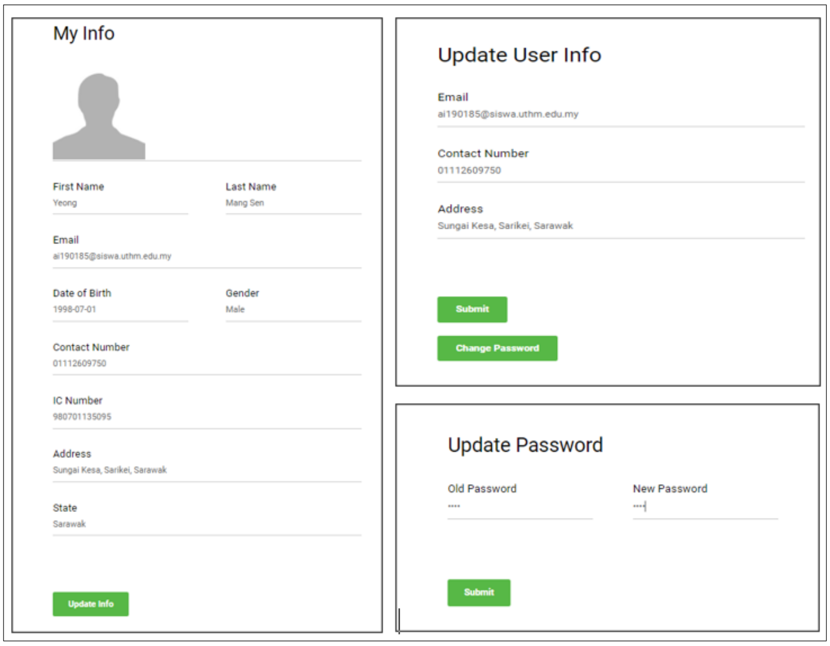

Figure 15: Testing for Change Profile Information and Reset Password

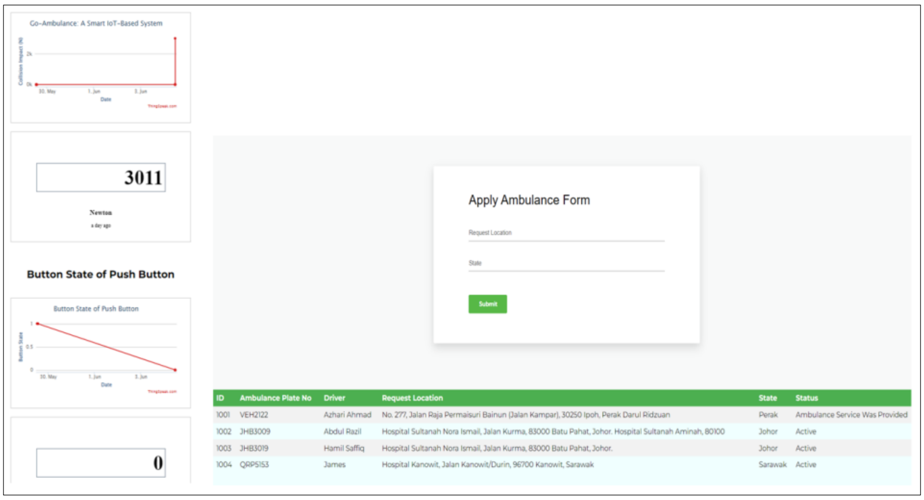

Figure 16: Testing for Report and Ambulance Status Module

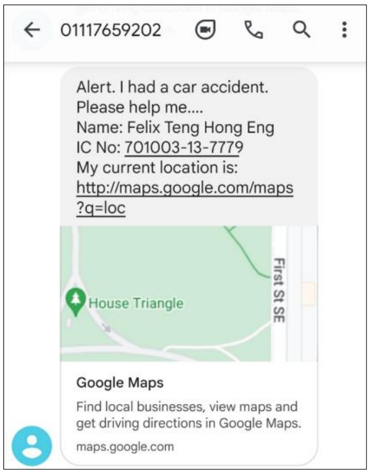

Figure 17: Testing for Alert Message

Figure 18 shows the sensor simulation. When the vehicle is started, the sensors are standby for their function. The LCD screen now is displaying “Sensor Standby” and the Wemos D1R1 ESP8266 is starting to connect ThingSpeak via hotspot.

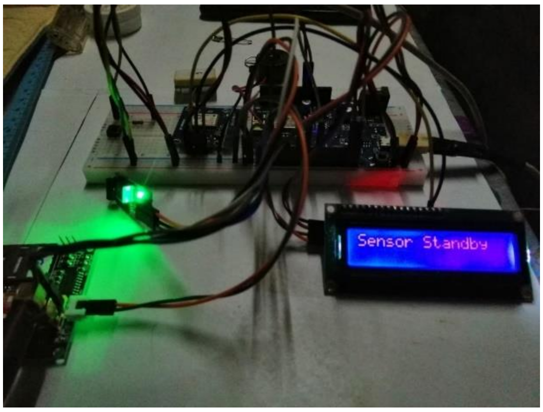

Figure 18: Sensor Simulation (Standby Mode)

Figure 19 shows the sensor simulation for collision impact detection. The LCD screen will display the collision impact value from the vibration sensor. If the collision impact is greater than the threshold for example 3000N collision impact or the push button is pressed, the sensor will assume there have a car accident. The GSM module are started to collect the incident location from GPS module and send an alert message to hospital and victim’s parent as shown in the Figure 20.

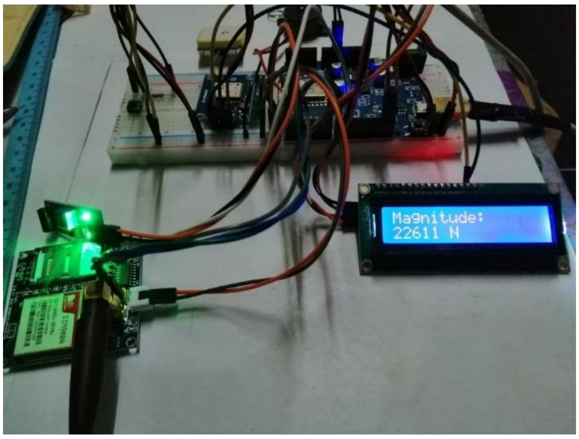

Figure 19: Sensor Simulation (Collision Impact Detection)

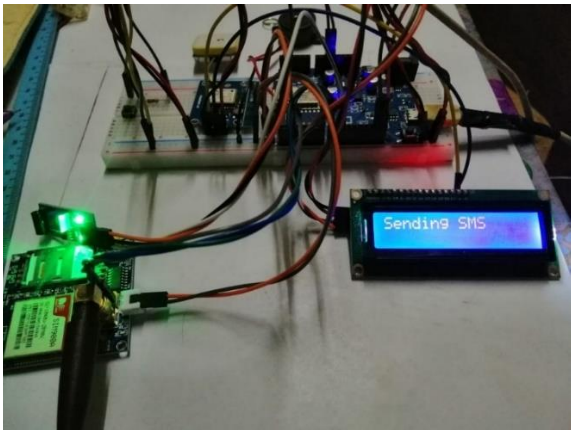

Figure 20: Sensor Simulation (Send Alert Message)

4.2 User Acceptance Test

The user acceptance test is a testing that carried out to the user for them to test the Go-Ambulance: A Smart IoT-Based System before they fill in their user acceptance questionnaire. There are about 10 respondents selected from the Sarikei community to test the system and fill in the questionnaire. This testing is important to know what the system will improve from the user's point view and their suggestion later can be used for the future improvement.

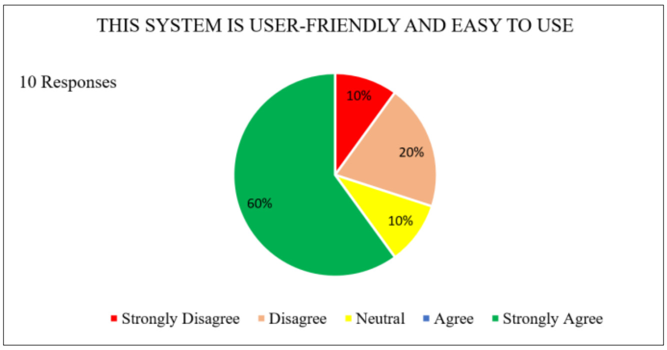

Figure 21: User Acceptance Test on easy to use and user-friendly system

Figure 21 shows the user acceptance on easy to use and user-friendly system. From the questionnaire given to the respondents, all the result was recorded and presented in the pie chart. From the result, there is about 6 respondents or 60% respondents strongly agree that this system is user-friendly and easy to use. There are 2 respondents or 20% respondents disagree that the system is user-friendly. The number of respondents for strongly disagree and neutral are the same which is 1 respondent or 10% respondent respectively.

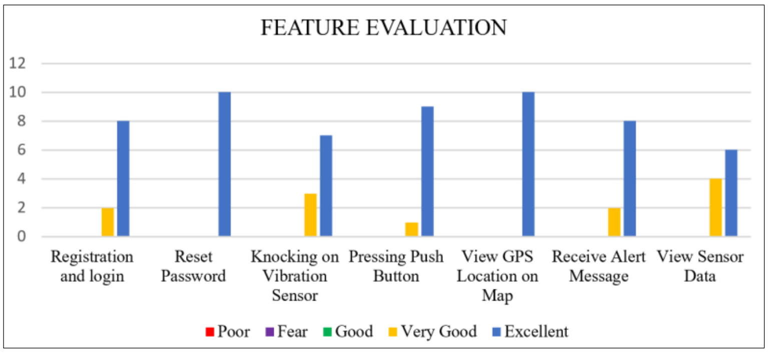

Figure 22: User Acceptance for IoT System Feature Evaluation

Figure 5.22 shows the user acceptance for IoT system feature evaluation. From the questionnaire given to the respondents, all the result was recorded and presented in the bar chart. From the bar chart, most of the people are satisfied with the feature that are stated.

5. Conclusion

In brief, Go-Ambulance with a smart IoT-based system has successfully developed. Some advantage, limitation and future improvements of this IoT system have been identified.

The advantage of the proposed IoT system is relatively cost effective as these materials are also relatively easy to obtain. The IoT system can provides an alert message and the current location of the vehicle collision which enhance the emergency services to locate the premises. It can save the victims at the right time. Besides that, the system requirements of the proposed IoT system are minimal and it can be installed in various types of vehicles.

Although Go-Ambulance with a smart IoT-based system has been successfully developed and the objectives has achieved, but there are still some limitations existed on this IoT system. The IoT system does not work without network. The GPS NEO-6M module sometimes needs 15 to 30 minutes to connect to the signal and the GPS module will lost signal when inside a house or under a roof. Besides that, the sim card should always have balance and active, if not it will be unsuccessful sending an alert message to hospital for requesting an ambulance service.

In future works, a few improvements can be done to enhance the functionality of Go-Ambulance with a smart IoT-based system. The system can be further worked on and improved by adding camera functionality. This will help to record the accident occurrence, provide a real time video stream to emergency services, and this recorded video can help in legal and insurance proceedings with the court, police, and insurance companies. The system can be implemented with 2G, 3G and 4G/5G networks and will actively operate on the stronger network based on network strength. The 4G/5G network will enable the live video and huge data transmission, while the 2G network will ensure that fundamental capabilities are always available, even in bad networks. In the future, the system could integrate driver monitoring functions to alert the driver if he or she is drowsy or sleeping while driving. In cases like a hit-and-run, a number plate monitoring system software with cameras can be used to log the number plates of the vehicles that were around the vehicle (with a system onboard) to identify the perpetrator vehicles.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] Saha, H. N., Konar, A., Guha, S., Mukherjee, A., Sarker, S., Chowdhury, S., ... & Biswas, S. Automatic Ambulance System using IOT.

[2] Saha, H. N., Raun, N. F., & Saha, M. (2017, August). Monitoring patient's health with smart ambulance system using Internet of Things (IOTs). In 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON) (pp. 91-95). IEEE.

[3] Bhagchandani, K., & Augustine, D. P. (2019). IoT based heart monitoring and alerting system with cloud computing and managing the traffic for an ambulance in India. International Journal of Electrical and Computer Engineering, 9(6), 5068.

[4] Vistro, D. M., Rehman, A. U., Mehmood, S., Idrees, M., & Munawar, A. (2020). AN IoT based approach for smart ambulance service using thingspeak cloud. J. Crit. Rev., 7(9), 1697-1703.

[5] Alvi, U., Khattak, M. A. K., Shabir, B., Malik, A. W., & Muhammad, S. R. (2020). A comprehensive study on IoT based accident detection systems for smart vehicles. IEEE Access, 8, 122480-122497.

[6] Singhal, A., & Tomar, R. (2016, October). Intelligent accident management system using IoT and cloud computing. In 2016 2nd international conference on next generation computing technologies (NGCT) (pp. 89-92). IEEE.

[7] Karmokar, P., Bairagi, S., Mondal, A., Nur, F. N., Moon, N. N., Karim, A., & Yeo, K. C. (2020, August). A novel IoT based accident detection and rescue system. In 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 322-327). IEEE.

[8] Gowri, S. M., Anitha, P., Srivaishnavi, D., & Nithya, M. (2019, December). Internet of things based accident detection system. In 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 159-163). IEEE.

[9] Sharma, S., & Sebastian, S. (2019). IoT based car accident detection and notification algorithm for general road accidents. International Journal of Electrical & Computer Engineering (2088-8708), 9(5).

[10] Soujanya, K. L. S., & Gutta, S. S. R. (2019). Accident alert system with IoT and mobile application. International Journal of Recent Technology and Engineering (IJRTE), 7(5S2), 337-340.