Mobile Vehicle Crash Detection System

Tey Han Yee and Phooi Yee Lau
Centre for Computing and Intelligent Systems (CCIS), Universiti Tunku Abdul Rahman
Kampar, Malaysia.
teyhanyee@1utar.my, laupy@utar.edu.my

***Abstract*— Vehicle collision detection (VCD) system is an Android Apps. VCD system requires the assistance of on-board sensors: (1) GPS, and (2) Accelerometer, and (3) Linear Acceleration. GPS is used to calculate the car speed, while the accelerometer is used to calculate the acceleration force. Our idea is to combine both sensors decision to evaluate an accident condition, i.e. the severity of potential accidents. Connectivity is required, as the database is located at the backend using backendless API, for data creation/updates. User can be service provider or driver, (1) providers will receive accidents sent by driver and (2) driver has to provide information surrounding the trip. The typical flow of the application (for driver) is: Step 1: Driver activate application and key-in relevant information, Step 2: When potential accident is detected, the system will automatically check for the closest Service Provider(s), and Step 3: Relevant information and notification to them. The application begins by prompting the number of passengers from the user (integer value). Then, the user activates the detection; the application will monitor speed and acceleration force of the vehicle. When a potential dangerous event data occurs, it will prompt the user to confirm whether an accident has occurred. User can cancel if no accident has happened, whereas if there is no responds from user for 20 seconds, the application will search for the nearest providers within 10km radius. If no providers are found within 10km radius, a message will be sent to 911. If a Provider is found within 10km radius, details of the accidents including number of passenger, current victim and location will be stored in a database and assigned to the provider. Lastly, a notification will be delivered to the assigned provider.**

Keywords—crash detection; real-time system; acceleration detection; mobile applications

I. INTRODUCTION

Traffic accidents are one of the main causes of death and injuries worldwide [1]. In 2014, Aina suggest that the World Health Ranking 2011 has ranked Malaysia at number 20 in its list of countries with the most deaths caused by road accidents [2]. There should be more implementation of car accident detection system to reduce the number of deaths resulted from car accidents as a recent study suggests that rapid treatment and transportation of victim to the hospital in the shortest time can reduce fatalities by six-percent [3]. General Motors launched the Onstar technology in 1997 that provides safety services to GM vehicle. In 2014, Creek, reported that the OnStar system has fully integrate the vehicle's onboard computer so this allows the system to fully utilize all the safety services [4]. Fig. 1 shows how Onstar Services works. The functions of these services can be accessed through the custom buttons installed on the rearview mirror or console touch screen. The three buttons provide access to the hands-free calling, OnStar advisors and emergency services. Several of these services can be access in manual condition but others are accessed automatically depending on the car condition. For an example, if a vehicle is involved in a moderate to severe crash, The OnStar will place a distress call on its own to the service center. An OnStar advisor then determine if the vehicle require assistance. However, this system is pricey as it requires hiring operators and the technology is expensive.

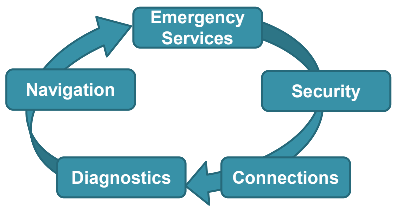

Fig. 1. OnStar services

ACN system is a technology that can reduce the time needed for information to reach the medical personnel. Standard automated car accident detection system is called Automatic Crash Notification (ACN), an example is the Ford SYNC 911 Assist, which is equipped inside a vehicle and the driver must link a Bluetooth-enabled mobile phone with the vehicle. It is typically triggered by an air bag deployment and an instant notification of the crash will be sent to the TSP answering point. The TSP advisor will then contact the emergency personnel (911) and help will come even when there is no witness. Fig. 2 is an example a typical flow of ACN system. AACN is the evolution of ACN; it further improves the data sent from the ACN version by incorporating the calculation of accident's severity. This calculation includes the types of injuries they are likely to find, and it can determine the type of trauma facility so that the correct treatment may be selected [5].

BMW Assist, Ford's 911 Assist, OnStar and Toyota's Safety Connect developed a technology named "Calling for Help Even If You Can't". It works as an Advanced Automatic Collision Notification (AACN) that send the call for help when it detects an accident [6]. This technology uses a Bluetooth-paired connected phone to immediately inform and dispatch Emergency personnel. Typically, when an accident happens and the vehicle's airbags or emergency fuel pump shut off,

978-1-5386-2615-3/18/$31.00 ©2018 IEEE

there will be an activation of ACN and the victim's car will automatically connect the car to an operator and request for verification. Then, the operator verifies the needs for help, if the communication of victim failed to respond the operator would also contact emergency personnel. However, this highly advanced and smart system is only available in limited model of vehicles.

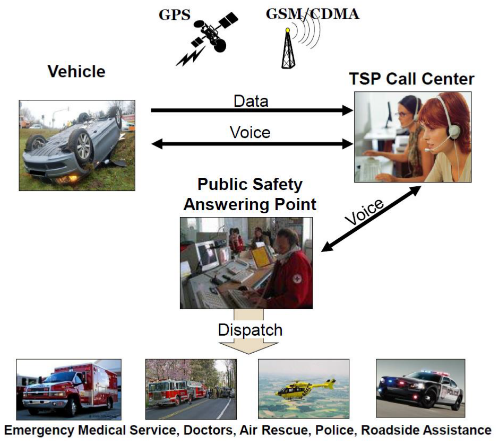

Fig. 2. Illustration of typical ACN System

The motivation of this work is to create a portable and easily accessible android application to substitute for the ACN system in modern cars, named Vehicle Collision Detection (VCD) System. All the standard ACN Systems are only available to specific types of vehicles, not all vehicles can gain access to the ACN system without proper modification. The conventional ACN system is build-in car lacks portability; it cannot be brought around as it can only function together as one with the vehicle. The build-in ACN system in car by default is pricey, not everyone would want to purchase car with such a system. Some people might desire some extra safety features that is built-in a car but cannot afford it. It takes longer to respond to an accident if there are less people around an accident scene where as a system can send the location instantaneously. The proposed Smartphone-based ACN system is an android-application for Android-based OS smartphone above SDK 16 Kit Kat.. This application is developed that can detect the vehicle moving speed and acceleration force using a combination of smartphone sensors (GPS, Linear Acceleration, Accelerometer).

The rest of the paper is organized as follows: Section II describes and discusses our methodology. Section III shows the VCD system. Section IV concludes with future work.

II. METHODOLOGY

The project will benefit people without the finance or capability to purchase modern car with embedded ACN system, as it tends to be expensive. This project is created with the intention of shortening response time for emergency personnel to arrive. In other words, when accidents occur, this project is meant to reduce the risk of passengers losing their life inside the car just because there is nobody around the accident area to call for ambulance. Moreover, this project also reduces the time taken for providers to arrive at the accident scene as the exact location and numbers of passengers have been informed to them. This project suggests a smartphone as a mean to combine all features of a traditional car ACN system, so it is portable and easily carried around to different vehicle. Passengers inside a vehicle can feel safer knowing that the smartphone will send help if potential dangerous events were to happen.

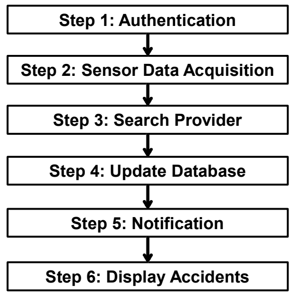

Fig. 3. Proposed VCD Framework

Fig. 3 shows the system flow of the application. Firstly, the Backendless is a Backend as a service (BaaS), also referred as MbaaS for mobile phones, being a method for developers to link to back-end cloud-based storage, most often for push notification, data storage, file storage, monitoring and configuration. It is an alternative to the traditional web server development. In order to consume backend as a service by Mobile Application, there must be a communication protocol which defines the data flow as well as the data structures between the client-side and the backend. Combining the protocol with the data structure definitions creates a programmable interface which the client-side can use to communicate with the server side which is the API.

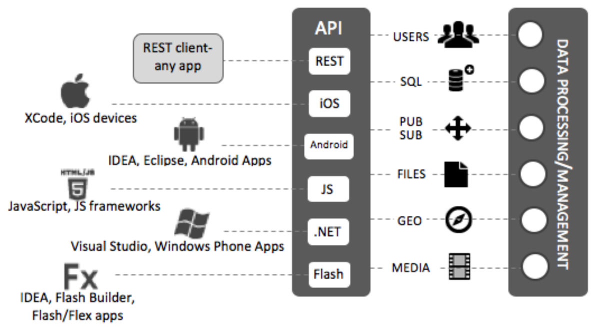

Fig. 4. Visual representation of mBaaS concept

Fig. 4 illustrates the concept of BaaS. In this project case, Client side is Android Apps. Android main thread of UI does not allow blocking calls and since the Backendless API always perform network-based communication, it naturally blocks calls. Thus, Asynchronous versions of the API are always used. To use backendless API, the dependency has to be added in build.gradle of the project in Android Studio and initialized in the program. It only needs to be initialized once and in the beginning of a program. Application-ID and Secret-Key can be obtained from Application Settings console from Fig. 5.

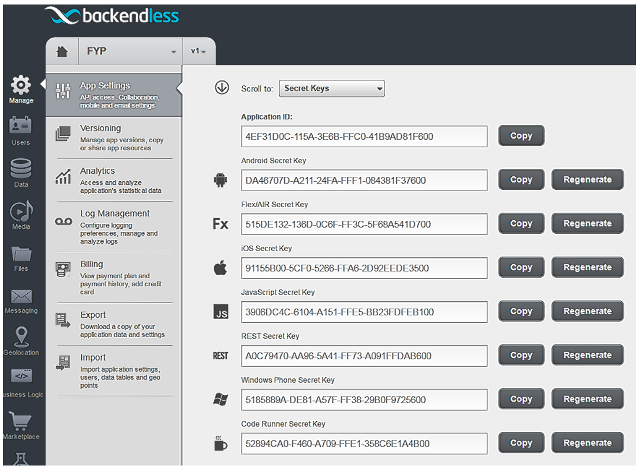

Fig. 5. Backendless Application Settings Console

A. *Step 1: Authentication*

Backendless API is used for instant backend without having to write server-side codes. This project will be using User Service API from backendless service, which provides functionality related to user authentication such as user registrations, logins, password recovery and logouts. Users can register/login as either Provider or User; Provider being the Emergency Centre that sends help in real world after receiving the accident information.

B. *Step 2: Sensor Data Acquisition*

This step assists the application to collect location data and speed data. An accelerometer sensor is a smartphone sensor that measures the acceleration force applied to the device and that includes the force of gravity, while linear accelerometer sensor has subtracted the force of gravity by default. Accelerometer does not measure the moving speed of vehicle so GPS has to be used to retrieve the ongoing vehicle speed. Permission needs to be added to manifests to use the functions. An accelerometer does not measure the full acceleration of the car, when the device is stationary with respect to the surface of the Earth, it will register an acceleration of 9.8m/s2 directed upward. To get the real acceleration of the device, the contribution of the force of gravity must be removed from the accelerometer data. This can be achieved by applying a high-pass filter or low-pass filter. The actual car speed of the vehicle can be attained using LocationManager class from Android.

C. *Step 3: Search Provider*

As for the backend side of storing the location for data, Backendless Geolocation Service API is used to manage this information. This location is stored in term of GeoPoint, GeoPoint consists of a pair of coordinates: latitude and longitude and optionally additional metadata. Each GeoPoint also belongs to a category that is a logical group of GeoPoints. Each accident's GeoPoint is being linked to a victim and a provider object. Fig. 6 illustrates these concepts. It also provides a method to search in radius using GeoQuery. The example below shows finding in Providers category using the center point latitude and longitude. Any GeoPoint's data that matches the search will be stored in GeoPoints.

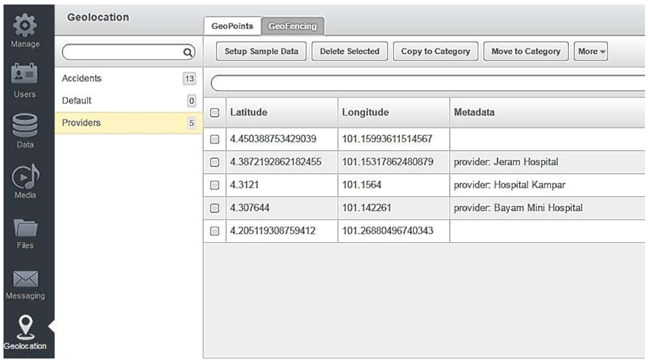

Fig. 6. Concepts of GeoPoint in Category

D. *Step 4: Update Database*

Backendless Data Service is used for all object data storage operations (Create, Retrieve, Update and Delete). The data service operates with persistent data at the object level, which means applications use the APIs to save, update, delete or search for objects rather than traditional database records.

E. *Step 5: Notification*

Backendless Messaging Service API supports push notification, which can be used to inform Providers. The API includes support for targeted delivery where a message is sent to a specific Provider that will attend to the accident. Fig. 7 shows the device appearing in the Backendless Messaging console for the devices registered. These devices can receive notification.

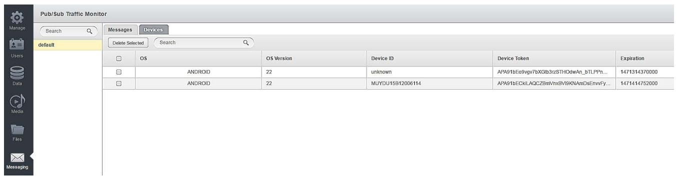

Fig. 7. Device registered for Backendless Messaging API

F. *Step 6: Displaying Accidents*

Accidents will be displayed in Google Map. Reverse Geocoding can be used to retrieve an address name or information concerning an administrative area corresponding to geo-coordinate given (longitude, latitude). All accident details when displayed to user/providers are actually displayed after reverse Geocoding.

III. VEHICLE DETECTION SYSTEM (VCD) SYSTEM

This VCD System is developed in Java programming language by using Android Studio as it is the official Integrated Development Environment (IDE) for Android platform development. Fig. 8 and Fig. 9 shows the use case diagram of the VCD system. The VCD system requires the assistance of on-board sensors: (1) GPS, and (2) Accelerometer, and (3) Linear Acceleration. GPS is used to calculate the car speed, while the accelerometer is used to calculate the acceleration force.

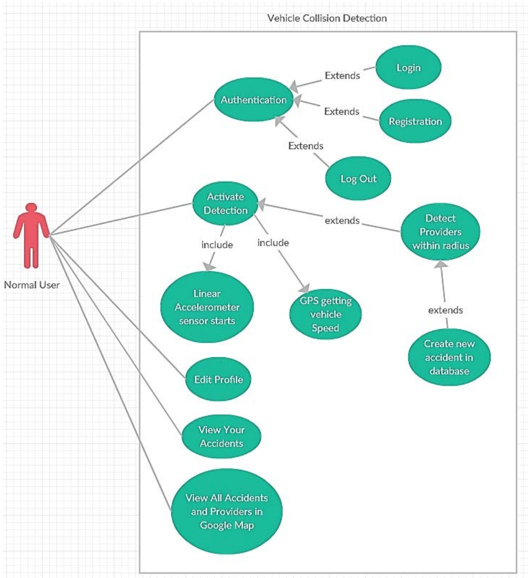

Fig. 8. Use Case diagram for User

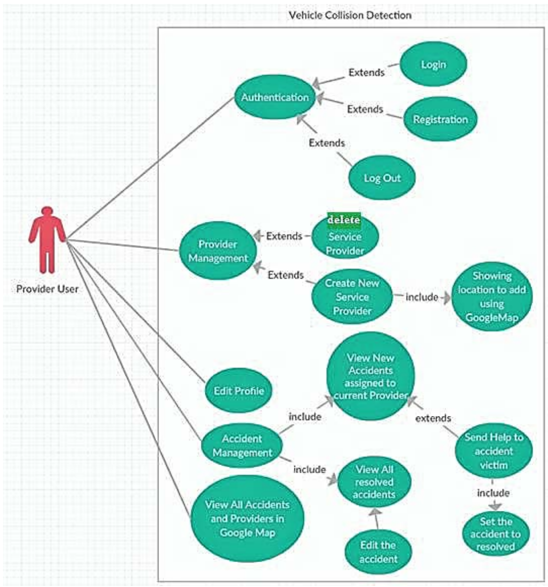

Fig. 9. Use Case diagram for Provider

User can be service provider or driver, (1) providers will receive accidents sent by driver and (2) driver has to provide information surrounding the trip. The application begins by prompting the number of passengers from the user (integer value). Then, the user activates the detection; the application will monitor speed and acceleration force of the vehicle. When a potential dangerous event data occurs, it will prompt the user to re-confirm whether an accident has occurred. User can cancel if no accident has happened, whereas if there is no responds from user for 20 seconds, the application will search for the nearest providers within 10km radius. If no providers are found within 10km radius, a message will be sent to 911. If a Provider is found within 10km radius, details of the accidents including number of passenger, current victim and location will be stored in a database and assigned to the provider. Lastly, a notification will be sent to the assigned provider. The GUI for the VCD system is shown in Fig. 10.

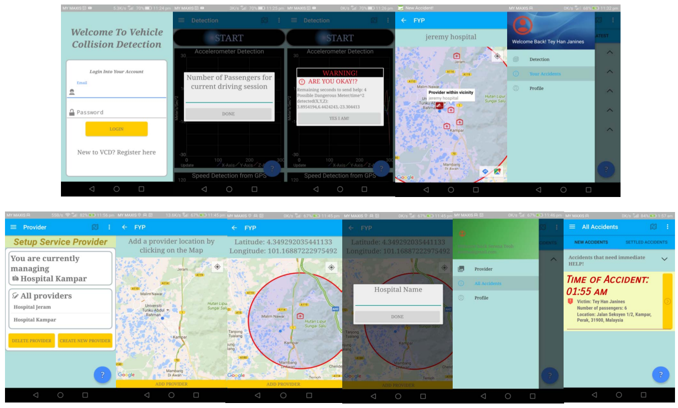

Fig. 10. Implementation GUI for the VCD System (above) User (below) Provider

IV. Discussion and Future Works

Conventional in-vehicle ACN can provide emergency respondents with more important and detailed information but is constrained by the portability and high costs. Smartphone is the most ideal platform since almost everyone has it and the on-board functions are enough to build an accident detection system. In an extreme case of accident, the smartphone may be destroyed but the server site is able to analyze the last known location before the GPS is turned off. Smartphones may also surpasses the standard In-Vehicle in certain way such that the smartphone sensors can measure forces closer to those that have been experienced by the victims. As future works, optimization can be done to the User detection page, instead of using main thread to run sensor service. Also, more customization for users and providers can be added such as preferred providers nearby, first person to contact when an accident occurs.

Acknowledgment

This work is supported by the UTAR Research Fund Project No. IPSR/RMC/UTARRF/2014-C2/L03 “A Framework for V2V / V2I Technologies for Smart Car System” from the Universiti Tunku Abdul Rahman, Malaysia.

References

[1] “WHO|Road traffic injuries”, World Health Organization.
[2] H. Aina, “Malaysia ranked 20th in road death”, News Straits Times.
[3] P. Connie, “Automatic Crash Notification: A Promising Resource for Fire EMS”, Fireengineering.com.
[4] C. Athey, “Assessment of Telematics Service Provider Data Feeds”, ENTERPRISE Transportation Pooled Fund Study TPF-5 (231), Enterprise.
[5] L. Jeremy, “Advanced Automatic Collision Notification”, CarTech.com.
[6] Ford Support, “911 Emergency Assistance | SYNC & SYNC with MyFord”. Ford.