Accident Detection and Smart Rescue System using Android Smartphone with Real-Time Location Tracking

Arsalan Khan, Farzana Bibi, Muhammad Dilshad,
Salman Ahmed, Zia Ullah
Department of Computer Science and Software Engineering
Al-hamd Islamic University, Islamabad, Pakistan

Haider Ali
Department of Pharmacy (Pharm-D)
Sarhad University of Science and Technology, Peshawar,
Pakistan

Abstract—**A large number of deaths are caused by Traffic accidents worldwide. The global crisis of road safety can be seen by observing the significant number of deaths and injuries that are caused by road traffic accidents. In many situations the family members or emergency services are not informed in time. This results in delayed emergency service response time, which can lead to an individual's death or cause severe injury. The purpose of this work is to reduce the response time of emergency services in situations like traffic accidents or other emergencies such as fire, theft/robberies and medical emergencies. By utilizing onboard sensors of a smartphone to detect vehicular accidents and report it to the nearest emergency responder available and provide real time location tracking for responders and emergency victims, will drastically increase the chances of survival for emergency victims, and also help save emergency services time and resources.**

Keywords—Traffic accidents; accident detection; on-board sensor; accelerometer; android smartphones; real-time tracking; emergency services; emergency responder; emergency victim; SOSafe; SOSafe Go; firebase

I. INTRODUCTION

The number of deaths due to traffic accidents is very high. Looking at the number of deaths and injuries due to road traffic accidents shows the global crisis of road safety. Nearly 1.3 million people are killed every year and about 50 million injured worldwide due to road accidents, which averages to 3,287 lives lost every day. More than 50 percent of road traffic deaths affect young adults between the age of 15-44. Around 400,000 individuals under the age of 25 dies in road traffic accidents every year. Even in countries with very good road safety measures, the number of road accident deaths is getting higher every year [1]. More than 90% of road traffic deaths occur in middle-income countries. In low-income countries the figure is even higher.

In Pakistan the last 10 year of statistics shows that an average of 15 individuals lost their lives due to traffic accidents daily. According to data from Pakistan Bureau of Statistics on traffic accidents in Pakistan from 2004 to 2013 [2], the overall deaths in road accidents are about 55 percent, which according to the specialists is very high. According to the data, total 51,416 individuals died in 97,739 road accidents across the country. Furthermore, the data shows that deaths per accident are 55 per cent around the country [3].

The most likely reason for an individual's death in an accident is lack of the first aid provision that is because of emergency services not receiving information about accident in time. Emergency response time is extremely vital when it involves incidents involving vehicle accidents. Analysis shows that if we decrease just 1-minute in accident response time that can increase chances of saving an individual's life up to six percent [4]. In order to reduce response time, implementation of enhanced traffic technologies would be necessary, which will help scale back response time and therefore reduce fatalities.

The purpose of this research is to design and implement such an automated system that uses smartphone to detect vehicle accidents and report it to the nearest available responders to help counter these emerging problems and reduce casualties as much as possible. The detection system would help reduce fatalities due to vehicle accidents by decreasing the response time of emergency services. The system will also provide other emergency services like Fire Brigade, Police Department and Medical emergency services.

In this work we are utilizing android smartphone to detect accidents and report it to the nearest available emergency responders with the exact location of victims in emergency. On an emergency responder side, the system will inform responders about the incidents that occur near to them and provide them with real time tracking of emergency victims on a Google map. This will help emergency responders keep track of victim's location and rescue them as soon as possible.

This paper is organized as follows: Section II describes the related works that has been done in the field, Section III presents various technologies that are utilized in our system, Section IV describes questions related to this work, Section V presents the architecture and implementation of the proposed system, Section VI presents front end design of the proposed system, Section VII presents backend design of the system, Section VIII describes performance results and tests performed of the proposed system, Section IX presents contributions of this work and finally the last Section X is the conclusion and future work for the proposed system.

II. RELATED WORK

Using smartphones to identify road traffic accidents is not a new subject. There are completed algorithms for systems which utilizes accelerometer as well as GPS to detect vehicle accidents using smartphones to detect accidents dates back to 2011. Because there is already a lot done on this subject, what we decided to do was to develop a complete system that is more reliable and have much more functionality than the existing ones, designed for the ongoing project in mind.

In [5] the authors developed a system which used Android smart-phones and ODB-II connection in a vehicle. When the system detects an accident, will sends an SMS to emergency contacts specified by the user, SMS will contain information about the accident and also a call to the emergency services is made automatically. All modern vehicles have ODB-II connection installed which transmits data about the vehicle in real-time such as acceleration, oil pressure, speed, etc. For the system to work a vehicle must support OBD-II standard. In US and this standard is necessary since 2001, European countries have also implemented a version of this standard, so vehicles in the US and in Europe can use this system and is not available to all vehicle in other countries. Other than that, upgrading and maintenance of this system is very expensive process [6].

In [7], the authors at the University of Baghdad Iraq developed a system which made use of the accelerometer, GPS and microphone to detect accidents. Upon detection of an accident sends an emergency notification to the web server and also sends an SMS to the emergency contacts, emergency responders have to access the web server to find out about an accident. Their system made use of the same sensors and hardware that the algorithm presented in this research work makes use of, except for a few features. The main issue with this system is that the notifications are sent to a web server and responders needs to check the web server for accident notification, there is no system for individual responder that responds to the emergency to track victim's location and also the system lacks the functionality to send emergency notification to the nearest emergency center in case there is more than one emergency center in the area.

In [8], the authors developed a system called WreckWatch which involves reading data from the accelerometer and acoustic data from the microphone to detect accidents. If an incident has occurred, the application contacts nearby emergency services and provides GPS-coordinates of accident location.

In [9], the authors have developed an android application that uses accelerometer sensor to detect accidents. After sensing the accident, application automatically sends a voice message to 108 ambulance emergency response service that is running in India. The issue is that this system is for specific emergency response services, only applicable in India. Also, the system is prone to increased false positives because there is no filter in place to verify if an accident detected by the smartphone is a real accident or just false alarm due to dropping smartphone, etc.

In this study we looked at various technologies and existing systems providing us with broad analysis and helped us in developing our system. From analysis we found that these systems can play a very important role in saving human lives. A new system is to be developed based on unique features that will help counter emergencies.

III. TECHNOLOGIES UTILIZED

A. Smartphones

Smartphones are mobile phones that have considerably a lot of functionality than a regular mobile phone. they're mobile computers. Smartphones are powerful and versatile as a result of built in sensors, powerful processors, multiple network interfaces and a high amount of memory for such small devices.

B. Android

Android is a Linux kernel based open-source mobile operating system which was developed by Google for phones, tablets, watches, TVs, cars and other electronic devices [10]. Being open-source, everyone has full access to the Android source code, with one restriction, it cannot be used for personal profit or any financial gain. It is the most popular mobile operating system.

Android alternatives include iOS by Apple, Windows Phone, BlackBerry, Symbian and a few others. We chose Android as it is the operating system that have the most programming experience with. Android's market dominance and cheap application release costs were also deciding factors.

C. Android Studio

Android Studio is the primary Android IDE (Integrated Development Environment). It provides an Android developer all the necessary tools to develop an Android application. More specifically, it allows writing code with auto-completion tools, debugging, testing [11], running the code on a physical or a virtual device and setting programming related or visual preferences. Java and XML are the only languages required to create Android applications with Android Studio [12]. Android Studio does not have any alternatives worth considering. It is possible to develop Android applications with Eclipse by using the Android Developer Tools plugin, but it is no longer supported by Google [13].

D. Java

Java is a class based general-purpose, object-oriented programming language [14]. It is a high-level, strongly typed language with garbage collection that incorporates concepts from several languages including C and C++, but it is not entirely the same. For example, Java does not allow writing unsafe code that might cause vulnerabilities and unexpected behavior. The main building blocks of a Java application are classes, interfaces and packages.

E. Accelerometer

An accelerometer works by detecting proper acceleration affecting the accelerometer to determine the G-forces affecting the accelerometer [15]. Proper acceleration means acceleration that is relative to free-fall [16]. An object in free-fall would as such have no acceleration affecting it while an object at rest on the surface of the earth would experience an acceleration of 9,81 m/s2 upwards due to the surface pushing the object

upwards to negate gravity. Accelerometers in smartphones bases their functionality on micro-electromechanical-systems (MEMS), which measure electric currents based on compression of a seismic mass, often silicon, caused by acceleration [17].

F. Google Play Services

Google Play services provide application developers a comprehensive set of useful features, for example, Maps and Google+ sign-in. The services include the Google Play services client library and the Google Play services Android Package Kit. The client library makes it possible to access any feature with a user's account and deals with different issues that may occur when using the services. The Android Package Kit communicates with the client library and provides access to a specific service when necessary. The use of Google Play services is a must when using Firebase. Important functionalities in Smart Rescue System, for example, viewing on a map and obtaining a user's location also rely on the services.

G. Google Location API

The Google Location Services API [18] is part of Google Play Services, provides a more robust, high-level framework that automatically chooses a suitable location provider and power management. Location Services also provides new features like activity detection which is not provided by framework API. Developers should consider using Location services API if they are using framework API and also if they are making their apps location –aware.

H. Android Google Map API

The Google Maps Android API is a service which is part of the Google Play services library. Allows access to Google Maps server automatically, displaying map, downloading data, and map gesture response. It also allows to add markers, polygons, and basic map overlays, and to transition the user's context of a specific map area [19].

I. Google Places API Web Service

The Google Places API Web Service is a service which returns information about places like locations, geographic, establishments and prominent points of interest using HTTP requests [20]. The main alternative of Google Places API is Foursquare Venues. In free version Google Places allows 150000 queries per day [21] and Foursquare Venues allows 120 000 queries per day [22]. These two services are mostly similar, we chose Google Places API because we were more familiar and experienced using it.

J. Google Directions API

The Google Directions API [23] is a service that uses HTTP request to calculate distance between locations. When calculating directions, the API returns the most economic routes. The API decides which route is most efficient on the basis of travel time, number of turns, distance, etc.

K. Retrofit

Retrofit [24] is a type-safe HTTP client for Android and Java, provided by Square. Retrofit makes it easy to communicate with a web server and get back data, as java

objects, it automatically supports a large collection of response types, including converting JSON objects into Plain Old Java Objects. Alternative of Retrofit is Googles Volley [25], which is also a HTTP library, for our system we chose Retrofit because it is light weight and has more documentation.

L. Backend as a Service

Backend as a Service (BaaS) is a hosted backend that has been premade for developing a web or mobile application. Developers do not have to write any or much backend specific code. It has all the necessary features of a backend and even more features such as Facebook and Google sign-in integration and cloud messaging are common. The features can be accessed by documented APIs that simplify the application development process.

M. Firebase

Firebase is one of many implementations of the BaaS model. Like other BaaS implementations, Firebase provides storage, push notifications, user authentication and a database. Other than the basic BaaS features, Firebase also give a test lab that permits testing a Firebase connected application with different configurations and devices. A feature that makes Firebase different from other BaaS implementations is the real-time database. When new data is added to the database, it becomes accessible instantly to all the users of the application.

N. Firebase Cloud Messaging

FCM (Firebase Cloud Messaging) is another adaptation of GCM (Google Cloud Messaging). It is a cross-platform messaging solution that allows us to reliably deliver and receive messages without any cost. Using FCM, we can send notification messages in order to re-engage users. [26].

O. GeoFire

GeoFire is an open-source library for Android/Java that allows us to save and query a set of keys according to their geographic location. At its core, GeoFire simply saves locations with string keys. Its main advantage however, is that it allows querying key in a specific geographic area in real time [27].

IV. RESEARCH QUESTIONS

Sr.no	Research Questions	Motivation
1	What are the benefits of such system?	This question will elaborate the pros and cons of accident detection system.
2	What are the research contributions?	This research aims to provide a complete system for both emergency victims and emergency responders.

V. SYSTEM OVERVIEW

The designed system consists of:

- **SOSafe:** An android application for emergency victims.

- **SOSafe Go:** An android application for emergency responders.

- **Firebase:** as Server, Database, File Storage, Cloud Messaging, Auth.

A. Used Services

    a) Google Play services

    b) FCM (Firebase Cloud Messaging) services.

    c) Firebase Realtime Database

    d) Firebase Authentication

    e) Firebase Storage

    f) Google API Client

    g) Google Maps API

    h) Google Location API

    i) Google Direction API

    j) Google Places API

    k) Retrofit
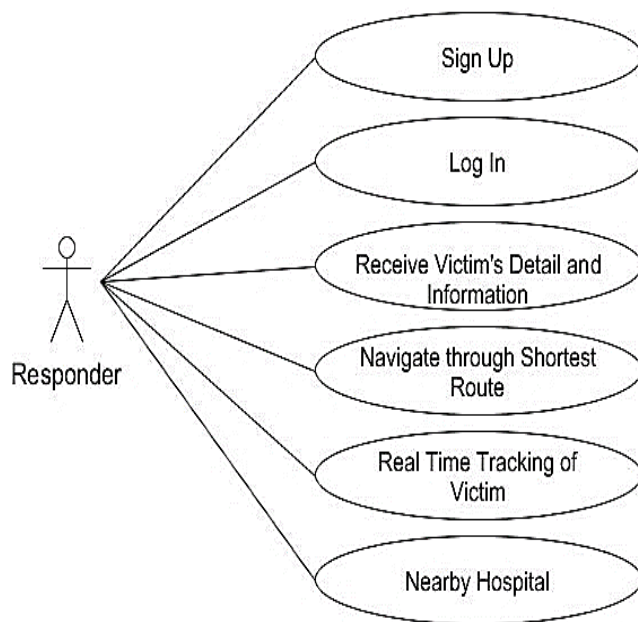
B. Use Case Diagrams

    1) SOSafe

Fig. 1. SOSafe use case.

0 shows the Top-Level Use Case of SOSafe, which indicates the user's full interaction with the system. It shows the user is firstly registering himself, user can then log in to the system using firebase email and password authentication. He/she can view/update his profile at any time after authenticated. The user can turn on Automatic Monitoring which will register an accelerometer service running in the background, it can now detect all kinds of jerks produced by the user on his/her smartphone and correctly differentiate between accidents and normal routine jerks.

Upon detecting the right accidents, the system will generate an alert containing an alarm sound on the emergency victim's phone. users will be able to cancel sending alert to emergency responders in case of false alert (accident didn't occurred) under 15 seconds. SOSafe will get victim's location using Google Location API and save it to firebase real time database, then search for nearest emergency responders from victim's

location and will send an alert notification to the nearest emergency responder (containing victim's location) using FCM. SOSafe will also send SMS to the emergency contacts containing location of the victim. When emergency responder accepts the request sent by victim, SOSafe will show real time location tracking of responder to the emergency victim on a Google map. SOSafe will also provide details about responders (name, vehicle number, phone, etc.).

In case of other emergencies users can select the type of emergency he/she is in (Fire, Ambulance, Police), then by pressing panic button the system will search and notify nearest responders available for the selected type of emergency.

    2) SOSafe Go

The system also consists of an application for emergency responders. Responders will be able to select the type of emergency services they provide and other information related to it. This application will show emergency notifications that are sent by emergency victims and provide real time location tracking of their locations. In case of medical emergencies, the system will also guide responders to a nearest hospital from emergency location.
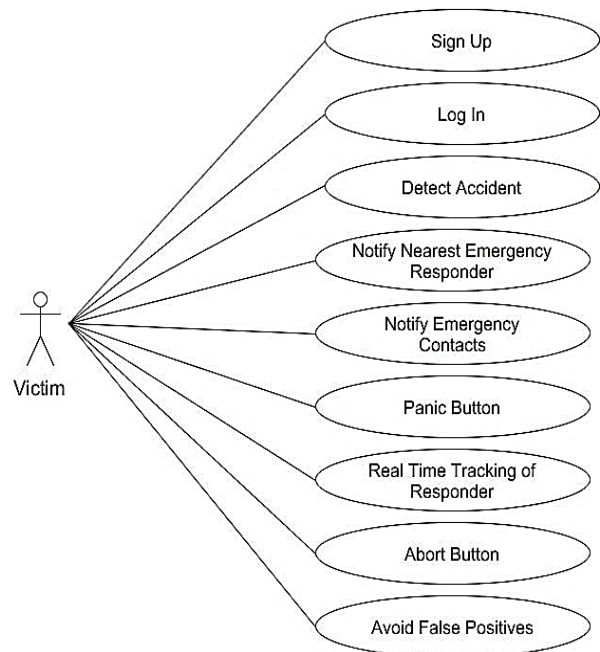

Fig. 2. SOSafe go use case.

Fig. 2 shows the Top-Level Use Case of SOSafe Go, which indicates the emergency responders' full interaction with the system. It shows the responders firstly registering themselves, responders can then log in to the system using firebase email and password authentication, responders can view/update thier profile at any time after authenticated. Emergency responders can then receive emergency requests from emergency victims using FCM.

When SOSafe Go receives emergency requests, an alert with sound and vibration triggered will be shown with location of the emergency. When requests are accepted by the

responders, responders will be able track the location of emergency victims in real time on a Google map with shortest route to location of emergency victims using Google Directions API. SOSafe Go will also show details about victims (name, address, blood group, etc.) when emergency victim is rescued, if victims are in need of medical assistance SOSafe GO will guide responders to the nearest hospital from emergency locations by utilizing Google Places and Google Directions API.

C. Activity Diagram

0 is showing the sequence of activities held in the system.
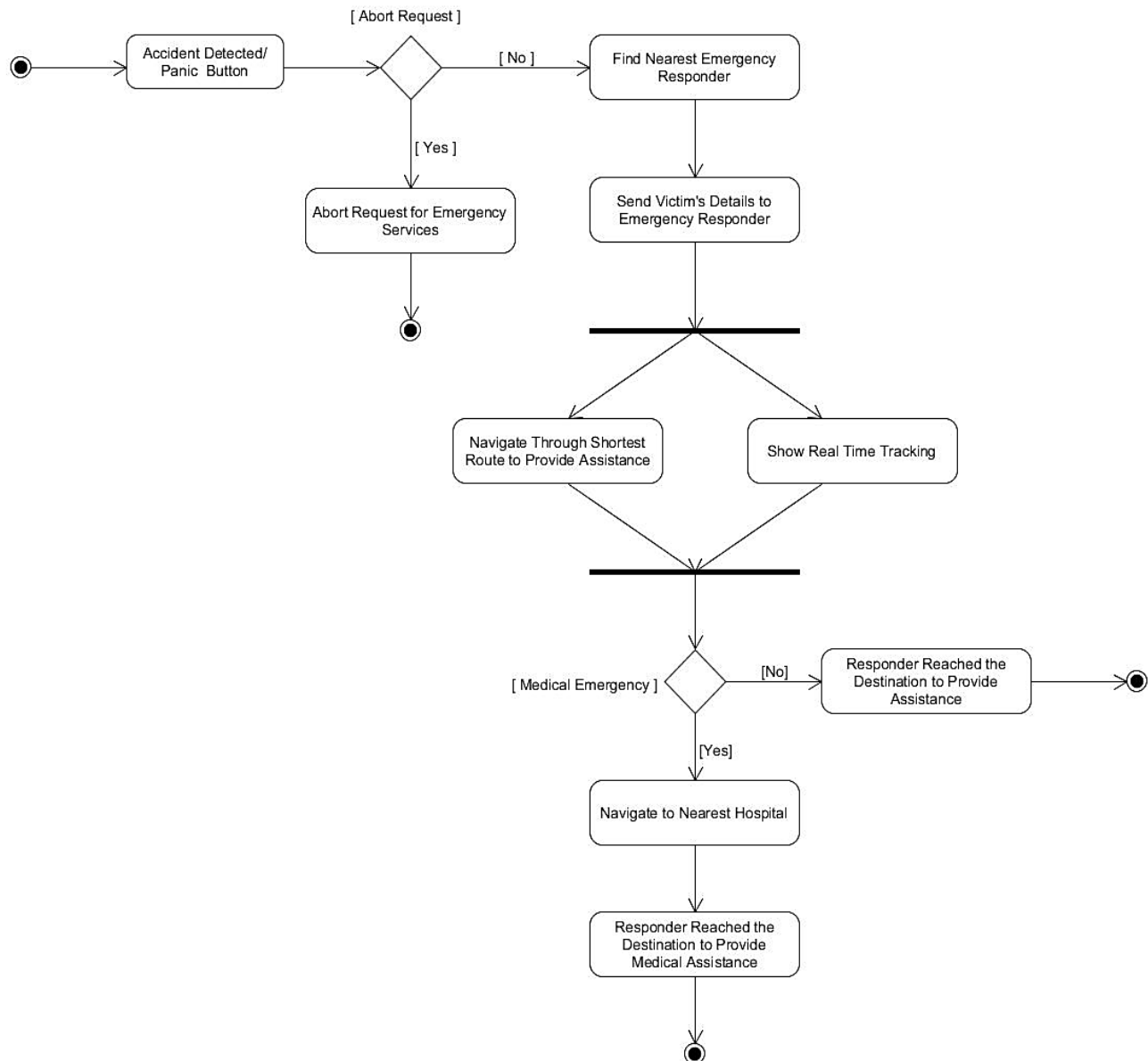
Fig. 3. Overall system activity diagram.
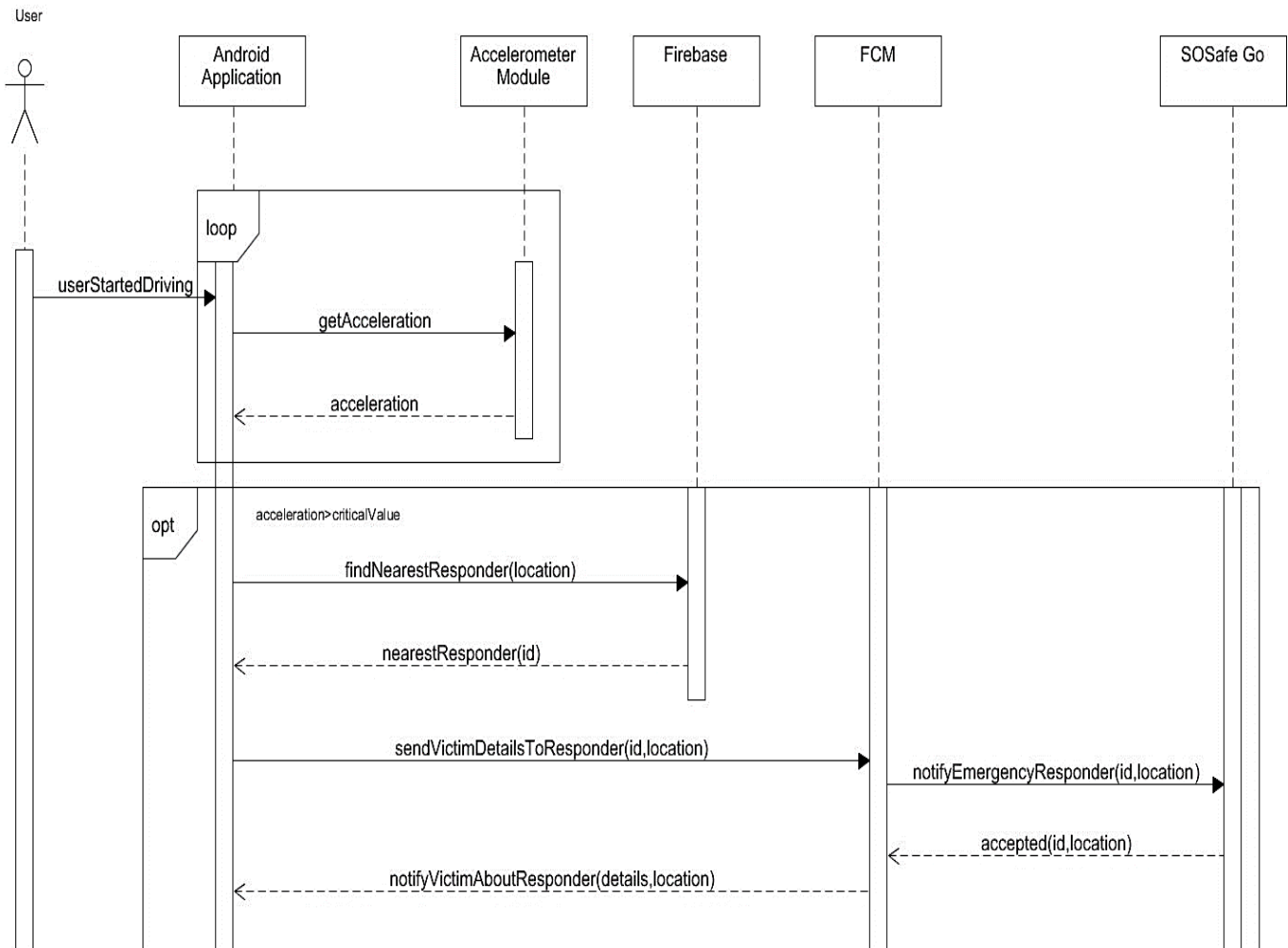
Fig. 4. System sequence diagram.

D. Sequence Diagram

In 0, sequence diagram shows the sequence in which emergency victims' application, Firebase and responders' application are performing their work.

E. Firebase as Backend

The brain of the whole system is Firebase. Firebase Authentication is used to authenticate users by Email and Password. Firebase Authentication also provides a number of Authentication API's like Google, Facebook, GitHub, etc. All information about emergency victims and emergency responders (availability, location. etc.) are stored at Firebase database. Information about emergency requests sent from emergency victims' side is also stored and is processed by matching attributes of the request to a nearest emergency responder available. A notification message about emergency request is then sent to the available responder through FCM. Firebase is used to store real time location data, using GeoFire library for Firebase. Firebase Cloud Messaging is used to exchange data and send notifications between emergency victim's app and emergency responder's app. The system will

also deal with unexpected scenarios like when there is no responder available and avoiding false positive.

F. Algorithm for Accident Detection

Algorithm that uses on-board accelerometer sensor of a smartphone to detect accidents was developed for our system in this research work. The question is that what to do with the values that are being generated by the accelerometer sensor and setting a threshold value that will trigger the accident alert. Accelerometer delivers acceleration values for each of the three axes. Accelerometer values are based on (1).

$$A_D = -g - \left(\frac{\sum F_s}{m}\right)$$

Equation (1) shows that the acceleration values that an accelerometer sensor (A_D) generates is force (F_s) divided by mass (m) which is affected by gravitational acceleration (-g). Acceleration for each axes (ax, ay, az) is derived based on (1).

Now we will use Pythagorean Theorem to derive values from accelerometer as shown in (2).

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

By using the value of (a) we can calculate the value of g-force (gravitational force). Equation (3) shows how g-force is calculated.

$$G = \frac{a}{g}$$

Equation (3) shows that g-force value is calculated from acceleration (a) divided by gravitational acceleration (G). G-force value will be approximately 1 if smartphone is resting on a table and will exceed 1 if the device is moving. Accidents can be detected by checking if g-force value exceeds a specified threshold, which in our system will be 4g, which then will generate an emergency alert.

G. Avoiding False Positives

Since accident detection system that uses smartphone can dispatch emergency services, it is necessary to avoid false alerts. Differentiating between accidents that occurred versus dropping your phone or sudden stop is relatively hard due to smartphone mobility. If the system cannot analyze and avoid false positives precisely, but can make it meaningless by wasting emergency services resources on false incident reports. We have added some features to improve our systems reliability, accuracy and avoid false positives.

1) Acceleration filter prevents false positive:

SOSafe will not trigger emergency event if the G-force value is below 4g. This value can detect accident but will avoid triggering emergency event on dropping a smartphone or sudden stop.

2) Count down timer alert to prevent false positive:

In case the system detects an accident, the system will generate a countdown alert dialog with sound and vibration for 15 sec. in case of false alert (accident didn't occur) the user will be able to cancel sending emergency alert to emergency responder under 15 secs. This will help in reducing false positive, as the user will be able to cancel sending an emergency notification in case of false event.

VI. FRONT END DESIGN

A. SOSafe

Emergency victims' side application SOSafe is developed in Java programming language using Android Studio as IDE. This prototype application is developed for android operating system having a minimum API level 17, and target API level 26. The application is fully working and implemented on the Android smartphone.
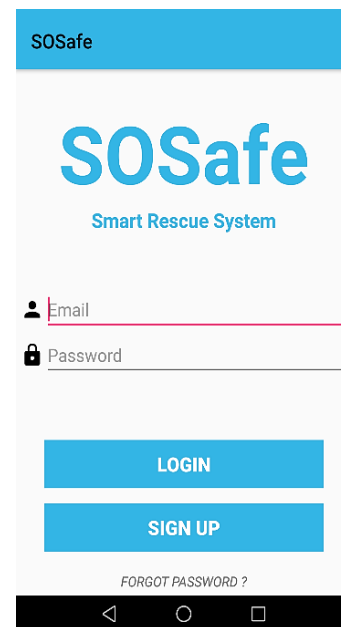
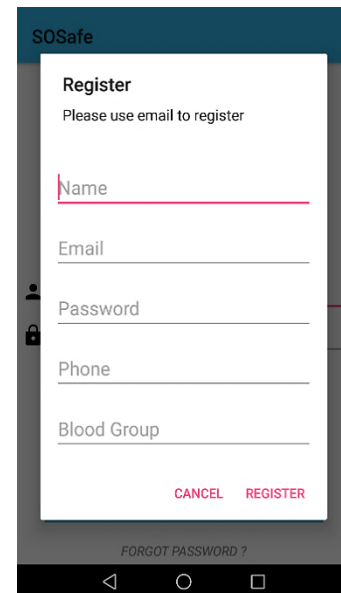Fig. 5. Log in screen (SOSafe).

Fig. 6. Sign up screen (SOSafe).

Fig. 5 shows log in screen of SOSafe, users can use email and password that they used to register, to log in to the system. After users logs in to the system, they will be able to use all system features.

Fig. 6 shows sign up screen of SOSafe, users provides email, password and other details (name, phone and blood group) in order to register. After registration all information will be saved at Firebase database.
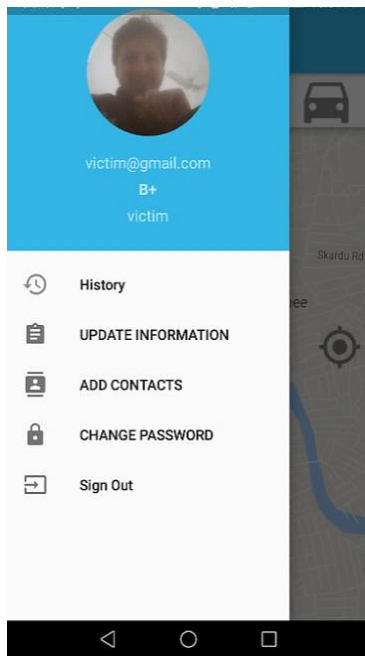
Fig. 7.   Navigation drawer (SOSafe).

Fig. 9.   Panic button (SOSafe).

Fig. 8.   Home screen (SOSafe).
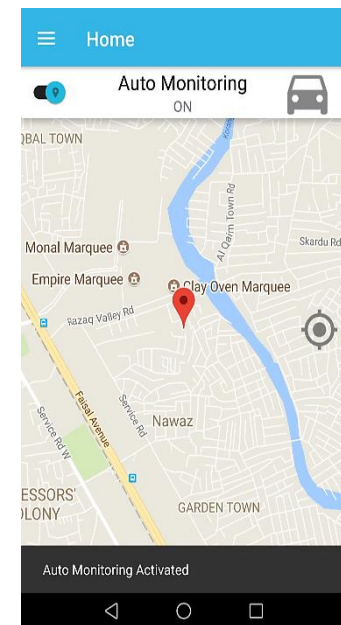
Fig. 10. Auto monitoring/auto accident detection (SOSafe).

Fig. 7 shows navigation drawer of SOSafe, users can view history of previous emergencies, update their information, add emergency contacts numbers, change password and sign out.

Fig. 8 shows home screen of SOSafe, user can turn on Auto Monitoring for automatic accident detection or send emergency request manually.

Fig. 9 shows Panic Button feature of SOSafe, users can select the type of emergency service and press "Request Emergency" button to manually send an emergency request to selected emergency service responder near to them.

Fig. 10 shows, users of SOSafe can turn on "Automatic Monitoring", the system will register an accelerometer service running in background; it can now detect accidents.
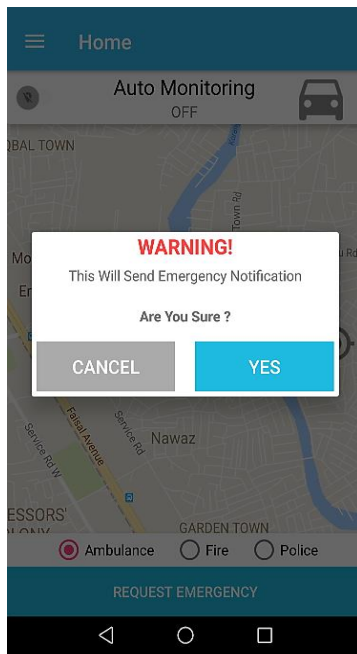
Fig. 11. Panic button alert dialog (SOSafe).

Fig. 13. Responder details (SOSafe).

Fig. 12. Accident detected alert dialog (SOSafe).

In Fig. 11, when user of SOSafe presses the panic button, the system will present an alert dialog to confirm the action before sending an emergency request to responder, this will help in situations when panic button is pressed accidently.

In Fig. 12 when SOSafe detects an accident, will present an alert dialog with a 15 sec count down timer, in case of false alert the user can abort sending request by pressing "Cancel" button. If there is no response from the user in 15 sec, it will be considered as an actual accident and the system will send an emergency alert to the nearest emergency responder and also to the emergency contacts.

Fig. 14. Real time tracking of responder (SOSafe).

Fig. 13 shows, when an emergency request sent by emergency victim is accepted by emergency responder, SOSafe will show details about that responder to the victim.

In Fig. 14, SOSafe will show real time location of emergency responder on a Google map to the emergency victim.

B. SOSafe Go

Emergency responders' side application SOSafe Go is developed in Java programming language using Android Studio as IDE. This prototype application is developed for android operating system having a minimum API level 17, and target API level 26. The application is fully working and implemented on the Android smartphone.
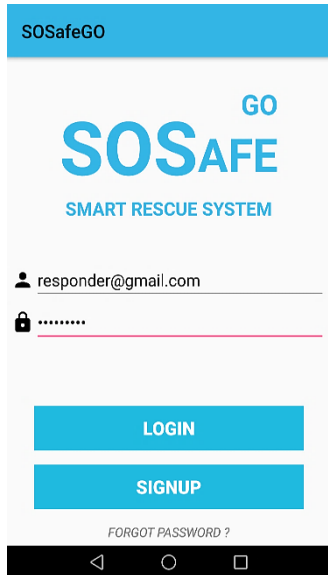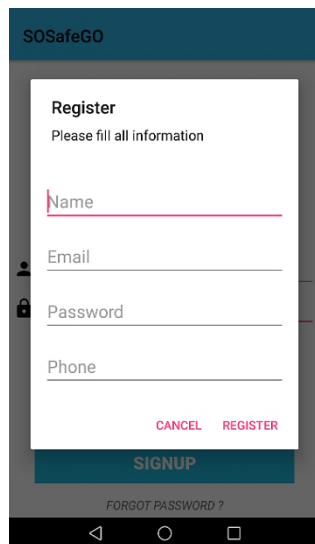

Fig. 15. Log in screen (SOSafe Go).

Fig. 16. Sign up screen (SOSafe Go).

Fig. 17. Home screen (SOSafe Go).

Fig. 18. Navigation drawer (SOSafe Go).

Fig. 15 shows log in screen of SOSafe Go, emergency responder can use email and password that they used to register, to log in to the system. After emergency responders logs in to the system, they will be able to use all system features.

Fig. 16 shows sign up screen of SOSafe Go, emergency responder needs to provide email and password and name in order to register. After registration all information will be saved at Firebase database.
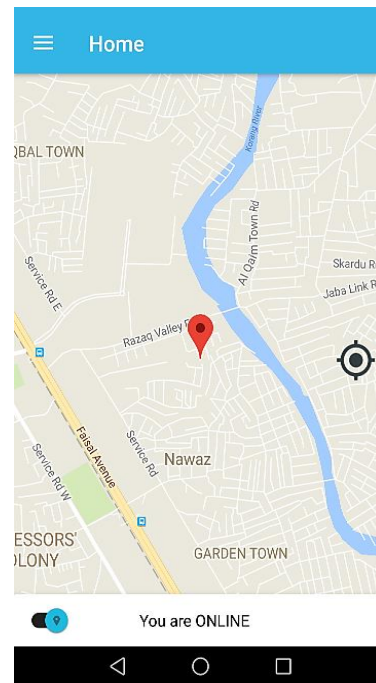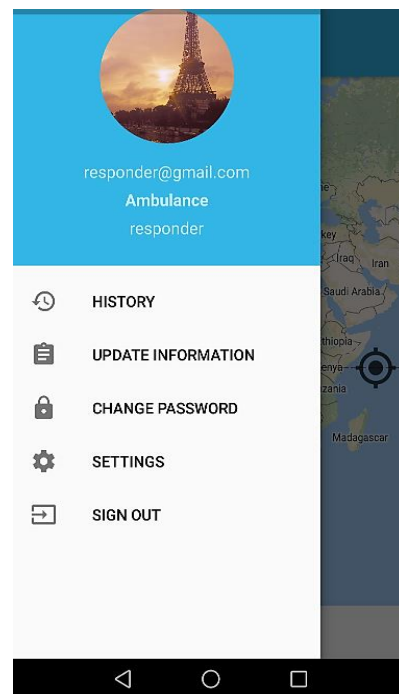
Fig. 17 shows home screen of SOSafe Go, responders can turn on the switch to go Online, responders can now receive emergency requests sent by emergency victim.

Fig. 18 shows the navigation drawer of SOSafe Go, emergency responders can view history of previous emergencies, update their information, change password, and select the type of emergency services they provide, e.g. Fire Brigade, Police Department or Ambulance.
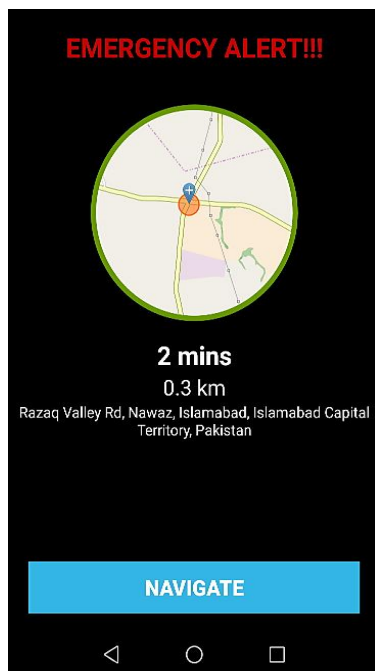
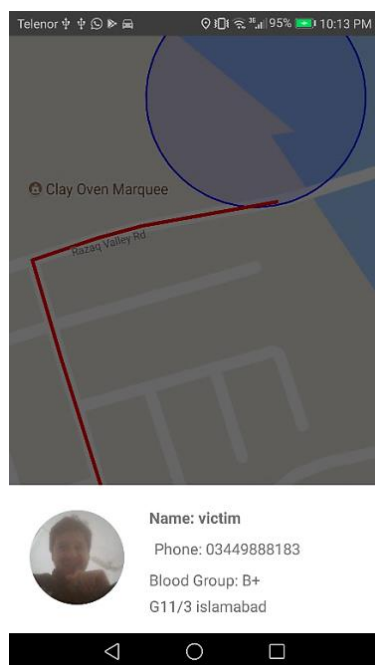Fig. 19. Emergency alert screen (SOSafe Go).

Fig. 20. Victim's details (SOSafe Go).

In Fig. 19, when SOSafe Go receives an emergency request, the system will show an emergency alert screen to the responder with sound and vibration turned on. Emergency responder can press "Navigate" button to get direction to the emergency location.

Fig. 20 shows, when an emergency request by emergency victim is accepted by emergency responder, SOSafe Go will show details (name, phone, blood group, and address) about that emergency victim.
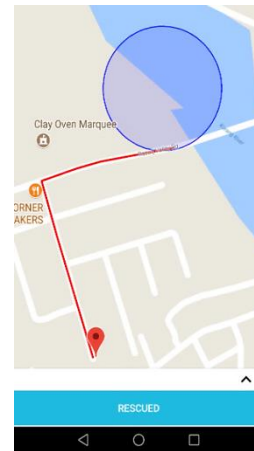

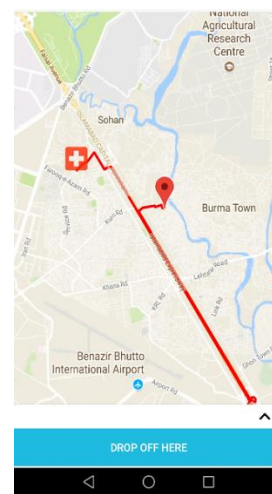Fig. 21. Real time tracking of victim (SOSafe Go).

Fig. 22. Nearby hospital route (SOSafe Go).

In Fig. 21, SOSafe Go will show real time location of emergency victim on a Google map to the emergency responder.

In Fig. 22 shows, upon rescuing emergency victim, if the victim is in need of medical assistance, SOSafe Go will guide emergency responders to the nearest hospital from emergency location.

VII. BACK END

In our proposed system we used only cloud-based server Firebase for data storage, user authentication, file sharing, location sharing and push messaging. Here we will discuss how firebase is utilized in our system.

A. User Authentication

1) Registration
In case of registering to the system user has to provide name, email address, password, phone, etc. Once user is registered into the system a passive user id will be generated and this id will always be used to identify user and access backend.

Fig. 23. User authentication database snapshot.

2) Log In

User has to provide email and password to login. Once the user is logged in, it is not necessary to login every time unless user is logged out. The firebase authentication system provides the user id which is synced with a device token that matches the user authenticity. Fig. 23 shows registered users, these users are authenticated and can log in to the system to use all system features.

B. Real Time Database

1) Responders

Fig. 24 shows the node in Firebase database where real time location data of responders currently online will be saved.
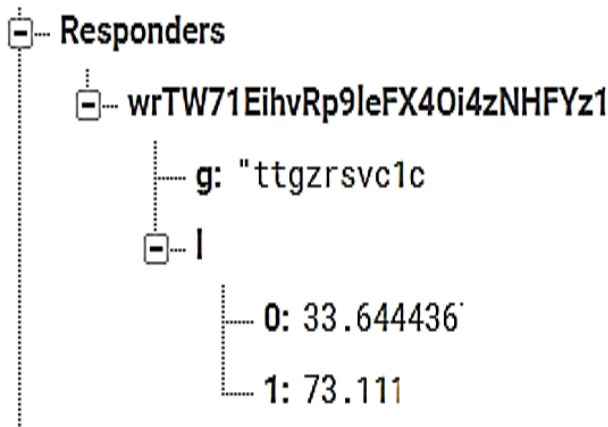

Fig. 24. Online responders database snapshot.

2) PickupRequest

In Fig. 25, this node in Firebase database will contain emergency requests that are sent by emergency victims, with their location. Each request will be created with a separate child node using the id of victim that sent the request.
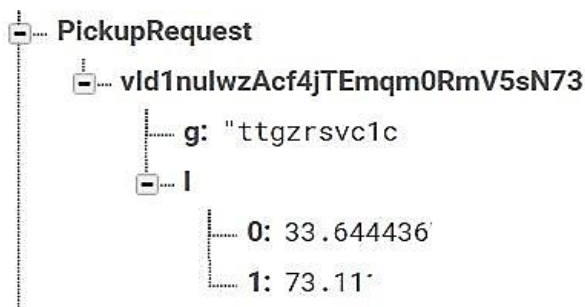

Fig. 25. Emergency requests database snapshot.

3) Tokens

Is an id issued by Firebase Cloud Messaging connection servers to client applications allowing it to receive messages. Tokens will be used to send notification messages to both responder and victim. Fig. 26 shows the node in Firebase database where tokens for each user will be saved.
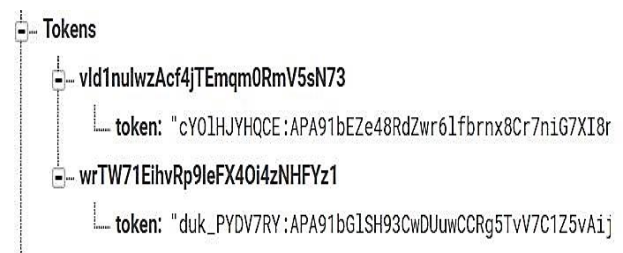

Fig. 26. Tokens database snapshot.

4) RespondersInformation:

In Fig. 27, when SOSafe Go user signs up, all information about that responder will saved under this node in Firebase database. It will contain details about responders like, name, email, phone, emergency service type, vehicle number, history, etc.
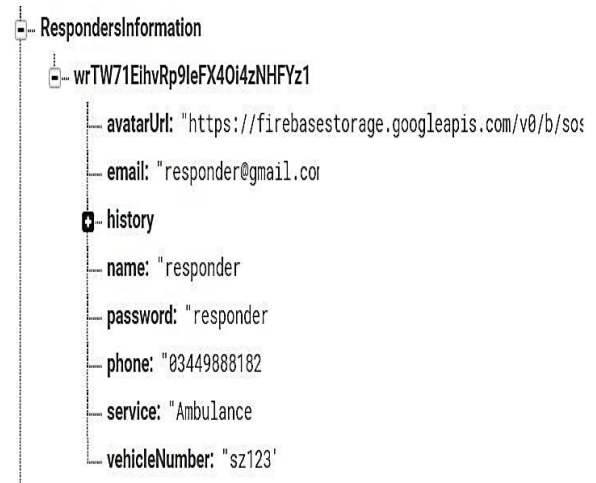

Fig. 27. Responders information database snapshot.

5) VictimsInformation

In Fig. 28, when a SOSafe user signs up, all information about that user will saved under this node in Firebase database. It will contain details about victims like, name, email, phone, blood group, history, etc.

Fig. 28. Victims information database snapshot.

6) History

In Fig. 29, this node in Firebase Database will contain information about previous emergencies that each responder has successfully responded to. It will contain victim id, responder id, location, destination, time, etc. for each emergency.
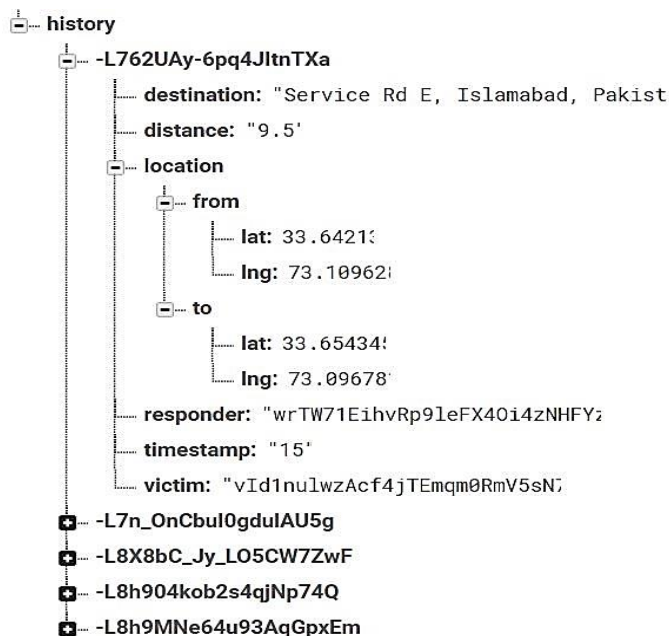

Fig. 29. History database snapshot.

VIII. RESULTS AND DISCUSSION (RQ1)

From emergency victim's point of view, during fatal accidents, emergency victims usually are not able to call an ambulance by themselves, in these situations the designed system will automatically detect the accident and will send an emergency notification to the nearest emergency responder available, to hopefully save victim's life. Sending an emergency alert is a lot easier and convenient because all essential functionalities reside together. In case of other

emergencies, the system also provides functionality to send request to the desired emergency service.

From emergency responder's point of view, the application will show the location of the emergency that occurs near to them, this will help in reducing response time, so that they will be able to track victims in real time and rescue them as soon as possible, resulting in a more efficient usage of emergency services resources.

We conducted some tests by dropping a smartphone from height of 10, 15, 30 and 40cm and recorded the g-force values which can be seen in Table I.

We also mounted the smartphone in a car and recorded g-force value during sudden brakes while driving. Due to limited resources and lack of vehicle crash test labs in Pakistan we were not able to test our system by conducting a real vehicle crash test, but the tests we have done by applying sudden brakes in a vehicle, are somewhat close. The results can be seen in Table II.

TABLE I. G-FORCE DURING FREE FALL

Height	Max	Min	Average
10 cm	2.484621402	1.126285167	1.805453284
15 cm	2.568356721	1.201817380	1.885087050
30 cm	2.981591310	1.766139158	2.373865234
40 cm	3.315491403	2.041593813	2.628542608

Table I shows the results from our drop testing we drop tested a smartphone as many as 10 times and recorded the results, we recorded that accidently dropping smartphone will not cause the system to generate an alert. The maximum g-force value we got in our testing by dropping smartphone from the height of 40cm was 3. 315491403. Our system will generate alert if the g-force value exceeds 4g.

TABLE II. G-FORCE WHILE APPLYING BRAKES IN VEHICLE.

Brakes	Max	Min	Average
Hard	3.025675430	2.220679102	2.623177266
Normal	2.357143091	1.301464250	1.829303670

Table II shows the results from testing by mounting a smartphone in a vehicle and applying brakes while driving, even in situation like applying sudden brakes in a vehicle the g-force values were lower than g-force during dropping a smartphone. The maximum value we got by applying sudden brakes in vehicle while driving was 3.025675430, which is lower than our critical threshold value. The threshold value at which our system will generate emergency alert is set to 4g, because during fatal traffic accidents the g-force value exceeds 4g. It can detect when an accident occurs but will avoid false positives in case of dropping a smartphone or applying sudden brakes in a vehicle.

IX. RESEARCH CONTRIBUTIONS (RQ2)

Many systems exist that uses conventional in vehicle sensors and also some work done in the field of Smartphone based accident detection systems, we aim to develop a

Smartphone based accident detection system with much more features and better user experience. We developed a complete system for both emergency victims and emergency responders. The system uses on-board sensors of a typical Smartphone to detect accidents and report it to nearest emergency responders. Emergency responders will be able to track the exact location of the victims on a Google map. The system also provides help during other types of emergencies like Fire, Police etc. With innovative and creative ideas, we intend to take the system to another level, staying a step ahead from system which pre-exists.

The proposed system will provide much more functionalities than the existing systems like.

- Directly sends emergency notifications to the nearest available responder.

- Real time location tracking for both responders and victims on a Google map.

- An android application for emergency responders that directly receives notifications about the emergency that occurs near to them and is provided with real time location of the victim and is also provided with the details about victim such as name, blood group, address.

- Provide directions to the nearest hospital during medical emergencies.

- Provide other emergency services such as Fire Department, Ambulance and Police.

X. CONCLUSION AND FUTURE WORK

In this research, we developed the accident detection and smart rescue system, which uses on board accelerometer sensor to detect accident and generate emergency alert and send it to the nearest emergency responder and will also send an SMS to emergency contact containing location coordinates of the accident. With real time location tracking for both victim and responder the system will drastically increase the survival rate of an accident victim by providing emergency aid in time. The system will also provide help during other emergencies such as during fire, robberies/theft and other medical emergencies. Emergency responder will be able pin point victim's location on a Google map in real time.

The probability of false positives in a smartphone-based accident detection and rescue system is inevitable. We have added some features to reduce these issues. Here are some features we added to reduce false positives.

- **Acceleration filter:** The system will ignore g-force values lower than 4g.

- **Count down timer alert:** On detection of an accident the system will present an alert dialog with 15 sec count down, which the user will be able to cancel in case accident didn't occur.

For future work, more research is needed in order to make the accident detection part more reliable and accurate which will help in reducing false positives. Adding additional sensors in combination with accelerometer for accident detection like gyroscope, microphone, camera (to automatically take pictures of the accident) and a voice recognition module to detect noises during a vehicle crash like noise when air bags are deployed, will drastically increase the reliability and accuracy of the system.

REFERENCES

[1] Asirt.org. (n.d.). Road Crash Statistics. [online] Available at: http://asirt.org/Initiatives/Informing-Road-Users/Road-Safety-Facts/Road-Crash-Statistics [Accessed 10 Dec. 2017].

[2] Pbs.gov.pk. (n.d.). Traffic Accidents (Annual) | Pakistan Bureau of Statistics. [online] Available at: http://www.pbs.gov.pk/content/traffic-accidents-annual [Accessed 11 Dec. 2017].

[3] Traffic accidents kill an average 15 people in Pakistan daily. (2015). [Blog] Available at: https://www.thenews.com.pk/print/58036-traffic-accidents-kill-an-average-15-people-in-pakistan-daily [Accessed 11 Dec. 2017].

[4] Evanco and William M., *"The Impact of Rapid Incident Detection on Freeway Accident Fatalities"*, technical report available from Mitretek , McLean, Virginia, USA, report No .WN 96W0000071, June 1996.

[5] J. Zaldivar, C. T. Calafate, J. C. Cano and P. Manzoni, "Providing accident detection in vehicular networks through OBD-II devices and Android-based smartphones," 2011 IEEE 36th Conference on Local Computer Networks, Bonn, 2011, pp. 813-819.

[6] Shahbaz Ahmed Khan Ghayyur, Salman Ahmed, Mukhtar Ali, Adnan Naseem, Abdul Razzaq and Naveed Ahmed, "A Systematic Literature Review of Success Factors and Barriers of Agile Software Development" International Journal of Advanced Computer Science and Applications(IJACSA), 9(3), 2018.

[7] Zainab S. Alwan Hamid M. Ali. "Car Accident Detection and Notification System Using Smartphone". In: International Journal of Computer Science and Mobile Computing 4.4 (Apr. 2015), pp. 620–635.

[8] J. & Dougherty B. & Albright A. & Schmidt DC Chris T. & White. "WreckWatch:Automatic Traffic Accident Detection and Notification with Smartphones". In: Journal of Mobile Networks and Applications manuscript(2011).

[9] Patel K.H., "Utilizing the Emergence of Android Smartphones for Public Welfare by Providing Advance Accident Detection and Remedy by 108 Ambulances", International Journal of Engineering Research & Technology (IJERT), Vol.2, Issue 9, PP 1340-1342, September – 2013.

[10] Chao Wang, Wei Duan, Jianzhang Ma and Chenhui Wang, "The research of Android System architecture and application programming " *Proceedings of 2011 International Conference on Computer Science and Network Technology*, Harbin, 2011, pp. 785-790.

[11] Shahbaz Ahmed Khan Ghayyur, Salman Ahmed, Adnan Naseem and Abdul Razzaq, "Motivators and Demotivators of Agile Software Development: Elicitation and Analysis" International Journal of Advanced Computer Science and Applications(ijacsa), 8(12), 2017.

[12] Android Developers. (n.d.). Meet Android Studio | Android Developers. [online] Available at: https://developer.android.com/studio/intro/ [Accessed 17 Dec. 2017].

[13] Android Developers. (n.d.). ADT Plugin (UNSUPPORTED) | Android Developers. [online] Available at: https://developer.android.com/studio/tools/sdk/eclipse-adt [Accessed 17 Dec. 2017].

[14] Gosling J., Joy B., Steele G., Bracha G., Buckley A. The Java Language Specification. Java SE 8 Edition. New Jersey: Pearson Education. 2014.

[15] Dimension Engineering LLC. A beginner's guide to accelerometers. 2017. [online] Available at: https://www.dimensionengineering.com/info/accelerometers [Accessed 1 Jan. 2018].

[16] Edwin F. Taylor & John Archibald Wheeler. "Spacetime Physics". In: 1st ed. Vol. 1. 1. San Francisco: W.H. Freeman, July 1966. Chap. 1, pp. 97–98. ISBN: 071670336X.

[17] James B. Angell & Stephen C. Terry & Phillip W. Barth. "Silicon Micromechanical Devices". In: Scientific American 4.4 (Apr. 1983), pp. 44–55.

[18] Android Developers. (n.d.). Location and context overview | Android Developers. [online] Available at: https://developer.android.com/training/location/ [Accessed 4 Jan. 2018].

[19] Google Developers. Overview | Maps SDK for Android | Google Developers. [online] Available at: https://developers.google.com/maps/documentation/android-sdk/intro [Accessed 4 Jan. 2018].

[20] Google Developers. (n.d.). Overview | Places SDK for Android | Google Developers. [online] Available at: https://developers.google.com/places/android-sdk/intro [Accessed 15 Jan. 2018].

[21] Google Developers. (n.d.). Places API Usage and Billing | Places API | Google Developers. [online] Available at: https://developers.google.com/places/web-service/usage-and-billing [Accessed 15 Jan 2018].

[22] Developer.foursquare.com. (n.d.). Rate Limits - Foursquare Developer. [online] Available at: https://developer.foursquare.com/docs/api/troubleshooting/rate-limits [Accessed 15 Jan. 2018].

[23] Google Developers. (n.d.). Developer Guide | Directions API | Google Developers. [online] Available at: https://developers.google.com/maps/documentation/directions/intro [Accessed 4 Feb. 2018].

[24] Square.github.io. (n.d.). Retrofit. [online] Available at: http://square.github.io/retrofit/ [Accessed 4 Jan. 2018].

[25] Android Developers. (n.d.). Volley overview | Android Developers. [online] Available at: https://developer.android.com/training/volley/ [Accessed 4 Jan. 2018].

[26] Firebase. (n.d.). Firebase Cloud Messaging | Firebase. [online] Available at: https://firebase.google.com/docs/cloud-messaging/ [Accessed 8 Feb. 2018].

[27] GitHub. (n.d.). firebase/geofire-java. [online] Available at: https://github.com/firebase/geofire-java [Accessed 26 Jan. 2018]