NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

BSc THESIS

Smartphone - Smartwatch Based Automatic Car Crash Detection & Emergency Notification Application

Konstantinos A. Pytharoulios

Supervisor: **Manolis Koubarakis,** Professor
Co-Supervisor: **George Stamoulis,** Ph.D. Candidate UoA

ATHENS

JENUARY 2021

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εφαρμογή Ανίχνευσης Τροχαίων Ατυχημάτων και Ειδοποίησης Πρώτων Βοηθειών για Έξυπνα Κινητά Τηλέφωνα & Ρολόγια

Κωνσταντίνος Α. Πυθαρούλιος

Επιβλέπων: **Μανώλης Κουμπαράκης,** Καθηγητής
Συνεπιβλέπων: **Γιώργος Σταμούλης,** Υποψήφιος Διδάκτωρ ΕΚΠΑ

ΑΘΗΝΑ

ΙΑΝΟΥΑΡΙΟΣ 2021

BSc THESIS

Smartphone - Smartwatch Based Automatic Car Crash Detection & Emergency Notification Application

Konstantinos A. Pytharoulios
S.N.: 1115201600142

SUPERVISOR: **Manolis Koubarakis,** Professor
CO-SUPERVISOR: **George Stamoulis,** Ph.D. Candidate UoA

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εφαρμογή Ανίχνευσης Τροχαίων Ατυχημάτων και Ειδοποίησης Πρώτων Βοηθειών για Έξυπνα Κινητά Τηλέφωνα & Ρολόγια

Κωνσταντίνος Α. Πυθαρούλιος

Α.Μ.:1115201600142

ΕΠΙΒΛΕΠΩΝ: **Μανώλης Κουμπαράκης,** Καθηγητής
ΣΥΝΕΠΙΒΛΕΠΩΝ: **Γιώργος Σταμούλης,** Υποψήφιος Διδάκτωρ ΕΚΠΑ

ABSTRACT

Undoubtedly cars have been proved to be one of the most important inventions of the last centuries, since they have managed to drastically change the way humans move. Not only do cars offer people a more convenient and easy means of transportation, but they also reduce the time needed for completing simple everyday tasks. However, as everything in life, all that glitters is not gold. According to statistics, more than 3.700 are those who are fatally injured every day in road accidents around the world. Consequently, more than a million innocent people per year are lost. Even more sorrowful is the fact that a significant number of them could have finally survived if the response were quicker. The question which inevitably comes to the surface is what kind of measures could be taken in order for this mortality to be reduced. One prospective suggestion which seems to have a great potential and requires minimum sources to work is taking the advantage of all the sensors that are embedded in everyday smartphones and smartwatches. This technique is able to provide a satisfactorily accurate detection of a car crash as long as a fast enough notification of the first responders.

This thesis is a complete documentation of the development of an application running on both iPhone and Apple Watch devices dedicated to be virtually present to the accident and take the right actions when no-one else is in the position to do so. At first, the user has the opportunity to sign up, filling in all their personal information and adding their emergency contacts. The main core of the application analyses data collected by device's GPS, gyroscope and microphone so as to detect any unusual incident and finally notifies user's emergency contacts whenever needed. Additionally, it stores all the details about all accidents in an online database for further inspection. Description of the sensors, along with case studies, are included.

SUBJECT AREA: Smartphone and Smartwatch Application, Car Crash Detection and Emergency Notification

KEYWORDS: Smartphone, iPhone, Smartwatch, Apple Watch, Application, Car Crash Detection, Emergency Notification, GPS, Microphone, Accelerometer, Gyroscope

ΠΕΡΙΛΗΨΗ

Αναμφίβολα, τα αυτοκίνητα αποτελούν μία από τις πιο σημαντικές εφευρέσεις των πρόσφατων αιώνων, εφόσον κατάφεραν να αλλάξουν μια για πάντα τον τρόπο με τον οποίο οι άνθρωποι κινούνται. Τα ίδια δεν αποτελούν απλά μία εύκολη και βολική λύση για τις καθημερινές μετακινήσεις αλλά και ένα μέσο ταχύτερης διεκπεραίωσης των αναγκών. Ωστόσο, όπως τα πάντα άλλωστε σε αυτή τη ζωή, ό τι λάμπει δεν είναι χρυσός. Σύμφωνα με τις επίσημες στατιστικές, καθημερινά περισσότεροι από 3.700 είναι εκείνοι που τραυματίζονται θανατηφόρα στο δρόμο παγκοσμίως, συνεπώς πολλοί περισσότεροι από ένα εκατομμύριο αθώοι συμπολίτες μας χάνουν τη ζωή τους ετησίως σε κάποιο τροχαίο δυστύχημα. Ακόμα πιο λυπηρό είναι το γεγονός ότι ένα μεγάλο μέρος αυτών θα μπορούσαν να είχαν σωθεί αν η βοήθεια είχε φτάσει εγκαίρως. Το ερώτημα, επομένως, που αναπόφευκτα προκύπτει είναι το τι είδους μέτρα θα μπορούσαν να ληφθούν προκειμένου η θνησιμότητα να μειωθεί. Μια πιθανή λύση, η οποία μάλιστα δείχνει να έχει πολλές δυνατότητες για περεταίρω βελτίωση και απαιτεί ελάχιστους πόρους για να έρθει εις πέρας είναι η εκμετάλλευση όλων των αισθητήρων εκείνων που τα έξυπνα κινητά τηλέφωνα και ρολόγια διαθέτουν. Μια τεχνική σαν και αυτή, είναι ικανή να επιφέρει πολύ ικανοποιητικά αποτελέσματα ως προς την άμεση ανίχνευση τροχαίων ατυχημάτων καθώς και εξαιρετικά γρήγορη ειδοποίηση των ατόμων που πρέπει να επιληφθούν της κατάστασης.

Η πτυχιακή αυτή αποτελεί μια λεπτομερή ανάλυση του τρόπου ανάπτυξης μιας εφαρμογής σχεδιασμένης να υποστηριχθεί από iPhone και Apple Watch συσκευές, που έχει σαν πρωταρχικό της μέλλημα να πράξει τα δέοντα όταν η κατάσταση το απαιτήσει. Συγκεκριμένα, ο χρήστης έχει αρχικά τη δυνατότητα να εγγραφτεί καταχωρόντας στο σύστημα τις απαραίτητες προσωπικές του πληροφορίες καθώς και τα στοιχεία των ατόμων που επιθυμεί να ενημερωθούν άμεσα στην περίπτωση εμπλοκής του σε τροχαίο ατυχήμα. Στο κύριο κομμάτι της εφαρμογής, στοιχεία που συλλέγονται συνεχώς από τους εξής αισθητήρες της συσκευής: GPS, Ταχύμετρο, Γυροσκόπειο και Μικρόφωνο αναλύονται με σκοπό τον εντοπισμό κάποιας ασυνίθιστης μεταβολής κάποιου εξογενούς παράγοντα και την ενημέρωση ,κατόπιν, των επαφών έκακτης ανάγκης. Επιπροσθέτως, οι πληροφορίες όλων των ατυχημάτων καταγράφονται σε μια διαδικτυακή βάση δεδομένων για εκτενέστερη μελέτη των συνθηκών. Περιλαμβάνεται πλήρης περιγραφή όλων των αισθητήρων που χρησιμοποιούνται αλλά καιανεξάρτητες μελέτες περιπτώσεων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Εφαρμογή για Έξυπνα Κινητά Τηλέφωνα και Ρολόγια, Ανίχνευση Τροχαίου Ατυχήματος και Ειδξποίηση Επαφών Έκτακτης Ανάγκης

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Έξυπνο Κινητό, iPhone, Έξυπνο Ρολόι, Apple Watch, Εφαρμογή, Ανίχνευση Τροχαίου Ατυχήματος, Ειδοποίηση Αριθμών ‘Εκτακτης Ανάγκης, GPS, Μικρόφωνο, Ταχύμετρο, Γυροσκόπειο

Αφιερωμένο στους γονείς μου, που δίχως την συναισθηματική και υλική τους υποστήριξη η ακαδημαϊκή μου πορεία ως αυτό το σημείο δεν θα ήταν τόσο ευχάριστη.

ΕΥΧΑΡΙΣΤΙΕΣ

Για τη διεκπεραίωση της παρούσας Πτυχιακής Εργασίας, θα ήθελα να ευχαριστήσω θερμά τον Υποψήφιο Διδάκτωρ Γεώργιο Σταμούλη για την επίβλεψη, την άμεση υποστήριξη και τις συμβουλές του όταν τις χρειάστηκα καθώς και για το ότι από την πρώτη στιγμή πίστεψε στην ιδέα. Θα ήθελα επίσης να ευχαριστήσω των καθηγητή μου, κύριο Μανώλη Κουμπαράκη, για την ευκαιρία που μου έδωσε να ασχοληθώ με κάτι που ανήκει στα άμεσα ενδιαφέροντα μου.

CONTENTS

LIST OF FIGURES

LIST OF IMAGES

LIST OF TABLES

PREFACE

Cars have always been one of my passions. I remember myself playing Gran Turismo on my old Playstation for hours and hours. So obsessed I was, that in my sixteens I even spent my whole year's pocket-money on buying one of the most expensive steering wheels in order to take my digital racing experience to a whole new level. Although I now dare say that this was a relatively immature move, I will never regret it. Using that fantastic steering wheel with the unbelievable responsiveness on my hands and the overwhelming texture, I was able to realise how dangerous cars can be if we are not very cautious. Fast forward to 2016, in the summer after my high school graduation and I had already been studying to get my driving licence. I could not wait any further. My parents, though, when I finally received the licence and started driving our family's car were not as excited as I was. They were really scared and worried about their son. And the truth is that they still are. So I wanted to do something for them. I wanted to find a way which could keep them calm and make them sleep restfully and feel sure that in case of a car emergency situation of mine they would be the first to be informed. This year was my moment or to be more accurate, their moment. The moment of all parents. The thesis I was assigned from the Department of Informatics (University of Athens), where I study, had the concept of building a smartwatch application. Therefore, I came up with the idea of creating an application running on both iPhone and Apple Watch devices which will keep track of their holder's while driving and notify their emergency contacts in case of an emergency. In a world that people separate from their friends more easily than from their smart gadgets, I firmly believe that we should, at least, take advantage of them to save some lives apart from sending text messages or surfing on the internet.

1. INTRODUCTION

Our society improves day by day, so do cars. They are becoming faster, safer, more energy-efficient, some are even getting hybrid or fully electrical [5]. However, one thing that remains the same is the underwhelming number of drivers who lose their lives each year in car accident [1], [2], [3], [4]. Technology is here to serve humanity in any case. Therefore, the current thesis is an attempt to build an application that could detect car crashes using iPhone and Apple Watch sensors and finally alert emergency contacts automatically, providing all the necessary information for the driver's rescue. The idea is that each time the user gets in their car, they will have the opportunity to run the application, which will be responsible for tracking their location and other factors like car internal noise, acceleration and more.

Furthermore, nobody could doubt that Statistics play a vital part in human life since it helps people through systematical observation improve their lives and come closer to wellbeing. For this reason, this application aims at creating a large corpus of information collected from car accidents happening around the world. It is very likely that this information could provide scientists with some serious insight into car accidents. The applications stemmed from this could vary. Some examples are the rebuilt of specific road parts where accidents occur frequently or the redesign of particular car parts to make them safer.

Finally, it is worth taking into consideration that not all users can have their iPhones always right next to them. For instance, some prefer to use their Apple Watch for their daily tasks while leaving their phone in their pocket or bag. Since the user's experience played a significant part from the beginning of this project, the application has also been designed for Apple Watches. Thus the user can simply start the application using their Apple Watch and manage everything from their wrist.

Welcome to the Street Guardian application.

2. BACKGROUND AND RELATED WORK

2.1. Technologies

It was about one year ago, when a story of a biker caught my attention [6]. While he was cycling, he fell from his bicycle and unfortunately the injuries he suffered were so sever that he could not even call his close family. It was late at night and nobody was walking at a near distance in order to give him a hand. After about 40 minutes, though, some family members approached him. To the biker's surprise, the Apple Watch, which he had on his wrist, had detected his fall and automatically informed his contacts. That was the moment when I came up with the idea of building an app which would be able to solve the same problem, but for car crashes instead. Later, while searching on the internet I found out that Google has built a solution like the one I was thinking. Additionally, I was able to find on the Apple Store an application called "Guardian" which among other features, it claims to have the ability to automatically call emergency services when needed, but I never had the opportunity to try this one on my own, since it is only available in the United States of America. Last but not least, another kind of application which has a lot of things in common with the previous is "Health", an application provided by Apple to its users in order to gather health data from all Apple devices belonging to them, so they can view all their progress and at the same time they can stay safe, as this one keeps records of their heart rates and detects unusual changes of it. However, what is worth mentioning, is the fact that no application running specifically on Smart Watches was found, indicating that there is a large field there for further research and development.

2.1.1 Apple Watch Fall Detection

The idea of the *Fall Detection* in Apple Watch is simple. There is just a specific sensor embedded in the Smart Watch, dedicated to recording falls [7]. In case the device detects a hard fall while being worn by its user, it taps them on their wrist, sounds an alarm and displays an alert. Then, the holder can choose to either contact emergency services or dismiss the alert by passing a button. However, if they have not responded for more than a minute, the Watch handles the situation by itself, calling automatically the emergency services.

Image 2.1: An Apple Watch which has just detected a hard fall.

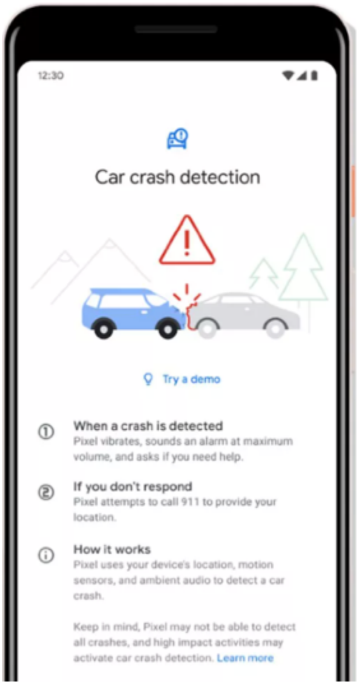

Image 2.2: Screenshot from Personal Safety App: User's Guide.

2.1.2 Google's Personal Safety App

Relying on various sensors like accelerometer and microphone *Google'e Personal Safety App* is attempting to detect car accidents. If there is evidence which raises precautions for an incident like that, the phone will loudly sound an alarm and if there is no response, it will automatically call 911 providing user's location [8].

According to the company behind the project, "Personal Safety" uses location, motion sensors and "ambient audio" from the microphone to figure out if an accident has occurred (a clue could probably be a broken glass or a collision). There is also the possibility for false alarms, but this is not a huge problem since the user can just inform their device that they are safe by pressing a button on the screen.

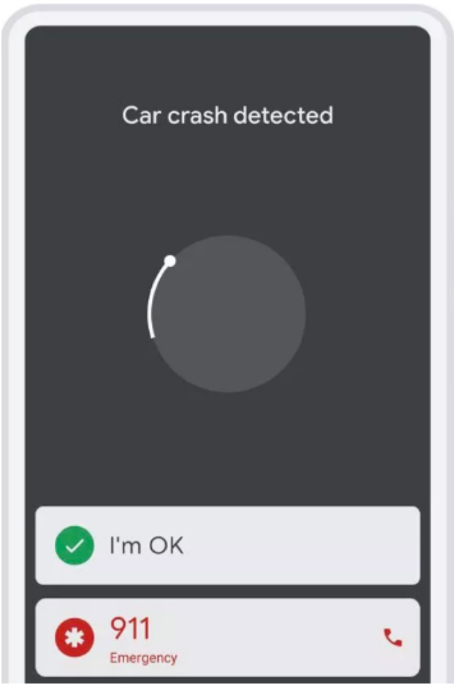

Image 2.3: Screenshot from Personal Safety App: When a car crash is detected.

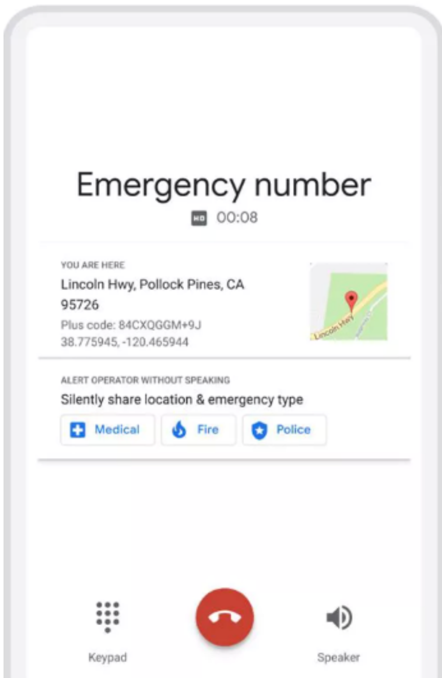

Image 2.4: This is what a user sees when his device has cales 911 during an emergency.

2.1.3 Guardian

The user of the *Guardian Application* enjoys the following features:

- Instant alerts are sent to chosen emergency contacts if their vehicle is involved in an accident.
- Messaging the owners of other cars when they have blocked their car/driveway.
- Review an other driver by writing some comments on the app.
- Find out public domain information about a Vehicle.

In order for the app to work perfectly, an internet connection is obligatory. Unfortunately, the service is currently available only in the US.

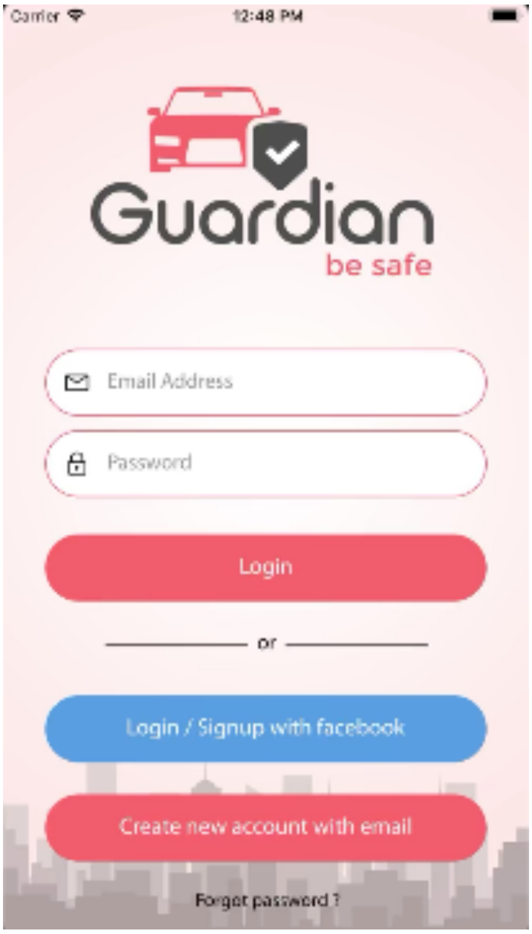

Image 2.5: This is the page where the user can sign in the application.

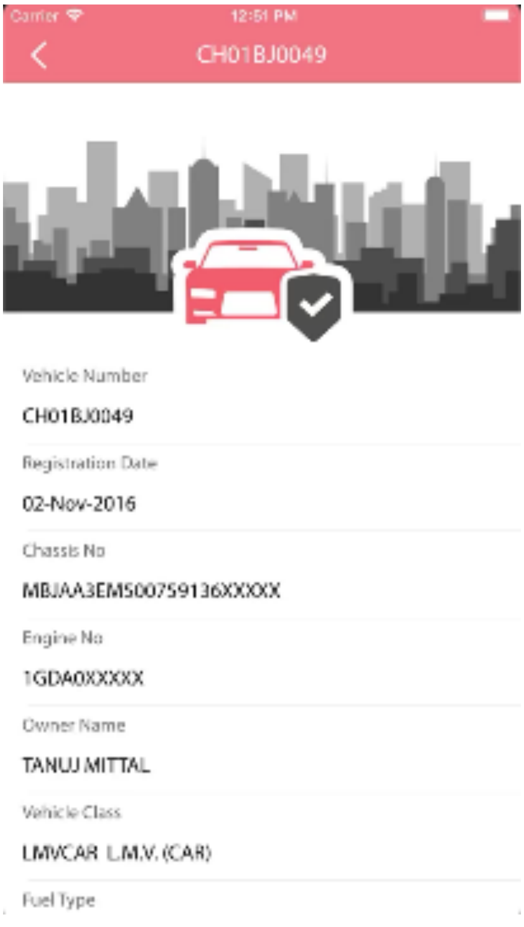

Image 2.6: Here the user can preview all their vehicle's information.

2.1.4 Apple Heath

The *Health* is an application provided by Apple to all its users for free [9]. It analyses data collected by the sensors which are embedded in both iPhone and Apple Watch devices, keeping track of the users' daily life, activity and progress. Each user has the chance to know exactly how many steps they have taken or which was their average heart rate throughout the day and a lot more information. For this reason, this is a good example for someone to understand deeply the importance of simple sensors in everyday life.

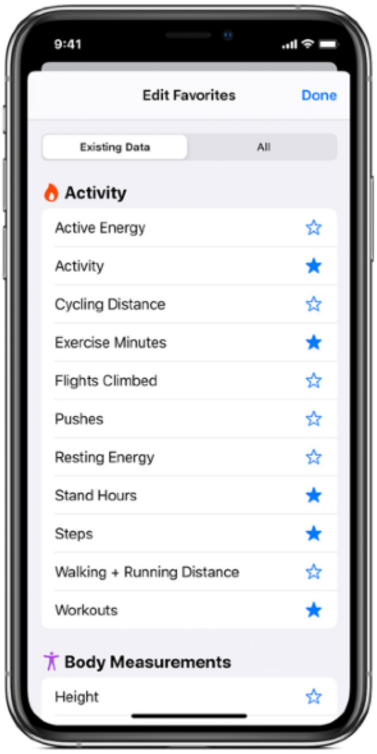

Image 2.7: This is a part of the categories which are tracked by the device.

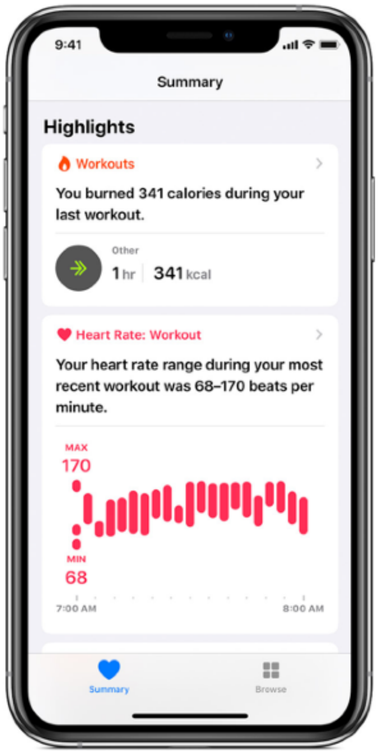

Image 2.8: In this page, the user can see how their health changes over time.

3. APPLICATION

3.1. Role of the app

Before starting with the app's full explanation, it must be pointed out that we are actually talking about two different applications. They are both dedicated to serving precisely the same purpose. However, they run on different devices. The first one is designed for iPhone devices, whereas the second one for Apple Watch ones. So, having this state in mind, we can now argue that the application's role is to keep track of a driver's way while driving so as to keep them safe. More specifically, sensors embedded in the user's smartphone or smartwatch device are used in order to detect abnormal changes in different factors like speed or interior sound.In the displeasing case that a prospective car crash has been detected(relying on the extraordinary changes mentioned earlier), what the application does is to ask the driver if they are safe, as there is always the possibility of having made a wrong guess. If there is not a quick response from them, then the device automatically informs the driver's emergency contacts with an email containing all the crucial information: location, altitude, date, and time. A short description of the application is also included in this email because it is inevitable that not all people are going to know about the existence of this app.

Another fact worth mentioning is that online services and an online database are used [10]. This means that the users are able to create their own accounts and store all their information online. They can, of course, have access to and change them at any time. By this implementation, not only do the devices become more memory and battery efficient, but their processors are also relieved from extra processes like email sending, which can now be carried out by the online services designed. As for the online database, we talk about a large prospective corpus of car accidents that will be detected among all application users and will be stored. The main intention is to ideally use this information to deduct crucial conclusions about road spots that are statistically dangerous for drivers.

3.2. Architecture

The Architecture of the Street Guardian Application can be split into two essential parts: Backend and Frontend. As it is being shown in the image 3.0 below, inside a car, there may be an iPhone or an Apple Watch or both of them. Each one of these two devices has the application preinstalled and uses some specific components, which in this case are sensors, that make all the measurements needed. There is also a connection between the iPhone and the Apple Watch. However, this one is not bidirectional. Only the Apple Watch needs to send information to the iPhone whenever it detects a prospective car accident in order for the latter to connect to the database. Another part of the Frontend is the device that a driver's emergency contact will receive an email notification concerning the car accident detected.

As for the backend, this includes a Firebase server [11], which is responsible for receiving different stimuli from the Frontend and for providing some online services. A stimulus may be a query to the database concerning a user's personal data or an instruction to store the details of a car crash. An online service can be the email service, which is triggered each time a new accident is stored in the Firestore (this is the application's database).

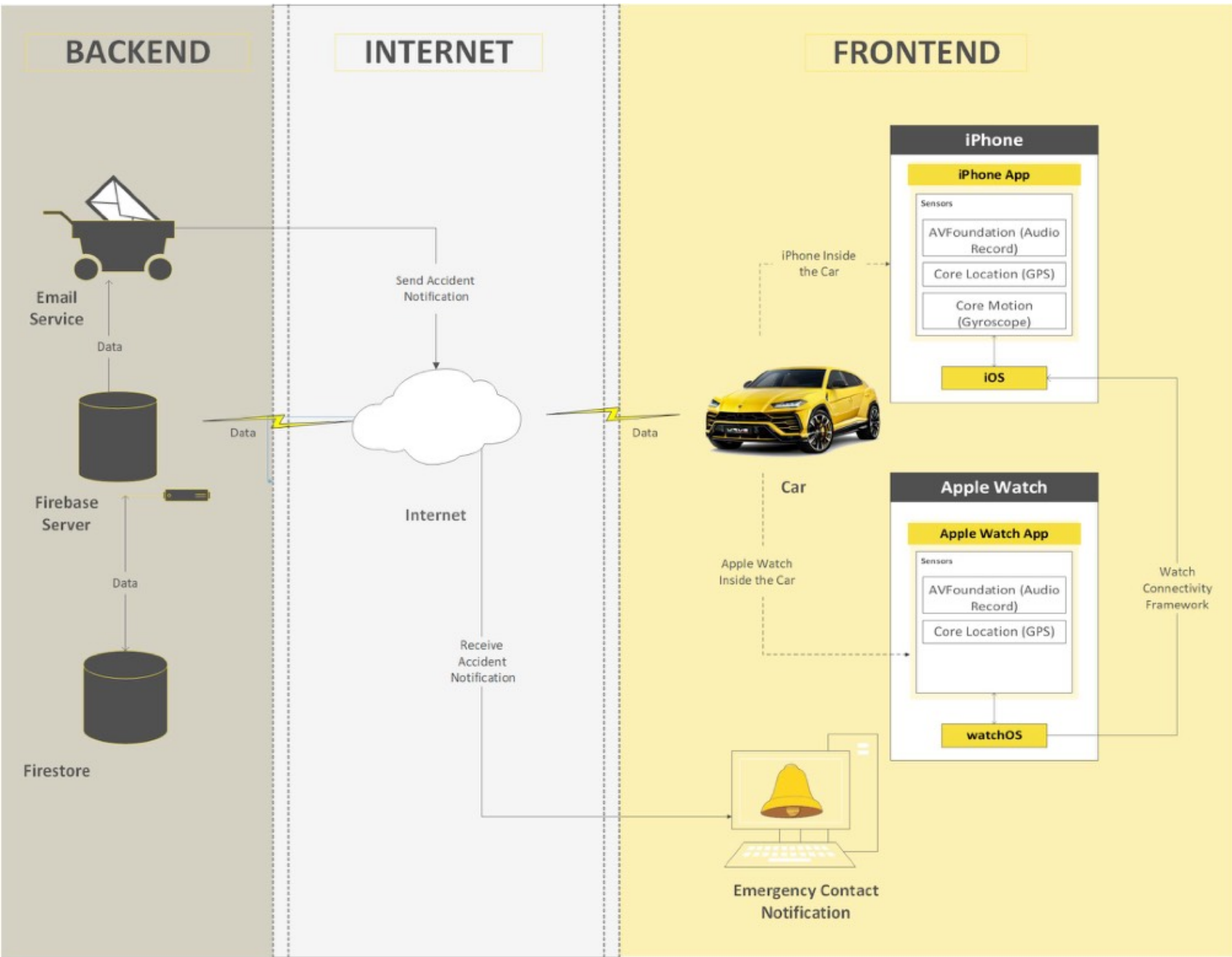

Figure 3.0: Application's Architecture (Backend and Frontend).

Now we are going to take a closer look at the application's screens architecture on the iPhone and Apple Watch devices. Afterwards, we will check each one of the screens separately.

Image 3.1: iPhone Application's Architecture.

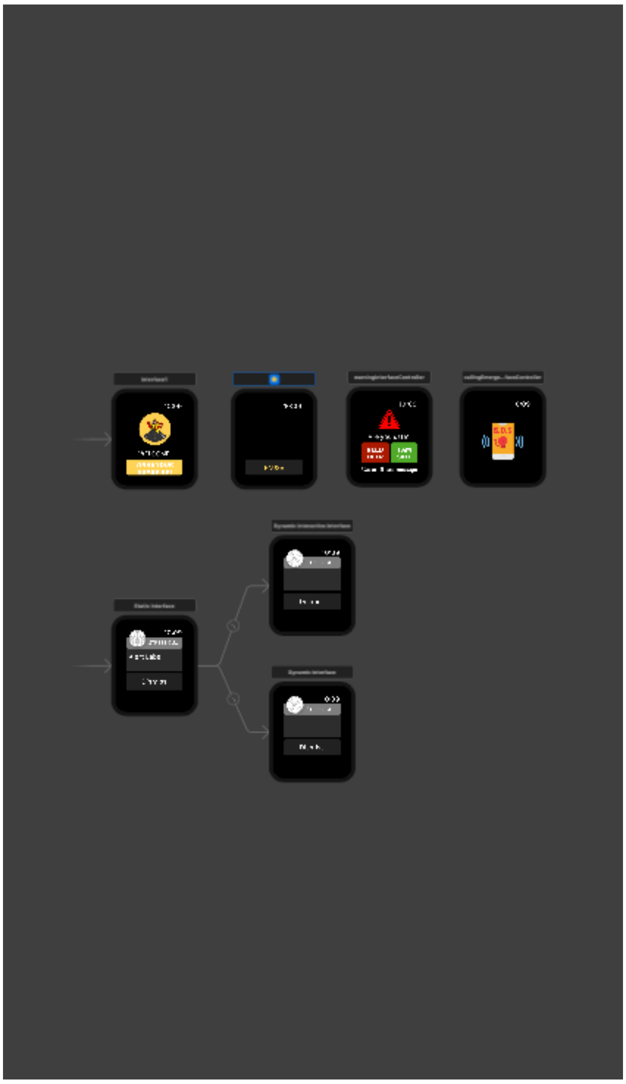

Image 3.2: Apple Watch Application's Architecture.

3.3. User Interface and Features

3.3.1 Onboarding

When the user opens the iPhone's app for the very first time, the initial screen which they will meet is a pop-up window, the well-known “onboarding" [12]. This is a quick and handful manual containing all the essential information that a new user should know for the app before using it.

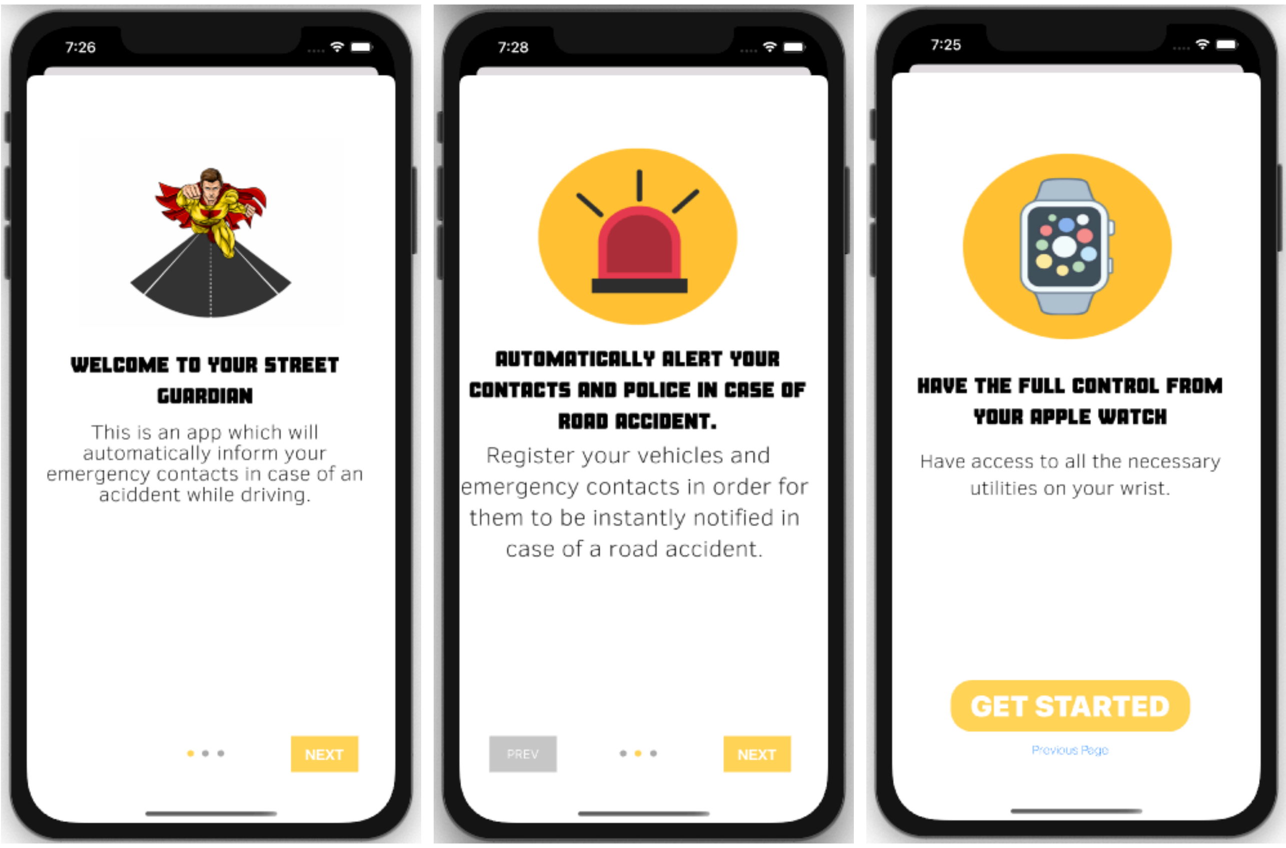

Image 3.3: This is the Onboarding. Essential information of the application are briefly mentioned.

The user can scroll between the three different screens as many times as they desire. After pressing the "Get Started" button, these screens are not going to show up again. The only way they could see them again would be to delete and reinstall the app. So the user is finally moving to the next series of screens where they can sign up and register their personal information as well as their emergency contacts' information.

3.3.2 Registration

It is now high time for the user to create an account and register their personal information. Firstly, they have to fill in their name, surname, and email address (Image 3.4). Of course, there is also a button to press for those who already have an account. If the spelling check of the information mentioned is successful, the user is to create a password that they have to type twice so as to avoid misspelling (Image 3.5). In case of discrepancy or low safety password use, an error message prompts the user to try again. Before finishing registration, they ought to provide the name, surname, and email address of at least one emergency contact. This is the person who will receive an alerting email in case of a user's emergency situation (Image 3.6). Finally, if all the gaps have been successfully filled in, the user has nothing else to do than merely pressing the button "Get me in" which will store the information given to the database, create their account, and show them the home screen of the app (Image 3.7).

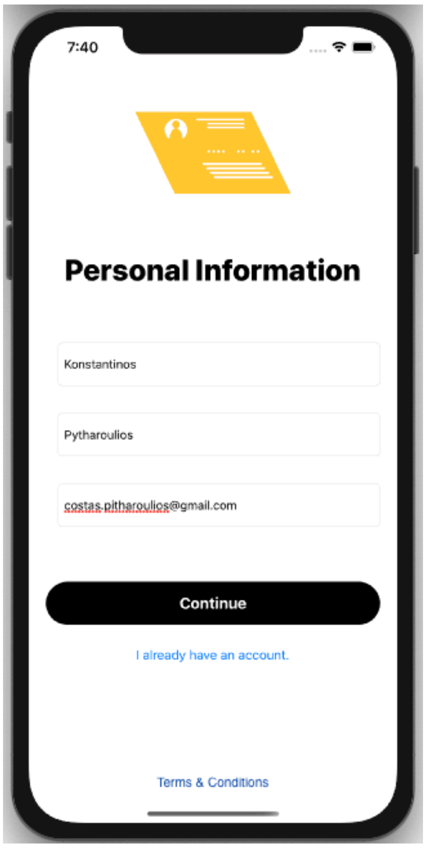

Image 3.4: User's personal information registration.

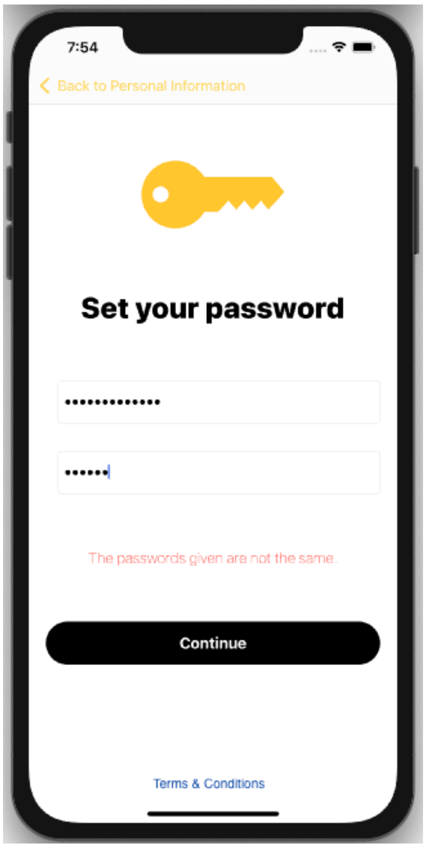

Image 3.5: Discrepancy between passwords.

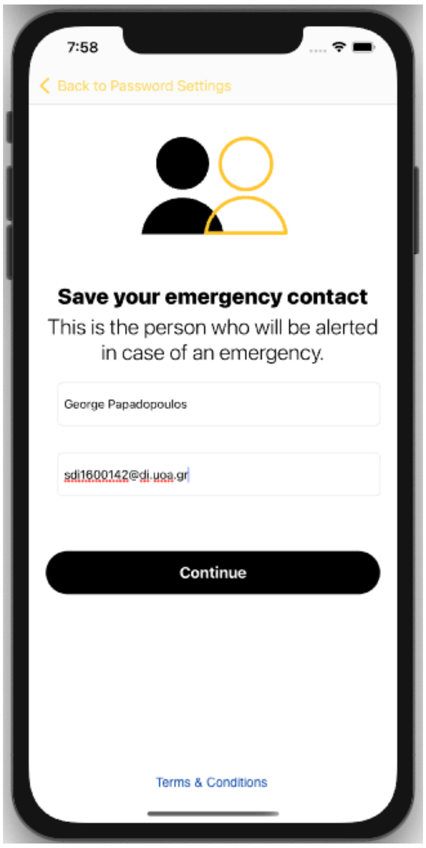

Image 3.6: Emergency contact's information.

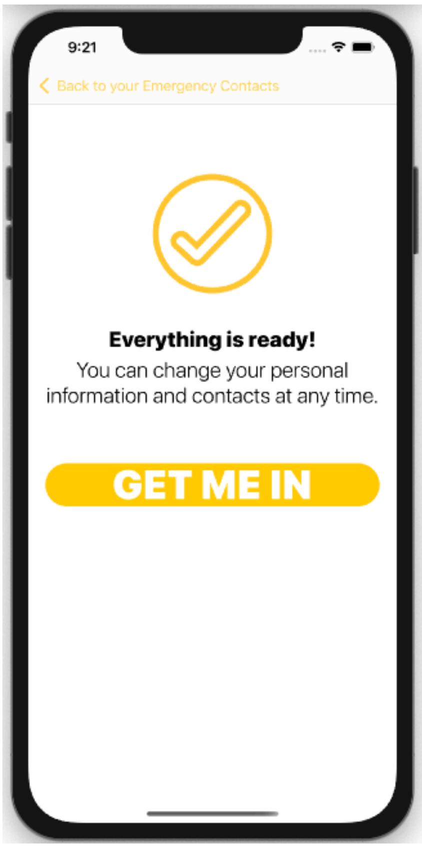

Image 3.7: The last screen before ending registration process.

3.3.3 Sign In and Sign Up Screens

After the very first launch of the app, each time the user tries to gain access to its content, they have to either sign in or sign up (if they would like to create a new account). In the log in screen (Image 3.8), if they enter invalid credentials, they will get a message informsing them about the error that happened (Image 3.9). Whereas, in case the credentials are valid, the user will successfully enter the app and see the Home Screen (Image 3.10).

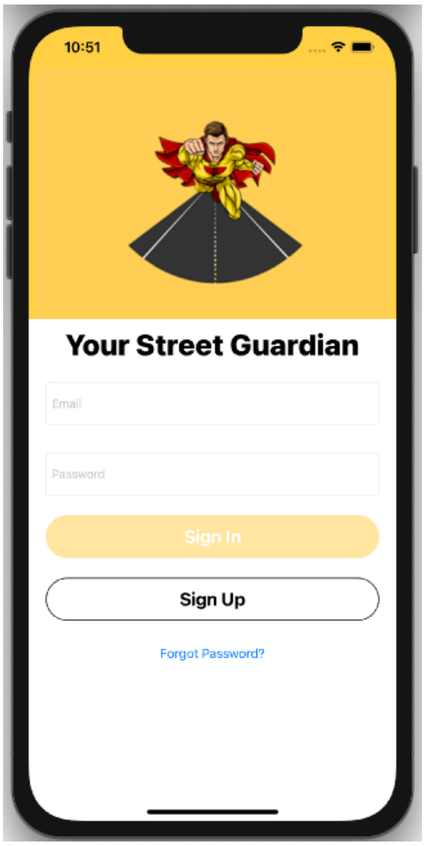

Image 3.8: The app's log in screen.

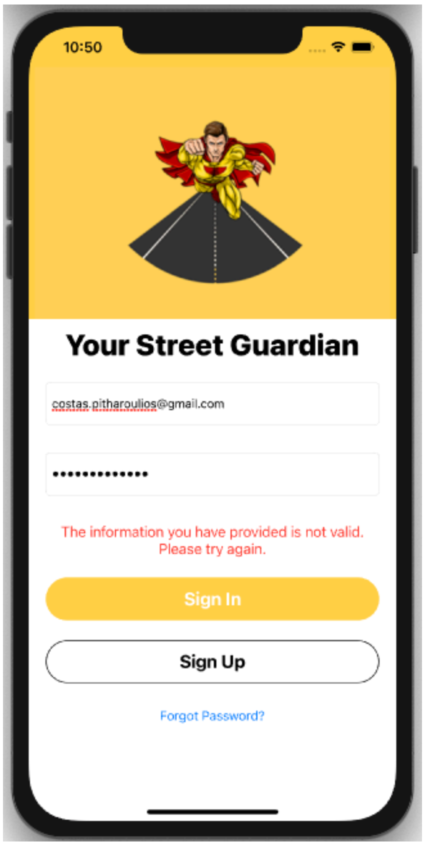

Image 3.9: Error message when wrong credentials are given.

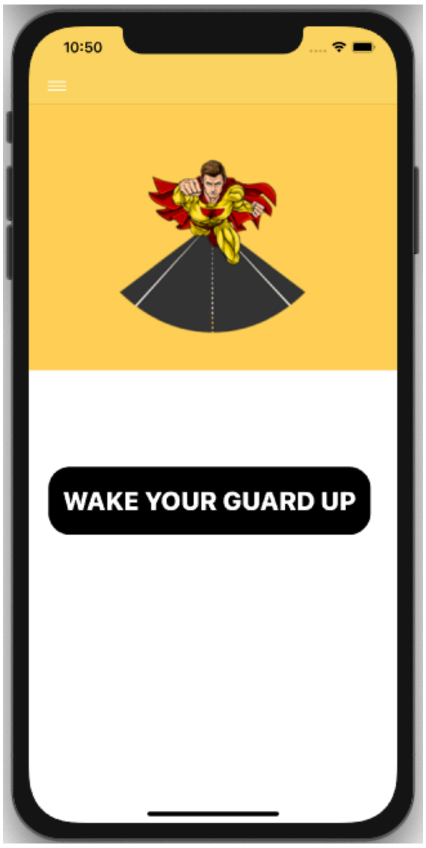

Image 3.10: iPhone Application Home Screen.

There is also the case that the user wants to create a new account. By pressing the "Sign Up" button (Image 3.8), they are able to follow the registration procedure, which was described in paragraph 3.2.1 above.

Last but not least, there is a button called "Forgot password?" which a user who has forgotten their password can press (Image 3.8). Then a new screen shows up where they can enter their email address (Image 3.11). If this email address given is already registered in the system, then the user instantly receives an email with all the details they need in order to define a new one (Image 3.12). In any other case, if for instance the email which was given is not registered, an error message is shown (Image 3.13).

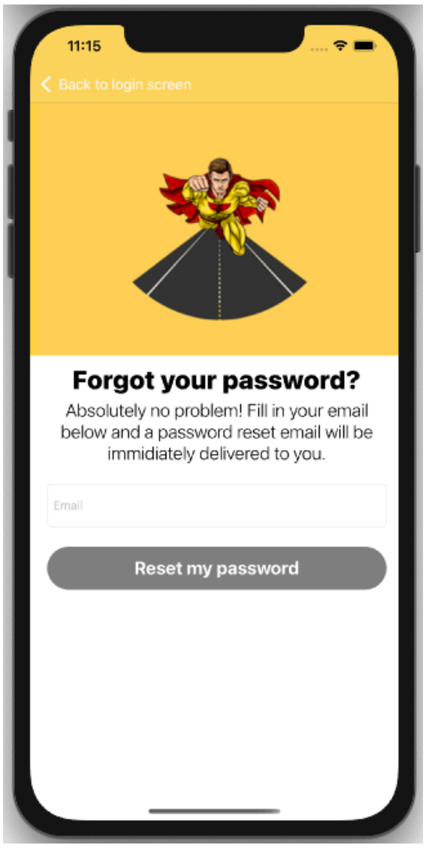

Image 3.11: Here the user can fill in their email to change their password.

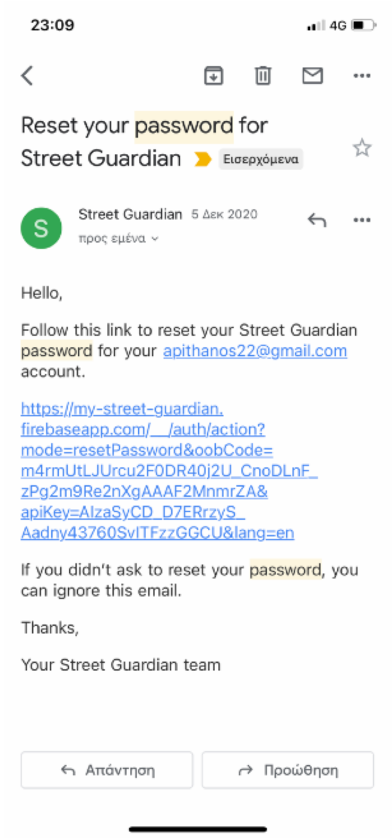

Image 3.12: The email for password renewal the user receives.

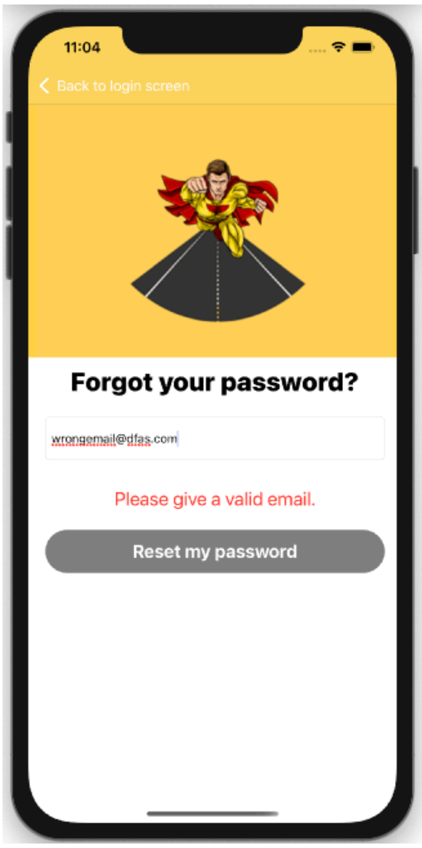

Image 3.13: The error message when the email given is not registered.

3.3.4 Menu and Categories

The user can browse through all the Menu Categories (Image 3.14) by simply swiping right or pressing the hamburger button at the top left corner (Image 3.10). By using it, the user can have instant access to and change any of their personal and emergency contact information which are stored in the online database. After finishing with the changes, the database gets automatically updated.

There is also an "Options" category where the users are able to choose on their own which sensors on their iPhone and Apple Watch devices best fit their lifestyle in order to minimise false alarms (Image 3.17). Let us consider, as an example, having an individual who usually keeps their phone in their pocket. Then for this particular person, the Rollover Sensor is going to be triggered each time they move their feet. And that is mainly because the Gyroscope Sensor will detect alarming orientation changes. Thus, it would be better for this kind of user to switch off the "Vehicle Rollover" detection. However, for a person who uses a steady holder for their phone, this sensor could be proven as a life saver.

Finally you can notice that there is a red button called “Log Out”, which the user can press whenever they need to exit their account.

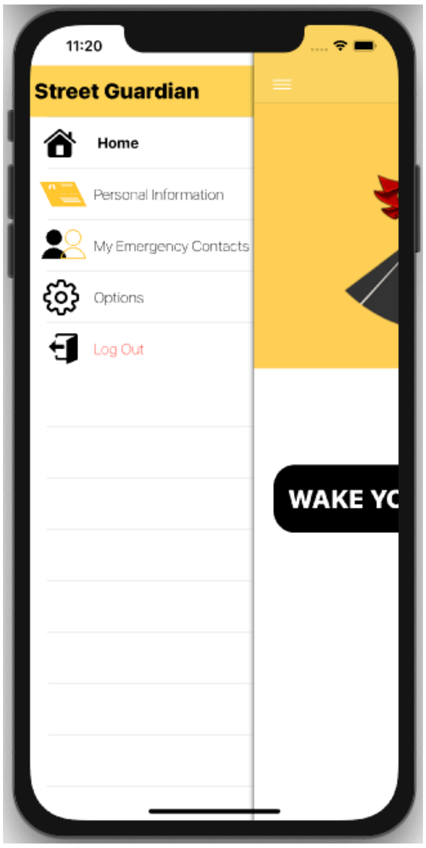

Image 3.14: This is the menu.

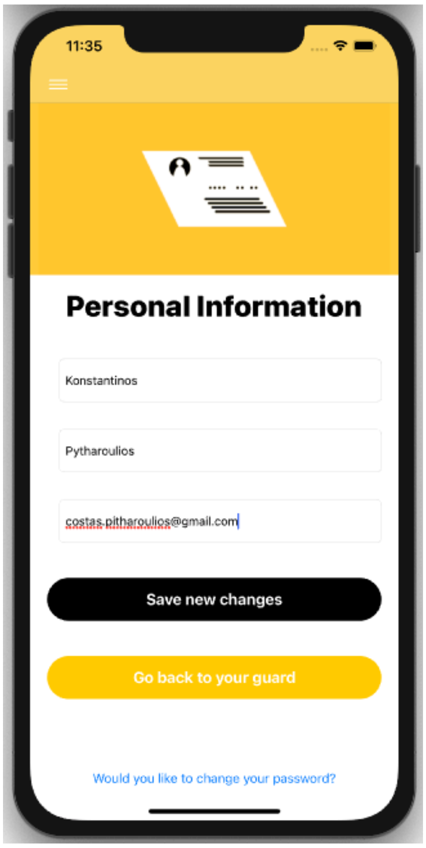

Image 3.15: Hereto use can change any of their personal. information.

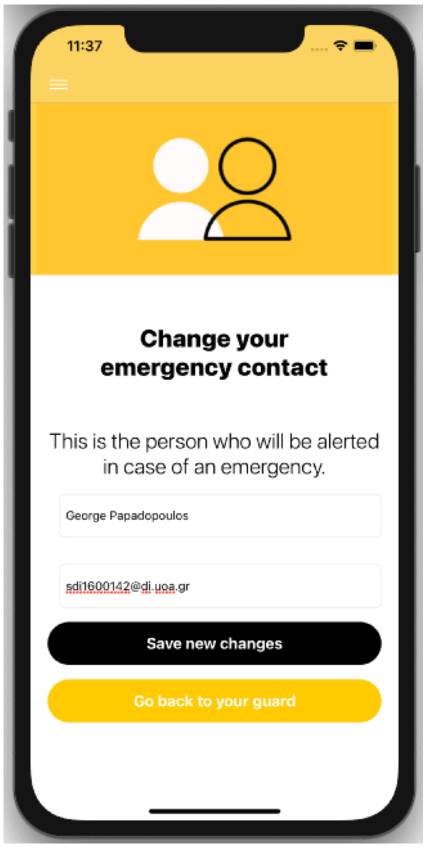

Image 3.16: Here the user can change any of their emergency contact information.

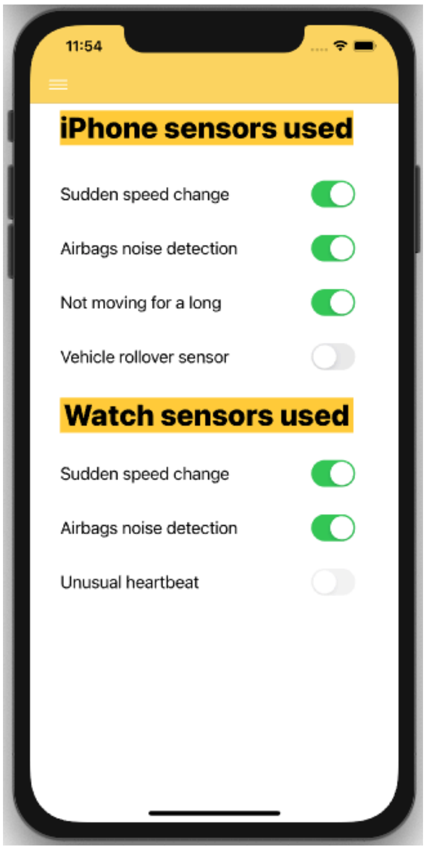

Image 3.17: Options Screen.

3.3.5 Tracking Screen

As we beforehand mentioned, the user has each time they want to use the application to choose between initialising it on iPhone or Apple Watch. These two devices work independently, so it is just up to the user's preference and convenience. The only step they have to take, either device they choose, is to press the button "Wake your guard up" on the home screen (Image 3.18 & 3.21). After this, the device immediately begins to show their position, tracks them and collects measurements from the sensors (Image 3.20 & 3.23).Of course, if this is the first time they are trying to use the tracking feature, they will be asked to give their permission (Image 3.19 & 3.22).

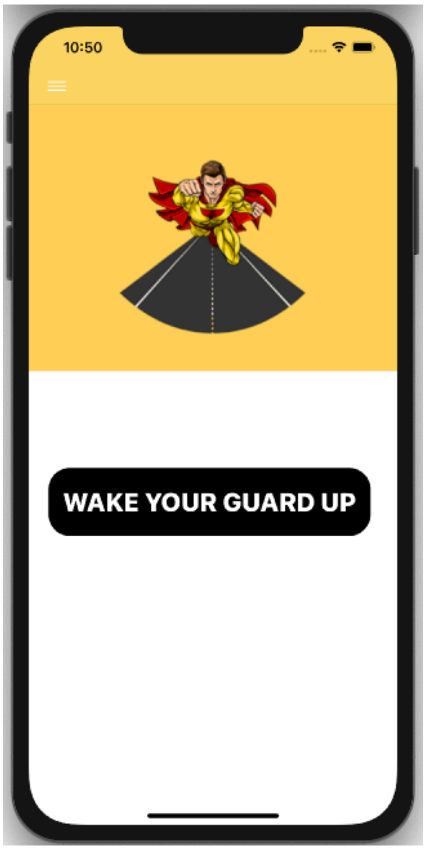

Image 3.18: iPhone Home screen.

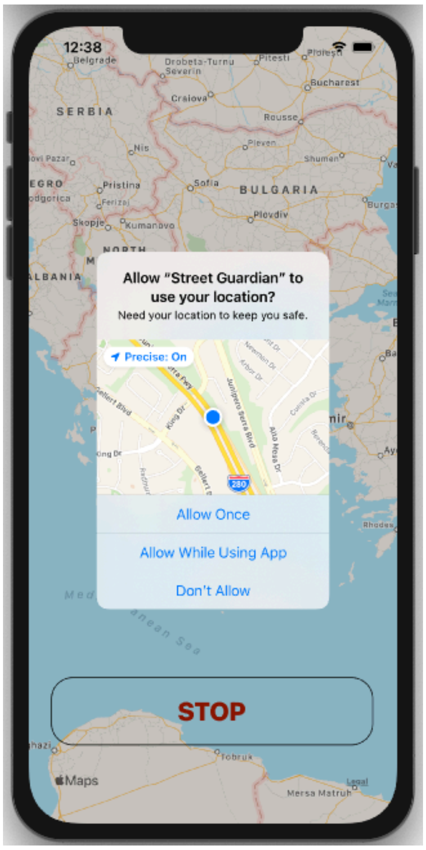

Image 3.19: iPhone asking for location use permission.

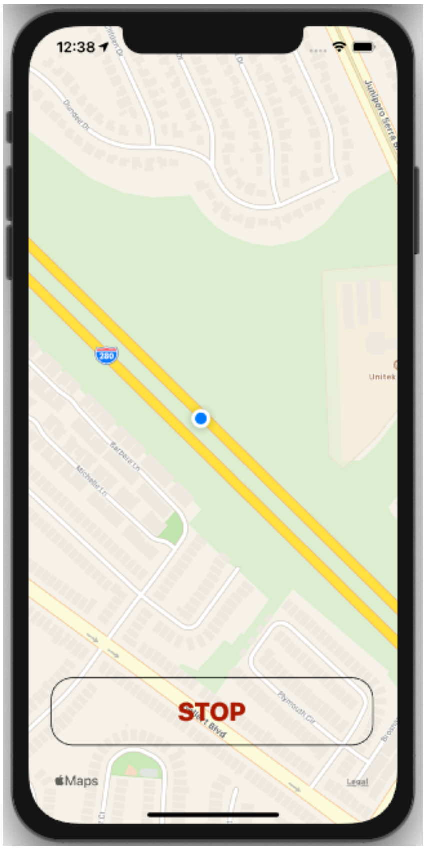

Image 3.20: iPhone is tracking the user.

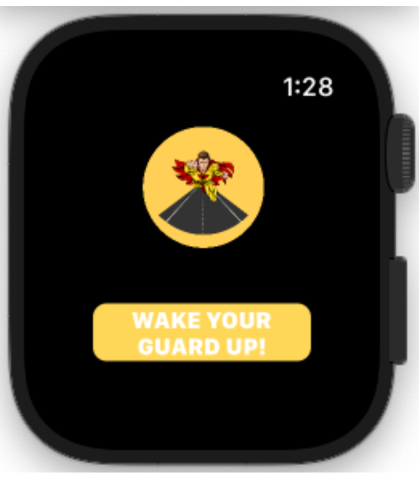

Image 3.21: Apple Watch Home screen.

Image 3.22: Apple Watch asking for location use permission.

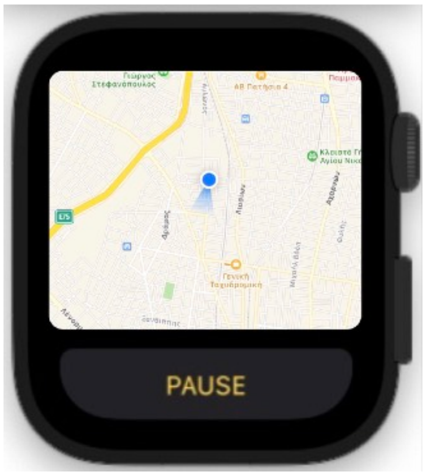

Image 3.23: Apple Watch is tracking the user.

3.3.6 Alarm Screens

Now let us suppose that the sensors have just detected some abnormal changes. Then, in this case, the tracking will stop and an alarming noise will start blaring. The user then has 30 seconds to press a button, which will inform their iPhone or Apple Watch whether they are safe or not (Image 3.24 & 3.26). If the 30 seconds time period expires without any user's response, the application will automatically move on to the next stage, which is to inform the emergency contacts about the prospective car accident, sending an email with all the crucial information (location, altitude, date, time and more) (Image 2.25 & 3.26).

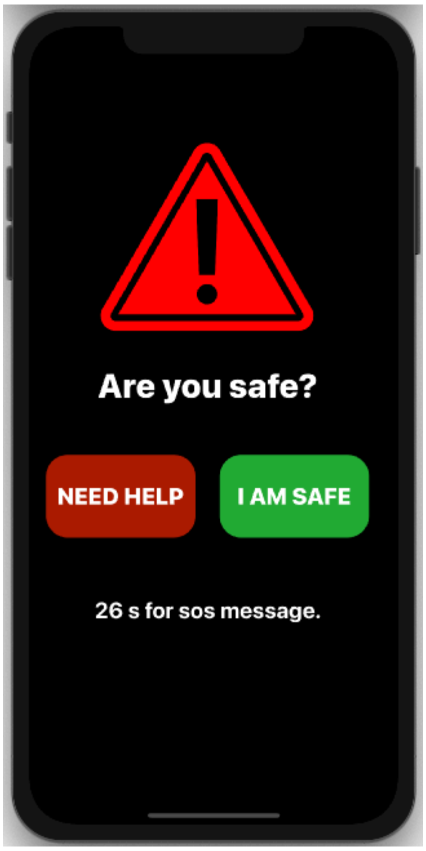

Image 3.24: The iPhone is giving 30s for the user to respond.

Image 3.25: The iPhone is now calling for emergency.

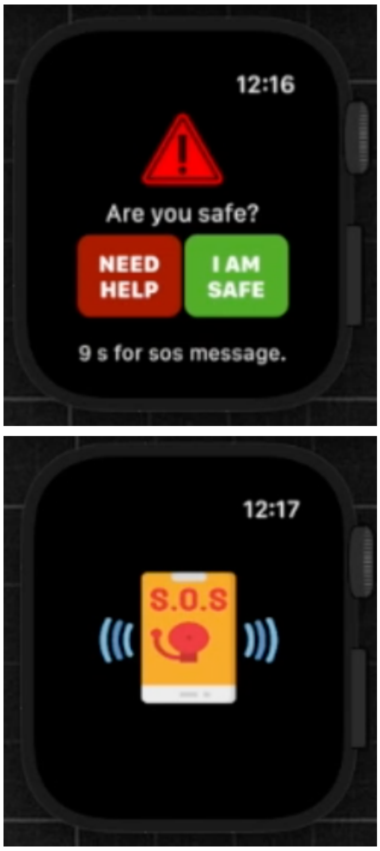

Image 3.26: The Apple Watch asks the user if they are safe and then it calls for emergency.

3.4. Technologies

To begin with the technologies used, the project is mainly built on Xcode which is a development environment dedicated to building applications running on Apple devices (Image 3.3) [13]. The language which was used, is Swift, which is an objective one [14]. Firebase is also used as a Database and Online Service. Especially for the second part, a Node.js function is built and integrated to Firebase [15] [16].

For a more descriptive depiction of the technologies used, let us use a case study. Suppose that you are an observer of an imaginary person, called Bob. Bob is the owner of an iPhone and an Apple Watch device, which both have the "Street Guardian" application preinstalled. He has just got on his car and he is now going to begin a long journey. He feels somewhat afraid of the idea that he is alone and relatively unsafe. For this reason, he decides to open up his iPhone and have the "Street Guardian" keep an eye on him while driving. After about two hours of driving on the highway, he accidentally falls asleep and inevitably loses control of his car. In no time, his car has crashed on the sidebar and the airbags have been opened. The sensors detect the sudden speed change as well as the massive noise of the explosion of the airbags. Subsequently his phone immediately starts the alarming noise waiting for his response. This never comes, so the phone moves on to inform his emergency contacts about the accident.

But how does this actually work? The answer is simple. Firstly the sensors detect an abnormal event and as a consequence, the iPhone establishes a connection with the Firebase, which will handle the rest of the processes. So, Firebase receives a message from the iPhone, including all the accident information (location, altitude, date, time). This information will be stored inside the database of accidents and as soon as this happens, a function will be triggered. This function, which is written in Node.js and is connected to Firebase, finds the driver's emergency contact information, which has already been stored in the database. After doing this, an email is sent to this emergency contact containing all the information that is going to be needed.

In case that Bob had initially opted for using his Apple Watch Street Guardian Application instead of the iPhone's one, then one more step would have been added to the whole procedure. More specifically, the Watch would have to send a message to the iPhone with the detected accident details in order for the phone to continue with the same process described above [17]. Since Apple does not allow the Apple Watch to connect with the Firebase directly, this additional step is necessary [18]. Fortunately enough, the iPhone is just running a background process that simply waits for a message from the Apple Watch to start the app and do the rest. If this message never comes, the iPhone application will not run; consequently, there will not be a drain on its sources.

Another significant factor which should definitely be taken into consideration, is that the main intention while building this application was to make a separation between the processes that should be held on the devices and those which can be completed in any other environment. This is based on the general belief that the use of memory, battery and processors of the device the user has on their hand should be minimised. In order to achieve this, online services are used and most specifically, Firebase. The Firebase provides us with a handy database where we can store all the users' and accidents' information. We can also have fast and easy authentication each time someone tries to log in. Additionally, we can connect our own Node.js functions, which are triggered after particular events.

In image 3.27 and image 3.28 a sample of Firebase's control panel is given. Both of them show the "Cloud Firestore" which serves as the application's database since all information collected is stored there.

The first column is named "Collections" and contains all categories of information. In that case, we have a category called "EmergencyLocations" where all the accidents' details are being kept and another one called "users", which holds all the personal information given by the registered users. The second column, called "Documents" keeps the key of each one of the instances in the third column. The third column shows all the fields of the document chosen in the second column.

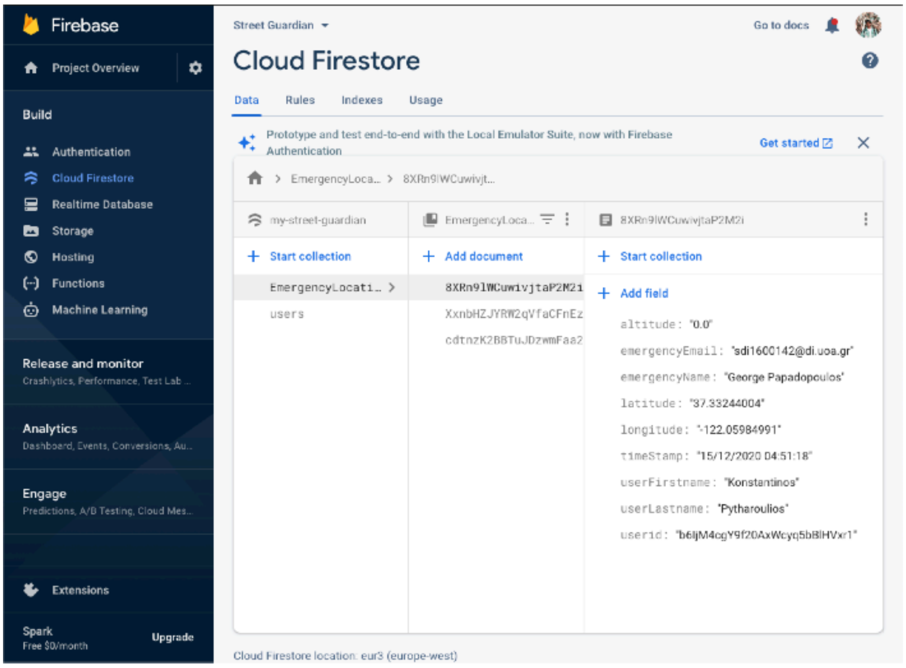

Image 3.27: An instance of an accident stored in Firebase.

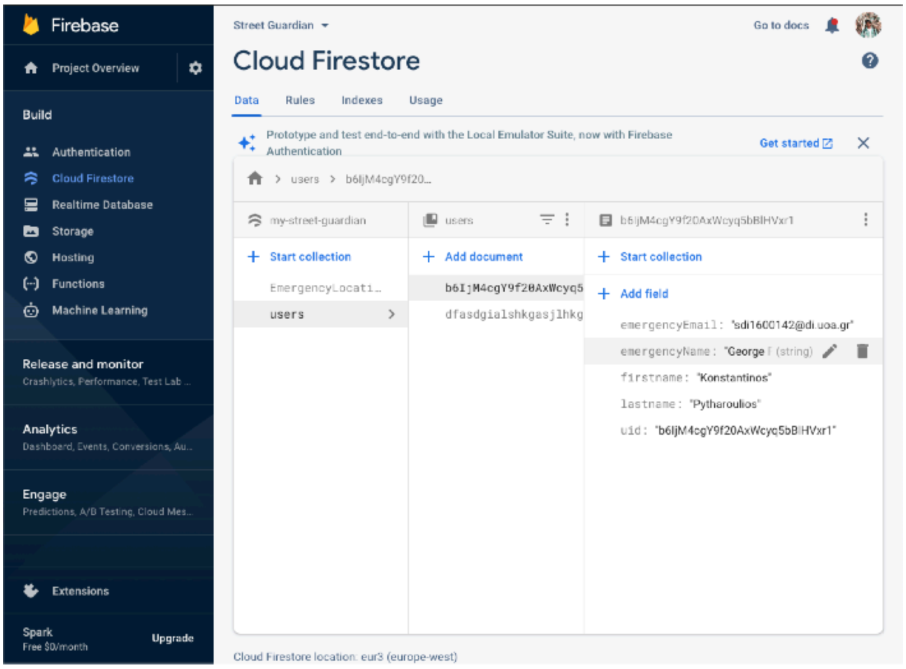

Image 3.28: An instance of a user's stored information in Firebase.

In the image 3.27 there is a collection of documents and each document contains details of a specific car crash that was detected by the app, whereas in the image 3.28 there is a collection of documents with users' personal information.

What is being shown in the Image 3.29 is the Authentication Center. The panel is connected with the registration form inside the app. This means that the panel gets updated with all the new credentials created after a new user's registration. The password is stored in an encrypted version that serves as a security measure which guarantees user's safety since not even the moderator of the panel is able see the actual password [19] .

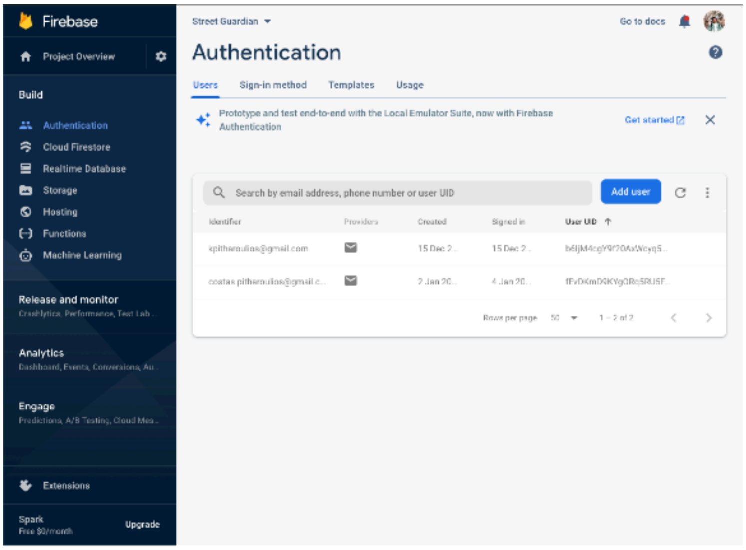

Image 3.29: Firebase Users Authentication Panel.

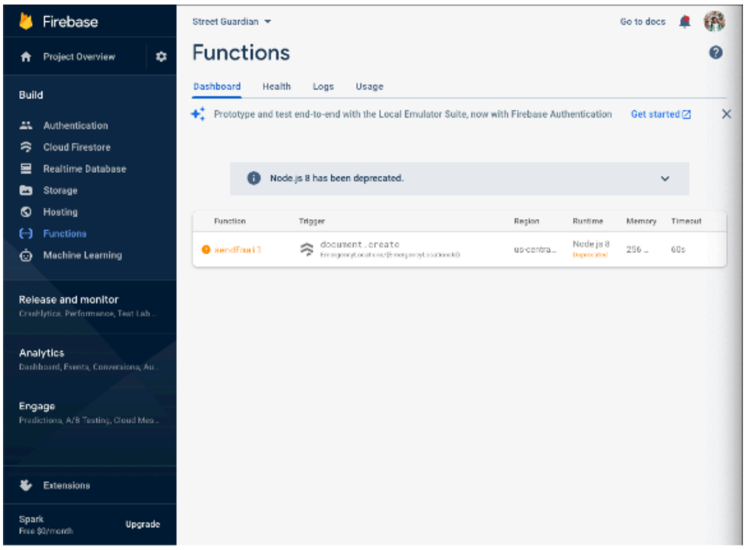

Image 3.30: Firebase Function.

The image 3.30 illustrates the integrated Node.js function which is triggered each time a new detected accident is stored in the database. This function finds the driver's emergency contact in the database and sends them an email containing all the information needed for their quick response (Image 3.31).

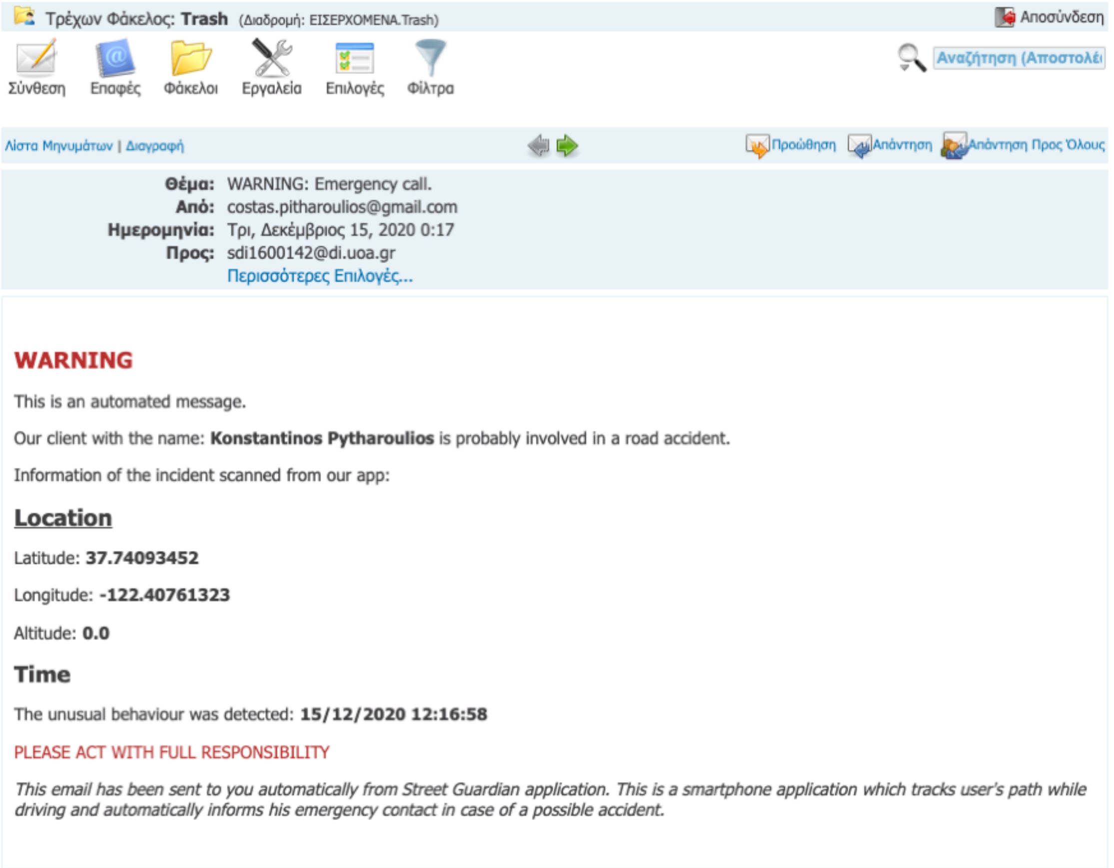

Image 3.31: The email that driver's emergency contact receives.

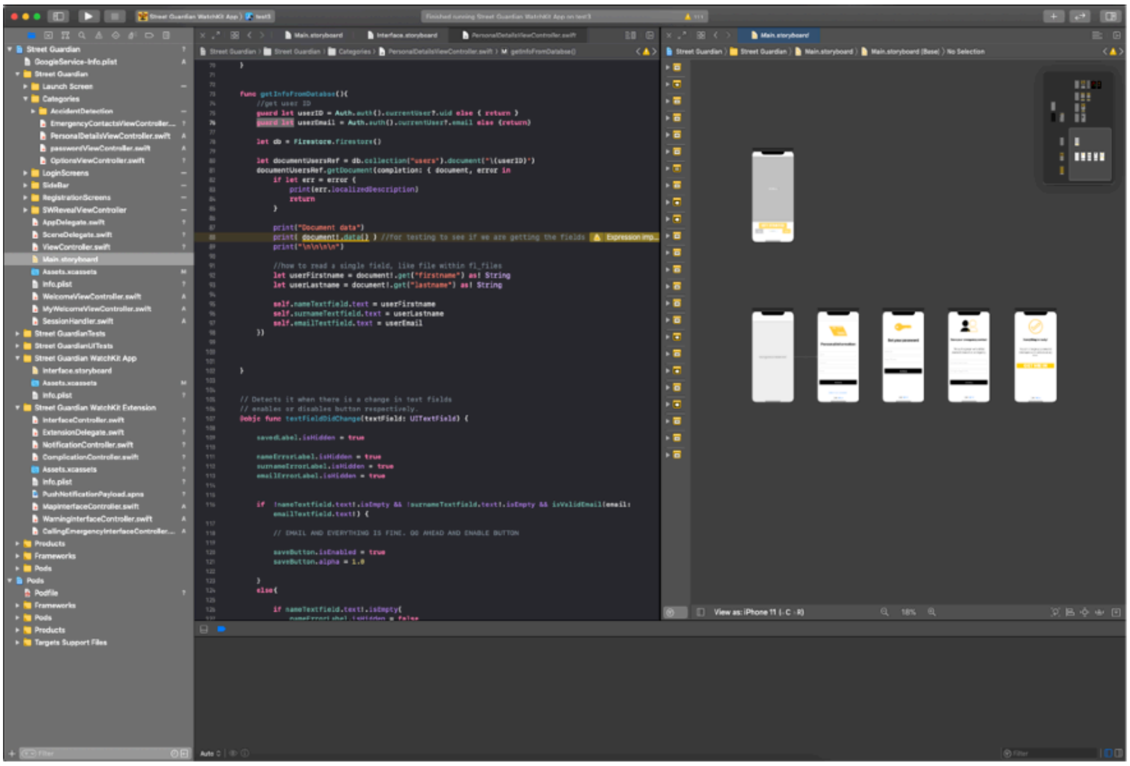

Image 3.32: The Xcode developing environment.

3.5. Sensors

One characteristic that makes this application special is that it combines measurements from different sensors to detect a prospective road accident. Sometimes, just a single measurement of one sensor can be enough to imply that a car crash has occurred. In contrast, in other cases, two or more different sensors are needed in order to show that something probably wrong has happened. We will first take a closer look at each one of the sensors used and then we will describe the different combinations used.

3.5.1 Global Position System (GPS)

The Global Position System (GPS) provided by Apple is a really sufficient system which is able to provide an application with extremely accurate measurements concerning latitude, longitude, altitude, even speed [20]. What it makes this GPS system extremely useful though, is the fact that its measurements are getting updated every single second. So this means that the "Street Guardian" application can have really firm data so as to make its calculations and estimations.

More specifically, the application tracks the speed from second to second. If a 20% or more speed change is observed from a second to the next one, the application will be immediately alarmed. This calculation is useful, especially for highways when a collision can happen. In accidents like this, cars usually lose much of their speed in a brief time period.

Additionally, if the speed is noticed to remain at zero for more than 30 minutes, that will also be an alarming state of affairs. The reason for this is the fact that the driver is more likely to stop the application if they stop driving. Thus, leaving the application running for thirty minutes without their car moving gives the sense that the driver probably had an accident that caused his car to stop somewhere and them to lose their conscience.

3.5.2 Microphone

Both iPhone and Apple Watch devices have a microphone embedded in them [21]. The application uses these microphones in order to record sounds inside the car cabin while the user is driving. According to researches conducted, when an airbags mechanism explodes in a car accident, the sound created by this is almost 165db. Moreover, 155db is a loudness which is almost never reached in a normal situation with even 5 passengers arguing [22]. Thus, the audio recording can be considered a real game-changer since it guarantees that the device will become aware each time airbags are on. It is good to have in mind that cars have a great number of sensors on their own, which are responsible for firing the airbags when they detect an accident. So now it is like the microphone is indirectly taking advantage of all those car sensors.

3.5.3 Gyroscope

The Gyroscope is a sensor that almost all smartphones contain [23]. What it basically does is to understand the phone's position in the three-dimensional world, so as to rotate the phone's screen respectively. However, this application tries to give a different role to this sensor. More specifically, it uses the Gyroscope in order to detect a rollover of the car. The only important detail for its proper functionality, is the phone to be mounted on a stable position in the car (for example, think of a phone stand). If the car accidentally rolls over, the phone will experience this as a reversion on the y axis. It is evident, though, that if the phone is inside the user's pocket, there will be many false-positive assumptions. Thus, it is not suggested for this kind of users who would rather leave the phone inside their pockets or move it very often while driving. Finally, the Gyroscope is embedded only in the iPhone, so Apple Watches are excluded from this feature.

3.5.4 Sensor Combinations

As we have already mentioned, not all sensors' measurements are convincing enough for the application to extract some conclusions. Of course, speed change and noise are not considered to belong to this category of events since they are solid evidence. Nevertheless, continuous stillness is not a very worrying situation. Additionally, the Gyroscope is a really sensitive sensor which can provide a lot of false positive predictions. For this reason, these two situations are combined. So, in order whether the indication of the Gyroscope that the car has rolled over is legit, this must be combined with 5 minutes of absolute inactivity (zero speed). Because if the application has an indication of a rollover incident, but the car still moves for a long, then it is almost certain that it was a false alarm.

4. FUTURE WORK

4.1 Access to Medical Information Stored on the iPhone

Apple offers its users an application which is called HealthKit [24]. In this application, they can store different medical information like blood type, age, illnesses, allergies and more. This information can be used in various ways when the user experiences a dangerous situation. Based on this, a possible cooperation between HealthKit and Street Guardian could be a true life saver. At this moment, the only thing that the Street Guardian application can do is to provide the emergency contacts with information about the location and the time period of an accident including no other driver's information except from their name. This is something negative since details like blood type could help the rescuers be well prepared before reaching the place of the accident.

4.2 Cooperation with eCall

From April 2018 all cars sold in the European Union are obliged to have a feature called “eCall”. The eCall automatically dials Europe's single emergency number 112 in the event of a serious road accident and communicates the vehicle's location to the emergency services[25]. This system relies on sensors and car’s airbags. Therefore it would be an excellent prospective, if Street Guardian application could use the same sensors to have more accurate indications or at least if these two similar technologies could work complementary. For example, eCall could detect and communicate the accidents data and the Street Guardian could inform user’s emergency contacts about the incident as well as send extra details about driver’s health state to the emergency services.

4.3 Use of Artificial Intelligence for Minimising False Positives

A nice idea with some excellent potential would be to use Artificial Intelligence. The application could be pre-trained in real-life situations in order to learn specific cases that look like accidents but are not. In addition to that, each time a possible accident is detected and the user is asked if they are safe, the data collected from the cases when the driver is actually safe could also train the algorithm.

4.4 Use Phone Calls to Make Emergency Contacts Aware

Whenever an accident happens, the emergency contact that the driver has predefined is getting informed about the accident via an email. However, no one could doubt that most people do not really check their emails very often. Thus, it would be great if the emergency contacts were also informed via a message on their phones or even an automated phone call.

4.5 Use of Heart Beat Sensor on Apple Watch

All Apple Watches include a sensor which is able to track the user’s heart beat. Having in mind that most of the times, when a serious accident has happened the heart rate changes abnormally, it would probably be a smart idea for the application to use those metrics in order to be able to understand when a driver is not well after an accident.

4.6 Android Support

The application is currently running on iOS devices. This is a limiting factor for those people who use Android products. So an extremely positive next step would definitely be to develop a similar application compatible with those devices. However, the sensors especially on Android Watches are not so efficient as those on Apple ones, thus there is a considerable amount of uncertainty about the final product’s usefulness.

5. APPENDICES

This section includes the analyses of some application's crucial functions. The whole code of the application has been uploaded on Github for further inspection (https://github.com/CostasPitharoulios/StreetGuardian.git).

5.1 Firebase function for emergency email notification

This is the implementation of a Node.js function integrated on Firebase. Its role is to automatically get triggered each time the details of a new accident are stored in the database. It gets the user's key, finds user's emergency contact information in the database and finally sends them an emergency email notification including all the incident's necessary information (location, driver's personal information, application's role, etc.).

```
const functions = require('firebase-functions');
const admin = require("firebase-admin")
const nodemailer = require('nodemailer');

admin.initializeApp()

//google account credentials used to send email
var transporter = nodemailer.createTransport({
    host: 'smtp.gmail.com',
    port: 465,
    secure: true,
    auth: {
        user: 'costas.pitharoulios@gmail.com',
        pass: 'yabcdpjmj'
    }
});

exports.sendEmail = functions.firestore
    .document('EmergencyLocations/{EmergencyLocationsId}')
    .onCreate((snap, context) => {

        const mailOptions = {
            from: `costas.pitharoulios`,
            to: `${snap.data().emergencyEmail}`,
            subject: 'WARNING: Emergency call.',
            html: `<p><strong><span style="font-size: 18px; color: rgb(184, 49, 47);">WARNING</
span></strong></p>
            <p>This is an automated message.</p>
            <p>Our client with the name:<strong> ${snap.data().userFirstname} $
{snap.data().userLastname} </strong>is probably involved in a road accident.</p>
            <p>Information of the incident scanned from our app:</p>
            <p><u><strong><span style="font-size: 18px;">Location</span></strong></u></p>
            <p>Latitude: <strong>${snap.data().latitude}</strong></p>
            <p>Longitude: <strong>${snap.data().longitude}</strong></p>
            <p>Altitude: <strong>${snap.data().altitude}</strong></p>
            <p><strong><span style="font-size: 18px;">Time</span></strong></p>
            <p>The unusual behaviour was detected: <strong> ${snap.data().timeStamp}</strong></p>
            <p><span style="color: rgb(184, 49, 47);">PLEASE ACT WITH FULL RESPONSIBILITY</span></
p>
            <p><em>This email has been sent to you automatically from Street Guardian application.
This is a smartphone application which tracks user&apos;s path while driving and automatically
informs his emergency contact in case of a possible accident.</em></p>
            <p><br></p>`
        };

        console.log("Hellooooo!\n\n")

        return transporter.sendMail(mailOptions, (error, data) => {
            if (error) {
                console.log(error)
                return
            }
            console.log("Sent!")
        });
    });
```

Image 5.1: Function integrated on Firebase for sending emergency email notifications.

5.2 Code for detecting abnormal speed changes

The code below detects abnormal speed changes. An abnormal case is either an over 20% speed change between two seconds or when the car has come to a halt for more than half an hour.

```
    func locationManager(_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) {

        let locationArray = locations as NSArray

        location = locationArray.lastObject as! CLLocation

        userLocationInfoSTART.removeAll(keepingCapacity: true)
        userLocationInfoSTART.append("\(location!.coordinate.latitude)")
        userLocationInfoSTART.append("\(location!.coordinate.longitude)")
        userLocationInfoSTART.append("\(location!.speed)")
        userLocationInfoSTART.append("\(location!.timestamp)")

        // IF we have a sudden speed change        // <OR>
        // IF the car has not moved for over half an hour
        // SEND WARNING MESSAGES
        if isSuddenSpeedChange(speedStart: userLocationInfoEND[2], speedEnd:
userLocationInfoSTART[2]) || isNotMovingForALong(speedStart: userLocationInfoEND[2], speedEnd:
userLocationInfoSTART[2]) {
            // True: The case that an abnormal changing of speed was detected
            print("WARNING: Sudden speed change was detected!!\n\n")

            // stop location detection
            [self.locationManager .stopUpdatingLocation()]

            //stop decibel counting
            finishRecording()



            // Store latitude in user defaults
            UserDefaults.standard.setValue("\(location!.coordinate.latitude)", forKey:
"latitudeEmergency")
            UserDefaults.standard.setValue("\(location!.coordinate.longitude)", forKey:
"longitudeEmergency")
            UserDefaults.standard.setValue("\(location!.altitude)", forKey: "altitudeEmergency")

            //performSegue(withIdentifier: "fromMapToWarning", sender: self)
            performSegue(withIdentifier: "fromMapToWarning", sender: self)

        }
        else{
           // FALSE: EVERYTHING SEEMS FINE WITH THE SPEED

            print(userLocationInfoEND)
            print(userLocationInfoSTART)
            print("\n\n")

            // Speed before, becomes speed after.
            userLocationInfoEND = userLocationInfoSTART

            let coordinate = CLLocationCoordinate2D(latitude: location!.coordinate.latitude,
longitude: location!.coordinate.longitude)

            let span = MKCoordinateSpan(latitudeDelta: 0.005, longitudeDelta: 0.005)

            let region = MKCoordinateRegion(center: coordinate, span: span )

            map.setRegion(region, animated: true)
        }
    }
```

Image 5.2: The locationManager function which calls abnormal speed detection functions and handles location data.

The *isNotMovingForALong* function checks whether the car is not moving for a suspicious time period. If 5 minutes have passed without a move, the gyroscope sensor will check if the the car has rolled over.

```
    // This is a function which checks if the car is not moving for a
    // a long time which is something suspicious.
    func isNotMovingForALong(speedStart: String, speedEnd: String) -> Bool{

        // division with / 1000.0 * 60.0 * 60.0 converts speed to km/h
        let doubleSpeedStart = Double(speedStart)! / 1000.0 * 60.0 * 60.0
        let doubleSpeedEnd = Double(speedEnd)! / 1000.0 * 60.0 * 60.0

        if doubleSpeedStart == 0.0 && doubleSpeedEnd == 0.0{
            self.secondsOfNotMoving = self.secondsOfNotMoving + 1 // one more second of not moving

            if self.secondsOfNotMoving == 1800{
                // IF THE CAR HAS BEEN STOPPED FOR HALF AN HOUR...
                return true
            }
            else if self.secondsOfNotMoving == 300{
                // IF THE CAR HAS BEEN STOPPED FOR 5 MINUTES CHECK WHETHER IT IS UPSIDE DOWN OR NOT

                /* GOING TO USE GYROSCOPE*/

                var uMM: CMMotionManager!
                uMM = CMMotionManager()
                uMM.accelerometerUpdateInterval = 0.2

                //  Using main queue is not recommended. So create new operation queue and pass it
to startAccelerometerUpdatesToQueue.
                //  Dispatch U/I code to main thread using dispach_async in the handler.
                uMM.startAccelerometerUpdates( to: OperationQueue() ) { p, _ in
                    if p != nil {
                        if ((p?.acceleration.y)! >= (p?.acceleration.x)!){
                            // Phone is in portrait mode
                            if (p?.acceleration.y)! > 0 {
                                // MAYBE THE CAR IS UPSIDE DOWN
                                print("WARNING: MAYBE THE CAR IS UPSIDE DOWN")
                                self.secondsOfNotMoving = 0
                            }
                        }

                    }
                }
            }
        }
        else{
            self.secondsOfNotMoving = 0
        }

        return false
    }
```

Image 5.3: Function which checks for car's extensive inactivity.

5.3 Code for detecting abnormal car interior sound

The *isSuddenSpeedChange* function bellow (image 5.3) checks if there is a speed change greater than 20% between two seconds which is a concerning factor, since that would be a rather abnormal behaviour.

```
// This is the function which checks the changing of speed.
   // Returns TRUE: When is has detected an abnormal changing in speed.
   // Returns FALSE: When everything seems ok.
   func isSuddenSpeedChange(speedStart:String, speedEnd:String) -> Bool{

       // division with / 1000.0 * 60.0 * 60.0 converts speed to km/h
       let doubleSpeedStart = Double(speedStart)! / 1000.0 * 60.0 * 60.0
       let doubleSpeedEnd = Double(speedEnd)! / 1000.0 * 60.0 * 60.0

       print(doubleSpeedEnd)

       // If the speed gets reduced by 20% or more, then we have a possible car crash
       if (doubleSpeedStart-doubleSpeedEnd) >= (doubleSpeedStart * 0.2){
           // We have a noticeable speed change

           //print("Start: \(doubleSpeedStart)")
          // print("Stop: \(doubleSpeedEnd) \n\n\n\n")
           return true
       }
       else{
           return false
       }

   }
```

Image 5.4: Function which checks for car’s sudden speed change.

In the next two code parts, two crucial functions are being presented. They are both responsible for the audio recording while the car is moving. The *startRecordingAudio* function makes all the necessary steps for audio recording initialisation process. Whereas, the *soundBarrierExceeded* function keeps track of the noise inside the car and responds to sudden high increases.

```
/* STARTS AUDIO RECORDING*/
   func startRecordingAudio(){

       let url = directoryURL() //else {
           //print("Unable to find a init directoryURL")
           //return false

       let recordSettings = [
           AVSampleRateKey : NSNumber(value: Float(44100.0) as Float),
           AVFormatIDKey : NSNumber(value: Int32(kAudioFormatMPEG4AAC) as Int32),
           AVNumberOfChannelsKey : NSNumber(value: 1 as Int32),
           AVEncoderAudioQualityKey : NSNumber(value: Int32(AVAudioQuality.medium.rawValue) as
Int32),
       ]

       let audioSession = AVAudioSession.sharedInstance()

       do {
           try audioSession.setCategory(AVAudioSession.Category.playAndRecord)
           audioRecorder = try AVAudioRecorder(url: url!, settings: recordSettings)
           audioRecorder.prepareToRecord()
           audioRecorder.record()
           try audioSession.setActive(true)
           audioRecorder.isMeteringEnabled = true
           recordForever(audioRecorder: audioRecorder)
       } catch let err {
           print("Unable start recording", err)
       }

   }
```

Image 5.5: Function for the initialisation of car's interior noise recording.

```
/* CHECKS IF SOUND BARRIER IS ECSEEDED - IF WE HAVE A CRASH SOUND*/
    func soundBarrierExceeded() -> Bool{
        // NOTE: seems to be the approx correction to get real decibels
        let correction: Float = 165
        let average = audioRecorder.averagePower(forChannel: 0) + correction
        let peak = audioRecorder.peakPower(forChannel: 0) + correction
        //self?.recordDatapoint(average: average, peak: peak)
        print("PEAK: \(peak)\n\n\n")
        if peak >= 115{
            // IF SOUND BARRIER HAS BEEN EXCEEDED
            print("WARNING: Very loud noise was detected!!\n\n")
            return true
        }
        else{
            // IF EVERYTHING IS OK KEEP GOING
            return false
        }
    }
```

Image 5.6: Function which detects abnormally high noise volume inside the car.

5.4 Code for connection establishment between Apple Watch and iPhone

As it has already been mentioned, if the user has chosen to run the Street Guardian application from their smartwatch, then their iPhone will likely have to handle some processes in case of an accident. More specifically, since the Apple Watch is not able to connect directly to the Firebase Services, the watch will send the accident details to the phone at first. Subsequently, the iPhone will establish a connection with the Firebase Server on the background in order to send the car crash report to it. In the code below, the function session is being presented. This one is in charge of creating a connection between the user's iPhone and Apple Watch. Unfortunately, if those two cannot pair, the accident will never be stored in the database, nor will the emergency contacts become informed about the situation.

```
    func session(_ session: WCSession, activationDidCompleteWith activationState:
WCSessionActivationState, error: Error?) {
        print("activationDidCompleteWith activationState:\(activationState) error:\
(String(describing: error))")

        // Sends message with details to phone
        /**
        *  The iOS device is within range, so communication can occur and the WatchKit extension is running in the
        *  foreground, or is running with a high priority in the background (for example, during a workout session
        *  or when a complication is loading its initial timeline data).
        */
        if isReachable() {
            /*session.sendMessage(["request" : "version"], replyHandler: nil, errorHandler:
{ (error) in
                print("Error sending message: %@", error)
            })*/
            session.sendMessage(locationData, replyHandler: { (response) in
                print("Reply: \(response) \n\n")
            }, errorHandler: { (error) in
                print("Error sending message: %@", error)
            })
            print("Iphone is reachable!")
        } else {
            print("iPhone is not reachable!!")
        }

    }
```

Image 5.7: Function which establishes a connection between the user's iPhone and Apple Watch.

ABBREVIATIONS - ACRONYMS

App	Application
GPS	Global Positioning System
USA	United States of America
db	Decibel

REFERENCES

[1] European Road Safety Observatory. “Annual Accident Report 2018”. europa.eu, https://www.who.int/violence_injury_prevention/road_safety_status/2015/TableA2.pdf?ua=1 (accessed Jan. 2, 2021).
[2] World Health Organization. “Estimated Traffic-Related Deaths 2015”. World Health Organisation, https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/pdf/statistics/dacota/asr2018.pdf (accessed Jan. 2, 2021).

[3] World Health Organisation, ed. (2018). Global Status Report on Road Safety 2018 (PDF)(official report). Geneva: World Health Organization (WHO). pp. xiv–xv, 1–13, 91ff (countries), 302–313 (table A2), 392–397 (table A11). ISBN 978-92-4-156568-4. (Retrieved Jan. 3, 2021). “Tables A2 & A11, data from 2016”

[4] World Health Organisation, ed. (2013). Global Status Report on Road Safety 2013: supporting a decade of action (PDF) (official report). Geneva, Switzerland: World Health Organization (WHO). pp. vii, 1–8, 53ff (countries), 244–251 (table A2), 296–303 (table A10). ISBN 978-92-4-156456-4. (Accessed Jan. 3, 2021). “Tables A2 & A10, data from 2010”

[5] "Reducing Pollution with Electric Vehicles". www.energy.gov, https://www.energy.gov/eere/electricvehicles/reducing-pollution-electric-vehicles (retrieved Jan. 2, 2021).
[6] Chauncey Alcorn. “Cyclist's Apple Watch with a 'fall detection feature' may have saved his life by calling 911 and alerting his son after he crashed his bike and was knocked unconscious”. Daily Mail, https://www.dailymail.co.uk/news/article-7505293/Washington-mans-Apple-Watch-calls-911-crashed-bike-crash-knocked-unconscious.html (accessed July 17 , 2020).
[7] Apple Inc. “Manage fall detection on Apple Watch”. developer.apple.com, https://support.apple.com/en-gb/guide/watch/apd34c409704/watchos (accessed Oct. 5, 2020)
[8] Chris Welch. “Google’s Personal Safety app detects car accidents and automatically calls 911”. The verge. https://www.theverge.com/2019/10/1/20892875/google-pixel-personal-safety-app-car-crash-detection (accessed Nov. 7, 2020)
[9] Apple Inc. “HealthKit Framework Reference”. developer.apple.com. https://developer.apple.com/documentation/healthkit (accessed Nov. 7, 2020)
[10] Techopedia. “Online Service”. Techopedia.com. https://www.techopedia.com/definition/3248/online-service (accessed Jan. 3, 2021)
[11] Doug Stevenson. “What is Firebase? The complete story, abridged.”. medium.com . https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0 (accessed Jan. 3, 2021)
[12] Kimberly de Silva. “The Importance of Onboarding Users in Your App”. BiznessAPPS . https://www.biznessapps.com/blog/importance-app-onboarding-users/ (accessed Jan. 3, 2021)
[13] Apple Inc. “Introducing Xcode 12”. developer.apple.com. https://developer.apple.com/xcode/ (accessed Dec. 23, 2020)
[14] Apple Inc. ”Swift”. developer.apple.com. https://developer.apple.com/swift/ (accessed Dec. 23, 2020)
[15] Firebase. “ Firebase helps you build and run successful apps”. Firebase. :https://firebase.google.com (accessed Dec. 23, 2020)
[16] Nodejs. “NodeJS Information” NodeJS.org. https://nodejs.org/en/ (accessed Dec. 23, 2020)
[17] Lito.”WatchOS 5 — Communication between iPhone and Apple Watch and vice versa on Swift — Part 2”. medium.com. availablehttps://medium.com/@litoarias/watchos-5-part-2-communication-between-iphone-and-apple-watch-and-vice-versa-ced41f453b11 (accessed Oct. 27, 2020)
[18] Hohteri. “Firebase Auth support for watchOS”. GitHub. https://github.com/firebase/firebase-ios-sdk/issues/4621 (accessed Dec. 4, 2020)
[19] Firebase. “Authenticate with Firebase using Password-Based Accounts on iOS”. Firebase. https://firebase.google.com/docs/auth/ios/password-auth (accessed Dec. 4, 2020)
[20] Apple Inc. “Core Location”. developer.apple.com. https://developer.apple.com/documentation/corelocation (accessed Nov. 12, 2020)
[21] Apple Inc. “AVAudioRecorder”. developer.apple.com. https://developer.apple.com/documentation/avfoundation/avaudiorecorder (accessed Nov. 14, 2020)
[22] Rachael Cayley. “Car Crashing and Noise”. The noise curmudgeon. https://thenoisecurmudgeon.wordpress.com/2017/03/18/cars-crashing-and-noise/ (accessed Sept. 11, 2020)
[23] Apple Inc. “Getting Raw Gyroscope Events”. developer.apple.com. https://developer.apple.com/documentation/coremotion/getting_raw_gyroscope_events (accessed Dec. 16, 2020)
[24] Apple Inc. “HealthKit”. developer.apple.com. https://developer.apple.com/documentation/healthkit (accessed Dec. 19, 2020)
[25] European Commission. "The interoperable EU-wide eCall”. ec.europa.eu. https://ec.europa.eu/digital-agenda/en/ecall-time-saved-lives-saved (accessed July. 4, 2020)