Road Hazard Proximity Alert system

Richard Rhodes
BSc in Computer Science

08/04/20

SCHOOL OF COMPUTING AND MATHEMATICS
Keele University
Keele
Staffordshire
ST5 5BG

Abstract

In this paper, the creation of a functional prototype system called the Road Hazard Proximity Alert system is discussed. This system allows users to report road hazards using a smartphone app, which communicates with a SQLite database. This means that hazards can be reported to other users that are driving, so they know to avoid the area or be more cautious.

The project was completed, but was unable to be evaluated due to the current COVID-19 global pandemic. In all other areas the project was a success.

Acknowledgements

Firstly I want to thank my supervisor Dr Theocharis Kyriacou, for helping me with the entire project, and making sure I understood the difference between a final 3rd year project and something less demanding.

Secondly, I want to thank my parents for encouraging the idea and also for letting me use them as guinea pigs for the user interface in the pre-alpha stage. Thanks also for checking through and proofreading the report.

Thirdly, I want to thank my testers and evaluators for testing and evaluating. Early user feedback is very important with any development.

Thirdly, I want to thank my testers and evaluators for testing and evaluating. Early user feedback is very important with any development.

I also want to thank the Social Distancing Bois (formerly the Breakfast Bois). You know who you are, and for helping me stay sane during development, and for putting up with my frustration when something didn’t work or broke.

Finally, I want to thank Domino’s Pizza for powering the developer. Also, not only delivering to Keele Campus, but for having the UNION123 code, and still delivering at 10pm.

Table of Contents

Glossary

RHPA - Road Hazard Proximity Alert system.

The name of the project discussed in this paper.

Road Hazard

A hazard is anything on a road which is likely to be dangerous or cause an accident. This could be a collision, a horse rider, road construction/repair, or anything that a driver would need to be aware of.

API - Application Programming Interface

A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.

GPS - Global Positioning System

System developed by the US military to aid navigation.

Git/GitHub

Git was developed by Linus Torvalds to assist with the development of the Linux Kernel. The system was then released as open source software under the GPL. GitHub is a free online implementation of Git.

1. Introduction

1.1 The Project

The aim of the RHPA project was to design and implement a system that had the potential to warn road users of new and existing road hazards, but to do so in a manner that did not cause added distraction to the road user. This would include both stationary hazards, i.e. accidents, and moving hazards, for example, horse riders, and large or wide loads. The developed system should be made easily available to a wide variety of road users. A sample overview of the RHPA system is shown in figure 1.

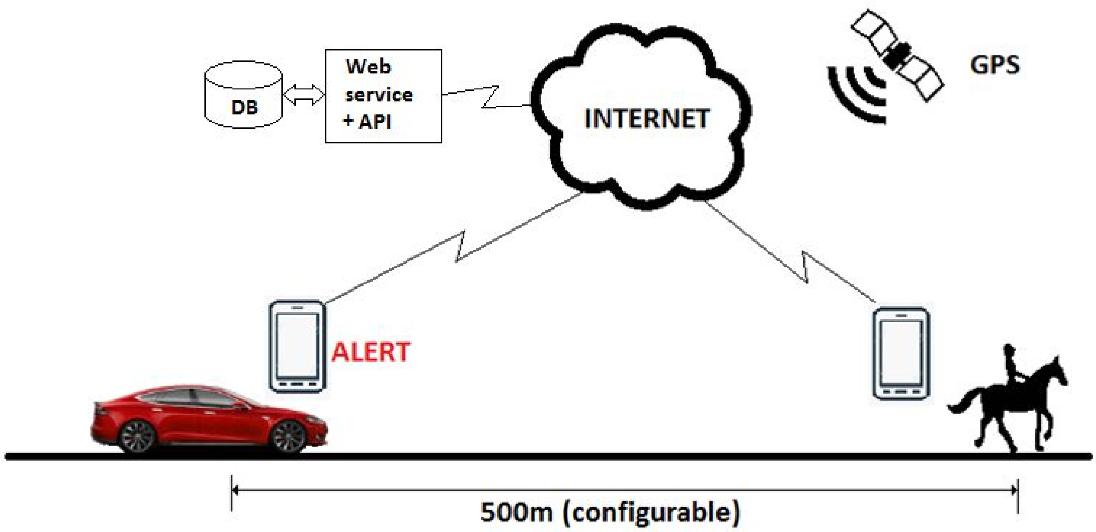


Figure 1, Sample overview of the RHPA system

1.2 Similar Solutions

Mitsubishi Electric and HERE Technologies (MH) developed a road hazard alert system, which was demonstrated on the 15th of May, 2019. The system built into the test vehicle could detect various hazards, such as a stopped vehicle, slippery conditions, a pothole, or debris in the road. This information would then be sent over a network to other vehicles. The system used a high-precision locator, developed by Mitsubishi, and completed two real-world tests, firstly in Japan during the winter, and then in California. Mitsubishi plans to offer the technology to other automakers for further testing. (S.Szymkowski, 2019)

The MH solution has similar features to the RHPA system developed in this project. However, the MH solution is intended to be built into new vehicles rather than offering an aftermarket solution and more importantly doesn't allow other road users to create their own hazard warnings. A further limitation of the Mitsubishi system was only to be used in vehicles, so would not work for horse riders and non-vehicle hazards e.g. flooding. These issues have been addressed in the RHPA system.

Google has recently added a similar feature to Google Maps. Users can report crashes, objects on the road, and many other types of hazard. All of these can be quickly and easily added by a user currently driving and then shown in Google maps. (Tseng, 2019)

This is perfectly suitable for stationary hazards, like crashes or roadworks, and users can easily be informed of an alternative route. However, moving hazards cannot be easily tracked, for example, horse riders and hedge cutters. The RHPA system developed in this project will also be able to track moving hazards so that road users in the affected area can be notified of the type of hazards.

1.3 Literature Review

1.3.1 Source 1:

"Simulating Cooperative Road Hazard Warning Application From the Perspective of V2V" (Liang, G., 2015)

This paper considers the method of transferring accident data from vehicle to vehicle (V2V) and Infrastructure to vehicle (I2V) so that vehicles approaching an accident area receive warning from other passing vehicles that an accident has occurred. The study uses simulations to analyse cooperative road hazard warning systems in order to design and deploy better collision avoidance systems.

1.3.2 Source 2:

"Road Hazard Point Acquisition, The Smartphone Way" (Jorritsma, J.S., 2012.)

The project outlined in this paper is similar to the RHPA project. The paper discussed developing a mobile application where users could report a hazard in the road and also be able to upload a picture of it.

However, this would rely upon having at least two people in the car, one to drive and another to use the app. Hazard alerts created by the RHPA app do not rely on the observations of other road users as discussed in this paper:

The requirement of governance on the quality of the data becomes obvious when looking at the reported hazards by different users at the same location. Validity of reported hazards needs to be assessed and maintained for the data to become and remain valuable for journey management. (Jorritsma, J.S., 2012.)

1.3.3 Source 3:

"Real-Time Multiple Sensors Microcontroller–Based Automobile Hazard Alert System" (Asafe, Y.N. and Mercy, O.F.)

In this paper a series of sensors are placed around the automobile, such as a smoke detector, to determine when an accident has occurred, and to report it automatically with an auto dialler and GPS chip, which will lead to a timely response from the emergency services.

1.3.4 Source 4:

"Autonomous pro-active real-time construction worker and equipment operator proximity safety alert system" (Teizer, J., Allread, B.S., Fullerton, C.E. and Hinze, J., 2010.)

In this paper a series of radio frequency (RF) receivers were fitted to construction vehicles, with transmitters being fitted to workers, equipment and materials. If a vehicle gets too close to something with an RF transmitter on it, the operator of the vehicle is alerted through auditory, visual, and vibrating alarms.

1.4 Scope of the project

The scope of this project is to create a functional prototype of the RHPA system that can be used as a proof of concept.

1.5 Aims and Objectives

Aim:

Create a mobile application that allows people to report and avoid road hazards.

Objectives:

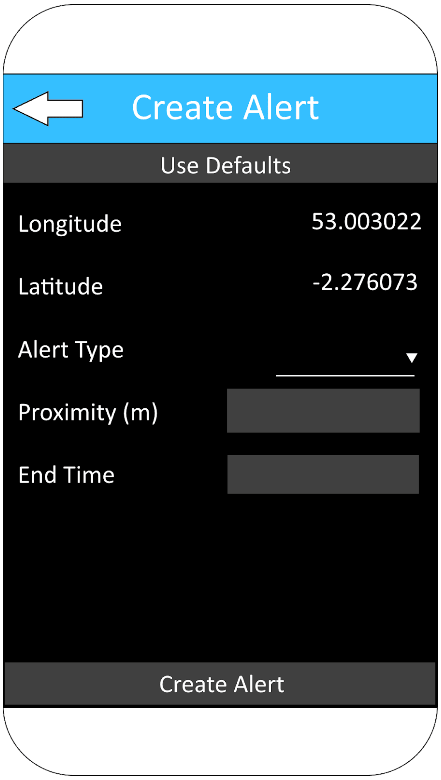
1. Creation of a mobile application
2. Creation of an easy to use interface
3. Creation of an external web server to handle hazard reporting
4. Allow mobile application to report hazards to the server
5. Evaluate the project with user feedback

2. Design

2.1 App Interface

The app has a main screen from which the three function screens and a settings screen can be accessed. Designed in common with existing apps, the user can easily switch between the settings screen and the following function screens; add a new alert, edit an existing alert and enter driving mode.

Figure 2, add alert screen design



The screens were designed in the following logical order:

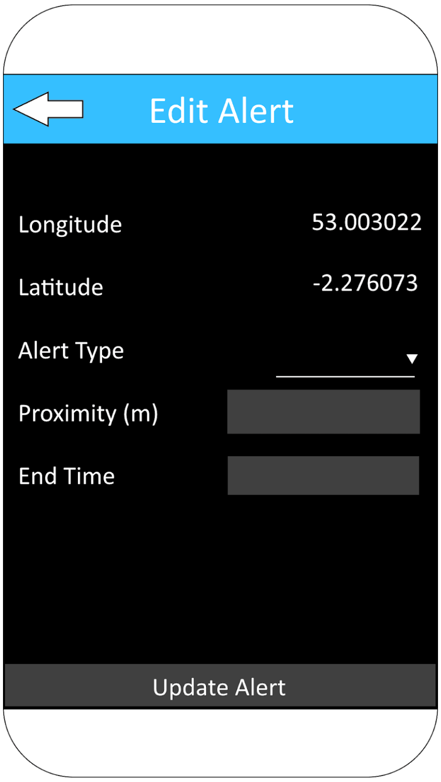


Figure 3, edit alert screen design

The Add Alert Screen (figure 2).

This screen allows the user to add a new alert onto the system. The user's latitude and longitude are populated using the coordinates obtained from the smartphone's Global Positioning System receiver (GPSr). The user is able to select a hazard type from a predetermined list, set the maximum proximity in meters from which their alert will be seen, and the time the alert will automatically end. All of this information is required before the alert can be created and added to the database where it remains until the end time is reached or if the user cancels it.

The Edit Alert Screen (figure 3). Allowing amendment to an existing alert gives the user the ability to change the hazard type, the proximity and the end time. The coordinates are updated automatically using the smartphone's GPSr once the alert has been updated and continue to update automatically until the end time is reached.

Driving Mode (figure 4). When entering driving mode there is an initial check over a ten-kilometre radius of the current position, again using the smartphone's GPSr, to collect data for valid alerts within range. The screen then shows the users current position and the nearest three alerts. As the driving mode user continues their journey, alerts are re-checked. When the maximum proximity of the creator's alert is reached an alarm is displayed and/or sounded.

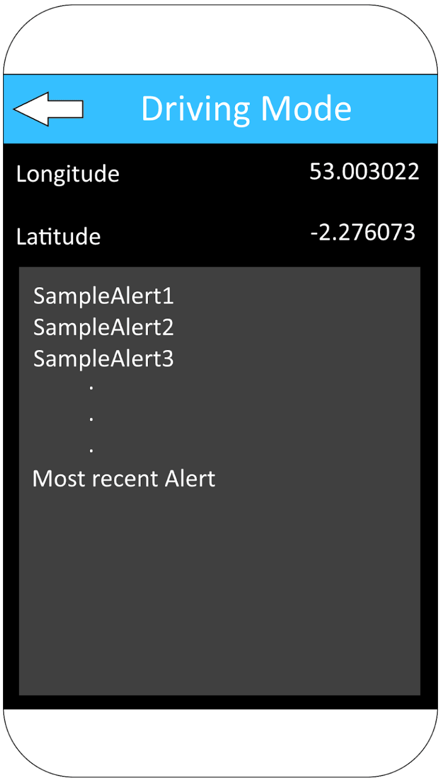
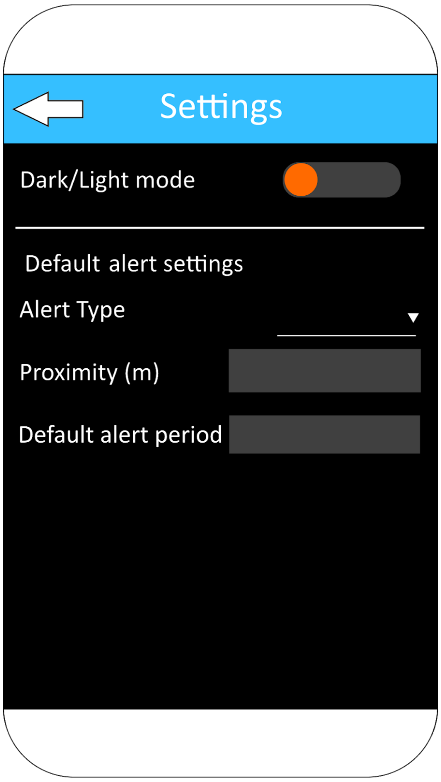


Figure 4, driving mode screen

Settings Screen (figure 5). This screen allows users to set their login details using an email address and password, and the defaults for type, proximity and time period for use when adding a new alert. The time period is added to the current time to calculate an end time when adding a new alert. The decision to have default alert settings was made to enable users to quickly and easily create new alerts. For example; if a police officer were to arrive on the scene of an accident, the officer would be able to create a new alert using the defaults purely by opening the app, selecting the create alert tag followed by the create alert button.

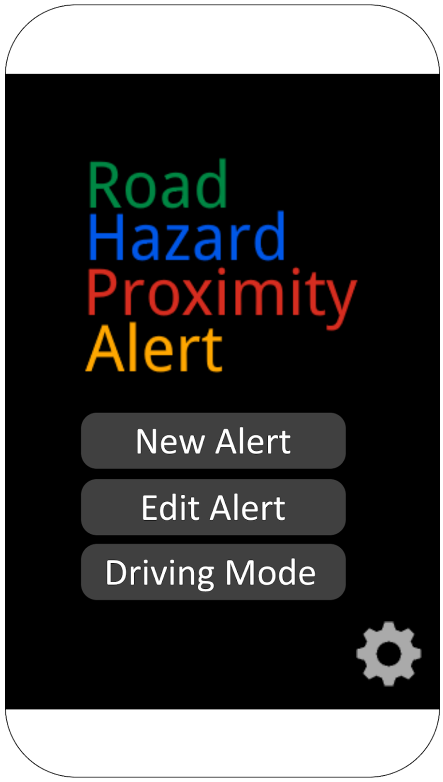
Figure 5, settings screen design



A decision was made after some testing to also include the option for Dark/Light mode giving two colour schemes for use during either day or night time.

Main Screen (figure 6). As this screen is acting as a menu, it only needed to have a simple design with four buttons to take the user to one of the corresponding screens.

Figure 6, main screen design



After making rough sketches of the screen designs, a graphics program was used to create the initial screens utilising Material Design to create a visually appealing app interface. Early versions of the user interface were shown to family, friends and my supervisor. The interface was found to be intuitive and operated with little explanation. Any further development to the look and feel was postponed due to the outbreak of the COVID-19 pandemic and the restrictions placed on social interaction.

2.2 Web Interface

The web interface, created purely for testing purposes rather than general users, was designed for functionality while still being as intuitive as possible.

The website consists of three screens.

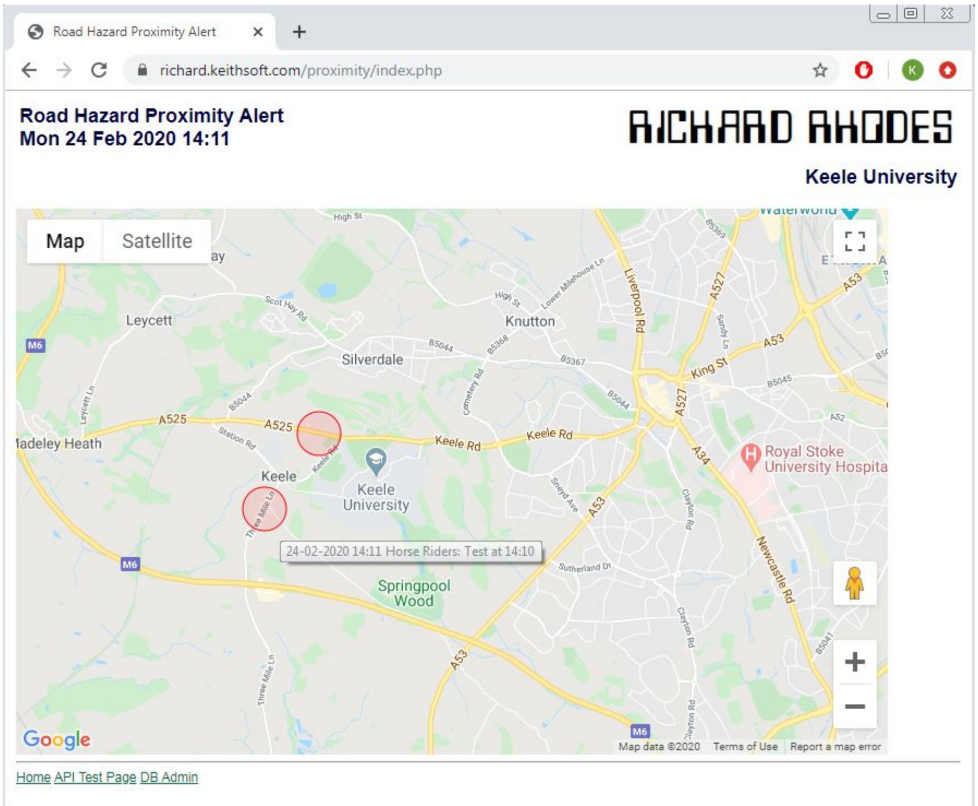


Figure 7, Main screen of the web interface, showing different alerts

The Interactive Map (figure 7). The map makes use of the Google map API, it shows the active alerts as red circles.

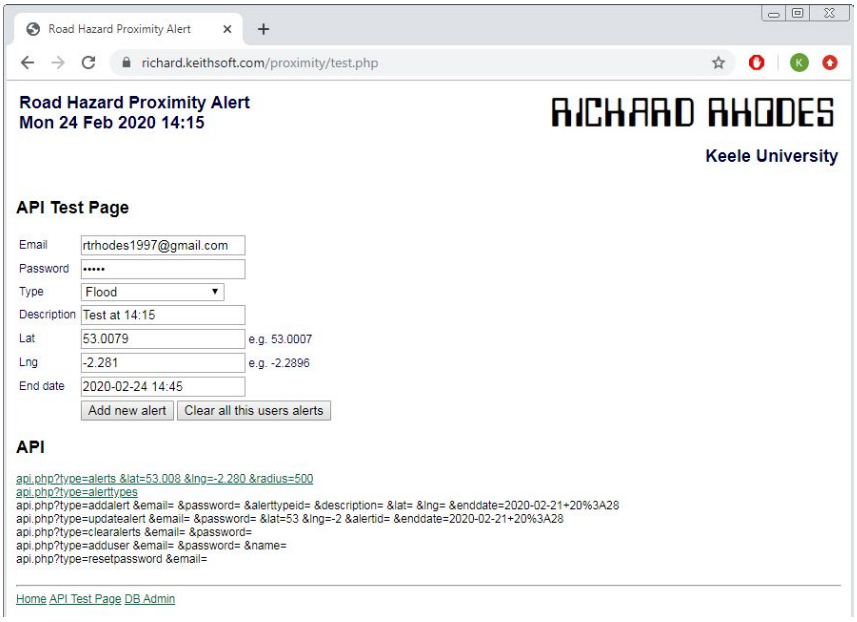


Figure 8, API test page

API Test Page (figure 8). The test page allows the operator to create alerts without going through the app, as well as API calls that can be made. When the API calls are clicked, a JSON string is returned, with the output of the API call inside.

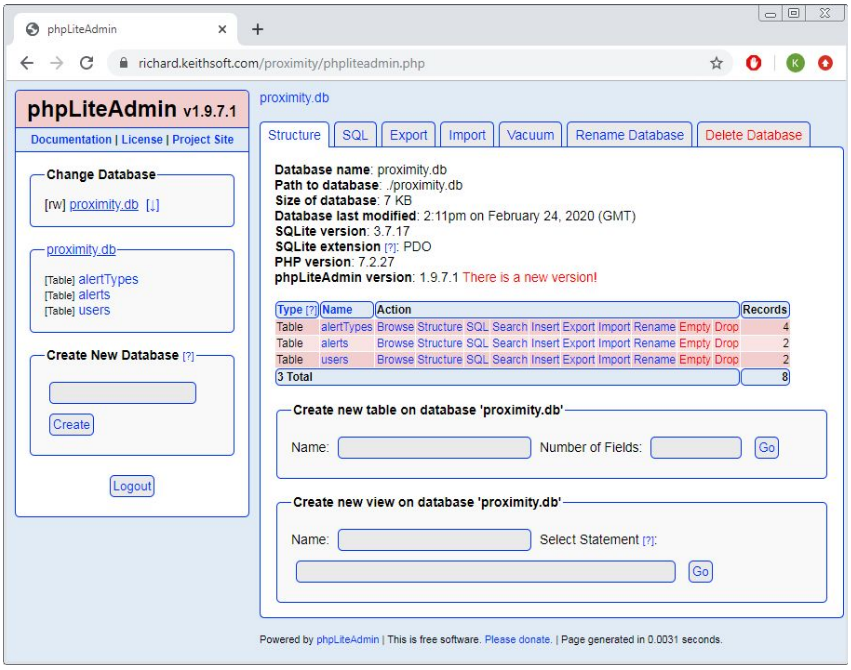


Figure 9, SQLite database access with PHPLiteAdmin

SQLite database (figure 9). The access page to the SQLite database was implemented by using phpLiteAdmin, both of which are free and easy to implement. This allowed direct inspection and manipulation of the data in the database, when required.

3. Methodology

3.1 App Methodology

Iterative prototyping was used to create the app. In total there were 12 iterations, for each one the prototype was created, tested, evaluated, and then improved upon to create the next prototype.

To keep track of the progress of each of the prototypes a version control system called Git was implemented. The developer already has two years of software development experience using this product. GitHub offers free online Git functionality and also provides the off-site backup of code as it develops. This ensures code can be recovered should development machines be lost or damaged.

Xamarin, which is a part of Microsoft Visual Studio, was used for the development of the smartphone app. Xamarin is well documented and supported online, the apps produced work on Android, iOS, and UWP with no need to re-write code. C# was used as the development language as the developer already has two years of software development experience and many of the online examples are given in C#.

A particularly useful feature of the Xamarin framework within the Visual Studio IDE is the ability to plug a smartphone into the development machine via a USB cable, enabling the developer to quickly and easily deploy the app to the device to test the functionality. When using this mode, it is also possible to set breakpoints in the code to help with debugging. When installing the app, the user is prompted to allow access to GPS data and internet access.

Phonegap or React Native were not used for the development of the app as the primary programming language for each of these technologies is Javascript. The developer did not have sufficient experience with using this language and given the time frame of this project, there was insufficient time to become adept.

The developer’s Google Pixel 3 was used to test the app, along with two Google Pixel 2XLs, a Samsung Galaxy J5 2016, a Samsung Galaxy J3 2016 and a Google Pixel 2 Emulator. These smartphones were chosen as they provide a large variety of resolution and screen sizes, and they were readily available.

No iOS devices were used when developing or testing the app due to the necessity of connecting the device to a Macbook and the prohibitive cost of submitting an app to the Apple app store. Having no access to either was not a problem as in the visual studio IDE on Windows, there is a feature that allows you to remotely connect to a Macbook, so that either an iPhone Emulator or a physical device can run the app.

Having previously built a mobile application for a different project as part of my part-time employment, it wasn’t felt necessary to test the app on an Apple device as a key feature of Xamarin is cross platform compatibility of any app created, so the app created would look and perform similarly across each platform.

Further iOS development on the RHPA project would involve submitting an app to the Apple app store and requires an Apple developer license, which is a yearly payment to Apple of $99 USD (£85.97 at time of writing), whereas the Google Play store only requires a one-time fee of $25 USD (£21.71 at time of writing).

3.2 Web Methodology

The server code was written using the waterfall methodology. There was no need to iteratively create a prototype since it required no user feedback, it just needed to work reliably. The server was written mainly in PHP using Macromedia Dreamweaver MX, because the developer already has experience using those technologies. Some Javascript was used to interface with the Google Maps API and create the interactive map.

3.3 App Development

Iteration 01 The Base App

During the first iteration of the app, the base of the app was created. This means that the main screen of the app was created as per the design in figure 6. The Git repository was initialised, then the solution was created using Visual Studio’s generator. At this stage it was important to check that all the Xamarin packages necessary were installed and up to date.

After the app was built and tested to ensure it worked, navigation buttons were added to the main screen. The ability to change the theme from dark mode to light mode based on a user setting was implemented.

A global control class was added so that information for screen creation could easily be passed to other screens during navigation, for example, the current theme.

Iteration 02: Adding Add Alert

In the next iteration of the app, the “Add Alert” screen was created as per the design in figure 2. During testing it was observed that adding more space around the “Add Alert” button meant that the user did not click it accidently before adding in all the details, similarly, with the “Use defaults” button.

Iteration 03: Reorganising the backend

In iteration 3 the backend of the app was reorganised. Prior to this, all of the functionality was in one large function, this was separated out into smaller functions that could then to be called again either elsewhere in the app, or re-used in other screens.

Iteration 04: Editing An Alert

The changes from the last iteration allowed re-use of code from the “Add Alert” screen into the “Edit Alert” screen. Both of them have a similar layout, apart from the “Use defaults” button.

Iteration 05: Driving Mode

The layout for driving mode was implemented. During this development it was realised that each alert could be stored as a set of coordinates and the user could choose the proximity in which alerts are reported. This change would need to be done in a future iteration as it was important to finish the UI and tidying up the existing code before implementing such a large change.

Iteration 06: Settings

During this iteration, the layout for the settings screen was created, and the dark/light mode switch functionality was added. At this point, all of the UI matches what is shown in figures 2 to 6.

Iteration 07: ServerConnection

The class to handle connecting to the server was added. The use of a class enabled more code reuse and simplified the code on individual screens.

Iteration 08: Location

The location functionality was fully developed. Originally the FusedLocationProvider was going to be used, unfortunately after many attempts and support forum searching it was clear that it could not be made to work reliably, probably due to developer inexperience.

The AndroidLocation API was used instead. This has the disadvantage that it only uses GPS, whereas the FusedLocationProvider can use GPS and WiFi or cellular data, automatically switching between them to get the best reading.

Iteration 09: Redoing ServerConnection

It was at this point the app was changed so that the user's proximity is used, instead of the alert's proximity. This was so that the app would only check for alerts every ten minutes but using a large radius e.g. 10km. If alerts are found within the 10km radius then increase the frequency of checks but only alert the user if the hazard is within the configurable 500m. This is not yet fully implemented and may be left for future work.

Iteration 10: Debugging

The backend of driving mode was altered to use a timer, which asynchronously fetched an updated list of alerts from the server that were around the user's location. The timer was set to ten seconds for testing.

Also, the final remnants of the user input proximity were removed. This included removing all of the proximity input checks for the app to send an alert to the server.

Iteration 11: More Debugging

The developer can set the location by commenting in and out some lines in the location handler, so that the app can be tested when no GPS connection can be made. This helped with the intermittent issues where the app would work when developing on campus, but not when developing at home.

Iteration 12: Yet More Debugging

Driving mode updates are now being reliably sent to the server. However, the server connection works intermittently for reasons that are currently unknown. The code from the serverConnection class was extracted to a console app where it always works reliably, however, for some reason when the same code is run in the app, it works intermittently.

A Note on Data Usage

When checking for alerts, the app communicates with the API and a very small amount of mobile data is used, a typical alert request is 93 bytes and a reply of no alerts is 2 bytes. However, this could still be a problem over extended periods due to the user's data allowance being consumed. To mitigate this the following options are noted for future development: -

- Detect when the user is moving and only check for alerts if the user is moving at speed, i.e. their GPS position has changed significantly since the last request indicating vehicle travel.
- Only check for alerts every 10 minutes but using a large radius e.g. 10km. If alerts are found within the 10km radius then increase the frequency of checks but only alert the user if the hazard is within 500m.

3.4 Web Development

Firstly, a specification was written for the API, which can be seen below. Then the API was developed in conjunction with a test harness.

The system made extensive use of a custom written RESTful API developed in PHP Version 7 and hosted on web space that the developer has access to. PHP was chosen because the developer has previous experience using PHP.

The API uses a single file: api.php to handle seven basic functions as described in the API Specification, requests can be made using GET or POST or a combination of the two.

The API is located at: https://richard.keithsoft.com/proximity/api.php

An example request:

https://richard.keithsoft.com/proximity/api.php?type=alerttypes

There is a test harness at: https://richard.keithsoft.com/proximity which uses the Google maps API to visualise any current alerts and to add example alerts using an API test page.

3.4.1 API Specification

NOTE: To ensure security, all requests should be made using HTTPS so that possibly sensitive information is not sent in clear text across the internet.

Type	alerts			
Description	Returns a JSON object array of currently active alerts, optional parameters allow for specific location and radius.			
Parameter	Type	Required	Example	Notes
lat	number	No	53.008	Latitude
lng	number	No	-2.28	Longitude
radius	number	No	500	Only return alerts from the radius in metres of the specified lat/lng
Return Value	A JSON array of active alerts, if lat and lng are specified, the return will include calculated distance in metres from the alert.			
Example	api.php?type=alerts&lat=53.008&lng=-2.280&radius=500			
Sample return	[{"AlertID":231,"StartDate":"24-02-2020 12:34","AlertType":"Horse Riders","Description":"Test at 12:28","Lat":53.0079,"Lng":-2.281,"Distance":67.82}]			

Type	alerttypes			
Description	Returns a list of available alert types, this function has no parameters.			
Parameter	Type	Required	Example	Notes
Return Value	Returns a JSON object array of current alert types from the database, the AlertTypeID is used for registering new alerts in the API.			
Example	api.php?type= alerttypes			
Sample return	[{"AlertTypeID":4,"Description":"Flood"},{"AlertTypeID":3,"Description":"Hedge Cutter"},{"AlertTypeID":2,"Description":"Horse Riders"},{"AlertTypeID":1,"Description":"Road Traffic Accident"}]			

Type	addalert			
Description	Adds an alert to the database			
Parameter	Type	Required	Example	Notes
email	text	Yes	user@sample.com	Registered users email address
password	text	Yes	P@55word	Users password
alerttypeid	number	Yes	2	Alert type see: alerttypes
description	text	Yes	Riding in Keele	Text description of the alert
lat	number	Yes	53.0079	Latitude
lng	number	Yes	-2.281	Longitude
enddate	date	Yes	2020-02-24 13:27	End date/time for the alert to finish, must be in the format YYYY-MM-DD HH:MM

Return Value	Returns the AlertID of the added alert or an error message.
Example	api.php?type=addalert&email=user@sample.com&password=P@55word &alerttypeid=2&description=Riding in Keele&lat=53.0079&lng=-2.281&enddate=2020-02-24 13:27
Sample return	228

Type	updatealert			
Description	Updates an existing alert with new location and end date/time.			
Parameter	Type	Required	Example	Notes
email	text	Yes	user@sample.com	Registered users email address
password	text	Yes	P@55word	Users password
alertid	number	Yes	228	AlertID returned from addalert
lat	number	Yes	53.0079	Latitude
lng	number	Yes	-2.281	Longitude
enddate	date	Yes	2020-02-24 13:27	End date/time for the alert to finish, must be in the format YYYY-MM-DD HH:MM
Return Value	Returns OK if the update was successful or an error message if not.			
Example	api.php?type=updatealert&email=user@sample.com&password=P@55word &alertid=228&lat=53.0179&lng=-2.291&enddate=2020-02-24 13:37			
Sample return	OK			

Type	clearalerts			
Description	Clears all of a user's active alerts from the database. It is not normally necessary to do this as alerts will expire automatically.			
Parameter	Type	Required	Example	Notes
email	text	Yes	user@sample.com	Registered users email address
password	text	Yes	P@55word	Users password
Return Value	Returns OK if the command was successful or an error message if not.			
Example	api.php?type=clearalerts&email=user@sample.com&password=P@55word			
Sample return	OK			

Type	adduser			
Description	Adds a user to the system so that they can register alerts.			
Parameter	Type	Required	Example	Notes
name	text	Yes	John Smith	

email	text	Yes	user@sample.com	Users email address
password	text	Yes	P@55word	Users password
Return Value	Returns OK if adding the user was successful or an error message if not.			
Example	api.php?type=adduser&name=John Smith&email=user@sample.com&password=P@55word			
Sample return	OK			

Type	resetpassword			
Description	Resets a users password and emails the new password to them.			
Parameter	Type	Required	Example	Notes
email	text	Yes	user@sample.com	Users email address
Return Value	Returns OK if the reset was successful or an error message if not.			
Example	api.php?type=resetpassword&email=user@sample.com			
Sample return	OK			

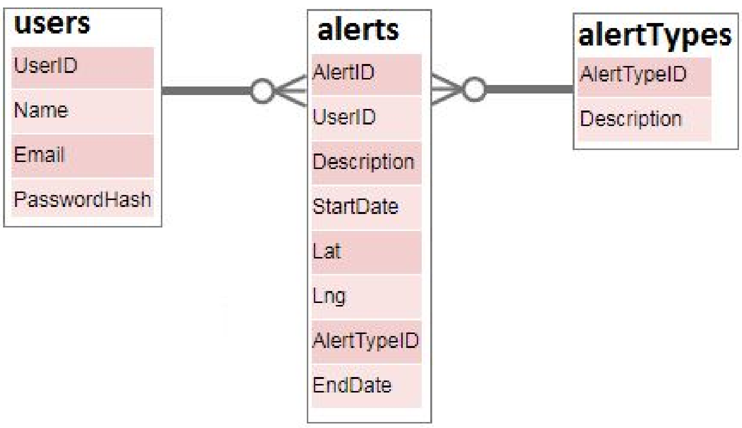
The 'alerts' API call can be used to find and return the proximity between the user's current position and the position of active alerts. The function to do this uses a free PHP function provided by https://www.geodatasource.com/developers/php, licenced using the LGPLv3. However, it should be noted that elevation does not feature in the calculation so there will be some inaccuracies, for example, a user may show as being a few metres from an alert but actually they are on a bridge several metres above an alert.

3.4.2 SQLite Database

ERD is shown in figure 10, and the SQLite code needed to create these tables is shown in figure 11

Table	**Description**
users	Contains all registered users of the system, only registered users are allowed to post alerts. For security, the user passwords are stored hashed using PHP built-in functions.
alerts	Contains a list of currently active alerts. Expired alerts are automatically removed from this table by the API.
alertTypes	Contains a list of available alert types so that user interfaces can offer a dropdown list of available types to users.

Figure 10, ERD for SQLite database



The PHP code used by the API runs very quickly and should be able to support several hundred users. If the app was commercially developed then the number of users could grow very quickly and a more robust and scalable host should be considered, a popular choice at the moment is AWS (Amazon Web Services)

```
CREATE TABLE 'users' ('UserID' INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
'Name' TEXT, 'Email' TEXT, 'PasswordHash' TEXT)

CREATE TABLE 'alerts' ('AlertID' INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
'UserID' INTEGER, 'Description' TEXT, 'StartDate' DATETIME, 'Lat' NUMERIC, 'Lng'
NUMERIC,'AlertTypeID' INTEGER, 'EndDate' DATETIME)

CREATE TABLE 'alertTypes' ('AlertTypeID' INTEGER PRIMARY KEY AUTOINCREMENT NOT
NULL, 'Description' INTEGER)
```

Figure 11, SQLite statements to create tables necessary for the project to work

The European Data Protection Regulation is applicable as of May 25th 2018 and covers the use of any personally identifiable information. Within the database, name and email address would be considered personally identifiable information, also where an alert has been registered the likely GPS position of the user would be stored. Permission would need to be obtained from the user to store the above information. The GPS position of unregistered users that are simply monitoring for alerts in their area would not be considered personally identifiable information as it is not connected to a named individual.

4. Evaluation

4.1 Testing

White box testing was used during development of the app to ensure all functions were tested fully. A feature would be designed, created and then tested on a smartphone which would give one of three results:

1. The program was executed and the output was as expected
2. The program was executed and the output was not as expected
3. The program failed to execute

For both results 2. and 3., the developer would cease testing and correct the programming until the feature produced the expected results. This pattern would then be repeated until all features allocated to the current iteration produced the expected results. If applicable, the data used for testing was altered and the testing repeated for completeness.

After the testing of each iteration was completed successfully, it was then black-box tested. The app was loaded onto the developer’s smartphone and passed to a couple of pilot users who were told what the system was designed to do, but who didn’t know the intricacies of how the app worked. This was to find any occurrences where a user was able to enter incorrect data that the developer had not accounted for or if there were any circumstances where the user was unable to understand the results.

4.2 Interface Questionnaire

A document containing a list of questions and screen shots of each of the initial screen designs within the app was compiled into the format of a questionnaire. The intention was to present the questionnaire to each of the participants and gauge the answers, which may have led to further questions or other ideas that could have been incorporated into the subsequent design phase.

Main screen (figure 6)

Question 1: *Do you think that this screen displays adequate information to allow you to start using the app?*
Experience of other apps indicates that unless they are easy to set up and use, they will not continue to be used or recommended to other potential users.

Question 2: *Do you think that the buttons appear in an appropriate sequence, if not why not?*
Buttons and questions are selected or answered more easily when they flow in a logical sequence from top to bottom or left to right.

Question 3: *What do you like about the design?*
Features that were liked could be incorporated into future work.

Question 4: *Do you think that there is anything that could improve the look and ease of use?*

This gives respondents the opportunity to make suggestions that may not have been considered.

Settings (figure 5)

Question 1: *Do you think the ability to set up default information is a useful feature?*

If the majority of answers were negative then the feature could be removed to streamline the app.

Question 2: *Are the entry fields on the screen in a logical sequence, if not why not?*

Buttons and questions are selected or answered more easily when they flow in a logical sequence from top to bottom or left to right.

Question 3: *The app uses metric measurements, do you think the option to switch to imperial measurements would be helpful?*

Some countries still measure distance in yards and miles and these are displayed on road signs.

Question 4: *Do you think that there is anything that could improve the look and ease of use?*

Buttons and questions are selected or answered more easily when they flow in a logical sequence from top to bottom or left to right.

Question 5: *Can you think of any other information that would be useful on this screen?*

Users may have ideas that have not previously been considered.

Add Alert (figure 2)

Question 1: *Do you understand the pre-populated information shown on the screen?*

Certain information may not be as intuitive to new users as to those who have been using the app for some time. For example, the Latitude and Longitude, there may be a need to add additional help text.

Question 2: *Are the entry fields on the screen in a logical sequence, if not why not?*

Buttons and questions are selected or answered more easily when they flow in a logical sequence from top to bottom or left to right.

Question 3: *Would you like the ability to add your own alert types?*

If the majority of answers were positive then the feature could be added in future work.

This would require validation to ensure new alert types met certain criteria.

Question 4: *Do you think that there is anything that could improve the look and ease of use?*

Buttons and questions are selected or answered more easily when they flow in a logical sequence from top to bottom or left to right.

Question 5: *Can you think of any other information that would be useful on this screen?*

Users may have ideas that have not previously been considered.

Edit Alert (figure 3)

Question 1: *Do you understand the pre-populated information shown on the screen?*

Certain information may not be as intuitive to new users as to those who have been using the app for some time. For example, the Latitude and Longitude, there may be a need to add additional help text.

Question 2: *Are the entry fields on the screen in a logical sequence, if not why not?*

Buttons and questions are selected or answered more easily when they flow in a logical sequence from top to bottom or left to right.

Question 3: *Do you think that there is anything that could improve the look and ease of use?*

Buttons and questions are selected or answered more easily when they flow in a logical sequence from top to bottom or left to right.

Question 4: *Can you think of any other information that would be useful on this screen?*

Users may have ideas that have not previously been considered.

Driving Mode (figure 4)

Question 1: *Do you understand the information as it is shown on the screen?*

The nearest three alerts are shown as type and distance. As this screen will be the most frequently used it is important that the information is shown in the clearest format.

Question 2: *Do you think an audible warning is sufficient?*

A visual warning has not been added as it may distract the driver, but users may have ideas that have not previously been considered.

Question 3: *Can you think of any other information that would be useful on this screen?*

Users may have ideas that have not previously been considered.

4.3 Evaluation questionnaire

Prior to this questionnaire, the participants would receive a demonstration of each screen in the app by the developer. The subsequent questionnaire consisted of six sections, one for each screen and one for general questions. Each section also included the question “any other comments” to give respondents the opportunity to make other suggestions that may not have been relevant to previous questions.

The results of this questionnaire would have been used to make any last minute changes to the app, should any be required. Other than that, the results could be used to drive any future work on the RHPA project.

Main screen

Question 1: *Do you think the layout of the screen made it easy to use?*

Experience of other apps indicates that unless they are easy to set up and use, they will not continue to be used or recommended to additional users.

Question 2: *Do you understand what all of the options were for, including the gear icon?*

There may be a requirement to rename or simplify the buttons or screens.

Question 3: *Any other comments?*

This is a deliberately open question to give respondents the opportunity to make other suggestions that may not have been relevant to previous questions.

Settings

Question 1: *Do you understand that the defaults can be changed at any time?*

This was asked to ensure the user did not assume the app had to be deleted and reinstalled should they wish to change the defaults.

Question 2: *Do you think that the option for dark or light mode is useful?*

If the majority of answers were negative then the feature could be removed to streamline the app.

Question 3: *Can you think of any additional hazard types that could be added?*

If similar additional hazards were suggested then these could be collated and added to the database.

Allowing users to free type their own hazard types was excluded due to the validation required before adding to the database. Any free typed hazard not in the database would not be visible to other users.

Question 4: *Any other comments?*

This is a deliberately open question to give respondents the opportunity to make other suggestions that may not have been relevant to previous questions.

Add Alert

Question 1: *Do you think it was easy to set up a new alert?*

Experience of other apps indicates that unless they are easy to use, they will not continue to be used or recommended to additional users.

Question 2: *Do you think that having the defaults is useful?*

If the majority of answers were negative then the feature could be removed.

Question 3: *Do you think that the information on the screen is clear and relevant?*

Certain information may not be intuitive. For example, the Latitude and Longitude labels, there may be a need to add additional help text or remove some details to simplify the screen.

Question 4: *Any other comments?*

This is a deliberately open question to give respondents the opportunity to make other suggestions that may not have been relevant to previous questions.

Edit Alert

Question 1: *Do you think it was easy to change the details of an existing alert?*

Experience of other apps indicates that unless they are easy to use, they will not continue to be used or recommended to additional users.

Question 2: *Do you think that the information on the screen was clear and relevant?*

Certain information may not be intuitive. For example, the Latitude and Longitude labels, there may be a need to add additional help text or remove some details to simplify the screen.

Question 3: *Any other comments?*

This is a deliberately open question to give respondents the opportunity to make other suggestions that may not have been relevant to previous questions.

Driving Mode

Question 1: *Do you understand the information shown on the screen?*

As this screen shows the alerts within the alert radius it is important that it is clear and easy to understand. Any changes that can be made to make it easier could be incorporated into future development.

Question 2: *Do you understand why the alerts changed during the demonstration?*

The information shown on the screen regarding hazards should be as clear as possible, any changes that can be made to make it clearer could be incorporated into future development.

Question 3: *Do you think an audible warning was sufficient?*

The developer did not want to incorporate any visual warning as this may distract the driver. Users may have ideas that have not previously been considered.

Question 4: *Any other comments?*

This is a deliberately open question to give respondents the opportunity to make other suggestions that may not have been relevant to previous questions.

General Questions

Question 1: *Do you think this app was easy to use?*

Experience of other apps indicates that unless they are easy to use, they will not continue to be used or recommended to additional users.

Question 2: *Do you think you could use this app without further instruction?*

Additional help text may be required.

Question 3: *Would you use this app?*

User feedback would be measured against future development requirements.

Question 4: *Would you feel safer in a vehicle if the driver was using this app?*

The aim of the RHPA project is to aid road safety, does the user feel that this app could help towards this.

Question 5: *Do you think that the app could help prevent accidents and ease traffic flow?*

The aim of the RHPA project is to aid road safety, does the user feel that this app could help towards this.

Question 6: *Do you think the concept of this app could be integrated with other services such as Google Maps?*

The aim of the RHPA project is to aid road safety, does the user feel that this app could help towards this.

Question 7: *Any other comments?*

This is a deliberately open question to give respondents the opportunity to make other suggestions that may not have been relevant to previous questions.

4.4 Results

Unfortunately, due to the outbreak of the COVID-19 pandemic and the restrictions placed on social interaction, neither questionnaire could be completed.

4.5 Lessons learned from development:

Use an Intel processor.

When I initially started development on the RHPA project I used my desktop PC, and this was going well until I got to the point of setting up the emulator. I was relying on the emulator to run the app, as this would be easier and quicker than using my phone.

When I came to run the app using the emulator it took a significant amount of time to start. After some investigation this was found to be due to the fact that Android runs on an ARM processor, instead of an x86 processor. To solve this problem, Intel developed the Intel Virtualisation Technology (VT), which speeds up the execution of x86. This can be installed with Intel's Hardware Accelerated Execution Manager (HAXM), which is available for Windows and Linux.

However, this is only compatible with an intel processor. My desktop, on the other hand, runs on an AMD Ryzen 1700X. AMD has their own virtualisation technology, AMD-V. However, this only supports Linux, which is an issue covered in the next point.

Use Windows

There is no officially supported way to create apps using Xamarin on Linux, an unofficial way was recently discussed on the 27th March 2020. In this article, Ubuntu 16.04 was used to manually install Xamarin.Android and configure Rider, a cross-platform IDE (JetBrains, 2020).

As this method was only discussed on the JetBrains website on the 27th of March, it was too late to be considered as a development environment for use in the RHPA project, as the app could not be developed, tested, and released during the 12-day period between then and the submission date.

In addition, the aforementioned Rider IDE is not free software, and costs $139 (£111.60 at time of writing) per year for a personal license, which is as prohibitive as the Apple cost discussed earlier.

Use version control

When the RHPA project was first started, the project was tracked in Git as I have lots of previous experience using this. It was only after the project was on a stable footing that it was uploaded to GitHub. Uploading the project to GitHub meant that changes could be tracked and the entire project could be moved to a new computer if necessary.

Xamarin/Visual Studio Crashes

Visual Studio and the Xamarin framework were chosen because I also have previous experience working with them, and as such I am aware how often that Xamarin causes Visual Studio to crash. Remembering to save the work often can mitigate this to some extent.

It's OK to start again

Very often when Visual Studio crashes, it would take a significant amount of time to get back to a point where I could continue work. This led me to learn that simply starting again, and re-importing the code that was needed was often a simpler solution, and would also take far less time.

5. Conclusion

5.1 Recapping the Aims and Objectives

Objectives:

1. Creation of a mobile application

The mobile application was created. The application can get its location from GPS and make API calls using that data. This means that the application can fully communicate with the website, and utilise the database to fulfill its functions.

2. Creation of an easy to use interface

The mobile application that was created was designed to be easy to use. This was going to be evaluated further by questionnaire, however due to the ongoing health crisis, this was not possible and only the pilot users made a contribution.

3. Creation of an external web server to handle Hazard reporting

The website that hosts the database was created using web space I had access to. The website is powered by an SQLite database and uses a custom-written API written in PHP.

4. Allow mobile application to report hazards to the server

The mobile application uses the website's API to report hazards to the server along with getting a list of current alerts, and getting a list of the type of alerts available.

5. Evaluate the project with user feedback

User feedback would have been used to evaluate the project. Errors found and improvement suggestions from this would have been fixed and any similar suggestions collated and implemented. However, as previously stated, due to the ongoing health crisis this could not be completed.

Aim: Create a mobile application that allows people to report and avoid road hazards

Points one to four in the above objective were completed and in turn created the working prototype of a mobile app. Other than acting on any recommendations by a wider audience I can conclude the project was successful.

5.2 Further Work

5.2.1 Questionnaires

I would like to demonstrate the application and have the questionnaires completed by a much wider group of volunteers.

User feedback is a critical part of any new development and as this is missing, should any future work be undertaken, this should be the first thing undertaken.

5.2.2 Revamp the visuals

At the moment, the visual design of the app is mediocre at best. In any future work, should any be undertaken, I would recreate all of the visuals for the app, closely following the material design guidelines by Google.

5.2.3 Integration with other services, such as Google Maps

During the development of the app. There was an idea to only show alerts if the user was moving towards them. This means that the user would only be shown alerts if the distance was repeatedly getting smaller. This could be integrated into the navigation functionality of Google Maps, where alerts along the route would be user-reported and only shown to users who are navigating that way. This could also automatically cause an alternate route to be selected, which would ease traffic flow around the hazard.

Appendices

References

Liang, G., 2015, July. Simulating Cooperative Road Hazard Warning Application From the Perspective of V2V. In *5th International Conference on Information Engineering for Mechanics and Materials*. Atlantis Press.

Jorritsma, J.S., 2012. Road Hazard Point Acquisition The Smartphone Way.

Asafe, Y.N. and Mercy, O.F., Real-Time Multiple Sensors Microcontroller–Based Automobile Hazard Alert System.

Teizer, J., Allread, B.S., Fullerton, C.E. and Hinze, J., 2010. Autonomous pro-active real-time construction worker and equipment operator proximity safety alert system. *Automation in construction*, *19*(5), pp.630-640.

Szymkowski, S., 2019. *New Tech Could Alert Drivers To Potholes*. [online] The Car Connection. Available at: <https://www.thecarconnection.com/news/1123080_new-tech-could-alert-drivers-to-potholes> [Accessed 29 March 2020].

Tseng, S., 2019. *New Ways To Report Driving Incidents On Google Maps*. [online] Google Blog. Available at: <https://www.blog.google/products/maps/new-ways-report-driving-incidents-google-maps/> [Accessed 29 March 2020].

JetBrains, 2020. *How To Develop Xamarin.Android Applications On Linux With Rider*. [online] JetBrains Rider Support. Available at: <https://rider-support.jetbrains.com/hc/en-us/articles/360000557259-How-to-develop-Xamarin-Android-applications-on-Linux-with-Rider> [Accessed 29 March 2020].

Project plan

UG Project Plan
CSC-30014

Project Overview and Description

Student Name: Richard Rhodes
Student Username: W8X41
Student Number: 1701 4242
Degree Title: Computer Science
Supervisor Name: Theocharis Kyriacou
Project Title: Developing a Road Hazard Proximity Alert System

Please provide a brief Project Description:

A mobile application that runs on the user's smartphone. The application allows registered users, such as the police, to activate a proximity alarm for other users of the app, in order to alert them of potential road hazards.

The alarm's location is updated regularly, until it is automatically deactivated. The alarm's creator has the option to increase the time that the alarm is active for. Once a user is within a configurable distance (default 500m), they are audibly alerted to the presence of a hazard.

What are the aims and objectives of the Project?

Aim:
Create a mobile application that allows people to report and avoid road hazards

Objectives:
- Creation of a mobile application
- Creation of an easy to use interface
- Creation of an external web server to handle Hazard reporting
- Allow mobile application to report hazards to server
- Evaluate the project with user feedback

Please provide a brief overview of the key literature related to the Project:

Many recent studies have contained elements of the system that I am planning to create. However, there is no paper that I have read where the system I plan to create has been fully developed.

The papers that I have read are listed in the References section, I also intend to explore some of the books and articles related to Xamarin development and User interface design for mobile applications.

Project Process and Method

Please provide a brief overview of the Methodology to be used in the Project (inc. an overview of best practice within the Methodology):

I will be using Iterative prototyping in order to develop the application. This involves creating a prototype of the application, evaluating the prototype, and then improving the prototype.

I will follow the following steps:

1. Develop a prototype
2. Test the prototype
3. Evaluate the prototype
4. Repeat steps 1-3 until the prototype has met all objectives
5. Tidy up the code and release

Which Data Collection Methods will be employed (e.g card sorts, questionnaires, simulations, …)?

Simulations involving Android device emulators
Questionnaires for user feedback

Briefly describe how you will ensure your project is in line with BCS Project Guidelines (BSc Computer Science Single Honours Students only)?

I will include the following in the report:

- A clear definition of the aim and objectives
- A well researched literature review
- A description of the methodology used to create the application, including the use of the tools needed to create the project
- Justifications for all decisions that were made during the process of developing the application
- An evaluation for the entire project, including potential improvements to the project that would be feasible to implement

Time and Resource Planning

Will Standard Departmental Hardware be used? NO

If NO please outline the Hardware/Materials to be used:

Android Devices / Emulators
External Web Hosting (Keele web server)

Will Software which is already available in department be used? NO

If NO please outline the Software to be used including how any necessary licences will be obtained:

Xamarin.Forms package for Visual Studio, MIT License

Will the project require any Programming? YES

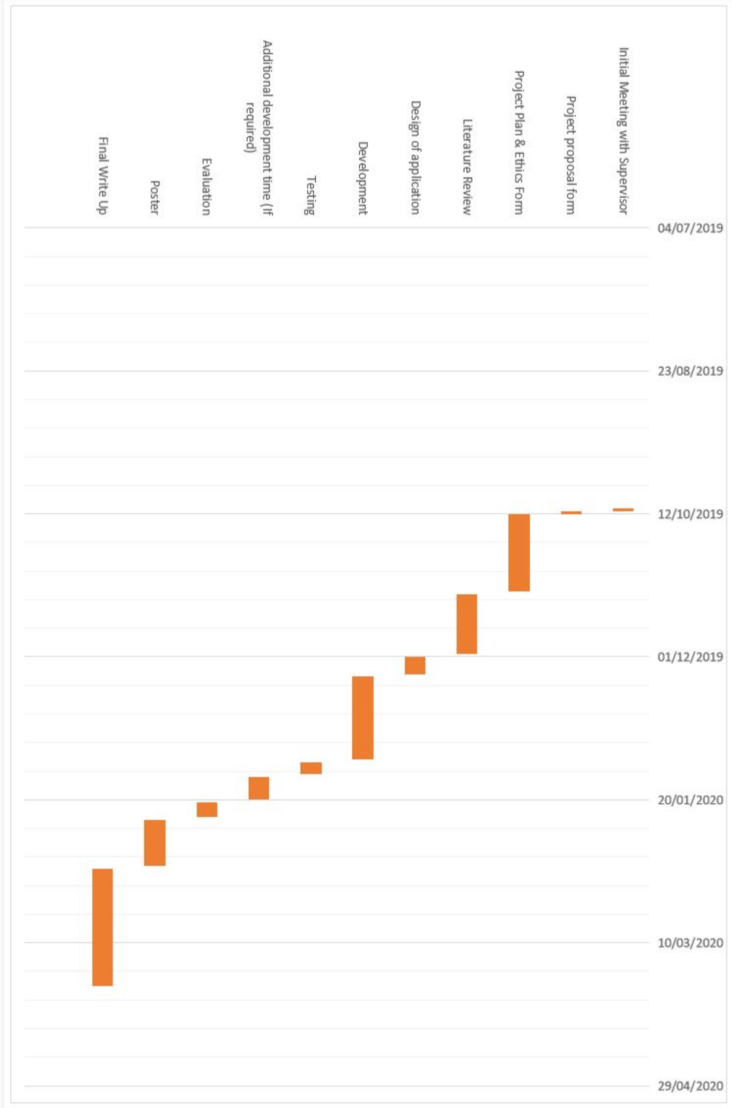
If YES please list the (potential) Programming Languages to be used (including any IDEs and Libraries you may make use of):

Visual Studio - IDE
Visual Studio Code - IDE
Vim - IDE
Xamarin.Forms - Library
C# - Language
PHP - Language
MySQL - Language
JavaScript - Language

Table of Risks (*if non Standard Hardware and/or Software to be used please include backup options/ contingency plans here*):

Description of Risk	**Likelihood**	**Best Practice to avoid**	**Action in case it happens anyway**
Running out of time	Low	Manage time	Present as PoC
Loss of data	Low	GitHub Commits, laptop backup every week, Use google docs	Revert from last GitHub commit
Xamarin Updating	Low	Don't update	Revert to last working update

Gantt Chart/ Pert Chart (must include milestones and deliverables):



References and Administration

Please include a list of References used in this Plan:

Simulating Cooperative Road Hazard Warning Application From the Perspective of V2V
https://www.atlantis-press.com/proceedings/icimm-15/25653

Road Hazard Point Acquisition, The Smartphone Way
https://dspace.library.uu.nl/handle/1874/238575

Real-Time Multiple Sensors Microcontroller–Based Automobile Hazard Alert System
http://www.indusedu.org/pdfs/IJRESS/IJRESS_448_12990.pdf

Autonomous pro-active real-time construction worker and equipment operator proximity safety alert system
https://www.sciencedirect.com/science/article/pii/S0926580510000361

Co-driver Alert Project
https://www.infona.pl/resource/bwmeta1.element.ieee-art-000004562180

Submission Date: 7/11/19

PLEASE NOTE THAT SHOULD YOUR PROJECT UNDERGO ANY MAJOR CHANGES FOLLOWING THE SUBMISSION OF THIS PLAN YOU ARE EXPECTED TO SUBMIT AN UPDATED PLAN WHICH ACCURATELY REFLECTS YOUR PROJECT.

Ethics Checklist

Computer Science CSC-30014 Project GDPR and Ethics Checklist 2019-20

STUDENT NAME: Richard Rhodes

STUDENT NUMBER & E-MAIL: 1701 4242 - W8X41@students.keele.ac.uk

PROJECT TITLE: Developing a Road Hazard Proximity Alert System

SUPERVISOR NAME: Theocharis Kyriacou

Complete all the questions below and electronically sign and date at the end. Ask your supervisor to check the responses and to also electronically sign and date below. You should submit this completed checklist to the KLE drop-box provided. Please retain your own copy of the completed checklist as all end of project reports/dissertations must provide a copy of this completed checklist in the Appendices.

If a Computer Science Final-Year Project Ethical Review Application Form has also had to be completed a copy of the final ethical approval certificate must also be included with your Project Plan document in your final report/dissertation Appendices.

GDPR Check

Does your project involve the use or collection of "personal data" for which permission will not have been explicitly granted?	**No**

Ethics Check

Will your project involve the use of human participants or capturing human data?	**Yes**

Significant Ethical Concern Checklist

Could the project expose the participants or the project student to images and/or information that they might find distressing (e.g. pictures or descriptions of injuries, symptomatic health conditions, or atrocities, or pictures or descriptions of tumours or cancerous cells, or creatures in distress)?	**No**
Does the project involve deception of the participants?	**No**
Could the project uncover information about identifiable individuals that could cause embarrassment or distress to one or more of those individuals (e.g. evidence of illegal or unethical behaviour, such as fraud or illegal drug use or a personal revelation)?	**No**
Could the project cause pain, discomfort or risk to the participants and/or the project student?	**No**

Will the project involve participants who are vulnerable in any way? (e.g. participants who are under 18, or who are mentally or physically impaired, or participants who may feel under pressure to participate.)	**No**
Does the project involve recall or discussion of personal or sensitive memories?	**No**
Does the project involve a significant risk of participants later regretting taking part?	**No**
Does the project involve procedures which are likely to provoke interpersonal or inter-group conflict?	**No**

If the answer to any of the Significant Ethical Concern Checklist questions is "Yes" (or you think "Maybe") you must discuss your project and aims with your supervisor and possibly with the CSC-30014 Ethics Advisor (and possibly with the CSC-30014 Module Co-ordinator) to assess whether an appropriate level of ethical scrutiny might be required via the completion of the *Faculty of Natural Sciences (non-Psychology) Research Ethics Application Form*.

GDPR Check Guidance:

Personal data includes any and all of: names, addresses, emails, phone numbers, bank details, employment details, IP addresses, date of birth, medical or health data, images, video or audio recordings.
If you answered "Yes" you must not proceed with your project.
It is illegal under European GDPR legislation to make use of personal data without explicit permission. Discuss your project with your supervisor and revise your plans to ensure you do not risk illegal use of personal data.
Note. Even if personal data is publicly available on the Internet, it must not be used without permission. (Also note that you cannot contact individuals to request permission to use their on-line data without prior ethical approval to do so).

It is strongly recommended that you either:

1. use non-personal data for your project, or
2. use existing, well-established, publicly available databases or data repositories, for example: https://archive.ics.uci.edu/ml/index.php, https://physionet.org/ or https://www.kaggle.com/ etc. (A list of acceptable data repositories is maintained on the CSC-30014 KLE pages.) You might also see, for example, https://blog.scrapinghub.com/web-scraping-gdpr-compliance-guide) for further information.

Continued on the following page ...

I confirm that the responses are correct and that the project, as proposed, is GDPR compliant and that ethical approval will be sought if required and that any work requiring ethical approval will not take place unless ethical approval has been granted. If, during the course of the project work, any of the information supplied on this checklist changes substantially a new checklist will need to be completed and then brought to the attention of the CSC-30014 Ethics Advisory team.

Signed (Student) Richard Rhodes	Date: 7/11/19

I confirm that I have read the form and that the project, as proposed, is GDPR compliant and that, to the best of my knowledge, the ethical information is correct.

Signed (Supervisor) 	Date: 14/11/2019

Electronically typed signatures and dates *are* acceptable.

Computer Science Final-Year Project Ethics Application Form 2019-20

Applicant details

Project title	**Developing a Road Hazard Proximity Alert System**
Name of final-year project student	**Richard Rhodes**
Keele email address of student	**W8X41@students.keele.ac.uk**
Name of project supervisor	**Theocharis Kyriacou**

Project Summary

Provide a short summary of your project (max 250 words).

A mobile application that runs on the user's smartphone. The application allows registered users, such as the police, to activate a proximity alarm for other users of the app, in order to alert them of potential road hazards. The alarm's location is updated regularly, until it is automatically deactivated. The alarm's creator has the option to increase the time that the alarm is active for. Once a user is within a configurable distance (default 500m), they are audibly alerted to the presence of a hazard.

In simple terms, provide a *short* description of your planned experimental method that involves participants (e.g., focus groups, questionnaires, interviews, experimental observations).

Questionnaires to review certain aspects of the mobile application, such as screen designs and overall useability.

Describe the characteristics of your participants, and any inclusion or exclusion criteria. Estimate the approximate number of participants.

All participants must be able to drive

Approximately 20 participants will be recruited to answer various questionnaires

Recruitment and Consent

Indicate how potential participants will be identified, approached and recruited and outline any relationship between the researcher and potential participants.

Students and staff will be invited via email with the following invitation:

You are invited to participate in a study for a final-year project entitled: Developing a Road Hazard Proximity Alert System. Participation will involve approximately one hour of your time for 1-2 questionnaires concerning mobile application interface designs and useability. These activities will be scheduled according to your availability. If you are interested in participating, please read the Information Sheet and reply to this email stating your interest.

Describe the process that will be used to seek and obtain informed consent.

If individuals express an interest in participating, they will be provided with an Information Sheet via email. If they are happy to proceed they will be asked to sign a Consent Form before participating.

Will consent be sought to use the data for other research?
Yes ☐ No ☒

Will consent be sought to contact the individual to participate in future research?
Yes ☐ No ☒

Can participants withdraw from the research?
Yes ☒ No ☐

If yes, state up to what point participants are able to withdraw from the research
Participants can withdraw their data up to one week after participating.

If yes, outline how participants will be informed of their right to withdraw, how they can do this and what will happen to their data if they withdraw.

Contact information for withdrawal is provided on the Participant Information Sheet.
If no, explain why they cannot withdraw (e.g., anonymous survey).
N/A

Confidentiality and anonymity

Outline the procedures that will be used to protect, as far as possible, the anonymity of participants and/or confidentiality of data during the conduct of the research and in the release of its findings.

No names of participants will be used at any stage. If I need to refer to specific data, I will use participant numbers, e.g “Participant 7”. No names (or other information that might reveal participant identity) will be included in the project dissertation or in any other presentation.

Storage, access to, management of, and disposal of data

Describe how participant data will be stored; where it will be stored and for how long; and how/when it will be disposed of.

All participant data will be securely stored until the end of the project. All electronic records will be stored on university drives. Only the applicant (student investigator) and supervisor will have access to participant data. At the end of the project, the participant data will be deleted.

Other ethical issues raised by the research

Are there any other ethical issues that may be raised by the research?
Yes ☐ No ☒

If yes, please give details.
N/A

Declarations

Declaration by student

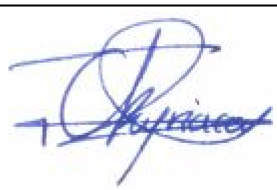
I confirm that:

- The form is accurate to the best of my knowledge.
- I will inform my supervisor and resubmit an ethical application if there are any changes to the project.
- I am aware of my responsibility to comply with the requirements of the law and any relevant professional guidelines.

Student name	Richard Rhodes
Student signature	Richard Rhodes
Date	18/12/19

Declaration by supervisor
I confirm that:

- I have read the application and am happy for it to proceed for ethical review.
- The application is accurate to the best of my knowledge.
- I am aware of my responsibility to ensure that the applicant is familiar with and complies with the requirements of the law and any relevant professional guidelines.

Supervisor name	Theocharis Kyriacou
Supervisor signature	
Date	/12/19

Poster

Road Hazard Proximity Alert

Student: Richard Rhodes, **Supervisor:** Theocharis Kyriacou

Introduction

In this project, I have created a system that warns users of road incidents and other hazards using an audible alert, which is suitable for when they are driving.

Project Description

A mobile application that runs on the user's smartphone. The application allows registered users, such as the police, to activate a proximity alarm for other users of the app to alert them of potential road hazards.

The alarm's location is updated regularly, until it is manually or automatically deactivated. The alarm's creator has the option to increase the time that the alarm is active for. Once a user is within a configurable distance (default 500m) they are audibly alerted to the presence of a hazard.

Aims & Objectives

Aim:
Create a mobile application that allows people to report and avoid road hazards

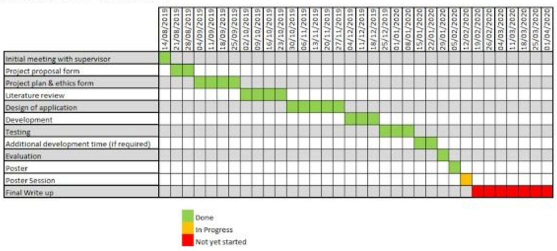
Objectives:
- Creation of a mobile application
- Creation of an easy to use interface
- Creation of an external web server for Hazard reporting
- Allow mobile application to report hazards to server
- Evaluate the project with user feedback

Methodology

I will be using iterative prototyping in order to develop the application. This involves creating a prototype of the application, evaluating the prototype, and then improving the prototype.

I followed these steps:
1. Develop a prototype
2. Test the prototype
3. Evaluate the prototype
4. Repeat steps 1-3 until the prototype has met all the objectives
5. Tidy up the code and release

GANTT Chart



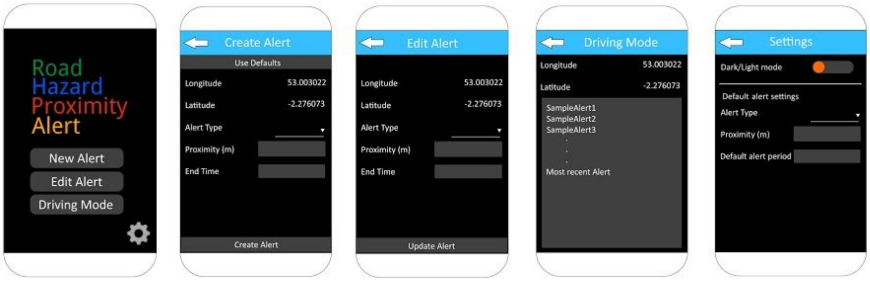
Web Database

The website is a platform to host the database.

This database holds information about registered users and alerts. It was built using SQLite and the interface was implemented in PHP. The website also makes use of phpMyAdmin so that a GUI can be used to manage the database.

User Interface

Below are the sample user interface designs that were created for development



References

Simulating Cooperative Road Hazard Warning Application From the Perspective of V2V - https://www.atlantis-press.com/proceedings/icimm-15/25653
Road Hazard Point Acquisition, The Smartphone Way - https://dspace.library.uu.nl/handle/1874/238575
Real-Time Multiple Sensors Microcontroller–Based Automobile Hazard Alert System - http://www.indusedu.org/pdfs/IJRESS/IJRESS_448_12990.pdf
Autonomous pro-active real-time construction worker and equipment operator proximity safety alert system - https://www.sciencedirect.com/science/article/pii/S0926580510000361
Co-driver Alert Project - https://www.infona.pl/resource/bwmeta1.element.ieee-art-000004562180