# **CA400 Final Year Computer Applications Project**

# **Functional Specification**

**‘Ice’** - a cross platform personal safety application.



Picture: our ICE application logo

Hannah O’Connor -16382283

Catherine Mooney - 16416052

Submission Date: X/11/19

# 0 Table of contents

**1. Introduction**

1.1 Overview…………………………………………………………………………………………………………………………\_

1.2 Business Context…………………………………………………………………………………………………………\_

1.3 Glossary…………………………………………………………………………………………………………………….…….\_

**2. General Description**

2.1 Product / System Functions……………………………………………………………………………….….\_

2.2 User Characteristics and Objectives………………………………………………………………….\_

2.3 Operational Scenarios…………………………………………………………………………………………….\_

2.4 Constraints…………………………………………………………………………………………………………………..\_

**3. Functional Requirements**

3.1

3.2

3.3

3.4

3.5

3.6

3.7

3.8

3.9

3.10

**4. System Architecture**

4.1 Overall

4.2 System

**5. High-Level Design**

5.1 Physical Data Flow Diagram

5.2 Logical Data Flow Diagram

5.3 Class Diagram

5.4 Use Case Diagram

5.5 Context Diagram

5.6 UI Sketches

**6. Preliminary Schedule**

6.1 Gantt Chart……………………………………………………………………………………………………………………..\_\_

6.2 Task List…….……………………………………………………………………………………………………………………...\_\_

6.3 Product Backlog

**7. Appendices**

7.1 Apendices………….………………………………………………………………………………………………………………..\_\_

# 1 Introduction

## **1.1 Overview**

Safety has always been a big concern and priority for adults and their children. From doing regular day to day errands to making your way home, alone, especially at night. Parents are constantly worried about where their children are and if they made it to their destination ok. Young girls have the constant fear of getting into taxis unaccompanied or walking down unlit streets. Men and women of all ages are aware of the on-going dangers faced by society today. It’s inevitable that people have to make trips alone, be it short or long journeys. But it wouldn’t be near as daunting if you have reassurance that your guardians know where you are and that help is easily accessible. While we have many uses for our smartphone devices, one thing that it could have great potential for is personal safety. This is where our application comes into use, providing that reassurance and extra comfort for those within today's society, through your phone.

ICE is a cross-platform iOS and Android personal safety mobile application built through the React Native framework. The name comes from the well known ‘In Case of Emergency’ and we took inspiration from the name ICE for the apps logo. The app will be designed for all types of user, with the idea being that only those within your trusted circle of contacts can monitor your real time location, communicate and be updated with notifications and alerts about your current status. Our mission is to establish a platform aiding those without protection and help to make a difference to when a situation could get serious. We plan to develop a web application alongside this, one that can be accessed via phone or desktop as another means of receiving updates from those within your inner circle. The central idea behind the application is that the app can provide a variety of personal safety features rather than just contains an SOS button and Gps tracking. Most personal safety app developed globally are delivered on a subscription basis and we believe that the price of safety should be free of cost.

## **1.2 Business Context**

As the safezone app is already very popular among students of DCU, we can already see that there is a desire for apps like this. However, ICE is not only for students, it is for the general public. We did some research and found that there is nothing like this on the market in Ireland designed for men, women and children alike.

ICE could be made into new instances and potentially be utilised within schools and universities, where they could have their own customized version. The SafeZone app is utilised specifically for institutes, Universities like DCU for example, and only cover the grounds of that institute, used to alert security of any emergencies. Our app, however, cover the user no matter where they are, including trips to and from these institutes and there is no security within your range.

Another possible direction for this application is to allow advertising upon uploading to App/Google Play store & generate revenue to support the future growth of this application.

## **1.3 Glossary**

**React-Native**

A JavaScript framework, based upon React, which allows you to create cross-platform applications for iOS, Android & Web Applications. All from a single codebase.

**Firebase**

A mobile application cloud-based development platform owned by Google, it acts as the server and provides tools in order to develop our app.

**Firebase Real-time Database**

No SQL, cloud hosted database that syncs and stores data across our app users within real time.

**Firebase Cloud Storage**

Firebase SDK to allow for the storage of user-generated content, such as photos.

**Firebase Authentication**

A service that can authenticate our users using through email and passwords. When a new user signs up with an email, it sends a verification email to that address before storing to database. It handles sending password reset emails too.

# 2 General Description

## **2.1 Product / System Functions**

Displayed below are the main functions in which we plan to incorporate into our app, the functionality is subject to change over the course of development, however, for the time being, our proposed functions are listed below. These functions will require a user to, first, be registered successfully and verified by the phone number.

* Network of Trust
* Network of Trusts Invitations (Add others/Be apart of others)
* SOS activation
* Network alerts & Tracking
* Push notifications
* Follow me session
* Fake call activation
* Emergency text generator
* User profiles
* Medical profiles
* Safe zones
* Safe word
* Social media feed
* Fingerprint access to modify user information
* Messaging Service
* SMS functionality

The above functionality will be discussed in further detail within section 3 *‘Functional Requirements’*.

## **2.2 User Characteristics and Objectives**

We plan to design our app in a way that is open to users of all ages. Our app is not exclusive of any particular groups as we feel personal safety shouldnt be an issue for anyone. The only condition which our users must adhere to is that they must carry a smart phone or device with a registered phone number.

Given that this application will be available to all, we want to establish a universal appeal for our user interface. Our apps logo is very welcoming so that from the outset, users feel relaxed in the knowledge that the app will be user friendly. From the moment the user enters the app, we plan to ensure ease of use throughout.Our main goal for the UI of ICE is to allow for usability, readability and consistency. We will make use of fun and bright, but distinguishable colors and be considerate of font sizes and controls. We will make sure that transition from one section of the app to the other is very smooth and easily navigated. Whether acquiring rich technology skills or not, this app will be accommodating for all. We will base all UI decisions off our acquired knowledge from the previous year module CA357 (Human Computer Interaction) .

## **2.3 Operational Scenarios**

**User signs up**

Following the user having installed the application onto their Android or iOS device, that user can then proceed to registration. If it the first time on the application or there has been no prior registration, then the user will be presented with a sign up screen. The user will be required to enter their information to the mandatory field. When completed, a message will be sent to the inputted phone number where the user will be required to verify it. Upon successful verification, the user must agree to the apps’ terms and usages. The given phone number will be automatically registered with the 112 SMS service & be able to operate the Emergency service messaging function.

**User edits their profile**

Following sign up, through navigating to ‘User Profile’ from within the main profile, the user will be presented with their profile. To modify it, the user must click on the ‘Edit’ button. Here the user can modify the following stored data:

* Medical Profile
* Safe Zones
* Circle of trust
* Safe word

Upon completion the user must click ‘Save’ at the bottom of the page.

**User starts a ‘Follow me’ session**

From navigating to the ‘Follow me’ from within the main menu, the must click on the function and wait for it to load. Here, the user will be asked to enter their destination and a route and approximated arrival time will be calculated. The user will have the option to modify this estimated time before clicking ‘Start’. Following this action, a notification will be sent to each member of the users’ network and their real time location will be retrieved and stored in database. Any given member of the user’s network can query the location through clicking on the received notification from app or checking their inbox.

**User triggers a network alert.**

Any given user of the app may want to alert their network of their real time location for some period of time without explicitly declaring any particular destination or arrival time. Here, the user will have clicked on the ‘Network Alert’ function and notification will be sent. The location of the user will be monitored within the background and can be viewable by any in their network.

**User adds/navigates to social media feed.**

Given that the user is within the main menu, they will be presented with the ‘Social Media’ function. Here, the user must click on it and wait for it to be loaded. Upon being within that function, the user can tap the ‘+’ icon in the top right corner. A forum will be presented where can enter the text for their new post. The option to post to either to their network or to the public will be available. Once done, the user must click ‘Share’.

**User activates SOS mode.**

A user can activate the SOS mode in one of three ways. These include:

* User says the safe word aloud whilst holding their device.
* User rigorously shakes their device.
* The button ‘HELP’ on main menu is touched by the user.

The activation of any of the above methods will create a 10 second countdown where the user can choose to cancel or wait. Notifications & alerts will be sent to the network. The mobiles' audio and camera footage will be recorded and sent to cloud storage where it can be then retrieved by their network.

**User sends an emergency text message**

User can send an emergency text through tapping on the function within the applications’ main menu. Here the user will be presented will two modifiable attributes. These include:

* Which emergency service type is required ( a scrollable list )
* Why its’ required. ( The default for this is: ‘I need help and am unable to speak’)

The user’s location will be queried and reversed geocoded to a human reabled address. The user must then press ‘Send’

**User generates a fake call.**

User can generate a fake call through navigating to the main menu and tapping on the ‘Fake Call’ button. Upon the loading of the function, the user can click ‘Call me’. The default template created by the user will be shown and can be changed before it is activated.

## **2.4 Constraints**

**Time constraint**

As we are quite restricted on our app development process due to the time, we will focus on implementing the basic functionality and ensure it all works accordingly, with sufficient testing conducted. Over time we can grow the application to incorporate more features and functionality to it. However, the management time will be need to be a top priority to our team. Especially with the additional juggling other lectures and coursework too.

**Firebase database constraint**

With a limit set on our free account on Firebase, we’re constrained on the amount that we can store and on the amount of simultaneous connections to our database. (The in-app messaging may be a concern, as it will take up most of the space.)

**Internet constraint**

In terms of the user tracking, if there is a loss of Internet connection on the mobile device then the real time location will be able to be retrieved & updated on private map to network of trust. A user must have internet access when attempting to open/operate the corresponding web app.

**Hardware constraint**

There is a hardware constraint when it comes to testing our devices on our machines as iOS devices can only be tested on OSX. Given that each member has a separate OS type on our laptops, it means that one can take over the management and testing of iOS on a MacBook and the other takes care of the android platform with their windows machine.

# 3 Functional Requirements

## **3.1 Registration**

***Description***

Registration is the first step and a requirement to using this application. The sign up form will require the new user to enter a valid mobile phone number. This step is crucial as it is how we can tie the user to that device. This avoids the possibility of a user being logged out and unable to activate safety functions .Other basic information will be required too, like name, email, etc.

***Criticality***

Registration is our most crucial function. Without it, users’ of the app wouldn't be able to access our services. Through this function we want to enforce our rule to that only one mobile can be tied to a single account at any given time.

***Technical issues***

For the phone number validation process we will leverage Firebase’s Authentication SDK.

***Dependencies with other requirements***

The user must have the application installed on their iOS/Android mobile device. Alongside that devices’ phone number.

## **3.2 Account Set Up**

***Description***

* Following successful registration
* Add members to circle of trust via phone number?
* Requests will be sent to those chosen members
* When accepted, update the network of trust
* Any request for the given for other circles to join will be located in the home menu.
* Add Medical Info - Skip and edit later.

*A full description of the requirement.*

Establish your network, set up your profile, enter your medical information.

***Criticality***

*Describes how essential this requirement is to the overall system.*

***Technical issues***

*Describes any design or implementation issues involved in satisfying this requirement.*

***Dependencies with other requirements***

User is required to be successfully registered with the app.

## **3.3 Follow me Session**

***Description***

*A full description of the requirement.*

***Criticality***

*Describes how essential this requirement is to the overall system.*

***Technical issues***

*Describes any design or implementation issues involved in satisfying this requirement.*

***Dependencies with other requirements***

*Describes interactions with other requirements*

## **3.4 SOS Mode**

***Description***

* The detailing of this will then be stored in the user's activity log to access footage from previous SOS activation.

***Criticality***

*Describes how essential this requirement is to the overall system.*

***Technical issues***

*Describes any design or implementation issues involved in satisfying this requirement.*

***Dependencies with other requirements***

*Describes interactions with other requirements*

## **3.5 Network Alert**

***Description***

*A full description of the requirement.*

***Criticality***

*Describes how essential this requirement is to the overall system.*

***Technical issues***

*Describes any design or implementation issues involved in satisfying this requirement.*

***Dependencies with other requirements***

*Describes interactions with other requirements*

## **3.6 Emergency Services Message**

***Description***

Template

***Criticality***

*Describes how essential this requirement is to the overall system.*

***Technical issues***

* Network provider
* Need to be registered to 112, hence why we do in within the user registration so this function can be used at any given time once registered.

*Describes any design or implementation issues involved in satisfying this requirement.*

***Dependencies with other requirements***

## **3.7 Social Media Feed**

***Description***

*A full description of the requirement.*

***Criticality***

*Describes how essential this requirement is to the overall system.*

***Technical issues***

*Describes any design or implementation issues involved in satisfying this requirement.*

***Dependencies with other requirements***

*Describes interactions with other requirements*

## **3.8 Safe Zones**

***Description***

The idea behind this is that rather than a user having to constant send reminder to their network, ICE will send a notification to those within your circle by default when you enter one of your defined safe zones.

***Criticality***

*Describes how essential this requirement is to the overall system.*

***Technical issues***

Through Using geofencings

***Dependencies with other requirements***

*Describes interactions with other requirements*

## **3.9 Fake Call**

***Description***

*A full description of the requirement.*

***Criticality***

*Describes how essential this requirement is to the overall system.*

***Technical issues***

*Describes any design or implementation issues involved in satisfying this requirement.*

***Dependencies with other requirements***

*Describes interactions with other requirements*

# 4 System Architecture

The below system architecture diagram shows the relationships between the different internal and external components of our system design.

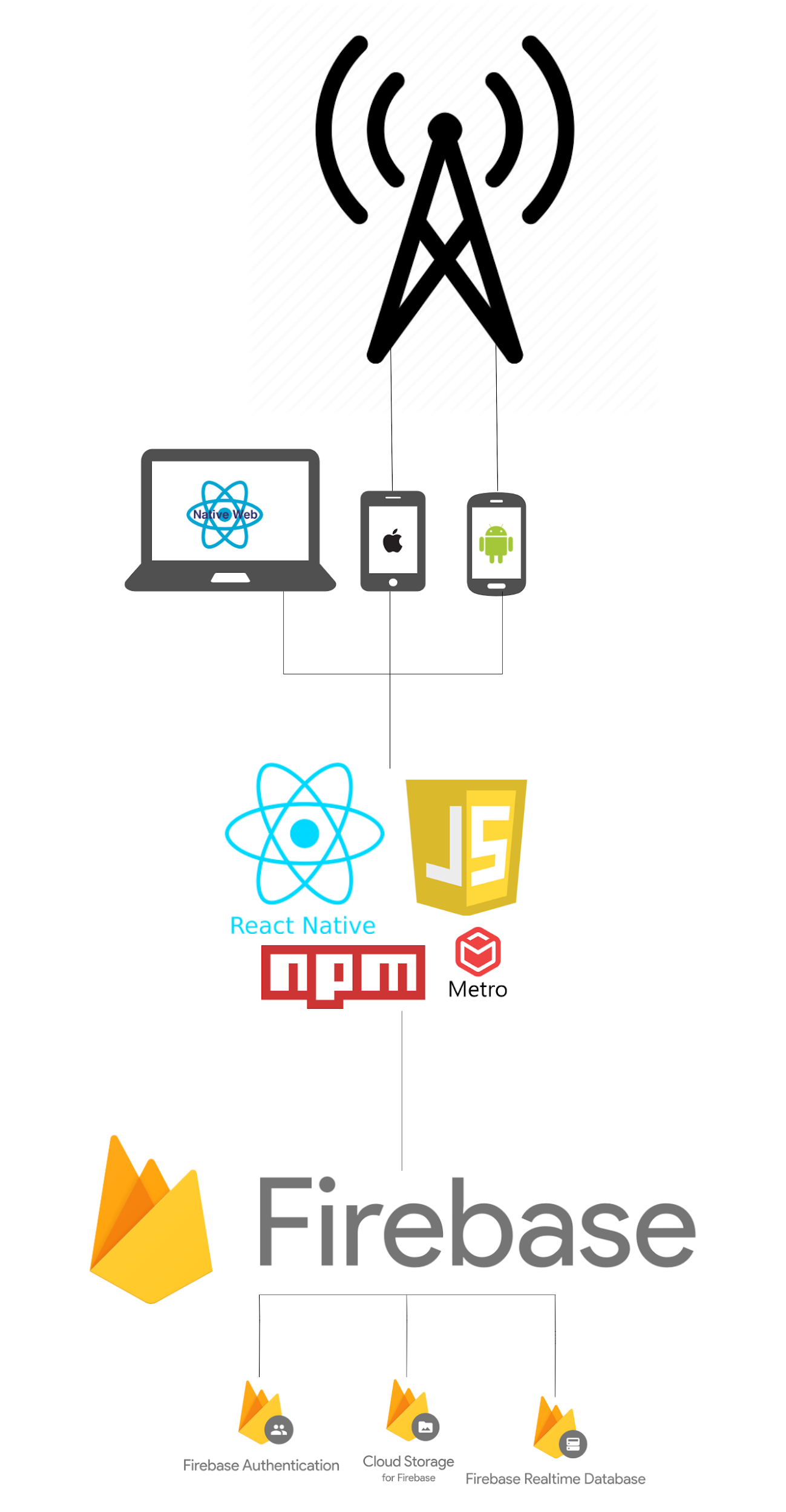
At the top level of the diagram, it shows the hardware components. Those representing the iOS and Android device. Along with the Web application. The tower is a representation of our plan to utilize the mobile devices’ network provider to send SMS messages from the application.

Below that shows the main software components of this application and what will be used to create it. The pictorial logos show the following: React Native, JavaScript, npm and Metro, a JavaScript Bundler. Our codebase will live within GitHub and will be written within VSCode where we can take advantage of its extensions and leverage JS linting and testing tools. It will also integrate multiple React Native libraries and APIs. Such examples include: react-native-shake, react-native-map, react-native-sms and react-native-phone-call. All of which will be required for the various functionalities that we plan to offer.

The bottom level of this diagram is our applications’ backend. Here the main component in this will be Google’s Firebase. Firebase itself will be responsible for the hosting of ICE. We will also utilize some of it’s available SDKs too. These include it’s user authentication system, the real-time cloud database and the cloud storage.

## **4.1 System Architecture Diagram**

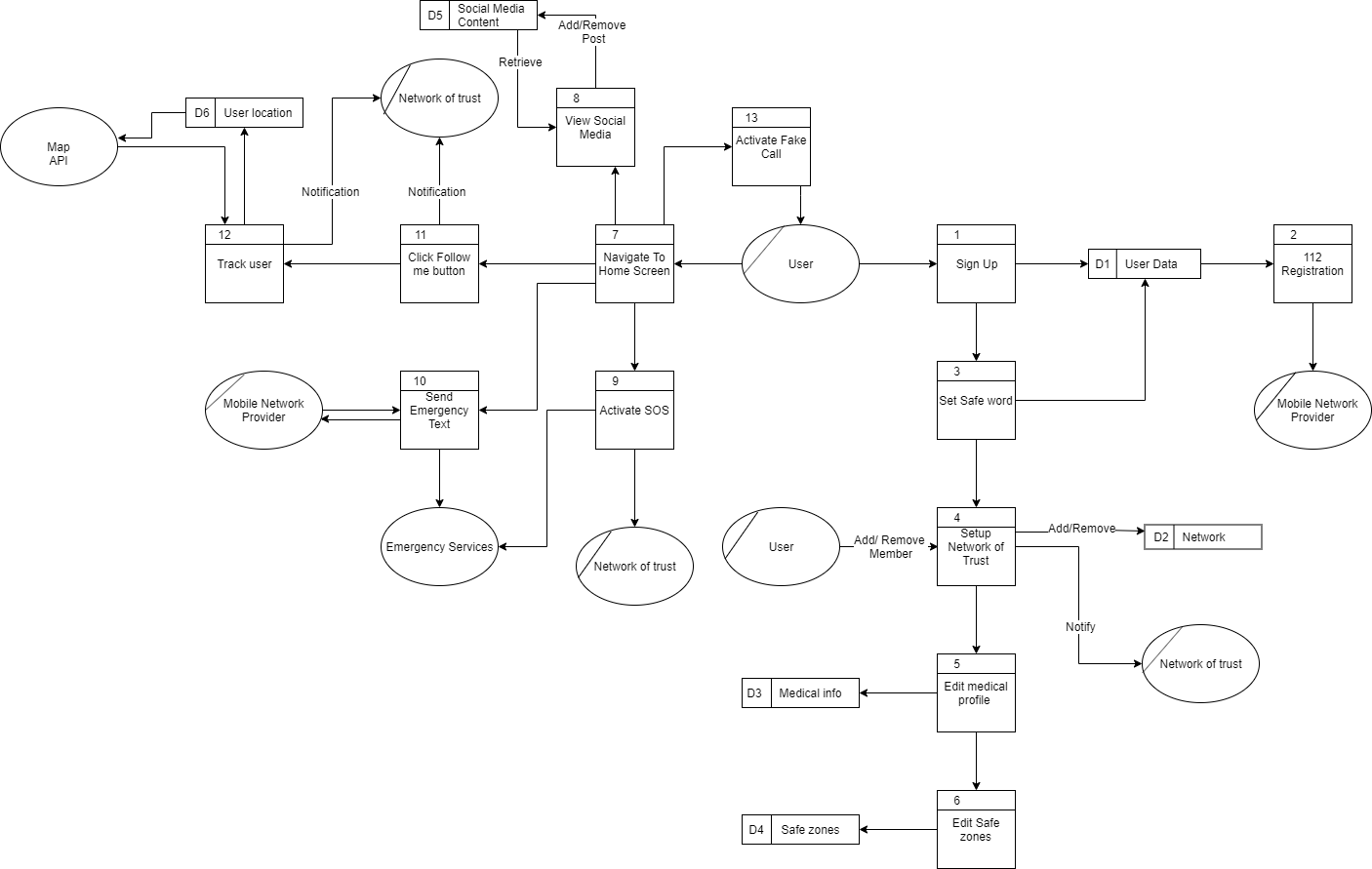
Diagram



# 5 High-Level Design

(Brief)

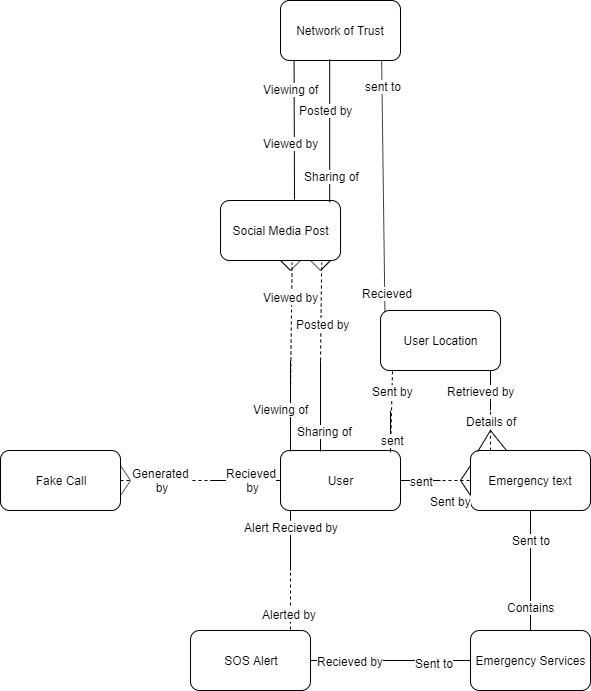
## **5.1 Physical Data Flow Diagram**

Diagram

Description

## **5.2 Logical Data Flow Diagram**

Diagram



Description

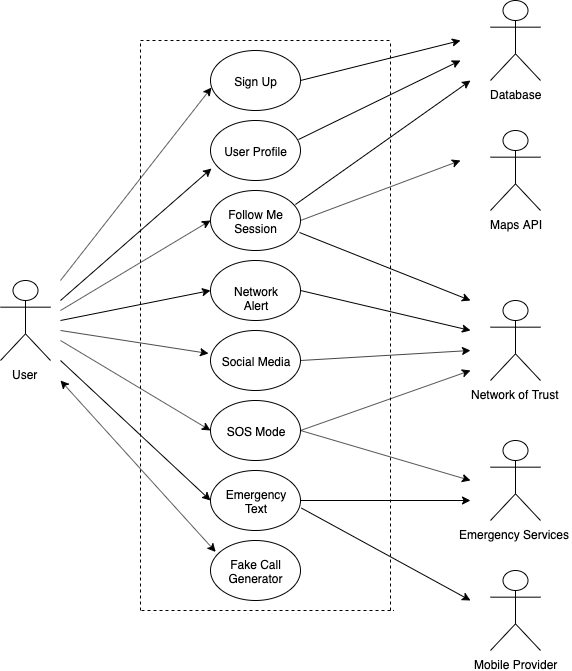
## **5.3 Class Diagram**

Diagram

Description

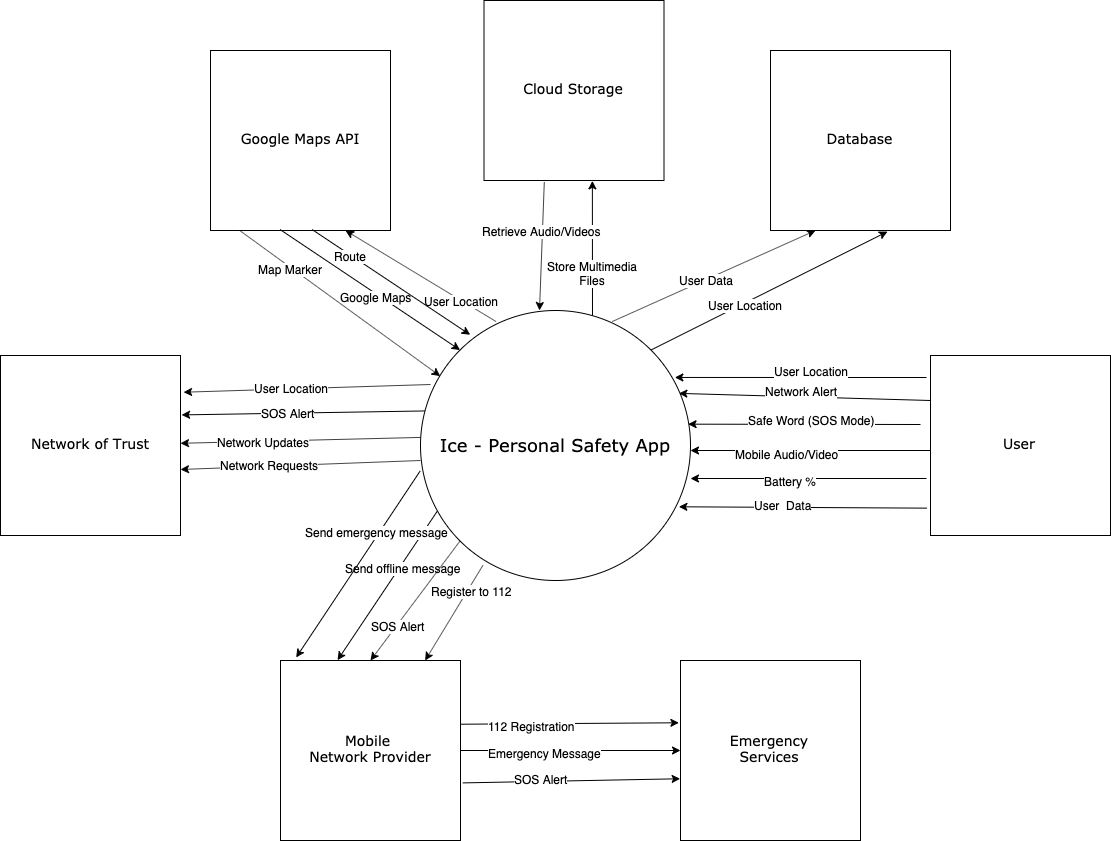
## **5.4 Use Case Model Diagram**

Diagram



Description

## **5.5 Context Diagram**



## **5.6 UI Sketches**

-> maybe add some UI sketches/Plans here too and discuss the considerations we would like to make when it comes to our interface.

# 

# 

# 6 Preliminary Schedule

## **6.1 Gantt Chart**

Chart

Description

## **6.2 Task List**

Chart

(Hannah) I’m currently working on this on laptop will copy over when done.

Description

## **6.3 Product Backlog (Trello Board)**

Screenshot of board

Brief description

Link: <https://trello.com/b/sDfu1uo7/ca400-fyp-project>

# 7 Appendices

* React Native:
* Firebase: [https://firebase.google.com](https://firebase.google.com/)
* Firebase Cloud Storage: <https://firebase.google.com/docs/storage/>
* Firebase Real-time Database: <https://firebase.google.com/docs/database/>
* Firebase Phone Authentication:
* 112 SMS functionality: <https://www.112.ie/Terms___Conditions/146>.
* 112 How to: <https://www.112.ie/Sending_a_text_to_112/144>
* Learning React Native (O’REILLY book): <https://www.oreilly.com/library/view/learning-react-native/9781491929049/>