Report on SafeWay: A Personal Safety App

University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Department of Computer Science

University College Cork

Ryan Lacey

April 2023

Final Year Project - BSc (Comp Sc) Single Hons (Software Ent)

Supervisor: Dr. Klaas-Jan Stol

Second Reader: Prof. Ken Brown

Abstract

SafeWay is an iOS app that provides safety features with a minimalistic UI to ensure ease of use in times of crisis. SafeWay aims to connect users and provide peace of mind to both concerned peers and those who may be in vulnerable positions themselves.

The app provides location sharing features as well as features to check on the well-being of those who one may be concerned about, for example an checking in on individuals walking alone, in dangerous situations or ensuring peers have made it home safely.

This report details the measures taken to design, plan, and develop the application as well as an evaluation covering the quality of the app as well as feedback on what could have been improved.

Keywords: Mobile, iOS, Safety

Technologies: Swift, ObjC, Firebase

Declaration of Originality

In signing this declaration, you are conforming, in writing, that the sub- mitted work is entirely your own original work, except where clearly at- tributed otherwise, and that it has not been submitted partly or wholly for any other educational award.

I hereby declare that:

- this is all my own work, unless clearly indicated otherwise, with full and proper accreditation;
- with respect to my own work: none of it has been submitted at any educational institution contributing in any way to an educational award;
- with respect to another’s work: all text, diagrams, code, or ideas, whether verbatim, paraphrased or otherwise modified or adapted, have been duly attributed to the source in a scholarly manner, whether from books, papers, lecture notes or any other student’s work, whether published or unpublished, electronically or in print.

Signed:.....Ryan.......... Date: 23/04/2023

Acknowledgments

I would like to express gratitude for my parents Paul and Maria Lacey who have helped and supported me throughout my time at UCC, as well as peers Tadhg and Mark for their academic encouragement.

Table of Contents

Abstract .... ***i***

Declaration of Originality .... ***ii***

Acknowledgments .... ***iii***

Chapter 1 Introduction ***1***

Chapter 2 Analysis ***4***

2.1 Introduction **4**

2.2 Use Cases of Safe Way **4**

Use Case 1: Registration 5
Use Case 2: Authentication 5
Use Case 3: Connect With a User 6
Use Case 4: Panic Button Pressed 6
Use Case 5: Sending a Check-in Request 7
Use Case 6: Responding to a Check-in Request: 8
Use Case 7: Mark as Home Safe: 8

2.3 Core-Values of SafeWay: **8**

2.4 Existing Applications **9**

2.4.1 bSafe 10
2.4.2 Life360 11

Chapter 3 Design ***13***

3.1 Introduction **13**

3.2 GitHub **13**

3.3 Firebase **13**

3.4 Firebase Authentication **14**

3.5 Database Architecture and Design **14**

3.5.1 Firestore 14
3.5.2 Firebase Realtime Database 16
3.5.3 GeoFire 17

3.6 Interface Design **17**

3.6.1 Interface Design Tools 18
3.6.2 Colour Palette 18

3.7 UI Layouts **20**

3.8 Designing the Panic Buttons **21**

Chapter 4 Implementation ***22***

4.1 Introduction **22**

4.2 Completion Handlers and Database Calls **22**

4.2.1 Snapshots 23

4.3 Issues with Firebase Auth **23**

4.4 Utilities **25**

4.5 Location sharing **25**

4.6 Panic Buttons and Messages **26**

4.7 Following a User 27

Chapter 5 Evaluation 28

5.1 Introduction 28

5.2 Focus group 28

5.2.1 Focus group selection 28
5.2.2 Features Evaluation 29
5.2.3 User flows evaluation 30

5.3 Survey Evaluations 30

5.3.1 Survey design 30
5.3.2 Survey responses 31

Chapter 6 Conclusion 34

6.1 Introduction 34

6.2 Summary of Conduction 34

Plan 34
Design 34
Development 35
Testing 35
Deployment 35

6.3 Reflection 35

References: 37

Appendix 1 39

Chapter 1 Introduction

The rate of technical progress is increasing exponentially and in general has benefitted mankind immensely. Computer technology has improved virtually every aspect of life including medicine and undoubtedly has saved countless lives. Similarly, we can see advancement in safety features such as automatic breaking and anti-lock braking systems across virtually all car brands. On a smaller level, it has provided many online services, apps, and websites which make day-to-day life easier and safer. Nonetheless, society is still faced with many issues that technology could potentially mitigate or alleviate. One such issue is travelling alone or being in unsafe and uncomfortable areas or situations. The application developed and described in this report aims to address this issue.

Many trivial apps are readily available and have achieved widespread use. In contrast, apps that provide a measurable improvement in personal safety and crime avoidance remain scarce. This is likely due in part to the fact that most popular apps focus on fun and convenience, rather than issues that we hope will never occur. Nonetheless, the need to ensure safety is a real and will not disappear. While crime prevention is ultimately the responsibility of the Gardaí, there are steps we can take to reduce personal risk. Given the frequency and impact of such offences even a small percentage reduction would justify the time and effort spent in developing SafeWay.

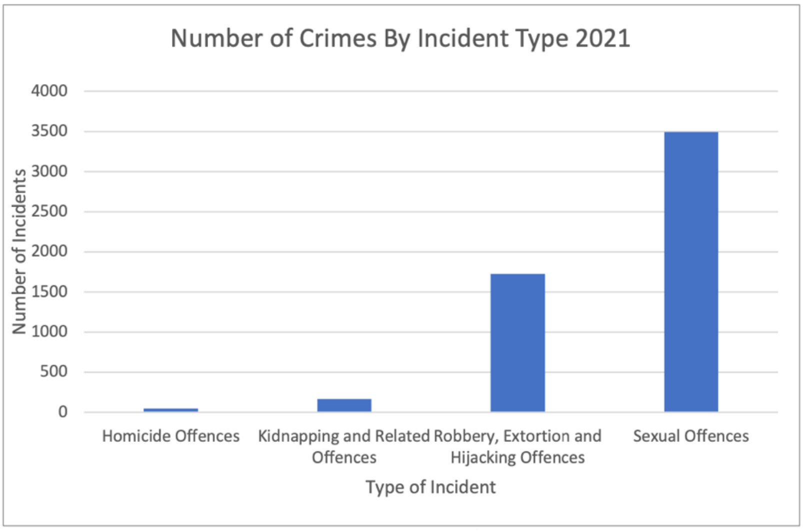
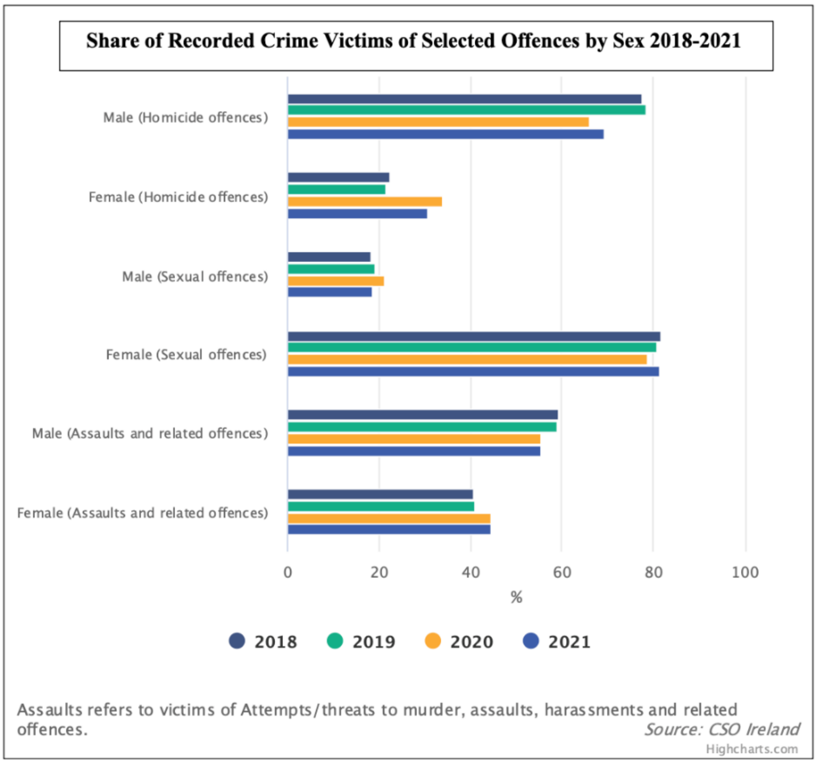


Figure 1: Number of Crimes by Incident[1] *[1]*

[1] The category of Attempts/Threats to Murder, Assaults, Harassment and Related offences is excluded from the graph for readability, see appendix for full graphic.

Figure 1 shows the number of crimes by incident type in Ireland in 2021. These are serious crimes against the person and occur with unacceptable frequency, being measured over a period of only 12 months. There is a clear need for improved safety and communication, especially amongst young adults who tend to be more vulnerable while socializing late at night. SafeWay aims to bring a serious and impactful solution to help counter this situation.



Assaults refers to victims of Attempts/threats to murder, assaults, harassments and related offences.

Source: CSO Ireland

Figure 2: Recorded Crime Victims of Selected Offences by Sex 2018-2021 [2]

Figure 2 shows offences segregated by the gender of the victim. Overall, the data shows women are four times more likely to be the victim of a sexual offence in Ireland. Approximately 56% of the victims are less than 18 years old, with a further 27% being between 18 and 29 [2]. This unacceptable statistic is a primary factor behind creation of SafeWay and leads to the app being primarily targeted at teens, young adults and family members, whilst providing benefits for all demographics.

SafeWay is an iOS application developed using Apples programming language; Swift. The aim of the application is to provide safety and peace of mind in a user friendly format. The app provides a platform which allows users to establish the well-being of other app members. In addition, it provides location sharing services to give peace of mind by allowing contacts[2] to determine when users arrive home safely or conversely to receive panic messages with an exact location in the event of an emergency. These and other features are described in the following report, which explains the main safety use cases of SafeWay in greater detail. The objective for the SafeWay app is to provide maximum user benefit, i.e. improved safety, while remaining easy to use. With this in mind the user interface is intuitive, even for individuals with limited online experience. In particular, the app is designed such that minimal input is required in the event of an emergency.

[2] In this context, the term "*contacts*" will be used to describe the list of users that the user in question is connected with via the SafeWay app.

Chapter 2 Analysis

2.1 Introduction

Chapter 2 discusses the functional objectives and use case scenarios for SafeWay. Analysis of existing applications is conducted in order to determine the state of the art and where these applications are lacking in order to achieve a better understanding in how SafeWay can be implemented.

2.2 Use Cases of Safe Way

The first step in the development process is to clearly define the expected use-cases to ensure a functional and efficient application fit for its intended purpose. The use-case diagram is a useful tool to visualise these objectives, followed by an in detail description of how each use-case must function in the final system.

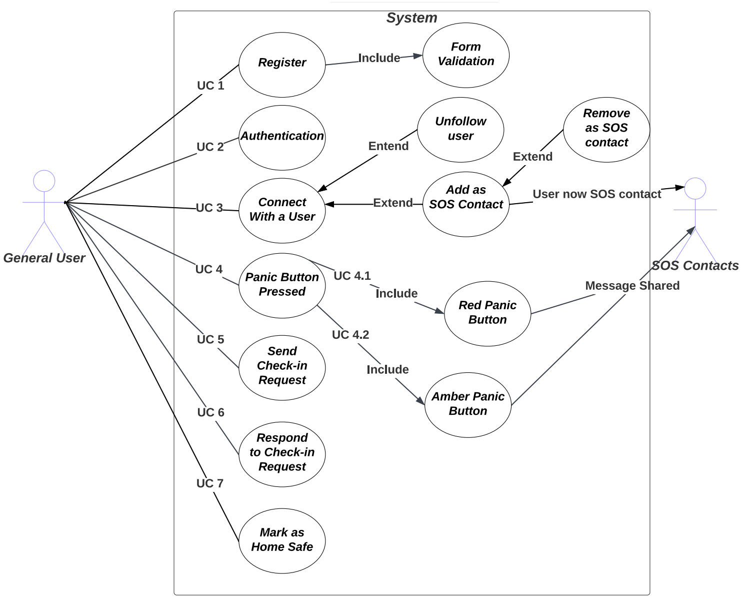


Figure 3: Use Case Diagram of SafeWay[3]

Figure 3 summarises the use-cases of SafeWay in a visual format. Each use-case is necessary to fulfil the overall objective to create a system of personal protection, as follows.

[3] UC refers to "Use Case" e.g. UC 1 means Use Case 1

Use Case 1: Registration

Use case 1 refers to the process a user completes in order to obtain an account within the SafeWay system. The process includes form validation, which ensures that the e-mail address is well formed, the names to not contain numbers or inappropriate characters and finally that the password meets a secure standard being at least eight characters long with one numeric character and one special character to be deemed as properly formed.

If all the above conditions are met it must be established that the email used during registration is unique and unassigned to any existing accounts, the same is done for usernames to prevent errors within the database and when connecting with users as will be described in use case 3.

Success Scenario:

1. The email and username are unique and the user data is correctly formed. An account is created for the user, the user is authenticated and presented the home page of SafeWay.

Alternate Scenarios:

1. Username is already in use and must be changed.
2. Email is already in use and must be changed.
3. A field is incorrectly formed and must be amended.

Use Case 2: Authentication

Authentication is conducted before a user can access any features within SafeWay using an email and password combination. Once authenticated a user is brought to the home page of SafeWay. Users remain authenticated until manually unauthenticated via a “logout” button. This ensures the application is easily accessed, as authentication is completed prior to any need to use SafeWay, i.e. in an emergency situation.

Success Scenarios:

1. Email and password match with the backend. The user is authenticated and shown the home page of SafeWay.

Alternate Scenarios:

1. Username incorrect, authentication failed.
2. Password incorrect authentication failed.

Use Case 3: Connect With a User

Users search for and connect with other users to establish a list of partners that the safety features can be used with. Many of the use cases which are described require the ability for two users accounts to communicate.

Extensions:

1. Once a user is followed[4], they must be able to mark the aforementioned user as an SOS contact. SOS contacts are notified in the case of an emergency and should normally be trusted close peers of the user.

Success Scenarios:

1. The username exists and the profile of the user is shown. If the "Follow user" button is pressed the user appears in the "Following" tab.

Alternate Scenarios:

1. No account with the searched username exists, no profile is shown.
2. The account searched is already followed by the user and the "Follow user" button is greyed out.

Use Case 4: Panic Button Pressed

Activating the red panic button.

When the red panic button is activated an alert is sent to the SOS contacts of the user in distress. For example, if the user presses the panic button, all of the individuals marked as an SOS contact are sent the live location of the user, "the code red" status and the time at which the button is pressed Initially red panic button was intended to phone an emergency line. However due to limitations by Apple, iOS applications are unable to make a call with one push of a button. Multiple inputs from the user are not feasible in an emergency situation and go counter to the philosophy of the app: quick and simple to use safety features in high pressure situations.

Activating the amber panic button.

The amber panic button works in a similar manner to the red panic button, but is used pre-emptively when a user would like a sense of safety when entering locations where they feel uncomfortable or threatened.

Both buttons provide the option to be used as a "release switch". This means as long as the button is held down nothing happens, but once the finger is removed the panic message is automatically transmitted. This method of activation allows

[4] "Following" a user means you are connecting with them via SafeWay

users to hold down a panic button as they walk and if they are approached or knocked and the phone falls on the ground, the button is activated and transmits a panic message. If the user wishes not to send a panic message they can slide their finger off of the button to cancel the action. Both alarms work in a silent fashion. A silent alarm is chosen over audible as in high pressure situations subtlety may be an asset and allow the user to buy time until SOS contacts are alerted and so not to escalate the situation. This approach is comparable to the use of silent alarms in places of businesses and banks.

Success Scenarios:

1. Red panic button is pressed and a red alert message is sent to SOS contacts of the user.
2. Amber panic button is pressed and an amber alert message is sent to SOS contacts of the user.

Alternate Scenarios:

1. The button is pressed but the user has no SOS contacts assigned. The button does not activate and the user will be notified that there are no SOS contacts to share the message with.
2. The user presses their finger down but slides off the button before raising the finger. In this case nothing happens and the button does not activate.

Use Case 5: Sending a Check-in Request

A check-in request is a means for users to keep in touch and establish the wellbeing of friends and family. The feature creates a “Check-in request” that is sent to the Check-in requests inbox located in the “following tab” of the application. This feature prompts the receiving user to tap a button, which checks the user in as safe. An appropriate message appears on the profile of the user and indicates to users following that they have checked in as safe with a time stamp. This feature allows for concerned peers to send quick check-ups during events such as nights out or while separated for any reason.

Success Scenarios:

1. A check-in request is successfully sent to the desired receiving user.
2. A check-in request has already been sent, the sending user presses the button again to replace the old request with another more recent one.

Alternate Scenarios:

1. The user is not following anyone to send a check-in request to.

Use Case 6: Responding to a Check-in Request:

Once a check-in request is received it appears in the check-in requests inbox. A button is located on the bottom right of the request to reply and mark the receiving user as safe.

Success Scenarios:

1. A request is received and is responded to successfully.

Alternate Scenarios:

1. The inbox is empty and there are no requests to respond to.

Use Case 7: Mark as Home Safe:

Users can mark themselves as "home safe". A user who is "home safe" has a house icon beside their username on their profile which is timestamped when they arrived home. This house icon is not permanent and rather expires after a certain time period in order for reuse of the button and to prevent the need to manually reset this feature, i.e. to align with the simplistic experience SafeWay aims to provide.

Success Scenarios:

1. User successfully marks themselves at home.

Alternate Scenarios:

N/A

2.3 Core-Values of SafeWay:

The core-values define the overall concepts and approach employed while developing the SafeWay app. These values dictate the design choices and methods with which the features are implemented to ensure the envisioned user experience, as follows:

1. Minimalistic UI

SafeWay imposes a strict colour palette for development. It is important the key features are immediately apparent and the application be intuitive to use in high pressure situations. Colour theory is researched in order to choose colours which sooth and provide calming effects.

2. Minimum Clicks to Completion

When developing a feature for SafeWay, ensure the amount of "clicks" needed for the use case to be completed be kept to a minimum. This is a safety app for which the intended use is a high stress environment. The minimum app interaction required the more likely a success scenario is achieved.

3. Ensure the app is accessible to all

The app be intuitive and easy to use. The application is targeted at a wide audience base, some of which may not be familiar with complex computer applications. The app is also to be inclusive of visually impaired individuals and provide accessibility options where possible. This is achieved by keeping text to a minimum for cases where the user may be using a screen reader.

4. Safety comes first

User safety must remain the key focus of the application. Notifications are kept to a minimum and eliminate non-essential messages unless directly pertaining to safety, i.e. the application does not send a notification to the user when they are followed by another user, as this could create a situation where a panic message is lost amongst other unimportant notifications not pertaining to safety. Rather, the approach uses subtle in app cues, which are not sent to the phone notification centre.

2.4 Existing Applications

In the current saturated iOS application market, it is difficult to create an app which has no overlap with an existing app. Many of these existing applications are lacking in certain areas however and can be improved on. A review of existing applications is performed to identify the desirable aspects and avoid poorly highlighting features, while also implementing unique characteristics that set SafeWay apart.

In particular, "*bSafe*" is an iOS app currently posted on the apple app store and aims to provide similar personal protection features. bSafe is downloaded and analysed to see how these features were implemented, and while the features may have been aiming to accomplish similar goals, the app is lacking in certain areas providing ample room for improvement.

2.4.1 bSafe

1. Overwhelming user interface

The most apparent avenue for improvement is the failure to keep the interface simple and uncluttered. In particular, the bSafe app employs a bright colour palette combined with an overwhelming amount of information in the application views.

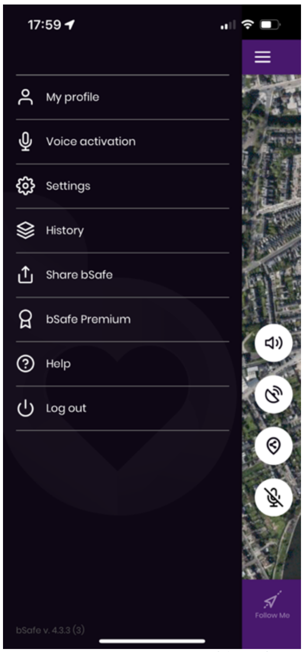
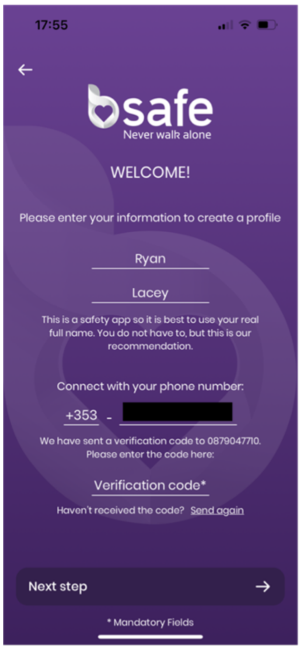


Figure 4: Menu of bSafe



WELCOME!

Please enter your information to create a profile

Ryan

Lacey

This is a safety app so it is best to use your real full name. You do not have to, but this is our recommendation.

Connect with your phone number:

+353 - [redacted]

We have sent a verification code to 0879047710. Please enter the code here:

Verification code*

Haven't received the code? Send again

* Mandatory Fields

Figure 5: Registration for bSafe

Figures 4 and 5 show screenshots of pages found within bSafe where the app appears convoluted and strays from the first and third core-value of SafeWay. This overwhelming amount of information, in even the registration, may confuse those who are not technically advanced or using screen readers. The menu also contains eight different sections, some of which are promoting in app purchases and sharing the application, which is unnecessary and can be done via the home screen of the phone or the app store.

The white icons visible in figure 4 also are not self-explanatory and their function is unclear. This may become clearer after familiarisation with the app, but is actively avoided in the design of SafeWay.

2. Buggy implementation

According to the app store, bSafe has not been updated in 5 months, this is worrying as from a brief use of the app quite a few bugs appear. The geographic location shown on the screen appears to be incorrect and shows two different markers, it is unclear what the difference between the two is and is likely a bug.

Bugs also appear as early as the registration stage, with difficulty creating an account and incorrect verification codes being sent to the mobile number used to register. Multiple verification codes were sent when only one is requested, likely a bug in the back end of the application.

Error messages show multiple times for an error which occurs only once i.e. unnecessary repeating errors. This seems quite a simple and obvious bug that has not been addressed and creates an untrustworthy impression of bSafe.

3. Paywalls

While bSafe is technically free, it does contain in app purchases. These purchases are subscription based and are required to use essentially all of the features. When prompted to purchase this subscription again the pop up is overworded in order to promote their features rather than keeping the text to a minimum

Conclusion of bSafe Analysis

While bSafe attempts to fulfil similar needs, the approach overall appears to be more financially motivated than safety based with little thought appearing to be put into the application itself. This is clearly evident from the numerous bugs and pushing of paid features. None of the safety features were available for evaluation even temporarily without first paying and the experience is negative.

The core-values discussed previously prevent of heading in a direction similar to bSafe, with the unique distinction of bSafe and SafeWay being Safeway's core-values and focus on providing a system not intended for profit but solely for safety.

2.4.2 Life360

1. Panic button design

Life360 is another similar application, while albeit better than bSafe, still is not perfect. This app is not as convoluted as bSafe but lacks ease of access to the panic button, which is not located on the home screen. If implemented similarly in the SafeWay application, this would violate core-value number 2, keeping the clicks to completion to a minimum.

2. Paywall

Similar to bSafe, Life360 requires a paid subscription to access many of its features, again moving the application towards a business rather than a fully-fledged safety application for which the primary goal is not profit. Life360 has a higher cost than bSafe.

3. UI design

The UI design of Life360 is effectively implemented in the sense that there were no unexplained or ambiguous symbols or buttons. For the most part the buttons and sections were self-explanatory and to an extent embodied core-value number one incorporating a minimalistic UI and core-value number four of ensuring the application is accessible to all.

The use of bright and deep colours throughout the app is one characteristic SafeWay avoids, as again they distract from similarly coloured features such as the panic button.

Conclusion of Life360 Analysis

Life360 appears to be one of the most polished apps in the personal safety sector of the Apple app store. The app overall is implemented effectively and while there is nothing objectively wrong with it there are features and choices which SafeWay would implement differently in order to keep in line with its core values.

SafeWay differentiates itself by providing a 100% free experience which does not rely on monthly payments or subscriptions. Given the nature of the application SafeWay finds these requests for money to access basic safety features to be exploitive and discriminatory of lower income individuals concerned for family and loved ones.

Chapter 3 Design

3.1 Introduction

Chapter 3 describes an overview of the architecture of SafeWay as well as the database designs and the choices made regarding both UI and UX of SafeWay. The high level overview of the back end services serving SafeWay are also discussed as well as a reasoning behind the choices.

3.2 GitHub

GitHub is used for version controlling of the application, committing regularly and branching when necessary to avoid corruption of the main branch. Version controlling is straight forward, while peer reviewing of the code would be ideal, however it is not feasible for this project and thus is not implemented.

3.3 Firebase

SafeWay relies on Firebase, a back end service based on Googles cloud services, which is generally aimed at mobile and web applications. Firebase provides the back end service for SafeWay such as: handling user authentication, and providing the two databases used. Firebase is selected due to its well established brand and tiered pricing scheme, which is free for low usages.

The Firebase console is an online web application used to manage and implement services used by an application. The Firebase console is home to all the configurations and settings of SafeWay.

The Firebase console also is home to the analytics of SafeWay, which shows usage statistics of the back end services such as the amount of data currently and historically stored on the databases as well as the queries that are being performed and the quantity of queries.

Firebase also facilitates pre-release distribution of applications. However, an Apple developer account must be purchased for €100 in order to avail of these features which does not appear feasible for this project.

3.4 Firebase Authentication

Firebase authentication is a service provided by Firebase. It maintains a secure log of all users, using their email and password combination to authenticate. This email and password is accompanied by a unique user ID, to facilitate integration with other necessary back end services such as databases.

The Firebase authentication spark plan is used in SafeWay, allowing for 3,000 authentications per day, a limit which is sufficient for the foreseeable development of SafeWay.

SMS multifactor authentication is a feature which ideally would have been incorporated into SafeWay in order to add an additional layer of account security and prevent unwanted access. This service works by sending a text message to a mobile phone number provided with the users account on registration. This text contains a pin, meaning the person trying to access the account would need to have both the phone as well as the email and password of the account holder. Unfortunately this service requires an upgraded Firebase account which is not feasible to obtain for this project.

3.5 Database Architecture and Design

3.5.1 Firestore

Firestore is the main database used by SafeWay, it is also provided by Firebase and can be observed and navigated using the Firebase console online. Firestore provides a document oriented database, known as NoSQL that is intuitive to navigate. Ease of implementation is one of the reasons Firestore is chosen, as well as the future proofing NoSQL databases provide, being more cost effective[3] and efficient than traditional relational databases. This allows for expansion to handle large amounts of data which relational databases are reported to struggle with[4].[5]

This document based system is organised into **collections.** These collections are made up of **documents,** each of which contains a unique field and a corresponding value. For example a collection of Users with documents called after the corresponding user names contains the information on that user such as names, user ID etc. It is important to note that these collections and documents can all contain **sub-collections** which is necessary for the database design.

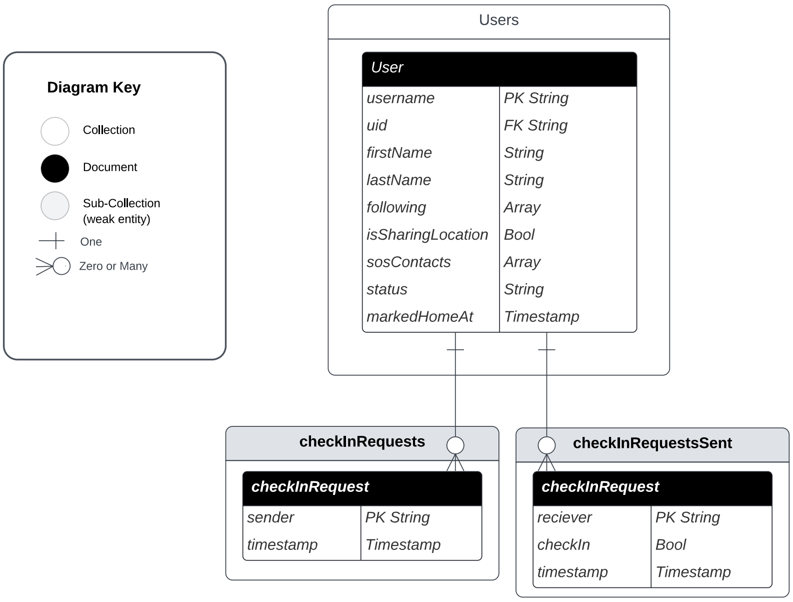


Figure 4: Entity relationship diagram of Firestore DB

Figure 4 shows the entity relationship diagram of the user database within Firestore. The nature of NoSQL databases allows for information to be closely grouped into collections. In this case they have been grouped into the collection “Users”, so all relevant data now stems from within each individual document in the Users database. In other words, the checkInRequests and checkInRequestsSent are grouped into subcollections, which exist within the parent user document in a denormalized structure, whereas in a traditional relational database this structure would be radically different and normalized. These sub-collections would usually be a separate entity referenced by a foreign key[6]. [5]

The subcollections are both weak entities, which cannot exist without a “User” document. “checkInRequestsSent” refers to a collection of documents that represent check-in requests sent by the user in the document containing this sub-collection. The checkInRequests subcollection is a collection of documents which represent the checkInRequests received by the user of the document containing the subcollection. The relationship between these documents is a zero or many to one, in other words the subcollections documents can only be attached to one user but the user can have many documents within these subcollections or none i.e. one user can have zero or N number of checkInRequests documents within their checkInRequestsSent sub-collection.

“PK” and “FK” can be seen in entities lying in figure 6. PK refers to Primary Key, and is the identifier used to find the document when queries are made in the SafeWay app and the documents named after these fields. The FK refers to a Foreign Key, which in this sense may not be a true foreign key as it does not refer to another collection or document. However FK does pertain to the connection between the authorised user and the Firestore database. This “Foreign Key” is used to identify which user is currently authenticated by comparing the UID[5] of the current user and the UID of the documents in the collection.

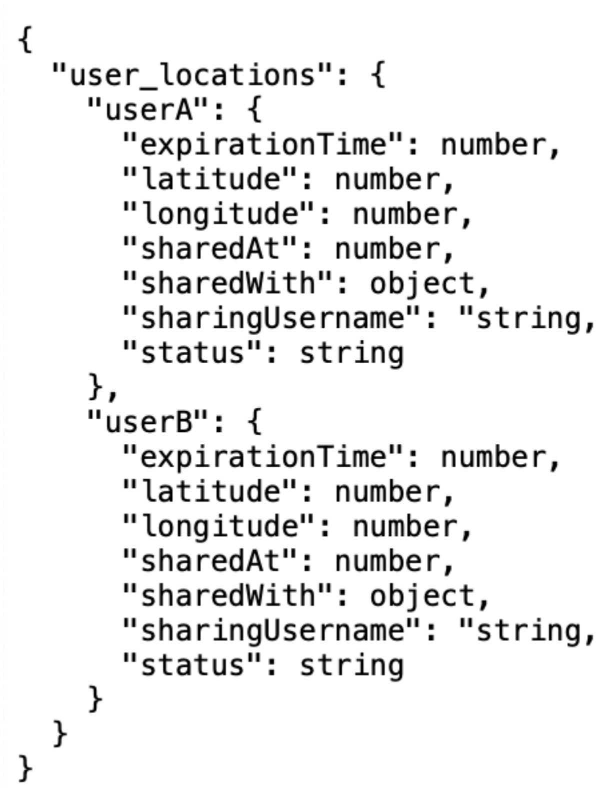
For the most part, the field names are self-explanatory. However the “status” field may require clarification; this string value is set to either “amber” or “red” when a panic button is pressed to symbolise the level the alert.

3.5.2 Firebase Realtime Database

The Firebase Realtime Database is used in SafeWay, solely for the location storage and sharing of users locations. Firebase stores data in a NoSQL key-value method using a JSON structure. This service is primarily chosen for its rapid synchronization capabilities, but it also helps with future proofing in the event that SafeWay broadens into the android sector, i.e. it allows iOS and android to share the same Realtime database, rather than having to create a new one specifically for the Android platform [7].

One of the key features is real-time updating, with Firebase claiming an update to the database is shared to all connected users “within milliseconds”[7]. This is the reason for the implementation of two databases into SafeWay. While it does marginally increase the complexity , it is deemed appropriate as the real time location sharing when a panic button is pressed will require up to date information for all parties involved to ensure maximum utility from these features.

[5] UID refers to user id, which is uniquely generated and assigned to accounts on registration.



```
{
  "user_locations": {
    "userA": {
      "expirationTime": number,
      "latitude": number,
      "longitude": number,
      "sharedAt": number,
      "sharedWith": object,
      "sharingUsername": "string,
      "status": string
    },
    "userB": {
      "expirationTime": number,
      "latitude": number,
      "longitude": number,
      "sharedAt": number,
      "sharedWith": object,
      "sharingUsername": "string,
      "status": string
    }
  }
}
```

Figure 5: JSON structure of Firebase Realtime Database

Figure 7 shows the general structure for the Realtime database. This database is minimalistic and best represented in its JSON structure as opposed to graphing, given the simplicity of the design. This simplicity ensures all non-location sharing/panic based information that needs to be accessed goes through the Firestore database, leaving the Realtime database uncluttered and used only when needed for optimal efficiency. [6]

3.5.3 GeoFire

GeoFire is a library chosen for the location services of SafeWay. GeoFire helps with the storage of location data in the Firestore Realtime database by storing relative information in a key value-manner. GeoFire is open source and lightweight, meaning it is unlikely to slow down or lead to cumbersome real-time location sharing, which is crucial for a working and useful functionality.

3.6 Interface Design

Interface design is a crucial part of any application. However, it is especially important for this application, which is intended for use in a highly stressed environment where the user is unlikely to have the ability to clearly see the operating screen. The need for a user friendly interface that requires minimal input is incorporated from project initiation and is at the forefront of every decision during the development of the application.

[6] "number" is used rather than a more specific number type e.g. int or float as this is how it appears in the Realtime database.

3.6.1 Interface Design Tools

Apple provides two main approaches to interface design and it is important to select and use the correct choice for the developments scenario. The following summarises the tools available:

1) Apples storyboard provides a more visual interaction with the interface design, showing the developer how each view will look and a tree of how each view is connected. Storyboard allows setting of details such as interface object properties via a graphical interface.

2) SwiftUI framework provides classes and other necessary objects for development and manipulation of UI elements in a more programmatic approach with less visualisation.

The storyboard approach seems the right choice, as it allows for visual representations of what is actually being developed and makes it instantly obvious whether a UI element fits in with the core objectives that the interface is to achieve. This storyboard is used in conjunction with UIKit, a graphical interface framework which for example is used to manually trigger segues[7] or a change of view needing a more detailed approach such as meeting certain criteria before a segue would begin.

3.6.2 Colour Palette

The colours used within SafeWay are carefully selected to align with the core-values, more specifically value number one, keeping UI as minimalistic as possible.

The key colours used within SafeWay are a palette of blue, grey, white and black, with the exception of the panic buttons.

Blue

Blue is used in buttons having low importance, i.e. buttons which are not likely to be needed in high pressure situations or do not need to stand out, for instance a button which will follow or unfollow a user. These buttons, while not necessarily the "important" buttons, frequently appear throughout almost all views in the application.

[7] A segue is the transition that takes place between two views of the application. A segue must be triggered by an action and can define how the view being called will appear on the screen.

Much research on the effects of the colour blue and blue light have been conducted. It has been found blue light increases alertness in viewers, and enhances tasks which require alert attention. This is the perfect fit for a safety application used in a high pressure situation, be it someone receiving a panic message or sending one, they need to be alert and focused. [8]–[10]

Blue is also associated with trustworthiness and quality e.g. blue ribbons in a competition [11]. This is something which would be vital in obtaining and building a user base and capturing early adopters who may initially be hesitant in trusting a new and relatively unheard of application.

Grey

Grey is used minimally in the application as it has been found to invoke negative emotions. While it is best to avoid colours which induce negative feelings, it is deemed necessary for subtle bordering and background colours which needed to be understated or in some cases disabled. [12]

White

White is used for the background of the application. White is viewed as pure[10] and more importantly provides a clean environment or "container" for all the application components to reside within, helping to keep SafeWay legible and uncluttered. The "bSafe" app which is discussed in the analysis portion is a prime example of why too much colour is a negative, their application appears dark cluttered and unappealing to use.

Black

Black is used for the text, and labels which are not user interactable. Such a basic and staple colour needed little review into the theory behind it and is the obvious choice with a white backdrop.

3.7 UI Layouts

Initial designs are drafted using Figma, an online modelling tool. The main home page is sketched in order to create a foundation with which the app would build off and thematically match. Each iteration of a new view that stems from the home page uses the same general design theme.

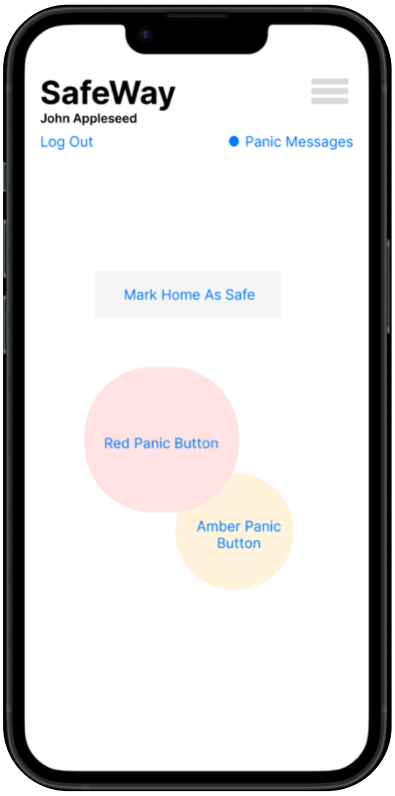


Figure 8A: Figma of the home page

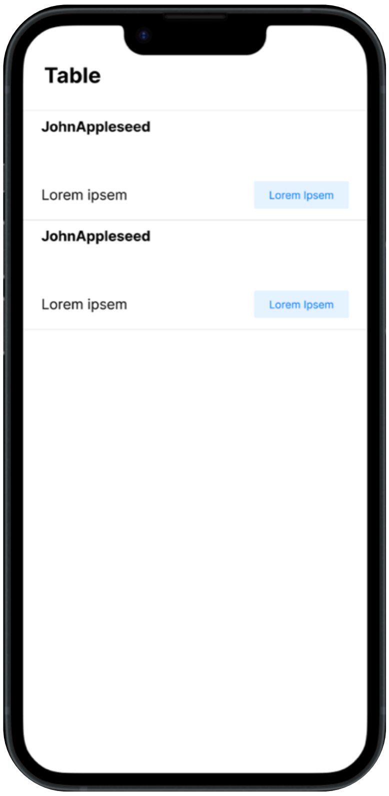


Figure 8B: Figma of generic table and cells

Figure 8A shows the original design theme which SafeWay is built around. It depicts a home screen of the user "John Appleseed" as well as the red and amber panic buttons and the Mark-Home-As-Safe button. The only colours used are specifically from the palette selected earlier, the exceptions being the panic buttons, intentionally done to make the buttons stand out and be clear in any situation.

Observe the use of blue text in sections that are enabled for user interaction such as log out or panic messages in accordance with the palette and colour theory.

This sketch is intended as a foundation for the application, not a hard objective of what the application must look like. As such, the application is expected to evolve throughout development.

Figure 8B shows an example of a table view for the application and any pages which incorporate a table view resemble this template. Again it is expected that each iteration throughout the app would vary slightly based on the use case and the specific needs of the table and cells.

Both figures adhere to the core-values of minimalistic UI. The key and most used features on the home page with little else reduces the "clicks to completion". This allows for panic buttons and the mark home safe button to be used in one click once the app is open, this is the lowest possible interaction to solve a use case in the application and thus is used for the highest priority services.

3.8 Designing the Panic Buttons

While the visual and UI aspects of the panic buttons are addressed and visualised using figure 8A, the UX[8] functionality must be designed.

Initially, the question is should the alarm be silent or audible, with two trains of thought. The first being an audible notification, as seen with car alarms it attracts attention and can potentially scare-off or deter any possible assailants. However, this is used in a case where the only person in the situation is the criminal who intends to break into the car, with very little chance for injury to an innocent third party unless they intentionally choose to get involved. In the case of a personal panic alarm the person in destress is likely to be in immediate danger. This danger is likely to escalate with the activation of a siren or alarm. In a sense, this is the same as the distressed individual announcing to the potential assailant they see them as a threat or danger and are looking for an escape.

Audible alarms do have the advantage of encouraging bystander intervention. However, this would only be in the case of the incident occurring in a place populated with others who are in a position to help. Even then, those in a position to help may chose not to do so and the point becomes moot.

Little data can be found on the efficacy of audible versus silent alarms[13]. As it stands the silent alarm appears the safer option, once a usable product is developed user feedback can be taken into account to redesign and implement some audio signal if requested. With the silent alarm no noise is produced with the idea being the assailant is unaware help is on the way and the user can stall or attempt to deescalate in the meantime.

[8] UX refers to "User Experience"

Chapter 4 Implementation

4.1 Introduction

Chapter 4 discusses the technical challenges faced by SafeWay as well as the major bugs as well as any intricate or code which is repeatedly used and deemed important to the functionality of SafeWay.

The implementation of SafeWay ran smoothly for the most part. While there is a learning curve for Swift it is overall implemented effectively, with minimal amounts of Objective-C being present in the application code, only where necessary.

4.2 Completion Handlers and Database Calls

Completion handlers are a vital mechanism present throughout the development process. Many asynchronous calls are used throughout the Safeway application. This is expected as the application is so close-knit with the backend i.e. almost all functions need to interact or manipulate data from either the Firestore or Realtime database, be it an actual function such as checking-in with a user or simply showing the name of the user logged in. This mechanism allows for the database call to occur and return, the code which relied on the data returned from this query would then reside within the completion handler. This completion handler would not run until the database query completed, preventing crashes from trying to read or manipulate data which has not yet returned. This mechanism is seen in every instance of a database call within the SafeWay application.

The database querying is provided by the Firebase SDK. The simplest query provided is to find a document by searching for the document name. This is one of the reasons the database design employs the username for document names as opposed to user ID's. While it is possible to find a document by using the document fields i.e. "return the document where the field UID is equal to XYZ123", this reduces the readability of the code, and since querying of the database is so common queries were kept as simple as possible.

4.2.1 Snapshots

A snapshot is a data item that has been retrieved from the database at a moment in time. It captures how the database appeared when the snapshot is obtained. As previously discussed, snapshots are captured asynchronously This allows the application to continue loading while the data is being retrieved from Firebase cloud servers. Data retrieval time is unpredictable and may bottle-neck the application speeds if the database query is run on the main thread. When the data has been retrieved in the background it is passed to the completion handler, which will then perform a task using the retrieved data.

Document Snapshots

A document snapshot contains one single document from the database and will never return more than one document, as each document in the database will have a unique ID. Retrieving a document snapshot will return both metadata on the document as well as the actual data contained within the document. This is represented as a dictionary with the key value being the field name and the value corresponding to the key being the data contained in that field.

Query snapshots

A query snapshot may return more than one document snapshot. It returns all documents where a condition is met, i.e. all documents where the users last name is "Murphy", as this field is not unique it may return an array of documents. However if no documents match the criteria an empty array is returned. The multiple document snapshots returned in a query snapshot capture both the data and the metadata, i.e. the document ID, associated with each document.

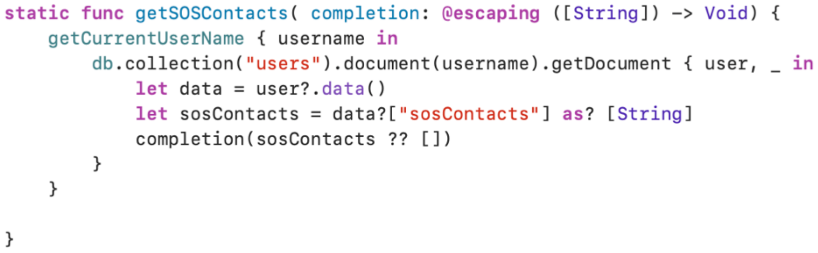
4.3 Issues with Firebase Auth

FirebaseAuth is a part of the Firebase SDK concerned with the authentication and registration of users. While it works for the most part, encrypting passwords, registering users in the authentication database in firebases backend and providing useful functions providing information on the currently authenticated user, it does have its flaws.

FirebaseAuth contains a function `Auth.auth().currentUser.` This function returns information on the currently authenticated user, for example `Auth.auth().currentUser.uid` returns the user ID of the authenticated user. This however proves difficult to use when querying the database as documents are named after usernames, not user id's (this is intentional as it provided easier querying in general). For example if a document is to be found for johnappleseed, in a scenario using UID it is found by finding all documents where UID equals XYZ123. This is not overly complex but is less readable than “find the document called ‘johnappleseed’”

This leads onto the issue, `Auth.auth().currentUser.displayName` could also be called to return the display name (username) of the currently authenticated user, which would have solved all the problems. This function (`Auth.auth().currentUser.displayName`) however is asynchronous. This is where the issue occurs, as the functions using this call to retrieve the username complete before the data is returned from the backend, which results in crashes. For unknown reason this issue does not occur when retrieving the UID.

The initial fix is to place this call for the display name within a function and include a completion handler. This is tried and still does not rectify the issue, eventually a new function is created called `getCurrentUserName()`. This function employs a completion handler which works as expected and fixes the issue. Now the query based on the UID of the user is only used in one function which is called and nested within any other functions or code that required the authenticated users username.



```
static func getSOSContacts( completion: @escaping ([String]) -> Void) {
    getCurrentUserName { username in
        db.collection("users").document(username).getDocument { user, _ in
            let data = user?.data()
            let sosContacts = data?["sosContacts"] as? [String]
            completion(sosContacts ?? [])
        }
    }

}
```

Figure 9: Example of getCurrentUserName nested in a function

An example of the getCurrentUserName function in action is seen in figure 9. The function is nested within the getSOSContacts function which, if the Firebase auth package worked correctly, a simple call and declaration of `username = Auth.auth().currentUser.displayName` would have sufficed. This is not possible and instead we see the solution of placing code reliant on the authenticated users username within the completion handler of the getCurrentUserName function. This completion handler is only run after the username is retrieved and thus solved the crashing bug which has been appearing consistently throughout the application.

While it may seem like a rudimentary problem and the underlying code is overly complex, it is an unexpected issue that should not occur. Thus it is difficult to debug find and create a solution to reduce any repeat code from appearing within the application

4.4 Utilities

It is important to implement reusable functions where possible in order to reduce code repetition. Any function which is deemed reusable is added to the "Utilities" class as a static function. This is frequently used for database calls, as the code necessary to query the Firestore database is bulky, repetitive, and difficult to read. As a result, these "utilities" or functions with multiple uses throughout the app improve legibility and coding efficiency.

An example of a commonly used Utilities function is the function. Note many functions in the app initially had a "forUser" argument, to increase the adaptability to different circumstances. For example if you call getFollowersList(forUser: "JohnAppleseed") it allows you to retrieve followers of whoever necessary by using their username as an argument depending on the situation. This in fact ended up being a case of over-engineering, and in most cases the functions only concern the currently logged in user, and as such the getCurrentUsername function discussed in the previous section is nested within leaving the argument redundant.

4.5 Location sharing

Implementing the location sharing aspects of SafeWay proves to be one of the more difficult portions of the application development. The GeoFire package is not straightforward and relies on complicated delegates and implementing a location services manager. This class, named LocationServicesManager, is crucial and implements functions to begin sharing and cease sharing a user's location, which is a core feature for SafeWay.

LocationServiceManager.startSharingLocation() initially took a parameter called sharedWith. This states what user to share the location with, e.g. `startSharingLocation(withUser:"johnAppleseed")`. This function is run on the press of a panic button, creating an object containing the information needed by the SOS contacts receiving the panic message and for writing the information to the Realtime database. While this initially worked, it is deemed more efficient to nest the getSOSContacts() within the function and the array of SOS contacts added in one iteration, rather than calling the function for each user as is initially the pattern.

4.6 Panic Buttons and Messages

It is important to discuss the main feature of SafeWay and address how the functionality is achieved and the steps undergone to achieve the end product.

The panic buttons are implemented using the silent alarm methodology as discussed before and work in conjunction with GeoFire. When the button is pressed, a function located in Utilities is ran in order to retrieve the SOS contacts of the currently authenticated users. These users will be listed under the "sharedWith" key in the object that GeoFire stores in the Realtime database. If the function finds the user has no SOS contacts a message is sent to the user informing them of this fact and thus the function ends with no continuation.

If the user has SOS contacts, the next check is to determine if the user is already sharing their location. In the case that the user is sharing their location it means the button is already activated and the user is trying to toggle it off and stop the location sharing process.

In the case the user is not sharing their location, the GeoFire Object containing the necessary information and location is sent to the Realtime database and updated as the location moves.

A snapshot listener is also initialised when the app is activated. This snapshot listener is used for ensuring the correct panic button is shown as in-use by observing the "status" field in the Firestore document of the authenticated user. Bugs initially occurred when a user logged-in while sharing their location and the button appeared inactive. This snapshot listener observes the status of the panic buttons and the status in order to change the colours of the panic buttons accordingly and solve this bug.

The logout button appears inactive for a moment after a panic button is toggled. This is due to an issue which occurs when the panic button is toggled and the user logs out before the action could be completed. It leads to unfinished objects in the database and crashing when logging in and trying to use the panic button. The fix identified is to disable the logout button while the database is being written to. The period is so brief that it is not foreseen to cause any adverse or unwanted UX issues.

During the actual implementation of the panic button a more simplistic rectangular button is employed with bevelled corners for a more elementary design. The choice is not based on programming constraints but rather external feedback of preferences from individuals shown the app. External feedback is discussed in greater detail in the evaluation section of the report.

4.7 Following a User

One of the final bugs was spotted in the Firestore database. Documents with what seem like duplicate names were appearing. The same issue is subsequently seen in arrays pertaining to lists of usernames, such as the followers array and the SOScontacts array. After investigation it is observed that these are in fact not duplicate entries, as is predicted to be impossible given document names must be unique. Rather, the second iteration includes a blank space at the end of the username. A simple issue, but a difficult one to identify given the very slight difference. Once the issue was located it is then another task to discover at what point in the flow of the application is this trailing blank space being added to the username and introducing the bug. It is initially hypothesised to be impossible that the blank space is being added when the user is first followed, given the nature of the function. The function would not show a user profile unless the username matched exactly with what is typed in the search bar and the database is queried when the text in the search bar is changed, meaning "johnappleseed" would show the username and the follow user button but "johnappleseed "[9] should not as there is no document of this name. The issue however did originate here, it occurs when the username is recognised by the phone. When the username is typed completely, the username will still be suggested and if the search bar is unselected the suggested username will populate the search bar and add a space to the end of the username. This would cause the "follow user" button to still show and use the username with the space at the end as an argument for the functions using this text.

This is fixed quickly and a trimming of trailing characters is added. Although it is a simple fix, it was a difficult bug to track down and trace bag to the source.

[9] The name plus a blank space at the end

Chapter 5 Evaluation

5.1 Introduction

Chapter 5 discussed the methods of evaluation used to ensure SafeWay is useable and fit for its intended purpose. This is done using a focus group of carefully selected individuals and an online survey sent to a larger group of potential participants

External impartial individuals are shown iterations of the application during the developmental stages. This process is to ensure the resulting program is optimised for fresh users with a neutral standpoint as opposed to the developer who's view of the application may have become distorted due to the close proximity with regards to code and design.

5.2 Focus group

A small focus group of five individuals is allowed to use the SafeWay application. The participants have no involvement in the design or implementation of the code, architecture or any other elements of the processes and thus are deemed relatively impartial. Given the resource and monetary limitations of the project scope it is not possible to conduct a truly random, impartial focus group which is what would have been done in ideal circumstances. The group is deemed sufficiently impartial to provide useful feedback and are informed that their candidness would be of great benefit.

5.2.1 Focus group selection

The focus group consists of a mixture of males and females in their early 20s. This mix of participants is selected given that the majority of users will be in this demographic The size of the group is limited to facilitate detailed feedback and one to one interaction when needed.

The focus group is assembled once a minimum viable product is produced. In other words, the core features of SafeWay are implemented in an unpolished package potentially containing some bugs and suboptimal user experiences. This point is intentionally selected as it is believed a focus group shown the application at this stage can provide two services; the first to observe any bugs or abnormalities which may have been overlooked, the second providing feedback for the preliminary design.

Before the application is shown, the participants are briefed on the outline of SafeWay as well as the primary core-values on which the application is based, i.e. the application should keep people safe, be simple to use, quick to navigate, and accessible to all.

5.2.2 Features Evaluation

The feature analysis is simple, with each participant shown the main use cases of the application. Guidance and interference is kept to a minimum and where intervention is requested or necessary a note is made of the ambiguous nature of the design as this would be considered a deviation from the visualised end product.

Multiple users noted that although a check-in request could be observed within the inbox, unless the inbox was manually opened there was no indicator of the status of the inbox. This is noted and rectified. In the finalised version the status is shown using a circle, when the circle is shaded the inbox contains a request and when the circle is unshaded (i.e. only an outline) the inbox is seen to be empty.

The design of the panic button is central to the aesthetics and operation of the overall application. As a result, a simplified A/B test is employed to evaluate this feature. This involves showing half of the group one version of the panic button, and the other half an alternate version.

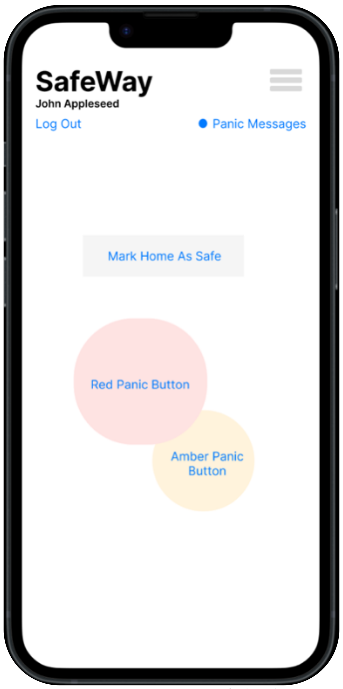


Figure 10A: Figma of the home page

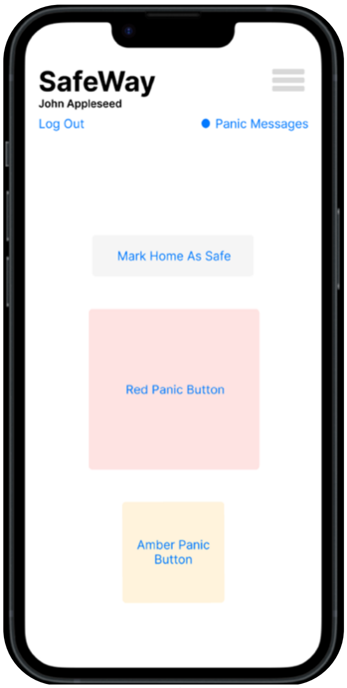


Figure 10B: "Alternate" panic button design

Group A are shown the home screen as designed in the original sketch seen in Figure 10A (reproduced from Figure 8A for comparison), whereas group B are shown an alternate design as seen in Figure 10B. Group A noted accidental pushing of the wrong button due to the overlap and close proximity of the two buttons, and when presented with the alternate design concur it is more accessible and functional. Thus is the change is implemented.

5.2.3 User flows evaluation

Once the key features were tested and discussed, user flows are then explored. Users are placed initially on the home screen of the application and instructed to navigate to a named feature. This is repeated between different views and features until the user flows were exhausted.

These trials indicate that the tabs used to navigate between views were ambiguously named. When participants 1, 3 and 4 attempt to find a user's profile in order to add them as an SOS contact there is uncertainty as to where this would be done. The tab listing the table of users being followed was initially labelled "Check-ins", which to someone not involved in development and no knowledge of the app does not imply this would be the path with which to find a user's profile.

A note of this is again made and the current version of the application has renamed the tab to "Following", a much more appropriate name for the view.

5.3 Survey Evaluations

Once a more complete and rounded product is developed, taking into account the notes from the focus group a process of composing a survey is begun. A cross sectional approach to the study is decided upon. A group of individuals, larger than the focus group is administered via an online method of distribution. A cross sectional study refers to one which captures information at a singular point in time, as opposed to longitudinal which would capture information over an extended period. A cross sectional study seems feasible and appropriate given the scope of the research [14].

5.3.1 Survey design

A total of ten questions are included in the survey, with six of these being open-ended qualitative questions, two qualitative numerical and two qualitative categorical questions.

The survey is designed with brevity in mind to ensure an adequate response rate, with an emphasis on open ended questions to allow for detailed replies on issues which may not have been predicted. It also allows for a survey response dense in information while keeping the survey duration to a minimum.

5.3.2 Survey responses[10]

A response rate of 75% is observed with overall positive results observed, with an average app rating of 4.25 out of five recorded and the user friendliness of the app rated at 4.8 out of five. This shows that the changes implemented following focus group conducted have improved the user experience.

One of the key categorical questions addresses the panic button functionality, more specifically whether it should be a silent or audible alarm. Feedback is evenly distributed, with no clear preference. The participant is then asked to include a rational for their choice. As hypothesised during the design of SafeWay, the schools of thought for each option appeared to be an audible alarm would alert bystanders and a silent alarm would prevent escalation of a situation. A quote from a participant who is for the audible alarm states *“An audible one[panic button] would also alert those nearby not just on the app”*

Conversely a participant advocating for silent alarms notes *“I think that an audible button could alert the person making you uncomfortable and may trigger violence”.*

As a result, it may be appropriate for either longitudinal A/B testing of each case and another cross sectional study in order to observe the experience each side had with their panic button alarm implementation. Due to time constraints this is not feasible to evaluate over a long-term and as a result the silent alarm deemed to be the lower risk option (less risk of aggravation) is implemented in the current version.

One participant raises a concern regarding the amber and red panic buttons, stating *“I think that the user should be more aware as to what constitutes a “red” alert or “amber” alert as to not cause undue panic to those receiving the alert in the case that the situation is, in fact, not threatening or harmful.”* This is a thought provoking response, as the application requires balance between being self-explanatory and not including clutter or excess text. With that in mind, an initial app tour would be beneficial. This brief tutorial would explain each feature and its utility, so as not to clutter the app with detailed instructions, while providing clarification to the processes available. This is

[10] The full results of the survey may be seen in section 1 of the appendix

not implemented in the current version as there were no other comments of similar nature and thus is not a priority in the pipeline of tasks to be completed. However, it may be something to consider if future development of the app is envisaged.

The participants are also questioned on what core feature of the app (panic buttons, check-in or mark-as-home safe) they are most likely to use or find the most utility from.

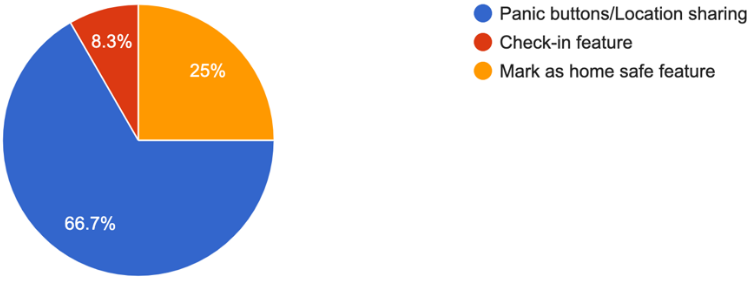


Figure 11: Pie chart of features based on users perception of highest utility

Figure 11 summarises the perceptions of the participants. Unsurprisingly, the panic buttons and location sharing with which is the core focus of the app is number one. As such it is logically allocated the most area in the central part of the application; the home screen.

Following at number two the mark-as-home feature is again centrally located within SafeWay, however to a much subtler degree than the panic button, i.e.in more muted colours with a smaller allocation of size.

Finally the check-in feature receives the least amount of votes. As such this feature is located away from the home screen of SafeWay, to avoid a cluttered environment.

Any privacy concerns the participants may have are addressed in an open ended question. Two key responses stood out, the first stated: *"I would be concerned as to the location sharing. Many apps nowadays require constant access to location and many users may be sceptical that their locations can be accessed by administrative persons at any time."* This is a difficult concern to address as location services are so important to SafeWay. One point to note is, that although location sharing is required to send panic messages, the app is usable with reduced functionality without these permissions enabled by the user. In other words, a user can deny SafeWay the use of location services and still receive panic messages containing the location of other users. The mark home-as-safe and check-in features will also remain fully functional. At the moment the database storing the locations is accessible by only one person, the system administrator of SafeWay, with the password not being shared with anyone else.

Firebase is also compliant with ISO27001, ensuring the Realtime database is GDPR compliant and the appropriate security measures are taken to protect the locations of the users. [15], [16]

The three main scenarios for use of SafeWay, according to the conducted survey, are indicated to be: walking home alone, nights "out" (inferred to mean late nights clubbing or attending pubs), and when meeting someone new. The core-values of SafeWay point the design in a direction where these scenarios can fully utilize the software. There is arguably most room for improvement in the scenario in which a user is meeting a new person. While currently the user can activate an amber alert if they feel nervous or as a precautionary measure, research into this area could prove useful. For example a "fake phone call" feature may provide a clean escape from uncomfortable meetings or dates.

When asked what additional features participants would like to see, one user wrote that they suggest: *"[A] Feature to contact emergency services".* This feature had been included during the initial stages of the application conception, and is revisited once this comment is received. This feature had a few obstacles, the first and most obvious is to simply implement a feature which would create a call to 999. This is feasible but not in the single click of a button (the panic button) and would require multiple points of user input. Something not feasible in a high pressure situation.

Another response addressed the 8 hour time limit on the location sharing once the panic button had been activated. When asked what features they would implement the participant commented: "*Maybe longer location sharing*". This was considered and it was decided that there should be no limit on the location sharing once a panic message has been pressed, and granted that it is not cancelled, the location should continue to be shared. This is changed for the current version and the expiration time is removed from the location services manager.

SafeWay is overall received positively, with constructive feedback implemented where appropriate in a manner that prioritises the utility provided to the end user.

Chapter 6 Conclusion

6.1 Introduction

Chapter 6 provides a reflection on the methodologies used for the engineering of SafeWay. The manner in which the project was approached as well as anything that would have been done differently, given alternate circumstances or if starting the project again.

The application is overall deemed to accomplish the main project goals, while adhering to the core-values set out in the beginning. The preliminary design was refined following user evaluation.

The overall project is conducted in an efficient manner, following a detailed plan rather than a "design-as-you-go" methodology, which can lead to inefficient use of valuable resources and time. The planning and structured agile inspired methodologies allow for an informal structure in order to keep work flowing smoothly and lead to maximisation of time spent coding to outputted code, especially given the nature of software development and the frequency with which unexpected issues or bugs arise.

6.2 Summary of Conduction

As previously discussed, a process loosely based on agile methodology is implemented for the design of SafeWay. For each major element or component, the following iterative structure is informally followed. This methodology is used for the implementation of the application once the generalised system architecture is decided on.

Plan

A plan of what the component needs to accomplish is first outlined and allows for a set goal to work towards. Loose timelines are set in order to keep work on track and provide self-appointed deadlines with which to work towards,

Design

The design is then considered, figure 8A and figure 8B were used as the basis for each iteration. The view is then adjusted in an interactive manner using XCode's storyboard. The storyboard allows for visual manipulation as well as drag and drop of any components such as buttons or labels. This facilitates exploration of alternative designs in an efficient manner, cutting down on the design time.

Development

Next development begins, with the goal of each development iteration being to keep software simple and maximize clarity where possible. Bugs noted in this stage are resolved as they appeared. This helped to simplify the development and reduce a back log of issues.

Testing

The app is tested as best it can be given the scope of the project i.e. it is not possible to have a dedicated team of testers for each feature as they are produced. This testing stage is not the primary quality validation of SafeWay, but is closer to the first line of defence. The main testing occurs during the evaluation stages, which were discussed earlier. It does however prove to be fruitful and uncover issues not seen during the development stages.

Deployment

Deployment is used as a loose term, the code which when deemed to be at a satisfactory level of completion and quality, is then merged into the main branch and a new iteration would once again commence.

This methodology is found to be effective and the informal structure given the nature of the project allowed for a relaxed production strategy while keeping a loose time line for expected completion

6.3 Reflection

While agile work environments are observed and experienced during work placement, the ability to conduct a self-motivated and create a self-directed software development structure by moulding existing methodologies to the specific scenarios is beneficial in the sense of providing insights into what works and what does not work. The methodology behind the development of this project allows for significant amounts of work to be conducted in time constrained situations.

While there are features and improvements which can still be made as discussed in the evaluation section, this is not seen as a downfall or failure of development but attests to the versatility and opportunities of expansion for SafeWay. The core application and concept provides ample room to grow and mould as any forward-thinking project should.

Were the project to begin afresh, one aspect which would likely be conducted in a different manner is the time point at which the survey is conducted. While it is possible to mould the app based on these suggestions and does provide the indication of a successful implementation of the envisioned design, if conducted earlier it may have allowed for weighing of what features should be added first given time constraints.

In an ideal environment the application would have been submitted for listing on the Apple app store, and as a pre-release app for Beta testers, which would have allowed for wider testing feedback. Testing in the current development required an in person setting and the users being shown emulations or manually installing a temporary test version from the machine of the developer. The issue is complicated by the fact that these manual installs become unusable after long periods and were only suitable for short term testing. A larger user base would have provided more insights and comments regarding the design and implementation.

Overall the experience is a positive one, providing knowledge in the full stack development that is required to get an application to a production level of quality. The experience in Swift language development is invaluable and enabled close knit work with database querying and authentication in the appropriate methods. The project enforced the importance of version control and having the opportunity to create a new iteration of SafeWay, an increased number of branches would've been created with more specific branch names.

The research conducted during the course of the project shows a clear use and utility for the application, especially with the college demographic. When asked, many of the focus group users are unfamiliar with any personal safety applications on the market today. SafeWay clearly shows potential for expansion and will likely continue to be built upon as a still ongoing project as an extracurricular activity.

References

[1] CSO and J. Dalton, ‘Recorded Crime incidents by detection status and Garda Region (Under Reservation)’, Nov. 17, 2022. https://data.cso.ie/ (accessed Mar. 07, 2023).

[2] CSO, ‘Recorded Victims 2021 - CSO - Central Statistics Office’, 2022. https://www.cso.ie/en/releasesandpublications/ep/p-rcvo/recordedcrimevictims2021andsuspectedoffenders2020/recordedvictims2021/ (accessed Mar. 07, 2023).

[3] Cloudera, ‘Cloudera Big Data and Hadoop Sessions’, Hyatt Hotel, Denver, CO., Hyatt Hotel, Denver, CO., May 21, 2014.

[4] C. Mohan, *EDBT ’13: Proceedings of the 16th International Conference on Extending Database Technology*. New York, NY, USA: Association for Computing Machinery, 2013.

[5] R. Mason, *NoSQL Databases and Data Modeling Techniques for a Document-oriented NoSQL Database*. 2015. doi: 10.28945/2245.

[6] A. B. Moniruzzaman and S. A. Hossain, ‘NoSQL database: New era of databases for big data analytics - Classification, characteristics and comparison.’, *International Journal of Database Theory and Application*, vol. 6, no. 4, pp. 1–14, 2013.

[7] ‘Firebase Realtime Database’. https://firebase.google.com/docs/database (accessed Apr. 09, 2023).

[8] C. Cajochen *et al.*, ‘High Sensitivity of Human Melatonin, Alertness, Thermoregulation, and Heart Rate to Short Wavelength Light’, *The Journal of Clinical Endocrinology & Metabolism*, vol. 90, no. 3, pp. 1311–1316, Mar. 2005, doi: 10.1210/jc.2004-0957.

[9] S. W. Lockley, E. E. Evans, F. A. J. L. Scheer, G. C. Brainard, C. A. Czeisler, and D. Aeschbach, ‘Short-Wavelength Sensitivity for the Direct Effects of Light on Alertness, Vigilance, and the Waking Electroencephalogram in Humans’, *Sleep*, vol. 29, no. 2, pp. 161–168, Feb. 2006, doi: 10.1093/sleep/29.2.161.

[10] A. J. Elliot, ‘Color and psychological functioning: a review of theoretical and empirical work’, *Front. Psychol.*, vol. 6, Apr. 2015, doi: 10.3389/fpsyg.2015.00368.

[11] L. I. Labrecque and G. R. Milne, ‘Exciting red and competent blue: the importance of color in marketing’, *Journal of the Academy of Marketing Science*, vol. 40, no. 5, pp. 711–727, Sep. 2012, doi: 10.1007/s11747-010-0245-y.

[12] H. R. Carruthers, J. Morris, N. Tarrier, and P. J. Whorwell, ‘The Manchester Color Wheel: development of a novel

way of identifying color choice and its validation in healthy, anxious and depressed individuals', *BMC Medical Research Methodology*, vol. 10, no. 1, p. 12, Feb. 2010, doi: 10.1186/1471-2288-10-12.

[13] C. Perkins, D. Beecher, D. C. Aberg, P. Edwards, and N. Tilley, 'Personal security alarms for the prevention of assaults against healthcare staff', *Crime Science*, vol. 6, no. 1, p. 11, Nov. 2017, doi: 10.1186/s40163-017-0073-1.

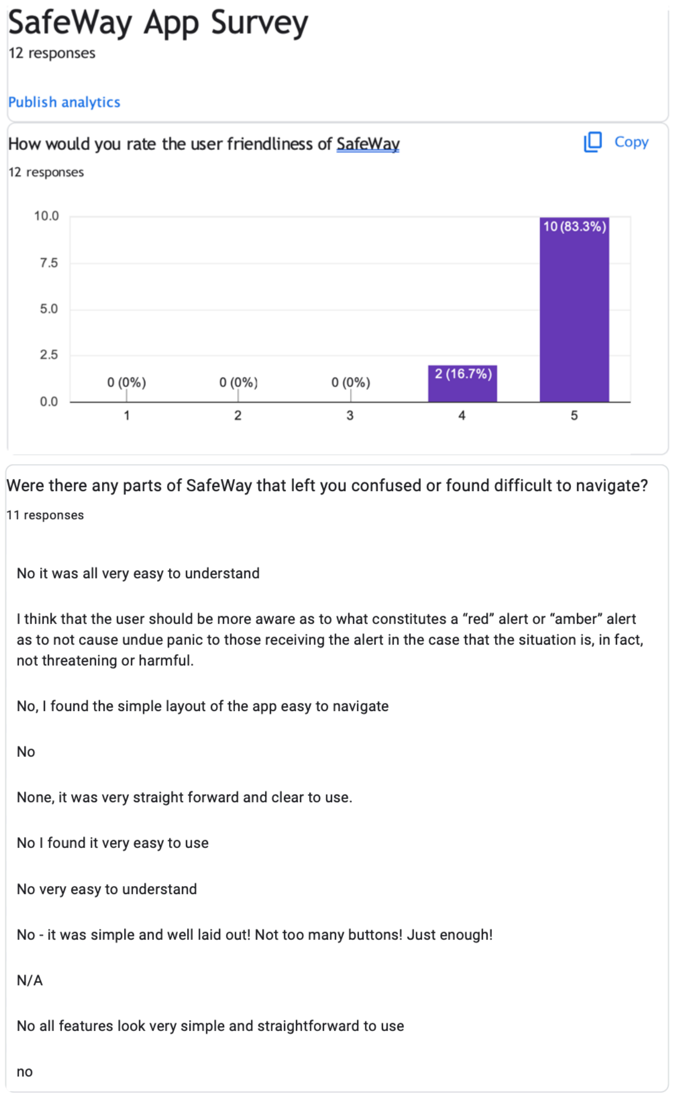
[14] A. Rindfleisch, A. J. Malter, S. Ganesan, and C. Moorman, 'Cross-Sectional versus Longitudinal Survey Research: Concepts, Findings, and Guidelines', *Journal of Marketing Research*, vol. 45, no. 3, pp. 261–279, Jun. 2008, doi: 10.1509/jmkr.45.3.261.

[15] Certification Europe, 'ISO 27001', *Certification Europe*. https://www.certificationeurope.com/iso-certification/iso-27001/ (accessed Apr. 12, 2023).

[16] Firebase, 'Privacy and Security in Firebase', *Firebase*. https://firebase.google.com/support/privacy (accessed Apr. 12, 2023).

Appendix 1

Survey Results



SafeWay App Survey

12 responses

How would you rate the user friendliness of SafeWay

12 responses

Were there any parts of SafeWay that left you confused or found difficult to navigate?

11 responses

No it was all very easy to understand

I think that the user should be more aware as to what constitutes a “red” alert or “amber” alert as to not cause undue panic to those receiving the alert in the case that the situation is, in fact, not threatening or harmful.

No, I found the simple layout of the app easy to navigate

No

None, it was very straight forward and clear to use.

No I found it very easy to use

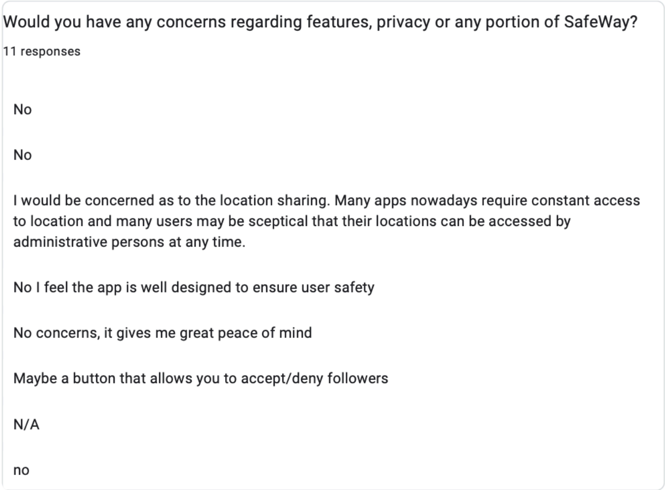
No very easy to understand

No - it was simple and well laid out! Not too many buttons! Just enough!

N/A

No all features look very simple and straightforward to use

no



Would you have any concerns regarding features, privacy or any portion of SafeWay?

11 responses

No

No

I would be concerned as to the location sharing. Many apps nowadays require constant access to location and many users may be sceptical that their locations can be accessed by administrative persons at any time.

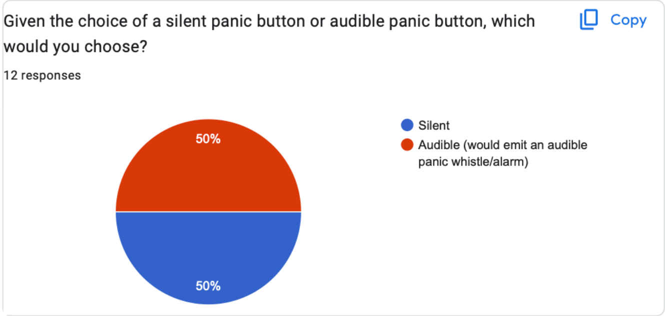
No I feel the app is well designed to ensure user safety

No concerns, it gives me great peace of mind

Maybe a button that allows you to accept/deny followers

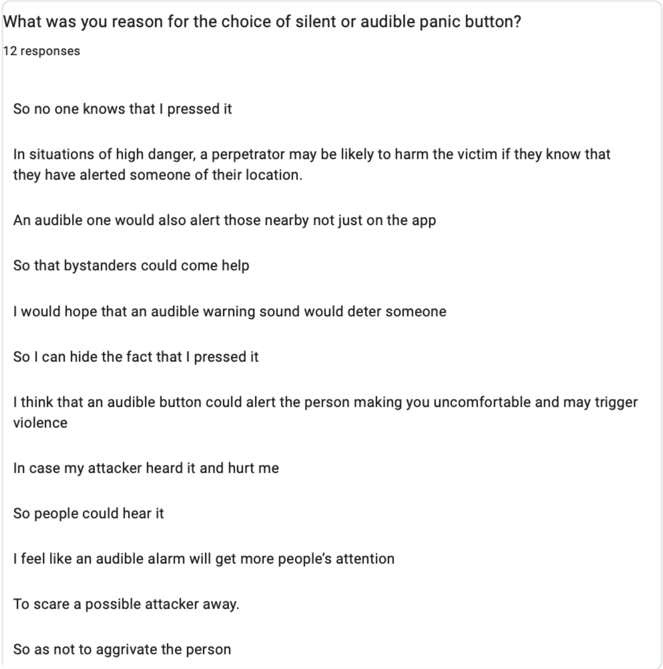
N/A

no



Given the choice of a silent panic button or audible panic button, which would you choose?

12 responses



What was you reason for the choice of silent or audible panic button?

12 responses

So no one knows that I pressed it

In situations of high danger, a perpetrator may be likely to harm the victim if they know that they have alerted someone of their location.

An audible one would also alert those nearby not just on the app

So that bystanders could come help

I would hope that an audible warning sound would deter someone

So I can hide the fact that I pressed it

I think that an audible button could alert the person making you uncomfortable and may trigger violence

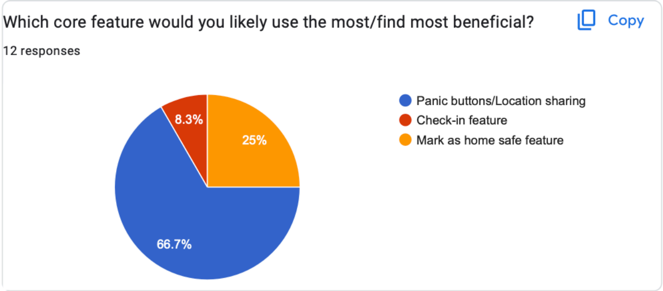
In case my attacker heard it and hurt me

So people could hear it

I feel like an audible alarm will get more people’s attention

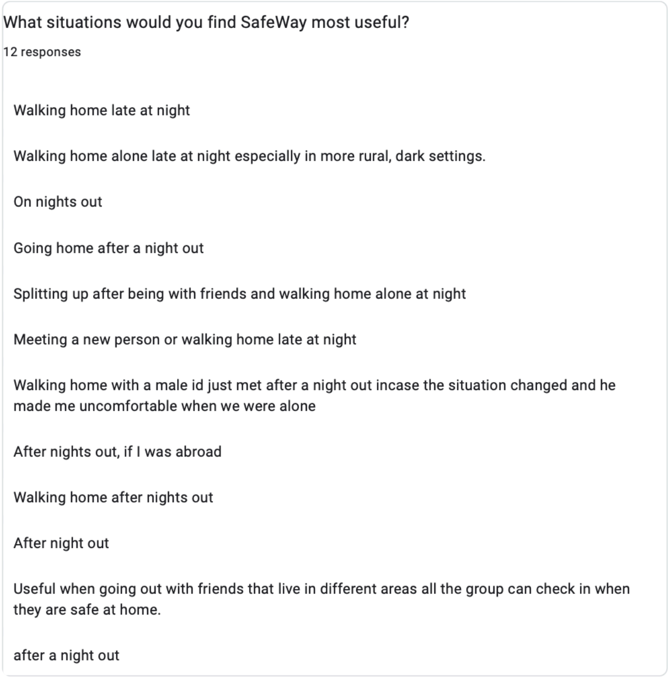
To scare a possible attacker away.

So as not to aggrivate the person



Which core feature would you likely use the most/find most beneficial?

12 responses



What situations would you find SafeWay most useful?

12 responses

Walking home late at night

Walking home alone late at night especially in more rural, dark settings.

On nights out

Going home after a night out

Splitting up after being with friends and walking home alone at night

Meeting a new person or walking home late at night

Walking home with a male id just met after a night out incase the situation changed and he made me uncomfortable when we were alone

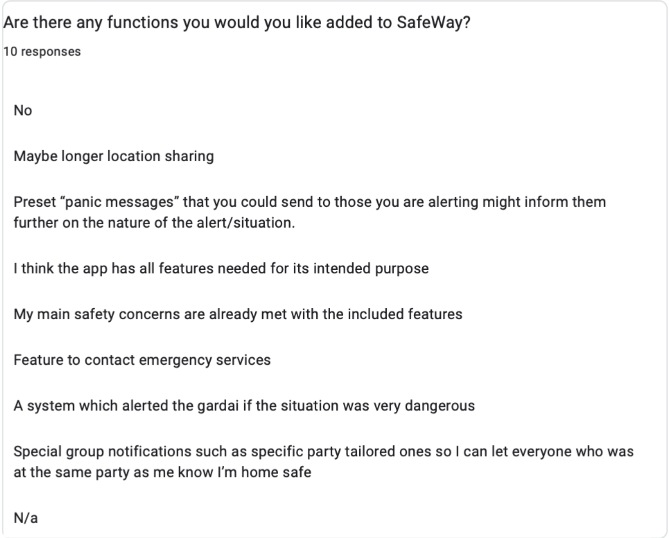
After nights out, if I was abroad

Walking home after nights out

After night out

Useful when going out with friends that live in different areas all the group can check in when they are safe at home.

after a night out



Are there any functions you would you like added to SafeWay?

10 responses

No

Maybe longer location sharing

Preset “panic messages” that you could send to those you are alerting might inform them further on the nature of the alert/situation.

I think the app has all features needed for its intended purpose

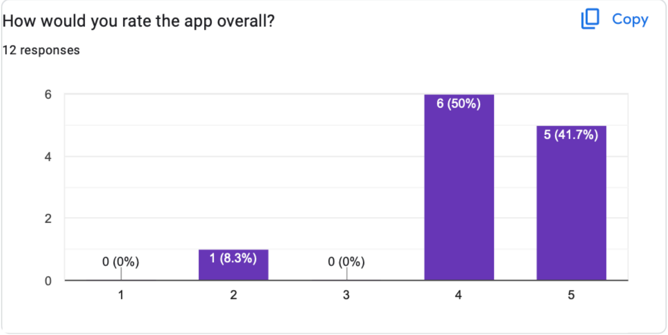
My main safety concerns are already met with the included features

Feature to contact emergency services

A system which alerted the gardai if the situation was very dangerous

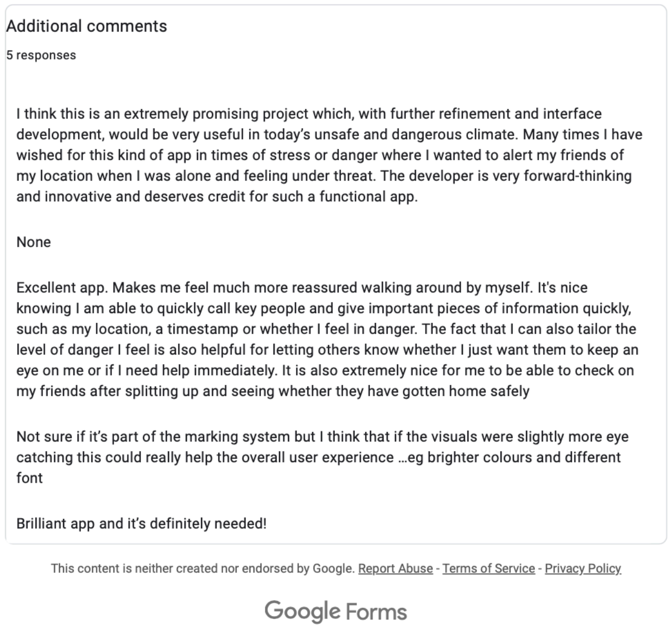
Special group notifications such as specific party tailored ones so I can let everyone who was at the same party as me know I’m home safe

N/a



How would you rate the app overall?

12 responses



Additional comments

5 responses

I think this is an extremely promising project which, with further refinement and interface development, would be very useful in today's unsafe and dangerous climate. Many times I have wished for this kind of app in times of stress or danger where I wanted to alert my friends of my location when I was alone and feeling under threat. The developer is very forward-thinking and innovative and deserves credit for such a functional app.

None

Excellent app. Makes me feel much more reassured walking around by myself. It's nice knowing I am able to quickly call key people and give important pieces of information quickly, such as my location, a timestamp or whether I feel in danger. The fact that I can also tailor the level of danger I feel is also helpful for letting others know whether I just want them to keep an eye on me or if I need help immediately. It is also extremely nice for me to be able to check on my friends after splitting up and seeing whether they have gotten home safely

Not sure if it's part of the marking system but I think that if the visuals were slightly more eye catching this could really help the overall user experience …eg brighter colours and different font

Brilliant app and it's definitely needed!