

GNG2101
Design Project User and Product Manual

Personal Safety Application

Submitted by:

Safety Squad A15

Jonathan Florus, 300178425

Laura Godfrey, 300082954

Mashal Joyaa, 300082650

Bonnie Lin, 300162118

Quinn Murnaghan, 300168522

Dec 5, 2021

University of Ottawa

Table of Contents

Introduction	7
Overview	8
Cautions & Warnings	9
This application is not a replacement for emergency services. It should only be used in non-emergency situations where calling 911 would be unnecessary. This app will not call 911 for its users, but instead alert the users emergency contacts. The emergency contacts of the user are responsible for any action taken after receiving the text from this application. The emergency contacts are responsible for being alert regarding receiving the texts from the application. Safety Squad is not responsible for any missed text messages that were sent properly. Safety Squad is also not responsible for the texts not sending properly as the application's messaging system may also fail in the event that there is a problem with the user's phone. If there is an emergency, do not rely on the 'Personal Safety App' and contact emergency services immediately.	9
The 'Personal Safety App' is relatively safe to use. It doesn't rely on any server to run and instead relies completely on the user's phone software to run properly. This means that the user's data security (such as phone numbers, names, and emails) will then rely on the phone's ability to encrypt and protect data.	9
Getting started	9
Set-up Considerations	10
User Access Considerations	12
Accessing the System	13
System Organization & Navigation	13
Exiting the System	14
Using the System	14
Check-In	14
Failing to check-in	16
4.2 Emergency Messaging	16
Troubleshooting & Support	18
Error Messages or Behaviors	18
Special Considerations	20
Maintenance	20
Support	21
Product Documentation	21
BOM (Bill of Materials)	21
Equipment List	22

6.2.1 Software	22
Subsystems Instructions	23
Check-In Subsystem	23
Notification Subsystem	24
Messaging Subsystem	24
Activity Subsystem	24
Turn-Off Subsystem	26
Testing & Validation	27
Conclusions and Recommendations for Future Work	28
APPENDIX I: Design Files	29

List of Figures

Figure 2.1: Main page of the application with buttons for the check-in, check-in set up, and emergency message	8
Figure 3.1: Block diagram summarizing the walkthrough for the use of the personal safety application	10
Figure 3.1.1: UI of the App with the Check-in Setup Icon Circled	11
Figure 3.1.2: Visualization of the 3 steps to turn on the check-in feature	12
Figure 4.1.1: Sample notification for the Safety Squad App on an Android device indicating a check-in in 30 minutes	14
Figure 4.1.2: Visualization of the steps to check in with the addition of red circles to indicate the buttons mentioned	15
Figure 4.1.2: Message in the form of SMS communication sent to the user's contacts when the user has checked in	16
Figure 4.1.1.1: Message in the form of SMS communication sent to the user's contacts if the user has not checked in	16
Figure 4.2.1: Visualization of the process of using the emergency messaging function	18
Figure 5.1.1: Error message triggered when the user types in an unaccepted input into the text field and then presses confirm	19
Figure 5.1.2: Error message triggered when the user leaves the text field blank and then presses confirm	20
Figure 6.2.1.1: iOS emulator on Xcode displaying an iPhone 11	22
Figure 6.3.4: Sample display of timed activity feature subsystem	25

List of Tables

Table 1: Acronyms	6
Table 6.1.1: Bill of Materials used in the Development of the ‘Personal Safety App’	21
Table 6.2.1: List of Equipment used in the Development of the ‘Personal Safety App’	22
Table 8.1: Referenced Documents	29

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
API	Application Programming Interface
App	Application
iOS	iPhone Operating System
SMS	Short Message Service
UI	User Interface

1 Introduction

This User and Product Manual (UPM) provides the information necessary for elderly and disabled individuals to effectively use the Safety Squad Application and for prototype documentation. The Safety Squad Application was made for an elderly client requesting a personal safety application that they can use to confirm their well-being on a regular basis. From gathering the client needs and taking note of the client feedback, the final prototype consists of daily check-in notifications, alerts sent to the user's contacts during emergencies, a turn-off function, and written confirmation to check-in.

Within this document, a summary of the design problem and the final prototype of the personal safety mobile application created as a solution to the problem is shown. Preliminary required knowledge is provided, such as the conventions of the application, as well as its warnings. Moreover, the report highlights the process of setting up the application and provides a step-by-step description on how to use the product. Additionally, the report also possesses information on how to solve technological issues that may arise while using the application.

This document is intended to be used by the target audience of elderly individuals and individuals with disabilities seeking to incorporate an additional sense of safety in their lifestyles. The report is also intended to be reviewed by future groups that may want to continue the work of this project.

As for security and privacy considerations, the application is considered reasonably safe as it does not work using a server, it runs using the phone itself; however, the product could possess some safety concerns because it relies solely on the phone's encryption to store and protect data. The level of security and privacy is thus not determined by the application, but by the device used by the user.

2 Overview

The client is in need of an application capable of sending emergency messages in the event that they lose the physical ability to do so themselves. The application is not meant to replace a call to 911, rather it is meant to work as a way for friends and family members to know to check up on the individual at risk in the event that they are in an accident.

More specifically, the client requested an application that sends text messages to an emergency in the event that the user has not checked into the application by a specific time. The check-in function needed to work via a typed phrase. This way, the client would need to have the cognitive capacity to actively recall the phrase and the physical dexterity to type it. This is done as a mechanism to protect them from certain conditions where the client may feel fine, but early warning signs of an impending accident are a loss of dexterity or active recall.

The client came to the group because other applications would only send emails to the emergency contacts. This was not a viable solution for them because they felt that their contacts would not be able to check their emails in a timely fashion in the event of an emergency. Moreover, the client was able to find applications that fit their needs, but they were only available in the US, or were not compatible with her device. The Safety Squad application will be available in Canada, both on iOS and Android to accommodate a variety of users. Additionally, the client noted that some applications had too many additional features that were not necessary, such as displaying different quotes daily. The product focuses on simply ensuring the safety of the user, while also emphasizing minimalism in order to avoid confusion while using the application.

The main page of the final prototype of the application is shown in Figure 2.1. It shows the main functions of the application.

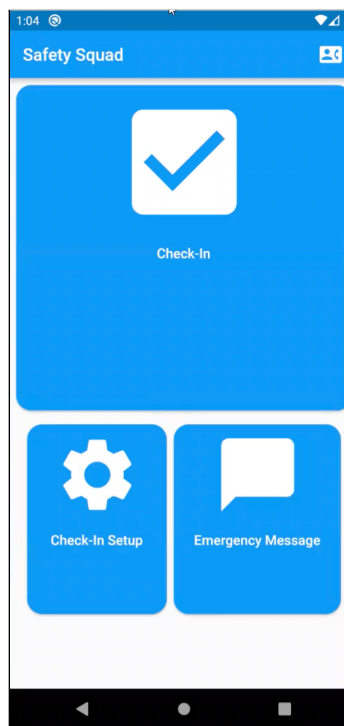


Figure 2.1: Main page of the application with buttons for the check-in, check-in set up, and emergency message

The major functions of the app are the check-in function that ensures the user is safe by making them input a phrase; in the event they fail to do so, a message is sent to an emergency contact. The application is built on a language called flutter and works both on iOS and android.

2.1 Cautions & Warnings

This application is not a replacement for emergency services. It should only be used in non-emergency situations where calling 911 would be unnecessary. This app will not call 911 for its users, but instead alert the users emergency contacts. The emergency contacts of the user are responsible for any action taken after receiving the text from this application. The emergency contacts are responsible for being alert regarding receiving the texts from the application. Safety Squad is not responsible for any missed text messages that were sent properly. Safety Squad is also not responsible for the texts not sending properly as the application's messaging system may also fail in the event that there is a problem with the user's phone. If there is an emergency, do not rely on the 'Personal Safety App' and contact emergency services immediately.

The 'Personal Safety App' is relatively safe to use. It doesn't rely on any server to run and instead relies completely on the user's phone software to run properly. This means that the user's data security (such as phone numbers, names, and emails) will then rely on the phone's ability to encrypt and protect data.

3 Getting started

The user must first download the application on their device and open the application to set up their preferred check-in time. Once a time is set, a series of notifications will popup on the device some time before the required check-in time. The user must check-in by the inputted check-in time to stop the notifications. If the user does not check-in, the notifications will continue and eventually alert the user's contacts through SMS communication.

A visualization of the process of using the application is displayed in Figure 3.1 through the use of a block diagram. It summarizes the above explanation.

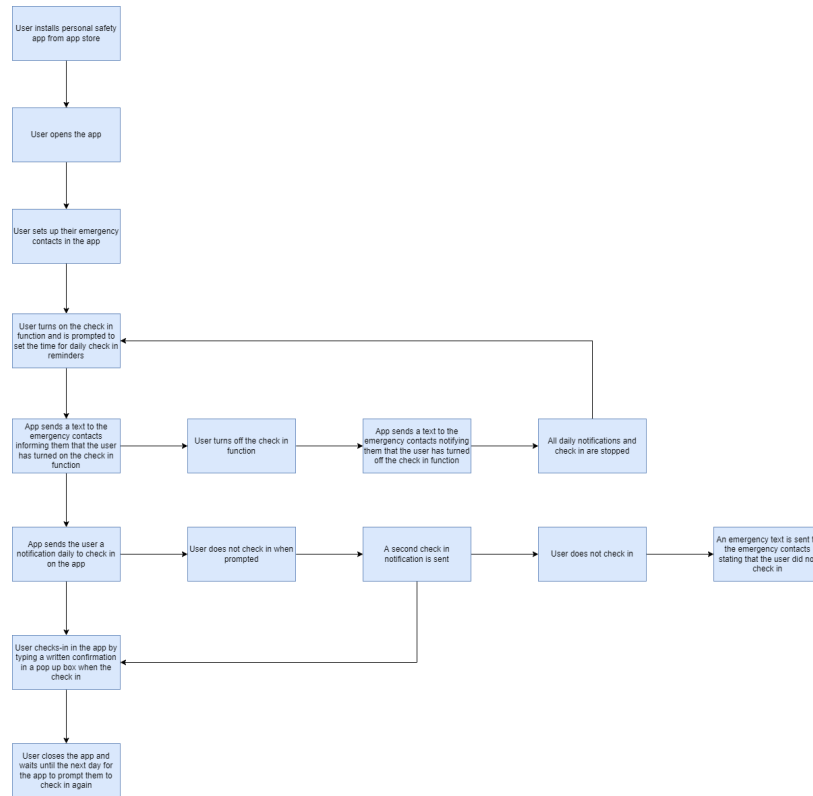


Figure 3.1: Block diagram summarizing the walkthrough for the use of the personal safety application

3.1 Set-up Considerations

The user may access the application once they have downloaded it from either the Apple or the GooglePlay store. They can now open the application which will bring them to the main page. This page will show buttons for the check-in feature at the top, as well as those for the check-in setup and emergency message located at the bottom.

To begin using the application, the user will need to set their preferred check-in time. They may do so by clicking on the button listed as 'Check-In Setup' on the bottom left hand side. This button is highlighted in Figure 3.1.1.

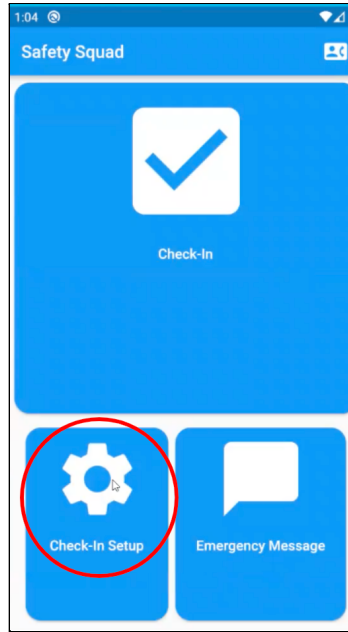


Figure 3.1.1: UI of the App with the Check-in Setup Icon Circled

Once the button has been clicked, the page to set up the check-in will open. At the top will be a red toggle switch that the user may tap on to turn on or off the check-in feature. Below the switch has information regarding the effects of turning off or on the check-in feature. It states that in order to turn off the check-ins, written confirmation is required to ensure that the user is of sound mind. It will also notify the user's contacts that the user has turned off the check-ins. To turn on the check-ins, no written confirmation is required. When opening the application for the first time, the toggle switch will be red, indicating that the check-in feature is in its initial 'off' state. To begin set-up, the user must tap the toggle switch (Figure 3.1.2a). A message will be displayed asking for confirmation from the user if they would like to turn on the feature. The user may tap on 'CONFIRM' to proceed with setting up the application (Figure 3.1.2b). For future reference, the user may accidentally hit the toggle switch, in this case, they may tap the 'CANCEL' button to have the check-in feature remain turned off; they may turn it back on at any time they would like. Once the user has hit the confirm button, a popup to set up their desired check-in time will be displayed. They may enter the hour and minute of their preferred check-in time. They must also indicate whether their check-in time will be AM or PM by tapping on either option to the right of the time. Once they have done so, they can tap on the confirmation button located at the bottom right of the popup display (Figure 3.1.2c). The popup will disappear and the toggle switch should now be green to indicate that check-ins are turned on (Figure 3.1.2d). If the user finds they have accidentally clicked on 'CONFIRM' during the first turn on pop up, they may click 'CANCEL' on the clock pop up. This will stop any process in turning on the daily check-ins, and the switch button will remain in the 'OFF' position. When the switch button moves to the 'ON' position, text appears below the switch button indicating it is 'ON' so that it is clear to the user which state the switch button is in.

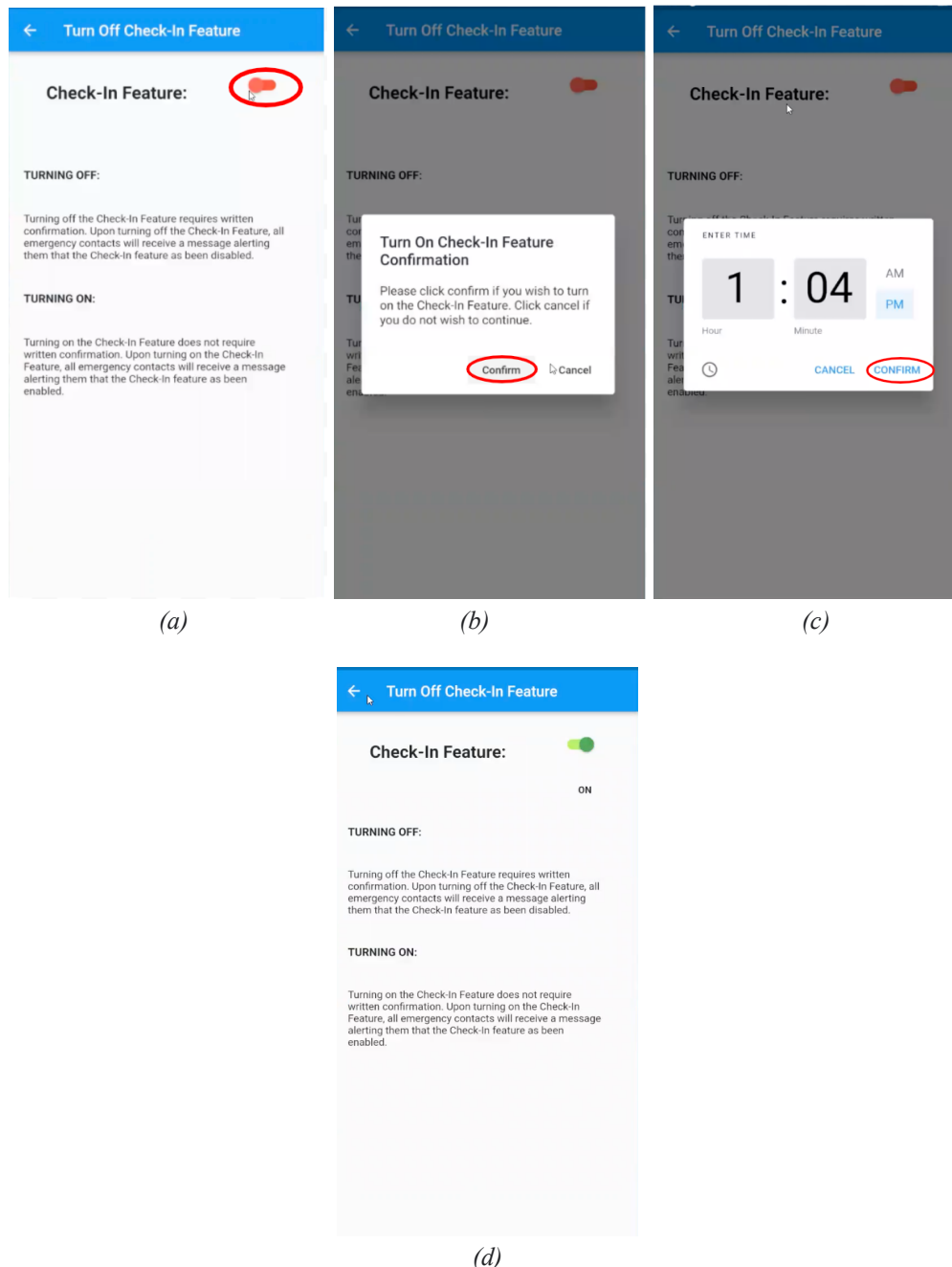


Figure 3.1.2: Visualization of the 3 steps to turn on the check-in feature. (a) tapping the toggle switch, (b) confirming the check-in, (c) setting check-in time, (d) confirming the check-in is on.

3.2 User Access Considerations

The intended users of the application are those looking to ensure their regular well-being with the use of check-ins, as well as those seeking to provide their loved-ones ease of mind. This could include individuals who are at higher risk of experiencing an accident, such as elderly individuals or individuals with disabilities. Although the application was designed to accommodate a variety of users, some functionalities may be limited to certain individuals. The application uses a blue and white colour scheme. Those who may have difficulties differentiating

colours could struggle to differentiate the different buttons; currently, the product does not offer an option to change the color scheme to accommodate colourblind individuals. The application is also limited to the English language; thus, those who do not speak English will need to translate the button labels and messages using an external system. Moreover, the homepage possesses large buttons for easy accessibility; however, the buttons, keyboard, and displays for each function accessed from the homepage could be difficult to perceive if the user has difficulty with viewing small displays and texts. Some devices offer options to zoom in on the application, but this must be done with the phone itself and not within the application.

3.3 Accessing the System

The use of the application does not require a user account. The user can access all features simply by downloading the application from either the app store or the GooglePlay store and opening it on their device. The user must enter valid phone numbers for their emergency contacts.

3.4 System Organization & Navigation

3.4.1 System Home Page and Menu

The system home page is the main screen, which also comprises the menu. On the home page, the check-in tab, check-in setup, and emergency messaging features are all visible. Clicking on any of the corresponding icons for these features, will open up their respective tabs that then allow the user to input the necessary information.

3.4.2 Check-In Menu

Opening the check-in button, from the system home page, takes you to the respective check-in tab. From here, the check-in button can be pressed to check-in, and the check-in confirmation occurs. Sliding out of this screen returns the user to the home page.

3.4.3 Check-In Setup

Opening the check-in setup button, from the system home page, takes you to the respective check-in setup tab. From here, the scheduling of the daily reminder system can be adjusted. Sliding out of this screen returns the user to the home page.

3.4.4 Emergency Message Menu

Opening the emergency messaging button, from the system home page, takes you to the respective emergency menu tab. From here, the user can add contact-numbers as desired, and send messages to the input telephone number(s).

3.5 Exiting the System

When the user would like to turn off the daily check-ins and notifications, they simply need to press the switch button located in the 'Check In Setup' page again. Pressing the switch button will trigger a pop up which asks the user to input the word 'Safe' into a text field. The variations of the word safe that the system accepts are, 'Safe', 'safe', and 'SAFE'. If the user clicks confirm with an empty text field, an error message will appear. If the user inputs any text into the text field that is not one of the accepted variations of safe, another error message appears. Both error messages contain a description of what went wrong and are dismissible by pressing an 'OK' button. If the user types any of the acceptable variations into the text field and clicks 'CONFIRM', the switch button will turn to the 'OFF' state. When the switch button moves to the 'OFF' state, the daily check-ins and notifications will be stopped until they are turned on again, a text will be sent to the emergency contacts alerting them that the user has turned off the daily check-ins, and the text below the switch button will change from displaying 'ON' to 'OFF' so that it is clear to the user which state the switch button is in. If the user cancels the turn-off pop up at any point the switch button will remain in the 'ON' position. The text field is cleared each time the user presses either 'CONFIRM' or 'CANCEL' on the turn-off pop up. Every pop up is dismissible. During the turn off process, the written confirmation pop up is dismissed when the 'CANCEL' button is pressed or when the 'CONFIRM' button is pressed when a proper variation of 'Safe' is inputted. The turn off pop up will not dismiss if the user presses confirm with an invalid text field. The error messages are dismissible by pressing the 'OK' button.

4 Using the System

The following subsections provide detailed, step-by-step instructions on how to use the various functions or features of the Personal Safety Application.

4.1 Check-In

After the user sets up the application properly as outlined in Section 3.1, the user will receive three check-in notifications preceding their check-in time; one reminder will occur 30 minutes before the check-in time, another will occur 10 minutes before the check-in time, and the last will occur 1 minute before. The notifications will be listed from 'Safety Squad App' and will contain a message that says 'Check-in Required in 'X' minutes' (Figure 4.1.1). The display of the notifications may differ according to the mobile device the user is using.

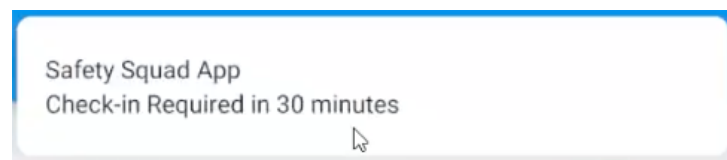
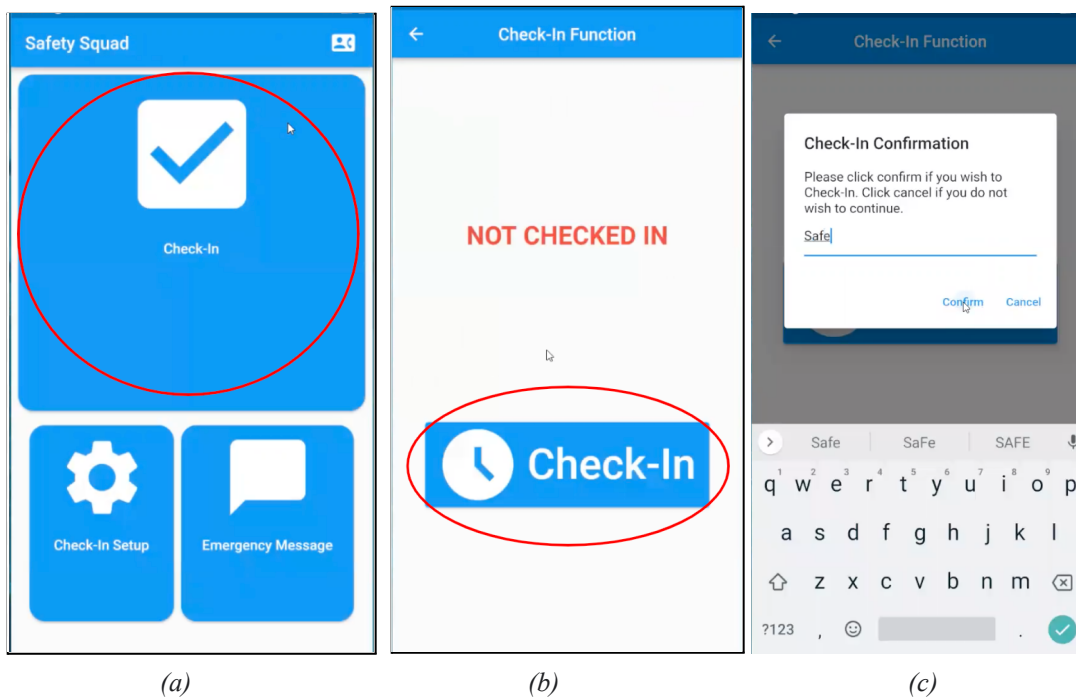
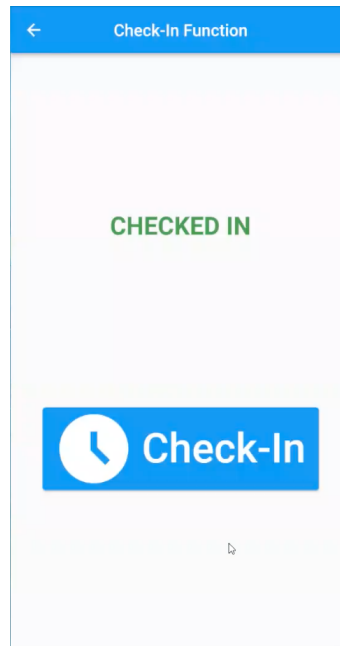


Figure 4.1.1: Sample notification for the Safety Squad App on an Android device indicating a check-in required in 30 minutes.

Before the set check-in time has occurred, the user must open the application and tap on the check-in button (Figure 4.1.2.a). The user will be shown the check-in page. The initial state will display the text 'NOT CHECKED IN' indicating that the user has not checked-in that day. The user must tap on the 'check-in button' with a clock icon in order to fully check-in (Figure

4.1.2.b). By tapping on the check-in button, a popup message will be displayed asking the user for written confirmation. The user must type the word 'Safe' into the textbox in order to confirm that they are of sound mind; they may do so by tapping on the textbox and using the keyboard that will subsequently appear to type in the word (Figure 4.1.2.c). Once the user has typed in the word 'Safe', they may hit 'confirm' located at the bottom right of the popup screen. If the user still does not check-in, another notification will let the user know that an alert is being sent out. If the user types the word 'Safe' correctly, they will be checked-in. The page will now display a green message saying 'CHECKED IN' instead of the original red message saying 'NOT CHECKED IN', to indicate that the check-in was successful (Figure 4.1.2.d). A message in the form of SMS communication will also be sent to the user's contacts indicating the user has checked in (Figure 4.1.3).





(d)

Figure 4.1.2: Visualization of the steps to check in with the addition of red circles to indicate the buttons mentioned. (a) location of check-in button on main page, (b) location of button to check-in within the check-in subsystem, (c) text input to type in the correct word, (d) confirmation of user check-in.

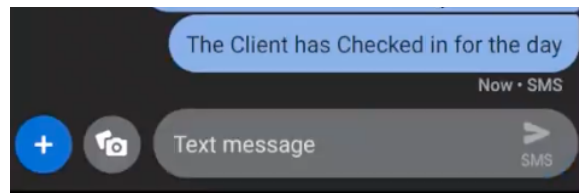


Figure 4.1.3: Message in the form of SMS communication sent to the user's contacts when the user has checked in.

4.1.1 Failing to check-in

If the user has not typed in the word correctly or has not opened the application, the user will not be checked-in and notifications will continue to appear until the check-in time. A secondary notification will notify the user that an alert will be sent to their contacts if they do not check-in and their contacts will receive the alert through SMS communication (Figure 4.1.1.1)

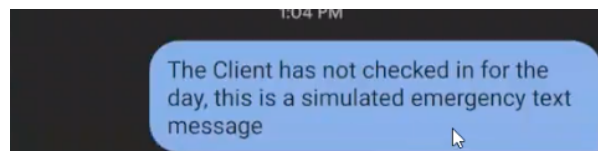
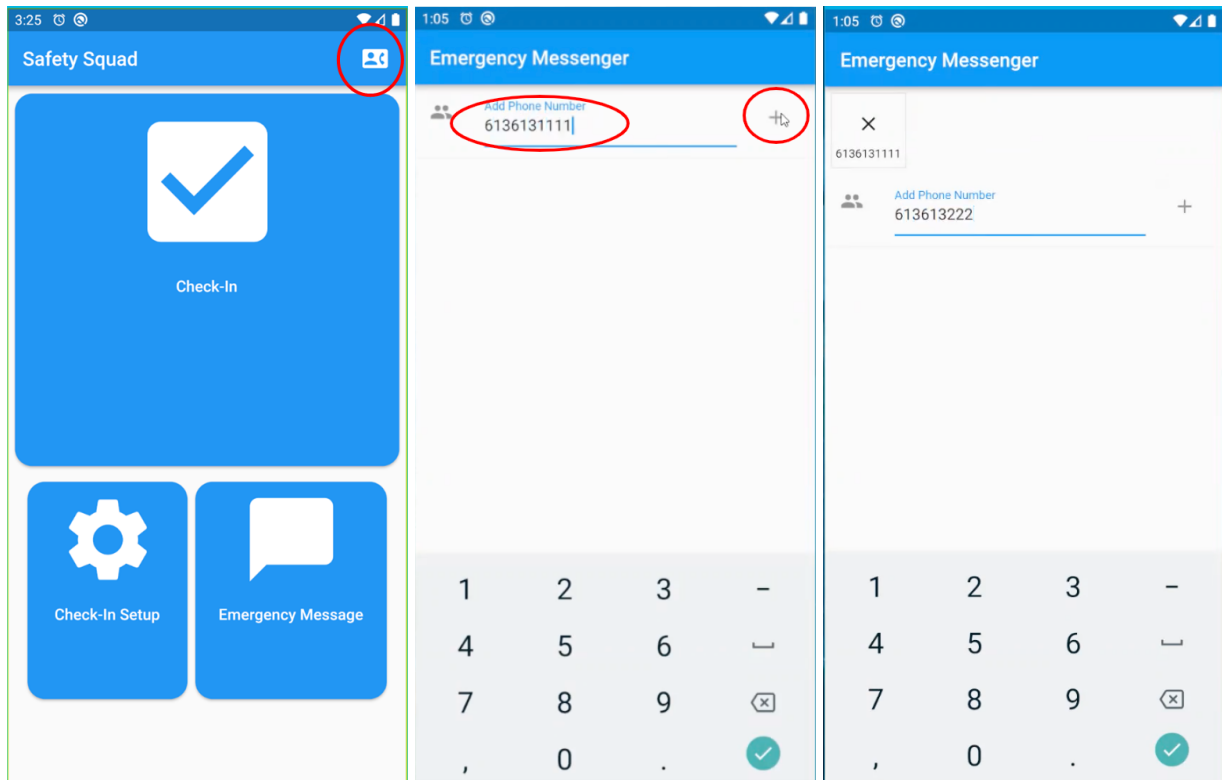


Figure 4.1.1.1: Message in the form of SMS communication sent to the user's contacts if the user has not checked in.

4.2 Emergency Messaging

The emergency messaging system can be used for trivial matters, such as the user not feeling well or minor updates. To use the messaging system, the user can tap on the emergency

message button on the homepage (Figure 4.2.1.a). After opening the emergency message page, the user may enter the phone number of the person they would like to contact by typing in the textbox at the top of the screen. Once they have typed in the phone number, they must tap on the plus sign to the right of the textbox (Figure 4.2.1.b) The typed-in number will appear at the top of the screen and the user may choose to add additional numbers if they desire (Figure 4.2.1.c). The user can now type in the message box a message of their choice (Figure 4.2.1.d).



(a)

(b)

(c)

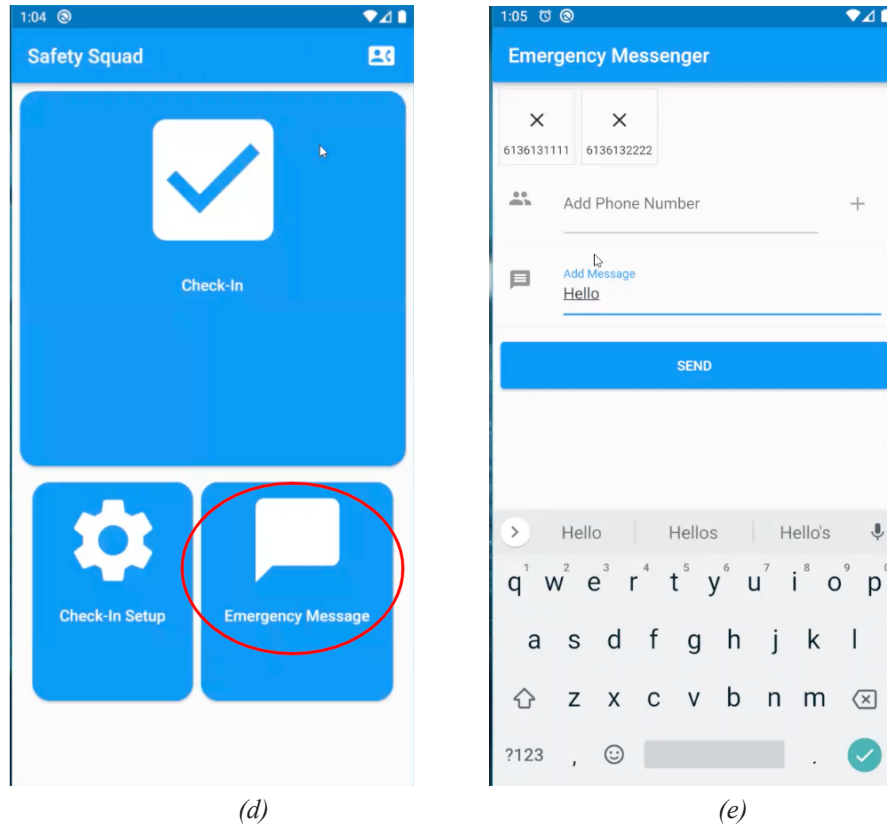


Figure 4.2.1: Visualization of the process of using the emergency messaging function. (a) red circle demonstrating the location of the emergency contact input function on the main page, (b) red circles indicating where the user can type in the number of an emergency contact and how to complete the action of adding the number, (c) display of the subsystem after a number has been added and how another number can be added, (d) red circle on the homepage indicating the button to press to access the emergency messaging system (e) display of the subsystem after number(s) have been added and how the user can write a custom message to the contact(s).

5 Troubleshooting & Support

5.1 Error Messages or Behaviors

An error message will appear when written confirmation is required and the user has typed in an incorrect word and has pressed confirm. Written confirmation is required to turn off the check-in feature and to check-in. If the user does not type the word 'Safe' correctly, then an error message will let the user know that they did not type in the correct word. The different ways the user can write the word are all upper case ('SAFE'), all lower case ('safe'), or only the first letter capitalized ('Safe'), any other variation will result in an error message. The error message in question is displayed in Figure 5.1.1.

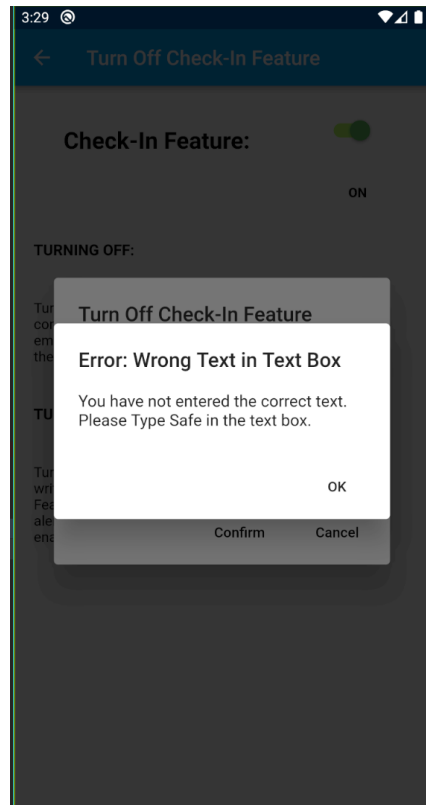


Figure 5.1.1: *Error message triggered when the user types in an unaccepted input into the text field and then presses confirm*

If the user leaves the textbox blank and presses confirm, an error message will appear indicating that there has been no user input. In order to solve this problem, the user must type in the word 'safe' correctly in the different variations as mentioned above.

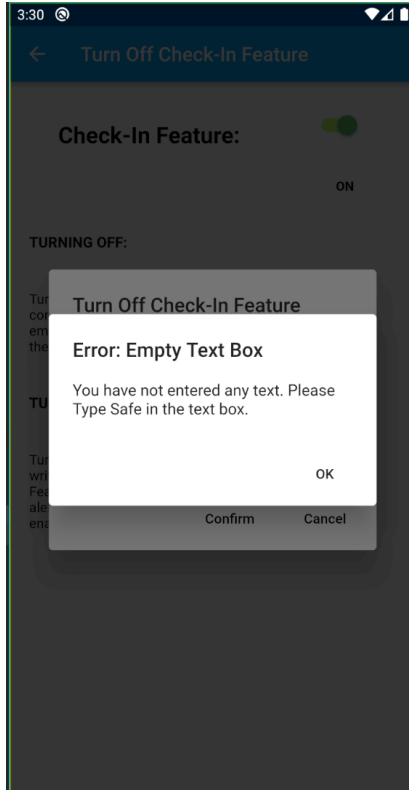


Figure 5.1.2: *Error message triggered when the user leaves the text field blank and then presses confirm*

5.2 Special Considerations

In the situation that the application does not perform correctly, and the user is unable to use the application, the current corrective procedure is to manually close the application on the device it is hosted on.

If the application continues to not function properly, in a more extreme case, the user can delete the application and reinstall it. This will delete user data and therefore the user must set up the application once more (ie. set check-in time, list emergency contacts).

5.3 Maintenance

In order to maintain the functioning state of the application, it must always be running in the background of the user's device. Reminders for check-ins will not form notifications if the user has fully closed the application on their device. The user may exit out of the application whenever they choose and still have notifications appear, but they cannot fully quit the application if they want to continue to receive reminders. If the user does not receive notifications for check-ins, it is possible that they have fully shut down the application. In this case, the user must open the application to resume check-in notifications.

5.4 Support

Any emergency assistance that is not solved by the troubleshooting procedure listed in the manual can be reported to the Safety Squad email address at safetysquadcanada@gmail.com

Please outline the exact version of the application being run, on which device model, and what support is required for.

6 Product Documentation

6.1 BOM (Bill of Materials)

The bill of materials consists of development platforms such as Visual Studio Code. The language that the team used to build the application DOT; the team will be using Flutter, a development kit that is written in DOT. Bluestacks is used for testing subsystems on an emulated device. Additionally an Apple Development account can be purchased to create a free TestFlight for the client, if necessary. However, to reduce costs the Apple Development account was not purchased, but future teams that may want to continue the development of the project may choose to buy a developer account.

Table 6.1.1: Bill of Materials used in the Development of the ‘Personal Safety App’

Item Number	Item Name	Description	Quantity	Unit Cost (After Taxes)	Extended Cost (After Taxes)
1	Bluestacks	Android Emulator	1	\$0	\$0
2	Visual Studio Code	Development Environment	2	\$0	\$0
3	Apple Developer Account	Developer Account	1	\$137.56	\$137.56
4	iOS Devices	Pre Owned Hardware	3	\$0	\$0
5	TestFlight	Beta Testing Program	1	\$0	\$0
Total					\$137.56

6.2 Equipment List

The equipment list consists of Xcode, Android Studios, and Flutter. These items were used by each team member to develop the product; thus, 5 of each were required.

Table 6.2.1: List of Equipment used in the Development of the ‘Personal Safety App’

Item number	Item Name	Description	Quantity
1	Xcode	Development environment	5
2	Android Studios	Development environment	5
3	Flutter	UI software development kit	5

6.2.1 Software

The team decided to use Flutter to develop the product. It was an important decision to choose which coding language to use in order to create the application as it needed to be able to code an app compatible with iOS, while also being beginner friendly to accommodate the team’s inexperience with coding. It was through research and comparison to other languages, such as JavaScript, that the team decided to use the language DOT and thus the UI software development kit Flutter which uses the language DOT to create the mobile application. It also provides the simultaneous development of the application on Android and iOS and is free to download.

Xcode was also mostly needed by the individual in the team that possessed a Macbook, but could also be used by other team members. Xcode was the ideal choice as it could emulate applications running on iOS devices, which would provide more accurate results for prototype testing than using an Android emulator (figure 6.2.1.1). An Android emulator can provide results on how the application could work on a mobile device, but the results could change when running the code designed for iOS devices.



Figure 6.2.1.1: iOS emulator on Xcode displaying an iPhone 11

The team also settled on using Android Studios for the development environment of the application. It is free to download and can be accessed by all the devices possessed by the team members.

6.3 Subsystems Instructions

The components used in each of the following subsystems are functions that are premade by flutter which needs the correct code to work in the desired fashion.

6.3.1 Check-In Subsystem

The check-in subsystem requires three main components. These are an elevated button, pop ups, and a text widget. How the subsystem should function with interaction from the user is outlined in Section 4.1.

The pop up triggered from the user pressing the elevated button labeled ‘Check-In’, requires written confirmation from the user to complete. The pop up should contain a text field to receive input from the user. The written confirmation is to avoid the user from checking in when they are mentally or physically imparied and require help from their emergency contacts. The pop up has a description on it, stating what the pop up is for, and what word is necessary to input. The pop up should accept three variations of the required word. The first variation is all lowercase letters, the second variation is all uppercase letters, and the third variation has the first letter capitalized. The pop up should contain a ‘CONFIRM’ and ‘CANCEL’ button. If the ‘CANCEL’ button is pressed at any point, the process should be immediately abandoned and the user will remain not checked in. The ‘CANCEL’ button is to ensure the user does not get stuck in the pop up and is able to exit the pop up if needed. The pop up should not be able to be dismissed by clicking on the screen outside of the pop up, and if it is possible to dismiss the pop up in this manner, then it should be treated as if the user has hit the ‘CANCEL’ button. If the user presses ‘CONFIRM’ with an accepted variation of the chosen word then the pop up will dismiss and the user will be checked in. Once checked in, the elevated button deactivates until the daily check in feature resets the next day.

Error message pop ups should be implemented to aid the user in determining what they did wrong during the process. If the user presses the ‘CONFIRM’ button with an empty text field or a text input that is not an accepted variation, an error message pop up will appear. These error messages will have an ‘OK’ button, which when pressed, will immediately dismiss the error message pop up. The error message pop ups will have a different description on it depending on which error happened in order to indicate to the user what they did wrong and what they need to change.

There is a text widget implemented above the elevated button stating whether the user has checked in or not for the day. When the user initially opens the check in page, the text widget will read ‘NOT CHECKED IN’ in red lettering. Once the user goes through the process described in Section 4.1, the text widget will read ‘CHECKED IN’ in green lettering. If the user presses ‘CANCEL’ on the written confirmation pop up during the check in process then the text widget will continue to read ‘NOT CHECKED IN’ until the user properly checks in. This text widget is

necessary to reassure the user that they have already checked in for the day, and prevent the user from forgetting they had checked in, trying to check in again, having the button not work, and then thinking that the app had not functioned correctly.

6.3.2 Notification Subsystem

To implement this subsystem, and have it work properly, two separate files must be created. A notification service file, and a main file. Each file contains their proper components.

The two premade functions require the `timezone` and `flutter_local_notifications` packages to be imported into the development environment. From there, the corresponding code to import and unload these packages into their respective dependencies in the development environment must be implemented. The notification service function works to synchronize the application with the global timezone database from Flutter, and to construct the native notification screen on the device the user is running the application on. The main service function uses the notification service function, connects the native notification screen built in the notification service function, and connects it to the check-in widget in the code.

The completed product of this subsystem shows the user their daily reminder for their check-in. This is setup upon initializing the application for the first time by the user, and runs 3 times a day. 30 minutes prior to the user scheduled check-in, 10 minutes prior, and 1 minute prior.

6.3.3 Messaging Subsystem

The messaging system works by calling the system's built-in sms function to send messages. Once the Turn-off system calls the messaging system a function call is made with the message and the phone number that it will be sent to are executed.

6.3.4 Activity Subsystem

The timed activity subsystem is not included in the final prototype; however, this was a feature requested by the client. Therefore, the instructions to build the subsystem could be used for future reference.

To create the timed activity subsystem, a timer is required. It should display the hours, minutes, and seconds. It should be an adjustable timer where the user can input their estimated time of completion for their activity of choice. This should be the main purpose of the subsystem, therefore it should take up most of the display. There should also be a start and stop button for the timer which allows the user to begin the timer, pause and resume it when they desire, and restart the timer. The timer should begin counting down once the user presses the start button.

The activity feature should also have a textbox for the user to enter their location. The location may be typed in by the user manually, or the application could use an autofill feature with the use of a Google API. The inputted text must be stored somewhere in the code in order for it to be used in the emergency text subsystem. The designer may also choose to use a GPS tracking feature in order to improve user experience.

The user must stop the timer before it ends in order to confirm their well-being. They may do so by pressing the stop button. Therefore, in the code, no additional commands must be called if the user presses the stop button before the timer has ended.

If the user has not pressed the button by the time the timer has ended, notifications must be sent out. The process of implementing the notifications is shown in section 6.3.2. One notification should appear 30 minutes after the timer has ended and the user has not responded. The notification should alert the user saying that an alert will be sent to the user's contacts in 10 minutes if the user does not confirm their well-being. In order for the user to confirm their well-being, they must open the application to the activity page. There, they will be prompted for written confirmation and must type the word 'safe' correctly into the text box. If the user manages to type the word correctly within the next 10 minutes, they will be able to access the hiking page again and can choose to either modify the time, or return back to the homepage. If the user does not type the word in correctly, a message should pop up telling the user they have put in the wrong input. This can be done in flutter with the use of alert dialog functions. If the user does not manage to type in the word correctly or interact with the application in the next 10 minutes, a notification should popup on the device and state that an alert will be sent to the user's contacts. The process of sending a message using SMS communication can be described in section 6.3.3. The alert sent to the user's contacts should include the location inputted by the user before they began their activity.

A sample visual of how the timed activity function can look is shown in Figure 6.3.4.

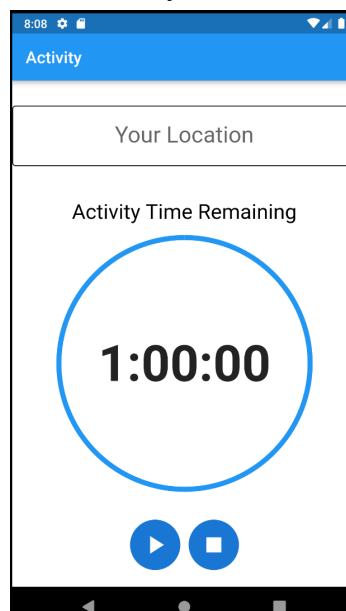


Figure 6.3.4: Sample display of timed activity feature subsystem

6.3.5 Turn-Off Subsystem

To create this subsystem there are three key components needed. A switch button, pop ups, and a clock input. The components should work in the same manner and order that is described in Section 3.1 and Section 3.5.

With the exception of the error message pop ups, each pop up should contain a 'CONFIRM' and 'CANCEL' button so that the user does not get stuck in a pop up and be unable to exit. The pop ups should not be dismissible by tapping on the screen outside of the pop up. If they are dismissible by tapping outside the pop up, there should be mechanisms in place that considers this action the same as pressing the cancel button. If the user presses 'CANCEL' on any of the pop ups at any point, the pop up is dismissed, the current process is abandoned, and the user is brought to the turn-off page in the same state it was before the user attempted to either turn off or on the switch button.

The clock input pop up appears after the initial turn on pop up which is triggered by the user attempting to turn on the daily check ins. The user inputted clock value needs to be sent to both the check in and notification system. These two systems require the time inputted to set the daily check in time, and to set the time the notifications must be sent out before the daily check in.

The turn off pop up must contain a text input from the user to create a written confirmation. This written confirmation is to avoid the user turning off the daily check-ins either by accident, or through mentally or physically impaired thinking. The written confirmation pop up has a description on it stating what the pop up is for, and what word is necessary to input. The input should take three variations of the required word. The three variations should be, all lowercase, all uppercase, and the first letter capitalized.

For the written confirmation pop ups, error message pop ups must be implemented. These will help the user be aware of what mistakes were made when attempting to complete the written confirmation. The error message pop ups should be triggered when the user either clicks the confirm button on the written confirmation pop up with an empty text field, or invalid input in the text field. The error message pop ups have a description on them stating what the user did wrong and how to fix it. They also have one 'OK' button which dismisses the pop up and brings the user back to the user confirmation pop up.

The switch values of being off and on must send different messages to the check in system, notification system, and messaging system. The messages can be a simple true or false value. The true or false value will trigger whether the daily check in and notifications are on or off, and it will determine what texts will be sent to the emergency contacts. These text messages either mention that the user has turned on or off the daily check in. When the daily check in is turned on, the text messages created for certain events in that system will work. When the daily check in is turned off, the text messages created for certain events in that system will not send (i.e. the messaging system should not send a text to the emergency contacts about the user not checking in if the daily check in feature has been turned off). The emergency messaging system is separate from the turn-off function and will not be affected.

The text underneath the switch button indicating whether the switch button is 'OFF' or 'ON' is optional. This text value should change with the switch button. If the switch button is

different colours when 'OFF' versus 'ON', then the text may not be needed. However, if the colours chosen for the switch button do not accurately describe whether the switch button is off or on, or if the switch button does not change colours at all, then it is highly recommended to include the indicated text in order to avoid confusion and uncertainty with the user.

6.4 Testing & Validation

In order to test the prototype for validation of the final design, testing the messaging system was done. To do so, the team sent various custom texts using the emergency message subsystem to a phone number and determined if the message sent using SMS communication was successful. The team also tested how the application would react to different variations of the typed confirmation word 'safe'. The application was able to accept three variations of the typed word, 'SAFE', 'safe', and 'Safe', which was as planned. The team also needed to test the messaging system when the check-in is complete and if the check-in is not complete. It must send a message to the user's contacts saying that the user has checked-in if the user has successfully checked in. If the user does not check in, a message must be sent at an appropriate time to the user's contacts indicating that the user has not checked-in. Furthermore, the notification system was tested when a check-in is successful and when it is not. If the user checks in by the check-in time, the notifications must stop being sent on the user's phone. If the user does not check in, notifications must be sent until the user checks in, or until an alert is sent out to the user's contacts if the user continues to not respond to the application. In addition, testing was done to make sure that the switch button turns on and off when the pop ups are confirmed or cancelled. This test also needed to demonstrate that messages are sent out when the user used the switch button to turn on or off the check-ins.

7 Conclusions and Recommendations for Future Work

The process of building the product came with many difficulties, but the team has learned several lessons through facing the challenges. The team has realized the importance of beginning the design process as early as possible. The application came with several subsystems that required much time and effort put into it. If the team had begun the development process earlier, then perhaps more features could have been implemented in the final prototype to enhance its functionality. The team has also learned to provide a continuous line of communication within team members when developing the application. Team members would individually work on subsystems and would collaborate for the final comprehensive prototype where all subsystems would be combined. While the method of dividing up the work has advantages such as the avoidance of task dependencies, it also possesses disadvantages. Miscommunication can easily occur and thus lead to a subsystem not working how it was intended in the final prototype. Therefore it is important for team members to communicate with each other when combining all the ideas that were developed separately.

In the scenario that the team was given more time to work on the project, the team has several aspects that they would improve and add to the prototype. The team would do the iOS transportation using a better laptop. The laptop that was used to attempt the iOS transportation was not sufficient; thus, the team would need to find a more suitable solution. Moreover, the team would build a cross platform messaging application specific to the application as the product is currently only using SMS communication; however, this only works on Android. Thus adding the cross platform communication would make the application more versatile on devices. Additionally, the team could consider creating separate versions of the application to accommodate different devices. There could be one version of the application solely for Android devices, and another solely for devices running on iOS. This could prove to be beneficial as it would have a high level of performance, it would be easier to test prototypes, there would be less bugs, but it would be more time consuming.

APPENDICES

8 APPENDIX I: Design Files

The information in this document refers to the personal safety application made by Safety Squad using the engineering design process. The code for the application can be found on MakerRepo under the name GNG2101-A15-Safety Squad. Within the MakerRepo, documents for all project deliverables can be found. The project deliverables detail each step and iteration within the design process.

Table 8.1 Referenced Documents

Document Name	Document Location and/or URL	Issuance Date
Deliverable B - A15.pdf	https://makerepo.com/blin023/929.gng2101a15safety-squad	Sept 23, 2021
Deliverable C.1 - A15.pdf	https://makerepo.com/blin023/929.gng2101a15safety-squad	Sept 30, 2021
Deliverable D.1 - A15 Modified.pdf	https://makerepo.com/blin023/929.gng2101a15safety-squad	Oct 24, 2021
Deliverable E - A15.pdf	https://makerepo.com/blin023/929.gng2101a15safety-squad	Nov 2, 2021
Deliverable F - A15.pdf	https://makerepo.com/blin023/929.gng2101a15safety-squad	Nov 4, 2021
Deliverable G - A15 modified.pdf	https://makerepo.com/blin023/929.gng2101a15safety-squad	Dec 5, 2021
android.zip	https://makerepo.com/blin023/929.gng2101a15safety-squad	Dec 5, 2021
pubspec.yaml	https://makerepo.com/blin023/929.gng2101a15safety-squad	Dec 5, 2021
lib.zip	https://makerepo.com/blin023/929.gng2101a15safety-squad	Dec 5, 2021