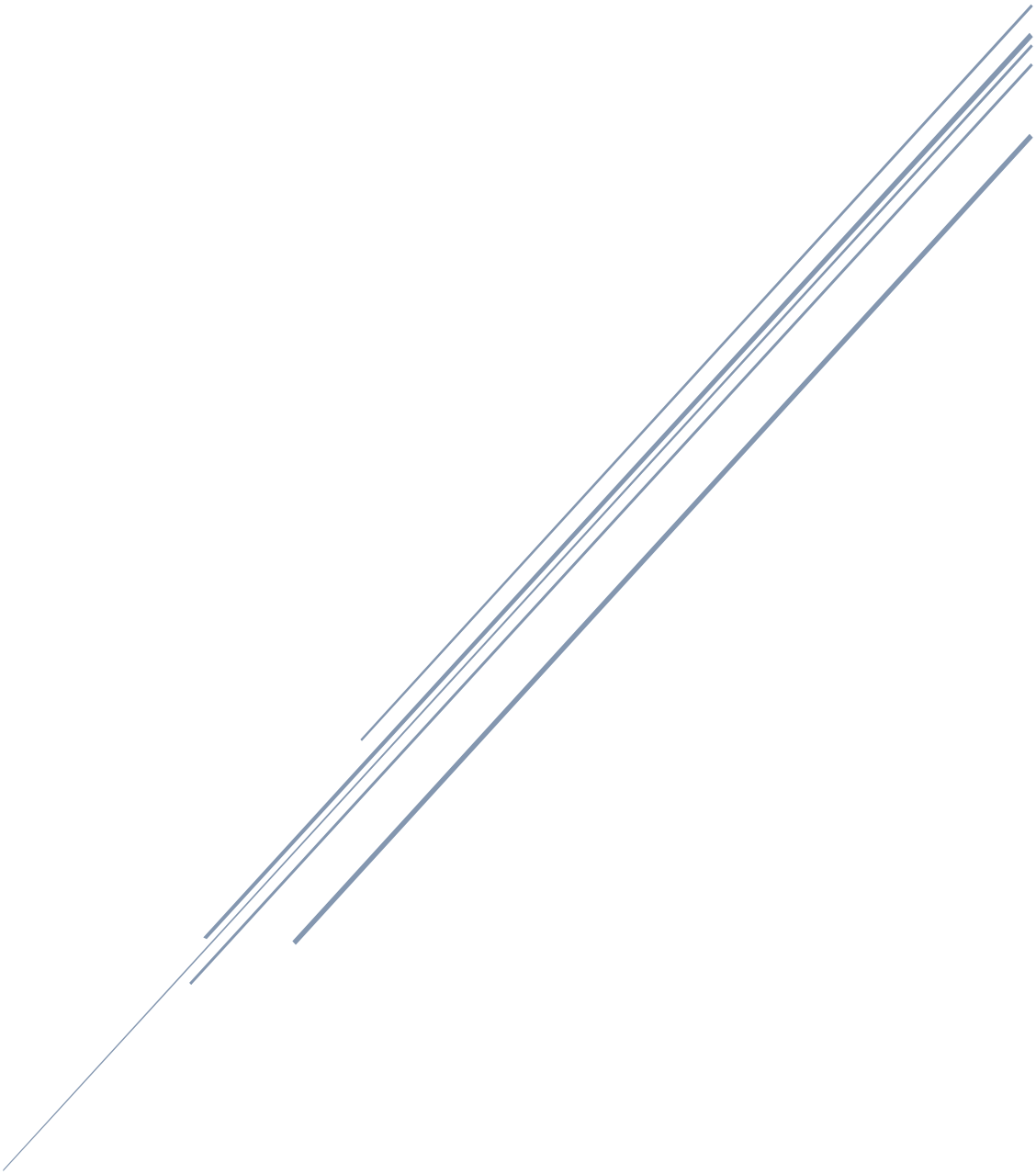
FIT3178: ASSESSMENT 1 – IOS APP DESIGN SPECIFICATION



Monash University
Bachelor of Computer Science

iOS App Design Specifications

Table of Contents

Thank you for your time reading this report. I have devoted hours and hours into this entire report. Which I believe it might be too much by the looking of it. However, I still hope you to read through the entire documentation if possible. Hope you enjoy it, thank you!

Section 1: Application Concept

In this section, I will be discussing about the overall concept of this application. Objectives, purpose, features, and the target audience.

Introduction:

[App name] is a mobile application that provides an extra safety layer within a certain area (based on the users' location) in recording and notifying any suspicious activity, safety concerns such as road damage, and so on that are reported by the public. This is designed for people who have concerns about their safety nearby their current location. By providing tools, resources, and tracking functions to improve the overall safety level.

Features that are available in this application (If time allowed):

1. Reporting, Saving, and Tracking Safety Issues: The application allows users to report safety issues they encounter. Then make it available to other users. At the same time, save the reported issues to further identify trends and patterns.

2. Google Map and GPS Location Tagging: The application can utilize the convenience of Google Maps and then implement the GPS technology to pinpoint the user's location (kind of like Uber Eats where they allow the users to move the pin to make accurate adjustments) which allows all the users to see the issue on the map. This will also show the users' nearest hospitals and police stations. Additionally, the user can also add their families or loved ones so their current location can be available between the members notifying members when they arrive at the specific location, sending help, etc.

3. Safety Resources and Tips and tools: Providing all users with safety tips and resources, personalized emergency contact and your own doctor's contact details, CPR instruction, and so on.

4. SOS: The application can contact emergency services under difficult circumstances without making any sound. There should be two options one is a normal call and the second one is a silence call so the emergency services can understand the situation and then take an appropriate approach.

5. Unique number for identification: Users can their profile which allows them to report issues on the platform. However, to not breach privacy, the application will automatically make every user anonymous. Ultimately, only the application should know about the individual's data.

6. Real-Time Safety Alerts: The application can push notifications to the users that inform users in case of any emergency events occurred or in response to any safety issue that is reported by other users.

7. Integration with Local Law Enforcement and Government: The application is also integrated with the government and local law enforcement to assist them to understand the safety

issues better. Also, making first-hand major safety issues available so the relevant government departments can be involved as soon as possible.

8. Compatible with smart devices: The users can sync and connect their smart devices such as apple watch with the application. So, when the smart devices detect any emergency circumstance the watch will contact emergency services and users' emergency contact immediately.

9. Customize timeline: This feature allows the user to upload their specific timelines. For example, if today the user wants to meet someone or visit a specific location, the user can add their timeline plan onto the app and then share it with their closed contact. Therefore, in case something goes wrong or is suspicious, their close contacts will have rough ideas of where the user has been to or whom they met up with.

There are few objectives that we can conclude in multiple points to elaborate:

1. Improving overall safety: This is the primary objective of the application is to improve everyone's safety in real-time by providing a rationalized location for reporting and tracking safety issues and concerns. Meanwhile, creating better commutations and engagement among the public. In addition, hope to ease any anxiety or concerns of people that are needed to travel to a certain location they have never been to before. This leads us to the second primary objective.

2. Building trust, integrity, and transparency. This application acts as a platform where people can safely express their safety concerns anonymously for reporting safety concerns. By doing so I hope this can improve the overall trust and transparency between the public and potentially local government or any law enforcement agencies. As the application is responsible to track the reported worry about safety.

3. Analysing trends and patterns related to safety concerns: The application can track and store all the reporting data. This enables the ability to perform data analytics and visualization tools to identify trends and patterns with safety issues in different areas and locations. Which can be corresponded to the second objective. This can be extremely useful for the local movements or safety agencies to understand better safety issues addressed by the community and help to allocate needed resources more effectively.

Section 2: Target Audience

Since the core of this application is for safety, the target audiences of [app name] are the general publics who have concern towards safety regardless of gender, occupation, interests, or any other potential factor. The general age should be 5+ as the experience with this application should be intuitive and easy to use. Aiming helping as many people as the application can.

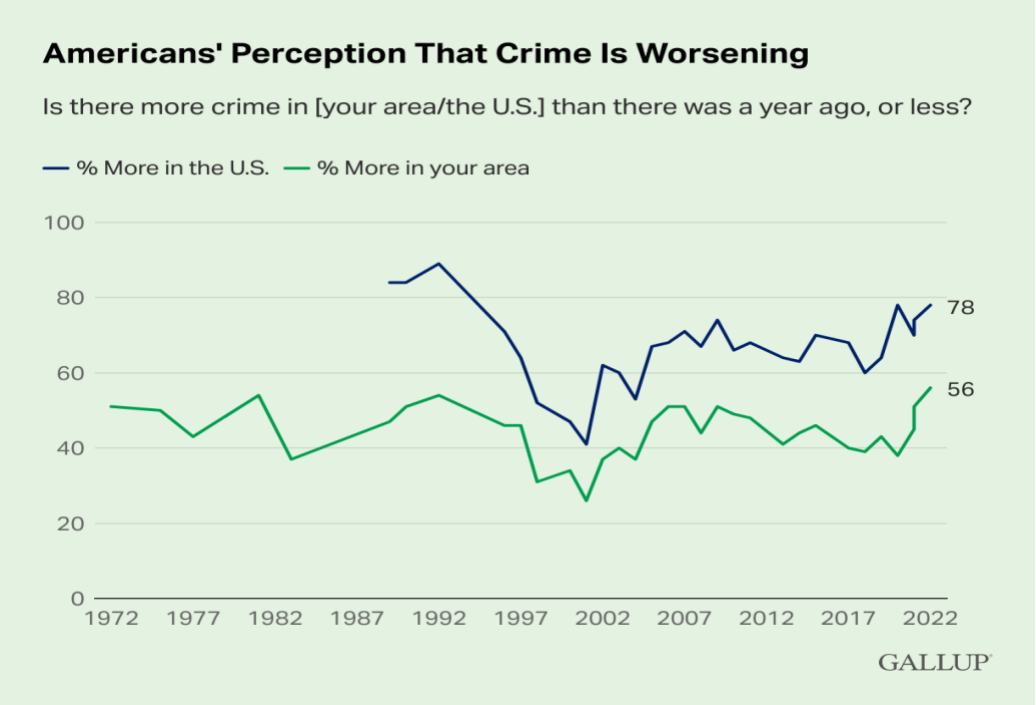
Additionally, targeting US market since the APIs I am using mainly provides US data rather than globally, especially towards safety. On the top of this, as this is an English based application US market is by far the largest. On top of that, there are huge amount of safety concerns increasing through the entire US nation in the past few years. I have gathered some data and graphs to perform analysis. Based on these data. This ensures my application a decent marketplace in the US, especially there are only a handful amount of safety apps that are handy and frequently updated.

Current major safety issue and overall safety issue:

Right now, in the US, the domestic violence and sexual assault rate are very high. According to the National Library of Medicine, approximately 1 in 3 women and 1 in 10 men 18 years of age or older experience domestic violence. Not only young adults but elderly and children also experience physical abuse. Statistically, there are about 25% of children exposed to at least one domestic violence in their lifetime. Annually, domestic violence is responsible for over 1500 deaths in the US.

As for sexual assault. In the US, by 2019 the sexual incidents and preceded by stalking increased 1.9%, and according to RAINN's (Rape, Abuse & Incest National Network) statistic that 1 in 6 women and 1 in 33 men will be raped annually in the US. And this is to be predicted to rise exponentially in the next three years' period. In the past 25 years, there are over 17 millions of women and 2 millions of men had been victims of attempted or completed rape. Yet, the worst part is that majority of the perpetrators and victims don't seek help due to the ear of victimized, society discrimination and other factors.

Overall US crime rate:



Despite the obvious trend of increasing crime rate. According to the State of Safety survey, there are only 55% of the Americans feel on average to live in the US. In other words, almost half of the population doesn't feel safe to life in their own country. Notably, there are six safety concerns increase drastically. School-aged child physically harmed, getting mugged, being attacked while driving, home being burglarized, sexual assault, and getting murdered. (Supported by the following graph)

Americans' Worry About Specific Crimes, 2021-2022

% Worry frequently/occasionally about each type of crime

	2021 %	2022 %	Change pct. pts.
Having your personal, credit card or financial information stolen by computer hackers	74	75	1
Being the victim of identity theft	72	73	1
Having a school-aged child of yours physically harmed while attending school*	**34**	**47**	13
Having your car stolen or broken into	43	47	4
Your home being burglarized when you are not there	43	45	2
Getting mugged*	**33**	**40**	7
Being attacked while driving your car*	**29**	**36**	7
Your home being burglarized when you are there*	**28**	**34**	6
Being the victim of a hate crime	31	30	–1
Being sexually assaulted*	**21**	**29**	8
Getting murdered*	**22**	**29**	7
Being the victim of terrorism	30	27	–3
Being assaulted or killed by a coworker or other employee where you work	9	9	0

* Statistically significant change

GALLUP

Current solution:

Currently the solution is to use limited safety apps, such as red panic button, noonlight, Red cross emergency app, or web services like RAINN (Rape, Abuse & Incest National Network). Yet, these apps and websites provide a good amount of knowledge and assist for people who are seeking for help if needed. However. The issue is that lots of them are not multiple purpose applications. For example, red panic button is purely designed for SOS help. As for both Red cross emergency app and

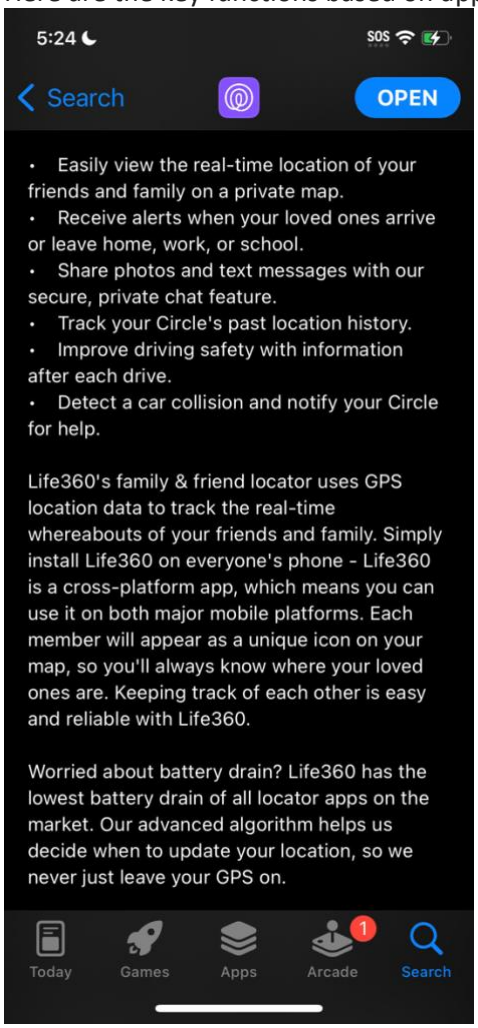
RAINN dedicated for providing emergency tips and information or who to contact. Resulting in users needed to download multiple apps to access to multiple functions or actively looking up for more information.

As for the purpose of this application. The goal is to assist as many people as it can. All the functions within this application are free to use and not “tech savvy” to use, everything is easy, clear, and intuitive. Therefore, users should adopt the interface easily, knowing how to use all the functions without overly detailed walk through. On top of this, providing a wide range of services to help people out when they are in danger regardless you are individual user or group users (family, friend groups, loved ones) everyone individual should get benefit by using [app name]. Most importantly, providing all the users a safe and secured platform to share their safety concern. Which encourages the users’ willingness to seek for help when needed, especially cases like sexual assault, domestic violence, being mugged, being attacked, etc.

Section 3: Competition and Innovation

One of the biggest competitor's "*Life360: Find Family & Friends*".

Here are the key functions based on app store's description.



Life360 Platinum
Be prepared for anything, anywhere with all the features of Life360 Gold, plus:
- $1M in Stolen Fund Reimbursement
- 50 miles Free towing
- $500 in Stolen Phone Protection
- Travel Support with Disaster Assistance
- Medical Support

Life360 Gold
Protect your family on the go with all the features of Life360 Silver, plus:
- 30 days of Location History
- Unlimited Place Alerts
- Individual Driver Reports
- Crash Detection with emergency dispatch and live agent support
- 24/7 Roadside Assistance
- $250 in Stolen Phone Protection
- ID Theft Protection and Data Breach Alerts
- $25k in Stolen Fund Reimbursement

Life360 Silver
Simplify safety with features like:
- 2 Places with unlimited Alerts
- 2 days of Location History
- Crash Detection
- Family Driving Summary
- Data Breach Alerts
- SOS Help Alert

Pros:

- Perfect for family or close friends' group who want to have easy access to each other's current location on the map.
- Driving safety information is provided. It will auto detect whether the user is driving or not. if any car accident happened the app will contact users' closed contacts and emergency services immediately.
- Able to track the users' history location.
- Low battery drains and very intuitive UI interface based on my actual use of it.
- Very high average rating (4.7 / 5) and frequent updates 25 updates in the past 11 months. Based on App store's version history.

Cons:

- The app itself targets group like families, friend groups, or loved ones. Majority of the functions are designed for group of people rather individual. Except SOS function and driving function, however, needing to pay for driving detection. Which results in potential misfit for individual user.

- All the advanced functions require level of subscriptions. Comparing to noonlight, noonlight provides way more free functions to the users. And it's friendly for both individual user and group users. In addition, noonlight doesn't have level of subscriptions. Whereas life360 separates into silver / gold / platinum level of plans. This can potentially upset customers who might just need a specific function either from platinum or gold causing extra penny for the users.

- Unable to use Apple ID or google account or any other way to log in. Potentially result in users forget their account passwords, required users to go through entire forget password process. Also, no auto-fill function if log out the user needs to re-enter everything and run through every verification process again. Which can be annoying time-to-time.

- SOS help alert requires premium subscription and didn't provide same level of functionalities comparing to noonlight. Noonlight provides free SOS service with silent, texting, or call 3 different style of SOS services. Which provides better flexibility when using SOS function.

Second current biggest competitor - Noonlight:

Here is the app description, including free and paid features.

Free Features:
• Meeting a stranger? See something suspicious? Simply add information to your Timeline and we'll share important details with first responders in an emergency.
• We've partnered with Tinder to provide backup on your next date! Share details about who you'll be meeting to your timeline, directly from the Tinder app. Or add a 'Protected by Noonlight' badge to your Tinder chat threads so everyone will know we have your back.
• Prefer texting? If you can't talk in an emergency, you can still communicate with us via text.
• Can't text or talk? No worries, we're still sending help to your exact location.
• On the move? We update your location even if you're in a taxi, bus, running, or walking.
• On-call team of friendly, certified dispatchers ready 24/7/365 to get you help.

Premium Features:
• Automatic Crash Response: Using the accelerometer, GPS, and gyroscope on your phone, Noonlight can detect if you've been in a car accident and automatically send first responders.
• In a panic? Add the Noonlight widget to "Send help!" directly from your lock screen.
• Apple Watch: Get emergency help from your wrist. In an emergency, tap "Send help!" to trigger an alarm. We'll text and call to verify you need help. If you're unable to answer, we'll still send help to your location.
• Wear OS: Get emergency help from your wrist using timer or panic mode. In an emergency, tap Trigger Alarm. Or Set Timer, and we'll send help if you don't enter your PIN when time's up.
• Alexa Skill: Can't reach your phone? We'll automatically send help when you say "Alexa, tell SafeTrek* to send help!" *Pending update to "Noonlight"
• Connect the Apple Health app so first responders have more info to better help you in an emergency.
• Connect your Lyft or Uber account to send your trip and driver information to first responders when you trigger a Noonlight alarm.
• Nest Protect Integration: Do you have pets or precious belongings at home? We'll automatically send help when your Nest Protect detects heavy smoke or CO. False alarm? Just cancel via text or call.
• IFTTT Integrations: From home security to voice commands, Noonlight can send smarter, faster help in an emergency.

If you choose to upgrade to Noonlight Premium, you'll get unlimited access to all Premium safety features and integrations. Payment will be charged to your iTunes account at confirmation of purchase, and will automatically renew at the end of each monthly period. You can turn off auto-renewal at any time from your account settings, but refunds will not be provided for any unused portion of the current period. Please note, if you wish to avoid auto-renewal, you must turn it off at least 24 hours prior to the end of the current period.

We take the satisfaction and safety of our customers very seriously. Please check out our reviews to see how Noonlight is helping people just like you!

Noonlight's Terms of Use: https://noonlight.com/terms
Noonlight's Privacy Policy: https://noonlight.com/privacy

Note: Continued use of GPS running in the background can dramatically decrease battery life. Noonlight will only use your GPS location in the background if you have an active alert.

Pros:

- Exceptional connection with other apps, such as apple health, Alexa, Uber, Lyft, Tinder, Wear OS. Which provides a more cohesive helping experience, comparing to life360 can only connect to Wear OS.

- Free timeline functions to use to track who you meeting and important details, free on call 24/7/365 certified dispatchers for help. Comparing to life360 they don't provide the same service, moreover, users need to pay to get customer service responds to issues within 24 hours.

- Unlimited access to all the functions once paid for subscription. Comparing to life360 users need to pay different prices for 3 levels of premium prices to access all features.

- Very high review (4.7 / 5) very clean UI interface and intuitive to use as well. I think it's matter of preference when comparing to life360's UI interface. Both are excellent.

- Great for all family, close friend group, loved one or individual use. Comparing to life360 mainly focusing on family and friend groups and loved ones.

Cons:

- Can drain battery life drastically if running GPS function at the background. Unlike life360, specifically designed for low battery drain while running GPS function.

- Doesn't seem to update that frequent in comparison to life360. Noonlight hasn't been updated for over 2 years based on app store version history.

- Noonlight's premium feature doesn't provide services such as stolen fund, stolen phone protection, stolen fund in Reimbursement. Though, noonlight has car crashing alert but life360 provides 24/7 roadside assistance. These features are excellent for the people who want extra cover while driving and stolen properties.

Innovation:

Here is the outline of innovations when compared to other similar applications. Yet, my application might share similar functions with others'. There are still opportunities for me to make improvements.

- Safety reports by the users are stored and analysed then convert into Swift charts. Rather than reading through the texts, the application should convert the gathered data into charts providing a better understandability and visual representation for the users to consume. Not only this, safety reports provided by users will also be stored, allowing users to track their history. Then a further analysis of patterns and trends related to safety concerns. The other two competitors' apps don't provide any data visualization for the users. Or the crime rate, and incidents over a period. This allows the users to process an excessive amount of information easily.

- Since the application will be running on an internet-connected device with necessary sensors. The application processes the capability of providing current reported safety concerns to nearby users' current location. Furthermore, using a pinpoint and gesture handling framework, users can tap on the map which can prompt more detailed information regarding safety reports. In comparison to the other two competitors. Both don't include the function of letting users report safety issues.

- Whenever the user publishes a safety concern report, their personal information will not be presented but the report information. Protecting the users from potential assault or harassment towards others. Also, only the people within the same group should see the users' detailed information. Such as families that use place alerts, the map will show everyone's current location. However, I believe both apps don't necessarily hide the users' details. Which has its pros and cons; however, I believe hiding personal information from other users is important, especially since this is a safety application, not social media application.

- Multiple ways to log in, implementing biometric authentication, Apple ID, and Google. Providing the users with ultimate convenience but also a very secure authentication process. For both Google and Apple login. Log-in will go through both company's authentication systems. Unlike, other competitors can only login in their way. Requiring users to run through phone verification every time if log out and potential cybersecurity. Which doesn't make much sense. Just think about what if today's user is in danger and needed to use an app, however, require going through all the process rather than a simple click or using biometrics to access securely and effortlessly.

Section 4: Feasibility and technology

In this section, I will discuss about the feasibility with the technologies that will be implemented in the application. This will be separated into three parts API, Framework, and UI elements. Under each part there will be a thorough elaboration relates to different part's sub-topic. This should provide a clear outline of feasibility along with the explanation of the technologies utilizing. Additionally, each API is linked to their website. Please click it if you are interested in it.

APIs:

1) *BandWidth*

   This API provides both Emergency calling API and Emergency notification API. The emergency calling API is used to provision and manage emergencies associated with telephone numbers. This means when the users of the application request an SOS service, this API will automatically gather essential information and then send it to the emergency service to help them to locate the user's current location quicker, thus, preventing a waste of time in locating. For the emergency API, it manages notifications sent when numbers provisioned on the 911 Access Dashboard make a call to emergency services. In addition, BandWidth is available globally which allows the user utilizes SOS service accordingly.

2) *Fake Data API*

   This API provides a core functionality that is to create fake call with fake data. With this API I can comply with one of the features in my application, fake call. As this allows the users to generate fake calls when needed. This API can additionally create fake address, payment, internet data, etc. Which might not be so relevant but extra functionality can be used potentially.

3) *Your Mapper2* & *CrimeoMeter*

   Your Mapper2 provides a wild range of datasets that include crime, property values, care accidents, sex offenders, bridge safety, toxic, etc across the entire nation of the US. This is used to help the users to understand each location's safety level and then further help the users to make an evaluation. On top of this, I am also combining another API that has similar functionality. CrimeoMeter provides information relates to the total number of incidents and the incident types in a defined area over some time. This can help the application to analyse current trend and provide valuable information to the users in Swift Charts.

4) *Sign in with Apple REST API*

   This API can help the log in service connects to Apple's authentication servers, to generate and validate the users' identity token.

Frameworks:

1) *Swift Charts*

   The use of Swift charts can help the application transform collected data into informative visualization for the users. That can use to communicate patterns and trends. As in the

application, one of the features is to track the data and store them, then further perform analysis and visualization. Utilizing swift charts can help the process and present the existing into a manner that is easy to understand and concise. Most importantly, the framework also supports localization and accessibility features that can be used to improve the quality of the application.

2) *Firebase Cloud Platform*

As one of the functionalities of the app is to allow users to upload their concerns about safety in real time. Which can be formed into a text message style. Therefore, implementing the firebase cloud platform enables the ability for users to perform in-app texting meanwhile storing the data on the cloud for later use in data analysis and visualization. Also, for login function from Google.

3) *Maps and local services and GPS*

The core technologies that will affect the entire application. As it's functionality within the application is to help locating the users' accurate location that provides relevant information to the users as the application is to provide safety services around the users' current location. Especially, now google maps can pinpoint the coordinates for the image using the photogrammetry process with GPS data. [1] Which further helps the application to ensure the exact location of the user.

4) *Location notifications*

This functionality can assist the users to ensure their loved ones, families, or friends to be notified when the change of the location. The users can choose to whether they want to notify their selected people when the users arrive at a specific location or leave from a location. This can be a very crucial function that can help everyone. As this location notification provides both arriving time and leave time. Thus, if the person didn't arrive on time for a while or left the location and unable to location the person. Which the user should be aware of the situation then contact the person to check whether they are in trouble or not. Even for polices, if anything went missing this can also help to location where did the person go on what time. Which complies with the purpose of my application.

5) *Gesture handling*

Implementing Handling UIKit gestures can be beneficial as it uses gesture recognizers to simplify touch handling and create a consistent user experience. Particularly, in the application I am using Google map service and allowing users to pinpoint their current location. Thus, having this UIKit can help improve the gestures handling while the users utilize the pinpoint function, also zoom in zoom out on the map, and tapping on the map to obtain extra information when needed. As such, having consistency and muti-gesture handling functionality can certainly improve users' experience while they are utilizing the application.

[1] https://webo.digital/blog/the-tech-behind-google-maps/

6) Authentication Services:

This is the framework that gives the users the ability to sign into my application with their Apple ID. Enabling look-up for users' stored passwords within the sign-in flow. Which can improve user experience as it also provides passwordless registration and authentication. This framework can enhance users' willingness to utilize the app as there is no annoying login, creating a new account, then consistently using forget password service. I believe this is the issue lots of people encounter while using multiple applications, including myself.

Technologies:

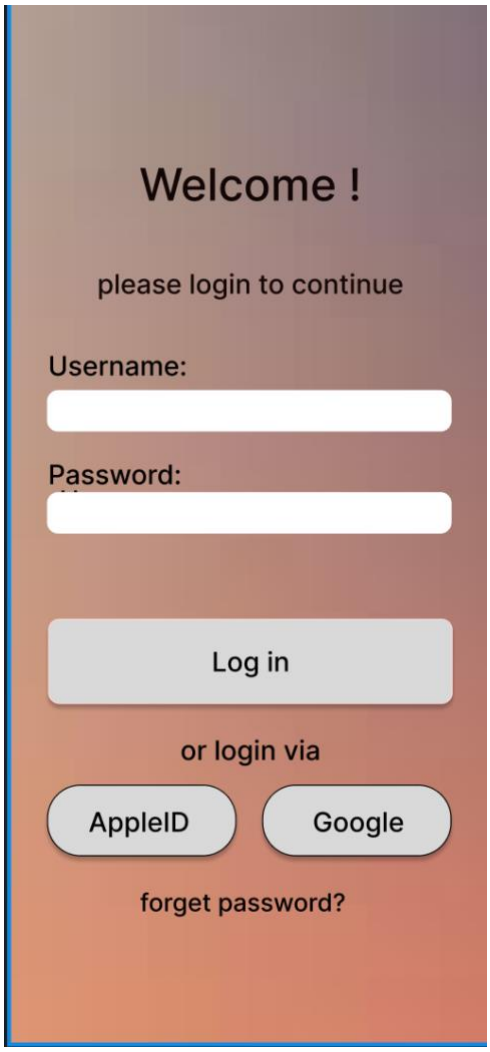
1) Sign in with Apple:

This is the technology provides by Apple that allows me to provide the users the ability to sign in with their apple account in the application. I think this is a great way to be less annoying about creating new account just for using the services. As, sign in with apple eliminates the need for additional set-up steps, allowing the users to focus more on the app. Not only this, since the log in service is provided by Apple, the authentication steps for verifying the users are more secured.

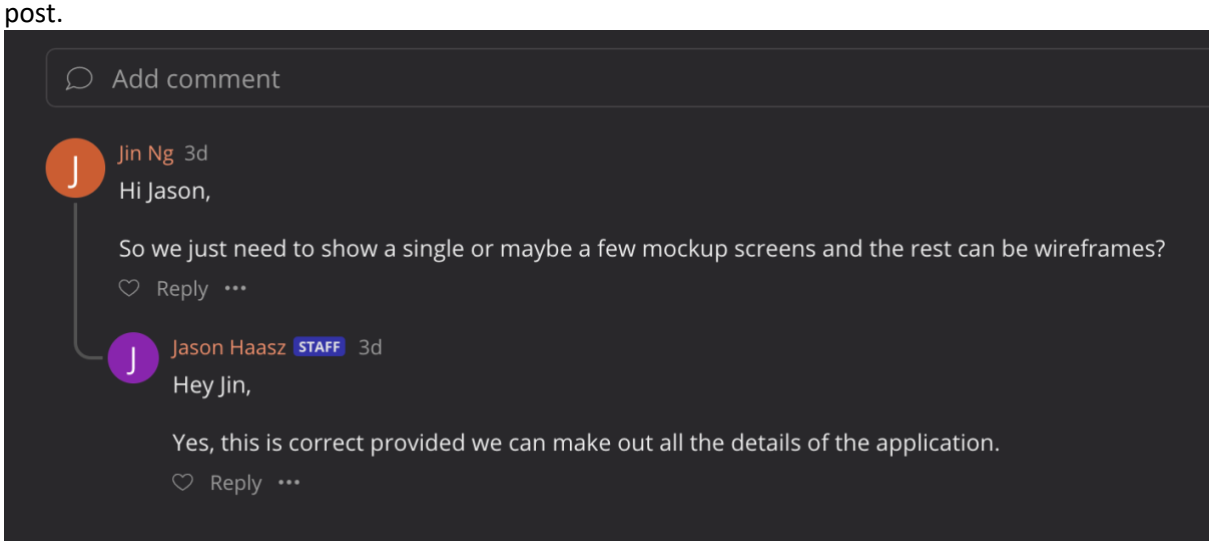
Section 5: Interface Design and Storyboard Mock-ups:

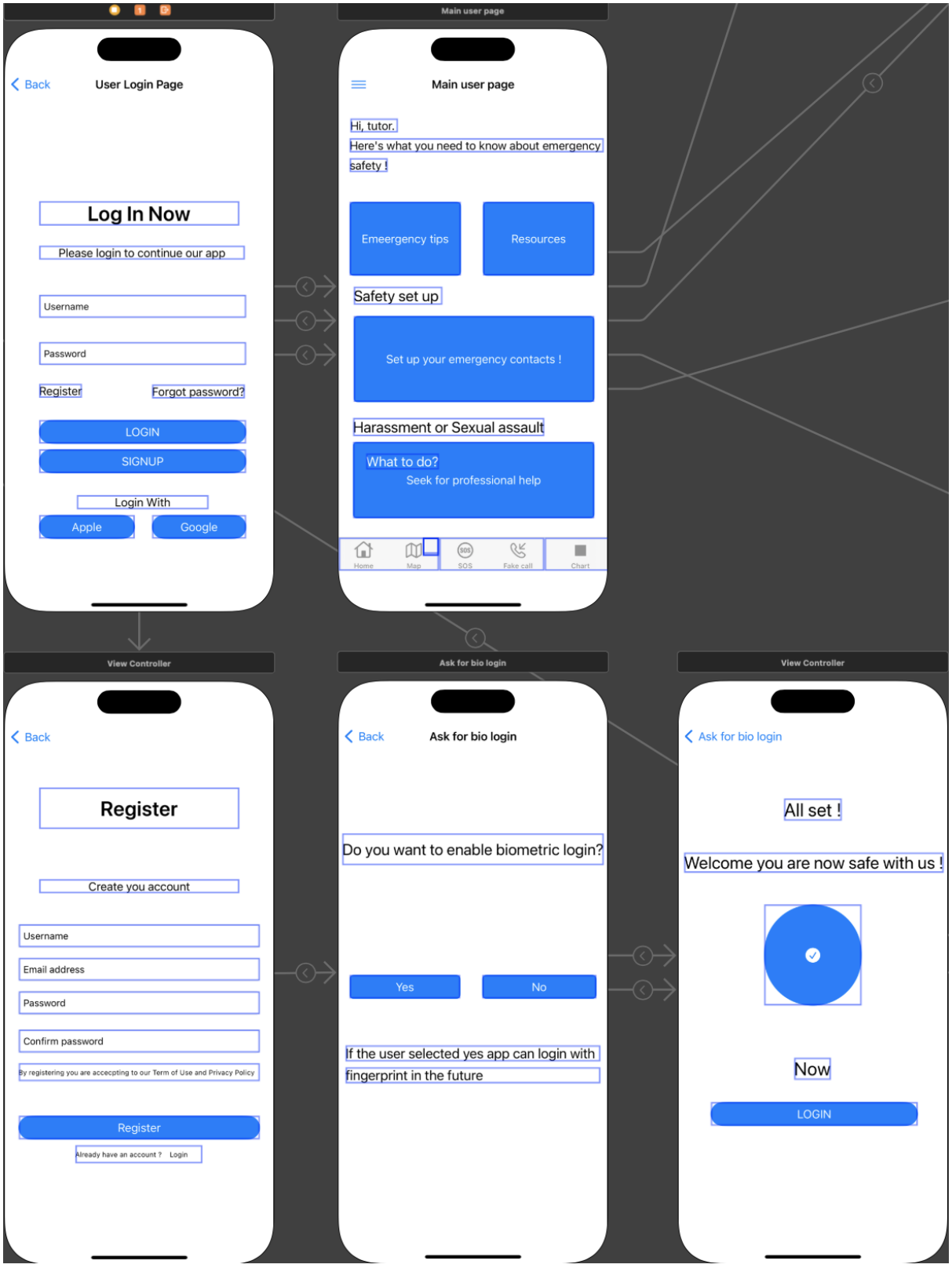
This is the mock-up design that the entire application is going to look like. As I have asked during the lab, I believe color isn't the most important at this stage. Also, the UI will comply with Apple's HIG guidance. The entire app will have two color modes, light and dark. The button in this application will be set to at least 44x44 points to accommodate a fingertip. This is to help the users to feel intuitive and easy to target the desired button. On each button, the short description will inform the users clearly what function can the button do. The entire UI element will adjust its size when using different size screens so the users will not experience any overlapping issues. Ensuring every user's ultimate experience. The overall style should be clear, minimalistic, and intuitive. This is also to follow HIG's "Accessibility – Simplicity" enabling familiar, consistent interactions that make complex tasks simple and straightforward for the users. Meanwhile, combining with multiple technologies, Maps, Sign in with Apple, Sign in with Google.

Like the screenshot shown. (Colouring I believe is not the core at this stage as we discussed during lab).



Also, I will be providing all my other screens with wireframe as this is conrimed by the staff on ed post.

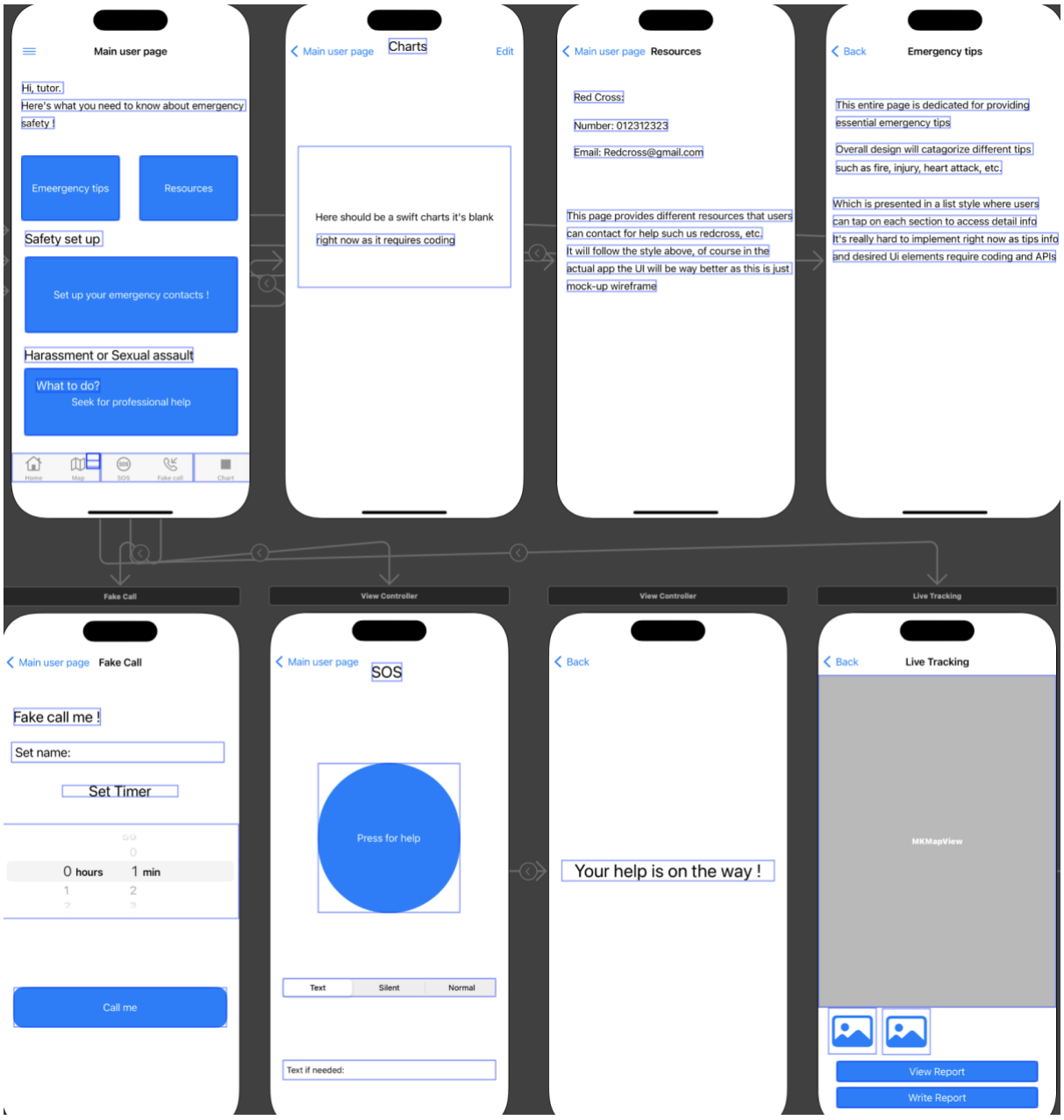




Here, when the user first opens the application, it will directly show the user's login page, wants to log in, or sign up. If it's a new user, the SIGNUP button is also available under LOGIN, which provides users with a very intuitive way to perform their desired action. Once hit SIGNUP, it will then transit to the register page (if the user didn't choose to log in with either Apple or Google). For the sign-in with both the apple and google buttons I am well aware of the HIG requirement(In the actual UI I

will change to it what Apple accepts but right now this will do for providing you the general mock-up idea).

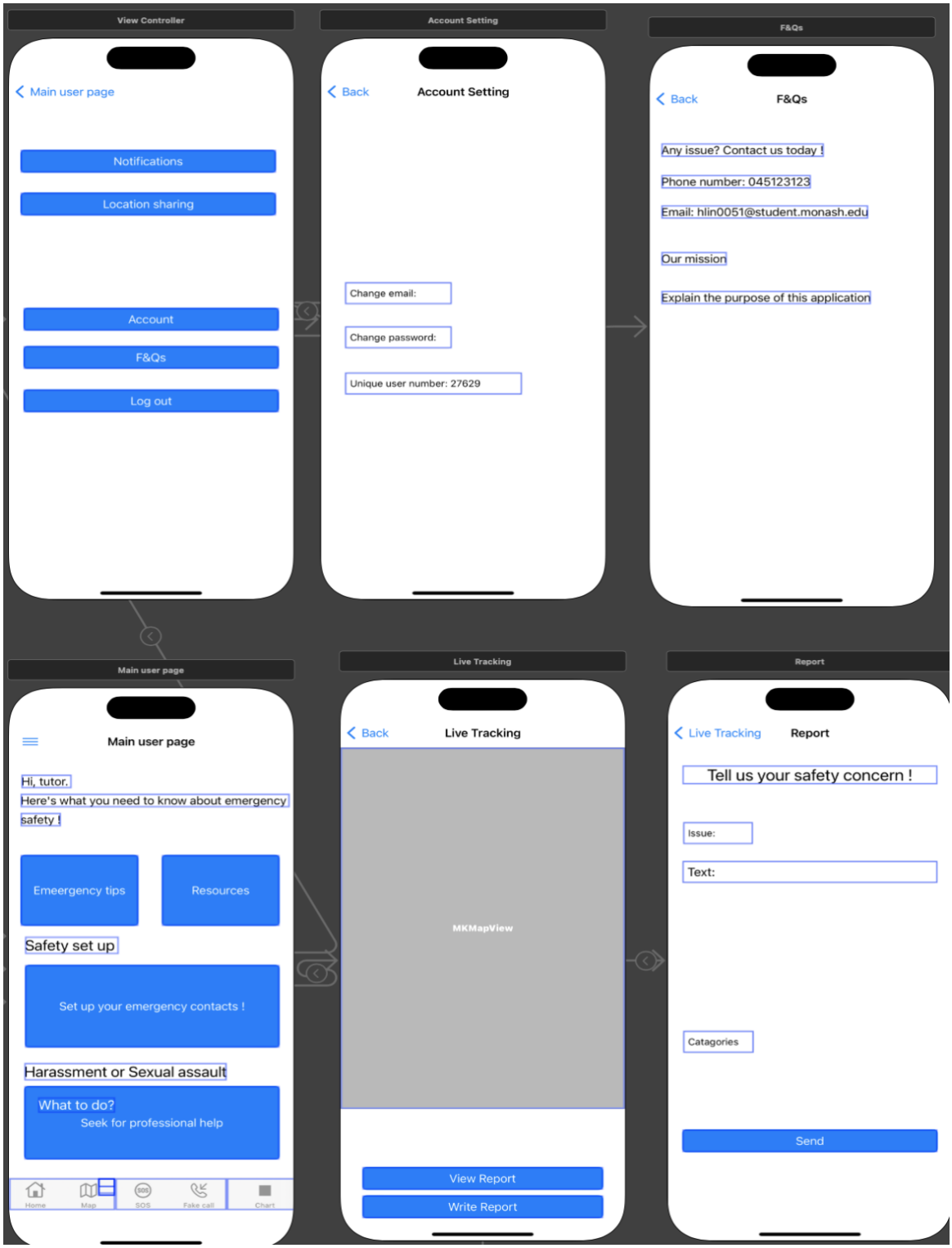
Here, in registering page we ask for users' info, then once finished entering, asking users whether they want to enable biometric login function or not (All users essentially will be asked once answered this will never be asked again unless change in app setting page). Then prompt a Welcome message to all users per HIG "Sign in with Apple" requirement as well. Next, requiring users to enter their new account. Once signed in direct users to the app home page. Else if the user is an existing user already, then once finished login, direct the user straight to the app home page.



After entering the home page, the users will see a greeting message. Also, with clear simple text button. Therefore, the users now what's the rough function of provided buttons. Meanwhile, utilizing Tab Bar UI element. And Each Tab Bar item complies with HIG requirement as well. It is designed for navigation rather than performing specific function, as you can see, it helps to direct to different areas in the app. Each tab has a concrete noun title and image so users can understand

type of content easily. In addition, I have also checked other competitors' app, they generally have 4-to-5-tab bar sections, in this case I used 5.
When clicks on Map, it directs users to live tracking. Showing user's current location using Google map function. And the user is able to view other users' safety report or write report if the user has any concern regards to safety. If clicks on SOS, it directs the user to SOS page, with huge button "Press for help". Once pressed, it will prompt a confirmed message to prevent mis-click this applies to lots of emergency app. Also, the user can choose text, silent and normal. Three different styles of calling SOS under different circumstances. This segmented control complies with HIG requirement as it's not overly crowded, the general size of each section is evenly spread out, it's provided for choices that will affect a state. In this case, the state of contacting SOS, doesn't mix with both image and text, and capitalization. Once user confirmed with SOS service, it will prompt message to reassure the user.

For the fake call, the user can set their desired name and even set timer for user's desired time the fake call will come in. for the Chart, as at this stage I am unable to show anything. Since it requires coding, and the idea is explained in the application concept. Having a chart in app also satisfies HIG requirement. As the visual hierarchy helps communicate the relative importance of various chart elements. Resulting in a decent accessibility for majority of users. For resources (more detail in screenshot), the page is aimed to provide different help stations, such as redcross or other organizations' contact details to save users' time searching up. As for emergency tips (more detail explain in screenshot), it provides various circumstance's emergency tips.



If the users clicked on the three lines, it would take the user to the setting page. Which provides 5 different options. For both notifications and Location sharing. Both would just prompt a message asking whether the user gives the app permission to perform both actions. For the account, it will direct the user to their account setting, where they can change a certain part of their information. However, since there isn't much the user can change, it looks empty on the screen. Plus, this is just a mock-up, so no color or decoration is provided. The same applies to F&Qs. It provides the user's the app company's detail, in this case, me. And providing a short description of the purpose of the app. As I stated, this page looks empty due to the mock-up stage and it's purely for providing info as well.

For, emergency contact. The user can create their contact details. So, when an emergency happens the app will notify them. Within the creating emergency contact section. It will ask for your name, relationship, and contact number. To help ensure whom the app contacts, the emergency services know how to take action accordingly. On top of that, if the user desires, they can also add their emergency contact's photo. Allowing the user to perform some customization. Lastly, the view report. This is very hard to present at this stage as well. Since these parts require API and actual map implementation for them to be functional. Therefore, I can only provide the design detail as descriptive as possible (see screenshot for details).

This should be the entire explanation for the UI mock-up. In this case, I have a huge number of elements that require me to implement the actual function for it to represent on the screen. Therefore, I tried my best to explain it in as much detail as possible. I hope whoever marking this understands it, and I appreciate it.

Section 6: Scope and Limitations

For the minimal functionality that the app should acquire are:

SOS, fake calls, Google Map GPS location services, updating concerns towards safety by the users, and every user in the relevant area should be able to access the information, emergency notifications, real-time safety alerts, and log-in services.

As I have listed previously under Sections 1 and 3, thus, I won't do an extra explanation about the minimal functionalities to avoid repetition. Therefore, I will do the justification about why those functionalities are needed to make the application viable.

For this safety app to be effective, it should provide users with up-to-date safety reports with real-time safety alerts to minimize the risk of danger. Having an SOS function that allows users to call for help silently in case of serious trouble, such as being held, hostage. This function is important as it ensures the user can discreetly call for police enforcement without alerting the hostage taker. In my opinion, this is a valuable function to have. On top of that, using GPS and Google map service can help to send accurate addresses when the users required emergency service along with emergency notifications to alert users' contacts directly. These basic functionalities are necessary for my safety app to fulfill its core value of minimizing the risk of danger and providing users with a sense of security.

I believe those are the basic requirements that I need to comply with to make my application at least to be a viable and useful product.

The need to learn is everything I listed in this document as I didn't have any experience related to app development. As such, to achieve this task, there is a list of things I need to learn and research.

- **APIs:**
  For the APIs, I need to learn and research how to cooperate with my application for the functionalities of SOS, emergency call, emergency notifications, fake call, and datasets of crime rates and incident types. Based on the initial research, those are the functionalities that the APIs I found provide. Moreover, read the documentation for each API to learn about how to access it and whether using it requires a key, and whether free or not. Additionally, searching for a complete guild that explains how to access and use APIs are important learning and searching step.

- **Databases and firebase:**
  I will need to learn how to use both in-built databases and firebase. Since there is a potential need for the users to log in, thus, I will have a need to store users' data on a cloud platform. Meanwhile, if not using the login function, then I will need to use built-in databases to store users' information locally.

- **Frameworks:**
  Based on my initial research, Xocde can utilize the functionality of Google Maps by downloading the SDK. Nevertheless, more research and learning needs to be done to make the user of the functions that Google Maps has provide. I believe studying this documentation, consulting with my demonstrators, or watching some tutorial videos can help this process. yet, learning how to use any utility and open-source libraries.

Same applies to learning other frameworks as well, gesture handling, Location notifications, and swift charts. Reading the existing documentation on Apple developer pages, then researching any part having difficulties understanding is the key process of learning those frameworks I believe.

- **UI design:**
  The need of learning and research a quality UI design is extremely crucial for all applications. Since plenty of the functionalities are linked together. Especially, extracting the data such as a particular area's crime rate, and safety issues reports, then combining with the Google map services. Where the users can pinpoint their current location precisely. On top of that, the corporation of gesture handling on the user's device. Allowing the users to use different gestures to trigger different information on the UI interface. Most importantly, following the HIG requirements by Apple. Ensuring the design has matched the quality of the UI interface so that users can navigate effectively.

I believe to implement, learning, and research the listed functionalities above should take weeks of time just based on the unit teaching rate and the required minimum hours I need to devote to this unit. Therefore, I don't believe I can complete my project within a matter of hours. Plus, I will need to learn pretty much everything from scratch due to my inexperience.

Nevertheless, if time allowed, Compatible with smart devices and Integration with Local Law Enforcement and Government and customized timeline will be the functions that I would like to implement. As for these functions, they are not essential features but rather enhance features. As for smart devices. Like the Apple watch. Since the smartwatch can detect users' heart rates and monitor health states, therefore, triggers emergency alerts quicker if anything happened to users. Timeline allows people to track, plan, and whom the user is meeting regarding the user's destination. Providing extra information for the close one in the case didn't arrive home on time, or not responding, etc.

Section 7: Estimated Project Timeline

Under this timeline section I am going to break down my plans into few prototypes:

Between now and to end of week 5:
Before implementing everything, I will need to research and learn all APIs, technologies, and frameworks that I am planning to use. This should take at least a week to 2 weeks to learn. Meanwhile, playing around with the UI design and then testing out things which I believe using Swift playground is a great place to do so. Potentially discuss with demonstrators about overall UI design and attend consultation if necessary.

Prototype 1 (Roughly end of week 6):
After all the initial testing and playing, I should be able to construct the majority UI where the functions are ready to be implemented. The UI might still need to configure constraints. Everything is in place, transitions, linking, protocols, and delegates. And the UI constraints should be able to adjust dynamically for different sizes of devices without any issues.

Prototype 2 (Roughly end of week 7):
In this prototype, I should have implemented rather simple functions such as fake calls, emergency tips information, button functions, and map function (zoom in zoom out), able to use log in function with firebase function, and personalized profile.

Prototype 4 (Roughly end of week 9):
In this prototype, I should have implemented all the functions I need for at least a viable application. Checking against the time if time allowed, I might be able to implement an extra function like compatibility of smart devices.

Week 10 onward - bug fixing and minor adjustment and optimization:
If everything goes to plan, I should start debugging my application and attempting to eliminate all the bugs it can potentially possess. Also, the optimize the code construction, memory usage, reusability of the code, avoid re-processing data, checking the use of correct data type, and most importantly the launch time reduction, UI interface. Then, checking everything works as intended. I believe this process should a few weeks to complete as debugging can be time-consuming and hard to resolve the issue.

Week 13 to 14:
At this stage the application should be ready for presentation however, I should test if the application doesn't malfunction before few days before the interview. Should just take a few minutes to complete this process, however, if there is an issue. At least, there are still a few days for me to resolve the issue before the final assignment interview.

Potential drawback to comply with the plan:

- **Collision with other assignments:**
  As I am aware this assignment is due at the end of week 14. During this time, there are many other assignments both individual and group assignment. Therefore, time organization can be a potential impact on the progression of the development. Which can cause a slightly delayed in the original timeline plan. And therefore, needing to do minor adjustments to the plan. Such as devoting more time to this project during the mid-semester break. Or using free time such as weekends, to keep up with the project's progression.

- **Potential health-related issues:**
  Since this is a solo project if I am sick today the progress of the development will be stopped completely until fully recovered or am at least able to continue the planned task. Therefore, having this expectation is crucial. Or if not sick, this can also be potential burnout that can be the drawback of the development progression. As burnout might impact the quality of work and the motivation of work. To prevent this to occur is to ensure the quality of resting and enough outdoor time, to keep both mentally and physically healthy.

- **The capability of problem-solving:**
  As a person who just started his programming journey a year ago. I think it is reasonable for me to state my experience in problem-solving and debugging ability is rather low. Especially, this is my first time developing a project purely by myself. As such, the potential issue can be the time I have taken to resolve any problem I have encountered throughout the entire project. Such as debugging, figuring out what functions to use, and integrating new technologies into existing code so on. Therefore, consumes more time to progress.

- **Timeline estimates infeasible:**
  As I previously stated, I am quite new to programming. I might have a false estimation of how long a larger project like this project will take due to inexperience. Leading to unrealistic expectations further causing issues to complete this project. Which can be costly if I didn't organize everything well. Having this said, using planning tools like smartSheet. Consistently monitor every week's progression and adjust if needed.