TraceAlert

Abstract

TraceAlert is an iOS application that aims to aid users, providing an alert when high-risk contacts are shown in the contact tree. This iOS application tracks users' contacts through mobile devices, reports total numbers of current contacts in different risks, and notifies users of high-risk contact information. This iOS application can also record users' contact information in the past 14 days to create a contact tree. We hope that this application can help users to safely travel during COVID-19 epidemic and control the spread of virus transmission, making Australia a safer society.

1 Introduction

The World Health Organization states that more than 100 million COVID-19 cases are confirmed after the global outbreak and around 30,000 cases are reported in Australia and the society has undergone huge life and property loss (Home, 2021). Discovering and blocking the chain of virus transmission plays a crucial role in relieving negative impact of COVID-19. In order to help stop the spread of COVID-19, our team would like to develop an assistant tool emulating the Covid tracer app to improve the safety of the users and information transparency of the current disease situation.

The South Australian Government has introduced a tracking QR code, COVID SAfe Check-In, to increase contact tracing, aiming to stop the spread of COVID-19 and enhance community safety. Although this application enables SA Health to immediately inform others who may have been exposed to the virus if they have been in close contact with COVID-19 positive cases, the users could not gain the bigger picture of the overall circumstances of the current outbreak. That is, COVID SAfe Check-In only provides a unilateral channel that merely allows users to upload their current locations and personal information without notifying them the current numbers of confirmed or suspected cases. The users still need to visit the government website to gain the latest information. In addition, users could not view their real-time numbers of contacts and determine whether their tracks are safe enough in the past two weeks. The users can only wait for the notification passively and this situation delays the golden time of self-quarantine or performing nucleic tests to discover cases and stop the spread of potential transmission. In order to solve the above problems, our group would like to develop an iOS application, which accounts for 52.84% share of the mobile operating system market in Australia in August 2020, instead of a website with QR code to fulfill the need of combining user report functions and system notification with contact trees and local COVID information displayed in one application (Granwal, 2021). The niche of our lightweight mobile application is to provide users with bidirectional interactions without switching between applications and multiple websites so that users could report cases for themselves or acquaintances, receive total numbers of current contacts with different risk levels and be alerted when high-risk contacts appear in the contact tree.

The fundamental objective of this application is to reduce the spread of COVID-19 for the community. This application consists of four main pages: Home, Declaration, Contact and Me. The application will be able to gain numbers of six different risks with detailed descriptions and show on the “Home” page. The application will allow users to report cases on the “Declaration” page. The application will show locations as a map and contact timing on the “Contact” page. The application will issue an alert, and display a contact tree, information of user inputs, language setting and outsources on the “Me” page. All the design will be easy to use and convenient to access by the users, and the data will be securely protected. Therefore, the application could cover most of the defects of the COVID SAfe Check-In by simply manipulating this application and making the community safer.

The remainder of this report will be laid out as four sections. In the Project Aims section, I will describe the goals of this project in detail and what this project is aimed to achieve. In the Approach section, I will explain the ideas, processes, approaches and tools that I used to meet these goals. In the Results section, I will discuss the achievements of this project with my personal contributions, what goals of this project have been met and useful lessons that I have learned from the development process. In the Conclusion section, I will conclude with a brief summary of the project outcomes, possible suggestions to improve the contribution and extensions of the future work.

2 Project Aims

The project aim is to provide a similar application of COVID tracker for users, halting the spread of COVID. The project will build an iOS application and can be accessed publicly on the internet. This lightweight application will track users' contacts in real-time with current contacts reported in different risk levels and notify users by an alarm if high-risk contacts are shown in the contact tree. To achieve the aims of the project, the iOS application should include the following core features:

1. Display total numbers of cases in different risk levels
2. Display a contact tree
3. Issue an alert when necessary
4. Track users' locations with a map and contact timing with a chart
5. Gather and display user inputs and information securely
6. User-friendly design

Extra features that could enhance user experience are also contained:

1. Declare cases from a form
2. Language selection and outsources links

3 Approach

3.1 Methodology

- The user launches the app; depending on the user status, new users will sign up through the registration form by inputting 6 pieces of personal information and existing users will log in with enrolled username and password
- If the login is successful, the user will be directed to the "Home" page, and the user can view numbers and detailed descriptions of 6 different categories: confirmed cases, suspected cases, close contact, high risk, low risk and healthy; if the login is failure, the "Login" page will display an error message, requiring the user to input the username and password again
- The user can report a case for himself/herself and others by submitting declaration forms with several essential information filled or view the submission records at the "Declaration" page
- The user can view a line chart of contacts with 3 different risk levels, high risk, low risk and healthy, in the past two weeks at the "Contact" page if he/she allows this application to locate and track his/her location and time
- The user can view his/her personal information, current language and contact tree, receive alerts under high-risk situation, and seek for external help at the "Me" page
- The user can repeat the above processes through clicking on the tab bar and the application will switch among pages according to corresponding icons. As a result, the user can check the latest information of each classification

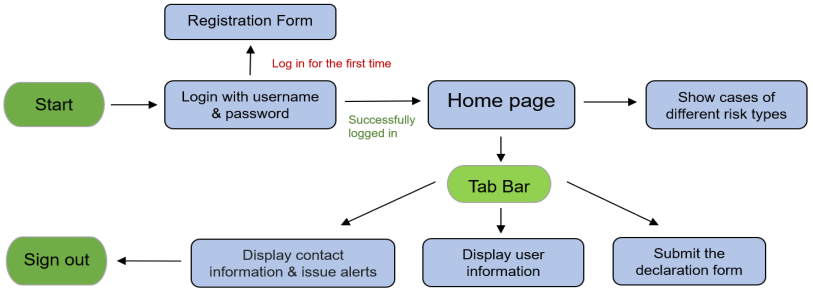


Figure 1. Key concept of the application

3.2 Tools and Software

Tools and Softwares	Brief Descriptions
Swift (5.3)	A powerful and intuitive programming language for iOS development

Xcode IDE (Version 12.4)	A productive environment for user interface design, coding, testing, and debugging of iOS development (Apple Developer Documentation, 2021)
DB Browser for SQLite (Version 3.12.2)	A lightweight and visual software to creatie, design and manipulate database files compatible with SQLite (DB Browser for SQLite, 2021)

3.3 Development Schedule

After the preparation phase, we separate the project into two major iterations: development and testing. After each iteration, we reviewed outcomes and reflected on defects for further improvement. Two major milestones of this project are shown in the appendix. The first milestone, completion of all non-functional development and part of the functional development of the application, is chosen to provide developers a basic structure to further develop major functionalities. The second milestone, completion of the database, optimization of all functions, and the testing and verification of the whole application, is chosen to deliver the project to real users, achieving the project goals.

3.4 System Architecture

Figure 2 illustrates three major components of this application. The front-end of the application is written by Swift and the user interface is presented on the iPhone. The mobile phone could store user data securely by encryption and send the data from this front-end interface to the back-end database. The user could browse each functional page through tapping the corresponding icons and also switch to another language by changing the setting of the mobile device. Two critical functionalities, contact tree and alert, are also demonstrated in the front-end. The back-end of the application is SQLite. This database is involved in registering and reading user information, providing the user with a daily contact table, contact time and contact tree. The third party API, apple maps, is used to position the user location if the user allows this application to track his/her location.

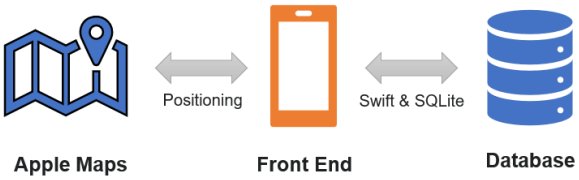


Figure 2. System architecture illustration

3.5 Contributions

Contributions	Task Description
1. Registration form	I added 3 text fields to gain more comprehensive user information and adjust the layout including the icon position, the margin size among user inputs, and “Register” button size so that this configuration can fit in either the small or big screen of the iphone for further real machine testing.
2. Login page	I added a slogan under our application icon based on the requirement report and adjust the layout including the position of icon and title, the margin size between inputs of username and password, and “Login” button size so that this configuration can fit in either small or big screen of the iphone for further real machine test.
3. Data security	Privacy is one of our main concerns so I implemented an encryption function for password by hashing user input with SHA256 and changed the type of password input from “TextField” to “SecureField” at the registration page, protecting the user input data and privacy. I also used the environment presentation mode functioning while

	clicking the "register" button, to pass the encrypted string to the login page so that this page can identify whether the user typed in the correct password or not. For testing purposes, I printed out the “hashString” to check if the hash function has functioned properly.
4. Database development	I initially implemented the draft structure of the database through Core Data and SQLite. After discussion, the group determined to apply SQLite so that I further developed the “bridgehead.h”, “DBHelper.swift” and “ViewController.swift”, allowing this application to connect to the database and manipulate data through SQL queries. Another teammate was responsible for the construction of the tables inside the database and displayed them.
5. User information storage and display	After the establishment of the environment presentation mode, I stored the user input information in the “userinfoDetail.swift” so that the information can be further displayed at the “Me” page. The demonstration function was successful and the screen indeed showed the personal information at the “Me” page, however, a memory warning occurred. To avoid affecting the working version of the current repository, I retrieved commits and stored the information to the “userinfo.swift” from the previously established database.
6. Adjust the layout of the “Home” page	I adjusted the layout by positioning the icon, title and text fields, and deleting the redundant “current situation” button shown at figure 1 to fulfill the concept of the lightweight application.
7. Adjust the layout of the “Declaration” page	I adjusted the layout by positioning the icon, title, text fields, and buttons shown at figure 1 to fulfill the concept of the lightweight application.
8. Apple Map	I added a map pin from MapKit so that the current location could be explicitly shown on Apple Map with an annotation.
9. Develop Alert, Help Center, Language and optimize Contact Tree, Information functions of the “Me” page	I developed the fundamental structure of the Alert button so that users can click on it to view the important message if it popped up. I changed the layout of the Contact Tree button so that the inside contents could fit in a smaller iphone. I built a connection between user inputs and the content of the Information button, however, due to the error mentioned in point 5, the final prototype did not include this connection function. I developed the Language button through NSLocalizedString() and added translations in the language specific Localizable.strings file so that this application can display the language based on the user's iPhone setting. I built the Help Center button for users to outsource through phone, website and email links.
10. Reorganize additional billing function	I added functions to display the balance of current account and payment methods for future extensions.
11. Defect, unit, integration and release testing	Defect testing was implemented every time when I submitted the code to GitHub repositories, ensuring that the program can be compiled and run successfully. Unit testing was performed under the inbuilt XCTest framework and after each completeness of the main page. Integration and release testing contains UI and functional tests. I stimulated real users’ operations through the Xcode Simulator to measure five main functions in this application: login, registration, contact tree, alert and positioning functions.
12. Documentation	The business case & draft plan, requirement report, testing plan, milestone plan and report, and poster were a combination of group

	work. I wrote agendas, minutes, README.md, and GitHub Projects individually and proofread all documents in pairs before submission.
13. Presentation and slides in the supervisor meeting and the Demo	The pitch and demo slides were a combination of group work. I presented with the supervisor during milestone 1 and played a role as facilitator to ask questions during milestone 2.
14. Gather user feedback	I asked friends about their user experiences of the application and made changes based on their opinions. Most feedback was about the layout of the application. A friend with a UI design background suggested that I consider color minimisation due to the concept of lightweight application and emphasize different visuals according to their priorities through different colors. Consequently, important functionalities were outlined in color red for easy recognition. Finally, the latest version of the application was under several updates.

3.6 Testing

Tests	**Outputs**
Defect testing	Successfully compiled, built and run at the Xcode Simulator
iPhone layout test	Successfully displayed all contents in the corresponding positions in either small (iPhone SE2) or large (iPhone 11) screens
Registration and login test	Successfully filled in the registration form and logged in through the newly registered username and password
User information display test	Successfully displayed the user information
Map display test	Successfully displayed the map with a red pin
Alert issuing test	Successfully displayed an alert at the “Me” page with the change of the background color after tapping
Contact tree display test	Successfully displayed a line chart with number of contacts and a table with 3 different risk levels

4 Results

4.1 Achievements

Achieved Required Features & Functionalities	**Description**
Display the contact tree and issue an alarm for users when a close contact has a high contact number	Same as planned
Store users’ data securely by encryption	Same as planned
Display users’ numbers of contacts in the past two weeks as a line chart	Same as planned
Achieved Additional Features & Functionalities	**Description**
Change languages of the application	Same as planned
Declare cases for myself or others	Same as planned

Display personal information from the user input	Same as planned
Display different risk levels with numbers and detailed descriptions	Same as planned

4.2 Application Demonstration

Figure 3 demonstrates the actual screenshots of the latest version of this application. Compared to the initial design (Appendix 1), four main pages still exist but in different layouts. These layouts allow users to easily switch among pages to view information that they are interested in. This lightweight design remains all essential functionalities mentioned in the project aims with enhanced efficiency. Additionally, each functionality can be accessed by simply clicking on the corresponding buttons at each page. Through this application, users can figure out crucial information such as a contact tree, an alarm and numbers of contacts at “Contact” and “Me” page, as well as additional information of risks and declaration forms at “Home” and “Declaration” page.

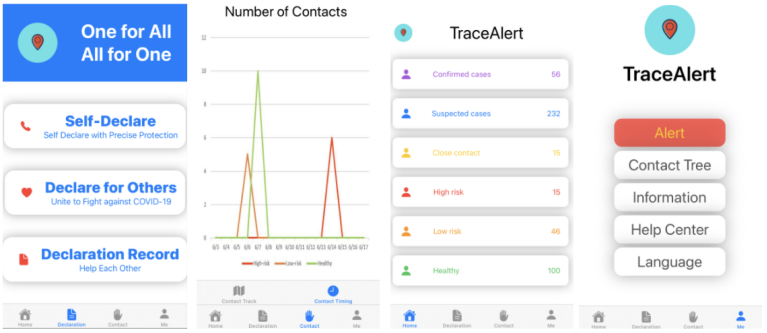
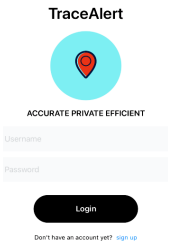
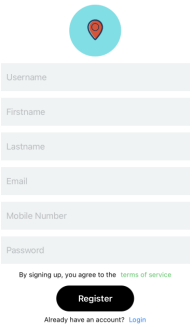
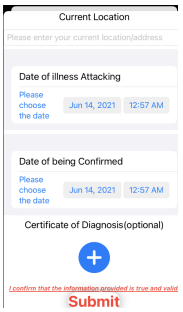
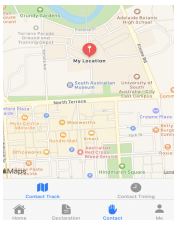
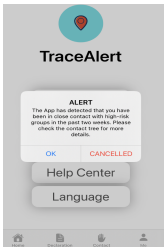
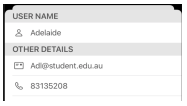
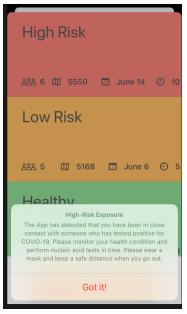
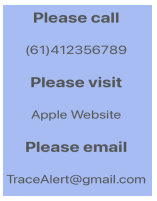


Figure 3. Final design of the application

Buttons	Descriptions	Layouts
Login Page	This page is displayed when the user clicks on the application icon in their iPhone. The user can then login with their existing username and password by the “Login” button. If the corrected pair of username and password is entered, the user is directed to the “Home” page with numbers of risk levels shown. If the user types in the wrong pairs, an error message is displayed so that the user can be notified to input the username and password again. The user can also create a new account by the “sign up” link and is directed to the “Register” page for further operation.	
Registration Page	This page is displayed when the user clicks on the “sign up” link from the “Login” page. The user is required to fill in information including username, first name, last name, email, mobile phone and password in this registration form and submit by “Register” button to create a new account. Once this button is clicked, the password is encrypted as figure 4 and the user is redirected to the “Login” page. If the user accidentally enters this page, he/she can return back to the previous page by the “Login” link.	

```
SHA256 digest:
    b3a8e0e1f9ab1bfe3a36f231f676f78bb30a519d2b21e6c530c0
    eee8ebb4a5d0
```

Figure 4. Password encryption

Self-Declare & Declare for Others Button (Declaration Page)	This page is displayed when the user clicks on the “self-Declare” or “Declare for Others” button at the “Declaration” page. The user is suggested to fill in information such as current location and dates with certification to report cases. Once the “Submit” button is clicked, the function of passing inputs to the database for record will be further developed.	
Contact Track Index (Contact Page)	This page is displayed when the user clicks on the “Contact” page. The user can view the location with a red pin and annotation if he/she agrees on sharing locations of his/her mobile device with this application.	
Alert Button (Me Page)	This page is displayed when the user clicks on the “Alert” button at the “Me page”. The user can view a warning message issued from this application when this button is on a red background. After responding either “OK” or “CANCELLED” to the pop up message, the background turns white, representing that the user is notified by the alert message.	
Information Button (Me Page)	This page is displayed when the user clicks on the “Information” button at the “Me page”. The user can view their personal information within a table.	
Contact Tree Button (Me Page)	This page is displayed when the user clicks on the “Contact Tree” button at the “Me page”. The user can clearly view information including numbers of contacts, locations of contact provided with district code, contact dates, and minutes of contact length under the categories of three different risks, high risk, low risk, and healthy, with three corresponding colors, red, orange and green. Additionally, the user can click on the “Important” button and another warning message is popped up to notify the user to be aware of recent contacts. The above message disappeared when the user taps the “Got it!” button.	
Help Center Button (Me Page)	This page is displayed when the user clicks on the “Help Center” button at the “Me page”. The user can reach out resources through links of the phone, website, and email.	
Language Setting	This page is displayed when the user reenters this application after changing the language setting from English to Simplified Chinese at iPhone settings. The user can view most of the information in Simplified Chinese, representing that this application enables the delivery of different selections that are suitable for global users after further development in language extension functions.	

4.3 Personal Learnings, Achievements and Reflections

Categories	Descriptions, Achievements & Reflections
Swift	Because I am responsible for the front-end, I spent a month getting familiar with Swift development and framework from online tutorials. Swift is a powerful and intuitive language so my previous coding experience helps me to get acquainted with this language quickly. Additionally, I found that directly implementing a project is the most efficient and effective way to learn a new coding language. During the development, I built pages with different functions, connected the front-end user data to the back-end database, cultivated my debugging skills through line by line testing and enhanced my problem-solving skills through StackOverflow and official Apple developer documentation (Apple Developer Documentation, 2021). After this project, I have the confidence to develop another application in Swift.
Apple Map API	This is my first-time using the Apple Map API, MapKit, to display a map within this application. I have learned a lot through implementing this service and recognized that it is convenient for developers to use third-party kits to build or simplify complex functions. These services increased code reusability and saved lots of time for developers because they did not need to reproduce repetitive code, boosting the productivity of developers. After this project, I have the knowledge to implement different APIs in future projects.
Database	I learned basic SQL syntax in my previous project, however, I only coded the SQL queries of the front-end in this project so that my teammate, who was responsible for building tables and retrieving useful data in the database, can easily manipulate data between this application and database. I learned more about how the database works in regard to construction and operation from my teammate. I am definitely going to educate myself further with this useful and powerful tool.
Project Management	Before actually coding the functionalities, confirming user requirements, project schedule, roles and responsibilities of each group mate, and development milestones are essential works to make this project successful in terms of project scope, time and cost management. Our team adapts agile development in this project to meet the above definition of project success. In each iteration, our group develops iterable functions and fixes them based on user feedback after weekly meetings with the supervisor. Additionally, becoming the scrum master trains my presentation skills and facilitates me to articulate our progress in the meeting and our application to the audience. The GitHub Projects and Gnatt are also great project management tools to track user backlogs and to do lists so that our development will not lose track and miss deadlines. Through writing minutes and agendas, I can easily track what functions required to be amended in this application, reflect on what can be improved in the next presentation and directly discuss the critical points during the meeting due to the advanced notification sent to the supervisor. Consequently, I figure out that smooth communication among teammates and with the supervisor is the most vital takeaway from this project. Once any difficulty occurs, smooth communication between groupmates can solve emergencies immediately and the supervisor can offer resources to handle them. In sum, cooperation as a team with proper communication makes the project successful.

4.4 Problems Encountered

Problems	Solutions
Fail to setup development	Due to the issue of linking GitHub enterprise with Xcode, we used GitHub desktop to solve this problem. All development could be functioned through git commands

environments	or GUI after setting the correct username and password in GitHub enterprise.
Update UserInterfaceState.xcuserstate file error at GitHub Desktop	The code fails to build due to the issue of UserInterfaceState xcuserstate file. From the StackOverflow, we knew that removing this file by the git rm --cached *xcuserstate command did not affect the functionalities of the project because they are meant to save GUI states of Xcode (Cannot Merge due to conflict with UserInterfaceState.xcuserstate, 2021). We decided to discard this file and ignored .DS_Store to prevent errors for future submissions.
Unable to store user data in Core Data	After building the Core Data in our application, we found that an unknown error prevented us from saving user data in the database. After discussing with the supervisor, he suggested that we try to implement another database if this error still existed or simply pass the data as an array to pages and display each element. We decided to adapt both suggestions, using DB Browser and sending all user input data as arrays to the information display pages.
Project management inside the group	Due to the geographical constraints among group mates, we planned a communication plan to schedule regular meetings online and record all discussed issues to increase efficiency by eliminating time waste on duplicate questions. We also strictly followed the milestones and Gnatt timelines to ensure that all tasks can be completed on time.

4.5 Project Extension

Extensions	**Descriptions**
Modify user information in-app	Current functions do not support in-app modification of user information. It will be inconvenient for users to manage their own data.
Apply this application with other platforms and devices	Current application only supports iOS devices. To increase the usage population and maximize the value of this application, an Android version will be further developed embedded with QR code, Bluetooth or random ID technology to perform precise tracking.
Authenticate submissions	Current application only records user declarations without further inspection. It will be insecure to trust shared data without checking.
Display different UI	Current application only displays one UI layout. From the user feedback, it will be better for the user to switch the screen to dark mode if possible.
Publish this application in the Apple Store	Current application only functioned under Xcode Simulator due to unauthoritative setting of Apple developer account. The group will further register a valid to deploy the application in the Apple Store.

5 Conclusion

I would like to first appreciate our supervisor Nick Falkner for his suggestions and guidance throughout this project. Professor Nick Falkner provided us with lots of useful resources, technologies and insights during our development process. I would also like to give the thumbs up to my group mates due to the effort made to make this project successful.

In conclusion, this application has achieved most of the required functionalities with some advanced features. This application is operated in the Xcode Simulator and enables users to receive alerts and numbers of contacts in the contact tree. The additional functions including user sign up, user login, user information display, language changing, display of risk levels, declaration forms are also presented in this application. The future work will include further optimization of current and new functionalities with UI improvements and publish this application in the Apple Store.

References

Developer.apple.com. 2021. Apple Developer Documentation. [online] Available at: <https://developer.apple.com/documentation/swift> [Accessed 15 June 2021].

Stack Overflow. 2021. Cannot Merge due to conflict with UserInterfaceState.xcuserstate. [online] Available at: <https://stackoverflow.com/questions/13870652/cannot-merge-due-to-conflict-with-userinterfacestate-xcuserstate> [Accessed 15 June 2021].

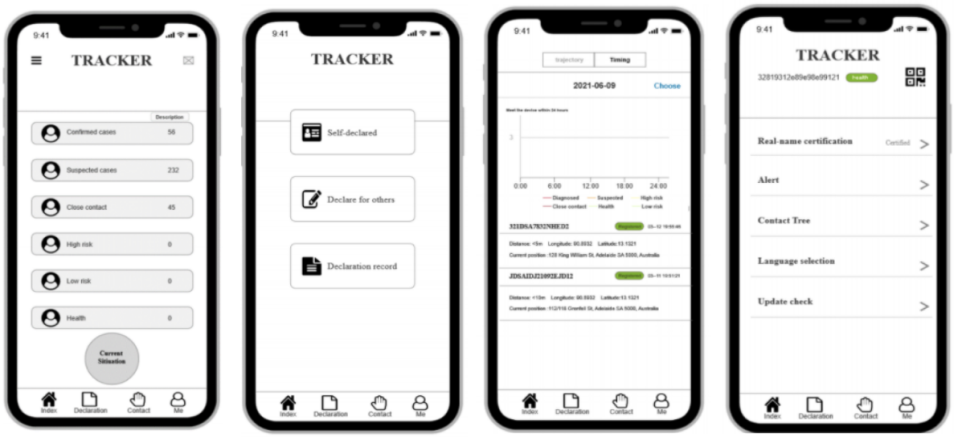
Sqlitebrowser.org. 2021. DB Browser for SQLite. [online] Available at: <https://sqlitebrowser.org/> [Accessed 15 June 2021].

Granwal, L., 2021. Australia - mobile OS share 2020 | Statista. [online] Statista. Available at: <https://www.statista.com/statistics/861532/australia-mobile-os-share/> [Accessed 15 June 2021].

Who.int. 2021. Home. [online] Available at: <https://www.who.int/> [Accessed 15 June 2021].

Appendix

1. Initial design of the application



2. Project schedule with Gnatt chart

At Risk	Task Name	Start Date	End Date	Duration	% Complete
	Preparation	03/03/21	04/02/21	23d	100%
	Kick Off	03/03/21	03/05/21	3d	100%
	Demand Analysis	03/06/21	03/12/21	6d	100%
	UI Design & Confirmation	03/13/21	03/19/21	6d	100%
	Pitch Presentation Delivery	03/20/21	03/31/21	9d	100%
	Business Case & Draft Plan Delivery	03/26/21	04/02/21	6d	100%
	Development	**04/03/21**	**05/14/21**	**31d**	**0%**
	Service Architecture Debugging & Construction	04/03/21	04/06/21	3d	0%
	APP Process Interfaces development (Non-functional)	04/07/21	04/20/21	10d	0%
	Milestone 1 Report Delivery	04/21/21	04/30/21	8d	0%
	Main Function Development (iOS Front-end)	04/21/21	05/07/21	13d	0%
	Back-end Database Development	05/08/21	05/14/21	6d	0%
	Final Milestone Plan	05/07/21	05/14/21	6d	0%
	Test & Acceptance	**05/07/21**	**06/15/21**	**28d**	**0%**
	Testing Plan	05/07/21	05/14/21	6d	0%
	Front-end & Back-end Joint Debugging Test	05/15/21	05/18/21	3d	0%
	Requirements Function Verification & Validation	05/19/21	05/28/21	8d	0%
	Project Acceptance	05/29/21	06/15/21	13d	0%
	Final Presentation	05/29/21	06/08/21	8d	0%
	Final Report	06/01/21	06/15/21	11d	0%

3. Milestone schedule

First Milestone (Week1-Mid Break)	Final Milestone (Mid Break-Week12)
Construct development environment in Xcode	Set-up SQLite database
Debug the service architecture of the application	Verification and validation of user requirements
Build the basic layout of all application pages including routing among pages through the tab bar icons	Develop and optimize all main functions and layouts including an alert, a contact tree, an user tracking map and a table of current cases
Build user registration, login and positioning functions	Debug and test for all functionalities through the Xcode simulator

4. Final Testing Results

Test Description	Input	Expected Output	Current Output
Defect testing	Compiled, built and run the application at the Xcode Simulator	Successfully compiled, built and run at the Xcode Simulator	Same as expected
Date: 07/06/2021		Responsible: All team members	

Test Description	Input	Expected Output	Current Output
Power blackout test	Rebuilt the Xcode Simulator after interruption	Successfully refreshed the Xcode Simulator to function as normal	Same as expected
Date: 07/06/2021		Responsible: All team members	

Test Description	Input	Expected Output	Current Output
Internet test	Turn off the internet of the mobile device and turn back on	Successfully reconnected to the internet	Same as expected
Date: 07/06/2021		Responsible: All team members	

Test Description	Input	Expected Output	Current Output
iPhone layout test	Access the application with large and small screens on different devices	Successfully displayed all contents in the corresponding positions in either small (iPhone SE2) or large (iPhone 11) screens through Xcode Simulator	Same as expected
Date: 07/06/2021		Responsible: All team members	

Test Description	Input	Expected Output	Current Output
Real iPhone test	Use real iPhones to	Successfully linked the	Errors due to

	operate the application	Xcode Simulator and iPhones to mimic the application operation	incompatible OS version with XCode version, testing only available on Xcode Simulator
Date: 07/06/2021		Responsible: All team members	

Test Description	**Input**	**Expected Output**	**Current Output**
Registration and login test	Filled in the registration form and logged in the application	Successfully filled in the registration form and logged in through the newly registered username and password	Same as expected
Date: 07/06/2021		Responsible: All team members	

Test Description	**Input**	**Expected Output**	**Current Output**
User information display test	Tapped the “Information” button of the “Me” page to view the content	Successfully displayed the user information after clicking the “Information” button of the “Me” page	Same as expected
Date: 07/06/2021		Responsible: All team members	

Test Description	**Input**	**Expected Output**	**Current Output**
Declaration form test	Typed in essential information listed on the declaration forms after clicking different buttons	Successfully directed to corresponding sheets after clicking different buttons and allowed user input data	Same as expected
Date: 07/06/2021		Responsible: All team members	

Test Description	**Input**	**Expected Output**	**Current Output**
Language changing test	Changed the language setting of the Xcode Simulator	Successfully changed language from English to Simplified Chinese	Same as expected
Date: 07/06/2021		Responsible: All team members	

Test Description	**Input**	**Expected Output**	**Current Output**
Map display test	Allowed the Apple Map to locate the current operation and viewed the “Contact” page	Successfully displayed the map with a red pin if the users allowed Apple Map to locate his/her location	Same as expected
Date: 07/06/2021		Responsible: All team members	

Test Description	**Input**	**Expected Output**	**Current Output**
Alert issuing test	Viewed the “Me” page and pressed the “Alert”	Successfully displayed an alert at the “Me” page	Same as expected

	button	with the change of the background color after tapping	
Date: 07/06/2021		Responsible: All team members	

Test Description	**Input**	**Expected Output**	**Current Output**
Contact tree display test	Viewed the “Contact” page with the “Contact Timing” clicked and switched to the “Contact Tree” button of the “Me” page	Successfully displayed a line chart with number of contacts at the “Contact” page and a table with 3 different risks at the “Contact Tree” button of the “Me” page	Same as expected
Date: 07/06/2021		Responsible: All team members	