A Mobile Application to help the Elderly to avoid Telephone Deception

Final Report

Ng Sze Ho Raymond 3035601499

Supervised by Dr. Chim T. W.

Date of submission: Apr 18th, 2022
Abstract

A surge in telephone deception cases in Hong Kong has been found with observable common trends. Firstly, old tricks like “Pretend Official” are still highly effective in Hong Kong, contributing to over half of the local telephone deception cases in 2020. Secondly, elderlies are usually involved in telephone deception cases with monetary loss. Although the government has organized campaigns to tackle telephone deception, citizens are still falling for old-fashioned scams. In this project, we create a localized application to tackle and educate users about telephone deception, with elderlies and their family members as our target audience. A main application for the elderly and a companion application for family members will be delivered at the end of this project. The project team make use of React Native and a local phone number database to create native Android applications for identifying malicious calls, paired with location-based features implemented with the help of Google Maps. The project was a success, with both the main application and the companion application ready to be deployed. Project details and deliverables can be found on https://wp.cs.hku.hk/fyp21027/.
Acknowledgements

I would like to show my gratitude for the Department of Computer Science, and the supervisor of this project, Dr. Chim T. W., for providing me a chance to work on this meaningful project. The continuous support and valuable suggestions from Dr. Chim are much appreciated.

I would also like to acknowledge my teammate Ng Wo Tai Walter. Walter sacrificed lots of time to discuss the details of the project with me, as well as implementing and improving our applications. Without his help the project would not be as polished as it is.

Furthermore, I would also like to acknowledge Dr. Anthony Tam, the second examiner, for his insight towards the expansion of our project. The suggestion from Dr. Tam shows me another path of implementing this project.

Last but not least, I would like to pay tribute to Miss Mable Choi from CAES for her support towards this report.
## Table of Contents

- Abstract .................................................................................................................. 2
- Acknowledgements ................................................................................................. 3
- List of Figures .......................................................................................................... 6
- Abbreviations and Acronym ....................................................................................... 8

1. Introduction
   - 1.1 Background & Motivation
     - 1.1.1 Overview of Telephone Deception in Hong Kong ........................................... 9
     - 1.1.2 Elderlies and Telephone Deception ............................................................... 10
     - 1.1.3 Existing Call Blockers .................................................................................. 11
     - 1.1.4 Flaws of existing Call Blockers .................................................................... 12
   - 1.2 Objective & Deliverables .................................................................................. 13
   - 1.3 Outline of the Report ....................................................................................... 13

2. Methodology
   - 2.1 Introduction ..................................................................................................... 14
   - 2.2 Platform ........................................................................................................... 14
   - 2.3 External Resources
     - 2.3.1 HKJunkCall Database ................................................................................ 14
     - 2.3.2 Google Maps API ....................................................................................... 15
     - 2.3.3 Firebase ..................................................................................................... 15
   - 2.4 Features ............................................................................................................ 16
   - 2.5 User Experience & UI Design ......................................................................... 16
   - 2.6 Summary ......................................................................................................... 18

3. Results & Findings
   - 3.1 Overview ......................................................................................................... 19
   - 3.2 Project deliverables & Considerations
     - 3.2.1 Flaws of existing call blockers ................................................................... 19
     - 3.2.2 Senior-focused Application ....................................................................... 22
     - 3.2.3 Our Implementation .................................................................................... 24
     - 3.2.4 Data exchange between the two applications .......................................... 34
   - 3.3 Challenges & Difficulties .............................................................................. 37
   - 3.4 Future Plan ..................................................................................................... 42

4. Conclusion .............................................................................................................. 43
List of Figures

1.1 Local Deception and Telephone Deception statistics
1.2 The greatest phone scam in Hong Kong’s history
1.3 Call Defender features (1)
1.4 Call Defender features (2)
1.5 Jima Caller call blocking customization interface
2.1 Structure of HKJunkCall’s URL
2.2 Examples of Glyph Icons
2.3 Navigation menu with glyph icons
2.4 Navigation menu without glyph icons
3.1 Screenshot from Call Defender
3.2 Screenshot from Jima Caller
3.3 Screenshot of smartphone with “Easy mode” enabled
3.4 Screenshot from Senior Citizen Card Scheme
3.5 Navigation bar of the applications
3.6 Locations page of main application
3.7 Location pin on the map
3.8 Call logs page of main application
3.9 Phone number checking notification
3.10 Call logs display
3.11 Persistent notification of main application
3.12 Connections page of main application
3.13 Connections page of companion application
3.14 Locations page of companion application
3.15 Call logs page of companion application
3.16 Persistent notification of companion application
3.17 Notification for recent malicious calls
3.18 Database structure
3.19 Sample record on Firebase
3.20 Call log checking algorithm
3.21 Web scraping on HKJunkCall
3.20 HKJunkCall call reporting interface
### Abbreviations and Acronym

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>UX</td>
<td>User Experience</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Background & Motivation

1.1.1 Overview of Telephone Deception in Hong Kong

Telephone deception refers to malicious phone calls made by criminals that tend to make similar promises and threats or ask you to pay in certain ways [1]. Due to COVID-19 pandemic, extensive use of internet communication makes people more susceptible to deception. With telephone deception being laborious to track and easy to accomplish, we can see a surge in deception attempts.

![Number of Deception and Telephone Deception cases in 2016 - 2020](image)

**Fig. 1.1** Local Deception and Telephone Deception statistics

The above figure shows the statistics of deception cases in Hong Kong during 2016 – 2020 (Fig. 1.1). Both deception and telephone deception cases saw a huge increase during 2020, possibly due to pandemic making people more vulnerable. According to telephone deception statistics, an 84.1% increase is found during 2019 – 2020 [2]. Considering telephone deception going viral in recent years, more attention should be paid towards telephone deception.
1.1.2 Elderlies and Telephone Deception

With telephone deception being the focus, further investigation has been made to identify the trend and the age group of victims. Even with the campaigns hosted by The Hong Kong Police Force and the Anti-Deception Co-ordination Centre (ADCC), “Pretend Officials” and “Guess Who” are still prevalent [2].

Since some of the senior citizens do not have access or the knowledge to access anti-fraud materials, elderlies are the most vulnerable group of people towards telephone deception, and the largest scale telephone deception was also found within this age group.

Hong Kong / Law and Crime

90-year-old woman living in mansion on The Peak conned out of US$32 million in Hong Kong’s biggest phone scam

- Woman received call from fraudsters who said her identity had been used in crime in mainland China and they needed to verify origins of her money
- She followed their instructions and transferred HK$254.9 million into three bank accounts before realising she was duped

In April 2021, a new record was reached for telephone deception in Hong Kong (Fig. 1.2). Criminals pretend to be Mainland Officials and request elderly women to transfer money for verification. Nearly HK$255 million was lost in a single telephone deception case, with the modus operandi of “Pretend Officials”.

Elderlies being targeted by criminals for telephone deception cannot be undone. Young citizens who notice this trend will try to adopt preventive measures for their older family members, such as installing call blocker applications for the elderly.
1.1.3 Existing Call Blockers

Whoscall, Call Defender and Jima Caller are three call blocker applications that feature similar functionalities and widely used in Hong Kong. Each of them has some unique aspects that separate them from each other.

Developed by Gogolook, Whoscall is a well-known international call blocker application. Being available for both iOS and Android, this application targets East Asia users and provides suspicious call detection. The application claims to have the largest database in East Asia with 1.6 billion phone number entries [4]. Whoscall has an emphasis on its own database and artificial intelligence (AI) to provide call blocks. While it is powerful, a custom blacklist/whitelist feature is a wanted feature. The free version does not have access to the premium database nor the database auto-update feature.

Call Defender is another application developed by Gogolook targeting Hong Kong users. This application makes use of databases from Whoscall and HKJunkCall for call blocking service [5].

![Manage Exception List](image)

**Fig. 1.3 Call Defender features (1)**

![Advanced Block Options](image)

**Fig. 1.4 Call Defender features (2)**

Call Defender has better customization on call blocking through custom exception lists (Fig. 1.3) and advanced blocking options (Fig. 1.4). An exception list allows users to set up their personal blacklist and whitelist for a bespoke experience. Users can also choose to block calls from certain categories or industries through advance blocking
options. While this makes Call Defender a superior choice for Hong Kong users over Whoscall, the lack of iOS support was its downside.

Jima Caller is a cross-platform call blocker application developed by a local company named Techmaxapp. This application make use of HKJunkCall database [6] to provide a localized call blocking experience.

![Jima Caller call blocking customization interface](image)

**Fig. 1.5** Jima Caller call blocking customization interface

Jima Caller offers similar features found in Call Defender, such as categorized call blocking and custom whitelist/ blacklist (Fig. 1.5). While Call Defender has a better user interface (UI), iOS support from Jima Caller allows iOS users to snoop into a localized call blocking experience.

While existing call blocker applications are available on the mainstream mobile operating systems (OS), such applications are not aimed towards the elderly.

### 1.1.4 Flaws of existing Call Blockers

Existing solutions such as Call Defender and Jima Caller are great call blockers, but the experience is horrendous when used by the elderly. As shown in Fig. 1.3 – 1.5, a text-heavy interface with small buttons is prominent for both applications. Extensive use of text in menus may overwhelm senior users, and small buttons could be a hassle during the configuration process. On top of that, all applications reviewed in this report do not have database auto-update in their free version, requiring elderlies to perform manual database update was not ideal.
1.2 Objective & Deliverables

In this project, a tailor-made application will be developed to tackle telephone deception and keeping the family members involved.

The main application is designed for the elderly. This application focuses on giving elderly information on calls they made or receive, as well as enabling the essential features of the companion application. The application can be used in conjunction with existing call blockers, but this application focuses on only giving them the necessary information and keep connected with the family members.

Family members will make use of the companion application to retrieve real-time information from the elderly. This application is mainly used for family members to keep track of calls received by the elderly, as well as providing assistance if needed.

The scope of this project is to educate users about telephone deception through the guidance of our application, but not to make judgements for the users.

1.3 Outline of the Report

This report is structured into four chapters. Chapter one offers an overview of local telephone deception trends, as well as reviewing existing call blocking solutions in the market. This chapter also annotates the goals of this project and the project deliverables.

Chapter two presents the methodology used in this project. The software framework used, features of the application, as well as external resources used to accomplish certain features will be explained. The user experience (UX) design will be covered as well.

Chapter three presents the findings and results of this project. This chapter will report the UX considerations and implementation on the project deliverables, as well as the challenges faced by the project team and the future of the application.

Chapter four serves to conclude this report.
2. Methodology

2.1 Introduction

This chapter explains the technologies used to implement the application, including the front-end coding framework and the back-end resources to achieve certain functionalities. The usability of the application among target users will be evaluated through investigating various UX designs.

2.2 Platform

A software framework that creates native applications will be used. React Native is our preferred software framework as this framework creates native applications for both Android [7], which ensures the performance of the deliverables. Though iOS version of the application was not developed, making use of React Native will allows us to port this application to iOS in the future with less hassle, as React Native creates native applications for iOS and the same codebase can be reused [7].

2.3 External Resources

2.3.1 HKJunkCall Database

HKJunkCall has the largest phone number database in Hong Kong. By making use of it, we could identify malicious calls easily through filtering call categories. However, HKJunkCall stopped accepting application programming interface (API) key applications back in 2017 due to different reasons [8].

Since accessing HKJunkCall with its API is not feasible, we have decided to use their website to accomplish call identification. HKJunkCall’s website allows users to input the phone number and search for results. Thus, we will make use of this feature and fetch the data to our application through web scraping.

Web scraping refers to extracting data through any means other than interacting with an API [9]. In order to query for a phone number, we can query this number directly by accessing HKJunkCall’s website and modify the Uniform Resource Locator (URL) accordingly.
Fig. 2.1 Structure of HKJunkCall’s URL

The above figure disassembles HKJunkCall’s URL structure (Fig. 2.1). Accessing “hkjunkcall.com/?” directs us to HKJunkCall’s home page. To query a phone number, “ft=” will be added to the URL, followed by the phone number “28592180” as an example. Accessing “hkjunkcall.com/?ft=28592180” shows the query result for that phone number, proving this phone number belongs to HKU CS Department. The same logic can be used for other phone numbers to deliver the fundamental feature of this application.

2.3.2 Google Maps API

Google Maps are known to be the most used mapping services around the world, sharing over 80% market share for mobile apps [10]. We believe integrating Google Maps in our application could help us to deliver reliable location-based features.

Google provides a free maps API for mobile [11], and the API has native support for JavaScript, the same programming language used by React Native. The access of Google Maps API will help us to implement the alert feature, which will be explained in the following chapter.

2.3.3 Firebase

Firebase is a platform owned by Google since 2014, and it is widely used for creating applications. Firebase includes lots of features such as a cloud database for apps, integrations with Google apps, and so much more [12].

Realtime Database is a database service provided by Firebase, and this application will make use of such service for the backend, to store user records for the connection between the main application and the companion application.
2.4 Features

As mentioned in Chapter 1.2, the application consists of two applications: a main application for the elderlies and a companion application for the family members.

The main application is designed to be a scam tracking application. Unlike existing solutions in the market, our application focuses on preventing telephone deception instead on preventing junk calls in general. Thus, the filtering will not filter out general junk calls, but only malicious calls.

To achieve scam tracking, contact book checking and database checking will be implemented. The application will read the 10 most recent calls from the call log and perform checking with HKJunkCall database if needed. Calls will be filtered to different types to give elderly an understanding on what calls they have made or receive.

The companion application is designed to provide more information for family members without checking elderlies’ phone directly. This application has access to the call log and location information of the elderly. The location-based features are not meant for tracking the elderly, but to enable family members to provide face-to-face assistance in urgent situations. On top of manual record checking, notifications will be delivered through the companion app once the elderly receives a malicious call.

2.5 User Experience & UI Design

Since this project aims to deliver a tailor-made experience for the elderly, ensuring the elderly can use the application was in our first priority. Thus, we have taken extra considerations when designing the UI and UX.

With reference to Occam’s razor [13], a simple solution is usually the best solution. With this in mind, we are keeping our application simple and lightweight. We only implement necessary features as bloating the application with unnecessary stuff violates simplicity. Keeping the features simple is just part of it, keeping the user interface simple and intuitive is crucial.
Fig. 2.2 shows examples of glyph icons. Glyph Icons are graphical symbols agreed to have certain meanings, and they can be used in conjunction with text to provide clear instructions. Use of glyph icons in UI are getting more and more common, and we believe our application could benefit by making use of it.

Above are screen captures from the Phone App in iOS 15 (Fig. 2.3) and Samsung’s One UI 3.1 (Fig. 2.4) respectively. iOS make use of glyph icons as the main indicator for different functions and preserved text descriptions. Intuitive glyph icons enable users to pick the feature they want without reading the assistive text. Samsung’s approach of omitting the glyph icons allows the navigation bar to take up less screen real estate, but users are required to read the text to pick the feature they want, potentially lengthening the time required to complete an action. In addition, recognizing icons is
easier than recognizing texts, since glyph icons have different shapes, and some UI may make use of specific words that was hard to understand by ordinary users. The use of glyph icons could contribute to quicker navigation within the application. Our application would follow a similar format when designing the navigation menu.

Combining the above practices should make the operations intuitive enough to feel like daily practices. We believe such implementation on UI and UX will be easy for the elderly to master.

2.6 Summary

This chapter explained the technologies required for the development of this tailor-made application. On top of the software framework and external resources, design considerations were explained to prove this application’s effectiveness towards elderlies. The following chapter will report our results on this project.
3. Results & Findings

3.1 Overview

This chapter reports the results and finding of the project. Chapter 3.2 provides a detailed report on the project deliverables and considerations taken, with Chapter 3.3 listing the challenges when working on this project. Chapter 3.4 shows the future path of the application development.

3.2 Project deliverables & Considerations

3.2.1 Flaws of existing call blockers

In the introductory chapter of this report, the project team has discussed why existing call blocking solutions do not fit the need of tackling telephone deception. In this chapter, we will discuss why such applications are not user friendly towards the elderly by reviewing their UI implementation.

Most call blocking applications adopt a text-based menu with multiple sub-menus under the same page. While this makes great use of screen real estate and allows the user to scroll less, such implementation was not ideal when designing an application for the elderlies.
Fig. 3.1 shows the settings interface from the application Call Defender. Nine menus can be found under the “Block” category. Tightly packed menus will be intimidating for elderlies who may not have much experience with complicated mobile applications. In addition, the menu item features technical terms without explanation, such as “Spam filter” and “Private Number”. Uncertainties regarding a toggle would cause frustration for the users, especially towards elderlies.

On top of a tight packed menu, most call blockers preferred small UI elements to make use of the screen real estate “at its finest”. We believe such implementation is also not user friendly towards elderlies.
Fig. 3.2 Screenshot from Jima Caller

Fig. 3.2 shows the settings interface from the application Jima Caller. Multiple buttons can be found and button legends like “identify” cannot even display properly due to the button being too small. In addition, assistive text featured in different menus have an absurdly small font size, which was tiny even when compared to the “not so big” fonts in the status bar. Inappropriately sized UI elements makes the UI hard to comprehend and navigate, and we believe such detrimental effects would be more prominent when the users are the elderly.

From these observations, we will avoid these design implementations when we set up the UX standard for our application.
3.2.2 Senior-focused Application

While the project team investigates existing call blockers and identifies UI implementations that should be avoided, we also investigated on senior-focused applications and learn from them.

Most modern smartphones feature a built-in “Easy mode” or “Elderly mode”. Enabling this changes many of the UI properties and makes your smartphone easier to use.

Fig. 3.3 Screenshot of smartphone with “Easy mode” enabled

Fig. 3.3 demonstrates the “Easy mode” interface. When compared to the normal interface, the UI is much simpler. “Easy mode” only shows the necessary and most used applications, showing them with enormous buttons. On top of that, essential UI elements such as battery meter and signal meter are also enlarged, keeping them easy to read. When designing an interface for the elderly, we just need to make sure they can access the necessary features and keep the UI as “stupid” as possible.
Other than smartphone UI designed for the elderly, there are also applications designed for them. Such applications adopted the same principle mentioned previously but giving it a slight twist when used in the UI of an application.

Fig. 3.4 Screenshot from Senior Citizen Card Scheme

Fig. 3.4 shows the interface of a senior-focused application developed by the Hong Kong Government. This page only has 3 large buttons, results in a less cramped interface and allows users to determine different items easily. Furthermore, large text and intuitive descriptions are used on the buttons to avoid misunderstandings.
Learning from these UI implementations, we believe keeping the UI simple and providing basic, yet necessary navigations, would be the key for creating an application with seniors in mind.

### 3.2.3 Our Implementation

With multiple observations made from reviewing different UI implementations, the project team have established a unified UX standard to be used on both the main application and the companion application. The two applications feature a similar user interface, but with different functionalities to split them apart.

Both the main application and the companion application split into 3 pages, which is the Locations (位置) page, Call logs (通話記錄) page and Connections (連結) page. Each page serves a different purpose to keep the features organized and avoid misunderstanding.

![Navigation bar of the applications](image)

**Fig. 3.5 Navigation bar of the applications**

Navigation is important for a multi-page application, so the project team have implemented a bottom navigation bar for both applications (Fig. 3.5). The persistent navigation bar allows users to access different pages at any time, and glyph icons are being used for quicker navigation, as explained in previous chapters. Another major design change is the use of Chinese for the UI elements. Since not every elderly in Hong Kong knows about English, using Chinese UI would allow them to get used to the application quicker.

The UX in general features less inputs from the user, so buttons should be rare to find in our application. Each page was only responsible for 1-2 features to keep things organized and keep unnecessary items away from the user. The main application features a “no-setup” design, allowing the elderly to utilize the application once they installed the application. Family members just need to input one code to use the companion application.

As mentioned above, our project consists of two applications, a main application for the elderly, and a companion application for the family members.
Fig. 3.6 shows the UI for the Locations page. This is a simple page that shows the current location of the user (the elderly himself), and it is the default page when one launches the App. A map pin is used to denote the current location, with the pin aligned to the middle of the interface.
Fig. 3.7 Location pin on the map

Tapping on the pin reveals additional information and shortcuts (Fig. 3.7). A brief description will show up when one taps on the pin, which was used to clarify what the pin was showing. In addition, “Navigation” and “open in Google Maps” shortcuts can be found in the bottom right corner.
Fig. 3.8 Call logs page of main application

Fig. 3.8 shows the UI for the call logs page. This page provides 2 core functionalities of the application: phone number checking (查詢電話號碼) and call log display (通話記錄). The phone number checking feature allows users to input any phone number and check whether it is malicious or not. This feature was not planned originally, but it was
developed during the preliminary implementation. The project team decided to keep this feature in the release version as we can see the benefits of having it.

Fig. 3.9 Phone number checking notification

A notification will be delivered to the user once the checking is done (Fig. 3.9). The notification message simply shows the checked phone number and its status. Phone number checking not only serves as an additional feature, but also served to allow elderly to verify phone number for other elderlies that are in need, spreading the sense and knowledge of avoiding telephone deception. While this checking is a manual process, the call log display (通話記錄) was fully automated.

Fig. 3.10 Call logs display

The 10 most recent call records on the device with type identification will be shown (Fig. 3.10). The call logs are sorted by date of call, and they will be classified into 3 types: Safe (安全), Malicious (可疑來電) and In Contact List (通訊錄聯絡人). The
application makes use of color-coded background for different calls, with the call type shown on the bottom right corner of each entry.

For Safe (安全) calls, a light-yellow background will be used for the entry. Green background was not used as the phone number database may not be 100% accurate, and we want the elderly to develop a sense that a so-called safe phone call may still be malicious. Basic information such as the phone number and date of call will be displayed.

A red background will be used for Malicious (可疑來電) calls. Calls that will be identified as malicious are verified malicious calls on HKJunkCall, so non-verified malicious calls will not fall into this category. Same as Safe calls, basic information of the call will be displayed.

Calls to phone number in the local contact list will bypass the checking and classify as In Contact List (通訊錄聯絡人) directly. A green background is used for such entries, and only the name of the contact will be displayed to avoid clunkiness and exposure of unnecessary personal data.

![Scam Tracker](image)

**Fig. 3.11** Persistent notification of main application

A persistent notification for the background process will show up on the phone (Fig. 3.11). This allows the elderly to ensure the application is running by just looking at the icon on the status bar. The location information and checked call log will be uploaded to database automatically even when the elderly is not using the application, through a background process.

Since the main application access on-device call log and geolocation parameters, Locations and Call Log access are required from the Android system. Such requirements will be prompted during the initial launch of the main application.
Fig. 3.12 shows the UI for the connections page. This page displays a static 8-digit alphanumerical code generated locally. 0 (zero) is excluded from code generation as it looks too similar to the letter “O”. The companion application will make use of this code for remote data access with the help of the database. The use of a randomly generated code aims to avoid duplicated code among different families. In addition, since elderlies cannot modify the code, such implementation helps to avoid accidental code changes.
The above concluded the usage and UX of the main application. The companion application was basically a carbon copy of the main application with a twist.

The connections page of the companion application features a vastly different UI when compared to the main application (Fig. 3.13). Instead of showing a locally generated code, an input box is shown on this page. Family members can input the user code from the elderly’s device to make a connection with the elderly. Once connected, the code will be shown above next to the Connected Code (已連結的代碼) text.

Fig. 3.13 Connections page of companion application
The locations page of the companion application features the same UI as the main application (Fig. 3.14). Instead of showing the user’s own location, the location of the connected user (elderly) will be shown instead. When no user is connected, a blank spot will be shown by default. Tapping on the pin reveals the same shortcuts as found in the main application, and the “Navigation” shortcut will be helpful if family members need to provide face-to-face assistance.
Call logs page follows a similar concept as the locations page, showing an identical UI when compared to the main application (Fig. 3.15). The 2 functionalities were kept, but the call logs display shows nothing by default, unless the companion app is connected to another user by a code. Once connected, the 10 most recent calls of the elderly will be displayed with the same structure and layout. The phone number checking section works exactly as it is when compared to the main application.
Since the family members were not expected to monitor the elderly 24/7, the companion application also makes use of a background process to fetch data from the database periodically.

Identical to the main application, a persistent notification will be displayed when the application is running in the background (Fig. 3.16). The background process only fetches the call log from the database in the background, as the user will open the app if they need to know where the elderly goes, leaving no reason to fetch location information in the background.

When the elderly receives a fresh malicious call, a notification will be delivered to the connected companion application users (Fig. 3.17). The processing was done in the background to keep the family members informed, without the need to open the application and check for the updated call log manually.

The data exchange between the two applications is achieved through the use of a customized algorithm, a unique user code and a database. More details will be explained in the following section.

### 3.2.4 Data exchange between the two applications

A Realtime Database was deployed for our application. Unlike the generic relational database, all data stored in Realtime Database is simply JSON data. Thus, we only need to ensure uniqueness on the top layer to avoid conflict records among different users.
The above figure shows how user data are structured in Realtime Database (Fig. 3.18). The top layer of each data entry is simply a user code. The database make use of the same code generated on devices to identify different users (elderlies). This also serves the purpose of a primary key if considered as a relational database.

Under each user code, location and call log data will be recorded. The actual location data, such as the latitude and longitude of the most recent location will be stored under the “Location” entry. For “callRecord” entry, up to 10 call records will be uploaded to the database, and each record will be identified with a number (0-9). Under each numbered entries stores the call data, such as the phone number and the identified call type.
<table>
<thead>
<tr>
<th>callRecord</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>type: &quot;Safe&quot;</td>
</tr>
<tr>
<td>dateTime: &quot;2022年4月13日 下午10:18:29&quot;</td>
</tr>
<tr>
<td>duration: 0</td>
</tr>
<tr>
<td>phoneNumber: &quot;39522495&quot;</td>
</tr>
<tr>
<td>rawType: 2</td>
</tr>
<tr>
<td>timestamp: &quot;1649859509908&quot;</td>
</tr>
<tr>
<td>type: &quot;OUTGOING&quot;</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>type: &quot;Malicious&quot;</td>
</tr>
<tr>
<td>dateTime: &quot;2022年4月13日 下午5:50:18&quot;</td>
</tr>
<tr>
<td>duration: 0</td>
</tr>
<tr>
<td>phoneNumber: &quot;92955335&quot;</td>
</tr>
<tr>
<td>rawType: 2</td>
</tr>
<tr>
<td>timestamp: &quot;1649843418247&quot;</td>
</tr>
<tr>
<td>type: &quot;OUTGOING&quot;</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>type: &quot;In Contact List&quot;</td>
</tr>
<tr>
<td>dateTime: &quot;2022年4月13日 下午5:16:32&quot;</td>
</tr>
<tr>
<td>duration: 0</td>
</tr>
<tr>
<td>name: &quot;Walter&quot;</td>
</tr>
<tr>
<td>phoneNumber: &quot;*85298846301&quot;</td>
</tr>
<tr>
<td>rawType: 2</td>
</tr>
<tr>
<td>timestamp: &quot;1649841392395&quot;</td>
</tr>
<tr>
<td>type: &quot;OUTGOING&quot;</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>type: &quot;In Contact List&quot;</td>
</tr>
<tr>
<td>dateTime: &quot;2022年4月12日 下午10:41:16&quot;</td>
</tr>
<tr>
<td>duration: 0</td>
</tr>
<tr>
<td>name: &quot;Raymond&quot;</td>
</tr>
<tr>
<td>phoneNumber: &quot;67021047&quot;</td>
</tr>
<tr>
<td>rawType: 3</td>
</tr>
<tr>
<td>timestamp: &quot;1649774476027&quot;</td>
</tr>
<tr>
<td>type: &quot;MISSED&quot;</td>
</tr>
</tbody>
</table>

Fig. 3.19 Sample record on Firebase
Fig. 3.19 shows an actual record of the uploaded by our application to Firebase. When data is uploaded to the database, Firebase takes care of duplications by overwriting data if a certain code was found. Thus, every new upload from the main application overwrites the old record, keeping the database record always up to date. For the companion application, data will be fetched to the application by accessing [database]/[User Code]/Location or [database]/[User Code]/callRecord, depending on which type of data is required. Since the data uploaded to Firebase are processed data, the companion app is not required to perform data manipulations.

3.3 Challenges & Difficulties

During the development of the application, the project team has faced different challenges. The challenges can be divided into 3 types: OS related issues, code related issues and external resource related issues.

The use of React Native in theory allows the project team to create cross-platform applications. While this is the ideal case, limitations on iOS have been a roadblock for the project team. A prime example for this will be call log access on iOS devices. User’s privacy is used by Apple as a guard to block application developers from accessing call log on iOS devices. Call log access and manipulation has been the core feature of our application, and force reading call log data on iOS device is only available through private APIs or hacking, which was not ideal for an application aimed towards general users. Because of that, the project team had aborted iOS version development at this moment.

With the introduction of CallKit since iOS 10, developers are able to work on Call Directory type applications for iPhone. While the project team can choose to implement a malicious call only call blocker for the iOS application, this would lead to severe functionality differences among applications on different platforms. In addition, since our call identification is done through web scraping, such solution leads to latency and other issues that are difficult to solve, which will be explained later.

During the coding process, the project team also came into different obstacles, and we are pleased to report that such obstacles are being solved. When we first started on implementing the background process, we have faced conflicts between the background process and the main UI tasks, which was solved by splitting their work. In the current implementation, all web scraping related tasks are done by the
background process to avoid multiple web scrapes at the same moment. In addition, all uploads on the main application are done by the background process to avoid conflicting updates.

This does not mean the background process implementation is flawless. The current implementation makes use of Headless JS, a native library of React Native to achieve background JavaScript tasks. A headless task (or an activity) is not the same as the background process, with the limitation that killing the parent Activity would kill the headless task simultaneously. In short, for our application to work properly, the user cannot kill the app through Task Manager. Many Android users has a common habit of killing all tasks through the Task Manager, which will lead to feature loss when using our application. Our solution to this is to run the background task with a persistent notification, to give the user an indicator on whether the application is running or not.

HKJunkCall is used to provide a bespoke experience for local users, but the lack of API requires the project team to resolve to using web scraping to make use of this database. While HKJunkCall is scrapable, during the testing stage, the project team has found out that the website has implemented anti-scrape measures to stop us from scraping HKJunkCall. We have found out that the website has IP detection, and a short burst of queries (~20 queries) on the site would lead to a page that requires Captcha before one can proceed. This would cause the scraping algorithm we implemented to stop working.

Our solution to this is to implement algorithms that reduce the amount of web scrapes required. Performing background call log checking on demand helps to reduce the number of web scrapes made to HKJunkCall website, to keep the background task running as intended.
Fig. 3.20 Call log checking algorithm

Fig. 3.20 shows the algorithm to determine whether a call log will be checked with HKJunkCall and pushed to the database. The first thing to check is whether the call log has updated, and no operations will be performed when no changes are found, since it is unnecessary. To save a few scraping cycles, numbers saved in contacts list will not be checked with HKJunkCall. Although this is not ideal, it is the best solution to ensure stability that the project team has at the current state. The lack of API access and anti-scrape measures forces the project team to prioritize of stability over accuracy and coverage.

Another issue regarding the web scraping is about its latency and accuracy. The current web scraping method implemented by the project teams was not particularly fast, especially in the case of using the Phone number checking (查詢電話號碼).
feature. From pressing the button to request for a checking, to the application sending a notification for results takes around 2 seconds. Though it was not absurdly slow, checking with an offline database that comes with the API is much faster.

In terms of web scraping accuracy, the current scraping algorithm focuses on the `<header>` tag, which omits lots of information on the actual page itself, potentially leads to a lower accuracy in determining malicious calls.

![Fig. 3.21 Web scraping on HKJunkCall](image)

Fig. 3.21 shows a keyword when I try to manually scrape the information using inspect view *(Please enlarge it to look for the text)*. I was looking up on a non-verified number, and the only information on this call is “提高警覺” found in the red container. While we can find the text with inspect view, a closer inspection found out that the text was bounded with multiple tags without class or ID attributes (for example the `<table>` and `<font>` tags).
Fig. 3.22 HKJunkCall call reporting interface

Fig. 3.22 shows an interface that was existent on all pages of the website. If we purely scrape for the keywords, the keywords found in the call reporting section under each page would screw our findings and lead to false results. So unfortunately for non-verified calls, we have no concrete solution to them at this moment.
3.4 Future Plan

The iOS version of the main application was aborted due to call log access restrictions, but the companion application should be fine to run on iOS. Due to time constraints, the project team focused on polishing the Android variant during our development timeline. The project team plans to port the companion application on iOS as the first step towards Apple users. Once the call log restrictions were relaxed by Apple, or other ethical access to iOS call log was established, the project team also plans to rework on the iOS variant to make the application truly cross-platform.

Currently the web scraping was done on the user’s device, which was also known as client-side scripting. In the future, the project team plans on moving towards server-side scripting to take more control of how information is scraped and deploying better anti-scrape measures. This could also contribute to less frequent app updates, which is beneficial towards user experience on our application.

In the previous presentation, Dr. Tam gives the project team an insight of performing call forwarding through the application. Though call forwarding can be done natively on both Android and iOS devices following certain policies, on-demand and situation-based call forwarding will be less obtrusive towards both the elderly and the family members. The project team wish to add this feature in the future, to allow family members to take more control when necessary.

While on-demand call forwarding will be a great feature to be added, the project team have concerns regarding its usage and privacy matters. Our applications are designed to run without any setups, but call forwarding needs a setup for it to work. If such functionalities are implemented, we recommend the family members to take the responsibility for this one-time setup. In terms of privacy matters, application-based call forwarding may require third-party VoIP services, which are not welcomed by privacy-focused users. The project teams plan to work on algorithms for native call forwarding before investigating on third party solutions.
4. Conclusion

As old as telephone deception grows and aided by global pandemic, Hong Kong people are still falling for this trick and led to a surge in telephone deception cases. Despite of the government educating citizens about telephone deception, elderlies are still highly vulnerable to such scams. In 2021, a telephone deception case with record-breaking monetary loss shows the severity of telephone deception among the seniors.

Existing call blocking solutions was not user-friendly towards many citizens. Thus, this project aims to provide an application as the solution. The application will provide bespoke scam tracking experience for the elderly, while alerting other family members about telephone deception. We believe this application could educate and protect citizens against telephone deception, and it is hoped that telephone deception cases will depreciate in the coming years.

In the future, the project team wish to implement solutions that was lacking in the market, such as a customizable on-demand call forwarding service in our application, to give elderlies a platform to seek for help from the family members, as well as enabling the family members to take more control on malicious calls received by the elderlies.
References


