

**科技部補助**  
**大專學生研究計畫研究成果報告**

計 畫 ： 基於Beacon技術之兒童居家危險示警平台 名 稱
---------------------------------------

執行計畫學生：徐子軒  
學生計畫編號：MOST 107-2813-C-035-128-E  
研究期間：107年07月01日至108年02月28日止，計8個月  
指導教授：陳錫民

處理方式：本計畫可公開查詢  
執行單位：逢甲大學資訊工程學系（所）  
中華民國 108年03月08日

# 基於 Beacon 技術之兒童居家危險警示平台

學校：逢甲大學

系所：資訊工程學系

學生：徐子軒

## 誌謝

很慶幸能夠獲得科技部的肯定，也很榮幸能夠請陳錫民老師擔任指導老師，感謝老師一路上的教導，不斷的提供方向省去我們走冤枉路的時間，曾經幫助過我們的學長姐，當我們遇到問題時你們也二話不說的給予我們協助，也感謝一路上相陪的組員，沒有你們的幫助我絕對無法完成此次計劃。

# 目錄

目錄.....	3
(一) 動機.....	6
(二) 系統開發平台.....	7
2.1 Visual Studio Code.....	7
2.2 Android Studio.....	8
2.3 Android.....	9
2.4 Raspberry Pi .....	10
2.5 Arduino .....	10
2.6 Git .....	10
(三) 系統功能.....	12
(四) 文獻回顧與探討.....	13
4.1 Sinch API .....	13
4.2 Bluetooth .....	13
4.3 Beacon .....	13
4.4 MQTT .....	14
4.5 資料庫管理系統.....	14
(五) 研究方法與步驟.....	16
5.1 研究方法.....	16
5.2 研究步驟.....	20
(六) 結果.....	22

(七)	實作介紹 .....	23
(八)	User Case.....	30
(九)	ER-Model .....	31
(十)	系統操作流程圖 .....	32
(十一)	操作介紹 .....	33
(十二)	未來展望 .....	34
(十三)	心得 .....	41
(十四)	參賽紀錄 .....	42
(十五)	參考文獻 .....	43

## 圖目錄

圖 1 Visual Studio Code 介面	7
圖 2 Android Studio 介面	9
圖 3 Android 系統架構圖	9
圖 4 Git 指令、狀態圖	12
圖 5 Beacon 裝置 [10]	14
圖 6 兒童居家危險示警平台之系統架構	16
圖 7 穿戴式 Beacon [17]	17
圖 8 gradle 檔配置文件信息	23
圖 9 類別目錄	24
圖 10 登入應用程式碼實作範例	25
圖 11 取得使用者資訊程式碼實作範例	26
圖 12 增加使用者資訊程式碼實作範例	26
圖 13 連接 MQTT Server 部分程式碼	27
圖 14 連接 Sinch 程式碼實作範例	28
圖 15 偵測 Beacon 訊號部分程式碼實作範例	29
圖 16 系統使用者的使用案例圖	30
圖 17 資料庫 ER-MODEL	31
圖 18 系統操作流程圖	32
圖 19 登入介面	33
圖 20 註冊介面	34
圖 21 連絡人介面	35
圖 22 設定介面	36
圖 23 監控介面(一)	37
圖 24 監控介面(二)	38
圖 25 接聽介面	39
圖 26 家中擺放之蜂鳴器	40

## (一) 動機

家看似是最安全的地方，但對於孩童來說可能是最危險的地方，因孩童的生活圈小，而最常去的地方就是家，且台灣為人口密度第二高的國家，因地狹人稠多數人會選擇居住公寓，而正因此窗戶成了最危險的死角，而我們也常在新聞上看到許多在家中發生的意外，這時就有個想法，如果能靠我們所學來製作一個系統來負責管理這些死角，因此我們找出了家中常發生災難的區域，並且想像了對策。這樣不但能夠減少家長的壓力，而且也有助於家中全體成員的安全。

有了想法後就必需考量是否可行，而近年來剛好搭上了物連網[1]和創客的風潮，Arduino 開發版中的各項感測器和已融入在我們生活中的通訊軟體正能符合需求，而想法和可行性都具備後則確定了能夠進行。

## (二) 系統開發平台

### 2.1 Visual Studio Code[2]

Visual Studio Code 是一套由微軟開發的開放原始碼跨平台整合開發環境 (Integrated Development Environment)，因其提供了強大的外掛機制，可用於開發 C/C++、Python、PHP、Web (HTML、CSS、JavaScript、XML ...)、JSP 等語言。

本系統使用 Visual Studio Code 以 Python 來開發後端 RESTful 功能、與資料庫的新增、刪除、修改、查詢功能。

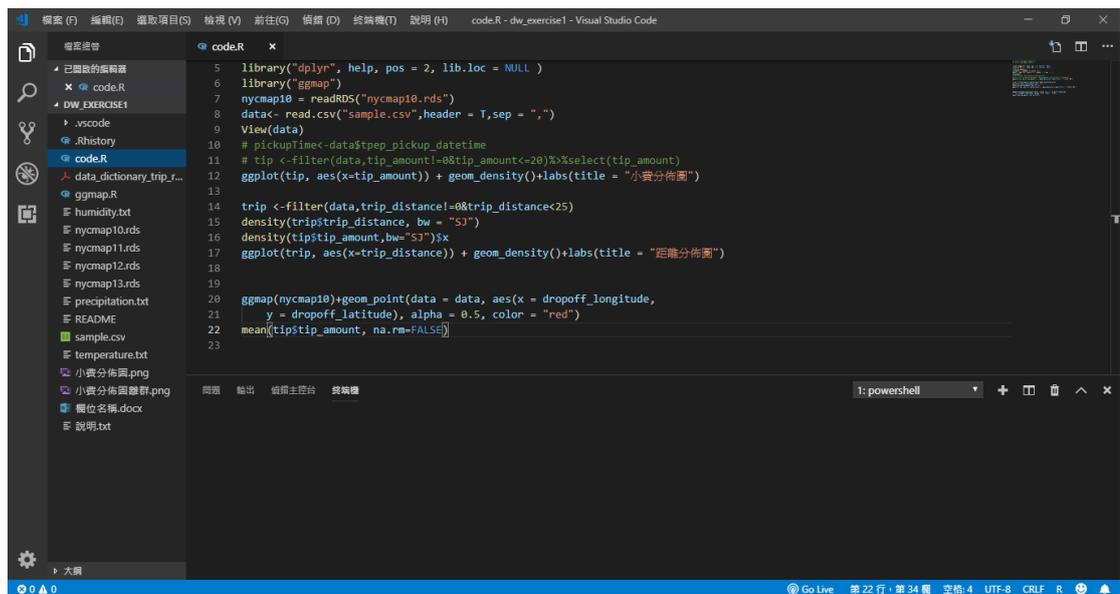


圖 1 Visual Studio Code 介面

## 2.2 Android Studio[3]

Android Studio 是一個為 Android 平台開發程式的整合式開發環境 (Integrated Development Environment)。2013 年 5 月 16 日在 Google I/O 上發布，可供開發者免費使用，運用的語言有 JAVA 和 Kotlin。

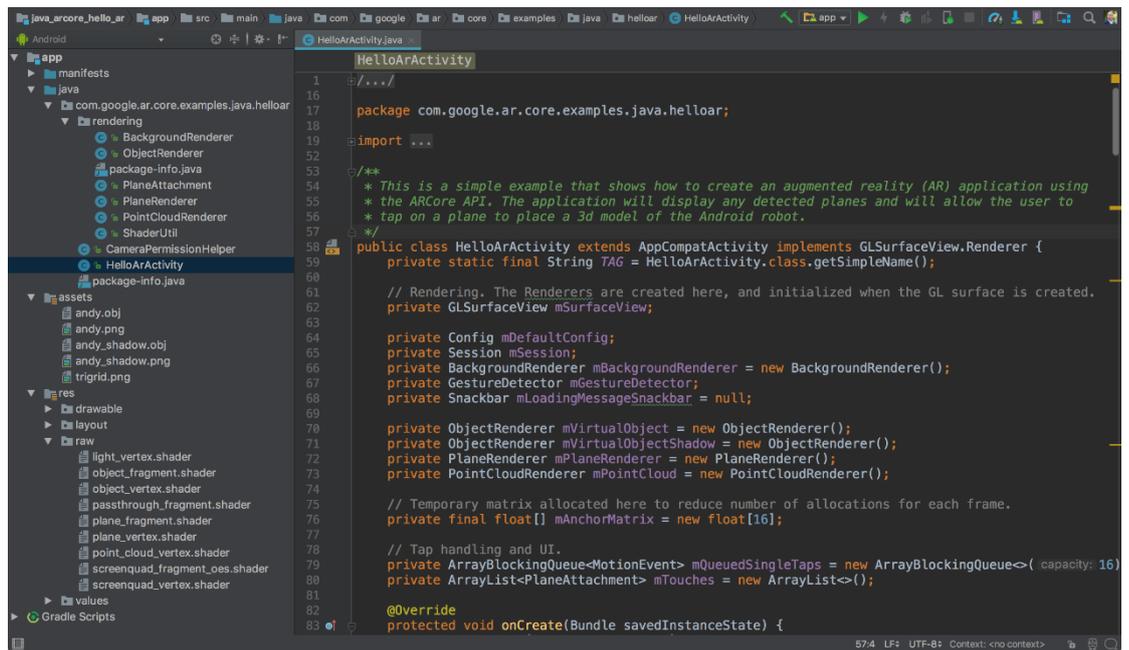


圖 2 Android Studio 介面

## 2.3 Android[4]

Android 作業系統平台是以 Linux 系統為基礎核心，擴充 Android 執行作業環境 (Android Runtime) 以及應用程式開發平台 (Application Platform)，Android 作業系統的軟體堆疊包含：

1. 應用程式 (Application)
2. 應用程式框架 (Application Framework)
3. Android 執行作業環境 (Runtime) 和各類函式庫
4. Linux 作業系統核心 (Linux Kernel)

根據 IDC[5]統計目前全球 Android 手機市佔率超過 80%，為此

本

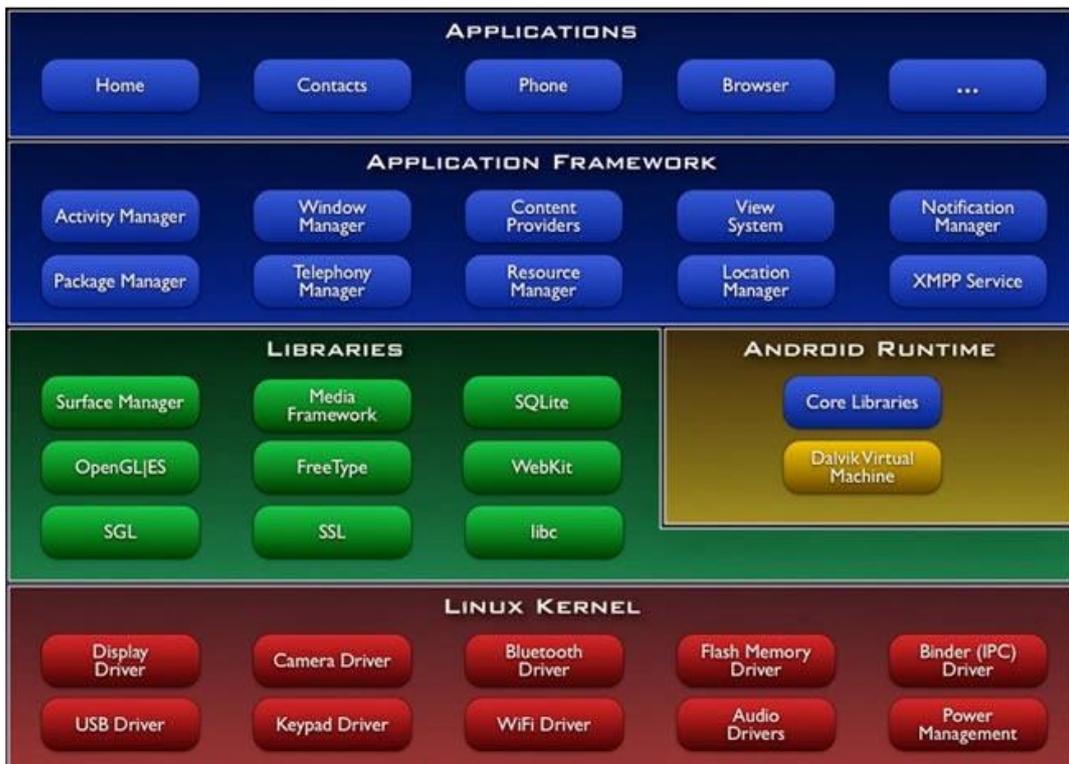


圖 3 Android 系統架構圖

## 2.4 Raspberry Pi [6][7]

Raspberry Pi 是一款基於 Linux 的單板機微型電腦。它由英國的樹莓派基金會所開發，只有火柴盒大小，擁有入門級的硬體，目的是以低價硬體及自由軟體促進學校的基本電腦科學教育。由於其功耗極低且易收納適合做為本計畫中是危險示警平台的開道器 (Gateway)，用以與家長手機、危險區域裝置以及物聯網設備互動與交換資料。

## 2.5 Arduino[8]

Arduino，是一個開放原始碼的單晶片微控制器，它使用了 Atmel AVR 單晶片，採用了開放原始碼的軟硬體平台，建構於簡易輸出/輸入介面板，並且具有使用類似 Java、C 語言的 Processing/Wiring 開發環境，也因為其容易開發的特性故可以應用於許多控制與感測設備上。

## 2.6 Git[9]

Git 是用於 Linux 內核開發的版本控制工具。與 CVS、Subversion 一類的集中式版本控制工具不同，它採用了分散式版本控制的作法，不需要伺服器端軟體，就可以運作版本控制，使得原始碼的釋出和交流極其方便。Git 最為出色的功能是它的合併追蹤 (merge tracing)，團隊在開發系統時，可以針對每個功能

的更新，去新增一個分支 (branch)，分支並不影響主線的進行，若功能修改完成，便可 merge 回主線上，如此一來主系統也順利增加了一個功能。因本系統包含了許多功能，故選擇使用 Git 來控管程式版本，以確保系統之完整性，也能使得開發過程較為順暢。

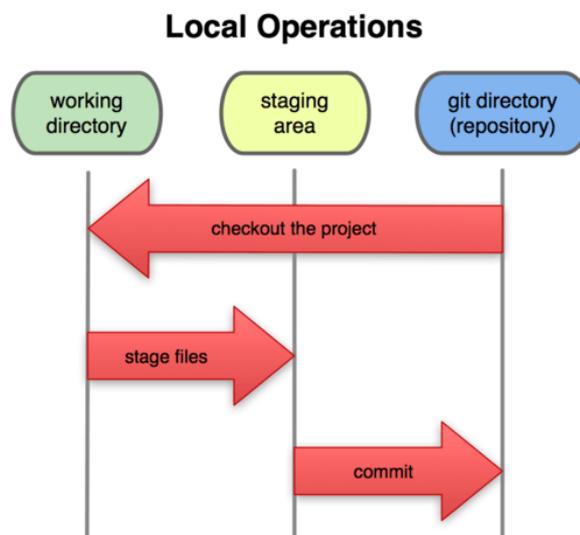


圖 4 Git 指令、狀態圖

### (三) 系統功能

使用者設定完我們的示警平台開道器後，即可透過 Android 手機的應用程式輸入 Beacon 裝置、Android 感測裝置，並將 Beacon 放至兒童身上，Android 感測裝置放至危險區域，隨即即可透過 Android 手機的應用程式設定安全距離和緊急連絡人。

一旦兒童接近危險區域，示警平台中的蜂鳴器會響起，告知在家中的家長，Android 感測裝置同時也會向綁定的緊急連絡人發送視訊電話，若未接通會一直打至使用者接聽或者兒童已離開危險區域。

而視訊電話的接通我們也有特別設計過，若受話方是 Android 感測裝置，則撥打後會自動接通，為的就是要讓家長能夠即打即看到危險區域的狀況。若受話方是家長時，我們為了能使家長在第一時間察覺到，不論家長手機是什麼模式本應用程式都會將其調整至響鈴模式，並且將鈴聲調至最大。

## (四) 文獻回顧與探討

### 4.1 Sinch API

Sinch 提供開發者以方便、精簡的方法開發語音、視訊的應用程式，其不只提供桌面行應用程式，亦提供了行動應用程式的開發套件。

### 4.2 Bluetooth

是一種近距離無線通訊技術標準，用來讓固定與行動裝置，在短距離間交換資料，以形成個人區域網路 (PAN)。其使用短波特高頻 (UHF) 無線電波，經由 2.4 至 2.485 GHz 的 ISM 頻段來進行通訊。

### 4.3 Beacon

Beacon 是基於藍牙 4.0 中 BLE 技術所發展而來的。如圖 5 所示，一般 iBeacon 的裝置只會發送簡單的訊號(如 UUID )給周遭的接收者，接收者可以根據所接收到 UUID 透過 APP 傳送到雲端取得接收者所在的位置，APP 可以根據該位置提供合適的服務，例如百貨業者可以在展示櫥窗前安裝 Beacon 裝置，當接收者靠近櫥窗接收到 Beacon 裝置所發出的訊號時，接收者 APP 上便會呈現櫥窗產品的詳細介紹。由於大部分的 Beacon 裝置只是單純地發送

低功率訊號給附近的接收者，因此其優點是使用一般的小型電池便可以持續使用一年之久不需更換電池。

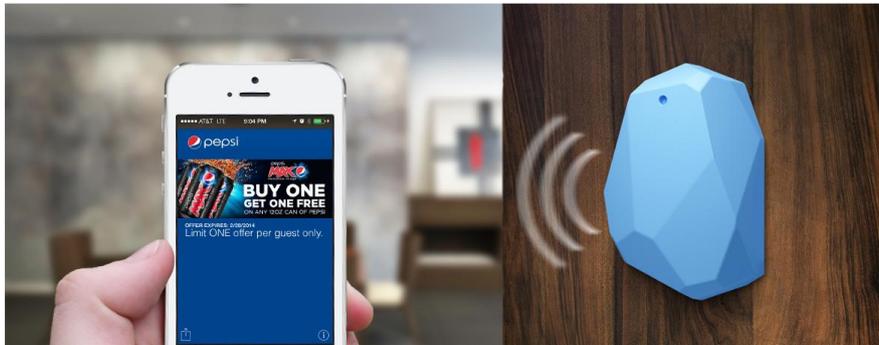


圖 5 Beacon 裝置[10]

#### 4.4 MQTT[11]

專為物聯網而設計的協定，並且它是透過 publish 和 subscribe 的方式來做訊息傳送。由於是為了物聯網而設計的協定，因此它所需要的網路頻寬是很低的，而所需要的硬體資源也是低的。

#### 4.5 資料庫管理系統[12]

目前常見資料庫系統主要可分為下列二種

管理系統：

1. 關聯式資料庫：採用表格的方式，儲存固定結構的資料，表格欄位中有中主鍵，用以區別每一筆資料的唯一性，以及外鍵用以關聯其他表格，關聯式資料庫大都採用 SQL 語言[13]用以新增、刪除、修改與查詢資料庫中的資料。目前常見的

關聯式資料庫有：Oracle Database、MySQL、PostgreSQL 與 SQL Server 等

2. NoSQL 資料庫：主要的特色是所儲存資料的結構可以不是固定的格式，讓資料儲存更有彈性。另外，不同於關聯式資料庫，大多採用 SQL 作為資料查詢語言，各 NoSQL 資料庫的開發組織有提供各自的資料存取語言與語法，相較於傳統關聯式資料庫，NoSQL 資料庫具有水平擴展的優點。常見的 NoSQL 資料庫有：MongoDB[14]、ElasticSearch[15] 與 Cassandra[16]等。

## (五) 研究方法與步驟

### 5.1 研究方法

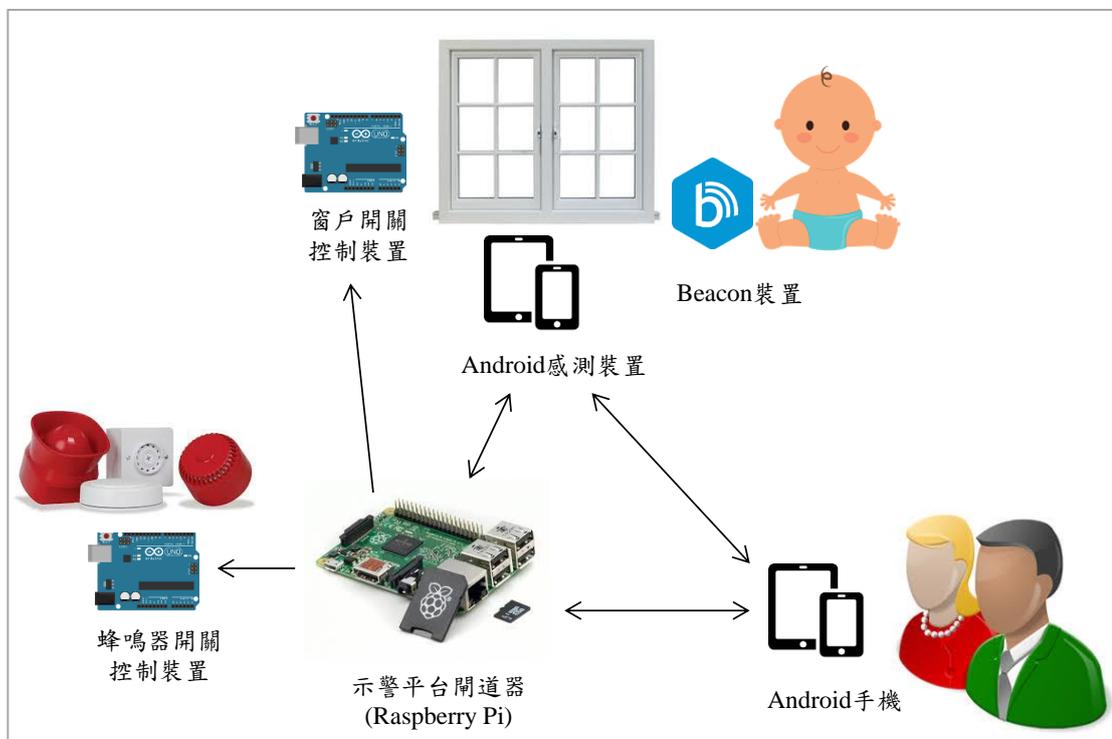


圖 6 兒童居家危險示警平台之系統架構

圖 6 為本計畫系統架構概念圖，其硬體架構主要分為以下幾個部分：(1)示警平台開道器；(2)Beacon 裝置；(3)Android 感測裝置；(4)Android 手機；(5)開關控制裝置。

1. 示警平台閘道器：其主要功能為串聯危險區域的 Android 感測裝置、Android 手機以及所部署的物聯網開關控制裝置。在示警平台閘道器中，我們將開發 MQTT 代理人(Broker)，用以管理家中部署的物聯網控制裝置，以及將 Android 感測裝置與 Android 手機所傳送 HTTP 控制訊息轉換成 MQTT 訊息格式，再將轉換後的控制訊息透過 MQTT 傳輸協定傳送至相對應的物聯網開關控制裝置。藉此，當兒童靠近危險區域時，系統可以主動地關閉危險設施或由家長透過手機控制危險設施。



圖 7 穿戴式 Beacon[17]

2. Beacon 裝置：如圖 7 所示，目前的 Beacon 裝置已經可以做成小體積的穿戴式手環且具低耗電性。不同於過去 Beacon 的應用是將 Beacon 裝置固定在特定的地點作為定位使用。在本系統中，我們反轉 Beacon 裝置與行動裝置的角色，將行動裝置固定在危險區域的周遭，作為距離感測使用，而將 Beacon 裝

置配戴在兒童身上，藉由行動裝置所接受到兒童身上 Beacon 裝置所發出藍芽訊號的強弱，計算兒童靠近危險區域的距離。

3. Android 感測裝置： Android 感測裝置為支援 BLE 通訊協議的手機或平板。由於目前大部分的 Android 手機與平板皆以支援 BLE 協議，在手機汰換率高的今天，使用者可以將不用的 Android 手機作為 Android 感測裝置，部署在危險區域周遭。我們將開發 Android 感測裝置中的感測與影像互動 APP，其主要功能為：(1)偵測 Beacon 裝置是否靠近此感測裝置；(2)當偵測到兒童接近時，透過聊天機器人與通訊軟體，向家長端地 Android 手機發出警示；(3) 啟動影像互動功能，讓家長與兒童可透過影音互動；(4) 當偵測到兒童接近時，發送控制訊息給示警平台閘道器，藉以關閉危險設施或開啟示警設備。
4. Android 手機：為家長端隨身攜帶的行動裝置，我們將整合現有的通訊軟體，安裝在家長的手機上，當偵測到兒童靠近危險區域時，通訊軟體中的聊天機器人可以與家長互動，發出示警訊息、開啟影音互動功能與控制關閉危險設施或開啟示警設備。

5. 開關控制裝置：其主要用以控制危險設施或示警設備的開關。

在開關控制裝置中，我們將開發 MQTT 客戶端程式，以訂閱屬於該裝置的控制訊號。藉此當偵測到兒童靠近危險區域時，可以關閉門窗與瓦斯爐，以及開啟蜂鳴器等避免意外發生。

## 5.2 研究步驟

### (1)學習相關知識

學習開發本計劃所需要的知識和技術，了解開發時所採用軟體工程方法與工具以及蒐集與本計劃相關的研究和產品，以奠定開發的基礎。

### (2)系統開發

- 規劃系統開發時程

為了能夠如期完成我們於開發前先行計劃該在什麼時間完成何種開發工作。

- 需求蒐集

有了初步想法後，我們將設想在家中所有可能發生的意外，並且將其和所想到的解決方案記錄下來，同時我們也將尋問有小孩的家長，確認使用者需求的可行性。

- 需求分析

承續者上述的需求我們也必需考量到能否達成該需求，並且也試想要以何種工具、何種技術來加以實踐，而對於評量後權重決定系統功能的開發順序。

- 軟體設計

根據的已確定的需求來規劃系統架構書，再加以設計高階的系統架構以及詳細的資料與軟體元件。

- 軟體實作

根據所設計的系統架構與設計，選用合適的程式語言和工具來開發，並建置實驗的場域，以部署開發的系統。

- 軟體測試

於開發同時我們也著手進行單元與整合測試，免於在完成之際時才發現有必需重頭來過的錯誤，系統完成後也會進行系統測試，以此來確保軟體的品質。

- 軟體維護

開發完成後我們會將所開發的系統給使用者使用，並且收集使用者給予的意見與問題回饋，做為日後改善的來源。

### (3) 撰寫成果報告

將開發過程詳細撰寫至文件及彙整成果報告，並與指導教授討論研究成果，撰寫研究論文並投稿。

## (六) 結果

本計畫將發展一套以 Beacon 技術為基礎的兒童居家危險示警平台，並期望具有下列成果：

- 將危險偵測設備設置於家中危險地點(如：陽台、窗戶)，當家中年幼成員靠近時，主動發出警訊，並且藉由通訊軟體傳送示警的內容至家庭群組。
- 當使用者收到示警訊息時，可透過此平台開啟危險偵測設備的螢幕以視訊的方式，了解兒童的行為並可藉由語音對話，告知兒童不要靠近此危險區域。
- 當使用者收到示警訊息時，可透過此平台控制居家設備，例如關閉門窗或瓦斯，以避免事故傷害發生。

## (七) 實作介紹

1. 首先在 android studio 中建立一個專案，並且開啟 gradle，gradle 是一個基於 Apache Ant 和 Apache Maven 概念的專案自動化建構工具。在執行專案時會去該項目跟目錄下讀取獲得需要的配置資料。

```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    implementation 'com.android.support:support-v4:27.1.1'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
    // https://mvnrepository.com/artifact/org.altbeacon/android-beacon-library
    implementation 'org.altbeacon:android-beacon-library:2.13.1'
    // https://www.hivemq.com/blog/mqtt-client-library-encyclopedia-paho-android-service
    implementation('org.eclipse.paho:org.eclipse.paho.android.service:1.0.2') {
        exclude module: 'support-v4'
    }
    // https://mvnrepository.com/artifact/com.android.volley/volley
    implementation 'com.android.volley:volley:1.1.0'
    // implementation files('libs/sinch-android-rtc-3.8.0-VIDEO-SNAPSHOT.jar')
    implementation(name: 'sinch-android-rtc', version: '+', ext: 'aar')
    implementation 'com.android.support:design:27.1.0'
}
```

圖 8 gradle 檔配置文件信息

2. 在 src/main 目錄下開發此應用的程式，將功能分為五大部分：
  - (1) AbstractClass：將程式碼共同的部分向上提出以節省重覆程式碼。

- (2) Fragment：各 Fragment 對應之詳細實作，和介面操作實作程式碼。
- (3) Model：將所需的資料建立模型，以有組織性的方法管理、操作傳輸的資料。
- (4) Services：包含各項前後景的服務，裡面包含了 Restful、手機本地端的 Data Base、傳輸 MQTT 和串接視訊 API 和檢測 Beacon 訊號之服務。
- (5) 各 Activity 實作：各 Activity 對應之詳細實作，和介面操作實作程式碼。

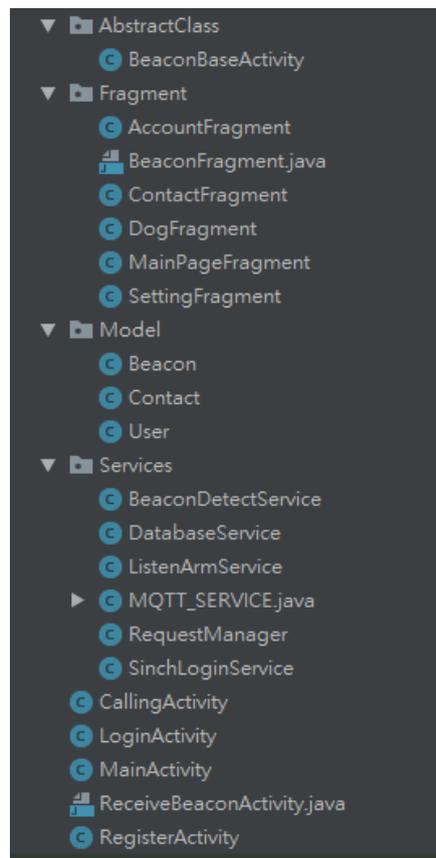


圖 9 類別目錄

### 3. 詳細介紹 Service 內容

#### (1) RequestManager：運用 volley 套件向後端要傳輸資料

或者取得資料，其中包含了登入、登出、註冊、取得 Beacon 資料、新增 Beacon 資料、刪除 Beacon 資料、傳送警報等功能

```
public static boolean loginRequest(final String username, final String password) {
    String url = host + "/login";
    if (username.isEmpty() || password.isEmpty()) {
        LoginActivity.dismissLoading();
        return false;
    }
    StringRequest request = new StringRequest(
        Request.Method.POST,
        url,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonObject = new JSONObject(response);
                    int status = jsonObject.getInt( name: "status");
                    if (status == 200) {
                        LoginActivity.sendToast("Success Login");
                        JSONObject userObject = jsonObject.getJSONObject("data");
                        String username = userObject.getString( name: "username");
                        String password = userObject.getString( name: "password");
                        String name = userObject.getString( name: "nickname");
                        String email = userObject.getString( name: "email");
                        String callerID = userObject.getString( name: "username");

                        DatabaseService.getDatabaseService().insertUser(
                            username,
                            password,
                            name,
                            email,
                            callerID
                        );
                        LoginActivity.startSinch(username);

                        LoginActivity.successLogin();
                    } else if (status == 404) {
                        LoginActivity.sendToast("Invalid Username Or Password.");
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
                LoginActivity.dismissLoading();
            }
        },
    );
}
```

圖 10 登入應用程式碼實作範例

(2) DataBaseService：提供將存取資料庫功能，內含取得使用者資訊、取得連絡人資訊、刪除使用者 資訊、新增 Beacon 資訊等。

```
public User getUser() {
    SQLiteDatabase db = _instance.getReadableDatabase();
    ContentValues values = new ContentValues();
    Cursor cursor = db.query( table: "User",
        columns: null,
        selection: null,
        selectionArgs: null,
        groupBy: null,
        having: null,
        orderBy: null);
    JSONArray obj = null;
    String data = cursorToJson(cursor);
    try {
        obj = new JSONArray(data);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    return jsonToUser(obj);
}
```

圖 11 取得使用者資訊程式碼實作範例

```
public void insertUser(String username, String password, String name, String email, String callerID) {
    SQLiteDatabase db = _instance.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("username", username);
    values.put("password", password);
    values.put("name", name);
    values.put("email", email);
    values.put("callerID", callerID);

    db.insert( table: "User", nullColumnHack: null, values);
}
```

圖 12 增加使用者資訊程式碼實作範例

(3) MQTT：提供連接 MQTT server 之服務，使應用能夠訂閱

Arduino 之狀態且，能發送緊急狀態封包給 Arduino。

```
handler = new Handler(){
    @Override
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        if(msg.what == 1) {
            Toast.makeText(mainContext, (String) msg.obj,
                Toast.LENGTH_SHORT).show();
            System.out.println("-----");
        } else if(msg.what == 2) {
            status = MqttStatus.Connected;
            setTextViewStatus(MqttStatus.Connected);
            Toast.makeText(mainContext, "連接成功",
                Toast.LENGTH_SHORT).show();
            try {
                client.subscribe(myTopic, 1);
                client.subscribe("before", 1);
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else if(msg.what == 3) {
            status = MqttStatus.Connecting;
            setTextViewStatus(MqttStatus.Failed);
            Toast.makeText(mainContext, "連接失敗，系統正在重連",
                Toast.LENGTH_SHORT).show();
        }
    }
}
```

圖 13 連接 MQTT Server 部分程式碼

(4) SinchLoginService：提供連接 Sinch 的服務，使應用程式能夠使用 Sinch 所提供的視訊服務。

```
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    try{
        user = intent.getStringExtra( name: "callerId");
        Log.i(TAG, user);
        start(user);
        Log.i(TAG, user);
    }catch (Exception e){

        DatabaseService databaseService = new DatabaseService( context: this);
        String user = databaseService.getUser().username;
        start(user);
    }
    return START_STICKY;
}

private void start(String callerId) {
    sinchClient = Sinch.getSinchClientBuilder()
        .context(getApplicationContext())
        .userId(callerId)
        .applicationKey(APP_KEY)
        .applicationSecret(APP_SECRET)
        .environmentHost(ENVIRONMENT)
        .build();

    sinchClient.setSupportCalling(true);
    sinchClient.startListeningOnActiveConnection();
    sinchClient.start();
    Log.i(TAG, msg: "start: ");
    sinchClient.getCallClient().addCallClientListener(new SinchCallClientListener());
}
}
```

圖 14 連接 Sinch 程式碼實作範例

(5) BeaconDetectService：提供連接以偵測 Beacon 訊號強度推斷距離的服務，使應用程式能夠知道是否別戴著 Beacon 的小孩接近 Android 感測裝置，若超過安全範圍則下因應措施。

```
if(distance >= calculateDistance(txPower, mRssi)) {
    mBluetoothLeScanner.stopScan(mScanCallback);
    user = ContactFragment.contacts.get(no).recipientID;
    PowerManager pm=(PowerManager) mContext.getSystemService(mContext.POWER_SERVICE);
    @SuppressWarnings("InvalidWakeLockTag") PowerManager.WakeLock wl =
        pm.newWakeLock( levelAndFlags: PowerManager.ACQUIRE_CAUSES_WAKEUP |
            PowerManager.SCREEN_DIM_WAKE_LOCK, tag: "wake");
    wl.acquire();
    wl.release();
    KeyguardManager km= (KeyguardManager) mContext.getSystemService(mContext.KEYGUARD_SERVICE);
    KeyguardManager.KeyguardLock kl = km.newKeyguardLock( tag: "unLock");
    kl.disableKeyguard();

    DogFragment.playMedia(R.raw.dog2, volume: 15);
    RequestManager.armAlarm(MainActivity.user, status: "danger");
    DogFragment.setAngryDog();
    RequestManager.armAlarm(MainActivity.user, status: "danger");
    ContactFragment.callUser(mContext, user);
    RequestManager.armAlarm(MainActivity.user, status: "danger");
    no = (no + 1) %ContactFragment.contacts.size();
}
```

圖 15 偵測 Beacon 訊號部分程式碼實作範例

(八) Use Case Diagram

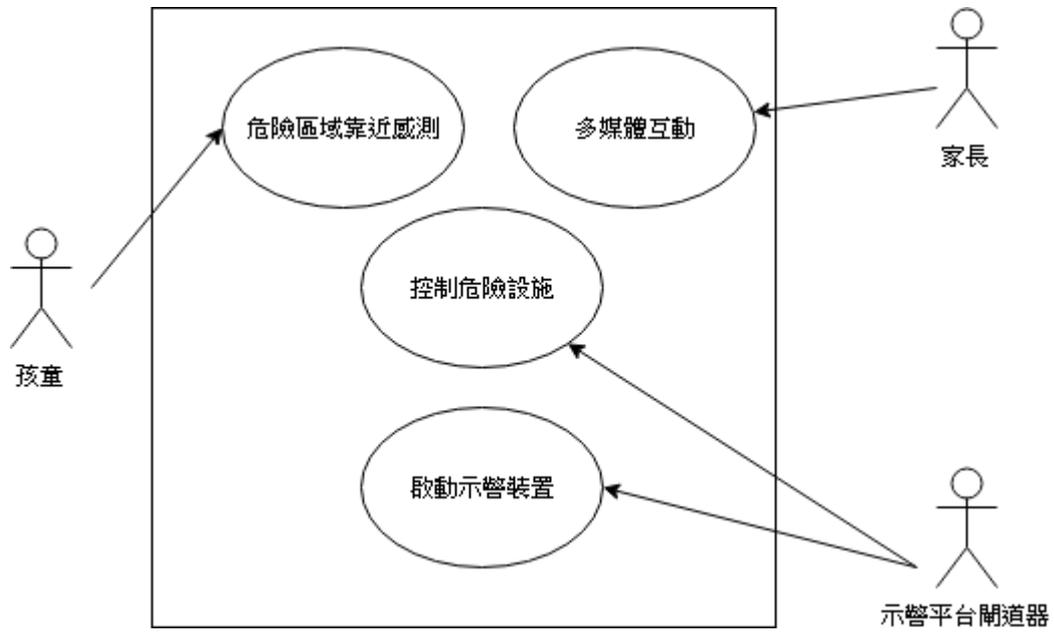


圖 16 系統使用者的使用案例圖

## (九) ER Diagram

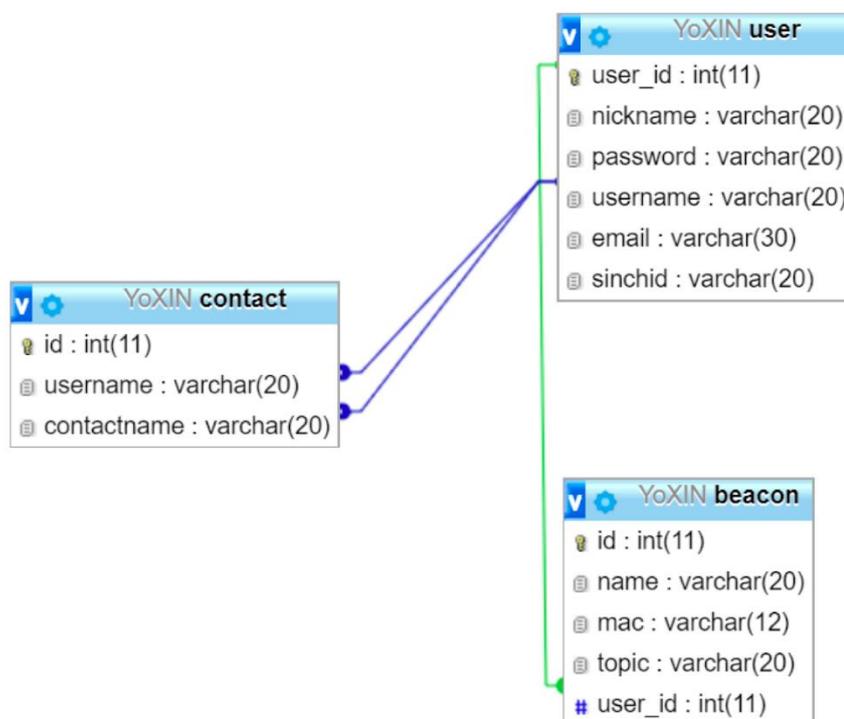


圖 17 資料庫 ER-MODEL

User 為使用者，sinchid 為 Android 使用到的套件上的 ID。

Contact 為聯絡人清單，存放 ID、名稱和聯絡人名字。

Beacon 為綁定至系統的 Beacon，mac 可存入 Beacon 的 MAC address。

# (十) 系統操作流程圖

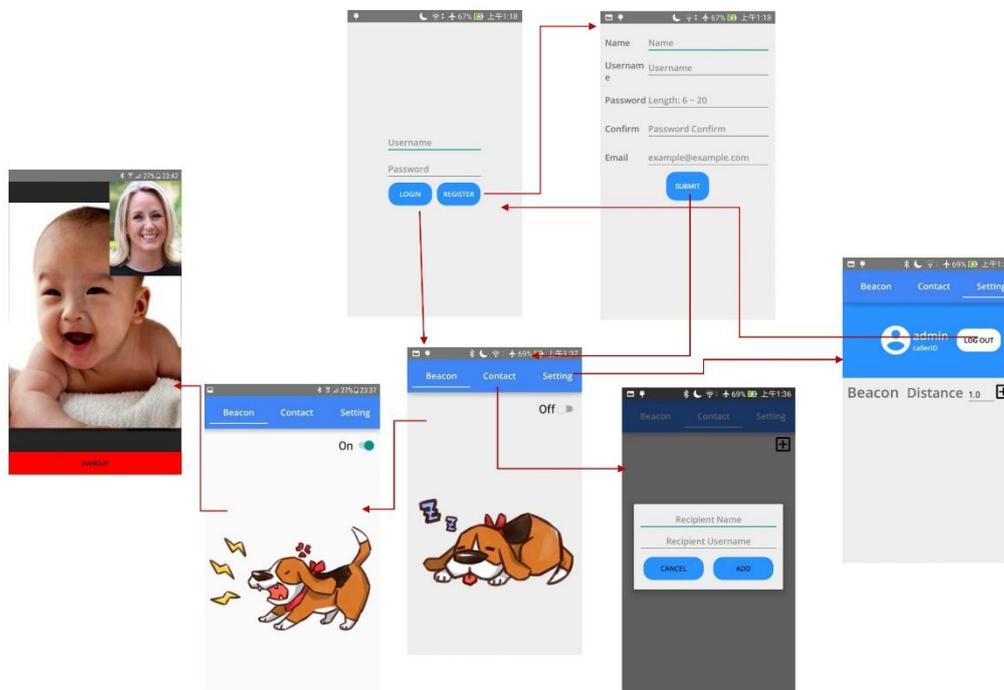


圖 18 系統操作流程圖

## (十一) 操作介紹

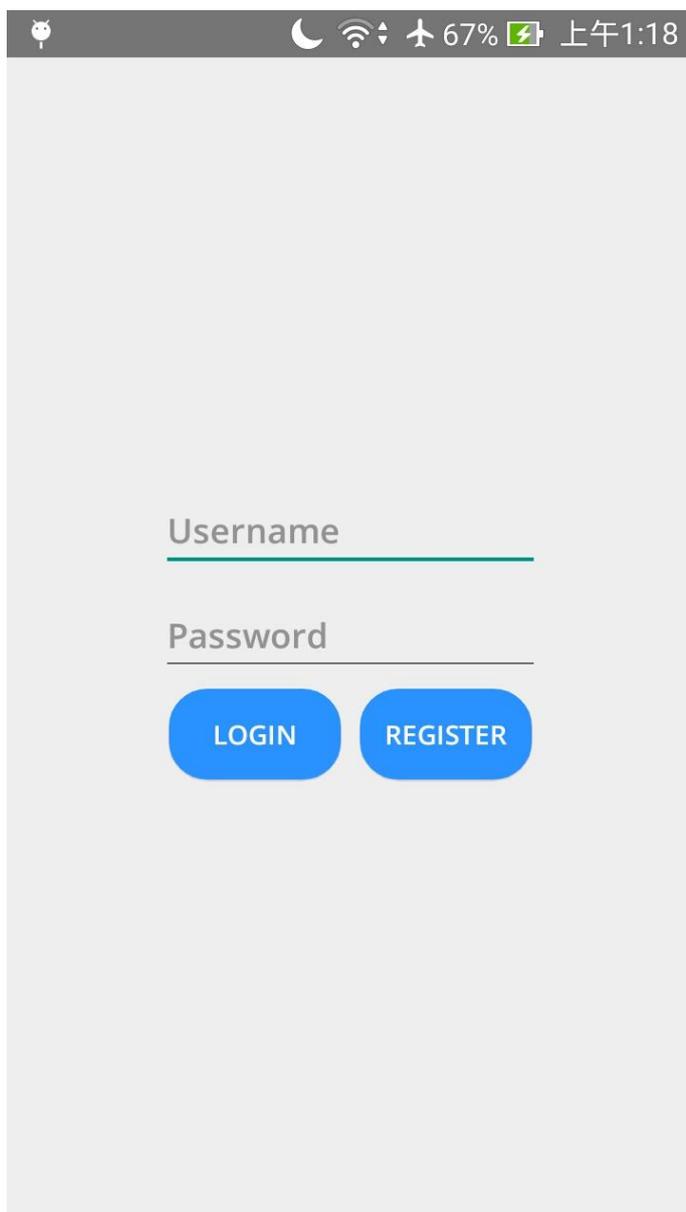


圖 19 登入介面

開啟 APP 可進入此畫面

若沒有帳號可按下註冊按鈕

Name

Username

Password

Confirm

Email

圖 20 註冊介面

輸入資料即可註冊成功

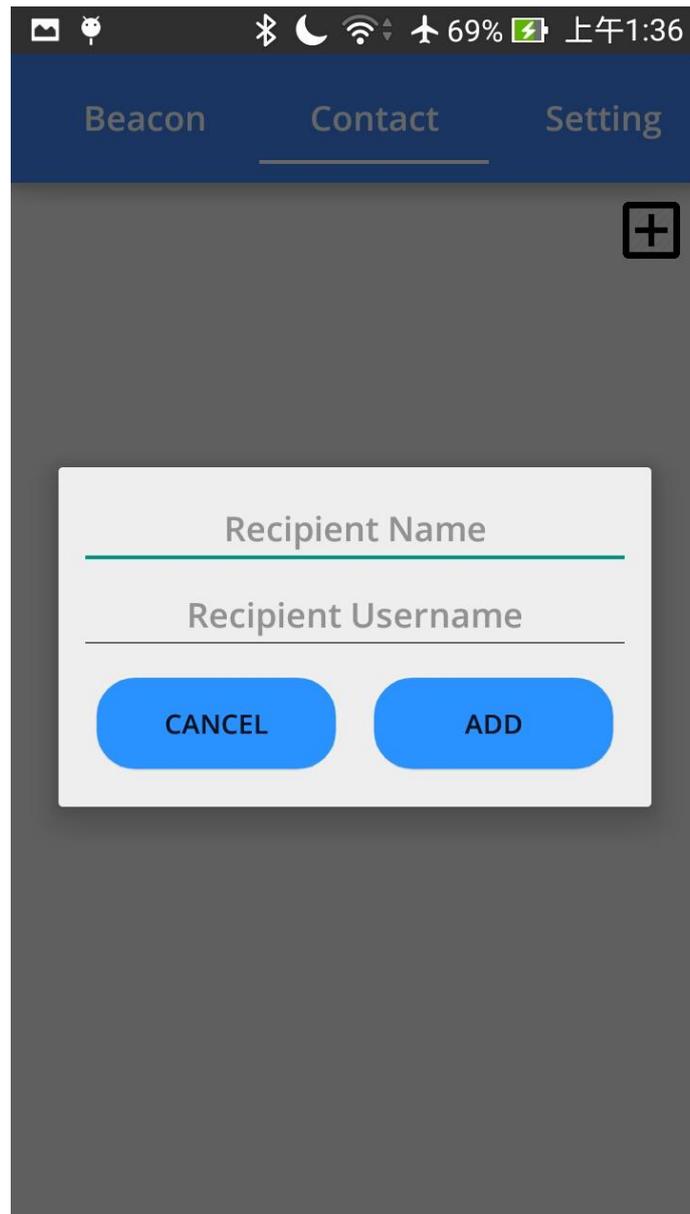


圖 21 連絡人介面

可在此頁面新增緊急連絡人

按下已新增的緊急聯絡人可撥打視訊電話

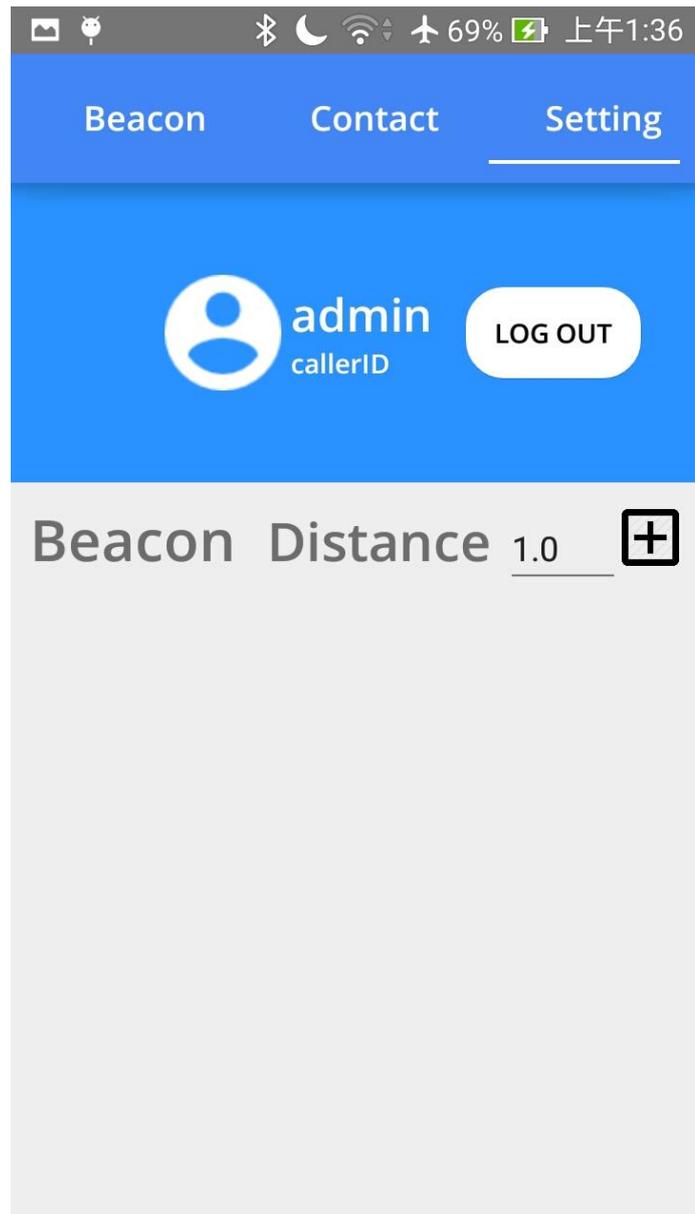


圖 22 設定介面

此介面可以登出並且能夠新增要被監控的小孩的 Beacon 和距離，亦考慮到了小孩不只一個，也可增加至超過一個。

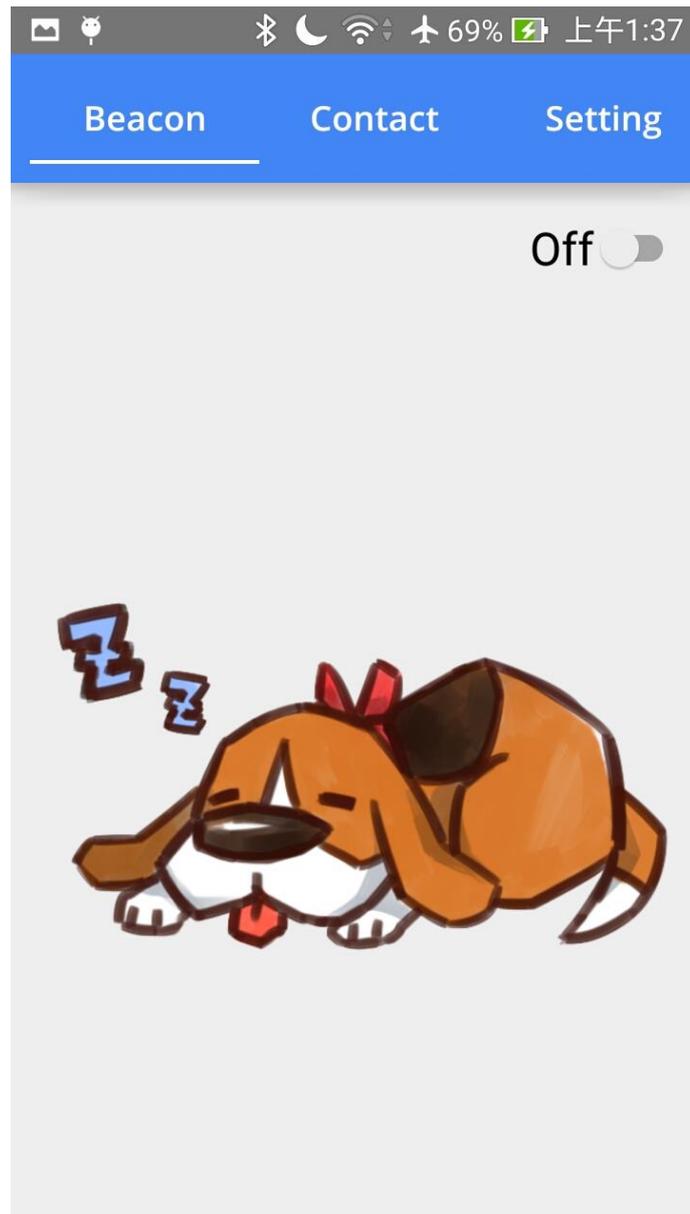


圖 23 監控介面 (一)

此介面能夠開啟關閉監控功能

上圖為未開啟時的畫面

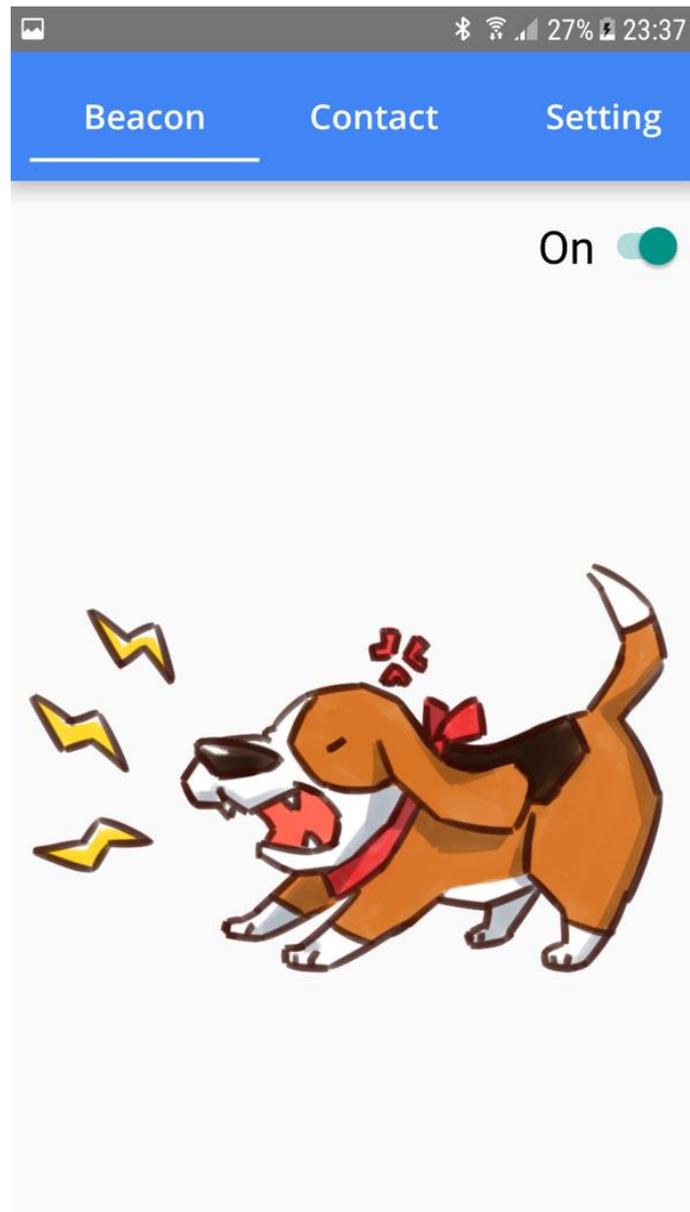


圖 24 監控介面 (二)

此介面能夠開啟關閉監控功能

上圖為開啟且已綁定的 Beacon 接近時的畫面，此時亦會發

出兇猛的狗叫聲以嚇阻小孩接近。



圖 25 接聽介面

當已綁定的 Beacon 接近時，家長端會收到此視訊來電，並且無論手機在任意模式皆會調整至鈴聲模式且調至最大音量。

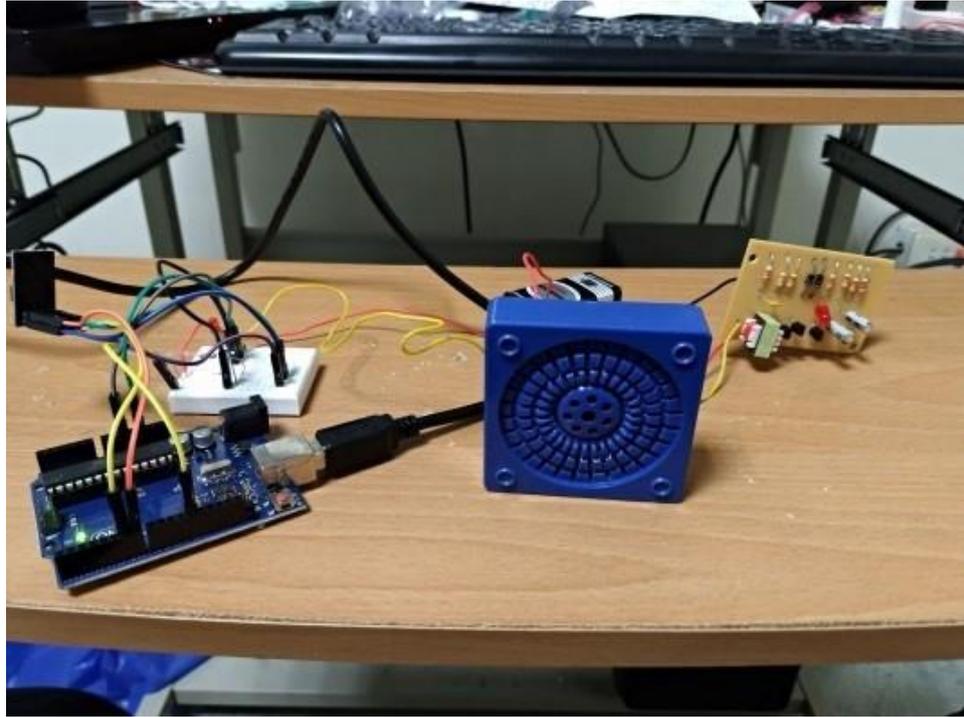


圖 26 家中擺放之蜂鳴器

已綁定的 Beacon 接近時，家中擺放的蜂鳴器亦會同時響起，以在第一時間告知在家中的家人。

## (十二) 未來展望

目前只有 Android 版本的，故希望能夠新增 iOS 版本，而 Raspberry Pi 的部分須使用者自行設定還太多，所以還有改良的空間，加強使用者經驗。

### (十三) 心得

本次計劃從一開始的一點點想法到耗費時間製作，最後完成，其中也有對此計劃能否如期的完成的不安感和同時的課業壓力，但看到它完成了，而且比當初的小想法還要多功能，真的感到十分有成就感，很多東西都是第一次實作，其中也有很多人協助我完成此計劃，感謝曾參與、協助本計劃的所有人，謝謝科技部能夠給予我此次機會。

## (十四) 參賽紀錄

智慧終端與人機互動軟體創作專題競賽

## (十五) 參考文獻

- [1] 張志勇、翁仲銘、石貴平、廖文華(2013)，物聯網概論，基峰。
- [2] Visual studio code，<https://code.visualstudio.com/>
- [3] Android Studio，<https://developer.android.com/studio>
- [4] Android, <https://www.android.com/>
- [5] Smartphone OS Market Share, International Data Corporation, <http://www.idc.com/promo/smartphone-market-share/os>
- [6] Matthew S. Gast(2014)，Building Applications with iBeacon: Proximity and Location Services with Bluetooth Low Energy，O'Reilly Media。
- [7] Eben Upton、Gareth Halfacree 著、曾吉弘 譯(2013)，Raspberry Pi 使用者手冊，馥林文化。
- [8] 孫駿榮、蘇海永(2015)，用 Arduino 全面打造物聯網，基峰。
- [9] 孫宏明，完整學會 Git, GitHub, Git Server 的 24 堂課(第二版)
- [10] Passbeemedia，<https://www.passbeemedia.com/ibeacon/>。
- [11] MQTT, <http://mqtt.org/>。
- [12] 高淑珍、吳建興(2012)。資料庫管理系統概論。藍海文化。
- [13] 朝井淳、陳奕苓(2007)。SQL 語法範例辭典。旗標。
- [14] MongoDB, <https://www.mongodb.com/>。
- [15] Elasticsearch, <https://www.elastic.co/>。
- [16] Cassandra, <http://cassandra.apache.org/>。
- [17] Lineabe Smartband，<http://www.lineable.net/englishsmartband>。