



# FB-Anti-unsent

非修課紀錄之成果作品

學校：臺北市立中正高級中學

---

班級：311

---

姓名：曾彥熹

---



# 目錄

前言.....	3
成品展示 .....	4
程式碼簡介 .....	7
程式碼連結 .....	10
遇到的困難與解決方式.....	11
學習省思 .....	13
參考資料 .....	14

## 前言

我平時和比較熟的朋友有一些群組，常常在 FB 上面聊天，在 FB 有了收回功能後，我們聊天之間也多了一個效果是用收回來表現，但那樣的話萬一有人沒看到就再也看不到了，於是我想要看到那些被收回的訊息（畢竟那些不是真正要收回的，只是為了達到效果），想試試看能不能寫出一個程式做到這個功能。

### 製作動機

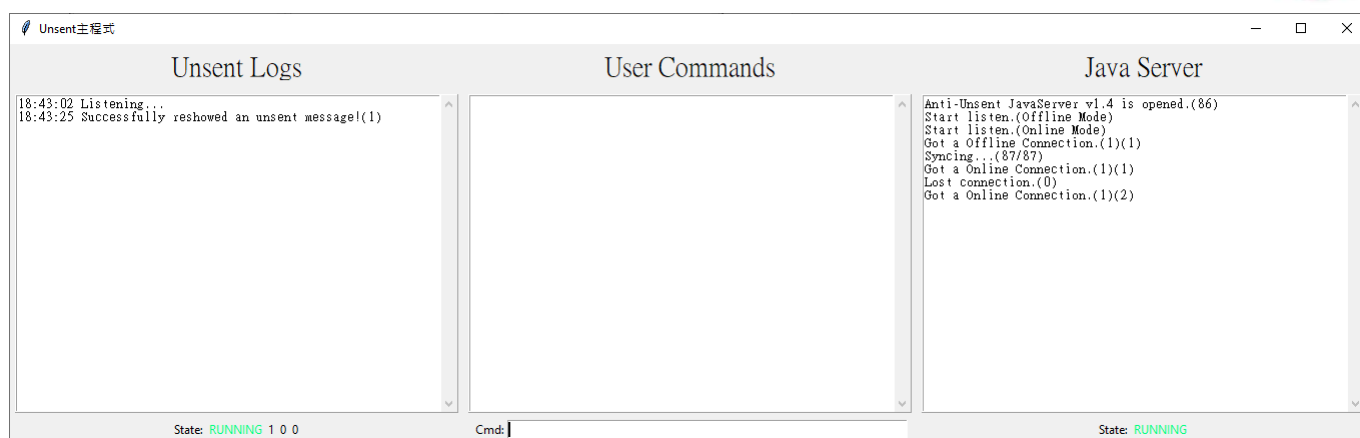
### 運作原理

利用 API 聆聽事件，當有人傳訊息時在電腦端進行緩存，當有人收回訊息時利用訊息 ID 搜尋緩存中的原始訊息，即實現了將收回的訊息還原的功能，在此功能上進行 UI 製作與連接至手機端，形成了完整的專案。

預期只要伺服器端能正常監聽訊息的事件且手機端可以正確接收伺服器端的資料即可，並進行一些 UI 上面的優化。

### 預設成果

## 成果展示



▲Python 伺服器介面(電腦)



▲客戶端介面(手機)-首頁



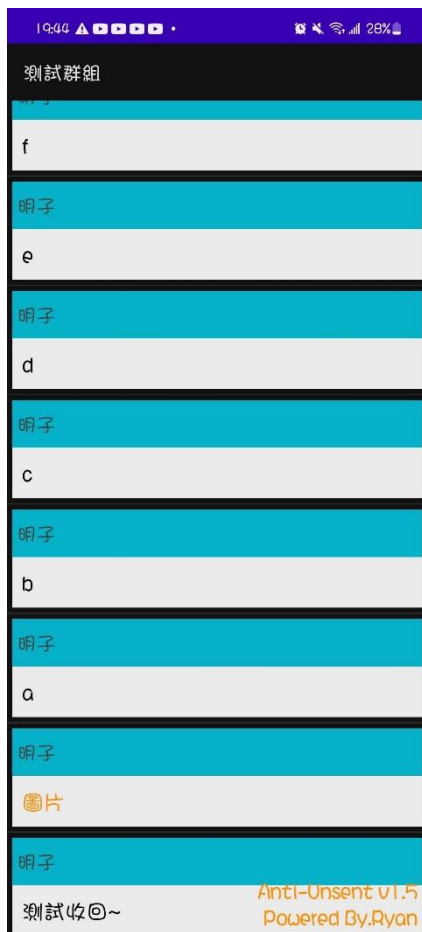
▲客戶端介面(手機)-設定頁面

## 成果展示

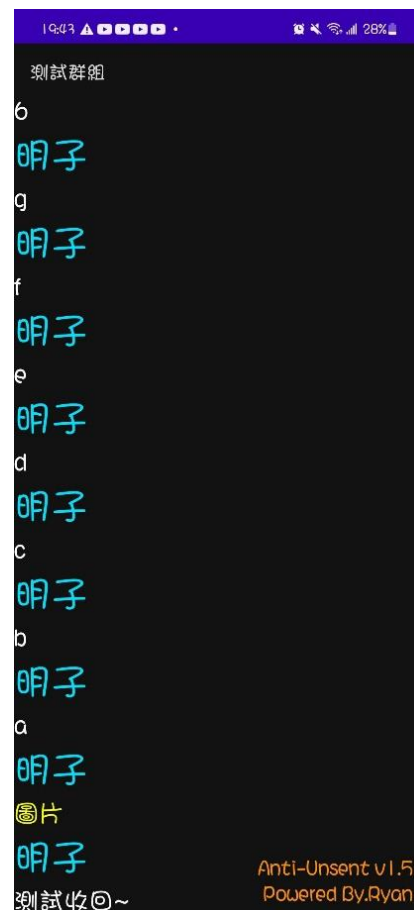
查看訊息介面有兩種版本，新介面的設計更有現代感，而舊介面只是單純的字串堆疊而已。兩者皆為輕觸訊息可查看詳細資料，而圖片長按才是查看詳細資料，短按則為預覽，如右圖。



▲客戶端介面(手機)▲  
預覽收回的圖片



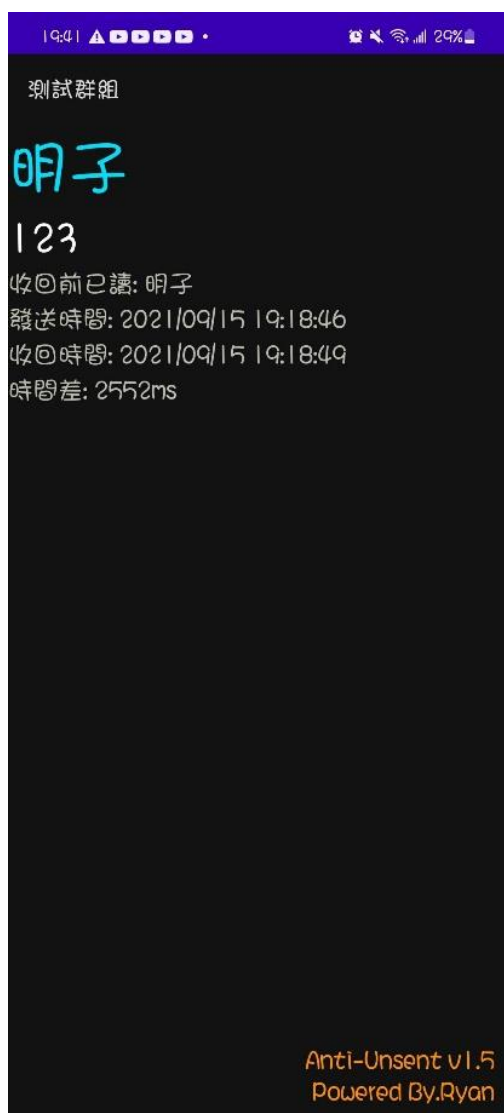
▲客戶端介面(手機)▲  
被收回的訊息們(新介面)



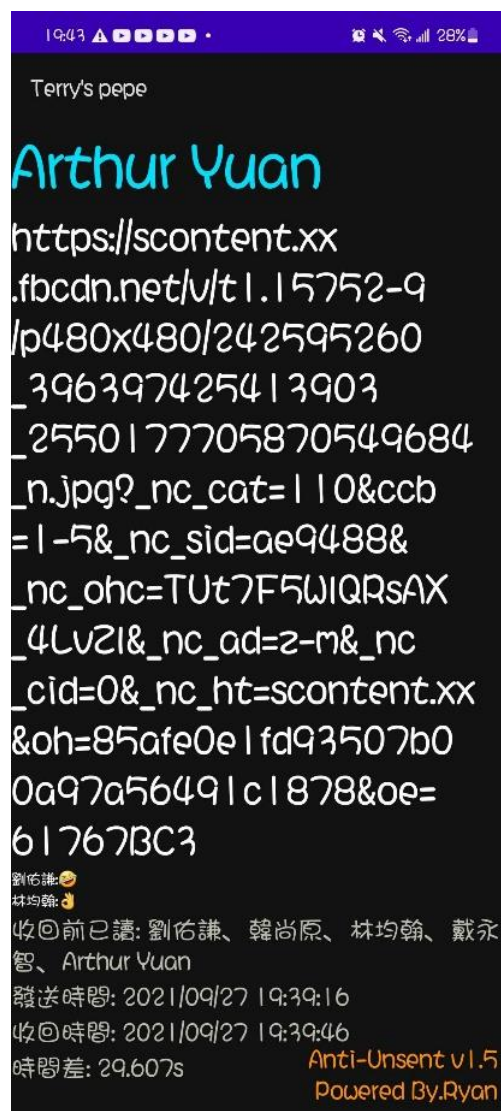
▲客戶端介面(手機)▲  
被收回的訊息們(舊介面)

## 成果展示

訊息的詳細資料包含作者、內容、有誰已讀、有誰給此訊息按了什麼表情、傳送及收回的時間，輕觸文字可以複製。



▲客戶端介面(手機)-範例收回訊息 1



▲客戶端介面(手機)-範例收回訊息 2

## 程式碼簡介

利用 API 收到收回的訊息後，將資訊做處理整合成一字串儲存，並傳送至 Java 伺服器端

```
with open(fp, 'a', encoding='utf8') as f:
    text += author.name+':'
    if message.text:
        text += message.text+"🗨️"
    if message.sticker:
        text += str(message.sticker.url)+"🗨️"
    if unsendable_files:
        for url in unsendable_files:
            text += str(url)+'🗨️'
    if files:
        for url in files:
            text += str(url)+'🗨️'
    if message.replied_to != None:
        rmsg = message.replied_to
        ran = self.R_FTI(rmsg.author).name
        text += ran + ":" + "🗨️" + self.handleM(rmsg)#(rmsg.text if rmsg.text != None else "None")
    text += "🗨️"
    if len(message.reactions) > 0:
        for k, v in message.reactions.items():
            text += self.R_FTI(k).name + ":" + str(v) + "🗨️"
        text = text[:-1]
    text += "🗨️"
    hasany = False
    for id in read:
        text += self.R_FTI(id).name+"`"
        hasany = True
    if hasany:
        text = text[:-1]
    else:
        text += "無"
    text += "🗨️"
    text += str(message.timestamp)+"🗨️"+str(ts)
    text = text.replace("\n", "🗨️")
    text += '\n'
    f.write(text)
    self.count += 1
    self.Print("Successfully reshowed an unsent message!(" + str(self.count) + ")")
    print("Successfully reshowed an unsent message!(" + str(self.count) + ")(with delay " + str(datetime
left = None
self.window.Update(0, self.count)
self.window.addUnsent("add" + "🗨️" + thread_id + "🗨️" + text)
```

▲ Python 伺服器背景-處理收回的訊息

## 程式碼簡介

利用 I/O 做程式之間的溝通，把 Python 伺服端收到的訊息傳至 Java 伺服端，在進行解析後儲存，客戶端上線時便可同步收回的內容

```
while(s.hasNext()) {
    String line = s.nextLine();
    if(line.isEmpty() || line.equals(" ") || line.equals("\n"))
        continue;
    if(line.equals("ExitClose"))
        break;
    //Write object to file(ready for app)
    BufferedWriter bw = null;
    try {
        bw = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(getContents(mess_c), true), "UTF-8"));
        bw.write(line + '\n');
    } catch (IOException e1) {
        e1.printStackTrace();
    } finally {
        try {
            bw.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    //send to app
    for(Client off : ass) {
        if(off.isSync())
            try {
                DataOutputStream os = new DataOutputStream(off.s.getOutputStream());
                if(line.startsWith("add"))
                    os.writeUTF(line + "@" + sent_index);
                else if(line.startsWith("ct"))
                    os.writeUTF(line);
                else
                    os.writeUTF(line);
                os.flush();
            } catch (IOException e) {
                System.out.println("Error on sending unsent-messages.");
                off.setSync(false);
                e.printStackTrace();
            }
        else
            System.out.println("It's not sync yet.");
    }
    sent_index++;
    if(sent_index > MAX_LINES) {
        sent_index = 0;
        File new_contents = new File(getContents(mess_c + 1));
        try {
            new_contents.createNewFile();
        } catch (IOException e) {
            e.printStackTrace();
        }
        update_mess();
    }
}
```

▲ Java 伺服端背景-處理收回的訊息



## 程式碼簡介

利用網路連線讓手機端與電腦端通訊，將文字解析後進行對應操作

```
String utf = in.readUTF();
String[] line = utf.split("@");
switch (line[0]) {
    case "add": {
        File f = new File(prefix_m + line[1] + ".txt");
        if (!f.exists()) {
            if (!f.getParentFile().exists())
                f.getParentFile().mkdirs();
            f.createNewFile();
            o.writeUTF("gn@" + line[1]);
        }
        BufferedWriter m = getUTF8Writer(f);
        if (!thread_ids.contains(line[1])){
            thread_ids.add(line[1]);
        }
        m.write(line[2] + '\n');
        m.flush();
        m.close();
        if (Integer.parseInt(line[3]) != receive_index++) {
            Log.d("WARNING", "index not same(" + receive_index + ")");
            SYNC = false;
        }
        if(receive_index > MAX_LINES)
            receive_index = 0;
        try {
            OutputStreamWriter os = new OutputStreamWriter(getApplication().openFileOutput(config, Context.MODE_PRIVATE));
            os.write(receive_index + "");
            os.flush();
            os.close();
        } catch (IOException e2) {
            Log.e("Exception", "File write failed: " + e2.toString());
            HandleError(e2);
        }
        break;
    }
    case "ct"://change title
    {
        String txt = prefix_m + line[1] + ".txt";
        List<String> lines = Files.readAllLines(Paths.get(txt), StandardCharsets.UTF_8);
        lines.remove(0);
        lines.add(0, line[2]);
        Files.write(Paths.get(txt), lines, StandardCharsets.UTF_8);
        break;
    }
    case "nt"://new title
    {
        String txt = prefix_m + line[1] + ".txt";
        List<String> lines = Files.readAllLines(Paths.get(txt), StandardCharsets.UTF_8);
        lines.add(0, line[2]);
        Files.write(Paths.get(txt), lines, StandardCharsets.UTF_8);
        break;
    }
    case "update": {
        if (Integer.parseInt(line[1]) == receive_index) {
            o.writeUTF("synched");
            o.flush();
        }
        break;
    }
}
```

▲ Java 客戶端背景-處理收回的訊息

此專案從 2021 年 12 月起利用課餘時間陸續的修修改改直到 2022 年 1 月寒假完工，停止維護。程式碼使用了兩種語言(Java 和 Python)，也跨了兩種平台(Windows 和 Android)，程式碼共分成三種：主程式 UI、電腦伺服器端、手機客戶端，完整程式碼連結：

<https://gitfront.io/r/user-8943678/7BypbIMDg4Fi/Anti-Unsent/>



The screenshot shows a GitHub repository interface. At the top, it says 'ht177970 Create README.md' with a commit hash '4af2798' and the date 'on Apr 15, 2021'. It also indicates '2 commits'. Below this is a list of files and folders:

File/Folder	Version	Last Commit
Client_Android-Java	v1.5	2 years ago
MainServer-Python	v1.5	2 years ago
ToClient-Java	v1.5	2 years ago
README.md	Create README.md	2 years ago

Below the file list is the 'README.md' content. It starts with the title 'Anti-Unsent v1.5'. The content is organized into three sections, each with a bullet point and a description:

- MainServer-Python  
電腦主程式，有UI介面，由Python製作。大部分的事情都在此處理(像是訊息處理、指令處理、連線處理等)
- toClient-Java  
電腦伺服器端，無介面(可用cmd)，由Java製作。用來橋接Python到客戶端手機Java的中間端
- Client\_Android-Java  
手機主程式，有UI介面，由Java製作。為觀看收回訊息的主要端，需搭配電腦端地開啟才能運作即時

▲ 專案完整原始碼

## 製作中的困難與解決方式

整個專案的製作過程中有遇到各種大大小小的問題，小的可能像是語法問題，由於我在這之前是完全不會 Python 的，非常仰賴網路上的解答與其他語言的經驗，接下來僅列出大問題或是很有收穫的問題。

問題簡述	困難說明與解決方式
fbchat 無法如期運行	<p>某次使用該 API 發現無法登入，會出現下列訊息</p> <pre>Attempt #1 failed, retrying Traceback (most recent call last):   File "C:\Users\Ryan\AppData\Local\Programs\Python\Python39\lib\site-packages\fbchat\_client.py", line 206, in login     self._state = State.login(   File "C:\Users\Ryan\AppData\Local\Programs\Python\Python39\lib\site-packages\fbchat\_state.py", line 147, in login     return cls.from_session(session=session)   File "C:\Users\Ryan\AppData\Local\Programs\Python\Python39\lib\site-packages\fbchat\_state.py", line 189, in from_session     fb_dtsg = FB_DTSG_REGEX.search(r.text).group(1) AttributeError: 'NoneType' object has no attribute 'group' Attempt #2 failed, retrying Traceback (most recent call last):   File "C:\Users\Ryan\AppData\Local\Programs\Python\Python39\lib\site-packages\fbchat\_client.py", line 206, in login     self._state = State.login(   File "C:\Users\Ryan\AppData\Local\Programs\Python\Python39\lib\site-packages\fbchat\_state.py", line 147, in login     return cls.from_session(session=session)   File "C:\Users\Ryan\AppData\Local\Programs\Python\Python39\lib\site-packages\fbchat\_state.py", line 189, in from_session     fb_dtsg = FB_DTSG_REGEX.search(r.text).group(1) AttributeError: 'NoneType' object has no attribute 'group'</pre> <p>研究一下發現此 API 早已停止維護，並且網路上也有許多相同的問題，原因是 FB 為了防止爬蟲，做任何該網頁的操作需附帶 data 中名為 fb_dtsg 的數值才能正常運作，而錯誤表明爬蟲找不到該檢核值，跳出錯誤無法繼續運行，進行了查詢與嘗試沒有得到解決，只好試著自己解決看看，開始循著錯誤點查看原始碼，了解 API 在該處在做的事，從電腦中模擬執行，可是我在 F12 狀態下無法從 FB 電腦版網頁中找到 fb_dtsg，又 API 登入時使用的是手機版 FB 網站，我便因此在那找到了我們需要的值，所以將原本爬蟲從 <a href="https://www.facebook.com/">https://www.facebook.com/</a> 抓的 fb_dtsg 程式碼片段改成從 <a href="https://m.facebook.com/">https://m.facebook.com/</a> 抓取後，就可以順利爬蟲了，將此問題解決。</p>

## 製作中的困難與解決方式

問題簡述	困難說明與解決方式
fbchat 無法如期運行	<p>不久後又出現了另一個問題</p> <p>似乎是因為頻繁重複登入導致的錯誤，被要求驗證身分</p> <pre>Traceback (most recent call last): File "/home/Paullux/.local/lib/python3.8/site-packages/fbchat/_client.py", line 205, in login self._state = State.login( File "/home/Paullux/.local/lib/python3.8/site-packages/fbchat/_state.py", line 153, in login raise _exception.FBchatUserError( fbchat._exception.FBchatUserError: Login failed. Check email/password. (Failed on url: https://m.facebook.com/login/checkpoint/)</pre> <p>並且該錯誤發生後，FB 帳號會直接被要求必須更改密碼，這導致 debug 難度大幅上升，上網搜尋也未找到解答，但大概了解可能是爬蟲模擬上出了問題，為了把爬蟲模擬的像用戶開網頁，通常會加入下列程式碼去修改 header</p> <pre>session.headers["Referer"] = "https://www.facebook.com" session.headers["Accept"] = "text/html" session.headers["User-Agent"] = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36"</pre> <p>在嘗試過多種版本的 User-Agent 後，仍然沒有解決此問題，最後直接把這三行都刪掉，卻意外地解決了這個問題！是一個非常特別的經驗</p>
傳輸資料過多	<p>由於原本採取的策略是每次手機想看收回的訊息時，電腦端會一次把所有儲存的收回訊息都合成一字串一次傳送，而用久了之後卻突然壞掉了，回家檢查後是因為字串長度太大，超過一次可傳送的限制。後來想到可以像 FB 訊息一樣，一則一則的方式儲存在手機裡，這樣可以節省每次傳輸所有資訊的流量，又可以避免出現超過一次傳輸長度限制。</p>
APP 無法背景運行	<p>在執行上述的新策略時，想要模仿 FB 訊息可以在背景接收，卻因為 Google 新政策不允許 Service 任意在背景執行，經過了很多的查詢與嘗試仍然沒有解決此問題，最終改成開啟 APP 時主動同步與電腦端的資訊落差的部分，想法是給訊息作編號，連線時進行編號確認，利用編號差來同步沒有收到的訊息。</p> <p>雖然沒有做到能夠及時背景收到訊息，但完全不影響使用，所以之後也沒解決無法背景運行的問題。</p>

在網路上進行一些搜尋後，找到現有的 API 可以使用，於是決定 Python 為主要架構開始寫程式，利用網路資源從 0 開始學習 Python，一路從純 cmd 視窗到有 tkinter UI，我想是多虧在這之前早已自學過 Java 與 C++ 語言，又這些語言的共通點多，學起來特別的快。而後又為了能夠在手機上也看到這些東西，才多了 Java 架構的部分，也因此多學到以前沒用過的 Socket 連線。中間也一度苦惱如何把 Python 收到的訊息給 Java 伺服器端，原本打算利用儲存檔案的修改時間去找收回的訊息的順序，但由於無論理論或實作都相當複雜，最後才想到可以用子程序的方式，利用內建 I/O 可以通訊，只需要小心用 bytes 處理訊息即可，因為兩者是不同的程式語言，有著不一樣的編碼，不能直接以字串的方式讀取。

整份專案製作過程中勢必還有遇到很多其他問題，像是 Python 語法錯誤、Python 如何使用 substr、Python 如何類型轉換等小問題，好在網路上的資源豐富，我利用英文搜尋大多都能找到答案，而由於小問題實在太多且通常找到答案後便能解決，故這邊不一一表列，而是將有幫助到我的所有頁面附在參考資料。

## 製作過程

## 習得能力

透過這次專案的製作，我認為我收穫很大，因為這份專案讓我多學會了一個程式語言 Python，也是我第一次製作一款專案是跨了語言、跨了平台，對我而言是件不簡單的事。

我強化了自己 Java 能力、寫 APP 程式的能力、Python 能力和問題解決的能力，也學到了不同程式語言間橋接的技術，由於過往都是用某單一程式語言進行設計到底，這次的製作有一種很不一樣的感覺。

經過一下與二上遊戲設計的練習後，回頭再看此專案，我認為程式碼風格的部分有很大可以進步的空間，由於當時並沒有物件導向的概念，需要什麼功能加什麼，導致出現了一個函式做很多很多事情、一個 class 包含了太多內容、函式沒有分類等問題，這些狀況會造成日後閱讀與維護的困難。未來打算做 2.0 版本，預期能整合原 API 內容並修正，也將我的程式碼翻新，避免上述的問題程式碼出現，這樣子日後要再更新就會少很多麻煩。

## 未來展望

## 參考資料

How to play GIF in android. Retrived from  
<https://stackoverflow.com/questions/20416383/how-to-play-gif-in-android>

Java Socket Programming. Retrived from  
<https://www.digitalocean.com/community/tutorials/java-socket-programming-server-client>

How to create an always running service in Android. Retrived from  
<https://gist.github.com/varunon9/f2beec0a743c96708eb0ef971a9ff9cd>

auto changing background in android app. Retrived from  
<https://stackoverflow.com/questions/37148304/auto-changing-background-in-android-app>

Android: resizing an ImageView in the XML. Retrived from  
<https://stackoverflow.com/questions/6267733/android-resizing-an-imageview-in-the-xml>

How to get the user input in Java? Retrived from  
<https://stackoverflow.com/questions/5287538/how-to-get-the-user-input-in-java>

How I can get data using background service in android? Retrived from  
<https://stackoverflow.com/questions/26735405/how-i-can-get-data-using-background-service-in-android>

Right Click menu using Tkinter. Retrived from  
<https://www.geeksforgeeks.org/right-click-menu-using-tkinter/>

Different Windows in TKinter. Retrived from  
<https://stackoverflow.com/questions/28892492/different-windows-in-tkinter>

Fbchat. Retrived from <https://github.com/fbchat-dev/fbchat>