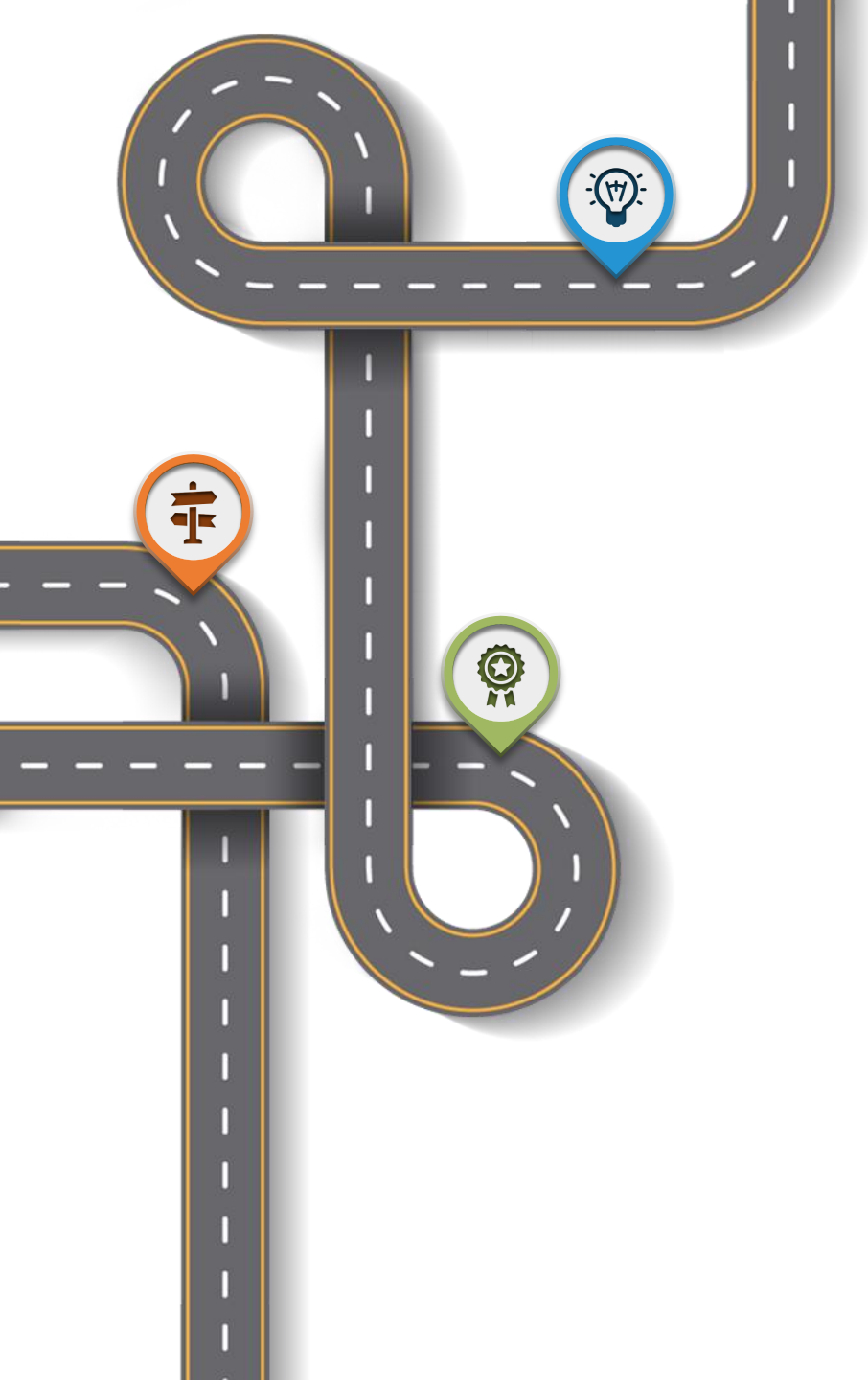


第三組

# 揪車Go

# Just Go

組員：劉語萱、陳柏尹、廖曉珺、洪鈺姍、柳映萱



# 目錄

---

- 01. 團隊介紹
- 02. 創作理念
- 03. 網頁佈告欄
- 04. 程式設計流程與架構
- 05. 應用到的課外技術
- 06. Line功能使用教學(Demo)



01



# 團隊介紹

---

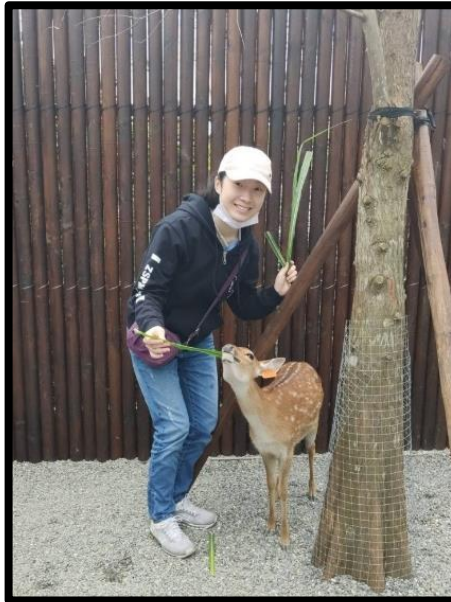
01

## 團隊介紹



巨資一A  
劉語萱

Art Designer  
Associate Engineer  
Line Bot Designer



巨資一A  
陳柏尹

Associate Engineer  
Web Designer



巨資一A  
廖曉珺

Main System Engineer

01

## 團隊介紹



企延B  
洪鈺嫻  
Planner  
System Engineer  
Database Engineer



巨資三A  
柳映萱  
System Engineer



02



## 創作理念

---



➤ 痛點：

1. 雖然有多班公車可到東吳外雙溪校區，但下車地點至校園仍常有段不算近的路程
2. 部分教學大樓位於山腰，557或專車最多也只開到校內平面，要到除了校門口那幾棟外的教學大樓仍需耗費一段腳程。
3. 校內的綜合大樓僅有的一部電梯，一旦接近上課時間排隊人潮便會增加，對於趕時間或行動較為不便的學生而言，簡直就是一大噩夢。
4. 若以計程車作為接駁工具，能直達目標大樓，但對一般的學生而言，此選擇的負擔較大。

➤ 解決方式：

士林捷運站至東吳雙溪的車資大約落在臺幣100~140元左右，倘若可以多召集一些擁有相同目的地的學生們一起共乘，不但可以減低個人負擔，也能讓社會資源達到更有效的利用。



1. 製作一個供東吳學生使用的P2P平台，匹配具有相同目的地的學生。  
【目的】用較低的時間與金錢成本擁有較高的生活品質
2. 使用者可以利用看板公告的功能查看即時的共乘需求，加入共乘後可知道發起人與計程車的相關訊息  
【目的】確保基本的搭乘安全。
3. 行程結束後會自動增加1點信用值。  
【目的】顯示在刊登頁面上以提高他人的匹配意願。







03



## 網頁佈告欄

---





**JUST GO**  
加翻共乘 相付相乘

歡迎使用揪車just go

確定發車(已訂車)

人數達標後發車(尚未訂車)

查看歷史紀錄

==使用教學==  
網址：[https://www.youtube.com/watch?v=bHd\\_7tuHBwI](https://www.youtube.com/watch?v=bHd_7tuHBwI)

==QA專區==  
Q：我想取消揪車，但取消時間已過，我還能取消嗎？  
A：您好，可取消時間是由發車人自行輸入的喔！  
如果超過時間，則會視為跑單，系統將扣除信用值或禁用帳號（規則如下：）  
第一次跑單將會暫停用戶使用帳號一個月，第二次三個月，第三次永久封號，還請注意喔！  
Q：我的學生證遺失，但我仍想認證帳號，請問有別的認證方法嗎？  
A：您好，考慮到在校身分確認，我們目前暫不開放東吳大學的學校email(學號@scu.edu.tw)以外的途徑認證喔，還請見諒。

==聯絡我們==

佈告欄功能：

1. 顯示三項大功能的即時資訊
2. 教學影片的連結
3. 簡易Q&A問答
4. 聯絡方式

註：目前網頁只可在電腦中開啟，但因為手機畫面的設定過多，期許未來可以將這個功能更加完善。



## 歡迎使用揪車just go

確定發車(已訂車)

人數達標後發車(尚未訂車)

查看歷史紀錄

==使用教學==

網址：[https://www.youtube.com/watch?v=bHd\\_7tuHBwI](https://www.youtube.com/watch?v=bHd_7tuHBwI)

==QA專區==

Q：我想取消揪車，但取消時間已過，我還能取消嗎？

A：您好，可取消時間是由發車人自行輸入的值喔！

如果超過時間，則會視為跑單，系統將扣除信用值或禁用帳號（規則如下：）

第一次跑單將會暫停用戶使用帳號一個月，第二次三個月，第三次永久封號，還請注意喔！

Q：我的學生證遺失，但我仍想認證帳號，請問有別的認證方法嗎？

A：您好，考慮到在校身分確認，我們目前暫不開放東吳大學的學校email(學號@scu.edu.tw)以外的途徑認證喔，還請見諒。

==聯絡我們==

## 歡迎使用揪車just go

確定發車(已訂車)

人數達標後發車(尚未訂車)

編號	P20210531001	編號	P20210601001
出發時間	13 : 30	出發時間	22 : 45
發起人/性別	柳同學/F	發起人/性別	柳同學/F
出發地	捷運劍南路站	出發地	捷運士林站
目的地	東吳大學城中校區	目的地	東吳大學外雙溪校區
上限人數	3	上限人數	3
發起人信用積分	9	發起人信用積分	9
備註	安靜&限同性	備註	安靜&無菸&限同性

即時資訊主要是依揪車型態分流兩張表，可讓使用者依據自身需求以最快速度方式尋找到合適的刊登資料，以達到便利搭乘的目的。

查看歷史紀錄			
編號	P20210520002	編號	P20210523004
車行	臺灣大車隊	車行	Uber
車牌號碼	TAC-3456	車牌號碼	TDP-7788
上車地	捷運士林站	上車地	捷運劍南路站
目的地	東吳大學城中校區	目的地	東吳大學外雙溪校區
行程總價	300	行程總價	155
編號	P20210529001	編號	P20210530002
車行	路邊攔車	車行	臺灣大車隊
車牌號碼	TDA-3209	車牌號碼	168-8E
上車地	捷運士林站	上車地	東吳大學城中校區
目的地	東吳大學城中校區	目的地	捷運士林站
行程總價	310	行程總價	290

查看歷史紀錄原理是利用使用者的學號生成專屬的網址，再將此發送給使用者，因此雖然即時刊登資訊都是共享的，但僅能看到自己的歷史紀錄，以讓使用者可以查詢其搭乘的歷程記錄，資訊主要有訂單編號、車行、車牌號碼、起訖點，及行程總價。



04



# 程式設計流程與架構

---

00

## 介面設計流程架構



### 會員

註冊、更改資料、查看信用值



### 即時資訊

即時刊登資訊、查看歷史記錄



### 我要揪車

一定要揪車、湊滿人才揪車



### 尋找共乘

一定要揪車、湊滿人才揪車



### 揪車情況/取消揪車

揪車現況、取消揪車、查看狀態



### 使用說明/QA/其他

使用說明、基本QA問答、  
其他聯絡資訊、



## 註冊

### ➤ 使用流程：

1. 會先判斷是否為新會員，若為舊會員則導引到更改資料的區塊。
2. 依序填入學號、姓名、系級、性別及密碼基本資料。



## 驗證

### ➤ 使用流程：

1. 系統會自動送出一封email的驗證信到該位學生的東吳信箱，內含6位數的驗證碼
2. 使用者須至Line Bot上回傳驗證訊息
3. 若回傳訊息與系統相符則顯示註冊成功且資料寫入雲端資料庫，
4. 若回傳訊息與系統不相符則須重新進行註冊流程。



## 更改資料

更改資料大致上跟註冊相似，不同點如下：

1. 可以自行決定要從哪一項資料開始更改，順序與註冊流程相同
2. 若是初步驗證發現錯誤，則需重新進行整個更改資料的流程。
3. 更改完成後不需進行信箱驗證，直接更新資料至雲端資料庫



## 查看信用值

讀取資料庫內該會員的信用值，並回傳給Line Bot以發送給使用者，以利其查詢至今使用平臺後的信用值。

註：信用值越高代表乘車次數越多，亦可初步判斷為長期客戶抑或是表現優良的使用者，未來若開發福利功能，可先向此群體發送。



## 02 即時資訊：刊登種類、建構即時刊登的下拉式網頁html



### 刊登 種類

- 依照刊登種類將訂單的即時資訊分成兩大類：
  1. 一定會發車
  2. 湊滿人才會發車



### 下拉式 網頁

1. 可用按鈕收縮相關資訊
2. 相關資訊為一欄兩列的表格。

【註】由於是先取得當下資料庫的資訊再以Python產生一個html網頁，因此若要取得之後的即時資訊須在執行一次取得即時資訊的流程。



當乘車者依序填完各項資訊後，我們將會特定條件向不同刊登種類的特定使用者傳送訊息。

➤ 最終結果：

- 一定要揪車：當達到最小乘車人數時向乘車者做最終確認。
- 湊滿人才揪車：當每多一人時向發起者告知目前人數。



當發起者依序填完各項資訊後，除了能接收到訂單的最新狀況外，還需要負責觸發訂單開始的功能。

➤ 確認流程：

1. 當輸入「車子已抵達」時，系統向發起人詢問實際乘車人數。
2. 若實際人數小於或與最低搭乘人數不符，則向所有搭乘者傳送「我已上車」。
3. 當系統接收到與實際乘車人數同數量的「我已上車」時，將自動判定未傳的使用者為跑單狀態，並依使用者的跑單次數進行停權的處罰。



當發起者輸入「共乘完畢」時觸發訂單結束的流程。

➤ 確認流程：

1. 發起者輸入總金額
2. 共乘者確認每人要付的實際金額
3. 若有共乘者回復「金額\_錯誤」則將回到步驟1
4. 當「金額\_正確」的回覆數量與乘車者數量一樣時，系統判斷訂單完成。



當共乘者收到此訂單每人需付的金額時，需對金額進行確認。

➤ 確認流程：

1. 金額正確：輸入「金額\_正確」
2. 金額錯誤：輸入「金額\_錯誤」



➤ 確認流程：

1. 傳送該訂單的可取消時間
2. 傳送信用值與警語
3. 傳送是否真的放棄的提問
4. 使用者回傳訊息後，傳送使用者狀態及使用者目前的信用值。

【註】執行步驟1及2的原因為達到告知取消搭乘需自行負擔後果的提醒義務，以免使用者未按取消，卻也未上車，造成信用值減分，甚至停權處置。



系統會依帳號是否被停權的狀態傳送兩種訊息。

➤ 最終結果：

- 帳號被停用：傳送含有最終停權日期的訊息。
- 帳號沒被停用：傳送使用者狀態，若有加入刊登則亦傳送訂單資訊給使用者。

【註】為了可以讓流程控制更加完善，從一開始即讓停權的使用者知道其目前無法使用服務，亦可讓搭乘的人知曉自己是否為共乘人，抑或是發起人，因為兩者的身分對於刊登的客製化些許不同，計算分擔價格，主揪者可以稍微分到些許的四捨五入小折扣。

## 06 使用說明/QA/其他：使用說明、基本QA問答、其他聯絡資訊



### 歡迎使用揪車just go

- 確定發車(已訂車)
- 人數達標後發車(尚未訂車)
- 查看歷史紀錄

==使用教學==

網址：<https://www.youtube.com/watch?v=hKWNcJh0EwE>

==QA專區==

Q：我想取消揪車，但取消時間已過，我還能取消嗎？  
A：您好，可取消時間是由發車人自行輸入的值喔！  
如果超過時間，則會視為跑單，系統將扣除信用值或禁用帳號（規則如下：）  
第一次跑單將會暫停用戶使用帳號一個月，第二次三個月，第三次永久封號，還請注意喔！

Q：我的學生證遺失，但我仍想認證帳號，請問有別的認證方法嗎？  
A：您好，考慮到在校身分確認，我們目前暫不開放東吳大學的學校email(學號@scu.edu.tw)以外的途徑認證喔，還請見諒。

==聯絡我們==

郵件：[justgocarpool@gmail.com](mailto:justgocarpool@gmail.com)



05



## 應用到的課外技術

---

## 01

## 利用Python傳送email給用戶



```
import smtplib
from email.mime.text import MIMEText
from email.header import Header
from email.mime.multipart import MIMEMultipart
from email.mime.application import MIMEApplication
from string import Template
from pathlib import Path
import random
```

- 運用smtplib套件，設定SMTP的連線伺服器。
- 應用Gmail的應用程式授權金鑰
- 撰寫基本的Html以設定郵件架構，並插入附件圖片以及內含一組六位數隨機驗證碼。
- 成功發送郵件給正進行註冊的用戶進行驗證



02

- [illegible]

- 若收到的回覆為文字訊息則將此訊息回傳給LINE帳號擁有者

```
@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    msg = event.message.text
    try:
        line_id=event.source.user_id
        line_bot_api.push_message('Uad5f20670570a6aad3f1f20c893fe5a2',TextSendMessage(text=line_id+'發了:'+msg))

    except:
        message = TextSendMessage(text="錯誤")
        line_bot_api.reply_message(event.reply_token, message)
        #line_bot_api.push_message(uu,TextSendMessage(text=uu+msg))
        if '會員' in msg:
            message = membersystem()
            line_bot_api.reply_message(event.reply_token, message)
        elif '尋找共乘' in msg:
            message = findjustgo()
            line_bot_api.reply_message(event.reply_token, message)
        elif '即時資訊' in msg:
```

- 將收到的回覆進行分流處理

```
elif '註冊' in msg:
    try:
        if line_id not in mydict:
            mydict[line_id]={}
            mydict[line_id]["register"]={}
            line_bot_api.push_message(line_id, TextSendMessage(text='檢測到此id未註冊過，進行註冊程序'))
        else:
            line_bot_api.push_message(line_id, TextSendMessage(text='檢測到此id註冊過，進行更改資料程序'))

    except:
        line_bot_api.push_message(line_id, TextSendMessage(text="失敗"))
        message = register(msg, line_id, mydict)
        line_bot_api.reply_message(event.reply_token, TextSendMessage(text=message))
```

- 將特定訊息回傳給特定使用者

## 03

## 架構網站：利用python生成網頁內容

```
def web():  
    if not database_connect():  
        print("連線未成功 網頁未成功開啟")  
    f=open('justgo.html','w',encoding='UTF-8')  
    head(f)  
    publish1=get_publish_D()  
    sure=[]  
    not_sure=[]  
    for i in list(publish1):  
        max_min=get_max_min(i[0])  
        if max_min[0]==0:  
            sure.append(i[0])  
        else:  
            not_sure.append(i[0])
```

用複寫檔案的方式，重新生成一個html檔

1. 抓取最新的刊登資料
2. 將資料進行分流

生成一個大表格並依資料數量生成相對應的小表格數量

```
if len(historylist)==0:  
    f.write("<span style='font-family:cwTeXYen';><font color='#FFFFFF'><h3><p>目前並無歷史紀錄<br></span>\n")  
else:  
    #title3(f)  
    for i in range(len(historylist)):  
        if i%2 ==0:  
            f.write(''  
<center><table border="0" width="800" style="border:0px #005AB5;">  
            <tr>  
                <td valign="top">\n''')  
            big_form(f,left)  
            history_detail=get_history(historylist[i])  
            left=historyform(f,left,history_detail)
```



## 03

## 架構網站：利用Flask架構網站

```
conn = ps.connect(host="...",
                  user="...",
                  password="...",
                  database="...",
                  port="5432")

if conn is not None:
    cursor = conn.cursor()
    conn.commit() # commit the changes

def get_name(uu):
    return

@app.route("/")
def home():
    return render_template("justgoexample.html", name=uu)

app.run()
```

- 將檔案與資料庫進行連線
- 利用Flask架構網站
- 將此檔案git push到該專案的heroku app上



06



# Line功能使用教學 (Demo)

---



感謝聆聽