

The background image shows a subway platform. On the left, a large red 'Exit' sign is visible. Below it, a digital display shows '2. 7 Flushing-Main' and '12:07 PM'. To the right, a train is stopped at the platform. The train has a blue and white color scheme with a red circle containing the number '7' on its front. The platform has a yellow tactile paving strip along the edge. The overall scene is slightly blurred and has a greenish tint.

台鐵時刻表查詢軟體

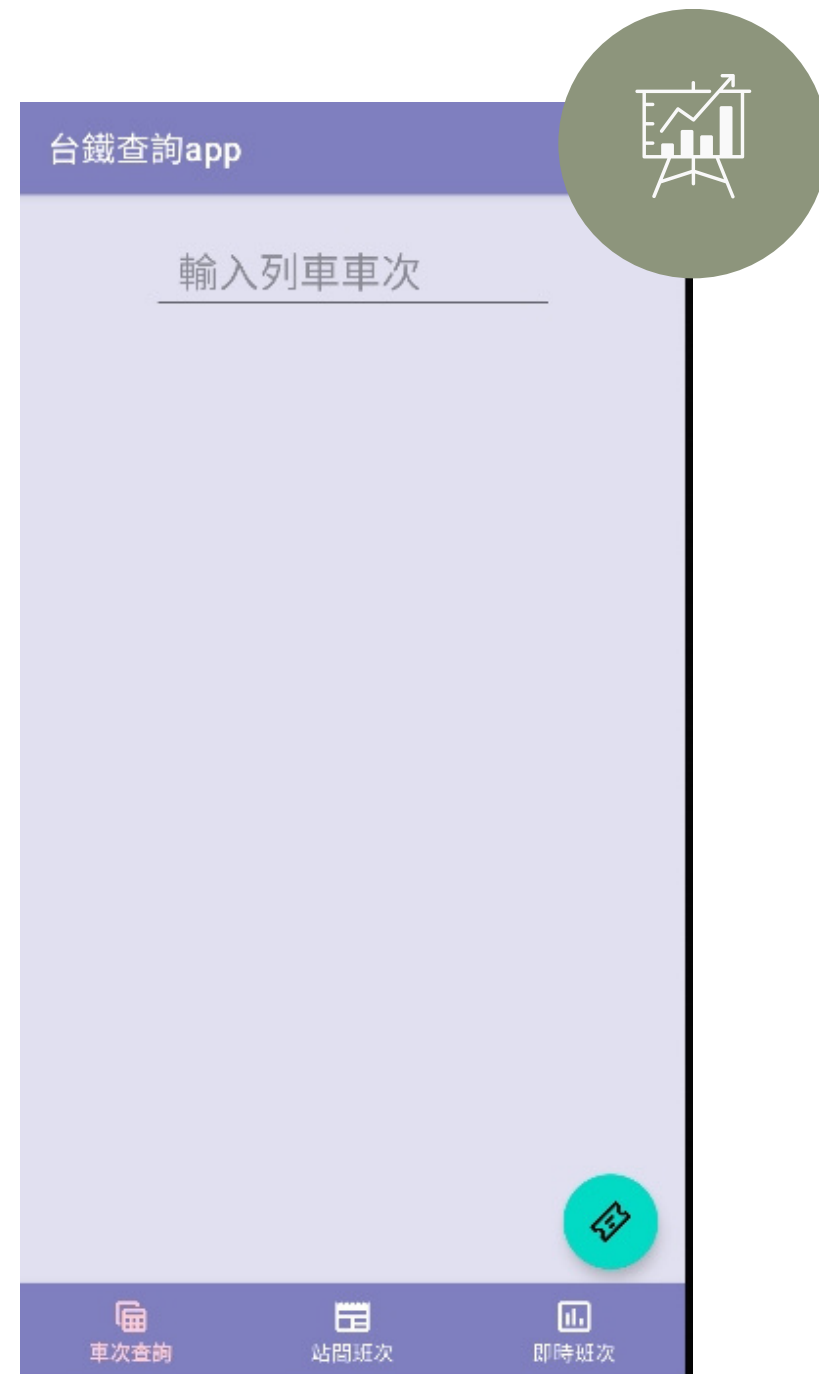
組員：詹士賢、黃振翔、王正楷



製作動機

因為平常很常搭乘火車往返彰化與家鄉，
所以想要試著寫寫看像台鐵e訂通一樣的APP，
可以透過這個程式查詢到各站點間的火車班次，
與某站點某段時間的時刻表。

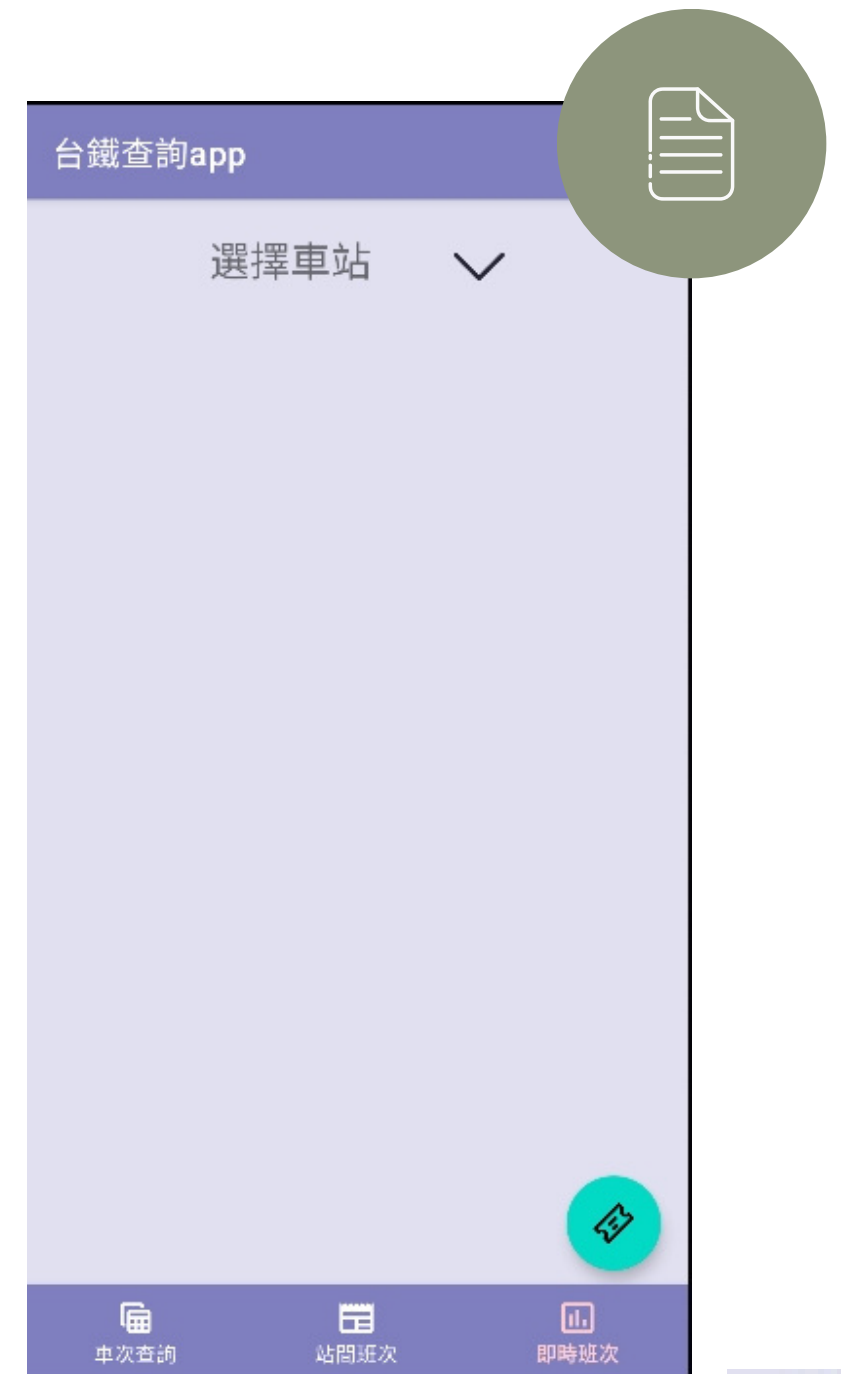
程式使用說明



車次查詢



站間班次



即時車次



輸入列車車次



車次查詢



站間班次



即時班次

車次查詢

輸入列車車次後
會跳出該台列車
之表定時間

2166

彰化	"10:34"	"10:34"
成功	"10:40"	"10:41"
新烏日	"10:44"	"10:45"
烏日	"10:47"	"10:47"
大慶	"10:50"	"10:51"
五權	"10:54"	"10:54"
臺中	"10:58"	"10:58"



車次查詢



站間班次



即時班次

選擇起點站



選擇終點站



選擇時間



確定



車次查詢



站間班次



即時班次

站間班次

選擇起點和終點站

再選擇搭乘日期

會跳出該日該站間的

所有班次

2114
彰化 -> 基隆
05:00 -> 08:44

102
斗六 -> 七堵
05:38 -> 08:16

2120
二水 -> 基隆
05:49 -> 09:36

2008
彰化 -> 基隆
05:58 -> 09:01

472
員林 -> 臺東
06:14 -> 08:37

2124
嘉義 -> 基隆
06:22 -> 10:09

278
彰化 -> 花蓮
06:48 -> 09:16

2134
嘉義 -> 基隆
07:00 -> 10:57

170
嘉義 -> 花蓮
07:05 -> 09:56

280
斗六 -> 花蓮



車次查詢



站間班次



即時班次

選擇車站



車次查詢



站間班次



即時班次

即時班次

選擇車站後會跳出
對應車站之即時車次

臺北



區間 2213 往嘉義
出發時間 13:51:00 誤點 0分鐘

區間 2164 往瑞芳
出發時間 13:50:00 誤點 1分鐘

自強 129 往枋寮
出發時間 14:00:00 誤點 0分鐘

自強 428 往新左營
出發時間 14:00:00 誤點 2分鐘

自強 116 往七堵
出發時間 14:04:00 誤點 0分鐘

自強 415 往樹林
出發時間 14:05:00 誤點 1分鐘



車次查詢



站間班次



即時班次



台鐵查詢app

447
臺東 -> 樹林
00:04 -> 00:12

2005
七堵 -> 彰化
05:15 -> 05:23

1107
七堵 -> 苗栗
05:26 -> 05:34

2133
基隆 -> 嘉義
05:38 -> 05:46

161
七堵 -> 臺東
05:52 -> 06:00

1117
基隆 -> 苗栗
06:04 -> 06:12

103
七堵 -> 潮州
06:19 -> 06:27

1121
基隆 -> 新竹
06:30 -> 06:38

4103
宜蘭 -> 北湖
06:39 -> 06:47

105
基隆 -> 屏東



車次查詢



站間班次



即時班次



確定要購買此票嗎?

取消

確定

1117
基隆 -> 苗栗
06:00 -> 06:17

447
臺東 -> 樹林
00:04 -> 00:12

確定要刪除這筆資料嗎

取消

確定

台鐵查詢app

1117
基隆 -> 苗栗
06:00 -> 06:17



紀錄車次

程式製作流程

取得ACCESS TOKEN



01

放進URL

02

製作車次查詢功能



03

製作站間車次功能



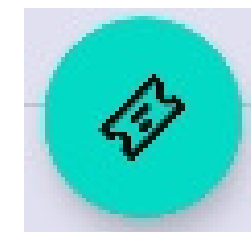
04

製作即時班次功能



05

製作儲存車次功能



06

取得ACCESS TOKEN



01

```
fun AccessToken(callback: (accessToken: String?) -> Unit){  
    val client = OkHttpClient()  
    val url = "https://tdx.transportdata.tw/auth/realms/TDX"  
    val requestBody = FormBody.Builder()  
        .add(name: "grant_type", value: "client_credentials")  
        .add(name: "client_id", value: "  
        .add(name: "client_secret", value: "  
        .build()  
    val request = Request.Builder()  
        .url(url)  
        .addHeader(name: "content-type", value: "application/x-www-form-urlencoded")  
        .post(requestBody)  
        .build()
```

Available authorizations

x

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.

API requires the following scopes. Select which ones you want to grant to Swagger UI.

TDX (OAuth2, clientCredentials)

TDX會員可至【會員專區-資料服務-服務金鑰】功能頁面，從預設金鑰(或建立新的金鑰)取得Client Id和Client Secret資訊，分別輸入至下方client_id和client_secret欄位、按下Authorize按鈕，即可開始於Swagger介面呼叫TDX API。

Token URL: <https://tdx.transportdata.tw/auth/realms/TDXConnect/protocol/openid-connect/token>

Flow: `clientCredentials`

client_id:

[Redacted]

client_secret:

.....

Authorize

Close

放進URL

02

```
val TrainTimeTableList: MutableList<String> = mutableListOf()
val accessToken = (activity as? MainActivity)?.accessToken
val client = OkHttpClient()
val url =
    "https://tdx.transportdata.tw/api/basic/v3/Rail/TRA/GeneralTrainTimetable/TrainNo/${stationID}?%24format=JSON"
val request = Request.Builder()
    .url(url)
    .addHeader( name: "accept", value: "application/json")
    .addHeader(
        name: "Authorization",
        value: "Bearer $accessToken"
    )
    .build()
```

```
AccessToken { accessToken ->
```

```
    this@MainActivity.accessToken = accessToken
    val firstFragment = FirstFragment.newInstance( param1: "param1", param2: "param2")
    val secondFragment = SecondFragment.newInstance( param1: "param1", param2: "param2")
    val thirdFragment = ThirdFragment.newInstance( param1: "param1", param2: "param2")
    supportFragmentManager.beginTransaction()
        .replace(R.id.activity_main_nav_fragment, thirdFragment)
        .add(R.id.activity_main_nav_fragment, secondFragment)
        .add(R.id.activity_main_nav_fragment, firstFragment)
        .commit()
}
```



製作車次查詢功能



03

- 1.放editText
- 2.把editText限定數字鍵盤
- 3.輸入後會跳出listView

選擇起點站



選擇終點站



選擇時間



確定



車次查詢



站間班次



即時班次

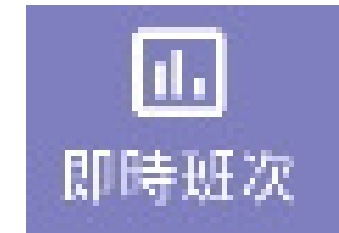


站間班次

04

- 1.放三個textView
- 2.監聽他們的點擊
- 3.點選前兩個textView會跳出清單
- 4.點選第三個會跳出選擇日期
- 5.建立並監聽確定按鈕
- 6.點擊確定按鈕後會關掉其他元件並跳出listview

輸入列車車次



- 1.放textView並監聽點擊
- 2.點擊textView後會跳出車站選單
- 3.選擇後會跳出listView

1117

基隆

->

苗栗

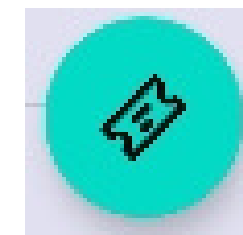
06:00

->

06:17



製作儲存車次功能



06

1. 建立textView並顯示它
2. 建立floatingActionButton做跳回主頁面的功能
3. 點選listView中的選項後可以刪除該筆資料

程式優化

車站選擇

01

會員登入功能

02

版面美化

03

車輛動態

04



結語

summary



分工內容

詹士賢

程式美觀、
書面報告

黃振翔

程式架構、
API串接

王正楷

製作簡報、
書面報告