

【1308】

資料庫專題

火車訂票系統

組別:B04

組員: D0746180 林則穎 (組長)

D0746056 陳宣燁

D0776706 林聖閔

D0746116 鄭志軒

教授:林明言

一. 主題：

火車訂票系統資料庫

二. 專題目的：

以台鐵火車訂票系統為例，建立出一個資料庫系統，旅客可以依自己的需求買票，可以用線上方式查看到車站、班次、到站、離站時間以及所需金額，提供更多元化的選擇，並在網站上直接購票，減少需現場購票時間，更不必擔心買不到票問題，也提供查詢訂票的功能，及時的知道自己的購票資訊，隨時都能取消車票，增加旅客的方便性確保旅程更加順利。

三. 系統使用者：

需查詢火車班次或有購票需求的旅客

四. 使用者需求功能：

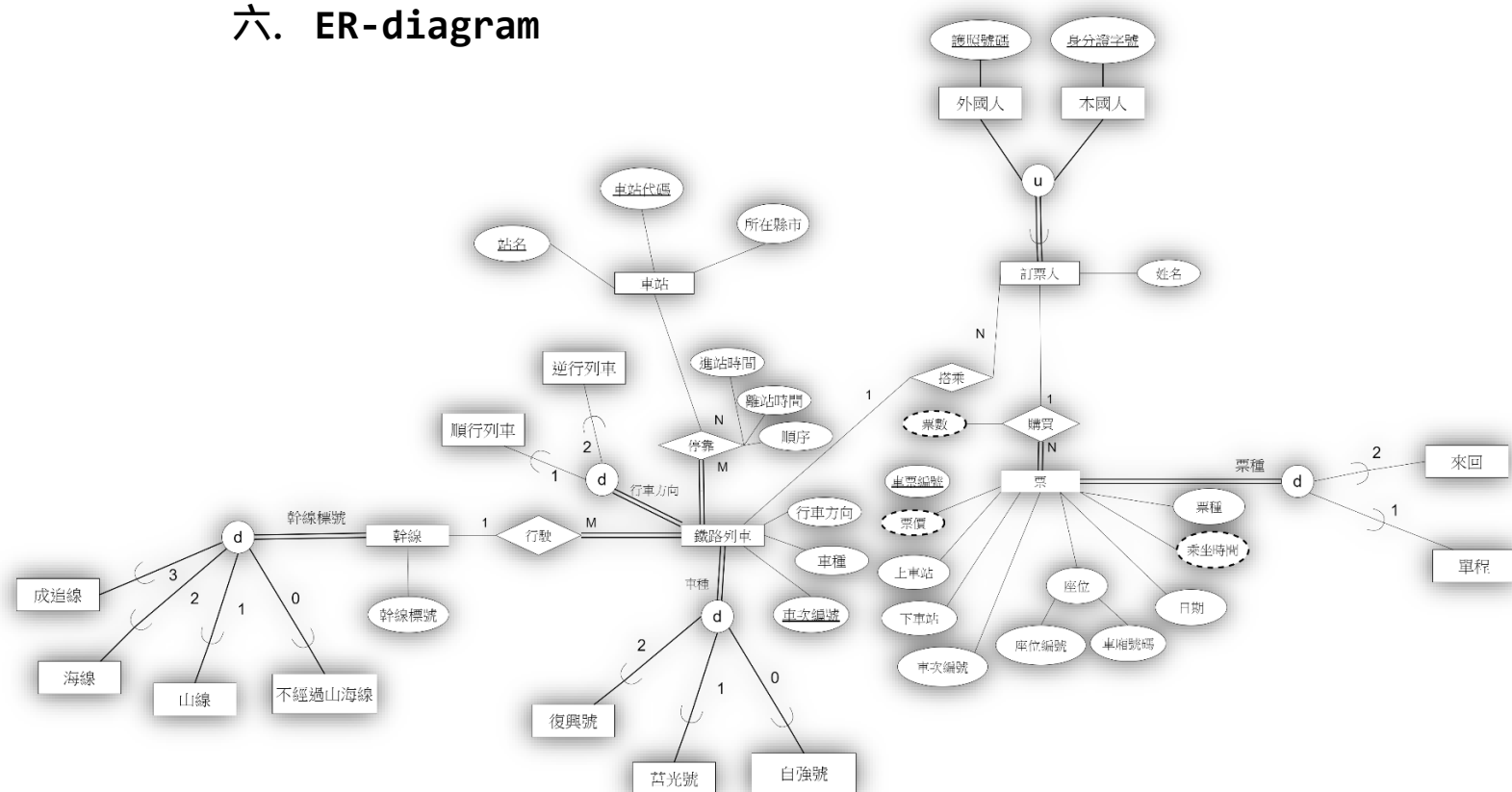
使用者輸入火車出發站以及抵達站，列出所有火車班次以及它的到站、離站時間、所需金額，選擇火車班次輸入使用者資料完成購票功能，輸入車票編號查詢車票資訊，取消車票需輸入原身分證字號/護照碼照，即可取消車票。

。

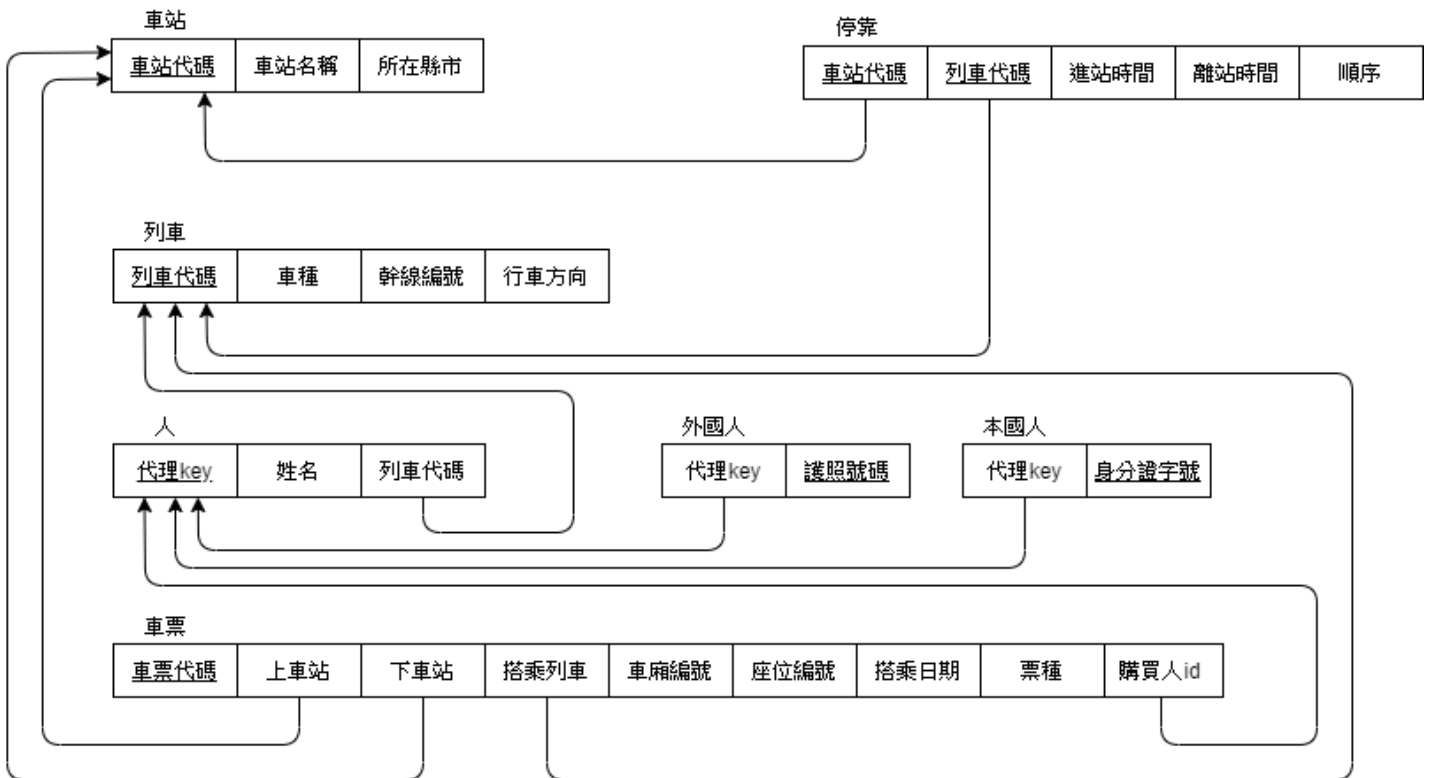
五. 輸入資料、輸出資料

輸入資料	輸出資料
出發站、抵達站、出發時間	火車車種班次、出發時間、抵達時間、經由(山線、海線、不經山海線)、進站時間、離站時間、金額
身分證字號、護照碼照、姓名、單程/來回、日期、車次、票數	訂票成功、訂票編號
車票編號	搭乘時間、車種車次、單程/來回、起始車站、終點車站、發車時間、終點時間、座位幾車幾號、金額、車票編號
車票編號、身分證字號/護照碼號	取消成功/取消失敗(身分證字號/護照碼號不符)

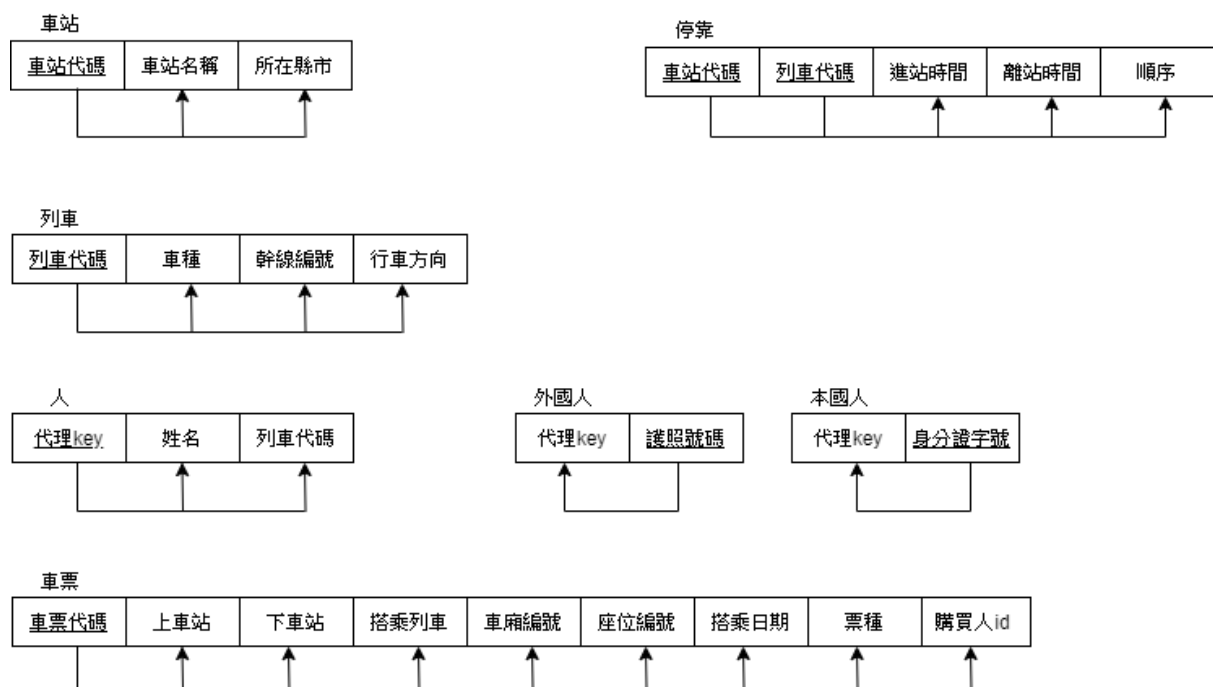
六. ER-diagram



七. Relational Schema



3NF Schema



八. Database Table

1) Table 欄位

a. Station (車站)

欄位名稱	中文名稱	資料型態	長度	NULL	備註
sid	車站代碼	char	4	No	PK
sname	車站名稱	varchar	4	No	
cty	所在縣市	char	3	No	

b. Train (列車)

欄位名稱	中文名稱	資料型態	長度	NULL	備註
tid	列車代碼	varchar	4	No	PK
kind	車種	tinyint		No	
line_on	行駛幹線	tinyint		No	
dir	行駛方向	tinyint		No	

c. stop_at (停靠)

欄位名稱	中文名稱	資料型態	長度	NULL	備註
sid	車站代碼	char	4	No	PK, FK
tid	列車代碼	varchar	4	No	PK, FK
torder	停靠順序	tinyint		No	
arrrtime	進站時間	time		No	
deptime	離站時間	time		No	

d. person (人)

欄位名稱	中文名稱	資料型態	長度	NULL	備註
pkind	代理 key	tinyint		No	PK
pname	姓名	varchar	15	No	
tid	列車代碼	varchar	4	No	FK

e. foreigner(外國人)

欄位名稱	中文名稱	資料型態	長度	NULL	備註
pkind	代理 key	tinyint		No	FK
psp_no	護照號碼	char	9	No	PK

f. native (本國人)

欄位名稱	中文名稱	資料型態	長度	NULL	備註
pkind	代理 key	tinyint		No	FK
ssn	身分證字號	char	10	No	PK

g. ticket (車票)

欄位名稱	中文名稱	資料型態	長度	NULL	備註
tid	車票代碼	char	15	No	PK
geton	上車站	char	4	No	FK
getoff	下車站	char	4	No	FK
ttrain	搭乘列車	varchar	4	No	FK
cno	車廂編號	tinyint		No	
sno	座位編號	tinyint		No	
tdate	搭乘日期	date		No	
ttype	票種	tinyint		No	
pkind	購買人代理 key	tinyint		No	FK

2) Tuple 範例

a. station (車站): 241 tuples

sid	sname	cty
0900	基隆	基隆市
0920	八堵	基隆市
0930	七堵	基隆市

b. train (列車): 174 tuples

tid	kind	line_on	dir
1	1	1	2
2	1	1	1
602	1	0	1

c. stop_at(停靠): 3340 tuples

sid	tid	torder	arrtime	deptime
1000	1	1	06:10:00	06:10:00
1020	1	2	06:18:30	06:20:00
1080	1	3	06:40:00	06:42:00

d. person(人): 0 tuple (等待使用者輸入，以下範例 3 tuples)

pkind	pname	tid
1	weizhe	1
2	chingwen	2
3	yuhan	602

e. foreigner(外國人): 0 tuple (等待使用者輸入，以下範例 3 tuples)

pkind	psp_no
1	123456789
2	145236987
3	541278963

f. native(本國人): 0 tuple (等待使用者輸入，以下範例 3 tuples)

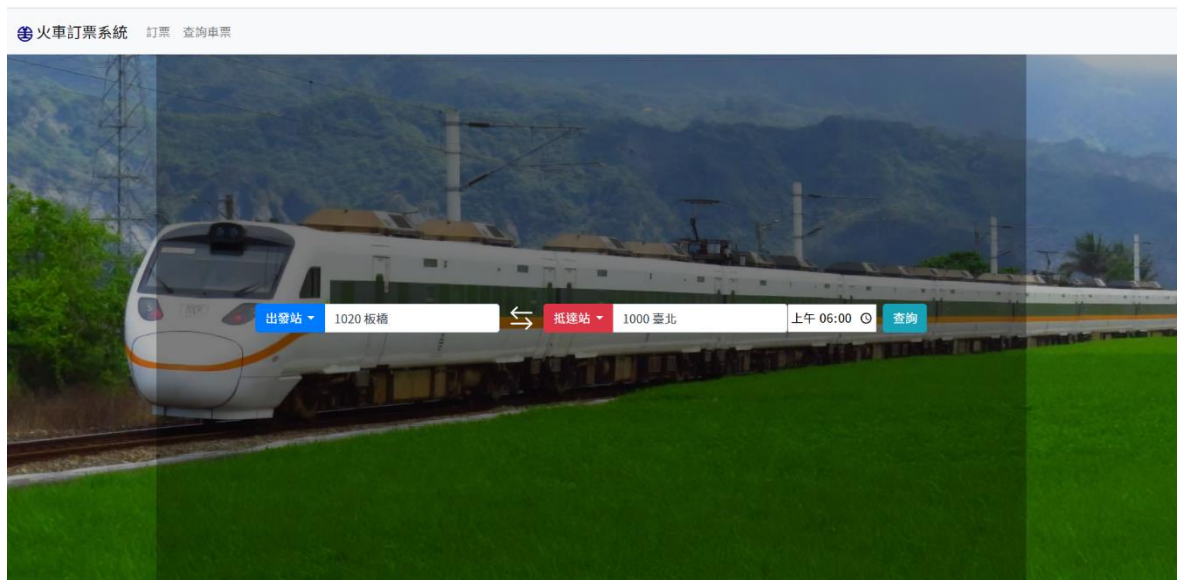
pkind	ssn
4	E123456789
5	T223557651
6	D124798154

g. ticket(車票) 0 tuple (等待使用者輸入，以下範例 3 tuples)

tid	geton	getoff	ttrain	cno	sno	tdate	ttype	pkind
1	0900	0920	129	4	12	2020.03.02	1	1
2	3300	4400	101	7	60	2020.06.06	1	2
3	0960	0980	117	2	4	2020.07.15	2	3

九. 實作網頁

(1)進入訂票首頁



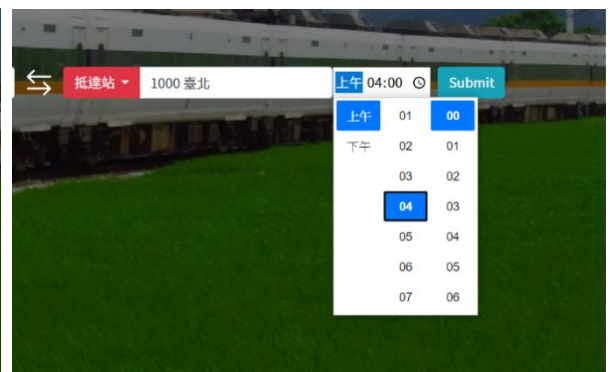
(2)左上可選訂票或查詢



(3)選擇訂票功能



(4)選擇出發站、抵達站、時間，並按送出



```
stations = Station.objects.all()
```

(5)查詢結果

火車訂票系統 訂票 查詢車票						
出發站 ▾ 1020 板橋		⇌	抵達站 ▾ 1000 臺北		上午 06:00 ⌚	查詢
車種車次	出發時間	抵達時間	行駛時間	經由	票價	訂票
自強 402	06:02	06:12	00:10	-	\$20	🔗
自強 204	06:06	06:20	00:14	-	\$28	🔗
自強 406	06:27	06:37	00:10	-	\$20	🔗
莒光 74	06:49	07:00	00:11	-	\$22	🔗
莒光 82	07:01	07:12	00:11	-	\$22	🔗
自強 272	07:09	07:23	00:14	山線	\$28	🔗
自強 408	07:23	07:35	00:12	-	\$24	🔗
自強 410	07:40	07:52	00:12	-	\$24	🔗
自強 102	08:05	08:16	00:11	山線	\$22	🔗

```

cursor.execute('''
    select st1.tid, st1.arrrtime, st2.arrrtime, t.kind, t.line_no
    from stop_at as st1, stop_at as st2, train as t
    where st1.deptime > '%s' and st1.sid = '%s' and st2.sid = '%s'
        and st1.tid = st2.tid and st1.torder < st2.torder and t.tid = st1.tid
    order by st1.arrrtime;
''') % (time, begin_station_id, dest_station_id))

trains = cursor.fetchall() # return tuple instead of Queryset

```

(6)找到所需班次，點選車次詳細資料

車種車次	出發時間	抵達時間	行駛時間	經由	票價	訂票
自強 402	06:02	06:12	00:10	-	\$20	

火車訂票系統 訂票 查詢車票

出發站 ▼ 1020 板橋

車種車次	出發時間
自強 402	06:02
自強 204	06:06
自強 406	06:27
莒光 74	06:49
莒光 82	07:01
自強 272	07:09
自強 408	07:23
自強 410	07:40
自強 102	08:05
自強 412	08:35
自強 212	08:46

自強 402

車站	到站時間	離站時間
樹林	05:51:00	05:55:00
板橋	06:02:00	06:04:00
臺北	06:12:00	06:14:00
松山	06:20:00	06:21:00
宜蘭	07:20:00	07:21:00
羅東	07:27:00	07:29:00
花蓮	08:19:00	08:22:00
瑞穗	08:57:30	08:59:00
玉里	09:12:00	09:13:00
池上	09:28:00	09:29:00
關山	09:38:00	09:39:00
臺東	10:00:00	10:02:00

上午 06:00 查詢

票價	訂票
\$20	
\$28	
\$20	
\$22	
\$22	
\$28	
\$24	
\$24	
\$22	
\$20	
\$22	

```

cursor.execute('''
    select s.sname, st.arrrtime, st.deptime
    from stop_at as st, train as t, station as s
    where st.sid=s.sid and st.tid=t.tid and t.tid = '%s'
    order by st.torder;
''') % tid)

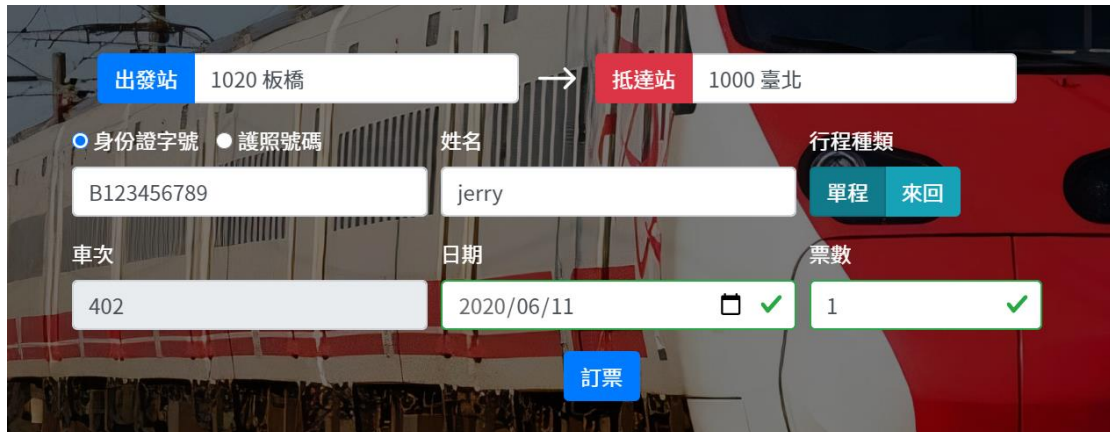
train_schedule = cursor.fetchall()

```

(7)找到所需班次，點選訂票

車種車次	出發時間	抵達時間	行駛時間	經由	票價	訂票
自強 402	06:02	06:12	00:10	-	\$20	

(8)輸入個人資訊，並按訂票



出發站 1020 板橋 → 抵達站 1000 臺北

☐ 身份證字號 ☐ 護照號碼 姓名 行程種類

B123456789 jerry 單程 來回

車次 日期 票數

402 2020/06/11 1

訂票

(8.1)判斷訂票人是否之前已訂過票，若訂過票的話會存在 native 或 foreigner table 裡

```
cursor.execute('''
    select count(*)
    from native
    where ssn='%s';
''') % ssn
result = cursor.fetchall() #確認輸入者是否存在本地人Table裡
```

```
cursor.execute('''
    select count(*)
    from foreigner
    where psp_no='%s';
''') % ssn
result = cursor.fetchall() #確認輸入者是否存在外國人Table裡
```

(8.2)取出目前最大的 person 代理 key，用於 insert 訂票人資訊的事前準備

```
cursor.execute('''
    select MAX(pkind)
    from person;
''')

result = cursor.fetchall()#找尋'person代理key'最大值
```

(8.3)若訂票人之前未訂過票，將訂票人資訊 insert into person table，並根據其輸入身分證字號或護照號碼，將其資訊 insert into 對應 table

(native/foreigner)

```
if surrogate_key is None: #如果'person代理key'為空，將'person代理key'insert值為0
    cursor.execute('''
        insert into person values (%d, '%s', '%s');
    ''' % (0, name, train_id))
```

```
else: #如果'person代理key'不為空，將'person代理key'insert當前最大值加1
    surrogate_key += 1
    cursor.execute('''
        insert into person values (%d, '%s', '%s');
    ''' % (surrogate_key, name, train_id))
```

```
if 'ssn' in ssn_type: #如果輸入者為本地人，將其相關資料insert進native table
    cursor.execute('''
        insert into native values (%d, '%s');
    ''' % (surrogate_key, ssn_value))
```

```
else: #如果輸入者為外國人，將其相關資料insert進foreigner table
    cursor.execute('''
        insert into foreigner values (%d, '%s');
    ''' % (surrogate_key, ssn_value))
return surrogate_key
```

(8.4)若訂票人之前已經訂過票，將其資訊從 native 或 foreigner table 抓取，用於等一下 insert ticket 用

```
cursor.execute('''
    select pkind
    from native
    where ssn='%s';
'' % ssn_value)
result = cursor.fetchall() #如果輸入者已存在native table，抓取其代理key
```

```
cursor.execute('''
    select pkind
    from foreigner
    where psp_no='%s';
'' % ssn_value)
result = cursor.fetchall() #如果輸入者已存在foreigner table，抓取其代理key
```

(8.5)隨機產生 ticket id 並檢查是否已存在 database(ticket id 不可重複)，用於之後 insert ticket

```
while True:
    ticket_id = ''.join(random.choice(string.digits) for x in range(15))
    cursor.execute('''
        select count(*)
        from ticket
        where tid='%s';
    '' % ticket_id)

    result = cursor.fetchall()
    if result[0][0] == 0:
        break
```


(8.6)隨機產生車廂號碼和座位號碼並檢查是否已存在 database(car_no seat_no 不可重複)・用於之後 insert ticket

```
cursor.execute('''
    select count(*)
    from ticket
    where cno = %d and sno = %d;
''') % (car_no, seat_no))
result = cursor.fetchall() #判斷隨機產生的'car_no'和'seat_no'是否已存在database
```

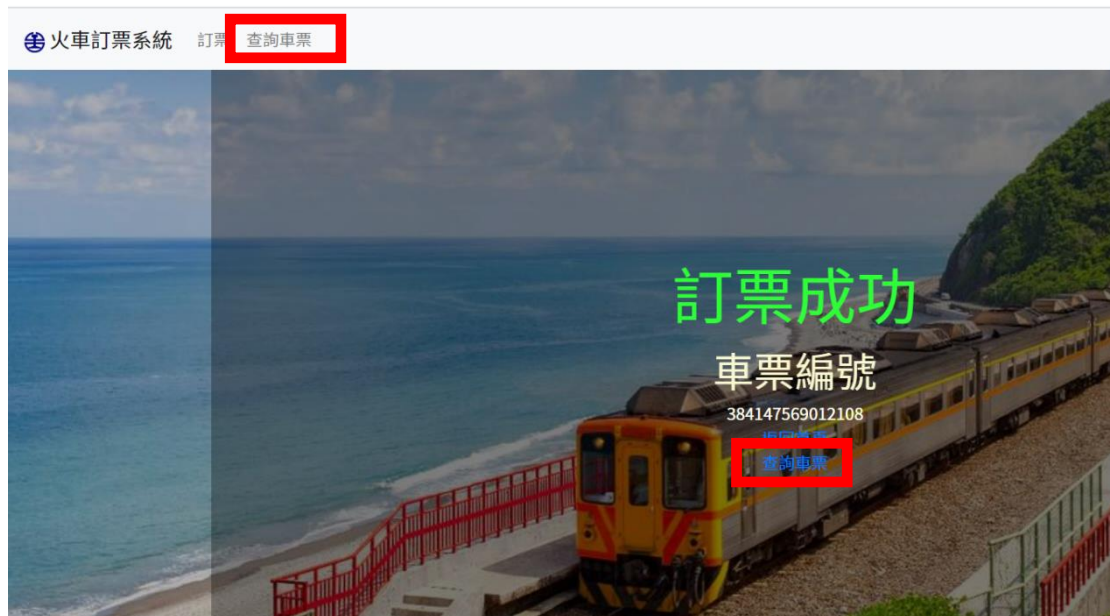
(8.7)成功將車票相關資訊 insert into ticket table

```
cursor.execute('''
    insert into ticket values ('%s','%s','%s','%s',%d,%d,'%s','%s',%d);
''') % (ticket_id, bsid, dsid, train_id, car_no, seat_no, date, ticket_type, surrogate_key))
# 將票的相關資訊insert into ticket table
```

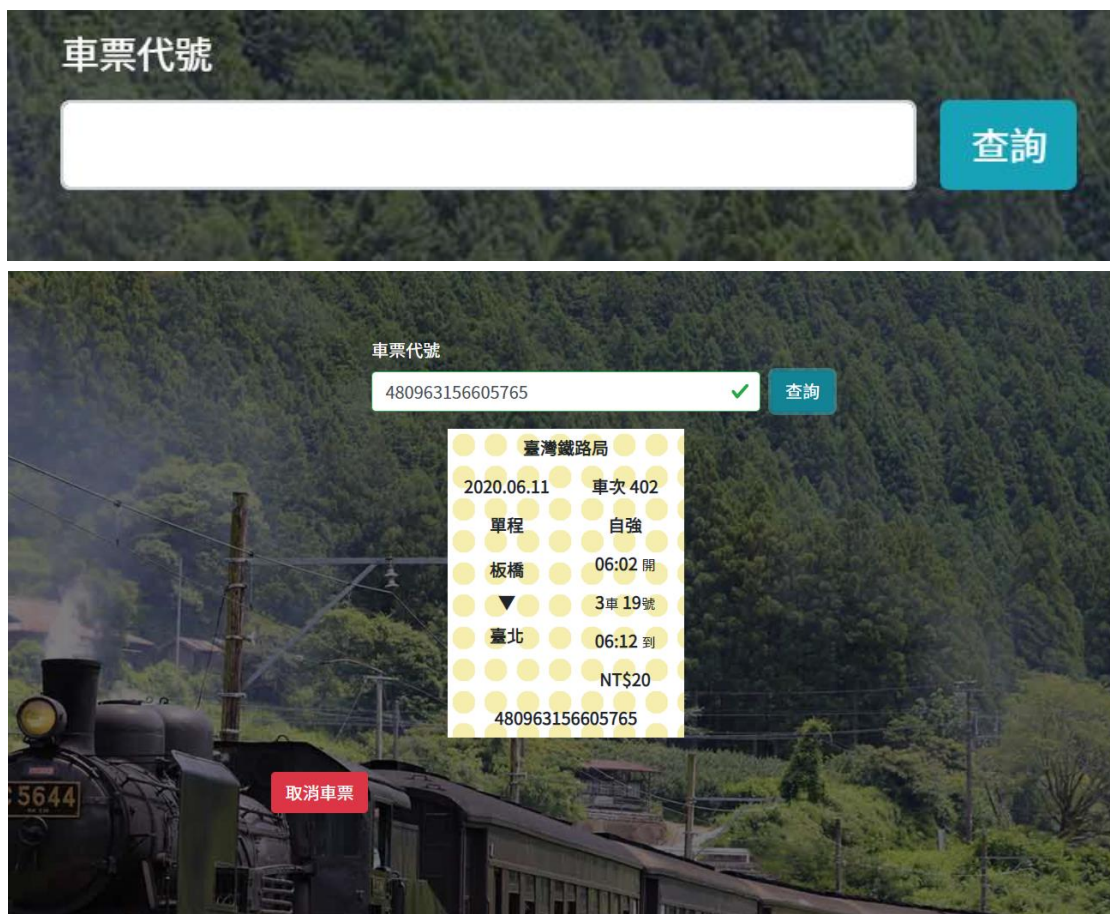
(9)顯示成功頁面



(10)查詢車票



(11)輸入訂票編號



(12)查詢成功

```
ticket_info = Ticket.objects.get(tid=tid)

get_on_station = Station.objects.get(sid=get_on_station_id).sname
get_off_station = Station.objects.get(sid=get_off_station_id).sname

get_on_time = StopAt.objects.get(tid=take_train_id, sid=get_on_station_id).arrrtime
get_off_time = StopAt.objects.get(tid=take_train_id, sid=get_off_station_id).arrrtime

train_kind = Train.objects.get(tid=take_train_id).kind
```

(12)取消車票

車票代號

480963156605765

查詢

臺灣鐵路局

2020.06.11 車次 402

單程 自強

板橋 06:02 開

▼ 3車 19號

臺北 06:12 到

NT\$20

480963156605765

●身份證字號 ●護照號碼

取消車票

輸入訂票人的身份證/護照號碼

提交

(12.1)透過 ticket id 和 person 代理 key 查詢是否有符合的 ticket

```
cursor.execute('''
    select *
    from ticket
    where tid = '%s' and pkind = '%s'
    ''' % (tid, pkind))
result = cursor.fetchall() # 透過'ticket id'和'person 代理key'判斷ticket是否存在
```


(12.2)若有符合的 ticket 將其從資料庫刪除

```
cursor.execute('''
    delete from ticket
    where tid = '%s' and pkind = '%s'
    ''' % (tid, pkind)) # 依據'ticket id'和'person 代理key'刪除ticket
```

(13)取消車票失敗(身分證或護照不符)

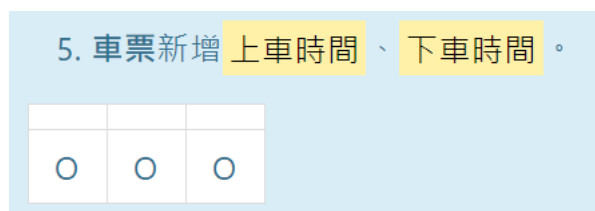


(13)取消車票成功

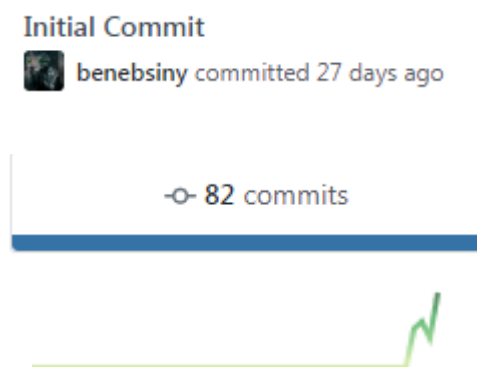


十. 心得

鄭志軒: 製作這次的專題頗刺激，因期中才開始製作專題，所以必須在一個月內把弄它出來，還好以前有學過一些 Web 應用程式，因此才可以在這次的專題上用到。但中途遇到的問題是，隊友他程式不足敗事有瑜，所以還要花不少時間跟他說這個模板的運作方式以及前端與後端的溝通管道。再來就是在製作應用程式的過程中，發現有些 column 在這次的專題中用不到，或者是某些屬性應該要加進 table 裡，諸如此類的一改就是個大工程，因為要改 ER model、Relation Model、還有資料庫的 schema，然後還要通知其他組員說這個 schema 有改到.....，這一連串做下來很累人的，還好我們有想到要使用 HackMD，將其當成公告版，在上面公告要改那些欄位，再交由其他組員來改就好了。



在最後終於把專題做出來了，不僅加深了這個模板的理解，更了解資料庫的應用，以後設計其他的應用程式時，就能搭配資料庫，做出個有聲有色的應用程式。最後附上的我們的辛酸血淚圖。



林則穎: 從開始設計 ER Model 時，遇到對於 disjoint 和 union 的應用感到疑惑，接著開始實作專題，在隊友有使用過 django 框架的經驗下，讓我在剛開始實作時，得到很大幫助，加速理解如何操作 django 與資料庫的溝通，還有與前端的連接，過程遇到一些程式語法的 bug，也漸漸學會試著自己去 google，雖

然還是會遇到花了很長的時間仍然找不到錯誤與解決方案，這時的自己總是格外的沮喪、失落，於是我會先放下手邊的工作沉澱自己，再重新出發去尋找答案。隨著程式的推演，發現一開始在設計 ER Model 和 Relation Model 的時候有考量不周的情況，這時要從 Database 到 Python 再到 JavaScript 一路往上的修正，真的是浩大工程。還有這次專題運用 GitHub 進行版本控制，雖然上學期已有接觸，但對指令的熟悉度還是不夠，有時候在處理衝突時不夠謹慎，導致隊友的程式碼無故被刪掉，或著總是累積一段時間才 commit，造成隊友的困擾，感到非常抱歉...

這次專題的完成，必須感謝隊友不厭其煩的幫忙、協助與包容，才能順利完成專題，自己也在過程中學習到很多程式的新知，是個收穫豐富的經驗。

林聖閔: 這次資料庫專題的製作，我們選擇以台鐵訂票為主題，由於生活中很常使用到，所以特別熟悉。但開始進入實作階段，種種困難像潑了我一頭冷水。最一開始製作架構時由於課堂理論進度沒有跟上，都只能依靠組員的能力完成。到期中考之後，對於這些 Relation Model、ER Model 才有更深的了解，才開始慢慢摸索並協助將細節補足。組員們都很厲害，不論是 SQL 的框架或是網頁的撰寫，都令人佩服尤其網頁美化的部分都比台鐵的訂票系統做得更有質感，程式只要我不懂地方的都會耐心的和我解說，從他們身上學習到了很多做事態度和設計程式的能力。這份專題憑一己之力，真的不可能完成，真的十分感謝組員們都拚命擠出時間，討論並且完成這個作業。

陳宣燁: 第一做這種規模的專題，我認為最困難的部分是學習新的程式語言，因為我沒有參與到初期的討論，所以這次專題所用的語言都是我第一次接觸的，剛開始寫的時候，連編輯器的安裝都有些問題，時不時還跑出一些問題，甚至編輯器崩潰，這些都只是一開始遇到的問題，而我因為第一次接觸 SQL、Python、PHP、HTML，基本上是會用到的我都不會，連一些問題都不知道如何去搜尋，時常需要去問對這些語言熟悉的朋友，或是泡在 Google 上，尋找自己遇到的問

題和程式的寫法，甚至遇到明明 Error 一樣，但解決方法不同的困境，這讓我覺得 2 個螢幕都不夠用，因為遇到的問題太多都是我不熟悉，瀏覽器裡時常是 20 幾個分頁常駐，在這麼艱苦的條件下還能完成，真的幸虧有一群很罩的隊友和半夜被我吵起來的朋友，沒有他們，我不可能完成這個專題，超感謝他們的！