國立清華大學資訊工程系 111 學年度上學期專題報告

專題名稱	第三方支付平台開發		
參加競賽或計畫	☐ 參加對外競賽	☐ 參與其他計畫	☑無參加對外競賽或任何計畫
學號	108062227		
姓名	鄭守維		

摘要

本次專題研究目標為與學校計通中心合作，開發一個第三方支付平台，並在開發完成後和學校之校務資訊系統對接供師生實際使用，並與行動支付業者合作，擔任學校系統和行動支付業者之間的中介平台，提供師生向學校繳納各項費用時額外的線上支付管道。

專題研究動機與目的

隨著行動支付的日益普及，日常生活中的許多消費都已經能夠一鍵快速完成交易，但學校校園內許多費用的繳納都還是需要經過繁瑣的程序，因此本專題希望藉由開發適當的平台與學校各系所處室進行整合銜接，帶領行動支付走入校園，提供師生更快速便利的支付工具。

專題開發之平台特點

在沒有此專題開發之第三方支付平台之前，如果學校的任一行政單位想要提供更多元的費用支付管道服務，則其必須**自行**完成以下業務:

- 與每一家行動支付或銀行業者接洽簽約
- 每一個行動支付管道都必須做一次 API 的串接
- 管理資料庫中總訂單紀錄與各個支付管道訂單紀錄的關聯性

以上業務耗時又費力，因此造成在校園行政單位內推動增加行動支付管道窒礙難行，為解決以上問題，本專題開發之平台能夠提供以下服務

- 平台內已經事先和行動支付業者完成簽約，並完成 API 的串接
- 平台內能夠幫忙完成商家和各行動支付管道的訂單資料庫關聯性管理，因此商家只要管理一個和付款人對應的訂單資料庫即可

以上服務，商家只要做一次和第三方支付平台的 API 串接，即可一次完成增加多個行動支付管道的服務，大大消除了以往開發多元支付管道的阻力，相信能因此進一步推動行動支付進入校園。

現有相關系統概況及比較

目前在台灣流通常見的第三方支付業者有綠界科技、紅陽科技還有藍新科技，本專題參考綠界的文件較多因此與綠界來做比較。

綠界金流服務提供了完整的交易串接技術給有收款需求的商店，其中也支援多元的收款方式讓消費者選擇，包括信用卡、網路 ATM、ATM 櫃員機、超商代碼、超商條碼及其他行動支付品牌等，達到多元付款的便利性。但也因為要提供全方位的金流服務，所以在 API 的串接上商家需要填寫的參數數量就非常龐大，包括商家指定付款管道、付款完成後的動作指定或者是有特別合作的店家就需要另外填寫參數等。

相較於綠界金融服務，本平台因為定位是服務學校內的行政單位，付款交易的流程較為單純，所以在 API 串接時可以縮減大量不必要的參數，減少商家(學校行政單位)在串接時的負擔。

平台系統開發實現

第三方支付平台付款交易流程圖

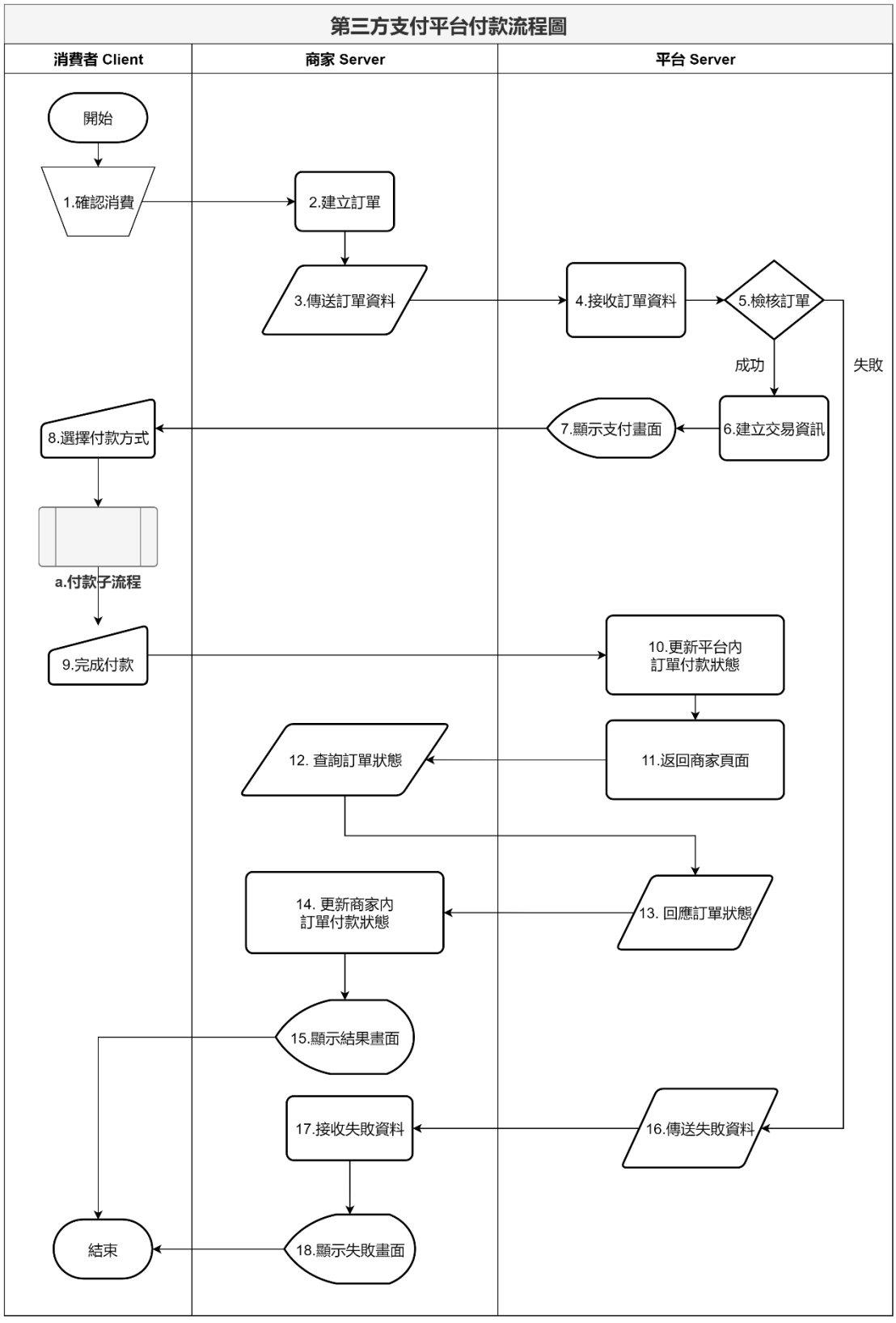

圖 1

表 1

流程名稱	處理角色	時機	說明
1.確認消費	消費者	消費後	消費者向店家發送確認消費請求
2.建立訂單	商家		店家確認消費者請求並建立訂單
3.傳送訂單資料	商家		店家平台提供的 API 將訂單資料傳送至平台系統並開始付款
4.接收訂單資料	平台		
5.檢核訂單	平台		平台接收訂單資料並做解析確認，成功即進入下個流程，失敗則終止並回傳錯誤資訊
6.建立交易資訊	平台		確認無誤後，在平台系統中建立一筆交易訂單資訊
7.顯示付款畫面	平台		依照店家傳入的付款選擇參數，顯示可用付款方式給消費者選擇
8.選擇付款方式	消費者		消費者選擇付款方式
9.完成付款	消費者		消費者依照其選擇付款方式完成付款
10.更新平台內付款狀態	平台		平台依照消費者付款狀況更新平台系統內的付款狀態
11.返回商家頁面	平台		平台確認消費者付款完成後，將消費者轉導回商家頁面
12.查詢訂單付款狀態	商家		進入商家頁面時，商家再次呼叫 API 向平台查詢訂單狀態
13.回應訂單付款狀態	平台		平台接收商家的查詢要求後，回傳訂單付款狀態
14.更新訂單付款狀態	商家		根據平台回傳的內容更新商家自己的訂單付款狀態
15.顯示結果畫面	商家		根據新的付款狀態結果顯示給消費者讓其知道交易已完成
16.傳送失敗資料	平台	「5.檢核訂單」之後	平台檢核店家傳送的訂單如果結果為失敗，則收集錯誤資訊並回傳至店家
17.接收失敗資料	商家		
18.顯示失敗資	商家		

料			

付款子流程

目前完成串接的為 LinePay PC 版

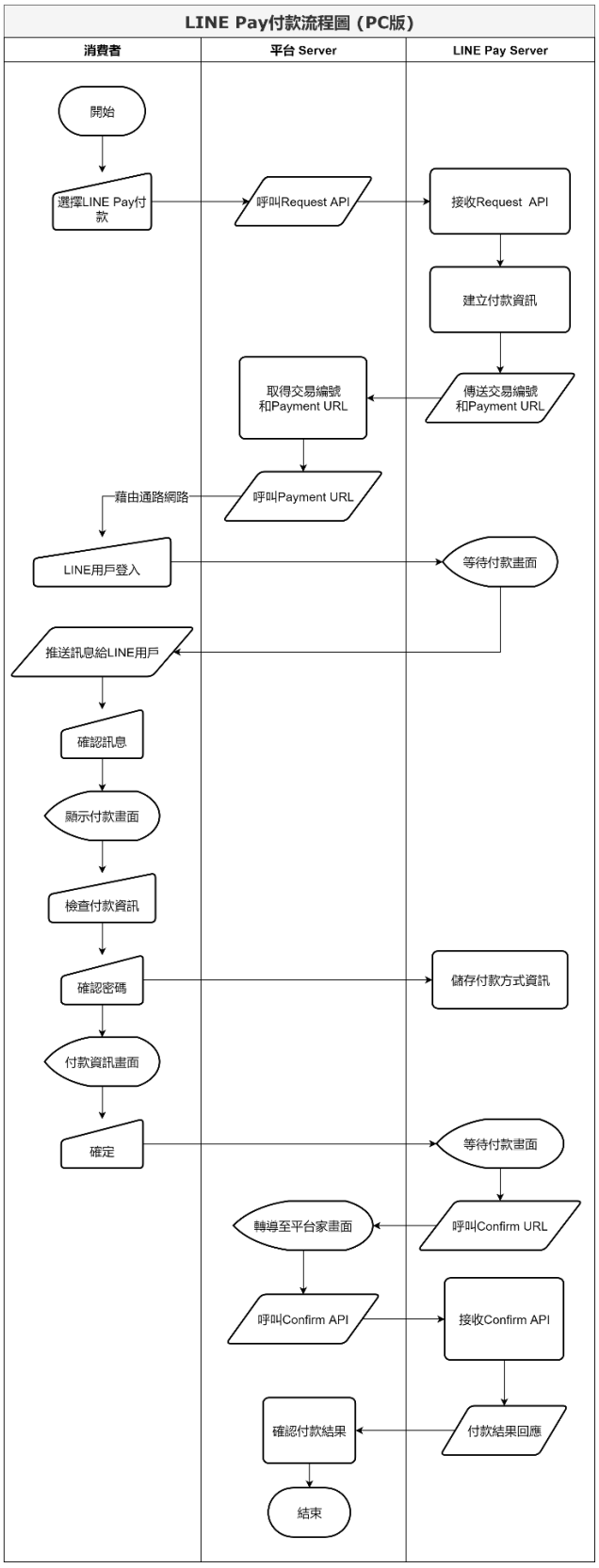

圖 2

付款流程

(1) 消費者選擇以 LINE Pay 作為付款方式
(2) 店家向 LINE Pay Server 呼叫 LINE Pay 的 Request API
(3) LINE Pay Server 建立付款資訊並將交易編號和 paymentURL 回傳給店家
(4) 店家呼叫步驟(3)接收到的 paymentURL
(5) 消費者經由網頁登入確認為 LINE 用戶
(6) LINE Pay Server 推送付款請求至消費者的 LINE 應用程式
(7) 消費者確認付款請求後畫面導向至付款畫面
(8) 消費者在 LINE Pay 付款畫面中選擇付款方式並輸入驗證密碼
(9) LINE Pay Server 儲存付款方式，並將付款狀態變更為已授權
(10) 消費者畫面導向至付款資訊畫面，消費者檢查並按下確定後，導向至平台所提供的 confirmURL
(11) 平台呼叫 Confirm API，待 LINE Pay Server 回傳資料後確認完成付款

產生訂單

表 2

參數	參數名稱	型態	說明	範例
MerchantID	商家編號 (平台提供)	String(10)		00001
MerchantTradeNum	訂單交易編號 (此為商家與消費者共用之交易編號)	String(20)	須為唯一值，不可重複 (為避免重複商家應先自行加密，但目前暫時忽略)	cs00001
MerchantTradeDate	商家交易時間	String(20)	格式: YYYYMMDDhhmmss	20220325153030
TotalAmount	交易金額	Int	須為整數不可有小數點 僅限新台幣	20000
TradeDesc	交易描述	String(200)	不得帶入特殊字元	註冊繳納學雜費
ItemName	商品名稱	String(400)	1. 如果有多筆商品要顯示，商品名稱以符號#分隔 2. 自述長度限制為中英數 400 字內，超過自動截斷	學費 10000 元#雜費 10000 元
ExpireDate	繳費期限	String(20)	格式: YYYYMMDDhhmmss	20220325120000
Remark	備註欄位	String(200)		

平台於資料庫中記錄之訂單資訊

表 3

名稱	型別	範例	對應之店家傳入參數
TradeRecordID	bigserial primary key	1	無
MerchantID	varchar(10) not null	00001	MerchantID
MerchantTradeNum	varchar(20) primary key	cs00001	MerchantTradeNum
MerchantTradeDate	varchar(20) not null	20220325153030	MerchantTradeDate
TotalAmount	int not null	20000	TotalAmount
ItemName	varchar(400) not null	學費 10000 元/雜費 10000 元	ItemName
TradeDesc	varchar(200)	註冊繳納學雜費	TradeDesc
Remark	varchar(200)		Remark
ExpireDate	varchar(20) not null	20220720000000	ExpireDate
Deletion	boolean	true	無
RandomString	varchar(20)	ffjrn3k1lv8dj4u3	無
CSRFToken	varchar(20)	lkv8vjd7d782l9ek	無
PaymentCompletion	boolean	true	無

參數說明:

- Deletion : 標記該筆訂單是否已被刪除
- RandomString : 儲存用於防範 Replay Attack 的隨機字串
- CSRFToken : 儲存用於防範 CSRF 的隨機字串
- PaymentCompletion : 標記該筆訂單是否已完成付款

平台 API 接收處理流程圖

Main function

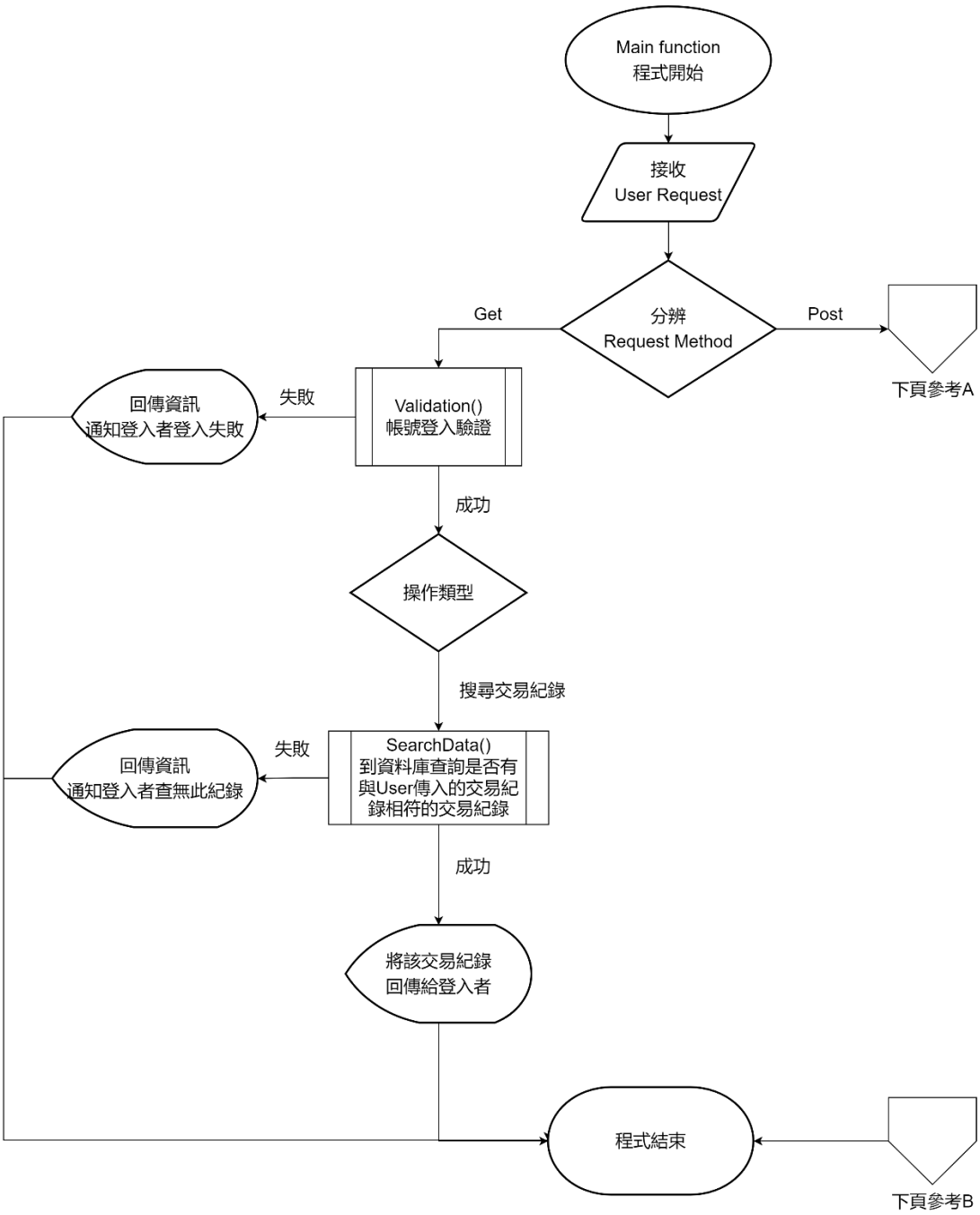

圖 3

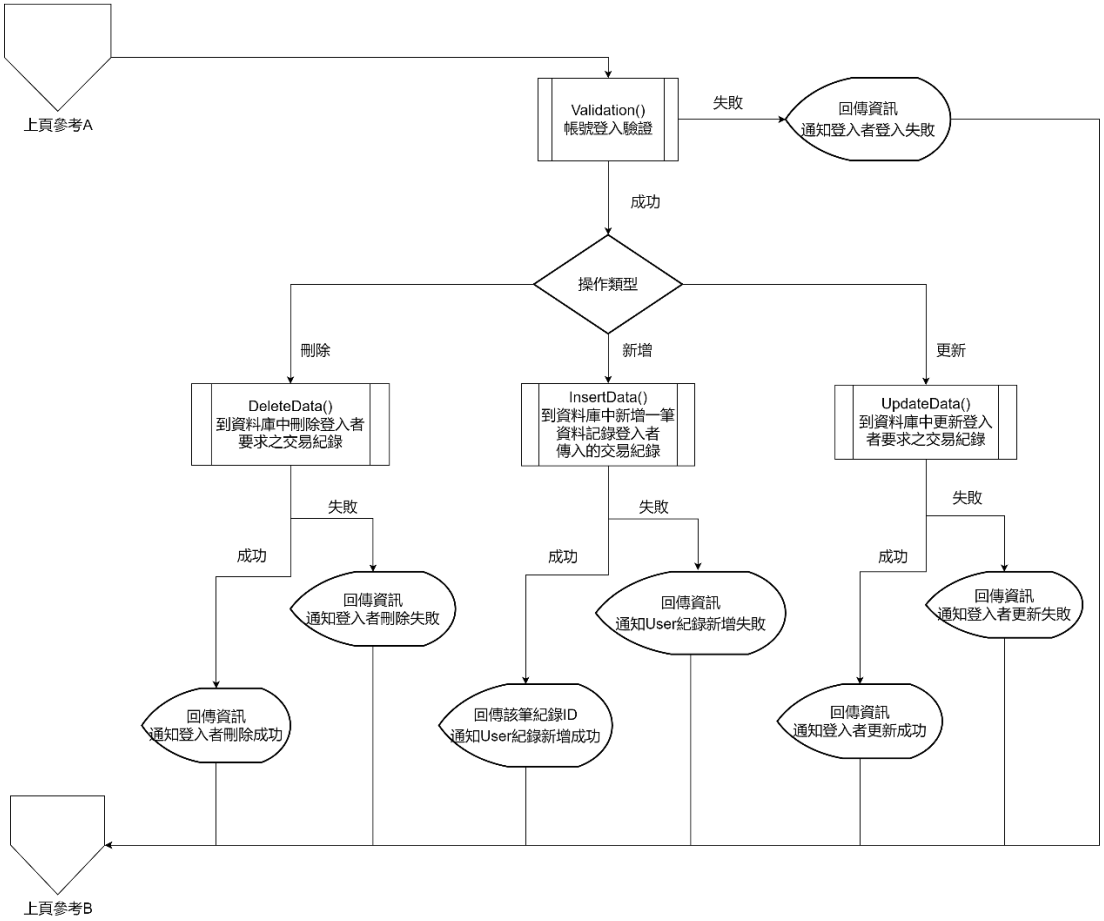

圖 4

圖 3 & 圖 4 說明:

API 的處理程序會依照不同 Request 使用以下 5 種函式

1. Validation( ) : 負責驗證呼叫平台 API 傳送要求的用戶是否是已經事先向平台註冊過的合法商家
2. SearchData( ) : 商家欲查詢某筆訂單記錄時，使用此函式查詢到平台資料庫提取該筆紀錄
3. InsertData( ) : 商家欲新增一筆訂單紀錄時，使用此函式到平台資料庫中新增紀錄
4. DeleteData( ) : 商家欲刪除一筆訂單紀錄時，使用此函式到平台資料庫中將該紀錄標記為已刪除
5. UpdateData( ) : 商家欲更新一筆訂單紀錄時，使用此函式到平台資料庫中更新該筆紀錄

Validation function

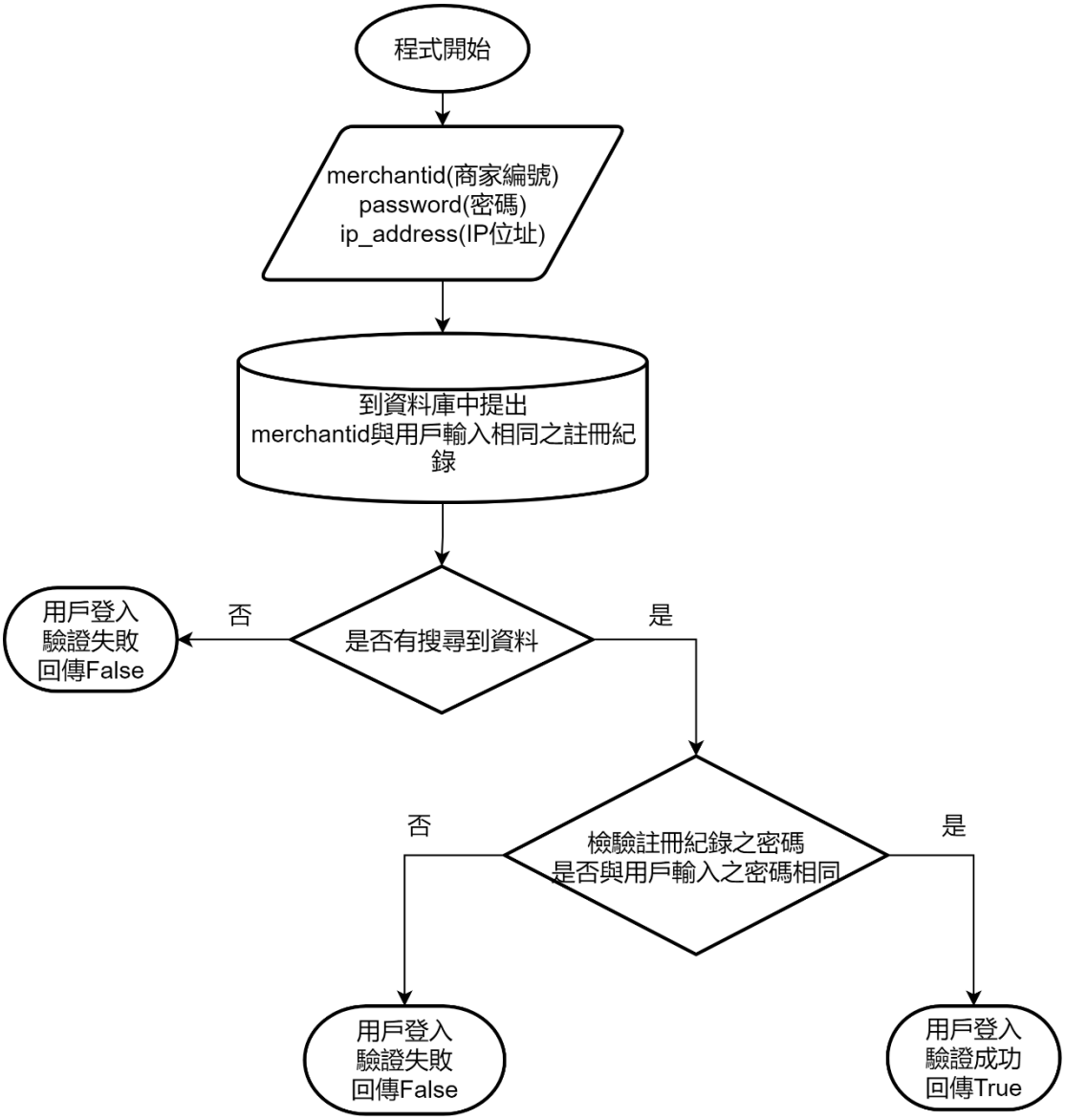

圖 5

圖 5 說明:

在將用戶輸入之密碼與資料庫中註冊之密碼進行比對時，是利用 php 內建的 hash_equals( )做比對，註冊時密碼會先經過 php 內建的 crypt( )雜湊函式再存入資料庫中，而本專題目前是使用 CRYPT_SHA256 這個 hash type 所以做完雜湊前面會帶有一個前綴的 salt 值，因此要做比對時就將該 salt 值對用戶輸入之密碼做雜湊，再和資料庫中記錄之密碼作比對看是否相同。

Validation function 安全性檢核表

Validation() 帳號登入驗證		
安全需求項目	說明	評量結果
與資料庫連線失敗時應採取之應對措施。	與資料庫連線失敗時，於日誌中記錄嘗試此次登入失敗事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者出現錯誤。	符合
進行密碼比對時使用防止時序攻擊(Timing attack)之比較函數。	從資料庫中提出與用戶輸入之帳號相同的註冊紀錄後，以 hash_equals()判斷註冊紀錄中的密碼是否與嘗試登入者輸入的密碼相同。	符合

SearchData function

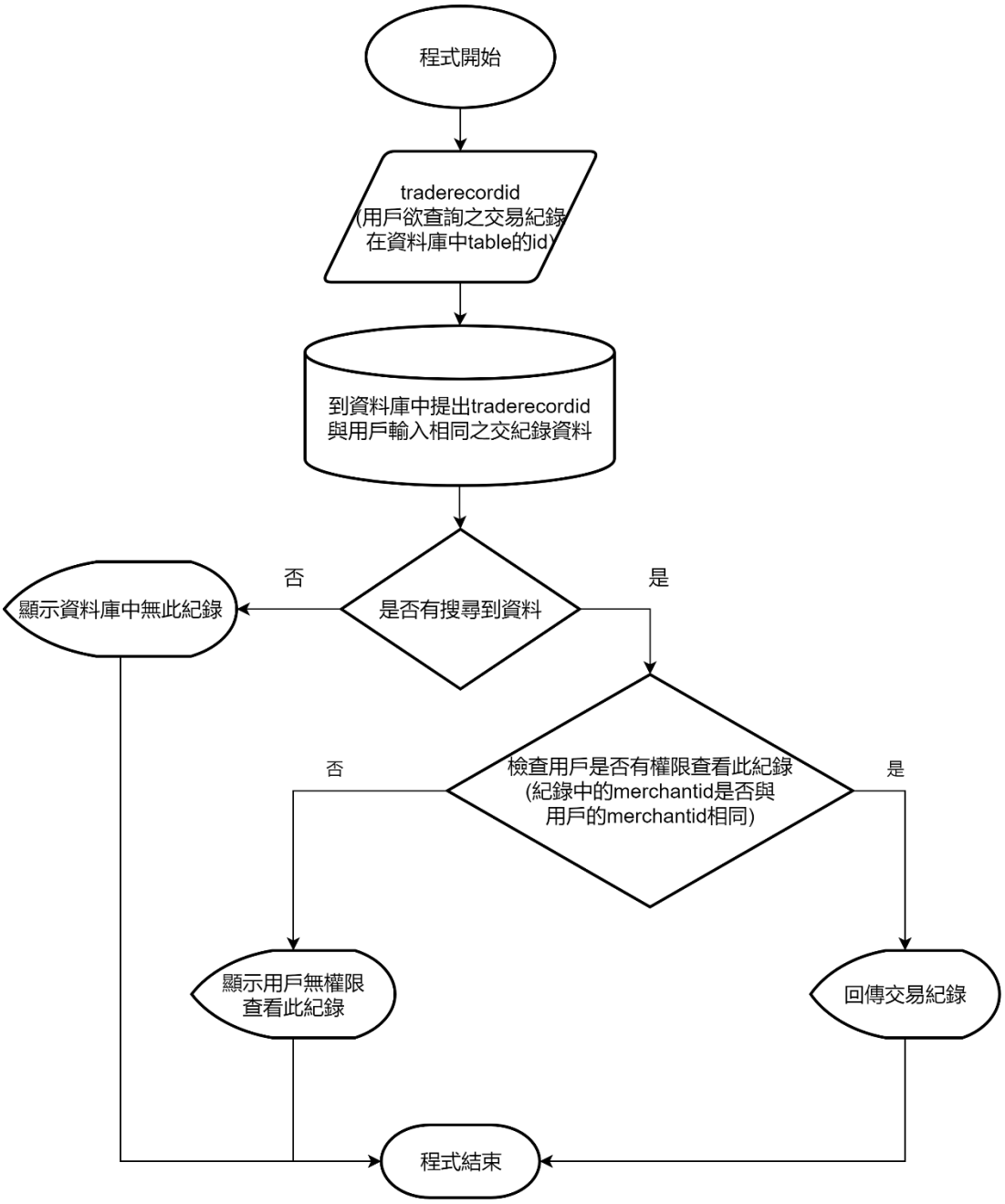

圖 6

SearchData function 安全性檢核表

SearchData() 查詢交易紀錄		
安全需求項目	說明	評量結果
與資料庫連線失敗時應採取之應對措施。	與資料庫連線失敗時，於日誌中記錄嘗試此次連線失敗事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者出現錯誤。	符合
登入者輸入之交易紀錄序號在資料庫中不存在之應對措施。	函式檢查發現查詢資料庫返回的資料集為空時，於日誌中記錄此次事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者查無此交易紀錄	符合
登入者嘗試查詢其他帳號創建的交易紀錄，意即登入者無權限查詢該筆交易紀錄之應對措施。	函式檢查發現該筆交易紀錄中的帳號與登入者之帳號不同時，於日誌中記錄此次事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者其無權限查詢該筆交易紀錄。	符合
登入者要求查詢之該筆交易紀錄已被刪除之應對措施。	函式檢查發現該筆交易紀錄之[Deletion]欄位為 True，意即該筆交易紀錄已被刪除時，於日誌中記錄此次事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者該筆交易紀錄已被刪除。	符合

If_aready_exist function

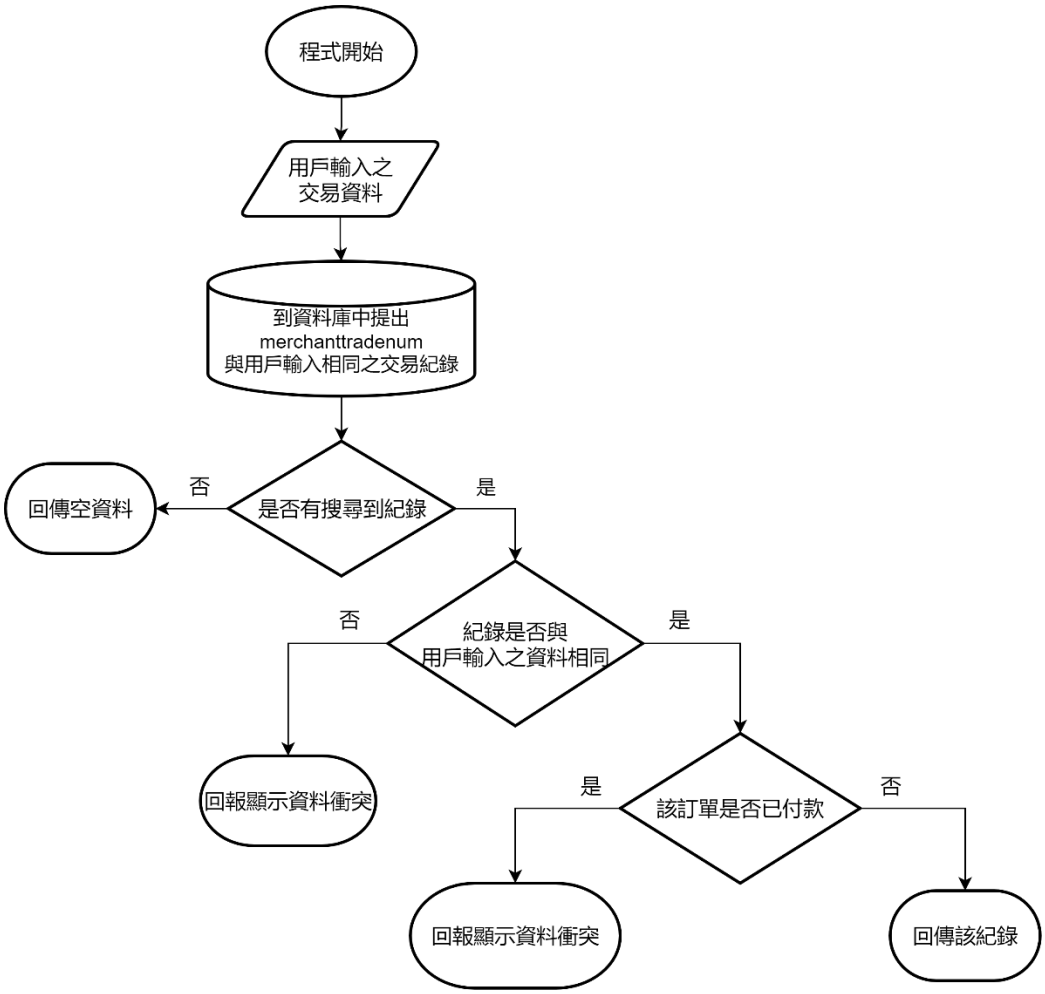

圖 7

圖 7 說明:

此為 InsertData( )執行過程中呼叫的函式，用途為檢察用戶要求新增之訂單紀錄是否已存在於平台資料庫中，也會檢驗以下內容

- 用戶要求新增之訂單紀錄的交易編號 MerchantTradeNum 是否和已存在於資料庫中的紀錄相同造成衝突。
- 用戶要求新增之訂單紀錄是否已完成付款

If_aready_exist function 安全性檢核表

If_already_exist() (新增交易紀錄前)檢查紀錄是否已存在		
安全需求項目	說明	評量結果
與資料庫連線失敗時應採取之應對措施。	與資料庫連線失敗時，於日誌中記錄嘗試此次連線失敗事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者出現錯誤。	符合
登入者輸入之交易資料與存在於資料庫中之交易紀錄不完全符合之應對措施。	函式檢查發現登入者輸入之交易資料之交易編號([MerchantTradeNum]欄位)與資料庫中某筆交易紀錄相同，但其餘欄位不完全相同時，意即這兩筆資料會產生衝突，應否決此次新增紀錄之要求，於日誌中記錄此次事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者該筆交易紀錄使用了重複的交易編號。	符合

InsertData function

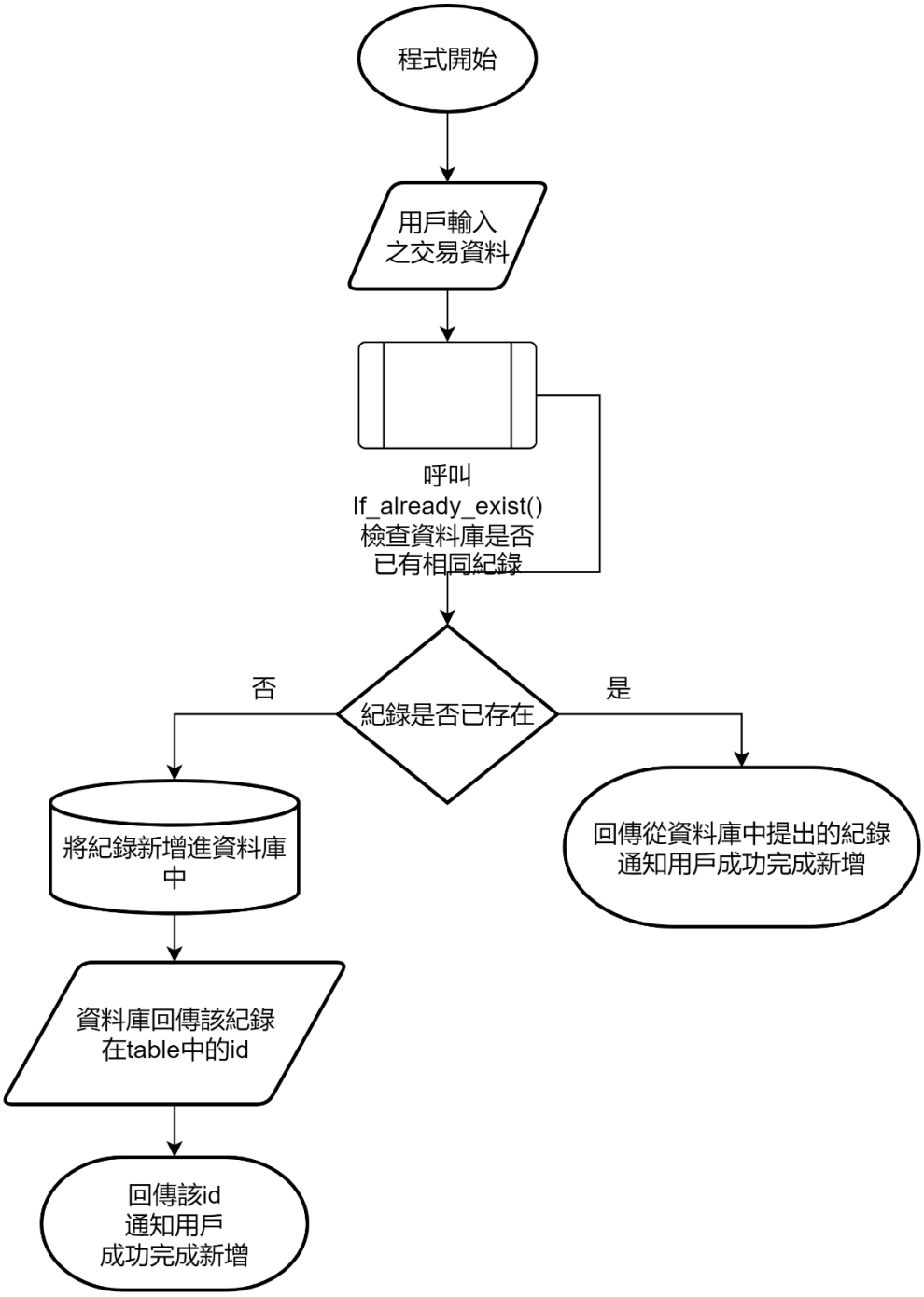

圖 8

圖 8 說明:

此函式有設立一防呆機制，如果用戶要求新增一筆已存在於平台資料庫中的訂單紀錄，則此函式呼叫 If_already_exist( )發現訂單紀錄重複後，不會新增紀錄，而會直接回傳該筆資料庫中紀錄的 TradeRecordID 通知用戶已成功新增。

InsertData function 安全性檢核表

InsertData() 新增交易紀錄		
安全需求項目	說明	評量結果
與資料庫連線失敗時應採取之應對措施。	與資料庫連線失敗時，於日誌中記錄此次連線失敗事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者出現錯誤。	符合
登入者輸入之交易資料已存在於資料庫中之應對措施。	函式呼叫 If_already_exist()檢查登入者輸入之交易資料是否已存在於資料庫中，如結果為是則回傳存在於資料庫中之交易紀錄的序號。	符合

DeleteData function

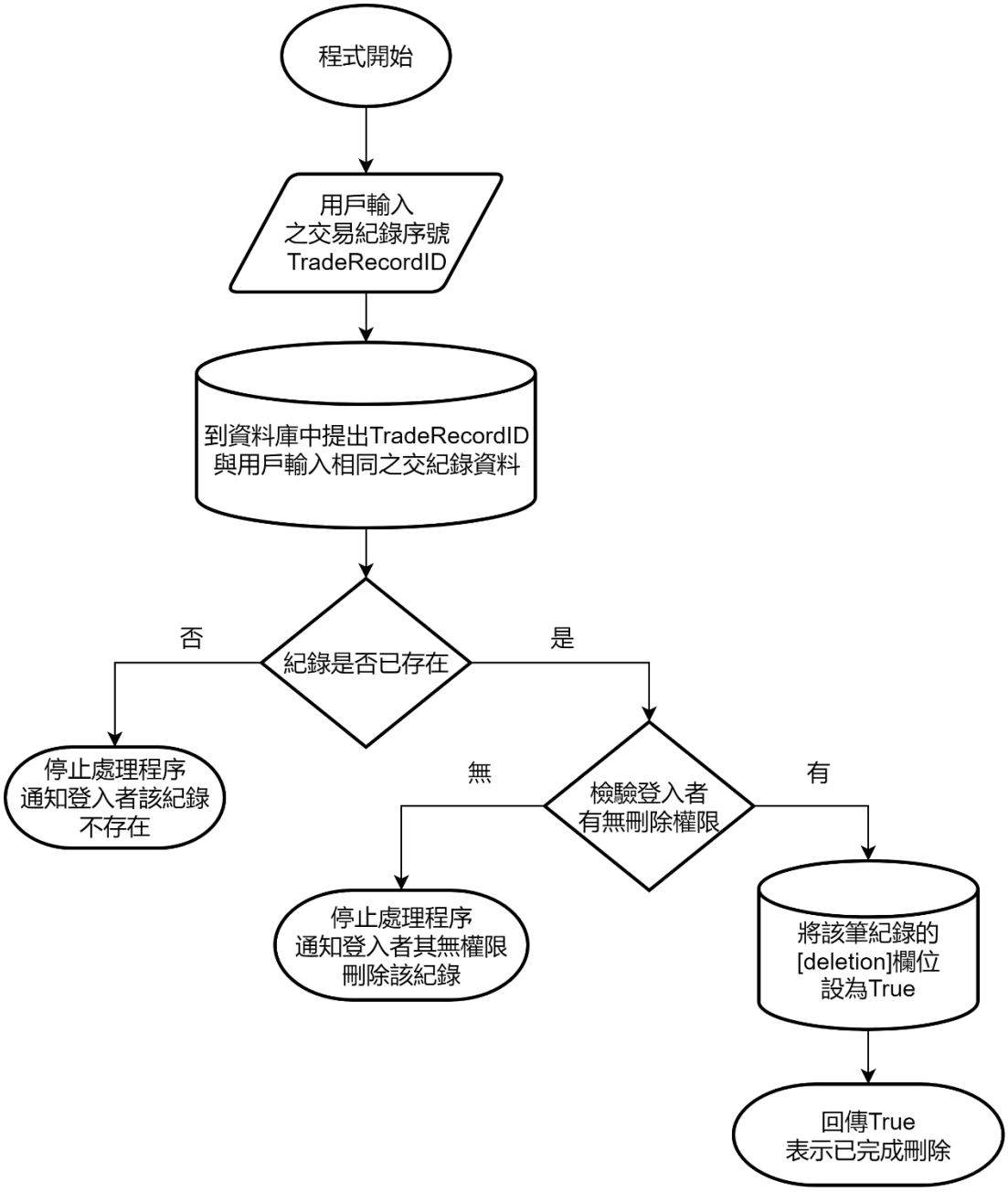

圖 9

圖 9 說明:

因為平台的服務內容涉及金融交易，在做刪除紀錄的動作時並不能真的刪除該紀錄，所以這裡的作法為將該筆紀錄的[Deletion]欄位設為 True 表示已刪除，之後搜尋資料時再檢查該欄位判斷是否需要忽略該紀錄。

DeleteData function 安全性檢核表

DeleteData() 刪除交易紀錄		
安全需求項目	說明	評量結果
與資料庫連線失敗時應採取之應對措施。	與資料庫連線失敗時，於日誌中記錄此次連線失敗事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者出現錯誤。	符合
登入者要求刪除的交易紀錄不存在於資料庫中之應對措施。	函式檢查發現該筆資料不存在於資料庫時，於日誌中記錄此次事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者該筆交易紀錄不存在。	符合
登入者帳號與其要求刪除之交易紀錄中記錄的帳號不同，意即登入者無權限要求刪除該筆交易紀錄之應對措施。	函式檢查發現登入者帳號與其要求刪除之交易紀錄中記錄的帳號不同時，於日誌中記錄此次事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者其無權限刪除該筆交易紀錄。	符合

UpdateData function

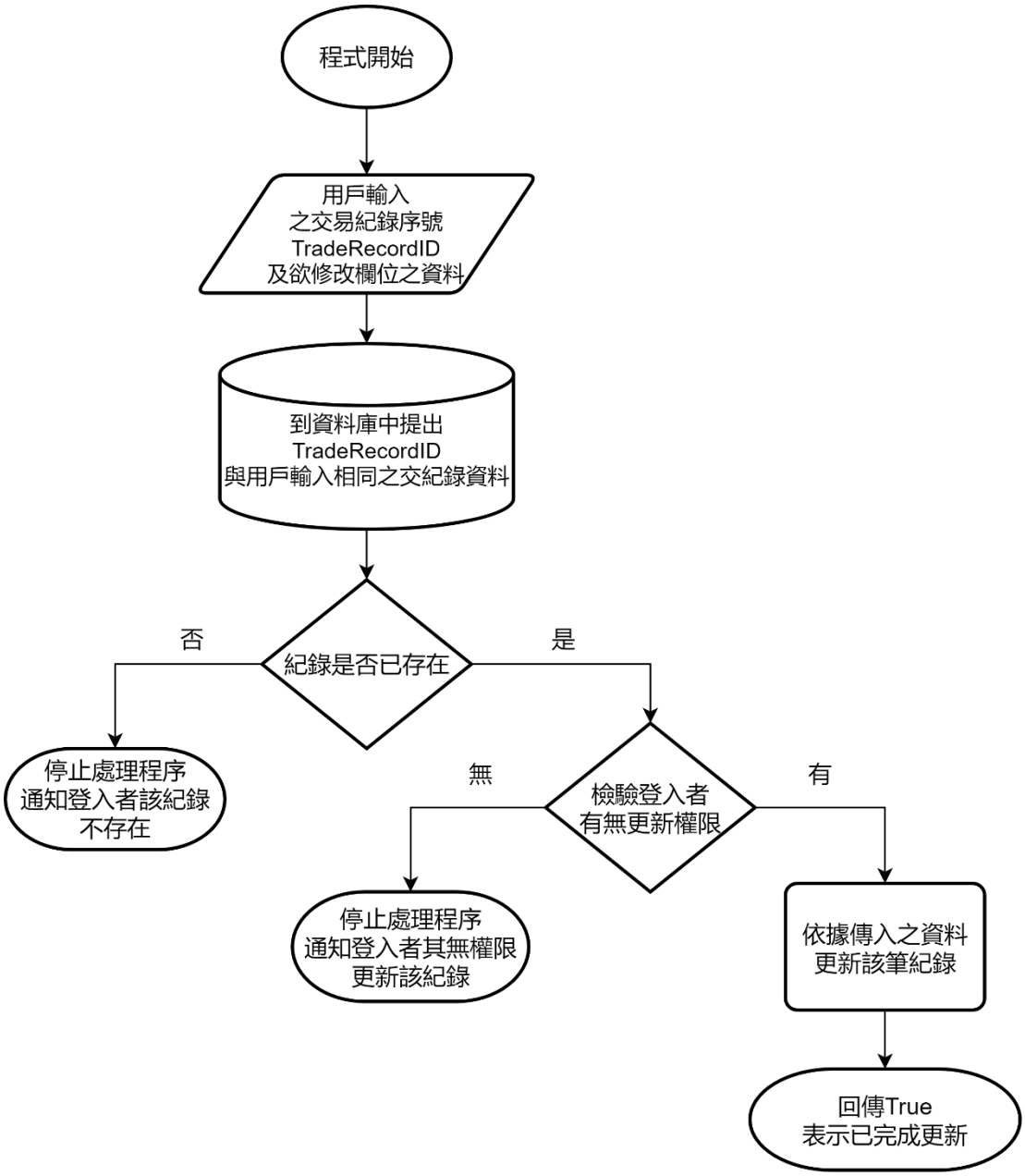

圖 10

UpdateData function 安全性檢核表

UpdateData() 更新交易紀錄		
安全需求項目	**說明**	**評量結果**
與資料庫連線失敗時應採取之應對措施。	與資料庫連線失敗時，於日誌中記錄此次連線失敗事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者出現錯誤。	符合
登入者要求更新的交易紀錄不存在於資料庫中之應對措施。	函式檢查發現該筆資料不存在於資料庫時，於日誌中記錄此次事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者該筆交易紀錄不存在。	符合
登入者帳號與其要求更新之交易紀錄中記錄的帳號不同，意即登入者無權限要求更新該筆交易紀錄之應對措施。	函式檢查發現登入者帳號與其要求刪除之交易紀錄中記錄的帳號不同時，於日誌中記錄此次事件的時間、帳號、IP 位址、事件類型以及其餘用戶有輸入的欄位之資料，並立即停止處理程序，回傳通知登入者其無權限更新該筆交易紀錄。	符合

付款人進入付款頁面之資料檢驗流程

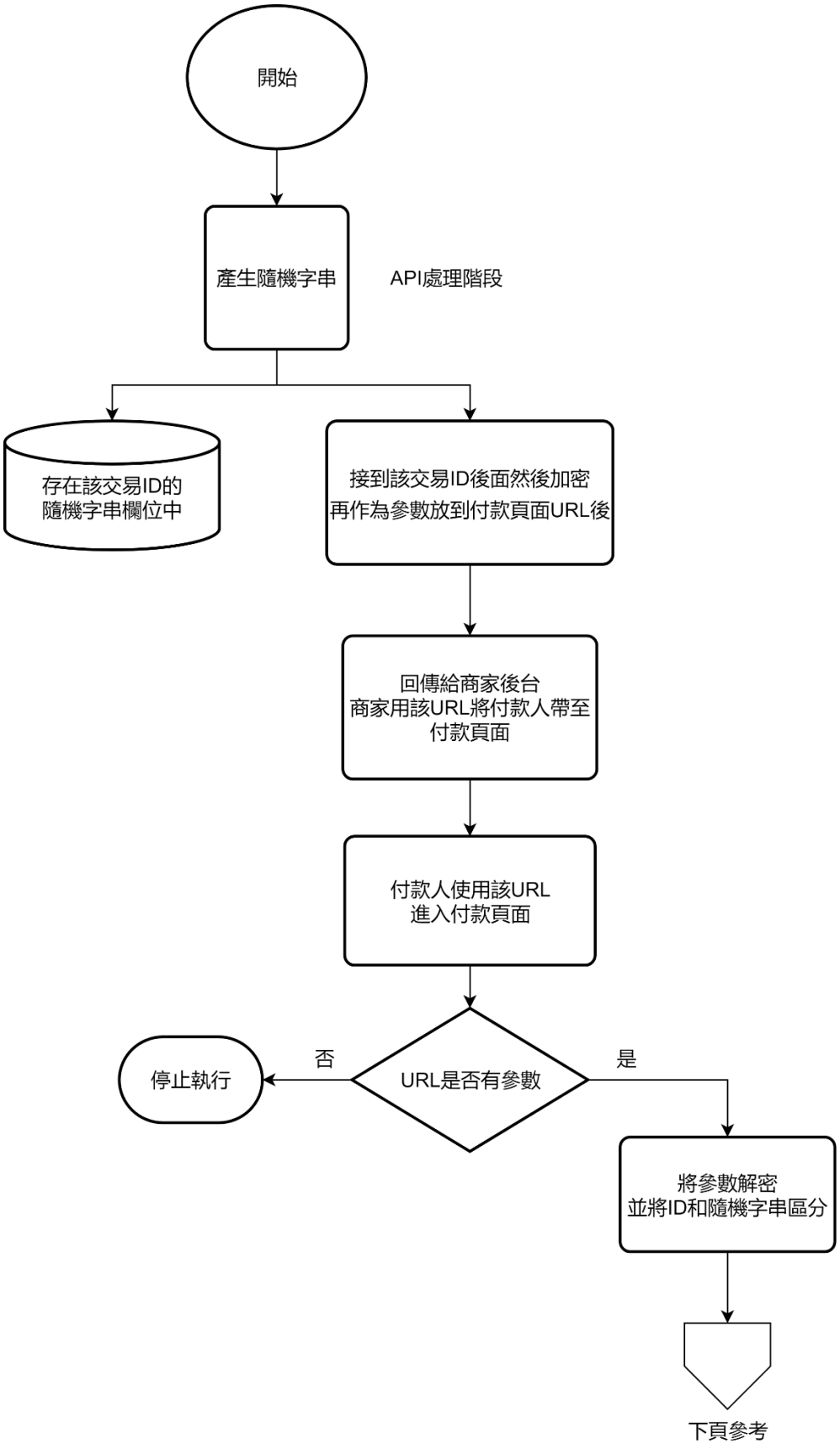

圖 11

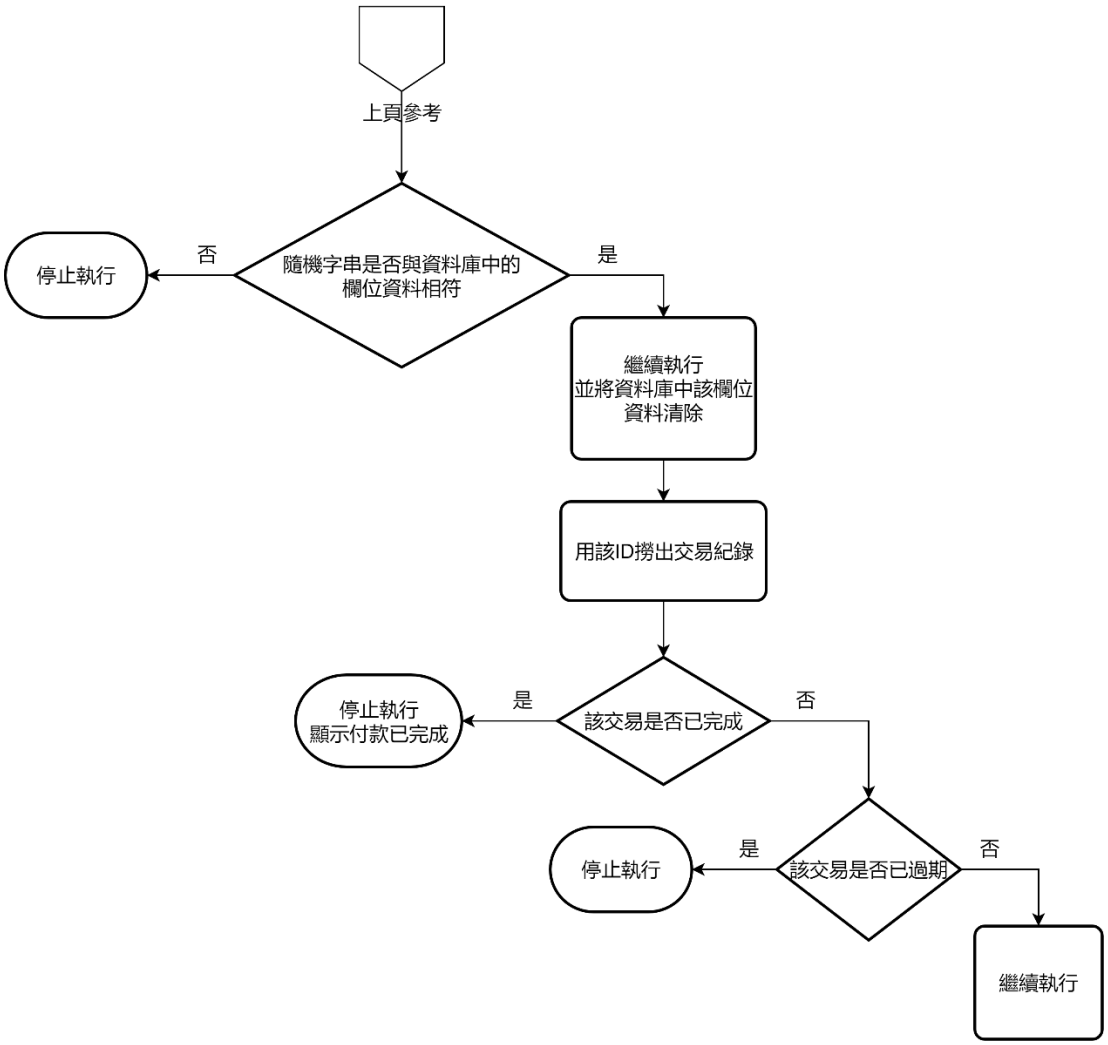

圖 12

圖 11 & 圖 12 說明:

為防止付款人使用本平台生成並提供給商家系統的 URL 進入本平台付款頁面的過程中被有心人士截取，獲得 URL 攜帶的參數得知資料庫中重要參數名稱或是利用 URL 進行 Replay Attack 攻擊，本平台使用了模擬 One-time pad 的防禦策略，在 API 處理流程中要生成 URL 給商家時先生成一組隨機字串接在訂單編號之後再使用與該商家共同持有的私鑰加密，同時存放該隨機字串在該筆紀錄的[RandomString]欄位，等到付款人使用該 URL 進入付款頁面時，將參數解密後與資料庫中該筆紀錄的隨機字串做比對，相同才能進入此頁面，同時刪除資料庫中該筆紀錄的隨機字串，因此就算有人事後拿到該 URL 也會因資料庫中記錄的隨機字串改變而無權限進入付款頁面。

這個策略的特點在於因為加密和解密的角色都是平台本身，所以可以輕易地做到 One-time pad，不需要考慮私鑰交換的問題。

CSRF 防範

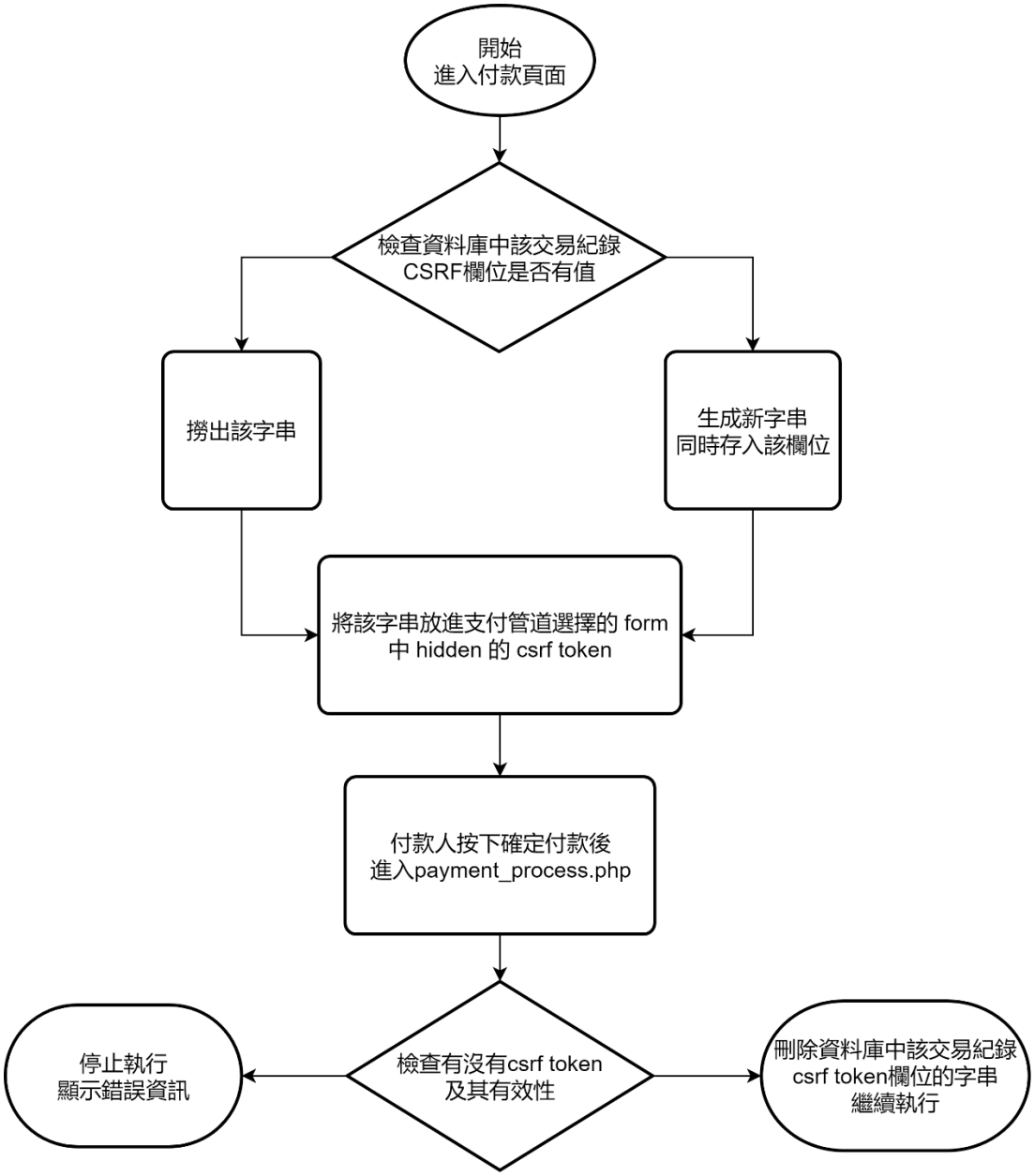

圖 13

圖 13 說明:

這裡需要防範 CSRF 風險的地方是在付款人在付款頁面中選擇支付管道的 Form 選擇完成後送出並跳轉到下一個處理付款交易流程的過程 (payment_process.php)，如果沒有防範的話可能會讓有心人士偽造 Form 讓無防備之付款人填寫因此洩漏資訊或破壞平台中的資料紀錄。本平台的防禦策略與之前防禦 Replay Attack 的策略雷同，模擬 One-time pad，使用同樣方式生成和儲存隨機字串，並將隨機字串放入 Form 中的 csrf token，後面的付款交易處理流程(payment_process.php)會負責檢驗只讓有合法 csrf token 的 Form 繼續執行，能有效防止偽造 Form 試圖侵入系統。

線上支付串接

以用 LinePay 支付為例

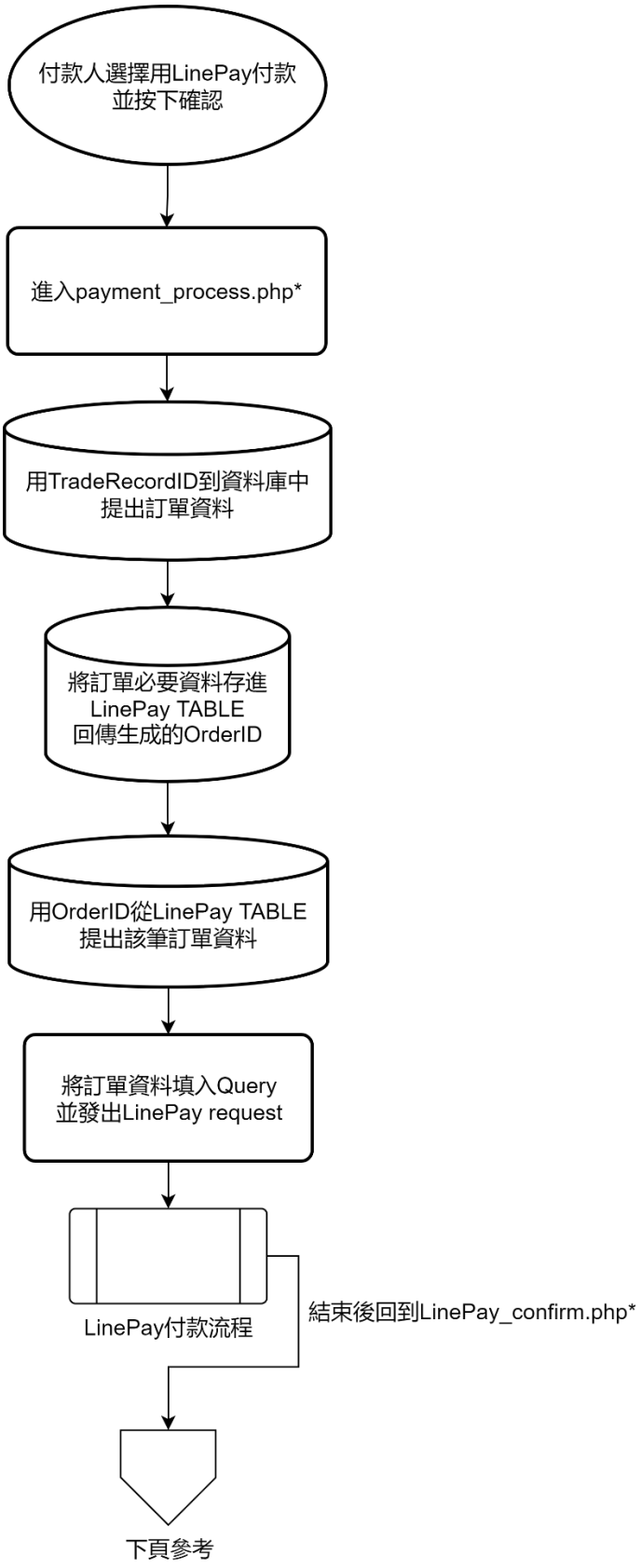

圖 14

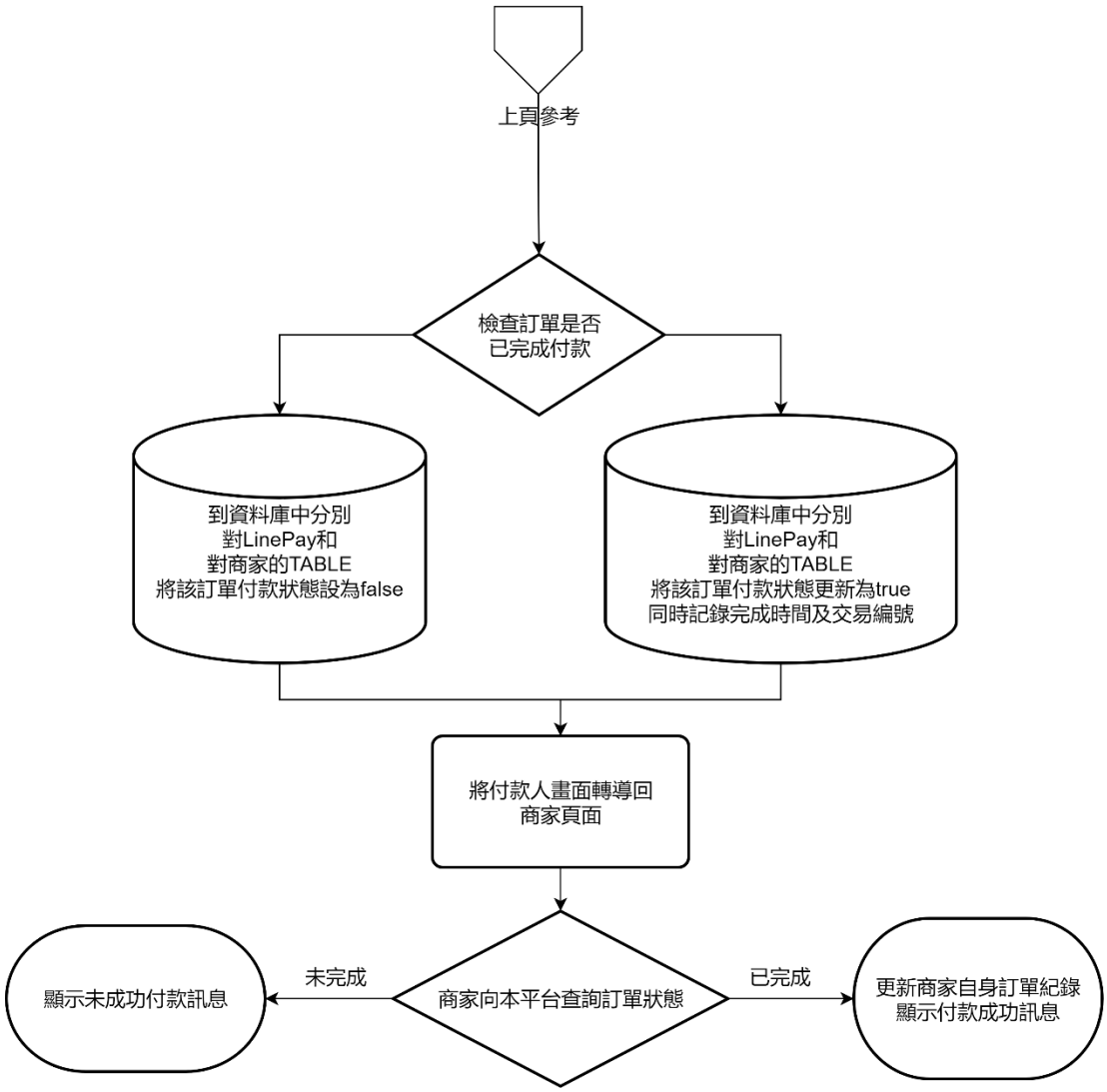

圖 15

到此完成一次成功的付款交易。

未來待開發功能

- **API 指令一次對多筆紀錄進行操作** ： 目前商家呼叫 API 只能一次對一筆訂單紀錄進行操作，如果商家希望能一次統整多筆紀錄進行一次性的操作就會有點麻煩。
- **提供信用卡、超商代碼、ATM 虛擬代碼等更多元支付管道 ：** 本平台目前只完成 LinePay 的串接，而信用卡、超商代碼、ATM 虛擬代碼等支付方式皆有不同的串接技術，所以無法及時串接完成
- **建立後台系統 ：** 目前正在持續開發中，目標建立一後台系統讓商家能夠人工查詢訂單系統，方便對帳。

參考文獻

- 綠界科技全方位金流介接技術文件
  網址: https://developers.ecpay.com.tw/?p=2509 (已改為線上文件)
- 讓我們來談談 CSRF
  網址: https://blog.techbridge.cc/2017/02/25/csrf-introduction/#post-comment-wrapper
- How to Prevent Replay Attacks on Your Website
  網址: https://www.sitepoint.com/how-to-prevent-replay-attacks-on-your-website/
- PHP Manual
  網址: https://www.php.net/manual/en/index.php
- Phpseclib documentation
  網址: https://phpseclib.com/
- LinePay API documentation
  網址: https://pay.line.me/jp/developers/apis/onlineApis?locale=zh_TW