

漸進式網頁應用程式研發實務：以校園通行證管理為例

黃鈴玲^{1*} 翁永昌²

^{1*}大葉大學多媒體數位內容學位學程

51591 彰化縣大村鄉學府路 168 號

²靜宜大學資訊工程系

43301 臺中市沙鹿區臺灣大道七段 200 號

*lhuang@mail.dyu.edu.tw

摘要

漸進式網頁應用程式 (Progressive Web Apps; PWA) 是一種新的網頁技術，讓網頁 App 可以擁有類似於原生 App 的操作體驗，比如依螢幕尺寸自動調整網頁內容、支援訊息推播。本文採用 PWA 來研製校園訪客通行證管理系統。該系統透過智慧型手機，以常見的二維條碼 QR code 形式提供一次性的臨時通行證，校警用手機掃描訪客出示的 QR code 即可決定是否放行。放行時系統會將訪客的身份以推播的方式通知系助理，方便其進行後續的人力調度。用戶端以瀏覽器為界面，可以跨平台 (個人電腦、平板、或手機) 執行。操作介面採用自適應的概念來設計，可以根據螢幕的尺寸調整內容與排版樣式。本系統將臨時通行證的發放電子化，除了簡化行政流程還可以提升訪客的使用體驗。

關鍵詞：通行證管理，網頁 App，訊息推播，自適應網頁設計

Design of a Progressive Web App Pass Management System

LINGLING HUANG^{1*} and YUNG-CHANG WONG²

^{1*}*Bachelor Program for Multimedia Digital Content, Da-Yeh University*

No.168, University Rd., Dacun, Changhua 51591, Taiwan, R.O.C.

²*Department of CSIE, Providence University*

No. 200, Sec. 7, Taiwan Boulevard, Shalu Dist., Taichung City 43301, Taiwan, R.O.C.

*lhuang@mail.dyu.edu.tw

ABSTRACT

Progressive Web Apps (PWAs) combine the best of both the web and apps. PWAs can be made to fit any form factor (desktop, mobile, or tablet) and make re-engagement easy through features such as push notifications. In this study, a novel campus pass management system in PWA was proposed and implemented, which offers a QR code-based one-time pass to campus visitors. School security scans visitor's QR codes to validate their pass. If the pass is valid, the visitor is allowed to enter the campus. The pass management system then simultaneously pushes a message to inform department staff of their visitor's arrival. The proposed system can help streamline administrative processes and



enhance overall visitor experience.

Key Words: pass management, web app, push notification, responsive web design

一、簡介

近年智慧型手機快速普及，手機上安裝的 App 能提供使用者各種便利的服務。除了 App 之外，手機使用者也透過瀏覽器來查閱數以萬計的網站並且使用它們提供的服務。讓網頁也能提供類似 App 的操作體驗是目前技術發展的趨勢，而具備這類功能的網頁我們稱之為 Web application (Web App) [18]。Google 在 2017 年提出漸進式網頁應用程式 (Progressive Web Apps; PWA) 的技術 [4]，讓網頁可以擁有類似於原生 App 的操作體驗，諸如依螢幕尺寸自動調整網頁內容的排版、支援訊息推播等。本文即是採用 PWA 技術來研製適合校園使用的通行證管理系統。

大學校園常有訪客洽公，因此有發放臨時通行證的需求。目前發放臨時通行證的流程大致如下：(1) 系助理確定活動日期與外賓名單；(2) 聯繫校警室告知活動內容以及外賓的基本資料；(3) 系助理製作臨時通行證，然後 e-mail 給校外來賓；(4) 外賓自行列印通行證，進校門時說明來訪緣由並出示通行證。在這個流程當中，有一些細節會影響使用體驗 (user experience, UX)。首先，訪客必須列印通行證並且隨身攜帶。其次，執勤的校警需要詳讀工作日誌了解當天有哪些活動，外賓到達時需人工查閱日誌才能核實外賓的身分 (如透過車牌號碼)。這兩點對於訪客及校警都會造成負擔。除此之外，系助理無法得知外賓到達的時間，只能讓工讀生提早在停車場等候以引導外賓。為了簡化行政流程以及提升訪客體驗，開發通行證管理系統是必要的工作。

隨著電腦視覺 (computer vision) 技術的成熟，使用影像來辨識車牌 [1-3] 已經有許多商業化的應用。由於車牌辨識系統的建置成本較為昂貴，對於不收取停車費的學校比較不受青睞。另一種停車管理的方法是採用 RFID 電子標籤 [8, 13]。電子標籤裡面儲存了一組不會重複的 ID，如果事先把它存放到停車管理系統，那就可以利用這組 ID 來辨識車輛。電子標籤的 ID 必需使用 RFID 讀卡機才能讀取，所以一般大眾無法得知這組 ID。基於這個原因，這種系統並不適合用在校園的訪客通行證管理。

本文提出以智慧型手機搭配 QR code 的形式產生校園臨時通行證。系助理僅需先將訪客資料輸入系統，並將系統產生的 QR code 寄給訪客，訪客到校時用手機出示 QR

code，校警再用手機掃描後即可決定是否放行，並且系統在放行時會推播訊息給系助理，告知某位外賓已經到校。此一臨時通行證系統的建置成本低，優於電子式通行證，在校警管理或查詢車輛的功能上則較傳統紙本更優。本系統並採用了 PWA 技術，讓用戶在存取網頁時可以享有原生 App 的操作體驗。

本論文章節安排如下：第二節將說明相關的背景知識，第三節將介紹本文提出的通行證管理系統架構，第四節展示實作的成果，第五節是結論及未來發展方向。

二、背景知識

本節將說明漸進式網頁應用程式 (PWA) 及其中的功能：響應式網頁設計 (RWD) 與支援推播的 Service Worker。此外，也介紹實際進行訊息推播的 Push Server。

(一) Progressive Web Apps

漸進式網頁應用程式 (Progressive Web Apps; PWA) 是 Google 近年大力推廣的網頁前端開發概念與技術 [4]，目的是讓使用者在觀看網頁時，能有類似手機 App 的操作體驗，因此 PWA 的特性主要歸納成以下幾項：

1. **Progressive**：網頁以漸進式增強的方式呈現內容，可快速啟動並在背景中漸進的載入資源。
2. **Responsive**：能適應各種尺寸的裝置。
3. **Connectivity Independent and Fresh**：透過 Service Worker 在背景執行的協助，在低速網路甚至離線時都能瀏覽網站，並且能自動更新網站內容，並發送推播通知。
4. **Safe**：使用 HTTPS 協定，確保資料安全性。

本論文由於開發的是較單純的校園通行證系統，因此在 PWA 的實作上，主要針對 Responsive 及 Service Worker 兩個部分來進行。

(二) Responsive Web Design

響應式網頁設計 (Responsive Web Design, RWD) 又稱為自適應網頁設計，是網頁前端為配合各種瀏覽裝置而產生的優化技術 [9, 19]。RWD 採用彈性畫面設計的理念，可以根據螢幕的尺寸 (解析度) 自動調整頁面的佈局與排版，讓不同的設備都可以正常瀏覽同一份網頁。



舉例來說，在 24 吋螢幕的個人電腦使用者可以看到完整的網頁資訊，包括水平排列的功能表選項。換成 10 吋螢幕的平板，原先水平排列的功能表選項會自動改成下拉式選單，以適應較小的螢幕。如果換成 5 吋螢幕的手機，原本左右排版的頁面都會變成直條狀以利使用者閱讀。網頁設計採用 RWD 的好處除了適合各類型裝置瀏覽外，工程師在製作與維護網頁時，不必另外再製作為該網頁製作一個手機版。

(三) Service Worker

Service Worker [10, 17] 是一隻獨立於瀏覽器之外的 script 程式，可以在背景模式下執行。Service Worker 扮演代理伺服器 (proxy server) 的角色，周旋於瀏覽器和網頁伺服器 (Web server) 之間：Service Worker 會攔截瀏覽器發出的 HTTP 請求 (request)，處理後再轉送給網頁伺服器。當網頁伺服器送出 HTTP 回應 (response)，Service Worker 會先緩存 (cache) 該回應，再將該回應加工處理後的結果遞送給瀏覽器。

Service Worker 緩解了 Web 用戶無法離線瀏覽的問題。當行動裝置處於離線狀態，沒有辦法得到更多資料之前，仍然可以透過 Service Worker 取得基本的服務。俟網路恢復連線，Service Worker 可以自動進行資料同步。另外，Service Worker 還可以提供 App 版本更新、推播通知等增值功能，提升用戶的使用體驗。

(四) Push Server

前面介紹的 Service Worker 是作用在使用者的瀏覽器上接收通知，而實際將訊息推播出來的則是 Push Server，目前有許多第三方平台提供訊息推播的服務。以能自行架站並做內容管理的 WordPress [20] 來說，就有需多外掛能提供 Web Push 的服務，如 One Signal [11] 等。對於不是使用 WordPress 所架設的網站，One Signal、SendPulse [15] 及 iZooto [7] 等也提供了推播服務。

以 iZooto 來說，網站管理者只要註冊並填寫一個表單，提供自己網站的網址，再將 iZooto 提供連接 API 的程式貼在自己網站的首頁，就能提供推播訊息的服務。想要推播的訊息內容可以直接填在 iZooto 的表單裡馬上推播出去，也可以寫進一個 XML 檔，指定 iZooto 去讀取再推播，但這種方式會有較大的延遲。當網站的使用人數較多，或是需要將群眾分類推播時，iZooto 服務就需要收費。

Firebase Cloud Message (FCM) [5] 是 Google 提供的訊息推播服務，支援跨平台 iOS、Android 及 Web 三種的推播

模式。由於 Firebase 是相當受 app 開發者歡迎的後端服務平台，並與 Google 其他產品有深度的整合，Google 在網站上也提供了詳細的使用教學[6]，因此，在我們建置的通行證管理系統中，我們選擇透過 FCM 的服務來提供推播的功能。

三、通行證管理機制

本節依序說明通行證管理系統的架構，以及訊息推播的流程。

(一) 系統架構

通行證管理系統的功能架構如圖 1 所示，分為用戶端與伺服器端。

用戶端採用自適應網頁的概念來設計，以方便各種尺寸的螢幕瀏覽。依照使用者的身份可區分為三個模組：

1. 系助理 (Assistant) 模組 (圖 1 (a))：系助理使用瀏覽器登入後可以執行底下的工作：
 - (1) 新增活動資訊：輸入活動名稱、日期地點、與訪客資料 (姓名、車牌號碼、手機號碼)。
 - (2) 產生通行證：輸入完成後，系統將資料寫入資料庫產生一筆紀錄，並產生唯一的編號，再將訪客要進入校門時的驗證網址與該編號參數結合，由系統的程式自動產生一個獨特的 QR code。由於編號與訪客的車牌及活動日期有相關性，每位訪客取得的 QR code 都不相同，他人無法拷貝冒用。系助理將 QR code 圖檔轉寄給訪客，訪客進入校園時就可以用手機顯示 QR code 作為臨時通行證。

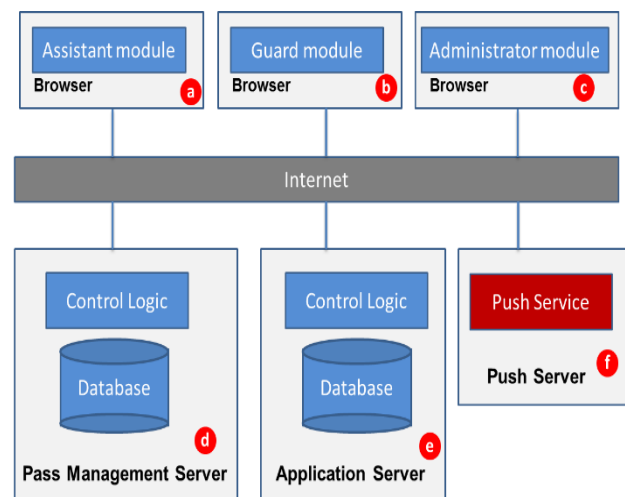


圖 1. 系統功能架構圖



2. 校警 (Guard) 模組 (圖 1 (b)):

- (1) 訪客出示 QR code 時，校警使用手機的 QR code reader 進行掃描。掃描後會啟動手機的瀏覽器並且將掃描的資訊送回通行證管理伺服器。
- (2) 如果 QR code 是有效的，瀏覽器將會顯示訪客要參加的活動名稱以及訪客的車牌號碼。校警僅需核對車牌是否吻合即可放行。

3. 管理者 (Administrator) 模組 (圖 1 (c)):

供系統管理員登入，可以新增系助理/校警帳號、修改資料庫的內容等。

接下來說明伺服器端的部分。系統包含底下三個伺服器：通行證管理 (Pass Management) 伺服器、應用 (Application) 伺服器、以及推播 (Push) 伺服器。

1. 通行證管理伺服器 (圖 1 (d))：負責產生通行證以及進行身分認證。資料庫中存有訪客資訊，如姓名、車牌、以及手機號碼。當校警掃描訪客的 QR code 時會把 QR code 上面攜帶的資訊送給通行證管理伺服器。伺服器執行身分辨識後把結果傳回給校警，同時也會把訪客已經到達的訊息送給應用伺服器。
2. 應用伺服器 (圖 1 (e))：負責把通行證管理伺服器送來的資訊轉送給訊息推播伺服器。當系助理同意接受推播通知，該瀏覽器所獲得的身分權杖 (Token) 會一併儲存在應用伺服器裡。
3. 推播伺服器 (圖 1 (f))：負責把應用伺服器送過來的訊息推播給指定的系助理模組。

(二) 訊息推播

訊息推播包含兩個階段：註冊以及推播通知。在註冊階段，系助理模組須向訊息推播伺服器註冊，以便取得身分權杖 (Token)。透過這個權杖，系助理模組才能收到器發出來的通知。圖 2 是系助理模組的註冊流程，相關步驟說明如下：

- 步驟 1. 系助理模組第一次啟動時，網頁內嵌的程式會詢問是否同意安裝伴生的 Service Worker (SW)，並且授權使用訊息推播。
- 步驟 2. 徵得使用者的同意後，瀏覽器開始安裝 SW。安裝完成後瀏覽器會啟動 SW，讓它負責監聽所有外來的推播通知。
- 步驟 3. 系助理模組向推播伺服器送出註冊訊息 (Register)，希望取得身分權杖。
- 步驟 4. 推播伺服器收到註冊訊息後發出身分權杖。此權杖將

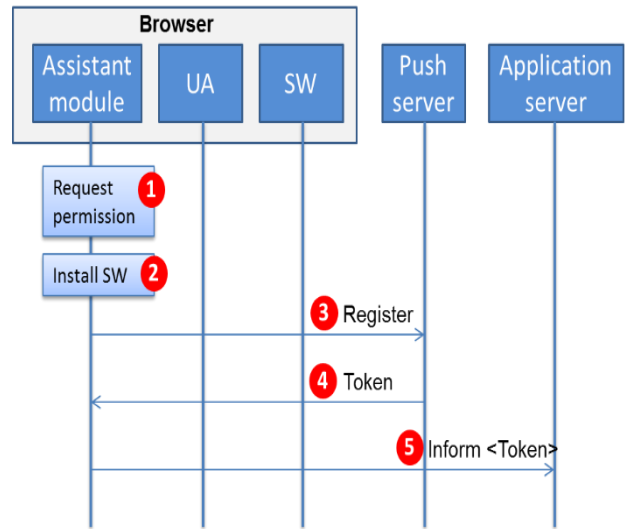


圖 2. 系助理模組註冊流程

做為訊息推播的目的地識別代號。

步驟 5. 系助理模組收到權杖後須轉送一份給應用伺服器，同時將權杖儲存一份在應用伺服器，以便後續將訊息推播給指定的系助理。

通行證管理伺服器收到校警送來的訪客資訊後會執行身分辨識，如果身分辨識通過，校警會看到「驗證通過」的訊息，當確認車牌相符後，按下「已到達」的按鈕，伺服器就需要通知系助理模組，告知某位訪客已經到校。為了完成這件工作，伺服器須進行訊息推播。圖 3 為訊息推播的流程，共包含四個步驟：

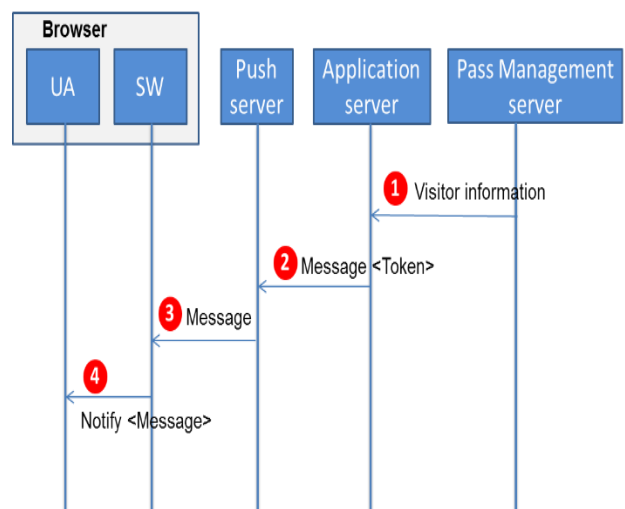


圖 3. 訊息推播流程



步驟 1. 通行證管理伺服器透過資料庫查詢，取得設定該通行證的系助理資訊，再將該訪客已經到達的訊息發送給應用伺服器，請它轉告系助理模組。

步驟 2. 應用伺服器找出系助理模組的身分權杖，連同訪客資訊一起轉送給推播伺服器。

步驟 3. 推播伺服器藉由身分權杖可以得知系助理模組的 Service Worker (SW) 所在的位置。確定位置之後，推播伺服器會把相關的資訊推播給目的地 SW。

步驟 4. SW 收到推播訊息後會通知瀏覽器，系助理的瀏覽器右下角就會彈出一個小視窗來顯示訪客已經到達的資訊，並有超連結方便助理點擊連到伺服器，觀看訪客到達的情形。



圖 5. 系統管理員-人員列表

四、系統實作

本節將說明校園通行證管理系統的實作方法，並且展示實作的成果。以下將分成通行證管理、推播通知、效能評估等三個部分進行介紹。

(一) 通行證管理

通行證管理系統的伺服器端採用 PHP [14] 搭配 MySQL 資料庫 [12] 來開發。圖 4 至圖 14 為通行證管理系統操作畫面的截圖。其中，圖 4 為系統首頁，除說明系統功能外，並提供系統管理員、系助理與校警等三種身分人員的登入。



圖 4. 登入頁面

圖 5 為系統管理員（預設是電算中心）登入後的畫面，一開始先提供各系助理與校警的「人員列表」，並能新增修改與刪除人員。由於本系統的 PC 版的畫面較小，我們將其中手機版的畫面顯示在圖 6。同一個頁面在不同裝置能顯示不同解析度的結果，方便使用者觀看，此即響應式網頁設計（RWD）的效果。

系統管理員若點選主選單裡的「活動列表」，會進入已建置的活動清單，也就是各系助理所新增的所有活動，並依登入當天的日期分為未來活動（如圖 7）與逾期活動（如圖 8）兩類，並依照活動日期排序，使開始日期最接近今天的活動顯示在最上面。



圖 6. 系統管理員-人員列表（手機版）





圖 7. 系統管理員-未來活動清單



圖 9. 系助理-活動清單



圖 8. 系統管理員-逾期活動清單

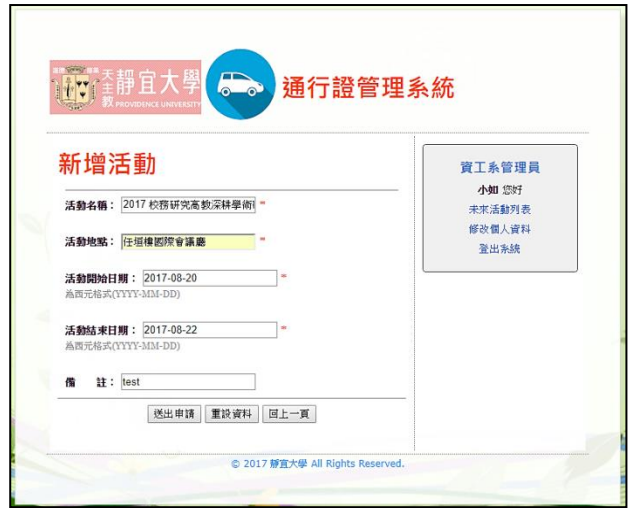


圖 10. 系助理-新增活動表單

圖 9 為系助理登入後的畫面，提供各系助理自行建立的活動列表，同樣分成未來與逾期兩類；對於列表上已建立的活動可進行通行證管理，也能新增修改刪除活動，及修改個人資料。點擊「新增活動」按鈕時，會進入如圖 10 的表單。

當活動建立完成後，即可在圖 9 的活動列表看到它，點擊左邊的「新增車證」按鈕，會進入如圖 11 的「新增通行證」表單，填寫完訪客資料並送出後，會出現如圖 12 的畫面，提供 QR code，系助理只要將此 QR code 寄給訪客，訪客即可憑此通行證進入校園。



圖 11. 系助理-新增指定活動之通行證





圖 12. 系助理-產生 QR code

由於系助理輸入訪客資料時，每筆拜訪的資料（主要是車牌及活動日期）將在系統資料庫裡新增一筆停車紀錄，並會產生唯一的編號，因此 QR code 的內容主要就是一個攜帶此編號參數的網址。由於該網址所對應的網頁程式會先檢查是否為校警登入，因此一般人掃描 QR code 僅能看到登入頁。即使以相同的網址帶不同編號偽造假的 QR code，由於無法在系統裡為假的編號建立符合的車牌號碼及拜訪日期，因此校警掃描後也無法驗證通過。即使同一訪客車牌，在活動過後仍想進入校園，由於拜訪日期不正確，前次用過的 QR code 也無法再次使用。

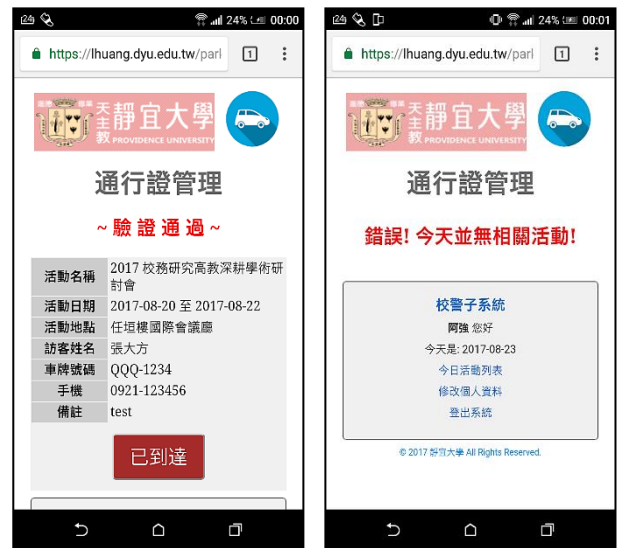
圖 13 是系助理在圖 9 的畫面點擊「車證列表」按鈕的結果，會列出已設定的通行證列表，若使用者已到達，也會列出到達的時間。

圖 14 為訪客持 QR code 進入校園時，校警掃描後所聯結的網頁畫面，由於校警在校園大門執勤，使用手機驗證較為方便，因此校警子系統的頁面採用響應式設計（RWD），方便手機瀏覽。考量到系統之安全性，QR code 導向的網頁會先檢驗校警是否有登入。校警可先在手機上登入通行證管理系統，並保持瀏覽器開啟狀態，使伺服器端的 Session 啟動維持登入狀態，加快驗證的速度。若校警忘了先登入，該網頁仍會要求校警登入，但此時校警就需要輸入帳密才能看到結果。

當 QR code 導向的網頁通過 Session 登入驗證後，若日期正確，會顯示如圖 14 (a) 之「驗證通過」畫面，並顯示車牌號碼提供校警比對。為避免有人持他人之 QR code 進入校園，校警仍需比對車牌是否一致。此外，若 QR code 所對應的活動日期不正確，則會顯示如圖 14 (b) 之錯誤訊息。此外，校警只要按下畫面中的「已到達」按鈕，系統會將該訪客「已到達」的訊息發送通知給系助理。



圖 13. 系助理-指定活動之通行證列表



(a) 驗證通過

(b) 驗證失敗

圖 14. 校警掃描訪客 QR Code 的結果



(二) 推播通知

如前面第二節所述，本文之推播通知功能採用 Google 的 Firebase Cloud Message (FCM) 做為第三方服務平台。此外，由於訊息推播功能需要使用 HTTPS 協定，以確保資料傳輸的安全，因此伺服器需要取得 SSL 憑證，本文所建置之網站是向 SSL For Free [16]申請免費的 SSL 憑證。

本系統開發後期，我們加入對 HTTPS 的支援後，卻讓 QR code 的模組出現問題，經檢查代碼，發現有些 3rd party 程式庫只能透過 HTTP 來叫用。為了解決問題，我們把該程式庫備份到本系統伺服器後，系統即可順利執行。通行證管理系統在推播部分的實作成果如圖 15，系助理登入後會先詢問是否接受推播通知，而圖 16 是系助理的電腦收到通知時的畫面。

(三) 系統效能評估

由於智慧型手機相當普及，校警及訪客應該都有自己的手機，因此本系統只需要建置網站，花費的成本很低。系統的效能評估可從三個部分來看：系助理需事先在系統輸入訪客資料，並將產生的 QR code 寄給訪客，整個過程應該不超過兩分鐘；訪客部分，只要在進入校園時將 email 中的 QR code 展示給校警看即可，無須擔心效能；校警部分，手機需開瀏覽器並事先登入本系統，在訪客到達時開啟具備掃描 QR code 功能的 APP，掃描後點入顯示的網址（含編號參數），比對車牌，若車牌正確，按下「已到達」的按鈕，系統再發通知給助理。由於整個過程中系統只需從資料庫將該編號所對應的紀錄叫出來，比對日期再顯示結果，使用的又是校園內部的網路，因此應該也只需要幾秒鐘，不至於造成校門車輛堵塞。

由於目前 QR code 的辨識率極高，而且訪客所展示的 QR code 是顯示在手機裡不會變形，因此不太需要擔心辨識的問題。若仍發生特殊狀況造成無法讀取，或訪客忘了帶 QR code，校警可由系統中針對當天已登記要入校的車牌進行查詢，查到該車牌即可放行。

五、結論與未來工作

本文提出具備推播通知功能的校園通行證管理系統，系統具有以下特色：(1) 跨平台：用戶端以瀏覽器為界面，可以在個人電腦、平板、或手機等平台執行。(2) 使用友善：用戶端採用自適應網頁的概念來設計，方便各種尺寸的螢幕瀏覽。(3) 提升使用體驗：執勤的校警掃描 QR code 即可辨

識放行，提高行政效率；訪客無須列印紙本通行證或是接受校警的盤查，可有愉快的參訪體驗。(4) 訪客到校通知：訪客到校後會推播到達的訊息給建立該通行證的系助理。(5) 建置成本低：由於智慧型手機已極為普遍，本系統只需再建置網站，因此花費的成本較其他電子式的停車證更為便宜。(6) 易於管理：傳統紙本通行證較難查詢，遇到違規停車的訪客很難通知，本系統中校警可以很容易由車牌號碼查出訪客姓名及手機，管理較為方便。

本論文所建置的校園通行證管理系統尚未臻完善，以大型研討會來說，若有上百位會員申請通行證，系助理一筆一筆輸入相當費時，比較好的方式是用檔案匯入的方式，或甚至將研討會報名功能與本系統結合，讓訪客報名研討會時就能同時取得本系統產生的 QR code。此外，系助理會在研討會當天開始前收到上百份到達通知，其實反而造成困擾，因此，讓訪客分類或讓系助理能選擇只針對重要貴賓才發送通知的功能，是未來在實作上需要改進的方向。



圖 15. 系助理-詢問是否同意開放推播通知的權限

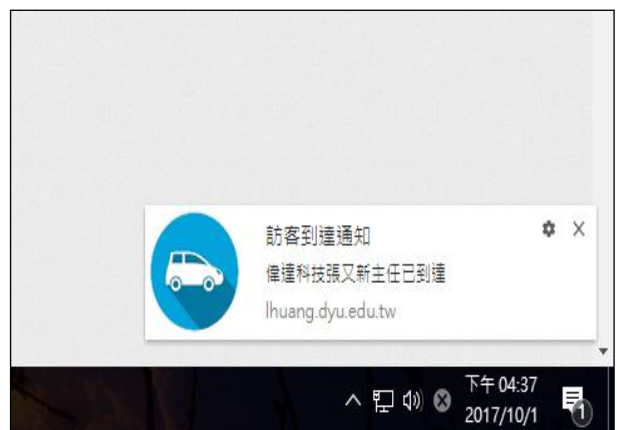


圖 16. 系助理-收到通知



誌謝

本論文由「靜宜大學 105 學年教學研究能量精進計畫-實務研究型」補助，計畫編號 PU105-SPR1-006。

參考文獻

1. Anagnostopoulos, C. E. (2014) License plate recognition: A brief tutorial. *IEEE Intelligent Transportation Systems Magazine*, 6(1), 59–67.
2. Du, H., M. Ibrahim, M. Shehata and W. Badawy (2012) Automatic license plate recognition (ALPR): a state-of-the-art review. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2), 311 – 325.
3. Farajian, N. and M. Rahimi (2014) Algorithms for licenseplate detection: A survey. 2014 International Congress on Technology, Communication and Knowledge, Mashhad, Iran.
4. Google, Inc. (2018) Progressive Web Apps, Retrieved January 03, 2018, from <https://developers.google.com/web/progressive-web-apps>.
5. Google, Inc. (2018) Firebase cloud messaging, Retrieved January 06, 2018, from <https://firebase.google.com/docs/cloud-messaging>.
6. Google, Inc. (2018) Adding push notifications to a Web App, Retrieved January 08, 2018, from <https://developers.google.com/web/fundamentals/codelabs/push-notifications>.
7. iZooto (2018) Push notifications, Retrieved January 10, 2018, from <https://www.izooto.com.tw>.
8. Li, C. (2010) Automatic vehicle identification (AVI) system based on RFID. International Conference on Anti-Counterfeiting Security and Identification in Communication, Chengdu, China.
9. Mozilla and individual contributors (2018) Responsive design mode, Retrieved January 02, 2018, from https://developer.mozilla.org/en-US/docs/Tools/Responsive_Design_Mode.
10. Mozilla and individual contributors (2018) Service worker API, Retrieved January 02, 2018, from https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API.
11. OneSignal (2018) OneSignal - free web push notifications, Retrieved January 03, 2018, from <https://wordpress.org/plugins/onesignal-free-web-push-notifications>.
12. Oracle Corporation (2018) MySQL, Retrieved January 02, 2018, from <https://www.mysql.com>.
13. Pala, Z. and N. Inanc (2007) Smart parking applications using RFID technology. 1st Annual RFID Eurasia, Istanbul, Turkey.
14. PHP Group (2018) PHP: hypertext preprocessor, Retrieved January 04, 2018, from <http://php.net>.
15. SendPulse (2017) Web push notifications, Retrieved September 08, 2017, from <https://sendpulse.com/features/webpush>.
16. SSL For Free (2018) Free SSL certificates in minutes, Retrieved January 05, 2018, from <https://www.sslforfree.com>.
17. W3C (2017) Service workers nightly, Retrieved August 23, 2017, from <https://w3c.github.io/ServiceWorker>.
18. Wikipedia (2018) Web application, Retrieved January 10, 2018, from https://en.wikipedia.org/wiki/Web_application.
19. Wikipedia (2017) Responsive web design, Retrieved July 15, 2017, from https://en.wikipedia.org/wiki/Responsive_web_design.
20. WordPress (2018) Create an outstanding website, Retrieved January 10, 2018, from <https://wordpress.com/learn-more/?v=site>.

收件：106.10.16 修正：107.01.11 接受：107.03.16

