

107 學年度築夢獎學金 成果報告

清大智慧宿舍計畫 智慧洗衣機專案



2020 年 9 月

目錄

壹、 專案概述	3
貳、 團隊成員	4
參、 計畫時程	5
肆、 專案成果	6
一、 行政管理	6
二、 系統架構	7
1. 需求	7
2. 各方案優缺點比較	7
3. 解決方案	8
三、 硬體、電路	9
1. esp8266 version	9
2. linkit version	11
四、 軟體	14
五、 資料處理	16
1. 邏輯判斷	16
2. 機器學習	18
六、 使用者介面	21
1. Linebot	21
2. 網頁	22
七、 宣傳、推廣	23
伍、 成員心得	25



壹、專案概述

清大智慧宿舍計畫於 107 年度下學期由載物、厚德兩書院的學生自主發起，以改造宿舍為目標，提供學生一個實現課堂所學，實際設計製造相關智慧設備的機會，並利用住宿書院成員凝聚力強、公共事務參與度極高等特性，提供一個良好的設備測試環境，這是個發展實驗性計畫的好機會。本計畫於 107 學年度獲得築夢獎學金經費支持。第一個著手開始進行的計畫為智慧洗衣機專案。

仁、實齋的洗衣間位於一樓戶外，常聽說住在高樓層的齋民們抬著衣服下樓梯，洗衣機卻全在運轉，需要來往跑許多趟才能順利排到空檔洗衣；洗衣的行程時間也非常不固定，不是下樓卻發現尚未洗好，就是早已經洗好被其他人取出，造成許多不便。於是我們設想是否能運用智慧宿舍計畫的資源，開發一套智慧洗衣機系統，提供宿舍洗衣機使用者們一個即時取得洗衣機使用狀況的管道。

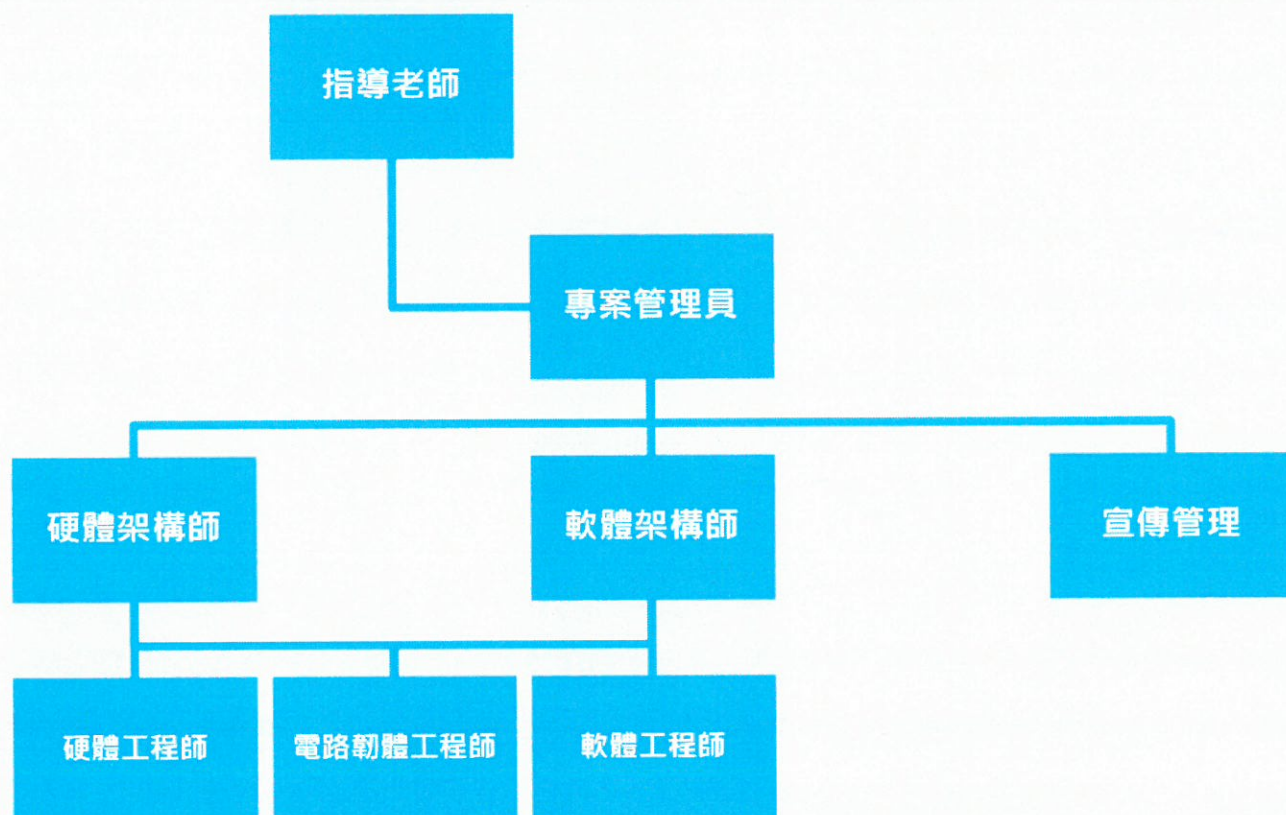
最終，我們順利在實齋女、仁齋男兩個洗衣場建置我們的智慧洗衣系統，使用者可以透過網頁、Line 查詢洗衣狀態與剩餘時間，此套系統亦準備好可以推廣到校內其他宿舍洗衣場。此結案報告將描述此系統的開發過程



貳、團隊成員

指導老師：動機系 陳榮順 教授

陳佐任	專案管理	動機/實驗教育 21
蔡闊光	軟體架構師	資工 22
倫皓正	硬體架構師	電機 22
陳柏霖	電路韌體工程師	電機 21
阮棠	軟體工程師	資工 23
鄭巧苓	軟體工程師	電機 20
李佩勳	硬體工程師	動機 21
張佩瑜	宣傳管理	人社 22



參、計畫時程

此專案分成四個階段：系統規劃、原型、量產、後期

階段	時程	名稱	簡介
系統規劃	Feb 2019 - May 2020	專案規劃	進行專案人力招募、分工
		系統評估	開始定義系統需求
原型	May 2019 - Dec 2020	原型製造	將設計內容實際加工製造
		系統測試	進行系統整合測試
量產版本	Dec 2020 - Mar 2020	量產版本製造建置	以原型為基礎，設計易製造、耐用的量產版本
		開放使用	加工、組裝量產版本，並隨著宿舍施工一起發包、裝設到各個房間
後期	Mar 2020 - Aug 2020	使用者意見蒐集	藉由使用者體驗系統後的回饋，進行系統的修正、改進

時間表：

2019/03/05 招募公告	2019/12/11 狀態邏輯判斷完成
2019/04/28 初版 IOT 設備完成加工測試	2020/02/26 linkit 版本 IOT 小型量產完成
2019/05/02 初版軟體系統完成測試	2020/03/08 line bot 開始內部測試
2019/05/04 收到第一筆數據	2020/03/15 開放 line 使用
2019/09/30 esp 版本 IOT 小型量產完成	2020/03/21 加入機器學習
2019/12/03 內部整合測試開始	



肆、專案成果

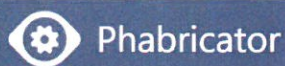
一、行政管理

專案管理上，我們使用 3 個工具進行團隊協作、溝通、檔案交換的媒介，分別是：phabricator、github、google drive。此專案開發工具的軟體架構，目前的 10 人小團隊規模就能夠輕易管理，未來也足以應付數十人甚至百人的團隊規模。

以下說明各軟體功能：

軟體

簡介



phabricator 是一套開源軟體，原為 facebook 內部專案管理工具。我們將它建置在我們自己的伺服器上，專屬於團隊內部使用，使用 web 介面操作編輯。此軟體用來記錄每個人的 task、專案開發的過程、記錄會議的討論內容。



github 用於軟體協作、存檔、版控，利於軟體在開發人員電腦、伺服器間進程式編寫、執行。據我們所知，校內僅有極少數學生團隊利用 github 進行專業的軟體版本控制解決方案。



我們利用校內的 G suite 授權，建立智慧宿舍計畫專屬的 google drive 小組雲端用於 1. 儲存硬體電路、機構的設計、加工檔案 2. 行政財務、宣傳檔案，小組雲端的權限設定能夠輕易管理使用者權限，並有簡易版本控制功能，不必擔心檔案誤刪遺失

二、系統架構

1.需求



系統需求：

- ’ IOT 設備故障不影響洗衣機的正常使用，即使 IOT 設備不幸損毀，不會連帶造成洗衣機廠商的損失。



設備需求：

- ’ 有能夠感測、回傳洗衣機資料的硬體設備
- ’ 有資料庫能夠儲存感測資料
- ’ 有能夠判斷使用狀態的軟體
- ’ 有提供使用者取得資料的介面
- ’ 不能使用過多電能造成宿舍額外開銷

2.各方案的優缺點比較

對於軟體環境的伺服器需求，我們進行優缺點評估，考量如下：

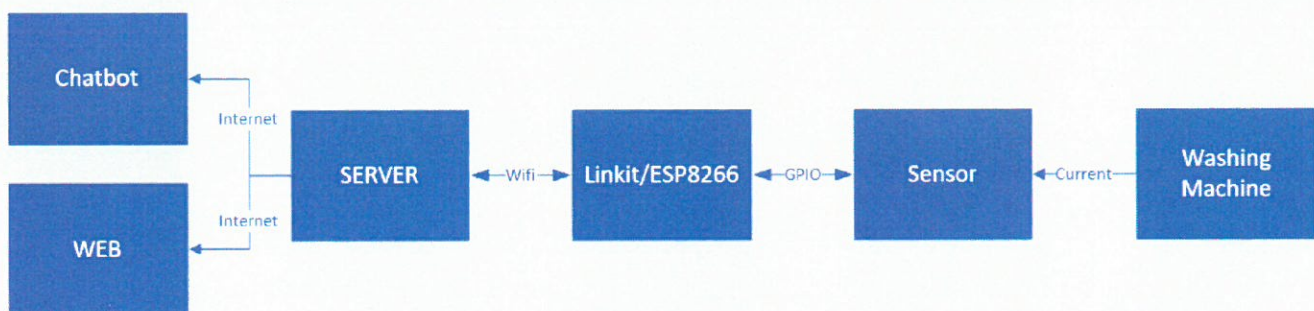
	使用 Azure/AWS/GCP 平台	自組 server
金額	需長期訂閱(~1000/月)	採購 server 經費(20000)
架構	使用現有提供的固定架構	較有彈性，我們有許多選擇可以自行設計
建立、維護	需要有相關操作經驗，學習較花時間	有程式基礎即可，須有完整檔案提供交接

最終，我們選擇自行組裝 server，並架設於有固定 IP 的宿舍網路，此方案我們可以輕易地掌控完整架構，對底層軟體較有掌握。

	簡述	優點	缺點
感測電流	利用電流感測器測量洗衣機的用電情形，藉此判斷運作/未運作	不須改裝、侵入洗衣機設備	電流資料需要進行資料處理方能判斷運作/未運作
感測震動	利用震動感測器測量洗衣機運作時的震動	不須侵入洗衣機設備	震動來源雜亂，資料處理十分複雜 感測器需固定於洗衣機上
取得洗衣機內部資訊	與洗衣機廠商合作，取得洗衣機內部處理器資訊	資料正確率極高	需要進行洗衣機改裝
感測螢幕	利用相機拍攝洗衣機液晶螢幕	資料正確率極高	影像辨識需要不少運算資源 會有隱私疑慮

3. 解決方案

為滿足上述需求，經討論並參考過去經驗，我們開出以下解決方案：



首先由感應電流感測器測量洗衣機使用市電的電流大小，由 IOT 設備的微處理器晶片將數據上傳伺服器資料庫；伺服器中的軟體取出電流資料，判定使用情形、計算預估時間，再儲存結果於資料庫；網頁 chatbot 服務從伺服器取得資料並藉由圖形化介面提供給使用者。

三、硬體、電路

1. 架構

依照系統解決方案，我們決定 IOT 硬體設備需求如下：

- ’ 有非接觸式感應電流感測器
- ’ 有微處理器可以處理電流感測器資料
- ’ 能夠支援通訊協定將數據上傳至伺服器資料庫

據此提出 esp8266、linkit 兩個解決方案，將於以下章節詳細說明。

2. esp8266 version

A. 電路

- ’ 電流測量方式：

我們在洗衣機的電源線上接了電流互感器 (Current Transformer)，可以得到一個參考電流（比洗衣機使用的電流小 800 倍）。透過橋式整流器把參考電流轉成直流電流後，讓電流流過一個電阻，然後微控制器的 ADC 讀取電阻的壓降就可知道電流的大小。

- ’ 通訊界面：

ESP8266 版本的電路是一個微控制器負責測量一台洗衣機的電流。ESP8266 內建 Wi-Fi 的功能，可以透過 HTTP 直接把測量到的電流值傳回伺服器。傳送資料的時候，我們也傳送協議的版本，以便以後可以擴充傳輸的資料也不影響舊版本的使用。

- ’ 電源：

電路板上有兩個 DC 接頭，所以相近的 PCB 可以串接在一起，共用一個 DC 電源。



B. 效益

ESP8266 這個微控制器耗電量很低，只有用 Wi-Fi 傳送資料的時候耗電量較大，若減低傳送資料頻率可以大大減少耗電量。

在半年的測試期裡，都沒有發生突然斷線或需要重置的情況，穩定可靠。

利用這個測量方式，電路簡單且電路板的面積小，一個完整的 PCB (包含微控制器) 大約 230 元，焊接組裝一片 PCB 只需要大約 5-10 分鐘。

C. Technical Spec

Size: 49.2(L)* 48.2(W)*16(H) mm (PCB)

Microcontroller: esp8266

Input voltage: 5V DC

DC current: 80mA (avg)

Sampling frequency: 2Hz

Data Uplink: Every 10 sec

Transfer package size: ~120 bytes

ADC resolution: 10bit

Channels: 1

3. linkit version

A. 電路

電流測量方式：

在電路上，我們使用電流互感器獲得參考電流（800 : 1），然後通過橋式整流器，以將其轉換為 DC，通過電阻獲得電壓值。該電壓值可通過微控制器 ADC 來讀取。Mcp3008 的 ADC 解析度為 10 bits，所以 analog 讀出的值範圍 0~1023 之間，經過計算後，電流的最小解析度是 0.046 安培。

B. 通訊界面：

一塊 linkIt 可以通過 spi 介面與 mcp3008 進行互動，以負責多台洗衣機。將它連接到 Wi-Fi 連接後，通過 HTTP 將數據發送回 server。另外，Protocol 的部分與數據一起發送，因此將來如果需要用別的 protocol 來傳送資料，都可以修改/擴充此 protocol。

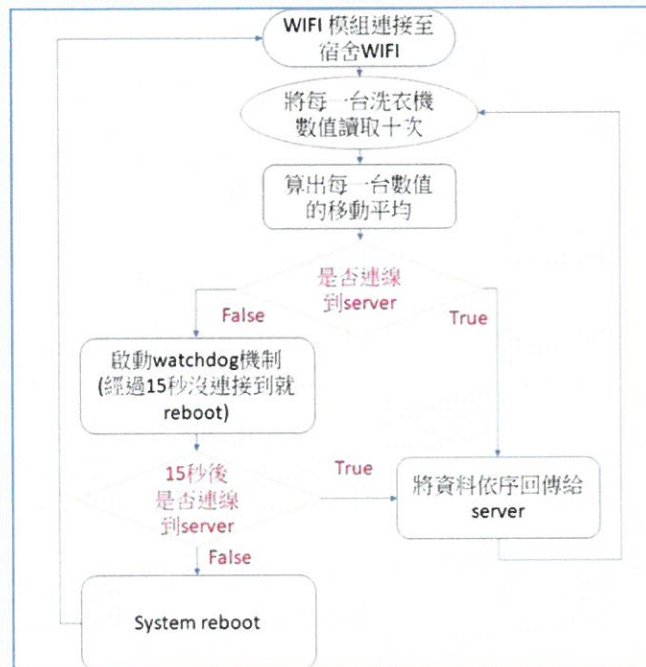
linkit 還有支援其他通訊界面，例如 UART、I2C、SPI 等等，可以透過更新韌體，與其他設備互傳資料。

C. 電源：

在電源上，PCB 版的設計使得可以將多個電源串接在一起，不需用到一個模組一個變壓器，相近的 PCB 可以共用一個 DC 電源，減少插座的需求。

C. 韌體

讀取洗衣機狀態並回傳至 Server 的流程如下圖 先讓 LinkIt 的 wifi 模組連接到宿舍 wifi ,然後因為因為訊號雜訊有點大 ,常常取到很大或很小的值 ,所以這裡取讀到十次值的移動平均 ,接著就連線到 server 並將資料回傳 其中 watchdog 機制是為了避免如果連續連線失敗 ,就讓 linkIt 重新開機。



D. 性能

LINKLT 內部晶片包含低功耗 1T1R 802.11b/g/n 2.4GHz Wi-Fi 子系統，使其電源效率高，主要在傳輸數據(每 7 秒傳一次)時，才會有大功耗。

E. 量產

🕒 加工時間：

一次送洗電路板(20 片)所需時間大概為一個禮拜到兩個禮拜，如果再加上零件採購(郵寄時間)所需要約一個星期的時間，整體的加工時間大概需要兩到三個禮拜。而焊接組裝一片大約需要 5~10 分鐘左右。

	單價	大量(1000 片單價)
LINKLT 7697	460	460
PCB 版	45	30
MCP3008	100	59
電子零組件(包含電阻、電容、橋式整	75	50
總價	655	599

成本估算：

各個零件的單價如下表，而其中 PCB 板可以請製造商批量生產，一個完整的 PCB 總價約為 655 元。

F. Technical spec

microcontroller: MediaTek MT7697

input voltage: 5V DC

DC current: ~50mA

Sampling frequency: 10Hz

Data Uplink: Every 7 sec

transfer package size: ~150 Byte

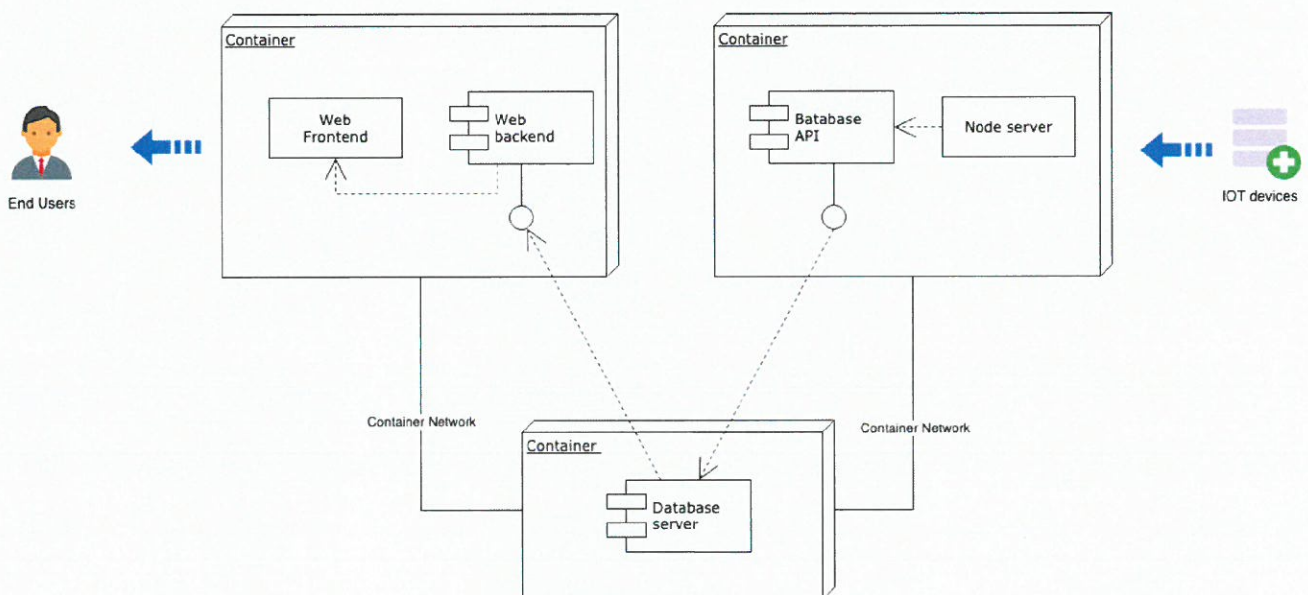
resolution: 10bit

channels: up to 6

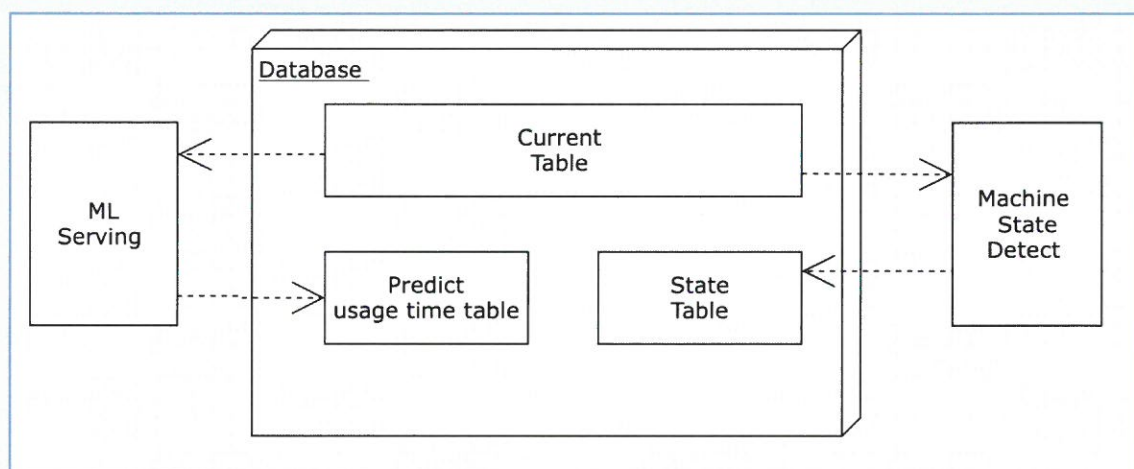
四、軟體

軟體為智慧洗衣機系統的核心角色，需接收來自硬體 IOT 設備的資料、進行資料處理分析、資料儲存、提供使用者介面。為了應付這些不同的需求，在我們自己架設的伺服器中，我們利用 Container base 的架構，使用 micro service 去模組化串接各種功能及需求。

1. 架構



在硬體端 Node server 的部分，我們利用 http server 去監聽 2580 port 上面由 IOT 設備傳來 http get 格式的資料，在將其寫入 mySQL 資料庫的 table 中，並再由一隻專用的 python 程式去把這些電流資料分析成洗機的狀態以及剩餘時間，將其寫入另一個 table 中。



在使用者這端，Linebot 以及 Web 的後端從 MySQL 資料庫中取得使用狀態和剩餘時間的資料，透過使用者的請求將其呈現給使用者。

2. 方法

在 Server 的硬體需求部分，因為因為需要 Container base 以及部分 VM 來進行備份等需求，因此需要選擇 CPU 核心數數量較多的以及支援比較多的儲存所以最終選擇以下組合：

硬體	型號	備註
CPU	R7-3700x	16 個 cores 可以滿足性能需求
Mother-Board	ASUS x370 ATX	選擇 asus x370 最大的原因是因為提供了八個 SATA 接口可以讓我們有足夠的硬碟做 Raid 保護資料
RAM	32G DDR-4	軟體 Raid 需要利用大量的 RAM 作為
Disk	4tb HDD *2 , 500G SSD	HDD 作為備份，SSD 用作系統以及服務
Power	台達 550w	

在作業系統上面選擇了開源的 Proxmox VE 同時具備了 VM 以及 Linux Conatiner 等功能可以彈性調度服務。

五、資料處理

蒐集留資料後，需要透過資料處理將資料轉換成洗衣機狀態，方能提供給使用者應用。我們有兩種方案，分別是 1. 邏輯判斷 2. 機器學習，兩者都是從資料庫中取出近期的電流資料，運算判斷後再轉存於洗衣狀態資料庫。機器學習的方案我們除了輸出使用狀態，也預測剩餘時間。

1. 邏輯判斷

 簡介：

邏輯判斷是透過觀察洗衣機形成的電流歷程，對此設計對應的邏輯，以此判斷洗衣的狀態。

 解決方案：

觀察過電流與洗衣機狀態的關係後，我們發現可以分為兩種狀況來判斷洗衣機的狀態。

第一種是，洗衣機在運轉時，其電流會大於非運轉時的電流，於是我們就可以透過設定一個閾值，來判斷洗衣機的狀態。像是 2、3 號洗衣機，在運轉時，其電流最大是一兩百，最小是九，而非運轉時，其電流多為零，有時候會出現二或三的數值。因此，我們可以將閾值設為四到八之間。當電流大於閾值的時候，判斷洗衣機為運轉時期，當電流小於閾值時，判斷為非運轉時期。

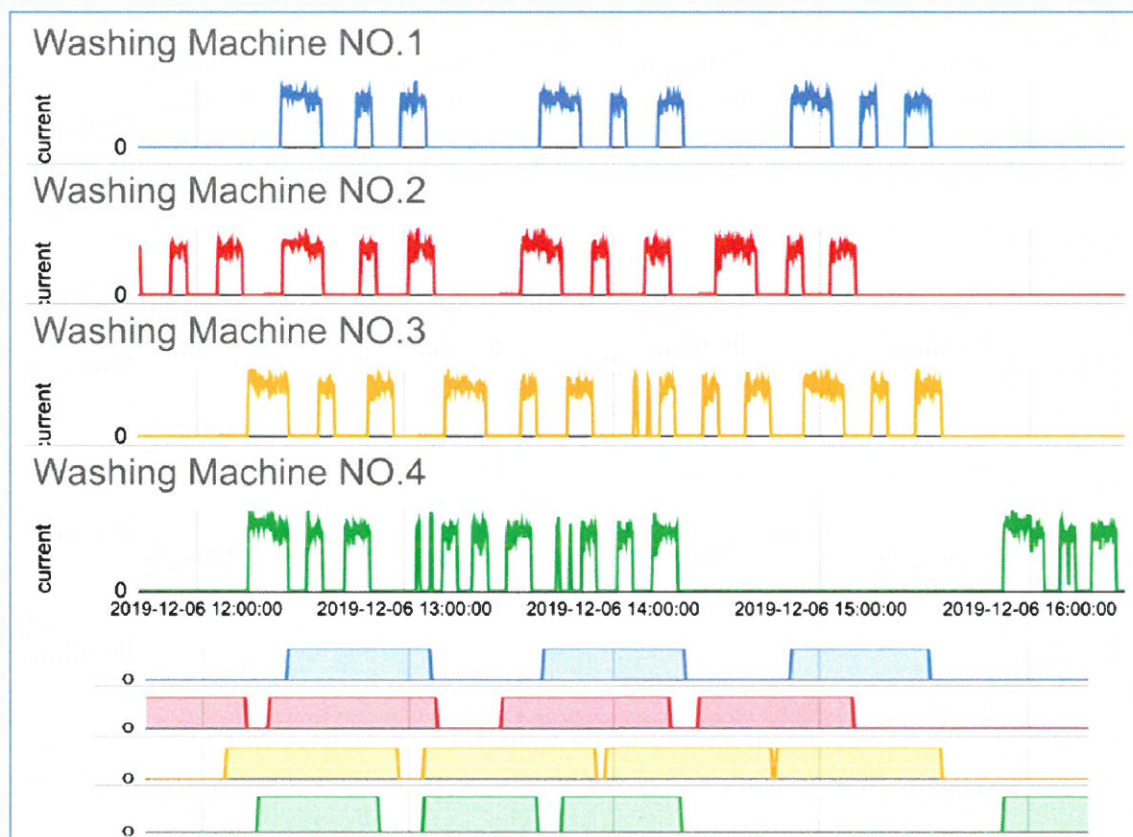
第二種相對複雜，因為其運轉過程中，其電流可能過低，導致我們無法用簡單的閾值來判斷。以 1、4 號機為例，他們在運轉時，會出現兩次的低電流期，其電流為零，和非運轉時期的電流相等。因此，以第一種方法判斷時，我們設閾值為二。則在運轉過程的低電流期，電流低於閾值，程式判斷為非運轉，就會出現誤判。此外，我們觀察到，洗衣機剛運轉時，1、4 號機的電流相對不穩，從下面的洗衣機電流圖可以看到，每一次洗衣機運轉的電流波型圖都沒有固定的模式。

為了排除這些困難，我們加入「時間」來協助判斷的完成。我們的作法如下，記錄洗衣機開始運轉的時間，在開始運轉後的固定時間內，例如 X 分鐘，我們都判斷洗衣機正在運轉，洗衣機運轉的 X 分鐘後，我們就觀察電流與閾值的關係，當電流大於閾值的時候，判斷洗衣機為運轉時期，當電流小於閾值時，判斷為非運轉時期。我們設計讓洗衣機運轉後的 X 分鐘，落在洗衣機運轉週期的最後幾分鐘，讓閾值判斷逃過低電流期，如此一來就可以避免誤判的情形發生。

以 1 號機為例，當電流大於閾值時且洗衣機先前狀態為非運轉時，我們判斷洗衣機開始運轉，並且記錄開始運轉的時間。我們設計在 36 分鐘內，都判斷洗衣機為運轉期。36 分鐘後，洗衣機不再出現低電流期，我們就用閾值判斷，設閾值為六，當電流小於六時，判斷為非運轉時期，代表洗衣機已經完成一個週期。

● 成效：

上面的折線圖為洗衣機 1 號到 4 號的電流對時間關係圖，縱軸為電流，橫軸為時間，下半部則為相對的洗衣機狀態圖，縱軸為判斷結果，橫軸為時間。藍色為 1 號機、紅色為 2 號機、黃色為 3 號機、綠色為 4 號機。由此圖我們可以看到，在每個洗衣機週期內，上述的邏輯判斷可以成功判別洗衣機狀態。



2. 機器學習



簡介：

機器模型我們採用常見的 python+TensorFlow+keras 模組，機器學習的模型採用 LSTM (Long Short-Term Memory network)，LSTM 屬於 RNN (Recurrent Neural Networks)的一種，適合時間序列的相關資料。模型訓練 google colab 雲端運算資源，進行模型訓練，訓練完成的模型在伺服器中，定期取資料庫數據，運算使用狀態與剩餘時間。



解決方案：

模型的訓練我們使用最近一周的所有資料，這些資料會使用 matlab，進行上資料標籤(labeling)的動作，上標籤的過程為半自動化，使用軟體判斷再以人工確認。最後完成標籤的所有資料以 80%、20%的比例分為訓練資料(training data)、測試資料(test data)進行訓練。

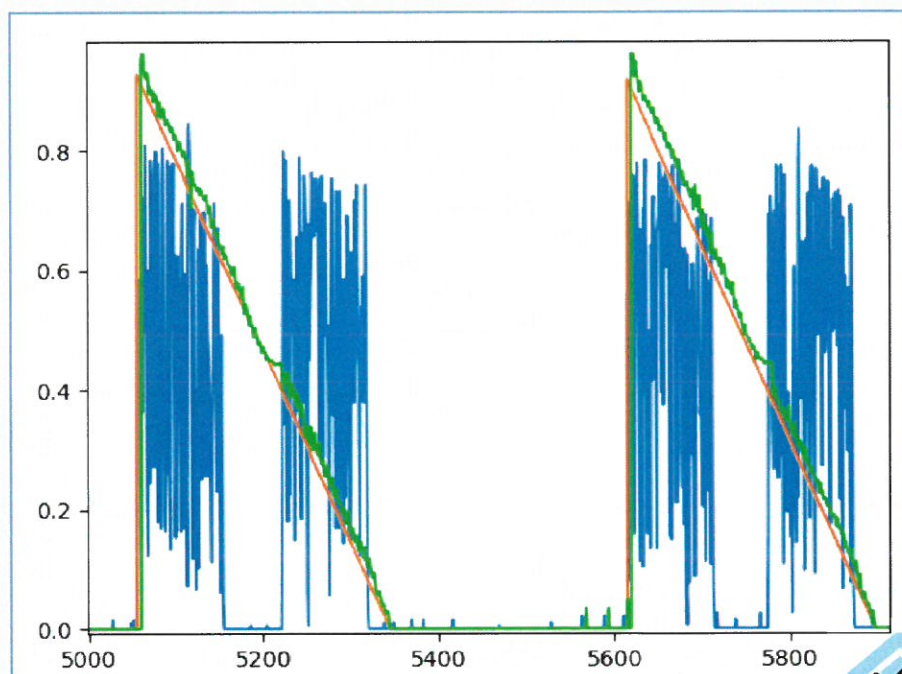
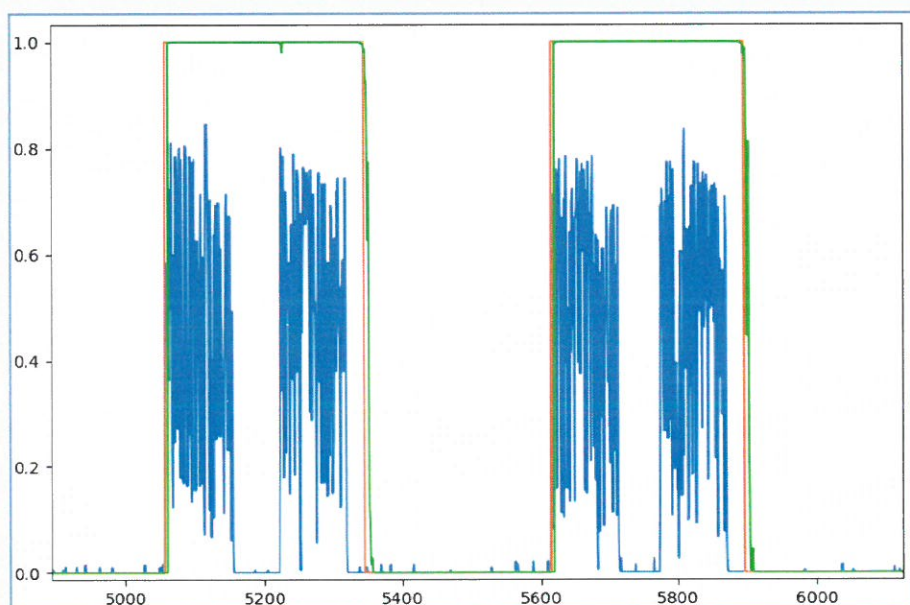
每次洗衣行程約 30 分鐘，因此模型的輸入會使用過去 30 分鐘內的資料，並依預先設定的權重進行資料篩選，減少輸入資料量來減少模型大小，最終會有 100 筆散布於 30 分鐘區間的電流資料作為模型的輸入。模型共有 4 層(layer)，第一層為 100 筆電流數據輸入，進入第一層 10 個節點的 LSTM，再進入第二層 4 個節點的 LSTM，最後第四層為 dense 層，依據輸出類別有兩種 dense 層的啟動函式，輸出洗衣狀態使用 sigmoid，剩餘時間使用 relu，主要差異為：洗衣狀態為是/否的邏輯，剩餘時間是一個數字。每次訓練會經過 100 個 epoch，共耗時約一個半小時。



成效：

下方的圖分別是洗衣狀態與剩餘時間機器學習成果，我們使用一周的資料進行訓練，並額外使用五筆數據做為驗證。兩張圖中，藍色的線段是來自洗衣機的電流資料，橘色為真實結果，綠色為機器學習利用電流資料所預測的結果，兩張圖的橫軸皆為時間，縱軸為分別使用狀態(0/1)，與剩餘時間(最大約 30 分鐘)。

從第一張圖我們可以發現，機器學習能夠使用電流資料準確預測洗衣的開始與結束。第二章圖中，雖然預測的時間不全然準確，也會有剩餘時間的不規則跳動，但對於洗衣機使用者的使用情形來說，已經足以提供使用者一個對於剩餘時間大致的瞭解。



六、使用者介面

洗衣機電流資料透過軟體轉換為使用狀態後，需要透過使用者介面傳遞給大眾使用，經過下表所列的評估後，我們決定將藉由 linebot、網頁兩種媒介提供給使用者。

	優點	缺點
聊天機器人	人人手機都有 Line，不必另外下載 聊天機器人使用方便 團隊有開發經驗 跨平台	增加私人通訊軟體的複雜度
網頁	一目了然 直接提供資訊 跨平台	無法針對特別需求
APP	功能可自行設計 設計更複雜的功能	開發耗時，使用者需額外下載
宿舍電子佈告欄	直覺查看	只有特定地方能取得資訊

1. Linebot

需求：

- 擁有硬體傳來的資訊後，透過使用者介面傳遞
- 使用者不須付費
- 須讓使用者隨時隨地能方便取得資訊

解決方案：

評估後，我們使用 Django 作為處理 linebot 訊息的後端伺服器，於 Django 專案中 使用不同的 APP function 模組化處理不同的功能，分別是蒐集 db 資料、處理來自 Line Server 的請求、回覆對應的訊息。我們採用 line-bot-sdk 套件，收送 Line Server 的 webhook，利用這個 SDK 中的 API，我們可以提供圖形化介面傳送洗衣機使用訊息給用戶。

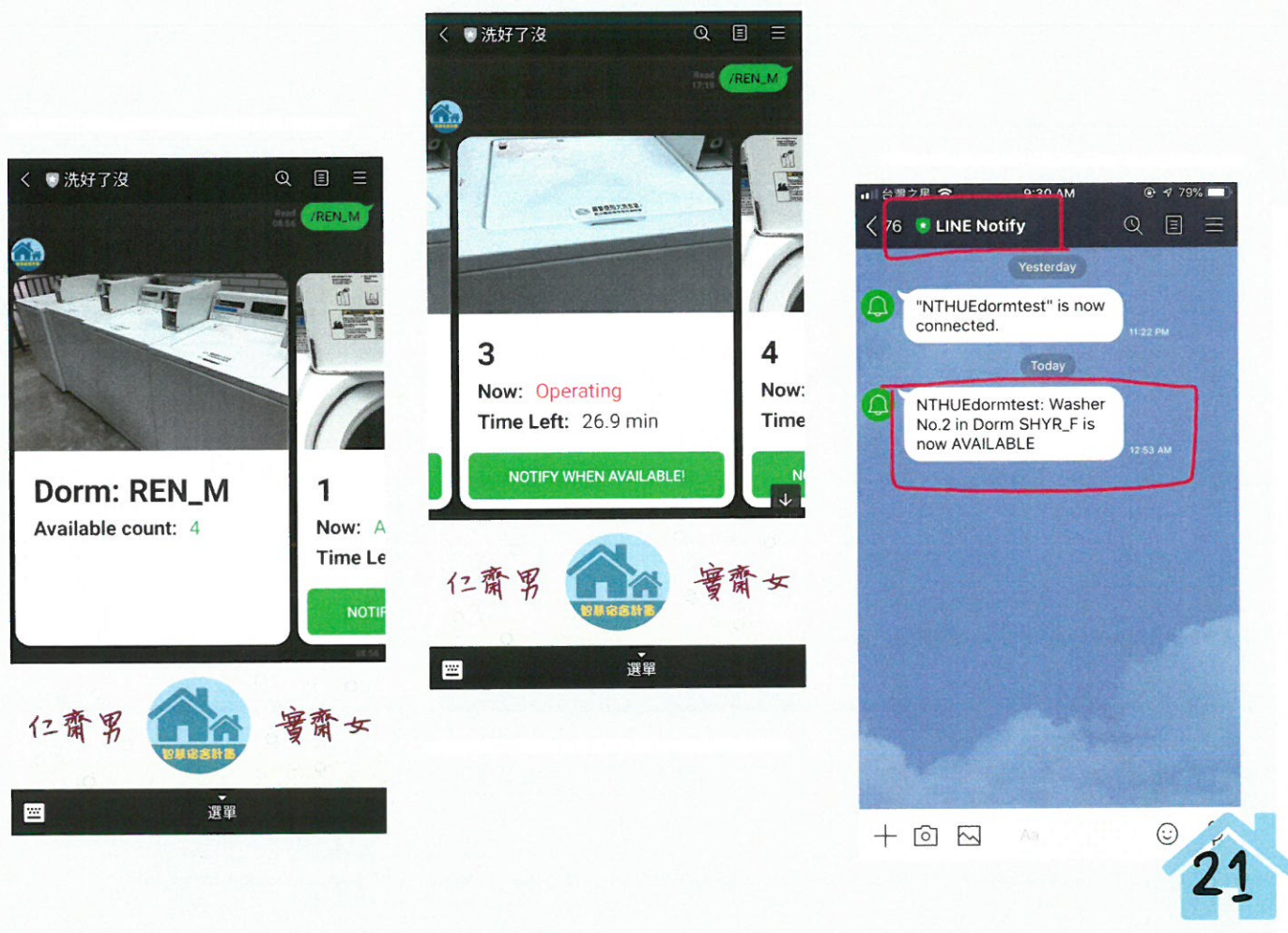
通知功能：

除了可以查詢使用狀態以及剩餘時間，於洗衣完成後，我們可以利用 line notify 的功能，提醒使用者收衣服或告知提它使用者有空的洗衣機可以使用。

此功能同樣架設於 Django 專案中，透過 line notify 的邀請，取得使用者的憑證，這些使用者 ID 與憑證將儲存於資料庫中。Cron jobs 的排程會每 5 分鐘定期更新洗衣機狀態，並從資料庫中取出開啟通知功能的的使用者資料，向使用者發送通知。

成果：

我們的 linebot 名為「洗好了沒」。加入好友後，使用者只要點選下方的選單，即可收到對應洗衣場各台洗衣機的使用狀態，使用中的洗衣機會顯示剩餘的時間，並可以使用通知功能，於洗衣完成後收到來自 linebot 的通知。

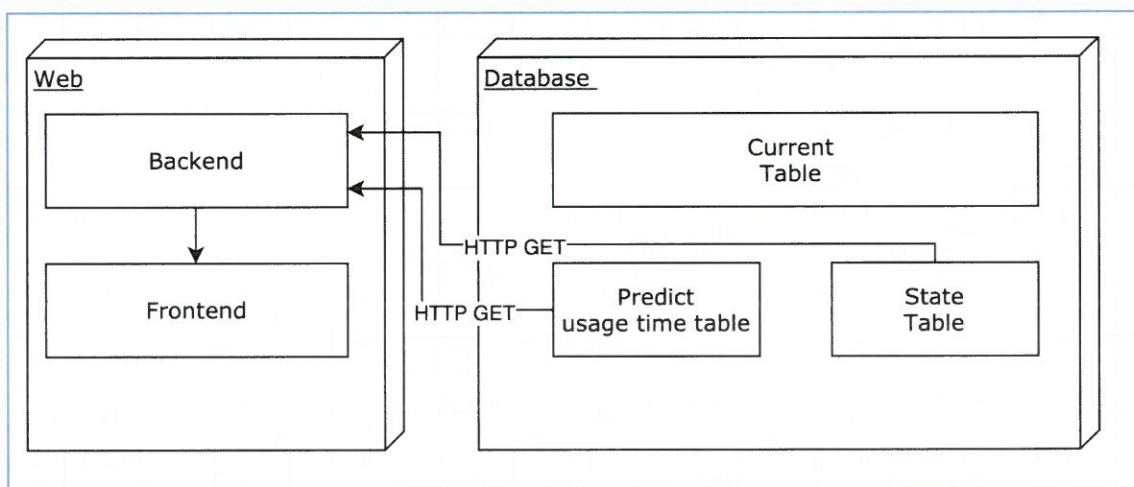


2. 網頁

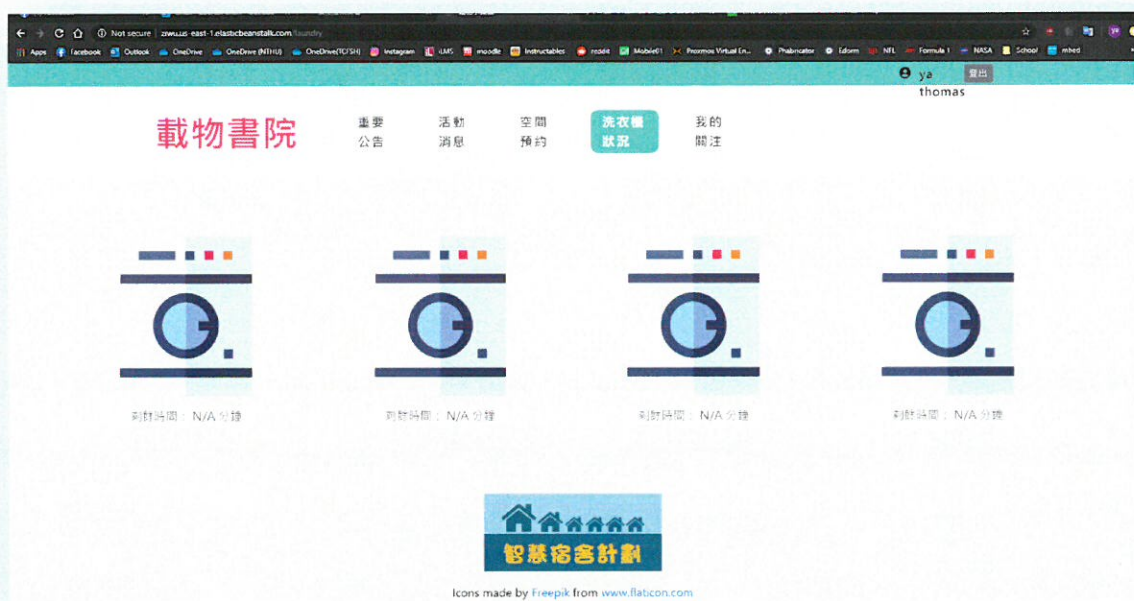
除了設計 chatbot 作為使用者介面，我們也有設計軟體 API，提供給有需求的其他人使用，API 使用 HTTP get request，向伺服器請求資料，使用狀態以及剩餘時間。

智慧洗衣機的網頁部分，經過評估我們認為不適合單獨架設一個只有查詢洗衣狀態的網頁，為了查詢洗衣狀態而開啟一個專屬網頁有些麻煩，觸及率不高造成成效不足。因此，我們與仁齋團隊合作，使用 API 將洗衣機資訊顯示於仁齋佈告欄的社群交流網站，藉此提高能見度。

洗衣狀態 API 架構



洗衣狀態網頁



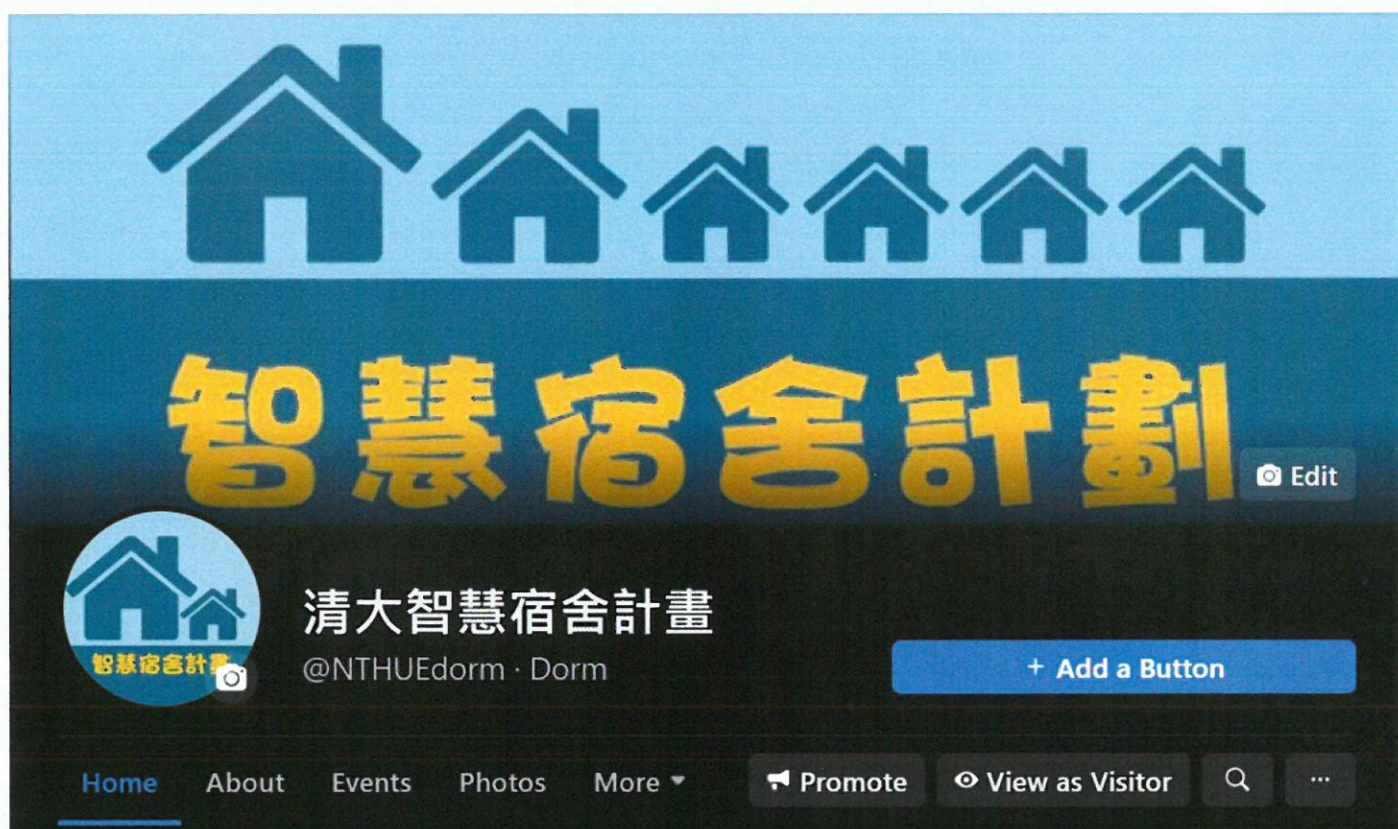
七、宣傳、推廣

1. 粉專

介紹文：

智慧洗衣機專案服務資訊主要是透過 Facebook 的粉絲專頁功能來推廣。粉絲專頁「清大智慧宿舍計畫」在 2019 年初建立，目前內容也以推廣智慧洗衣機服務佔最大比例，其餘則是宣傳計畫 logo，以及人才招聘等等。

考量截至目前為止計畫無營利，且使用到設備的對象僅限於居住在仁、實齋的學生，因此粉絲專業沒有透過買廣告等方式加強宣傳，另外從內容經營層面來看，貼文的發布頻率也不固定，僅在有必要消息時才發文。綜合以上種種因素，粉絲專頁截至 2020 年 8 月追蹤人數僅 75 人，客觀而言粉絲專頁本身並不具有太大的宣傳功用。





成效：

為了彌補粉專本身宣傳功能不足，在發布重大消息時，我們會使用私人的 Facebook 帳號，以個人名義將貼文分享至仁、實齋宿舍社群，透過宿舍社群本身大量的人數與頻繁的點閱，爭取觸及率。該方法成效卓越，三篇被分享的貼文都能藉此突破千人的觸及率。

	第一次	第二次	第三次
發布日期	2020/3/15	2020/5/21	2020/6/11
內容大意	服務使用方式	技術介紹及徵才	智慧洗衣宣傳
分享次數	4	3	2
點讚次數	11	10	5
留言次數	0	0	4
觸及人數	2164	1890	1093

2.洗衣場布置

考慮到洗衣機上已經有廠商布置的其他宣傳，因此我們選擇將所有的服務內容集結在 a4 大小的紙張，影印並護貝，張貼在宿舍洗衣場顯而易見的地方，有意願使用服務的同学即可掃描紙上的 QR code，連結到專案設置的 linebot，立即使用功能。



x



【洗好了沒】 @nthu_ewasher

伍、成員心得

陳佐任：

這是我第一次帶領如此大的一個專案，很榮幸能有這麼多來自不同科系的夥伴們加入，共同為這一個智慧洗衣機專案努力。這次的成果規模雖然不足以造成一個多大的衝擊，但是這個專案的發展，涵蓋了軟體、韌體、硬體；網頁前端、網頁後端、伺服器管理；擬定需求、規劃、原型驗證、測試、公測、宣傳、體驗回饋。我們順利地將一個概念，從無到有，完成一項產品的完整的開發歷程，這展現了清大大學生是有能力學以致用，獨立執行一個系統的開發專案。

領導這個專案的過程中，我採用較輕鬆的態度，以不影響課業的前提安排大家進度，並依每個人的專長、興趣進行分工，整個開發過程，大家非常積極認真看待每一個歷程，以開發量產商業產品的態度來看待這個專案，很高興最終能順利完成，提供宿舍這樣的一套智慧洗衣機系統。

蔡闊光：

在一開始的時候我們就知道這個計畫必須做出一個解決方案為目標，而提出一個解決方案的想法跟發明一個新的東西有很多不一樣之處，其中最主要的是在一個解決方案內，我們會去大量使用到一些已經成熟的技術並且把這些東西組織起來變成一個產品，而不是從頭發明很多東西。這是和我一開始所想的很不一樣，尤其是在我負責的系統這個部分，絕大部分得時間是花在從這些已有的技術中選擇適合我們開發的選項，並且想辦法整合這些東西，其中也必須不段的去和 設計電路板以及設計網站的同學溝通，訂出一個通用的資料傳輸及儲存格式。

倫皓正：

在參與智慧宿舍計劃以前，我就有自己用 Arduino 和 ESP8266 動手做了一些 DIY 的東西，可是參與之後才發現這種要量產的電路有更多需要考量的東西，比如耗電量、向前和向後兼容性(forwards and backwards compatibility)、安全性、穩定性等等。以洗衣機的部分來說，多個電路串聯起來除了可以省下插座數量的需求還可以減低耗電量；如果微控制器與 Wi-Fi 的連線斷開了，它會自動重啟並嘗試重新連線，這些都是以前沒有考慮過的東西。

另外在電路板的設計這個部分我也是學到很多。我學到了如何減少電路板的大小，如把比較高的電子元件(如微控制器)疊在比較矮的元件之上(如電阻等無源元件)。另外，在電路板布線的時候的種種考量都是寶貴的知識。最後除了技術上的部分，我也學習到如何與其他人合作，雖然我是負責軟體的部分，在大家的合作及互相幫忙之下成功做出成果，收穫很多。

陳柏霖：

一開始加入這個企劃以為只是玩玩 LinkIt 就好，加上之前就有碰過一些軟體相關的小東西像是 arduino 的藍芽通訊，所以應該是簡單的，但後來才發現事情沒有想像中容易，還是有許多例如連接 wifi、繪製 PCB、如何整理收集到的資料等等都是需要多加點心思去想怎麼才能不會出 bug，如果有又要怎麼解決，不像是以前自己摸索時只要確定有連接就結束那麼單純。

也因為如此，在這個專案學到很多，第一就是更能了解怎麼把資料有順序的傳回給 server，有時候不是因為邏輯不對造成 server 收到太多資料而爆掉，就是沒了解 function 的用法而根本沒連到 server 等等。第二就是在繪製 PCB 板時因為經驗不夠，常常畫錯把不同的地接在一起，或是有些電路應該連在一起卻沒有，幸好有專家在旁指導才沒導致太大的差錯，但這也讓養成更仔細的去檢查每條線是否畫對的習慣，此外就是學到怎麼跟別人互相合作，因為這是第一次參加這種大型的企劃，有許多事情是第一次做，常常不會就停滯好幾天，但想到只要做好就可以讓整個宿舍的學生都使用就很有動力，而且旁邊都會有大師指導，讓我能夠放膽嘗試我想做的事。

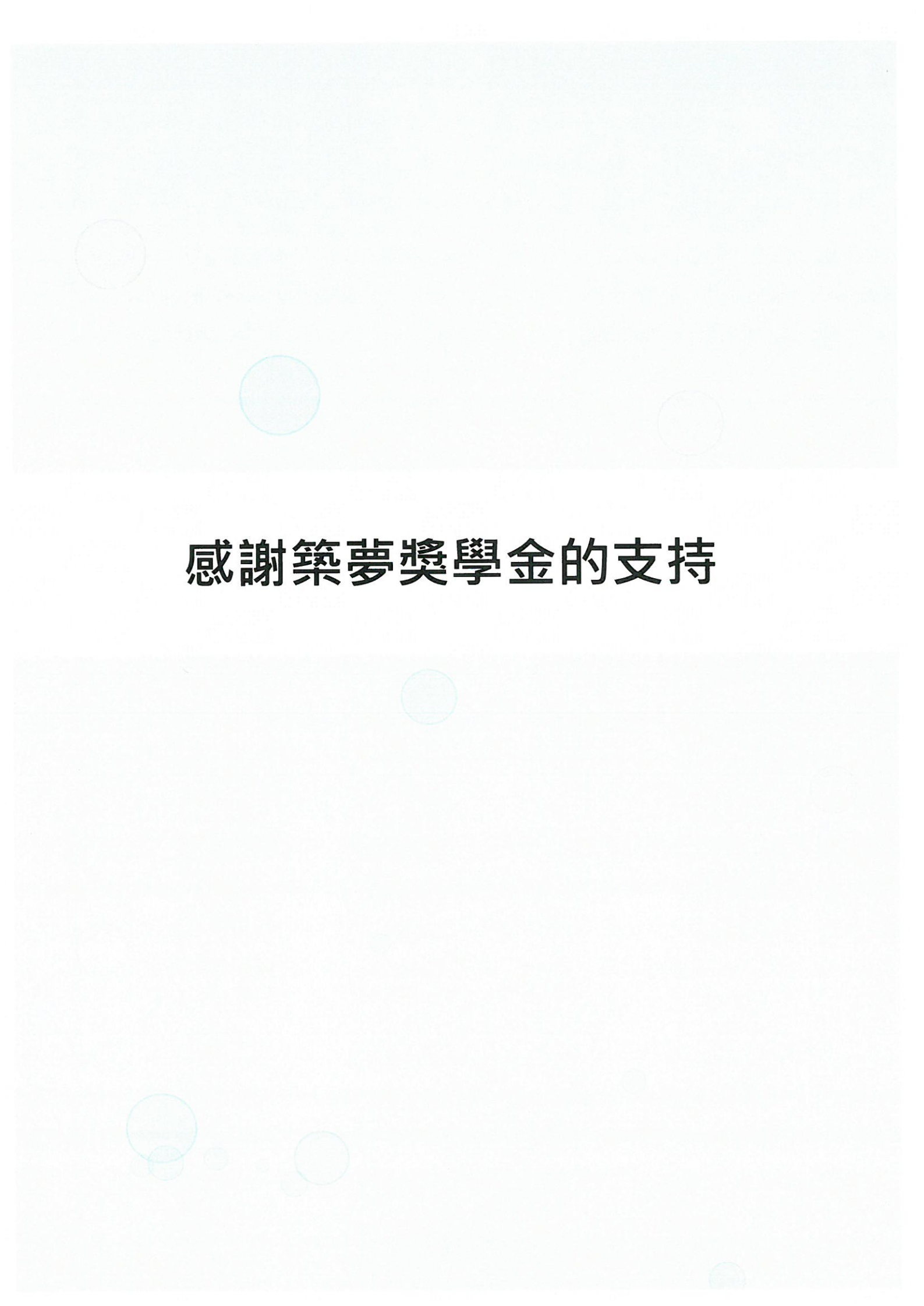
阮棠：

原本沒想過自己可以參與這個計畫。在這個計畫中我主要負責用 Line 提供的 API 串接我們的服務，讓使用者可以透過 Line 的聊天機器人得知洗衣機的使用情形。成功把聊天機器人做出來的時候我覺得很感動，因為驅使我做這個專案的動力就是因為我自己是仁齋四樓的居民，每次洗衣服都要走到一樓，有了這個服務之後我就可以先看好是不是有可供使用的洗衣機，以免白跑一趟。進行這個專案的過程中我學到如何架設伺服器(NGINX、PVE、Linux)、寫後端(Django)、認識資料庫(MySQL)、使用 Python 串 API 和處理聊天機器人邏輯，同時也學會使用 Git 進行版本控制。除了技術層面，也學習在團隊中與他人合作的方法。過程中常常覺得很多地方做不好，重頭做起，對於使用者體驗也還有很多可以改進的地方，相信未來還有很大的進步空間。感謝智慧宿舍計畫的大家提供我很多協助，無論是技術或調適課業跟專案，都給予我很多非常有幫助的建議。

李家好：

當初參加這個計畫是因為對於 Arduino 很感興趣，想要好好的接觸這個對我來說全新的領域。加入這個計畫後，大部分的時間都是在基礎的部分打滾，如何安裝、如何寫 python、如何找到需求對應的電子零件、如何在材料行中分別各個電子零件、如何焊接、如何配置電路等等等，資料種類繁雜、所接觸領域廣泛，讓對於面對這個全新領域的我有點吃不消，但是由於計畫負責人、組員們都非常熱心的解決所有的疑惑，加熱情，讓我漸漸熟悉這個領域。

在洗衣機這個專案中學到許多東西，學到 Arduino 的控制、電路板的焊接等等，讓我對於一個專案的從無到有、組織架構、執行時程與推進都有極大的了解，也在最後的階段幫忙安裝、檢查設備，雖然我對於這個專案的貢獻可以說是微乎其微，只是幫忙跑腿與勞力，但過程中的收穫讓我有自信在進行下一個計畫的時候，我可以將我的所學好好的發揮出來、能夠參與更核心的設計。

The background of the page is white with several light blue circles of varying sizes scattered across it. Some circles are solid, while others are faint outlines. They are positioned in the top left, top center, top right, middle left, middle right, and bottom left areas.

感謝築夢獎學金的支持