

組別：E6

系統名稱：e-Dorm

指導老師：鄭啟斌 老師

成員：

姓名	學號	修課班級	座號
張鳳軒	407630267	資管 4B	12
林星卉	407630358	資管 4B	15
翁紹鈞	407630564	資管 4B	21
廖崇皓	407630689	資管 4B	25
黃柔綺	407630861	資管 4B	31
簡儀丞	408637147	資管 4B	57

110 學年度

110 學年度

組別：E6

系統名稱：e-Dorm

指導教授：鄭啟斌 老師

# 目錄

一、	前言.....	1
(一)	動機.....	1
(二)	目的.....	1
二、	系統分析.....	2
(一)	系統目標.....	2
(二)	系統使用對象.....	2
(三)	系統功能簡介.....	2
(四)	系統使用環境.....	3
(五)	系統開發工具.....	3
(六)	系統技術需求.....	3
三、	物聯網裝置.....	4
(一)	物聯網裝置結構示意圖.....	4
(二)	使用模組與電子零件介紹.....	4
1.	Arduino Uno 開發板 .....	4
2.	ESP8266 WIFI 模組 .....	5
3.	繼電器.....	6
4.	WCS1800 霍爾感測器模組.....	6
(三)	功能說明.....	7
四、	系統設計.....	9
(一)	系統特色.....	9
(二)	系統架構圖.....	9

1. APP 架構圖 .....	9
2. 使用者案例圖 .....	10
3. 循序圖 .....	11
4. 部屬圖 .....	12
5. 後臺管理架構圖 .....	13
6. 系統運行架構圖 .....	22
(三) 機器學習 .....	24
1. 資料蒐集 .....	25
2. 丟入參數 Example .....	26
3. 資料處理 .....	27
4. 建立模型 .....	29
5. 模型評估 .....	30
6. 儲存模型 .....	31
(四) 資料庫設計 .....	32
1. Firestore Database .....	32
2. Realtime Database .....	34
五、系統功能說明 .....	36
(一) 住宿生端 .....	37
1. 垃圾車提醒 .....	38
2. 我要洗衣 .....	39
(二) 管理員端 .....	47
1. 報修通知 .....	47
2. 包裹提醒 .....	50
3. 包裹簽收紀錄 .....	51

六、	結語.....	52
七、	未來展望.....	52
	（一） 偵測其他宿舍設備.....	52
	（二） 新增預約制度.....	52
	（三） 與自助洗衣店合作.....	53
八、	參考資料.....	53

## 圖目錄

▲圖 1	物聯網裝置結構示意圖.....	4
▲圖 2	Arduino Uno 開發板 .....	5
▲圖 3	ESP8266 WIFI 模組 .....	5
▲圖 4	繼電器.....	6
▲圖 5	WCS1800 霍爾感測器模組.....	7
▲圖 6	成品展示 1.....	8
▲圖 7	產品展示 2.....	8
▲圖 8	APP 架構圖 .....	9
▲圖 9	使用者案例圖-住宿生 .....	10
▲圖 10	使用者案例圖-管理員 .....	10
▲圖 11	循序圖.....	11
▲圖 12	部屬圖.....	12
▲圖 13	後臺管理架構圖.....	13
▲圖 14	APP Request .....	13
▲圖 15	管理員-發送包裹提醒 .....	14
▲圖 16	管理員-查詢學生包裹 .....	15
▲圖 17	住宿生-簽收包裹 .....	16
▲圖 18	住宿生-查詢機台現狀 .....	17
▲圖 19	住宿生-查詢個時段機台使用率 .....	18
▲圖 20	住宿生-預估洗衣完成時間 .....	19
▲圖 21	管理員-審核報修回報 .....	20
▲圖 22	住宿生-提交報修申請 .....	21
▲圖 23	系統運行架構圖.....	22

▲圖 24	模型訓練、預測過程.....	24
▲圖 25	丟入參數.....	26
▲圖 26	資料處理 1.....	27
▲圖 27	資料處理 2.....	28
▲圖 28	資料處理 3.....	28
▲圖 29	建立模型.....	29
▲圖 30	模型評估 1.....	30
▲圖 31	模型評估 2.....	30
▲圖 32	模型評估 3.....	31
▲圖 33	模型對應時間.....	31
▲圖 34	儲存模型.....	31
▲圖 35	Firestore Database.....	32
▲圖 36	管理員資料集.....	33
▲圖 37	床號資料集.....	33
▲圖 38	學生資料集.....	34
▲圖 39	Realtime Database .....	34
▲圖 40	機器資料表.....	35
▲圖 41	主頁.....	36
▲圖 42	宿舍管理頁.....	36
▲圖 43	住宿生頁.....	37
▲圖 44	住宿生-我要洗衣頁 .....	37
▲圖 45	垃圾車提醒提示窗頁.....	38
▲圖 46	垃圾車提醒頁.....	39
▲圖 47	垃圾車提醒.....	39

▲圖 48	洗衣機頁 .....	40
▲圖 49	2F-6F 洗衣機頁 .....	40
▲圖 50	1F 洗衣機頁 .....	41
▲圖 51	洗衣提醒視窗 .....	41
▲圖 52	洗衣提醒 .....	42
▲圖 53	洗衣機配置圖頁 .....	42
▲圖 54	使用率頁 .....	43
▲圖 55	洗衣機報修頁 .....	44
▲圖 56	待領包裹頁 .....	45
▲圖 57	包裹提醒頁 .....	45
▲圖 58	包裹簽收確認視窗 .....	46
▲圖 59	管理員頁 .....	47
▲圖 60	報修通知頁-審核中 .....	48
▲圖 61	狀況說明頁 .....	48
▲圖 62	報修確認視窗頁 .....	49
▲圖 63	報修通知頁-處理中 .....	49
▲圖 64	包裹提醒頁 .....	50
▲圖 65	包裹簽收紀錄頁 .....	51



# 一、前言

## (一) 動機

在學生宿舍中，校方往往提供了各式各樣的設備與服務來滿足我們生活上大大小小的需求，舉凡洗衣設備、冰箱、代收包裹，也有固定可以倒垃圾的時間，遇到設備故障也能主動向住輔組告知。雖然學校已提供充分的設施和一套完整的系統流程，但在入住期間不免還是會遇到一些困擾，像是：

1.洗衣機幾乎集中於一處，如果走到了卻沒機台可以使用，可能只能帶著一籃衣服在洗衣場排隊，或者是回到自己房間，等過了一會兒再去看看是否有釋出的空機。

2.包裹提醒是以紙本方式傳遞，容易遭到忽略。

3.垃圾車抵達時是不會有任何音樂聲作為提醒的，如果不小心忘記就只能等到隔天。

針對以上難以利用實質制度進行改善的部分，我們團隊想要透過科技的力量，將解決方法整合至一系統中，讓學生們不再是只能被動地接受日常生活中的不便，而是能透過自己的雙手，主動獲取自己需要的資訊，進而改善生活。

## (二) 目的

針對以上住宿生與管理員會遇到的問題，我們列出了幾個解決方案：

1.住宿生能透過我們的 APP 可即時查看洗衣機的使用狀況並能就近使用，讓每台洗衣機發揮到最大化。

2.包裹領取改由無紙化流程，包裹經由管理員發送後住宿生能直接接收到包裹提醒並簽收，管理員也能掌握包裹去向。

3.發送垃圾車提醒，住宿生依照自身需求設定垃圾車來臨前多久要提醒自己。

4.為提升實際性，我們選擇與本校現有的淡江 i 生活產學合作來擴大使用率，讓此 APP 發揮到最大效用。

## 二、系統分析

### (一) 系統目標

- 1.自動偵測洗衣機狀態、預測其使用完畢之時間以及提供各時段使用率之統計供住宿生查看。
- 2.提供住宿生洗衣機之報修系統，並且管理員收到通知後也可即時讓其他住宿生知道該機台故障。
- 3.使管理員和住宿生可在系統上管理和查看包裹紀錄。
- 4.提供住宿生手動設置倒垃圾時間之提醒功能。

### (二) 系統使用對象

淡江大學之住宿生和管理員，亦可推廣至各大宿舍環境。

### (三) 系統功能簡介

我們的系統功能主要可以分為以下三個面向：

- 1.洗衣機狀態：以 Arduino 平台開發的智慧插座，能偵測洗衣機狀態後回傳至平台端後，在使用者介面上顯示其是否正在使用中及洗衣預計完成時間，以供住宿生查看，並能同時在使用中機台設定洗衣提醒來提醒自己要記得拿衣服了。此外還有個時段洗衣機使用率圖表，讓住宿生可以避開每天的洗衣尖峰時段，省下更多時間。若住宿生發現洗衣機有故障情況，也可以利用系統通報管理員，當管理員確認該機台故障後，其權限可以手動將該故障機台的狀態設為「故障」，以免學生誤用。

- 2.包裹系統：管理員將住宿生包裹登記在系統上後，住宿生不但會收到手機提醒外，也可以在系統內看到自己是否有尚未領取之包裹；並且改以電子方式簽收，而包裹紀錄仍會存於資料庫中，以使管理員仍能調閱。
- 3.垃圾車提醒系統：住宿生可自行設定是否需要提醒，以及需要在多久之前提醒自己。

#### **(四) 系統使用環境**

- 1.行動裝置作業系統 (Android 8.1)
- 2.伺服器作業系統 (Windows Server 2012)

#### **(五) 系統開發工具**

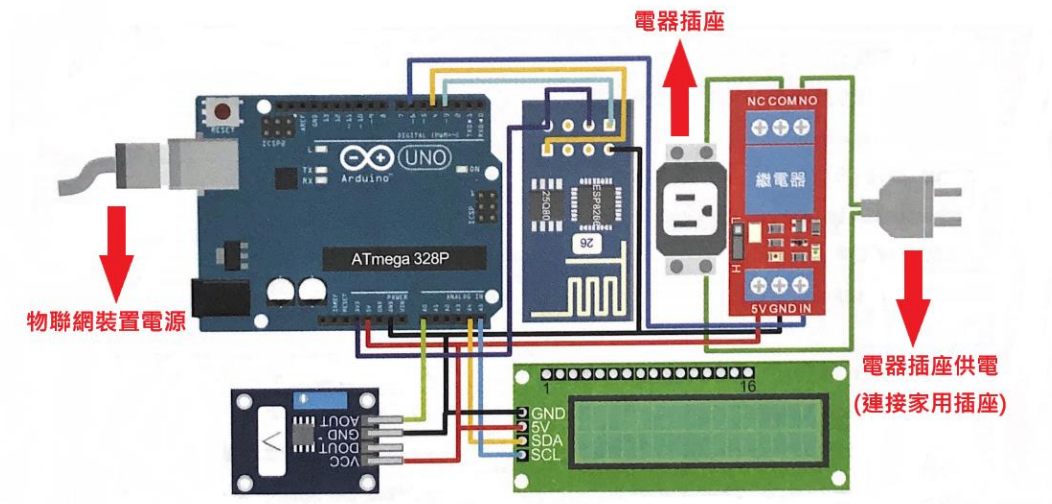
- 1.行動裝置：Android Studio
- 2.物聯網：Arduino
- 3.後端程式語言：Python、Tensorflow、Keras
- 4.UI 設計：Adobe Illustrator 2020

#### **(六) 系統技術需求**

Kotlin、Arduino(C++)、Python、NoSQL、Adobe Illustrator

### 三、物聯網裝置

#### (一) 物聯網裝置結構示意圖



▲圖 1 物聯網裝置結構示意圖

(修改自 Arduino 物聯網最佳入門應用：打造智慧家庭輕鬆學 7-37 頁)

#### (二) 使用模組與電子零件介紹

##### 1. Arduino Uno 開發板

Arduino Uno (圖 2) 是基於 Microchip ATmega328P 微控制器並由 Arduino.cc 開發的開源微控制器板。該板配備了可與各種擴展板和其他電路接口的數字和模擬輸入/輸出引腳組，經常利用於各種 IoT 上。



▲圖 2 Arduino Uno 開發板

## 2. ESP8266 WIFI 模組

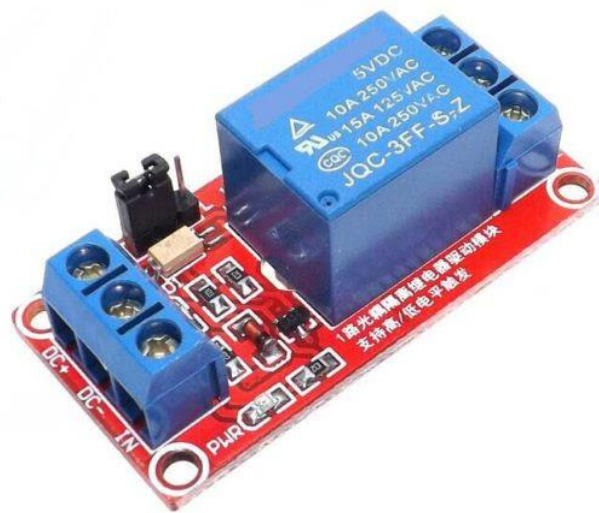
ESP8266（圖 3）是一款超低功耗的 UART-WiFi 透傳模組，擁有業內極富競爭力的封裝尺寸和超低能耗技術，專為移動設備和物聯網應用設計，可將使用者的設備連接到 Wi-Fi 無線網路上，進行互聯網或局域網通信，實現聯網功能。



▲圖 3 ESP8266 WIFI 模組

### 3. 繼電器

繼電器(Relay) (圖 4)，也稱電驛，是一種電子控制器件，它具有控制系統(又稱輸入迴路)和被控制系統(又稱輸出迴路)，通常應用於自動控制電路中,它實際上是用較小的電流去控制較大電流的一種「自動開關」。故在電路中起著自動調節、安全保護、轉換電路等作用。



▲圖 4 繼電器

### 4. WCS1800 霍爾感測器模組

WCS1800 (圖 5) 包含一精準、低溫漂、有溫度補償設計的線性霍爾 IC 與一經高溫燒成的 C 型環的電流轉換器，其內部 9.0mm 直徑的電流通道可允許使用者監控任何電流路徑並無須破壞或改變原有的系統機構。當電流通過內部電流通道時，C 型環的電流轉換器會將此一電流成比例的轉換成磁場，而線性霍爾 IC 又會將此一磁場成比例的轉換成輸出電壓。

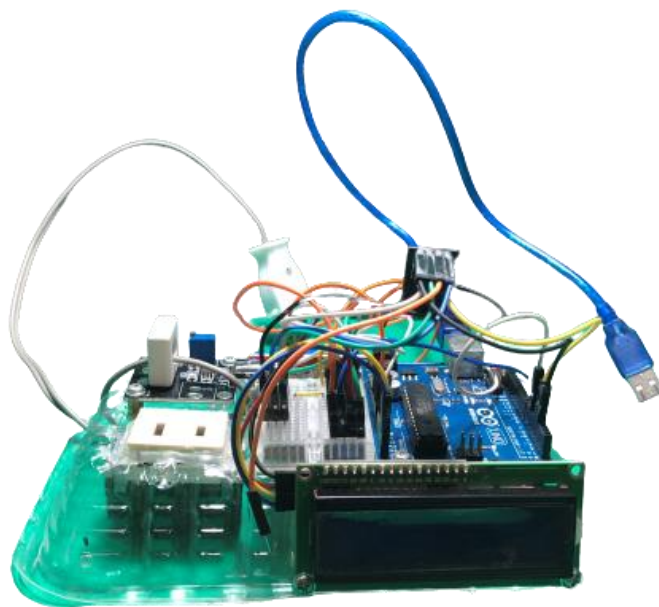


▲圖 5 WCS1800 霍爾感測器模組

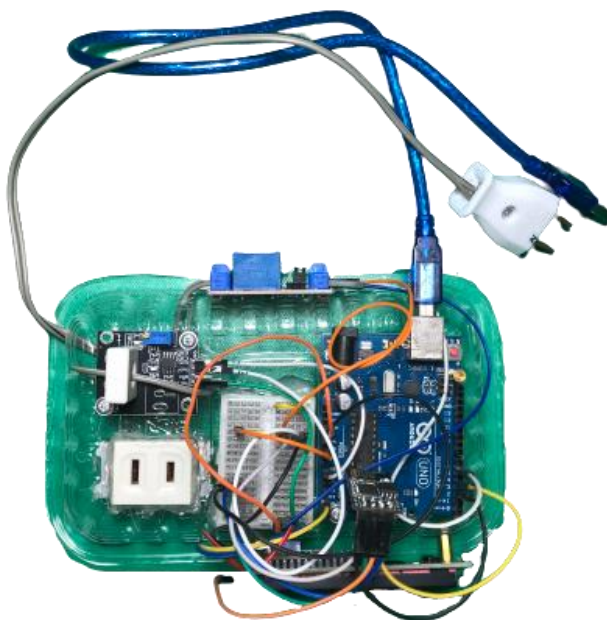
### (三) 功能說明

利用 ESP8266 模組使用 Wi-Fi 連線，LCD 在連線過程中會顯示網路的連線及上傳狀態，同時也會顯示目前插座上負載的消耗功率。連線成功後，Arduino 控制板會透過 ESP8266 模組上傳插座的負載消耗功率至伺服器，本地也可以透過 Excel 來記錄每秒電器使用時的功率。

## 成品展示



▲圖 6 成品展示 1



▲圖 7 產品展示 2



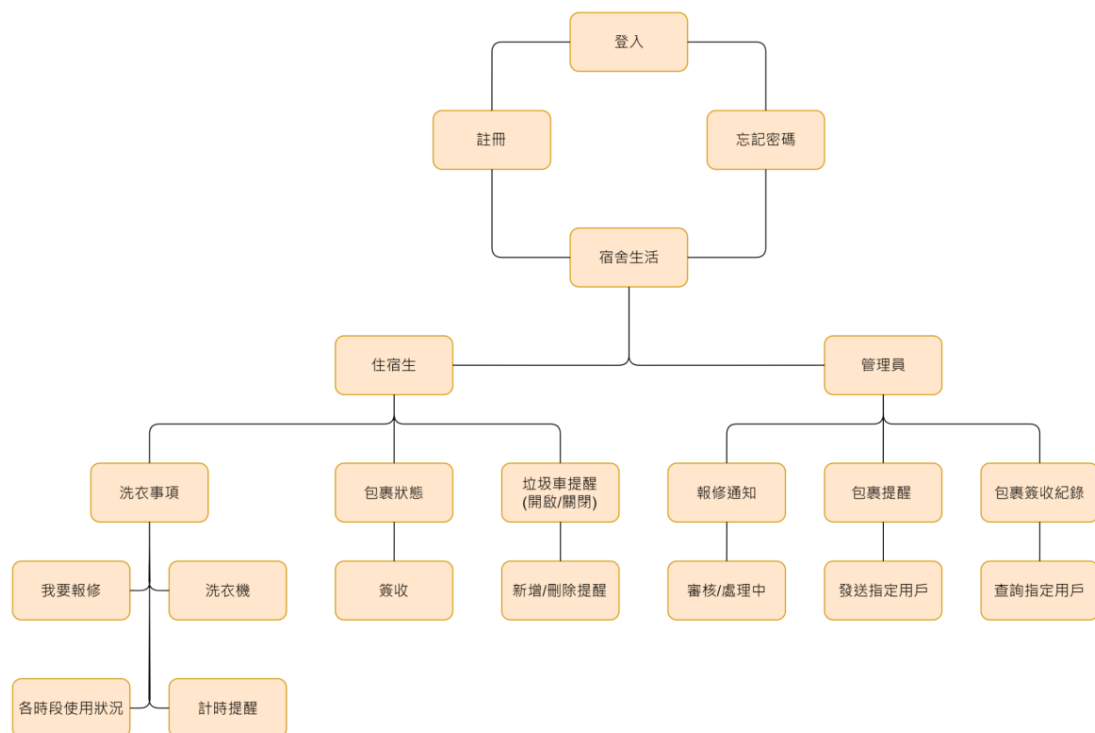
## 四、系統設計

### (一) 系統特色

系統特色主要為透過 AI 分析從物聯網裝置所讀取的資料，進行運算後得出使用者目前選擇洗衣模式，再推估預計完成洗衣的時間，資料庫採即時監聽，當機台使用狀態變化時(例如:使用中、可使用、故障)，使用者 APP 端也將即時更新為現況，另外 APP 上除了可以報修洗衣機台，還可查看以往洗衣機的使用率統計，以長條圖告訴使用者哪些為巔峰時段及離峰時段，進而達到有效控管洗衣機台。其他宿舍管理功能含有包裹/垃圾車提醒，包裹提醒可針對指定住宿生或全體使用者發送通知，垃圾車提醒會在使用者設置的固定時間跳出提醒，利用以上功能更全面的管理宿舍。

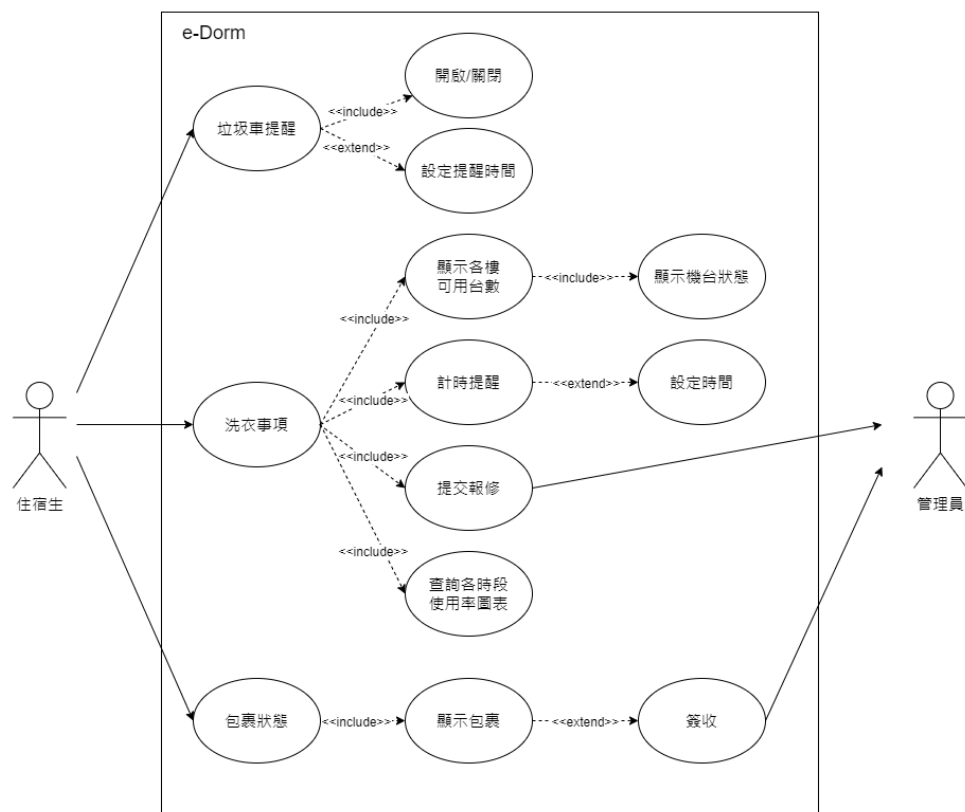
### (二) 系統架構圖

#### 1. APP 架構圖

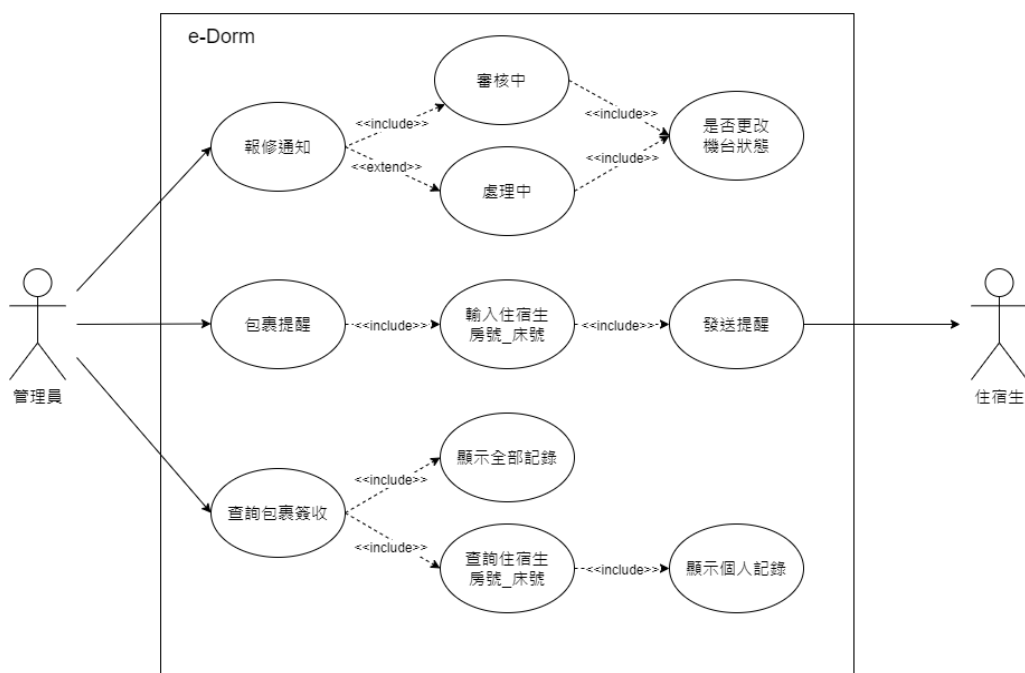


▲圖 8 APP 架構圖

## 2. 使用者案例圖

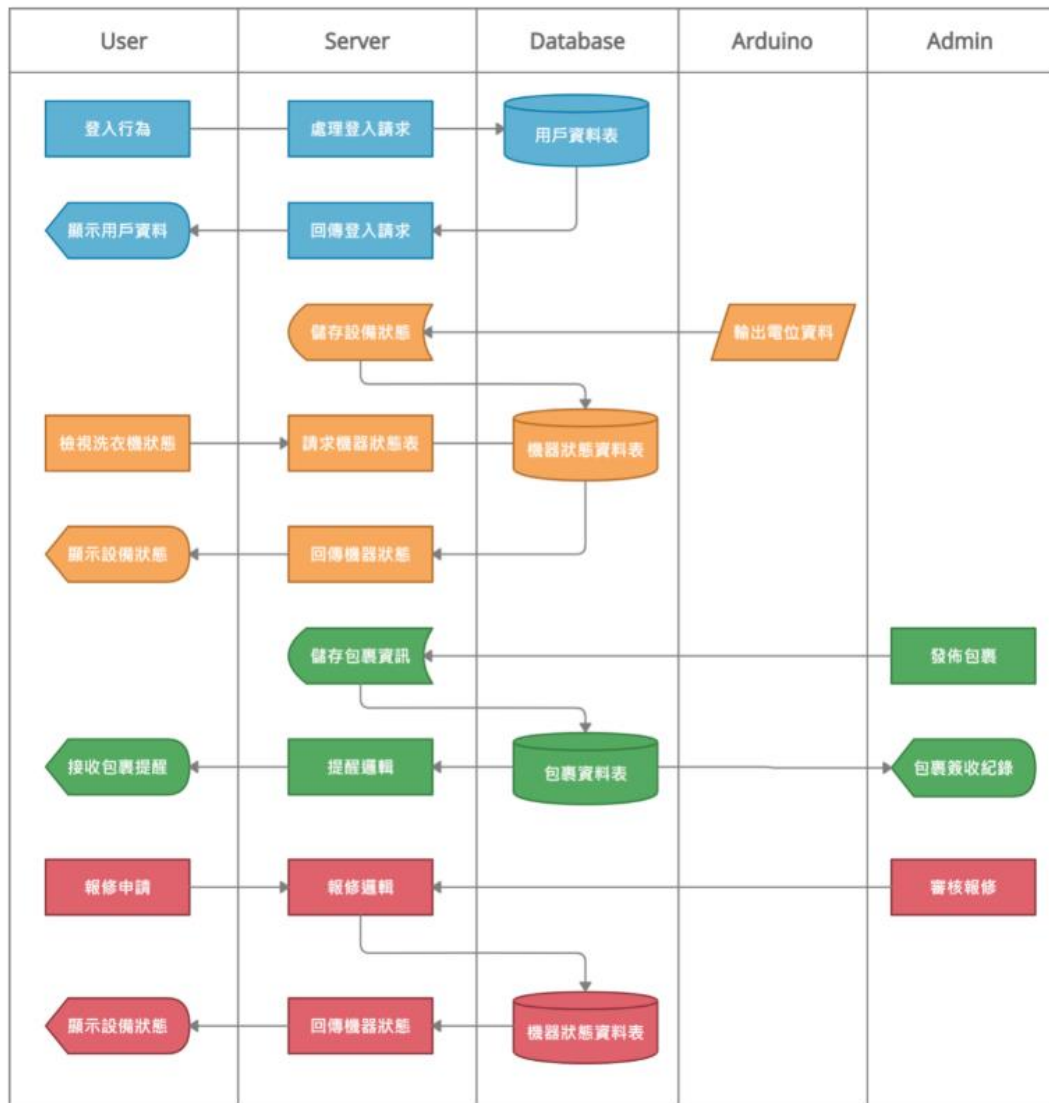


▲圖 9 使用者案例圖-住宿生



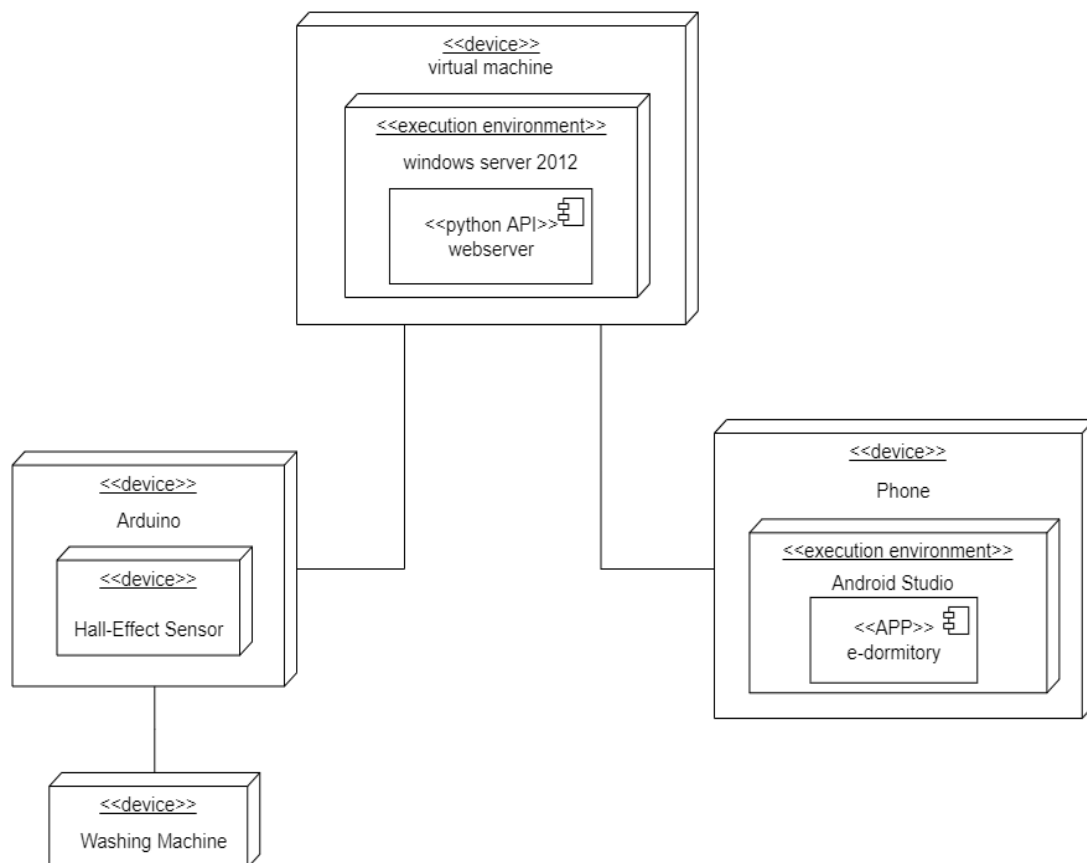
▲圖 10 使用者案例圖-管理員

### 3. 循序圖



▲圖 11 循序圖

#### 4. 部屬圖

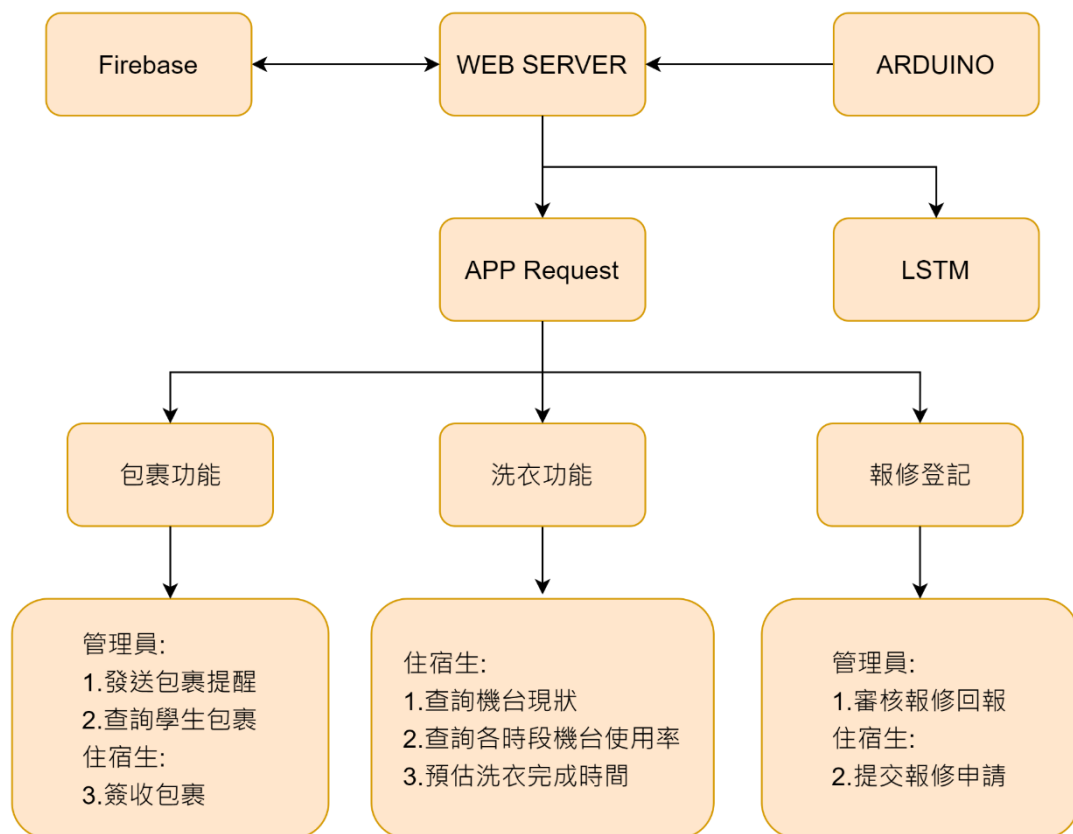


▲圖 12 部屬圖

## 5. 後臺管理架構圖

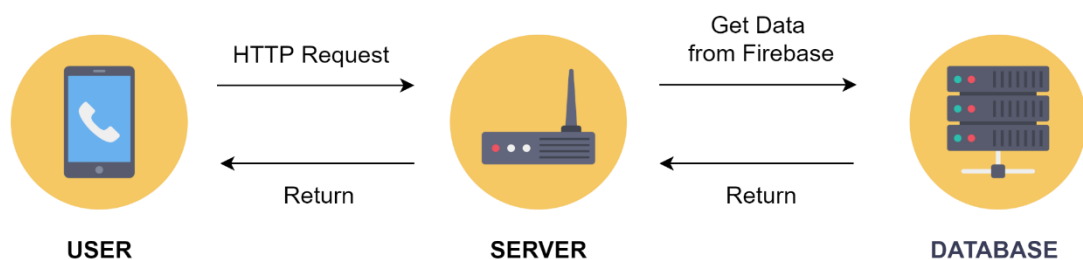
伺服器設有以下三種路由：

- **power** 路由供 Arduino 使用 GET 方法更新電數據
- **repair** 使用 POST&PUT 方法更新報修資料
- **package** 提供 GET&POST 方法取得資料及修改



▲ 圖 13 後臺管理架構圖

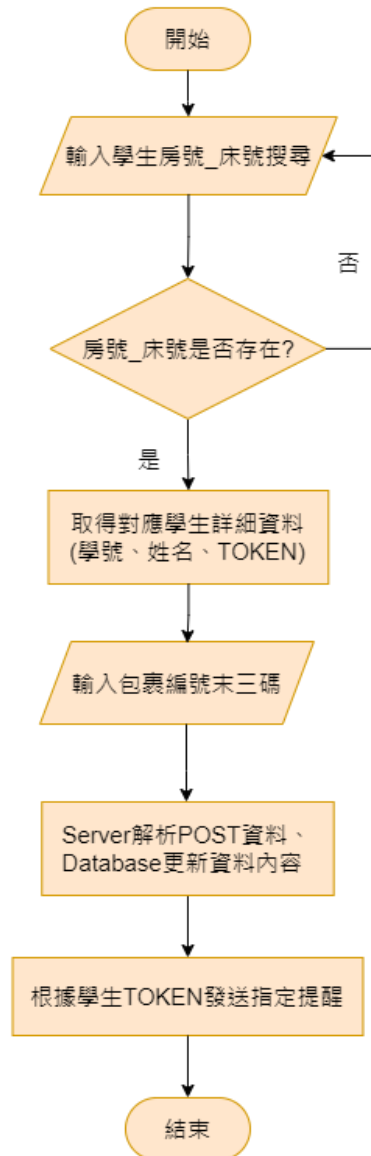
### APP Request



▲ 圖 14 APP Request

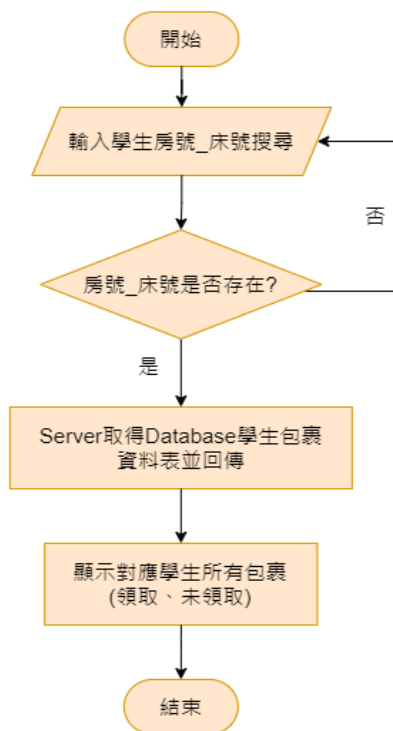
包裹功能流程圖：

(1) 管理員端-發送包裹提醒



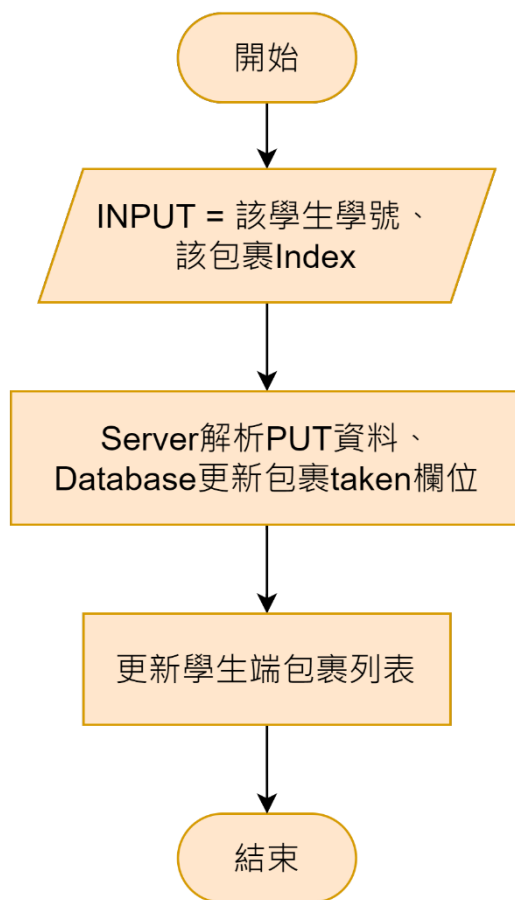
▲圖 15 管理員-發送包裹提醒

(2) 管理員端-查詢學生包裹



▲圖 16 管理員-查詢學生包裹

(3) 住宿生端-簽收包裹

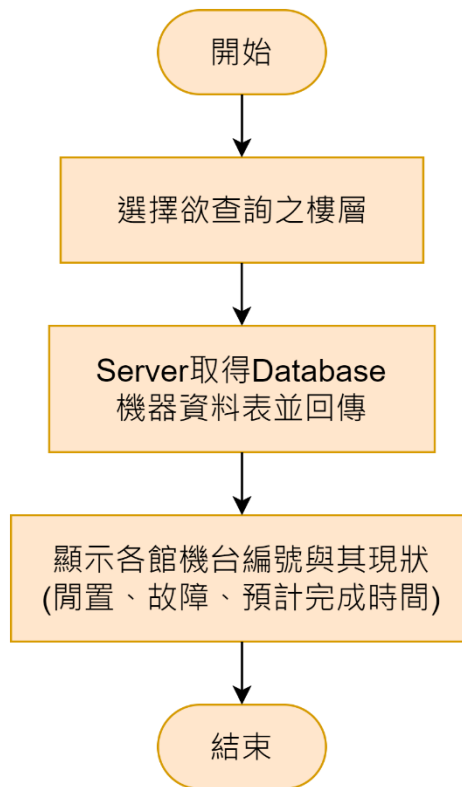


▲圖 17 住宿生-簽收包裹



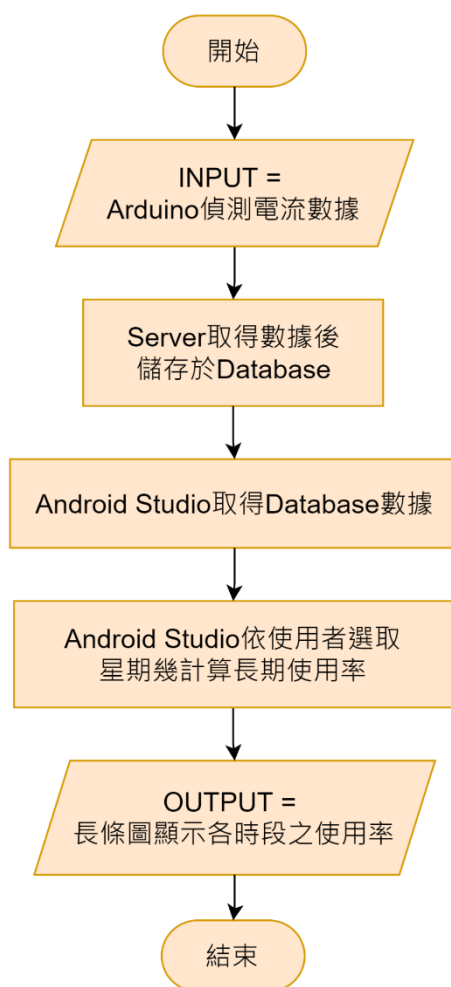
洗衣功能流程圖：

(1) 住宿生端-查詢機台現狀



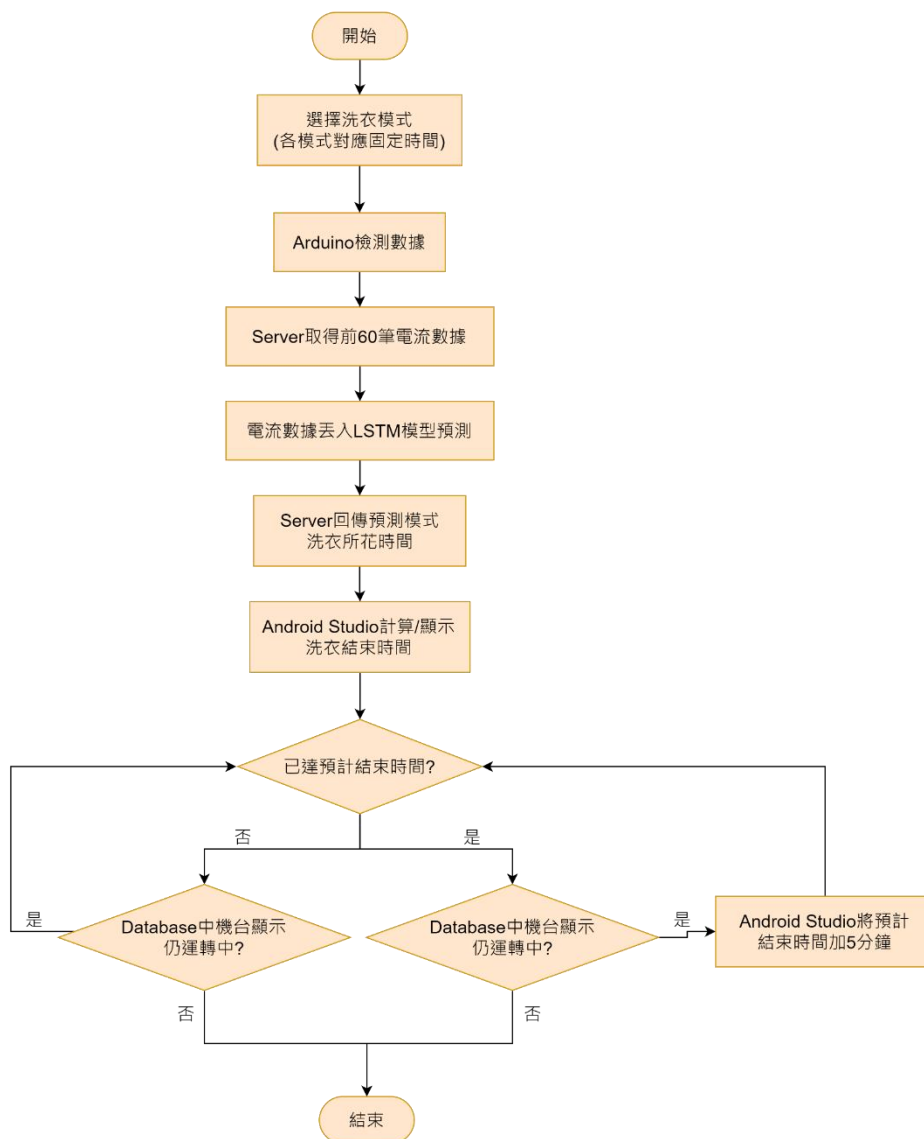
▲圖 18 住宿生-查詢機台現狀

(2) 住宿生端-查詢個時段機台使用率



▲圖 19 住宿生-查詢個時段機台使用率

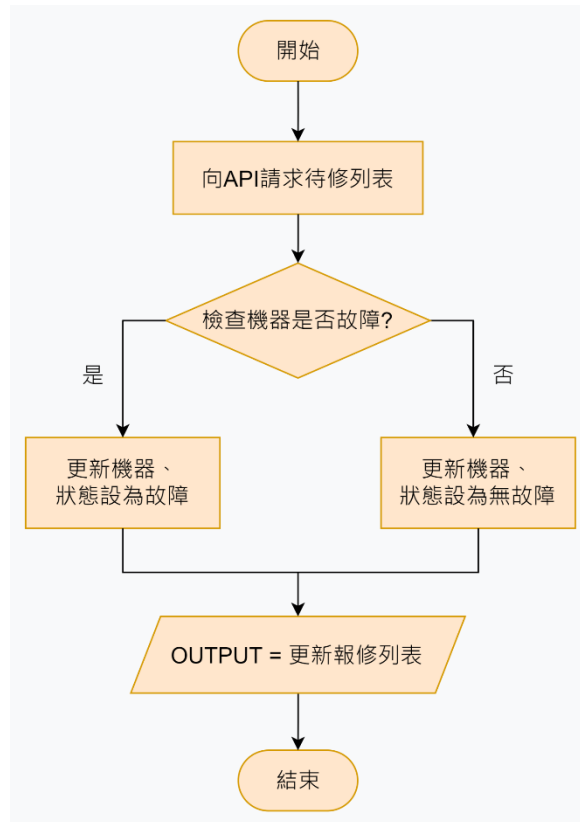
### (3) 住宿生端-預估洗衣完成時間



▲圖 20 住宿生-預估洗衣完成時間

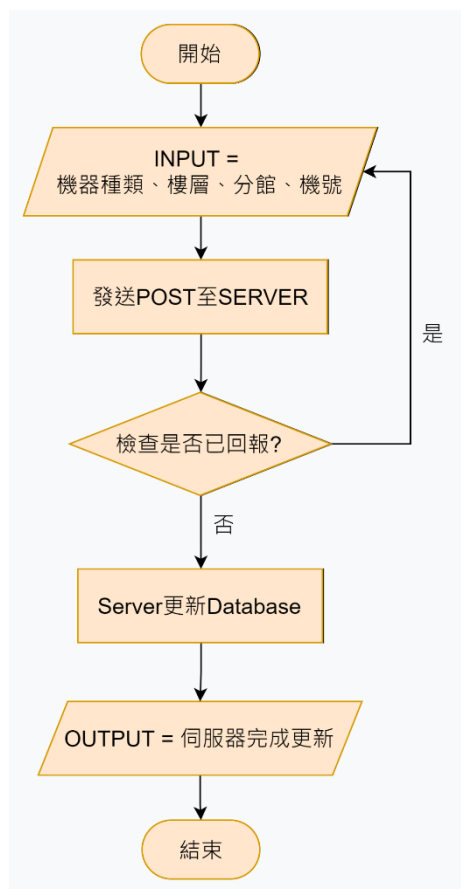
報修登記流程圖：

(1) 管理員端-審核報修回報



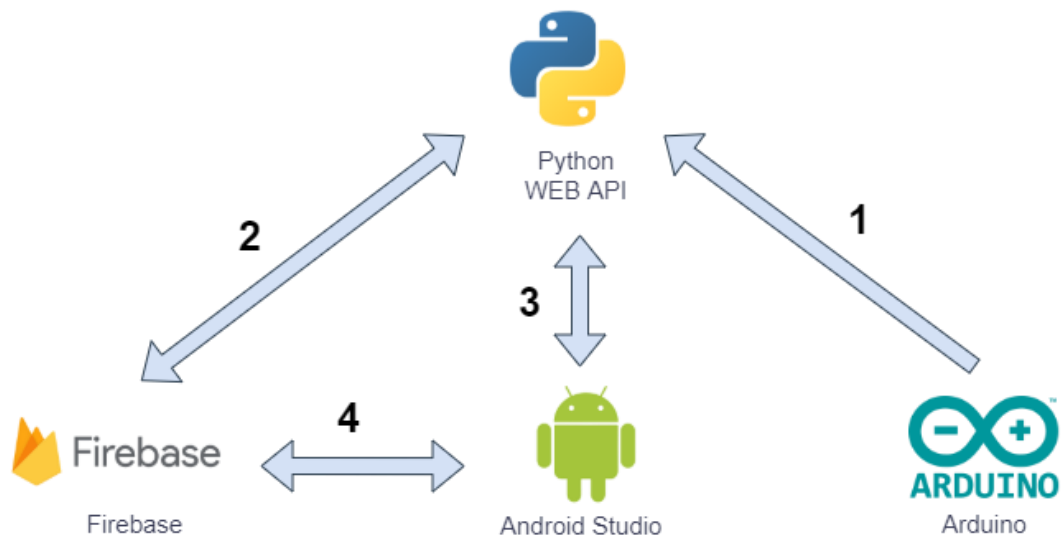
▲圖 21 管理員-審核報修回報

(2) 住宿生端-提交報修申請



▲圖 22 住宿生-提交報修申請

## 6. 系統運行架構圖



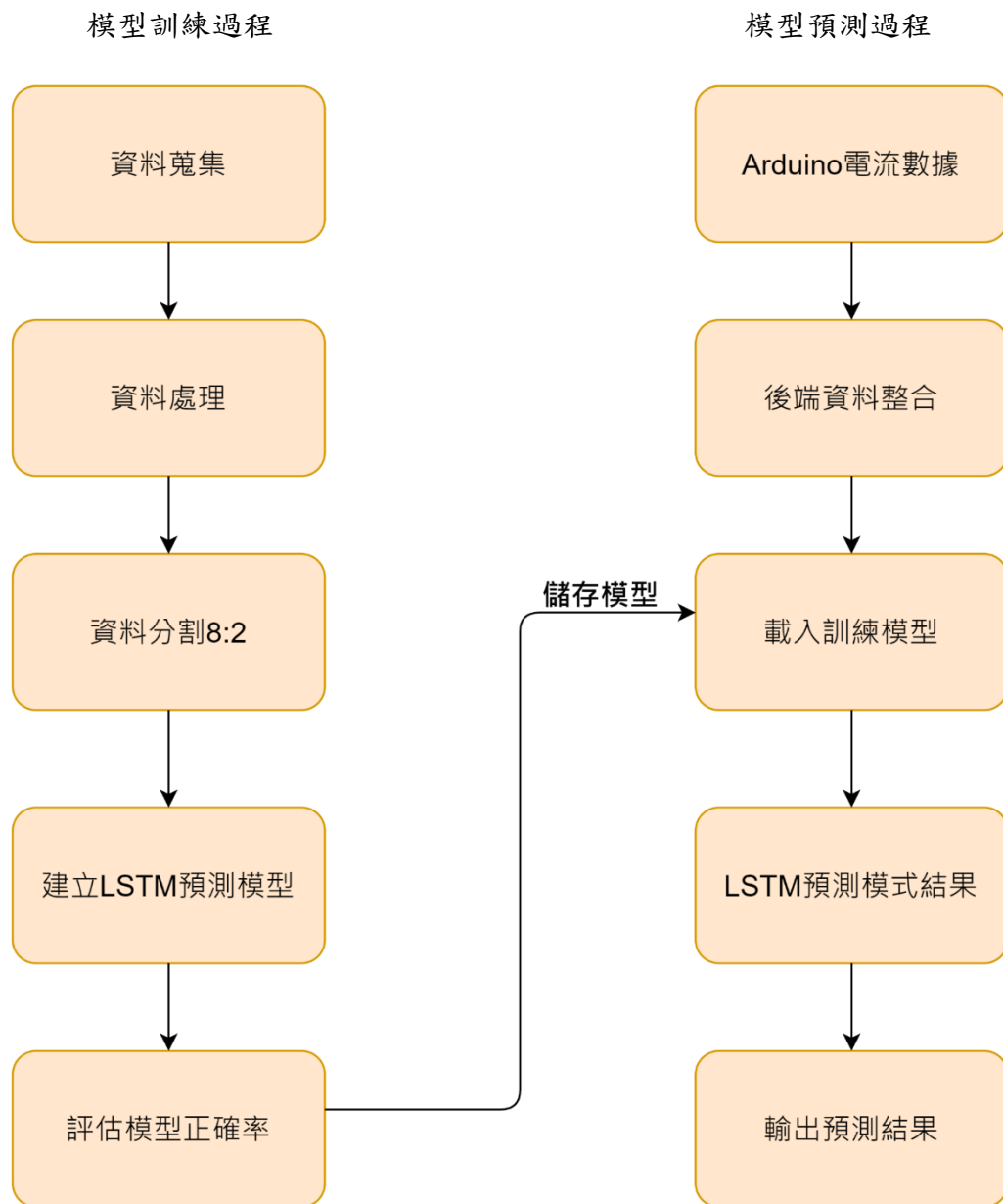
▲圖 23 系統運行架構圖

### 系統運行說明：

- (1) Arduino 利用特定 port 與伺服器連接，偵測電流數據後，發送狀態至伺服器，並由伺服器接手處理資料庫修改機器目前狀態(如:使用中、可使用、故障)，此外伺服器內部的 LSTM 模型，將讀取機器剛啟動時前 60 筆電流數據，並把 60 筆數據分成兩份長度為 30 筆的資料分別丟入模型預測，並得出兩種預測出模式，兩種模式所需花費時間相加後除以 2，即為最終伺服器回傳的預測模式所需時間，在第 25 頁機器學習中會進行更詳細 LSTM 模型解說。
- (2) 利用 Python 套件 Flask 編寫 WEB API 方法，架設伺服器，將 Firebase 資料庫連接伺服器，API 使用 dbkey 可以對資料庫進行 CRUD(新增、修改、刪除、查詢)操作，EX:伺服器可至資料庫內機器資料表更新機器現狀。

- (3) Android Studio 內可使用 HTTP Request，接收資料庫內容顯示在 APP 介面上。Firebase RealTime 即時資料庫允許專用 APP 唯讀，讓 Android 客戶端可以跳過 HTTP Request 階段，監聽最新的資料庫更新，EX:當 Android Studio 發現資料庫中機器狀態改變時，將立即做出對應並更改 APP 端呈現機台資訊。Firebase FCM 也可針對單獨裝置或是所有群體進行廣播發送推播，EX:管理員可發送個人包裹提醒給指定學生。
- (4) Firebase RealTime 即時資料庫允許專用 APP 唯讀，讓 Android 客戶端可以跳過 HTTP Request 階段，監聽最新的資料庫更新。Firebase FCM 也可針對單獨裝置或是所有群體進行廣播發送推播。當使用者選取洗衣模式後，利用 LSTM 模型預判須花費之時間，Android Studio 端畫面將顯示預計完成時間為幾點幾分，若洗衣機畫面預測時間已達，機台偵測仍為使用中，Android Studio 可更新資料庫中預測時間的欄位(增加 5 分鐘)。

### (三) 機器學習



▲圖 24 模型訓練、預測過程



### 機器學習說明:

由於洗衣機運行時，選取特定模式洗衣所需時間為固定值，且電流輸出值與時間存在相關性，因此利用 LSTM 模型分析電流數據預估使用者當前選擇的模式，利用此方法有效減少使用者等待空機的時間，且最大化洗衣機使用效率，三種模式與對應的洗衣時間分別為 **A(洗清次數為 2 次)=26 分鐘、B(洗清次數為 3 次)=48 分鐘、C(洗清次數為 4 次)=68 分鐘。**

為減少預估預約錯誤影響，伺服器內的 LSTM 模型分析偵測到的前 60 筆資料，得出兩種模式後，將取出並回傳兩模式的所花時間平均值，以增加預測容錯率，此外 Android Studio 將即時監聽資料庫，若預估時間已達時，仍偵測機台使用電流則將原預估時間加 5 分鐘，反之若預估時間內，機台已無電流通過即將機台狀態改為閒置。

#### 1. 資料蒐集

Arduino 裝置連接洗衣機插座，蒐集洗衣機當前電流使用量，利用 WiFi 將電流數據即時上傳，以此得知目前該洗衣/烘衣機台是否正在被使用，若判斷出機台正在使用中，利用蒐集機台開始運作後的前 30 筆電流訊號，丟入 LSTM 模型後，得出目前機台運作模式以及模式對應之運轉時間，**只取出前 30 筆資料訓練原因為進行預測時間有限，須在機台開始運轉後不久即判斷出使用者選取之模式。**

## 2. 丟入參數 Example

Mode A			Mode B			Mode C		
<u>Index</u>	<u>Power</u>	<u>Status</u>	<u>Index</u>	<u>Power</u>	<u>Status</u>	<u>Index</u>	<u>Power</u>	<u>Status</u>
<u>1.</u>	119	Busy	<u>1.</u>	25	Busy	<u>1.</u>	68	Busy
<u>2.</u>	68	Busy	<u>2.</u>	8	Busy	<u>2.</u>	16	Busy
<u>3.</u>	93	Busy	<u>3.</u>	25	Busy	<u>3.</u>	17	Busy
<u>4.</u>	59	Busy	<u>4.</u>	59	Busy	<u>4.</u>	34	Busy
<u>5.</u>	119	Busy	<u>5.</u>	17	Busy	<u>5.</u>	16	Busy
<u>6.</u>	51	Busy	<u>6.</u>	16	Busy	<u>6.</u>	67	Busy
<u>7.</u>	93	Busy	<u>7.</u>	8	Busy	<u>7.</u>	34	Busy
<u>8.</u>	85	Busy	<u>8.</u>	25	Busy	<u>8.</u>	8	Busy
<u>9.</u>	85	Busy	<u>9.</u>	16	Busy	<u>9.</u>	25	Busy
<u>10.</u>	85	Busy	<u>10.</u>	17	Busy	<u>10.</u>	25	Busy
<u>11.</u>	85	Busy	<u>11.</u>	33	Busy	<u>11.</u>	8	Busy
<u>12.</u>	0	Busy	<u>12.</u>	8	Busy	<u>12.</u>	16	Busy
<u>13.</u>	76	Busy	<u>13.</u>	33	Busy	<u>13.</u>	25	Busy
<u>14.</u>	102	Busy	<u>14.</u>	8	Busy	<u>14.</u>	34	Busy
<u>15.</u>	76	Busy	<u>15.</u>	16	Busy	<u>15.</u>	8	Busy
<u>16.</u>	102	Busy	<u>16.</u>	16	Busy	<u>16.</u>	50	Busy
<u>17.</u>	51	Busy	<u>17.</u>	51	Busy	<u>17.</u>	25	Busy
<u>18.</u>	93	Busy	<u>18.</u>	33	Busy	<u>18.</u>	8	Busy
<u>19.</u>	85	Busy	<u>19.</u>	0	Busy	<u>19.</u>	16	Busy
<u>20.</u>	85	Busy	<u>20.</u>	33	Busy	<u>20.</u>	25	Busy
<u>21.</u>	102	Busy	<u>21.</u>	0	Busy	<u>21.</u>	68	Busy
<u>22.</u>	76	Busy	<u>22.</u>	25	Busy	<u>22.</u>	16	Busy
<u>23.</u>	8	Busy	<u>23.</u>	0	Busy	<u>23.</u>	17	Busy
<u>24.</u>	76	Busy	<u>24.</u>	25	Busy	<u>24.</u>	34	Busy
<u>25.</u>	102	Busy	<u>25.</u>	16	Busy	<u>25.</u>	0	Busy
<u>26.</u>	51	Busy	<u>26.</u>	42	Busy	<u>26.</u>	229	Busy
<u>27.</u>	85	Busy	<u>27.</u>	33	Busy	<u>27.</u>	25	Busy
<u>28.</u>	76	Busy	<u>28.</u>	8	Busy	<u>28.</u>	0	Busy
<u>29.</u>	85	Busy	<u>29.</u>	33	Busy	<u>29.</u>	8	Busy
<u>30.</u>	93	Busy	<u>30.</u>	8	Busy	<u>30.</u>	25	Busy

▲圖 25 丟入參數

### 3. 資料處理

A、B、C 模式皆將資料標籤化後，若 Status 欄位出現連續 Busy，即判斷機台正在使用中，抓取其後 30 筆數據的 Power 值(圖 26)，並且增加 Type 欄位幫助紀錄數據所屬的模式，為增加訓練數據的數量，另外根據 A、B、C 三種模式模擬各 300 筆資料丟入訓練，由於取前 30 筆資料分析( $n \geq 30$ )，在大部分統計實例中樣本將符合常態分佈，為了更貼近實際情況將模擬資料中以其中 95%和 5%分開產生，模擬資料 300 筆中的 95%數據，產生符合三模式實測數據常態分布特徵的變數(圖 27)，剩下 5%數據則由將三模式實測數據 Power 值大小切成 0~5、5~10、10~40、40 以上數值段落，各段落配合產生的隨機變數加減(圖 27)，利用隨機加減部分藉此產生噪訊，最後將 900 筆所有資料分出 80%當作訓練資料(圖 28)，做完 One-hot encoding 後丟入訓練模型。

day	time	power	bound	status
2021/8/14	下午 02:32:32	119	Over	Busy
2021/8/14	下午 02:32:33	68	Over	Busy
2021/8/14	下午 02:32:34	93	Over	Busy
2021/8/14	下午 02:32:35	59	Under	Busy
2021/8/14	下午 02:32:36	119	Over	Busy
2021/8/14	下午 02:32:37	51	Under	Busy
2021/8/14	下午 02:32:38	93	Over	Busy
2021/8/14	下午 02:32:39	85	Over	Busy
2021/8/14	下午 02:32:40	85	Over	Busy
2021/8/14	下午 02:32:41	85	Over	Busy
2021/8/14	下午 02:32:42	85	Over	Busy
2021/8/14	下午 02:32:43	0	Under	Busy
2021/8/14	下午 02:32:44	76	Over	Busy
2021/8/14	下午 02:32:45	102	Over	Busy
2021/8/14	下午 02:32:46	76	Over	Busy
2021/8/14	下午 02:32:47	102	Over	Busy
2021/8/14	下午 02:32:48	51	Under	Busy
2021/8/14	下午 02:32:49	93	Over	Busy
2021/8/14	下午 02:32:50	85	Over	Busy
2021/8/14	下午 02:32:51	85	Over	Busy
2021/8/14	下午 02:32:52	102	Over	Busy
2021/8/14	下午 02:32:54	76	Over	Busy
2021/8/14	下午 02:32:56	8	Under	Busy
2021/8/14	下午 02:32:57	76	Over	Busy
2021/8/14	下午 02:32:57	102	Over	Busy
2021/8/14	下午 02:32:58	51	Under	Busy
2021/8/14	下午 02:32:59	85	Over	Busy
2021/8/14	下午 02:33:00	76	Over	Busy
2021/8/14	下午 02:33:01	85	Over	Busy
2021/8/14	下午 02:33:02	93	Over	Busy
2021/8/14	下午 02:33:03	85	Over	Busy
2021/8/14	下午 02:33:04	8	Under	Busy
2021/8/14	下午 02:33:05	76	Over	Busy
2021/8/14	下午 02:33:06	119	Over	Busy
2021/8/14	下午 02:33:07	59	Under	Busy
2021/8/14	下午 02:33:08	85	Over	Busy
2021/8/14	下午 02:33:09	76	Over	Busy
2021/8/14	下午 02:33:10	85	Over	Busy
2021/8/14	下午 02:33:11	85	Over	Busy
2021/8/14	下午 02:33:12	76	Over	Busy
2021/8/14	下午 02:33:13	93	Over	Busy
2021/8/14	下午 02:33:14	76	Over	Busy
2021/8/14	下午 02:33:15	8	Under	Busy
2021/8/14	下午 02:33:16	161	Over	Busy

2021/8/14	下午 02:33:16	161	Over	Busy
2021/8/14	下午 02:33:17	85	Over	Busy
2021/8/14	下午 02:33:18	76	Over	Busy
2021/8/14	下午 02:33:19	85	Over	Busy
2021/8/14	下午 02:33:20	85	Over	Busy
2021/8/14	下午 02:33:21	85	Over	Busy
2021/8/14	下午 02:33:22	85	Over	Busy
2021/8/14	下午 02:33:23	76	Over	Busy
2021/8/14	下午 02:33:24	85	Over	Busy
2021/8/14	下午 02:33:25	34	Under	Busy
2021/8/14	下午 02:33:26	110	Over	Busy
2021/8/14	下午 02:33:27	42	Under	Busy
2021/8/14	下午 02:33:28	85	Over	Busy
2021/8/14	下午 02:33:29	93	Over	Busy
2021/8/14	下午 02:33:30	76	Over	Busy
2021/8/14	下午 02:33:31	85	Over	Busy
2021/8/14	下午 02:33:32	85	Over	Busy
2021/8/14	下午 02:33:34	110	Over	Busy
2021/8/14	下午 02:33:35	76	Over	Busy
2021/8/14	下午 02:33:36	85	Over	Busy
2021/8/14	下午 02:33:37	16	Under	Busy
2021/8/14	下午 02:33:38	76	Over	Busy
2021/8/14	下午 02:33:39	93	Over	Busy
2021/8/14	下午 02:33:40	68	Over	Busy
2021/8/14	下午 02:33:41	102	Over	Busy
2021/8/14	下午 02:33:42	85	Over	Busy
2021/8/14	下午 02:33:43	76	Over	Busy
2021/8/14	下午 02:33:44	85	Over	Busy
2021/8/14	下午 02:33:45	68	Over	Busy
2021/8/14	下午 02:33:46	85	Over	Busy
2021/8/14	下午 02:33:47	76	Over	Busy
2021/8/14	下午 02:33:48	8	Under	Busy
2021/8/14	下午 02:33:49	76	Over	Busy
2021/8/14	下午 02:33:50	0	Under	Busy
2021/8/14	下午 02:33:51	93	Over	Busy
2021/8/14	下午 02:33:52	281	Over	Busy
2021/8/14	下午 02:33:53	323	Over	Busy
2021/8/14	下午 02:33:54	153	Over	Busy
2021/8/14	下午 02:33:55	25	Under	Busy
2021/8/14	下午 02:33:56	76	Over	Busy
2021/8/14	下午 02:33:57	8	Under	Busy
2021/8/14	下午 02:33:58	85	Over	Busy
2021/8/14	下午 02:33:59	8	Under	Busy
2021/8/14	下午 02:34:00	76	Over	Busy
2021/8/14	下午 02:34:01	8	Under	Busy
2021/8/14	下午 02:34:02	76	Over	Busy

▲圖 26 資料處理 1

```

# 原資料當作母體填入平均&標準差產生模擬資料
def generate_data(mean,std,n):
    g=np.round(np.random.normal(mean,std,n))
    return np.abs(g)

# 噪音資料產生
def noise(array):
    res = []
    for i in array:
        if i<10 and i>=5:
            res.append(i+np.random.randint(-3,55))
        elif(i>10 and i<40):
            res.append(i+np.random.randint(-5,60))
        elif(i>=0 and i<5):
            res.append(i+np.random.randint(0,55))
        elif(i>40):
            res.append(i+np.random.randint(-5,60))
    return res

# 產生資料量
total_size=300 #每part總資料量
noise_rate=0.05 #雜訊比率
dataSize = round(total_size*(1-noise_rate))
noise_size=round(total_size*0.05)

```

▲圖 27 資料處理 2

```

# MixData get Simple for x_train
mixData=pd.concat([df3,df2,df1],axis=0)
x_mixData=np.array(mixData.power.values)
y_mixDate=np.array(mixData.type.values)
# 切割資料集
x_train , x_test , y__train , y_test =train_test_split(x_mixData,y_mixDate,test_size=0.2,random_state=10)
x_train=np.array(list(x_train))
x_train.shape=(-1,30,1)

```

▲圖 28 資料處理 3

#### 4. 建立模型

使用機器學習的工具為 keras、pandas，程式語言為 Python，LSTM 輸入層與時間序列相關，因此須代入三維資料(batch\_size,時間序列,電數值)，連接 3 點輸出神經元。epochs 迭代的次數為 50 次，fit 參數 batch 設定為 5，總訓練資料 720 筆中有 25%當做驗證集，使用的 Activation Function 為"Softmax"，可將 Y 值轉為機率大小，損失函數設為對數損失，適合用於多分類之問題，經過多次試驗得出的較佳模型。

```
model = Sequential()
model.add(LSTM(10,input_length=30,input_dim=1))
model.add(Dense(units=3,activation='softmax'))
model.compile(loss=losses.BinaryCrossentropy(),optimizer=optimizers.Adam(), metrics=[ 'accuracy' ])
model.summary()
```

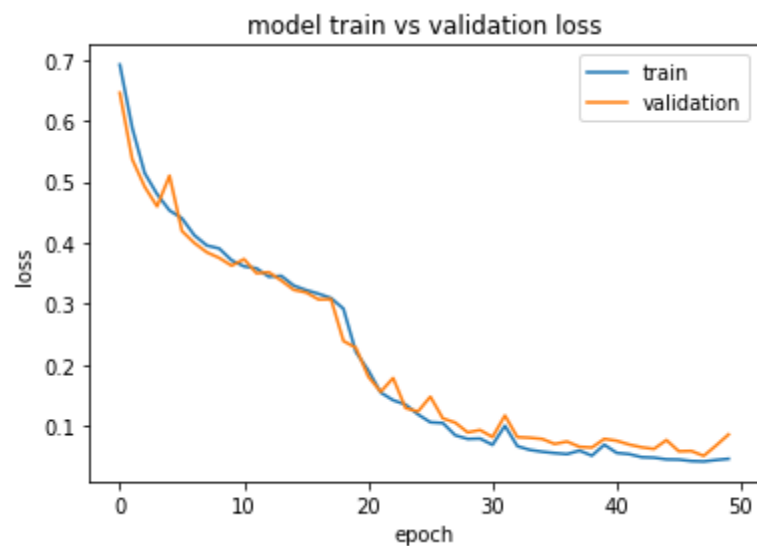
Model: "sequential\_4"

Layer (type)	Output Shape	Param #
lstm_4 (LSTM)	(None, 10)	480
dense_4 (Dense)	(None, 3)	33
Total params: 513		
Trainable params: 513		
Non-trainable params: 0		

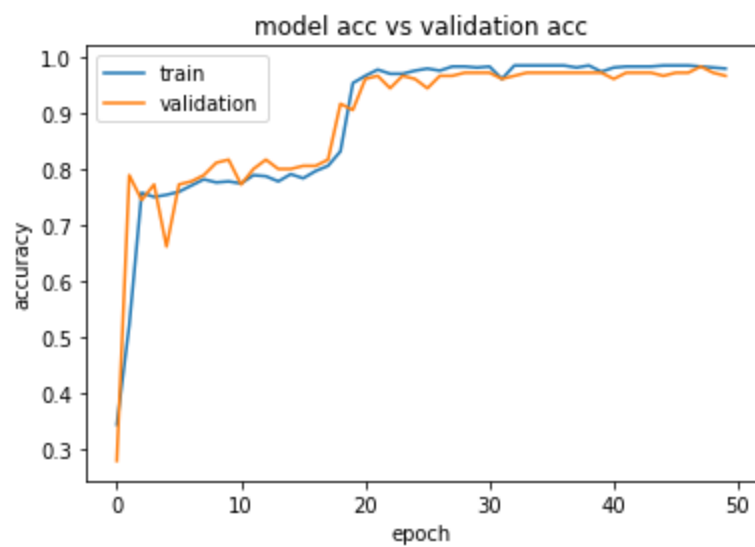
▲圖 29 建立模型

## 5. 模型評估

以  $\text{Test} = 0.2$  分出測試集，測試集準確度約為 95% ~ 97%，當 A、B、C 三種模式前幾筆資料(不含已訓練過)，模型多數時候皆可正確判讀。



▲圖 30 模型評估 1



▲圖 31 模型評估 2

acc: 0.9666666666666667

	predict_result	label_result
A	56	56
B	56	59
C	68	65

▲圖 32 模型評估 3

最終模式對應時間：

模式	A(27分鐘)	B(40分鐘)	C(68分鐘)
----	---------	---------	---------

▲圖 33 模型對應時間

## 6. 儲存模型

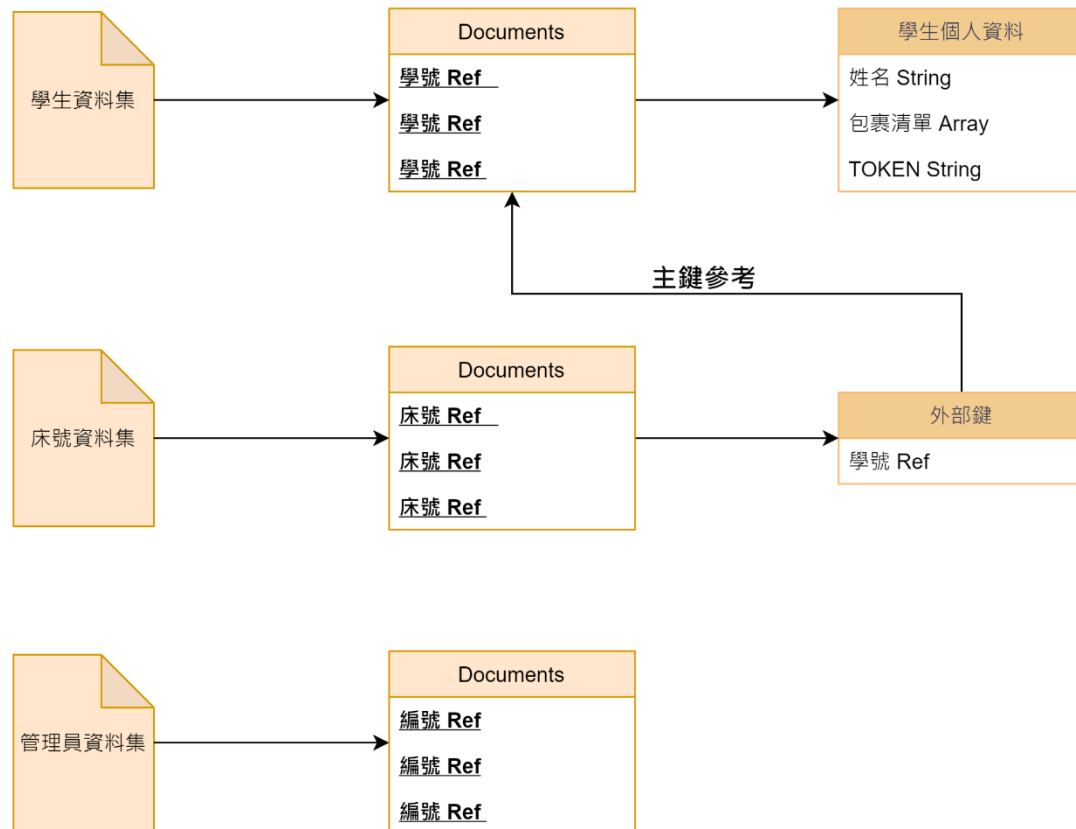
利用以下程式碼將最終模型儲存，機器學習完成。

```
model.save("myModel.h5")
```

▲圖 34 儲存模型

## (四) 資料庫設計

### 1. Firestore Database



▲圖 35 Firestore Database



## (1) 管理員資料集

如下圖管理員資料集儲存管理員清單，以管理員編號作為文件主鍵。

+ 新增集合	+ 新增文件
admin >	8iAefpYcxBzMtnH3GBsT >
roomList	SeLZAS5RgkU2EXHZCBQ9
student	b1KcYEJKSEN4vK08kide
	kH3v23r19G9EG8GQUKTg
	y0GQS3B1yknYrGBrrZax
	ySPiiIL4nzc0JfcUBFdr

▲圖 36 管理員資料集

## (2) 床號資料集

以住宿生的房號及床號作為主鍵，設有 Reference 欄位對應到學生資料集，以便將學生包裹資料添加至其學號底下，並且管理員可透過床號搜尋對應學生包裹。

+ 新增集合	+ 新增文件	+ 新增集合
admin	1218_1	+ 新增欄位
roomList >	1645_2	stdRef: /student/407630861
student	2307_3	
	2506_4	
	3409_4 >	
	3631_4	

▲圖 37 床號資料集

### (3) 學生資料集

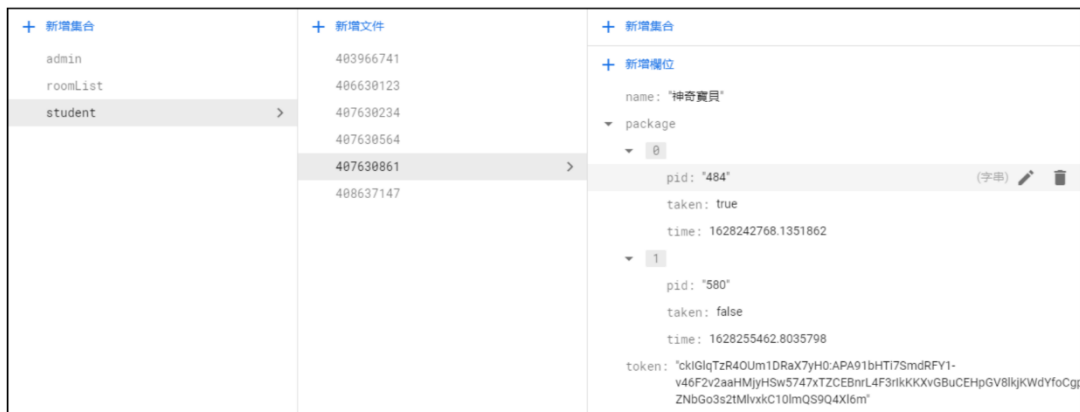
儲存住宿生資訊、以學號作為文件主鍵。文件內部儲存格式為：

學生姓名:STRING

包裹陣列:ARRAY

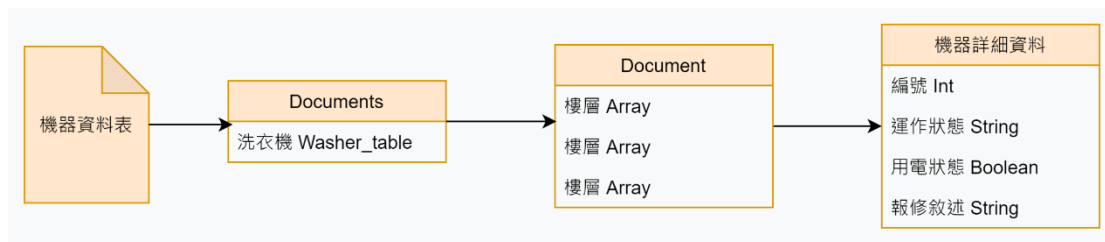
登入裝置 Token:STRING

包裹內容:MAP{包裹編號、領取狀態、建立時間}



▲圖 38 學生資料集

## 2. Realtime Database



▲圖 39 Realtime Database

### (1) 機器資料表

Realtime 資料庫為 JSON 格式，內有洗衣機目錄，目錄內部由分館編號加樓層作為 Key，鍵對應的值為 map 陣列，陣列儲存該區域所有機台資料，con、ele 欄位會隨伺服器端接收到的 Arduino 現況改變，偵測現況若為有電流通過(使用中)，則將 con 設為 using、ele 設為 true，反之若不通電 ele 設為 false、con 為 broken 或 usable 由伺服器是否接收到報修決定，id 欄位為各機台之編號，rep 為學生報修時附加之敘述，當機台被使用時，伺服器發送更新使機台資料添加 mode、finish 欄位，mode 儲存 LSTM 模型預測的使用模式，finish 儲存機台運作預計的完成時間(單位:毫秒)。

機台資料:MAP{運作狀態，用電狀態，機台編號，報修敘述}

```
"Washer" : {
  "1F" : [ {
    "con" : "usable",
    "ele" : false,
    "id" : 1,
    "rep" : ""
  }, {
    "con" : "using",
    "ele" : true,
    "id" : 2,
    "rep" : ""
  }, {
    "con" : "broken",
    "ele" : false,
    "id" : 3,
    "rep" : "YVDS😄😄"
  }, {
    "con" : "using",
    "ele" : true,
    "id" : 4,
    "rep" : ""
  }, {
    "con" : "usable",
    "ele" : false,
    "id" : 5,
    "rep" : ""
  }, {
    "con" : "usable",
    "ele" : true,
    "id" : 6,
    "rep" : ""
  }, {

```

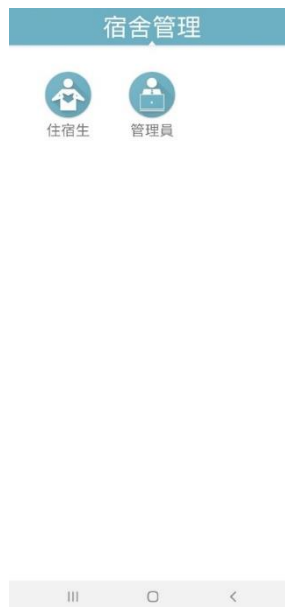
▲圖 40 機器資料表

## 五、系統功能說明

為達到有效的運作及管理，進入宿舍管理後我們將使用者分為「住宿生」及「管理員」，以下先介紹住宿生端的個別功能，而後再闡述管理員端與住宿生端間的連動功能。



▲圖 41 主頁



▲圖 42 宿舍管理頁

## (一) 住宿生端



▲圖 43 住宿生頁



▲圖 44 住宿生-我要洗衣頁

## 1. 垃圾車提醒

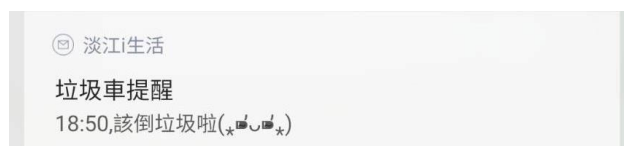
若住宿生需提醒自己要記得倒垃圾，可點選新增提醒按鈕自行設定要在垃圾車前來多少分鐘要提醒（圖 45），時間到後將顯示至狀態列上。（圖 47）



▲圖 45 垃圾車提醒提示窗頁



▲圖 46 垃圾車提醒頁



▲圖 47 垃圾車提醒

## 2. 我要洗衣




為本 App 之重點功能，讓住宿生不再有想洗衣服走到洗衣間卻沒有洗衣機可用的窘境，以下分別介紹其中功能並詳述之。

### (1) 洗衣機

進入畫面後會先依照樓層進行分類（圖 48），住宿生可依照自己所住的樓層查看有幾台洗衣機可使用，再點選進入後，會依照宿舍配置分為一館、二館、三館，並可看到洗衣機由不同顏色顯示其狀態為閒置、運轉或故障（圖 49）。而在運轉的機台中，住宿生可直接透過點選即跳出洗衣提醒視窗（圖 51），當洗衣機狀態由運轉中變回閒置，就會發送提醒給住宿生。（圖 52）

※依照學校配置，一樓為洗衣房，因此加上配置圖方便住宿生對照機台編號。

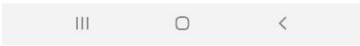
(圖 53)

洗衣機		
	1F	共有9台可用 >
	2F	共有8台可用 >
	3F	共有8台可用 >
	4F	共有8台可用 >
	5F	共有9台可用 >
	6F	共有7台可用 >



▲圖 48 洗衣機頁

3F洗衣機		
一館	二館	三館
	一館-3F-1	閒置
	一館-3F-2	故障
	一館-3F-3	閒置



▲圖 49 2F-6F 洗衣機頁  
40

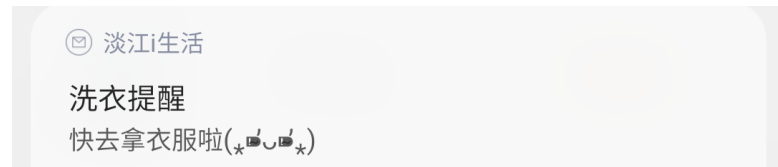




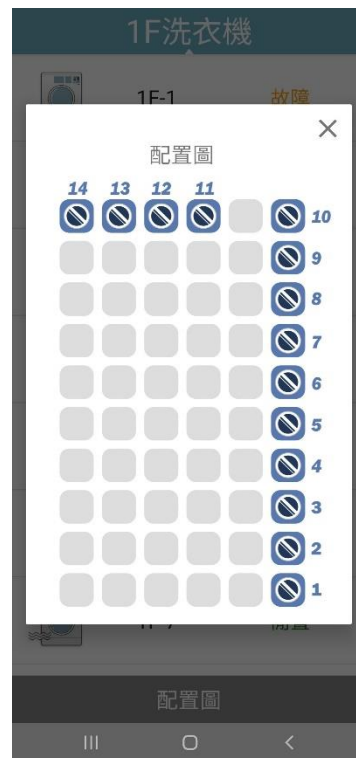
▲圖 50 1F 洗衣機頁



▲圖 51 洗衣提醒視窗



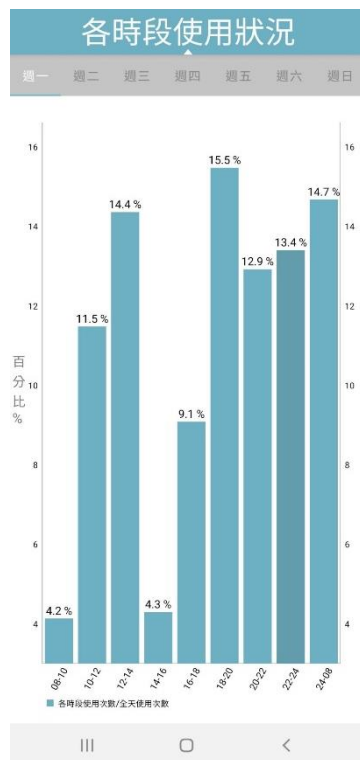
▲圖 52 洗衣提醒



▲圖 53 洗衣機配置圖頁

## (2) 使用率

住宿生可查看週一到週日各時段機台的使用情況，利用柱狀圖圖表清楚呈現哪些時間為熱門時段，以避開洗衣人潮。



▲圖 54 使用率頁

### (3) 洗衣機報修

住宿生遇到機台故障時，可透過此表單填寫機台編號和狀況說明回報給管理員。

洗衣機報修

洗衣機

三樓 一館 2

狀況說明

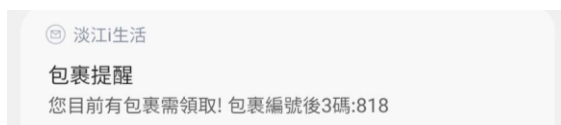
漏水了 😊

報修

▲圖 55 洗衣機報修頁

#### (4) 待領包裹

當管理員傳送的包裹通知後，會在住宿生的手機狀態列跳出提醒（圖 56），住宿生領取完包裹按下簽收按鈕後即清空此欄（圖 58），在管理員端會將包裹狀態由未領改為已領。（圖 57）



▲圖 56 待領包裹頁



▲圖 57 包裹提醒頁



▲圖 58 包裹簽收確認視窗

## （二） 管理員端



▲圖 59 管理員頁

### 1. 報修通知

分為「審核中」（圖 59）以及「處理中」（圖 62）。管理員在接收到住宿生所發送的報修通知後，會在管理員主頁面顯示標記通知，點選進入就能在審核中看到報修的機台，查看狀況說明並實際確認機台是否故障（圖 60），若有故障，則可將該機台狀態設為故障（圖 61），畫面將跳轉到處理中，修理完後再從處理中將狀態改為可使用。



▲圖 60 報修通知頁-審核中



▲圖 61 狀況說明頁





▲圖 62 報修確認視窗頁



▲圖 63 報修通知頁-處理中

## 2. 包裹提醒

管理員接收包裹後在輸入框打上該住宿生的房號和床號按下搜尋按鈕，確認其姓名學號正確後，再輸入包裹上編號末三碼按下發送，即發送包裹提醒到住宿生端，並可同時在包裹簽收紀錄中看到住宿生簽收包裹的狀態顯示。

包裹提醒

2504\_3 🔍

床號：2504\_3  
姓名：派大星  
學號：407630358  
包裹編號末3碼：818

送出

☰ □ <

▲圖 64 包裹提醒頁

### 3. 包裹簽收紀錄

透過下拉式選單，管理員可選擇以最新、已領或未領的篩選條件，可再加上搜尋房號床號的方式來查看住宿生包裹簽收的狀態，藉此來控管包裹去向。



▲圖 65 包裹簽收紀錄頁

## 六、 結語

本專題團隊利用智慧插座感測洗衣機狀態並結合人工智慧判斷模式的技術、住宿生與管理員之間的包裹功能，除了增加宿舍的方便性與機能性外，也讓宿舍進行更有效的管理。

目前硬體有網路連線不穩與檢測數據不正確的問題會影響到使用狀態與洗衣模式的判斷，且無法直接在 APP 上顯示洗衣機運轉倒數時間，是我們須進一步突破的部分。待未來技術成熟後在硬體上的升級以幫助我們更準確的監測機台，並增加更多的互動性介面，帶給每位使用者更佳的操作體驗及服務。

## 七、 未來展望

### （一） 偵測其他宿舍設備

目前我們使用的硬體設備可正確偵測洗衣機電流，對於宿舍內其他機種像是烘衣機、脫水機等則尚未經過測試，因而在 APP 以及後端系統當中，尚保留機種擴充的空間，待硬體能檢測更多元的設備時，APP 可以透過調整程式以及新增 LSTM 訓練模型，來檢測不同機台的不同模式，讓宿舍內所有機台使用上規劃更全面。

### （二） 新增預約制度

未來若能夠推出預約制度將使整體運作更有效率。雖然目前看來會因為無法知道上一位使用者何時會去拿衣服，以及有人插隊等種種潛在因素而較為窒礙難行，皆有待硬體功能的再擴充更制度的完整化，但如果能夠建立一預約系統，住宿生們便能夠提前進行洗衣機的預約，直接確定他何時能夠使用。

### (三) 與自助洗衣店合作

除了應用於學校宿舍，我們也認為能將此系統的硬體應用在自助洗衣店的管理上，同理地使客人能夠在 APP 上能夠知道是否有機台可供使用，透過與自助洗衣店的合作，也能為我們增加額外的營收。

## 八、參考資料

- (1)楊明豐(2018)。Arduino 物聯網最佳入門與應用：打造智慧家庭輕鬆學。臺北市：碁峰。
- (2)遞歸神經網路（RNN）和長短期記憶模型（LSTM）的運作原理（民 106 年 6 月 27 日），取自：[https://brohrer.mcknote.com/zh-Hant/how\\_machine\\_learning\\_works/how\\_rnn\\_lstm\\_work.html](https://brohrer.mcknote.com/zh-Hant/how_machine_learning_works/how_rnn_lstm_work.html)
- (3)長短期記憶神經網路（LSTM）介紹以及簡單應用分析（民 108 年 10 月 5 日），取自：<https://www.itread01.com/content/1570279682.html>