國立彰化師範大學數學學系

人工智慧

期末專題報告

Chatbot 訂餐機器人實作

授課教師

陳巧旻老師

作者：

資工四 許加宜 (S0854006)

民 國 1 1 2 年 6 月 26 日

**目 錄**

1. **研究動機與目的**
   1. 研究動機與目的
2. **系統描述與架構**

2.1 系統架構

2.2 系統功能與環境說明

**三、系統開發流程**

3.1 Dialogflow開發

3.1.1 Dialogflow對話設計

3.1.2 Entity設計

3.1.3 Context設計

3.1.4 Intent設計

3.1.5 操作

3.2 Lionfree Server開發

3.3 APP開發

**四、成果展示**

**五、結論與心得**

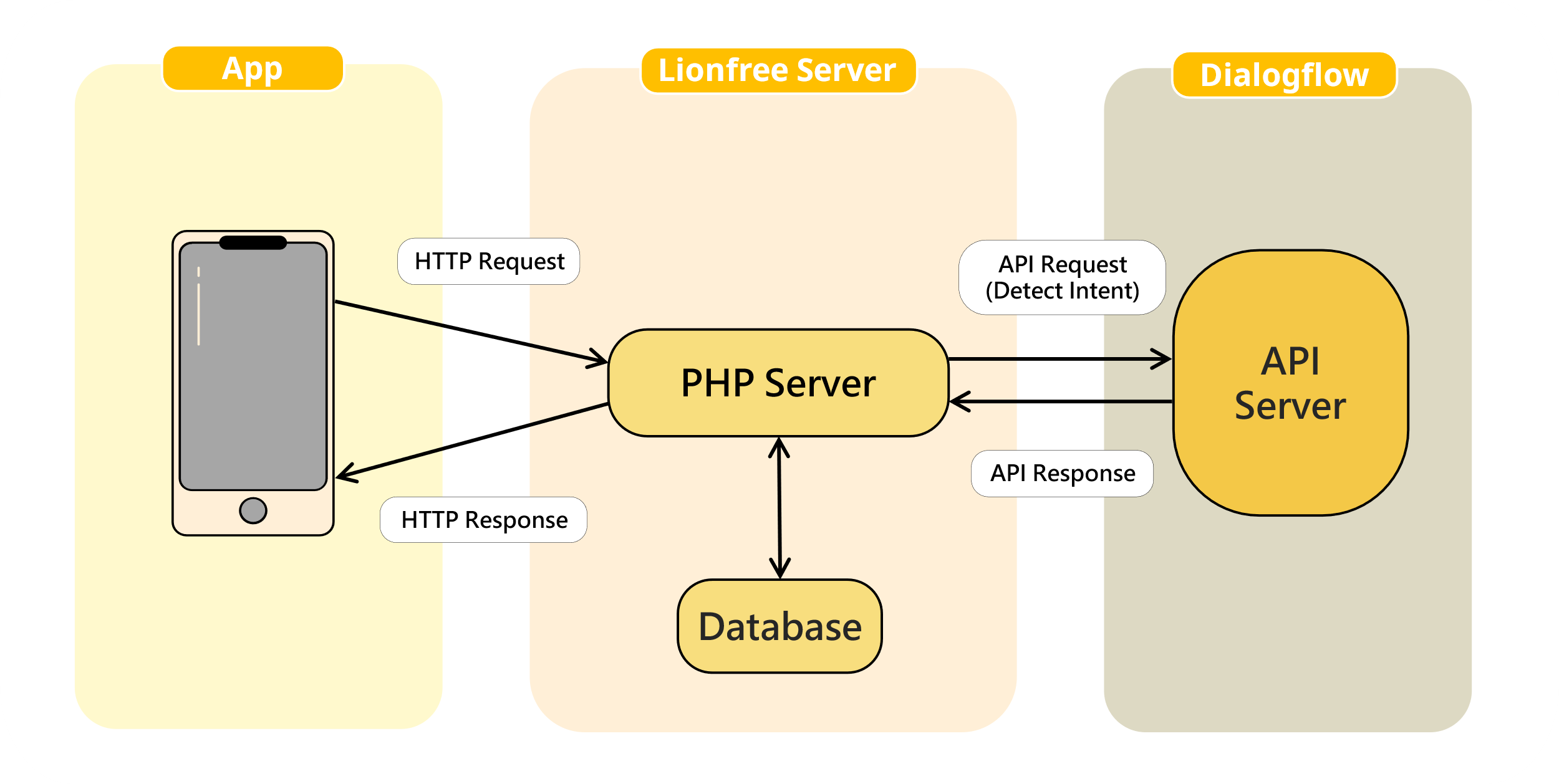
5.1 遭遇問題與解決方式

5.2 心得

1. **研究動機與目的**
   1. 研究動機與目的

現今聊天機器人的技術漸漸成形，也逐漸成為一個熱門的趨勢。因此想藉由這個機會，實作chatbot。

另外想在壽司訂購APP中加入自然語言處理助手，例如語音聊天機器人助手等，所以先從核心的部分做起，就是文字的聊天機器人。

1. **系統描述與架構** 
   1. 系統架構

圖一 系統架構圖

訂餐機器人的系統架構如圖一示，使用者利用APP發送HTTP request向Lionfree Server請求服務，Lionfree Server 再傳送API Request給Dialogflow的API Server來偵測Intent，並回傳結果給Lionfree Server ，Lionfree Server再根據需求讀寫資料庫，最後再將聊天機器人的回覆結果回傳結果給APP，將所需之資料呈現。

* 1. 系統功能與環境說明

1. 推薦商品：隨機推薦一個不在使用者購物車的商品。
2. 推薦新品：隨機推薦一個不在使用者購物車的商品。
3. 介紹商品：提供使用者詢問的商品詳細資料。
4. 加入購物車：幫助使用者將商品及購買數量加入購物車。
5. 個人訂單查詢：查看此使用者送出過的所有訂單。
6. 環境：
   1. APP設計：
      * 1. Android Studio
   2. PHP開發：
      * 1. Visual Studio Code
        2. Lionfree
7. 資料庫：
   1. MySQL
8. 聊天機器人：
   1. Dialogflow
9. **系統開發流程**

3.1 Dialogflow開發

1. Dialogflow是由 Google 開發的自然語言處理平台，用於建立自動化的對話系統，使使用者能夠通過自然語言chatbot進行互動。
2. 其使用了一種基於RNN的技術，稱為序列到序列學習（Sequence-to-Sequence Learning）。
3. 對於 Dialogflow 的具體模型架構和訓練方法，Google 並沒有公開詳細的技術細節。
4. Entity、Intent、Context為構成對話的重要元素。
   * 1. Entity：代表參數，例如日期或是數量。
     2. Intent：代表使用者的意圖。
     3. Context：在對話中傳遞上下文。

3.1.1 Dialogflow對話設計

一張含有 文字, 圖表, 行, 字型 的圖片

自動產生的描述

圖二 Dialogflow對話設計

圖二為訂餐機器人之對話設計，矩形的框為Intent，其中的紅字為使用者輸入的對話，下方的黑字為聊天機器人回應的字串。雲朵狀的圖案是Context。

由圖可見本系統共有12個Intent，4個context。最左邊五個Intent對應到本系統的功能，為不需要Input Context就能被啟動的主Intent，其他的Intent都需要有Input Context才能被啟動。

這個設計並不要求對話要完成整個流程，對話隨時可以被五個主Intent 打斷，從而進行一個新的流程。

3.1.2 Entity設計

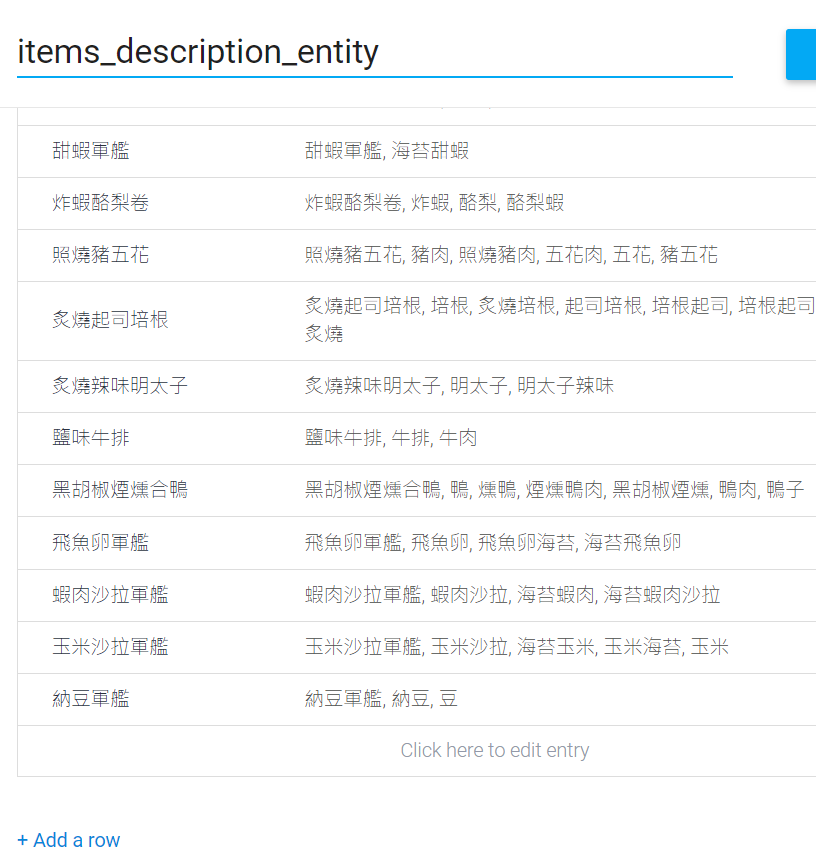
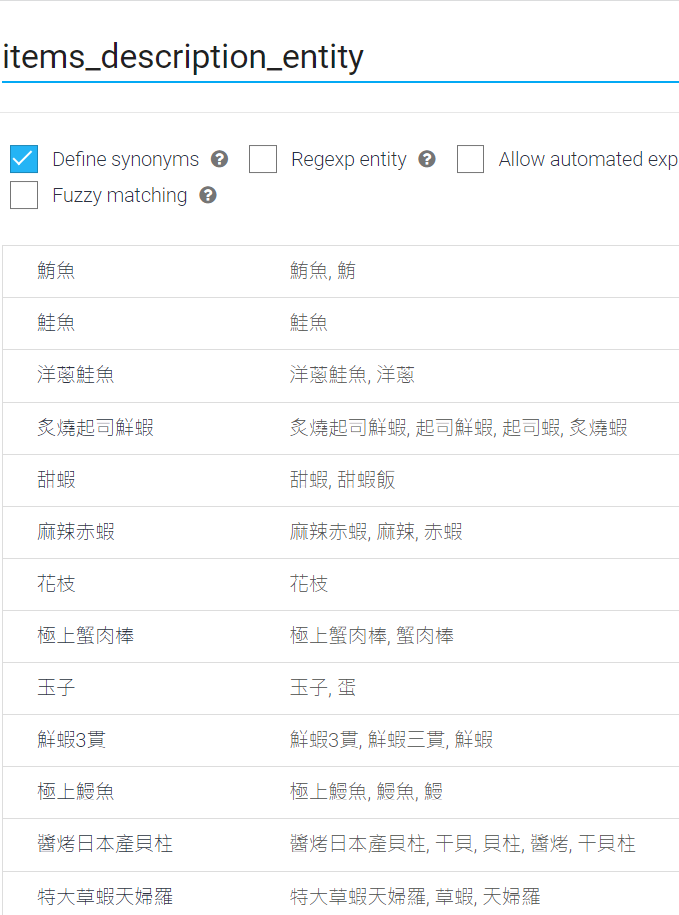
本系統設定了兩個Entity：

1. Quantity：

為系統預設的數量，包含一、二、三…以及1、2、3…等。

1. items\_description\_entity：

為自行設定的Entity，內容為壽司訂購APP中所有的商品品項，亦可設定同義詞，如圖三。



圖三 Entity設計

3.1.3 Context設計

本系統設定了四個Context：

1. add\_cart1: 需要幫您加到購物車嗎？ → 好的∕不用
2. add\_cart2: 還有其他的需求嗎？ → 就這樣了
3. end\_cart: 需要核對詳細資料嗎？ → 好的∕不用
4. verify: 是否送出訂單？ → 好的∕不用

3.1.4 Intent設計

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Intent name | User input | response | Input context | Output context |
| 1. Product\_content | 介紹一下鮪魚壽司。 | [1]，需要幫您加到購物車嗎？ |  | add\_cart1 |
| 1. Product\_new | 最近有什麼新品嗎？ | [2] ，需要幫您加到購物車嗎？ |  | add\_cart1 |
| 1. Product\_recommend | 你有什麼推薦的壽司嗎？ | [3] ，需要幫您加到購物車嗎？ |  | add\_cart1 |
| 1. add\_yes | 好的，一份。 | 好的，幫您加 [4] 到購物車。還有其他需求嗎？ | add\_cart1 | add\_cart2 |
| 1. add\_no | 不用 | 好的，還有其他需求嗎？ | add\_cart1 | add\_cart2 |
| 1. Add\_to\_cart | 我要一份鮪魚壽司。 | 好的，幫您加一份鮪魚壽司到購物車。還有其他需求嗎？ |  | add\_cart2 |
| 1. Add\_to\_cart\_end | 就這樣了。 | 好的，總共 [5] 元，要核對詳細資料嗎？ | add\_cart2 | end\_cart |
| 1. varify\_no | 不用 | 好的，訂單已經送出，謝謝購買。 | end\_cart |  |
| 1. verify\_yes | 好的 / 麻煩你 / 是 | [6]，是否送出訂單？ | end\_cart | verify |
| 1. Send\_yes | 好 | 好的，訂單已經送出，謝謝購買。 | verify |  |
| 1. Send\_no | 不要 | 好的，謝謝。 | verify |  |
| 1. Check\_checkout | 我要查看訂單。 | [7] |  |  |

表一 Intent設計

表一為本系統之Intent設計，本系統共有12個Intent。在User input的欄位中的紅字就是該Intent需要的Entity。

3.1.5 操作

1. Intent設定

這裡以Add\_to\_cart Intent為例。圖四為Intent的input/output context設定。

圖五為使用多個輸入範例訓練偵測Intent的過程，可以看到在Add\_to\_cart Intent需要quantity以及items\_description\_entity兩個Entity，所以若輸入的範例中沒有包含quantity，chatbot就會回覆：「好的，您要幾份呢？」；若輸入的範例中沒有包含items\_description\_entity，chatbot就會回覆：「好的，您要什麼食物品項呢？」

圖六為若使用者的輸入同時包含quantity以及items\_description\_entity兩個Entity，chatbot的回覆。

|  |
| --- |
| 圖四 Intent的input/output context |
| 圖五 |
| 圖六 chatbot的回覆 |

1. 啟用Dialogflow API

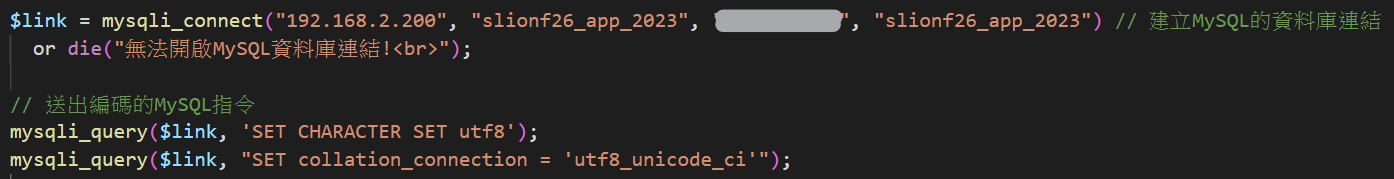
在Dialogflow中選擇Google Project → API與服務 → 啟用 Dialogflow API。

1. 建立服務與下載 credentials file

在Dialogflow中選擇Google Project → IAM管理 → 建立服務帳號 → 下載 credentials Json file。

3.2 Lionfree Server 開發

1. Lionfree資料庫連線

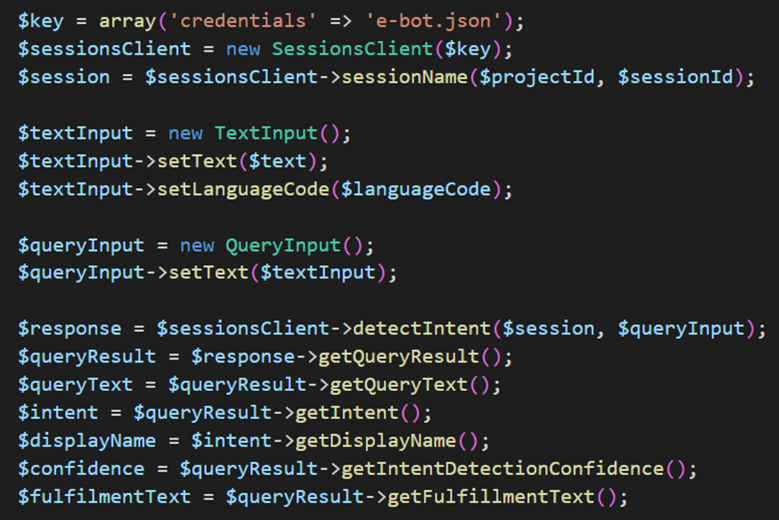


圖七 資料庫連線

1. Dialogflow PHP API

使用API提供的函式和類別來建立chatbot的客製化整合。並將Dialogflow整合到PHP中，如圖八。

圖八中的黃色框為連結credentials 檔案，紅色框為偵測User輸入的Intent。

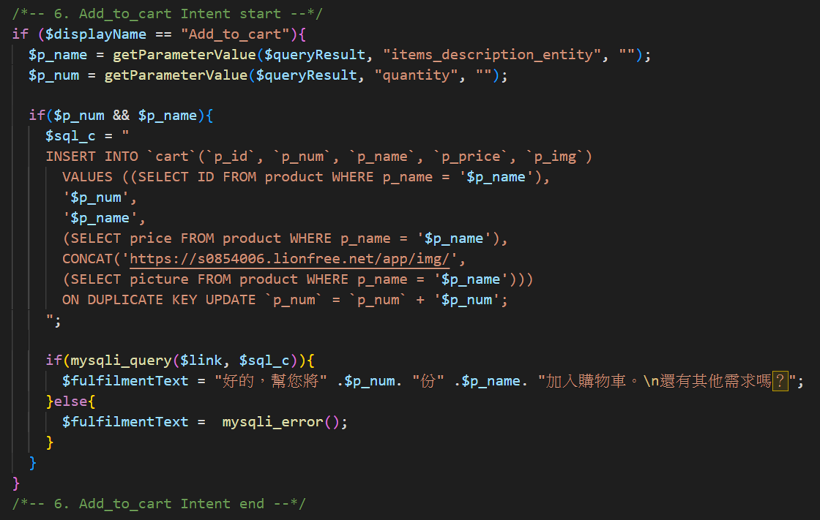


圖八 Dialogflow PHP API

1. 客製化整合

圖九為客製化整合的程式，以Add\_to\_cart Intent為例。首先透過Dialogflow判斷出此Intent為Add\_to\_cart，接著存取使用者輸入的兩個Entity (quantity以及items\_description\_entity)。

接著用SQL語法將商品名稱和購買數量寫入購物車資料表，若寫入成功就會將回應字串填入回應參數中。



圖九

3.3 APP 開發

在Android Studio建立對話框的Layout，並建立Adapter，把對話內容(使用者輸入以及GET到的)呈現到對話框中，成果如圖十。

一張含有 文字, 螢幕擷取畫面, 字型, 設計 的圖片

自動產生的描述

圖十 在APP中的chatbot

1. **成果展示**

|  |  |  |
| --- | --- | --- |
| 圖十一 **推薦商品功能**  (Product\_recommend Intent ) | 圖十二 add\_yes Intent | 圖十三 **推薦新品功能**  (Product\_new Intent ) |
| 圖十四 add\_no Intent | 一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片  自動產生的描述  圖十五 **介紹商品功能**  (Product\_content Intent ) | 圖十六 add\_yes Intent  (只有items\_description\_entity ) |

|  |  |  |
| --- | --- | --- |
| 圖十七 add\_yes Intent  (輸入了quantity entity ) | 圖十八 **加入購物車功能**  (Add\_to\_cart Intent ) | 圖十九 Add\_to\_cart\_end Intent |
| 圖二十 verify\_yes Intent | 圖二十一 Send\_yes Intent  以及 Check\_checkout Intent | 一張含有 文字, 螢幕擷取畫面, 網頁, 網站 的圖片  自動產生的描述  圖二十二 在APP中查看訂單  和chatbot回答的相同 |

1. **結論與心得**

5.1 遭遇問題與解決方式

一開始對於將Dialogflow結合到APP的架構感到很困擾，後來參考Google的資料才決定要採用 API/PHP/APP 這樣的三層設計。這個架構乍看之下似乎有些疊床架屋，但事實上是有其必要性的。

首先，PHP Server 是無法省略的，就算讓 APP 直接連結 Dialogflow ，我們仍然需要 PHP Server 做資料庫的維護。其次，如果不將 API 呼叫的程式寫在 PHP Server，則各個前端 (Android, Windows, iOS 等)需要重複API 呼叫的程式，而不是像現在這樣用簡單的HTTP Request，徒增開發以及維護的複雜度。最後，使用這種架構，credentials file只需存放在Server內，不需要存放在APP中，可以避免credentials file資料洩漏，這也是為什麼Google建議這種架構的原因。

5.2 心得

為了能將這次的專題和APP的專題結合，花了很多時間在思考題目，後來想到可以添加一個語音介面，方便使用者使用系統。也找到Dialogflow這個很好用的平台，學習到很多新的東西，收穫良多。

由於時間有限，目前只能完成核心，也就是自然語言處理的部分。希望之後有機會將前端（語音辨識）和後端（語音合成）補齊。