

南華大學科技學院資訊管理學系

碩士論文

Department of Information Management


College of Science and Technology

Nanhua University

Master Thesis

智能機器人點餐系統之研究

Research on Intelligent Robot Ordering System



魏宏恩

Hong-En Wei

指導教授：陳萌智 博士

Advisor: Meng-Zhi Chen, Ph.D.

中華民國 108 年 1 月

January 2019

南華大學  
科技學院資訊管理學系  
碩士學位論文

智能機器人點餐系統之研究  
Research on Intelligent Robot Ordering System

研究生：魏宏恩

經考試合格特此證明

口試委員：翁富美  
洪銘建  
陳麗智

指導教授：陳麗智

系主任(所長)：陳信良

口試日期：中華民國108年01月06日

南華大學碩士班研究生  
論文指導教授推薦函

資訊管理系碩士班 魏弘恩 君所提之論文

智能機器人點餐系統之研究

Research on Intelligent Robot Ordering System

係由本人指導撰述，同意提付審查。

指導教授 沈前智

108年1月10日

## 南華大學資訊管理學系碩士論文著作財產權同意書

立書人：魏宏恩之碩士畢業論文  
中文題目：智能機器人點餐系統之研究

英文題目：Research on Intelligent Robot Ordering System

指導教授：陳萌智 博士

學生與指導老師就本篇論文內容及資料其著作財產權歸屬如下：

- 共同享有著作權  
 共同享有著作權，學生願「拋棄」著作財產權  
 學生獨自享有著作財產權

學 生：魏宏恩 (請親自簽名)

指導老師：陳萌智 (請親自簽名)

中華民國 108 年 1 月 10 月

## 誌 謝

很感謝今年老師的指導，在以前從一開始還在迷航裡，大學畢業的我不知道從哪裡開始，陳萌智老師給了我許多建議，在老師建議下我來讀了碩士班，一開始完全不習慣，這是與大學生不一樣的生活，從剛開始時間都很緊湊，幾乎沒有自己的時間，但相對的我越來越覺得這種生活還滿充實的。大學時期完全不想寫程式的我，經過老師的指導也有了不一樣的想法，從完全不想碰到非常有成就感，老師的教育方式與其他老師不同，屬於會讓你自動自發的學習，從簡單的範例教起，讓學生有自己創作的空間以及基礎上的建立，讓我對於程式語言這塊領域不在害怕；在專案能力上，老師也帶領我們參加許多的大型展覽以及各種活動，讓我們有自己發揮的舞台，每場活動下來，雖然事前準備與過程非常累，但是也讓我學習到各種處理事情的態度以及狀況；在工作經驗方面，老師給予我很多機會，在課堂上擔任 TA，協助大學生上課以及課前準備，讓我能在學校就體驗到不同的事務處理。其他在學校社團上，參加資訊志工上山，協助偏鄉地區小朋友學習資訊課程，從大二就開始參與，在學校的課程壓力外，還持續有社團活動可以調劑，雖然整體的活動並不輕鬆，但是在最後營隊結束，

總是讓我想起那群小朋友，雖然他們很皮，但是非常有趣；團隊裡，大家也都相互分工合作，有爭吵、淚水與歡樂，這才造就今天的我們。最後就學過程也有一群好朋友陪伴，很感謝育嘉、偉哲、俊億、志仁、紀延、睿宇等等從大學就一起的朋友，在繁瑣事務裡，真的是有這群白癡的朋友才能度過，雖然平常都是在互相損對方，但是真的是有這群朋友才能度過的，最後要感謝在過程中幫助我的所有的學長姐、學弟妹、老師以及朋友，因為有你們的幫助我才能畢業，謝謝你們，給了我一個難忘的碩士生活。

魏宏恩 謹誌

中華民國一〇八年一月

# 智能機器人點餐系統之研究

學生：魏宏恩

指導教授：陳萌智

南 華 大 學 資 訊 管 理 學 系 碩 士 班

## 摘 要

科技的進步，機器人的發展越來越迅速，過去傳統機器人只用於單一性工作已慢慢被取代，智能型多功能機器人已慢慢被運用在各個領域，目前餐飲業競爭越來越激烈，餐飲業者為了競爭，服務品質也越來越高，服務品質的提高，需要更多的人力上的需求，也因此本研究希望透過智能機器人點餐系統的開發與設計，提供機器人減少店家人力與工時上的成本問題，並透過分析受測者需求與問題，歸納出機器人點餐系統之服務需求。

系統方面以對話代理人 DDE 建立餐點資料與對話設計，加以 ASP 伺服器端將資料進行存取，將餐點資料送到後臺店家顯示，最後建立 Zenbo SDK 與機器人間互動、對話以及帶位功能，建立一套完整的點餐系統。

研究方面採用服務體驗工程法中服務體驗洞察法架構，將觀察到活動、環境、互動、物件以及使用者五構面進行分析，並以互動、文化、序列、工具器物以及實體模型五個不同構面分析體驗洞察所蒐集資料，從中找出顧客的潛在需求。透過體驗訪談以及行為塑模分析結果如下：(1)建立簡易操作流程，(2)豐富機器人語句設計，(3)增加機器人的真實性與動作豐富性，(4)機器人帶位突發狀況以及速度改善。

關鍵詞：機器人、餐飲業、服務體驗工程

# Research on Intelligent Robot Ordering System

Student: Hong-En Wei

Advisor: Meng-Zhi Chen

Department of Information Management

The Graduated Program

Nan-Hua University

## ABSTRACT

With the advancement of technology, the development of robots is rapidly emerging. In the past, traditional robots with single-function natures have been slowly replaced; intelligent multi-functional robots have been used in various fields. At present, the competition in the catering industry is becoming fiercer. In order to stand out from the competition, the service industry has become more sophisticated, with improved quality that requires more manpower. As such, this research hopes to provide a solution that can reduce the required manpower and save time through the development and design of an intelligent robot ordering system and sum up the service needs of the robot ordering system by analyzing the needs and problems of the subjects.

In the system aspect, the dialogue agent DDE is used to establish meal information and dialogue design. The ASP server accesses the data and sends the meal data to the backend for shop display. Finally, interaction, dialogue, and seat guidance functions are added through Zenbo SDK to establish a complete ordering system.

In the research aspect, the service-experience insight structure in the service-experience engineering method is adopted, and the five aspects of activity, environment, interaction, object, and user are observed and analyzed. aspects of interaction, culture, sequence, tool artifacts, and physical model are used to analyze the collected information from experience and insights to learn the potential needs of customers. The results of experience interviews and behavioral modeling are as follows: (1) establishing a simple operation process, (2) enriching the design of robotic statements, (3) increasing the authenticity and action richness of the robot, and (4) improving the robotic emergencies and speeds.

Keywords: Robot, Catering Industry, Service Experience Engineering



# 目錄

論文指導教授推薦函.....	I
論文著作財產權同意書.....	II
誌謝.....	III
摘要.....	V
ABSTRACT.....	VI
目錄.....	VII
表目錄.....	IX
圖目錄.....	X
<b>第一章 緒論 .....</b>	<b>1</b>
第一節 研究背景與動機 .....	1
第二節 研究目的.....	2
第三節 研究範圍與對象 .....	3
<b>第二章 文獻探討 .....</b>	<b>3</b>
第一節 服務機器人.....	3
第二節 代理人.....	6
第三節 ZENBO 代理人 .....	9
第四節 服務體驗工程法 .....	10
<b>第三章 研究方法 .....</b>	<b>14</b>
第一節 脈絡洞察法 .....	14
第二節 研究對象與實施過程.....	15
第三節 研究設計.....	15
第四節 資料收集-脈絡洞察 .....	41
<b>第四章 結果與分析.....</b>	<b>45</b>
第一節 受訪者基本資料.....	45
第二節 受訪者服務體驗與訪談.....	46

第三節 五大行為模型.....	52
<b>第五章 結論與建議.....</b>	<b>55</b>
第一節 研究結論.....	55
第二節 研究建議.....	56
<b>第六章 參考文獻.....</b>	<b>56</b>



## 表目錄

表 2-1 FIPAACL 訊息結構.....	7
表 2-2 FIPAACL 溝通行為類型的描述.....	8
表 2-3 國內使用體驗工程法所研究的議題 .....	13
表 3-1 體驗觀察五大構面 .....	41
表 4-1 受測者基本資料 .....	45
表 4-2 A.E.L.O.U 五構面問題彙整.....	51
表 4-3 互動模型資料彙整 .....	52
表 4-4 序列模型資料彙整 .....	52
表 4-5 文化模型資料彙整 .....	53
表 4-6 工具模型資料彙整 .....	53
表 4-7 實體模型資料彙整 .....	54
表 4-8 五大模型問題與需求彙整 .....	54

## 圖目錄

圖 1-1 102 年~106 年我國餐飲業之銷售額(資料來源：財政部資料中心).....	2
圖 2-1 資料來源：引自 GIA 及本研究整理 .....	4
圖 2-2 FIPA 定義的抽象架構.....	7
圖 2-3 ZENBO 對話模型圖(取自華碩 ZENBO 官網資料) .....	10
圖 2-4 資料來源:資策會創新應用服務研究所(2008).....	11
圖 3-1 WEB 伺服器端系統架構.....	16
圖 3-2 本研究系統架構.....	17
圖 3-3 主餐菜單設計 .....	18
圖 3-4 點心菜單設計 .....	18
圖 3-5 飲料菜單設計 .....	19
圖 3-6 關鍵語句設計-1 .....	19
圖 3-7 關鍵語句設計-2 .....	20
圖 3-8 關鍵語句設計-3 .....	20
圖 3-9 關鍵語句設計-4 .....	21
圖 3-10 菜單與關鍵語句連結-1.....	21
圖 3-11 菜單與關鍵語句連結-2 .....	22
圖 3-12 菜單與關鍵語句連結-3 .....	22
圖 3-13 選取餐點 .....	23
圖 3-14 ZENBO 點餐介面程式碼-1 .....	24
圖 3-15 ZENBO 點餐介面程式碼-2 .....	24
圖 3-16 ZENBO 點餐介面程式碼-3 .....	25
圖 3-17 ZENBO 點餐介面程式碼-4 .....	25
圖 3-18 ZENBO 功能設計程式碼-1 .....	26
圖 3-19 ZENBO 功能設計程式碼-2 .....	26
圖 3-20 ZENBO 功能設計程式碼-3 .....	27
圖 3-21 ZENBO 功能設計程式碼-4.....	27
圖 3-22 ZENBO 功能設計程式碼-5 .....	28

圖 3-23 ZENBO 功能設計程式碼-6 .....	28
圖 3-24 ZENBO 功能設計程式碼-7 .....	29
圖 3-25 ZENBO 功能設計程式碼-8 .....	29
圖 3-26 ZENBO 功能設計程式碼-9 .....	30
圖 3-27 ZENBO 功能設計程式碼-10 .....	30
圖 3-28 ZENBO 功能設計程式碼-11.....	31
圖 3-29 ZENBO 功能設計程式碼-12 .....	31
圖 3-30 進行帶位動作 .....	32
圖 3-31 餐點菜單 .....	33
圖 3-32 目前點選餐點 .....	33
圖 3-33 確認最後餐點 .....	34
圖 3-34 點餐完成 .....	34
圖 3-35 ASP.NET 程式設計-1.....	35
圖 3-36 ASP.NET 程式設計-2.....	35
圖 3-37 ASP.NET 程式設計-3.....	36
圖 3-38 ASP.NET 程式設計-4.....	36
圖 3-39 ASP.NET 程式設計-5.....	37
圖 3-40 ASP.NET 程式設計-6.....	37
圖 3-41 ASP.NET 程式設計-7.....	38
圖 3-42 ASP.NET 程式設計-8.....	38
圖 3-43 ASP.NET 程式設計-9 .....	39
圖 3-44 ASP.NET 程式設計-10.....	39
圖 3-45 ASP.NET 程式設計-11.....	40
圖 3-46 後臺餐點接收 .....	40
圖 3-47 ZENBO 點餐機器人體驗經驗框架.....	42
圖 3-48 ZENBO 點餐機器人互動模型.....	43
圖 3-49 ZENBO 點餐機器人序列模型.....	44

# 第一章 緒論

## 第一節 研究背景與動機

隨著科技的進步以及企業與國家開始投入研究人工智慧，過去機器人性質較為單一，只能做一些重複性高單一性工作，現在由於技術漸漸成熟，開始能讓機器人完成較複雜性的大數據分析之工作，於 2018 年 IFR (國際機器人聯邦，International Federation of Robotics) 報告指出：專業用途服務機器人的銷售額增長了 39%，達到 66 億美元。該機器人類別的總銷量增長了 85% (2017 年)，證明了機器人已漸漸進入人類的生活內。在 2015 年天下雜誌中提到，全球人口漸漸老化、成長速度趨緩以及加上亞洲廉價勞動力逐漸消失，沒有掌握機器人開發的國家，競爭力將快速下降。

餐飲業作為國家的民生必需產業，擁有良好的餐飲業發展能反映出國家發展程度及人民生活品質，作為滿足人類生理需求的基礎行業，具有進入門檻低及勞力密集之特性，此產業進入並不需要科技技術及鉅額資金等需求，也因此產業內廠商數量眾多。根據 2018 臺灣趨勢研究(TTR)數據顯示，餐飲業銷售額方面，雖然近 5 年是呈現逐年成長趨勢，自 102 年的 3,749 億元，成長至 106 年的 5,163 億元，每年成長率在 6% 至 10% 之間，其中 103 年及 105 年成長率皆達到 9.3%，但在 106 年間成長慢慢趨緩，增長 6.8% (如下圖 1-1)，這顯示出餐飲業正面臨產業趨近飽和狀態，目前餐飲業者入門門檻低，也因此造成餐飲業者眾多，在這個接近飽和狀態下，競爭只會越來越激烈，除了競爭激烈外，餐飲業者也因食物價格波動大以及人員流動率高等情況，造成許多新進餐飲業無法生存，也因此如何控管成本成為餐飲業經營的關鍵。

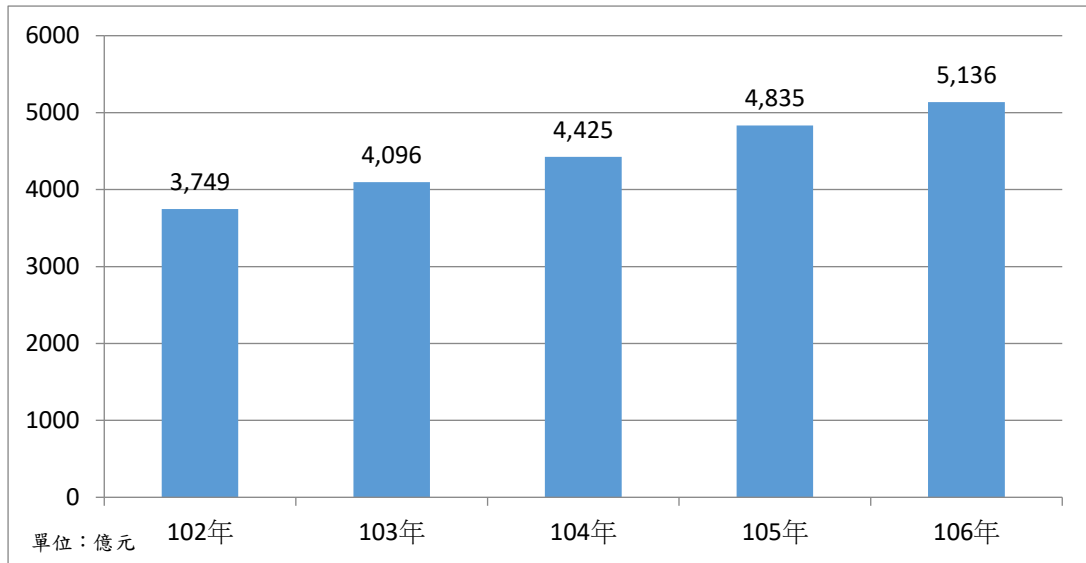


圖 1-1 102 年~106 年我國餐飲業之銷售額(資料來源：財政部資料中心，2018)

隨著機器人時代來臨，本研究有感於目前餐飲業所面臨問題，在如此競爭的餐飲市場下，尤其是服務性質的工作，需要較多人力投入，透過設計機器人點餐系統，讓機器人來解決目前人力的老化造成的工作人口的減少，對於店家而言，機器人也較不會有工時以及流動等問題發生，藉由大數據與物聯網發展，收集顧客的購買資料，利用機器人預測銷售情況及人與機器人協同工作，來提高工作生產力。

## 第二節 研究目的

在如此競爭的餐飲業中，各家餐飲業者的產品以及服務品質就成為吸引消費者的主因，為了提高服務品質，根據 Rehman (2011)學者提出，良好的人力資源管理措施會直接影響顧客所認知的服務品質，因此本研究以機器人點餐系統為研究主題，透過機器人的優勢統一管理以及高勞動力特性，以建立更好的服務品質。

為了探討機器人點餐系統設計的服務需求，以有使用過 APP 應用程式的年輕族群為測驗對象，並以一次點餐作為任務條件，再引用服務體驗工程法為理論基礎，以活動(Activities)、環境(Environments)、互動(Interactions)、物件(Objects)、使用者(Users)五構面深入訪談與觀察(蕭淑玲等，2010)，深入探討顧客行為中隱藏的意義，並歸納出五大彙整模型，模擬出使用者面臨環境與內、外在因素的問

題與狀況，從中歸納出機器人點餐系統之服務需求。

### 第三節 研究範圍與對象

本研究採服務體驗脈絡洞察法，隨機挑選同樣條件有使用 APP 應用程式的 10 名 18~25 歲年輕族群為研究對象。

## 第二章 文獻探討

### 第一節 服務機器人

#### 一、服務型機器人之發展

人類的生活物質需求越來越高，為求方便性投入許多成本研究機器人，隨者科技的進步，目前服務型機器人漸漸融入到餐飲、醫療、教育、娛樂、清潔、導覽等多項領域，服務型機器人經由協助人類的生活便利，在各個行業提升了精確度及可信度，也漸漸改變人類的生活習慣。

#### 二、智慧型機器人的定義

很早之前機器人都只是人類的憑空想像，後來人類的想像越來越豐富，終於機器人一開始出現於小說中 1921 年，捷克的科幻小說家卡佩克(Karel Capek)所寫 Rossums's Universal Robots 《中譯：羅素姆的萬能機器人》書中出現，原文為「Robot」(溫九力，2016)，科技也是就是如此而來的，一切都是始於人類的需求與想像。

智慧機器人(Intelligent robot)從字面上的意思就是包含了「智慧」為人工智慧和「服務」為硬體(機械結構與電子元件)的組合。智慧又可以稱「人工智慧」，為可以事先設定邏輯參數自動回應(Reacting)，並透過自動感應偵測(Sensing)使機器人感測外在環境動態狀態變化，後續機器人能夠自行整理(Processing)、歸納及判斷(Reasoning)，最後透過先前設定參數來調整系統作業的能力，不需要其他人為輔助。根據國際智慧機器人協會(International Federation of Robotics 簡稱 IFR)定義，智慧型機器人為可以替人類以及設備有效率工作，並能以半自主或全自主運行作業。

根據行政院在 2005 年召開「智慧型自動化產業策略規劃會議(SRB, Strategy Review Board)」當時將智慧型機器人定義為「能夠多功能進行多軸自動或半自動



機械裝置，它能夠利用程式化執行各種活動，也能結合人工智慧和感測技術的應用，具備跟人互動功能且提供人們生活、健康、安全、娛樂等各種服務，未來將成為人類忠實及信賴的同伴」。

### 三、服務型機器人分類應用

服務型機器人的應用領域，隨著科技的進步，許多類型機器人的出現，每個對服務機器人的分類都大不相同，參考 GIA(Global Industry Analysts, Inc., 2013) 之分類，將服務型機器人分成個人化服務型以及專用型兩大類，並依據其功能和應用分為各種相關的用途，如圖 2-1

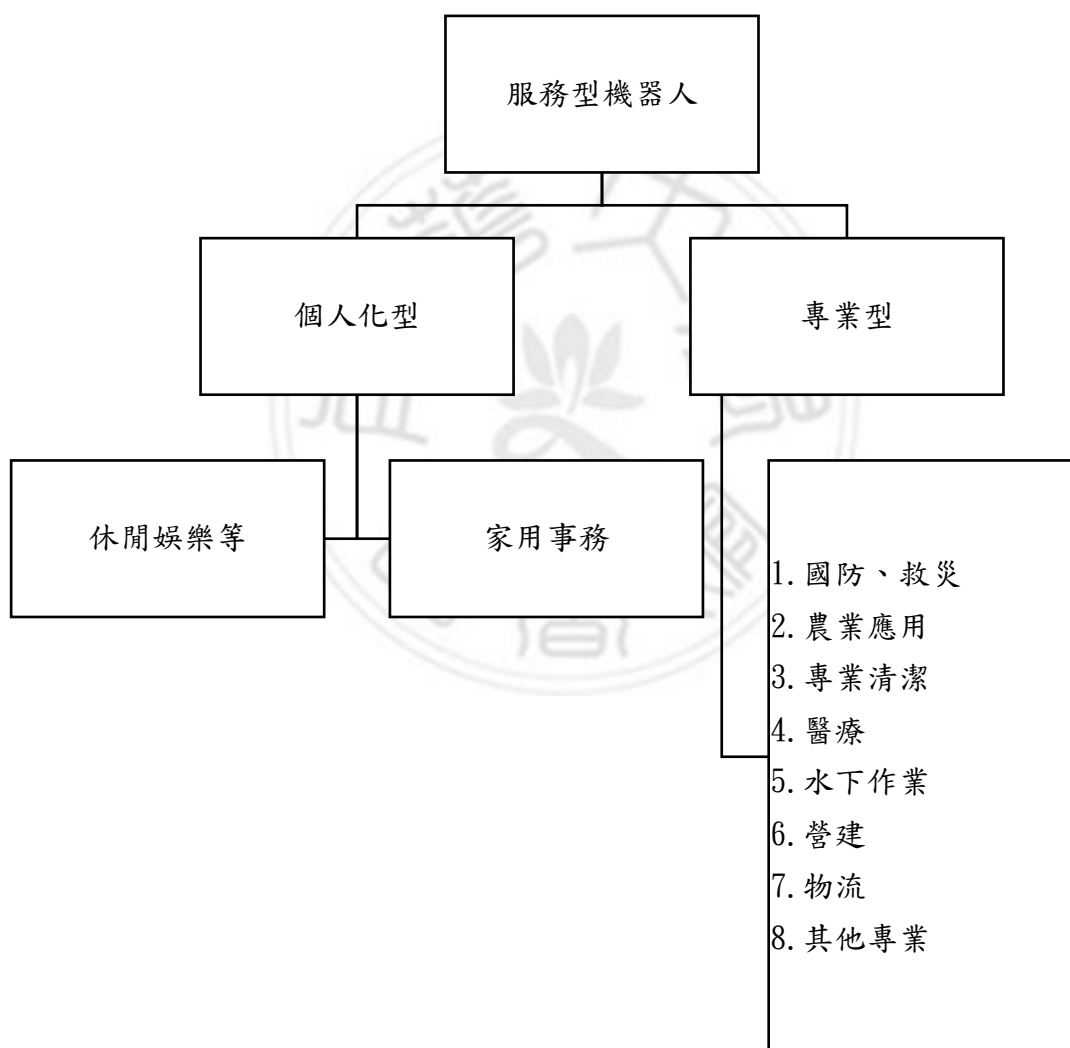


圖 2-1 資料來源：引自 GIA(2013)及本研究整理

依據財團法人精密機械研究發展中心(2011)於「2011 智慧機器人產業分析報告」中指出，GIA 把專用服務型機器人區分成國防和救災、農用、專業清潔、

醫療、水下作業、營建、物流與其他專業領域八大類：

1. 國防及救災機器人(Defense, Rescue and Security Robots):可以分為地面、水下作業及空中 UAV 等，主要運用在戰場以及救災應用，其目的為避免人類的傷亡，如地雷(水雷)探測機器人、反恐與防爆機器人、小型偵察機器人、無人駕駛機器人、太空探測型機器人等。

2. 農業應用機器人(Field Robots):目前最大應用是擠牛、羊奶，其他如農作物的採收、摘果、噴灑農藥、蔬果嫁接、除草與伐木等第一級農業應用。

3. 專業清潔機器人(Cleaning Robots):應用於非家庭內之特殊清潔專業用途，主要包括了大樓窗戶或外牆清潔、工廠樓板清潔、公共場所清潔、油槽清洗、管路清洗...等相關專業清潔用途。

4. 醫療機器人(Medical Robots):以協助提供醫療和照護相關行為，包括了外科手術輔助、復健支援、雷射治療、電動代步輔具、健康照護和遠距醫療等應用。

5. 水下作業系統(Underwater Systems):用於在水域之石油或天然氣探勘以及開採作業。人類在水下作業具高度不確定及危險性，故為了避免人類傷亡，大多指派機器人進行水下任務。

6. 營建用機器人(Construction & Demolition Robots):主要應用於高度危險或高勞力需求之建築構工、施工物料搬運以及建物拆除作業，例如：汽車模板建置工業機器人(Decker, M., Fischer, M., Ott, I, 2017)。

7. 物流機器人(Logistic Robotics):為節省貨物物流過程人力，以及提高貨物處理精確性和速度，而主要應用在貨品裝卸、堆疊、包裝、分揀、搬運及交付過程，因此能夠提高倉儲和配送中心作業的效率。

8. 其他專業用途機器人(Others):科技的進步，以 SLAM 技術應用於室內機器人(Cadena, et al., 2016)的身上已普及，常見的掃地機器人(Li, Li, Mak, & Tang, 2016; ASUS, 2017)就是以此技術高效率仿造出室內的地圖(Belter, et al., 2016)，還有居家服務型機器人，其具備有移動、觀看、說話、聆聽、音效、連接、學習與等功能，其他專業用途的機器人有採礦、管路探勘、下水道工作、巡邏、保全、消防救援、挖掘救災、無人搬運載具、導覽等類型機器人，應用領域將隨著科技創新與研發持續發展。

## 第二節 代理人

### 一、代理人定義

在科技的進步下，身處在資訊爆炸的時代，越來越多複雜的系統軟體出現，替換傳統的資訊系統，以代替人類處理更多更複雜的事，例如：機器人作為專業判斷之預測、評估、決策、模擬人行為思考模式、處理非結構化資料以及自主完成任務等。

代理人一詞源自人工智慧領域，目前人工智慧發展迅速，後因人工智慧領域發展快速，而衍生出多代理人系統(Multi-Agent System, MAS)、行動代理人系統(Mobile Agent)、代理人導向之程式設計(Agent-oriented Programming, AOP)等研究，其最新定義為替代使用者完成某項特定任務之統合型系統(卓人杰，2011)。

代理人起源人工智慧領域，主要目的為幫助人類完成特定活動之整合系統。Russell, & Norvig(2003)兩位學者認為代理人應該可以透過系統自主性、互動性、適應性等模組，且應具備認知和行為能力。Wooldridge, Jennings, & Kinny (2000)等學者更進一步提出代理人系統應擁有(1)反應能力：指代理人能夠感知外在環境變化給予及時回應能力；(2)自發能力：能透過其知識主動進行判斷、推理、學習；(3)社交能力：代理人能和社群其他代理人進行協商合作。代理人間的溝通以 FIPA (Foundation of Intelligent Physical Agents) 制定的代理人溝通語言(Agent Communication Language, ACL)達成。FIPA 為一個非營利的代理人技術標準化組織 (FIPA, 2002; Panti, et al., 2000)，FIPA 制定在異質平台代理人通訊的統一標準，在代理人平台定義主要四種元件，(1)訊息傳遞(Message Transport)；(2)代理人目錄(Agent Directory)；(3)服務目錄(Service Directory)；(4)代理人溝通語言(ACL)。代理人溝通語言即是 FIPA 制定為統一規定的溝通協議，其中包含知識本體(Ontology)、溝通語言(Language)、溝通協定(Protocol)，如圖 2-2 所示。

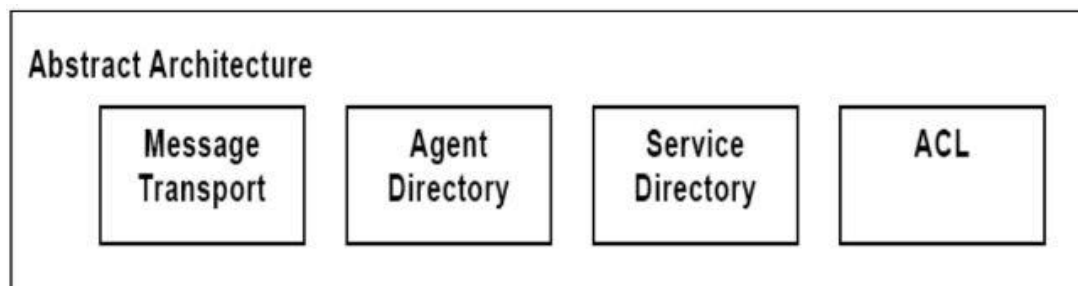


圖 2-2 FIPA 定義的抽象架構(FIPA, 2002; Panti, et al., 2000)

## 二、訊息結構

代理人可由 FIPA 制定代理人所用溝通語言(Agent Communication Language, ACL)與其他平台或不同程式語法間進行資訊間的交換,跟一般程式模式差異為,代理人透過彼此溝通,建立更好的溝通模式,並可以透過代理人了解雙方的需求與主要目的,以下為 FIPAACL 訊息結構,透過描述該次溝通的主要需求。其定義各種類型用以描述語句訊息的結構如下表 2-1 所示。

表 2-1 FIPAACL 訊息結構(盧賢豪, 2014)

參數	類型	說明
Performative	溝通	為 Agent 欲進行的動作
REQUEST	建議	建議協商並請求訊息接收者提供資料
Sender	參與者	寄件端代理人的名稱
Receiver	參與者	接收端代理人的名稱
Content	訊息內容	訊息的內容
Language	內容描述	訊息內容使用溝通語言
Ontology	內容描述	訊息內容使用 Ontology
Encoding	內容描述	訊息內容的編碼格式
Protocol	對話控制	發送訊息的通訊協定
Conversation-id	對話控制	多代理人溝通時,用來辨識交談的對象
Reply-with	對話控制	當回覆時用來識別此訊息的表示
n-reply-to	對話控制	用來回覆先前的溝通時的訊息識別
Reply-by	對話控制	用來註明最後應收到回覆的日期或時間

利用 FIPA 定義之代理人溝通言語形式,程式設計師可在建立代理人溝通訊息時,可具備各種效果以及生動方式呈現每個代理人傳達的訊息溝通間語意(林聖熙, 2010),以下為程式設計查用協定做簡單溝通描述如下表 2-2 所示。

表 2-2 FIPAACL 溝通行為類型的描述(林聖熙，2010)

定義	類型	說明
ACCEPT_PROPOSAL	建議協商	允許接受其他 Agent 的提議
AGREE	工作發佈	接受一個 Agent 的 Request
CANCEL	工作發佈	結束該 Agent 的請求
CONFIRM	建議協商	對提議不確定時，通知接收者先前所提是正確
DISCONFIRM	服務資訊	對提議不確定時，通知接收者先前所提是正確
DISCONFIRM	服務資訊	對提議不確定時，通知接收者先前所提是不正確
FAILURE	錯誤服務	試圖執行某動作失敗
INFORM	服務資訊	傳送訊息告知另外一個 Agent
INFORM_IF	服務資訊	有條件的傳送訊息告知另外一個 Agent
INFORM_REF	服務資訊	參照傳送訊息告知另外一個 Agent
NOT_UNDERSTOOD	錯誤服務	無法理解所要執行的活動
PROPOSE	建議協商	回應協商訊息給起始者 Agent
QUERY_IF	查詢服務	有條件的查詢某服務
QUERY_REF	查詢服務	查詢參照的服務
REFUSE	拒絕服務	協商結果不滿意發出訊息通知
REJECT_PROPOSAL	拒絕服務	協商回應者的提議被拒絕時發出訊息
REQUEST	請求服務	請求該 Agent 執行某動作
REQUEST_WHEN	請求服務	有條件的 Agent 被請求某動作
REQUEST_WHENEVE R	請求服務	特定時間條件下的該 Agent 執行某動作
SUBSCRIBE	查詢服務	向工具代理人(Utility Agents)註冊
PROXY	服務資訊	將訊息中的子訊息傳給指定的代理人
PROPAGATE	服務資訊	告知所有 Agent
UNKNOWN	服務資訊	未知的訊息

### 第三節 ZENBO 代理人

DDE(對話開發環境)編輯器是個對話開發平台，本研究透過此平台設計，測試和發布自己的腳本對話，智能代理和 Android 應用程式之間的對話，以便 DDE 代理人與 Zenbo 點餐系統程式進行交互。其目的是簡化機器人或其他智能代理應用程式創建和建立對話的過程。

#### 一、開發者：

1. 開發人員向 DS(Dialogue System)編輯器添加交叉意圖，例如：拍照，用編輯軟件包名稱以啟動應用程式。

2. 使用 DS 編輯器定義應用程式操作，以便可以使用語音命令來控制應用程式。

3. 通過使用 Zenbo SDK 開發應用程式，並包含子類動作，視覺，機器人和實用程序。

#### 二、使用者：

1. 使用語音命令 Hey Zenbo 呼叫機器人，然後說出交叉意圖(例如拍攝照片)。

2. DS 系統將語音命令發送到雲端 Server 並返回口語理解模組 (Spoken Language Understanding / SLU)，將語音辨識資訊轉化為結構化的語意資料，並提供給正確的 Domain Name 處理，使它能了解話中的 Intent(意圖)。

3. 機器人通過解析 SLU 結果來打開應用程式。

4. 應用程式通過使用 Zenbo SDK 接收在 DS 編輯器中定義的應用程式內 SLU 結果並執行相應的操作。

5. APP 可以通過 Zenbo SDK 開啟機器人的功能。

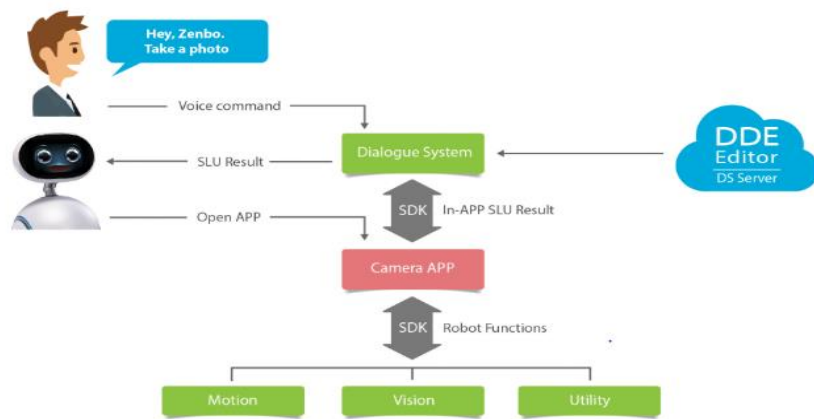


圖 2-3 Zenbo 對話模型圖(取自華碩 Zenbo 官網資料)

## 第四節 服務體驗工程法

### 一、服務體驗工程方法論

「服務體驗工程法」(Service experience engineering, S.E.E.)經由資訊工業策進會提出的方法論，該會有感於台灣產業對於服務創新沒有合適方法論或工具作為規範，資策會於是整合服務、應用、平台三大中心和 12 團隊研發的輔導經驗，採擷德國工研院 IAO 發展 41 種方法的服務工程方法論及美國 IDEO 設計公司在顧客體驗洞察中所研發 51 種洞察方法之各方專長，和義大利學者 Roberta 在服務設計 40 個工具，結合辛辛那提大學教授、美國智能維修系統研究中心(IMS Center)主任與上海交通大學產業技術研究院院長李傑博士開發的創新矩陣服務發想工具，創造適合台灣企業能使用的服務設計理論。服務體驗工程從農業時代到工業時代，最後到服務經濟的時代，目前進入體驗經濟的時代，將服務與體驗合併以顧客需求作為基礎加入創新的概念，為服務體驗工程。服務體驗工程以有創新發想及提出新穎的想法為基礎，但有還是有許多創新想法沒辦法實現，此工程目的是為了將創新的想法實踐，如何從一開始想法到加以實踐間之過程為服務體驗工程的重點(參考創新電子報，2013)。

服務體驗工程為服務與研發工作的流程模式方法論，整體模式主要三大階段(Phase)，分別是趨勢研究(FIND)、服務價值鏈研究(InnoNet)和服務實驗(Design Lab)。而服務體驗工程方法論在此模式架構下把創新服務發展所需要研究的之面

向及可使用的工具與方法進行系統性統整，以下為服務體驗工程整體流程圖：

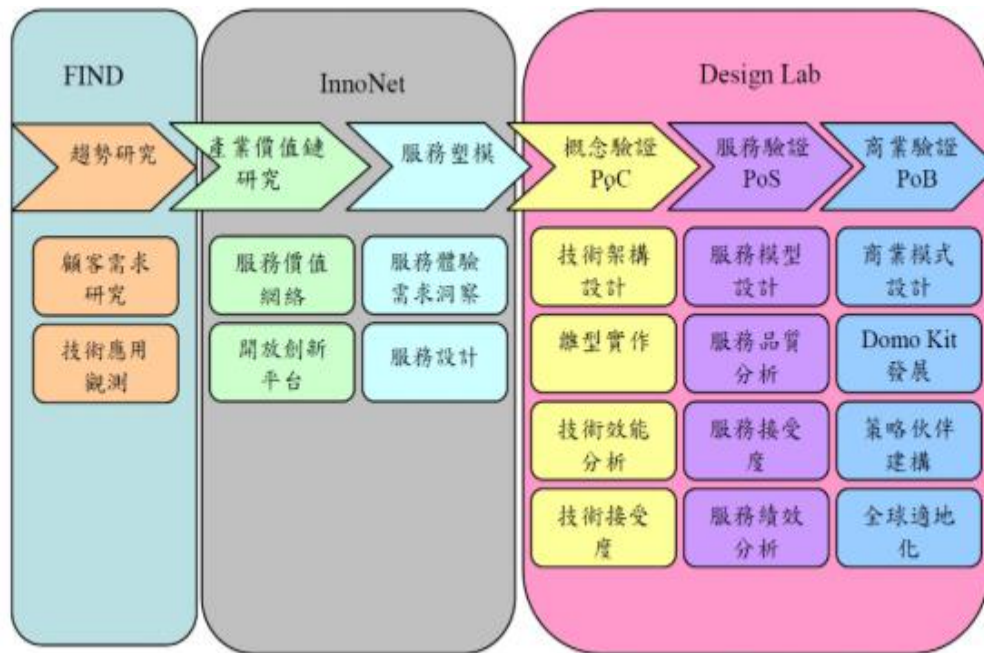


圖 2-4 資料來源:資策會創新應用服務研究所(2008)

(一)趨勢研究：

一個創新服務的發展，一開始工作為做創意蒐集與分析的研究。實作經驗顯示，一般企業擁有很多的新服務的創意，但常常遇到困難的地方其實是執行層面。如何系統化的尋找、蒐集並開始研發創意為最困難的地方。服務體驗工程方法幫助發展新服務之企業執行系統化的創意蒐集與研究。服務體驗工程方法論中第一個階段為 FIND 階段，從研究大環境趨勢的發展，尋找出消費者需求或潛在的商機。此階段主要工作為利用研究資訊技術的發展趨勢或舊有研究資料等大環境趨勢變化，進行新穎服務的創意蒐集，並加以具體化以及評估過濾的過程。這個階段的最主要目的是決定一個新服務的創新可行性。節省新服務的研發成本，並確認新服務的市場之接受度，須要針對新創意可行性以及市場潛力提早做出評估與調查。FIND 階段研究結果為產出成功率高、可行性高的新服務創意。然後就是進入創新服務的研發工作(王熙哲，2010)。

(二)服務價值鏈研究：

此階段分成產業價值鏈研究以及服務塑模兩個程序。服務是經由許多企業提供各種不同服務所形成，並由特定業者整合成價值網絡。產業價值鏈研究的主要目的在於，界定創新服務產業所覆蓋的企業，完成創新服務的雛型，當作服務



塑模階段時參考的依據。在服務塑模部分有兩個主要工作，經由服務體驗洞察，尋找使用者隱藏的真實需求，並利用系統化的服務設計將服務模型與藍圖具體呈現出來。

### (三)服務實驗：

創新服務進入市場前，須經過測試，免得上市後才發現問題重重，造成資源浪費並影響企業形象。SEE 提出「生活實驗室」創新系統，用實際生活環境作為場景，研究受試者的參與、使用狀況，測試服務設計成果，進行必需的調整及改良。服務實驗的工作，可分成概念驗證（Proof of Concept，POC）、服務驗證（Proof of Service，POS）及商業驗證（Proof of Business，POB），驗證某服務概念的可行性與商業價值(參考創新電子報，2013)。

## 二、服務體驗脈絡洞察

服務體驗脈絡洞察是直接進入使用者實際的情境脈絡中，採用最直接的觀察及訪談方式與使用者互動，並以系統化的模型工具，分析使用者的原始研究資料，幫助研究者瞭解特定領域的使用者行為模式（蕭淑玲等，2010）。進行服務體驗脈絡洞察要選擇多少個具有代表性的研究樣本？樣本的數量通常是根據想要解決的問題決定，當問題的範圍越狹窄或是研究樣本間的工作及環境越一致，那麼樣本所需的數量就可以越少。依 Beyer 與 Holzblatt 之建議，在執行實務上，研究樣本以 10 到 15 名較為適合（蕭淑玲等，2010）。服務體驗脈絡洞察的進行可分為「界定議題與洞察目標」、「規劃洞察計畫」、「進行現場洞察」與「洞察結果的解讀與呈現」四個階段（資策會，2008）。

## 三、體驗訪談

體驗訪談主要是依據活動(Activities)、環境(Environments)、互動(Interactions)、物件(Objects)、使用者(Users)五構面作為訪談內容(蕭淑玲等，2010)，接著本研究想透過訪談過程中結合參與式跟非參與式觀察法，採用體驗經驗框架讓使用者能邊操作邊接受訪問，以下為訪談法說明。

### (一)體驗旅遊框架：

此觀察訪談法主要先界定使用機器人點餐系統過程中每個重要階段活動，事先分為好多個特定框架，並預先設想被觀察者在這些活動接觸點中會有哪些行為或活動。後在這些重要的接觸點裡，尋找出使用者對本系統的隱藏需求或服務

潛在失效點，然後記錄使用者在體驗過程中行為模式。

(二)行為塑模：

運用五個圖形來表達所用之方法，會被文字表現更加具體。五大模型互動(Flow)、序列(Sequence)、工具(Artifact)、文字(Cultural)以及實體(Physical)模型(資策會，2008)。

(三)過往文獻

根據過往文獻顯示，服務體驗工程法中的脈絡洞察，能夠協助本研究，了解顧客需求以及行為模式，所以本研究以脈絡洞察為研究方法，並透過包含體驗觀察、訪談，最後藉由行為塑模來整理出潛在需求列表。以下是國內使用體驗工程法所研究的議題：

表 2-3 國內使用體驗工程法所研究的議題

年份	作者	題目	研究內容
2010	林曉琪	應用服務體驗工程方法於銀髮族家事服務系統設計	此研究以服務體驗工程為主要方法，從不同角度了解居家照護產業中，透過實際的供給與需求的情況，且更深入了解銀髮族潛在的需求，改善原有服務，並加以更進一步提出創新的服務。
2011	彭康孫	服務體驗工程與層級結構分析法解析虛擬社群網站服務與使用者需求間關係之研究	此研究選擇以「虛擬社群」類型網站為基礎，利用「服務體驗工程」(Service Experience Engineering, SEE)中「InnoNet」之服務解析方法與層級結構分析法(Alytic ierarchyProcess,AHP)，以最新方法對網站服務和使用者關係進行研究。
2012	林鶴樺	運用服務體驗工程方法導入雲端資訊應用系統-以ERP為例	運用服務體驗的方法建立起導入雲端ERP系統主要架構的準則，提出雲端ERP系統的服務架構與導入模式，來協助企業建構出雲端ERP的相關決策，並實際建立一個雲端ERP系統，加以介紹雲端系統的優點與功用。
2013	謝伊萍	智慧型販賣機之服務設計-服務體驗工程方法之應用	透過服務體驗方法(S.E.E.)，此研究初步的完成了個案公司-T公司智慧型販賣機的服務設計；並展現出科學服務、管理、工程與設計(SSMED)之產學合作成效。

2014	洪素蜜	應用服務體驗洞察法於銀髮族手杖使用之功能需求探討	此研究透過「服務體驗工程方法」作為理論架構，從使用者觀點以體驗觀察及訪談的方式實行服務體驗工程中之脈絡洞察，再以五大模型：互動模型、工具器物模型、文化模型、實體模型，針對銀髮族手杖使用的潛在需求進行分析。
2016	荇蒞宓 啟茵	服務體驗工程方法應用於部落文化健康站餐飲服務之研究	此研究透過服務體驗工程評估此計畫之建置，深入了解使用者體驗並感受之本質，以及檢視目前之服務與功能，找出部落老人使用者現有需求及隱藏的服務契機。
2018	蘇政勛	比較以使用者經驗方法與服務體驗工程之設計流程與產品差異，以電動捲線器為例	以電動捲線器為例子，透過「服務體驗工程」創造新的服務網絡、服務模型，對未來產品設計流程作為應用探討。也利用此研究之需求分析、產業價值鍊、服務塑模、驗證與對比舊有電動捲線器之差異的過程，建立產品以服務體驗工程法之研究的架構上作為基礎，並試著在產品設計和創新服務之間取得平衡，並導出產品的服務設計方法與流程。

### 第三章 研究方法

本研究在確立研究主題後，著手蒐集相關文獻，根據過往研究者發現服務體驗工程中的服務體驗脈絡洞察法可以找出現有使用者的需求以及隱藏的服務契機，藉此找出本系統「Zenbo 點餐」功能：對話代理人、前台點餐設計和 ASP 伺服器端服務模式間的優劣點，以提高顧客對系統之滿意度，再以五大模型做為分析工具，五大行為是透過五種不同面向去觀察，包括互動模型、序列模型、文化模型、工具器物模型和實體模型等，透過行為統整後利用圖形的方式表達，經過分析將使用者潛在需求表達出來，以此提出 Zenbo 點餐系統之服務與功能需求，感受使用者實際的情境脈絡。

#### 第一節 脈絡洞察法

脈絡洞察法是藉由直接進入使用者實際體驗場域中，此方法包含體驗觀察、體驗訪談以及體驗分析，其中體驗觀察採用非參與和參與式方法，了解使用者在真實情境最直接行為和使用者在系統服務模式中，特殊行為與動作，體驗訪談主要是透過使用者操作過程中直接參與，在使用者操作每一步驟過程直接訪談與觀察以了解每個環節使用者的真實狀況，

透過前面訪談以及觀察資料蒐集，並利用互動模型、序列模型、工具器物模型、文化模型及實體模型五大行為塑模方式，說明單一使用者在進行特定步驟或活動的行為模式及需求；為了掌握每個使用者共同的行為模式，因此集結全部研究對象的行為模型整合成「彙整行為模型」，以完整呈現「Zenbo 點餐系統」整體使用者行為模式和需求。

## 第二節 研究對象與實施過程

根據過往研究者 Beyer 以及 Holzblatt 建議，具有代表性研究樣本建議在 15 到 20 名，但是樣本的數量常常依據所想解決之問題來決定，問題範圍較狹窄或研究樣本間的性質較相近時，那麼樣本就可以越少，可以降至 10 到 15(蕭淑伶等，2010)，也因此本研究徵求了 10 位 18 歲到 25 歲年輕族群，這些族群所屬工作性質以及環境較為一致。

由使用者操作 Zenbo 點餐系統，模擬機器人與人類互動之過程，每個使用者以點一份餐點作為任務，並在正式進行研究前，事前徵詢受訪者的意願，同意讓研究者在旁邊進行觀察、訪談並說明其詳細研究進行方式。

## 第三節 研究設計

### 一、系統設計

本研究系統所運用之軟體與設計方法如下：

#### (一) Dialogue Development Environment(DDE)建立

透過 Zenbo 官網上 DDE(對話開發環境)，測試和發布自己的腳本對話，建立智能代理人和 Android 應用程序之間的對話，以便應用程序可以與代理人機器人進行交流。

#### (二) Access 資料庫建立

透過建立餐點資料庫，將餐點分類，分為主食餐點、點心餐點、飲料餐點等，並建立顧客購買資料，讓顧客購買歷史能提供店家後續使用。

### (三) Visual Studio Web 伺服器端建立

前端點餐資訊建立後，利用伺服端的建立，設定網際網路資訊服務(IIS)，允許資料庫存取權限，完成點餐，並回傳餐點資訊以及價格，以供顧客再次確認，如下圖 3-1 所示。

### (四) Zenbo SDK Android Studio 建立

以 DDE(對話開發環境)連接 Zenbo SDK，並建立對話事件，讓 Zenbo 能在點餐時做出臉部表情以及動作，模擬虛擬社交狀態，以拉近與顧客間關係，並透過 WebView 元件，與伺服器端進行連線。

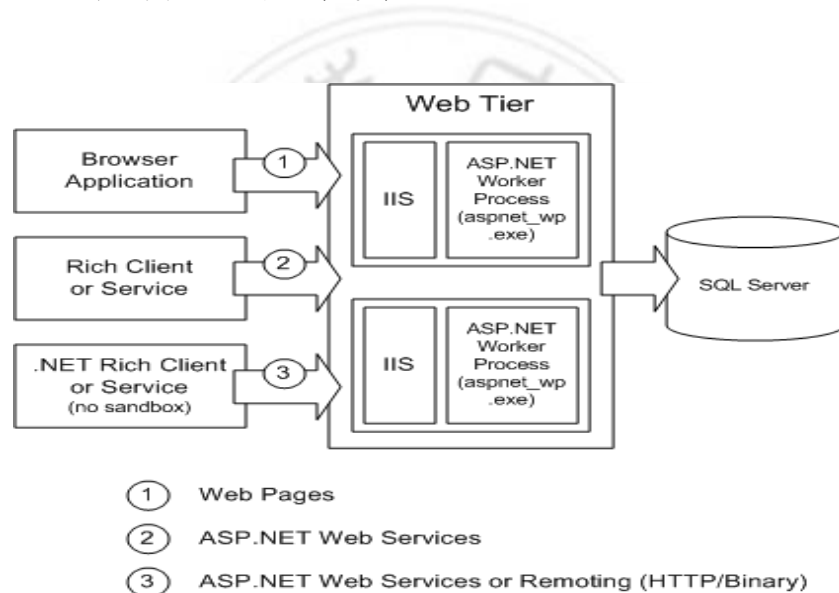


圖 3-1 Web 伺服器端系統架構

## 二、系統架構

本研究設計 Zenbo 機器人點餐系統之建置，一開始主要用 DDE 設計對話代理人建立語料庫，並同時設計餐點資料庫，之後建立 Visual Studio Web 伺服器端以讓資料庫進行存取，最後以 Zenbo SDK Android 程式結合以上軟體，設計出本研究系統，最後進行系統測試階段。上述為本研究設計概念並提出以下系統概念之架構圖(如圖 3-2)，其操作流程為，顧客進入餐廳時，透過靠近或者喚醒，使機器人抓取顧客所在位置，接著透過語音需求詢問帶位需求，後帶領顧客先到位子上，到位子上後觀看餐單，並用語音或點擊方式點餐，最後將訂單資訊傳至後檯

處理。依照顧客所發問題，透過 Zenbo 代理人解析語音轉為文字，控制機器人行動，在此同時透過設計引導式對話，將顧客引導正確步驟，已完成本系統點餐之服務。

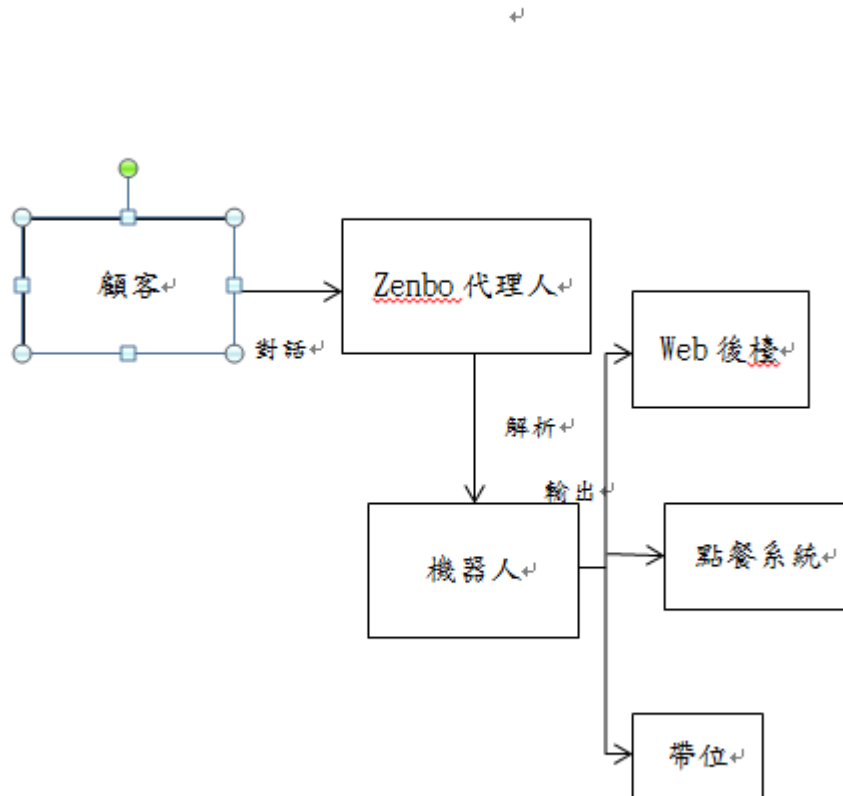


圖 3-2 本研究系統架構

## (一)DDE 語料庫

### (1)主餐菜單設計-主餐關鍵字

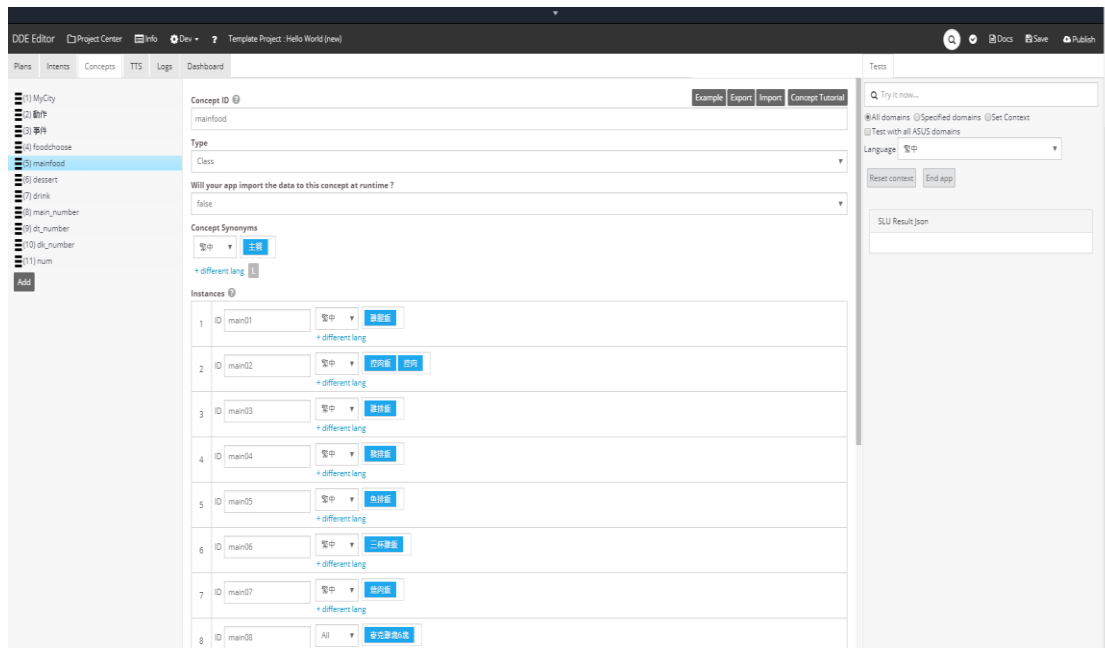


圖 3-3 主餐菜單設計

### (2)點心菜單設計-點心關鍵字

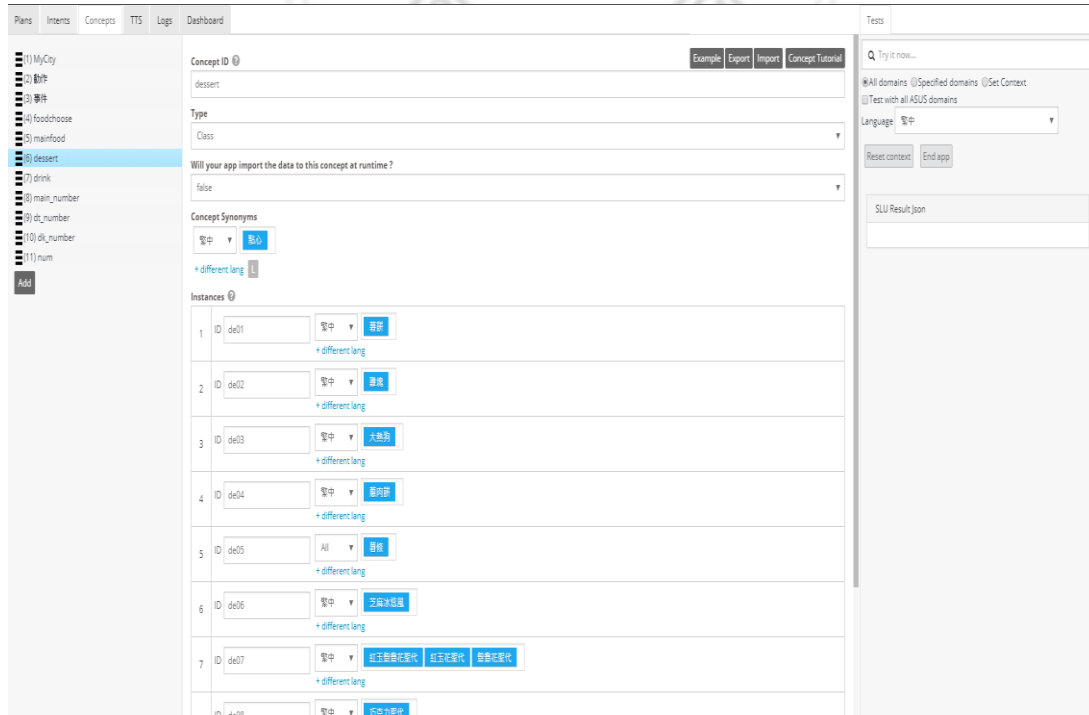


圖 3-4 點心菜單設計

### (3) 飲料菜單設計-飲料關鍵字

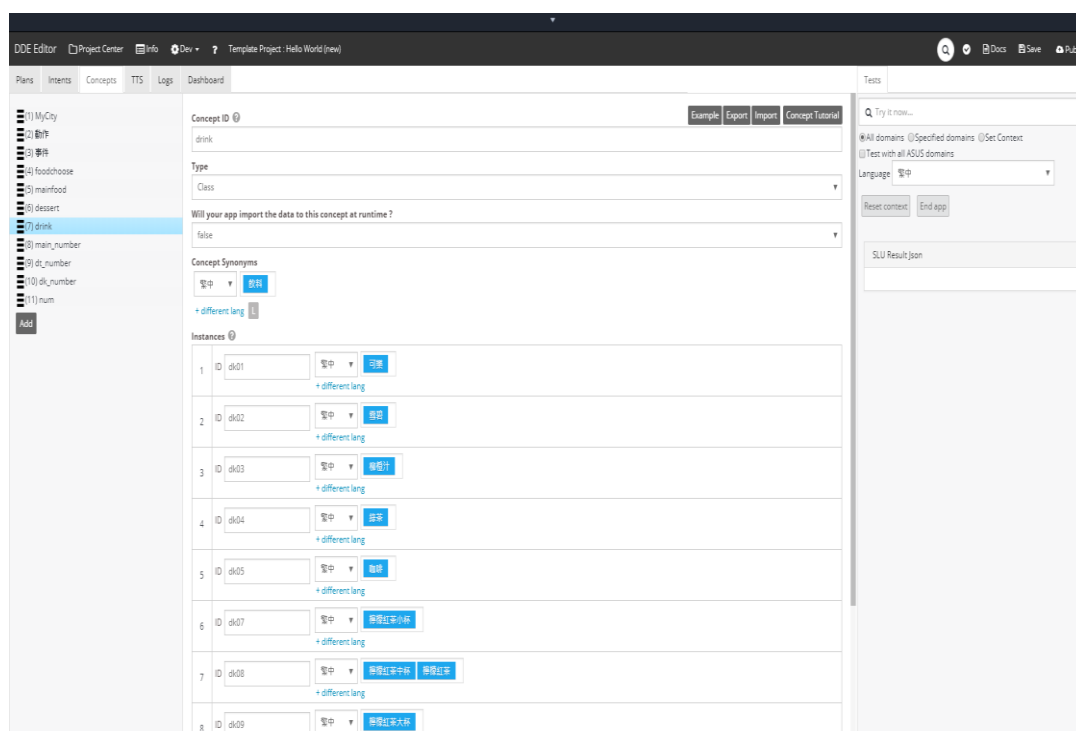


圖 3-5 飲料菜單設計

### (4) 關鍵語句設計-點餐時語句設計

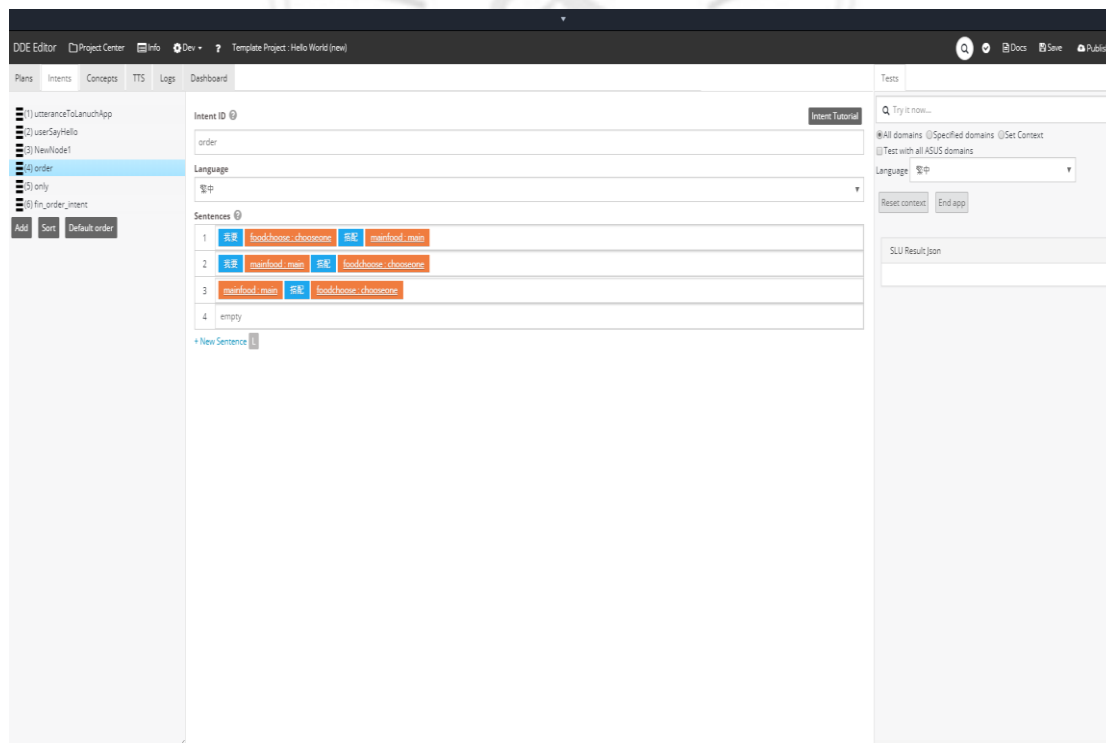


圖 3-6 關鍵語句設計-1



### (5) 關鍵語句設計-點餐語句設計

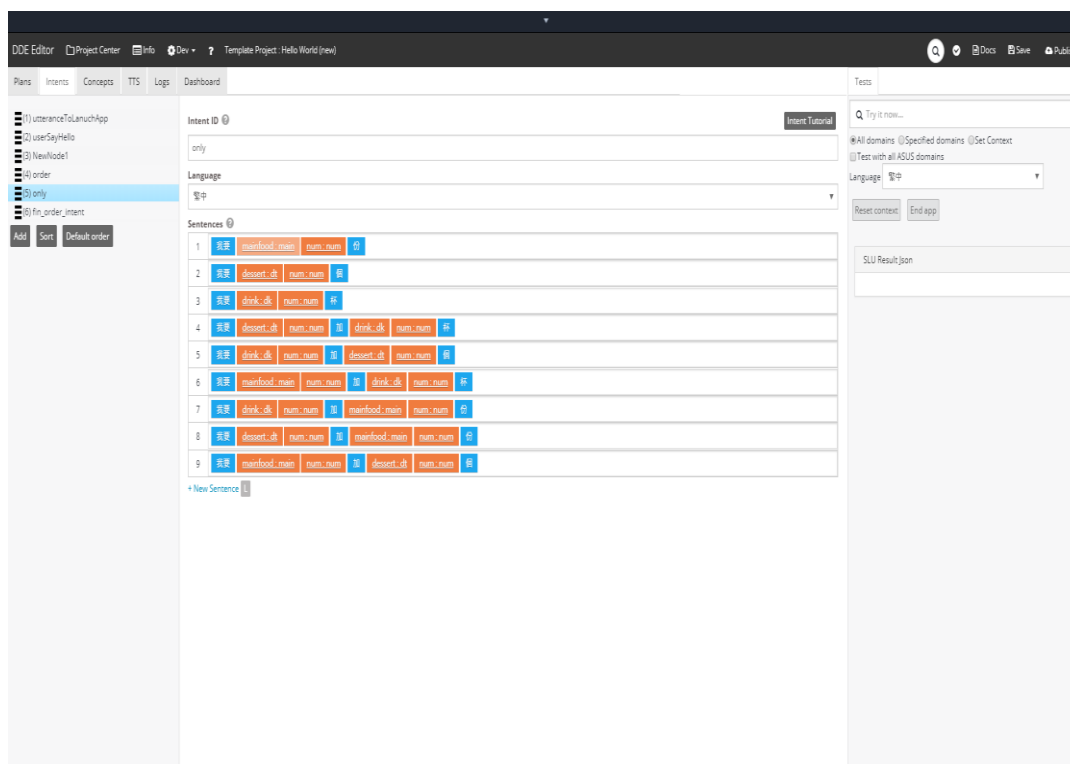


圖 3-7 關鍵語句設計-2

### (6) 關鍵語句設計-開始點餐語句設計

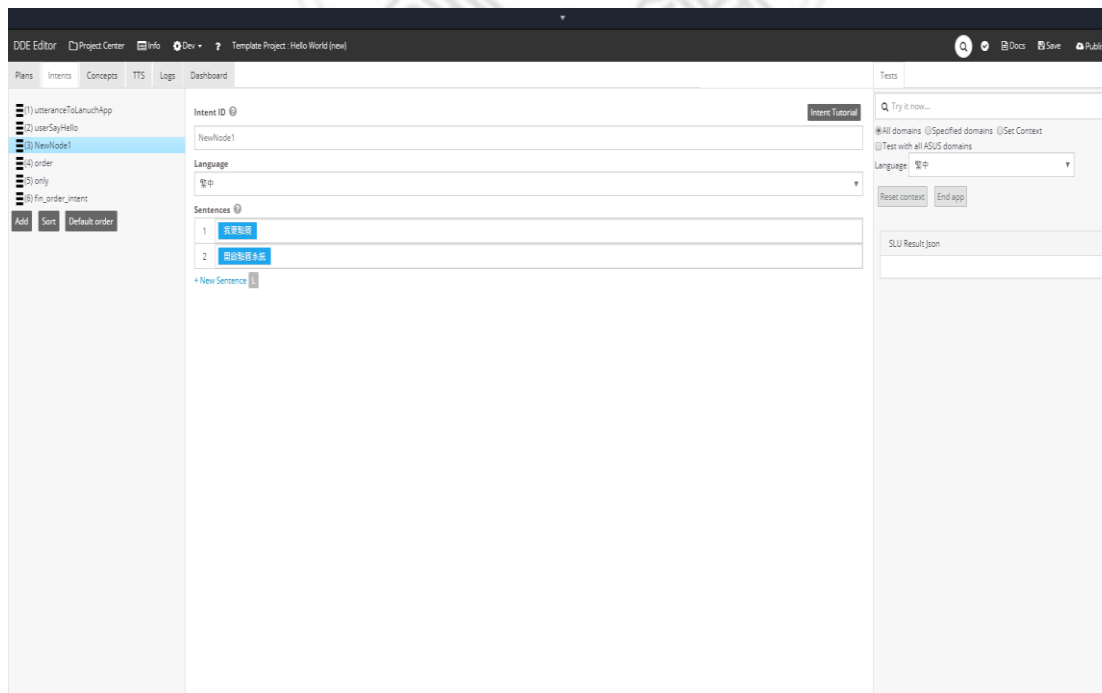


圖 3-8 關鍵語句設計-3

### (7) 關鍵語句設計-完成點餐語句設計

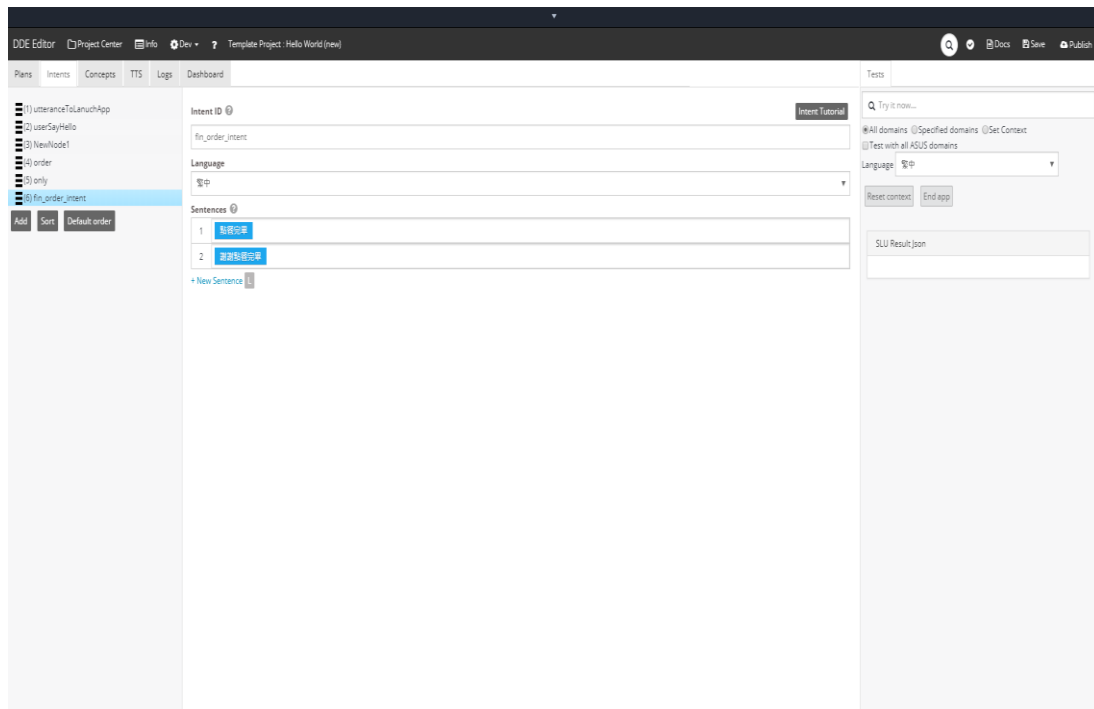


圖 3-9 關鍵語句設計-4

### (8) 菜單與關鍵語句連結-主餐、點心和飲料結合點餐語句設計

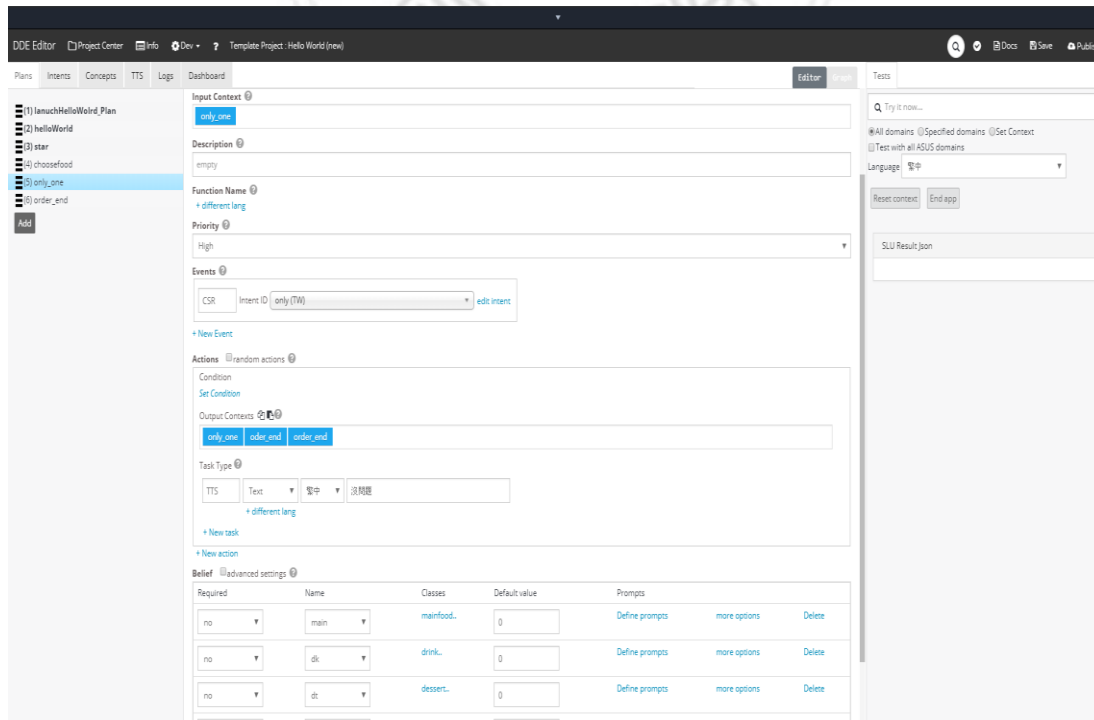


圖 3-10 菜單與關鍵語句連結-1

### (9) 菜單與關鍵語句連結-結合開始點餐語句

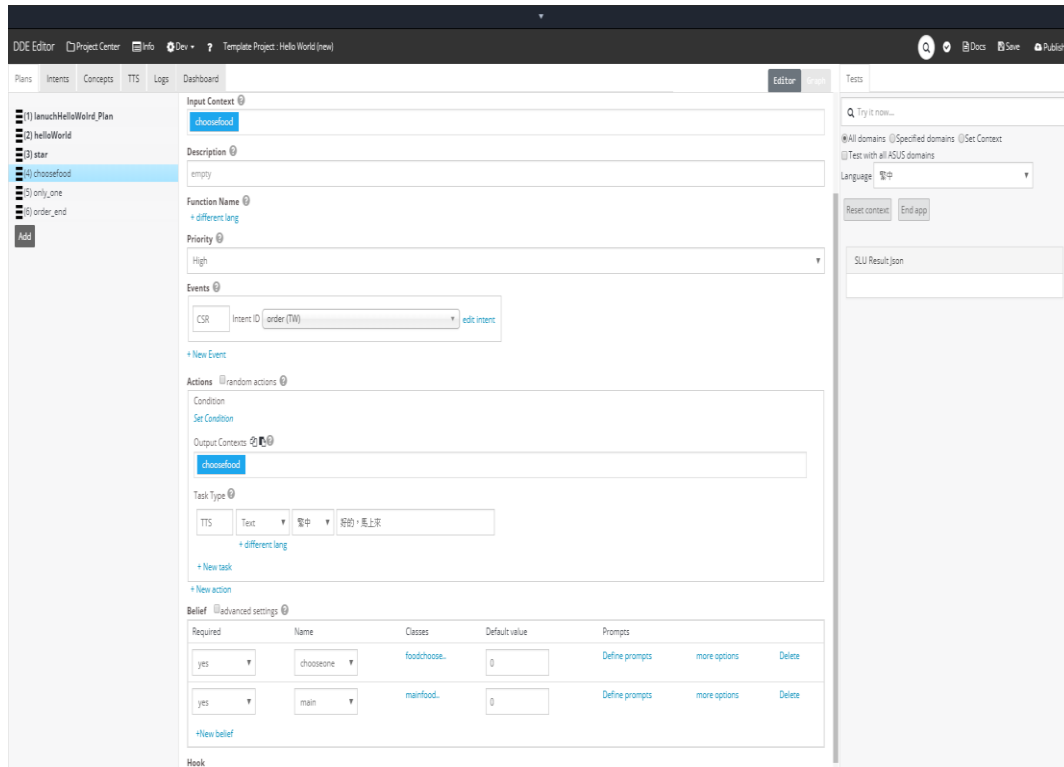


圖 3-11 菜單與關鍵語句連結-2

### (10) 菜單與關鍵語句連結-結合結束點餐語句

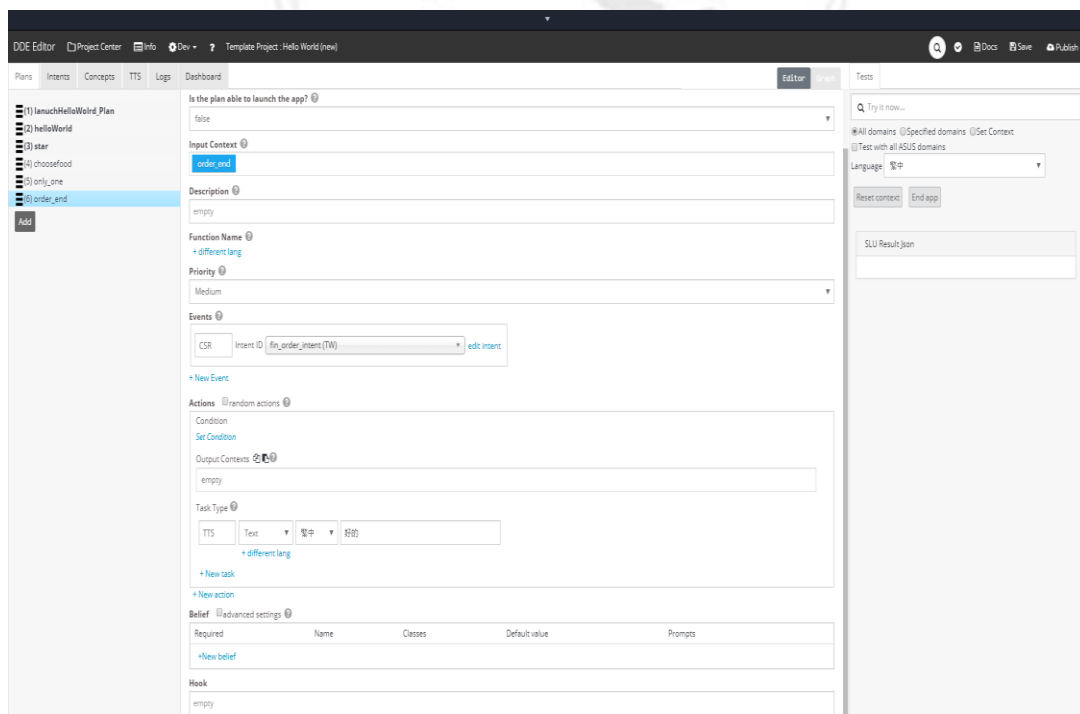


圖 3-12 菜單與關鍵語句連結-3

- 實際操作：  
透過 Zenbo 菜單自行選擇，後以關鍵語句「我要點某餐點」或是以餐點名稱直接進行點餐，套餐須具備主餐、飲料以及點心，若沒有選取，Zenbo 就會提醒使用者，如下圖 3-13 所示。

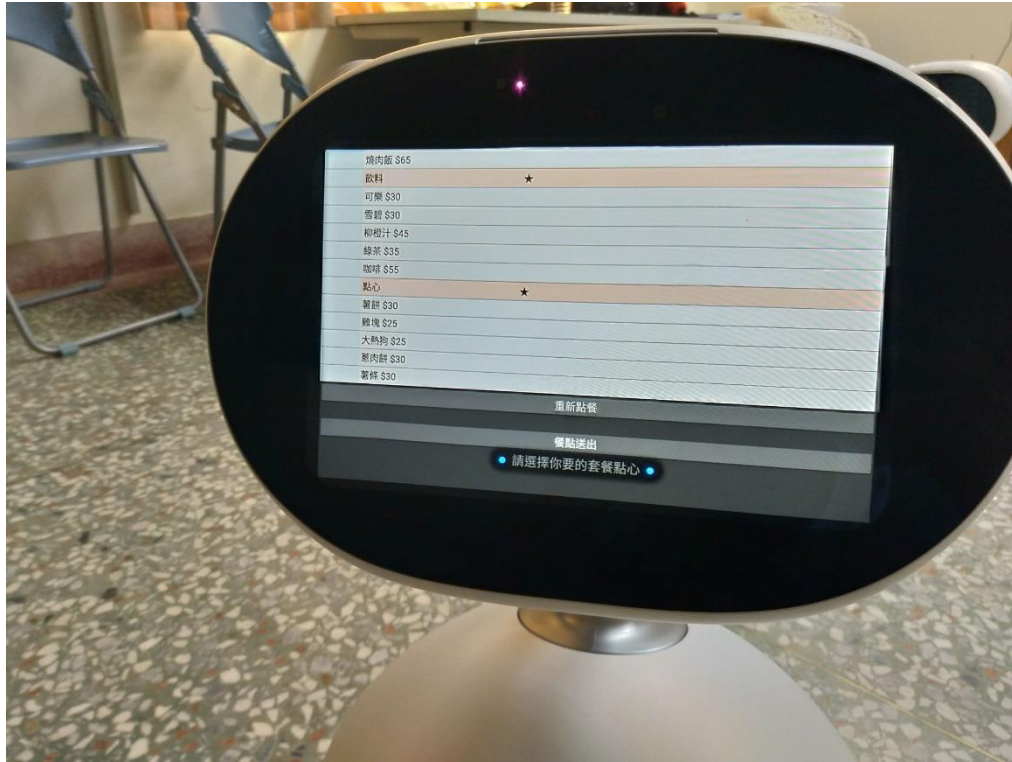


圖 3-13 選取餐點

## (二) Zenbo SDK 設計

### (1) 點餐介面程式碼-餐點清單介面設計 1

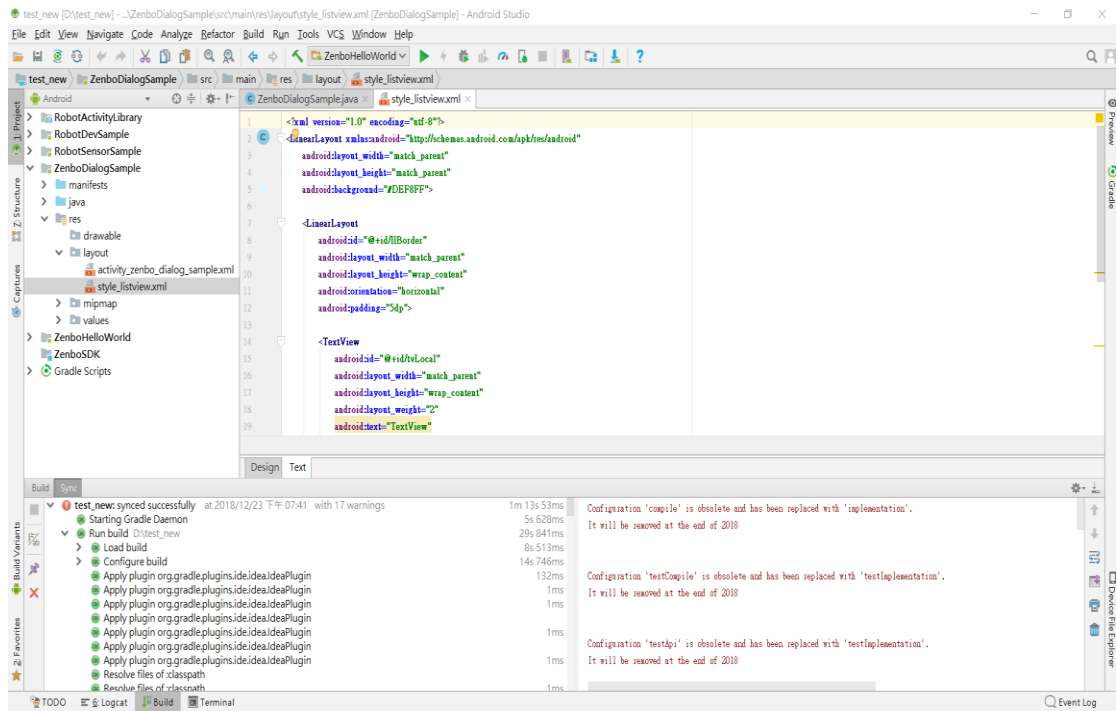


圖 3-14 Zenbo 點餐介面程式碼-1

## (2) 點餐介面程式碼-餐點清單介面設計 2

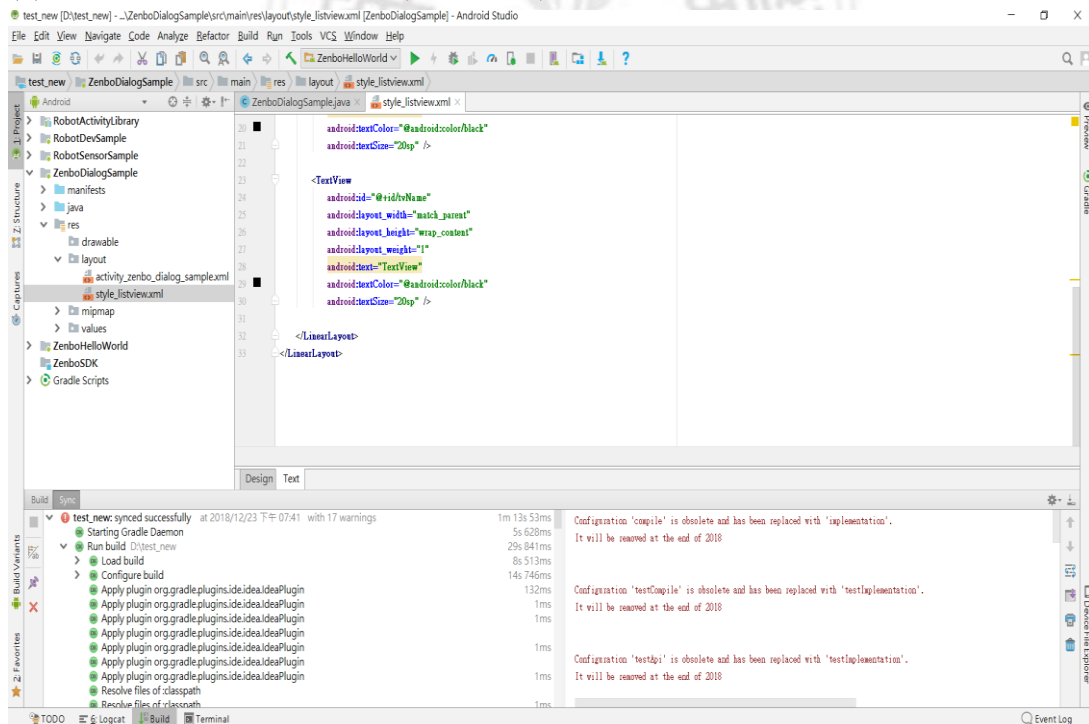


圖 3-15 Zenbo 點餐介面程式碼-2

### (3)點餐介面程式碼-進入整體畫面設計 1

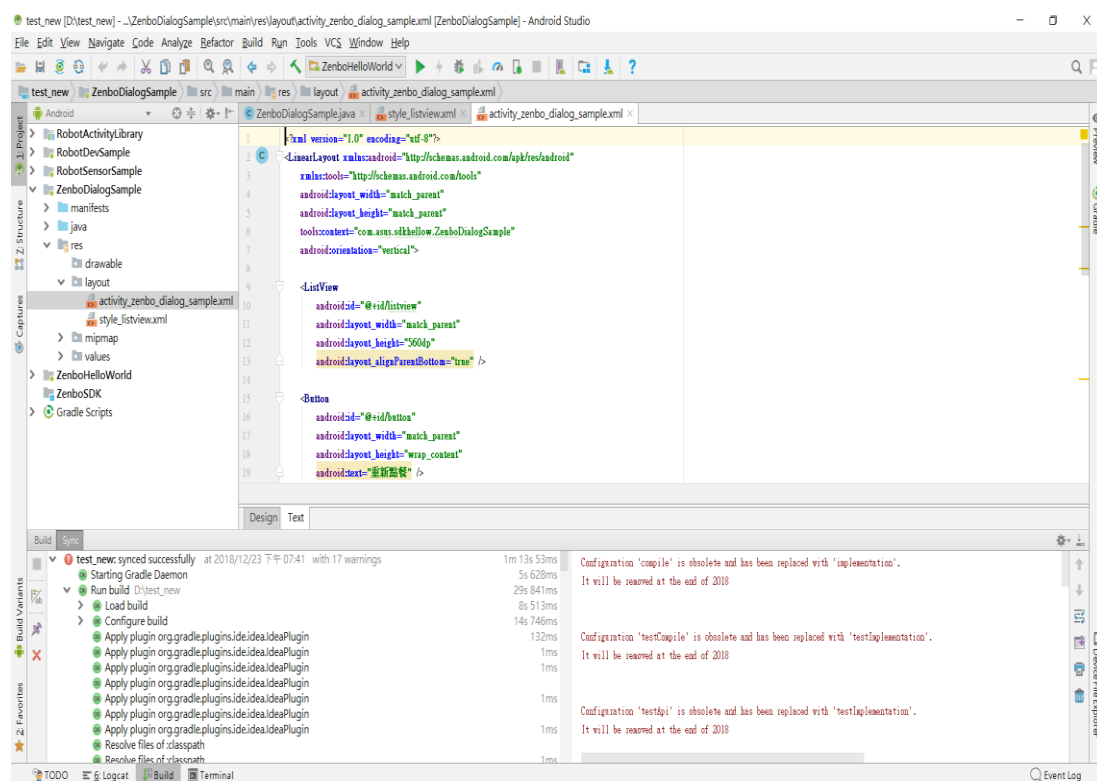


圖 3-16 Zenbo 點餐介面程式碼-3

### (4)點餐介面程式碼-進入整體畫面設計 2

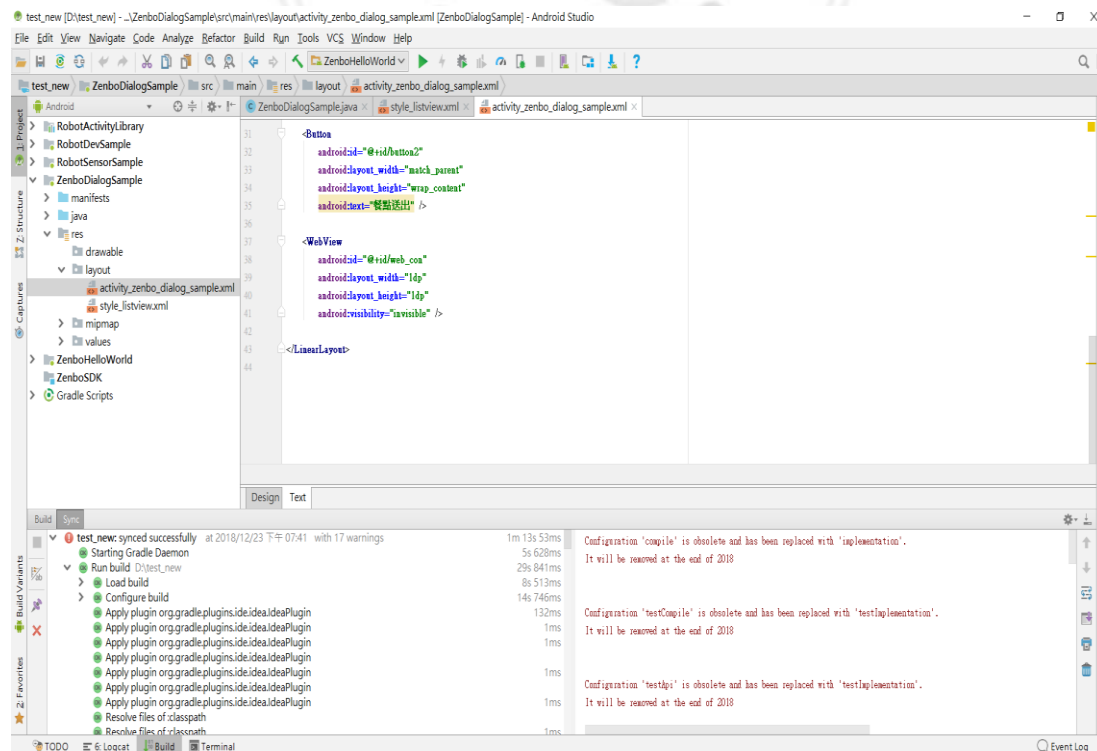


圖 3-17 Zenbo 點餐介面程式碼-4

### (5) Zenbo 功能設計程式碼-帶位功能

```

73 };
74 @Override
75 protected void onCreate(Bundle savedInstanceState) {
76     super.onCreate(savedInstanceState);
77     setContentView(R.layout.activity_zenbo_dialog_sample);
78
79     textView = (TextView) findViewById(R.id.txtText);
80     btnosa_close = (Button) findViewById(R.id.btnosa);
81     webView = (WebView) findViewById(R.id.webvosa);
82
83     robotAPI.robot.setExpressions(RobotFace.ACTIVE, "歡迎來到前鋒飯店");
84     robotAPI.robot.setExpressions(RobotFace.ERROR, "讓我先幫你帶位哦");
85
86     mCountDownTimer = new CountdownTimer( (millisInFuture: 6000, countDownInterval: 1000) {
87         @Override
88         public void onTick(long millisUntilFinished) {
89             robotAPI.action.setText("告罄");
90         }
91
92         @Override
93         public void onFinish() {
94             robotAPI.robot.setExpressions(RobotFace.HIDEFACE, "");
95             robotAPI.robot.speak("帶定位後，可以點選我進行點餐哦", new Config().timeout(10));
96             robotAPI.robot.setExpressions(RobotFace.HIDEFACE, "請選擇你要的餐盤主餐");
97         }
98     });
99
100     mCountDownTimer.start();
101
102     btnosa_close.setOnClickListener(new View.OnClickListener() {
103         @Override

```

圖 3-18 Zenbo 功能設計程式碼-1

### (6) Zenbo 功能設計程式碼-對話語句結合 DDE

```

104 btnosa_close.setOnClickListener(new View.OnClickListener() {
105     @Override
106     public void onClick(View v) {
107         main_order = null;
108         dt_order = null;
109         dt_order = null;
110         all_order = "";
111         textView.setText("");
112         listView.scrollToPosition(0);
113     }
114 });
115 btnosa = (Button) findViewById(R.id.btnosa2);
116 btnosa.setOnClickListener(new View.OnClickListener() {
117     @Override
118     public void onClick(View v) {
119         robotAPI.robot.setExpressions(RobotFace.ACTIVE, "餐盤馬上到齊，請稍作片刻");
120         main_order = ((main_order.replaceAll("ld", replacement: "")),replaceAll( (regularExpression: "lS", replacement: ""))).trim();
121         dt_order = ((dt_order.replaceAll( (regularExpression: "ld", replacement: "")),replaceAll( (regularExpression: "lS", replacement: ""))).trim();
122         dt_order = ((dt_order.replaceAll( (regularExpression: "ld", replacement: "")),replaceAll( (regularExpression: "lS", replacement: ""))).trim();
123         String url = "http://192.168.9.200/zenbofec.aspx?"+ "mainfood="+ main_order + "&drink="+ dt_order + "&dessert="+ dt_order;
124         webView.loadUrl(url);
125         textView.setText("");
126         robotAPI.action.setText("點餐");
127     }
128 });
129 listView = (ListView) findViewById(R.id.listView);
130 LayoutInflater inflater = (LayoutInflater) getSystemService(Context.LAYOUT_INFLATER_SERVICE);
131 ViewAdapter adapter = new ViewAdapter(data, inflater);
132 listView.setAdapter(adapter);
133 listView.setOnItemClickListener((onClickListView); //指定事件 Method
134
135

```

圖 3-19 Zenbo 功能設計程式碼-2

### (7) Zenbo 功能設計程式碼-選取菜單後跳到下一個未點組合

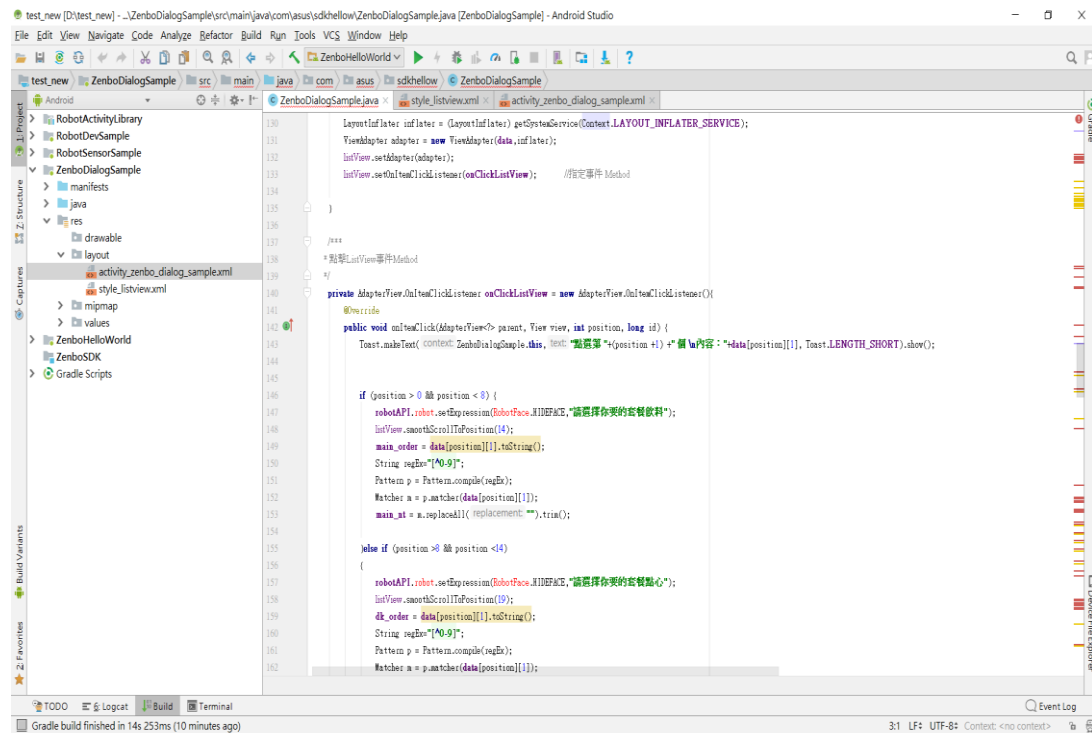


圖 3-20 Zenbo 功能設計程式碼-3

### (8) Zenbo 功能設計程式碼-提醒使用者那些未點組合

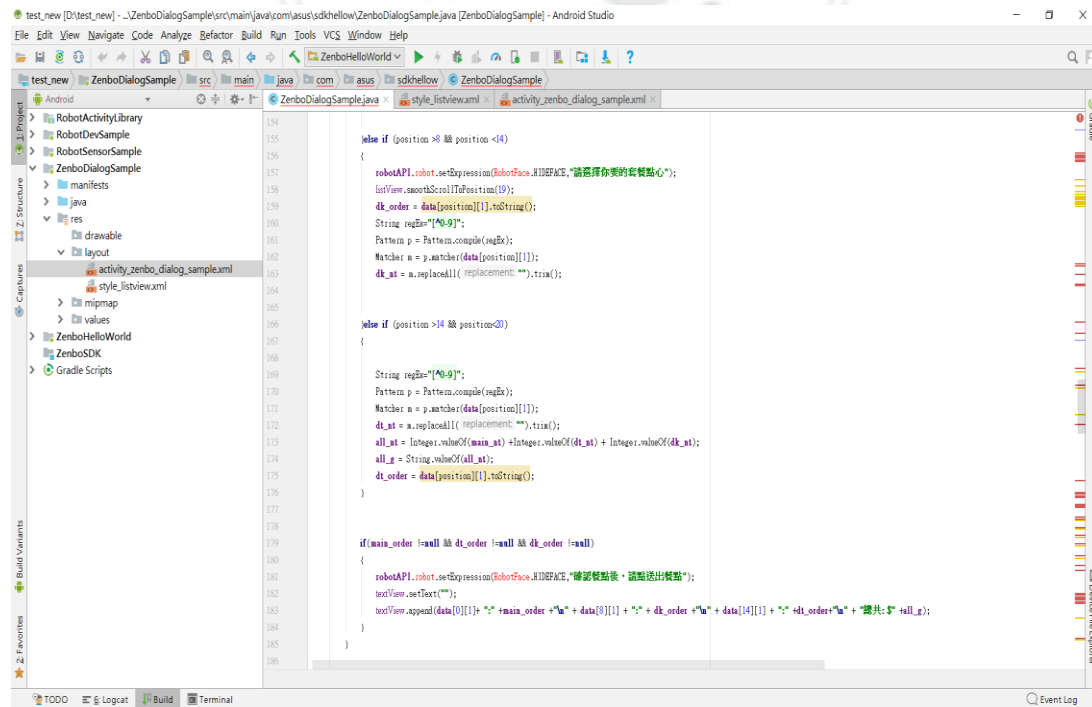
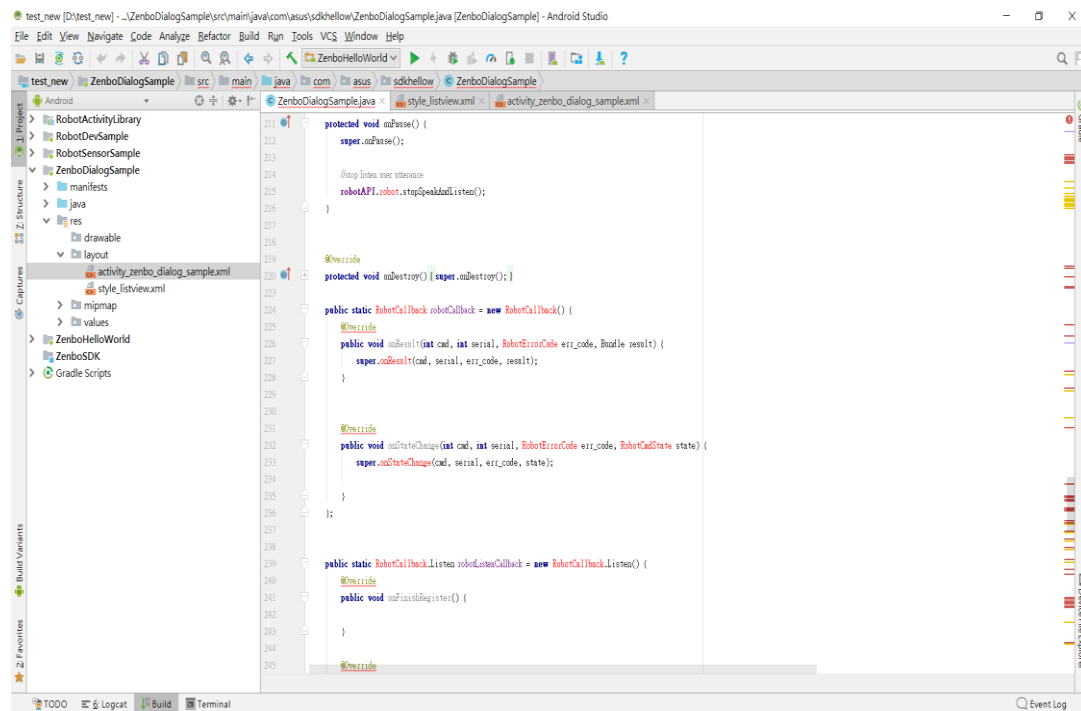


圖 3-21 Zenbo 功能設計程式碼-4



## (9) Zenbo 功能設計程式碼-程式跳出時狀態



```
protected void onPause() {
    super.onPause();

    //stop listen user utterance
    robotAPI.robot.stopSpeakAndListen();
}

@Override
protected void onDestroy() {
    super.onDestroy();
}

public static RobotCallBack robotCallBack = new RobotCallBack() {
    @Override
    public void onResult(int cmd, int serial, RobotErrorCode err_code, Bundle result) {
        super.onResult(cmd, serial, err_code, result);
    }
}

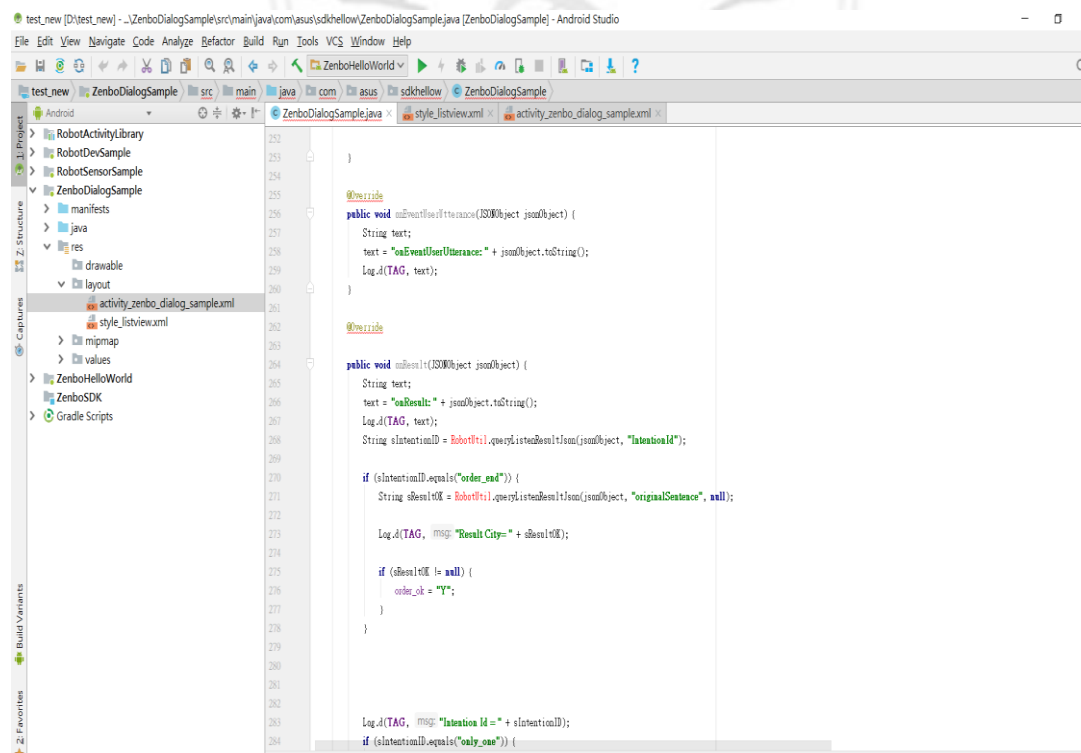
@Override
public void onStateChange(int cmd, int serial, RobotErrorCode err_code, RobotCmdState state) {
    super.onStateChange(cmd, serial, err_code, state);
}

};

public static RobotCallBack.Listen robotListenCallBack = new RobotCallBack.Listen() {
    @Override
    public void onUiThreadRegister() {
    }
}
}
```

圖 3-22 Zenbo 功能設計程式碼-5

## (10) Zenbo 功能設計程式碼-接收 DDE 的資料



```
}

@Override
public void onEventReceived(IInterface (ISpeechObject jspeechObject) {
    String text;
    text = "onEventUserUtterance:" + jspeechObject.toString();
    Log.d(TAG, text);
}

@Override
public void onResult(ISpeechObject jspeechObject) {
    String text;
    text = "onResult:" + jspeechObject.toString();
    Log.d(TAG, text);
    String sIntentionID = RobotAPI.speechIntentionResult(jspeechObject, "IntentionID");

    if (sIntentionID.equals("order_end")) {
        String sResult = RobotAPI.speechIntentionResult(jspeechObject, "originalSentence", null);
        Log.d(TAG, MSG: "Result City=" + sResult);

        if (sResult != null) {
            order_ok = "Y";
        }
    }

    Log.d(TAG, MSG: "Intention ID = " + sIntentionID);
    if (sIntentionID.equals("only_ones")) {
    }
}
```

圖 3-23 Zenbo 功能設計程式碼-6

## (11) Zenbo 功能設計程式碼-DDE 資料做比對

```

282
283
284 Log.d(TAG, msg: "intention id = " + intentionID);
285
286 if (intentionID.equals("only_one")) {
287     @ResInt main = RobotUtil.queryIntentionResultFrom(jsonObject, "main", null);
288     @ResInt hit = RobotUtil.queryIntentionResultFrom(jsonObject, "hit", null);
289     @ResInt = RobotUtil.queryIntentionResultFrom(jsonObject, "dt", null);
290     @ResInt main1 = RobotUtil.queryIntentionResultFrom(jsonObject, "area01", null);
291     @ResInt main2 = RobotUtil.queryIntentionResultFrom(jsonObject, "area02", null);
292     @ResInt main3 = RobotUtil.queryIntentionResultFrom(jsonObject, "area03", null);
293     String sResult = RobotUtil.queryIntentionResultFrom(jsonObject, "originalSentence", null);
294     Log.d(TAG, msg: "Result City:" + @ResInt main + @ResInt main1 + @ResInt hit + @ResInt main2 + @ResInt main3);
295
296     if (@ResInt main != null || @ResInt hit != null || @ResInt main1 != null) {
297         @ResInt hit += @ResInt;
298         //TextView.setText("你幫的是" + @ResInt main + @ResInt main1 + @ResInt hit + @ResInt main2 + @ResInt main3 + @ResInt hit);
299
300         String a[] = @ResInt hit.split( regularExpression: "我表(\\d)");
301         for (int i = 0; i < a.length; i++) {
302
303             int[i] = a[i];
304
305         }
306
307         adapter.notifyDataSetChanged();
308     }
309
310 }
311
312
313 @Override
314 public void onActivityResult(int requestCode, int resultCode, Intent data) {

```

圖 3-24 Zenbo 功能設計程式碼-7

## (12) Zenbo 功能設計程式碼-菜單畫面框大小設計

```

1 package com.asus.sdkhellow;
2
3 import ...
4
5
6
7
8
9
10
11
12 public class ViewAdapter extends BaseAdapter {
13
14
15     private String[] ElementsData; //資料
16     private LayoutInflater inflater; //版面layout
17     private int indentationBase; //字元縮排
18
19     //優化ListView 避免重新加載
20     //避免宣告你會動到他的元件
21     static class ViewHolder {
22         LinearLayout llBorder;
23         TextView Name;
24         TextView Local;
25     }
26
27     //初始化
28     public ViewAdapter(String[] data, LayoutInflater inflater) {
29         this.ElementsData = data;
30         this.inflater = inflater;
31         indentationBase = 100;
32     }
33
34     //取得數量
35     @Override
36     public int getCount() { return ElementsData.length; }
37
38     //取得Item
39     @Override
40     public Object getItem(int position) { return ElementsData[position]; }

```

圖 3-25 Zenbo 功能設計程式碼-8

### (13) Zenbo 功能設計程式碼-菜單清單變數設定

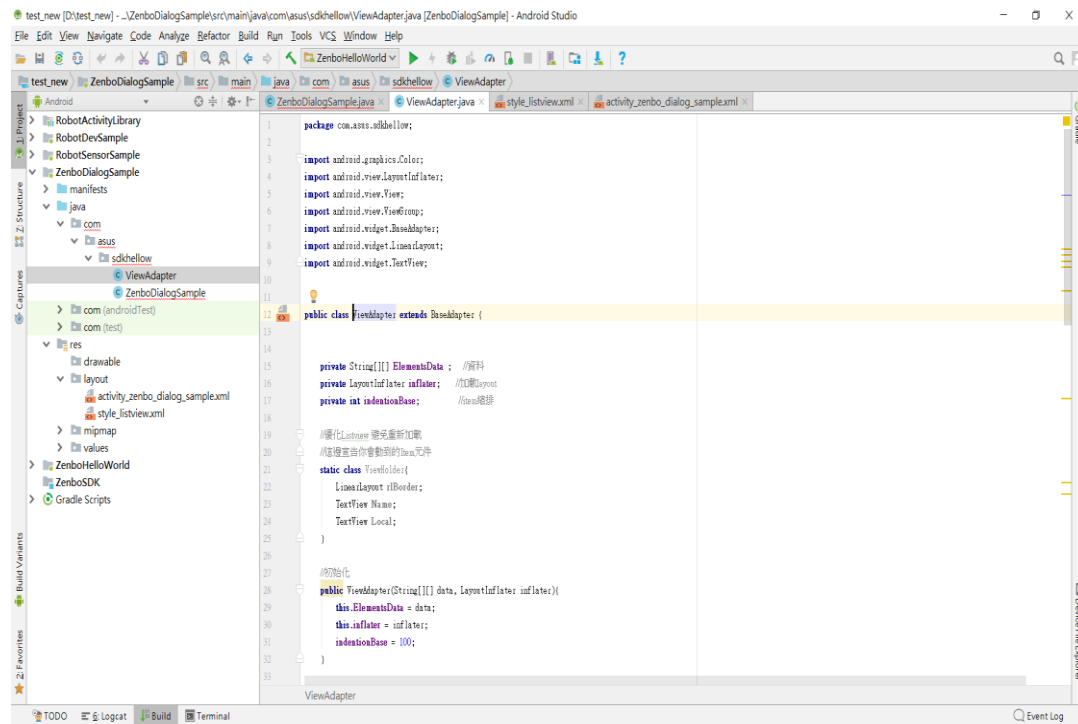


圖 3-26 Zenbo 功能設計程式碼-9

### (14) Zenbo 功能設計程式碼-菜單對應設計餐點文字套用

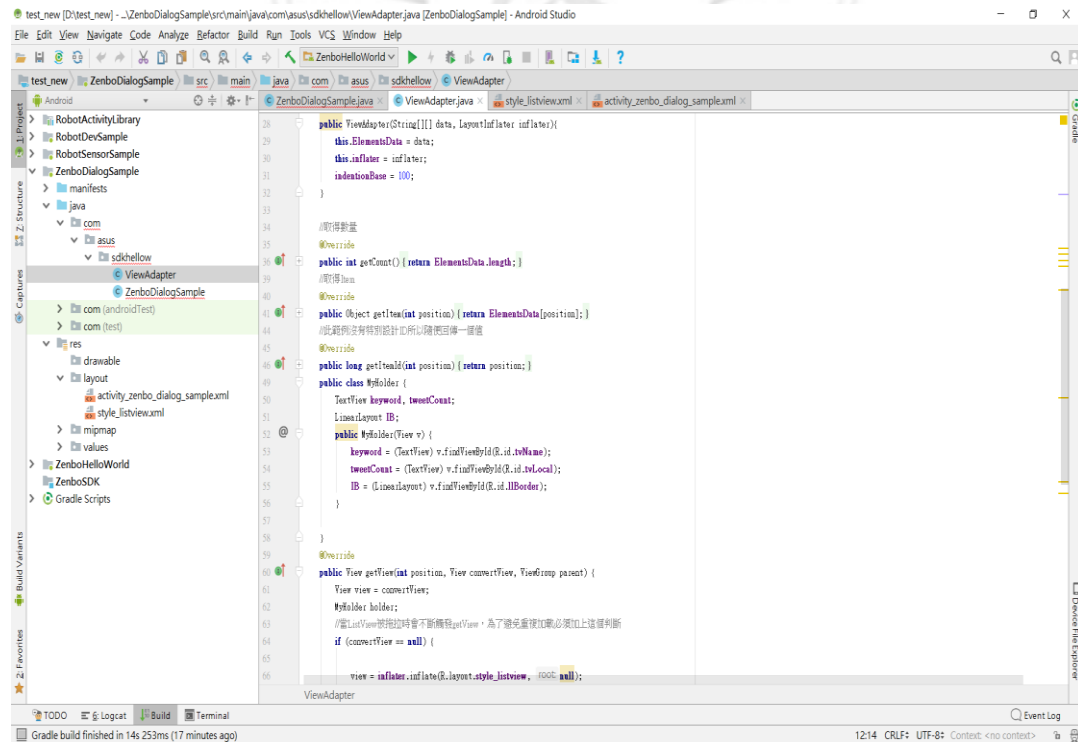
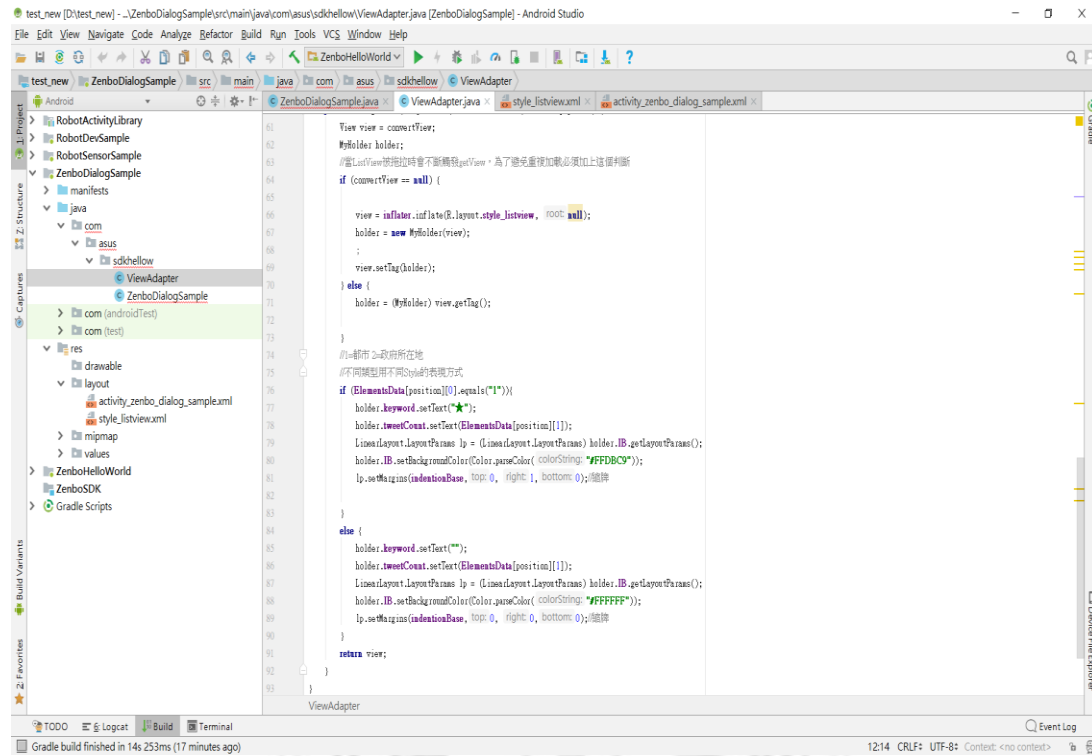


圖 3-27 Zenbo 功能設計程式碼-10

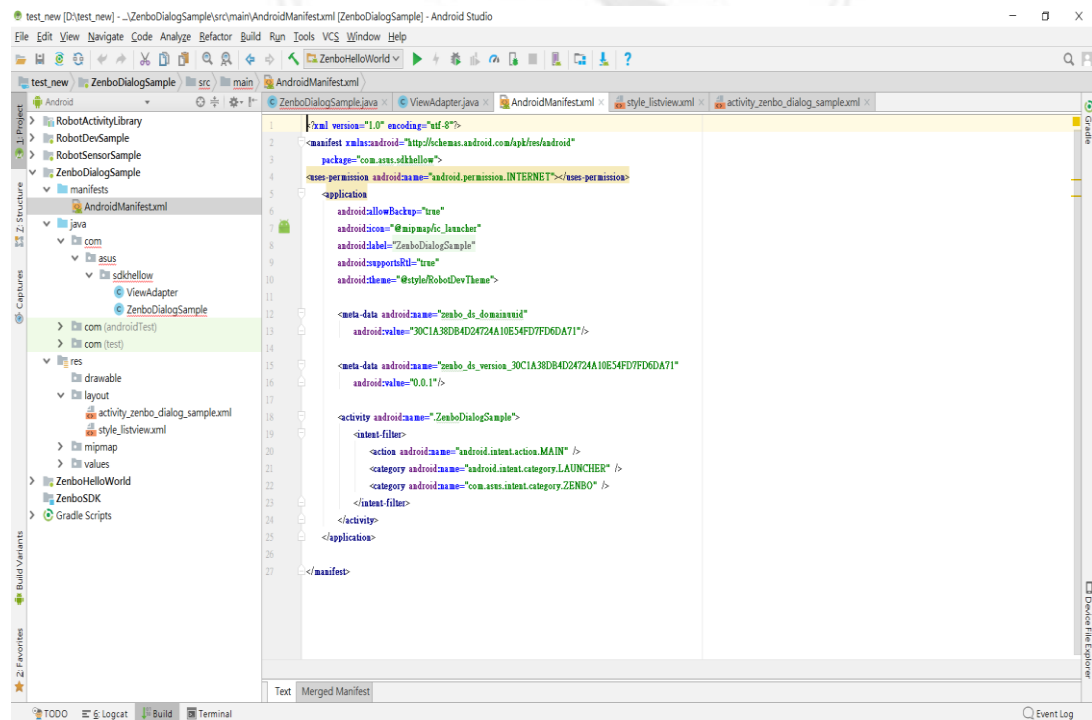
## (15) Zenbo 功能設計程式碼-菜單主要標題設計與餐點顯示設計



```
61 View view = convertView;
62 ViewHolder holder;
63 //當ListView被拖拉時會不斷重設convertView，為了避免重複加載必須加上這個判斷
64 if (convertView == null) {
65
66     view = inflater.inflate(R.layout.style_listview, root, null);
67     holder = new ViewHolder(view);
68     ;
69     view.setTag(holder);
70 } else {
71     holder = (ViewHolder) view.getTag();
72 }
73
74 //城市 = 政府所在地
75 //不同類型用不同convertView的表現方式
76 if (ElementData[position][0].equals("1")){
77     holder.keyword.setText("★");
78     holder.timesCount.setText(ElementData[position][1]);
79     LinearLayout.LayoutParams lp = (LinearLayout.LayoutParams) holder.lB.getLayoutParams();
80     holder.lB.setBackgroundColor(Color.parseColor("FFD8C9"));
81     lp.setMargins(indentationBase, top, 0, right, 1, bottom, 0); //縮排
82 }
83
84 else {
85     holder.keyword.setText("");
86     holder.timesCount.setText(ElementData[position][1]);
87     LinearLayout.LayoutParams lp = (LinearLayout.LayoutParams) holder.lB.getLayoutParams();
88     holder.lB.setBackgroundColor(Color.parseColor("FFFFFF"));
89     lp.setMargins(indentationBase, top, 0, right, 0, bottom, 0); //縮排
90 }
91
92 return view;
93 }
```

圖 3-28 Zenbo 功能設計程式碼-11

## (16) Zenbo 功能設計程式碼-Zenbo 介面設計



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.asus.sdkhellow">
4     <uses-permission android:name="android.permission.INTERNET"></uses-permission>
5
6     <application
7         android:allowBackup="true"
8         android:icon="@mipmap/ic_launcher"
9         android:label="@string/app_name"
10        android:roundIcon="@mipmap/ic_launcher_round"
11        android:supportRtl="true"
12        android:theme="@style/RobotDevTheme">
13
14        <meta-data android:name="zenbo_dt_40mainuid"
15            android:value="30CIA38DB4D24724A10E54FD7FD6DA71"/>
16
17        <meta-data android:name="zenbo_dt_version_30CIA38DB4D24724A10E54FD7FD6DA71"
18            android:value="0.0.1"/>
19
20        <activity android:name=".ZenboDialogSample">
21            <intent-filter>
22                <action android:name="android.intent.action.MAIN" />
23                <category android:name="android.intent.category.LAUNCHER" />
24                <category android:name="com.asus.intent.category.ZENBO" />
25            </intent-filter>
26        </activity>
27    </application>
28 </manifest>
```

圖 3-29 Zenbo 功能設計程式碼-12

- 實際操作：

一開始進入簡餐店，讓 Zenbo 機器人先將客人進行帶位動作(如圖 3-30)，坐定位後，開始看到菜單畫面(如圖 3-31)，可以進行餐點選取，點選後會自動顯示訊息你點擊餐點以進行確認，並跳到其他未點組合(如圖 3-32)，點選主餐、飲料與點心完成後，確認點選餐點(如圖 3-33)，若有問題再點擊重新點餐，若沒有問題就可以點擊餐點送出，完成全部點餐步驟(如圖 3-34)。



圖 3-30 進行帶位動作





圖 3-31 餐點菜單

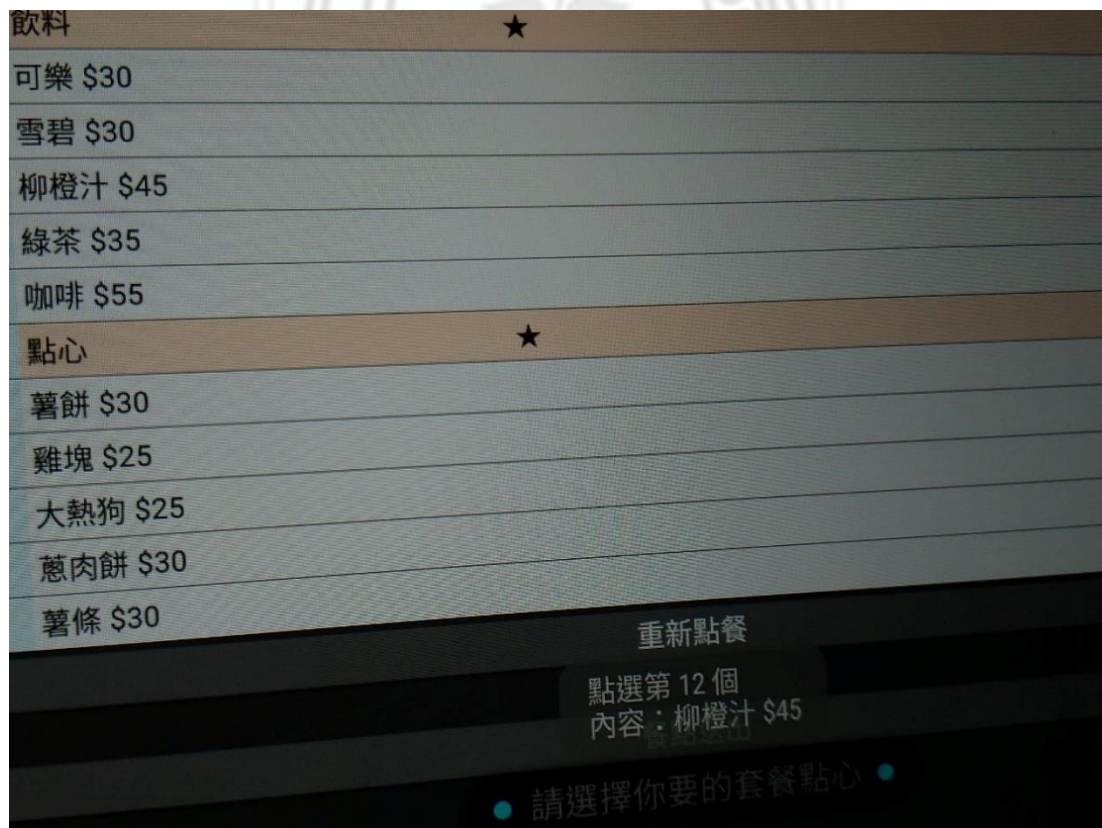


圖 3-32 目前點選餐點

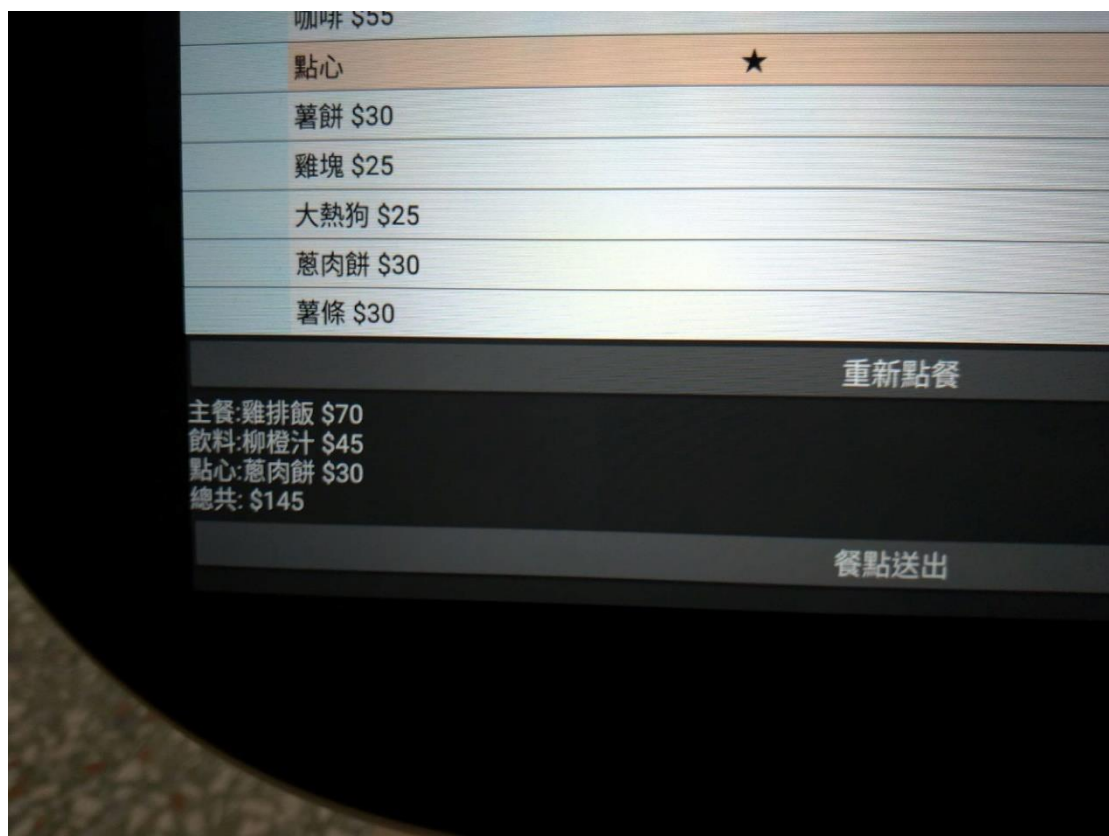


圖 3-33 確認最後餐點



圖 3-34 點餐完成



### (三)ASP 伺服器端

#### (1) ASP.NET 程式設計-後臺接收清單畫面設計 1

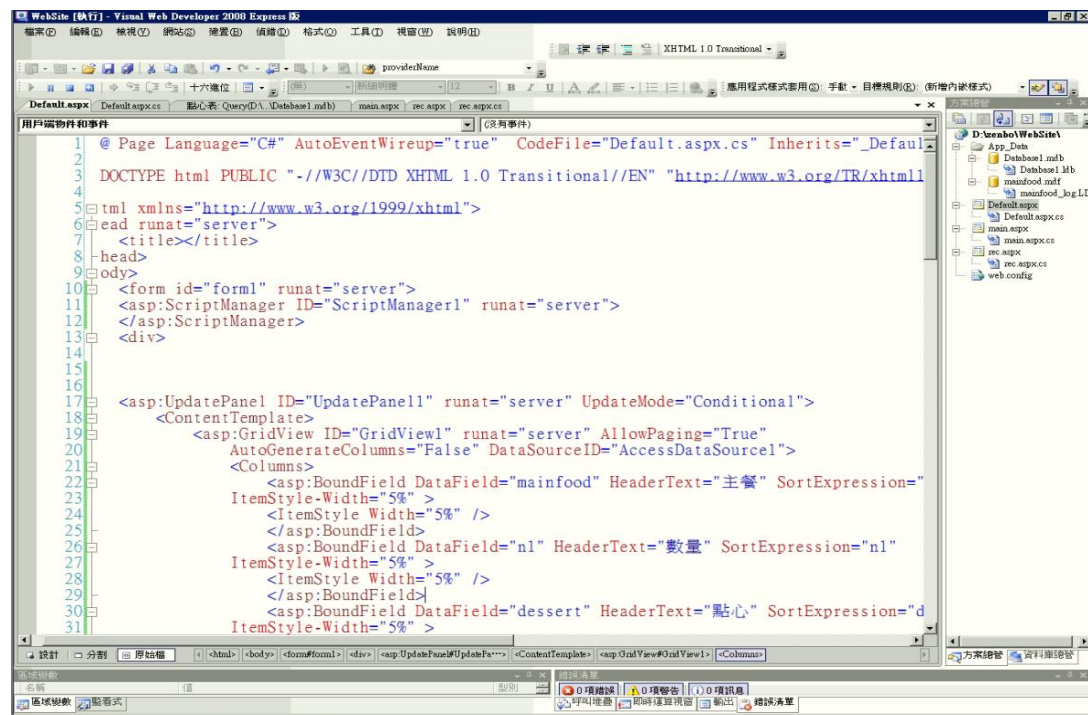


圖 3-35 ASP.NET 程式設計-1

#### (2) ASP.NET 程式設計-後臺接收清單畫面設計 2

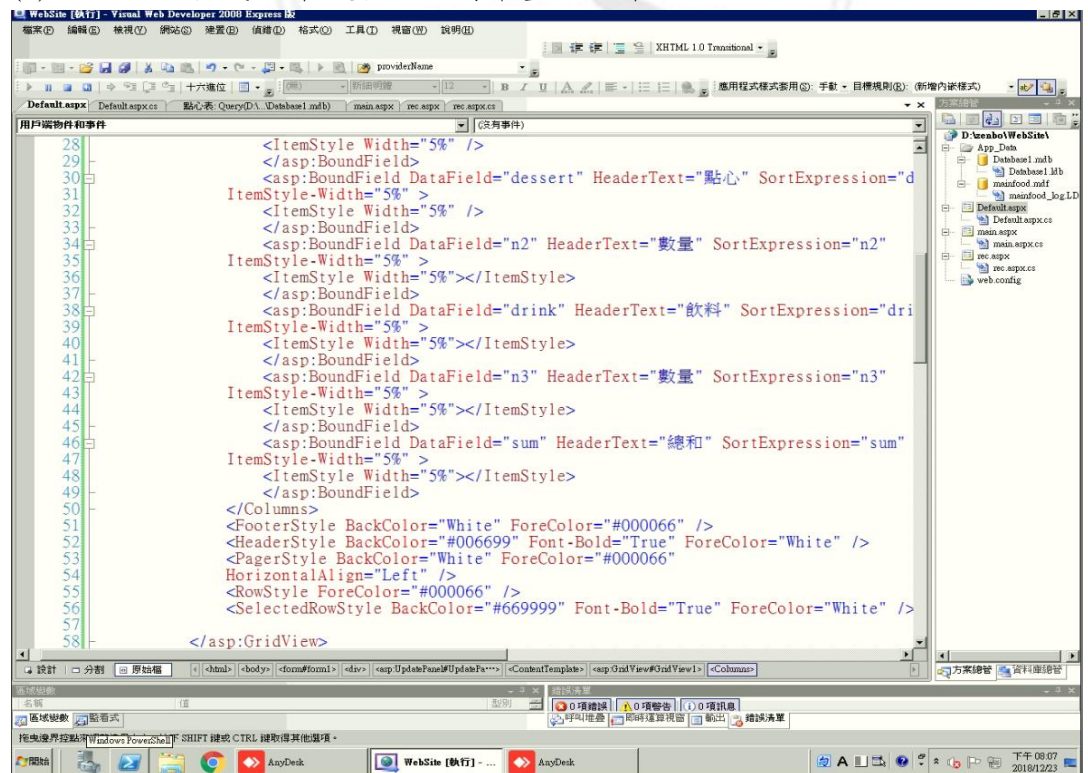


圖 3-36 ASP.NET 程式設計-2



(3) ASP.NET 程式設計-後臺接收清單查詢資料庫語法設定

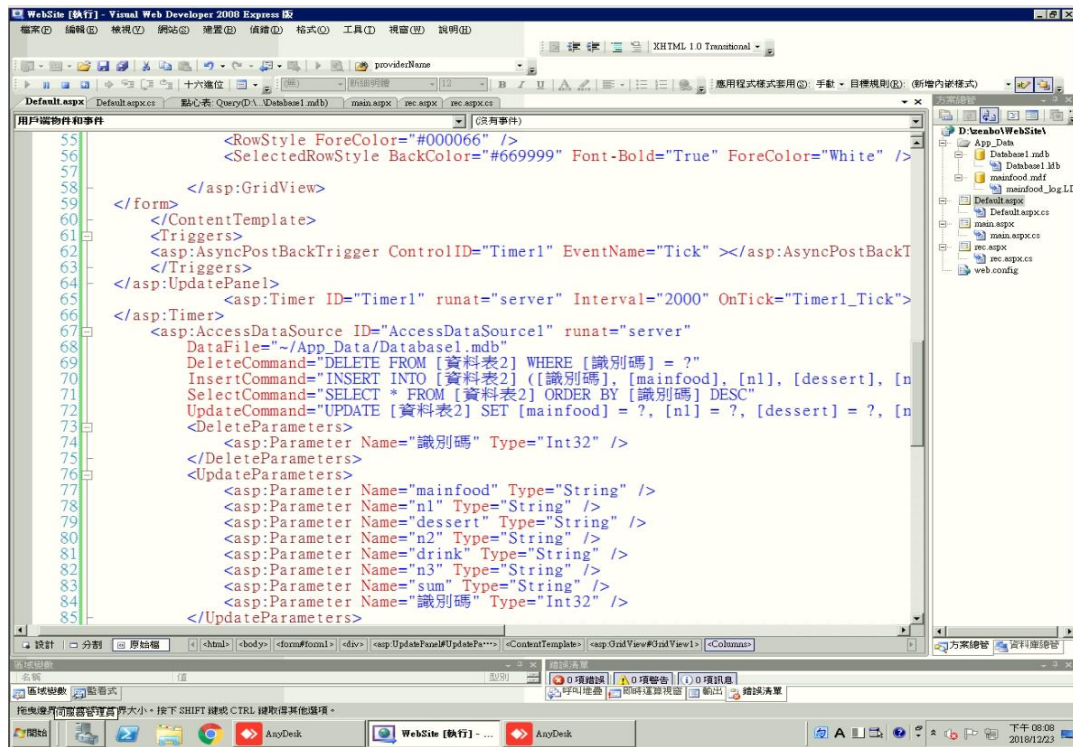


圖 3-37 ASP.NET 程式設計-3

(4) ASP.NET 程式設計-後臺接收清單查詢資料庫語法設定

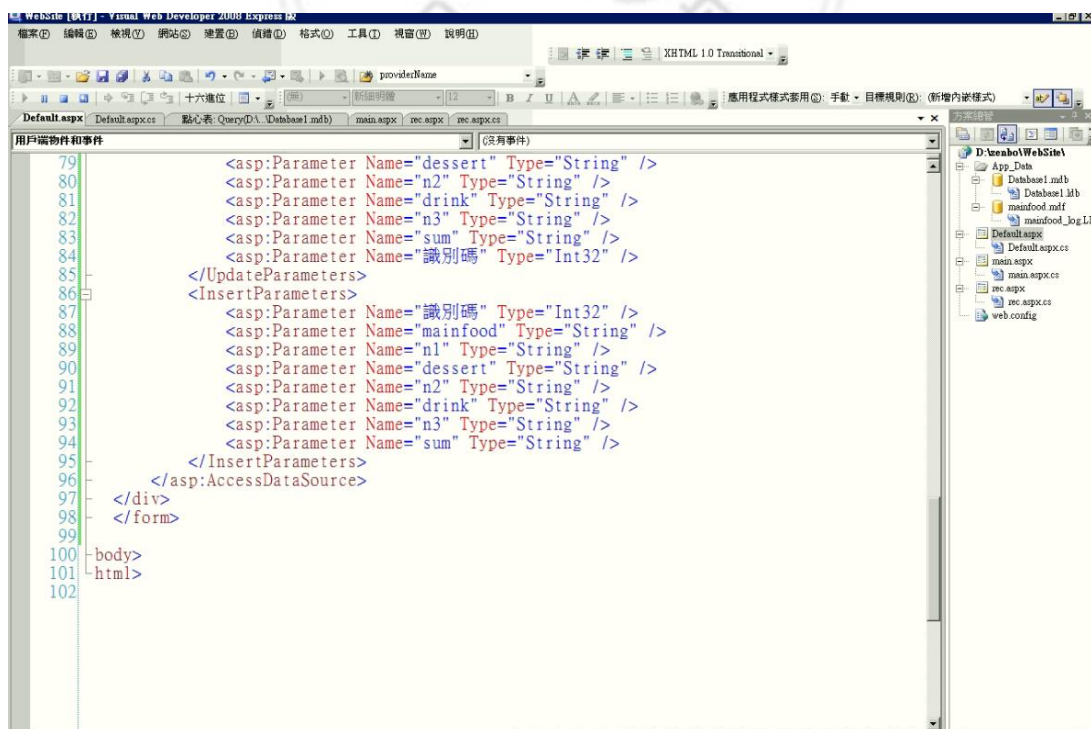


圖 3-38 ASP.NET 程式設計-4

(5) ASP.NET 程式設計-後臺頁面接收到訂餐資訊後重新整理

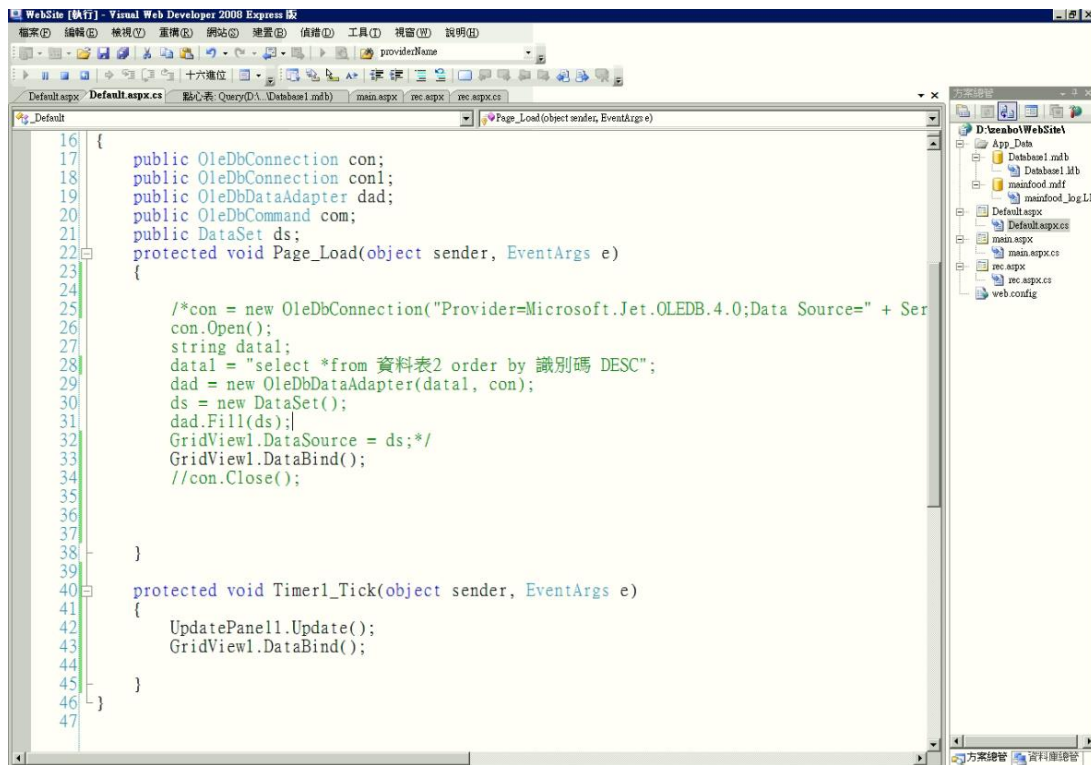


圖 3-39 ASP.NET 程式設計-5

(6) ASP.NET 程式設計-新增資料庫語法設定

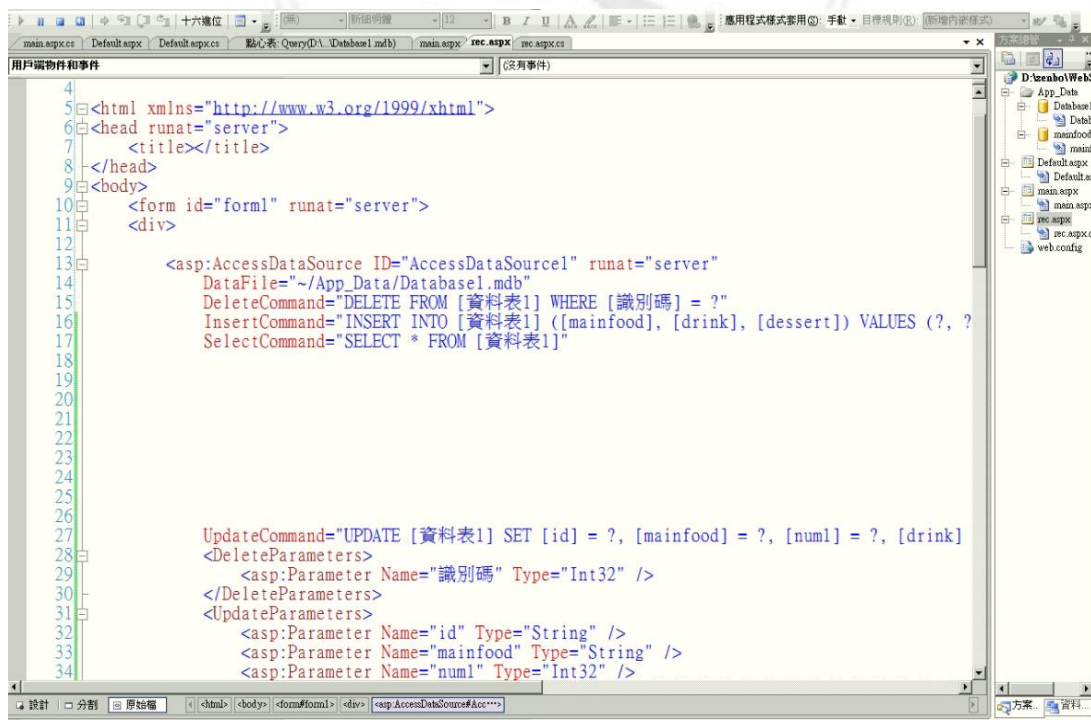
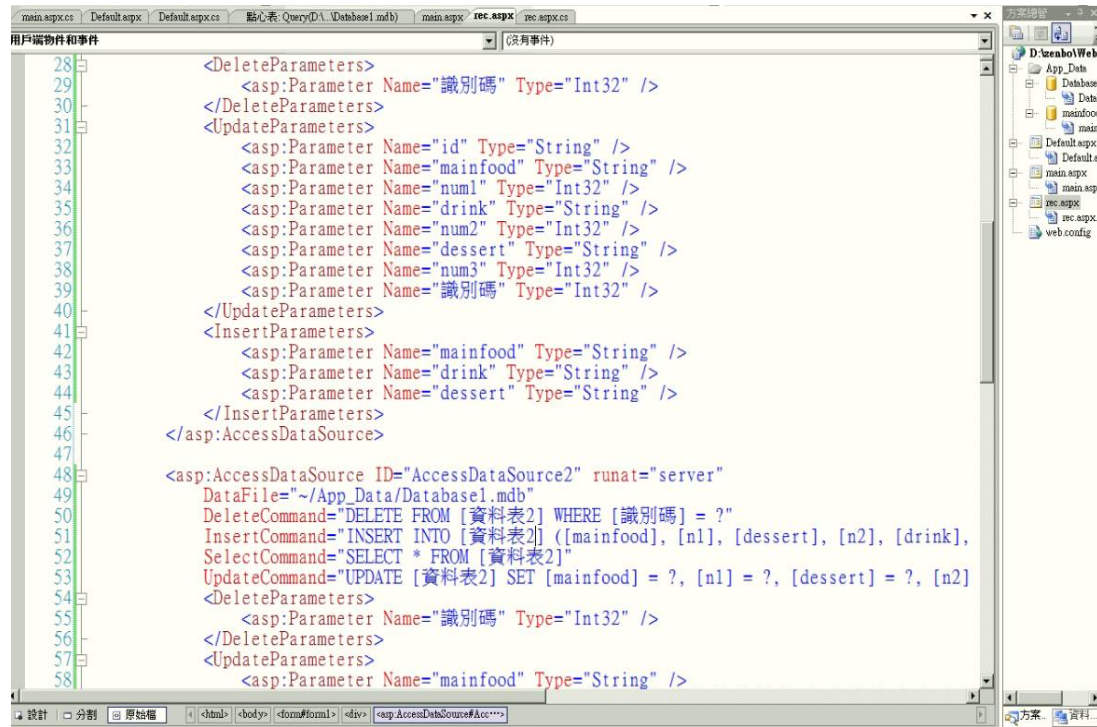


圖 3-40 ASP.NET 程式設計-6

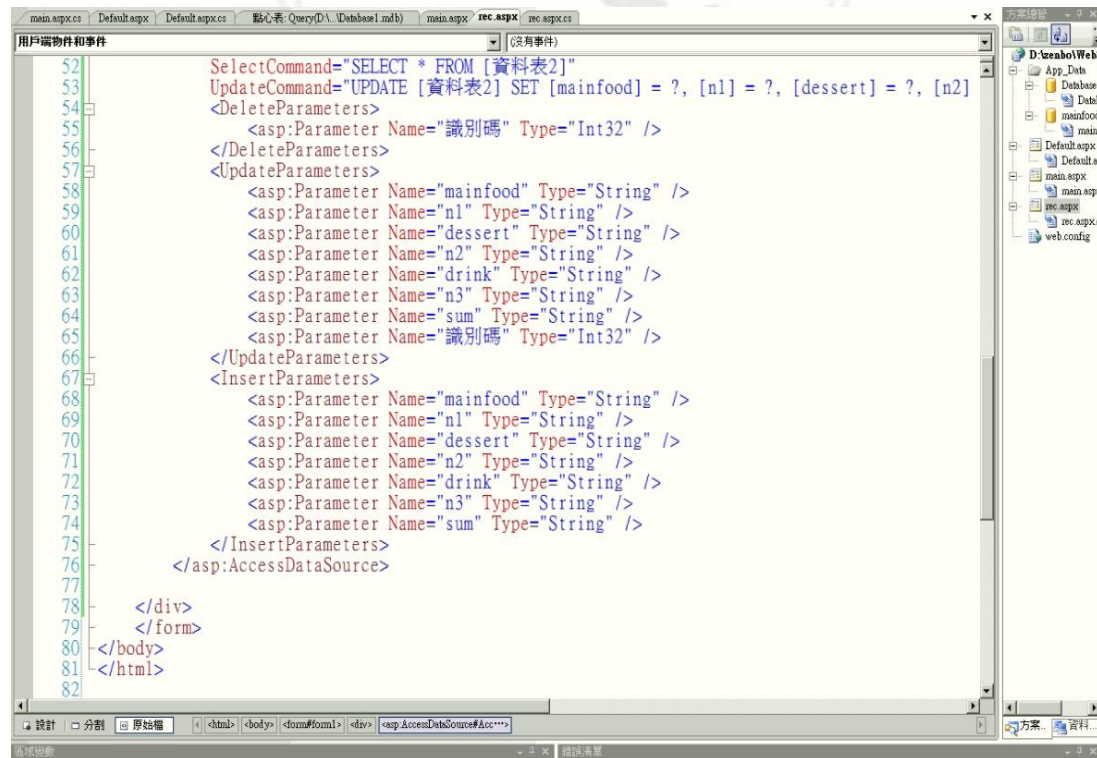
### (7) ASP.NET 程式設計-新增資料庫對應欄位



```
28 <DeleteParameters>
29 <asp:Parameter Name="識別碼" Type="Int32" />
30 </DeleteParameters>
31 <UpdateParameters>
32 <asp:Parameter Name="id" Type="String" />
33 <asp:Parameter Name="mainfood" Type="String" />
34 <asp:Parameter Name="num1" Type="Int32" />
35 <asp:Parameter Name="drink" Type="String" />
36 <asp:Parameter Name="num2" Type="Int32" />
37 <asp:Parameter Name="dessert" Type="String" />
38 <asp:Parameter Name="num3" Type="Int32" />
39 <asp:Parameter Name="識別碼" Type="Int32" />
40 </UpdateParameters>
41 <InsertParameters>
42 <asp:Parameter Name="mainfood" Type="String" />
43 <asp:Parameter Name="drink" Type="String" />
44 <asp:Parameter Name="dessert" Type="String" />
45 </InsertParameters>
46 </asp:AccessDataSource>
47
48 <asp:AccessDataSource ID="AccessDataSource2" runat="server"
49 DataFile="~/App_Data/Databasel.mdb"
50 DeleteCommand="DELETE FROM [資料表2] WHERE [識別碼] = ?"
51 InsertCommand="INSERT INTO [資料表2] ([mainfood], [n1], [dessert], [n2], [drink],
52 SelectCommand="SELECT * FROM [資料表2]"
53 UpdateCommand="UPDATE [資料表2] SET [mainfood] = ?, [n1] = ?, [dessert] = ?, [n2]
54 <DeleteParameters>
55 <asp:Parameter Name="識別碼" Type="Int32" />
56 </DeleteParameters>
57 <UpdateParameters>
58 <asp:Parameter Name="mainfood" Type="String" />
```

圖 3-41 ASP.NET 程式設計-7

### (8) ASP.NET 程式設計-新增資料庫對應欄位型別

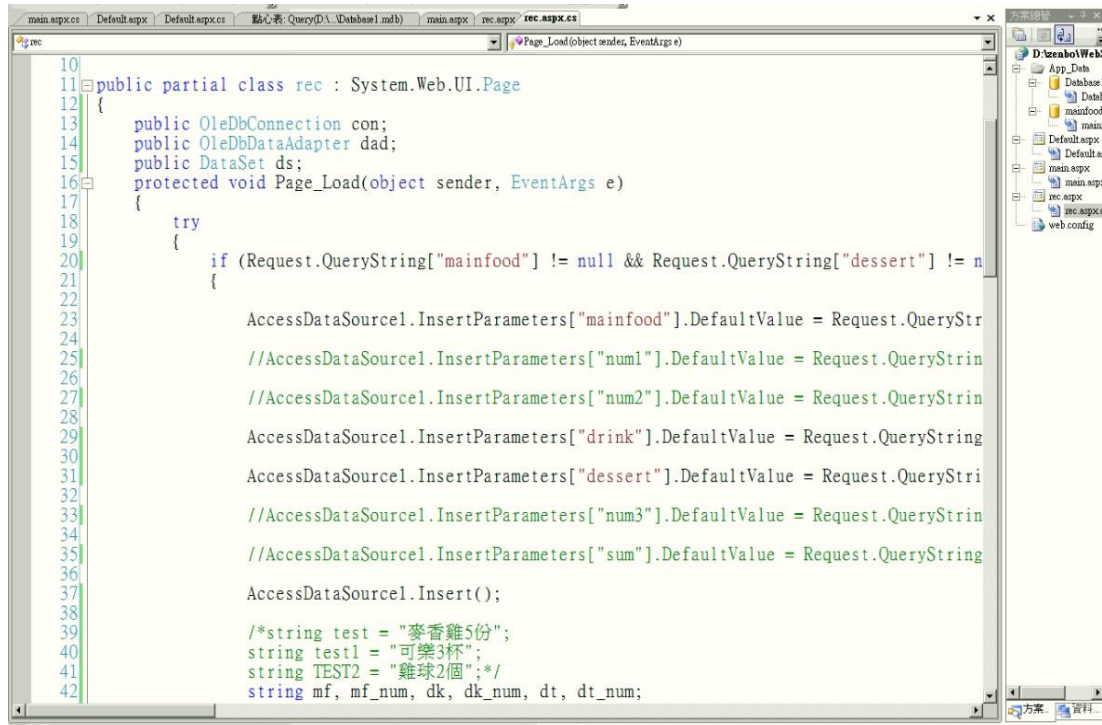


```
52 SelectCommand="SELECT * FROM [資料表2]"
53 UpdateCommand="UPDATE [資料表2] SET [mainfood] = ?, [n1] = ?, [dessert] = ?, [n2]
54 <DeleteParameters>
55 <asp:Parameter Name="識別碼" Type="Int32" />
56 </DeleteParameters>
57 <UpdateParameters>
58 <asp:Parameter Name="mainfood" Type="String" />
59 <asp:Parameter Name="n1" Type="String" />
60 <asp:Parameter Name="dessert" Type="String" />
61 <asp:Parameter Name="n2" Type="String" />
62 <asp:Parameter Name="drink" Type="String" />
63 <asp:Parameter Name="n3" Type="String" />
64 <asp:Parameter Name="sum" Type="String" />
65 <asp:Parameter Name="識別碼" Type="Int32" />
66 </UpdateParameters>
67 <InsertParameters>
68 <asp:Parameter Name="mainfood" Type="String" />
69 <asp:Parameter Name="n1" Type="String" />
70 <asp:Parameter Name="dessert" Type="String" />
71 <asp:Parameter Name="n2" Type="String" />
72 <asp:Parameter Name="drink" Type="String" />
73 <asp:Parameter Name="n3" Type="String" />
74 <asp:Parameter Name="sum" Type="String" />
75 </InsertParameters>
76 </asp:AccessDataSource>
77
78 </div>
79 </form>
80 </body>
81 </html>
82
```

圖 3-42 ASP.NET 程式設計-8



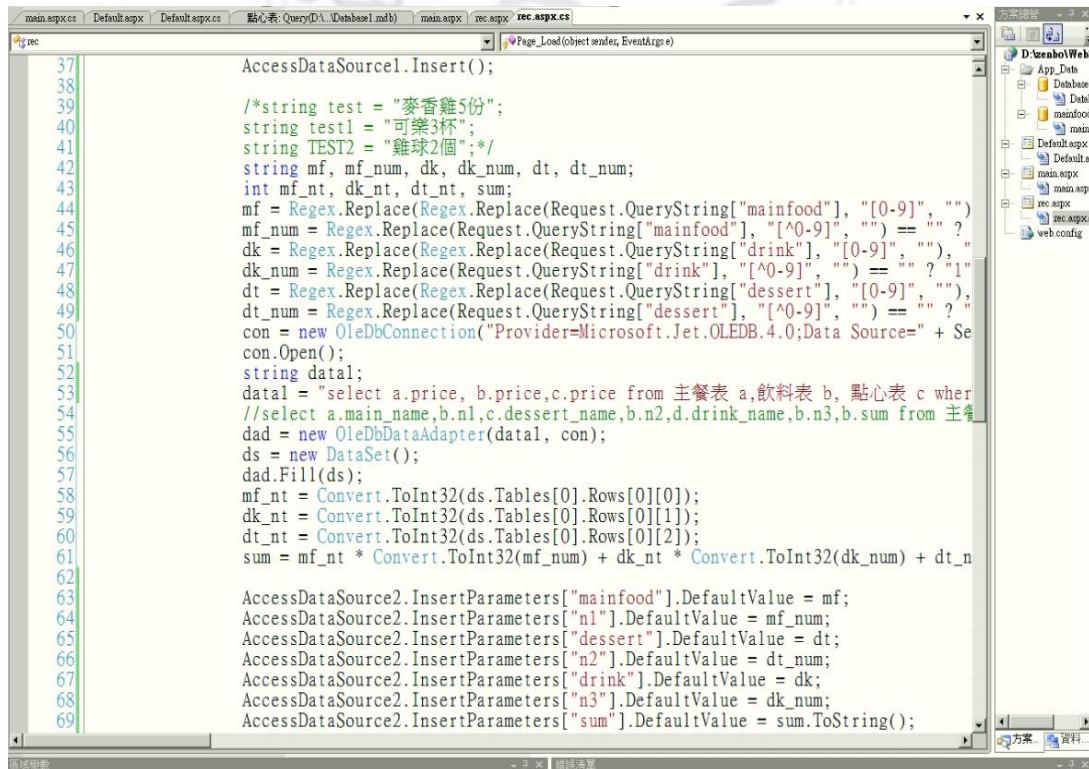
### (9) ASP.NET 程式設計-接收到資料後，傳入資料庫



```
10
11 public partial class rec : System.Web.UI.Page
12 {
13     public OleDbConnection con;
14     public OleDbDataAdapter dad;
15     public DataSet ds;
16     protected void Page_Load(object sender, EventArgs e)
17     {
18         try
19         {
20             if (Request.QueryString["mainfood"] != null && Request.QueryString["dessert"] != null)
21             {
22                 AccessDataSource1.InsertParameters["mainfood"].DefaultValue = Request.QueryString["mainfood"];
23                 //AccessDataSource1.InsertParameters["num1"].DefaultValue = Request.QueryString["num1"];
24                 //AccessDataSource1.InsertParameters["num2"].DefaultValue = Request.QueryString["num2"];
25                 AccessDataSource1.InsertParameters["drink"].DefaultValue = Request.QueryString["drink"];
26                 AccessDataSource1.InsertParameters["dessert"].DefaultValue = Request.QueryString["dessert"];
27                 //AccessDataSource1.InsertParameters["num3"].DefaultValue = Request.QueryString["num3"];
28                 //AccessDataSource1.InsertParameters["sum"].DefaultValue = Request.QueryString["sum"];
29                 AccessDataSource1.Insert();
30
31                 /*string test = "麥香雞5份";
32                 string test1 = "可樂3杯";
33                 string TEST2 = "雞球2個";*/
34                 string mf, mf_num, dk, dk_num, dt, dt_num;
```

圖 3-43 ASP.NET 程式設計-9

### (10) ASP.NET 程式設計-接收資料做處理



```
37     AccessDataSource1.Insert();
38
39     /*string test = "麥香雞5份";
40     string test1 = "可樂3杯";
41     string TEST2 = "雞球2個";*/
42     string mf, mf_num, dk, dk_num, dt, dt_num;
43     int mf_nt, dk_nt, dt_nt, sum;
44     mf = Regex.Replace(Regex.Replace(Request.QueryString["mainfood"], "[0-9]", ""), "[^0-9]", "");
45     mf_num = Regex.Replace(Regex.Replace(Request.QueryString["mainfood"], "[^0-9]", ""), "[^0-9]", "");
46     dk = Regex.Replace(Regex.Replace(Request.QueryString["drink"], "[0-9]", ""), "[^0-9]", "");
47     dk_num = Regex.Replace(Regex.Replace(Request.QueryString["drink"], "[^0-9]", ""), "[^0-9]", "");
48     dt = Regex.Replace(Regex.Replace(Request.QueryString["dessert"], "[0-9]", ""), "[^0-9]", "");
49     dt_num = Regex.Replace(Regex.Replace(Request.QueryString["dessert"], "[^0-9]", ""), "[^0-9]", "");
50     con = new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + Server.MapPath("Database1.mdb"));
51     con.Open();
52     string data1;
53     data1 = "select a.price, b.price, c.price from 主餐表 a, 飲料表 b, 點心表 c where a.main_name=b.n1,c.dessert_name=b.n2,d.drink_name=b.n3,b.sum from 主餐表";
54     //select a.main_name,b.n1,c.dessert_name,b.n2,d.drink_name,b.n3,b.sum from 主餐表
55     dad = new OleDbDataAdapter(data1, con);
56     ds = new DataSet();
57     dad.Fill(ds);
58     mf_nt = Convert.ToInt32(ds.Tables[0].Rows[0][0]);
59     dk_nt = Convert.ToInt32(ds.Tables[0].Rows[0][1]);
60     dt_nt = Convert.ToInt32(ds.Tables[0].Rows[0][2]);
61     sum = mf_nt * Convert.ToInt32(mf_num) + dk_nt * Convert.ToInt32(dk_num) + dt_nt * Convert.ToInt32(dt_num);
62
63     AccessDataSource2.InsertParameters["mainfood"].DefaultValue = mf;
64     AccessDataSource2.InsertParameters["n1"].DefaultValue = mf_num;
65     AccessDataSource2.InsertParameters["dessert"].DefaultValue = dt;
66     AccessDataSource2.InsertParameters["n2"].DefaultValue = dt_num;
67     AccessDataSource2.InsertParameters["drink"].DefaultValue = dk;
68     AccessDataSource2.InsertParameters["n3"].DefaultValue = dk_num;
69     AccessDataSource2.InsertParameters["sum"].DefaultValue = sum.ToString();
```

圖 3-44 ASP.NET 程式設計-10

## (11) ASP.NET 程式設計-未接收到資料與例外狀況

```

64         AccessDataSource2.InsertParameters["n1"].DefaultValue = mf_num;
65         AccessDataSource2.InsertParameters["dessert"].DefaultValue = dt;
66         AccessDataSource2.InsertParameters["n2"].DefaultValue = dt_num;
67         AccessDataSource2.InsertParameters["drink"].DefaultValue = dk;
68         AccessDataSource2.InsertParameters["n3"].DefaultValue = dk_num;
69         AccessDataSource2.InsertParameters["sum"].DefaultValue = sum.ToString();
70         AccessDataSource2.Insert();
71         con.Close();
72         Response.Write("ok");
73     }
74     else
75     {
76         Response.Write("");
77     }
78 }
79 }
80 catch (Exception ex)
81 {
82     Response.Write(ex);
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 protected void AccessDataSource1_Selecting(object sender, SqlDataSourceSelectingEventArgs
92 {
93 {
94 }
95 }
96 }
    
```

圖 3-45 ASP.NET 程式設計-11

- 實際操作：  
最後店家這邊，就可以等待餐點的通知，由上而下為最新一筆訂單，店家就可以知道顧客先後順序以及餐點(如圖 3-46)。

主餐	數量	點心	數量	飲料	數量	總和
豬排飯	1	大熱狗	1	柳橙汁	1	130
豬排飯	1	大熱狗	1	柳橙汁	1	130
魚排飯	1	薯條	1	綠茶	1	125
雞排飯	1	蔥肉餅	1	柳橙汁	1	145
雞肉飯	1	大熱狗	1	柳橙汁	1	135
雞排飯	1	雞塊	1	綠茶	1	130
三杯雞飯	1	雞塊	1	綠茶	1	125
雞排飯	1	大熱狗	1	咖啡	1	150
三杯雞飯	1	大熱狗	1	咖啡	1	145
三杯雞飯	1	雞塊	1	可樂	1	120

圖 3-46 後臺餐點接收

## 第四節 資料收集-脈絡洞察

### 一、體驗觀察

為了避免使用者操作上問題，造成使用者無法順利完成任務，因此，本研究採用非參與式現場觀察法來了解使用者使用「Zenbo 點餐系統」中的行為模式，也使用參與式觀察法，觀察每個使用流程，使用者可能會有的狀況，藉此了解使用者的需求，做為資料的蒐集方法。

兩者觀察都是以人類生活方式進行各個面項的研究，取此方法可以描述發生了什麼、所涉及的人或物、事發的時間和地點、發生的過程和原因等研究者所關注的問題，即回答何時(When)、在什麼地方(Where)、對哪些對象(Who)、採取哪一種或幾種方式(What)以及如何發生(How)、為什麼發生(Why)等問題。所以本研究觀察重點為活動(Activities)、環境(Environments)、互動(Interactions)、物件(Objects)、使用者(Users)，為 A.E.I.O.U 五種構面(蕭淑玲等，2009)。以下為五個構面之內容敘述與說明：

- A. 活動(A)：在特定的活動中，人們的行為模式為何?會有哪些流程?
- B. 環境(E)：空間使用上是個人空間還是共享空間?空間的特色為何?
- C. 互動(I)：在人與人或人與物件間有哪些互動行為特別、特殊的互動行為?
- D. 物件(O)：使用者的活動環境中有哪些物品和設備?這些物品和設備是跟哪些活動相關?
- E. 使用者(U)：使用者的價值觀及或偏見為何?

非參與式現場觀察法為在不破壞和影響觀察對象的原有結構和內部關係，進行使用者在操作階段的觀察與紀錄，藉此能夠獲得有關較深層的結構和關係的材料；同時，更容易貼近使用者真正行為。

表 3-1 體驗觀察五大構面

觀察構面	觀察重點
活動	1. 使用者點餐時進行哪些事? 2. 使用者點餐時遇到困難點? 3. 使用者點餐時時間與交談內容長短與相關性?
環境	1. 使用者認為點餐系統方便性如何? 2. 使用者認為點餐機器人在現實場景中感覺?
互動	1. 在操作點餐時，需要工作人員協助嗎?

	2. 使用者與機器人間有哪些互動及交談內容?
物件	1. 點餐系統介面是不是關切重點? 2. 機器人樣貌及動作是不是關切重點?
使用者	1. 使用者操作此系統前的觀感? 2. 使用者操作此系統後觀感? 3. 使用者認為機器人點餐系統的認知與價值觀如何? 4. 使用者操作時使用的態度?

## 二、體驗訪談

體驗訪談與傳統一般「你問我答」訪談方式不同，體驗訪談過程須結合「參與式現場觀察法」與「體驗經驗框架」讓使用者在操作同時也接受研究者立即的詢問和觀察，此方法優點能夠真實也豐富當時操作行為，藉此了解使用者內心正在做什麼、想什麼以及為何如此進行。

### (一)參與式現場觀察法

需要在使用者同意下進行特定活動，在從中觀察使用者當時的特殊行為，並在使用者進行每個動作時，馬上進行訪談，藉此了解每個環節使用者的真實狀況。

### (二)體驗經驗框架

在操作系統時，每個系統之服務關鍵接觸點，將他區分出來，用以了解使用者在每個接觸點中使用感受以及其重要行為。以下為本研究以使用者體驗 Zenbo 點餐系統做一次點餐任務時體驗經驗框架(如圖 3-47)

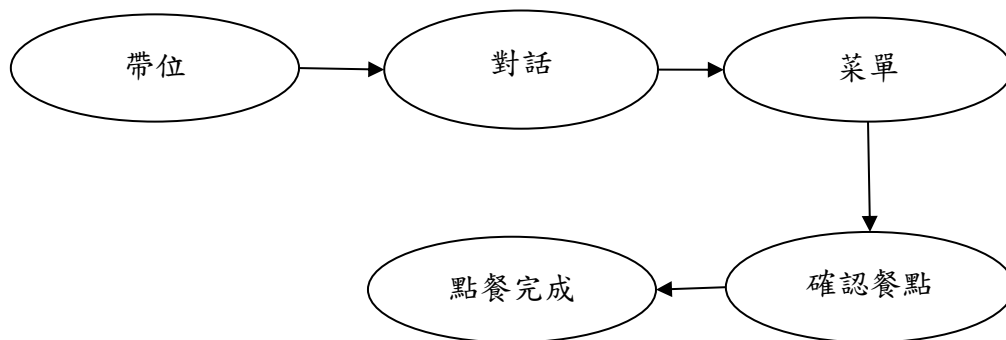


圖 3-47 Zenbo 點餐機器人體驗經驗框架

### (三)行為塑模

脈絡洞察法中的「行為塑模」為脈絡洞察所延續下的活動，其目的為了使團隊內、外的成員能去了解計畫中所有的細部內容，清楚地知道每個環節。更進一步有系統地整理所有觀察到之細節。其最後目的使模型的塑造過程中，能有團隊內、外的共同語言或符號(圖像化)，使整理過後的研究資料能繼續傳遞下去，這種表達方式會比純文字更具體。也因此必須要有「行為塑模」的存在，而「行為塑模」主要分成五大行為模型，分別有：互動模型 (Flow Model)、文化模型 (Cultural Model)、序列模型(Sequence Model)、工具器物模型(Artifact Model)、實體模型(Physical Model)。

#### (1) 互動模型

互動模型主要指使用者在執行某種任務時，會與哪些人、事、物接觸與溝通，藉此協助研究者了解使用者從事活動時，與誰溝通、如何互動等有關內容。藉由互動模型，可以將機器人點餐系統使用者進行「一次點餐」任務時，了解使用者在事件過程中，與其他入、事、物接觸與互動之過程。

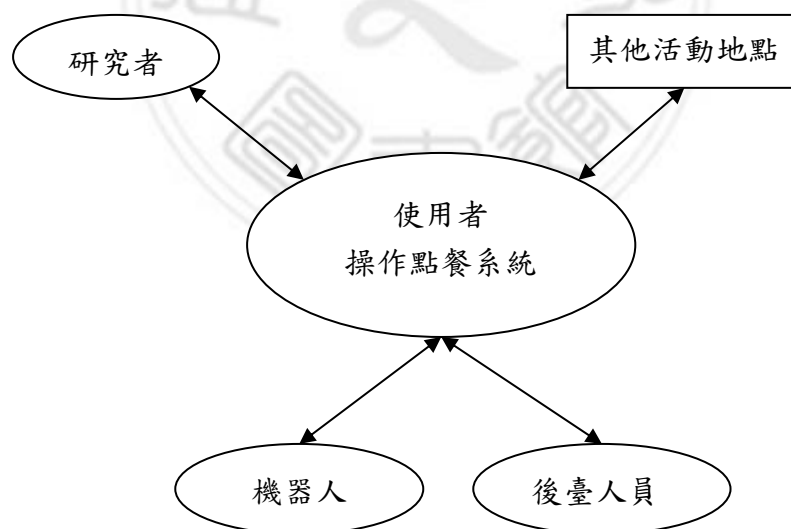


圖 3-48 Zenbo 點餐機器人互動模型

#### (2) 序列模型

將整體活動流程按照先後順序呈現，依序每階段流程的行為，察覺出活動的目的與發現。此模型能使使用者觀察出，互動、溝通的內容為何，看出其動作的意向是什麼，也就是研究者可以從一些不起眼之行為中，發現使用者隱藏的問



題與不便之處。

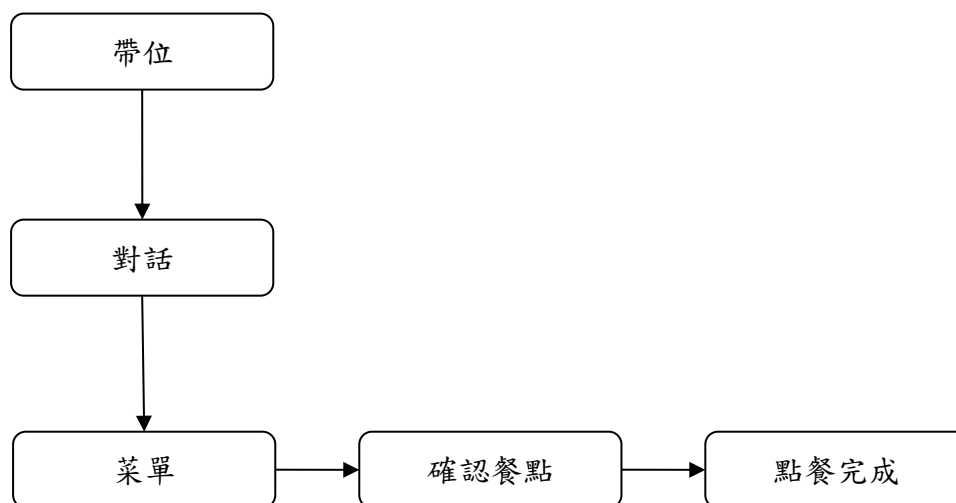


圖 3-49 Zenbo 點餐機器人序列模型

### (3)工具模型

在使用者行為過程中使用的工具，以及工具使用方式、工具的型號、及如何利用工具來達成目的。研究時透過照片的拍攝記錄或繪圖所使用工具並加以說明，可以依此紀錄製成需求表。

### (4)文化模型

指使用者受到內、外因素影響(如:法律、產業規定、工會、情感、態度)後，所受到影響使用者的行為為何。文化模型較難具體化的呈現出受到行為影響之表現。「橢圓形」圖案大小表示使用者受到影響程度越大，多個橢圓形的重疊也可以看出影響者間互相影響之多寡。

### (5)實體模型

以場域作為觀點，根據整體的場地與設備放置區分布，並利用使用者在進行某活動時所觀察出來的動線，清楚了解使用者的流程與顧客的活動為何。而實體模型的設計並不是要像設計師設計室內空間圖，而是標示場地的設備擺設位置、器具分布、使用者該如何運用這些設備及服務的動線，如遇到阻礙及困難則以閃電符號表示。

## 第四章 結果與分析

### 第一節 受訪者基本資料

本研究以服務體驗之需求脈絡洞察的模式，來針對年輕族群來使用 ZENBO 機器人進行點餐體驗，以進行洞察研究。為了找出使用者對於本系統服務產生的需求以及問題，故此本研究過程中採以參與式觀察法以及訪談法來進行，透過上述方式進行，藉此了解現有服務設計缺口，以及對於使用者會產生的困難點與需求。在研究樣本的部分，本研究採隨機徵求有使用過 APP 應用程式的 10 名年輕族群，並在以下作說明本研究在體驗過程中所蒐集資料。

表 4-1 受測者基本資料

編號	性別	年齡	職業	使用 APP 應用程式的經驗	使用過機器人語音控制服務
受訪者一	男	21	學生	有	有
受訪者二	女	23	學生	有	有
受訪者三	女	21	學生	有	無
受訪者四	女	20	學生	有	無
受訪者五	男	21	學生	有	無
受訪者六	女	24	學生	有	無
受訪者七	男	21	學生	有	無
受訪者八	女	22	學生	有	無
受訪者九	男	22	學生	有	無
受訪者十	男	25	學生	有	有

根據受訪者的基本資料(表 4-1)，本研究中全部受訪者都有使用過 APP 應用程式，但是並不是所有受訪者有過機器人服務的經驗。未有過機器人體驗服務經驗佔本樣本的多數，但也能更貼近使用者的需求，因為目前機器人服務也並未普及，從這些角度而言，本研究以參與式觀察以及訪談法，希望了解受訪者對於本研究設計之服務體驗所遇到的問題，以作為後續改善本研究系統服務的問題，藉此希望能更貼近使用者需求。

## 第二節 受訪者服務體驗與訪談

經過 10 名年輕族群測試者體驗後，多次訪談分析，並以活動(Activities)、環境(Environments)、互動(Interactions)、物件(Objects)、使用者(Users)五構面來呈現出使用者在體驗機器人點餐系統真實狀況與觀感。以下內容為受訪者接受的問題，並以編號來區別受訪者：

### 一、活動(Activities)：

本研究以受訪者在實驗參與過程中，受訪者所遭遇困難點以及在操作點餐流程中與機器人交談之內容為觀察主軸，並提出了以下問題：

#### (一)訪談設計：

以本研究機器人點餐系統，讓使用者點完一份套餐，並讓機器人帶位到位子上的整體過程。

(1)問題一：你在過程中，對於點餐系統會注意到哪些地方？

(2)問題二：與機器人互動過程中，你有哪些感覺？

#### (二)訪問內容：

(1)受測者一：我是有用過類似機器人，但是點餐的還是第一次滿有趣的。有趣歸有趣，但既然是跟簡餐結合，應該會有些介紹吧，不然其實我都不知道有哪些餐點比較划算。

(2)受測者二：整體而言滿一般的，都沒有介紹或是說明，會讓我第一次來不知道點哪些餐點。

(3)受測者三：整體過程還滿有趣的，但是在點餐時，餐點訊息太少，說不定先由機器人介紹餐點開始會較好，而且語句對話時，有時機器人聽不懂我說的話，會讓我不耐煩。

(4)受測者四：還滿有趣的，步驟經過說明也沒那麼複雜，第一次會比較不知道如何操作。

(5)受測者五：整體而言還不錯，而且點餐過程也滿快速的，有搭配機器人來介紹餐點應該會更好。

(6)受測者六：機器人太可愛了，整體步驟有經過說明還滿簡單的，但是對於一般人可能就不一樣了，可以讓他跟我們說阿，這樣我會覺得更有趣，不然就感覺呆呆的，像平板點餐一樣。

(7)受測者七：整體畫面就很一般，而且機器人點餐我可能還要低下頭看他內容上的文字，滿不方便的。

(8)受測者八：還滿好玩的，希望機器人未來可以做更多事，用途會很廣，但是這個機器人怎麼有時候會聽不清楚我說的。

(9)受測者九：整體而言還滿簡單有趣的，但是感覺他再帶位時候會卡卡的，走走停停會讓我覺得到很煩。

(10)受測者十：進去點餐都沒有介紹我一下，而且定完餐畫面字滿小的，帶我到位子還不錯，但是一下走一下停，讓我覺得他好呆，並不像我所想像的機器人依樣聰明。

## 二、環境(Environments)

### (一)訪談設計：

本研究主要呈現讓受測者在測試當下所模擬的一次點餐任務，在期間觀察受測者的反應與便利性。

(1)問題一：對於在點餐過程的便利性如何？

(2)問題二：點餐帶位呈現於店家環境的感覺如何？

### (二)訪問內容：

(1)受測者一：在操作過程還算滿方便的，會自動帶我到位子上座，但是會比較慢，有時還要等他判斷路線。

(2)受測者二：人過來就自動幫我帶位真的還滿不錯的，操作也滿便利的，只是動作有點慢，是不是人太多了，造成這樣的呢？

(3)受測者三：就是帶位有點卡，不然整體呈現還滿便利的，如果中途叫他又會停止動作，是否可以加個突發事件之類的。

(4)受測者四：點餐過程也滿快速的，不用在那邊排隊，如果未來可以全面實行，應該是滿便利的，只不過帶位時會卡住，還需要等他移動。

(5)受測者五：常在轉彎時他會停頓一下，還需要等他移動，不然整體點餐也算便利，而且可以直接跟他點餐，也不用移動到櫃檯，未來是不是可以運用到其他地方，整體而言還不錯。

(6)受測者六：機器人還滿有趣的，還會跟你對話，整體而言還算不錯，希望可以用到更多地方，一定會很方便。

(7)受測者七：人走動時會有點亂，機器人也會停頓一下，其實會讓我覺得有點不方便。

(8)受測者八：點餐還滿簡單的，可以直接對他下命令，而且可以跟他對話，真好玩。

(9)受測者九：機器人在移動會停頓找路，讓我覺得很不方便，有時候還會有走錯方向問題。

(10)受測者十：很有趣的機器人，還會帶位，真的很厲害，就是帶位時會停頓一下，不然整體來說都很棒。

### 三、互動(Interactions)

#### (一)訪談設計：

本研究在受測者進行點餐任務時，觀察機器人與受測者間對話的語句，以了解受測者對於機器人的直接反應。

(1)問題一：與機器人對話之間會問些什麼？

(2)問題二：是否會遇到困難？如何解決？

#### (二)訪談內容：

(1)受測者一：我會問他一些餐點的問題，有些有答案，但是很多機器人都聽不懂，其他是沒什麼問題。

(2)受測者二：我有時候在遇到不懂的地方，機器人還沒有我清楚，遇到困難時，如果有緊急通知，就不會僵持一段時間。

(3)受測者三：帶位時，無法聽到我講話，整體而言還算可以，對話就直接點餐，快速方便。

(4)受測者四：機器人帶位時，是不是無法對話，與他對話都不理我，如果我想叫他暫停動作或是換位子都沒有反應。

(5)受測者五：一開始我會跟機器人說我要點餐，機器人就只有直接問你需要哪些餐點，如果中間過程多加些介紹的話會更好。

(6)受測者六：簡餐店都會有特餐之類的，但是我問機器人並沒有回答，感覺對話有點少，應該介紹一下店裡的資訊。

(7)受測者七：機器人有些動作還不錯，但是跟他對會都只能點餐的問題，其他的互動就比較少。

(8)受測者八：有些我講的話聽不懂，我遇到操作不會他會稍微提醒，但是整體而言還需要加強。

(9)受測者九：遇到點餐失敗，機器人會重新提醒，但是說明不夠仔細，還有帶位他都不會聽見我說的，這樣我覺得不太方便。

(10)受測者十：點餐過程中，會出現一些問題，比如餐點重新點，機器人會提示你，但是有些我還是不瞭解，會造成操作上的困難。

#### 四、物件(Objects)

##### (一)訪談設計：

本研究主要測試受測者在進行點餐任務時，觀察機器人的臉部表情以及動作是否會影響使用者觀感。

(1)問題一：你觀察到哪些機器人的動作與表情？

(2)問題二：這些設計搭配機器人感覺如何？

##### (二)訪談內容：

(1)受測者一：機器人動作主要有點頭、搖頭之類的，每次對話完還會一直笑，都讓我笑出來了，還滿有趣的。

(2)受測者二：臉部表情很多，還可以與機器人間互動，非常有趣，或許還可以讓機器人來場表演，會更有趣。

(3)受測者三：點餐完成還會轉一圈，語句上不通時，還會有問號表情，整體還算滿有趣的。

(4)受測者四：整體來說還不錯，尤其是每次對話完都會有回應動作與表情，還以為機器人都是一板一眼的。

(5)受測者五：原本以為機器人設計會很呆版，動作會卡卡的，整體都還算有趣，動作上還滿流暢的。

(6)受測者六：超可愛的，還會有害羞的表情，互動感十足，未來說不定可以跳舞表演之類的。

(7)受測者七：動作雖然比較簡單，但是跟機器人對話時都會有回饋，感覺很好，可以設計更多動作，會更好。

(8)受測者八：會點頭與搖頭，每次對話都會有回應，還不錯，臉部表情也還滿有趣的。

(9)受測者九：每次對話時都會有回應設計的還不錯，希望未來加上更多設計，會更有趣。

(10)受測者十：有些小動作是不錯的，感覺是有兩邊是有在交流的，不會讓我感到很生硬尷尬。

## 五、使用者(Users)

### (一)訪談設計：

本研究主要觀察受測者在測試前後對於機器人點餐時有什麼樣的想法，以此來做後續探討使用者對於使用前後需求。

(1)問題一：(測試前)你對機器人點餐有什麼想法與期待？

(2)問題二：(測試後)你對機器人點餐有什麼想法與期待？

### (二)訪談內容：

(1)受測者一：在測試前我有用過類似的機器人，以前與機器人溝通還沒有那麼順暢；在測試前問，也是讓我覺得溝通較沒有那麼順暢，不然其他設計還不錯，動作表情不會像我以前使用過的呆板，未來希望可以改善溝通上的不方便。

(2)受測者二：剛開始會有很多想法，讓機器人繳費付款、帶位和點餐，還覺得滿期待的；測試過後，雖然沒有付款的動作，但是整體來說還算不錯，未來說不定可以加入這塊。

(3)受測者三：點餐系統結合機器人一開始沒接觸過，還滿期待的；後來接觸後，整體還算不錯，後續這塊應該可以更多發展。

(4)受測者四：一開始覺得機器人的動作應該會很多停頓，講話應該頻率會不太好，用在點餐會不會很奇怪；測試後，覺得還不錯，動作簡單但不會有太多停頓點，講話會有一定的頻率但是搭配表情看起來沒那麼呆版，以後說不定真的可以看見機器人在做服務。

(5)受測者五：機器人對於我們來說應該是滿方便的，用在點餐應該會不錯；測試後，對話時有些話機器人聽不太懂，遇到點餐錯誤時，有需操作不太方便了解，未來希望可以改善操作這塊。

(6)受測者六：一開始覺得機器人應該是比較聰明的；測試後感覺不錯，但是沒有想像中的聰明，還是需要特定的關鍵字才有反應，不過機器人還能帶位我覺得整體還是算不錯的。

(7)受測者七：機器人點餐應該會滿有趣的，一開始還會擔心很難控制，後來測試後，覺得操作起來還算簡單的，而且還會有動作和表情，真的滿有趣的。

(8)受測者八：機器人會不會比較難控制，點餐完成後還會帶位，還滿期待的；測試後覺得操作步驟還是需要多說明，以及點餐訊息太少，希望後續能改善。

(9)受測者九：機器人點餐第一次接觸，還滿期待的，剛開始還在想會不會很難控制，後來是覺得還好，還是要花點時間才能點餐，應該要多點說明，才方便使用者來操作。

(10)受測者十：一開始還滿期待的，後續操作過後也覺得很不錯，機器人對話起來會有動作表情，還滿有趣的。

表 4-2 A.E.L.O.U 五構面問題彙整

A.E.L.O.U 五構面	問題敘述	需求點
活動(Activities)	介面資訊單調	1. 系統畫面設計須改善 2. 提高系統內容豐富度
	餐點未有進一步說明	1. 系統簡介餐點應與本系統結合
環境(Environments)	機器人帶位會較慢	1. 場地限制問題 2. 多人走動影響機器人狀態
互動(Interactions)	機器人語句互動不足	1. 建立機器人模擬點餐對話腳本
	例外狀況無法掌握	1. 須建立機器人狀況點即通知櫃台人員 2. 回報訊息，結合後端作為管理
物件(Objects)	機器人擬人化影響使用者的觀感	1. 應以擬人化口語來呈現 2. 機器人表情動作給予更多社交狀態的感覺產生
使用者(Users)	使用者對於機器人點餐之價值觀(使用前)	1. 對於機器人點餐的新鮮感 2. 對於機器人點餐期望值很高



	使用者對於機器人點餐之感覺與態度(使用後)	<ol style="list-style-type: none"> <li>1. 對機器人點餐模式感到新鮮感</li> <li>2. 對於機器人語句感到有趣</li> </ol>
--	-----------------------	--

### 第三節 五大行為模型

本研究為了解機器人點餐系統之服務需求，以服務體驗脈絡洞察法中五大彙整行為模型，透過非參與式以及參與式觀察法，將過程中觀察以及訪談並蒐集受測者在期間內、外在行為資料加以統整，以此了解受測者真實的需求。

#### 一、彙整互動模型

透過此互動模型彙整，將受測者當下所與機器人互動過程中，當下所互動現象，最後經過整理與歸納，來探討機器人點餐系統，因為互動中所產生的問題與遭遇困難，並記錄受測者真實狀況，以了解使用者真實的需求點。

表 4-3 互動模型資料彙整

問題	狀況說明	需求點
對於機器人操作不熟悉	點餐時會有些許狀況，造成使用者會不知所措	建立簡易操作說明，並用語音做互動防呆
語音語句互動不足	在對話過程中，較少其他互動語句設計	增加對話腳本

#### 二、彙整序列模型

本研究透過先後順序的排序，將每個流程過程，加以記錄與彙整，了解使用者在一些不起眼的行為中，所遭受到的問題與不方便性。本研究記錄各個步驟，在經過統計與彙整，了解使用者其實對於服務設計是感到很有趣的，對話以及動作這設計，讓受測者感到新鮮感，大多受測者也是可以自行操作完成點餐，但是還是有一些問題發生，將這些問題整理如下圖 4-4。

表 4-4 序列模型資料彙整

問題	狀況說明	需求點
使用者在對話時間為最久	產生新鮮感，但是對於使用者會造成點餐時間拉長	引導式步驟，但如何在互動與效率間達到平衡

### 三、彙整文化模型

文化模型主要呈現受測者內在的影響，以了解使用者對於點餐機器人真實的感受與行為，本研究以影響機器人點餐系統文化因素，分為以下問題，在使用經驗上的差異以及語言上不同，所遭遇的整體問題匯總，以了解受測者真實行為中的需求點。

表 4-5 文化模型資料彙整

問題	狀況說明	需求點
使用經驗	大多使用者並未有過使用機器人經驗	說明機器人對話腳本與系統功能
特殊口語對話	部分使用者點餐會有自己的口語表達方式	以引導式簡化點餐時關鍵語句，例如：是或否來判斷

### 四、彙整工具模型

工具模型主要是以研究者透過工具是如何完成任務，並觀察在這些任務中所面臨的問題。本研究點餐系統是以機器人所完成的，也因此以機器人為主要任務受測者使用的工具，也因此觀察重點為受測者對於機器人設計的問題為重點，以下為本研究受測者所提出問題。

表 4-6 工具模型資料彙整

問題	狀況說明	需求點
點餐介面單調	點餐時，對於商家的簡介或是餐點介紹較為不足	需加強系統內容豐富度
機器人臉型動作	部分使用者點餐會覺得機器人臉型與動作不搭配，互動性不足	加強點餐時，動作與臉部表情設計

### 五、彙整實體模型

實體模型主要為在任務執行過程，透過實體的環境，了解空間環境上，有哪些限制與約束。本研究為以簡餐店為例，在一次點餐過程中，環境因素造成哪些問題為觀察重點。

表 4-7 實體模型資料彙整

問題	狀況說明	需求點
障礙物多	由於店家擺設，會造成機器人帶位時速度較慢	建立較好的動線，以供機器人移動

## 六、五大模型彙整

以下為綜合五大行為模型所彙整的問題，與當時的狀況說明，以了解使用者的內、外在因素行為，以及整體工具與環境的影響，歸納出使用機器人點餐系統之需求如下表 4-8

表 4-8 五大模型問題與需求彙整

問題	狀況說明	需求點
對於機器人操作不熟悉	點餐時會有些許狀況，造成使用者會不知所措	建立簡易操作說明，並用語音作互動防呆
語音語句互動不足	在對話過程中，較少其他互動語句設計	增加對話腳本
使用者在對話時間為最久	產生新鮮感，但是對於使用者會造成點餐時間拉長	引導式步驟，但如何在互動與效率間達到平衡
使用經驗	大多使用者並未有過使用機器人經驗	說明機器人對話腳本與系統功能
特殊口語對話	部分使用者點餐會有自己的口語表達方式	以引導式簡化點餐時關鍵語句，例如：是或否來判斷
點餐介面單調	點餐時，對於商家的簡介或是餐點介紹較為不足	需加強系統內容豐富度
機器人臉型動作	部分使用者點餐會覺得機器人臉型與動作不搭配，互動性不足	加強點餐時，動作與臉部表情設計
障礙物多	由於店家擺設，會造成機器人帶位時速度較慢	建立較好的動線，以供機器人移動

## 第五章 結論與建議

### 第一節 研究結論

本研究透過服務體驗工程法中的脈絡洞察法進行，探討在點餐一次任務中所表現出的，五個構面活動(Activities)、環境(Environments)、互動(Interactions)、物件(Objects)、使用者(Users)；以及互動模型、文化模型、序列模型、工具模型以及實體模型分析，以了解受測者所遇到困難以及問題，並加以彙整這些資料進一步找出使用者需求點以及設計上的不足之處，以下為本研究探討所設計的機器人點餐系統之研究結論。

#### 一、建立簡易操作流程：

大多數都是第一次接觸機器人點餐系統，操作上最後都可以完成，但是由於對於流程上還是不太熟悉，點餐過程其實比實際沒有用機器人點餐還要慢，造成其實點餐速度會有點慢，所以須建立操作簡易說明，當點餐時提醒使用者接下來該做哪些步驟，並設計一個緊急通知功能，讓使用者能遇到無法解決問題時，可以及時與店家溝通。

#### 二、豐富機器人語句設計：

受訪者對於機器人語句設計感覺不夠多，會有時聽不懂受測者的語句，並且語句會有點制式化，應多增加一些互動性，但是如何在點餐效率與互動性之間做平行，是需要做更深入的評估的，並且有些特殊口語化語言，需要避免掉，可以設計成關鍵字來做判斷，以減少對話時機器人的錯亂。

#### 三、增加機器人的真實性與動作豐富性：

一般機器人會有讓人覺得呆版的印象，造成機器人會有點遲鈍或是不專業的感覺，受訪者覺得大部分設計是滿有趣的，但是整體來說還是跳脫不了機器人的問題，希望增加機器人在動作與畫面配合的豐富度，才不會讓使用者感到不夠符合真實感。

#### 四、機器人帶位突發狀況以及速度改善

機器人在帶位過程中，會遇到過多障礙物，造成速度緩慢問題，受訪者期待能夠改善帶位速度問題，而且在帶位過程中，若有些突發狀況或是問題，無法控制機器人，這也是本系統需要改善的地方。

## 第二節 研究建議

一、本研究因為由於時間以及可行性因素，研究樣本只有 10 位受測者，也因此實驗的深度以及服務需求程度會有需不足，因此建議後續研究者可以增加研究樣本，以便更深入訪談研究，探討更多不同使用者的不同需求，讓機器人點餐系統能夠更加完善。

二、隨著人工智慧技術的進步，慢慢提升了我們生活上的便利性，未來也會有更多人工智慧應用於器人的結合，本研究成果也可以做為未來更多人在開發設計時參考依據，藉由本研究要點，設計出更多機器人系統，使我們的生活能更加便利。

## 第六章 參考文獻

### 一、中文部份

1. 陳睿銓(2017)。社交平台經濟模式對傳統餐飲業未來趨勢與發展。國立清華大學碩士論文。
2. 段伴虬、陳聰堅、蘇德義(2013)。以服務體驗洞察觀點形塑銀髮族遠距健康照顧服務需求之研究。南開科技大學福祉科技與服務管理系。
3. 黃靖璇(2017)。台灣機器人產業之未來發展策略角色。國立雲林科技大學企業管理系。
4. 蔡靜佩(2017)。服務型機器人導入餐飲服務之分析設計與規劃。國立勤益科技大學流通管理系碩士論文。
5. 黃韋誠(2018)。以服務體驗洞察退休銀髮族進行失智評估之桌遊設計。國立雲林科技大學工業設計系碩士論文。
6. 曹書齊(2012)。智慧型關節五軸機械手臂設計開發。義守大學機械與自動化工程學系碩士論文。
7. 卓人杰(2011)。對話代理人需求塑模方法況理論模式電腦疲勞防護之研究。國立台東大學資訊管理學系碩士論文。
8. 林聖熙(2010)。基於知識本體之對話代理人社群塑模與轉換。國立臺東大學資訊管理學系碩士論文。

9. 戴耀賽(2014)。服務體驗工程運用於非營利組織經營之個案研究 - 以慢飛兒咖啡坊為例。中華大學企業管理學系碩士論文。
10. 王熙哲(2010)。應用服務體驗工程方法於銀髮族家事服務系統設計。龍華科技大學資訊管理所碩士論文。
11. 彭康宇(2011)。服務體驗工程與層級結構分析法解析虛擬社群網站服務與使用者需求間關係之研究。清雲科技大學資訊管理所碩士論文。
12. 林鶴樺(2012)。運用服務體驗工程方法導入雲端資訊應用系統 - 以 ERP 為例。逢甲大學工業工程與系統管理學研究所碩士論文。
13. 謝伊萍(2013)。智慧型販賣機之服務設計-服務體驗工程方法之應用。明新科技大學服務事業管理研究所碩士論文。
14. 洪素蜜(2014)。應用服務體驗洞察法於銀髮族手杖使用之功能需求探討。南開科技大學福祉科技與服務管理所碩士論文。
15. 荅蒺宓啟茵(2017)。服務體驗工程方法應用於部落文化健康站餐飲服務之研究。南開科技大學福祉科技與服務管理所碩士論文。
16. 蘇政勛(2018)。比較以使用者經驗方法與服務體驗工程之設計流程與產品差異，以電動捲線器為例。國立雲林科技大學工業設計系碩士論文。
17. 溫九力(2016)。遠端體感控制機械手。義守大學機械與自動化工程碩士論文。
18. 財團法人資訊工業策進會創新應用服務研究所(2008)。服務體驗工程方法指引(初版)。台北市：經濟部技術處。
19. 盧賢豪(2014)。知識管理系統中應用智慧型代理人之研究。國立中山大學資訊管理研究所。

## 二、西文部份

1. Rehman, M. S. (2011). Human Resource Practices and Job Performance in Pakistan.
2. Decker, M., Fischer, M., Ott, I.(2017).Service Robotics and Human Labor.
3. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., ... & Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age.
4. Belter, D., Nowicki, M., & Skrzypczyński, P. (2016). Accurate map-based RGB-D SLAM for mobile robots.
5. Stuart Russell & Peter Norving.(2013). “Artificial Intelligence : a modern approach”.
6. Wooldridge, M., Jennings.NR & Kinny,D.(2000). Autonomous Agents and Multi-Agent Systems.
7. Panti, M.,et.al.(2000). A FIPA compliant agent platform for federated information systems.
8. Beyer, H., & Holtzblatt, K. (1998). Contextual design: Defining customer-centered systems. Morgan Kaufmann Publishers.
9. Li, R. Y. M., Li, H. C. Y., Mak, C. K., & Tang, T. B. (2016). Sustainable Smart Home and Home Automation: Big Data Analytics Approach.
10. Wooldridge, M. Jennings, N. Kinny, D(2000.).The Gaia Methodology for Agent-Oriented Analysis and Design Autonomous Agents and Multi-Agent Systems Volume 3, Issue 3.
11. M. C. Hsieh, W. S. Hung, S. S. Shin, S. W. Lin and T. H. Huang(2008). “Spoken dialogue agent interface requirements modeling based on PASSI methodology,” Proceedings of the 8th International Conference on Intelligent Systems Design and Applications.
12. M., Penserin, L., Spalazzi, L., and Valenti, S. (2000). A FIPA Compliant

### 三、網際網路部分

1. International Federation of Robotics(2018)。Service robots – global sales value up 39 percent。取自  
<https://ifr.org/ifr-press-releases/news/service-robots-global-sales-value-up-39-percent>。
2. 陳威珞 (2018)。產業分析：餐飲業發展趨勢。取自  
[http://www.twtrend.com/share\\_cont.php?id=63](http://www.twtrend.com/share_cont.php?id=63)。
3. Global Industry Analysts(2013)。全球機器人產業發展強勁。取自  
[https://www.digitimes.com.tw/iot/article.asp?cat=158&id=0000335900\\_v87lzu3gl1kkhr127al3s](https://www.digitimes.com.tw/iot/article.asp?cat=158&id=0000335900_v87lzu3gl1kkhr127al3s)。
4. 華碩 Zenbo 語音對話建立。取自 <https://zenbo.asus.com/tw/>。
5. 楊仁達 (2013)。創新思維與創新方法。服務創新電子報。取自  
<https://innoservice.org/>。