****

**AI-智能餐點機器人 ---開放人生小組出品3.0**

**指導老師：陳柏豪**

**2018年6月20日**

# Contents

1. **Introduction5**
   1. [Purpose 5](#_TOC_250028)
   2. [Document Conventions 5](#_TOC_250027)
   3. [Intended Audience and Reading Suggestions 5](#_TOC_250026)
   4. [Project Scope 5](#_TOC_250025)
   5. [References 5](#_TOC_250024)
2. Overall Description6
   1. [Product Perspective 6](#_TOC_250023)
   2. [Product Functions 6](#_TOC_250022)
   3. [User Classes and Characteristics 6](#_TOC_250021)
   4. [Operating Environment 6](#_TOC_250020)
   5. [Design and Implementation Constraints 7](#_TOC_250019)
   6. [User Documentation 7](#_TOC_250018)
   7. [Assumptions and Dependencies 7](#_TOC_250017)
3. External Interface Requirements8
   1. [User Interfaces 8](#_TOC_250016)
   2. [Hardware Interfaces 8](#_TOC_250015)
   3. [Software Interfaces 8](#_TOC_250014)
   4. [Communications Interfaces 8](#_TOC_250013)
4. System Features9
   1. [System Feature 1 9](#_TOC_250012)
      1. [Description and Priority 9](#_TOC_250011)
      2. [Stimulus/Response Sequences 9](#_TOC_250010)
      3. [Functional Requirements 9](#_TOC_250009)
   2. [System Feature 2 (and so on) 9](#_TOC_250008)
5. Other Nonfunctional Requirements10
   1. [Performance Requirements 10](#_TOC_250007)
   2. [Safety Requirements 10](#_TOC_250006)
   3. [Security Requirements 10](#_TOC_250005)
   4. [Software Quality Attributes 10](#_TOC_250004)
   5. [Business Rules 11](#_TOC_250003)
6. Other Requirements12
   1. [Appendix A: Glossary 12](#_TOC_250002)
   2. [Appendix B: Analysis Models 12](#_TOC_250001)
   3. [Appendix C: To Be Determined List 12](#_TOC_250000)

小組成員

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名： | 林辰晏 | 閻俊宇 | 曹均 | 黃錦立 | 陳紀佑 |
| 學號： |  |  |  | 1041471 |  |

# 版本歷史

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| 初始 | 6.10 | 16 | 1.0 |
| 進階 | 6.16 | 25 | 2.0 |
| 最終 | 6.21 | 32 | 3.0 |

1. **引言**

## 目的

*<*在快節奏的社會，點餐方式將不僅僅限於人工服務，智能機器人的存在將大大節省人力成本，同時帶來高效益的匯報.我們小組所開發的產品將以此為出發點，打造一款可以為用戶帶來點餐服務，同時可在地性的結合天氣情況，給用戶提供反饋與建議。*>*

## 文件約束

*<*Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.*>*

## 目標讀者和閱讀建議

*<* 本份描述文檔不僅僅針對少部分的讀者，而是面向所有的讀者，包括開發人員、項目經理、測試人員以及文檔編寫人員。在程式軟體的開發過程中，我們的這份描述文檔將會採用最通俗易懂語言介紹，讀者只需要按序從頭開始閱讀，便可以對我們的產品有一個清晰的認識.*>*

## 項目範圍

*<*我們小組所開發的這款軟體主要是應用於點餐服務，可以使業者大大減少成人力成本的投入，同時提高顧客點餐的效率，縮短顧客的等待時常，進一步優化企業的收益。在短期時間內獲得較好的反饋，長遠時間利於業者站穩時常，擁有核心競爭力。*>*

## 參考

*<*王選仲，玩轉資料與機器學習-以自然語言處理為例系列

---https://ithelp.ithome.com.tw/users/20107576/ironman/1450?page=1

Wolke，Microsoft Bot Framework 30天上手+自幹Line Builder串接+Line Bot Demo

---https://ithelp.ithome.com.tw/users/20046160/ironman/1510?page=1

Duran Hsieh，利用 Ms Bot framework與Cognitive Service建構自用智慧小秘書

---https://ithelp.ithome.com.tw/users/20091494/ironman/1411?page=1

maduka，建立Microsoft LUIS的App服務，進行語意是別的訓練並整合Bot framework

---https://dotblogs.com.tw/maduka/2016/06/24/113852*>*

# 總體描述

## 產品透視

*<*我們小組團隊所開發的產品是我們依據網路上前輩的經驗所改造而成，是我們的初代產品，主要是使用了Microsoft公司旗下提供的框架與接口。*>*

## 產品功能

*<*Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.*>*

## User Classes and Characteristics

*<*Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, tech- nical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.*>*

## Operating Environment

*<*Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or appli- cations with which it must peacefully coexist.*>*

## Design and Implementation Constraints

*<*Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific tech- nologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).*>*

## User Documentation

*<*List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user docu- mentation delivery formats or standards.*>*

## Assumptions and Dependencies

*<*List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).*>*

# External Interface Requirements

## User Interfaces

*<*Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.*>*

## Hardware Interfaces

*<*Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.*>*

## Software Interfaces

*<*Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and inte- grated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed applica- tion programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.*>*

## Communications Interfaces

*<*Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any com- munication standards that will be used, such as FTP or HTTP. Specify any communica- tion security or encryption issues, data transfer rates, and synchronization mechanisms.*>*

# System Features

*<*This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.*>*

## System Feature 1

*<*Don’t really say “System Feature 1.” State the feature name in just a few words.*>*

### Description and Priority

*<*Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).*>*

### Stimulus/Response Sequences

*<*List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.*>*

### Functional Requirements

*<*Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.*>*

*<*Each requirement should be uniquely identified with a sequence number or a mean-

ingful tag of some kind.*>* REQ-1: REQ-2:

## 4.2 System Feature 2 (and so on)

1. **Other Nonfunctional Requirements**

## Performance Requirements

*<*If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.*>*

## Safety Requirements

*<*Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.*>*

## Security Requirements

*<*Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.*>*

## Software Quality Attributes

*<*Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, avail- ability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantita- tive, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.*>*

## Business Rules

*<*List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional require- ments in themselves, but they may imply certain functional requirements to enforce the rules.*>*

# Other Requirements

*<*Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse ob- jectives for the project, and so on. Add any new sections that are pertinent to the project.*>*

## Appendix A: Glossary

*<*Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.*>*

## Appendix B: Analysis Models

*<*Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.*>*

## Appendix C: To Be Determined List

*<*Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.*>*