SOFTWARE REQUIREMENTS SPECIFICATION

for

<chatroom robot assistant>

Version 1.0 approved

Prepared by :
1031408 劉彥呈
1031452 何浩璘
1033329 林仕翔
1033336 尹法堯
1041459 梁澤洲

<開放平台軟體 第15組>

June 26, 2018

Contents

Revision History

學號	Name	Date	Reason For Changes	Version
1031408	劉彥呈	5/19	顯示時間	1
1033336	尹法堯	5/19	Client進入訊息	2
1031408	劉彥呈	5/20	Client離開訊息	3
1033336	尹法堯	5/20	區分相同的name	4
1033336	尹法堯	6/14	完成資料庫所有功能和Server新增帳號	5
1031408	劉彥呈	6/14	完成資料庫與介面的互動	5
1031408	劉彥呈	6/17	顯示再現人數	7
1033336	尹法堯	6/18	Server刪除帳號	8
1031408	劉彥呈	6/19	Server踢除Client	9
1031452	何浩璘	6/19	Bot加入以及weather-api的使用	10
1031408	劉彥呈	6/19	更換介面的顏色	11
1031452	何浩璘	6/23	Bot增加匯率轉換-api和顯示日期區域時間	12

1 Introduction

1.1 Purpose

該文件提供cahtroom robot assistant Version12的相關說明，包含改產品的開發理念、最終 成果，產品的提供對象及適用範圍，系統的程式碼架構，不同物件之間的上下關係，程式運行環境以及產品特色

1.2 Document Conventions

<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.>

1.3 Intended Audience and Reading Suggestions

本文件適合以下職位的人員觀看，部分內容須有程式邏輯相關知識:
（1）專案經理：專案經理可以根據該文檔瞭解預期產品的功能，以此進行系統設計。
（2）設計員：對需求進行分析，並設計出系統，包括資料庫的設計。
（3）程式師：瞭解系統功能，對後續更新版本或產品維護提供程式架構。
（4）測試員：根據本文檔對軟體產品進行功能性測試和非功能性測試。
（5）銷售人員：瞭解預期產品的功能和性能。
（6）用戶：瞭解預期產品的功能和性能，與分析人員對整個需求進行討論和協商。

1.4 Project Scope

該產品的開發理念爲節省使用者在聊天室與夥伴聊天時，爲查詢生活中的小資訊而必須頻繁切換應用程式的時間 ，希望使用者可以在與夥伴聊天時，在時間上有很好的使用者體驗，或者討論單日行的旅遊，或者與分隔兩地的朋友聊天時， 可以方便的查詢當地天氣相關資訊以提供聊天話題。

1.5 References

1. https://pypi.org/project/weather-api/
2. https://pypi.org/project/currency.converter/

3. http://pyqt.sourceforge.net/Docs/PyQt5/
4.https://github.com/mongodb/mongo-python-driver
5.https://pip.pypa.io/en/stable/

2 Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Functions

<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.>

2.3 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, technical 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.>

2.4 Operating Environment

1. OS: Window7 以上版本 (未在其他作業系統測試過)
2. 開發語言: python3
3. 使用python api: weather-api, currency_converter
4. 其他python開發套件: socket, threading, os, sys, time, PyQt5
5. 資料庫: mongoDB

2.5 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 technologies, 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).>

2.6 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 documentation delivery formats or standards.>

2.7 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).>

Title											
1	2	3	4	5	6	7	8	9	10	11	12

3 External Interface Requirements

3.1 User Interfaces
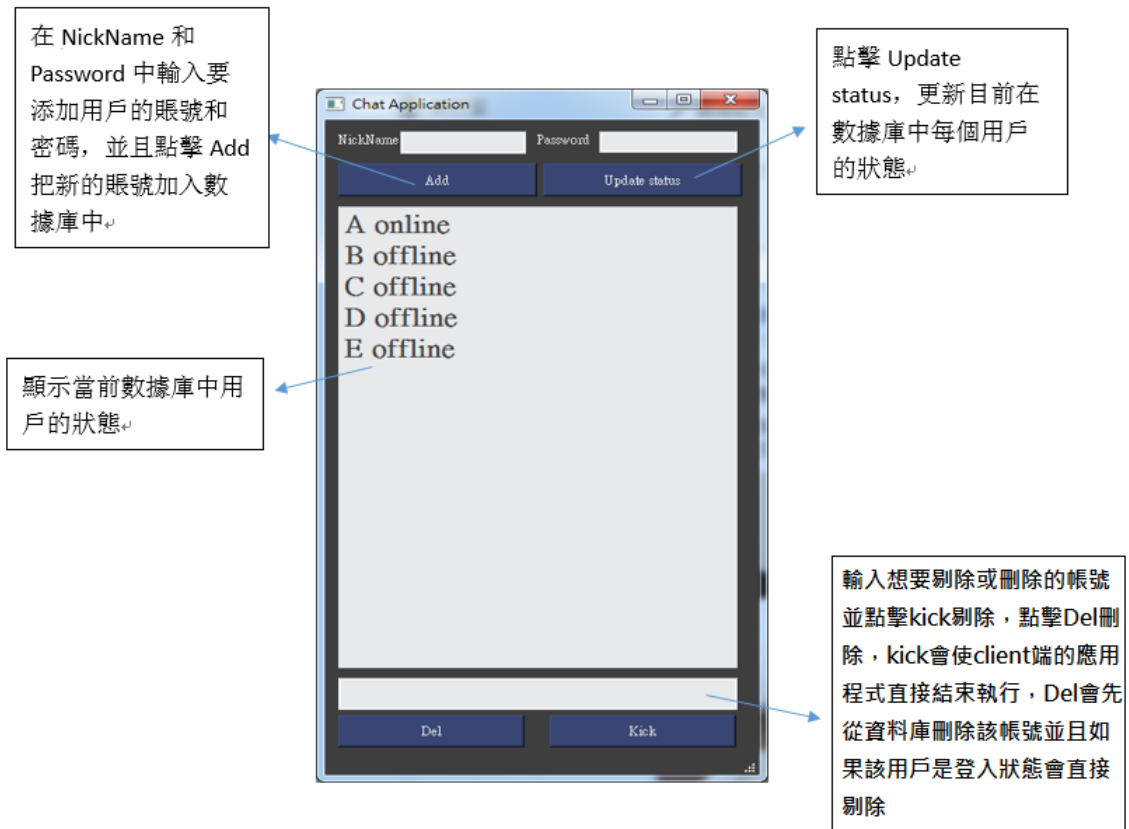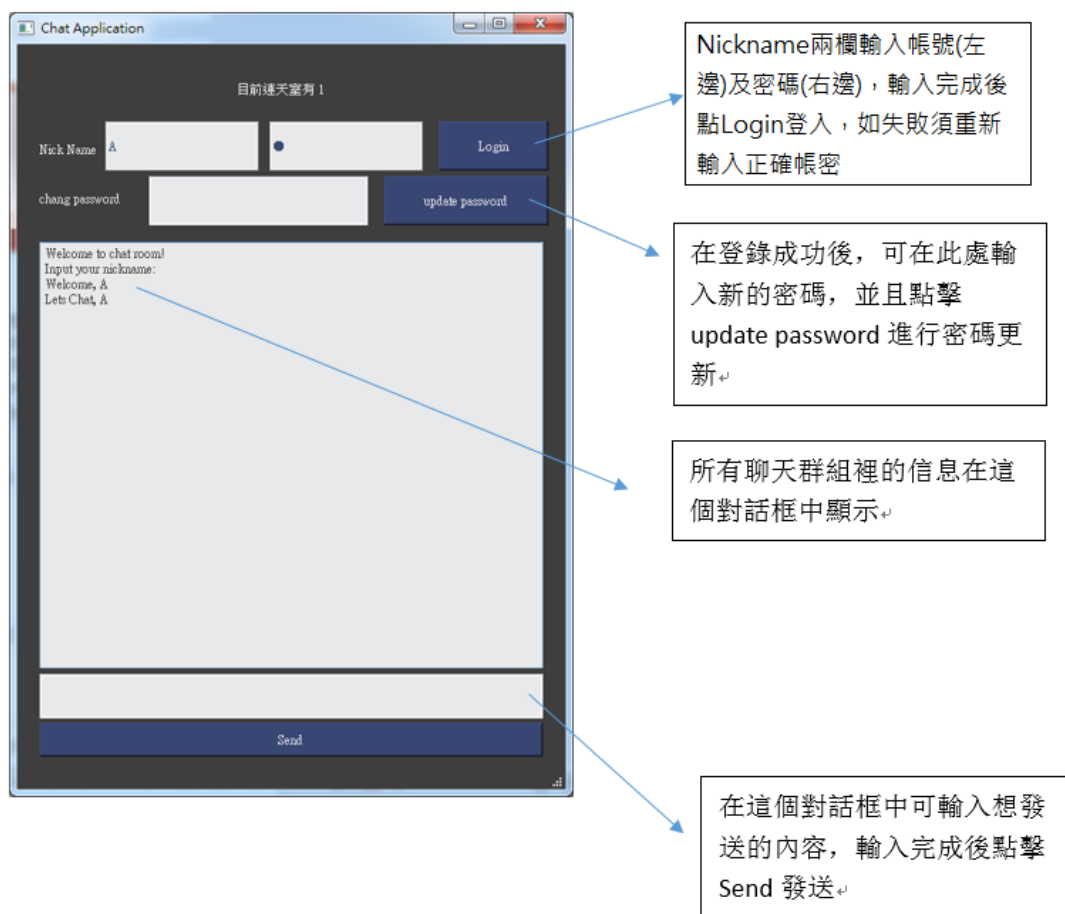

Figure 3.1: sever interface

Figure 3.2: client interface

3.2 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.>

3.3 Software Interfaces

一、 資料庫
(1) mongoDB 3.6：用於管理帳號密碼以及狀態資訊
二、python 套件

(1) socket：用於處理server與client之間的訊息傳遞

(2) threading：用與處理接收訊息、傳遞訊息、GUI之間的線程管理

(3) os：用於強制讓程式結束運作

(4) time：用於顯示發送或接收訊息時，當下的時間資訊

(5) PyQt5 5.10.1：用於呈現server與client的UI介面、按鈕事件控制

(6) pymongo 3.6.4：python用於控制mongoDB的函示庫

(7) weather-api 1.0.4：用於取得某地的天氣資訊，輸入爲地區名稱，輸出會當地溫度、天氣狀況、濕度等資訊

(8) CurrencyConverter 0.13.5：用於轉換兩個不同的貨幣，輸入爲金額、原始貨幣、轉換後貨幣，輸出爲轉換後貨幣的金額

3.4 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 communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4 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.>

4.1 Fast Search Weather and Exchange rate Information

該特色可以幫助使用者快速的取得特定地區的天氣資訊以及兩個不同貨幣之間的貨幣量值轉換

4.1.1 Description and Priority

基於希望使用者可以直接在聊天室取得簡單的資訊並節省時間的目的，我們初步完成了能查詢天氣資訊和匯率轉換的功能。這兩項功能的好處是，假如三五好友利用閒暇的時間在討論想去哪裡玩時，可以直接向連天室的機器人詢問當地氣候，可以讓連天室的討論快速的進行下去，又或者與分隔兩地的人交談時，可以快速的查詢對方當地的氣候，順利的找到話題並繼續聊天。匯率的部分可以在討論該去哪個國家玩時，或者已經決定要去哪個國家時，可以方便的查詢目前的匯率決定換錢的時機，這兩個特色可以簡單的就在聊天室向機器人詢問取得，而不需要再切換額外的應用程式查詢。

4.1.2 Stimulus/Response Sequences

以下舉例使用者實際上需要輸入的句子以及得到的答案:
(1) 使用者輸入 - bot weather 桃園 機器人回復 - Taoyuan City Mostly Clear 27
表示桃園目前的天氣晴朗，並且溫度是27度
(2) 使用者輸入 - bot convert 1000 JPY USD 機器人回復 - 1000 JPY=9.065600246097056 USD
使用首先輸入bot convert，接著輸入須轉換金額，接著輸入轉換的原始貨幣再輸入轉換後貨幣，以上面的例子，我們可以得到1000日幣約等於9美元

4.1.3 Functional Requirements

(1) 詢問氣候 : 使用者登入後在client端的UI上輸入 — bot weather 桃園 — 指令，bot幫助程式判斷我現在要詢問的對象是機器人，接著weather代表要詢問的事情是天氣的資訊，地點是桃園，server收到訊息後，從指令中取得weather-api所需的字句，取得答案後將答案送給client，client進而把取得的資訊顯示在UI上

(2) 詢問匯率轉換：使用者在client端的UI輸入 — bot convert 1000 JPY USD — 指令，bot幫助程式判斷我現在要詢問的對象是機器人，接著convert代表要詢問的是貨幣的轉換，需要轉換的金額是1000，原始貨幣是日圓，轉換後貨幣是美元，server收到訊息後，從指令中取得CurrencyConverter所需的字句，取得答案後將答案送給client，client進而把取得的資訊顯示在UI上

5 Other Nonfunctional Requirements

5.1 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.>

5.2 Safety Requirements

Client在發送訊息時，會有兩個指令會造成錯誤：
(1)systeM==!!：是使用者登入時，告訴Server現在有一個用戶要登入了，並統計人數，告訴所有的Client現在的在線人數是多少。
(2)kick==：它是Server會發送訊息告訴Client，經由Client辨別，來決定是否主動斷開連線。
在Server介面時，當有兩個以上使用者在登入時，要等待所有使用者登入完畢，不能有一位登入，其他還未登入時，使用Kick功能，這會造成Server的損壞。

5.3 Security Requirements

目前只有針對帳號密碼登入時，把它們與資料庫進行比對，比對錯誤將會有訊息顯示在client端的UI要求重輸，沒有其他包含針對資料庫或連線進行惡意攻擊的防護措施

5.4 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, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 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 requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6 Other Requirements

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

6.1 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.>

6.2 Appendix B: Analysis Models

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

6.3 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.>