|  |
| --- |
| 國立臺灣大學  106學年度第一學期軟體工程課程 |
| Software Requirement Specification |
| Shake and Chat |
|  |
| 2017 Fall軟體工程 |
| 指導教授：李允中教授 |
| 成員:  王瀚磊  陳立誠  李旭恩  綦家志  王本奕  高睦修 |

|  |
| --- |
|  |

# Outline

[Outline 2](#_Toc355013563)

[Revision history 3](#_Toc355013564)

[1. System Architecture 4](#_Toc355013565)

[1.1 Introduction 4](#_Toc355013566)

[1.2 Architecture Expression 5](#_Toc355013567)

[2. Functional Requirement 6](#_Toc355013568)

[2.1 Front-end Functional Requirements 6](#_Toc355013569)

[2.2 Back-end Functional Requirements 7](#_Toc355013570)

[3. Interface Requirement](#_Toc355013571) 8

[3.1 External Interface Requirements](#_Toc355013572) 8

[3.2 Internal Interface Requirements](#_Toc355013573) 9

[4. Nonfunctional Requirement](#_Toc355013574) 9

[5. Goal-driven Use Case Diagram 10](#_Toc355013575)

[5.1 Introduction to Goal-driven Approach 10](#_Toc355013576)

[5.2 Goal-driven Use Case Diagram 11](#_Toc355013578)

[5.3 Use case specification 12](#_Toc355013580)

[6. Traceability Matrix](#_Toc355013581) 12

[6.1 Traceability Matrix of Requirements V.S Requirements 12](#_Toc355013582)

[6.2 Traceability Matrix of Requirements V.S Use Case 12](#_Toc355013583)

[6.3 Traceability Matrix of Requirements V.S subsystem function 12](#_Toc355013584)

Revision history

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版次 | 負責人 | 日期 | 變更項目敘述 | 審查者 | 審查日期 |
| 0.1 | 李旭恩  王本奕 | 10/28/17 | 建立SRS文件架構 | 全體組員 | 10/29/17 |
| 0.2 | 王瀚磊 | 10/31/17 | 修改Introduction  新增Interface Requirements | 全體組員 | 10/31/17 |
| 0.3 | 陳立誠 | 11/6/17 | 新增architecture, UC, FR | 全體組員 | 11/6/17 |
| 0.4 | 陳立誠 | 11/7/17 | UC2,7 目錄除錯 | 全體組員 | 11/7/17 |

1. System Architecture

## 1.1 Introduction

隨著資訊科技的進步，現代人的交友範圍從現實逐漸虛擬化，利用網路作為交友平台的軟體如雨後春筍般出現。

我們團隊決定反其道而行，雖然利用網路，但卻希冀能利用網路，使人們能踏出螢幕，重新建立人與人之間的實際互動。

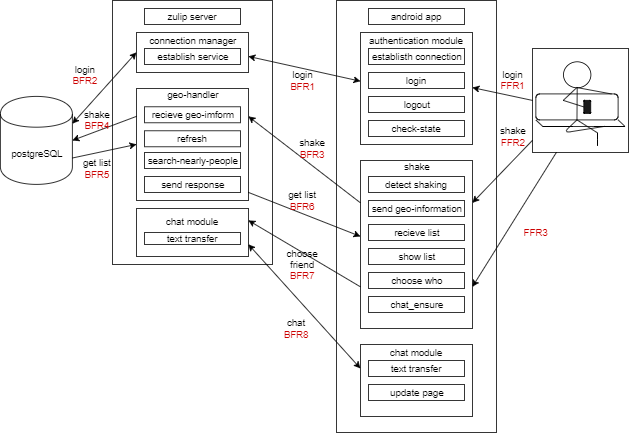
這款在zulip上透過搖一搖手機配對附近之陌生人並使之交流的功能將支援定位系統，讓使用者可以為自己發出「訊號」找尋在一定範圍內一樣在線上活動的使用者，搜尋的範圍可以彈性設定，訊號廣播的時間也很自由，因應安全性上的疑慮，亦可過濾是否要搜尋完全不認識的陌生人，或是只需要搜尋在線的好友名單。專案強調以地點、持續時間為主軸，來進行朋友的搜尋建議。並期望可利用此系統，使線上交友不只是在虛擬平台，更達成人與人之間的真心交流。

專案內容主要包含網頁APP的設計、訊息傳遞設計、資料庫演算法設計、後端管理介面等項目。

## 1.2 Architecture Expression

Following are the main functionalities of the system and system architecture:

|  |  |
| --- | --- |
| Module | Descriptions |
| Geochat Module | Get loacation information of client and response to server , and broadcast to other users nearby. |
| Chat Module | Origianl functionality of zulip. |



1. Functional Requirement

2.1 Front-end Functional Requirements

|  |  |
| --- | --- |
| FFR1 | Login front-end UI |
|  | This function provides a login interface to allow users to login zulip. |
| FFR2 | **Shake and Chat Page** |
|  | The function provide a page that acknowledge user to start shaking and show the nearby users |
| FFR3 | **Choose Friend** |
|  | Provide an interface to see friends who are nearby and just shook.  Show the name of theses people and “CHAT” button for each. |

* 1. Back-end Functional Requirements

|  |  |
| --- | --- |
| BFR1 | Login and establish connection |
|  | Client Android app sends login request to server , and establishes a new connection , and checks connection state. |
| BFR2 | **Login and tell database** |
|  | Server receives the login information, and updates the database. |
| BFR3 | **Send Shaken Signal to Server** |
|  | After a user has shaken, send the shaken message to server |
| BFR4 | **Update Shaken Message to Database** |
|  | Server receives that a user shaken, write the message into the database. |
| BFR5 | **Server Get Nearby User List from Database** |
|  | This function provides a server interface to allow the server to get all the other user information that is nearby the specific user. The user information includes account and geometric location. |
| BFR6 | **Server send the Nearby User information to Client** |
|  | This function provides the interface that allows the server to send all the nearby user information to client. The user information includes the user account that allow the user to chat via it, and geometric location which allow to show the distance between the users in the client application. |
| BFR7 | **Choose friend** |
|  | The app send a number from 0 to 3 corresponding to the none, first, second, third person on the list. |
| BFR8 | **Chat** |
|  | The system get message from sender, and transmit message to receiver. |

1. Interface Requirements

3.1 External Interface Requirements

|  |  |
| --- | --- |
| EIR1 | Geochat button |
|  | The user can go to geochat by clicking a button. When the user enter the geochat, the user can shake the mobile and chat with all the people nearby. |
| EIR2 | **Geochat list** |
|  | When the users shake the mobile, all the people nearby the user will show on the list. |
| EIR3 | **Chatroom** |
|  | After showing the list of the people nearby, the user can choose one/multiple user for chatting. The chatting page is same as zulip’s original design. |

3.2 Internal Interface Requirements

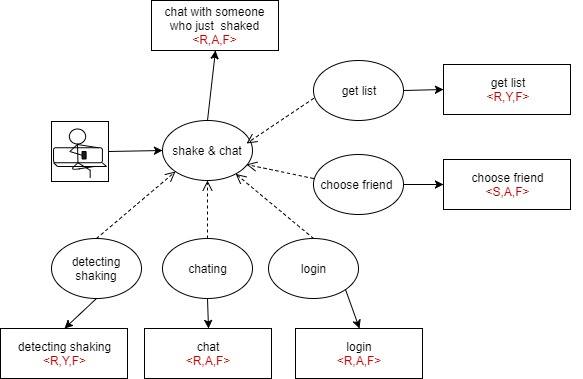
|  |  |
| --- | --- |
| IIR1 | Upload user’s GPS location |
|  | After the user shaking the mobile, the mobile will send a request to server which means this account allow to be found by other user who is using shake and chat. Besides, the mobile will upload the GPS location to server 1 time per second after shaking, and keep for 5 minutes. |
| IIR2 | **Get the information of the people who is nearby** |
|  | The server will send the response that includes all the information of the user who is nearby. |
| IIR3 | **Disable for being searched by shake and chat** |
|  | Other user who is using shake and chat for 5 minutes can search each user. After 5 minutes, the server will disable the user for being searched by shake and chat. |

# 5. Goal-driven Use Case Diagram

## 5.1 Introduction to Goal-driven Approach

本專案採用Goal Driven(目標導向)Use Case表示，強化表現出本系統所想達到的目標，以及此系統的非功能性需求的描述，以及和使用案例的相依關係，此系統分為三個層面去做分析：能力面、觀點面、內容面。根據能力面，判斷此目標是否該被完全滿足，分出強制性的目標和非強制性的目標。根據觀點面，判斷目標若是以角色觀點來描述，將之稱為角色相關；若與系統相關，稱作系統相關。在從內容面判斷，若此目標為系統上定義的功能，稱作功能性目標；相反的將之稱為非功能性目標。

## 5.2 Goal-driven Use Case Diagram



## 5.3 Use case specification

|  |  |
| --- | --- |
| Use Case ID | UseCase-001 |
| Use Case Name | Users login to the zulip |
| Goal | Users logins to the zulip chat system and update the date in server and database. |
| Requirements | [FFR1 : Login front-end UI]  [BFR1 : Login and establish connection]  [BFR2 : Login and tell database] |
| Description | Users login to the system to enjoy the service. |
| Actor |  |
| Assumptions |  |
| Constraints |  |
| Priority | High |
| Pre-Conditions | 1. The user has valid account and password. 2. The user has been invited to a zulip group. |
| Post-Conditions |  |
| Basic Flow | 1. The user input the correct account and password. 2. Login to the system, and then backend will establish a connection, and update the data in database. |
| Alternative Flows | 1. If user doesn’t have account, redirect to register page. |
| Exceptional Flows |  |
| Use Use Case |  |
| Extend Use Case | [UseCase-002] Detect Shaking |
| Business Rules |  |
| Special Requirements |  |
| Artifacts |  |
| Use Case Glossary |  |

|  |  |
| --- | --- |
| Use Case ID | UseCase-002 |
| Use Case Name | Shake Interface |
| Goal | A page that acknowledge user to start shaking |
| Requirements | [FFR2 Shake] |
| Description | When a user want to chat with nearby people, he/she can enter the page and start shaking |
| Actor | User |
| Assumptions |  |
| Constraints |  |
| Priority | High |
| Pre-Conditions | A user click the shake and chat button |
| Post-Conditions | Enter the shake page and detect if a user is shaking or not |
| Basic Flow | 1. User click the shake button 2. Enter the shake page 3. User decide to shake or cancel |
| Alternative Flows |  |
| Exceptional Flows |  |
| Use Use Case |  |
| Extend Use Case |  |
| Business Rules |  |
| Special Requirements |  |
| Artifacts |  |
| Use Case Glossary |  |

|  |  |
| --- | --- |
| Use Case ID | UseCase-003 |
| Use Case Name | Show the nearby user information |
| Goal | Display all the information of the user that is nearby, and let the user to decide who he/she wants to chat with. |
| Requirements | [BFR5: Server Get Nearby User List from Database]  [BFR6: Server send the Nearby User information to Client] |
| Description | Display all the information of the user that is nearby, and let the user to decide who he/she wants to chat with. |
| Actor |  |
| Assumptions |  |
| Constraints |  |
| Priority | High |
| Pre-Conditions | 1. The user needs to login first. 2. The mobile detects that the user is shaking mobile. |
| Post-Conditions | The nearby user information will display on the mobile that can let the user decide who he/she wants to chat with. |
| Basic Flow | 1. The server will collect the information of the user who is nearby the specific user. 2. After collecting, server will send all the user information to client application. When the client receives the information, it will display in a specific format. |
| Alternative Flows | If server cannot collect any information of the nearby user, the server can alternative increase the threshold, deciding who is nearby. |
| Exceptional Flows | If the user leave the shake and chat page when the client application is receiving the information of nearby user, the client will drop all the request/response. |
| Use Use Case |  |
| Extend Use Case | [UseCase-002] Detect Shaking |
| Business Rules |  |
| Special Requirements |  |
| Artifacts |  |
| Use Case Glossary |  |

|  |  |
| --- | --- |
| Use Case ID | UseCase-004 |
| Use Case Name | Choose friend |
| Goal |  |
| Requirement | [FFR3 Choose friend]  [BFR3 Choose friend]  [BFR7 Choose friend] |
| Description | Show 0~3 people who are nearby and just shook.  Let user choose a friend from these people. |
| Actor | User |
| Assumptions |  |
| Constraints |  |
| Priority | High |
| Pre-Conditions | The user just shook its device. |
| Post-Conditions | The app sends the user index (0~3) to the server. |
| Basic Flow | 1. Show up 0~3 name and “Chat” button for each. 2. If “Chat” is pressing, send the signal (1~3). |
| Alternative Flows | If the interface is closed, send the signal (0). |
| Exceptional Flows |  |
| Use Use Case |  |
| Extends Use Case |  |
| Business Rules |  |
| Special Requirements |  |
| Artifacts |  |
| Use Case Glossary |  |

|  |  |
| --- | --- |
| Use Case ID | UseCase-005 |
| Use Case Name | Transmit Messages |
| Goal | Achieve “Communicate Conveniently” |
| Requirement | [BFR8 Chat] |
| Description | A user can transmit messages to specific users. |
| Actor | User |
| Assumptions |  |
| Constraints | One user can just |
| Priority | High |
| Pre-Conditions | System admits user can send message to a designated user. |
| Post-Conditions | System sends the message to receiver. |
| Basic Flow | 1. User gets into send message page. 2. User writes the message. 3. User sends the message. 4. System gets the message. 5. System sends message to receiver. 6. Receiver’s app show the message. |
| Alternative Flows |  |
| Exceptional Flows | System shows “Your message is blocked.” |
| Use Use Case |  |
| Extends Use Case |  |
| Business Rules |  |
| Special Requirements |  |
| Artifacts |  |
| Use Case Glossary |  |

|  |  |
| --- | --- |
| Use Case ID | UseCase-006 |
| Use Case Name | Shake phone and Chat |
| Goal | Achieve “Shake to talk with stranger” |
| Requirement | [FFR1 Login]  [FFR2 Shake]  [FFR3 choose friend]  [BFR1 Login]  [BFR2 Login]  [BFR3 Shake]  [BFR4 Shake]  [BFR5 Get List]  [BFR6 Get List]  [BFR7 Choose Friend]  [BFR8 Chat] |
| Description | A user can get into “Shake it” mode, checking who is online nearby, and chat with a person on the list. |
| Actor | User |
| Assumptions |  |
| Constraints |  |
| Priority | High |
| Pre-Conditions | User open the app. |
| Post-Conditions | User close app or logout. |
| Basic Flow | 1. User open app. 2. User login 3. System check login 4. User gets into “shake it” page. 5. User send request to get list 6. System get list 7. User check list and pick one stranger 8. System update database 9. System check receiver can chat 10. User send message 11. System get message. 12. System send message to receiver. 13. User logout app |
| Alternative Flows | User may close app abruptly, system has to check whether user is online. |
| Exceptional Flows |  |
| Use Use Case |  |
| Extends Use Case |  |
| Business Rules |  |
| Special Requirements |  |
| Artifacts |  |
| Use Case Glossary |  |

|  |  |
| --- | --- |
| Use Case ID | UseCase-007 |
| Use Case Name | Send shaken message to server |
| Goal | Server know that the user is shaken, update database |
| Requirement | [BFR2 Shake]  [BFR3 Shake]  [BFR4 Shake]  [FFR2 Shake] |
| Description | When a user has shaken the phone, server needs to update the database; therefore, if other people shake, they can see each other. |
| Actor |  |
| Assumptions |  |
| Constraints |  |
| Priority | High |
| Pre-Conditions | A user has entered the shake page and shaken the phone. |
| Post-Conditions | Database updated |
| Basic Flow | 1. User has shaken their phone 2. Front-end detect the shake and send the user’s location and time to server 3. Server update the database |
| Alternative Flows |  |
| Exceptional Flows |  |
| Use Use Case |  |
| Extends Use Case |  |
| Business Rules |  |
| Special Requirements | Server should save the time the user shaken since afterwards the system should return the user a user list that has shaken in a small period of time |
| Artifacts |  |
| Use Case Glossary |  |

# 6. Traceability Matrix

## 6.1 Traceability Matrix of Requirements V.S Requirements

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 需求 | FFR1 | FFR2 | FFR3 | BFR1 | BFR2 | BFR3 | BFR4 | BFR5 |
| FFR1 | **--** |  |  | **V** | **V** |  |  |  |
| FFR2 |  | **--** |  |  |  | **V** | **V** |  |
| FFR3 |  |  | **--** |  |  | **V** |  |  |
| BFR1 | **V** |  |  | **--** | **V** |  |  |  |
| BFR2 | **V** | **V** |  | **V** | **--** |  | **V** |  |
| BFR3 |  | **V** | **V** |  |  | **--** | **V** |  |
| BFR4 |  | **V** | **V** |  | **V** | **V** | **--** |  |
| BFR5 |  |  |  |  |  |  |  | -- |
| BFR6 |  |  |  |  |  |  |  | **V** |
| BFR7 |  |  |  |  |  | **V** |  |  |
| BFR8 |  |  |  |  |  |  |  |  |
| 需求 | BFR6 | BFR7 | BFR8 |
| FFR1 |  |  |  |
| FFR2 |  |  |  |
| FFR3 |  |  |  |
| BFR1 |  |  |  |
| BFR2 |  |  |  |
| BFR3 |  | **V** |  |
| BFR4 |  |  |  |
| BFR5 | **V** |  |  |
| BFR6 | **--** |  |  |
| BFR7 |  | **--** |  |
| BFR8 |  |  | **--** |

## 6.2 Traceability Matrix of Requirements V.S Use Case

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 需求 | UC  001 | UC  002 | UC  003 | UC  004 | UC  005 | UC  006 | UC  007 |
| FFR1 | **V** |  |  |  |  | **V** |  |
| FFR2 |  | **V** |  |  |  | **V** | **V** |
| FFR3 |  |  |  |  |  | **V** |  |
| BFR1 | **V** |  |  |  |  | **V** |  |
| BFR2 | **V** |  |  |  |  | **V** | **V** |
| BFR3 |  |  |  | **V** |  | **V** | **V** |
| BFR4 |  | **V** |  |  |  | **V** | **V** |
| BFR5 |  |  | **V** |  |  | **V** |  |
| BFR6 |  |  | **V** |  |  | **V** |  |
| BFR7 |  |  |  | **V** |  | **V** |  |
| BFR8 |  |  |  |  | **V** | **V** |  |

## 6.3 Traceability Matrix of Requirements V.S subsystem function

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 需求 | Connection manager Module | Geo-handler Module | Chat(server) Module | Authentication Module | Shake Module | Chat(client) Module |
| FFR1 |  |  |  | **V** |  |  |
| FFR2 |  |  |  |  | **V** |  |
| FFR3 |  |  |  |  | **V** |  |
| BFR1 | **V** |  |  | **V** |  |  |
| BFR2 | **V** |  |  |  |  |  |
| BFR3 |  | **V** |  |  | **V** |  |
| BFR4 |  | **V** |  |  |  |  |
| BFR5 |  | **V** |  |  |  |  |
| BFR6 |  | **V** |  |  | **V** |  |
| BFR7 |  |  | **V** |  | **V** |  |
| BFR8 |  |  | **V** |  |  | **V** |