|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Subject | QPlay QChat APP - PIS | | Doc.: | S2-PIS | | Rev. | V1.0 |
|  | | | Release Date: | 2017/07/06 | | Revision Status: | |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
| Approved By: | | Reviewed By: | | | Prepared By: | | |
| PM / | | QA / | | | SD / | | |
|  | | SE / | | | SE / | | |
|  | | SD / | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |

[1.0](#_gjdgxs) 系統設計 (System Design) 4

[1.1](#_30j0zll) 系統架構圖 (System Framework Diagram) 4

[1.2](#_1fob9te) 系統流程圖 (System Flow) 5

[1.3](#_3znysh7) 類別圖 (Package/Class Diagram) 6

[2.0](#_2et92p0) 資料庫設計 (Database Design) 6

[2.1](#_tyjcwt) 資料庫物件清單 (Database Object List) 6

[2.2](#_1t3h5sf) 循序圖 (Sequence Diagram) 7

[2.3](#_4d34og8) 資料表結構 (Table Layout) 8

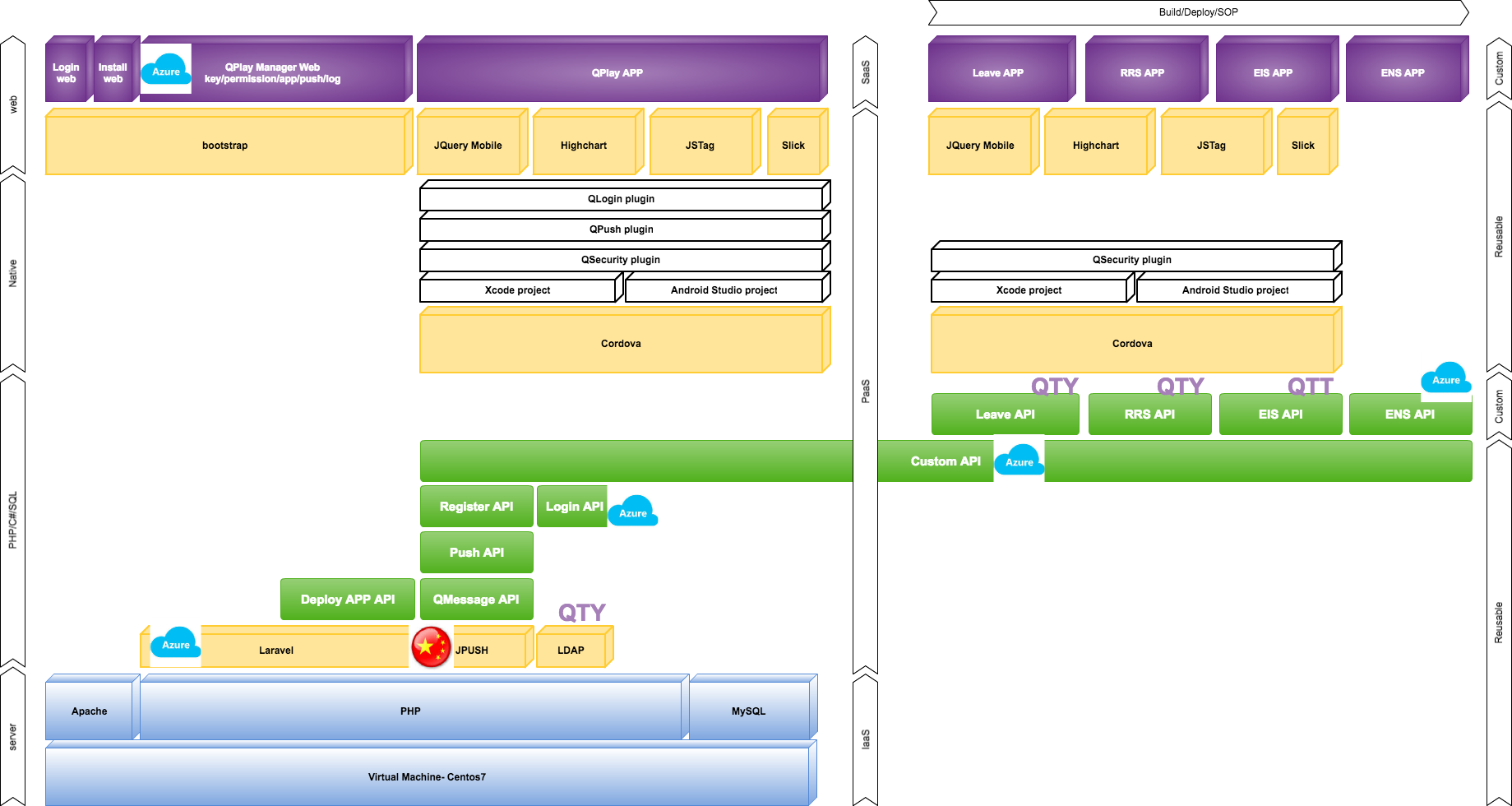
[3.0](#_2s8eyo1) Implementation 8

[3.1](#_17dp8vu) 公用元件 (Public Component) 8

[3.2](#_3rdcrjn) 共用界面 (Public Interface) 9

# 系統設計 (System Design)

## 系統架構圖 (System Framework Diagram)



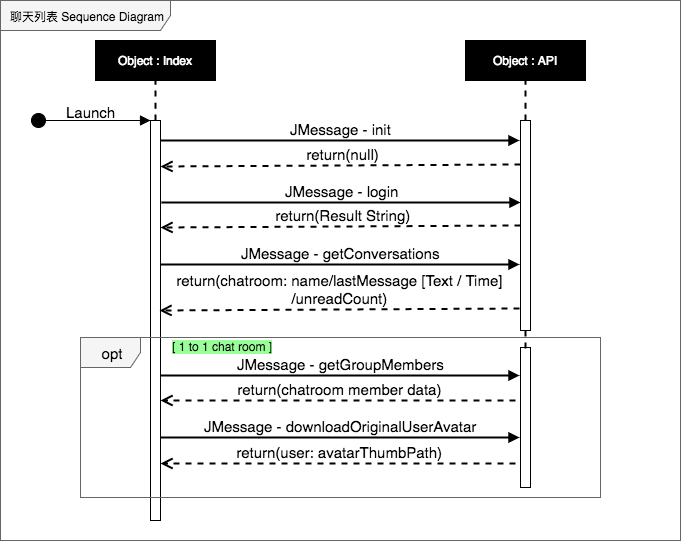
## 系統流程圖 (System Flow)

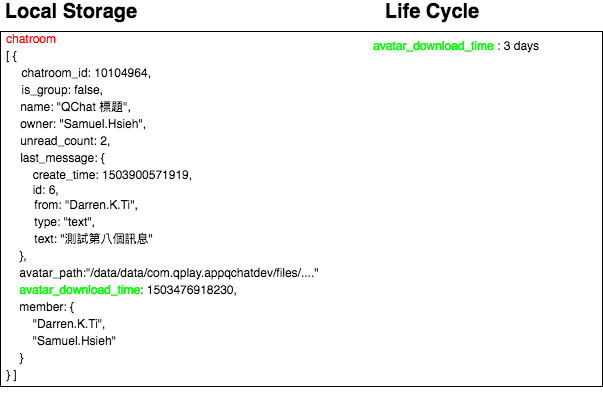
## 類別圖 (Package/Class Diagram)

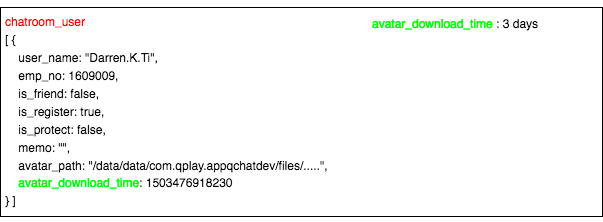
# 資料庫設計 (Database Design)

## 資料庫物件清單 (Database Object List)

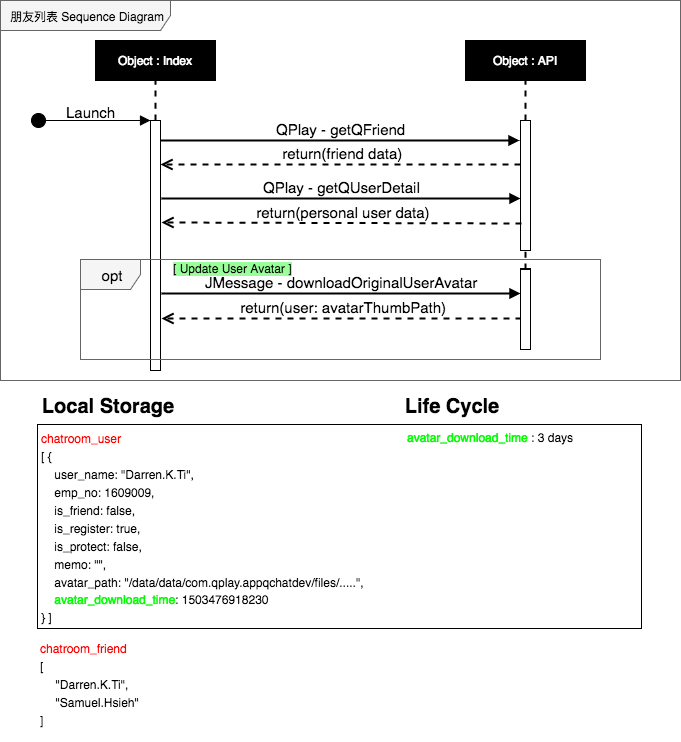
## 循序圖 (Sequence Diagram)



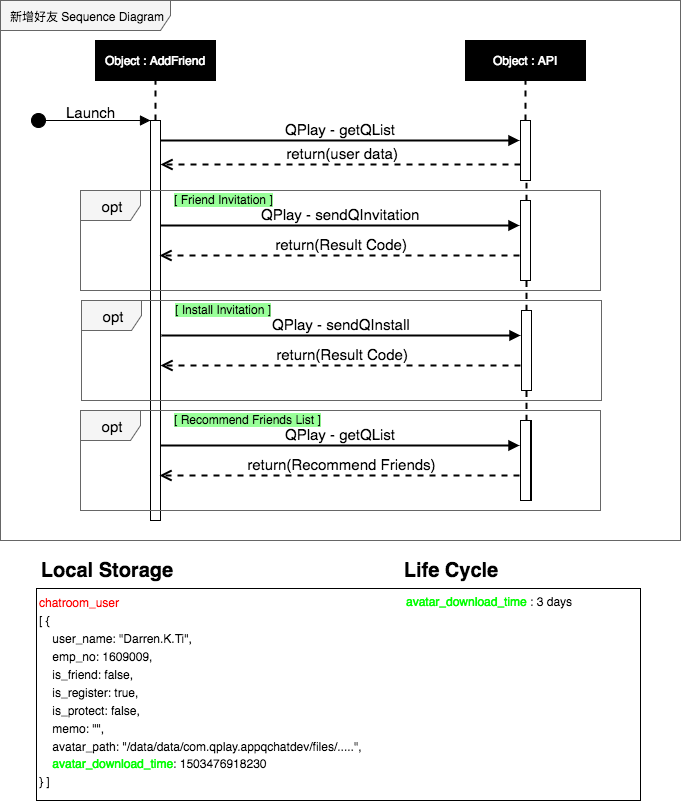




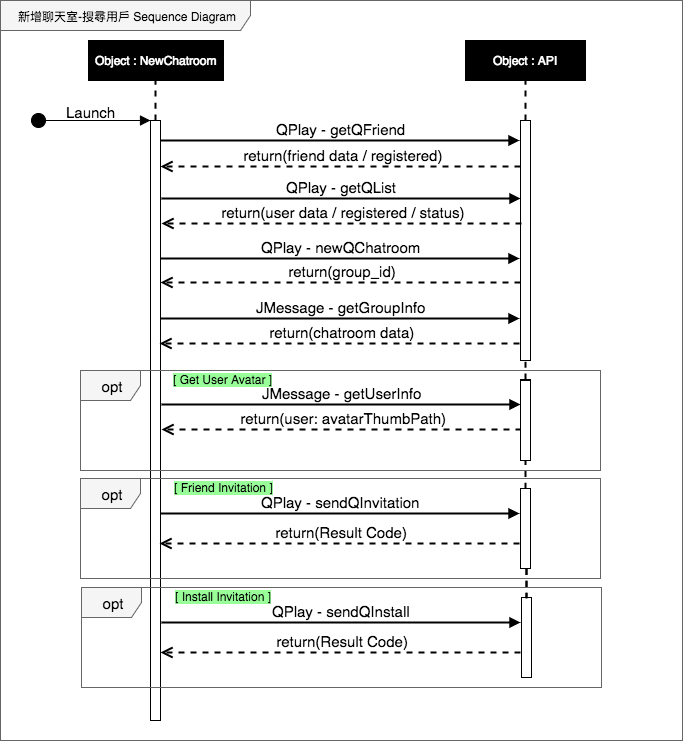
1. [ JMessage API - init ]
   1. 所有 JMessage 的運作，必須先執行此 API，由於此 API 沒有 callback，無法得知是否已經完成，另外，如果 init () 執行後，馬上執行 login ()，有機率會失敗，可能是 init () 尚未完成。
   2. 承上，所以，init () 執行後，延遲2秒後才繼續下一個動作。
2. [ JMessage API - login ]
   1. 確定 success 後才能開始使用 JMessage。
3. [ JMessage API - getConversations ]
   1. 取得聊天室列表資料，將資料儲存在 APP Local 端。
   2. 下次 call API 之後，在新增 / 更新 APP Local 端的資料。
4. [ JMessage API - getGroupMember ]
   1. [ 1 to 1 Chatroom ] 在一對一的聊天室執行。
   2. 目的是為了取得對方的大頭照縮圖路徑，儲存在 APP Local 端。
5. [ JMessage API - downloadOriginalUserAvatar ]
   1. 下載對方的大頭照的原始圖檔。
   2. 已經下載過大頭照原圖的聊天室，將圖片的生命週期期限記錄在 APP Local 端，生命週期暫定為3天。

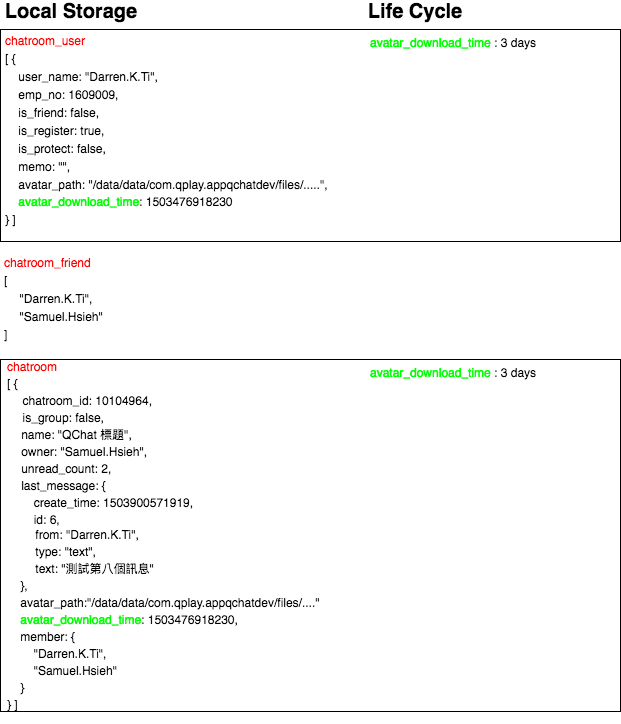


1. [ QPlay API - getQFriend ]
   1. 取得朋友資料，顯示 Name / Memo。
2. [ QPlay API - getQUserDetail ]
   1. 取得 User 自己的 Memo。
3. [ JMessage API - downloadOriginalUserAvatar ]
   1. 檢查大頭照的生命週期是否過期，過期的話進行更新。

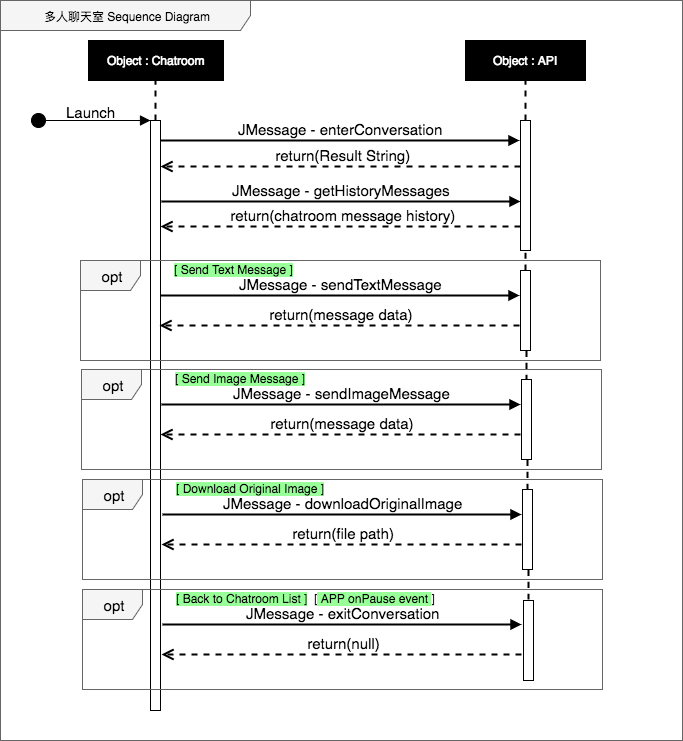


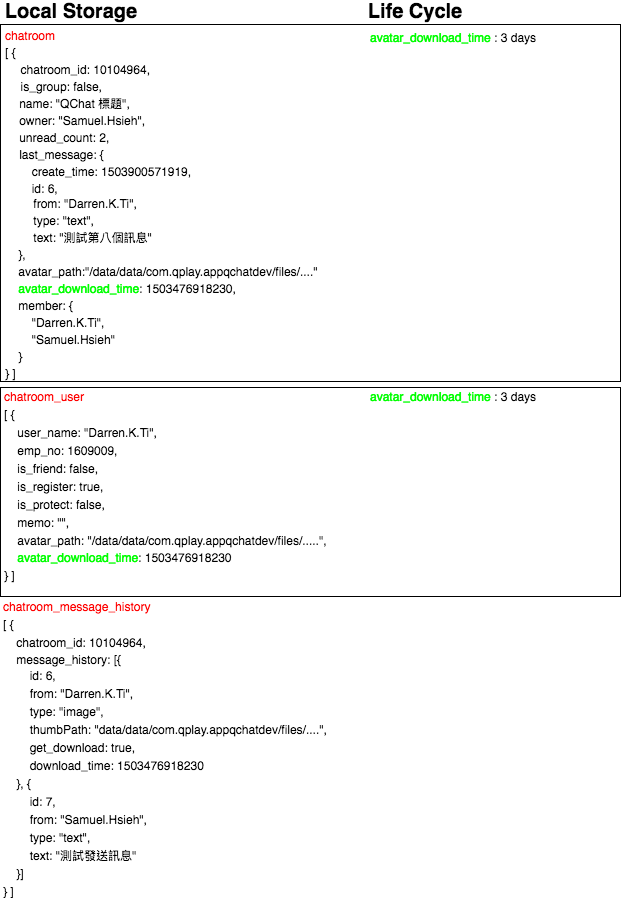
1. [ QPlay API - getQList ]
   1. [ Recommend Friends List ] 帶入 User 的 emp\_no，QPlay 回傳同公司同部門的推薦名單。
   2. 一般使用，依照 User 輸入的姓名，QPlay 搜尋後回傳資料。
2. [ QPlay API - sendQInviation ]
   1. 針對指定 User 送出加入好友邀請。
3. [ QPlay API - sendQInstall ]
   1. 針對指定 User 送出安裝邀請。



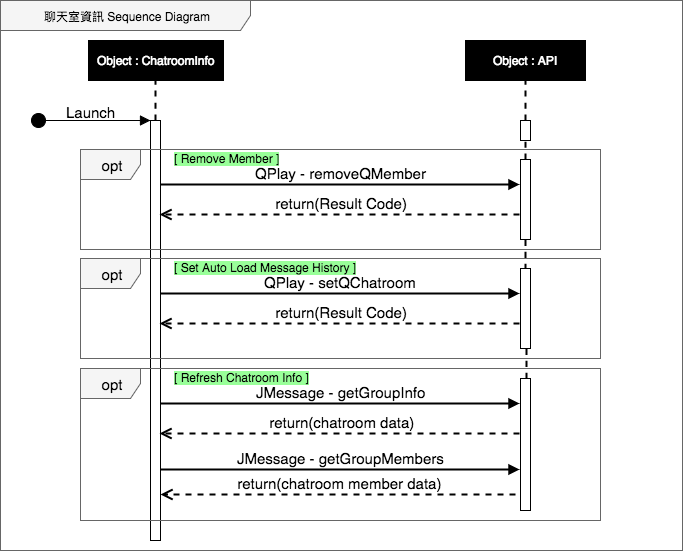


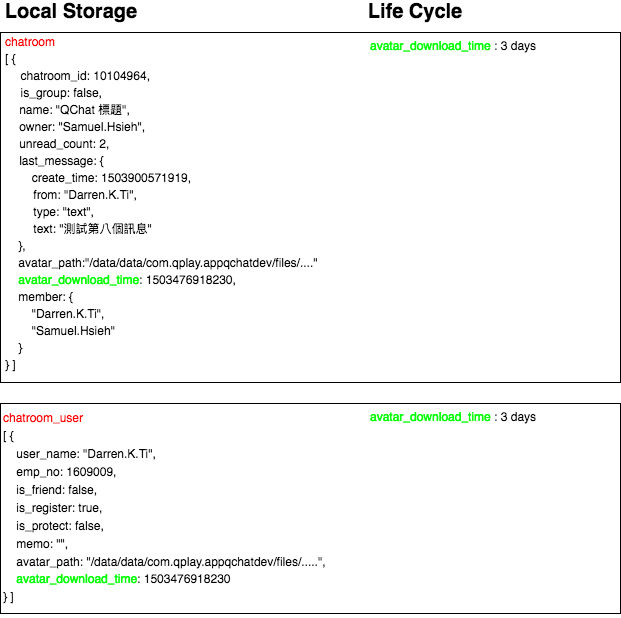
1. [ QPlay API - getQFriend ]
   1. 取得朋友資料儲存在 APP Local 端，此頁面只執行一次。
   2. User 後續搜尋時以 APP Local 端的資料做搜尋，不再透過 QPlay 搜尋。
2. [ QPlay API - getQList]
   1. 依照 User 輸入的姓名，QPlay 搜尋後回傳資料。
   2. 必須過濾掉跟 Friend List 重複的資料。
3. [ JMessage API - getUserInfo ]
   1. 如果 getQList 回傳的員工資料，不存在 chatroom\_user，表示此員工尚未有大頭照縮圖儲存在 APP Local 端，必須透過此 API 取得此員工的大頭照縮圖。
4. [ QPlay API - sendQInviation ]
   1. 針對指定 User 送出加入好友邀請。
5. [ QPlay API - sendQInstall ]
   1. 針對指定 User 送出安裝邀請。
6. [ QPlay API - newQChatroom]
   1. 針對所選擇的成員，透過 QPlay 建立聊天室，成功後，QPlay 會回傳此聊天室的 group\_id。
7. [ JMessage API - getGroupInfo ]
   1. 承上，使用 QPlay 回傳的 group\_id，透過此 API 檢查確定聊天室已經存在。



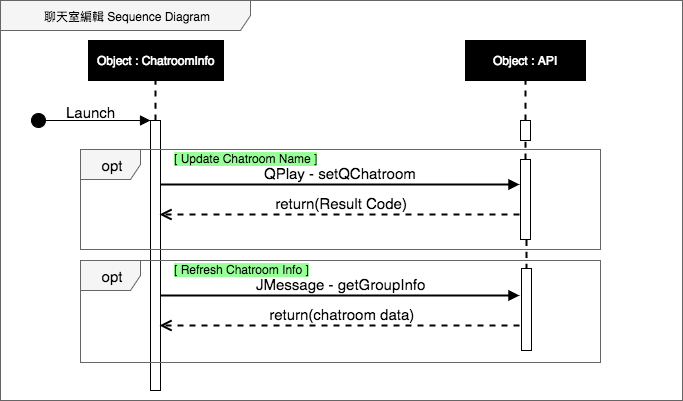


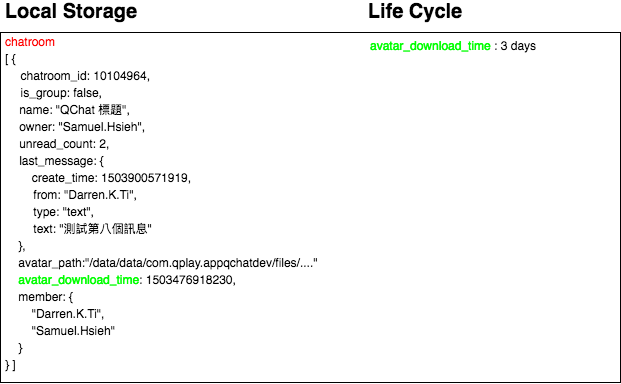
1. [ JMessage API - enterConversation ]
   1. 當 User 一進入此聊天室，執行此 API，User 就不會收到此聊天室的推播。
2. [ JMessage API - getHistoryMessages ]
   1. 針對此 User，依據 chatroom > last\_message > id 以及 chatroom > unread\_count 這兩個資料，取得在這個聊天室的歷史訊息。
3. [ JMessage API - sendTextMessage ]
   1. 發送文字訊息。
4. [ JMessage API - sendImageMessage ]
   1. 發送圖片訊息。
5. [ JMessafe API - downloadOriginalImage ]
   1. 當 User 點擊圖片訊息，透過 chatroom\_message\_history > get\_download 檢查是否已經下載過原始檔案。
   2. 承上，如果尚未下載過，則下載此圖片的原始檔案並且儲存至 APP Local 端。
   3. 承上，已經下載過原始檔案的圖片訊息，將 download\_time 記錄至 chatroom\_message\_history。
6. [ JMessage API - exitConversation ]
   1. 執行此 API，原先停止收到此聊天室的推播，會恢復為可以收到推播。
   2. 在下列兩種狀況下會執行:
      1. 離開聊天室，回到聊天室列表。
      2. 當手機退到背景執行時。





1. [ QPlay API - removeQMember ]
   1. 移除聊天室成員。
   2. 只有聊天室 Owner 可以執行。
2. [ QPlay API - setQChatroom ]
   1. 設定 (自動匯入歷史紀錄)，使用 JMessage 的資料欄位 desc 來做記錄。
   2. 只有聊天室 Owner 可以執行。
3. [ JMessage API - getGroupInfo ]
   1. 聊天室資料異動時，更新資料。
4. [ JMessage API - getGroupMembers ]
   1. 聊天室成員自動時，更新資料。





1. [ QPlay API - setQChatroom ]
   1. 更新聊天室名稱。
2. [ JMessage API - getGroupInfo ]
   1. 聊天室資料異動時，更新資料。

## 資料表結構 (Table Layout)

# Implementation

* Index.js
  + Define pageList
  + Define appKey, appSecretKey
  + Define window.initialSuccess()
  + Define onBackKeyDown() for android platform
  + Define onPause() / onResume() when APP work in background / foreground
* viewIndex.html
* viewIndex.js
* viewChatroom.html
* viewChatroom.js
* viewNewChatroom.html
* viewNewChatroom.js
* viewChatroomInfo.html
* viewChatroomInfo.js
* viewAddMember.html
* viewAddMember.js
* viewAddFriend.html
* viewAddFriend.js
* viewMyInfo.html
* viewMyInfo.js

## 公用元件 (Public Component)

## 共用界面 (Public Interface)

* App.css
* App.html
* App.js