

資工系實務專題研究計畫成果報告

實作一個以開放式即時通訊協定為基礎的群體服務系統

專題編號：100-CSIE -S008

執行期限：99 年 1 學期至 100 年 1 學期

指導教授：劉傳銘 副教授

專題計劃參與人員：林宜輝

一、中文摘要

在現今社會，智慧型手機已經成為人人必備且不可或缺的隨身配件，而大眾上網習慣也從原本 PC 上網轉變成現在的手機上網。人們可以使用手機看影片、聊天、發訊息，而在大部份的使用狀況下，社交溝通還是為多數使用者使用手機上網最重要的主因。人類社交行為不外乎分享資訊，以及發起活動等行為，進而達到溝通與加強彼此之間連結的關係，所以本計畫將針對溝通與於不同社群平台間可以互相分享資訊的目的，開發了 SPACT(SuPer ACTivity)群體服務系統，藉此提供更簡便的方式讓群體做交流。

關鍵詞：Android、即時通訊、XMPP

二、緣由與目的

網路社群在近幾年快速崛起，改變過去傳統人們社交的方式，在現今網路蓬勃發展的社會，透過社群所提供的平台，可快速的與人們進行群體交流，雖然社群網站已滿足大部分使用者的溝通要求，但是，卻無法提供在不同社群平台間，人與人的距離，在本計畫中，預期能夠提供使用者以更簡單有效的方式於不同平台間完成群體性的活動。在設計上，將針對呼朋引伴(揪團)功能以及投票功能，以拉近不同平台，朋友與朋友間的關係。

三、相關背景與技術

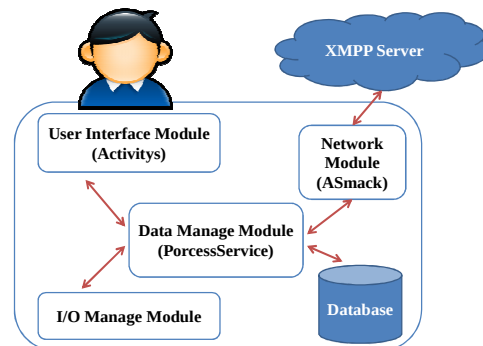
目前手機開發主流約可區分為兩類，一類是以 Open Source 為基底的"Android"開發環境[1]，以及 Apple 公司所提供的"iOS"作業系統。本專題所要設計的即時通訊程式 SPACT(SuPer ACTivity)，以開發便利性做為考量，最後決定採用 Android v2.3 做為開發平台。XMPP(Extensible

Messaging and Presence Protocol)[2]為目前所流行的即時通訊協定，它是由 XML 架構所組成，透過 XML 包裝資訊(Ex:OpenID)以達到平台間溝通傳遞目的，進而讓雙方能夠進行交流。現階段各大社群平台所提供的即時應用程式如 Apple 的 iChat、Google 的 Gtalk[5]、Facebook Chat[4]皆採用 XMPP 做為通訊手段，在本次計畫中將對該上述即時應用程式進行整合，故使用 XMPP 做為即時通訊協定之規範。

本專題的開發工具為 Eclipse IDE for Java Developers，並以 HTC Magic 做為預設的模擬工具。

四、開發過程與方法

本系統設計上共分為四個模組區塊如[圖一]，分別為(1)數據處理模組(Data Manage Module, DMM)：負責訊息與事件處理的主要模組；(2)網路模組(Network Module, NM)：負責建立行動同儕網路連線的部份、接收來自系統網路的訊息並將之送往 DMM 處理；(3)裝置輸出入管理模組(I/O Manage Module, IOMM)：接收外部裝置的訊息，處理其事件後將訊息送往 DMM；(4)使用者功能介面層(User Interface Module, UIM)：接收並過濾使用者事件後，將事件送往 DMM 處理並呈現執行結果予使用者。



[圖一]:系統架構圖

本計畫提供使用者群體服務著重於提供跨平台之群體聊天室，其中在群體通訊部分，雖然在 XMPP 與 ASmack 函式庫[3]中以提供有類似方式，但卻無提供跨平台連線功能，為了達到不同伺服器所屬的帳號都可以使用，所以在群體通訊的部分將利用 XMPP 最基本的功能「一對一傳輸」，並加上樹狀階層結構的方式，即可實踐群體聊天室。

五、系統成果展現

本計畫所完成的即時通訊程式 SPACT 提供以下幾種服務：

好友訊息、地點訊息：如同簡訊的方式，選擇收件者後編寫文字；地點訊息則是在編寫文字後選擇要傳送的範圍，系統會自動將訊息傳送給在此範圍內的在線好友。

聊天室：提供兩種開啟聊天室的方式，除傳統的列表式清單外，提供以地點為基礎的朋友地圖[圖二]。

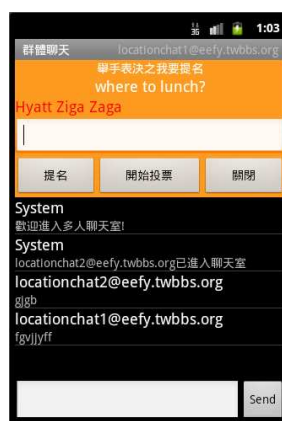
呼朋引伴、舉手投票：在兩個群體互動中，最常使用的功能為呼朋引伴功能。該功能相較於現行社群平台所提供的一般揪團功能服務，增加時間區塊功能，以動態的視覺方式讓彼此好友們藉由觀看，而安排的時間[圖三]；舉手投票功能則是將學校推舉幹部的運作模式套用於聊天室中，由提議人發起主題後，再讓其他人參與該主題，並且可以發起回應[圖四]，再配合系統進行投票過程，以動態並且記名的方式進行，讓使用者知道好友投哪個選項[圖五]。



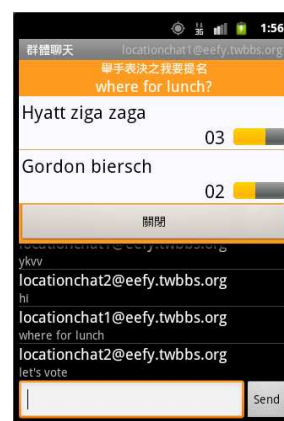
[圖二]:
好友地圖



[圖三]:
團體時間約定



[圖四]:
投票—提名



[圖五]:
投票—即時結果

六、專題貢獻與結論

為達到跨伺服器的目的，本專題利用 XMPP「一對一傳輸」之 API，將群體通訊建立在點與點之間。然而 XMPP 在雙方未建立認證前，無法直接建立連線。有鑑於此，本專題考慮以第三方轉送的方式，利用樹狀階層結構，達到資訊傳送之目的。

本計畫執行時，在軟體測試上遭遇相當大的問題，為了模擬出群體實際的使用情況，必須同時操作多台模擬器，並且記錄操作的過程，以確保所有情況皆通過測試，此部分仍有改進的空間；撰寫使用者介面部分，由於經驗不足，導致在實際執行上，無法實現原先的設計，使軟體的品質下滑，此部分有待改進。

七、參考文獻

- [1] Android SDK | Android Developer <http://developer.android.com/sdk/index> (Viewed on Mar, 2011)
- [2] XMPP(Extensible Messaging and Presence Protocol), http://en.wikipedia.org/wiki/Extensible_Messaging_and_Presence_Protocol (Viewed on Oct, 2011)
- [3] ASmack, <http://code.google.com/p/asmack/> (Viewed on Oct, 2011)
- [4] Facebook Chat API <http://developers.facebook.com/docs/chat/> (Viewed on Oct, 2011)
- [5] Google Talk for Developers <http://code.google.com/apis/talk/> (Viewed on Oct, 2011)