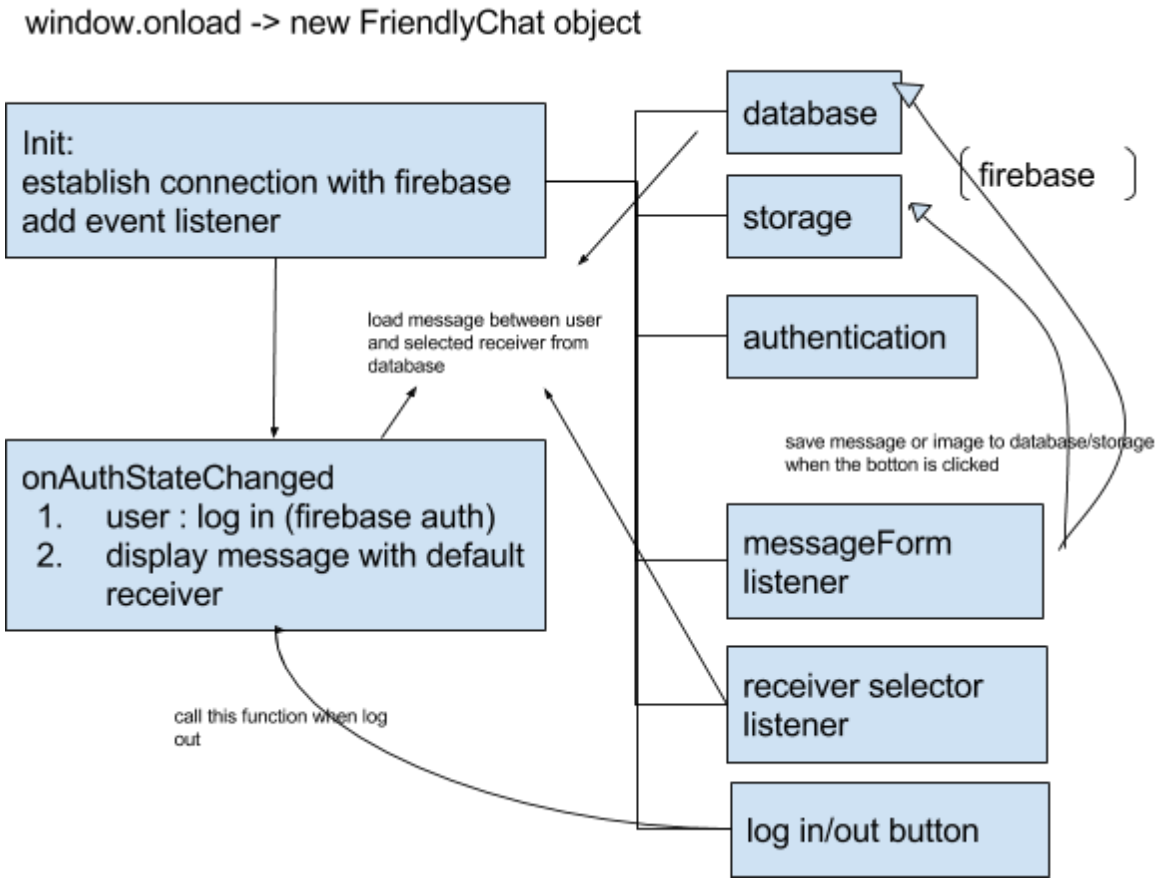
Lab1 Report

電機四 b02502094 蔡秉璇
電機四 b02901115 陳昱安

1. 功能

- 選擇對話對象
- 傳送文字/圖片
- 利用 google 帳號登入

2. 架構



對話系統架構大致如上圖。

在開啟瀏覽器視窗時，會創造一個新的 FriendlyChat 的 function 物件，而這個物件內部的 function 能夠使得對話系統正常運作。

我們的後台是使用 Firebase 建的。好處是能夠更簡潔的管理，且有 real-time database，不需要一直寫一堆 sync 的東西。而且可以 deploy 至 Firebase hosting。

3. 功能及 Functions

A. 選擇對話對象

我們在 html 裡增加了<select></select>，目的是為了讓使用者能夠選擇要跟誰對話，同時視窗內也只會顯示兩人之間的對話，而和這項功能有關的 function 及 component 有：

- messageReceiver
  為 html 裡的 select element，我們加了一個 listener 再上面，若是使用者變動了目前所選的值，就會呼叫 refreshMessages 刷新頁面。
- FriendlyChat.prototype.refreshMessages
  這個 function 會 remove 掉所有現在顯示的 message 並且重新呼叫 loadMessages 來 load database 中的 messages。
- FriendlyChat.prototype.loadMessages
  這裡我們會根據 select 現在的 value 以及目前的 user，來從 database 選出對應的 sender, receiver 相關的 messages，並且呼叫 displayMessages 把他顯示於視窗，但目前 selector 內的 option 還是我們預先設定好，我們有嘗試讓他能夠動態根據目前的 user lists 來 update 可以選擇的 receiver，但最終還未成功。

B. 登入登出

- FriendlyChat.prototype.signIn
  Firebase authentication 的 API 有整合使用 Facebook, Google…etc. 登入的功能，這部分我們是用 google authentication provider。然後利用 Firebase authentication API 裡的 signInPopup 跳出登入的視窗。
- FriendlyChat.prototype.signOut
  使用 Firebase authentication 的 API 的 signOut function。

C. 傳送文字/圖片

在登入之後可以給對應的接受者傳送文字/圖片。

- FriendlyChat.prototype.checkSignedInWithMessage
  這個 function 會根據後台 authentication 提供的 API 去判斷是否有 current user，如果沒有 user 的話網頁會跳出 “You must sign in first” 的文字。
- FriendlyChat.prototype.saveMessage

在確定為登入狀態及在 message input 有文字輸入的情況下，會送一個 .json，內容包括 sender, receiver, message content 及 sender 的 profile image url，至後台的 database。

- FriendlyChat.prototype.saveImageMessage
  在 user 選完 image 且 submit 之後，會先 call preventDefault()，並將 target 裡的 file 拿出來，然後把 document 中的 image-form 做 reset。在確定 target file 是 image.*之後，會 push 一個 json 至後台的 database，內容包括 sender, receiver, imageUrl 及 sender 的 profile image url，且 imageUrl 會是一個表示在 loading 的.gif。
  因為 Firebase database 的 API 中，push function return 的是剛 pushed 的 .json 的 reference + promise 所形成的物件，所以在 push 之後可以 access 剛剛這則 message 的資訊。得到這則 message 的資訊之後，先將剛剛 user 選擇的 image file 上傳到 Firebase 的 cloud storage，並將 return 回來的 image path 拿來 update 剛剛這則 message 裡的 imageUrl。
- FriendlyChat.ResetMaterialTextfield(element)
  在送完訊息之後會清空 element 裡的 text。