

Scenarios	Steps	Web Pages	Functions	Notes	Related API
1.在後台 schedule 一個會議	1.使用者登入 twhoa.anz.com 網站 (網址名稱暫定)。	admin/login.jsp	登入功能		
	2.登入完成後，列出所有已經發生的影片，即將發生的影片清單，及特定一個 Google Calendar (參考 Google Developer Live 放的方式)，包括未來一周及過去的影片。	admin/home.jsp	(1) 列出會議清單 (2) 建立會議 (3) 編輯會議 (4) 複製會議連結 (5) 顯示報表 (6) 顯示問題清單		Calendar API
	3.建立一個會議，註明主題、日期、時間、法律條款、是否直播、錄影網址(如果非直播)、直播網址(如果直播)。	admin/updateSeminar.jsp	建立/編輯會議功能	(1) 剛建立時還沒有 YouTube Live Id，此 Id 需等直播開始之後來回填。 (2) 可選擇是否開啓來賓查看問題清單功能。 (3) 建立完成後，透過 Calendar API 新增一筆，並記錄 Id。 (4) 修改完成後，根據 Id 透過 Calendar API 修改資料。	Google OAuth2 Calendar API
	4.產生一個對外網址，連結該網址則以後在直播時可直接看到畫面。例如：http://twhoa.anz.com/view.jsp?id=1234567	view.jsp	(1) 直播顯示功能 (2) 錄影檔案顯示功能	需確認： (1) 還沒開始前要播放什麼畫面 (2) 開始時是否可以自動切換，還是需要不斷 reload	Hangouts On Air API
	5.產生一個加入 Outlook 連結，使用該連結即可加入 outlook。				
	6.Update YouTube video privacy			YouTube	
2.開啟一場 Hangouts On Air 會議	1.使用者登入 plus.google.com，進入 Hangouts 後按下「發起視訊聚會」。		Google+ 標準功能		
	2.新增聚會名稱。		Google+ 標準功能		
	3.點擊「啟用 Hangouts 直播功能」。		Google+ 標準功能		
	4.按下開始，並確定網路等是否正常。		Google+ 標準功能		
	5.登入 twhoa.anz.com 後台，將影片 YouTube Live ID 複製到會議資訊內。	admin/home.jsp admin/seminar.jsp	複製 YouTube Live Id 到「直播網址」		
	6.點擊會議之後的問題按鈕，查看客戶提出的問題。	admin/questions.jsp	列出問題，含清單	需定時 refresh 或透過 ajax 技術 refresh。	
3.使用者觀看即時影片	1.登入前台頁面	login.jsp	前台登入	是否需結合既有會員系統	
	2.顯示目前直播影片清單及過去清單	home.jsp	直播中影片列表 歷史影片列表		Calendar API

	3.點擊影片連結後顯示影片	view.jsp	1.計數器+1	1.Server side紀錄 who、when、what 2.利用 JavaScript 偵測若離開瀏覽器，將最後觀看時間及長度記錄下來 3.時間 (關閉瀏覽器的時間)、長度 (用 JavaScript 取 max)	YouTube Player	
	4.直播中影片可提出問題	view.jsp	1.問問題	1.留下 form 可提出問題 2.不可影響觀看流程		
4.使用者透過 Email 內連結上網登錄之後看直播或影片	1.透過情境一所產生之網址，例如 http://twhoa.anz.com/view.jsp?id=1234567，點擊後會先確認是否有登入，若尚未登入則先顯示登入畫面	login.jsp				
	2.顯示影片	view.jsp	1.先判斷若非直播，則直接 embedded顯示影片網址 2.若為直播，則 embedded直播網址 3.若為直播，需有一個程式自動偵測時間，直播開始時 refresh頁面		YouTube Player	
5.會員管理	reuse 既有會員資料庫	N/A				
6.報表	1.登入後台，查看影片列表	admin/home.jsp				
	2.點擊影片資料的 Report 按鈕。	admin/seminar.jsp		純文字即可		