

Zenbo：平面巡邏機器人

指導老師：曾士桓 教授

參賽組員：陳柏家、許志宇

摘要

此報告將介紹我們對 Zenbo 開發新系統的架構，首先講述整個系統的簡介、背景、動機、目的，接下來以客觀的角度分析使用者的功能需求，每項需求皆有對應的功能，再來詳細地描述整個系統的功能，並架構出完整的功能架構圖，當分析完功能後，進行整個系統的架構進行描述，我們也將初步架構出 Zenbo 最後預期開發完成的系統架構圖，最後我們會以統一塑模語（英語：Unified Modeling Language，縮寫 UML）內的三個主要的模型 1. 功能模型：從用戶的角度展示系統的功能，包括使用個案圖 2. 物件模型：採用物件，屬性，操作，關聯等概念展示系統的結構和基礎，比如類別圖 3. 動態模型：展現系統的內為。包括序列圖，活動圖，狀態圖、合作圖對新開發的系統進行各種不同角度的分析。

1. 背景

能夠促使我們可以開始進行此系統開發的背景主要是因為，機器人製作的技術進步，小時候的機器人可能只是玩具，甚至是卡通、或是科幻電影、電視劇等等會出現的東西，如今機器人結合許多的感應器，比如：聲音感測器、紅外線感測器(避障用)，還有許多在此就不在列舉，現今的機器人結合許多的感應器令我們能夠設計更多元的功能應用在機器人上。

2. 動機

目雖然製作機器人的技術進步，但若是要客製化打造一個顧客特定需求的機器人，所花費的

造價就會非常昂貴。加上近年來捷運站上影響公共安全的事件頻傳，比如：持刀揮舞、尾隨、偷拍色狼。以上兩點促成我們想要去打造這個系統。

3. 構想

1. 客制化機器人造價昂貴：

我們的想法一來是 Zenbo 造價不貴，二是我們所使用的皆是 Zenbo 內建應用程式或是一些網路上可取得的資源進行開發，如此一來便可以大幅度改善客製化機器人花費昂貴的題。

2. 捷運站上影響公共安全的事件頻傳：

打算建置一個無死角的巡邏系統，加上能夠感應聲音立即回報主控台，達到無死角立即回報的效果以解決上述問題。

4. 目的

主要在製作一個由兩個子系統 1. 建地圖系統 2. 巡邏系統構成的巡邏機器人，建立地圖 Zenbo 內建的應用程式下去做建立，這個系統主要是由整個系統開發完成後予使用者自行去建立編輯地圖，巡邏的整個子系統構成則由三個部分組成 1. 巡邏 2. 聲音辨識 3. 避障 4. 呼叫緊急聯絡人。巡邏系統則由開發者完成，整個系統的流程首先會由使用者建立地圖與聯絡人資訊，接著就可以開始巡邏了，遇到障礙物時會自動閃躲，當機器人聽到特定的字句時，便會呼叫已建立好的緊急聯絡人，在我們所使用的情境捷運站月台，此處使用者與緊急聯絡人將是監控中心人員，呼叫完畢人員前往現場處理，機器人重新啟動巡邏。

表 1. 《Zenbo：平面巡邏機器人》使用者需求分析表

使用者需求	對應功能
發生事故時，求救要有反應	聲音感測功能:根據使用者發出的字句來做偵測
	連絡相關人士:藉由聯絡系統播出通話請求救援
是否可以進行避障(ex:閃躲人或障礙物)	避障系統:以紅外線感測器進行障礙物感測
可設定巡邏範圍跟地圖	建地圖與巡邏:利用建好地圖進行巡邏

表 2. 《Zenbo：平面巡邏機器人》功能項目說明表

功能項目	功能項目操作	說明	
巡邏功能	聲音辨識	辨識民眾所說的語句:救命 搶劫 有變態	
	路徑規劃	為 zenbo 規劃路徑走向 ex:1. 隨機亂走、2. 有軌跡的走	
	避障	遇到障礙物或者民眾時可以自動避障	
	聯絡系統	藉由輸入到聯絡人的資料播打給聯絡人通知危害發生	
功能項目	功能項目操作	功能細項	說明
使用者介面	地圖介面	建立地圖	建立地圖才可以進行路徑規劃
		編輯地圖	設置地圖上的有障礙物的區域
		標記地圖	分別標記地圖各範圍的名稱
	聯絡人介面	新增聯絡人	供 zenbo 遇到危急情況時呼叫
		刪除聯絡人	更換使用者時會使用這個功能
	啟用巡邏功能	啟用	開啟巡邏保全功能
		關閉	關閉巡邏保全功能

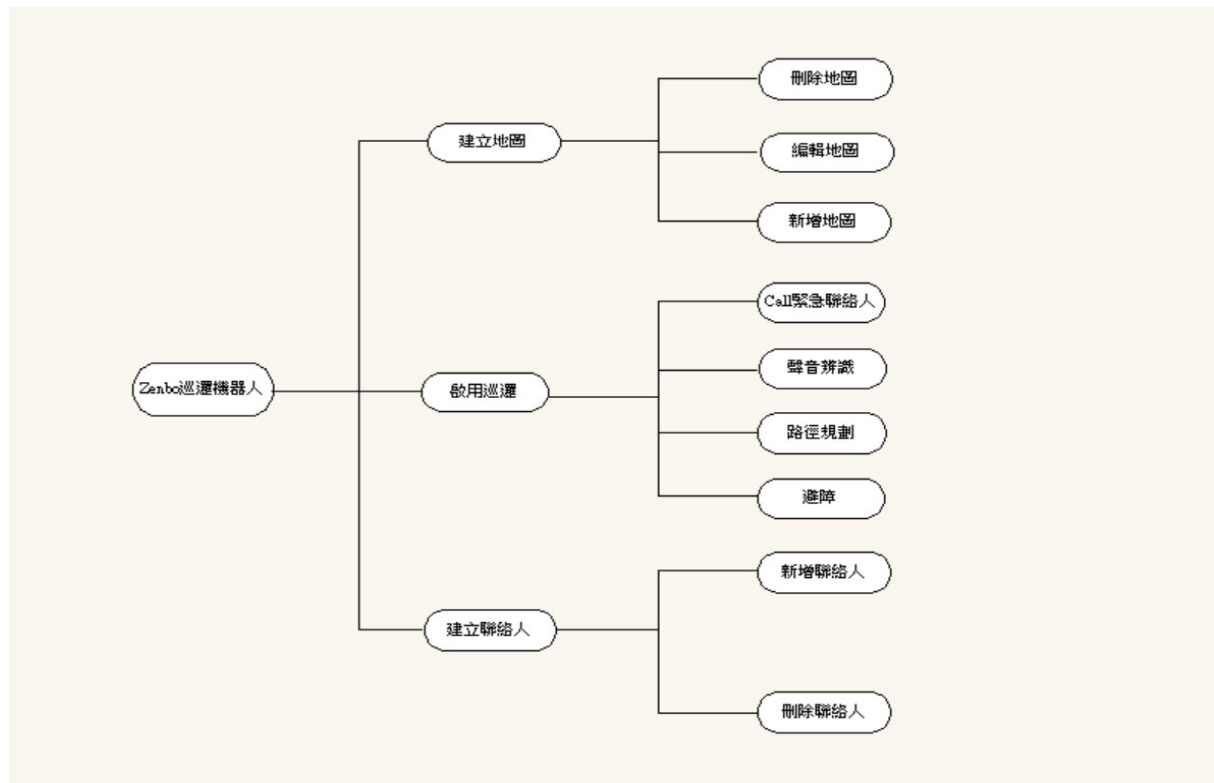


圖 1. 《Zenbo：平面巡邏機器人》功能架構圖

5.系統架構設計

我們在設計此專題主要以兩個子系統為主，如圖 2.所示：

- 1.建置地圖系統：使用者會透過 Zenbo 本身的 app 將巡邏的地圖系統建置進 zenbo 當中的資料庫做存取，並藉由 app 去啟用巡邏功能，當巡邏功能被啟用後，Zenbo 會啟動巡邏系統裡的三樣對應系統
- 2.巡邏系統：我們寫入三個子系統包在巡邏系統內，當巡邏啟用後 Zenbo 會啟用這三

樣系統來進行巡邏：

- 1.路徑規劃:Zenbo 會提取使用者存入地圖資料庫裡的資料去做路徑規劃巡邏
- 2.避障:在 Zenbo 之中裝配上推斷偵測去判斷前方是否有障礙物來做規避
- 3.聲音感測:藉由偵測聲音感測器接收進來的聲音到寫好的 sdk 中，去判斷聲音是否為設定好的對應字句，若為是的話會啟用 Robot Contact 去根據提取輸入再聯絡人資料庫裡的聯絡人播打號碼及時通知狀況發生

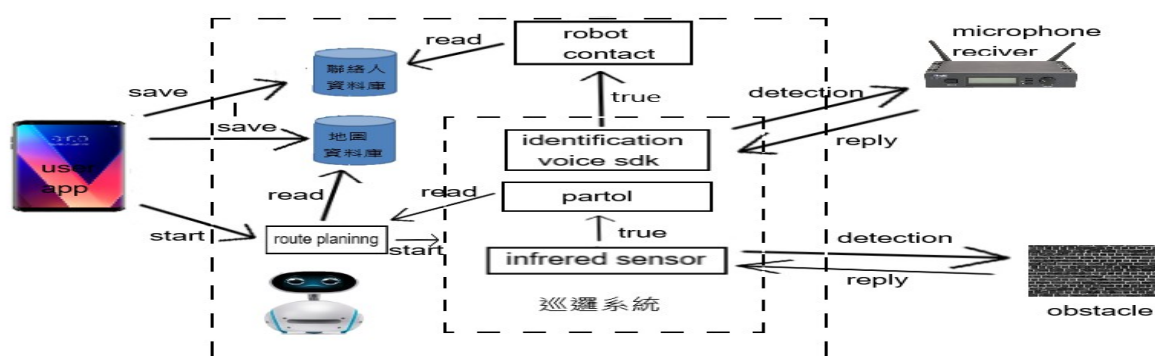


圖2.《Zenbo：平面巡邏機器人》系統架構圖

5.UML 使用者案例圖

使用案例圖（Use Case Diagram，或稱為使用個案圖）可以描述系統功能和其提供的服務，它是以使用者或客戶角度來看系統，強調系統能作什麼事，而不是如何作這些事。圖 3 為建立地圖系統的使用案例圖，建立地圖這個系統主要動作者（Primary Actor）為監控中心人員，支援動作者（Supporting Actor）則是地圖資料庫（MAPdatabase），使用案例分別有 1.Build MAP 2.edit MAP 3. markregion。

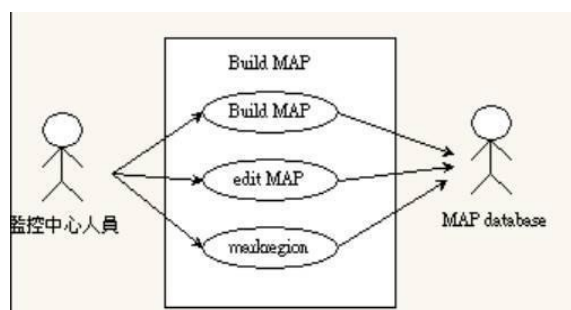


圖 3.《Zenbo：平面巡邏機器人》建地圖案例圖

圖 4 為巡邏系統，巡邏系統的主要動作（Primary Actor）為監控中心人員，支援動作（Supporting Actor）則分別有對話系統（DS）、地圖資料庫（MAPdatabase）使用案例分別有 1.停止巡邏（stop）2.開始巡邏（start），開始巡邏（start）包含以下幾個功能，聲音辨識（identification voice）、路徑規劃（route planning）、避障（obstacle avoidance）。

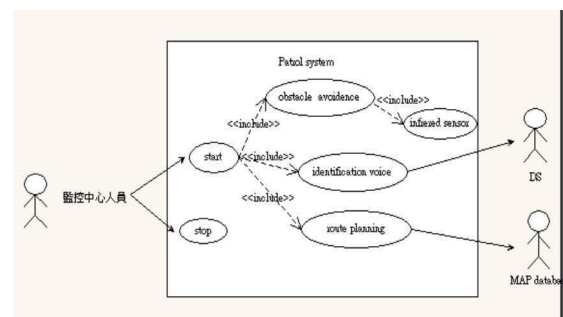


圖 4.《Zenbo：平面巡邏機器人》巡邏案例圖

6. UML 類別圖

類別圖中的類別節點可以把每一個類別所有的內容呈現出來，也可以只有顯示類別之間的關聯，這些關聯就是程式碼之間的相依性。使用類別圖型可以让你清楚的瞭解某一些特定的相依性，尤其在一個高度抽象化的軟體系統中，瞭解這個特性更加重要。zenbo 巡邏機器人的類別圖主要由 USER GUI 這個介(interface)為起始點往下做延伸，由於我們 zenbo 的整個系統的設定主要是由建置地圖與巡邏兩個子系統構成所以此介面需要實作 (implement) 建地圖(MAPbuild()) 與開始巡邏 (start patro()) 兩個方法，而整個系統還需要一個系統資料庫來儲存資料，於是也設置了系統資料庫的介面(interface) 供需要用到資料庫的類別進行實作 (implement)。建置

地圖需要一個資料庫來存放建置完成的地圖所以必須實作 (implement) 系統資料庫(Systemdatabase)作為地圖的資料庫，Mapbuild 類別將會(<<use>>)到地圖資料庫以便日後刪除新增與修改。另一個實作的子系統巡邏 (start patro())，將會囊括四個子(UML 類別圖組合關係)1.路徑規劃：需要使用(<<use>>)地圖資料庫的資料進行起點到終點的最短路徑規劃演算法 (short path algorithm)2.偵測聲音：回傳值為 boolean 判斷是否偵測到特定字句3.避障：回傳值仍為 boolean，使用紅外線感測器進行判斷4.呼叫使用者：此類別將會使用到使用者資料庫類別需要實作(implement) 系統資料庫來達成儲存使用者資料的目的，如此一來才能使用呼叫使用者這個類

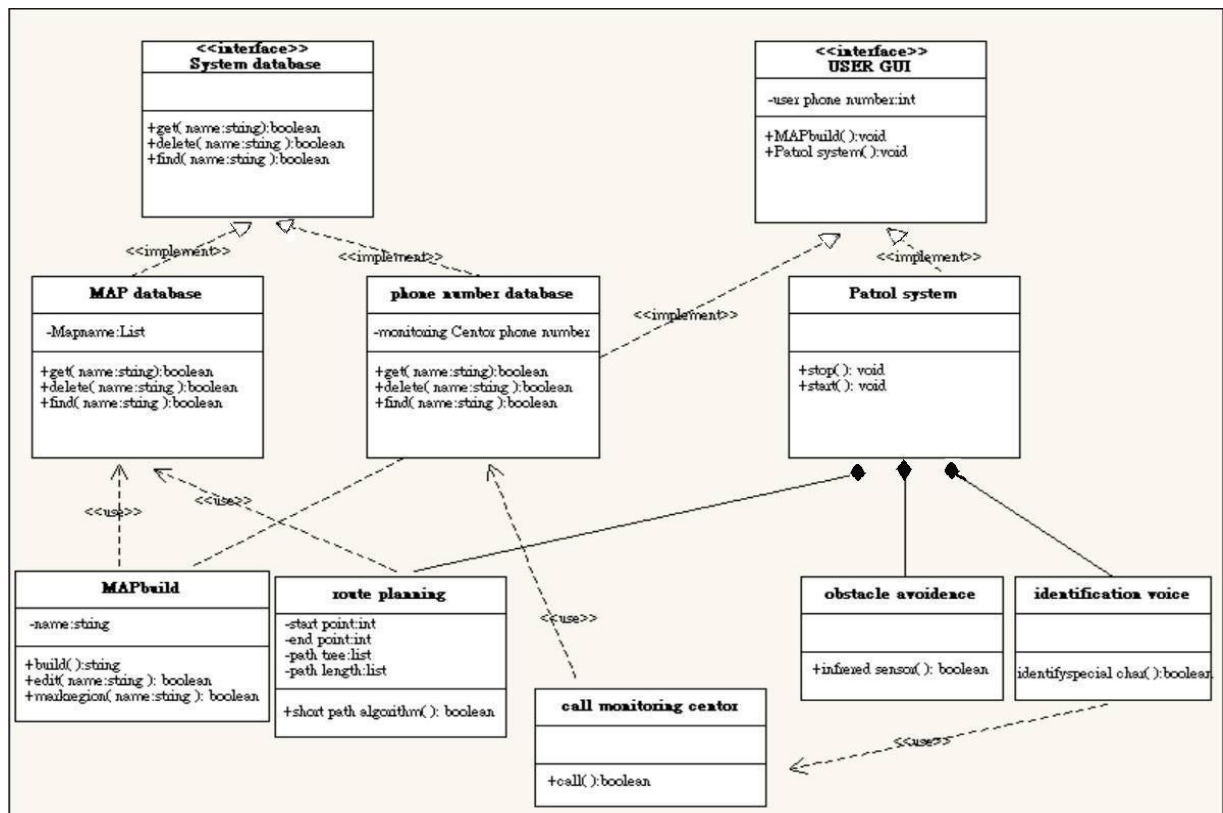


圖 5.《Zenbo：平面巡邏機器人》UML 類別圖

7. UML 循序圖

建立循序圖型主要的基礎在「時間順序」和「合作物件」的概念上，可以從圖型中清楚的瞭解有哪些物件在完成某一項任務，更重要的是圖型中可以顯示出完成這樣任務的先後順序。

圖 6 為建置地圖系統的 UML 循序圖，首先由使用者建立 (<<create>>) Build MAP 這個類別接下來 build MAP 類別由於建置地圖必須掃描環境所以接著呼叫 Scanning -environment 這個類別，掃描環境的當下需要同步的將資料存進 Mapd

abase 所以 Scanning-environment 這個類別開啟時也立即建立(<<create>>)Mapdatabase 類別當環境掃描完成時先結束 Scanning -environment 類別在結束 Scanning -environment 類別，Zenbo 建置地圖完畢後必須要在進行最後一次編輯地圖，才得已完成建置地圖的類別所以當環境掃描與建置完成時接著 Build MAP 類別會建立(<<create>>) editMAP 類別在開啟地圖資料庫類別將編輯過後的資料存入資料庫裡，待編輯完成後結束 BuildMAP 類別。

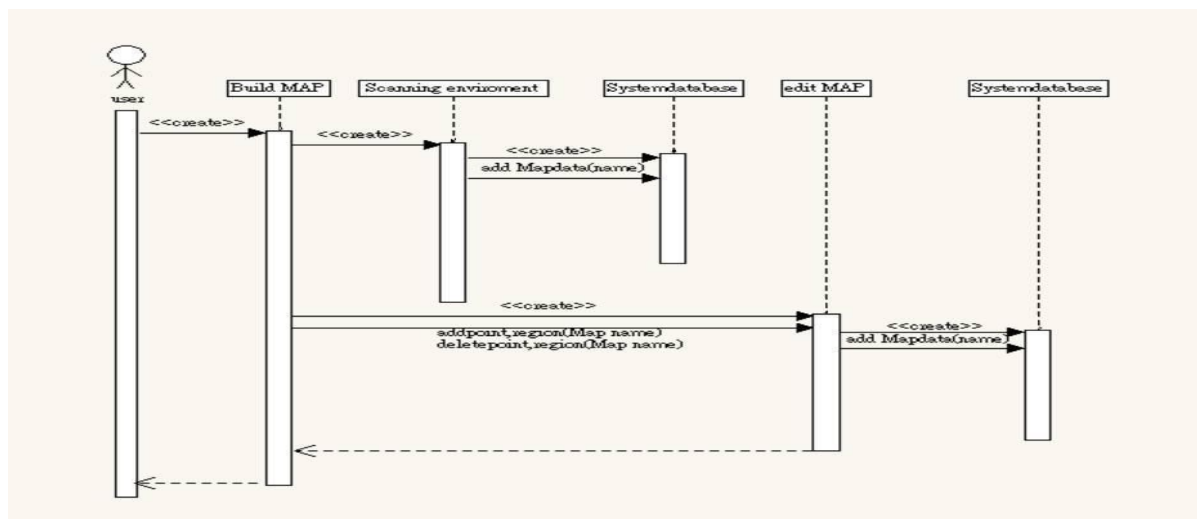


圖 6.《Zenbo：平面巡邏機器人》建地圖 UML 循序圖

圖 7 為巡邏系統的 UML 循序圖，使用者建立(<<create>>) patrol system 而巡邏需要依附地圖資料與路徑規劃，所以開啟巡邏系統後便接著先建立 (<<create>>) MAPdatabase 類別使 search 方法尋找地圖資料庫內我們將會使用到的地圖資料，待資料讀取完畢後回報巡邏系統，巡邏系統接著建立(<<create>>)路徑規劃類別，同樣等待路徑規劃完成後回報巡邏系統，而此時巡邏所需要的資料與方法皆具備後便開使真正地進行巡邏，而在我們系統的設計下巡邏必須搭配

1.辨識聲音 2.避障，於是在巡邏被建立後便同時建立以上所舉的兩種類別，而聲音辨識類別需要接著開啟對話系統因為對於語句的設定是儲存在對話系統(DS)，避障類別則需要搭配紅外線感測器，避障類別會接著開啟紅外線感測器在巡邏開啟後會一直持續運行到使用者直接關閉它，或是感測到聲音後結束避障與聲音感測系統接著開啟呼叫使用者類別後才會結束巡邏關閉整個巡邏系統。

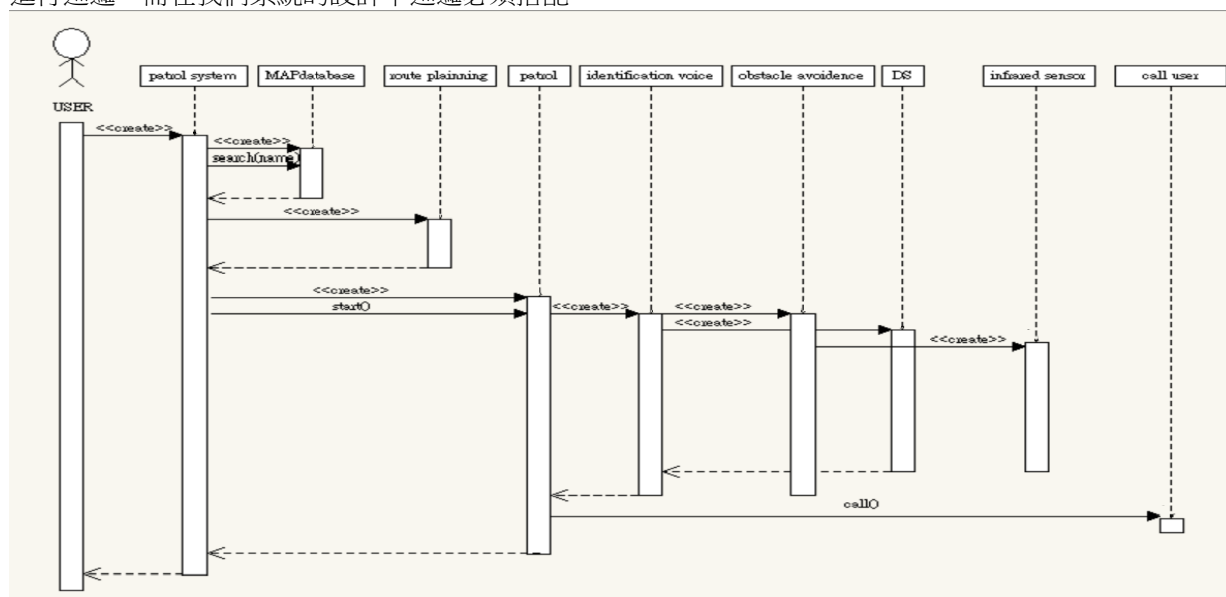


圖 7.《Zenbo：平面巡邏機器人》巡邏 UML 循序圖

8.UML 合作圖

合作圖型、Collaboration Diagrams 與循序圖型所顯示的資訊，有部份是重複的，都是用來呈現物件在時間上的互動。但是循序圖型以時間順序來顯示物件的互動，而合作圖型則把焦點又拉回物件，以物件來顯示互動的順序。當你想要瞭解軟體系統中的關鍵物件，與其它物件一起合作

完成一件任務的順序時，就要使用合作圖型。圖 8 是建置地圖時物件間的合作圖，BuildMAP 的執行過程主要是由 editMAP 與掃描環境合作完成會先執行掃描環境，掃描環境將開啟系統資料庫存放資料，執行結束後，開啟 editMAP 物件，而 editMAP 同樣是會開啟系統資料庫存放資料。

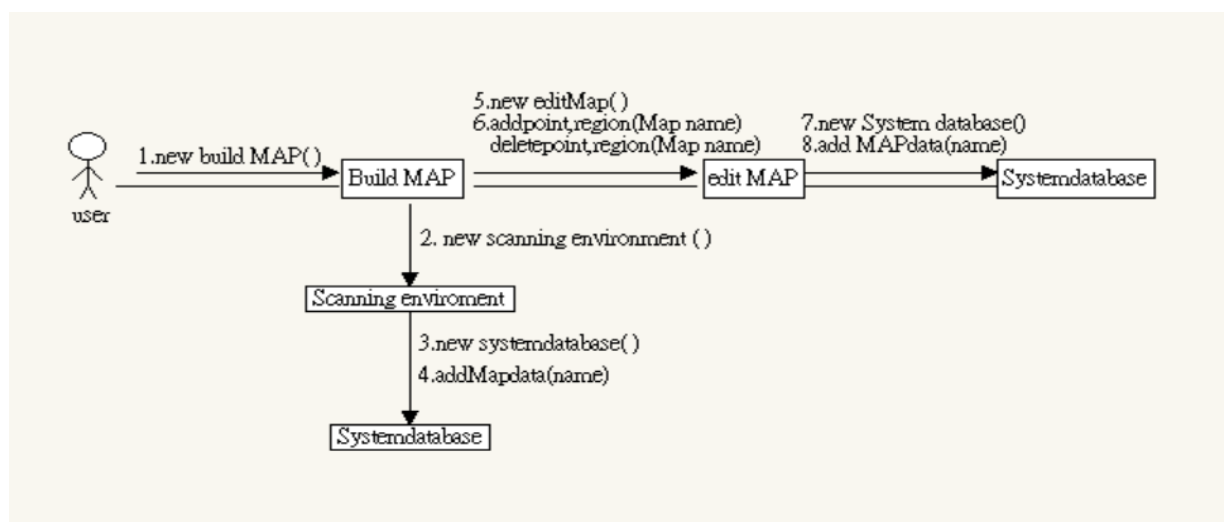


圖 8.《Zenbo：平面巡邏機器人》建地圖 UML 合作圖

圖 9 巡邏系統的合作圖，巡邏系統整個主系統依序由 1.Mapdatabase 2.route planning 3.patrol 三個物件合作完成。而 patrol 則由 1. identification voice 與 obstacle avoidance 兩物件為同時發生 2.ph one system 依序組成，

phone system 執行前 identification voice 會先開啟 DS 來感測語音，在接下來由 obstacle avoidance 開啟 infrared sensor 來避障最終在呼叫 phone system。

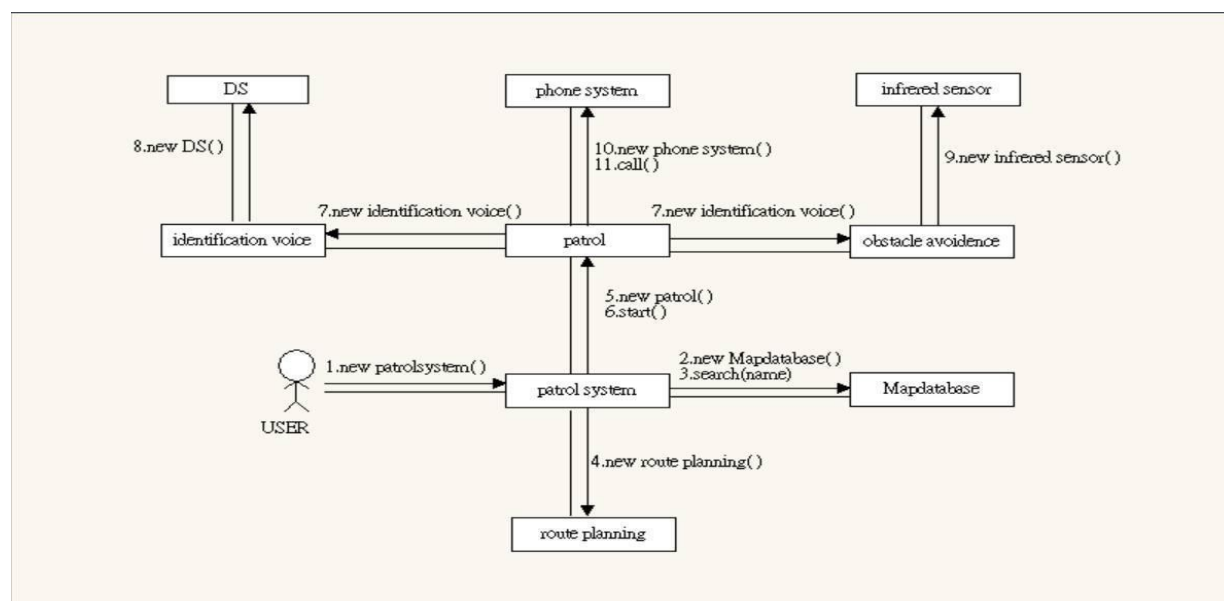


圖 9.《Zenbo：平面巡邏機器人》UML 巡邏系統合作圖

9.UML 活動圖

「活動圖型、Activity diagrams」用來顯示軟體系統中特定的活動情形，活動圖型與其它圖型有一個最大的差異，就是在顯示活動情形的時候，通常不會考慮與物件或類別相關的問題。所以活動圖型是用來呈現軟體系統中某一些特定的活動，使用一般的流程圖概念來繪製。

圖 10 為建置地圖的活動圖，首先建置地圖需要先掃描環境接著會開啟系統資料庫，假若掃描失敗則重回環境掃描步驟，成功後便接著開始編輯地圖，同樣地會先開啟系統資料庫，若是失敗就重新編輯地圖，成功的話就完成整個建置地圖的系統。

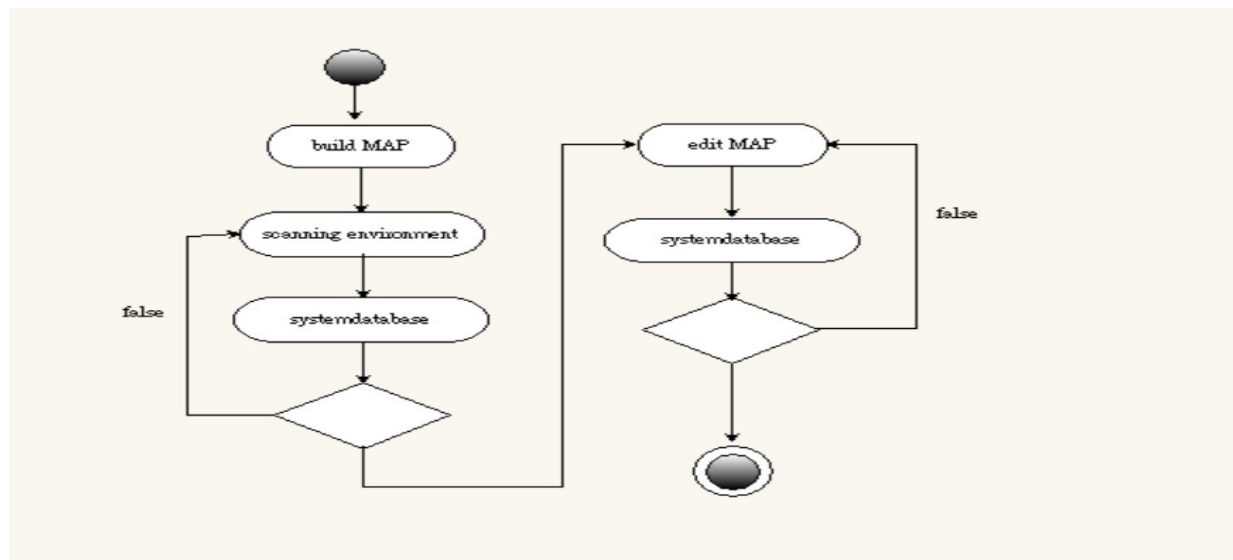


圖 10. 《Zenbo：平面巡邏機器人》UML 建置地圖活動圖

圖 13 為巡邏系統的活動圖首先會開啟巡邏的系統，接著開啟系統資料庫取出地圖資料進行路徑規劃若是路徑規劃失敗則再次開啟地圖資料庫重新規劃路徑，成功的話便開始巡邏而巡邏開始時會同時的開啟聲音辨識與避障系統，避障系統這端會開啟紅外線感應器，感應前方是否有

障礙物，為否時回到避障系統持續避障，為真時進行閃躲，閃躲結束後仍回到避障系統，直至另一端聲音辨識端偵測到特定聲音才會結束與聲音辨識端一起結束。聲音辨識這端會持續辨識特定語句直至辨識到開發者設定的語句時便呼叫使用者並終止整個巡邏系統。

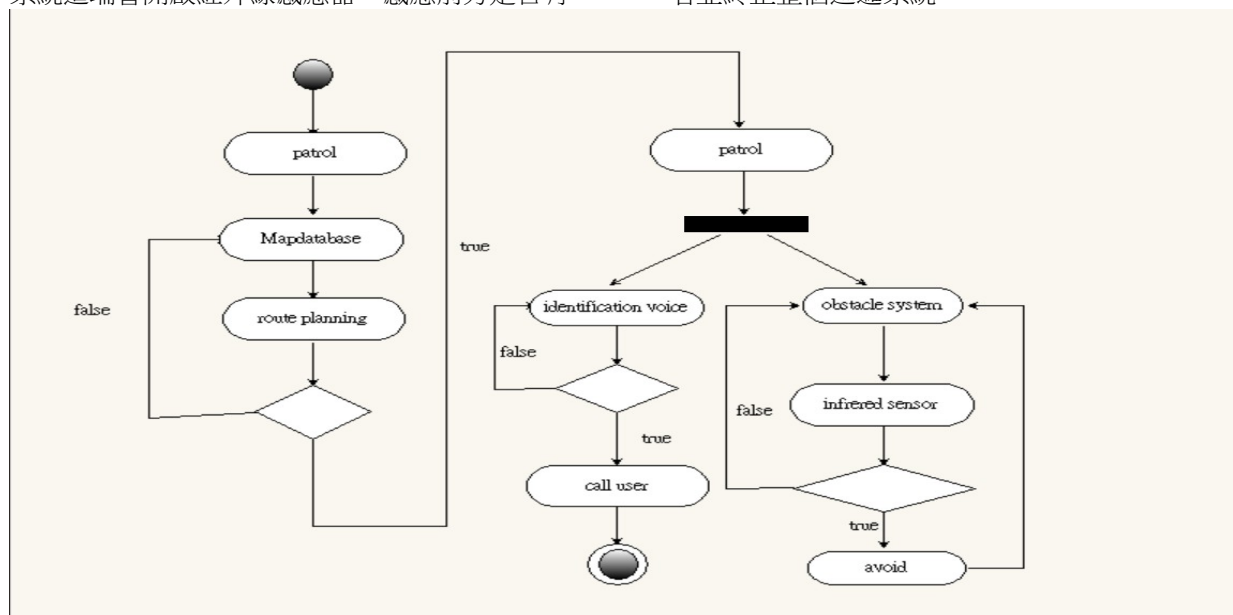


圖 13. 《Zenbo：平面巡邏機器人》UML 巡邏系統活動圖