



嶺東科技大學  
LING TUNG UNIVERSITY

資訊管理系

# MVC 架構的網站開發：以 零件 庫存系統為例

指導教授： 夏則智 教授

馮曼琳 教授

組員名單： 林文祺 A18C010

邱信達 A18C070

林宜穎 A18C133

陳姿仰 A18C145

邱琦智 A18C154

中華民國 105年 6月



嶺東科技大學  
LING TUNG UNIVERSITY

資訊管理系

# MVC 架構的網站開發：以 零件庫存系統為例

指導教授： 夏則智 教授

馮曼琳 教授

組員名單： 林文祺 A18C010

邱信達 A18C070

林宜穎 A18C133

陳姿仰 A18C145

邱琦智 A18C154

指導教授簽名

嶺東科技大學

資訊管理系

MVC架構的網站開發：  
以零件庫存系統為例

中華民國一〇五年六月



嶺東科技大學  
LING TUNG UNIVERSITY

資訊管理系專題口試委員審定書

# MVC 架構的網站開發：以零件 庫存系統為例

指導教授： 夏則智 教授

馮曼琳 教授

組員名單： 林文祺 A18C010

邱信達 A18C070

林宜穎 A18C133

陳姿仰 A18C145

邱琦智 A18C154

指導教授：  
\_\_\_\_\_

口試委員：  
\_\_\_\_\_

中華民國

年

月

日

## 謝 誌

隨著專題圓滿完成之際，也意味著求學階段也將告一個段落；在這漫長歲月中，特別感謝夏則智老師及馮曼琳不辭辛勞的教導。從最初沒有頭緒的討論主題，則智老師與曼琳老師總是耐心的指導我們，在研究過程中遭遇困難他們總是耐心溫柔的回答，這段期間我們受益良多，對於老師的感謝我們永遠銘記在心，也辛苦老師這些時日的付出與指導。

其次，感謝口試委員則智老師以及江富老師給予我們口試時的指導與講評，更感謝口試委員費心的審視報告書，在遇到困難時所給予的教導，是我們成長的好機會！

最後，感謝系上諸位老師在各學科領域的熱心指導，助我們提升實務上的經驗以及我們的本質學能，增進資訊管理知識範疇，讓我們擁有更堅強的實力踏入社會及職場，在此一併致上最高謝意。

林文祺、邱信達、林宜穎、陳姿仰、邱琦智 謹誌

中華民國 105 年 5 月於嶺東

## 摘要

本專題將開發一個簡單的零件庫存管理系統。總共分為系統分析、介面設計、系統實作三個階段。在系統分析階段中使用 Software Ideas Modeler 來畫出 UML 裡的使用個案圖、活動圖及類別圖。在介面設計階段則是使用了 GUI Design Studio 以及 W2UI，GUI Design Studio 的快速、易於使用以及無需任何編碼的優點，使其成為軟體設計師、業務分析師、開發人員和諮詢專家的省選原型設計工具。在系統實作階段，我們使用了 XAMPP，包含 Apache(網頁伺服器)、MySQL(資料庫，儲存資料使用)、PHP(腳本語言解析，搭配 Apache)、Perl 來當作主要架構，因此可以更容易開發滿足使用者需求的資訊系統。

關鍵詞：統一塑模語言(UML)、Software Ideas Modeler、GUI Design Studio、XAMPP、MySQL

# 目錄

摘要.....	I
表目錄.....	III
圖目錄.....	IV
<b>第一章 緒論.....</b>	<b>3</b>
1.1 研究背景.....	3
1.2 研究動機.....	3
1.3 研究目的.....	4
<b>第二章 文獻回顧與探討.....</b>	<b>5</b>
2.1 統一塑模語言.....	5
2.1.1 UML 的定義與圖形架構.....	5
2.1.2 UML 發展階段使用圖.....	8
2.2 XAMPP.....	13
2.2.1 XAMPP 軟體.....	13
2.3 GUI Design Studio.....	14
2.3.1 介面設計畫面.....	14
2.4 PHP (Hypertext Preprocessor).....	16
2.4.1 PHP 編寫畫面.....	16
<b>第三章 研究方法.....</b>	<b>19</b>
3.1 研究方法.....	19
3.2 研究流程.....	20
3.3 資料庫規畫流程.....	21
3.4 ERD 實體關係圖.....	22

3.5 零件活動圖 .....	23
3.6 進貨活動圖 .....	24
3.7 出貨活動圖 .....	25
3.8 預定進度甘特圖表 .....	26
3.9 工作分配表 .....	27
<b>第四章 系統實作 .....</b>	<b>28</b>
4.1 系統需求分析 .....	28
4.1.1 CIM-1:定義企業流程 .....	28
4.1.2 CIM-2:分析企業流程 .....	29
4.1.3 CIM-3 定義系統範圍 .....	30
<b>第五章 成果與結論 .....</b>	<b>32</b>
5.1 成果展示 .....	33
5.2 結論 .....	41
參考文獻 .....	42

## 表目錄

表 1 UML2.2 圖形分類架構表(共 14種圖形).....	6
表 2使用案例圖元件介紹.....	8
表 3:工作分配表.....	27
表 4 企業 UC簡述 .....	29
表 5 系統 UC 描述.....	31

## 圖目錄

圖 1 UML2.2圖形階層架構圖 .....	6
圖 2：4+1觀點 .....	7
圖 3：XAMPP開啟畫面 .....	13
圖 4:設計畫面 .....	14
圖 5:設計畫面 .....	15
圖 6:設計畫面圖 .....	15
圖 7:編寫畫面圖 .....	16
圖 8:編寫畫面圖 .....	17
圖 9:編寫畫面圖 .....	17
圖 10:編寫畫面圖 .....	18
圖 11:研究流程圖 .....	20
圖 12:資料庫流程規劃圖 .....	21
圖 13: ERD 實體關係圖 .....	22
圖 14: 零件活動圖 .....	23
圖 15: 進貨活動圖 .....	24
圖 16: 出貨活動圖 .....	25
圖 17: 預定進度甘特圖 .....	26
圖 18: 企業 UC 圖 .....	28
圖 19: 企業一般流程活動圖 .....	29
圖 20: 客戶能啟動的系統 UC 圖 .....	30
圖 21: 業務能啟動的系統 UC 圖 .....	30
圖 22: 系統完成畫面圖 .....	32
圖 23: 供應商管理畫面圖 .....	33
圖 24: 供應商管理-新增資料圖 .....	33
圖 25: 供應商管理-資料修改圖 .....	34
圖 26: 訂單管理圖 .....	34
圖 27: 零件管理圖 .....	35

圖 28: 零件管理-新增資料圖.....	35
圖 29: 零件管理-修改資料圖.....	36
圖 30: 零件管理-刪除資料圖.....	36
圖 31: 零件進貨圖.....	37
圖 32: 零件選擇圖.....	37
圖 33: 成品管理圖.....	38
圖 34: 成品管理-新增資料圖.....	38
圖 35: 成品管理-修改資料圖.....	39
圖 36: 成品管理-刪除資料圖.....	39
圖 37: 使用零件圖.....	40
圖 38: 出售圖.....	40

# 第一章 緒論

## 1.1 研究背景

傳統產業使用人力與紙本的方式管理包括的人事資料、廠商資料、顧客資料及商品庫存資料等等，但是使用這種方式不但沒有效率還放浪費很多的人力資源，更可能增加公司成本讓利潤降低網路時代的來臨，人們可以利用網路實現很多不可能實現的願望，隨著資訊社會越來越發達，人們的生活也越來越便利，我們將利用 XAMPP 設定及利用統一塑模語言(UNL)加上使用者圖形介面 GUI Design Studio，架設出零件庫存管理系統

庫存的定義為一個機構中所存放的需用項目或資源，一個庫存系統就是決策和控管措施，用以監控庫存量、什麼時候必須補充庫存急需採購的量；存貨管理的意義在於控制存量與訂貨次數的均衡，借以提高生產力並提高庫存週轉率及銷貨利益，使資金獲得合理應用。

## 1.2 研究動機

現在網路時代的來臨，現今資訊社會的科技已今非昔比，在以前的社會資料管理這方面常常使得產業需要投入一筆成本到這一方面，由於管理是以人力與紙本的方式，所以使資料保存並不完整與錯誤率也高了一些。

如今教授給了我們這個研究題目，而我們小組討論過後也想發揮我們所學的專長，使得現在的產業能利用現今的科技發達使他們在資料管理的方面能減少所花費的時間與人力，問題發生時能迅速並清楚的解決問題，進而使得公司產業的成本大大的減少並提高許多的便利性。

不良的存貨管理，將會導致存貨過多，而囤積太多存貨，則會增加倉儲費用與流動資金的要求，形成種種風險成本，進而降低了獲利能力。所以就想運用資訊科技的技術，設計一個系統，可以替代傳統的人力管理。

### 1.3 研究目的

現在是科技的社會，使用電腦來管理資料，跟從前的紙本相比，資料的保存性以及完整性提高了許多，在問題發生的時候在查詢的方面能比從前快速許多，大大提升解決問題的效率，現在上市上櫃的公司基本上都備有一套屬於自己的系統，但是相對於中小型企業，並沒有足夠的資金與人力去管理這套系統，所以本專題設計出一套零件管理系統，來解決中小企業資金不足，人手也不夠的問題，這套系統非常的簡單操作，也能輕鬆的管理過多且複雜的貨品。

我們利用 XAMPP 設定及利用統一塑模語言(UNL)加上使用者圖形介面 GUI Design Studio，架設出零件庫存管理系統，簡單不困難，清楚不複雜的介面使得一般的人就可輕易上手，利用科技去有效的管理我們的資料，讓我們在眾多選擇當中能讓大家去選擇使用我們所研發的零件管理系統，來提高公司的利潤、減少公司的成本、提升大家使用上的便利性，在這科技發達的社會，使人們可以利用科技網路實現許多以前所達不到的事情。

管理庫存的目的為了維護作業的獨立性，工作站中物料的充分供給能使此工作連續運作，不管是對於供應商或是對於零售商不可或缺最終要的一環。

## 第二章 文獻回顧與探討

### 2.1 統一塑模語言

UML 是由三位物件導向方法論專家 Grady Booch、James Rumbaugh 及 Ivar Jacobson 共同合作制定的物件導向塑模語言。根據物件管理組織 (Object Management Group, OMG) 的定義「用來詳細說明、呈現、建置、記錄軟體系統工作成果的一種塑模語言,它也可以用來建立企業模型以及其它非軟體系統。」

UML 塑模語言的優點：(1) Grady Booch 的 Booch method 是設計及建置導向的方法,對強調工程的系統提供極佳的表示法。(2) James Rumbaugh 的 OMT 是分析導向的方法,對強調資料的系統提供極佳的表示法。(3) Ivar Jacobson 的 OOSE 是 Use-Case 導向的方法,對強調商業工程及需求分析的系統,提供極佳的表示法。

UML 的軟體架構觀點：從使用者、系統開發者等多重觀點,架構一套完整的自分析、設計到產生程式碼的塑模語言,其中包含了所謂的「4+1 觀點」。

#### 2.1.1 UML 的定義與圖形架構

統一塑模語言 (UML: Unified Modeling Language) 是非專利的第三代塑模和規約語言。UML 是一種開放的方法,用於說明、可視化、構建和編寫一個正在開發的、物件導向的、軟體密集系統的製品的開放方法。UML 展現了一系列最佳工程實踐,這些最佳實踐在對大規模,複雜系統進行塑模方面,特別是在軟體架構層次已經被驗證有效。

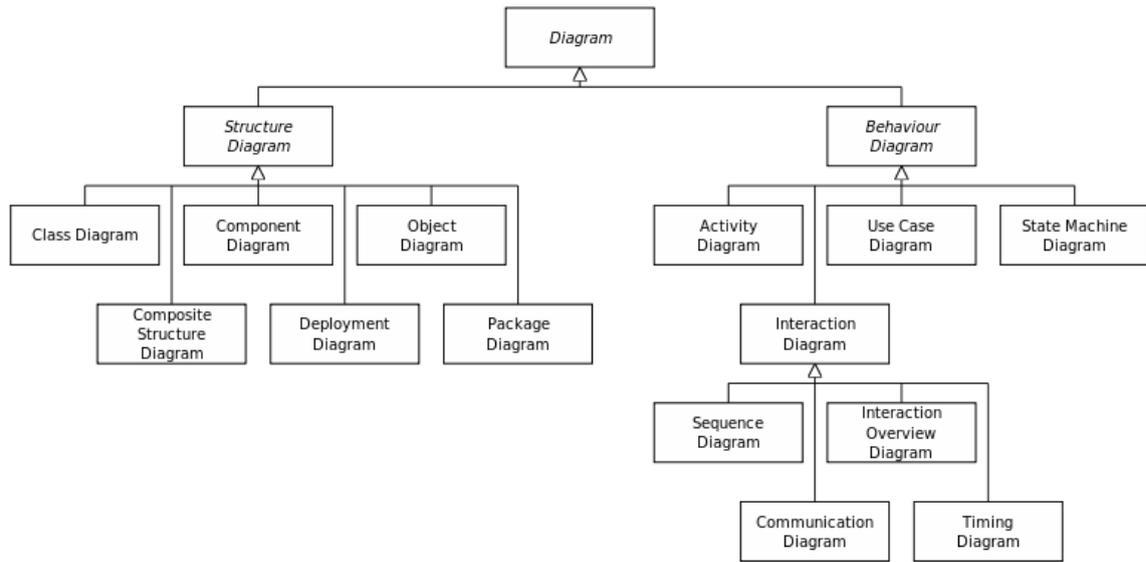


圖 1 UML2.2 圖形階層架構圖

表 1 UML2.2 圖形分類架構表(共 14 種圖形)

結構性圖形 (Structure diagrams) 強調的是系統式的塑模：	靜態圖 (static diagram)	類別圖(Class Diagram)
		物件圖(Object diagram)
		套件圖(Package diagram)
	實現圖 (implementation diagram)	組件圖(Component diagram)
		部屬圖(Deployment diagram)
	剖面圖(profile diagram)	
複合結構圖(Composite structure diagram)		
行為式圖形 (Behavior diagrams) 強調系統模型中觸發的事件：	活動圖(Activity diagram)	
	使用個案圖(Use Case Diagram)	
	狀態圖(State diagram)	
	交互性圖形 (Interaction diagrams)，屬於行為圖形的子集合，強調系統模型中的資料流程：	通訊圖 (Communication diagram)
		互動概觀圖 (Interaction diagram)
		時序圖(Timing Diagram)
		循序圖 (Sequence diagram)

(1) 使用案例觀點(Use Case View)

可說是 UML 的核心,顯示外界的行動者(Actiors)所使用到的系統功能。這個觀點並不說明一個軟體系統內部的構造,而是純粹地解說系統與行動者間的互動。

(2) 邏輯觀點(Logical View)

這個觀點主要顯現出系統內部如何透過物件類別的責任分配與互動運作以提供系統功能,提供系統設計開發者使用。

(3) 實作觀點(Implementation View)

相關的一組類別經編譯之後產生可執行的軟體元件,其間的互動說明系統內部各模組元件的資訊及彼此之間的相依性(Dependency)。

(4) 程序觀點(Process View)

在這個觀點中考量的重點就是如何透過對執行程序,或是執行緒(Thread) 的組態(Comfiguration), 來完成系統功能性的需求。同時能符合系統中非功能性(Nonfunctional)的屬性。

(5) 部署觀點(Deployment View)

這個觀點主要顯示出系統軟硬體架構的實體佈署。

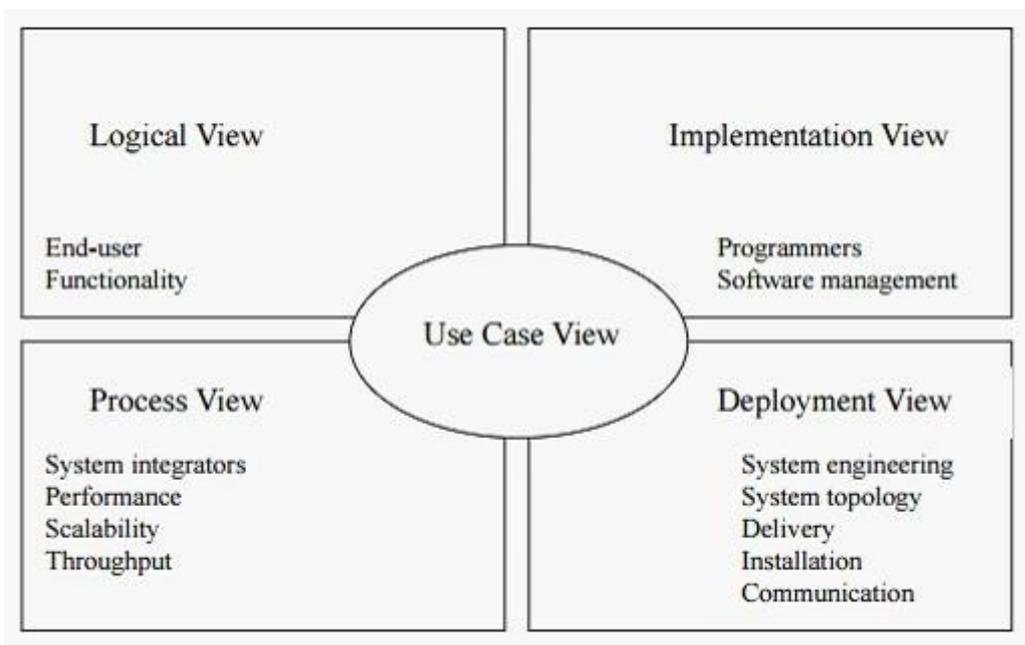


圖 2：4+1 觀點

UML 定義了九種型態的圖形供各發展階段使用:類別圖(Class Diagram)、物件圖 (Object Diagram)、使用案例圖 (Use Case Diagram)、循序圖 (Sequence Diagram)、合作圖(Collaboration Diagram)、狀態圖(Statechart Diagram)、活動圖 (Activity Diagram)、元件圖(Component Diagram)和部署圖(Deployment Diagram)。所有圖形都是圍繞著使用案例繪製,並且依循物件導向分析設計的循序漸進發展特性進行系統發展。繪製圖形的基本原則就是將概念以符號描繪,並將概念之間的相互關係描繪為連接符號的路徑。以下分別詳述各種圖形的意義和用途

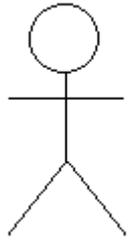
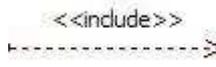
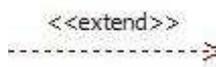
### 2.1.2 UML 發展階段使用圖

#### 1. 使用案例圖(Use Case Diagram)

Use Case 描述系統做了什麼(What)而非如何做(How), 是行動者(Actor) 為了達到某一目的而與系統有所互動的情形。Use Case 是需求陳述,主要表達的是功能需求。並以文字描述行為者與系統之間一連串的互動。

使用案例圖敘述了系統的功能,以及此系統的使用者,更仔細的說,獲取系統需求並表達使用案例與行動者間的靜態相互關係,使用個案圖中,行為者常以人形表示,使用個案以橢圓表示,使用個案與使用個案之間的關係如表所示。

表 2 使用案例圖元件介紹

元件	元件圖示
行為者(Actor)	
使用個案(Use Case)	
行為者與使用個案間之互動	
使用個案間之 Include 關係	
使用個案間之 Extend 關係	
使用個案間之 Generalization 關係	

## 2. 活動圖(Activity Diagram)

活動圖敘述一個類別的活動,即探索與描述在一個類別裏的一個工作流(Workflow),此工作流是類別的操作所進行的一些行動(Action)。活動圖也可以應用於描述組織內部的企業程序(Business Processes)及工作流,活動圖是狀態圖的特例,而且也使用相似的表達方式,但活動圖所描述的是一個類別由一個活動到下一個活動之間的控制流。狀態圖則是描述類別回應外部事件,由一個狀態轉為另一個狀態之間的控制流。

活動圖和狀態圖非常類似。這些圖形是用來了解操作方法,為了執行某些預期的行為或功能以及在系統中這些物件如何與其它物件互動,系統中的類別必須涵蓋這些操作方法。活動圖是系統動態模型的一部份。

## 3. 類別圖(Class Diagram)

類別圖述的是系統的靜態結構,或是如何建立此結構,而非它的行為。

## 4. 狀態圖(Statechart Diagram)

狀態圖敘述一個類別對於外來刺激(事件)的反應的回應和狀態的改變。狀態圖將物件、子系統、系統在生命週期間,可能的狀態、對不同的事件所造成的狀態的改變都記錄下來。

狀態圖有兩個動態的表現,分別是交互關係、內部狀態的改變。交互關係描述物件對外的行為(Behavior),如送訊息或與其它物件發生連結和中斷連結。內部狀態的改變是描述物件如何改變,如物件屬性的不同狀態。我們可從狀態圖得知、了解訊息和操作方法。為了執行某些預期的行為或功能,系統中的類別必須提供這些訊息和操作方法。狀態圖是系統動態模型的一部份。

## 5. 循序圖(Sequence Diagram)

循序圖敘述物件之間的互動關係。這些互動關係被塑模為訊息的傳送。這些圖所關注的焦點在物件及它們所傳送的訊息,藉以達成某些預期的行為。循序圖其實是一種互動圖,簡言之就是以訊息傳送的先後順序來表達物件之間的互動關係。

循序圖可以用來表達 Use Case 中的情節,就像物件圖可以用來確認類別模型的合理性,循序圖也可以用來確認情節的合理性。一組循序圖的集合則是系統之動態模型的一部份。它們指明了類別物件角色之間的訊息交換,藉以實現使用案例。

## 6. 合作圖(Collaboration Diagram)

合作圖敘述參與互動的物件之間的組成關係。這些互動被塑模成物件之間的訊息交換。簡單的說合作圖就是描述一組物件之間的訊息交換,它其實是一種互動圖。

合作圖可以用來表達使用案例中的情節。就像物件圖可以用來確認類別模型的合理性,合作圖也可以用來確認使用案例的情節的合理性。這項確認包括了合作關係所使用的物件角色,以及存在於物件角色中的結合關係角色。一個合作圖的集合則是系統之動態模型的一部份。合作圖與循序圖所表達的意涵相同。循序圖是以時間為主軸,而合作圖是以空間為主軸。

## 7. 物件圖(Object Diagram)

物件圖描述了某一特定時刻系統的靜態結構。即類別模型所敘述的是所有狀況的概觀,物件模型所敘述的則是一個特定的狀況。

物件圖通常用來確認一個系統之類別模型的合理性。如果我們發現有一個物件圖會使類別模型產生例外,我們就必須改變類別模型,或是決定此物件圖是錯誤的。

## 8. 元件圖(Component Diagram)

元件圖敘述軟體實作元件的組織結構及其相依關係。這些圖形包含元件,可視為系統中一個實體及可替代的部分,主要是用來表現實做時的實體架構,包括原始程式碼、目的碼、以及可行程式碼。元件與類別的差別在於類別是真實世界的個體抽象化的表現,而元件是真實世界的實體事物。

元件圖最主要在表現當系統開發完成後的軟體架構,以及元件之間彼此之間相依關係,通常會與下面要介紹的部署圖結合作為系統之軟、硬體實體架構模型。

## 9. 部署圖(Deployment Diagram)

部署圖敘述處理硬體資源元素的組態,以及軟體實作元件的對應方式。這些圖形包含了元件和節點,節點表示一種處理或計算的資源,包括電腦、印表機等。即是用來描述軟體系統裡的硬體佈署情形。在企業模型裏並不使用此圖。

## 2.2 XAMPP

XAMPP 是一個功能強大的軟體。它可以在 Windows、Linux、Solaris 和 Mac OS X 的等多種操作系統下安裝使用，並且支持多種語言如英文、簡體中文、繁體中文、韓文、俄文、日文等。而 XAMPP 中，X 就是多系統的意思，因為它可以跨多種平台使用，A 就是 Apache 網頁伺服器、M 就是資料庫系統 MySQL，而兩個 P 則是 PHP 與 Perl。XAMPP 是最流行的 PHP 開發環境，是完全免費且易於安裝的版本，其中包含 MariaDB、PHP 和 Perl。XAMPP 開放源碼套件的設置讓安裝和使用將會更加容易。

### 2.2.1 XAMPP 軟體

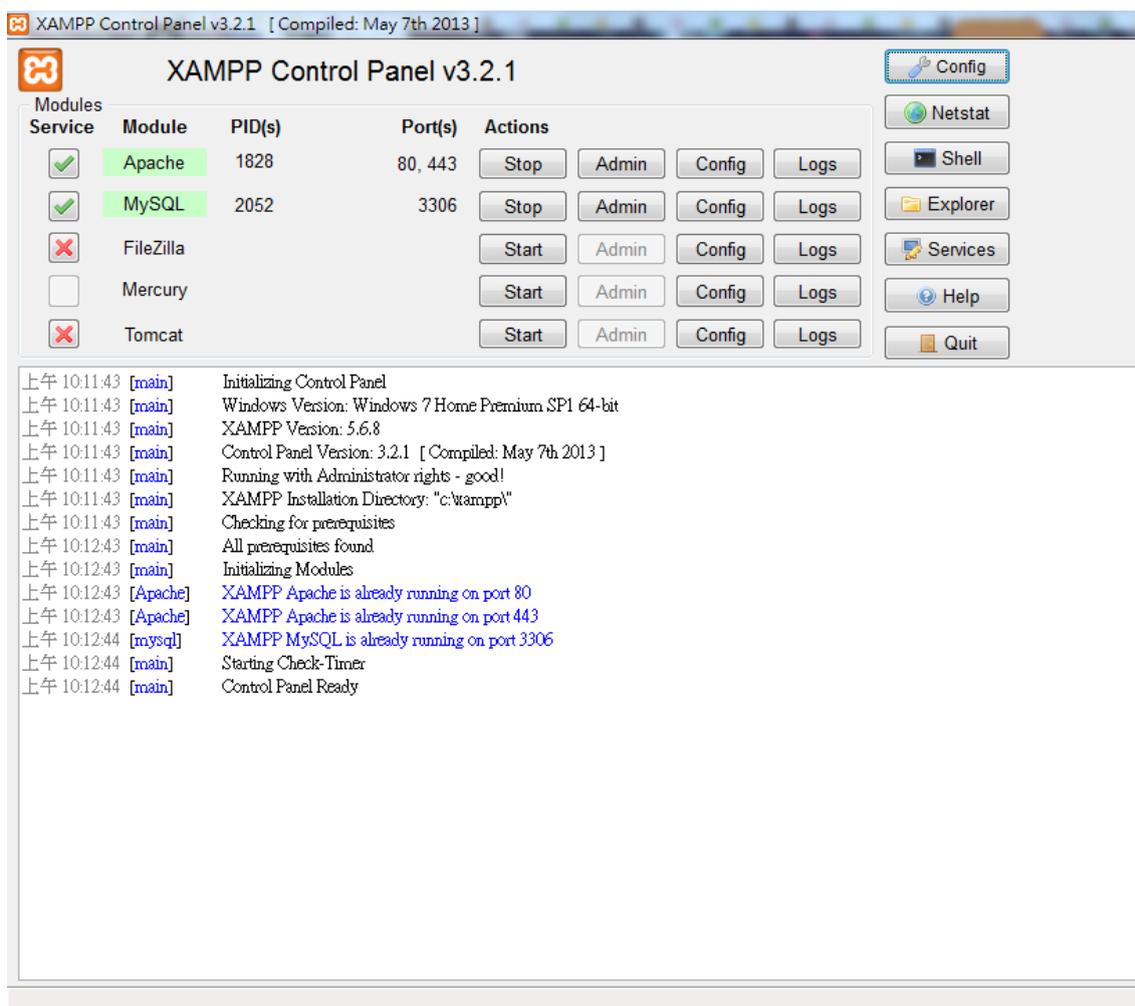


圖 3：XAMPP 開啟畫面

## 2.3 GUI Design Studio

縮小設計者與使用者認知之差異：

設計者之認知模型直接地表達在人機介面中軟體的操作互動模型中，如果與使用者的認知模型不一致的話，介面所表達的意象很容易被使用者誤解，使用者因而無法了解電腦系統究竟如何反應他的命令，當然也就沒辦法預期電腦系統如何來反應他的操作，也就無法順利地找出達成目的的一種操作方法了。

### 2.3.1 介面設計畫面

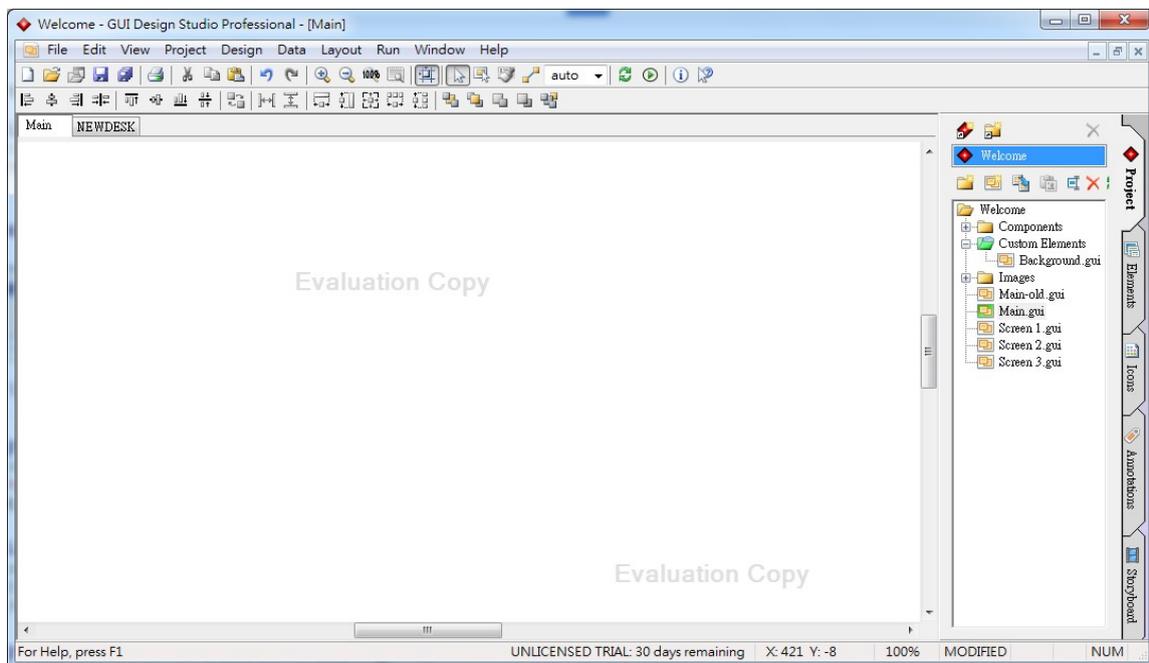


圖 4:設計畫面

圖 4 是說明 GUI 使用者介面設計類似文書處理的介面讓使用者一目瞭然

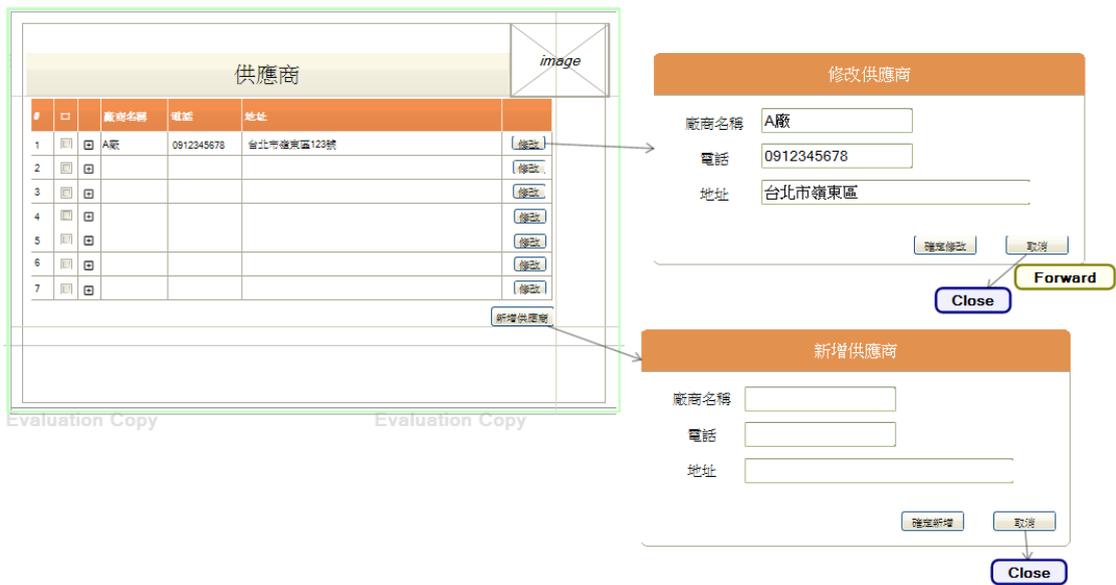


圖 5:設計畫面

圖 5 是說明供應商與零售商現階段雛型為相似，但在後台的新增修改有所不同



圖 6:設計畫面圖

## 2.4 PHP (Hypertext Preprocessor)

超文字預處理器，是一種通用的電腦程式語言，尤其適用於網路開發並可嵌入 HTML 中使用。PHP 的主要目標是允許網路開發人員快速編寫動態頁面。

PHP 的應用範圍相當廣泛，尤其是在網頁程式的開發上。一般來說 PHP 大多執行在網頁伺服器上，透過執行 PHP 程式碼來產生使用者瀏覽的網頁。PHP 可以在多數的伺服器和作業系統上執行，而且使用 PHP 完全是免費的。

### 2.4.1 PHP 編寫畫面



圖 7:編寫畫面圖



圖 8:編寫畫面圖

圖 8 是說明目前網站邊寫畫面，目前較為簡單更多細項會陸續新增



圖 9:編寫畫面圖



圖 10:編寫畫面圖

# 第三章 研究方法

## 3.1 研究方法

本研究之研究目標是為了幫助企業組織，管理供應商與廠商，讓管理者能夠迅速的了解企業組織之營運狀況，製作出網頁架構，為了讓溝通上更淺顯易懂所以我們也利用介面設計圖，以便管理者能快速的找到差異之所在，以便在最短的時間內做出決策

系統使用的工具，皆是目前市面上容易找到且主流好用的軟體，如 PHP、MySQL 等等，結合容易上手使用的網頁型式開發，讓系統的使用變得簡單，只要有電腦和網路便能即時控管資訊。結合物料管理與電腦 e 化的庫存管理系統，在賣場或企業的營運中扮演不可或缺的角色，但作業的繁複讓系統的建構也十分困難，而在經過系統的分析工作與使用者的需求和研究賣場的作業後，逐漸能夠勾勒出一些雛形。

## 3.2 研究流程

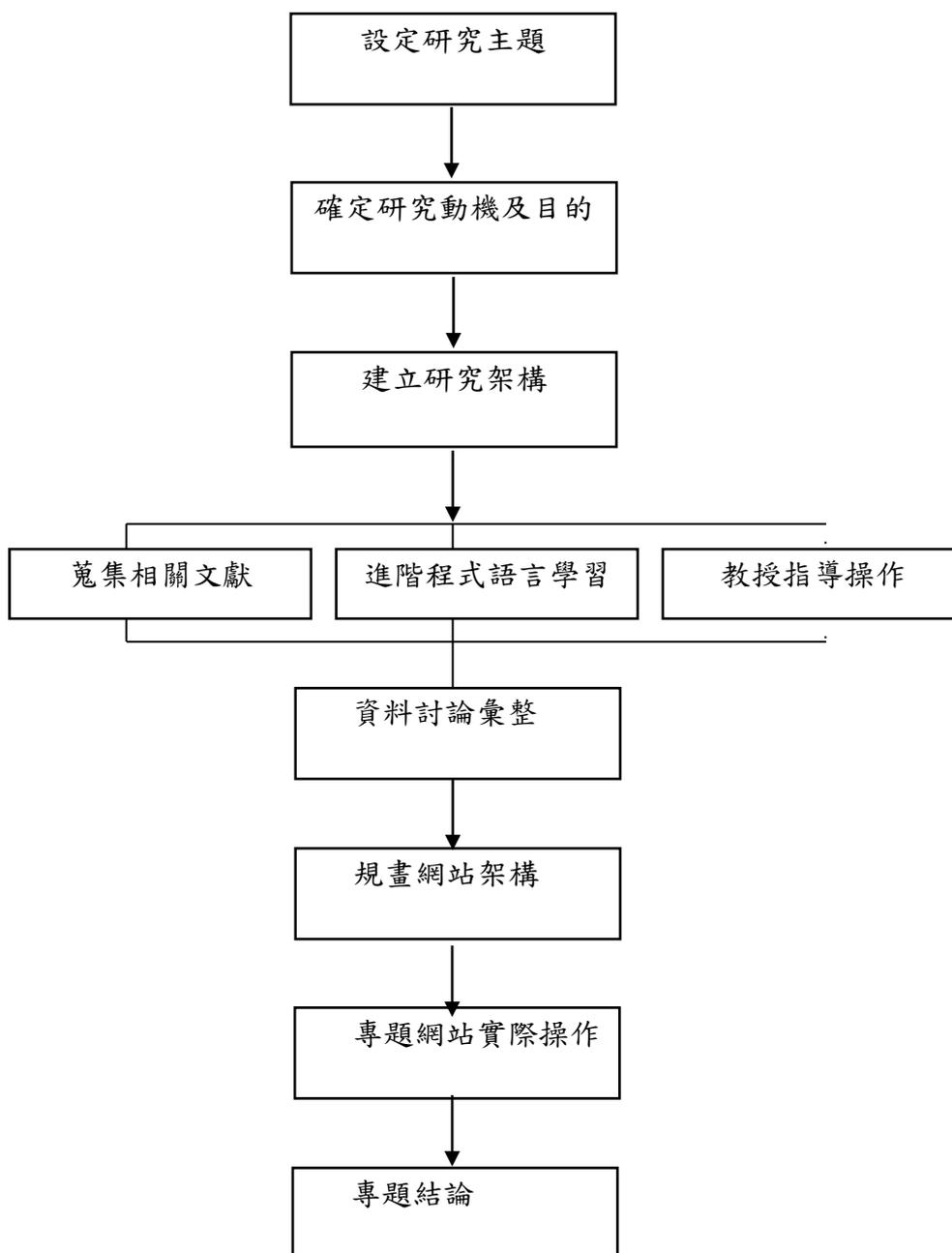


圖 11:研究流程圖

### 3.3 資料庫規畫流程

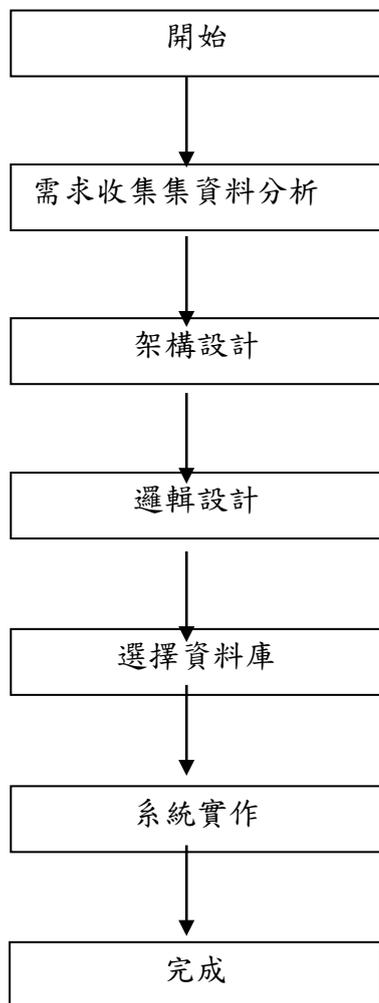


圖 12: 資料庫流程規劃圖

### 3.4 ERD 實體關係圖

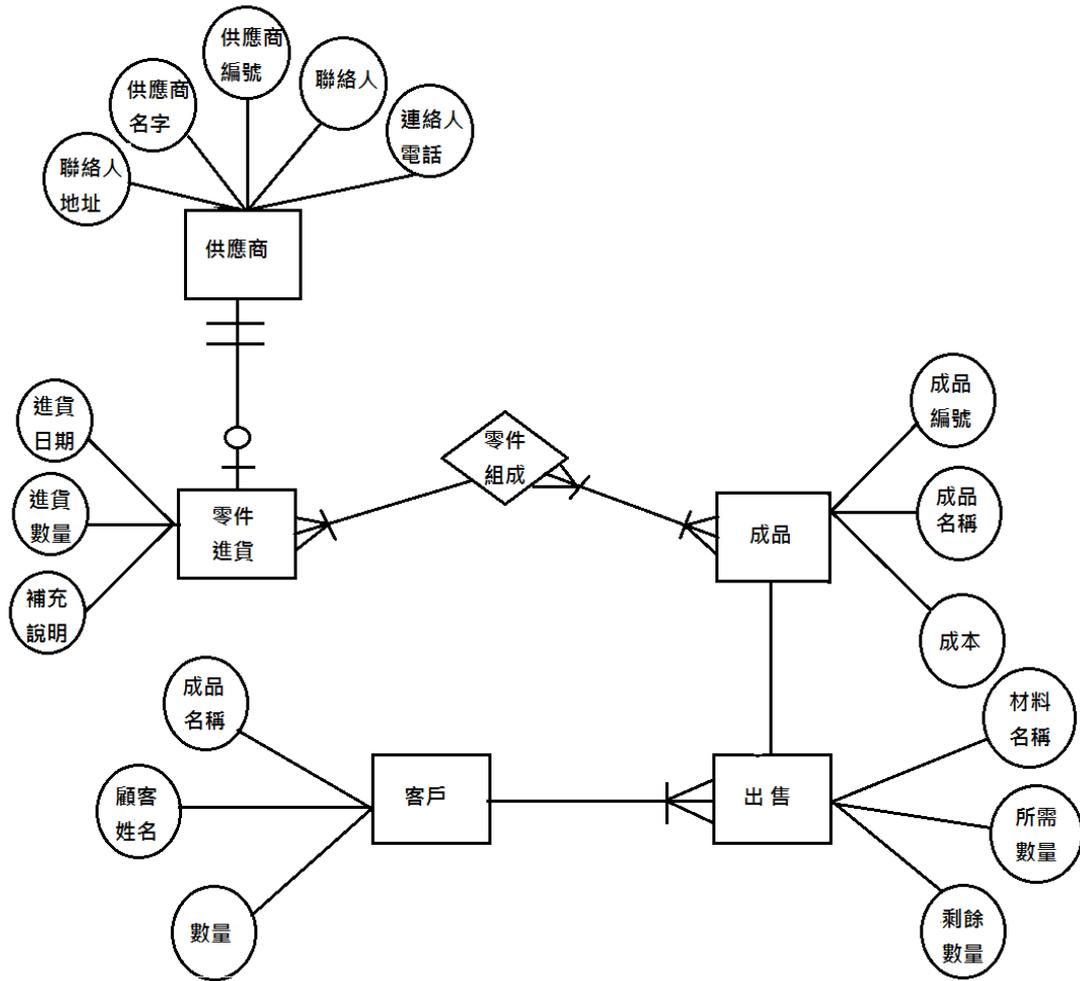


圖 13 ERD 實體關係圖

### 3.5 零件活動圖

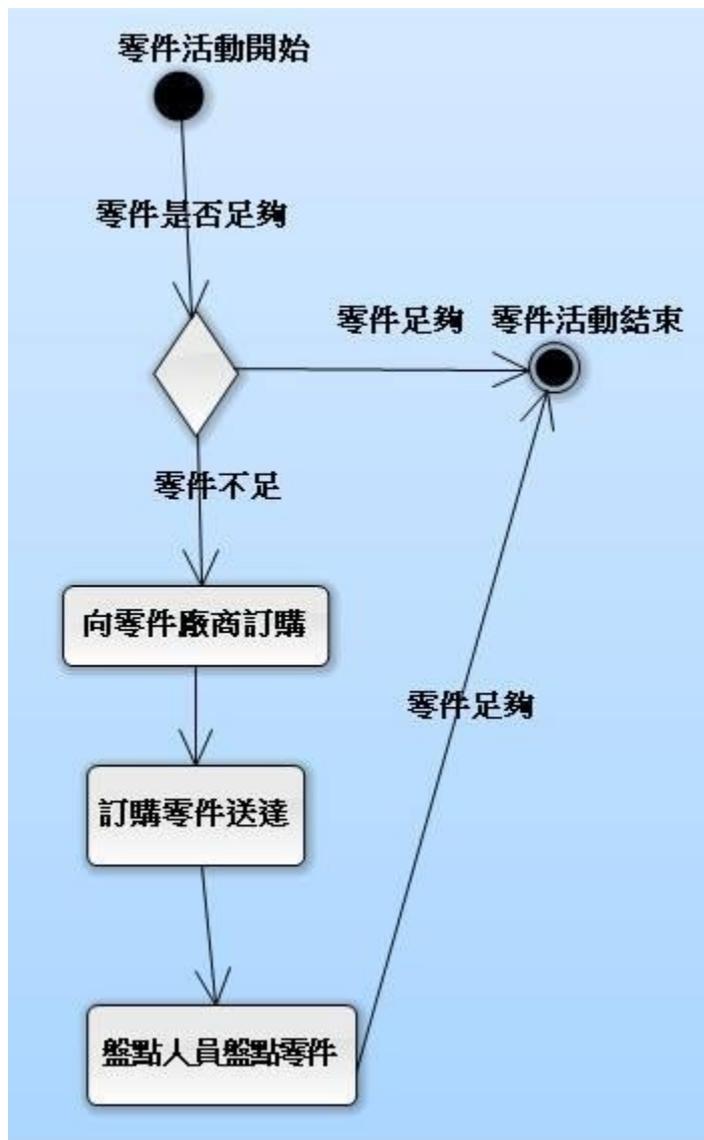


圖 14:零件活動圖

圖 14 是當零件活動開始會先進行判斷零件的數量是否足夠如不足的話就直接向廠商訂購，當零件充足的話則直接出貨，零件活動結束。

### 3.6 進貨活動圖

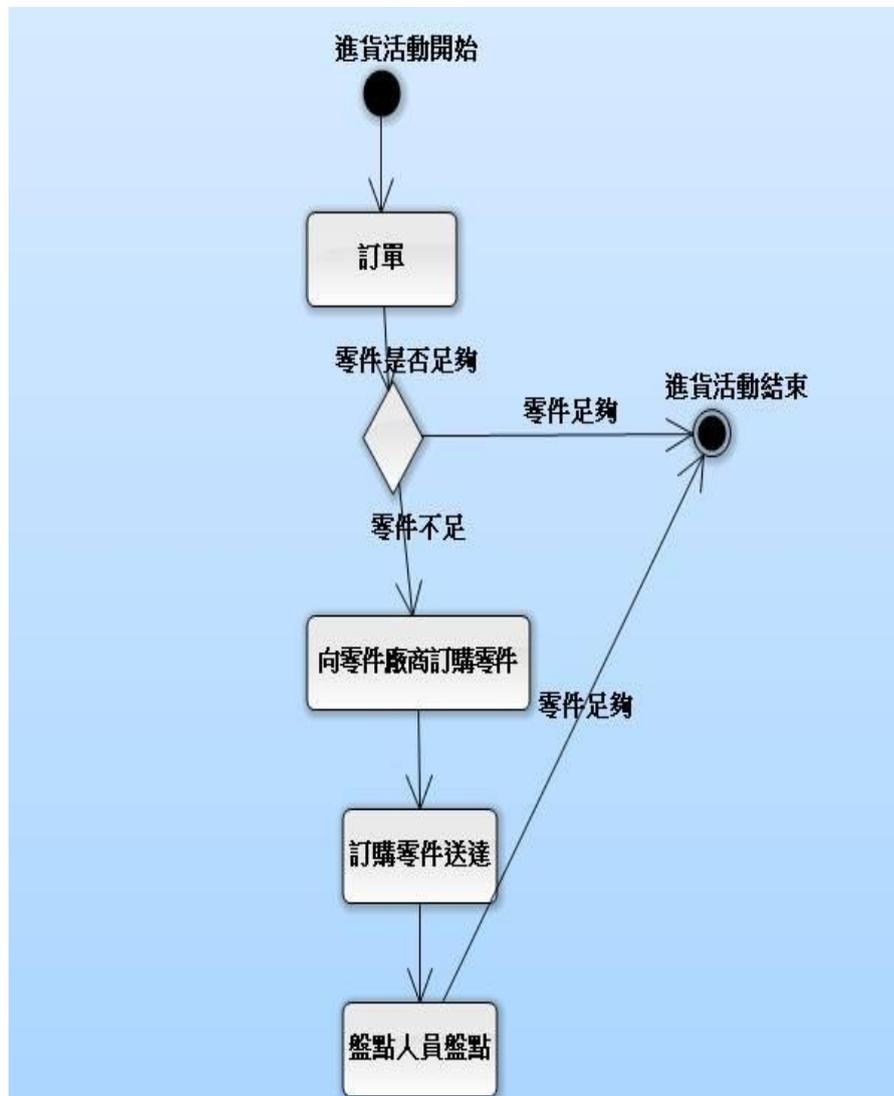


圖 15:進貨活動圖

圖 15 是當進貨活動開始時先判斷所訂購數量是否足夠，如果不足則向廠商進行訂購，如果足夠則進貨活動結束。

### 3.7 出貨活動圖

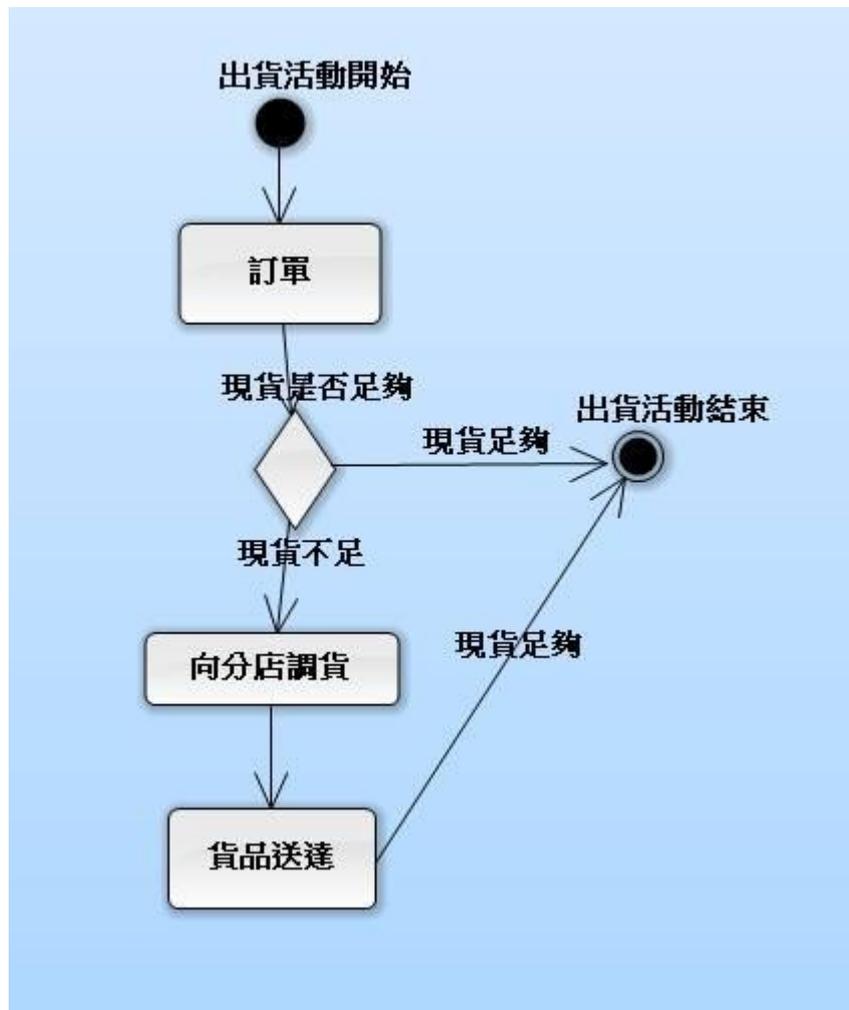


圖 16: 出貨活動圖

圖 16 是當出貨活動開始時先確認訂單內所需要的量在現貨是否充足，如不足則先行向分店調貨，如現貨足夠則出貨，出貨活動結束。

### 3.8 預定進度甘特圖表

	103年							104年					負責組員	
	月3	月5	月6	月7	月9	月10	月11	月12	月1	月2	月3	月4		月5
研究動機														全體
確定主題														全體
蒐集資料														林文祺 邱琦智
程式學習														邱信達 邱琦智
程式編寫														邱信達 陳姿仰
介面繪製														林宜穎 林文祺
程式除錯														邱信達
製作成果報告														林宜穎
完成度	10%	12%	14%	20%	28%	35%	56%	66%	75%	89%	95%	100%	X	

圖 17: 預定進度甘特圖

### 3.9 工作分配表

表 3:工作分配表

	陳姿仰	林宜穎	邱信達	邱琦智	林文祺
蒐集資料	◎	◎	◎	◎	◎
環境分析	□	□	◎	◎	◎
文書處理	◎	□	□	□	□
排版美編	◎	◎	□	□	□
撰寫程式	□	□	◎	□	□
介面繪製	□	◎	□	□	□
程式除錯	□	□	◎	◎	◎
成果報告	□	□	□	◎	□
※註記：◎主要、□次要					

# 第四章 系統實作

## 4.1 系統需求分析

本節主要說明流程上的需求分析的截取與轉換，如何運用以模型驅動架構的三階段的步驟使系統快速完整開發，三階段分別為 CIM、PIM 及 PSM。此外，CIM 階段又分為三步驟，(1)CIM-1:定義企業流程，產出企業 UC 模式(2)CIM-2:分析企業流程，產出活動圖(3)CIM-3: 定義系統範圍，產出系統 UC 圖。

### 4.1.1 CIM-1:定義企業流程

CIM-1 定義企業流程的產出，主要有企業 UC 圖和企業 UC 簡述，企業 UC 圖中，每一個企業 UC 代表一條企業流程，企業參與者 則代表位於企業外但會參與企業流程者，企業 UC 簡述則是用來記錄 和分辨企業流程。

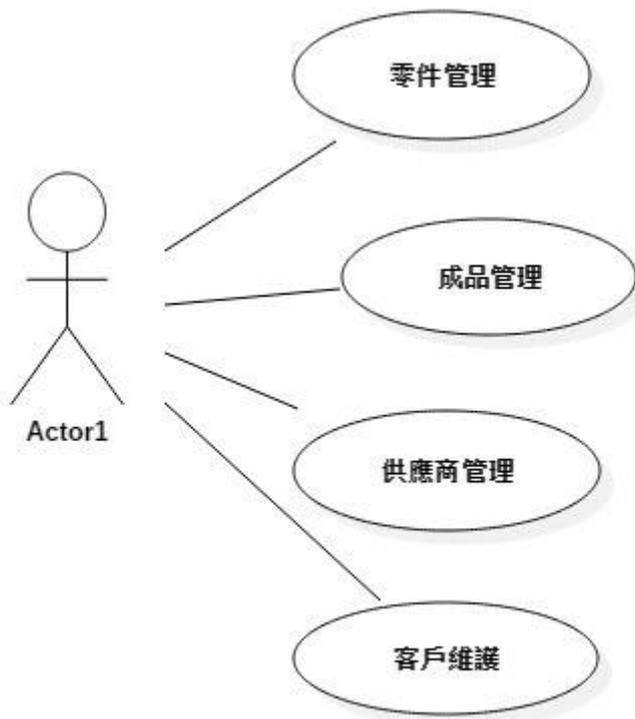


圖 18:企業 UC 圖

表 4 企業 UC 簡述

企業 UC 名稱	簡述
零件管理	當使用者定貨時，零件就會有所短缺，所以需要定時管理。
成品管理	當使用者完成成品時，需要定時管理。
供應商管理	當使用者向廠商進貨時，需要做進貨管理。
客戶維護管理	當完成成品並且也做好管理後，若有訂單則需從出貨系統出貨。

### 4.1.2 CIM-2:分析企業流程

針對 CIM-1 的每一個企業 UC，分析每一個的工作流程，並繪製活動圖。

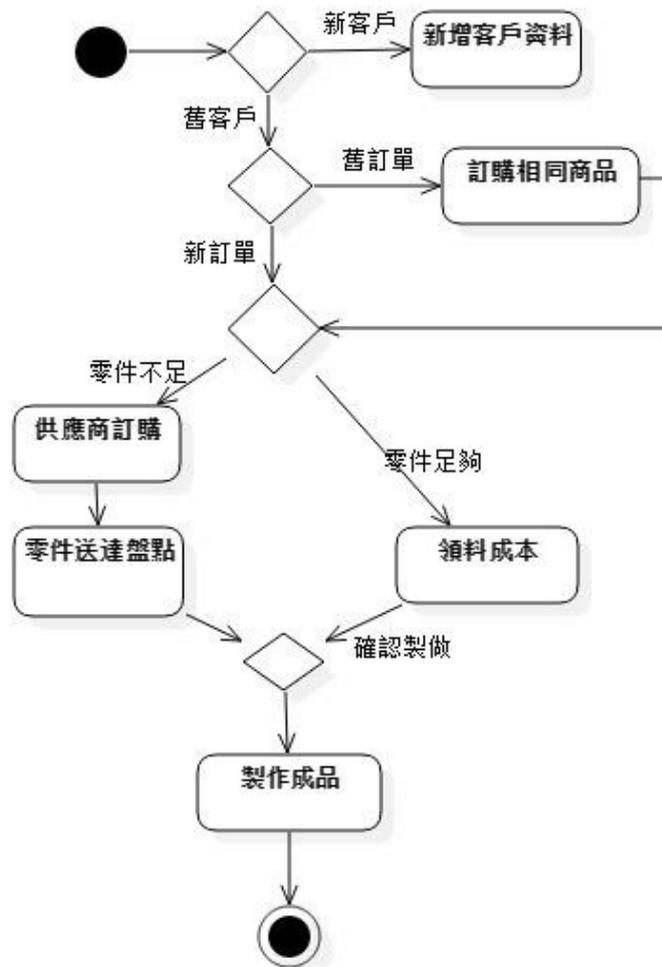


圖 19:企業一般流程活動圖

### 4.1.3 CIM-3 定義系統範圍

在 CIM-2 活動圖中的每一個行動，都可能成為 CIM-3 的系統 UC，而 CIM-1 中的企業參與者，以及 CIM-2 中的行動負責人，都可能成為 CIM-3 的系統參與者(System Actor)。



圖 20 客戶能啟動的系統 UC

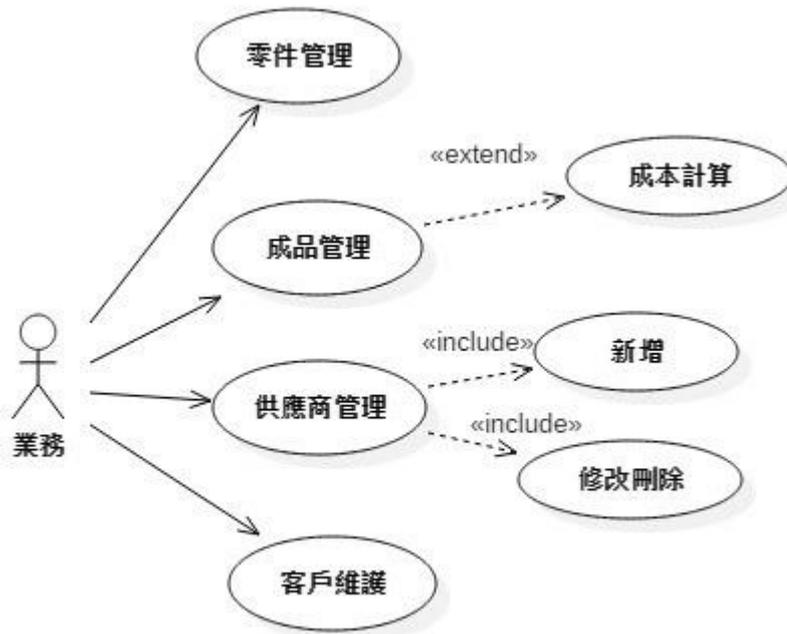


圖 21 業務能啟動的系統 UC

表 5 系統 UC 描述

系統 UC 名稱	描述
零件管理	業務依據實際需求管理進貨零件以及已使用零件。
成品管理	業務依據已組合好之零件管理成品。
成本計算	業務依據成品所使用之靈盡計算成本。
供應商管理	業務依據實際需求進貨管理不同的供應商。
新增	供應商新增。
修改刪除	供應商修改刪除。
客戶資料	新客戶所填之客戶資料。
訂單	客戶依據需求向廠商下訂單。

## 第五章 成果與結論

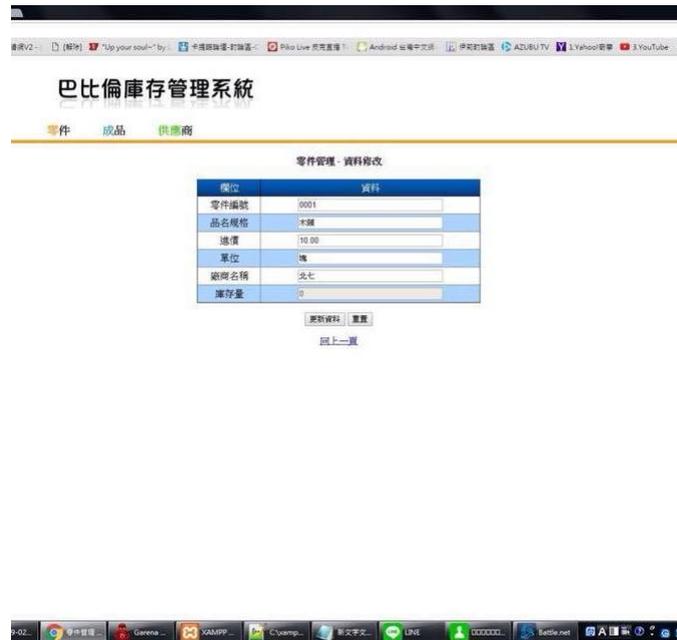


圖 22 系統完成畫面圖\*

本專題為建構零件庫存管理系統，希望藉由這個平台，提供中小型企業使用，讓就算資金不足、人手也不足的中小型企業能夠更佳完善的整理自己的商品庫存，此系統除了零件管理部分，也有增加供應商管理以及成本初算。

網站完成之後我們先做實體測試，在經過完整的美化以及編寫希望將這個網站更加完美甚至能真的提共實際使用。

(1) 本網站完成了三個目標:

- a. 架設零件庫存管理系統
- b. 提供供應商管理資料庫
- c. 提供零售商管理成本

(2) 在未來想繼續改進的目標

- a. 增加系統美化程度
- b. 追蹤清單:已進貨但是未到齊之商品能加入清單種以利搜尋

## 5.1 成果展示



圖 23 供應商管理畫面圖

圖 23 為供應商管理，如畫面上看到的，欄位名稱依序為供應商編號，供應商名稱，聯絡人名字，聯絡人電話與地址，如這些資料有所變更我們也可以做一個修改或者刪除的動作。



圖 24 供應商管理-新增資料圖



圖 25 供應商管理-資料修改圖



圖 26 訂單管理圖

圖 26 為訂單管理的畫面，如畫面上看到的，欄位名稱依序為成品名稱，顧客姓名，數量，如這些資料有所變更，即可以將它刪除。而在畫面最上方，我們有一個可搜尋資料的功能。



圖 27 零件管理圖

圖 27 為零件管理的畫面，如畫面上看到的，欄為名稱依序為，零件編號，零件名稱，進價，單位，供應商名稱，庫存量，而這些資料如有變動，也可以新增，修改或將它刪除，而在畫面最上方，我們有一個可搜尋資料的功能。



圖 28 零件管理-新增資料圖



圖 29 零件管理-資料修改圖



圖 30 零件管理-資料刪除圖



圖 31 零件進貨圖

圖 31 為零件進貨畫面，如畫面上看到的，欄為名稱依序為，零件編號，品名規格及庫存量，在畫面下方可以看到，有進貨日期的選擇，進貨的數量及補充說明，有了這些備註，可以更清楚記錄所有進貨的細節。



圖 32 零件選擇圖



圖 33 成品管理圖

圖 33 為成品管理畫面，如畫面上看到的，欄為名稱依序為，成品編號，成品名稱及成本，而這些資料如有變動，也可以新增，修改或將它刪除。畫面中有一個零件的按鈕，是可以查詢這個成品所需要的零件。而在畫面最上方，我們有一個可搜尋資料的功能。



圖 34 成品管理-新增資料圖



圖 35 成品管理-修改資料圖



圖 36 成品管理-刪除資料圖

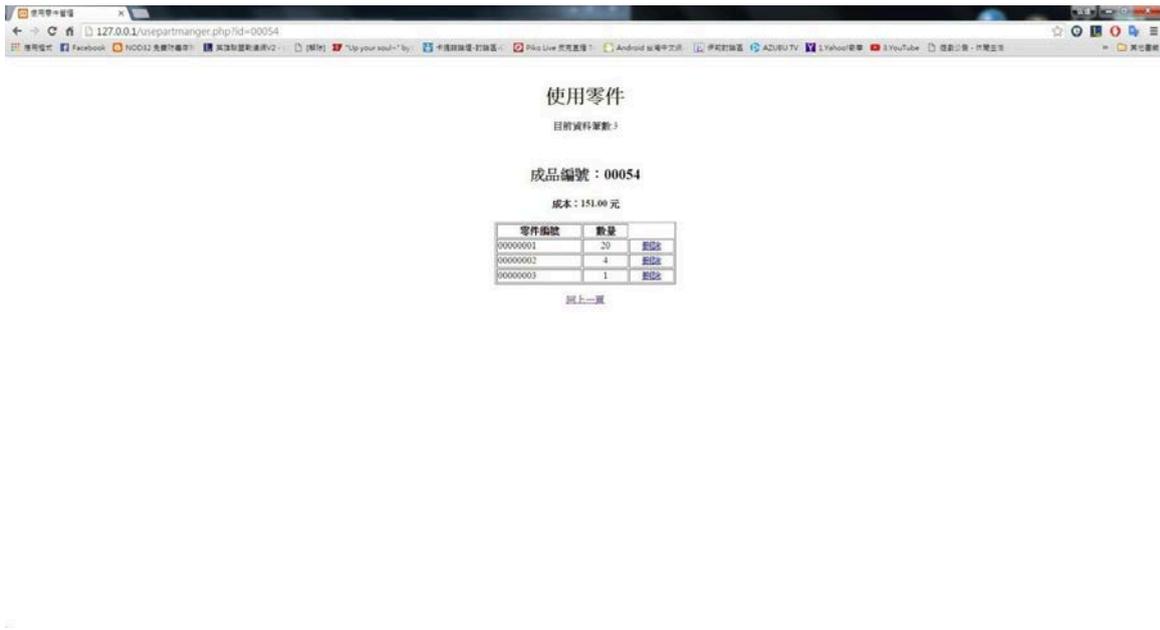


圖 37 使用零件圖

圖 37 為使用零件的畫面，在這個畫面中可以看到有零件編號及數量的欄位，而這個畫面可以依照我們所輸入的資料，幫我們計算出所需要的成本為多少錢，而這些資料如有變更也可以將之刪除。



圖 38 出售圖

圖 38 為出售的畫面，畫面中我們可以看到有兩個表格，上面那個表格裡呈現的是，材料名稱，所需數量及剩餘資料，而下面的表格呈現的是，成品編號，成品名稱，成本這個部分供客戶自行輸入，並填寫完顧客資料後即可按下儲存鍵，這樣上面那個表個就會依照輸入的資料自行變更。如輸入的資料超過我們的庫存量，會通知顧客前往進貨的畫面進行進貨。

## 5.2 結論

以 PHP 語言工具架構網頁，並以參考支援 PHP 開發環境 XAMPP 輔助，在加上 GUI 圖形介面圖案讓使用者更方便理解，相信系統建置完成後，將為企業主帶來極大的便利，提供原零件送貨時間的安全保障，能獲得經濟訂購量的好處，比方說訂購量越大，訂單就越少，就會讓成本越小當運數兩大的時候單位運輸的運費也會越少，降低成本提高利益。

我相信這個系統還有很多可以發展及成長的空間，我們會繼續參考資料以及教授的指導將這個庫存管理系統達到水準，如果後續能與廠商讓我們進一步的更面對這個社會的真實需求，並且利用這個系統能夠達到協助的角色，我相信會是很大的進步。

## 參考文獻

(1) 吳仁和系統分析與設計：理論與實務應用(六版) Systems Analysis and Design: Theory and Application，智勝出版社，民國 102 年

(2) 吳仁和物件導向系統分析與設計：結合 MDA 與 UML

(3) 王翠蘭感性工學應用於 3C 產品線上購物網站使用者介面設計之研究 (4)

UML

<http://nccur.lib.nccu.edu.tw/bitstream/140.119/35187/6/35601906.pdf>

(5) XAMPP

<http://baike.baidu.com/item/XAMPP>

(6) 網路資源- 網路教室

<http://www.w3cschool.cc/>

(7) JSON

<http://www.w3cschool.cc/json/json-tutorial.html>

(8) MongoDB

<http://www.w3cschool.cc/mongodb/mongodb-tutorial.html>

(9) AngularJS

<http://www.w3cschool.cc/angularjs/angularjs-tutorial.html>

(10) 物件導向系統分析與設計教材 - 模型驅動架構 (MDA)

<https://copy.com/wzPosHM0O19EViqq>

(11) Node.js

<http://www.nodejs.org/download/>

(12) 前端使用者介面框架(backend)-官方

<http://www.ngwidgets.com/>

<http://www.ngwidgets.com/angularjs-widgets/>

(13)系統開發平台

<http://mean.io/#/>

(14) jqwidgets-維基百科

[http://speedydeletion.wikiaglobal.com/wiki/Wikipedia\\_talk:Articles\\_for\\_creation/jqwidgets](http://speedydeletion.wikiaglobal.com/wiki/Wikipedia_talk:Articles_for_creation/jqwidgets)

(15) ) GUI Design Studio

<http://www.carettasoftware.com/guidesignstudio/>