

逢 甲 大 學

資 訊 工 程 學 系 專 題 報 告

電 子 學 習 系 統

以逢甲大學資訊工程學系學生選課設計為例



學 生 ：

李	任	修(四丙)
李	建	達(四丙)
方	昱	翔(四丙)

指 導 教 授 ：

謝	信	芳
---	---	---

中 華 民 國 九 十 二 年 十 一 月

摘要

資訊時代的來臨，讓電腦已經普及化居家生活之中，可以說是項民生必需品，而網際網路的大量普遍化也是目前的潮流，現在即使坐在家中也可以靠著網際網路完成大大小小的事情，利用資訊化的作業來節省傳統繁忙的事務，可以說是當今的趨勢了。

電子學習系統更是屬於 E 化校園的一部份，也是政府目前正在準備推行的運動之一，此系統可以拿來建立學生與學校和老師之間良好的溝通管道、互動關係，讓學生們可以在平時就與老師討論功課，妥善的安排規劃自己行程。

系統主要是以微軟 .NET 和物件導向的架構下去開發，再利用學生個人課表，結合學校的行事曆、選課等事務，製作個人專屬的電子學習環境，讓學習在家中就可以輕鬆上網完成。

本系統包含以下功能：

1. 最新消息：掌握學校最新資訊。
2. 行事曆：個人行程安排，妥善規劃時間。
3. 選課及課程查詢：輕鬆方便完成選課，並可查詢課程。
4. 記事本：讓個人可以想記什麼就記什麼。

目 錄

摘要	I
目錄	II
圖目錄	VI
第一章 導論	1
1.1 動機	1
1.2 目的	3
1.2.1 系統方面	3
1.2.2 學生方面	4
1.3 範圍及限制	6
1.2.1 系統方面	6
1.2.2 學生方面	6
第二章 系統概述	7
2.1 電子學習選課系統概述	7
2.1.1 背景	7
2.1.2 甘特圖	18
2.2 選課及課程查詢子系統	19
2.2.1 逢甲大學資訊工程學系課程大綱	19
2.2.2 現有系統架構	48

2.2.3	新系統架構	50
2.2.4	課程查詢子系統需求分析	52
2.2.5	選課子系統需求分析	52
2.2.6	課業子系統需求分析	53
2.2.7	課程資料子系統需求分析	53
2.2.8	學生資料子系統需求分析	54
2.2.9	學校行事曆資料子系統需求分析	54
2.2.10	記事本子系統需求分析	55
2.2.11	教師子系統需求分析	55
2.2.12	登入子系統需求分析	56
第三章	系統分析與設計	57
3.1	系統開發流程模式	57
3.1.1	系統開發流程模式介紹	57
3.2	軟體開發工具分析	60
3.2.1	Web 應用程式	60
3.2.2	資料庫管理系統	63
3.2.3	單元測試框架	64
3.3	硬體設備	64
3.4	系統分析設計工具	65

3.5 系統模型	71
3.6 資料庫設計	77
3.6.1 資料需求	79
3.6.2 實體關係模型-ER Model	81
3.6.3 關聯式資料庫綱要	95
3.6.4 內部綱要	97
第四章 系統建置	102
4.1 登入子系統	102
4.2 選課子系統、課程查詢子系統	104
4.3 學生資料子系統	105
4.4 行事曆子系統	106
4.5 記事本子系統	107
第五章 系統評估	108
5.1 作正確的事	108
5.2 需加強的事	108
5.3 行事曆系統	108
5.4 記事本系統	108
5.5 登入子系統	109
5.6 教師子系統	109

5.7 作正確的事.....	109
5.8 需加強的事.....	110
第六章 心得報告.....	111
6.1 任修.....	111
6.2 建達.....	112
6.3 昱翔.....	113
參考資料.....	114



圖目錄

圖 2.1、電腦系統學程課程流程圖.....	11
圖 2.2、軟體工程學程課程流程圖.....	12
圖 2.3、3C 學程課程流程圖.....	13
圖 2.4、網際網路工程學程課程流程圖.....	14
圖 2.5、甘特圖.....	18
圖 2.6、選課系統使用登入圖.....	48
圖 2.7、課務查詢系統使用圖.....	49
圖 2.8、全景圖.....	50
圖 2.9、圖 0.....	51
圖 2.10、課程查詢子系統需求.....	52
圖 2.11、選課子系統需求.....	52
圖 2.12、課業子系統需求.....	53
圖 2.13、課程資料子系統需求.....	53
圖 2.14、學生資料子系統需求.....	54

圖 2.15、學校行事曆子系統需求	54
圖 2.16、記事本子系統需求	55
圖 2.17、教師子系統需求	55
圖 2.18、登入子系統需求	56
圖 3.1、課程資料庫-學生實體圖	87
圖 3.2、課程資料庫-教師實體圖	88
圖 3.3、課程資料庫-教師-學院	89
圖 3.4、課程資料庫-科目-課程別	90
圖 3.5、課程資料庫-科目-教室別	91
圖 3.6、課程資料庫-學生-課程	92
圖 3.7、課程資料庫-教師-科目	93
圖 3.8、課程資料庫-學生-科目 2	94
圖 3.9、學生內部資料表	97
圖 3.10、教師內部資料表	97
圖 3.11、科目資料表	98
圖 3.12、班級資料表	98
圖 3.13、教室別資料表	99
圖 3.14、書籍資料表	99
圖 3.16、課程資料表	99

圖 3.17、學院資料表	99
圖 3.18、教室資料表	99
圖 3.19、大樓資料表	99
圖 3.20、學生修習科目資料表	100
圖 3.21、學生期中考監考科目資料表	100
圖 3.22、學生期末考監考科目資料表	100
圖 3.23、學生畢業提前考監考科目資料表	100
圖 3.24、學生授課科目資料表	100
圖 3.25、教師期中考監考科目資料表	100
圖 3.26、教師期末考監考科目資料表	100
圖 3.27、教師畢業提前考監考科目資料表	101
圖 3.28、教師授課科目資料表	101
圖 3.29、班級開課科目資料表	101
圖 3.30、科目指定教科書資料表	101
圖 3.31、科目上課教室資料表	101
圖 3.32、教師請益時間資料表	101
圖 3.33、科目每週上課內容資料表	101
圖 4.1 登入子系統	102
圖 4.2 登入失敗	103

圖 4.3 主畫面	104
圖 4.4 個人設定	105
圖 4.5 行事曆	106
圖 4.6 記事本	107



第一章 導論

1.1 動機

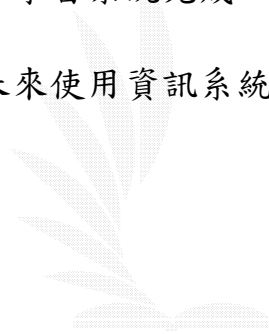
學校的選課系統，相信大家都曾經歷過，也曾感嘆過，許多的科目是應該選到的卻選不到，也曾懷疑到點選的科目到底是否合乎自己的興趣？所點選的老師上課方式如何？等諸多疑問吧！

在使用過學校的選課系統後，會發覺到學校所給予的資訊並不足夠，就以資訊工程學系本科目種類來說，所選到的科目名稱對某些學弟妹來講，太過於陌生、抽象，並沒有足夠的資訊可以讓學弟妹們先去探查、了解那門科目，以致於總是看著科目名稱猜想該科目的上課內容，說不定卻選上了自己並不喜歡的科目，卻還要忙著去退選或勉為其難的強迫自己接受都是件很麻煩的事情，雖然本系已分成四個學程，但有些科目並不在學程的領域中，或學程中並沒有有的科目而有學弟妹對那科目有興趣想去修習，都沒有充分的資料顯示；鑒此本組成員以電子學習系統作為專題，針對逢甲大學資訊系大學生選課系統，提供課程所需要的所有資料，藉以增進資訊系學生對選課系統發揮到最大的利用。

因為電子學習系統，目前僅在初步萌芽的階段，在逢甲校園內，有如火如荼般地盛行，身為資訊工程學系的我們當然不能落後於他系之後，因此更需要一個完善的電子學習系統，針對目前的選

課系統做搭配，並且可擴充到多媒體教學，課後練習，將來也可造福以後的學弟妹們。

本組認為，選課系統對於資訊工程學系來說，可以說是非常親密的一項資訊系統，而其功能卻使得許多資訊工程學系學生無法成功的選取到其所喜愛的科目，因此建立一個電子學習系統來改善加強選課系統，是回饋於資訊工程學系無數學生的好方法，並且由於其中牽涉到龐大的資料庫，對於我們來說更是興趣所驅使，因此更會努力的將這個電子學習系統完成，也是希望藉此機會讓大家了解電子學習系統將是未來使用資訊系統的新潮流。



1.2 目的

學校的選課系統基本上可以達成選課目的，但是並不能從選課系統內知道 what you need，因此想以該系統為基石作出如下功能的擴充：

1.2.1 系統方面

a. 改善功能

以往的選課系統是使用四、五年以前，Sybase 公司所出產的 power builder 中的 Web.PB 當作主程式，只要同時上線人數達到一定人數，就會產生不穩、執行緩慢、當機或是不准其他使用者上線等情況發生，因此現在使用新的軟體且採用分散式系統架構開發，可將主要運算移到其他主機，減輕伺服器的負擔，讓系統可以正常運作。

b. 提高效率

採用 windows 最新 sever 技術，由於 windows 新的 sever 增加了許多新的功能來提供使用者使用其程式時的效率，因此可以增進資料庫程式等使用時效率。

1.2.2 學生方面：

a. 擴充選課資訊

由於學校的選課就好像強迫學生修習某些指定課程，且事先無法了解其內容、資訊等，因此本系統一定要擴充選課資訊。

b. 提供教師個人資訊

由於選課系統並無法提供擔任此科目之教師之機關資訊，因此將會提供教導此科目之教師個人資訊包括教導此科目之方式等。

c. 提供上課內容及教科書說明

選課之前當然要熟悉此科目之上課內容說明，以及各學程所規定，並且附上教師使用哪本教科書，及其大綱等使學生在選課之時不致混亂。

d. 提供上課講義、投影片

將老師在其課堂上所需要的投影片、講義等收錄於網路上，方便學生在課餘時間即可上網觀看，不需多花錢影印講義，省錢也可環保。而老師則可以在自己所希望的時間，上傳、更改自己上課所需要用的投影片。

e. 方便學生繳交作業系統

可讓學生事先繳交選課前的作業，讓自己了解是否適合此科目並且了解是不是對此科目有興趣。

f. 提供線上模擬考試

線上模擬考試可以為學生事先測試是否有符合此科目的興趣以及標準，也可讓日後老師在課堂之後作為測驗學生的一個好工具。

g. 提供圖形化介面

由於目前的選課系統採用文字介面，讓學生若想選其他班課程卻不能容易知道有沒有衝堂課等，因此採用圖形化介面，讓選課時課程時間容易了解何時有空堂。

h. 提供學生個人記事本、教學成果評估等附屬功能

方便學生做上課筆記，及投票哪些老師較好，以及記錄行事曆等。

1.3 範圍及限制

1.3.1 範圍

本系統針對逢甲大學資訊工程系一、二年級各班學生所使用，且針對資訊工程學系一、二年級系上所開之課程為主，並不包含通識、體育課程等非系上必選修課程，但是系上所開選修課程則班級與班級間可以互換；只要是符合此條件者，均有資格使用本系統，包括轉學生、及重修生人等。

1.3.2 限制

- a. 總學分數不可以超過系上所規定 25 學分且每學期不得少於十二學分。若為超修者則暫時不在本系統中所接受。
- b. 凡本班所開必修科目，經系主任核准外不得在他班修習。
- c. 學生不得在同一時間內修習兩個(含)以上之科目，衝堂之科目均應予以退選。
- d. 若未達成選課前所需規定修習過的科目，則不能修習其後科目。修課人數不得超過系上規定人數。
- e. 每個進入此系統學生必使用學校之學號，並且密碼不得過於簡單，若發現學號不符合系統所規定之範圍內，則需另外申請。
- f. 學生若要加選，則必先填寫系統加選單給加選科目老師。

第二章 系統概述

2.1 電子學習選課系統概述

2.1.1 背景

a. 資訊系背景

我國資訊工業，在世界科技潮流的影響及政府的大力推展之下，日益精進。資訊化與自動化的科技下可以提高生產力，促進工業升級，進而提昇國力。而本系所成立之主要目的 在培育資訊工程人才，以應社會對人才之需求，並研究開發軟體及硬體工程技術，以提高資訊工業之結構層次。同時利用本系所設備，為國內工商業界及政府機構提供服務，建立學術與工商業界之橋樑。

民國五十八年(西元 1969 年)本校成立電子計算機科學系。民國七十一年(西元 1982 年)後更名為資訊工程學系。民國七十三年(西元 1984 年)成立碩士班。民國八十二年(西元 1993)成立博士班。民國八十八年(西元 1999 年)成立碩士在職專班。畢業校友在國內外嶄露頭角，表現極為優異。

研究領域及發展方向，著重於高速運算、計算機結構、平行分散式處理、軟品工程、計算機網路、影像處理及多媒體處理等，其所開課程和師資聘請亦針對此目的進行，以造就高級資訊工程科技

研究人才，並培養大專院校資訊工程師資格，期望能配合我國資訊工業發展，和對國家經濟建設之發展能有更積極之貢獻。

(1) 系上班級、人數

資訊一甲 53 人

資訊一乙 59 人

資訊一丙 58 人

總共 180 人

資訊二甲 62 人

資訊二乙 61 人

資訊二丙 56 人

總共 179 人

(2) 課程

大學部的課程乃以奠定學生在計算機科學與資訊工程等方面為基礎。課程內容編排上，本系以美國電子電機工程學會以及計算機協會建議的課程為藍本，同時參考國外各大學的專業課程計劃，並配合國內的實際情形訂定一套較具彈性的課程規劃。除共同必修的通才教育科目外，本系設有「電腦系統」、「軟體工程」、「3C」以及「網際網路工程」等四個學程。

其中「資訊專題製作」是本系訓練學生實際動手的重要特色。學生於大二下學期選擇指導老師，擇定資訊相關的專題題目親自參與製作撰寫專題論文，最後經由三位教授的口試評審認可後才能通過。學生經此專題訓練後，不管是再繼續深造或就業，都能獲得各機關單位與研究所的認同與讚賞。本系榮獲教育部八十九年度通訊科技教育改進計畫 86-89 四年執行成效優等獎，就是最佳證明。

未來，更將繼續延攬優良師資，增設專題實驗室加強各重點領域，以帶動整合性大型研究計劃，一方面加強教學品質和積極爭取公民營機構的研究計劃，另一方面也積極參與和舉辦各種專業和學術研討會，提昇本系所在資訊工程領域學術上及專業上的水準。

(3) 學程目標

a. 電腦系統學程

以電腦系統的設計與整合為出發點，提供系統所需軟硬體的基本認識和理論學程，並為系統整合設計等設有進階的專業課程，以電腦為主體的各種應用的技術和觀念的培養等。期使修習本學程的學生能對於電腦系統的整合與設計具有專業素養，熟悉電腦系統的相關理論與應用技術外，更能具備獨立思考解決問題的能力。

電腦系統學程流程圖請參閱圖 2.1。

b. 軟體工程學程

資訊科技的核心仍然是由電腦軟體來控制，本學程提供基礎的軟體技術與理論課程，以及發展快速、高效率、即時性、多元性等大型軟體系統的課程。專業課程方面強調「電腦系統」軟體與「應用系統」軟體的研究與發展為範疇，提供學生修習本領域的專業知識與實務，以培養學生成為具備軟體工程專長的資訊人才。

軟體工程學程流程圖請參閱圖 2.2。

c. 3C 學程

未來 3C 整合產品的應用概念，其架構大致可分為使用者資訊家電，通訊網路架構及資訊應用服務三大群組。本學程便是依據這三大群組為主軸強調 3C 工程領域的應用性研究，主要學習方向包括電腦、通訊及電子基礎課程，微處理機單晶片系統、網際網路、積體電路設計等。

3C 學程流程圖請參閱圖 2.3。

d. 網際網路工程學程

在網際網路應用方面，主要將上述課程和資訊學理，推廣到實務應用，課程方面則提供資料庫系統、多媒體整合製作、虛擬實境、電子商務等課程。另外，為了加強學生在網際網路方面的實作能力，本學程設有網際網路工程專題課程。

網際網路工程學程流程圖請參閱圖 2.4。

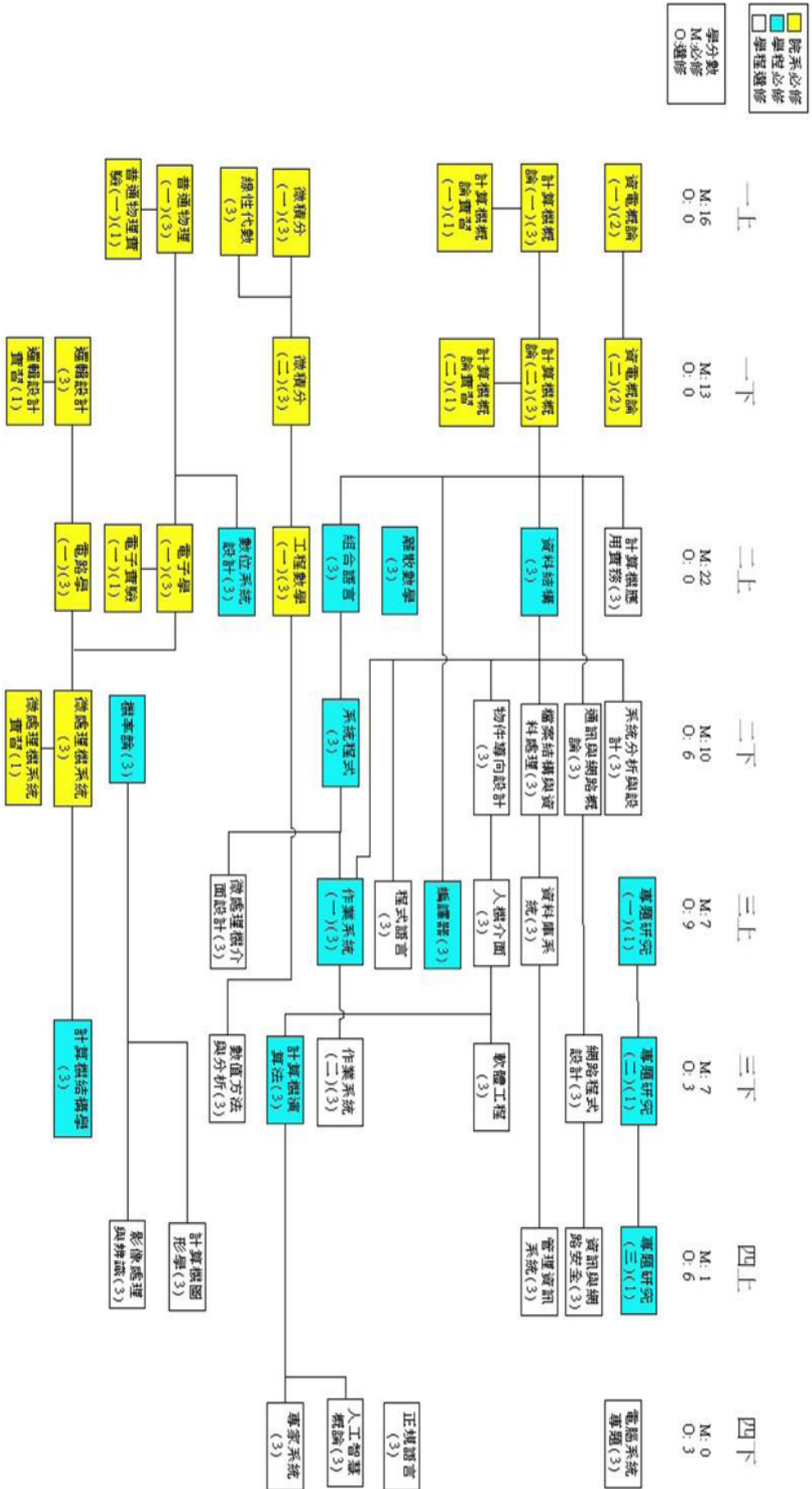
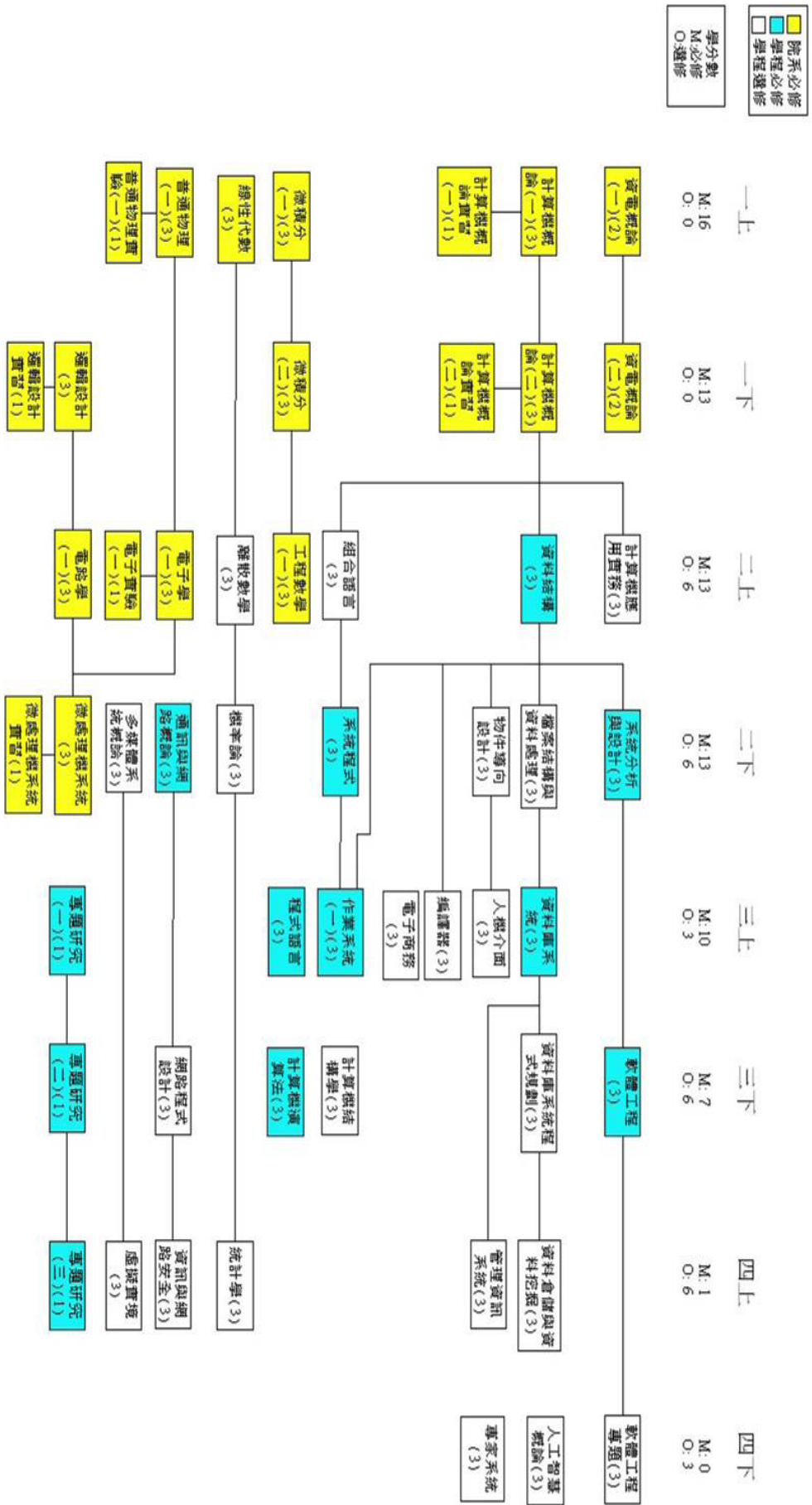
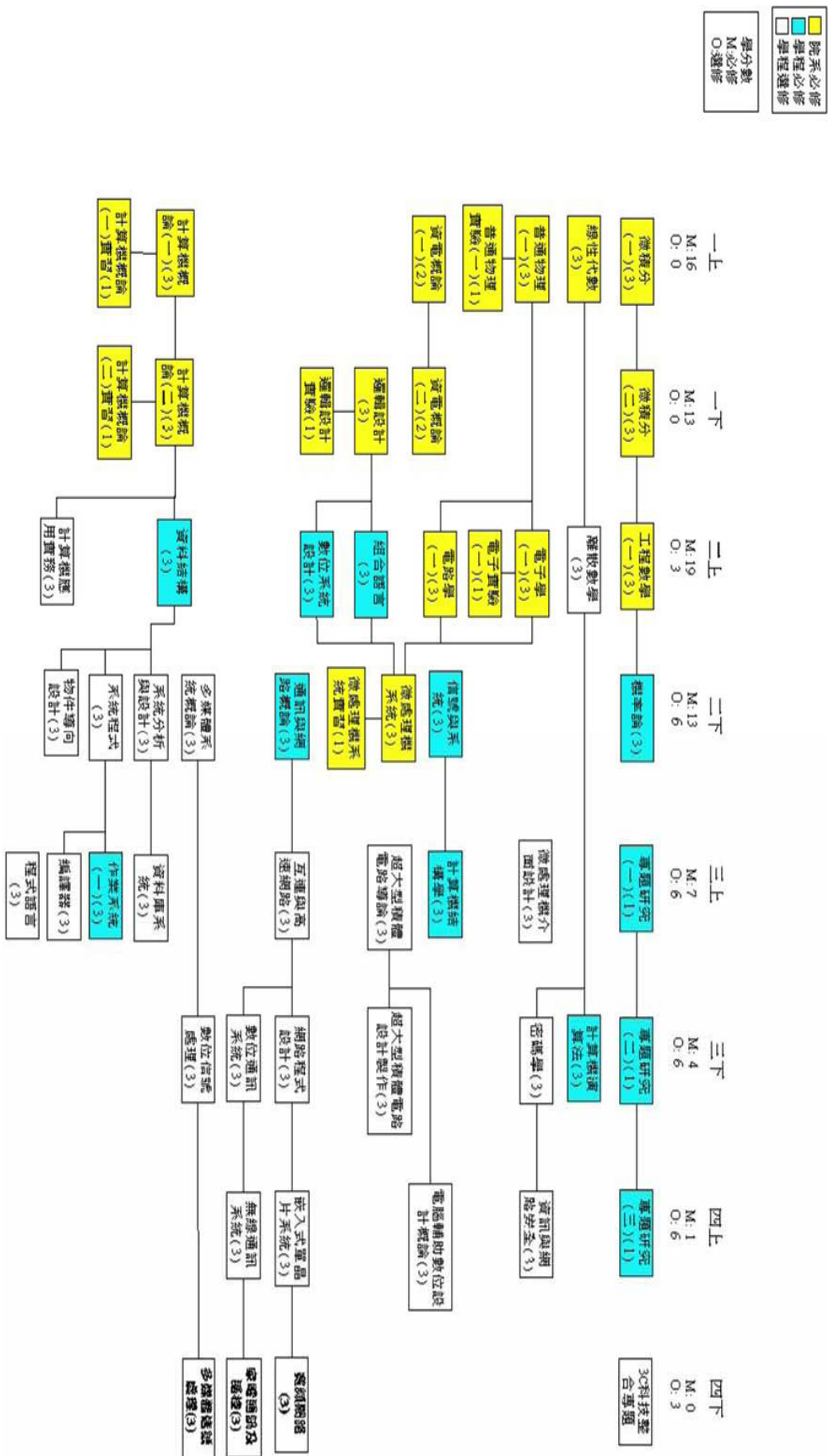


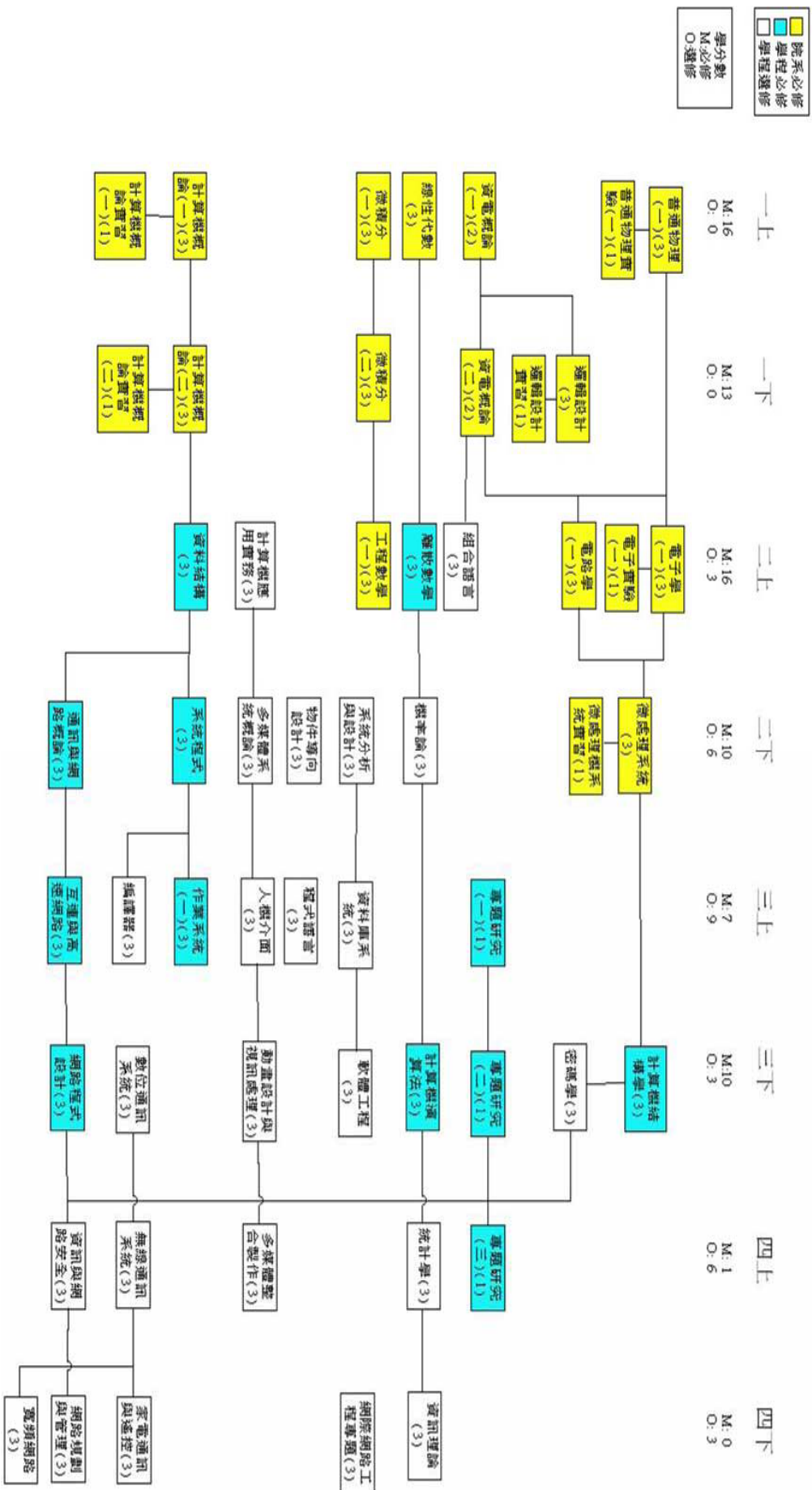
圖 2.1、電腦系統學課程流程圖



附圖 2.2、軟體工程學課程流程圖



附圖 2.3、3C 課程課程流程圖



附圖 2.4、網際網路工程學課程流程圖

(4)選課規定依據—逢甲大學學生選課辦法

- i 本辦法依據本校學則訂定之。
- ii 本校學生辦理選課時，應按本辦法及各項選課須知規定辦理。
- iii 加退選規定如下：
 - (a). 學生得於加退選期間，依規定辦理加退選，逾期不得辦理加選，齊已經選定之科目而自行輟修者，該科目以零分計。
 - (b). 必修及選修科目於加退選截止後至期中考試開始一週前，經任課教師同意得申請退選，退選後修習學分總數不得低於規定之最低標準。
- iv 凡本班所開必修科目，除經系主任核准外不得在他班修習。
- v 學生修習非本系開受之科目學分，得經本系主任核定後列入最低畢業學分數內。
- vi 學生不得在同一時間內修習兩個(含)以上之科目，衝堂之科目均應予以退選。
- vii 應屆畢業學生得經系所主管及任課教師核准修習研究所碩士班科目，其學分的計入最低畢業學分數內。
- viii 學生加退選後之修習學分總數，大學部及進修學士班每學期部的多於廿五學分，但情況特殊者，得經系主任核可超

修：大學部一至三年級(含建築系四年級)每學期不得少於十二學分，第四學年(含建築系五年級)每學期不得少於九學分，進修學士班每學期至少應選修一個科目。

- ix 學生學期學業名次在該系該班學生前百分之二十者，於次學期經系主任核可得超修至多四學分，又每學期名次在該系該班學生數前百分之五以內者，次學期得經系主任核可加修一至二科目，並得修習高年級必修課程。

以上兩者加修後之修習學分數不受每學年最高修習學分數之限制。

- x 凡全學年之科目，其第一學期成績達四十分者，第二學期仍可續修，各學期修習及格之學分分別予以核計。

- xi 學生不按規定選課修習者均應於退選，如因作業錯誤者，得經任課教師同意及系主任核可後准予改選。

- xii 大學部及進修學士班得經各系主任核准後互選課程，最多以七學分為限，但情形特殊者，進修學士班學生得經系主任核准後超修(應包含在每學期限修學分數之內)。

- xiii 各科目每班修習人數最少不得低於開班規定：學士班該系該年級若為單班至少十人，若為雙班至少二十人，若為三或四班至少三十人，共同科目至少二十五人，通識教育核心課程至少四十人；碩士班各科目每班至少五人；博士班

各科目至少三人。

xiv 學生重補修必修科目與修習新舊課程處理規定如下：

(a). 本系該科目原為必修，爾後年度入學新生實施之科目改為

選修或停開，則仍應修必該科目或經系主任核可改修本系

或他系內容相近之科目使得畢業，若無內容相近之科目，

經系主任核可得以免修。

(b). 重補修必修科目學分數較原科目學分數為多時，其所超修

之學分計入最低畢業學分數內。

(c). 重補修必修科目學分數較原科目學分數為少時：

I. 原為一學年課程改為一學期課程，其原已修習及格之

上或下學期學分仍計入最低畢業學分數內。

II. 新舊均為一學年課程，但學分數減少，修習及格後視

為補修完成。

(d). 原為兩種科目合併為一個科目，學生修習其中一科目不

及格，均需修習新開之科目學分，其原已修習及格之科目

學分亦計入最低畢業學分數內。

xv 本辦法經教務會議通過，簽請 校長核定後實施，修正時亦

同。

2.1.2 工作分配與時程

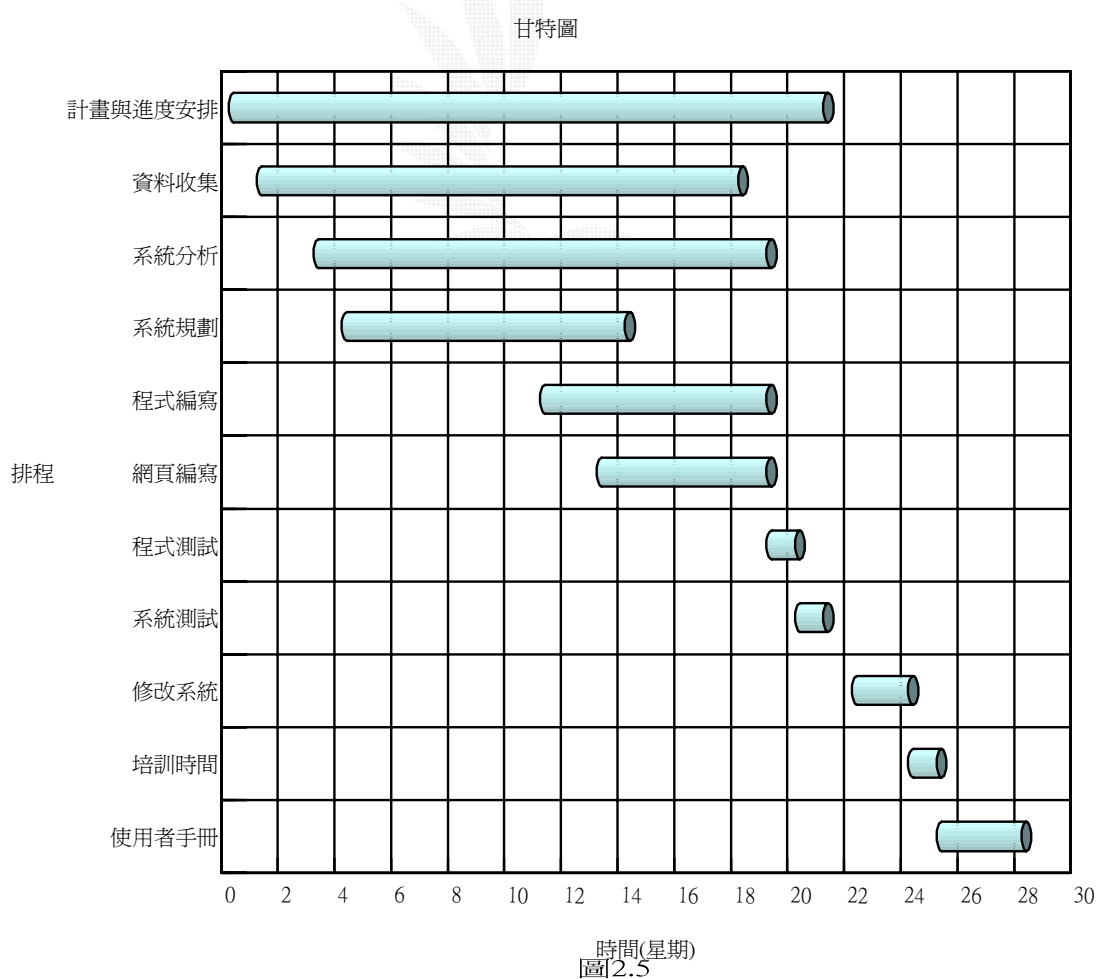
1. 工作分配

李任修：負責伺服器架設、維護、資料庫程式編寫。

李建達：應用程式開發、編寫。

方昱翔：網頁程式編寫、系統測試。

2. 工作時程



2.2 選課及課程查詢子系統

2.2.1 逢甲大學資訊工程學系課程大綱

(1). 九十學年度學程課程科目表：

資訊電機學院 資訊工程 學系 電腦系統 學程表

第一年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	國文(一)	2			
	英文(一)	3			
	英文聽講訓練(一)	0			
	史學概論	2			
院共同 必修	微積分(一)	3			
	普通物理(一)	3			
	普通物理實驗(一)	1			
	計算機概論(一)	3			
	計算機概論實習(一)	1			
	資電概論(一)	2			
	線性代數	3			
	小計	23			

第一年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	國文(二)	2			
	英文(二)	3			
	英文聽講訓練(二)	0			
	歷史通識課程	2			
院共同 必修	微積分(二)	3			
	計算機概論(二)	3			
	計算機概論實習(二)	1			
	資電概論(二)	2			
	邏輯設計	3			
	邏輯設計實習	1			
	小計	20			

備註*：其餘校共同必修(共七學門課程)於二~四年級內修畢即可。

第二年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修*	校共同必修(1)：	2			
院共同 必修	電子學實驗(一)	1			
	工程數學(一)	3			
	電子學(一)	3			
	電路學(一)	3			
學程 必修 課程	資料結構	3			
	組合語言	3			
	數位系統設計	3			
	離散數學	3			
學程 選修 課程 **					
	小計 (必修 24 選修 0)	24			

*校共同必修在下列課程任選一門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1)計算機應用實務[3]

第二年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同必修*	校共同必修(1):	2			
院共同必修	微處理機系統	3			
	微處理機系統實習	1			
學程必修課程	系統程式	3			
	機率論	3			
學程選修課程**	學程選修(1):	3			
	學程選修(2):	3			
	小計 (必修 12 選修 6)	18			

*校共同必修在下列課程任選一門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1)通訊與網路概論[3]，(2)系統分析與設計[3]，(3)物件導向設計[3]，(4)檔案結構與資料處理[3]

第三年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
學程 必修 課程	作業系統(一)	3			
	編譯器	3			
	專題研究(一)	1			
學程 選修 課程 **	學程選修(1)：	3			
	學程選修(2)：	3			
	學程選修(3)：	3			
	小計 (必修 9 選修 9)	18			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

- (1) 資料庫系統[3]，(2)人機介面[3]，(3)微處理機介面設計[3]，(4)程式語言[3]

第三年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
學程 必修 課程	計算機演算法	3			
	計算機結構	3			
	專題研究(二)	1			
學程 選修 課程 **	學程選修(1)：	3			
	小計 (必修 9 選修 3)	12			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1)網路程式設計[3]，(2)軟體工程[3]，(3)作業系統(二)[3]，(4)數值方法與分析[3]

第四年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
	校共同必修(2)：	2			
學程 必修 課程	專題研究(三)	1			
學程 選修 課程 **	學程選修(1)：	3			
	學程選修(2)：	3			
	小計 (必修 5 選修 6)	11			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1) 資訊與網路安全[3]，(2) 管理資訊系統[3]，(3) 影像處理與辨識[3]，(4) 計算機圖形學[3]

第四年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
學程 必修 課程					
學程 選修 課程 **	學程選修(1)：	3			
	小計 (必修 2 選修 3)	5			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1) 電腦系統專題[3]，(2) 正規語言[3]，(3) 專家系統[3]，(4) 人工智慧概論[3]

資訊電機學院 資訊工程 系 軟體工程 學程
 第一年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	國文(一)	2			
	英文(一)	3			
	英文聽講訓練(一)	0			
	史學概論	2			
院共同 必修	微積分(一)	3			
	普通物理(一)	3			
	普通物理實驗(一)	1			
	計算機概論(一)	3			
	計算機概論實習(一)	1			
	資電概論(一)	2			
	線性代數	3			
	小計	23			

第一年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	國文(二)	2			
	英文(二)	3			
	英文聽講訓練(二)	0			
	通識歷史課程	2			
院共同 必修	微積分(二)	3			
	計算機概論(二)	3			
	計算機概論實習(二)	1			
	資電概論(二)	2			
	邏輯設計	3			
	邏輯設計實習	1			
	小計	20			

*備註：其餘校共同必修(共七學門課程)於二~四年級內修畢即可。

第二年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)	2			
院共同 必修	電子學實驗(一)	1			
	工程數學(一)	3			
	電子學(一)	3			
	電路學(一)	3			
學程 必修 課程	資料結構	3			
學程 選修 課程 **	學程選修(1)	3			
	學程選修(2)	3			
	小計 (必修 15 選修 6)	21			

*校共同必修在下列課程任選一門:

- (1)藝術美學相關課程[2]，(2)倫理宗教哲學相關課程[2]，(3)社會心理教育相關課程[2]，
(4)法律與政治相關課程[2]，(5)商管經濟相關課程[2]，(6)中華民國憲法[2]，
(7)國學相關課程[2].

**學程選修課程:

- (1)計算機應用實務[3]，(2)組合語言[3]，離散數學[3].

第二年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)	2			
院共同 必修	微處理機系統	3			
	微處理機系統實習	1			
學程 必修 課程	系統程式	3			
	系統分析與設計	3			
	通訊與網路概論	3			
學程 選修 課程 **	學程選修(1)	3			
	學程選修(2)	3			
	小計 (必修 15 選修 6)	21			

*校共同必修在下列課程任選一門：

- (1)藝術美學相關課程[2]，(2)倫理宗教哲學相關課程[2]，(3)社會心理教育相關課程[2]，
 (4)法律與政治相關課程[2]，(5)商管經濟相關課程[2]，(6)中華民國憲法[2]，
 (7)國學相關課程[2].

**學程選修課程：

- (1)多媒體系統概論[3]，(2)檔案結構與資料處理[3]，(3)物件導向設計[3]，(4)機率論[3].

第三年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)	2			
學程 必修 課程	作業系統(一)	3			
	資料庫系統	3			
	專題研究(一)	1			
	程式語言	3			
學程 選修 課程 **	學程選修(1)	3			
	小計 (必修 12 選修 3)	15			

*校共同必修在下列課程任選一門:

- (1)藝術美學相關課程[2]， (2)倫理宗教哲學相關課程[2]， (3)社會心理教育相關課程[2]，
 (4)法律與政治相關課程[2]， (5)商管經濟相關課程[2]， (6)中華民國憲法[2]，
 (7)國學相關課程[2].

**學程選修課程:

- (1)編譯器[3]， (2)人機介面[3]， (3)電子商務[3].

第三年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)	2			
學程 必修 課程	計算機演算法	3			
	軟體工程	3			
	專題研究(二)	1			
學程 選修 課程 **	學程選修(1)	3			
	學程選修(2)	3			
	小計 (必修 9 選修 6)	15			

*校共同必修在下列課程任選一門:

- (4) (4) 藝術美學相關課程[2]， (2)倫理宗教哲學相關課程[2]， (3)社會心理教育相關課程[2]，
 (4)法律與政治相關課程[2]， (5)商管經濟相關課程[2]， (6)中華民國憲法[2]，
 (7)國學相關課程[2].

**學程選修課程:

- (1)資料庫系統程式規劃[3]， (2)網路程式設計[3]， (3)計算機結構學[3].

第四年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)	2			
	校共同必修(2)	2			
學程 必修 課程	專題研究(三)	1			
學程 選修 課程 **	學程選修(1)	3			
	學程選修(2)	3			
	小計 (必修 5 選修 6)	11			

*校共同必修在下列課程任選兩門:

- (1)藝術美學相關課程[2]， (2)倫理宗教哲學相關課程[2]， (3)社會心理教育相關課程[2]，
 (4)法律與政治相關課程[2]， (5)商管經濟相關課程[2]， (6)中華民國憲法[2]，
 (7)國學相關課程[2].

**學程選修課程:

- (1)資料倉儲與資料挖掘[3]， (2)管理資訊系統[3]， (3)統計學[3]， (4)資訊與網路安全[3]，
 (5)虛擬實境[3].

第四年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)	2			
學程 必修 課程					
學程 選修 課程 **	學程選修(1)	3			
	小計 (必修 2 選修 3)	5			

*校共同必修在下列課程任選一門：

- (1)藝術美學相關課程[2]，(2)倫理宗教哲學相關課程[2]，(3)社會心理教育相關課程[2]，
 (4)法律與政治相關課程[2]，(5)商管經濟相關課程[2]，(6)中華民國憲法[2]，
 (7)國學相關課程[2].

**學程選修課程：

- (1) (1)軟體工程專題[3]，(2)人工智慧概論[3]，(3)專家系統[3].

總計：

校共同必修	28
院共同必修	43
學程必修	30
<hr/>	
合計必修	101
選修	30
<hr/>	
總計必選修	131

資訊電機學院 資訊工程 系 3C 學程

第一年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	國文(一)	2			
	英文(一)	3			
	英文聽講訓練(一)	0			
	史學概論	2			
院共同 必修	微積分(一)	3			
	普通物理(一)	3			
	普通物理實驗(一)	1			
	計算機概論(一)	3			
	計算機概論實習(一)	1			
	資電概論(一)	2			
	線性代數	3			
	小計	23			

第一年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	國文(二)	2			
	英文(二)	3			
	英文聽講訓練(二)	0			
	歷史通識課程	2			
院共同 必修	微積分(二)	3			
	計算機概論(二)	3			
	計算機概論實習(二)	1			
	資電概論(二)	2			
	邏輯設計	3			
	邏輯設計實習	1			
	小計	20			

備註*：其餘校共同必修(共七學門課程)於二~四年級內修畢即可。

第二年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
院共同 必修	電子實驗(一)	1			
	工程數學(一)	3			
	電子學(一)	3			
	電路學(一)	3			
學程 必修 課程	資料結構	3			
	組合語言	3			
	數位系統設計	3			
學程 選修 課程 **	學程選修(1)：	3			
	小計 (必修 21 選修 3)	24			

*校共同必修在下列課程任選一門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1)計算機應用實務[3]，(2)離散數學[3]

第二年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
院共同 必修	微處理機系統	3			
	微處理機系統實習	1			
學程 必修 課程	通訊與網路概論	3			
	機率論	3			
	訊號與系統	3			
學程 選修 課程 **	學程選修(1)：	3			
	學程選修(2)：	3			
	小計 (必修 15 選修 6)	21			

*校共同必修在下列課程任選一門：

- (1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，
(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

- (1)系統程式[3]，(2)系統分析與設計[3]，(3)物件導向設計[3]，(4)多媒體系統概論[3]

第三年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
學程 必修 課程	作業系統(一)	3			
	計算機結構學	3			
	專題研究(一)	1			
學程 選修 課程 **	學程選修(1)：	3			
	學程選修(2)：	3			
	小計 (必修 9 選修 6)	15			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1) 程式語言[3]，(2)互連與高速網路[3]，(3)編譯器[3]，(4)超大型積體電路設計導論[3]，(5)資料庫系統[3]，(6)微處理機介面設計[3]

第三年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
學程 必修 課程	計算機演算法	3			
	專題研究(二)	1			
學程 選修 課程 **	學程選修(1)：	3			
	學程選修(2)：	3			
	小計 (必修 6 選修 6)	12			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1) 網路程式設計[3]，(2) 超大型積體電路設計製作[3]，(3) 密碼學[3]，
(4) 數位信號處理[3]，(5) 數位通訊系統[3]

第四年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
	校共同必修(2)：	2			
學程 必修 課程	專題研究(三)	1			
學程 選修 課程 **	學程選修(1)：	3			
	學程選修(2)：	3			
	小計 (必修 5 選修 6)	11			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1) 資訊與網路安全[3]，(2)電腦輔助數位設計概論[3]，(3)嵌入式單晶片系統[3]，(4)無線通訊系統[3]

第四年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
學程 必修 課程					
學程 選修 課程 **	學程選修(1)：	3			
	小計 (必修 2 選修 3)	5			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1) 3C 科技整合專題[3]，(2) 多媒體訊號處理[3]，(3) 寬頻網路[3]，(4) 家電通訊及遙控[3]

資訊電機學院 資訊工程 系 網際網路工程 學程
 第一年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	國文(一)	2			
	英文(一)	3			
	英文聽講訓練(一)	0			
	史學概論	2			
院共同 必修	微積分(一)	3			
	普通物理(一)	3			
	普通物理實驗(一)	1			
	計算機概論(一)	3			
	計算機概論實習(一)	1			
	資電概論(一)	2			
	線性代數	3			
	小計	23			

第一年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	國文(二)	2			
	英文(二)	3			
	英文聽講訓練(二)	0			
	歷史通識課程	2			
院共同 必修	微積分(二)	3			
	計算機概論(二)	3			
	計算機概論實習(二)	1			
	資電概論(二)	2			
	邏輯設計	3			
	邏輯設計實習	1			
	小計	20			

備註*：其餘校共同必修(共七學門課程)於二~四年級內修畢即可。

第二年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
院共同 必修	電子實驗(一)	1			
	工程數學	3			
	電子學	3			
	電路學	3			
學程 必修 課程	資料結構	3			
	離散數學	3			
學程 選修 課程 **	學程選修(1)：	3			
	小計 (必修 18 選修 3)	21			

*校共同必修在下列課程任選一門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1)計算機應用實務[3]，(2)組合語言[3]，

第二年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
院共同 必修	微處理機系統	3			
	微處理機系統實習	1			
學程 必修 課程	系統程式	3			
	通訊與網路概論	3			
學程 選修 課程 **	學程選修(1)：	3			
	學程選修(2)：	3			
	小計 (必修 12 選修 6)	18			

*校共同必修在下列課程任選一門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1)多媒體系統概論[3]，(2)系統分析與設計[3]，(3)物件導向設計[3]，(4)機率論[3]

第三年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
學程 必修 課程	作業系統(一)	3			
	專題研究(一)	1			
	互連與高速網路	3			
學程 選修 課程 **	學程選修(1)：	3			
	學程選修(2)：	3			
	學程選修(3)：	3			
	小計 (必修 9 選修 9)	18			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1) 資料庫系統[3]，(2)人機介面[3]，(3)編譯器[3]，(4)程式語言[3]

第三年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
學程 必修 課程	計算機演算法	3			
	計算機結構學	3			
	網路程式設計	3			
	專題研究(二)	1			
學程 選修 課程 **	學程選修(1)：	3			
	小計 (必修 12 選修 3)	15			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1)軟體工程[3]，(2)動畫設計與與視訊處理[3]，(3)密碼學[3]，(4)數位通訊系統[3]

第四年上學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
	校共同必修(2)：	2			
學程 必修 課程	專題研究(三)	1			
學程 選修 課程 **	學程選修(1)：	3			
	學程選修(2)：	3			
	小計 (必修 6 選修 6)	12			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1) 資訊與網路安全[3]，(2)統計學[3]，(3)多媒體整合製作[3]，(4)無線通訊系統[3]

第四年下學期

	課程名稱	學分數	修習學期	成績	備註
校共同 必修 *	校共同必修(1)：	2			
學程 必修 課程					
學程 選修 課程 **	學程選修(1)：	3			
	小計 (必修 2 選修 3)	5			

*校共同必修在下列課程任選兩門：

(1)中華民國憲法，(2)國學相關課程，(3)法律與政治相關課程，(4)藝術美學相關課程，

(5)倫理宗教哲學相關課程，(6)社會心理教育相關課程，(7)商管經濟相關課程

**學程選修課程：

(1)(1) 網路規劃與管理[3]，(2)家電通訊及遙控[3]，(3)資訊理論[3]，(4)寬頻網路[3]，

(5) 網際網路工程專題[3]

總計：

校共同必修	28	
院共同必修	43	
學程必修	30	選修 30
合計必修	101	選修 30

附註:

1. 學程必修是核心科目，考慮基礎性和廣度;選修科目深入特定技術或理論，考慮深度。

目前的規劃是以學生只修一個學程可畢業。(可以再選修輔學程)。

2. 校共同必修和院共同必修完全按院排定

3. 移動課程如下:

- (1)(1) 通訊與網路概論，由三上移到二下
- (2)(2) 互連與高速網路，由三下移到三上
- (3)(3) 網路程式設計(原網路程式規劃)，由四上移到三下
- (4)(4) 編譯器和人機介面，由三下移到三上
- (5)(5) 軟體工程，由四上移到三下

*4-5 配合軟體工程學程

4. 新增課程如下:

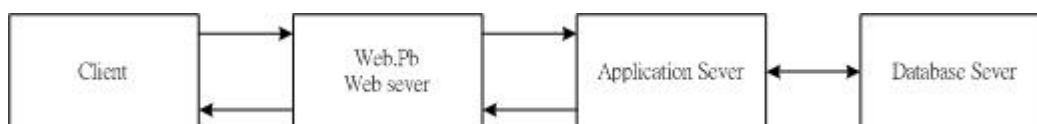
- (1) 密碼學，三下
- (2) 網路安全，四上(軟體工程學程之資訊安全)
- (3) 網路規劃與管理，四下
- (4) 網際網路工程專題，四下(原計算機網路專題)

5. 增加學分課程如下:

- (1)(1) 計算機操作實務-UNIX 改為三學分(二上)
- (2)(2) 個人計算機應用實務 改為三學分(二上)
- (3)(3) 多媒體系統概論 改為三學分(二下)
- (4)(4) 專題實驗 改為三學分(三下一學分，四上二學分)。

5/15/2000

2.2.2 現有系統架構



由於目前學校的選課系統是使用 Power Builder 系統所開發的應用程式作為介面，而課務查詢系統則是使用網頁當做介面，但是兩者間的資料庫互不相通，因此學校在開放選課系統的同時深怕由於選課資料庫與網頁資料庫間的更新速度不一，造成了選課系統與課務查詢系統間的不穩定，也為了怕同學們一邊選課一邊查詢課務資訊，造成了學校網路頻寬的負擔，因此近年來都會於選課期間暫停課務查詢系統的開放，而由於這樣造成了同學們在使用時，不能夠同時選課以及查詢課務資訊，使得同學必須用複雜的方法記下自己所需的資訊，因此若是將這兩套系統融合在一起，就可以彌補以上的缺失。

以下為選課系統以及課務查詢系統的使用圖：



圖 2.6、選課系統使用登入圖

逢甲大學校務資訊檢索系統

查詢條件

學年度：
 92年 91年 90年 89年 88年

學期：
 上學期 下學期 暑修上 暑修下

查詢項目：
 教師 科目 教室 系所班級(通識)
 時間 學生 綜合查詢(含時間)

個人資料：
 課表 選課一覽表 考程表 預選資料
 扣考資料 缺考資料 缺曠課資料 補考資料 期中考成績

查詢關鍵字：
關鍵字
(時間)週 第 節至第 節

檢索說明

個人資訊連結

請直接點選項目中的超連結，就可以直接查詢您的個人課程或考試資訊。

查詢條件

1. 請先選擇學年度、學期、查詢項目之後，在下方查詢關鍵字中的欄位填入（選擇）要搜索的關鍵字後，系統即會依據您給的條件進行關鍵字搜尋。
2. 查詢項目中的教師選項可以查詢授課時間、請益時間和監考表的資訊。
3. 查詢項目中的科目選項可以查詢課程大綱、授課教師、開課班級、修課學生和上課教室的資訊。
4. 查詢項目中的教室選項可以查詢教室課表的資訊。
5. 查詢項目中的學生選項可以查詢選課資料、預選資料、考程表、扣考資料的資訊。
6. 查詢項目中的系所班級選項可以查詢班級課程表、班級學生及班級開課一覽表的資訊。
7. 若要查詢上課時間請至綜合查詢(含時間)

圖 2.7、課務查詢系統使用圖

2.2.3 新系統架構

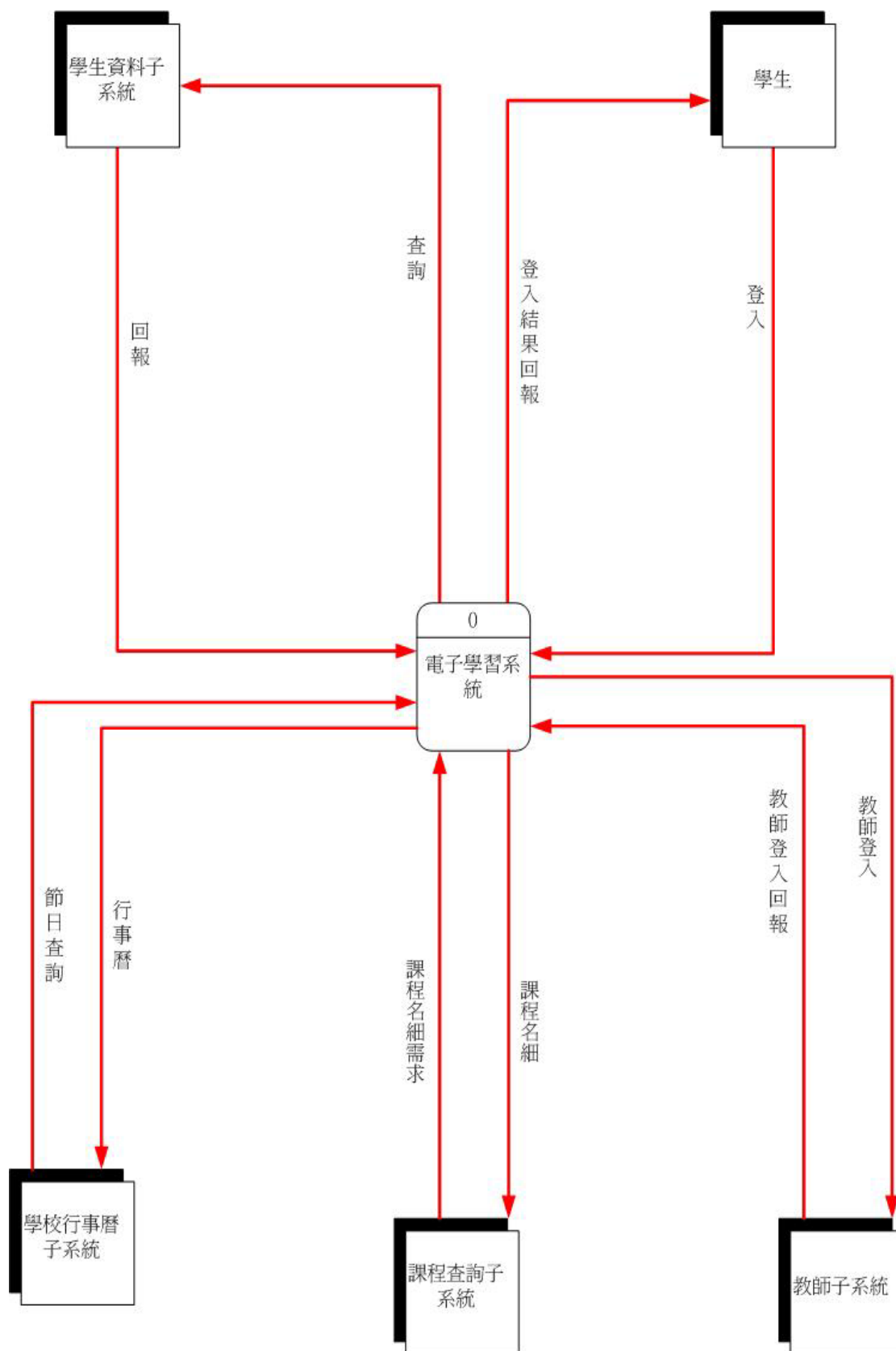


圖 2.8、全景圖

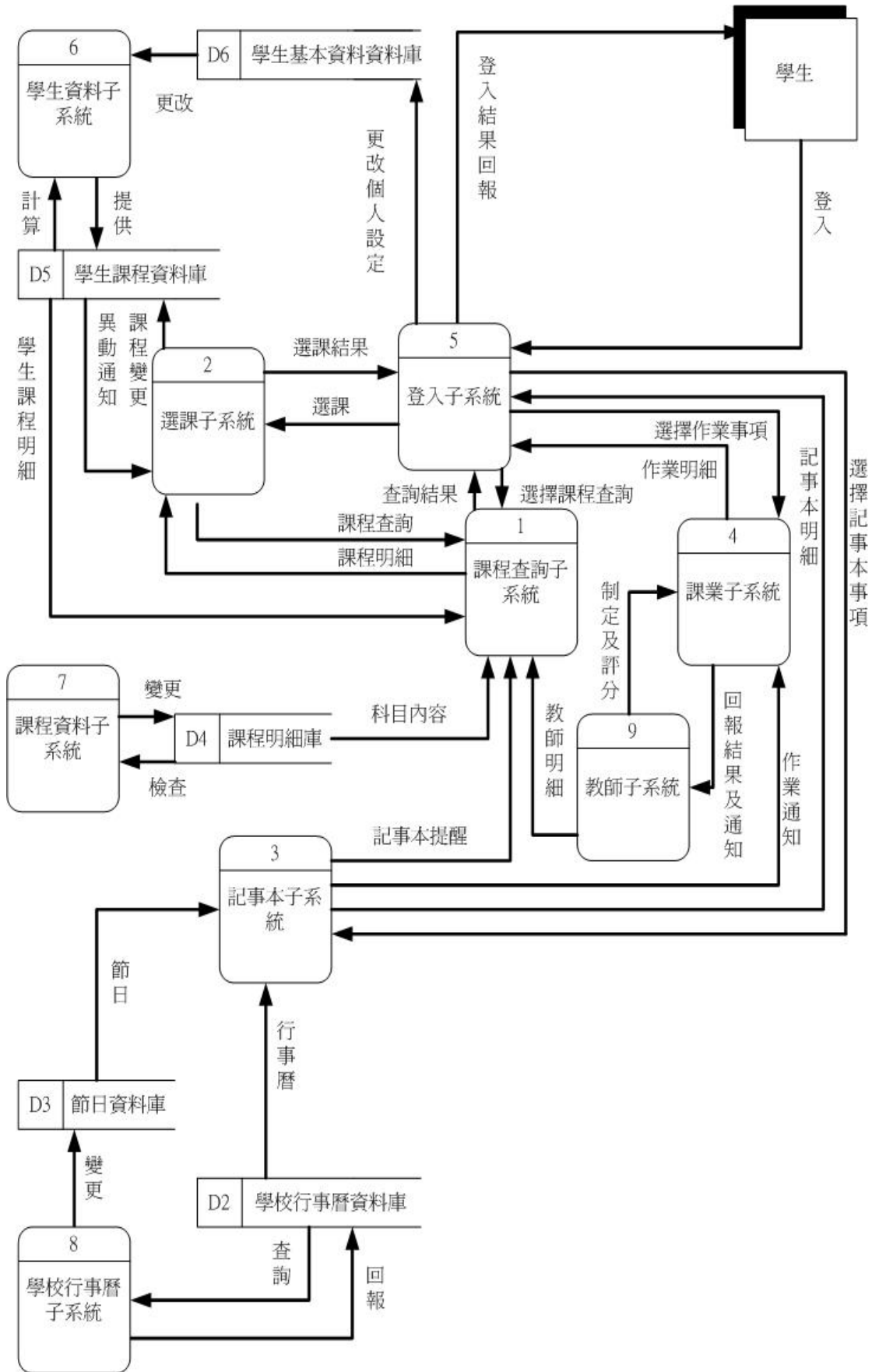
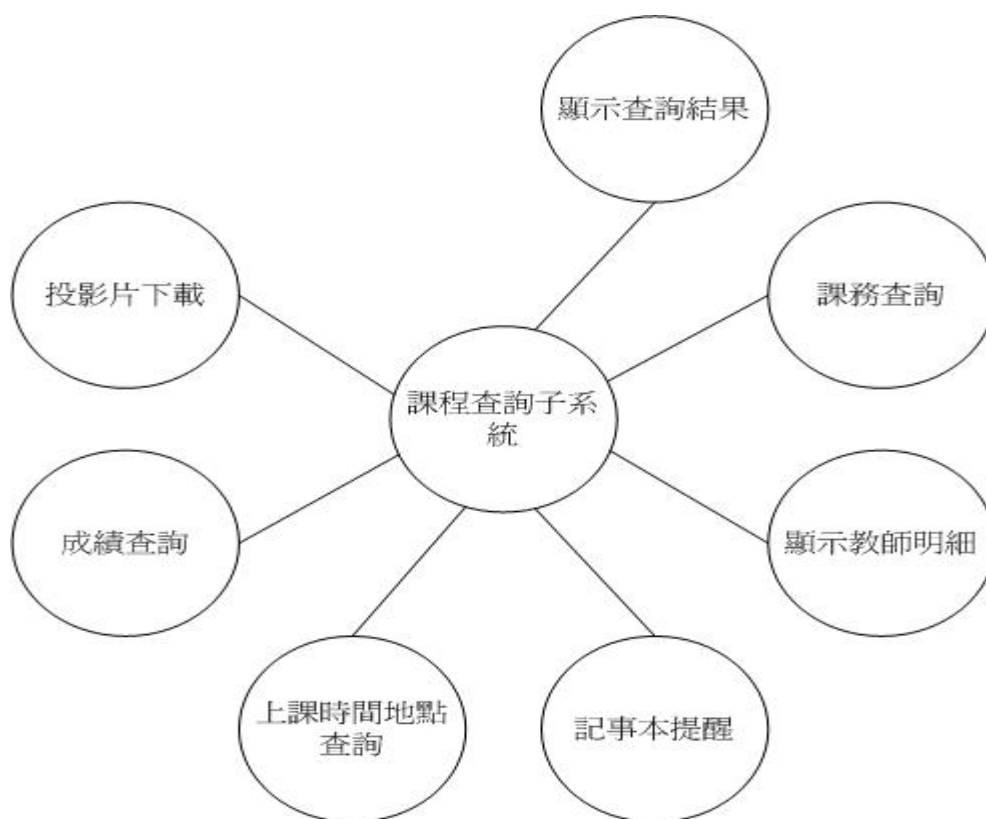


圖 2.9、圖 0

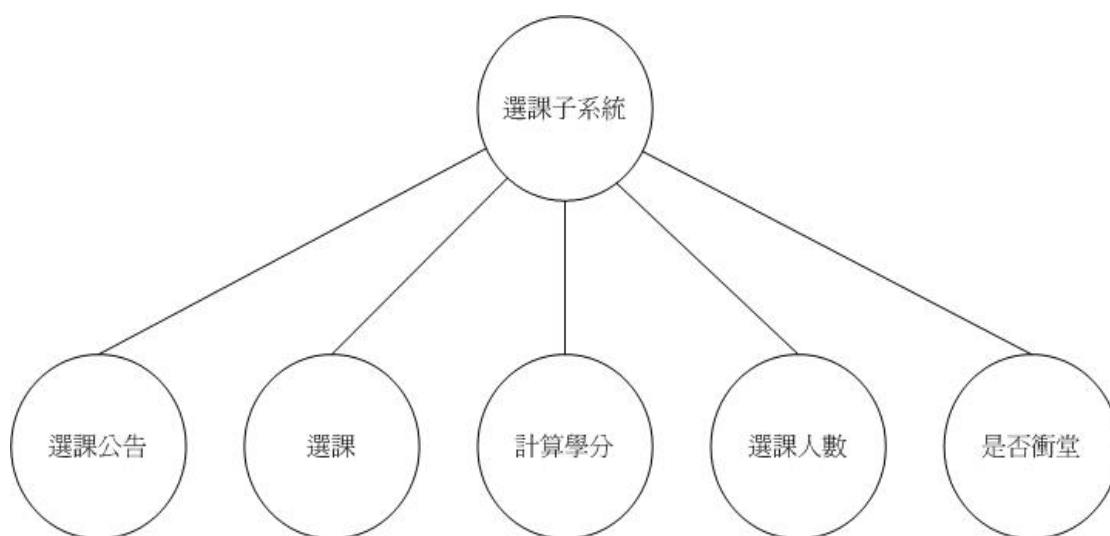
2.2.4 課程查詢子系統需求分析



課程查詢子系統

圖 2.10、課程查詢子系統需求

2.2.5 選課子系統需求分析



選課子系統

圖 2.11、選課子系統需求

2.2.6 課業子系統需求分析

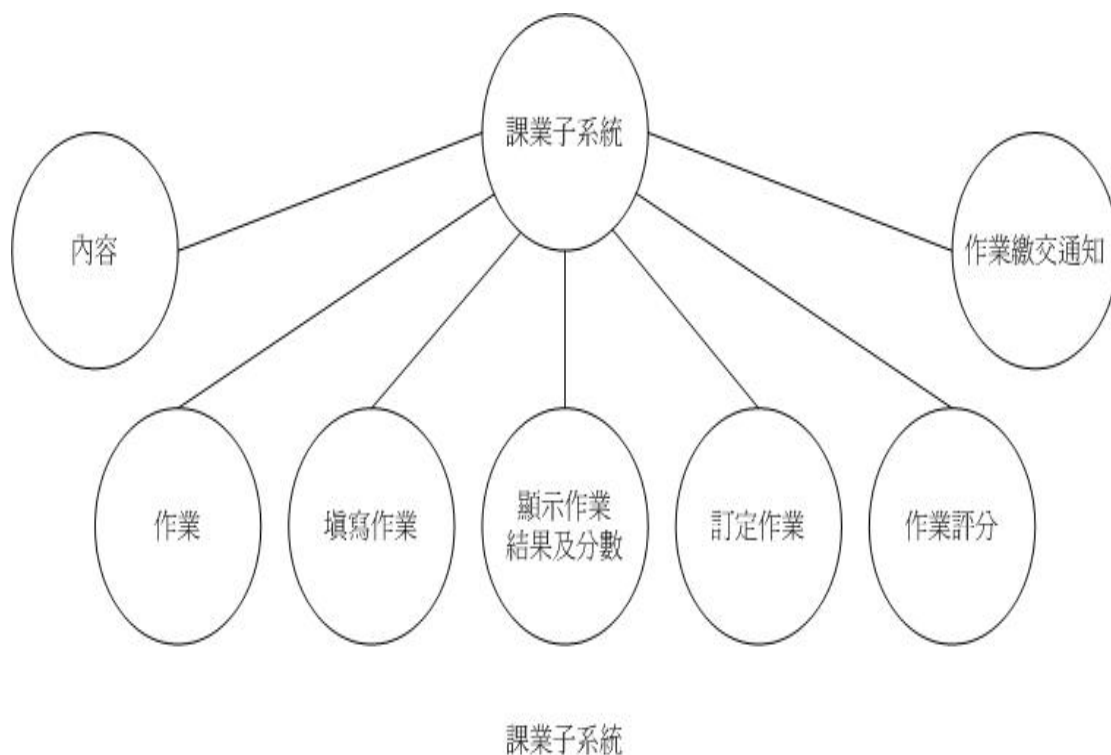


圖 2.12、課業子系統需求

2.2.7 課程資料子系統需求分析

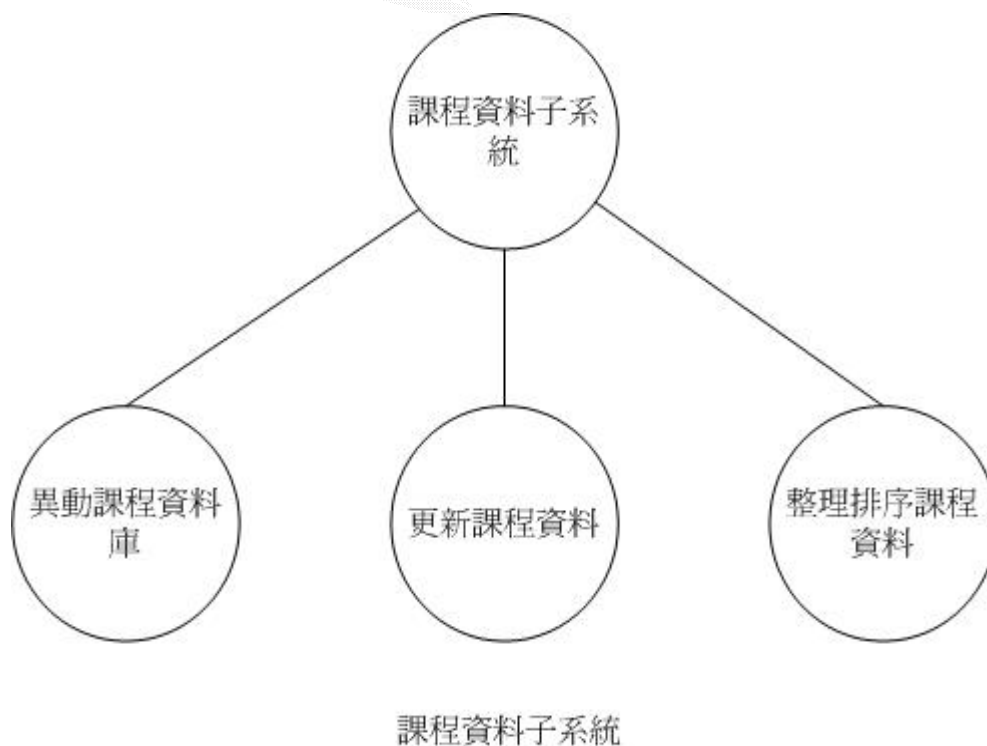
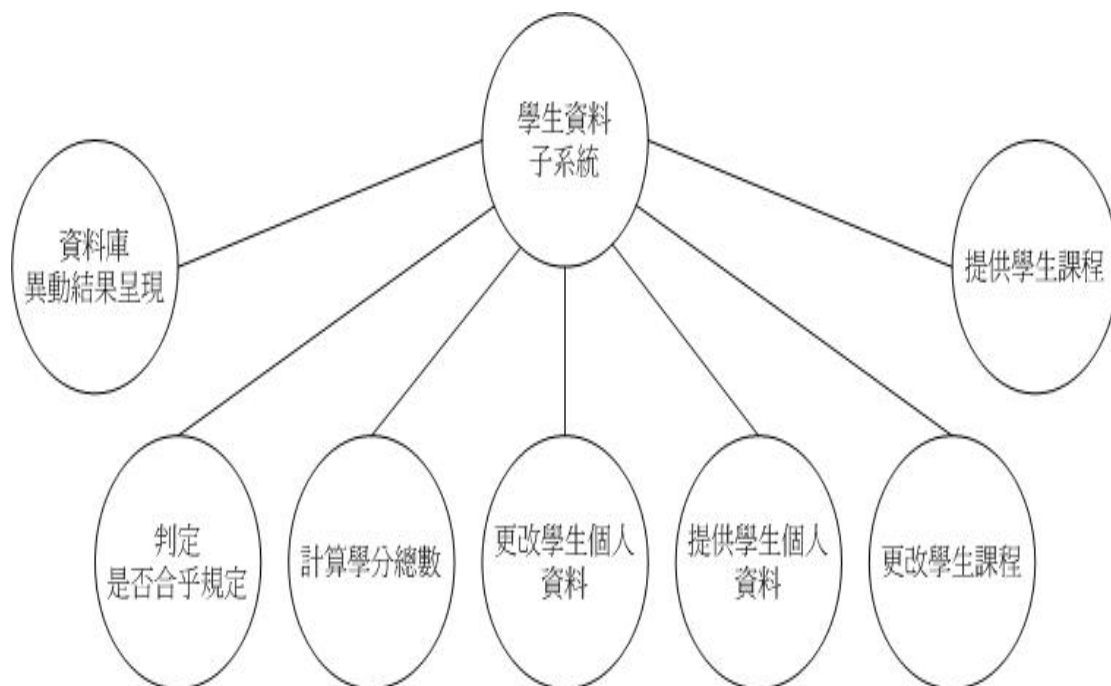


圖 2.13、課程資料子系統需求

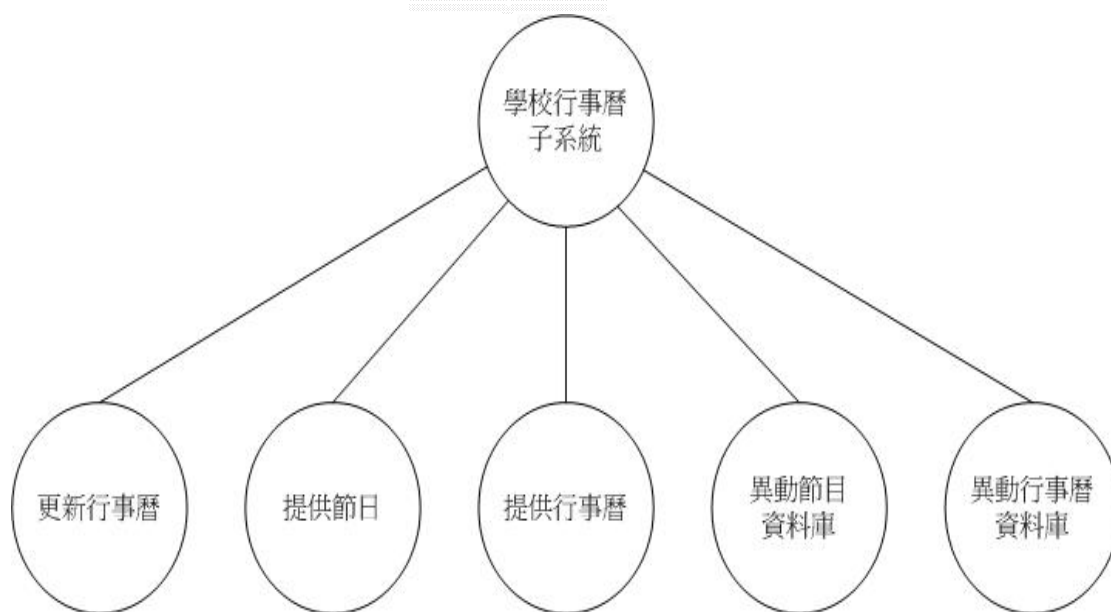
2.2.8 學生資料子系統需求分析



學生資料子系統

圖 2.14、學生資料子系統需求

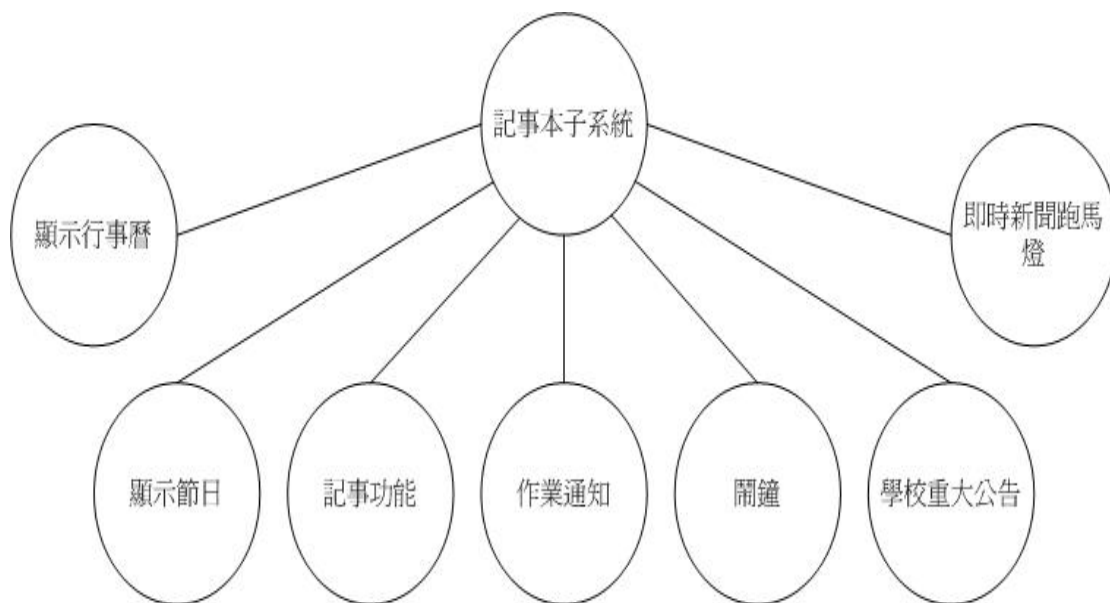
2.2.9 學校行事曆子系統需求分析



學校行事曆子系統

圖 2.15、學校行事曆子系統需求

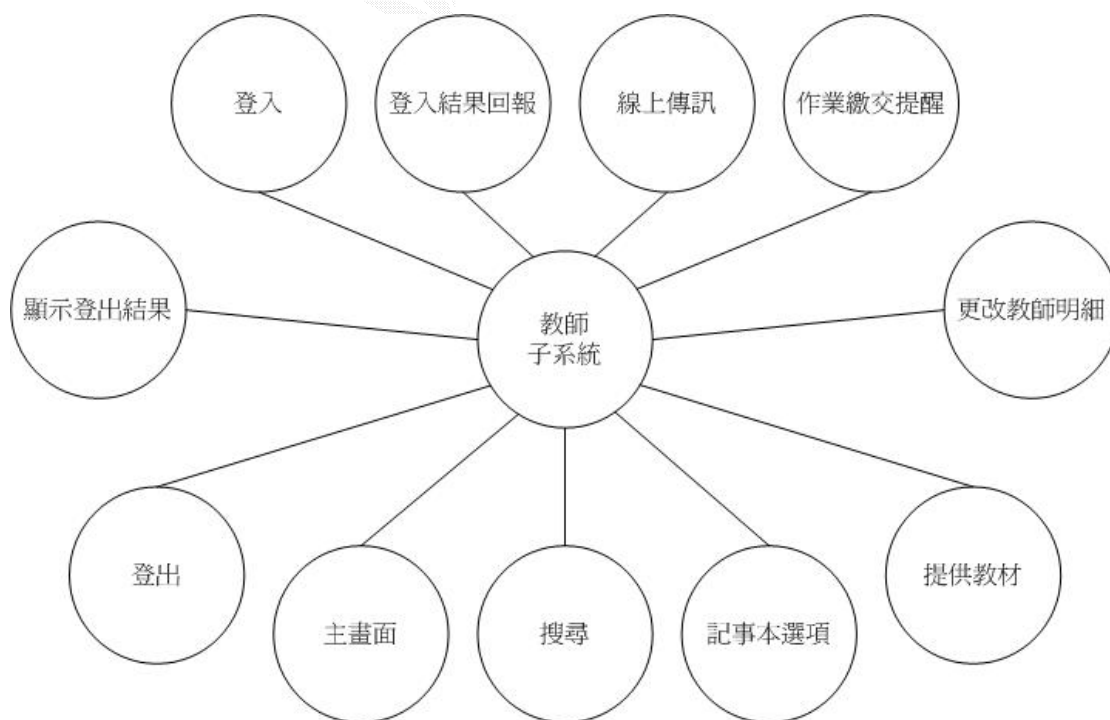
2.2.10 記事本子系統需求分析



記事本子系統

圖 2.16、記事本子系統需求

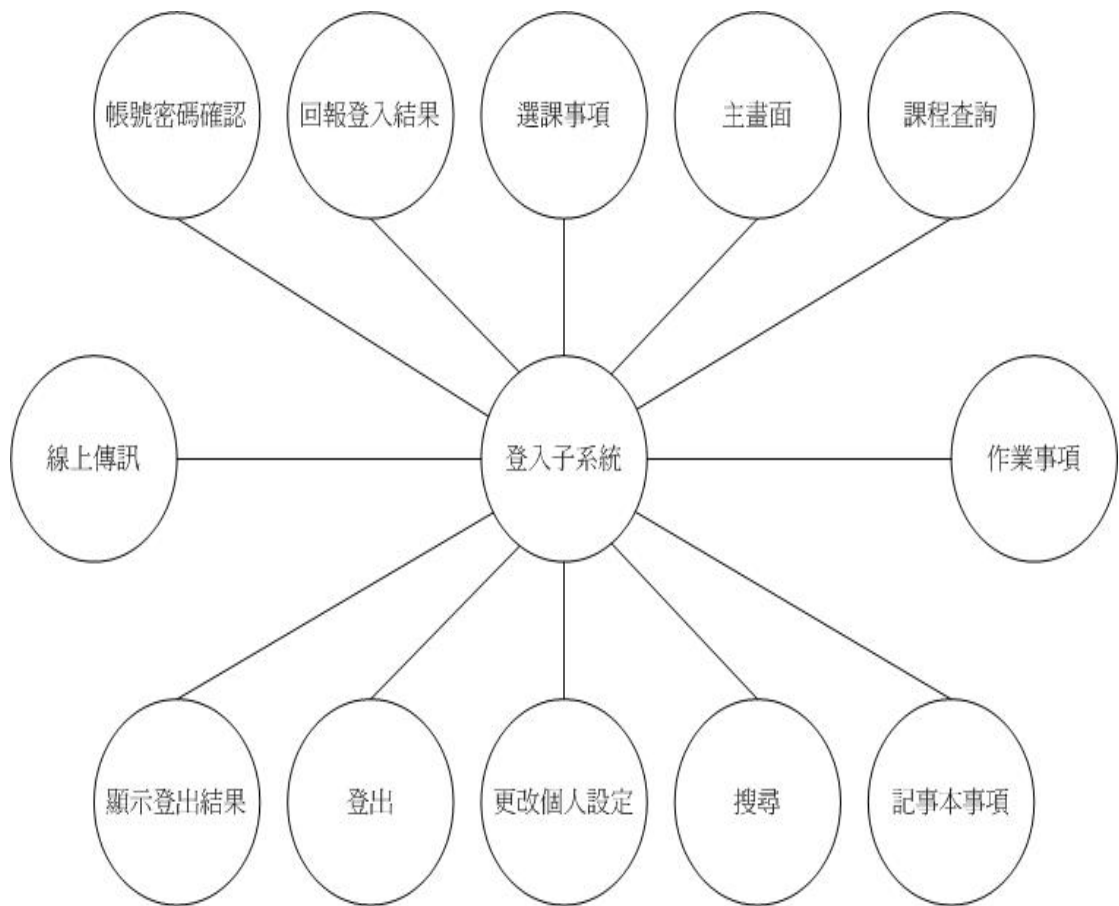
2.2.11 教師子系統需求分析



教師子系統

圖 2.17、教師子系統需求

2.2.12 登入子系統需求分析



登入子系統

圖 2.18、登入子系統需求

第三章 系統分析與設計

3.1 系統開發流程模式

3.1.1 系統開發流程模式介紹

由於軟硬體的技术不斷成長，資訊系統的需求量與複雜度亦不斷地大幅提高，因此如何有效的提高軟體的生產量和品質便是一件非常重要的課題。軟體開發流程模式是資訊系統開發活動一系列的步驟及執执行程序。當系統開發依循系統化、邏輯化的步驟進行時，有利於標準、規範與政策之推行和建立，開發的過程將會更有效率，更能確保品質，也更容易管理。許多專家分別研究出不同的資訊系統開發流程模式，以便處理各種不同情況的系統開發。以下便介紹各主要開發流程模式的執行步驟及其適用情形。

1. 瀑布模式(waterfall model)

瀑布模式開發是一種有系統、符合邏輯的方法。它依軟體生命週期規則規劃了一序列的階段，且每一階段的產出都必須經過確認、驗證。所以每一階段的結束都是一個開發的里程碑。應用瀑布模式開發時，各階段必須考量完整的需求，每一階段的輸出將作為下一階段的輸入。是故，前一階段必須完全完成，且有一個完整的結果，才能移至下一階段。

瀑布模式一班適用於低風險、需求變動小又可以清楚表達的專案。它強調完整的分析與設計文件，可有效的確保系統品質，但倘若需求變更變得要回到分析階段重作完整的系統分析，將文件重新大幅度的修改。並且，當系統能夠正式運作時，已是專案的開發末期了，若在此時才發現錯誤或是不合乎需求，便會造成前功盡棄，因此客戶若經常的變更需求，或是無法在專案開始時便能清楚描述需求時，瀑布模式便明顯的不適用。

2. 原型模式(prototyping model)

通常，客戶只會定義出一些系統要達成的目標，而不會確認詳細的輸出、輸入或操作需求，或當開發者沒有足夠的經驗來了解使用者需求，亦可能一時無法找出解決問題的方法或技術等。這種情況下，原型模式便可能提供最好的方法。

其執行程序與原則為先針對使用者需求叫清潔的部份或是設計師較能掌握的部份，依分析、設計與實施等步驟，快速開發系統原型。此時的焦點放在軟體中客戶可以看到的部份，並由客戶來進行評估，加以確定需求。雙方透過雛型之操作與回饋，以釐清、修改及擴充需求，並藉以修

改與擴充原型。原型不斷的調整，直到系統符合雙方約定為止。

然而使用此種方模式亦會產生下列的問題：

- (1) 因缺乏整體規劃、分析與設計，靠著不斷快速修改原型，難以顧及品質及長程的可維護性。
- (2) 開發者可能為了使原型能夠快速工作而對實行部份作出讓步，結果可能使用了不合適的架構或演算法，只因為它是便於使用且熟悉的。

3. 螺旋模式摘要(spiral model)

螺旋模式是一種將原型的性質和瀑布模型的控制結合在一起的進化模型。在螺旋模式中，軟體是以一系列的增量版來進行開發，各週期之進行均強調規劃及風險評估，並逐漸的建立出完整的系統。

螺旋模式每一週期的執行三步驟：

(1) 找出系統的目標，可行之實施方案與限制

一週期有而有不同的詳細程度，當週期反覆進行時，描述會越來越詳細。

(2) 依目標與限制評估方案

此步驟主要是找出各方案之不確定處並設法解決。

(3) 由剩下之相關風險決定下一步驟

當通過風險評估後，依照剩下的相關風險決定下一步驟應朝何種方向發展。

如同其他模式一般，螺旋模式並不是萬能的。它可能會難以說服客戶這種進化的方法是可以控制的。它需要考慮風險評估的專門意見，若是有一種主要的風險並未被發現且納入管理，便可能使得專案失敗。¹

3.2 軟體開發工具分析

3.2.1 Web 應用程式

Web 應用程式，是以網頁為基礎的程式架構。類似傳統的 Client-Server 架構，透過網際網路，不限制任何作業系統平台，客戶只要使用符合標準瀏覽器便能操作系統，無須下載安裝任何客戶程式。就算是更新系統，亦只要更新 Web Server 這端即可，客戶無須作何更新動作，便能享受新的服務。

現行的 Web 應用程式開發技術，最主要的有 ASP、JSP、PHP 以及 ASP.NET。

(1) ASP，全名為 Active Server Pages，是 Microsoft 公司所開發出來的 Server 端網頁技術，採用腳本語言 VBScript 和 Jscript 作為開發語言，可開發動態且高性動的 Web 應用程式。

(2) JSP, Java Server Page, 是 Sun 公司推出的新一代網站開發語言, 使 Java 在網頁方面的應用, 除了可使用 Java Applet 之外的另一個新技術。JSP 可以在 Servlet 和 JavaBean 的支援下, 完成功能強大的 Web 應用程式。

(3) PHP, 是一種跨平台的伺服器端嵌入式腳本語言。它大量的借用 C, Java 和 Perl 語言的語法, 並耦合 PHP 自己的特性, 使 WEB 開發者能夠快速的寫出動態產生頁面。PHP 是自由軟體聯盟所發展的, 因此 PHP 是完全免費的, 而且可以不受限制的獲得原始碼, 甚至可以從中加進你自己需要的特色。

(4) ASP.NET, 是 Microsoft 公司推出的新一代的網頁應用技術。屬於 Microsoft 公司 .NET 願景之一的重要技術。利用強大的 .NET Framework 平台和 Web Services 建構出網際網路的分散式架構, 並且可使用符合 .NET 規範的任何程式語言, 如 Basic、C++、Cobol、Jscript 等作為開發的語言。

就上述的四種 Web 應用程式技術, 我們分別以我們的線上學生系統所要求的速度、元件化、行動通訊這三個目的來對其作討論及選擇。

(1) 快速：

由於我們要改善原有的課程查詢系統速度上的缺點, 我以

執行速度是我們的第一考量。就架構而言，ASP 和 PHP 是利用 Script 語言作為開發語言，因為直譯式程式執行的先天限制，所以在速度上無法跟經過編譯的 JSP 和 ASP.NET 相比，所以就速度方面而言，ASP 和 ASP.NET 是比較符合我們的需求。

(2) 元件化：

除了 PHP 外，另外三種技術皆有分層式的元件設計。可以將商業性的邏輯運算分開獨立包裝成一個元件。就技術方面來講，JSP 和 ASP.NET 因為採用物件導向的架構，因此對於撰寫元件的便利性較高。

(3) 行動通訊：

以這方面而言，ASP.NET 略勝一籌，她特別為行動通訊裝置設計 Mobile ASP.NET，將網頁標籤包裝成一個個元件，依照前端瀏覽器的不同，自動判斷產生合乎其特性的標籤。例如客戶使用 WAP 通訊裝置連接，便會回應 WML 標籤；或是客戶以 PDA 連接，便會回應 HTML 標籤。使得開發者只需撰寫一次，便可適應所有不同的裝置。

綜合以上三點的分析討論，我們認為 ASP.NET 是比較適合且方便達成我們的系統所要求的速度、元件化、行動通訊這三個目的，所以我們選擇使用 ASP.NET 作為我們的 Web 應用程式開發技術。

另外，選擇使用 ASP.NET 還有以下幾個優點：

(1) 功能完善的核心：

.NET Framework 提供了完整的物件導向核心，可使用數種程式語言開發應用程式，並且有功能強大的執行環境，提供了跨語言的執行平台、自動資源回收和型別安全檢查等功能。

(2) 良好的開發框架：

ASP.NET 的 Web Form 設計，透過事件驅動和資料繫結，讓開發 Web 應用程式的方式和一般開發普通的 Windows 應用程式的方式一致，使得開發更易上手。並且透過 Code-Behind 的方式，讓 HTML 標籤與程式碼可以完全地分開撰寫，免除了二種程式混在一起分不清楚的惡夢。

(3) 擁有良好的整合式開發環境：

Visual Studio.net 為微軟新衣戴用來開發 .NET 平台上程式的開發工具。自動化和視覺化的設計工具，使得程式設計師能專心地撰寫程式的邏輯部份，而不需要把精力放在與使用者操作的 UI 介面上。

3.2.2 資料庫管理系統

因為採用微軟所推行的 .NET 技術來開發系統，故對於資料庫管理系統亦採用同公司所出的 SQL Sever 2000。其優點有：

1..NET 架構對於 SQL Sever 2000 整合良好，並且有最佳化的處理。

2. SQL Sever 2000 在中小型資料庫上的表現卓越。

基於以上幾點，所以我們決定採用 SQL Sever 2000 做為我們的資料庫管理系統。

3.2.3 單元測試框架(Case Tool)

由於我們使用了單元測試來確保系統的正確性，所以我們需要一自動化的工具，來自動的幫我們完成測試的工作。

就目前而言，約有三套單元測試框架式可用自動化測試.NET 上的應用程式，而它們都是依照 JUnit 這套用來測試 Java 程式的框架，作為開發的範本，所以在使用方面上，差異度並不十分地大。我們選擇採用目前成熟度較高的 NUnit2 這套產品。其擁有圖形介面自動測試執行工具，可分別驗證各單元是否正確，並回應完整或是自定的錯誤報告，並且在語法方面使用屬性標示出測試程式碼的段落，較為簡單明瞭。

3.3 硬體設備

開發環境：P4 2.4G

512 RAM

WINDOWS XP

SQL SEVER 2000

3.4 系統分析設計工具

對於我們的系統，我們採用了物件導向的分析方式，以物件之間的互動，塑造出整個系統的模型。如同建照高樓大廈所需要的藍圖或是管線配置圖等，除了可以讓工程師按圖施工外，亦可讓人能概略的了解整棟大樓建立起來時的模型。而軟體系統的模型也是一樣，以物件的方式建造我們系統的模型，除了可以方便達成元件化的目標之外，更可藉著概念的模型將複雜的系統清楚的表達出來，使思維更加透徹。在更進階的做法，還可直接利用模型在和物件導向語言之間作出正向及逆向工程，一方面減輕編碼負擔，一方面還可以確保文件和程式碼之間的一致性，以下分別介紹塑模時所使用的描述語言—UML，以及採用的塑模技術—Design Patterns

(1) UML(Unified Modeling Language)

UML 是一種視覺化語言，用來為系統建立模型。他並非是一種軟體方法論，並不是規定系統應該如何建制的流程，而只是一種定義系統模型如何表示的方法。

在 UML 中圖形是根據簡化的觀點所表現出系統的投射，而每一種圖形皆是由元素和代表元素間關係的線所組合而成的。在系統開發時不同時期，我們對系統的看法會有不同抽象等級，

UML 中定義了九種的圖形，分別在系統開發時不同的時期描述系統，好讓我們根據不同的角度來看待系統。這九種圖形分別為：

- a. 類別圖(Class diagram)
- b. 物件圖(Object diagram)
- c. 使用案例圖(Use case diagram)
- d. 順序圖(Sequence diagram)
- e. 合作圖(Collaboration diagram)
- f. 狀態圖(Statechart diagram)
- g. 活動圖(Activity diagram)
- h. 元件圖(Component diagram)
- i. 部署圖(Depolyment diagram)

以下我們分別簡介類別圖、順序圖和活動圖這三種在我們的系統中，我們用來建制系統模型的圖形。

- a. 類別圖：

類別圖是物件導向方法中最核心的部分，也是最被廣泛使用來建立模型的概念。從概念的角度，顯示使用者如何看得這個世界；從規格的角度，則顯示軟體元件的介面；而從物件類別的角度，則顯示出軟體元件的實作。類別圖示一種描繪出類

別(class)、類別的內部結構和操作，以及類別和類別之間的靜態關係的圖形。其中類別是用來定義物件的詳盡描述，包含了欄位(field)、方法(method)等。

UML 表示抽象類別(abstract class)和具像類別的圖形。類別是以長方形方塊來表示，其中類別的名子位於方塊上方，中間列出類別的屬性、個體變數，最下面列出類別的方法。屬性和方法的型別資訊可有可無，方法傳回執型態列於方法的後面。名字若是斜體的，便代表此一類別或方法為抽象類別或是抽象方法。

顯示類別之間的各种關係。UML 用一個空心三角形的實線箭號表示繼承關係，由子類別連到父類別。直線代表是聯繫(association)關係，表示類別互有關聯，而菱形的箭頭是一種特殊的聯繫關係，為組合關係，表示菱形箭頭處的類別擁有另一端的類別的實例。我們亦可在官系上標示出參與物件數量上的多重性。

b. 順序圖：

順序圖用來描繪一群物件間的一些行為，說明它們之間訊息流動的順序。

順序圖中時間是由上而下的遞增，垂直虛線代表了物件的

生命線，垂直長方形代表物件取得控制權，正在處理外界送來的訊息要求。訊息是以在兩個物件生命線之間的水平時線表示，箭頭指向訊息的目的物件，箭號上附加訊息的名稱，而訊息的回覆則是以虛線箭頭來表示它。

物件導向程式中最讓人難以理解的地方在於控制的流程，有時很難可從程式碼中看出物件間方法執行呼叫的順序。有了順序圖之後，便能幫助我們清楚了解整個執行的先後順序和過程。

c. 活動圖：

活動圖示 UML 裡面用來塑造系統動態角度的五種圖形之一。活動圖基本上就是流程圖，可以顯示活動與活動之間的控制流程。活動圖支援條件是與平行行為列用活動圖可以塑造出系統計算流程中的循序步驟和同步步驟，或者是一個複雜的訊序性算法。實心圖和加框實心圖分鼻代表控制流程的啟動與結束，圓角矩形代表系統中活動或動作的狀態。箭號為活動狀態完成，將控制權轉換到下一個動作或活動上。空心的菱形比表示控制轉換的分歧以布林表示式進行條件式的轉換，水平的直線棒，則是代筆著同步流程時狀態的分叉和結合。

活動圖跟傳統的結構化設計中流程圖的功能十分的相像，都

是在說明活動的控制流程，利用活動圖，更可明白表示出系統中的活動順利以及工作流程。

(2) Design patterns

在 UML 中，只定義了如何用圖形表達物件導向設計。相反的 patterns 看的是結果－已經設計好的模型範例。

Patterns 是描述作事情的共通方法。對於某些類型的問題，往往會被設計成相同的架構，而這些架構或是設計，已被證明是有效且能重複使用解決問題的，這些設計便被蒐集起來並且給他一個名子。這種有名子且能重複套用的設計，便稱之為 patterns。每一個 Patterns 都是前人經驗累積的精華，它描述了一個重複發生在我們環境中發生的問題，也描述了問題的核心解法。它提供了設計者一個共通的名子，以幫助團隊溝通，也提供了如何將問題的解決成為一個高凝聚力、低耦合力的設計。Design patterns 給予我們專注在問題本身，以及設計物件導向時更高的觀點，讓我們不會過早被太仔細的細節所牽絆住。利用 design patterns 設計系統可將複雜的系統變成單純的解決一個一個的問題，並且擁有高度彈性的架構。要創造良好的物件導向設計，有幾個策略可使用，如：

- 對介面設計

- 使用組合，勝過繼承
- 找出變化，並將它封裝

這便是 design patterns 的精神。以下舉例簡介何為

Singleton patterns：

有時候我們必須卻表某些類別只能有一個物件實體，譬如說，製作一個印表機的佇列。該如何才能卻表此一類別只能生成一個實體，但又能方便的存取這個物件？我們可以使用一個全域變數來存放這個實體，但是缺點是無法卻表別人刻意不使用這個變數，自己生成實體出來。

比較好的做法是讓列別自己管理這個唯一的實體，藉由一個特殊的方法去生成這實體，並讓他自己保證絕對無法再生出第二個實體出來。這就是 singleton patterns。

Singleton patterns 藉由一個特殊的方法，當它被呼叫時，它會檢查是否已有實體。如果有，則傳新物件的參考。並且，為了確保別人無法擅自生成實體，將此列別的建構式定義為 protected 或是 private。

以 C# 為例，實際的 Singleton 類別會宣告成類似以下的程式碼：

```
public class Singleton
```



```
{ public :  
  
    static Singleton GetInstance() ;  
  
    {  
  
        if(instance==null)  
  
            instance=new Singleton() ;  
  
        return instance ;  
  
    }  
  
private :  
  
    Singleton() ;  
  
    Static Singleton instance ;  
  
}
```

這種設計比單單只是定義成全域或是靜態變數更為優良有彈性，並且確保了類別實體的唯一。像這種可輕易設計出良好解決特定問題的架構，便是 design patterns 的威力。

3.5 系統模型

最概括的說，系統可描述以下的模型。

現在考慮系統中的架構，因採用 ASP.NET 來時作課務資訊系統，故需依照 ASP.NET 所訂定的架構來建立系統。在 ASP.NET 的架構裡，網頁上的表單控制項皆被包裝成 Web Control 元件，並使用了事件驅動

和資料繫結的開發方式，使得讓前端的 UI 介面和後端的程式碼可清楚的分開撰寫，即是 Design Patterns 中的 Mediator 模式和 Observer 模式的混合運用。ASP.NET 又透過 Code-Behind 的技術，將程式碼和展示用的 HTML 標籤分開在不同的檔案，使程式更可讀性，其中 HTML 標籤是寫在 *.aspx 的檔案中，而程式的部分則寫在 *.cs 中。故對所有的子系統，我們可一般化成下列模型。

因此，我們只需思考在商業邏輯層中的架構設計即可。首先考慮系統中有可能的所有變化：

1. 使用者：依使用者的身分不同，所要求的結果便可能不同。例如學生在個人課表中，應顯示出其上課的課表，反之，老師則應顯示出其授課的課表。
2. 系統資料庫：雖然我們使用 SQL Server 2000 作為系統資料庫，但因各學校所採用的系統不同，亦不可能強迫學校購買 SQL Server 2000，故在移植到新學校時會採用的資料庫系統的不同而變更相對應的程式碼。

首先我們先考慮使用者的問題，因使用者的不同，所需對應的邏輯層便不同，例如以教師帳號登錄的使用者，其首頁便不應該出現作業繳交通知的功能，故系統應能依使用者的類別不同，顯示出符合其身分的功能。

第一種做法是更改顯示層，顯示為不同的使用者建立相對應頁面再使用者登入時便導向所應該顯示的頁面，此種做法會使得每一頁的重複性過高，並且維護不易，若再新增一種使用者，一樣要花費同樣的功夫建構。

第二種做法是依使用者不同，更改其對應的邏輯，使用程式去更改頁面成符合使用者身份的顯示。為了不使得對應的邏輯寫死於程式中，造成日後維護的困難，故我們使用了 Strategy 模式，將依使用者不同所對應的邏輯給封裝起來。

整個架構便成了 Smalltalk 裡的 MVC 設計。View 負責顯示畫面，為 HTML 標籤，(即*.aspx 檔)，而 System.Web.UI.Page 退化成了 MVC 設計中的 Controller，只負責對使用者介面的輸入作出相對應的處理和將結果對應於 View。其餘的部分則為 Model，負責應用軟體物件。我們訂定出了 Page 的責任，接下便可繼續思考 Model 中的系統的架構設計。

抽象類別 Group 定義出在此系統中所需那些運算，再分別依學生或老師的不同分別實現其實際的動作。Config 類別用來存取 Config.xml 檔，此檔案紀錄著系統管理這所作的設定。

此設計的好處在於讓 Controller 可以對所有的使用者皆一視同仁，作出相同的動作。而使用者之間的所需差別，則由使用者所屬的

群組自行負責，日後若要新增新種類的使用者，只需增加相對應的群組作出其所需的邏輯即可。

但目前的設計卻會使得資料庫與系統過分耦合，一旦資料庫系統更換，便得重新編寫 Student 和 Teacher 中對資料庫動作的所有程式碼。

另外在 .NET Framework 中，對資料庫的操作模型為 ADO.NET。它提供了 Data Provider 物件用來連接資料庫，執行資料操作指令、取回查詢資料等功能。但對於所連接的資料庫不同，其所對應的 Data Provider 物件也不同。若連接的對象為 SQL Server，便應使用 SQL Server .NET Data Provider，若連接 Oracle 資料庫便應使用 .NET Data Provider，若連結 Oracle 資料庫，便應使用 .NET Data Provider for Oracle。也因此，對於使用的資料庫的不同，便要重新編寫所有有關資料庫連接部分的程式碼。

所以讓我們先考慮系統資料庫，依學校的不同，所使用的資料庫便可能不同。若將資料庫的操作方式寫死在程式中，將來移植時必然要花費很大心力，並且每一 Group 子類別皆各自連接資料庫，出錯的機會亦會大大上升，也造成維護上的困擾。

所以我們使用 Facade 模式把所有對資料庫的動作單純化為 Select、Insert、Delete 和 Update 四個動作，以及使用 Bridge 模式將操作資料庫和如何連接資料庫的時作部份分開。

如此，一但要新增連接到埠同的資料庫，便只需使用 Adapter 模式，將其所對應的 Data Provider 轉換成符合 ConnectDB 的介面即可。

但是在 Group 中仍然需要指名所需的 ConnectDB 子類別為何，並無法在程式執行時動態的去挑選要聯結的資料庫引擎，我們希望系統能和使套裝軟體一般的方便可依系統管理者的設定，自動創造合適的 ConnectDB 子類別，如此在移植時便能作到最小的改變。所以加上了退化的 Abstract Factory 模式，讓系統自行判斷，動態生成合適的 ConnectDB 子類別。

CreatConnectDB 會依 Config.xml 的內容，選擇要生成何種的 ConectDB 子類別。

接下來，我們在繼續考慮關於課程資料庫的變化。為了操作的方便性和一至性，我們亦利用 Adapter 模式將對課程資料庫的操作包裝成 ConnectDB 的介面。

考慮課程資料庫的變動，和課務資訊系統資料庫一樣，隨著學校的不同，課程資料庫可能為不同種類的資料庫系統，而且絕大多數的 Table 定義都可能不同，Table 名稱不同，欄位名稱不同，Table 之間的關係也不同，甚至可能分散到好幾個資料庫裡。所以想要轉移系統至其他學校，修改程式碼是無可避免的。

我們必須思考一種好的做法使修改的程式碼維持最小，讓系統的主

要邏輯不受到資料庫變更的影響。是故，利用資料庫中 VIEW 的觀念並運用『找出變化，並封裝它』的準則，我們定義出了一個虛擬的資料庫，讓系統對於課程資料庫的操作皆使用這個虛擬資料庫的定義，再由虛擬資料庫轉換為真正課程資料庫的定義，使得課程資料庫整個隱藏起來，系統只對表面上的虛擬資料庫作動作而已。如此，當課程資料庫變更時便單單只需改變虛擬資料庫這部份的實作即可。

接著，為了讓課程資料庫的改變對系統的影響減至最小。我們將虛擬資料庫移出系統，並包裝成 COM+ 文件，讓它真正與系統獨立運作，並可依課程資料庫的改變，輕易抽換，而不用重新改寫系統。

最後考慮 Config 物件，因為系統是架構在 Web 之上，因此我們在存取 Config.xml 檔，必須要先把 Web 伺服器上的虛擬路徑轉換成為實體的檔案路徑，才可存取的檔案，是故在 Config 中，本身應有能力把 Config.xml 的虛擬路徑轉為實體檔案路徑以供後續動作使用。但在 ASP.NET 中，提供此轉換功能的物件地址能由 ASP.NET 應用程式實體化，所以 Config 勢必得提供一方法，接受此一物件，才能在內部進行路徑的轉換。

問題在於此一物件是 System.Web.UI.Page 中的屬性，若要讓 Group 包含 Config，就必須讓物件先由 Page 轉向 Group，再由 Group 傳至 Config，讓 Config 進行轉換，這麼做的話便會使得系統的耦合性提高。

若讓 Config 在 Page 初始化完成後在交由 Group 使用，又會讓 Page 的責任變的模糊不清，讓系統的設計不合理。

因此，為了解決此一問題，我們配合了 ASP.NET 的設計，在 ASP.NET 應用程式開始執行時先行初始化 Config，為了避免(也無法)使用全域變數，故利用 Singleton 模式讓 Config 可保持此實體，使得雖然不用宣告 Config 型態的全域變數，也讓 Group 物件能夠使用 Config。

3.6 資料庫設計

在這一節中，我們將要介紹「電子學習(e-learning)系統-以逢甲大學資訊工程學系學生選課設計為例」資料庫的設計過程，至於資料庫的設置成果，將留到第四章系統設置時，在行介紹。

現在，我們先簡單討論資料庫的設計過程。一般而言，資料庫的設計可分成下面五個主要的階段：

(1) 需求的收集與分析(Requirements collection and analysis)

在這一階段中，我們主要是詳細分析資料庫使用者與預期資料庫使用者的期望，並將與作條列式的描述，即為此系統的「資料需求」。這一部分我們將在 3.7.1 節介紹。

(2) 概念資料庫設計(Conceptual database design)

這一階段中，是將上一階段的「資料需求」轉化成「高階概念資料模型」，我們選擇「實體關係模型」-ER model(Entity

Relationship Model)作為高階概念資料模型。此一部份，將在 3.7.2 節在做介紹。

(3) 邏輯資料庫設計(Logical database design)

這一階段是將「高階概念資料模型」轉化成「概念綱要」。由於，我們先前以決定要使用 SQL 2000 作為資料庫設計管理的工具，而 SQL 2000 的「象徵實作資料模型」為「關聯式」，因此，在此階段中，我們便將之前的 ER Model 轉化成「關聯式資料庫綱要」。這一部份，我們將在 3.7.3 節中介紹

(4) 實體資料庫設計(Physical database design)

此一階段，是將「關聯式資料庫綱要」轉化成「內部綱要」，即為「低階實體資料模型」。在此我們定義了各資料行的資料型別、長度、是否允許 Null，以及預設值。我們將在 3.7.4 節中，作這一部份的介紹。

(5) 資料庫系統實作(Database system implementation)

在這一階段中，我們將「內部綱要」，即「低階實體資料模型」，藉由 SQL 2000 建立了所有的系統資料庫，並且輸入資料至資料庫中。至此，資料庫建設完成，我們將再第四章系統建製時再作介紹。

3.6.1 資料需求

在我們實際設計一個資料庫之前，我們必須儘量詳細的了解資料庫的使用者與預期資料庫使用者的期望，做好資料的收集與分析。因此，我們必須仔細的收集各方資料，研究分析相關的現存文件探討未來系統的輸入、輸出資料，找出整個系統所需的資料。

在系統中，我們將資料庫分成許多部分，有「課程」資料庫和「學生課程」資料庫等。其中，因為「課程」資料庫是原本學校就有的，將來只要將重新設計的「課程查詢系統」直接進入學校內部的課程資料庫即可。因此，原本想直接模擬逢甲大學的課程資料庫，但由於逢甲大學資訊處認為，課程資料庫的實際儲存格式，是項重要資訊，不可外洩，所以我們只好重新收集、分析「課程查詢子系統」、「預選課表子系統」所需的資料，並參照收集到的「學生資料建檔及整合應用規劃」、「逢甲大學各系、所、教學組開課一覽表」、「學生個人選課資料一覽表」等現存文件，自行重新規劃一個「課程」資料庫。

下列即為「課程」資料庫的「資料需求」描述：

- a. 每個系都有個系主任，且一個系只有一個系主任。
- b. 每個系有自己的專有教師即分機號碼，每個老師都只能歸屬一個系所有。

- c. 每個班級有一位導師，負責各班學生的生活狀況。
- d. 每個班級有所屬的系所、學院、班別，所謂的班別即為：
大學部、碩士班、碩士在職專班、博士班、海外青年訓練班、大學學分班等。
- e. 研究生可以擔任某些科目的助教，有其實驗室的分機號碼。
- f. 研究生或教師在期中考、期末考、畢業提前考時擔任監考人員。
- g. 教師必會教授某些科目。
- h. 每位學生都有自己所屬的班級及序號，此部分由註冊組負責，在每學期開始前由系統產生。
- i. 每位學生有其各自的修課紀錄，包含各科上課座號，各期中、期末、畢業提前考扣考狀況。
- j. 科目會由班級或課程開課，而每個課程有各自所屬的課程別，例如：「投資與理財」是由「通識-商管經濟」這門課程所開，而此門課程所屬的課程別為「通識核心課程」。
- k. 每個科目都有其上課教室與上課時間。
- l. 每間教室會有所在的大樓樓層與所屬的教室別，所謂的教室別為：多媒體教室、英語聽講訓練室、實習機房等。
- m. 每位學生有其學號、身分證字號、姓名、性別等資料。

- n. 每位教師有教師編號、身分證字號、姓名、性別、職稱、請益時間、分機號碼、個人首頁、E-mail 等資料。
- o. 每間教室有其可容納人數與教室設備等資料。
- p. 每項課目有下列資料：選課代號、科目名稱、預收人數、實收人數、開課時間、開課別、修習別、學分數、限修條件、先修科目、期中考、期末考、畢業提前考的日期與範圍、語言費、網路選課、及每週上課內容、課程大綱、教學目標、授課方式、成績評定等授課概要。

3.6.2 實體關係模型 ER-Model

當完成第一階段「需求的收集與分析」後，我們就要進行第二階段「概念資料庫設計」，將系統的「資料需求」以「高階概念資料模型」來表示。

「高階概念資料模型」包括了資料型態、關係、限制的詳細描述，但並包括任何實作的細節，所以通常較容易被了解，並可用來與非技術性的使用者作溝通，以確保滿足所有的使用者需求，避免互相衝突的使用者需求。又因不用關心資料儲存的細節，所以資料庫設計師可以專注於資料需求上，如此較易做好概念資料庫設計。

目前，已有很多「高階概念資料模型」被提出來，如：

實體關係模型(Entity-Relationship Model, 簡稱: ER Model)、
加強型實體關係模型(Enhanced-ER, 簡稱: EER Model)、
函數資料模型(Functional data Model, 簡稱: FDM)、
巢狀資料模型(Nested Relational Model)、
結構資料模型(Structure Data Model)、
語意資料模型(Semantic Data Model, 簡稱: SDM)……等等。
在此, 我們選擇最熟悉亦普遍被使用的 ER Model。來作概念資料庫設計。

ER Model 包含了實體狀態、關係狀態、屬性、鍵值、和結構限制, 下面我們就簡單介紹 ER Model 各元件, 及其圖示。

實體(Entity): 在真實世界中獨立存在的一個「事物」。如: 職員、房子、車、公司、工作、大學課程……等等。實體的代表圖示為單實線矩形方塊。實體名稱標在矩形方塊中央。

屬性(Attribute): 用來描述實體的特殊性質。例如, 一個實體為職員的屬性有姓名、年齡、地址、薪水和職務等。而每個屬性都有值(Value), 是儲存在資料庫的主要部分, 例如, 屬性為“姓名”、“年齡”者, 其值分別為“李蛋蛋”、“55”或“李阿修”、“66”。屬性的代表圖示為單實線橢圓形, 屬性名稱標在橢圓形中央。

此外，屬性又可分為下列幾種型態：

簡單屬性(Simple Attribute)：為不可分割的屬性，例如：年齡。

複合屬性(Composite Attribute)：為簡單屬性的組合，例如，“上課時間”是一個複合屬性，他可以拆成兩個簡單屬性，分別為“星期”、“節次”。複合屬性的代表圖示為單實線橢圓形，複合屬性名稱標在橢圓形中央，並以直線連結組合複合屬性的各位簡單屬性。

單值屬性(Single-valued Attribute)：屬性有一個以上的值。例如，一個人可能有一個 E-mail，也有可能沒有，也有可能兩個以上，因此，“E-mail”這個屬性就是屬於多值屬性。多值屬性的代表圖示為雙實線橢圓形，多值屬性名稱標在橢圓形中央。

儲存屬性(Stored Attribute)：屬性的值不能由其他屬性的值導出。如：出生日期。

導出屬性(Derived)：屬性的值可以從儲存屬性的值導出。例如，年齡屬性的值可以從出生日期屬性的值導出，因此，年齡即為導出屬性，出生日期就是儲存屬性。導出屬性的代表圖示為單虛線橢圓形，導出屬性名稱標在橢圓形中央。

鏈值屬性(Key Attribute)：屬性的每一個值在其實體中均是唯一的，分別代表實體中的每一個個別成員。有時符合此定義的屬性並不只一個，我們必須從中挑選一個來當作鏈值屬性。例如，“學生”實體的屬性“學號”與“身分證字號”，均符合鏈值屬性的定義，但我們只能選擇其中一個，作為“學生”實體的鏈值屬性，通常，我們都會以“學號”當作“學生”實體的鏈值屬性。鏈值屬性的代表圖示為單實線橢圓形，鏈值屬性名稱加上實線底線後，標在橢圓形中央。

關係(Relationship)：兩個實體間有關聯時，就產生了關係。例如“教師”跟“科目”兩個實體間有關聯，便產生了“授課”關係。關係的代表圖示為單實線菱形方塊，關係名稱標在菱形方塊中央。

角色名稱(Role Name)：給個實體名均可當作角色名稱，此時就不用特別標出來。但某些情況下，角色名稱會與實體名稱不一樣，此時，就要在實體與關係之間的直線附近，標明角色名稱，如圖 xxx。例如，在“教師” ”管理” “班級” 這個關係中，“教師”所扮演的角色名稱為“導師”。

基數率(Cardinality Ratio)：以 1 : 1、1 : N、M : N 表示兩實體在此關係中所佔的比例。例如，在“職員” “管理” “部門”

的關係中，若“職員”實體：“部門”實體=1：1時，表示各位職員只能管理一個部門，每個部門只能被一位職員管理，在 ER Model 圖示中，我們便在“職員”實體與“管理”關係的直線附近標上“1”，也在“部門”實體與“管理”關係的直線附近標上“1”，來表示此種狀態，如圖 xxx。若“職員”實體：“部門”實體=1：N時，表示各位職員可以管理多個部門，但每個部門只能被一位職員管理，在 ER Model 圖示中，我們便在“職員”實體與“管理”關係的直線附近標上“1”，在“部門”實體與“管理”關係的直線附近標上“N”，來表示此種狀態，如圖 xxx。若“職員”實體：“部門”實體=M：N時，表示每個職員可以管理多個部門，而每個部門也能被多位職員管理，在 ER Model 圖示中，我們便在“職員”實體與“管理”關係的直線附近標上“M”，在“部門”實體與“管理”關係的直線附近標上“N”，來表示此種狀態。

參與限制(Participation Constraint)：以“全部參與”及“部分參與”來表示實體中的個別成員參與關係的程度。例如，在“教師”“管理”“班級”的關係中，若每位教師都要擔任導師管理班級的話，那麼對“教師”這個實體而言，就是“全部參與”此關係，在 ER Model 圖示中，我們便以“雙實線”連接“教

師”實體與“管理”關係；若並非各位教師都要擔任管理班級的話，那麼對“教師”這個實體而言，就是“部分參與”此關係，在 ER Model 圖示中，我們便以“單實線”連接“教師”實體與“管理”關係。

弱勢實體(Weak Entity)：沒有任何鍵值屬性的實體。弱勢實體必須透過關係與另一個實體結合，才能分辨弱勢實體中的每一個個別成員。而上述的「另一個實體」便稱為「辨認擁有者」(Identifying Owner)。例如，弱勢團體「親屬」必須與辨認擁有者「學生」產生關聯，才有意義。弱勢團體的代表圖視為雙實線矩形方塊，弱勢實體名稱標在矩形方塊中央。

辨認關係(Identifying Relationship)：弱勢實體與辨認擁有者所產生的關係。由於，弱勢實體必須與辨認擁有者產生關聯，才有意義，因此弱勢實體必“全部參與”其所對應的辨認關係。例如，“學生” “有” “親屬”的關聯中，“有”即為其“辨認關係”，而“親屬”必全部參與“有”這個辨認關係。辨認關係的代表圖示為雙實線菱形方塊，辨認關係名稱標在方塊中央。

部分鍵值屬性(Partial Attribute)：為弱勢實體所擁有。當辨認擁有者成員為同一個時，我們可藉由部分鍵值來分辨不同的弱勢實體成員。例如：若同一學生的親屬間，不會有相同名字時，

則弱勢實體“親屬”的“姓名”屬性即為部分鍵值屬性。部分鍵值屬性的代表圖示為單實線橢圓形，部分鍵值屬性名稱加上虛線底線後，標在橢圓形中央。

最後，將我們根據上一節的資料需求，所繪出的「線上學生課務資訊系統」資料庫的 ER Model 附上。

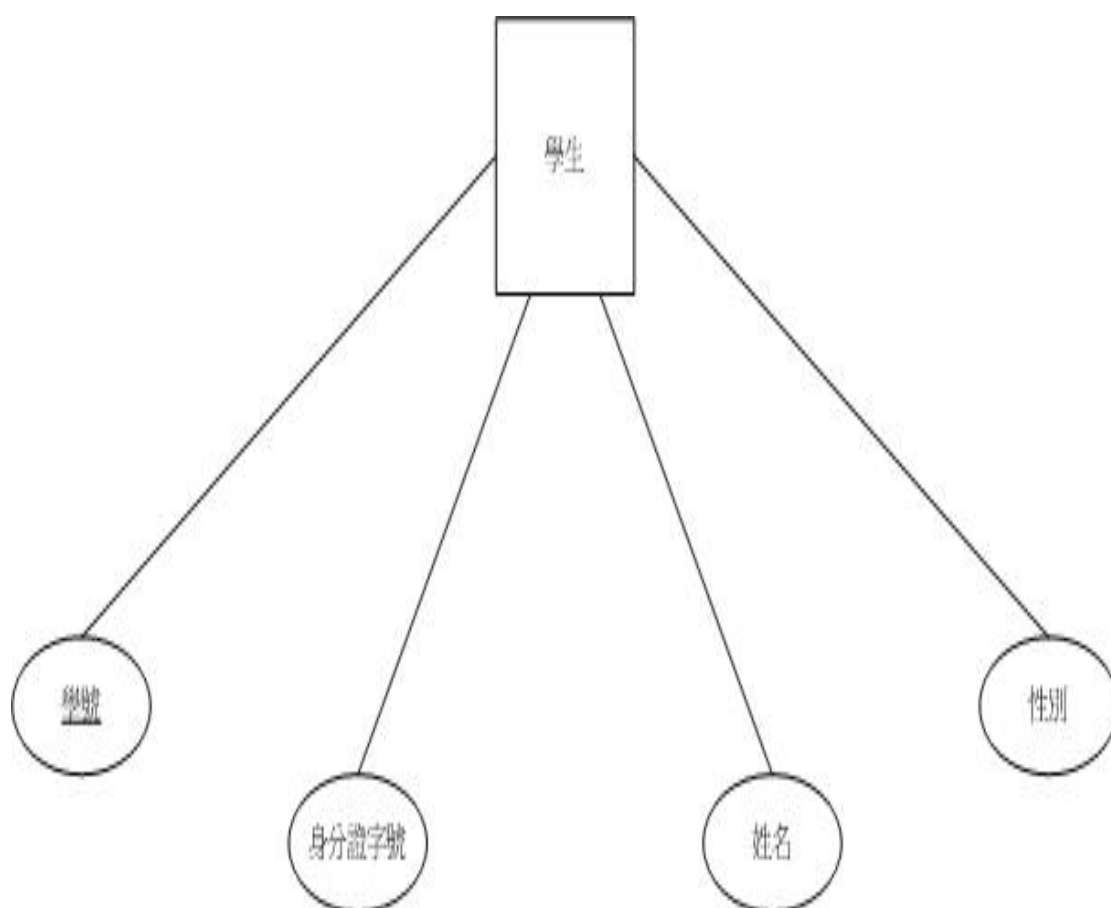


圖 3.1、課程資料庫-學生實體圖

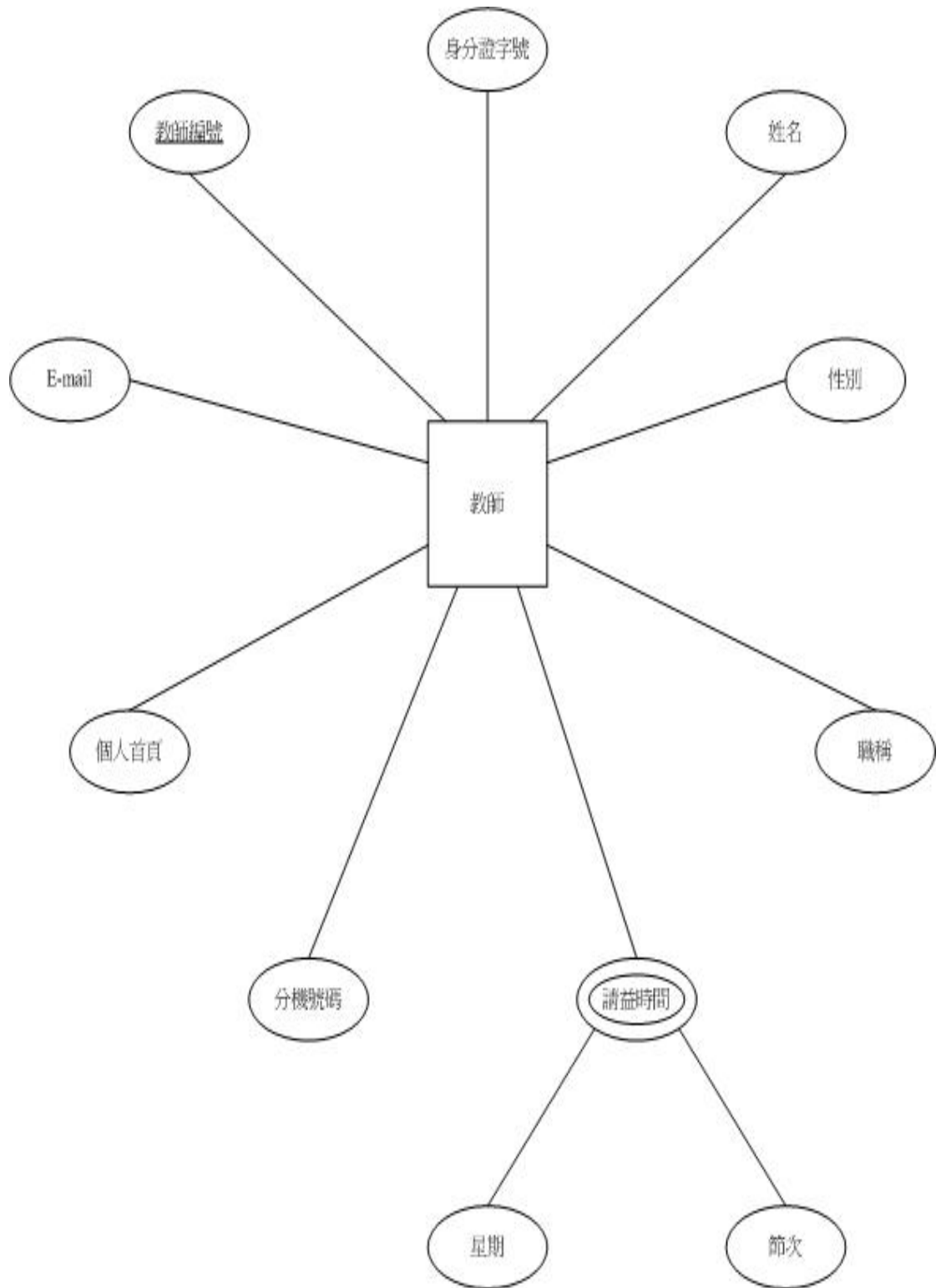


圖 3.2、課程資料庫-教師實體圖

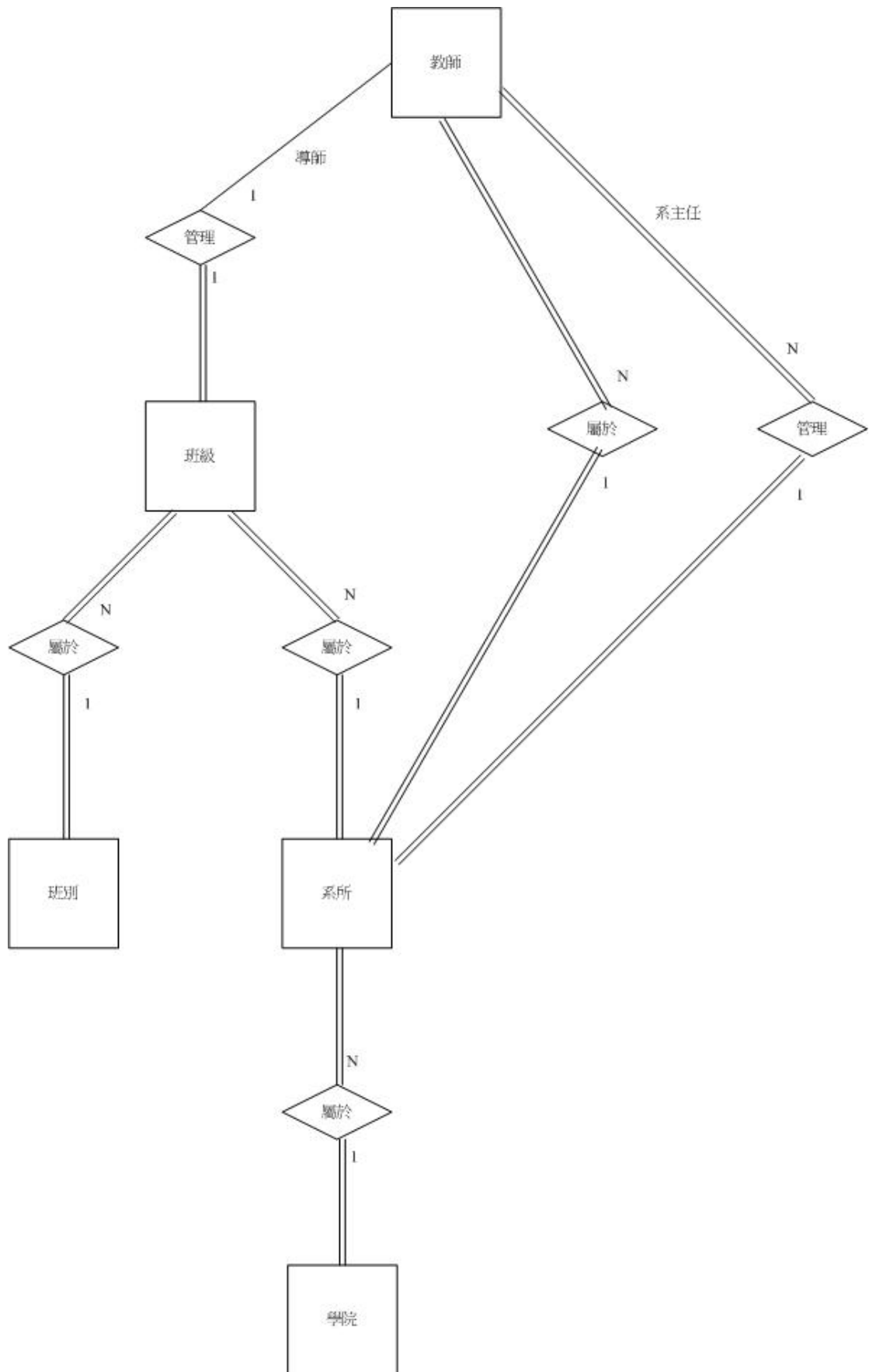


圖 3.3、課程資料庫-教師-學院

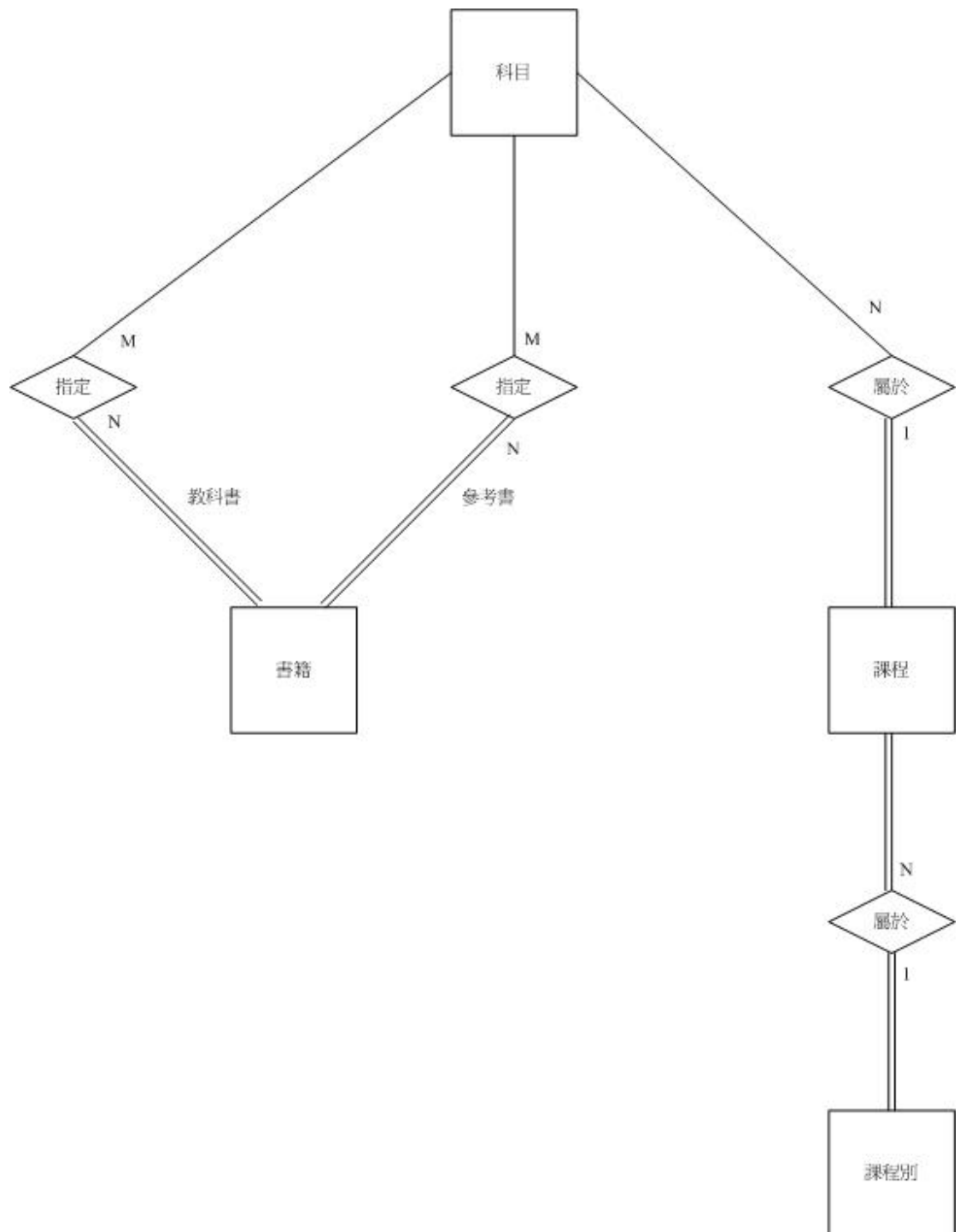


圖 3.4、課程資料庫-科目-課程別

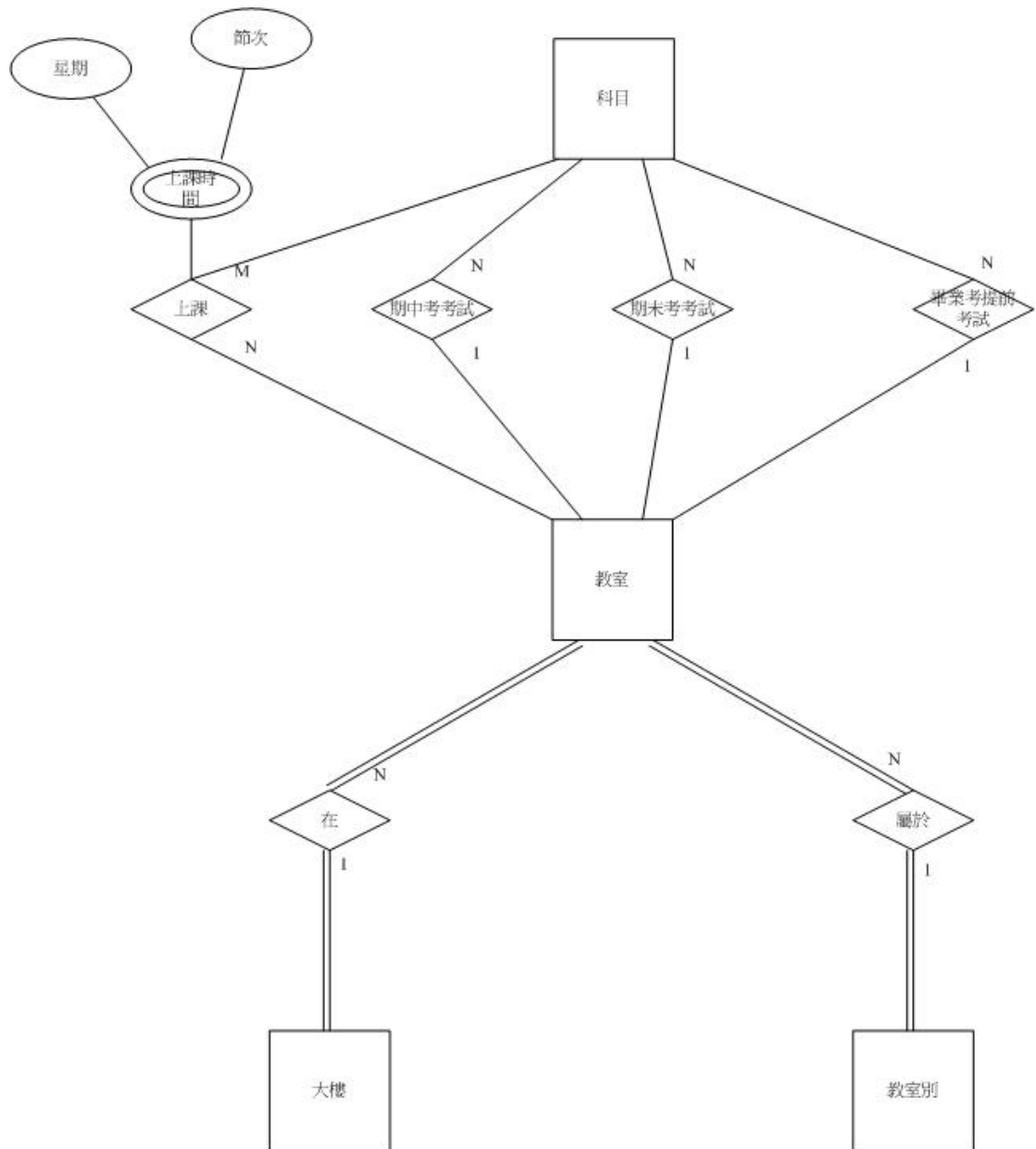


圖 3.5、課程資料庫-科目-教室別

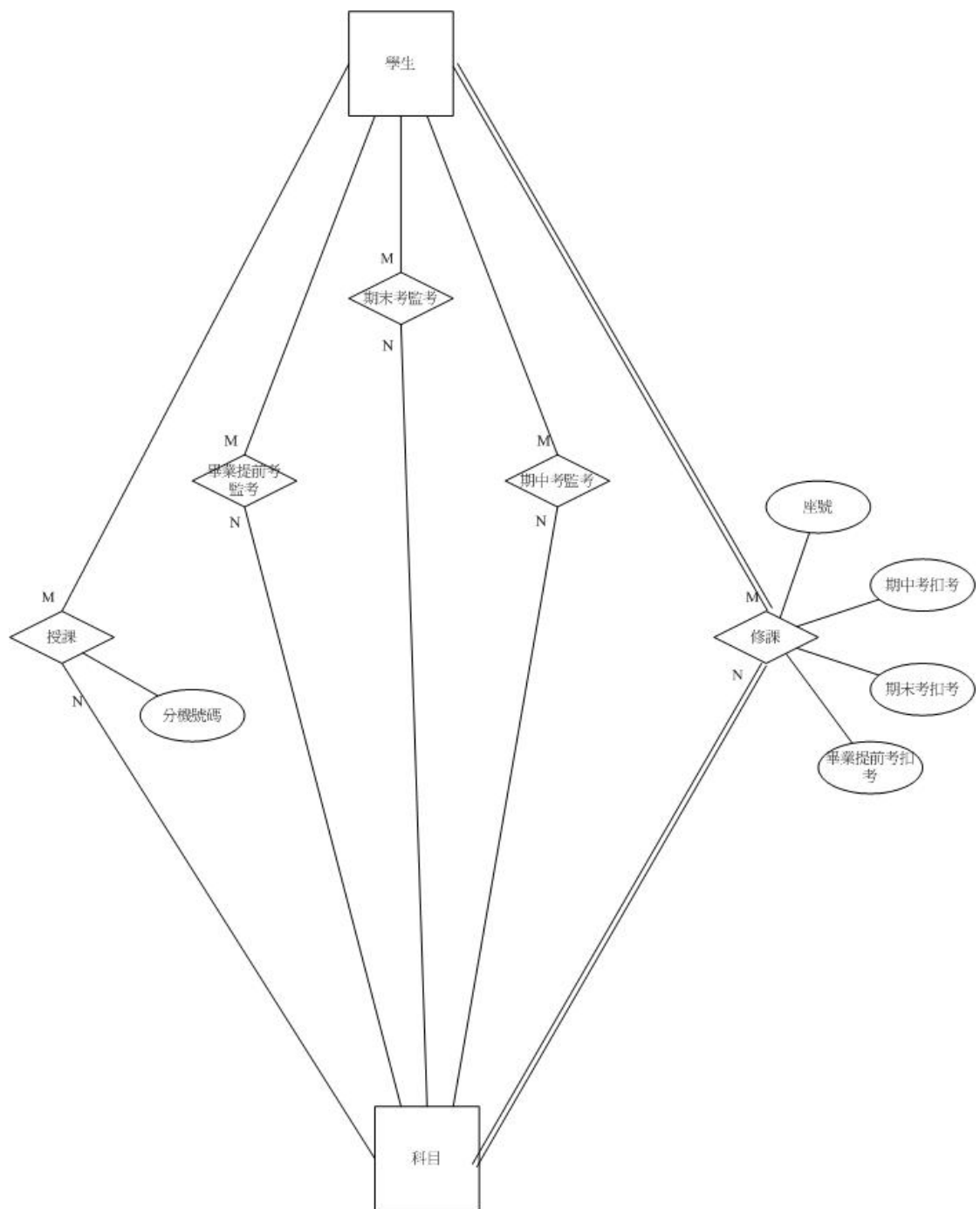


圖 3.6、課程資料庫-學生-課程

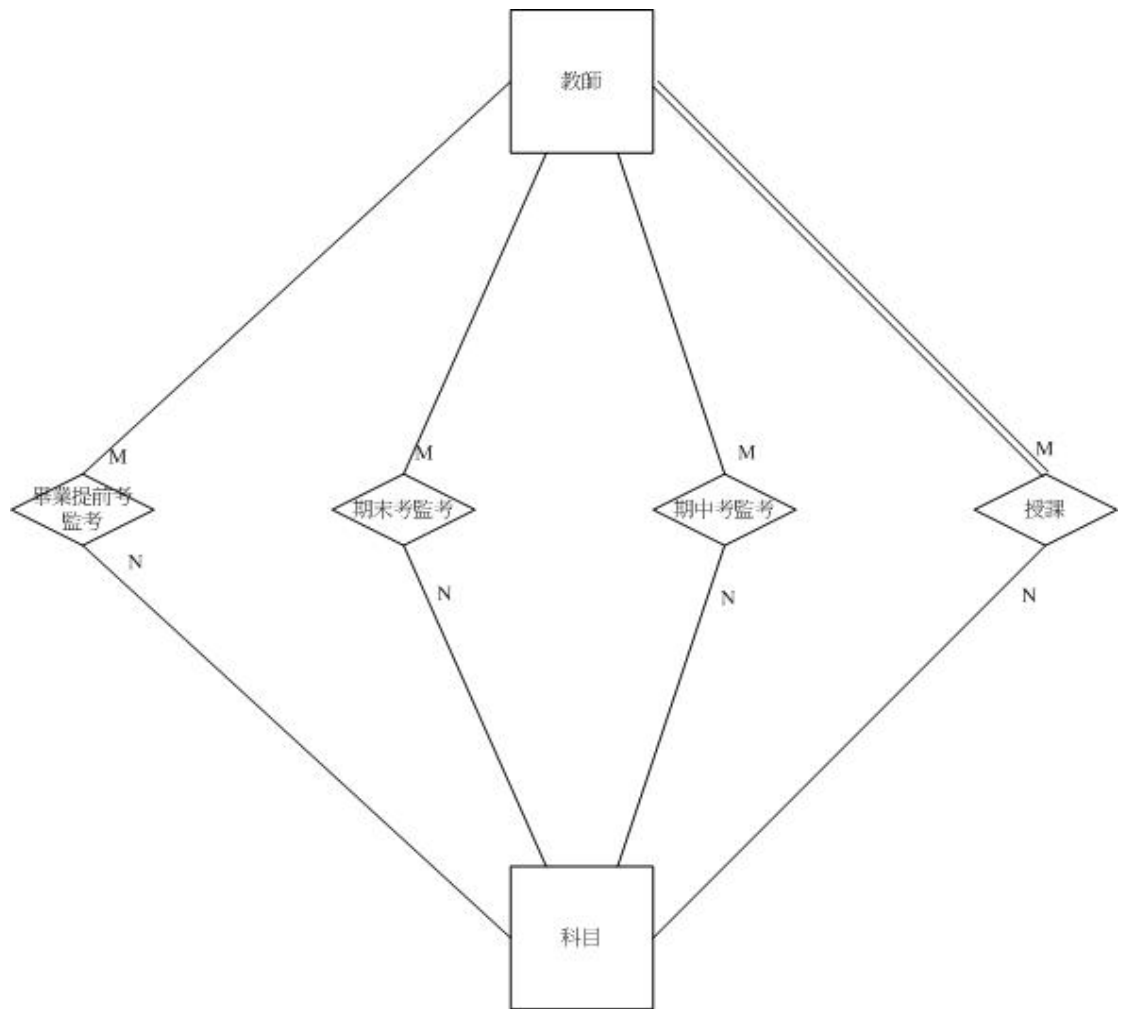


圖 3.7、課程資料庫-教師-科目

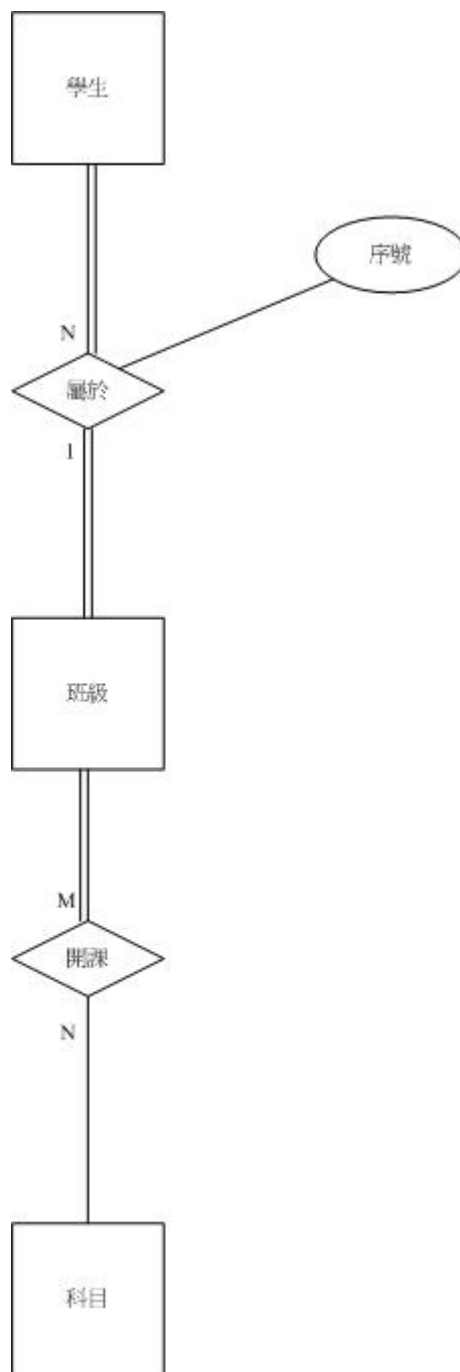


圖 3.8、課程資料庫-學生-科目 2

3.6.3 關聯式資料庫綱要

資料庫設計的第三階段為「邏輯資料庫設計」，此一階段的任務是將第二階段的「高階概念資料模型」—「ER Model」轉化成「概念綱要」—「關聯式資料庫綱要」。

首先，我們先介紹將「ER Model」轉化成「關聯式資料庫綱要」的步驟，最後再附上「現在學生課務資訊系統」資料庫的「關聯式資料庫綱要」。

現在，我們概略描述將「ER Model」轉化成「關聯式資料庫綱要」的六個步驟。

步驟一：為 ER Model 中的「實體」E，建立一個「實體關聯」ER，其中包含了實體 E 中的所有簡單屬性 A(A1、A2、A3)。將實體 E 中的「鏈值屬性」KA 當作實體關聯 ER 的「主鍵」。

步驟二：為 ER Model 中的「弱勢實體」WE，建立一個「實體關聯」WER，其中包含了弱勢實體 WE 中的所有簡單屬性 A(A1、A2、A3)。實體關聯 WER 的「主鍵」為弱勢實體 WE 的「辨認擁有者之鏈值屬性」IKA 與「弱勢實體 WE 之部分鏈值屬性」PKA 的組合。而「辨認擁有者之鏈值屬性」IKA 亦為死實體關聯 WER 的「外鍵」。

步驟三：在 ER Model 中，若基數率為 $E1 : E2 = 1 : 1$ 者，且 E1 的參與限制為完全參與的話，就將 E2 的鏈值屬性放入 E1 的實

體關聯中，當作外鍵。然後，將關係上的簡單屬性，全放入 E1 的實體關聯中。

步驟四：在 ER Model 中，若基數率為 $E1 : E2 = 1 : N$ 者，將 E2 的鏈值屬性放入 E1 的實體關聯中，當作外鍵。然後，將關係上的簡單屬性，全放入 E1 的實體關聯中。

步驟五：在 ER Model 中，若基數率為 $E1 : E2 = M : N$ 者，就建立一個「關係關聯」RR。將「實體 E1 的鏈值屬性」E1KA 與「實體 E2 的鏈值屬性」E2KA 組合起來，作為關係關聯 RR 的「主鍵」。然後，將關係上的簡單屬性 RA(RA1、RA2、RA3)，全放入關聯 RR 中。而「實體 E1 的鏈值屬性」E1KA 與「實體 E2 的鏈值屬性」E2KA 亦為此關聯 RR 的「外鍵」。

步驟六：為 ER Model 中的每一個多值屬性 MA，建立一個新的關聯 MAR。而關聯 MAR 的「主鍵」為擁有此多值屬性 MA 的實體或關係的鏈值屬性，為此多值屬性 MA 的組合。

以上為「ER Model」轉化成「關聯式資料庫綱要」的六個步驟，最後，附上我們在這一階段所作的「選課系統」資料庫的「關聯式資料庫綱要」。

3.6.4 內部綱要

資料庫設計的第四階段是實體資料庫設計，此一階段中，我們
要將「關聯式資料庫綱要」轉化成「內部綱要」，即「低階實體資
料模型」。

在此，我們根據 SQL2000 所提供的資料型別，即其內部儲存格
式定義了各資料行的資料型別、長度、是否允許 NULL，以及預設
值。下面我們就列出「線上學生課務資訊系統」資料庫的內部綱
要。

課程資料庫

	資料行名稱	資料型別	長度	是否允許 Null
🔑	學號	nchar	8	
	身分證字號	nchar	10	
	姓名	nvarchar	10	
	性別	bit	1	
	班級編號	nchar	8	
	序號	nchar	2	

圖 3.9、學生內部資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	教師編號	nchar	6	
	系所編號	nchar	2	✓
	職稱	nchar	1	
	身分證字號	nchar	10	
	姓名	nvarchar	10	
	性別	bit	1	
	分機號碼	nchar	4	✓
	個人首頁	nvarchar	50	✓
	[E-mail]	nvarchar	50	✓

圖 3.10、教師內部資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	科目編號	nchar	7	
	選課代號	nchar	4	
	學年度	nchar	2	
	學期	nchar	1	
	科目名稱	nvarchar	50	
	課程編號	nchar	4	✓
	開課別	nchar	1	
	修習別	bit	1	
	學分	nchar	1	
	預收人數	nchar	3	
	實收人數	nchar	3	
	網路選課	bit	1	
	語言費	bit	1	
	先修科目	nvarchar	100	✓
	限制條件	nvarchar	100	✓
	備註	nvarchar	100	✓
	期中考	bit	1	
	期中考日期	smalldatetime	4	✓
	期中考範圍	nvarchar	200	✓
	期中考教室編號	nchar	7	✓
	期末考	bit	1	
	期末考日期	smalldatetime	4	✓
	期末考範圍	nvarchar	200	✓
	期末考教室編號	nchar	7	✓
	畢業提前考	bit	1	
	畢業提前考日期	smalldatetime	4	✓
	畢業提前考範圍	nvarchar	200	✓
	畢業提前考教室編號	nchar	7	✓
	課程大綱	nvarchar	500	✓
	教學目標	nvarchar	500	✓

圖 3.11、科目資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	編號	nchar	8	
	學年度	nchar	2	
	學期	nchar	1	
	年	nchar	1	
	班組	nvarchar	20	
	導師編號	nchar	6	
	系所編號	nchar	2	
	班別編號	nchar	1	
	班級簡稱	nchar	4	

圖 3.12、班級資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	編號	nchar	1	
	名稱	nvarchar	20	

圖 3.13、教室別資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	編號	nchar	10	
	書名	nvarchar	100	
	作者	nvarchar	100	
	出版社	nvarchar	50	

圖 3.14、書籍資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	編號	nchar	4	
	名稱	nvarchar	20	
	課程別編號	nchar	2	

圖 3.16、課程資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	編號	nchar	2	
	名稱	nvarchar	20	

圖 3.17、學院資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	編號	nchar	7	
	名稱	nvarchar	15	
	教室別編號	nchar	1	
	大樓編號	nchar	2	
	樓層	nchar	2	
	設備	nvarchar	200	✓
	容納人數	nchar	3	

圖 3.18、教室資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	編號	nchar	2	
	名稱	nvarchar	20	
	地上樓層數	nchar	2	
	地下樓層數	nchar	2	

圖 3.19、大樓資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	學號	nchar	8	
🔑	科目名稱	nchar	7	
	座號	nchar	3	
	期中考扣考	bit	1	
	期末考扣考	bit	1	
	畢業提前考扣考	bit	1	

圖 3.20、學生修習科目資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	學號	nchar	8	
🔑	科目編號	nchar	7	

圖 3.21、學生期中考監考科目資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	學號	nchar	8	
🔑	科目編號	nchar	7	

圖 3.22、學生期末考監考科目資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	學號	nchar	8	
🔑	科目編號	nchar	7	

圖 3.23、學生畢業提前考監考科目資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	學號	nchar	8	
🔑	科目編號	nchar	7	
	分機號碼	nchar	4	

圖 3.24、學生授課科目資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	教師編號	nchar	6	
🔑	科目編號	nchar	7	

圖 3.25、教師期中考監考科目資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	教師編號	nchar	6	
🔑	科目編號	nchar	7	

圖 3.26、教師期末考監考科目資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	教師編號	nchar	6	
🔑	科目編號	nchar	7	

圖 3.27、教師畢業提前考監考科目資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	教師編號	nchar	6	
🔑	科目編號	nchar	7	

圖 3.28、教師授課科目資料表

	資料行名稱	資料型別	長度	是否允許 Null
▶🔑	班級編號	nchar	8	
🔑	科目編號	nchar	7	

圖 3.29、班級開課科目資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	科目編號	nchar	7	
🔑	教科書編號	nchar	10	

圖 3.30、科目指定教科書資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	科目編號	nchar	7	
🔑	教室編號	nchar	7	
🔑	上課星期	nchar	1	
🔑	上課節次	nchar	2	

圖 3.31、科目上課教室資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	教師編號	nchar	6	
🔑	星期	nchar	1	
🔑	節次	nchar	2	

圖 3.32、教師請益時間資料表

	資料行名稱	資料型別	長度	是否允許 Null
🔑	科目編號	nchar	7	
🔑	週次	nchar	2	
	內容	nvarchar	100	

圖 3.33、科目每週上課內容資料表

第四章 系統建置

4.1 登入子系統

主要功能：

顯示最新消息

登入

選擇身分

(1) 顯示最新消息：

系統自動輸出最新三筆消息

(2) 登入：

輸入：使用者帳號及密碼

輸出：主畫面或是登入失敗

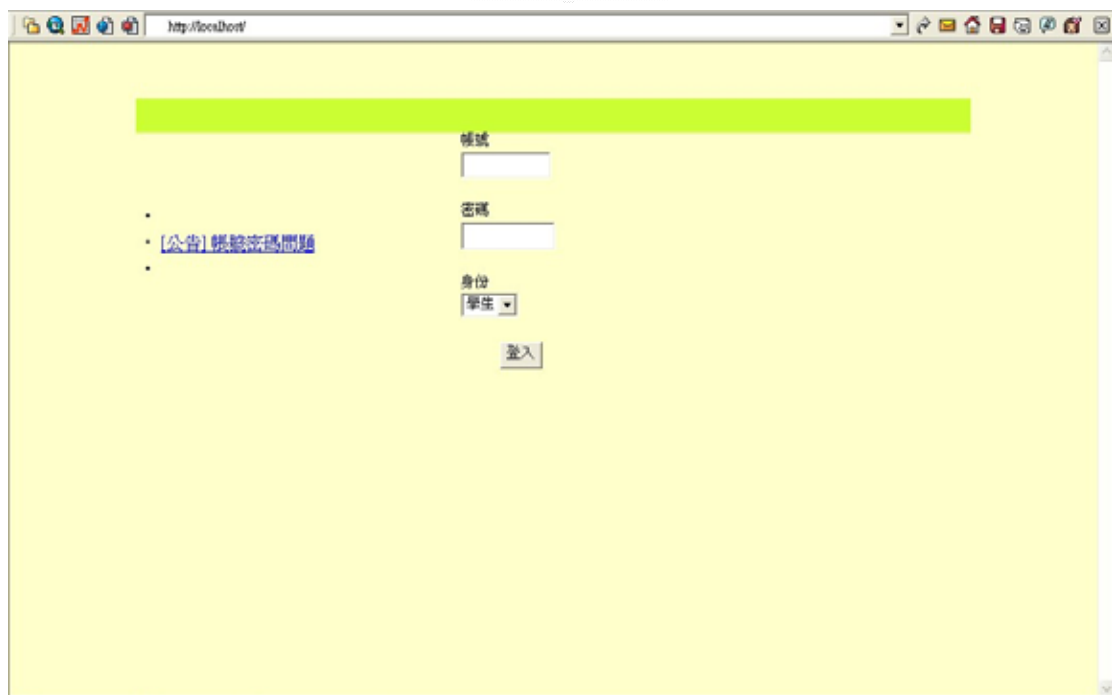


圖 4.1 登入子系統

主要功能：

使用者再次登入

密碼查詢

(1) 使用者再次登入：

輸入：使用者帳號及密碼

輸出：主畫面或網頁首頁

(2) 密碼查詢：

輸入：使用者帳號及申請帳號時信箱

輸出：密碼以寄送自信箱



圖 4.2 登入失敗

4.2 選課子系統、課程查詢子系統

主要功能：

選課

查詢系上開課課程

(1) 選課：

輸入：點選學生欲選課之課程

輸出：已加入課程中

課程已滿

(2) 查詢繫上開課課程

輸入：在系上開課點選

輸出：系上開課課別表



圖 4.3 主畫面

4.3 學生資料子系統

主要功能：

更改使用者個人資料

(1) 更改使用者個人資料：

輸入：密碼等事項

輸出：更改成功

每個項目皆要填寫

http://localhost/

我的課程 個人區 系統區
說明 > 登出/回首頁

個人設定	
帳號	d8901893
密碼	<input type="text"/>
確認密碼	<input type="text"/>
姓名*	李任修
性別*	<input type="radio"/> 女生 <input checked="" type="radio"/> 男生
生日	西元 1982 年 11 月 20 日
身分證或護照號碼	<input type="text"/>
E-mail*	d8901893@knight.fcu.
Homepage	<input type="text"/>
地址	<input type="text"/>
職業/系級	<input type="text"/>
單位	<input type="text"/>

儲存 重來

圖 4.4 個人設定

4.4 行事曆子系統

主要功能：

顯示行事曆事項

新增/編輯/刪除行事曆事項

(1) 顯示行事曆事項：

輸入：選擇年月

輸出：當月行事曆

(2) 新增/編輯/刪除行事曆事項

輸入：點選欲增加行事曆日期

輸出：行事曆內容

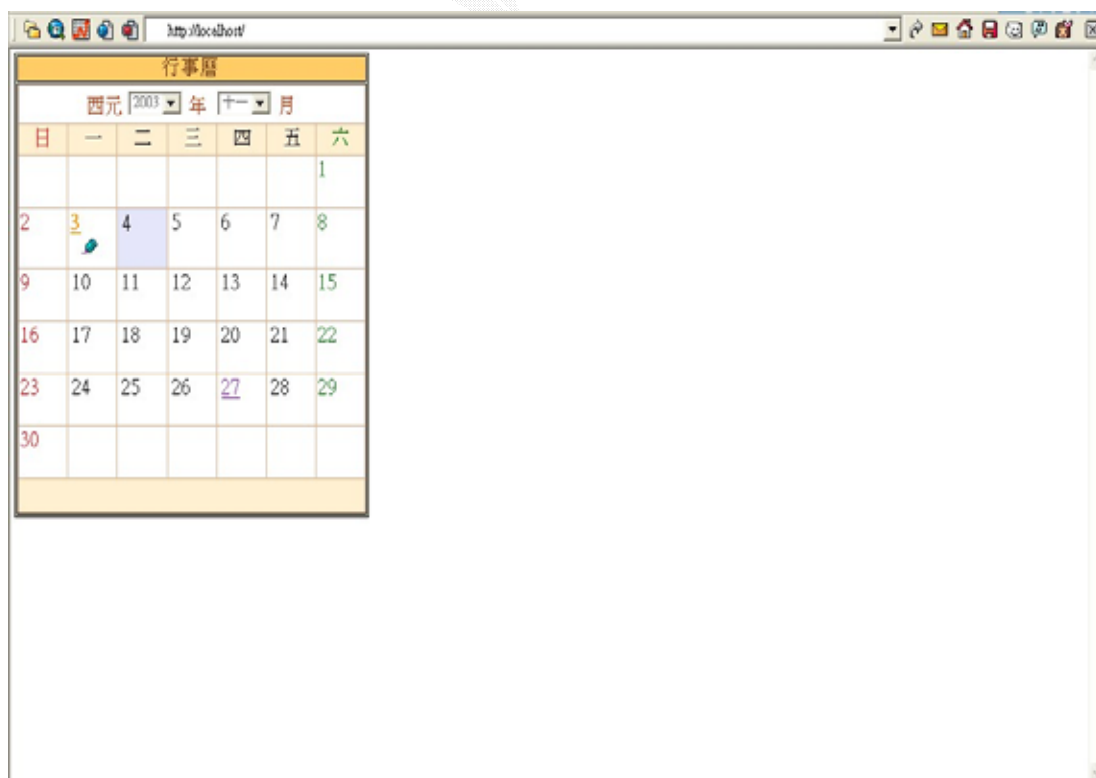


圖 4.5 行事曆

4.5 記事本子系統

主要功能：

提醒使用者

(1)提醒使用者：

輸入：使用者點選增加

輸出：使用者填寫記事畫面



圖 4.6 記事本

第五章 系統評估

5.1 選課系統

選課系統在最後完成的時候卻與課程查詢系統結合，是當初沒有想過的事情，然後在經過許多的思考後也覺得如果整合選課系統與課程查詢系統是個比較好的方向，畢竟每過一個年級資料庫就會將資料更新，學校也可以事先將每科科目先擬定給每位學生，再由學生選擇不想修的科目退選，省時也可以少了許多麻煩。

5.2 學生資料庫系統

學生資料庫系統在後來的時候，發現到許多沒用到的欄位，因此把他最佳化到目前的狀態，可說是有利有弊，而且如果把課程資料等也一併放入學生資料庫中將會造成兩邊資料重複，反而有浪費空間的感覺。

5.3 行事曆系統

行事曆系統在目前來說，是絕得做的比較好的系統，雖然功能沒有當初想的強大，例如鬧鐘等，但是在實用性上的功能可說是滿意了，只是還差在畫面不夠美化。

5.4 記事本系統

記事本系統並沒有做到當初所構想的詳盡，並不能將記事本與行事曆整合是最大的遺憾，記事本的模式也有點簡陋，可說是經驗不夠因

此不能完成的系統之一。

5.5 登入子系統

登入系統達到了他所應該擁有的功能，只是在當初也沒有把登入系統的介面構思的很美麗，因此只能有簡略的畫面。

5.6 教師子系統

由於老師隱私以及上課方式的問題，教師子系統方面並沒有辦法順利完成，實在是一種遺憾。

5.7 作正確的事

系統分析

在開始這個專題的時候，做了許久的系統分析，雖然花了許多的時間，但也對整個系統可說是非常了解，到最後要將功能整合也可以順利的完成，這都要多虧系統分析的福，而在資料庫系統上，選擇 SQL SEVER2000 開發，雖然剛開始的時候遇到了許多的難關，如語法必需自己一個一個打不如 POWER BUILDER 來的方便，但在克服難關後總算也得到了不少的幫助，在跟 ASP.NET 網頁結合就方便了許多。

ASP.NET

選擇 ASP.NET 開發網頁，也可說是正確的決定，畢竟現在 .NET 的技術也越來越成熟，而微軟在目前的時代中也算是最普遍的作業系統，因此用 ASP.NET 開發出來的程式，還不會被大家所不能接受。

5.8 需加強的事

時間的掌握

由於經驗的不夠，在系統分析上的時間還是佔了太久的時間，就算是在後期了也還有一部分在做系統分析，這樣造成了許多部分不能夠一下子整合，必須拖更久的時間之後才能整合，導致於某些功能不能完成，最後的工作也是在熬夜之中完成。

功能不夠完善

在專題中提到了許多的功能，然而有些功能卻無法順利的完成，不外乎程式能力方面的不足，以及影響系統的能力等而被否決掉，最好的例子就是鬧鐘，因為怕造成使用端的不安全因此就把功能否決掉了。

介面不夠美觀

因為大家對於美術沒有什麼概念，組員也都是主要希望完成功能性，因此在介面上面絕得不能放上檯面，在時間上也沒有辦法有最後的時間更改介面的感覺。

安全性

選課系統最怕的就是安全性，由於能力不夠無法思考許多有關於封包的安全性等問題，在實作的時候就沒有考慮安全性，這點需要非常的改進。

第六章 心得報告

6.1 任修

開始這份專題，是在兩年前開始構思，在這之前對於專題題目也沒有什麼方向，然而在因緣際會之下接觸到了電子學習系統，因此對這方面產生興趣，另一方面也覺得為什麼資訊系沒有自己的電子學習系統，然而在實作完之後發現到這個程式還不能說是完整的電子學習系統，只是以後有了這個經驗必定可以完成一個標準的電子學習系統的。

在我們這一組中，由於算是小組的隊長，因此也承受了更大的壓力，想要把這份專題做好，想要把整個組在自己規劃的時間內完成該做的事項，這一段時間真的可以說是很痛苦，然而完成之後，痛苦可以說是減輕了，回想這段時間，卻又覺得有那麼點懷念，組員們在一起認真的討論專題的模樣，一個系統初步完成的感動，都深深地烙印在自己的腦海中，也得到了許多的經驗，如果在配合系上所教導我們的新知識，相信如果還有機會碰到類似的專題，一定可以得心應手的完成它。

另一份收穫就是為了完成這份專題，待在書店裡一直找書、買書的回憶也令人記憶猶新，因為這段時間之類閱讀了許多專業的書籍，也讓自己在專業知識上面提升了不少，讓自己看起來也特別的有書卷氣息，等以後需要用到時也不怕書到用時方恨少了。

最後要感謝我們的指導教授—謝信芳老師，在我們完成這段專題的時間之類，老師幫助了我們許多，不論是在點醒我們方向，或是系統分析上面所遇到的困難等，都是由於老師的幫忙才讓我們順利通過難關，也從老師身上學到了許多團隊合作的道理。還要感謝自己的組員們，由於有他們分工合作的幫忙和互相交流，才能完成這偉大的任務。

6.2 建達

寫完這次的專題，感覺自己受益不少，若沒有專題的催促，強迫自己學習，也無法讓自己看了這麼多書，自己讀自己理解自己親手把程式寫出來，那種感覺很快樂，可以一直保持下去，在程式方面一開始真的很痛苦，從零開始一點頭緒都沒有，還在連結資料庫的方面卡了很久，再來是著手伺服器端的程式光是畫哪些林林總總的圖也花了不少功夫，可惜不畫那些圖，未來的工作總無法順利進行，在有些期望的功能中，還有一些未能做的盡善盡美

合作的重要性也是我這次專題得到的重要心得跟組員互動有問題互相支援互相解決，在組員內找到各個專長的領域，並借重來完成這次的專題，自己也從中學習到別人領域中學習的方式跟精神，並學到許多解決問題的正面方法，積極面對問題，尋找解答的方式，最後感謝輔導老師熱心的支持與關心，鼓勵我們在挫折中找回信心。

6.3 昱翔

自從做專題以來也已經一年了，在這期間，有經歷過瓶頸，而且還要兼顧學業，再加上考慮畢業之後到底要往哪個方向走，還有看著身邊的同學，一個又一個的，說要下學期才發表專題，有些是因為懶惰，有些是因為覺得作不好，甚至還有一些，其實已經作的很好了，但是為了想要精益求精，所以才選擇下學期發表，到了前一陣子，問題一一浮現，不禁心煩意亂了起來，好顯有師長跟朋友的鼓勵還有安慰，讓我跟我的組員，有繼續做專題的動力。

終於，到了最關鍵的時刻，離專題發表的時間越來越接近，第一次的緊張感，又慢慢的浮上了檯面，深怕上台的時候會發表的不盡人意，總之，要努力獲得老師的肯定，才不會辜負了我們專題組員，這一年來的努力辛苦付出。

其實，專題越做到後來，才會發現系上為什麼把專題作成必修，因為從小組的研討，所作的東西逐項修正，所遭遇的挫折，如果都成功的一一克服，當而然之，出了社會，可以比別人多了一個經驗，比別的競爭者多了一份實力，到後來，獲利的還是我們學生。

這一年中，多謝支持我們，讓我們更有動力的師長朋友，讓我們學得更多經驗，知道做專題所能學得的東西。

參考資料

1. 李廷桂，SQL SEVER 2000 設計實務，高立，台北，2001
2. 孫三才，許薰伊，精通 C#與 ASP.NET 程式設計，學貫，台北，2002
3. 王國榮，ASP.NET 網頁製作教本旗標，台北，2003
4. 吳仁和，林信惠，系統分析與設計：理論與應用實務，智勝文化，台北，2002
5. 曹祖聖，吳明哲，黃世陽，林義証，蔡文龍，ASP.NET 實戰 For C#.NET

