

逢 甲 大 學

資 訊 工 程 學 系 專 題 報 告

軟 體 評 估 工 具



學 生：吳俊逸（四甲）
林志遠（四甲）

指 導 教 授：薛 念 林

中 華 民 國 九 十 二 年 十 二 月

目錄

目錄.....	1
圖表目錄.....	4
摘要.....	6
第一章 導 論.....	7
1.1 動機.....	7
1.2 系統目標.....	7
第二章 系統功能.....	8
2.1 JSP 網頁部分.....	8
2.2 APPLICATION 部分.....	8
2.3 CK Metrics.....	9
第三章 設計與建構.....	11
3.1 JSP 網頁部分.....	11
3.1.1 需求分析.....	11
3.1.2 實體架構.....	13
3.3.3 系統架構.....	14
3.1.4 所用工具介紹.....	15
3.1.5 學生註冊流程.....	16
3.1.6 使用者登錄流程.....	17
3.1.7 檔案上傳流程.....	18
3.1.8 密碼查詢.....	19
3.2 APPLICATION 部分.....	20
3.2.1 需求分析.....	20
3.2.2 設計架構.....	21
3.3.3 類別關係.....	22
3.3.4 類別規格.....	23
3.3 Metrics 計算部分.....	25
第四章 相關技術研究.....	29
JSP 網頁部分.....	29
4.1.1 JavaBean 介紹.....	29
4.2 APPLICATION 部分.....	32
4.2.1 JavaNCSS 介紹.....	32

4.3	Metrics 計算部分.....	34
4.3.1	JavaCC 介紹.....	34
4.3.2	Lexical analyzer.....	35
4.3.3	Parser.....	36
4.3.4	JJTree 和 JTB.....	37
4.3.5	Composite Design Pattern.....	38
4.3.6	Visitor Design Pattern.....	40
4.3.7	範例.....	41
第五章	系統評估與討論.....	46
5.1	遭遇困難.....	46
5.2	心得與經驗.....	47
參考資料	48
附錄	49
附錄 A	工作分配表.....	49
附錄 B	安裝步驟.....	50
附錄 C	執行結果.....	58



圖表目錄

(圖一 A class hierarchy)	9
(表一 JSP 網頁功能性需求)	11
(表二 JSP 網頁非功能性需求)	12
(圖二 實體架構圖)	13
(圖四 學生註冊流程圖)	16
(圖五 使用者登錄流程圖)	17
(圖七 密碼查詢流程圖)	19
(表三 Application 功能性需求)	20
(表四 Application 非功能性需求)	20
(圖八 類別關係圖)	22
(表五 Main 類別規格)	23
(表六 MainFrame 類別規格)	23
(表七 MainMenuBar 類別規格)	23
(表八 MainToolBar 類別規格)	24
(表九 NewFrame 類別規格)	24
(表十 選單功能規格)	24
(圖九 JJTree 所產生的類別圖)	26
(圖十 JavaCC 所產生的類別圖)	27
(圖十一 類別的繼承關係圖)	28
(圖十二 Bean 結構圖)	30
(圖十三 Token Manager)	35
(圖十四 Parser)	36
(圖十五 Composite Pattern 的類別圖)	38
(圖十六 Visitor Pattern 的類別圖)	40
(表十二 範例的類別)	41
(圖十七 程式範例的類別圖)	42
(圖十八 程式範例的流程圖)	44
(表十三 工作分配表)	49
(圖十九 執行畫面一)	58
(圖二十 執行畫面二)	58
(圖二十一 執行畫面三)	59
(圖二十二 執行畫面四)	59

(圖二十三 執行畫面五)..... 60



摘要

以往程式作業的評分，著重在功能性上的衡量——即功能做到了就可以得到比較高的分數，而不論其程式的結構好壞。

隨著軟體品質越來越為重視，學生的作業評分應將程式的品質列為重要的參考因素。

本專題針對物件導向程式設計實作一物件品質衡量的系統。老師可透過本系統觀看學生作業的各項物件品質指標，如 DIT、LCOM、WMC 等，以了解學生是否以合宜的物件結構來實作作業。

本系統以網頁應用程式 JSP 為介面，方便學生及老師上網交作業與改作業，其核心以 Java Bean 實作之，提供再使用的軟體架構。

為了能夠提供物件的各項品質指標，本系統必須解析程式的原始碼，這也是本專題最大的挑戰。另外，各種衡量方式的研究亦即是本專題的學習重點。



第一章 導 論

1.1 動機

軟體的好壞，似乎永遠沒有一定的標準，有人喜好程式精簡易讀，然而也有人喜好以功能強大結構完整為設計重點，從早期的結構化程式設計觀點，到現在的物件導向概念，不管是何種想法，總是找不出一些客觀的評估方式，然而在軟體工程的觀點上，其實對程式架構早已有一些客觀分析的方式，像程式碼的長短、函式的多寡、呼叫的方式…等，其實這都是可明確定義出來的，雖單看一點，也無法說出程式的好壞，然而結合多種訊息，再將之結合，其實要達到不需觀看程式碼，也可比較軟體的效果是可行的。若撰寫的都是同樣程式時，比較起來可以更加客觀明白，這種情況常常在像學校中教授叫學生做程式作業時發生，因而使我們有了做這次專題的想法。

1.2 系統目標

預定分為兩大部分，包含 JSP 的網頁和 Java Application 的記事本程式，此兩者擁有著相同的核心，來分析程式碼，而此部分，是用 JAVACC 工具所編寫的 Java 程式來完成，計算的數學函式則會在相關技術研究中的核心部分會討論到。

網頁部分大致包含資料庫的連結，和 E-mail 傳送、檔案上傳…等功能，在預定功能中會討論到，當然安全上的機制也是考量的重點，使用者的登錄、權限的區分都為相當重要的部分。

記事本程式部分，介面主要為 JAVA SWING 建構而成，由於介面為主要重點，故參考平常所用文書工具來製作，功能上希望能達到簡單的文書編輯、開檔存檔…等動作，詳細功能同樣在後面章節會討論到。

第二章 系統功能

2.1 JSP 網頁部分

由於是網頁部分，簡單的美工當然不可少，為了使畫面美觀，Flash 等使畫面美觀的軟體，都是必要用到的，除此之外，網頁便利與安全為同等重要的事，系統功能描述：

登錄登出系統

教師對學生人員的管理

學生程式文件上傳

Metrics 計算分析

人員的註冊

密碼查詢功能

即時新聞

留言板

2.2 APPLICATION 部分

參考文書軟體，類似記事本，能獨立運作，介面為其特色，提供多功能的整合環境，方便學生評量自己的程式架構，並結合 Metrics 計算分析，更利於程式的撰寫，這部分為功能包含下列幾項：

文書編輯

檔案儲存

Metrics 計算分析

介面的變更

外部軟體呼叫

說明文件

2.3 CK Metrics

Chidamber 與 Kemerer 所提出對物件導向系統的 Metric，以下介紹其中的四個對 class-based design 的 metric，也就是我們在專題中有實作出來的部分，分別是 WMC、DIT、NOC、LCOM。

WMC [Weighted methods per class]：

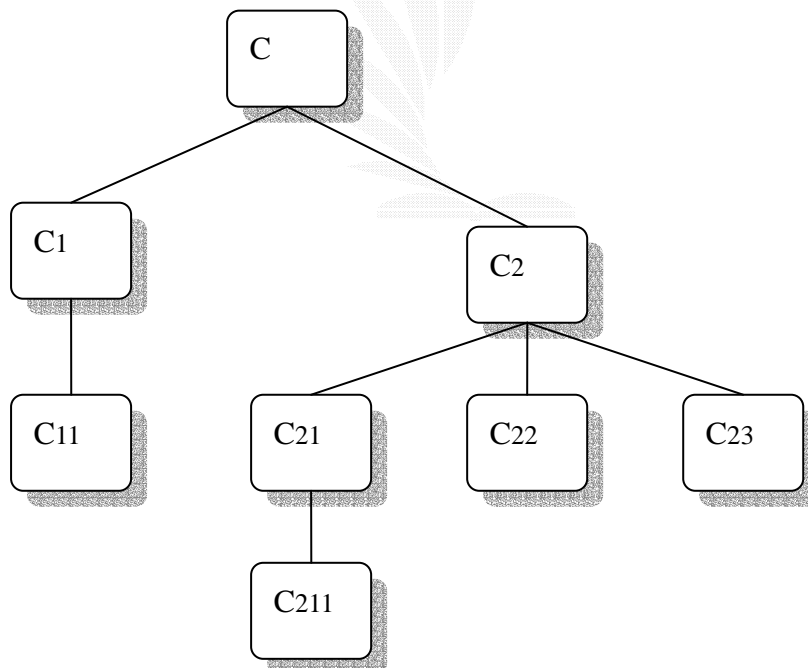
假設一個 class 中定義了 n 個 methods，每個 method 的複雜度分別是 $C_1、C_2、C_3 \dots、C_n$ 。這裡的複雜度可以是某一種特定的 complexity metric，若是將它簡單化，可以將它們以 1.0 來計算。

$$WMC = \sum C_i, i = 1 \text{ to } n$$

當 WMC 越大時，就代表這個 class 有越多的特定應用，可再用性就比較低。所以我們應該使 WMC 的值盡量小，在合理的情況下。

DIT [Depth of the inheritance tree]：

這個 metric 的意思就是「從此節點到 root 節點的最大距離」。如圖， C_{211} 的 DIT 就是 4。



(圖一 A class hierarchy)

當 DIT 越大，就代表下層的 class 會繼承到許多的 method。這也會讓我們很難去預測一個 class 的行為。越深的繼承關係〔DIT 越大〕，也代表這個設計的複雜度越大。但是，以另一個角度來看，大的 DIT 值暗示出有許多 method 可以被再用〔reused〕。

NOC〔Number of children〕：

某個 class 的 subclass 之中，直接隸屬於這個 class 的 subclass 就稱為 child。而 child 的總數就是 NOC。如上圖中的 C₂，它的 subclass 包含 C₂₁、C₂₂、C₂₃、C₂₁₁，但是直接隸屬於 C₂ 的 subclass 只有 C₂₁、C₂₂、C₂₃，這三個就是 C₂ 的 child，所以 C₂ 的 NOC 就是 3。

當 NOC 越大，代表可再用性也越大。不過，所要做的測試工作也會越多。

LCOM〔Lack of cohesion in methods〕：

一個 class 裡的每個 method 中，有可能會有兩個以上的 method 去存取同一個或多個 attribute。LCOM 就是在一個 class 裡，計算共同存取同樣變數的 method 個數。如果沒有 method 存取到同樣的 attribute，則 LCOM = 0。

Definition: Consider a Class C₁ with n methods M₁, M₂, ..., M_n. Let {I_j} = set of instance variables used by method M_i.

There are n such sets {I₁}, ..., {I_n}. Let P = {(I_i, I_j) | I_i ∩ I_j = ∅} and Q = {(I_i, I_j) | I_i ∩ I_j ≠ ∅}. If all n sets {I₁}, ..., {I_n} are ∅ then let P = ∅.

$$\text{LCOM} = |P| - |Q|, \text{ if } |P| > |Q| \\ = 0 \text{ otherwise}$$

舉例，假設一個 class 裡有三個 method，M₁、M₂、M₃。{I₁} = {a, b, c, d, e} and {I₂} = {a, b, c} and {I₃} = {x, y, z}，所以 {I₁} ∩ {I₂} 非空集合，但是 {I₁} ∩ {I₃} 和 {I₂} ∩ {I₃} 是空集合。所以 P = 2，Q = 1，LCOM 就是 1。

如果 LCOM 的值很高，代表 methods 之間藉由同樣存取的 attribute，使他們的關係成對。通常，高的 LCOM 值暗示著應該要把這個 class 分成兩個或更多的 class。所以，若是要保持高的 cohesion，就要讓 LCOM 的值小一點。

第三章 設計與建構

3.1 JSP 網頁部分

3.1.1 需求分析

- 系統說明：提供一個完整的網站，簡化學生繳交程式作業，及教授批改學生程式作業成績的流程
- 系統使用對象：學生及老師
- 功能性需求：

概要	內容
可多人連線	可同時提供同時 200 人(大約 4 個班級的人數)上線服務
提供檔案上傳功能	用來給學生上傳輸面報告，以及程式碼，除此外還必須限制學生無法互相更改別人的檔案
專門的程式評量工具	類似 JavaNCSS 的工具，並結合資料庫(MySQL)，以統計並評量學生作業，以利教授批改及登記作業的流程，並可 parser Java Code，輸出 parsing tree
學生資料管理功能	主要是給教授所用的功能，方便教授新增或刪除學生資料
佈告欄	提供留言發問的場所
個人化設定	可依個人喜好來配置功能按鈕的擺放位置
Client 不需另外安裝軟體	擁有瀏覽器及可以連上 internet 的電腦即可使用本系統
將結果 e-mail 給使用者	將評量完的結果寄回給學生
提供會員制度，帳號管理功能	區分學生、教授及一般觀看者

(表一 JSP 網頁功能性需求)

● 非功能性需求：

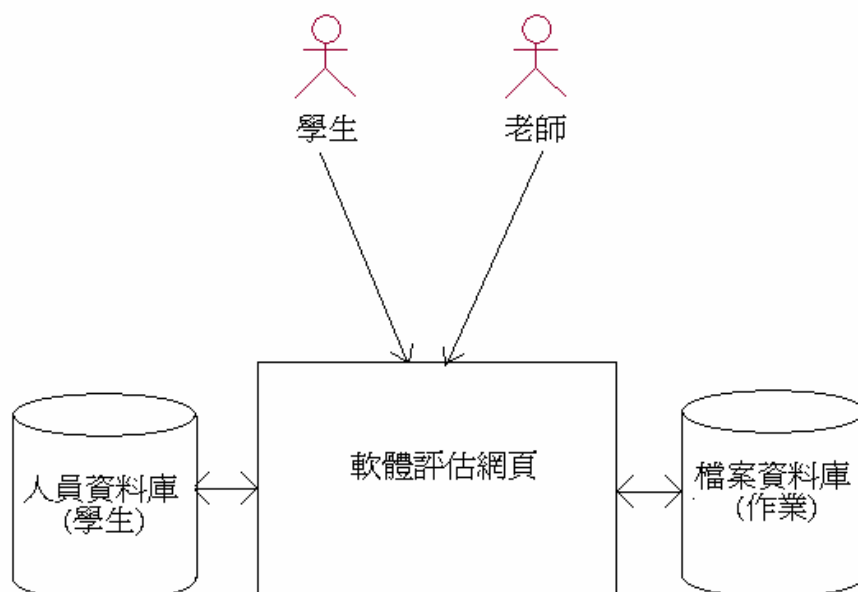
概要	內容
容易操作	提供圖形化使用者整合介面 (GUI)
反應時間快	每個網頁最好不要超過 5 秒的處理時間，計算 metrice 不超過 30 秒
提供說明	提供線上文件說明使用方法，並在選項後面加以註解
易維修管理	最好提供與平台無關的介面，方便資料的搬移和處理

(表二 JSP 網頁非功能性需求)



3.1.2 網頁系統示意圖

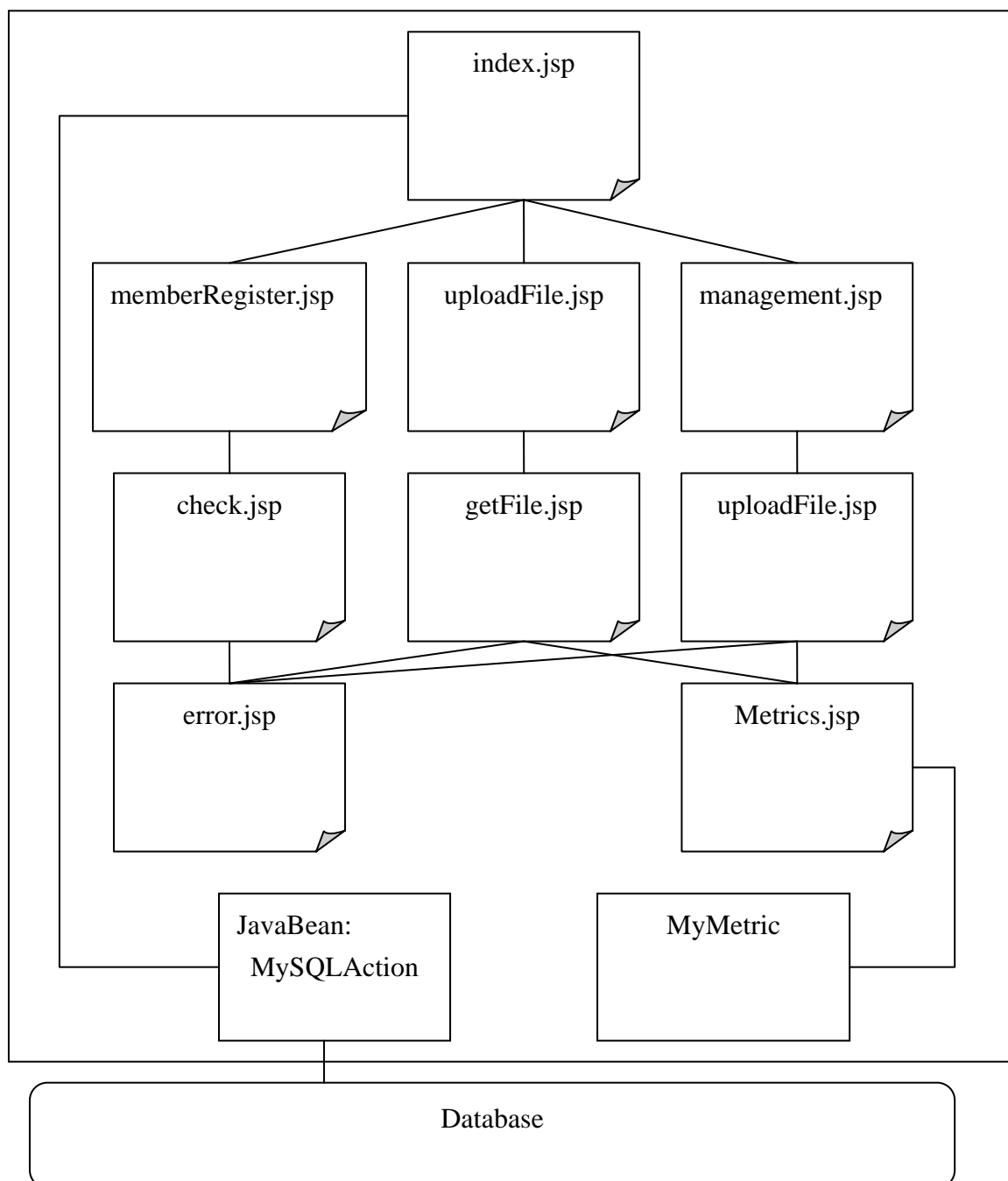
下圖為實體架構圖，說明了整個系統所用到的軟體環境，主要為三大部分，各為人員資料庫，檔案資料庫，以及軟體評估系統，軟體評估系統為主要我們所必須建構的，而人員資料庫和檔案資料庫，都有現成的軟體可完成，只需加入欄位即可，將人員及檔案歸為資料庫管理，主因為安全性及效能上的考量，資料庫軟體本身對於資料已有較好的保護，故可使的我們減輕不少安全上的程式負擔，而運作的方式，人員與上傳檔案都為資料庫控管，軟體評估系統，只有單純和網路使用者的溝通，及最主要的軟體分析，和回傳結果。



(圖二 網頁系統示意圖)

3.3.3 系統架構

此部分為編寫 JSP 網頁的主要分配，去除掉旁支的功能，所畫出的圖，較為特別的是在 JavaBean 的部分，由於要能多人連線，故我們將與資料庫連線的部分，改以用 JavaBean 來完成，提高執行的效能，也易於將系統未來可加入 Connection Pool 的連線方式，強化系統執行效率。



(圖三 系統架構圖)

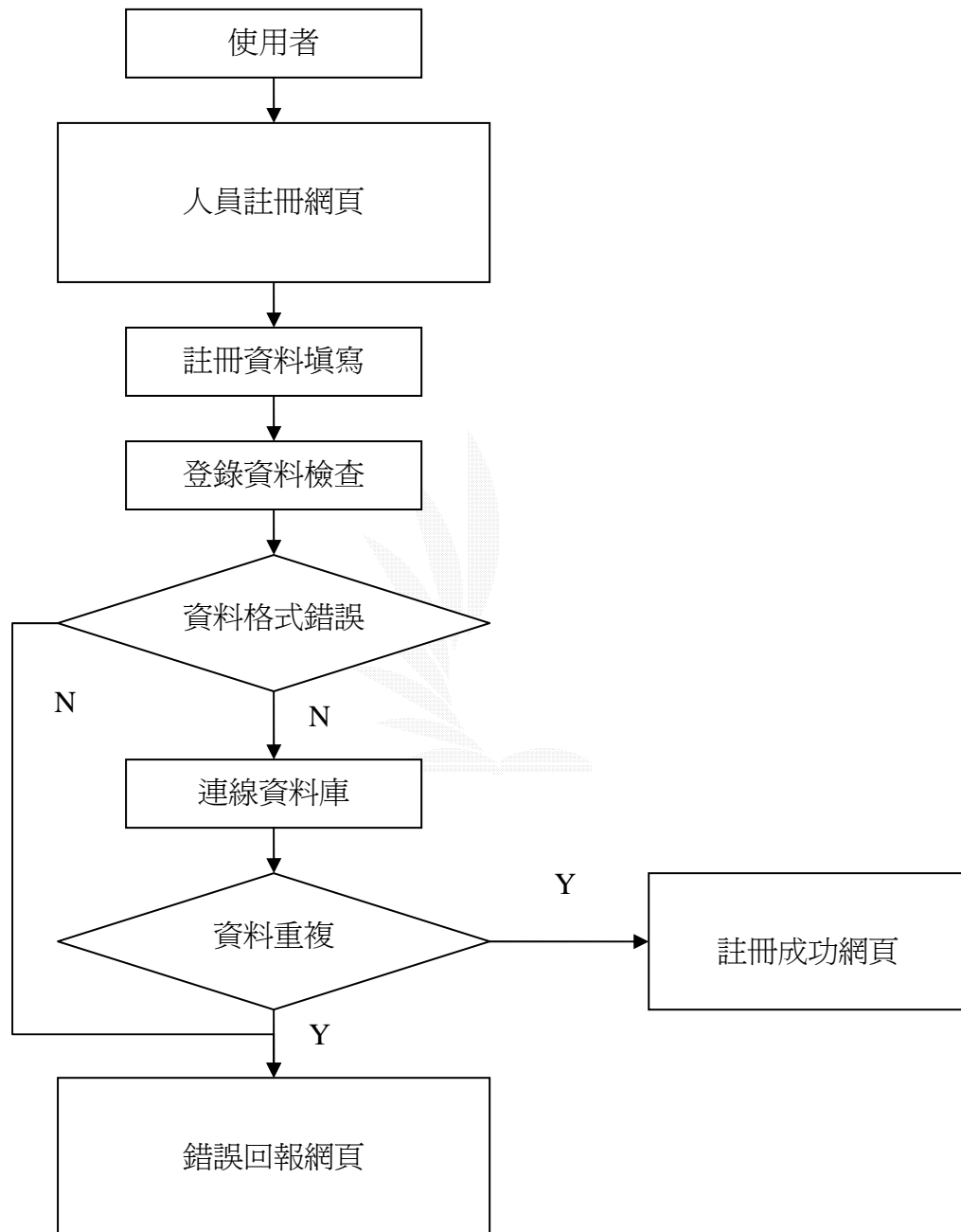
3.1.4 所用工具介紹

架設本系統，需要多項工具，詳細安裝內容請見”[附錄 B 安裝步驟](#)”，以下為安裝軟體的粗略介紹。

- Apache
此為 http 伺服器的著名軟體，功能強大，且較為安全，有多種平台的版本，包含 UNIX、LINUX…，其他類似的軟體有 IIS、PWS。
- Tomcat
同 Apache 為伺服器，但只專門處理 JSP 文件，對 JSP 有較好的執行效率，但對於靜態網頁，如 html，還是需 Apache 來處理，才能換得較好效能，故必須結合 Apache 來使用。
- MySQL
免費的關聯式資料庫軟體，相較其他資料庫，其功能並不強大，但 MySQL 除不用收費外，其效率可說是數一數二，故選擇以此資料庫系統，作為主要處理資料之用。
- phpAdmin
由於 MySQL 並無像 ACCESS 如此強大的圖形介面，管理起來可能較為複雜，phpAdmin 即可補強這點，提供完整的使用者圖形介面來管理資料庫。
- PHP
是一種類似 JSP 同為伺服器端的語言。由於 phpAdmin 為 php 所寫成的軟體，故還必須架設能執行 PHP 的環境，才能使用，所以要將 PHP 執行套件和伺服器做連結。
- NO-IP
動態轉址服務，能將不定的 IP，轉換成同一 Domain Name。

3.1.5 學生註冊流程

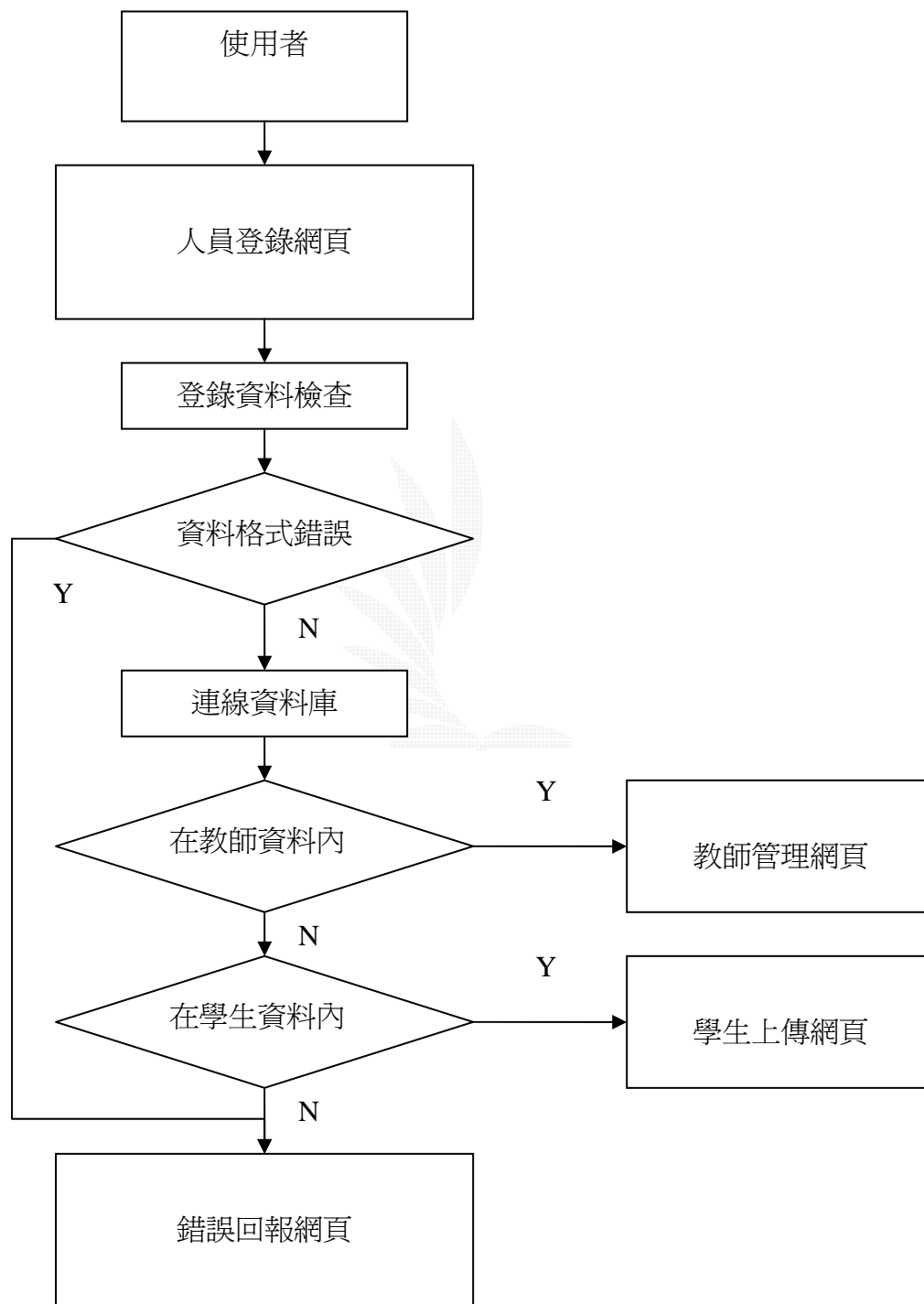
此部分為學生註冊流程，主要動作是將學生所填寫好的個人資料，建立到資料庫系統內，為防止學生隨意輸入，還必須做資料的檢測，完成時還要回報完成結果。



(圖四 學生註冊流程圖)

3.1.6 使用者登錄流程

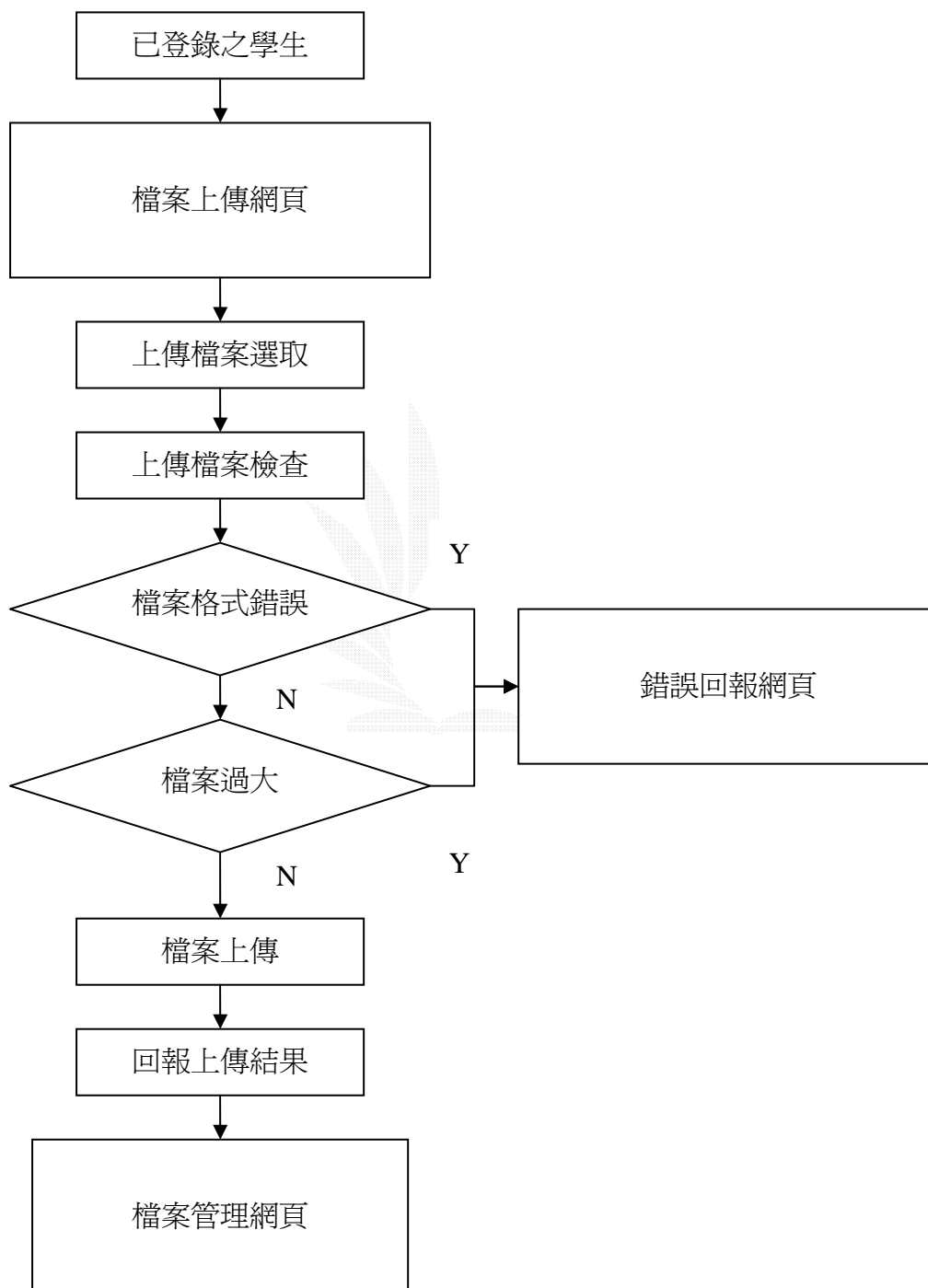
下圖為登錄流程，這部分是要將連線系統的使用者辨識出來，是學生、教授還是一般觀看者，並會各自轉往其特製網頁，除此之外也會將使用者資訊暫存到伺服器的 Session 中，方便其他網頁過濾無權限的使用者，防止不正確的存取網頁。



(圖五 使用者登錄流程圖)

3.1.7 檔案上傳流程

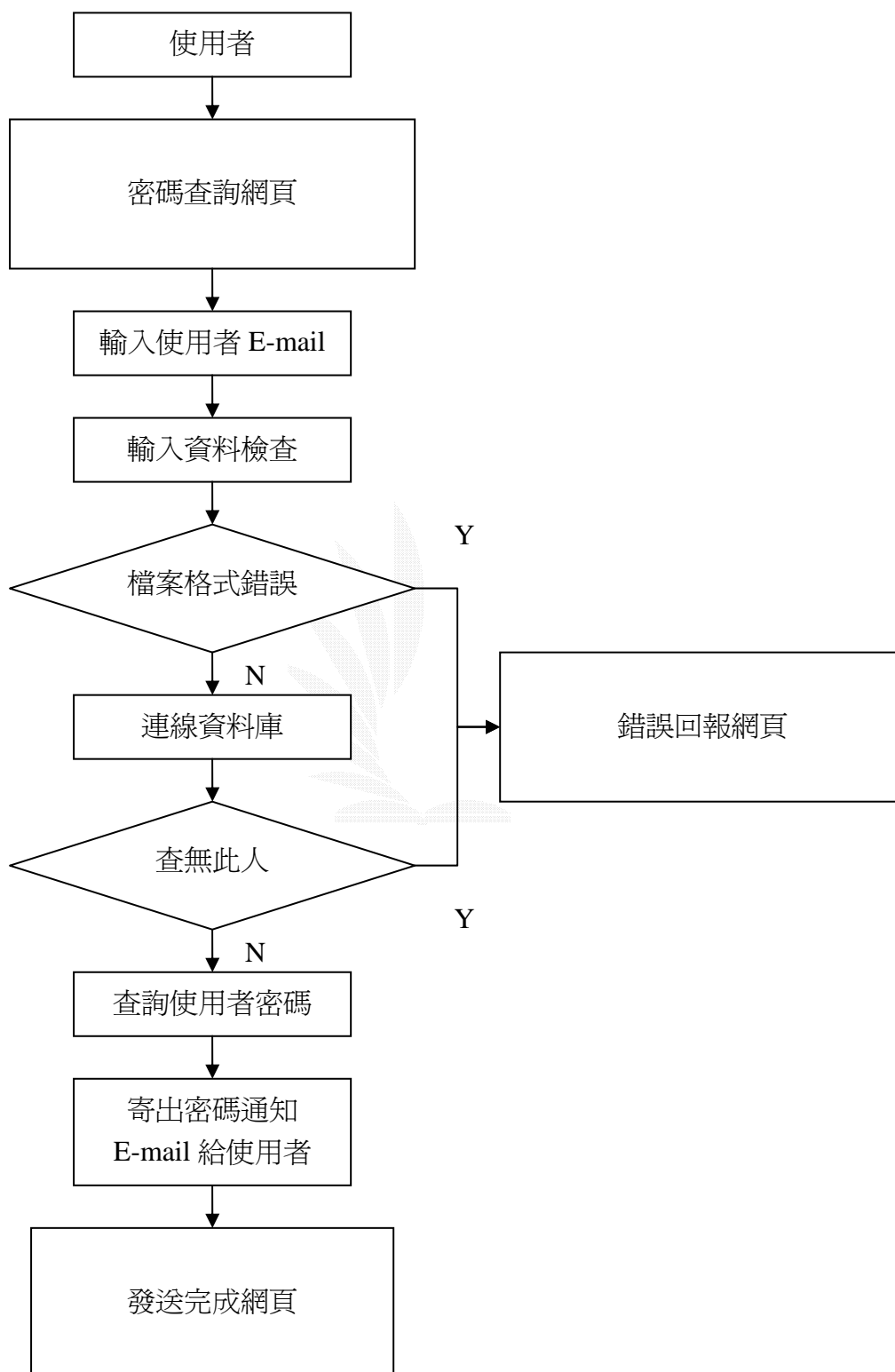
檔案上傳的檢查過程如下，上傳檔案的使用對象，是已登錄的學生，這部分的辨別，在使用者登錄程流程已說明過，而檔案的格式及大小，還必須加以限制，以防過大的檔案癱瘓系統，最後同樣的會回報訊息給學生，以了解上傳情況。



(圖六 檔案上傳流程圖)

3.1.8 密碼查詢

這部分為防止使用者忘記密碼，而設的機制，只要輸入使用者 E-mail，系統就會自動回覆密碼文件到使用者信箱。



(圖七 密碼查詢流程圖)

3.2 APPLICATION 部分

3.2.1 需求分析

- 系統說明：提供一個類似記事本的程式，能做簡單的文書編輯，並儲存程式。
- 系統使用對象：學生
- 功能性需求：

概要	內容
專門的程式評量工具	類似 JavaNCSS 的工具，並結合資料庫(MySQL)，以統計並評量學生作業，以利教授批改及登記作業的流程，並可 parser Java Code，輸出 parsing tree
檔案處理	能開啟 Java 檔原始檔，並能儲存修改結果
個人化設定	可依個人喜好來配置背景顏色，內框形式…等
可呼叫外部工具	方便使用者執行外部的軟體，如 Java 的編譯器、其他編輯程式…等

(表三 Application 功能性需求)

- 非功能性需求：

概要	內容
容易操作	提供圖形化使用者整合介面(GUI)
提供說明	提供給使用者使用上的說明
易維修管理	將各介面和核心部分做切割，使其各自獨立，以易於改寫

(表四 Application 非功能性需求)

3.2.2 設計架構

我們將系統拆解為以下幾部分：

圖形類別：

圖形介面包含 frame、internalframe、toolbar、popupmenu、menubar，為建構使用者介面的重點。設計將各元件都獨立撰寫，以方便未來擴充時，只需繼承原來類別即可，不需額外改寫原來程式。

事件處理類別：

用來處理所有的事件，也就是由圖形類別所產生的各項結果，將圖形介面所產生的動作，做出回應

parser 類別：

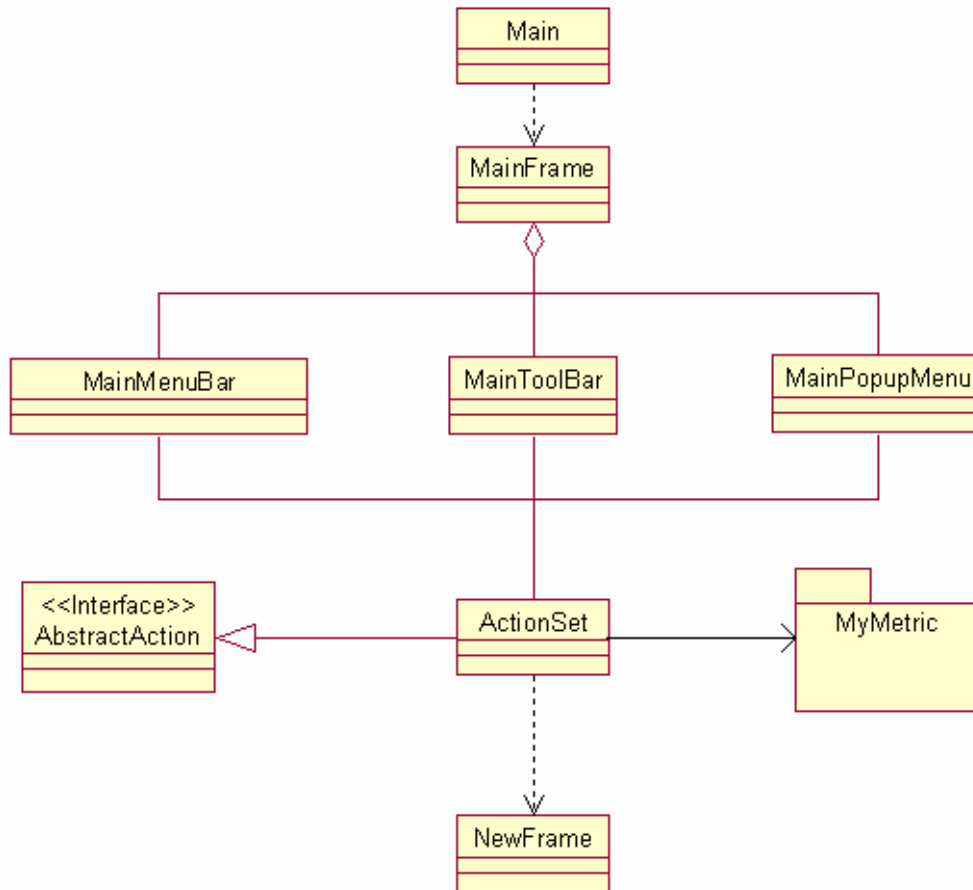
自製的 MyParser package，分析程式碼，其執行的時期，由事件類別來控制。

檔案類別：

過濾非 JAVA 的檔案，以防止程式讀取到不明檔案，以致於程式出錯。

3.3.3 類別關係

此圖並不是正規的表示，但簡單的說明各類別呼叫的關係，箭頭的方向，指的是呼叫、啟動(程式中 NEW 出來)或間接觸發(產生事件)的動作。



(圖八 類別關係圖)

3.3.4 類別規格

class Main

屬性	回傳型態	函式名	參數	內容	進度
public static	void	main	(String[] args)	程式啟動點	1

(表五 Main 類別規格)

class MainFrame extends JFrame implements FrameCtrlSet

屬性	回傳型態	函式名	參數	內容	進度
protected	void	buildContent	()	建構視窗及 狀態內容	1
protected	void	buildMenuBar	()	MenuBar	1
protected	void	buildToolBar	()	ToolBar	2
protected	void	buildPopupMenu	()	PopupMenu	2

(表六 MainFrame 類別規格)

class MainMenuBar extends JMenuBar

屬性	回傳型態	函式名	參數	內容	進度
protected	void	buildMenu	(String[] functions)	建立選單項 目	1
private	JMenu	addItem	(JMenu menu, String functions, char mnemonic)	增加項目內 容	1

(表七 MainMenuBar 類別規格)

class MainToolBar extends JToolBar

屬性	回傳型態	函式名	參數	內容	進
----	------	-----	----	----	---

					度
protected	void	buildMenu	(String[] functions)	建立選單項目	1
private	JMenu	addItem	(JMenu menu, String functions, char mnemonic)	增加項目內容	1

(表八 MainToolBar 類別規格)

class NewFrame extends JInternalFrame

屬性	回傳型態	函式名	參數	內容	進度
public	JTextArea	getTextArea	()	取得所在視窗的內容	1
public	void	changeTitle	(String addItem)	改變視窗名稱	2

(表九 NewFrame 類別規格)

選單功能

File	Edit	View	Tool	Window	Help
Open	Undo	Change UI	Console		Help
Save	Redo				About Us
Save as	Cut				
Exit	Copy				
	Paste				

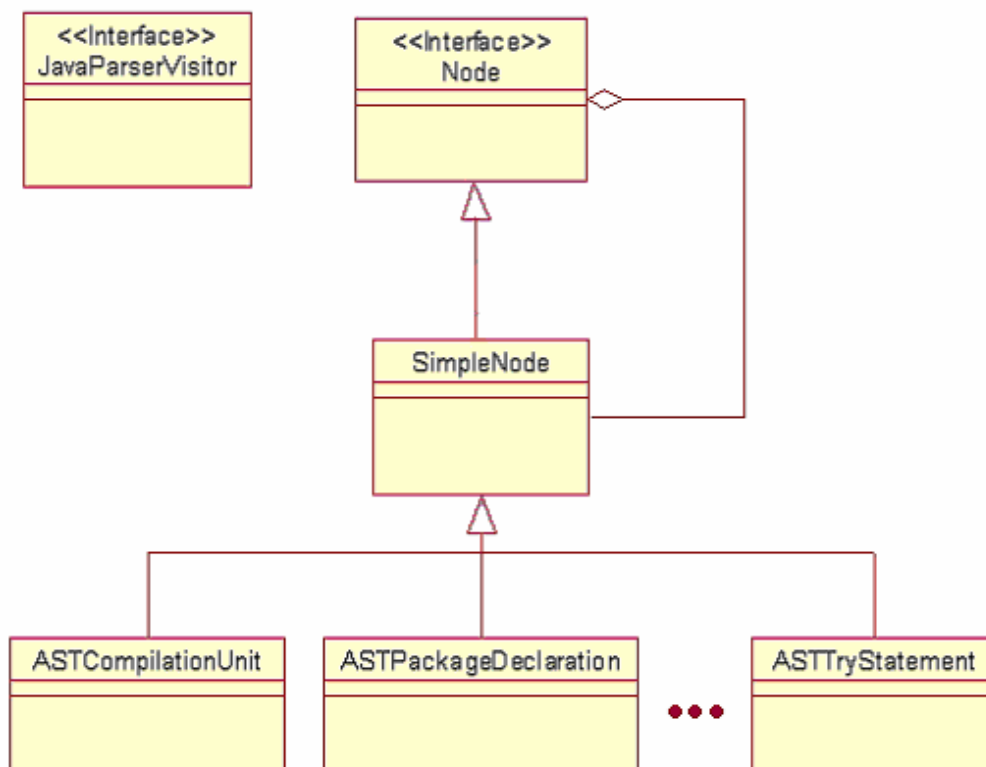
(表十 選單功能規格)

3.3 Metrics 計算部分

這一部分的程式是利用 JavaCC 完成的，我們所要 parse 的目標是 Java 語言，所以必須提供給 JavaCC 完整的語言敘述，lexical 的 regular expression 以及 parser 的 production rule。這些其實在 JavaCC 內附的範例就有提供，不過它只支援到 Java1.1。若是解析到新版 Java 才有的語法會發生錯誤。網路上也有修改好的版本，可支援到 Java1.4。一開始我們是以 Java1.1 的版本來開發，之後才改成用 1.4 版的。支援 Java1.4 的 .jj 檔在以下網址可以取得 <http://www.cobase.cs.ucla.edu/pub/javacc/>，其內還包含許多其他的語言的 .jj 檔，如 C/C++、VB、Ada、HTML、XML、JavaScript... 等等。若是有需要解析這些語言，只要將原來的 .jj 檔作修改，在 regular expression 及 production rule 之後加上要做的事情。regular expression 和 production rule 就不用自己寫了。

我們還利用了 JJTree 程式來幫助我們建立 syntax tree，我們將原來的 .jj 檔修改成 JJTree 可接受的格式，且將副檔名改成 jjt。將此 .jjt 檔交由 JJTree 來處理，JJTree 會為我們建立加強後的 .jj 檔，還有 Node.java、SimpleNode.java、JJTJavaParserState.java、JavaParerVisitor.java、JavaParserTreeConstants.java 以及他為每個 production rule 都建立了一個 .java。以下是他們的類別關係圖 [部分]。

JJTree所產生的類別

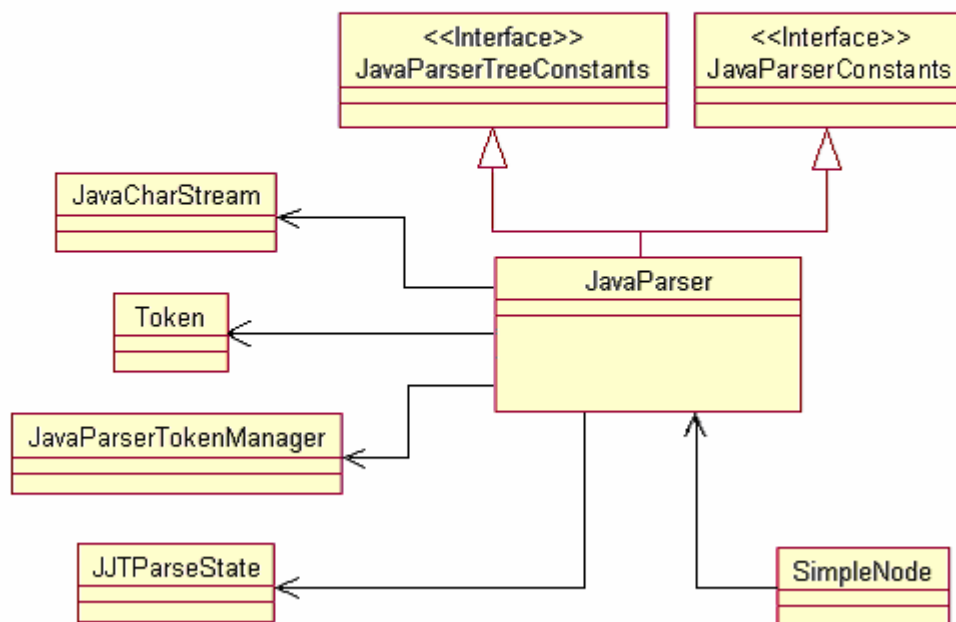


(圖九 JJTree 所產生的類別圖)

Node 是一個 interface，SimpleNode 繼承它，它提供了增加子節點、修改子節點…等操作以及 accept 方法〔Visitor Pattern〕。每個 production rule 所建立的類別都繼承 SimpleNode。

接著將 JJTree 所產生的 .jj 檔，交由 JavaCC 來產生 parser 的 Java Code，JavaCC 會產生 "TokenMgrError.java"、"ParseException.java"、"Token.java"、"JavaCharStream.java"、"JavaParserTokenManager.java" 還有最重要的 "JavaParser.java"，以下是類別關係圖：

JavaCC產生的類別圖



(圖十 JavaCC 所產生的類別圖)

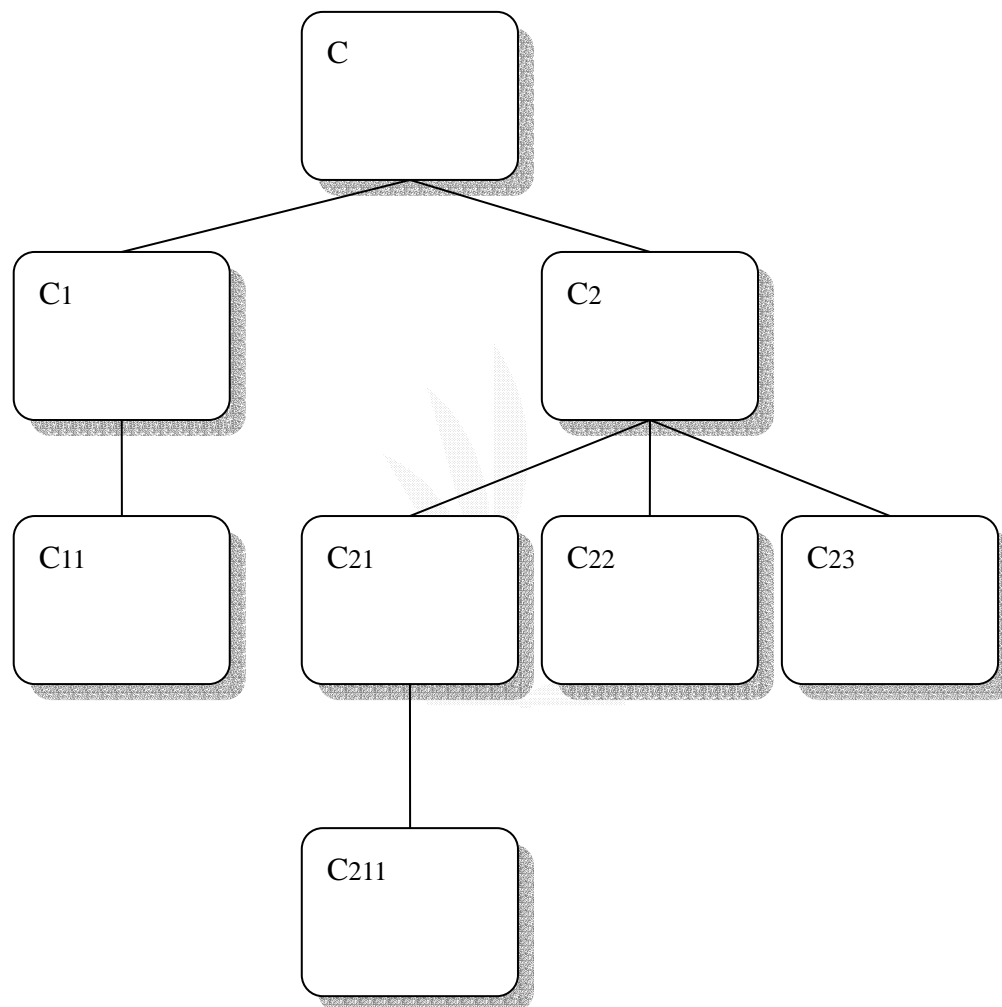
這個圖也只是略圖而已，並沒有畫出所有的細節。JavaParser 實作 JavaParserTreeConstants 和 JavaParserConstants 兩個 interface。JavaParser 聚合 JavaCharStream、Token、JavaParserToken、JavaParserTokenManager、JJTParseState。而 SimpleNode 聚合 JavaParser。

MyJavaParserVisitor 和 MakeTreeVisitor 都是繼承 JavaParserVisitor interface [JJTree 幫我們產生的]，必須實作每個 visit() 方法。StructOfClass 是存放 Class 資訊的資料結構，如每個 MethodDeclator 的 Node，以及他繼承自誰，有哪些已知的子類別等等...

StructOfDirectory 存放整個資料夾的每個 class 的 StructOfClass，所以 StructOfDirectory 聚合 StructOfClass。我們

採用 2-pass 的模式。

- Pass-1：對每一個 class 個別處理，計算可獨立運算的 Metric，如 WMC 及 LCOM。以及為其他 Metric 作準備，例如紀錄下此 class 繼承的是哪個 class，以利 pass-2 來計算。
- Pass-2：因為 DIT 以及 NOC 無法在單個 class 中看出來。有了 pass-1 所提供的資訊，就能找到 DIT 及 NOC。如圖，



(圖十一 類別的繼承關係圖)

以 C₂ 為例，在 pass-1 可以知道 C₂ 繼承 C，C₂₁、C₂₂、C₂₃ 都繼承 C₂。所以在計算 NOC 時就可把 C₂ 計入 C 的子類別，C₂₁、C₂₂、C₂₃ 計入 C₂ 的子類別。計算 DIT 時，C₂ 就能找到其父類別 C，因為 C 沒有父類別了，所以不需要再找下去，C₂ 的 DIT 就是 1。以 C₂₂ 為例，往上找到 C₂ 是它的父類別，C₂ 又可以找到它的父類別 C。所以 C₂₂ 的 DIT 就是 2。

第四章 相關技術研究

JSP 網頁部分

4.1.1 JavaBean 介紹

"A Java Bean is a reusable software component that **can be manipulated visually in a builder tool.**" 上面的這句話是 JavaBeans Specification 對 Java Bean 的定義。舉例來說，在 Java 1.1 裡一個單獨的 AWT Component 都是一個 Bean。我們可以透過一些 tools，如 IBM 的 Visual Age J++，Borland 的 JBuilder，或是 Sun 的 JDK 等等的 editors，來幫我們使用這些 components。

- 目的
 - 軟體程式發展的目的之一就是使程式碼可以在其他的程式中**重覆使用**。由於像 JAVA 這種程式語言的出現，使得我們所寫的程式不用作任何的改變就可以使其在許多不同的平台上執行。
 - 然而，JAVA 並不會自動地讓軟體程式變的可以重複使用，因為所謂可以重複使用的軟體程式其目標是要讓程式發展者可以**不需要再次編譯該程式碼**就可以使用該程式碼。其中暗示著程式發展者不用再次地編譯其程式碼，就可以整合該程式碼片段至他們的程式中。這些便是 JAVA Beans API 的目標。
- 所需具備的知識(除了一般的 Java program 能力外)
 1. 如果是要發展一些 GUI editors，application builders 等等，或是要做一個實際可用的 Java Bean，那就必須要對 JavaBean API 很熟。
 2. 如果是利用一些如 JBuilder 的 editor 來使用 Java Bean，或是來發展 applications，那可以不必對 JavaBean API 做太深入的研究(也就是說不用去理它)，只要對所使用的 components(Beans)下功夫即可，熟悉所使用的 Beans 的功能。
 3. 不過或多或少都必須知道 Java Bean 的基本構造。Bean 的結構。
- 所需要的工具(建議)

- 一個支援 Java Bean 的 GUI visual editor，因為 Java Bean 或多或少都會有 GUI 介面，所以有一個這樣的工具，不但可以測試你所寫的 Java Bean，還可以節省你開發的時間。
- 如何寫一個 Java Bean
 0. Java Bean 並不像 Applet，必須繼承一個 Applet 的 class，雖然 Java Bean 也有一個叫 Beans 的 class 可供繼承，但是並不強迫一定要繼承。另一方面雖然很多的 Beans 是 AWT 的 Components，但是你也可以寫一個不會 show 任何東西在螢幕上的 Bean（但是可以被你的 edit tool 看見，例如一個 sound player）。
 1. A Bean exports *properties*, *events*, and *methods*.



- Bean 的結構

(圖十二 Bean 結構圖)

- Properties：

一個 Bean 的 properties 描述該 Bean 的屬性，如顏色，大小，或是 label 等等。Properties 通常透過一對 methods 如 set<property> / get<property> 來改變 / 取得 此 property。

- Methods：

和一般的 Java methods 一樣，不過在 Java Bean 中，只有 public 的 methods 才會被 edit tool 看見。

- Event :

Java Bean 使用和 Java AWT 1.1 一樣的 events model。除此之外它還定義了一種 event，用來增加和移除 event listener objects。

- Bean 的一些 feature :

- Introspection :

利用一些 rule 來讓你的 Java Bean 提供的 methods 可以被 builder tool 來看見。它是依照下面幾個 rules 來搜尋。

- Bean 的 Methods :

它會把 public getXXXXX/setXXXXX 以及 boolean isXXXXX 的 methods 找出來。接著把你宣告成 public 的 methods 也挖出來。然後它會尋找有加 BeanInfo 的 class 名稱(Ex: 如果你的 Java Bean 叫 **myBean**，那 **myBeanBeanInfo** 就是它搜尋的目標)。

- Persistence :

利用 Object Serialization 的機制來 Save/Load Java Bean 的狀態改變。

- Customization :

當我們製作一個大型的 Java Bean(或是複雜一點的)，光是用 set/get 的方法來改變 properties 是不夠的，因此 Java Bean 提供一個 customizer 的 class 來自己定義一些改變 properties 的方法。

4.2 APPLICATION 部分

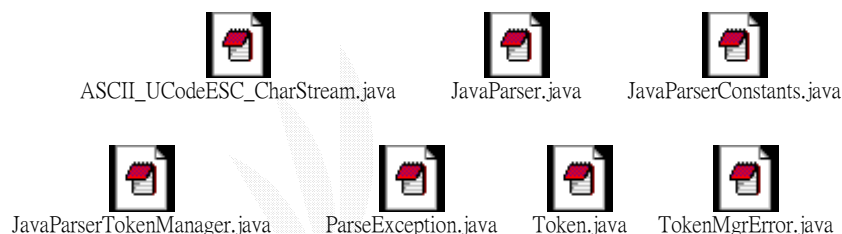
4.2.1 JavaNCSS 介紹

- 說明：JAVANCSS 為一個免費自由軟體，其網址為 <http://www.kclee.com/clemens/java/javancss/>，此軟體可分析程式架構，算出各類 METRIC，並可將所算出的結果存成 XML 格式
- JavaNCSS 架構
- JavaNCSS 依檔案分為兩部分：

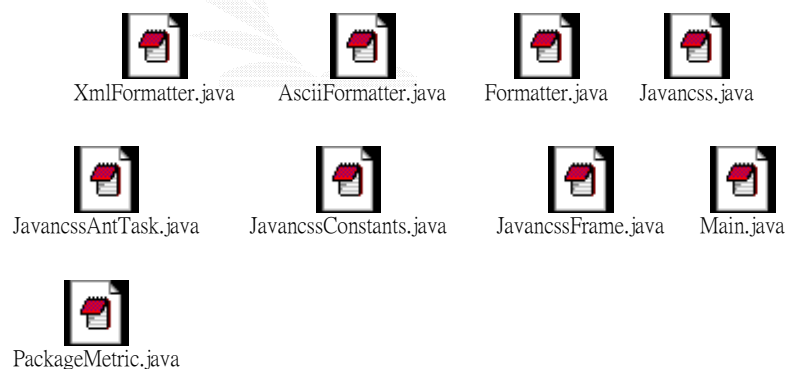


- parser 部分：由 JavaCC 所產生的程式碼，而 `Javac1.1.jj` 就是

指主文件



- 主程式部分：



依照內容再細分

- 執行類別：包含介面的設定和整體呼叫
- 格式轉換類別：Xml、Ascii 的格式轉換
- 計算測量類別：量測計算結果並傳回結果
- parser 類別：找出 token 並檢查語法

- JavaNCSS 分析結果

JavaNCSS 雖已有我們專題所需做到的結果，但其程式碼改寫不易，由於之前改寫的為一個的德國人，其註解不是看不懂，就是德文字碼無法顯示，使得可讀性很低，加之其很多部分為用到 JavaCC 的文件，使的我們讀起來格外吃力，故我們讀完後放棄以改寫此軟體的做法，只採用其系統的粗略架構，如 parser 的做法、類別的切割方式…等，並重新規劃構想，簡化設計，將系統設計成只要做到查出 class、function…等 METRIC，所以由 Javal.1.jj 產生的 parser 類別可以簡略，改以自己編寫，而輸出輸入格式，也都定為 java 原始檔，故也可將之簡化，但界面部分，由於要求以圖形介面，再加以強化擴充。



4.3 Metrics 計算部分

因為我們計算 Metrics 的部分是利用 JavaCC 這個工具，而且該工具還利用到了一些 Design Pattern，如存放 syntax tree 時用到 Composite Pattern，要對 tree 做處理時要利用 Visitor Pattern，有關這部分的觀念技術，簡述如後。

4.3.1 JavaCC 介紹

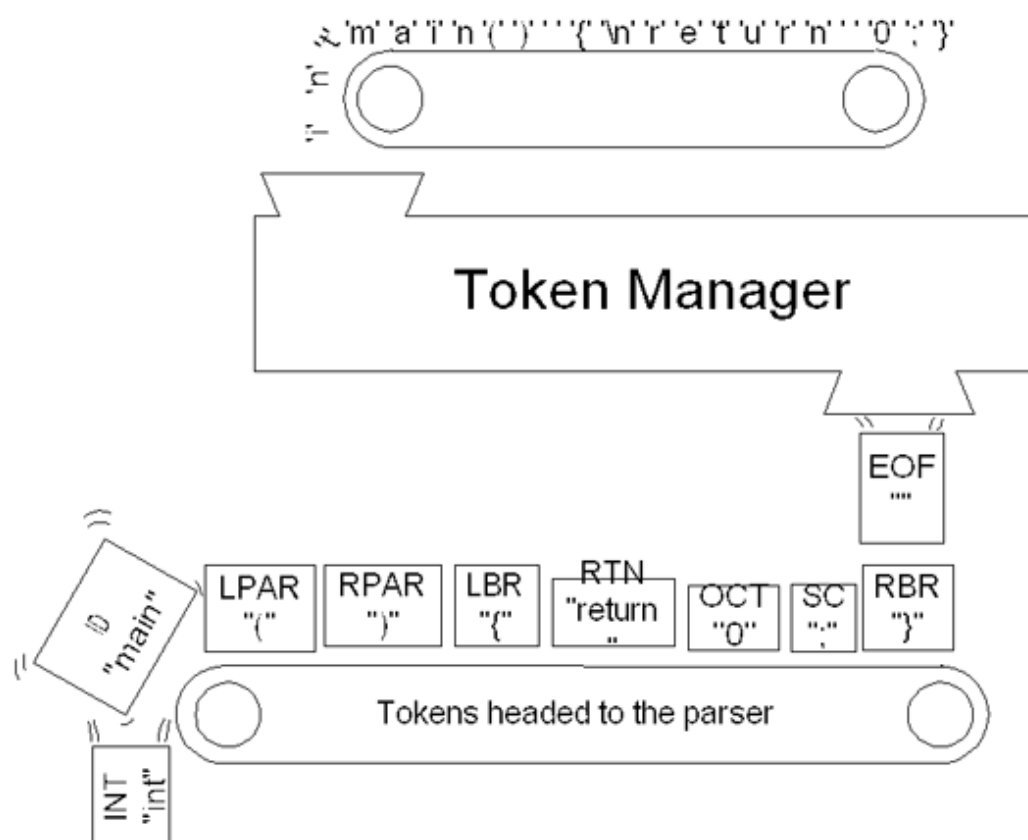
JavaCC 是 Java compiler-compiler 的縮寫，這是一個 Parser 和 lexical analyzer 產生器。JavaCC 會讀進一個語言敘述，產生 java 程式碼。產生出來的程式碼經過編譯後就可讀進該語言且進行分析。

若是沒有這類 parser generator 的工具，要做一個複雜的 parser 將會非常困難，且修改也很不容易。” compiler-compiler” 意即 compiler 某語言的規則，產生這個語言的 compiler，像是 UNIX 裡的 yacc。不過 JavaCC 還有內含像是 lex 的 lexical analyzer 產生器的功能。所以 JavaCC 可以看作是 Lex 加上 Yacc 的 Java 版。不過還是有一些不同，Lex 和 yacc 是產生 C 的程式碼，而 JavaCC 產生的是 Java 的程式碼。Parser 的部分，Yacc 是 bottom-up，JavaCC 是 top-down。雖然 JavaCC 是用 top-down 的方式，但是它的 Lookahead 能力可以解決大部分 top-down parsing 的問題。

JavaCC 是 Sreeni Viswanadha 和 Sriram Sankar 在 Sun 工作時所寫的。現在 JavaCC 是 open source，由原作者繼續維護。且他們也不斷地在加強它。我們可以從 <https://javacc.dev.java.net/> 下載到最新版的 JavaCC，含有原始碼、文件及一些範例。

4.3.2 Lexical analyzer

在 JavaCC 裡又叫做 token manager，token manager 讀進一串連續的字元。依規則產生一個一個 token 出來。這裡的規則就是要解析的語言中不可分離的最小單位。例如 C 語言裡的 “int” “main” “(“ “void” “}” “;” 等等，就是 C 語言的 token。當然 JavaCC 並不只限於解析 C 語言，只要我們能提供一組規則給 JavaCC。這一組規則要以 regular expression 來表示。

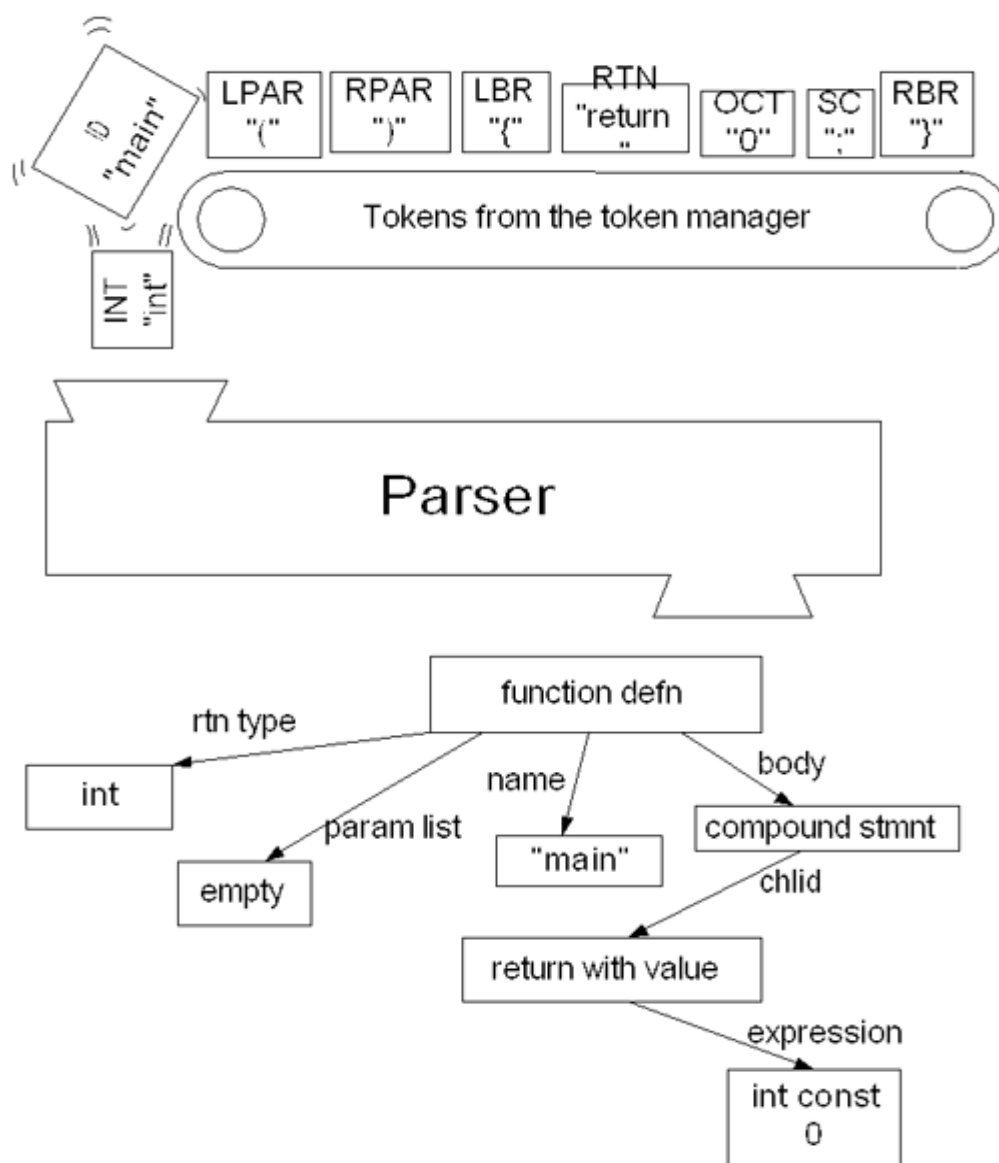


(圖十三 Token Manager)

4.3.3 Parser

Parser 會讀進一連串的 token，分析她們的結構，作相對應的處理，最後要產生什麼，可依程式設計者來決定。若只是要產生 syntax tree，可利用 JJTree 及 JTB 來完成，在以下我會做介紹。

程式設計者必須提供一組” Extended BNF production rules”來描述文法規則，在每一個 production rule 之後，可寫入一段 Java 程式碼，來告訴 JavaCC 在符合此 production rule 時，要做什麼處理動作，該產生什麼。JavaCC 最後會產生出一個負責的 parser 的 Java 類別，程式設計者就可以利用此類別來完成 parsing 的工作。



(圖十四 Parser)

4.3.4 JJTree 和 JTB

JJTree 是和 JavaCC 附在一起的一個程式，它是 JavaCC 的前置處理器，可以幫助程式設計者產生 syntax tree 的結構，它會產生出可以被 JavaCC 接受的格式，在每個 EBNF 之後加入新增 Node 的指令，將每一個 production rule 都當作 Tree 裡的一個 Node，不需程式設計者自己寫。只要將 JJTree 所產生的 .jj 檔案在經過 JavaCC 產生 Java code，這些 Java code 再編譯後，就能將欲解析的語言 parse 結果輸出成 Tree 結構。JJTree 還會產生一個 Visitor 的 interface 以及每個 Node 的類別，程式設計者可以利用基本的 Visitor Design Pattern 進行 Tree 結構的 Tracing。

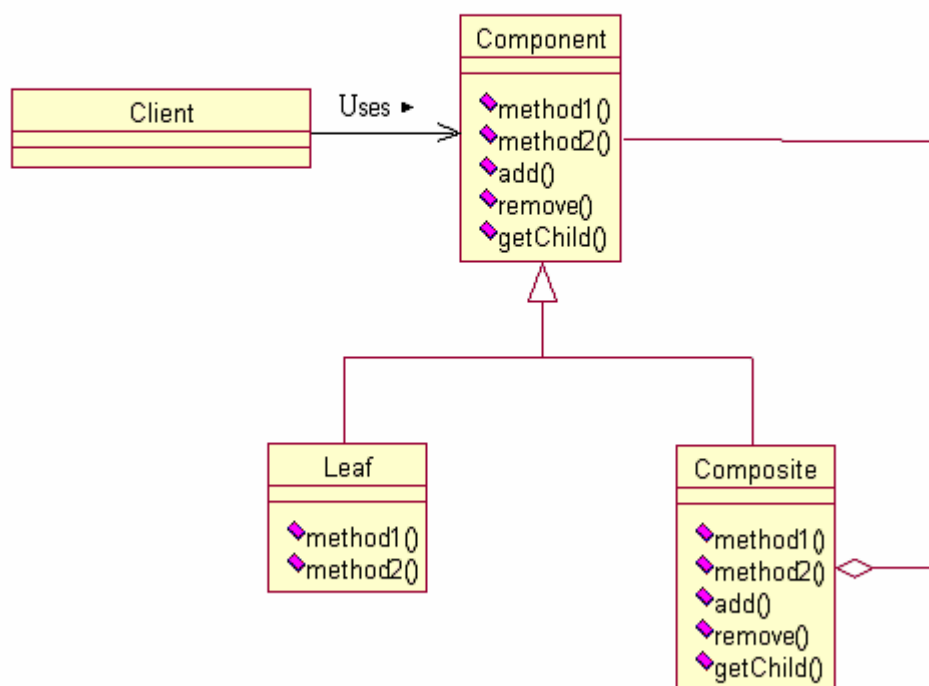
JTB(Java Tree Builder)可在 <http://www.cs.purdue.edu/jtb/index.html> 取得，做的事和 JJTree 差不多，不過它比 JJTree 還要多功能。



4.3.5 Composite Design Pattern

節錄自『Design Patterns 於 Java 語言上的實習應用 ----- 結城浩 著』一書。電腦的檔案系統裡，「檔案」要放在「目錄」〔或叫做「資料夾」〕裡，而「目錄」也可以放在某個「目錄」裡。雖然檔案和目錄是兩種不一樣的東西，但是兩者都是「可以放在目錄裡」的東西。所以我們可以把它們視為同一種類。像這種「畫中有畫」的遞迴結構，我們可以利用 Composite Pattern，對容器〔目錄〕和內容〔檔案〕一視同仁，建立遞迴結構的 Design Pattern。以下是 Composite Pattern 的類別圖：

Composite Pattern的類別圖



(圖十五 Composite Pattern 的類別圖)

以下說明出現的所有參與者：

- Leaf〔葉子〕參與者
表示「內容」的參與者，不能放入其他東西，如檔案結構的「檔案」。

- Composite [複合體] 參與者
表示「容器」的參與者，可放入 Leaf 參與者或 Composite 參與者，
如檔案結構的「目錄 [資料夾]」。
- Component 參與者
對 Leaf 參與者和 Composite 參與者一視同仁的參與者。Component
參與者是 Leaf 參與者和 Composite 參與者的共同父類別。
- Client [委託人] 參與者
利用 Composite Pattern 的人。

Composite Pattern 範例

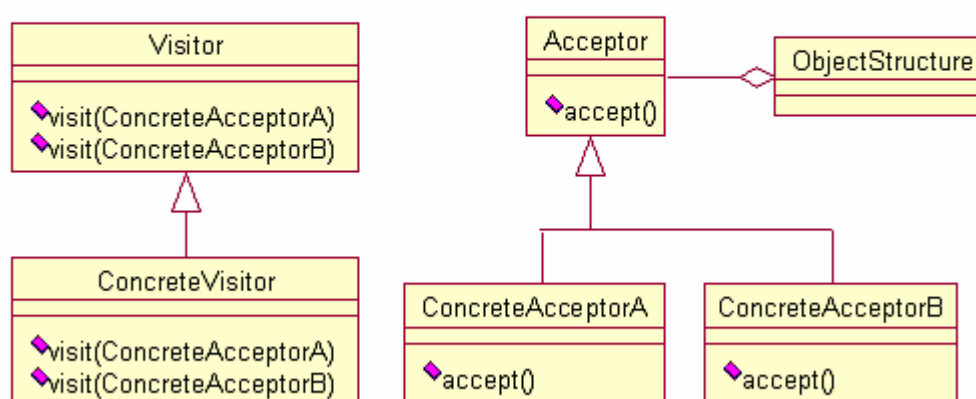
[放在和 Visitor Pattern 一起講解]



4.3.6 Visitor Design Pattern

若我們現在想對一個資料結構裡的所有元素做某項「處理」，這個「處理」動作的程式碼應該放在哪裡。最直覺的相法是寫在表示資料結構裡的類別裡，但是當我們的「處理」動作不只一個時，就需要在類別裡寫很多的處理動作。而且若是我們想要新增某個處理時，不得不修改原始的資料結構類別。Visitor Pattern可以把資料結構和處理兩者分開，另外寫一個表示在資料結構內穿梭來去的「訪客」類別。當我們想增加新的處理動作時，只要再寫一個「訪客」類別就好了。而資料結構這邊，要建立可以接受「訪客」呼叫的機制即可。以下是 Visitor Pattern 的類別圖：

Visitor Pattern 的類別圖



(圖十六 Visitor Pattern 的類別圖)

以下說明出現的所有參與者：

- Visitor [訪客] 參與者

Visitor 參與者是對每個資料結構中的具體元素

[ConcreteAcceptor] 宣告「已經找過 XXX」的 visit(XXX) 方法。visit(XXX) 是處理 XXX 的方法，實際原始碼寫在 ConcreteVisitor 裡。

- ConcreteVisitor [具體的訪客] 參與者

ConcreteVisitor 參與者實作 Visitor 參與者。它必須實作每個 visit(XXX)，敘述各個 ConcreteAcceptor 參與者的處理。

- Acceptor [接受者] 參與者
Acceptor 參與者是表示 Visitor 參與者的訪問對象。宣告接受訪客的 accept 方法。Visitor 參與者則被傳遞給 accept 方法的引數。
- ConcreteAcceptor [具體的接受者] 參與者
ConcreteAcceptor 參與者實作 Acceptor 參與者。
- ObjectStructure [物件結構] 參與者
ObjectStructure 參與者是處理 Acceptor 參與者的集合的參與者。讓 ConcreteVisitor 參與者有可處理所有 Acceptor 參與者的方法。

4.3.7 範例

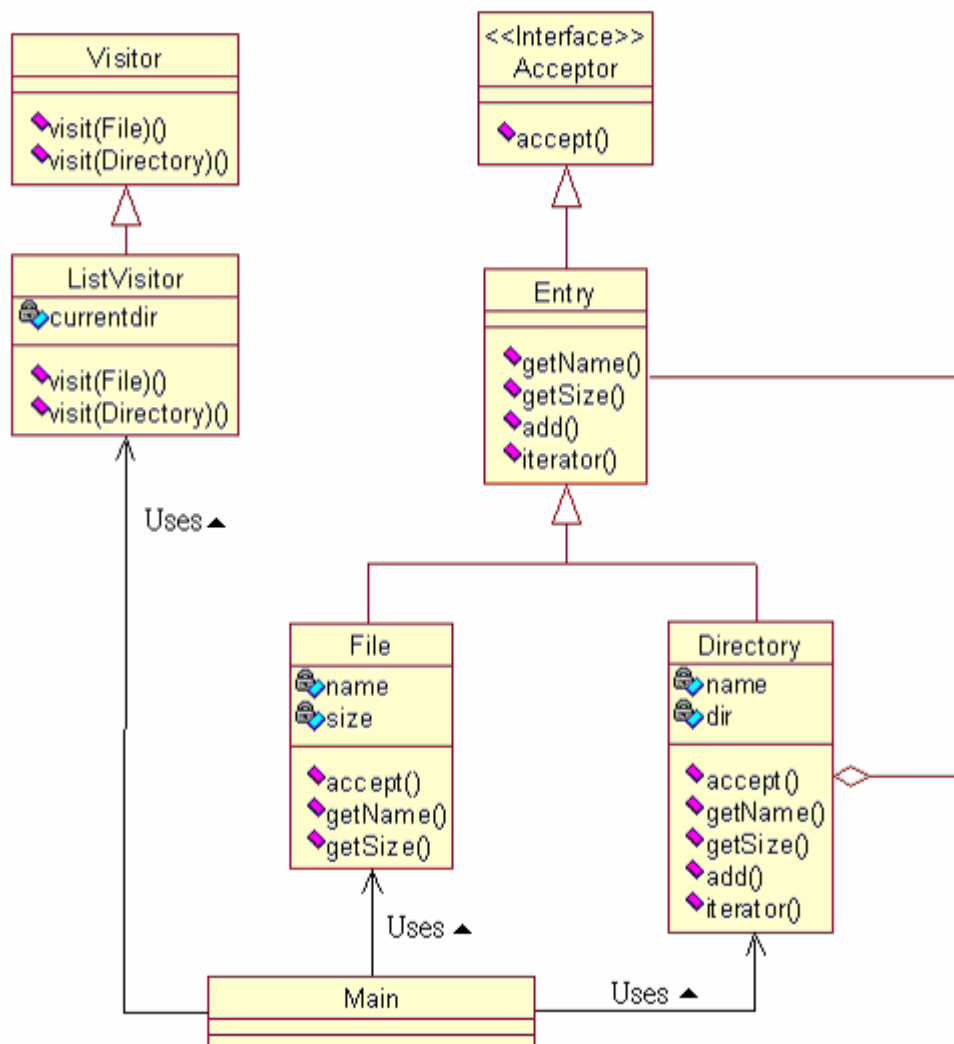
以下是本範例的類別與介面一覽表

名稱	說明
Visitor	表示訪問檔案或目錄的訪客之抽象類別
Acceptor	表示接受 Visitor 類別的物件個體的資料結構之介面
ListVisitor	Visitor 類別的子類別，列印檔案或目錄總覽之類別
Entry	File 和 Directory 的父類別的抽象類別 〔有時作 Acceptor 介面〕
File	表示檔案的類別
Directory	表示目錄的類別
FileTreatmentException	發生在對 File 進行 add 時的例外類別
Main	動作測試用的類別

(表十二 範例的類別)

以下是本範例的類別圖

程式範例的類別圖



(圖十七 程式範例的類別圖)

〔完整程式碼部分請見『Design Patterns 於 Java 語言上的實習應用 ----- 結城浩 著』一書。〕

因為這個範例所「Visit」的資料結構就是 Composite Pattern。所以把它們放在一起說明，以下是兩個 Pattern 的各別說明：

Composite Pattern 的部分：

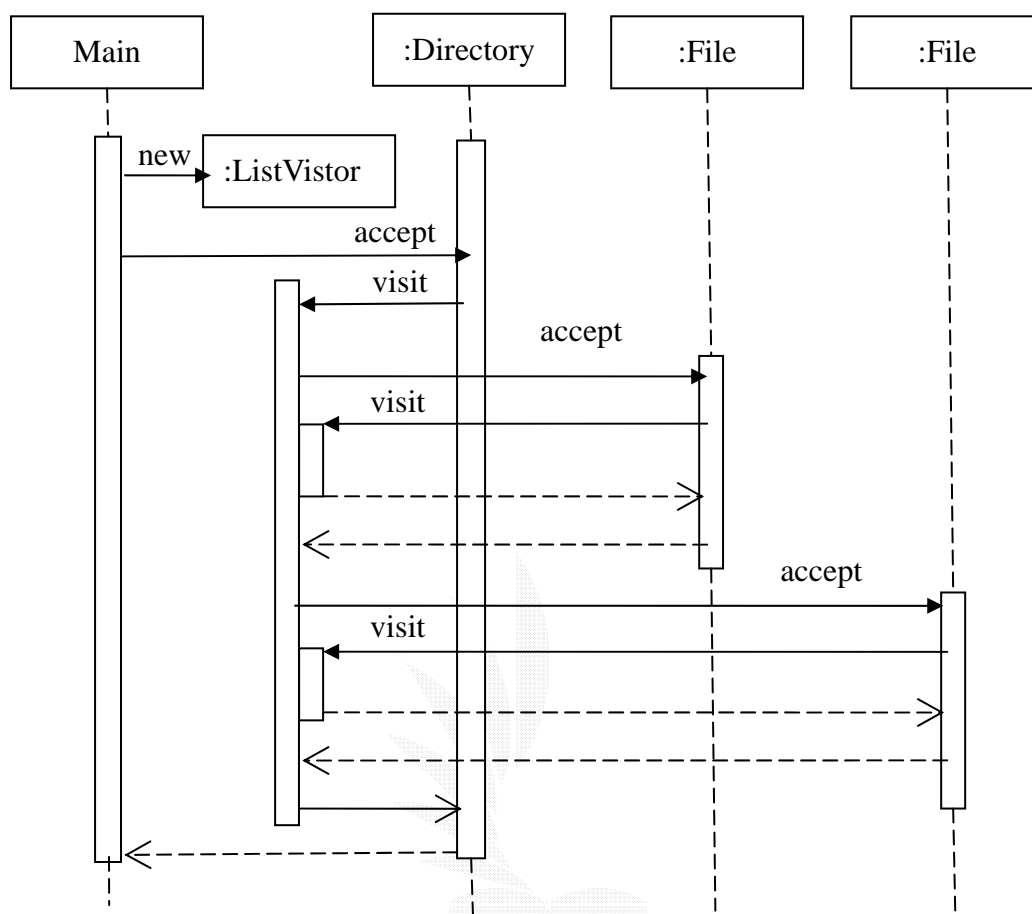
包括了 Entry 虛擬類別，File 類別，Directory 類別。File 類別和 Directory 類別的父類別都是 Entry 虛擬類別。Directory 還可聚合其他的 Entry 類別，也就是 Directory 可以放入 File 也可以放入 Directory 類別。這時，類別的操作就可以把 File 和 Directory 一視同仁，例如雖然 Directory 中的 getSize() 方法

```
Public int getSize() {  
    int size = 0;  
    Iterator it = dir.iterator();  
    While (it.hasNext()) {  
        Entry entry = (Entry)it.next();  
        Size += entry.getSize();  
    }  
    return size;  
}
```

Size += entry.getSize() 這一行，呼叫 entry 的 getSize 類別，但是實際上呼叫的是依照當時 entry 是參考到什麼類別，如果是 File 類別，則呼叫 File 類別的 getSize() 方法〔單純傳回 size 欄位的值〕。若參考到的是 Directory 類別，則呼叫的是上方的程式碼〔傳回整個目錄的 size〕。這裡我們同樣是呼叫 entry.getSize()，一視同仁的做法，就是 Composite Pattern 帶來的好處。

Visitor Pattern 的部分：

光是看類別圖很難體會 Visitor Pattern，如果加上流程圖會更好說明整個過程。



(圖十八 程式範例的流程圖)

此處理流程圖是假設在 1 個目錄下只有 2 個檔案。Main 對 Directory 的物件個體呼叫 accept 方法。Main 產生的 ListVisitor 物件個體會傳遞給引數，不過在途中沒有表示出來。

Directory 的物件個體則呼叫引數所傳遞過來的 ListVisitor 物件個體的 visit(Directory) 方法。ListVistor 接著就會照著 visit(Directory) 的程式碼，開始在目錄中尋找並呼叫第一個檔案的 accept 方法，把自己本身〔this〕傳遞給引數。

File 的物件個體則呼叫引數所傳遞過來的 ListVisitor 的 visit(File) 方法。則 ListVisitor 就可對這個 File 物件個體進行處

理。〔此時，ListVisitor 的 visit(Directory)還在呼叫紀錄〔call stack〕裡。〕

當 visit(File)及 accept 回傳後，接著再呼叫另一個 File 物件個體的 accept 方法，把 ListVisitor 自己本身〔this〕傳遞給引數。

接著和之前一樣，File 物件個體呼叫 visit(File)。當 visit(File)、accept 回傳後，visit(Directory)以及 Main 呼叫的 accept 也回傳了。

由此可知，「處理」的部分都集中在 ListVistor 物件個體上。達到了資料結構和處理分離的目的。



第五章 系統評估與討論

5.1 遭遇困難

不管是哪部分都有著相當難度，單指 APPLICATION 部分，由於是以介面為考量，故設計時都以使用者的觀點來想程式，但常常遇到使用上和設計上相衝的情況發生，例如在使用者的觀點，復原和重做是在普遍不過的功能，然而設計上時就無法如此容易，因為上一次的動作這樣的抽象概念，可能衍生出非常多的可能，程式難以掌控，好在 JAVA 有專屬的類別，專門處理這類的動作，故後來才能輕鬆的解決問題，而在網頁設計部分上，由於還要考慮到系統的安全性，使得顧慮相當得多，如學生登入和教授的權限就極為不同，必須要有完整的區隔才行；還有當兩人用同一台電腦連入系統時，有可能造成後者，利用前者的身分登錄；或者有人直接在 URL 內容上填入網頁位置和參數，也有可能造成系統的破壞…等等。使得為了顧慮安全上的考量，延遲了不少進度。除此之外，效能上的考量也是重要的議題，由於要多人連線使用，I/O 和資料庫連線的動作絕對不能太多，然而這兩點又是大量資料取得的最快方法，在效率與功能上的取捨，又是讓我們困擾不已。

在核心分析程式功能，我們用到了一些 COMPILER 技術，雖然不至於太過高深，但沒有中文文件，也使的困難加劇，本以為簡單的功能，卻都遲遲做不出來，使的進度不斷延長。

其他問題，像是伺服器的安裝、設定，網頁圖示的設計，版面的配置，雖不是專題的重點，卻也是無法忽視的部分，也還是要廢些苦工才能解決，當稍有成果時，以為能稍微鬆一口氣，卻發現報告的編寫也是一件累人的事，雖然也已閱讀不少資料，然而要如何呈現，去蕪存精，不參廢話還是相當令人頭痛。

5.2 心得與經驗

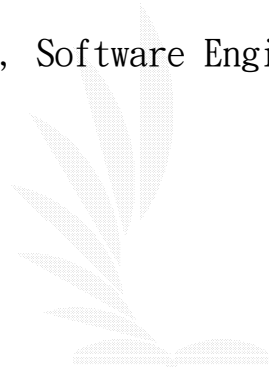
由於所有部份都由 Java 寫成，對於 Java 程式設計和架構，都有更深入的了解，在真正撰寫程式，發現 Java 最為有力的地方，就是其函式架構，由於 Java 有特別的類別設計，像 package、interface 對於類別有開放，但又不雜亂的管理，使得程式的編寫較好發揮，還有在 Jsp 網頁上用的 Java bean，又可說是另一高度彈性的做法，使得程式的再用大為提升。

除此之外，最主要還是軟體的熟悉，從伺服繫軟體，如 Apache、Tomcat，動態轉址軟體:NO-IP，資料庫系統:MySQL，MySQL 資料庫管理軟體:phpAdmin，PHP，Java SDK…等。每一項都是以前從未碰過的，



參考資料

- [1]林智楊 范明翔 陳錦輝,精通Java Swing程式設計, 金禾資訊股份有限公司, Nov. 2001
- [2]鄭吉峰, Java Server Page觀念與應用實務, 學貫行銷股份有限公司, Nov. 2001
- [3]江義華, Java優質學習篇, 金禾行資訊銷股份有限公司, Nov. 2001
- [4]結城浩, Design Patterns於Java語言上的實習應用, 博碩文化, Feb. 2002
- [5]IAN SOMMERVILLE, Software Engineering, ADDISON WESLEY, 2001



附錄

附錄 A 工作分配表

	吳俊逸	林志遠
資料收集	◎◎	◎◎◎◎
製作規格書	◎◎◎	◎◎
工具學習	◎◎	◎◎◎
設計資料庫建立	◎◎◎	◎
設計使用介面	◎◎◎◎◎	
系統建立	◎◎	◎◎◎◎◎
報告編寫與校定	◎◎◎	◎◎◎
系統發表	◎◎	◎◎

注：◎愈多表示工作越多

(表十三 工作分配表)

附錄 B 安裝步驟

一、安裝前準備

你必須有以下軟體：

Apache 2.0.45：[apache_2.0.45-win32-x86-no_ssl.msi](#) (*基本必須)

MySQL 3.23.56：[mysql-3.23.56-win.zip](#) (要使用資料庫必須)

PHP 4.3.1：[php-4.3.1-Win32.zip](#) (要跑 PHP 必須，只跑 jsp 則免)

phpMyAdmin 2.5.0：[phpMyAdmin 2.5.0](#) (用 PHP 寫的 MySQL 資料庫管理程式)

Tomcat 4.1.24：[tomcat-4.1.24.exe](#) (要跑 JSP 必須，只跑 PHP 則免)

mod_jk_2.0.45.dll：[mod_jk_2.0.45.dll](#) (要跑 JSP 必須，只跑 PHP 則免)

二、安裝開始

Apache 2.0.45 安裝：

滑鼠雙點 [apache_2.0.45-win32-x86-no_ssl.msi](#) 開始安裝

滑鼠按一下 Next >

滑鼠點選 I accept the terms in the agreement，然後滑鼠按一下 Next >

滑鼠按一下 Next >

Network Domain & Server Name 都填入 localhost (除非你有固定 IP 才填你的 IP 位址)

Administrator's Email Address 填入你的 Email 信箱，然後滑鼠按一下 Next >

滑鼠點擇 Typical 典型的安裝，然後滑鼠按一下 Next >

Apache 安裝的路徑(建議照預設值)，然後滑鼠按一下 Next >

滑鼠按一下 Install 開始安裝

滑鼠按一下 Finish，恭喜您！ Apache 安裝完成。

MySQL 3.23.56 安裝：

解壓縮 mysql-3.23.56-win

進入解壓縮出來的目錄裡執行 SETUP.EXE

一路照預設值安裝底 Next > Next > Next >.....

最後滑鼠按一下 Finish，完成安裝

到你安裝 MySQL 的目錄 C:\mysql\bin 裡，執行 winmysqladmin.exe 管理程式

第一次啟動會要求你輸入管理者帳號密碼，輸入後按確定即可

不要懷疑！你成功架設 MySQL 資料庫了！（預設安裝在你的 C:\mysql）

PHP 4.3.1 安裝：

解壓縮 php-4.3.1-Win32.zip 到 c:\PHP

檔案總管切換到目錄 c:\php 將裡面的 php.ini-dist 檔名改為 php.ini

拷貝 php.ini 到 WIN2000 的系統目錄下（預設是 C:\WINNT）

用筆記本開啟 php.ini

查找 doc_root 在等號後加入 C:\Program Files\Apache Group\Apache2\htdocs

再查找 extension_dir 將 ./ 改成 C:\php 完成後存檔

拷貝 php4ts.dll 到 WIN2000 的系統安裝目錄下的 system32 (預設是 C:\WINNT\system32)

開啟 Apache 的配置檔 httpd.conf
([開始] -> 程式集 -> Apache HTTP Server 2.0.45 -> Configure Apache Server -> Edit the Apache httpd.conf Configuration File)

查找 DirectoryIndex(設定預設首頁,你可以更改前後順序)
將 index.php 加到 DirectoryIndex 後面,記得中間要空一格喔

修改後會看起來會像這樣 DirectoryIndex X X X X X. X
X X X X X X X. X X X index.php

於配置檔最後面加入以下內容後存檔關閉
LoadModule php4_module c:/php/sapi/php4apache2.dll
AddType application/x-httpd-php .php

重新啟動 Apache 之後就完成了

phpMyAdmin-2.4.0 安裝:

解壓縮 phpMyAdmin-2.5.0-php.zip 至 C:\Program Files\Apache Group\Apache2\htdocs\phpMyAdmin

進入解壓縮出來的目錄 phpMyAdmin 裡執行修改 config.inc.php

查找 \$cfg['PmaAbsoluteUri'] = ''; 改成
\$cfg['PmaAbsoluteUri'] =
'http://localhost/phpMyAdmin/';

開啟瀏覽器,輸入網址
http://localhost/phpMyAdmin/index.php

Ya 成功!好好玩玩資料庫吧~

J2SDK, SE 1.4.1_02 安裝:

滑鼠雙點 2sdk-1_4_1_02-windows-i586.exe 開始安裝

一路照預設值安裝底 Next > Next > Next >.....

最後滑鼠按一下 Finish 完成安裝!

接著配置系統環境變數：

在我的電腦上按滑鼠右鍵 --> 內容 --> 進階 --> 環境變數

編修"系統變數"，記得用 ; 分隔

編輯變數名：path

編輯變數值：C:\j2sdk1.4.1_02\bin

新增變數名：JAVA_HOME

新增變數值：C:\j2sdk1.4.1_02

新增變數名：CLASSPATH

新增變數

值：.\;C:\j2sdk1.4.1_02\lib\dt.jar;C:\j2sdk1.4.1_02\lib\tools.jar;

重開機使配置的系統環境變數生效

重開機後 J2SDK 已安裝在你的電腦了！（預設安裝在你的 C:\j2sdk1.4.1_02）

tomcat-4.1.24 安裝：

滑鼠雙點 tomcat-4.1.24.exe 開始安裝

安裝程式會找到你安裝 J2SDK 的目錄，按 確定 進行下一步

出現版權說明，按 I Agree

下拉式選單選 Full (w/ Source Code) 完全安裝，按 Next >

Tomcat 安裝的路徑（建議照預設值），滑鼠按一下 Install 開始安裝

安裝最後會要求你設定連接埠 8080 以及請你自訂管理者帳號密碼，滑鼠按一下 Next 完成安裝

Close 安裝程式，就這樣！Tomcat 安裝完成！***如果你不想和 Apache 做整合，到這步驟即可（直接到 C:\Program Files\Apache Group\Tomcat 4.1\bin 目錄下運行 startup.bat 啟動 Tomcat），就已經可以運行 JSP 檔了（使用 8080 埠）；雖然 Tomcat 有內建 Apache 的 HTTP 服務，但專長還是對 JSP 程式有比較好的執行效率和性能，處理靜態頁面的速度不如 Apache，所以為了建構完美的 WEB Server，我們繼續將 Apache 和 Tomcat 結合在一起！

開啟 Apache 的配置檔 httpd.conf

（ [開始] -> 程式集 -> Apache HTTP Server 2.0.45 -> Configure Apache Server -> Edit the Apache httpd.conf Configuration File）

查找 DirectoryIndex （設定預設首頁，你可以更改前後順序）

將 index.jsp 加到 DirectoryIndex 後面，記得中間要空一格喔

修改後會看起來會像這樣 DirectoryIndex X X X X X. X
X X X X X X X X X index.jsp

於配置檔最後面加入以下內容後存檔關閉

```
#  
# Load mod_jk  
#  
<IfModule !mod_jk.c>  
LoadModule jk_module modules/mod_jk_2.0.45.dll  
</IfModule>
```

```
#  
# Configure mod_jk  
#  
JkWorkersFile "C:\Program Files\Apache
```

```
Group\Tomcat 4.1\conf\worker.properties"
JkLogFile logs/mod_jk.log
JkLogLevel info
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"

#To open as a web folder
JkMount /examples ajp13
JkMount /examples/* ajp13
JkMount /*.jsp ajp13
JkMount /*/servlet/ ajp13
```

用筆記本新增一個名為 worker.properties 的配置檔
(注意!存檔時不要存成 worker.properties.txt) ,
複製/貼上 以下紅色內容

#注意!!

**#如果你不是照預設值安裝的話，請先修改好再複製/貼
上 以下紅色內容**

#以下兩個路徑請改為你安裝的路徑

#workers.tomcat_home 是你安裝 Tomcat 的路徑

#workers.java_home 你安裝 J2SDK 的路徑

#範例如下

```
#workers.tomcat_home=C:\Program Files\Apache
Group\Tomcat 4.1
```

```
#workers.java_home=C:\j2sdk1.4.1_02
```

將此 worker.properties 配置檔存在 C:\Program
Files\Apache Group\Tomcat 4.1\conf 目錄裡

接著用筆記本開啟 server.xml Tomcat 核心檔 (預設是在
C:\Program Files\Apache Group\Tomcat 4.1\conf)

將

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
```

```
<!--
```

```
<Connector
```

```
className="org.apache ajp.tomcat4.Ajp13Connector"
```

```
port="8009" minProcessors="5" maxProcessors="75"
```

```
acceptCount="10" debug="0" />
```

```
-->
```

的<!-- --> 去掉，因為預設的 Tomcat 沒有啟用 Ajp13 的 8009 的埠給 mod_jk 使用，修改後如下：

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
```

```
<Connector
```

```
className="org.apache ajp tomcat4. Ajp13Connector"
```

```
port="8009" minProcessors="5" maxProcessors="75"
```

```
acceptCount="10" debug="0" />
```

將

```
<!-- Tomcat Root Context -->
```

```
<!--
```

```
<Context path="" docBase="ROOT" debug="0" />
```

```
-->
```

的<!-- --> 去掉，修改後如下：

```
<!-- Tomcat Root Context -->
```

```
<Context path="" docBase="ROOT" debug="0" />
```

這個 Tomcat Root Context 是設定網址的對應路徑，

所以日後放置檔案的路徑如下：

(1) 將 JSP 檔案放在對應路徑或者其下層子目錄（也就是 C:\Program Files\Apache Group\Tomcat 4.1\webapps\ROOT）。

(2) 將 *.class 檔案放在對應路徑的 WEB-INF\classes\子目錄底下（也就是 C:\Program Files\Apache Group\Tomcat 4.1\webapps\ROOT\WEB-INF\classes）。

(3) 將 *.jar 檔案放在對應路徑的 WEB-INF\lib\子目錄底下（也就是 C:\Program Files\Apache Group\Tomcat 4.1\webapps\ROOT\WEB-INF\lib）。

將這個 Tomcat 核心檔 server.xml 存檔

現在你的 Tomcat 和 Apache 的設定就完成了。

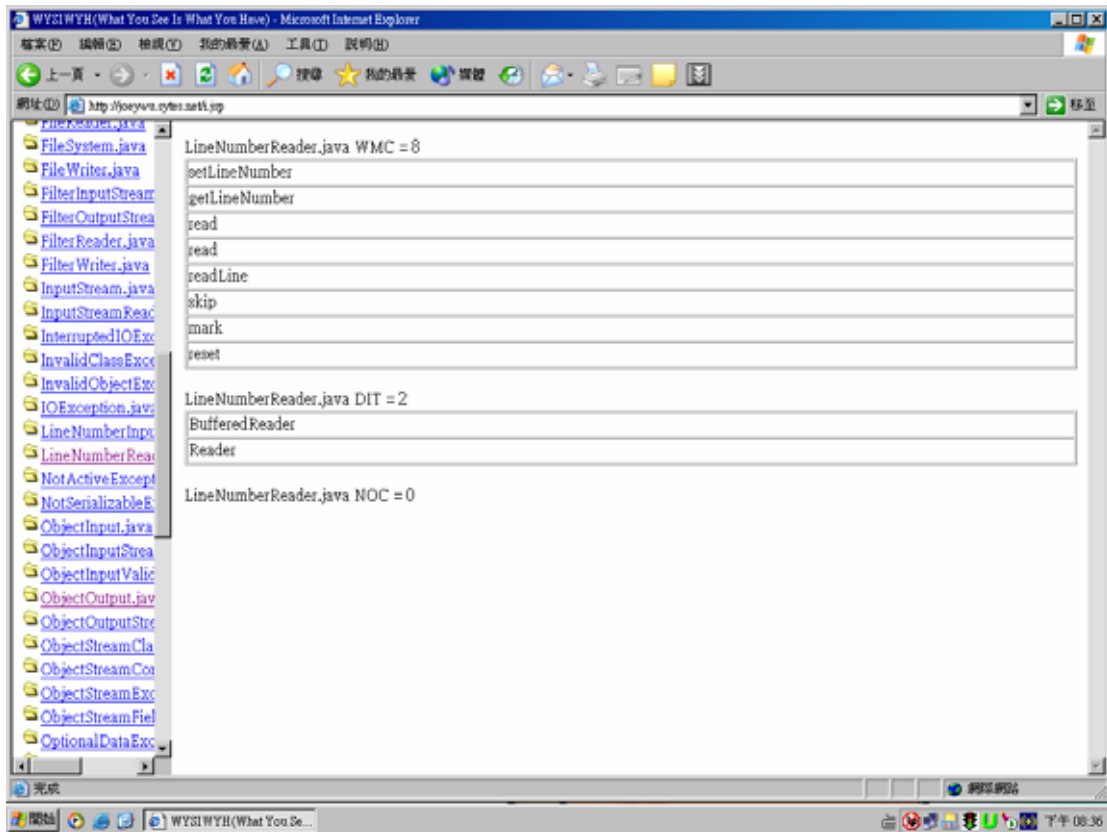
mod_jk_2.0.45.dll 安裝：

將下載的 mod_jk_2.0.45.dll 複製到 C:\Program Files\Apache Group\Apache2\modules 即可

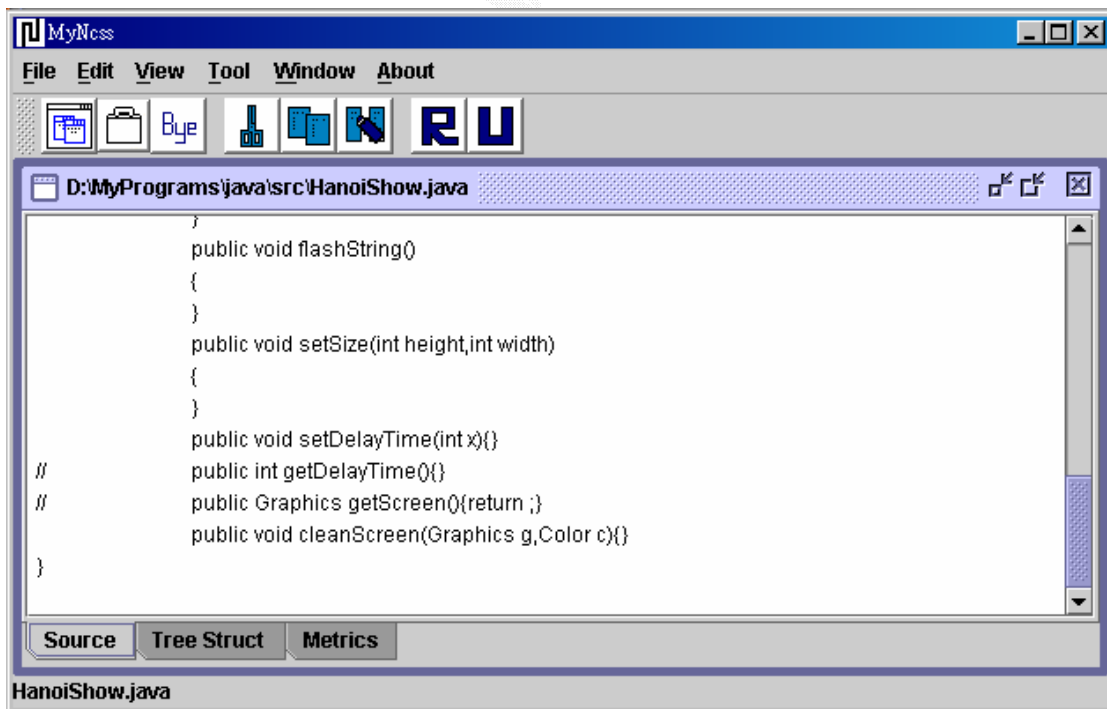
重新啟動電腦即大功告成~



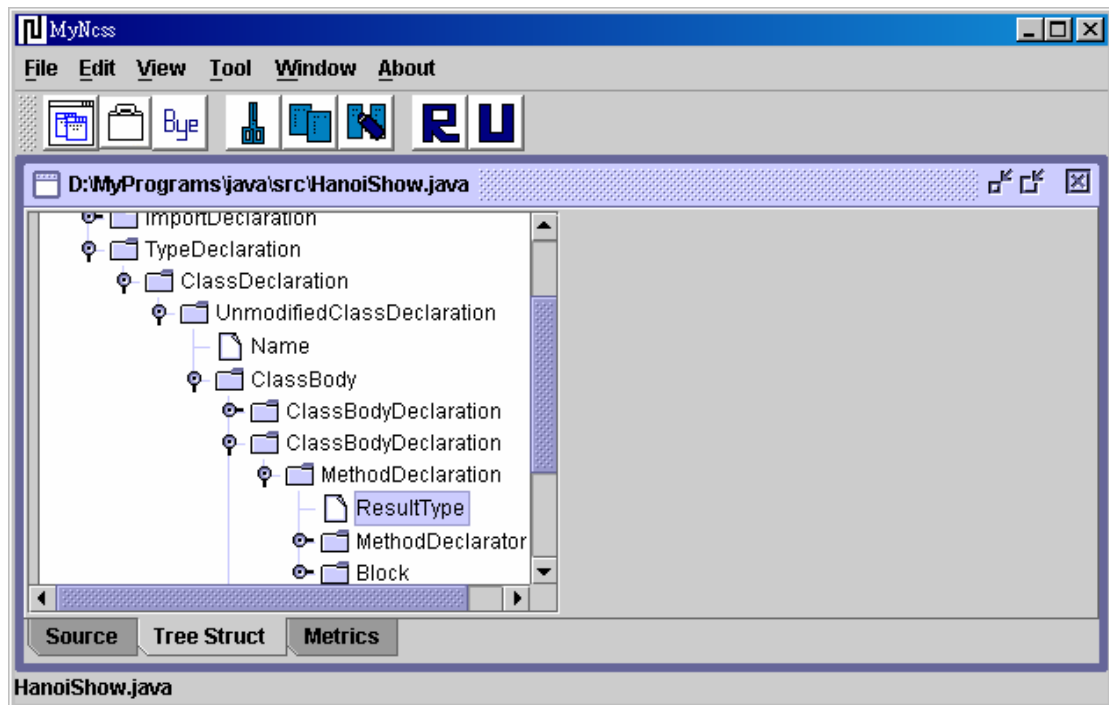
附錄 C 執行結果



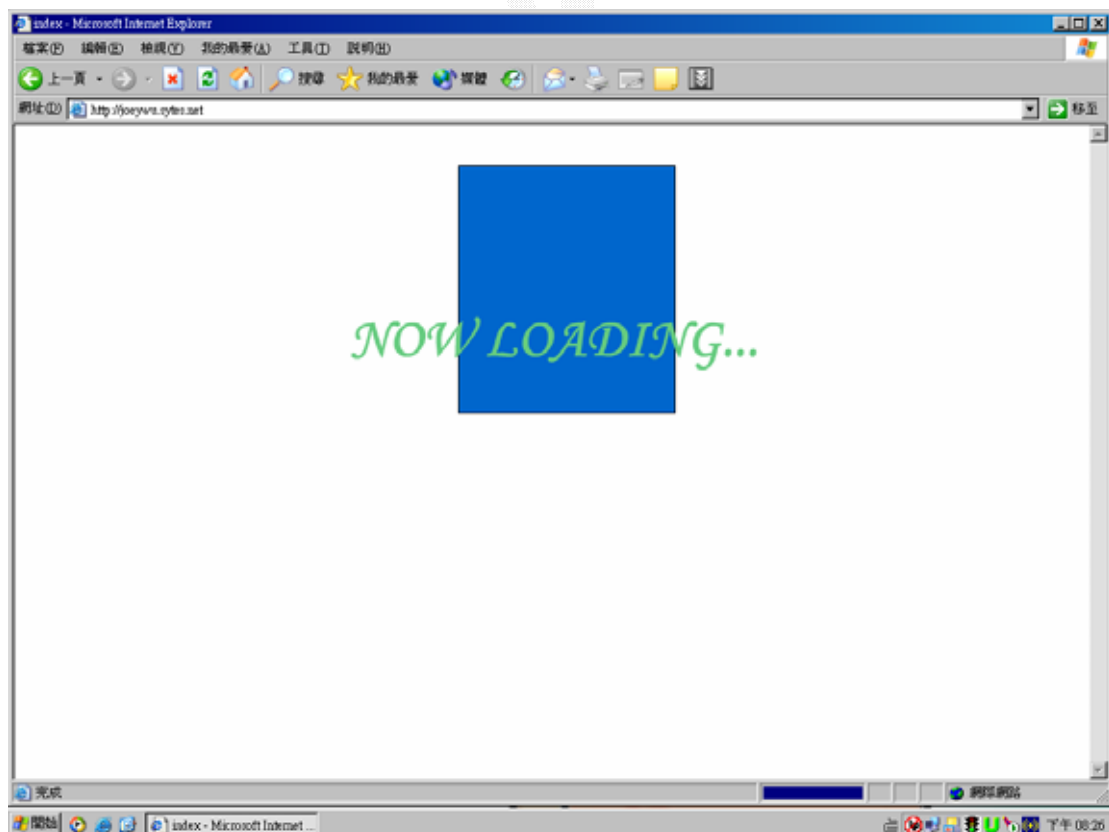
(圖十九 執行畫面一)



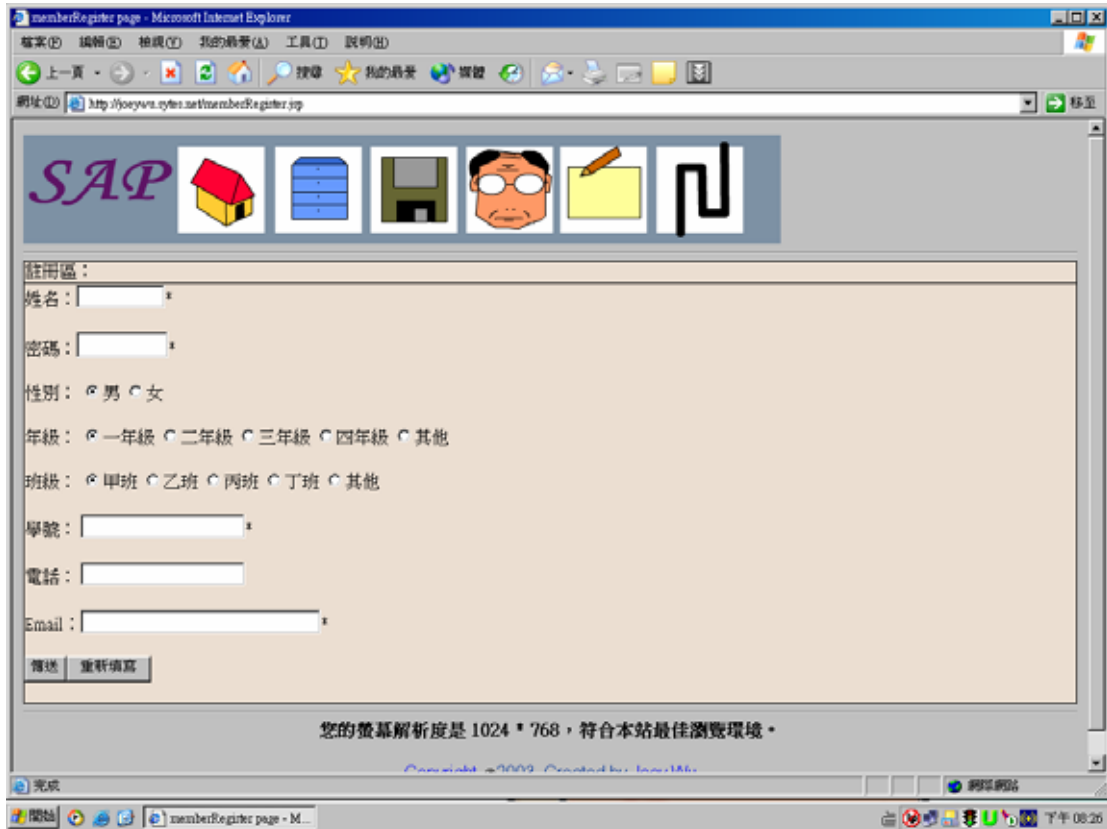
(圖二十 執行畫面二)



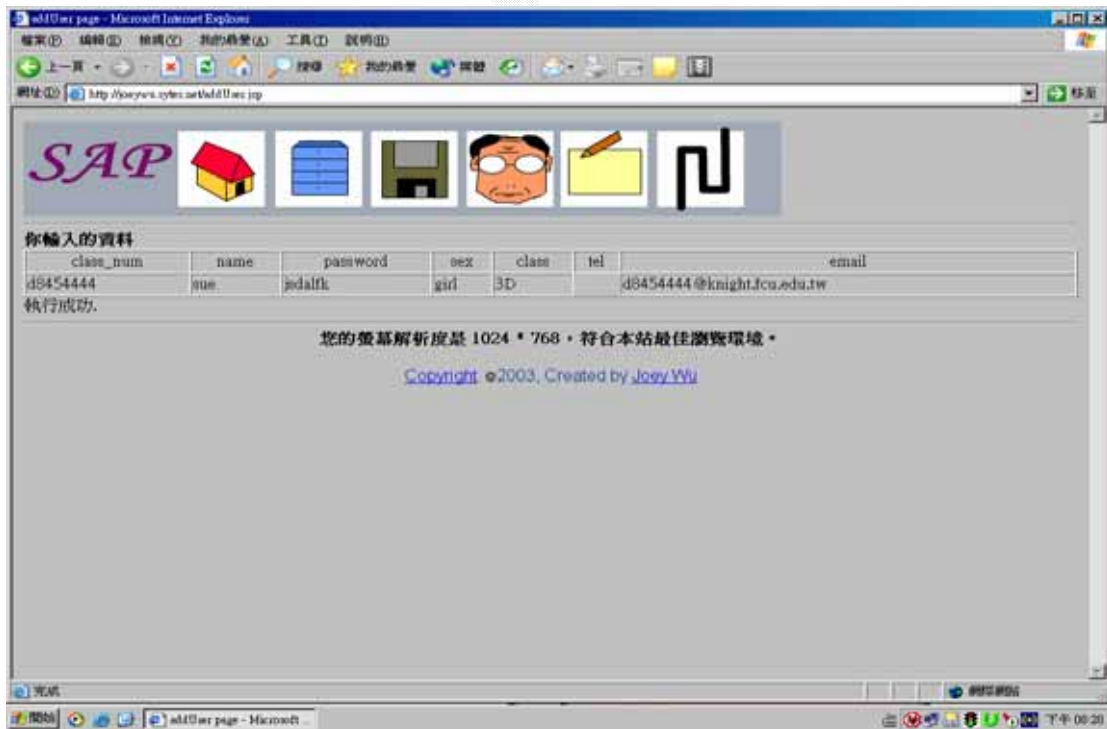
(圖二十一 執行畫面三)



(圖二十二 執行畫面四)



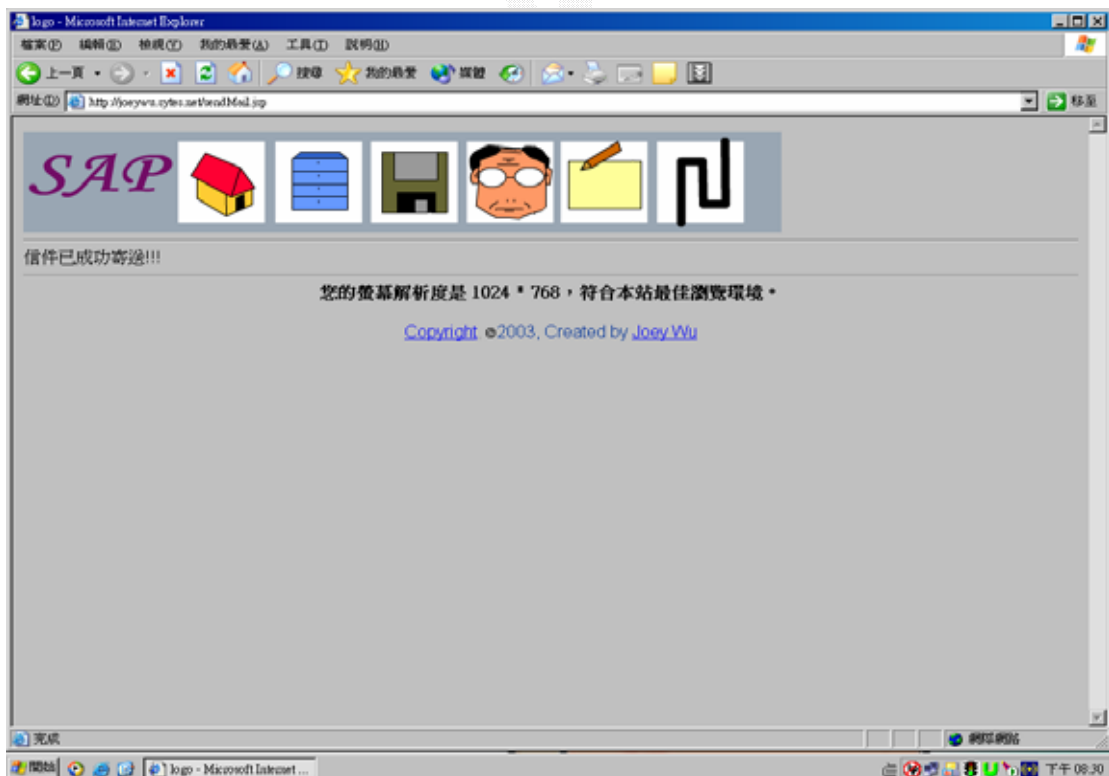
(圖二十三 執行畫面五)



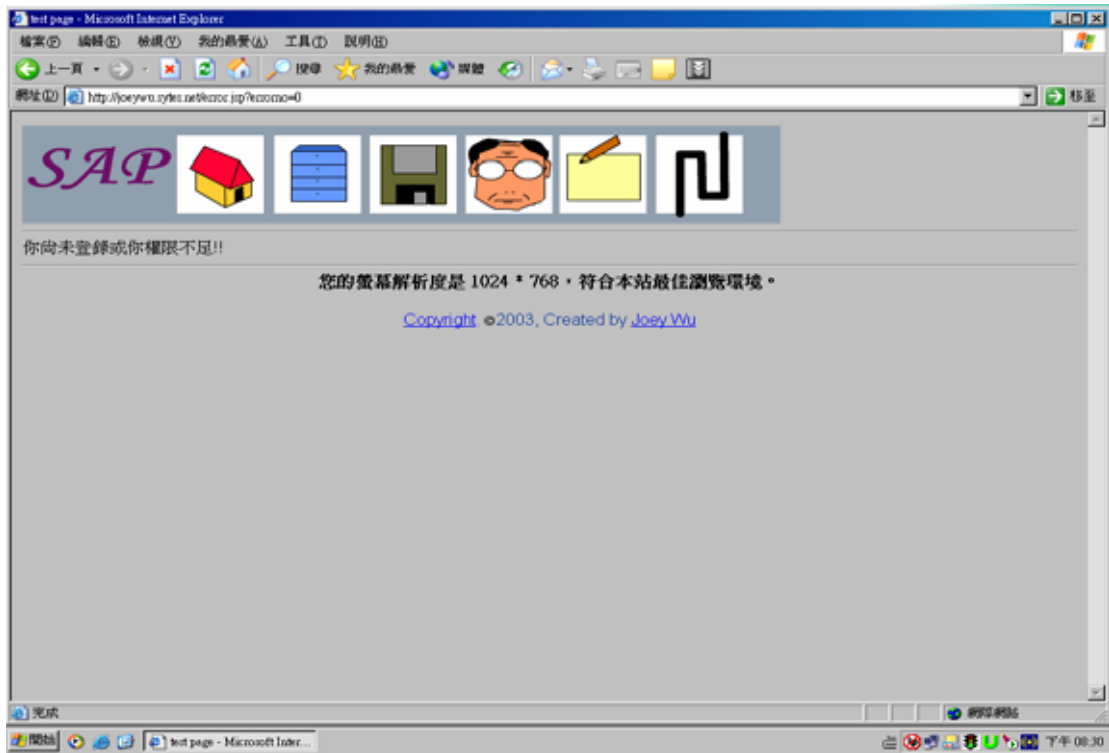
(圖二十四 執行畫面六)



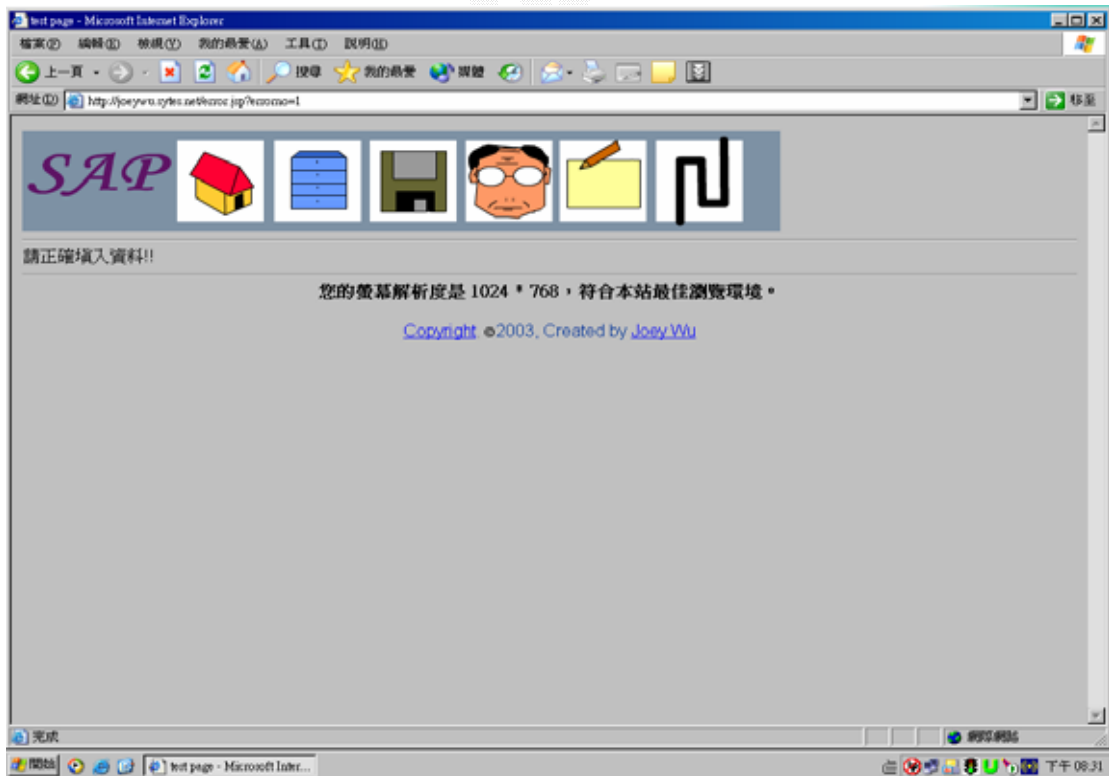
(圖二十五 執行畫面七)



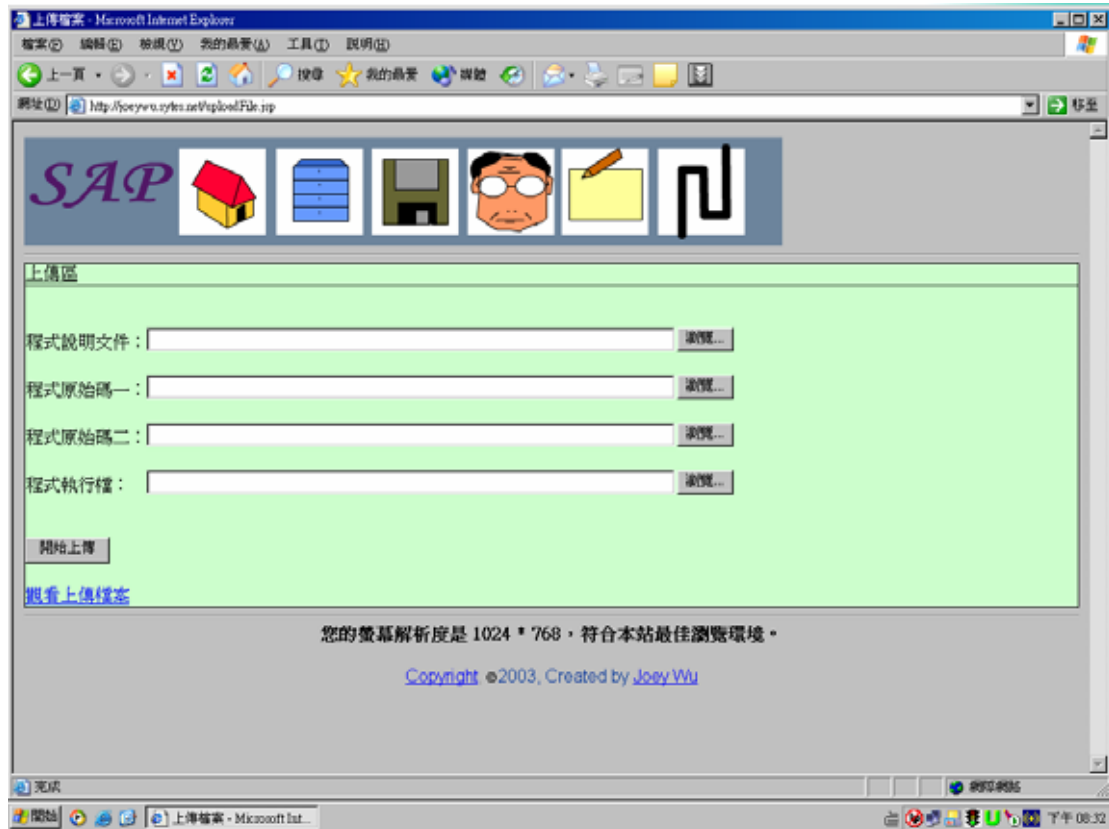
(圖二十六 執行畫面八)



(圖二十七 執行畫面九)



(圖二十八 執行畫面十)



(圖二十九 執行畫面十一)