**國立中央大學**

**資　訊　管　理　學　系**

**108（一）系　統　分　析　與　設　計**

**系統軟體分析規格書**

第14組

資管三A 106403518 呂文楷

資管三A 106403015 華崧淇

資管三A 106403201 陳威捷

企管四B 105401530 余函倩

資管三A 106403505 蔡苑萍

資管三A 106403007 蔣書珊

指導教授：許智誠　教授、陳以錚　教授

<分析修改部分>

第一章：

1.2 系統範圍： 7個模組修正成6個，刪除金流模組

第二章：

2.1 修正： 4個actor改為5個actor、7個子系統改為6個子系統、共有27個使用案例(金流子系統刪除，增加金融服務actor。

使用案例圖修改)

2.2 使用案例的編號修正、訂票子系統的表1修改、活動圖與強軔圖名稱修正

2.3 使用案例的編號修正、供應商子系統表2修改、活動圖與強軔圖名稱修正

第三章：資料庫設計

會員資料表、票券資料表、訂單資料表、演唱會資料表、管理者資料表、供應商資料表修正

第四章：

類別圖修正

第五章：

5.1 系統架構圖內容修正(刪除金流模組)

# 目錄

[目錄 iii](#_Toc25783840)

[表目錄 v](#_Toc25783841)

[圖目錄 vi](#_Toc25783842)

[版本修訂 1](#_Toc25783843)

[第1章 簡介 2](#_Toc25783844)

[1.1 文件目的 2](#_Toc25783845)

[1.2 系統範圍 2](#_Toc25783846)

[1.3 參考文件 3](#_Toc25783847)

[1.4 文件架構 3](#_Toc25783848)

[第2章 系統動態分析圖 4](#_Toc25783849)

[2.1 使用案例圖 4](#_Toc25783850)

[2.2 使用案例1.0：訂票子系統模組 1](#_Toc25783851)

[2.2.1 使用案例1.2：購票 2](#_Toc25783852)

[2.2.1.1 活動圖 2](#_Toc25783853)

[2.2.1.2 強韌圖 3](#_Toc25783854)

[2.3 使用案例2.0：供應商子系統 6](#_Toc25783858)

[2.3.1 使用案例3.1：新增票券資料 7](#_Toc25783859)

[2.3.1.1 活動圖 7](#_Toc25783860)

[2.3.1.2 強韌圖 8](#_Toc25783861)

[第3章 資料庫設計 12](#_Toc25783866)

[第4章 類別圖 16](#_Toc25783867)

[第5章 系統開發環境 17](#_Toc25783868)

[5.1 系統架構圖 17](#_Toc25783869)

[5.2 MVC架構 18](#_Toc25783870)

# 表目錄

[表 1：商業流程編號3.0訂票子系統模組 6](#_Toc29058549)

[表 2：商業流程編號7.0供應商子系統模組 9](#_Toc29058550)

[表 3：分析階段之會員資料表（members）資料結構 12](#_Toc29058551)

[表 4：分析階段之商品資料表（ticket）資料結構 12](#_Toc29058552)

[表 5：分析階段之訂單資料表(order)資料結構 13](#_Toc29058553)

[表 6：分析階段之演唱會資料表(concert)資料結構 13](#_Toc29058554)

[表 7：分析階段之管理者資料表(manager)資料結構 14](#_Toc29058555)

[表 8：分析階段之供應商資料表(supplier)資料結構 14](#_Toc29058556)

# 圖目錄

[圖 1：搶票系統使用案例圖 5](#_Toc29058434)

[圖 2：使用案例3.1購票之活動圖 8](#_Toc29058435)

[圖 3：使用案例3.1購票之強韌圖 9](#_Toc29058436)

[圖 4：使用案例7.1新增票券資料之活動圖 10](#_Toc29058437)

[圖 5：使用案例7.1新增票券資料之強韌圖 11](#_Toc29058438)

[圖 6：實體關係圖 12](#_Toc29058439)

[圖 7：分析階段之類別圖 15](#_Toc29058440)

[圖 8：分析階段之系統架構圖 16](#_Toc29058441)

[圖 9：MVC架構 17](#_Toc29058442)

# 版本修訂

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修訂者 | 修訂簡述 | 日期 |
| V0.1.0 | 十四組 | Draft | 2019/11/28 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 第1章 簡介

軟體分析規格書（software analysis description，SAD）係依據軟體產品、專案之主要使用者之需求規格文件（software requirements specification，SRS），主要用於描述邏輯之軟體架構與系統範圍之文件。藉由本文件得以分析軟體系架構之目的，並作為軟體設計階段之依據。

本專案之文件採用統一塑模語言（Unified Modeling Language，UML）說明與建構本系統之方法與架構，包含：使用案例圖（Use Case Diagram）、泳道圖（Swim-lane）與循序圖（Sequence Diagram）等。

## 1.1 文件目的

本文件之目的用於提供軟體系統開發人員分析之規範與藍圖，透過軟體分析規格書，開發人員可以明確了解軟體系統之邏輯與運作方式，並得以此為據遵照共同訂定之規格設計軟體系統。

本文件針對系統之分析為邏輯階段（logical phase）而非是實際設計階段（physical phase）之內容，分析模型與系統設計與實作環境無關之邏輯結構（logical Structure），得以使用邊界、控制和實體物件呈現系統資訊（information）、行為（behavior）和展示（presentation）三個層面。

## 1.2 系統範圍

本系統範圍用於搶票系統，其中主要包含會員、訂單、訂票、管理者、與供應商、票券等六個模組，並且能進行相關新增、查閱與維護工作，藉由此系統支持完成搶票系統所需的管理流程。詳細各模組之功能與內容可參閱第一份文件系統軟體需求規格書。

## 1.3 參考文件

1. 系統分析與設計—需求（Requirement）

## 1.4 文件架構

本文件共分為五個章節，用以闡述本專案之分析相關內容：

1. 第1章針對本文件進行簡介，說明本文件重要之處。
2. 第2章依據本專案前份文件之使用者案例依序進行分析，於本章節依照使用者案例將產出所需活動圖與強韌圖。
3. 第3章分析本專案所需之資料庫架構與資料表內容。
4. 第4章則是陳列出本專案所需之類別、屬性與方法的類別圖。
5. 第5章說明本專案所需之系統開發環境，其中包含系統架構圖與MVC架構之說明。

# 第2章 系統動態分析圖

在本章節中，將透過在前一份文件中所分析之使用者案例（use case）逐一進行詳細之系統動態分析。首先須先將使用者案例之主要流程轉換成活動圖，再者依照所分析之活動圖產生強韌圖以找出分析之類別。

## 2.1 使用案例圖

依據第一份文件—系統軟體需求規格書（Software Requirement Specification），本搶票系統預計共有5位動作者與27個使用案例，並依照不同之模組區分成不同子系統共計六個子系統，其中包含以下：會員子系統、票券子系統、訂單系統、訂票子系統、供應商子系統、管理者子系統，下圖（圖 1）為本系統之使用案例圖：

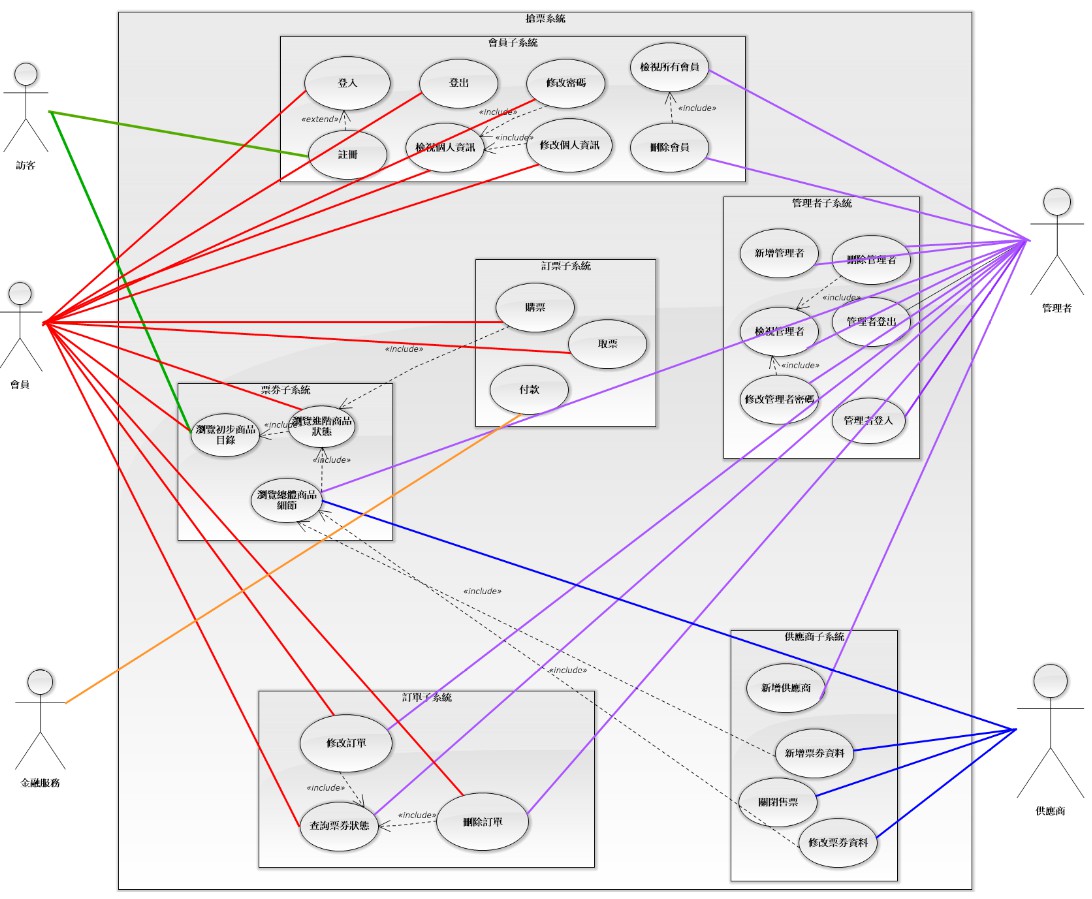


圖 1：搶票系統使用案例圖

根據上圖所分析之使用案例規格，需先逐一將每個使用案例轉換成活動圖與強韌圖，並逐一詳細闡述，並且進行分析。再者依據就前揭所述之活動圖、強韌圖產生所需之通訊圖與狀態機圖，進行統整並繪製出分析階段之循序圖，依照本系統之架構劃分而繪製而成。

## 2.2 使用案例3.0：訂票子系統模組

本章節中，將針對會員模組進行系統動態分析圖之分析，訂票子系統模組一共包含兩個功能：購票、取票，以下並根據此些使用案例進行必要圖形之分析。

表 1：商業流程編號3.0訂票子系統模組

|  |  |  |
| --- | --- | --- |
| 模組 | 功能名稱 | 說明 |
| 3.0 模組：訂票子系統 | 3.1 購票 | 會員可以購票 |
| 3.2瀏覽進階商品狀態 | 讓會員瀏覽票種(價錢、區域)、各票種售票狀態、剩餘票券這三種資訊 |
| 3.3 取票 | 會員可以取票 |

### 2.2.1 使用案例3.1：購票

#### 2.2.1.1 活動圖

* 商業流程編號「3.1購票」之主要流程如下所示：
  + 1. 顯示演唱會清單
  + 2. 點選開始購票
  + 3. 進入選擇日期畫面
  + 4. 顯示演唱會清單，並按下立即訂購
  + 5. 進入選擇票種畫面
  + 6. 輸入票種、數量，按下一步
  + 7. 進入確認個人資訊畫面
  + 8. 輸入個人資料，按下一步
  + 9. 進入選擇付款方式畫面
  + 10. 選擇付款方式，按開始繳費
  + 11. 顯示系統訊息(警示確認)，按確認鍵
  + 12. 進入跳轉畫面
  + 13. 進入付款畫面
* 透過以上主流程可將其轉換成活動圖，如下圖（圖 2）所示：

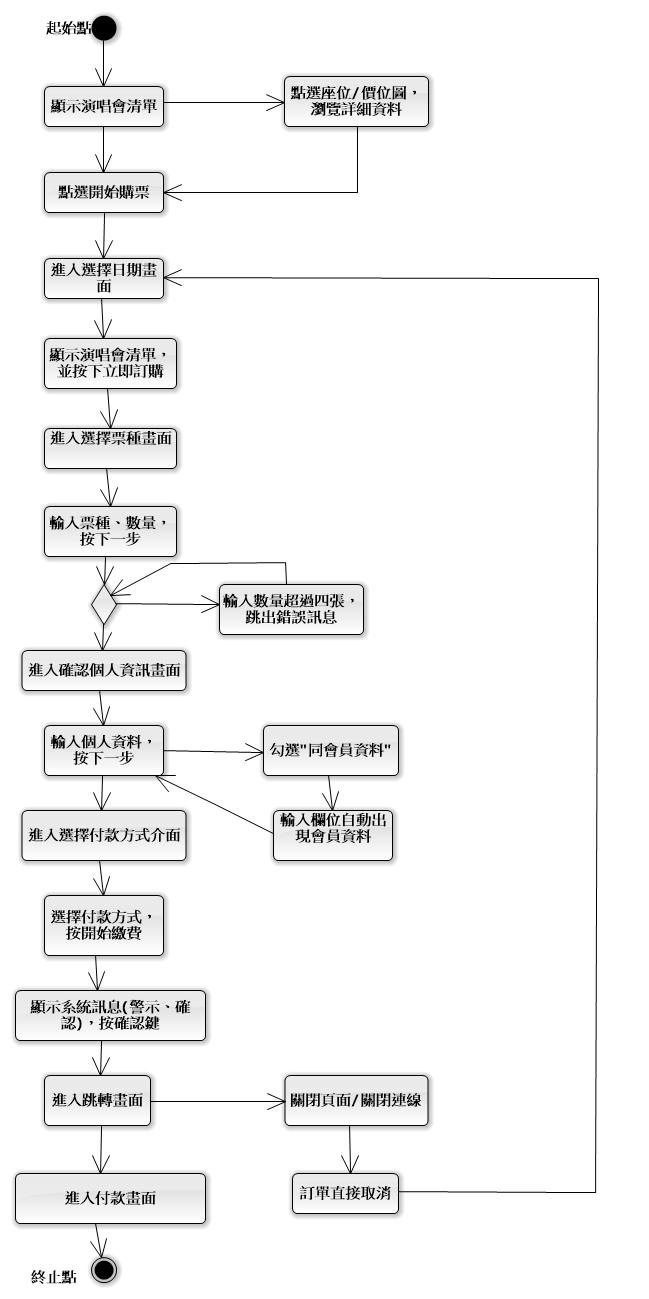


圖 2：使用案例3.1購票之活動圖

2.2.1.2 強韌圖

* 依據該使用案例之活動圖，可建立強韌圖以找出分析之類別，如下圖（圖 3）所示：

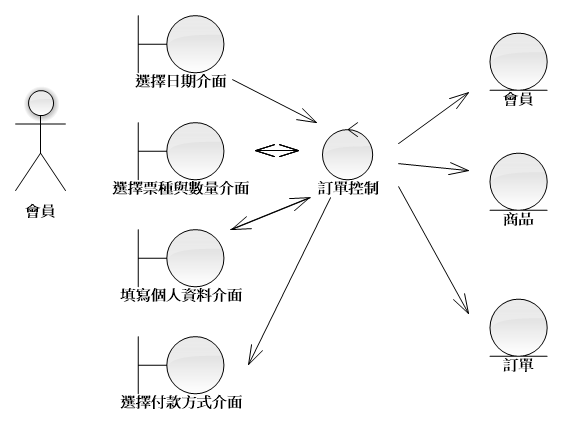


圖 3：使用案例3.1購票之強韌圖

### 2.3 使用案例7.0：供應商子系統

本章節中，將針對訂購商品模組進行系統動態分析圖之分析，結帳商品模組一共包含四個功能：新增供應商、新增票券資料、關閉售票、修改票券資料，以下並根據此些使用案例進行必要圖形之分析。

表 2：商業流程編號7.0供應商子系統模組

|  |  |
| --- | --- |
| 模組 | 功能名稱 |
| 7.0 模組：供應商子系統 | 7.1 新增供應商 |
| 7.2 新增票券資料 |
| 7.3 關閉售票 |
| 7.4 修改票券資料 |

### 2.3.1 使用案例7.1：新增票券資料

#### 2.3.1.1 活動圖

* 商業流程編號「7.1新增票券資料」之主要流程如下所示：
  + 1. 按下新增場次
  + 2. 進入新增場次頁面
  + 3. 輸入演唱會詳細資訊
  + 4. 點選送出
  + 5. 上架完成
* 透過以上主流程可將其轉換成活動圖，如下圖（圖 6）所示：

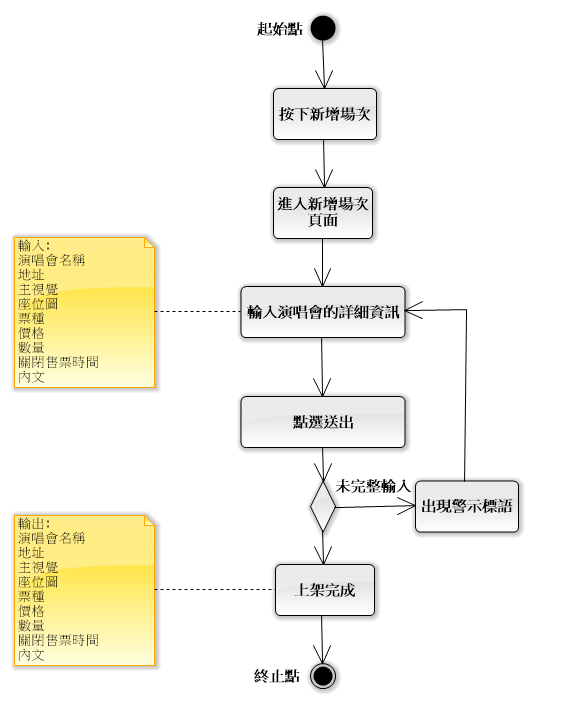


圖 4：使用案例7.1新增票券資料之活動圖

#### 2.3.1.2 強韌圖

* 依據該使用案例之活動圖，可建立強韌圖以找出分析之類別，如下圖（圖 7）所示：

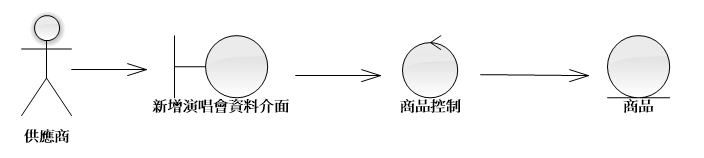


圖 5：使用案例7.1新增票券資料之強韌圖

### 第3章 資料庫設計

本專案之搶票系統提供使用者簡單的搶票、訂單管理與會員資料維護的服務，對於管理者來說，亦能以簡便方式進行商品維護和管理供應商等後台作業，希冀不僅能提供最新與詳盡之商品說明，更能提供快速且準確的搶票服務。

以下分析階段之資料庫設計採用實體關係圖（Entity-Relation Diagram）表示，並根據管理者與使用者之需求進行歸納與整理初步之系統條件。

以下詳述系統之資料庫需求，並將其整理成下圖（圖 10）之實體關係圖共計包含 6 個實體（Entity）、 5 個關係（Relationship）、 0 個複合性實體（Compound Entity）：

1. 一般訪客可以註冊成為會員且必須以電子郵件作為登入之帳號使用，同時系統會自動給予每位會員編號，註冊時也需要輸入姓名、密碼、DOB、身分證字號、電話號碼和地址。
2. 會員可以新增訂單，訂單具有獨一無二的編號。
3. 訂單資料庫會記錄票券的狀況，同時記錄每一筆訂單是誰買的、什麼時間，如何付款、訂單總費用以及是否完成付款。
4. 票券資料庫會記錄票券上記錄的資訊，包含獨一無二的編號、哪一場演場會、是否已經入場、還有訂單編號。
5. 管理者可以新增票券供應商，需要有供應商的電話、名字和帳號，而密碼會以暗碼儲存，並且新增、修改、刪除所有資料庫的資料。
6. 供應商可以新增演唱會資料，包含演唱會名字、地點、時間、售票截止日、簡介，還可以上傳照片、座位圖。平台會計算剩餘票券的數量，並在每次新增訂單的時候更新演唱會資料表中的剩餘票券。

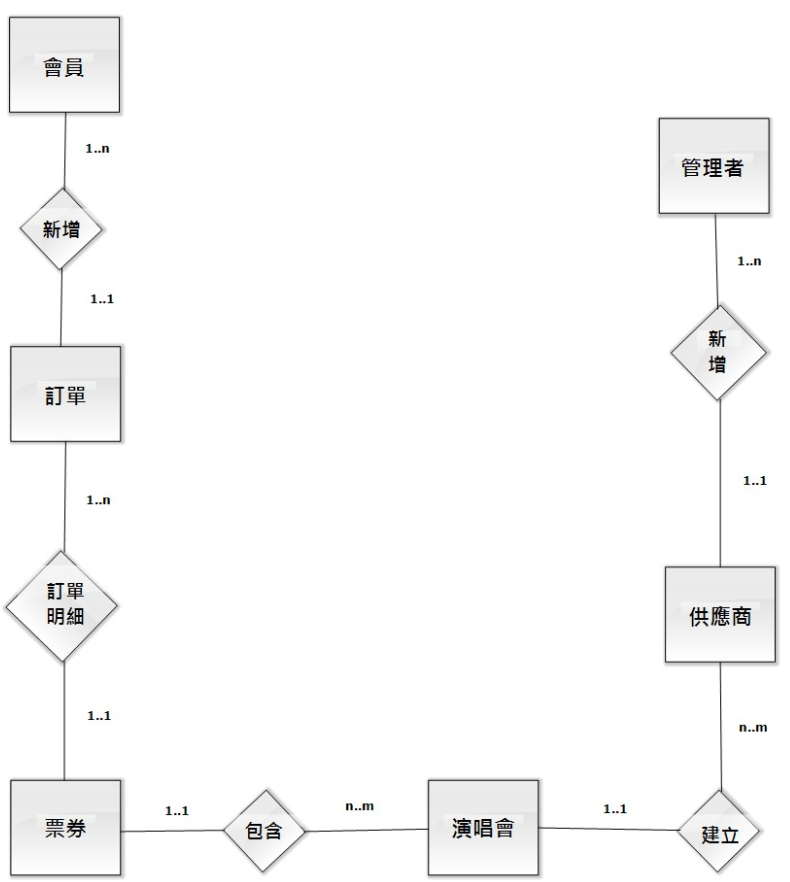


圖 6：實體關係圖

根據實體關係圖分析本專案所需之資料庫架構，以下將針對每張資料表進行描述，由於本範例僅實作後台管理者會員模組，因此資料表僅就會員與商品進行呈現，實際上仍需將所有資料表之分析呈現於此：

1. 會員資料表（members）

表 3：分析階段之會員資料表（members）資料結構

|  |  |  |
| --- | --- | --- |
| Key | 名稱 | 類型 |
| P.K. | idmember | Int |
|  | name | Varchar(255) |
|  | email | Varchar(255) |
|  | password | Varchar(255) |
|  | dateofbirth | DATE |
|  | idnumber | Varchar(255) |
|  | phonenumber | Varchar(255) |
|  | address | Varchar(255) |

1. 票券資料表（ticket）

表 4：分析階段之商品資料表（ticket）資料結構

|  |  |  |
| --- | --- | --- |
| Key | 名稱 | 類型 |
| P.K. | idticket | Int(11) |
|  | concertid | Int(11) |
|  | orderid | Int(11) |
|  | seatarea | Varchar(255) |
|  | seatid | Int(11) |
|  | isused | Tinyint(1) |
|  | email | Varchar(255) |
|  | phonenumber | Varchar(255) |
|  | name | Varchar(255) |

1. 訂單資料表 (order)

表 5：分析階段之訂單資料表(order)資料結構

|  |  |  |
| --- | --- | --- |
| Key | 名稱 | 類型 |
| P.K. | idorder | Int(11) |
|  | memberid | Int(11) |
|  | payment | Varchar(255) |
|  | paid | TINYINT(1) |
|  | ticketamount | Int(11) |
|  | createdtime | Datetime |
|  | totalprice | Int(45) |

1. 演唱會資料表 (concert)

表 6：分析階段之演唱會資料表(concert)資料結構

|  |  |  |
| --- | --- | --- |
| Key | 名稱 | 類型 |
| P.K. | idconcert | Int(11) |
|  | name | Varchar(45) |
|  | supplierid | Int(11) |
|  | location | Varchar(45) |
|  | picture | Varchar(5000) |
|  | seatpicture | Varchar(5000) |
|  | endsellingtime | Datetime |
|  | content | Varchar(10000) |
|  | ticketstatus | JSON |
|  | concertstarttime | Datetime |
|  | concertendtime | Datetime |
|  | session | Varchar |

1. 管理者資料表 (manager)

表 7：分析階段之管理者資料表(manager)資料結構

|  |  |  |
| --- | --- | --- |
| Key | 名稱 | 類型 |
| P.K. | idmanager | Int(11) |
|  | account | Varchar(255) |
|  | password | Varchar(255) |
|  | lastlogintime | Datetime |

1. 供應商資料表 (supplier)

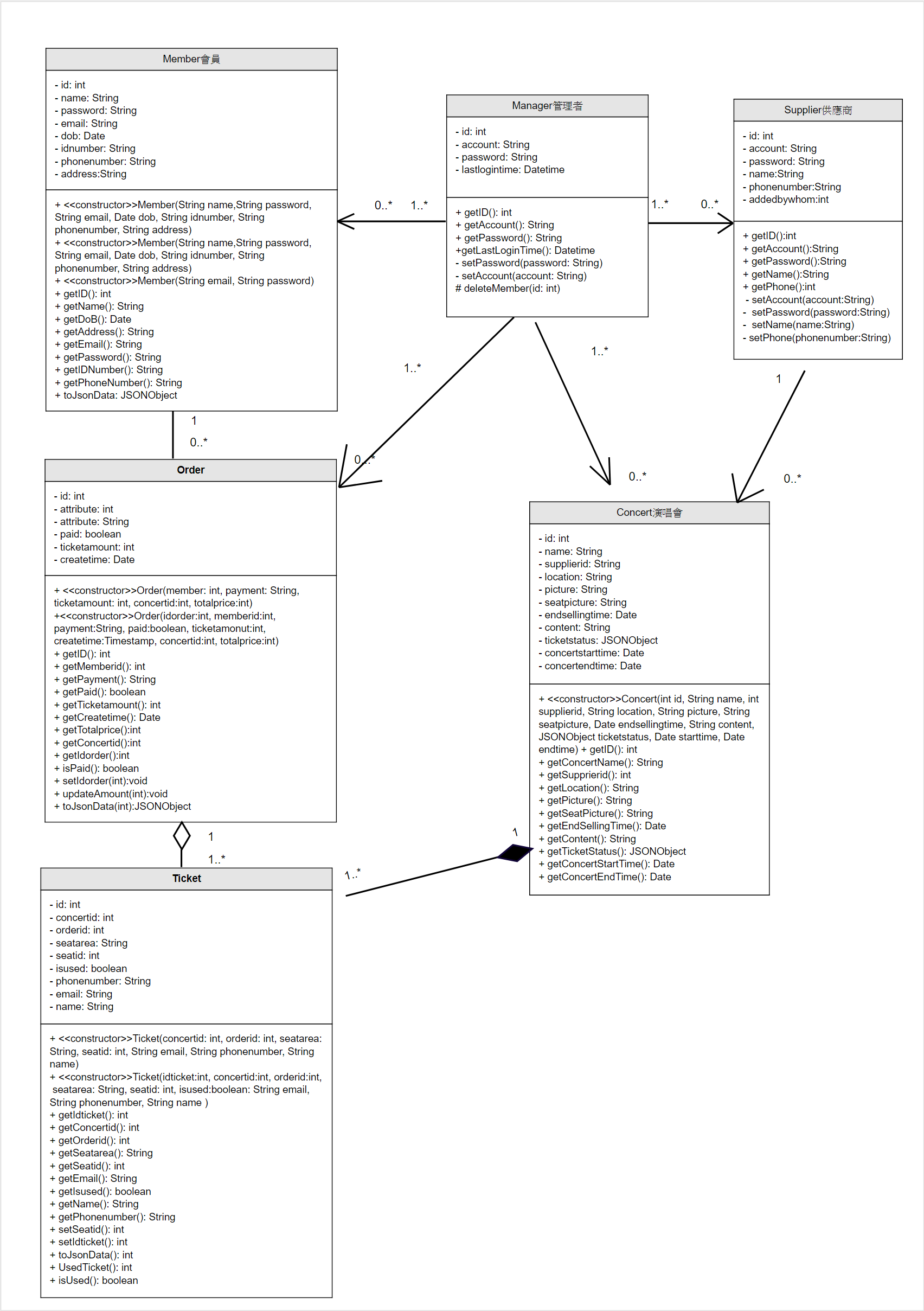
表 8：分析階段之供應商資料表(supplier)資料結構

|  |  |  |
| --- | --- | --- |
| Key | 名稱 | 類型 |
| P.K. | idsupplier | Int(11) |
|  | account | Varchar(255) |
|  | password | Varchar(255) |
|  | name | Varchar(255) |
|  | phonenumber | Varchar(255) |
|  | addedbywhom | Int(11) |

# 第4章 類別圖

分析階段之類別圖（class diagram）係依據第一份文件所述之使用案例找出並分析類別，同時也參照前章節（第3章 資料庫設計）以建立本專案之搶票系統分析模型之類別圖。

該階段之類別圖僅列出控制（controller）和實體之類別，其內部之詳細屬性與方法僅大略進行定義，詳細之設計細節與使用之參數屬性與方法於第三份文件—設計（design）詳細描述。



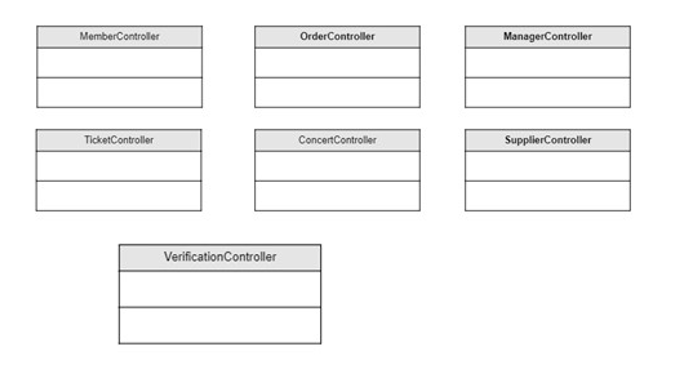


圖 7：分析階段之類別圖

# 第5章 系統開發環境

## 5.1 系統架構圖

本專案之整體架構如下圖（圖 12）所示，主要採用Java語言所撰寫之電子商務網站之應用程式，並預期採用Java平台技術之Servlet框架建構Web應用程式：

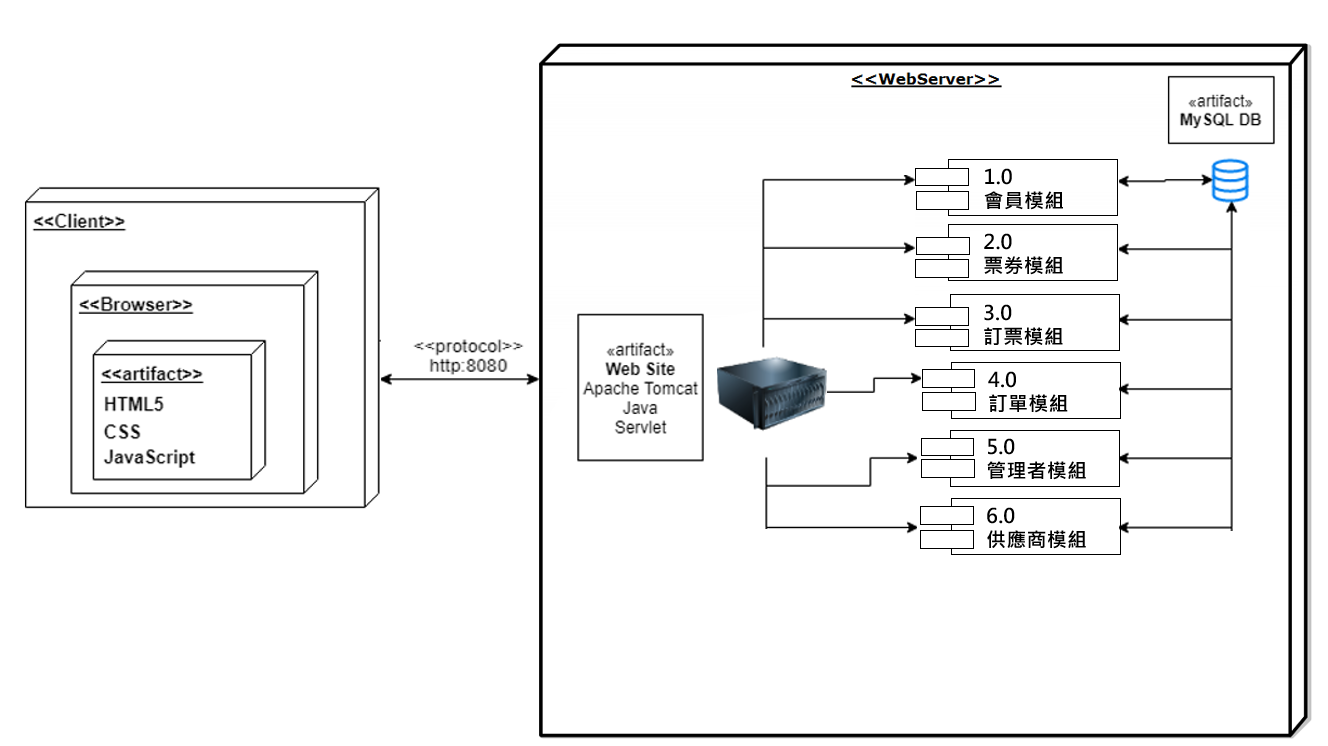


圖 8：分析階段之系統架構圖

1. 由於本專案之撰寫語言為Java因此需要採用Apache Tomcat作為伺服器軟體，預設http之埠號（port number）為8090。
2. 資料庫採用關聯性資料庫Oracle MySQL進行使用，開發階段使用community版本即可。
3. 本專案依照use case共計有六個模組，每個模組在下份文件中必須進行細部之設計與說明。
4. 使用者之裝置僅須透過瀏覽器即可瀏覽本電子商務網站。

## 5.2 MVC架構

本專案預期採用MVC架構，將應用程式的資料模型、使用介面和控制邏輯分割成Model、View和Controller三種元件，詳細如下圖（圖 13）所示。

其優點在於明確分割使用者介面和應用程式邏輯，將使用介面從商業邏輯中獨立出來，可以大幅降低建立大型應用程式時的複雜度，讓程式碼更有結構，容易測試、維護和重複使用。

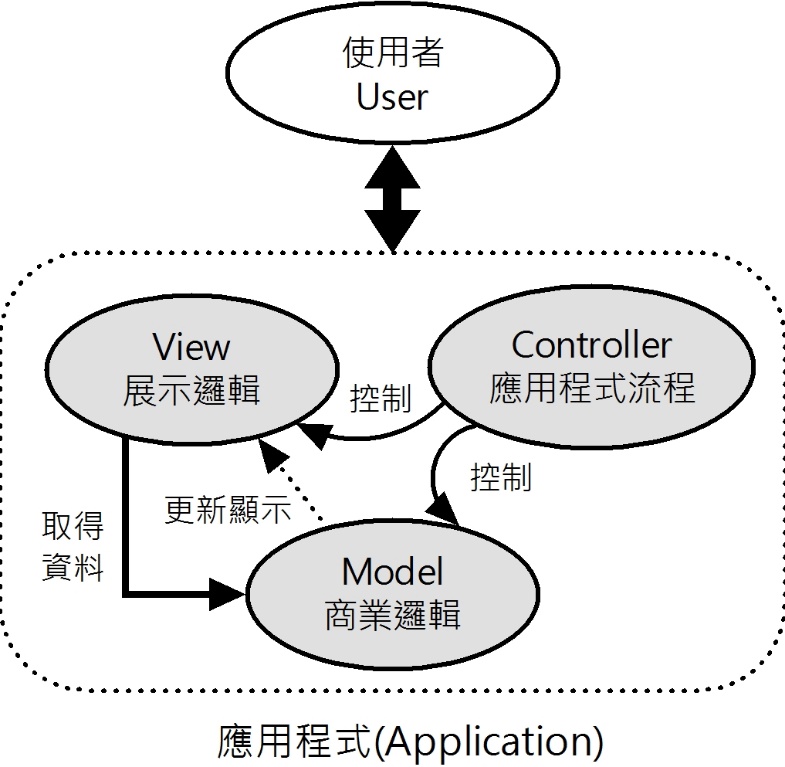


圖 9：MVC架構

1. Model：
   * 在MVC的Model元件是實作如何儲存應用程式的資料，包含資料和驗證規則。
   * 以Web應用程式來說，Model元件負責Web應用程式的資料存取和處理，即存取和處理儲存在資料庫、文字檔案或XML檔案的資料。
2. View：
   * 實作展示邏輯（Presentation Logic）的物件。
   * Web應用程式是建立使用者在瀏覽器看到的HTTP回應訊息，通常就是HTML網頁。
   * 使用Model物件儲存的資料來產生輸出結果，所以，View元件可以透過Model元件取得資料庫的資料，然後將資料庫的資料轉換成有用的資訊來呈現給使用者檢視
3. Controller：
   * 整個應用程式的中心，連接View和Model元件來協調和控制應用程式的執行。
   * Web應用程式的Controller元件是控制資料處理流程的控制器，負責接收使用者從瀏覽器送出的HTTP請求，依請求執行所需操作，即下達指令給Model取出所需的資料，然後送至View元件來產生顯示結果的HTML網頁。