

# 國立中央大學

## 資訊管理學系

### 111 (一) 系統分析與設計

#### 系統軟體分析規格書

#### 第九組

資管三 A 109403019 鄒翔宇

資管三 A 109403022 劉宗翰

企管四 A 107401511 何羽軒

資管三 B 109403530 陳俐吟

資管三 B 109403035 張羽慈

資管三 B 109403529 陳侑宣

經濟三 108409532 林喬楚

指導教授：葉羅堯 教授

中華民國 111 年 11 月 30 日

# 目錄

目錄.....	i
表目錄.....	iii
圖目錄.....	iv
版本修訂.....	v
第 1 章 簡介 .....	1
1.1 文件目的 .....	1
1.2 系統範圍 .....	1
1.3 參考文件 .....	1
1.4 文件架構 .....	2
第 2 章 系統動態分析圖 .....	3
2.1 使用者案例圖 .....	3
2.2 使用案例 4.0：課程結帳模組 .....	5
2.2.1 使用案例 4.1：課程結帳.....	5
2.2.1.1 活動圖 .....	5
2.2.1.2 強韌圖 .....	7
2.3 使用案例 10.0：挑戰課程模組 .....	7
2.3.1 使用案例 10.1：挑戰課程.....	8
2.3.1.1 活動圖 .....	8
2.3.1.2 強韌圖 .....	9
第 3 章 資料庫設計 .....	11
第 4 章 類別圖 .....	13
第 5 章 系統開發環境 .....	14

5.1	系統架構圖 .....	14
5.2	MVC 架構 .....	15

## 表目錄

表 1：商業流程編號 4.0 課程結帳模組.....	5
表 2：商業流程編號 10.0 挑戰課程模組.....	7

## 圖目錄

圖 1：使用者案例圖.....	4
圖 2：使用者案例 4.1 課程結帳之活動圖.....	6
圖 3：使用案例 4.1 課程結帳之強韌圖.....	7
圖 4：使用案例 10.1 挑戰課程之活動圖.....	9
圖 5：使用案例 10.1 挑戰課程之強韌圖.....	10
圖 6：實體關係圖.....	11
圖 7：分析階段之會員資料表（member）資料結構.....	12
圖 8：分析階段之課程資料表（course）資料結構.....	12
圖 9：分析階段之實體類別圖.....	13
圖 10：分析階段之系統架構圖.....	14
圖 11：MVC 架構.....	15

## 版本修訂

版本	修訂者	修訂簡述	日期
v0.1.0	鄒翔宇	Draft	2022/11/27
v1.0.0	何羽軒	分析文件第一版	2022/11/30
v1.1.0	鄒翔宇	<ul style="list-style-type: none"><li>✓ 插入CARRYU SRS文件連結</li><li>✓ 將所有圖片轉換成svg向量圖</li><li>✓ 5.1 添加 Controller</li></ul>	2022/11/30

# 第 1 章 簡介

軟體分析規格書 (software analysis description, SAD) 為依據軟體產品、專案主要使用者之需求規格文件 (software requirements specification, SRS)，主要用於描述軟體架構與系統範圍之邏輯。藉由本文件得以達成分析軟體系統架構之目的，並作為軟體設計階段之依據。

本專案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)、活動圖 (Activity Diagram)、強韌圖 (Robustness Diagram)、實體關係圖 (Entity-Relation Diagram)、及類別圖 (Class Diagram)。

## 1.1 文件目的

本文件目的為提供軟體系統開發人員分析規範與藍圖，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，並得以此為依據，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

本系統範圍用於線上課程平台，其中主要包含會員管理、課程資訊、購物車、課程結帳、購買紀錄、退貨管理、課程管理、管理員管理、課程推薦、挑戰課程與學習等十一個模組，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線上課程平台所需的管理流程。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

[CARRYU\\_v1.3.2—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提供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 使用者案例圖

依據第一份文件—系統軟體需求規格書（Software Requirement Specification），本線上課程平台系統預計共有 6 位動作者與 49 個使用案例，並依照不同之模組區分成不同子系統，共計十一個子系統，依序為①會員子系統、②課程資訊子系統、③購物車子系統、④課程結帳子系統、⑤購買紀錄子系統、⑥退貨管理子系統、⑦課程管理者子系統、⑧管理員管理子系統、⑨課程推薦子系統、⑩挑戰課程子系統、⑪學習子系統，如下圖（圖 1）所示。

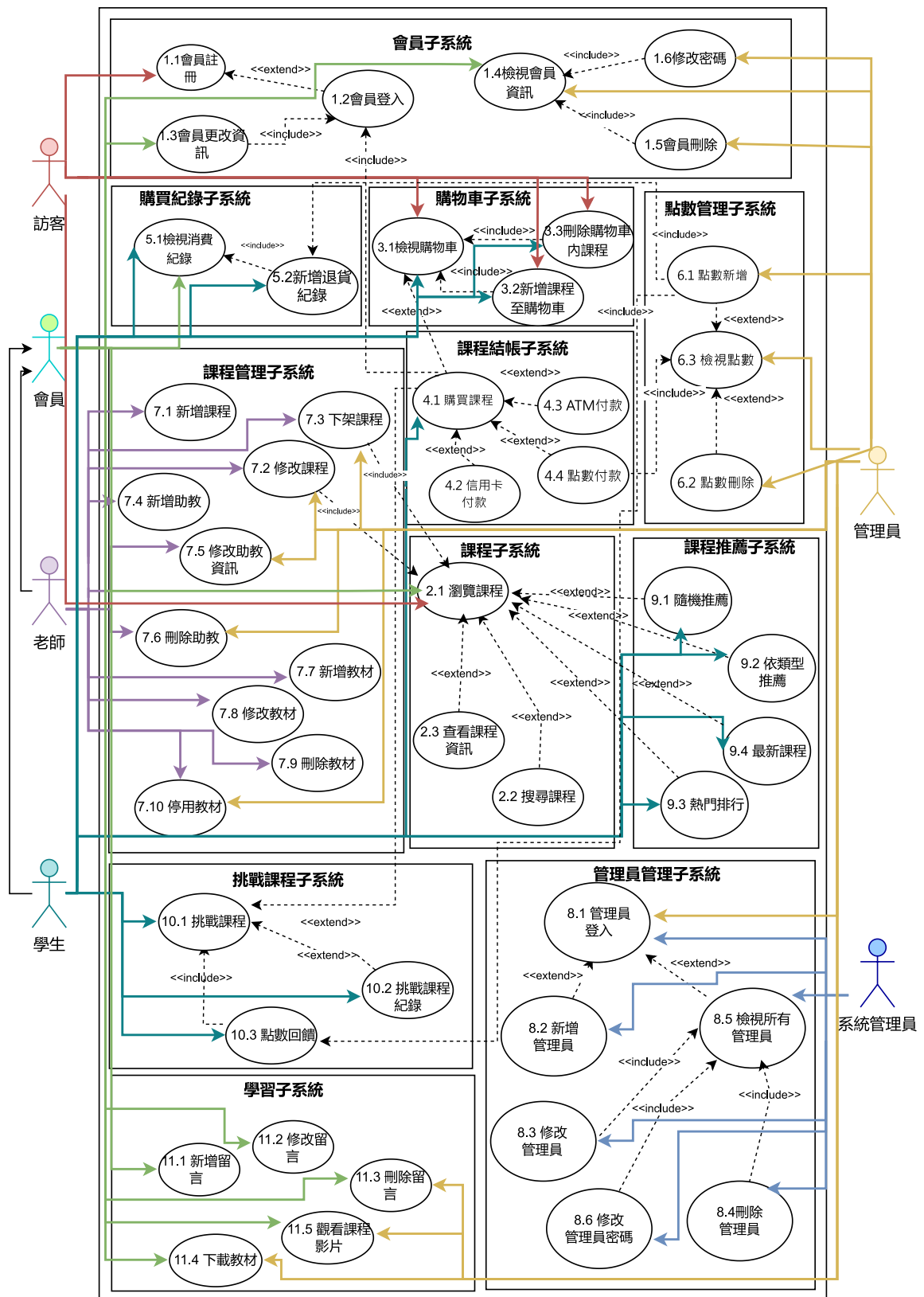


圖 1：使用者案例圖

根據上圖所分析之使用案例規格，將使用案例逐一轉換成活動圖與強韌圖，進行詳細闡述而後分析。依據前揭所述之活動圖、強韌圖產生所需之通訊圖與狀態機圖，進行統整並依照本系統之架構劃分而繪製出分析階段之循序圖。

## 2.2 使用案例 4.0：課程結帳模組

本章節中，將針對課程結帳模組進行系統動態分析圖之分析。課程結帳模組一共包含四個功能：①課程結帳、②信用卡付款、③ATM 付款、④點數付款資訊，其中商業流程編號 4.2 信用卡付款、商業流程編號 4.3 ATM 付款應當在商業流程編號 4.1 課程結帳後選擇其中一個使用案例進行，商業流程編號 4.4 點數付款則可以與商業流程編號 4.2 及商業流程編號 4.3 合併或單獨使用，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 4.0 課程結帳模組

模組	功能名稱
4.0 模組：課程結帳	4.1 課程結帳
	4.2 信用卡付款
	4.3 ATM付款
	4.4 點數付款

### 2.2.1 使用案例 4.1：課程結帳

#### 2.2.1.1 活動圖

- 商業流程編號「4.1 課程結帳」之主要流程如下所示：

1. include：：檢視購物車。
2. 點選「結帳」。
3. 系統查詢客戶資料。
4. 系統顯示訂單確認介面。

5. 輸入付款資料。

6. 系統顯示訂單編號和成功完成結帳的訊息。

- 替代流程：

5a. 若是扣款失敗或是繳交時間過期，則顯示結帳失敗的訊息。

- 透過以上主流程可將其轉換成活動圖，如下圖（圖 2）所示：

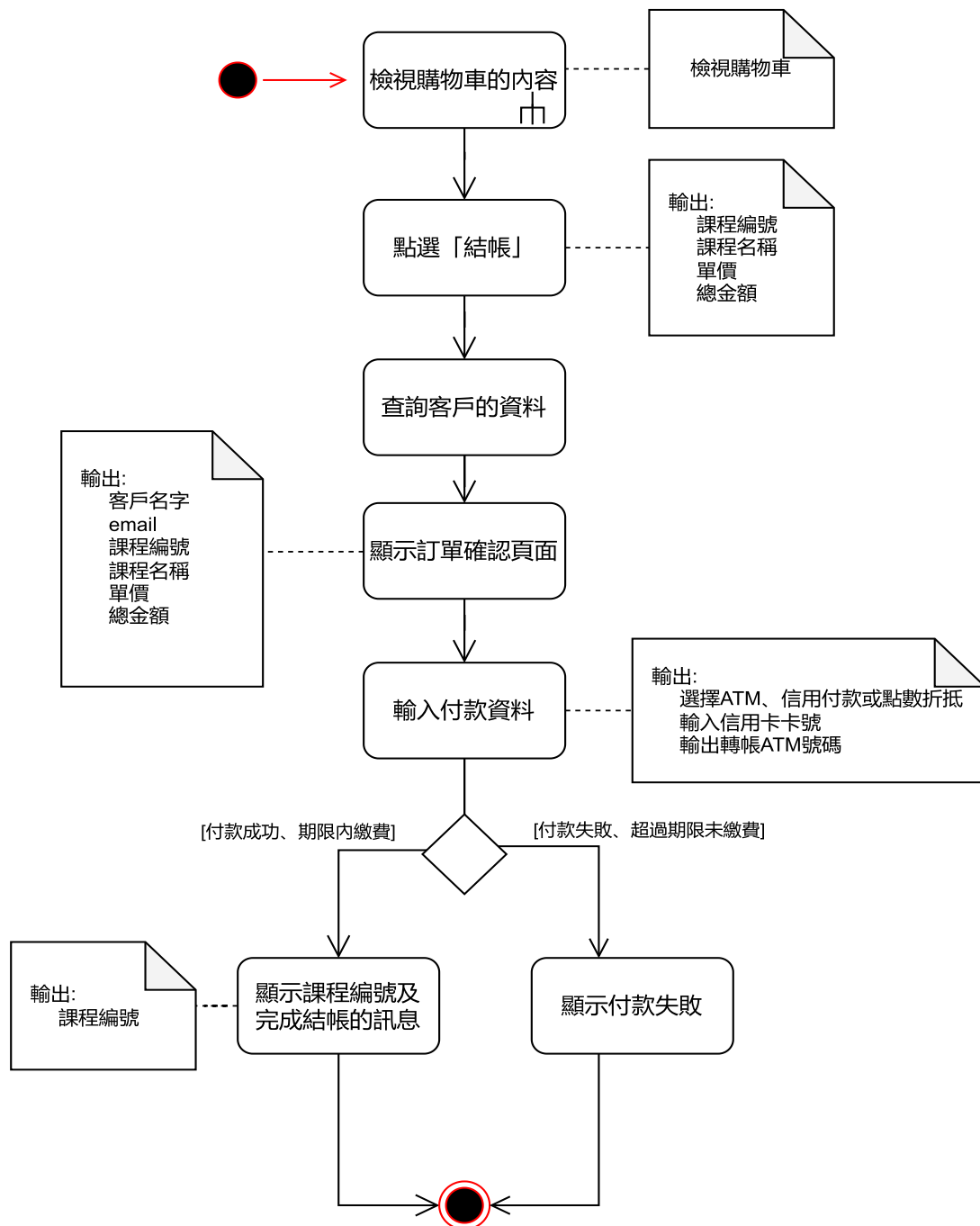


圖 2：使用者案例 4.1 課程結帳之活動圖

## 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：

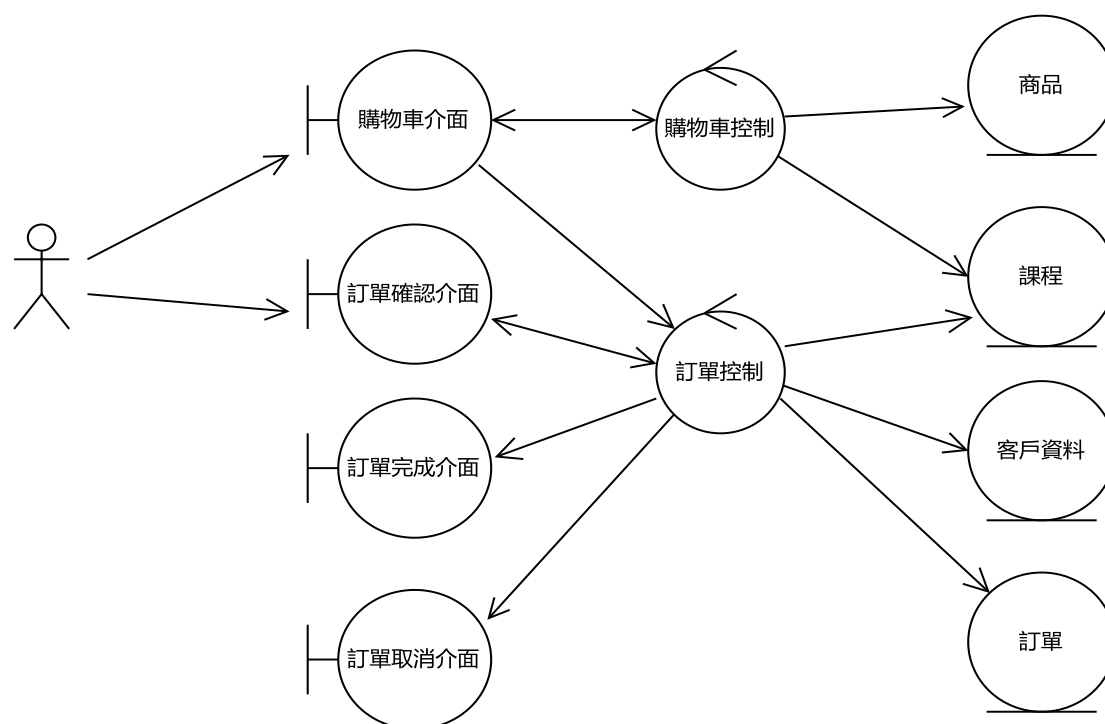


圖 3：使用案例 4.1 課程結帳之強韌圖

## 2.3 使用案例 10.0：挑戰課程模組

本章節中，將針對挑戰可成模組進行系統動態分析圖之分析，挑戰課程模組一共包含三個功能：① 挑戰課程、② 挑戰課程紀錄、③ 點數回饋，其中商業流程編號10.1挑戰課程商業流程編號4.1課程結帳的延伸點，商業流程編號10.3點數回饋應於商業流程編號10.1挑戰課程中得到挑戰成功紀錄同時才執行，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 10.0 挑戰課程模組

模組	功能名稱
10.0模組：挑戰課程	10.1 挑戰課程
	10.2 挑戰課程紀錄
	10.3點數回饋

## 2.3.1 使用案例 10.1：挑戰課程

### 2.3.1.1 活動圖

- 商業流程編號「10.1 挑戰課程」之主要流程如下所示：
  1. 學生購買五堂課程。
  2. 選擇要照類別選課程或是隨機選課程。
  3. 系統分發課程。
  4. 顯示課程資訊。
  5. 學生決定是否挑戰。
  6. 挑戰後結帳此課程。
  7. 新增挑戰紀錄。
  8. 挑戰成功後新增挑戰成功紀錄及新增點數。
  9. 顯示挑戰成功訊息及點數餘額。
- 替代流程：
  - 5a. 選擇不挑戰顯示不挑戰訊息。
  - 7a. 未達成挑戰要求新增挑戰失敗紀錄並顯示失敗訊息。
- 透過以上主流程可將其轉換成活動圖，如下圖（圖 4）所示：

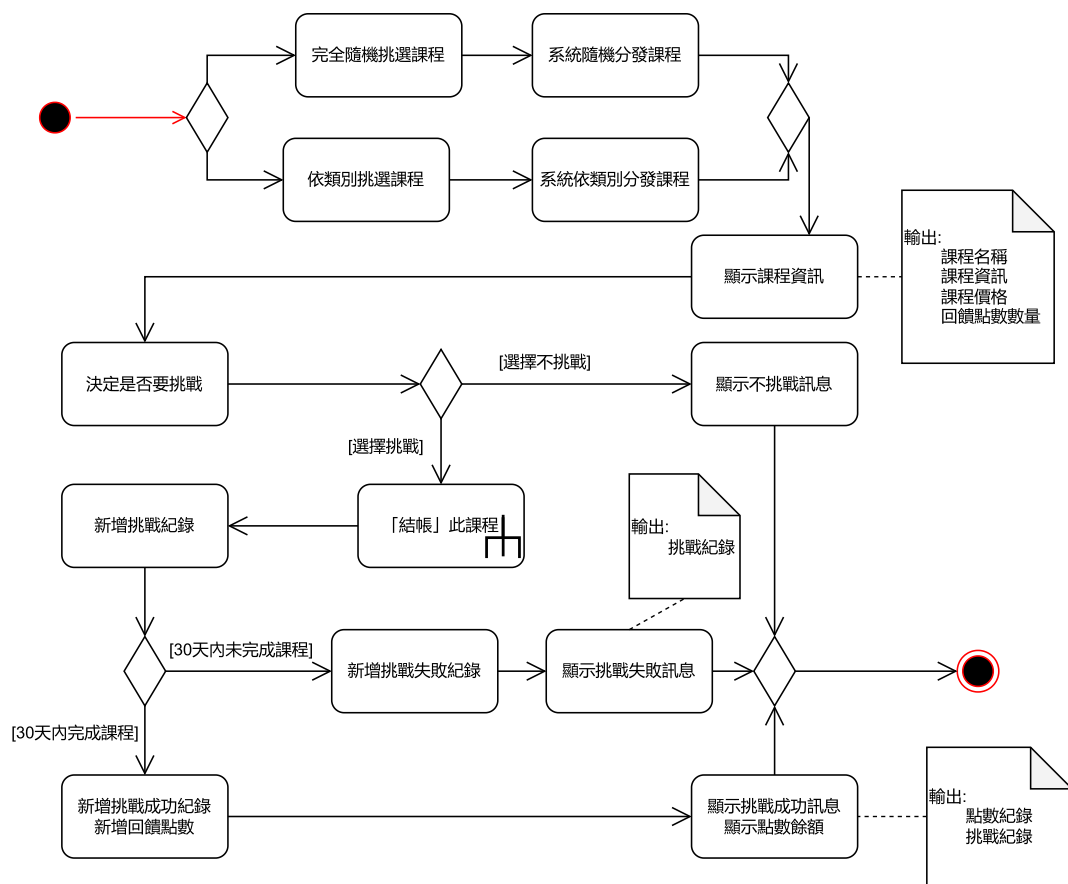


圖 4：使用案例 10.1 挑戰課程之活動圖

### 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 5）所示：

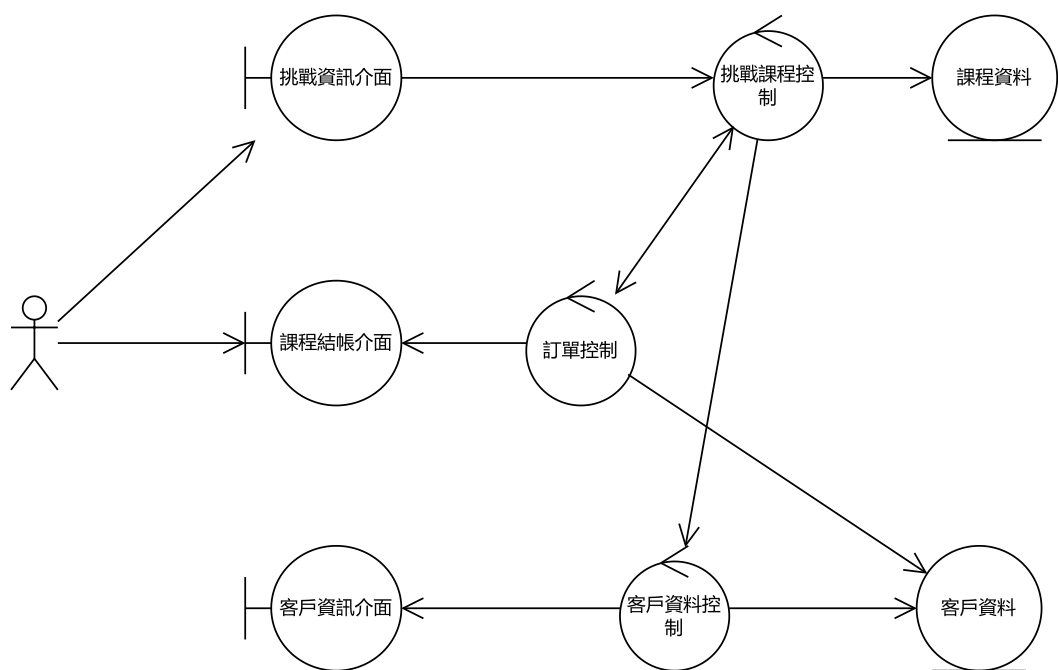


圖 5：使用案例 10.1 挑戰課程之強韌圖



## 第 3 章 資料庫設計

本專案之線上課程平台系統提供使用者購買想學習的線上課程、購物車管理與會員資料維護之服務。對於資料庫管理員來說，能夠直接於 MongoDB Atlas 中管理資料以及新增、刪除其他管理員與修改其權限。希望不僅能提供最新與高品質之課程內容，更能提供良好的學習體驗。

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 6）之實體關係圖共計包含 5 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。
2. 一般訪客與會員皆可瀏覽課程商品將並將其加入購物車，但伺服器不儲存訪客資料，而是存於使用者本地端。
3. 會員可以對購物車商品進行結帳，以完成訂單，同時可以查閱訂單詳情。
4. 管理者可以管理商品之異動與會員之資料，並且進行維護作業，同時要記錄帳號建立時間與異動之動作。

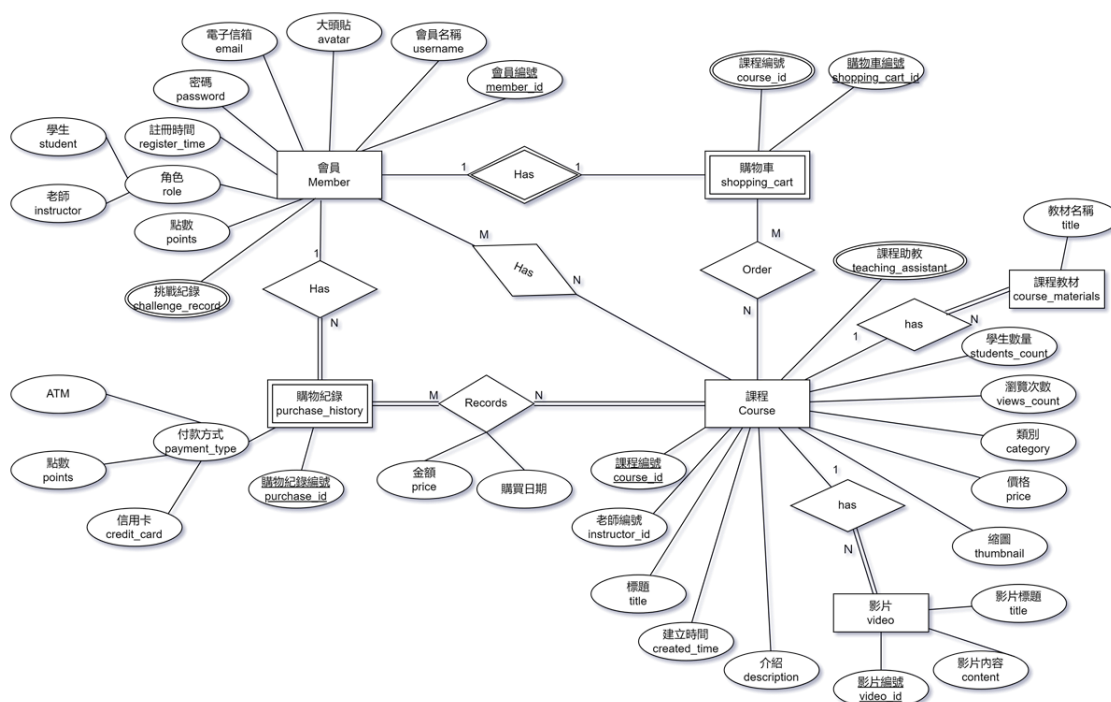


圖 6：實體關係圖

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：

### 1. 會員資料表 (member)

<b>Member(會員資料表)</b>		
Pk	member_id	String
	username	String
	avatar	BLOB
	email	String
	password	String
	register_time	Date
	role	String
	shopping_cart	Array
	course_id	String
	purchase_history	Array
	purchase_record	
	course_id	String
	price	int
	payment_type	String
	status	String
	points	int
	challenge_history	Array
	challenge_record	
	course_id	String
	challenge_date	Date

圖 7：分析階段之會員資料表 (member) 資料結構

### 2. 課程資料表 (course)

<b>Course(課程資料表)</b>		
PK	course_id	String
	instructor_id	String
	title	String
	description	String
	thumbnail	BLOB
	price	int
	category	String
	views_count	int
	students_count	int
	students	Array
	student_id	String
	videos	Array
	video	
	serial_number	int
	content	BLOB
	title	String
	course_materials	Array
	material	
	serial_number	int
	content	BLOB
	title	String
	teaching_assistant	Array
	member_id	String
	created_time	Date

圖 8：分析階段之課程資料表 (course) 資料結構

## 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，並參照前章節（第 3 章 資料庫設計）以建立本專案之電子商務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。

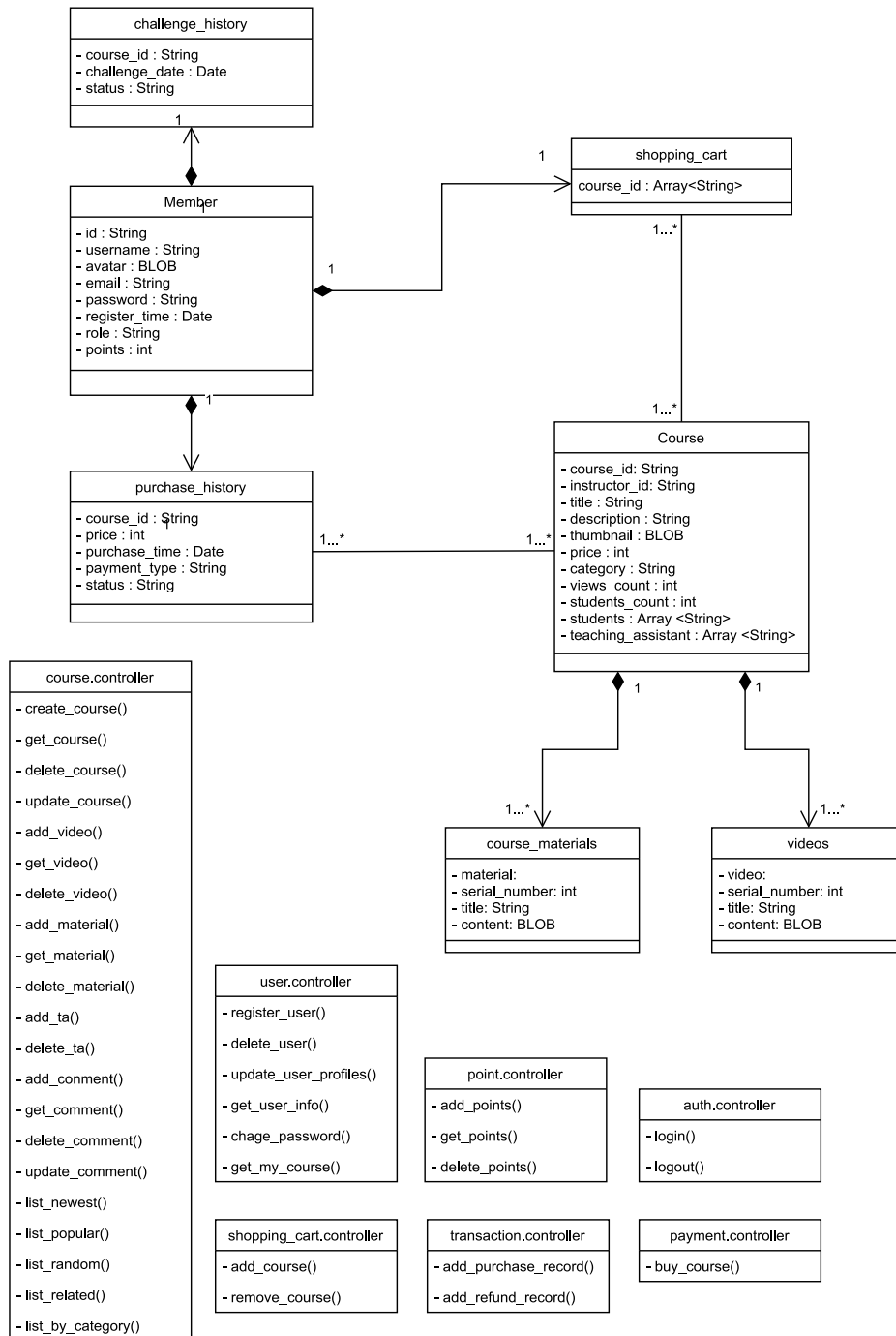


圖 9：分析階段之實體類別圖

## 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 JavaScript 語言來撰寫線上課程影音平台之應用程式，並預期採用 JavaScript 平台技術之 Node.js 框架建構 Web 應用程式：

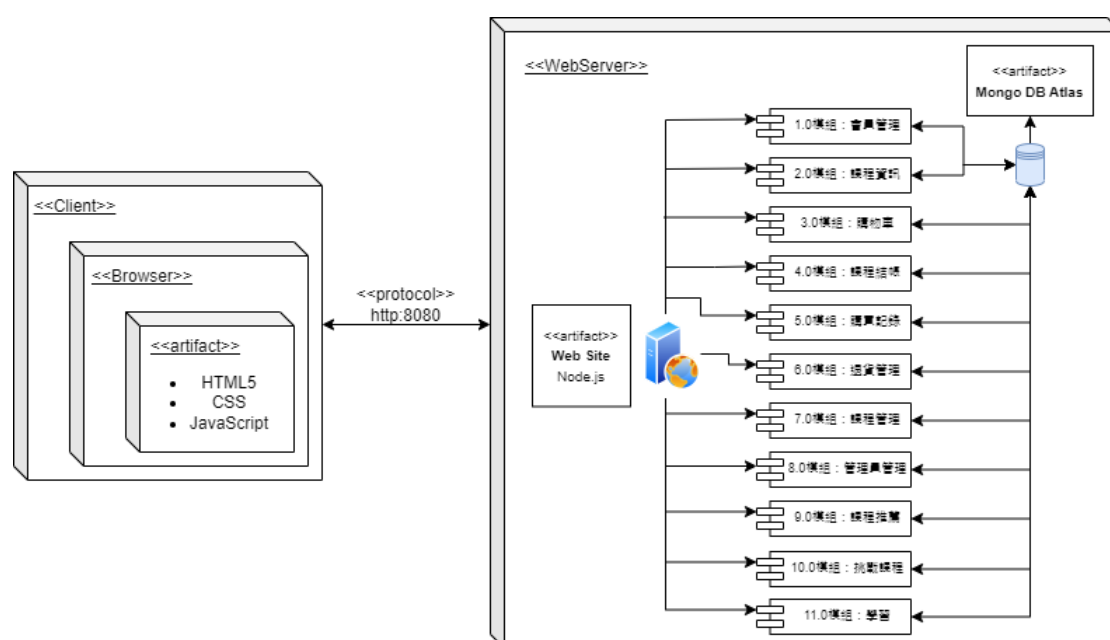


圖 10：分析階段之系統架構圖

1. 由於本專案之撰寫語言為 JavaScript 因此需要採用 Node.js 作為網頁伺服器軟體，預設 http 之埠號（port number）為 8080。
2. 資料庫採用非關聯性資料庫 MongoDB，並使用其雲端資料庫服務 MongoDB Atlas 進行。開發階段使用 Shared 免費資料庫，並將雲服務供應商之伺服器位置設在離我們主要客戶群較近的台灣。
3. 本專案依照use case共計有十一個模組，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課程影音平台。

## 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。

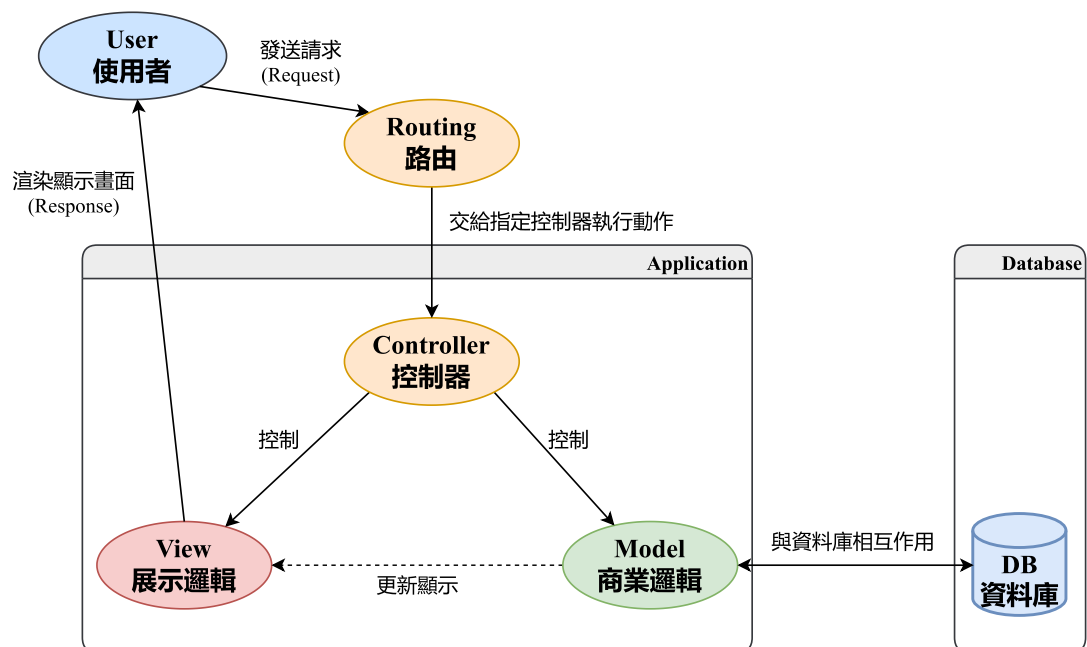


圖 11：MVC 架構

### 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。

### 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。