

# 系統需求規格書

## 電影與影評管理系統

第17組

41171102H 許家寧

41171113H 王方瑩

41171116H 董承翰

41171215H 周俊辰

---

### 目錄

1. 第一章 簡介
  - 1.1 需求概述
  - 1.2 系統範圍
2. 第二章 系統描述
  - 2.1 系統架構圖
  - 2.2 功能性需求
    - 2.2.1 電影資料管理
    - 2.2.2 影評管理
    - 2.2.3 使用者與權限管理
    - 2.2.4 搜尋與檢索功能
    - 2.2.5 評分統計與報表
  - 2.3 ER-diagram設計圖
    - 2.3.1 系統資料分析
    - 2.3.2 系統需要儲存的資料
    - 2.3.3 ER-diagram所表達的概念與限制
  - 2.4 介面需求
    - 2.4.1 前台使用者介面
    - 2.4.2 後台管理介面
  - 2.5 操作界面與情境

### 3. 第三章 系統規格表

---

# 第一章 簡介

## 1.1 需求概述

當前多數影評管理系統功能已趨成熟，主要聚焦於電影資料的儲存、影評撰寫與評分統計，例如：提供電影基本資訊維護、影評撰寫與查詢、使用者註冊與登入、以及基本的熱門度統計與報表功能等。然而，這類系統多偏重文字與數據分析，缺乏創新互動與視覺延伸應用，使用者參與度與體驗仍有提升空間。

本系統除具備上述基礎功能外，特別導入創新設計，包括：

- 劇照管理：每部電影可搭配多張劇照，作為內容展示與風格輔助，並支援後台上傳與管理，強化視覺體驗，並作為評論與創作的素材。
- 穿搭推薦：根據電影中的角色造型與劇照風格，提供穿搭建議與品牌參考，延伸觀影後的生活應用與娛樂價值，讓影迷從電影延伸至現實穿搭參考。
- 串流平台資訊推薦：系統可顯示與該電影相關的串流平台資訊(EX.短影音、訪談節目等)，以提升資訊完整性與實用性。

本系統在傳統電影與影評管理基礎上，導入多項創新設計，旨在打造一套兼具延伸性與沉浸感的互動平台，突破僅限於資料查詢與影評撰寫的侷限。整體系統融合觀影、評論、時尚與串流播放等多元功能，滿足影迷在資訊探索、內容互動與生活應用上的多重需求，讓每一次觀影體驗都更加立體與豐富。

## 1.2 系統範圍

本系統主要涵蓋以下使用者角色與功能範圍：

- 使用者
  - 註冊登入
  - 瀏覽電影列表及詳細資訊。
  - 撰寫、編輯、刪除自己的影評與評分。
  - 利用搜尋功能查詢電影資料與影評。
- 管理員
  - 後台登入與權限控管。
  - 電影資料的新增、修改與刪除。

- 管理所有用戶影評。

## 第二章 系統描述

### 2.1 系統架構圖

本系統採用典型的B/S(Browser/Server)架構, 主要組成部分包括:

- 前端用戶介面:使用HTML、CSS、JavaScript等技術設計, 提供電影與影評瀏覽、撰寫及搜尋功能。
- 後端伺服器:利用Python伺服器端語言處理業務邏輯。
- 資料庫:採用MySQL儲存電影資料、影評、使用者資訊、電影劇照、串流影音平台等。



### 2.2 功能性需求

#### 2.2.1 電影資料管理

- 電影資料新增:管理員可透過後台新增電影資訊, 內容包括片名、導演、演員、類型、上映日期、劇情簡介、海報圖片、劇照管理、穿搭推薦、串流影音平台等。
- 電影資料修改與刪除:允許管理員對既有電影資料進行修改或刪除操作。

#### 2.2.2 影評管理

- 影評撰寫與編輯: 註冊用戶可針對電影撰寫影評, 並可修改或刪除自己的評論。
- 評分功能: 用戶可對電影進行打分, 系統自動計算平均評分。

### 2.2.3 使用者與權限管理

- 會員註冊與登入: 提供新用戶註冊與現有用戶登入功能, 並實現密碼加密及驗證。
- 權限控管: 區分使用者與管理員, 並根據角色賦予不同操作權限。
- 個人資料管理: 允許使用者修改個人資訊及管理歷史評論紀錄。

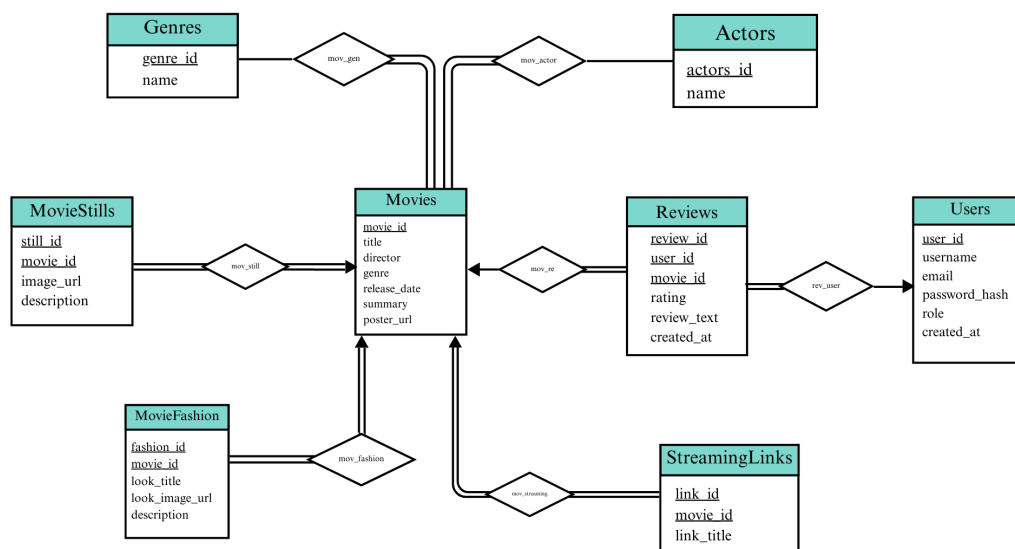
### 2.2.4 搜尋與檢索功能

- 電影搜尋: 根據片名、類型、導演、演員等條件進行電影資料的查詢與篩選。
- 影評查詢: 支持按電影、用戶、評分等條件檢索影評記錄。

### 2.2.5 評分統計與報表

- 評分統計: 系統自動彙整每部電影的評分數據, 計算平均分與評價分佈。
- 報表生成: 管理員可產生各類統計報表, 用於分析電影受歡迎程度與用戶評論趨勢。

## 2.3 ER-diagram設計圖



### 2.3.1 系統資料分析(Data Analysis)

這個系統的核心目的是管理電影、影評與延伸應用資料, 包括劇照、穿搭與串流平台資訊。資料關係清楚劃分為多個Entity與Relationship, 以支援下列用途:

核心資料實體:

1. Movies: 電影基本資料的主體
2. Users: 使用者帳號資料(含身分與註冊)
3. Reviews: 使用者撰寫的影評
4. Genres: 電影類型
5. Actors: 演員資料
6. MovieStills: 電影劇照
7. MovieFashion: 電影穿搭建議
8. StreamingLinks: 串流相關資訊(預告、花絮等)

關聯關係重點:

- 多對多:
  - 一部電影對多個類型(Genres) → mov\_gen
  - 一部電影包含多位演員(Actors) → mov\_actor
- 一對多:
  - 一部電影被多個使用者評論(Reviews) → mov\_re
  - 一位使用者可寫多篇評論 → rev\_user
  - 一部電影對應多張劇照(MovieStills) → mov\_still
  - 一部電影擁有多筆穿搭建議(MovieFashion) → mov\_fashion
  - 一部電影擁有多筆串流資訊(StreamingLinks) → mov\_streaming

### 2.3.2 系統需要儲存的資料

1. Users (使用者)

欄位名稱	型別	說明
user_id	INT (PK)	使用者唯一識別碼
username	VARCHAR(50)	使用者名稱
email	VARCHAR(100)	電子郵件
password_hash	VARCHAR(255)	密碼
role	ENUM	使用者角色 (User/Admin)
created_at	DATETIME	註冊時間

---

2. Movies (電影)

欄位名稱	型別	說明
movie_id	INT (PK)	電影唯一識別碼
title	VARCHAR(100)	電影標題
director	VARCHAR(100)	導演名稱
genre	VARCHAR(50)	類型 (或搭配 MovieGenres)
release_date	DATE	上映日期
summary	TEXT	劇情簡介
poster_url	VARCHAR(255)	海報圖片連結

---

3. Reviews (影評)

欄位名稱	型別	說明
review_id	INT (PK)	影評唯一識別碼
user_id	INT (FK)	撰寫者 (對應 Users)
movie_id	INT (FK)	評論的電影 (對應 Movies)
rating	FLOAT	評分 (例如 1~5 分)
review_text	TEXT	評論內容
created_at	DATETIME	評論時間

---

#### 4. Genres (電影類型)

欄位名稱	型別	說明
genre_id	INT (PK)	類型 ID
name	VARCHAR(50)	類型名稱

---

#### 5. Actors (演員)

欄位名稱	型別	說明
actor_id	INT (PK)	演員 ID
name	VARCHAR(100)	演員名稱

---

#### 6. MovieStills (電影劇照)

欄位名稱	型別	說明
still_id	INT (PK)	劇照唯一識別碼



movie_id	INT (FK→ Movies)	電影唯一識別碼
image_url	VARCHAR(255)	劇照連結
description	TEXT	描述這張劇照的場景

---

## 7. MovieFashion (私服穿搭推薦)

欄位名稱	型別	說明
fashion_id	INT (PK)	私服穿搭推薦唯一識別碼
movie_id	INT (FK→ Movies)	電影唯一識別碼
look_title	VARCHAR(50)	名稱: 如「主角的黑色風衣造型」
look_image_url	VARCHAR(255)	推薦穿搭配照
description	TEXT	文字說明 如: 風格分析、推薦品牌等

---

## ◆ 8. StreamingLinks (電影相關串流資源)

欄位名稱	型別	說明
link_id	INT (PK)	串流連結 ID
movie_id	INT (FK)	對應 Movies.movie_id
link_title	VARCHAR(50)	串流資源標題 (例: 主演訪談、預告片)
video_url	VARCHAR(255)	串流嵌入網址

## ◆ 9. mov\_actor

欄位名稱	型別	說明
movie_id	INT (FK)	對應 Movies.movie_id
actor_id	INT (FK)	對應 Actors.actor_id

#### ◆ 10. mov\_gen

欄位名稱	型別	說明
genre_id	INT (FK)	對應 Genres.genre_id
movie_id	INT (FK)	對應 Movies.movie_id

#### ◆ 11. mov\_still

欄位名稱	型別	說明
movie_id	INT (FK)	對應 Movies.movie_id
still_id	INT (FK)	對應 MovieStills.still_id

#### ◆ 12. mov\_re

欄位名稱	型別	說明
movie_id	INT (FK)	對應 Movies.movie_id
review_id	INT (FK)	對應 Reviews.review_id
user_id	INT (FK)	對應 Reviews.user_id

#### ◆ 13. rev\_user

欄位名稱	型別	說明
review_id	INT (FK)	對應 Reviews.review_id
user_id	INT (FK)	對應 Users.user_id

#### ◆ 14. mov\_fashion

欄位名稱	型別	說明
mov_id	INT (FK)	對應 Movies.movie_id
fashion_id	INT (FK)	對應 MovieFashion.fashion_id

#### ◆ 15. mov\_streaming

欄位名稱	型別	說明
mov_id	INT (FK)	對應 Movies.movie_id
link_id	INT (FK)	對應 StreamingLink.link_id

### 2.3.3 ER-diagram所表達的概念與限制

表達的概念：

1. 關係型資料庫模型(**Relational Model**)
  - 每個實體對應一個資料表, 每筆資料皆有唯一主鍵
2. 資料正規化
  - 將演員、類型等抽成獨立表, 避免重複冗餘
3. 多對多關係 → 關聯表處理
  - 使用關聯菱形(如 mov\_gen、mov\_actor)來表示中介關聯表
4. 資料擴充性與延伸性
  - 系統支援視覺化內容與生活應用(如: 劇照、穿搭、影片連結)

表示的限制條件：

1. 唯一性限制(**Primary Key**)
  - 每個實體均設有主鍵欄位(如 movie\_id, user\_id)
2. 參照完整性(**Foreign Key**)

- 影評必須對應一部已存在的電影與一個使用者
- 劇照、穿搭、串流等皆依賴 movie\_id
- 3. 不允許獨立資料
  - 沒有對應電影的穿搭、劇照、評論不得獨立存在(透過 FK 實作)
- 4. 多對多不能直接存於欄位中
  - 不將類型或演員直接放在 Movies 表中, 而是使用中介關聯表分離儲存

## 2.4 介面需求

### 2.4.1 前台使用者介面

#### 1. 系統概述

1. 使用技術: HTML + PHP + MySQL
2. 功能類型: 前後端整合式動態網頁系統
3. 資料儲存: 所有結構化資料存入 MySQL; 圖片與影片以路徑(URL)儲存於資料表中, 避免資料庫資料量過大

#### 2. 首頁(index.php)

功能說明:

顯示所有電影的清單與基本資訊:

- 電影名稱(可點擊進入詳情頁)
- 導演
- 上映日期
- 平均評分(從使用者評論中計算)

電影與影評管理系統主頁

#### 3. 註冊登入介面(login.php/login\_process.php/logout.php)

功能說明：

- 註冊 / 登入
  - 使用者可註冊帳號，登入後才能使用互動功能
  - 使用者角色分為：一般使用者與 Admin
- 權限分級
  - Admin 可新增 / 修改 / 刪除電影與資料
  - 一般使用者僅可留言、評分、刪除自己留言
  - 未登入者僅能瀏覽內容

#### 4. 電影詳情頁(movie\_detail.php)

內容模組：

- 基本資訊顯示
  - 電影標題、導演、上映日期、簡介
  - 海報圖片
- 圖片展示
  - 多張劇照(MovieStills)
  - 穿搭推薦圖片與文字(MovieFashion)
- 串流影片預覽
  - 播放器使用 <iframe> 嵌入影片連結
  - 使用「下一部」按鈕輪播其他影片
  - 嵌入影片來源不佔用資料庫容量
- 評論系統
  - 顯示所有評論(包括評分、評論文字、時間、作者)
  - 自動計算平均評分(顯示至小數點一位)
  - 登入使用者可：
    - 發表新評論(含文字 + 評分)
    - 刪除自己留言
  - 未登入使用者僅能檢視

### 2.4.2 後台管理介面

#### 1. 管理員登入

### (1)規則

A.以相同網址進入登入頁面

使用者無論是一般使用者還是管理員，都是透過相同的 login.php 來登入。

B.登入時需要輸入管理員(Admin)的帳號與密碼

系統根據 Users 資料表中 role 欄位來判斷該帳號是否為 Admin。

### (2)技術實現

登入邏輯處理 login\_process.php

```
$username = $_POST['username'];  
$password = $_POST['password'];
```

B.查詢 Users 資料表中對應的帳號(username)

C.檢查密碼

D.根據使用者 role 欄位判斷是否為 Admin

```
$_SESSION['role'] = $user['role'];
```

E.設定 Session 並導向不同頁面

## 2.管理員權限

### (1)規則

A.新增刪減電影清單(index.php)

- 管理員可在 index.php 頁面查看所有電影清單
- 提供「新增電影」按鈕導向 add\_movie.php
- 每部電影旁提供「刪除」按鈕，會執行 delete\_movie.php

B.修改電影相關資訊(movie\_detail.php)

- 使用者(通常為管理員)可在 movie\_detail.php 頁面：
  - 查看特定電影的所有資訊
  - 修改該電影的：

- 基本資料(標題、導演、簡介等)
- 劇照(MovieStills)
- 穿搭推薦(MovieFashion)
- 串流連結(StreamingLinks)
- 影評(Reviews)
- 每一區塊都有「前往編輯」的按鈕, 連結到對應的 edit\_xxxx.php 頁面進行修改

## (2)技術實現

### A. index.php 技術邏輯(電影列表與操作介面)

- 連線資料庫, 從 Movies 表撈出所有電影。
- 用 HTML 表格列出每部電影的基本資訊。
- 每筆資料都有「刪除」按鈕(超連結到 delete\_movie.php)。
- 上方有「新增電影」的超連結導向 add\_movie.php。

### B. add\_movie.php 技術邏輯(新增電影表單)

- 顯示 HTML 表單, 欄位包括:
  - title、director、release\_date
- 使用 POST 方法送出資料

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    $title = trim($_POST['title']);

    $director = trim($_POST['director']);

    $release_date = $_POST['release_date'];
```

- 接收表單資料並 INSERT INTO Movies (...) VALUES (...) 加入資料庫

```
$stmt = $conn->prepare("INSERT INTO Movies (title, director, release_date) VALUES (?, ?, ?)");
```

### C. delete\_movie.php 技術邏輯(刪除功能)

- 使用 GET 取得 movie\_id

```
$stmt2 = $conn->prepare("DELETE FROM Movies WHERE movie_id = ?");

$stmt2->bind_param("i", $movie_id);

$stmt2->execute();
```

#### D. 主頁面: movie\_detail.php

- 接收 movie\_id 參數
- 撈出該電影的資料

```
$movie_sql = "SELECT * FROM Movies WHERE movie_id = ?";

$stmt = $conn->prepare($movie_sql);

$stmt->bind_param("i", $movie_id);

$stmt->execute();
```

- 顯示電影資訊 + 各類子資料(包含 stills、fashion、streaming、review)
- 每個資料區塊附有「修改」按鈕

```
<a href="edit_movie.php?movie_id=<?=$movie_id ?>"class="edit-button">修改</a>

<a href="edit_stills.php?movie_id=<?=$movie_id ?>"class="edit-button">修改</a>

<a href="edit_fashion.php?movie_id=<?=$movie_id ?>"class="edit-button">修改</a>

<button type="submit" style="color: red; border: none; background: none; cursor:
pointer;">🗑️</button>

<a href="edit_streaming.php?movie_id=<?=$movie_id ?>">修改</a>
```

#### E. 編輯頁面範例: edit\_movie.php

- 顯示一個表單, 讓使用者編輯:
  - title, director, genre, release\_date, summary, poster\_url
- POST 更新資料庫:



```
$stmt = $conn->prepare("UPDATE Movies SET title=?, director=?, genre=?, release_date=?,
summary=?, poster_url=? WHERE movie_id=?");

$stmt->bind_param("sssssi", $title, $director, $genre, $release_date, $summary, $poster_url,
$movie_id);

$stmt->execute();
```

其他編輯頁面 edit\_stills.php、edit\_fashion.php、edit\_streaming.php

- 一樣是從資料庫撈出資料 → 顯示在表單中
- 使用者修改後送出 → 執行 UPDATE 動作
- 多筆資料用 foreach ()更新每一筆, 如下

```
foreach ($_POST['stills'] as $still_id => $stillData)
```

## F. 權限驗證

- 每個編輯頁面頂部都有這類權限檢查：

```
if (!isset($_SESSION['role']) || $_SESSION['role'] !== 'Admin') {

    die("你沒有權限訪問此頁面");}
```

## 2.5 操作界面與情境

### 1.使用者角色與操作流程

角色	權限	操作流程
一般訪客	瀏覽電影資訊、劇照、評論	進入首頁 → 點選電影查看詳細資訊

註冊使用者	評論、上傳資訊	註冊 → 登入 → 撰寫評論 / 上傳資訊
管理員(進階使用者)	新增、修改、刪除所有資料	登入 → 進入管理介面 → 維護資料

## 2.操作介面導覽

### (1) 首頁 (index.php)

- 顯示所有電影清單。
- 每部電影皆有「查看詳情」按鈕，導向電影詳細資訊頁。
- 若已登入，將出現「新增電影」、「登出」等操作選項。

登入 | 註冊

電影清單

電影名稱	導演	上映日期	平均評分
<a href="#">conclave</a>	charie	2025-05-30	尚無評分
<a href="#">Avengers</a>	Joss Whedon	2012-05-04	尚無評分
<a href="#">Inception</a>	Christopher Nolan	2010-07-16	4.5
<a href="#">Harry Potter</a>	Chris Columbus	2001-11-16	4.8
<a href="#">The Matrix</a>	Wachowskis	1999-03-31	4.0
<a href="#">Titanic</a>	James Cameron	1997-12-19	5.0

## (2) 使用者註冊與登入

### 註冊 ([register.php](#))

- 填寫帳號、密碼等資訊建立新帳號。
- 表單填寫完畢後按下「註冊」，系統將儲存使用者資料至資料庫。

A screenshot of a web registration form titled "註冊新帳號" (Register New Account). The form is contained within a light gray rounded rectangle. It features three input fields: "帳號:" (Username) with the value "charlie", "電子郵件:" (Email), and "密碼:" (Password) with masked characters ".....". Below these fields is a prominent blue button labeled "註冊" (Register). At the bottom of the form, there is a link that says "已有帳號？點此登入" (Already have an account? Click here to login).

### 登入 ([login.php](#))

- 輸入帳號與密碼。
- 成功後將導向首頁並顯示歡迎訊息。

### 登出 ([logout.php](#))

- 清除登入狀態並導回首頁。



A login form titled "登入" (Login). It contains two input fields: "帳號:" (Username) with the value "charlie" and "密碼:" (Password) with masked characters ".....". Below the fields is a blue button labeled "登入" (Login).

### (3) 電影管理(需登入)

#### 新增電影 ([add\\_movie.php](#))

- 欄位包括電影名稱、年份、導演、類型等。
- 按下「新增」後資料將儲存至資料庫。




A form titled "新增電影" (Add Movie) with a movie icon. It contains three input fields: "電影名稱 (必填)" (Movie Name, required), "導演" (Director), and "上映日期 (必填)" (Release Date, required). The date field has a placeholder "年 / 月 / 日" and a calendar icon. Below the fields is a green button labeled "新增電影" (Add Movie) and a blue link labeled "回到電影頁面" (Back to Movie Page).

#### 編輯電影 ([edit\\_movie.php](#))

- 點選某部電影後，按下「編輯」進入修改頁面。


- 可更新各項電影資訊。

 修改電影資訊

標題：

導演：

類型：

上映日期：  
 

簡介：

海報連結：

[儲存變更](#)

[回到電影頁面](#)

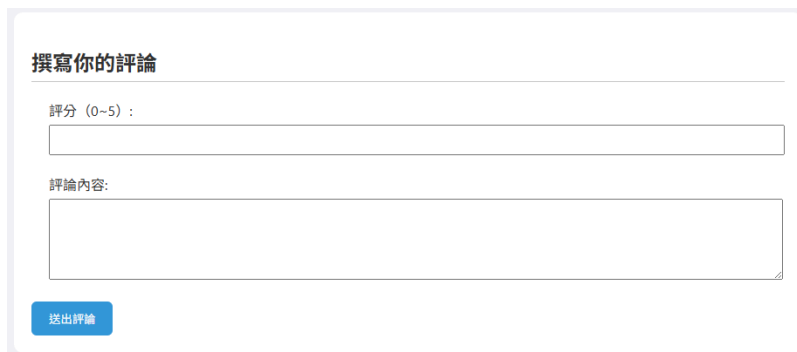
### 刪除電影 (delete\_movie.php)

- 按下「刪除」按鈕後系統會提示確認, 確認後資料刪除。

## (4) 評論功能

### 撰寫評論 (review\_submit.php)

- 登入後於電影詳細頁可填寫評論內容與星等評分(1-5)。
- 評論將顯示於該電影頁下方。



撰寫你的評論

評分 (0~5) :

評論內容:

送出評論

### 刪除評論 (delete\_review.php)

- 僅能刪除自己撰寫的評論。


## (5) 劇照與時尚元素管理

### 劇照管理

- 上傳劇照圖片與描述。
- 可編輯與刪除上傳之劇照資訊。

## 時尚元素管理

- 可新增與電影角色造型有關的時尚描述與搭配。
- 編輯與刪除功能與電影管理類似。



Look 標題 :

Iron Man Suit

圖片連結 :


armor.jpg

描述 :

Futuristic high-tech armor

儲存變更

刪除推薦

 新增穿搭推薦

Look 標題 :

圖片連結 :

描述 :

新增穿搭推薦

[回到電影頁面](#)

## 3.常見使用情境範例

### 情境一：我要新增一部電影

1. 登入系統。

2. 點選「新增電影」。
3. 輸入電影基本資料。
4. 點選「確認送出」。

情境二：我想評論一部剛看完的電影

1. 登入系統。
2. 進入該電影的詳細頁面。
3. 在評論區填寫心得與評分。
4. 點選「提交評論」。

情境三：我要編輯一筆錯誤的電影資料

1. 登入後點選該電影。
2. 點選「編輯」按鈕。
3. 修改所需資訊並儲存。

## 第三章 系統規格表

### 3.1 硬體規格

為確保系統穩定、高效運行，建議配置以下硬體規格：

- 伺服器處理器(CPU)：至少雙核心處理器(Dual-core CPU)，建議採用Intel Xeon或AMD EPYC等企業級處理器，以支援多工作執行緒及長時間運作負



載。

- 記憶體(**RAM**):最低4GB, 建議8GB以上, 以提供足夠資源給後端應用服務、資料庫與快取機制使用。
- 儲存空間(**Storage**):視數據成長量規劃初始容量, 建議採用SSD硬碟以提升I/O效能, 並預留擴充空間。亦可考慮RAID架構以提升資料可靠性。
- 備援電源系統(**UPS**):建議配備不斷電系統, 避免突發斷電造成資料損失。

### 3.2 網路設備與連線環境

- 網路頻寬:應具備穩定且高速的網路連線, 至少100Mbps以上(視系統規模調整)。
- 設備配置:建議採用企業級路由器、防火牆與交換器, 支援VLAN與QoS等網路管理機制。
- 連線穩定性:確保具備固定IP或DNS服務, 並與主機機房維持高可用性連線品質。

### 3.3 軟體環境

- 作業系統:
  - 選項一:Linux(Ubuntu Server、CentOS、Debian等發行版)
  - 選項二:Windows Server 2016以上版本
  - 視實際部署需求與人員熟悉度選擇, 並確保定期安全更新與維護。
- Web伺服器:

- 可選擇Apache HTTP Server、Nginx, 或支援應用語言的其他伺服器（如Tomcat、Gunicorn等）。
- 需具備反向代理、負載平衡能力（視系統規模而定）。
- 開發語言與架構：
  - 前端:HTML5、CSS3、JavaScript（建議搭配框架如Vue.js、React.js）
  - 後端:PHP 7以上、Java（Spring Boot）、Python（Django/Flask）等, 依功能需求與開發團隊技能選擇。
- 安全性防護：
  - 全站使用HTTPS協定, 透過SSL/TLS憑證實現加密傳輸。
  - 配置防火牆與WAF（Web Application Firewall）防範常見攻擊。
  - 實作身份驗證與權限控管機制（如OAuth2、JWT等）。

### 3.4 資料庫管理系統(DBMS)

- 資料庫選型：
  - 採用關聯式資料庫, 如MySQL、MariaDB、PostgreSQL等。
  - 視應用情境決定是否導入NoSQL（如MongoDB）作為輔助資料儲存。
- 資料安全性與備份機制：
  - 設置每日自動備份排程, 並定期驗證備份檔案可恢復性。
  - 備份資料應離線儲存, 並實施版本管理與異地備援。

- 資料庫連線管理與防護：

- 實作參數化查詢(Prepared Statements)以防SQL Injection攻擊。
- 設定使用者帳號權限細分、限制IP連線來源，強化資料庫安全性。
- 配置連線池(Connection Pooling)提升連線效率並避免資源枯竭。

### 分工表

任務項目	負責人	其他
資料庫建立與測試	許家寧	使用 MySQL 建立資料庫與初始資料，確認連線功能
網站基礎架構建立	許家寧	所有頁面建立與設計架構
首頁介面	董承翰	實作電影清單頁面 (index.php)
登入/註冊/登出介面	周俊辰	設計 login.php、register.php、logout.php 相關功能頁面
電影詳情頁介面	王方瑩	實作 movie_detail.php 頁面 照片與影片的路徑連結，不需透過後端資料庫，使用者在前端即可完成(當初跟老師報告時沒有講明這點，讓老師誤解我的意思)
評分與留言區功能	許家寧	設計評論填寫與顯示功能 (含 delete_review.php)
評論新增與刪除邏輯	許家寧	review_submit.php、delete_review.php 資料寫入與刪除

使用者登入與權限管理	周俊辰	帳號驗證、Session 管理與角色分權
電影新增/刪除功能	董承翰	新增電影(add_movie.php)與刪除(delete_movie.php)實作
電影資料修改功能	王方瑩	edit_movie.php 功能與SQL 更新語法實作
照片及影片編輯功能	王方瑩	海報/劇照/穿搭 edit_stills.php、 edit_fashion.php、 edit_streaming.php 功能實作
後台管理介面設計	王方瑩、周俊辰	管理員介面包含電影、劇照、穿搭、串流影片等 CRUD 操作
網站美編	周俊辰	網頁排版與美編
查詢功能	周俊辰	透過電影名稱或倒名稱來查詢電影

## 個人心得

### 許家寧

經過這次資料庫期末專題的實作後，對於初步建立、連結資料庫，以及資料庫可以完成的功能有更全面的瞭解，不僅僅使用到老師上課所介紹的多種應用，也發現藉由不同表的關聯性，可以達成許多資料的運用，原本認為資料放在同一個表或是分成兩個有關係的表差異不大，但實際實作後，發現確實要理解每一個表建立的意義，才能夠正確區分資料庫各表格的設計，也認為在上課中，老師會對於我們容易混淆或產生疑問的地方更加詳細的做講解，讓我在實作時能夠更清楚每一個key應該如何設置，經過這門課的學習，確實很充分的理解了資料庫的基本架構與使用！

#### 41171113H 王方瑩

這是我第一次完整接觸資料庫設計與實作，自己想一遍資料庫的關聯，資料庫間是什麼關係，是一對多，多對多，或是至少為一，都釐清後，再來一步一步完成網站，現在講起來就是短短的幾行文字，但是實操時真的會覺得，啊...一環接著一環，互相卡來卡去。

我主要負責movie\_detail.php的電影詳情、穿搭、劇照與串流功能，這部分需要讀寫與修改資料庫的資料。過程中，我遇到了一個初學者應該都會遇到的問題，當我想要刪除某筆電影資料時，無論怎麼刪都刪不掉，我一直以為是參數沒被讀取到，所以才沒有執行動作，最後才發現是因為關聯表沒有同步刪除，MySQL 的外鍵限制阻止了這項操作。為了解決這個問題，on delete cascade這個語法很重要，要記得!!!才能確保主資料刪除時，關聯資料也能正確清除。

最後要來感謝一下老師，期末demo時，我們這組是額外約時間的，謝謝老師在報告完後跟我們聊了許多，實際關切一下雙主修生面臨到了哪些問題，並且跟我們說明薪水固然重要，但大學時也要記得去摸索自己真正的熱愛。高中畢業後，很少遇到老師會對一個班級那麼用心，熟悉班上學生的名字，對每位同學有初步印象，甚至期末demo設立了獎勵機制，沒有因為學校課程16週，就草草結束，真的誠心感謝老師。此外，助教對這個班上的貢獻也不少，班上同學這麼多，每周幾乎一個作業，一學期過完，應該批改了大於700份作業(?)恩...很佩服，總之，辛苦了~助教們暑假快樂~

#### 41171215H 周俊辰

在這個課程中，我學到了許多有關資料庫的知識，像是我可以看懂ER-diagram想表達的意思、不同key所代表的意思、還有各種SQL查詢語句的意思。這些原本只是抽象的理論，但透過這次的專題，我將所學的觀念實際應用出來，像是在最一開始設計網站時，我們使用ER-diagram來畫出資料庫的架構；還有在實作程式時，我們能夠知道在不同的地方要用那些對應的SQL查詢語句。在製作專題的最後，我們也成功建立了一個簡易的網站，讓其他人可以成功連線近來。這些經驗讓我能夠對之前那些比較抽象的理論有更具體的理解，也讓我了解了資料庫在一個網站中扮演了什麼角色，讓我受益良多。

#### 41171116H 董承翰

這學期的資料庫，讓我收穫非常多。從一開始的資料表設計，到後來實際在網站上操作資料庫，整個過程讓我真正體會到資料庫在專案中的重要性。

像是 ER 圖的部分，之前只是學理論，其實不太熟。這次實際畫出來再照圖實作，就能比較清楚整個資料架構怎麼連在一起，也讓後續的 SQL 查詢更順利。最後真

的要感謝老師和助教，過程中給了我們很多實用的指導，不管是觀念還是實作，幫助都很大。特別是在期末報告時，老師還特別關心我們的學習狀況和未來方向，讓我覺得這不只是一堂技術課，更是一次很有溫度的學習經驗。這次專題讓我不只是學會怎麼用資料庫，更懂得背後的邏輯和設計思維。對我未來如果要做資料相關的工作，會是一個很紮實的基礎。