

星光電影院

系統需求規格書

Software Requirements Specification (SRS) Version: 4.0

姓名	學號	E-mail
張哲瑋	107590004	t107590004@ntut.org.tw
陳昌右	110590454	t110590454@ntut.org.tw
蔡佳彣	110590451	t110590451@ntut.org.tw
莊于潔	110590452	t110590452@ntut.org.tw
李昶泰	110590455	t110590455@ntut.org.tw
陳毅	110590456	t110590456@ntut.org.tw

**Department of Computer Science & Information Engineering
National Taipei University of Technology**

01/10/2024

目錄

Section 1 簡介 (Introduction)	1
1.1 目的 (Purpose)	1
1.2 系統名稱 (Identification)	1
1.3 概觀 (Overview)	2
1.4 符號描述 (Notation Description)	2
Section 2 系統 (System)	4
2.1 系統描述 (System Description)	4
2.1.1 系統架構圖	4
2.2 操作概念 (Operational Concepts)	5
2.3 功能性需求 (Functional Requirements)	7
2.4 技術限制 (Technological Limitations)	7
2.5 介面需求 (Interface Requirements)	8
2.5.1 使用者介面需求 (User Interfaces Requirements)	8
2.5.2 廠商操作介面需求 (Manufacturer's Operating Interface Requirements)	8
2.5.3 內部介面需求 (Inner interface requirements)	9
2.5.4 訪客介面需求 (Guest interface requirements)	9
2.5.5 外部介面需求 (External interface requirements)	9
2.6 非功能性需求 (Non-Functional Requirements)	9
2.6.1 效能需求 (Performance Requirements)	9
2.6.2 資安需求 (Security Requirements)	10
2.7 其他需求 (Other Requirements)	10
2.7.1 環境需求 (Environmental Requirements)	10
2.7.2 安裝需求 (Installation Requirements)	10
2.7.3 測試需求 (Test Requirements)	10
2.8 商業規則與限制 (Business Rules and Integrity Constraints)	11
Section 3 資料庫概念設計 (Conceptual Design of the Database)	11
3.1 Entity Relationship ER Model	11
Section 4 邏輯資料庫綱要 (Logic Database Schema)	12
4.1 Schema of the Database	12
4.2 Domain of the Database	12

4.3 Expectation of the possible DB operations, frequencies and data volumes	16
4.4 SQL Statements Used to Construct the Schema	18
4.5 SQL Statements Used to Insert the data - data population	20
4.6 The implementation of tables in target DBMS	21
4.7 Estimation of the size of each table	22
Section 5 Functional Dependencies and Database Normalization	23
5.1 Functional Dependencies	23
Section 6 The Use of the Database System	24
6.1 System Installation Description	24
6.2 The Use of the System	24
Section 7 Suggestions of Database Turning	34
Section 8 Additional Queries and Views	35
Section 9 Conclusions and Future Work	35
9.1 Conclusions	35
9.2 Future Work	35
Glossary	36
References	38
Appendix	39

Section 1簡介 (Introduction)

1.1目的 (Purpose)

透過本學期修習的資料庫系統課程，我們致力於深入理解資料庫的運作和設計流程，並將過去學習的網頁程式設計、網際網路技術、以及各種程式語言的基礎知識應用於實際情境。為此，我們決定開發一個全功能的電影訂票系統，名為「星光電影院」。這個系統將讓使用者能夠輕鬆進行電影訂票、選擇座位，並享受便捷的購票體驗。我們的目標是提供一個精緻且易於使用的界面，讓每位使用者都感受到賓至如歸的服務和極致的電影體驗。

本系統主要目標為：

- 商場管理者可以輕鬆管理電影院的場次、票價和座位安排。
- 商場管理者可以查詢電影院的總營業額和各場次的銷售統計資料。
- 電影院的影片管理者可以輕鬆管理影片的上映時間表和座位安排。
- 顧客可以在線上瀏覽電影資訊、搜索電影場次，並輕鬆訂購電影票。
- 顧客可以查詢他們的購物車內的訂單和目前的訂票狀態。
- 顧客可以為觀看過的電影評分，幫助其他顧客做出選擇。

1.2系統名稱 (Identification)

本專案範圍包含建置下面主系統與各項子系統，主系統為：

線上商場系統 (Web-based Mall Business System, WMBS)

各子系統分別為：

- 影城管理子系統 (Cinema Management Subsystem, CMS)
- 客服管理子系統 (Customer Service Management Subsystem, CSMS)
- 座位管理子系統 (Seat Management Subsystem, SMS)
- 會員管理子系統 (Account and Member Management Subsystem, AMMS)
- 購物車子系統 (Shopping Cart Management Subsystem, SCMS)
- 商品管理子系統 (Product Management Subsystem, PMS)
- 財務管理子系統 (Financial Management Subsystem, FMS)
- 商品搜尋推薦管理子系統 (Search and Recommend Management Subsystem, SRMS)
- 資料庫子系統 (Database System, DBS)

1.3概觀 (Overview)

為了適應現代網頁趨勢並提高開發效率，我們決定將系統分為前端和後端，採用前後端分離的架構。這種架構的主要優勢在於可以讓前端和後端的開發團隊專注於各自的工作，降低了彼此之間的耦合性，使得系統更容易維護和擴展。

在前端方面，我們選擇了 React 作為前端框架。React 是一個廣泛使用的前端框架，擁有龐大的生態系統和活躍的社群支持。它能夠協助我們建立動態且高效的用戶界面，並輕鬆處理前端的互動和顯示邏輯。

至於後端，我們選擇了 Python 作為後端語言，並採用 Flask 框架來構建後端應用程序。Flask 是一個輕量級的微框架，適合快速開發和部署，同時具有良好的擴展性和可維護性。這使得我們能夠輕鬆處理後端的業務邏輯並提供必要的 API 接口。

最後，作為系統的資料存儲引擎，我們選擇了 MariaDB，一個開源的關聯式資料庫管理系統。MariaDB具有高性能、可擴展性和安全性的特點，能夠滿足系統的資料管理需求。

總的來說，這個系統架構將有助於我們以現代方式開發和部署應用程序，並採用了現代和流行的技術選擇。希望這個新的系統架構能夠幫助我們實現系統的需求，同時提高開發效率。

1.4 符號描述 (Notation Description)

WMBS 1.0.0	The WMBS system will be labeled with the number 1.0.0.
CMS 1.0.n	The CMS system will be labeled with the number 1.1.n.
CSMS 1.1n	The CSMS system will be labeled with the number 1.2.n.
SMS 1.2.n	The SMS system will be labeled with the number 1.3.n.
AMMS 1.3.n	The AMMS system will be labeled with the number 1.4.n.
SCMS 1.4.n	The SCMS system will be labeled with the number 1.5.n.
PMS 1.5.n	The PMS system will be labeled with the number 1.6.n.
FMS 1.6.n	The FMS system will be labeled with the number 1.7.n.
SRMS 1.7.n	The SRMS system will be labeled with the number 1.8.n.
DBS 1.8.n	The DBS system will be labeled with the number 1.9.n.

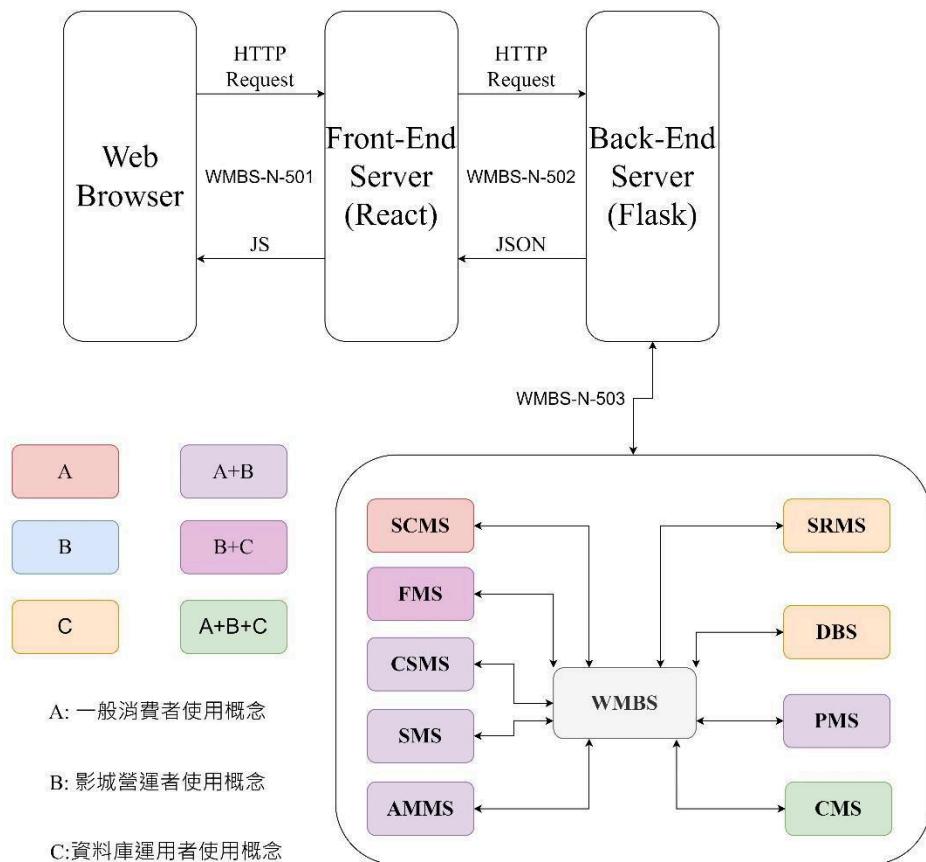
WMBS-F-nnn	WMBS功能性需求(Functional Requirements)
WMBS-N-nnn	WMBS非功能性需求(Non-Functional Requirements)
CMS-F-nnn	CMS功能性需求(Functional Requirements)
CMS-N-nnn	CMS非功能性需求(Non-Functional Requirements)
CSMS-F-nnn	CSMS功能性需求(Functional Requirements)
CSMS-N-nnn	CSMS非功能性需求(Non-Functional Requirements)
SMS-F-nnn	SMS功能性需求(Functional Requirements)
SMS-N-nnn	SMS非功能性需求(Non-Functional Requirements)
AMMS-F-nnn	AMMS功能性需求(Functional Requirements)
AMMS-N-nnn	AMMS非功能性需求(Non-Functional Requirements)
SCMS-F-nnn	SCMS功能性需求(Functional Requirements)
SCMS-N-nnn	SCMS非功能性需求(Non-Functional Requirements)
PMS-F-nnn	PMS功能性需求(Functional Requirements)
PMS-N-nnn	PMS非功能性需求(Non-Functional Requirements)
FMS-F-nnn	FMS功能性需求(Functional Requirements)
FMS-N-nnn	FMS非功能性需求(Non-Functional Requirements)
SRMS-F-nnn	SRMS功能性需求(Functional Requirements)
SRMS-N-nnn	SRMS非功能性需求(Non-Functional Requirements)
DBS-F-nnn	DBS功能性需求(Functional Requirements)
DBS-N-nnn	DBS非功能性需求(Non-Functional Requirements)

Section 2 系統(System)

2.1系統描述 (System Description)

本系統主要分為九個部分，分別為商場管理子系統(Cinema Management Subsystem, CMS 1.1.0)、客服管理子系統 (Customer Service Management Subsystem, CSMS 1.2.0)、座位管理子系統(Seat Management Subsystem, SMS 1.3.0)、會員管理子系統(Account and Member Management Subsystem, AMMS 1.4.0)、購物車子系統(Shopping Cart Management Subsystem, SCMS 1.5.0)、商品管理子系統(Product Management Subsystem, PMS 1.6.0)、財務管理子系統(Financial Management Subsystem, FMS 1.7.0)、商品搜尋推薦管理子系統(Search and Recommend Management Subsystem, SRMS 1.8.0)、資料庫子系統(Database System, DBS 1.9.0)。

2.1.1系統架構圖



2.2操作概念 (Operational Concepts)

Scenario A, 一般消費者使用概念 (Client Operational Concepts)

消費者使用者經由「影城管理子系統CMS」選擇電影。可以使用「座位管理子系統SMS」，選擇座位。選購電影票後，加入「購物車子系統SCMS」。購票前須申請會員，進入「會員管

理子系統AMMS」。確認訂單以及追蹤訂單進入「商品管理子系統PMS」。針對電影票問題，透過「客服管理子系統CSMS」進行客服。

操作描述	分類說明
電影Logo, 點擊任電影Logo後進入該電影簡介及購票網站。	首頁
電影搜尋, 輸入想觀看的電影, 並顯示該電影相關資訊。	首頁
註冊, 點擊主頁註冊按鈕後於新分頁顯示會員註冊頁面。	首頁
登入, 點擊主頁登入按鈕後於新分頁顯示會員登入頁面。	首頁
購物車, 點擊主頁顯示購物車內容頁面。	首頁
訂單確認, 點擊訂單追蹤後顯示目前已完成即未完成付款的所有票券。	首頁
詢問客服, 點擊詢問客服按鈕後顯示與對話窗口。	首頁
註冊具有輸入使用者名稱、E-mail、手機號碼、使用者密碼、確認密碼, 並具有使用者條款與確認按鈕。	新會員註冊
登入可以輸入帳號、密碼、具有忘記密碼功能與確認登入按鈕。	會員登入
忘記密碼功能能夠將使用者引導至重設密碼頁面。	會員登入
會員登入成功後會引導至首頁, 並可進行訂票等功能。	會員成功登入
忘記密碼功能可讓使用者輸入 E-mail 帳號, 並發送密碼重置驗證信。	會員忘記密碼
輸入新密碼及確認新密碼並按下確認按鈕後, 將頁面導引至會員登入頁面。	新密碼設置
顯示電影海報、劇情簡介, 點擊下方訂購票券後引導至購票頁面, 點擊有興趣按鈕後將該電影加入購物車內。	電影簡介
選擇想觀看的影城場次, 並挑選張數後, 將其引導至座位挑選頁面。	購票頁面
呈現real-time的座位表, 讓使用者自行劃位, 並按下確認按鈕後引導至確認資訊頁面。	座位挑選頁面
確認票券張數、位置、時間等正確後, 按下確認按鈕後導引至付款頁面。	確認資訊頁面
讓使用者選擇以信用卡付款或是稍後付款, 選擇前者則須輸入信用卡資訊並導入確認付款頁面; 後者則直接導引至主頁並在訂單確認中新增此項目, 狀態顯示未付款。	付款頁面
等待付款成功後, 將頁面導引至主頁並在訂單確認中新增此項目, 狀態顯示已付款, 並將電子票券寄送至使用者所綁定的E-mail。	確認付款頁面
點選未完成付款票券, 並按下付款後引導至付款頁面。	訂單確認頁面
顯示已加至購物車的所有電影, 並點選該電影Logo導入該電影簡介及購票網站。	購物車頁面
基本資料顯示, 文字顯示, 會員名稱/E-mail/手機號碼。編輯符號, 按鈕/點擊, 修改基本資料(會員名稱/E-mail/手機號碼)。	會員資訊頁面
點擊修改資料, 導引至修改基本資料頁面。	會員資訊頁面
輸入密碼後方可讓使用者更改會員名稱/E-mail/手機號碼/會員密碼。	修改資料頁面
輸入欲詢問之問題, 並會有專員為使用者服務。	詢問客服頁面

Scenario B, 影城營運管者使用概念(Operating Officer Concepts)

透過「影城管理子系統CMS」，讓消費者了解影城資訊，以及知道為什麼選擇我們的影城。透過「商品管理子系統PMS」及「座位管理子系統SMS」，管理電影票狀態：包括上架、下架電影票，管理電影票數量、定價，發放調整電影票文案。透過「客服管理子系統CSMS」，針對消費者問題進行客服。透過「會員管理子系統AMMS」了解消費者個別購物情形，客制定商品投放資訊。透過「財務管理子系統FMS」，公式自動生成財務報表，進而分析影城的財務狀況。

Scenario C, 資料庫運用者使用概念(Database user Concepts)

消費者購物數據於「資料庫子系統DBS」，結合資料庫與市場調查數據資訊，經由營運、行銷管理者與數據分析管理者之結果分析，再制定、調整演算法。透過「商品搜尋推薦管理子系統SRMS」方便快速推薦消費者想要、要搜尋的電影。透過「影城管理子系統CMS」，不僅顯示符合搜尋字詞的所有電影票，同時推薦相關電影以及消費者可能喜歡之電影。透過「財務管理子系統FMS」，制定商品財務會計及營運分析報表公式於每期自動生成報表，提供營運、行銷管理者與數據分析管理者之結果分析。

2.3功能性需求 (Functional Requirements)

WMBS-F-001	電影訂票系統提供會員註冊/修改資料。
WMBS-F-002	可依照搜尋電影進入商品瀏覽頁面，其包含電影與推薦電影。
WMBS-F-003	可於商品瀏覽頁面選擇電影看其詳情進入電影資訊頁面。
WMBS-F-004	可添加商品進購物車，購物車內容頁面可更新。
WMBS-F-005	購物車內之商品可選擇結帳進入付款與取票方式選擇頁面。
WMBS-F-006	顧客可以追蹤付款紀錄於歷史訂單及付款查詢頁面。
WMBS-F-007	提供常見問題頁面解決可能之問題。
WMBS-F-008	提供客服留言頁面解決個別問題。
WMBS-F-009	透過行銷管理資料報表可制定行銷活動。
WMBS-F-010	透過營運管理資料報表可了解營運狀況。
WMBS-F-011	透過商品管理子系統可上下架與編輯電影資訊。
WMBS-F-012	透過演算法分析資料庫，達到搜尋引擎優化。
WMBS-F-013	透過演算法分析資料庫，達到電影推薦。

2.4技術限制 (Technological Limitations)

WMBS-N-001	電影資訊。
WMBS-N-002	記錄消費者搜尋電影以優化搜尋。
WMBS-N-003	記錄消費者訂購相關電影票以優化推薦系統。
WMBS-N-004	金流系統(線上刷卡)
WMBS-N-005	會員認證(OAuth2 串接 Facebook 、 Google)

2.5界面需求 (Interface Requirements)

WMBS-N-1 01	電影瀏覽介面
----------------	--------

WMBS-N-1 02	購票車介面
WMBS-N-1 03	搜尋介面
WMBS-N-1 04	會員登入及註冊介面
WMBS-N-1 05	訂單查詢介面
WMBS-N-1 06	電影管理介面
WMBS-N-1 07	財務報表介面
WMBS-N-1 08	影城管理介面

2.5.2 廠商操作介面需求(Manufacturer's Operating Interface Requirements)

WMBS-N-2 01	可上/下架電影票券
WMBS-N-2 02	可上/下架周邊商品
WMBS-N-2 03	可設定商品優惠
WMBS-N-2 04	檢視商品庫存
WMBS-N-2 05	檢視商品銷售狀況

2.5.3 內部介面需求 (Inner interface requirements)

WMBS-N-3 01	可購買電影票券或周邊商品
----------------	--------------

WMBS-N-3 02	可追蹤特殊影城
WMBS-N-3 03	可使用結帳頁面選擇付款方式
WMBS-N-3 04	檢視已購買的電影票券
WMBS-N-3 05	會員資料管理
WMBS-N-3 06	登出系統

2.5.4 訪客介面需求(Guest interface requirements)

WMBS-N-4 01	註冊會員
----------------	------

2.5.5 外部介面需求(External interface requirements)

WMBS-N-501	使用者使用瀏覽器通過 HTTP 通訊協定與 WMBS Web 主機前端通訊
WMBS-N-502	前端透過 HTTP 發送 API 請求給後端 server
WMBS-N-503	後端透過 Flask 解讀 web 所發送之 API, 存取本系統

2.6 非功能性需求 (Non-Functional Requirements)

2.6.1 效能需求(Performance Requirements)

WMBS-N-6 01	載入主頁時間不可超過1.5秒
WMBS-N-6 02	搜尋相關電影時間不可超過3秒
WMBS-N-6 03	選擇電影座位的資料須為real-time

WMBS-N-6 04	線上刷卡功能應於20秒內完成
WMBS-N-6 05	訂票流程化簡，應不超過5步驟
WMBS-N-6 06	高流量時無服務中斷

2.6.2 資安需求(Security Requirements)

WMBS-N-701	以 Acunetix 進行弱點掃描，確保網站無任何可利用弱點，或是評估過後可不修補之低風險弱點。
WMBS-N-702	以 Fortify 進行靜態程式碼掃描，確保無程式碼誤用造成弱點產生。
WMBS-N-703	對網站進行白箱測試，確保無程式碼誤用造成弱點產生。
WMBS-N-704	以 Nessus 對 Server 進行掃描，確保 Server 無任何已知風險。

2.7 其他需求 (Other Requirements)

2.7.1 環境需求 (Environmental Requirements)

WMBS-N-801	網站能夠在 Windows 運行且通過單元測試。
WMBS-N-802	網站能夠在 Linux Ubuntu 18.04, 20.04, 22.04 運行且通過單元測試。
WMBS-N-803	網站能夠在 macOS 運行且通過單元測試。

2.7.2 安裝需求(Installation Requirements)

WMBS-N-901	用戶端無需安裝額外的軟體。訪問系統只需標準的Web瀏覽器。
------------	-------------------------------

2.7.3 測試需求(Test Requirements)

WMBS-N-100 1	每個子系統功能都應被測試過。
WMBS-N-100 2	提供使用者GUI介面的測試，確認一致性。
WMBS-N-100 3	提供各式平台使用系統的測試。
WMBS-N-100	測試不同使用者角色功能和權限。

4	
WMBS-N-100	測試最大容納能力，避免性能問題。
5	

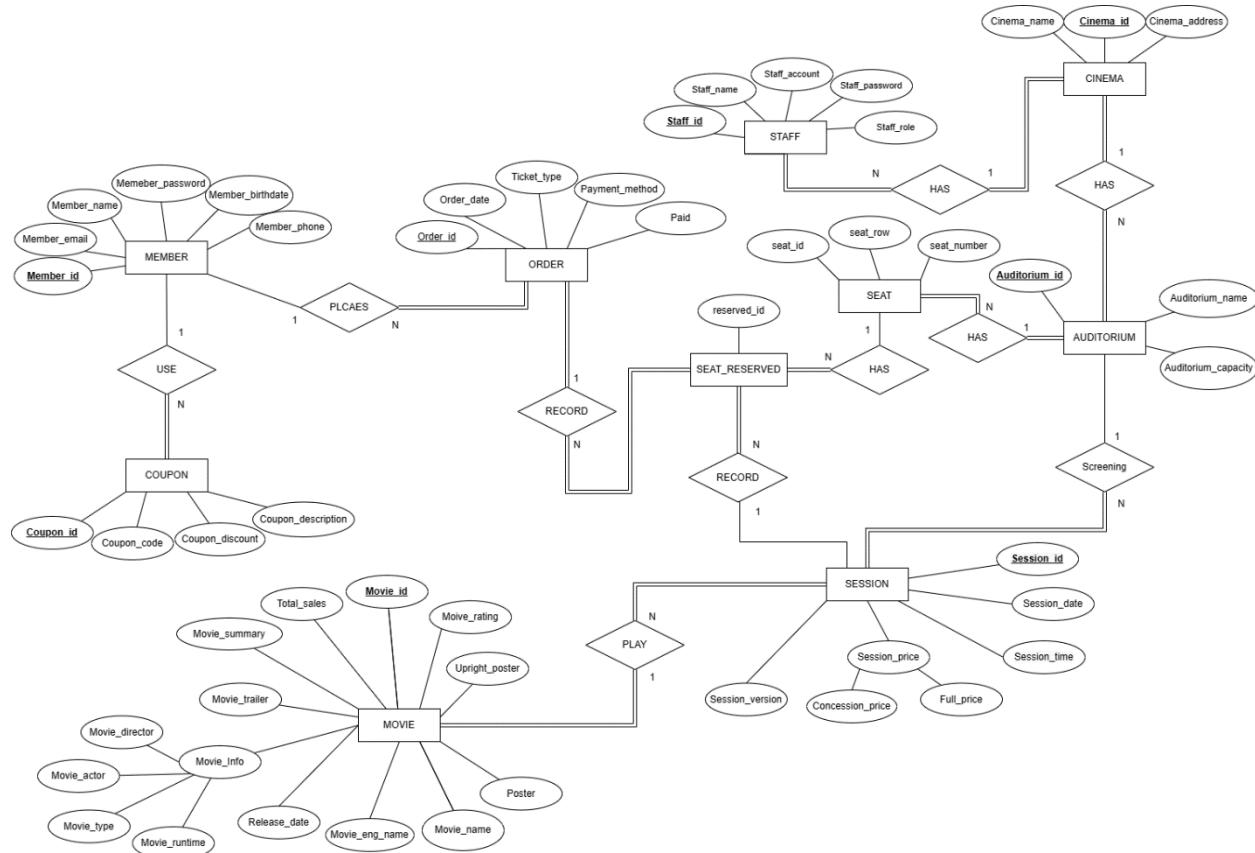
2.8商業規則與限制 (Business Rules and Integrity Constraints)

星光電影院使用 MIT 授權條款，授予任何人免費獲得本軟體和相關文件檔案「軟體」副本的許可，不受限制地處理本軟體，包括但不限於使用、複製、修改、合併、發布、分發、再許可的權利，被授權人有權利使用、複製、修改、合併、出版發行、散布、再授權/或販售軟體及軟體的副本，及授予被供應人同等權利。

這是測試。

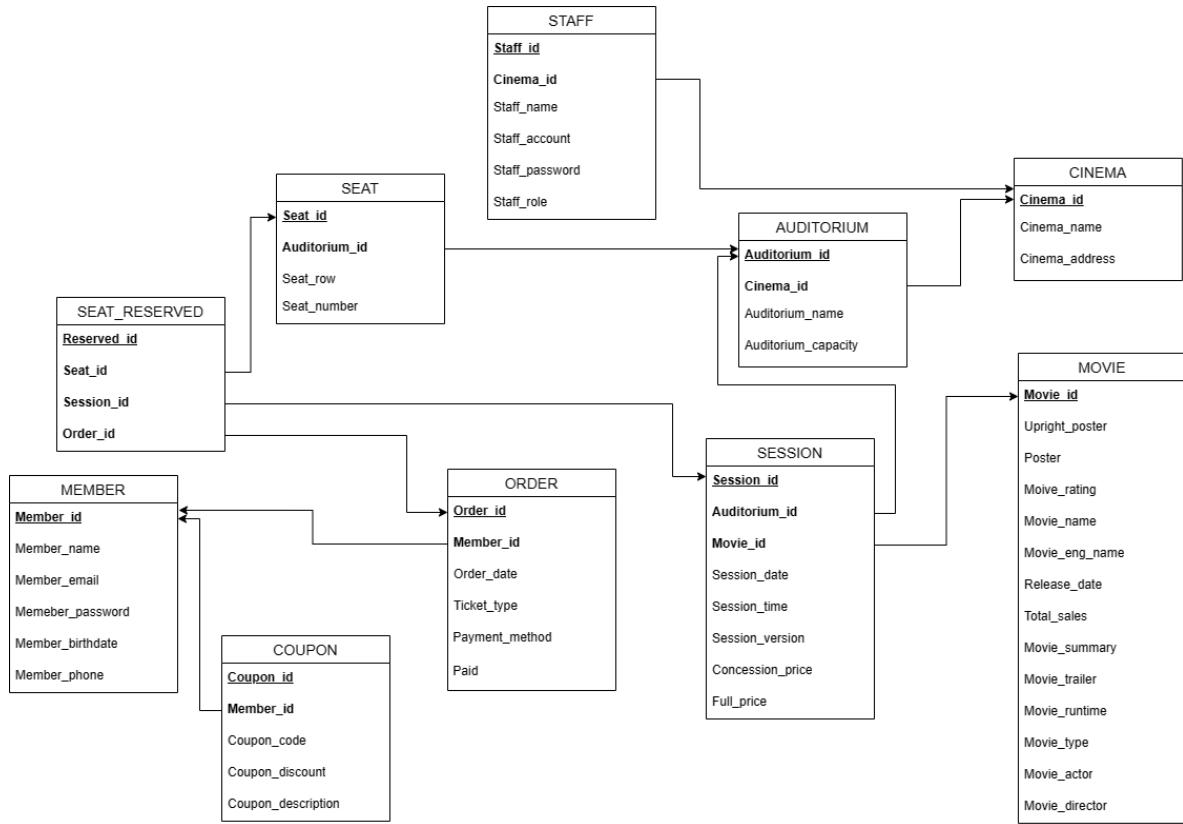
Section3 資料庫概念設計(Conceptual Design of the Database)

3.1 Entity Relationship ER Model



Section4 邏輯資料庫綱要 (Logic Database Schema)

4.1 Schema of the Database



4.2 Domain of the Database

MEMBER				
Description: 存放會員資料				
Attribute	Domain Type	Key	Nullable	Description
Member_id	integer	primary	not null	會員編號
Member_name	varchar	unique	not null	會員暱稱
Member_email	varchar	unique	not null	會員帳號
Memeber_password	TEXT		not null	會員密碼
Member_birthdate	datetime		not null	會員生日

Member_phone	varchar	unique	not null	會員手機號碼
---------------------	---------	--------	----------	--------

COUPON				
Description: 存放優惠卷資料				
Attribute	Domain Type	Key	Nullable	Description
Coupon_id	integer	primary	not null	優惠券編號
Member_id	integer	foreign	not null	會員編號
Coupon_code	varchar	unique	not null	優惠券代號
Coupon_discount	integer		not null	優惠券折扣金額
Coupon_description	varchar		not null	優惠券敘述

SEAT_RESERVED				
Description: 存放座位保留資料				
Attribute	Domain Type	Key	Nullable	Description
Reserved_id	integer	primary	not null	保留編號
Seat_id	integer	foreign	not null	座位編號
Session_id	integer	foreign	not null	場次編號
Order_id	integer	foreign	not null	訂單編號

SEAT				
Description: 存放座位資料				
Attribute	Domain Type	Key	Nullable	Description
Seat_id	integer	primary	not null	座位編號
Auditorium_id	integer	foreign	not null	影廳編號
Seat_row	char		not null	座位列別

Seat_number	varchar		not null	座位號碼
--------------------	----------------	--	-----------------	------

ORDER				
Description: 存放訂單資料				
Attribute	Domain Type	Key	Nullable	Description
Order_id	integer	primary	not null	訂單編號
Member_id	integer	foreign	not null	會員編號
Reserved_id	integer	foreign	not null	座位保留編號
Order_date	datetime		not null	訂單日期
Ticket_type	varchar		not null	訂單類別
Payment_method	bool		not null	付款方式 (0 信用卡 1 現金)
Paid	bool		not null	是否付款

STAFF				
Description: 存放影城員工資料				
Attribute	Domain Type	Key	Nullable	Description
Staff_id	integer	primary	not null	員工編號
Cinema_id	integer	foreign	not null	影城編號
Staff_name	varchar		not null	員工姓名
Staff_account	varchar	unique	not null	員工帳號
Staff_password	TEXT		not null	員工密碼
Staff_role	varchar		not null	員工權限

AUDITORIUM				
Description: 存放影廳資料				
Attribute	Domain Type	Key	Nullable	Description
Auditorium_id	integer	primary	not null	影廳編號
Cinema_id	integer	foreign	not null	影城編號
Auditorium_name	varchar	unique	not null	影廳名稱
Auditorium_capacity	integer		not null	影廳座位數

CINEMA				
Description: 存放影城資料				
Attribute	Domain Type	Key	Nullable	Description
Cinema_id	integer	primary	not null	影城編號
Cinema_name	varchar	unique	not null	影城名稱
Cinema_address	varchar	unique	not null	影城地址

SESSION				
Description: 存放場次的相關資料				
Attribute	Domain Type	Key	Nullable	Description
Session_id	integer	primary	not null	場次編號
Auditorium_id	integer	foreign	not null	影廳編號
Movie_id	integer	foreign	not null	電影編號
Session_date	datetime		not null	場次日期
Session_time	time		not null	場次時間
Session_version	varchar		not null	場次版本
Concession_price	integer		not null	優惠票價

Full_price	integer		not null	全票票價
------------	---------	--	----------	------

MOVIE				
Description: 存放電影資料				
Attribute	Domain Type	Key	Nullable	Description
Movie_id	integer	primary	not null	電影編號
Upright_poster	varchar		not null	直式海報
Poster	varchar		not null	海報
Moive_rating	float		not null	電影評分
Movie_name	varchar	unique	not null	電影名稱
Movie_eng_name	varchar	unique	not null	電影英文名稱
Release_date	datetime		not null	發布日期
Total_sales	integer		not null	總銷售額
Movie_summary	varchar		not null	電影簡介
Movie_trailer	varchar		not null	電影預告片
Movie_runtime	time		not null	電影時長
Movie_type	varchar		not null	電影類別
Movie_actor	varchar		not null	電影演員
Movie_director	varchar		not null	電影導演

4.3 Expectation of the possible DB operations, frequencies and data volumes

Table	可能操作	預估使用頻率 (per day)	表格資料量 (tuples)	系統負擔 (worst case)
member	User 登錄身分驗證	2000	50000	1000000次 Query / day
member	新增會員資料	200	50000	50000次 Insert / day

staff	新增員工帳號資料	Not often	1000	don't care
staff	修改員工帳號資料	Not often	1000	don't care
staff	登錄員工帳號	30	1000	1000次 Insert / day
auditorium	新增影廳資料	Not often	1000	don't care
auditorium	修改影廳資料	Not often	1000	don't care
movie	新增電影資料	Not often	50000	don't care
movie	修改電影資料	Not often	50000	don't care
movie	選擇電影	1000	50000	1000次 Insert / day
cinema	新增影城資料	Not often	50000	don't care
cinema	修改影城資料	Not often	50000	don't care
cinema	選擇影城	1000	50000	1000次 Insert / day
order	新增訂單	1000	50000	30000次 Query / day
order	修改訂單	500	50000	250000次 Insert / day
coupon	新增優惠券	Not often	1000	don't care
coupon	修改優惠券	Not often	1000	don't care
coupon	輸入優惠券	800	1000	500000次 Query / day
session	新增場次	15	50000	125000 Insert / day
session	修改場次	15	50000	50000次 Query / day
session	選擇場次	1000	50000	100次 Insert / day
seat	新增座位	Not often	50000	don't care
seat	修改座位	Not often	50000	don't care
seat	選擇座位	1000	50000	1000次 Insert / day
seat_reserved	記錄座位是否被選擇	1000	50000	1000次 Insert / day

4.4 SQL Statements Used to Construct the Schema

```
CREATE DATABASE IF NOT EXISTS StarlightCinema;
USE StarlightCinema;

CREATE TABLE `cinema` (
    `cinema_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
    `name` VARCHAR(255) NOT NULL,
    `address` VARCHAR(255) NOT NULL,
    CONSTRAINT `pk_cinema` PRIMARY KEY (`cinema_id`),
    CONSTRAINT `uk_cinema_name` UNIQUE (`name`)
);

CREATE TABLE `auditorium` (
    `auditorium_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
    `cinema_id` INT UNSIGNED NOT NULL,
    `name` VARCHAR(255) NOT NULL,
    `capacity` INT UNSIGNED NOT NULL,
    CONSTRAINT `pk_auditorium` PRIMARY KEY (`auditorium_id`),
    CONSTRAINT `uk_auditorium_name` UNIQUE (`name`),
    CONSTRAINT `fk_auditorium_cinema` FOREIGN KEY (`cinema_id`) REFERENCES `cinema` (`cinema_id`) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE `movie` (
    `movie_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
    `upright_poster` VARCHAR(2083) NOT NULL,
    `poster` VARCHAR(2083) NOT NULL,
    `rating` TINYINT UNSIGNED NOT NULL,
    `name` VARCHAR(255) NOT NULL,
    `eng_name` VARCHAR(255) NOT NULL,
    `release_date` DATE NOT NULL,
    `total_sales` INT UNSIGNED NOT NULL,
    `summary` TEXT NOT NULL,
    `trailer` VARCHAR(2083) NOT NULL,
    `runtime` TIME NOT NULL,
    `type` VARCHAR(255) NOT NULL,
    `actor` VARCHAR(255) NOT NULL,
    `director` VARCHAR(255) NOT NULL,
    CONSTRAINT `pk_movie` PRIMARY KEY (`movie_id`),
    CONSTRAINT `uk_movie_name` UNIQUE (`name`),
    CONSTRAINT `uk_movie_eng_name` UNIQUE (`eng_name`)
);

CREATE TABLE `session` (
    `session_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
    `auditorium_id` INT UNSIGNED NOT NULL,
    `movie_id` INT UNSIGNED NOT NULL,
    `play_date` DATETIME NOT NULL,
    `version` VARCHAR(255) NOT NULL,
    `concession_price` INT UNSIGNED NOT NULL,
    `full_price` INT UNSIGNED NOT NULL,
    CONSTRAINT `pk_session` PRIMARY KEY (`session_id`),
    CONSTRAINT `fk_session_auditorium` FOREIGN KEY (`auditorium_id`) REFERENCES `auditorium` (`auditorium_id`) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT `fk_session_movie` FOREIGN KEY (`movie_id`) REFERENCES `movie` (`movie_id`) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE `member` (
    `member_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
    `email` VARCHAR(255) NOT NULL,
    `name` VARCHAR(70) NOT NULL,
    `password` TEXT NOT NULL,
    `birthday` DATETIME NOT NULL,
    `phone` VARCHAR(20) NOT NULL,
    CONSTRAINT `pk_member` PRIMARY KEY (`member_id`),
    CONSTRAINT `uk_member_email` UNIQUE (`email`),
    CONSTRAINT `uk_member_name` UNIQUE (`name`),
    CONSTRAINT `uk_member_phone` UNIQUE (`phone`)
);
```

```

CREATE TABLE `staff` (
    `staff_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
    `cinema_id` INT UNSIGNED NOT NULL,
    `name` VARCHAR(70) NOT NULL,
    `account` VARCHAR(255) NOT NULL,
    `password` VARCHAR(255) NOT NULL,
    `role` VARCHAR(255) NOT NULL,
    CONSTRAINT `pk_staff` PRIMARY KEY (`staff_id`),
    CONSTRAINT `uk_staff_account` UNIQUE (`account`),
    CONSTRAINT `fk_staff_cinema` FOREIGN KEY (`cinema_id`) REFERENCES `cinema` (`cinema_id`) ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE `order` (
    `order_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
    `member_id` INT UNSIGNED NOT NULL,
    `date` DATETIME NOT NULL,
    `ticket_type` TINYINT UNSIGNED NOT NULL,
    `payment_method` TINYINT UNSIGNED NOT NULL,
    `paid` TINYINT UNSIGNED NOT NULL,
    CONSTRAINT `pk_order` PRIMARY KEY (`order_id`),
    CONSTRAINT `fk_order_member` FOREIGN KEY (`member_id`) REFERENCES `member` (`member_id`) ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE `seat` (
    `seat_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
    `auditorium_id` INT UNSIGNED NOT NULL,
    `row` CHAR(1) NOT NULL,
    `number` INT UNSIGNED NOT NULL,
    CONSTRAINT `pk_seat` PRIMARY KEY (`seat_id`),
    CONSTRAINT `fk_seat_auditorium` FOREIGN KEY (`auditorium_id`) REFERENCES `auditorium` (`auditorium_id`) ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE `seat_reserved` (
    `seat_reserved_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
    `seat_id` INT UNSIGNED NOT NULL,
    `session_id` INT UNSIGNED NOT NULL,
    `order_id` INT UNSIGNED NOT NULL,
    CONSTRAINT `pk_seat_reserved` PRIMARY KEY (`seat_reserved_id`),
    CONSTRAINT `fk_seat_reserved_seat` FOREIGN KEY (`seat_id`) REFERENCES `seat` (`seat_id`) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT `fk_seat_reserved_session` FOREIGN KEY (`session_id`) REFERENCES `session` (`session_id`) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT `fk_seat_reserved_order` FOREIGN KEY (`order_id`) REFERENCES `order` (`order_id`) ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE `coupon` (
    `coupon_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
    `member_id` INT UNSIGNED NOT NULL,
    `code` VARCHAR(255) NOT NULL UNIQUE,
    `discount` INT UNSIGNED NOT NULL,
    `description` TEXT NOT NULL,
    CONSTRAINT `pk_coupon` PRIMARY KEY (`coupon_id`),
    CONSTRAINT `fk_coupon_member` FOREIGN KEY (`member_id`) REFERENCES `member` (`member_id`) ON UPDATE CASCADE ON DELETE CASCADE
);

```

4.5 SQL Statements Used to Insert the data - data population

```
USE StarlightCinema;
-- 插入到 `cinema`
INSERT INTO cinema (name, address) VALUES
('星光電影院', '台北市信義區松高路100號'),
('月光影城', '高雄市左營區明華路200號'),
('陽光影城', '台中市西屯區文華路300號');

-- 插入到 `auditorium`
INSERT INTO auditorium (cinema_id, name, capacity) VALUES
(1, '大廳A', 120),
(2, '大廳B', 150),
(3, '大廳C', 200);

-- 插入到 `movie`
INSERT INTO movie (upright_poster, poster, rating, name, eng_name, release_date, total_sales, summary, trailer, runtime, type, actor, director) VALUES
('https://m.media-amazon.com/images/M/MV5BMzUzNDM2NzM2MV5BM15BanBnXkFtZTgwNTM3NTg4OTE@._V1_.jpg', 'https://ntvb.tmsimg.com/assets/p12386480_v_h10_az.jpg?w=1280&h=720', 8, '夢想之城', 'La La Land', '2016-12-09', 445000000, '《夢想之城》是一部愛情音樂劇情片，由達米恩·查澤雷執導，瑞恩·高斯林和艾瑪·史東主演。影片講述了一名有抱負的女演員和一名才華洋溢的爵士音樂家在洛杉磯相遇，發展出一段充滿激情和音樂的愛情故事。電影榮獲多項奧斯卡獎，包括最佳導演和最佳女主角。', 'https://www.youtube.com/watch?v=0pdqf4P9MB8', '02:08:00', '愛情/音樂', '瑞恩·高斯林, 艾瑪·史東', '達米恩·查澤雷'),
('https://m.media-amazon.com/images/M/MV5BNDYXNjQyMjAtNTdiOS00NGYwLWFmNTAtNThmYjU5ZGI2YTI1XkEyXkFqcGdeQXVyMTMxODk2OTU@._V1_.jpg', 'https://variety.com/wp-content/uploads/2014/04/01-avengers-2012.jpg', 7, '復仇者聯盟', 'The Avengers', '2012-04-11', 1518812988, '《復仇者聯盟》是一部超級英雄電影，由小羅伯特·唐尼、克里斯·埃文斯等主演，由喬斯·韋登執導。電影講述了地球上最強大的英雄必須聯手對抗一位威脅全球的反派。影片獲得了廣泛的讚譽，並成為漫威電影宇宙的重要一部分。', 'https://www.youtube.com/watch?v=eOrNdBpGMv8', '02:23:00', '動作/科幻', '小羅伯特·唐尼, 克里斯·埃文斯', '喬斯·韋登'),
('https://m.media-amazon.com/images/M/MV5BMyJ1wMjE1Nzc4NzC4NV5BM15BanBnXkFtZTgwNDg40TA1NzM@._V1_.jpg', 'https://media-amazon.com/images/M/MV5BMTMyNjg0MzQwM15BM15BanBnXkFtZTcwMTU2NTI3Ng@._V1_.jpg', 9, '獅子王', 'The Lion King', '1994-06-15', 968511805, '《獅子王》是一部迪士尼動畫經典，講述了一名年輕的獅子王子辛巴的成長故事，他必須挑戰邪惡的叔叔綠蔭獅子斯卡的統治，並恢復他的王國。影片以它精美的動畫和動人的音樂而聞名，深受觀眾喜愛。', 'https://www.youtube.com/watch?v=4sj1MT05lAA', '01:29:00', '動畫/劇情', '馬修·布洛德里克, 杰瑞米·艾恩斯', '羅傑·奧勒斯, 羅伯·明卡夫');

-- 插入到 `session`
INSERT INTO session (auditorium_id, movie_id, play_date, version, concession_price, full_price) VALUES
(1, 1, '2023-10-01 19:00:00', '原版', 200, 280),
(2, 2, '2023-10-02 20:00:00', '3D', 250, 300),
(3, 3, '2023-10-03 16:00:00', '2D', 200, 200);

-- 插入到 `member`
INSERT INTO member (email, name, password, birthday, phone) VALUES
('alice@example.com', '愛麗絲', 'alicepassword', '1985-04-12 00:00:00', '0912345678'),
('bob@example.com', '鮑勃', 'bobpassword', '1990-08-25 00:00:00', '0923456789'),
('carol@example.com', '卡洛兒', 'carolpassword', '1982-11-30 00:00:00', '0934567890');

-- 插入到 `staff`
INSERT INTO staff (cinema_id, name, account, password, role) VALUES
(1, '張經理', 'manager1', 'manager1password', '經理'),
(2, '李工作人員', 'staff1', 'staff1password', '工作人員'),
(3, '王售票員', 'staff2', 'staff2password', '售票員');

-- 插入到 `order`
-- 假設每個訂單購買一張電影票，座位隨機選擇。
INSERT INTO `order` (member_id, date, ticket_type, payment_method, paid) VALUES
(1, '2023-10-01 18:00:00', 0, 0, 0),
(2, '2023-10-02 19:30:00', 1, 1, 1),
(3, '2023-10-03 15:30:00', 1, 1, 1);
```

```

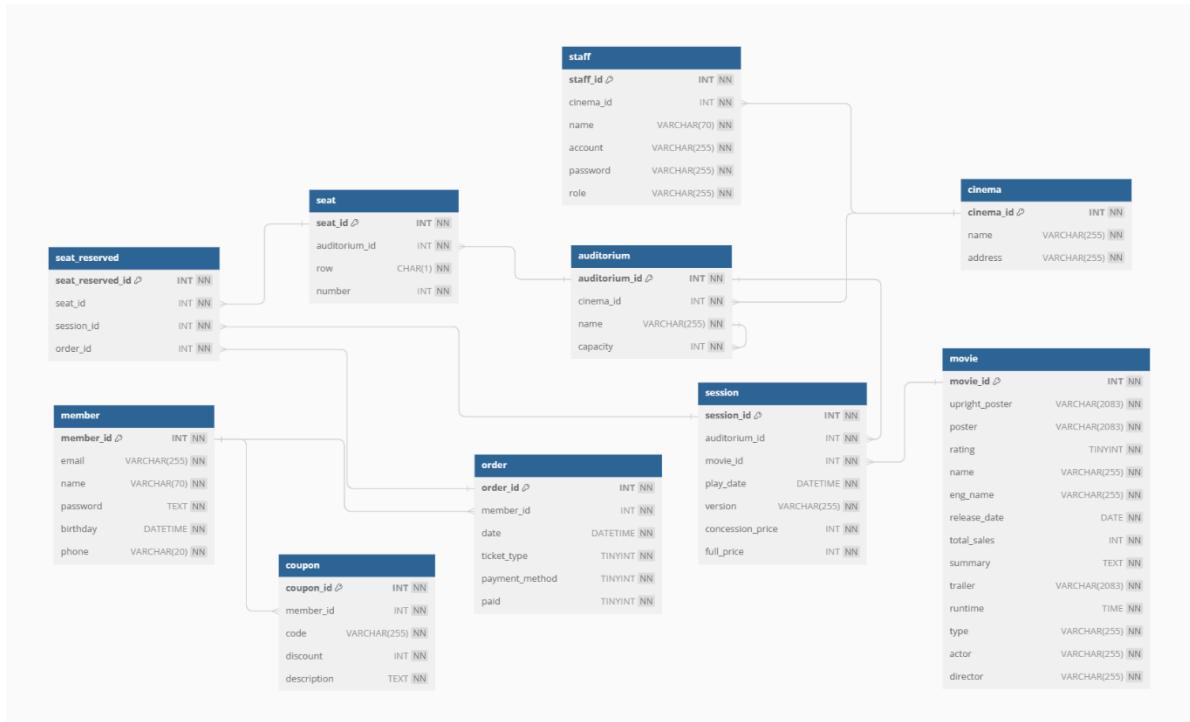
-- 插入到 `seat`
-- 假設每個劇院都有10排，每排10個座位。
INSERT INTO seat (auditorium_id, row, number) VALUES
(1, 'A', 1), (1, 'A', 2), (1, 'A', 3),
(2, 'B', 4), (2, 'B', 5), (2, 'B', 6),
(3, 'C', 7), (3, 'C', 8), (3, 'C', 9);

-- 插入到 `seat_reserved`
-- 假設每個訂單預訂一個座位。
INSERT INTO seat_reserved (seat_id, session_id, order_id) VALUES
(1, 1, 1), (2, 2, 2), (3, 3, 3);

-- 插入到 `coupon`
-- 使用符合節日的優惠券資料
INSERT INTO coupon (member_id, code, discount, description) VALUES
(1, 'XMAS2023', 30, '聖誕節優惠券'),
(2, 'NYE2023', 25, '跨年優惠券'),
(3, 'VACATION2023', 20, '暑假優惠券');

```

4.6 The implementation of tables in target DBMS

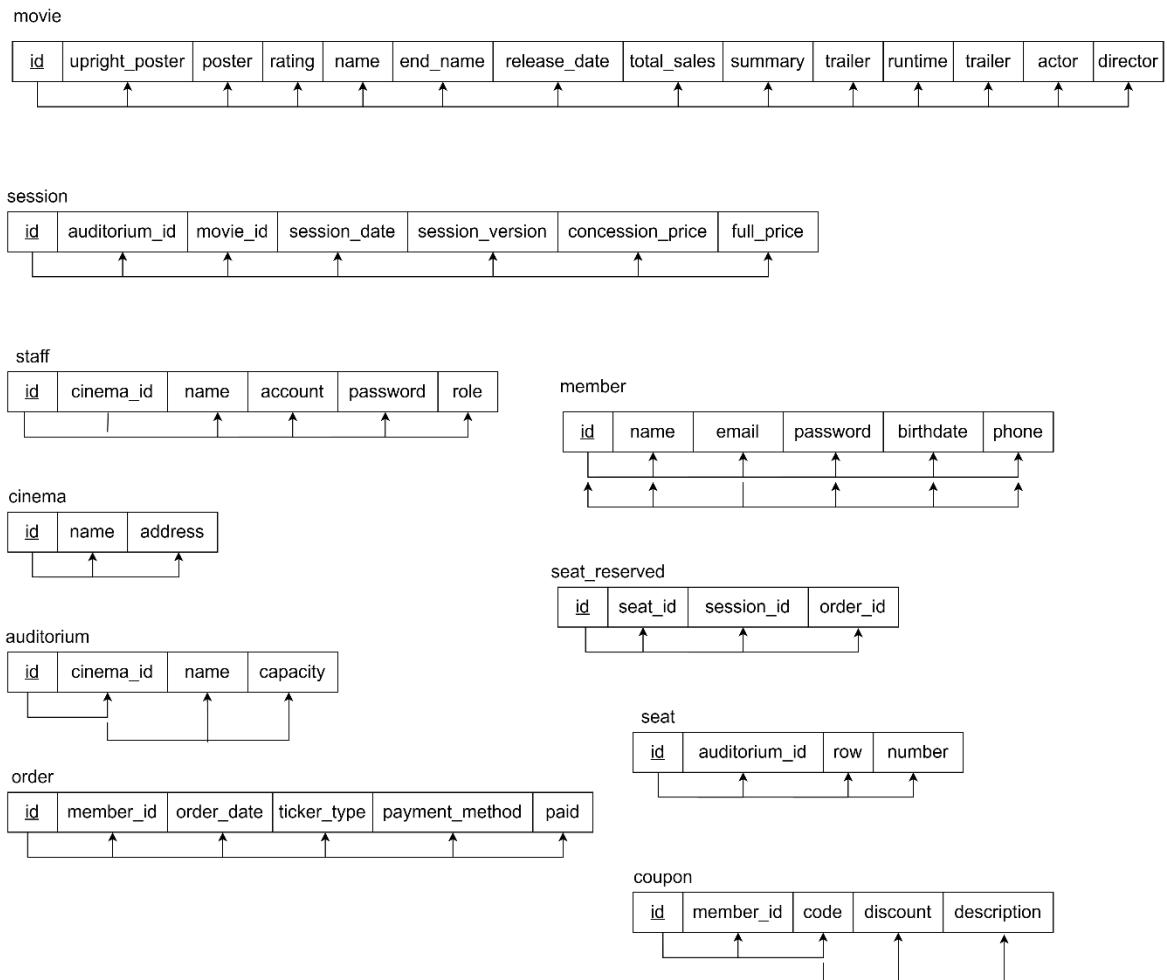


4.7 Estimation of the size of each table

Table	The length of table schema (Bytes)	The average number of tuples	Estimated size (Bytes)
cinema	520	20	10K
auditorium	264	30	8K
movie	810	1K	810K
session	300	500K	150M
member	65K	5K	325M
staff	270	50	13.5K
order	312	100K	31.2M
seat	268	20	5.4K
seat reserved	264	1K	264K
coupon	300	500K	150M

Section 5 Functional Dependencies and Database Normalization

5.1 Functional Dependencies



Section 6 The Use of the Database System

6.1 System Installation Description

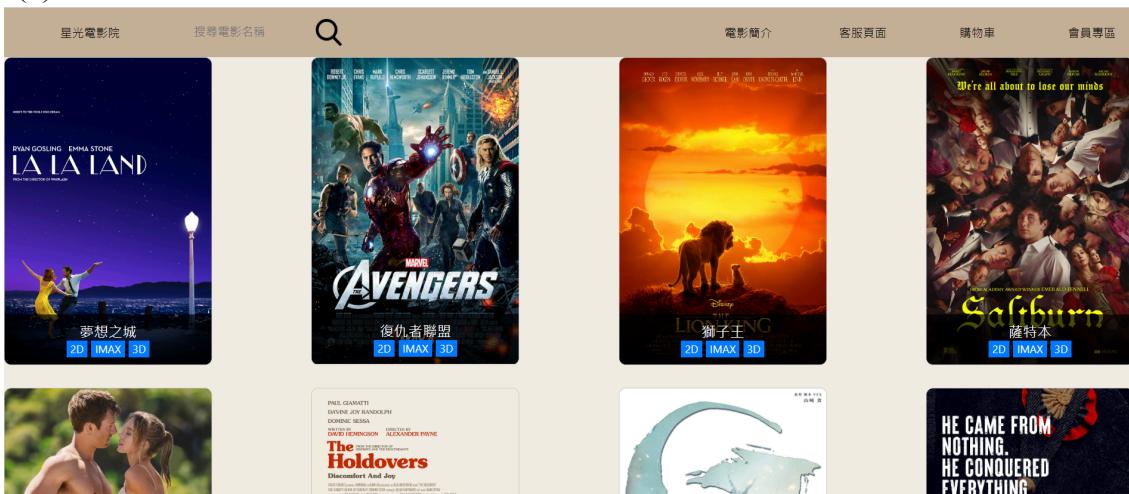
6.2 The Use of the System

6.2.1 使用者頁面

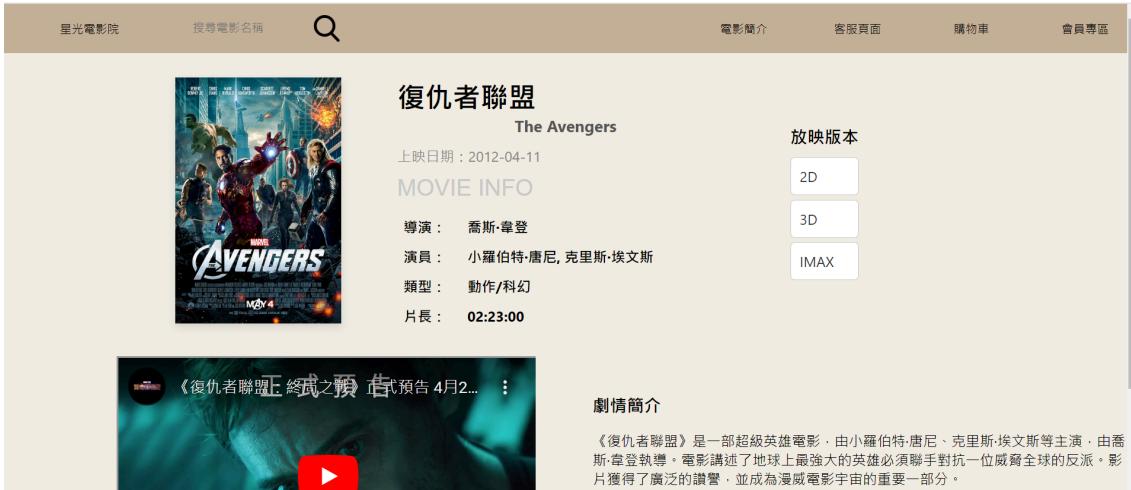
(a)電影訂票網站首頁



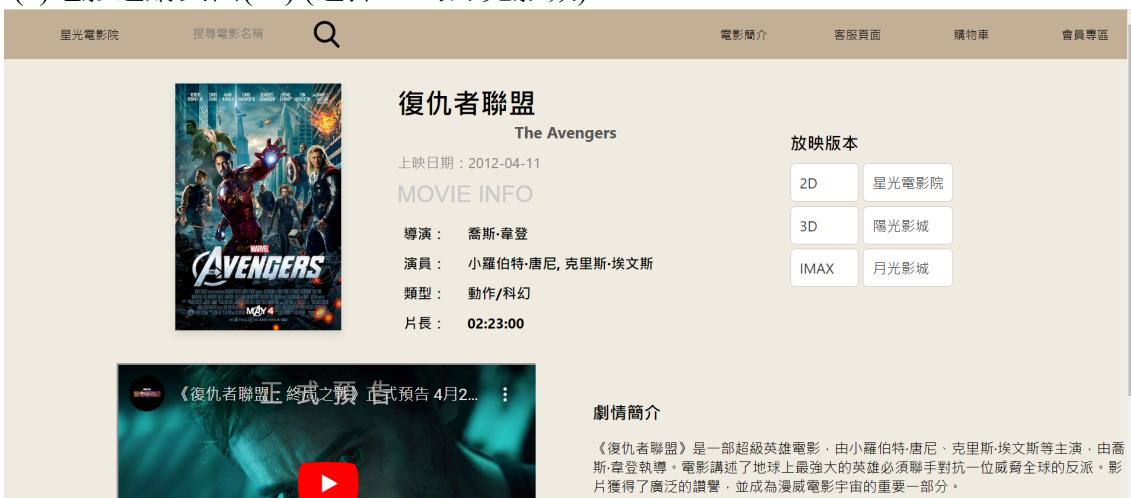
(b)電影簡介



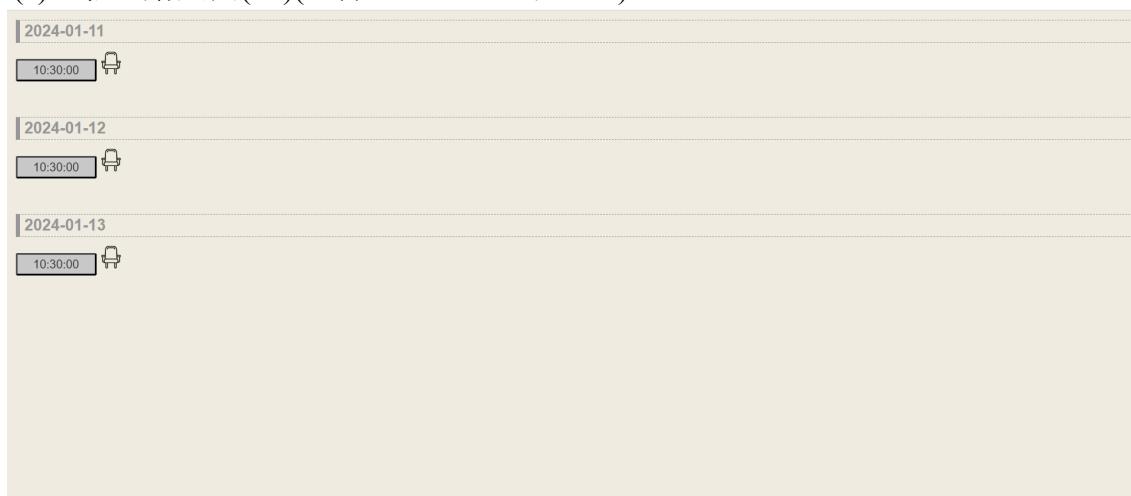
(c)電影詳細介紹及選購



(d) 電影選購頁面(一)(選擇2D的月光影城)



(e) 電影選購頁面(二)(選擇2024/01/11的10:30)



(f) 電影選購頁面(三)

星光電影院 搜尋電影名稱  電影簡介 客服頁面 購物車 會員專區

一般票種

票種	價格	數量	合計
全票	350	<input type="text" value="2"/>	700
優待票 <small>(優待票包括學生票、基督教、狂歡節、購買後未使用者，需於後場映照時出示相關優待證件，無證件者須補足全票金額。)</small>	250	<input type="text" value="0"/>	0
總計			700

[BACK](#) [Next](#)

(g) 電影選購頁面(四)(紅色為已售出, 黃色則為挑選座位)

星光電影院 搜尋電影名稱  電影簡介 客服頁面 購物車 會員專區



←   →

[BACK](#) [Next](#)

(h) 電影選購頁面(五之一)

星光電影院 搜尋電影名稱  電影簡介 客服頁面 購物車 會員專區

請注意！星光電影院絕不會要求顧客至 ATM操作解除分期或補繳金額，請貴賓提高警覺避免受騙。星光電影院與您一起防範詐騙；歡迎致電影院查詢更多反詐騙資訊。

[線上即時付款](#)

一般 / 銀行優惠 / 儲值金會員票

GENERAL / BANK PRIVILEGE TICKET / iShow SVC DISCOUNT TICKET

[影城臨櫃付款](#)

(i) 電影選購頁面(五之二)

星光電影院 搜尋電影名稱 電影簡介 客服頁面 購物車 會員專區

江志宇次

一、訂票及取票規定[一般票種]

1. 網路訂票每張票需加收NT\$20手續費
 2. 片長 150分鐘(含)以上之電影需加價NT\$20，每增加30分鐘需另加價NT\$10。
 3. 每筆訂票張數以 10 張為限，儲值金會員票每日訂票張數以10張為限。
 4. 銀行優惠票種與儲值金會員票無法於同筆訂單中，請分次訂購，唯兩筆訂票無法保證座位。
 5. 各銀行優惠票種之張數與折扣金額，請依各銀行信用卡優惠規則為準。銀行優惠票種預訂以電影播放日期為準，非刷卡日計算。每卡每日購買張數限制依影片版本與觀影日期相關規定限制。
 6. 請注意此交易金額將於購票步驟完成後，即刻於您的信用卡帳戶或儲值金帳戶中進行扣款。
 7. 購票完成後，請憑「訂票序號」至購影城進行取票，亦可至超商進行取票，但若交易內含餐飲品項，須至購影城臨櫃領取電影票與餐點。
 8. 消費者訂購電影票後，威秀影城仍保留影廳更換之權利，如有任何變更將於場次當日臨櫃告知，敬請見諒；其餘未盡事項依影城現場公告為主。

二、退換票規定[一般票種]

1. 線上購票後若已領取電影票，因故無法如期觀影，請於開演前 20 分鐘持未使用之電影票與原訂購之信用卡親至觀影影城辦理退換票。已取票或取餐者，無法進行線上退款。
 2. 線上購票後若尚未領取電影票，因故無法如期觀影，可於威秀影城官網登入會員訂票紀錄中進行線上退票，電影票最遲應於影片開演2小時前完成線上退票程序。
 3. 電影票遺失恕不補發，亦無法辦理退換票。
 4. 因天災或特殊事故取消電影放映時，請於購票場次7日內憑『未使用之原票券』至原購影票櫃台辦理退換票，影城未營業期間均不收取手續費。
 5. 退、換票注意事項：若是信用卡付款，則需攜帶購票之信用卡辦理退票（如未攜帶原購票信用卡，恕無法辦理退換票）。
 6. 其餘未盡事項依影城現場公告為主。

我已詳閱上述規定
[返回上頁](#) [前往訂單](#)

(j) 電影選購頁面(六)

星光電影院 搜尋電影名稱 電影簡介 客服頁面 購物車 會員專區

影城: 星光電影院
電影: 蒼鶯與少年
場次: 2024-01-11 - 10:30:00
票數: 2
票種: 一般票
座位: B5 B6
總計: 700

載具輸入
 載具輸入/郵寄電子發票/統編
 信用卡卡號、安全碼、到期日(月、年)
 信用卡安全碼
 信用卡到期日(月、年)

(k) 會員登入頁面

星光電影院 搜尋電影名稱 電影簡介 客服頁面 購物車 會員專區

Starlight Cinema Login Page

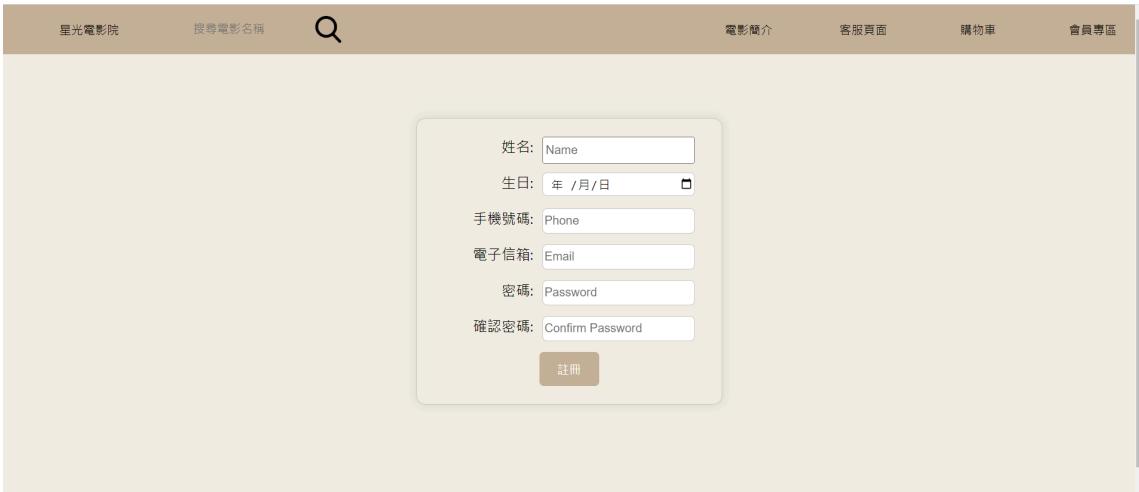
請輸入電子信箱

請輸入密碼

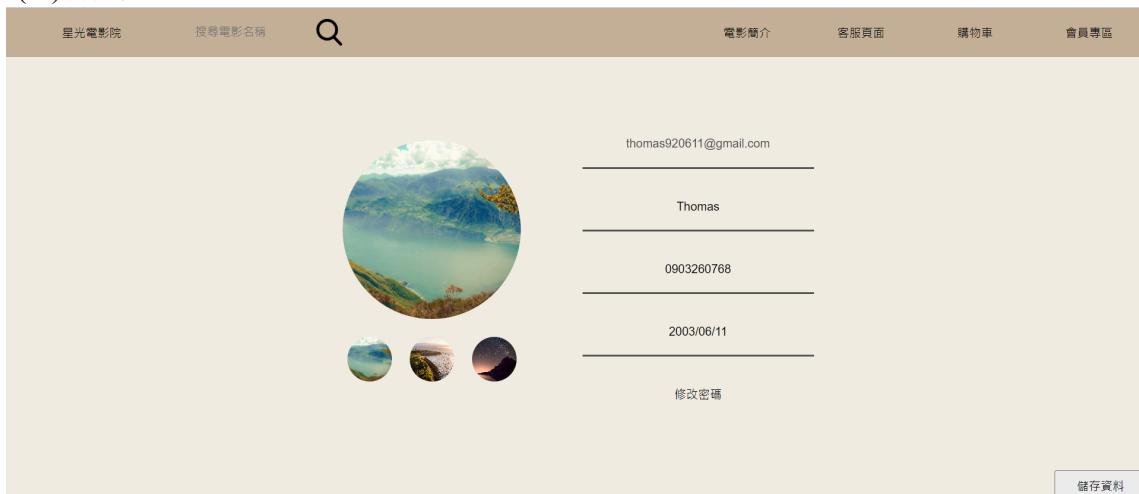
[Sign up](#) [Login](#)

[忘記密碼](#)

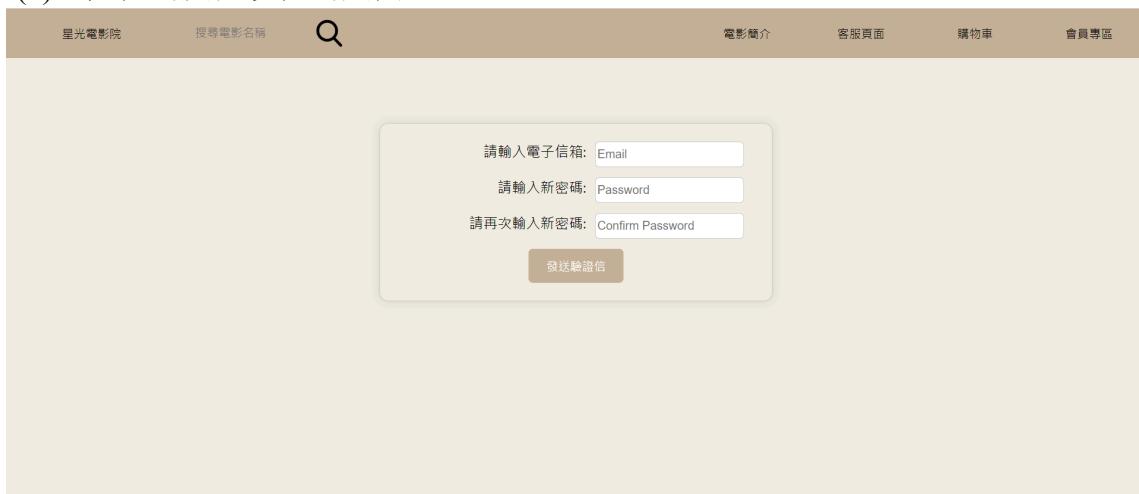
(l) 註冊會員畫面



(m)會員資料修改頁面



(n)忘記密碼及修改密碼頁面



(o)購物車頁面

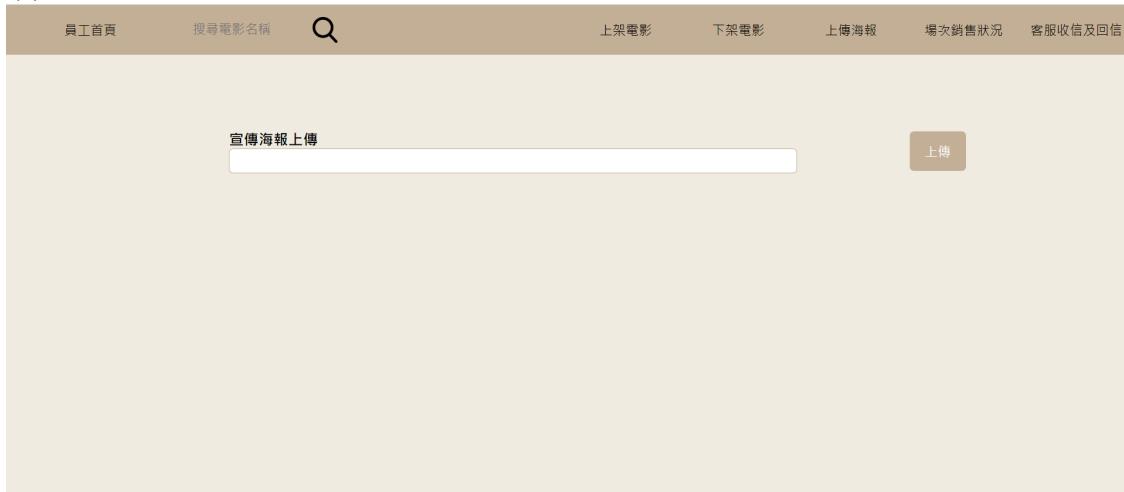
星光電影院		搜尋電影名稱		電影簡介	客服頁面	購物車	會員專區
Product	Price	Quantity		Remove	Choose		
Top Gun	\$350	2		Remove	Pay		
君たちはどう生きるか	\$280	1		Remove	Pay		

(p)客服頁面

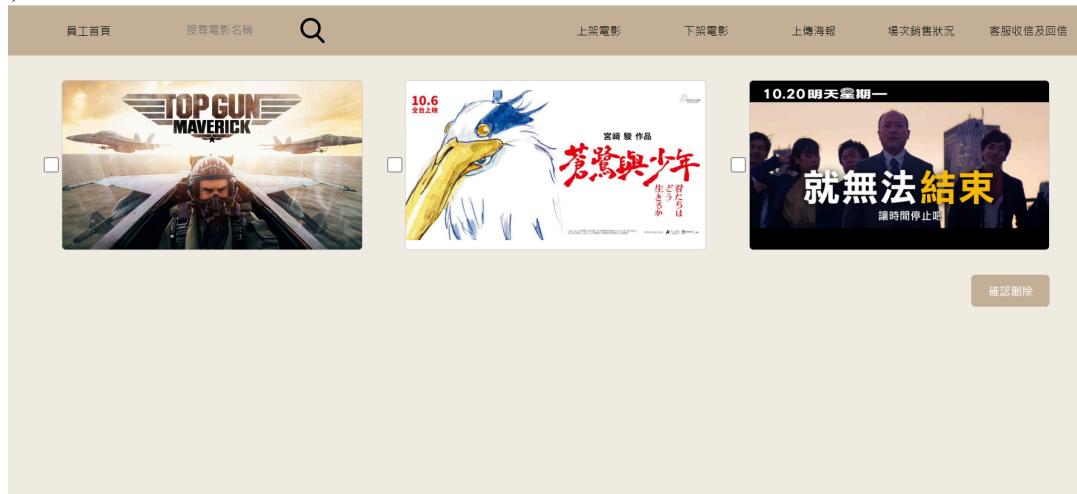
員工首頁	搜尋電影名稱		上架電影	下架電影	上傳海報	場次銷售狀況	客服收信及回信
<div style="border: 1px solid #ccc; padding: 10px;"> <p>姓名: <input type="text"/></p> <p>性別: <input type="radio"/> 男 <input type="radio"/> 女</p> <p>連絡電話: <input type="text"/></p> <p>Email: <input type="text"/></p> <p>建議之影城: <input type="text" value="cinema1"/></p> <p>建議內容: <input type="text"/></p> <p style="text-align: center;">送出您的意見</p> </div>							

6.2.2 員工使用頁面

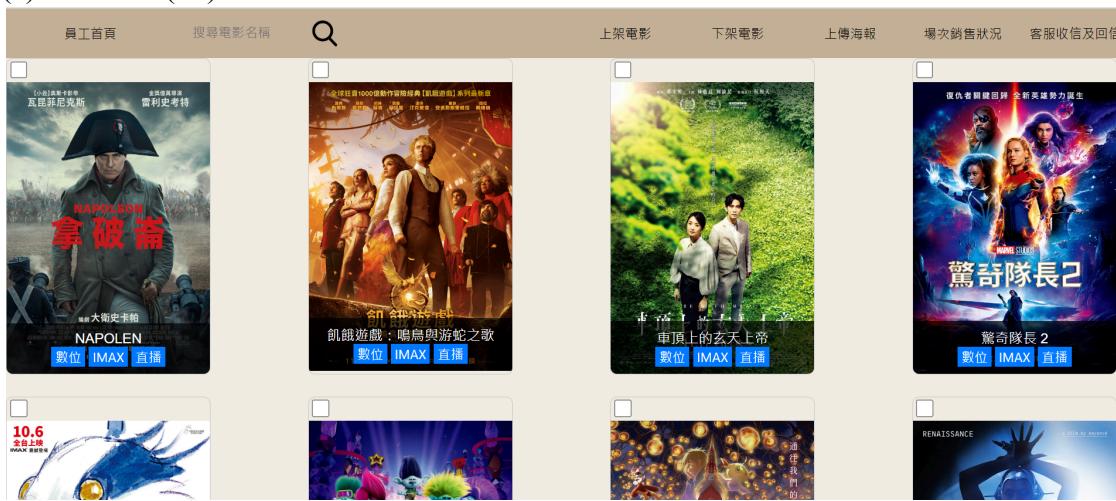
(a) 首頁宣傳海報上傳



(b) 首頁宣傳海報移除



(c) 下架電影(一)



(d) 下架電影(二)



(e) 上架電影(一)



(f) 上架電影(二)

一般票價

特殊票價

BACK NEXT

(g) 上架電影(三)

電影名稱
蒼鶻與少年

電影簡介
 《蒼鶻與少年》是一部2023年上映的日本動畫電影，由宮崎駿執導兼編劇，吉卜力工作室製作。電影原文片名是引用自小說家吉野源三郎在1937年發表的著書《你想活出怎樣的人生》，內容則為宮崎駿原創。劇情講述因太平洋戰爭喪母的少年牧真人與一隻會說話的蒼鶻相遇，展開不可思議的探索旅程。

一般票票價
100

特殊票票價
80

BACK FINISH

(h) 員工登入

Employee Login Page

Login

(i) 客服信件回覆(一)

員工首頁 搜尋電影名稱 上架電影 下架電影 上傳海報 場次銷售狀況 客服收信及回信

顧客反映信箱

影城分店

	信件標題	日期
台北信義威秀影城	建議1	2021/10/01
台北京站威秀影城	建議2	2021/10/02
板橋大遠百威秀影城	建議3	2021/10/03

上一頁 下一頁

(j) 客服信件回覆(二)

員工首頁 搜尋電影名稱 上架電影 下架電影 上傳海報 場次銷售狀況 客服收信及回信

顧客資訊

姓名	暉暉暉
性別	暉暉暉
連結電話	暉暉暉
email	暉暉暉
影城分店	暉暉暉

建議內容

以下省略

信件回覆

Section 7 Suggestions of Database Turning

在這次的專題實作中，我們發現我們的電影院購票系統的效能和細節仍有能夠改進的地方。以下是一些我們有做到的優化以及認為能做的改進：

系統優化：

1. **React**和**Flask**的系統讀寫分離：分開處理讀取和寫入操作，以提升系統效能。
2. 新增索引：針對大量資料的訂單、商店和產品等操作新增索引，以提高查詢效率。

效能提升：

1. **Redis**快取：導入**Redis**快取機制，優化I/O查詢效能，尤其在購物車操作方面。

其他考量：

1. 隊列排程：使用隊列排程（例如**RabbitMQ**），有效降低大量資料查詢的請求時間。
2. **WebSocket**推播：採用**WebSocket**推播訊息，改善訊息通知體驗，例如即時餐點進度通知。

資料庫設計及實作分析：

1. 新增屬性雖然暫時看不出效能影響，但若資料量增加，這類屬性將影響效能。
2. 尚未實作搜尋產品功能部分，可考慮增強。

這些改進方案和資料庫設計的分析有助於提升系統效能、改善使用者體驗，也為未來處理更多用戶和複雜操作需求鋪就了道路。

Section 8 Additional Queries and Views

Section 9 Conclusions and Future Work

9.1 Conclusions

透過這次的資料庫系統專題，我們獲得了許多寶貴的經驗。我們從了解資料庫的運作模式，到實際設計並實作了屬於我們的資料庫。在專題過程中，我們不斷思考並完善最初的設計，以確保最終的成品達到最高水準。

我們使用React和Flask來規劃和執行專題的任務，以提高效率。經過多次討論，我們成功地設計了星光電影院的購票系統，讓消費者能夠輕鬆使用。

設計整個系統是一項挑戰，我們以不同的角色身份思考和分析使用者在購票系統中的操作，以及管理者對電影院的管理。我們細緻地列出了介面需求、功能需求和非功能需求，並將系統功能和資料相互關聯，將系統分割成多個子系統，以確保每個子系統能夠有效地執行其負責的功能。

透過設計需求和實作，我們確保了整個流程的順暢性。雖然在實作過程中偶爾會發現設計上的瑕疵，需要進行微調，但這些調整讓我們更深入地了解設計的不足之處。幸運的是，整體而言，我們順利地完成了專題。

這次的資料庫系統專題使我們對未來團隊合作的工作模式有了初步的了解。我們體驗了前後端分離的工作方式，並成功地將各成員的工作有效地串連起來，形成完整的購票系統。這次專題讓我們不僅僅是在理論上學習，更實際地運用所學，將理論應用於實際的開發中。

9.2 Future Work

我們在製作專題時立下了一個目標，那就是創建一個功能完善且可實際應用的星光電影院購票系統。在多次討論後，我們參考了現有購票系統的介面、購買流程以及網站運作，並不斷調整和新增功能，從而使專題的複雜度與系統的完整性相應增加。為了在短時間內展示專題成果，我們從系統的主要功能入手，使用React搭建成前端界面，完成了Flask的後端設計，並整合了API。隨後，我們逐步實現了網站內的細節功能。我們也為未來的發展制定了計劃，希望讓星光電影院購票系統能夠呈現更完整的功能。我們計劃增加熱門影片的展示、記錄顧客最喜歡的電影、推出更多優惠活動，以及完善訂單狀態的管理等功能，以提供使用者更便利的服務，同時確保系統的穩定性，以應對更多的用戶需求。我們的目標不僅僅是完成這堂課的要求，更希望將系統實際應用於生活中，並與星光電影院建立合作關係，在校園中實際運行。

Glossary

React

React 是一個流行的 JavaScript 庫，用於構建用戶界面。它允許開發人員創建可重用的 UI 組件，並在底層數據更改時高效地更新和渲染它們。React 遵循基於組件的方法，使得管理複雜的 UI 和構建交互式 Web 應用程序更加容易。

Flask

Flask 是一個輕量級的 Python Web 框架。它提供了一種簡單靈活的方式來構建 Web 應用程序。Flask 遵循模型-視圖-控制器(MVC)模式，提供路由、模板渲染以及處理 HTTP 請求和響應的功能。它以其簡單性和易用性而聞名，是構建中小型 Web 應用程序的熱門選擇。

DBMS

資料庫管理系統(DBMS)是用於管理資料庫的軟體系統。它作為使用者和資料庫之間的接口，允許使用者發出處理資料庫的命令。DBMS 提供了創建、修改和查詢資料庫的功能，並控制資料訪問和確保資料完整性。

MariaDB

MariaDB 是一個開源的關聯式資料庫管理系統(RDBMS)。它是 MySQL 的一個分支，保持與 MySQL API 和命令的兼容性。MariaDB 提供了比 MySQL 更好的性能、可擴展性和安全性功能。它支持 ACID 符合的事務，並提供各種存儲引擎，適用於各種應用程序。

PostgreSQL

PostgreSQL是一個物件關聯式資料庫管理系統(ORDBMS)，具有傳統專業資料庫系統的特點，同時也包含了先進DBMS系統的強化功能。它是一個開源的、穩定可靠的資料庫解決方案。

Git

Git 是一個分散式版本控制系統，用於跟蹤軟體開發過程中源代碼的變化。它允許多個開發人員協作項目，提供分支、合併和版本歷史管理等功能。Git 以其速度、靈活性和對小型和大型項目的支持而廣泛使用。它也因其與 GitHub 和 GitLab 等平台的集成而受歡迎，這些平台為 Git 存儲庫提供了托管和協作功能。

References

Appendix

