

國立虎尾科技大學 資訊工程系

專題報告

# 飯店管理系統

參與成員：陳彥福 黃子峻

楊祐宇 羅文鍵

中華民國 年 月 日

# 目錄

一、摘要 .....	1
二、應用情境 .....	2
三、系統需求說明 .....	2
(一)住宿管理系統 .....	2
(二)餐廳管理系統 .....	2
四、資料庫建立 .....	3
(一)實體完整性 (Entity Integrity) .....	6
(二)參照完整性 (Referential Integrity) .....	7
(三)資料型態限制 (Domain Constraints) .....	8
(四)業務邏輯完整性 (Business Rules) .....	8
(五)領域完整性限制 (Domain Constraints) .....	9
1. Employee (員工): .....	9
2. Customer (顧客): .....	9
3. Room_Type (房型): .....	10
4. Room (房間): .....	10
5. Season (季節): .....	10
6. Room_Season_Rate (房間季節價格): .....	11
7. Booking (預訂): .....	11
8. Meal_Plan (餐點方案): .....	11
9. Menu_Item (菜單項目): .....	12
10. Restaurant (餐廳): .....	12
五、VIEW .....	13
(一)角色建立 .....	13
(二)權限設定範例 .....	13
1.顧客 (Customer View) .....	13
2.前台人員 .....	15
3.廚師 .....	18
4.後台管理員 .....	21
六、系統運作流程 .....	25
七、分工 .....	28
八、心得 .....	28

# 表目錄

表格 1 Employee 資料表 .....	3
表格 2 Customer 資料表 .....	3
表格 3 Room_Type 資料表 .....	3
表格 4 Room 資料表 .....	4
表格 5 Season 資料表 .....	4
表格 6 Room_Season_Rate 資料表 .....	4
表格 7 Meal_Plan 資料表 .....	4
表格 8 Menu_Item 資料表 .....	5
表格 9 Meal_Plan_Menu 資料表 .....	5
表格 10 Booking 資料表 .....	5
表格 11 Restaurant_Employee 資料表 .....	6
表格 12 Room_Cleaning 資料表 .....	6
表格 13 Restaurant 資料表 .....	6
表格 14 實體完整性表格 .....	6
表格 15 參照完整性表格 .....	7
表格 16 資料型態限制表格 .....	8
表格 17 業務邏輯完整性表格 .....	8
表格 18 Employee 完整性限制表格 .....	9
表格 19 Customer 完整性限制表格 .....	9
表格 20 Room_Type 完整性限制表格 .....	10
表格 21 Room 完整性限制表格 .....	10
表格 22 Season 完整性限制表格 .....	11
表格 23 Room_Season_Rate 完整性限制表格 .....	11
表格 24 Booking 完整性限制表格 .....	11
表格 25 Meal_Plan 完整性限制表格 .....	12
表格 26 Menu_Item 完整性限制表格 .....	12
表格 27 Restaurant 完整性限制表格 .....	12

## 一、摘要

此專題利用 MariaDB 建立一個完整的飯店管理系統資料庫，能夠協助飯店進行日常營運管理。系統可供管理者或員工使用，處理包含顧客資料、客房預訂、入住與退房、餐飲服務與帳單等功能，讓每一筆顧客與營運資料都能被妥善記錄與查詢。

在顧客資料管理部分，能紀錄顧客的聯絡資訊，例如電話、電子郵件等，並加入資料格式的限制，確保輸入內容正確。客房預訂功能則可讓使用者根據房型與日期建立訂單，並將顧客與房間關聯，入住與退房時系統也會即時更新房間狀態，讓管理者清楚知道哪些房間已被使用或尚可入住。

在餐飲與帳單功能方面，用戶可以紀錄每位顧客在住宿期間所使用的餐飲服務，例如早餐、飲料、房間送餐等，所有費用會自動加總至帳單中，系統會根據消費紀錄計算總金額，供退房時結帳使用。

此外，系統還能管理員工的基本資料與排班狀況，讓飯店可以更有效安排人力。同時透過查詢功能，也可以快速找出熱門房型、每日住房率、總收入等資訊，幫助經營者做出更有效的決策。

整體系統以資料表之間的主外鍵關聯為核心，確保資料一致性與完整性，並且具備良好的擴充性，未來可搭配前端介面或雲端平台做整合。這套系統不僅能提升飯店的營運效率，也讓我們在實作過程中學會如何從實際需求中設計出一個穩定且實用的資料庫架構。

## 二、應用情境

小羅是一間中小型飯店的前台工作人員，平常的工作內容包含處理顧客入住與退房、預約查詢、餐飲紀錄、以及帳務結算等。某天有客戶打電話預約下週末的住宿，小羅透過系統中的預約功能，快速輸入客戶資料與入住日期，系統自動比對可用房型並完成預訂。幾天後，這位客戶打來更改日期，小羅只需修改訂單紀錄，系統會即時更新房間使用狀態。

## 三、系統需求說明

### (一)住宿管理系統

#### 1.客房資料管理

- 建立與管理房型、樓層、價格、設備等資訊

#### 2.房況即時查詢

- 前台可查詢目前所有房間狀態（正常、待清潔、清潔中、維修中）

#### 3.線上預訂管理

- 提供前台或客戶預訂房間之功能，可選日期與房型

#### 4.入住/退房流程

- 記錄客人入住資料，辦理入住與退房作業，自動更新房況

#### 5.維修申請與追蹤

- 前台或房務可通報設備異常，維修人員可接單與回報完成

#### 6.報表輸出

- 產出入住率、房間使用統計、維修清潔紀錄報表（每日、每月）

### (二)餐廳管理系統

#### 1.菜單管理

- 可設定各餐別(早餐、午餐、晚餐、宴會)菜單內容與價格

#### 2.餐點訂單與出菜

- 廚房可根據訂單製作餐點並列出出餐明細

#### 3.餐飲報表

- 出餐統計、熱門菜色排行、食材使用與成本報表

## 四、資料庫建立

### 資料表欄位說明

表格 1 Employee 資料表

欄位名稱	資料型別	限制條件	說明
EmployeeID	INTEGER	PRIMARY KEY	員工 ID
Name	VAR-CHAR(255)	NOT NULL	員工名字
Position	VAR-CHAR(100)	NOT NULL	職位
Department	VAR-CHAR(100)	NOT NULL	部門
HireDate	DATE	NOT NULL	雇用日期
Phone	VARCHAR(50)	NOT NULL	手機號碼
IsActive	BOOLEAN	NOT NULL	是否在職

EmployeeID 是用來識別每一位員工的主鍵，當新增員工資料時會自動產生，系統透過這個 ID 來確認是誰處理了入住、退房或帳務作業，確保責任歸屬與紀錄清楚，其他欄位便是員工基本資料。

表格 2 Customer 資料表

欄位名稱	資料型別	限制條件	說明
CustomerID	INTEGER	PRIMARY KEY	顧客 ID
Name	CHAR(255)	NOT NULL	姓名
Phone	CHAR(50)	NOT NULL	電話
Email	CHAR(255)	UNIQUE	電子郵件

CustomerID 是顧客的唯一編號，系統透過這個 ID 來記錄每位顧客的訂房與消費紀錄，避免重複建立資料。其他為顧客基本資料，方便工作人員在有狀況時能即時通知到顧客。

表格 3 Room\_Type 資料表

欄位名稱	資料型別	限制條件	說明
RoomTypeID	INTEGER	PRIMARY KEY	床位類型名稱
Name	CHAR(100)	NOT NULL	姓名
BedCount	INTEGER	NOT NULL	床位數

BasePrice	DECIMAL(10, 2)	NULL CHECK (BasePrice >= 0)	床位價格
-----------	----------------	-----------------------------	------

Name 是房間實際名稱，由 BedCount、BasePrice 紀錄所需價格。

表格 4 Room 資料表

欄位名稱	資料型別	限制條件	說明
RoomID	INTEGER	PRIMARY KEY	房間 ID
RoomTypeID	INTEGER	NOT NULL	床位類型名稱
RoomNumber	CHAR(10)	NOT NULL	房間編號
RoomStatus	ENUM(4)	NULL DEFAULT '正常'	房間狀態

紀錄房間基本資訊，並可由人員及時修改房間狀態。

表格 5 Season 資料表

欄位名稱	資料型別	限制條件	說明
SeasonID	INTEGER	PRIMARY KEY	節日 ID
Name	VAR-CHAR(100)	NOT NULL	節日名字
StartDate	DATE	NOT NULL	開始日期
EndDate	DATE	NOT NULL	結束日期
PriceAdjustmentPercent	DECIMAL(5,2)	NOT NULL	價格調整描述

可由節日 ID 辨識並調整 PriceAdjustmentPercent 達到價格的更替。

表格 6 Room\_Season\_Rate 資料表

欄位名稱	資料型別	限制條件	說明
RoomTypeID	INTEGER	NOT NULL	床位類型名稱
SeasonID	INTEGER	NOT NULL	節日 ID
AdjustedPrice	DECIMAL(10,2)	NOT NULL CHECK (AdjustedPrice >= 0)	調整後價格

專門由季節調整倍率的部分。

表格 7 Meal\_Plan 資料表

欄位名稱	資料型別	限制條件	說明
MealPlanID	INTEGER	PRIMARY KEY	餐點方案 ID

Name	INTEGER	NOT NULL	名稱
ExtraCharge	DECIMAL(10,2)	CHECK (ExtraCharge >= 0)	額外費用

表格 8 Menu\_Item 資料表

欄位名稱	資料型別	限制條件	說明
MenuItemID	INTEGER	PRIMARY KEY	菜單項目 ID
RestaurantID	INTEGER	NOT NULL	餐廳 ID
Name	VARCHAR(100)	NOT NULL	餐點名稱
Category	VARCHAR(100)	NOT NULL	餐點種類
Price	DECIMAL(10,2)	NOT NULL CHECK (Price >= 0)	價格

表格 9 Meal\_Plan\_Menu 資料表

欄位名稱	資料型別	限制條件	說明
MealPlanMenuID	INTEGER	PRIMARY KEY	餐點方案菜單 ID
MealPlanID	INTEGER	NOT NULL	餐點方案 ID
MenuItemID	INTEGER	NOT NULL	菜單項目 ID

表格 10 Booking 資料表

欄位名稱	資料型別	限制條件	說明
BookingID	INTEGER	PRIMARY KEY	餐點方案菜單 ID
CustomerID	INTEGER	NOT NULL	餐點方案 ID
RoomID	INTEGER	NOT NULL	房間 ID
MealPlanID	INTEGER	NOT NULL	餐點方案 ID
EmployeeID	INTEGER	NOT NULL	員工 ID
CheckInDate	DATE	NOT NULL	入住時間
CheckOutDate	DATE	NOT NULL	退房時間
NumberOfGuests	INTEGER	NOT NULL CHECK (NumberOfGuests > 0)	入住人數
FinalPrice	DECIMAL(10,2)	NOT NULL CHECK (FinalPrice >= 0)	最終價格



表格 11 Restaurant\_Employee 資料表

欄位名稱	資料型別	限制條件	說明
EmployeeID	INTEGER	PRIMARY KEY	員工 ID
RestaurantID	INTEGER	PRIMARY KEY	餐廳 ID

表格 12 Room\_Cleaning 資料表

欄位名稱	資料型別	限制條件	說明
CleaningID	INTEGER	PRIMARY KEY	清潔 ID
EmployeeID	INTEGER	NOT NULL	員工 ID
RoomID	INTEGER	NOT NULL	房間 ID
CleaningDate	DATETIME	NOT NULL	清潔日期
Notes	TEXT	NULL	備註

表格 13 Restaurant 資料表

欄位名稱	資料型別	限制條件	說明
RestaurantID	INTEGER	PRIMARY KEY	餐廳 ID
Name	VARCHAR(255)	NOT NULL CHECK (TRIM(Name) <> "")	名稱
DayOfWeek	VARCHAR(50)	NOT NULL CHECK (TRIM(DayOfWeek) <> "")	星期日期
OpenTime	TIME	NOT NULL	開門時間
CloseTime	TIME	NOT NULL	關門時間

完整性限制:

### (一)實體完整性 (Entity Integrity)

主鍵不可為空值 (NULL)，並且其值在資料表中必須是唯一的。

表格 14 實體完整性表格

資料表 (Table)	主鍵欄位 (Primary Key Column(s))	說明
Customer	CustomerID	每位顧客都有唯一的識別編號
Room	RoomID	每個房間都有唯一的識別編號
Booking	BookingID	每筆訂房記錄都有唯一的識別編號
Meal_Plan	MealPlanID	每個餐飲方案都有唯一

		的識別碼
Season	SeasonID	每個季節定義都有唯一的識別碼
Restaurant	RestaurantID	每間餐廳都有唯一的識別碼
Menu_Item	MenuItemID	每個菜單項目都有唯一的識別編號
Employee	EmployeeID	每位員工都有唯一的識別編號
Room_Type	RoomTypeID	每種房型都有唯一的識別碼
Meal_Plan_Menu	MealPlanMenuID	餐飲方案與菜單項目的關聯記錄有唯一識別碼
Room_Season_Rate	RoomTypeID, SeasonID	複合主鍵，定義特定房型在特定季節的唯一價格調整

## (二)參照完整性 (Referential Integrity)

外鍵欄位的值必須參照到其主參考資料表中已存在的主鍵值，或者外鍵值可以為空值 (NULL)，除非該外鍵欄位被定義為不可為空。

表格 15 參照完整性表格

子資料表 (Child Table)	外鍵欄位 (Foreign Key Column(s))	參照主資料表 (Parent Table)	說明
Booking	CustomerID	Customer	每筆訂房記錄必須關聯到一位已存在的顧客
Booking	RoomID	Room	每筆訂房記錄必須關聯到一個已存在的房間
Booking	MealPlanID	Meal_Plan	訂房記錄可以選擇一個已存在的餐飲方案
Booking	EmployeeID	Employee	訂房記錄可以關聯到一位處理該訂單的員工
Room	RoomTypeID	Room_Type	每個房間必須歸屬於一個已存在的房型
Employee	RestaurantID	Restaurant	每位員工可以歸屬於一間已存在的餐廳
Menu_Item	RestaurantID	Restaurant	每個菜單項目必須歸屬於一間已存在的餐廳

Meal_Plan_Menu	MealPlanID	Meal_Plan	此關聯記錄中的餐飲方案必須是已存在的餐飲方案
Meal_Plan_Menu	MenuItemID	Menu_Item	此關聯記錄中包含的菜單項目必須是已存在的項目
Room_Season_Rate	RoomTypeID	Room_Type	價格調整所適用的房型必須是已存在的房型
Room_Season_Rate	SeasonID	Season	價格調整所適用的季節必須是已存在的季節定義

### (三)資料型態限制 (Domain Constraints)

表格 16 資料型態限制表格

欄位	資料型態	限制與說明
CustomerID, RoomID 等	int	整數，主鍵必須為正整數且唯一
Name, RoomType 等	varchar(n)	字元長度上限（例如 varchar(50)）
Phone	varchar(15)	最多 15 字元，建議符合電話格式
Email	varchar(100)	建議符合 Email 格式
價格欄位如 BasePrice、FinalPrice	decimal(10,2)	最大值 99999999.99，應為非負
PriceAdjustmentPercent	decimal(5,2)	例：20 表示 +20%
CheckInDate, CheckOutDate 等	date	必須符合日期格式 yyyy/mm/dd

### (四)業務邏輯完整性 (Business Rules)

表格 17 業務邏輯完整性表格

條件	限制說明
CheckOutDate > CheckInDate	退房日需晚於入住日
FinalPrice ≥ BasePrice	最終價格應包含基本價格與其他費用
ROOM.RoomStatus	限定值如：正常、待清潔、清潔中、維修中
SEASON.StartDate < EndDate	季節開始日需早於結束日
MEAL_PLAN_MENU 組合唯一	一道菜不能重複在同一方案中出現
ROOM_SEASON_RATE 組合唯一	同一房型與季節不能重複設定價格

### (五)領域完整性限制 (Domain Constraints)

#### 1. Employee (員工):

表格 18 Employee 完整性限制表格

欄位	資料型態	舉例與說明 (包含 CHECK 限制)
Name (姓名)	文字類型。	例子： "楊祐宇", "Yang-You-Yu" CHECK (TRIM(Name) <> " AND Name NOT REGEXP '[0-9]')
Position (職位)	文字類型。	例子： "經理", "櫃檯人員" CHECK (TRIM(Position) <> ")
Department (部門)	文字類型。	例子： "客房部", "餐飲部" CHECK (TRIM(Department) <> ")
HireDate (雇用日期)	日期類型。	例子： "2023-08-15" CHECK (HireDate <= CURDATE())
Phone (電話)	文字或數字類型，限制格式。	例子： "0912345678" CHECK (Phone REGEXP '^09[0-9]{8}\$' OR Phone REGEXP '^(\+8869
IsActive (是否在職)	布林值 (True/False) 或類似的表示方式 (例如 0/1)。	例子： True (表示在職) (此限制由 BOOLEAN 型別強制)

#### 2. Customer (顧客):

表格 19 Customer 完整性限制表格

欄位	資料型態	舉例與說明 (包含 CHECK 限制)
Name (姓名)	文字類型。	例子： "羅文鍵", "wenchian Luo" CHECK (TRIM(Name) <> " AND

		Name NOT REGEXP '[0-9]')
Phone (電話)	文字或數字類型，限制格式。	例子： "0987654321" CHECK (Phone REGEXP '^09[0-9]{8}\$' OR Phone REGEXP '^(\+8869
Email (電子郵件)	文字類型，應符合電子郵件格式(含@符號)。	例子： "test@example.com" CHECK (Email REGEXP '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\$')

### 3. Room\_Type (房型):

表格 20 Room\_Type 完整性限制表格

欄位	資料型態	舉例與說明 (包含 CHECK 限制)
Name (名稱)	文字類型。	例子： "單床雙人房" CHECK (TRIM(Name) <> ")
BedCount (床位數)	數字類型 (整數)。	例子： 1, 2 CHECK (BedCount > 0)
BasePrice (基本價格)	數字類型 (貨幣)。	例子： 2000.00 CHECK (BasePrice >= 0)

### 4. Room (房間):

表格 21 Room 完整性限制表格

欄位	資料型態	舉例與說明 (包含 CHECK 限制)
Name (名稱)	文字類型。	例子： "單床雙人房" CHECK (TRIM(Name) <> ")
BedCount (床位數)	數字類型 (整數)。	例子： 1, 2 CHECK (BedCount > 0)

### 5. Season (季節):

表格 22 Season 完整性限制表格

欄位	資料型態	舉例與說明 (包含 CHECK 限制)
Name (名稱)	文字類型。	例子： "單床雙人房" CHECK (TRIM(Name) <> ")
BedCount (床位數)	數字類型 (整數)。	例子： 1, 2 CHECK (BedCount > 0)
BasePrice (基本價格)	數字類型 (貨幣)。	例子： 2000.00 CHECK (BasePrice >= 0)
欄位	資料型態	舉例與說明 (包含 CHECK 限制)

6. Room\_Season\_Rate (房間季節價格):

表格 23 Room\_Season\_Rate 完整性限制表格

欄位	資料型態	舉例與說明 (包含 CHECK 限制)
AdjustedPrice (調整後價格)	數字類型 (貨幣)。	例子： 3000.00 CHECK (AdjustedPrice >= 0)

7. Booking (預訂):

表格 24 Booking 完整性限制表格

欄位	資料型態	舉例與說明 (包含 CHECK 限制)
CheckInDate (入住日期)	日期類型。	例子： "2024-10-01" (此限制由 DATE 型別強制日期格式)
CheckOutDate (退房日期)	日期類型，應晚於 CheckInDate。	例子： "2024-10-05" CHECK (CheckOutDate > CheckInDate)
NumberOfGuests (入住人數)	數字類型 (整數)。	例子： 1, 2, 3 CHECK (NumberOfGuests > 0)
FinalPrice (最終價格)	數字類型 (貨幣)。	例子： 12000.00 CHECK (FinalPrice >= 0)

8. Meal\_Plan (餐點方案):

表格 25 Meal\_Plan 完整性限制表格

欄位	資料型態	舉例與說明 (包含 CHECK 限制)
Name (名稱)	文字類型。	例子： "含早餐", "僅住宿" CHECK (TRIM(Name) <> "")
ExtraCharge (額外費用)	數字類型 (貨幣)。	例子： 300.00, 0.00 CHECK (ExtraCharge >= 0)

9. Menu\_Item (菜單項目):

表格 26 Menu\_Item 完整性限制表格

欄位	資料型態	舉例與說明 (包含 CHECK 限制)
Name (名稱)	文字類型。	例子： "牛排", "凱撒沙拉" CHECK (TRIM(Name) <> "")
Category (類別)	文字類型 (例如 '主食', '飲品')，應有一組預定義的值。	例子： "主菜", "開胃菜" CHECK (TRIM(Category) <> "")
Price (價格)	數字類型 (貨幣)。	例子： 850.00 CHECK (Price >= 0)

10. Restaurant (餐廳)

表格 27 Restaurant 完整性限制表格

欄位	資料型態	舉例與說明 (包含 CHECK 限制)
Name (名稱)	文字類型。	例子： "牛排", "凱撒沙拉" CHECK (TRIM(Name) <> "")
Category (類別)	文字類型 (例如 '主食', '飲品')，應有一組預定義的值。	例子： "主菜", "開胃菜" CHECK (TRIM(Category) <> "")
Price (價格)	數字類型 (貨幣)。	例子： 850.00 CHECK (Price >= 0)
欄位	資料型態	舉例與說明 (包含 CHECK 限制)

## 五、VIEW

### 使用者權限管理

為了保護資料安全並確保不同使用者只能存取其職責範圍內的資訊，我們可以透過資料庫的角色和權限管理機制來實現。

#### (一)角色建立

針對不同角色，建立不同的 rule 規則。

```
-- 建立角色
CREATE ROLE customer_role;
CREATE ROLE front_desk_role;
CREATE ROLE admin_role;
CREATE ROLE chef_role;

-- 建立資料庫使用者
CREATE USER 'customer_user'@'localhost';
CREATE USER 'front_desk_user'@'localhost';
CREATE USER 'admin_user'@'localhost';
CREATE USER 'chef_user'@'localhost';

-- 將角色賦予給使用者
GRANT customer_role TO 'customer_user'@'localhost';
GRANT front_desk_role TO 'front_desk_user'@'localhost';
GRANT admin_role TO 'admin_user'@'localhost';
GRANT chef_role TO 'chef_user'@'localhost';
```

#### (二)權限設定範例

##### 1.顧客 (Customer View)

➤ 權限設定:

- Customer: SELECT (自己的資料), UPDATE (自己的資料).
- Room\_Type, Room (例如房號、狀態，但不含內部管理資訊),  
Season, Room\_Season\_Rate, Meal\_Plan: SELECT.
- Booking: SELECT (自己的訂單), INSERT (新訂單), UPDATE (修改



自己的訂單，如取消).

➤ **SQL 範例:**

```
-- 授予顧客查詢房型資訊的權限
GRANT SELECT ON Room_Type TO customer_role;
GRANT SELECT (RoomNumber, RoomStatus, RoomTypeID) ON Room TO customer_role; -- 限制可見欄位
GRANT SELECT ON Season TO customer_role;
GRANT SELECT ON Room_Season_Rate TO customer_role;
GRANT SELECT ON Meal_Plan TO customer_role;

-- 授予查詢及管理自己訂單的權限
GRANT SELECT, INSERT ON Booking TO customer_role;
-- 授予查詢及修改自己基本資料的權限
GRANT SELECT, UPDATE (Name, Phone, Email) ON Customer TO customer_role;
```

➤ **View: 建立一個只顯示顧客自己訂單的 view。**

- a. **顯示顧客資料 (Customer Profile):** 允許顧客查看自己的基本資料。更新資料的操作則透過對 Customer 表的 UPDATE 權限進行。

```
CREATE VIEW CustomerProfileView AS
SELECT
    CustomerID,
    Name,
    Phone,
    Email
FROM
    Customer
WHERE
    CustomerID = CURRENT_USER_CUSTOMER_ID(); -- 假設 CURRENT_USER_CUSTOMER_ID() 是一個能獲取當前登入顧客ID的函數
```

```
GRANT SELECT ON CustomerProfileView TO customer_role;
-- 顧客更新資料的權限已在前面 GRANT SELECT, UPDATE (Name, Phone, Email) ON Customer TO customer_role; 中授予
```

- b. **顯示房型與房間狀態 (Room Information):** 讓顧客可以查看不同房型的資訊以及房間的即時狀態。

```
CREATE VIEW RoomAvailabilityView AS
SELECT
    rt.Name AS RoomTypeName,
    rt.BedCount,
    rt.BasePrice,
    r.RoomNumber,
    r.RoomStatus
FROM
    Room r
    JOIN Room_Type rt ON r.RoomTypeID = rt.RoomTypeID;

GRANT SELECT ON RoomAvailabilityView TO customer_role;
```

- c. **顯示季節與房價 (Seasonal Pricing):** 顧客可以查詢不同季節下各

房型的價格。

```
CREATE VIEW RoomSeasonalPricingView AS
SELECT
    rt.Name AS RoomTypeName,
    s.Name AS SeasonName,
    s.StartDate AS SeasonStartDate,
    s.EndDate AS SeasonEndDate,
    COALESCE(rsr.AdjustedPrice, rt.BasePrice * (1 + s.PriceAdjustmentPercent / 100)) AS FinalPricePerNight
FROM
    Room_Type rt
    CROSS JOIN Season s -- 顯示所有房型與季節的組合
    LEFT JOIN Room_Season_Rate rsr ON rt.RoomTypeID = rsr.RoomTypeID AND s.SeasonID = rsr.SeasonID;

GRANT SELECT ON RoomSeasonalPricingView TO customer_role;
```

- d. **顯示餐飲類別 (Meal Plan Options):** 顧客可以查看可選的餐飲計劃及其價格。

```
CREATE VIEW MealPlanOptionsView AS
SELECT
    MealPlanID,
    Name,
    ExtraCharge
FROM
    Meal_Plan;

GRANT SELECT ON MealPlanOptionsView TO customer_role;
```

- e. **顯示顧客自己的訂單 (Customer Bookings):** 建立一個只顯示顧客自己訂單的 view。

```
CREATE VIEW CustomerSelfBookings AS
SELECT
    b.BookingID,
    rt.Name AS RoomTypeName,
    r.RoomNumber,
    mp.Name AS MealPlanName,
    b.CheckInDate,
    b.CheckOutDate,
    b.NumberOfGuests,
    b.FinalPrice
FROM
    Booking b
    JOIN Room r ON b.RoomID = r.RoomID
    JOIN Room_Type rt ON r.RoomTypeID = rt.RoomTypeID
    JOIN Meal_Plan mp ON b.MealPlanID = mp.MealPlanID
WHERE
    b.CustomerID = CURRENT_USER_CUSTOMER_ID(); -- 假設 CURRENT_USER_CUSTOMER_ID() 是一個能獲取當前登入顧客ID的函數

GRANT SELECT ON CustomerSelfBookings TO customer_role;
```

## 2. 前台人員

前台人員需要管理顧客資料、訂房、房況等。

- **權限設定:**

Customer: SELECT, INSERT, UPDATE.

Room\_Type, Season, Room\_Season\_Rate, Meal\_Plan: SELECT.

Room: SELECT, UPDATE (例如 RoomStatus).

Booking: SELECT (所有), INSERT, UPDATE.

Room\_Cleaning: SELECT, INSERT, UPDATE.

Employee: SELECT (部分欄位，例如姓名、職位，排除敏感資訊).

➤ **SQL 範例：**

```
GRANT SELECT, INSERT, UPDATE ON Customer TO front_desk_role;
GRANT SELECT ON Room_Type TO front_desk_role;
GRANT SELECT, UPDATE (RoomStatus) ON Room TO front_desk_role;
GRANT SELECT, INSERT, UPDATE ON Booking TO front_desk_role;
GRANT SELECT, INSERT, UPDATE ON Room_Cleaning TO front_desk_role;
GRANT SELECT (EmployeeID, Name, Position, Department, Phone) ON Employee TO front_desk_role;
GRANT SELECT ON Meal_Plan TO front_desk_role;
GRANT SELECT ON Menu_Item TO front_desk_role;
GRANT SELECT ON Meal_Plan_Menu TO front_desk_role;
GRANT SELECT ON Restaurant TO front_desk_role;
GRANT SELECT ON Season TO front_desk_role; -- 前台人員查詢價格時可能需要
GRANT SELECT ON Room_Season_Rate TO front_desk_role; -- 前台人員查詢價格時可能需要
```

➤ **View:**

為了支援前台人員的日常工作流程，並對應提供的流程圖，可以建立或利用以下 Views：

a. **顧客資料管理 (Manage Customer Data)**

**查詢/查看顧客資料：** 為方便前台人員查詢顧客，可建立以下

View：

```
CREATE VIEW AllCustomersView AS
SELECT CustomerID, Name, Phone, Email
FROM Customer;
```

```
GRANT SELECT ON AllCustomersView TO front_desk_role;
-- 前台人員可使用此 View 搜尋顧客。根據「管理顧客資料」流程圖，查詢後可選擇查看或更新。
-- 查看顧客詳細資料是透過對 Customer 表的 SELECT 權限。
```

**新增顧客資料：** 根據實際需求，前台人員通常也需要新增顧客。

此操作透過對 Customer 表的 INSERT 權限直接執行。

b. 訂房管理 (Manage Bookings)

查詢/查看訂房：為方便前台人員查詢所有訂房記錄，可建立以下

View：

```
CREATE VIEW AllBookingsView AS
SELECT
    b.BookingID,
    c.Name AS CustomerName,
    c.Phone AS CustomerPhone,
    rt.Name AS RoomTypeName,
    r.RoomNumber,
    mp.Name AS MealPlanName,
    b.CheckInDate,
    b.CheckOutDate,
    b.NumberOfGuests,
    b.FinalPrice,
    e.Name AS BookingEmployeeName
FROM
    Booking b
    JOIN Customer c ON b.CustomerID = c.CustomerID
    JOIN Room r ON b.RoomID = r.RoomID
    JOIN Room_Type rt ON r.RoomTypeID = rt.RoomTypeID
    JOIN Meal_Plan mp ON b.MealPlanID = mp.MealPlanID
    JOIN Employee e ON b.EmployeeID = e.EmployeeID;
```

```
GRANT SELECT ON AllBookingsView TO front_desk_role;
```

對於每日活動（如今日入住/退房），可使用已有的 DailyActivitySummary View：

```
CREATE VIEW DailyActivitySummary AS
SELECT B.BookingID, C.Name AS CustomerName, R.RoomNumber, B.CheckInDate, B.CheckOutDate, R.RoomStatus
FROM Booking B
    JOIN Customer C ON B.CustomerID = C.CustomerID
    JOIN Room R ON B.RoomID = R.RoomID
WHERE B.CheckInDate = CURDATE() OR B.CheckOutDate = CURDATE();
```

```
GRANT SELECT ON DailyActivitySummary TO front_desk_role;
-- 根據「管理訂房」流程圖，若無新增需求，則進行查詢，之後可選擇查看或更新。
```

新增訂房：如流程圖所示，若有新的訂房需求，則透過

對 Booking 表的 INSERT 權限直接操作。「確認訂房」為後續應用程式邏輯。

**更新訂房:** 如流程圖所示，查詢到訂房記錄後，若需更新，則透過對 Booking 表的 UPDATE 權限直接操作。

### c. 房況管理 (Manage Room Status)

**查看房況:** 為方便前台人員查看所有房間的即時狀態，可建立以下 View：

```
CREATE VIEW FullRoomStatusView AS
SELECT
    r.RoomNumber,
    rt.Name AS RoomTypeName,
    r.RoomStatus,
    rt.BedCount
FROM
    Room r
    JOIN Room_Type rt ON r.RoomTypeID = rt.RoomTypeID
ORDER BY
    r.RoomNumber;
```

```
GRANT SELECT ON FullRoomStatusView TO front_desk_role;
-- 根據「管理房況」流程圖，前台人員可選擇查看房況。
```

**更新房況:** 如流程圖所示，若需要更新房況，則透過對 Room 表的 UPDATE (RoomStatus) 權限直接操作

## 3.廚師

廚師主要需要查看菜單、餐點方案以及當日或近期的餐點需求。

### ➤ 權限設定:

Menu\_Item: SELECT. (若允許廚師標記食材狀態或暫時下架，可考慮 UPDATE 特定欄位)

Meal\_Plan: SELECT.

Meal\_Plan\_Menu: SELECT.

Booking: SELECT (例如 MealPlanID, CheckInDate, CheckOutDate 及相關顧客數量，以預估備餐量).

Restaurant: SELECT (查看其所屬餐廳的資訊).

➤ **SQL 範例:**

```
GRANT SELECT ON Menu_Item TO chef_role;
GRANT SELECT ON Meal_Plan TO chef_role;
GRANT SELECT ON Meal_Plan_Menu TO chef_role;
-- 授予廚師查看訂單中餐飲相關資訊的權限
GRANT SELECT (BookingID, CustomerID, RoomID, MealPlanID, CheckInDate, CheckOutDate) ON Booking TO chef_role;
GRANT SELECT ON Restaurant TO chef_role;
```

➤ **View:**

為了支援廚師的日常工作流程，可以建立以下 Views：

a. **查看菜單 (ChefMenuView)**

顯示所有菜單項目及其詳細資訊（名稱、類別、價格）。

View:

```
CREATE VIEW ChefMenuView AS
SELECT
    mi.MenuItemID,
    mi.Name,
    mi.Category,
    mi.Price,
    r.Name AS RestaurantName
FROM
    Menu_Item mi
    JOIN Restaurant r ON mi.RestaurantID = r.RestaurantID;

GRANT SELECT ON ChefMenuView TO chef_role;
```

b. **查看餐點方案 (ChefMealPlanView)**

顯示餐點方案及其包含的菜單項目。

View:

```
CREATE VIEW ChefMealPlanView AS
SELECT
    mp.Name AS MealPlanName,
    mp.ExtraCharge,
    mi.Name AS MenuItemName,
    mi.Category AS MenuItemCategory,
    mi.Price AS MenuItemPrice
FROM
    Meal_Plan mp
    JOIN Meal_Plan_Menu mpm ON mp.MealPlanID = mpm.MealPlanID
    JOIN Menu_Item mi ON mpm.MenuItemID = mi.MenuItemID;

GRANT SELECT ON ChefMealPlanView TO chef_role;
```

c. 查看餐點方案查看當日及近期餐點需求

(UpcomingMealRequirementsView)

顯示未來一段時間內各餐點方案的預訂數量及預估用餐人數。

View:

```
CREATE VIEW UpcomingMealRequirementsView AS
SELECT
    B.CheckInDate,
    MP.Name AS MealPlanName,
    COUNT(B.BookingID) AS NumberOfBookings,
    SUM(B.NumberOfGuests) AS EstimatedGuests
FROM Booking B
    JOIN Meal_Plan MP ON B.MealPlanID = MP.MealPlanID
WHERE B.CheckInDate >= CURDATE() AND B.CheckInDate <= DATE_ADD(CURDATE(), INTERVAL 7 DAY) -- 例如未來7天
GROUP BY B.CheckInDate, MP.Name
ORDER BY B.CheckInDate, MP.Name;

GRANT SELECT ON UpcomingMealRequirementsView TO chef_role;
```

d. 查看品項類別摘要 (MenuItemCategorySummaryView)

顯示目前所有品項類別及其包含的菜單項目數量，輔助「查看現有品項類別」及「查看品項類別的統計資訊」。

View:

```
CREATE VIEW MenuItemCategorySummaryView AS
SELECT
    Category,
    COUNT(MenuItemID) AS NumberOfItems,
    GROUP_CONCAT(Name ORDER BY Name SEPARATOR ', ') AS ItemsInCategory -- 列出該類別下的部分品項
FROM
    Menu_Item
GROUP BY
    Category
ORDER BY
    Category;

GRANT SELECT ON MenuItemCategorySummaryView TO chef_role;
```

➤ **管理品項類別操作說明：**

根據流程圖，廚師還需要管理品項類別。這部分操作主要透過

對 Menu\_Item 表的 INSERT, UPDATE, DELETE 權限來實現

- **查看現有品項類別 及 統計資訊：**可透過 MenuItemCategorySummaryView 達成。
- **新增品項類別：**當廚師新增一個 Menu\_Item 時，可以在 Category 欄位輸入新的類別名稱，從而建立新的品項類別。
- **編輯品項類別：**廚師可以修改現有 Menu\_Item 的 Category 欄位。若要將一個類別名稱（例如 "點心"）統一修改為另一個名稱（例如 "甜點"），則需要更新所有相關 Menu\_Item 的 Category。

```
-- 範例：將所有 '點心' 類別的菜品改為 '甜點' 類別（此操作由具有 UPDATE 權限的廚師執行）
-- UPDATE Menu_Item SET Category = '甜點' WHERE Category = '點心';
```

- **刪除品項類別：**如果一個品項類別下的所有 Menu\_Item 都被刪除，或者它們的類別都被修改為其他類別，則該品項類別在實質上即被移除。廚師可透過刪除相關 Menu\_Item 或修改其 Category 來達成。

#### 4.後台管理員



後台管理員通常擁有最高權限，可以管理所有資料和系統設定。

➤ **權限設定:**

對所有資料表擁有 SELECT, INSERT, UPDATE, DELETE 權限。

管理員工資料 (Employee)、房型價格 (Room\_Type.BasePrice, Room\_Season\_Rate)、季節設定 (Season)、餐廳與菜單 (Restaurant, Menu\_Item) 等。

➤ **SQL 範例:**

```
GRANT ALL PRIVILEGES ON hotel.* TO admin_role;
```

➤ **操作說明與輔助 Views (Operation Description and Helper Views):**

後台管理員憑藉 ALL PRIVILEGES 權限，可直接對資料庫進行全面的管理。以下根據流程圖說明主要管理功能，並提供輔助查詢的 View 範例：

a. **管理員工資料 (Manage Employee Data)**

- 管理員可直接對 Employee 表進行查詢 (SELECT)、新增 (INSERT)、修改 (UPDATE) 及刪除 (DELETE) 操作。
- 為方便查詢員工資料，可建立 View：

```
CREATE VIEW AdminEmployeeDetailsView AS  
SELECT EmployeeID, Name, Position, Department, HireDate, Phone, IsActive  
FROM Employee;  
-- GRANT SELECT ON AdminEmployeeDetailsView TO admin_role; (已由 ALL PRIVILEGES 涵蓋)
```

b. **管理房間資訊 (Manage Room Information)**

- 管理房型價格 (Room\_Type.BasePrice):
  - 查詢價格: 直接 SELECT Room\_Type 表。
  - 設定價格: 直接 UPDATE Room\_Type 表的 BasePrice 欄位。

- 輔助查詢 View：

```
CREATE VIEW AdminRoomTypeInfoView AS
SELECT RoomTypeID, Name, BedCount, BasePrice
FROM Room_Type;
-- GRANT SELECT ON AdminRoomTypeInfoView TO admin_role;
```

- 設定季節 (Season):

- 管理員可直接對 Season 表進行查詢 (SELECT)、新增 (INSERT)、修改 (UPDATE) 及刪除 (DELETE) 操作。
- 輔助查詢 View：

```
CREATE VIEW AdminSeasonInfoView AS
SELECT SeasonID, Name, StartDate, EndDate, PriceAdjustmentPercent
FROM Season;
-- GRANT SELECT ON AdminSeasonInfoView TO admin_role;
```

- 設定季節房價 (Room\_Season\_Rate):

- 管理員可直接對 Room\_Season\_Rate 表進行查詢 (SELECT)、新增 (INSERT)、修改 (UPDATE) 及刪除 (DELETE) 操作。
- 輔助查詢 View：

```
CREATE VIEW AdminRoomSeasonRateDetailsView AS
SELECT
    rsr.RoomTypeID,
    rt.Name AS RoomTypeName,
    rsr.SeasonID,
    s.Name AS SeasonName,
    rsr.AdjustedPrice
FROM Room_Season_Rate rsr
JOIN Room_Type rt ON rsr.RoomTypeID = rt.RoomTypeID
JOIN Season s ON rsr.SeasonID = s.SeasonID;
-- GRANT SELECT ON AdminRoomSeasonRateDetailsView TO admin_role;
```

### c. 管理餐廳資訊 (Manage Restaurant Information)

- 管理餐廳資訊 (Restaurant):

- 管理員可直接對 Restaurant 表進行查詢 (SELECT)、新增 (INSERT)、修改 (UPDATE) 及刪除 (DELETE) 操作。
- 輔助查詢 View :

```
CREATE VIEW AdminRestaurantInfoView AS
SELECT RestaurantID, Name, DayOfWeek, OpenTime, CloseTime
FROM Restaurant;
-- GRANT SELECT ON AdminRestaurantInfoView TO admin_role;
```

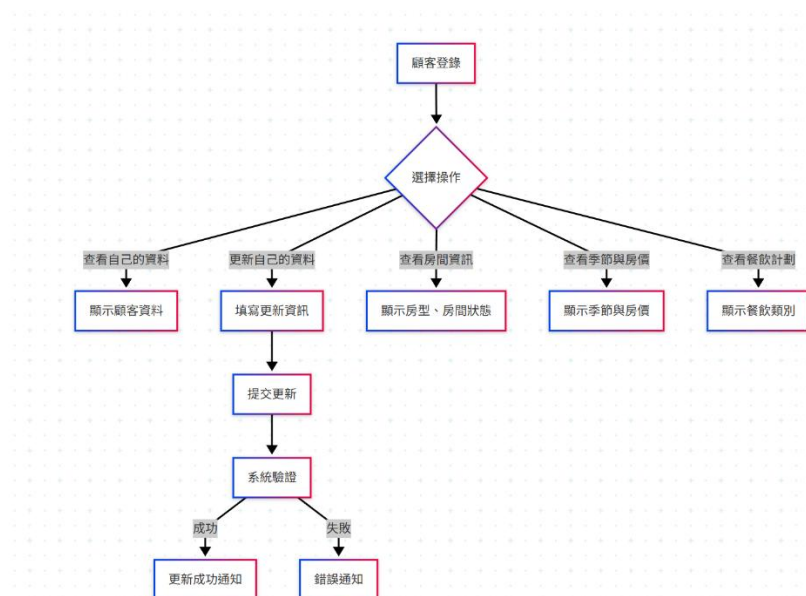
- 管理菜單項目 (Menu\_Item):

- 管理員可直接對 Menu\_Item 表進行查詢 (SELECT)、新增 (INSERT)、修改 (UPDATE) 及刪除 (DELETE) 操作。
- 輔助查詢 View :

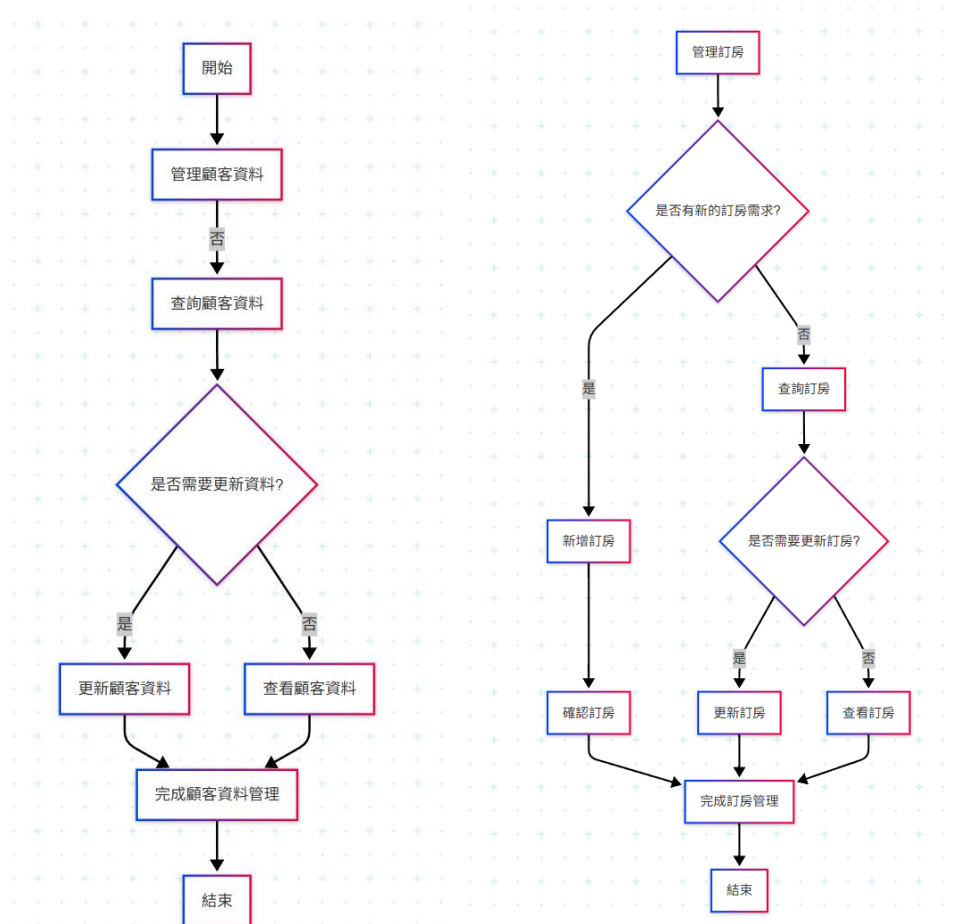
```
CREATE VIEW AdminMenuItemDetailsView AS
SELECT
    mi.MenuItemID,
    mi.RestaurantID,
    r.Name AS RestaurantName,
    mi.Name AS MenuItemName,
    mi.Category,
    mi.Price
FROM Menu_Item mi
JOIN Restaurant r ON mi.RestaurantID = r.RestaurantID;
-- GRANT SELECT ON AdminMenuItemDetailsView TO admin_role;
```

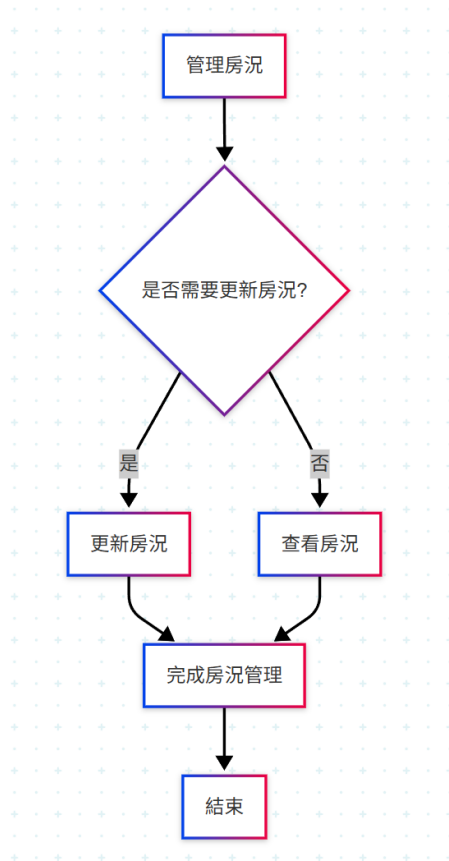
## 六、系統運作流程

### 1. 顧客

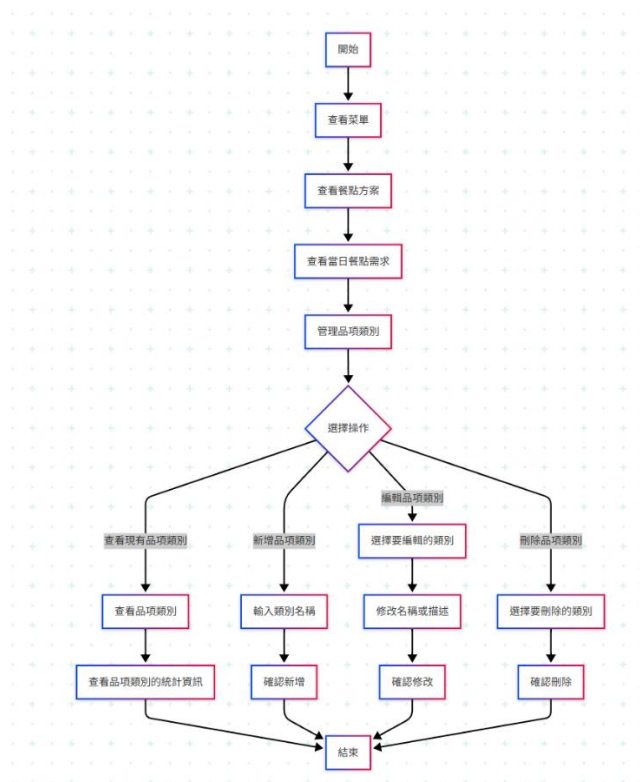


### 2. 前台人員

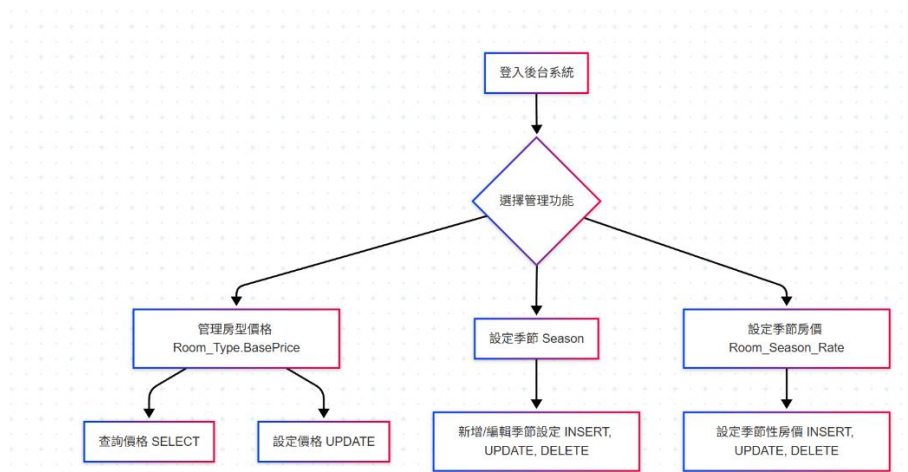
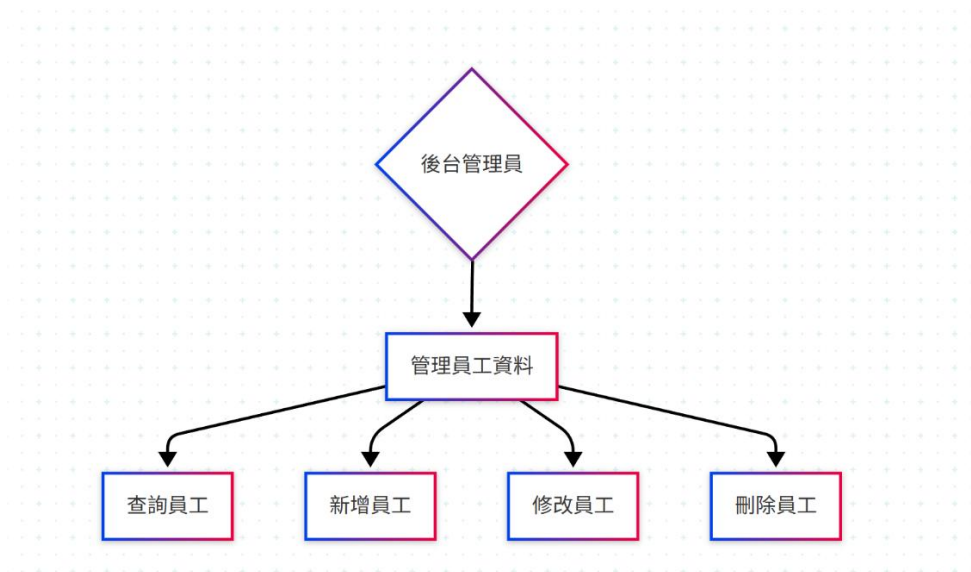


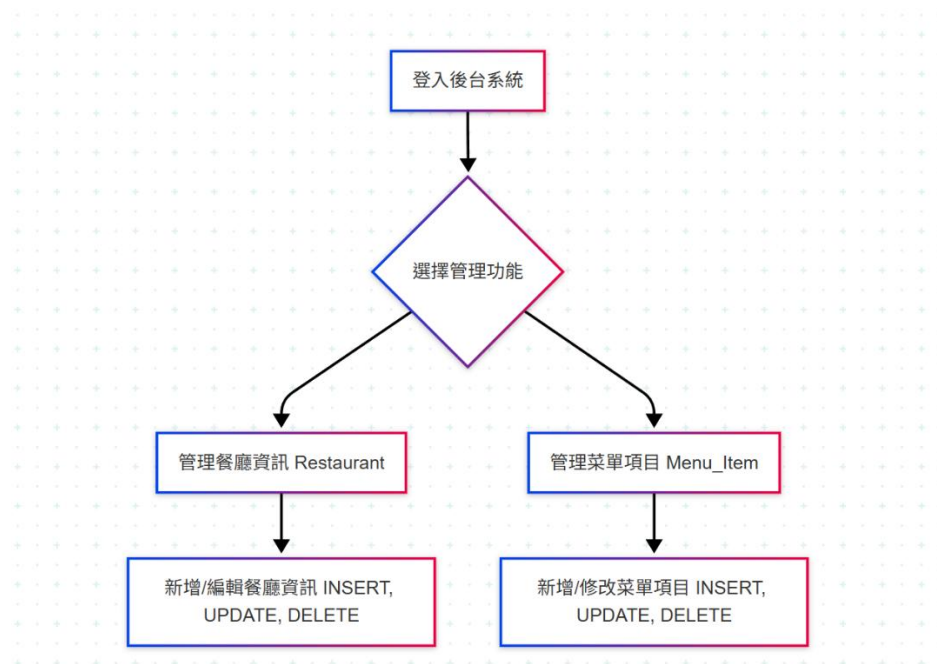


### 3. 廚師



#### 4. 後臺管理人員





## 七、分工

組長: 41143259 羅文鍵

副組長: 41143236 陳彥福

組員: 41143241 黃子峻

組員: 41143245 楊祐宇

## 八、心得

組長羅文鍵:

透過這幾次的作業，讓我學習到很多有關資料庫的知識，一開始我以為資料庫只需要儲存與讀取相關資料，沒想到在資料庫中還有那麼多的學問，例如：規劃 ER diagram、概念層設計、系統運作流程設計等等。我擔任本次專題報告的組長職位，也學習到工作分配與團隊合作的相關經驗等等，我覺得這次是非常寶貴的經驗。這次的期末報告除了讓我學到團隊合作的重要，也讓我體會到溝通和協調能力的重要性。在以後的工作中，我們可能經常需要面對不同的意見和想法，如何有效地整合這些意見並達成共識是團隊成功的關鍵。在這次的專題報告中，不僅提升了我的專業技能，也增強了我的自信心，讓我對未來的學習和工作充滿期待。

副組長陳彥福:

在此次期末專題資料庫設計中，從最初 ER diagram 設計資料表及其彼此關係跟對每個資料表欄位的完整性限制，經過上課內容與老師對資料表的相關建議，使我們專題組員在討論中也有諸多收穫，面對從零做出一個資料庫，一開始或多或少摸不清頭緒，但還好作業指示循序漸進的程序慢慢導進我們專題進度，使我們從 ER diagram、資料插入、權限設定到 View 設計，甚至於團隊分工，一步步按部就班完成，若以整體資料庫專題來說最困難的大概是 ER diagram 設計以及 View 設計，因為在資料庫設計重點之一是應存儲的資料跟使用者可以看見的資訊種類以及權限相關設定，此外也需要與組員溝通課餘時間完成以及分工解決各項進度，這讓我對溝通與分工的重要性更加重視。

組員黃子峻：

這次的期末專題從最初的構思到實作階段，過程中獲得了老師許多寶貴的建議，原本設計的資料表較為簡單，但在老師的指導下，我們逐步將各個資料表之間建立起緊密的關聯，並補足先前未完善的完整性限制。此外，老師也要求我們使用 ER diagram 來檢視飯店管理系統中可能的缺漏，藉此進行調整與修正，讓整體系統架構更為完整與合理，再到後面 View 的部分依照資料庫內的資料將其提取並展示，而在製作 View 的部分我們組檢視了前面資料庫的架構討論後再次對我們的飯店管理系統進行優化。此次的專題報告讓我了解到了資料庫的設計與建構需要經過縝密的思考，詳細分析資料與資料之間的關聯，讓我的規劃能力有近一步的成長，在實作時與組員的討論，讓我學習到溝通的重要性。

組員楊祐宇：

在做這幾次的資料庫報告和作業的過程中，無論是老師的建議或是組員間的討論，都讓我在這次報告中學到很多資料庫相關的知識。從完整性限制到 ER Diagram 到流程圖再到 View，每個地方都經過一番思考與討論，特別是 ER Diagram，我們不斷地修正與優化，起初的設計或許有些粗糙，但經過多次的修改後，從最初的”不甚完整”到現在的模樣，完整性限制也從原本的型態限制成長到標準表示式。非常感謝老師提供這次機會進行資料庫小組，除了 SQL 的基本學習之外，我也在這次的資料庫系統小組合作中體會到溝通與分工的重要性。每位組員都非常有自己的想法，而如何整合大家的意見、分配適當的任務、並在時限內完成每一階段的作業需求，都是這次報告中不可或缺的一環。最後特別感謝副組長督促這次作業的順利進行。