



系統名稱:飯店住宿管理系統

組別:第12組

組長姓名:江昀泰 系班別:四訊二丙 學號:3B117108

組員姓名:賴皇謀 系班別:四訊三乙 學號:3B017062

組員姓名:薛文庭 系班別:職四訊四甲 學號:9A917042

中華民國 一 一 三 年 一 月 二 日

目錄

一、資料與流程塑模.....	4
(一)DFD系統環境圖.....	4
(二)資料字典.....	6
(三)程序說明.....	7
二、物件塑模.....	8
(一)物件導向分析.....	8
(二)物件與類別的關係.....	10
(三)統一塑模語言(UML).....	12
三、資料設計.....	14
(一)資料設計概念.....	14
(二) DBMS組成元件.....	16
(三)Web式設計.....	17
(四)實體關係圖.....	18
(五)資料正規化.....	19
參考資料.....	21
附錄.....	22

圖(表)目錄

圖1.1 DFD系統環境圖 Level 0.....	4
圖1.1.2 DFD系統環境圖 Level 1.....	5
表1.2.1 資料字典.....	6
圖2.2.1 物件與類別的關係.....	12
圖2.3.1 塑模語言-用例圖.....	14
圖3.1.1 customers 集合.....	15
圖3.1.2 reservations 集合.....	15
圖3.4.1 實體關係圖.....	19
圖3.5.1 第一正規化.....	20
圖3.5.2 第二正規化.....	21

一、資料與流程塑模

(一)DFD系統環境圖

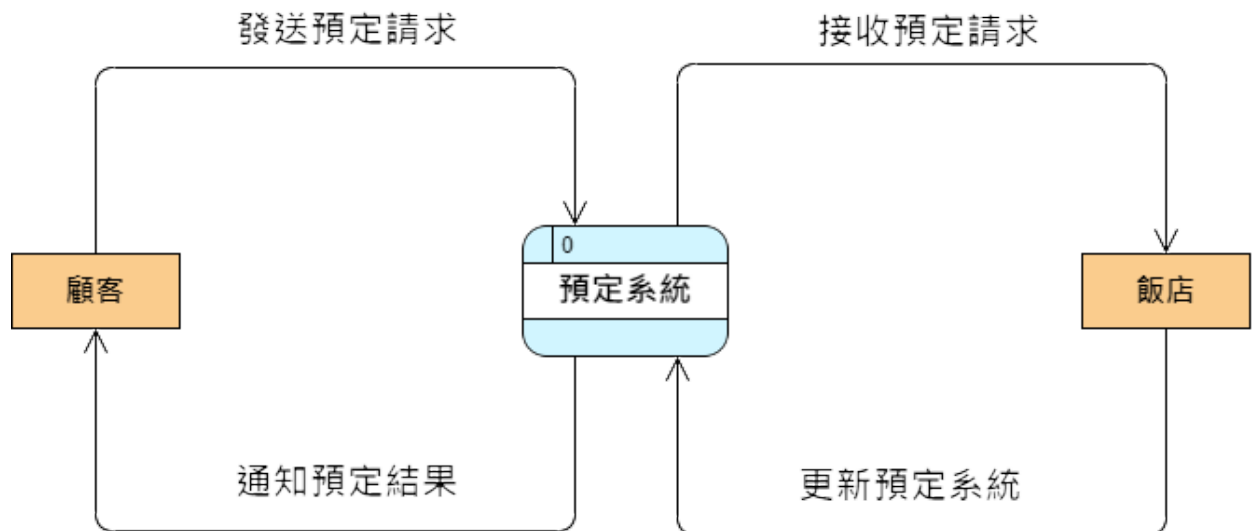


圖1.1 DFD系統環境圖 Level 0

Level 0 DFD

主要流程：顧客預定流程

參與者：顧客、預定系統、飯店

流程描述：

顧客透過預定系統向飯店發送預定請求。

飯店接收預定請求，進行處理，並更新預定系統。

預定系統將預定結果通知顧客。

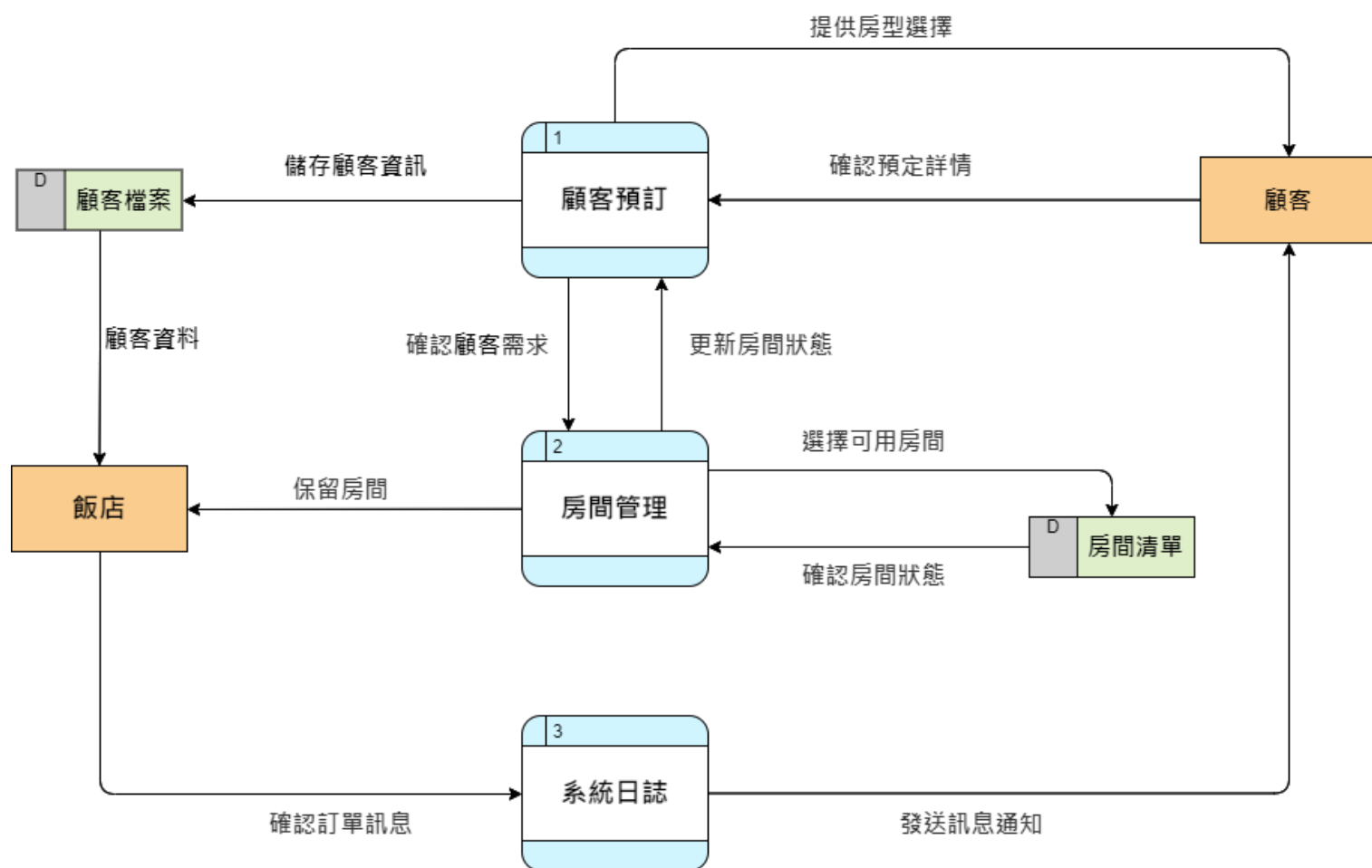


圖1.1.2 DFD系統環境圖 Level 1

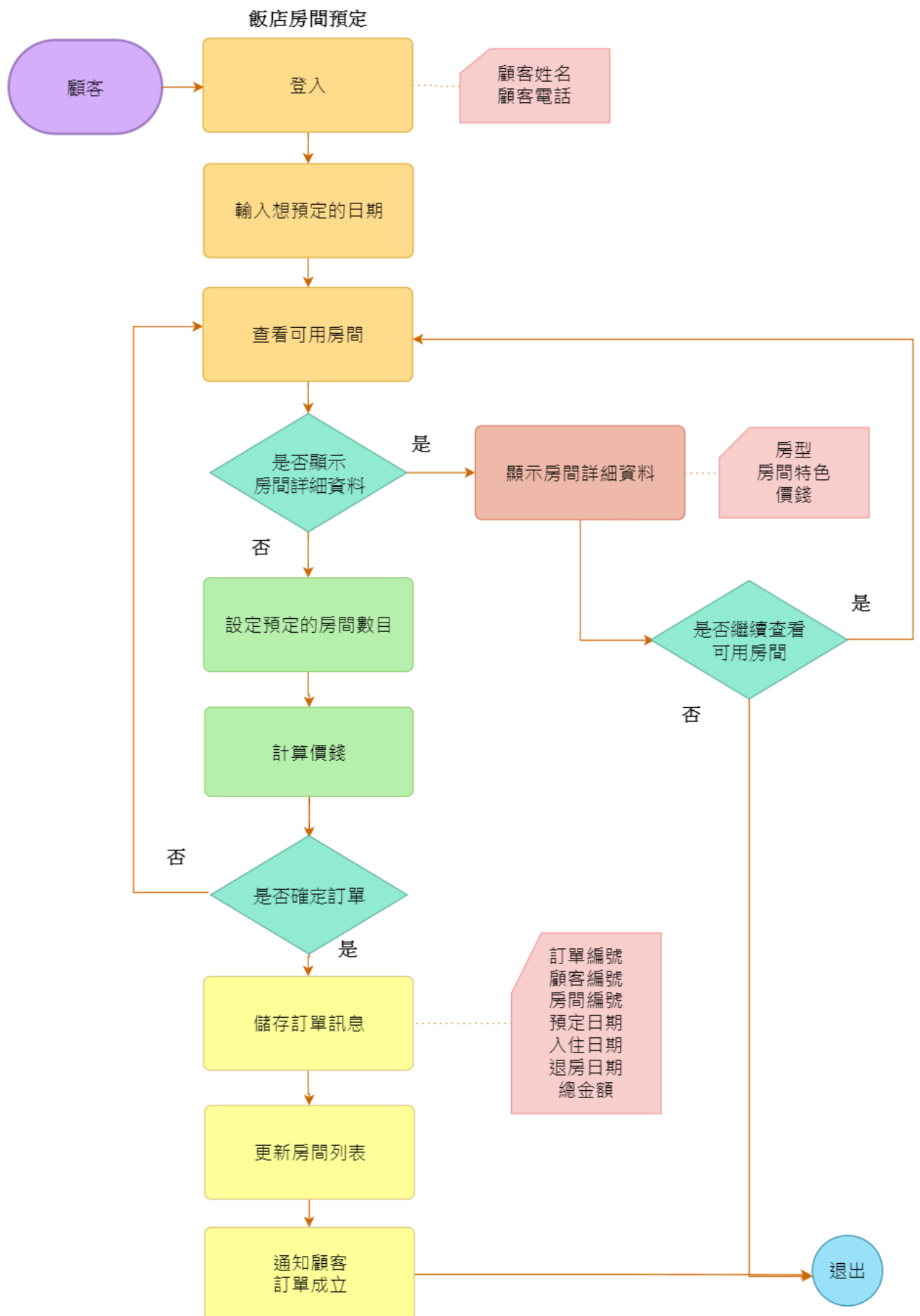
(二)資料字典

表1.2.1 資料字典

顧客資料	預定資料	飯店庫存
資料名稱：顧客資料	資料名稱：預定資料	資料名稱：飯店庫存
描述：存儲顧客的基本資料	描述：包含顧客的預定詳細資訊	描述：記錄不同房型的庫存數量
欄位： <ul style="list-style-type: none">顧客ID(CustomerID)姓名 (Name)電子郵件 (Email)聯絡電話 (Phone)	欄位： <ul style="list-style-type: none">預定ID (ReservationID)顧客ID (CustomerID)預定日期 (BookingDate)入住日期 (CheckInDate)退房日期 (CheckOutDate)房型 (RoomType)狀態 (Status)	欄位： <ul style="list-style-type: none">房型 (RoomType)可用庫存 (AvailableInventory)總庫存 (TotalInventory)

系統日誌	通知資訊
資料名稱：系統日誌	資料名稱：通知資訊
描述：紀錄系統操作和事件的日誌	描述：存儲發送給顧客和飯店的通知
欄位： <ul style="list-style-type: none">日誌ID (LogID)操作者 (Operator)操作時間 (Timestamp)操作內容 (Action)結果 (Result)	欄位： <ul style="list-style-type: none">通知ID (NotificationID)收件人 (Recipient)通知內容 (Content)日期時間 (DateTime)

(三)程序說明



二、物件塑模

(一)物件導向分析

顧客資料
日誌ID (LogID) 操作者 (Operator) 操作時間 (Timestamp) 操作內容 (Action) 結果 (Result)
新增顧客資料

預定資料
預定ID (ReservationID) 顧客ID (CustomerID) 預定日期 (BookingDate) 入住日期 (CheckInDate) 退房日期 (CheckOutDate) 房型 (RoomType) 狀態 (Status)
新增預定資料

飯店庫存
房型 (RoomType) 可用庫存 (AvailableInventory) 總庫存 (TotalInventory)
更新庫存

系統日誌
日誌ID (LogID) 操作者 (Operator) 操作時間 (Timestamp) 操作內容 (Action) 結果 (Result)
記錄系統操作

通知資訊
通知ID (NotificationID) 收件人 (Recipient) 通知內容 (Content) 日期時間 (DateTime)
新增通知

系統設定
設定ID (SettingID) 設定名稱 (SettingName) 設定值 (SettingValue)
取得設定值

(二)物件與類別的關係

Customer(顧客)與 Reservation(預定)的關係:

關聯型別: 1..*

敘述:

一個顧客可以進行多次預定。箭頭指向Reservation類別。

顧客可以透過呼叫 submitReservationRequest() 方法提交預定請求。

Reservation 類別記錄了與預定相關的詳細信息, 包括預定日期、入住日期、退房日期等。

HotelInventory(飯店庫存)和 Reservation(預定)的關係:

關聯型別: 1..1

敘述:

每次預定都關聯到特定的房型庫存。箭頭指向HotelInventory類別。

HotelInventory 類別記錄了不同房型的庫存。

與 Reservation 類別關聯, 表示預定與庫存之間的關係。這種關係可以用來檢查庫存是否足夠, 以滿足客戶的預定需求。

SystemLog(系統日誌)與 HotelBookingSystem(飯店預定系統)的關係:

關聯型別: 1..*

敘述:

飯店預定系統使用系統日誌來記錄操作。箭頭指向SystemLog類別。

SystemLog 類別記錄了系統的操作日誌。

與 HotelBookingSystem 類別關聯, 表示系統操作與日誌記錄之間的關係。這樣可以追蹤系統的各種操作和事件。

Notification(通知)與 HotelBookingSystem(飯店預定系統)的關係:

關聯型別: 1..*

敘述: 飯店預定系統使用通知來向顧客和飯店發送通知。箭頭指向Notification類別。

Notification 類別記錄了發給顧客和飯店的通知。

與 HotelBookingSystem 類別關聯, 表示系統產生通知的功能。這種關係用於在特定事件(例如預定確認或取消)發生時向相關方發送通知。

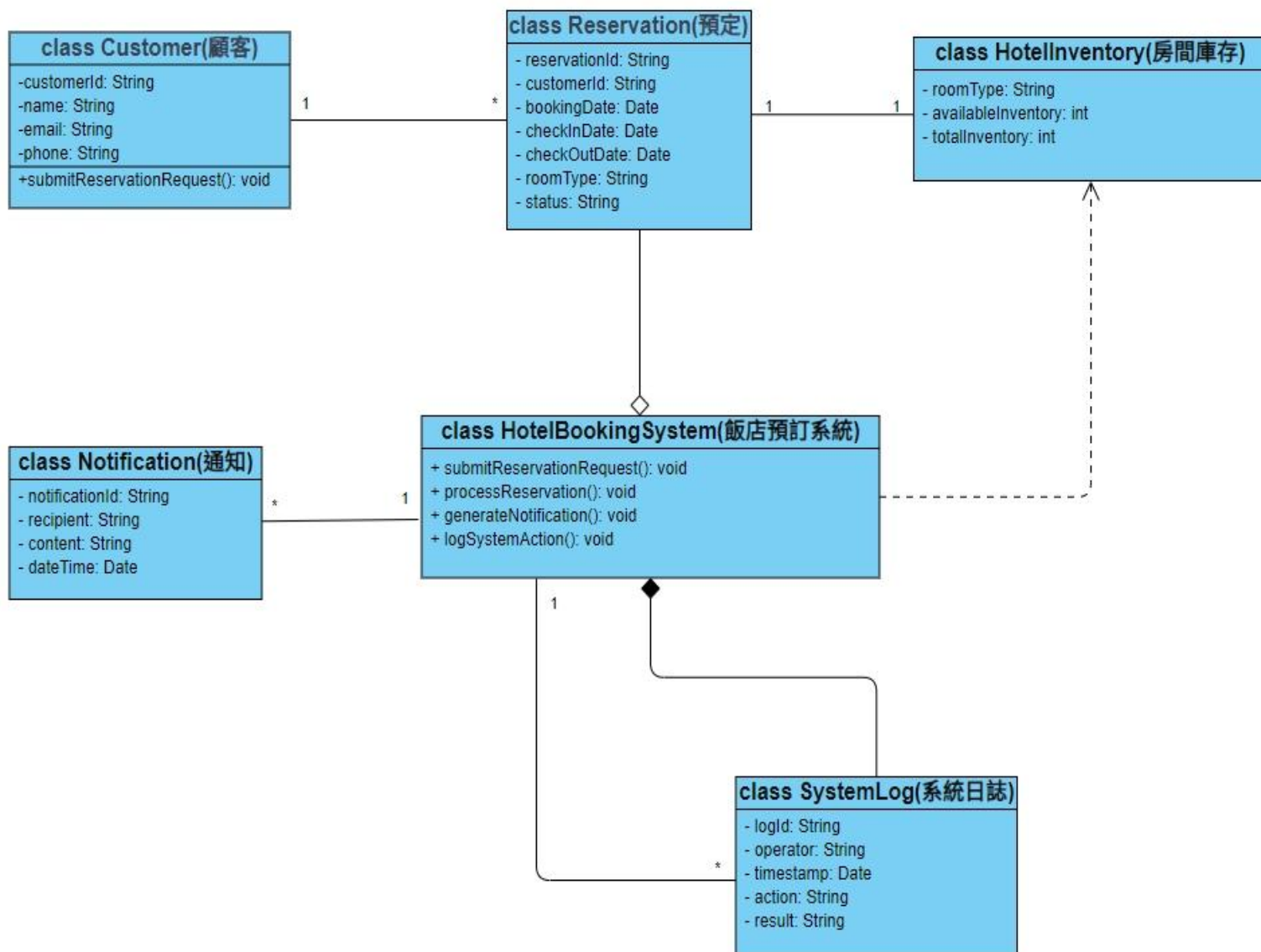


圖2.2.1 物件與類別的關係

(三)統一塑模語言(UML)

使用者角色：

Customer(顧客)：使用系統提交預定請求。

System User(系統使用者)：處理預定請求、生成通知、記錄系統操作。

Hotel(飯店)：接收通知並處理預定請求。

用例：

Submit Reservation Request(提交預定請求)：顧客使用系統提交預定請求。

Process Reservation(處理預定請求)：系統使用者處理顧客提交的預定請求。

Generate Notification(生成通知)：系統使用者為顧客和飯店生成通知。

Log System Action(記錄系統操作)：系統使用者記錄系統的操作和事件。

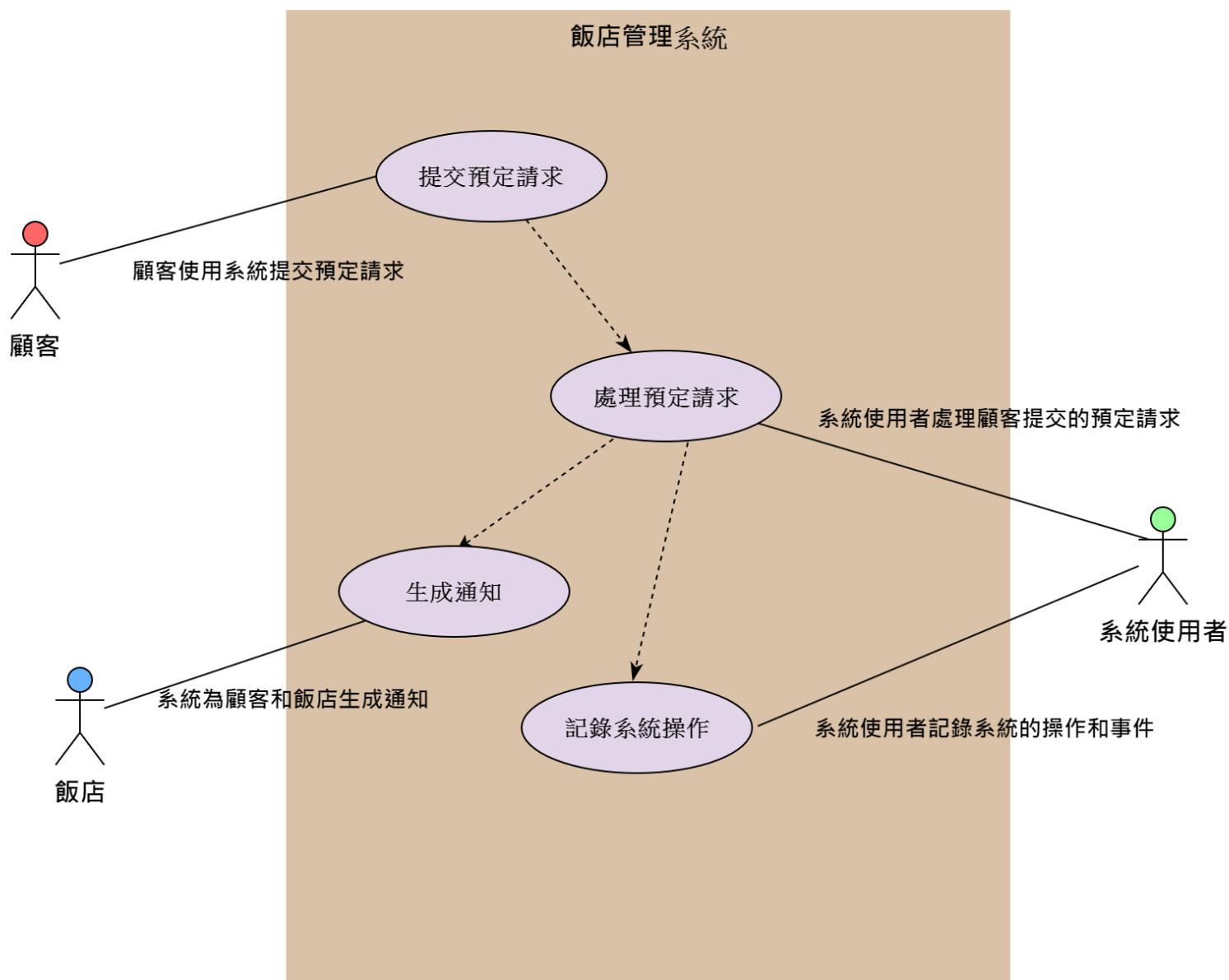


圖2.3.1 塑模語言-用例圖

三、資料設計

(一)資料設計概念

MongoDB 是一個文檔導向的 NoSQL 資料庫，與傳統的關聯式資料庫不同。

```
28   const Customer = mongoose.model("Customer", {  
29     customerId: String,  
30     name: String,  
31     email: String,  
32     phone: String,  
33   });  
34
```

圖3.1.1 customers 集合

customers 集合用於存儲顧客的基本信息，每個文檔表示一個顧客。

```
const Reservation = mongoose.model("Reservation", {  
  reservationId: String,  
  customerId: String,  
  bookingDate: { type: Date, default: Date.now },  
  checkInDate: Date,  
  checkOutDate: Date,  
  roomType: String,  
});
```

圖3.1.2 reservations 集合

reservations 集合用於存儲預訂信息，每個文檔表示一個預訂。

- 顧客信息和預訂信息分別存儲在不同的集合中，可以提高查詢效率，同時減少冗余信息。
- 在 customers 集合中，可以使用唯一的 customerId 作為顧客的辨識符。
- 在 reservations 集合中，使用 customerId 來建立與 customers 集合的關聯，以便查詢特定顧客的預訂信息。
- 可以考慮添加其他字段，如價格、特殊需求等，根據實際需求進行擴展。

優點

- 靈活的架構：MongoDB 是一個 NoSQL 數據庫，使用文檔對數據進行建模，而不需要固定的架構。這使得數據庫能夠輕鬆擴展和適應不斷變化的需求。
- 容易擴展：MongoDB 允許在不停機的情況下水平擴展，即增加更多的節點以處理更大的數據量。這提供了良好的擴展性，能夠應對高流量和大規模的應用。
- 高性能：MongoDB 通常具有快速的查詢性能。它支援多種查詢操作，並提供索引和分片等功能，以加速數據訪問。
- 自動分片：MongoDB 具有自動分片的功能，可以水平分割數據庫，使數據能夠均勻地分佈在多個伺服器上，提高擴展性。
- 支援 **JSON** 格式：MongoDB 使用 BSON (Binary JSON) 格式來存儲數據，這使得在應用和數據庫之間進行數據交換變得簡單。BSON 是 JSON 的擴展，支援更多的數據類型，同時更有效率。
- 強大的查詢語言：MongoDB 支援豐富的查詢語言，包括範圍查詢、正則表達式、排序、索引等，使得查詢和分析數據變得更加方便。
- 社區和生態系統：MongoDB 擁有龐大的開發者社區和生態系統，這意味著可以輕鬆找到相關的工具、庫和文檔。這也為解決問題和持續改進提供了支持。
- 應用廣泛：MongoDB 適用於多種應用場景，包括內容管理、用戶數據管理、日誌和活動跟蹤等，並且在 Web 開發中非常流行。

(二) DBMS組成元件

1. 數據庫(Database):

MongoDB 提供一個數據庫存儲和組織數據的地方。在飯店預訂系統中，你可能會創建一個專門的數據庫，用於存儲和管理與預訂相關的信息。

2. 集合(Collection):

MongoDB 中的集合類似於關聯型數據庫中的表。在飯店預訂系統中，你可以有多個集合，例如一個用於存儲顧客信息的集合，另一個用於存儲預訂信息的集合。

3. 文件(Document):

MongoDB 使用文檔的概念，它類似於關聯型數據庫中的行。每個文檔是一個 JSON 格式的對象，包含具體的數據。在飯店預訂系統中，一個文檔可以表示一個顧客或一個預訂信息。

4. 字段(Field):

文檔中的屬性稱為字段。在飯店預訂系統中，一個文檔可能包含姓名、入住日期、退房日期、房型等字段。

5. 索引(Index):

索引可加速查詢操作，提高數據訪問的效率。在飯店預訂系統中，你可能會創建索引以加速按照顧客ID或預訂日期進行查詢。

6. 連接器(Connector):

連接器是用於在應用程序和 MongoDB 數據庫之間建立連接的工具或庫。在 Node.js 中，你可能使用 Mongoose 或 MongoDB 驅動程序等連接器。

(三)Web式設計

1. 非同步事件驅動架構：

Node.js 使用非同步的事件驅動模型，使其非常適合處理大量且併發的請求，這對於 Web 應用程式來說是至關重要的。這種模型有助於提高系統的效能，因為它不會阻塞在 I/O 操作上。

2. 快速的執行速度：

Node.js 使用了 Google 的 V8 引擎，這是一個高效能的 JavaScript 引擎。這使得 Node.js 適用於處理實時應用程式，如即時通訊、遊戲伺服器，以及飯店預訂系統中需要即時更新的功能。

3. 模塊化架構：

Node.js 採用 CommonJS 模塊系統，這使得代碼結構更加模塊化和易於管理。這是一個重要的特性，特別是當應用程式變得複雜時，可以更容易地維護和擴展。

4. 豐富的套件生態系統：

Node.js 擁有龐大而活躍的套件生態系統，你可以使用 npm(Node.js 套件管理器)來輕鬆地集成第三方套件和模組，這有助於節省開發時間，因為你可以使用他人已經開發和測試過的模組。

5. 跨平台性：

Node.js 可以在多個平台上運行，包括 Windows、Linux 和 macOS。這使得開發人員能夠在不同的環境中部署和運行他們的應用程式。

6. 支援即時通訊：

Node.js 對於建立即時通訊的應用程式非常適合，例如聊天應用、即時通知等。這對於飯店預訂系統中需要及時更新預訂狀態的情境很有幫助。

7. 容易學習的語法：

JavaScript 是一種廣泛使用的語言，大多數前端開發人員都具備 JavaScript 的基礎知識。使用相同的語言來開發前端和後端，有助於加快開發速度並降低學習成本。

8. 資料庫整合：

Node.js 配合各種資料庫(例如 MongoDB、MySQL、PostgreSQL)的驅動程式，使得資料庫整合相對容易。在飯店預訂系統中，你可以使用 Node.js 配合 MongoDB 來輕鬆地存儲和檢索預訂資料。

(四)實體關係圖

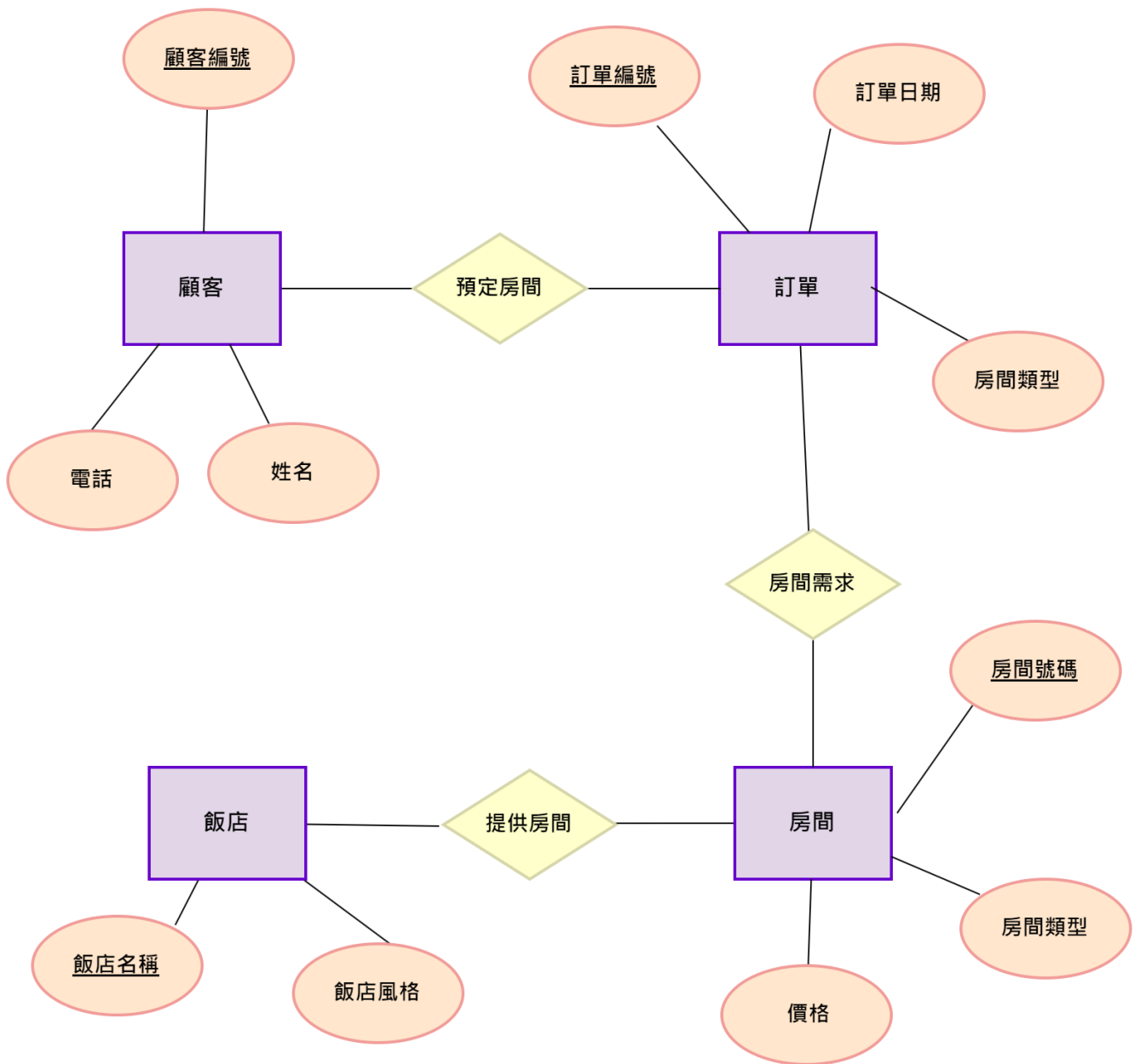


圖3.4.1 實體關係圖

(五)資料正規化

第一正規化

確保每個表格中的每個欄位中的值都是不可再分的。



圖3.5.1 第一正規化

第二正規化

在進行第二正規化時，我們將確保每一個非主鍵屬性完全依賴於主鍵。



圖3.5.2 第二正規化

顧客表 (Customers) 的 顧客ID 為主鍵 (PK)。
預定表 (Reservations) 的 預定ID 為主鍵 (PK),
並且 顧客ID 是外鍵 (FK) 參照 顧客表 (Customers) 中的主鍵。
飯店庫存表 (HotelInventory) 的 房型 為主鍵 (PK)。
系統日誌表 (SystemLogs) 的 日誌ID 為主鍵 (PK)。
通知表 (Notifications) 的 通知ID 為主鍵 (PK)。

參考資料

<https://ithelp.ithome.com.tw/articles/10261018>

<https://israynotarray.com/nodejs/20191228/1009061739/#%E5%8E%9F%E5%9B%A0>

<https://www.ithome.com.tw/pr/158123>

https://developer.mozilla.org/zh-TW/docs/Learn/Server-side/Express_Nodejs/mongoose

<https://ithelp.ithome.com.tw/articles/10311067?sc=pt>

附錄

工作內容

江昀泰:WORD報告內容10%、簡報內容45%

賴皇謀:DFD系統環境圖、塑模語言-用例圖、程序說明圖、實體關係圖

WORD報告內容30%、簡報內容25%

薛文庭:程式實作、資料查詢、WORD報告內容60%、簡報內容30%

實作

[飯店預訂系統 - Google Chrome 2024-01-08 22-34-30.mp4](#)

[MongoDB Compass - localhost_27017_hotelReservation.reservations 2024-01-08 22-38-27.mp4](#)