

物件導向設計與模擬

第4組 小組報告

組員：

學號	姓名	學號	姓名
b06505002	張在然	b06505006	陳奕舟
b06505004	莊博翰	b06505047	陳銘杰

系統設計：

component:

主要設計理念便是希望設計出來的component 都可以對應到一個真實世界的物件，如USER, ORDER, PLAN, ...等等，如此一來我們寫出來的class 和class 之間自然可以有直觀的對應關係，也可以比較輕易的達成Low-Coupling and High Cohesion，原因是本來現實中會產生這些類別必然代表這些類別彼此之間不會有太多互相牽連的關係才會被分解成不同類別，同理，同一個類別中的物件也因該是有依存關係才會被分在同類，因此我們才使用這個設計理念。

以下舉出幾個比較關鍵的class。

class User:

主要的設計理念是希望這個class 可以整合各種有關USER的資料如書籤，訂單

查詢紀錄....等等。同時這也是一個主要使用database 的class，這裡我們的想法是，任何一個database的改動必然與一個USER 有關，那何不把USER寫成一個使用database的介面，這樣database 的取用更動也比較好管理，同時USER中也存了一份和DB中一樣資料，作為DB的catch，可以大幅加速查詢的效率

Class Order:

對應到真實世界的訂單，Order 中自然也會有edit, delete這些method，也就是只要拿到這個object，就可以直接修改它，不需要而外使用其他class來輔助

exp : Order.edit(content) 而非 Order.getUser().edit(OrderID,contant)

它會自己確認現在登入的user，並自行呼叫user同步變更，其他method也都遵循這個理念。

Class Plan:

也就是方案，可能是存了選擇的飯店，check in out 日期，房間數等等，可以隨意地被更動而不會影響到USER，或DB的值，簡單來講就是給使用者(可能是其他programmer)一個暫存資料並且會及時檢查資料正確性和可行性的介面，並可以幫忙計算一些價格之類的，只有使用者呼叫Plan.toOrder()，才會真的生成一個訂單，並存到USER，和DB中。

Class Hotel:

一個用來記錄飯店資料的class，有一個static hotel[] hotelList，會記錄一些不變飯店資料，如此一來我們可以直接用hotel 裡的static method search()

來進行搜尋，先比對不變的資料，最後在去DB確認該飯店在指定的時間是否還有空房，盡可能降低DB的使用量。

Reusibility:

簡單來說，就是盡可能地讓我們寫的class 可以在不同地方被重複利用，不要因為缺了什麼功能或data member，導致我們要另外寫一個class。基於這個理由，我們再設計時竟可能不把class 裡的data member寫死，而是拆出一個class，以下會做說明

Class : RoomNum CheckinoutDate

我們特別把什麼類型的房間有幾間，這個資料寫成一個class 如此一來任何PLAN, ORDER, 或search 只要有關房間數的資料都可以重複利用這個class

同樣的，check in check out 的時間也可以包成一個class 在各處使用。

我們也在這些class 裡寫了一些method 如計算總天數，人數，使這些計算

不需要在每次使用重寫code

Class: Plan

我們的想法是，一個計劃可以被當成一個Order裡的data member 同時也可以

被當作一個書籤讓使用者紀錄下來，所以我們把plan 寫成可以重複被兩種情況使用。

Class:Searchinput

也就是一個搜尋時輸入的資料，把它獨立寫成class 的原因有兩個，一個是

如此一來我們可以把它視為搜尋紀錄，這樣我們查詢搜尋紀錄的功能就可重複利用這個class。另一個理由是如果將來我們想要做其他搜尋方式，可以繼承searchInput，提高程式的extensibility。

Design pattern:

基本上我們寫code使用到design pattern 都是為了讓撰寫更容易和有結構性
大部分時間都是為了讓code 更簡潔和有彈性才會使用到design pattern。

builder pattern

建立user的時候，我們會一步一步加入各類資料(訂單，書籤，紀錄)，也就是不會在new 的時候就一口氣把所有資料都輸進去，而是用set()的方式依樣一樣加，會採取這個作法的原因是因為USER是一個十分複雜的class 如果要一口氣

initialize 會使code 便很雜亂。也容易出錯

singleton pattern

我們寫了一個叫做DB的class，其中有一個static data member db 作為整支程式和database 的连接點，當然也寫了static get()method，如此一來就不必擔心重複connect db而造成bug。

State Pattern:

ORDER這個class的object有 valid invalid 兩個state 可以視為現在db中到底是否存在這個訂單，Order invalid 時無法再對其進行修改，只能讀取。會存在invalid 這個state 的原因是我們可能無法同時移除程式所有有關這個訂單的reference 但卻可以輕易的把這個訂單標記為invalid，以代表它被刪除了。

Prototype Pattern

對於幾乎所有class 除了 USER ,DB，我們都有 overwrite clone()，也就是讓object 自己決定複製時那些member 要給reference 那些要在new 一份，以減少其他programmer 在操作上的困難，也可以避免複製許多const object 造成浪費許多memory。

Chain Of Responsibility Pattern

有時候同樣的工作可能會在不同的class 產生需求，比如說書籤，有時候可能是 `USER.deletePageMark(id)` 也可能是 `Plan.unmark()`，因此我們的設計是會把這個指令往 user 送，途中如果發現有問題(如user 已登出)就會被擋下來，

不然最後就是統一由User 解決。同樣`order.edit()`也是同理， order 會自己先檢查修改是否合理，沒問題才把需求送給user 去修改DB。

Database設計

一、目標

1. 以上層定義的物件類別定義表格，以利模組化分工
2. 與上層之間傳遞的資料多半是包裝過後的物件
3. 犧牲儲存空間換取較直觀簡易的Schedule操作

二、Table (參考Appendix – DB)

1. 由上層定義的類別：User、Hotel、Order、Plan、Search
2. 輔助管理飯店訂房資訊：Schedule、Today
3. 額外功能輔助：Comment、Usermail

三、Modulization

1. 以上層定義的類別型態定義表格

在DB中，需要維護 User、Hotel、Order等類別型態的getter、setter亦或是任何編輯、刪除這些資料的請求，而這些類別又多互相存有 (如User 即包含Order、Plan、Search，Order和User也互相存有)，故在規劃表格時，皆直接以這些類別來定義表格內容，以利模組化分工。

2. 拆解與儲存

如上一點所提及，有多種類別的data member是互相持有的，故當上層丟下來一個User並要儲存時，模組化的設計表格就有利儲存的簡便性，僅需在記下User的ID後分別呼叫Plan、Order、Search的setter個別儲存在它們的專屬Table中即可完整記錄一位使用者的所有資訊。

3. 拼接及創建

模組化的設計表格同樣也給DB中各個類別的getter帶來簡便性並且也非常直觀，假如今天DB收到一個User ID並要回傳一個User物件，DB僅需以此ID為媒介，分別呼叫Order、Plan、Search的getter，將User的各個元件依序補上，並以結果回傳之。

在此種設計模式下，越上層且資料越龐大的資料型態(data member多為自定義類別)，它所儲存在表格的內容會越少，而越低階的內容越單純的資料型態(data member多為基礎資料型態)，儲存在表格的內容會越多，就如同生物的外皮與骨肉的關係，創建User即是先準備好一副外皮，再以ID將它的骨肉所填滿，透過模組化的設計表格，可將此過程的分工達到極簡化，也更好進行除錯與擴充。

四、Schedule

在如何管理剩餘房間的策略上也與上學期有所不同，上學期是採取一間飯店一個固定長度的一維向量，並將此向量與飯店資訊存在一起，這學期做了一些改變以提升彈性與執行效率：

1. 各家飯店的行程表各自存成表格

1500間飯店都有屬於它們的行程表，並以每條row為一天，記錄下任一間飯店在任何一天所剩餘的三種房間數量，雖然這樣的方式犧牲的是儲存空間，但由於對行程表的操作多只針對單一飯店，故每次操作也只會涉及單一表格，在編輯、瀏覽行程表上會更有效率且更為簡便。

2. 可記錄的時間範圍更為彈性

上學期採用的是固定長度為365的向量，並僅能指定範圍在一月一號到十二月三十一號，而由於此次系統設計將行程表完全表格化，行程表的範圍可隨時透過該表格的init() 函數重新定義，也增加了許多程式運行上的彈性。

3. 引入真實時間的概念

雖然沒有連結到網路，但本系統仍能透過讀取電腦日期來模擬出真實環境的一些操作，如在每天的第一次開啟系統時，即會檢查是否有訂單的退房日期已在今天之前，並會協助其完成訂單；也可透過檢查今日日期，阻擋對今日以前的行程表所做的任何操作。

4. 擴增日期與增加房間數量 (參考Appendix – DB)

受益於每間飯店各自獨立的行程表編排方式，系統可對檢察：同時間範圍內最大房間可增加數量、同房型組合能擴增的最長訂房日期範圍以上兩點作出快速的判斷，以達到對「加訂房間」與「增加日期」的前檢查，這也是上學期沒有做到的功能。

5. 模組化

模組化的概念也運用到編輯訂房資訊的操作上，會更動行程表的函式有三種：新增訂單、刪除訂單、編輯訂單，新增與刪除訂單又會呼叫同一支函式以更動行程表，唯一不同的是房間數量會以正負值區分該動作是新增或是刪除訂單，而編輯訂單是由新增訂單與刪除訂單兩個動作組合起來完成的，由於在上一點中DB已提供給上層判斷編輯訂單合理性的函式，在DB中就不需考慮會收到任何不合常理的變更，本來在上學期編輯訂單是最複雜的一道手續，但本系統透過函式工作內容的單一化，達到越是複雜的手續，越能以分工的方式完成的概念。

UI設計

目標:

1. 將系統的各個功能獨立出來，簡化使用者的操作流程。
2. 能夠以最簡單的方式與中心的class溝通，讓系統的功能須添加或修改時更加的方便。
3. 建立直觀且容易理解的操作介面，讓使用者能夠簡單的來理解操作方法。

與主程式的溝通:

由於訂房系統的核心部分已經充分地與database做好連結，所以進行UI設計時只需要承接並運用核心的class與function，除了例外處理外幾乎不需要碰到database。

在主程式中，如上方所提，我們已經有User、Hotel、Order、Plan、Search等類別，分別對應到UI的不同頁面：

GUI_login:

是系統開啟後首先出現的頁面，使用者可以選擇登入，註冊或是以訪客模式登入。以訪客模式進入系統的話，只能使用搜尋飯店的功能，無法進行訂房的動作。

使用的主要函數: User.signup(), User.login()。

在登入之後，整個系統將保持同一個使用者，直至登出為止。

GUI_user:

即為使用者介面中，類似首頁般的存在，連接著系統的四大功能: 用戶資料修改, 查詢飯店, 訂單修改/刪除, 搜尋紀錄/書籤。此外，這也是系統提供使用者登出的地方: 在系統中的任一頁面，都可以直接連結至此，並隨時進行操作。

使用的主要函數: User.logout()。

GUI_user_data:

使用者的密碼及email修改。要更改密碼和電子郵件信箱時，需先輸入舊密碼。

使用的主要函數: editEmail(), editpassword()。

GUI_search:

系統最主要的功能，用來查詢飯店列表，使用者可以輸入飯店星級範圍, 地點, 價格範圍, 入住日期範圍, 房間數量以及人數。按下搜尋鍵後，程式會把使用者輸入的資料做成搜尋物件Search_input，並交由核心部分的search函數進行適合飯店的查找，和人數, 房間數量的自動分配，並回傳一個Plan的矩陣plan_input，記錄著所有適合的結果。

接下來，程式會把plan_input矩陣以JTable的方式呈現，使用者還可以選擇以星級，飯店序號或價格來做排列。按下排列的按鈕後，會直接呼叫sort函數，並依照程式核心部分已撰寫好的不同種comparator來進行排列。

接下來，使用者可以點選清單中所推薦的飯店計畫，並選擇要不要加入書籤，或直接進入訂房的下一步驟。

使用的主要函數: Hotel.search(), Arrays.sort()。

GUI_hotelpage:

呈現飯店的資料，使用者評論，以及飯店方案的修改。程式會以getter的方式去找GUI_search中選中的plan_input(除非使用者是從record的書籤處進入的)，並當作此頁面的資料來源。在資料呈現的部分，除了飯店的地址，使用者想訂的房型和日期資料等，還提供了google map的展示。

使用者評論的部分，我們規定好，使用者必須在已經訂過該飯店之後才能進行評論，否則會跳出exception。最後飯店修改的地方，我們提供使用者對方案中房間數量的彈性修改，以對應使用者的可能需求。

使用者按下確定後，會將plan儲存起來變成order，進入到下一頁面。

使用的主要函數: loadcomments(), addcomments(), Desktop.browse() (開啟地圖)。

GUI_order_confirm:

進行訂單的最後確認以及修改。頁面在初始化時，會先確認使用者是如何進入此一頁面的(來自hotelpage或order_manage)。使用者同樣可以更改房間數量，但這裡還加入了日期範圍的修改，以便使用者在訂購之後對訂單進行更動。以登入的使用者按下確認鍵後，即訂購/修改成功，系統會寄出一封電子郵件(若使用者有事先登記)。若是以訪客模式進入，系統會跳出登入的提醒。

使用的主要函數: maxExtendDate().contain() (用來確定可修改的日期範圍), Order.confirm(), editOrder()。

GUI_order_manage:

進行訂單的刪除與修改。頁面會先載入使用者的訂單列表，使用者可以選擇直接刪除，還是進入order_confirm修改。此外，使用者也可以透過此頁面，進入飯店資料頁(hotelpage)，來進行評論的動作。

使用的主要函數: User.getOrderList() (為訂單的arraylist), User.deleteOrder()。

GUI_record:

即使用者的搜尋紀錄以及記錄下來的書籤。搜尋紀錄的資料類別對應Search_input，書籤則是對應Plan，使用者可以在兩個模式間隨意切換，並對兩者進行刪除的動作，或以此跳至搜尋頁面和飯店資料頁面。

使用的主要函數: `User.getRecord()`, `User.getPageMark()`, `User.clearRecord()` (會直接刪除所有搜尋紀錄), `User.getPageMark().remove(index_plan)` (刪除選中的書籤)。

共通部分:

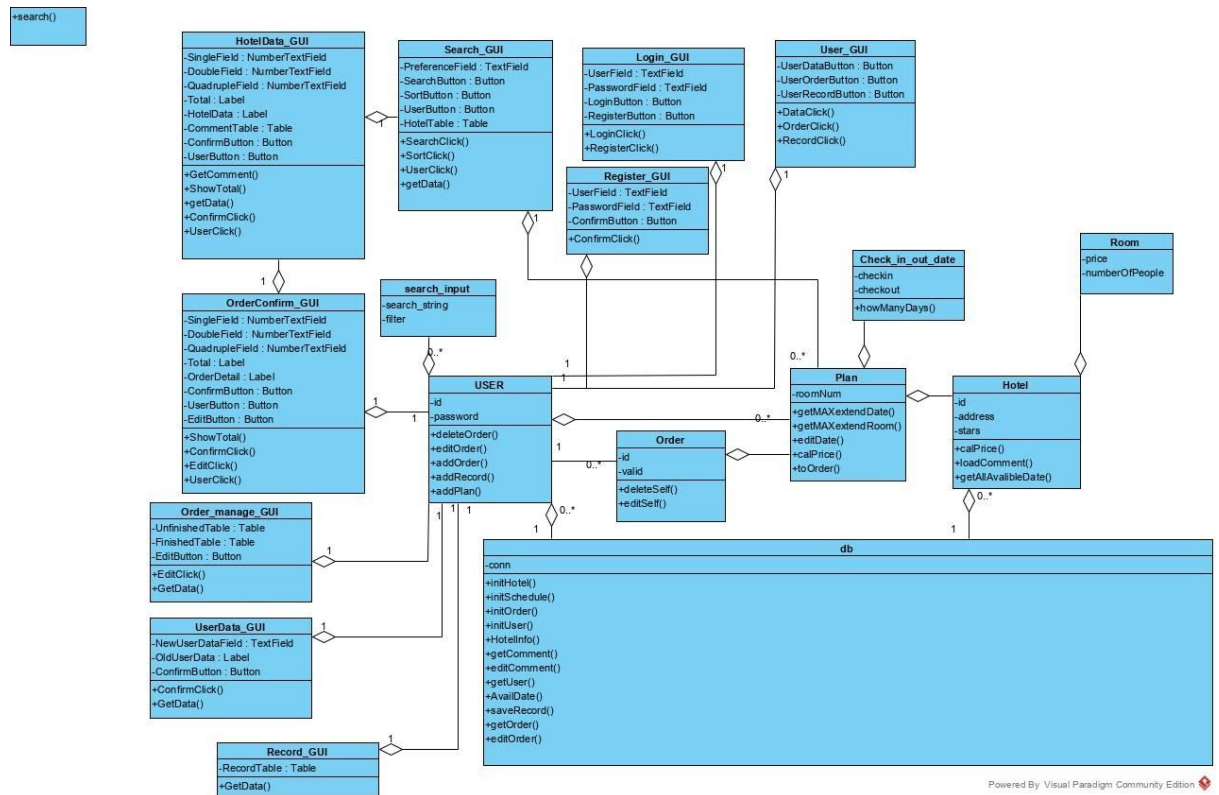
Exception: 在任何地方有接到exception時，都會跳出小視窗提醒使用者。

分歧來源: 指的是當一個頁面的資料來源不同時，如何進行處理。舉例來說，`hotelpage`的`plan`資料可能是來自`search`，也可能來自`record`。此時只要先用一布林值，如`isRecord`，就可以判定資料來源，並在頁面初始化時依照此布林值來決定載入的資料。

背景: 算是系統美化的部分，在所有的頁面中都存在，利用`ContentPane`的方式來載入並設定背景。

Appendix diagram

1. Class diagram



Appendix DB

1. Table

(a) Hotel (飯店資訊)

Hotel ID	star	address	Single price	Double price	Quad price	Single room #	Double room #	Quad room #
0	2	台北...	518	1251	2122	29	21	13
⋮								
1499	3	高雄	1062	2070	3137	20	29	24

(b) User (使用者)

User ID	Password
dadala	mumimumi

(c) Order (訂單)

Order ID	User ID	Plan ID
2	dadala	5

(d) Plan (書籤, 也可是訂房計畫)

(date: number of milliseconds since January 1, 1970, 00:00:00 GMT)

Plan ID	Single room #	Double room #	Quad room #	Check in date	Check out date	Hotel ID	User ID
5	3	3	1	2019-12-5	2019-12-7	3	dadala

(e) Search (搜尋紀錄)

User ID	Lower star	Higher star	Address
dadala	2	3	台北...

(f) Schedule (每一間飯店有一張的行程表)

Date	Single room #	Double room #	Quad room #
2019-12-1	23	15	20
		⋮	
2020-6-30	23	15	20

(g) Comment (飯店評論)

Hotel ID	comment
0	[爛透了,有夠髒,...]
	⋮
1499	[太貴,態度差,...]

(h) Today (上次登入系統的時間)

Today
2020-1-6

(i) Usermail (使用者信箱(選填))

User ID	Mail
dadala	b06505002@ntu.edu.tw
mumi	shindondon@gmail.com

2. 以Schedule為對象判斷可擴增房間或日期的方式

(a) 固定房型組合，判斷最長日期

CheckInOutDate extendDate (int hotelid, RoomNum rn, CheckInOutDate ck)

- 檢查在當前訂單的房型組合下，所能增加的最長日期範圍

Order		Schedule	sin	dou	qua
Check in	12/5	12/4	3	2	0
Check out	12/6	12/5	3	2	2
(sin,dou,qua)	(3,2,1)	12/6	3	2	2
		12/7	3	2	2

時間擴增：12/5 ~ 12/8

(b) 固定日期範圍，判斷最大房型組合

RoomNum extendRoom (int hotelid, RoomNum rn, CheckInOutDate ck)

- 檢查在當前訂單的Check-in-out範圍內，所能擴增的最大房間數量

Order		Schedule	sin	dou	qua
Check in	12/4	12/4	32	53	66
Check out	12/8	12/5	22	5	28
(sin,dou,qua)	(3,2,1)	12/6	3	55	50
		12/7	26	43	6

房間數擴增：(3,5,6)

3. 郵件服務運行狀況



ilovecene25@gmail.com

寄給 我 ▾

您的訂單已成立!!

飯店: 416 地址: 台北松山區基隆路一段8號17樓
 入住日期: 2019-12-10 ~ 2019-12-26
 單人房 2 間 雙人房 0 間 四人房 0 間
 訂單編號: 124
 價格: 35008

請記得來喔母咪!!

...



ilovecene25@gmail.com

寄給 我 ▾

您的訂單已取消!!

飯店編號: 0 地址: 台北民生東路一段28號
 入住日期: 2020-01-23 ~ 2020-01-24
 單人房 1 間 雙人房 0 間 四人房 0 間
 訂單編號: 128
 價格: 518

請記得不要來喔母咪!!

...



ilovecene25@gmail.com

寄給 我 ▾

您的訂單已變更!! 您的新訂單

飯店編號: 0 地址: 台北民生東路一段28號
 入住日期: 2020-01-23 ~ 2020-01-24
 單人房 1 間 雙人房 0 間 四人房 0 間
 訂單編號: 128
 價格: 518

不要搞錯喔母咪!!



ilovecene25@gmail.com

寄給 我 ▾

您的訂單已完成!!

飯店編號: 9 地址: 台北中正區信義路二段73號
 入住日期: 2020-01-01 ~ 2020-01-02
 單人房 1 間 雙人房 0 間 四人房 0 間
 訂單編號: 131
 價格: 851

期待下次光顧喔母咪!!

Appendix GUI

GUI_login



The login interface features a dark blue background with a starry night sky pattern. It includes input fields for '用戶名' (Username) and '密碼' (Password), both with a small eye icon for toggling visibility. Below the inputs are three buttons: '註冊' (Register), '確定' (Confirm), and '訪客登入' (Guest Login).

GUI_search



The search interface has a dark blue background with a starry night sky pattern. It includes a search bar with filters for '星級' (Star Rating), '價格' (Price), '人數' (Number of People), '單人房' (Single Room), '雙人房' (Double Room), and '四人房' (Four-person Room). Below the search bar are input fields for '入住日期' (Check-in Date), '退房日期' (Check-out Date), and '縣市' (City/County). A '搜尋' (Search) button is located to the right of the '縣市' field. Below the search bar is a table with the following columns: Hotel, Check in/out date, Room Plan, and Price. The table contains 33 rows of hotel listings. To the right of the table are four buttons: '以價格排列' (Sort by Price), '以星級排列' (Sort by Star Rating), '以飯店排列' (Sort by Hotel), and '加入書籤' (Add to Bookmarks). At the bottom right are two buttons: '使用者' (User) and '確定' (Confirm).

Hotel	Check in/out date	Room Plan	Price
ID: 0 Star: 2 Address: 台北民生東路...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	3108
ID: 1 Star: 4 Address: 台北中山北路...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	11214
ID: 2 Star: 2 Address: 台北西門町成...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	3912
ID: 3 Star: 4 Address: 台北重慶南路...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	9522
ID: 4 Star: 3 Address: 台北中正區重...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	5178
ID: 5 Star: 3 Address: 台北復興南路...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	6462
ID: 6 Star: 3 Address: 台北中正區中...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	7086
ID: 7 Star: 2 Address: 台北中正區寶...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	5748
ID: 8 Star: 3 Address: 台北中正區青...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	5988
ID: 9 Star: 2 Address: 台北中正區信...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	5106
ID: 10 Star: 4 Address: 台北信義區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	8784
ID: 12 Star: 3 Address: 台北忠孝東...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	7092
ID: 13 Star: 2 Address: 台北大安區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	4380
ID: 14 Star: 3 Address: 台北中正區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	7788
ID: 15 Star: 4 Address: 台北中山區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	9570
ID: 16 Star: 3 Address: 台北大安區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	6480
ID: 17 Star: 2 Address: 台北大安區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	4164
ID: 19 Star: 4 Address: 台北北投區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	9558
ID: 20 Star: 3 Address: 台北林森北...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	5262
ID: 21 Star: 3 Address: 台北北投區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	4638
ID: 22 Star: 4 Address: 台北北投區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	8766
ID: 23 Star: 3 Address: 台北民生東...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	5502
ID: 24 Star: 3 Address: 台北大同區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	7812
ID: 25 Star: 4 Address: 台北萬華區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	9630
ID: 26 Star: 3 Address: 台北鬆仁大...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	7158
ID: 28 Star: 4 Address: 台北中華路...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	9978
ID: 29 Star: 3 Address: 台北中正區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	7680
ID: 31 Star: 3 Address: 台北中正區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	4272
ID: 32 Star: 4 Address: 台北南京東...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	10104
ID: 33 Star: 4 Address: 台北信義區...	checkin: 2020-01-13checkout: 202...	單人房數量: 2 雙人房數量: 0 四人...	11520

GUI_hotelpage

飯店資訊	
飯店號	416
星級	2
地址	台北松山區基隆路一段8號17樓
原房數	單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, ...
原總價	35008

評論...

評論

view on google map

單人房 雙人房 四人房

總價: 35008

使用者

確定

GUI_order_confirm

訂單資訊	
用戶	x
訂單號	125
飯店號	16
星級	3
地址	台北大同區民生西路198號2樓
原日期	checkin: 2020-01-13 checkout: 2020-01-16da...
原訂房	單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, ...

入住日期 ... 退房日期 ...

總價: 6480

使用者

確定

GUI_order_manage

已完成訂單

Planid: 102, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 416 Star: 2 Address: 台北松山區基隆路一段8號17樓 size:1 price:10...

刪除訂單

修改訂單

使用者

旅館資訊

GUI_record

搜尋紀錄

書籤

紀錄

Planid: 12, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 416 Star: 2 Address: 台北松山區基隆路一段8號17樓 size:1 price:1094 size...
Planid: 14, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 436 Star: 2 Address: 台北基隆路二段15之2號 size:1 price:1000 size:2 pric...
Planid: 16, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 441 Star: 2 Address: 台北基隆路二段177號 size:1 price:992 size:2 price:1...
Planid: 19, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 6 Star: 3 Address: 台北中正區中華路二段1號 size:1 price:1181 size:2 price...
Planid: 22, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 49 Star: 4 Address: 台北南京東路一段9號 size:1 price:1940 size:2 price:29...
Planid: 24, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 54 Star: 3 Address: 台北大安區仁愛路三段160號 size:1 price:967 size:2 pri...
Planid: 44, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 416 Star: 2 Address: 台北松山區基隆路一段8號17樓 size:1 price:1094 size...
Planid: 67, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 4 Star: 3 Address: 台北中正區重慶南路一段86號7-11樓 size:1 price:863 si...
Planid: 70, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 129 Star: 3 Address: 台北北投區中山路32號 size:1 price:782 size:2 price:1...
Planid: 72, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 129 Star: 3 Address: 台北北投區中山路32號 size:1 price:782 size:2 price:1...
Planid: 74, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 129 Star: 3 Address: 台北北投區中山路32號 size:1 price:782 size:2 price:1...
Planid: 82, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 129 Star: 3 Address: 台北北投區中山路32號 size:1 price:782 size:2 price:1...
Planid: 89, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 4 Star: 3 Address: 台北中正區重慶南路一段86號7-11樓 size:1 price:863 si...
Planid: 102, 單人房數量: 2, 雙人房數量: 0, 四人房數量: 0, 總人數: 2, 飯店資訊ID: 416 Star: 2 Address: 台北松山區基隆路一段8號17樓 size:1 price:1094 siz...

刪除

使用者

前往

GUI_user



GUI_user_data

The image shows a user interface for 'GUI_user_data'. It features a dark blue background with a starry night sky pattern. The form contains three input fields: 'Email (Optional)', '舊密碼' (Old Password), and '新密碼' (New Password). Below the input fields are two buttons: '取消' (Cancel) and '確定' (Confirm). The labels for the input fields are in Chinese and English. The buttons are white with black text.

