

# Software Design: Homework6

目錄(可點擊跳轉)

目錄(可點擊跳轉)	1
Homework Number: #1	1
Homework Number: #2	8
Homework Number: #3-1	18
Homework Number: #3-2	25
Homework Number: #4	30
Homework Number: #5	36
Homework Number: #6	40
UI 設計頁面	46

Homework Number: #1  
Homework Start Date: 20250307  
Homework Due Date: 20250313  
Homework Submitted Date: 20250312  
Student: 軟體三 賴筱喻  
Student ID: 411177031

Answer:

角色
<ul style="list-style-type: none"><li>顧客: 下單、付款、查詢訂單狀態</li><li>店員: 接收訂單、準備飲料、標記訂單完成</li></ul>

- 系統:處理訂單、支付、通知店員與顧客

## 功能需求

### 1. 點餐功能

- a. 顧客可以選擇飲品、甜度、冰量、加料等
- b. 可加購或刪除餐點

### 2. 付款功能

- a. 支援現金、信用卡、行動支付(如 Apple Pay)
- b. 訂單成立後, 付款資訊需儲存

### 3. 訂單管理

- a. 店員可查看新訂單並開始製作
- b. 訂單有不同狀態(待製作、製作中、完成、已取餐)
- c. 店員有權限修改訂單(更改訂單狀態、換飲品、退款)

### 4. 通知功能

- a. 訂單完成後, 顧客會收到通知(如螢幕叫號、簡訊通知)

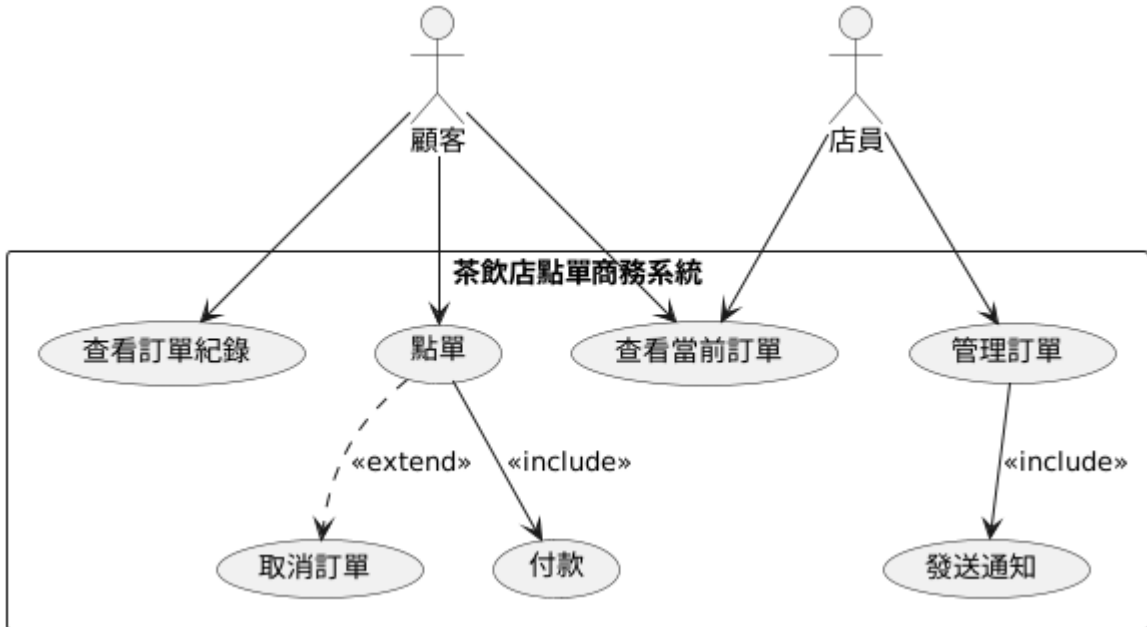
### 5. 歷史訂單查詢

- a. 顧客可以查詢過去的訂單紀錄

Methods	My Outcomes	
draw.io		
	Items	Content
	SharedFile	<a href="#">draw.io: HW1</a>
	Type	Use Case

	<div data-bbox="407 202 600 1062"> <p><b>Diagram</b></p> </div> <div data-bbox="622 210 1525 1042"> <pre> graph LR     subgraph "茶飲店點單商務系統"         direction TB         subgraph Actors             direction TB             C[顧客]             S[店員]         end         subgraph UseCases             direction TB             U1((查看訂單紀錄))             U2((查看當前訂單))             U3((點單))             U4((管理訂單))             U5((付款))             U6((取消訂單))             U7((發送訂單完成通知))         end         C --&gt; U1         C --&gt; U2         C --&gt; U3         S --&gt; U4         U3 -.-&gt; &lt;&lt;include&gt;&gt;  U5         U3 -.-&gt; &lt;&lt;extend&gt;&gt;  U6         U4 -.-&gt; &lt;&lt;include&gt;&gt;  U7     end </pre> </div>						
	<div data-bbox="407 1070 600 1145"> <p><b>MyLearning</b></p> </div> <div data-bbox="622 1070 1991 1145"> <ul style="list-style-type: none"> <li>● 在畫圖過程中了解許多茶飲店的線上系統</li> <li>● 了解到漸進式網路應用程式(PWA)的實際應用, 如: 迷克夏的迷點系統</li> </ul> </div>						
<p>PlantUML</p>	<table border="1"> <thead> <tr> <th>Items</th><th>Content</th></tr> </thead> <tbody> <tr> <td>SharedFile</td><td><a href="#">PlantUML: HW1</a></td></tr> <tr> <td>Type</td><td>Use Case</td></tr> </tbody> </table>	Items	Content	SharedFile	<a href="#">PlantUML: HW1</a>	Type	Use Case
Items	Content						
SharedFile	<a href="#">PlantUML: HW1</a>						
Type	Use Case						

	<b>Code</b>	<pre>@startuml  actor 顧客 actor 店員  rectangle "茶飲店點單商務系統" {     usecase "點單" as Order     usecase "付款" as Payment     usecase "取消訂單" as CancelOrder     usecase "查看當前訂單" as ViewCurrentOrder     usecase "查看訂單紀錄" as ViewOrderHistory     usecase "管理訂單" as ManageOrder     usecase "發送通知" as SendNotification      顧客 --&gt; Order     Order --&gt; Payment : &lt;&lt;include&gt;&gt;     Order ..&gt; CancelOrder : &lt;&lt;extend&gt;&gt;     顧客 --&gt; ViewCurrentOrder     顧客 --&gt; ViewOrderHistory      店員 --&gt; ViewCurrentOrder     店員 --&gt; ManageOrder     ManageOrder --&gt; SendNotification : &lt;&lt;include&gt;&gt; }  @enduml</pre>
--	-------------	--

	<div>Diagram</div> <div></div>										
	<div>MyLearning</div> <div><ul style="list-style-type: none"><li>● 第一次用 plantUML 畫圖，熟練的話畫圖效率會很高</li></ul></div>										
ChatGPT	<table><tr><th>Items</th><th>Content</th></tr><tr><td>SharedFile</td><td><a href="#">ChatGPT: HW1</a></td></tr><tr><td>Type</td><td>Use Case</td></tr><tr><td>ChatGPT Version</td><td>GPT-4o</td></tr><tr><td>ChatGPT Prompt</td><td>1. 點餐功能 顧客可以選擇飲品、甜度、冰量、加料等 可加購或刪除餐點 2. 付款功能 支援現金、信用卡、行動支付 (如 Apple Pay) 訂單成立後, 付款資訊需儲存 3. 訂單管理 店員可查看新訂單並開始製作 訂單有不同狀態(待製作、製作中、完成、已取餐) 店</td></tr></table>	Items	Content	SharedFile	<a href="#">ChatGPT: HW1</a>	Type	Use Case	ChatGPT Version	GPT-4o	ChatGPT Prompt	1. 點餐功能 顧客可以選擇飲品、甜度、冰量、加料等 可加購或刪除餐點 2. 付款功能 支援現金、信用卡、行動支付 (如 Apple Pay) 訂單成立後, 付款資訊需儲存 3. 訂單管理 店員可查看新訂單並開始製作 訂單有不同狀態(待製作、製作中、完成、已取餐) 店
Items	Content										
SharedFile	<a href="#">ChatGPT: HW1</a>										
Type	Use Case										
ChatGPT Version	GPT-4o										
ChatGPT Prompt	1. 點餐功能 顧客可以選擇飲品、甜度、冰量、加料等 可加購或刪除餐點 2. 付款功能 支援現金、信用卡、行動支付 (如 Apple Pay) 訂單成立後, 付款資訊需儲存 3. 訂單管理 店員可查看新訂單並開始製作 訂單有不同狀態(待製作、製作中、完成、已取餐) 店										

		<p>員有權限修改訂單(更改訂單狀態、換飲品、退款)</p> <p>4. 通知功能 訂單完成後, 顧客會收到通知(如螢幕叫號、簡訊通知)</p> <p>5. 歷史訂單查詢 顧客可以查詢過去的訂單紀錄</p> <p>幫我根據以上需求寫出一份優化的用plantuml程式碼製作的use case diagram</p>	
	<p><b>ChatGPT Code</b></p>	<pre> @startuml  actor 顧客 actor 店員  usecase "點餐" as U1 usecase "加購/刪除餐點" as U2 usecase "付款" as U3 usecase "訂單管理" as U4 usecase "通知顧客" as U5 usecase "查詢歷史訂單" as U6 usecase "修改訂單" as U7  顧客 --&gt; U1 顧客 --&gt; U2 顧客 --&gt; U3 顧客 --&gt; U6  U1 ..&gt; U2 : &lt;&lt;include&gt;&gt; U1 ..&gt; U3 : &lt;&lt;include&gt;&gt;  店員 --&gt; U4 店員 --&gt; U7 店員 --&gt; U5  U4 ..&gt; U5 : &lt;&lt;include&gt;&gt; </pre>	

		U4 ..> U7 : <<include>>
		@enduml
	Diagram	<pre> graph TD     Customer((顧客))     Staff((店員))     OrderManagement((訂單管理))     PlaceOrder((點餐))     AddRemoveMenu((加購/刪除餐點))     Payment((付款))     CheckHistoryOrder((查詢歷史訂單))     NotifyCustomer((通知顧客))     ModifyOrder((修改訂單))      Customer --&gt; PlaceOrder     Customer --&gt; AddRemoveMenu     Customer --&gt; Payment     Customer --&gt; CheckHistoryOrder     Staff --&gt; OrderManagement     Staff --&gt; NotifyCustomer     Staff --&gt; ModifyOrder      PlaceOrder -.-&gt; «include»  AddRemoveMenu     PlaceOrder -.-&gt; «include»  Payment     OrderManagement -.-&gt; «include»  NotifyCustomer     OrderManagement -.-&gt; «include»  ModifyOrder </pre>
	MyLearning	<ul style="list-style-type: none"> <li>向ChatGPT提出的要求越簡要, 越容易得到接近想像的圖</li> </ul>

Homework Number: #2

Homework Start Date: 20250314

Homework Due Date: 20250314

Homework Submitted Date: 20250314

Student: 軟體三 賴筱喻

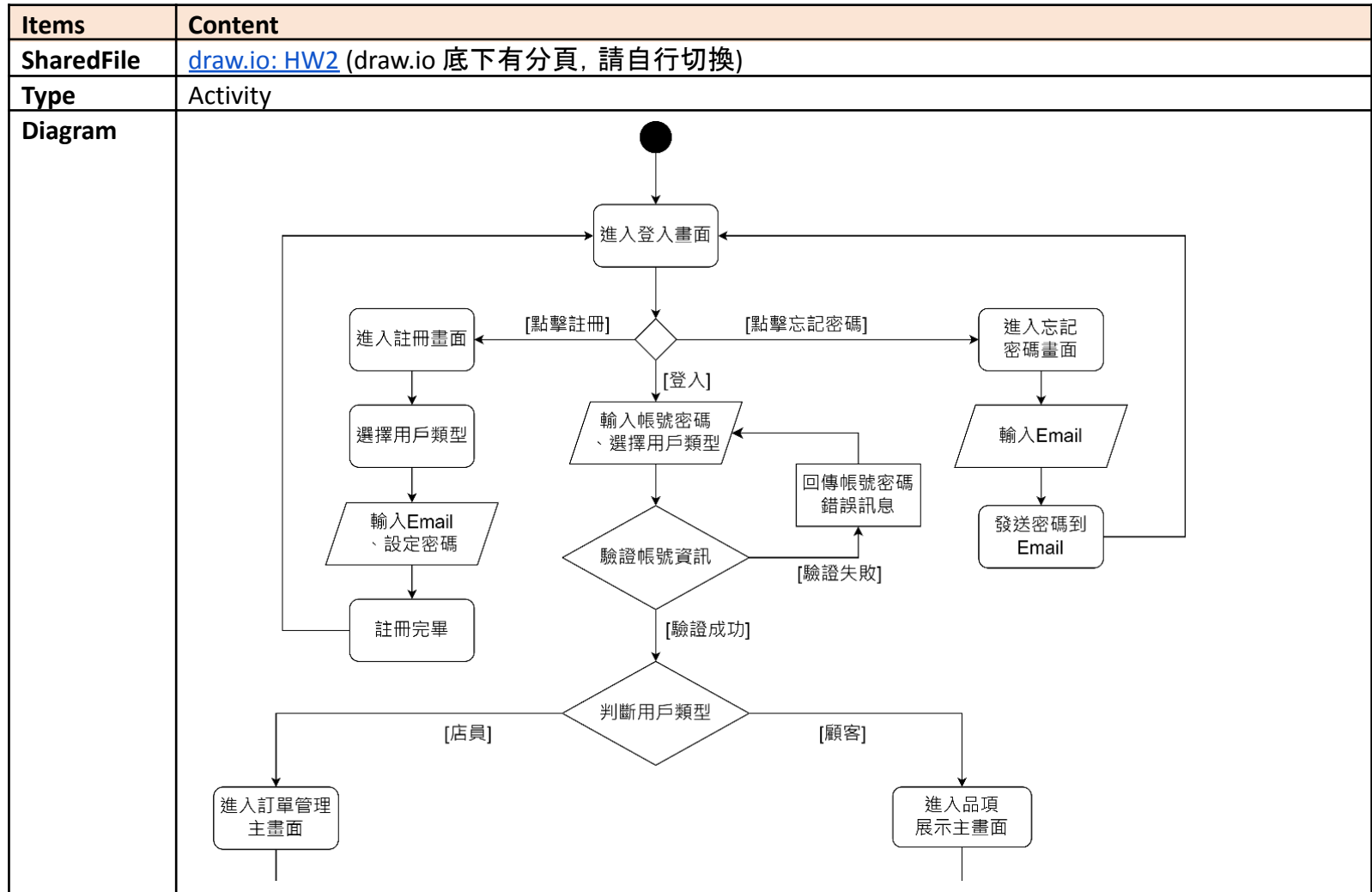
Student ID: 411177031

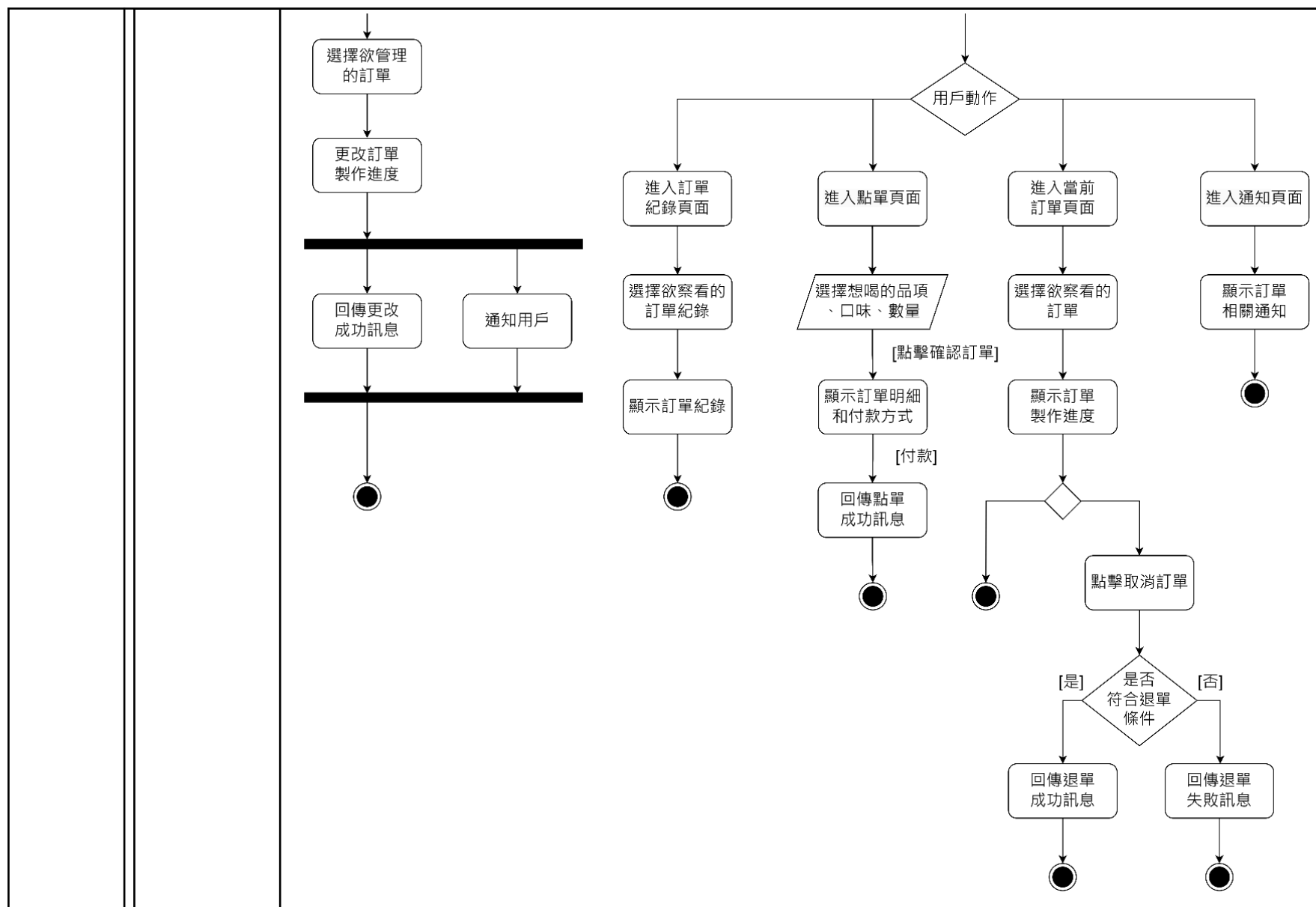
Answer:

Methods	My Outcomes
---------	-------------



draw.io

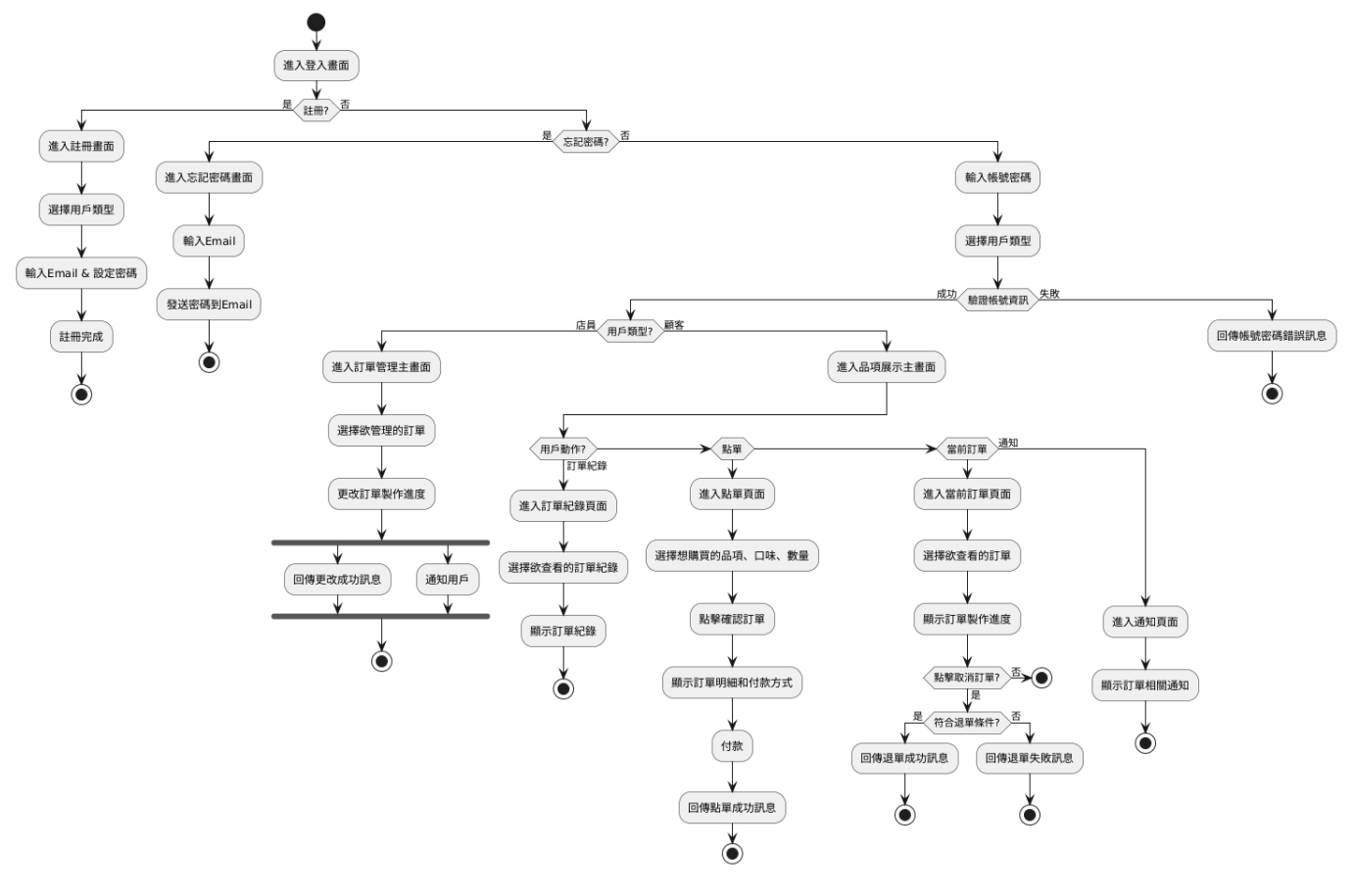




	MyLearning	<ul style="list-style-type: none"> <li>在畫圖過程中再次複習了活動圖的畫法</li> </ul>
PlantUML		
	Items	Content
	SharedFile	<a href="#">PlantUML: HW2</a>
	Type	Activity
	Code	<pre> @startuml start  :進入登入畫面;  if (註冊?) then (是)     :進入註冊畫面;     :選擇用戶類型;     :輸入Email &amp; 設定密碼;     :註冊完成;     stop else (否)     if (忘記密碼?) then (是)         :進入忘記密碼畫面;         :輸入Email;         :發送密碼到Email;         stop     else (否)         :輸入帳號密碼;         :選擇用戶類型;         if (驗證帳號資訊) then (成功)             if (用戶類型?) then (店員)                 :進入訂單管理主畫面;                 :選擇欲管理的訂單;             </pre>

		<pre>:更改訂單製作進度; fork   :回傳更改成功訊息; fork again   :通知用戶; end fork stop else (顧客) :進入品項展示主畫面;   if (用戶動作?) then (訂單紀錄)     :進入訂單紀錄頁面;     :選擇欲查看的訂單紀錄;     :顯示訂單紀錄;     stop   else if (點單)     :進入點單頁面;     :選擇想購買的品項、口味、數量;     :點擊確認訂單;     :顯示訂單明細和付款方式;     :付款;     :回傳點單成功訊息;     stop   else if (當前訂單)     :進入當前訂單頁面;     :選擇欲查看的訂單;     :顯示訂單製作進度;     if (點擊取消訂單?) then (是)       if (符合退單條件?) then (是)         :回傳退單成功訊息;       stop     end if   end if end if</pre>
--	--	--

		<pre>else (否) :回傳退單失敗訊息; stop endif else (否) stop endif else (通知) :進入通知頁面; :顯示訂單相關通知; stop endif endif else (失敗) :回傳帳號密碼錯誤訊息; stop endif endif endif @enduml</pre>
--	--	--



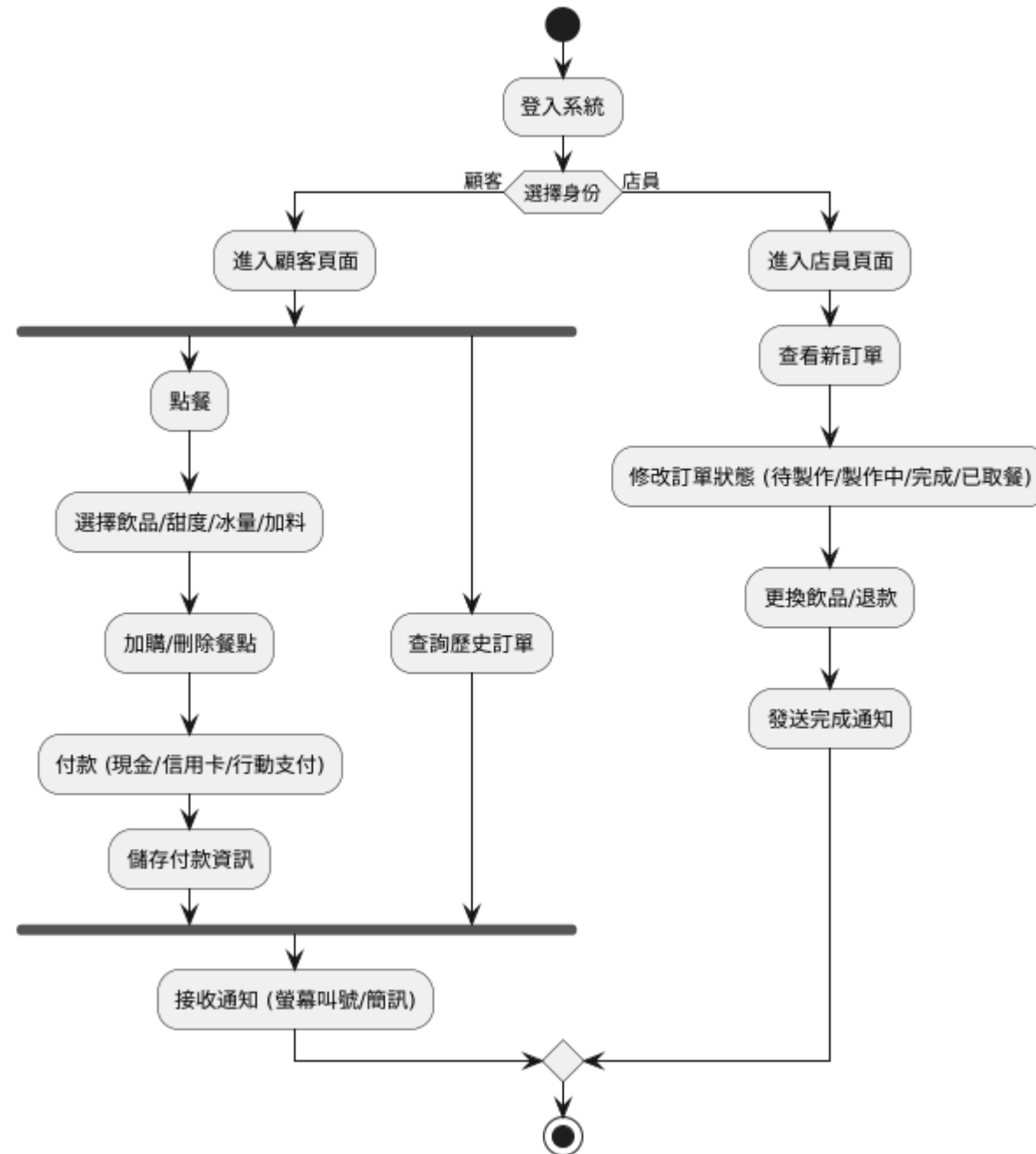
- 練習了 plantUML 畫活動圖的語法，有點難

ChatGPT		
	<b>Items</b>	<b>Content</b>
	<b>SharedFile</b>	<a href="#">ChatGPT: HW2</a>
	<b>Type</b>	Activity
	<b>ChatGPT Version</b>	GPT-4o
	<b>ChatGPT Prompt</b>	<p>幫我根據以下需求寫出一份優化的用plantuml程式碼製作的Activity diagram, Activity diagram中, 顧客和店員能在登入時選擇自己的身份, 並進入不同的頁面, 使用身份對應的功能。</p> <ol style="list-style-type: none"> <li>1.點餐功能 顧客可以選擇飲品、甜度、冰量、加料等 可加購或刪除餐點</li> <li>2.付款功能 支援現金、信用卡、行動支付(如 Apple Pay) 訂單成立後, 付款資訊需儲存</li> <li>3.訂單管理 店員可查看新訂單並開始製作 訂單有不同狀態(待製作、製作中、完成、已取餐) 店員有權限修改訂單(更改訂單狀態、換飲品、退款)</li> <li>4. 通知功能 訂單完成後, 顧客會收到通知(如螢幕叫號、簡訊通知)</li> <li>5.歷史訂單查詢 顧客可以查詢過去的訂單紀錄</li> </ol>
	<b>ChatGPT Code</b>	<pre>@startuml  使用者  start :登入系統; if (選擇身份) then (顧客)     :進入顧客頁面; fork     :點餐;</pre>

		<ul style="list-style-type: none"><li>:選擇飲品/甜度/冰量/加料;</li><li>:加購/刪除餐點;</li><li>:付款 (現金/信用卡/行動支付);</li><li>:儲存付款資訊;</li></ul> fork again <ul style="list-style-type: none"><li>:查詢歷史訂單;</li></ul> end fork <ul style="list-style-type: none"><li>:接收通知 (螢幕叫號/簡訊);</li></ul> else (店員) <ul style="list-style-type: none"><li>:進入店員頁面;</li><li>:查看新訂單;</li><li>:修改訂單狀態 (待製作/製作中/完成/已取餐);</li><li>:更換飲品/退款;</li><li>:發送完成通知;</li></ul> endifstop@enduml	
--	--	--	--



Diagram



	<b>MyLearning</b> <ul style="list-style-type: none"> <li>先給出大方向再提出的要求, 更容易得到接近想像的圖</li> </ul>	
--	---	--

Homework Number: #3-1

Homework Start Date: 20250321

Homework Due Date: 20250321

Homework Submitted Date: 20250321

Student: 軟體三 賴筱喻

Student ID: 411177031

Answer:

Methods	My Outcomes	
draw.io		
	<b>Items</b>	<b>Content</b>
	<b>SharedFile</b>	<a href="#">draw.io: HW3</a> (draw.io 底下有分頁, 請自行切換)
	<b>Type</b>	Class

	<b>Diagram</b>	<pre> classDiagram     class Customer {         +customerID         +customerEmail         +customerName         +placeOrder()         +viewOrderHistory()     }     class Order {         +orderID         +customerID         +orderTime         +orderItems         +orderStatus         +paymentStatus         +updateStatus()     }     class Clerk {         +clerkID         +clerkEmail         +clerkName         +manageOrder()     }     class MenuItem {         +itemName         +itemPrice     }     class Payment {         +paymentMethod         +transactionID         +paymentStatus         +processPayment()     }     class Notification {         +notificationContent         +notificationTime         +sendNotification()     }     Customer "1" --&gt; "0..*" Order     Clerk "0..*" --&gt; "0..*" Order     Order "1..*" *-- "1..*" MenuItem     Order "1" *-- "1" Payment     Order "1" o-- "0..*" Notification   </pre>
	<b>MyLearning</b>	<ul style="list-style-type: none"> <li>在畫圖過程中再次複習了類別圖的畫法</li> </ul>
PlantUML	<b>Items</b>	<b>Content</b>
	<b>SharedFile</b>	<a href="#">PlantUML: HW3</a>
	<b>Type</b>	Class
	<b>Code</b>	@startuml

		<pre>class Customer {   + customerID   + customerEmail   + customerName   + placeOrder()   + viewOrderHistory() }</pre>
--	--	---

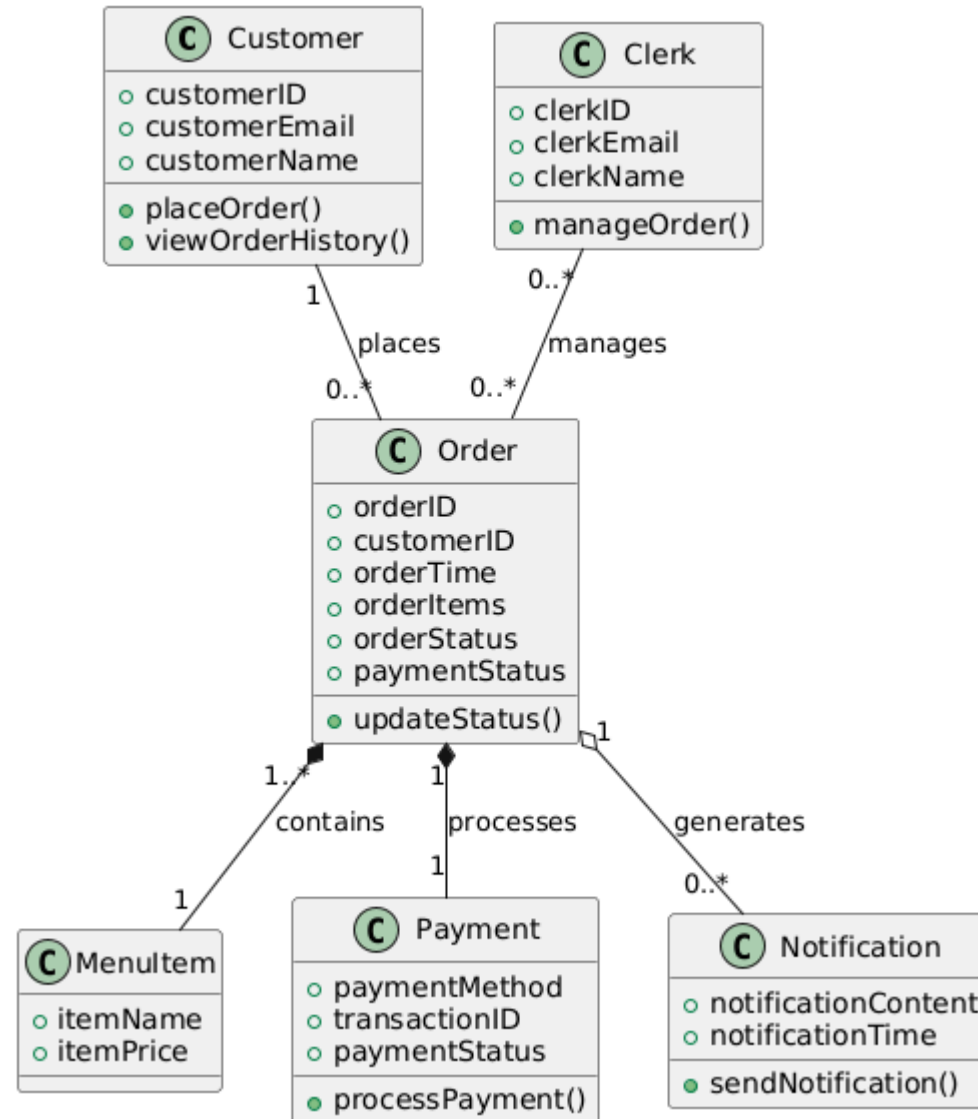
```
class Order {  
  + orderID  
  + customerID  
  + orderTime  
  + orderItems  
  + orderStatus  
  + paymentStatus  
  + updateStatus()  
}
```

```
class Clerk {  
  + clerkID  
  + clerkEmail  
  + clerkName  
  + manageOrder()  
}
```

```
class MenuItem {  
  + itemName  
  + itemPrice  
}
```

	<pre>class Payment { + paymentMethod + transactionID + paymentStatus + processPayment() }  class Notification { + notificationContent + notificationTime + sendNotification() }  ' 關聯關係 Customer "1" -- "0..*" Order : places Clerk "0..*" -- "0..*" Order : manages Order "1..*" *-- "1" MenuItem : contains Order "1" *-- "1" Payment : processes Order "1" o-- "0..*" Notification : generates @enduml</pre>
--	---

Diagram



	<div>MyLearning</div> <div><ul style="list-style-type: none"><li>練習了 plantUML 畫類別圖的語法, 比活動圖簡單多了</li></ul></div>												
ChatGPT	<table><tr><th>Items</th><th>Content</th></tr><tr><td>SharedFile</td><td><a href="#">ChatGPT: HW3</a></td></tr><tr><td>Type</td><td>Class</td></tr><tr><td>ChatGPT Version</td><td>GPT-4o</td></tr><tr><td>ChatGPT Prompt</td><td>幫我根據以下類別寫出一份優化的用plantuml程式碼製作的Class diagram, Class應該包含: Clerk、Customer(方法有OrderHistory)、Order(屬性有OrderStatus)、MenuItem、Payment、Notification, 以及各個 Class 間要有關連箭頭(例如: 多對多、Dependency、Association等)</td></tr><tr><td>ChatGPT Code</td><td>@startuml  class Clerk {     +manageOrder() }  class Customer {     +OrderHistory() }  class Order {     - OrderStatus     +addItem(MenuItem item)     +processPayment(Payment payment) }  class MenuItem {     - name</td></tr></table>	Items	Content	SharedFile	<a href="#">ChatGPT: HW3</a>	Type	Class	ChatGPT Version	GPT-4o	ChatGPT Prompt	幫我根據以下類別寫出一份優化的用plantuml程式碼製作的Class diagram, Class應該包含: Clerk、Customer(方法有OrderHistory)、Order(屬性有OrderStatus)、MenuItem、Payment、Notification, 以及各個 Class 間要有關連箭頭(例如: 多對多、Dependency、Association等)	ChatGPT Code	@startuml  class Clerk { +manageOrder() }  class Customer { +OrderHistory() }  class Order { - OrderStatus +addItem(MenuItem item) +processPayment(Payment payment) }  class MenuItem { - name
Items	Content												
SharedFile	<a href="#">ChatGPT: HW3</a>												
Type	Class												
ChatGPT Version	GPT-4o												
ChatGPT Prompt	幫我根據以下類別寫出一份優化的用plantuml程式碼製作的Class diagram, Class應該包含: Clerk、Customer(方法有OrderHistory)、Order(屬性有OrderStatus)、MenuItem、Payment、Notification, 以及各個 Class 間要有關連箭頭(例如: 多對多、Dependency、Association等)												
ChatGPT Code	@startuml  class Clerk { +manageOrder() }  class Customer { +OrderHistory() }  class Order { - OrderStatus +addItem(MenuItem item) +processPayment(Payment payment) }  class MenuItem { - name												

		<pre>- price }  class Payment {     +process() }  class Notification {     +sendNotification() }  Customer "1" -- "*" Order : places &gt; Order "*" -- "*" MenuItem : contains &gt; Order "1" -- "1" Payment : processes &gt; Clerk "1" -- "*" Order : manages &gt; Order "1" -- "*" Notification : triggers &gt;  @enduml</pre>	
--	--	--	--



	Diagram	<pre> classDiagram     class Clerk {         +manageOrder()     }     class Customer {         +OrderHistory()     }     class Order {         +OrderStatus         +addItem(Menuitem item)         +processPayment(Payment payment)     }     class MenuItem {         +name         +price     }     class Payment {         +process()     }     class Notification {         +sendNotification()     }     Clerk "1" --&gt; "*" Order : manages     Customer "1" --&gt; "*" Order : places     Order "*" --&gt; "*" MenuItem : contains     Order "1" --&gt; "1" Payment : processes     Order "1" --&gt; "*" Notification : triggers   </pre>
	MyLearning	<ul style="list-style-type: none"> <li>先給出大方向再提出的要求, 更容易得到接近想像的圖</li> </ul>

Homework Number: #3-2

Homework Start Date: 20250321

Homework Due Date: 20250321

Homework Submitted Date: 20250321

Student: 軟體三 賴筱喻

Student ID: 411177031

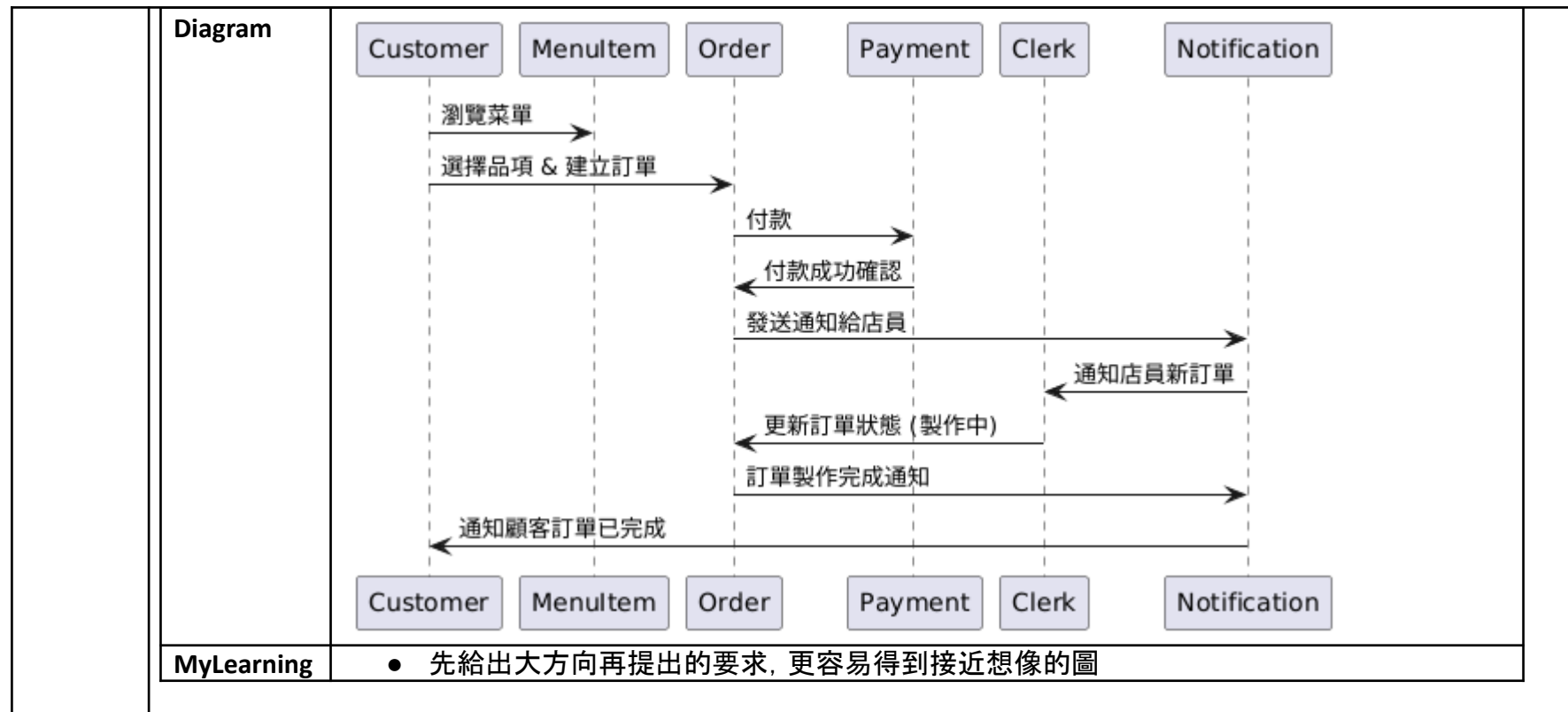
Answer:

Methods	My Outcomes	
draw.io	Items	Content
	SharedFile	<a href="#">draw.io: HW3</a> (draw.io 底下有分頁, 請自行切換)
	Type	Sequence
	Diagram	<pre>sequenceDiagram     participant Customer     participant Order     participant MenuItem     participant Payment     participant Clerk     participant Notification      Customer-&gt;&gt;Order: placeOrder()     activate Order     Order-&gt;&gt;MenuItem: addItem()     activate MenuItem     MenuItem-&gt;&gt;Payment: processPayment()     activate Payment     Payment--&gt;&gt;Order: paymentStatus = "Success"     deactivate Payment     Order-&gt;&gt;Clerk: notifyClerk()     activate Clerk     Clerk-&gt;&gt;Order: manageOrder()     deactivate Clerk     Order-&gt;&gt;Notification: sendNotification()     activate Notification     Notification--&gt;&gt;Customer: notifyOrderStatus()     deactivate Notification     deactivate Order</pre> <p>The diagram is a UML Sequence Diagram titled "茶飲店商務系統點單" (Tea Shop Business System Ordering). It shows the interaction between six lifelines: Customer, Order, MenuItem, Payment, Clerk, and Notification. The process begins with the Customer sending a <code>placeOrder()</code> message to the Order object. The Order object then sends an <code>addItem()</code> message to the MenuItem object. Next, the MenuItem object sends a <code>processPayment()</code> message to the Payment object. The Payment object returns a <code>paymentStatus = "Success"</code> message to the Order object. The Order object then sends a <code>notifyClerk()</code> message to the Clerk object. The Clerk object sends a <code>manageOrder()</code> message back to the Order object. The Order object then sends a <code>sendNotification()</code> message to the Notification object. Finally, the Notification object sends a <code>notifyOrderStatus()</code> message back to the Customer. The diagram uses solid arrows for messages and dashed arrows for return messages. Lifelines are represented by vertical dashed lines, and messages are represented by horizontal arrows between them.</p>
	MyLearning	<ul style="list-style-type: none"><li>在畫圖過程中再次複習了循序圖的畫法</li></ul>

PlantUML		
	<b>Items</b>	<b>Content</b>
	<b>SharedFile</b>	<a href="#">PlantUML: HW3</a>
	<b>Type</b>	Sequence
	<b>Code</b>	<pre> @startuml participant Customer participant Order participant MenuItem participant Payment participant Clerk participant Notification  Customer -&gt; Order : placeOrder() Order -&gt; MenuItem : addItem() Order -&gt; Payment : processPayment() Payment --&gt; Order : paymentStatus = "Success" Order -&gt; Clerk : notifyClerk() Clerk -&gt; Order : manageOrder() Order -&gt; Notification : sendNotification() Notification --&gt; Customer : notifyOrderStatus() @enduml </pre>

	<table><tr><td>Diagram</td><td></td></tr><tr><td>MyLearning</td><td><ul style="list-style-type: none"><li>練習了 plantUML 畫循序圖的語法, 比活動圖簡單多了</li></ul></td></tr></table>	Diagram		MyLearning	<ul style="list-style-type: none"><li>練習了 plantUML 畫循序圖的語法, 比活動圖簡單多了</li></ul>								
Diagram													
MyLearning	<ul style="list-style-type: none"><li>練習了 plantUML 畫循序圖的語法, 比活動圖簡單多了</li></ul>												
ChatGPT	<table><tr><th>Items</th><th>Content</th></tr><tr><td>SharedFile</td><td><a href="#">ChatGPT: HW3</a></td></tr><tr><td>Type</td><td>Sequence</td></tr><tr><td>ChatGPT Version</td><td>GPT-4o</td></tr><tr><td>ChatGPT Prompt</td><td>幫我根據以下類別寫出一份優化的用plantuml程式碼製作的Sequence Diagram, 在前一張Class Diagram中, Class包含: Clerk、Customer、Order、MenuItem、Payment、Notification</td></tr><tr><td>ChatGPT Code</td><td>@startuml participant "Customer" as C</td></tr></table>	Items	Content	SharedFile	<a href="#">ChatGPT: HW3</a>	Type	Sequence	ChatGPT Version	GPT-4o	ChatGPT Prompt	幫我根據以下類別寫出一份優化的用plantuml程式碼製作的Sequence Diagram, 在前一張Class Diagram中, Class包含: Clerk、Customer、Order、MenuItem、Payment、Notification	ChatGPT Code	@startuml participant "Customer" as C
Items	Content												
SharedFile	<a href="#">ChatGPT: HW3</a>												
Type	Sequence												
ChatGPT Version	GPT-4o												
ChatGPT Prompt	幫我根據以下類別寫出一份優化的用plantuml程式碼製作的Sequence Diagram, 在前一張Class Diagram中, Class包含: Clerk、Customer、Order、MenuItem、Payment、Notification												
ChatGPT Code	@startuml participant "Customer" as C												

		<p>             participant "MenuItem" as M              participant "Order" as O              participant "Payment" as P              participant "Clerk" as K              participant "Notification" as N           </p> <p>             C -&gt; M: 瀏覽菜單              C -&gt; O: 選擇品項 &amp; 建立訂單              O -&gt; P: 付款              P -&gt; O: 付款成功確認              O -&gt; N: 發送通知給店員              N -&gt; K: 通知店員新訂單              K -&gt; O: 更新訂單狀態 (製作中)              O -&gt; N: 訂單製作完成通知              N -&gt; C: 通知顧客訂單已完成              @enduml           </p>	
--	--	---	--



Homework Number: #4

Homework Start Date: 20250328

Homework Due Date: 20250328

Homework Submitted Date: 20250328

Student: 軟體三 賴筱喻

Student ID: 411177031

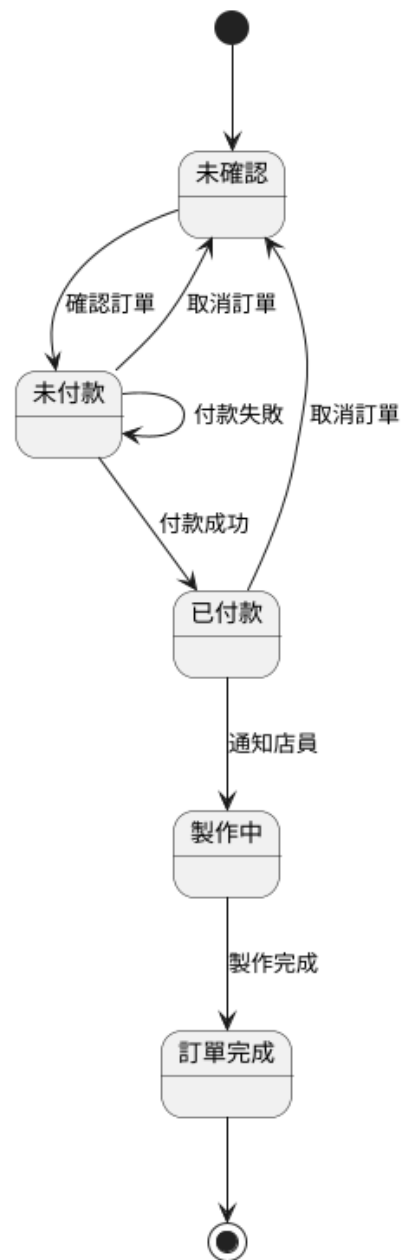
Answer:

Methods	My Outcomes	
draw.io	<b>Items</b>	<b>Content</b>
	<b>SharedFile</b>	<a href="#">draw.io: HW4</a> (draw.io 底下有分頁, 請自行切換)
	<b>Type</b>	State
	<b>Diagram</b>	<p>狀態機圖：訂單</p> <pre> stateDiagram-v2     [*] --&gt; 未確認     未確認 --&gt; 未付款 : 確認訂單     未付款 --&gt; 已付款 : 付款成功     未付款 --&gt; 未確認 : 取消訂單     已付款 --&gt; 未確認 : 取消訂單     已付款 --&gt; 製作中 : 通知店員     製作中 --&gt; 訂單完成 : 製作完成     訂單完成 --&gt; [*]   </pre>
	<b>MyLearning</b>	<ul style="list-style-type: none"> <li>在畫圖過程中再次複習了狀態圖的畫法</li> </ul>
PlantUML	<b>Items</b>	<b>Content</b>
	<b>SharedFile</b>	<a href="#">PlantUML: HW4</a>
	<b>Type</b>	State
	<b>Code</b>	<pre> @startuml [*] --&gt; 未確認  未確認 --&gt; 未付款 : 確認訂單  未付款 --&gt; 已付款 : 付款成功 未付款 --&gt; 未付款 : 付款失敗 未付款 --&gt; 未確認 : 取消訂單  已付款 --&gt; 未確認 : 取消訂單 已付款 --&gt; 製作中 : 通知店員   </pre>

		製作中 --> 訂單完成 : 製作完成 訂單完成 --> [*] @enduml
--	--	--

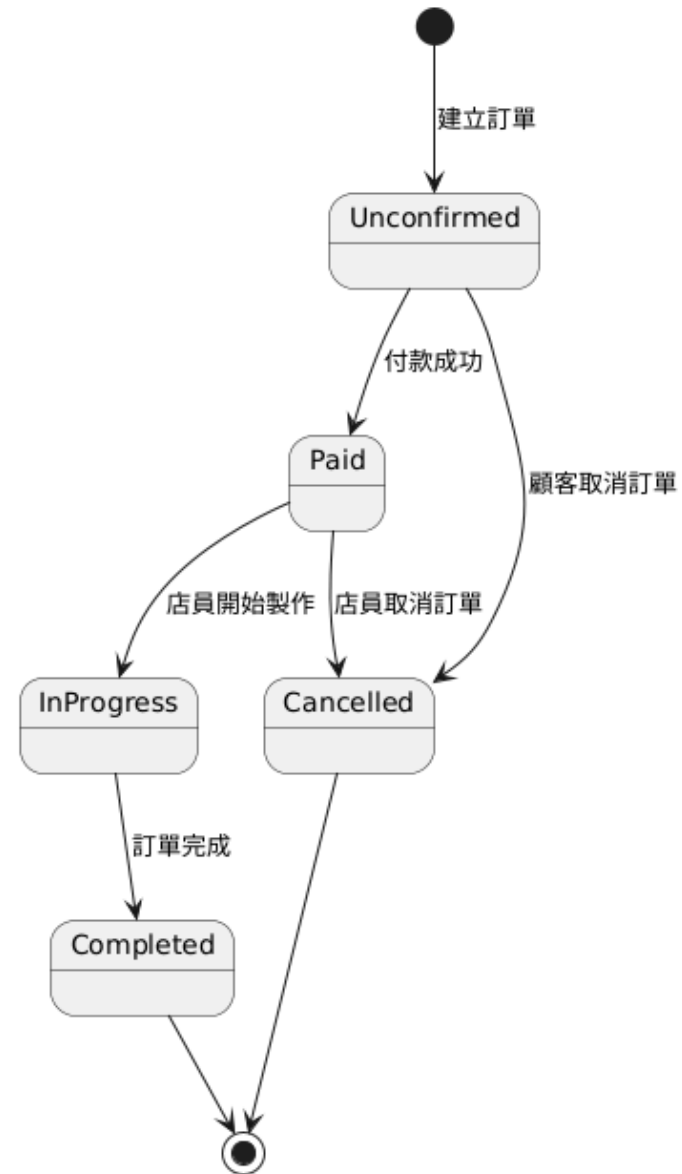


Diagram



	<div>MyLearning</div> <div><ul style="list-style-type: none"><li>練習了 plantUML 畫狀態圖的語法，箭頭線條有點亂</li></ul></div>												
ChatGPT	<table><tr><th>Items</th><th>Content</th></tr><tr><td>SharedFile</td><td><a href="#">ChatGPT: HW4</a></td></tr><tr><td>Type</td><td>State</td></tr><tr><td>ChatGPT Version</td><td>GPT-4o</td></tr><tr><td>ChatGPT Prompt</td><td>幫我針對訂單寫出一份優化的用plantuml程式碼製作的State Diagram, 狀態包含訂單未確認、已付款、製作中、已完成等</td></tr><tr><td>ChatGPT Code</td><td>@startuml [*] --&gt; Unconfirmed : 建立訂單  Unconfirmed --&gt; Paid : 付款成功 Unconfirmed --&gt; Cancelled : 顧客取消訂單  Paid --&gt; InProgress : 店員開始製作 Paid --&gt; Cancelled : 店員取消訂單  InProgress --&gt; Completed : 訂單完成  Cancelled --&gt; [*] Completed --&gt; [*] @enduml</td></tr></table>	Items	Content	SharedFile	<a href="#">ChatGPT: HW4</a>	Type	State	ChatGPT Version	GPT-4o	ChatGPT Prompt	幫我針對訂單寫出一份優化的用plantuml程式碼製作的State Diagram, 狀態包含訂單未確認、已付款、製作中、已完成等	ChatGPT Code	@startuml [*] --> Unconfirmed : 建立訂單  Unconfirmed --> Paid : 付款成功 Unconfirmed --> Cancelled : 顧客取消訂單  Paid --> InProgress : 店員開始製作 Paid --> Cancelled : 店員取消訂單  InProgress --> Completed : 訂單完成  Cancelled --> [*] Completed --> [*] @enduml
Items	Content												
SharedFile	<a href="#">ChatGPT: HW4</a>												
Type	State												
ChatGPT Version	GPT-4o												
ChatGPT Prompt	幫我針對訂單寫出一份優化的用plantuml程式碼製作的State Diagram, 狀態包含訂單未確認、已付款、製作中、已完成等												
ChatGPT Code	@startuml [*] --> Unconfirmed : 建立訂單  Unconfirmed --> Paid : 付款成功 Unconfirmed --> Cancelled : 顧客取消訂單  Paid --> InProgress : 店員開始製作 Paid --> Cancelled : 店員取消訂單  InProgress --> Completed : 訂單完成  Cancelled --> [*] Completed --> [*] @enduml												

Diagram



	MyLearning	<ul style="list-style-type: none"> <li>先給出大方向再提出的要求, 更容易得到接近想像的圖</li> </ul>	
--	------------	---	--

Homework Number: #5

Homework Start Date: 20250411

Homework Due Date: 20250411

Homework Submitted Date: 20250411

Student: 軟體三 賴筱喻

Student ID: 411177031

Answer:

Methods	My Outcomes	
draw.io		
	Items	Content
	SharedFile	<a href="#">draw.io: HW5</a> (draw.io 底下有分頁, 請自行切換)
	Type	Deployment
	Diagram	<pre> graph LR     subgraph ClientDevices [Client Devices]         CU[Customer/ Clerk UI]     end     subgraph WebServer [Web Server]         OC[OrderController]         NS[NotificationService]         PC[PaymentController]     end     subgraph DatabaseServer [Database Server]         OD[OrderDB]     end     ClientDevices &lt;--&gt; HTTP/HTTPS  WebServer     WebServer &lt;--&gt; JDBC  DatabaseServer </pre>

	MyLearning	<ul style="list-style-type: none"> <li>在畫圖過程中再次複習了部署圖的畫法</li> </ul>
PlantUML		
	Items	Content
	SharedFile	<a href="#">PlantUML: HW5</a>
	Type	Deployment
	Code	<pre> @startuml skinparam componentStyle rectangle  node "Client Devices" {     [Customer/Clerk UI] as UI }  node "Web Server" {     component "OrderController" as OC     component "PaymentController" as PC     component "NotificationService" as NS }  node "Database Server" {     database "OrderDB" as DB }  ' UI 呼叫 OrderController &amp; PaymentController UI &lt;--&gt; OC : HTTP/HTTPS UI --&gt; PC : HTTP/HTTPS  ' Controller 呼叫 DB OC &lt;--&gt; DB : JDBC PC --&gt; DB : JDBC </pre>

		<p>' NotificationService 呼叫 DB (例如發送紀錄存檔) NS --&gt; DB : JDBC</p> <p>' OrderController 使用 NotificationService (發送通知) OC --&gt; NS : Internal Call @enduml</p>
	Diagram	<pre> graph TD     subgraph Client_Devices [Client Devices]         CUI[Customer/Clerk UI]     end     subgraph Web_Server [Web Server]         OC[OrderController]         PC[PaymentController]         NS[NotificationService]     end     subgraph Database_Server [Database Server]         OD[(OrderDB)]     end     CUI -- HTTP/HTTPS --&gt; OC     OC -- HTTP/HTTPS --&gt; CUI     OC -- Internal Call --&gt; NS     NS -- JDBC --&gt; OD     PC -- JDBC --&gt; OD     OC -- JDBC --&gt; OD   </pre>
	MyLearning	<ul style="list-style-type: none"> <li>練習了 plantUML 畫部署圖的語法, 比活動圖簡單多了</li> </ul>

ChatGPT		
	<b>Items</b>	<b>Content</b>
	<b>SharedFile</b>	<a href="#">ChatGPT: HW5</a>
	<b>Type</b>	Deployment
	<b>ChatGPT Version</b>	GPT-4o
	<b>ChatGPT Prompt</b>	幫我根據以下架構寫出一份優化的用plantuml程式碼製作茶飲店點餐系統的Deployment diagram，架構應該包含: Client Devices、Web Server、Database Server
	<b>ChatGPT Code</b>	<pre> @startuml node "Client Devices" {     [Mobile App]     [Web Browser] }  node "Web Server" {     component "Order System Backend"     component "Payment Gateway API"     component "Notification Service" }  node "Database Server" {     database "Order DB"     database "User DB" }  ' 連線設定 [Mobile App] --&gt; "Order System Backend" : REST API Calls [Web Browser] --&gt; "Order System Backend" : HTTP Requests  "Order System Backend" --&gt; "Order DB" : Read/Write Orders </pre>

		<p>"Order System Backend" --&gt; "User DB" : Read/Write Users</p> <p>"Order System Backend" --&gt; "Payment Gateway API" : Payment Processing</p> <p>"Order System Backend" --&gt; "Notification Service" : Send Notifications</p> <p>@enduml</p>
	Diagram	<pre> graph TD     subgraph Client_Devices [Client Devices]         MA[Mobile App]         WB[Web Browser]     end     subgraph Web_Server [Web Server]         OSB[Order System Backend]         PGA[Payment Gateway API]         NS[Notification Service]     end     subgraph Database_Server [Database Server]         OD[(Order DB)]         UD[(User DB)]     end     MA -- "REST API Calls" --&gt; OSB     WB -- "HTTP Requests" --&gt; OSB     OSB -- "Payment Processing" --&gt; PGA     OSB -- "Send Notifications" --&gt; NS     OSB -- "Read/Write Orders" --&gt; OD     OSB -- "Read/Write Users" --&gt; UD   </pre> <p>The diagram illustrates the architecture of an order system. It consists of three main components: Client Devices, Web Server, and Database Server. The Client Devices (Mobile App and Web Browser) interact with the Order System Backend on the Web Server via REST API Calls and HTTP Requests. The Order System Backend interacts with the Payment Gateway API for payment processing, the Notification Service for sending notifications, and the Database Server for reading and writing orders and users. The Database Server contains two databases: Order DB and User DB.</p>
	MyLearning	<ul style="list-style-type: none"> <li>先給出大方向再提出的要求, 更容易得到接近想像的圖</li> </ul>

Homework Number: #6

Homework Start Date: 20250502



Homework Due Date: 20250507

Homework Submitted Date: 20250507

Student: 軟體三 賴筱喻

Student ID: 411177031

Answer:

Methods	My Outcomes	
draw.io	<b>Items</b>	<b>Content</b>
	<b>SharedFile</b>	<a href="#">draw.io: HW6</a> (draw.io 底下有分頁, 請自行切換)
	<b>Type</b>	Component
	<b>Diagram</b>	<p>The diagram illustrates the components and their interactions within a '茶飲店商務系統' (Tea Shop Business System). It features four internal components and one external component:</p> <ul style="list-style-type: none"><li><b>«Component» 品項展示</b> (Item Display): Provides the <b>菜單品項</b> (Menu Item) interface to the <b>點單服務</b> component.</li><li><b>«Component» 點單服務</b> (Order Service): Receives the menu item and provides the <b>訂單內容</b> (Order Content) interface to the <b>通知服務</b> and <b>訂單資料管理</b> components.</li><li><b>«Component» 通知服務</b> (Notification Service): Provides the <b>通知內容</b> (Notification Content) interface to the <b>訂單資料管理</b> component.</li><li><b>«Component» 訂單資料管理</b> (Order Data Management): Receives order content and notification content, and provides the <b>支付結果</b> (Payment Result) interface to the external payment platform.</li><li><b>«External» 第三方支付平台</b> (Third-party Payment Platform): Provides the <b>支付資料</b> (Payment Data) interface to the <b>點單服務</b> component and receives the <b>支付結果</b> (Payment Result) from the <b>訂單資料管理</b> component.</li></ul>
	<b>MyLearning</b>	<ul style="list-style-type: none"><li>在畫圖過程中再次複習了元件圖的畫法</li></ul>

PlantUML		
	<b>Items</b>	<b>Content</b>
	<b>SharedFile</b>	<a href="#">PlantUML: HW6</a>
	<b>Type</b>	Component
	<b>Code</b>	<pre> @startuml skinparam componentStyle rectangle  package "茶飲店點餐系統" {     [點單服務] &lt;&lt;Component&gt;&gt;     [通知服務] &lt;&lt;Component&gt;&gt;     [品項展示] &lt;&lt;Component&gt;&gt;     [訂單資料管理] &lt;&lt;Component&gt;&gt; }  [第三方支付平台] &lt;&lt;External&gt;&gt;  ' 點單服務查詢品項 [品項展示] --&gt; [點單服務] ' 點單服務建立訂單 [點單服務] --&gt; [訂單資料管理] ' 點單服務串接第三方付款 [點單服務] --&gt; [第三方支付平台] : 發起付款 ' 付款完成, 通知訂單模組更新狀態 [第三方支付平台] --&gt; [訂單資料管理] : 回傳付款結果 ' 資料管理通知顧客 [訂單資料管理] --&gt; [通知服務] @enduml </pre>

	<div>Diagram</div> <div><div>茶飲店點餐系統</div><div><div>«Component» 品項展示</div><div>«Component» 點單服務</div><div>«Component» 訂單資料管理</div><div>«Component» 通知服務</div></div><div><div>«External» 第三方支付平台</div></div><div><div>發起付款</div><div>回傳付款結果</div></div></div>								
	<div>MyLearning</div> <div><ul style="list-style-type: none"><li>練習了 plantUML 畫元件圖的語法, 比活動圖簡單多了</li></ul></div>								
ChatGPT	<table><tr><th>Items</th><th>Content</th></tr><tr><td>SharedFile</td><td><a href="#">ChatGPT: HW6</a></td></tr><tr><td>Type</td><td>Component</td></tr><tr><td>ChatGPT</td><td>GPT-4o</td></tr></table>	Items	Content	SharedFile	<a href="#">ChatGPT: HW6</a>	Type	Component	ChatGPT	GPT-4o
Items	Content								
SharedFile	<a href="#">ChatGPT: HW6</a>								
Type	Component								
ChatGPT	GPT-4o								

	<b>Version</b>	
	<b>ChatGPT Prompt</b>	幫我根據以下架構寫出一份優化的用plantuml程式碼製作茶飲店點餐系統的Component diagram，元件應該包含五個: 支付管理«Component»、點單服務«Component»、通知服務«Component»、品項展示«Component»、訂單資料管理«Component»，並將他們以箭頭相連。
	<b>ChatGPT Code</b>	<pre> @startuml ' 使用 Skinparam 增強視覺呈現(可省略) skinparam componentStyle rectangle  package "茶飲店點餐系統" {      [點單服務] &lt;&lt;Component&gt;&gt;     [支付管理] &lt;&lt;Component&gt;&gt;     [通知服務] &lt;&lt;Component&gt;&gt;     [品項展示] &lt;&lt;Component&gt;&gt;     [訂單資料管理] &lt;&lt;Component&gt;&gt;      ' 點單服務依賴品項展示與訂單資料管理     [點單服務] --&gt; [品項展示] : 查詢商品資訊     [點單服務] --&gt; [訂單資料管理] : 新增訂單 / 變更狀態      ' 點單服務也會呼叫支付管理     [點單服務] --&gt; [支付管理] : 發起付款請求      ' 訂單資料變更後, 支付與通知都需接收     [訂單資料管理] --&gt; [通知服務] : 發送訂單更新通知     [支付管理] --&gt; [訂單資料管理] : 更新付款狀態 }  @enduml </pre>

	Diagram	<div data-bbox="629 212 1619 1203"> <p>茶飲店點餐系統</p> <pre> graph TD     A["«Component» 點單服務"] -- "發起付款請求" --&gt; B["«Component» 支付管理"]     A -- "查詢商品資訊" --&gt; C["«Component» 品項展示"]     B -- "更新付款狀態" --&gt; D["«Component» 訂單資料管理"]     C -- "新增訂單 / 變更狀態" --&gt; D     D -- "發送訂單更新通知" --&gt; E["«Component» 通知服務"]           </pre> </div>
	MyLearning	<ul style="list-style-type: none"> <li>先給出大方向再提出的要求, 更容易得到接近想像的圖</li> </ul>

## UI 設計頁面

登入頁面	菜單頁面
<div><p>請選擇身份登入</p><p>我是顧客</p><p>我是店員</p></div>	<div><h3>菜單</h3><div><div>珍珠奶茶 - 50元</div><div>加入</div></div><div><div>冬瓜檸檬 - 45元</div><div>加入</div></div><div><div>綠茶 - 30元</div><div>加入</div></div><div><div>點單</div><div>通知</div><div>當前訂單</div><div>訂單紀錄</div></div></div>
點單頁面	當前訂單頁面

## 點餐頁面

珍珠奶茶 - \$50

刪除

綠茶 - \$30

刪除

提交訂單

回首頁

## 當前訂單

訂單編號：ORD1746622792486

訂單時間：2025/5/7 下午8:59:52

狀態：進行中

品項：

- 珍珠奶茶 x 1 ( \$50 )
- 冬瓜檸檬 x 1 ( \$45 )

總金額：\$95

取消訂單

回首頁

通知頁面

訂單紀錄頁面

## 通知中心

### 訂單完成

您的訂單編號 ORD1746605703080 已被標記為「完成」。  
2025/5/7 下午7:43:50

### 訂單完成

您的訂單編號 ORD1746618202189 已被標記為「完成」。  
2025/5/7 下午7:44:09

### 訂單完成

您的訂單編號 ORD1746618218917 已被標記為「完成」。  
2025/5/7 下午7:44:30

### 訂單已完成

您的訂單編號 ORD1746618202189 已被標記為「已完成」。

## 訂單紀錄

訂單編號：ORD1746605703080

狀態：已取消

時間：2025/5/7 下午4:15:03

品項：

- 珍珠奶茶 x1 ( \$50 )

總金額：\$50

訂單編號：ORD1746618202189

狀態：已完成

時間：2025/5/7 下午7:43:22

品項：

- 冬瓜檸檬 x1 ( \$45 )
- 珍珠奶茶 x1 ( \$50 )

總金額：\$95

店員管理頁面



## 店員後台 - 訂單管理

訂單編號：ORD1746622792486

時間：2025/5/7 下午8:59:52

狀態：進行中

品項：

- 珍珠奶茶 x1
- 冬瓜檸檬 x1

總金額：\$95

標記為完成

取消訂單