

Software Design: Homework [Template]

Homework Number: #1

Homework Start Date: 20240322

Homework Due Date: 20240324

Homework Submitted Date: 20240322

Student: 軟體系大三陳昶亨

Student ID: 411077013

Answer:

客人選擇商品，甜度冰塊及客製化選項

系統顯示商品價格以及詢問客人飲料甜度冰塊

系統顯示其他客製化選項

系統計算客人的商品總額並收款

系統列出客人的商品並印出廚吧單

系統以商品數量計算成本及利潤製成日報表和月報表

Hw1---使用案例圖

Methods	My Outcomes	
draw.io	Items	Content
	SharedFile	https://drive.google.com/drive/folders/1hMirzIE9AQT2kAaABqyCXsgcxGSgSju1?usp=sharing
	Type	使用案例圖
	Diagram	<pre> graph LR 客人((客人)) --> UC1(選擇商品及數量) UC1 --> UC2(選擇甜度冰塊) UC2 --> UC3(填寫客製化選項) UC3 --> UC4(付款) 系統((系統)) --> UC5(顯示價格和飲料選項) UC5 --> UC6(顯示其他客製化選項) UC6 --> UC7(計算價錢並收款) 系統 --> UC8(列印廚吧單) UC8 --> UC9(計算成本和利潤製作報表) </pre>
	MyLearning	1. 依照修改過後的 plantuml 畫一次就很快了，而且可以依照美觀進行調整

PlantU ML	Items	Content
	SharedFile	https://drive.google.com/drive/folders/1hMirzIE9AQT2kAaABqyCXGcxGSgSju1?usp=sharing
	Type	使用案例圖
	Code	<pre> @startuml left to right direction actor 客人 as Customer actor 系統 as System rectangle "購物流程" { Customer --> (選擇商品及數量) (選擇商品及數量) --> (選擇甜度冰塊) (選擇甜度冰塊) --> (填寫客製化選項) (填寫客製化選項) --> (付款) System --> (顯示價格和飲料選項) (顯示價格和飲料選項) --> (顯示其他客製化選項) (顯示其他客製化選項) --> (計算商品總額並收款) (計算商品總額並收款) --> (列印出廚吧單) (列印出廚吧單) --> (計算成本和利潤製作報表) } @enduml </pre>

	Diagram <pre> graph LR subgraph "購物流程" direction LR C1(選擇商品及數量) --> C2(選擇甜度冰塊) C2 --> C3(填寫客製化選項) C3 --> C4(付款) C5(顯示價格和飲料選項) --> C6(顯示其他客製化選項) C6 --> C7(計算商品總額並收款) C7 --> C8(列印出廚吧單) C8 --> C9(計算成本和利潤製作報表) end Actor[客人] --> C1 Actor[客人] --> C2 Actor[客人] --> C3 Actor[系統] --> C5 Actor[系統] --> C6 Actor[系統] --> C7 Actor[系統] --> C8 Actor[系統] --> C9 </pre>	
	MyLearning 1. 先用 chatgpt 生成一次，在自己針對錯誤和缺少的地方修改會比較快且較正確	
ChatGPT	Items	Content
	SharedFile	https://drive.google.com/drive/folders/1hMirzIE9AQT2kAaABqyCXSgcxGSgSju1?usp=sharing
	Type	使用案例圖
	ChatGPT Version	V3.5
	ChatGPT Prompt	請以以下需求，用 plantuml 的格式，輸出使用案例圖
	ChatGPT Code	<pre> @startuml left to right direction actor 客人 as Customer actor 系統 as System rectangle "購物流程" { </pre>

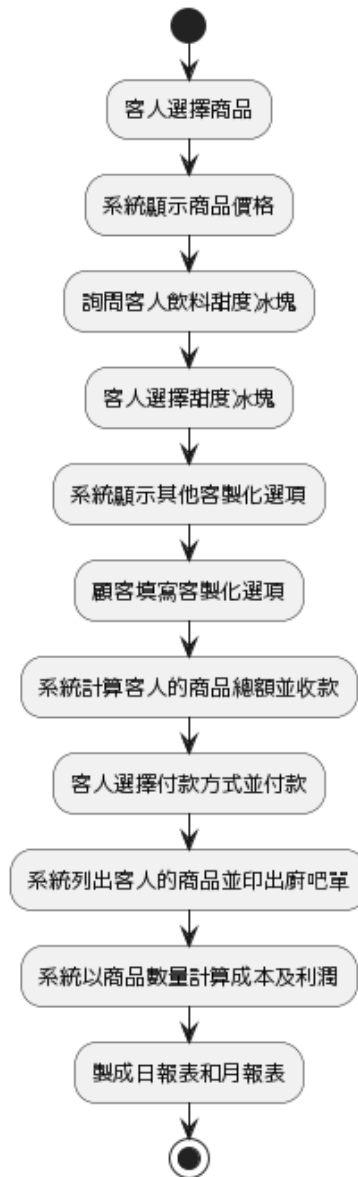
		<p>Customer --> (選擇商品)</p> <p>(選擇商品) --> (顯示價格和詢問飲料選項)</p> <p>(顯示價格和詢問飲料選項) --> (顯示其他客製化選項)</p> <p>(顯示其他客製化選項) --> (計算商品總額並收款)</p> <p>(計算商品總額並收款) --> (列出商品並印出廚吧單)</p> <p>(列出商品並印出廚吧單) --> (計算成本和利潤製作報表)</p> <p>}</p> <p>@enduml</p>
	Diagram	<pre> sequenceDiagram actor 客人 participant 系統 usecase UC1 as 選擇商品 usecase UC2 as 顯示價格和詢問飲料選項 usecase UC3 as 顯示其他客製化選項 usecase UC4 as 計算商品總額並收款 usecase UC5 as 列出商品並印出廚吧單 客人 --> UC1 UC1 --> UC2 UC2 --> UC3 UC3 --> UC4 UC4 --> UC5 </pre>
	MyLearning	<ol style="list-style-type: none"> 1. 善用 chatgpt 可以快速整理出圖表架構 2. Chatgpt 雖然好用，但是正確性不一定，需要再手動檢查修改

Hw2---活動圖

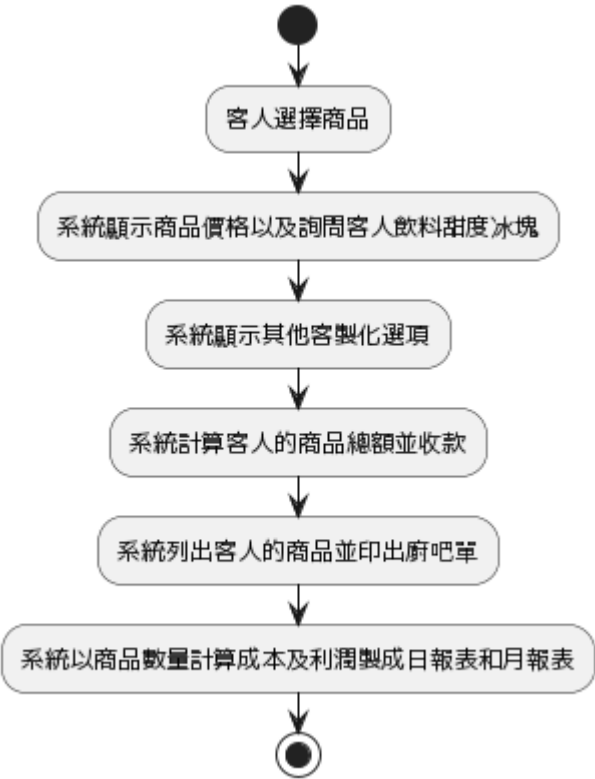
Methods	My Outcomes	
draw.io	Items	Content
	SharedFile	https://drive.google.com/drive/folders/1jMVRO1w0d6tFrPjJ7vIUxChZ3ZKD79Nd?usp=sharing
	Type	活動圖
	Diagram	
	MyLearning	<ol style="list-style-type: none"> 1. 依照修改過後的 plantuml 畫一次就很快了，而且可以依照美觀進行調整 2. Drawio 有泳道圖的功能，可以將顧客和系統的行為區分
PlantUML	Items	Content
	SharedFile	https://drive.google.com/drive/folders/1jMVRO1w0d6tFrPjJ7vIUxChZ3ZKD79Nd?usp=sharing
	Type	活動圖
	Code	@startuml

		<p>start</p> <p>:客人選擇商品;</p> <p>:系統顯示商品價格;</p> <p>:詢問客人飲料甜度冰塊;</p> <p>:客人選擇甜度冰塊;</p> <p>:系統顯示其他客製化選項;</p> <p>:顧客填寫客製化選項;</p> <p>:系統計算客人的商品總額並收款;</p> <p>:客人選擇付款方式並付款;</p> <p>:系統列出客人的商品並印出廚吧單;</p> <p>:系統以商品數量計算成本及利潤;</p> <p>:製成日報表和月報表;</p> <p>stop</p> <p>@enduml</p>
--	--	--

Diagram



	MyLearning	<ol style="list-style-type: none"> 1. 先用 chatgpt 生成一次，在自己針對錯誤和缺少的地方修改會比較快且較正確 2. 在 plantuml 沒有找到泳道圖的相關語法，需要用 drawio 自行修改 3. 詳細的需求要自己新增
ChatGPT	Items	Content
	SharedFile	https://drive.google.com/drive/folders/1jMVRO1w0d6tFrPjJ7vIUxChZ3ZKD79Nd?usp=sharing
	Type	活動圖
	ChatGPT Version	V3.5
	ChatGPT Prompt	請以以下需求，用 plantuml 的格式，輸出活動圖
	ChatGPT Code	<pre> @startuml start :客人選擇商品; :系統顯示商品價格以及詢問客人飲料甜度冰塊; :系統顯示其他客製化選項; :系統計算客人的商品總額並收款; :系統列出客人的商品並印出廚吧單; :系統以商品數量計算成本及利潤製成日報表和月報表; stop @enduml </pre>

	<p>Diagram</p>	 <pre> graph TD Start(()) --> A[客人選擇商品] A --> B[系統顯示商品價格以及詢問客人飲料甜度冰塊] B --> C[系統顯示其他客製化選項] C --> D[系統計算客人的商品總額並收款] D --> E[系統列出客人的商品並印出廚吧單] E --> F[系統以商品數量計算成本及利潤製成日報表和月報表] F --> End((())) </pre>	
	<p>MyLearning</p>	<ol style="list-style-type: none"> 1. 善用 chatgpt 可以快速整理出圖表架構 2. 活動圖的需求比較明確，圖輸出比較正確 	

Hw3---類別圖

Methods	My Outcomes	
draw.io	Items	Content
	SharedFile	https://drive.google.com/drive/folders/1lfKbxnrnwb-f3tZXzJmkl5uEt5wxzE3Y?usp=sharing
	Type	類別圖
	Diagram	<pre> classDiagram class System { displayPrice(item: item): void displayCustomizations(item: item): void calculateTotal(customer: Customer): float } class Customer { name: string selectedItem: list<string> selectPayWay: string<PayWay> chooseItem(item: item): void chooseCustomization(chooseCustomization: chooseCustomization): void choosePayWay(PayWay: string): void } class Item { name: string price: float customization: list<customization> getPrice(): float } class Report { year, month, date: string sale: list<string, int> cost: list<string, int> profit: int generateDailyReport(): void generateMonthlyReport(): void } class Customization { name: string options: list<string> } System --> Customer : generate System --> Report : generate System --> Customization : display Customer --> Item : select Customer --> Customization : select Item --> System : calculate </pre> <p>The diagram illustrates the relationships between five classes: System, Customer, Item, Report, and Customization. The System class has methods for displaying prices, customizations, and calculating totals. The Customer class manages item selection and payment methods. The Item class represents individual products with their prices and customizations. The Report class handles report generation. The Customization class manages product options. Relationships include: System generating reports and displaying customizations; Customer selecting items and customizations; and Item providing data for System calculations.</p>

	MyLearning	1. 依照修改過後的 plantuml 畫一次就很快了，而且可以依照美觀進行調整 2. 格子很多不好拉
PlantUML	Items	Content
	SharedFile	https://drive.google.com/drive/folders/1IfKbxnrnwb-f3tZXzJmkl5uEt5wxzE3Y?usp=sharing
	Type	活動圖
	Code	<pre> @startuml class Customer { - name: string - selectedItems: list<Item> - selectPayWay:string<PayWay> + chooseItem(item: Item): void + chooseCustomization(customization: Customization): void + choosePayWay(PayWay:string):void } class Item { - name: string - price: float - customizations: list<Customization> + getPrice(): float </pre>

```
}
```

```
class Customization {
```

```
    - name: string
```

```
    - options: list<string>
```

```
}
```

```
class System {
```

```
    + displayPrice(item: Item): void
```

```
    + displayCustomizations(item: Item): void
```

```
    + calculateTotal(customer: Customer): float
```

```
}
```

```
class Report {
```

```
    - year,month,date: string
```

```
    - sale: list<string,int>
```

```
    - cost:list<string,int>
```

```
    - profit:<int>
```

```
    + generateDailyReport(): void
```

```
    + generateMonthlyReport(): void
```

```
}
```

```
Customer -- Item : selects >
```

```
Customer -- Customization : selects >
```

		<p>System -- Report : generate ></p> <p>System -- Item : displays ></p> <p>System -- Customization : displays ></p> <p>System -- Customer : calculates, generates ></p> <p>@enduml</p>	
--	--	--	--

	<p>Diagram</p>	<pre> classDiagram class System { +displayPrice(item: Item): void +displayCustomizations(item: Item): void +calculateTotal(customer: Customer): float } class Customer { +name: string +selectedItems: list<Item> +selectPayWay: string<PayWay> +chooseItem(item: Item): void +chooseCustomization(customization: Customization): void +choosePayWay(PayWay: string): void } class Item { +name: string +price: float +customizations: list<Customization> +getPrice(): float } class Customization { +name: string +options: list<string> } class Report { +year,month,date: string +sale: list<string,int> +cost: list<string,int> +profit: <int> +generateDailyReport(): void +generateMonthlyReport(): void } System --> Customer : calculates, generates System --> Report : generate Customer --> Item : selects Customer --> Customization : selects Item --> System : displays Customization --> Report : displays </pre> <p>The diagram illustrates the relationships between five classes: System, Customer, Item, Customization, and Report. Each class has a set of attributes and methods. The relationships are as follows:</p> <ul style="list-style-type: none"> System (Methods: <code>displayPrice(item: Item): void</code>, <code>displayCustomizations(item: Item): void</code>, <code>calculateTotal(customer: Customer): float</code>) <ul style="list-style-type: none"> Calculates and generates data for Customer. Generates data for Report. Customer (Attributes: <code>name: string</code>, <code>selectedItems: list<Item></code>, <code>selectPayWay: string<PayWay></code>; Methods: <code>chooseItem(item: Item): void</code>, <code>chooseCustomization(customization: Customization): void</code>, <code>choosePayWay(PayWay: string): void</code>) <ul style="list-style-type: none"> Selects Item and Customization. Item (Attributes: <code>name: string</code>, <code>price: float</code>, <code>customizations: list<Customization></code>; Method: <code>getPrice(): float</code>) <ul style="list-style-type: none"> Displays data to System. Customization (Attributes: <code>name: string</code>, <code>options: list<string></code>) <ul style="list-style-type: none"> Displays data to Report. Report (Attributes: <code>year,month,date: string</code>, <code>sale: list<string,int></code>, <code>cost: list<string,int></code>, <code>profit: <int></code>; Methods: <code>generateDailyReport(): void</code>, <code>generateMonthlyReport(): void</code>)
	<p>MyLearning</p>	<ol style="list-style-type: none"> 1. 將報表另外列出來成一個類別 2. 按照年月日分別，內容有銷售、成本、利潤 3. 將 custom 內不太適合的東西刪除

ChatGPT	Items	Content
	SharedFile	https://drive.google.com/drive/folders/1lfKbxnrnwb-f3tZXzJmkl5uEt5wxzE3Y?usp=sharing
	Type	類別圖
	ChatGPT Version	V3.5
	ChatGPT Prompt	請以以下需求，用 plantuml 的格式，輸出 Class Diagram
	ChatGPT Code	<pre> @startuml class Customer { - name: string - selectedItems: list<Item> + chooseItem(item: Item): void + chooseCustomization(customization: Customization): void } class Item { - name: string - price: float - customizations: list<Customization> + getPrice(): float } class Customization { </pre>

		<pre>- name: string - options: list<string> } class System { + displayPrice(item: Item): void + displayCustomizations(item: Item): void + calculateTotal(customer: Customer): float + generateReceipt(customer: Customer): void + generateDailyReport(): void + generateMonthlyReport(): void } Customer -- Item : selects > Customer -- Customization : selects > System -- Item : displays > System -- Customization : displays > System -- Customer : calculates, generates > @enduml</pre>
--	--	---

	Diagram	<pre> classDiagram class System { <<C>> displayPrice(item: Item): void displayCustomizations(item: Item): void calculateTotal(customer: Customer): float generateReceipt(customer: Customer): void generateDailyReport(): void generateMonthlyReport(): void } class Customer { <<C>> +name: string +selectedItems: list<Item> chooseItem(item: Item): void chooseCustomization(customization: Customization): void } class Item { <<C>> +name: string +price: float +customizations: list<Customization> getPrice(): float } class Customization { <<C>> +name: string +options: list<string> } System --> Customer : calculates, generates Customer --> System : displays Customer --> Item : selects Item --> Customer : selects Customization --> System : displays System --> Customization : displays </pre> <p>The diagram illustrates the relationships between four classes: System, Customer, Item, and Customization. Each class is marked with a green circle containing a 'C'.</p> <ul style="list-style-type: none"> System Class: <ul style="list-style-type: none"> Methods: <code>displayPrice(item: Item): void</code>, <code>displayCustomizations(item: Item): void</code>, <code>calculateTotal(customer: Customer): float</code>, <code>generateReceipt(customer: Customer): void</code>, <code>generateDailyReport(): void</code>, <code>generateMonthlyReport(): void</code>. Customer Class: <ul style="list-style-type: none"> Attributes: <code>name: string</code>, <code>selectedItems: list<Item></code>. Methods: <code>chooseItem(item: Item): void</code>, <code>chooseCustomization(customization: Customization): void</code>. Item Class: <ul style="list-style-type: none"> Attributes: <code>name: string</code>, <code>price: float</code>, <code>customizations: list<Customization></code>. Method: <code>getPrice(): float</code>. Customization Class: <ul style="list-style-type: none"> Attributes: <code>name: string</code>, <code>options: list<string></code>. <p>Relationships:</p> <ul style="list-style-type: none"> System to Customer: A directed association labeled "calculates, generates". Customer to System: A directed association labeled "displays". Customer to Item: A directed association labeled "selects". Item to Customer: A directed association labeled "selects". Customization to System: A directed association labeled "displays". System to Customization: A directed association labeled "displays".
MyLearnin	1.	善用 chatgpt 可以快速整理出圖表架構

	g	<div>2. Chatgpt 在理解需求上我們不太一樣</div> <div>3. 要將報表另外生成一個類別會比較妥當</div>