

Software Design: Homework #1: Use Case Diagram

Homework Number: #1

Homework Start Date: 20240308

Homework Due Date: 20240310

Homework Submitted Date: 20240308

Student: 軟體系大三 蔡岳哲

Student ID: 411077010

Answer:

Requirements
<ol style="list-style-type: none">1. Customer can add or delete products into shopping cart.2. Customer can browse menu.3. Menu would had the price, name, category of products.4. Customer can order py sending the shopping cart.5. Order system can receive the order after customer send the order.6. Shopping cart can show the total price and change it after added or deleted the products.7. Order system can print the receipt after receiving the order.8. Customer can create its own profile with name, phone, email.9. Customer can login by entering email, phone.10. Customer can track the order.

Keywords:

Shopping cart, menu, order system, customer profile, login, track order.

Use Case Diagram:

Methods	My Outcomes	
draw.io	Items	Content
	SharedFile	https://drive.google.com/file/d/19SPaTe4Vw3s6HVEhGjDncvx8V_gl0rHu/view?usp=sharing
	Type	Use Case Diagram
	Diagram	<pre> graph LR subgraph Order_System [Order System] track_order((track order)) Add_product((Add product)) Delete_product((Delete product)) Show_total_price((Show total price)) send_order((send order)) browse_menu((browse menu)) create_menu_product((create menu's product)) receive_order((receive order)) print_receipt((print the receipt)) create_new_profile((create new profile)) name((name)) phone((phone)) email((email)) login((login)) end Customer[Customer] --> track_order Customer --> Add_product Customer --> Delete_product Customer --> send_order Customer --> browse_menu Order_System[Order System] --> receive_order Order_System --> print_receipt Add_product --> Show_total_price Delete_product --> Show_total_price browse_menu -.-> <<extend>> price((price)) browse_menu -.-> <<extend>> name((name)) browse_menu -.-> <<extend>> category((category)) create_menu_product -.-> <<include>> price create_menu_product -.-> <<include>> name create_menu_product -.-> <<include>> category create_new_profile -.-> <<include>> name create_new_profile -.-> <<include>> phone create_new_profile -.-> <<include>> email name -.-> <<extend>> login((login)) phone -.-> <<extend>> login email -.-> <<extend>> login create_menu_product --> create_menu_product </pre>
	MyLearning	<ol style="list-style-type: none"> 1. How to draw a use case diagram. 2. What actors, use caes, containers means. 3. What extend, include means.

PlantUML	Items	Content
	SharedFile	http://www.plantuml.com/plantuml/dpng/TPBVRjiW5CRIUOgZtcoxg1JYFscAkhKtMzlJGTWS9gWOR3WU5arxTrd6bCSQby1lzy77xPu22Gy3PrX1fy9o87NnnE1qXubQMSP4vASXySXaElGJykRRrwXXwUNGDZTW0Zm2Yvhnk9cOOy6o-1VDWIK8Y1y92_aBt1HVue7_7ZaMNvH2hvxfmP9aznc-HKD4XQ-olxfvCv0Zel1tck9KwBE2OylzM-yQc-o3cZLbMkp6q-q17H2U3saJ_sx7FdftNaSExJ39E-odDzedx4LNQ9yax4KX4VdNvBYC-LHqANbGu3-wcVmxRkSnNCCFMIprbCjrzSPzoTdtYJENP2cyJRp8qxQfluZlrZBBqRkBqzH5Y2RGcg4hGbk2i1B_V4xjbx62pm9PoVClOUB5fLO_pAisZuCdiuVFzIRRI30eFXdXctN5lxLFdnK5_Y6qwh9bioRueeYclYrjsP9LiwXgTwfDQbTUeburpR9v68UXC_y1
	Type	Use Case Diagram

	Code <pre> @startuml left to right direction actor Customer actor "Order System" as OrderSystem rectangle "Order System" { usecase (track order) as UC1 usecase (Add Product) as UC2 usecase (Delete Product) as UC3 usecase (Show total price) as UC4 usecase (receive order) as UC5 usecase (send order) as UC6 usecase (print the receipt) as UC7 usecase (browse menu) as UC8 usecase (price) as UC9 usecase (name) as UC10 usecase (category) as UC11 usecase (create menu's product) as UC12 usecase (create new profile) as UC13 usecase (name) as UC14 usecase (phone) as UC15 usecase (email) as UC16 usecase (login) as UC17 } </pre>
--	--

		<p>Customer --> UC1 Customer --> UC2 Customer --> UC3 Customer --> UC6 Customer --> UC8 Customer --> UC13 Customer --> UC17 OrderSystem --> UC5 OrderSystem --> UC7 OrderSystem --> UC12</p> <p>UC2 --> UC4 UC3 --> UC4</p> <p>UC12 --> UC9 :<<include>> UC12 --> UC10 :<<include>> UC12 --> UC11 :<<include>> UC9 --> UC8 :<<extend>> UC10 --> UC8 :<<extend>> UC11 --> UC8 :<<extend>></p> <p>UC13 --> UC16 :<<include>> UC13 --> UC15 :<<include>> UC13 --> UC14 :<<include>></p>
--	--	---

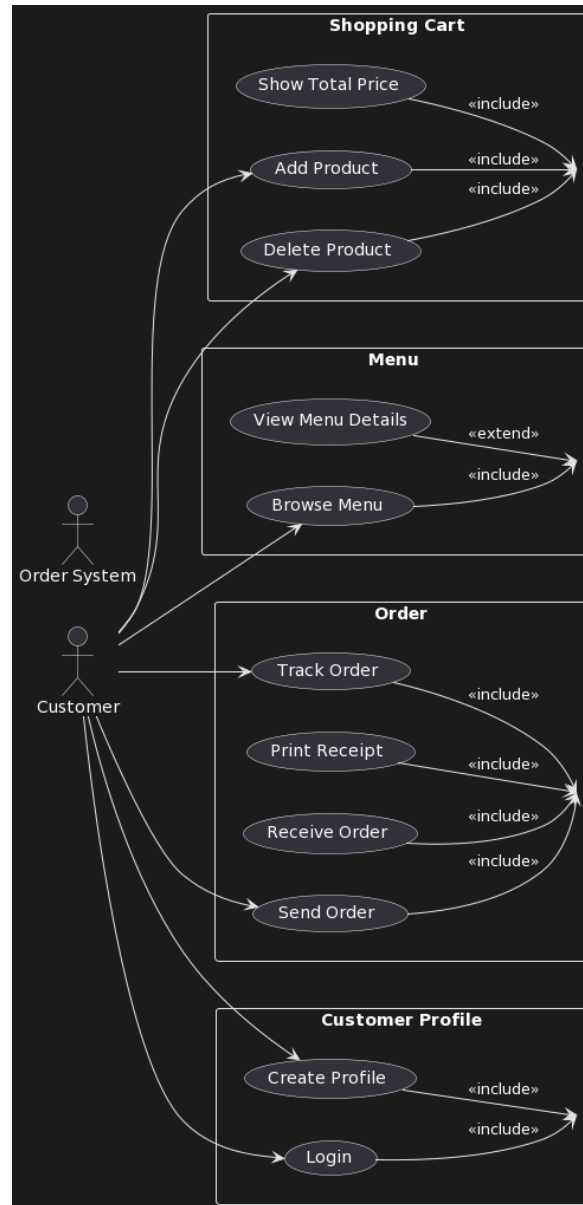
		<p>UC16 --> UC17 :<<extend>></p> <p>UC14 --> UC17 :<<extend>></p> <p>UC15 --> UC17 :<<extend>></p> <p>@enduml</p>
	Diagram	<pre> graph LR Customer((Customer)) subgraph Order_System [Order System] direction TB send_order([send order]) create_new_profile([create new profile]) track_order([track order]) Add_Product([Add Product]) Delete_Product([Delete Product]) receive_order([receive order]) print_receipt([print the receipt]) create_menu_product([create menu's product]) phone([phone]) name1([name]) email([email]) category([category]) name2([name]) price([price]) login([login]) browse_menu([browse menu]) Show_total_price([Show total price]) end Order_System_actor((Order System)) Customer --> send_order Customer --> create_new_profile Customer --> track_order Customer --> Add_Product Customer --> Delete_Product Customer --> login Customer --> browse_menu Order_System_actor --> receive_order Order_System_actor --> print_receipt Order_System_actor --> create_menu_product Order_System_actor --> login Order_System_actor --> browse_menu create_new_profile --> «include» phone create_new_profile --> «include» name1 create_new_profile --> «include» email Add_Product --> «include» Show_total_price create_menu_product --> «include» category create_menu_product --> «include» name2 create_menu_product --> «include» price login --> «extend» phone login --> «extend» name1 login --> «extend» email browse_menu --> «extend» category browse_menu --> «extend» name2 browse_menu --> «extend» price </pre>
	MyLearning	<ol style="list-style-type: none"> 1. How to use plantuml. 2. How to code in plantuml.(using actor, rectangle, usecase, and lines including extend and include.)

PlantUML + ChatGPT	Items	Content
	SharedFile	https://chat.openai.com/share/fc6a8bbb-4978-4986-8a55-27eaa4eba1c6
	Type	Use Case Diagram
	ChatGPT Version	V3.5
	ChatGPT Prompt	<p>Write an UML Use Case diagram and use PlantUML script.</p> <ol style="list-style-type: none"> 1. Customer can add or delete products into shopping cart. 2. Customer can browse menu. 3. Menu would had the price, name, category of products. 4. Customer can order py sending the shopping cart. 5. Order system can receive the order after customer send the order. 6. Shopping cart can show the total price and change it after added or deleted the products. 7. Order system can print the receipt after receiving the order. 8. Customer can create its own profile with name, phone, email. 9. Customer can login by entering email, phone. 10. Customer can track the order.

	ChatGPT Code	<pre>@startuml left to right direction actor Customer actor "Order System" as OrderSystem rectangle "Shopping Cart" { usecase (Add Product) as UC1 usecase (Delete Product) as UC2 usecase (Show Total Price) as UC6 } rectangle "Menu" { usecase (Browse Menu) as UC3 usecase (View Menu Details) as UC11 } rectangle "Order" { usecase (Send Order) as UC4 usecase (Receive Order) as UC5 usecase (Print Receipt) as UC7 usecase (Track Order) as UC10 } rectangle "Customer Profile" {</pre>
--	-------------------------	--

		<pre>usecase (Create Profile) as UC8 usecase (Login) as UC9 } Customer --> UC1 Customer --> UC2 Customer --> UC3 Customer --> UC4 Customer --> UC8 Customer --> UC9 Customer --> UC10 UC1 --> "Shopping Cart" : <<include>> UC2 --> "Shopping Cart" : <<include>> UC3 --> "Menu" : <<include>> UC4 --> "Order" : <<include>> UC5 --> "Order" : <<include>> UC6 --> "Shopping Cart" : <<include>> UC7 --> "Order" : <<include>> UC8 --> "Customer Profile" : <<include>> UC9 --> "Customer Profile" : <<include>> UC10 --> "Order" : <<include>> UC11 --> "Menu" : <<extend>> @enduml</pre>
--	--	---

Diagram



	MyLearning	<ol style="list-style-type: none">1. How to ask question so that GPT can generate the code I need.2. That GPT can generate the umlcode very fast.3. That GPT's code would not be always properly to my requirements.
--	-------------------	--

Software Design: Homework #2: Activity Diagram

Homework Number: #2

Homework Start Date: 20240315

Homework Due Date: 20240317

Homework Submitted Date: 20240315

Student: 軟體系大三 蔡岳哲

Student ID: 411077010

Answer:

Requirements
<ol style="list-style-type: none">1. Customer can add or delete products into shopping cart.2. Customer can browse menu.3. Menu would had the price, name, category of products.4. Customer can order py sending the shopping cart.5. Order system can receive the order after customer send the order.6. Shopping cart can show the total price and change it after added or deleted the products.7. Order system can print the receipt after receiving the order.8. Customer can create its own profile with name, phone, email.9. Customer can login by entering email, phone.10. Customer can track the order.

Keywords:

Shopping cart, menu, order system, customer profile, login, track order.

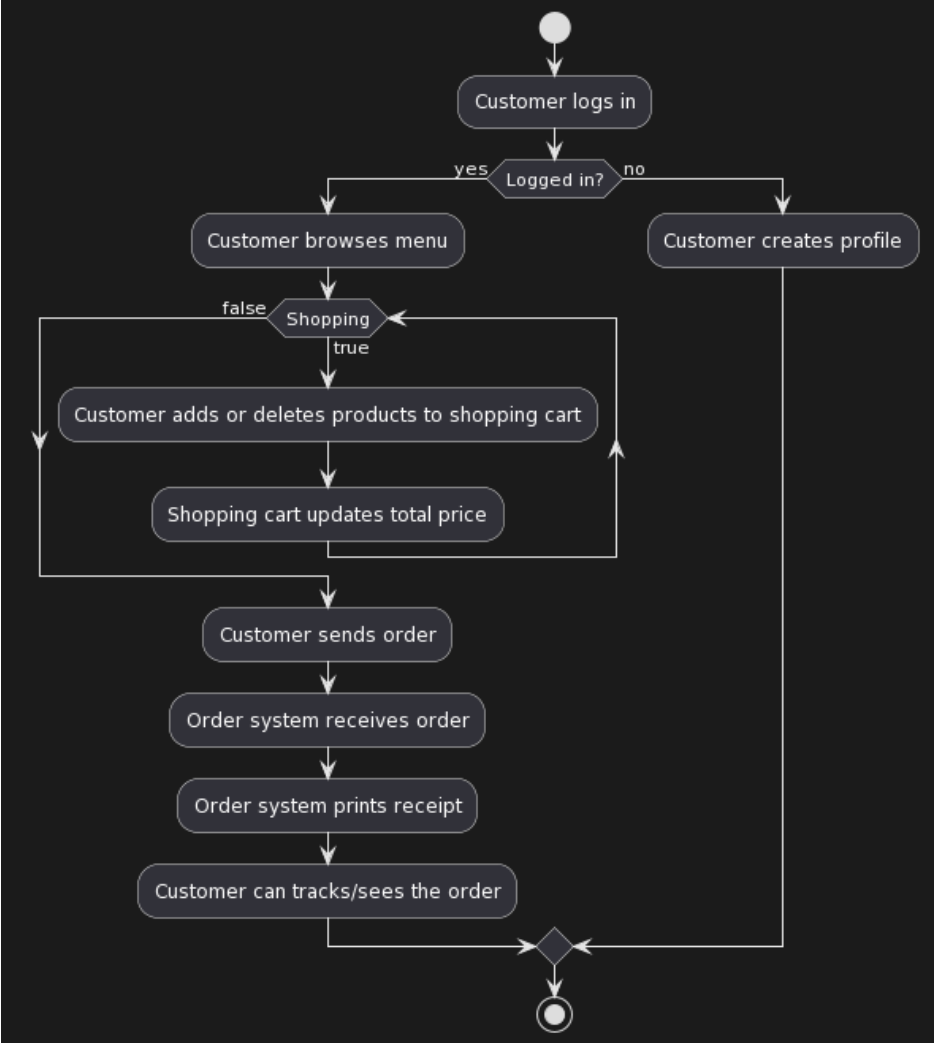
Activity Diagram:

Methods	My Outcomes	
draw.io	Items	Content
	SharedFile	https://drive.google.com/file/d/1hPHPKBH-Nkox1a1X_Q_4GcYuxEVbY66/view?usp=sharing
	Type	Activity Diagram

	<p>Diagram</p>	<pre> graph TD Start(()) --> Login[Customer login] Login --> Dec1{Already had account?} Dec1 -- Yes --> LoginSucc[Login successfully] Dec1 -- No --> Dec2{Create account?} Dec2 -- Yes --> CreateAcc[Create account] CreateAcc --> Login Dec2 -- No --> Shopping[Shopping] LoginSucc --> Browse[Shopping/Browse menu] Browse --> Shopping Shopping --> AddProd[Add or delete product from shopping cart] AddProd --> UpdatePrice[Shopping cart update total price in time] UpdatePrice --> Dec3{Finish shopping?} Dec3 -- No --> Shopping Dec3 -- Yes --> SendOrder[Send order] SendOrder --> SysRecv[System receive order] SysRecv --> SysPrint[System print receipt] SysPrint --> TrackInfo[Customer can tracks/sees order info] TrackInfo --> End((())) Dec2 -- No --> End </pre>
	<p>MyLearning</p>	<ol style="list-style-type: none"> 1. How to draw a activity diagram. 2. What actors, use caes, containers means. 3. What extend, include means.

PlantUML	Items	Content
	SharedFile	https://www.plantuml.com/plantuml/dpng/TOynRiCm34LtdOAZFNMIxmReMg13Jg1AtxPGMH98ga5kNobnqNZ8Hkg_Hr8dKSjQjsXkXJ7czRsAvWrCClZ28Kqcp3Hyv6M1R-tRiBeYqN25Z8Red_YV148RKXrQibb3H4qdDTSIaZBI45eKAxesgDelHVAJ1uH2g72sLUdGff9TfkSPPtkuldnZMhnLj5XjR6vmQ5pl7u_OBPHS3nLa6uhFNZgmMUlIAwYs8ZX47xmD6yhaieTApeT1ZkRIDcwRtaHzADM_8r15D2GydWK67R_ynmY6fPyc8rfSJ6drjGj_W80
	Type	Activity Diagram

	Code	<pre>@startuml start :Customer logs in; if (Logged in?) then (yes) :Customer browses menu; while (Shopping) is (true) :Customer adds or deletes products to shopping cart; :Shopping cart updates total price; endwhile (false) :Customer sends order; :Order system receives order; :Order system prints receipt; :Customer can tracks/sees the order; else (no) :Customer creates profile; endif stop @enduml</pre>
--	-------------	---

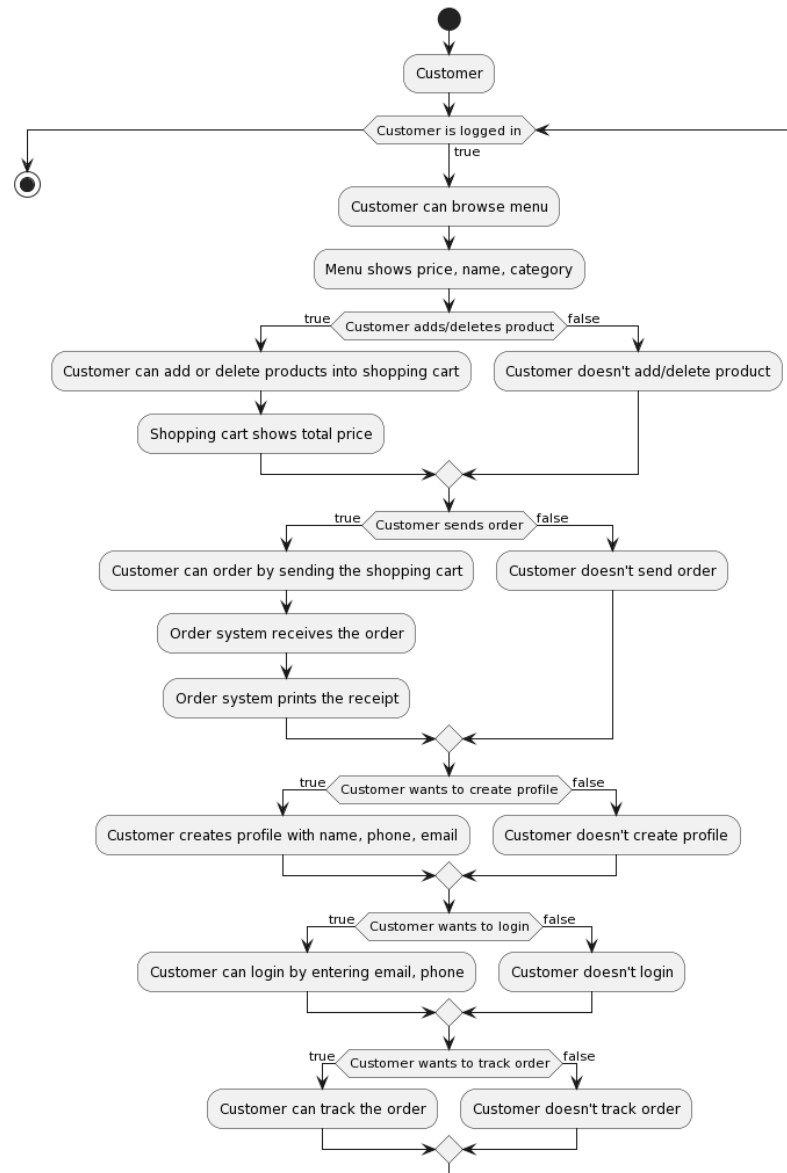
	Diagram	 <pre> graph TD Start(()) --> Login[Customer logs in] Login --> Logged{Logged in?} Logged -- yes --> Browse[Customer browses menu] Logged -- no --> Create[Customer creates profile] Browse --> Shopping{Shopping} Shopping -- false --> Shopping Shopping -- true --> Add[Customer adds or deletes products to shopping cart] Add --> Update[Shopping cart updates total price] Update --> Shopping Update --> Send[Customer sends order] Send --> Receive[Order system receives order] Receive --> Print[Order system prints receipt] Print --> Track[Customer can tracks/sees the order] Create --> Merge{ } Track --> Merge Merge --> End((())) </pre> <p>The diagram illustrates the customer order process. It begins with a start node leading to 'Customer logs in'. A decision node 'Logged in?' branches the flow: if 'yes', the customer browses the menu; if 'no', the customer creates a profile. From 'browses menu', the flow enters a loop at a 'Shopping' decision. If 'false', it loops back to 'Shopping'. If 'true', the customer adds or deletes products to the shopping cart, which then updates the total price. This leads to 'Customer sends order', followed by 'Order system receives order', 'Order system prints receipt', and 'Customer can tracks/sees the order'. The 'Customer creates profile' path also leads to a merge point before the final end node, which is reached after the tracking step.</p>
	MyLearning	<ol style="list-style-type: none"> 1. How to use plantuml and draw a activity diagram. 2. How to code in plantuml.(using black circle, encircled black circle, stadia, diamonds, usecase, bars.)

PlantUML + ChatGPT	Items	Content
	SharedFile	https://chat.openai.com/share/92cc741e-cf9a-4253-bcbf-024d9c109041
	Type	Activity Diagram
	ChatGPT Version	V3.5
	ChatGPT Prompt	<p>Write an UML Activity diagram and use PlantUML script. Please be careful of syntax error such as "(Login --> Customer "</p> <ol style="list-style-type: none"> 1. Customer can add or delete products into shopping cart. 2. Customer can browse menu. 3. Menu would had the price, name, category of products. 4. Customer can order py sending the shopping cart. 5. Order system can receive the order after customer send the order. 6. Shopping cart can show the total price and change it after added or deleted the products. 7. Order system can print the receipt after receiving the order. 8. Customer can create its own profile with name, phone, email. 9. Customer can login by entering email, phone. 10. Customer can track the order.

	ChatGPT Code	<pre>@startuml start :Customer; while (Customer is logged in) is (true) :Customer can browse menu; :Menu shows price, name, category; if (Customer adds/deletes product) then (true) :Customer can add or delete products into shopping cart; :Shopping cart shows total price; else (false) :Customer doesn't add/delete product; endif if (Customer sends order) then (true) :Customer can order by sending the shopping cart; :Order system receives the order; :Order system prints the receipt; else (false) :Customer doesn't send order; endif if (Customer wants to create profile) then (true) :Customer creates profile with name, phone, email; else (false) :Customer doesn't create profile; endif if (Customer wants to login) then (true)</pre>
--	-------------------------	--

		<pre> :Customer can login by entering email, phone; else (false) :Customer doesn't login; endif if (Customer wants to track order) then (true) :Customer can track the order; else (false) :Customer doesn't track order; endif endwhile stop @enduml</pre>
--	--	--

Diagram



	MyLearning	<ol style="list-style-type: none">1. That GPT can generate the plantUML's code very fast.2. That GPT's code may be wrong sometime that I need to change the question's description.
--	-------------------	--

Software Design: Homework #3: Class Diagram

Homework Number: #3

Homework Start Date: 20240322

Homework Due Date: 20240324

Homework Submitted Date: 20240322

Student: 軟體系大三 蔡岳哲

Student ID: 411077010

Answer:

Requirements
<ol style="list-style-type: none">1. Customer can add or delete products into shopping cart.2. Customer can browse menu.3. Menu would had the price, name, category of products.4. Customer can order py sending the shopping cart.5. Order system can receive the order after customer send the order.6. Shopping cart can show the total price and change it after added or deleted the products.7. Order system can print the receipt after receiving the order.8. Customer can create its own profile with name, phone, email.9. Customer can login by entering email, phone.10. Customer can track the order.

Keywords:

Shopping cart, menu, order system, customer profile, login, track order.

Class Diagram:

Methods	My Outcomes	
draw.io	Items	Content
	SharedFile	https://drive.google.com/file/d/1MIRF8H2VLTDZ4CwjPOTmFx3qTE4-KGY/view?usp=sharing
	Type	Class Diagram

	<p>Diagram</p>	<pre> classDiagram class Customer { -id: int +addProductToShoppingCart(): void +removeProductFromShoppingCart(): void +browseMenu(): void +sendOrder(): void +createProfile(): void +login(): void +logout(): void +trackOrder(): void } class Menu { -id: int -product: List<String> +addProduct(): void +removeProduct(): void } class Product { -id: int -name: String -price: float -category: String } class ShoppingCart { -id: int -products: List<String> -totalPrice: float +addProduct(): void +removeProduct(): void +calculateTotalPrice(): void +getTotalPrice(): float } class OrderSystem { +receiveOrder(): void } class Order { -id: int -orderItems: List<String> +printReceipt(): void +returnOrderStatus(): void } class OrderItem { -name: String -quantity: int +calculateSubtotal(): float } class Profile { -id: int -name: String -phone: String -email: String } Customer "1..1" --> "1..*" Menu Customer "1..1" --> "1..*" Profile Customer "1..*" --> "1" ShoppingCart Menu "1..*" --> "1" Product ShoppingCart "*" --> "1" OrderSystem OrderSystem "1..*" --> "1" Order Order "*" --> "*" OrderItem </pre>
	<p>MyLearning</p>	<ol style="list-style-type: none"> 1. How to use drawio to draw(link)the different classes. 2. How to draw class diagram. 3. The relation about different classes.

PlantUML	Items	Content
	SharedFile	https://www.plantuml.com/plantuml/duml/bLJRQjmm47ttLuXzSYyExQie8H0e15fsgVC3YZHnH7LnFT6M4FhljILPHC8XoJvuLtEePkRC6U-LH-OmQDLnnRmdry6ZrU38KqUMpo059JS6gy4m3PICwAIPKcH-iAO9WMPILQ5FJ8YJio9mlBNNp64_fnCbEVo1ahEL8h4TQ7k6Z7ntLhz-vSxPlnv-WWczNXwKh3yhXWSZZaw0wwimTy1mhNKI5VJFzNqcHLiwLhgg9CfEqIGHFxfJkTNIq37-E_LWroSbyR1n_dNPWl71plDoUPMzQqCUXQVaX_JuBG_aCe5eaQcJanmekLUMOMj1_AmczamnODY2BSbPKsO02ierlj2eWdl4_3-Jakn3BnrRzi13PmXJDOz5bx975tQpHijl47GcvBZ5iXQbZ86Vm47EU-q7DdZer_uv6aJkN3Fy8OjAQMspjeN94N_16PGuY9B6cny7SDTt9RAkl87SFXw8CDmkNvNAXdBIETiG_djsWMVpw3hgiqlh2yN5nyFHLKNj-ebCAPHqJAgxWgC25hz1m00
	Type	Class Diagram

	Code	<pre>@startuml class Customer { -id: Int +addProductToCart(): void +removeProductFromCart(): void +browseMenu(): void +sendOrder(): void +createProfile(): void +login(): void +logout(): void +trackOrder(): void } class ShoppingCart { -id: Int -products: List<String> -totalPrice: float +addProduct(): void +removeProduct(): void +calculateTotalPrice(): void +getTotalPrice(): float } class Menu { -id: Int</pre>
--	-------------	---

		<pre>-products: List<String> +addProduct(): void +removeProduct(): void } class Product { -id: Int -name: String -price: float -category: String } class Order { -id: Int -orderItems: List<String> +printReceipt(): void +returnOrderStatus(): void } class OrderSystem { +receiveOrder(): void } class OrderItem { -product: Product</pre>
--	--	--

		<pre>-quantity: int +calculateSubtotal(): float } class Profile{ -id: Int -name: String -phone: String -email: String } Customer "1" --> "1" ShoppingCart Customer "1" --> "1" Profile Customer "1" --> "1" Menu ShoppingCart "1..*" --> "1" OrderSystem ShoppingCart "1..*" --> "1..*" Product Menu "1" --> "1..*" Product OrderSystem"1" --> "1" Order Order "1" --> "1..*" OrderItem @enduml</pre>
--	--	--

	<p>Diagram</p>	<pre> classDiagram class Customer { +id: Int +addProductToCart(): void +removeProductFromCart(): void +browseMenu(): void +sendOrder(): void +createProfile(): void +login(): void +logout(): void +trackOrder(): void } class Menu { +id: Int +products: List<String> +addProduct(): void +removeProduct(): void } class ShoppingCart { +id: Int +products: List<String> +totalPrice: float +addProduct(): void +removeProduct(): void +calculateTotalPrice(): void +getTotalPrice(): float } class Profile { +id: Int +name: String +phone: String +email: String } class Product { +id: Int +name: String +price: float +category: String } class OrderSystem { +receiveOrder(): void } class Order { +id: Int +orderItems: List<String> +printReceipt(): void +returnOrderStatus(): void } class OrderItem { +product: Product +quantity: int +calculateSubtotal(): float } Customer "1" -- "1" Menu Customer "1" -- "1" ShoppingCart Customer "1" -- "1" Profile Menu "1" -- "1..*" Product Product "1..*" -- "1..*" ShoppingCart ShoppingCart "1..*" -- "1" OrderSystem OrderSystem "1" -- "1" Order Order "1" -- "1..*" OrderItem </pre>
	<p>MyLearning</p>	<ol style="list-style-type: none"> 1. How to draw class diagram. 2. The relation about every classes.

PlantUML + ChatGPT	Items	Content
	SharedFile	https://chat.openai.com/share/fbac65c3-77b2-4c51-8c11-0a4c705f920b
	Type	Class Diagram
	ChatGPT Version	V3.5
	ChatGPT Prompt	<p>Please write an uml class diagram and use PlantUML script.</p> <ol style="list-style-type: none"> 1. Customer can add or delete products into shopping cart. 2. Customer can browse menu. 3. Menu would had the price, name, category of products. 4. Customer can order py sending the shopping cart. 5. Order system can receive the order after customer send the order. 6. Shopping cart can show the total price and change it after added or deleted the products. 7. Order system can print the receipt after receiving the order. 8. Customer can create its own profile with name, phone, email. 9. Customer can login by entering email, phone. 10. Customer can track the order.

	ChatGPT Code	<pre>@startuml class Customer { -name: String -phone: String -email: String +addProductToCart(product: Product): void +removeProductFromCart(product: Product): void +browseMenu(menu: Menu): void +sendOrder(): void +createProfile(name: String, phone: String, email: String): void +login(emailOrPhone: String): void +trackOrder(order: Order): void } class ShoppingCart { -products: List<Product> -totalPrice: float +addProduct(product: Product): void +removeProduct(product: Product): void +calculateTotalPrice(): void +getTotalPrice(): float } class Menu {</pre>
--	-------------------------	--

		<pre>-products: List<Product> +addProduct(product: Product): void +removeProduct(product: Product): void } class Product { -name: String -price: float -category: String } class Order { -orderItems: List<OrderItem> +printReceipt(): void } class OrderSystem { +receiveOrder(order: Order): void } class OrderItem { -product: Product -quantity: int +calculateSubtotal(): float }</pre>
--	--	--

		<p>Customer "1" --> "1" ShoppingCart Customer "1" --> "1" Order Customer "1" --> "1" Profile Customer "1" --> "1" OrderSystem</p> <p>ShoppingCart "1" --> "0..*" Product</p> <p>Menu "1" --> "0..*" Product</p> <p>Order "1" --> "0..*" OrderItem</p> <p>@enduml</p>
--	--	---

	Diagram	<pre> classDiagram class Customer { +String name +String phone +String email +addProductToCart(product: Product) void +removeProductFromCart(product: Product) void +browseMenu(menu: Menu) void +sendOrder() void +createProfile(name: String, phone: String, email: String) void +login(emailOrPhone: String) void +trackOrder(order: Order) void } class Order { +List<OrderItem> orderItems +printReceipt() void } class OrderSystem { +receiveOrder(order: Order) void } class Profile { } class ShoppingCart { +List<Product> products +float totalPrice +addProduct(product: Product) void +removeProduct(product: Product) void +calculateTotalPrice() void +getTotalPrice() float } class Menu { +List<Product> products +addProduct(product: Product) void +removeProduct(product: Product) void } class OrderItem { +Product product +int quantity +calculateSubtotal() float } class Product { +String name +float price +String category } Customer "1" -- "1" Order Customer "1" -- "1" OrderSystem Customer "1" -- "1" Profile Customer "1" -- "1" ShoppingCart Customer "1" -- "1" Menu Order "1" -- "0..*" OrderItem ShoppingCart "1" -- "0..*" Product Menu "1" -- "0..*" Product </pre>
	MyLearning	<ol style="list-style-type: none"> 1. I find out that gpt can't make the right script from my requirement. 2. There might be some bugs in gpt's script.

Software Design: Homework #4: Statechart Diagram

Homework Number: #4

Homework Start Date: 20240329

Homework Due Date: 20240331

Homework Submitted Date: 20240329

Student: 軟體系大三 蔡岳哲

Student ID: 411077010

Answer:

Requirements
<ol style="list-style-type: none">1. Customer can add or delete products into shopping cart.2. Customer can browse menu.3. Menu would had the price, name, category of products.4. Customer can order py sending the shopping cart.5. Order system can receive the order after customer send the order.6. Shopping cart can show the total price and change it after added or deleted the products.7. Order system can print the receipt after receiving the order.8. Customer can create its own profile with name, phone, email.9. Customer can login by entering email, phone.10. Customer can track the order.

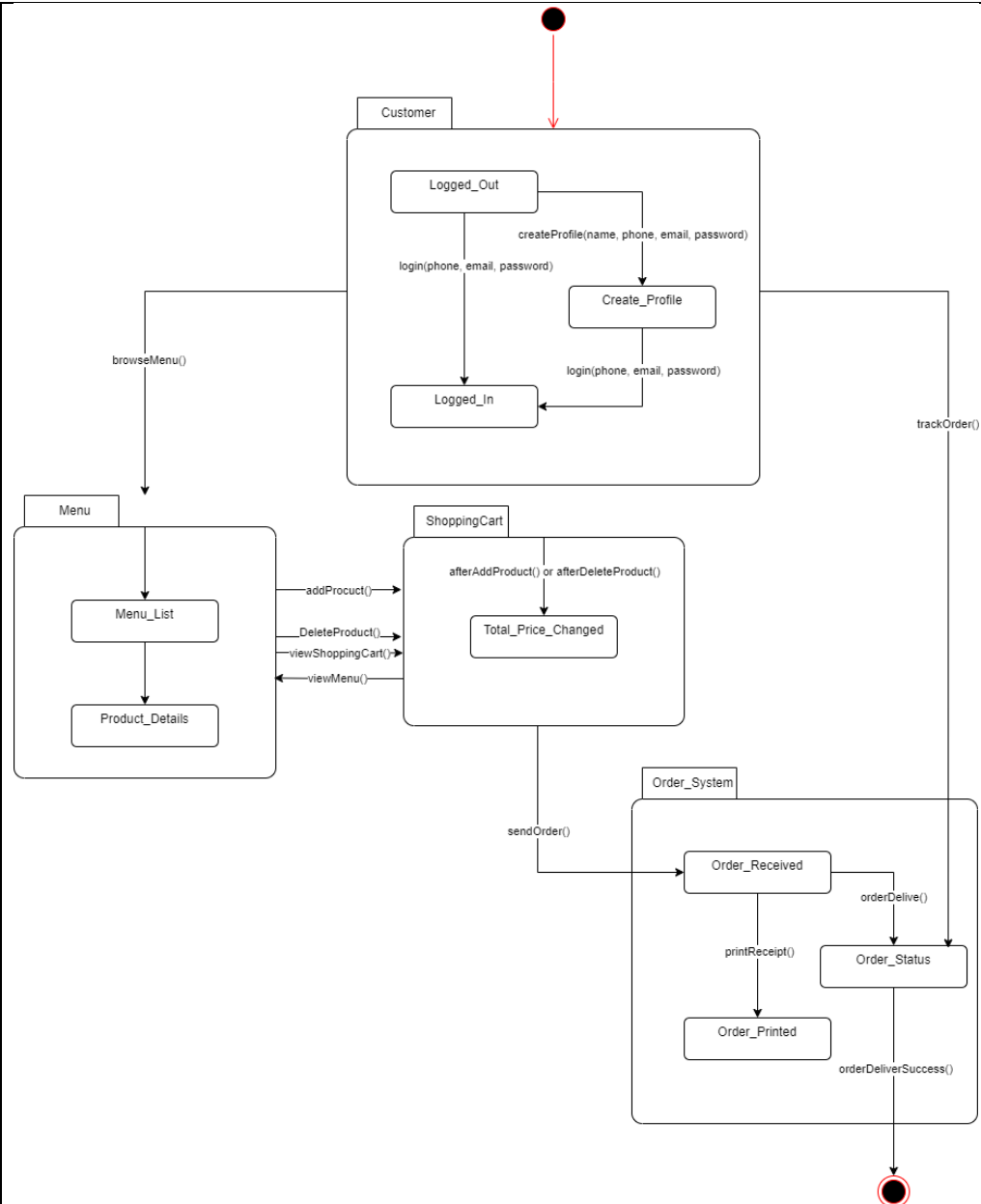
Keywords:

Shopping cart, menu, order system, customer profile, login, track order.

Statechart Diagram:

Methods	My Outcomes	
draw.io	Items	Content
	SharedFile	https://drive.google.com/file/d/1m3j32XqV8_Ht3WkDDycM3nHKuA0PcWdN/view?usp=sharing
	Type	Statechart Diagram

Diagram

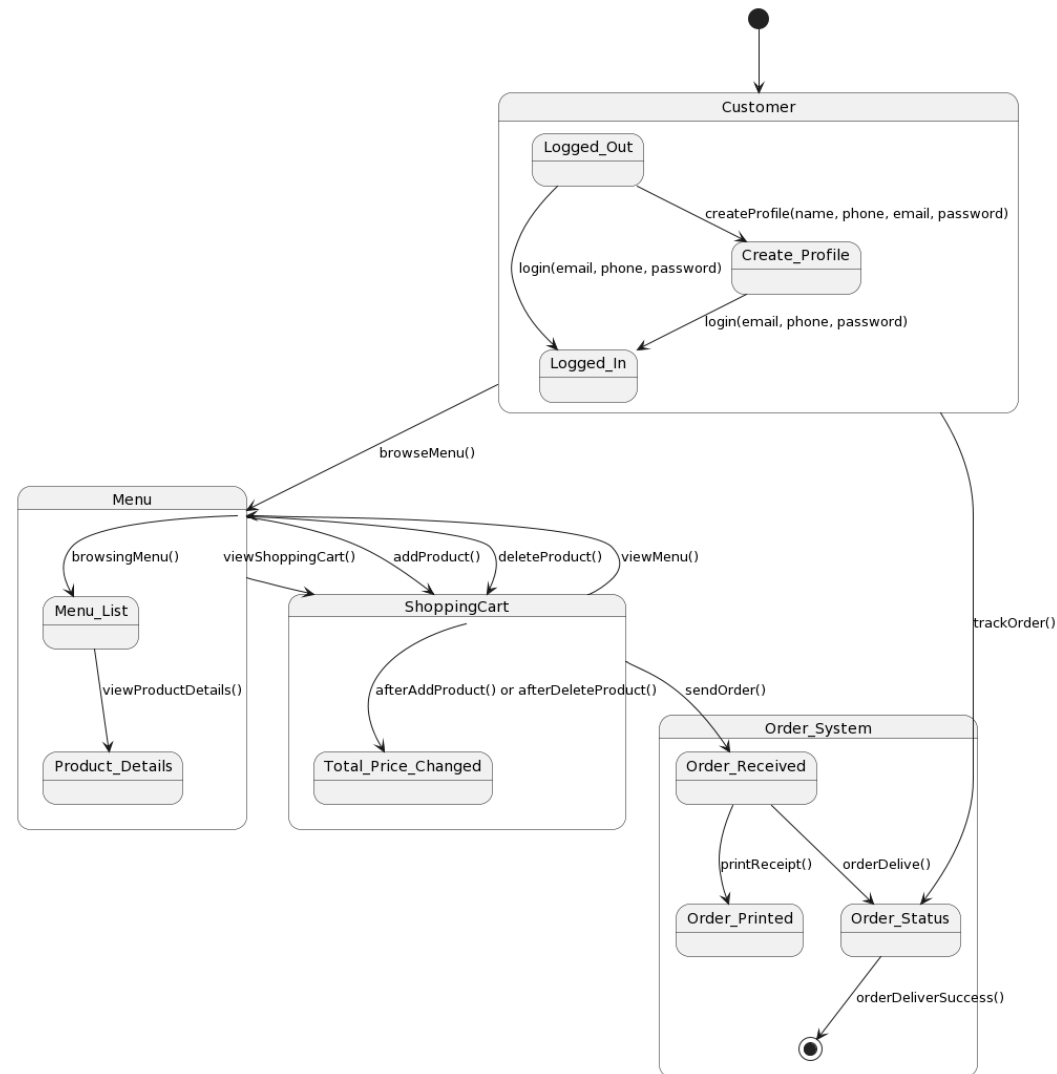


	MyLearning	<ol style="list-style-type: none"> How to draw state chart diagram. Plant uml is more easy to do diagram.(By coding)
PlantUML	Items	Content
	SharedFile	https://www.plantuml.com/plantuml/uml/jPJ1JiCm38RIVoeS1s8lm04DoWK9j8djXb0L6jD5j4cLkfiGujs9LtUhfk020xii_ID_jXKdNNle0_PDdMKlwbwzgFhzHUT-XRo1aMTn3ECRgArFnDzeUVLM1APOzaZm9fxiFJbshsbVMpQ3HjhvlxTOxk4WXUG2DKAo2VxSrHAua0JYUETt0WCS_e6H35edziJXvYBWBVjzPLnLFuFf8lL4C5Cm6kpGGSciCnR6kwKkCZ3Q60sQa8rd7G0q80akKMczzsvAlwq2LTXRseqOIWzAMAAaSWBSy2os0W56jmLAO3Pmu2soUR9-ypb61tOA8_34k92Cy9MmNh48XgQNLmj2WejX14b-vCnnkFkYPI2KM-rQwYf68MoeVKEIV9tv-sl43SyYReyYFQ8eMq78_YUtnBTFZyZexJvQwAUq1jwuuhiihYD0UFuRD3Q3YNaFvDjJ2yMD0ziHZaGRHbxnGsl86zDtv0G00
	Type	Statechart Diagram

	Code	<pre> @startuml [*] --> Customer state Customer { Logged_Out --> Logged_In : login(email, phone, password) Logged_Out --> Create_Profile : createProfile(name, phone, email, password) Create_Profile --> Logged_In : login(email, phone, password) } state Menu { Menu --> Menu_List : browsingMenu() Menu_List --> Product_Details : viewProductDetails() } state ShoppingCart { ShoppingCart --> Total_Price_Changed : afterAddProduct() or afterDeleteProduct() } state Order_System { ShoppingCart --> Order_Received : sendOrder() Order_Received --> Order_Printed : printReceipt() Order_Received --> Order_Status : orderDelive() Order_Status --> [*]: orderDeliverSuccess() } </pre>
--	-------------	--

		Customer --> Menu : browseMenu() Menu --> ShoppingCart : viewShoppingCart() Menu --> ShoppingCart : addProduct() Menu --> ShoppingCart : deleteProduct() ShoppingCart --> Menu : viewMenu() Customer --> Order_Status : trackOrder() @enduml
--	--	--

Diagram



MyLearning

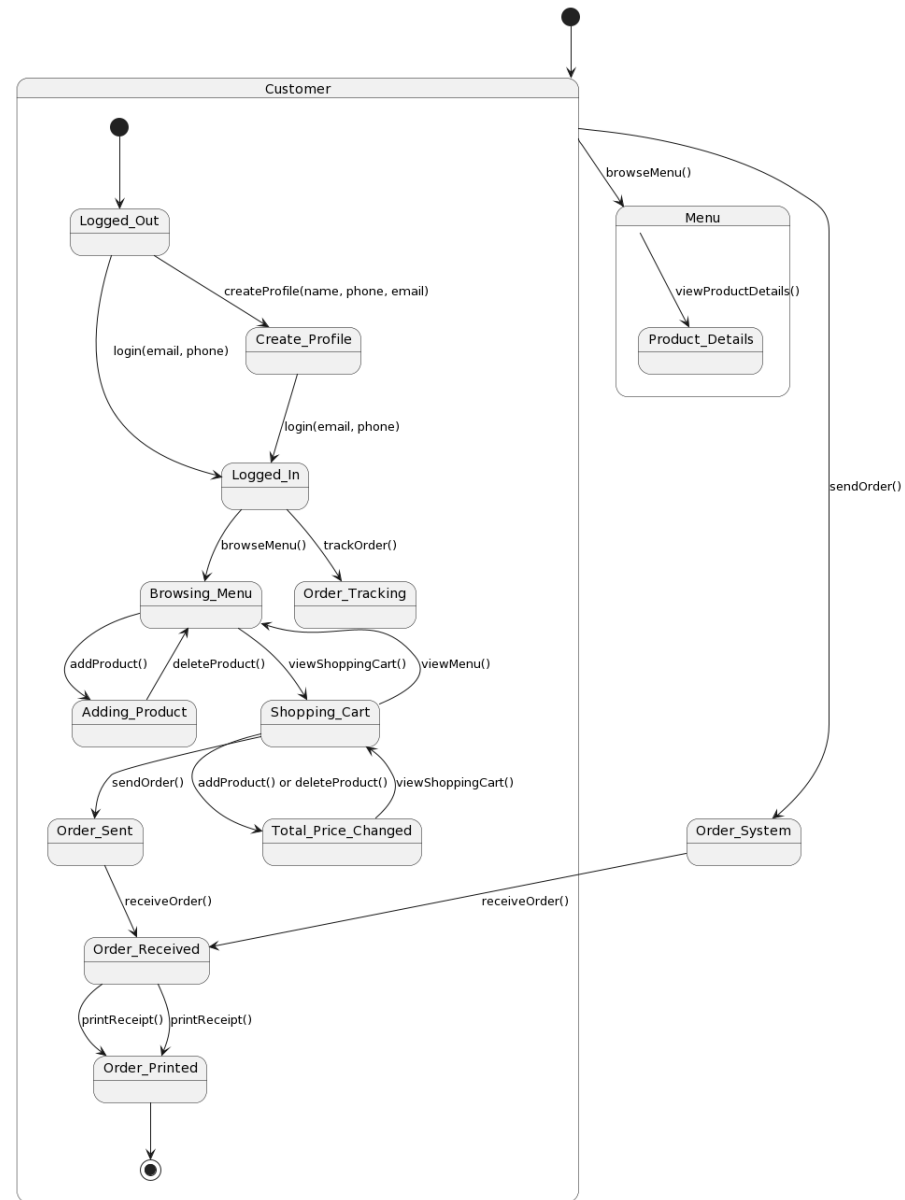
1. How the syntax for statechart diagram and plantuml.

PlantUML + ChatGPT	Items	Content
	SharedFile	https://chat.openai.com/share/c83d68f4-b7c9-44e1-b834-b92a90f71793
	Type	Statechart Diagram
	ChatGPT Version	V3.5
	ChatGPT Prompt	<p>Write an UML Statechart diagram and use PlantUML script.</p> <ol style="list-style-type: none"> 1. Customer can add or delete products into shopping cart. 2. Customer can browse menu. 3. Menu would had the price, name, category of products. 4. Customer can order py sending the shopping cart. 5. Order system can receive the order after customer send the order. 6. Shopping cart can show the total price and change it after added or deleted the products. 7. Order system can print the receipt after receiving the order. 8. Customer can create its own profile with name, phone, email. 9. Customer can login by entering email, phone. 10. Customer can track the order. <p>thanks</p>

	ChatGPT Code <pre> @startuml [*] --> Customer state Customer { [*] --> Logged_Out Logged_Out --> Logged_In : login(email, phone) Logged_Out --> Create_Profile : createProfile(name, phone, email) Create_Profile --> Logged_In : login(email, phone) Logged_In --> Browsing_Menu : browseMenu() Browsing_Menu --> Adding_Product : addProduct() Adding_Product --> Browsing_Menu : deleteProduct() Browsing_Menu --> Shopping_Cart : viewShoppingCart() Shopping_Cart --> Browsing_Menu : viewMenu() Shopping_Cart --> Order_Sent : sendOrder() Order_Sent --> Order_Received : receiveOrder() Order_Received --> Order_Printed : printReceipt() Order_Printed --> [*] Shopping_Cart --> Total_Price_Changed : addProduct() or deleteProduct() Total_Price_Changed --> Shopping_Cart : viewShoppingCart() Logged_In --> Order_Tracking : trackOrder() } state Menu { Menu --> Product_Details : viewProductDetails() </pre>
--	--

		<pre>} state Order_System { Order_System --> Order_Received : receiveOrder() Order_Received --> Order_Printed : printReceipt() } Customer --> Menu : browseMenu() Customer --> Order_System : sendOrder() @enduml</pre>
--	--	--

Diagram



	MyLearning	<ol style="list-style-type: none">1. How the state chart diagram works.2. How state works and means.
--	-------------------	---

Software Design: Homework #5: Sequence Diagram

Homework Number: #5

Homework Start Date: 20240405

Homework Due Date: 202400407

Homework Submitted Date: 20240407

Student: 軟體系大三 蔡岳哲

Student ID: 411077010

Answer:

Requirements
<ol style="list-style-type: none">1. Customer can add or delete products into shopping cart.2. Customer can browse menu.3. Menu would had the price, name, category of products.4. Customer can order py sending the shopping cart.5. Order system can receive the order after customer send the order.6. Shopping cart can show the total price and change it after added or deleted the products.7. Order system can print the receipt after receiving the order.8. Customer can create its own profile with name, phone, email.9. Customer can login by entering email, phone.10. Customer can track the order.

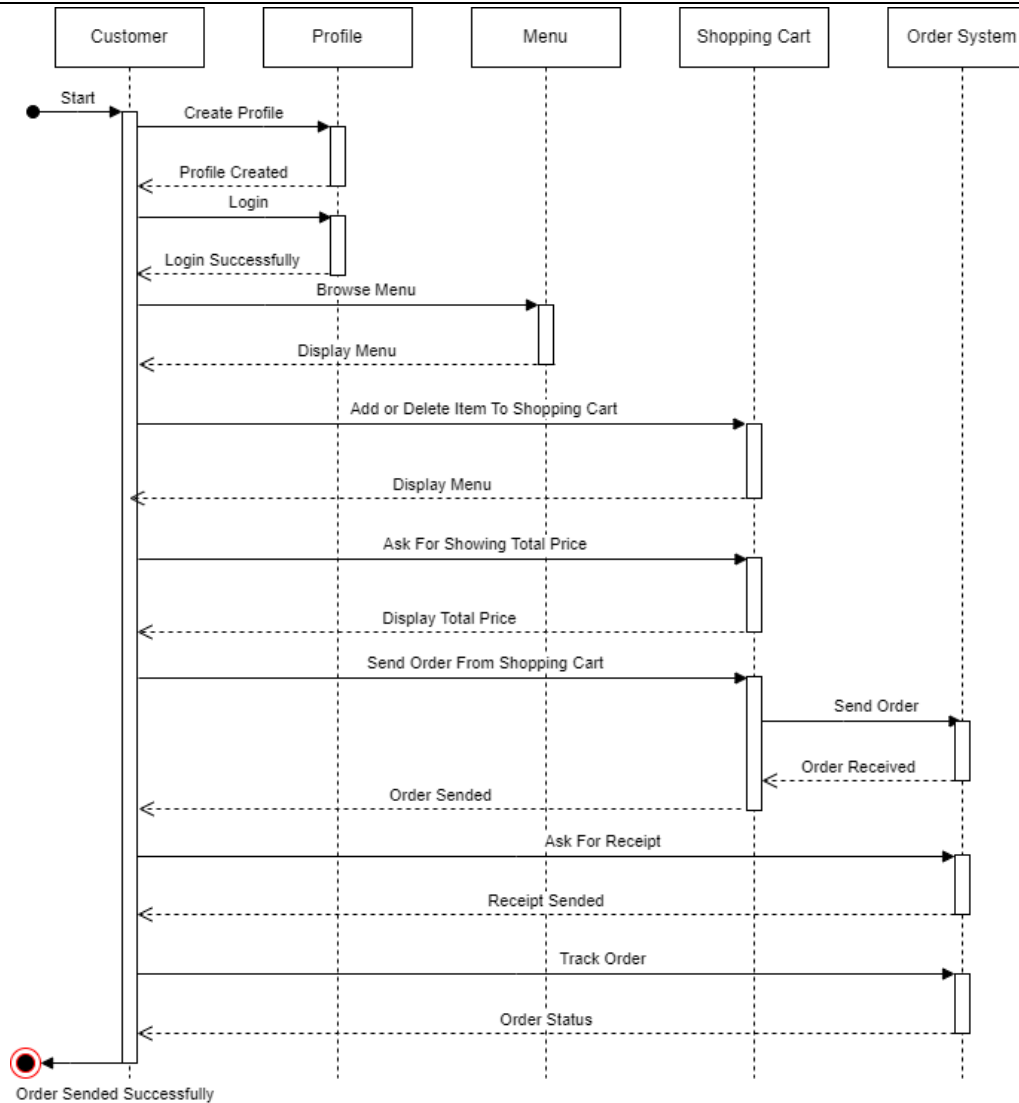
Keywords:

Shopping cart, menu, order system, customer profile, login, track order.

Sequence Diagram:

Methods	My Outcomes	
draw.io	Items	Content
	SharedFile	https://drive.google.com/file/d/1qGj-gZOyPdJ9buH-o_QDCqOHp7CAhP37/view?usp=sharing
	Type	Sequence Diagram

Diagram



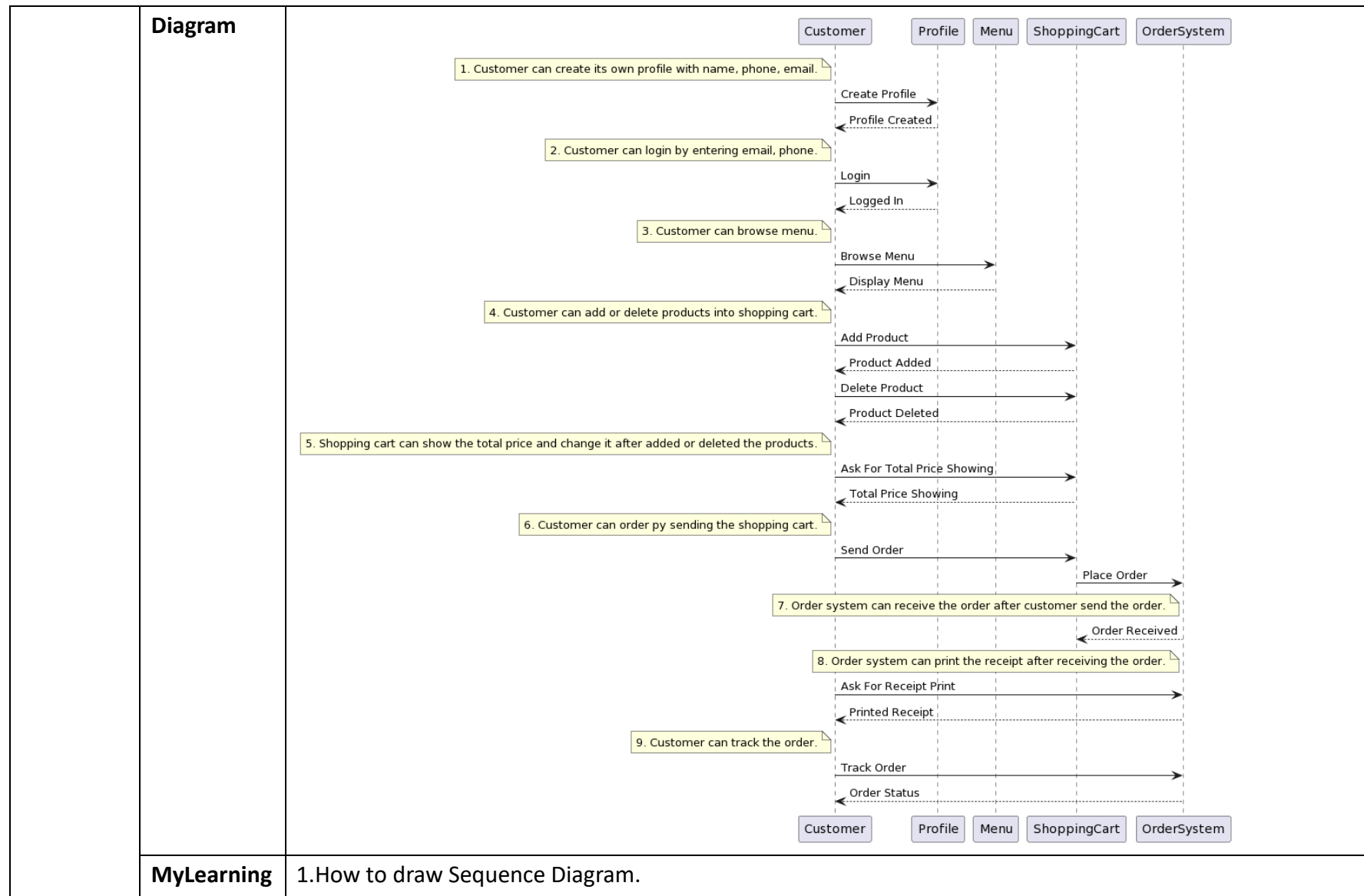
MyLearning

1. How to draw a Sequence diagram.

PlantUML	Items	Content
	SharedFile	https://www.plantuml.com/plantuml/uml/ZPNTZjem48NlynJd0Gfl_zjShBPbLQblgw9bNy0R3s1jOaVsf8YthqTEqZXi43SapC-SptEmk0-iF7TDNHHjV36LQPLbRBh0hY6V1RVUxKrDMUmtsIuBx8wkROqzR68il_pncIpk79YQeh2E2JNj6Mu_YfLIr-CxAcLHUL9CC1pWJXPjqiV9y15MDVG6xT7P-A16cNfTZCshkm6sn2OD6U3x9rQnPXGUidsnNW9yDmEitS5OF9z1bid7KoUIldci1w9SqBN34t84qVjeb0hoV8JnxTmg4XcoNwuap9Rwdj7mfv6Eczs12MwjpoYz9VfX9AgtXF3JLn2lkWaxY06FP8VJsngBF6L3sooZnBGxPfjPYcia1Xn8f9trbt4E2kNrYwbYs6X_Ns4qF8qUNuvt0Hm8xLZLQRog2iXhLKTa3mJ3KdidBZcYo9lrDuwgkxYMyu4ViUn81R4L0AawnuCgfNYjVFDcdcPzERYRQCm9P7Hk5zNOZTx4fNUuPtztqoiUzrwgY5Ch9fcKeyNcT0WWf8elUAZ9_IS1wshJbQg0lyl_PTJ4TkRf0JlCVqr2zJ5FYooMcc6vPn19MErYU8AF812JRMpPvyFcndx6LgHVS_OtkP2aWF3SMYILzdzh9NrSjDJ4zlcKBB9CFwTgou2qLnJrRB_yS_
	Type	Sequence Diagram

	Code @startuml participant Customer participant Profile participant Menu participant ShoppingCart participant OrderSystem note left of Customer:1. Customer can create its own profile with name, phone, email. Customer -> Profile: Create Profile Profile --> Customer: Profile Created note left of Customer:2. Customer can login by entering email, phone. Customer -> Profile: Login Profile --> Customer: Logged In note left of Customer: 3. Customer can browse menu. Customer -> Menu: Browse Menu Menu --> Customer: Display Menu note left of Customer: 4. Customer can add or delete products into shopping cart. Customer -> ShoppingCart: Add Product ShoppingCart --> Customer: Product Added Customer -> ShoppingCart: Delete Product ShoppingCart --> Customer: Product Deleted
--	--

		<p>note left of Customer: 5. Shopping cart can show the total price and change it after added or deleted the products.</p> <p>Customer -> ShoppingCart: Ask For Total Price Showing</p> <p>ShoppingCart --> Customer: Total Price Showing</p> <p>note left of Customer: 6. Customer can order py sending the shopping cart.</p> <p>Customer -> ShoppingCart: Send Order</p> <p>ShoppingCart -> OrderSystem: Place Order</p> <p>note left of OrderSystem : 7. Order system can receive the order after customer send the order.</p> <p>OrderSystem --> ShoppingCart: Order Received</p> <p>note left of OrderSystem: 8. Order system can print the receipt after receiving the order.</p> <p>Customer -> OrderSystem: Ask For Receipt Print</p> <p>OrderSystem --> Customer : Printed Receipt</p> <p>note left of Customer: 9. Customer can track the order.</p> <p>Customer -> OrderSystem: Track Order</p> <p>OrderSystem --> Customer : Order Status</p> <p>@enduml</p>
--	--	--



PlantUML + ChatGPT	Items	Content
	SharedFile	https://chat.openai.com/share/30bdf500-00a9-4c11-a8a0-57f905f2f82e
	Type	Sequence Diagram
	ChatGPT Version	V3.5
	ChatGPT Prompt	<p>Write an UML Sequence diagram and use PlantUML script.</p> <ol style="list-style-type: none"> 1. Customer can add or delete products into shopping cart. 2. Customer can browse menu. 3. Menu would had the price, name, category of products. 4. Customer can order py sending the shopping cart. 5. Order system can receive the order after customer send the order. 6. Shopping cart can show the total price and change it after added or deleted the products. 7. Order system can print the receipt after receiving the order. 8. Customer can create its own profile with name, phone, email. 9. Customer can login by entering email, phone. 10. Customer can track the order.

	ChatGPT Code	<pre> @startuml participant Customer participant ShoppingCart participant Menu participant OrderSystem Customer -> ShoppingCart: Add Product Customer -> ShoppingCart: Delete Product Customer -> Menu: Browse Menu Menu --> Customer: Show Menu Customer -> OrderSystem: Send Shopping Cart OrderSystem -> OrderSystem: Receive Order OrderSystem -> ShoppingCart: Calculate Total Price ShoppingCart --> Customer: Show Total Price OrderSystem -> OrderSystem: Print Receipt Customer --> OrderSystem: Track Order Customer -> Customer: Create Profile Customer -> Customer: Login @enduml </pre>
--	-------------------------	--

	Diagram	<pre> sequenceDiagram participant Customer participant ShoppingCart participant Menu participant OrderSystem Customer->>ShoppingCart: Add Product Customer->>ShoppingCart: Delete Product Customer->>Menu: Browse Menu Menu-->>Customer: Show Menu Customer->>OrderSystem: Send Shopping Cart OrderSystem->>OrderSystem: Receive Order OrderSystem->>ShoppingCart: Calculate Total Price ShoppingCart-->>Customer: Show Total Price OrderSystem->>OrderSystem: Print Receipt Customer->>OrderSystem: Track Order OrderSystem-->>Customer: Create Profile OrderSystem-->>Customer: Login </pre>
	MyLearning	<ol style="list-style-type: none"> 1. GPT can't always give me the correct draw. 2. GPT can only draw part of the requirements.