Prompt
Create **use-case diagram** in Plant UML format for the following functionality

Authentication

Customer will have the following Attributes: Username (unique), Password, Email, Postal Address, Member/Non-member status.

Manager will have the following Attributes: Hardcoded Username, Password, Email Address.

Login/Logout: Customers and managers can log in and log out; login enables logout and checkout, with a 30-minute session timeout. No password recovery available.

Membership

Customers can sign up to become members or remain non-members. Membership status is captured as a boolean value.

Shopping Cart-Customers can add one or more books to the shopping cart. Duplicate copies of the same book are not allowed.

Inventory Management

Book will have following Attributes: Title, Author, Price, Stock/Quantity.

Stop-Order Management: Each book has a stop-order status (on/off). If stock falls below a threshold, managers are notified unless a stop-order is applied.

Manager will perform following Actions: Update stock quantities, change book prices, set reorder thresholds, and view transaction logs. If stock reaches the reorder threshold, an email is sent to the manager, and items are automatically reordered unless a stop-order is active.

Promotions

Promotion Attributes: Percentage discount, expiration date.

Promotion Use: Available to members only; one promotion code per purchase.

Manager Actions: Create/update promotions, define discounts and expiration dates. Promotions are auto-deleted after expiration, and emails are sent to all members.

Checkout

Checkout is restricted to logged-in customers. During checkout:

A 16-digit credit card number is collected.

Transactions are logged and recorded.

Order details are sent via email.

Inventory stock is decremented based on purchased items.

PlantUML

```
@startuml
```

```
actor Customer
actor Manager

usecase "Sign Up" as UC1
usecase "Login" as UC2
usecase "Logout" as UC3
usecase "Auto Logout" as UC4
usecase "Checkout" as UC5
usecase "Add to Cart" as UC6
usecase "Maintain Inventory" as UC7
usecase "Update Stock" as UC8
usecase "Set Prices" as UC9
usecase "View Logs" as UC10
usecase "Specify Stop-Orders" as UC11
usecase "Create Promotions" as UC12
usecase "Email Promotions" as UC13
usecase "Auto Delete Promotions" as UC14

Customer --> UC1
Customer --> UC2
Customer --> UC3
Customer --> UC4
Customer --> UC5
Customer --> UC6

Manager --> UC2
Manager --> UC3
Manager --> UC7
Manager --> UC8
Manager --> UC9
Manager --> UC10
Manager --> UC11
Manager --> UC12
Manager --> UC13
Manager --> UC14

UC2 --> UC3 : <<include>>
UC2 --> UC5 : <<include>>
UC2 --> UC4 : <<include>>
UC12 --> UC13 : <<include>>
UC12 --> UC14 : <<include>>

@enduml
```

Similarity Score

V1	13
V2	18

E1	17
E2	26
Mv	12
ME	12
Node Similarity Ratio	0.66666667
Edge Similarity Ratio	0.46153846
Similarity Score	0.56410256

Adjusted Prompt
Create **use-case diagram** in Plant UML format for the following functionality

Authentication

Customer will have the following Attributes: Username (unique), Password, Email, Postal Address, Member/Non-member status.

Manager will have the following Attributes: Hardcoded Username, Password, Email Address.

Login/Logout: Customers and managers can log in and log out; login enables logout and checkout, with a 30-minute session timeout. No password recovery available.

Membership

Customers can sign up to become members or remain non-members. Membership status is captured as a boolean value.

Shopping Cart-Customers can add one or more books to the shopping cart. Duplicate copies of the same book are not allowed.

Inventory Management

Book will have following Attributes: Title, Author, Price, Stock/Quantity.

Stop-Order Management: Each book has a stop-order status (on/off). If stock falls below a threshold, managers are notified unless a stop-order is applied.

Manager will perform following Actions: Update stock quantities, change book prices, set reorder thresholds, and view transaction logs.
System will send an email to the manager, If stock reaches the reorder threshold, . items are automatically reordered unless a stop-order is active.

Promotions

Promotion Attributes: Percentage discount, expiration date.

Promotion Use: Available to members only; one promotion code per purchase.

Manager Actions: Create/update promotions, define discounts and expiration dates. Promotions are auto-deleted after expiration, and emails are sent to all members.

Checkout

Checkout is restricted to logged-in customers. During checkout:

A 16-digit credit card number is collected.

Transactions are logged and recorded.

Order details are sent via email.

Inventory stock is decremented based on purchased items.

```plantuml
@startuml
actor Customer
actor Manager

usecase "Sign Up" as UC1
usecase "Login" as UC2
usecase "Logout" as UC3
usecase "Auto Logout" as UC4
usecase "Checkout" as UC5
usecase "Add to Cart" as UC6
usecase "Maintain Inventory" as UC7
usecase "Update Stock" as UC8
usecase "Set Prices" as UC9
usecase "View Logs" as UC10
usecase "Specify Stop-Orders" as UC11
usecase "Create Promotions" as UC12
usecase "Email Promotions" as UC13
usecase "Auto Delete Promotions" as UC14

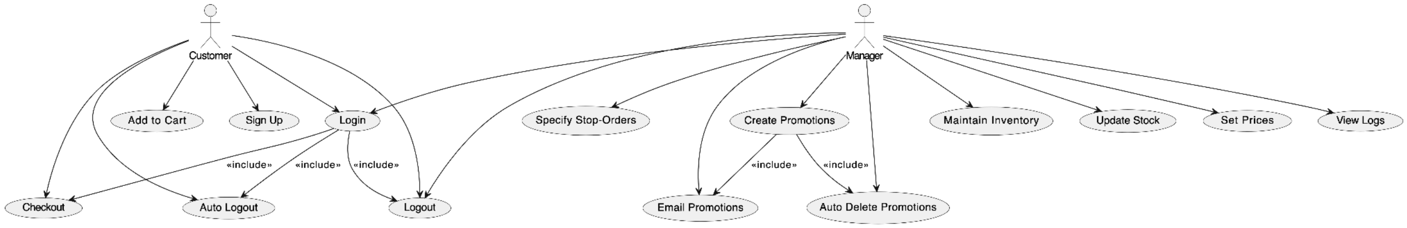
Customer --> UC1
Customer --> UC2
Customer --> UC3
Customer --> UC4
Customer --> UC5
Customer --> UC6

Manager --> UC2
Manager --> UC3
Manager --> UC7
```

~~~~
Manager --> UC8
Manager --> UC9
Manager --> UC10
Manager --> UC11
Manager --> UC12
Manager --> UC13
Manager --> UC14

UC2 --> UC3 : <<include>>
UC2 --> UC5 : <<include>>
UC2 --> UC4 : <<include>>
UC12 --> UC13 : <<include>>
UC12 --> UC14 : <<include>>

@enduml
```
~~~~



Similarity Score

V1	13
V2	18
E1	17
E2	26
Mv	13
ME	13
Node Similarity Ratio	0.72222222
Edge Similarity Ratio	0.5
Similarity Score	0.61111111

Prompt
Create class diagram for the following functionality
Create class diagram in Plant UML Diagram for the following system

Authentication- Customer will have the following Attributes: Username (unique), Password, Email, Postal Address, Member/Non-member status.

Manager will have the following Attributes: Hardcoded Username, Password, Email Address.

Login/Logout: Customers and managers can log in and log out; login enables logout and checkout, with a 30-minute session timeout. No password recovery available.

Membership-Customers can sign up to become members or remain non-members. Membership status is captured as a boolean value.

Shopping Cart-Customers can add one or more books to the shopping cart. Duplicate copies of the same book are not allowed.

Inventory Management

Book will have following Attributes: Title, Author, Price, Stock/Quantity.

Stop-Order Management: Each book has a stop-order status (on/off). If stock falls below a threshold, managers are notified unless a stop-order is applied.

Manager will perform following Actions: Update stock quantities, change book prices, set reorder thresholds, and view transaction logs. If stock reaches the reorder threshold, an email is sent to the manager, and items are automatically reordered unless a stop-order is active.

Promotions

Promotion Attributes: Percentage discount, expiration date.

Promotion Use: Available to members only; one promotion code per purchase.

Manager Actions: Create/update promotions, define discounts and expiration dates. Promotions are auto-deleted after expiration, and emails are sent to all members.

Checkout

Checkout is restricted to logged-in customers. During checkout:

A 16-digit credit card number is collected.

Transactions are logged and recorded.

Order details are sent via email.

}

Inventory stock is decremented based on purchased items.

PlantUML

```
@startuml

class Customer {
    - String username
    - String password
    - String email
    - String postalAddress
    - boolean memberStatus
    + login()
    + logout()
    + signUp()
    + addToCart(Book book)
    + checkout(Cart cart, String creditCardNumber)
}

class Manager {
    - String username
    - String password
    - String email
    + login()
    + logout()
    + updateStock(Book book, int quantity)
    + changePrice(Book book, double newPrice)
    + setReorderThreshold(Book book, int threshold)
    + viewTransactionLogs()
    + createPromotion(Promotion promotion)
}

class Book {
    - String title
    - String author
    - double price
    - int stock
    - boolean stopOrderStatus
}

class Cart {
    - List<Book> books
    + addBook(Book book)
    + removeBook(Book book)
}

class Promotion {
    - double discountPercentage
    - Date expirationDate
}
```

```
class InventoryService {
    + checkStock(Book book)
    + updateInventory(Cart cart)
    + notifyManager(Book book)
}

class EmailService {
    + sendOrderEmail(String orderDetails)
    + sendPromotionEmail(Promotion promotion)
    + sendReorderNotification(Book book, String managerEmail)
}

Customer --> Cart : has
Customer --> Book : adds
Manager --> Book : manages
Manager --> Promotion : creates
InventoryService --> Book : manages
InventoryService --> Manager : notifies
EmailService --> Manager : notifies
EmailService --> Customer : notifies

@enduml
```

Similarity Score

V1	10
V2	7
E1	13
E2	7
Mv	7
ME	7
Node Similarity Ratio	0.7
Edge Similarity Ratio	0.53846154
Similarity Score	0.61923077

Prompt

Create sequence diagram for the following functionality
Create Plant UML Sequence Diagram for the following system

Authentication- Customer will have the following Attributes: Username (unique), Password, Email, Postal Address, Member/Non-member status.

Manager will have the following Attributes: Hardcoded Username, Password, Email Address.

Login/Logout: Customers and managers can log in and log out; login enables logout and checkout, with a 30-minute session timeout. No password recovery available.

Membership-Customers can sign up to become members or remain non-members. Membership status is captured as a boolean value.

Shopping Cart-Customers can add one or more books to the shopping cart. Duplicate copies of the same book are not allowed.

Inventory Management

Book will have following Attributes: Title, Author, Price, Stock/Quantity.

Stop-Order Management: Each book has a stop-order status (on/off). If stock falls below a threshold, managers are notified unless a stop-order is applied.

Manager will perform following Actions: Update stock quantities, change book prices, set reorder thresholds, and view transaction logs. If stock reaches the reorder threshold, an email is sent to the manager, and items are automatically reordered unless a stop-order is active.

Promotions

Promotion Attributes: Percentage discount, expiration date.

Promotion Use: Available to members only; one promotion code per purchase.

Manager Actions: Create/update promotions, define discounts and expiration dates. Promotions are auto-deleted after expiration, and emails are sent to all members.

Checkout

Checkout is restricted to logged-in customers. During checkout:

A 16-digit credit card number is collected.

Transactions are logged and recorded.

Order details are sent via email.

Inventory stock is decremented based on purchased items.

Plant UML

```
@startuml

actor Customer
actor Manager

participant "AuthenticationService" as AuthService
participant "MembershipService" as MemberService
participant "CartService" as CartService
participant "InventoryService" as InventoryService
participant "PromotionService" as PromoService
participant "CheckoutService" as CheckoutService
participant "EmailService" as EmailService

Customer -> AuthService: login(username, password)
AuthService -> Customer: loginSuccess()
Customer -> AuthService: logout()
AuthService -> Customer: logoutSuccess()

Customer -> MemberService: signup(username, password, email, postalAddress,
memberStatus)
MemberService -> Customer: signupSuccess()

Customer -> CartService: addToCart(bookId)
CartService -> InventoryService: checkStock(bookId)
InventoryService -> CartService: stockAvailable()
CartService -> Customer: bookAddedToCart()

Manager -> InventoryService: updateStock(bookId, quantity)
InventoryService -> Manager: stockUpdateSuccess()

Manager -> InventoryService: changePrice(bookId, newPrice)
InventoryService -> Manager: priceUpdateSuccess()

Manager -> InventoryService: setReorderThreshold(bookId, threshold)
InventoryService -> Manager: thresholdSetSuccess()

Manager -> InventoryService: viewTransactionLogs()
InventoryService -> Manager: transactionLogs()

InventoryService -> InventoryService: checkReorderThreshold()
InventoryService -> EmailService: sendReorderNotification(bookId, managerEmail)
EmailService -> Manager: reorderNotificationEmail()

Manager -> InventoryService: specifyStopOrder(bookId, stopOrderStatus)
InventoryService -> Manager: stopOrderUpdateSuccess()

Manager -> PromoService: createPromotion(discountPercentage, expirationDate)
```

```
PromoService -> EmailService: sendPromotionEmail(promotion)
EmailService -> Customer: promotionEmailSent()
PromoService -> Manager: promotionCreated()

Customer -> CheckoutService: checkout(cart, creditCardNumber)
CheckoutService -> AuthService: verifyLogin()
AuthService -> CheckoutService: loginVerified()
CheckoutService -> InventoryService: updateInventory(cart)
InventoryService -> CheckoutService: inventoryUpdated()
CheckoutService -> EmailService: sendOrderEmail(orderDetails)
EmailService -> Customer: orderEmailSent()
CheckoutService -> Customer: checkoutSuccess()

@enduml
```

Similarity Score

V1	8
V2	8
E1	18
E2	28
Mv	6
ME	16
Node Similarity Ratio	0.75
Edge Similarity Ratio	0.57142857
Similarity Score	0.66071429

Prompt

Create state diagram in the Plant UML Diagram for the following system
Authentication- Customer will have the following Attributes: Username (unique), Password, Email, Postal Address, Member/Non-member status.
Manager will have the following Attributes: Hardcoded Username, Password, Email Address.
Login/Logout: Customers and managers can log in and log out; login enables logout and checkout, with a 30-minute session timeout. No password recovery available.
Membership-Customers can sign up to become members or remain non-members. Membership status is captured as a boolean value.

Shopping Cart-Customers can add one or more books to the shopping cart. Duplicate copies of the same book are not allowed.

Inventory Management

Book will have following Attributes: Title, Author, Price, Stock/Quantity.

Stop-Order Management: Each book has a stop-order status (on/off). If stock falls below a threshold, managers are notified unless a stop-order is applied.

Manager will perform following Actions: Update stock quantities, change book prices, set reorder thresholds, and view transaction logs. If stock reaches the reorder threshold, an email is sent to the manager, and items are automatically reordered unless a stop-order is active.

Promotions

Promotion Attributes: Percentage discount, expiration date.

Promotion Use: Available to members only; one promotion code per purchase.

Manager Actions: Create/update promotions, define discounts and expiration dates. Promotions are auto-deleted after expiration, and emails are sent to all members.

Checkout

Checkout is restricted to logged-in customers. During checkout:

A 16-digit credit card number is collected.

Transactions are logged and recorded.

Order details are sent via email.

Inventory stock is decremented based on purchased items.

PlantUML

```
Login and Logout State Diagram
@startuml
[*] --> LoggedOut

LoggedOut : User is logged out
LoggedOut --> LoggedIn : login()

LoggedIn : User is logged in
LoggedIn --> LoggedOut : logout()
LoggedIn --> LoggedOut : sessionTimeout()
@enduml
Signup State Diagram
@startuml
[*] --> NotSignedUp

NotSignedUp : User is not signed up
NotSignedUp --> SignedUp : signup()

SignedUp : User is signed up
@enduml
Membership State Diagram
@startuml
[*] --> NonMember

NonMember : User is not a member
NonMember --> Member : becomeMember()
```

```
Member : User is a member
@enduml
```

Add to Cart State Diagram

```
@startuml
[*] --> EmptyCart

EmptyCart : Shopping cart is empty
EmptyCart --> NonEmptyCart : addBook()

NonEmptyCart : Shopping cart has items
NonEmptyCart --> EmptyCart : removeBook()
@enduml
```

Maintain Inventory State Diagram

```
@startuml
[*] --> Inventory

Inventory : Inventory management
Inventory --> Inventory : updateStock()
Inventory --> Inventory : specifyStopOrder()
Inventory --> Inventory : notifyReorder()
@enduml
```

Specify Stop-Order State Diagram

```
@startuml
[*] --> StopOrderOff

StopOrderOff : Stop-order is off
StopOrderOff --> StopOrderOn : specifyStopOrder(on)

StopOrderOn : Stop-order is on
StopOrderOn --> StopOrderOff : specifyStopOrder(off)
@enduml
```

Notify Reorder State Diagram

```
@startuml
[*] --> AboveThreshold

AboveThreshold : Stock is above reorder threshold
AboveThreshold --> BelowThreshold : stockFallsBelowThreshold()

BelowThreshold : Stock is below reorder threshold
BelowThreshold --> AboveThreshold : stockReplenished()
@enduml
```

Create Promotions State Diagram

```
@startuml
[*] --> NoPromotion

NoPromotion : No active promotion
NoPromotion --> ActivePromotion : createPromotion()
```

```
ActivePromotion : Promotion is active
ActivePromotion --> NoPromotion : promotionExpires()
@enduml
```

Checkout State Diagram

```
@startuml
[*] --> NotCheckedOut

NotCheckedOut : User has not checked out
NotCheckedOut --> CheckedOut : checkout()

CheckedOut : User has checked out
@enduml
```

Similarity Score

V1	15
V2	13
E1	25
E2	20
Mv	12
ME	16
Node Similarity Ratio	0.8
Edge Similarity Ratio	0.64
Similarity Score	0.72