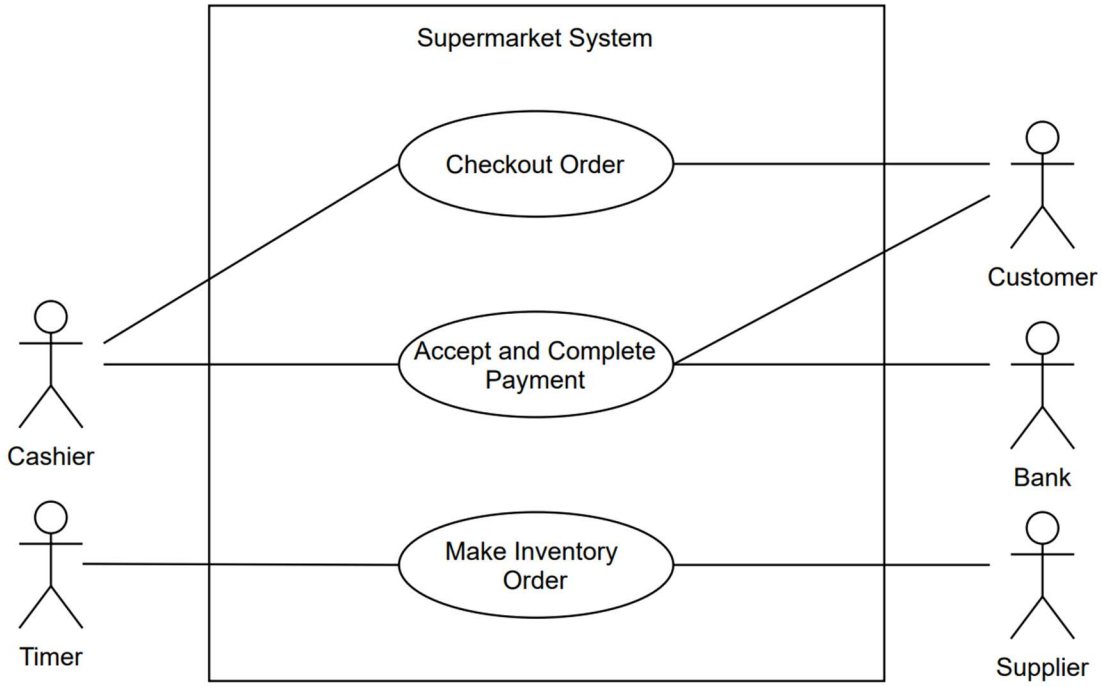
CS 3365 Software Engineering I – Supermarket System Project

## 1. Requirements Modeling:



**Use Case Name:** Checkout Order

**Summary:** Cashier scans customers items. Customer is prompted to enter loyalty membership information. After all items have been scanned, cashier prompts cash register to compute and display total.

**Actor:** Cashier, Customer

**Precondition:** Customer has at least one item.

**Main Sequence**

1. Cashier enters (or scans, but not possible without physical barcode scanner) product's identification number.
2. Product information is obtained from product inventory and updates inventory for the entered product.
3. Product information is sent to the receipt printer and displayed to the cashier and customer.
4. If loyalty membership is checked, go to step 1, or if all items are scanned, the cashier presses the TOTAL function key.
5. The total price including tax is calculated and the till opens.
6. If loyalty membership is verified, total price is sent to the customer's account to calculate and add credit points.
7. Total price is sent to the receipt printer and displayed to the cashier and customer.

**Alternative Sequences:**

Step 3: Product is a bulk item and must be weighed on the scale. The cashier is notified to weigh product. Cashier presses the SCALE function key and final weight is calculated. Continue to step 3 including weight and price in product information.

Step 4: If loyalty membership is not checked, the system prompts the customer, and the customer enters the phone number and member PIN to verify membership, or cancels. Go to step 1, or if all items are scanned, the cashier presses the TOTAL function key.

Step 6: Customer is not a loyal member; total price is not sent to the customer's account. Continue to step 7.

**Postcondition:** Cashier has pressed TOTAL function key and order total has been calculated.

**Use Case Name:** Accept and Complete Payment

**Summary:** Cashier accepts payment type of debit/credit card, check, or cash and receipt is printed.

**Actor:** Cashier, Customer, Bank

**Precondition:** All items have been scanned and cashier has pressed total function key.

**Main Sequence**

1. If customer pays with a credit/debit card, the cashier enters payment amount and presses the credit/debit type button.
2. Customer inserts a card into the credit/debit card reader and is prompted to insert PIN.
3. Request is sent to credit/debit authorization center to authorize card and card number is sent to receipt printer.
4. If approved, an authorization code is returned.
5. The authorization code is sent to the receipt printer.
6. The amount of change is computed.
7. Amount of change is sent to the receipt printer and displayed to the cashier and customer.
8. The receipt is printed.

**Alternative Sequences:**

Step 1: If customer pays with cash, the cashier enters payment amount and presses the cash type button. Continue to step 6. If customer pays with check, the cashier enters payment amount and presses the check type button. The check is scanned by the check reader (Verified by Cashier due to physical limitation in implementation). If verified, the cashier is prompted to place check in receipt printer. A line containing date, time, store identity, cashier identity, and order number are printed on the check. Continue to step 6.

Step 4: If card request is denied and no other payment is offered, the order is cancelled. If customer uses another card, continue to step 1. If customer uses cash or check, continue to step 1 in alternatives.

**Postcondition:** Order has been paid and completed.

**Use Case Name:** Make Inventory Order

**Summary:** At the end of the night, inventory orders with quantities of products are sent to the supplier and recorded in the system.

**Actor:** Timer, Supplier

**Precondition:** At least one inventory message was created during the day.

**Main Sequence**

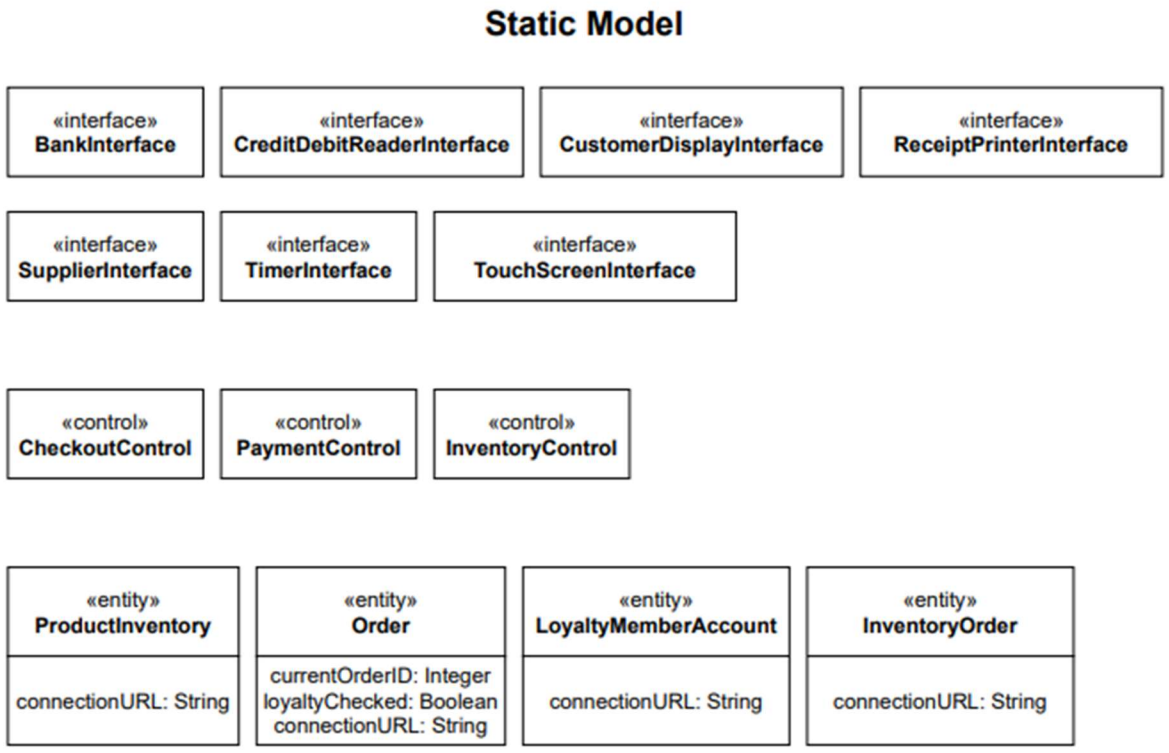
1. Timer inputs current time.
2. Inventory messages are searched for products below desired level of stock.
3. If messages are found, order is created with products below desired level of stock.
4. Order is stored in system and sent to supplier.

**Alternative Sequences:**

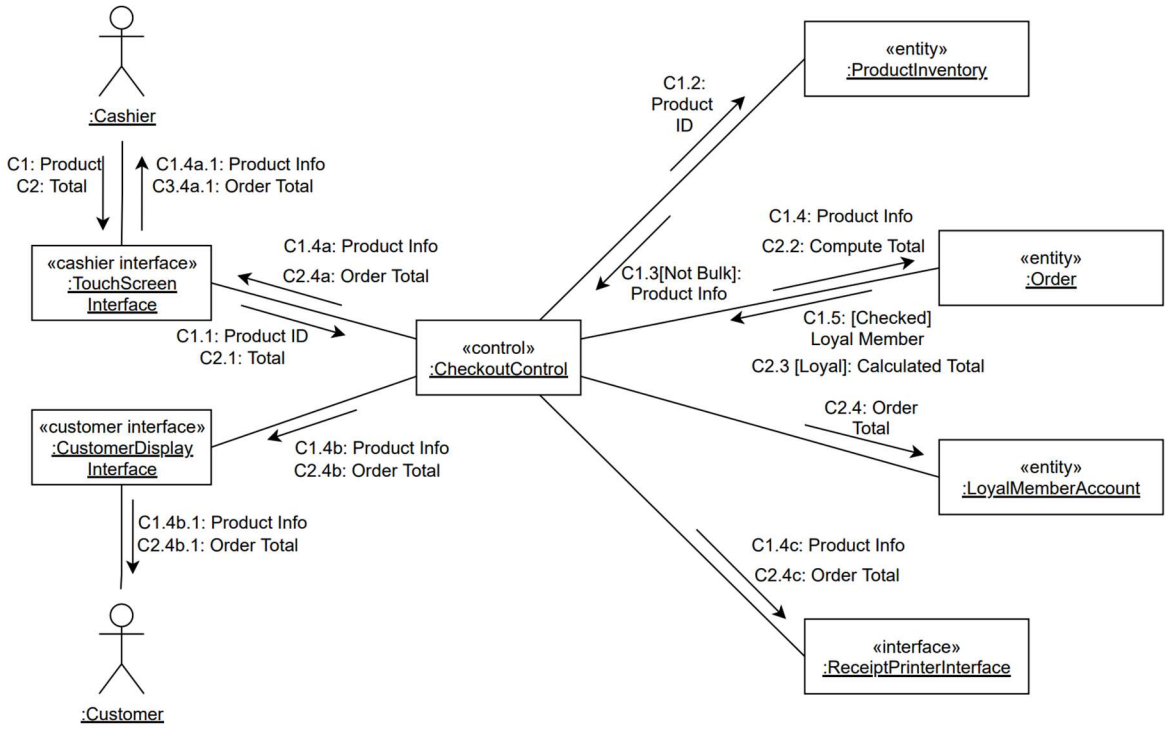
Step 3: No products were below desired level of stock. No order created.

**Postcondition:** Inventory Order is created and sent to suppliers.

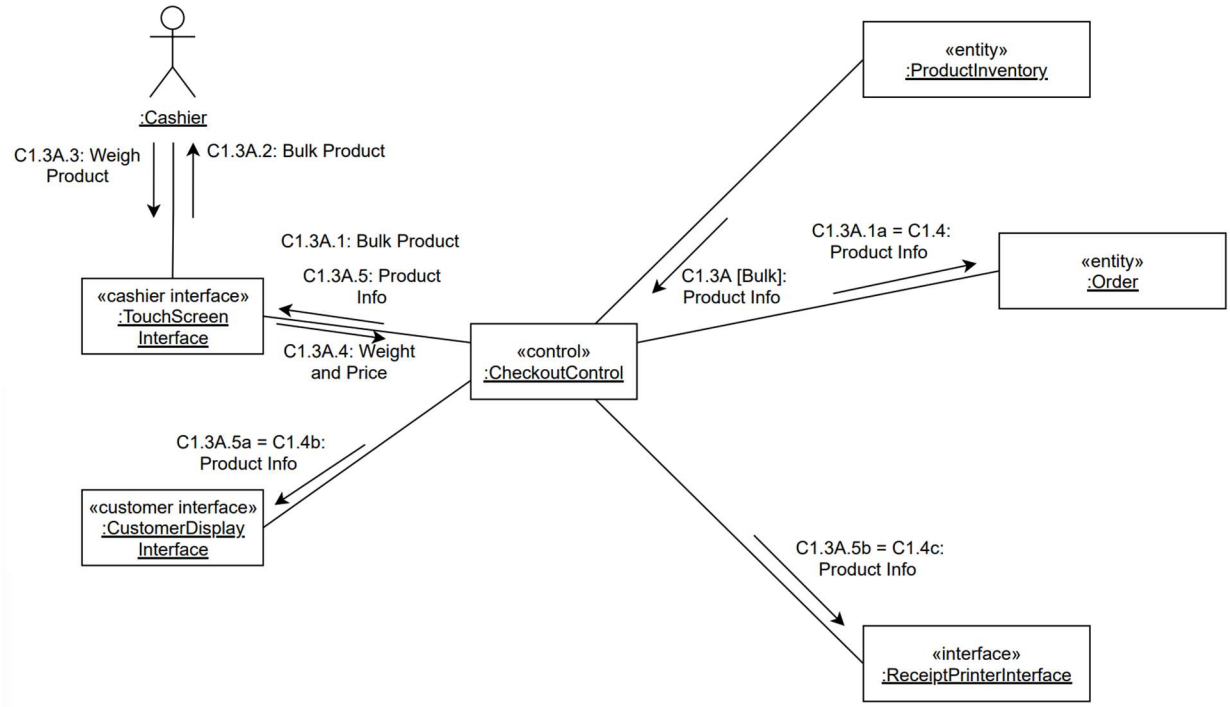
## 2. Requirements Analysis:



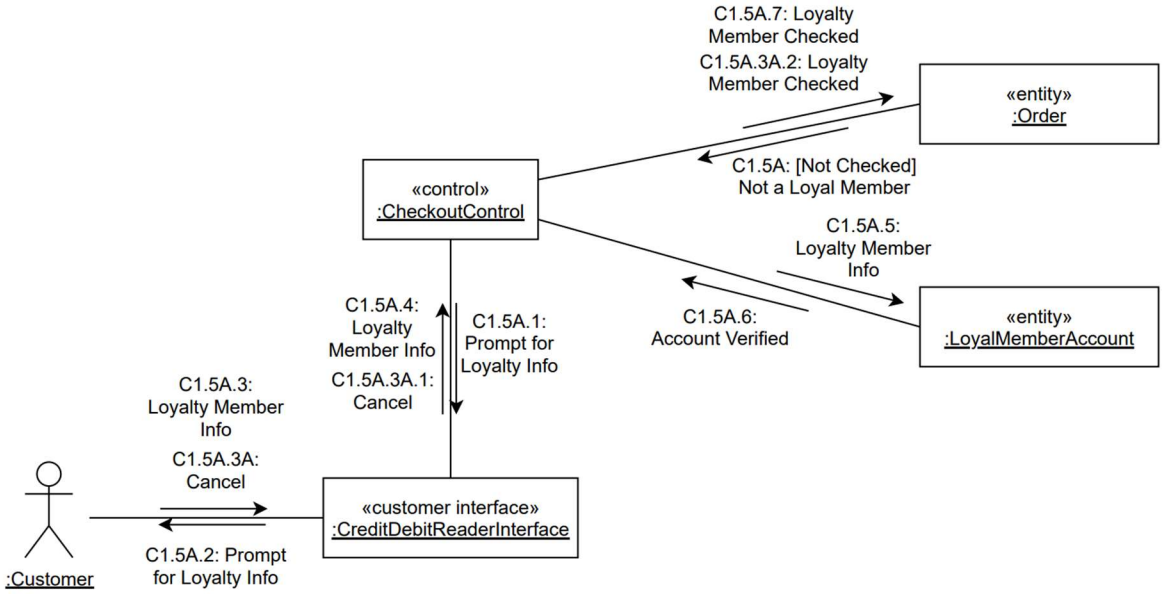
Checkout Order Use Case
(Main)



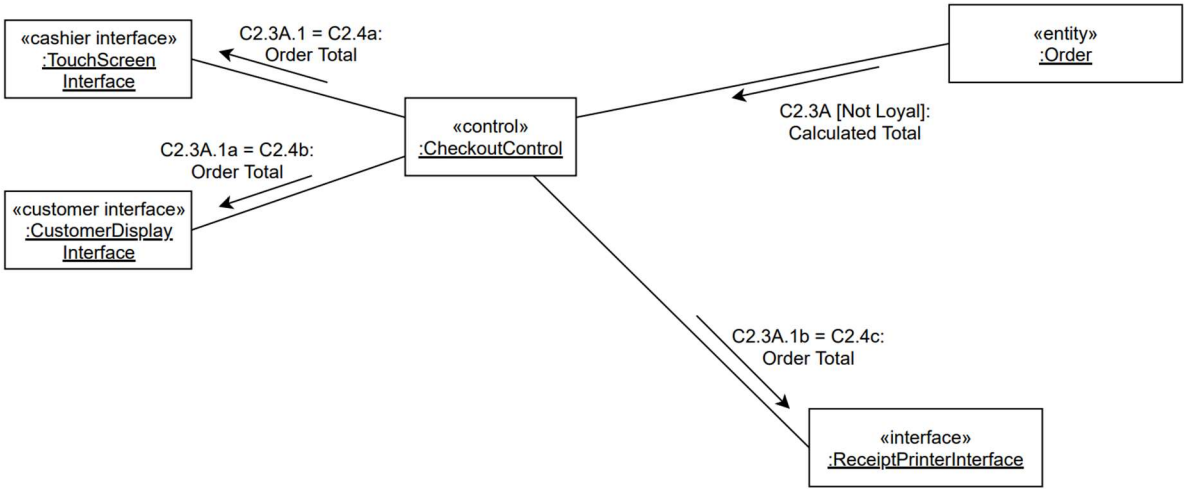

Checkout Order Use Case
- Bulk Item (Alternative)



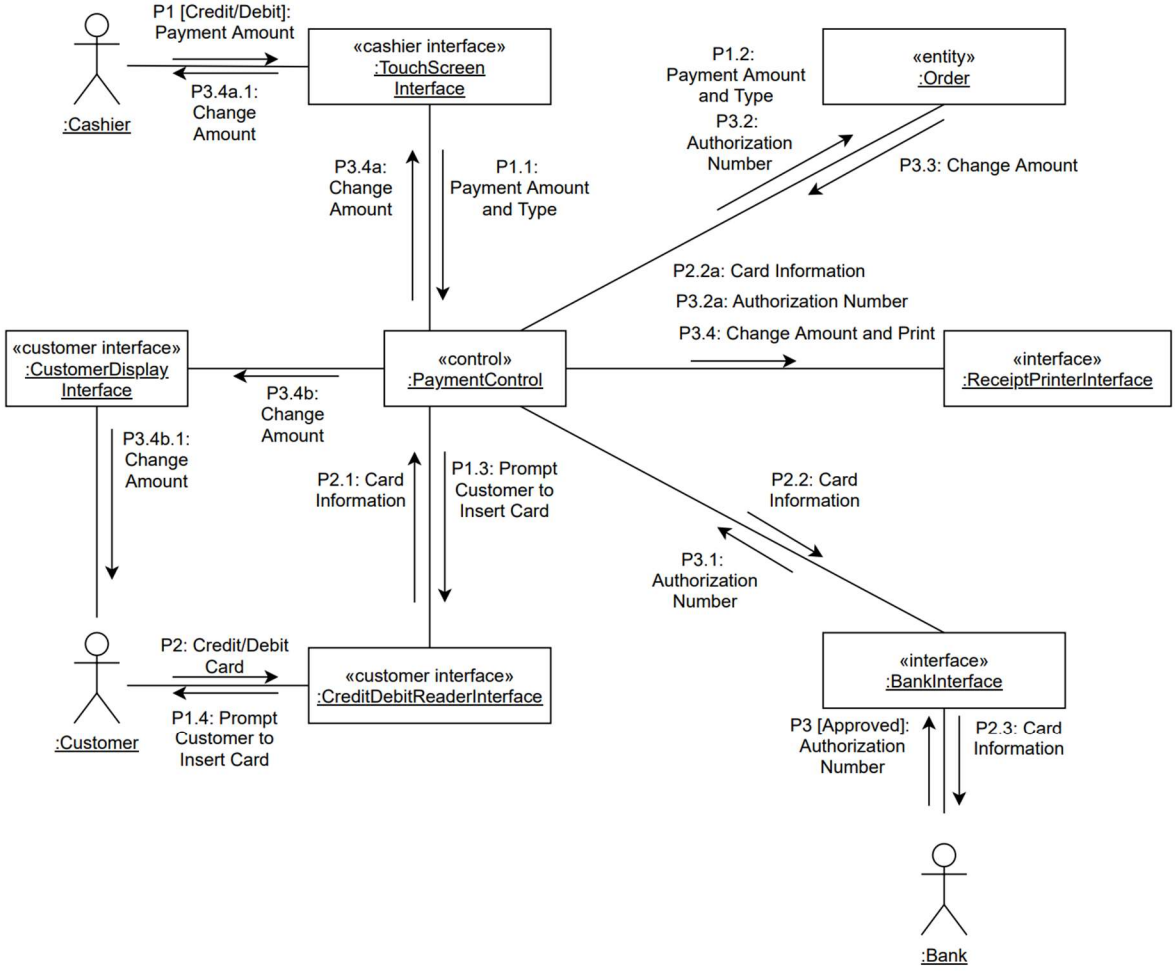
Checkout Order Use Case -
Loyalty Not Checked (Alternative)

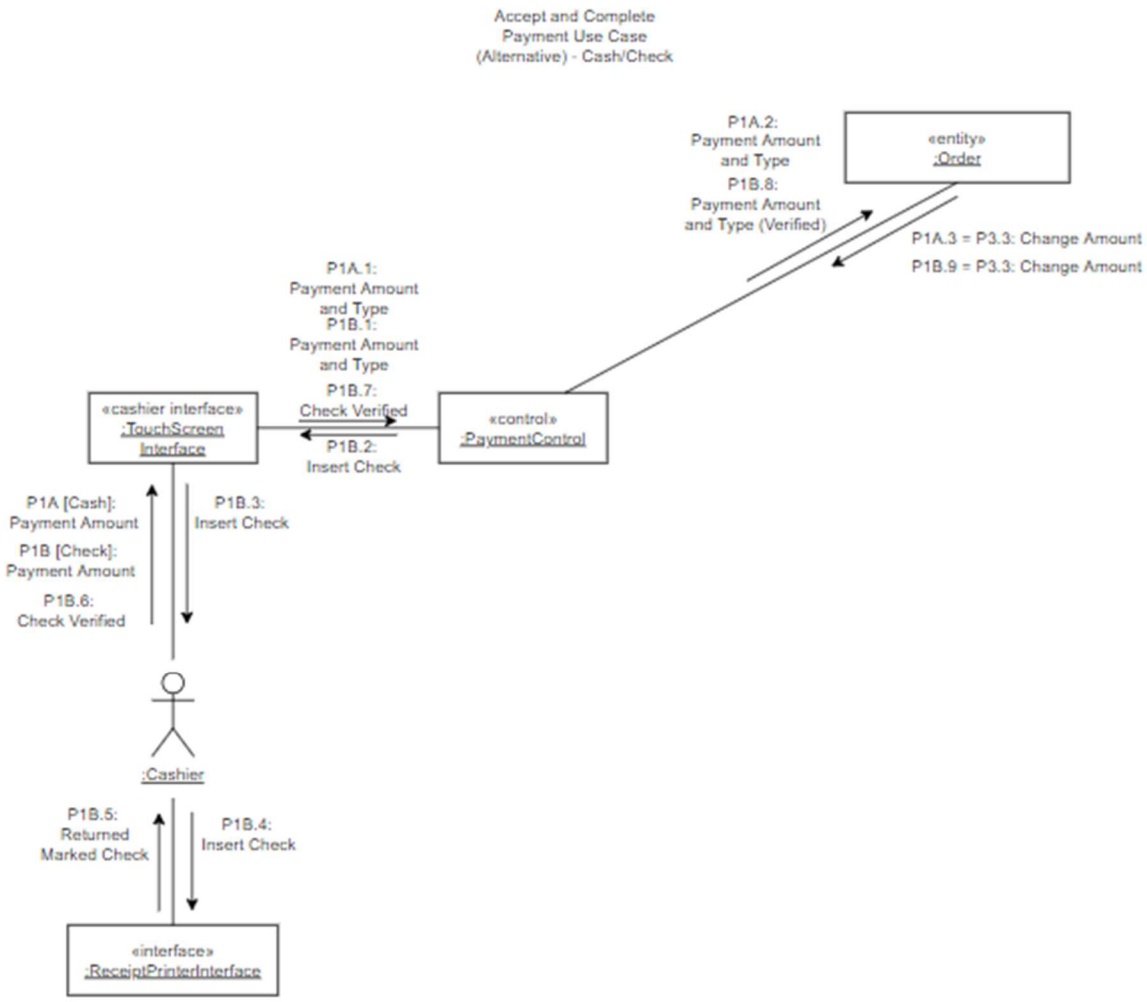


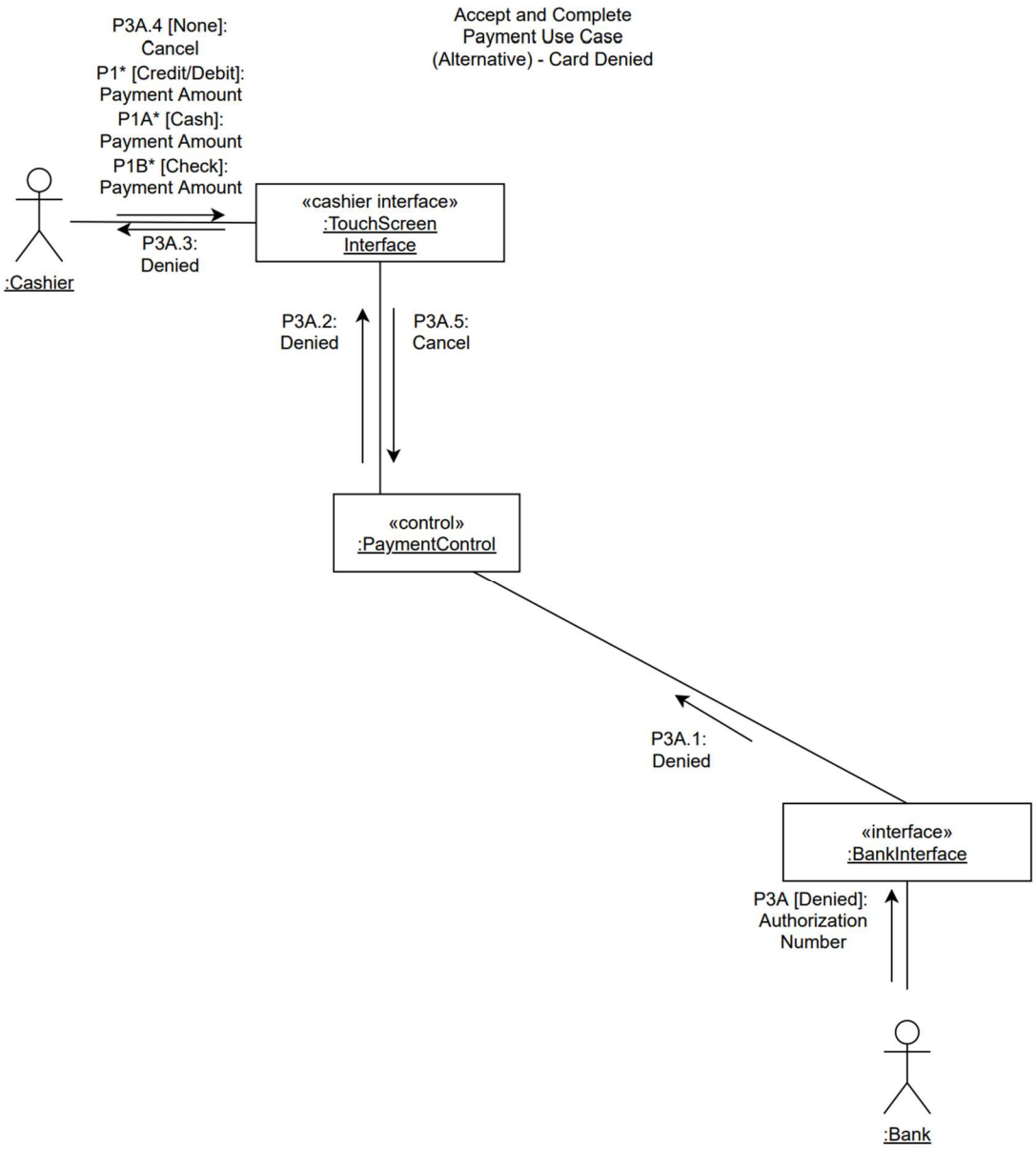

Checkout Order Use Case -
Not Loyal Member (Alternative)

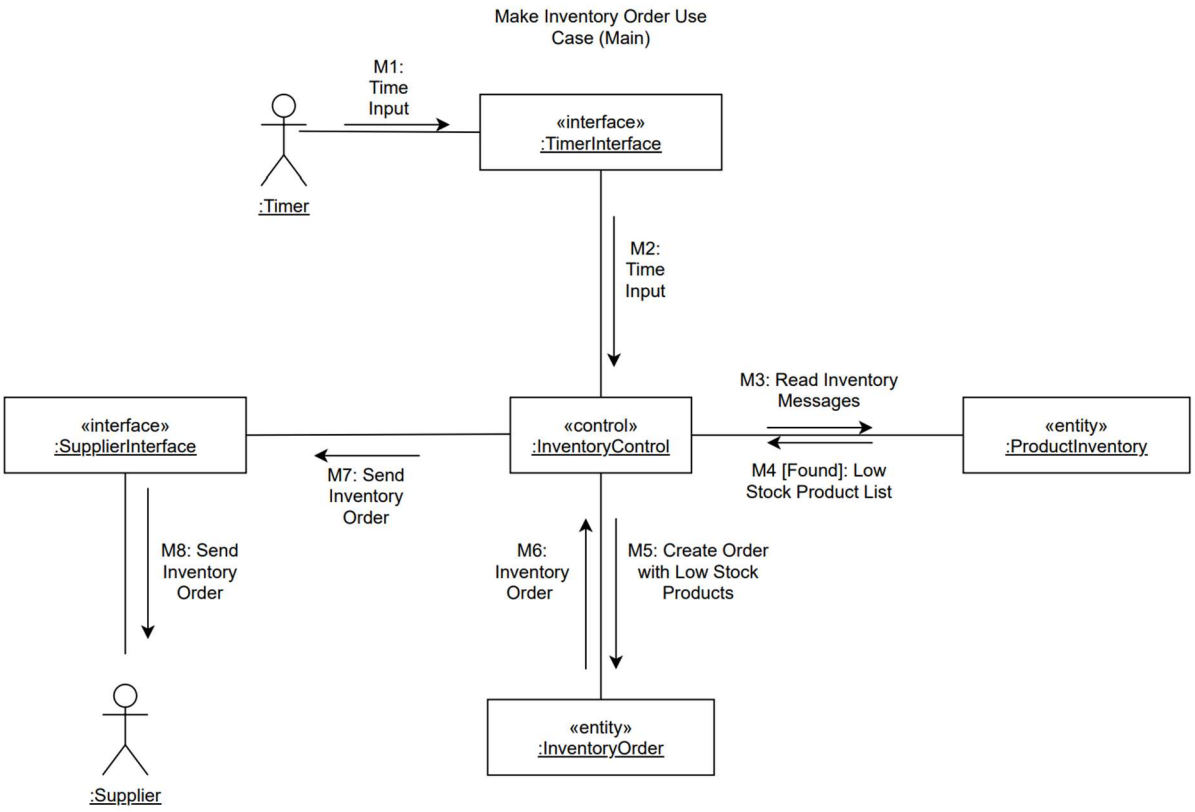


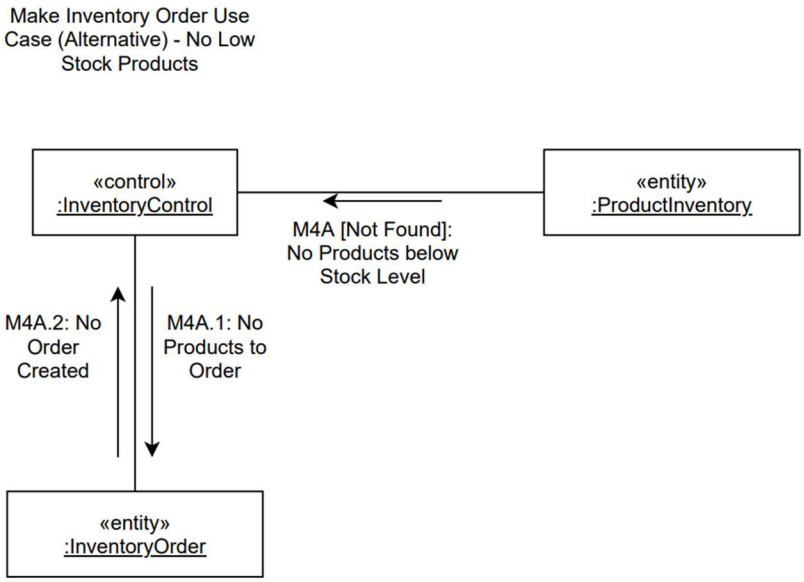
Accept and Complete
Payment Use Case (Main)




Accept and Complete
Payment Use Case
(Alternative) - Cash/Check
P1A.2:
Payment Amount
and Type
P1B.8:
Payment Amount
and Type (Verified)
«entity»
:Order
P1A.3 = P3.3: Change Amount
P1B.9 = P3.3: Change Amount
P1A.1:
Payment Amount
and Type
P1B.1:
Payment Amount
and Type
P1B.7:
Check Verified
P1B.2:
Insert Check
«cashier interface»
:TouchScreen
Interface
«control»
:PaymentControl
P1A [Cash]:
Payment Amount
P1B [Check]:
Payment Amount
P1B.6:
Check Verified
P1B.3:
Insert Check
:Cashier
P1B.5:
Returned
Marked Check
P1B.4:
Insert Check
«interface»
:ReceiptPrinterInterface


Accept and Complete
Payment Use Case
(Alternative) - Card Denied
P3A.4 [None]:
Cancel
P1* [Credit/Debit]:
Payment Amount
P1A* [Cash]:
Payment Amount
P1B* [Check]:
Payment Amount
:Cashier
«cashier interface»
:TouchScreen
Interface
P3A.3:
Denied
P3A.2:
Denied
P3A.5:
Cancel
«control»
:PaymentControl
P3A.1:
Denied
«interface»
:BankInterface
P3A [Denied]:
Authorization
Number
:Bank


Make Inventory Order Use Case (Main)
M1: Time Input
:Timer
«interface» :TimerInterface
M2: Time Input
«interface» :SupplierInterface
«control» :InventoryControl
M3: Read Inventory Messages
M4 [Found]: Low Stock Product List
«entity» :ProductInventory
M7: Send Inventory Order
M8: Send Inventory Order
M6: Inventory Order
M5: Create Order with Low Stock Products
:Supplier
«entity» :InventoryOrder


Make Inventory Order Use Case (Alternative) - No Low Stock Products
«control» :InventoryControl
«entity» :ProductInventory
M4A [Not Found]: No Products below Stock Level
M4A.2: No Order Created
M4A.1: No Products to Order
«entity» :InventoryOrder

# Static Model with Operations

| <<Interface>><br>BankInterface |
| --- |
| |
| + authorizeCreditDebit (cardNumber : String, expDate : String, CVV : Integer, billingZipCode : Integer) : Integer |

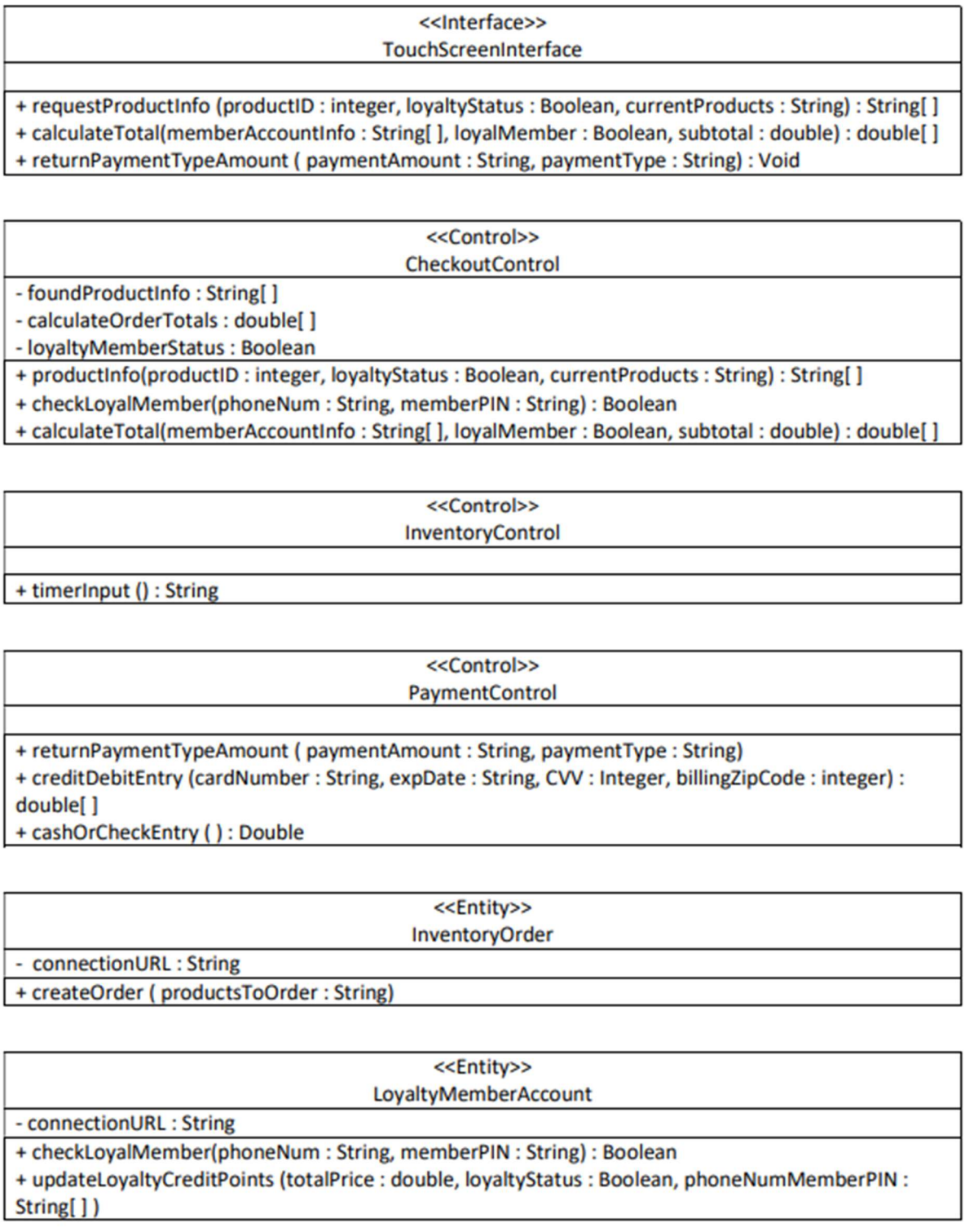
| <<Interface>><br>CreditDebitReaderInterface |
| --- |
| |
| + loyalMemberEntry (phoneNum : String, memberPIN : String) : Boolean<br>+ creditDebitEntry (cardNumber : String, expDate : String, CVV : Integer, billingZipCode : integer) : double[ ]<br>+ cashOrCheckEntry () : double |

| <<Interface>><br>CustomerDisplayInterface |
| --- |
| |
| + returnProductInfoCustomer (foundInfoCustomer : String[ ]) : String[ ]<br>+ returnOrderTotal (orderTotal : double[ ]) : double[ ]<br>+ returnChangeAmount (changeAmount : double) : double |

| <<Interface>><br>ReceiptPrinterInterface |
| --- |
| |
| + returnProductInfoReceipt ( foundInfoReceipt : String[ ] ) : String[ ]<br>+ returnOrderTotal (orderTotal : double[ ]) : double[ ]<br>+ returnCardNumber (cardNum : String) : String<br>+ returnAuthNumber ( authNum : Integer ) : Integer<br>+ returnChangeAmountAndPrint ( changeAmount : double) : double |

| <<Interface>><br>SupplierInterface |
| --- |
| |
| + sendOrder(productsToOrder : String) : String |

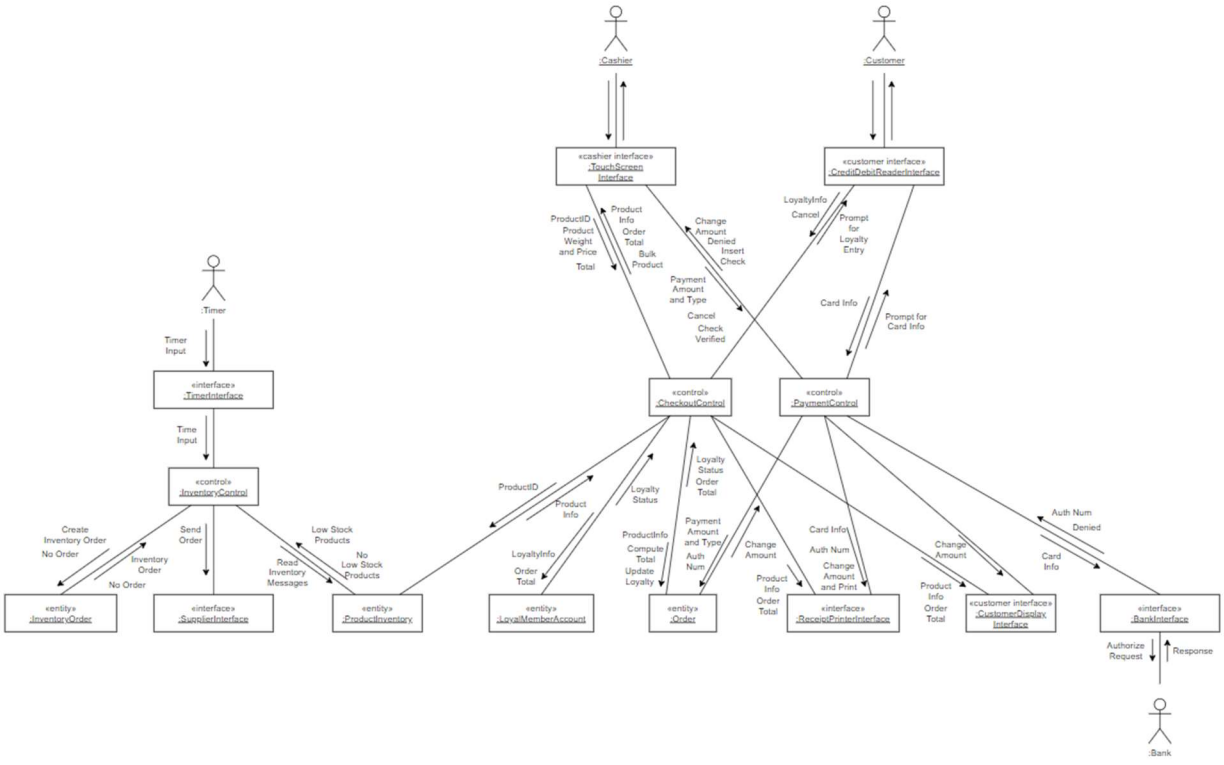
| <<Interface>><br>TimerInterface |
| --- |
| |
| + timerInput() : String |

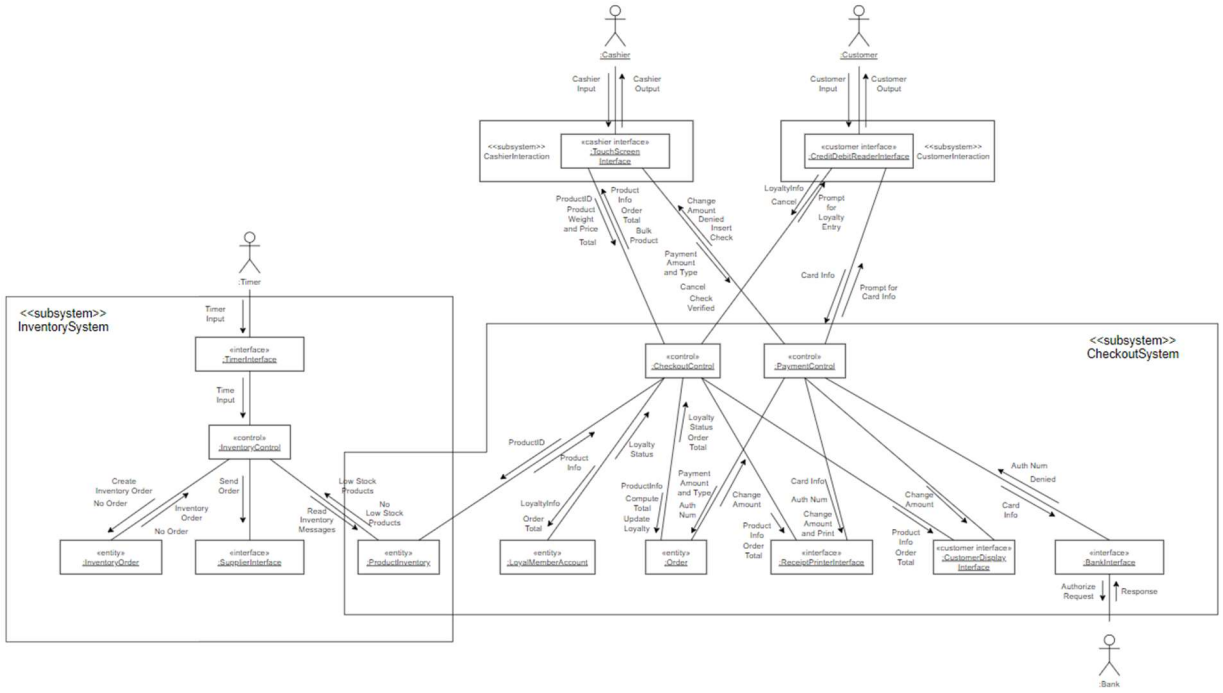

<<Interface>>
TouchScreenInterface
+ requestProductInfo (productID : integer, loyaltyStatus : Boolean, currentProducts : String) : String[ ]
+ calculateTotal(memberAccountInfo : String[ ], loyalMember : Boolean, subtotal : double) : double[ ]
+ returnPaymentTypeAmount ( paymentAmount : String, paymentType : String) : Void
<<Control>>
CheckoutControl
- foundProductInfo : String[ ]
- calculateOrderTotals : double[ ]
- loyaltyMemberStatus : Boolean
+ productInfo(productID : integer, loyaltyStatus : Boolean, currentProducts : String) : String[ ]
+ checkLoyalMember(phoneNum : String, memberPIN : String) : Boolean
+ calculateTotal(memberAccountInfo : String[ ], loyalMember : Boolean, subtotal : double) : double[ ]
<<Control>>
InventoryControl
+ timerInput () : String
<<Control>>
PaymentControl
+ returnPaymentTypeAmount ( paymentAmount : String, paymentType : String)
+ creditDebitEntry (cardNumber : String, expDate : String, CVV : Integer, billingZipCode : integer) : double[ ]
+ cashOrCheckEntry ( ) : Double
<<Entity>>
InventoryOrder
- connectionURL : String
+ createOrder ( productsToOrder : String)
<<Entity>>
LoyaltyMemberAccount
- connectionURL : String
+ checkLoyalMember(phoneNum : String, memberPIN : String) : Boolean
+ updateLoyaltyCreditPoints (totalPrice : double, loyaltyStatus : Boolean, phoneNumMemberPIN : String[ ] )

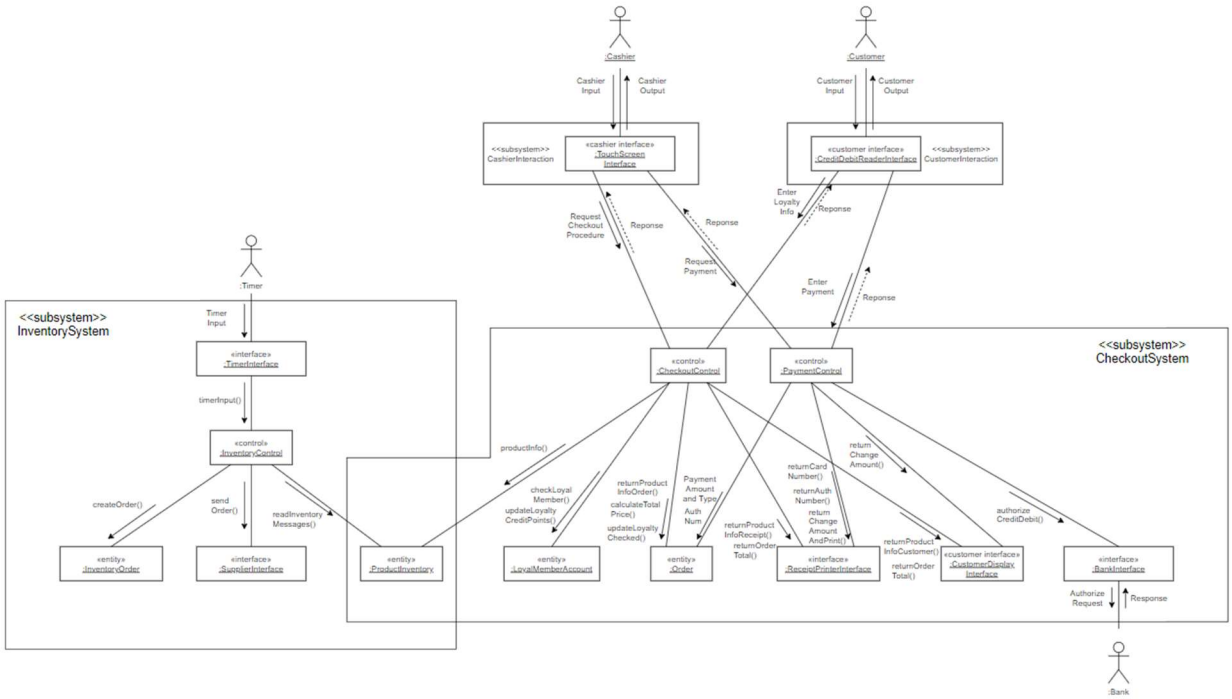
| <<Entity>><br>Order |
|---|
| - currentOrderID : Integer<br>- loyaltyChecked : Boolean<br>- connectionURL : String |
| + returnProductInfoOrder ( productID : Integer, currentStatus : Boolean, currentProducts : String )<br>+ updatedLoyaltyChecked (memberStatus : boolean) : Boolean<br>+ calculateTotalPrice ( subtotal : double ) : double[ ]<br>+ updateOrderTotal ( subTotalAndTotal : double[ ] )<br>+ updatePaymentTypeAmount ( paymentAmount : String, paymentType : String )<br>+ updateAuthorizationNumCalculateChange (authorizationNum : Integer ) : double<br>+ cashOrCheckChange ( ) : double |

| <<Entity>><br>ProductInventory |
|---|
| - connectionURL : String |
| + productInfo ( productID : Integer ) : String[ ] |

## 3. Design:




:Cashier
:Customer
Cashier Input
Cashier Output
Customer Input
Customer Output
<<subsystem>> CashierInteraction
«cashier interface» :TouchScreen Interface
«customer interface» :CreditDebitReaderInterface
<<subsystem>> CustomerInteraction
ProductID
Product Weight and Price
Total
Product Info
Order Total
Bulk Product
Change Amount Denied
Insert Check
Payment Amount and Type
Cancel
Check Verified
LoyaltyInfo
Cancel
Prompt for Loyalty Entry
Card Info
Prompt for Card Info
:Timer
<<subsystem>> InventorySystem
Timer Input
«interface» :TimerInterface
Time Input
«control» :InventoryControl
Create Inventory Order
No Order
Inventory Order
No Order
Send Order
Low Stock Products
Read Inventory Messages
No Low Stock Products
«entity» :InventoryOrder
«interface» :SupplierInterface
«entity» :ProductInventory
<<subsystem>> CheckoutSystem
«control» :CheckoutControl
«control» :PaymentControl
ProductID
Product Info
LoyaltyInfo
Order Total
Loyalty Status
ProductInfo
Compute Total
Update Loyalty
Loyalty Status
Order Total
Payment Amount and Type
Auth Num
Change Amount
Card Info
Auth Num
Change Amount and Print
Product Info
Order Total
Change Amount
Auth Num
Denied
Card Info
«entity» :LoyalMemberAccount
«entity» :Order
«interface» :ReceiptPrinterInterface
«customer interface» :CustomerDisplay Interface
«interface» :BankInterface
Authorize Request
Response
:Bank


:Cashier
:Customer
Cashier Input
Cashier Output
Customer Input
Customer Output
<<subsystem>> CashierInteraction
«cashier interface» :TouchScreen Interface
«customer interface» :CreditDebitReaderInterface
<<subsystem>> CustomerInteraction
Request Checkout Procedure
Reponse
Reponse
Request Payment
Enter Loyalty Info
Reponse
Enter Payment
Reponse
:Timer
<<subsystem>> InventorySystem
Timer Input
«interface» :TimerInterface
timerInput()
«control» :InventoryControl
createOrder()
send Order()
readInventory Messages()
«entity» :InventoryOrder
«interface» :SupplierInterface
«entity» :ProductInventory
<<subsystem>> CheckoutSystem
«control» :CheckoutControl
«control» :PaymentControl
productInfo()
checkLoyal Member()
updateLoyalty CreditPoints()
returnProduct InfoOrder()
calculateTotal Price()
updateLoyalty Checked()
Payment Amount and Type
Auth Num
returnProduct InfoReceipt()
returnOrder Total()
returnCard Number()
returnAuth Number()
return Change Amount AndPrint()
return Change Amount()
returnProduct InfoCustomer()
returnOrder Total()
authorize CreditDebit()
«entity» :LoyalMemberAccount
«entity» :Order
«interface» :ReceiptPrinterInterface
«customer interface» :CustomerDisplay Interface
«interface» :BankInterface
Authorize Request
Response
:Bank

# Database Tables

| ProductInventory | |
|---|---|
| PK | **ProductID INT** |
| | ProductInfo VARCHAR(100) |
| | ProductPrice DECIMAL(18,2) |
| | BulkProduct VARCHAR(50) |
| | DiscountInfo DECIMAL(18,2) |
| | StockLevel INT |
| | InventoryMessage VARCHAR(50) |

| Order | |
|---|---|
| PK | **OrderID INT** |
| | Products VARCHAR(200) |
| | LoyaltyMemberStatus VARCHAR(50) |
| | OrderSubtotal DECIMAL(18,2) |
| | OrderTotal DECIMAL(18,2) |
| | PaymentAmount DECIMAL (18,2) |
| | PaymentType VARCHAR(50) |
| | AuthorizationNum INT |

| LoyaltyMembers | |
|---|---|
| PK | **MemberID INT** |
| | Name VARCHAR(200) |
| | PhoneNum VARCHAR(100) |
| | Email VARCHAR(150) |
| | MemberPin INT |
| | CreditPoints INT |

| Bank | |
|---|---|
| PK | **AccountID INT** |
| | CardNumber VARCHAR(100) |
| | ExpDate VARCHAR(50) |
| | CVV INT |
| | ZipCode INT |

| InventoryOrder | |
|---|---|
| PK | **InventoryOrderID INT** |
| | ProductsAndQuantities VARCHAR(200) |
| | OrderStatus VARCHAR(50) |