CRM Order Management System

Scenario

The CRM Order Management System is a software platform designed to handle customer orders, manage product inventory, process payments, and track the interaction with external systems such as supplier management platforms. The system provides an interface for customers to place orders and manage their profiles.

Core Features:
Order Placement and Tracking: Customers can place orders via a web application and track them in real-time.
Payment Processing: The system integrates with an external payment gateway to securely process payments.
Supplier Interaction: Orders are sent to external supplier systems for fulfillment.
Product and Inventory Management: Products are managed within the system, and their availability is updated when an order is placed.

Architecture Components:
Web Application: The user interface for customers to interact with the system.
API: Handles business logic and interactions with external services.
Database: Stores customer, product, order, and inventory data.

UML Diagrams with plantUML codes

1) Class Diagram

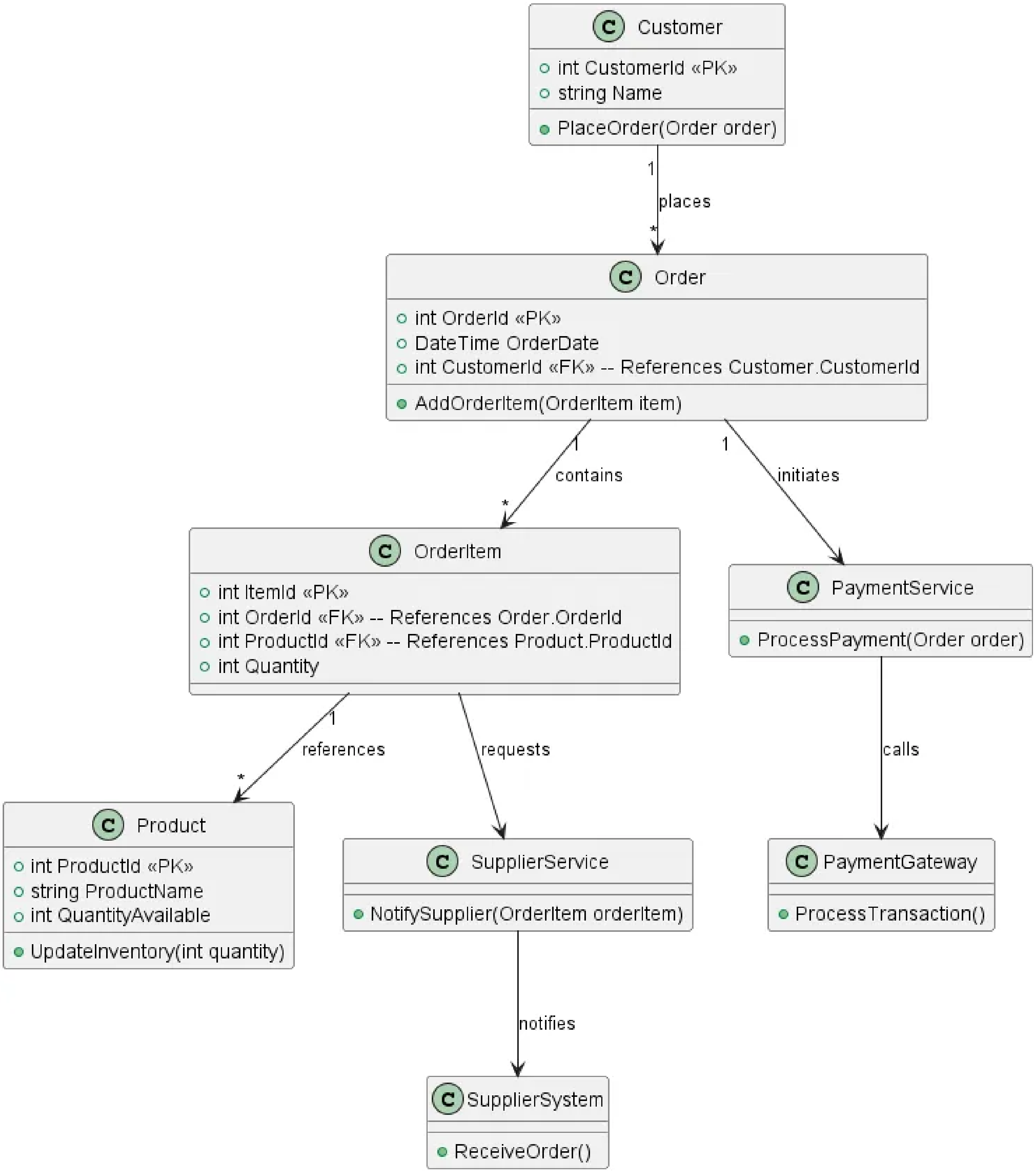
PlantUML code:

```
@startuml

title CRM Order Management System - C4 - Class Diagram

class Customer {
  + int CustomerId <<PK>>
  + string Name
  + PlaceOrder(Order order)
}

class Order {
  + int OrderId <<PK>>
  + DateTime OrderDate
  + int CustomerId <<FK>> -- References Customer.CustomerId
  + AddOrderItem(OrderItem item)
}
```

```
class OrderItem {
  + int ItemId <<PK>>
  + int OrderId <<FK>> -- References Order.OrderId
  + int ProductId <<FK>> -- References Product.ProductId
  + int Quantity
}

class Product {
  + int ProductId <<PK>>
  + string ProductName
  + int QuantityAvailable
  + UpdateInventory(int quantity)
}

class PaymentService {
  + ProcessPayment(Order order)
}

class SupplierService {
  + NotifySupplier(OrderItem orderItem)
}

class PaymentGateway {
  + ProcessTransaction()
}

class SupplierSystem {
  + ReceiveOrder()
}

Customer "1" --> "*" Order : places
Order "1" --> "*" OrderItem : contains
OrderItem "1" --> "*" Product : "references"
Order "1" --> PaymentService : initiates
PaymentService --> PaymentGateway : calls
OrderItem --> SupplierService : "requests"
SupplierService --> SupplierSystem : notifies
@enduml
```

Diagram:

CRM Order Management System - C4 - Class Diagram

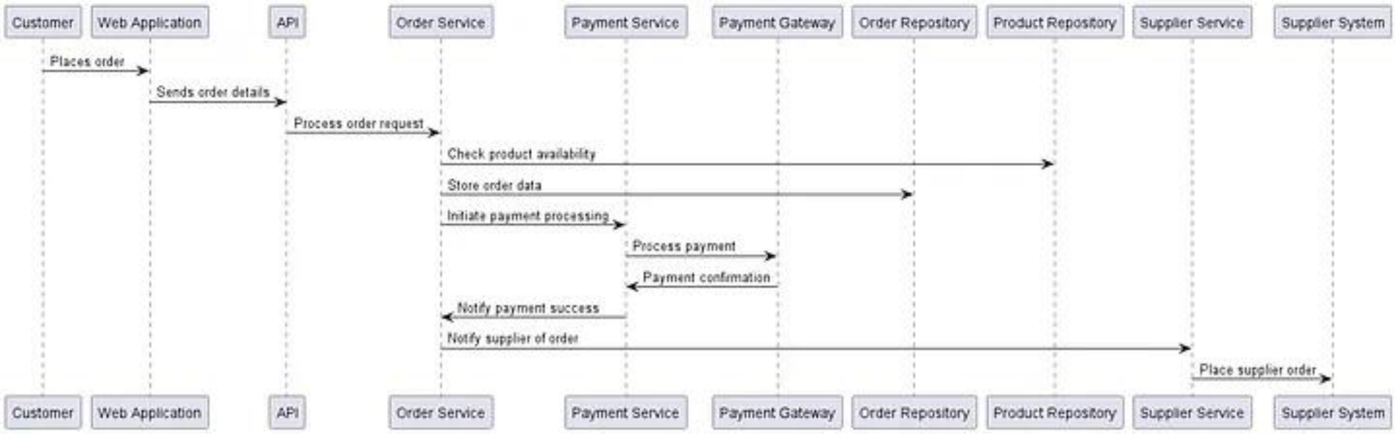


2) Sequence Diagram

PlantUML code:

```
@startuml

participant "Customer" as Customer
participant "Web Application" as WebApp
participant "API" as API
participant "Order Service" as OrderService
participant "Payment Service" as PaymentService
participant "Payment Gateway" as PaymentGateway
participant "Order Repository" as OrderRepository
participant "Product Repository" as ProductRepository
participant "Supplier Service" as SupplierService
participant "Supplier System" as SupplierSystem

' Step 1: Customer places an order on the Web Application
Customer -> WebApp : Places order

' Step 2: Web Application sends order data to API
WebApp -> API : Sends order details

' Step 3: API forwards order to Order Service
API -> OrderService : Process order request

' Step 4: Order Service validates order and stores it in the database
OrderService -> ProductRepository : Check product availability
OrderService -> OrderRepository : Store order data

' Step 5: Payment Service processes the payment
OrderService -> PaymentService : Initiate payment processing
PaymentService -> PaymentGateway : Process payment

' Step 6: Payment Gateway confirms the payment
PaymentGateway -> PaymentService : Payment confirmation
PaymentService -> OrderService : Notify payment success

' Step 7: Order Service notifies Supplier Service
OrderService -> SupplierService : Notify supplier of order

' Step 8: Supplier Service sends order to external Supplier System
SupplierService -> SupplierSystem : Place supplier order

@enduml
```

Diagram:

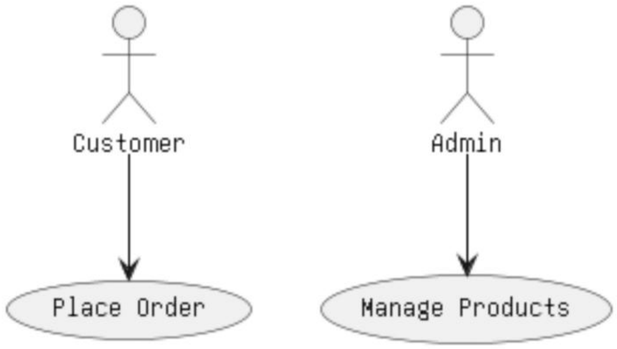


3) Use Case Diagram

PlantUML code:

```
@startuml
actor Customer
actor Admin
usecase "Place Order" as UC1
usecase "Manage Products" as UC2
Customer --> UC1
Admin --> UC2
@enduml
```

Diagram:



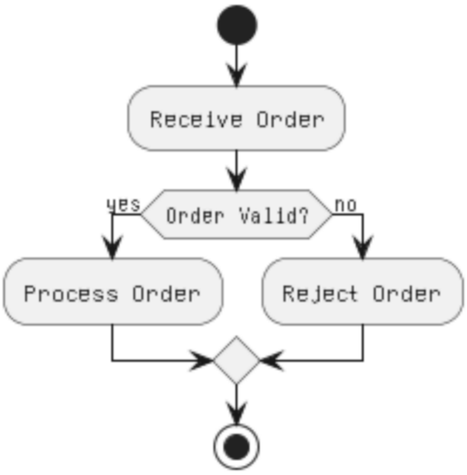
4) Activity Diagram

PlantUML code:

```
@startuml
start
:Receive Order;
if (Order Valid?) then (yes)
  :Process Order;
else (no)
```

```
  :Reject Order;
endif
stop
@enduml
```

Diagram:

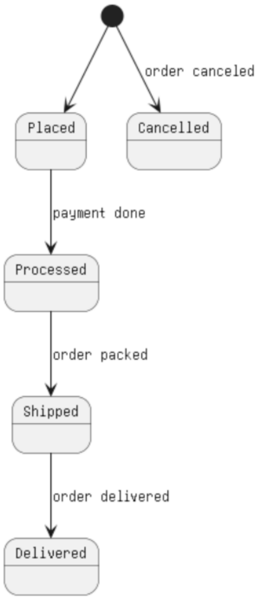


5) State Diagram

PlantUML code:

```
@startuml
[*] --> Placed
Placed --> Processed : payment done
Processed --> Shipped : order packed
Shipped --> Delivered : order delivered
[*] --> Cancelled : order canceled
@enduml
```

Diagram:

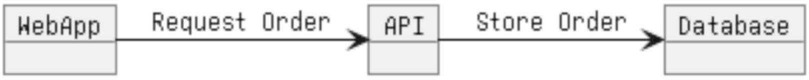


6) Collaboration Diagram

PlantUML code:

```
@startuml
object WebApp
object API
object Database
WebApp -> API : Request Order
API -> Database : Store Order
@enduml
```

Diagram:


WebApp
Request Order
API
Store Order
Database