

Question1: domain design

After considering Cafe Bluebik needs, we can breakdown requirement into 3 system to implement

1. online shopping and ordering system
2. payment system
3. notification system

currently we already have 3 domains

1. ms-user
2. ms-shop
3. ms-product

we should introduce new 4 domains including

4. ms-cart
5. ms-order
6. ms-payment
7. ms-notification

foreach new domain will handle specific responsibilities following

1. ms-cart

this cart service will manage user selected/de-selected products and add to cart and then place a order

responsibilities

- get products in cart
- add products to cart
- remove products from cart
- update product quantities in cart

- calculate total product price

apis requirement

1. get items in cart [GET]: <cart_service_url>/api/v1/cart
 - a. get all items and total price in cart for user
2. add/remove/update product [PUT]: <cart_service_url>/api/v1/cart
 - a. to simplify api, use only single endpoint to manage item in cart, Frontend manage items state (calculate quantitie) in cart before sending data to Backend

2. ms-order

this order service will manage order lifecycle start from creation, making a payment unti order finish.

responsibilities

- create an order
- update order status
- integrate with payment service to make transaction
- integrate with notification service to notify user when tranasaction is complete

apis requirement

1. creata new order [POST]: <order_service_url>/api/v1/orders
 - a. initiate order with pending status after cart checkout confirm
2. pay an order [POST]: <order_service_url>/api/v1/orders/<order_id>/payment
 - a. initiate transaction with pending status after pay confirm

3. ms-payment

this payment service will manage transaction process with/without third party.

responsibilities

- get available payment methods
- validate payment method

- create transaction
- update transaction status

apis requirement

1. get available payment methods [GET]:
 <payment_service_url>/api/v1/payment/methods
 a. send back available payment methods for user
2. validate payment method: [POST]:
 <payment_service_url>/api/v1/payment/validate
 a. validate payment before start transaction
3. create new transaction [POST]: <payment_service_url>/api/v1/transactions
 a. start financial transaction
4. update transaction status [PATCH]:
 <payment_service_url>/api/v1/transactions/status
 a. callback url for third party update transaction status

4. ms-notification

this notification service will send notification to user base on which event in this case is payment successful

responsibilities

- notify user after order payment success

apis requirement

1. send notification [POST]: <notification_service_url>/api/v1/notifications
 a. send push notification to user related to order event
2. update notification status [PATCH]:
 <notification_service_url>/api/v1/notifications/status
 a. callback url for consumer/third party update notification status