

# <미스터대박디너서비스>

## 설계 산출물

문서버전	2.1
문서 ID	SE-2020-002
최종변경일	2020-12-02
문서상태	릴리즈

### 요 약

미스터 대박 디너 서비스의 설계 산출물을 기술.  
서브 시스템의 구성과 각 서브 시스템의 구조를 기술.  
주요 산출물

- ✓ 아키텍처도
- ✓ 클래스도
- ✓ 순차도
- ✓ 상태도

표 1 문서 변경 기록

문서이름	<미스터대박디너서비스> 설계 산출물		
문서 ID	SE-2020-002		
버전	세부 버전	변경일	설명
1	0	2020-11-04	시퀀스 다이어그램 초안 작성 상태 다이어그램 작성
	1	2020-11-05	시퀀스 다이어그램 최종 작성 아키텍처 다이어그램 작성 클래스 다이어그램 작성
2	0	2020-11-10	아키텍처도 수정
	1	2020-12-02	아키텍처도 수정 Order 객체 상태도 수정

# 1 개 요

## 1.1 목 적

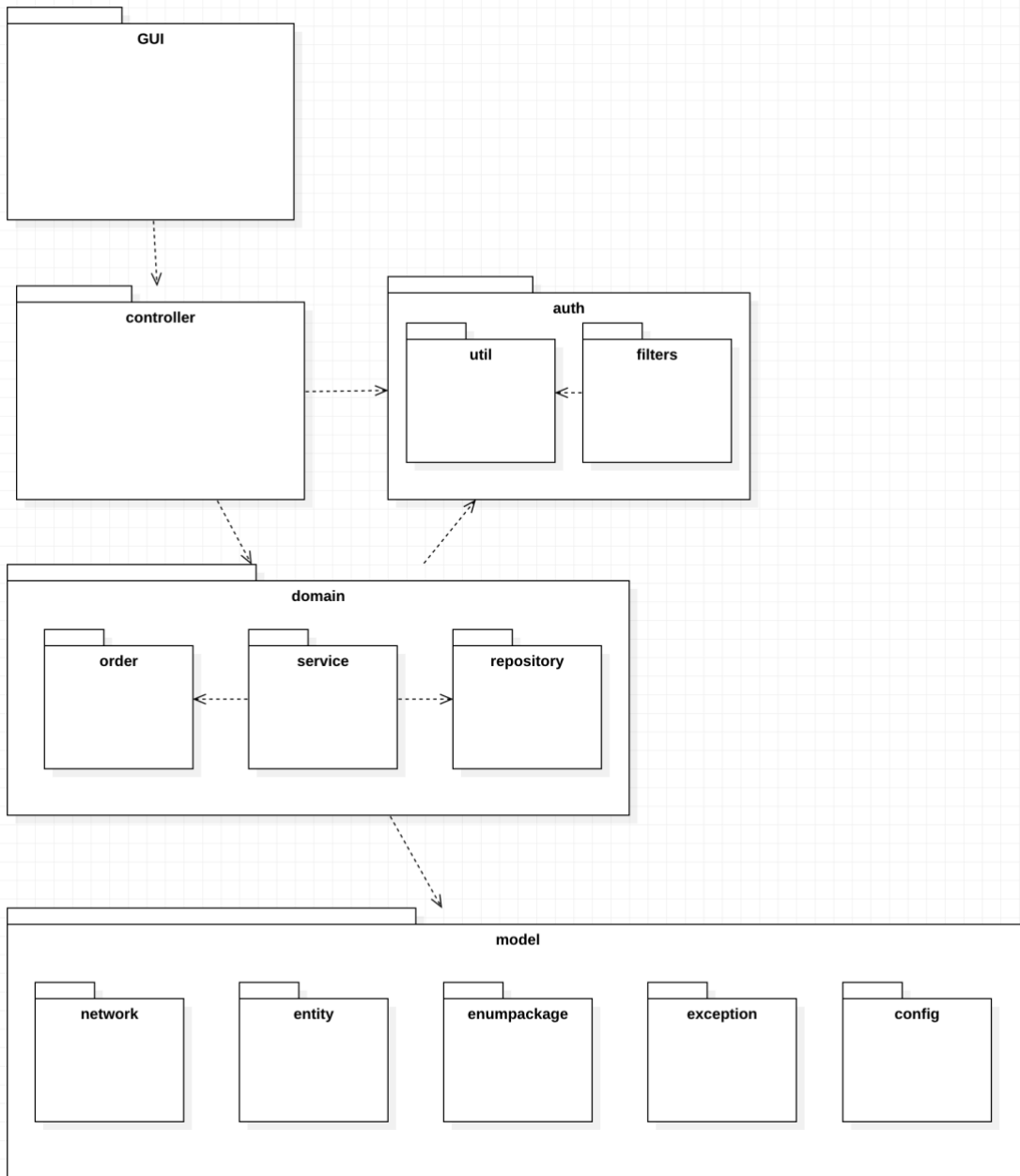
본 문서는 미스터 대박 디너 서비스의 설계 산출물을 기술한다.

- ✓ 전체 시스템이 어떤 서브 시스템으로 구성되어 있는지 “아키텍처도”를 통해 보여준다.
- ✓ 서브 시스템이 어떤 클래스로 구성되어 있고 클래스들간에 어떤 관계가 있는지를 “클래스도”를 통해서 보여준다.
- ✓ 객체간에 발생하는 동적인 행위를 “순차도”를 통해서 보여준다.
- ✓ 한 객체의 상태 변화를 “상태도”를 통해서 보여준다.

## 1.2 참고 문헌

없음.

## 2 아키텍처도



전체 시스템은 아래의 서브 시스템으로 구성된다.

- GUI 시스템
- controller 시스템
- auth 시스템
- domain 시스템
- model 시스템

### 1. GUI 시스템

사용자 인터페이스로, 사용자의 입력을 받고 사용자에게 domain 시스템의 출력을 반환하는 시스템이다.

### 2. controller 시스템

GUI 시스템으로부터 넘어온 요청을 auth 시스템에서 권한을 확인해서 domain 시스템으로 넘겨주는 시스템이다.

3. auth 시스템

접근 권한을 부여하고 접근한 유저 정보를 제공한다.

4. domain 시스템

시스템의 데이터 로직을 담당하며 controller 시스템으로부터 받은 요청을 처리하는 시스템이다.

5. model 시스템

전체 시스템에 필요한 엔티티들을 제공하는 시스템이다.

아키텍처도에서 서브 시스템간의 점선은 다음의 경우를 의미한다.

- 시작 서브 시스템의 클래스가 대상 서브 시스템의 클래스로의 의존관계를 가지는 경우 (상속이나 구현의 대상, 변수의 타입)
- 시작 서브 시스템으로부터 대상 서브 시스템으로 메시지가 전달되는 경우

## 2.1 GUI 서브시스템 세부 설계

## 2.2 Domain 서브시스템 세부 설계

domain 시스템은 controller 시스템에서 요청하는 데이터를 Model 시스템이 제공하는 형식으로 가져와서 가공한 후 다시 controller 시스템에 전송하는 역할을 한다.

domain 시스템은 다음과 같은 주요 API 들로 구성되어 있다.

- create : 사용자가 제공하는 데이터를 검사하고 데이터베이스에 저장한다.
- read : 사용자가 요청하는 데이터를 형식에 맞게 가공하여 제공한다.
- readAll : 사용자가 요청하는 데이터를 형식에 맞게 리스트로 제공한다.
- update : 데이터베이스에 존재하는 데이터를 사용자의 요청에 따라 수정한다.
- delete : 사용자가 요청한 데이터를 데이터베이스에서 삭제한다.

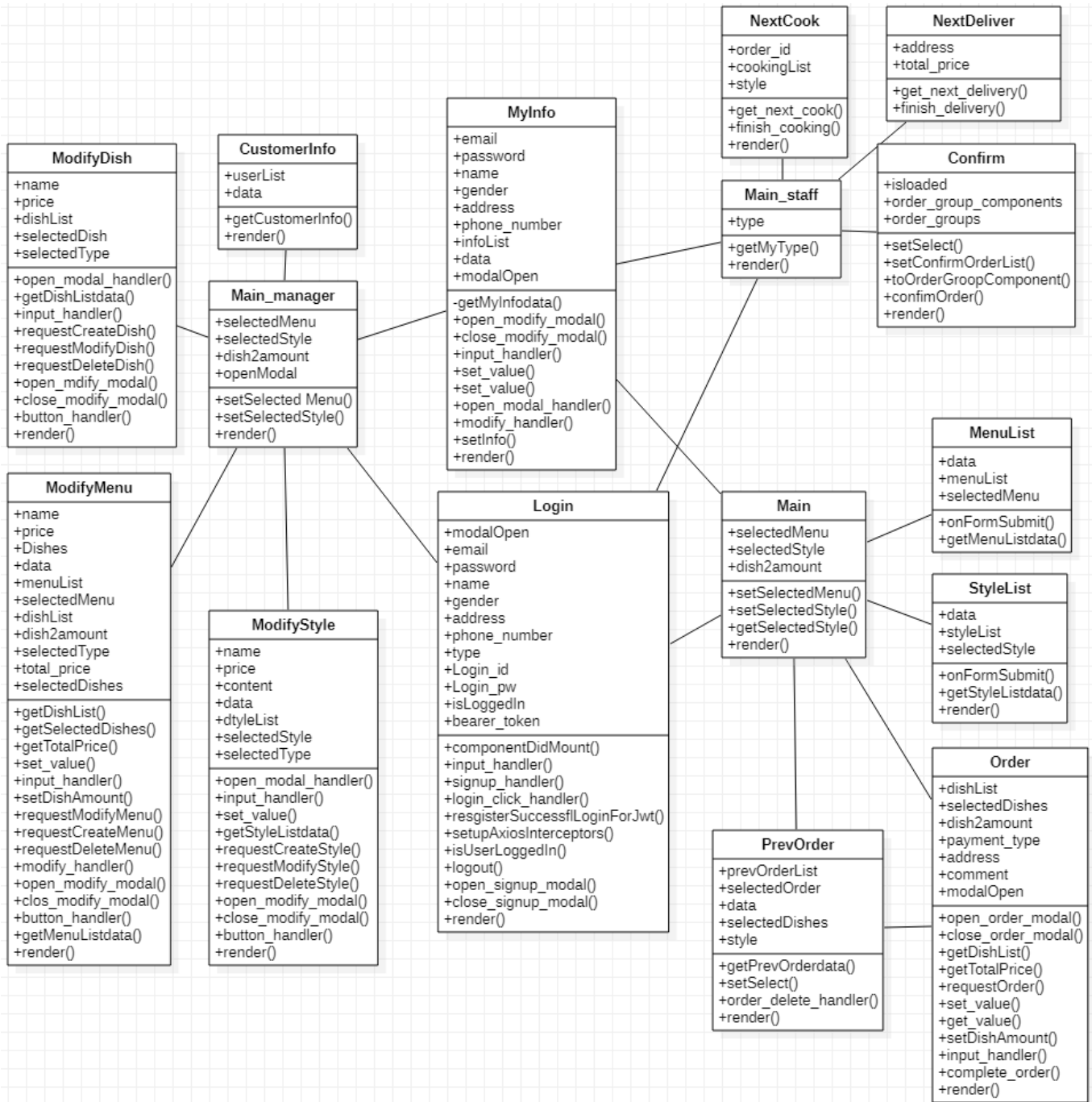
## 2.3 Auth 서브시스템 세부 설계

auth 시스템은 제공받은 요청을 분석해서 사용자를 구분하고 사용자의 정보를 다시 controller 시스템에게 돌려준다. 또한, domain 시스템의 호출에 따라 사용자에게 권한 정보가 담긴 토큰을 제공한다.

- createToken: 사용자를 구분할 수 있는 토큰을 제공한다.
- getClaims: 사용자를 구분하여 사용자의 정보를 제공한다.

## 3.1 메인 클래스도

### 3.1.1 GUI 시스템



#### Main

- 로그인 한 후, user 의 type 이 customer 일 때 표시되는 첫 화면이다.
- 메인 화면에서 바로 메뉴와 스타일 리스트를 나타낸다.
- customer 는 모든 페이지에서 상단 로고 링크를 통해 메인 화면 접속 가능하다.

#### Login

- 직원/고객/매니저 회원에 따라 메인 페이지의 구성 및 접근 가능 페이지 차이가 있다.

- 직원: Main\_staff
- 매니저: Main\_manager
- 고객: Main

## Order

- customer 가 메뉴, 스타일을 선택 후, 세부 수량을 선택하고 주문을 할 수 있는 화면을 띄운다.
- PrevOrder 또는 Main 에서 요리 리스트와 스타일을 받아와서 가격을 계산하고, 내부적으로는 세부 수량선택, 사용자 주문정보 저장 등을 수행한다.

## MyInfo

- 모든 user 에 대하여 사용자 개인정보를 보여주고, 수정할 수 있는 화면을 띄운다.
- 사용자가 정보 수정을 입력하면 해당 정보 수정을 요청한다.

## ModifyMenu, Style, Dish

- Main\_manager 를 통해서만 접근할 수 있으며, manager 만 접속 가능하다.
- main 페이지에 띄울 Menu List, Style List, Dish List 를 세부 수정, 삭제, 생성 가능하다.

## 3.1.2 Controller 시스템

OrderGroupApiController
-orderGroupApiService
+create() +readAll() +nextConfirm() +confirm() +nextCook() +finishCook() +nextDelivery() +finishDelivery() +cancel()

StaffApiController
-userApiService
+create() +read() +readAll() +update() +delete()

IngredientApiController
-ingredientApiService
+create() +read() +readAll() +update() +delete()

MenuApiController
-menuApiService
+create() +read() +readAll() +update() +delete()

DishApiController
-dishApiService
+create() +read() +readAll() +update() +delete()

StyleApiController
-styleApiService
+create() +read() +readAll() +update() +delete()

UserApiController
-userApiService
+signin() +create() +read() +readAll() +update()

ExceptionHandler
+permissionDeniedException() +invalidInputException() +entityNotFoundException() +existedUserException() +loginFailedException()



## OrderGroupApiController

- 주문과 관련된 요청을 받는 클래스

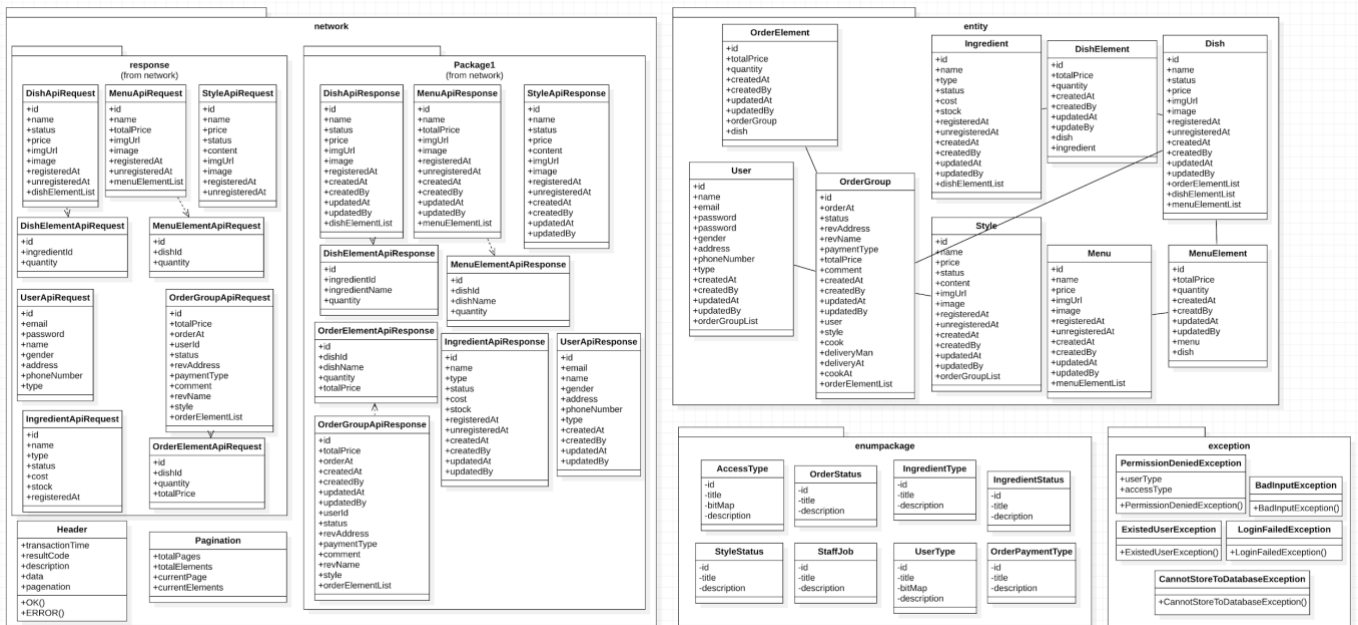
## UserApiController

- 유저와 관련된 요청을 받는 클래스
- 로그인, 회원가입, 읽기, 수정 등의 요청을 라우팅한다.

## ExceptionHandler

- 요청을 수행하는 도중 예외가 발생했을때 응답을 담당한다.
- 각종 예외에 대해서 어떻게 응답할 것인지 정의되어있다.

## 3.1.3 model 시스템



### network

- request 와 response 형식에 관련된 클래스들을 정의해놓은 패키지
- 페이징 처리 관련 클래스 제공

### entity

- 데이터베이스 ORM 매핑용 클래스 제공

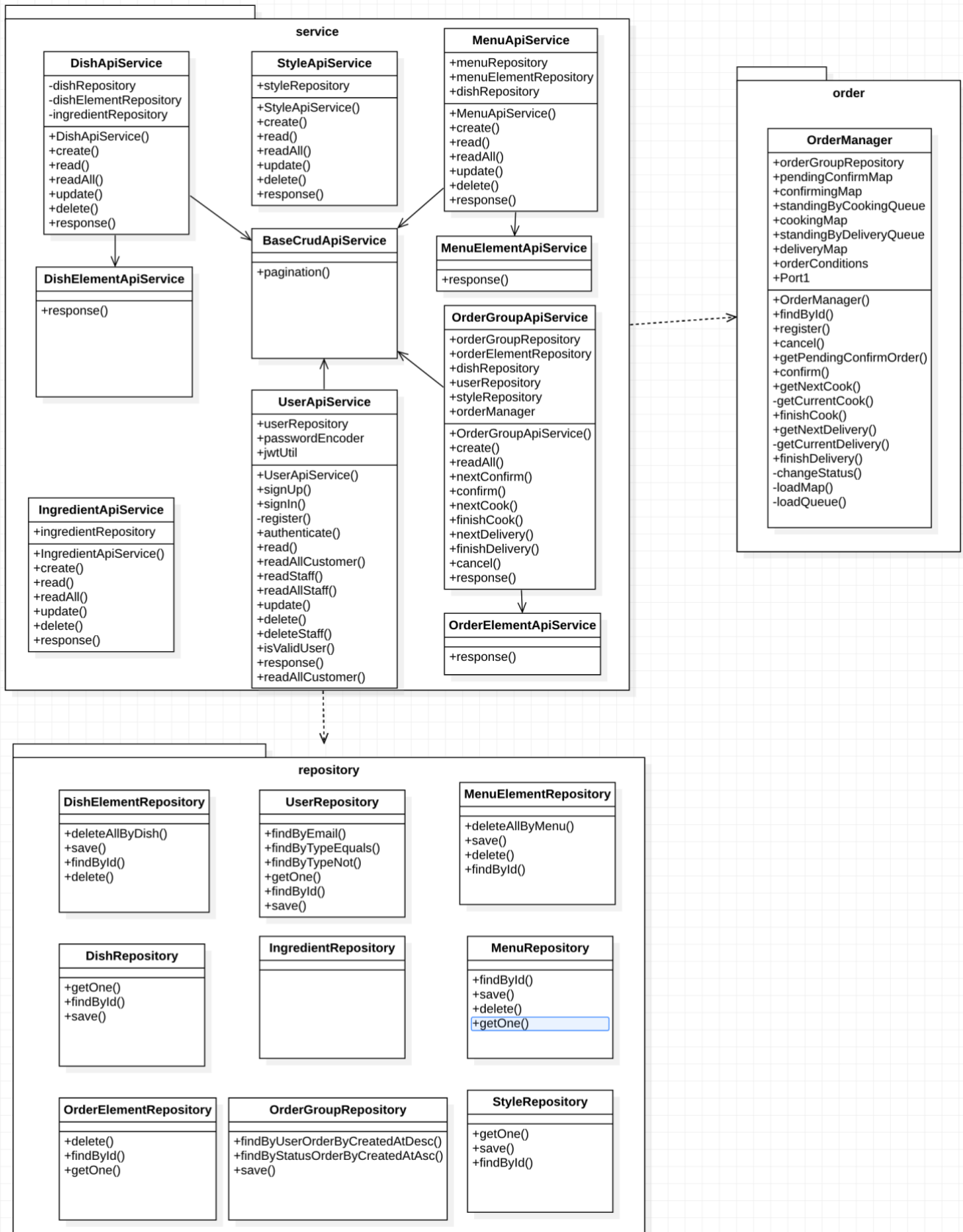
### enumpackage

- 시스템에서 사용하는 열거형 객체 제공

### exception

- 시스템에서 발생하는 사용자 정의 예외 제공

#### 3.1.4 domain 시스템



service

- 비즈니스 로직을 처리

order

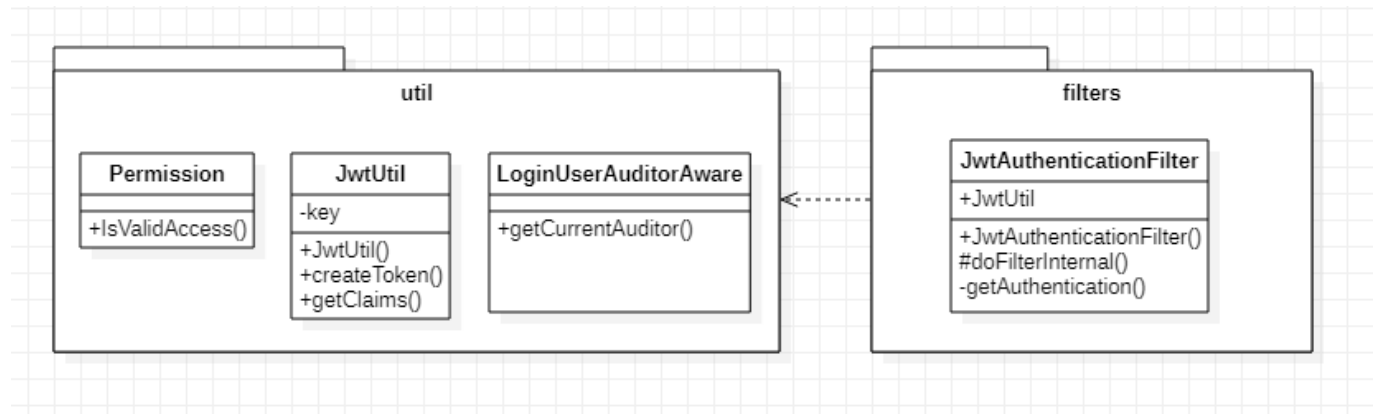
- OrderManger
  - 주문 관리 기능을 제공

- 현재 완료되지 않은 주문들을 캐싱하는 로컬 저장소 역할을 수행

repository

- 데이터베이스 인터페이스를 담당
- java interface 들로 이루어진 패키지
- JPA Hibernate 에 의해서 구현되기 때문에 실제 구현 코드는 존재하지 않음

### 3.1.5 auth 시스템



#### JwtUtil

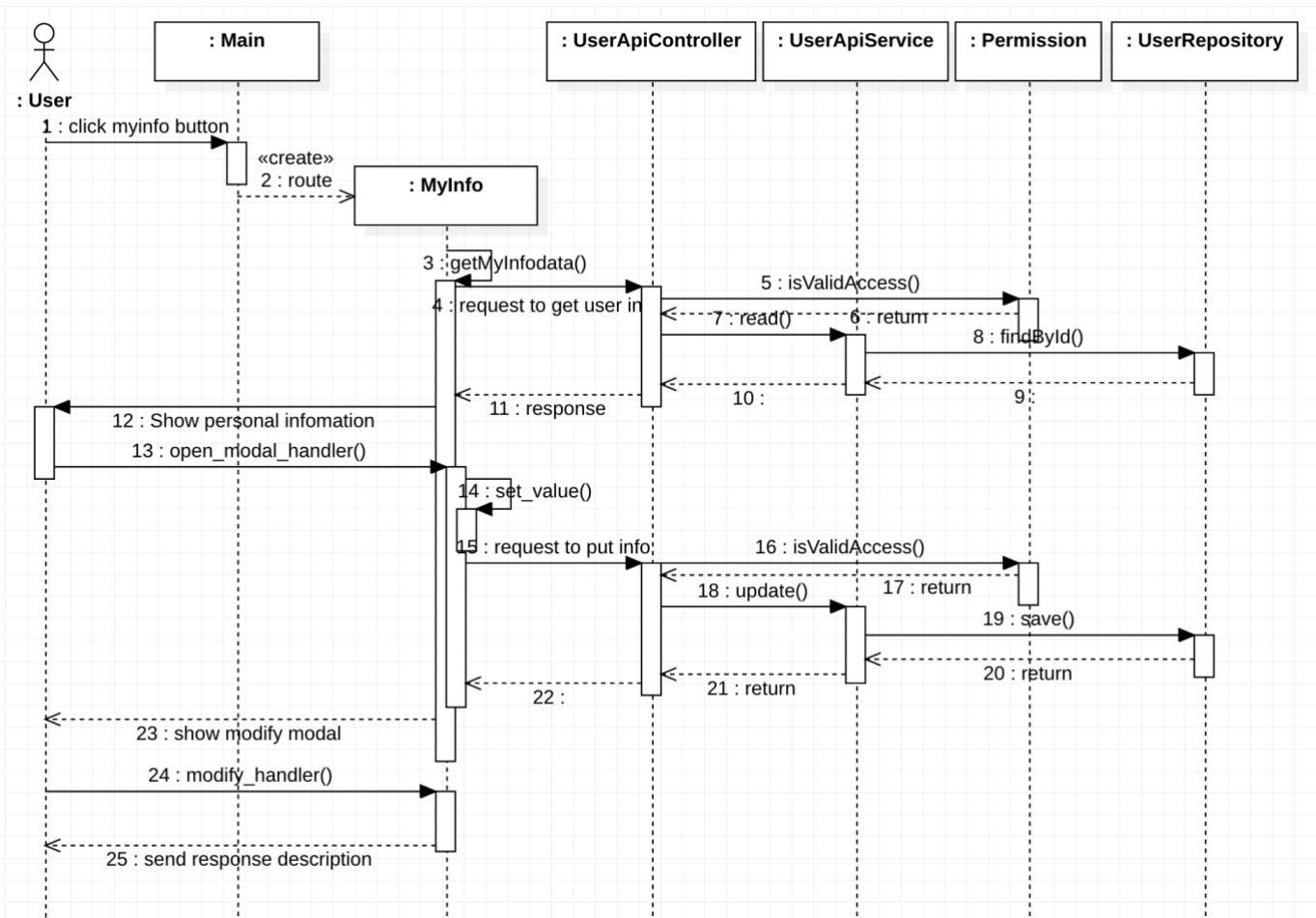
- JWT 토큰과 관련된 클래스
- JWT 토큰을 생성하거나 받은 JWT 토큰을 파싱한다.

#### JwtAuthenticationFilter

- spring boot 로 들어온 요청을 필터링하는 클래스
- 사용자 정의 spring boot filter 클래스이다.

## 4. 교류도

### 4.1 Modify personal info



1. 사용자가 myinfo 를 클릭하면 main->myinfo 페이지로 넘어가서 get myInfo 를 수행
2. isValidAccess 를 통하여 유효한 접근인지 확인하고, 해당 정보를 화면에 띄운다.
3. 사용자가 수정한 개인 정보를 입력하면 권한을 확인하고 데이터베이스에 있는 데이터를 수정한다.
4. 수정된 결과를 사용자에게 반환한다.

## 4.2 Order

- 
- ```

sequenceDiagram
    participant Customer as : customer
    participant Main as : Main
    participant PrevOrder as : PrevOrder
    participant Order as : Order
    participant Permission as : Permission
    participant OrderGroupApiController as : OrderGroupApiController
    participant OrderGroupApiService as : OrderGroupApiService
    participant UserRepository as : UserRepository
    participant OrderGroupRepository as : OrderGroupRepository
    participant BaseCrudApiService as : BaseCrudApiService
    participant StyleRepository as : StyleRepository
    participant OrderManager as : OrderManager
    participant NextCook as : NextCook
    participant NextDelivery as : NextDelivery
    participant Confirm as : Confirm
    participant Staff as : staff

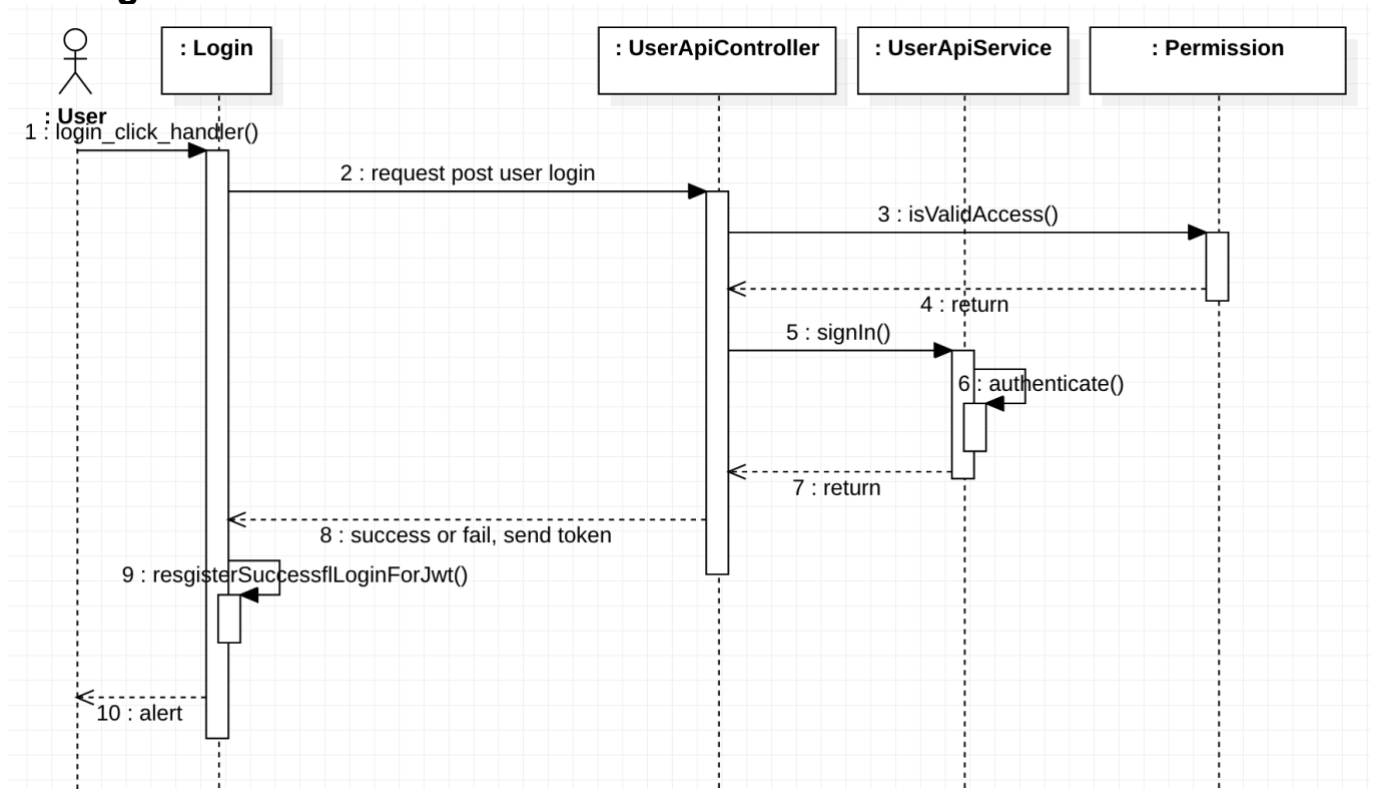
    Customer->>Main: seq Order
    Customer->>Main: click order button
    Main->>PrevOrder: 2 : route
    PrevOrder->>Order: 3 : route
    Order->>OrderGroupApiController: 4 : getPrevOrderData()
    OrderGroupApiController->>Permission: 6 : isValidateAccess()
    Permission-->>OrderGroupApiController: 7 : return
    OrderGroupApiController->>OrderGroupApiService: 8 : readAll()
    OrderGroupApiService->>UserRepository: 9 : getOne()
    UserRepository-->>OrderGroupApiService: 10 : return
    OrderGroupApiService->>OrderGroupRepository: 11 : findByUserOrderlyCreatedAtDesc()
    OrderGroupRepository-->>OrderGroupApiService: 12 : return OrderGroup pagination()
    OrderGroupApiService-->>OrderGroupApiController: 14 : return pagination
    OrderGroupApiController-->>Order: 15 : return response
    OrderGroupApiController-->>Order: 16 : response prev order
    Order->>Main: 17 : show prev order list
    Main->>Main: 18 : setSelect()
    Main->>Main: 19 : click button
    Main->>Main: 20 : setDishAmount()
    Main->>Main: 21 : DishList()
    Main->>Main: 22 : show Dish list
    Main->>Main: 23 : setDishAmount()
    Main->>Main: 24 : show modified dish quantity and price
    Main->>Order: 25 : requestOrder()
    Order->>OrderGroupApiController: 26 : request post order
    OrderGroupApiController->>Permission: 27 : isValidateAccess()
    Permission-->>OrderGroupApiController: 28 : return
    OrderGroupApiController->>OrderGroupApiService: 29 : create()
    OrderGroupApiService->>UserRepository: 30 : getOne()
    UserRepository-->>OrderGroupApiService: 31 : return user
    OrderGroupApiService->>StyleRepository: 32 : getOne()
    StyleRepository-->>OrderGroupApiService: 33 : return style
    OrderGroupApiService->>OrderGroupRepository: 34 : save()
    OrderGroupRepository-->>OrderGroupApiService: 35 : return saved orderGroup
    OrderGroupApiService-->>OrderGroupApiController: 36 : register()
    OrderGroupApiController->>OrderManager: 37 : changeStatus()
    OrderManager-->>OrderGroupApiController: 38 : return
    OrderGroupApiController->>Order: 39 : return
    OrderGroupApiController-->>Order: 40 : return
    OrderGroupApiController-->>Order: 41 : 43 : isValidateAccess()
    Order-->>OrderGroupApiController: 42 : get confirm order list
    OrderGroupApiController->>OrderGroupApiService: 44 : nextConfirm()
    OrderGroupApiService->>OrderGroupRepository: 45 : getPendingConfirmOrder()
    OrderGroupRepository-->>OrderGroupApiService: 46 : return OrderGroup list
    OrderGroupApiService-->>OrderGroupApiController: 47 : changeStatus()
    OrderGroupApiController->>OrderManager: 48 : response confirm order list
    OrderManager-->>OrderGroupApiController: 49 : changeStatus()
    OrderGroupApiController->>Order: 50 : response confirm order list
    OrderGroupApiController-->>Order: 51 : 52 : setConfirmOrderList()
    Order-->>OrderGroupApiController: 53 : click confirm order
    OrderGroupApiController->>OrderGroupApiService: 54 : post confirm order
    OrderGroupApiService->>NextCook: 55 : confirm()
    NextCook-->>OrderGroupApiService: 56 : return
    OrderGroupApiService-->>OrderGroupApiController: 57 : nextCook()
    OrderGroupApiController-->>Order: 58 : return response
    OrderGroupApiController-->>Order: 59 : 60 : isValidateAccess()
    Order-->>OrderGroupApiController: 61 : nextCook()
    OrderGroupApiController->>OrderGroupApiService: 62 : get next cook
    OrderGroupApiService->>NextCook: 63 : getNextCook()
    NextCook-->>OrderGroupApiService: 64 : return currentCook
    OrderGroupApiService-->>OrderGroupApiController: 65 : changeStatus()
    OrderGroupApiController->>Order: 66 : return next cook
    OrderGroupApiController-->>Order: 67 : 68 : return next cook
    OrderGroupApiController-->>Order: 69 : 70 : isValidateAccess()
    Order-->>OrderGroupApiController: 71 : nextCook()
    OrderGroupApiController->>OrderGroupApiService: 72 : post finish cook
    OrderGroupApiService->>NextCook: 73 : finishCook()
    NextCook-->>OrderGroupApiService: 74 : return cook
    OrderGroupApiService-->>OrderGroupApiController: 75 : findById()
    OrderGroupApiController-->>Order: 76 : return
    OrderGroupApiController-->>Order: 77 : 78 : isValidateAccess()
    Order-->>OrderGroupApiController: 79 : finishCook()
    OrderGroupApiController->>OrderGroupApiService: 80 : get next delivery
    OrderGroupApiService->>NextDelivery: 81 : Message37
    NextDelivery-->>OrderGroupApiService: 82 : finishDelivery()
    OrderGroupApiService-->>OrderGroupApiController: 83 : changeStatus()
    OrderGroupApiController->>Order: 84 : response next delivery
    OrderGroupApiController-->>Order: 85 : 86 : return next delivery
    OrderGroupApiController-->>Order: 87 : 88 : isValidateAccess()
    Order-->>OrderGroupApiController: 89 : nextDelivery()
    OrderGroupApiController->>OrderGroupApiService: 90 : post finish delivery
    OrderGroupApiService->>NextDelivery: 91 : finishDelivery()
    NextDelivery-->>OrderGroupApiService: 92 : return deliveryMan
    OrderGroupApiService-->>OrderGroupApiController: 93 : changeStatus()
    OrderGroupApiController->>Order: 94 : return deliveryMan
    OrderGroupApiController-->>Order: 95 : 96 : return deliveryMan
    OrderGroupApiController-->>Order: 97 : 98 : isValidateAccess()
    Order-->>OrderGroupApiController: 99 : finishDelivery()
    OrderGroupApiController->>OrderGroupApiService: 100 : get next delivery
    OrderGroupApiService->>NextDelivery: 101 : Message37
    NextDelivery-->>OrderGroupApiService: 102 : finishDelivery()
    OrderGroupApiService-->>OrderGroupApiController: 103 : changeStatus()
    OrderGroupApiController->>Order: 104 : response next delivery
    OrderGroupApiController-->>Order: 105 : 106 : return next delivery
    OrderGroupApiController-->>Order: 107 : 108 : isValidateAccess()
    Order-->>OrderGroupApiController: 109 : nextDelivery()
    OrderGroupApiController->>OrderGroupApiService: 110 : post finish delivery
    OrderGroupApiService->>NextDelivery: 111 : finishDelivery()
    NextDelivery-->>OrderGroupApiService: 112 : return deliveryMan
    OrderGroupApiService-->>OrderGroupApiController: 113 : changeStatus()
    OrderGroupApiController->>Order: 114 : return deliveryMan
    OrderGroupApiController-->>Order: 115 : 116 : return deliveryMan
    OrderGroupApiController-->>Order: 117 : 118 : isValidateAccess()
    Order-->>OrderGroupApiController: 119 : finishDelivery()
    OrderGroupApiController->>OrderGroupApiService: 120 : get next delivery
    OrderGroupApiService->>NextDelivery: 121 : Message37
    NextDelivery-->>OrderGroupApiService: 122 : finishDelivery()
    OrderGroupApiService-->>OrderGroupApiController: 123 : changeStatus()
    OrderGroupApiController->>Order: 124 : response next delivery
    OrderGroupApiController-->>Order: 125 : 126 : return next delivery
    OrderGroupApiController-->>Order: 127 : 128 : isValidateAccess()
    Order-->>OrderGroupApiController: 129 : nextDelivery()
    OrderGroupApiController->>OrderGroupApiService: 130 : post finish delivery
    OrderGroupApiService->>NextDelivery: 131 : finishDelivery()
    NextDelivery-->>OrderGroupApiService: 132 : return deliveryMan
    OrderGroupApiService-->>OrderGroupApiController: 133 : changeStatus()
    OrderGroupApiController->>Order: 134 : return deliveryMan
    OrderGroupApiController-->>Order: 135 : 136 : return deliveryMan
    OrderGroupApiController-->>Order: 137 : 138 : isValidateAccess()
    Order-->>OrderGroupApiController: 139 : finishDelivery()
    OrderGroupApiController->>OrderGroupApiService: 140 : get next delivery
    OrderGroupApiService->>NextDelivery: 141 : Message37
    NextDelivery-->>OrderGroupApiService: 142 : finishDelivery()
    OrderGroupApiService-->>OrderGroupApiController: 143 : changeStatus()
    OrderGroupApiController->>Order: 144 : response next delivery
    OrderGroupApiController-->>Order: 145 : 146 : return next delivery
    OrderGroupApiController-->>Order: 147 : 148 : isValidateAccess()
    Order-->>OrderGroupApiController: 149 : nextDelivery()
    OrderGroupApiController->>OrderGroupApiService: 150 : post finish delivery
    OrderGroupApiService->>NextDelivery: 151 : finishDelivery()
    NextDelivery-->>OrderGroupApiService: 152 : return deliveryMan
    OrderGroupApiService-->>OrderGroupApiController: 153 : changeStatus()
    OrderGroupApiController->>Order: 154 : return deliveryMan
    OrderGroupApiController-->>Order: 155 : 156 : return deliveryMan
    OrderGroupApiController-->>Order: 157 : 158 : isValidateAccess()
    Order-->>OrderGroupApiController: 159 : finishDelivery()
    OrderGroupApiController->>OrderGroupApiService: 160 : get next delivery
    OrderGroupApiService->>NextDelivery: 161 : Message37
    NextDelivery-->>OrderGroupApiService: 162 : finishDelivery()
    OrderGroupApiService-->>OrderGroupApiController: 163 : changeStatus()
    OrderGroupApiController->>Order: 164 : response next delivery
    OrderGroupApiController-->>Order: 165 : 166 : return next delivery
    OrderGroupApiController-->>Order: 167 : 168 : isValidateAccess()
    Order-->>OrderGroupApiController: 169 : nextDelivery()
    OrderGroupApiController->>OrderGroupApiService: 170 : post finish delivery
    OrderGroupApiService->>NextDelivery: 171 : finishDelivery()
    NextDelivery-->>OrderGroupApiService: 172 : return deliveryMan
    OrderGroupApiService-->>OrderGroupApiController: 173 : changeStatus()
    OrderGroupApiController->>Order: 174 : return deliveryMan
    OrderGroupApiController-->>Order: 175 : 176 : return deliveryMan
    OrderGroupApiController-->>Order: 177 : 178 : isValidateAccess()
    Order-->>OrderGroupApiController: 179 : finishDelivery()
    OrderGroupApiController->>OrderGroupApiService: 180 : get next delivery
    OrderGroupApiService->>NextDelivery: 181 : Message37
    NextDelivery-->>OrderGroupApiService: 182 : finishDelivery()
    OrderGroupApiService-->>OrderGroupApiController: 183 : changeStatus()
    OrderGroupApiController->>Order: 184 : response next delivery
    OrderGroupApiController-->>Order: 185 : 186 : return next delivery
    OrderGroupApiController-->>Order: 187 : 188 : isValidateAccess()
    Order-->>OrderGroupApiController: 189 : nextDelivery()
    OrderGroupApiController->>OrderGroupApiService: 190 : post finish delivery
    OrderGroupApiService->>NextDelivery: 191 : finishDelivery()
    NextDelivery-->>OrderGroupApiService: 192 : return deliveryMan
    OrderGroupApiService-->>OrderGroupApiController: 193 : changeStatus()
    OrderGroupApiController->>Order: 194 : return deliveryMan
    OrderGroupApiController-->>Order: 195 : 196 : return deliveryMan
    OrderGroupApiController-->>Order: 197 : 198 : isValidateAccess()
    Order-->>OrderGroupApiController: 199 : finishDelivery()
    OrderGroupApiController->>OrderGroupApiService: 200 : get next delivery
    OrderGroupApiService->>NextDelivery: 201 : Message37
    NextDelivery-->>OrderGroupApiService: 202 : finishDelivery()
    OrderGroupApiService-->>OrderGroupApiController: 203 : changeStatus()
    OrderGroupApiController->>Order: 204 : response next delivery
    OrderGroupApiController-->>Order: 205 : 206 : return next delivery
    OrderGroupApiController-->>Order: 207 : 208 : isValidateAccess()
    Order-->>OrderGroupApiController: 209 : nextDelivery()
    OrderGroupApiController->>OrderGroupApiService: 210 : post finish delivery
    OrderGroupApiService->>NextDelivery: 211 : finishDelivery()
    NextDelivery-->>Order
```

5. 매니저가 해당 주문을 승인하면 주문의 상태는 STANDINGBYCOOKING 상태로 변하고 standingByCookinQueue 에 저장된다.
6. 요리사가 다음 요리할 주문 불러오기를 눌렀을때 이 주문 차례가 되었다면 주문은 COOKING 상태로 변하고 cookingMap 에 저장된다.
7. 요리를 완료한 뒤 요리완료를 누르면 이 주문은 STANDINGBYDELIVERY 상태가 되고 standingByDeliveryQueue 에 저장된다.
8. 배달원이 다음 배달 불러오기를 누르면 이 주문은 deliveryMap 에 저장된다.
9. 그 후 배달원이 배달을 완료하면 해당 주문은 OrderManager 에서 삭제되고 데이터베이스에서만 유지된다.

#### <이전 주문을 불러올 경우>

1. 과거의 주문 내역을 불러와서 사용자에게 제공한다.
2. 사용자가 과거 주문 내역 중 하나를 선택하여 주문버튼을 누르면 Order 에서 dishes 와 style 을 넘겨받는다.
3. 나머지 과정은 <새로운 주문일 경우>의 3 번으로 이어진다

## 4.3 Login



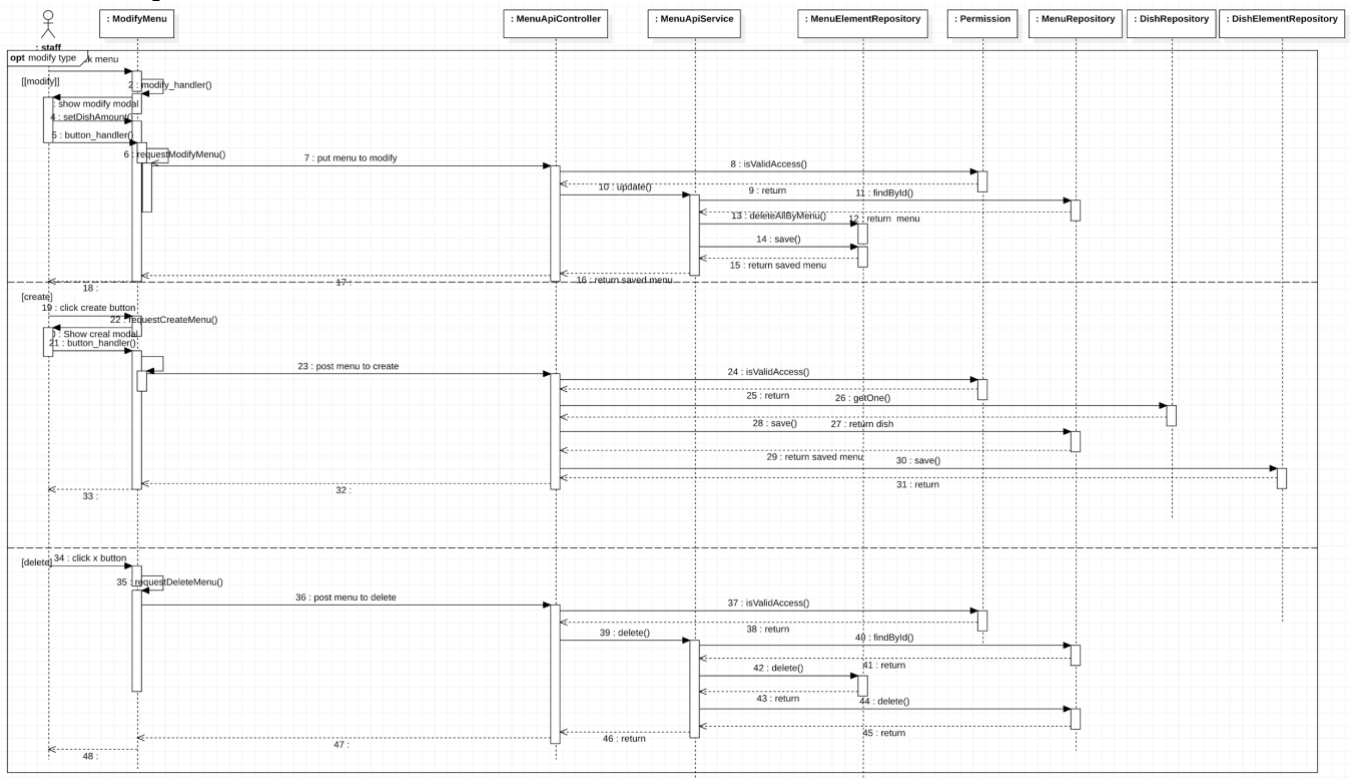
1. 사용자가 아이디와 비밀번호를 입력하면 UserApiController 에 post 요청한다
2. Permission 객체의 isValidAccess 를 이용하여 API 접근이 가능한지 확인한다.
3. 모든 유저에게 접근가능하므로 바로 UserApiService 객체의 authenticate()를 이용하여 아이디 비밀번호의



유효성을 검사한후 결과를 반환한다.

- 이후 Login 객체에 token 을 반환하거나 실패메시지를 반환하여 header 에 token 을 세팅하고 사용자에게 알림창을 띄운다.

## 4.4 Modify items



<수정>

- 수정할 메뉴를 선택하면 모달 창이 나타난다.
- 수정할 내용을 모두 입력한뒤 변경 버튼을 누르면 수정 내용이 MenuApiController 클래스로 전달된다.
- MenuApiController 클래스는 Permission 클래스를 통해 권한 확인을 받고 권한이 확인되면

MenuApiService 클래스를 호출한다.

- MenuApiController 클래스는 MenuRepository 클래스를 이용하여 데이터베이스에 수정 사항을 저장하고 수정된 내용을 다시 MenuApiController 클래스에 반환한다.
- MenuApiController 는 반환받은 수정내용을 다시 사용자에게 전달한다.

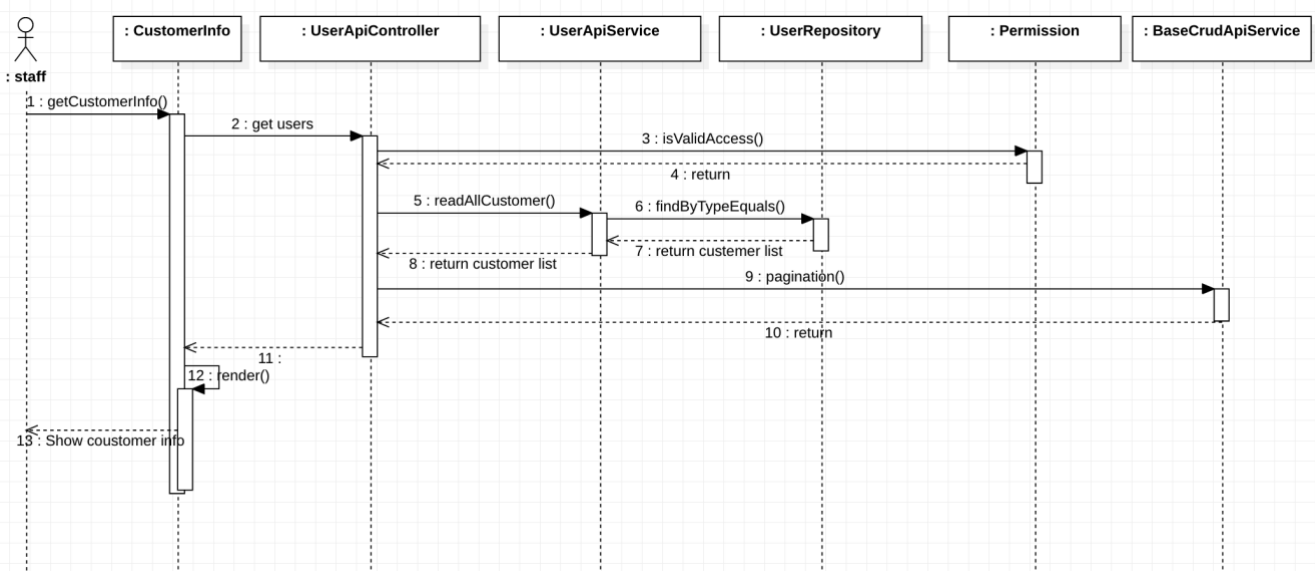
#### <추가>

- 추가 버튼을 선택하면 모달 창이 나타난다.
- 새로운 메뉴 내용을 모두 입력한 뒤 추가 버튼을 누르면 새로운 메뉴 내용이 MenuApiController 클래스로 전달된다.
- MenuApiController 클래스는 Permission 클래스를 통해 권한 확인을 받고 권한이 확인되면 MenuApiService 클래스를 호출한다.
- MenuApiController 클래스는 MenuRepository 클래스를 이용하여 데이터베이스에 새로운 메뉴를 저장하고 추가된 내용을 다시 MenuApiController 클래스에 반환한다.
- MenuApiController 는 반환받은 새로운 메뉴 내용을 다시 사용자에게 전달한다.

#### <삭제>

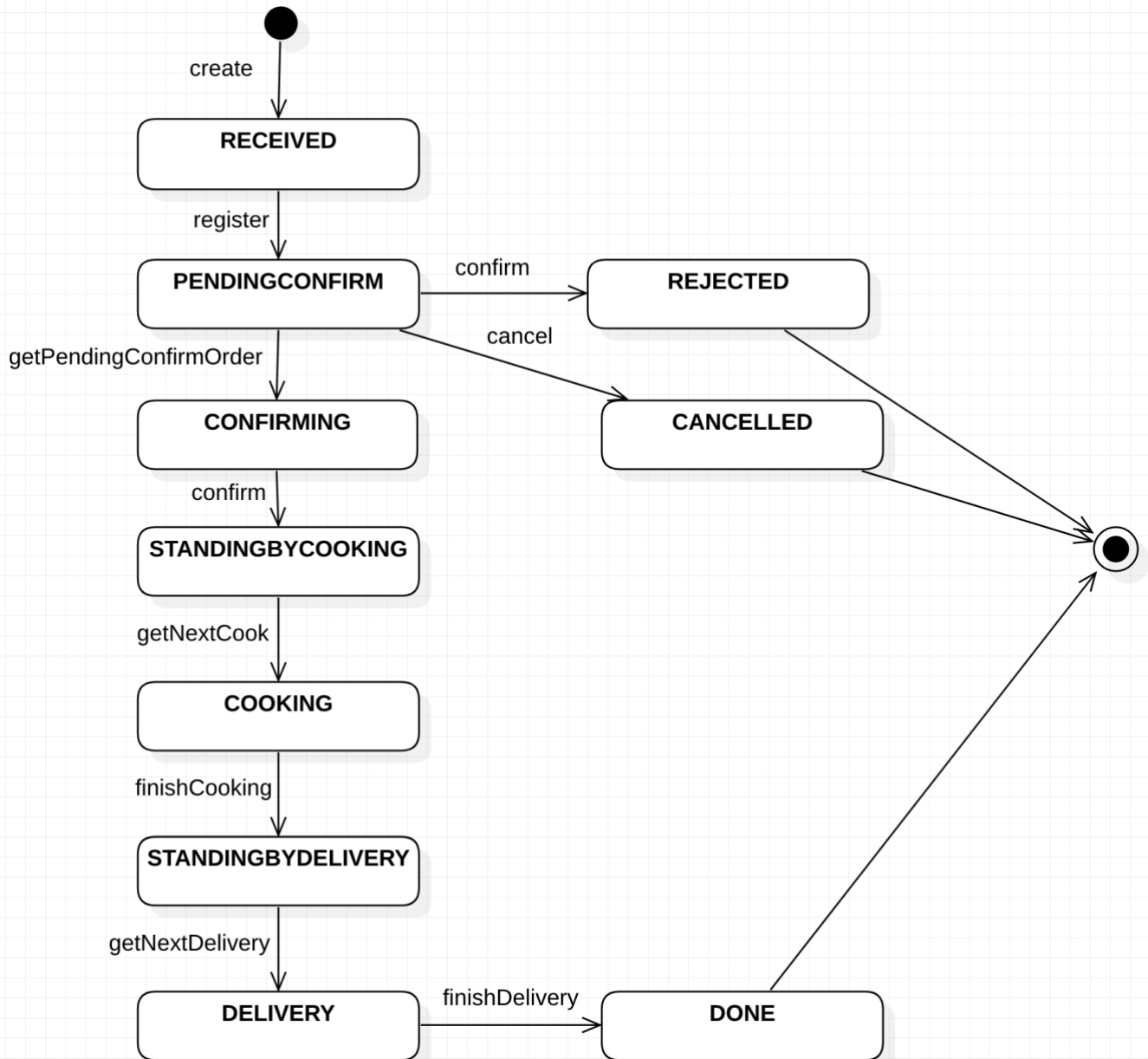
- 삭제할 메뉴 옆의 삭제 버튼을 누른다.
- RESTapi 를 통해 삭제할 메뉴의 id 를 MenuApiController 에게 넘겨준다.
- MenuApiController 는 MenuApiService 에게 삭제할 메뉴의 id 를 넘겨주고 MenuApiService 는 id 를 받아서 MenuRepository 를 이용하여 메뉴 내용을 가져온다.
- 메뉴 내용으로부터 MenuElement 들을 알아내고 MenuElementRepository 를 이용하여 삭제할 Menu 와 관련된 MenuElement 들을 모두 삭제한다.
- MenuElement 들을 모두 삭제한 뒤 Menu 를 데이터베이스에서 삭제한다.

## 4.5 View customer info



1. staff 가 CustomerInfo 를 통해 고객정보를 불러오면 get API 를 통해 UserApiController 에 요청을 보낸다.
2. 해당 요청이 유효한지 검사하여 그 결과를 반환하고, 유효하면 UserRepository 에서 customerlist 를 반환한다.
3. CustomerInfo 는 반환된 user 정보를 세팅 후 렌더링하여 staff 사용자에게 띄운다.

## 5. Order 객체 상태도



Order 객체는 주문을 나타내는 클래스이고 주문의 현재 상태에 따라서 Order 객체의 상태가 변한다.

- RECEIVED : 주문이 막 접수되었을 때의 상태이다. 데이터베이스에 등록되면 pendingConfirm 상태 바뀐다.
- PENDINGCONFIRM : 매니저의 승인을 기다리는 상태이다. 이때, 고객의 주문 취소 요청에 따라 CANCELLED 상태로 변할 수 있고 매니저가 주문 승인을 위해 PENDINGCONFIRM 상태의 주문들을 요청하면 PENDINGCONFIRM 상태에서 CONFIRMING 상태로 바뀐다.
- CONFIRMING : 매니저가 주문 승인을 위해 승인대기 전의 주문을 불러왔을때의 상태이다. 이때 매니저가 승인할 주문 목록을 넘겨주면 CONFIRMING 상태에서 STANDINGBYCOOKING 상태로 변하고 나머지 주문들은 REJECTED 상태가 된다.
- REJECTED : 매니저가 주문 승인을 거부하면 주문은 REJECTED 상태로 바뀐다.
- CANCELLED : PENDINGCONFIRM 상태에서 고객이 취소한 주문의 상태이다. 이 상태에서 주문은 종료되고 Order 객체는 주문 내역으로 이동된다.
- STANDINGBYCOOKING : 매니저가 주문 승인한 상태이다. 이 상태는 요리사가 해당 주문을 배정받기 전까지 유지된다.

- COOKING : 요리사가 해당 주문을 요리 중인 상태이다.
- STANDINGBYDELIVERY: 주문이 배달 준비를 끝마치고 배달을 대기하고 있는 상태이다.
- DELIVERY : 배달원이 해당 주문을 배달 중인 상태이다.
- DONE : 배달이 완료되어 주문 전 과정이 완료된 상태이다. 완료된 주문은 주문 내역으로 이동된다.