

# 대구요 배달 웹\_프로젝트

## 스토리보드 & 스타일가이드

임베디드 소프트웨어 융합 풀스택 개발자 양성 과정 B

2조 이재민, 백승하, 박민영, 정세인

# INDEX

## 스토리보드

- 01. 프로젝트 개요
- 02. 기술 스택
- 03. 프로젝트 설계
- 04. 핵심 기능 및 구현 내용
- 05. 스크린리스트
- 06. 프로세스
- 07. 플로차트
- 08. 정책
- 09. 권한

## 스타일가이드

- 10. 구동환경
- 11. 레이아웃
- 12. 디자인 프로토타입

# 01 스토리보드\_대구요 프로젝트 개요

## 프로젝트 명 : 대구요

### 개발 동기

팬데믹 이후 비대면 소비 확대로 인해 배달 서비스 시장 급성장

→ 배달 웹 서비스 구조 파악 필요성 인지

기존 배달 플랫폼의 대안 수요에 주목, 지역 밀착형 배달 서비스 ‘대구요’ 기획

### 개발 목적

음식주문 웹 사이트 제작 및 활용

대구시 지역 밀착형 배달 웹 서비스 구현



# 01 스토리보드\_대구요 프로젝트 개요 - 개발 히스토리

2023-09-11 ~ 2023-09-15(05Day) : UML 다이어그램 작성, 데이터 베이스 설계 등 프로젝트 설계 작업

2023-09-16 ~ 2023-09-17(02Day) : 개발환경 구축(GITHUB, GIT, INTELIJ, MYSQL)

2023-09-18 ~ 2023-09-24(07Day) : 사용 API 선별 및 기능 테스트

2023-09-25 ~ 2023-10-04(10Day) : Front-End 작업

2023-10-05 ~ 2023-10-09(05Day) : Front-End 코드 상호 피드백 및 수정

2023-10-10 ~ 2023-10-19(10Day) : Back-End 작업 진행

2023-10-20 ~ 2023-10-23(04Day) : 모든 작업 수정 및 완료

# 01 스토리보드\_대구요 프로젝트 개요 - 구성인원

**이재민** : FrontEnd, BackEnd (FrontEnd(매장선택, 메뉴상세), BackEnd(기본 레이아웃 및 틀 정리, RESTful api, 매장결제승인, 회원가입, 입장바구니, 매장선택)

**백승하**: AWS EC2 배포, FrontEnd, BackEnd (FrontEnd(매장결제 승인, 입점회원가입,로그인) , BackEnd(입점회원가입, 회원정보수정)

**정세인**: FrontEnd, BackEnd (FrontEnd(웹 기획, 메인페이지 구현) , BackEnd(로그인, 주문내역 BackEnd)

**박민영** : FrontEnd, BackEnd (FrontEnd(마이페이지, 주문내역) , BackEnd(메뉴상세, 마이페이지 BackEnd)

# 02 스토리보드 대구요 기술 스택

## 개발 환경

OS : WINDOW Server 2022 base

CPU : i7 Intel

RAM : 16GB SAMSUNG DDR4

DISK : 100GB SSD

## Front-End

Markup Language : HTML5

Styling : CSS3

Scripting : JavaScript, jQuery

Icons : Font Awesome

## Back-End

Programming Language: Java

## DEV TOOL

Framework : SpringBoot

IDE : IntelliJ IDEA

Database : MYSQL

Version Control : GitHub

## DEV Ops

### Cloud Service

: Amazon Web Service EC2(Deploy Server)

: Amazon Web Service RDS(Remote Database Server)

Version Control : Git & Github

Design Tool : Adobe XD

## Software 목록

Build Tool : Gradle

### Database

: Mysql Server 8.1.0

: Mysql Workbench 8.0.34

ORM : MyBatis 2.3.1

Messaging : NaverCloud SMS 4.5.13

### Data Processing

: JSON 1.1.1, JACKSON 2.14.2

: GSON 2.10.1

# 03 스토리보드\_대구요 프로젝트 설계 - 회원 요구사항 분석

## 인증 서비스

- 회원가입 요청시 사용자 입력 정보를 받아 저장한다 C
- 로그인 이후 회원 정보 조회가 가능하다 R
- 로그인 이후 회원 수정이 가능하다 U
- 회원 탈퇴 요청시 본인 확인 후 회원 정보 삭제가 가능하다 D

## 상품 서비스

- 회원은 상품을 조회할 수 있다 R

## 장바구니 서비스

- 회원은 장바구니에 상품을 담을 수 있다 C
- 회원은 장바구니 내역을 조회할 수 있다 R
- 회원은 장바구니 내역을 수정할 수 있다 U
- 회원은 장바구니 내역을 삭제할 수 있다 D

## 주문 서비스

- 회원은 장바구니에 담은 상품을 주문할 수 있다 C
- 회원은 주문한 상품을 조회할 수 있다 R
- 회원은 주문 내역을 문의 후 수정할 수 있다 U
- 회원은 주문 내역을 문의 후 삭제할 수 있다 D

## 결제 서비스

- 회원은 결제 요청을 통해 결제를 할 수 있다 C
- 회원은 결제 내역을 조회할 수 있다 R
- 회원은 결제 취소를 할 수 있다 D

## 리뷰 서비스

- 회원은 리뷰를 작성할 수 있다 C
- 회원은 리뷰를 조회할 수 있다 R
- 회원은 리뷰를 수정할 수 있다 U
- 회원은 리뷰를 삭제할 수 있다 D

# 03 스토리보드\_대구요 프로젝트 설계 - 비회원 요구사항 분석

## 인증 서비스

회원가입 요청시 사용자 입력 정보를 받아 저장한다 C

# 03 스토리보드\_대구요 프로젝트 설계 - 매장 요구사항 분석

## 인증 서비스

- 매장 가입 요청시 사용자 입력 정보를 받아 저장한다 C
- 로그인 이후 매장 정보 조회가 가능하다 R
- 로그인 이후 매장 정보 수정이 가능하다 U
- 매장 탈퇴 요청시 본인 확인 후 매장 회원 정보 삭제가 가능하다 D

## 상품 서비스

- 매장은 상품을 등록할 수 있다 C
- 매장은 상품을 조회할 수 있다 R
- 매장은 상품을 수정할 수 있다 U
- 매장은 상품을 삭제할 수 있다 D

## 리뷰 서비스

- 매장은 리뷰 답변을 작성할 수 있다 C
- 매장은 리뷰를 조회할 수 있다 R
- 매장은 관리자 문의 후 리뷰를 수정할 수 있다 U
- 매장은 관리자 문의 후후 리뷰를 삭제할 수 있다 D

## 주문 서비스

- 매장은 주문 요청 내역을 조회할 수 있다 R
- 매장은 주문 요청 내역을 삭제할 수 있다 D

## 결제 서비스

- 매장은 결제 내역을 조회할 수 있다 R
- 매장은 결제를 취소할 수 있다 D

# 03 스토리보드\_대구요 프로젝트 설계 - 유스케이스 명세서

## 회원가입

**행위자 :** 비회원

**개요 및 설명 :** 비회원은 가입 절차 진행 후 회원으로 등록할 수 있다.

**사전 조건 :** 비회원은 대구요 웹 사이트에 접속한 상태다.

**• 정상 흐름**

- 사이트에 접속 한다
- 사이트 상단에 로그인을 클릭하여 로그인 페이지로 넘어간다
- 로그인 페이지 내에서 회원가입을 눌러 회원가입 창으로 넘어간다
- 회원가입 폼에 맞춰 값을 입력한다 (A1)
- 모든 값을 입력 하였으면 하단에 있는 회원가입 / 확인하기 버튼을 누른다
- 모든 값이 정상 적으로 입력이 되었다면 시스템은 회원가입을 축하한다는 메세지를 전달한다.

**• 대안 흐름**

(A1) 회원가입 중 잘못된 값을 넣었을 경우

- 비회원은 해당 위치에 값을 잘 못 넣었을 경우 지우고 새로 작성할 수 있다
- 비회원이 비밀번호를 잘못 입력하였을 경우우 뒤로 가기 또는 취소 버튼을 눌러 다시 작성 할 수 있다.

## 메뉴 수정

**행위자 :** 매장, 관리자

**개요 및 설명 :** 매장은 자신 매장에 있는 메뉴를 수정할 수 있다.

**사전 조건 :** 매장은 로그인 후 매장 페이지에 접속한 상황이다.

**• 정상 흐름**

- 매장은 메뉴 관리 옵션을 선택한다
- 시스템은 현재 등록된 메뉴 리스트를 보여준다
- 매장은 수정하려는 메뉴를 선택한다
- 시스템은 선택한 메뉴의 현재 정보를 보여준다
- 매장은 수정할 항목을 선택한다 (A1)  
5-1) 선택 후 수정 완료 버튼을 클릭한다(A2)
- 시스템은 변경 사항을 확인하고 저장 후 "메뉴가 수정되었습니다"라는 알림을 보여준다(E1)

**• 대안 흐름**

(A1) 잘못된 입력 값을 넣을 경우

- 시스템은 에러 메시지를 보여주며 올바른 값을 입력하도록 요청한다

(A2) 취소를 누를 경우

- 모든 변경 사항은 저장되지 않으며 이전 화면으로 돌아간다

**• 예외 흐름**

(E1) 서버에서 오류가 발생하여 데이터를 저장할 수 없는 경우

- 시스템은 "서버 오류가 발생했습니다. 다시 시도해주세요"라는 메시지를 보여준다

# 03 스토리보드\_대구요 프로젝트 설계 - 유스케이스 명세서

## 회원 수정

**행위자 :** 회원  
**개요 및 설명 :** 회원은 자신의 정보를 수정 할 수 있다.  
**사전 조건 :** 회원은 회원가입(아이디,비밀번호,주소,이메일, 휴대폰 번호)과 인증을 마친 상태다.

**• 정상 흐름**

1. 회원은 마이페이지로 접속한다.
2. 마이페이지의 회원정보 수정 버튼을 클릭 후 회원정보수정 페이지로 접속한다.
3. 회원은 자신의 정보를 양식에 맞게 수정한다.(A1,A2)
  - a. 아이디 수정일 경우
    - i. 중복된 다른 아이디가 없도록 입력한다(A3)
    - ii. 기존의 아이디와 다르게 입력한다.(A4)
    - iii. 양식에 맞게 입력한다(A5)
    - iv. 모두 맞게 입력하면 수정 표시 아이콘이 나타난다.
  - b. 비밀번호 수정일 경우
    - i. 양식에 허용되는 특수기호, 비밀번호 양식에 맞게 입력한다.(A5)
    - ii. 비밀번호와 비밀번호가 일치할 시 수정가능 표시 아이콘이 나타난다.(A6)
  - c. 주소 수정일 경우
    - i. 주소검색을 통해 기존의 주소정보를 수정한다.(A8)
  - d. 휴대폰 번호 수정일 경우
    - i. 휴대폰 인증하기 버튼을 누른다.
    - ii. 인증번호를 확인하라는 문구가 나타나며 번호 확인란이 활성화 된다 or 숨겨져있다 나타난다.
    - iii. 사용자가 정해진 시간내에 인증번호를 확인 후 정보를 입력한다.
    - iv. 인증완료라는 문구가 뜨며 정보가 수정된다.(E1)
  - e. 이메일 수정일 경우
    - i. 이메일 수정하기 버튼을 클릭한다
    - ii. 요청시간내에 이메일 확인 번호를 입력하라는 칸이 나타난다. 혹은 이메일 메세지를 확인하라는 문구가 나타난다.
    - iii. 본인확인 후 이메일을 수정한다.(E2)
4. 회원은 회원수정 버튼을 클릭한다
5. 시스템은 회원의 요청을 검토 후 페이지에 회원정보수정 완료 화면을 띄운다(A7)
6. 정보 수정 완료 시 메인 홈페이지로 이동한다.(E3)

**• 대안 흐름**

- (A1) 사용자가 변경 할 회원정보가 없다고 인식해 다른페이지 버튼을 클릭한다.
  - 1) 경고문구 없이 해당 화면만 이동한다.
- (A2) 사용자가 변경 할 회원정보가 없다고 인식해 뒤로가기를 누른다
  - 1) 경고문구 없이 해당 화면만 이동한다.
- (A3) 중복된 아이디를 입력할 경우
  - 1) "이미 존재하는 아이디입니다"라는 메세지를 출력한다
- (A4) 기존의 아이디를 입력할 경우
  - 1) "기존과 동일한 아이디입니다"라는 메세지를 출력한다
- (A5) 양식에 맞지 않는 특수기호를 사용했을 경우(아이디, 비밀번호)
  - 1) "양식에 맞게 아이디 or 비밀번호를 입력해주세요"라는 메세지를 출력한다
- (A6) 비밀번호와 비밀번호 확인란이 다를 경우
  - 1) "비밀번호가 일치하지 않습니다"라는 메세지를 출력한다.
- (A7) 회원정보 입력창에 빈칸이 있는 상태로 회원정보 수정 버튼을 클릭 할 경우
  - 1) 해당 입력칸을 확인하라는 경고창을 띄운다
- (A8) 주소입력을 사용자가 취소할 경우
  - 1) 팝업이 종료되며 기존의 수정화면으로 돌아간다.

**• 예외 흐름**

- (E1) 휴대폰 인증 실패 할 경우
  - 1) 확인 번호 입력시간이 지난 시 해당 번호로 인증이 불가능하게 처리한다.
- (E2) 이메일인증 실패 할 경우
  - 1) 이메일 확인 번호 입력시간이 지난 시 해당 번호로 인증이 불가능하게 처리한다.
- (E3) 회원정보 수정이 실패 할 경우(서버 에러 등의 기타 사유)
  - 1) 회원정보 수정이 실패 했다는 문구를 띄운 후 회원정보 수정 페이지로 돌아간다.

# 03 스토리보드\_대구요 프로젝트 설계 - 유스케이스 명세서

## 주문하기 및 결제

**행위자 :** 회원, 매장, 결제 시스템

**개요 및 설명 :** 회원은 장바구니에 담긴 메뉴들을 주문할 수 있다.

**사전 조건 :**

- 회원은 회원 권한 계정으로 로그인해 사용자 인증을 마친 상태다.
- 회원은 배달 요청 주소를 입력했다.
- 시스템은 회원의 배달 요청 주소를 기준으로 배달 가능 매장만을 검색해 사용자 화면에 출력했다.
- 회원은 주문할 메뉴를 하나 이상 장바구니에 추가했다.

### • 정상 흐름

- 회원은 장바구니 페이지로 이동한다.
- 시스템은 장바구니에 추가된 메뉴의 기본정보, 수량, 옵션, 총합 가격 정보를 화면에 출력한다.
- 회원은 장바구니에 담긴 메뉴 목록 중 주문할 메뉴를 선택한다.
- 회원은 주문 버튼을 클릭한다. (E1)
- 시스템은 회원의 주문 요청을 검토한다.
- 시스템은 요청 주문 목록과 배달지 정보를 출력하고, 추가 정보 입력과 결제 수단 선택을 요청한다.
- 회원은 기존 등록 정보를 사용하거나, 새로이 상세 요청 폼을 입력하고, 결제 수단을 선택한다. (E2)
- 회원은 결제 버튼을 클릭한다. (E3)
- 시스템은 선택된 결제 수단에 따른 결제 정보 입력 화면을 출력한다.
- 사용자는 결제 정보를 입력하고 결제를 진행한다. (A1)
- 시스템은 결제가 승인됨을 확인한 후 매장에 주문을 요청한다.
- 시스템은 주문 대기 화면을 띄운다.
- 매장은 주문 목록을 확인하고 요청을 수락한다. (E4)
- 시스템은 주문 내역 페이지를 띄우고 주문에 성공함을 출력한다.

### • 대안 흐름

#### A1) 회원이 결제 수단에 쿠폰을 포함시킬 경우

- 기준 가격 정보에 해당 쿠폰 할인을 적용한 가격을 계산한다.
- 회원 결제 금액을 쿠폰 적용 결과 만큼 차감한다.

### • 예외 흐름

#### (E1) 장바구니 선택 메뉴 총합 가격이 배달 최소 주문 금액 미만인 상태에서 주문 버튼을 클릭한 경우

- 최소 주문 금액 미만은 주문 할 수 없다는 에러 메세지를 출력한 후 장바구니 페이지로 돌아간다.

#### (E2) 주문 상세 정보를 입력하지 않고 결제 버튼을 클릭할 경우

- 주문 상세 정보를 모두 입력해야 한다는 메세지를 출력한다.

#### (E3) 결제 승인이 거부됐을 경우

- 결제 실패 메세지를 출력한 후 결제 수단 선택 페이지로 돌아간다.

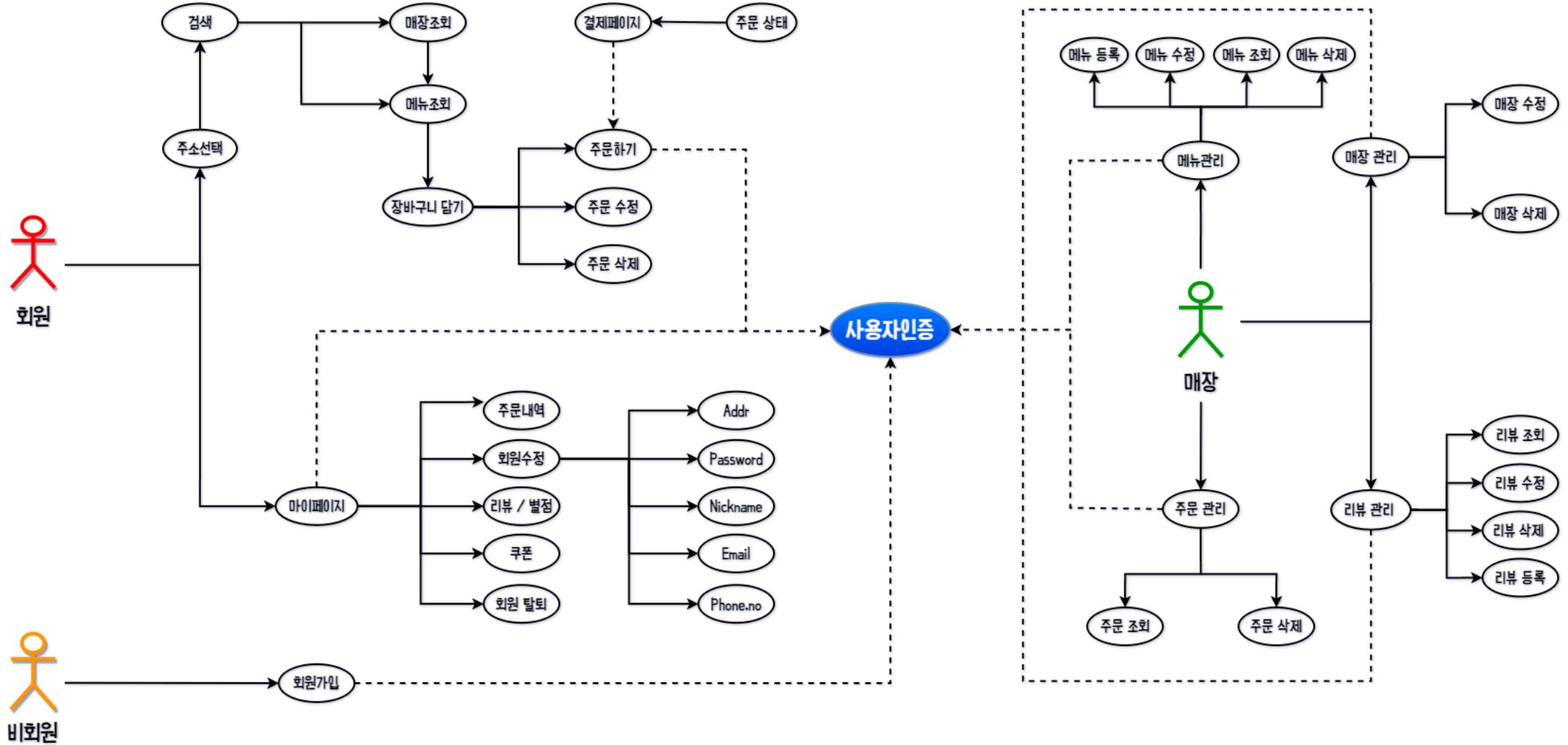
#### (E4) 매장이 주문 요청을 거부할 경우

- 해당 주문 건에 대한 결제 취소 처리를 진행한다.
- 사용자에게 주문 실패 안내 sms를 발송한다.

- 시스템은 주문 내역 페이지를 띄우고 주문이 취소됨을 회원에게 알린다

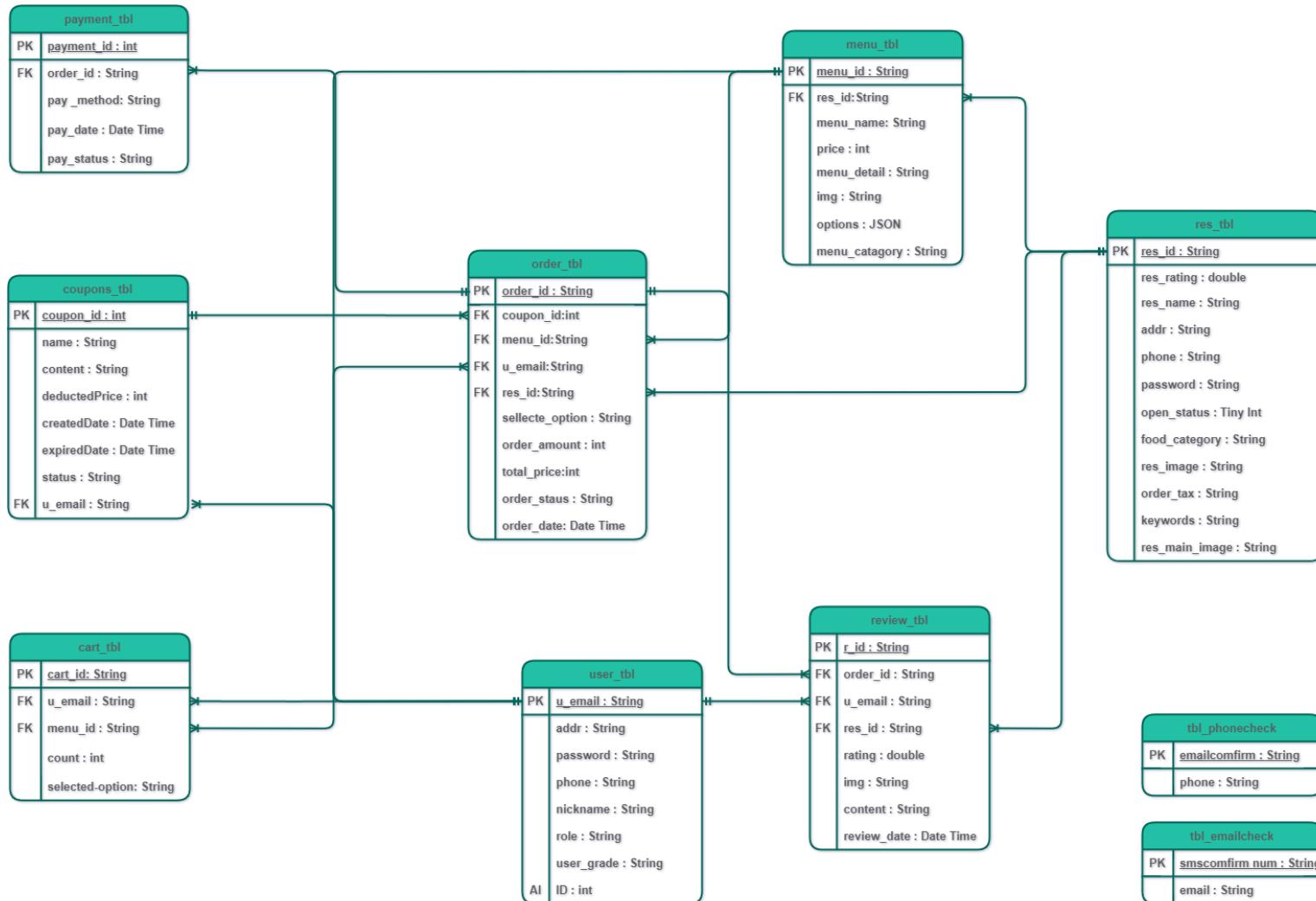
# 03 스토리보드\_대구요

## 프로젝트 설계 - Usecase diagram

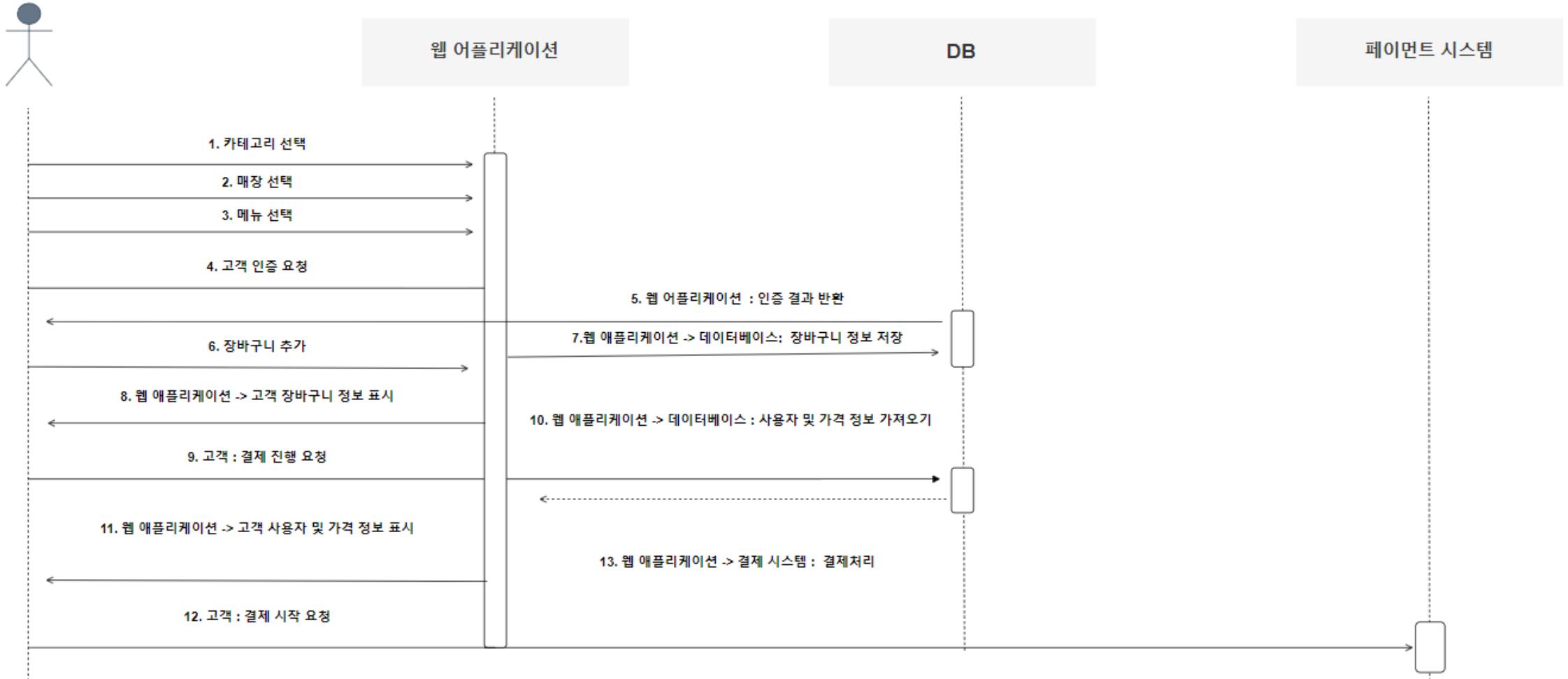


# 03 스토리보드\_대구요

## 프로젝트 설계 - ERD



# 03 스토리보드\_대구요 프로젝트 설계 - Sequence diagram



# 03 스토리보드\_대구요 프로젝트 설계 - MVC패턴 적용

Controller

```

userController
+ userService : UserService

loginPage() : String
getMyPage(Model model, HttpSession session) : String
memberJoinPage() : void
memberJoin(UserDto dto) : String
memberUpdate(Model model, String email) : String

memberUpdate(UserDto dto) : int
memberDelete(String u_email) : int

```

Model

```

User Service
- passwordEncoder : BCryptPasswordEncoder
- mapper : UserMapper
- couponMapper : CouponMapper
- reviewMapper : ReviewMapper
- orderMapper : OrderMapper

+ memberJoin(UserDto dto) : Int
+ userSearch(String u_email) : UserDto
+ userCouponCount(String u_email) : Int
+ userReviewCount(String u_email) : Int
+ userOrderCount(String u_email) : Int
+ getUserInfoByEmail(String u_email) : UserDto
+ memberUpdate(UserDto dto) : Int
+ memberDelete(String u_email) : Int

```

```

UserMapper
+ selectAll() : List<ResDto>
+ selectOne(String u_email) : UserDto
+ insertUser(UserDto) : Int
+ updateUserAddr(UserDto) : Int
+ updateUserSimple(UserDto) : Int
+ updateUserWithAddr(UserDto) : Int
+ deleteUser(String u_email) : Int

```

```

UserDto
- id : Int
- u_email : String
- addr : String
- password : String
- phone : String
- nickname : String
- user_grade : String
- role : String

```

Database



```

ResController
- resService : ResService

+ resList(Model model) : String
+ resCList(Model model, String food_catogory) : String

```

```

ResService
- mapper : ResMapper

+ searchAll() : List<ResDto>
+ searchCategory(String food_catogory) : List<ResDto>
+ searchOne(String res_id) : ResDto

```

```

ResMapper
+ selectAll() : List<ResDto>
+ selectOne(String res_id) : ResDto
+ selectCategory() : ResDto
+ insertRes() : Int
+ updateRes() : Int
+ deleteRes() : Int

```

```

ResDto
- res_id : String
- res_name : String
- addr : String
- phone : String
- password : String
- res_rating : double
- open_status : boolean
- food_catogory : String
- res_image : String
- min_order_price : String
- order_text : String
- keywords : String
- res_main_image : String

```

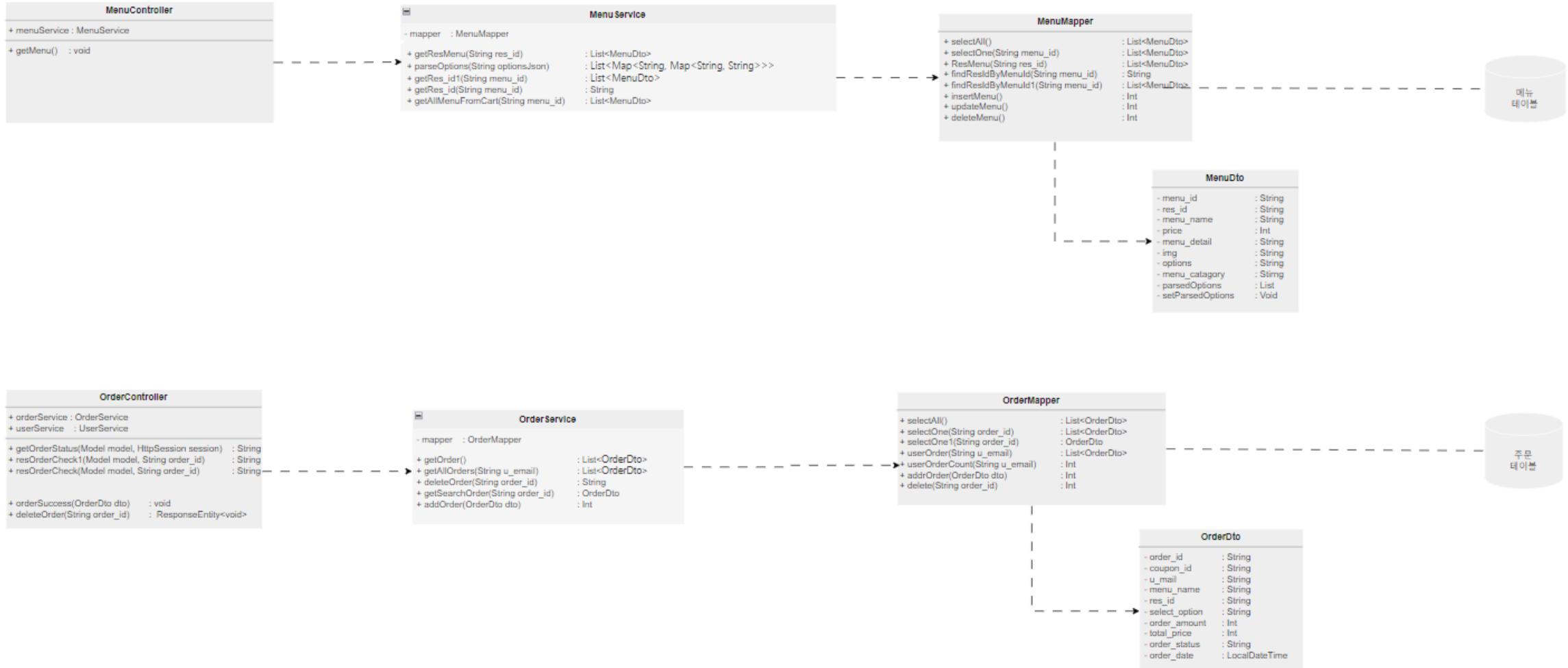


# 03 스토리보드\_대구요 프로젝트 설계 - MVC패턴 적용

Controller

Model

Database



# 03 스토리보드\_대구요 프로젝트 설계 - MVC패턴 적용

Controller

```

CartController
- cartService : CartService
- resService : ResService
- menuService : MenuService

+ getCart(Model model, HttpSession session) : String
+ updateOrder(CartDto dto) : void
+ addToCart(CartDto dto) : ResponseEntity<Map<String, Boolean>>
+ deleteMenu(String cart_id) : int
  
```

Model

```

CartService
- cartMapper : CartMapper
- mapper : OrderMapper
- menuMapper : MenuMapper
- payMapper : PaymentMapper

+ selectAllByEmail(String u_email) : List<CartDto>
+ updateOrderAmount(CartDto dto) : Void
+ addToCart(CartDto cartDto) : Boolean
+ deleteFromCart(String cart_id) : Int
  
```

```

CartMapper
+ CartList() : List<CartDto>
+ updateOrder(CartDto dto) : Void
+ deleteOrder(String cart_id) : Int
+ selectAll(String u_email) : List<CartDto>
+ updateCount(String cart_id, Int count) : Void
+ findMenuIdByEmail(String u_email) : String
+ deleteByEmail(String u_email) : Void
+ insertCart(CartDto cartDto) : Int
  
```

Database



```

CartDto
- cart_id : String
- u_email : String
- menu_id : String
- count : Int
- selected_option : String
- res_name : String
- price : Int
  
```

```

PaymentController
- apiKey : String
- IMPORT_TOKEN_URL : String
- accessToken : String

+ getToken2(Model model) : String
+ searchOne(String payment_id) : String
+ searchAll() : String
+ refund() : String
  
```

```

PaymentMapper
+ selectAll() : List<PaymentDto>
+ selectOne(String payment_id) : PaymentDto
+ InsertPayment(PaymentDto paymentData) : Int
+ updatePayment() : Int
+ deletePayment() : Int
  
```



```

PaymentDto
- payment_id : String
- order_id : String
- pay_method : String
- pay_date : LocalDateTime
- pay_status : String
  
```

# 03 스토리보드\_대구요 프로젝트 설계 - MVC패턴 적용

Controller

Model

Database

**ReviewController**

```
- reviewService : ReviewService
- menuService : MenuService
- resService : ResService
+ getMenu(Model model, String res_id) : String
```

**ReviewMapper**

```
selectAll() : List<ReviewDto>
selectOne(String ReviewDto) : ReviewDto
insertReview() : Int
userReview(String u_email) : ReviewDto
userReviewCount(String u_email) : Int
resReview(String res_id) : ReviewDto
updateReview() : Int
deleteReview() : Int
```

**ReviewDto**

```
r_id : String
u_email : String
order_id : String
res_id : String
rating : Double
img : byte[]
content : String
reviewDate : LocalDateTime
```

리뷰  
테이블

**CouponController**

```
- couponService : CouponService
```

**CouponMapper**

```
selectAll() : List<CouponDto>
selectOne(String coupon_id) : CouponDto
userCoupon(String u_email) : CouponDto
userCouponCount(String u_email) : Int
insertCoupon(Int) : Int
updateCoupon(Int) : Int
deleteCoupon(Int) : Int
```

**CouponDto**

```
coupon_id : Int
name : String
content : String
deductedPrice : Int
createDate : LocalDateTime
expiredDate : LocalDateTime
status : String
u_email : String
```

쿠폰  
테이블

# 03 스토리보드\_대구요 프로젝트 설계 - MVC패턴 적용

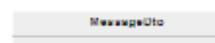
## Controller



## Model



## Database



# 04 스토리보드\_대구요 핵심 기능 및 구현 내용 - 구현 서비스

## • 회원 (UserService)

- 회원 가입
  - Email, Sms 인증 절차로 본인 확인
  - 회원의 기존 등록 여부 확인
- 이메일에 따른 단일 회원 정보 조회
  - 회원 개인 정보 조회
  - 쿠폰 수량 조회
  - 주문 횟수 조회
  - 리뷰 작성 수 조회
- 회원 정보 수정
- 회원 탈퇴

## • 메뉴 (MenuService)

- 특정 매장의 모든 메뉴 조회
- 개별별 메뉴 이름, 설명, 옵션, 가격 등 상세 정보 조회
- 특정 메뉴의 등록 매장 정보 조회

## • 장바구니 (CartService)

- 회원의 장바구니 정보 추가
- 회원의 장바구니 정보 조회
- 회원의 특정 장바구니 정보 수정
- 회원의 모든 장바구니 정보 삭제
- 특정 장바구니 정보 삭제

## • 주문 (OrderService)

- 선택된 항목 주문
- 모든 주문 조회
- 주문 ID에 따른 단일 주문 조회
- 회원의 모든 주문 내역 조회
- 특정 주문 정보 삭제

## • 매장 (Restaurant)

- 모든 매장 리스트 조회
- 카테고리에 따른 매장 리스트 조회
- 단일 매장 위치, 영업 여부, 별점 등 상세정보 조회

# 04 스토리보드\_대구요 핵심 기능 및 구현 내용 - API 설계

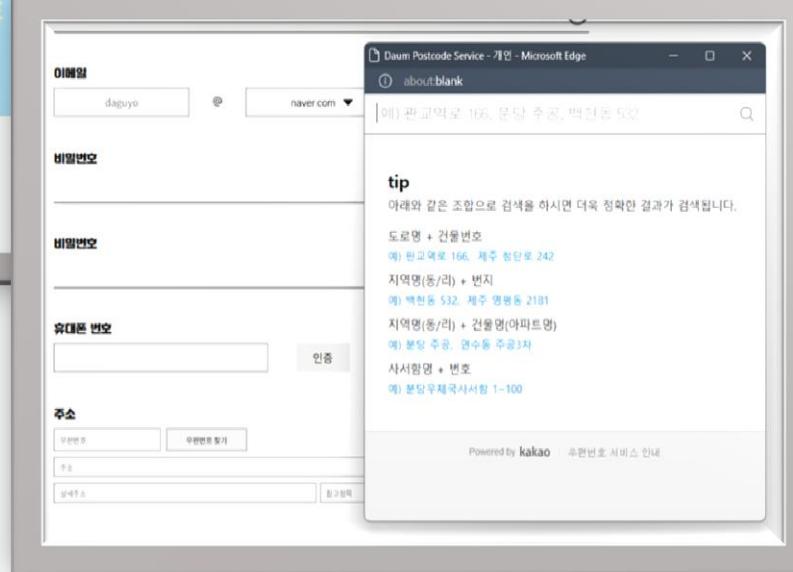
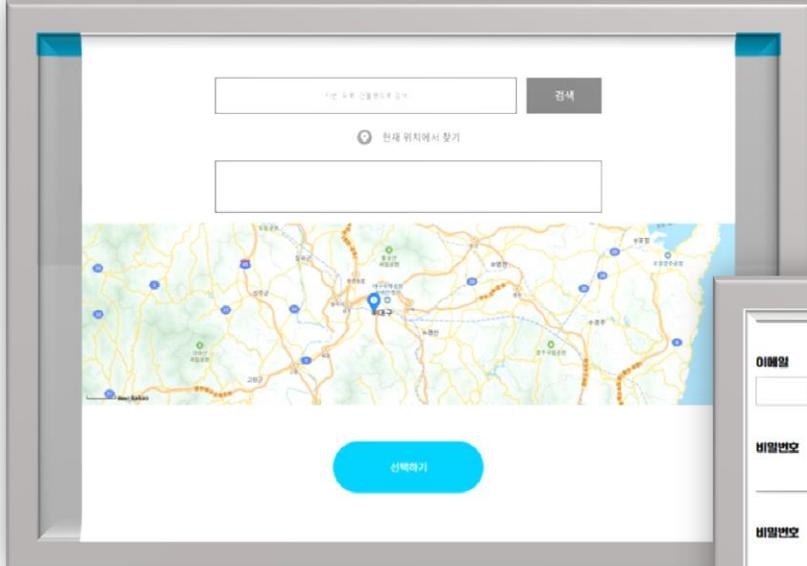
	Method	URL	Description	Return Page	Return Value
Home	GET	/	메인 페이지	Index	
	GET	/import	Import 페이지	Import	
Menu	GET	/menu_category	메뉴 카테고리 조회		Void
	GET	/review_tab	리뷰 탭에서 메뉴 조회	review_tab	
User	GET	/login	로그인 페이지		Void
	GET	/myPage	마이페이지 조회	Mypage	
	GET	/memberJoin	회원 가입 페이지		Void
	POST	/memberJoin	회원 가입 처리	성공 => login 실패 => memberJoin	
	GET	/memberUpdate	회원 정보 수정 페이지	memberUpdate	
	POST	/memberUpdate	회원 정보 수정 처리		Int (수정 결과)
	PUT	/memberUpdate	회원 정보 업데이트		Int (수정 결과)
	DELETE	/memberUpdate	회원 정보 삭제		Int (수정 결과)
Restaurant	GET	/selectRest	모든 레스토랑 조회	selectRest	
	GET	/selectRest/{food_catagory}	특정 카테고리 레스토랑 조회	selectRest	
	GET	/resCheck	레스토랑 주문 확인		Void
Review	GET	/review_tab	리뷰 탭에서 메뉴 조회	review_tab	

# 04 스토리보드\_대구요 핵심 기능 및 구현 내용 - API 설계

	Method	URL	Description	Return Page	Return Value
Cart	GET	/cart	장바구니 정보 조회	cart	
	POST	/cart	장바구니 메뉴 추가		JSON(성공여부)
	DELETE	/deleteMenu	장바구니 메뉴 삭제		Int(삭제 결과)
	POST	/updateOrder	장바구니 수량 정보 수정		void
Order	GET	/orderStatus	주문 상태 조회	orderStatus	
	GET	/orderCheck	주문 확인		resCheck
	GET	/orderCheck1	주문 확인(특정 주문 ID 검색)		resCheck
	POST	/orderSuccess	주문 성공 처리		void
	DELETE	/deleteOrder/{order_id}	주문 삭제		JSON (삭제 결과)
Payment	GET	/verifylamport2/getToken	토큰 발급 페이지 반환	/verifylamport2/getToken	
	POST	/verifylamport2/getToken	토큰 값 저장 및 토큰 발급 결과 조회		
	GET	/verifylamport2/searchOne	주문 단건 조회		Index
	GET	/verifylamport2/searchAll	주문 전체 조회		index
Sms	POST	/sms/send			SmsResponseDto
	GET	/checkPhone			ResponseEntity<String>
Mail	POST	/sendEmail	이메일 전송		JSON (확인 메시지)
	GET	/checkEmail	이메일 확인		JSON (서버인증번호/오류 메시지)

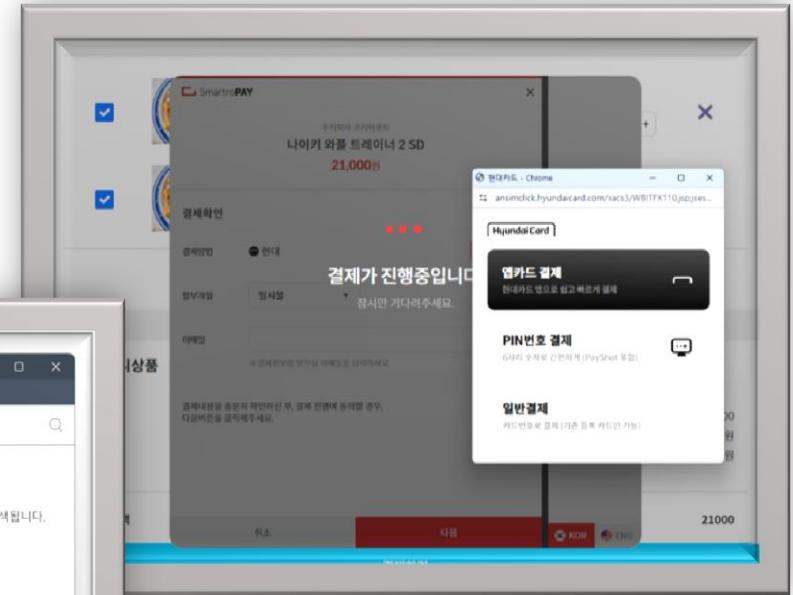
# 04 스토리보드\_대구요 핵심 기능 및 구현 내용 - 사용 API

## 카카오 맵 API



## 다음 주소 API

## 아임포트 결제 API

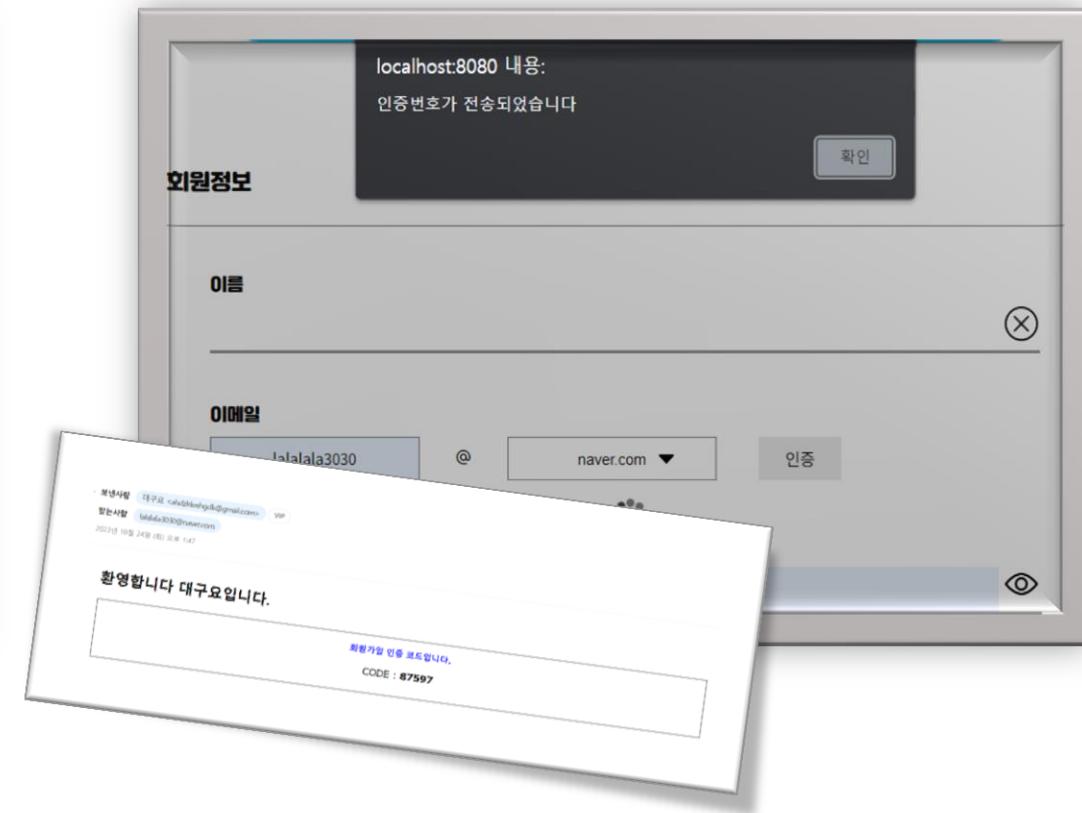


# 04 스토리보드\_대구요 핵심 기능 및 구현 내용 - 사용 API

## Naver Sms API



## JavaMail



# 04 스토리보드\_대구요 핵심 기능 및 구현 내용 - 보안

## Spring Security 사용

접근 주체는 로그인을 통해 대구요 서버 사용자 인증 요청 전송 가능

인증된 사용자의 'ROLE'에 따라 대구요 API 접근 가능 범위와 그 권한을 인가

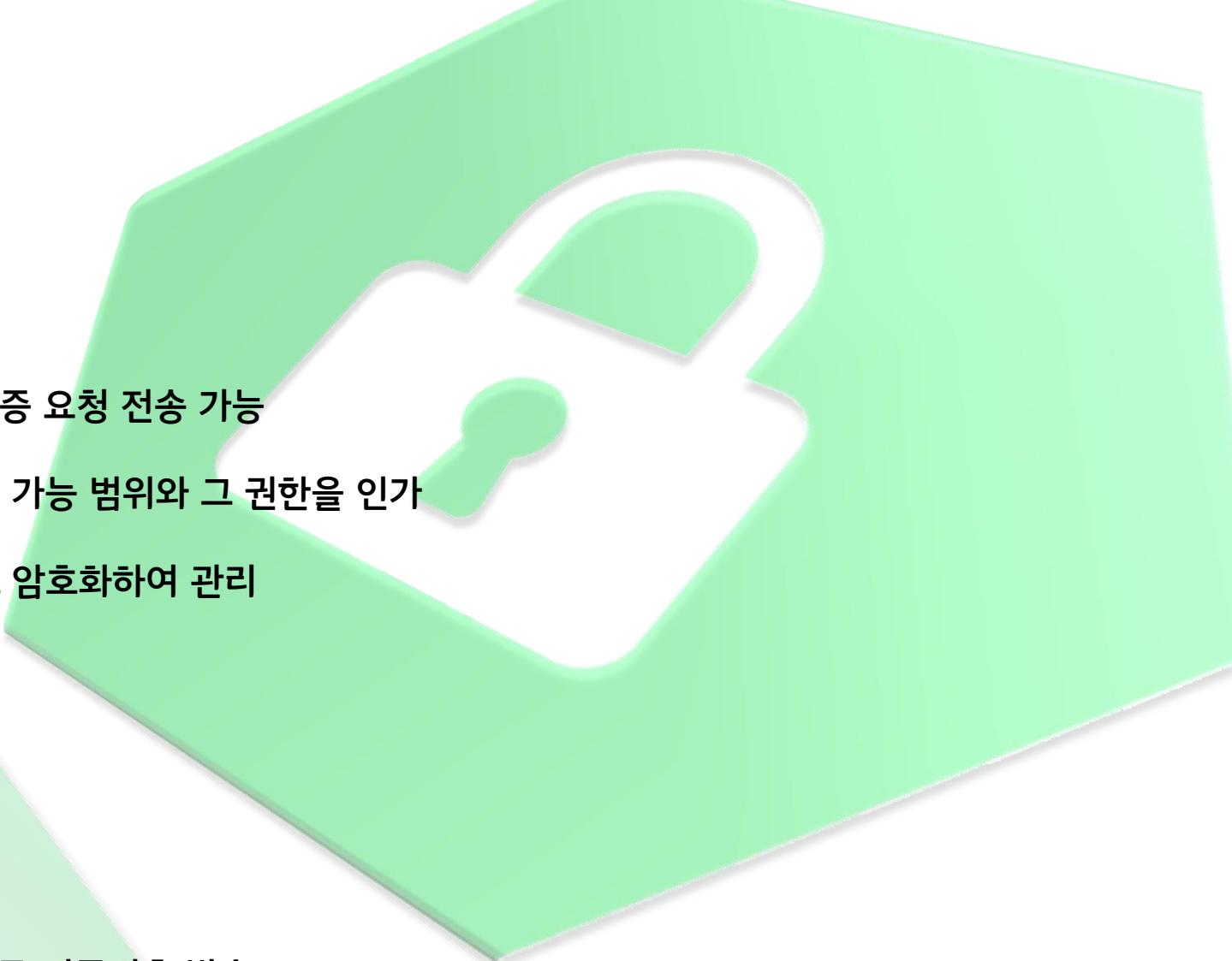
사용자 비밀번호는 BcryptPasswordEncoder로 암호화하여 관리

## 회원가입 본인 확인 절차 강화

JavaMail, NaverSms API 사용

회원가입 시, 사용자가 입력한 e-mail과 전화번호로 인증번호 발송

발송된 인증번호와 사용자가 입력한 인증번호 일치할 경우 회원가입 가능

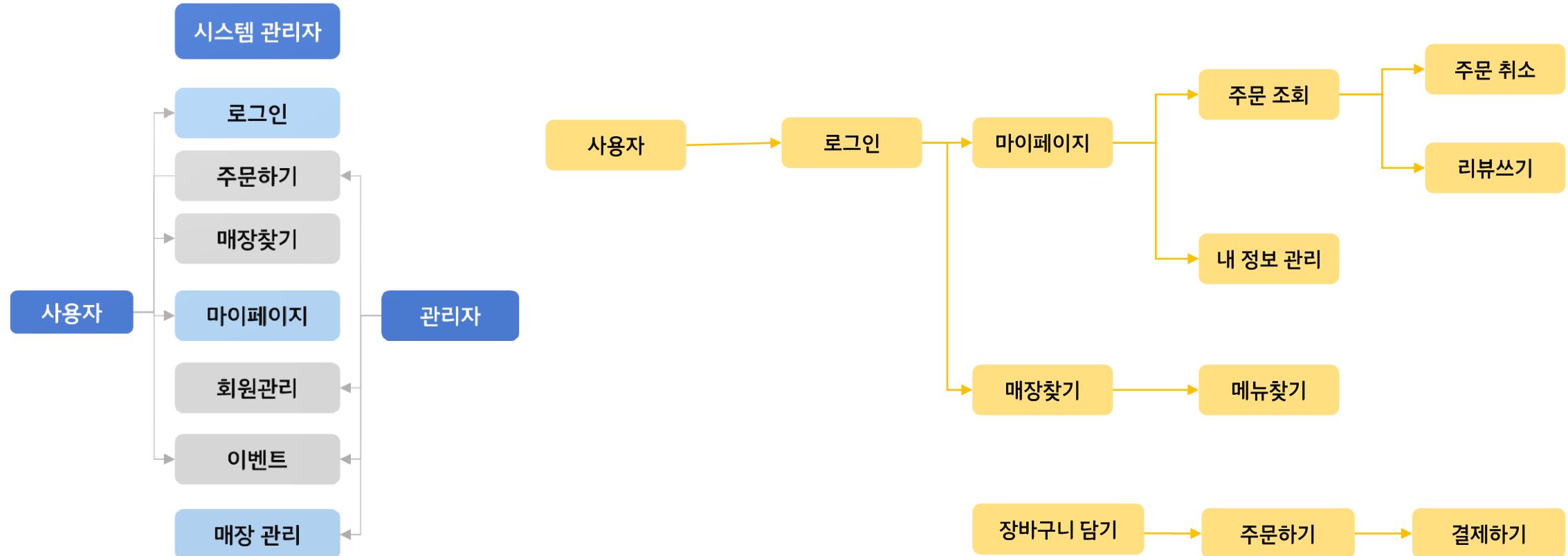


# 05 스토리보드\_대구요

## 스크린리스트

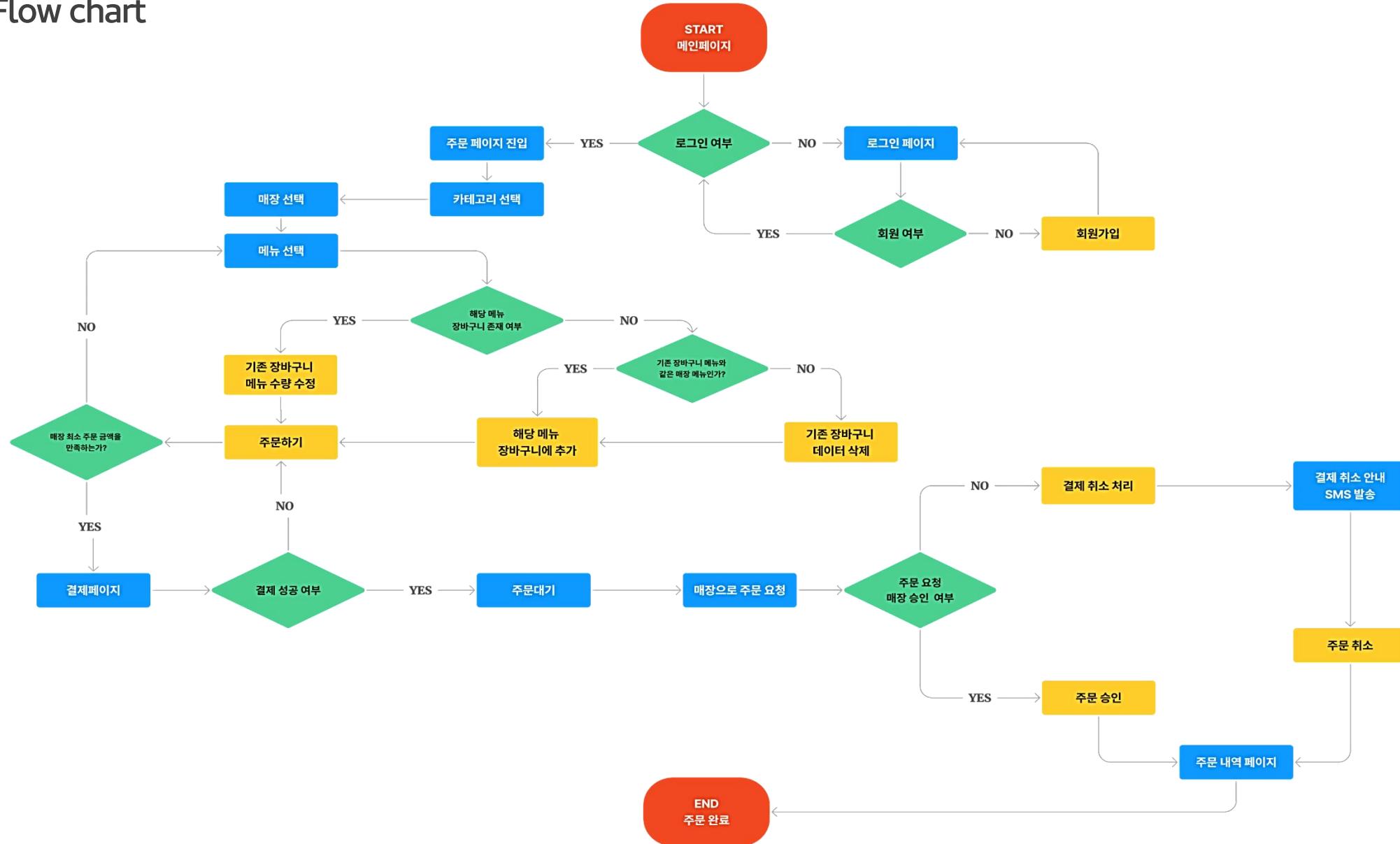
대메뉴	중메뉴	소메뉴	Screen ID	Page title	Description	비고
로그인			UI-A-1011	로그인	로그인 페이지	
			UI-A-1012	회원가입	회원가입 페이지	
홈	추천메뉴		UI-B-1011	홈 > 추천메뉴	현재 가장 할인율이 높은 순으로 표시	
	최근 주문내역		UI-B-1012	홈 > 최근 주문내역	최근 주문한 내역에서 내림차순으로 표시	
	이벤트		UI-B-1013	홈 > 이벤트	이용 가능한 모든 이벤트 확인	
주문	배달주문		UI-C-1011	주문 > 배달주문	배달주문이 가능한 가게 조회	
	포장주문		UI-C-1012	주문 > 포장주문	포장주문이 가능한 가게 조회	
	선물하기		UI-C-1013	주문 > 선물하기	내가 받은 선물 내역을 확인	
	메뉴 카테고리 선택	매장 선택	UI-C-1014	주문 > 배달(포장)주문 -> 카테고리-> 매장	선택된 수령 방식, 카테고리에 알맞은 가게 조회	
	매장 선택	메뉴 담기	UI-C-1015	주문 > 배달(포장)주문 -> 카테고리 -> 매장 > 메뉴 선택	선택된 가게의 메뉴 조회 및 메뉴 선택	
마이페이지	회원정보수정		UI-D-1011	마이페이지 > 회원정보 수정	사전에 기입한 회원정보 수정 및 삭제	
	리뷰관리		UI-D-1012	마이페이지 > 리뷰관리	작성했던 리뷰 조회, 수정, 삭제	
	결제관리		UI-D-1013	마이페이지 > 결제관리	결제한 내역 조회	
	공지사항		UI-D-1014	마이페이지 > 공지사항	관리자가 작성한 공지사항 확인	
	문의사항	전화상담	UI-D-1015	마이페이지 > 문의사항 > 전화상담	전화문의 서비스 신청 및 이용	
		AI상담	UI-D-1016	마이페이지 > 문의사항 > AI상담	자주하는 질문 위주의 문의 서비스 이용	
장바구니	주문결제	결제창	UI-E-1011	장바구니 > 주문결제 > 결제창	장바구니에 담은 내역 수정 및 결제	

# 06 스토리보드\_대구요 Process



# 07 스토리보드\_대구요

## Flow chart



# 08 스토리보드\_대구요

## Permission

대분류	중분류	사용자	보기	수정	삭제	쓰기	비고
홈	매장조회	회원	O	X	X	X	
		비회원	X	X	X	X	
		관리자	O	O	O	O	
	최근 주문내역	회원	O	X	O	X	
		비회원	X	X	X	X	최근 주문내역은 회원만 조회 가능
		관리자	O	O	O	O	
	장바구니	회원	O	O	O	O	
		비회원	X	X	X	X	
		관리자	O	O	O	O	

# 09 스토리보드\_대구요 POLICY

대구분	중구분	Description	비고
회원가입 정책	아이디	<ul style="list-style-type: none"><li>- ID는 영문자나,숫자,또는 영문자와 숫자의 조합으로 4~12까지 조합하여 만들 수 있으며 한글,특수문자,띄어쓰기 등을 포함할 수 없음</li></ul>	
	패스워드	<ul style="list-style-type: none"><li>- 대/소문자, 숫자, 특수문자를 반드시 혼용하여 10글자 이상의 패스워드를 사용하도록 한다.</li><li>아이디와 같은 패스워드는 사용하지 못하도록 한다.</li></ul>	
	주소	<ul style="list-style-type: none"><li>- 주소입력은 도로명 주소를 기본으로 한다</li><li>- 지번주소는 선택 시 사용한다.</li></ul>	
	이메일	<ul style="list-style-type: none"><li>- 개별메일명@도메인명을 기본으로 입력 받는다.</li><li>- 도메인명은 선택하여 지정할 수 있다</li></ul>	
	닉네임	<ul style="list-style-type: none"><li>- 닉네임은 한글,대/소문자,숫자를 혼합하여 만들수 있으며 특수문자는 사용할 수 없으며, 10글자까지의 조합으로 한다.</li><li>-중복한 닉네임은 사용할 수 없다.</li></ul>	

# 10 스타일 가이드\_대구요 구동환경

## 구동환경

컴퓨터 OS

Window

웹 브라우저

Chrome

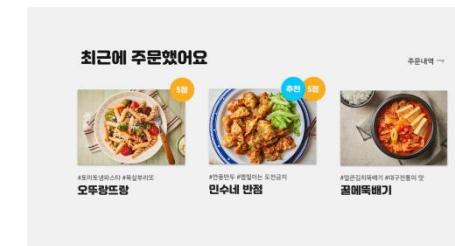
모니터 해상도

1920\*1080

## 프레임세트

구분	One Frame
컨텐츠구성	<ul style="list-style-type: none"><li>- 홈, 주문, 마이 페이지</li><li>- 장바구니로그인 · 로그아웃, 배달 위치 찾기</li></ul>
디자인	<ul style="list-style-type: none"><li>- 공통 헤더영역(로고 &amp; 메뉴 삽입)</li><li>- 컨텐츠 영역 색상 및 선으로 구분</li></ul>
레이아웃	[스크롤형식] 상단로고 및 메뉴, 메인영역, 컨텐츠영역, 푸터 영역

## 스크롤 형식



# 11 스타일 가이드\_대구요 레이아웃

1. **색상**: 홈페이지 대표색(01c0f1), 강조색(ffdf3e), 배경색(f2f2f2)

2. **폰트**: Notosans(일반문단), 한돈삼겹살체 (일반문구),여기어때(강조)

3. **로고**: 기존의 대구로의 로고를 리뉴얼

색상



FFFFF



폰트

**NOTOSANS**

**여기어때 잘난체**

**한돈 삼겹살체**

로고



헤더 1300 \*150(PX)

메인 1300 \*500(PX)

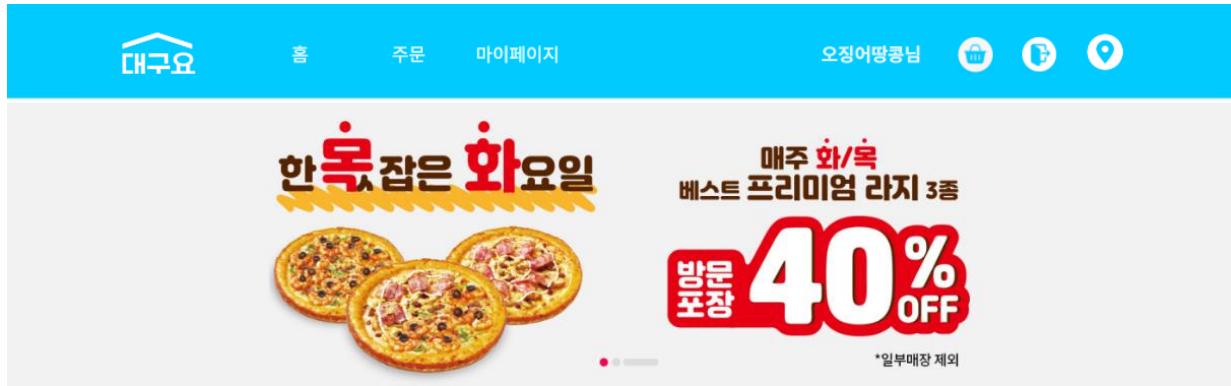
컨텐츠 1300 \*1000(PX)

푸터 1300 \*300(PX)

# 11 스타일 가이드\_대구요 레이아웃\_헤더



# 11 스타일 가이드\_대구요 레이아웃\_컨텐츠



## 불금엔 '쿨'하게 대구로



KFC 19시 부터  
전메뉴 50%



베스킨라빈스  
신메뉴 최대 30%



도미노피자  
전메뉴 최대 30%

전체보기 →

## 메인

- 화면 자동 슬라이드
- 몇 번 째 화면인지, 화면이 몇 개인지를 나타내는 원형 버튼 구현

## 컨텐츠

- 홈: 이벤트, 추천메뉴
- 주문: 카테고리 선택, 가게선택, 메뉴선택
- 마이페이지: 주문내역, 정보수정
- 장바구니: 주문과정, 결제과정

# 11 스타일 가이드\_대구요 레이아웃\_푸터

- 클릭 시 소셜미디어 페이지 이동

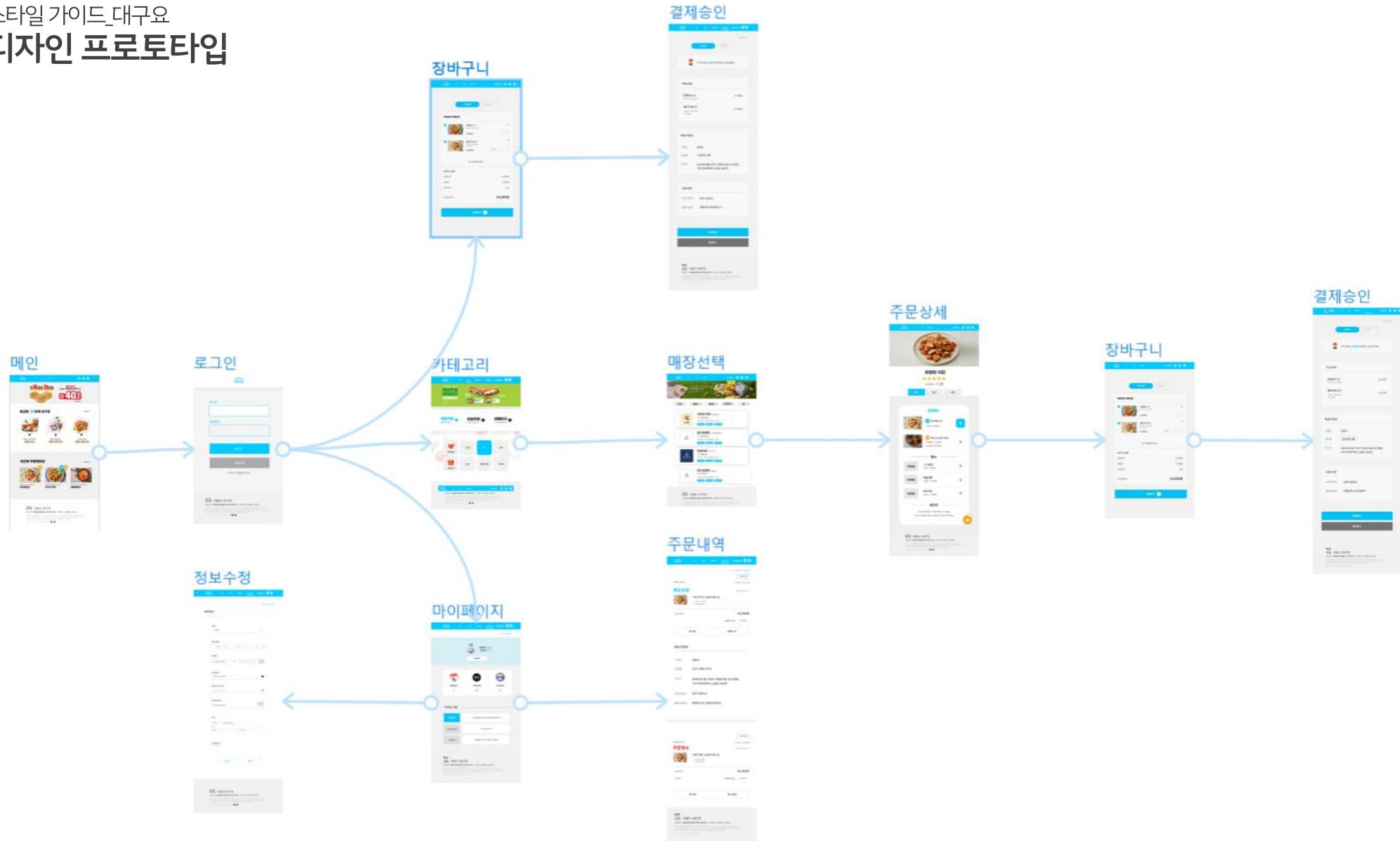
## 푸터로고

- 클릭 시 메인화면으로 이동

- 약관 및 기타 정보

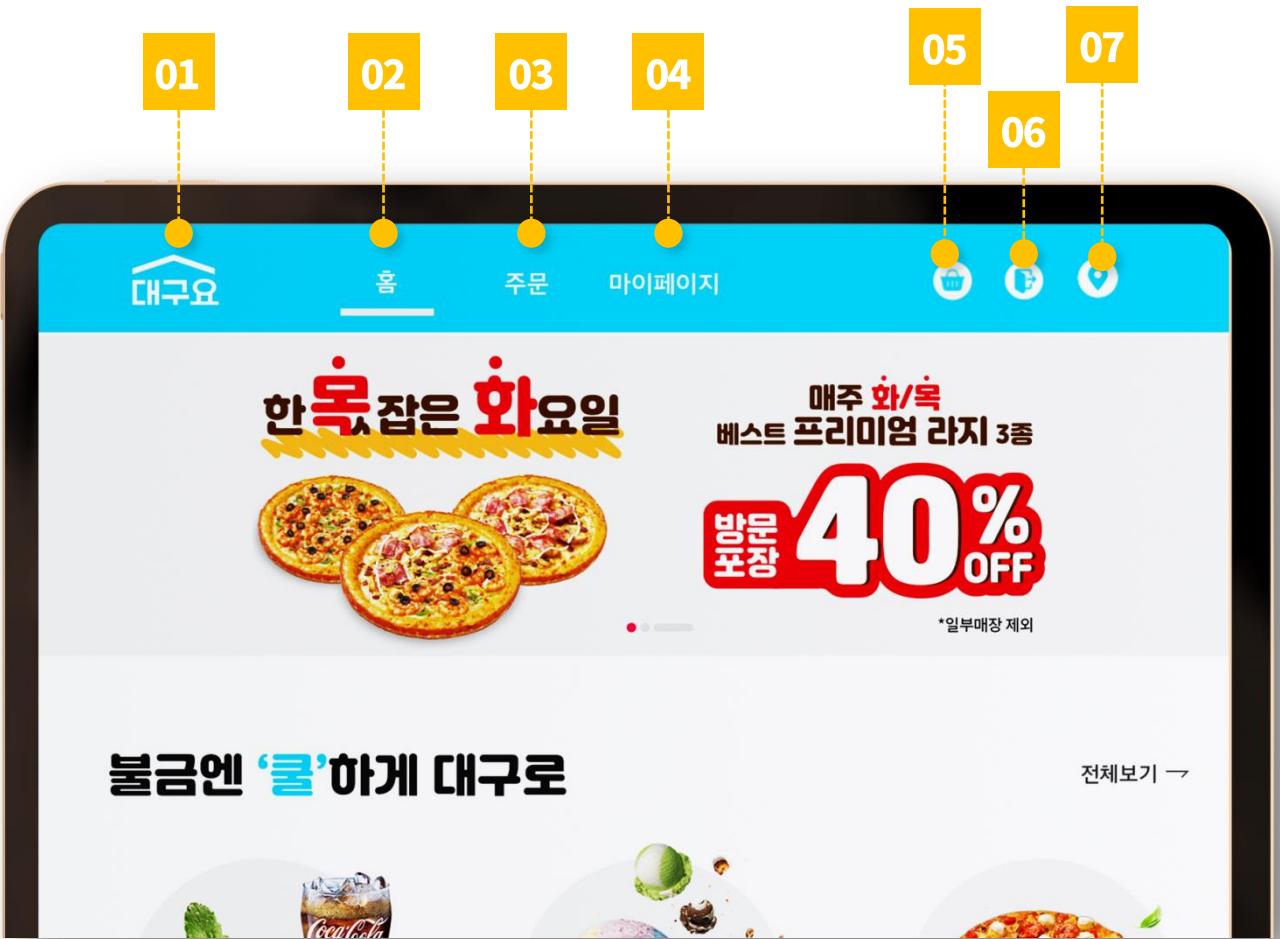


# 12 스타일 가이드\_대구요 디자인 프로토타입



## Description

- 01 로고 클릭 시 홈페이지 이동
- 02 클릭 시 홈페이지 이동(기본 홈화면)
- 03 음식 카테고리 선택 페이지 이동
- 04 클릭 시 마이페이지 이동
- 05 클릭 시 장바구니 이동
- 06 로그아웃
- 07 배달위치 찾기

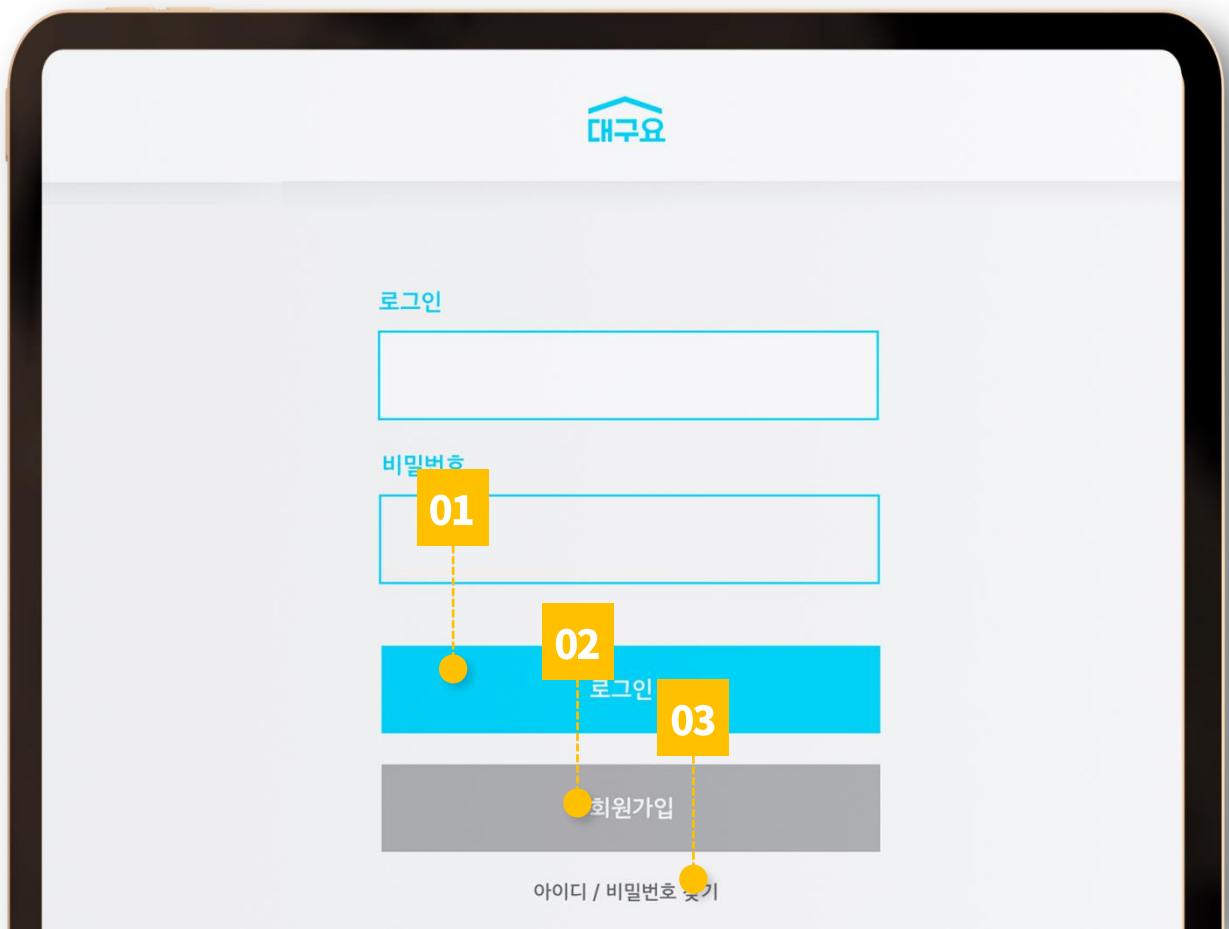


SCREEN PATH

메인화면 &gt; 로그인

## Description

- 01 로그인 성공 시 홈 화면 이동  
로그인 실패 시 로그인 이동
- 02 회원가입 페이지 이동
- 03 아이디/ 비밀번호 찾기 페이지 이동



## Description

UI-B-1000, UI-A-1012

PAGETITE

JOIN

SCREEN

UI-A-1012

AUTHOR

3

DATE

2023.10.26

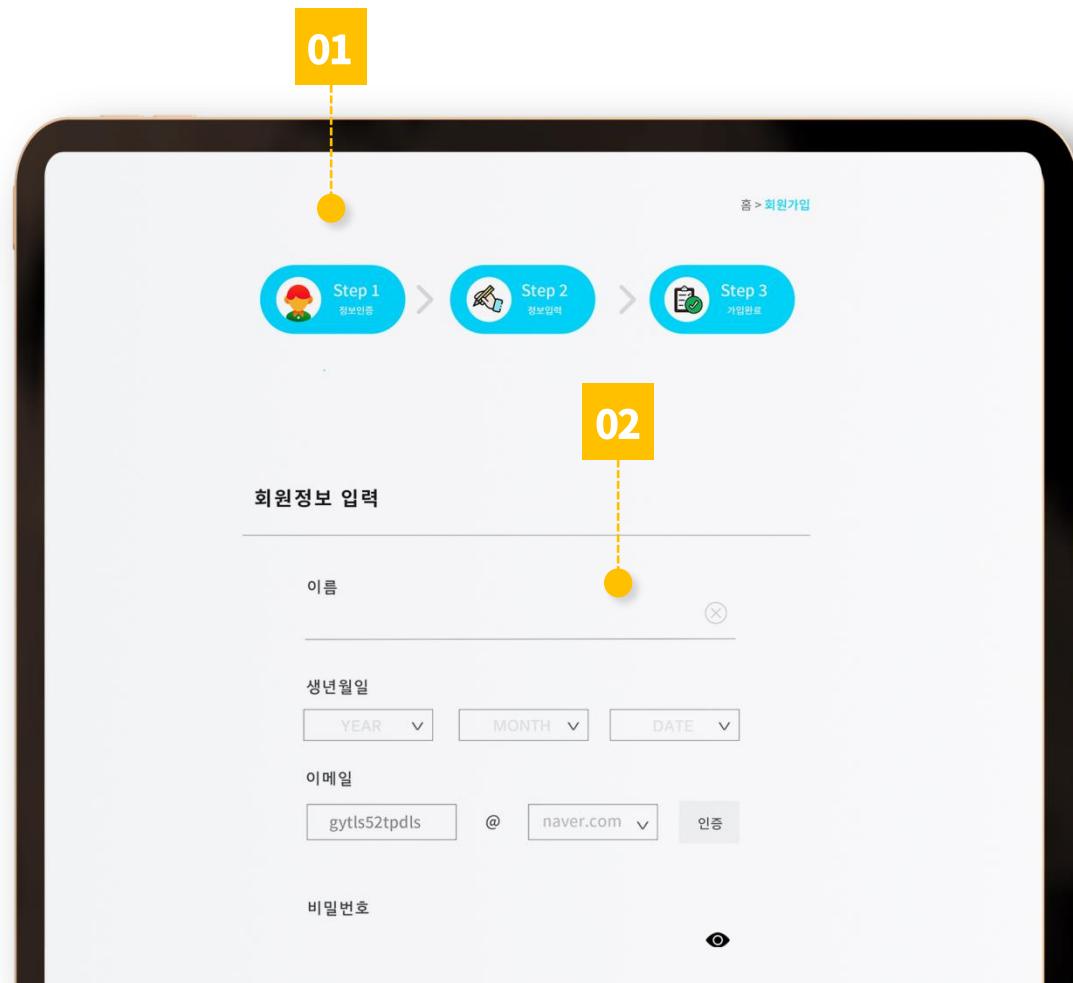
SCREEN PATH

메인화면 > 로그인 > 회원가입

## Description

01 회원가입 절차 표시

02 회원정보 입력 및 인증절차



## Description

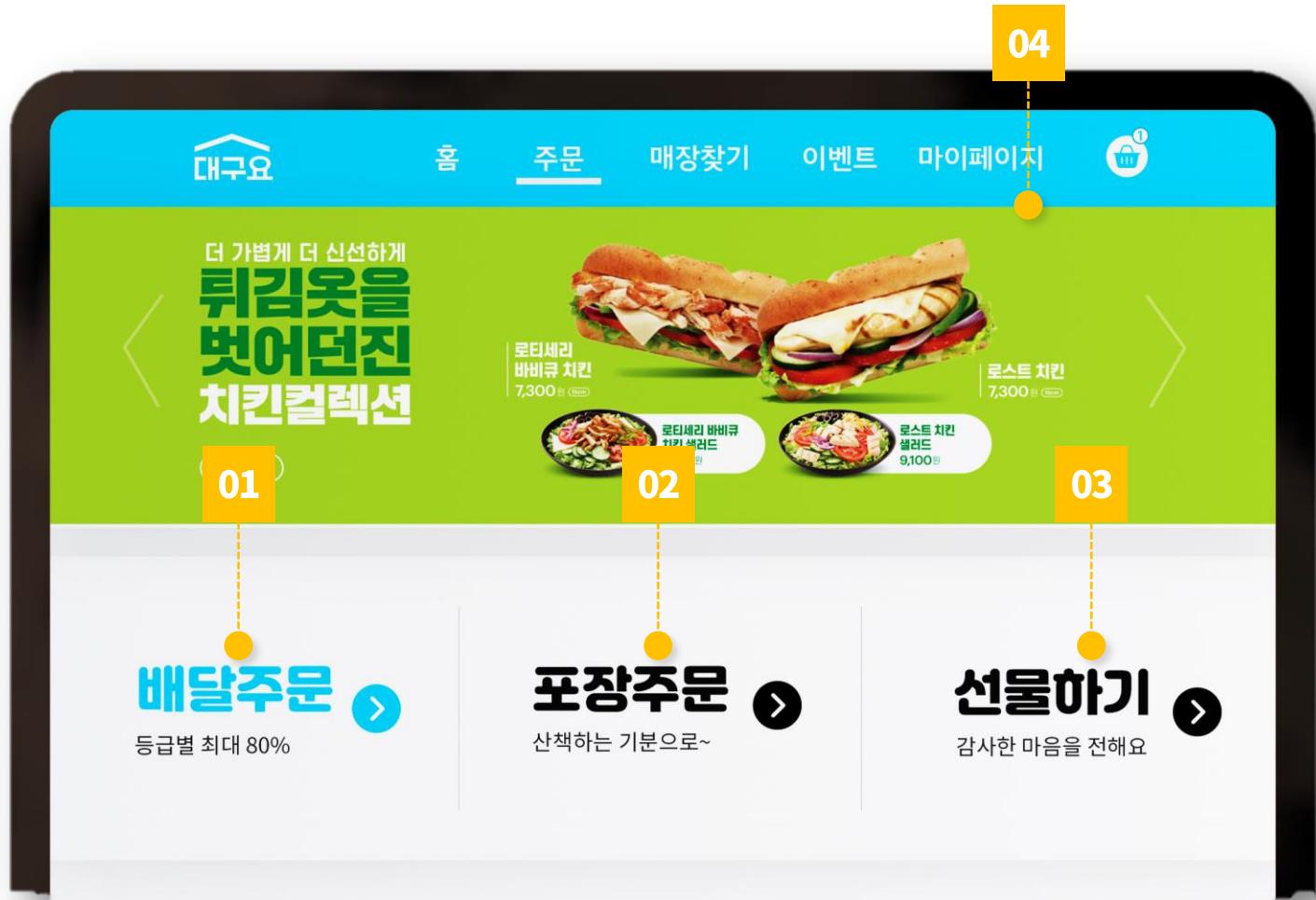
UI-C-1000

SCREEN PATH

메인화면 &gt; 로그인 &gt; 메뉴 카테고리 선택

## Description

- 01 배달주문 옵션 클릭 시  
옵션에 따라 카테고리가 변경됨
- 02 포장주문 옵션 클릭 시  
옵션에 따라 카테고리가 변경됨
- 03 선물하기 옵션 클릭 시  
옵션에 따라 카테고리가 변경됨



## Description

- 
- 05 선물 찾기 및 추천메뉴 버튼은 옵션이 바뀌어도 변하지 않는 디폴트 버튼
  - 06 매장 선택 페이지로 이동

## Description

---

UI-C-1011, UI-C-1012,  
UI-C-1013, UI-C-1014

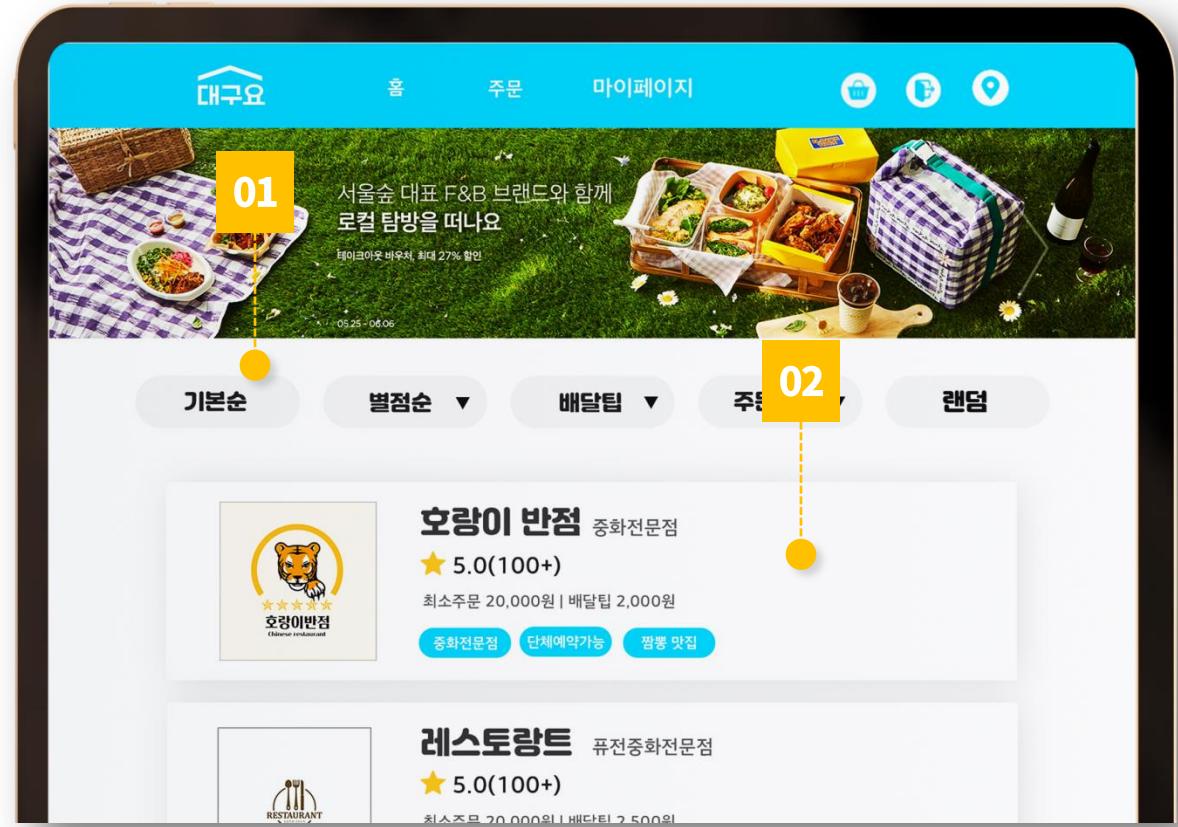


PAGETITE	RESTAURAN T	SCREEN	UI-C-1014	AUTHOR	3	DATE	2023.10.26
SCREEN PATH	메인화면 > 로그인 > 카테고리 선택 > 매장선택						

## Description

01 클릭 시 가게 순서 정렬

02 메뉴 선택 페이지로 이동



## Description

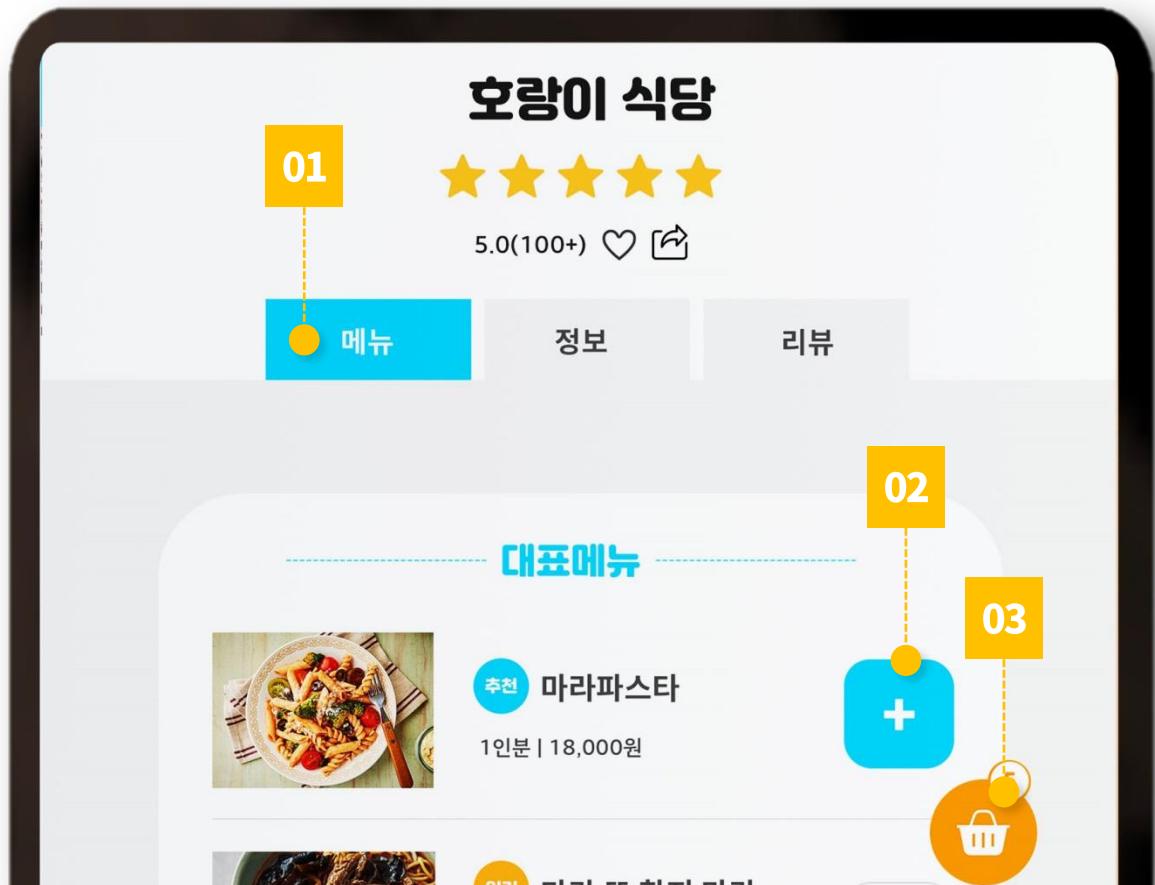
UI-C-1014, UI-C-1015

SCREEN PATH

메인화면 &gt; 로그인 &gt; 카테고리 선택 &gt; 매장선택 &gt; 메뉴선택

## Description

- 01 매장메뉴, 매장정보, 리뷰 선택
- 02 장바구니에 담기
- 03 장바구니로 이동



## Description

UI-E-1011

SCREEN PATH

메인화면 &gt; 로그인 &gt; 카테고리 선택 &gt; 매장선택 &gt; 메뉴선택 &gt; 장바구니 OR 메인화면 &gt; 장바구니

## Description

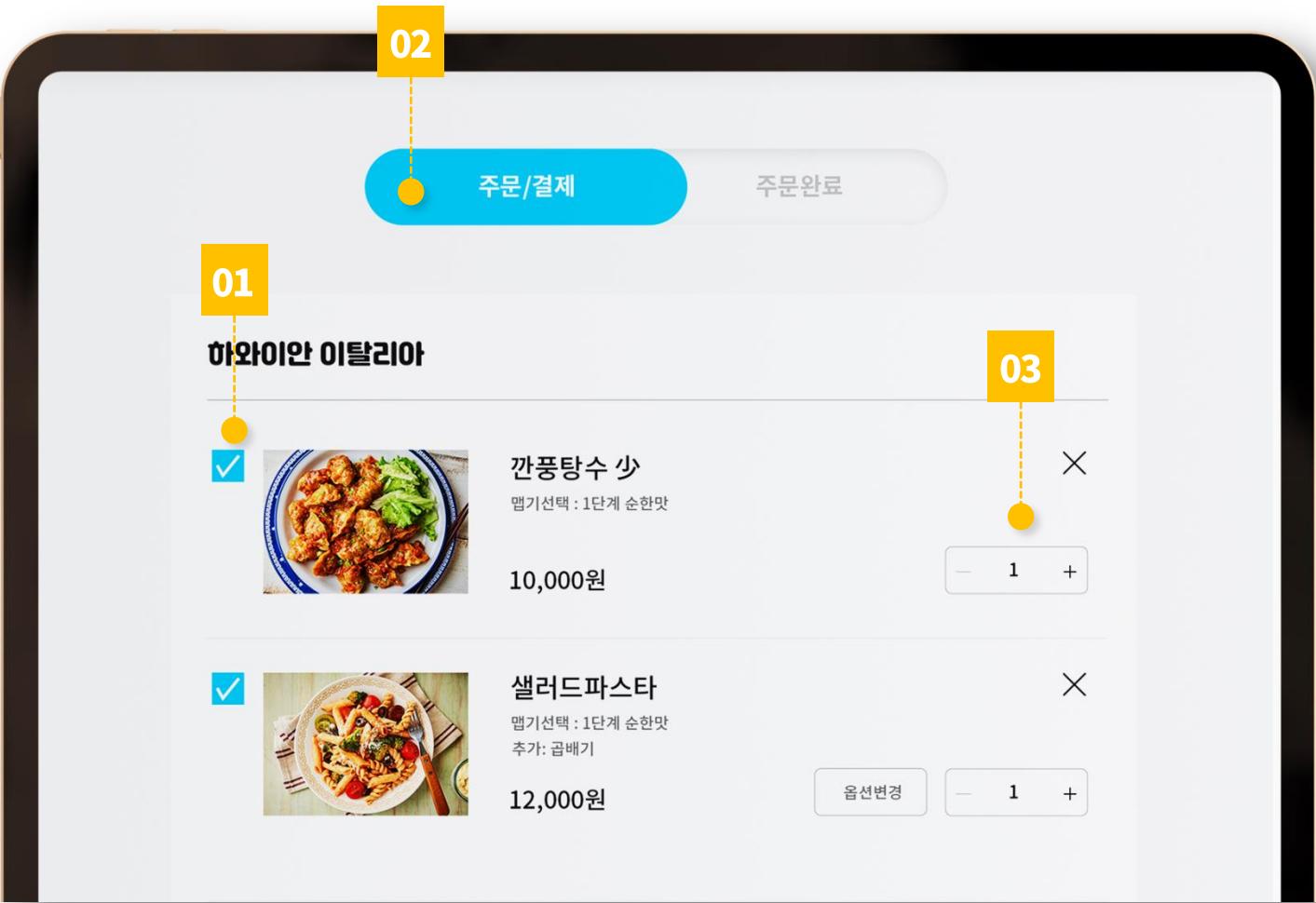
01 최종메뉴선택

02 주문완료 전 진행사항 표시

03 수량 및 옵션변경

## Check point

01 주문에서 장바구니에 담은 항목만 선택 가능



## Description

- 01 장바구니 항목 추가
- 02 최종 장바구니 합계 조회
- 03 결제하기 클릭 시 결제 페이지로 이동
- 04 결제 창



SCREEN PATH

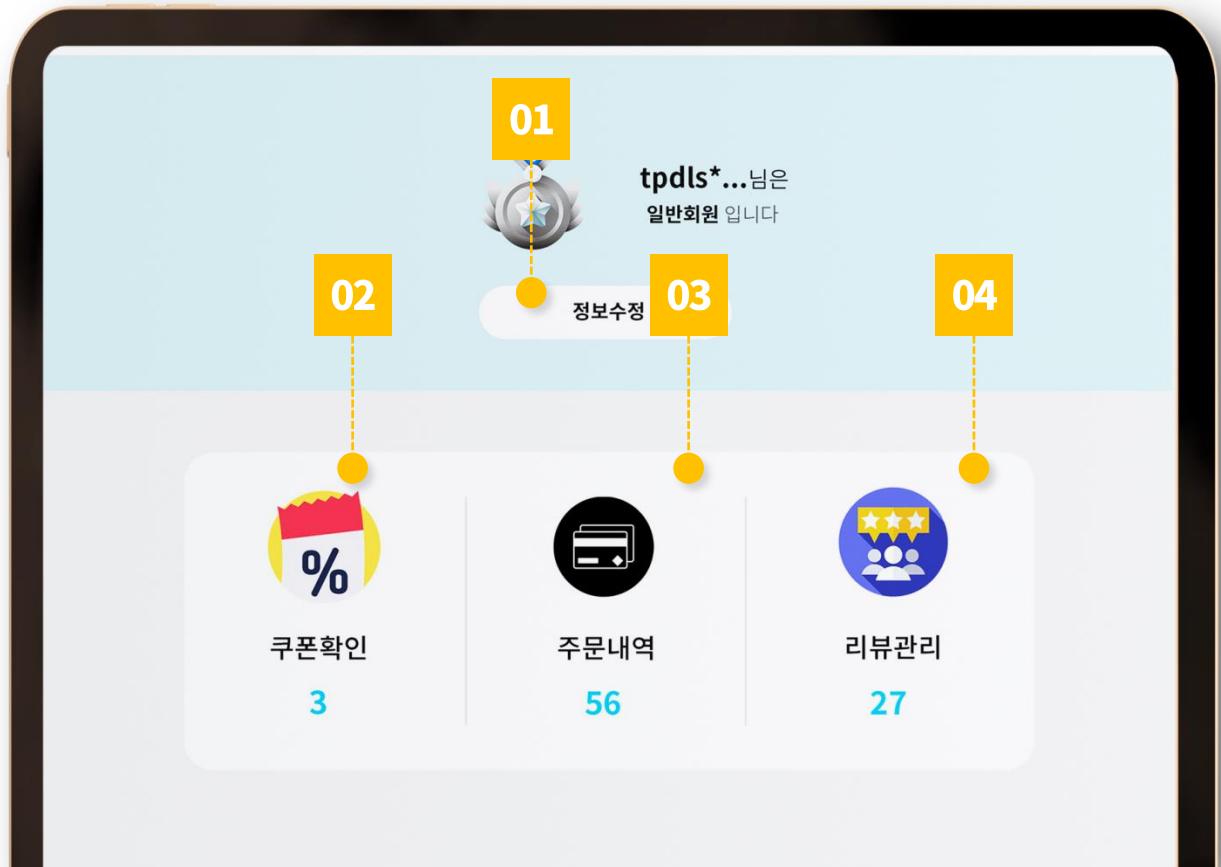
메인화면 &gt;마이페이지

## Description

- 01 회원 정보수정페이지 이동
- 02 쿠폰확인 페이지 이동
- 03 주문내역페이지 이동
- 04 리뷰관리페이지 이동

## Check point

- 01 회원만 마이페이지를 이용할 수 있다



SCREEN PATH

메인화면 &gt; 마이페이지 &gt; 회원정보수정

## Description

- 01 이메일, 비밀번호, 주소 등 회원정보 수정
- 02 인증 완료 시 수정하기 승인
- 03 수정완료 시 마이페이지로 이동
- 04 회원탈퇴완료 시 마이페이지 이동



## Check point

- 01 회원만 마이페이지를 이용할 수 있다

SCREEN PATH

메인화면 &gt;마이페이지 &gt; 주문내역조회

## Description

- 01 배송완료, 대기, 취소 정보
- 02 주소내역 내 주문내역 삭제
- 03 재주문 및 리뷰쓰기
- 04 배송지, 주문내역 정보



## Check point

- 01 마이페이지를 통해서만 주문내역을 볼 수 있다.

# LINK

대구요 : <http://15.164.139.112:8080/>

# 감사합니다

임베디드 소프트웨어 융합 풀스택 개발자 양성 과정B

2조 이재민, 백승하, 박민영, 정세인