

# 요구사항 명세서

## 1. 음식점 원격 대기(Wating)

- 기능 설명: 사용자가 원격으로 음식점에 대기 신청을 할 수 있다.
- 요구사항
  - 사용자는 음식점 목록에서 원격 대기를 신청할 수 있다.
  - 원격 대기 상태를 실시간으로 확인할 수 있다.
  - 대기 중 취소 기능이 제공된다.
  - 대기 순서가 임박하면 대기 준비 알림이 간다.
  - 입장 순서가 되면 입장 알림이 간다.

## 2. 사용자 위치 기반 대기/예약/포장 음식점 리스트

- 기능 설명: 사용자가 현재 위치를 기반으로 대기, 예약, 포장 가능한 음식점 목록을 조회할 수 있다.
- 요구사항
  - 사용자의 현재 위치를 기반으로 음식점 리스트를 제공한다.
  - 사용자는 필터를 통해 대기, 예약 가능한 음식점을 선택할 수 있다.
  - 음식점의 상세 정보를 조회할 수 있다.
  - 대기중인 사람의 수를 볼 수 있다.

## 3. 찜 목록(wishlist)

- 기능 설명: 사용자가 가고 싶은 음식점을 저장하여 목록을 조회할 수 있다.
- 요구사항
  - 사용자가 가게를 찜 목록에 추가할 수 있어야 한다.
  - 이미 찜한 가게를 중복해서 찜하지 않도록 해야 한다.
  - 사용자가 본인의 찜 목록을 조회할 수 있어야 한다.
  - 찜한 가게의 상세 정보 (가게 이름, 주소 등)를 함께 제공해야 합니다.
  - 유저가 목록의 특정 항목을 수정할 수 있어야 한다.
  - 사용자가 찜 목록에서 가게를 제거할 수 있어야 한다.

## 4. 예약(Reservation)

- 기능 설명: 사용자가 음식점을 예약할 수 있다.
- 요구사항
  - 사용자는 음식점 목록에서 예약을 신청할 수 있다.
  - 예약을 할때 인원 수 설정을 해야한다
  - 예약이 완료되면 카톡으로 예약확인 알림이 간다
  - 예약 중 취소 기능이 제공된다.
  - 취소는 예약시간 30분 전까지 가능하다
  - 예약시간 10분 경과시 자동취소된다.

## 6. 공지사항 알림

- 기능 설명: 사용자가 공지사항을 받을 수 있다.
- 요구사항
  - 시스템 공지사항을 사용자에게 알림으로 제공한다.

## 8. 유저 그룹

- 기능 설명: 사용자가 그룹을 생성하여 예약 상태를 공유할 수 있다.
- 요구사항
  - 그룹의 프로필(사진, 이름 등)을 설정할 수 있다.
  - 사용자는 그룹을 생성하고 다른 사용자들을 그룹에 초대할 수 있다.
  - 그룹 내 예약 상태를 공유할 수 있다.

## 9. 음식점 리뷰

- 기능 설명: 사용자가 음식점에 리뷰를 남길 수 있다.
- 요구사항
  - 사용자는 음식점에 대한 리뷰를 작성하고 조회할 수 있다.
  - 리뷰에 사진을 첨부할 수 있다.

## 10. 포스팅 (Posting)

- 기능 설명: 특정 권한을 가진 사용자가 음식점을 추천하는 포스팅을 작성할 수 있다.
- 요구사항
  - 소수의 특정 권한을 가진 사용자만 포스팅을 작성할 수 있다.
  - 포스팅은 블로그 형식으로 작성된다.
  - 모든 사용자가 포스팅을 조회할 수 있다.

## 11. 회원가입

- 기능 설명 : 사용자 등록
- 요구사항
  - 신규 사용자는 이름, 닉네임, 전화번호, 이메일, 비밀번호를 입력해야한다.
  - 닉네임, 전화번호, 이메일은 중복이 불가능하다.
  - 간편 로그인(네이버, 카카오 등)이 가능하다

## 12. 로그인

- 기능 설명: 사용자가 고유한 이메일, 비밀번호를 입력하여 서비스를 이용할 수 있다.
- 요구사항
  - 시스템에 등록되어 있는 이메일, 비밀번호를 입력하여 로그인할 수 있다.
  - 소셜 로그인 방식 이용 시, 토큰을 받아와서 자동 로그인 기능을 사용할 수 있다.

# 개념적 설계

## 주요 엔터티

### 1. **User** (사용자)

- user\_id (PK)
- name
- user\_type
- nickname
- email
- phone
- password

### 2. **Store**(음식점)

- store\_id (PK)
- name
- address(do,si,gu,detail)
- phone
- category
- remote\_tabling
- onsite\_tabling
- ratings
- status
- review\_cnt

### 3. **Waiting** (대기)

- id (PK)
- user\_id (FK)
- store\_id (FK)
- group\_id (FK)
- status
- created\_at
- updated\_at

### 4. **Reservation** (예약)

- id (PK)
- user\_id (FK)
- store\_id (FK)
- group\_id(FK)
- status
- created\_time

**5. Notification (공지사항)**

- id (PK)
- title
- content
- created\_at

**6. User\_group (유저 )그룹**

- id (PK)
- user\_id(FK)
- group\_id(FK)

**7. group (그룹)**

- id (PK)
- name
- created\_time
- detail
- profile\_img

**8. Review (리뷰)**

- id (PK)
- user\_id (FK)
- store\_id(FK)
- title
- content
- rating
- helpful
- created\_at

**9. Wishlist (가고 싶은 음식점)**

- id (PK)
- user\_id (FK)
- store\_id(FK)
- created\_time
- updated\_time

**10. Posting (포스팅)**

- id (PK)
- user\_id (FK)

- store\_id(FK)
- title
- content
- photo\_url(posting\_image 테이블)
- created\_time
- updated\_time

#### 11. posing\_comment (댓글)

- id (PK)
- user\_id (FK)
- posting\_id (FK)
- title
- content
- created\_at

#### 12. Announcement (공지사항)

- announcement\_id (PK)
- title
- content
- created\_time

## ERD (Entity-Relationship Diagram)

### 사용자 (User)와의 관계

1. **\*\*User (사용자)\*\*와 Waiting (대기):**
  - User 하나는 여러 개의 Waiting을 신청할 수 있습니다 (1 관계).
  - Waiting은 반드시 하나의 User에게 속합니다.
2. **\*\*User (사용자)\*\*와 Reservation (예약):**
  - User 하나는 여러 개의 Reservation을 신청할 수 있습니다 (1 관계).
  - Reservation은 반드시 하나의 User에게 속합니다.
3. **\*\*User (사용자)\*\*와 Notification (공지사항):**
  - User 하나는 여러 개의 Notification을 받을 수 있습니다 (1 관계).
  - Notification은 반드시 하나의 User에게 속합니다.
4. **\*\*User (사용자)\*\*와 UserGroup (유저 그룹):**
  - User 하나는 여러 개의 UserGroup을 생성할 수 있습니다 (1 관계).
  - UserGroup은 반드시 하나의 User에게 속합니다.
5. **\*\*User (사용자)\*\*와 GroupMember (그룹 멤버):**

- User 하나는 여러 개의 GroupMember에 속할 수 있습니다 (1 관계).
- GroupMember는 반드시 하나의 User에게 속합니다.
- 6. **\*\*User (사용자)\*\*와 Review (리뷰):**
  - User 하나는 여러 개의 Review를 작성할 수 있습니다 (1 관계).
  - Review는 반드시 하나의 User에게 속합니다.
- 7. **\*\*User (사용자)\*\*와 Wishlist (가고 싶은 음식점):**
  - User 하나는 여러 개의 Wishlist를 가질 수 있습니다 (1 관계).
  - Wishlist는 반드시 하나의 User에게 속합니다.
- 8. **\*\*User (사용자)\*\*와 Posting (포스팅):**
  - User 하나는 여러 개의 Posting을 작성할 수 있습니다 (1 관계).
  - Posting은 반드시 하나의 User에게 속합니다.
- 9. **\*\*User (사용자)\*\*와 posing\_comment (댓글):**
  - User 하나는 여러 개의 Comment를 작성할 수 있습니다 (1 관계).
  - Comment는 반드시 하나의 User에게 속합니다.

## 음식점 (store)와의 관계

1. **\*\*store (음식점)\*\*와 Waiting (대기):**
  - store 하나는 여러 개의 Waiting을 받을 수 있습니다 (1 관계).
  - Waiting은 반드시 하나의 store에 속합니다.
2. **\*\*store (음식점)\*\*와 Reservation (예약):**
  - store 하나는 여러 개의 Reservation을 받을 수 있습니다 (1 관계).
  - Reservation은 반드시 하나의 store에 속합니다.
  - Takeout은 반드시 하나의 store에 속합니다.
3. **\*\*store (음식점)\*\*와 Review (리뷰):**
  - store 하나는 여러 개의 Review를 받을 수 있습니다 (1 관계).
  - Review는 반드시 하나의 store에 속합니다.
4. **\*\*store (음식점)\*\*와 Wishlist (가고 싶은 음식점):**
  - store 하나는 여러 개의 Wishlist에 포함될 수 있습니다 (1 관계).
  - Wishlist는 반드시 하나의 store에 속합니다.

## 대기 (Waiting)와의 관계

1. **\*\*Waiting (대기)\*\***는 반드시 하나의 User와 하나의 store에 속합니다.

## 예약 (Reservation)와의 관계

1. **\*\*Reservation (예약)\*\***는 반드시 하나의 **User**와 하나의 **store**에 속합니다.

## 공지사항 (**Notification**)와의 관계

1. **\*\*Notification (공지사항)\*\***는 반드시 하나의 **User**에게 속합니다. - 하나의 유저에 속하지 않는거 같아요

## 유저 그룹 (**UserGroup**)와의 관계

1. **\*\*UserGroup (유저 그룹)\*\***는 반드시 하나의 **User**에게 속합니다.

## 그룹 멤버 (**GroupMember**)와의 관계

1. **\*\*GroupMember (그룹 멤버)\*\***는 반드시 하나의 **User**와 하나의 **UserGroup**에 속합니다.

## 리뷰 (**Review**)와의 관계

1. **\*\*Review (리뷰)\*\***는 반드시 하나의 **User**와 하나의 **store**에 속합니다.

## 가고 싶은 음식점 (**Wishlist**)와의 관계

1. **\*\*Wishlist (가고 싶은 음식점)\*\***는 반드시 하나의 **User**와 하나의 **store**에 속합니다.

## 포스팅 (**Posting**)

### 와의 관계

1. **\*\*Posting (포스팅)\*\***는 반드시 하나의 **User**에게 속합니다.

## 댓글 (**posting\_comment**)와의 관계

1. **\*\*Comment (댓글)\*\***는 반드시 하나의 **User**와 하나의 **Posting**에 속합니다.

## 사용 시나리오

### 시나리오 1: 음식점 원격 대기

1. 사용자가 앱에 로그인합니다.
2. 현재 위치 또는 원하는 지역을 설정해 음식점 목록을 조회합니다.
3. 음식점을 선택합니다.
4. 원격 대기를 신청합니다.
5. 대기 상태를 실시간으로 확인합니다.
6. 만약 원격 대기를 취소하고 싶다면 취소 버튼을 누릅니다.

### 시나리오 2: 음식점 예약

1. 사용자가 앱에 로그인합니다.
2. 현재 위치를 기반으로 음식점 목록을 조회합니다.
3. 예약을 원하는 음식점을 선택합니다.
4. 예약을 신청합니다.
5. 예약 상태를 실시간으로 확인합니다.
6. 예약 중 취소 기능을 사용할 수 있습니다.
7. 예약 시간이 다가오면 카톡 알림을 받습니다.

### 시나리오 3: 사용자 위치 기반 대기/예약 음식점 리스트 조회

1. 사용자가 앱에 로그인합니다.
2. 현재 위치를 기반으로 음식점 목록을 조회합니다.
3. 대기, 예약 가능한 음식점을 필터링하여 조회합니다.
4. 음식점의 상세 정보를 확인합니다.

### 시나리오 4: 사용자별 가고 싶은 음식점 저장 및 조회

1. 사용자가 앱에 로그인합니다.
2. 가고 싶은 음식점 목록을 조회합니다.
3. 새로운 음식점을 목록에 추가하거나 기존의 목록에서 삭제합니다.

### 시나리오 5: 공지사항 알림

1. 사용자가 앱에 로그인합니다.
2. 시스템에서 공지사항이 발생하면 사용자에게 알림을 보냅니다.
3. 사용자는 앱에서 공지사항을 조회합니다.

### 시나리오 6: 유저 그룹 생성 및 예약 상태 공유

1. 사용자가 앱에 로그인합니다.
2. 유저 그룹을 생성합니다.
3. 그룹에 다른 사용자들을 초대합니다.
4. 그룹 내에서 예약 상태를 공유합니다.

### 시나리오 7: 음식점 리뷰 작성 및 도움돼요

1. 사용자가 앱에 로그인합니다.
2. 방문한 음식점에 대해 리뷰를 작성합니다.
3. 다른 사용자의 리뷰를 조회하고 도움돼요를 누릅니다.

### 시나리오 8: 포스팅 작성 및 조회

1. 사용자가 앱에 로그인합니다.
2. 특정 권한을 가진 사용자는 포스팅을 작성합니다.
3. 모든 사용자가 포스팅을 조회합니다.
4. 포스팅에 댓글을 작성하거나 좋아요를 누릅니다.

### 시나리오 9: 레스토랑 관리자의 메뉴 관리



- 5. 1. 레스토랑 관리자는 앱에 로그인한다.
- 6. 2. 메뉴 관리 페이지에서 새로운 메뉴 아이템을 추가한다.
- 7. 3. 메뉴 아이템의 이름, 가격, 설명을 입력하고 저장한다.
- 8. 4. 시스템은 메뉴 아이템을 저장하고 사용자에게 알림을 보낸다.

## 시나리오 10: 사용자 조회

- 9. 1. 사용자 로그인 시 SNS 로그인 및 이메일
- 10. 2. 메뉴 관리 페이지에서 새로운 메뉴 아이템을 추가한다.
- 11. 3. 메뉴 아이템의 이름, 가격, 설명을 입력하고 저장한다.
- 12. 4. 시스템은 메뉴 아이템을 저장하고 사용자에게 알림을 보낸다.