설계서

프로젝트명	테이블고		
작성일	2023.12.10	버전	1.0
변경 이력	–		
팀장	김현수		
팀원	오승민, 이상현, 정동재		

목 차

1. 개요

○ 시스템 개요 ····· 1
○ 개발 범위 ····· 1
○ 제약 사항 ····· 1

2. 아키텍처 설계

○ 소프트웨어 아키텍처 ····· 2
○ 시스템 아키텍처 ····· 4
○ 아키텍처 요구사항 및 구현방안 ····· 6

3. 사용자 인터페이스 설계

○ 사용자 인터페이스 구성 ····· 14
○ 사용자 인터페이스 목록 ····· 15
○ 사용자 인터페이스 설계 ····· 16

4. 컴포넌트 설계

○ 컴포넌트 구조도 ····· 38
○ 컴포넌트 목록 ····· 39
○ 컴포넌트 설계 ····· 40
○ 인터페이스 설계 ····· 48

5. 클래스 설계

○ 클래스 목록 ····· 65
○ 클래스 설계 ····· 66

6. 인터페이스 설계

○ 인터페이스 목록 ····· 74
○ 인터페이스 설계 ····· 74

7. 엔티티 관계 모형 설계 - ER 다이어그램까지

○ 엔티티 관계도 (ERD) ····· 77
○ 엔티티 설계 ····· 79

표 목차

표 1 소프트웨어 아키텍처 - 클라이언트 사이드(Client-Side) ····· 2
표 2 소프트웨어 아키텍처 - 서버 사이드(Server-Side) ····· 3
표 3 소프트웨어 아키텍처 - 데이터베이스 사이드 (Database-Side) ····· 3
표 4 시스템 아키텍처 - 요소 설명 ····· 5
표 5 아키텍처 요구사항 및 구현방안 - 시스템 보안 강화 (ARC-001) ····· 6
표 6 아키텍처 요구사항 및 구현방안 - 시스템 성능 최적화 (ARC-002) ····· 6
표 7 아키텍처 요구사항 및 구현방안 - 장애 복구 및 내결함성 보장 (ARC-003) ····· 7
표 8 아키텍처 요구사항 및 구현방안 - 구축 시스템 하드웨어 환경 (ECR-001) ····· 7
표 9 아키텍처 요구사항 및 구현방안 - 온라인 도움말 (INR-001) ····· 8
표 10 아키텍처 요구사항 및 구현방안 - 통합 UI (INR-002) ····· 8
표 11 아키텍처 요구사항 및 구현방안 - 데이터 수집 (DAR-001) ····· 9
표 12 아키텍처 요구사항 및 구현방안 - 데이터 이동 및 전송 (DAR-002) ····· 9
표 13 아키텍처 요구사항 및 구현방안 - DB 설계 방안 구체화 (DAR-003) ····· 10
표 14 아키텍처 요구사항 및 구현방안 - 개인정보 보호 및 암호화 (SER-001) ····· 11
표 15 아키텍처 요구사항 및 구현방안 - 보안 취약 사항 관리 (SER-002) ····· 11
표 16 아키텍처 요구사항 및 구현방안 - 네트워크 보안 (SER-003) ····· 11
표 17 아키텍처 요구사항 및 구현방안 - 인증 및 권한 (SER-004) ····· 12
표 18 아키텍처 요구사항 및 구현방안 - 시스템 신뢰성 (QUR-001) ····· 12
표 19 아키텍처 요구사항 및 구현방안 - 장애 대응 (QUR-002) ····· 12
표 20 아키텍처 요구사항 및 구현방안 - 프로그램 학습성 (QUR-003) ····· 13
표 21 아키텍처 요구사항 및 구현방안 - 변경 관리 (QUR-004) ····· 13
표 22 사용자 인터페이스 구성 ····· 13
표 23 사용자 인터페이스 목록 ····· 14
표 24 사용자 인터페이스 설계 - 로그인 (UI-001) ····· 16
표 25 사용자 인터페이스 설계 - 회원가입 (UI-002) ····· 18
표 26 사용자 인터페이스 설계 - 메인 화면-손님용 SW (UI-003) ····· 22
표 27 사용자 인터페이스 설계 - 메인 화면-매장용 SW (UI-004) ····· 24
표 28 사용자 인터페이스 설계 - 매장 검색 (UI-005) ····· 26
표 29 사용자 인터페이스 설계 - 매장 정보 (UI-006) ····· 28
표 30 사용자 인터페이스 설계 - 줄서기 신청 (UI-007) ····· 30
표 31 사용자 인터페이스 설계 - 내 정보 (UI-008) ····· 32
표 32 사용자 인터페이스 설계 - 현장 줄서기 신청 (UI-009) ····· 34
표 33 사용자 인터페이스 설계 - 대기 명단 관리 (UI-0010) ····· 36
표 34 사용자 인터페이스 설계 - 내 매장 정보 (UI-011) ····· 38
표 35 컴포넌트 목록 ····· 39
표 36 컴포넌트 설계 - 사용자 관리 (CP-001) ····· 40
표 37 컴포넌트 설계 - 줄서기 관리 (CP-002) ····· 41

표 38 컴포넌트 설계 - 매장 정보 관리 (CP-003) ····· 42
표 39 컴포넌트 설계 - 알림 서비스 (CP-004) ····· 43
표 40 컴포넌트 설계 - 지도 및 위치 서비스 (CP-005) ····· 44
표 41 컴포넌트 설계 - 데이터베이스 (CP-006) ····· 45
표 42 컴포넌트 설계 - 결제 서비스 (CP-007) ····· 46
표 43 컴포넌트 설계 - 리뷰 서비스 (CP-008) ····· 47
표 44 인터페이스 설계 - UserManageable (IF-001) - registerUser ····· 48
표 45 인터페이스 설계 - UserManageable (IF-001) - loginUser ····· 48
표 46 인터페이스 설계 - UserManageable (IF-001) - logoutUser ····· 49
표 47 인터페이스 설계 - UserManageable (IF-001) - getUserInfo ····· 49
표 48 인터페이스 설계 - UserManageable (IF-001) - addUser ····· 50
표 49 인터페이스 설계 - ListManageable (IF-002) - removeUser ····· 50
표 50 인터페이스 설계 - ListManageable (IF-002) - getUserList ····· 50
표 51 인터페이스 설계 - ListManageable (IF-002) - changeSeq ····· 51
표 52 인터페이스 설계 - ListManageable (IF-002) - callUser ····· 51
표 53 인터페이스 설계 - StoreManageable (IF-003) - addStore ····· 52
표 54 인터페이스 설계 - StoreManageable (IF-003) - getStoreInfo ····· 52
표 55 인터페이스 설계 - StoreManageable (IF-003) - getStoreReview ····· 52
표 56 인터페이스 설계 - StoreManageable (IF-003) - updateStoreInfo ····· 53
표 57 인터페이스 설계 - NotificationService (IF-004) - sendKakaoNotification ····· 54
표 58 인터페이스 설계 - NotificationService (IF-004) - saveNotification ····· 54
표 59 인터페이스 설계 - MapManageable (IF-005) - showStoreLocationOnMap ····· 55
표 60 인터페이스 설계 - MapManageable (IF-005) - showUserLocationOnMap ····· 55
표 61 인터페이스 설계 - MapManageable (IF-005) - showNearbyStoresOnMap ····· 55
표 62 인터페이스 설계 - DataManageable (IF-006) - getData ····· 56
표 63 인터페이스 설계 - DataManageable (IF-006) - updateData ····· 56
표 64 인터페이스 설계 - DataManageable (IF-006) - updateData ····· 56
표 65 인터페이스 설계 - BackupManageable (IF-007) - makeBackup ····· 57
표 66 인터페이스 설계 - BackupManageable (IF-007) - restoreBackup ····· 57
표 67 인터페이스 설계 - PayManageable (IF-008) - processPayment ····· 58
표 68 인터페이스 설계 - PayManageable (IF-008) - getPaymentHistory ····· 58
표 69 인터페이스 설계 - CardManageable (IF-009) - addCard ····· 59
표 70 인터페이스 설계 - CardManageable (IF-009) - removeCard ····· 59
표 71 인터페이스 설계 - SecurityAuthenticator (IF-010) - checkAccessPermission ····· 60
표 72 인터페이스 설계 - SecurityAuthenticator (IF-010) - decryptData ····· 60
표 73 인터페이스 설계 - SecurityAuthenticator (IF-010) - encryptData ····· 60
표 74 인터페이스 설계 - PaymentGateway (IF-011) - sendPaymentRequest ····· 61
표 75 인터페이스 설계 - ReviewManageable (IF-012) - writeReview ····· 62
표 76 인터페이스 설계 - ReviewManageable (IF-012) - getReviews ····· 62
표 77 인터페이스 설계 - ReviewManageable (IF-012) - editReview ····· 63

표 78 인터페이스 설계 - ReviewManageable (IF-012) - deleteReview ····· 63
표 79 인터페이스 설계 - ReviewManageable (IF-012) - writeReply ····· 64
표 80 클래스 목록 ····· 65
표 81 클래스 설계 - UserManager (CL-001) ····· 66
표 82 클래스 설계 - UserInfo (CL-002) ····· 67
표 83 클래스 설계 - ListManager (CL-003) ····· 68
표 84 클래스 설계 - StoreManager (CL-004) ····· 68
표 85 클래스 설계 - StoreInfo (CL-005) ····· 69
표 86 클래스 설계 - NotificationManager (CL-006) ····· 69
표 87 클래스 설계 - MapManager (CL-007) ····· 70
표 88 클래스 설계 - DataManager (CL-008) ····· 70
표 89 클래스 설계 - BackupManager (CL-009) ····· 71
표 90 클래스 설계 - PayManager (CL-010) ····· 71
표 91 클래스 설계 - CardManager (CL-011) ····· 72
표 92 클래스 설계 - ReviewManager (CL-012) ····· 72
표 93 클래스 설계 - Review (CL-013) ····· 73
표 94 인터페이스 목록 ····· 74
표 95 인터페이스 설계 - 결제관리 (NI-001) ····· 74
표 96 인터페이스 설계 - 매장위치관리 (NI-002) ····· 75
표 97 인터페이스 설계 - 손님 관리 (NI-003) ····· 76
표 98 엔티티 관계도 - UserPayment (ERD-001) ····· 77
표 99 엔티티 관계도 - UserStoreReview (ERD-002) ····· 77
표 100 엔티티 관계도 - StoreWaiting (ERD-003) ····· 78
표 101 엔티티 설계 - User (EN-001) ····· 79
표 102 엔티티 설계 - Review (EN-002) ····· 79
표 103 엔티티 설계 - Store (EN-003) ····· 80
표 104 엔티티 설계 - WaitingInformation (EN-004) ····· 80
표 105 엔티티 설계 - Payment (EN-005) ····· 81

그림 목차

그림 1 소프트웨어 아키텍처 ····· 2
그림 2 시스템 아키텍처 ····· 4
그림 3 로그인 UI ····· 16
그림 4 로그인 순차 다이어그램 ····· 16
그림 5 회원가입 UI ····· 18
그림 6 회원가입 순차 다이어그램 ····· 18
그림 7 메인 화면 - 손님용 SW UI ····· 20
그림 8 메인 화면 - 손님용 SW 순차 다이어그램 ····· 20
그림 9 메인 화면 - 매장용 SW UI ····· 22
그림 10 메인 화면 - 매장용 SW 순차 다이어그램 ····· 22
그림 11 매장 검색 UI ····· 24
그림 12 매장 검색 순차 다이어그램 ····· 24
그림 13 매장 정보 UI ····· 26
그림 14 매장 정보 순차 다이어그램 ····· 26
그림 15 줄서기 신청 UI ····· 28
그림 16 줄서기 신청 순차 다이어그램 ····· 28
그림 17 내 정보 UI ····· 30
그림 18 내 정보 순차 다이어그램 ····· 30
그림 19 현장 줄서기 신청 UI ····· 32
그림 20 현장 줄서기 신청 순차 다이어그램 ····· 32
그림 21 대기 명단 관리 UI ····· 34
그림 22 대기 명단 관리 순차 다이어그램 ····· 34
그림 23 내 매장 정보 UI ····· 36
그림 24 내 매장 정보 순차 다이어그램 ····· 36
그림 25 컴포넌트 구조도 ····· 38
그림 26 컴포넌트 설계 - 사용자 관리 ····· 40
그림 27 컴포넌트 설계 - 줄서기 관리 ····· 41
그림 28 컴포넌트 설계 - 매장 정보 관리 ····· 42
그림 29 컴포넌트 설계 - 알림 서비스 ····· 43
그림 30 컴포넌트 설계 - 지도 및 위치 서비스 ····· 44
그림 31 컴포넌트 설계 - 데이터베이스 ····· 45
그림 32 컴포넌트 설계 - 결제 서비스 ····· 46
그림 33 컴포넌트 설계 - 리뷰 서비스 ····· 47
그림 34 엔티티 관계도 - UserPayment ····· 77
그림 35 엔티티 관계도 - UserStoreReview ····· 77
그림 36 엔티티 관계도 - StoreWaiting ····· 78

1. 개요

○ 시스템 개요

현재 우리가 다루고 있는 프로젝트는 '테이블고'라는 이름의 모바일 전용 웨이팅 애플리케이션이다. 이 애플리케이션은 매장 앞에서의 대기 과정을 효율화하고 손님들이 웨이팅을 원격으로 관리한다. 매장은 앱 내에서 매장 정보를 등록하고 웨이팅을 받을 수 있고, 손님들은 매장 정보를 확인하고 원격으로 웨이팅을 진행한다. 이로써 매장과 손님 모두가 웨이팅을 관리하고 이용하여 수익을 창출하는 것이 목표다.

○ 개발 범위

이 프로젝트의 개발 범위는 크게 매장 및 손님의 앱 개발과 연동, 웨이팅 관리 시스템의 구현이다. 매장 앱에서는 매장 정보를 등록하고 웨이팅을 관리하며, 손님 앱에서는 매장 정보를 확인하고 원격으로 웨이팅을 진행한다. 또한, 웨이팅이 완료되면 손님에게 알림을 보내고, 이후 매장으로의 입장 절차를 원활하게 이끈다.

○ 제약 사항

-대기시간 예측의 한계

정확한 대기시간 예측은 어렵기 때문에, 손님에게 예상 대기시간을 제공하되 정확성을 보장하기 위해 가게 별 평균 대기시간을 계산하여 예상시간을 제공한다.

-외부 서비스 의존

앱 개발 단계에서 네이버 맵 API, 카카오톡 API, 오픈 뱅킹 API 등 외부 서비스를 활용하는데, 이 서비스의 변경이나 중단에 대비하는 방안을 마련하고 유연한 대응을 위한 계획이 필요하다.

2. 아키텍처 설계

○ 소프트웨어 아키텍처

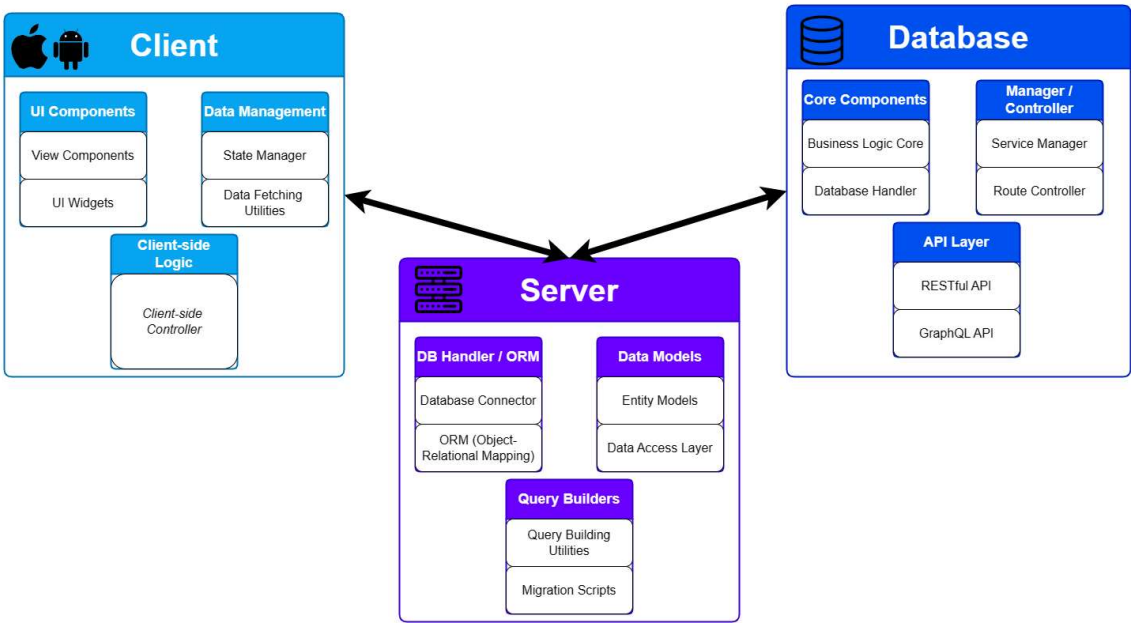

그림 1 소프트웨어 아키텍처

ㅇ 클라이언트 사이드(Client-Side)

View Components	React Native를 사용하여 메인 화면, 매장 상세 정보, 대기 목록 표시.
UI Widgets	React Navigation을 활용한 네비게이션 바, 버튼, 입력 필드.
State Manager	Redux 또는 Context API를 활용하여 앱의 상태 관리.
Data Fetching Utilities	Axios 또는 Fetch API를 사용하여 RESTful 및 GraphQL API와 통신.
Client-side Controller	매장 찜하기, 리뷰 작성 및 관리, 원격 예약 등 비즈니스 로직 처리.

표 1 소프트웨어 아키텍처 - 클라이언트 사이드(Client-Side)

○ 서버 사이드(Server-Side)

RESTful API	Express.js를 사용하여 클라이언트와의 HTTP 통신을 위한 엔드포인트 구축.
GraphQL API	Apollo Server를 활용하여 클라이언트의 유연한 데이터 요청에 응답.
Service Manager	Node.js 서버에서 비즈니스 로직 처리 및 데이터 관리를 위한 서비스.
Route Controller	Express Router를 사용하여 URL을 적절한 서비스로 라우팅.
Business Logic Core	주문 및 예약 관리, 매장 정보 업데이트, 대기 목록 처리 등의 핵심 비즈니스 로직.
Database Handler	Sequelize나 Mongoose를 사용하여 데이터베이스와의 상호 작용.

표 2 소프트웨어 아키텍처 - 서버 사이드(Server-Side)

○ 데이터베이스 사이드(Database-Side)

Database Connector	PostgreSQL, MongoDB와의 연결을 담당하는 모듈.
ORM	Sequelize ORM, Mongoose를 통해 데이터베이스 테이블과의 객체 매핑.
Entity Models	Sequelize 모델 또는 Mongoose 스키마를 통한 데이터 모델 정의.
Data Access Layer	데이터 액세스 및 조작을 위한 쿼리 작성 및 실행.
Query Building Utilities	SQL 쿼리 빌더 (Sequelize Query Interface) 또는 MongoDB 쿼리 작성.
Migration Scripts	Sequelize Migration 또는 MongoDB Migration을 통한 데이터베이스 스키마 관리.

표 3 소프트웨어 아키텍처 - 데이터베이스 사이드 (Database-Side)

○ 시스템 아키텍처

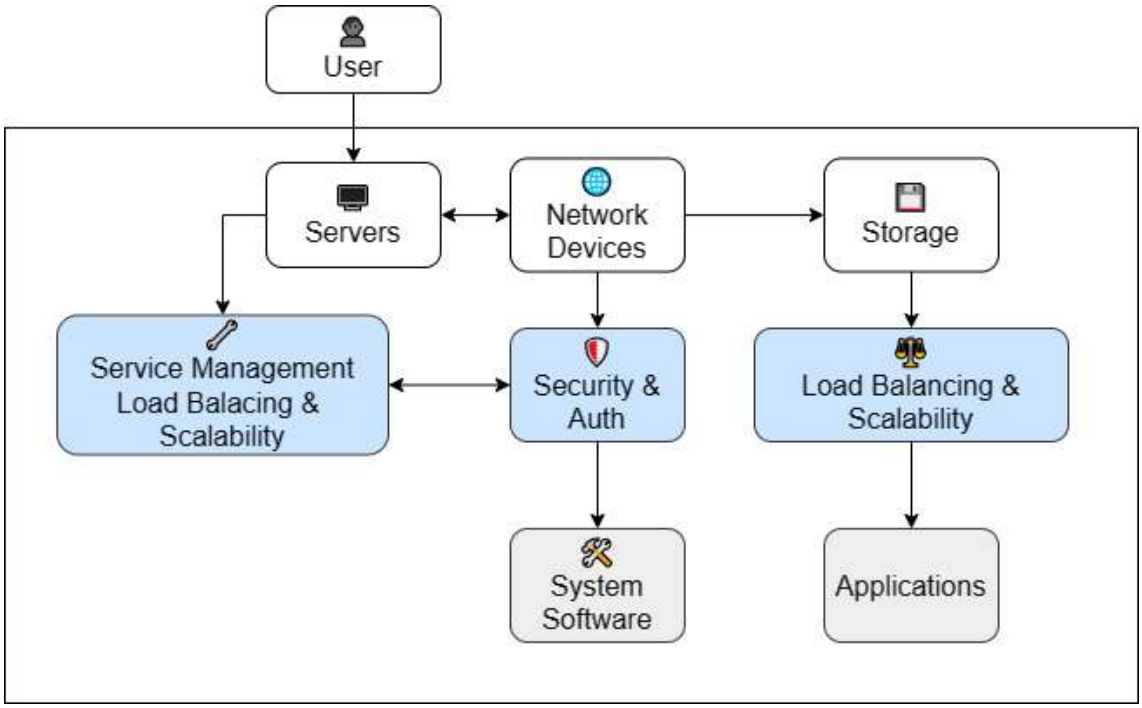

그림 2 시스템 아키텍처

○ 이 시스템 아키텍처는 전반적으로 사용자, 서버, 네트워크, 스토리지, 그리고 보안 및 관리 요소들로 구성된다. 여러 하위 시스템으로 분할되어 각 하위 시스템은 특정 기능 또는 역할을 담당한다.

사용자(User)	시스템의 외부에서 서비스를 이용하는 주체로, 서비스에 접속하여 기능을 활용.
서버(Servers)	주요 데이터 및 서비스를 호스팅하는 기기들로, 고성능 프로세서와 메모리, 저장장치 등을 갖추고 서비스 관리와 데이터 처리를 담당.
네트워크 디바이스(Network Devices)	네트워크 인프라를 구성하는 장비들로, 안정적인 네트워크 환경을 유지하고 데이터 전송을 관리.
저장장치(Storage)	데이터를 저장하고 관리하는 장치로, 대용량 데이터를 보관하고 필요한 정보에 빠르게 접근 가능.

서비스 관리(Service Management) / 로드 밸런싱 & 확장성(Load Balancing & Scalability)	서버 자원을 효율적으로 분배하고 서비스의 확장성을 유지합니다. 사용자 요청을 적절한 서버로 분산하고, 서버 자원을 증감하여 부하를 분산하거나 대규모 트래픽을 처리.
보안 및 인증(Security & Authentication)	시스템 보안을 유지하고 데이터 및 접속에 대한 인증을 담당합니다. 방화벽, 암호화, 사용자 인증 등을 통해 시스템을 보호.
시스템 소프트웨어(System Software)	하드웨어와 응용 프로그램 간 상호작용을 관리하고 운영체제, 데이터베이스 관리 시스템 등을 포함.
응용 프로그램(Applications)	사용자가 실제로 상호작용하는 소프트웨어로, 서비스를 제공하고 사용자 요청을 처리.

표 4 시스템 아키텍처 - 요소 설명

○ 아키텍처 요구사항 및 구현방안

요구사항 ID	ARC-001	요구사항 명	시스템 보안 강화
요구사항 내용			
시스템의 안전성을 강화하고 외부 침입을 방지하기 위해 데이터베이스 접근 및 사용자 인증 등 중요한 기능에 대한 보안 강화가 필요하다. 시스템 보안은 시스템 전체적인 안정성과 신뢰성에 큰 영향을 미치며, 사용자 데이터 및 인증 정보가 안전하게 보호되어야 한다. 이는 개인정보 보호법 및 관련 법규 준수를 포함한다.			
구현 방안			
-데이터베이스 암호화 민감한 사용자 정보는 AES 또는 SHA 알고리즘 등을 활용하여 데이터베이스에 저장할 때 암호화하여 저장한다. -접근 제어 및 권한 관리 사용자별 접근 권한을 엄격히 관리하고, 필요한 경우 두 요소 인증(2FA)과 같은 추가적인 보안 요소를 도입하여 시스템 접근을 제어한다. -보안 패치 및 업데이트 관리 서버 및 시스템 보안을 유지하기 위해 주기적으로 보안 패치 및 업데이트를 설치하고 관리한다. -보안 로그 및 감사 추적 시스템 내 모든 로그를 기록하고, 이를 정기적으로 감사하여 비인가된 접근이나 이상 활동을 탐지하고 대응한다.			

표 5 아키텍처 요구사항 및 구현방안 - 시스템 보안 강화 (ARC-001)

요구사항 ID	ARC-002	요구사항 명	시스템 성능 최적화
요구사항 내용			
시스템의 응답 시간과 처리량을 최적화하여 사용자 경험을 향상시키고, 동시 접속자 수 및 처리 능력을 향상시켜야 한다. 사용자들은 빠르고 안정적인 서비스를 기대하며, 응답 지연이나 서버 다운타임은 사용성과 신뢰성에 부정적인 영향을 미친다			
구현 방안			
-캐싱 및 미리 로딩 빈번하게 접근되는 데이터나 페이지는 캐싱하여 더 빠른 응답 시간을 제공하고, 필요한 리소스는 미리 로딩하여 지연을 최소화한다. -분산 시스템 구축 로드 밸런싱 및 분산 처리를 통해 동시 접속자 수와 처리 능력을 향상시킨다. 클라우드 기술을 활용하여 필요에 따라 리소스를 확장할 수 있도록 구성한다. -성능 테스트 및 모니터링 주기적인 성능 테스트를 수행하여 병목 현상을 식별하고, 모니터링 도구를 통해 실시간으로 시스템의 성능을 모니터링하고 대응한다.			

표 6 아키텍처 요구사항 및 구현방안 - 시스템 성능 최적화 (ARC-002)

요구사항 ID	ARC-003	요구사항 명	장애 복구 및 내 결함성 보장
요구사항 내용			
요구사항 내용: 시스템의 장애 발생 시 빠른 복구를 통해 서비스 중단을 최소화하고, 시스템의 내 결함성을 보장해야 한다. 장애 발생은 언제든 일어날 수 있으며, 이에 대비하여 서비스의 지속성을 보장하는 것이 중요하다.			
구현 방안			
-백업 및 복구 전략 데이터베이스의 정기적인 백업을 수행하고, 이를 안전한 곳에 저장하여 장애 시 빠른 복구를 가능케 한다. -가용성과 격리성 확보 마이크로서비스 아키텍처와 같은 격리된 서비스 구성을 통해 한 서비스의 장애가 전체 시스템에 영향을 미치지 않도록 한다. -모니터링 및 자동화된 대응 서비스의 상태를 지속적으로 모니터링하고, 이상 징후를 감지할 경우 자동화된 장애 대응 시스템을 동작시켜 서비스 중단을 최소화한다.			

표 7 아키텍처 요구사항 및 구현방안 - 장애 복구 및 내결함성 보장 (ARC-003)

요구사항 ID	ECR-001	요구사항 명	구축 시스템 하드웨어 환경
요구사항 내용			
서버 사양 고성능 프로세서, 64GB 이상 RAM, 1 Gbps 이상 네트워크, 10TB SSD 스토리지, 클라우드 서비스 도입 및 SSL/TLS 보안 강화 데이터베이스 서버: 높은 가용성과 용량, 속도의 스토리지 필요 네트워크 인프라: 안전하고 안정적인 고속 네트워크 환경 필요 보안 및 인증 장비: 방화벽, 침입 탐지 시스템(IDS), 암호화 등의 보안 장비, 사용자 인증 및 접근 제어 장비 필요			
구현 방안			
-고성능 서버 프로세서 및 RAM, 클라우드 서비스의 도입을 통해 확장성 확보한다. -데이터베이스의 높은 가용성을 위해 고속 및 안정적인 스토리지 구성한다. -보안 강화를 위해 SSL/TLS 암호화 및 방화벽, IDS 도입한다. -사용자 인증 및 접근 제어를 위한 강력한 보안 장비 구성한다.			

표 8 아키텍처 요구사항 및 구현방안 - 구축 시스템 하드웨어 환경 (ECR-001)

요구사항 ID	INR-001	요구사항 명	온라인 도움말
요구사항 내용			
온라인 도움말은 모든 사용자 기능에 대한 설명을 제공하고, 사용자가 발생한 오류에 대응하기 위한 에러 메시지와 각 에러 번호에 대한 해결 방법을 제공한다.			
구현 방안			
기능별 상세한 온라인 매뉴얼 제공: 각각의 기능에 대한 상세한 설명과 함께, 사용자가 기능을 어떻게 사용해야 하는지에 대한 가이드를 포함한 온라인 매뉴얼을 제공한다. 이는 사용자가 시스템의 각 부분을 보다 잘 이해하고 활용할 수 있도록 도와준다. 에러 메시지와 해결 방법 디스플레이: 오류가 발생한 경우에는 명확하고 사용자 친화적인 에러 메시지를 제공합니다. 각 에러 번호와 관련된 문제 해결 방법을 사용자에게 제시하여 사용자가 빠르게 문제를 해결할 수 있도록 돕는다.			

표 9 아키텍처 요구사항 및 구현방안 - 온라인 도움말 (INR-001)

요구사항 ID	INR-002	요구사항 명	통합 UI
요구사항 내용			
통합 UI 요구사항은 시스템 전반에 걸쳐 일관된 사용자 인터페이스(UI)를 제공하여 사용자 편의성과 직관성을 유지하며, 다양한 디바이스 및 환경에서의 UI 호환성을 보장하는 것을 목표로 한다.			
구현 방안			
-통일된 디자인 가이드라인 적용 시스템 전체적으로 일관성 있는 디자인 가이드라인을 정의하고, 이를 모든 UI 요소에 적용하여 일관성을 유지한다. -UI/UX 테스트 및 개선 사용자 경험(UI/UX)을 개선하기 위해 다양한 사용자 그룹으로부터 피드백을 수집하고, UI를 직관적이고 사용하기 편리하도록 지속적으로 테스트하고 개선한다. -다양한 디바이스 및 환경 호환성 테스트 다양한 환경(모바일, 데스크톱, 브라우저 등)에서 시스템을 테스트하여 UI 호환성을 보장하고 사용자가 어떤 환경에서도 일관된 경험을 할 수 있다 .			

표 10 아키텍처 요구사항 및 구현방안 - 통합 UI (INR-002)

요구사항 ID	DAR-001	요구사항 명	데이터 수집
요구사항 내용			
데이터 수집 요구사항은 사용자 맞춤 정보를 수집하여 어플리케이션 품질을 향상하고, 사용자 경험을 개선하기 위한 목적으로 데이터를 수집하는 것을 목표로 한다.			
구현 방안			
-사용자 디바이스 정보 수집 사용자의 디바이스 정보(기기 종류, 운영 체제, 해상도 등)를 수집하여 이를 통해 최적화된 서비스를 제공하고 사용자 환경을 이해한다. -사용자 피드백 및 오류/로그 수집 사용자의 피드백을 수집하여 어플리케이션의 개선 방향을 파악하고, 오류와 로그 데이터를 수집하여 시스템의 안정성을 향상시킨다. -익명화 및 동의서 도입 사용자의 개인정보를 보호하기 위해 수집된 데이터를 익명화하여 저장하고, 사용자 동의를 얻기 위한 동의서를 도입하여 데이터 수집에 대한 투명성을 제공한다.			

표 11 아키텍처 요구사항 및 구현방안 - 데이터 수집 (DAR-001)

요구사항 ID	DAR-002	요구사항 명	데이터 이동 및 전송
요구사항 내용			
DB 이전 및 데이터 관리를 위한 충돌과 중복을 방지하며, 보안 취약점을 점검하고 데이터 완결성을 유지한다.			
구현 방안			
-데이터 이전 시 충돌 및 중복 방지 DB 이전 프로세스에 충돌을 최소화하는 방법을 도입하여 데이터 중복과 충돌을 방지한다. -보안 취약점 점검 및 완결성 유지 데이터 이전 및 전송 시 보안 취약점을 점검하여, 데이터의 완결성을 유지하고 데이터 이동 과정에서의 보안을 강화한다.			

표 12 아키텍처 요구사항 및 구현방안 - 데이터 이동 및 전송 (DAR-002)

요구사항 ID	DAR-003	요구사항 명	DB 설계 방안 구체화
요구사항 내용			
DB 무결성과 정확성을 보장하며, 보안을 강화하여 데이터베이스를 안전하게 관리한다.			
구현 방안			
-데이터베이스 무결성 유지를 위한 제약 조건과 정규화 활용 데이터베이스의 무결성을 유지하기 위해 제약 조건과 정규화를 적용한다. -보안 강화 데이터 암호화와 접근 제어를 통해 데이터를 안전하게 보호하고, 데이터베이스의 보안을 강화한다			

표 13 아키텍처 요구사항 및 구현방안 - DB 설계 방안 구체화 (DAR-003)

요구사항 ID	SER-001	요구사항 명	개인정보 보호 및 암호화
요구사항 내용			
사용자 정보를 안전하게 보호하고, 개인정보를 적절히 암호화하여 관리한다.			
구현 방안			
-안전한 비밀번호 저장 (해시화) 사용자 비밀번호를 안전하게 저장하기 위해 해시화된 형태로 보관한다. -최소한의 개인정보 수집 및 동의서 도입 개인정보 수집을 최소화하고, 사용자의 동의를 받아 개인정보를 안전하게 관리한다.			

표 14 아키텍처 요구사항 및 구현방안 - 개인정보 보호 및 암호화 (SER-001)

요구사항 ID	SER-002	요구사항 명	보안 취약 사항 관리
요구사항 내용			
취약점을 관리하고 보안을 강화하여 시스템을 안전하게 관리한다.			
구현 방안			
-정기적인 취약점 점검과 업데이트 시스템에서 발생할 수 있는 취약점을 정기적으로 점검하고, 최신 보안 업데이트를 적용하여 보안성을 유지한다. -오픈 API 보안 관리 오픈 API의 보안을 강화하여 시스템에 대한 외부 접근을 안전하게 관리한다.			

표 15 아키텍처 요구사항 및 구현방안 - 보안 취약 사항 관리 (SER-002)

요구사항 ID	SER-003	요구사항 명	네트워크 보안
요구사항 내용			
데이터 보안과 무결성을 유지하고, 안전한 데이터 통신을 위해 네트워크를 보호한다			
구현 방안			
-안전한 데이터 통신을 위한 암호화 데이터 전송 시 안전한 통신을 위해 암호화 기술을 적용한다. -안정적인 세션 관리와 운영 체제/라이브러리 최신화 세션 관리를 통해 데이터 통신을 안전하게 유지하고, 운영 체제와 라이브러리를 최신 상태로 유지하여 보안을 강화한다.			

표 16 아키텍처 요구사항 및 구현방안 - 네트워크 보안 (SER-003)

요구사항 ID	SER-004	요구사항 명	인증 및 권한
요구사항 내용			
사용자 인증을 강화하고 이상 행위를 탐지하여 시스템을 안전하게 관리한다.			
구현 방안			
-강력한 인증 기능 도입 (비밀번호 정책, 멀티 팩터 인증) 사용자 인증을 강화하기 위해 안전한 비밀번호 정책과 멀티 팩터 인증을 도입한다. -이상 행위 감시 및 모니터링 시스템에서 이상 행위를 감지하고, 모니터링하여 빠르게 대응한다.			

표 17 아키텍처 요구사항 및 구현방안 - 인증 및 권한 (SER-004)

요구사항 ID	QUR-001	요구사항 명	시스템 신뢰성
요구사항 내용			
시스템 가용성과 복구성을 확보하여 안정적인 서비스를 제공한다.			
구현 방안			
-99.9% 이상의 가용성 보장 시스템의 가용성을 99.9% 이상으로 유지하여 시스템의 안정성을 보장한다. -백업 절차와 빠른 업데이트 기능 데이터의 백업 절차를 마련하고, 빠르고 정확한 업데이트 기능을 제공하여 시스템 복구를 지원한다.			

표 18 아키텍처 요구사항 및 구현방안 - 시스템 신뢰성 (QUR-001)

요구사항 ID	QUR-002	요구사항 명	장애 대응
요구사항 내용			
장애 발생 시 빠른 대응과 공지를 통해 사용자에게 신속한 정보를 제공한다.			
구현 방안			
-신속한 공지와 조치 체계 마련 장애 발생 시 신속하게 사용자에게 공지하고, 조치를 취할 수 있는 체계를 마련하여 빠른 대응을 지원한다. -자동화된 모니터링 시스템 구축 장애 발생을 실시간으로 감지하는 모니터링 시스템을 구축하여 사용자에게 사전 경고 및 알림을 제공하고, 신속한 대응이 가능하다.			

표 19 아키텍처 요구사항 및 구현방안 - 장애 대응 (QUR-002)

요구사항 ID	QUR-003	요구사항 명	프로그램 학습성
요구사항 내용			
사용자 학습성을 향상시켜 프로그램을 보다 쉽게 이해하고 활용할 수 있도록 지원한다.			
구현 방안			
-상세한 매뉴얼과 도움말 제공 기능에 대한 상세한 매뉴얼과 도움말을 제공하여 사용자가 프로그램을 쉽게 이해하고 활용할 수 있도록 지원한다. -인터랙티브한 학습 리소스 제공 온라인 튜토리얼, 비디오 자습자료, 또는 사용자가 프로그램 기능을 직접 시도하고 테스트할 수 있는 시뮬레이션 등을 제공하여 사용자의 학습성을 증진시킨다.			

표 20 아키텍처 요구사항 및 구현방안 - 프로그램 학습성 (QUR-003)

요구사항 ID	QUR-004	요구사항 명	변경 관리
요구사항 내용			
변경 요구를 관리하고 문서화하여 시스템의 변경을 체계적으로 관리한다.			
구현 방안			
-요구사항 관리와 테스트를 통한 변경사항 확인 변경 요구사항을 체계적으로 관리하고, 테스트를 통해 변경사항을 확인하여 시스템 변경을 관리한다. -변경사항의 우선순위 및 영향도 평가 변경사항을 우선순위에 따라 분류하고, 시스템에 미치는 영향을 평가하여 변경 관리를 더욱 체계적으로 진행한다. -변경 이력 및 문서화 강화 변경사항을 정확하게 문서화하고, 변경사항의 이력을 상세하게 기록하여 추후 문제 발생 시 신속한 대응이 가능하다.			

표 21 아키텍처 요구사항 및 구현방안 - 변경 관리 (QUR-004)

3. 사용자 인터페이스 설계

○ 사용자 인터페이스 구성

Level 1	Level 2	Level 3	Level 4
손님용 SW			
	사용자 시스템		
		로그인 시스템	
			로그인
			회원가입
		사용자정보 시스템	
			내 정보
	매장 시스템		
		매장 정보 시스템	
			매장 검색
			매장 정보
		줄서기 시스템	
			줄서기 신청
			매직 패스 구매
	메인 시스템		
		메인 시스템	
			메인 화면
매장용 SW			
	사용자 시스템		
		로그인 시스템	
			로그인
			회원가입
	매장 시스템		
		매장 정보 시스템	
			내 매장 정보
			매장 정보 수정
			영업 상태 변경
		줄서기 시스템	
			대기 명단 관리
			현장 줄서기 신청
	메인 시스템		
		메인 시스템	
			메인 화면

표 22 사용자 인터페이스 구성

○ 사용자 인터페이스 목록

UI 고유번호	UI 명	관련 유스케이스 ID	SW
UI-001	로그인	USC-001, USC-013	공통
UI-002	회원가입	USC-002, USC-004	공통
UI-003	메인 화면-손님용 SW	-	손님
UI-004	메인 화면-매장용 SW	-	매장
UI-005	매장 검색	USC-005	손님
UI-006	매장 정보	USC-003,USC-006, USC-008,USC-009	손님
UI-007	줄서기 신청	USC-003	손님
UI-008	내 정보 화면	USC-006,USC-007, USC-010~USC-012	손님
UI-009	현장 줄서기 신청	USC-014	매장
UI-010	대기 명단 관리	USC-015~USC-017	매장
UI-011	내 매장 정보	USC-018~USC-023	매장

표 23 사용자 인터페이스 목록

○ 사용자 인터페이스 설계

UI 고유번호	UI-001	UI 명	로그인
관련 유스케이스 ID	USC-001, USC-013		
UI 유형	입력	메뉴경로	로그인
UI 개요	로그인 처리 화면		

그림 3 로그인 UI

처리 내용

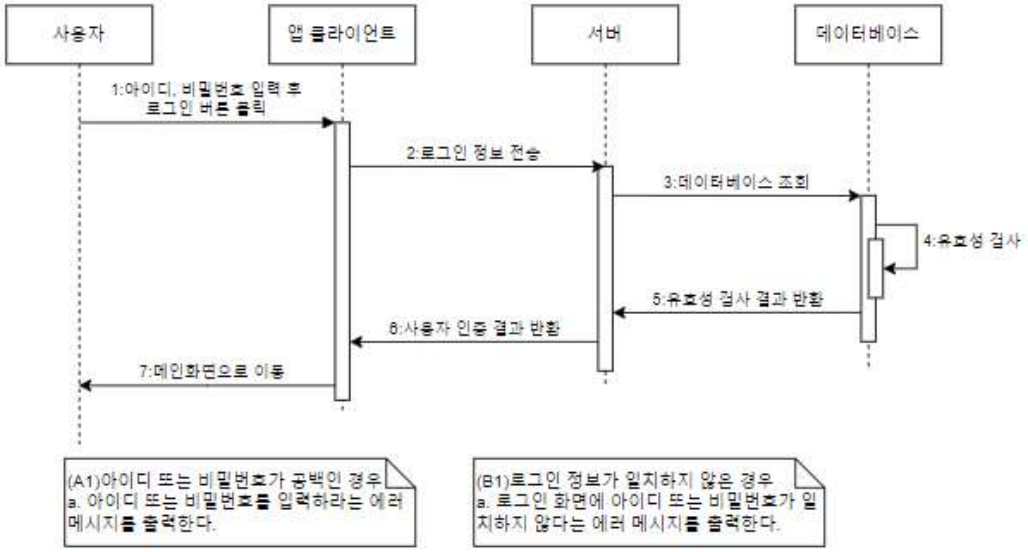

그림 4 로그인 순차 다이어그램

– 화면 시작

· 아이디와 비밀번호를 입력한다.

– 로그인

· 계정이 있는 사용자의 경우 로그인 버튼을 클릭하여 아이디와 비밀번호를 확인받고, 메인 화면으로 이동한다. (아이디 또는 비밀번호 미입력 시 에러 메시지 출력, 로그인 정보 불일치 시 에러 메시지 출력)

– 회원가입 화면 이동

· 계정이 없는 사용자의 경우 회원가입 버튼을 클릭하여 회원가입 화면으로 이동한다.

– 카카오톡으로 가입 및 로그인

· 카카오톡으로 시작하기 버튼을 클릭하여 카카오톡 계정 연동으로 가입 및 로그인 화면으로 이동한다.

입출력 항목			
항목명	타입/길이	속성	비고
아이디(이메일)	String/ 1~254	입력	필수
비밀번호	String/ 8~64	입력	필수
카카오톡 연동	Button	입력	선택
로그인	Button	입력	필수
회원가입	Button	입력	선택
고려 사항			

표 24 사용자 인터페이스 설계 - 로그인 (UI-001)

UI 고유번호	UI-002	UI 명	회원가입
관련 유스케이스 ID	USC-002, USC-004		
UI 유형	입출력	메뉴경로	로그인/회원가입
UI 개요	회원가입 화면		

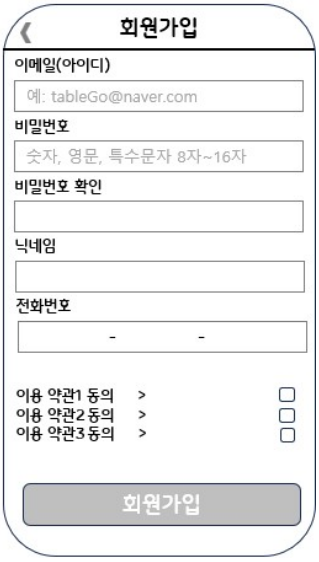

그림 5 회원가입 UI

처리 내용

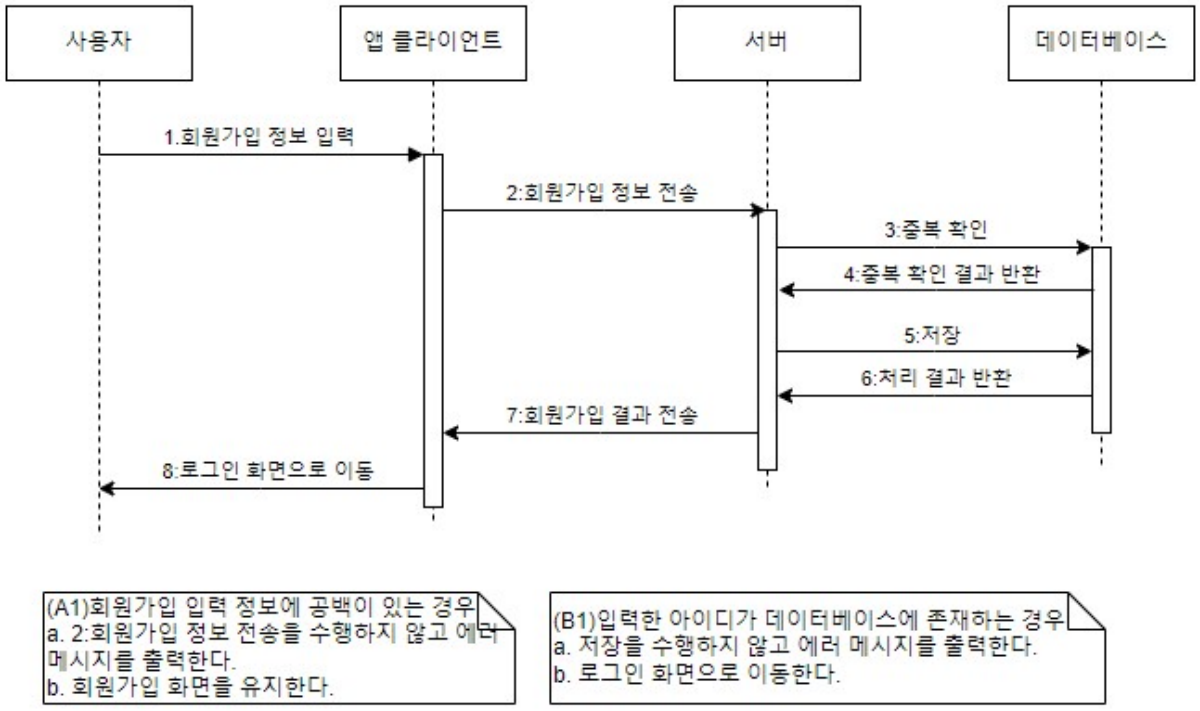

그림 6 회원가입 순차 다이어그램

- 화면 시작
  · 회원가입에 필요한 필수 정보들을 입력하고 이용 약관 동의 칸에 체크를 한다.
- 회원가입
  · 이메일이 데이터베이스에 존재하지 않고, 다른 필수 입력 정보에 문제가 없다면 데이터베이스에 회원 정보를 저장하고, 로그인 화면으로 이동한다. (문제 발생 시, 관련 에러 출력 후 회원가입 화면 유지)
- 뒤로가기
  · 입력 정보에 대한 저장 없이 로그인 화면으로 이동한다.

입출력 항목			
항목명	타입/길이	속성	비고
이메일	String/ 1~254	입력	필수
비밀번호	String/ 8~64	입력	필수
비밀번호 확인	String/ 8~64	입력	필수
닉네임	String/ 3~15	입력	필수
전화번호	String/ 11	입력	필수
이용약관	String	출력	-
이용약관 동의	Bool	입력	필수
회원가입	Button	입력	필수
고려 사항			

표 25 사용자 인터페이스 설계 - 회원가입 (UI-002)

UI 고유번호	UI-003	UI 명	메인 화면-손님용 SW
관련 유스케이스 ID	-		
UI 유형	입력	메뉴경로	메인 화면
UI 개요	손님용 SW의 메인 화면		

그림 7 메인 화면 - 손님용 SW UI

처리 내용

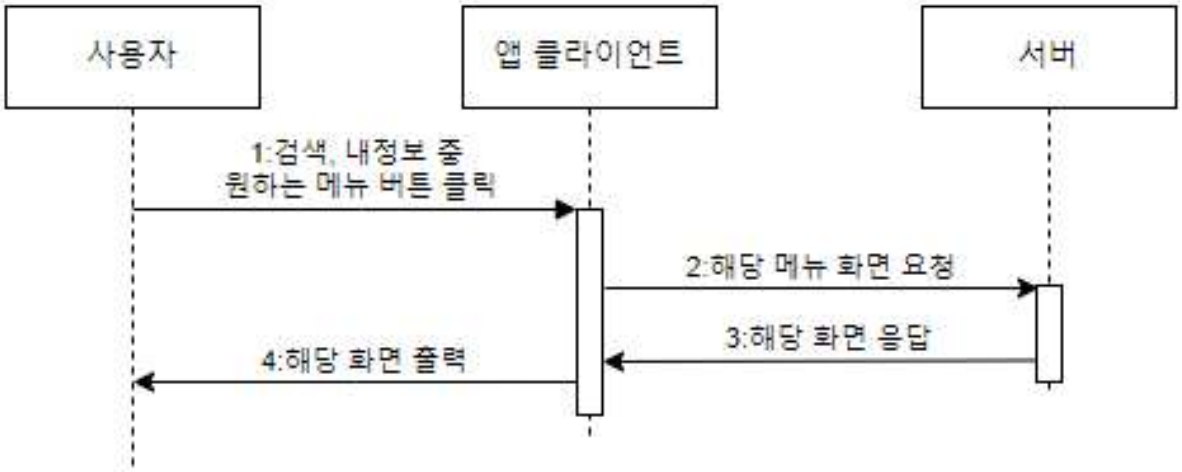

그림 8 메인 화면 - 손님용 SW 순차 다이어그램

- 화면 시작
  - 하단 바에 선택할 수 있는 메뉴 버튼이 존재하고, 이를 선택할 때 해당 화면을 출력할 중간에 큰 화면이 있다.
- 검색
  - 데이터베이스에 등록된 매장들을 검색할 수 있는 화면을 출력한다.

– 내 정보 · 앱에 로그인된 계정에 대한 정보 화면을 출력한다.			
입출력 항목			
항목명	타입/길이	속성	비고
검색	Button	입력	선택
내 정보	Button	입력	선택
고려 사항			

표 26 사용자 인터페이스 설계 - 메인 화면-손님용 SW (UI-003)

UI 고유번호	UI-004	UI 명	메인 화면-매장용 SW
관련 유스케이스 ID	-		
UI 유형	입력	메뉴경로	메인 화면
UI 개요	매장용 SW의 메인 화면		

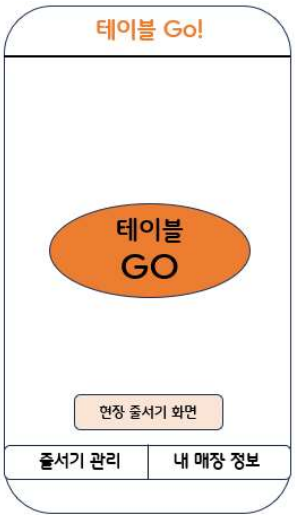

그림 9 메인 화면
- 매장용 SW UI

처리 내용

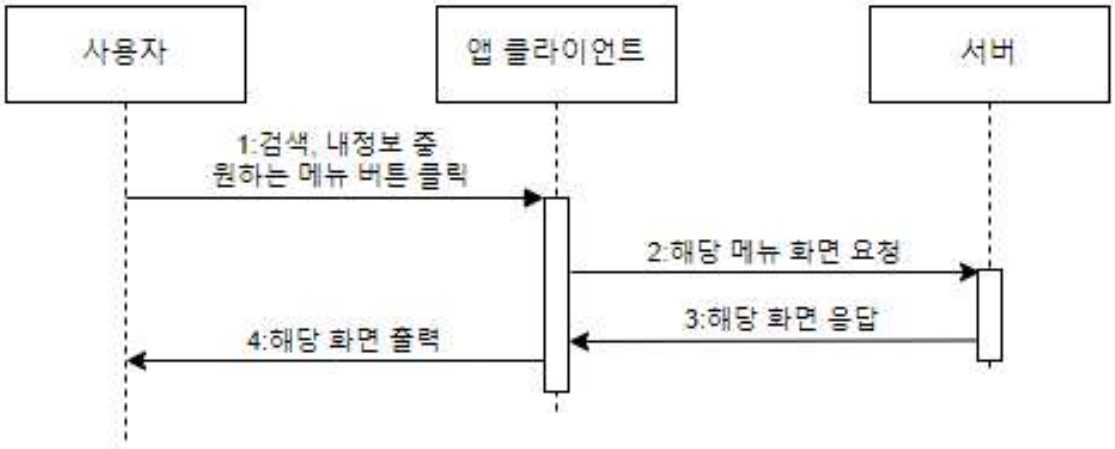

그림 10 메인 화면 - 매장용 SW 순차 다이어그램

- 화면 시작
  - 하단 바에 선택할 수 있는 메뉴 버튼이 존재하고, 이를 선택할 때 해당 화면을 출력할 중간에 큰 화면이 있다.
- 현장 줄서기 화면
  - 손님들이 매장 앞에서 줄서기 등록을 하기 위한 화면으로 이동한다.

– 줄서기 관리

· 현재 사용자의 매장에 줄서기 등록된 손님들을 관리하는 화면으로 이동한다.

– 내 매장 정보

· 로그인된 사용자의 계정에 등록된 매장에 대한 정보를 조회하고 수정하는 화면으로 이동한다.

입출력 항목			
항목명	타입/길이	속성	비고
현장 줄서기 화면	Button	입력	선택
줄서기 관리	Button	입력	선택
내 매장 정보	Button	입력	선택
고려 사항			

표 27 사용자 인터페이스 설계 - 메인 화면-매장용 SW (UI-004)

UI 고유번호	UI-005	UI 명	매장 검색
관련 유스케이스 ID	USC-005		
UI 유형	입출력	메뉴경로	메인 화면/매장 검색
UI 개요	매장 검색 화면		

그림 11 매장 검색 UI

처리 내용

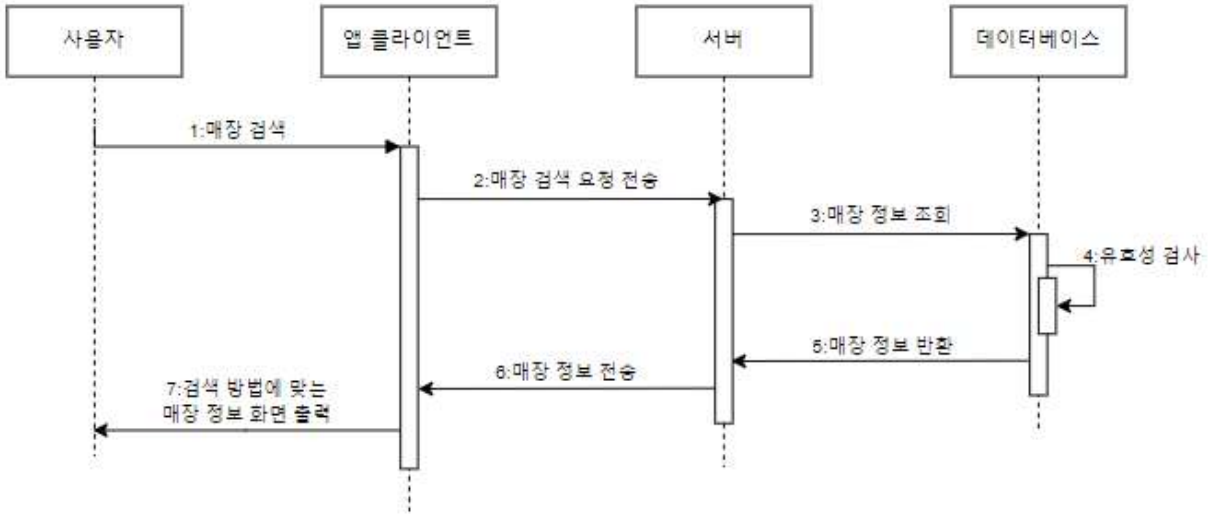

그림 12 매장 검색 순차 다이어그램

- 화면 시작
  - 지역명/매장명 버튼을 통해 어떤 조건으로 검색할 것인지를 선택하면 해당 조건에 맞는 화면이 중앙 화면에 출력되고, 검색어를 입력하여 매장 탐색한다.
- 지역명
  - 지역명 버튼 선택 시, 화면 중앙에 지도가 출력되고 지역명을 입력하면 해당 지역 반경 500m

이내에 테이블고에 등록된 매장이 지도에 추가된다.(영업 중인 매장은 빨간 테두리, 아니면 회색 테두리로 구분) 입력한 지역명이 반경보다 넓은 지역이라면 해당 지역의 정중앙을 기점으로 반경 500m 이내에 존재하는 매장만 지도에 추가된다. 매장 이미지를 클릭해서 매장 정보 화면으로 이동한다.

– 매장명

· 매장명 버튼 선택 시, 화면 중앙에 1열짜리 행렬 그리드가 출력되고 매장명 입력 시, 해당 매장명과 같고 테이블고에 등록된 매장이 그리드에 추가된다. (문자가 추가되거나 삭제될 때마다 업데이트된다.) 그리드에 추가된 매장을 클릭해서 매장 정보 화면으로 이동한다.

– 뒤로가기

· 메인 화면–손님용 SW로 이동한다.

입출력 항목			
항목명	타입/길이	속성	비고
지역명/매장명	Button	입력	필수
검색어	String/ 100	입력	필수
매장 리스트	List<Table>	입력/출력	–
지도	Image	출력	–

고려 사항

- 화면에 출력되는 지도는 네이버 지도 기반으로 그 위에 매장 이미지를 추가함으로써 구현한다.
- 리스트의 매장들이 화면 내에 담기지 않는 경우를 대비해 스크롤 형식으로 구현한다.
- Table 타입은 여러 타입이 혼합된 타입이다.

표 28 사용자 인터페이스 설계 - 매장 검색 (UI-005)

UI 고유번호	UI-006	UI 명	매장 정보
관련 유스케이스 ID	USC-003, USC-006, USC-008, USC-009		
UI 유형	입출력	메뉴경로	메인 화면/매장 검색/매장 정보
UI 개요	매장 정보를 보여주는 화면		

그림 13 매장 정보 UI

처리 내용

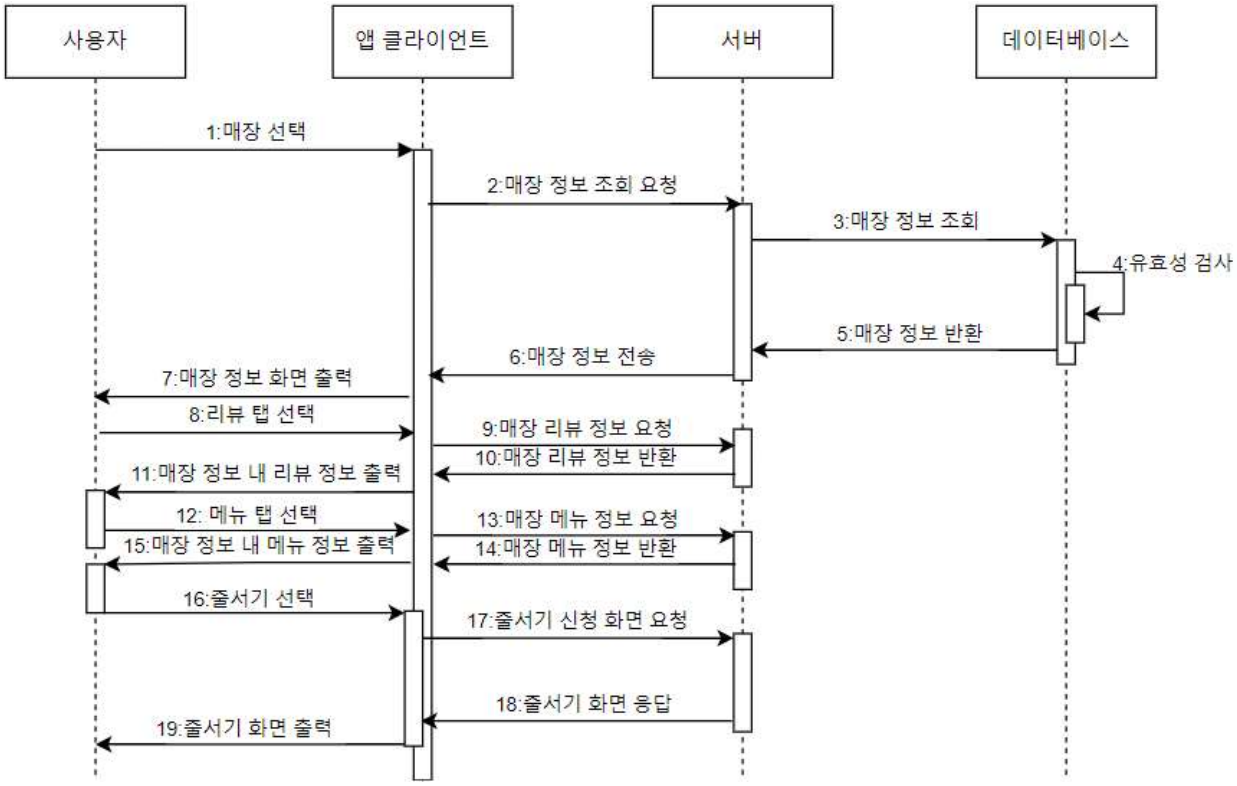

그림 14 매장 정보 순차 다이어그램

- 화면 시작
  · 매장 정보, 메뉴, 리뷰 탭을 선택하면 중앙 화면에 관련 정보가 출력된다.
- 매장 정보
  · 매장 정보 선택 시, 주소, 사업자, 전화번호 등의 매장 정보가 중앙 화면에 출력된다.
- 메뉴
  · 메뉴 선택 시, 모든 메뉴와 가격, 사진이 리스트 형태로 중앙 화면에 출력된다.
- 리뷰
  · 리뷰 선택 시, 모든 리뷰와 별점이 리스트 형태로 중앙 화면에 출력된다.
- 찜
  · 매장명 상단의 하트 클릭 시, 하트의 색깔이 붉은색으로 바뀌고 해당 매장이 사용자 계정의 찜 상태가 된다. 다시 클릭하여 찜 상태를 해제할 수 있다.
- 줄서기
  · 줄서기 버튼을 클릭 시, 해당 매장의 줄서기 신청 화면으로 이동한다.
- 뒤로가기
  · 뒤로가기 버튼 클릭 시, 메인 화면-손님용 SW로 이동한다.

입출력 항목			
항목명	타입/길이	속성	비고
매장 정보	Table/제한 없음	출력	-
메뉴	List<Table>/제한없음	출력	-
리뷰	List<Table>/제한없음	출력	-
찜	Button	입력	선택
줄서기	Button	입력	선택

고려 사항

표 29 사용자 인터페이스 설계 - 매장 정보 (UI-006)

UI 고유번호	UI-007	UI 명	줄서기 신청
관련 유스케이스 ID	USC-003		
UI 유형	입출력	메뉴경로	메인 화면/매장 검색/매장 정보/ 줄서기 신청
UI 개요	줄서기 신청과 매직 패스 신청 화면		

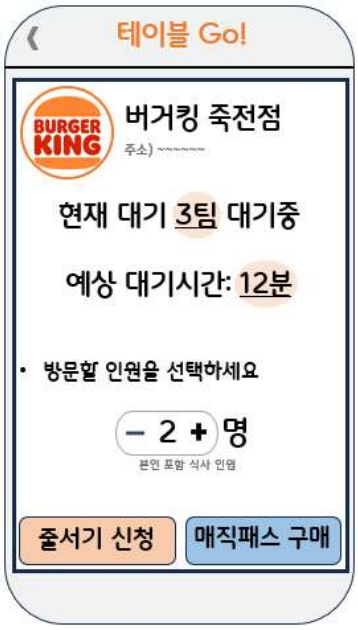

그림 15 줄서기 신청 UI

처리 내용

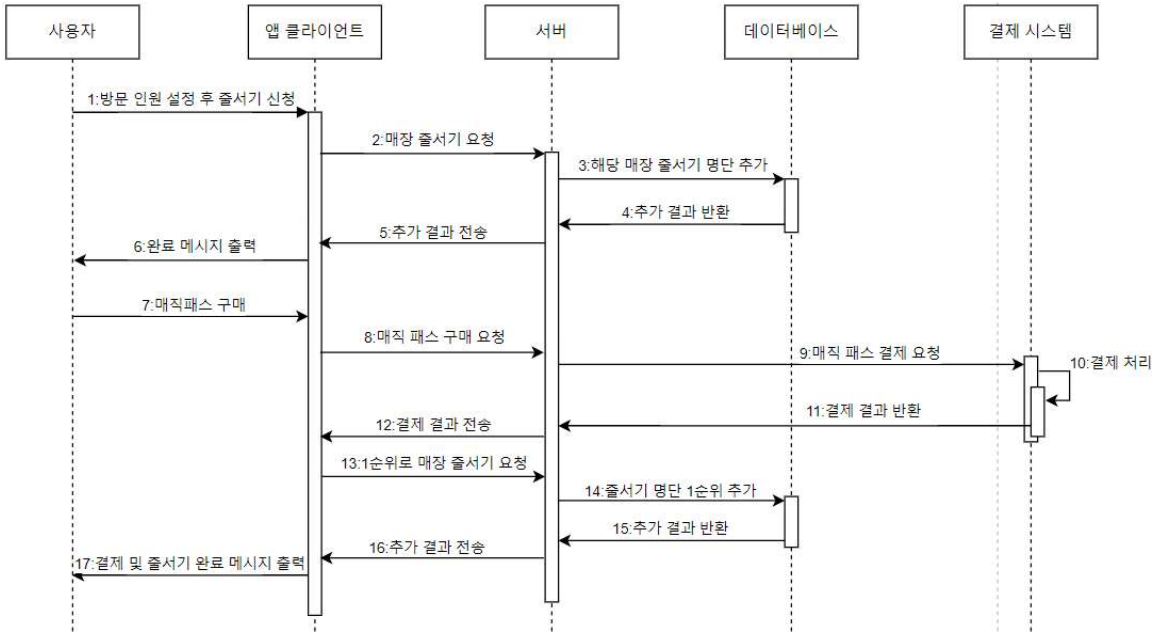

그림 16 줄서기 신청 순차 다이어그램

– 화면 시작
  · 매장 정보, 메뉴, 리뷰 탭을 선택하면 중앙 화면에 관련 정보가 출력된다.
– 방문 인원
  · 방문 인원을 +와 - 버튼으로 설정한다. +버튼을 누르면 증가하고 -버튼은 감소한다.
– 줄서기 신청
  · 설정한 인원으로 해당 매장의 줄서기 신청을 한다.
– 매직 패스 구매
  · 대기 번호 1번으로 줄서기 신청할 수 있는 매직 패스를 구매한다.
– 뒤로가기
  · 뒤로가기 버튼 클릭 시, 매장 정보 화면으로 이동한다.

입출력 항목			
항목명	타입/길이	속성	비고
매장 정보	Table/제한 없음	출력	–
대기 중 팀 수	int/제한 없음	출력	–
예상 대기시간	int/제한 없음	출력	–
인원수	Button/ 2	입력	필수
줄서기 신청	Button	입력	필수
매직 패스 구매	Button	입력	선택

고려 사항

- 해당 화면은 다른 화면 위에 팝업 형식으로 띄워져야 한다.
- 뒷배경의 다른 화면은 회색으로 처리한다.

표 30 사용자 인터페이스 설계 - 줄서기 신청 (UI-007)

UI 고유번호	UI-008	UI 명	내 정보
관련 유스케이스 ID	USC-006, USC-007, USC-010, USC-011, USC-012		
UI 유형	입출력	메뉴경로	메인 화면/ 내 정보
UI 개요	내 정보 화면		

그림 17 내 정보 UI

처리 내용

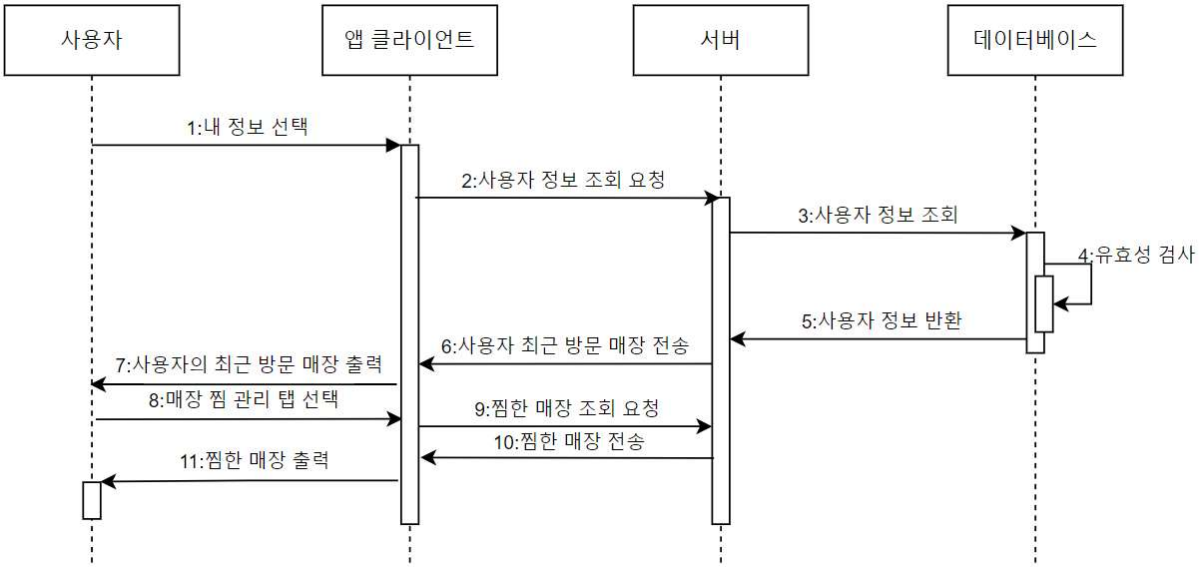

그림 18 내 정보 순차 다이어그램

- 화면 시작
  - 중앙 화면에 가장 최근에 방문한 매장 정보가 출력된다. 다른 탭을 선택하면 해당 탭에 대한 정보가 중앙 화면에 출력된다.
- 매장 찜 관리

· 매장 찜 관리 탭 선택 시, 사용자가 찜해놓은 매장들의 리스트를 중앙 화면에 출력한다.

– 줄서기 내역

· 줄서기 내역 탭 선택 시, 사용자의 이전 줄서기 내역 리스트를 중앙 화면에 출력한다.

– 리뷰 내역

· 리뷰 내역 탭 선택 시, 사용자가 작성했던 리뷰들 리스트를 중앙 화면에 출력한다.

– 결제 수단 관리

· 결제 수단 탭 선택 시, 저장된 사용자의 결제 수단을 중앙 화면에 출력한다.

– 뒤로가기

· 뒤로가기 버튼 클릭 시, 메인 화면–손님용 SW로 이동한다.

입출력 항목			
항목명	타입/길이	속성	비고
최근 방문 매장	Table/1	출력	–
매장 찜 관리	Button	입력	선택
줄서기 내역	Button	입력	선택
리뷰 내역	Button	입력	선택
결제수단 관리	Button	입력	선택

고려 사항

표 31 사용자 인터페이스 설계 - 내 정보 (UI-008)

UI 고유번호	UI-009	UI 명	현장 줄서기 신청
관련 유스케이스 ID	USC-014		
UI 유형	입출력	메뉴경로	메인 화면/현장 줄서기 실행
UI 개요	현장 줄서기 신청 화면		

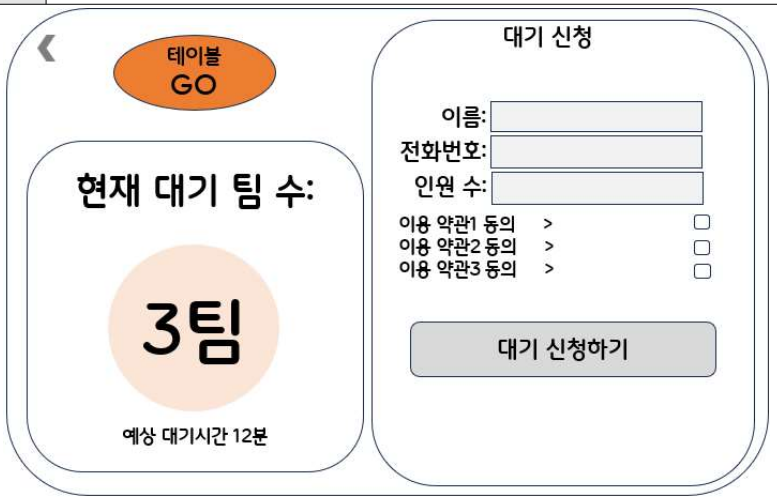

그림 19 현장 줄서기 신청 UI

처리 내용

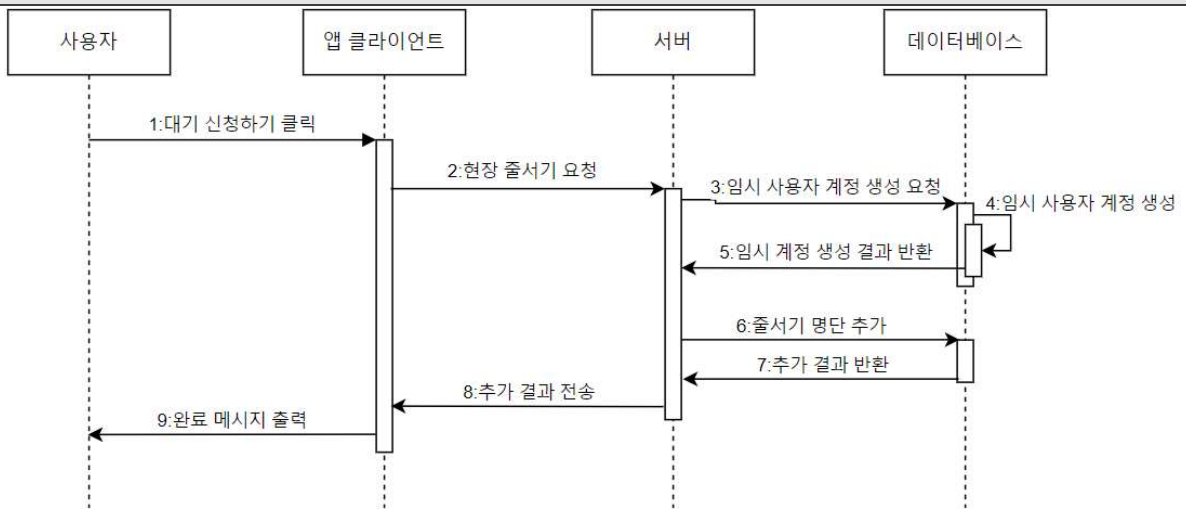

그림 20 현장 줄서기 신청 순차 다이어그램

- 화면 시작
  - 화면 좌측에는 현재 매장 줄서기 팀 수와 예상 대기시간을 출력하고, 우측에 이름, 전화번호, 인원수, 약관 동의를 한다.
- 대기 신청하기
  - 버튼 클릭 시, 입력한 정보들로 임시 사용자 계정을 만들어서 매장의 줄서기 명단에 추가한다. (누락된 정보가 있을 시, 에러 메시지 출력) 이후 완료 메시지를 출력한다.

– 뒤로가기

· 뒤로가기 버튼 클릭 시, 메인 화면–매장용 SW로 이동한다.

입출력 항목			
항목명	타입/길이	속성	비고
이름	String/1~15	입력	필수
전화번호	String/11	입력	필수
인원수	int/2	입력	필수
이용약관	String	출력	–
이용약관 동의	boolean	입력	필수
대기 신청하기	Button	입력	필수
대기 팀 수	int	출력	–
예상 대기시간	int	출력	–

고려 사항

- 임시 사용자 계정 생성 시, 카카오톡 계정을 연동해서 알림을 보낼 수 있도록 한다.
- 해당 화면은 가로 상태로만 실행된다.

표 32 사용자 인터페이스 설계 - 현장 줄서기 신청 (UI-009)

UI 고유번호	UI-010	UI 명	대기 명단 관리
관련 유스케이스 ID	USC-015, USC-016, USC-017		
UI 유형	입출력	메뉴경로	메인 화면/줄서기 관리
UI 개요	줄서기 대기명단 관리 화면		

그림 21 대기 명단 관리 UI

처리 내용

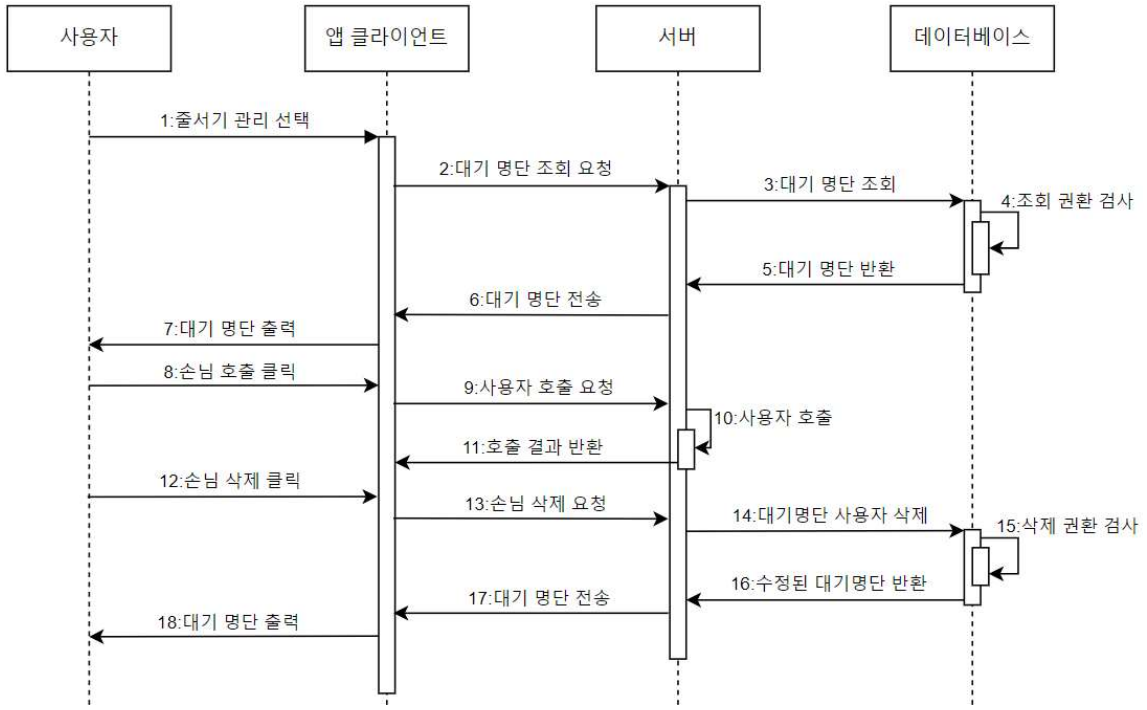

그림 22 대기 명단 관리 순차 다이어그램

– 화면 시작
  · 매장의 현재 줄서기 대기 명단을 화면에 출력한다.(출력 정보에는 대기 번호, 이름이 있다)
– 순서 변경
  · 대기 번호 좌측 바를 이용해 드래그 앤 드롭 방식으로 대기자들의 순서를 변경한다.
– 호출
  · 호출 버튼 클릭 시, 해당 대기자에게 카카오톡 알림톡이 전송된다.
– 삭제
  · 삭제 버튼 클릭 시, 해당 대기자가 대기 명단에서 삭제된다.
– 뒤로가기
  · 뒤로가기 버튼 클릭 시, 메인 화면–매장용 SW로 이동한다.

입출력 항목			
항목명	타입/길이	속성	비고
대기 번호	int	출력	–
이름	String/1~15	출력	–
호출	Button	입력	선택
삭제	Button	입력	선택

고려 사항

- 대기 명단은 실시간으로 업데이트돼야 한다.
- 명단이 화면을 넘어갈 경우를 대비해서 스크롤 방식으로 구현한다.

표 33 사용자 인터페이스 설계 - 대기 명단 관리 (UI-010)

UI 고유번호	UI-011	UI 명	내 매장 정보
관련 유스케이스 ID	USC-018, USC-019, USC-020, USC-021, USC-022, USC-023		
UI 유형	입출력	메뉴경로	메인 화면/내 매장 정보
UI 개요	내 매장 정보 조회 및 수정 화면		

그림 23 내 매장 정보 UI

처리 내용

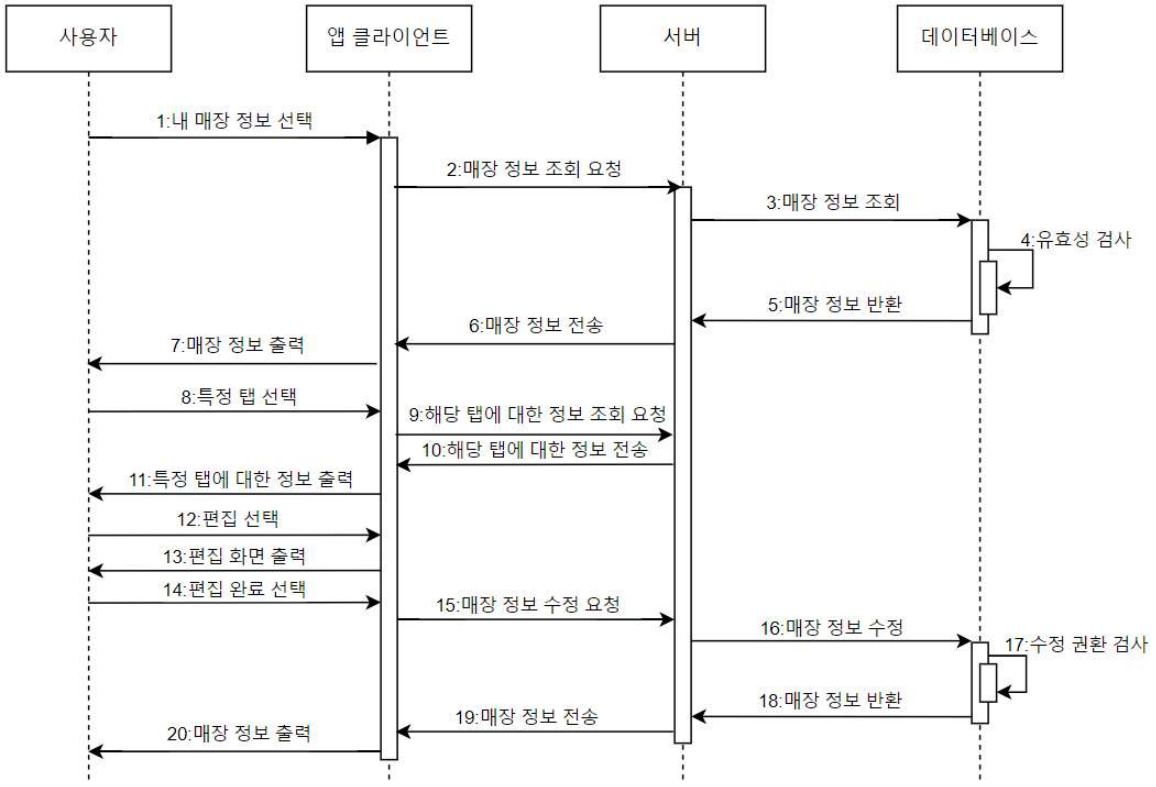

그림 24 내 매장 정보 순차 다이어그램

- 화면 시작
  · 매장 정보, 메뉴, 리뷰 탭을 선택하면 중앙 화면에 관련 정보가 출력된다.
- 매장 정보
  · 매장 정보 선택 시, 주소, 사업자, 전화번호 등의 매장 정보가 중앙 화면에 출력된다.
- 메뉴
  · 메뉴 선택 시, 모든 메뉴와 가격, 사진이 리스트 형태로 중앙 화면에 출력된다.
- 리뷰
  · 리뷰 선택 시, 모든 리뷰와 별점이 리스트 형태로 중앙 화면에 출력된다.
- 편집
  · 편집 버튼 클릭 시, 매장 정보에 관한 내용들을 수정 가능한 형태로 변경한다.(수정이 끝난 후 같은 위치의 완료 버튼을 클릭하면 적용된다.)
- 영업 상태
  · 영업 상태 버튼 클릭 시, 매장의 영업 여부가 변경된다.
- 뒤로가기
  · 뒤로가기 버튼 클릭 시, 메인 화면-매장용 SW로 이동한다.

입출력 항목			
항목명	타입/길이	속성	비고
매장 정보	Table/제한 없음	출력	-
메뉴	List<Table>/제한없음	출력	-
리뷰	List<Table>/제한없음	출력	-
영업 상태	Button	입력	선택
편집	Button	입력	선택

고려 사항

표 34 사용자 인터페이스 설계 - 내 매장 정보 (UI-011)

4. 컴포넌트 설계

○ 컴포넌트 구조도

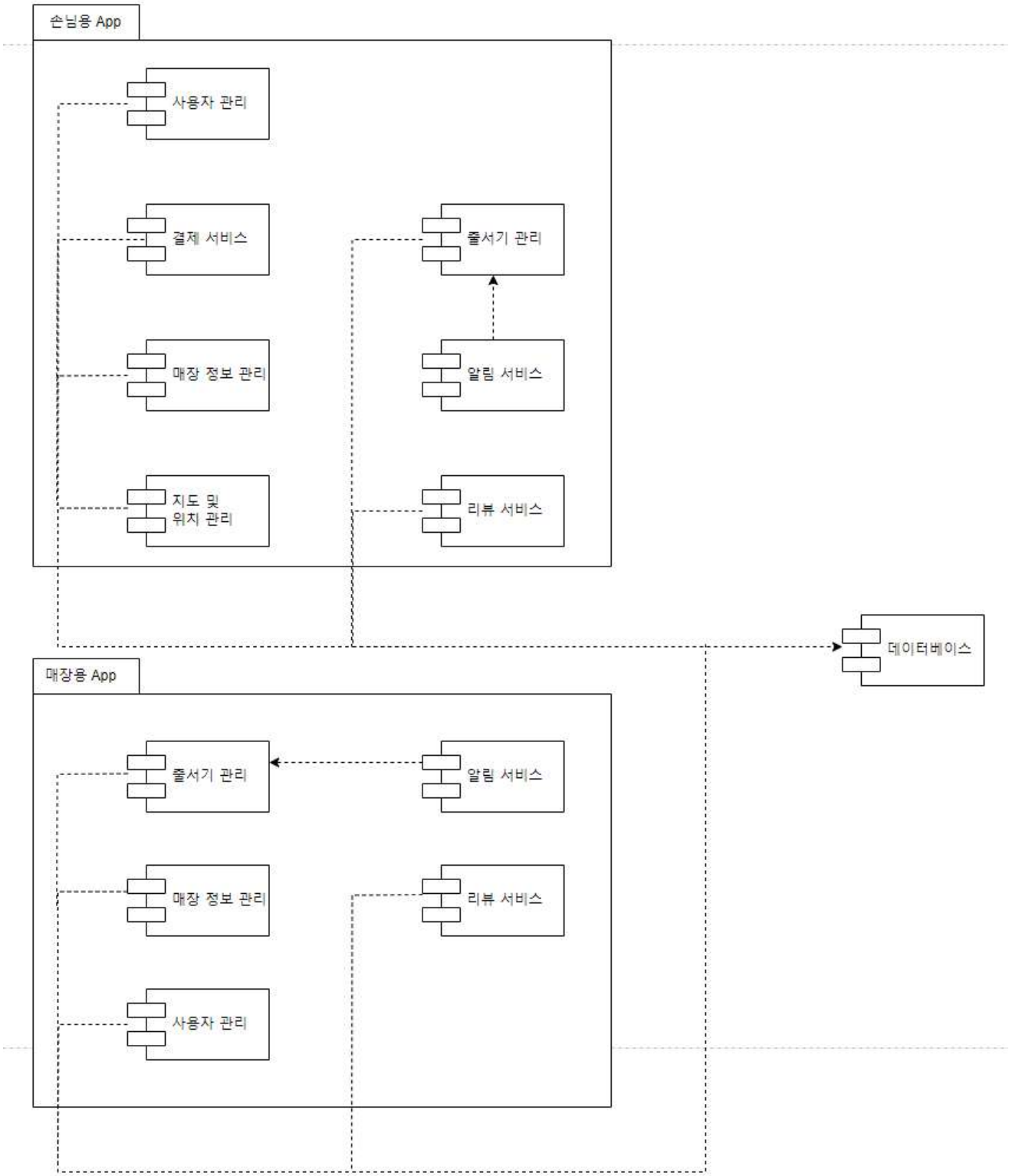

그림 25 컴포넌트 구조도

○ 컴포넌트 목록

컴포넌트 ID	컴포넌트 명	관련 유스케이스 ID
CP-001	사용자 관리	USC-001, USC-002, USC-004, USC-013, USC-024
CP-002	줄서기 관리	USC-003, USC-014, USC-015, USC-016
CP-003	매장 정보 관리	USC-018~USC-021
CP-004	알림 서비스	USC-017
CP-005	지도 및 위치 관리	USC-005
CP-006	데이터베이스	USC-001~USC-024
CP-007	결제 서비스	USC-012
CP-008	리뷰 서비스	USC-021, USC-022

표 35 컴포넌트 목록

○ 컴포넌트 설계

컴포넌트 고유번호	CP-001	컴포넌트 명	사용자 관리
관련 유스케이스 ID	USC-001, USC-002, USC-004, USC-013, USC-024		
컴포넌트 개요	회원가입, 로그인, 로그아웃 같은 사용자 관리 담당 컴포넌트		

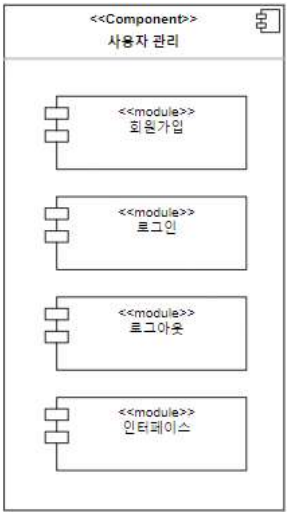

그림 26 컴포넌트 설계
- 사용자 관리

내부 클래스		
클래스 ID	클래스 명	비고
CL-001	UserManager	회원가입, 로그인, 로그아웃 처리, 사용자 정보 조회
CL-002	UserInfo	이메일, 비밀번호, 닉네임, 전화번호, 고유 ID에 대한 정보가 존재하는 사용자 클래스

인터페이스 클래스			
인터페이스 ID	인터페이스 명	오퍼레이션 명	비고
IF-001	UserManageable	registerUser	
		loginUser	
		logoutUser	
		getUserInfo	

표 36 컴포넌트 설계 - 사용자 관리 (CP-001)

컴포넌트 고유번호	CP-002	컴포넌트 명	줄서기 관리
관련 유스케이스 ID	USC-003, USC-014, USC-015, USC-016		
컴포넌트 개요	줄서기 리스트로 사용자의 줄서기 요청을 처리, 줄서기 상태 업데이트, 예상 대기시간 계산, 줄서기 손님 관리		

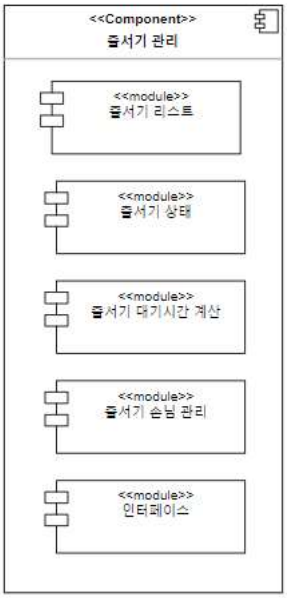

그림 27 컴포넌트 설계
- 줄서기 관리

내부 클래스		
클래스 ID	클래스 명	비고
CL-003	ListManager	리스트로 목록 관리, 대기 명단 조회/삭제/순서 변경/호출 기능 작업, 알림 서비스를 멤버로 보유해서 순서에 따라 알림 보내는 작업 수행

인터페이스 클래스			
인터페이스 ID	인터페이스 명	오퍼레이션 명	비고
IF-002	ListManageable	addUser	
		removeUser	
		getUserList	
		changeSeq	
		callUser	

표 37 컴포넌트 설계 - 줄서기 관리 (CP-002)

컴포넌트 고유번호	CP-003	컴포넌트 명	매장 정보 관리
관련 유스케이스 ID	USC-018~USC-021		
컴포넌트 개요	매장 등록, 등록된 매장 정보들 관리 컴포넌트		

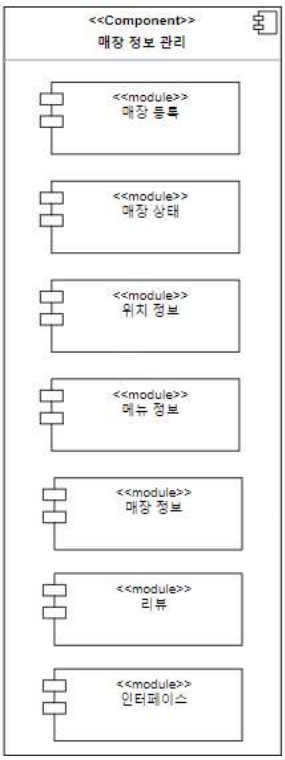

그림 28 컴포넌트 설계
- 매장 정보 관리

내부 클래스		
클래스 ID	클래스 명	비고
CL-004	StoreManager	새로운 매장 등록, 매장 정보 변경, 리뷰 서비스 접근 작업 수행
CL-005	StoreInfo	매장 이름, 위치, 사업자, 고유 ID, 전화번호, 메뉴, 매장 소개, 영업상태에 대한 정보가 존재하는 매장 클래스

인터페이스 클래스			
인터페이스 ID	인터페이스 명	오퍼레이션 명	비고
IF-003	StoreManageable	addStore	
		getStoreInfo	
		getStoreReview	
		updateStoreInfo	

표 38 컴포넌트 설계 - 매장 정보 관리 (CP-003)

컴포넌트 고유번호	CP-004	컴포넌트 명	알림 서비스
관련 유스케이스 ID	USC-017		
컴포넌트 개요	카카오톡 알림톡 API를 이용한 알림 전송, 저장 담당 컴포넌트		

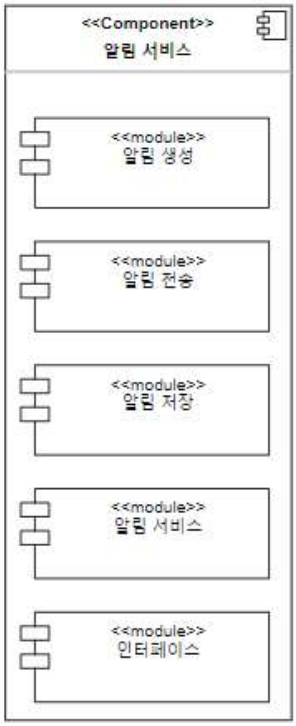

그림 29 컴포넌트 설계
- 알림 서비스

내부 클래스		
클래스 ID	클래스 명	비고
CL-006	NotificationManager	알림 전송, 저장

인터페이스 클래스			
인터페이스 ID	인터페이스 명	오퍼레이션 명	비고
IF-004	NotificationService	sendKakao Notification	카카오톡 API로 구현
		saveNotification	

표 39 컴포넌트 설계 - 알림 서비스 (CP-004)

컴포넌트 고유번호	CP-005	컴포넌트 명	지도 및 위치 서비스
관련 유스케이스 ID	USC-005		
컴포넌트 개요	네이버 지도 API를 이용한 매장의 지도 표시, 위치 검색 담당 컴포넌트		

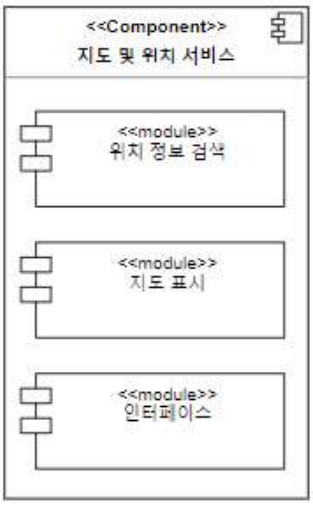

그림 30 컴포넌트 설계
- 지도 및 위치 서비스

내부 클래스			
클래스 ID	클래스 명	비고	
CL-007	MapManager	네이버 지도 API 이용한 다양한 작업 수행	
인터페이스 클래스			
인터페이스 ID	인터페이스 명	오퍼레이션 명	비고
IF-005	MapManageable	showStoreLocationOnMap	네이버 지도 API로 구현
		showUserLocationOnMap	네이버 지도 API로 구현
		showNearbyStoresOnMap	네이버 지도 API로 구현

표 40 컴포넌트 설계 - 지도 및 위치 서비스 (CP-005)

컴포넌트 고유번호	CP-006	컴포넌트 명	데이터베이스
관련 유스케이스 ID	USC-001~USC-024		
컴포넌트 개요	데이터베이스에 접근하여 데이터 저장, 쿼리 처리, 트랜잭션 관리 담당 컴포넌트		

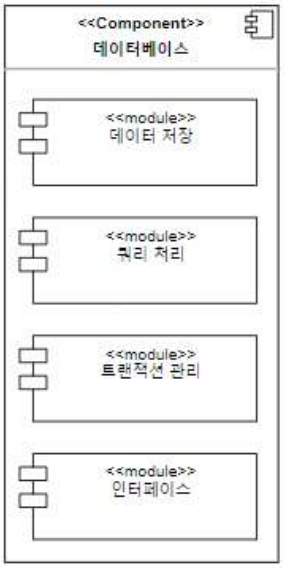

그림 31 컴포넌트 설계
- 데이터베이스

내부 클래스			
클래스 ID	클래스 명	비고	
CL-008	DataManager	데이터 저장/조회	
CL-009	BackupManager	데이터 백업	
인터페이스 클래스			
인터페이스 ID	인터페이스 명	오퍼레이션 명	비고
IF-006	DataManageable	getData	parameter로 받는 데이터 종류에 따라 다른 작업 수행
		updateData	
		removeData	
IF-007	BackupManageable	makeBackup	
		restoreBackup	

표 41 컴포넌트 설계 - 데이터베이스 (CP-006)

컴포넌트 고유번호	CP-007	컴포넌트 명	결제 서비스
관련 유스케이스 ID	USC-012		
컴포넌트 개요	결제 처리, 결제 수단 관리, 보안 담당 컴포넌트		

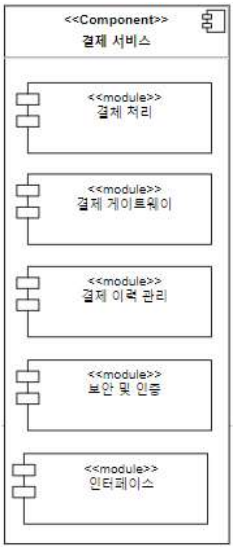

그림 32 컴포넌트 설계
- 결제 서비스

내부 클래스		
클래스 ID	클래스 명	비고
CL-010	PayManager	API를 사용하여 결제 지원
CL-011	CardManager	카드 추가/ 제거/ 그에 대한 보안 지원

인터페이스 클래스			
인터페이스 ID	인터페이스 명	오퍼레이션 명	비고
IF-008	PayManageable	processPayment	
		getPaymentHistory	
IF-009	CardManageable	addCard	
		removeCard	
IF-010	Security Authenticator	checkAccess Permission	
		decryptData	
		encryptData	
IF-011	PaymentGateway	sendPayment Request	오픈뱅킹 API로 구현

표 42 컴포넌트 설계 - 결제 서비스 (CP-007)

컴포넌트 고유번호	CP-008	컴포넌트 명	리뷰 서비스
관련 유스케이스 ID	USC-021, USC-022		
컴포넌트 개요	리뷰 작성, 조회, 수정 담당 컴포넌트		

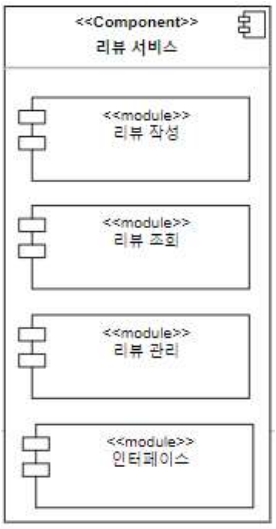

그림 33 컴포넌트 설계
- 리뷰 서비스

내부 클래스		
클래스 ID	클래스 명	비고
CL-012	ReviewManager	리뷰 관리에 대한 모든 작업 수행
CL-013	Review	매장 고유 ID, 작성자 고유 ID, 별점, 작성 내용, 작성 일자, 리뷰 ID가 존재하는 리뷰 클래스

인터페이스 클래스			
인터페이스 ID	인터페이스 명	오퍼레이션 명	비고
IF-012	ReviewManageable	writeReview	
		getReviews	
		editReview	
		removeReview	
		writeReply	

표 43 컴포넌트 설계 - 리뷰 서비스 (CP-008)

○ 인터페이스 설계

인터페이스 고유번호	IF-001	인터페이스 명	UserManageable
오퍼레이션 명	registerUser		
오퍼레이션 개요	파라미터로 입력받은 값들로 UserInfo 객체를 만들어 데이터베이스에 추가한다.		
사전 조건	1) 사용자의 이메일로 가입된 계정이 없어야 한다. 2) 필수 정보들이 모두 입력되어야 한다.		
사후 조건	1) 사용자가 데이터베이스에 등록되어야 한다. 2) 고유 ID가 생성되어서 저장돼야 한다.		
파라미터	1) String:이메일 2) String:비밀번호 3) String:닉네임 4) String:전화번호		
반환값	없다.		

표 44 인터페이스 설계 - UserManageable (IF-001) - registerUser

인터페이스 고유번호	IF-001	인터페이스 명	UserManageable
오퍼레이션 명	loginUser		
오퍼레이션 개요	파라미터로 입력받은 이메일과 비밀번호에 해당하는 사용자 계정으로 로그인한다.		
사전 조건	1) 해당 사용자 계정이 데이터베이스에 존재해야 한다.		
사후 조건	1) 해당 사용자 계정 정보를 불러와 로그인돼야 한다.		
파라미터	1) String:이메일 2) String:비밀번호		
반환값	boolean:로그인 성공 여부		

표 45 인터페이스 설계 - UserManageable (IF-001) - loginUser

인터페이스 고유번호	IF-001	인터페이스 명	UserManageable
오퍼레이션 명	logoutUser		
오퍼레이션 개요	로그인 되어있는 사용자 계정을 로그아웃한다.		
사전 조건	1) 로그인 상태여야 한다.		
사후 조건	1) 사용자 계정이 로그아웃되어야 한다.		
파라미터	없다.		
반환값	없다.		

표 46 인터페이스 설계 - UserManageable (IF-001) - logoutUser

인터페이스 고유번호	IF-001	인터페이스 명	UserManageable
오퍼레이션 명	getUserInfo		
오퍼레이션 개요	파라미터로 입력받은 사용자 객체를 불러온다.		
사전 조건	1) 불러오려는 사용자는 데이터베이스에 존재해야 한다.		
사후 조건	없다.		
파라미터	1) String:사용자 고유 ID		
반환값	1) UserInfo:사용자 객체		

표 47 인터페이스 설계 - UserManageable (IF-001) - getUserInfo

인터페이스 고유번호	IF-002	인터페이스 명	ListManageable
오퍼레이션 명	addUser		
오퍼레이션 개요	대기열에 사용자를 추가한다.		
사전 조건	1) 추가하려는 사용자는 데이터베이스에 존재해야 한다. 2) 대기열은 List 형식으로 default는 뒤쪽에 추가하는 방식이다.		
사후 조건	1) 사용자 객체가 대기열 List 뒤쪽에 추가된다.		
파라미터	1) UserInfo:사용자		
반환값	없다.		

표 48 인터페이스 설계 - UserManageable (IF-001) - addUser

인터페이스 고유번호	IF-002	인터페이스 명	ListManageable
오퍼레이션 명	removeUser		
오퍼레이션 개요	대기열에서 사용자를 삭제한다.		
사전 조건	1) 삭제하려는 사용자가 대기열 List에 존재해야 한다.		
사후 조건	1) 사용자 객체가 대기열 List에서 제거돼야 한다. 2) 제거된 사용자 이후의 사용자 객체들은 1씩 인덱스가 줄어들어야 한다.		
파라미터	1) String:사용자 고유 ID		
반환값	없다.		

표 49 인터페이스 설계 - ListManageable (IF-002) - removeUser

인터페이스 고유번호	IF-002	인터페이스 명	ListManageable
오퍼레이션 명	getUserList		
오퍼레이션 개요	대기열을 불러온다.		
사전 조건	없다.		
사후 조건	1) 대기열 List가 반환돼야 한다.		
파라미터	없다.		
반환값	1) List<UserInfo>:대기열 List		

표 50 인터페이스 설계 - ListManageable (IF-002) - getUserList

인터페이스 고유번호	IF-002	인터페이스 명	ListManageable
오퍼레이션 명	changeSeq		
오퍼레이션 개요	파라미터로 입력받은 사용자 고유 ID에 해당하고 대기열에 있는 사용자의 순서를 변경한다.		
사전 조건	1) 새로운 순서는 대기열의 크기를 넘지 않아야 한다.		
사후 조건	1) 대기열이 변경돼야 한다. 2) 목적지 인덱스 이후 사용자들은 인덱스들이 1씩 밀려야 한다.		
파라미터	1) String:사용자 고유 ID 2) Int:목적지 인덱스		
반환값	없다.		

표 51 인터페이스 설계 - ListManageable (IF-002) - changeSeq

인터페이스 고유번호	IF-002	인터페이스 명	ListManageable
오퍼레이션 명	callUser		
오퍼레이션 개요	대기열에서 파라미터로 입력받은 인덱스 번호에 해당하는 사용자에게 알림을 보내 호출한다.		
사전 조건	없다.		
사후 조건	1) 해당 사용자에 대해 NotificationService 인터페이스의 sendKakaoNotification 오퍼레이션이 실행돼야 한다.		
파라미터	1) Int:대기열의 인덱스 번호		
반환값	없다.		

표 52 인터페이스 설계 - ListManageable (IF-002) - callUser

인터페이스 고유번호	IF-003	인터페이스 명	StoreManageable
오퍼레이션 명	addStore		
오퍼레이션 개요	데이터베이스에 매장을 추가한다.		
사전 조건	1) 데이터베이스에 추가하려는 매장과 같은 매장이 없어야 한다.		
사후 조건	1) 데이터베이스에 매장이 추가돼야 한다.		
파라미터	1) StoreInfo:추가하려는 매장 객체		
반환값	없다.		

표 53 인터페이스 설계 - StoreManageable (IF-003) - addStore

인터페이스 고유번호	IF-003	인터페이스 명	StoreManageable
오퍼레이션 명	getStoreInfo		
오퍼레이션 개요	매장 고유 ID를 통해 원하는 매장 객체를 불러온다.		
사전 조건	1) 매장 고유 ID에 해당하는 매장이 데이터베이스에 존재해야 한다.		
사후 조건	없다.		
파라미터	1) String:매장 고유 ID		
반환값	1) StoreInfo:매장명에 해당하는 매장		

표 54 인터페이스 설계 - StoreManageable (IF-003) - getStoreInfo

인터페이스 고유번호	IF-003	인터페이스 명	StoreManageable
오퍼레이션 명	getStoreReview		
오퍼레이션 개요	ReviewManageable 인터페이스의 getRiviews 오퍼레이션을 통해서 매장의 리뷰들을 불러온다.		
사전 조건	없다.		
사후 조건	없다.		
파라미터	1) String:매장 고유 ID		
반환값	1) List<Review>:리뷰 리스트		

표 55 인터페이스 설계 - StoreManageable (IF-003) - getStoreReview

인터페이스 고유번호	IF-003	인터페이스 명	StoreManageable
오퍼레이션 명	updateStoreInfo		
오퍼레이션 개요	영업시간, 연락처 등의 매장 정보를 갱신한다.		
사전 조건	1) 갱신하려는 매장이 데이터베이스에 존재해야 한다.		
사후 조건	1) 갱신된 매장 정보를 데이터베이스에 업데이트해야 한다.		
파라미터	1) String:매장 고유 ID 2) StoreInfo:원하는 부분을 갱신한 매장 객체		
반환값	없다.		

표 56 인터페이스 설계 - StoreManageable (IF-003) - updateStoreInfo

인터페이스 고유번호	IF-004	인터페이스 명	NotificationService
오퍼레이션 명	sendKakaoNotification		
오퍼레이션 개요	카카오톡 알림톡 API를 사용하여 사용자에게 카카오톡 알림톡을 전송한다.		
사전 조건	1) 사용자의 카카오톡 계정이 연동된 상태여야 한다.		
사후 조건	없다.		
파라미터	1) String:사용자 고유 ID 2) String:알림톡 내용		
반환값	없다.		

표 57 인터페이스 설계 - NotificationService (IF-004) - sendKakaoNotification

인터페이스 고유번호	IF-004	인터페이스 명	NotificationService
오퍼레이션 명	saveNotification		
오퍼레이션 개요	매장마다 전송하려는 알림톡 내용의 형식을 저장한다.		
사전 조건	없다.		
사후 조건	1) 알림톡 내용의 형식이 데이터베이스에 저장된다.		
파라미터	1) String:매장 고유 ID 2) String:원하는 알림톡 내용		
반환값	없다.		

표 58 인터페이스 설계 - NotificationService (IF-004) - saveNotification

인터페이스 고유번호	IF-005	인터페이스 명	MapManageable
오퍼레이션 명	showStoreLocationOnMap		
오퍼레이션 개요	네이버 지도 API를 사용하여 파라미터로 입력받은 매장의 위치를 지도에 보여준다.		
사전 조건	1) 매장은 데이터베이스에 존재해야 한다.		
사후 조건	1) 매장의 위치가 지도에 출력돼야 한다.		
파라미터	1) String:매장 고유 ID		
반환값	1) String:매장 주소		

표 59 인터페이스 설계 - MapManageable (IF-005) - showStoreLocationOnMap

인터페이스 고유번호	IF-005	인터페이스 명	MapManageable
오퍼레이션 명	showUserLocationOnMap		
오퍼레이션 개요	네이버 지도 API를 사용하여 사용자의 현재 위치를 지도에 보여준다.		
사전 조건	1) 사용자는 프로그램에 로그인된 사용자이다.		
사후 조건	1) 사용자의 위치가 지도에 출력돼야 한다.		
파라미터	없다.		
반환값	1) String:사용자의 현재 주소		

표 60 인터페이스 설계 - MapManageable (IF-005) - showUserLocationOnMap

인터페이스 고유번호	IF-005	인터페이스 명	MapManageable
오퍼레이션 명	showNearbyStoresOnMap		
오퍼레이션 개요	네이버 지도 API를 사용하여 파라미터로 입력받은 주소 반경 매장들을 보여준다.		
사전 조건	없다.		
사후 조건	없다.		
파라미터	1) String:주소		
반환값	1) List<StoreInfo>:매장 List		

표 61 인터페이스 설계 - MapManageable (IF-005) - showNearbyStoresOnMap

인터페이스 고유번호	IF-006	인터페이스 명	DataManageable
오퍼레이션 명	getData		
오퍼레이션 개요	데이터베이스의 파라미터로 받는 테이블에서 데이터를 가져온다.		
사전 조건	1) 원하는 데이터가 데이터베이스에 존재해야 한다.		
사후 조건	없다.		
파라미터	1) String:원하는 데이터가 있는 테이블 명 2) String:가져올 데이터의 ID		
반환값	1) Object:데이터		

표 62 인터페이스 설계 - DataManageable (IF-006) - getData

인터페이스 고유번호	IF-006	인터페이스 명	DataManageable
오퍼레이션 명	updateData		
오퍼레이션 개요	데이터베이스의 특정 테이블에 해당 데이터가 존재하면 데이터를 갱신하고, 존재하지 않으면 새롭게 추가한다.		
사전 조건	없다.		
사후 조건	없다.		
파라미터	1) String:테이블명 2) String:갱신하려는 데이터 ID 3) Object:데이터를 추가해야 할 때, 추가할 데이터 (default=null)		
반환값	없다.		

표 63 인터페이스 설계 - DataManageable (IF-006) - updateData

인터페이스 고유번호	IF-006	인터페이스 명	DataManageable
오퍼레이션 명	removeData		
오퍼레이션 개요	데이터베이스의 특정 테이블에서 데이터를 삭제한다.		
사전 조건	1) 삭제할 데이터가 테이블에 존재해야 한다.		
사후 조건	1) 테이블을 업데이트 한다.		
파라미터	1) String:테이블명 2) String:데이터 ID		
반환값	없다.		

표 64 인터페이스 설계 - DataManageable (IF-006) - updateData

인터페이스 고유번호	IF-007	인터페이스 명	BackupManageable
오퍼레이션 명	makeBackup		
오퍼레이션 개요	데이터베이스의 백업을 수행한다.		
사전 조건	없다.		
사후 조건	1) 백업된 데이터의 집합이 파라미터로 입력받은 특정 경로에 저장된다.		
파라미터	1) String:백업 파일 경로		
반환값	없다.		

표 65 인터페이스 설계 - BackupManageable (IF-007) - makeBackup

인터페이스 고유번호	IF-007	인터페이스 명	BackupManageable
오퍼레이션 명	restoreBackup		
오퍼레이션 개요	특정 시점의 데이터베이스로 복원한다.		
사전 조건	1) 이전 데이터베이스 백업 파일이 존재해야 한다.		
사후 조건	없다.		
파라미터	1) String:백업 파일 경로		
반환값	없다.		

표 66 인터페이스 설계 - BackupManageable (IF-007) - restoreBackup

인터페이스 고유번호	IF-008	인터페이스 명	PayManageable
오퍼레이션 명	processPayment		
오퍼레이션 개요	결제 요청을 받아 처리하고 결제 결과를 반환한다.		
사전 조건	없다.		
사후 조건	1) 결제에 대한 정보를 데이터베이스에 저장한다.		
파라미터	1) String:주문 ID 2) BigDecimal:총액 3) String:사용자 고유 ID 4) String:결제 방법		
반환값	1) boolean:결제 결과		

표 67 인터페이스 설계 - PayManageable (IF-008) - processPayment

인터페이스 고유번호	IF-008	인터페이스 명	PayManageable
오퍼레이션 명	getPaymentHistory		
오퍼레이션 개요	저장된 사용자의 결제 이력을 조회한다.		
사전 조건	1) 결제 이력은 데이터베이스에 저장돼 있어야 한다.		
사후 조건	없다.		
파라미터	1) String:사용자 고유 ID		
반환값	1) List<Payment>:결제 정보 리스트		

표 68 인터페이스 설계 - PayManageable (IF-008) - getPaymentHistory

인터페이스 고유번호	IF-009	인터페이스 명	CardManageable
오퍼레이션 명	addCard		
오퍼레이션 개요	사용자가 간편하고 빠르게 결제할 수 있도록 카드를 등록한다.		
사전 조건	없다.		
사후 조건	1) 카드 정보가 데이터베이스에 저장돼야 한다.		
파라미터	1) String:카드번호 2) String:유효일자 3) String:CVC번호		
반환값	1) boolean:등록 여부		

표 69 인터페이스 설계 - CardManageable (IF-009) - addCard

인터페이스 고유번호	IF-009	인터페이스 명	CardManageable
오퍼레이션 명	removeCard		
오퍼레이션 개요	등록된 카드를 삭제한다.		
사전 조건	1) 등록된 카드가 존재해야 한다.		
사후 조건	1) 데이터베이스에서 카드 정보를 삭제해야 한다.		
파라미터	1) String:카드번호		
반환값	1) boolean:삭제 여부		

표 70 인터페이스 설계 - CardManageable (IF-009) - removeCard

인터페이스 고유번호	IF-010	인터페이스 명	SecurityAuthenticator
오퍼레이션 명	checkAccessPermission		
오퍼레이션 개요	사용자의 접근 권한을 확인한다.		
사전 조건	없다.		
사후 조건	1) 사용자의 접근 권한을 임시로 저장해 일정 시간 유지해야 한다.		
파라미터	1) String:사용자 고유 ID 2) String:접근하려는 자원		
반환값	1) boolean:접근 권한 여부		

표 71 인터페이스 설계 - SecurityAuthenticator (IF-010) - checkAccessPermission

인터페이스 고유번호	IF-010	인터페이스 명	SecurityAuthenticator
오퍼레이션 명	decryptData		
오퍼레이션 개요	암호화된 민감한 정보를 복호화한다.		
사전 조건	1) 암호화 복호화에 대한 체계가 잡혀있어야 한다.		
사후 조건	없다.		
파라미터	1) String:암호화되어 있는 데이터		
반환값	1) String:복호화된 데이터		

표 72 인터페이스 설계 - SecurityAuthenticator (IF-010) - decryptData

인터페이스 고유번호	IF-010	인터페이스 명	SecurityAuthenticator
오퍼레이션 명	encryptData		
오퍼레이션 개요	결제에 필요한 민감한 정보를 암호화한다.		
사전 조건	1) 암호화 복호화에 대한 체계가 잡혀있어야 한다.		
사후 조건	없다.		
파라미터	1) String:데이터		
반환값	1) String:암호화된 데이터		

표 73 인터페이스 설계 - SecurityAuthenticator (IF-010) - encryptData

인터페이스 고유번호	IF-011	인터페이스 명	PaymentGateway
오퍼레이션 명	sendPaymentRequest		
오퍼레이션 개요	오픈뱅킹 API의 결제 서비스와 통신하여 결제 요청을 전송한다.		
사전 조건	IF-008의 processPayment가 실행 중이어야 한다.		
사후 조건	없다.		
파라미터	1) String:주문 ID 2) BigDecimal:총액 3) String:사용자 고유 ID 4) String:결제 방법		
반환값	1) String:결제 응답		

표 74 인터페이스 설계 - PaymentGateway (IF-011) - sendPaymentRequest

인터페이스 고유번호	IF-012	인터페이스 명	ReviewManageable
오퍼레이션 명	writeReview		
오퍼레이션 개요	리뷰를 작성한다.		
사전 조건	1) 리뷰 작성을 시도하는 사용자는 해당 매장에 대한 주문 이력이 존재해야 한다.		
사후 조건	1) 작성된 리뷰는 객체화되어 데이터베이스에 저장돼야 한다.		
파라미터	1) String:주문 ID 2) String:사용자 고유 ID 3) float:별점 4) String:리뷰 내용		
반환값	1) boolean:작성 완료 여부		

표 75 인터페이스 설계 - ReviewManageable (IF-012) - writeReview

인터페이스 고유번호	IF-012	인터페이스 명	ReviewManageable
오퍼레이션 명	getReviews		
오퍼레이션 개요	리뷰를 조회한다.		
사전 조건	없다.		
사후 조건	없다.		
파라미터	1) String:매장 고유 ID or 사용자 고유 ID		
반환값	1) List<Review>:매장 리뷰 리스트		

표 76 인터페이스 설계 - ReviewManageable (IF-012) - getReviews

인터페이스 고유번호	IF-012	인터페이스 명	ReviewManageable
오퍼레이션 명	editReview		
오퍼레이션 개요	리뷰를 수정한다.		
사전 조건	1) 이미 작성된 리뷰가 존재해야 한다. 2) 사용자 본인이 작성한 리뷰만 수정 가능해야 한다.		
사후 조건	1) 수정된 리뷰를 데이터베이스에 업데이트한다.		
파라미터	1) String:리뷰 ID 2) float:별점 3) String:수정된 내용		
반환값	1) boolean:수정 완료 여부		

표 77 인터페이스 설계 - ReviewManageable (IF-012) - editReview

인터페이스 고유번호	IF-012	인터페이스 명	ReviewManageable
오퍼레이션 명	deleteReview		
오퍼레이션 개요	리뷰를 삭제한다.		
사전 조건	1) 리뷰 삭제는 사용자 본인만이 할 수 있어야 한다.		
사후 조건	1) 데이터베이스에서 해당 리뷰를 삭제한다.		
파라미터	1) String:리뷰 ID 2) String:사용자 고유 ID		
반환값	1) boolean:삭제 완료 여부		

표 78 인터페이스 설계 - ReviewManageable (IF-012) - deleteReview

인터페이스 고유번호	IF-012	인터페이스 명	ReviewManageable
오퍼레이션 명	writeReply		
오퍼레이션 개요	리뷰에 답글을 작성한다.		
사전 조건	1) 답글 작성은 해당 매장에 대한 수정 권한이 있는 사용자만이 가능해야 한다. 2) 리뷰는 데이터베이스에 존재해야 한다.		
사후 조건	1) 데이터베이스에서 리뷰와 같은 테이블에 저장돼야 한다.		
파라미터	1) String:리뷰 ID 2) String:사용자 고유 ID 3) String:내용		
반환값	1) boolean:작성 완료 여부		

표 79 인터페이스 설계 - ReviewManageable (IF-012) - writeReply

5. 클래스 설계

○ 클래스 목록

클래스 ID	클래스 명	관련 컴포넌트 ID
CL-001	UserManager	CP-001
CL-002	UserInfo	CP-001
CL-003	ListManager	CP-002
CL-004	StoreManager	CP-003
CL-005	StoreInfo	CP-003
CL-006	NotificationManager	CP-004
CL-007	MapManager	CP-005
CL-008	DataManager	CP-006
CL-009	BackupManager	CP-006
CL-010	PayManager	CP-007
CL-011	CardManager	CP-007
CL-012	ReviewManager	CP-008
CL-013	Review	CP-008

표 80 클래스 목록

○ 클래스 설계

클래스 고유번호	CL-001	클래스 명	UserManager
속성			
속성명	가시성	타입	기본값	비고
userName	protected	String	-	사용자 이름
userPassword	protected	String	-	비밀번호
userEmail	protected	String	-	이메일
is_logged_in	private	boolean	false	로그인 상태
is_registered	private	boolean	false	가입 상태
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
loginUser	public	String userEmail,String userPassword	boolean	로그인
logoutUser	public	void	void	로그아웃
registerUser	public	userName,Password,Email	String	회원가입

표 81 클래스 설계 - UserManager (CL-001)

클래스 고유번호	CL-002	클래스 명	UserInfo
속성			
속성명	가시성	타입	기본값	비고
userName	protected	String	-	사용자 이름
userPassword	protected	String	-	비밀번호
userEmail	protected	String	-	이메일
userNickname	protected	String	-	닉네임
userPhoneNumber	protected	int	-	전화번호
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
setUserPassword	public	String userPassword	void	비밀번호 저장
setUserPhoneNumber	public	int userPhoneNumber	void	전화번호 저장
setUserEmail	public	String userEmail	void	이메일 저장
setUserNickName	public	String userNickName	void	닉네임 저장

표 82 클래스 설계 - UserInfo (CL-002)

클래스 고유번호	CL-003	클래스 명	ListManager
속성			
속성명	가시성	타입	기본값	비고
userName	private	String	-	사용자 이름
waitingList	private	List	-	대기자 명단 리스트
notification	private	List	-	알림 서비스
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
addUser	public	userName	void	대기 명단에 추가
removeUser	public	userName	void	대기 명단에서 삭제
changeSeq	public	waitingList	void	순서 변경
callUser	public	userName	void	손님 호출

표 83 클래스 설계 - ListManager (CL-003)

클래스 고유번호	CL-004	클래스 명	StoreManager
속성			
속성명	가시성	타입	기본값	비고
storeLocation	private	String	-	가게 위치
storeMenu	private	String	-	메뉴
storeList	private	List	-	가게 정보 리스트
storeName	private	String	-	가게 이름
category	private	String	-	가게 카테고리
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
addStore	public	storeName,category	void	매장 등록
updateMenu	public	storeMenu	void	메뉴 변경
getReview	public			리뷰 관리

표 84 클래스 설계 - StoreManager (CL-004)

클래스 고유번호	CL-005	클래스 명	StoreInfo
속성			
속성명	가시성	타입	기본값	비고
storeLocation	private	String	-	가게 위치
storeMenu	private	String	-	메뉴
storeList	private	List	-	가게 정보 리스트
storeName	private	String	-	가게 이름
StoreID	private	String	-	고유 ID
isOpen	private	boolean	false	영업 상태
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
setStore	public	String storeName	void	매장 이름 저장
setStoreLocation	public	String StoreLocation	void	가게 위치 저장
setMenu	public	String storeMenu	void	메뉴 저장

표 85 클래스 설계 - StoreInfo (CL-005)

클래스 고유번호	CL-006	클래스 명	NotificationManager
속성			
속성명	가시성	타입	기본값	비고
message	private	String	void	알림 메시지
userID	protected	String	void	손님 아이디
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
sendNotification	public	String userId, String message	void	알림 전송
setNotification	public	String message	String	알림 저장

표 86 클래스 설계 - NotificationManager (CL-006)

클래스 고유번호	CL-007		클래스 명	MapManager
속성				
속성명	가시성	타입	기본값	비고
storeName	private	String	-	매장 이름
storeLocation	private	String	-	매장 위치
storeList	private	List	-	매장 리스트
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
showStore	public	String storeName, String storeLocation	void	매장 조회
findStore	public	storeList	storeLocation	매장 검색

표 87 클래스 설계 - MapManager (CL-007)

클래스 고유번호	CL-008		클래스 명	DataManager
속성				
속성명	가시성	타입	기본값	비고
StoreData	private	List	void	매장 데이터
UserData	private	List	void	손님 데이터
NewData	private	List	void	새로운 데이터
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
getData	public	StoreData, UserData	void	데이터 저장
updateData	public	NewData	void	데이터 수정
removeData	public	StoreData, UserData	void	데이터 삭제

표 88 클래스 설계 - DataManager (CL-008)

클래스 고유번호	CL-009		클래스 명	BackupManager
속성				
속성명	가시성	타입	기본값	비고
StoreData	private	List	void	매장 데이터
UserData	private	List	void	손님 데이터
backupQuery	private	String	void	백업할 경로
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
makeBackup	private	StoreData,User Data	List	백업 생성
restoreBackup	private	backupQuery	List	백업 복구

표 89 클래스 설계 - BackupManager (CL-009)

클래스 고유번호	CL-010		클래스 명	PayManager
속성				
속성명	가시성	타입	기본값	비고
userID	private	int	-	
cardNumber	private	int	-	
expirationDate	private	String	-	
cvc	pivate	String	-	
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
addPaymentInfo	public	userID,cardNumber,expirationDate,cvc	void	결제 정보 추가 메서드
removePaymentInfo	public	userID,cardNumber	void	결제 정보 삭제 메서드
PaymentHistory	public	userID	String	결제 정보 조회

표 90 클래스 설계 - PayManager (CL-010)

클래스 고유번호	CL-011		클래스 명	CardManager
속성				
속성명	가시성	타입	기본값	비고
cardNumber	private	int	-	
expirationDate	private	String	-	
cvc	pivate	String	-	
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
addCard	private	cardNumber,expirationDate,cvc	void	카드 추가
removeCard	private	cardNumber,expirationDate,cvc	void	카드 제거

표 91 클래스 설계 - CardManager (CL-011)

클래스 고유번호	CL-012		클래스 명	ReviewManager
속성				
속성명	가시성	타입	기본값	비고
storeID	private	String	void	매장 고유 ID
userID	private	String	void	작성자 고유 ID
review	private	List	void	리뷰 리스트
date	private	String	void	작성 날짜
rating	private	float	void	별점
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
writeReview	public	storeID,userID,review,date,rating	List	리뷰 작성
editReview	public	storeID,userID,review,rating	List	리뷰 수정
removeReview	public	void	void	리뷰 삭제

표 92 클래스 설계 - ReviewManager (CL-012)

클래스 고유번호	CL-013		클래스 명	Review
속성				
속성명	가시성	타입	기본값	비고
storeID	private	String	void	매장 고유 ID
userID	private	String	void	작성자 고유 ID
review	private	List	void	리뷰 리스트
date	private	String	void	작성 날짜
rating	private	float	void	별점
오퍼레이션				
오퍼레이션명	가시성	파라미터	반환타입	비고
setReview	public	storeID,userID,review,date,rating	List	리뷰 저장

표 93 클래스 설계 - Review (CL-013)

6. 인터페이스 설계

○ 인터페이스 목록

인터페이스 ID	인터페이스 명	관련 유스케이스 ID
NI-001	결제관리	USC-012
NI-002	매장위치관리	USC-005,USC-018
NI-003	손님 관리	USC-001,USC-002, USC-017

표 94 인터페이스 목록

○ 인터페이스 설계

<table>
<tr><td>인터페이스 고유번호</td><td>NI-001</td><td>인터페이스 명</td><td>결제관리</td></tr>
<tr><td>인터페이스 개요</td><td colspan="3">앱 내 결제를 위한 뱅킹 오픈 api 연동</td></tr>
<tr><td colspan="4">송신</td></tr>
<tr><td>송신 대상</td><td>결제 정보 관리 시스템</td><td>고유번호</td><td>P-A001</td></tr>
<tr><td colspan="4">전달</td></tr>
<tr><td>전달 방식</td><td>RESTful API,HTTP POST</td><td>데이터 형식</td><td>JSON</td></tr>
<tr><td>세부정보</td><td colspan="3"><table>
<tr><td>속성명</td><td>데이터 타입</td><td>길이</td><td>비고</td></tr>
<tr><td>amount</td><td>int</td><td>10</td><td>결제 금액</td></tr>
<tr><td>name</td><td>String</td><td>15</td><td>구매 품목</td></tr>
<tr><td>buyer</td><td>String</td><td>5</td><td>구매자</td></tr>
<tr><td>cardNumber</td><td>String</td><td>30</td><td>카드 번호</td></tr>
<tr><td>expirationDate</td><td>String</td><td>10</td><td>만료일</td></tr>
<tr><td>cvc</td><td>int</td><td>3</td><td>CVC</td></tr>
</table></td></tr>
<tr><td colspan="4">수신</td></tr>
<tr><td>수신 대상</td><td>오픈뱅킹 api</td><td>고유번호</td><td>REQ-IF-RG001</td></tr>
<tr><td>관련 요구사항 ID</td><td colspan="3">USC-012</td></tr>
<tr><td>비고</td><td colspan="3">손님의 결제 정보 전송 시 보안을 위해 신뢰할 수 있는 CA로부터 SSL인증서를 발급 받는다.
손님의 금융 정보를 보호하기 위해 결제 세부 정보를 토큰화한다.
사용자의 카드 정보를 JSON 형식으로 저장하여 전송한다.</td></tr>
</table>

표 95 인터페이스 설계 - 결제관리 (NI-001)

<table>
<tr><td>인터페이스 고유번호</td><td>NI-002</td><td>인터페이스 명</td><td>매장위치관리</td></tr>
<tr><td>인터페이스 개요</td><td colspan="3">앱 내 매장 위치 정보 제공을 위한 네이버맵 API 연동</td></tr>
<tr><td colspan="4">송신</td></tr>
<tr><td>송신 대상</td><td>앱</td><td>고유번호</td><td>P-A002</td></tr>
<tr><td colspan="4">전달</td></tr>
<tr><td>전달 방식</td><td>Rest API</td><td>데이터 형식</td><td>XML</td></tr>
<tr><td>세부정보</td><td colspan="3"><table>
<tr><td>속성명</td><td>데이터 타입</td><td>길이</td><td>비고</td></tr>
<tr><td>App_Name</td><td>String</td><td>5</td><td>앱 이름</td></tr>
<tr><td>Client_Id</td><td>String</td><td>10</td><td></td></tr>
<tr><td>Client_Key</td><td>String</td><td>10</td><td></td></tr>
</table></td></tr>
<tr><td colspan="4">수신</td></tr>
<tr><td>수신 대상</td><td>네이버맵 API</td><td>고유번호</td><td>REQ-IF-RG002</td></tr>
<tr><td>관련 요구사항 ID</td><td colspan="3">USC-005,USC-018</td></tr>
<tr><td>비고</td><td colspan="3">모바일 앱에서 사용하기 위해 Mobile Dynamic Map API를 사용한다.
xml에 Client_ID와 Client_Key를 추가하고 맵 api를 추가한다.</td></tr>
</table>

표 96 인터페이스 설계 - 매장위치관리 (NI-002)

<table>
<tr><td>인터페이스 고유번호</td><td>NI-003</td><td>인터페이스 명</td><td>손님 관리</td></tr>
<tr><td>인터페이스 개요</td><td colspan="3">소셜 연동 가입과 대기 손님 호출을 위해 카카오톡 API를 연동</td></tr>
<tr><td colspan="4">송신</td></tr>
<tr><td>송신 대상</td><td>앱</td><td>고유번호</td><td>P-A002</td></tr>
<tr><td colspan="4">전달</td></tr>
<tr><td>전달 방식</td><td>EAI,Transact-SQL</td><td>데이터 형식</td><td>SQL</td></tr>
<tr><td>세부정보</td><td colspan="3"><table>
<tr><td>속성명</td><td>데이터 타입</td><td>길이</td><td>비고</td></tr>
<tr><td>storeName</td><td>String</td><td>10</td><td>매장 이름</td></tr>
<tr><td>number</td><td>int</td><td>5</td><td>손님 대기 번호</td></tr>
<tr><td>storeNumber</td><td>int</td><td>8</td><td>매장 번호</td></tr>
</table></td></tr>
<tr><td colspan="4">수신</td></tr>
<tr><td>수신 대상</td><td>카카오톡 api</td><td>고유번호</td><td>REQ-IF-RG003</td></tr>
<tr><td>관련 요구사항 ID</td><td colspan="3">USC-001,USC-002,USC-017</td></tr>
<tr><td>비고</td><td colspan="3">앱에서 손님이 웨이팅 걸어둔 매장 이름, 손님 대기 번호, 매장 번호 등의 데이터를 api에 제공하여 알림톡을 전송한다.</td></tr>
</table>

표 97 인터페이스 설계 - 손님 관리 (NI-003)

7. 엔티티 관계 모형 설계 - ER 다이어그램까지

○ 엔티티 관계도 (ERD)

ERD 고유번호	ER-001	ERD 명	UserPayment

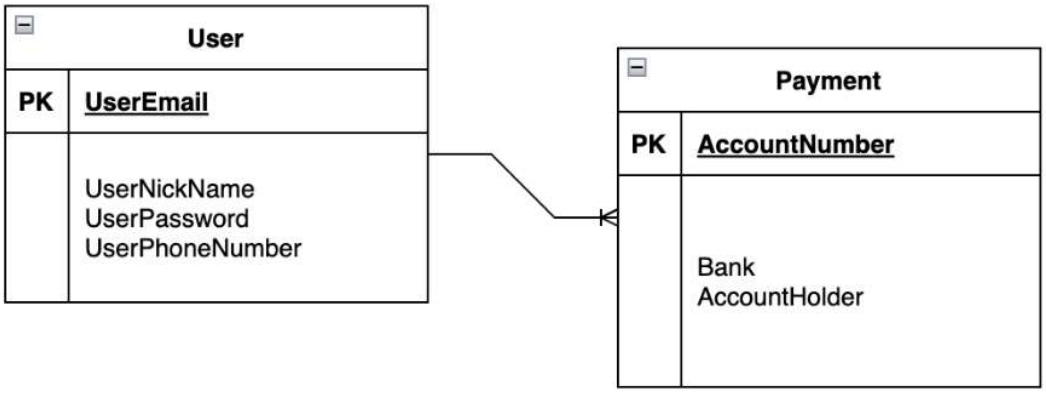

그림 34 엔티티 관계도 - UserPayment

표 98 엔티티 관계도 - UserPayment (ERD-001)

ERD 고유번호	ER-002	ERD 명	UserStoreReview

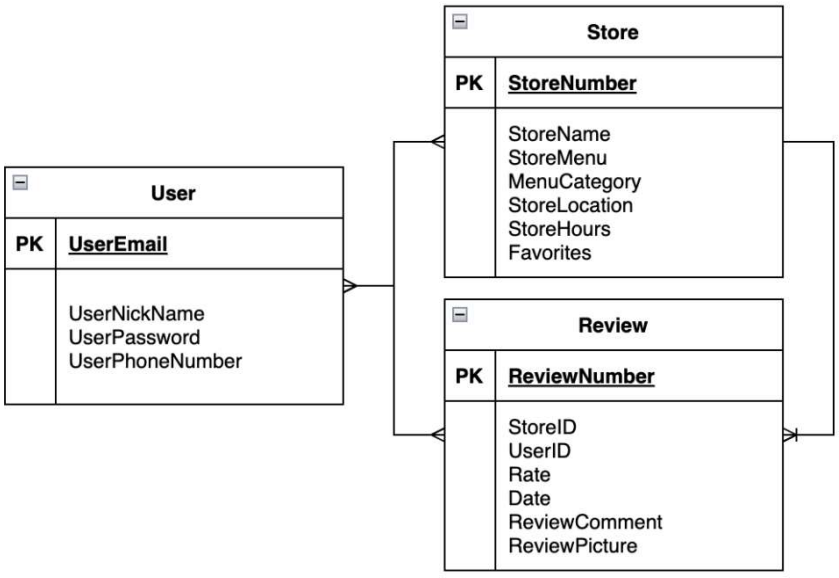

그림 35 엔티티 관계도 - UserStoreReview

표 99 엔티티 관계도 - UserStoreReview (ERD-002)

ERD 고유번호	ER-003	ERD 명	StoreWaiting

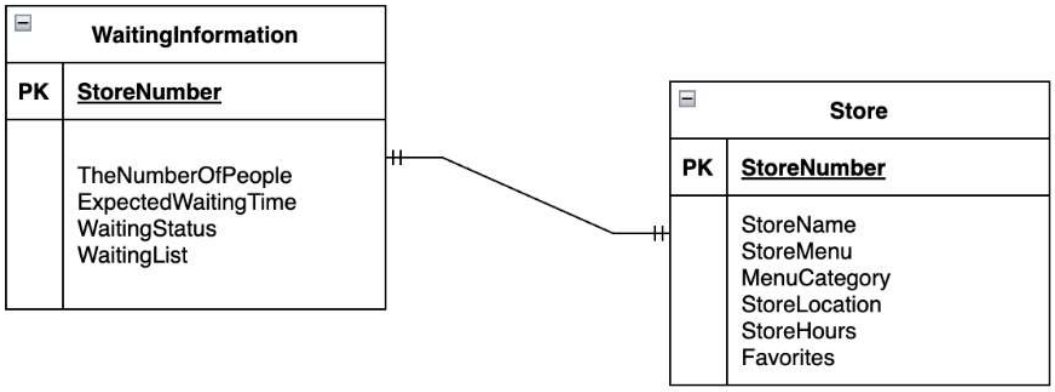

그림 36 엔티티 관계도 - StoreWaiting

표 100 엔티티 관계도 - StoreWaiting (ERD-003)

○ 엔티티 설계

엔티티 고유번호	EN-001	엔티티 명	User
관련 컴포넌트	CP-001		
관련 클래스	CL-001, CL-002		
엔티티 설명	손님의 회원정보를 포함하는 엔티티.		

엔티티 정보					
속성명	데이터 타입	길이	필수유무	기본값	제약조건
UserEmail	String	1~254byte	O	'email @email.com '	중복허용X
UserNickname	String	3~15byte	O	'nickname'	중복허용X
User Password	String	8~64byte	O	'00000000'	
UserPhone Number	String	11byte	O	'00000000000 '	중복허용X

표 101 엔티티 설계 - User (EN-001)

엔티티 고유번호	EN-002	엔티티 명	Review
관련 컴포넌트	CP-008		
관련 클래스	CL-012, CL-013		
엔티티 설명	매장에 등록된 리뷰 정보를 포함하는 엔티티.		

엔티티 정보					
속성명	데이터 타입	길이	필수유무	기본값	제약조건
Review Number	Integer	8byte	O	00000000	중복허용X
StoreID	String	3~15byte	O	'nickname'	
UserID	String	8~64byte	O	'00000000'	
Rate	float	11byte	O	5.0	
Date	String	10byte	O	2000-01-01	
Review Comment	String	10~300byte	O	null	
Review Picture	URL	1~8000byte	X	null	

표 102 엔티티 설계 - Review (EN-002)

엔티티 고유번호	EN-003	엔티티 명	Store
관련 컴포넌트	CP-003		
관련 클래스	CL-004		
엔티티 설명	매장의 정보를 포함하는 엔티티.		

엔티티 정보					
속성명	데이터 타입	길이	필수유무	기본값	제약조건
StoreNumber	Integer	8byte	O	00000000	중복허용X
StoreName	String	3~15byte	O	'nickname'	
StoreMenu	String	8~64byte	O	'00000000'	
Menu Category	String	11byte	O	'5.0'	
StoreLocation	String	10byte	O	2000-01-01	
StoreHours	Integer	10~300byte	O	null	
Favorites	Integer	6byte	O	000000	

표 103 엔티티 설계 - Store (EN-003)

엔티티 고유번호	EN-004	엔티티 명	WaitingInformation
관련 컴포넌트	CP-002		
관련 클래스	CL-003		
엔티티 설명	줄서기 정보를 포함하는 엔티티.		

엔티티 정보					
속성명	데이터 타입	길이	필수유무	기본값	제약조건
StoreNumber	Integer	8bytes	O	00000000	중복 허용X
TheNumber OfPeople	Integer	4byte	O	0	
Expected WaitingTime	Integer	4byte	O	0	
Waiting Status	Boolean	1byte	O	false	
WaitingList	String	3~15byte	O	null	

표 104 엔티티 설계 - WaitingInformation (EN-004)

엔티티 고유번호	EN-005	엔티티 명	Payment
관련 컴포넌트	CP-007		
관련 클래스	CL-010, CL-011		
엔티티 설명	손님의 결제 정보를 포함한 엔티티.		

엔티티 정보					
속성명	데이터 타입	길이	필수유무	기본값	제약조건
Account Number	Integer	1~20byte	O	'00000000000000000000'	중복허용X
Bank	String	10byte	O	null	
Account Holder	String	3~20byte	O	null	

표 105 엔티티 설계 - Payment (EN-005)