FRRA SRS (Software Requirements Specification)

프로젝트명: FRRA (Food Reservation Recommend Application)

작성일: 2025-04-03

문서 버전: 1.0

팀장: 이수종

팀원: 고유찬, 김관식

# 문서 이력

|  |  |  |
| --- | --- | --- |
| 파일명 | 수정일자 | 작성자명 |
| FRRA(고유찬, 김관식, 이수종).doc | 2025년 4월 3일 | 이수종 |
| FRRA(고유찬, 김관식, 이수종)1차수정본.doc | 2025년 5월 8일 | 이수종 |
| FRRA(고유찬, 김관식, 이수종)2차 수정본.docx | 2025년 5월 9일 | 고유찬 |
| FRRA(고유찬, 김관식, 이수종)3차 수정본.docx | 2025년 5월10일 | 이수종 |
| FRRA(고유찬, 김관식, 이수종)4차 수정본.docx | 2025년 5월 26일 | 고유찬 |
| FRRA(고유찬, 김관식, 이수종)5차 수정본.docx | 2025년 5월 29일 | 김관식 |
| FRRA(고유찬, 김관식, 이수종)6차 수정본.docx | 2025년 5월 30일 | 이수종 |
| FRRA(고유찬, 김관식, 이수종)7차 수정본.docx | 2025년 6월 1일 | 이수종 |
| FRRA(고유찬, 김관식, 이수종)8차 수정본.docx | 2025년 6월 2일 | 이수종 |

# 목차

1. 개발 개요

1.1 제안 개요

1.2 제안 시스템

2. 설계

2.1 기능 및 비기능 요구사항

2.2유즈케이스(Use Case)

2.3 UML 흐름도

2.4 클래스 다이어그램

2.5 시퀀스 다이어그램

3. 화면 구성

4. 참고 문헌

1. 개발 개요

1.1 제안 개요

FRRA(Food Reservation Recommend Application)는 낯선 지역을 방문한 사용자들이 해당 지역의 유명 음식점이나 대표적인 음식 정보를 확인하고 예약할 수 있도록 도와주는 모바일 애플리케이션이다.  
사용자는 자신의 음식 취향을 등록하고, 해당 정보를 기반으로 맞춤형 음식점 추천을 받을 수 있다. 또한 음식점에 대한 리뷰와 별점을 열람하거나 직접 작성할 수 있어, 신뢰도 높은 정보를 바탕으로 식당을 선택할 수 있다.

추가적으로, 각 음식점의 실시간 인원 현황을 확인할 수 있으며, 예약 기능을 통해 대기 시간 없이 방문할 수 있는 경험을 제공한다.

1.2 제안 시스템

FRRA 시스템은 다음과 같은 기능들로 구성된다.

* 사용자 음식 취향 등록 및 맞춤형 음식점 추천
* 현재 위치 기반 주변 음식점 탐색
* 음식점 예약 및 인원 현황 확인 기능
* 리뷰 및 별점 열람, 작성 기능
* 점주의 음식점 메뉴 등록 및 수정 기능
* 전 연령층이 사용할 수 있도록 직관적인 UI/UX 설계

2. 설계

2.1 기능 및 비기능 요구사항

기능 조건

1. 사용자는 현재 위치를 기점으로 주변 음식점을 조회할 수 있다.

2. 사용자는 음식 취향을 등록할 수 있다.

3. 시스템은 사용자의 음식 취향에 따라 맞춤 음식점을 추천해야 한다.

4. 사용자는 음식점의 인원현황을 실시간으로 확인 할 수 있다.

5. 사용자는 음식점 예약을 생성, 수정, 취소 등을 할 수 있다.

6. 사용자는 다른 사용자들의 리뷰를 확인 할 수 있다.

7. 사용자는 음식점에 대한 리뷰를 남기고 별점을 줄 수 있다.

8. 점주는 자신의 음식점에 대한 메뉴 정보를 등록 및 수정할 수 있다.

9. 리뷰에 비속어가 들어가면 자동으로 필터링을 할 수 있다.

비기능 조건

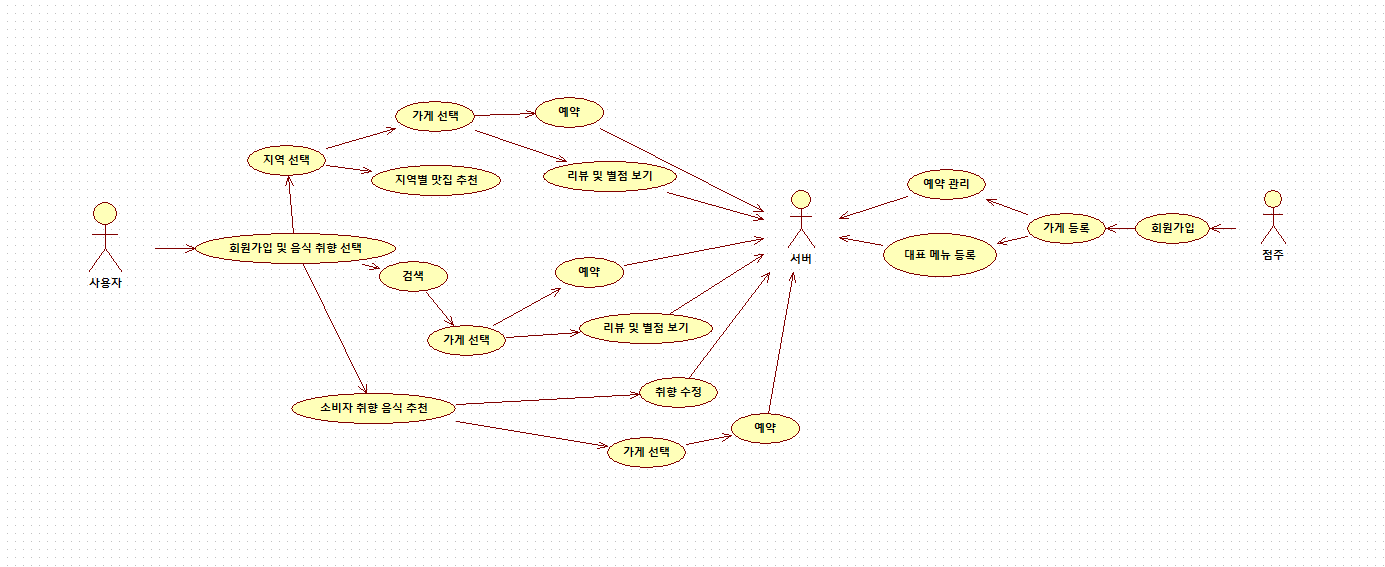
1. UI는 전 연령층이 쉽게 사용할 수 있도록 직관적으로 설계해야 한다.

2. 리뷰 작성 시 욕설 필터링이 되어야 한다.

3. 사용자의 개인정보는 암호화되어 저장되어야 한다.

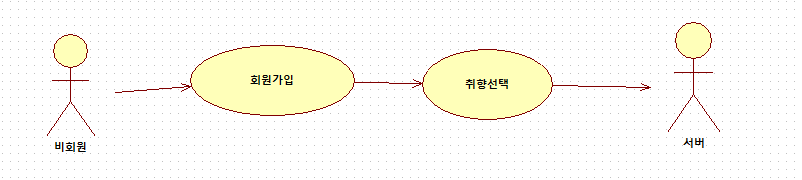
**2.2 유스케이스**

**2.2.1 UML 전체 흐름도**



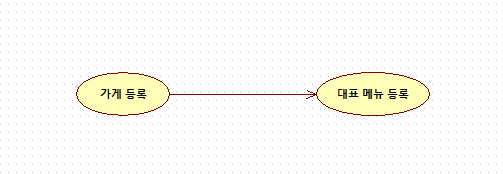
[그림. 1] 전체 유스케이스 흐름도

**2.2.2 가게 및 대표 메뉴등록**



|  |  |
| --- | --- |
| **Use Case#001** | |
| **General Characteristics** | |
| 요약 | 비회원이 회원가입을 통해 본인 취향 음식을 서버에 전송한다. |
| 작성자 | 고유찬 |
| 액터 | 비회원, 서버 |
| 사전 조건 | 비회원이 어플에 접속해 있는 상태 |
| 사후 조건 | 본인 취향 음식 정보가 서버에 저장된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 비회원이 어플 실행 후 회원가입을 클릭한다. |
| 2 | 비회원의 정보와 같이 본인 취향의 음식을 설정한다. |
| 3 | 비회원의 정보(사는 지역, 취향 음식)을 서버에 전송한다. |
| **대안 흐름** | |
| 1 | 비회원이 본인 음식 취향을 따로 설정하지 않을 경우 |
| 1-1 | 팝업 창을 띄워 음식 취향을 설정하게 한다. |

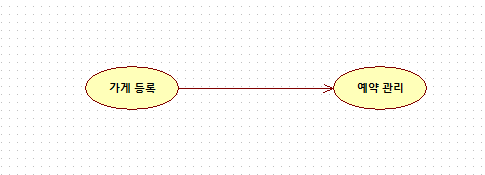
**2.2.3 가게 및 대표 메뉴등록**

****

[그림. 2] 가게 및 대표 메뉴등록 흐름도.

|  |  |
| --- | --- |
| **Use Case#001** | |
| **General Characteristics** | |
| 요약 | 업주가 가게를 등록한 후, 해당 가게의 대표 메뉴를 추가로 등록한다. |
| 작성자 | 김관식 |
| 사전 조건 | 업주가 로그인되어 있고, 가게 등록이 완료된 상태이어야한다. |
| 사후 조건 | 대표 메뉴 정보가 해당 가게에 연결되어 저장된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 업주가 가게 정보를 입력하고 등록을 완료한다. |
| 2 | 등록된 가게에 대해 대표 메뉴 추가 등록 버튼을 선택한다. |
| 3 | 메뉴명, 가격, 설명 등을 입력한 후 등록을 완료한다. |
| **대안 흐름** | |
| 없음 |  |

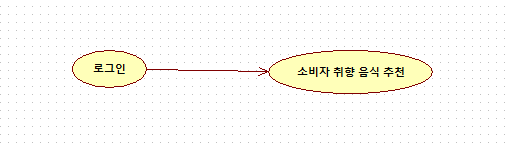
**2.2.4 가게 등록 후 예약 관리**



[그림. 3] 가게 등록 후 예약 관리 흐름도

|  |  |
| --- | --- |
| **Use Case#002** | |
| **General Characteristics** | |
| 요약 | 업주가 가게를 등록한 후, 해당 가게에 대한 고객 예약 정보를 확인하고 관리한다. |
| 작성자 | 김관식 |
| 사전 조건 | 업주가 로그인되어 있고, 가게가 등록된 상태여야한다. |
| 사후 조건 | 예약 요청 상태가 확인되며, 수락 또는 거절 등의 조치가 가능하다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 업주가 가게 등록을 완료한다. |
| 2 | 가게 등록 완료 후 예약 관리 메뉴로 이동한다. |
| 3 | 등록된 가게에 들어온 예약 요청들을 확인한다. |
| 4 | 예약 요청에 대해 수락 또는 거절 등의 처리 작업을 수행한다. |
| **대안 흐름** | |
| 없음 |  |

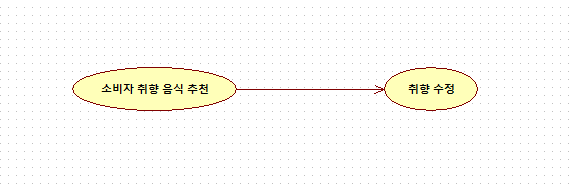
**2.2.5 소비자 취향 음식 추천**



[그림. 4] 소비자 취향 음식 추천 흐름도

|  |  |
| --- | --- |
| **Use Case#003** | |
| **General Characteristics** | |
| 요약 | 사용자가 로그인 후 본인의 취향 정보에 기반하여 맞춤형 음식 추천을 받는다. |
| 작성자 | 김관식 |
| 사전 조건 | 사용자가 로그인되어 있어야 하며, 취향 정보가 등록되어 있어야 한다. |
| 사후 조건 | 사용자에게 맞춤 음식점 리스트가 추천되어 화면에 표시된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 시스템에 로그인한다. |
| 2 | 로그인 후 메인 화면에서 “취향 기반 음식 추천” 버튼을 선택한다. |
| 3 | 시스템은 저장된 취향 정보를 바탕으로 음식점 추천 리스트를 생성한다. |
| 4 | 추천 리스트가 사용자 화면에 표시된다. |
| **대안 흐름** | |
| 1 | 사용자 취향 정보가 등록되지 않은 경우 |
| 1-1 | “추천을 위해 취향 정보를 먼저 설정해주세요” 메시지 출력 후 취향 설정 화면으로 이동한다. |

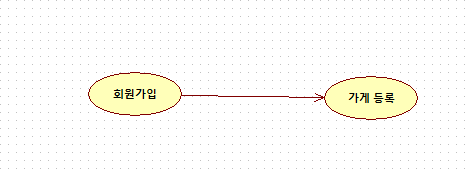
**2.2.6 소비자 취향 수정**

****

[그림. 5] 소비자 취향 수정 흐름도

|  |  |
| --- | --- |
| **Use Case#004** | |
| **General Characteristics** | |
| 요약 | 사용자가 음식 추천 결과를 바탕으로 자신의 취향 정보를 수정한다. |
| 작성자 | 김관식 |
| 사전 조건 | 사용자가 로그인되어 있고, 기존에 취향 정보가 등록되어 있어야한다. |
| 사후 조건 | 수정된 취향 정보가 저장되어, 이후 추천 결과에 반영된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 음식 추천 결과 화면을 확인한다. |
| 2 | 화면 하단 또는 설정 메뉴에서 “취향 수정”을 선택한다. |
| 3 | 기존 취향 정보가 자동으로 불러와지고, 사용자가 항목을 수정한다. |
| 4 | 수정 완료 후 저장 버튼을 누르면 새로운 취향 정보가 반영된다. |
| **대안 흐름** | |
| 1 | 사용자가 기존 취향 정보를 불러오지 못한 경우 |
| 1-1 | “기존 취향 정보를 불러오는 데 실패했습니다” 메시지 출력 및 재시도 유도 |

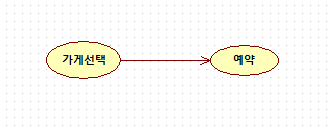
**2.2.7 업주의 회원가입 및 가게 등록**

****

[그림. 6] 회원가입 및 가게 등록 흐름도

|  |  |
| --- | --- |
| **Use Case#005** | |
| **General Characteristics** | |
| 요약 | 업주가 회원가입을 완료한 후, 자신의 가게를 시스템에 등록한다. |
| 작성자 | 김관식 |
| 사전 조건 | 업주가 앱에 접속한 상태이다. |
| 사후 조건 | 업주 계정이 생성되고, 가게 정보가 시스템에 등록된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 업주가 회원가입 양식을 작성하고 제출한다. |
| 2 | 시스템은 업주 계정을 생성하고 로그인 상태로 전환한다. |
| 3 | 업주는 가게 등록 메뉴에서 정보(이름, 주소 등)를 입력한다. |
| 4 | 등록 버튼을 눌러 가게 정보를 시스템에 제출한다. |
| **대안 흐름** | |
| 1 | 필수 입력값이 누락된 경우 |
| 1-1 | “모든 항목을 입력해주세요” 메시지 출력 후 가입 불가 |

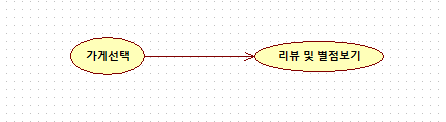
**2.2.7가게 선택 후 예약 요청**

****

[그림. 7] 가계선택 예약 흐름도

|  |  |
| --- | --- |
| **Use Case#006** | |
| **General Characteristics** | |
| 요약 | 사용자가 음식점 목록에서 가게를 선택한 후, 해당 가게에 대해 예약을 요청한다. |
| 작성자 | 김관식 |
| 사전 조건 | 사용자가 로그인되어 있고, 가게 정보가 시스템에 등록되어 있어야한다. |
| 사후 조건 | 서버에 예약 정보가 전송된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 음식점 목록에서 가게를 선택한다. |
| 2 | 사용자가 가게 상세 페이지에서 “예약” 버튼을 클릭한다. |
| 3 | 사용자가 원하는 날짜 및 시간, 인원 수를 선택하고 예약 요청을 보낸다. |
| 4 | 예약 요청이 서버로 전송되어 처리된다. |
| **대안 흐름** | |
| 1 | 예약 시간 또는 인원 수 입력 누락 |
| 1-1 | “모든 예약 정보를 입력해주세요” 메시지 출력 후 예약 불가 처리 |

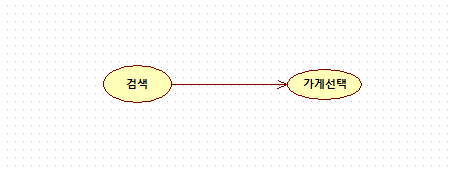
**2.2.8 가게 리뷰 및 별점 열람**

****

[그림. 8] 가게 리뷰 및 별점 열람 흐름도

|  |  |
| --- | --- |
| **Use Case#007** | |
| **General Characteristics** | |
| 요약 | 사용자가 음식점을 선택한 후, 해당 가게의 리뷰와 별점을 열람한다. |
| 작성자 | 김관식 |
| 사전 조건 | 사용자가 로그인되어 있고, 선택된 가게에 리뷰 데이터가 등록되어 있어야 한다. |
| 사후 조건 | 사용자에게 해당 가게의 리뷰 목록과 평균 별점이 제공된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 음식점 목록 또는 지도에서 가게를 선택한다. |
| 2 | 가게 상세 정보 화면에서 “리뷰 및 별점 보기” 버튼을 클릭한다. |
| 3 | 시스템은 해당 가게의 리뷰 목록과 평균 별점을 사용자에게 제공한다. |
| **대안 흐름** | |
| 1 | 해당 가게에 리뷰가 없는 경우 |
| 1-1 | “아직 등록된 리뷰가 없습니다”라는 메시지를 출력한다. |

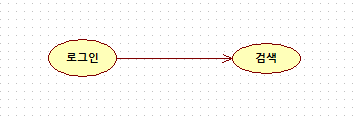
**2.2.9 음식점 검색 및 선택**

****

[그림. 9] 음식점 검색 및 선택 흐름도

|  |  |
| --- | --- |
| **Use Case#008** | |
| **General Characteristics** | |
| 요약 | 사용자가 검색 기능을 이용해 음식점을 찾고, 원하는 가게를 선택한다. |
| 작성자 | 김관식 |
| 사전 조건 | 사용자가 로그인되어 있고, 검색 가능한 음식점 데이터가 시스템에 등록되어 있어야 한다. |
| 사후 조건 | 사용자가 선택한 가게의 상세 정보로 이동한다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 검색창에 키워드(음식종류, 지역 등)를 입력한다. |
| 2 | 시스템이 입력한 조건에 맞는 가게 목록을 제공한다. |
| 3 | 사용자가 목록 중 하나의 가게를 선택한다. |
| 4 | 선택된 가게의 상세 정보 화면으로 이동한다. |
| **대안 흐름** | |
| 1 | 검색 결과가 없는 경우 |
| 1-1 | “검색 결과가 없습니다. 조건을 다시 확인해주세요” 메시지를 출력한다. |

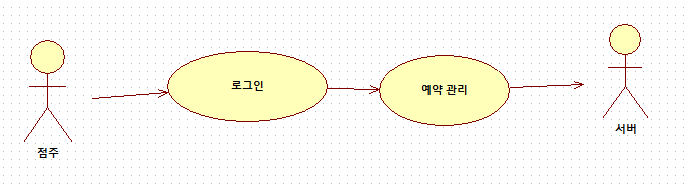
**2.2.10 로그인 후 음식점 검색**



[그림. 10] 로그인 후 검색 흐름도

|  |  |
| --- | --- |
| **Use Case#009** | |
| **General Characteristics** | |
| 요약 | 사용자가 로그인 후 검색 기능을 통해 원하는 음식점을 찾는다. |
| 작성자 | 김관식 |
| 사전 조건 | 사용자가 로그인에 성공해야 하며, 음식점 데이터가 시스템에 존재해야 한다. |
| 사후 조건 | 사용자가 조건에 맞는 음식점 목록을 확인한다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 로그인한다. |
| 2 | 메인 화면 또는 메뉴에서 검색창을 선택한다. |
| 3 | 키워드 또는 필터 조건(지역, 종류 등)을 입력한다. |
| 4 | 시스템이 해당 조건에 맞는 음식점을 목록 형태로 제공한다. |
| **대안 흐름** | |
| 1 | 로그인 실패 (아이디 또는 비밀번호 오류) |
| 2 | “로그인에 실패했습니다” 메시지를 출력하고 재시도 유도 |

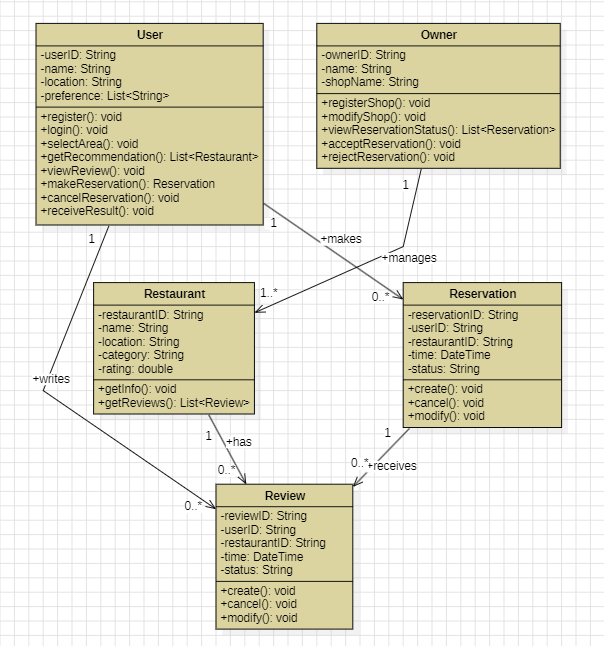
**2.2.11 예약 관리 흐름도**



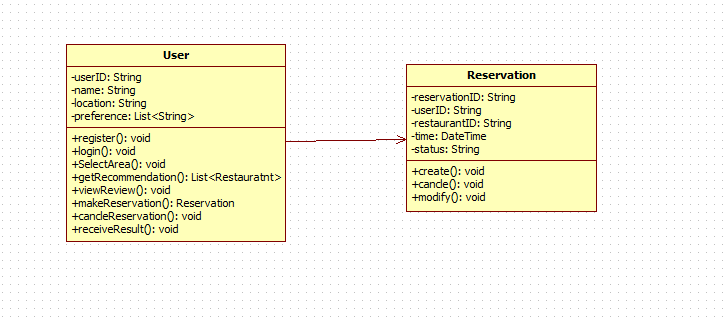
|  |  |
| --- | --- |
| **Use Case#010** | |
| **General Characteristics** | |
| 요약 | 점주가 로그인 후 예약 현황을 확인 후 수락/거절 처리를 서버에 전송한다. |
| 작성자 | 고유찬 |
| 액터 | 점주, 서버 |
| 사전 조건 | 점주가 어플을 실행한 상태 |
| 사후 조건 | 예약 처리 결과가 서버에 전송된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 점주가 로그인을 한다. |
| 2 | 점주가 본인의 가게의 예약 현황을 확인한다. |
| 3 | 예약 목록을 확인 후 수락/거절 처리를 서버에 전송한다. |
| 4 | 예약 처리 결과가 서버에 전송된다. |

**2.4 클래스 다이어그램**

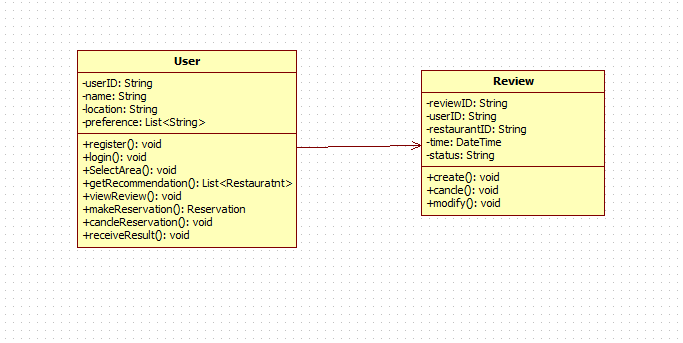
FRRA USE CASE’ 를 기반으로 전송되는 데이터 구조인 클래스 다이어그램을StarUML을 통해 작성 하였다. 또 각 기능에 대해 세분화시켜서 세분화된 클래스 다이어그램을 나누어보았다.



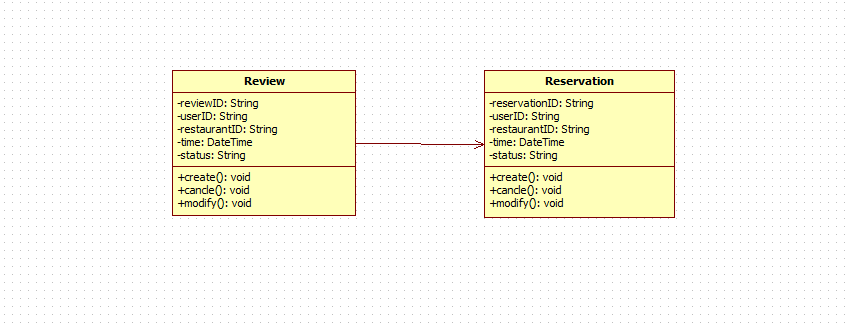
**2.4.1 사용자 예약**

****

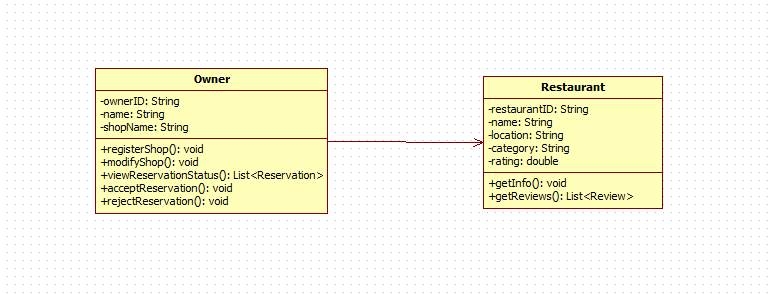
**2.4.2 사용자 리뷰**

****

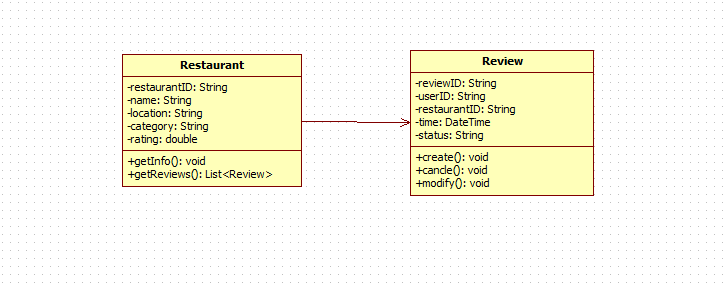
**2.4.3 리뷰 및 예약**

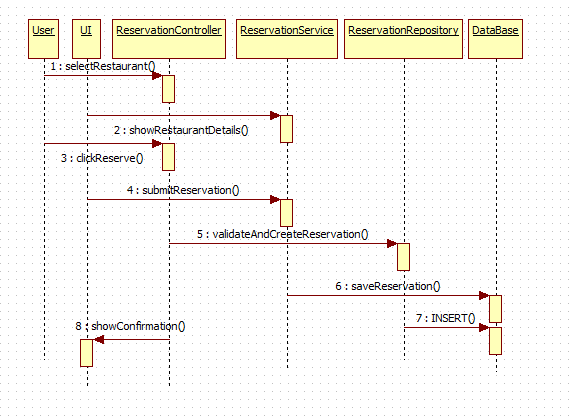
****

**2.4.4 점주 예약 관리**

****

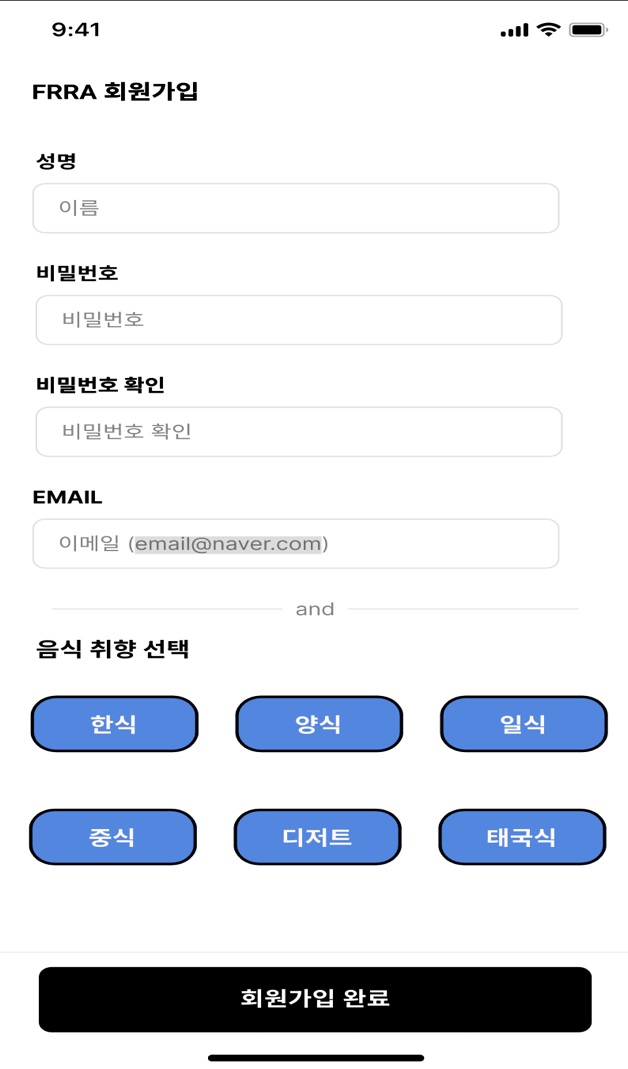
**2.4.5 식당 리뷰**

****

**2.5 시퀀스 다이어그램**

1. **식당 선택**  
   사용자는 예약하고자 하는 식당을 선택한다.  
   이 동작은 UI를 통해 전달되며, ReservationController 객체가 생성된다.
2. **식당 상세 정보 확인**  
   ReservationController는 선택된 식당의 상세 정보를 조회하고, UI에 전달한다.  
   UI는 사용자에게 해당 식당의 상세 정보를 화면에 출력한다.
3. **예약 버튼 클릭**  
   사용자는 화면에 표시된 식당 정보에서 ‘예약’ 버튼을 클릭한다.  
   이 동작은 UI를 통해 ReservationController로 전달된다.
4. **예약 정보 제출**  
   ReservationController는 예약 요청을 수신하고, ReservationService 객체에 예약 정보를 전달한다.  
   이때 사용자가 입력한 정보가 함께 포함된다.
5. **예약 유효성 검사 및 객체 생성**  
   ReservationService는 전달받은 예약 정보를 바탕으로 유효성을 검사한다.  
   예약이 유효한 경우, 새로운 예약 객체를 생성한다.
6. **예약 정보 저장 요청**  
   ReservationService는 ReservationRepository에 생성된 예약 객체를 전달하여 데이터 저장을 요청한다.
7. **DB에 예약 정보 저장**  
   ReservationRepository는 데이터베이스에 연결하여 예약 정보를 INSERT 쿼리로 저장한다.  
   데이터베이스는 저장 성공 여부를 반환한다.
8. **예약 완료 메시지 출력**  
   모든 절차가 정상적으로 완료되면 UI는 사용자에게 “예약이 완료되었습니다”와 같은 확인 메시지를 출력한다.  
   사용자는 해당 메시지를 통해 예약이 성공적으로 처리되었음을 확인한다.

3. UI 화면구성



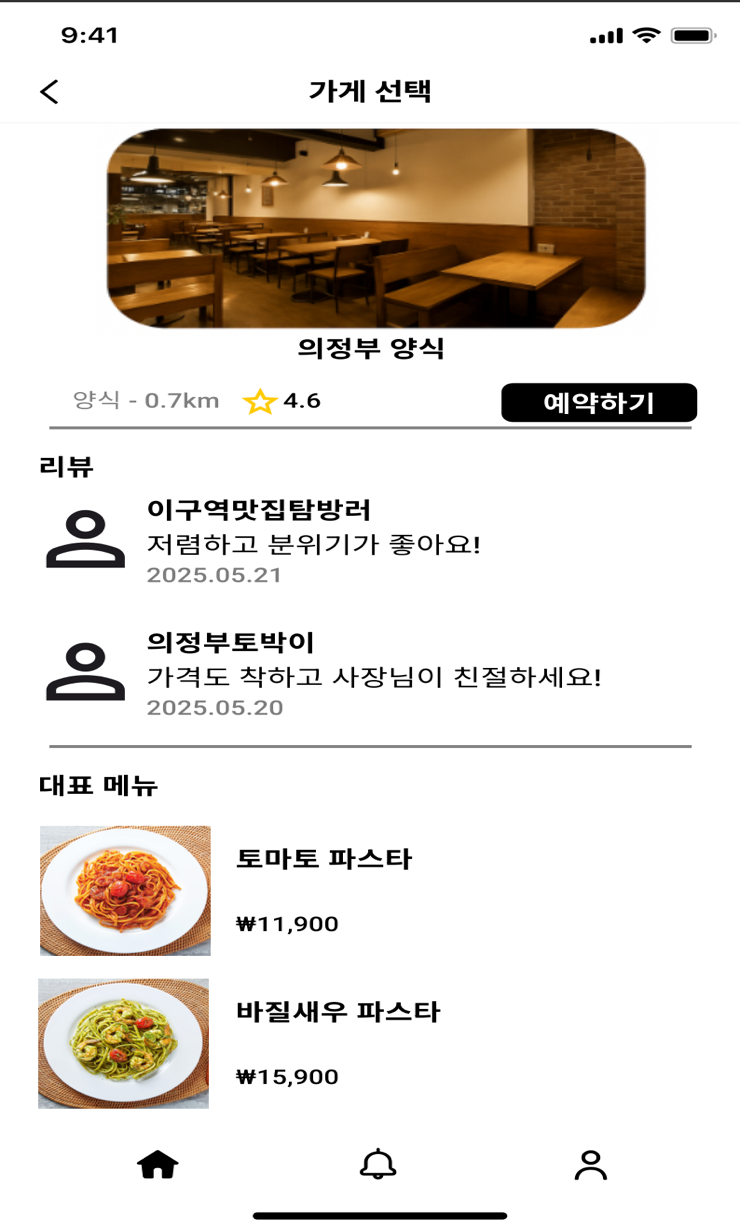
**3.1. 사용자 등록 및 로그인기능**

사용자는 이메일, 이름, 비밀번호를 입력하여 회원가입을 진행할 수 있다.  
비밀번호는 확인 절차를 통해 동일성을 검증하며, 가입 완료 후 음식 취향 설정 단계로 이동한다.  
이미 가입된 사용자는 로그인 기능을 통해 계정에 접근할 수 있다.

****

**3.2 음식 취향 선택 및 지역 기반 추천 설정**

회원가입을 완료한 사용자는 음식 취향(한식, 양식, 일식 등)을 선택하고, 선호 지역(예: 의정부시)을 지정하여 맞춤형 음식점 추천을 받을 수 있다.

****

**3.3 음식점 상세 조회 및 예약**

선택된 음식점의 상세 정보를 보여주며, 대표 메뉴와 사용자 리뷰를 함께 제공한다.  
사용자는 해당 화면에서 “예약하기” 버튼을 클릭하여 예약을 진행할 수 있다.

****

**3.4 가게 등록 및 대표 메뉴 설정**

음식점 운영자는 가게의 정보를 등록할 수 있다.  
사진, 상호명, 주소, 사업자 번호, 대표 메뉴 등을 입력하면 DB에 음식점 정보가 저장된다.