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 |
|  |  |  |
|  |  |  |
|  |  |  |

# 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.

비기능 조건

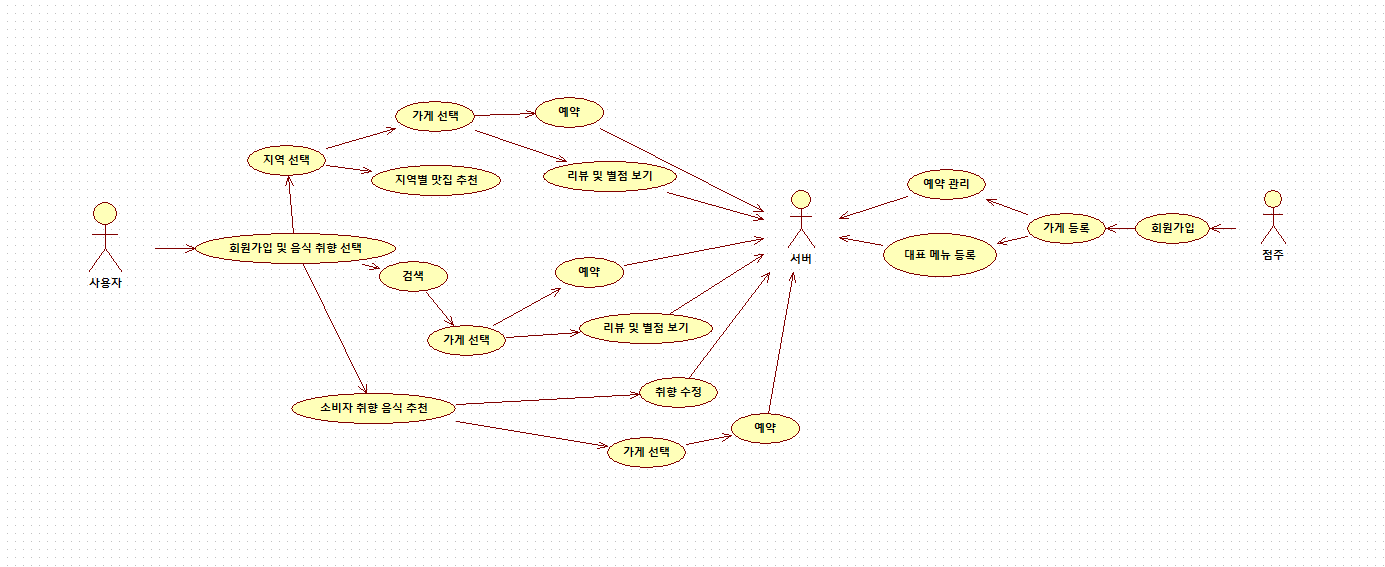
1. UI는 전 연령층이 쉽게 사용할 수 있도록 직관적으로 설계해야 한다.

2. 리뷰 작성 시 욕설 필터링이 되어야 한다.

3. 사용자의 개인정보는 암호화되어 저장되어야 한다.

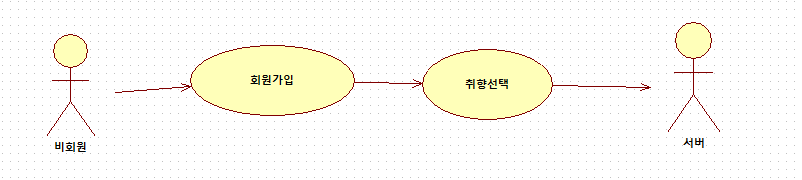
**2.2 유스케이스**

**2.2.1 UML 전체 흐름도**



[그림. 1] 전체 유스케이스 흐름도

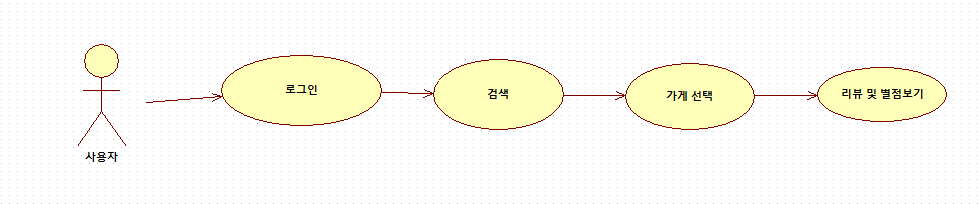
**2.2.2 비회원 정보 전송**



[그림. 2] 회원가입 흐름도

|  |  |
| --- | --- |
| **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. |

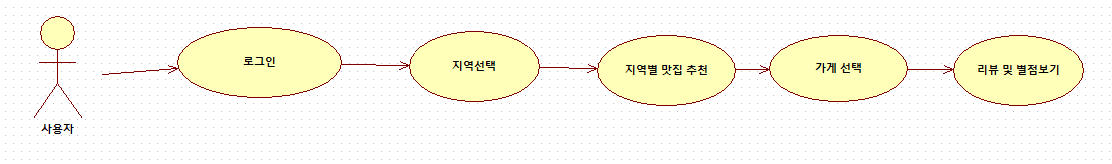
**2.2.3 사용자 리뷰 별점 보기 흐름도**



[그림. 3] 리뷰 및 별점 보기 흐름도 1

|  |  |
| --- | --- |
| **Use Case#002** | |
| **General Characteristics** | |
| 요약 | 사용자가 로그인 후 검색을 통해 리뷰와 별점을 확인한다. |
| 작성자 | 고유찬 |
| 액터 | 사용자 |
| 사전 조건 | 사용자가 어플에 로그인되어 있어야 한다. |
| 사후 조건 | 사용자에게 리뷰 및 별점 정보가 제공된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 앱에 로그인한다. |
| 2 | 음식점을 검색 및 선택한다. |
| 3 | 해당 음식점의 리뷰 보기 버튼을 클릭한다. |
| **대안 흐름** | |
| 1 | 사용자가 로그인하지 않은 상태로 리뷰를 보려고 할 경우 |
| 1-1 | 로그인 창으로 전환되며 로그인 후 다시 리뷰 화면으로 이동한다. |

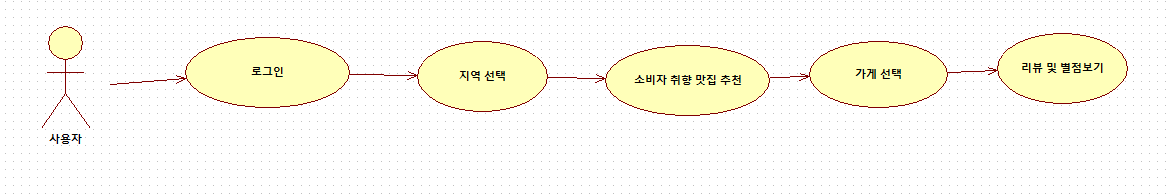
**2.2.4 지역별 맛집 추천과 리뷰 및 별점 보기 흐름도**



[그림. 4] 리뷰 및 별점 보기 흐름도 2

|  |  |
| --- | --- |
| **Use Case#003** | |
| **General Characteristics** | |
| 요약 | 사용자가 로그인 후 지역별 추천 음식점에 대한 리뷰와 별점을 확인한다. |
| 작성자 | 고유찬 |
| 액터 | 사용자 |
| 사전 조건 | 사용자가 어플에 로그인되어 있어야 한다. |
| 사후 조건 | 사용자에게 지역별 추천 맛집의 리뷰 및 별점 정보가 제공된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 앱에 로그인한다. |
| 2 | 사용자가 선택한 지역을 기반으로 맛집을 추천한다. |
| 3 | 사용자가 추천된 목록에서 가게를 선택한다. |
| 4 | 선택 된 가게의 리뷰 및 별점을 사용자에게 전송한다. |
| **대안 흐름** | |
| 1 | 사용자가 로그인하지 않은 상태로 리뷰를 보려고 할 경우 |
| 1-1 | 로그인 창으로 전환되며 로그인 후 다시 리뷰 화면으로 이동한다. |

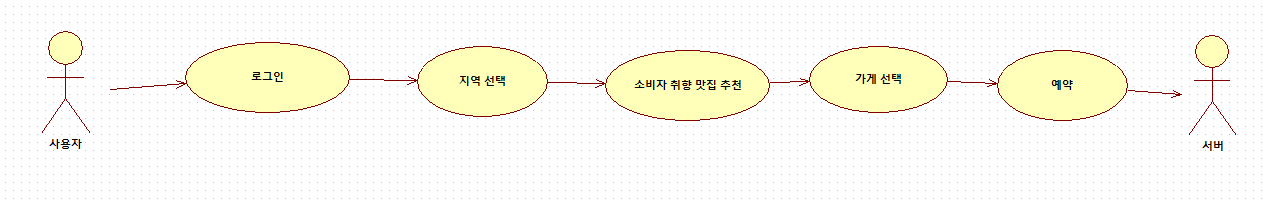
**2.2.5 사용자 취향을 토대로 가게 추천과 리뷰 및 별점 보기 흐름도**



[그림. 5] 리뷰 및 별점 보기 흐름도 3

|  |  |
| --- | --- |
| **Use Case#004** | |
| **General Characteristics** | |
| 요약 | 사용자가 로그인 후 취향 음식 추천 기능을 통해 맛집을 추천해주고 리뷰 및 별점을 확인한다. |
| 작성자 | 고유찬 |
| 액터 | 사용자 |
| 사전 조건 | 사용자가 어플에 로그인되어 있어야 한다. |
| 사후 조건 | 사용자 취향 음식을 바탕으로 음식점의 리뷰 및 별점 정보가 제공된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 어플에 로그인 및 지역을 선택한다. |
| 2 | 사용자가 선택한 음식 취향을 기반으로 맛집을 추천한다. |
| 3 | 사용자가 추천된 목록에서 가게를 선택한다. |
| 4 | 선택 된 가게의 리뷰 및 별점을 사용자에게 전송한다. |
| **대안 흐름** | |
| 1 | 사용자가 로그인하지 않은 상태로 리뷰를 보려고 할 경우 |
| 1-1 | 로그인 창으로 전환되며 로그인 후 다시 리뷰 화면으로 이동한다. |
| 2 | 사용자가 설정한 취향 음식의 정보가 부족해 추천이 어려운 경우 |
| 2-1 | “맛집 추천을 위한 취향 음식 정보가 더 필요합니다.”라는 메시지 출력 및 설정 화면으로 이동한다. |
| 3 | 사용자가 설정한 취향 음식과 관련된 맛집이 주변에 없을 경우 |
| 3-1 | “설정하신 취향 음식과 관련된 음식점이 주위에 없어요.”라는 메시지를 출력 |

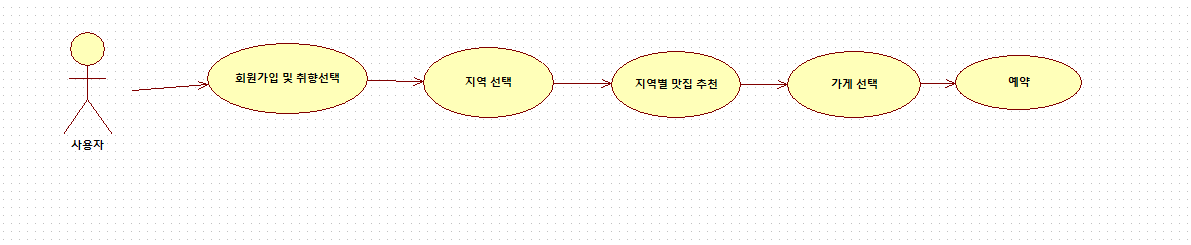
**2.2.6 사용자 취향을 토대로 가게 추천 및 예약 흐름도**



**[그림. 6] 예약 흐름도 1**

|  |  |
| --- | --- |
| **Use Case#005** | |
| **General Characteristics** | |
| 요약 | 사용자가 로그인 후 사용자 음식 취향에 맞는 맛집 추천을 받은 뒤, 선택한 음식점을 예약한다. |
| 작성자 | 고유찬 |
| 액터 | 사용자, 서버 |
| 사전 조건 | 사용자가 어플에 로그인되어 있어야 한다. |
| 사후 조건 | 서버에 예약 정보가 전송된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 어플에 로그인 및 지역을 선택한다. |
| 2 | 사용자에게 사용자 취향 음식에 관련된 주변 맛집을 추천한다. |
| 3 | 사용자가 추천된 목록에서 가게를 선택한다. |
| 4 | 예약을 눌러 서버에 예약정보를 전송한다. |
| **대안 흐름** | |
| 1 | 사용자가 로그인하지 않은 상태로 예약을 하려는 경우 |
| 1-1 | 로그인 창으로 전환되며 로그인 후 다시 예약 화면으로 이동한다. |
| 2 | 사용자가 설정한 정보가 예약에 필요한 정보에 비해 부족한 경우 |
| 2-1 | 회원 수정으로 이동한 후 추가적인 정보를 작성하게끔 유도한다. |

**2.2.7지역별 맛집 추천 및 예약 흐름도(수정 필요)**

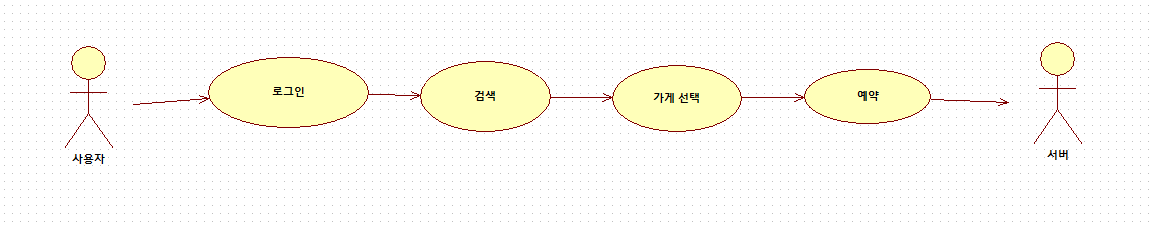


[그림. 7] 예약 흐름도 2

|  |  |
| --- | --- |
| **Use Case#006** | |
| **General Characteristics** | |
| 요약 | 사용자가 로그인 후 지역별 맛집 추천을 받은 뒤, 선택한 음식점을 예약한다. |
| 작성자 | 고유찬 |
| 액터 | 사용자, 서버 |
| 사전 조건 | 사용자가 어플에 로그인되어 있어야 한다. |
| 사후 조건 | 서버에 예약 정보가 전송된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 어플에 로그인 및 지역을 선택한다. |
| 2 | 사용자에게 선택된 지역의 맛집을 추천한다. |
| 3 | 사용자가 추천된 목록에서 가게를 선택한다. |
| 4 | 예약을 눌러 서버에 예약정보를 전송한다. |
| **대안 흐름** | |
| 1 | 사용자가 로그인하지 않은 상태로 예약을 하려는 경우 |
| 1-1 | 로그인 창으로 전환되며 로그인 후 다시 예약 화면으로 이동한다. |
| 2 | 사용자가 설정한 정보가 예약에 필요한 정보에 비해 부족한 경우 |

|  |  |
| --- | --- |
| 2-1 | 회원 수정으로 이동한 후 추가적인 정보를 작성하게끔 유도한다. |

**2.2.8 음식점 검색 및 예약 흐름도**

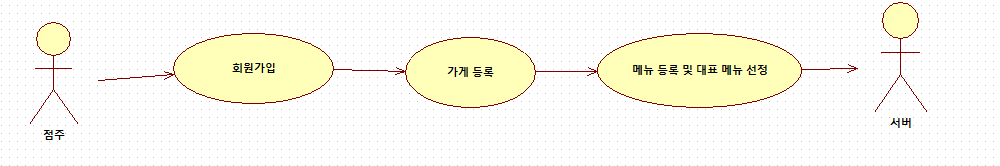


[그림. 8] 예약 흐름도 3

|  |  |
| --- | --- |
| **Use Case#007** | |
| **General Characteristics** | |
| 요약 | 사용자가 음식점을 검색한 후 선택 후 해당 음식점을 예약한다. |
| 작성자 | 고유찬 |
| 액터 | 사용자, 서버 |
| 사전 조건 | 사용자가 어플에 로그인되어 있어야 한다. |
| 사후 조건 | 서버에 예약 정보가 전송된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 사용자가 어플에 로그인한다. |
| 2 | 사용자가 원하는 음식점을 검색한다. |
| 3 | 사용자가 해당 음식점을 선택한다. |
| 4 | 예약을 눌러 서버에 예약정보를 전송한다. |
| **대안 흐름** | |
| 1 | 사용자가 로그인하지 않은 상태로 예약을 하려는 경우 |
| 1-1 | 로그인 창으로 전환되며 로그인 후 다시 예약 화면으로 이동한다. |
| 2 | 사용자가 설정한 정보가 예약에 필요한 정보에 비해 부족한 경우 |

|  |  |
| --- | --- |
| 2-1 | 회원 수정으로 이동한 후 추가적인 정보를 작성하게끔 유도한다. |
| 3 | 검색 결과가 없는 경우 |
| 3-1 | “검색 결과가 없습니다”라는 메시지 출력 및 검색창 재출력 |

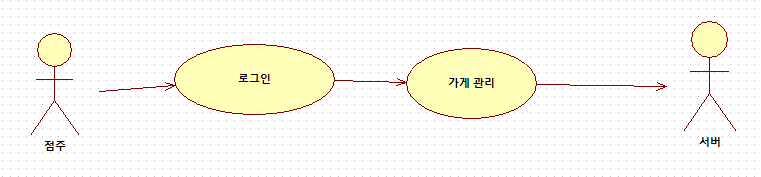
**2.2.9 가게 등록 흐름도**



[그림. 9] 가게 등록 흐름도

|  |  |
| --- | --- |
| **Use Case#008** | |
| **General Characteristics** | |
| 요약 | 점주가 회원가입 후 가게정보를 등록하고 해당 가게 정보를 서버에 전송한다. |
| 작성자 | 고유찬 |
| 액터 | 점주, 서버 |
| 사전 조건 | 점주가 회원가입하고 가게를 등록한다. |
| 사후 조건 | 가게 정보가 서버에 전송된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 점주가 회원가입을 눌러 진행한다. |
| 2 | 점주가 본인의 가게 정보 및 대표메뉴를 입력한다. |
| 3 | 입력된 가게 정보(주소, 가게명, 대표 메뉴 등)를 서버에 전송한다. |
| 4 | 서버가 가게 정보를 저장 및 관리한다. |
| **대안 흐름** | |
| 1 | 가게 정보 일부가 입력되지 않은 경우 |
| 1-1 | “가게 등록에 필요한 모든 필수 정보를 입력해주세요”라는 메시지 출력 후 등록 불가 처리 |

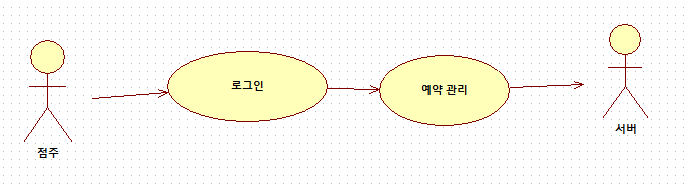
**2.2.10 가게 관리 흐름도**



[그림. 10] 가게 관리 흐름도

|  |  |
| --- | --- |
| **Use Case#009** | |
| **General Characteristics** | |
| 요약 | 점주가 로그인 후 본인의 가게 정보를 수정하며 해당 정보를 서버에 전송한다. |
| 작성자 | 고유찬 |
| 액터 | 점주, 서버 |
| 사전 조건 | 관리 및 수정하려는 가게가 사전에 서버에 등록되어 있어야한다. |
| 사후 조건 | 수정된 가게 정보가 서버에 전송된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 점주가 로그인을 한다. |
| 2 | 점주가 본인의 가게 정보를 조회 및 수정한다. |
| 3 | 수정된 가게 정보(주소, 가게명, 대표 메뉴 등)를 서버에 전송한다. |
| 4 | 서버가 가게 정보를 저장 및 관리한다. |
| **대안 흐름** | |
| 없음 |  |

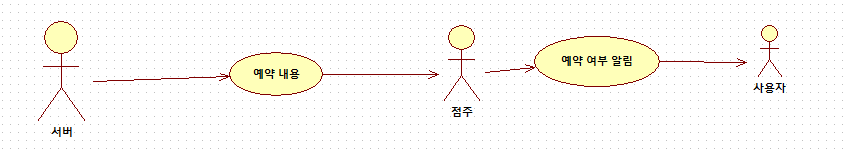
**2.2.11 예약 관리 흐름도**



[그림. 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. |
| **대안 흐름** | |
| 없음 |  |

**2.2.12 예약 여부 전송 흐름도**

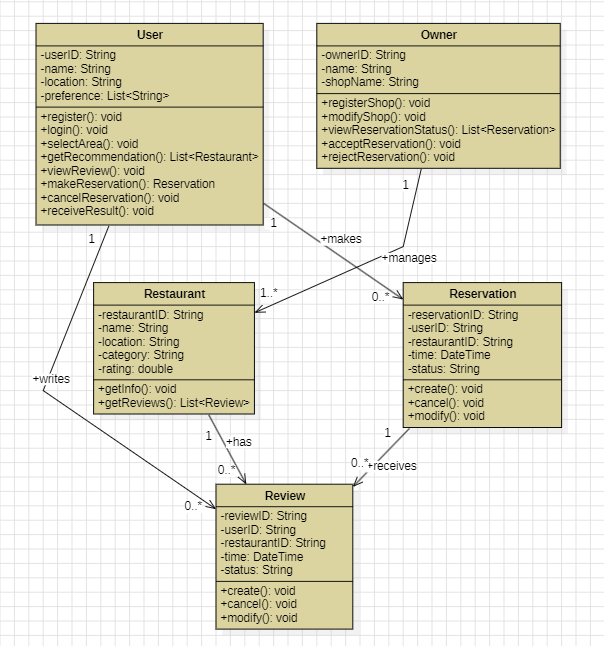


[그림. 12] 예약 여부 알림 흐름도

|  |  |
| --- | --- |
| **Use Case#011** | |
| **General Characteristics** | |
| 요약 | 서버가 점주에게 예약한 고객 정보를 전송하고, 점주는 예약 여부 알림을 사용자에게 전송한다. |
| 작성자 | 고유찬 |
| 액터 | 서버, 점주, 사용자 |
| 사전 조건 | 서버에 사용자 예약 요청이 전송된 경우  점주가 어플을 실행한 상태  사용자가 어플을 실행한 상태 |
| 사후 조건 | 예약 처리 결과가 사용자에게 전송된다. |
| **기본 흐름** | |
| 단계 | 행위 |
| 1 | 서버가 에약 정보를 점주에게 전송한다. |
| 2 | 점주는 해당 내용을 확인 후 예약 여부 처리를 한다. |
| 3 | 해당 예약 여부 결과를 사용자에게 전송한다. |
| **대안 흐름** | |
| 없음 |  |

**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.



**2.5 시퀀스 다이어그램**

3. UI 화면구성