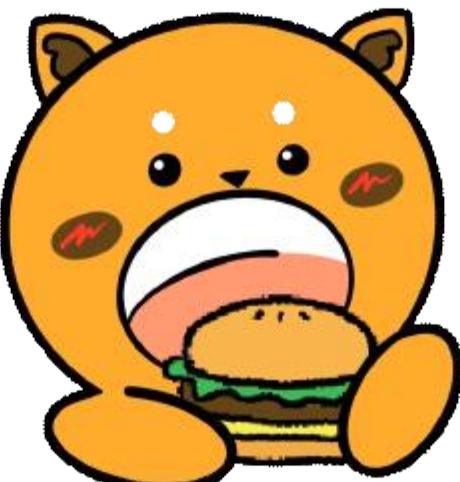


2. SDS (Software Design Specification)

DGAJ



DGAJ

Student No	Name
21912113	송재민
21912085	김현수
22012114	김나연
22012079	구교영
22010642	홍아랑
21912075	박해세

[Revision history]

Revision date	Version #	Description	Author
11/03/2023	1.00	DGAJ - SDS 작성	테크자이언트

= Contents =

1. Introduction	1
2. Use case analysis	2
3. Class diagram	23
4. Sequence diagram	52
5. State machine diagram	86
6. User interface prototype	88
7. Implementation requirements	108
8. Glossary	109
9. References	110

= Authors for each section =

Introduction - 구교영

Use case analysis - 구교영

Class diagram - 김나연, 박해세

Sequence diagram - 흥아랑, 송재민

State machine diagram - 김현수

User interface prototype - 김현수

Implementation requirements - 김현수

Glossary - 김현수

References - 김현수

1. Introduction

본 문서는 테크자이언트¹⁾가 개발하고자 하는 시스템인 DGAJ의 design specification(S DS)이다. 이 문서는 기존의 SRS에서 정의한 요구사항을 기능적으로 구분하여 효과적으로 구현하기 위해 시스템을 여러 관점에서 설계한다.

DGAJ는 현대 사회에서 발생하는 여러 문제에 도전하고, 해결책을 제공하는 혁신적인 안드로이드 애플리케이션이다. 이 앱은 사용자들이 별도의 광고 없이 진정으로 원하는 음식점을 찾을 수 있도록 돋는 동시에, 소상공인들에게 맛과 서비스를 통한 공정한 경쟁 기회를 제공한다. 또한, 음식점에서 편리한 대기 서비스를 제공하고 남은 재료들을 기부 할 수 있는 기능을 통해 사회•경제적 가치 창출을 기대할 수 있다. 이러한 기대는 현대 사회에서 다양한 관계자들에게 많은 이점으로 작용할 것이 예상된다.

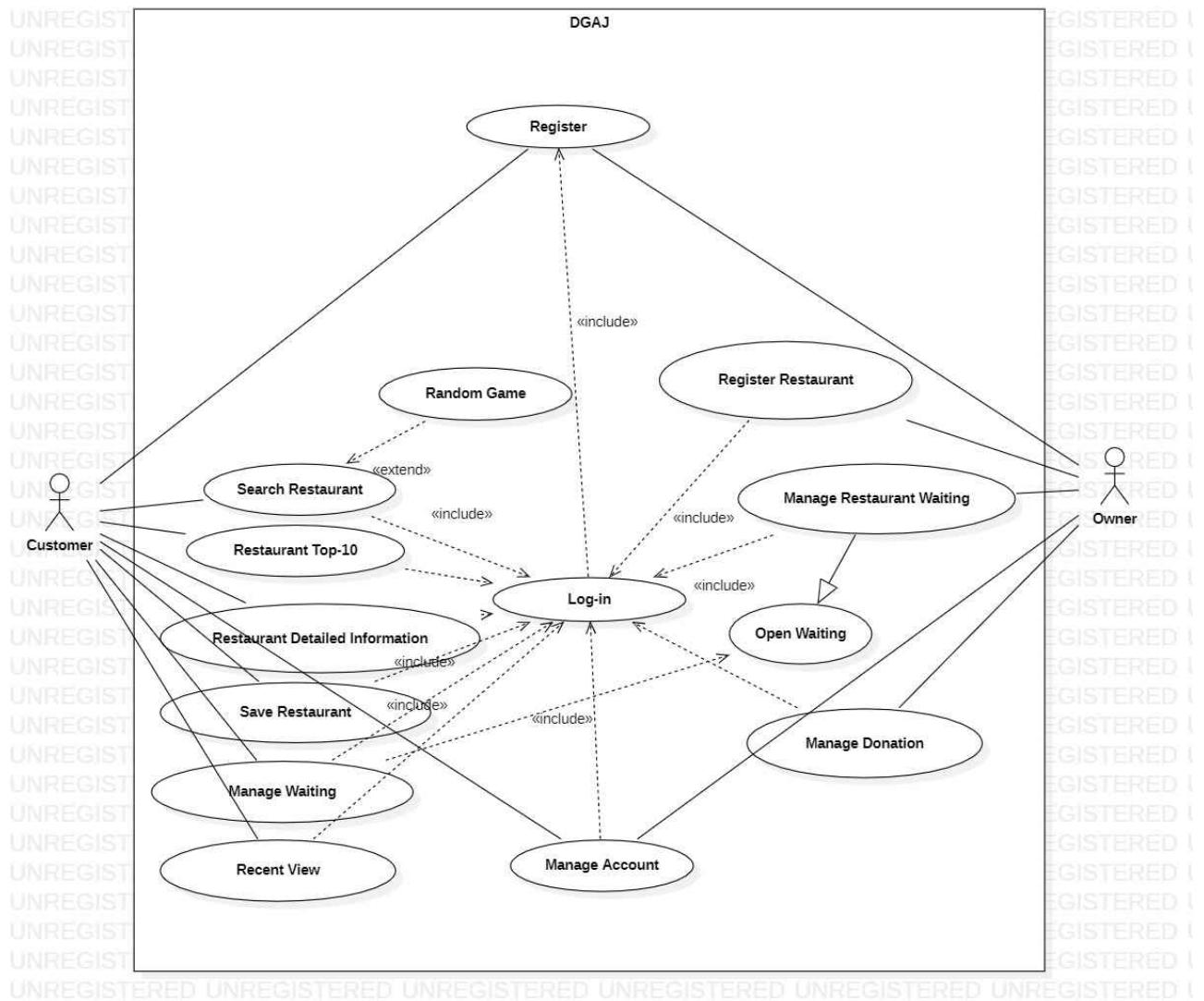
본 문서에서는 Use case analysis, Class diagram, Sequence diagram, State machine diagram, User interface prototype을 이용하여 DGAJ의 기능과 특징을 효과적으로 설명한다. Use case analysis는 사용자 관점에서 소프트웨어가 제공하는 기능을 서술하는 단계이고, Use case diagram은 사용자와 사용자가 사용할 Use case 간의 상호작용을 보여주는 자료이다. DGAJ는 고객과 사업자로 두 가지의 Actor가 존재한다. Class diagram은 DGAJ의 클래스와 속성, 동작 방식 등 구조적 양상을 표현하는 자료이다. Sequence diagram은 각 기능의 동작 순서와 상호작용하는 객체 간의 관계를 표현하고, State machine diagram 특정 기능 상태에 따라 변경되는 모습을 도식화한 자료이다. User interface는 사용자의 인터페이스 관점으로 DGAJ를 서술한 것이다.

본 문서는 DGAJ의 개발자를 비롯한 모든 Stakeholder를 위한 문서이며, DGAJ 시스템의 전반적인 구조와 설계 과정에 대한 이해도를 높이는 것에 도움을 준다.

1) 본 프로젝트를 진행하는 개발팀의 명칭

2. Use case analysis

Use case analysis에서는 use case diagram과 use case description을 통해 시스템의 기능을 상세하게 설명한다.



[그림 2] Use case diagram

Use case #1 : Register	
GENERAL CHARACTERISTICS	
Summary	DGAJ의 모든 사용자는 시스템을 이용하기 위해서 회원가입을 통해 계정을 등록해야 한다.
Scope	DGAJ(Daegu Go And Joy)
Level	Customer, Owner
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Customer, Owner
Preconditions	사용자는 안드로이드 앱에서 DGAJ 앱을 설치한 상태이다.
Trigger	사용자가 첫 화면에서 회원가입 버튼을 누를 때
Success Post Condition	DB(Firebase)에 사용자의 계정이 등록되고 사용자는 로그인을 진행할 수 있다.
Failed Post Condition	사용자는 로그인을 진행할 수 없고 시스템을 사용할 수 없다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 회원가입을 한다.
1	사용자가 로그인 화면에서 회원가입 버튼을 누른다.
2	회원가입 화면에서 사용자는 사용자 태입, 아이디, 비밀번호, 전화번호를 입력한다.
3	사용자는 아이디 체크 버튼으로 입력한 아이디가 중복되지 않은 아이디인지 확인한 후, 등록 버튼을 누른다.
4	시스템은 회원가입이 성공한 것인지 판단한다.
5	이 Use case는 회원가입이 성공하면 끝난다.
EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 입력한 아이디가 이미 존재하는 아이디이다. ...3a1. 입력한 아이디가 중복된 아이디라는 메세지를 Toast창으로 보여준다. ...3a2. 아이디를 다시 입력하는 단계로 돌아간다. (Use case #1-2)
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사용자당 1회
<Concurrency>	제한 없음
Due Date	2023.11.03.

Use case #2 : Log-in	
GENERAL CHARACTERISTICS	
Summary	DGAJ의 모든 사용자는 시스템을 사용하기에 앞서 로그인을 해야 한다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Customer, Owner
Preconditions	사용자는 회원가입을 모두 마친 상태이다.
Trigger	로딩 화면 이후 로그인 화면이 출력될 때
Success Post Condition	사용자는 고객과 사업자로 나뉘어 각각의 시스템 기능을 이용할 수 있다.
Failed Post Condition	사용자는 로그인에 실패하여 DGAJ 앱의 기능을 사용할 수 없다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 로그인을 한다.
1	사용자가 로그인 화면에서 아이디와 비밀번호를 입력한다.
2	로그인 정보를 모두 입력한 후, 로그인 버튼을 누른다.
3	시스템은 로그인이 성공한지 판단한다.
4	이 Use case는 로그인이 성공하면 끝난다.
EXTENSION SCENARIOS	
Step	Branching Action
2	2a. 입력한 아이디나 비밀번호의 오류로 로그인에 실패한다. ...2a1. 로그인 실패 메시지를 Toast창으로 보여준다. ...2a2. 아이디와 비밀번호를 다시 입력하는 단계로 돌아간다. (Use case #2-1)
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	시스템을 새로 시작할 때마다
<Concurrency>	제한 없음
Due Date	2023.11.03.

Use case #3 : Search restaurant
GENERAL CHARACTERISTICS

Summary	이 use case는 고객을 위한 기능으로 음식점을 검색하기 위한 검색어를 결정한다. 검색어는 메뉴 검색어와 위치 검색어로 나뉜다. 메뉴 검색어는 고객이 원하는 메뉴를 설정하거나 Use Case #4 Random Game을 통해 결정되고 위치 검색어는 GPS를 활용한 현재 위치나 고객이 설정한 위치로 결정된다. API를 사용하여 설정된 검색어를 조회한다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Customer
Preconditions	고객은 고객 유형으로 로그인을 마친 상태이다.
Trigger	고객 유형의 아이디로 로그인을 완료하고 음식점 검색 기능을 이용하려 할 때
Success Post Condition	설정된 검색어에 부합하는 음식점을 API를 통해 검색한다.
Failed Post Condition	음식점 검색에 실패하고 화면이 전환되지 않는다.

MAIN SUCCESS SCENARIO

Step	Action
S	고객이 원하는 음식 메뉴 및 위치에 적합한 음식점을 검색한다.
1	고객이 음식점을 검색하는 화면에서 위치 검색어를 입력한다.
2	고객이 음식점을 검색하는 화면에서 메뉴 검색어를 입력한다.
3	고객은 두 가지의 검색어를 모두 입력하고 검색 버튼을 누른다.
4	시스템은 API를 통해 설정한 검색어에 부합하는 음식점을 조회한다.
5	시스템은 음식점 검색이 성공하였는지 판단한다.
6	이 Use case는 음식점 검색에 성공하면 끝난다.

EXTENSION SCENARIOS	
Step	Branching Action
1	<p>1a. 위치 검색어 직접 입력</p> <p>…1a1. 고객이 원하는 위치를 설정하기 위해 검색창에 정보를 직접 입력한다.</p> <p>…1a2. 시스템은 입력된 검색어를 화면의 검색창에 보여준다.</p>
1	<p>1b 위치 검색어 자동 입력</p> <p>…1b1. 위치 검색어 우측의 현재 위치 버튼을 누른다.</p> <p>…1b2. 고객의 현재 위치를 GPS로 파악하여 위치 검색어에 입력한다.</p> <p>…1b3. 입력된 검색어를 화면의 검색창에 보여준다.</p>
2	<p>2a. 메뉴 검색어 직접 입력</p> <p>…2a1. 고객이 원하는 메뉴를 설정하기 위해 검색창에 정보를 직접 입력한다.</p> <p>…2a2. 시스템은 입력된 검색어를 화면의 검색창에 보여준다.</p>
2	<p>2b 메뉴 랜덤 기능으로 메뉴 검색어 자동 설정</p> <p>…2b1. 고객은 화면에서 랜덤 버튼을 누른다.</p> <p>…2b2. 시스템은 설정된 메뉴 중에서 하나를 랜덤으로 선택하여 검색어에 입력한다.</p> <p>…2b3. 시스템은 입력된 검색어를 화면의 검색창에 보여준다.</p>
2	<p>2c 메뉴 고르기 게임 기능으로 메뉴 정하기</p> <p>…2c1. 고객이 화면에서 메뉴 정하기 버튼을 누른다.</p> <p>…2c2. 게임 화면으로 전환된 후, 고객은 룰렛 돌리기와 제비뽑기 중 하나의 게임을 선택해 해당 버튼을 누른다.</p> <p>…2c3. 고객은 게임을 통해 메뉴를 정한다.</p> <p>…2c4. 고객이 홈 버튼을 누르면 시스템은 음식점 검색 화면으로 이동한다.</p>
3	<p>3a. 위치 검색어나 메뉴 검색어 공백 처리</p> <p>…3a1. 시스템은 공백인 위치 검색어를 GPS로 파악한 고객의 현재 위치 정보로 대체한다.</p> <p>…3a2. 시스템은 공백인 메뉴 검색어를 모든 메뉴에 대한 검색으로 대체한다.</p>
RELATED INFORMATION	
Performance	≤ 1 seconds
Frequency	고객당 하루 평균 5회 이상
<Concurrency>	제한 없음
Due Date	2023.11.03.

Use case #4 : Random game
GENERAL CHARACTERISTICS

Summary	이 use case는 고객이 메뉴 검색어를 입력하기에 앞서 메뉴 설정에 도움을 주는 기능을 설명한다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Customer
Preconditions	고객은 고객 유형으로 로그인을 마치고, 음식점 검색 화면이 제공되었다.
Trigger	고객이 메뉴 설정을 위한 게임을 실행하려고 할 때
Success Post Condition	게임이 실행되고 고객은 게임을 통해 메뉴를 결정한다.
Failed Post Condition	메뉴 정하기 게임 화면이 출력되지 않는다.

MAIN SUCCESS SCENARIO

Step	Action
S	고객이 메뉴를 결정하기 위해 게임을 실행한다.
1	고객이 음식점을 검색하는 화면에서 메뉴 정하기 버튼을 누른다.
2	고객이 제비뽑기 버튼을 누른다.
3	고객이 6가지 후보 메뉴를 입력하고 게임하기 버튼을 누른다.
4	시스템은 6가지 중 한 가지 메뉴를 무작위 출력한다.
5	고객이 랜덤돌리기 버튼을 누른다.
6	고객이 6가지 후보 메뉴를 입력하고 게임하기 버튼을 누른다.
7	시스템은 6가지 중 한 가지 메뉴를 무작위 출력한다.

EXTENSION SCENARIOS

Step	Branching Action

RELATED INFORMATION	
Performance	≤ 1 seconds
Frequency	고객당 하루 평균 3회 이상
<Concurrency>	제한 없음
Due Date	2023.11.03.

Use case #5 : Restaurant Top-10
GENERAL CHARACTERISTICS

Summary	Use case #3. Search Restaurant에서 호출되는 Use case이다. 시스템은 고객이 검색한 음식점 리스트를 화면에 출력하고, 해당 음식점의 위치를 지도를 통해 나타낸다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Customer
Preconditions	고객은 음식점 검색 화면에서 검색어를 설정하여 검색 버튼을 누른 상태여야 한다.
Trigger	고객이 위치 검색어와 메뉴 검색어를 설정하여 검색한 음식점의 결과를 확인하려 할 때
Success Post Condition	화면을 통해 검색된 음식점을 확인할 수 있다.
Failed Post Condition	음식점 검색에 실패하여 올바른 음식점 정보 확인이 불가능하다.

MAIN SUCCESS SCENARIO

Step	Action
S	고객이 검색한 음식점의 결과를 확인한다.
1	시스템은 검색어에 부합한 검색어 중 10개의 음식점을 위치 검색어와 가까운 순으로 화면에 제공한다. (오늘의 가게 추천)
2	시스템은 화면에 제공된 음식점의 위치 정보를 화면 상단의 지도에 표시하여 제공한다.
3	시스템은 Use case #6 Restaurant Detailed Information 기능으로 해당 음식점의 상세정보 페이지를 제공한다.

EXTENSION SCENARIOS

Step	Branching Action
1	1a. 위치 검색어 변경 ...1a1. 고객이 화면 상단의 위치 검색창에 위치 검색어를 재설정한다. ...1a2. 시스템은 입력된 검색어를 화면의 검색창에 보여준다. ...1a3. 고객은 검색 버튼을 눌러 음식점을 재검색한다.
3	3a. 가게 상세 정보 확인 ...3a1. 고객이 음식점을 선택한다. ...3a2. 시스템은 선택된 음식점의 상세정보를 제공한다.

RELATED INFORMATION	
Performance	≤ 3 seconds
Frequency	고객당 하루 평균 5회 이상
<Concurrency>	제한 없음
Due Date	2023.11.03.

Use case #6 : Restaurant Detailed Information	
GENERAL CHARACTERISTICS	
Summary	Use case #5. Restaurant Top-10에서 호출되는 Use case이다. 고객이 검색된 음식점 중 하나를 선택하여 누르면 해당 음식점의 상세 정보 페이지를 제공한다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Customer
Preconditions	고객은 음식점 검색을 완료한 상태이다.
Trigger	고객이 검색된 음식점의 상세정보를 확인하기 위해 음식점 목록을 눌렀을 때
Success Post Condition	고객은 가게 상세정보 페이지 화면으로 가게의 상세정보를 확인한다.
Failed Post Condition	고객은 가게의 상세정보를 확인할 수 없다. 시스템은 가게 상세정보 창을 제공하지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	고객이 검색한 음식점의 상세정보를 확인한다.
1	시스템은 선택된 가게의 기본 정보 및 상세정보를 제공한다.
2	고객은 Use case #7 Save Restaurant으로 선택한 음식점을 저장 목록에 추가한다.
3	고객은 Use case #8 Manage Waiting으로 선택한 음식점에 대기자 신청을 한다.
EXTENSION SCENARIOS	
Step	Branching Action
RELATED INFORMATION	
Performance	≤ 1 seconds
Frequency	고객당 하루 평균 10회 이상
<Concurrency>	제한 없음
Due Date	2023.11.03.

Use case #7 : Save Restaurant
GENERAL CHARACTERISTICS

Summary	이 Use case는 고객이 선택한 가게를 저장 목록에 추가, 조회, 삭제하는 기능을 설명한다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Customer
Preconditions	저장목록에 저장할 가게의 상세 정보 페이지가 열린 상태이거나 나의 저장목록 화면이 열린 상태여야 한다.
Trigger	고객이 인터페이스에서 저장하기와 관련된 기능을 하려고 할 때
Success Post Condition	선택된 가게를 저장목록에 추가한다. 저장목록을 조회한다. 저장목록에서 특정 가게를 삭제한다.
Failed Post Condition	저장목록에 대한 변경 사항이 반영되지 않는다.

MAIN SUCCESS SCENARIO

Step	Action
S	고객이 저장하기와 관련된 기능을 수행하려 한다.
1	고객은 가게 상세정보 페이지에서 가게 저장하기 버튼을 누른다.
2	시스템은 나의 저장목록 화면에서 로그인 고객의 저장목록을 출력한다.
3	고객은 나의 저장목록에서 삭제하고 싶은 가게의 하트 버튼을 누른다.
4	시스템은 저장목록 삭제 확인 알림창을 출력한다.
5	고객은 삭제 확인 버튼을 누른다.

EXTENSION SCENARIOS

Step	Branching Action
4	4a. 저장 목록 삭제 취소 ...4a1. 고객은 저장 목록 삭제 확인 알림창에서 취소 버튼을 눌러 삭제를 취소한다. ...4a2. 시스템은 저장 목록 삭제 알림창을 닫는다.

RELATED INFORMATION

Performance	≤ 2 seconds
Frequency	고객당 하루 평균 7회 이상
<Concurrency>	제한 없음
Due Date	2023.11.03.

Use case #8 : Manage Waiting
GENERAL CHARACTERISTICS

Summary	이 Use case는 고객의 가게 대기와 관련된 기능인 대기 신청, 대기 현황 조회, 대기 삭제를 설명한다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Customer
Preconditions	대기 신청하고자 하는 가게의 상세정보 페이지를 열거나 사이드바에서 대기 현황 버튼을 누른다.
Trigger	고객이 인터페이스에서 대기와 관련된 기능을 하려고 할 때
Success Post Condition	선택한 가게에 대기자 신청을 한다. 대기 신청을 완료한 가게를 조회한다. 대기 신청 취소 버튼을 눌러 대기를 취소한다.
Failed Post Condition	대기에 대한 변경사항이 반영되지 않는다.

MAIN SUCCESS SCENARIO

Step	Action
S	고객이 대기와 관련된 기능을 수행하려 한다.
1	고객은 가게 상세정보 페이지에서 출서기 버튼을 누른다.
2	시스템은 대기 신청 정보를 입력할 수 있는 화면을 제공한다.
3	고객은 대기 신청 정보(전화번호, 이름, 인원수)를 기입하고 확인 버튼을 누른다.
4	시스템은 제공된 정보를 기반으로 대기자를 추가한다.
5	시스템은 대기 현황 페이지에서 현재 로그인한 고객의 대기 신청 완료 현황(대기 신청 가게, 대기 순번)을 출력한다.
6	고객은 대기 현황 페이지에서 대기 취소 버튼을 눌러서 대기 신청을 취소한다.

EXTENSION SCENARIOS

Step	Branching Action
3	3a. 대기 신청 취소 ...3a1. 고객은 대기 신청 정보를 모두 기입하고 확인 버튼을 누른다. ...3a2. 시스템은 대기 신청을 확인하는 알림창을 출력한다. ...3a3. 고객은 취소 버튼을 눌러 대기 신청을 취소한다.
6	6a. 신청 완료 대기 삭제 취소 ...6a1. 고객은 대기 현황 페이지에서 취소하고자 하는 대기의 취소 버튼을 누른다. ...6a2. 시스템은 대기 취소를 확인하는 알림창을 제공한다. ...6a3. 고객은 알림창의 취소 버튼을 눌러 대기 삭제를 취소한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	고객당 하루 평균 1회 이상
<Concurrency>	제한 없음
Due Date	2023.11.03.

Use case #9 : Recent View	
GENERAL CHARACTERISTICS	
Summary	고객이 검색한 음식점의 목록을 최신순으로 저장하고 조회, 삭제하여 관리하는 기능이다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Customer
Preconditions	고객이 상세 정보를 확인한 가게가 한 개 이상이다.
Trigger	고객이 인터페이스에서 최근 본 가게의 목록을 확인하려고 할 때
Success Post Condition	고객이 최근에 검색하여 확인 가게 정보를 최신순으로 제공한다.
Failed Post Condition	고객에게 최근 본 목록을 올바르게 제공하지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	고객이 최근 확인한 가게의 정보를 다시 찾고자 한다.
1	고객은 사이드바에서 최근 본 목록 버튼을 누른다.
2	시스템은 최근 본 목록 페이지를 제공한다.
3	고객은 최근 본 목록에서 확인하고 싶은 가게를 누른다.
4	시스템은 Use case #6 Restaurant Detailed Information 기능으로 선택한 가게의 상세 정보를 제공한다.
5	고객은 최근 본 목록의 삭제 버튼을 눌러서 선택한 가게를 목록에서 삭제한다.
EXTENSION SCENARIOS	
Step	Branching Action
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	고객당 하루 평균 2회 이상
<Concurrency>	제한 없음
Due Date	2023.11.03.

Use case #10 : Manage Account	
GENERAL CHARACTERISTICS	
Summary	이 Use case는 사용자의 기본 정보(비밀번호, 전화번호)를 변경하는 기능과 계정 탈퇴 기능을 관리한다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Customer, Owner
Preconditions	사용자는 DGAJ에 계정을 등록하고 로그인을 한 상태이다.
Trigger	사용자가 인터페이스에서 계정 정보를 변경하거나 계정을 탈퇴하려고 할 때
Success Post Condition	사용자의 계정 정보가 변경 또는 삭제된다.
Failed Post Condition	사용자 계정의 변경사항이 반영되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 계정 정보를 변경하거나 삭제하려고 한다.
1	사용자는 사이드바에서 내 계정 버튼을 누른다.
2	시스템은 내 계정의 세부 기능을 나타내는 화면을 제공한다.
3	사용자는 비밀번호 변경 버튼을 누르고 시스템은 비밀번호 변경 정보를 입력하는 화면을 제공한다.
4	사용자는 비밀번호 변경창에 변경 전 비밀번호와 변경할 비밀번호, 비밀번호 확인을 입력하고 변경 버튼을 누른다.
5	사용자는 전화번호 변경 버튼을 누르고, 시스템은 전화번호 변경 정보를 입력하는 화면을 제공한다.
6	사용자는 전화번호 변경창에 기존 전화번호를 입력하고 확인 버튼을 눌러 기존 정보가 맞는지 확인한 후, 변경 전화번호, 전화번호 확인을 입력하여 변경 버튼을 누른다.
7	사용자는 탈퇴하기 버튼을 누르고, 시스템은 탈퇴 여부를 재확인하는 알림창을 화면에 제공한다.
8	사용자는 사이드바의 다른 버튼을 눌러 다른 기능을 수행한다.

EXTENSION SCENARIOS	
Step	Branching Action
2	<p>2a. 내 계정 화면</p> <p>…2a1. 시스템은 비밀번호를 변경하기 위해 비밀번호 변경 버튼을 화면에 제공한다.</p> <p>…2a2. 시스템은 전화번호를 변경하기 위해 전화번호 변경 버튼을 화면에 제공한다.</p> <p>…2a3. 시스템은 계정을 탈퇴하기 위해 탈퇴하기 버튼을 화면에 제공한다.</p>
4	<p>4a. 비밀번호 변경 창의 모든 정보가 입력되지 않았다.</p> <p>…4a1. 시스템은 비밀번호를 변경하기 위한 정보를 입력하라는 메시지를 출력한다.</p> <p>…4a2. 비밀번호 변경 정보를 입력하는 단계로 돌아간다.</p>
6	<p>6a. 전화번호 변경 창의 모든 정보가 입력되지 않았다.</p> <p>…6a1. 시스템은 전화번호를 변경하기 위한 정보를 입력하라는 메시지를 출력한다.</p> <p>…6a2. 전화번호 변경 정보를 입력하는 단계로 돌아간다.</p>
7	<p>7a. 탈퇴하기 취소</p> <p>…7a1. 사용자는 탈퇴 기능을 확인하는 알림창에서 취소 버튼을 누른다.</p> <p>…7a2. 시스템은 알림창을 닫고 내 계정 페이지를 화면에 출력한다.</p>
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사용자당 평균 2회 이상
<Concurrency>	제한 없음
Due Date	2023.11.03.

Use case #11 : Manage Restaurant Waiting
GENERAL CHARACTERISTICS

Summary	이 Use case는 사업자가 운영하는 가게의 대기열을 생성, 삭제, 수정하여 관리하는 기능을 설명한다. Use case #8 Manage Waiting과 연결된 Use case이다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Owner
Preconditions	사용자가 사업자 유형으로 로그인을 완료한 상태이다. Use case #12 Register Restaurant 기능으로 연결된 가게가 한 개 이상이다.
Trigger	사업자가 운영하는 가게의 대기열을 관리하는 기능을 하려고 할 때
Success Post Condition	가게의 대기열을 생성한다. 대기열을 삭제한다. 대기열에 추가된 고객들을 관리한다.
Failed Post Condition	해당 가게의 대기열에 대한 변경사항이 반영되지 않는다.

MAIN SUCCESS SCENARIO

Step	Action
S	사업자가 운영하는 가게의 대기열을 관리하고자 한다.
1	시스템은 DB와 연결하여 현재 가게의 대기열 상태를 화면에 제공한다.
2	사업자는 대기열을 생성하거나 변경한다.
3	사업자가 대기열 열기 버튼을 누른다.
4	사업자가 대기자를 선택하여 누른다.
5	시스템은 선택한 대기자(고객)의 대기 정보를 제공한다.
6	사업자가 입장시키고자 하는 대기자의 입장 버튼을 누른다.
7	시스템은 선택한 대기자 정보를 대기열에서 삭제하고 변경사항을 반영한다.
8	사업자가 대기 취소하고자 하는 대기자의 삭제 버튼을 누른다.
9	시스템은 선택한 대기자 정보를 대기열에서 삭제하고 변경사항을 반영한다.
10	사업자가 화면에서 대기열 닫기 버튼을 누른다.
11	시스템은 사업자의 가게 대기열을 삭제하고 변경사항을 반영한다.

EXTENSION SCENARIOS	
Step	Branching Action
1	<p>1a. 가게의 대기열이 존재하지 않는다.</p> <p>…1a1. 시스템은 대기열 열기 버튼을 화면에 제공한다.</p> <p>…1a2. 사업자가 열기 버튼을 누른다.</p> <p>…1a3. 시스템은 대기열을 생성한다.</p> <p>1b. 가게의 대기열이 존재한다.</p> <p>…1b1. 시스템은 대기열을 불러와 화면에 출력한다.</p>
6	<p>6a. 대기자 입장 취소</p> <p>…6a1. 시스템은 선택한 대기자의 입장을 확인하는 알림창을 출력한다.</p> <p>…6a2. 사업자는 알림창에서 취소 버튼을 누른다.</p> <p>…6a3. Step 1로 돌아간다.</p>
8	<p>8a. 대기자 삭제 취소</p> <p>…8a1. 시스템은 선택한 대기자의 삭제를 확인하는 알림창을 출력한다.</p> <p>…8a2. 사업자는 알림창에서 취소 버튼을 누른다.</p> <p>…8a3. Step 1로 돌아간다.</p>
10	<p>10a. 대기열 닫기 취소</p> <p>…10a1. 시스템은 대기열 닫기를 확인하는 알림창을 출력한다.</p> <p>…10a2. 사업자는 알림창에서 취소 버튼을 누른다.</p> <p>…10a3. Step 1로 돌아간다.</p>
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사업자당 하루 평균 10회 이상
<Concurrency>	제한 없음
Due Date	2023.11.03.

Use case #12 : Register Restaurant	
GENERAL CHARACTERISTICS	
Summary	이 Use case는 사업자가 자신의 가게와 로그인 계정을 연동하기 위해 사업장을 관리하는(등록, 삭제) 기능을 설명한다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Owner
Preconditions	사용자가 사업자 유형으로 로그인을 완료한 상태이다.
Trigger	사업자가 자신의 가게를 로그인한 계정과 연동하려고 할 때
Success Post Condition	선택된 가게와 현재 로그인한 사업자 계정이 연동된다.
Failed Post Condition	사업자의 계정에 해당 가게가 등록되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사업자가 운영하는 가게를 로그인한 계정과 연동하고 관리하려 한다.
1	사업자는 사이드바에서 가게 등록하기 버튼을 누른다.
2	시스템은 가게 등록에 필요한 정보를 입력하는 화면을 출력한다.
3	사업자는 필수 정보(사업장 이름, 전화번호, 주소, 사업장 설명)를 입력하고, 등록 버튼을 누른다.
4	시스템은 사업자가 입력한 정보를 DB(FireStore)에 저장한다.
5	사업자가 사업장 삭제하기 버튼을 누른다.
6	시스템은 사업자 계정에서 선택된 사업장을 삭제한다.
EXTENSION SCENARIOS	
Step	Branching Action
5	5a. 사업장 삭제 취소 ...5a1. 시스템은 사업장 삭제를 재확인하는 알림창을 출력한다. ...5a2. 사업자는 알림창에서 취소 버튼을 누른다. ...5a3. Step 2로 돌아간다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사업자당 평균 2회 이상
<Concurrency>	제한 없음
Due Date	2023.11.03.

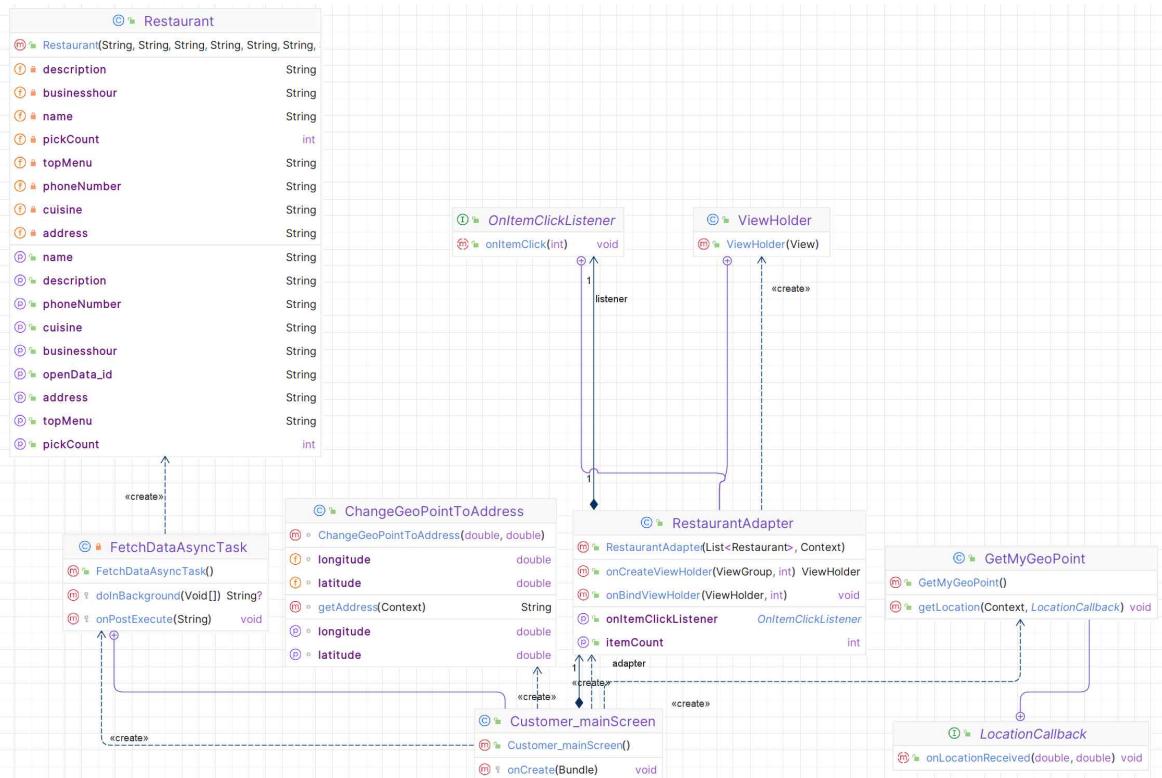
Use case #13 : Manage Donation	
GENERAL CHARACTERISTICS	
Summary	이 Use case는 사업자가 운영하는 가게의 재고를 다른 기관이나 가게에 기부하는 기능을 관리한다.
Scope	DGAJ(Daegu Go And Joy)
Level	User level
Author	테크자이언트
Last Update	2023. 11. 03
Status	Analysis
Primary Actor	Owner
Preconditions	사업자가 운영하는 가게는 시스템에 등록된 상태이고, 가까운 위치의 기부 대상 기관이 등록된 상태이다.
Trigger	사업자가 기부 관련 기능을 하려고 할 때
Success Post Condition	신청한 기부가 성공적으로 이루어진다.
Failed Post Condition	기부가 이루어지지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사업자가 사업장 재고를 기부하고자 한다.
1	사업자는 사이드바에서 기부하기 버튼을 누른다.
2	시스템은 기부하기 화면에 사업자가 기부한 목록을 제공한다.
3	사업자는 기부 신청하기 버튼을 누른다.
4	시스템은 기부를 신청하는 화면을 제공한다.
5	사업자는 기부 물품 유형, 개수, 기부 기관을 입력하고 신청하기 버튼을 누른다.
EXTENSION SCENARIOS	
Step	Branching Action
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사업자당 하루 평균 3회 이상
<Concurrency>	제한 없음
Due Date	2023.11.03.

3. Class diagram

3.1 Restaurant Class diagram

식당 정보를 저장하고 관리하는 Restaurant CD입니다.

- OnItemClickListener 인터페이스는 RecyclerView의 아이템 클릭 이벤트를 처리하기 위한 리스너를 정의합니다.
- ViewHolder 클래스는 RecyclerView의 각 아이템 뷰의 레이아웃 요소를 저장하고 관리합니다.
- RestaurantAdapter 클래스는 RecyclerView에 식당 목록을 표시하기 위한 어댑터 역할을 합니다.
- FetchDataAsyncTask 클래스는 비동기적으로 데이터를 가져오는 작업을 수행합니다.
- ChangeGeoPointToAddress 클래스는 지리적 좌표를 주소로 변환하는 역할을 합니다.
- GetMyGeoPoint 클래스는 현재 위치의 지리적 좌표를 가져오는 역할을 합니다.
- LocationCallback 클래스는 위치 관련 이벤트를 처리하기 위한 콜백을 정의합니다.
- Customer_mainScreen 클래스는 고객의 메인 화면을 나타내는 액티비티입니다.



[그림 3-1] Restaurant CD

Restaurant			
Class Description	Restaurant 클래스는 식당 정보를 관리하는 클래스, 식당의 이름, 요리 종류, 전화번호, 주소, 설명, 영업시간, 대표 메뉴, 오픈데이터 ID 및 Pick 수를 저장하고 제공하는 class		
Attributes	Name	Type	Visibility
	name	String	private
	식당의 이름을 나타내는 변수		
	cuisine	String	private
	식당의 요리 종류를 나타내는 변수		
	phoneNumber	String	private
	식당의 전화번호를 나타내는 변수		
	address	String	private
	식당의 주소를 나타내는 변수		
	description	String	private
	식당에 대한 설명을 나타내는 변수		
	businesshour	String	private
	식당의 영업시간을 나타내는 변수		
	topMenu	String	private
	식당의 대표 메뉴를 나타내는 변수		
	opendata_id	String	private
	오픈데이터 ID를 나타내는 변수		
	pickCount	int	private
	식당이 선택받은 (찜)수를 나타내는 변수		
Operations	Name	Argument	Returns
	Description		
	Restaurant	String name, String cuisine, String phone Number, String address, String description, String businesshour, String topMenu, String opendata_id, int pickCount	private
	Restaurant 클래스의 생성자. 매개변수로 받은 값을 각 속성에 설정		
	getName	None	String
	식당의 이름을 반환하는 메서드		
	getCuisine	None	String
	식당의 요리 종류를 반환하는 메서드		
	getPhoneNumber	None	String
	식당의 전화번호를 반환하는 메서드		
	getAddress	None	String
	식당의 주소를 반환하는 메서드		
	getDescription	None	String
	식당에 대한 설명을 반환하는 메서드		
	getBusinesshour	None	String
	식당의 영업시간을 반환하는 메서드		
	getTopMenu	None	String
	식당의 대표 메뉴를 반환하는 메서드		
	getOpenData_id	None	String
	오픈데이터 ID를 반환하는 메서드		
	getPickCount	None	int
	식당이 선택받은 (찜)수 반환하는 메서드		

RestaurantAdapter

Class	RestaurantAdapter 클래스는 RecyclerView에 식당 목록을 표시하기 위한 어댑터 클래스. 식당 정보를 받아와서 각각의 아이템 뷰에 표시	
Description	Name	Type
	Description	Visibility
Attributes	restaurantList	List<Restaurant>
	식당 목록을 담고 있는 리스트	private
	context	Context
	어댑터를 사용하는 변수	private
	listener	OnItemClickListener
Operations	아이템 클릭 이벤트를 처리하기 위한 리스너	private
	Name	Argument
	Description	Returns
	setOnItemClickListener	OnItemClickListener listener
	아이템 클릭 이벤트 리스너를 설정하는 메서드	void
Operations	RestaurantAdapter	List<Restaurant> restaurantList, Context context
	RestaurantAdapter 클래스의 생성자입니다. 식당 목록과 컨텍스트를 매개 변수로 받아 초기화하는 메서드	None
	onCreateViewHolder	ViewGroup parent, int viewType
	아이템 뷰를 위한 ViewHolder 객체를 생성하여 반환하는 메서드	ViewHolder
	onBindViewHolder	ViewHolder holder, int position
Operations	ViewHolder에 식당 정보를 바인딩하여 표시하는 메서드	void
	getItemCount	None
	식당 목록의 크기를 반환하는 메서드	int
	ViewHolder	None
	각 아이템 뷰의 레이아웃 요소를 저장하는 ViewHolder와 관한 메서드	static class
Operations	OnItemClickListener	None
	아이템 클릭 이벤트 리스너 인터페이스	interface

ChangeGeoPointToAddress

Class Description	ChangeGeoPointToAddress 클래스는 지리적 좌표를 주소로 변환하는 기능을 제공하는 클래스. Geocoder를 사용하여 좌표를 주소로 변환하고, 결과로 얻은 현재 위치 주소를 반환		
	Name	Type	Visibility
Attributes	Description		
	latitude	double	private
Operations	현재 위치의 위도 좌표를 저장하는 변수		
	longitude	double	private
Operations	현재 위치의 경도 좌표를 저장하는 변수		
	Name	Argument	Returns
Operations	Description		
	ChangeGeoPointToAd	double latitude, double longitude	None
Operations	위도와 경도를 초기화하기 위해 생성자에서 값을 받아온다		
	setLatitude	double latitude	None
Operations	위도를 설정하는 메서드		
	setLongitude	double longitude	None
Operations	경도를 설정하는 메서드		
	getLatitude	double	double
Operations	현재 저장된 위도 값을 반환하는 메서드		
	getLongitude	double	double
Operations	현재 저장된 경도 값을 반환하는 메서드		
	getAddress	Context mContext	String
Operations	좌표를 주소로 변환하여 현재 위치 주소를 반환하는 메서드		

GetMyGeoPoint

Class Description	GetMyGeoPoint 클래스는 현재 장치의 위치 정보를 가져오는 기능을 제공하는 클래스. LocationManager와 LocationListener를 사용하여 위치 정보를 수신하고, 마지막으로 수신된 위치 정보를 반환		
	Name	Type	Visibility
Attributes	Description		
	locationManager	LocationManager	private
Attributes	위치 관리자 변수		
	locationListener	LocationListener	private
Attributes	위치 변경 이벤트를 수신하는 리스너 객체		
	lastLocation	Location	private
Operations	마지막으로 수신된 위치 정보를 저장하는 변수		
	Name	Argument	Returns
Operations	Description		
	getLocation	Context context, LocationCallback callback	None
Operations	현재 장치의 위치 정보를 가져오는 메서드		
	interface LocationCallback	None	None
Operations	위치 정보를 전달하기 위한 콜백 인터페이스		

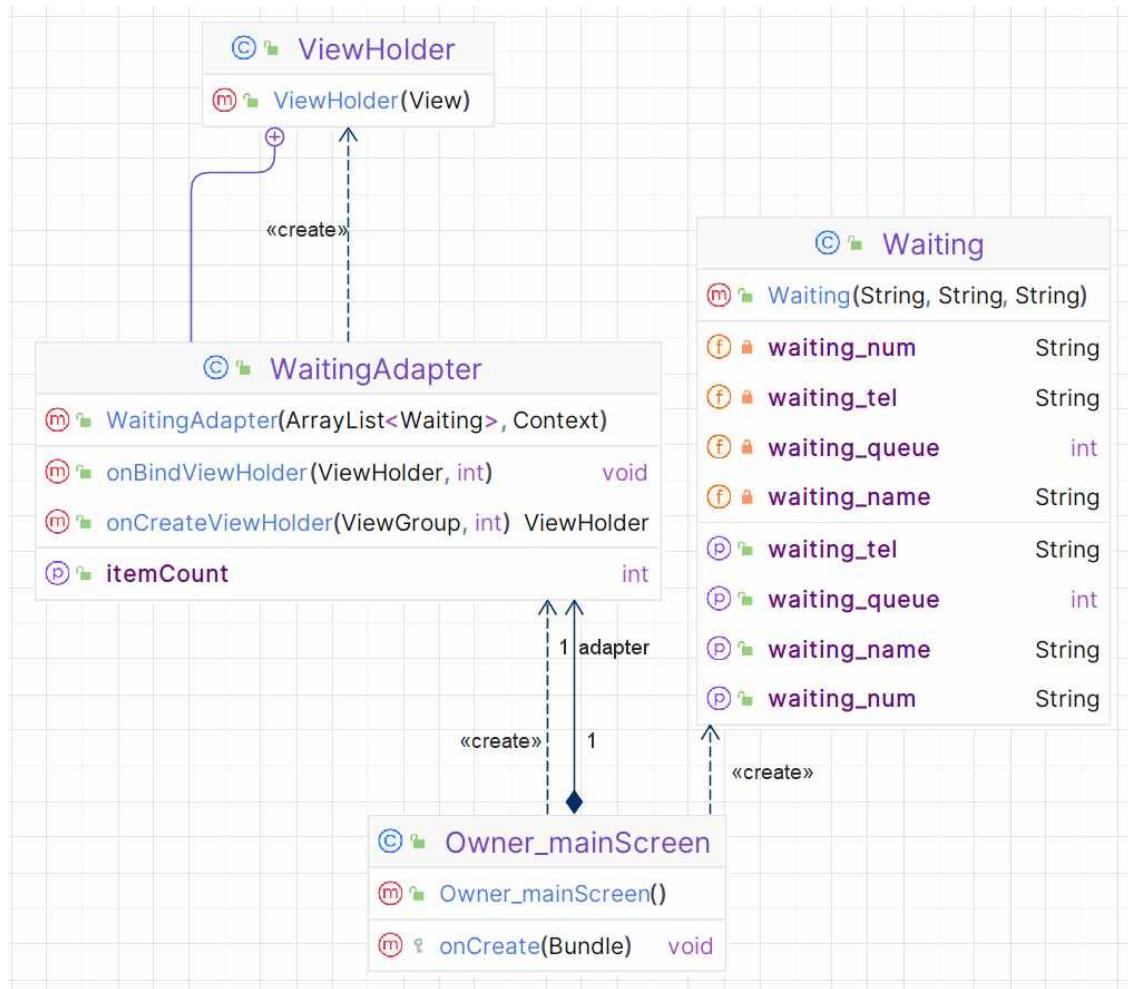
Customer_mainScreen

Class Description	사용자가 이용한 main화면을 관리하는 class		
	Name	Type	Visibility
	Description		
Attributes	mMap	GoogleMap	private
	Google Map기능을 이용하기 위한 변수		
	adapter	RestaurantAdapter	private
	restaurantList를 사용자와 연결시켜주는 변수		
	restaurantList	List<Restaurant>	private
	사용자가 이용할 수 있는 모든 가게 정보를 가진 list		
	nowAddress	String	private
	사용자의 현 위치를 가진 변수		
	Lineup	TextView	private
	사용자가 이용할 줄서기 기능을 나타내는 변수		
	searchMenu	String	private
	검색하고 싶은 해당 음식을 기록하는 변수		
	recyclerMenu	RecyclerView	private
	상세 기능을 이용하기 위해 누르는 버튼		
	searchPos	String	private
	사용자의 현위치를 간단하게 나타낸 변수		
Operations	drawerLayout2	DrawerLayout	private
	화면 왼쪽에 상세 메뉴가 담긴 창을 숨겨놓았다가 사용자가 상세 메뉴버튼을 누르면 화면에 나타나는 레이아웃		
	menu2	Button	private
	상세기능을 가진 창을 해당 창에 나타내기 위한 버튼		
	addressParts	String[]	private
	사용자가 있는 주소를 기록하는 변수		
	shop_page	TextView	private
	사용자가 이용할 가게 상세페이지를 나타내는 변수		
	address_gu	String	private
	사용자의 주소 중 해당 구를 나타내는 변수		
	Name	Argument	Returns
	Description		
	onCreate	None	protected
	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		
	onNavigationItemSelected	MenuItem	boolean
	화면 왼쪽에 숨겨놓은 상세메뉴가 담긴 창을 사용자가 상세메뉴버튼을 눌렀을 때 사용자가 이용할수 있도록 창을 나타내주는 메서드		

3.2 Waiting Class diagram

대기 목록을 확인하고 관리하는 Waiting CD입니다.

- ViewHolder 클래스는 RecyclerView의 각 아이템 뷰의 레이아웃 요소를 저장하고 관리합니다.
- WaitingAdapter 클래스는 RecyclerView에 대기 목록을 표시하기 위한 어댑터 역할을 합니다.
- Waiting 클래스는 대기 정보를 저장하고 관리하는 역할을 합니다.
- Owner_mainScreen 클래스는 가게 주인의 메인 화면을 나타내는 액티비티입니다.



[그림 3-2] Waiting CD

Waiting			
Class Description	줄서기 기능을 제공하고 관리하는 class		
	Name	Type	Visibility
	Description		
Attributes	waiting_queue	int	private
	줄서기 기능을 이용하는 사용자 수를 기록하기 위한 변수		
	waiting_num	String	private
	줄서기 기능을 이용하는 사용자를 기록하기 위한 변수		
	waiting_name	String	private
	사용자가 줄서기 기능을 이용하기 위해 입력한 이름을 기록하기 위한 변수		
Operations	waiting_tel	String	private
	사용자가 줄서기 기능을 이용하기 위해 입력한 전화번호를 기록하기 위한 변수		
		Argument	Returns
	Description		
	onCreate	None	protected
	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		
	getWaiting_tel	None	None
	waiting_tel 변수를 가지고 오는 메서드		
	getWaiting_queue	None	None
	waiting_queue 변수를 가지고 오는 메서드		

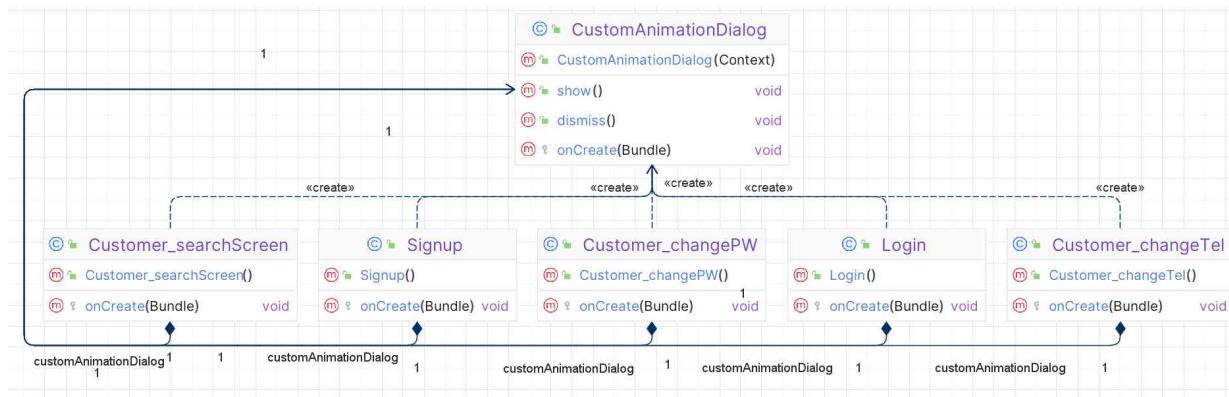
WaitingAdapter

Class Description	Waiting class가 ListView를 가져오기 위해 도움을 주는 class		
	Name	Type	Visibility
	Description		
Attributes	Waitinglist	ArrayList<Waiting>	private
	배열형태로서 줄서기한 사용자를 기록하는 변수		
	context	Context	private
	줄서기 한 사용자의 정보를 담은 변수		
	RES_ID	String	private static final
	사용자가 줄 서기한 식당을 기록하는 변수		
Operations	mStore	FirebaseFireStore	private
	Firebase store에 접근하기 위한 변수		
	Name	Argument	Returns
	Description		
	onCreate	None	protected
	ViewHolder기능을 이용하기 위해 ViewHolder를 생성하는 메서드		
	onBindViewHolder	WaitingAdapter.ViewHolder, int	None
	생성된 ViewHolder에 data를 입력시키는 역할을 하는 메서드		
	getItemCount		
	줄서기 기능을 이용하는 손님이 있는지 그 수를 확인 할 수 있는 메서드		
	waitingAdapter	ArrayList<Waiting>, Content	None
	배열 형태의 Waitinglist를 연결하는 메서드		
	onSuccess	None	None
	줄서기 기능을 이용한 손님을 성공적으로 삭제 했는지 확인하는 메서드		
	onComplete	Task<DocumentSnaps hot>	None
	줄서기 기능을 이용한 손님 명단을 계속해서 업데이트하는 메서드		
	onFailure	Exception	None
	줄서기 기능을 이용한 손님이 제대로 삭제되지 않을 경우 예외처리를 위한 메서드		

3.3 Signup Class diagram

사용자 인터페이스와 사용자 정보를 관리하는 Signup CD입니다.

- CustomAnimationDialog 클래스는 사용자 정의 애니메이션을 가진 대화 상자를 표시하는 역할을 합니다.
- Customer_searchScreen 클래스는 고객이 식당을 검색하는 화면을 나타내는 액티비티입니다.
- Signup 클래스는 사용자의 회원가입을 처리하는 역할을 합니다.
- Customer_changePW 클래스는 고객의 비밀번호 변경을 처리하는 역할을 합니다.
- Login 클래스는 사용자의 로그인을 처리하는 역할을 합니다.
- Customer_changeTel 클래스는 고객의 전화번호 변경을 처리하는 역할을 합니다.



[그림 3-3] Signup CD

Customer_changePW

Class Description	사용자가 기존에 등록한 비밀번호를 다른 비밀번호로 변경하는 class		
	Name	Type	Visibility
Attributes	Description		
	et_existPW	EditText	private
	기존의 비밀번호를 기록하기 위한 변수		
	mStore	FirebaseFirestore	private
	Firebase store에 접근하기 위한 변수		
	check	boolean	private
	기존의 비밀번호와 일치할 경우 true를 기록할 변수		
	btn_ok	Button	private
	비밀번호 변경을 확정하기 위한 버튼		
	et_checkPW	EditText	private
	변경된 비밀번호를 기록하기 위한 변수		
	btn_checkPW	Button	private
	기존의 비밀번호와 일치하는 비밀번호를 입력했는지 확인하는 버튼		
	et_newPW	EditText	private
Operations	변경할 새로운 비밀번호를 기록할 변수		
	mAuth	FirebaseAuth	private
	Firebase authentication에 접근해 인증받기위한 변수		
	customAnimation	CustomAnimation	private
	Dialog	Dialog	private
Operations	로딩화면을 구성하기 위한 변수		
	Name	Argument	Returns
	Description		
	onCreate	None	protected
	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		
	onComplete	Task<DocumentSnapshot>	None
	사용자의 새로운 비밀번호를 변경해주는 메서드		

Login			
Class Description	Login 클래스는 사용자 로그인 기능을 구현하는 클래스. Firebase Authentication과 Firestore를 사용하여 사용자 인증 및 데이터 조회를 수행		
	Name	Type	Visibility
	Description		
Attributes	id_login	EditText	private
	이메일 입력을 위한 EditText		
	pw_login	EditText	private
	비밀번호 입력을 위한 EditText		
	btn_login	Button	private
	로그인 버튼		
	btn_signup	Button	private
	회원가입 버튼		
	mAuth	FirebaseAuth	private
	Firebase Authentication 인스턴스		
Operations	mStore	FirebaseFirestore	private
	Firebase Firestore 인스턴스		
	RES_ID	String	private
	상수로 정의된 식당 ID		
	customAnimationDialog	CustomAnimationDialog	private
	로딩 화면을 표시하기 위한 CustomAnimationDialog 인스턴스		
	onCreate	Bundle savedInstanceState	None
	액티비티가 생성될 때 호출되는 메서드, 레이아웃 초기화 및 버튼 클릭 이벤트 등을 설정, 로그인 버튼 클릭 시 Firebase Authentication을 사용하여 로그인을 수행, Firestore에서 사용자 정보를 조회하고, 사용자 역할에 따라 액티비티를 전환		
	Name	Argument	Returns
	Description		

CustomAnimationDialog

Class Description	앱 실행 중 나타나는 로딩 화면을 커스텀으로 구현하기 위한 클래스		
	Name	Type	Visibility
	Description		
Attributes	c	Context	None
	CustomAnimationDialog의 인스턴스를 생성할 때 전달된 Context 객체		
	imgLogo	ImageView	private
	로딩 화면에 표시되는 로고 이미지를 나타내는 ImageView 객체		
	Name	Argument	Returns
	Description		
Operations	CustomAnimationDialog	Context context	private
	CustomAnimationDialog 클래스의 생성자. 로딩 화면에 필요한 설정을 수행		
	onCreate	Bundle savedInstanceState	None
	로딩 화면의 레이아웃을 설정하고, 로고 이미지에 애니메이션을 적용		
	show	None	None
	로딩 화면을 표시하는 메서드입니다. ProgressDialog의 show() 메서드를 호출		
	dismiss	None	None
	로딩 화면을 종료하는 메서드입니다. ProgressDialog의 dismiss() 메서드를 호출		

Customer_searchScreen

Class	오늘의 메뉴를 검색하거나 다른 세부 기능들을 이용할 수 있도록 연결시켜주는 class		
Description	Name	Type	Visibility
	Description		
Attributes	txt_search_pos	EditText	private
	사용자의 위치를 기록하는 변수		
	btn_search	Button	private
	사용자가 해당 음식을 입력한 후 검색하기 위한 버튼		
	btn_game	Button	private
	사용자가 게임 기능을 이용할 때 선택하는 버튼		
	menu	Button	private
	사용자가 이용할 수 있는 기능 menu들을 모아놓은 창을 이용하기 위해 선택하는 버튼		
	customAnimation	CustomAnimation	
	Dialog	Dialog	private
	로딩화면을 구성하기 위한 변수		
	drawerLayout	DrawerLayout	private
	화면 왼쪽에 상세 메뉴가 담긴 창을 숨겨놓았다가 사용자가 상세 메뉴버튼을 누르면 화면에 나타나는 레이아웃		
Operations	txt_search_menu	EditText	private
	검색하고 싶은 해당 음식을 기록하는 변수		
	Name	Argument	Returns
	Description		
	onCreate	None	protected
	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		
	onNavigationItemSelected	MenuItem	boolean
	화면 왼쪽에 숨겨놓은 상세메뉴가 담긴 창을 사용자가 상세메뉴버튼을 눌렀을 때 사용자가 이용할 수 있도록 창을 나타내주는 메서드		

Customer_changeTel

Class Description	사용자가 기존에 등록한 전화번호를 다른 전화번호로 변경하는 class		
	Name	Type	Visibility
Attributes	Description		
	check	boolean	private
	기존의 전화번호와 일치할 경우 true를 기록할 변수		
	mStore	FirebaseFirestore	private
	Firebase store에 접근하기 위한 변수		
	customAnimation	CustomAnimation	
	Dialog	Dialog	private
	로딩화면을 구성하기 위한 변수		
	et_checkTel02	EditText	private
	사용자가 변경하고 싶은 새로운 전화번호 마지막 4자리를 제대로 입력했느니 확인할 변수		
	et_newTel02	EditText	private
	사용자가 변경하고 싶은 새로운 전화번호 마지막 4자리를 기록할 변수		
	et_existTel01	EditText	private
	사용자의 기존 전화번호 중간 4자리를 기록하는 변수		
Operations	Name	Argument	Returns
	Description		
	onCreate	None	protected
	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		
	onComplete	Task<Document Snapshot>	None
	사용자의 새로운 전화번호를 변경해주는 메서드		

Signup			
Class Description	Signup 클래스는 사용자 회원가입 기능을 구현하는 클래스. Firebase Authentication과 Firestore를 사용하여 사용자 계정을 생성하고 정보를 저장		
Attributes	Name	Type	Visibility
	Description		
	pw_join	EditText	private
	비밀번호 입력을 위한 EditText		
	name_join	EditText	private
	이름 입력을 위한 EditText		
	id_join	EditText	private
	아이디 입력을 위한 EditText		
	tel_join01	EditText	private
	휴대폰 번호 첫 번째 자리를 입력하는 EditText		
	tel_join02	EditText	private
	휴대폰 번호 두 번째 자리를 입력하는 EditText		
	btn_join	Button	private
	회원가입 버튼		
	btn_id_check	Button	private
	아이디 중복 체크 버튼		
	choose_join_type	RadioGroup	private
	사용자 유형 선택을 위한 RadioGroup		
	btn_choose_user	RadioButton	private
	사용자 유형을 나타내는 RadioButton (일반 사용자)		
	btn_choose_owner	RadioButton	private
	사용자 유형을 나타내는 RadioButton (식당 주인)		
	mAuth	FirebaseAuth	private
	Firebase Authentication 인스턴스		
	mStore	FirebaseFirestore	private
	Firebase Firestore 인스턴스		
	spinnerEmailDomains	Spinner	private
	이메일 도메인 선택을 위한 Spinner		
	customAnimationDial	CustomAnimationDialog	private
	로딩 화면을 표시하기 위한 CustomAnimationDialog 인스턴스		
Operations	Name	Argument	Returns
	Description		
	onCreate	Bundle savedInstanceState State	None
	액티비티가 생성될 때 호출되는 메서드, 레이아웃 초기화 및 버튼 클릭 이벤트 등을 설정, 아이디 중복 체크 버튼 클릭 시 Firestore에서 중복된 아이디가 있는지 확인, 회원가입 버튼 클릭 시 FirebaseAuth을 사용하여 회원가입을 수행하고, Firestore에 사용자 정보를 저장		

3.4 Customer Class diagram

고객이 사용할 수 있는 기능을 정의한 Customer CD입니다.

- Customer_game_byLots 클래스는 고객이 랜덤으로 식당을 선택하는 게임을 플레이하는 화면을 나타냅니다.
- Customer_game_roulette 클래스는 고객이 룰렛을 돌려 식당을 선택하는 게임을 플레이하는 화면을 나타냅니다.
- Customer_restaurantDetail 클래스는 고객이 선택한 식당의 상세 정보를 확인하는 화면을 나타냅니다.
- Customer_game_choose 클래스는 고객이 선택지 중에서 식당을 선택하는 게임을 플레이하는 화면을 나타냅니다.
- Customer_game_setItem 클래스는 고객이 게임에서 사용할 아이템을 설정하는 화면을 나타냅니다.
- Customer_applyWaiting 클래스는 고객이 대기열에 등록하는 화면을 나타냅니다.
- Customer_myAccount 클래스는 고객의 계정 정보를 확인하고 관리하는 화면을 나타냅니다.
- Customer_myloveList 클래스는 고객이 좋아하는 식당 목록을 확인하는 화면을 나타냅니다.
- Customer_myWaiting 클래스는 고객이 등록한 대기 목록을 확인하는 화면을 나타냅니다.
- Customer_recentList 클래스는 최근 방문한 식당 목록을 확인하는 화면을 나타냅니다.
- Customer_topList 클래스는 인기 있는 식당 목록을 확인하는 화면을 나타냅니다.



[그림 3-4] Customer function CD

Customer_applywaiting

Class Description	줄서기 기능을 관리하는 class		
	Name	Type	Visibility
Attributes	Description		
	lineup_tel	EditText	private
	줄서기 기능 이용시, 입력해야 하는 전화번호를 기록하는 변수		
	lineup_name	EditText	private
	줄서기 기능 이용시, 입력해야 하는 이름을 기록하는 변수		
	reservation_num	EditText	private
	줄선 사용자를 정의하기 위한 변수		
	btn_confirm_line_up	Button	private
	줄서기를 사용자가 확정할 때 누르는 버튼		
	queue	long	private
Operations	줄서는 사용자를 기록하기 위한 변수		
	mStore	FirebaseFirestore	private
	Firebase store의 접근하기 위한 변수		
	restaurant	String	public
	Opendata_id		
	사용자가 줄 선 가게의 id를 저장하는 변수		
	Name	Argument	Returns
	Description		
	onCreate	None	protected
	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		
	onSuccess	None	None
	사용자가 줄서기 하고 싶은 가게의 줄서기 명단에 사용자를 입력해주는 메서드		
	onFailure	Exception	None
	줄서기 기능을 이용하고 싶은 사용자를 가게 줄서기 명단에 입력하지 못한 경우 예외처리를 위한 메서드		

Customer_game_byLots

Class Description	사용자가 이용할 수 있는 게임 중 제비뽑기 게임에 관한 class		
	Name	Type	Visibility
	Description		
Attributes	txt_printMenu	TextView	private
	사용자가 이용할 게임을 나타내는 변수		
	btn_byLots	Button	private
	룰렛을 돌리는 버튼		
	btn_reset	Button	private
	입력한 내용을 리셋하기 위해 이용하는 버튼		
	et_input_menu	EditText	private
	룰렛에 들어갈 메뉴를 입력하는 변수		
	btn_inputMenu	Button	private
	룰렛에 들어갈 메뉴를 입력하고 룰렛에 넣는 버튼		
Operations	btn_home	Button	private
	게임 화면이 아닌 홈 화면으로 돌아가기 위해 이용하는 버튼		
	Name	Argument	Returns
	Description		
Operations	onCreate	None	protected
	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		

Customer_game_choose

Class Description	사용자가 이용할 게임을 선택하는 class		
	Name	Type	Visibility
	Description		
Attributes	btn_choose	ImageButton	private
	Roulette		
	게임중 룰렛을 선택하는 경우 이용하는 버튼		
	btn_chooseByLots	ImageButton	private
	게임 중 제비뽑기를 선택하는 경우 이용하는 버튼		
Operations	txt_chooseGame	TextView	private
	사용자가 이용할 게임을 나타내는 변수		
	Name	Argument	Returns
	Description		
Operations	onCreate	None	protected
	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		

Customer_game_roulette

Class Description	사용자가 이용할 수 있는 게임 중 룰렛 게임에 관한 class		
	Name	Type	Visibility
Attributes	Description		
	wheelItem	List<WheelItem>	public
	룰렛판에 사용자가 입력한 메뉴들을 담고 있는 list		
	luckyWheel	LuckyWheel	private
	룰렛 기능을 담고 있는 변수		
	point	String	public
	룰렛의 결과를 나타내줄 변수		
	Name	Argument	Returns
	Description		
	onCreate	None	protected
Operations	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		
	rotateWheelTo	point	None
	룰렛을 돌리는 메서드		
	OnReachTarget	None	None
	룰렛 결과를 나타내주는 메서드		
	generateWheelItems	None	None
	룰렛을 꾸며주는 메서드		
	addWheelItems	wheelItems	None
	룰렛에 사용자가 입력한 메뉴를 넣는 메서드		
	drawable ToBitmap		None
	룰렛을 그리는 메서드		

Customer_game_setItem

Class	사용자가 룰렛게임을 이용할 경우, 룰렛에 들어갈 item을 setting해주는 class		
Description	Name	Type	Visibility
	Description		
Attributes	item3	EditText	private
	룰렛에 적힐 메뉴를 기록할 변수3		
	btn_next	Button	private
	룰렛에 모든 메뉴를 입력하기 위해 누르는 버튼		
	txt_menu_game	TextView	private
	사용자가 이용할 게임을 나타내는 변수		
	item1	EditText	private
	룰렛에 적힐 메뉴를 기록할 변수1		
	item	String[]	private
	룰렛에 적힌 모든 메뉴를 가지고 있는 string 배열		
	item4	EditText	private
	룰렛에 적힐 메뉴를 기록할 변수4		
	item2	EditText	private
	룰렛에 적힐 메뉴를 기록할 변수2		
	item5	EditText	private
	룰렛에 적힐 메뉴를 기록할 변수5		
	item6	EditText	private
	룰렛에 적힐 메뉴를 기록할 변수6		
Operations	Name	Argument	Returns
	Description		
Operations	onCreate	None	protected
	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		

Customer_myAccount

Class Description	사용자의 계정을 관리하는 class		
	Name	Type	Visibility
	Description		
Attributes	mStore	FirebaseFirestore	private
	Firebase store에 접근하기 위한 변수		
	btn_change_password	Button	private
	비밀번호를 변경하기 위해 선택하는 버튼		
	btn_change_tel	Button	private
	전화번호를 변경하기 위해 선택하는 버튼		
	btn_delete_account	Button	private
	계정을 탈퇴하기 위해 선택하는 버튼		
	mFirebaseAuth	FirebaseAuth	private
Operations	Firebase authentication에 접근하기 위한 변수		
	Name	Argument	Returns
	Description		
Operations	onCreate	None	protected
	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		
	Name	Argument	Returns

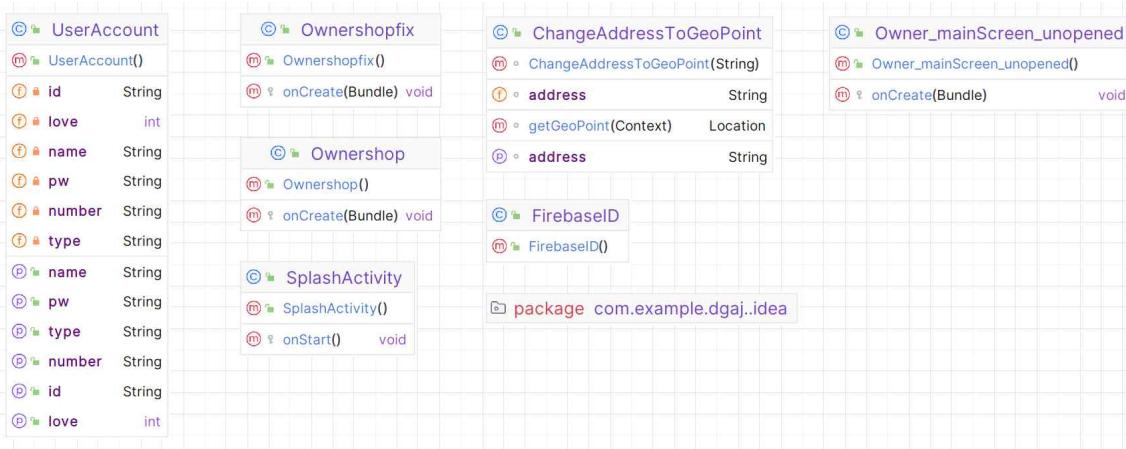
Customer_restaurantDetail

Class Description	사용자가 선택한 가게의 상세 페이지를 보여주는 class		
	Name	Type	Visibility
	Description		
Attributes	lineup	TextView	public
	줄서기 기능을 나타내는 변수		
	btn_lineup	Button	private
	사용자가 줄서기 기능을 이용하기 위해 선택하는 버튼		
	mStore	FirebaseFirestore	private
Operations	Firebase store에 접근하기 위한 변수		
	Name	Argument	Returns
	Description		
Operations	onCreate	None	protected
	시스템이 실행될 때 항상 같이 실행되는 메서드로써 Bundle 상태가 null인지 확인하고 null일 경우 없어진 이전의 instance를 복원하지 않고 새로운 instance를 생성시켜주는 메서드		
	onComplete	Task<DocumentSnapshot>	None
가게 상세정보를 계속해서 업데이트하는 메서드			

3.5 Extra Class diagram

기타 다양한 기능들을 모아둔 Extra CD입니다.

- UserAccount 클래스는 사용자 계정 정보를 관리하는 역할을 합니다.
- Ownershopfix 클래스는 가게 주인이 가게 정보를 수정하는 화면을 나타내는 역할을 합니다.
- ChangeAddressToGeoPoint 클래스는 주소를 지리적 좌표로 변환하는 역할을 합니다.
- Owner_mainScreen_unopened 클래스는 가게 주인의 미확인 주문을 확인하는 화면을 나타내는 역할을 합니다.
- Ownershop 클래스는 가게 정보를 저장하고 관리하는 역할을 합니다.
- SplashActivity 클래스는 애플리케이션의 시작 시 화면을 나타내는 역할을 합니다.
- FirebaseID 클래스는 Firebase에서 사용자의 식별자를 관리하는 역할을 합니다.



[그림 3-5] Extra CD

SplashActivity

Class Description	앱 실행 시 로딩 화면을 보여주기 위한 역할을 수행하는 클래스		
	Name	Type	Visibility
Attributes	Description		
	imageView	ImageView	private
	로딩 화면을 표시하기 위한 ImageView 객체		
Operations	animationDrawable	AnimationDrawable	private
	로딩 화면의 애니메이션을 제어하기 위한 AnimationDrawable 객체		
	Name	Argument	Returns
Operations	Description		
	onStart	None	protected
	일정 시간이 지난 후 Intent를 사용하여 SplashActivity에서 Login Activity로 이동하게 하는 메서드		

ChangeAddressToGeoPoint

Class Description	changeAddressToGeoPoint 클래스는 주소를 지리적 좌표로 변환하는 기능을 제공하는 클래스. Geocoder를 사용하여 주소를 좌표로 변환하고, 결과로 얻은 위도(latitude)와 경도(longitude)를 저장		
	Name	Type	Visibility
Attributes	Description		
	latitude	double	private
	주소의 위도 좌표를 저장하는 변수		
Attributes	longitude	double	private
	주소의 경도 좌표를 저장하는 변수		
	address	String	private
Operations	변환하고자 하는 주소를 저장하는 변수		
	Name	Argument	Returns
	Description		
Operations	ChangeAddressToGeoPoint	String address	None
	주소를 초기화하기 위해 생성자에서 주소를 받아온다.		
	setAddress	String address	private
Operations	주소를 설정하는 메서드		
	getAddress	None	String
	현재 저장된 주소를 반환하는 메서드		
Operations	getGeoPoint	Context mcontext	Location
	주소를 지리적 좌표로 변환하여 Location 객체로 반환하는 메서드		

FirebaseID

Class Description	FirebaseID 클래스는 Firebase Firestore에서 사용되는 컬렉션 및 문서 식별자를 관리하는 클래스. 각 상수는 해당하는 컬렉션 또는 문서의 이름을 나타낸다. (변수 설정으로 오타 방지용 클래스)		
	Name	Type	Visibility
Constants	Description		
	id_list	String	public
	ID 목록 컬렉션의 이름		
	waiting_list	String	public
	대기 목록 컬렉션의 이름		
	post	String	public
	게시물 컬렉션의 이름		
	upload	String	public
	업로드 컬렉션의 이름		
	title	String	public
	제목을 나타내는 필드의 이름		
	contents	String	public
	내용을 나타내는 필드의 이름		
	save	String	public
	찜목록 컬렉션의 이름		
	documentId	String	public
	문서 식별자를 나타내는 필드의 이름		
	email	String	public
	이메일을 나타내는 필드의 이름		
	password	String	public
	비밀번호를 나타내는 필드의 이름		
	tel	String	public
	전화번호를 나타내는 필드의 이름		
	name	String	public
	이름을 나타내는 필드의 이름		
	type	String	public
	타입을 나타내는 필드의 이름		
	wait_name	String	public
	대기자 이름을 나타내는 필드의 이름		
	wait_tel	String	public
	대기자 전화번호를 나타내는 필드의 이름		
	wait_num	String	public
	대기 번호를 나타내는 필드의 이름		
	collectionId	String	public
	컬렉션 식별자를 나타내는 필드의 이름		

UserAccount

Class	UserAccount 클래스는 사용자 계정 정보를 관리하는 클래스. Firebase Firestore에서 사용자 계정 데이터를 가져오거나 저장할 때 사용	
Description	Name	Type
	Description	Visibility
Attributes	id	String
	사용자의 ID	private
	love	ArrayList<User>
	사용자의	private
	name	String
	사용자의 이름	private
	number	String
	사용자의 전화번호	private
	pw	String
	사용자의 비밀번호	private
Operations	type	int
	사용자의 유형을 나타내는 정보	private
	Name	Argument
	Description	Returns
	getId()	None
	사용자의 ID를 반환	String
	setId	String id
	사용자의 ID를 설정	None
	getName	None
	사용자의 이름을 반환	String
	setName	String name
	사용자의 이름을 설정	None
	getNumber	None
	사용자의 전화번호를 반환	String
	setNumber	String number
	사용자의 전화번호를 설정	None
	getPw	None
	사용자의 비밀번호를 반환	String
	setPw	String pw
	사용자의 비밀번호를 설정	None
	getType	None
	사용자의 유형을 반환	String
	setType	String type
	사용자의 유형을 설정	None

Owner_mainScreen			
Class	Owner_mainScreen 클래스는 식당 주인의 메인 화면을 구현하는 클래스.		
Description	식당 주인은 대기 목록을 확인하고 대기장을 닫을 수 있다.		
		Name	Type
		Description	
Attributes	menu	Button	private
	메뉴 버튼		
	btn_res_close	Button	private
	대기장 닫기 버튼		
	drawerLayout	DrawerLayout	private
	네비게이션 드로어를 포함하는 레이아웃		
	recyclerView	RecyclerView	private
	대기 목록을 표시하는 RecyclerView		
	adapter	WaitingAdapter	private
	대기 목록을 관리하는 어댑터		
	arrayList	ArrayList<Waiting>	private
	대기 목록 데이터를 담는 ArrayList		
Operations	onCreate	Bundle savedInstanceState	None
	Name	Argument	Returns
	Description		
	액티비티가 생성될 때 호출되는 메서드, 레이아웃 초기화 및 버튼 클릭 이벤트 등을 설정. Firestore에서 대기 목록 데이터를 가져와 RecyclerView에 표시, 대기장 닫기 버튼 클릭 시 대기 목록을 삭제하고 Owner_mainScreen_unopened 액티비티로 전환		

Owner_mainScreen_unopened

Class Description	Owner_mainScreen_unopened 클래스는 식당 주인의 메인 화면(대기창이 아직 열리지 않은 상태)을 구현하는 클래스. 식당 주인은 대기창을 열 수 있다.		
	Name	Type	Visibility
Attributes	Description		
	menu	Button	private
	메뉴 버튼		
	btn_res_open	Button	private
	대기창 열기 버튼		
	drawerLayout	DrawerLayout	private
	네비게이션 드로어를 포함하는 레이아웃		
	mstore	FirebaseFirestore	private
Operations	Firebase Firestore 인스턴스		
	Description	Argument	Returns
	res_id	String	private
	식당 ID		
	Name	Argument	Returns
	Description		
	onCreate	Bundle savedInstanceState State	None
	액티비티가 생성될 때 호출되는 메서드, 레이아웃 초기화 및 버튼 클릭 이벤트 등을 설정, 대기창 열기 버튼 클릭 시 Firestore에 대기 목록을 추가하고 Owner_mainScreen 액티비티로 전환		

Ownershop

Class Description	Ownershop 클래스는 식당 주인이 가게 카테고리를 선택하는 기능을 구현하는 클래스		
	Name	Type	Visibility
Attributes	Description		
	btn_ownershop_categ ory	Button	private
	가게 카테고리 선택 버튼		
	SelectedItems	List<String>	private
	선택된 카테고리를 저장하는 리스트		
	builder	AlertDialog.Builder	private
	다이얼로그를 생성하기 위한 AlertDialog.Builder 객체		
	Description	Argument	Returns
Operations	Name	Argument	Returns
	Description		
	onCreate	Bundle savedInstanceState State	None
	액티비티가 생성될 때 호출되는 메서드, 레이아웃 초기화 및 버튼 클릭 이벤트 등을 설정, 가게 카테고리 선택 버튼 클릭 시 다이얼로그를 표시		

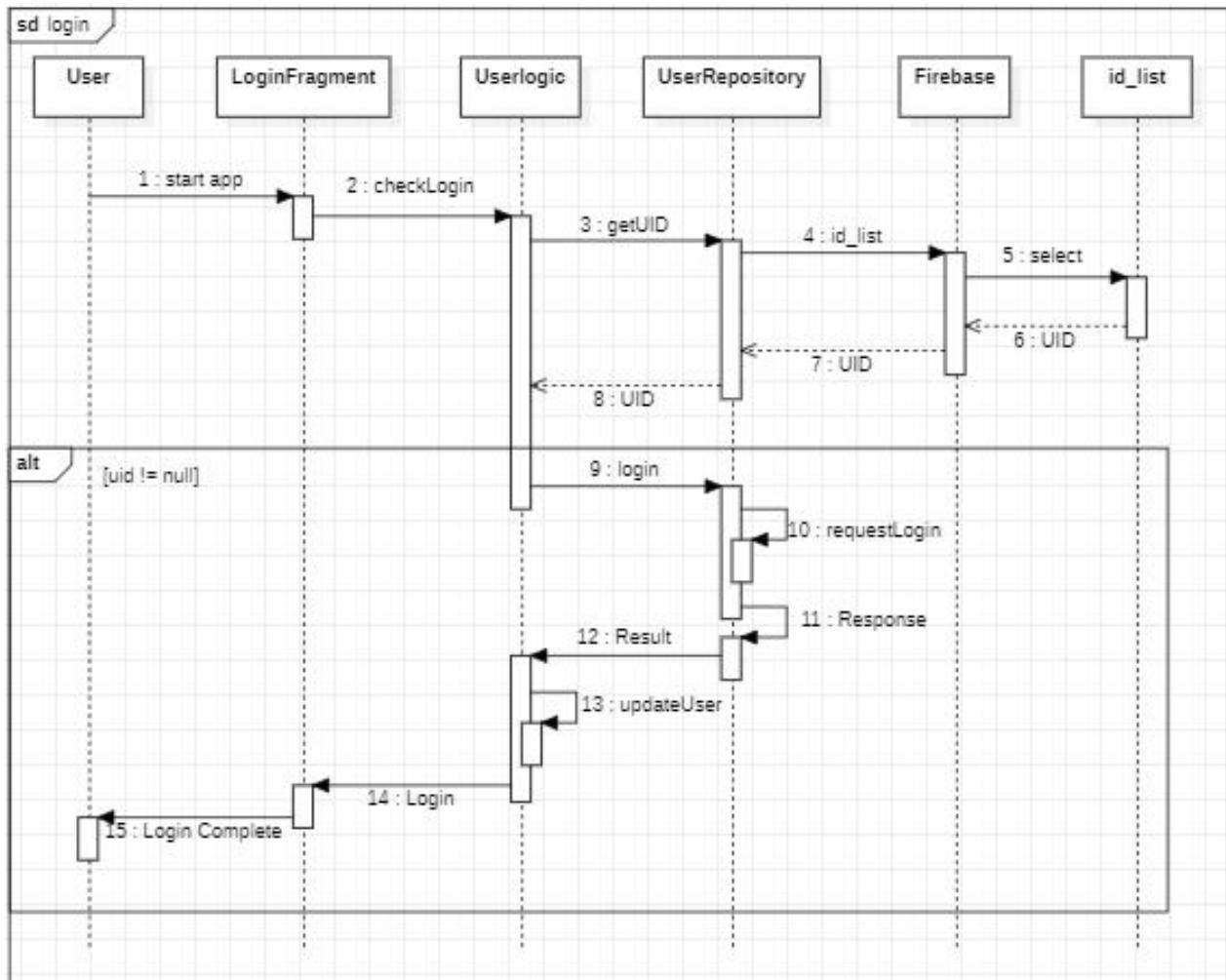
Ownershopfix			
Class	Ownershopfix 클래스는 식당 주인이 가게 카테고리를 수정하는 기능을 구현하는 클래스		
Attributes	Name	Type	Visibility
	Description		
	btn_ownershop_categ oryfix	Button	private
	가게 카테고리 수정 버튼		
	SelectedItems	List<String>	private
Operations	선택된 카테고리를 저장하는 리스트		
	builder	AlertDialog.Builder	private
	다이얼로그를 생성하기 위한 AlertDialog.Builder 객체		
	Name	Argument	Returns
	Description		
	onCreate	Bundle savedInstanceState	None
	액티비티가 생성될 때 호출되는 메서드, 레이아웃 초기화 및 버튼 클릭 이벤트 등을 설정, 가게 카테고리 수정 버튼 클릭 시 다이얼로그를 표시	State	

ViewHolder

Class	각 View의 객체를 보관해 View의 내용에 접근하기 위해 필요한 메소드의 반복을 줄이는 역할을 하는 class		
Description	Name	Type	Visibility
	Description		
Attributes	customer_name	TextView	private
	사용자가 줄서기 기능을 이용하기 위해 입력한 이름을 기록하기 위한 변수		
	customer_num_of_group	TextView	private
	줄서기 기능을 이용한 사용자의 수를 담기 위한 변수		
	customer_tel_number	TextView	private
	사용자가 줄서기 기능을 이용하기 위해 입력한 전화번호를 기록하기 위한 변수		
	btn_enter_customer	TextView	private
	줄서기 기능을 이용한 손님을 차례대로 입장 시킬 때 손님을 입장시킬 수 있는 버튼		
	btn_delete_customer	TextView	private
Operations	Name	Argument	Returns
	Description		
	ViewHolder	View	None
각 View의 객체를 ViewHolder에 보관하는 기능을 하는 메서드			

4. Sequence diagram

Login

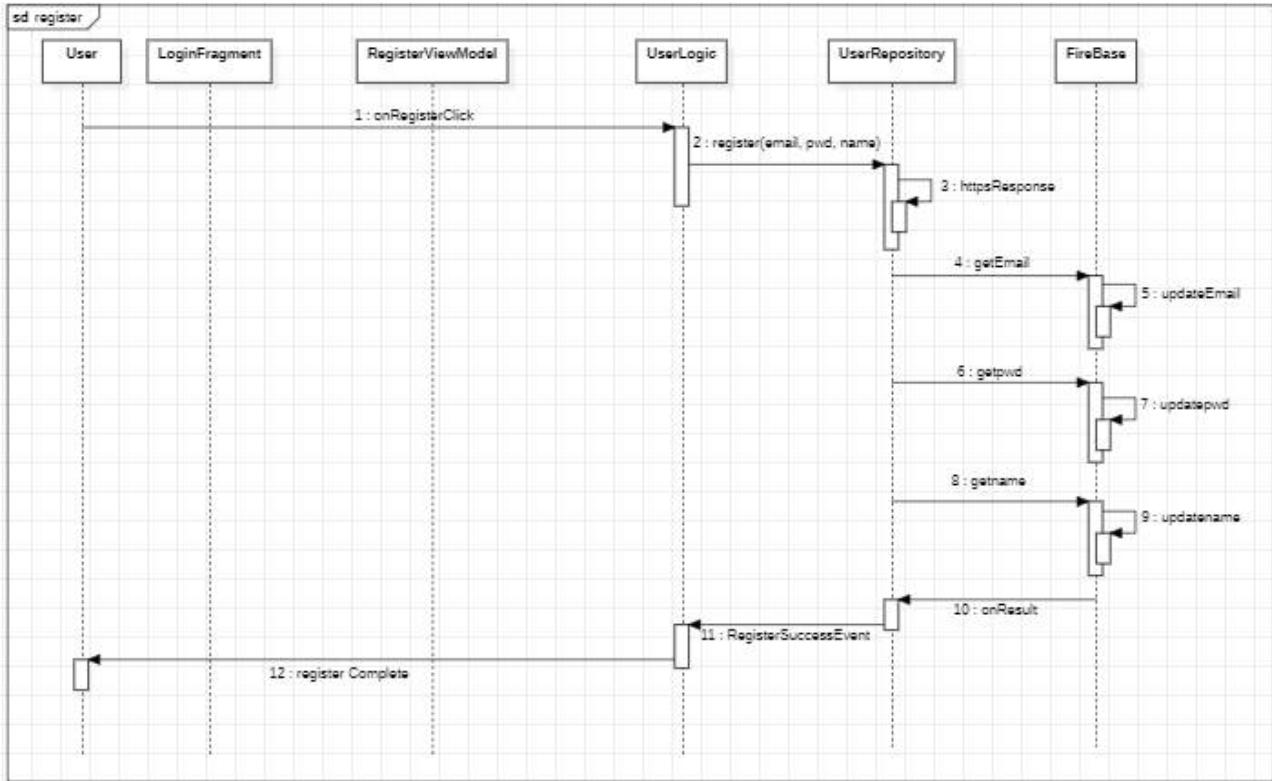


[그림 4-1] Login SD

사용자가 시스템에 로그인하는 Use Case를 나타내는 Sequence Diagram이다. User Class Description에서 <Use Case #2>의 경우이다.

앱이 시작되면서 기능을 실행한다. FireStore에 등록된 계정이 있을 경우, 해당 계정을 입력하여 로그인 요청을 하면 FireStore에서의 값과 비교하여 메뉴 검색 화면으로 넘어 가게 된다. 계정이 존재하지 않을 경우, 계정이 존재하지 않는다는 메시지가 뜨게 된다.

Register

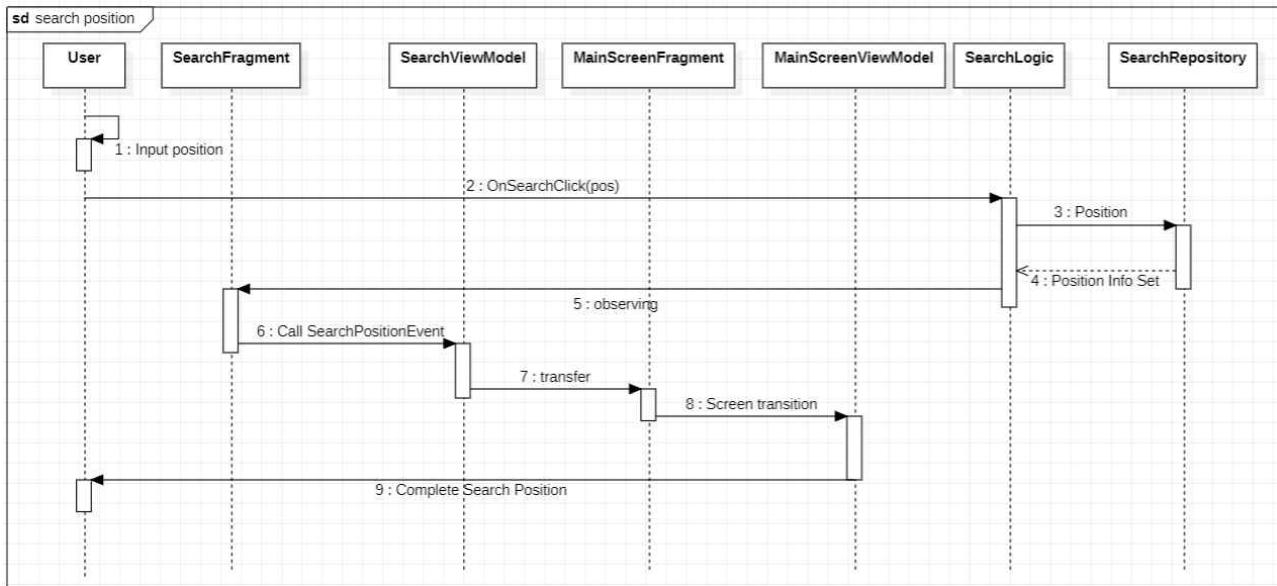


[그림 4-2] Register SD

사용자가 시스템에 회원가입하는 Use Case를 나타내는 Sequence Diagram이다. Class Description에서 <Use Case #1>의 경우이다.

시작화면에서 회원가입 버튼을 입력하면 실행된다. 회원가입 화면에서 정해진 양식에 따라 아이디(이메일), 비밀번호, 이름, 전화번호, 사용자 유형을 선택한다. 양식에 맞는 값을 입력한 후, 회원가입 버튼을 누르면 회원가입 요청을 하고, Firebase(Authentication, Fi reStore)에 계정 정보를 저장한다. 이후, 로그인 화면으로 전환된다.

Search Position

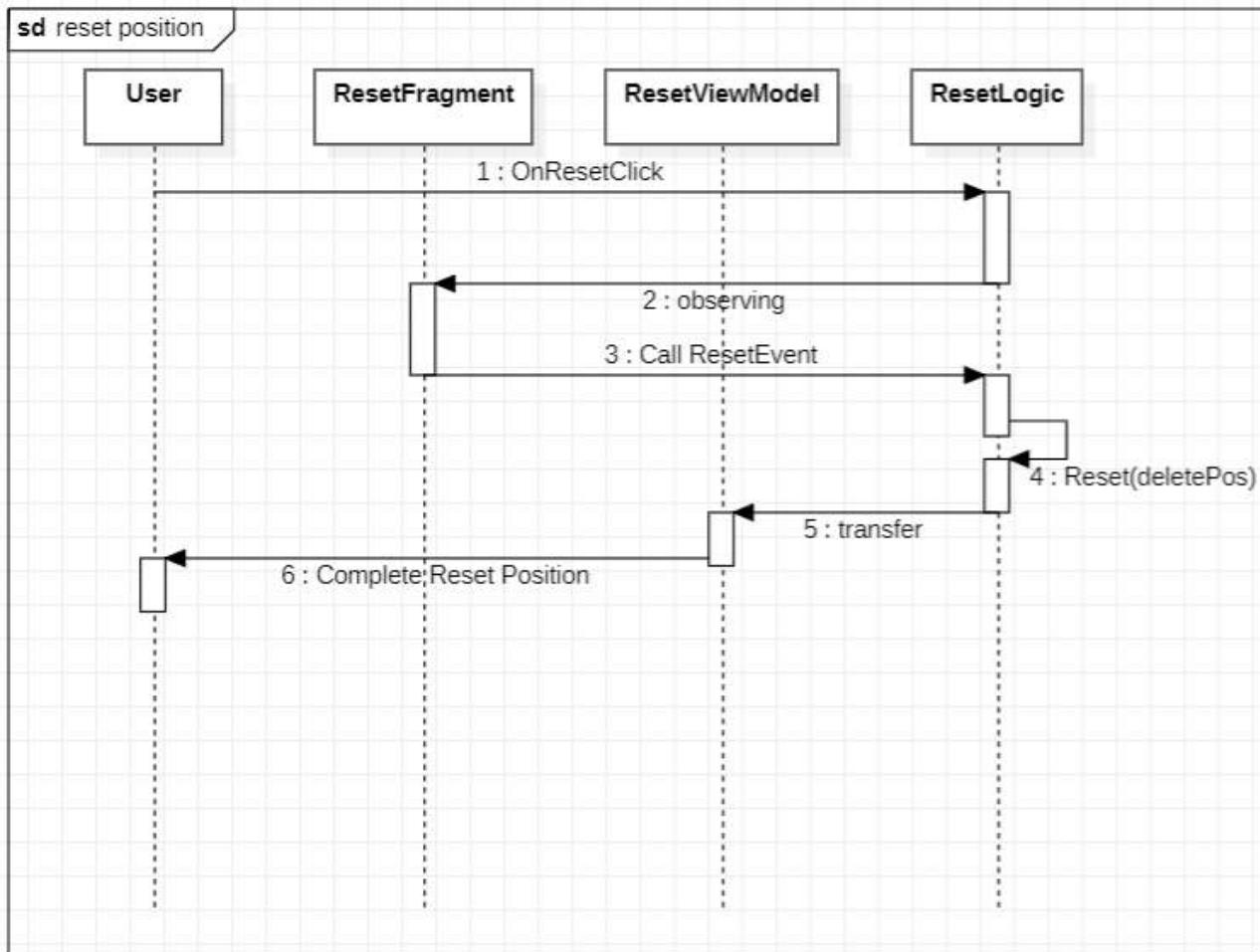


[그림 4-3] search position SD

고객이 검색하고자 하는 위치의 맛집을 검색하는 use case를 나타내는 sequence diagram이며, <Use Case #3-1a>의 경우이다.

고객이 위치를 입력하면 이를 Input으로 받는다. 사용자가 ‘검색’ 버튼을 누르면 검색 위치를 변수로 저장한 뒤에 해당 위치를 가진 맛집들을 API 리스트에서 Set으로 가져온다. Main Screen으로 화면을 전환한다. Main Screen에는 해당 위치를 기준으로 한 지도와 Set으로 가져온 맛집들을 추천 리스트로 제공한다.

Reset Position

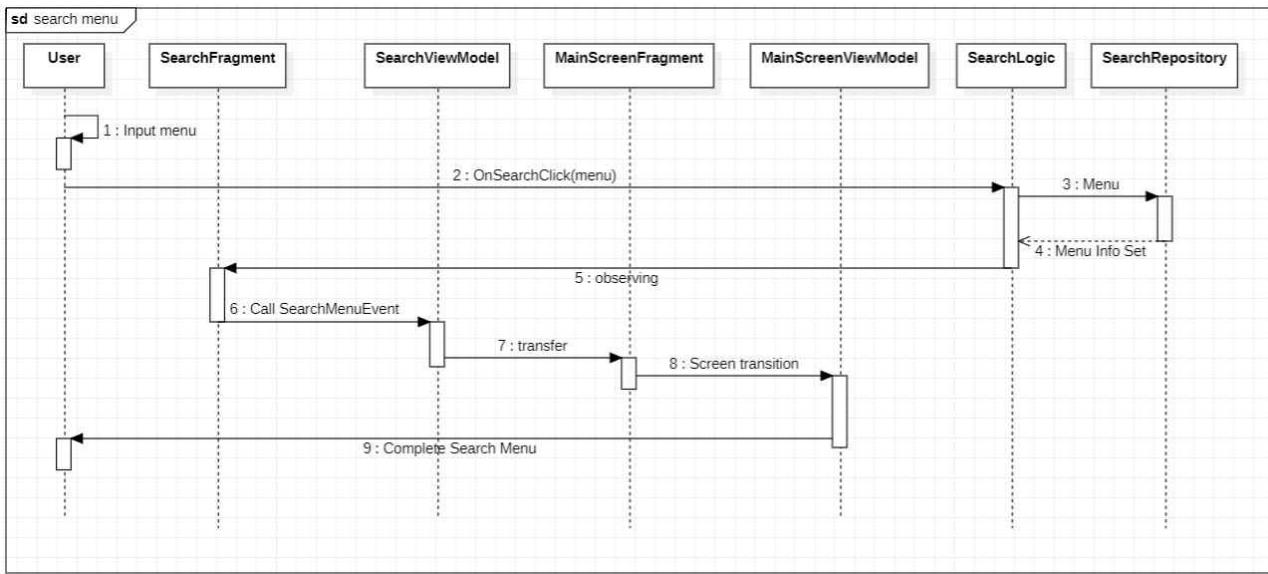


[그림 4-4] reset position SD

고객이 위치를 재검색하기 위한 reset 기능의 use case를 나타내는 sequence diagram이며, <Use Case #3-1b>의 경우이다.

고객이 reset을 의미하는 아이콘 버튼을 클릭하면 위치를 입력하는 검색창의 text를 모두 삭제한다. 빈 검색창을 고객에게 제공한다.

Search Menu

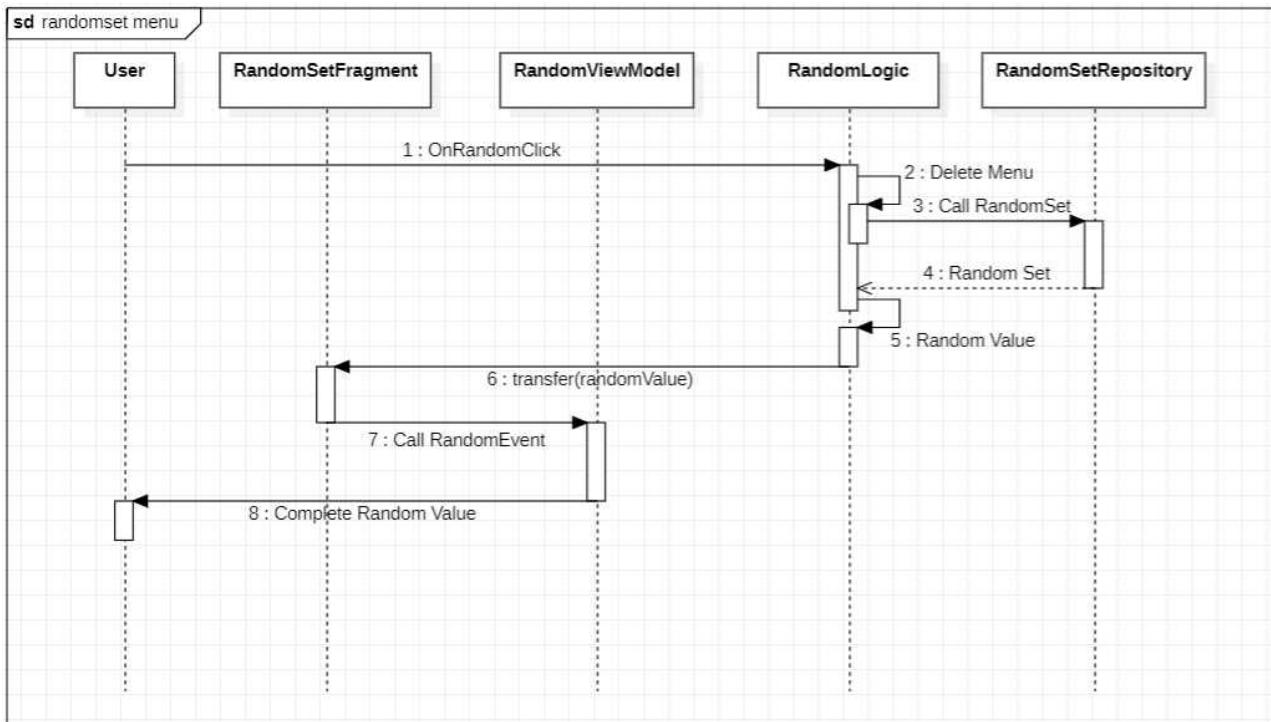


[그림 4-5] search menu SD

고객이 검색하고자 하는 메뉴를 가진 맛집을 검색하는 use case를 나타내는 sequence diagram이며, <Use Case #3-3>의 경우이다.

고객이 메뉴를 입력하고 ‘검색’ 버튼을 클릭하면 해당 메뉴를 Input으로 받는다. 해당 메뉴를 변수로 저장한 뒤에 맛집 API 리스트 내에 해당 메뉴를 가진 맛집들을 Set으로 들고 온다. Main Screen으로 화면을 전환한다. Main Screen에는 Set으로 가져온 해당 메뉴를 가진 맛집들을 추천 리스트로 제공한다.

Randomset Menu

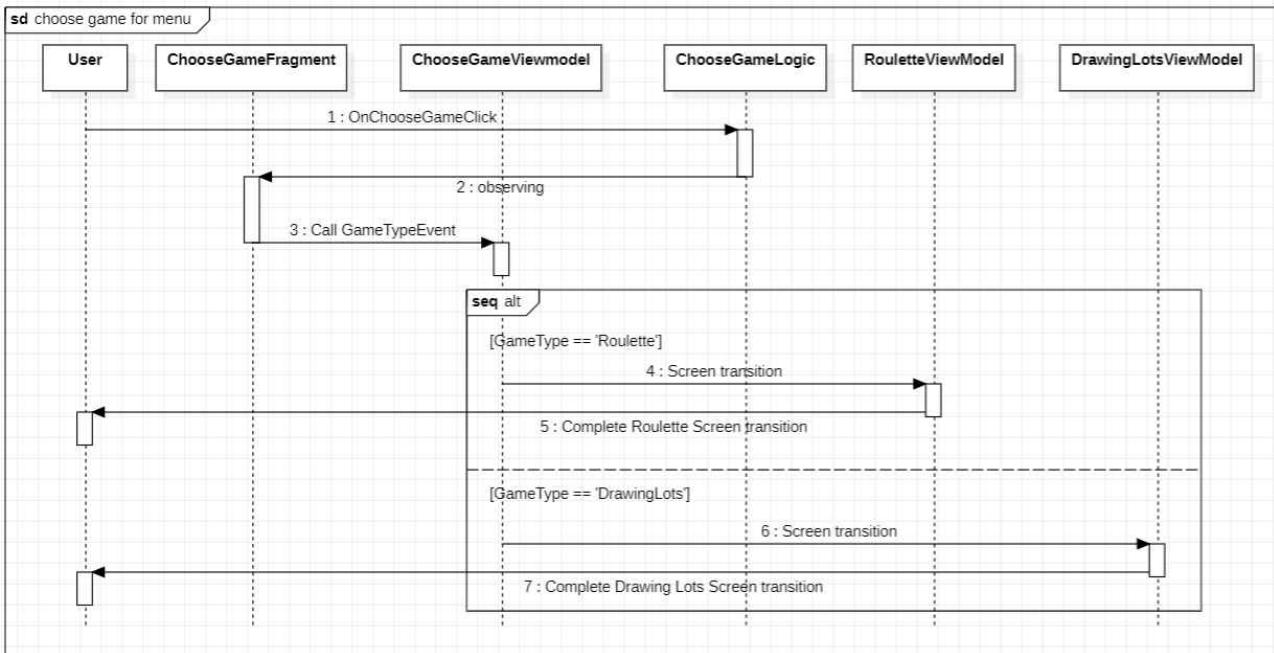


[그림 4-6] randomset menu SD

고객에게 랜덤으로 메뉴를 추천해주는 기능의 use case를 나타내는 sequence diagram이며, <Use Case #3-2b>의 경우이다.

고객이 '랜덤' 버튼을 누르면 우선 메뉴 검색창 내의 text를 삭제한다. 프로그램 내의 랜덤을 위한 메뉴 리스트 중 하나의 메뉴를 랜덤으로 뽑아와 고객에게 제공한다.

Choose Game for Menu

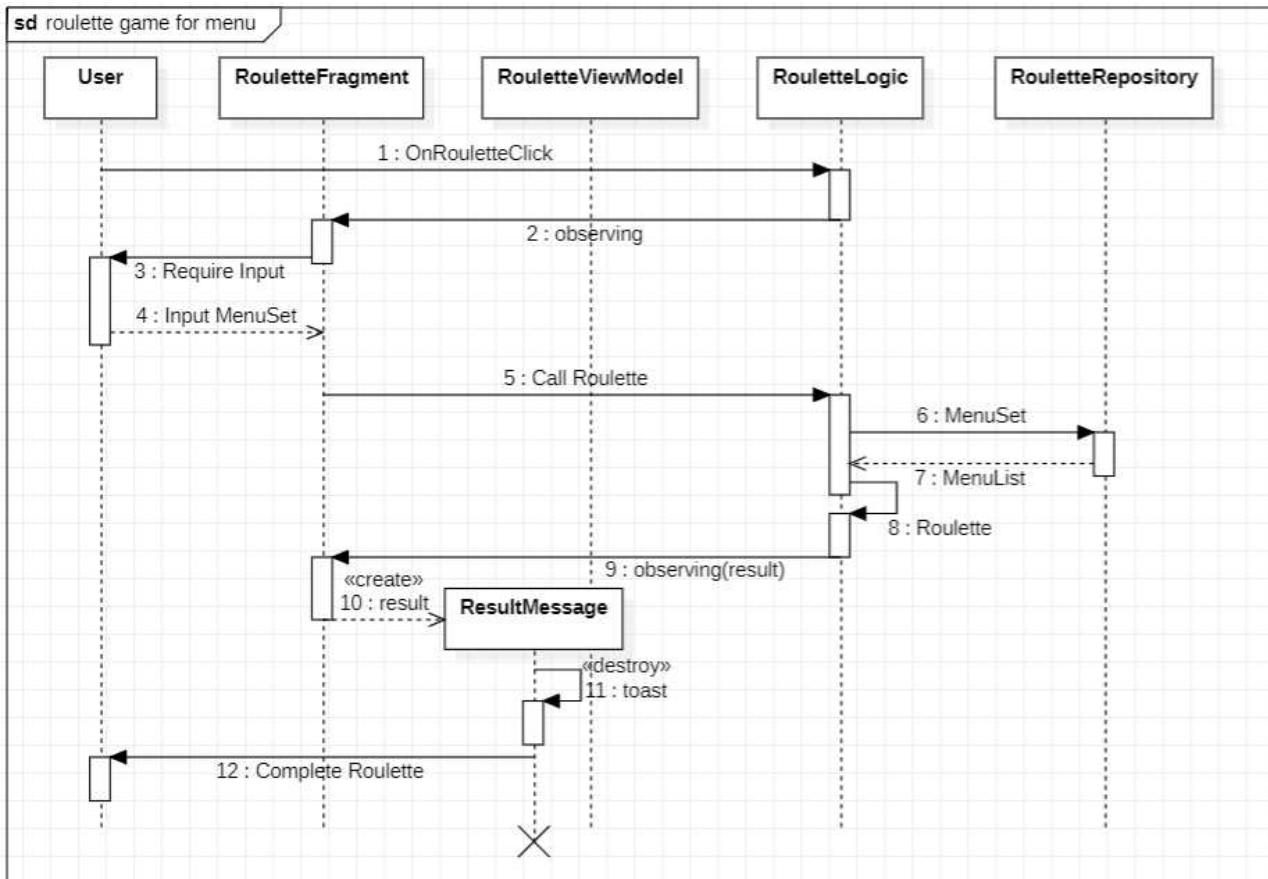


[그림 4-7] choose game for menu SD

고객이 자신의 메뉴를 결정하기 위한 게임(룰렛 돌리기, 제비뽑기)을 고르는 use case를 나타내는 sequence diagram이며, <Use Case #3-2c>의 경우이다.

고객이 ‘게임 정하기’ 버튼을 클릭하면 게임(룰렛 돌리기, 제비뽑기)을 정할 수 있는 화면으로 전환된다. 고객이 ‘룰렛’ 버튼을 클릭하면 룰렛 게임에 대한 화면으로 전환되고, ‘제비뽑기’ 버튼을 클릭하면 제비뽑기에 대한 화면으로 전환된다. (이는 alt문에 대한 설명이다.)

Roulette Game for Menu

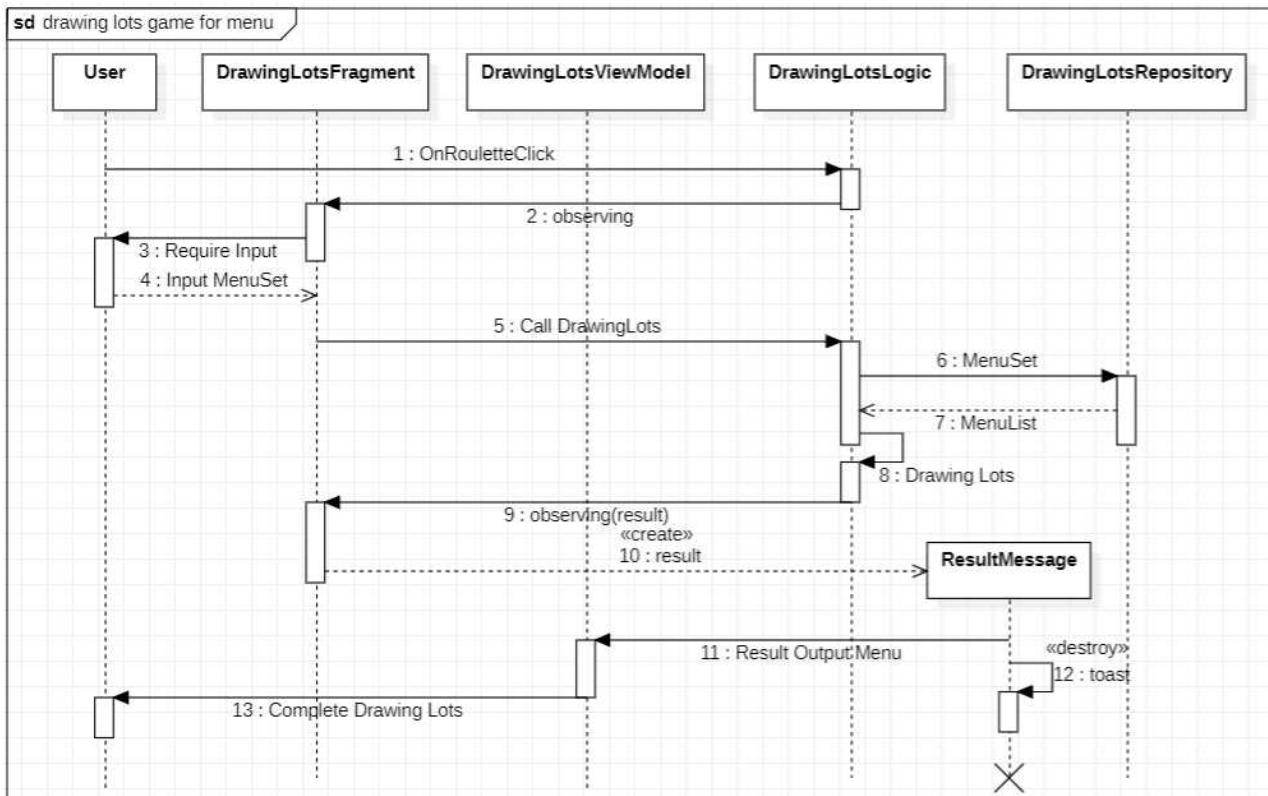


[그림 4-8] roulette game for menu SD

고객이 메뉴를 결정하기 위한 룰렛 게임의 use case를 나타내는 sequence diagram이며, <Use Case #3-2c2>의 경우이다.

고객이 룰렛을 돌릴 6가지 메뉴를 입력하면 해당 메뉴들을 Input Set으로 받는다. 고객이 ‘다음’ 버튼을 누르면 고객이 입력한 6가지 메뉴가 적힌 룰렛이 있는 화면으로 전환된다. 고객이 ‘룰렛 돌리기’ 버튼을 누르면 게임이 시작된다. 룰렛 게임의 결과로 선택된 하나의 메뉴가 toast 메시지로 출력된다.

Drawing lots game for Menu

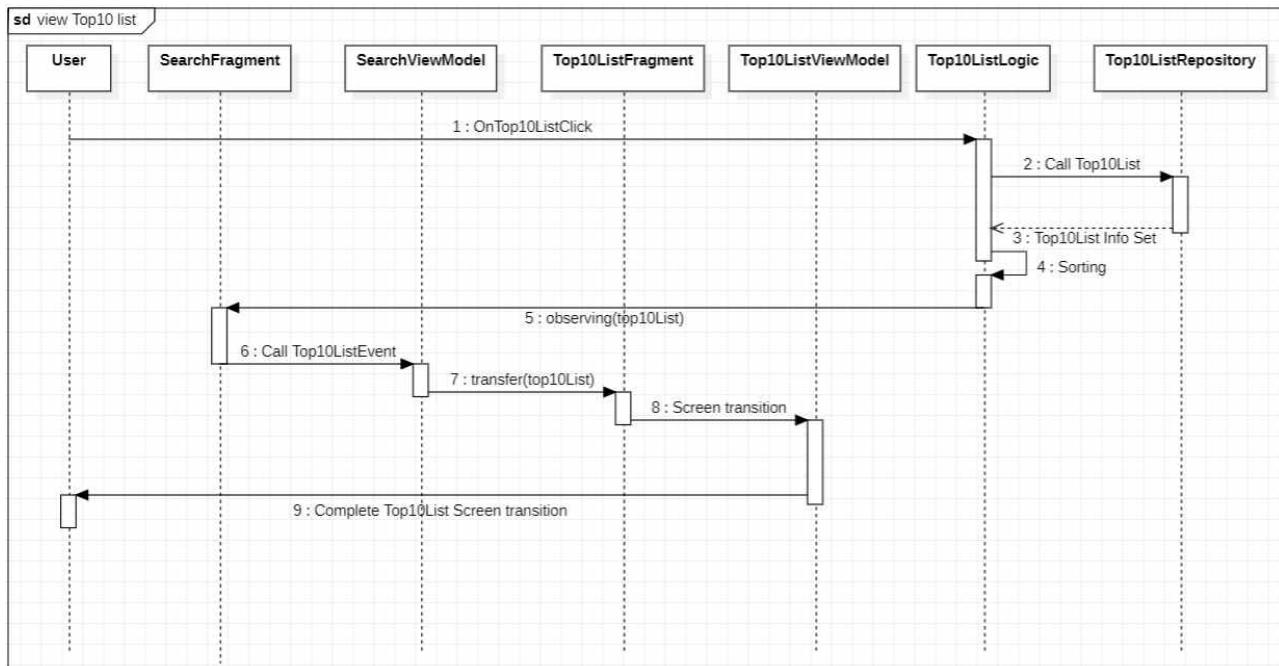


[그림 4-9] drawing lots game for menu SD

고객이 메뉴를 결정하기 위한 제비뽑기 게임의 use case를 나타내는 sequence diagram이며, <Use Case #3-2c2>의 경우이다.

고객이 제비뽑기를 위한 메뉴를 개수 상관없이 입력한다. 고객이 입력한 만큼 리스트에 저장된다. 고객이 ‘제비뽑기’ 버튼을 누르면 고객이 입력한 메뉴 리스트 중 하나의 메뉴를 랜덤으로 뽑아 결괏값으로 저장한다. 제비뽑기 게임의 결과로 선택된 하나의 메뉴가 toast 메시지로 출력된다.

View top10 list

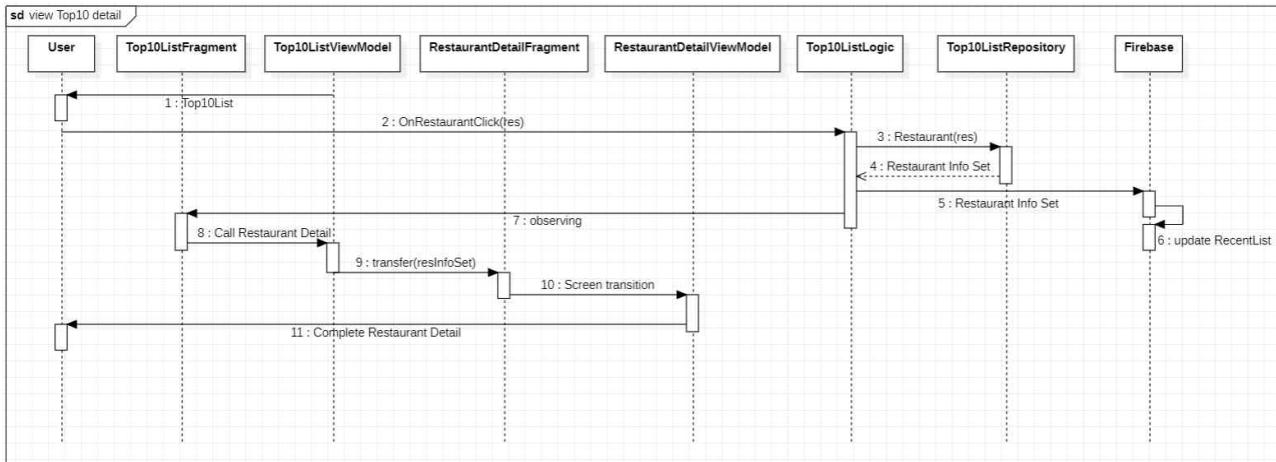


[그림 4-10] view top10 list SD

고객에게 top10 맛집들을 리스트로 제공해주는 기능의 use case를 나타내는 sequence diagram이며, <Use Case #5>의 경우이다.

고객이 'top10' 버튼을 누르면 맛집 API 중 맛집 기준에 대한 top10을 상세 정보와 함께 리스트 Set으로 저장해 들고 온다. 해당 리스트 set을 포함한 해당 화면으로 전환된다. 고객에게 상세 정보가 포함된 맛집의 top10 리스트를 제공한다.

View top10 detail



[그림 4-11] view top10 detail SD

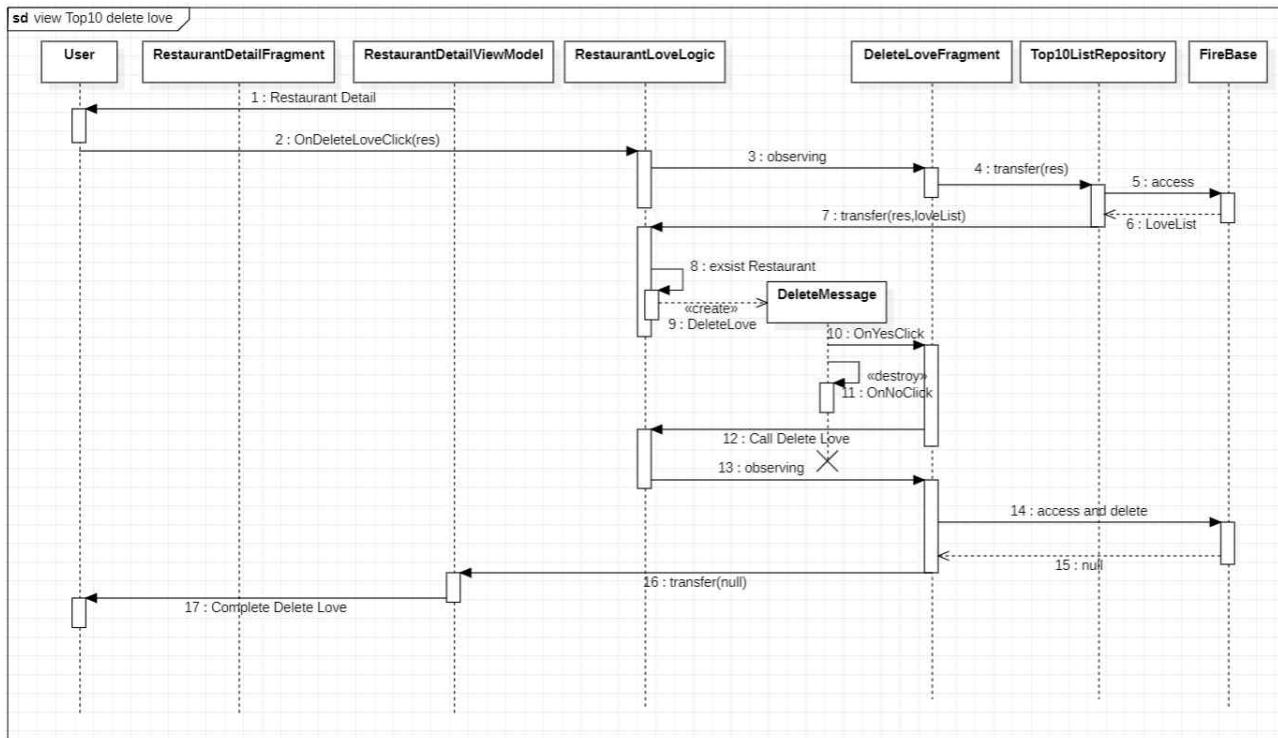
고객에게 제공한 top10 맛집 리스트 내 각 맛집의 상세 정보를 제공해주는 기능의 use case를 나타내는 sequence diagram이며, <Use Case #5-3a>의 경우이다.

고객이 top10 맛집 리스트 중 상세 정보를 보고자 하는 맛집을 클릭한다. 고객이 클릭한 해당 맛집의 이름을 변수로 저장한 뒤, 맛집 API에서 변수와 일치하는 맛집의 상세 정보를 Set으로 가져온다.

이때 해당 프로그램의 ‘최근 본 목록’ 기능을 위해 고객이 열람한 맛집 이름과 상세 정보를 FireBase에 저장한다.

고객이 선택한 맛집의 상세 정보가 담긴 화면으로 전환되고 고객은 원하는 맛집의 상세 정보를 제공받는다.

View top10 delete love



[그림 4-12] view top10 delete love SD

고객에게 제공한 맛집의 상세 정보 중 고객의 저장 목록에 포함된 맛집을 저장 목록에서 삭제하는 기능을 나타내는 sequence diagram이다.

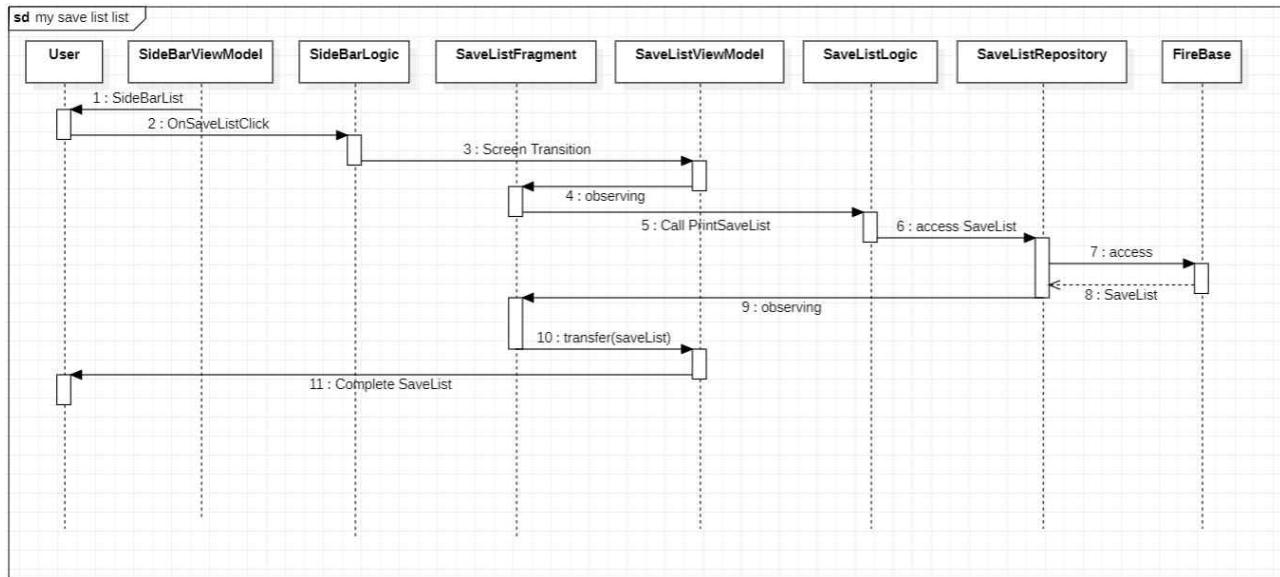
고객이 저장을 취소하는 버튼을 클릭하면 해당 맛집의 이름을 변수로 저장한 뒤, Firebase 내 고객의 저장 목록에 접근해 해당 맛집을 찾는다. 저장 목록에 해당 맛집이 존재하면 고객에게 한 번 더 삭제 여부를 묻는다.

고객이 'Yes' 버튼을 클릭하면 다시 Firebase 내 저장 목록에 접근해 해당 맛집에 대한 삭제를 진행한다.

고객이 'No' 버튼을 클릭하면 어떠한 이벤트도 일어나지 않고 메시지는 삭제된다.

4.3 User SideBar Sequence diagram

My Save List View

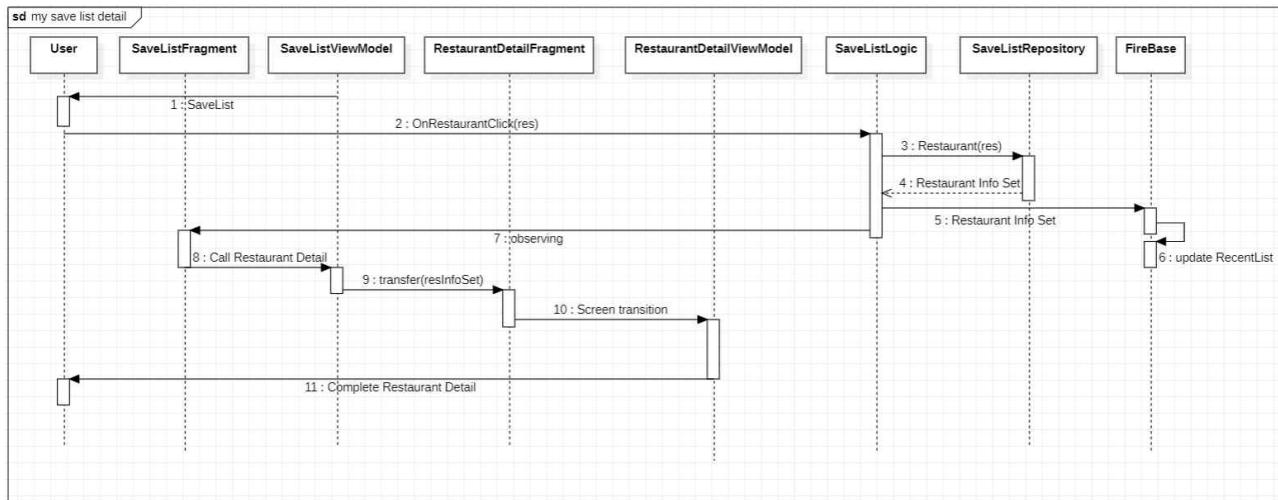


[그림 4-13] my save list view SD

고객이 저장한 맛집들이 포함된 저장 목록을 고객에게 제공하는 기능의 use case를 나타내는 sequence diagram이며, <Use Case #7>의 경우이다.

고객이 사이드바에서 ‘나의 저장 목록’을 클릭하면 해당 화면으로 전환된다. Firebase 내 고객의 저장 목록에 접근해 저장 목록을 리스트로 들고 온다. 저장 목록 리스트를 해당 화면에 출력해 고객에게 저장 목록을 제공한다.

My Save List Detail



[그림 4-14] my save list detail SD

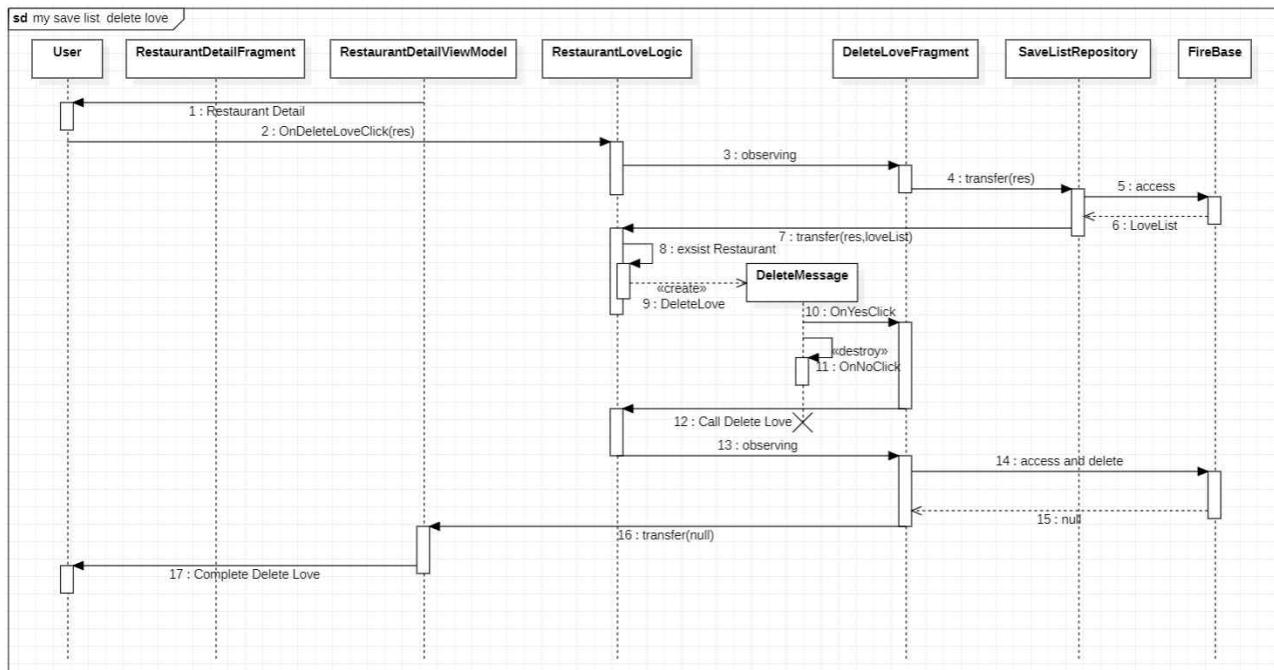
고객의 저장 목록 리스트 내 각 맛집의 상세 정보를 제공해주는 기능의 use case를 나타내는 sequence diagram이다.

고객이 저장 목록 리스트 중 상세 정보를 보고자 하는 맛집을 클릭한다. 고객이 클릭한 해당 맛집의 이름을 변수로 저장한 뒤, 맛집 API에서 변수와 일치하는 맛집의 상세 정보를 Set으로 가져온다.

이때 해당 프로그램의 ‘최근 본 목록’ 기능을 위해 고객이 열람한 맛집 이름과 상세 정보를 FireBase에 저장한다.

고객이 선택한 맛집의 상세 정보가 담긴 화면으로 전환되고 고객은 원하는 맛집의 상세 정보를 제공받는다.

My Save List Delete Love



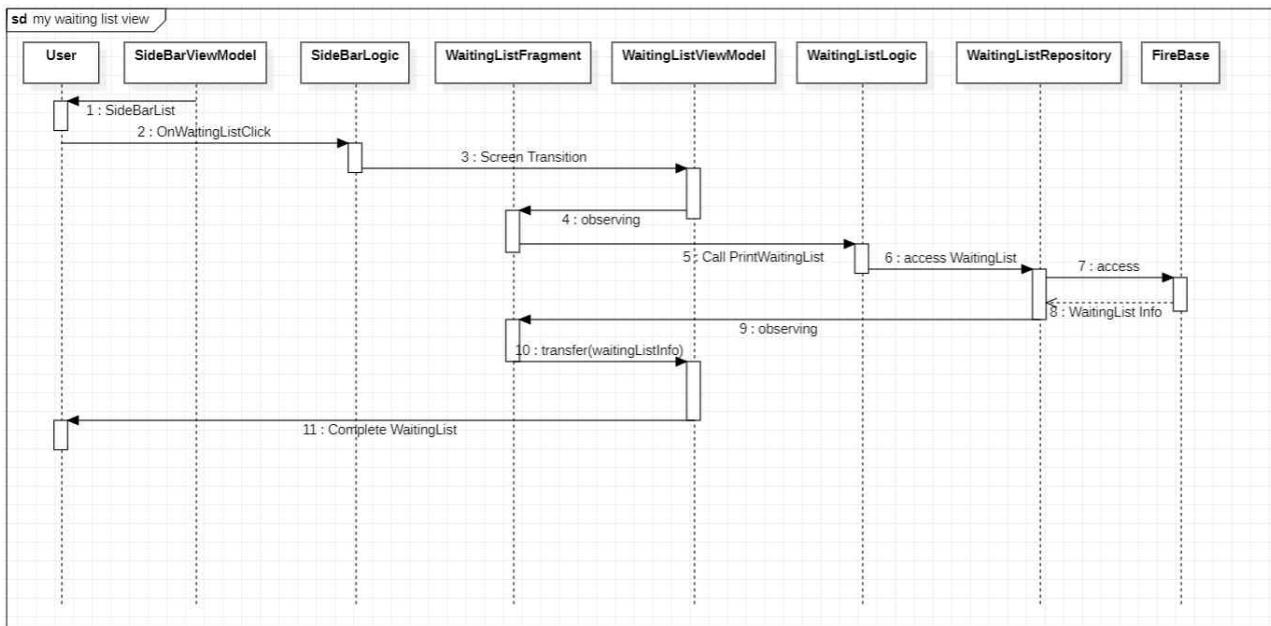
[그림 4-15] my save list delete love SD

고객에게 제공한 맛집의 상세 정보 중 고객의 저장 목록에 포함된 맛집을 저장 목록에서 삭제하는 기능을 나타내는 sequence diagram이며, <Use Case #7-4a>의 경우이다. 고객이 저장을 취소하는 버튼을 클릭하면 해당 맛집의 이름을 변수로 저장한 뒤, Firebase 내 고객의 저장 목록에 접근해 해당 맛집을 찾는다. 저장 목록에 해당 맛집이 존재하면 고객에게 한 번 더 삭제 여부를 묻는다.

고객이 'Yes' 버튼을 클릭하면 다시 Firebase 내 저장 목록에 접근해 해당 맛집에 대한 삭제를 진행한다.

고객이 'No' 버튼을 클릭하면 어떠한 이벤트도 일어나지 않고 메시지는 삭제된다.

My Waiting List View

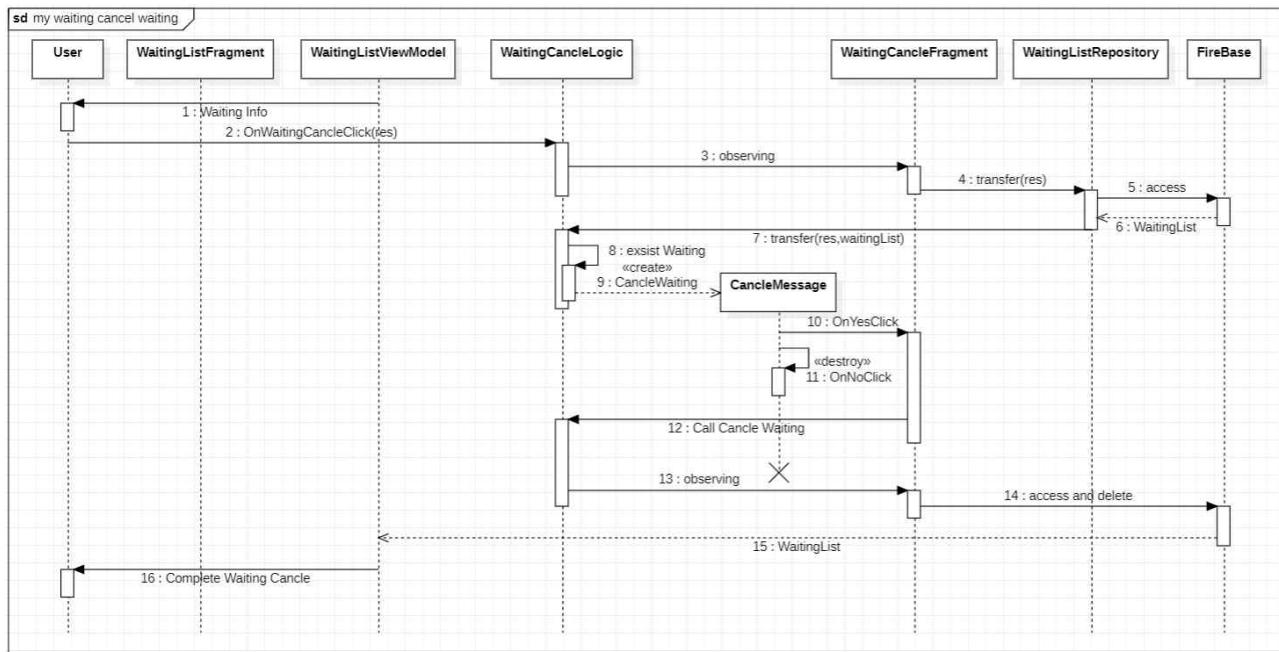


[그림 4-16] my waiting list view SD

고객의 맛집 대기 신청현황을 고객에게 제공하는 기능의 use case를 나타내는 sequence diagram이며, <Use Case #8-5>의 경우이다.

고객이 사이드바에서 ‘대기 현황’을 클릭하면 해당 화면으로 전환된다. Firebase 내 고객의 대기 현황이 담긴 리스트에 접근해 대기 현황을 리스트로 들고 온다. 대기 현황 리스트를 해당 화면에 출력해 고객에게 대기 현황을 제공한다.

My Waiting list Cancel Waiting



[그림 4-17] my waiting list cancel waiting SD

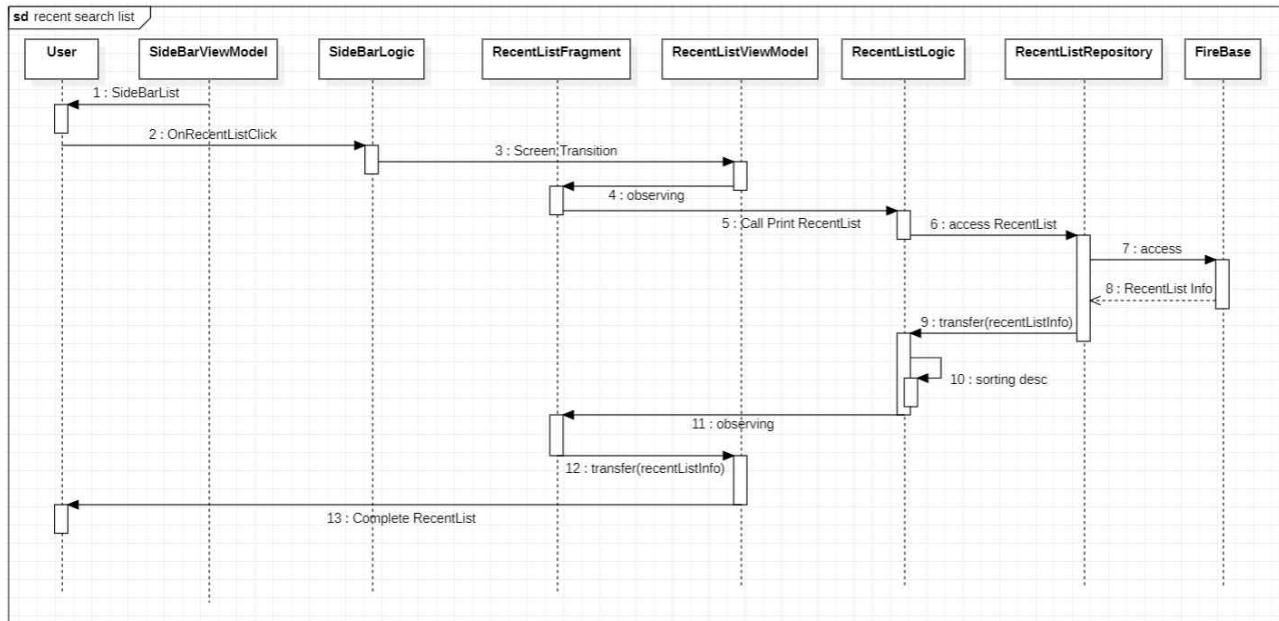
고객이 대기 신청했던 맛집에 대한 대기를 취소하는 기능을 나타내는 sequence diagram이며, <Use Case #8-6>의 경우이다.

고객이 대기를 취소하는 버튼을 클릭하면 해당 맛집의 이름을 변수로 저장한 뒤, Firebase 내 고객의 대기 현황 리스트에 접근해 해당 맛집을 찾는다. 대기 현황 리스트에 해당 맛집이 존재하면 고객에게 한 번 더 삭제 여부를 묻는다.

고객이 'Yes' 버튼을 클릭하면 다시 Firebase 내 대기 현황 리스트에 접근해 해당 맛집에 대한 대기 취소를 진행한다.

고객이 'No' 버튼을 클릭하면 어떠한 이벤트도 일어나지 않고 메시지는 삭제된다.

Recent Search List



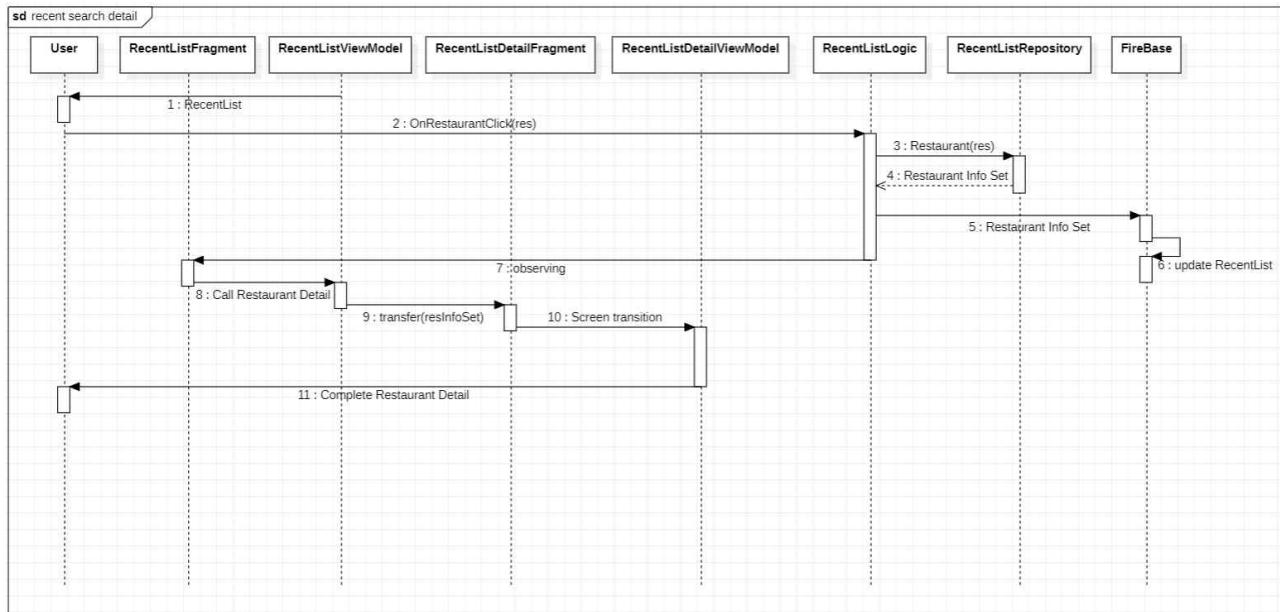
[그림 4-18] recent search list SD

고객에게 최근 본 맛집들을 제공해주는 기능의 use case를 나타내는 sequence diagram이며, <Use Case #9>의 경우이다.

고객이 프로그램을 이용하면서 상세 정보를 열람한 맛집 가게들을 고객에게 리스트로 제공해준다.

고객이 사이드바에서 '최근 본 목록'을 클릭하면 해당 화면으로 전환된다. Firebase 내 고객이 최근 본 맛집 리스트에 접근해 최근 본 목록을 리스트로 들고 온다. 최근 본 목록 리스트를 해당 화면에 출력해 고객에게 해당 기능을 제공한다.

Recent Search Detail



[그림 4-19] recent search detail SD

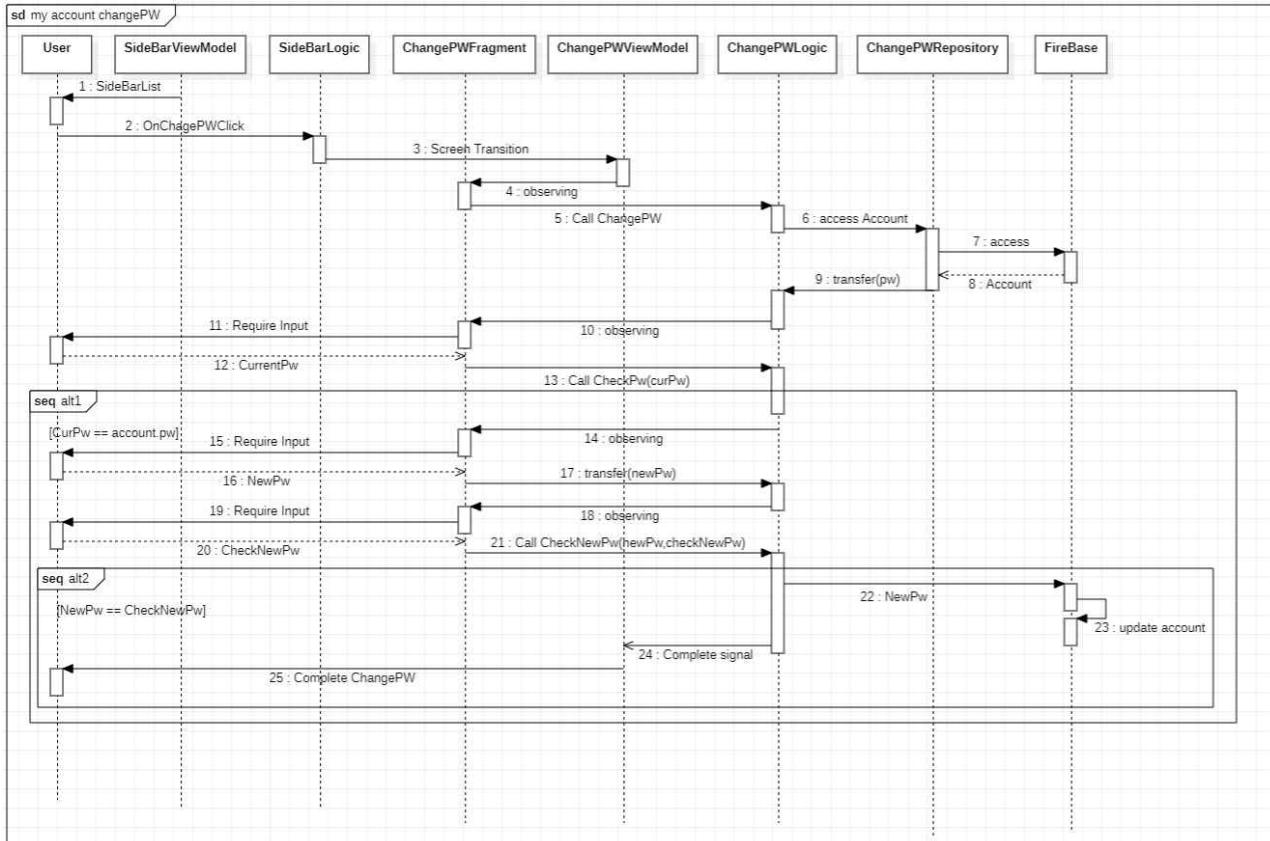
고객의 최근 본 목록 리스트 내 각 맛집의 상세 정보를 제공해주는 기능의 use case를 나타내는 sequence diagram이며, <Use Case #9-4>의 경우이다.

고객이 최근 본 목록 리스트 중 상세 정보를 보고자 하는 맛집을 선택한다. 고객이 선택한 해당 맛집의 이름을 변수로 저장한 뒤, 맛집 API에서 변수와 일치하는 맛집의 상세 정보를 Set으로 가져온다.

이때 해당 프로그램의 ‘최근 본 목록’ 기능을 위해 고객이 열람한 맛집 이름과 상세 정보를 FireBase에 저장한다.

고객이 선택한 맛집의 상세 정보가 담긴 화면으로 전환되고 고객은 원하는 맛집의 상세 정보를 제공받는다.

My Account Change Password



[그림 4-20] my account change password SD

고객의 비밀번호를 변경하는 기능의 use case를 나타내는 sequence diagram이며, <Use Case #10-3>의 경우이다.

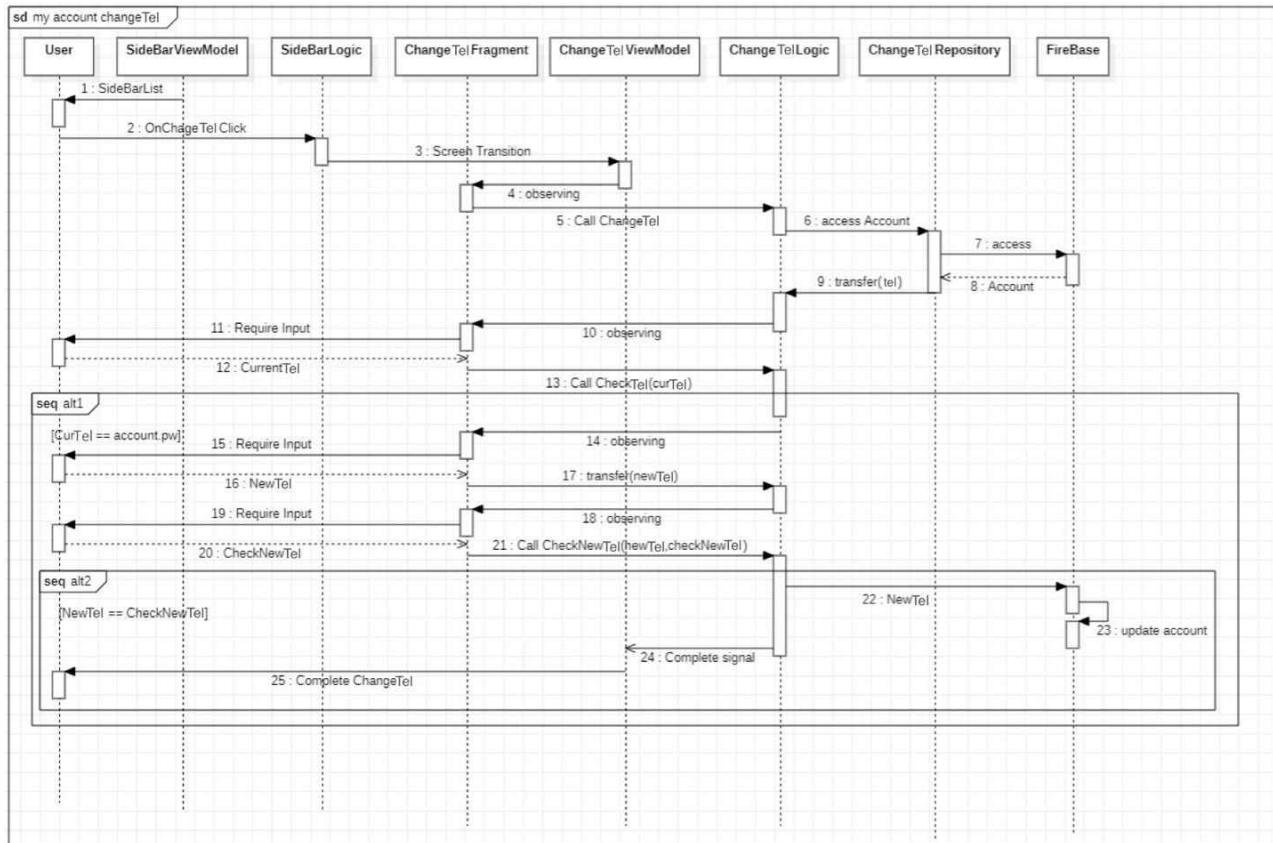
고객이 사이드바에서 ‘내 계정’을 클릭하면 계정에 대한 기능을 제공하는 화면으로 전환된다. 계정에 대한 기능 중 ‘비밀번호 변경’ 버튼을 클릭하면 해당 화면으로 전환된다.

FireBase 내 고객의 정보에 접근해 현재 비밀번호를 들고 온다.

비밀번호 변경을 위해 고객이 기존 비밀번호를 입력해 ‘확인’ 버튼을 누른다. 입력한 비밀번호와 FireBase에 저장되어 있던 현재 비밀번호가 일치하면 변경할 새 비밀번호를 입력한다. (이는 alt1문에 대한 설명이다.)

새 비밀번호를 입력하고 확인을 위해 새 비밀번호를 한 번 더 입력한 뒤, ‘변경’ 버튼을 누르면 새 비밀번호와 확인을 위해 다시 입력한 새 비밀번호가 일치하는지 검사한다. 일치하면 새 비밀번호를 변수로 저장해 들고 와 FireBase에 접근해 고객의 계정 내 비밀번호를 새 비밀번호로 변경하고 계정을 업데이트한다. (이는 alt2문에 대한 설명이다.)

My Account Change Tel



[그림 4-21] my account change tel SD

고객의 전화번호를 변경하는 기능의 use case를 나타내는 sequence diagram이며, <Use Case #10-5>의 경우이다.

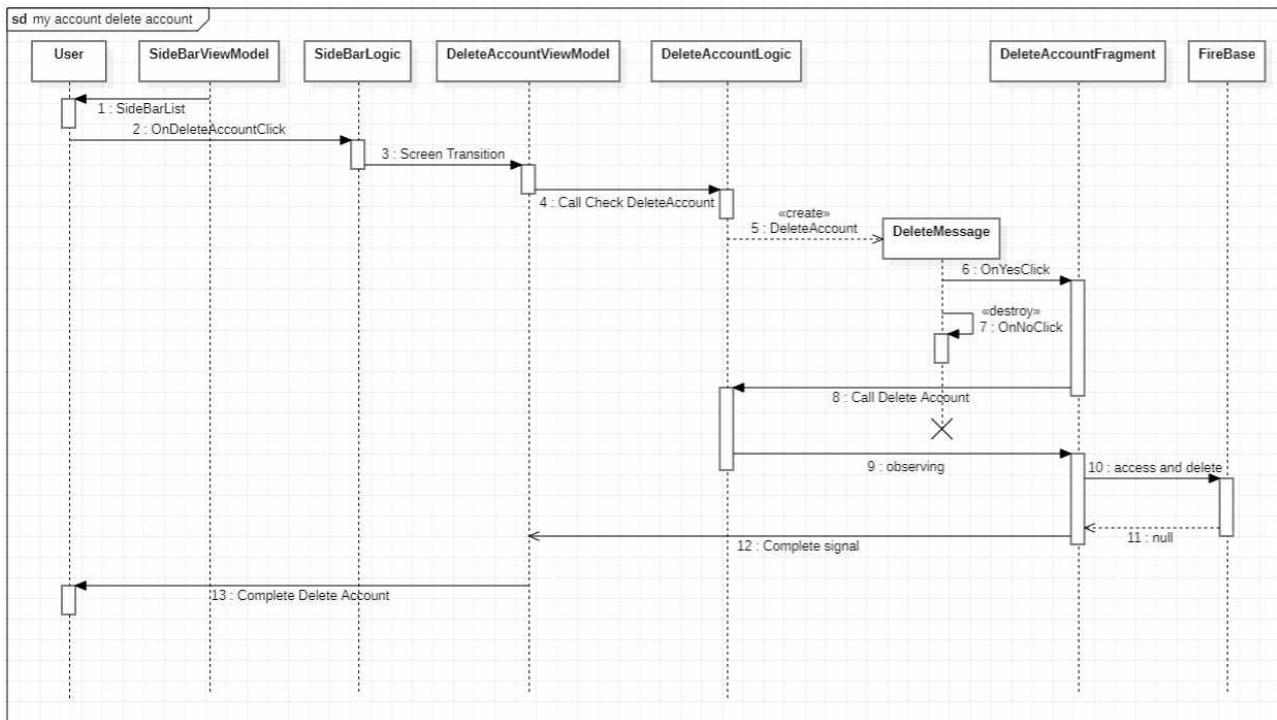
고객이 사이드바에서 ‘내 계정’을 클릭하면 계정에 대한 기능을 제공하는 화면으로 전환된다. 계정에 대한 기능 중 ‘전화번호 변경’ 버튼을 클릭하면 해당 화면으로 전환된다.

FireBase 내 고객의 정보에 접근해 현재 전화번호를 들고 온다.

전화번호 변경을 위해 고객이 기존 전화번호를 입력해 ‘확인’ 버튼을 누른다. 입력한 전화번호와 FireBase에 저장되어 있던 현재 전화번호가 일치하면 변경할 새 전화번호를 입력한다. (이는 alt1문에 대한 설명이다.)

새 전화번호를 입력하고 확인을 위해 새 전화번호를 한 번 더 입력한 뒤, ‘변경’ 버튼을 누르면 새 전화번호와 확인을 위해 다시 입력한 새 전화번호가 일치하는지 검사한다. 일치하면 새 전화번호를 변수로 저장해 들고 와 FireBase에 접근해 고객의 계정 내 전화번호를 새 전화번호로 변경하고 계정을 업데이트한다. (이는 alt2문에 대한 설명이다.)

My Account Delete Account



[그림 4-22] my account delete account SD

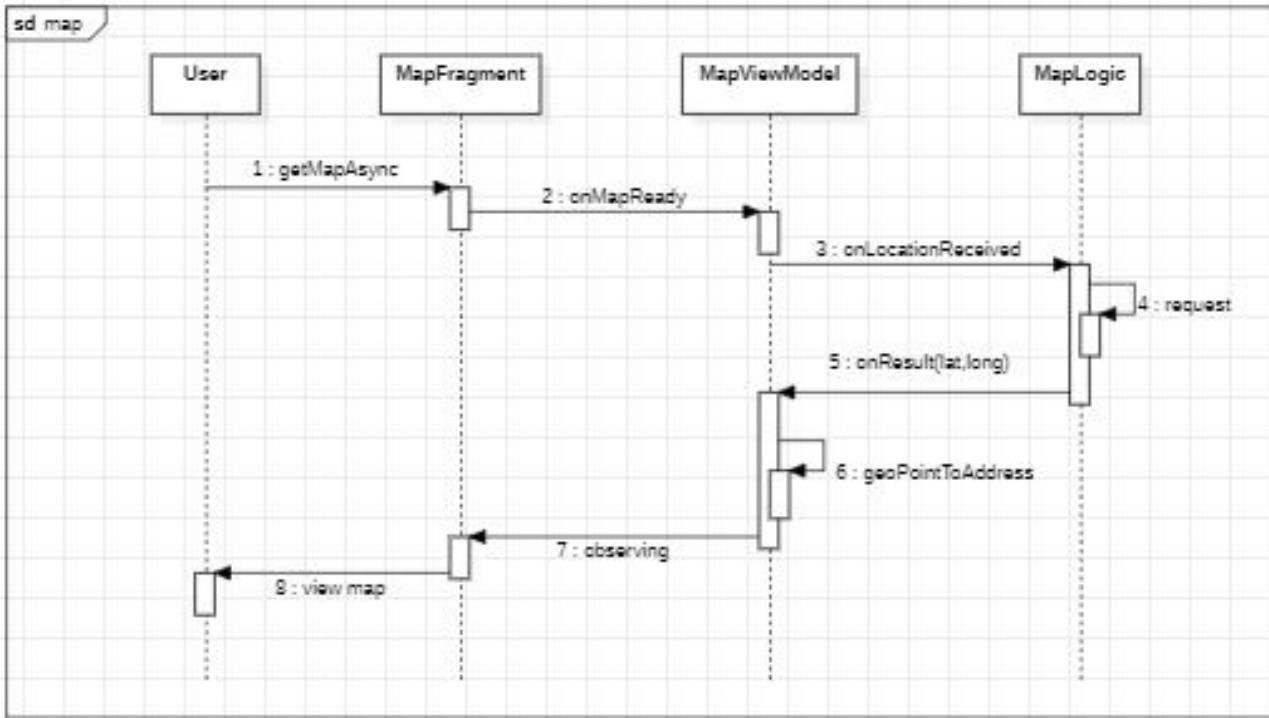
고객이 계정을 탈퇴하는 기능을 나타내는 sequence diagram이며, <Use Case #10-7>의 경우이다.

고객이 사이드바에서 ‘내 계정’을 클릭하면 계정에 대한 기능을 제공하는 화면으로 전환된다. 계정에 대한 기능 중 ‘계정 탈퇴’ 버튼을 클릭하면 고객에게 한 번 더 탈퇴 여부를 묻는다.

고객이 ‘Yes’ 버튼을 클릭하면 다시 Firebase 내 고객의 계정에 접근해 고객에 대한 모든 정보를 삭제하고 계정 탈퇴가 진행된다.

고객이 ‘No’ 버튼을 클릭하면 어떠한 이벤트도 일어나지 않고 메시지는 삭제된다.

Map

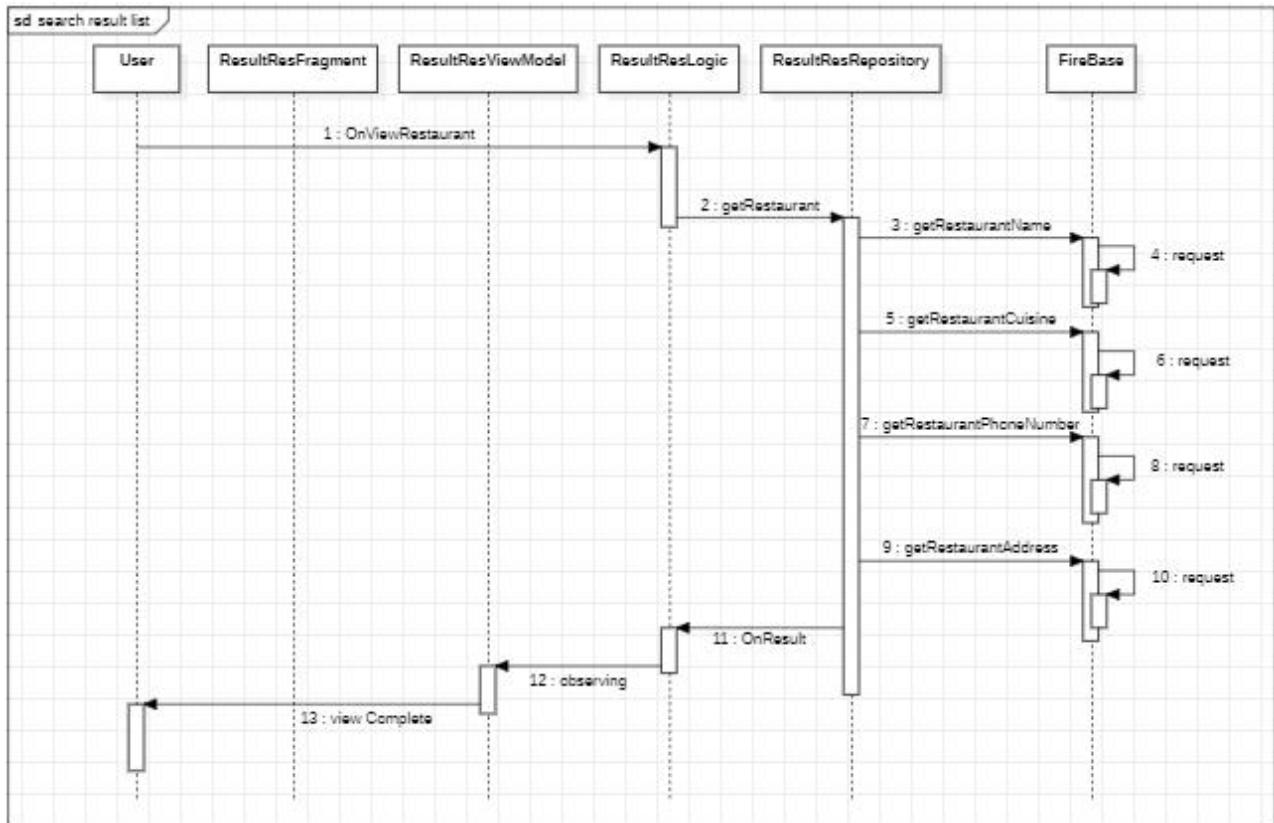


[그림 4-23] Map SD

사용자가 메뉴를 검색했을 때 전환되는 화면에서 나타나는 지도이며, <Use Case #5-2>의 경우이다.

지도 아래의 식당 리스트들이 지도의 핀포인트로 띄워지게 된다. API에 등록된 식당들의 주소 정보를 불러와 이를 위도와 경도값으로 변경하여 지도에 포인트로 띄운다. 지도 화면의 경우, 터치를 통해 지도의 기준점을 옮길 수 있다.

Search Result List

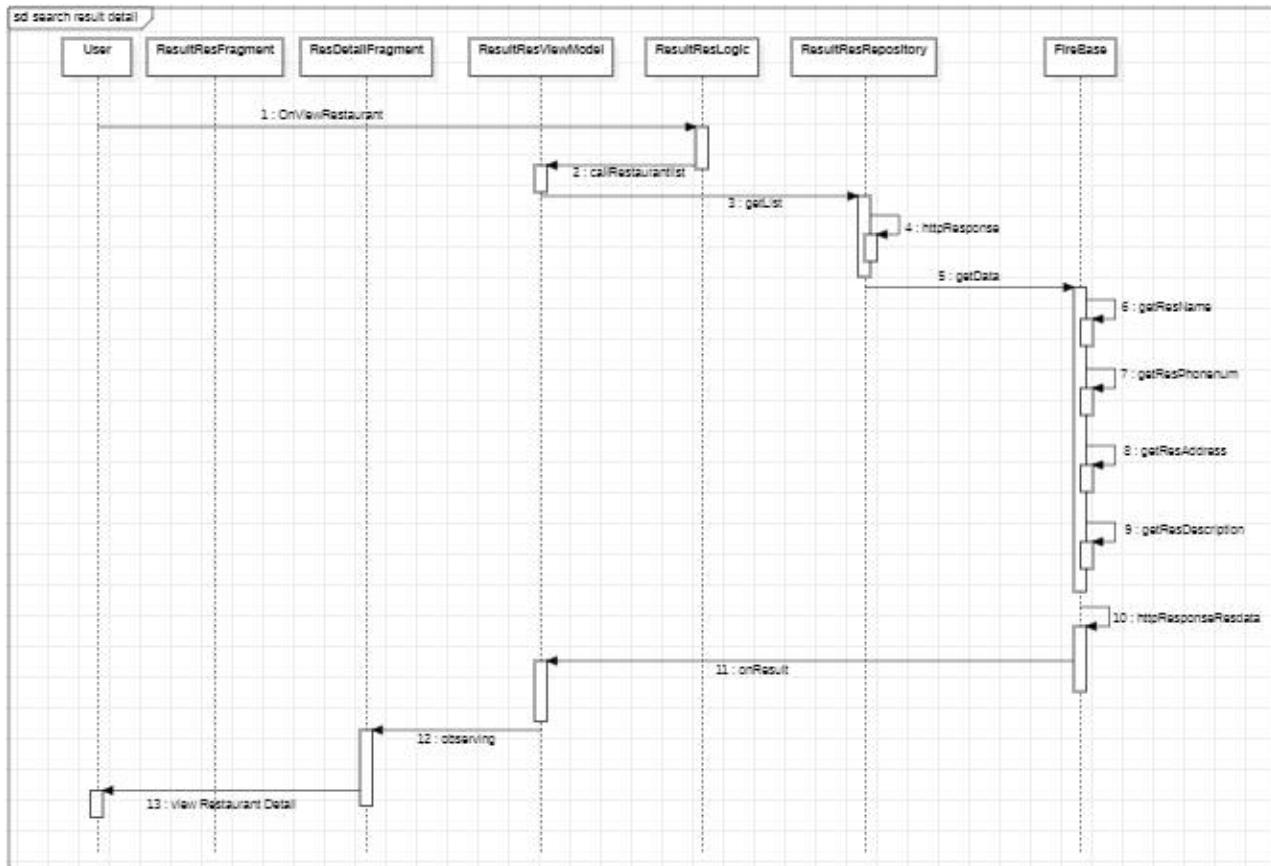


[그림 4-24] Search Result List SD

사용자가 메뉴를 검색했을 때 전환되는 화면에서 보여지는 식당들의 리스트 뷰이며, <Use Case #5>의 경우이다.

화면이 전환되기 전에 입력한 메뉴와 위치 정보에 대한 값에 알맞은 식당을 API에서 검색하여 불러온다. 불러온 식당의 정보 중 필요한 값을 추출하여 리스트 뷰에 보이게 하고, 검색 결과에 해당하는 식당이 여러 개가 있을 경우, 이를 리스트 형식으로 보이게 하여 화면에 출력한다.

Search Result Detail

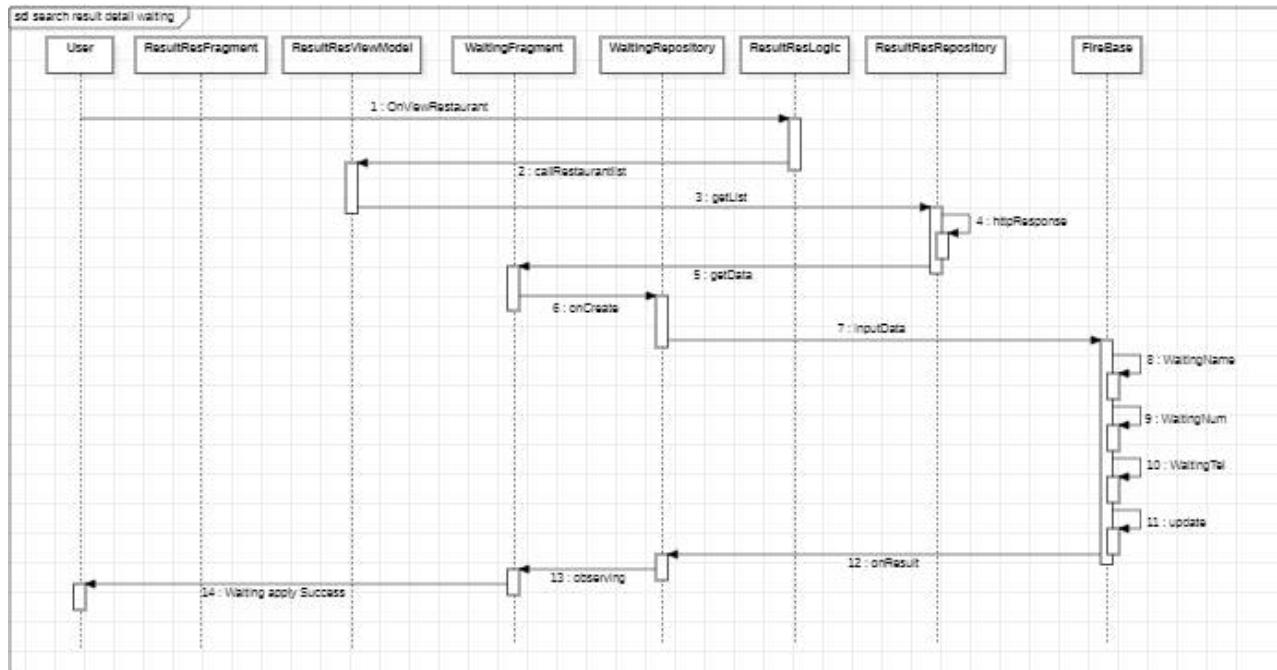


[그림 4-25] Search Result Detail SD

사용자가 식당을 선택했을 때 전환되는 식당에 상세 정보 페이지이며, <Use Case #5-3>의 경우이다.

사용자가 검색한 결과에 해당하는 식당들이 리스트 뷰로 보여지고, 그 중 하나를 선택할 경우, 해당 가게에 대한 이름, 전화번호, 주소, 카테고리, 대표 메뉴, 찜 개수가 출력되는 화면을 띄운다. 식당의 이름, 전화번호, 주소, 카테고리, 대표 메뉴에 대한 정보는 API에서 불러오며, 찜 개수의 경우 FireStore에서 값을 불러온다.

Search Result Detail Waiting

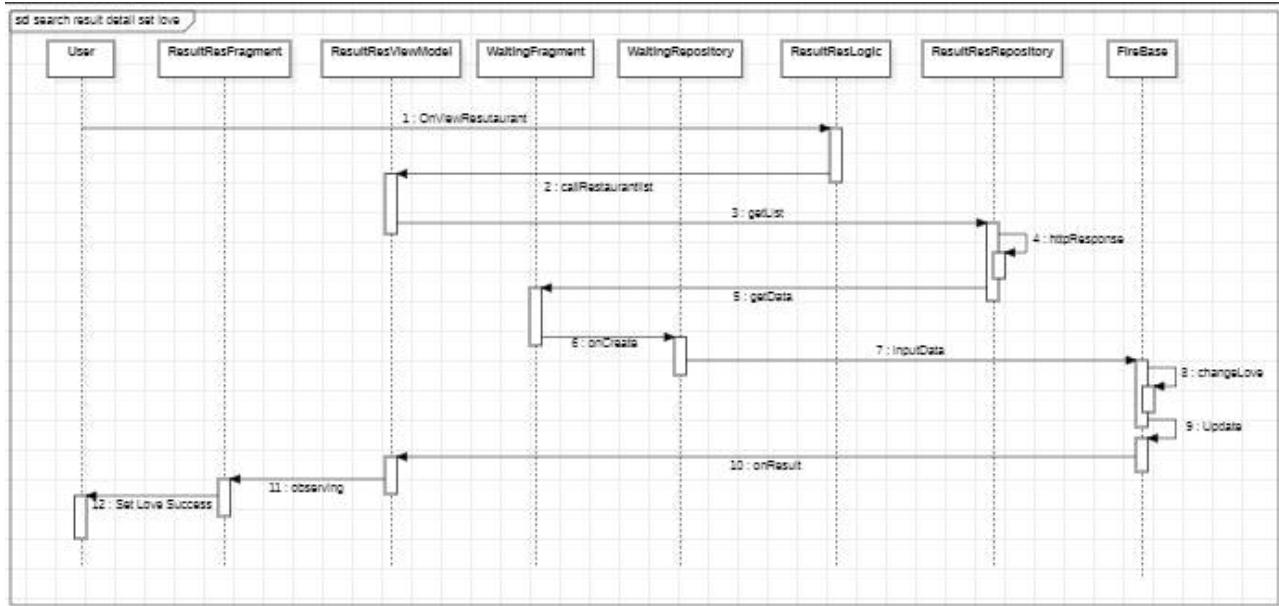


[그림 4-26] Search Result Detail Waiting SD

사용자가 가게 상세 페이지 화면에서 줄서기 버튼을 눌렀을 때 전환되는 화면이며, <Use Case #8>의 경우이다.

사용자가 가게 상세 페이지에서 줄서기 버튼을 눌렀을 때, 줄서기 신청자의 이름, 줄서기 전체 인원수, 신청자의 전화번호를 입력하고 확인 버튼을 누르면 현재 식당에 등록된 대기 팀의 수를 알려주는 창이 뜨고, 확인 버튼을 누르게 되면, FireStore에 해당 정보가 저장되게 된다.

Search Result Detail Set Love

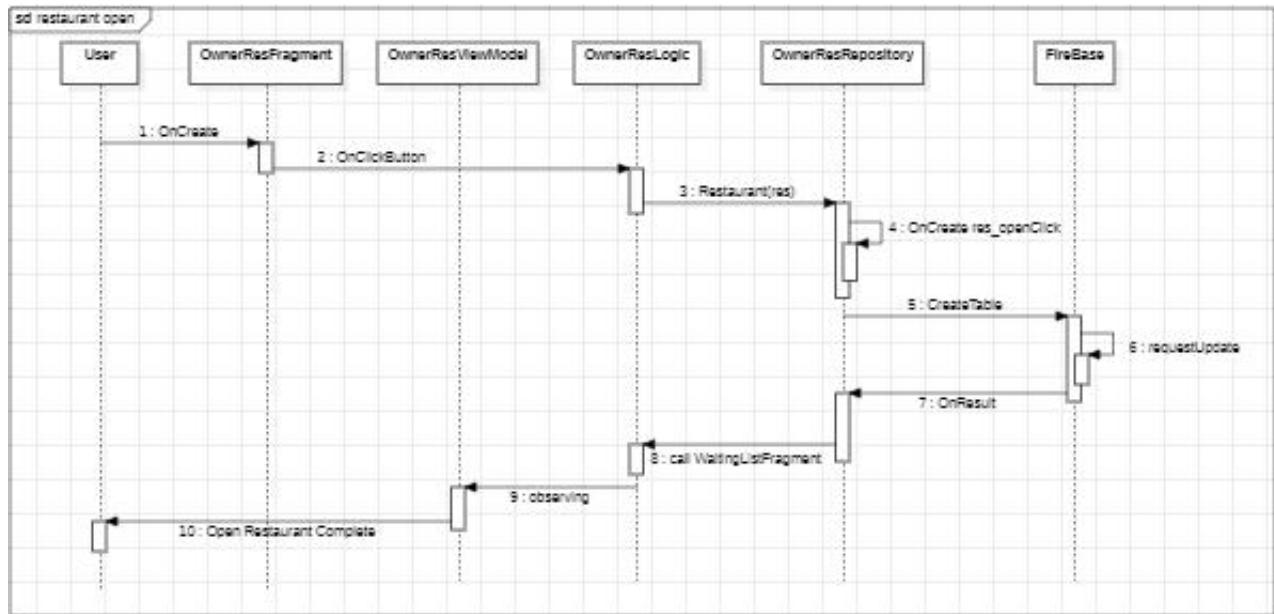


[그림 4-27] Search Result Detail Set Love SD

사용자가 가게 상세 페이지 화면에서 가게 찜하기 버튼을 눌렀을 때 실행되는 기능이며, <Use Case #7-3>의 경우이다.

사용자가 가게 상세 페이지에서 찜하기 버튼을 눌렀을 때, 해당 가게를 자신의 저장목록 리스트에 추가하게 된다. 이는 FireStore에 저장되어 있는 자신의 계정 정보 안에 값이 저장되게 된다.

Restaurant Open

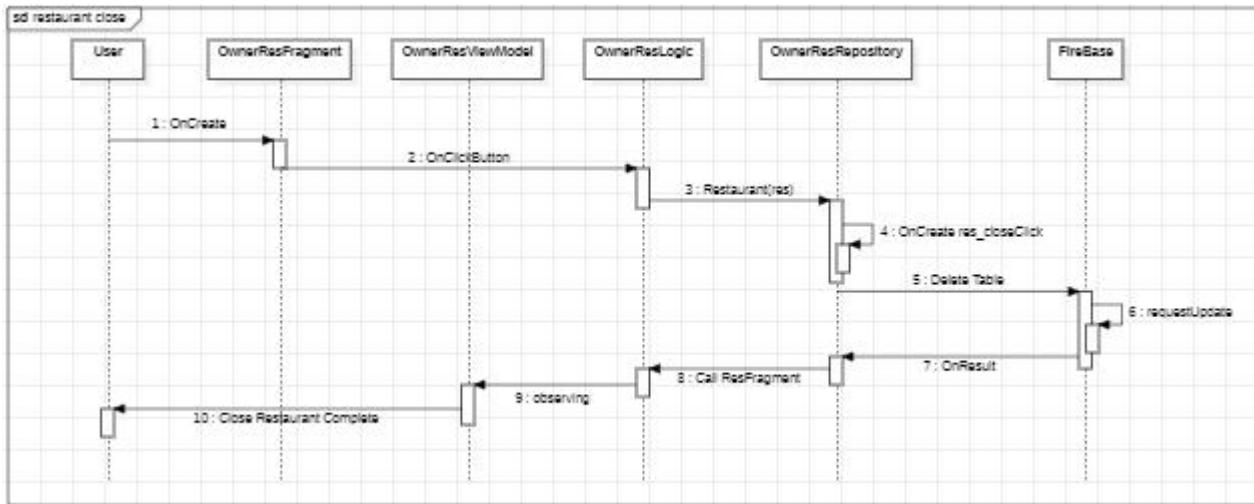


[그림 4-28] Restaurant Open SD

사용자가 사업자의 계정으로 로그인했을 때 보이는 화면이며, <Use Case #11-3>의 경우이다.

사용자가 자신의 식당의 대기 여부를 가능하게 하려고 할 때 실행되는 기능으로, Open 버튼을 누르면 일반 고객이 자신의 식당에 대기 신청이 가능하게 된다. 사용자가 Open 을 누르지 않는다면, 일반 고객은 해당 식당에 대기 신청이 불가능하다.

Restaurant Close

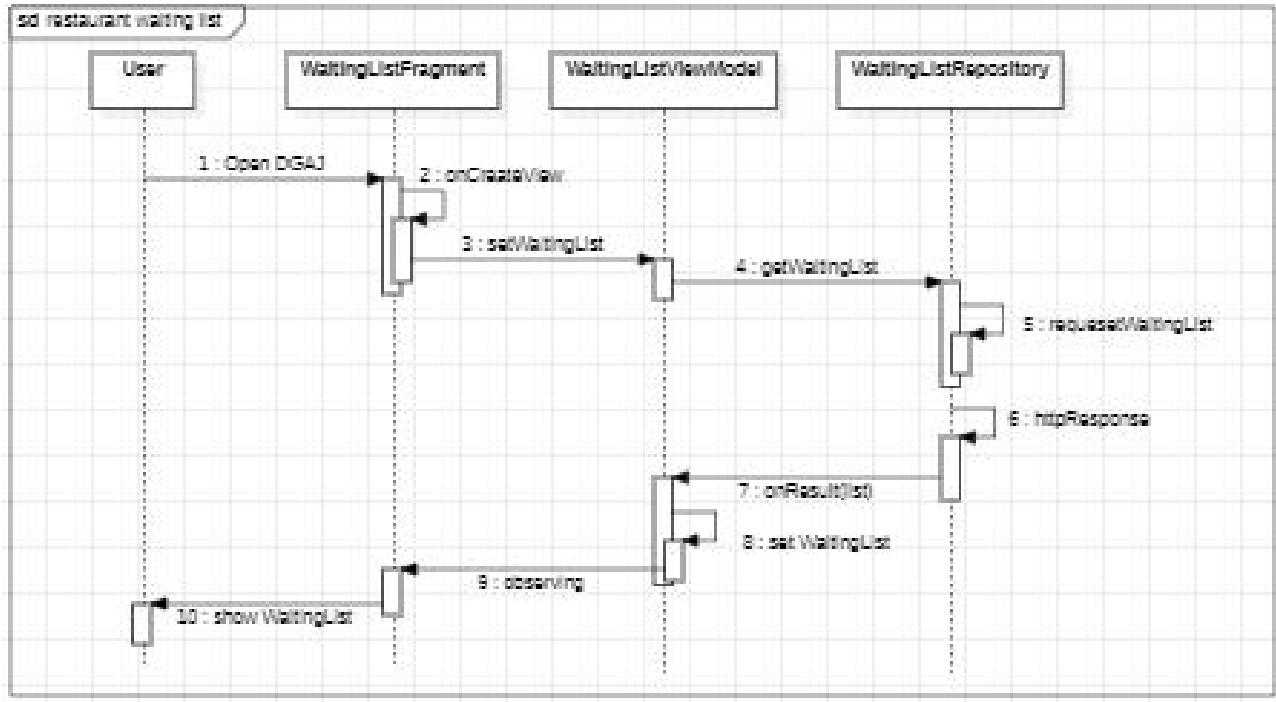


[그림 4-29] Restaurant Close SD

사용자가 사업자의 계정으로 로그인했을 때 보이는 화면의 하단에 있는 버튼으로 기능하며, <Use Case #11-10>의 경우이다.

사용자가 화면 하단의 Close버튼을 누르면 기능이 시작된다. 사용자가 자신의 식당의 대기 여부를 불가능하게 하려고 할 때 실행되는 기능으로, Close버튼을 누르면 일반 고객이 자신의 식당에 대기 신청이 불가능하게 된다.

Restaurant Waiting list

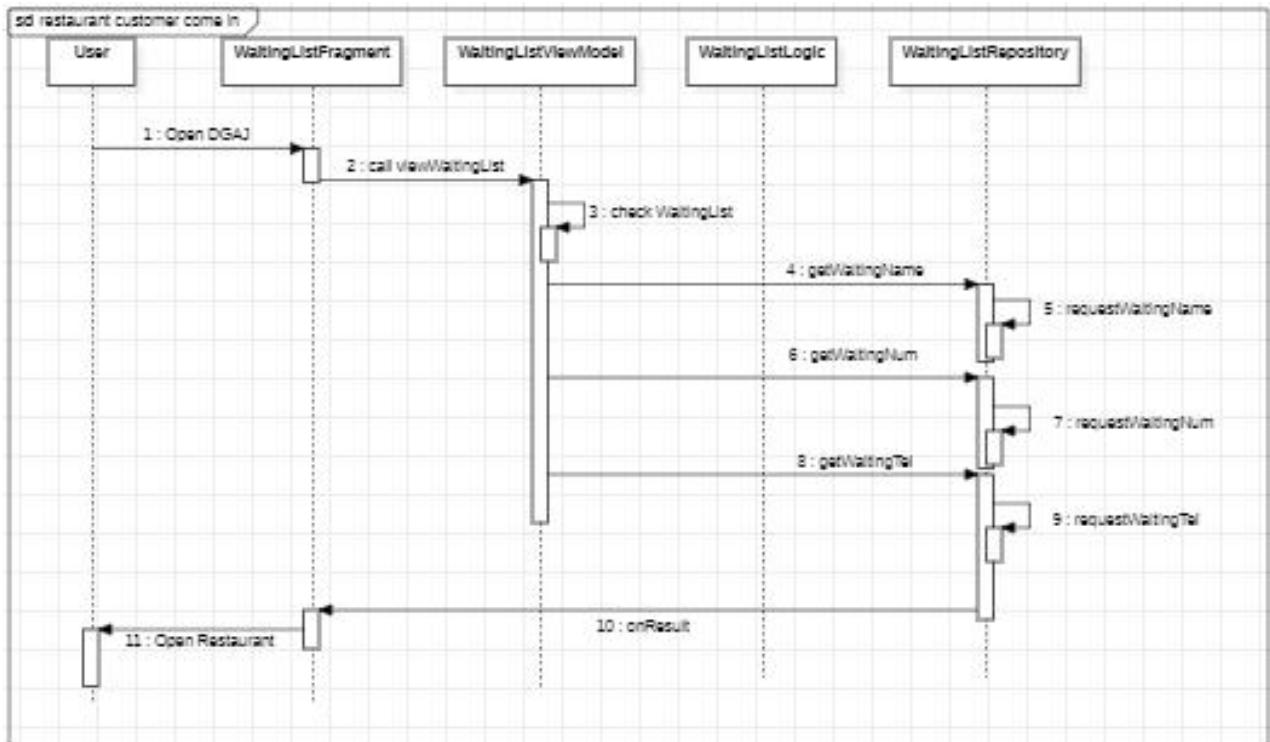


[그림 4-30] Restaurant Waiting List SD

사용자가 사업자의 계정으로 로그인했을 때 보여지는 화면이며, <Use Case #11-1>의 경우이다.

일반 고객이 자신의 식당에 대기 신청을 하였을 경우, 대기 신청 수만큼 리스트 뷰로 보 이게 된다. 한 항목당 대기 신청자의 이름, 대기 총인원 수, 신청자의 전화번호가 뜨게 되고, 오른쪽에는 입장 버튼과, 삭제 버튼이 있다.

Restaurant Customer Come in

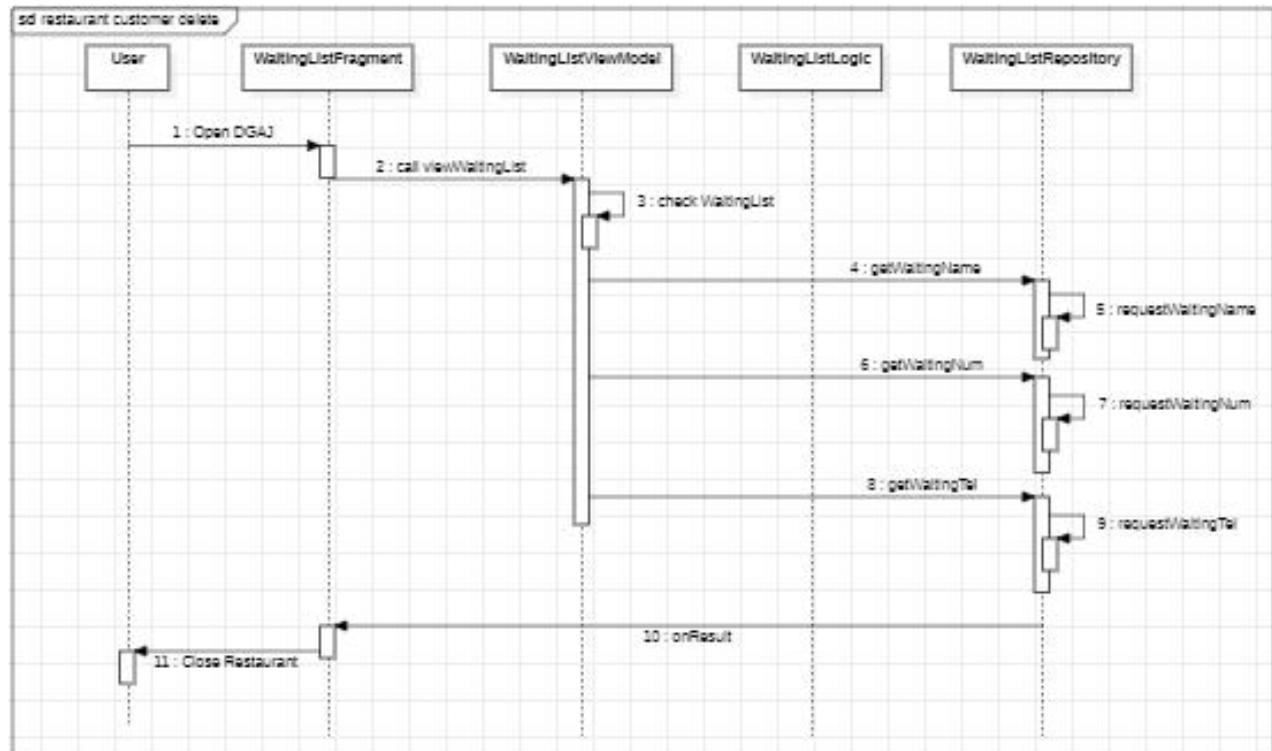


[그림 4-31] Restaurant Customer Come in SD

사용자가 사업자의 계정으로 로그인했을 때 보여지는 화면이며, <Use Case #11-6>의 경우이다.

자신의 식당에 대기 신청을 한 고객들의 정보가 뜨는 뷰의 입장 버튼을 누르면 실행되는 기능으로, 버튼을 누르면 Firestore의 데이터베이스에 등록된 대기 팀의 정보가 삭제된다.

Restaurant Customer Delete

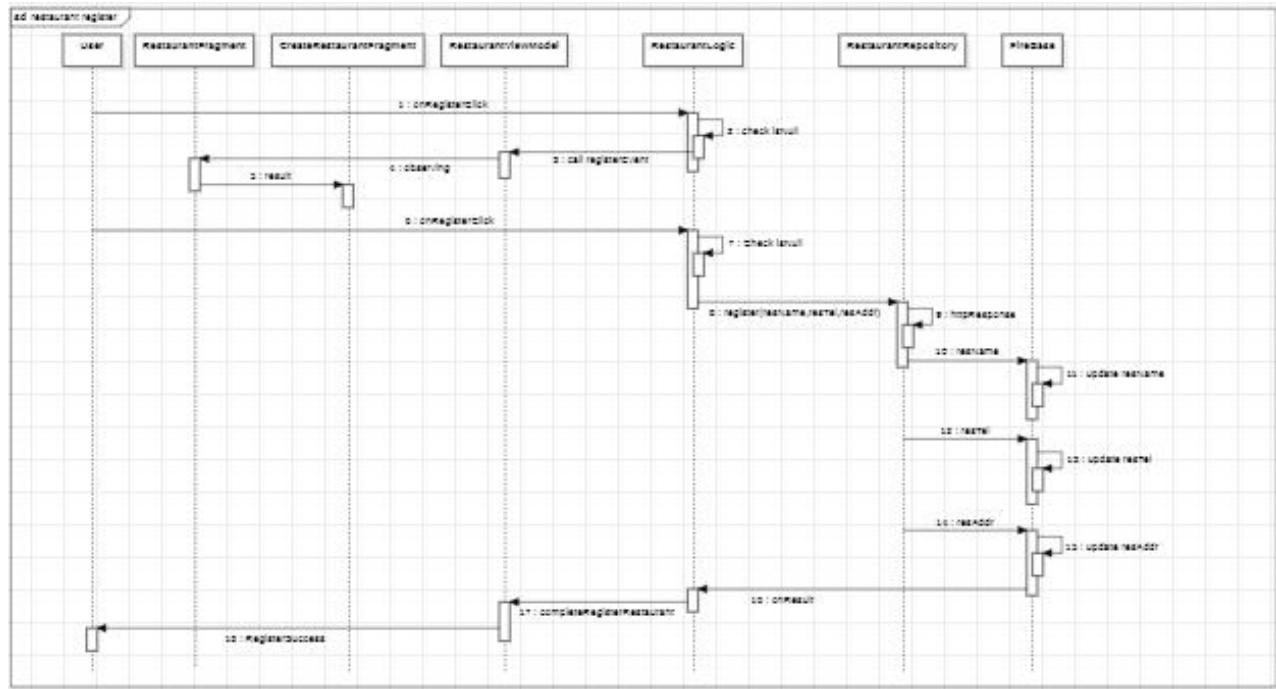


[그림 4-32] Restaurant Customer Delete SD

사용자가 사업자의 계정으로 로그인했을 때 보여지는 화면이며, <Use Case #11-8>의 경우이다.

자신의 식당에 대기 신청을 한 고객들의 정보가 뜨는 뷰의 삭제 버튼을 누르면 실행되는 기능으로, 버튼을 누르면 Firestore의 데이터 베이스에 등록된 대기 팀의 정보가 삭제된다.

Restaurant Register

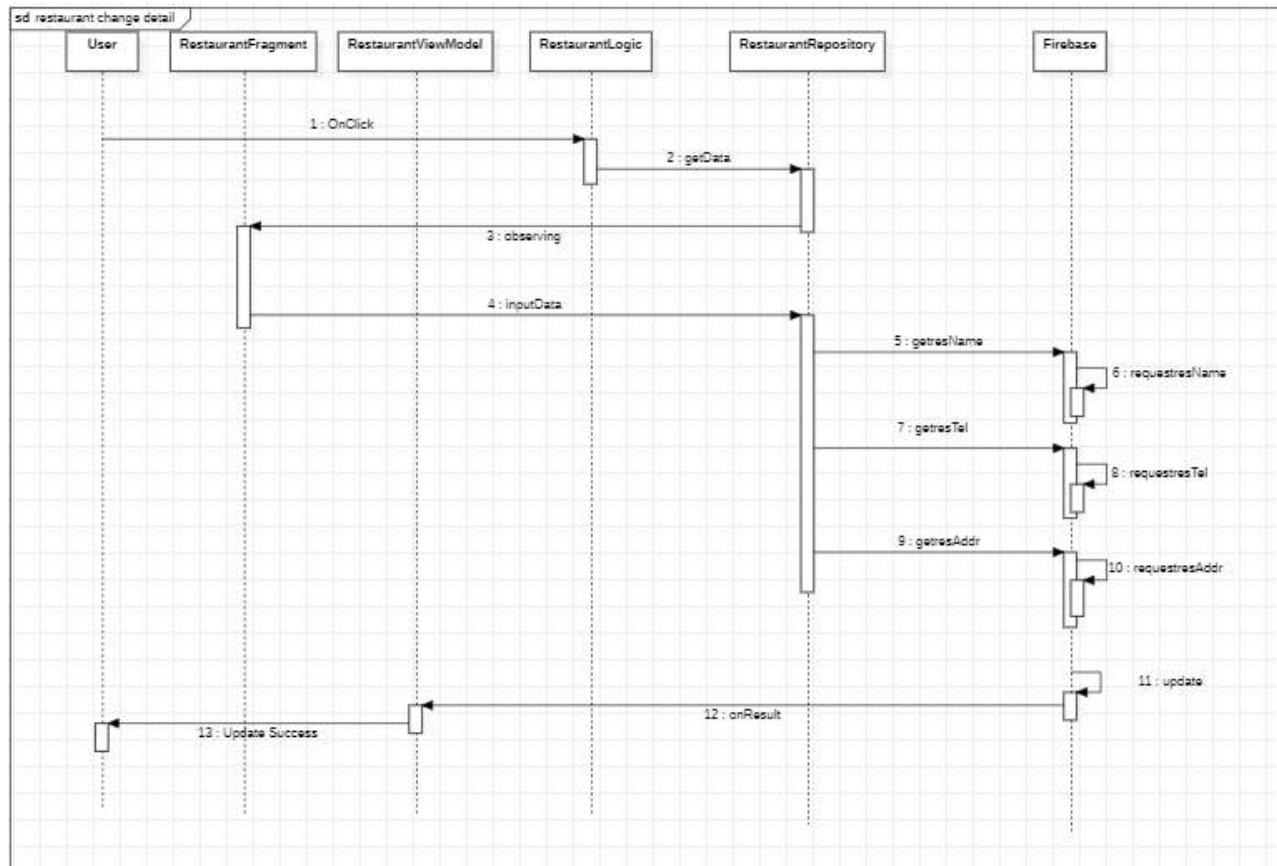


[그림 4-33] Restaurant Register SD

사용자가 사이드바에서 사업장 등록을 선택했을 때 보이는 화면이며, <Use Case #12>의 경우이다.

자신의 계정에 등록된 식당이 없는 경우, 이 기능을 통해 자신의 식당을 등록할 수 있다. 사업장 이름, 전화번호, 주소, 음식 카테고리, 가게 소개를 양식에 맞춰 작성하면 Fir eStore에 식당에 대한 정보가 등록된다.

Restuarant Change Detail

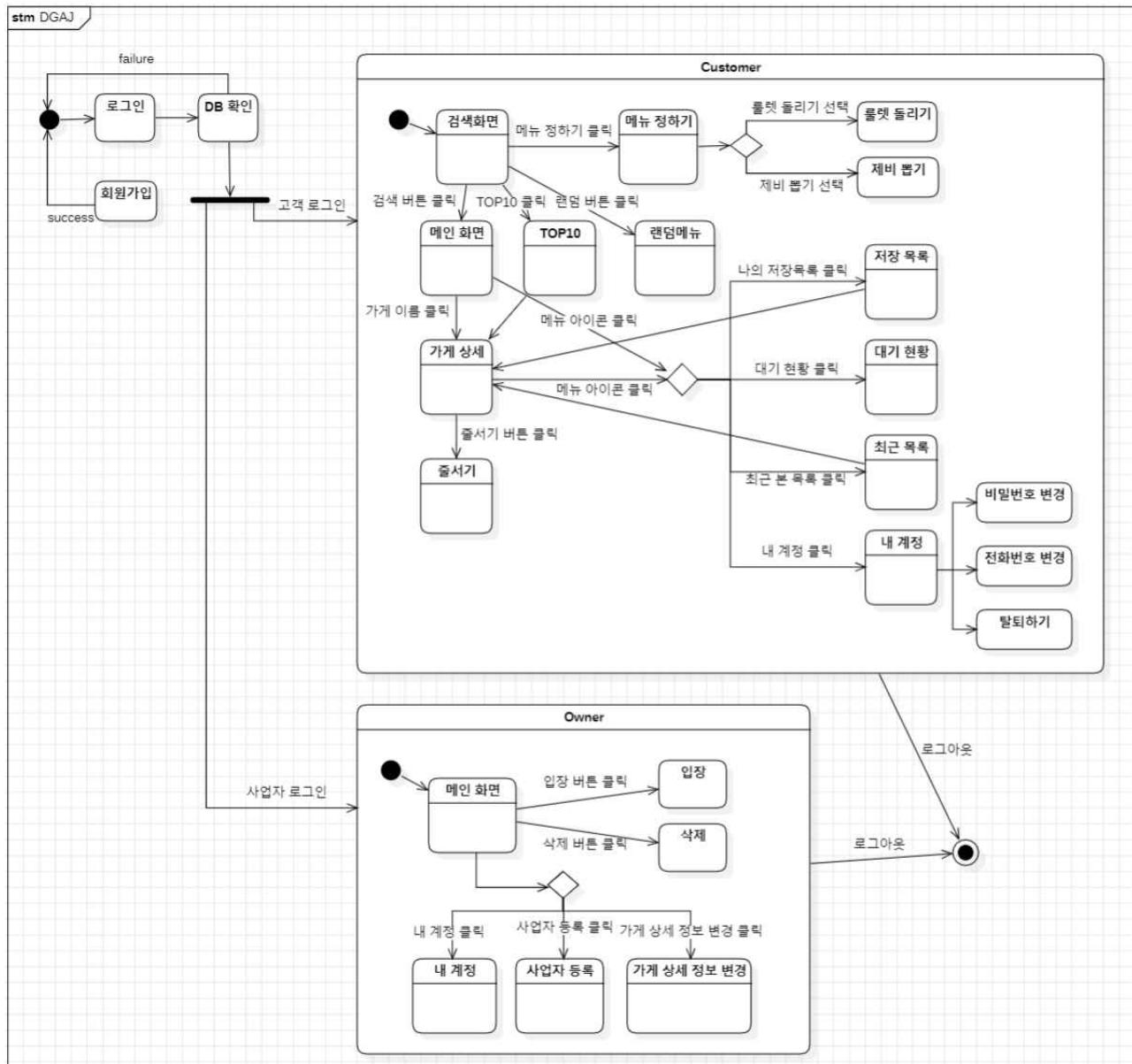


[그림 4-34] Restaurant Change Detail SD

사용자가 사이드바에서 가게 상세 정보 변경을 선택했을 때 보이는 화면이다.

자신이 등록한 식당의 정보가 변경되었을 경우, 사업장 등록과 마찬가지로 양식에 맞춰 새로 변경된 정보들을 작성하고 수정하기 버튼을 누르면 FireStore에 해당 수정사항이 반영되어 업데이트가 된다.

5. State machine diagram



5.1 State machine diagram description

앱 실행 시 로딩 화면이 시작되며 로딩 화면이 종료되면 Login 페이지가 나타난다. 이전의 가입 유무에 따라 로그인 기능과 회원가입 기능을 사용할 수 있고, 이전에 가입한 정보가 기억나지 않을 시 아이디/비밀번호 찾기 기능을 사용하여 해당 정보 조회가 가능하다. 회원가입의 경우 사용자가 입력한 정보가 설정된 형식에 적합할 경우 해당 사용자의 정보를 DB에 저장한다. 로그인과 아이디/비밀번호 찾기 기능은 DB에 저장된 사용자 정보를 바탕으로 적절한 절차를 거쳐 인증이 완료되면 정상적인 동작이 가능하다. 로그인에 성공할 경우, 로그인된 사용자의 타입(고객, 사업자)에 따라 각각 다른 초기 화면이 나타나게 된다.

고객 계정으로 로그인이 성공하면 현 위치와 메뉴를 입력하여 현재 위치를 기준으로 입력한 메뉴를 판매하는 주변 가게들의 검색이 가능한 검색 화면으로 전환된다. 검색 화면에서는 입력한 주소와 메뉴에 해당하는 가게 리스트가 지도와 함께 나타난다. 해당 페이지에서는 TOP10, 메뉴 정하기 등의 다른 기능들의 사용도 가능하다. TOP10은 시스템에서 설정한 검색기준을 바탕으로 상위 10개의 가게 리스트를 뽑아 추천해주는 기능이다. 메뉴 정하기 기능은 쉽게 메뉴를 결정하기 힘들거나 재미있게 메뉴 선정을 하고 싶은 사용자들을 위해 마련된 항목으로써, '룰렛 돌리기', '제비뽑기' 등의 게임을 이용하여 랜덤으로 메뉴 선정이 가능하다. 추가로, 좌측 상단의 메뉴 아이콘을 클릭하면 고객 메뉴창이 나타나게 되는데 해당 메뉴창을 통해 이전에 저장(찜)했던 가게 리스트를 확인할 수 있는 '나의 저장 목록', 본인이 신청한 웨이팅 상황을 확인할 수 있는 '대기 현황', 최근 조회했던 가게 리스트를 확인할 수 있는 '최근 본 목록', 개인정보 수정이 가능한 '내 계정'으로의 화면 이동이 가능하고 '로그아웃'을 클릭하여 로그아웃을 할 수 있다.

사업자 계정으로 로그인에 성공할 경우, 현재 본인의 가게의 고객들의 웨이팅 목록을 확인할 수 있다. 사업자 메뉴창에서는 '사업자 등록', '가게 상세 정보 변경', '내 계정'으로의 화면 이동이 가능하고 '로그아웃'을 클릭하여 로그아웃을 할 수 있다. 사업자 등록 및 가게 상세 정보 변경에서는 설정된 양식에 맞게 내용을 기입하여 적합할 경우 DB에 정보를 등록 및 업데이트한다.

6. User interface prototype



[그림 6-1] 로그인 화면

앱 실행이 성공하였을 때(로딩화면 종료된 후) 나타나게 되는 화면으로, 기존 가입 정보가 존재한다면 이메일과 비밀번호를 입력한 후 ‘로그인 버튼’을 클릭하여 로그인을 할 수 있다. 이전에 가입한 이력이 없는 경우 ‘회원가입 버튼’을 클릭하여 회원가입으로 이동할 수 있다.

회원가입

회원 사업자

아이디 Item 1

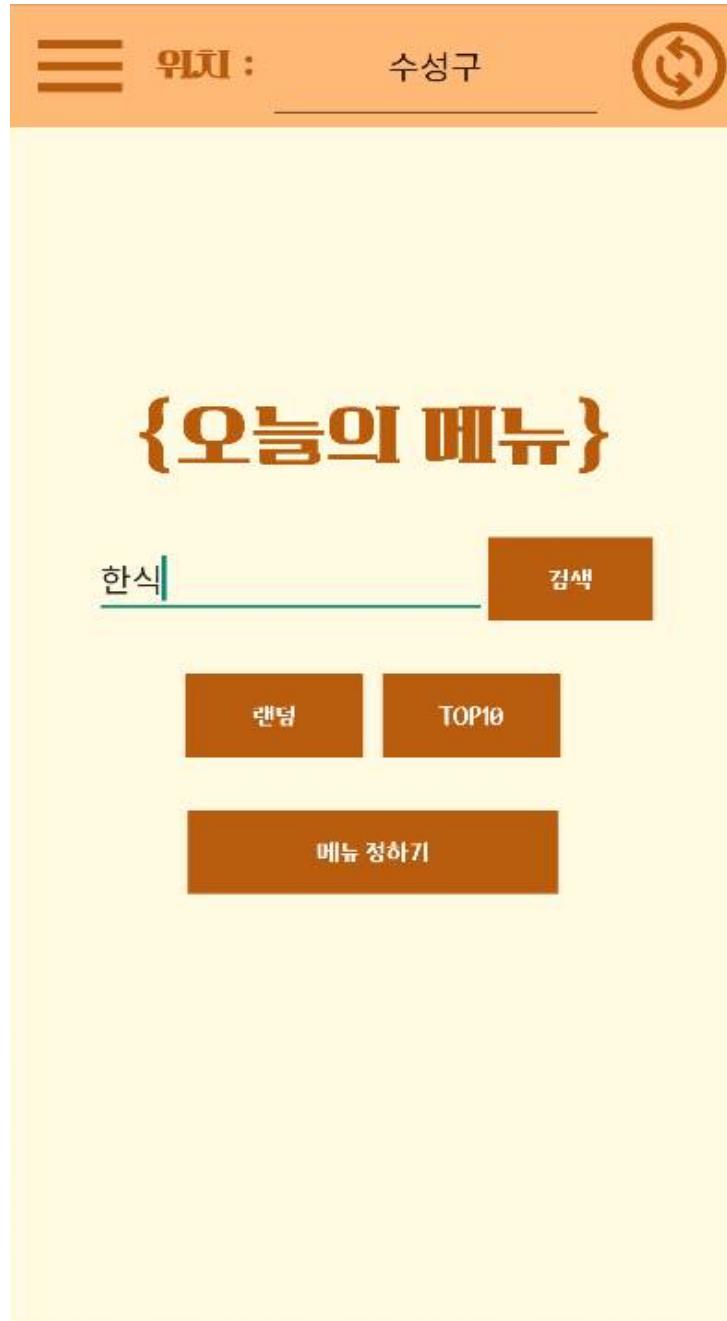
비밀번호

이름

전화번호 010 - -

[그림 6-2] 회원가입

사업자 사이드바에서 ‘가게 상세 정보 변경’ 항목을 클릭하면 위와 같이 ‘가게 상세 정보 수정하기’ 화면으로 전환된다. 사업자는 가게 정보와 사업자 설명을 형식에 따라 기입한 후, ‘수정하기’ 버튼을 눌러서 가게 상세 정보를 업데이트할 수 있다.



[그림 6-3] 검색 화면

고객 아이디로 로그인이 성공하면 위와 같이 메뉴를 검색할 수 있는 검색 화면으로 전환된다. 상단의 위치 입력창에 위치(ex. 수성구)를 입력하여 기준이 될 위치를 지정하고 화면 중단부의 메뉴 입력창에 메뉴(ex. 한식)를 입력한 후, ‘검색’ 버튼을 클릭하면 입력받은 입력한 위치를 기준으로 입력한 메뉴에 해당하는 가게 목록들이 나타나는 메인 화면으로 전환된다.



[그림 6-4] 사용자 메인 화면

검색 화면에서 현재 위치와 메뉴를 입력한 후 ‘검색’ 버튼을 누르면 위와 같은 고객 메인 화면으로 전환된다. 고객 메인 화면에는 고객이 입력한 위치를 기반으로 검색한 메뉴의 주변 가게들의 리스트가 지도와 함께 나타난다. 사용자는 스크롤 기능을 이용하여 많은 가게 정보들을 확인할 수 있고, 가게를 클릭하면 해당 가게의 상세 정보 페이지로의 이동도 가능하다.

가게 상세 페이지

이름: 곽옥자삼대곰탕
전화번호: 053-766-3356
주소: 대구광역시 수성구 두산동 70-1

상세 설명: 곽옥자삼대곰탕은 전통적인 전라도식 곰탕, 경상도식 곰탕, 그리고 맛있는 진주냉면까지 함께 즐길 수 있는 곰탕과 냉면 전문점입니다.

카테고리: 한식

대표메뉴: [저염메뉴] 곰탕 12,000원

 나주곰탕 11,000원
 양곰탕 16,000원
 모리곰탕 25,000원
 진주냉면 + 바싹불고기 16,000원

찜 개수: 0

[줄서기](#)

[가게 찜하기](#)

[그림 6-5] 가게 상세 페이지

고객 계정으로 로그인에 성공하여 메뉴 검색을 하였을 때, 목록에서 가게를 선택하면 위와 같이 해당 가게의 상세 정보를 나타내는 가게 상세 페이지로 화면이 전환된다. 사용자는 해당 페이지를 통해 가게의 상세 정보와 대표 메뉴 등을 확인할 수 있고, ‘줄서기’ 버튼과 ‘가게 찜하기’ 버튼을 활용하여 웨이팅 신청과 해당 가게를 저장 목록에 등록이 가능하다.

줄서기	줄서기
전화번호	전화번호
01012345678	01012345678
이름	이름
홍길동	홍길동
인원수	인원수
4	4
현재 대기 현황	•
확인	확인

▲ Question

현재 대기 중인 대기팀 : 3팀
대기 신청 하시겠습니까?

NO **YES**

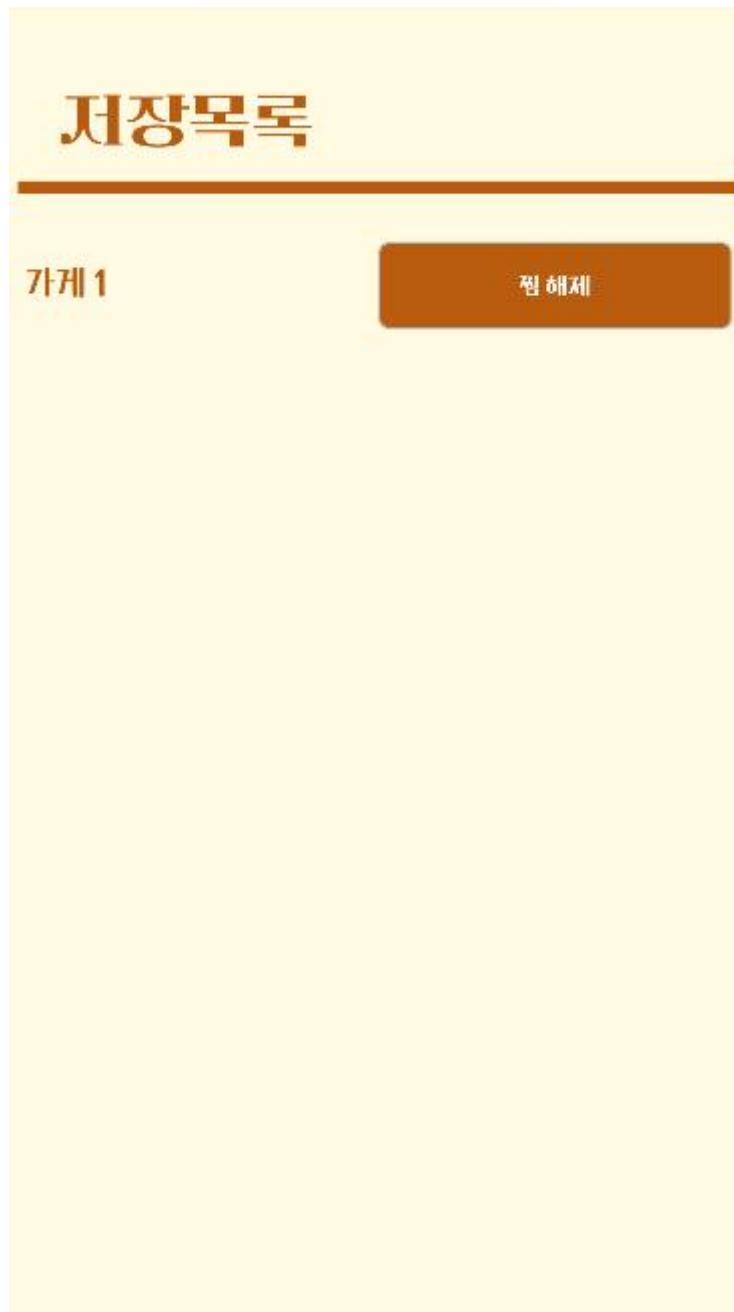
[그림 6-6] 줄서기 화면

가게 상세 정보 페이지에서 줄서기 버튼을 클릭하면 위와 같이 대기 신청 화면으로 전환된다. 전화번호, 이름, 인원수 등을 기입하여 확인 버튼을 누르면 팝업 창을 통해 실시간으로 현재 대기 현황 확인이 가능하고, ‘YES’버튼을 클릭하면 대기 신청이 완료된다.



[그림 6-7] 사용자 메뉴 화면

고객 계정으로 로그인에 성공하였을 때, 메뉴 아이콘이 있는 화면에서 메뉴 아이콘을 클릭하면 위의 사진과 같이 사용자 사이드바가 화면 좌측에서 나타나게 된다. 사용자는 사이드바를 통해 ‘나의 저장 목록’, ‘대기 현황’, ‘최근 본 목록’, ‘내 계정’으로의 화면 이동이 가능하고 ‘로그아웃’을 클릭하여 로그아웃을 할 수 있다.



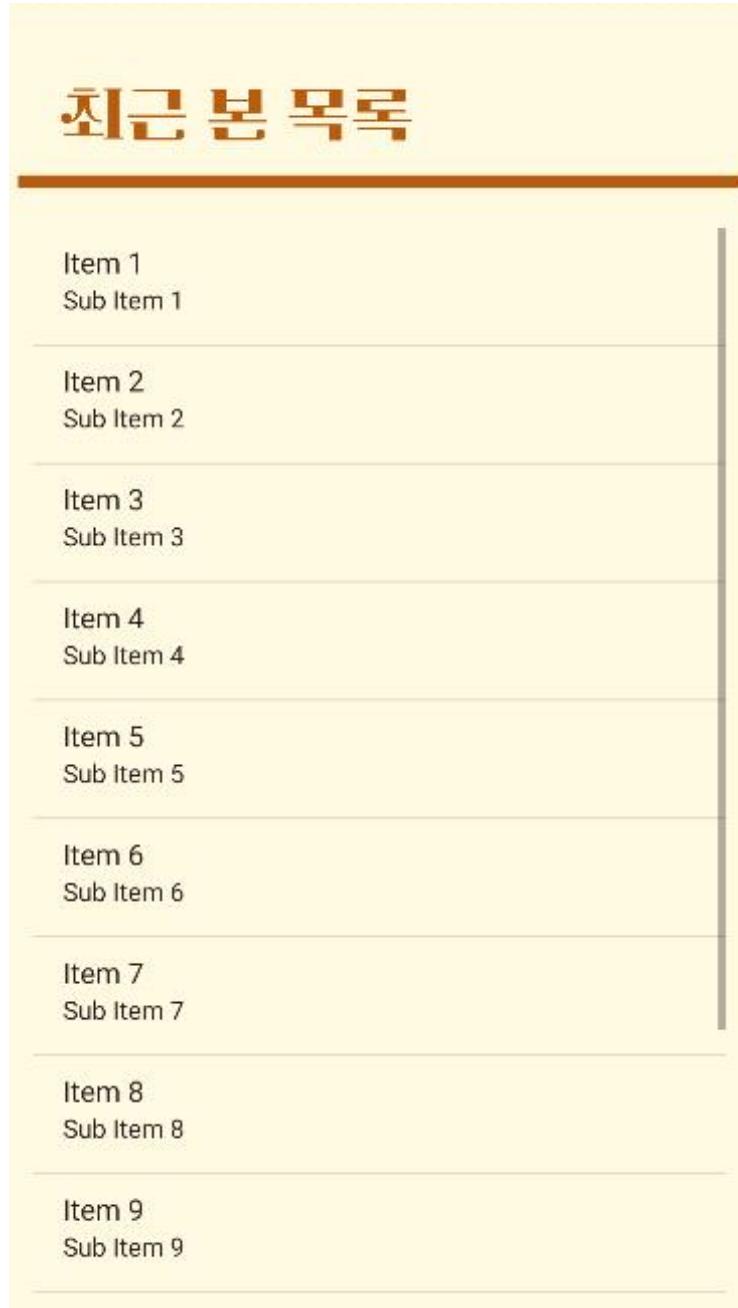
[그림 6-8] 나의 저장 목록

고객 사이드바에서 ‘나의 저장 목록’을 클릭하면 위와 같이 나의 저장 목록을 화면으로 전환된다. 사용자는 해당 페이지를 통하여 본인이 찜(저장)했던 가게들의 목록을 살펴볼 수 있고, 가게를 선택하여 해당 가게의 상세 정보 페이지로 이동하거나 해당 가게의 찜 해제를 할 수 있다.



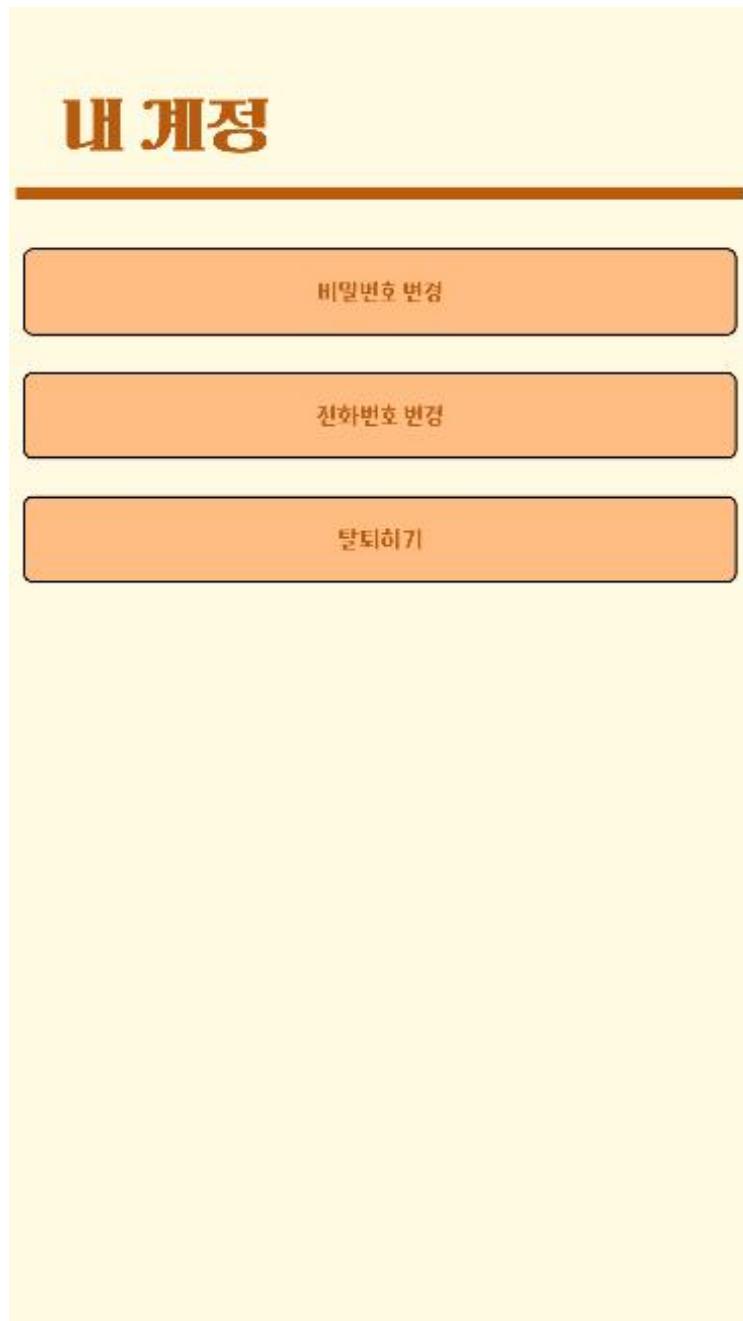
[그림 6-9] 대기 현황

고객 사이드바에서 ‘대기 현황’을 클릭하면 위와 같이 대기 현황 화면으로 전환된다. 사용자는 해당 페이지를 통하여 본인이 대기 신청을 한 가게의 이름과 현재 대기 현황 및 대기 순번을 확인할 수 있고, ‘대기 취소’ 버튼을 클릭하여 대기 취소도 가능하다.



[그림 6-10] 최근 본 가게 목록

고객 사이드바에서 ‘최근 본 목록’을 클릭하면 위와 같이 ‘최근 본 목록’으로 화면이 전환된다. 해당 페이지에서는 사용자가 최근에 상세 정보를 조회했던 가게들의 목록이 최신순으로 나타나게 되고, 사용자는 각 가게들을 클릭하여 상세 정보 페이지로 이동하거나 찜하기 등의 기능을 활용할 수 있다.



[그림 6-11] 정보 수정 화면

사이드바에서 ‘내 계정’을 클릭하면 위 그림과 같이 정보 수정 화면으로 넘어간다. 사용자는 비밀번호 변경, 전화번호 변경, 탈퇴하기 기능을 사용할 수 있다.



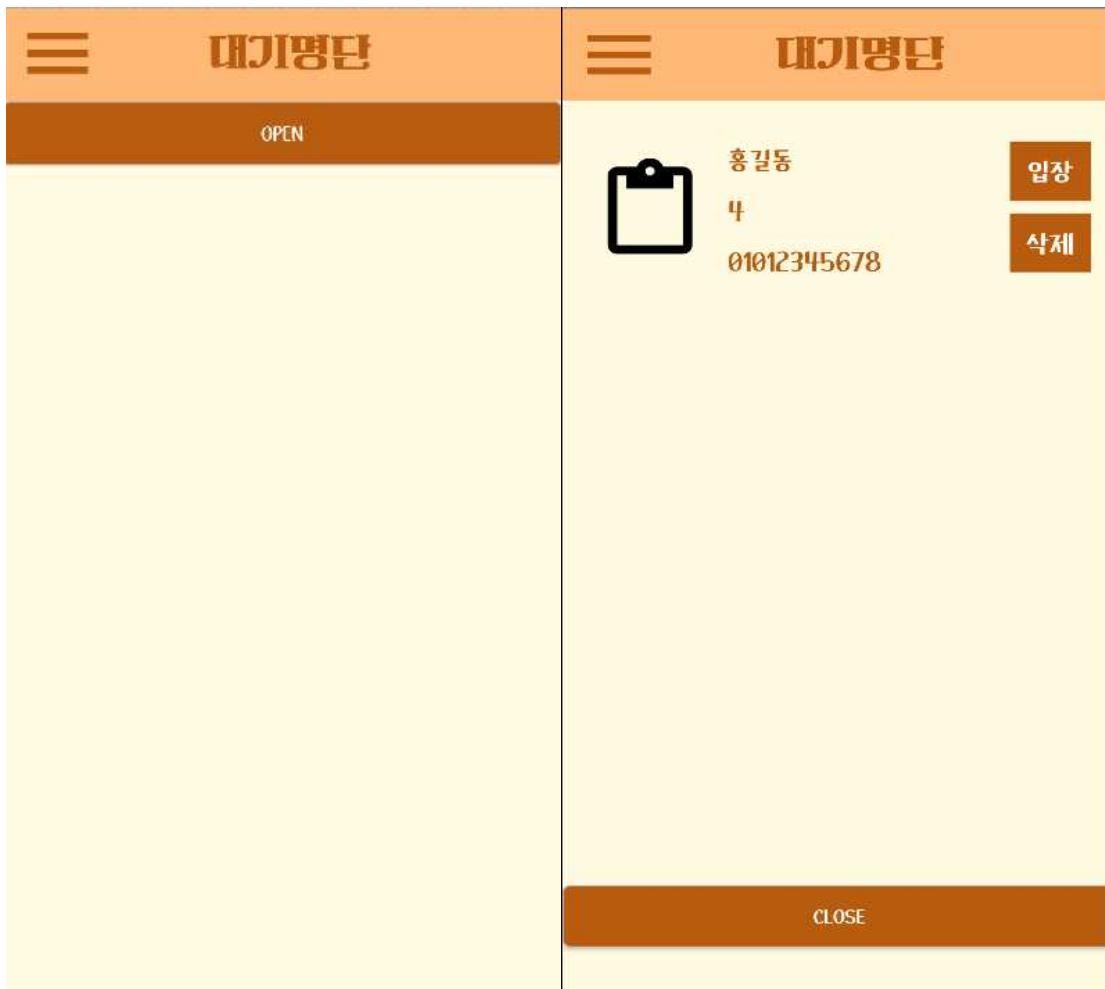
[그림 6-12] 비밀번호 수정 화면

사이드바에서 ‘내 계정’을 클릭하여 비밀번호 변경 버튼을 클릭하면 위와 같이 비밀번호 변경 화면으로 전환된다. 사용자는 기존 비밀번호를 입력하여 존재 여부를 확인한 후, 새로운 비밀번호를 입력하여 기존 비밀번호를 업데이트할 수 있다.



[그림 6-13] 전화번호 수정 화면

사이드바에서 ‘내 계정’을 클릭하여 전화번호 변경 버튼을 클릭하면 위와 같이 전화번호 변경 화면으로 전환된다. 사용자는 기존 전화번호를 입력하여 존재 여부를 확인한 후, 새로운 전화번호를 입력하여 기존 전화번호를 업데이트할 수 있다.



[그림 6-14] 대기 명단 확인 화면(사업자 계정 메인 화면)

사업자 계정으로 로그인에 성공할 경우 위와 같이 현재 해당 가게의 대기 명단이 메인 화면으로 노출되게 된다. 사업자는 ‘OPEN’ 버튼과 ‘CLOSE’ 버튼을 활용하여 대기 명단을 숨기거나 나타나게 할 수 있고, ‘입장’ 버튼과 ‘삭제’ 버튼을 통하여 실시간으로 대기 현황을 업데이트할 수 있다.



[그림 6-15] 사업자 메뉴 화면

사업자 계정으로 로그인에 성공하였을 때, 메뉴 아이콘이 있는 화면에서 메뉴 아이콘을 클릭하면 위의 사진과 같이 사업자 사이드바가 화면 좌측에서 나타나게 된다. 사용자는 사이드바를 통해 ‘사업자 등록’, ‘가게 상세 정보 변경’, ‘내 계정’으로의 화면 이동이 가능하고 ‘로그아웃’을 클릭하여 로그아웃을 할 수 있다.

사업자 등록하기

[가게 정보]

사업장 이름	사업장명을 등록하세요
전화번호	<input type="text"/> - <input type="text"/> - <input type="text"/>
주소	주소를 입력하세요

[사업장 설명]

카테고리	Item 1	▼
가게 소개	간단한 설명을 적어주세요.	

등록하기

[그림 6-16] 사업자 등록하기 화면

사업자 사이드바에서 ‘사업자 등록’ 항목을 클릭하면 위와 같이 사업자 등록하기 화면으로 전환이 된다. 사업자는 가게 정보와 사업자 설명을 형식에 따라 입력한 후, ‘등록하기’ 버튼을 눌러서 사업자 등록을 완료할 수 있다.

가게 상세정보 수정하기

[가게 정보]

사업장 이름

전화번호 - -

주소

[사업장 설명]

카테고리 Item 1 ▾

가게 소개

수정하기

[그림 6-17] 가게 상세 정보 수정하기

사업자 사이드바에서 ‘가게 상세 정보 변경’ 항목을 클릭하면 위와 같이 ‘가게 상세 정보 수정하기’ 화면으로 전환된다. 사업자는 가게 정보와 사업자 설명을 형식에 따라 입력한 후, ‘수정하기’ 버튼을 눌러서 가게 상세 정보를 업데이트할 수 있다.



[그림 6-18] 게임 선택 화면

검색 화면에서 ‘메뉴 정하기’ 버튼을 클릭하여 게임 선택 화면으로 이동할 수 있다. 사용자는 ‘룰렛 돌리기’ 또는 ‘제비뽑기’ 게임을 활용하여 재미있고 랜덤하게 메뉴선정이 가능하다.

메뉴 정하기

게임에 넣을 메뉴들을 입력해주세요 (6개)

마라탕

초밥

삼겹살

피자

냉면

쌀국수

다음



[그림 6-19] 룰렛 돌리기

게임 선택 화면에서 ‘룰렛 돌리기’ 버튼을 클릭하면 위와 같이 ‘룰렛 돌리기’ 게임 화면이 나타난다. 사용자가 6가지 항목들을 입력하여 ‘다음’ 버튼을 누르면 입력한 항목들로 이루어진 룰렛이 생성되고, ‘돌리기’ 버튼을 클릭하면 랜덤으로 하나의 메뉴가 팝업 화면으로 출력된다.



[그림 6-20] 제비뽑기 화면

게임 선택 화면에서 ‘제비뽑기’ 버튼을 누르면 제비뽑기 게임 화면으로 전환된다. 사용자는 제비뽑기를 하고 싶은 메뉴명이나 종류를 Input Menu에 입력하고 메뉴 넣기 버튼을 눌러 List에 항목들을 추가할 수 있고, ‘제비뽑기’ 버튼과 ‘초기화’ 버튼을 이용하여 게임 실행 및 List 초기화가 가능하다.

7. Implementation requirements

<H/W platform requirements>

Application

- Processor: Qualcomm Snapdragon 835 MSM8998+
- RAM: 4GB+
- Storage: < 200MB
- Network: connected to WAN over TCP/IP

Server

- Firebase의 경우 서버의 사양을 직접 제어하지 않음. 대신, Google Cloud Platform에서 관리되는 서버에서 호스팅 되는, Firebase의 다양한 서비스를 이용한다.

<S/W platform requirements>

Application

- Android Studio 2020.3.1+
- Android SDK Tools 23+
- JDK 1.8.0+
- JRE 1.8.0+
- App OS: Pixel 3a API 34
- Develop Environment OS: Windows 8/10
- Implementation Language: Java, xml

Server

- Firebase SDK: Firebase 서비스를 이용하기 위한 SDK를 사용.
- DBMS: Firebase Realtime Database 또는 Firestore
- Authentication: Firebase Authentication
- Storage: Firebase Cloud Storage

8. Glossary

Words	Account
Customer	앱을 이용하는 사용자(고객)
Owner	앱을 이용하는 사업자
WaitingAdapter	Waiting class가 ListView를 가져오기 위해 도움을 주는 class
RestaurantAdapter	RecyclerView에 식당 목록을 표시하기 위한 어댑터 클래스.
GetMyGeoPoint	현재 장치의 위치 정보를 가져오는 기능을 제공하는 클래스.
FirebaseID	FirebaseID 클래스는 Firebase Firestore에서 사용되는 컬렉션 및 문서 식별자를 관리하는 클래스.

9. References

- [1] Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2016.
- [2] Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, Pearson, 2021.
- (3) Shari Lawrence Pfleeger and Joanne M. Atlee, Software Engineering: Theory and Practice, 4th Edition, Pearson, 2009.
- [4] 파일베이스 주소
https://console.firebaseio.google.com/u/1/project/dgaj-2a484.firebaseio/data/~2Fid_list~2F3mmullSYvVU8Vd0CE810ZxBI1Tg2?hl=ko
- [5] Google Cloud Console
<https://console.cloud.google.com/apis/credentials?authuser=1&project=dgaj-2a484&supportedpurview=project>
- [6] 공공데이터 포털
<https://www.data.go.kr/index.do>