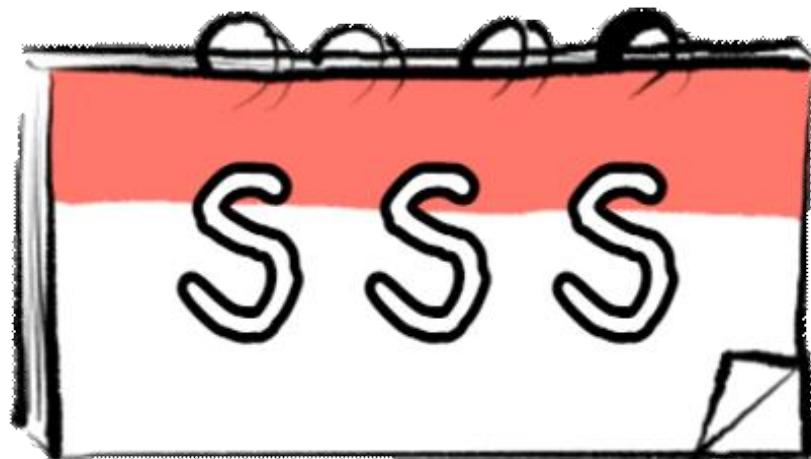


SSS(Saehoon's Scheduling System)

-System Design Specification-



[Revision history]

Revision date	Version #	Description	Author
	1.1	CD 누락 부분(DB 접근 다이어그램), Glossary 누락 부분 추가	XXX
	2.0	요구사항과 구현 디테일의 변경으로 인한 전반적인 수정 Use case description – 요구사항 번역 CD/SD/STD/UI – operations, attributes 변경/추가/삭제	XXX

= Contents =

1. Introduction	1
2. Use case analysis	2
3. Class diagram	21
3.1. DB Class diagram	21
3.2 상속 Class diagram	24
3.3 DB 접근 Class diagram	25
3.4 데이터 타입 Classes	28
3.5 Util and Static Classes	35
3.6 기능 Class diagram	39
4. Sequence diagram	76
4.1 User sequence diagram	77
4.2 Plan sequence diagram	78
4.3 Group sequence diagram	90
4.4 Friend sequence diagram	97
4.5 Profile sequence diagram	101
4.6 Recommend a place sequence diagram	106
4.7 Track real-time location sequence diagram	107
5. State machine diagram	109
6. User interface prototype	112
7. Implementation requirements	134
8. Glossary	135
9. References	136

= Authors by section =

Introduction - XXX, XXX

Use case diagram - XXX

Class diagram

- DB Class diagram - XXX
- 상속 Class diagram - XXX
- DB 접근 Class diagram - XXX
- Data type Class diagram - XXX
- Util and Static Classes - XXX
- User Class diagram - XXX
- Plan Class diagram - XXX
- Group Class diagram - XXX
- Friend Class diagram - XXX
- Profile Class diagram - XXX
- Recommend a place Class diagram - XXX
- Track Class diagram - XXX

Sequence diagram

- User Sequence diagram - XXX
- Plan Sequence diagram - XXX
- Group Sequence diagram - XXX
- Friend Sequence diagram - XXX
- Profile Sequence diagram - XXX
- Recommend a place Sequence diagram - XXX
- Track Sequence diagram - XXX

State machine diagram

- Application - XXX
- Server - XXX

User interface prototype - XXX, XXX

1. Introduction

본 문서는 우리 조가 개발하고자 하는 시스템인 Saehoon's Scheduling System(이하 SSS)의 design specification(SDS)이다. 문서에서는 SSS를 위해 기존에 식별한 기능적 요구사항을 구현하기 위해 시스템을 여러 가지 관점으로 바라보고 설계한다. Use case analysis는 사용자 관점에서 소프트웨어가 제공하는 기능을 서술했고 class diagram은 시스템의 구조적 관점, sequence diagram과 state machine diagram은 시스템의 동적 관점을 묘사했다. User interface는 사용자 인터페이스의 관점에서 시스템을 설계한다.

본 문서를 작성할 때 가장 중요하게 생각한 점은 diagram간의 일관성이다. 즉 다른 종류의 diagram간에 모순이 없는지 가장 주의를 들여 작성했다. 예를 들어, 특정 기능의 sequence diagram에서 쓰인 object, message는 모두 같은 기능의 class diagram에 있는 association, attribute, operation과 일치한다. 또한 sequence diagram의 흐름은 use case description의 main success scenario와 일치한다. 이와 같은 일관성은 각 장마다 다시 설명한다. 또한 같은 종류의 diagram내에서는 형식의 통일을 추구했다.

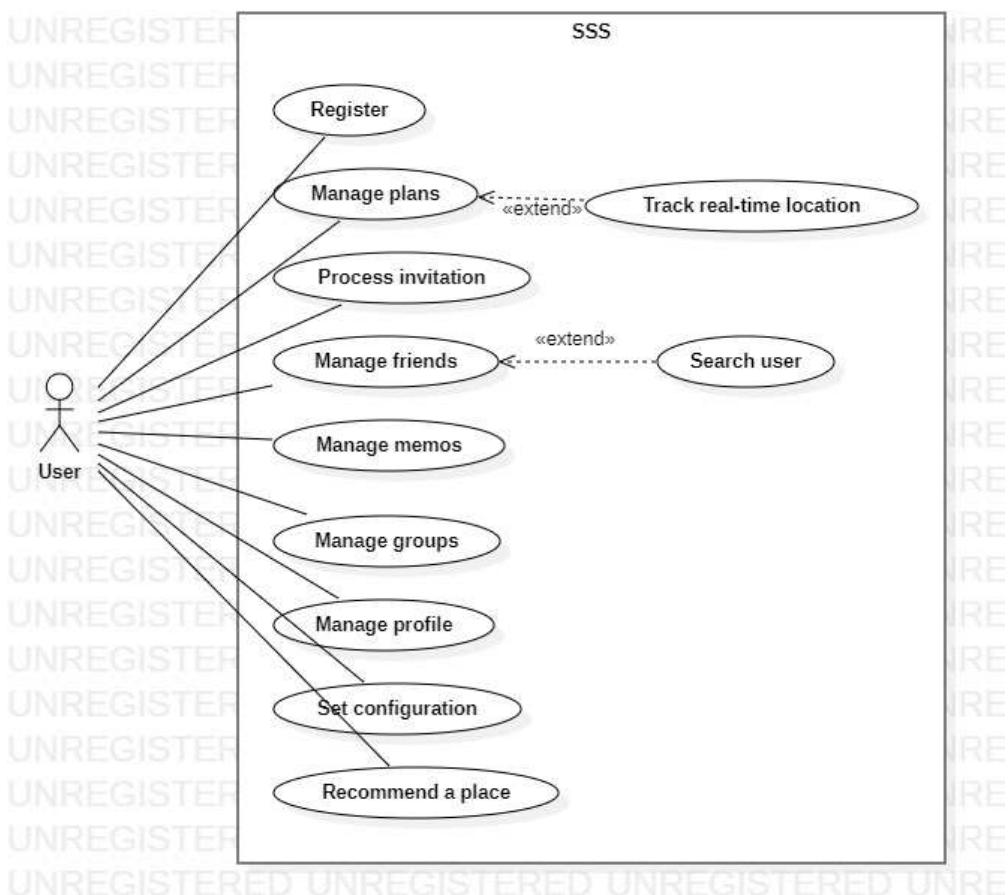
SSS는 서버와 안드로이드 어플리케이션으로 이뤄져있다. 안드로이드 어플리케이션의 구조는 MVVM을 채택했다. 본 문서는 클라이언트 측 소프트웨어인 안드로이드 어플리케이션의 설계를 중점적으로 다루며 서버에 대한 설계는 DB의 관점에서 보는 class diagram과 server state machine diagram에 간략히 설명되어 있다. 이 문서의 가장 중요한 부분은 use case description과 기능 class diagram이다. use case description은 후에 그려지는 거의 모든 UML diagram에 영향을 미칠 뿐 아니라 요구사항이 잘 반영되어 있는지 알 수 있으며 기능 class diagram은 구현 시에 가장 직접적인 영향을 미칠뿐만 아니라 시스템 구조를 파악할 수 있게 도와준다.

2. Use case analysis

이번 장은 use case diagram과 use case description을 제공한다. Diagram에 관한 고려사항은 아래와 같다.

- Use case diagram은 description에 대한 부가적인 결과물에 불과하다. 따라서 이 문서를 읽는 독자는 use case diagram보다 use case description에서 use cases에 대한 충분한 정보를 얻길 바란다.
- Use case diagram에 나타난 use case는 모두 user-goal level use cases이다.
- User는 로그인을 한 사용자이다. SSS는 기본적으로 로그인을 하지 않으면 기능을 이용할 수 없다. (회원가입 기능 제외)
- Manage로 시작하는 use cases는 여러 기능, 연산을 합친 것이며 use case description 고려사항에서 자세히 설명한다.

[그림 2-1]은 SSS의 use case diagram이다.



[그림 2-1] Use case diagram

이번 장의 남은 부분은 use case description에 할당한다. Use case description을 읽기 전 고려해야할 사항은 아래와 같다.

- 각 use case의 level은 summary, user-goal, subfunction로 구분한다.[1][2]
 - Summary는 전체 시스템의 문맥을 나타낸다.
 - 시나리오의 각 단계는 SSS의 핵심 user-goal use case이다.
 - extension scenario에서 오류(실패)와 관련된 부분은 다른 모든 use case에서도 동일하게 일어날 수 있다.
 - Use case #1
 - User-goal은 사용자가 지금 가장 원하는 기능이다.
 - 다른 level에 비해 가장 높은 중요도를 가진다.
 - description에서 user level로 표시한다.
 - 모든 user-goal use cases는 summary use case의 각 단계를 unfolding 한 것이거나 각 단계에서 추가적으로 제공하는 기능이다.
 - Use case #2~13
 - Subfunction은 user-goal use case의 시나리오 한 부분을 떼어내 만든 것이다.
 - 문서에서 모든 subfunction use cases를 고려하지 않지만 Use case #14, 15는 중요도와 필요성을 고려해 따로 use case로 만들어 관리한다.
 - Use case #14 Login: 시스템 사용을 위해 사용자에게 필수적으로 요구되며 기능에 대해 상대적으로 많은 요구사항(api 사용 등)이 있어 중요도가 높다고 판단함.
 - Use case #15 Invite user: 다양한 use case에서 공통된 흐름을 가지고 진행되는 기능이므로 중복을 제거하기 위해 사용함.
- Manage로 시작되는 모든 user-goal use cases는 각 분야(친구, 메모, 약속, 그룹, 프로필)에 대한 CRUD와 기본 연산들을 묶은 use case이다.
 - <<Writing effective use case>>에 따르면 Ivar Jacobson은 사람들이 너무 낮은 레벨의 use cases를 너무 많이 만든다고 한다. 또 아무리 복잡한 시스템이라도 use cases는 7개를 넘지 않는다 한다. 아마도 summary use case만을 고려한 결과일 것이다. 하지만 책의 저자는 시스템의 기능적 요구사항으로 use case를 보기 위해선 user-goal level use case가 더 중요하다고 한다.[3] 본 문서에서 우리 조는 이와 같은 전문가들의 의견들을 고려해 use case를 줄이되 summary level까지 줄이지 말고 기능적 요구사항을 충분히 식별할 수 있을 정도로 하여 총 15개의 use case를 추출했다.
 - CRUD와 기본 연산을 묶는 use case의 예시로 <<Writing effective use case>>를 참조했다.[4]

Use case #1 : Manage schedule	
GENERAL CHARACTERISTICS	
Summary	사용자는 시스템을 이용해 약속들을 관리한다. 이에 대해 SSS의 전반적이고 핵심적인 기능들을 위주로 시스템의 문맥을 나타낸다.
Scope	SSS (Sae hoon's scheduling system)
Level	Summary level
Author	XXX
Last Update	2022.11.30
Status	Analysis
Primary Actor	User
Preconditions	사용자는 시스템을 설치한 상태여야 한다. 사용자는 시스템에 회원가입을 해야만 한다. (Use case #2. Register) 사용자는 로그인해야만 한다. (Use case #14. Login)
Trigger	시스템을 이용해 약속들을 관리하려 할 때
Success Post Condition	사용자는 시스템을 이용해서 약속들을 효율적으로 관리할 수 있다.
Failed Post Condition	사용자는 시스템을 이용해서 약속들을 효율적으로 관리할 수 없다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자는 Use case #3. Manage plans를 이용해 약속들을 관리하려 한다.
1	사용자는 약속을 생성한다.
2	시스템은 약속 리스트, 각 약속에 대한 상세 정보 등을 제공한다.
3	사용자는 약속을 완료한다.
4	단계 1~4를 사용자가 원할 때까지 계속 반복한다.
EXTENSION SCENARIOS	
Step	Branching Action
1	1a. 친구 초대 …1a1. 사용자는 Use case #5. Manage friends를 사용해 친구를 초대한다.
1	1b. 그룹 초대 …1b1. 사용자는 Use case #7. Manage groups를 사용해 그룹에 친구를 초대한다.
3	3b. 메모 추가 …3b1. 사용자는 Use case #6. Manage memos를 사용해 메모를 추가/관리한다.
4	4a. 지난 약속 보기 …4a1. 사용자는 Use case #3-5a. Manage plans-지난 약속 알림을 통해 지난 약속을 본다.

4	4b. 지난 약속에 대한 통계 자료 보기 …4b1. 사용자는 Use case #8. Manage profile를 사용해 통계를 본다.
All	사용자의 요청과 데이터가 네트워크 문제로 시스템에 받아들여지지 않는다. …1. 사용자측의 시스템이 내부 서버에 요청을 3회 시도한다. …2. 사용자측의 시스템은 기능을 중단하고 사용자에게 에러 메시지를 제공한다.
RELATED INFORMATION	
Performance	약속 생성부터 완료까지에 대한 기간은 제한 없음.
Frequency	사용자당 1번
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #2 : Register	
GENERAL CHARACTERISTICS	
Summary	시스템의 이용자가 되기 위한 절차이며 모든 사용자는 사용에 앞서 회원가입을 해야 한다.
Scope	SSS (Sae hoon's scheduling system)
Level	User level
Author	XXX
Last Update	2022.11.30
Status	Analysis
Primary Actor	User
Preconditions	사용자는 시스템을 설치한 상태여야 한다.
Trigger	첫 화면에서 회원가입 버튼을 누를 때
Success Post Condition	사용자는 로그인을 진행할 수 있다.
Failed Post Condition	사용자는 로그인을 진행할 수 없다. 사용자는 시스템을 이용할 수 없다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 회원가입을 한다.
1	사용자는 처음 화면에서 회원가입 버튼을 누르고 회원가입 종류를 선택한다.
2	시스템은 회원가입하려는 계정의 종류에 맞게 회원가입 화면을 출력한다.
3	사용자는 닉네임과 ID, 비밀번호, 비밀번호 확인을 입력하고 회원가입 버튼을 누른다.
4	시스템은 회원가입이 성공한지 판단한다.
5	이 use case는 회원가입이 성공하면 끝난다.
EXTENSION SCENARIOS	
Step	Branching Action
1	1a. 소셜 회원가입 <ul style="list-style-type: none"> …1a1. 사용자는 구글 또는 네이버 회원가입 버튼을 눌러 계정을 연동한다. …1a2. 시스템은 회원가입 화면에 닉네임과 ID 입력 칸만을 표시한다. …1a3. 사용자는 닉네임과 ID를 입력하고 회원가입을 한다. (Use Case #2-3)
3	3a. 입력 오류로 회원가입에 실패한다. <ul style="list-style-type: none"> …3a1. 회원가입 실패 메시지를 보여준다. …3a2. 회원정보 입력 단계로 돌아간다. (Use case #2-4)
RELATED INFORMATION	
Performance	≤ 1 second
Frequency	사용자당 1번
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #3 : Manage plans
GENERAL CHARACTERISTICS

Summary	이 use case는 약속과 관련된 기본 기능들인 생성, 읽기, 갱신, 삭제, 초대, 강퇴, 취소, 약속 공개 여부 설정을 설명한다. Use case #1. Manage schedule에 의해 호출된다.
Scope	SSS (Sae hoon's scheduling system)
Level	User level
Author	XXX
Last Update	2022.11.30
Status	Analysis
Primary Actor	User
Preconditions	약속 생성을 제외한 기능들에 대해 약속이 최소 1개 이상 있어야 한다.
Trigger	사용자가 인터페이스에서 약속과 관련된 원하는 기능을 하려할 때
Success Post Condition	약속이 생성된다. 약속이 화면에 출력된다. 약속이 수정된다. 혹은 약속이 완료된다.
Failed Post Condition	약속에 대한 변경사항이 반영되지 않는다.

MAIN SUCCESS SCENARIO

Step	Action
S	사용자가 약속과 관련된 기능들을 수행하려 한다.
1	사용자는 약속 추가 버튼을 누른다.
2	시스템은 약속에 관련된 정보를 입력할 수 있는 창을 제공한다.
3	사용자는 약속 상세 정보(이름, 날짜, 장소, 카테고리, 그룹)를 기입하고 저장 버튼을 누른다.
4	시스템은 제공된 정보를 기반으로 약속을 생성한다.
5	사용자는 약속 목록에서 약속 상세 정보 버튼을 누른다.
6	시스템은 약속 상세 정보를 보여주는 창을 제공한다.
7	사용자는 약속 완료 버튼을 직접 눌러서 약속을 완료한다.
8	시스템은 완료된 약속을 지난 약속으로 바꿔 저장하고 사용자에게 100 포인트를 지급한다.
9	단계 5 이후로 사용자는 약속과 관련된 정보를 어느 때나 수정할 수 있다.

EXTENSION SCENARIOS

Step	Branching Action
1	1a. 타이핑을 통한 약속 생성(Text to schedule) ...1a1. 사용자는 약속 목록 화면의 세부 메뉴에서 타이핑으로 약속 추가 버튼을 누른다. ...1a2. 시스템은 문장을 입력할 수 있는 입력 폼을 제공한다. ...1a3. 사용자는 약속에 대한 정보를 양식에 맞게 타이핑하고 완료 버튼을 누른다. ...1a4. 시스템은 문자열을 파싱해 정보를 얻은 후 약속을 생성한다. ...1a5. 단계 4로 돌아간다.
1	1b. 손글씨를 통한 약속 생성(OCR) ...1b1. 사용자는 약속 목록 화면의 세부 메뉴에서 사진으로 약속 추가 버튼을 누른다. ...1b2. 사용자는 약속에 대한 정보를 적은 글에 대한 사진을 찍은 후 업로드 버튼을 누른다.

	<ul style="list-style-type: none"> …1b3. 시스템은 업로드한 사진을 보여준다. …1b4. 사용자는 사진을 확인하고 완료 버튼을 누른다. …1b5. 시스템은 OCR을 이용해 사진을 문자열로 변환한 후 문자열을 파싱해 정보를 얻은 후 약속을 생성한다. …1b6. 단계 4로 돌아간다.
5	<p>5a. 지난 약속 보기</p> <ul style="list-style-type: none"> …5a1. 사용자는 필터를 지난 약속으로 바꾼다. …5a2. 시스템은 지난 약속을 출력한다.
5	<p>5b. 약속 공개 여부 설정</p> <ul style="list-style-type: none"> …5b1. 사용자(약속 주최자)는 약속 목록에서 각 약속에 대한 공개 여부를 설정한다. …5b2. 시스템은 변경된 사항을 저장한다. …5b3. 시스템은 다른 사용자가 사용자의 프로필을 볼 때 공개된 약속을 출력한다.
6	<p>6a. 약속 삭제</p> <ul style="list-style-type: none"> …6a1. 사용자는 약속 상세 정보 화면의 세부메뉴에서 삭제 버튼을 누른다. …6a2. 시스템은 ‘약속을 삭제하시겠습니까?’라는 메시지를 출력한다. …6a3. 사용자는 확인 버튼을 누른다. …6a4. 시스템은 약속을 삭제한다.
6	<p>6b. 약속 초대</p> <ul style="list-style-type: none"> …6b1. 사용자는 약속 상세 정보 화면의 세부메뉴에서 초대 버튼을 누른다. …6b2. 사용자는 Use case #15. Invite user를 이용해 친구를 초대한다.
6	<p>6c. 약속 강제 퇴장</p> <ul style="list-style-type: none"> …6c1. 사용자(약속 주최자)는 약속 상세 정보 화면의 세부메뉴에서 강제 퇴장 버튼을 누른다. …6c2. 시스템은 약속 구성원 목록을 출력한다. …6c3. 사용자(약속 주최자)는 약속 구성원 목록에서 퇴장시킬 사용자들을 선택하고 강제 퇴장 버튼을 누른다. …6c4. 시스템은 대상 사용자들을 퇴장시킨다.
6	<p>6d. 약속 강제 퇴장 취소</p> <ul style="list-style-type: none"> …6d1. 사용자는 약속 강제 퇴장 화면에서 취소 버튼을 눌러 강제 퇴장을 취소한다. …6d2. 시스템은 약속 강제 퇴장 화면을 닫는다.
6	<p>6e. 약속 참석 취소</p> <ul style="list-style-type: none"> …6e1. 사용자는 구성원의 수가 2명 이상인 약속의 약속 상세 정보 화면의 세부메뉴에서 약속 참석 취소 버튼을 누른다. …6e2. 시스템은 ‘참석을 취소하시겠습니까?’라는 메시지를 출력한다. …6e3. 사용자는 확인 버튼을 누른다. …6e4. 시스템은 사용자의 약속 참석을 취소한다.
7	<p>7a. 자동 약속 완료</p> <ul style="list-style-type: none"> …7a1. 사용자는 약속 수행 후 종료 시간이 지난 약속을 그대로 둔다. …7a2. 시스템은 종료 시간이 지난 약속을 자동으로 완료로 바꾼다. …7a3. 단계 8로 돌아간다.

RELATED INFORMATION

Performance	≤ 1 second
Frequency	사용자당 1번
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #4 : Process invitation
GENERAL CHARACTERISTICS

Summary	다른 사용자의 초대를 수락하여 약속 혹은 그룹을 공유한다. Use case #15. Invite user로부터 호출된다.
Scope	SSS (Sae hoon's scheduling system)
Level	User level
Author	XXX
Last Update	2022.11.30
Status	Analysis
Primary Actor	User
Preconditions	사용자는 현재 초대받은 약속 혹은 그룹이 1개 이상이어야 하며, 약속 수락 화면이 열린 상태여야 한다. 사용자는 초대 푸시 알람 설정을 사용으로 해야 한다.
Trigger	사용자가 초대 푸시 알림을 누르거나 앱을 실행 중 상태일 때
Success Post Condition	사용자는 초대받은 약속 혹은 그룹을 수락 또는 거절한다.
Failed Post Condition	사용자는 초대받은 약속 혹은 그룹을 수락 또는 거절하지 못한다.

MAIN SUCCESS SCENARIO

Step	Action
S	사용자가 초대받은 약속을 수락 또는 거절한다.
1	사용자가 앱을 실행 중인 상태이거나 푸시 알림을 누른다.
2	시스템이 약속 초대를 수락/거절할 수 있는 알림 화면을 제공한다.
3	사용자는 수락 또는 거절 버튼을 누른다.
4	시스템은 변경된 내용을 서버에 저장하고, 완료 메시지를 화면에 출력한다.

EXTENSION SCENARIOS

Step	Branching Action
4	4a. 약속 초대의 경우 초대받은 약속의 약속 시간과 중복되는 기존의 약속이 존재하면 약속 초대에 실패한다. …4a1. 시스템은 오류 메시지를 출력한다.

RELATED INFORMATION

Performance	≤ 2 seconds
-------------	-------------

Frequency	사용자당 하루에 평균 5번 미만
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #5 : Manage friends	
GENERAL CHARACTERISTICS	
Summary	이 use case는 친구와 관련된 기본 기능들인 친구 추가, 목록 보기, 삭제, 차단을 설명한다. Use case #1. Manage schedule에 의해 호출된다.
Scope	SSS (Sae hoon's scheduling system)
Level	User level
Author	XXX
Last Update	2022.11.30
Status	Analysis
Primary Actor	User
Preconditions	친구 추가와 사용자 검색을 제외한 기능들에 대해 이미 친구인 사용자가 최소 1명 이상은 있어야 한다. 친구 추가 기능을 사용하려면 Use case #10. Search user를 사용해 원하는 사용자를 찾은 상태여야 한다.
Trigger	사용자가 인터페이스에서 친구와 관련된 원하는 기능을 하려할 때
Success Post Condition	다른 사용자를 친구로 추가한다. 친구 목록을 본다. 친구인 사용자를 약속에 초대할 수 있다. 친구인 사용자를 친구 목록에서 삭제하고 차단한다.
Failed Post Condition	친구에 대한 변경사항이 반영되지 않는다.

MAIN SUCCESS SCENARIO

Step	Action
S	사용자가 친구와 관련된 기능들을 수행하려 한다.
1	사용자는 검색된 사용자에 대한 친구 추가 버튼을 누른다.
2	시스템은 사용자의 친구 목록에 대상 사용자를 친구로 추가한다.
3	사용자는 친구 목록 보기 버튼을 누른다.
4	시스템은 친구 목록과 각각에 대한 접속 및 약속 진행 여부를 출력한다.

EXTENSION SCENARIOS

Step	Branching Action
4	4a. 친구 차단 ...4a1. 사용자는 친구 목록에서 차단 버튼을 누른다. ...4a2. 시스템은 대상 사용자를 사용자의 친구 목록에서 삭제하고 대상 사용자가 사용자를 향해 한 초대 등을 차단시킨다.
4	4b. 친구 삭제

	...4b1. 사용자는 친구 목록에서 삭제 버튼을 누른다. ...4b2. 시스템은 대상 사용자를 사용자의 친구 목록에서 삭제한다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사용자당 하루 평균 2번
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #6 : Manage memos	
GENERAL CHARACTERISTICS	
Summary	이 use case는 메모와 관련된 기본 기능들인 메모 추가, 보기, 수정(삭제 후 추가), 삭제를 설명한다. Use case #1. Manage schedule에 의해 호출된다.
Scope	SSS (Sae hoon's scheduling system)
Level	User level
Author	XXX
Last Update	2022.11.04
Status	Analysis
Primary Actor	User
Preconditions	약속 상세 정보 화면이 열린 상태여야 한다.
Trigger	사용자가 인터페이스에서 메모와 관련된 원하는 기능을 하려할 때
Success Post Condition	메모를 추가한다. 메모를 수정한다. 메모를 해당 약속의 다른 구성원들이 볼 수 있다.
Failed Post Condition	메모에 대한 변경사항이 반영되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 메모와 관련된 기능들을 수행하려 한다.
1	사용자는 약속 상세 정보 화면에서 메모 등록 버튼을 누른다.
2	시스템은 메모를 적을 수 있는 입력 품을 제공한다.
3	사용자는 메모를 작성하고 입력 버튼을 누른다.
4	시스템은 메모를 저장하고 화면에 출력한다.
5	단계 3이후로 사용자는 어느 때나 메모를 삭제하고 다시 등록할 수 있다.
EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 작성한 메모가 100자를 초과하는 경우 약속 메모 등록에 실패한다.

	…3a1. 글자 수 초과 알림 메시지를 출력한다. …3a2. 메모 작성 단계로 돌아간다.
3	3b. 메모 등록 취소 …3b1. 사용자는 메모 등록 화면에서 취소 버튼을 눌러 메모 등록을 취소한다. …3b2. 시스템은 메모 등록 화면을 닫는다.

RELATED INFORMATION

Performance	≤ 1 second
Frequency	사용자당 하루 평균 1번
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #7 : Manage groups
GENERAL CHARACTERISTICS

Summary	이 use case는 그룹과 관련된 기본 기능들인 그룹 생성, 수정, 삭제, 목록 보기, 초대, 강제 퇴장, 탈퇴에 대해 설명한다. Use case #1. Manage schedule에 의해 호출된다.
Scope	SSS (Sae hoon's scheduling system)
Level	User level
Author	XXX
Last Update	2022.11.30
Status	Analysis
Primary Actor	User
Preconditions	그룹 생성을 제외한 기능들에 대해 사용자가 속해있는 그룹이 최소 1개 이상 있어야 한다.
Trigger	사용자가 인터페이스에서 그룹과 관련된 원하는 기능을 하려할 때
Success Post Condition	그룹이 추가된다. 그룹에 사용자가 추가된다.
Failed Post Condition	그룹에 대한 변경사항이 반영되지 않는다.

MAIN SUCCESS SCENARIO

Step	Action
S	사용자가 그룹과 관련된 기능들을 수행하려 한다.
1	사용자는 그룹 생성 버튼을 누른다.
2	시스템은 그룹과 관련된 정보를 입력할 수 있는 창을 제공한다.
3	사용자는 그룹명, 그룹 설명, 그룹 색깔을 선택하고 저장 버튼을 누른다.
4	시스템은 그룹을 생성한다.
5	사용자는 Use case #15. Invite user를 사용해 친구들을 그룹에 초대한다.
6	사용자는 그룹 상세 정보 버튼을 누른다.
7	시스템은 그룹명, 그룹 색깔, 그룹 인원 수, 그룹 설명에 대한 상세 정보를 보여주는 창을 제

	공한다.
8	단계 6이후로 사용자는 그룹과 관련된 정보를 어느 때나 수정할 수 있다.
EXTENSION SCENARIOS	
Step	Branching Action
3	<p>3a. 그룹명이 20자를 초과하는 경우 그룹 생성에 실패한다. ...3a1. 시스템은 그룹 생성 실패 알림 메시지를 출력한다. ...3a2. 그룹 정보 작성 단계로 돌아간다.</p> <p>3b. 그룹 설명이 30자를 초과하는 경우 그룹 생성에 실패한다. ..3b1. 시스템은 그룹 생성 실패 알림 메시지를 출력한다. ..3b2. 그룹 정보 작성 단계로 돌아간다.</p>
4	<p>4a. 그룹 삭제 …4a1. 사용자(그룹 주인)는 그룹 상세 정보 화면의 세부메뉴에서 그룹 삭제 버튼을 누른다. …4a2. 시스템은 ‘그룹을 삭제하시겠습니까?’ 메시지를 출력한다. …4a3. 사용자는 확인 버튼을 누른다. …4a4. 시스템은 그룹을 삭제한다.</p>
4	<p>4b. 그룹 탈퇴 …4b1. 사용자는 그룹 상세 정보 화면의 세부메뉴에서 그룹 탈퇴 버튼을 누른다. …4b2. 시스템은 사용자를 그룹으로부터 탈퇴시킨다.</p>
5	<p>5a. 그룹 초대 ..5a1. 사용자(그룹 주인)은 그룹 상세 정보 화면의 세부메뉴에서 친구 초대 버튼을 누른다. ..5a2. 시스템은 친구 목록을 온라인, 이름 순서의 정렬 기준으로 출력한다. ..5a3. 시스템은 초대할 사용자를 선택 후 확인 버튼을 누른다. ..5a4. 시스템은 해당 사용자를 그룹에 초대하고 푸시 알림을 전송한다.</p>
5	<p>5b. 그룹 강제 퇴장 …5b1. 사용자(그룹 주인)은 약속 상세 정보 화면의 세부메뉴에서 강제 퇴장 버튼을 누른다. …5b2. 시스템은 그룹 구성원 목록을 출력한다. …5b3. 사용자(그룹 주인)은 그룹 구성원 목록에서 퇴장시킬 사용자들을 선택하고 강제 퇴장 버튼을 누른다. …5b4. 시스템은 대상 사용자들을 퇴장시킨다.</p>
5	<p>5c. 그룹 강제 퇴장 취소 …5c1. 사용자는 그룹 강제 퇴장 화면에서 취소 버튼을 눌러 강제 퇴장을 취소한다. …5c2. 시스템은 그룹 강제 퇴장 화면을 닫는다.</p>
8	<p>8a. 그룹명이 20자를 초과하는 경우 그룹 수정에 실패한다. …8a1. 시스템은 그룹 수정 실패 알림 메시지를 출력한다. …8a2. 그룹 정보 작성 단계로 돌아간다.</p> <p>8b. 그룹 설명이 30자를 초과하는 경우 그룹 수정에 실패한다. ..8b1. 시스템은 그룹 수정 실패 알림 메시지를 출력한다. ..8b2. 그룹 정보 작성 단계로 돌아간다.</p>
RELATED INFORMATION	
Performance	≤ 1 second

Frequency	사용자당 하루 평균 1번
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #8 : Manage profile	
GENERAL CHARACTERISTICS	
Summary	이 use case는 프로필과 관련된 기본 기능들인 프로필 수정, 보기, 지난 약속들에 대한 통계 보기, 약속 포인트 활용을 설명한다. 약속 포인트는 사용자가 약속을 완료하면 제공되며 자신의 지난 약속에 대한 여러 통계 자료를 해금한다. Use case #1. Manage schedule에 의해 호출된다.
Scope	SSS (Sae hoon's scheduling system)
Level	User level
Author	XXX
Last Update	2022.11.30
Status	Analysis
Primary Actor	User
Preconditions	지난 약속 통계 보기를 수행하기 위해서 지난 약속이 최소 1개 이상 존재해야 하고 약속 포인트로 각 통계를 해금해야 한다.
Trigger	사용자가 인터페이스에서 프로필과 관련된 원하는 기능을 하려할 때
Success Post Condition	프로필을 본다. 프로필이 수정된다.
Failed Post Condition	프로필에 대한 변경사항이 반영되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 프로필과 관련된 기능들을 수행하려 한다.
1	사용자는 마이 페이지 버튼을 누른다.
2	시스템은 사진, 닉네임, 아이디, 이름, 나이, 성별에 대한 정보를 보여주는 창을 제공한다.
3	사용자는 통계 보기 버튼을 누른다.
4	시스템은 현재 약속 포인트와 통계 자료를 보여주는 창을 제공한다.
5	단계 2이후로 사용자는 자신의 프로필에 대한 정보를 어느 때나 수정할 수 있다.
EXTENSION SCENARIOS	
Step	Branching Action
6	6a. 다른 사람의 프로필 보기 …6a1. 사용자는 친구 목록에서 원하는 친구의 이름을 누른다. …6a2. 시스템은 해당 사용자의 프로필 화면과 공개된 약속들을 표시한다.
4	4a. 약속 포인트 활용 …4a1. 사용자는 통계 화면에서 포인트를 사용해 통계를 해금한다.

	…4a2. 시스템은 해금된 통계 정보를 제공한다.
RELATED INFORMATION	
Performance	≤ 1 second
Frequency	사용자당 하루 평균 1번
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #9 : Set configuration	
GENERAL CHARACTERISTICS	
Summary	알림, 초대 거부 등 시스템에서 제공하는 기능에 대한 설정을 변경 한다.
Scope	SSS (Sae hoon's scheduling system)
Level	User level
Author	XXX
Last Update	2022.11.30
Status	Analysis
Primary Actor	User
Preconditions	사용자의 프로필 화면이 열려 있어야 한다.
Trigger	시스템의 설정 버튼을 눌렀을 때
Success Post Condition	사용자는 설정 화면에서 설정을 변경할 수 있다.
Failed Post Condition	사용자는 설정 화면에서 설정을 변경할 수 없다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 시스템 기능을 설정한다.
1	사용자는 사용자의 프로필 화면에서 설정 버튼을 누른다.
2	시스템은 설정 화면을 제공한다.
3	사용자는 알림, 약속 혹은 그룹 초대를 허용/거부하도록 설정한다.
4	시스템은 변경사항을 저장한다.
EXTENSION SCENARIOS	
Step	Branching Action
RELATED INFORMATION	
Performance	≤ 2 seconds

Frequency	사용자당 하루 평균 1번
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #10 : Search user	
GENERAL CHARACTERISTICS	
Summary	다른 사용자를 검색한다. Use case #5. Manage friends에서 호출된다.
Scope	SSS (Sae hoon's scheduling system)
Level	User level
Author	XXX
Last Update	2022.11.30
Status	Analysis
Primary Actor	User
Preconditions	시스템에 본인을 제외한 다른 사용자가 한 명 이상 있어야 한다.
Trigger	검색하고자 하는 사용자의 닉네임 또는 아이디를 입력하고 검색 버튼을 누를 때
Success Post Condition	사용자는 검색 조건과 매치되는 사용자의 목록을 볼 수 있다.
Failed Post Condition	사용자는 검색 조건과 매치하는 사용자의 목록을 볼 수 없다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 다른 사용자를 검색한다.
1	사용자는 검색하고자 하는 사용자의 아이디나 닉네임을 입력한다.
2	사용자는 검색 버튼을 눌러서 사용자를 검색한다.
3	시스템은 1.에서 입력한 아이디나 닉네임과 일치하는 사용자 목록을 출력한다.
EXTENSION SCENARIOS	
Step	Branching Action
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	사용자당 하루 평균 2번
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #11 : Recommend a place	
GENERAL CHARACTERISTICS	
Summary	사용자들이 최근에 자주 방문한 인기 있는 약속 장소를 보여준다.
Scope	SSS (Sae hoon's scheduling system)
Level	User level
Author	XXX
Last Update	2022.11.03
Status	Analysis (Finalize)
Primary Actor	User
Preconditions	다른 사용자가 시스템을 이용해 수행한 약속이 한 개 이상 있어야 한다.
Trigger	약속 장소를 보기 위해 메인 화면에서 화면을 옆으로 넘기거나 버튼을 눌렀을 때
Success Post Condition	사용자는 인기 약속 장소를 볼 수 있다.
Failed Post Condition	사용자는 인기 약속 장소를 볼 수 없다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 인기 약속 장소를 본다.
1	사용자는 메인화면에서 화면을 옆으로 넘기거나 버튼을 누른다.
2	시스템은 인기 약속 장소들을 가져와 화면에 출력한다.
EXTENSION SCENARIOS	
Step	Branching Action
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	약속당 평균 1 번
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #12 : Track real-time location	
GENERAL CHARACTERISTICS	
Summary	약속 일정 시간 전부터 약속의 구성원들의 실시간 위치 정보와 도착 예상

	시간을 보여준다.
Scope	SSS (Sae hoon's scheduling system)
Level	User level
Author	XXX
Last Update	2022.11.30
Status	Analysis (Finalize)
Primary Actor	User
Preconditions	다른 사용자들과 공유하는 약속을 최소 한 개 이상 가지고 있어야 한다. 약속의 구성원들이 위치 정보제공 동의를 해야 한다.
Trigger	약속 구성원들의 위치와 예상 도착 시간을 실시간으로 알고 싶을 때
Success Post Condition	사용자는 자신과 약속 구성원들의 위치와 도착 예정 시간을 실시간으로 파악한다. 위치를 공유하고 있는 구성원들의 목록을 확인한다
Failed Post Condition	사용자는 다른 약속 구성원들의 위치 정보를 알 수 없다.

MAIN SUCCESS SCENARIO

Step	Action
S	사용자가 약속 구성원들의 위치를 본다.
1	사용자는 약속 상세 정보 화면의 세부 메뉴에서 위치 공유 버튼을 눌러 자신의 위치와 예상 도착시간을 약속 구성원들에게 공유한다.
2	시스템은 실시간 약속 추적 화면과 지도 화면을 제공한다.
3	시스템은 구성원들의 위치를 지도에 표시하고 예상 도착시간을 출력한다.
4	단계 1 이후로 사용자는 약속 구성원들의 위치와 예상 도착시간을 어느 때나 확인할 수 있다.

EXTENSION SCENARIOS

Step	Branching Action

RELATED INFORMATION

Performance	≤ 2 seconds
Frequency	알 수 없음
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #13 : Login

GENERAL CHARACTERISTICS

Summary	시스템을 사용하고자 하는 사용자가 API를 사용한 소셜 로그인을 하거나 자체 아이디를 통해 로그인을 한다. Use case #1. Manage schedule에 의해 호출된다.
Scope	SSS (Sae hoon's scheduling system)

Level	Subfunction level
Author	XXX
Last Update	2022.11.30
Status	Analysis
Primary Actor	User
Preconditions	<p>사용자는 회원가입이 되어있어야 한다.</p> <p>사용자는 현재 로그인되지 않은 상태여야 한다.</p>
Trigger	로그인 화면에서 아이디와 비밀번호를 입력한 후 로그인을 하려고 할 때
Success Post Condition	사용자는 로그인에 성공하여 시스템의 모든 기능을 사용할 수 있다.
Failed Post Condition	사용자는 로그인에 실패하여 시스템의 회원 가입을 제외한 모든 기능을 사용하지 못한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 로그인을 한다.
1	사용자는 로그인 화면에서 아이디와 비밀번호를 입력한 후 로그인 버튼을 누른다.
2	시스템은 사용자 정보를 체크하여 로그인 성공 유무를 판단한다.
3	시스템은 등록된 사용자라면 로그인에 성공하고 약속 목록 화면을 제공한다.
EXTENSION SCENARIOS	
Step	Branching Action
1	1a. 자동 로그인 …1a1. 시스템은 기존에 사용자가 로그인된 기록이 있다면 로그인 성공으로 처리한 후 Use case를 끝낸다.
1	1b. 소셜 로그인 …1b1. 사용자는 로그인 화면에서 아이디와 비밀번호를 입력하지 않고, Naver 로그인 또는 Google 로그인 버튼을 누른다. …1b2. 시스템은 연동할 Naver 또는 Google 계정의 로그인 화면을 제공한다. …1b3. 사용자는 해당 계정의 아이디와 비밀번호를 입력한 후 로그인 버튼을 누른다.
3	3a. 등록되지 않은 정보면 로그인에 실패한다. …3a1. 로그인 실패 메시지를 출력한다. …3a2. 아이디 입력 단계로 돌아간다. (Use case #14-1)
RELATED INFORMATION	
Performance	≤ 1 second
Frequency	사용자당 하루 평균 1번
<Concurrency>	제한 없음
Due Date	2022.11.05

Use case #14 : Invite user	
GENERAL CHARACTERISTICS	
Summary	그룹 혹은 약속에 사용자를 초대한다. Use case #3. Manage plan, Use case #7. Manage groups에 의해 호출된다.
Scope	SSS (Sae hoon's scheduling system)
Level	Subfunction level
Author	XXX
Last Update	2022.11.04
Status	Analysis
Primary Actor	User
Preconditions	친구가 한 명 이상 있어야 한다. 그룹 혹은 약속의 상세 정보 화면이 열려 있어야 한다.
Trigger	그룹 혹은 약속에 사용자를 초대하려고 할 때
Success Post Condition	초대에 성공한다.
Failed Post Condition	초대에 실패한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 다른 사용자를 초대한다.
1	사용자는 그룹 혹은 약속의 상세 정보 화면의 세부 메뉴에서 초대 버튼을 누른다.
2	시스템은 사용자의 친구 목록을 출력한다.
3	사용자는 친구 목록에서 초대할 사용자들을 선택하고 확인 버튼을 누른다.
4	시스템은 대상 사용자들에게 푸시 알림을 보낸다.
5	대상 사용자들은 Use case #4. Process invitation을 사용해 초대 알림을 처리한다.
EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 초대 취소 …3a1. 사용자는 그룹 혹은 약속 초대 화면에서 취소 버튼을 눌러 초대를 취소한다. …3a2. 시스템은 초대 화면을 닫는다.
RELATED INFORMATION	
Performance	≤ 1 second
Frequency	사용자당 하루 평균 1번
<Concurrency>	제한 없음
Due Date	2022.11.05

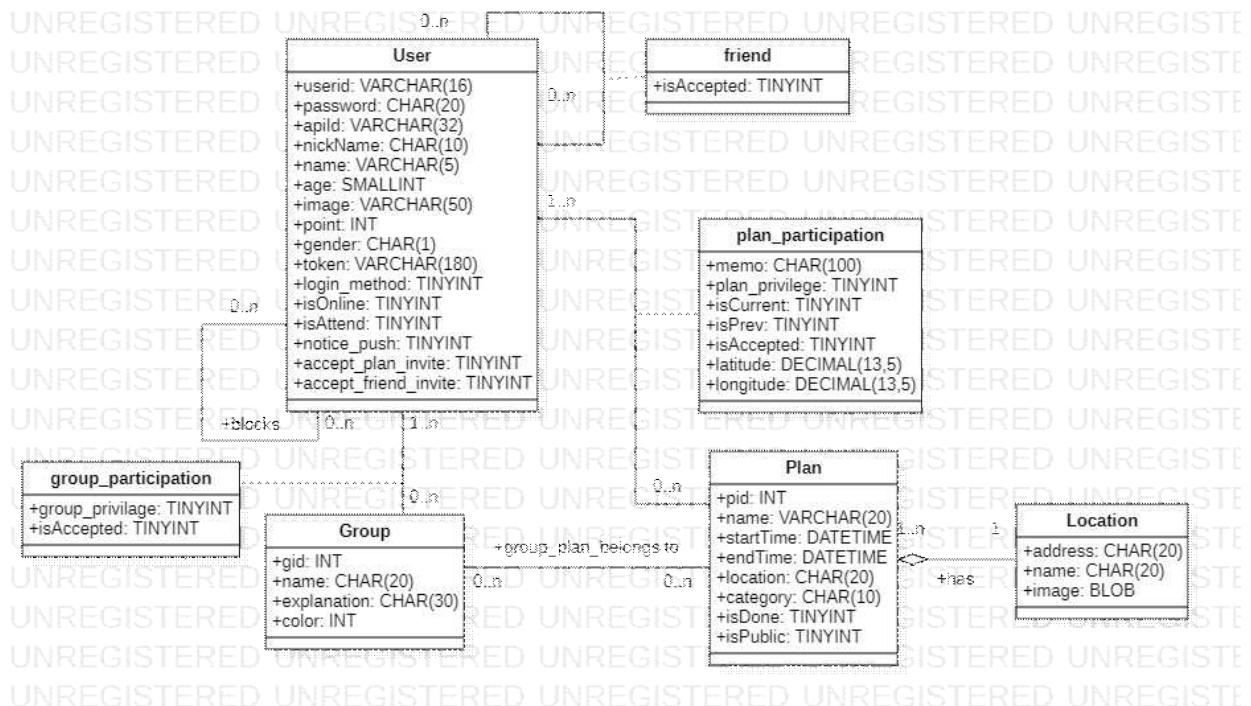
3. Class diagram

이번 장은 다양한 관점에서 바라본 class diagram(이하 CD)과 각각에 대한 설명을 첨부한다. CD를 볼 때 고려해야 할 사항은 다음과 같다.

- 본 시스템의 CD는 서버용과 안드로이드 어플리케이션용으로 나뉜다.
- 안드로이드 어플리케이션용으로 작성한 CD는 기능, 상속, DB 접근로 나눌 수 있다.
 - 각각의 기능 CD는 하나 혹은 여러 개의 use case와 연결된다.
 - 기능을 제외한 CD은 구현 시 고려해야 할 사항들에 대한 것이며 각각의 description 파트에서 역할을 자세히 설명한다.
- 데이터 타입과 Util, static은 CD로 그리진 않았지만 추출한 클래스들을 한데 모아 별도의 장들로 설명한다.

3.1 DB Class diagram

- 서버의 구조를 파악하기 위해 DB의 관점에서 본 CD을 작성했다.
 - ER diagram을 먼저 작성한 후 이를 CD로 변환시켰으며[5] 이를 domain analysis의 결과로 나온 domain CD로 봐도 충분할 것이다.



[그림 3-1] DB Class diagram

User			
Class Description	사용자의 정보를 저장하는 class		
구분	Name	Type	Visibility
Attributes	age 사용자의 나이를 저장하는 변수	SMALLINT	private
	image 사용자의 프로필 사진의 경로를 나타내는 변수	VARCHAR(50)	private
	userid 사용자를 구분하기 위한 고유의 변수	INT	private
	password 사용자의 비밀번호를 나타내는 변수	CHAR(20)	private
	login_method 로그인 방법을 나타내는 변수	TINYINT	private
	point 사용자의 포인트를 나타내는 변수	INT	private
	nickName 사용자의 별명을 나타내는 변수	CHAR(10)	private
	apiId Api 로그인시 사용되는 변수	SMALLINT	private
	isOnline 사용자의 접속 상태를 나타내는 변수	TINYINT	private
	token 푸시 알림을 보내기 위해 firebase에서 받는 토큰	VARCHAR(180)	private
	isAttend 사용자의 약속 참석 여부를 나타내는 변수	TINYINT	private
	noticeOption 알림 설정 상태를 변경하는 변수	TINYINT	private
	planInviteOption 약속 초대 설정 상태를 변경하는 변수	TINYINT	private
	friendInviteOption 친구 초대 설정 상태를 변경하는 변수	TINYINT	private

Plan			
Class Description	약속 정보를 저장하는 class		
구분	Name	Type	Visibility
Attributes	name 약속 이름을 나타내는 변수	VARCHAR(20)	private
	category 약속이 포함된 카테고리를 나타내는 변수	CHAR(10)	private
	pid 약속을 구분하기 위한 고유의 변수	INT	private
	isDone 약속의 완료 여부를 나타내는 변수	TINYINT	private
	endTime 약속이 끝나는 시간을 나타내는 변수	DATETIME	private
	startTime 약속이 시작하는 시간을 나타내는 변수	DATETIME	private
	location 약속 장소를 나타내는 변수	CHAR(20)	private
	isPublic	TINYINT	private

	약속 공개 여부를 설정하는 변수
--	-------------------

Location

Class Description	약속 장소 정보를 저장하는 class		
구분	Name	Type	Visibility
		Description	
Attributes	name 약속 장소명을 나타내는 변수	CHAR(20)	private
	address 약속 장소의 주소를 나타내는 변수	CHAR(20)	private
	image 약속 장소의 사진을 나타내는 변수	BLOB	private

Group

Class Description	그룹 정보를 저장하는 class		
구분	Name	Type	Visibility
		Description	
Attributes	gid 그룹을 구분하기 위한 고유의 id 변수	INT	private
	color 그룹 색깔을 나타내는 변수	INT	private
	name 그룹 이름을 나타내는 변수	CHAR(20)	private
	explanation 그룹 설명을 나타내는 변수	CHAR(30)	private

group_participation

Class Description	사용자와 그룹 관계에 필요한 정보를 저장하는 class		
구분	Name	Type	Visibility
		Description	
Attributes	group_privilege 그룹원의 권한을 나타내는 변수	TINYINT	private
	isAccepted 그룹 초대 여부를 나타내는 변수	TINYINT	private

plan_participation

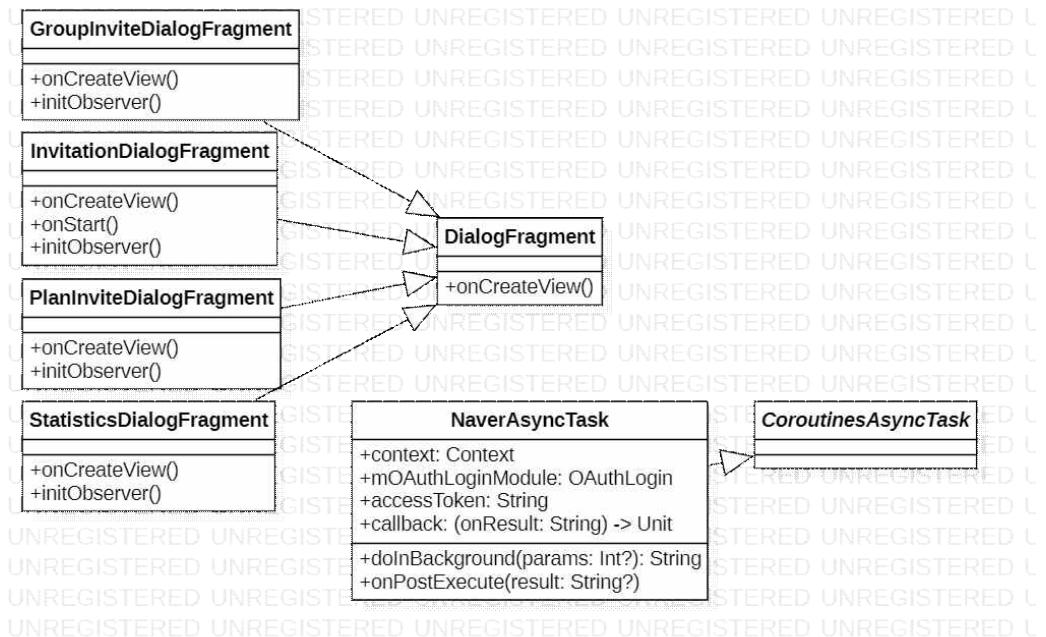
Class Description	사용자와 약속 관계에 필요한 정보를 저장하는 class		
구분	Name	Type	Visibility
		Description	
Attributes	memo 약속 메모를 나타내는 변수	CHAR(100)	private
	plan_privilege 약속 구성원의 권한을 나타내는 변수	TINYINT	private
	isCurrent (모름)	TINYINT	private
	isAccepted 약속 수락 여부를 나타내는 변수	TINYINT	private
	latitude 장소의 위도를 나타내는 변수	DECIMAL(13,5)	private
	longitude	DECIMAL(13,5)	private

	장소의 경도를 나타내는 변수
--	-----------------

friend			
Class	사용자 간 친구 관계를 관리하는 테이블		
구분	Name	Type	Visibility
Attributes	isAccepted	TINYINT	private
	상대방이 친구 추가 요청을 수락했는지 저장하는 변수		

3.2 상속 Class diagram

- [그림 3-2]는 DialogFragment와 이를 상속하는 클래스들의 관계를 나타낸 CD이다.
- Naver AsyncTask와 Coroutines AsyncTask에 관한 설명은 Util Class Diagram Description에 포함되어 있다.



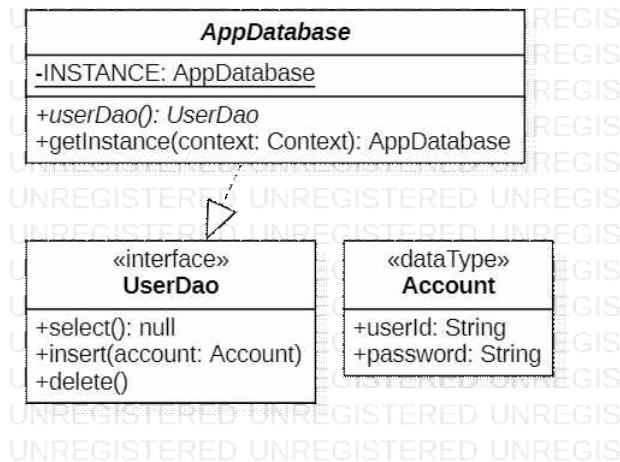
[그림 3-2] 상속 Class Diagram

DialogFragment	
----------------	--

DialogFragment	
Class	출력된 사용자 목록에서 원하는 사용자들을 선택하고 확인할 때 쓰는 fragment.
Description	메모 등록도 이름이 DialogFragment로 끝나지만 이 클래스를 상속받지 않는 이유는 사용자 목록을 출력하지 않고 문장(메모)만 입력받기 때문이다.

3.3 DB 접근 Class diagram

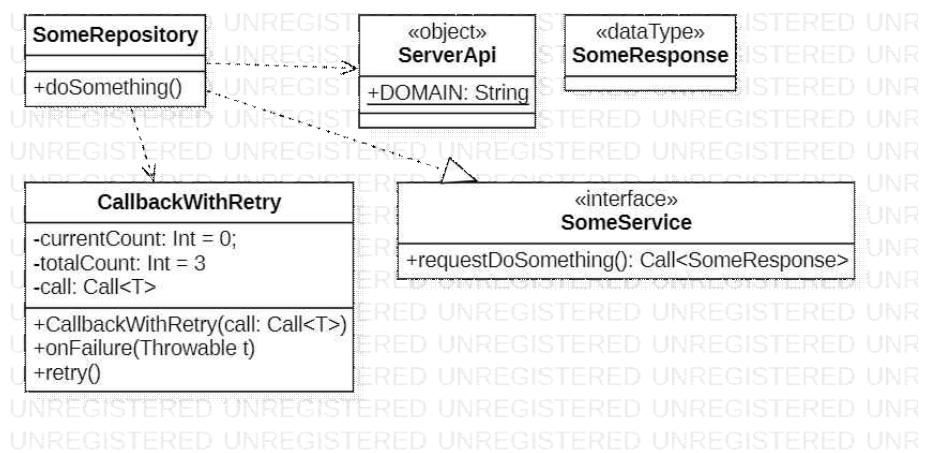
이 절은 데이터베이스를 접근하기 위한 클래스를 추출해 CD로 그린 결과물을 나타낸다.



[그림 3-3] 내부 DB 접근 클래스 다이어그램

AppDatabase			
Class Description	로컬 데이터베이스 인스턴스를 생성하고 제공하는 class		
구분	Name	Type	Visibility
Attributes	INSTANCE 로컬 데이터베이스 인스턴스를 저장하는 객체	AppDatabase	private
구분	Name	Argument	Returns
Operations	getInstance 로컬 데이터베이스 인스턴스를 가져오는 operation이다. 인자로 Context 타입의 context를 인자로 받는다.	Context	AppDatabase
	userDao User 테이블에 접근할 수 있는 DAO 객체를 가져오는 operation이다.	none	UserDao

UserDao			
Class Description	User 테이블에 접근할 수 있는 method를 제공하는 class		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
	select	none	Account
	User 테이블에서 계정을 조회하는 operation이다.		
	insert	Account	none
	User 테이블에 계정을 삽입하는 operation이다.		
	delete	none	none
	User 테이블에 계정을 모두 삭제하는 operation이다.		



[그림 3-4] 외부 서버 접근 클래스 다이어그램

CallbackWithRetry

Class Description	실패한 요청을 서버에 재시도하기 위해 사용하는 class		
구분	Name	Type	Visibility
	Description		
Attributes	currentCount	Int	private
	현재 서버에 시도한 요청의 횟수를 담는 객체		
	totalCount	Int	private
Operations	서버에 시도할 수 있는 요청의 최대 횟수를 담는 객체		
	call	Call<T>	private
	서버에 요청을 보내고 응답을 받는 Retrofit의 호출을 담는 객체		
구분	Name	Argument	Returns
	Description		
Operations	CallbackWithRetry	Call<T>	none
	Retrofit의 호출을 가져와서 저장하는 생성자이다. 요청하고 응답을 받는 Retrofit의 호출인 Call<T> 타입의 call을 인자로 받는다.		
	onFailure	Throwable	none
Operations	서버에 요청에 실패하였을 때 실행되는 operation이다.		
	retry	none	none
	서버에 요청을 다시 시도하는 operation이다.		

SomeRepository

Class Description	특정한 기능 처리와 서버의 요청 및 응답을 담당하는 class		
구분	Name	Type	Visibility
	Description		
Attributes	None		
구분	Name	Argument	Returns
	Description		
Operations	doSomething	none	none
	특정한 기능을 수행하기 위해 서버로 요청을 보내고 응답을 받는 operation이다.		

SomeService

Class Description	서버로 보내기 위한 http 요청이 있는 interface		
구분	Name	Type	Visibility
	Description		
Attributes	None		
구분	Name	Argument	Returns
	Description		
Operations	requestDoSomething	none	Call<SomeResponse>
	인자로 받은 값을 쿼리 파라미터로 사용하여 웹서버 요청을 만드는 operation		

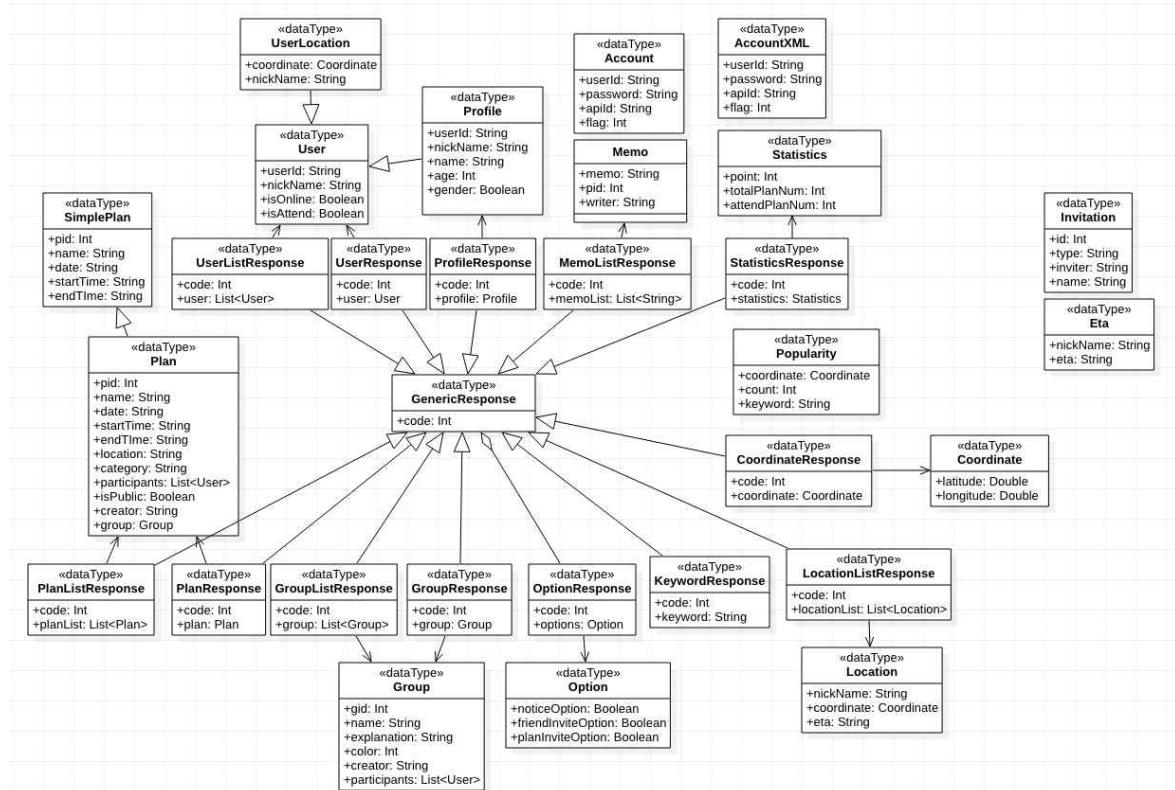
SomeResponse

Class Description	서버로 보낸 요청에 대한 응답을 담는 data class		
구분	Name	Type	Visibility
Description			
Attributes	None		
구분	Name	Argument	Returns
Description			
Operations	None		

ServerApi

Class Description	서버의 도메인 주소를 저장하는 object class		
구분	Name	Type	Visibility
Description			
Attributes	DOMAIN	String	public
서버의 도메인 주소를 담는 객체			
구분	Name	Argument	Returns
Description			
Operations	None		

3.4 데이터 타입 Classes



[그림 3-5] DataType Class Diagram

UserPlanPK			
Class Description	사용자 정보와 약속 정보에 대한 primary key를 가지는 data		
구분	Name	Type	Visibility
Attributes	pid 약속을 구분하기 위한 고유의 id 변수	Int	private
	uid 사용자를 구분하기 위한 고유의 id 변수	String	private
구분	Name	Argument	Returns
Operations	None		

User			
Class Description	사용자 정보를 저장하는 data		
구분	Name	Type	Visibility
	Description		
Attributes	userId 사용자를 구분하기 위한 고유의 id 변수	String	private
	nickName 사용자의 별명을 나타내는 변수	Boolean	private
구분	Name	Argument	Returns
	Description		
Operations	None		

Friend			
Class Description	사용자 정보와 상태를 저장하는 data		
구분	Name	Type	Visibility
	Description		
Attributes	userId 사용자를 구분하기 위한 고유의 id 변수	String	private
	nickName 사용자의 별명을 나타내는 변수	String	private
	isOnline 사용자가 접속 중인지 여부를 나타내는 변수	Boolean	private
	isAttend 사용자가 약속 진행 중인지 여부를 나타내는 변수	Boolean	private
구분	Name	Argument	Returns
	Description		
Operations	None		

SimplePlan			
Class Description	약속에 대한 간단한 정보를 저장하는 data		
구분	Name	Type	Visibility
	Description		
Attributes	pid 약속을 구분하기 위한 고유의 id 변수	Int	private
	name 약속의 이름을 나타내는 변수	String	private
	date 약속의 날짜를 나타내는 변수	String	private
	startTime 약속이 시작하는 시간을 나타내는 변수	String	private
	endTime 약속이 끝나는 시간을 나타내는 변수	String	private
구분	Name	Argument	Returns
	Description		
Operations	None		

Plan

Class Description	약속 정보를 저장하는 data		
구분	Name	Type	Visibility
Description			
Attributes	pid 약속을 구분하기 위한 고유의 id 변수	Int	private
	name 약속의 이름을 나타내는 변수	String	private
	date 약속의 날짜를 나타내는 변수	String	private
	startTime 약속이 시작하는 시간을 나타내는 변수	String	private
	endTime 약속이 끝나는 시간을 나타내는 변수	String	private
	location 약속 장소를 나타내는 변수	String	private
	category 약속이 포함된 카테고리를 나타내는 변수	String	private
	participants 약속 참석자들을 나타내는 List 변수	List<User>	private
Operations	구분	Name	Argument
	Description		
Operations	None		

Memo

Class Description	메모 내용과 작성자를 저장하는 data		
구분	Name	Type	Visibility
Description			
Attributes	memo 메모 내용을 나타내는 변수	String	private
	nickName 메모를 작성한 사용자의 별명을 나타내는 변수	String	private
Operations	구분	Name	Argument
	Description		
Operations	None		

Account

Class Description	사용자의 계정 정보를 저장하는 data		
구분	Name	Type	Visibility
Description			
Attributes	userId 사용자를 구분하기 위한 고유의 id 변수	String	private
	password 사용자 계정의 비밀번호를 나타내는 변수	String	private
Operations	구분	Name	Argument
	Description		
Operations	None		

Profile			
Class	프로필 정보를 저장하는 data		
구분	Name	Type	Visibility
			Description
Attributes	userId	String	private
	사용자를 구분하기 위한 고유의 id 변수		
	nickName	String	private
	사용자의 별명을 나타내는 변수		
	name	String	private
	사용자의 이름을 나타내는 변수		
Operations	age	Int	private
	사용자의 나이를 나타내는 변수		
	gender	Boolean	private
	사용자의 성별을 나타내는 변수		
	Image	Int	private
	사용자의 프로필 사진을 나타내는 변수		
구분	Name	Argument	Returns
			Description
Operations	None		

Statistics			
Class	통계 정보를 저장하는 data		
구분	Name	Type	Visibility
			Description
Attributes	point	Int	private
	사용자의 현재 포인트를 나타내는 변수		
	totalPlanNum	Int	private
	총 약속 횟수를 나타내는 변수		
	attendPlanNum	Int	private
	약석 참석 횟수를 나타내는 변수		
Operations	absentPlanNum	Int	private
	약속 불참 횟수를 나타내는 변수		
	attendPercentage	Double	private
	약속 참여율을 나타내는 변수		
구분	Name	Argument	Returns
			Description
Operations	None		

Popularity			
Class Description	인기 약속 장소 정보를 저장하는 data		
구분	Name	Type	Visibility
		Description	
	coordinate 약속 장소를 나타내는 변수	Coordinate	private
	count 약속 장소로 선택된 횟수를 나타내는 변수	Int	private
	keyword 약속 장소의 지명을 나타내는 변수	String	private
구분	Name	Argument	Returns
		Description	
Operations	None		

Coordinate			
Class Description	약속 장소의 위치 정보를 저장하는 data		
구분	Name	Type	Visibility
		Description	
	latitude 약속 장소의 위도를 나타내는 변수	Double	private
	langitude 약속 장소의 경도를 나타내는 변수	Double	private
구분	Name	Argument	Returns
		Description	
Operations	None		

GenericResponse			
Class Description	서버의 응답을 저장하는 data		
구분	Name	Type	Visibility
		Description	
	code 서버의 응답 상태를 나타내는 변수	Int	private
구분	Name	Argument	Returns
		Description	
Operations	None		

UserListResponse			
Class Description	서버의 사용자 목록을 저장하는 data		
구분	Name	Type	Visibility
		Description	
	code 서버의 응답 상태 코드를 나타내는 변수	Int	private
	userList 사용자 목록을 나타내는 List 변수	ArrayList<User>	private
구분	Name	Argument	Returns
		Description	
Operations	None		

OcrResponse

Class Description	서버의 Ocr 정보를 저장하는 data		
구분	Name	Type	Visibility
	Description		
Attributes	code 서버의 응답 상태 코드를 나타내는 변수	Int	private
	recognizedStr Ocr로 인식된 문자열을 나타내는 변수	String	private
구분	Name	Argument	Returns
	Description		
Operations	None		

PopularityResponse

Class Description	서버의 인기 약속 장소를 저장하는 data		
구분	Name	Type	Visibility
	Description		
Attributes	code 서버의 응답 상태 코드를 나타내는 변수	Int	private
	popularity 인기 약속 장소 목록을 나타내는 List 변수	List<Popularity>	private
구분	Name	Argument	Returns
	Description		
Operations	None		

KeywordResponse

Class Description	서버의 인기 약속지명을 저장하는 data		
구분	Name	Type	Visibility
	Description		
Attributes	code 서버의 응답 상태 코드를 나타내는 변수	Int	private
	keyword 약속 장소의 지명을 나타내는 변수	String	private
구분	Name	Argument	Returns
	Description		
Operations	None		

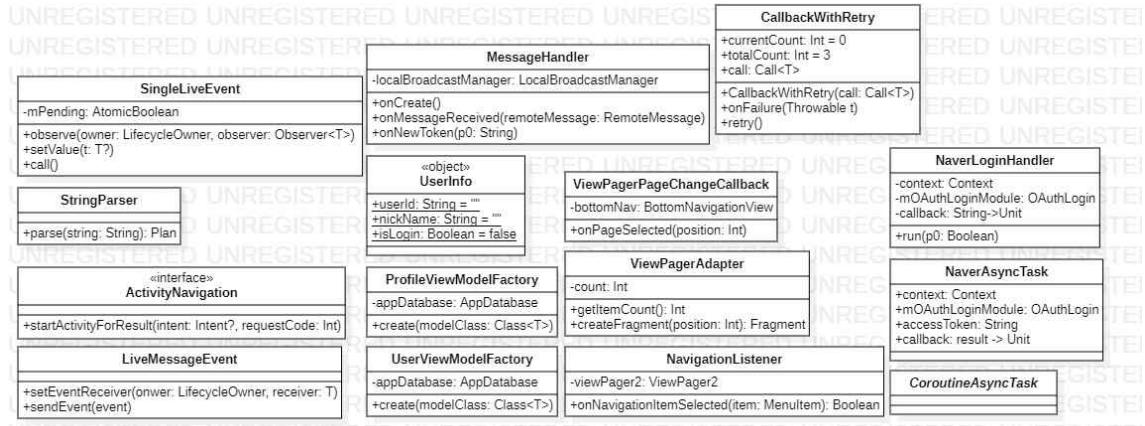
CoordinateResponse

Class Description	서버의 약속 장소 위치를 저장하는 data		
구분	Name	Type	Visibility
	Description		
Attributes	code 서버의 응답 상태 코드를 나타내는 변수	Int	private
	coordinate 약속 장소 위치를 나타내는 변수	Coordinate	private
구분	Name	Argument	Returns
	Description		
Operations	None		

etaResponse			
Class Description	서버의 도착 예정 시간을 저장하는 data		
구분	Name	Type	Visibility
Attributes	code	Int	private
	서버의 응답 상태 코드를 나타내는 변수		
Operations	eta	LocalTime	private
	도착 예정 시간을 나타내는 변수		
구분	Name	Argument	Returns
Operations	None		
		Description	

3.5 Util and Static Classes

- [그림 3-6]는 클라이언트 어플리케이션에서 사용하는 유ти 클래스와 정적 클래스를 모아둔 그림이다.
- UserInfo는 3.6절에, CallbackWithRetry는 3.3절에 설명한다.
- 특성상 구현 방법에 상당히 의존하기 때문에 개발 중 변동성이 클 가능성성이 높다.



[그림 3-6] object와 유ти 클래스들

SingleLiveEvent

Class Description	MutableLiveData를 상속받아 보다 이벤트 호출에 적합하게 만든 클래스		
구분	Name	Type	Visibility
Attributes	mPending 사용자가 호출할 때만 값이 변경되는 것을 보장하게 한다.	AtomicBoolean	private
구분	Name	Argument	Returns
Operations	observe 람다식을 받아 옵저버로 등록한다. setValue 값을 바꾼다. 이 때 mPending을 이용해 MutableLiveData와 차별을 둔다. call 값을 바꾸지 말고 옵저버만 호출한다.	LifecycleOwner, Observer<T> t: T? Unit	public public public

StringParser

Class Description	OCR 혹은 타이핑으로 약속 생성 시 문자열을 파싱해 약속으로 생성하는 유ти 클래스		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
Operations	parse 문자열을 받아 파싱해 약속으로 변환한다	String	Plan

ActivityNavigation

Class Description	구글 로그인 기능을 MVVM에 맞게 사용하기 위해 쓰는 인터페이스 출처: https://stackoverflow.com/questions/52287013/google-facebook-sign-in-in-mvvm		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
Operations	startActivitiyForResult 구글 로그인 activity를 실행하기 위한 함수	Intent?, Int	public

LiveMessageEvent

Class Description	구글 로그인 기능을 MVVM에 맞게 사용하기 위해 쓰는 클래스 출처: https://stackoverflow.com/questions/52287013/google-facebook-sign-in-in-mvvm		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
Operations	setEventReceiver 람다식을 받아 옵저버로 등록한다. sendEvent 이벤트를 보낸다. 등록된 옵저버가 호출되면서 이벤트가 처리된다.	LifecycleOwner, T T.() -> Unit	

MessageHandler

Class Description	Firebase에서 오는 알림(약속/그룹 초대)을 관리한다. 단, 어플이 foreground에 있을 때만 이 클래스를 이용해 관리하며 background에 있다면 푸시 알림을 보낸다.		
구분	Name	Type	Visibility
Attributes	localBroadcastManager	LocalBroadcastManager	private
Operations	받은 메시지를 액티비티로 보내는 변수이다.		
구분	Name	Argument	Returns
Operations	onCreate		
	localBroadcastManager를 초기화한다.		
	onMessageReceived	RemoteMessage	
Operations	메시지가 왔을 때를 처리한다. localBroadcastManager를 이용한다.		
	onNewToken	String	
Operations	새로운 토큰을 받는다. 현 시스템에선 활용되지 않는다.		

ProfileViewModelFactory / UserViewModelFactory

Class Description	ProfileViewModel과 UserViewModel은 모두 각 Repository에 로컬 DB에 접근하기 위한 변수를 줘야 한다. 따라서 생성자로 변수 appDatabase를 받게 하기 위해 factory를 사용한다.		
구분	Name	Type	Visibility
Attributes	appDatabase	AppDatabase	private
Operations	로컬 DB를 접근하기 위한 변수		
구분	Name	Argument	Returns
Operations	create	Class<T>	T
	appDatabase를 가지는 뷰모델을 반환한다.		

ViewPagerPageChangeCallback

Class Description	슬라이딩으로 화면이 전환될 때 하단 메뉴도 같이 변경되게 하는 클래스		
구분	Name	Type	Visibility
Attributes	bottomNav	BottomNavigationView	private
Operations	하단 메뉴		
구분	Name	Argument	Returns
Operations	onPageSelected	Int	
	변환되는 위치를 인자로 받아 하단 메뉴에 반영한다.		

ViewPagerPageChangeCallback

Class Description	슬라이딩으로 화면이 전환될 때 하단 메뉴도 같이 변경되게 하는 클래스		
구분	Name	Type	Visibility
Attributes	bottomNav	BottomNavigationView	private
Operations	하단 메뉴		
구분	Name	Argument	Returns
Operations	onPageSelected	Int	
	변환되는 위치를 인자로 받아 하단 메뉴에 반영한다.		

ViewPagerAdapter

Class Description	ViewPager2로 이동시 사용되는 클래스이다.		
구분	Name	Type	Visibility
	Description		
Attributes	count adapter에 있는 아이템의 수	Int	private
	fa 상위 클래스에서 사용하는 변수	FragmentActivity	
구분	Name	Argument	Returns
	Description		
Operations	getItemCount 람다식을 받아 옵저버로 등록한다.		Int
	createFragment 화면 전환에 따라 새로운 fragment를 생성한다.	Int	Fragment

NavigationListener

Class Description	하단 메뉴에 따라 화면을 전환시키는 클래스. Viewpager2에 반영시킨다.		
구분	Name	Type	Visibility
	Description		
Attributes	viewPager2 viewPager2 변수	ViewPager2	private
구분	Name	Argument	Returns
	Description		
Operations	onNavigationItemSelected 하단 메뉴가 선택되면 호출되면 viewPager2의 상태를 바꾼다.	MenuItem	Boolean

CoroutineAsyncTask

Class Description	기존의 AsyncTask를 코루틴 버전으로 변형한 코드. 네이버 로그인시 사용한다. 출처: https://github.com/ladrahul25/CoroutineAsyncTask/blob/master/app/src/main/java/com/example/background/CoroutinesAsyncTask.kt		
구분	Name	Type	Visibility
	Description		
Attributes	None		
구분	Name	Argument	Returns
	Description		
Operations	None		

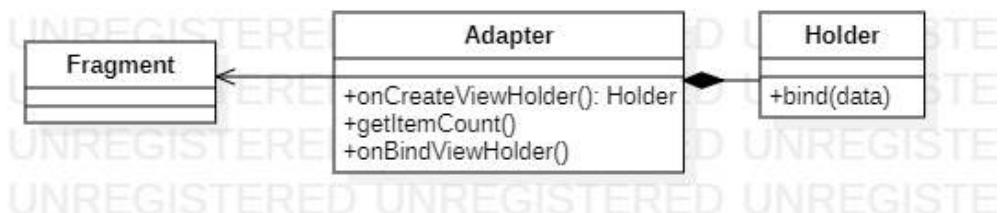
NaverAsyncTask			
Class Description	네이버 로그인을 비동기 작업으로 처리하게 하는 클래스		
구분	Name	Type	Visibility
	Description		
Attributes	context, mOAuthLoginModule, accessToken api 요청을 위해 사용한다	Context, OAuthLogin, String	private
	callback 작업 완료 후 호출하는 함수	String -> Unit	
구분	Name	Argument	Returns
	Description		
Operations	doInBackground 작업 시작	Int?	String
	onPostExecute 작업 결과 처리	String?	

NaverLoginHandler			
Class Description	네이버 로그인에 관련된 리스너 등을 관리하는 클래스		
구분	Name	Type	Visibility
	Description		
Attributes	context, mOAuthLoginModule api 요청에 사용함. 여기서 accessToken을 이용해서 NaverAsyncTask를 실행한다.	Context, OAuthLogin	private
	callback 작업 완료 후 호출하는 함수. NaverAsyncTask에 인자로 넘긴다	String -> Unit	
구분	Name	Argument	Returns
	Description		
Operations	run 네이버 로그인 버튼 클릭 시에 호출된다.	Boolean	

3.6 기능 Class diagram

- MVVM 아키텍처를 적용해 설계했다.
- 구글이 single-Activity architecture를 추천함에 따라[7] View는 모두 Fragment로 표시한다. 다만 activity는 2개를 사용했다. LoginActivity에서 사용자 인증/가입에 관련된 활동을 담당하고 MainActivity에서 대부분의 기능을 담당한다.
- Viewpager2와 navigation, bottom navigation의 통합적인 사용을 위해 container fragment를 사용한다. Friend/ Group /Plan /ProfileContainer Fragment로 총 4개이며 다른 클래스들과 직접적인 연관이 없기 때문에 따로 CD에 표시하지 않는다.
- Fragment 클래스들에 공통적으로 나타나는 initObserver()는 MutableLiveData 혹은 SingleLiveEvent의 observer를 초기화하는 operation이다.
- 모든 Fragment 클래스에 공통적으로 나타나는 onCreateView()와 initObserver()는 따로 설명에 적지 않는다.

- 일반적인 MVVM에서의 Viewmodel과 달리 본 앱에서는 가독성 향상과 역할의 분리를 위해 이를 Viewmodel과 Logic으로 분리한다. 실제로는 Viewmodel이 Logic을 가지고 있으므로 하나의 layer이다.
- Repository의 operation들은 내부DB 혹은 시스템 내의 서버와 통신하기 위한 목적으로 생성됐으며 전자는 Room, 후자는 Retrofit을 사용한다는 가정 하에 설계했다.
- AppDatabase, ServerAPI, UserDao, ServerService는 모두 DB 접근을 위한 클래스이고 DB [3.3 DB 접근 CD]에 자세한 설명이 있다.
- 구현 언어는 Kotlin이다. 코틀린은 항상 내부적으로 getter와 setter로 변수에 접근하기 때문에 특수한 목적이 있지 않는 한 따로 이를 operation으로 작성해줄 필요가 없다.
- Var 변수는 외부 참조 여부에 따라 setter, getter의 visibility를 고려하여 작성한다. (기본은 private)
- Val 변수는 값을 수정할 수 없어 setter에 대해 고려하지 않으므로 getter가 자동으로 구현된 private attribute로 간주한다.
- 각 fragment에서 리스트를 사용하는 경우(예를 들어, 친구 목록, 약속 목록 등), adapter와 holder를 사용한다. 하지만 기능 CD마다 이를 넣는 것은 가독성을 해치는 중복 문제를 만들기 때문에 여기서 일반화된 CD를 제시한다. 구현시 RecyclerView를 사용한다.

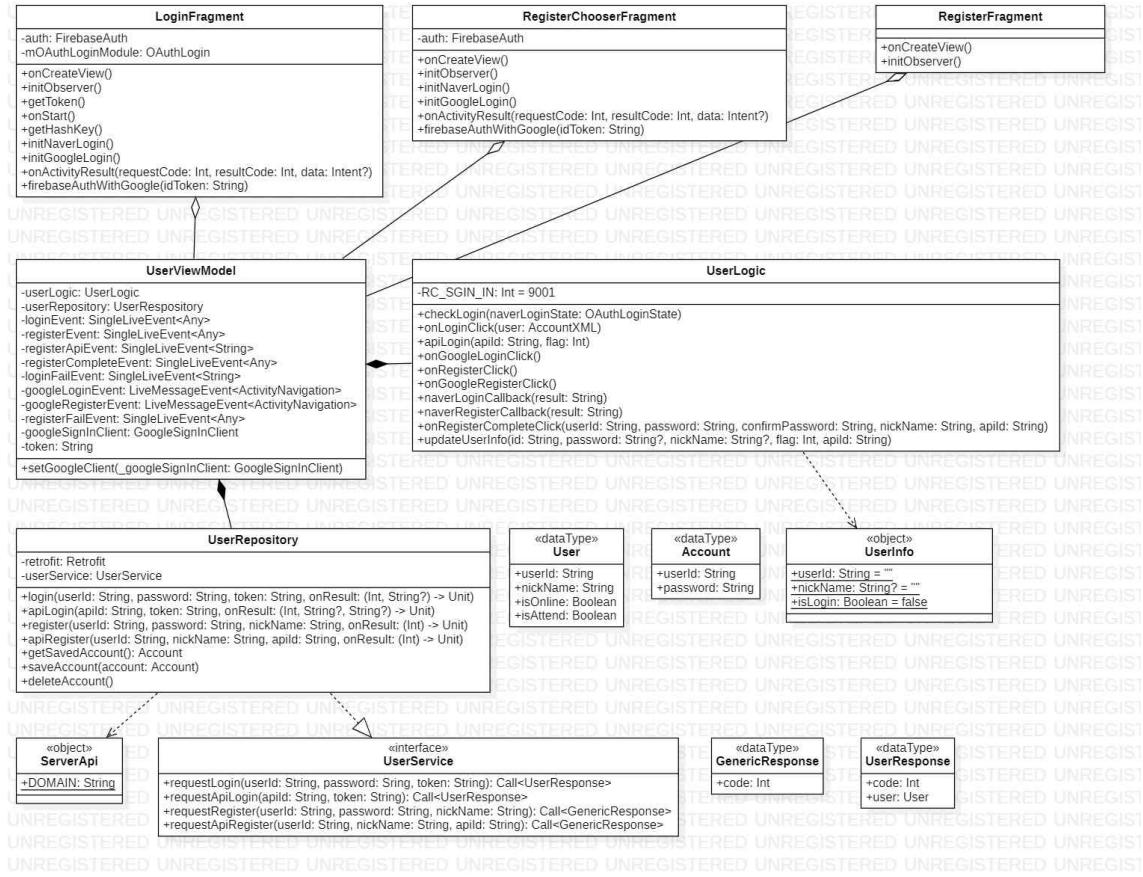


[그림 3-7] 리스트 표현 시 생기는 구조

Adapter			
Class	특정 데이터를 recyclerView에 표시되는 뷰와 바인딩하는 클래스		
Description	RecyclerView.Adapter를 상속받음		
구분	Name	Type	Visibility
	Description		
Attributes	None		
구분	Name	Argument	Returns
	Description		
Operations	onCreateViewHolder 바인딩 된 헌더를 반환하는 함수		Holder
	getItemCount 어댑터에 있는 아이템의 수를 반환하는 함수		Int
	onBindViewHolder 특정 위치의 아이템을 표시하기 위해 사용되는 함수		

Holder			
Class	각 아이템의 뷰와 데이터를 설명하는 클래스		
Description	RecyclerView.ViewHolder를 상속받음		
구분	Name	Type	Visibility
	Description		
Attributes	None		
구분	Name	Argument	Returns
	Description		
Operations	bind 뷰에 아이템을 바인딩함	Item	

3.6.1 User class diagram



[그림 3-8] User 기능 클래스 다이어그램

LoginFragment			
Class	사용자가 로그인할 때 제공하는 class		
구분	Name	Type	Visibility
Attributes	auth 구글 로그인시 사용한다. 그리고 구글 로그인을 사용하지 않는 사용자가 OCR을 사용하기 위한 익명 로그인시 사용하는 변수	FirebaseAuth	private
	mOAuthLoginModule 네이버 로그인시 사용하는 변수	OAuthLogin	private
구분	Name	Argument	Returns
Operations	getToken 푸시 알림을 위해 사용자의 토큰을 받아 서버로 보낸다.	none	none
	onStart OCR 사용을 위한 익명 로그인을 한다. Activity를 요하기 때문에 fragment에 넣는다.	none	none
	getHashKey Api 등록 요청을 위해 사용하는 operation. 후에 출시 시엔 삭제해도 된다.	none	none
	initNaverLogin 네이버 로그인과 관련된 클릭 리스너, 이미지 등을 세팅하는 operation	none	none
	initGoogleLogin 네이버 로그인과 관련된 리스너 등을 세팅하는 operation	none	none
	onActivityResult 구글 로그인 정보를 받은 operation	Int, Int, Intent?	none

<code>firebaseAuthWithGoogle</code>	String	none
OnAcvitiyResult에서 호출되어 실질적인 로그인을 처리하는 operation.		

RegisterFragment

Class Description	사용자가 로컬 회원가입 할 때 제공하는 class		
구분	Name	Type	Visibility
	Description		
Attributes	None		
구분	Name	Argument	Returns
	Description		
Operations	None		

RegisterChooserFragment

Class Description	사용자가 회원가입 방법을 선택할 때 제공하는 class		
구분	Name	Type	Visibility
	Description		
Attributes	auth 구글 로그인시 사용하는 변수	FirebaseAuth	private
구분	Name	Argument	Returns
	Description		
Operations	initNaverLogin	none	none
	네이버 로그인과 관련된 클릭 리스너, 이미지 등을 세팅하는 operation		
	initGoogleLogin	none	none
	네이버 로그인과 관련된 리스너 등을 세팅하는 operation		
	onActivityResult	Int, Int, Intent?	none
	구글 로그인 정보를 받은 operation		
	firebaseAuthWithGoogle	String	none
	OnAcvitivResult에서 호출되어 실직적인 로그인을 처리하는 operation.		

UserViewModel

Class Description	사용자의 계정 정보와 이벤트를 관리하는 class		
구분	Name	Type	Visibility
	Description		
Attributes	userLogic 사용자 관련 로직을 저장하는 객체	UserLogic	private
	userRepository 사용자 관련 처리를 통신하기 위한 method를 저장하는 객체	UserRepository	private
	loginEvent 사용자가 로그인 기능을 시작할 때 알리는 이벤트. 결과로 약속 목록 화면으로 전환한다.	SingleLiveEvent<Any>	private
	registerEvent 사용자가 회원가입 기능을 시작할 때 알리는 이벤트. 결과로 회원가입 선택 화면으로 전환한다.	SingleLiveEvent<Any>	private
	registerApiEvent API 회원가입에서 회원가입 기능을 시작할 때 알리는 이벤트, 결과로 회원가입 화면으로 전환한다.	SingleLiveEvent<String>	private
	registerCompleteEvent 회원가입 기능을 마칠 때 알리는 이벤트, 결과로 약속 목록 화면으로 전환한다.	SingleLiveEvent<Any>	private

loginFailEvent	SingleLiveEvent<Any>	private	
사용자가 로그인 기능에 실패할 때 알리는 이벤트, 결과로 에러 메시지를 띠운다.			
googleLoginEvent	LiveMessageEvent<Activity Navigation>	private	
사용자가 구글 API 로그인 기능을 시작할 때 알리는 이벤트. 결과로 구글 로그인 화면으로 전환한다.			
googleRegisterEvent	LiveMessageEvent<Activity Navigation>	private	
사용자가 구글 API 회원가입 기능을 시작 할 때 알리는 이벤트, 결과로 구글 로그인 화면으로 전환한다.			
registerFailEvent	SingleLiveEvent<Any>	private	
사용자가 회원가입 기능을 실패 할 때 알리는 이벤트, 결과로 에러 메시지를 띠운다.			
googleSignInClient	GoogleSignInClient	private	
구글 로그인 클라이언트를 저장하는 객체			
token	String	private	
서버에서 사용자 장치 식별을 하기 위한 토큰을 저장하는 객체			
구분	Name	Argument	Returns
		Description	
Operations	setGoogleClient	GoogleSignInClient	none
	구글 클라이언트를 저장하기 위한 operation이다.		

UserLogic			
Class Description	Fragment에서의 동작과 내부 로직을 관리하는 class		
구분	Name	Type	Visibility
		Description	
Attributes	RC_SIGN_IN	Int	private
	구글 로그인을 위한 요청 코드를 저장하는 객체		
구분	Name	Argument	Returns
		Description	
Operations	checkLogin	OAuthLoginState	none
	로그인을 확인하고 자동 로그인을 하기 위해 호출되는 operation이다.		
	onLoginClick	AccountXML	none
	사용자가 로그인 버튼을 눌렀을 때 호출되는 operation이다.		
	apiLogin	String, Int	none
	API 로그인을 했을 때 호출되는 operation이다.		
	onGoogleLoginClick	none	none
	사용자가 구글 API 로그인 버튼을 눌렀을 때 호출되는 operation이다.		
	onRegisterClick	none	none
	사용자가 회원가입 버튼을 눌렀을 때 호출되는 operation이다.		
	onGoogleRegisterClick	none	none
	사용자가 구글 API 회원가입 버튼을 눌렸을 때 호출되는 operation이다.		
	naverLoginCallback	String	none
	네이버 API 로그인을 위해 호출되는 callback operation이다.		
	naverRegisterCallback	String	none
	네이버 API 회원가입을 위해 호출되는 callback operation이다.		
	onRegisterCompleteClick	String, String, String, String, String	none
	사용자가 회원가입 완료 버튼을 눌렀을 때 호출되는 operation이다.		
	updateUserInfo	String, String?, String?, Int, String	none
	로그인된 사용자 정보를 업데이트하기 위해 호출되는 operation이다.		

UserRepository			
Class Description	외부 서버와의 통신을 위한 method를 관리하는 class		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
Operations	login	String, String, String, (Int, String?) -> Unit	none
		서버에 로그인 요청을 보내고 응답을 받는 operation이다. 사용자의 아이디인 String 타입의 userId, 비밀번호인 String 타입의 password, 토큰인 String 타입의 token, 그리고 onReult 함수를 인자로 받는다. onResult 함수는 로그인 요청의 응답을 처리한다.	
	apiLogin	String, String, String, (Int, String?, String?) -> Unit	none
		서버에 API 로그인 요청을 보내고 응답을 받는 operation이다. 사용자의 API 아이디인 String 타입의 apild, 토큰인 String 타입의 token, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 API 로그인 요청의 응답을 처리한다.	
	register	String, String, String, (Int) -> Unit	none
		서버에 회원가입 요청을 보내고 응답을 받는 operation이다. 사용자의 아이디인 String 타입의 userId, 비밀번호인 String 타입의 password, 닉네임인 String 타입의 nickName, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 회원가입 요청의 응답을 처리한다.	
	apiRegister	String, String, String (Int) -> Unit	none
		서버에 API 회원가입 요청을 보내고 응답을 받는 operation이다. 사용자의 아이디인 String 타입의 userId, 닉네임인 String 타입의 nickName, API 아이디인 String 타입의 apild, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 API 로그인 요청의 응답을 처리한다.	
	getSavedAccount	none	Account
		로컬 DB에 저장된 계정 정보를 가져온다.	
	saveAccount	Account	none
		로컬 DB에 계정 정보를 저장한다. 사용자의 계정인 Account 타입의 account를 인자로 받는다.	
	deleteAccount	none	none
		로컬 DB에 저장된 계정 정보를 삭제한다.	

UserInfo			
Class Description	사용자 정보를 저장하는 object class		
구분	Name	Type	Visibility
Attributes	userId	String	public
	로그인한 사용자 아이디를 저장하는 객체		
	nickName	String	public
	로그인한 사용자의 닉네임을 저장하는 객체		
	isLoggedIn	String	public
	로그인 여부를 나타내는 객체		

로그인 여부를 저장하는 객체			
구분	Name	Argument	Returns
	Description		
Operations	None		

3.6.2 Plan class diagram

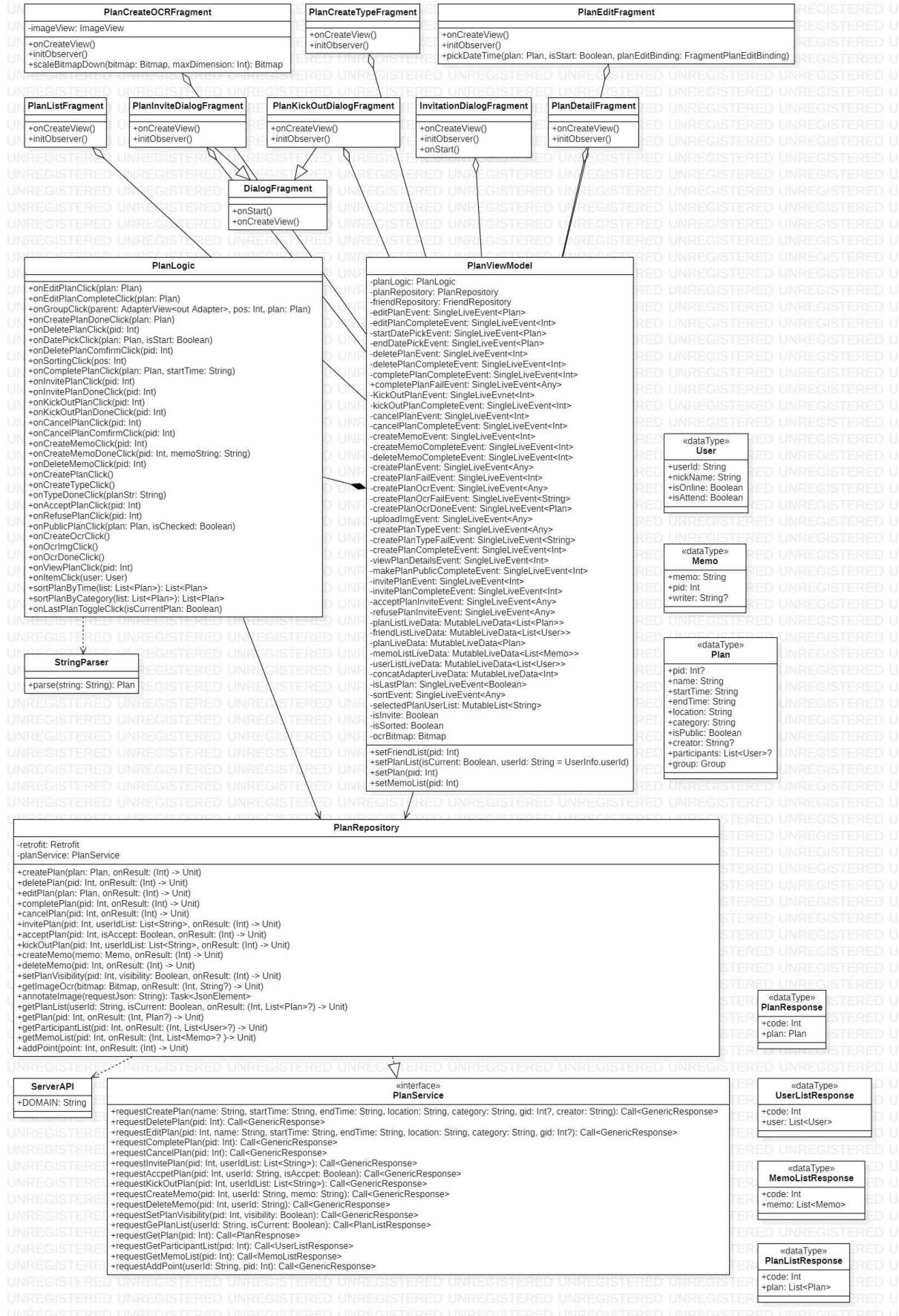


그림 3-9] Plan 기능 클래스 다이어그램

PlanCreateOCRFragment

Class Description	사용자가 OCR을 이용하여 약속을 추가하려 할 때 제공하는 fragment		
구분	Name	Type	Visibility
	Description		
Attributes	imageView 사용자가 등록한 이미지를 화면에 보여주기 위해 사용함	ImageView	private
Operations	scaleBitmapDown 비트맵의 크기를 조정하기 위한 operation이다.		
	Bitmap, Int	Bitmap	

PlanCreateTypeFragment

Class Description	사용자가 문장을 입력하여 약속을 추가하려 할 때 제공하는 fragment		
구분	Name	Type	Visibility
	Description		
Attributes	None		
Operations	None		

PlanEditFragment

Class Description	사용자가 생성된 약속을 생성하거나 수정할 때 제공하는 fragment		
구분	Name	Type	Visibility
	Description		
Attributes	None		
Operations	pickDateTime Plan의 날짜와 시간을 설정하기 위한 operation이다.	Plan, Boolean, FragmentPlanEditBinding	none

PlanListFragment

Class Description	사용자에게 약속 목록을 제공하는 fragment		
구분	Name	Type	Visibility
	Description		
Attributes	None		
Operations	None		

PlanDetailFragment

Class Description	사용자에게 약속의 세부 정보를 제공하는 fragment		
구분	Name	Type	Visibility
			Description
Attributes	None		
구분	Name	Argument	Returns
			Description
Operations	None		

PlanInviteDialogFragment

Class Description	사용자에게 초대할 사용자 목록 dialog를 제공하는 fragment		
구분	Name	Type	Visibility
			Description
Attributes	None		
구분	Name	Argument	Returns
			Description
Operations	None		

PlanKickOutDialogFragment

Class Description	사용자에게 강제 퇴장할 사용자 목록 dialog를 제공하는 fragment		
구분	Name	Type	Visibility
			Description
Attributes	None		
구분	Name	Argument	Returns
			Description
Operations	None		

PlanInvitationFragment

Class Description	약속을 초대받은 사용자에게 약속의 정보를 제공하고 수락 여부를 묻는 fragment		
구분	Name	Type	Visibility
			Description
Attributes	None		
구분	Name	Argument	Returns
			Description
Operations	None		

PlanViewModel

Class Description	약속, 약속 목록, 친구 목록, 메모 목록, 사용자 목록에 대한 정보를 관리하는 class		
구분	Name	Type	Visibility
			Description

Attributes	planLogic	PlanLogic	private
	약속 관련 로직을 저장하는 객체		
	planRepository	PlanRepository	private
	약속 관련 처리를 통신하기 위한 method를 저장하는 객체		
	freindRepository	FriendRepository	private
	친구 관련 처리를 통신하기 위한 method를 저장하는 객체		
	editPlanEvent	SingleLiveEvent<Plan>	private
	사용자가 약속 수정 기능을 시작할 때를 알리는 이벤트. 결과로 화면 전환을 한다.		
	editPlanCompleteEvent	SingleLiveEvent<Int>	private
	사용자가 약속 수정 기능을 완료할 때를 알리는 이벤트. 결과로 경우에 따라 오류 메시지를 출력한다.		
	startDatePickEvent	SingleLiveEvent<Plan>	private
	사용자가 약속 날짜 및 시간 선택 기능을 시작할 때를 알리는 이벤트. 결과로 날짜와 시간 선택 화면을 보여준다.		
	endDatePickEvent	SingleLiveEvent<Plan>	private
	사용자가 약속 날짜 및 시간 선택 기능을 완료할 때를 알리는 이벤트. 결과로 경우에 따라 오류 메시지를 출력한다.		
	deletePlanEvent	SingleLiveEvent<Int>	private
	사용자가 약속 삭제 기능을 시작할 때를 알리는 이벤트. 결과로 경고 메시지를 출력한다.		
	deletePlanCompleteEvent	SingleLiveEvent<Int>	private
	사용자가 약속 삭제 기능을 완료할 때를 알리는 이벤트. 결과로 경우에 따라 오류 메시지를 출력한다.		
	completePlanCompleteEvent	SingleLiveEvent<Int>	private
	사용자가 약속 완료 기능을 완료할 때를 알리는 이벤트.		
	completePlanFailEvent	SingleLiveEvent<Any>	private
	약속 시작 시간이 지금보다 미래일 때 오류를 발생시키는 이벤트. 에러 메시지를 출력한다.		
	kickOutPlanEvent	SingleLiveEvent<Int>	private
	사용자가 강제 퇴장 기능을 시작할 때를 알리는 이벤트. 결과로 화면 전환을 한다.		
	kickOutPlanCompleteEvent	SingleLiveEvent<Int>	private
	사용자가 강제 퇴장 기능을 완료할 때를 알리는 이벤트. 결과로 경우에 따라 오류 메시지를 출력한다.		
	cancelPlanEvent	SingleLiveEvent<Int>	private
	사용자가 약속 취소 기능을 시작할 때를 알리는 이벤트. 결과로 경고 메시지를 출력한다.		
	cancelPlanCompleteEvent	SingleLiveEvent<Int>	private
	사용자가 약속 취소 기능을 완료할 때를 알리는 이벤트. 결과로 경우에 따라 오류 메시지를 출력한다.		
	createMemoEvent	SingleLiveEvent<Int>	private
	사용자가 메모 추가 기능을 시작할 때를 알리는 이벤트. 결과로 화면 전환을 한다.		
	createMemoCompleteEvent	SingleLiveEvent<Int>	private
	사용자가 메모 추가 기능을 완료할 때를 알리는 이벤트. 결과로 경우에 따라 오류 메시지를 출력한다.		
	deleteMemoCompleteEvent	SingleLiveEvent<Int>	private
	사용자가 메모 삭제 기능을 완료할 때를 알리는 이벤트. 결과로 경우에 따라 오류 메시지를 출력한다.		
	createPlanEvent	SingleLiveEvent<Any>	private
	사용자가 약속 추가 기능을 시작할 때를 알리는 이벤트. 결과로 화면 전환을 한다.		
	createPlanFailEvent	SingleLiveEvent<Int>	private

사용자가 약속 추가 기능에 실패할 때를 알리는 이벤트. 결과로 오류 메시지를 출력한다.

createPlanOcrEvent	SingleLiveEvent<Any>	private
---------------------------	----------------------	---------

사용자가 사진으로 약속 추가 기능을 시작할 때를 알리는 이벤트. 결과로 화면 전환을 한다.

createPlanOcrFailEvent	SingleLiveEvent<String>	private
-------------------------------	-------------------------	---------

사용자가 사진으로 약속 추가 기능에 실패할 때를 알리는 이벤트. 결과로 오류 메시지를 출력한다.

createPlanOcrDoneEvent	SingleLiveEvent<Plan>	private
-------------------------------	-----------------------	---------

사용자가 사진으로 약속 추가 기능을 완료할 때를 알리는 이벤트. 결과로 경우에 따라 오류 메시지를 출력한다.

uploadImgEvent	SingleLiveEvent<Any>	private
-----------------------	----------------------	---------

사용자가 시스템에 사진 업로드를 시작할 때를 알리는 이벤트. 결과로 사진을 업로드할 수 있게 한다.

createPlanTypeEvent	SingleLiveEvent<Any>	private
----------------------------	----------------------	---------

사용자가 문장으로 약속 추가 기능을 시작할 때를 알리는 이벤트. 결과로 화면 전환을 한다.

createPlanTypeFailEvent	SingleLiveEvent<String>	private
--------------------------------	-------------------------	---------

사용자가 문장으로 약속 추가 기능에 실패할 때를 알리는 이벤트. 결과로 오류 메시지를 출력한다.

createPlanCompleteEvent	SingleLiveEvent<Int>	private
--------------------------------	----------------------	---------

사용자가 약속 추가 기능을 완료할 때를 알리는 이벤트. 결과로 경우에 따라 오류 메시지를 출력한다.

viewPlanDetailsEvent	SingleLiveEvent<Int>	private
-----------------------------	----------------------	---------

사용자가 약속 상세 내용 보기 기능을 시작할 때를 알리는 이벤트. 결과로 화면 전환을 한다.

makePlanPublicCompleteEvent	SingleLiveEvent<Int>	private
------------------------------------	----------------------	---------

사용자가 약속 공개 범위를 전환할 때를 알리는 이벤트. 결과로 경우에 따라 오류 메시지를 출력한다.

invitePlanEvent	SingleLiveEvent<Int>	private
------------------------	----------------------	---------

사용자가 약속 초대 기능을 시작할 때를 알리는 이벤트. 후에 화면 전환을 한다.

invitePlanCompleteEvent	SingleLiveEvent<Int>	private
--------------------------------	----------------------	---------

사용자가 약속 초대 기능을 완료할 때를 알리는 이벤트. 결과로 경우에 따라 오류 메시지를 출력한다.

acceptPlanInviteEvent	SingleLiveEvent<Any>	private
------------------------------	----------------------	---------

사용자가 약속 수락 기능을 사용할 때를 알리는 이벤트. 결과로 화면을 제거한다.

refusePlanInviteEvent	SingleLiveEvent<Any>	private
------------------------------	----------------------	---------

사용자가 약속 거절 기능을 사용할 때를 알리는 이벤트. 결과로 화면을 제거한다.

planListLiveData	MutableLiveData<List<Plan>>	private
-------------------------	-----------------------------	---------

약속 목록에 대한 정보를 저장하는 객체. 현재/과거 약속 목록 화면에서 약속 목록이다.

friendListLiveData	MutableLiveData<List<User>>	private
---------------------------	-----------------------------	---------

사용자와 친구 관계를 맺은 다른 사용자들의 리스트를 저장하는 객체. 초대, 강제퇴장 목록에서 친구 목록이다.

planLiveData	MutableLiveData<Plan>	private
---------------------	-----------------------	---------

단일 약속에 대한 정보를 저장하는 객체. 현재/과거 약속 보기에서 약속이다.

memoListLiveData	MutableLiveData<List<Memo>>	private
-------------------------	-----------------------------	---------

	메모 목록에 대한 정보를 저장하는 객체. 현재/과거 약속 보기에서 메모 목록이다.		
	<code>userListLiveData</code>	<code>MutableLiveData<List<User>></code>	private
	사용자 목록을 저장하는 객체. 약속 참여자 리스트 등을 저장한다.		
	<code>concatAdapterLiveData</code>	<code>MutableLiveData<Int></code>	private
	메모 목록과 사용자 목록을 관리하는 객체. 메모와 사용자 목록이 변경되었는지 여부를 파악한다.		
	<code>isLastPlan</code>	<code>SingleLiveEvent<Boolean></code>	private
	사용자가 현재/과거 약속 목록을 변경할 때를 알리는 이벤트. 결과로 갱신된 약속 목록 화면을 보여준다.		
	<code>sortEvent</code>	<code>SingleLiveEvent<Any></code>	private
	사용자가 약속 목록의 정렬 방식을 변경할 때를 알리는 이벤트. 결과로 정렬된 약속 목록 화면을 보여준다.		
	<code>selectedPlanUserList</code>	<code>MutableList<String></code>	private
	선택된 사용자 목록을 저장하는 객체. 약속 초대, 강제 퇴장에서 대상 사용자 목록을 저장한다.		
	<code>isInvite</code>	<code>Boolean</code>	private
	사용자가 친구 초대와 강제 퇴장 버튼 중 어떤 버튼을 선택했는지 확인하는 객체		
	<code>isSorted</code>	<code>Boolean</code>	private
	약속 목록의 정렬 여부를 저장하는 객체		
	<code>ocrBitmap</code>	<code>Bitmap</code>	private
	OCR을 사용한 약속 생성을 위해 사진의 비트맵을 저장하는 객체		
구분	Name	Argument	Returns
	Description		
Operations	<code>setPlanList</code>	<code>Boolean, String</code>	none
	planListLiveData에 대한 setter이다.		
	<code>setPlan</code>	<code>Int</code>	none
	planLiveData에 대한 setter이다. 개별 약속을 식별하기 위한 Int 타입의 pid 변수를 인자로 받는다.		
	<code>setFriendList</code>	<code>Int</code>	none
	friendListLiveData와 userListLiveData에 대한 setter이다.		
	<code>setMemoList</code>	<code>Int</code>	none
	userList에 대한 setter이다.		

PlanRepository			
Class Description	외부 서버와의 통신을 위한 method를 관리하는 class		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
Operations	Description		
	<code>editPlan</code>	<code>Plan, (Int) -> Unit</code>	none
	약속 수정 내용을 DB 서버에 저장할 때 호출된다. 약속을 식별하기 위한 Plan 타입의 약속 객체, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 약속 수정 요청의 응답을 처리한다.		
	<code>deletePlan</code>	<code>Int, (Int) -> Unit</code>	none
	삭제할 약속의 식별 정보를 DB 서버에 전달할 때 호출된다. 사용자와 약속을 식별하기 위한 Int 타입의 pid와 onResult 함수를 인자로 받는다. onResult 함수는 약속 삭제 요청 응답을 처리한다.		

정의 응답을 처리한다.

completePlan	Int, (Int) -> Unit	none
완료할 약속의 식별 정보를 DB 서버에 전달할 때 호출한다. 약속을 식별하기 위한 Int 타입의 pid와 onResult 함수를 인자로 받는다. onResult 함수는 약속 완료 요청의 응답을 처리한다.		
invitePlan	Int, List<String>, (Int) -> Unit	none
약속에 사용자를 초대할 때 해당 약속과 사용자의 식별 정보를 DB 서버에 전달할 때 호출한다. 약속을 식별하기 위한 Int 타입의 pid와 초대받을 사용자 아이디 리스트인 List<String> 타입의 userIdList, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 약속 초대 요청의 응답을 처리한다.		
kickOutPlan	Int, List<String>, (Int) -> Unit	none
약속에 사용자를 강제 퇴장할 때 해당 약속과 사용자의 식별 정보를 DB 서버에 전달할 때 호출한다. 사용자와 약속을 식별하기 위한 UInt 타입의 pid와 강제 퇴장당할 사용자 아이디 리스트인 List<String> 타입의 userIdList, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 강제 퇴장 요청의 응답을 처리한다.		
cancelPlan	Int, (Int) -> Unit	none
취소할 약속의 식별 정보를 DB 서버에 전달할 때 호출한다. 약속을 식별하기 위한 Int 타입의 pid와 onResult 함수를 인자로 받는다. onResult 함수는 약속 취소 요청의 응답을 처리한다.		
acceptPlan	Int, Boolean, (Int) -> Unit	none
수락이나 거절할 약속의 식별 정보를 DB 서버에 전달할 때 호출한다. 약속을 식별하기 위한 Int 타입의 pid, 수락여부를 위한 Boolean 타입의 isAccept, onResult 함수를 인자로 받는다. onResult 함수는 약속 수락 또는 거절 요청의 응답을 처리한다.		
createMemo	Memo, (Int) -> Unit	none
메모를 생성할 때 연결된 약속의 식별 정보와 메모 내용을 DB 서버에 저장할 때 호출한다. 약속을 식별하기 위한 Int 타입의 pid와 메모 내용이 포함 된 Memo 타입의 memo 변수, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 메모 생성 요청의 응답을 처리한다.		
deleteMemo	Int, (Int) -> Unit	none
메모를 삭제할 때 연결된 약속의 식별 정보를 DB 서버에 전달할 때 호출한다. 약속을 식별하기 위한 Int 타입의 pid와 onResult 함수를 인자로 받는다. onResult 함수는 메모 삭제 요청의 응답을 처리한다.		
createPlan	Plan, (Int) -> Unit	none
약속을 생성할 때 약속에 대한 정보를 DB 서버에 저장할 때 호출한다. 약속 정보인 Plan 객체, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 약속 생성 요청의 응답을 처리한다.		
setPlanVisibility	Int, Boolean, (Int) -> Unit	none
약속을 공개 또는 비공개 약속으로 전환하는 것을 DB 서버에 알릴 때 호출한다. 약속을 식별하기 위한 Int 타입의 pid와 onResult 함수를 인자로 받는다. onResult 함수는 약속 공개 및 비공개 전환 요청의 응답을 처리한다.		
getImageOcr	Bitmap, (Int, String?) -> Unit	none
이미지 비트맵을 OCR API로 전송해 결과 문장을 받아오고자 할 때 호출한다. 이미지의 비트맵인 Bitmap 타입의 bitmap와 onResult 함수를 인자로 받는다. onResult 함수는		

	OCR 판독 요청의 응답을 처리한다.	
annotateImage	String	Task<JsonElement>
getImageOcr	함수에서 OCR API 요청을 만들기 위해 호출한다. OCR 요청인 String 타입의 requestJson를 인자로 받고 Task<JsonElement> 타입의 만든 Task를 반환한다.	
getPlanList	String, Boolean, (Int, List<Plan>) -> Unit	none
사용자의 약속 목록을 DB 서버에서 받아올 때 호출한다. 사용자를 식별하기 위한 String 형의 userId, 지난 약속을 받아올지 판단하는 Boolean 타입의 isCurrent, 그리고, onResult 함수를 인자로 받는다. onResult 함수는 약속 목록 요청의 응답을 처리한다.		
getPlan	Int, (Plan) -> Unit, (Int, Plan?) -> Unit	none
단일 약속의 세부 내용을 DB 서버에서 받아올 때 호출한다. 약속을 식별하기 위한 Int 타입의 pid와 onResult 함수를 인자로 받는다. onResult 함수는 약속 세부 정보 요청의 응답을 처리한다.		
getMemoList	Int, (Int, List<Memo>?) -> Unit	none
해당 약속의 메모 목록을 DB 서버에서 받아올 때 호출한다. 약속을 식별하기 위한 Int 타입의 pid와 onResult 함수를 인자로 받는다. onResult 함수는 메모 목록 요청의 응답을 처리한다.		
getParticipantList	Int, (Int, List<User>?) -> Unit	none
해당 약속에 참석 되어있는 사용자들의 목록을 DB 서버에서 받아올 때 호출한다. 약속을 식별하기 위한 Int 타입의 pid와 onResult 함수를 인자로 받는다. onResult 함수는 참석자 목록 요청의 응답을 처리한다.		
addPoint	Int, (Int) -> Unit	none
사용자의 포인트 증가를 DB 서버에 요청할 때 호출한다. 추가하고자 하는 포인트의 양을 위한 Int 타입의 point와 onResult 함수를 인자로 받는다. onResult 함수는 포인트 증가 요청의 응답을 처리한다.		

PlanLogic			
Class Description	Fragment에서의 동작과 내부 로직을 관리하는 class		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
Operations	onEditPlanClick	Plan	none
	사용자가 약속 수정 버튼을 클릭할 때 호출되는 operation이다.		
	onEditPlanCompleteClick	Plan	none
	사용자가 약속 수정을 완료하고, 완료 버튼을 클릭할 때 호출되는 operation이다.		
	onGroupClick	AdapterView<out Adapter, Int, Plan>	none
	사용자가 약속 생성 또는 수정 화면에서 그룹을 클릭할 때 호출되는 operation이다.		

onDeletePlanClick	Int	none
사용자가 약속 삭제 버튼을 클릭할 때 호출되는 operation이다.		
onDeletePlanConfirmClick	Int	none
사용자가 약속 삭제 경고 창에서 확인 버튼을 클릭할 때 호출되는 operation이다.		
onDatePickerClick	Plan, Boolean	none
사용자가 약속 생성 또는 수정 화면에서 시작일 또는 종료일 버튼을 클릭할 때 호출되는 operation이다.		
onSortingClick	Int	none
사용자가 약속 목록 화면에서 정렬을 선택할 때 호출되는 operation이다.		
onCompletePlanClick	Plan, String	none
사용자가 약속 완료 버튼을 클릭할 때 호출되는 operation이다. 시작시간 문자열을 입력받아 약속이 시작 전인지 확인하고 처리한다.		
onInvitePlanClick	Int	none
사용자가 약속 초대 버튼을 클릭할 때 호출되는 operation이다.		
onInvitePlanDoneClick	Int	none
사용자가 약속에 초대할 사용자를 선택하고 나서 완료 버튼을 클릭할 때 호출되는 operation이다.		
onKickOutPlanClick	Int	none
사용자가 현재 약속에서 강제 퇴장 버튼을 클릭할 때 호출되는 operation이다.		
onKickOutPlanDoneClick	Int	none
사용자가 현재 약속에서 강제 퇴장할 사용자를 선택하고 나서 완료 버튼을 클릭할 때 호출되는 operation이다.		
onCancelPlanClick	Int	none
사용자가 현재 약속에서 약속 취소 버튼을 클릭할 때 호출되는 operation이다.		
onCancelPlanConfirmClick	Int	none
사용자가 현재 취소 경고 창에서 확인 버튼을 클릭할 때 호출되는 operation이다.		
onCreateMemoClick	Int	none
사용자가 현재 약속에서 메모 생성 버튼을 클릭할 때 호출되는 operation이다.		
onCreateMemoDoneClick	Int, String	none
사용자가 메모 생성 창에서 메모 내용을 입력하고 완료 버튼을 클릭할 때 호출되는 operation이다. String 타입의 현재 입력한 메모 내용인 memoString 변수를 인자로 받는다.		
onDeleteMemoClick	none	none
사용자가 메모 삭제 버튼을 클릭할 때 호출되는 operation이다.		
onCreatePlanClick	none	none
사용자가 약속 생성 버튼을 클릭할 때 호출되는 operation이다.		
onCreatePlanDoneClick	Plan	none
사용자가 약속 생성 창에서 약속 정보를 입력하고 완료 버튼을 클릭할 때 호출되는 operation이다. Plan 타입의 약속 정보를 담은 객체인 plan을 인자로 받는다.		
onCreateTypeClick	none	none
사용자가 문장으로 약속 생성 버튼을 클릭할 때 호출되는 operation이다.		
onTypeDoneClick	String	none
사용자가 문장으로 약속 생성 창에서 약속 정보를 문장으로 입력하고 완료 버튼을 클릭할 때 호출되는 operation이다.		
onAcceptPlanClick	Int	none
사용자가 약속 수락 화면에서 수락 버튼을 클릭할 때 호출되는 operation이다.		
onRefusePlanClick	Int	none
사용자가 약속 수락 화면에서 거절 버튼을 클릭할 때 호출되는 operation이다.		
onPublicPlanClick	Plan, Boolean	none
사용자가 약속 공개/비공개 버튼을 클릭할 때 호출되는 operation이다.		
onCreateOcrClick	none	none

사용자가 사진으로 약속 생성 버튼을 클릭할 때 호출되는 operation이다.

onOcrImgClick	none	none
사용자가 사진으로 약속 생성 창에서 사진 등록 버튼을 클릭할 때 호출되는 operation이다.		

onOcrDoneClick	none	none
사용자가 사진으로 약속 생성 창에서 사진을 등록하고 완료 버튼을 클릭할 때 호출되는 operation이다.		

onViewPlanClick	Int	none
사용자가 약속 목록 화면에서 약속 객체를 클릭할 때 약속의 상세 정보를 보여주기 위해 호출되는 operation이다.		

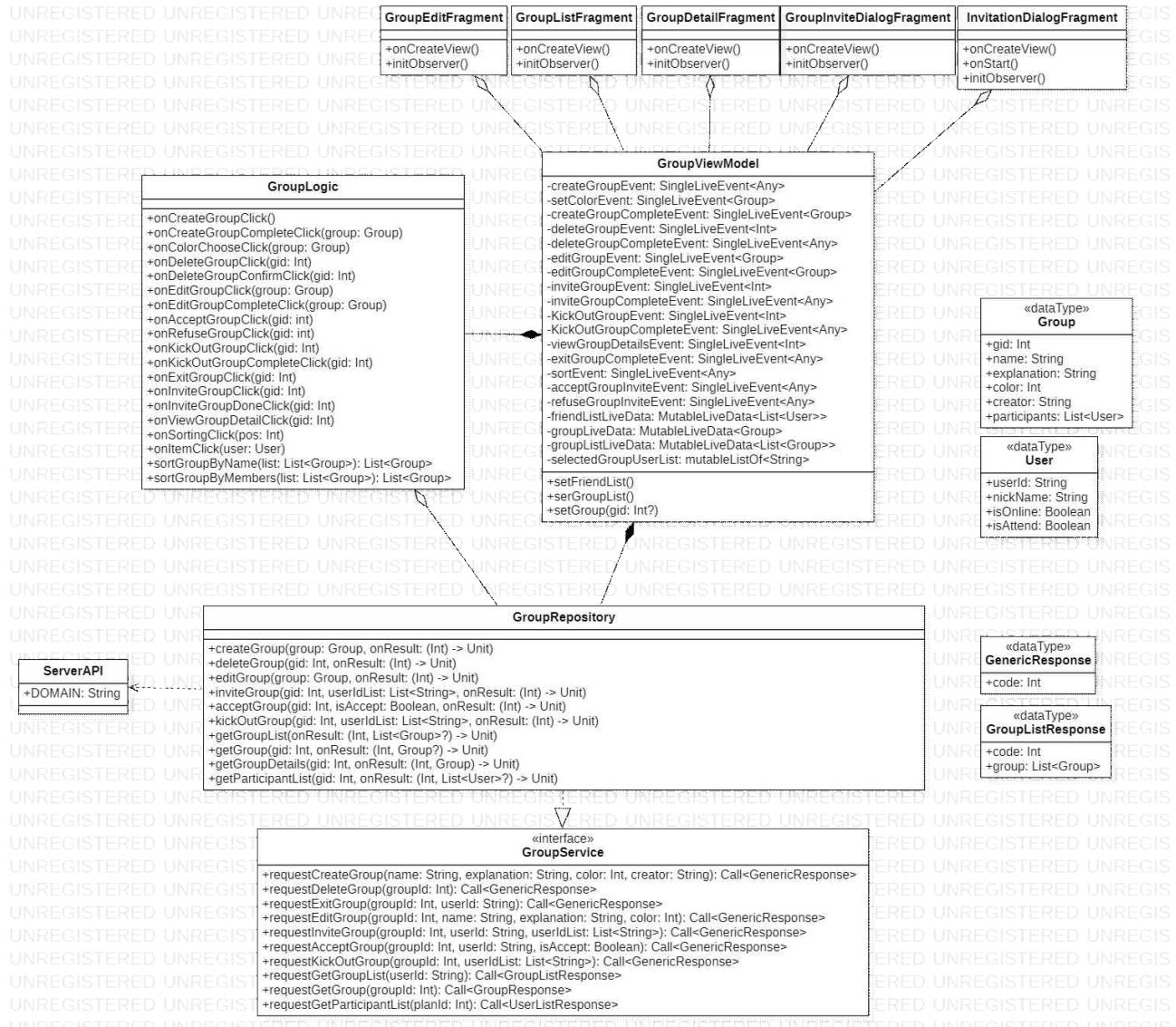
onItemClick	User	none
사용자가 약속 초대 또는 강제 퇴장 화면에서 대상을 선택할 때 호출되는 operation이다.		

sortPlanByTime	List<Plan>	none
사용자의 약속 목록들을 시간 순서대로 정렬하기 위해 호출되는 operation이다.		

sortPlanByCategory	List<Plan>	none
사용자의 약속 목록들을 카테고리 순서대로 정렬하기 위해 호출되는 operation이다.		

onLastPlanToggleClick	Boolean	none
사용자가 이전 약속 보기 버튼을 클릭할 때 호출되는 operation이다.		

3.6.3 Group class diagram



[그림 3-10] Group CD

GroupEditFragment			
Class	그룹 정보를 수정하는 fragment		
Description			
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
Operations	None		

GroupInviteDialogFragment				
Class Description	그룹을 초대하는 fragment			
구분	Name	Type	Visibility	
	Description			
Attributes	None			
구분	Name	Argument	Returns	
	Description			
Operations	None			

GroupDetailFragment				
Class Description	그룹 상세정보를 표시하는 fragment			
구분	Name	Type	Visibility	
	Description			
Attributes	None			
구분	Name	Argument	Returns	
	Description			
Operations	None			

GroupListFragment				
Class Description	그룹 목록을 표시하는 fragment			
구분	Name	Type	Visibility	
	Description			
Attributes	None			
구분	Name	Argument	Returns	
	Description			
Operations	None			

InvitationDialogFragment				
Class Description	초대장을 표시하는 fragment			
구분	Name	Type	Visibility	
	Description			
Attributes	None			
구분	Name	Argument	Returns	
	Description			
Operations	onStart	none	none	
	Fragment가 생성될 때 시작되는 Operation이다.			

GroupViewModel				
Class Description	그룹의 생성, 삭제, 수정, 초대, 차단, 목록 보기, 상세정보 보기, 그룹데이터처리를 하는 class			

구분	Name	Type	Visibility
	Description		
Attributes	<code>createGroupEvent</code>	<code>SingleLiveEvent<Int></code>	private
	사용자가 그룹 생성 버튼을 눌렀을 때 호출하는 이벤트		
	<code>setColorEvent</code>	<code>SingleLiveEvent<Group></code>	private
	사용자가 그룹의 색상을 선택할 때 호출하는 이벤트		
	<code>createGroupCompleteEvent</code>	<code>SingleLiveEvent<Group></code>	private
	사용자가 그룹 생성 완료 버튼을 눌렀을 때 호출하는 이벤트		
	<code>deleteGroupEvent</code>	<code>SingleLiveEvent<Int></code>	private
	사용자가 그룹 삭제 버튼을 눌렀을 때 호출하는 이벤트		
	<code>deleteGroupCompleteEvent</code>	<code>SingleLiveEvent<Any></code>	private
	사용자가 그룹 삭제 완료 버튼을 눌렀을 때 호출하는 이벤트		
	<code>editGroupEvent</code>	<code>SingleLiveEvent<Group></code>	private
	사용자가 그룹 정보 수정 버튼을 눌렀을 때 호출하는 이벤트		
	<code>editGroupCompleteEvent</code>	<code>SingleLiveEvent<Group></code>	private
	사용자가 그룹 정보 수정 완료버튼을 눌렀을 때 호출하는 이벤트		
	<code>inviteGroupEvent</code>	<code>SingleLiveEvent<Int></code>	private
	사용자가 그룹 초대 버튼을 눌렀을 때 호출하는 이벤트		
	<code>inviteGroupCompleteEvent</code>	<code>SingleLiveEvent<Any></code>	private
	사용자가 그룹 초대 완료 버튼을 눌렀을 때 호출하는 이벤트		
	<code>KickOutGroupEvent</code>	<code>SingleLiveEvent<Int></code>	private
	사용자가 그룹 차단 버튼을 눌렀을 때 호출하는 이벤트		
	<code>KickOutGroupCompleteEvent</code>	<code>SingleLiveEvent<Any></code>	private
	사용자가 그룹 차단 완료 버튼을 눌렀을 때 호출하는 이벤트		
	<code>viewGroupDetailsEvent</code>	<code>SingleLiveEvent<Int></code>	private
	사용자가 그룹 상세정보를 눌렀을 때 호출하는 이벤트		
	<code>exitGroupCompleteEvent</code>	<code>SingleLiveEvent<Any></code>	private
	사용자가 그룹 탈퇴 확인 버튼을 눌렀을 때 호출하는 이벤트		
Operations	<code>sortEvent</code>	<code>SingleLiveEvent<Any></code>	private
	사용자가 그룹 목록에서 정렬 기준 버튼을 눌렀을 때 호출하는 이벤트		
	<code>acceptGroupInviteEvent</code>	<code>SingleLiveEvent<Any></code>	private
	사용자가 그룹 초대를 수락했을 때 호출하는 이벤트		
	<code>refuseGroupInviteEvent</code>	<code>SingleLiveEvent<Any></code>	private
	사용자가 그룹 초대를 거절했을 때 호출하는 이벤트		
	<code>friendListLiveData</code>	<code>MutableLiveData<List<User>></code>	private
	사용자의 친구 목록을 저장하는 객체		
	<code>groupLiveData</code>	<code>MutableLiveData<Group></code>	private
	사용자가 속한 단일 그룹 정보를 저장하는 객체		
구분	<code>groupListLiveData</code>	<code>MutableLiveData<List<Group>></code>	private
	사용자가 속한 그룹들의 정보를 저장하는 객체		
	<code>SelectedGroupUserList</code>	<code>mutableListOf<String></code>	public
	그룹에 속한 사용자들 중 현재 사용자가 목록에서 선택한 사용자들을 저장하는 객체		
Operations	Name	Argument	Returns
	Description		
	<code>setFriendList</code>	none	none
	friendListLiveData에 대한 setter이다.		
	<code>setGroupList</code>	none	none
	groupListLiveData에 대한 setter이다.		
	<code>setGroup</code>	Int	none
	groupLiveData에 대한 setter이다. 인자로 Int 타입의 그룹 구분자인 gid를 받아 서버		

에 해당 gid와 일치하는 그룹의 정보를 요청하여 groupLiveData에 저장한다.

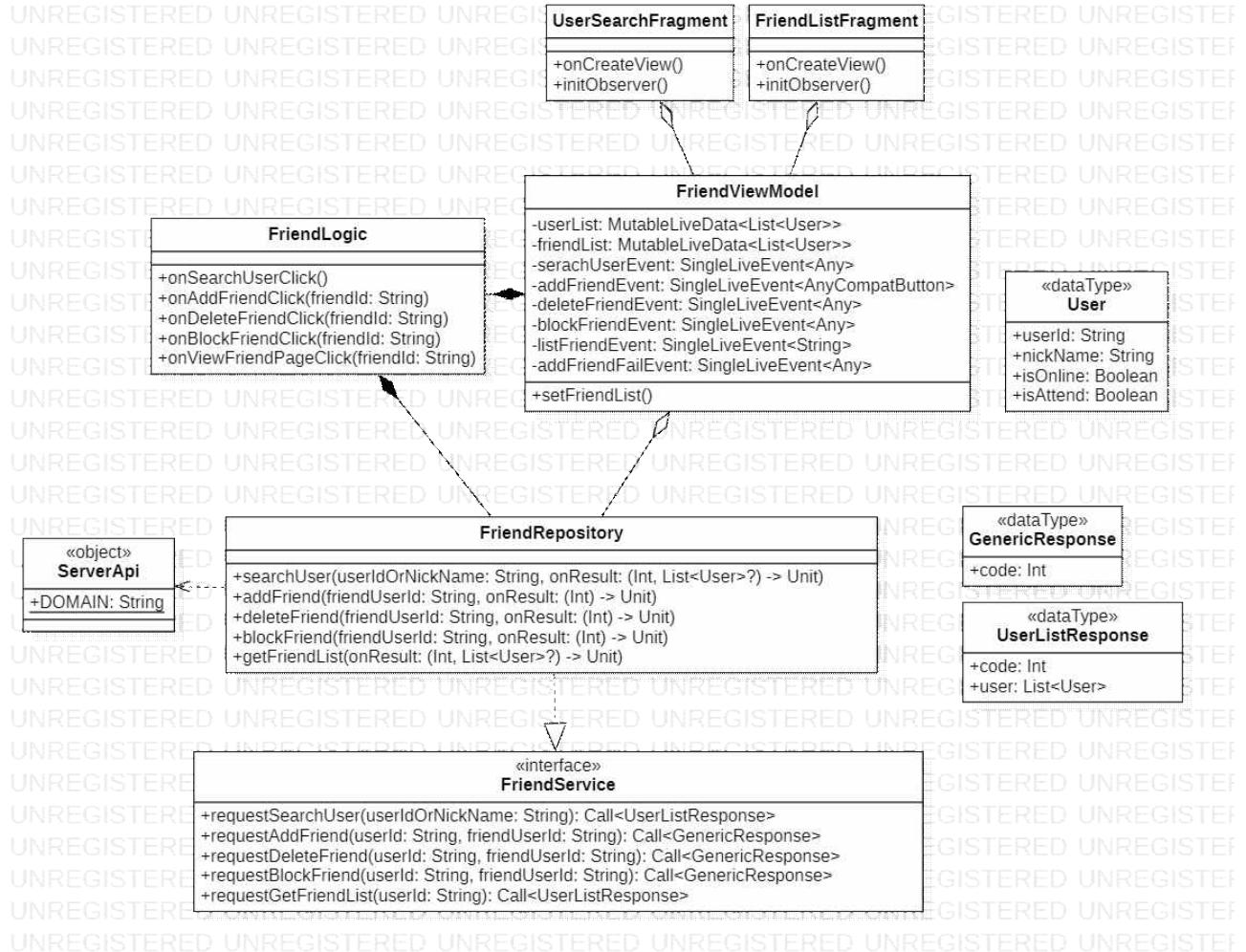
GroupLogic			
Class Description	Group과 관련된 method를 관리하는 class		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
		Description	
Operations	onCreateGroupClick	none	none
	사용자가 그룹 생성 버튼을 눌렀을 때 호출되는 operation이다.		
	onCreateGroupCompleteClick	Group	none
	사용자가 그룹 생성을 완료하고, 완료 버튼을 클릭할 때 호출되는 operation이다. 사용자가 설정한 그룹의 세부 정보를 Group 타입의 객체로 받아 groupRepository를 통해 서버에 그룹을 생성할 것을 요청한다.		
	onColorChooseClick	Group	none
	사용자가 그룹의 색상을 선택했을 때 호출되는 operation이다. 사용자가 설정한 그룹의 색상 정보를 Group 타입의 객체로 받는다.		
	onDeleteGroupClick	Int	none
	사용자가 그룹 삭제 버튼을 눌렀을 때 호출되는 operation이다. 사용자가 선택한 그룹의 구분자를 Int 타입의 객체로 받는다.		
	onDeleteGroupConfirmClick	Int	none
	사용자가 그룹 삭제를 완료하고, 완료 버튼을 클릭할 때 호출되는 operation이다. 사용자가 선택한 그룹의 구분자를 Int 타입의 객체로 받아 groupRepository를 통해 서버에서 해당 그룹을 삭제할 것을 요청한다.		
	onEditGroupClick	Group	none
	사용자가 그룹 정보 수정 버튼을 눌렀을 때 호출되는 operation이다. 사용자가 재설정한 그룹의 세부 정보를 Group 타입의 객체로 받는다.		
	onEditGroupCompleteClick	Group	none
	사용자가 그룹 정보 수정을 완료하고, 완료 버튼을 클릭할 때 호출되는 operation이다. 사용자가 재설정한 그룹의 세부 정보를 Group 타입의 객체로 받아 groupRepository를 통해 서버에서 해당 그룹의 정보를 수정할 것을 요청한다.		
Operations	onAcceptGroupClick	Int	none
	사용자가 그룹 초대 수락 버튼을 눌렀을 때 호출되는 operation이다. 어떤 그룹에서 초대받았는지 알기 위한 그룹의 구분자 gid를 받아 groupRepository를 통해 서버에 사용자의 초대 수락을 요청한다.		
	onRefuseGroupClick	Int	none
	사용자가 그룹 초대 거절 버튼을 눌렀을 때 호출되는 operation이다. 어떤 그룹에서 초대받았는지 알기 위한 그룹의 구분자 gid를 받아 groupRepository를 통해 서버에 사용자의 초대 거절을 요청한다.		
	onKickOutGroupClick	Int	none
	사용자가 그룹 강퇴 버튼을 눌렀을 때 호출되는 operation이다.		
	onExitGroupClick	Int	none
사용자가 그룹 탈퇴 버튼을 눌렀을 때 호출되는 operation이다. 어떤 그룹에서 탈퇴하는지 알기 위한 그룹의 구분자 gid를 받아 groupRepository를 통해 서버에 사용자의 그룹 탈퇴를 요청한다.			
onInviteGroupClick			
Int			
none			

사용자가 그룹 초대 버튼을 눌렀을 때 호출되는 operation이다.		
onInviteGroupDoneClick	Int	none
사용자가 그룹에 초대할 사용자를 선택하고 완료 버튼을 눌렀을 때 호출되는 operation이다.		
onViewGroupDetailClick	Int	none
사용자가 그룹 상세정보를 보기위해 해당 그룹을 눌렀을 때 호출되는 operation이다. 어떤 그룹인지 알지 위해 그룹의 구분자 gid를 받아 groupRepository를 통해 서버에 해당 그룹의 상세 정보를 요청한다.		
onSortingClick	Int	none
사용자가 선택한 그룹 목록의 정렬 기준에 따라 groupListLiveData의 목록을 정렬한다. 정렬 기준을 알기 위해 Int 타입의 pos 변수를 입력받는다.		
onItemClick	User	none
(추정)사용자가 그룹의 참여자 목록에서 사용자를 선택할 때 호출되는 operation이다.		
sortGroupByName	List<Group>	List<Group>
Group 객체의 목록을 인자로 받아 그룹 이름을 기준으로 정렬하여 정렬된 Group 객체의 목록을 return하는 operation이다.		
sortGroupByMembers	List<Group>	List<Group>
Group 객체의 목록을 인자로 받아 그룹의 참여자 수를 기준으로 정렬하여 정렬된 Group 객체의 목록을 return하는 operation이다.		

GroupRepository			
Class	외부 서버와의 통신을 위한 method를 관리하는 class		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
Operations	createGroup	Group, (Int) -> Unit	none
	생성된 그룹 정보를 서버에 저장할 때 호출하는 operation이다. 그룹의 정보가 담겨있는 Group과 onResult 함수를 인자로 받는다. onResult 함수는 그룹 생성 요청의 응답을 처리한다.		
	deleteGroup	Int, (Int) -> Unit	none
	삭제할 그룹의 식별 정보를 서버에 전달할 때 호출하는 operation이다. 그룹을 삭제하기 위한 Int 타입의 gid와 onResult 함수를 인자로 받는다. onResult 함수는 그룹 삭제 요청의 응답을 처리한다.		
	exitGroup	Int, (Int) -> Unit	none
	탈퇴할 그룹의 식별 정보를 서버에 전달할 때 호출하는 operation이다. 그룹을 삭제하기 위한 Int 타입의 gid와 onResult 함수를 인자로 받는다. onResult 함수는 그룹 탈퇴 요청의 응답을 처리한다.		
	editGroup	Group, (Int) -> Unit	none
	그룹의 세부정보 수정 내용을 서버에 저장할 때 호출하는 operation이다. 그룹을 삭제하기 위한 Int 타입의 gid와 그룹의 정보가 담겨있는 Group 타입의 객체, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 그룹 수정 요청의 응답을 처리한다.		
	inviteGroup	Int, List<String>, (Int) -> Unit	none
	그룹에 사용자를 초대할 때 해당 그룹과 사용자의 userId를 서버에 전달할 때 호출하는		

operation이다. 그룹을 식별하기 위한 gid와 초대받을 사용자들의 userId, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 그룹 초대 요청의 응답을 처리한다.	acceptGroup	Int, Boolean, (Int) -> Unit	none
사용자가 그룹 초대를 수락 또는 거절할 때 호출되는 operation이다. 그룹을 식별하기 위한 Int 타입의 gid와 초대/거절 여부를 판단하는 Boolean 타입의 isAccept 변수, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 그룹 초대 수락/거부 요청의 응답을 처리한다.	kickOutGroup	Int, List<String>, int	none
그룹에서 사용자를 강제 퇴출할 때 해당 그룹과 사용자의 식별 정보를 DB 서버에 전달할 때 호출하는 operation이다. 사용자와 약속을 식별하기 위한 gid와 강제 퇴출 당할 사용자들의 userId 목록을 위한 List<String> 타입의 userIdList, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 강제 퇴출 요청의 응답을 처리한다.	getGroupList	Int, (Int, List<Group>) -> Unit	none
현재 사용자의 그룹 목록을 서버에서 받아올 때 호출하는 operation이다. 그룹을 식별하기 위한 gid, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 그룹 목록 요청의 응답을 처리한다.	getGroupDetails	Int, (Int, List<Group>) -> Unit	none
현재 그룹의 목록을 서버에서 받아올 때 호출하는 operation이다. 그룹을 식별하기 위한 Int 형의 gid, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 그룹 목록 요청의 응답을 처리한다.	getParticipantList	Int, (Int, List<User>) -> Unit	none
현재 그룹에 속한 사용자 목록을 서버에서 받아올 때 호출하는 operation이다. 그룹을 식별하기 위한 Int 형의 gid, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 그룹 목록 요청의 응답을 처리한다.			

3.6.4 Friend class diagram



[그림 3-11] Friend CD

UserSearchFragment				
Class	사용자를 검색할 때 제공하는 class			
구분	Name	Type	Visibility	
Attributes	None			
구분	Name	Argument	Returns	
Operations	None			

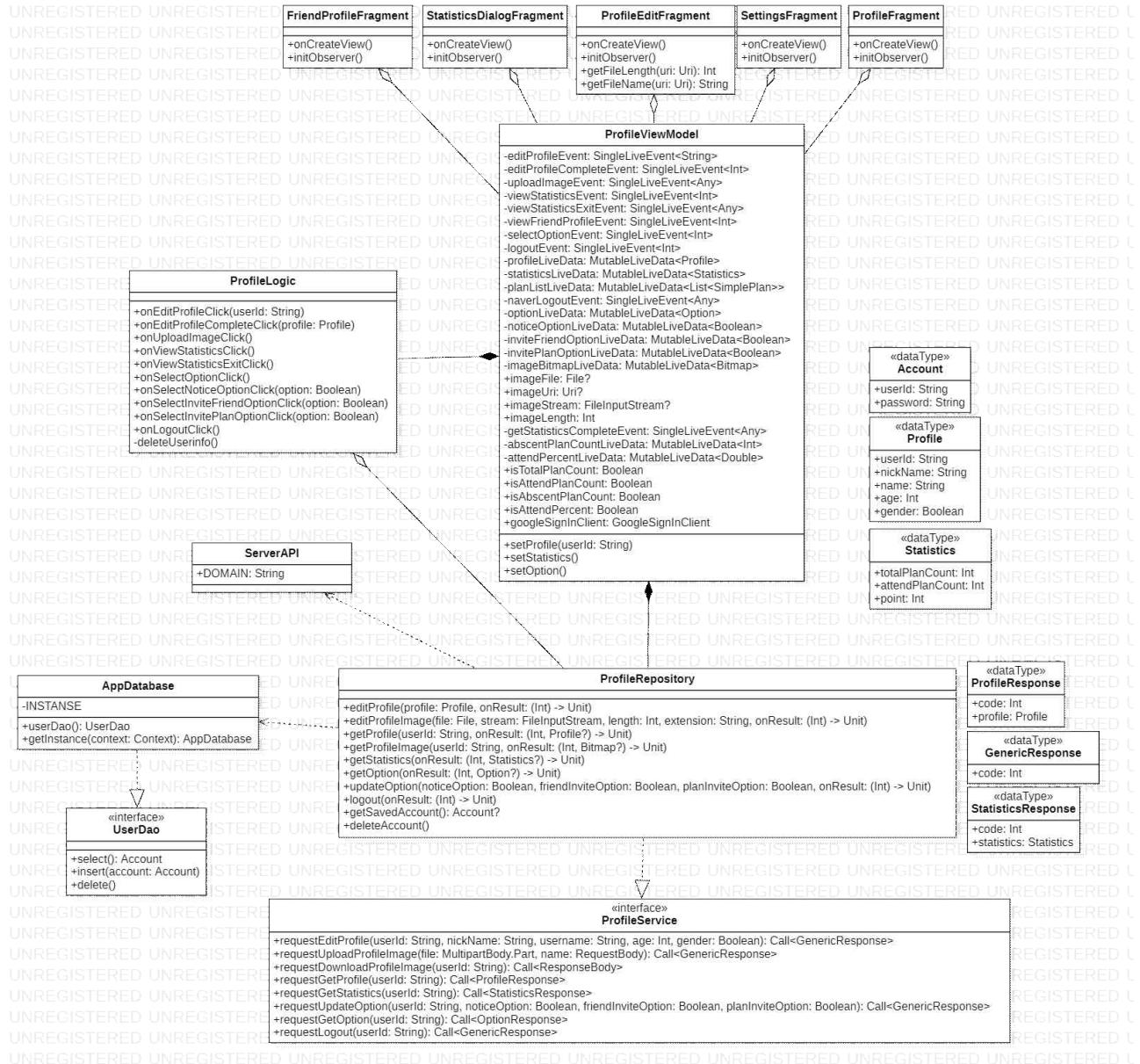
FriendListFragment				
Class Description	친구 목록을 확인할 때 제공하는 class			
구분	Name	Type	Visibility	
	Description			
Attributes	None			
구분	Name	Argument	Returns	
	Description			
Operations	None			

FriendViewModel			
Class Description	친구 정보를 관리하는 class		
구분	Name	Type	Visibility
	Description		
Attributes	userList	MutableLiveData<List<User>>	private
	사용자 검색에서 검색된 사용자 목록을 저장하는 객체		
	friendList	MutableLiveData<List<User>>	private
	친구 목록 보기에서 친구 목록을 저장하는 객체		
	searchUserEvent	SingleLiveEvent<Any>	private
	사용자가 사용자 검색 기능을 시작할 때 알리는 이벤트. 결과로 검색한 사용자 목록을 출력한다.		
	addFriendEvent	SingleLiveEvent<Any> atButton	private
	사용자가 친구 추가 기능을 시작할 때 알리는 이벤트		
	deleteFriendEvent	SingleLiveEvent<Any>	private
	사용자가 친구 삭제 기능을 시작할 때 알리는 이벤트		
Operations	blockFriendEvent	SingleLiveEvent<Any>	private
	사용자가 친구 차단 기능을 시작할 때 알리는 이벤트		
	listFriendEvent	SingleLiveEvent<String>	private
	사용자가 친구 목록 보기 기능을 시작할 때 알리는 이벤트. 결과로 친구 목록을 출력한다.		
	addFriendFailEvent	SingleLiveEvent<Any>	private
구분	사용자가 친구 추가 기능 중 취소했을 때 알리는 이벤트		
	Name	Argument	Returns
Operations	setFriendList	none	none
	FriendRepository에서 친구 목록을 받아와 FriendList에 저장한다.		

FriendLogic			
Class Description	Fragment에서의 동작과 내부 로직을 관리하는 class		
구분	Name	Type	Visibility
	Description		
Attributes	None		
구분	Name	Argument	Returns
	Description		
Operations	onSearchUserClick	none	none
	사용자가 사용자 검색 기능에 진입했을 때 호출되는 operation이다.		
	onSearchUserCompleteClick	String	none
	사용자가 사용자 검색 버튼을 눌렀을 때 호출되는 operation이다. 호출되면 입력한 String 타입의 name을 가지고 FriendRepository를 통해 서버에 사용자를 검색할 것을 요청한다. 그리고 그 결과를 userList에 저장한다.		
	onAddFriendClick	String	none
	사용자가 친구 추가 버튼을 눌렀을 때 호출되는 operation이다. 호출되면 FriendRepository를 통해 서버에 친구 추가를 요청한다. 친구를 추가하고자 하는 사용자를 식별하기 위한 String 타입의 friendId를 인자로 받는다.		
	onDeleteFriendClick	String	none
사용자가 삭제할 친구를 선택 후 삭제 버튼을 눌렀을 때 호출되는 operation이다. 호출되면 FriendRepository를 통해 서버에 친구 삭제를 요청한다. 삭제하고자 하는 친구를 식별하기 위한 String 타입의 friendId를 인자로 받는다.			
onBlockFriendClick			
사용자가 삭제할 친구를 선택 후 차단 버튼을 눌렸을 때 호출되는 operation이다. 호출되면 FriendRepository를 통해 서버에 친구를 차단한다. 차단하고자 하는 친구를 식별하기 위한 String 타입의 friendId를 인자로 받는다.			
onViewFriendPageClick			
사용자가 다른 사용자를 눌렸을 때 호출되는 operation이다. 호출되면 FriendRepository를 통해 서버에 친구의 마이페이지 정보를 요청한다. 마이페이지를 보여주고자 하는 친구를 식별하기 위한 String 타입의 friendId를 인자로 받는다.			

FriendRepository				
Class Description	외부 서버와의 통신을 위한 method를 관리하는 class			
구분	Name	Type	Visibility	
	Description			
Attributes	None			
구분	Name	Argument	Returns	
	Description			
Operations	searchUser	String, (Int, List<User>) -> Unit	none	
	서버에 사용자 아이디 또는 닉네임으로 사용자 검색 요청을 보내고 응답을 받는 operation이다. 검색하고자 하는 사용자 아이디 또는 닉네임인 String 타입의 userIdOrNickName, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 사용자 검색 요청의 응답인 검색된 사용자 목록을 가져와 처리한다.			
	addFriend	String, (Int) -> Unit	none	
	서버에 친구 추가 요청을 보내고 응답을 받는 operation이다. 추가하고자 하는 사용자 아이디인 String 타입의 friendUserId, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 친구 추가 요청의 응답을 처리한다.			
	deleteFriend	String, (Int) -> Unit	none	
	서버에 친구 삭제 요청을 보내고 응답을 받는 operation이다. 삭제하고자 하는 친구 아이디인 String 타입의 friendUserId, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 친구 삭제 요청의 응답을 처리한다.			
	blockFriend	String, (Int) -> Unit	none	
	서버에 친구 차단 요청을 보내고 응답을 받는 operation이다. 차단하고자 하는 친구 아이디인 String 타입의 friendUserId, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 친구 차단 요청의 응답을 처리한다.			
	getFriendList	String, (Int, List<User>) -> Unit	none	
	서버에 친구 목록 요청을 보내는 응답을 받는 operation이다. 사용자의 아이디인 String 타입의 userId, 그리고 onResult 함수를 인자로 받는다. onResult 함수는 친구 목록 가져오기 요청의 응답인 사용자의 친구 목록을 가져와 처리한다.			

3.6.5 Profile class diagram



[그림 3-12] Profile class diagram

ProfileFragment			
Class Description	사용자가 프로필 화면을 확인할 때 제공하는 class		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
Operations	None		

FriendProfileFragment			
Class Description	다른 사용자의 프로필 화면을 확인할 때 제공하는 class		
구분	Name	Type	Visibility
	Description		
Attributes	None		
구분	Name	Argument	Returns
	Description		
Operations	None		

ProfileEditFragment			
Class Description	사용자가 프로필을 수정할 때 제공하는 class		
구분	Name	Type	Visibility
	Description		
Attributes	None		
구분	Name	Argument	Returns
	Description		
Operations	getFileLength Uri로부터 file을 읽어 file의 크기를 알아오는 operation이다. 인자로 받은 uri로부터 file의 크기 Int 값을 return한다. getFileName Uri로부터 file을 읽어 file의 이름을 알아오는 operation이다. 인자로 받은 uri로부터 file의 이름 String 값을 return한다.	Uri	Int String

StatisticsDialogFragment			
Class Description	사용자가 통계를 확인할 때 dialog를 제공하는 class		
구분	Name	Type	Visibility
	Description		
Attributes	None		
구분	Name	Argument	Returns
	Description		
Operations	None		

SettingsFragment			
Class Description	사용자가 설정을 확인할 때 제공하는 class		
구분	Name	Type	Visibility
Attributes			Description
Attributes	None		
구분	Name	Argument	Returns
Operations	None		Description

ProfileViewModel			
Class Description	사용자의 프로필 정보와 통계 정보를 관리하는 class		
구분	Name	Type	Visibility
Attributes			Description
	editProfileEvent	SingleLiveEvent<Any>	private
	사용자가 프로필 수정 기능을 시작할 때를 알리는 이벤트.		
	editProfileCompleteEvent	SingleLiveEvent<Int>	private
	사용자가 프로필 수정을 완료할 때를 알리는 이벤트.		
	uploadImageEvent	SingleLiveEvent<Any>	private
	사용자가 프로필 이미지를 업로드할 때를 알리는 이벤트		
	viewStatisticsEvent	SingleLiveEvent<Int>	private
	사용자가 통계 정보를 볼 때를 알리는 이벤트.		
	viewStatisticsExitEvent	SingleLiveEvent<Any>	private
	사용자가 통계 정보 확인을 그만둘 때를 알리는 이벤트.		
	viewFriendProfileEvent	SingleLiveEvent<Int>	private
	사용자가 다른 사용자의 프로필을 볼 때를 알리는 이벤트.		
	selectOptionEvent	SingleLiveEvent<Int>	private
	사용자가 옵션을 변경할 때를 알리는 이벤트.		
	logoutEvent	SingleLiveEvent<Int>	private
	사용자가 로그아웃 기능을 시작할 때를 알리는 이벤트.		
	profileLiveData	MutableLiveData<Profile>	private
	사용자의 프로필 정보를 저장하는 객체		
	statisticsLiveData	MutableLiveData<Statistics>	private
	사용자의 통계 정보를 저장하는 객체		
	planListLiveData	MutableLiveData<List<SimplePlan>>	private
	사용자의 약속 리스트를 저장하는 객체		
	naverLogoutEvent	SingleLiveEvent<Any>	private
	Naver 소셜 API를 사용하여 로그인한 사용자가 로그아웃 할 때를 알리는 이벤트		
	optionLiveData	MutableLiveData<Option>	private
	사용자의 기능 설정 정보를 저장하는 객체		
	noticeOptionLiveData	MutableLiveData<Boolean>	private
	사용자의 기능 설정 정보 중 푸시 알림 설정 정보를 저장하는 객체		
	inviteFriendOptionLiveData	MutableLiveData<Boolean>	private
	사용자의 기능 설정 정보 중 친구 초대 수신 설정 정보를 저장하는 객체		
	invitePlanOptionLiveData	MutableLiveData<Boolean>	private
	사용자의 기능 설정 정보 중 약속 초대 수신 설정 정보를 저장하는 객체		

	imageBitmapLiveData	MutableLiveData<Bitmap>	private
	사용자의 프로필 사진을 저장하는 객체		
	imageFile	File	public
	사용자의 프로필 사진 파일		
	imageUri	Uri	public
	사용자의 프로필 사진 경로		
	imageStream	FileInputStream	public
	사용자의 프로필 사진 파일 스트림		
	imageLength	Int	public
	사용자의 프로필 사진 크기		
	getStatisticsCompleteEvent	SingleLiveEvent<Any>	private
	서버에 요청한 사용자 약속 통계 정보를 수신받았음을 알리는 이벤트		
	abscentPlanCountLiveData	MutableLiveData<Int>	private
	완료하지 않은 약속의 개수를 저장하는 객체		
	attendPercentLiveData	MutableLiveData<Double>	private
	사용자의 전체 약속의 완료율을 저장하는 객체		
	isTotalPlanCount	Boolean	public
	전체 약속의 개수인지 확인하는 플래그		
	isAttendPlanCount	Boolean	public
	완료한 약속의 개수인지 확인하는 플래그		
	isAbsentPlanCount	Boolean	public
	완료하지 않은 약속의 개수인지 확인하는 플래그		
	isAttendPercent	Boolean	public
	전체 약속의 완료율인지 확인하는 플래그		
	googleSignInClient	GoogleSignInClient	public
	Google 로그인 API를 사용하기 위한 client 객체		
구분	Name	Argument	Returns
		Description	
Operations	setProfile	String	none
	profileLiveData에 대한 setter이다. 사용자를 구분하기 위해 String 타입의 userId를 인자로 받는다.		
	setStatistics	none	none
	statisticsLiveData에 대한 setter이다.		
	setOption	none	none
	optionLiveData에 대한 setter이다.		

ProfileLogic			
Class Description	Fragment에서의 동작과 내부 로직을 관리하는 class		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
Operations	onEditProfileClick	String	none
	사용자가 프로필 수정 버튼을 눌렀을 때 호출되는 operation이다. String 타입의 현재 사용자의 userId 변수를 인자로 입력받는다.		
	onEditProfileCompleteClick	Profile	none
	사용자가 프로필 수정 화면에서 완료 버튼을 눌렀을 때 호출되는 operation이다. Profile 타입의 프로필 객체를 입력받아 ProfileRepository를 통해 서버에 프로필 수정		

을 요청한다.

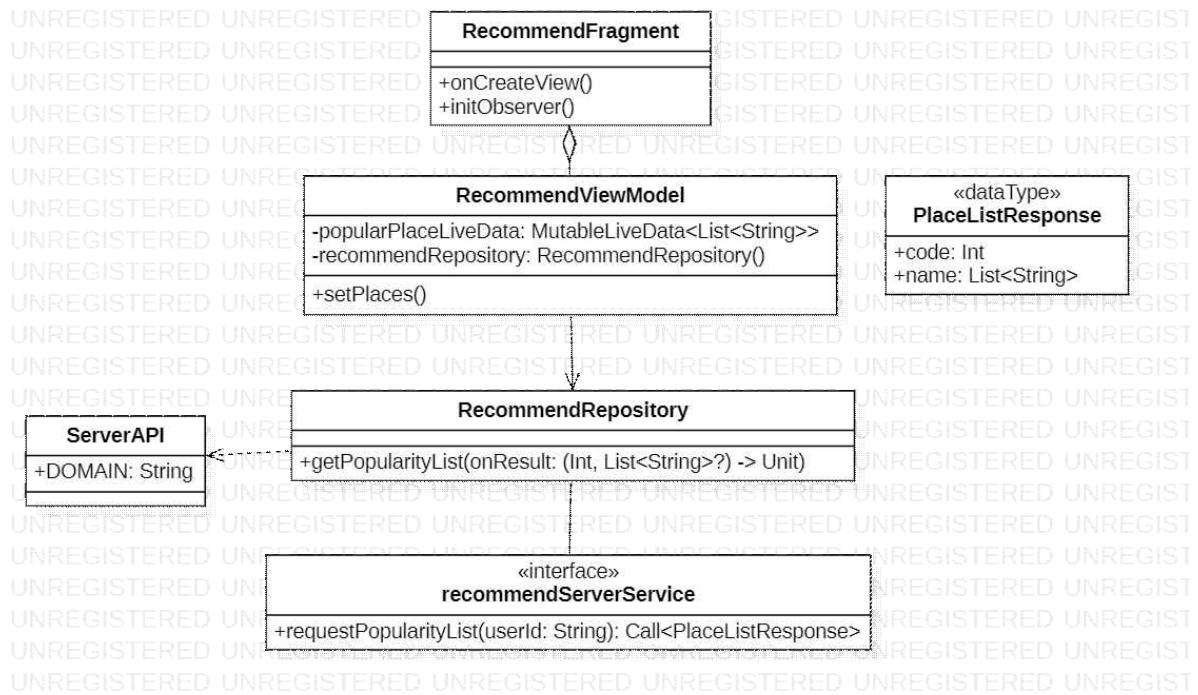
onUploadImageClick	none	none
사용자가 imageView를 눌렀을 때 호출되는 operation이다.		
onViewStatisticsClick	none	none
사용자가 통계 확인 버튼을 눌렀을 때 호출되는 operation이다.		
onViewStatisticsExitClick	none	none
사용자가 통계 화면에서 확인 버튼을 눌렀을 때 호출되는 operation이다.		
onSelectOptionClick	none	none
사용자가 설정 버튼을 눌렀을 때 호출되는 operation이다.		
onSelectNoticeOptionClick	Boolean	none
사용자가 푸시 알림 설정 스위치를 눌렸을 때 호출되는 operation이다. 호출되면 ProfileRepository를 통해 서버에 푸시 알림 설정 변경을 요청한다. 변경된 약속을 확인하기 위해 Boolean 타입의 option을 인자로 받는다.		
onSelectInviteFriendOptionClick	Boolean	none
사용자가 약속 초대 설정 스위치를 눌렸을 때 호출되는 operation이다. 호출되면 ProfileRepository를 통해 서버에 약속 초대 설정 변경을 요청한다. 변경된 약속을 확인하기 위해 Boolean 타입의 option을 인자로 받는다.		
onSelectInvitePlanOptionClick	Boolean	none
사용자가 친구 추가 설정 스위치를 눌렸을 때 호출되는 operation이다. 호출되면 ProfileRepository를 통해 서버에 친구 추가 설정 변경을 요청한다. 변경된 약속을 확인하기 위해 Boolean 타입의 option을 인자로 받는다.		
onLogoutClick	none	none
사용자가 로그아웃 버튼을 눌렀을 때 호출되는 operation이다. 호출되면 ProfileRepository를 통해 서버에서 로그아웃한다.		

ProfileRepository

Class Description	외부 서버와의 통신을 위한 method를 관리하는 class		
구분	Name	Type	Visibility
Description			
Attributes	None		
구분	Name	Argument	Returns
Description			
Operations	editProfile	Profile, (Int) -> Unit	none
	서버에 프로필 정보 수정을 요청하고 응답을 받는 operation이다. 프로필 정보를 포함하는 Profile 타입의 profile과 onResult 함수를 인자로 받는다. onResult는 프로필 수정 요청의 응답을 처리한다.		
	editProfileImage	File, FileInputStream, Int, String, (Int) -> Unit	none
	서버에 프로필 사진 수정을 요청하고 응답을 받는 operation이다. File 타입의 프로필 사진 file, 프로필 사진의 FileInputStream, Int 타입의 사진의 크기 정보인 length, String 타입의 확장자 정보인 extension 변수, 그리고 onResult 함수를 인자로 받는다. onResult는 프로필 사진 수정 요청의 응답을 처리한다.		
	getProfile	String, (Int, Profile)-> Unit	none
	서버에 프로필 정보를 요청하고 응답을 받는 operation이다. 대상 식별을 위한 String 타입의 userId와 onResult 함수를 인자로 받는다. onResult는 프로필 정보 요청의 응답을 처리한다.		
	getProfileImage	String, (Int, Bitmap)->	none

Unit		
서버에 프로필 사진을 요청하고 응답을 받는 operation이다. 대상 식별을 위한 String 타입의 userId와 onResult 함수를 인자로 받는다. onResult는 프로필 사진 정보 요청의 응답을 처리한다.		
getPlanList	String, (List<SimplePlan>)-> Unit	none
서버에 약속 정보를 요청하고 응답을 받는 operation이다. 대상 식별을 위한 String 타입의 userId와 onResult 함수를 인자로 받는다. onResult는 약속 목록 요청의 응답을 처리한다.		
getStatistics	String, (Int, Statistics)-> Unit	none
서버에 통계 정보를 요청하고 응답을 받는 operation이다. 대상 식별을 위한 String 타입의 userId와 onResult 함수를 인자로 받는다. onResult는 통계 정보 요청의 응답을 처리한다.		
getOption	(Int, Option)-> Unit	none
서버에 사용자의 기능 설정 정보를 요청하고 응답을 받는 operation이다. onResult 함수를 인자로 받는다. onResult는 기능 설정 정보 요청의 응답을 처리한다.		
updateOption	Boolean, Boolean, Boolean, (Int)-> Unit	none
서버에 사용자의 기능 설정 변경을 요청하고 응답을 받는 operation이다. 설정 상태를 구분하기 위한 Boolean 타입의 noticeOption, friendInviteOption, planInviteOption, 그리고 onResult 함수를 인자로 받는다. onResult는 기능 설정 변경 요청의 응답을 처리한다.		
logout	String, (Int)-> Unit	none
서버에 로그아웃 요청을 보내고 응답을 받는 operation이다. 대상 식별을 위한 String 타입의 userId와 onResult 함수를 인자로 받는다. onResult는 로그아웃 요청의 응답을 처리한다.		
getSavedAccount	none	Account
로컬 DB에 저장된 계정 정보를 가져오는 operation이다. 로컬 DB에서 Account 타입의 계정 정보를 return한다.		
deleteAccount	none	none
로컬 DB에 저장된 계정 정보를 삭제하는 operation이다.		

3.6.6 Recommend a place class diagram

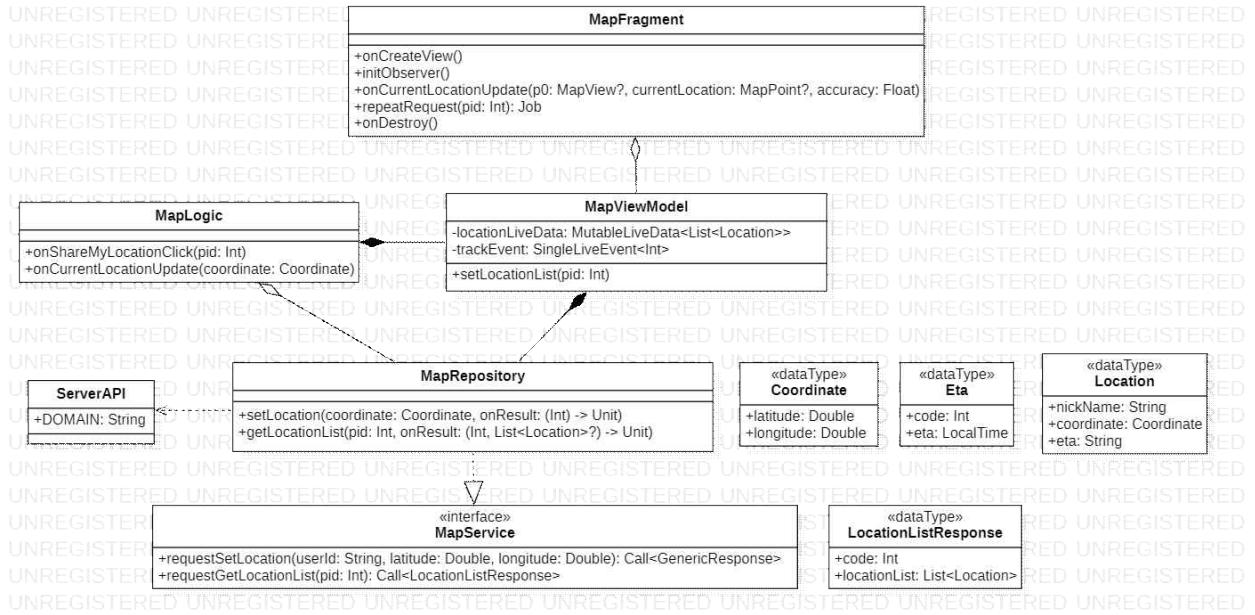


[그림 3-13] Recommend class diagram

RecommendViewModel			
Class Description	추천 장소에 대한 정보를 관리하는 class		
구분	Name	Type	Visibility
Attributes	popularPlaceLiveData	MutableLiveData<List<String>>	private
	인기 약속 장소의 이름을 저장하는 객체		
	recommendRepository	RecommendRepository	private
	추천 장소 요청을 보내기 위한 repository를 저장하는 객체		
구분	Name	Argument	Returns
Operations	setPlaces	none	none
	popularPlaceLiveData에 대한 직접적인 setter이다.		

RecommendRepository			
Class Description	외부 서버와의 통신을 위한 method를 관리하는 class		
구분	Name	Type	Visibility
Attributes	None		
구분	Name	Argument	Returns
Operations	getPopularityList	(Int, List<String>?) -> Unit	none
	DB 서버에서 장소의 인기도순으로 정렬된 리스트를 가져오는 method이다. onResult 함수를 인자로 받는다. 서버에서 인기 장소 목록 요청의 응답을 처리한다.		

3.6.7 Track real-time location class diagram



[그림 3-15] Track real-time location CD

MapFragment				
Class Description	사용자에게 지도 및 위치 정보를 제공하는 fragment			
구분	Name	Type	Visibility	
Attributes	None		Description	
구분	Name	Argument	Returns	
Operations	onCurrentLocationUpdate	MapView?, MapPoint?, Float	MapView?, MapPoint?, Job	none
	사용자의 현재 위치를 서버에 저장하는 Operation이다.			
	repeatRequest	Int	Job	
	서버로부터 약속에 참여한 사용자들의 위치 요청을 반복적으로 실행하는 Operation이다.			
	onDestroy	none	none	
	MapFragment가 소멸할 때 실행되는 Operation이다.			

MapViewModel				
Class Description	사용자의 위치 정보, 도착 예상 시간을 관리하는 class			
구분	Name	Type	Visibility	
Attributes	locationLiveData	MutableLiveData<List<Location>>	private	
	사용자들의 위치 좌표값을 저장하는 객체			
Operations	trackEvent	SingleLiveEvent<int>	private	
	위치 공유 버튼을 눌렀을 때를 알리는 객체. 결과로 경우에 따라 오류 메시지를 출력한다.			
구분	Name	Argument	Returns	
Operations	setLocationEvent	Int	none	
	약속에 참여한 사용자들의 위치를 불러오는 Operation이다.			

MapLogic				
Class Description	MapFragment에서의 동작과 내부 로직을 관리하는 class			
구분	Name	Type	Visibility	
Attributes	None			
구분	Name	Argument	Returns	
Operations	onShareMyLocationClick	Int	none	
	사용자가 위치 공유 버튼을 눌렀을 때 호출되는 operation이다.			
Operations	onCurrentLocationUpdate	Coordinate	none	
	사용자 디바이스의 현위치 좌표값을 통보받을 수 있다.			

MapRepository				
Class Description	외부 서버와의 통신을 위한 method를 관리하는 class			
구분	Name	Type	Visibility	
Attributes	None			
구분	Name	Argument	Returns	
Operations	setMyLocation	Coordinate, (Int) -> Unit	none	
	DB 서버에 사용자 디바이스의 현재 위치를 저장할 때 사용하는 operation이다. 좌표 정보인 Coordinate 타입의 myCoord와 onResult 함수를 인자로 받는다. onResult 함수는 위치 좌표 저장 요청의 응답을 처리한다.			
Operations	getLocation	Int, (Int, List<Location>?) -> Unit	none	
	DB 서버에서 다른 사용자의 좌표 정보를 받아올 때 사용하는 operation이다. 대상 사용자를 식별하기 위한 uid값을 Int 타입과 onResult 함수를 인자로 받는다. onResult 함수는 대상 사용자의 위치 좌표 요청의 응답을 처리한다.			

4. Sequence diagram

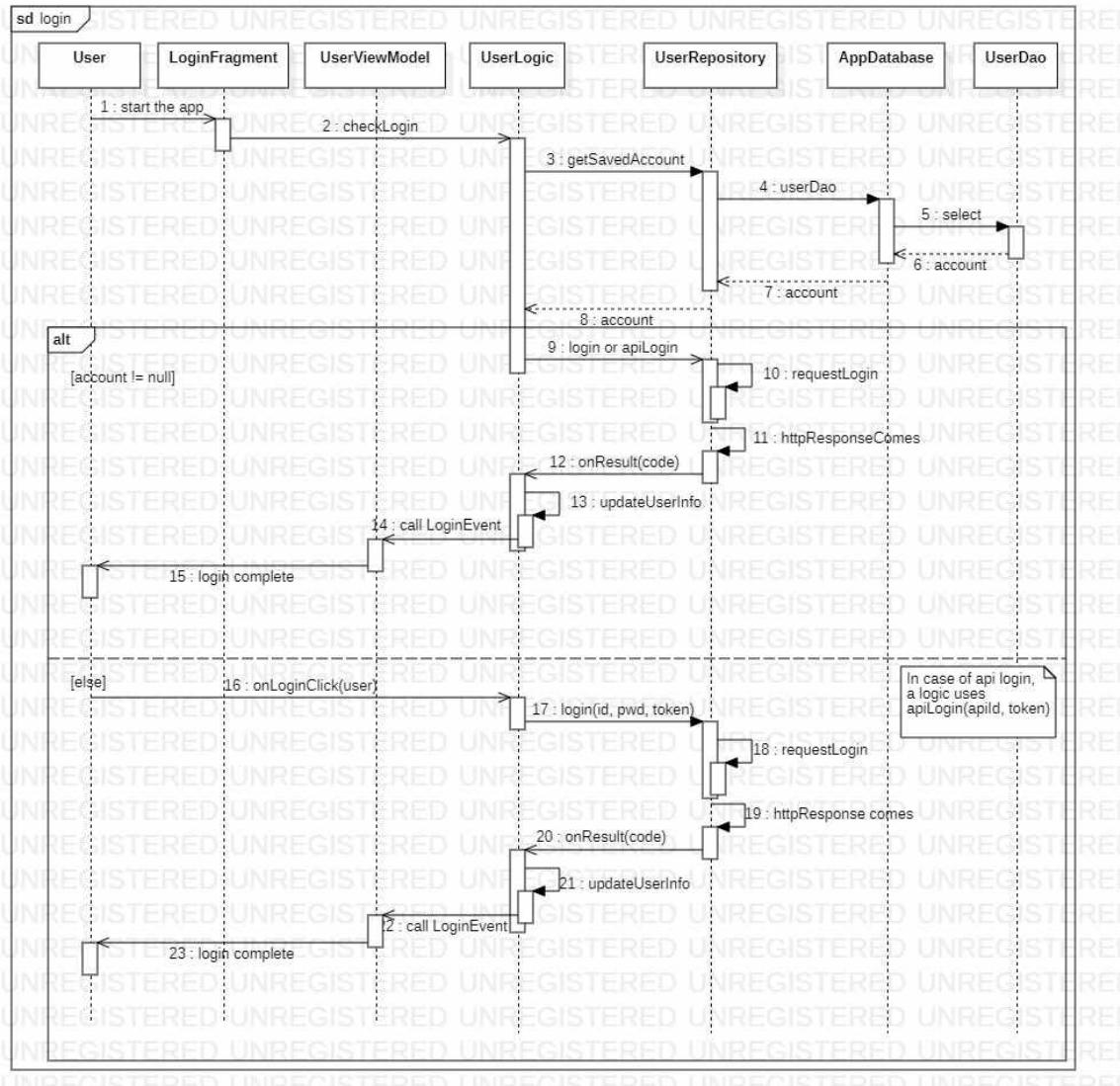
이 장은 sequence diagrams(이하 SD)와 그 설명을 제공한다. 고려 사항은 다음과 같다.

- 모든 SD는 특정 use case와 대응된다.
- 모든 SD는 특정 use case의 main/extension success scenario의 흐름을 따른다.
- 본 시스템에선 객체 이름을 중요하게 생각하지 않으므로 생명선의 class 명칭은 객체이름과 함께 표현하지 않고 클래스 이름만 적었다.
- 메시지를 주고받는 클래스들은 모두 class diagram에서 연결 관계가 있다.
- SD의 메시지는 CD의 operation 혹은 attributes이거나 CD에 없는 개념적인 (혹은 시스템 내부의) 과정(e.g. observing)/결과(e.g. creating plan complete)를 나타낸다.
- User는 실제 코드 상에 있는 클래스가 아니라 사용자를 뜻하는 개념적 클래스다.
- Repository에서 서버에 요청을 보낼 때는 Retrofit을 사용하는 것을 전제로 설계 했으며 비동기 메시지 전달 방법(Call.enqueue()[8]로 구현)을 사용한다. -> 함수 호출 시 주로 비동기 메시지로 호출한다.
- Http request를 여러 개 보낼 때 편의를 위해 결과가 오는 순서를 지정했지만 실제 환경에서 그 순서는 다를 수 있다.
- Util 클래스들의 내부 동작까지 모두 기술하는 것은 크게 필요하지 않은 동작들로 가독성을 해칠 뿐만 아니라 대부분은 시스템이 자동으로 처리하는 것이기 때문에 SD에 반영하지 않았다.
- OnCreateView와 initObserver는 모든 view에서 실행하기 때문에 따로 적지 않는다. 단, OnCreateView에서 view만의 고유한 연산을 실행한다면 명시한다.
- Repository에서 요청을 보낼 때 (request~ 연산) 모두 ~Service의 연산을 사용한다. 자세히 하자면 repository에서 Service~의 request~를 호출하면 Call 객체를 반환한다. Repository는 call을 이용해 요청을 보낸다. 하지만 모두 공통적인 흐름 일 뿐만 아니라 가독성을 해치기 때문에 request~를 repository 내부의 연산으로 단순화했다.
- 모든 request~는 code를 반환한다. 모든 SD는 code = 0(성공)일 때를 가정한다. (success scenario)
- SD의 description은 관련된 use case, class diagram과 흐름 묘사로 이뤄져 있다.
- 흐름 묘사는 SD를 그대로 글에 옮겨 적은 것이다. 일치성과 모순성을 검사하기

위해 사용한다. 혹은 diagram으로 이해되지 않는 부분을 해소하는 데 사용한다.

4.1 User sequence diagram

Login



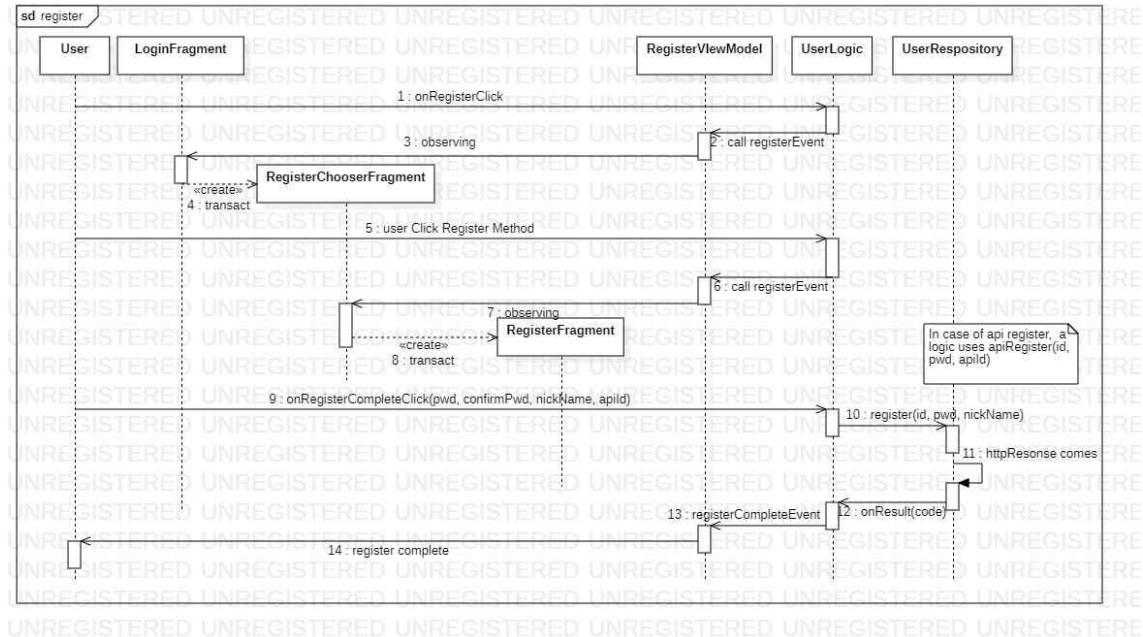
[그림 4-11] Login SD

사용자가 시스템에 로그인하는 use case를 나타내는 sequence diagram이다. User class description에서 <Use case #14>의 경우이다.

시스템이 시작되면서 기능을 시작한다. 우선 로그인 되어있는지 확인하기 로컬 DB에 저장된 계정 정보를 가져온다. 계정이 존재하는 경우 서버에 접속하였다느 요청을 보내고 약속 목록 화면으로 전환한다. 계정이 존재하지 않는다면 사용자가 로그인 화면에 아이디, 비밀번호를 입력한 후 로그인 버튼을 누르면 서버에 로그인 하고, 사용자의

계정을 로컬 DB에 저장하고 약속 목록 화면으로 전환한다.

Register



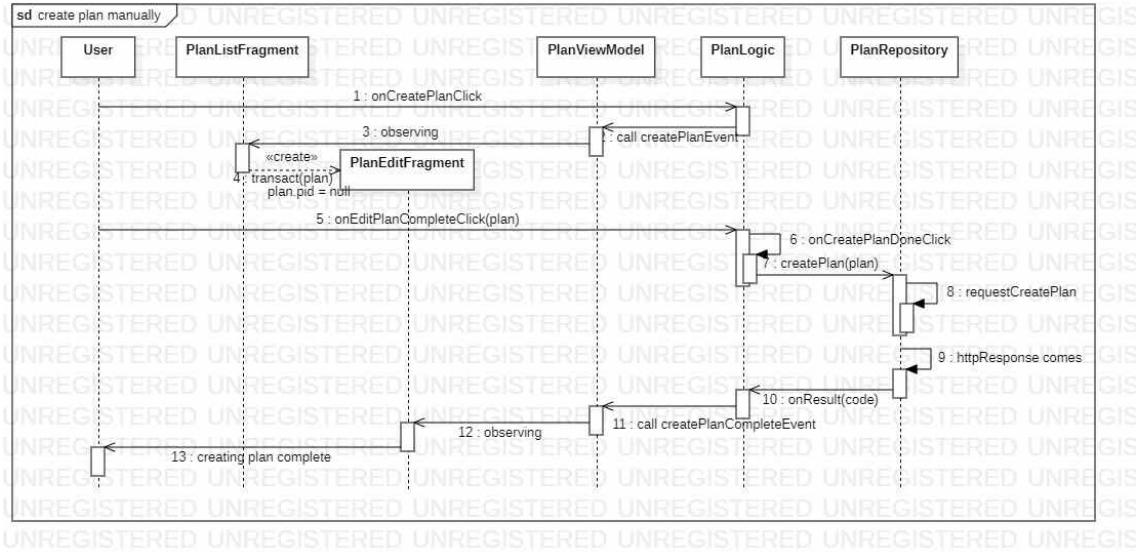
[그림 4-2] Register SD

사용자가 시스템에 회원가입하는 use case를 나타내는 sequence diagram이다. User class description에서 <Use case #2>의 경우이다.

사용자가 회원가입 버튼을 누르면 회원가입 방법(자체 회원가입, 구글, 네이버)을 묻는 화면이 나온다. 사용자가 원하는 방법을 선택하면 회원가입 화면이 표시된다. 사용자가 회원가입 화면에 아이디, 비밀번호, 비밀번호 확인, 닉네임을 입력한 후 회원가입 버튼을 누르면 서버에 회원가입 요청을 하고 응답을 받으면 로그인 화면으로 전환한다. API를 사용한 로그인시엔 비밀번호를 받지 않는다.

4.2 Plan sequence diagram

Create plan manually

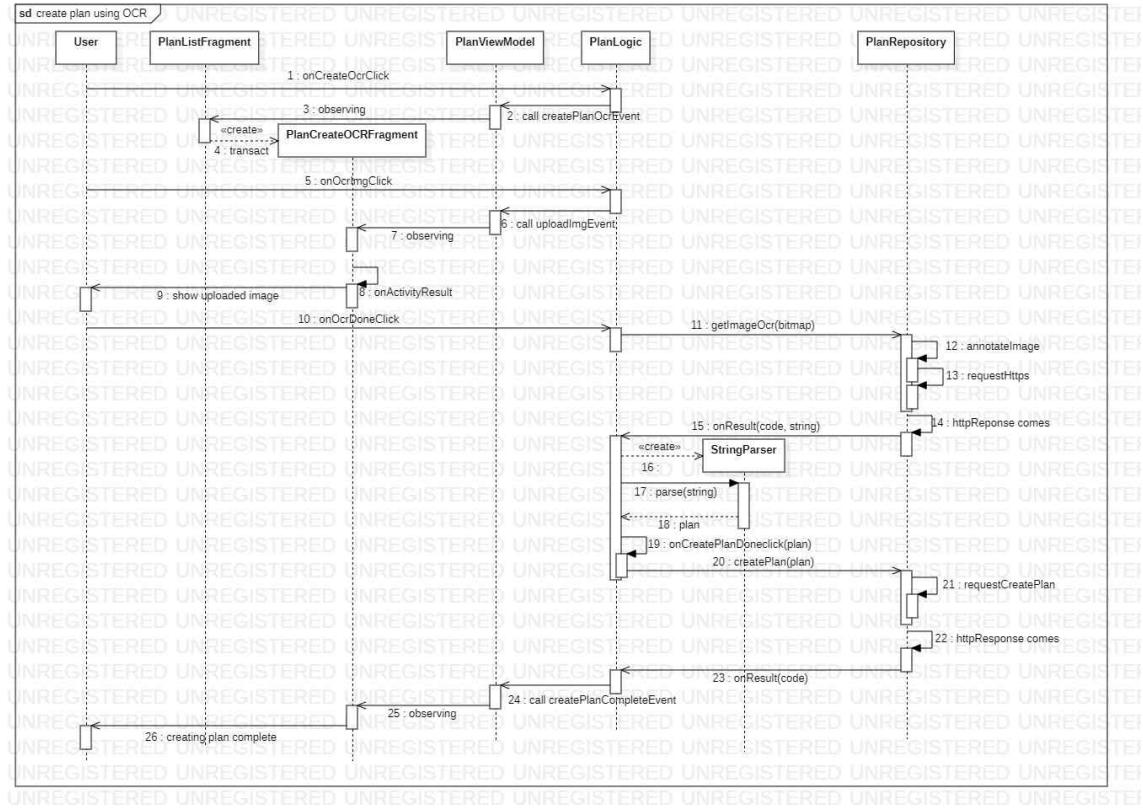


[그림 4-3] Create plan manually SD

사용자가 새 약속을 생성하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-1>의 경우이다.

사용자가 약속 생성 버튼을 누름으로써 시작한다. 약속 생성 화면으로 전환 후 사용자는 약속 상세 정보를 입력한다. 화면은 수정과 생성 모두 같은 화면이지만 내부적으로 생성은 화면 생성시 받는 인자 plan의 pid가 null이라는 점이 다르다. 완료 버튼을 누르면 서버에 요청을 보내 약속을 생성한다.

Create plan using OCR

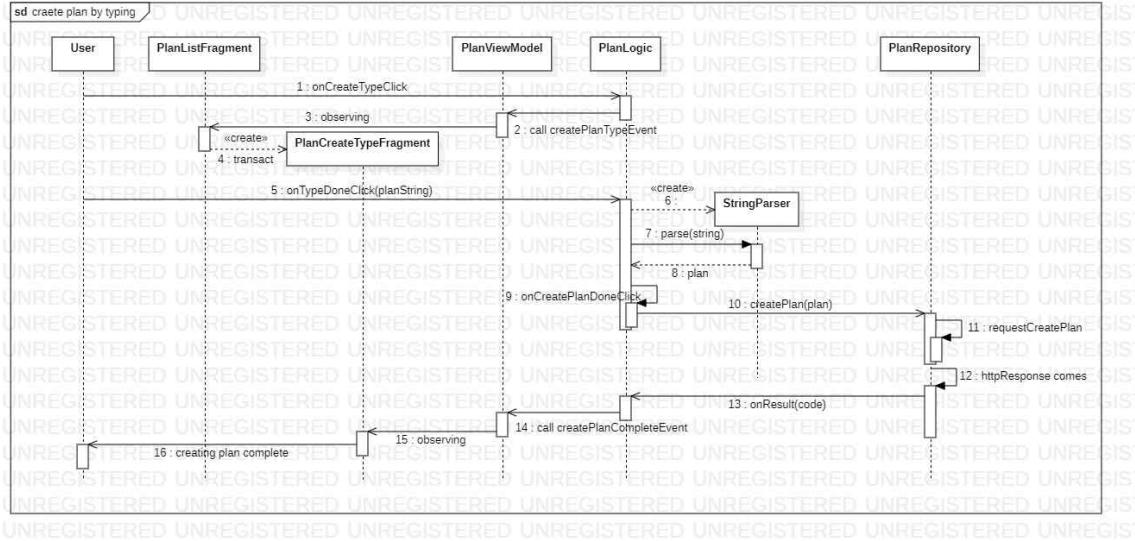


[그림 4-4] Create plan using OCR SD

사용자가 약속 정보가 써진 사진을 이용하여 새 약속을 생성하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-1b. 손글씨를 통한 약속 생성>의 경우이다.

사용자가 OCR로 약속 생성 버튼을 누름으로써 기능이 시작된다. OCR 약속 생성 화면으로 전환 후 사용자는 사진을 업로드한다. 업로드 후 사진을 화면에 표시해서 사용자가 확인할 수 있게 한다. 사용자가 완료 버튼을 누르면 OCR API를 통해 사진에서의 약속 정보가 담겨있는 문자열을 판독하고, StringParser class를 이용해 문자열을 약속으로 바꾼다. 서버에 요청을 보내 약속을 생성한다.

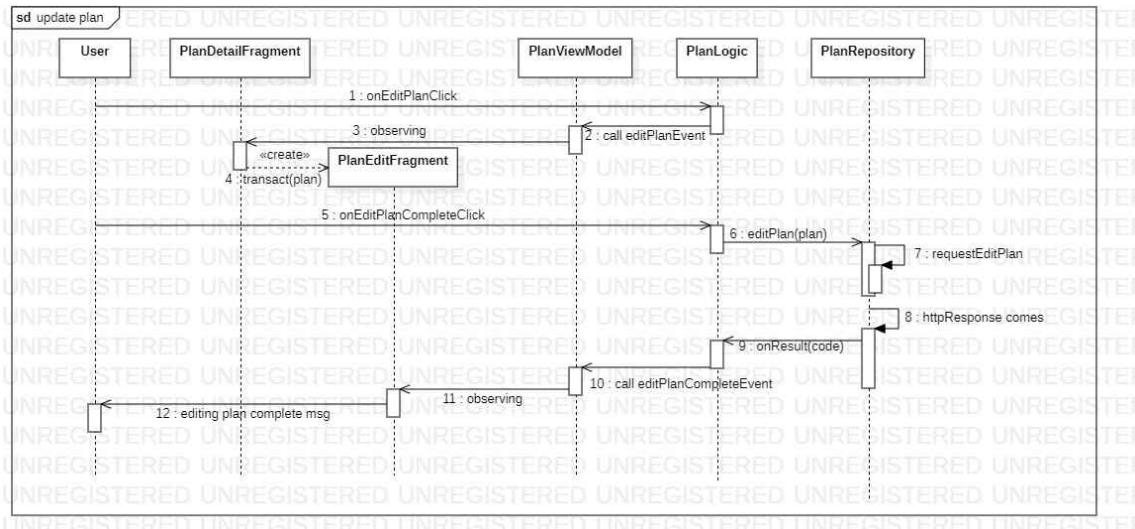
Create plan by typing



[그림 4-5] Create plan by typing SD

사용자가 문장으로 새 약속을 생성하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-1a. 타이핑을 통한 약속 생성>의 경우이다. 사용자가 문장으로 약속 생성 버튼을 누름으로써 기능이 시작된다. 문장 약속 생성 화면으로 전환 후 사용자는 문장을 입력한다. 사용자가 완료 버튼을 누르면 StringParser class를 이용해 문자열을 약속으로 바꾼다. 서버에 요청을 보내 약속을 생성한다.

Update plan

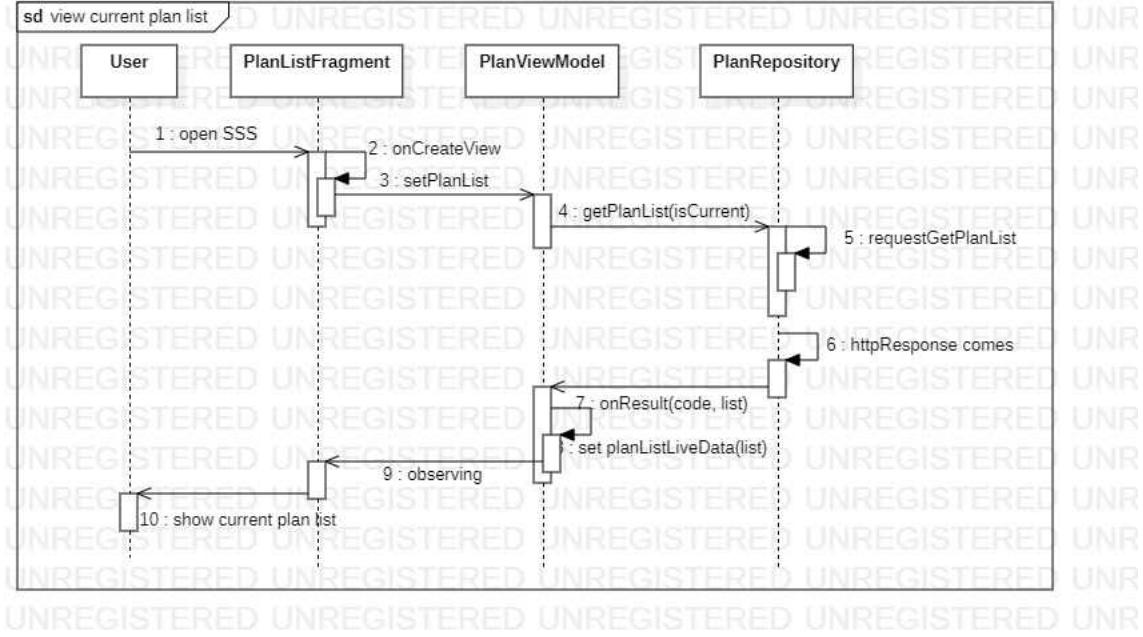


[그림 4-6] Update plan SD

사용자가 약속을 수정하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-9>의 경우이다.

사용자가 약속 수정 버튼을 누르면 기능은 시작된다. 약속 수정 화면으로 전환 후 사용자는 정보를 수정한다. 사용자가 완료 버튼을 누르면 서버에 요청을 보내 약속을 수정한다.

View current plan list

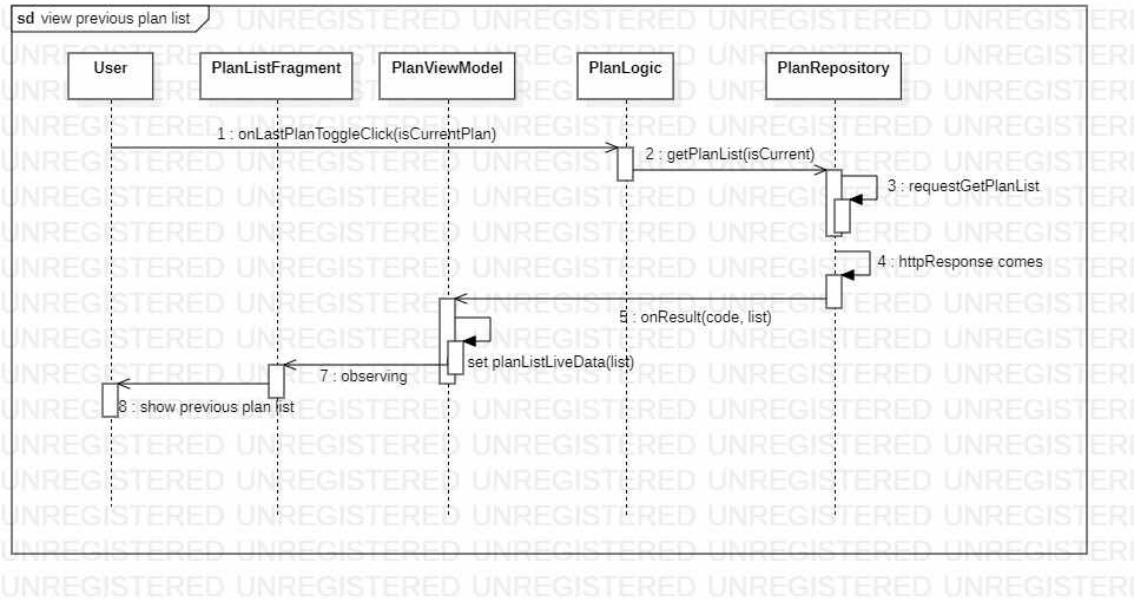


[그림 4-7] View current plan list SD

사용자가 현재 진행 중인 약속 목록을 열람하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-5>의 경우이다.

본 기능은 사용자가 로그인을 한 후 시작된다. 화면이 전환되면서 서버에 요청을 보내 약속 목록을 가져오고 사용자에게 제공한다.

View previous plan list

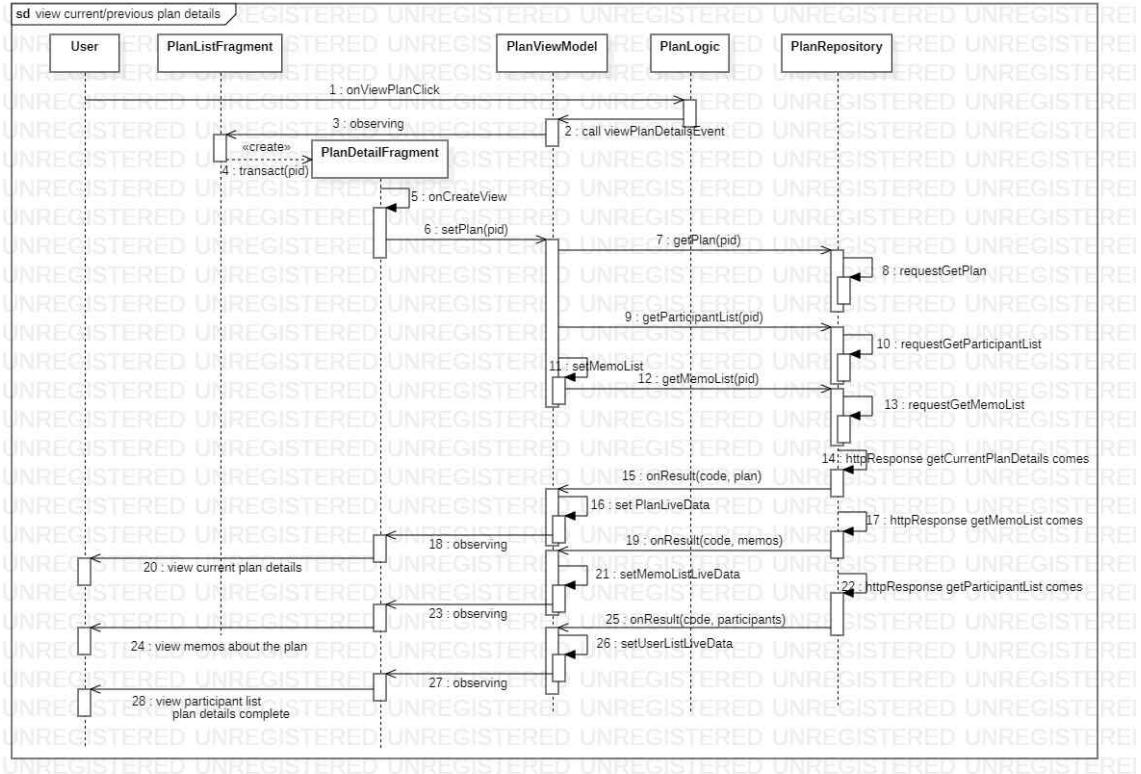


[그림 4-8] View previous plan list SD

사용자가 지난 약속 목록을 열람하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-5a. 지난 약속 보기>의 경우이다.

본 기능은 View current plan list에서 버튼의 상태를 옛날 약속 보기로 바꾸면 시작한다. 서버에 요청을 보내 예전 약속 목록을 가져오고 바꿔 사용자에게 제공한다.

View current/previous plan details

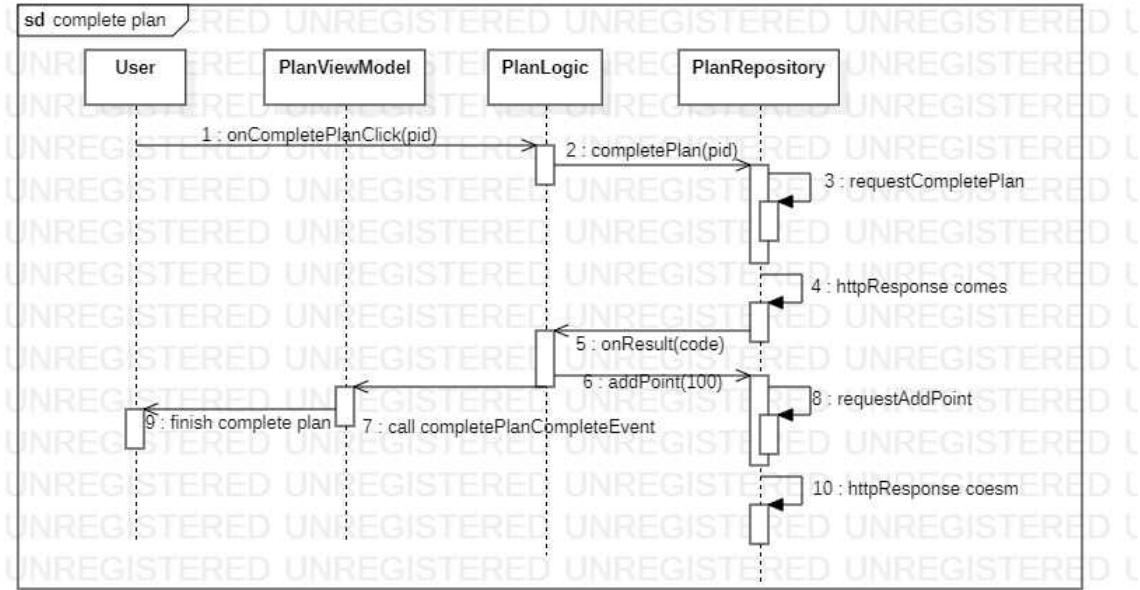


[그림 4-9] View current/previous plan details SD

사용자가 약속의 세부 내용을 열람하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-6>의 경우이다.

사용자가 View plan list에서 특정 약속을 누르면 기능을 시작한다. 약속 상세 내용 화면으로 전환되면서 서버에 요청을 보내 약속 그 자체, 메모, 참석자 정보를 가져오고 사용자에게 제공한다.

Complete plan

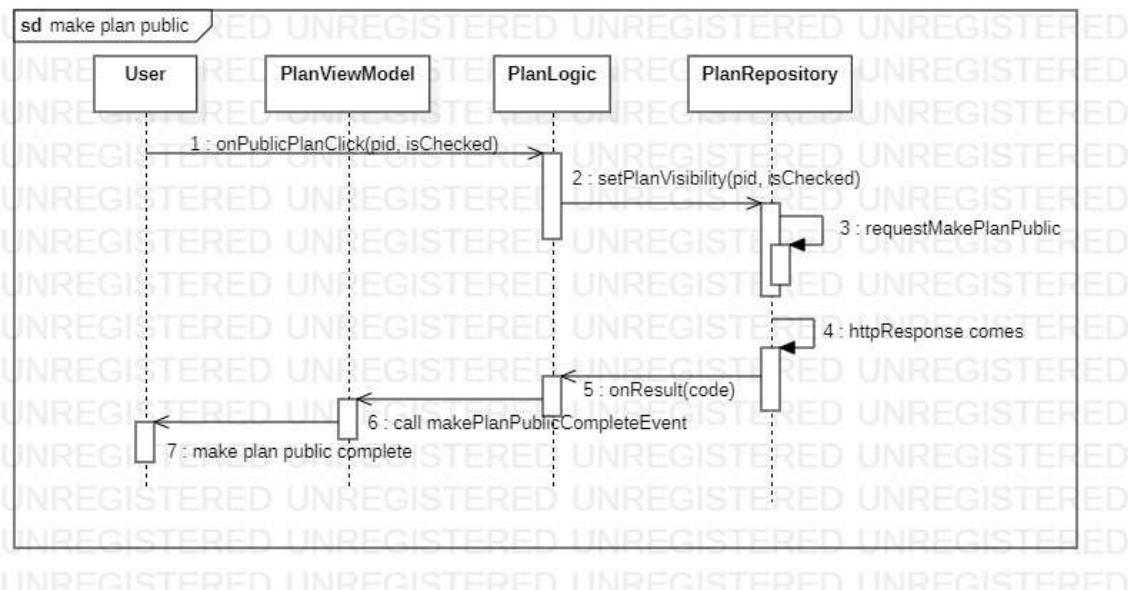


[그림 4-10] Complete plan SD

사용자가 수동적으로 약속을 완료하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-7>의 경우이다.

사용자는 약속 완료 버튼을 클릭하면 기능을 시작한다. 서버에 약속 완료 작업을 요청하고 변동 내용을 바꿔 출력한다.

Make plan public

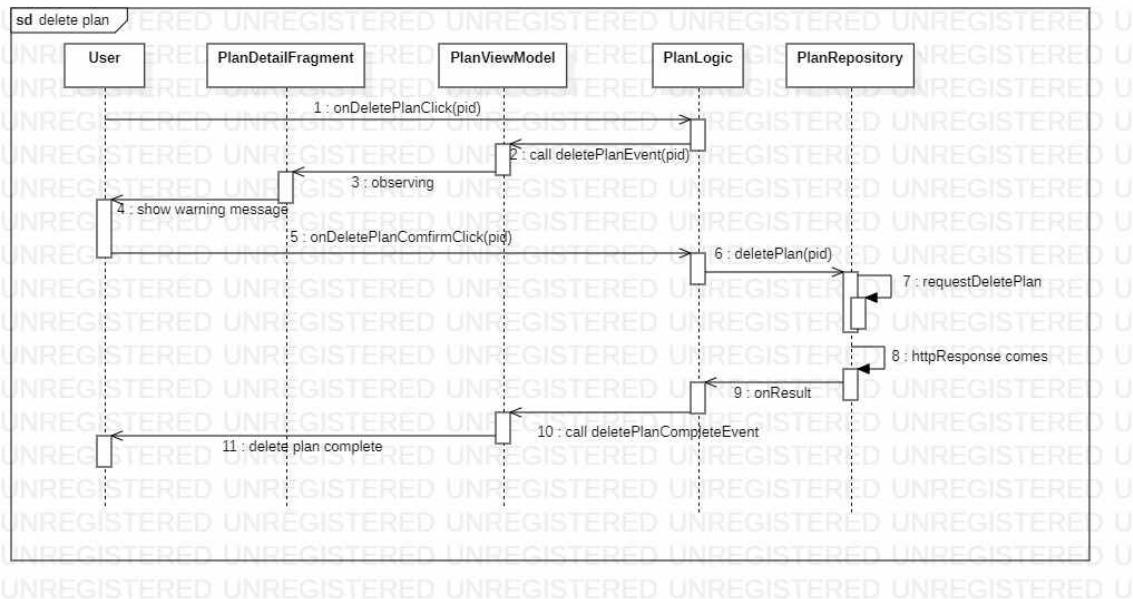


[그림 4-11] Make plan public SD

사용자가 약속의 공개 여부를 설정하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-5b. 약속 공개 여부 설정>의 경우이다.

사용자가 약속을 공개 약속으로 전환하고자 공개 약속으로 전환 버튼을 클릭하면 기능을 시작한다. 서버에 해당 약속을 공개 약속으로 전환 요청을 하고 해당 요청의 응답을 받아오면 변동 내용을 바꿔 출력한다.

Delete Plan

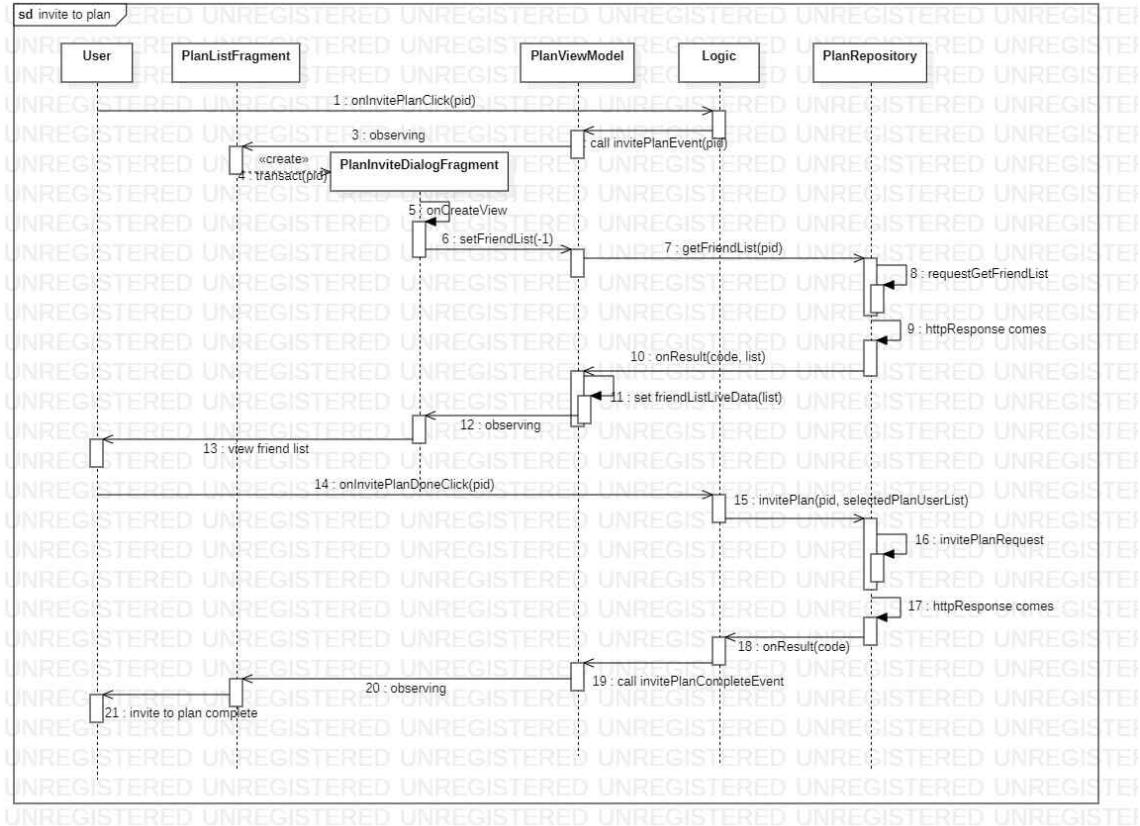


[그림 4-12] Delete Plan SD

사용자가 생성된 약속을 삭제하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-6a. 약속 삭제>의 경우이다.

사용자가 약속 상세 정보 화면에서 삭제 버튼을 누르면 기능을 시작한다. 사용자에게 경고 메시지를 출력하고 사용자가 확인 버튼을 누르면, 서버에 약속 삭제 요청을 하고 변동 내용을 바꿔 출력한다.

Invite to plan

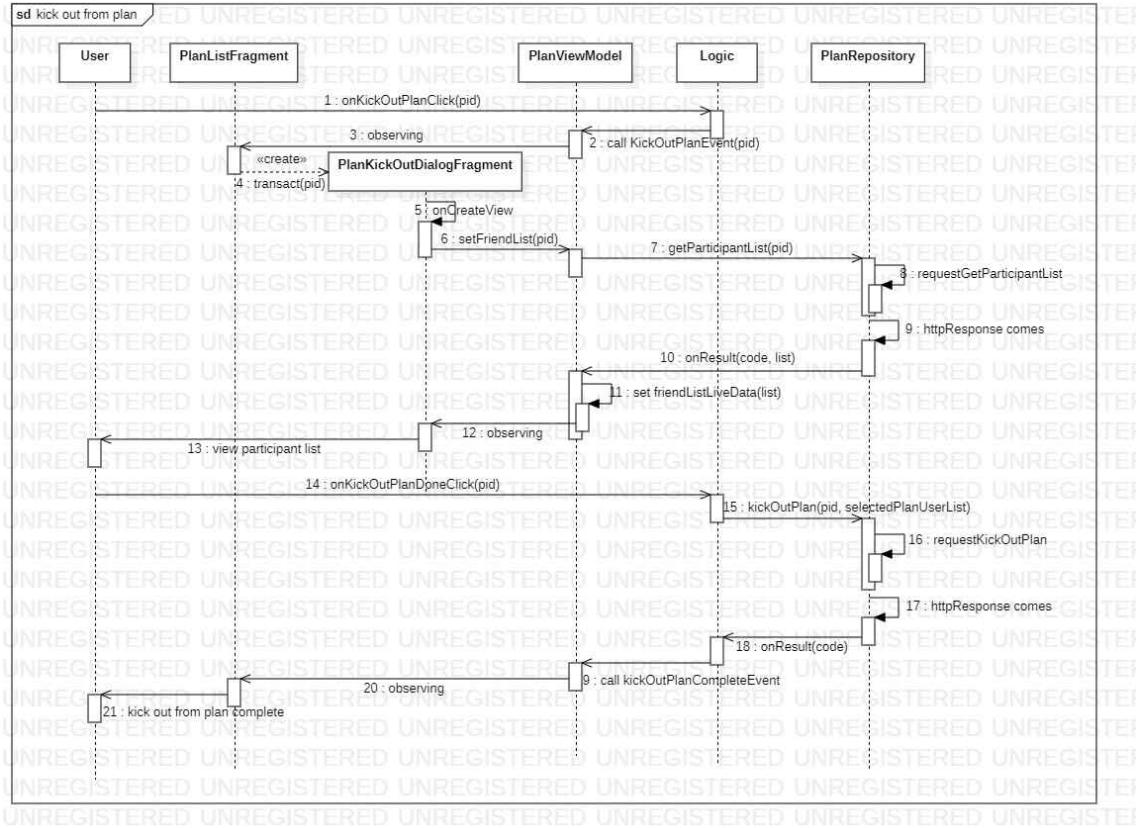


[그림 4-13] Invite to plan SD

사용자가 생성된 약속에 다른 사용자를 초대하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-6b. 약속 초대>의 경우이다.

사용자가 약속 상세 정보 화면에서 친구 초대 버튼을 누르면 기능을 시작한다. 약속 초대 화면을 Dialog의 형태로 화면을 생성한다. 서버에 친구 목록 가져오기 요청을 하고 사용자의 친구 목록을 가져와 약속 초대 화면에 친구 목록을 출력한다. 약속 초대 화면에서 초대하고자 하는 친구를 선택 후 초대 버튼을 누르면 서버에 약속 초대 요청을 하고 해당 약속의 참석자에 초대한 사용자가 추가된다.

Kick out from plan

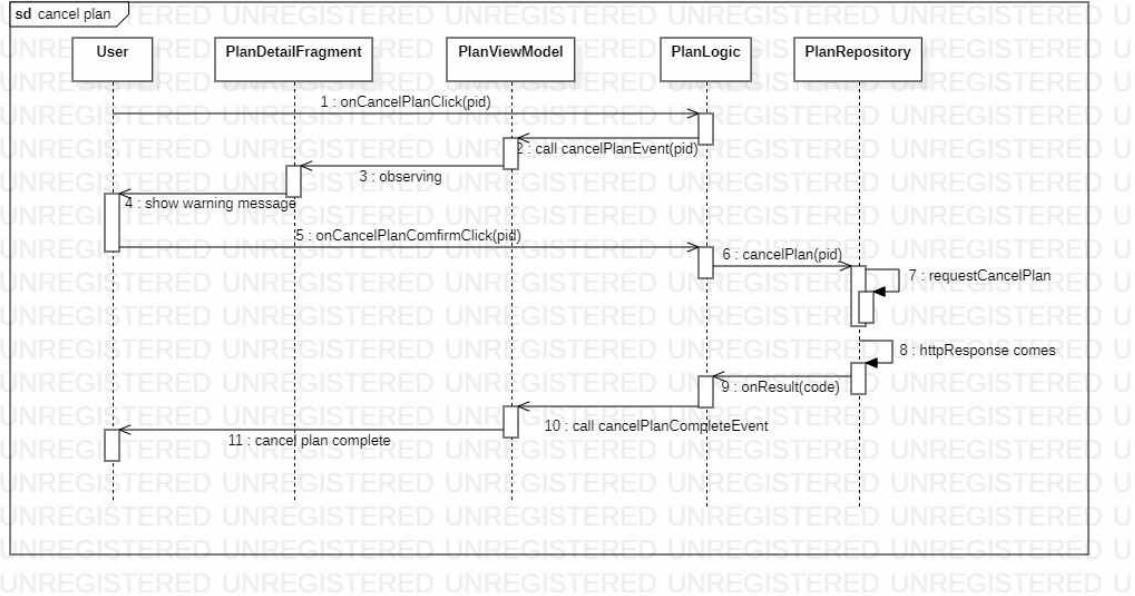


[그림 4-14] Kick out from plan SD

사용자가 생성된 약속에서 다른 참가자를 강제 퇴장하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-6c. 약속 강제 퇴장>과 <Use case #3-6d. 약속 강제 퇴장 취소>의 경우이다.

사용자가 약속 상세 정보 화면에서 강제 퇴장 버튼을 누르면 기능을 시작한다. 강제 퇴장 화면을 Dialog의 형태로 화면을 생성하고 서버에서 참석자 목록을 가져와 강제 퇴장 화면에 참석자 목록을 출력한다. 강제 퇴장 화면에서 강제 퇴장 하고자 하는 참석자를 선택 후 강제 퇴장 버튼을 누르면 서버에 강제퇴장 요청을 하고 해당 참석자들은 약속에서 제외된다.

Cancel plan

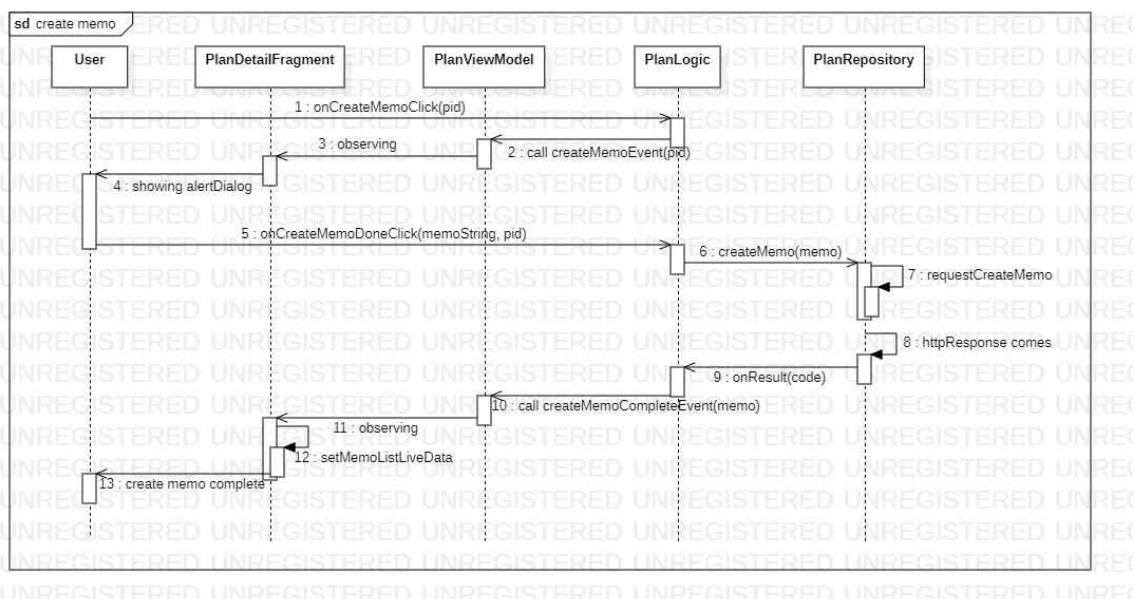


[그림 4-15] Cancel plan SD

사용자가 생성된 약속 참석을 취소하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #3-6e. 약속 취소>의 경우이다.

사용자가 약속 상세 정보 화면에서 참석 취소 버튼을 누르면 기능을 시작한다. 사용자에게 경고 메시지를 표시한다. 그리고 경고 메시지 창에서 사용자가 확인 버튼을 누르면 서버에 참석 취소 요청을 하고 사용자는 해당 약속의 참석이 취소된다.

Create memo

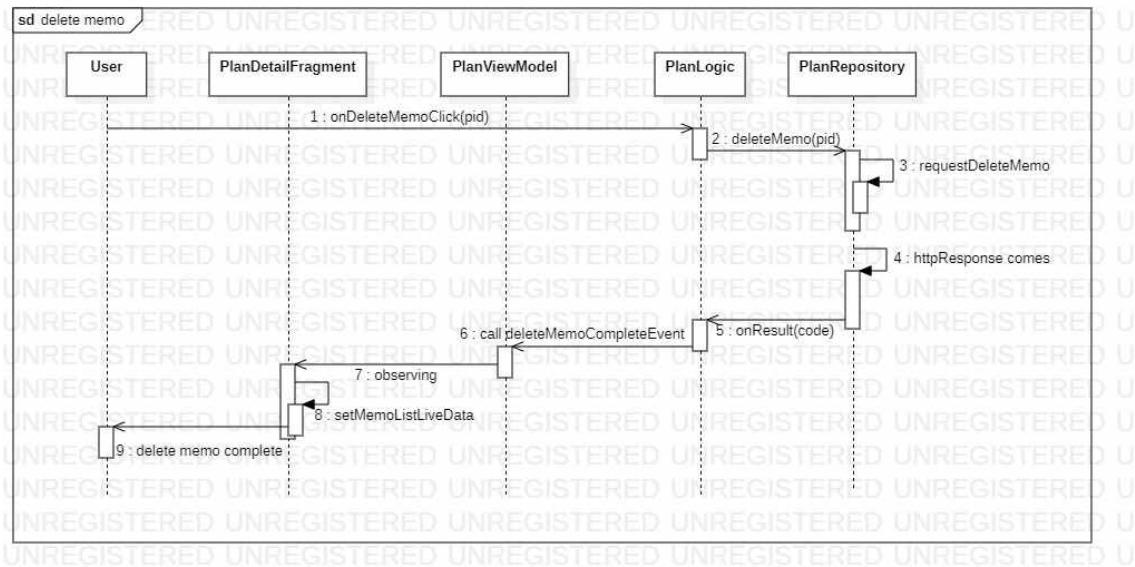


[그림 4-16] Create memo SD

사용자가 새로운 메모를 생성하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #6-1>의 경우이다.

사용자가 약속 상세 정보 화면에서 메모 등록 버튼을 누르면 기능을 시작한다. 메모 등록 다이알로그로 전환하고 사용자가 메모를 작성한 후 등록 버튼을 누르면 메모를 등록한다.

Delete memo



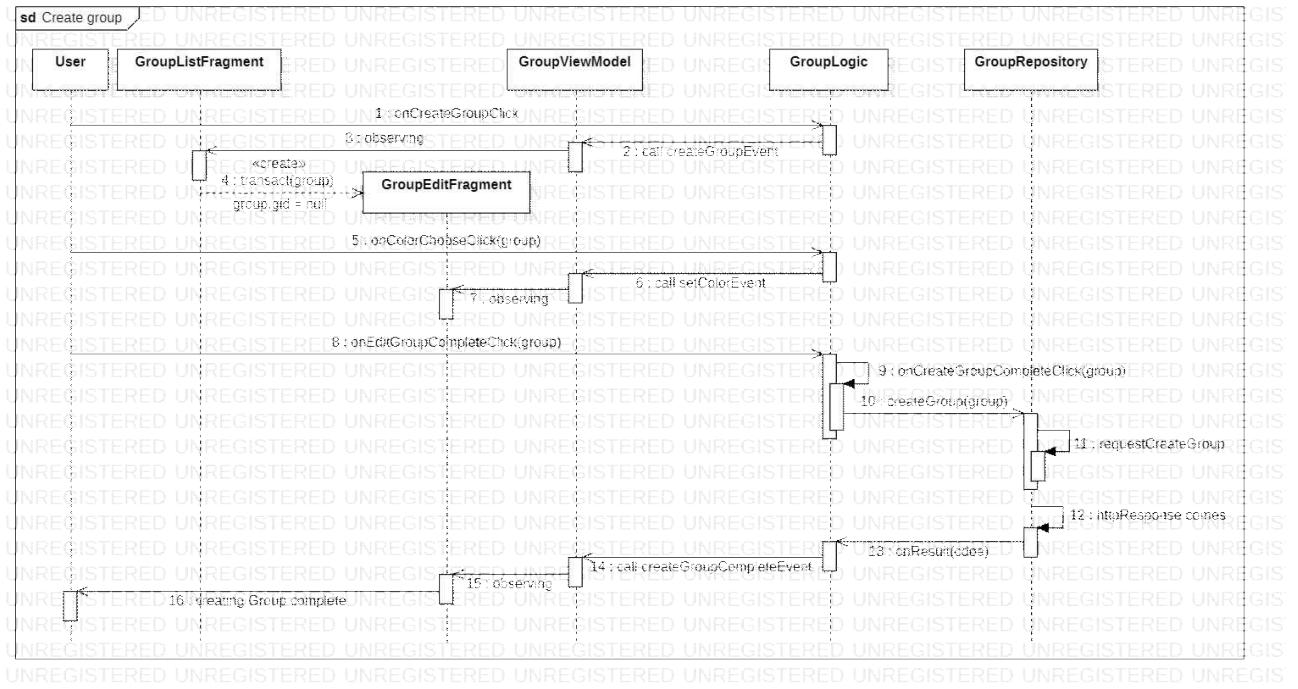
[그림 4-17] Delete memo SD

사용자가 생성된 메모를 삭제하는 use case를 나타내는 sequence diagram이다. Plan class description에서 <Use case #6-5>의 경우이다.

사용자가 약속 상세 정보 화면에서 자신의 메모에 있는 메모 삭제 버튼을 누르면 서버에 메모 삭제 요청을 보낸다.

4.3 Group sequence diagram

Create group

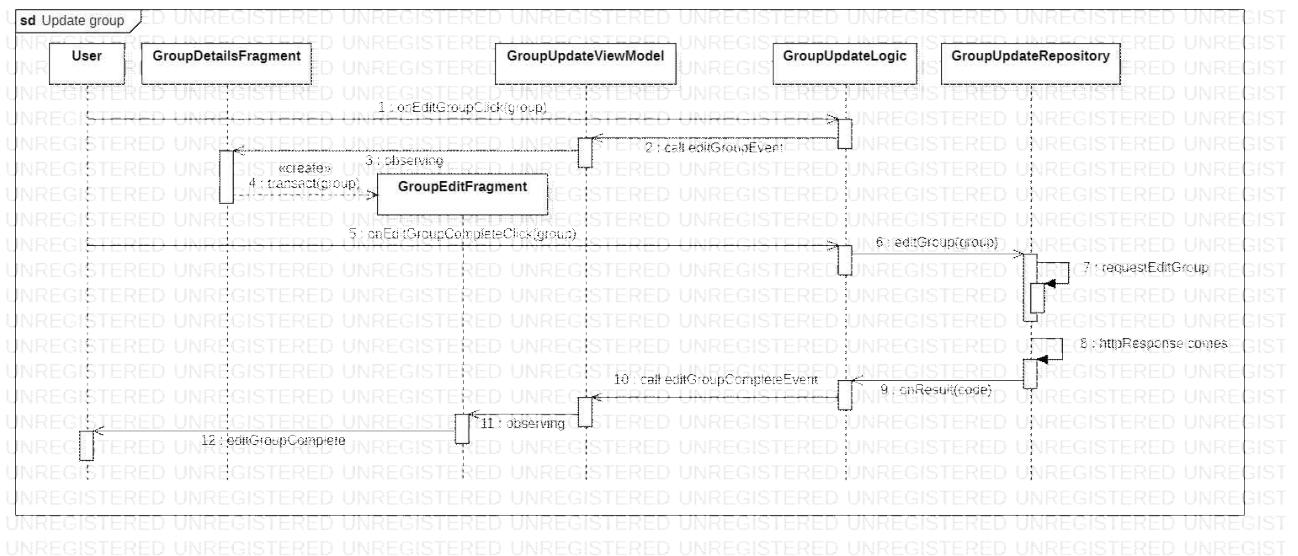


[그림 4-18] Create group SD

사용자가 신규 그룹을 생성하는 use case를 나타내는 sequence diagram이다. Group class diagram에서 <Use case #7-4>의 경우이다.

사용자가 그룹 생성 버튼을 누름으로써 시작한다. 그룹 생성 화면으로 전환 후 사용자는 그룹 이름, 설명을 작성하고 그룹 색상을 선택한다. 완료 버튼을 누르면 서버에 요청을 보내 그룹을 생성한다.

Update group

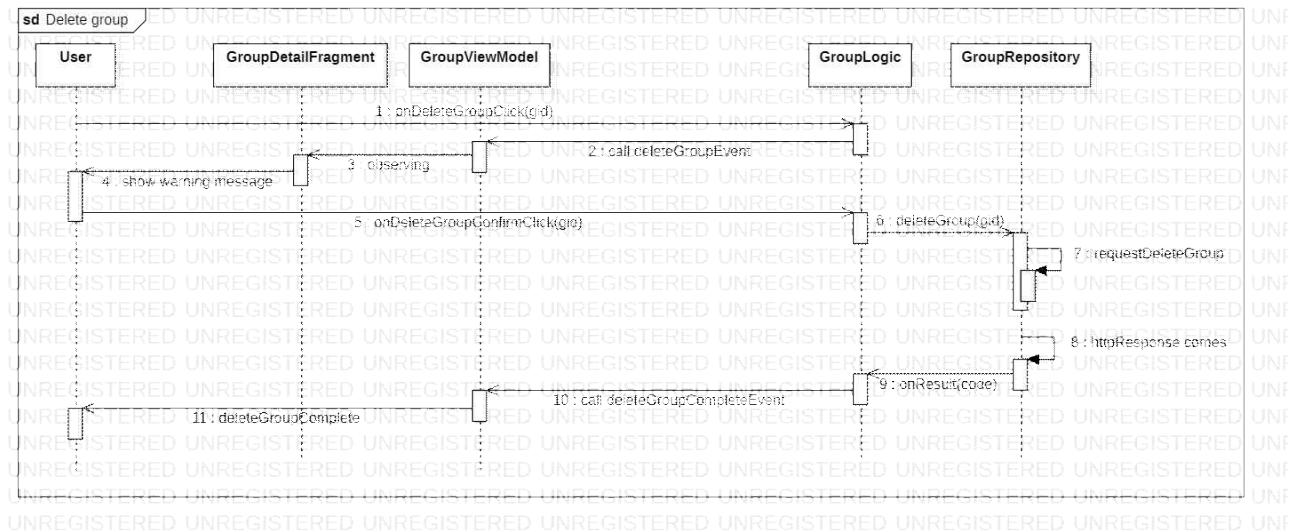


[그림 4-19] Update group SD

사용자가 기존 그룹을 수정하는 use case를 나타내는 sequence diagram이다. Group class diagram에서 <Use case #7-7>의 경우이다.

사용자가 그룹 상세 정보 화면에서 그룹 수정 버튼을 누르면 기능은 시작된다. 그룹 수정 화면으로 전환 후 사용자는 그룹의 정보를 수정한다. 사용자가 완료 버튼을 누르면 서버에 요청을 보내 그룹 정보를 수정한다.

Delete group

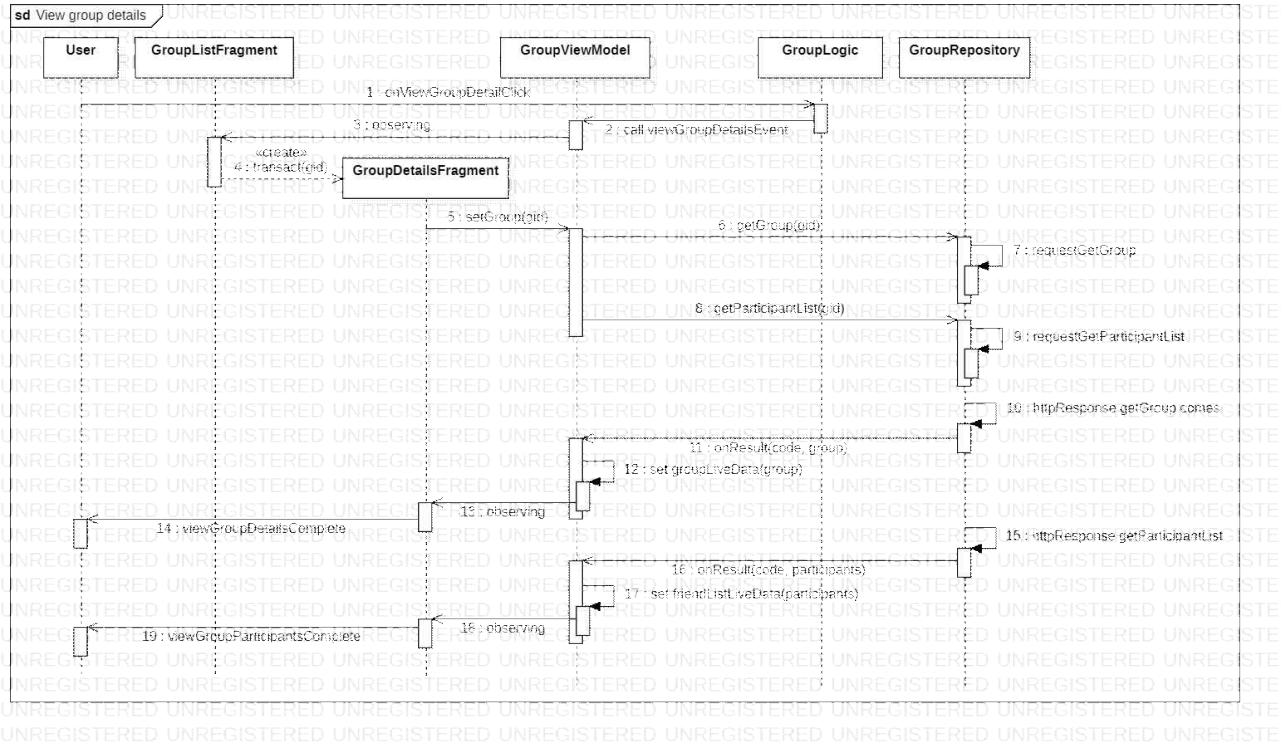


[그림 4-20] Delete group SD

사용자가 그룹을 삭제하는 use case를 나타내는 sequence diagram이다. Group class diagram에서 <Use case #7-4a. 그룹 삭제>의 경우이다.

그룹의 주인인 사용자가 그룹 상세 정보 화면에서 삭제 버튼을 누르면 기능을 시작한다. 사용자에게 경고 메시지를 출력하고 사용자가 확인 버튼을 누르면, 서버에 그룹 삭제 요청을 하고 변동 내용을 바꿔 출력한다.

View group details

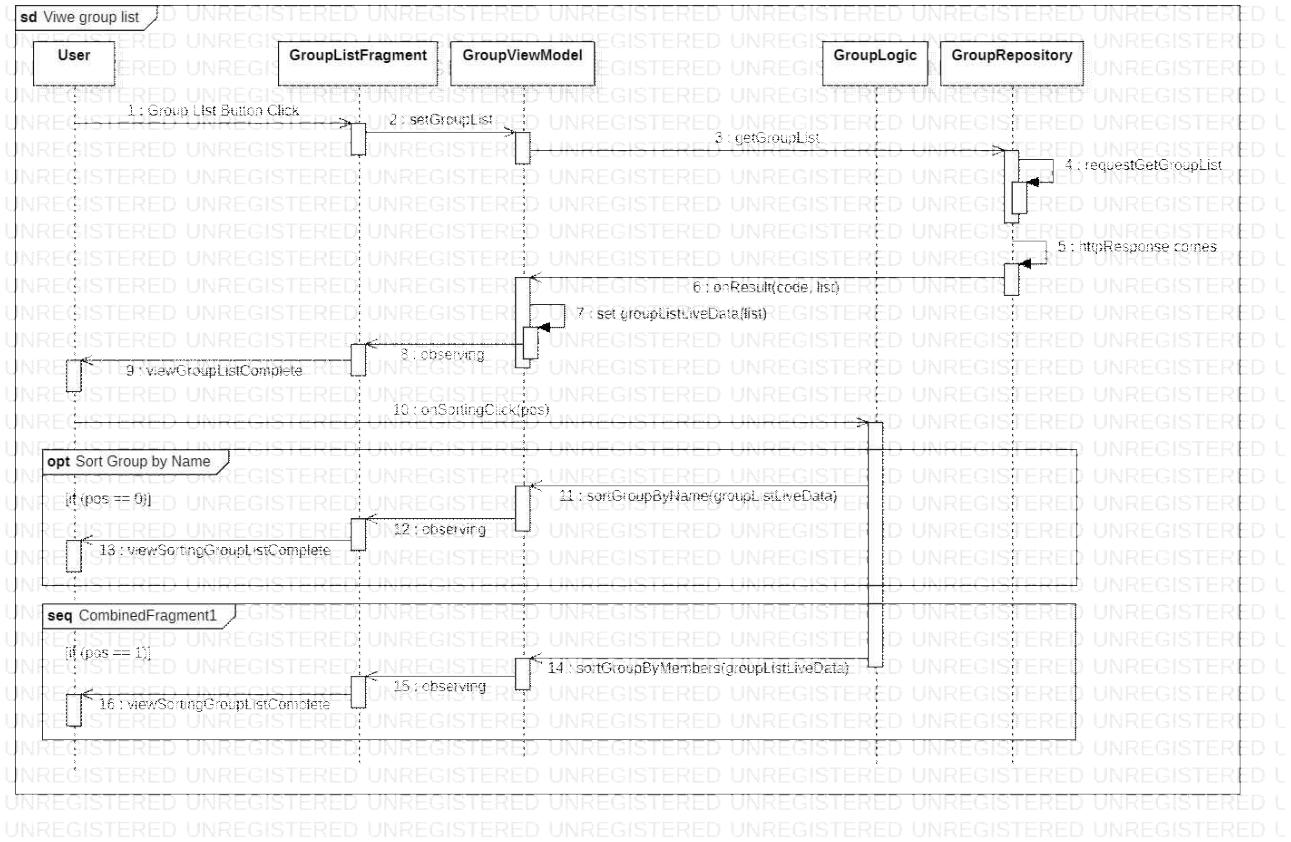


[그림 4-21] View group details SD

사용자가 그룹의 상세 정보를 열람하는 use case를 나타내는 sequence diagram이다. Group class diagram에서 <Use case #7-7>의 경우이다.

사용자가 그룹 목록에서 특정 그룹을 누르면 기능을 시작한다. 그룹의 상세 정보 화면으로 전환되면서 서버에 요청을 보내 그룹 이름, 그룹 인원 수, 그룹 설명, 그룹 색상 정보를 가져오고 사용자에게 제공한다.

View group list

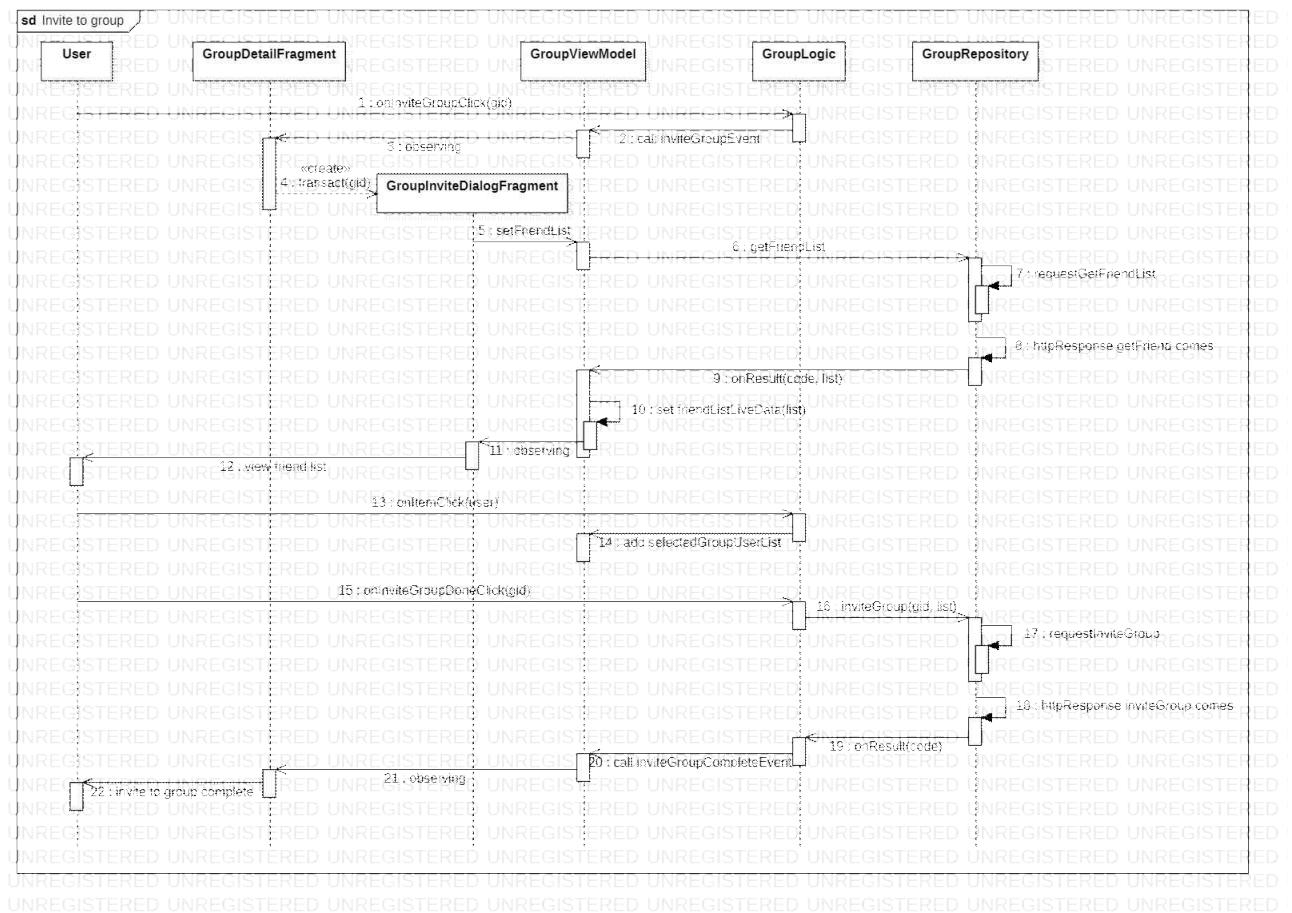


[그림 4-22] View group list SD

사용자가 현재 속한 그룹의 목록을 열람하는 use case를 나타내는 sequence diagram이다. Group class diagram에서 <Use case #7>의 경우이다.

그룹 목록 버튼을 누르면 기능이 시작된다. 그룹 목록 화면으로 전환되면서 서버에 요청을 보내 사용자가 속한 그룹 목록 정보를 가져오고 사용자에게 제공한다. 또한 사용자는 정렬 기준을 골라 이름순, 인원수순으로 정렬된 그룹 목록을 볼 수 있다.

Invite to Group

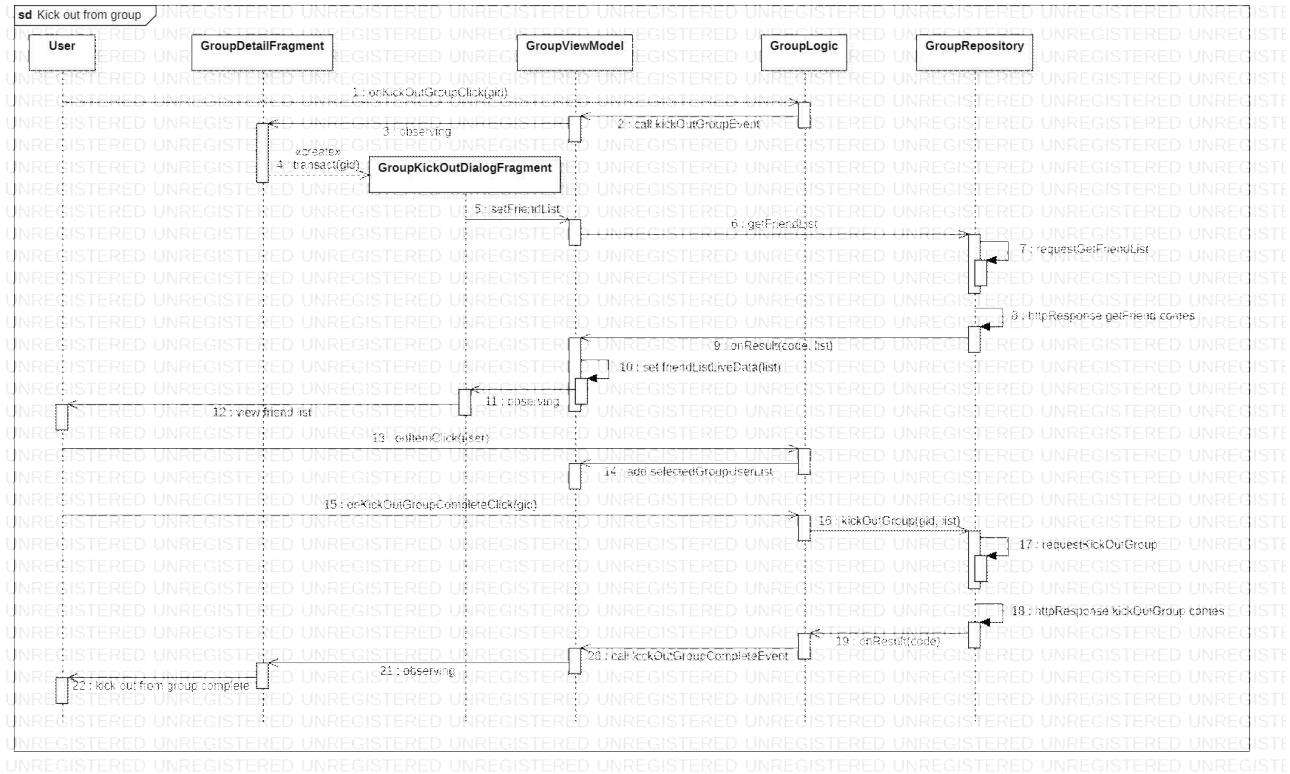


[그림 4-23] Invite to group SD

사용자가 속한 그룹에 다른 사용자를 초대하는 use case를 나타내는 sequence diagram이다. Group class diagram에서 <Use case #7-5a>의 경우이다.

사용자가 그룹 상세 정보 화면의 세부메뉴에서 친구 초대 버튼을 누르면 기능이 시작된다. 그룹 초대 화면으로 전환되면서 서버에 요청을 보내 사용자의 친구 목록을 가져오고 사용자에게 제공한다. 사용자가 초대할 친구를 선택해 초대 버튼을 누르면 서버에 요청을 보내 해당 사용자를 그룹에 초대한다.

Kick out from Group

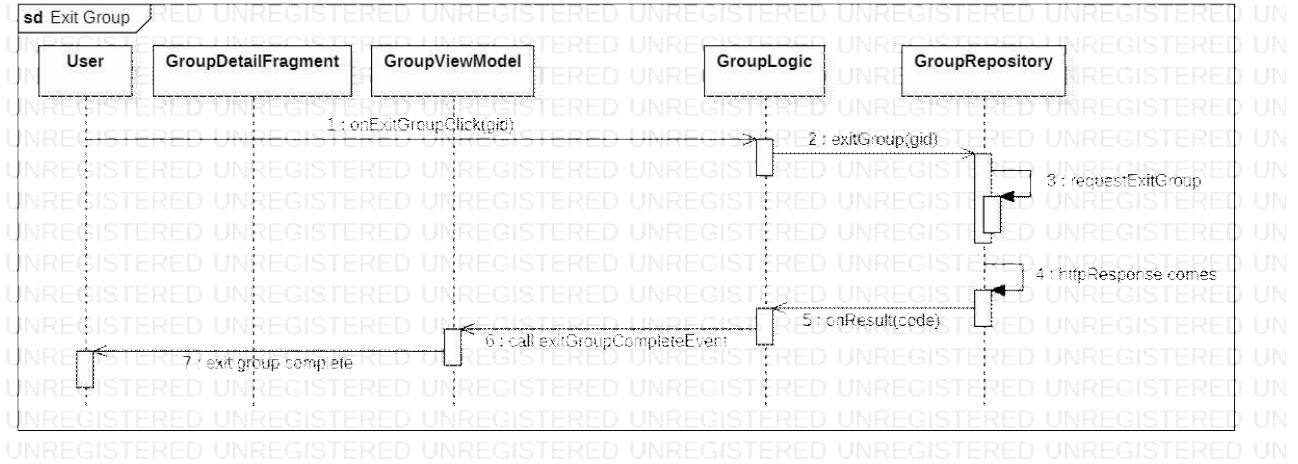


[그림 4-24] Kick out from group SD

사용자(그룹 주인)의 그룹에 소속된 다른 사용자를 강제 퇴장시키는 use case를 나타내는 sequence diagram이다. Group class diagram에서 <Use case #7-5b>의 경우이다.

사용자가 그룹 상세 정보 화면의 세부메뉴에서 강제 퇴장 버튼을 누르면 기능이 시작된다. 그룹 강제 퇴장 화면으로 전환되면서 서버에 요청을 보내 사용자의 친구 목록을 가져오고 사용자에게 제공한다. 사용자가 퇴장시킬 친구를 선택해 강제 퇴장 버튼을 누르면 서버에 요청을 보내 해당 사용자를 그룹으로부터 퇴장시킨다.

Exit group



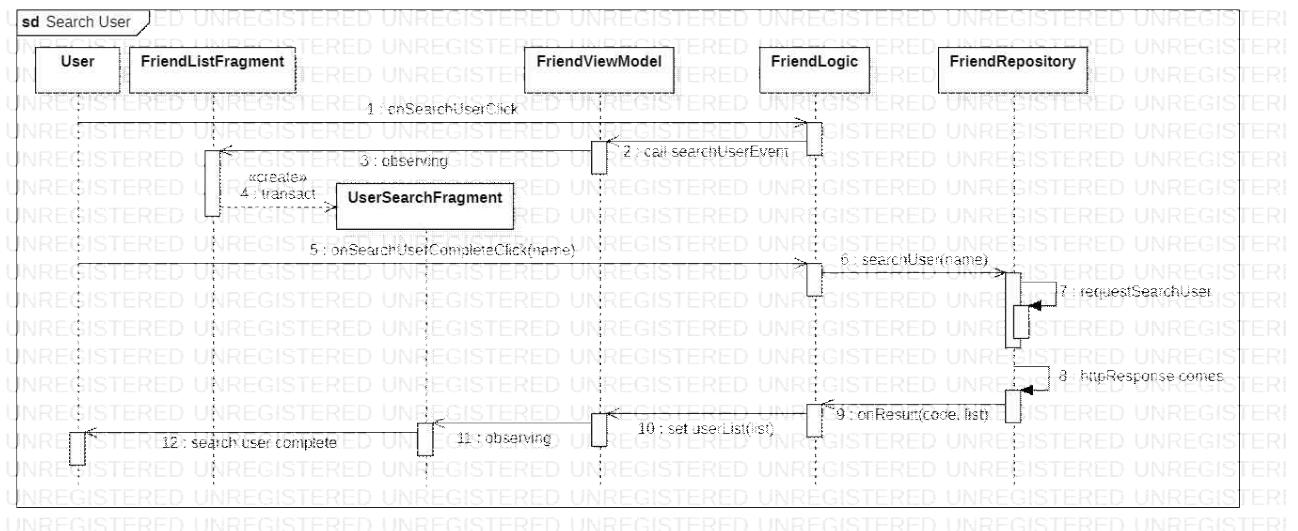
[그림 4-25] Exit group SD

사용자가 그룹을 탈퇴하는 use case를 나타내는 sequence diagram이다. Group class diagram에서 <Use case #7-4b. 그룹 삭제>의 경우이다.

사용자가 그룹 상세 정보 화면에서 탈퇴 버튼을 누르면 기능을 시작한다. 서버에 그룹 탈퇴 요청을 하고 변동 내용을 바꿔 출력한다.

4.4 Friend sequence diagram

Search user



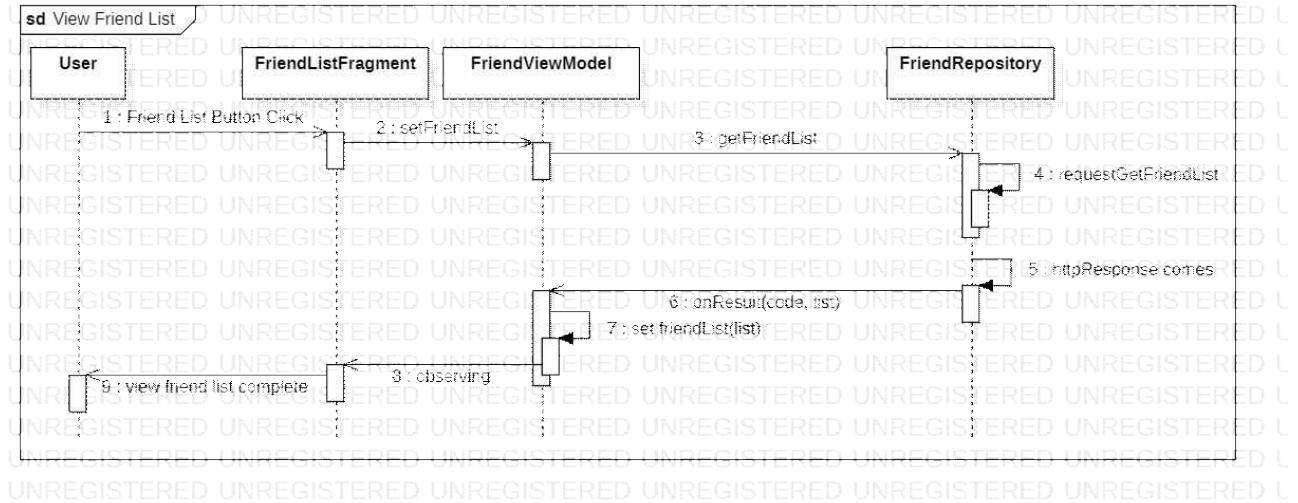
[그림 4-26] Search user SD

사용자가 다른 사용자를 검색하는 use case를 나타내는 sequence diagram이다. User class diagram에서 <Use case #10>의 경우이다.

사용자가 친구 목록 화면의 세부 메뉴 중 친구 추가 버튼을 눌러 사용자 검색 화면으

로 넘어갈 수 있다. 이후 사용자 검색 화면에서 검색할 사용자의 아이디나 닉네임을 입력한 후 검색 버튼을 누르면 기능을 시작한다. 서버에 사용자 검색 요청을 하고 검색된 사용자 목록을 가져와 사용자에게 제공한다.

View friend list

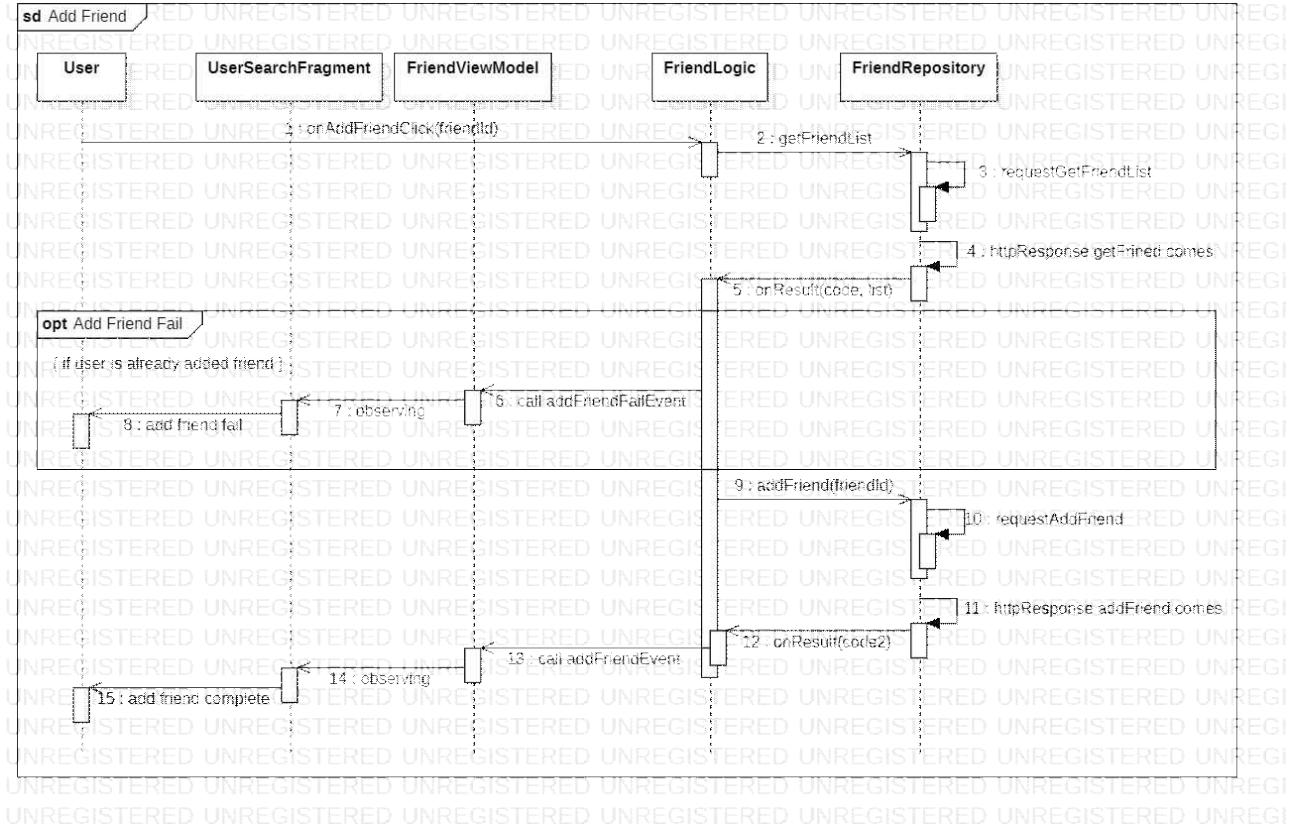


[그림 4-27] View friend list SD

사용자가 친구 목록을 열람하는 use case를 나타내는 sequence diagram이다. Friend class diagram에서 <Use case #5-4>의 경우이다.

사용자가 하단 메뉴에서 친구 목록 버튼을 누르면 기능을 시작한다. 친구 목록 화면으로 전환하고 서버에 친구 목록을 요청하여 사용자의 친구 목록을 갱신받아 화면에 적용한다.

Add friend

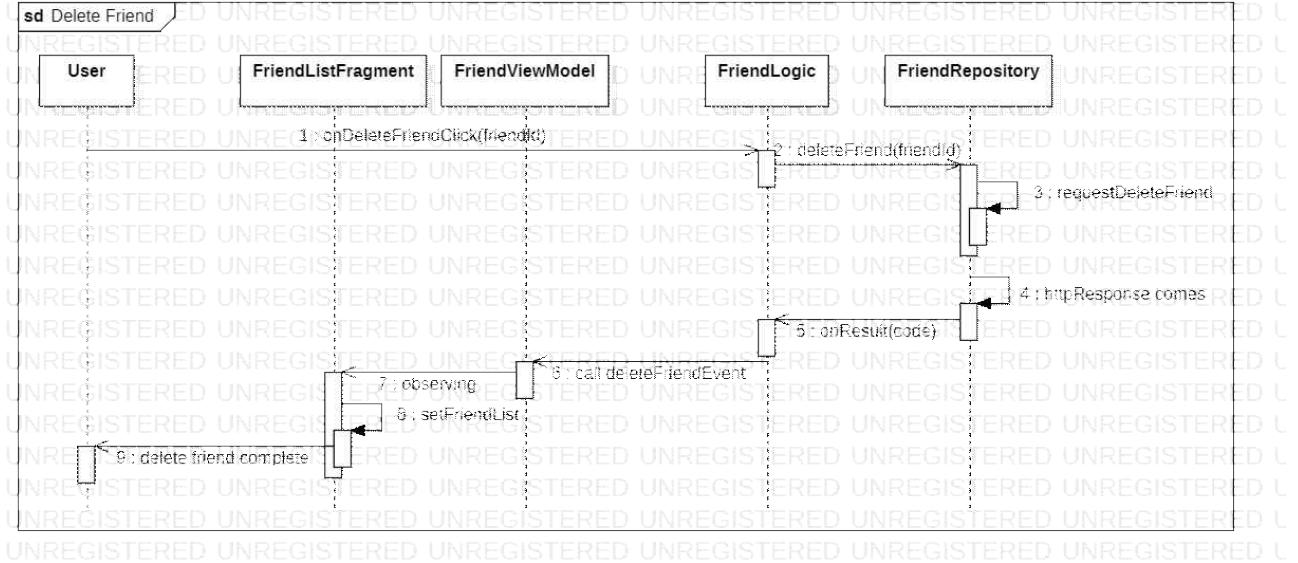


[그림 4-28] Add friend SD

사용자가 새 친구를 추가하는 use case를 나타내는 sequence diagram이다. Friend class diagram에서 <Use case #5-2>의 경우이다.

사용자가 사용자 검색 화면에서 특정 사용자를 골라 추가 버튼을 누르면 기능을 시작 한다. 서버에 친구 추가 요청을 하고 사용자의 친구 목록에 변동 사항을 적용한다. 만약 이미 친구 관계인 사용자를 친구로 추가하려 한다면 에러 메시지를 출력한다.

Delete friend

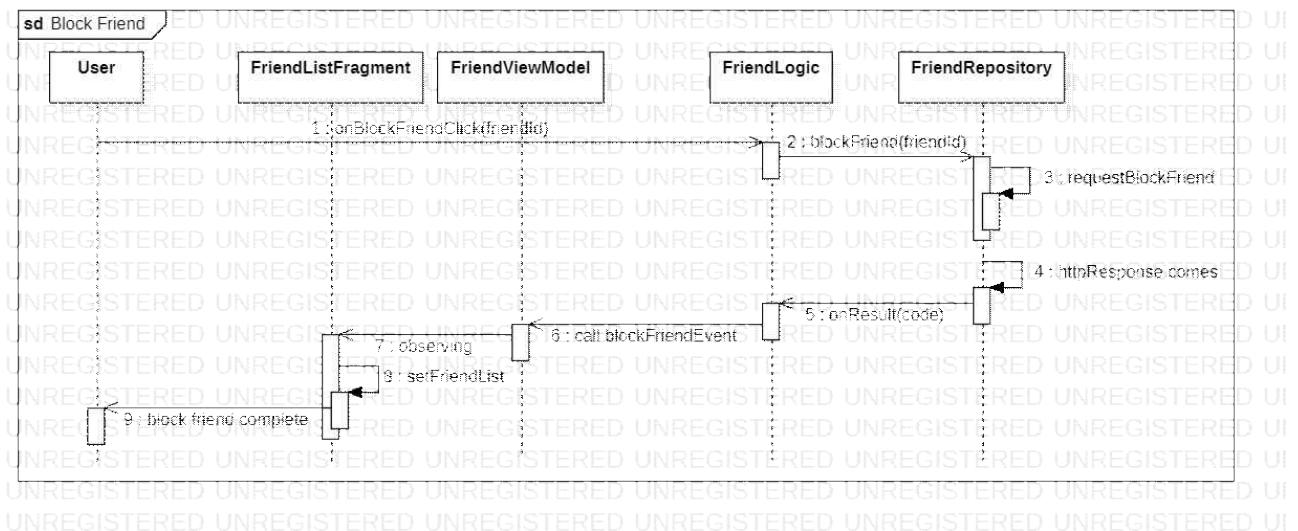


[그림 4-29] Delete friend SD

사용자가 기존의 친구를 삭제하는 use case를 나타내는 sequence diagram이다. Friend class diagram에서 <Use case #5-4b. 친구 삭제>의 경우이다.

사용자가 친구 목록 화면에서 삭제할 친구를 고른 후 삭제 버튼을 누르면 서버에 친구 삭제 요청을 하고 사용자의 친구 목록에서 해당 사용자가 삭제된다.

Block friend



[그림 4-30] Block friend SD

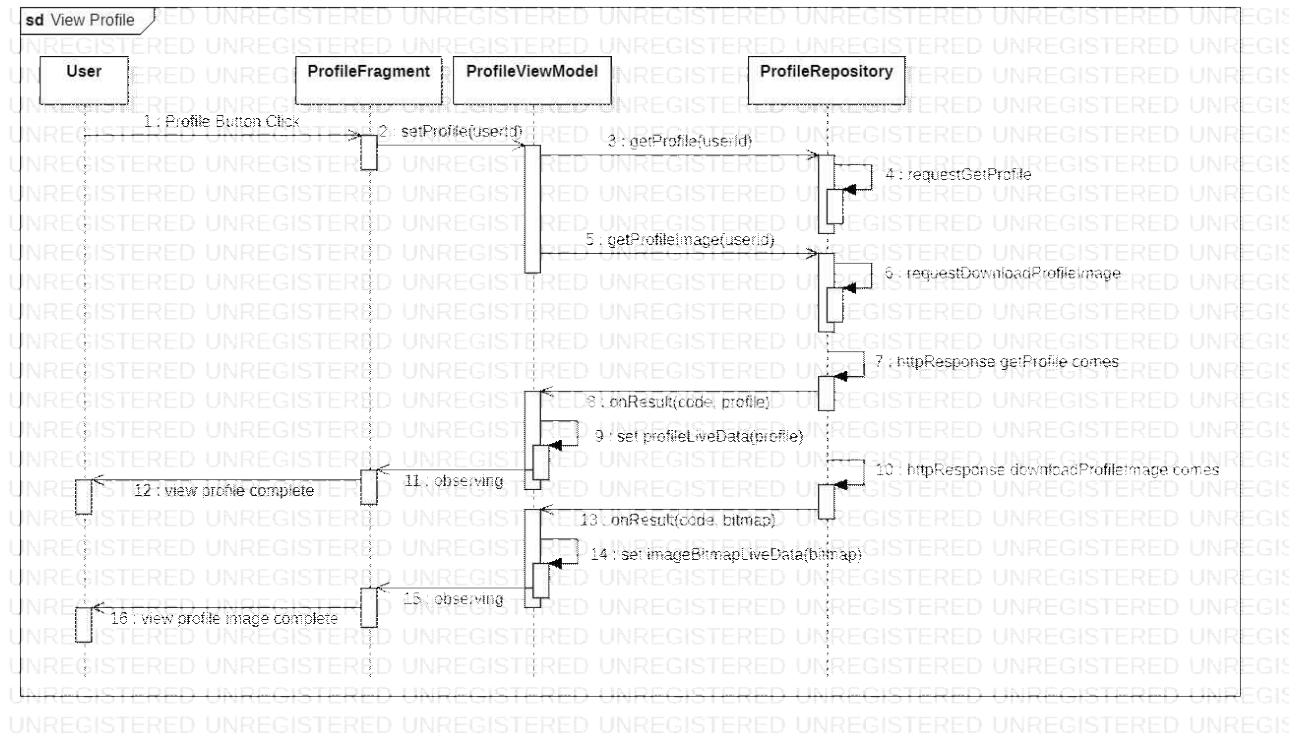
사용자가 기존의 친구를 차단하는 use case를 나타내는 sequence diagram이다. Friend class diagram에서 <Use case #5-4a. 친구 차단>의 경우이다.

사용자가 친구 목록 화면에서 차단할 친구를 고른 후 차단 버튼을 누르면 서버에 친

구 차단 요청을 하고 사용자의 친구 목록에서 해당 사용자가 삭제되고 차단 처리된다.

4.5 Profile sequence diagram

View profile

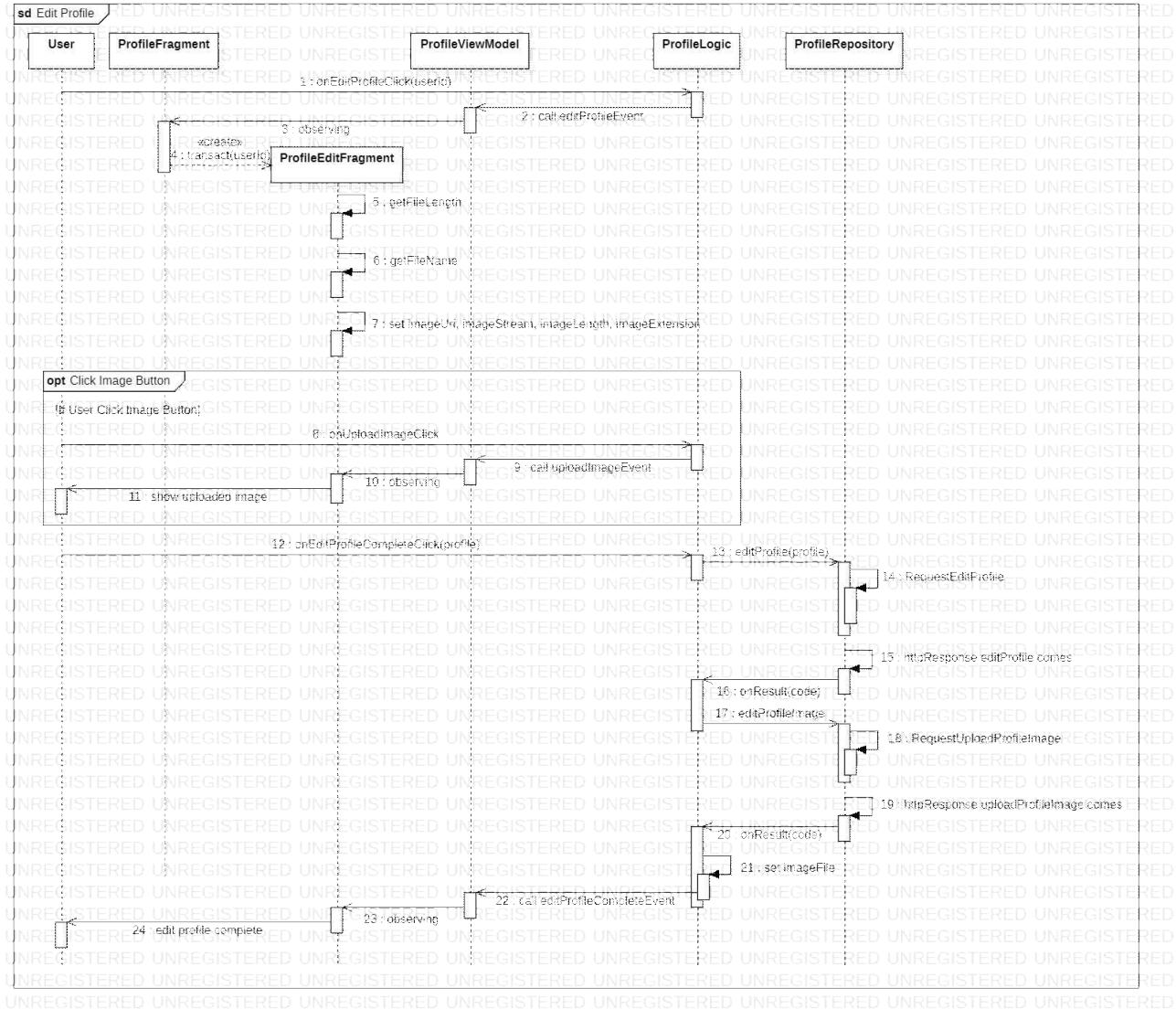


[그림 4-31] View profile SD

사용자가 자신의 프로필을 확인하는 use case를 나타내는 sequence diagram이다. Profile class diagram에서 <Use case #8-6>의 경우이다.

사용자가 하단 메뉴의 마이 페이지 버튼을 누르면 기능을 시작한다. 사용자의 프로필 정보를 확인할 수 있는 프로필 화면으로 전환하고, 서버에서 프로필 정보를 가져와 화면에 출력한다.

Edit profile

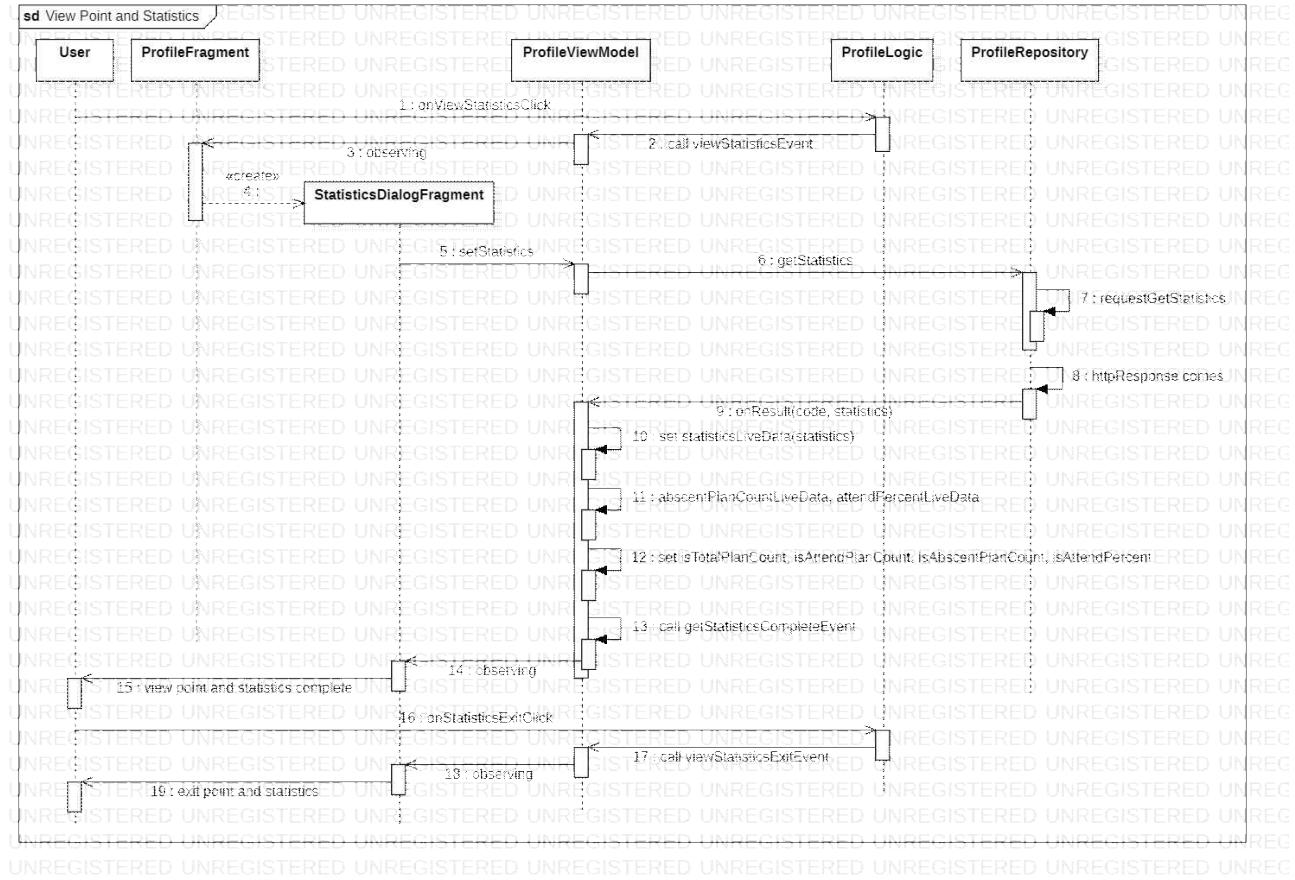


[그림 4-32] Edit profile SD

사용자가 자신의 프로필 정보를 수정하는 use case를 나타내는 sequence diagram이다. Profile class diagram에서 <Use case #8-5>의 경우이다.

사용자가 프로필 화면에서 프로필 수정 버튼을 누르면 기능을 시작한다. 프로필 수정 화면으로 전환하고 프로필 정보를 수정한 후 수정 버튼을 누르면 서버에 프로필 수정을 요청한다. 또한 사용자는 프로필을 수정할 때 ImageView를 눌러 프로필 사진을 업로드할 수 있다.

View point and statistics

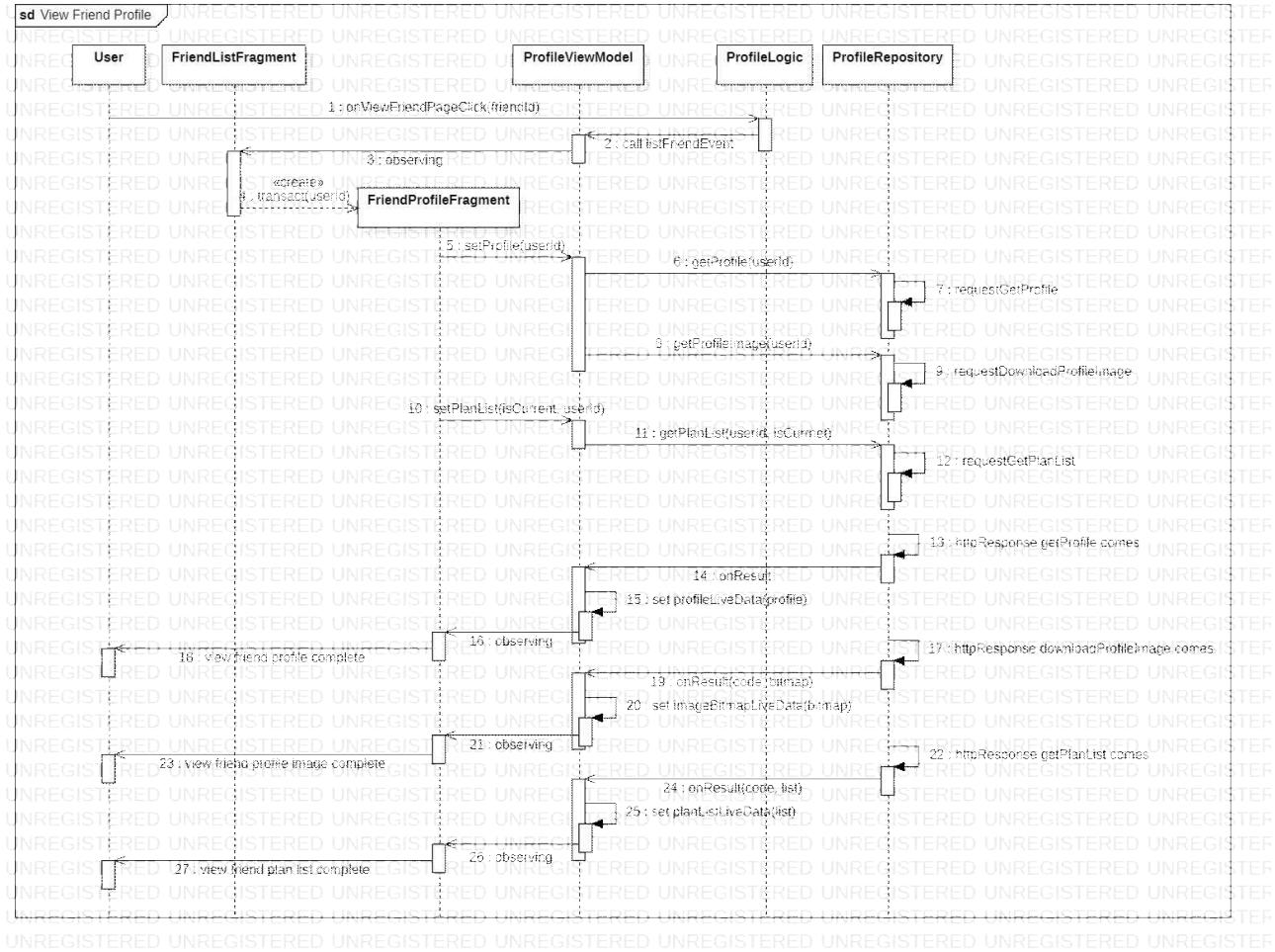


[그림 4-33] View point and statistics SD

사용자가 자신의 약속 관련 통계와 포인트를 확인하는 use case를 나타내는 sequence diagram이다. Profile class diagram에서 <Use case #8-4>의 경우이다.

사용자가 프로필 화면에서 통계 보기 버튼을 누르면 기능을 시작한다. 통계 화면으로 전환한다. View profile과 유사하게 서버에 통계 정보를 요청하고 응답을 받는 일련의 과정을 거친 뒤 서버로부터 받은 통계 정보를 화면에 출력한다.

View friend profile

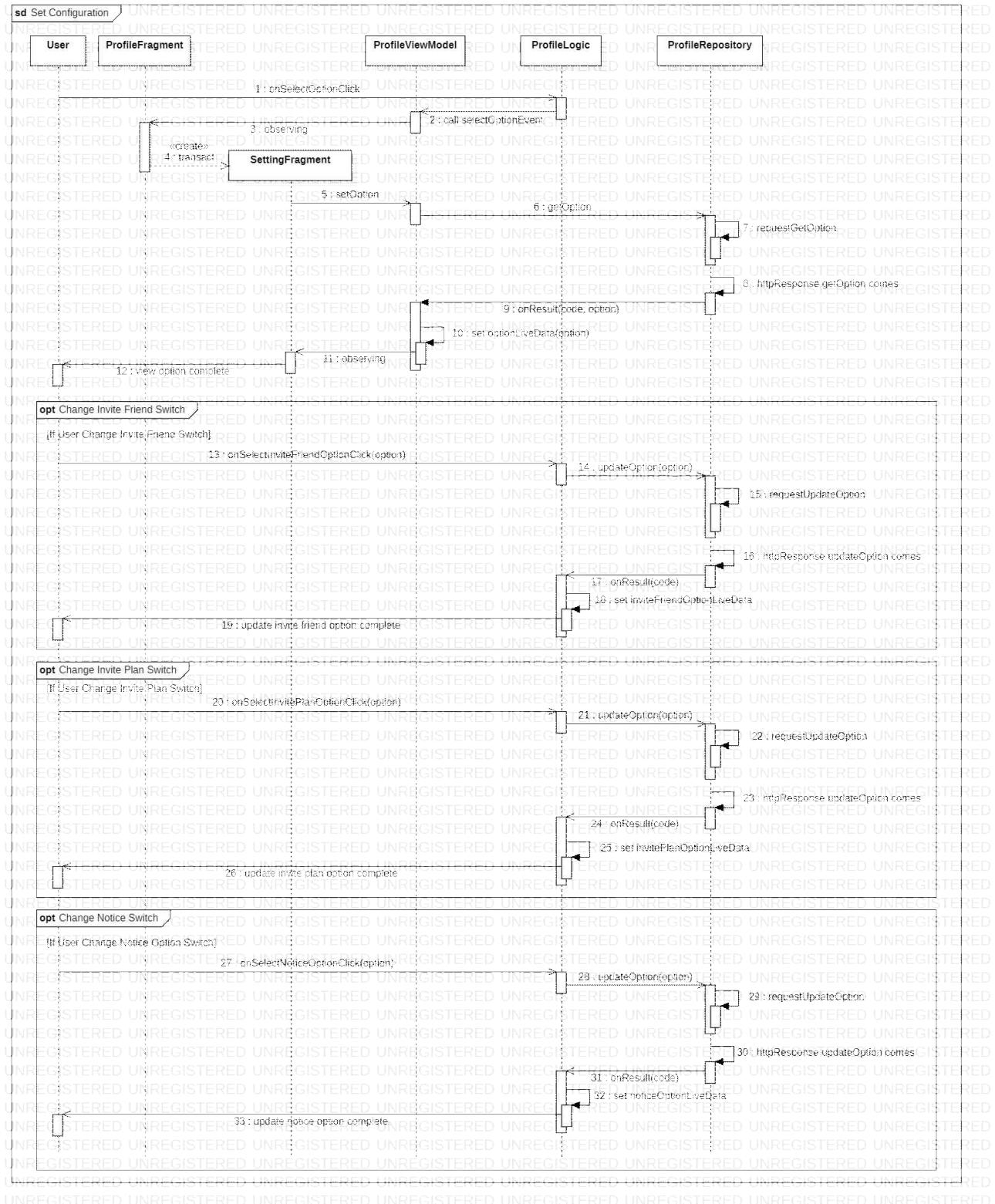


[그림 4-34] View friend profile SD

사용자가 다른 사용자의 프로필을 보는 use case를 나타내는 sequence diagram이다. Profile class diagram에서 <Use case #8-6a. 다른 사람의 프로필 보기>의 경우이다.

사용자가 친구 목록에 있는 사용자를 누르면 기능을 시작한다. 해당 사용자의 프로필 화면으로 전환하고 View Profile과 유사하게 서버에 프로필 정보와 약속 목록을 받아와 프로필 정보와 약속 목록을 화면에 출력한다.

Set configuration

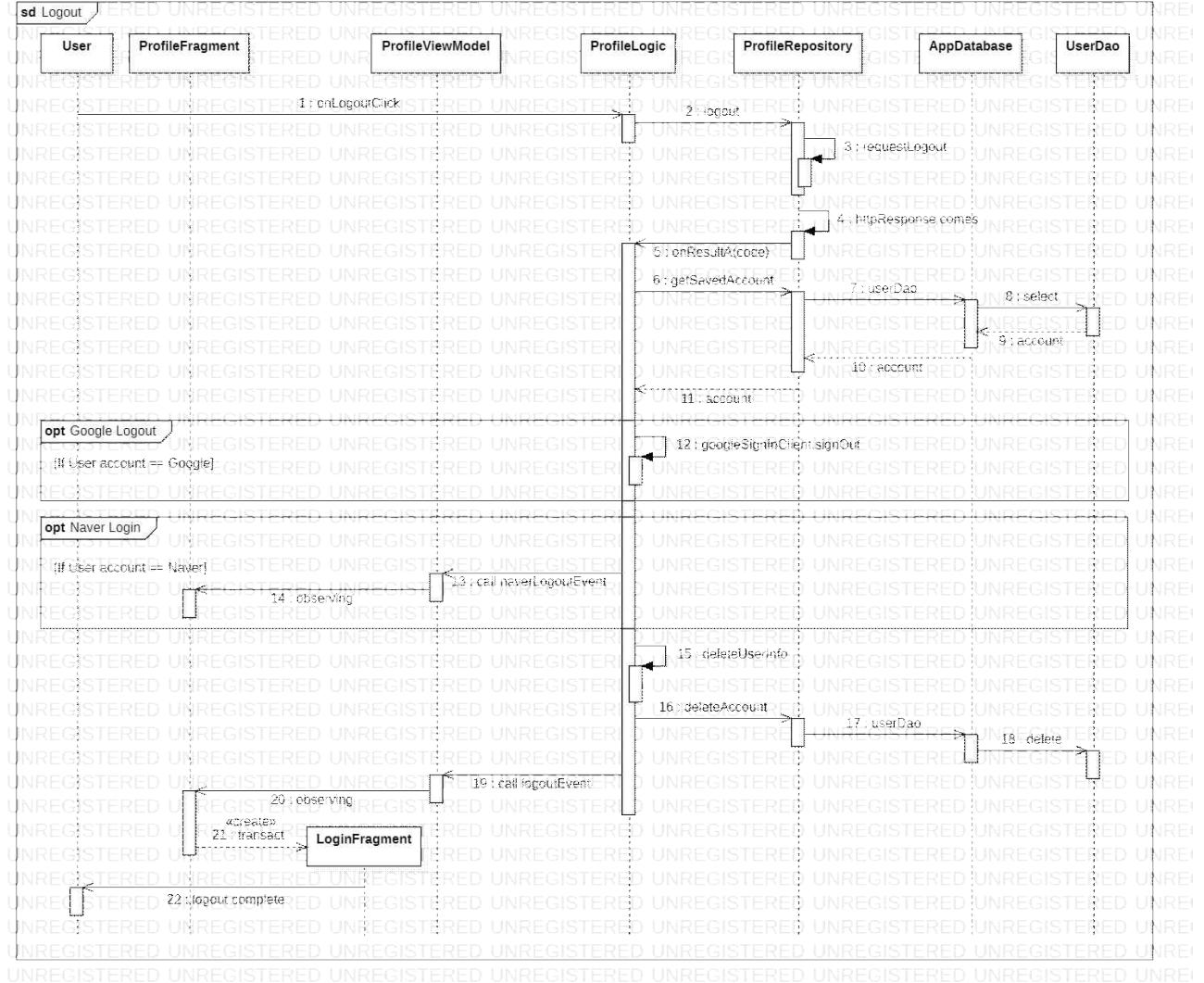


[그림 4-35] Set configuration SD

사용자가 알림, 초대 거부 등 시스템에서 제공하는 기능에 대한 설정을 변경하는 use case를 나타내는 sequence diagram이다. Setting class diagram에서 <Use case #9>의 경우이다.

사용자가 프로필 화면에서 설정 버튼을 누르면 기능을 시작한다. 설정 화면으로 전환하여 설정 화면에서는 푸시 알림, 약속 초대, 친구 초대에 대한 허용 여부를 변경할 수 있다. 각각의 설정은 스위치를 누른 순간 변경된 설정 정보가 서버에 반영되고 사용자는 화면에서 변경 여부를 바로 확인할 수 있다.

Logout

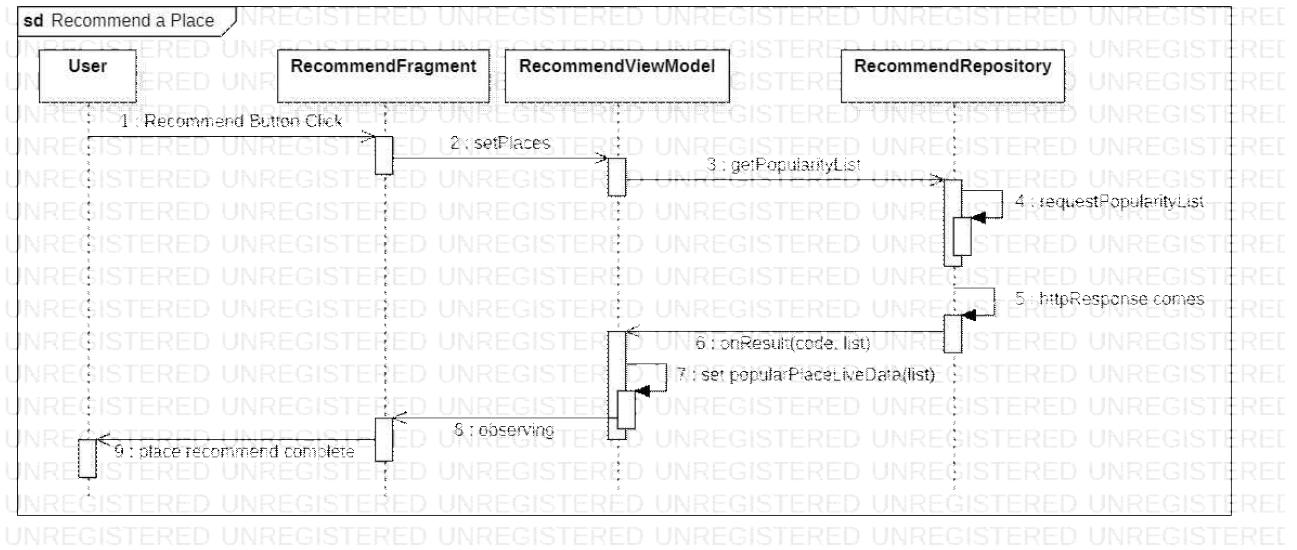


[그림 4-36] Logout SD

사용자가 시스템에서 로그아웃하는 use case를 나타내는 sequence diagram이다. 사용자가 프로필 화면에서 로그아웃 버튼을 누르면 기능을 시작한다. 서버에 로그아웃 요청을 보내고 로그아웃에 성공했을 경우 사용자의 화면을 로그인 화면으로 전환하고, 자동 로그인을 위한 계정 정보를 로컬 DB에서 삭제한다.

4.6 Recommend a place sequence diagram

Recommend a place



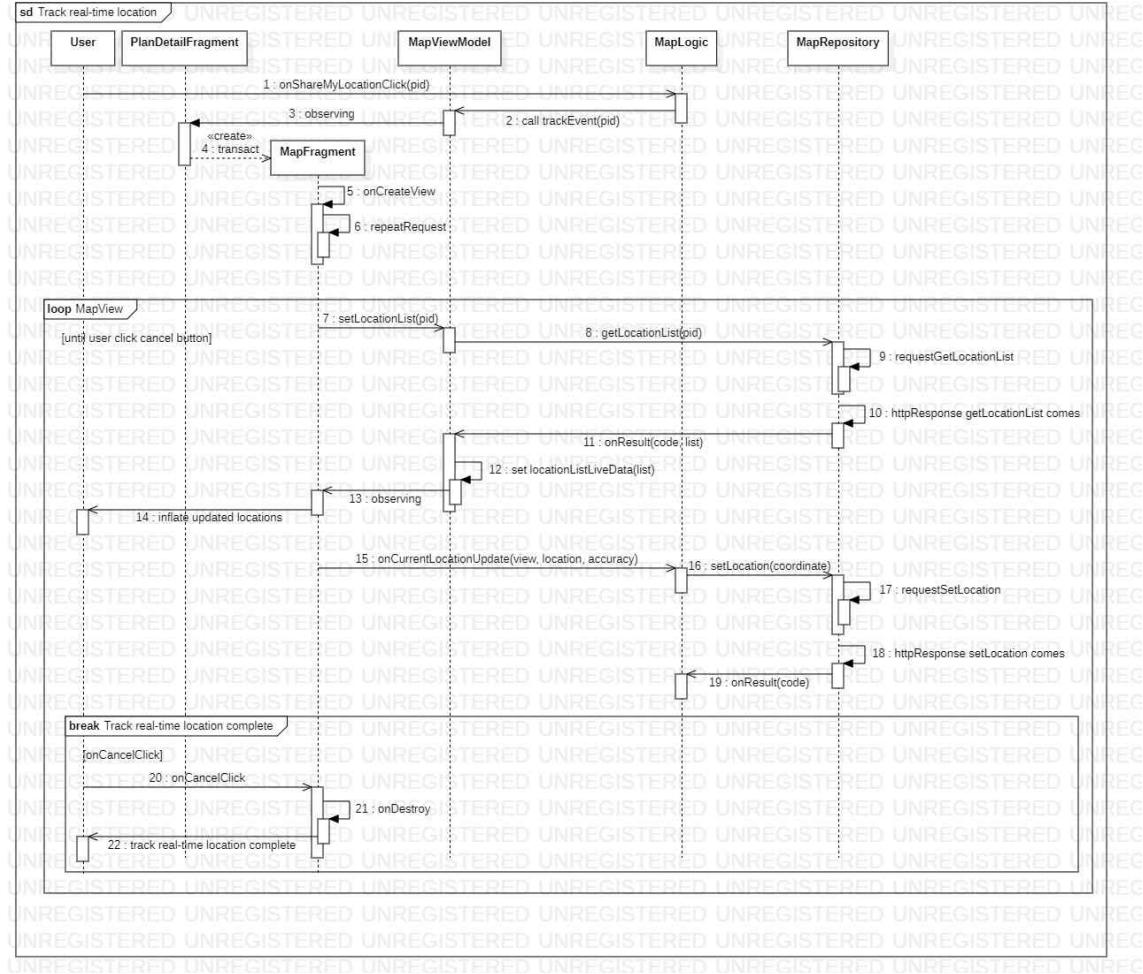
[그림 4-37] Recommend a place SD

사용자가 약속 장소를 추천받는 use case를 나타내는 sequence diagram이다. PlaceR ecommend class diagram에서 <Use case #12>의 경우이다.

사용자가 약속 목록 화면에서 약속 장소 추천 버튼을 누르거나 화면을 옆으로 슬라이드하면 기능을 시작한다. 약속 장소 추천 화면으로 전환한다. 그리고 서버에 인기 장소 정보를 받아올 것을 요청하고 인기 장소 정보를 받아오면 해당 장소들을 사용자에게 제공한다.

4.7 Track real-time location sequence diagram

Track real-time location



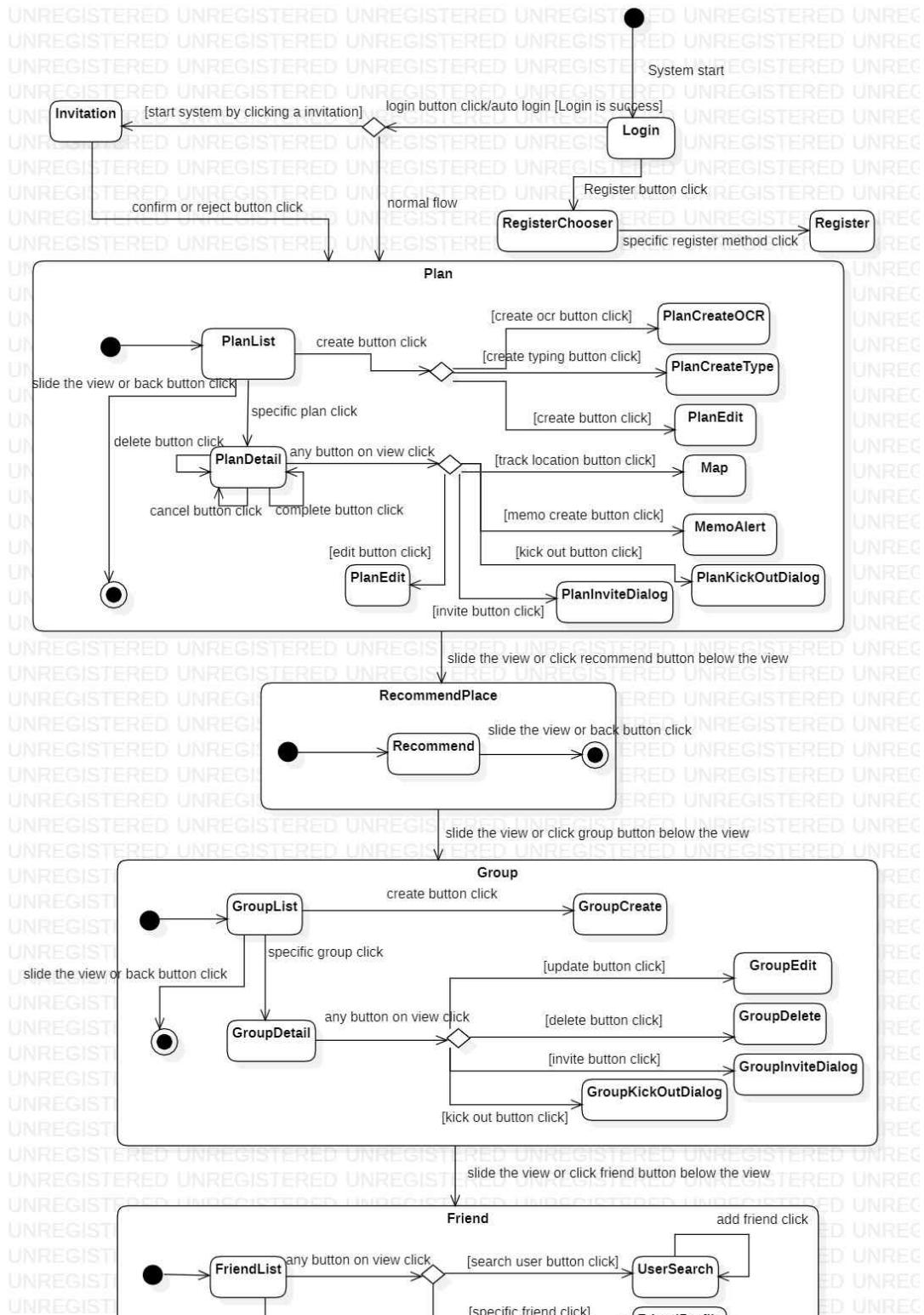
[그림 4-38] Track real-time location SD

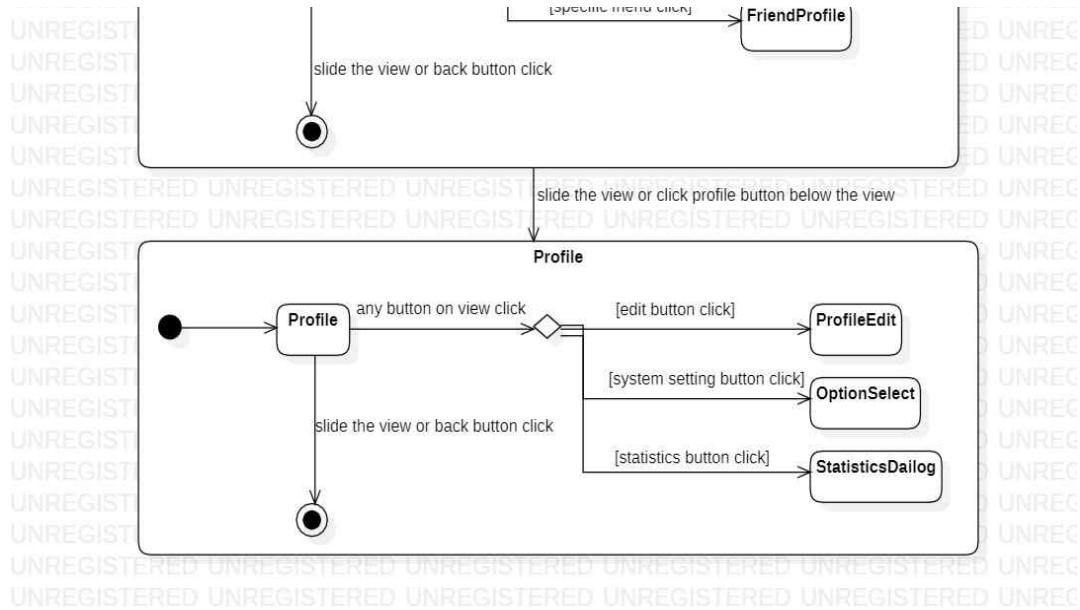
사용자가 자신의 위치와 참석자들의 현재 위치를 공유하고 예상 도착시간을 확인하는 use case를 나타내는 sequence diagram이다. Map class diagram에서 <Use case #1 3>의 경우이다.

사용자가 약속 상세 정보 화면에서 위치 공유 버튼을 누르면 기능을 시작한다. 지도 화면으로 전환하고 사용자가 뒤로 가기 버튼을 누르기 전까지 위치 정보와 도착 예정 시간을 지속적으로 사용자에게 제공한다. 서버에 자신의 위치 정보 저장과 약속 참석자들의 위치 정보를 받아오면 약속 참석자들의 예상 도착 시간을 서버에서 다시 받아오는 일련의 요청-응답에 대한 반복 과정을 사용자가 중단하기 전까지 계속 반복하면서 위치 정보와 도착 예정 시간을 실시간으로 제공한다.

5. State machine diagram

이 장은 시스템의 state machine diagram(이하 SMD)을 그리고 설명한다. [그림 5-1]은 클라이언트 측의 시스템(안드로이드 어플리케이션)에 관한 diagram이다.

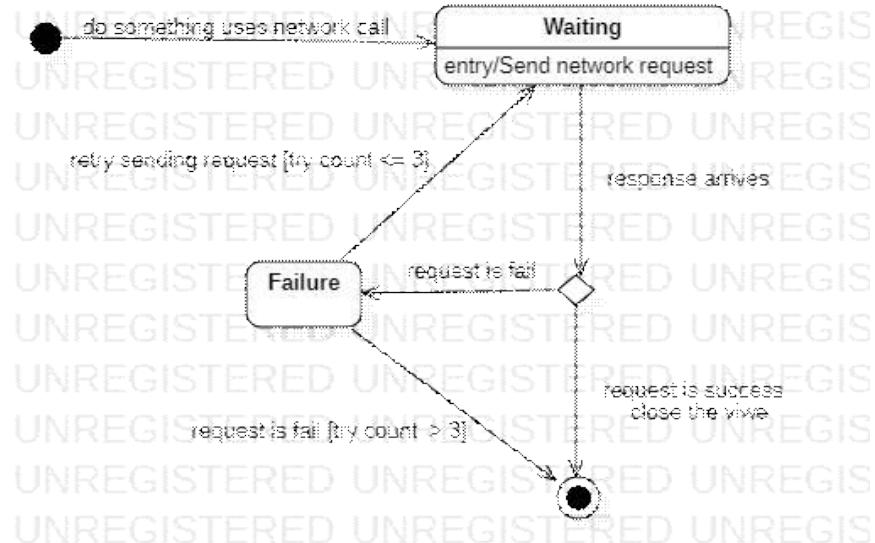




[그림 5-1] 어플리케이션 state machine diagram

이에 관한 설명은 다음과 같다.

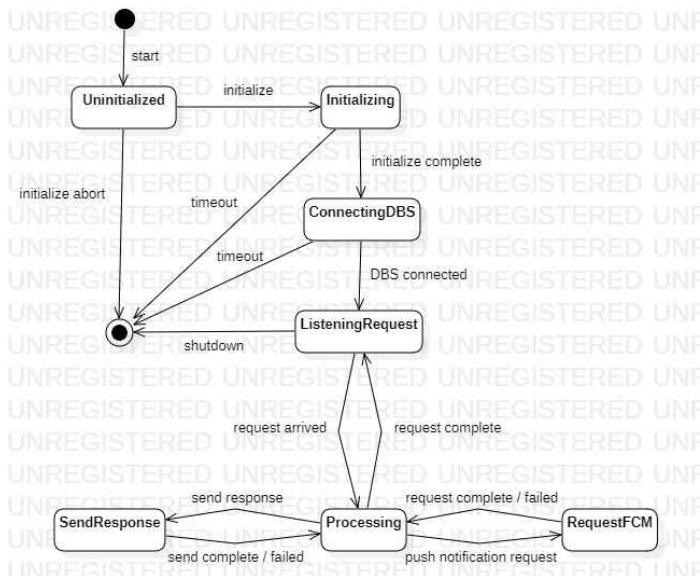
- 각 state는 안드로이드 어플리케이션이 사용자에게 어떤 것을 보여주고 있는지에 대한 상태이다. 따라서 state의 이름은 CD의 Fragment 클래스들과 1:1 대응된다.
- 본 SMD에는 5개의 composite state가 있다. 사용자는 시스템을 다른 composite state로 옮기기 위해 화면을 슬라이딩하거나 하단 버튼을 누를 수 있다.
 - 화면 슬라이딩은 각 state 내부의 end point와 연결된 state에서만 가능하다.
 - 하단 버튼 클릭은 SMD에 자세히 표현되지 않았지만 Dialog로 끝나는 state를 제외한 모든 state에서 실행 가능하다.(다른 composite state로 이동할 수 있다.)
- [그림 5-1]에서 상황에 따라 state 내부에서 네트워크 요청을 보낼 때가 있는데 이 때 공통된 성공 조건은 network response is valid이다. 요청을 보낼 때의 상태는 [그림 5-2]로 구체화했다. 다만 이는 Background에서 바뀌는 상태이므로 foreground 기준의 [그림 5-1]과 다른 life cycle을 가지며 그 결과는 [그림 5-1]에 반영된다.
 - 어떨 때 네트워크 요청을 보내는지는 SD에서 찾아볼 수 있다.



[그림 5-1] 네트워크 요청 시 시스템의 background를 묘사한 state machine diagram

- SM내의 어떤 한 state가 끝나거나 사용자가 뒤로가기 버튼을 누르면 바로 전 화면(state)로 돌아간다. 예를 들어, PlanCreateOCR이 끝나 성공적으로 약속이 생성되거나 중간에 뒤로가기 버튼을 누르면 PlanList로 돌아간다.
- 이 SM내에서 view란 사용자에게 보이는 화면을 의미한다.

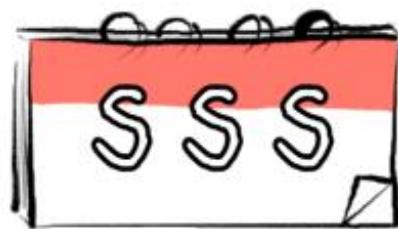
서버용 STD는 [그림 5-2]와 같다. 시스템 시작 후 초기화를 하고 요청을 기다리는 형태이다.



[그림 5-2] 서버의 STD

6. User interface prototype

이 장은 예상 UI와 UI안의 각 구성요소를 설명한다. 실제 개발된 어플의 UI와 디자인은 달라지지만 내용은 같다.



로그인



ID



Password

회원가입

로그인

다른 계정으로 로그인

N 로그인

Google

[그림 6-1] 로그인 화면

위 그림은 로그인 화면이다. 아이디와 비밀번호를 입력하여 로그인을 할 수 있고 네이버 또는 구글 로그인 버튼을 눌러 소셜 계정으로 로그인을 할 수 있다.

sss

아이디 _____

비밀번호 _____

비밀번호 확인 _____

닉네임 _____

회원가입

[그림 6-2] 회원가입 화면

위 그림은 회원가입 화면이다. 아이디, 비밀번호, 비밀번호 확인과 닉네임을 입력하여 회원가입 할 수 있다. 또한 구글 계정이나 네이버 계정을 사용하여 아이디와 닉네임만을 입력하여 회원가입 할 수 있다.

sss

정렬 기준 시간순 ▾

2021/11/05 12:00 ~ 13:00 반월당역	비공개
2021/10/31 13:30 ~ 18:00 영남대학교 IT관	비공개



약속 목록

[그림 6-3] 약속 목록 화면

위 그림은 약속 목록 화면이다. 정렬 기준을 선택하여 약속들을 시간순, 카테고리순으로 볼 수 있고 각 약속의 시작 시간, 약속 이름, 공개 여부를 확인할 수 있다. 또한 약속을 클릭하면 약속의 상세 정보 화면을 보여준다. 토글 버튼을 눌러 지난 약속을 확인할 수 있으며, 플로팅 버튼을 눌러 약속을 추가할 수도 있다.

sss

2021년 10월 31일

13:30 ~ 18:00

영남대학교 IT관

반복 여부: 반복 없음

카테고리: 카테고리 없음

그룹: Young University



[그림 6-4] 약속 상세 정보 화면

위 그림은 약속 상세 정보 화면이다. 약속 이름, 시작일과 종료일, 장소, 카테고리, 그룹, 그리고 작성된 메모와 약속 참석자 목록을 볼 수 있다. 플로팅 버튼을 눌러 약속 수정, 삭제, 완료, 친구 초대, 강제 퇴장, 참석 취소, 메모 등록, 위치 공유 기능을 이용할 수 있으며, 작성한 메모는 삭제 버튼을 눌러 삭제할 수 있다.



2021년 10월 31일

13 : 30

18 : 00

장소

반복 여부

반복 없음

카테고리

그룹

카테고리 없음

Young Unive..

 구글 캘린더 연동

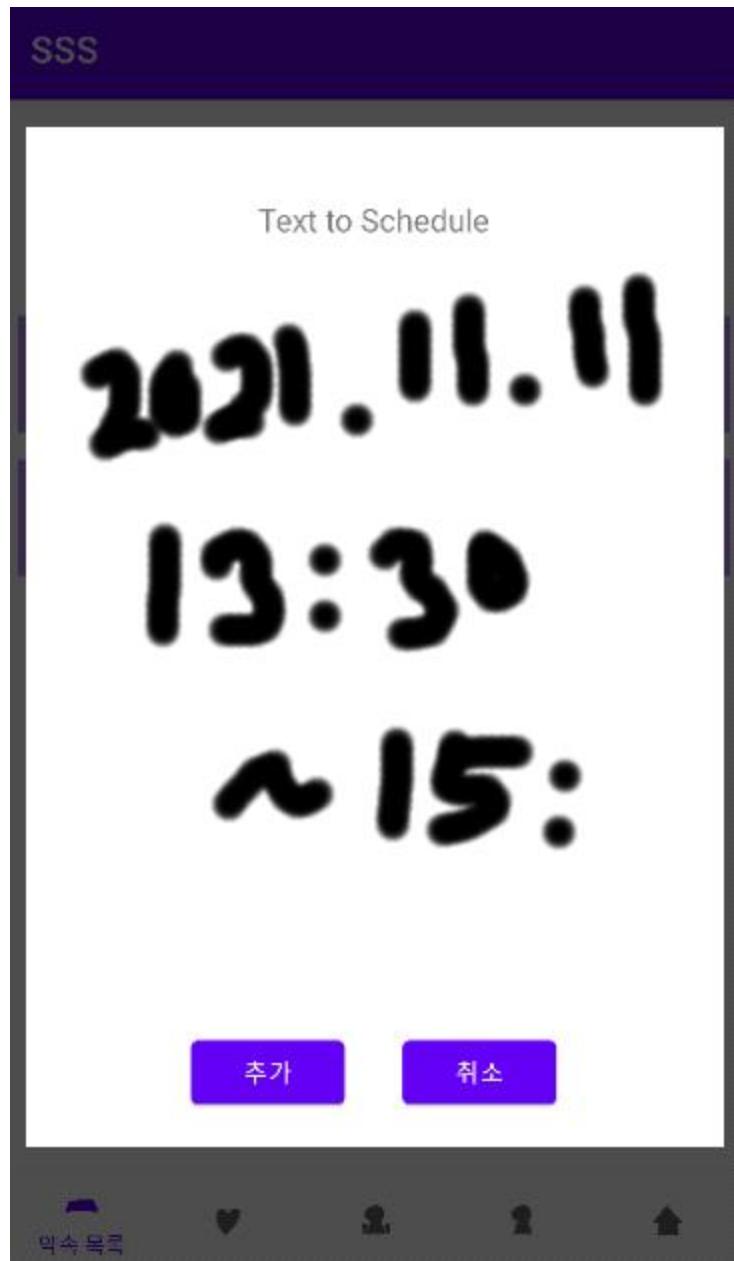
저장

약속 목록



[그림 6-5] 약속 작성 화면

위 그림은 약속 작성 화면이다. 약속 이름, 장소를 작성하고 카테고리와 그룹을 선택한다. 또한 시작일과 종료일 버튼을 누르면 날짜와 시간을 선택할 수 있는 다이얼로 그 화면이 열린다. 모든 내용을 작성한 후 저장 버튼을 누르면 약속이 등록된다. 약속 수정 기능을 통해 해당 화면으로 넘어왔다면 해당 약속의 정보가 각 칸에 표시되고 저장 버튼을 눌러 수정사항을 반영한다.



[그림 6-6] Text to Schedule 화면

위 그림은 약속을 자동으로 인식하여 추가하는 타이핑 화면과 OCR 화면이다. 타이핑 화면에서 약속의 정보를 양식에 맞게 작성하고 추가 버튼을 누르면 작성한 내용을 파싱하여 해당 약속을 생성한다. OCR 화면에서는 이미지를 불러와 해당 이미지의 글자를 파싱하여 새로운 약속을 생성한다.



[그림 6-7] 메모 등록 화면

위 그림은 메모 등록 화면이다. 간단한 메모를 입력하고 등록 버튼을 누르면 메모가 등록되고 약속 상세 정보 화면에서 확인 할 수 있다.



정렬 기준 이름순 ▾



그룹 생성



[그림 6-8] 그룹 목록 화면

현재 사용자가 속한 그룹의 이름과 그룹 색상을 확인할 수 있다. 그룹을 눌러 해당 그룹의 상세 정보를 볼 수 있고, 상단의 필터를 적용하여 그룹 목록을 이름순 또는 인원순으로 정렬하여 볼 수 있다. 또한 플로팅 버튼을 눌러 새로운 그룹을 생성할 수 있다.

sss

Young University ●

인원 수: 2 / 5 명

팀 프로젝트 그룹.
일정 약속을 편하게 잡기 위해 만들어졌습니다.

정수용
접속 중
진행 중인 약속 없음

최문빈
미접속
약속 진행 중



[그림 6-9] 그룹 상세 정보 화면

이전의 그룹 목록 화면에서 선택한 그룹의 상세 정보를 확인할 수 있다. 그룹의 이름, 인원, 색상, 설명을 바로 확인할 수 있고 우측 하단의 플로팅 버튼을 탭하여 그룹 정보 수정, 삭제, 초대, 강제 퇴장, 탈퇴 기능을 이용할 수 있다.

sss

그룹명

그룹 설명

인원 수

5

색상

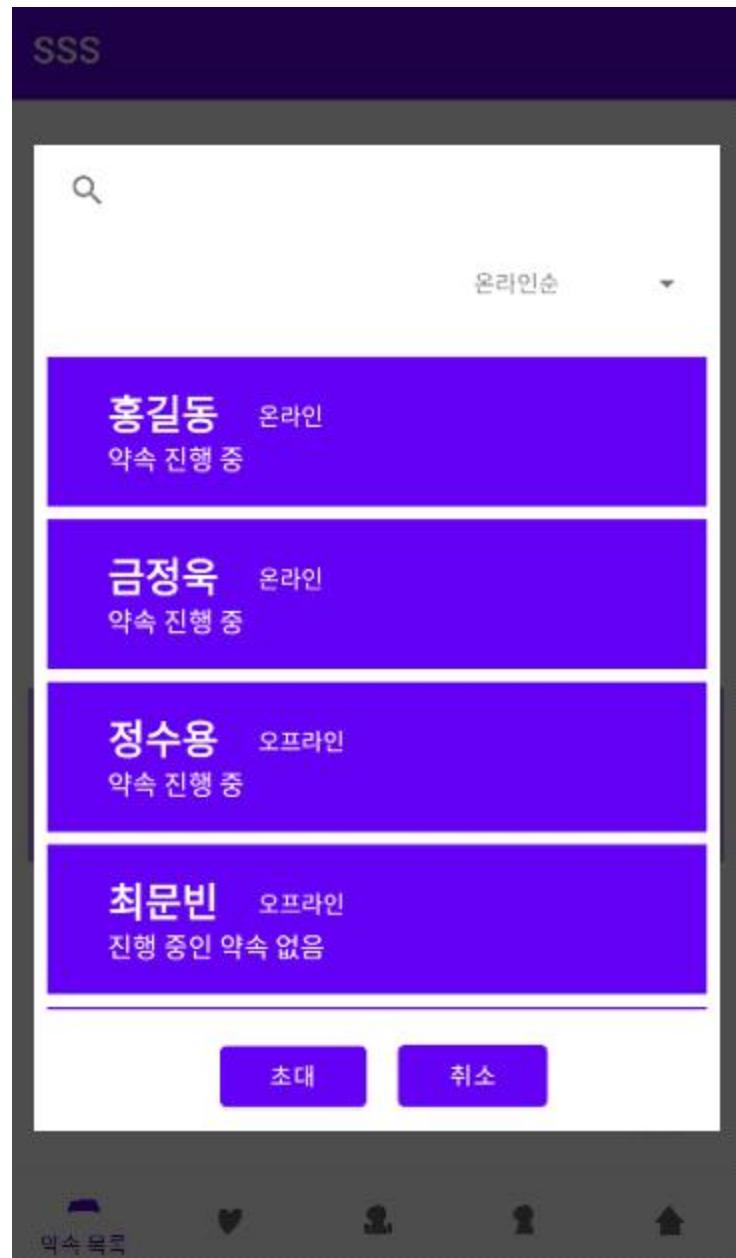
빨강

생성

그룹 목록

[그림 6-10] 그룹 생성 화면

이전의 그룹 목록 화면에서 그룹 생성 버튼을 누른 후 생성할 그룹의 이름, 설명, 그룹 색상을 설정할 수 있다. 설정을 완료하였으면 저장 버튼을 클릭하여 그룹을 생성한다. 그룹 수정 기능을 통해 해당 화면으로 넘어왔다면 해당 그룹의 정보가 각 칸에 표시되고 저장 버튼을 눌러 수정사항을 반영한다.



[그림 6-11] 약속(그룹) 초대 화면

친구 목록에서 초대하고 싶은 사용자들을 선택하여 약속 혹은 그룹에 초대할 수 있다. 표시되는 목록은 우측 상단의 필터를 통해 온라인순 혹은 약속 없는 순으로 정렬하여 볼 수 있으며, 상단의 검색 바를 통해 이름으로 강제 퇴장할 사용자를 검색할 수도 있다. 초대할 사용자들을 모두 선택하였으면 하단의 초대 버튼을 클릭하여 사용자들을 약속에 초대한다.

sss

홍길동님이 약속초대를 하셨습니다.

2021년 10월 30일

13시 00분

영남대학교

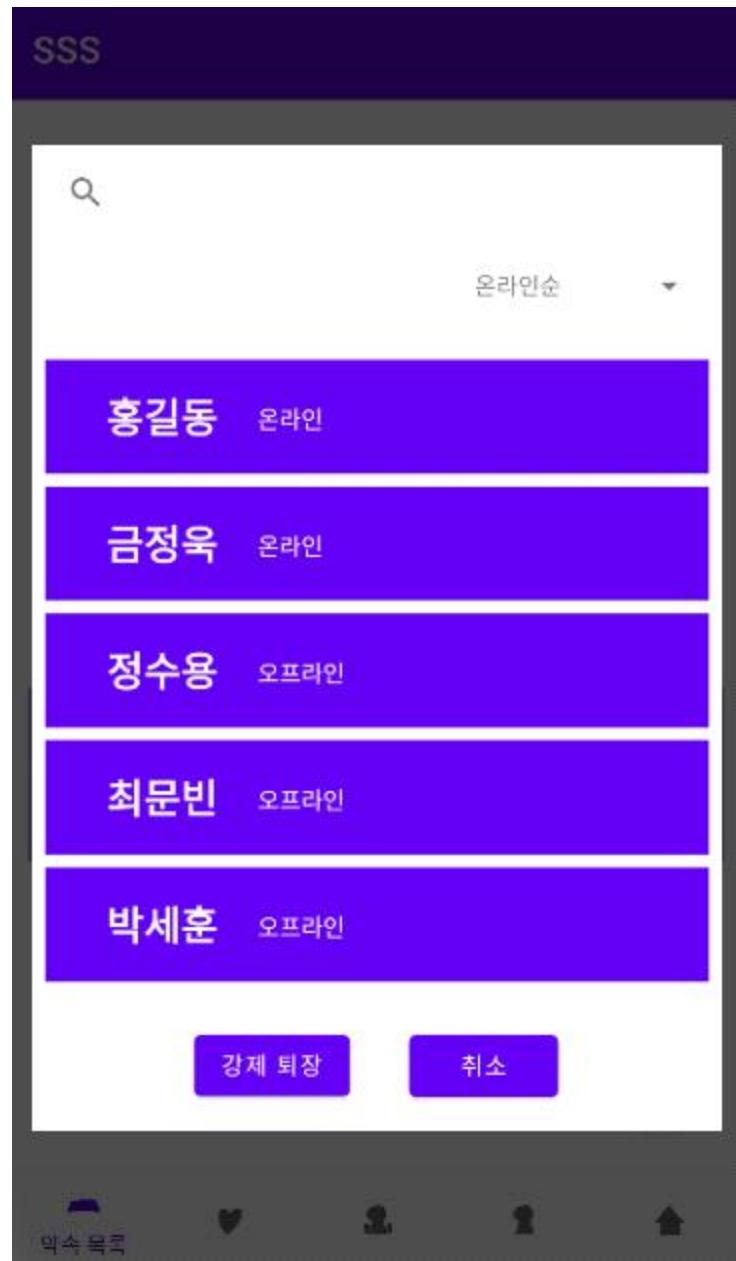
초대를 수락하시겠습니까?

수락

거절

[그림 6-12] 초대 수락 화면

약속에 초대 받으면 나타나는 화면이다. 누가 어떤 약속 혹은 그룹에 초대했는지 알 수 있으며, 하단의 수락 버튼과 거절 버튼을 눌러 초대를 수락하거나 거절할 수 있다.



[그림 6-13] 약속(그룹) 강제 퇴장 화면

현재 약속 혹은 그룹에 속한 사용자들의 목록에서 이름을 탭하여 선택한 사용자들을 약속 혹은 그룹에서 강제 퇴장시킬 수 있다. 표시되는 목록은 우측 상단의 필터를 통해 온라인순 혹은 오프라인순으로 정렬하여 볼 수 있으며, 상단의 검색 바를 통해 이름으로 강제 퇴장할 사용자를 검색할 수도 있다. 강제 퇴장시킬 사용자들을 모두 선택하였으면 하단의 강제 퇴장 버튼을 클릭하여 사용자들을 약속 혹은 그룹에서 강제 퇴장시킨다.



[그림 6-14] 사용자 검색 화면

위 그림은 사용자 검색 화면이다. 아이디나 닉네임을 입력하여 사용자를 검색할 수 있고 추가 버튼을 눌러 해당 사용자와 친구가 될 수 있다.



[그림 6-15] 친구 목록 화면

위 그림은 친구 목록 화면이다. 친구로 등록된 사용자들의 목록이 표시되고 각각의 사용자들에 대해 접속 여부와 약속 진행 여부를 확인할 수 있다. 사용자는 친구 칸을 눌러 해당 사용자의 프로필을 볼 수 있고, 차단이나 삭제 버튼을 눌러 사용자를 차단하거나 삭제할 수 있다. 또한 Floating Action 버튼을 누르면 사용자 검색 화면 메뉴가 뜬다.



닉네임(아이디)

이름 : 흥길동
나이 : 23
성별 : 남

프로필 수정

통계 보기

설정

로그아웃



[그림 6-16] 프로필 화면

위 그림은 프로필 화면이다. 프로필 사진, 닉네임, 아이디, 이름, 나이, 성별 정보를 확인할 수 있다. 또한 Floating Action 버튼을 눌러 로그아웃, 설정, 프로필 수정, 통계 보기의 기능을 사용할 수 있다.



[그림 6-17] 프로필 수정 화면

프로필 화면에서 Floating Action 버튼을 눌러 프로필 수정 버튼을 누르면 위 그림과 같이 프로필 수정 화면으로 넘어간다. 사용자는 프로필 수정 화면에서 닉네임, 이름, 나이, 성별, 프로필 사진을 변경할 수 있다.



[그림 6-18] 통계 화면

프로필 화면에서 Floating Action 버튼을 눌러 통계 보기 버튼을 누르면 위 그림과 같이 통계 화면으로 넘어간다. 사용자는 자신의 포인트, 총 약속 횟수, 참석 횟수, 참석률을 확인할 수 있으며, 포인트를 모아 통계를 해금할 수 있다.



푸시 알림 허용 

약속 초대 거부 

친구 추가 거부 

개발자

금정욱

최문빈

박세훈

정수용

이소연



[그림 6-19] 설정 화면

프로필 화면에서 설정 버튼을 누르면 위 그림과 같이 설정 화면으로 넘어간다. 사용자는 푸시 알림, 약속 초대, 친구 초대 여부에 대한 설정을 변경할 수 있으며, 개발자 정보를 확인할 수 있다.

sss



닉네임(아이디)

이름 : 홍길동

나이 : 23

성별 : 남

2021/11/05 12:00 ~ 13:00

반월당역

2021/10/31 13:30 ~ 18:00

영남대학교 IT관



친구 목록

[그림 6-20] 다른 사용자의 프로필 화면

친구 목록에서 친구 칸을 누르면 위 그림과 같이 다른 사용자의 프로필 화면으로 넘어간다. 프로필 사진, 닉네임, 아이디, 이름, 나이, 성별과 같은 프로필 정보와 해당 사용자의 약속 목록을 확인할 수 있다.



[그림 6-21] 인기 약속 장소 추천 화면

하단 메뉴의 약속 장소 추천 버튼을 누르거나 약속 목록 화면에서 화면을 슬라이드하면 인기 약속 장소 추천 화면으로 넘어가고, 사용자는 인기 약속 장소 목록을 확인할 수 있다.

sss

도착 예상 시간: 3분 ●



정수용 ●
1시간 30분

최모비

익속 목록

[그림 6-22] 실시간 위치 추적 화면

약속 상세 정보 화면에서 내 위치 공유 버튼을 누르면 실시간 위치 추적 화면으로 넘어간다. 이후 사용자는 자신의 위치와 도착 예정 시간을 약속 구성원들과 공유할 수 있으며, 다른 약속 구성원들의 위치와 도착 예정 시간을 확인할 수 있게 된다.

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

App server

- Amazon Web Server EC2 Instance

- Instance type: t2.micro
- CPU: 1 vCPU
- RAM: 1GB
- Storage: 30GB

DB server

- Amazon Web Server RDS Instance

- Instance type: db.t2.micro
- CPU: 1 vCPU
- RAM: 1GB
- Storage: 20GB

S/W platform requirements

Application

- Android Studio 2020.3.1+
- Android SDK Tools 23+
- JDK 1.8.0+
- JRE 1.8.0+
- App OS: Android 5.0 (Lollipop)+
- Develop Environment OS: Windows 8/10
- Implementation Language: Kotlin, xml

Server

- OS: Ubuntu 20.04 LTS
- Implementation Language: Node.js
- DBMS: MySQL Community

Nonfunctional requirements

- Modularity
- Performance

8. Glossary

이름	설명
로컬 DB	앱 내부에 데이터를 저장하기 위한 데이터베이스
세부메뉴	화면 내의 정보에서 추가적으로 수정, 삭제 등의 부가기능들을 모아 버튼으로 접근하는 메뉴
인스턴스	클래스의 현재 생성된 객체
파싱	Parsing의 한글 표기
푸시	Push notification의 한글 줄임 표기

9. References

- [1] Alistair Cockburn, Writing Effective Use Cases (n.p.: Addison-Wesley Longman, 1999), 46-52, 125-127, 170, 174-177.
- [2] ORACLE, Getting Started With Use Case Modeling - An Oracle White Paper (n.p.: ORACLE, 2007), 8, 9.
- [3] Alistair Cockburn, Writing Effective Use Cases (n.p.: Addison-Wesley Longman, 1999), 49.
- [4] Alistair Cockburn, Writing Effective Use Cases (n.p.: Addison-Wesley Longman, 1999), 174-177.
- [5] Ramez Elmasri, Shamkant B. Navathe, FUNDAMENTALS OF Database Systems SEVENTH EDITION (n.p.: PERASON, n.d.), 85-88.
- [6]
<https://github.com/android/architecture-samples/blob/dev-todo-mvvm-live/todoapp/app/src/main/java/com/example/android/architecture/blueprints/todoapp/SingleLiveEvent.java>
- [7] "Use Android Jetpack to Accelerate Your App Development," android developers, n.d. 수정,
<https://android-developers.googleblog.com/2018/05/use-android-jetpack-to-accelerate-your.html?m=1>.
- [8] "retrofit2 Interface Call," square.github, n.d. 수정,
[https://square.github.io/retrofit/2.x/retrofit/retrofit2/Call.html#:~:text=decoding%20the%20response.-,enqueue,-void%C2%A0enqueue\(Callback.](https://square.github.io/retrofit/2.x/retrofit/retrofit2/Call.html#:~:text=decoding%20the%20response.-,enqueue,-void%C2%A0enqueue(Callback.)