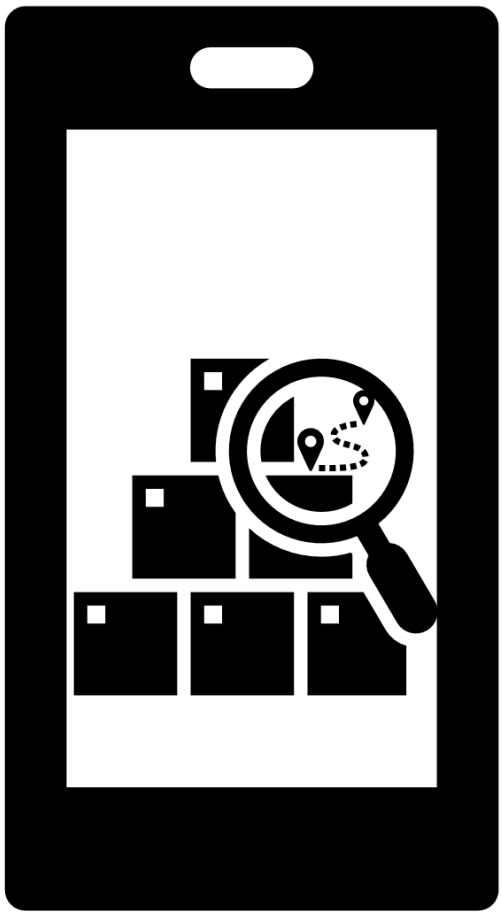
Yeungnam University

Lost & Found App

3. Design Document



Yeungnam University

[Revision history]

Revision date	Version #	Description	Author
	0.01	First Documentation	
	0.02	Class diagram, Sequence diagram 수정	

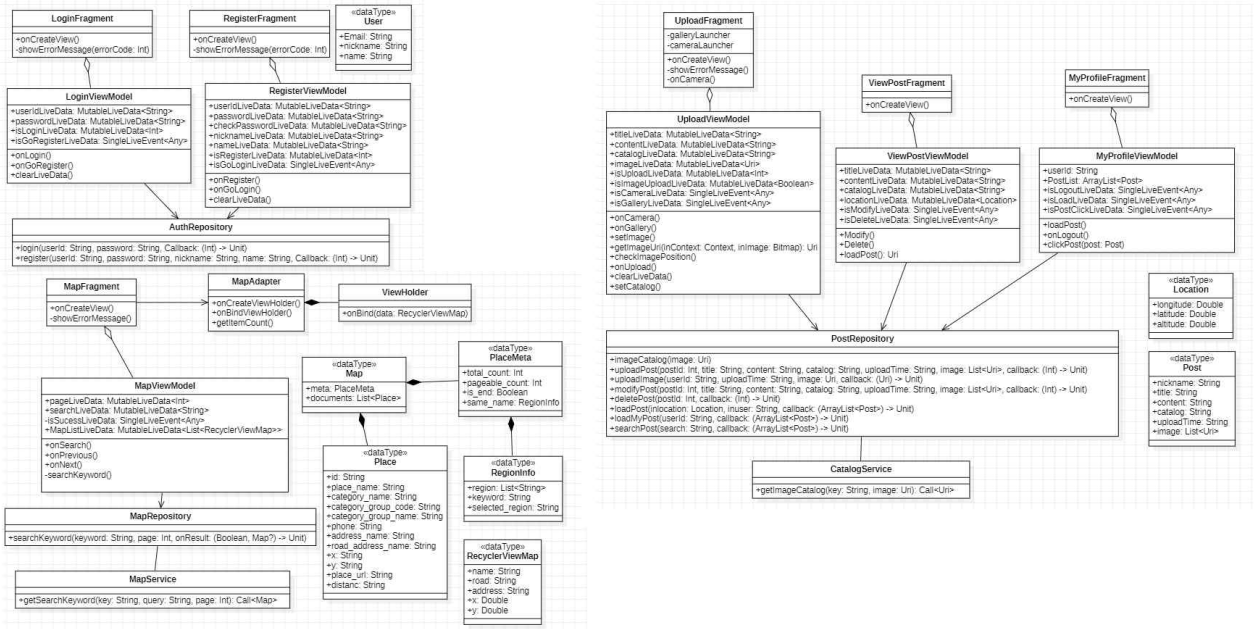
= Contents =

1. Introduction

분실물들을 어디서 잃어버렸는지 기억하지 못해 찾지 못하는 경우 또는 분실물센터가 있으나 어느 분실물센터에서 찾아야 할지 막막한 상황이 많다. 이를 스마트폰을 이용해 시공간적인 측면에서 많은 불편함을 해결하는 것이 분실물 습득 앱의 목적이다.

본 문서는 Analysis에 이은 Design 단계의 문서로 class diagram, sequence diagram, state machine diagram을 그리고 각 diagram에 대한 설명을 한다. 추가로 implementation requirements를 기술하여 본 시스템을 구현하기 위한 소프트웨어 및 하드웨어의 요구사항을 기술한다. 본 시스템은 안드로이드 운영체제 기반의 스마트폰에서 개발될 것이며, 아키텍처 중 mvvm 패턴을 사용한다.

2. Class diagram



[그림 1] Class diagram

[그림 1]의 Class diagram에서 화면을 구성하는 클래스는 설명을 하지 않을 것이다.

1) LoginViewModel

Attributes
+userIdLiveData: MutableLiveData<String> : 사용자가 입력한 아이디
+passwordLiveData: MutableLiveData<String> : 사용자가 입력한 비밀번호
+isLoginLiveData: MutableLiveData<Int> : 로그인 결과를 가지는 값
+isGoRegisterLiveData: SingleLiveEvent<Any> : 회원가입 버튼을 눌렀을 때 변경되는 값
Methods
+onLogin() : 로그인 실행
+onGoRegister() : 로그인 화면에서 회원가입 화면으로 전환
+clearLiveData() : 로그아웃 후 로그인 정보 초기화

2) RegisterViewModel

Attributes
+userIdLiveData: MutableLiveData<String> : 사용자가 입력한 아이디 +passwordLiveData: MutableLiveData<String> : 사용자가 입력한 비밀번호 +checkPasswordLiveData: MutableLiveData<String> : 사용자가 입력한 비밀번호 확인 +nicknameLiveData: MutableLiveData<String> : 사용자가 입력한 닉네임 +nameLiveData: MutableLiveData<String> : 사용자가 입력한 이름 +isRegisterLiveData: MutableLiveData<Int> : 회원가입 결과를 가지는 값 +isGoLoginLiveData: SingleLiveEvent<Any> : 로그인 버튼을 눌렀을 때 변경되는 값
Methods
+onRegister() : 회원가입 실행 +onGoLogin() : 회원가입 화면에서 로그인 화면으로 전환 +clearLiveData() : 회원가입 성공 후 입력값 초기화

3) AuthRepository

Attributes
Methods
+login(userId: String, password: String, Callback: (Int) -> Unit) : 로그인 실행 후 결과값 콜백함수 호출로 전달 +register(userId: String, password: String, nickname: String, name: String, Callback: (Int) -> Unit) : 회원가입 실행 후 결과값 콜백함수 호출로 전달

4) User

Attributes
+Email: String : 사용자의 이메일 +nickname: String : 사용자의 닉네임 +name: String : 사용자의 이름
Methods

5) UploadViewModel

Attributes
+titleLiveData: MutableLiveData<String> : 사용자가 작성한 글의 제목
+contentLiveData: MutableLiveData<String> : 사용자가 작성한 글의 내용
+catalogLiveData: MutableLiveData<String> : 사용자가 작성한 글의 카테고리
+imageLiveData: MutableLiveData<Uri> : 사용자가 작성한 글의 사진
+isUploadLiveData: MutableLiveData<Int> : 업로드 후 성공/실패 값을 갖는 변수
+isImageUploadLiveData: MutableLiveData<Boolean> : 사진이 등록된지 확인하는 변수
+isGalleryLiveData: SingleLiveEvent<Any> : 갤러리 앱 실행 알리는 이벤트
+isCameraLiveData: SingleLiveEvent<Any> : 카메라 앱 실행 알리는 이벤트
Methods
+onCamera() : 카메라 실행
+onGallery() : 갤러리 실행
+setImage(data: Uri) : 사진 등록
+getImageUri(inContet: Context, inImage: Bitmap): Uri : 비트맵 데이터를 Uri로 전환
+clearLiveData : 글 등록 정보 초기화
+checkImagePosition() : 사진에 위치정보가 있는지 확인
+setCatalog() : 사진으로 글 카테고리 설정
+onUpload() : 글 업로드 실행

6) ViewPostViewModel

Attributes
+isModifyLiveData: SingleLiveEvent<Any> : 게시글 수정을 알리는 이벤트
+isDeleteLiveData: SingleLiveEvent<Any> : 게시글 삭제를 알리는 이벤트
+titleLiveData: MutableLiveData<String> : 게시글의 제목을 갖는 데이터
+contentLiveData: MutableLiveData<String> : 게시글의 내용을 갖는 데이터
+catalogLiveData: MutableLiveData<String> : 게시글의 카테고리를 갖는 데이터
+locationLiveData: MutableLiveData<Location> : 게시글의 위치를 갖는 데이터
Methods
+Modify() : 글 수정 실행
+Delete() : 글 삭제 실행
+loadPost(): Uri : 게시글 정보 설정

7) MyProfileViewModel

Attributes
+userId: String : 사용자의 아이디 정보 갖는 데이터 +postList: ArrayList<Post> : 사용자가 작성한 게시글 리스트 +isLogoutLiveData: SingleLiveEvent<Any> : 로그아웃 실행 알리는 이벤트 +isLoadLiveData: SingleLiveEvent<Any> : 게시글 불러오기 실행완료를 알리는 이벤트 +isPostClickLiveData: SingleLiveEvent<Any> : 게시글이 클릭된 것을 알리는 이벤트
Methods
+loadPost() : 사용자가 작성한 게시글 리스트 load +onLogout() : 로그아웃 실행 +clickPost(post: Post) : 선택된 게시글 정보 load

8) PostRepository

Attributes
Methods
+imageCatalog(image: Uri) : 사진의 카테고리를 설정 +uploadPost(userId: String, title: String, content: String, catalog: String, uploadTime : String, image: Uri, callback: (Int) -> Unit) : 게시글 업로드 +uploadImage(userId: String, uploadTime: String, image: Uri, callback: (Uri) -> Unit) : 사진 업로드 +ModifyPost(postId: Int, title: String, content: String, catalog: String, uploadTime : String, image: Uri, callback: (Int) -> Unit) : 게시글 수정 +deletePost(postId: Int) : 게시글 삭제 +loadPost(location: Location, user: String, callback: (ArrayList<Post>) -> Unit) : 설정한 위치와 비슷한 위치의 게시글 load +loadMyPost(userId: String, callback: (ArrayList<Post>) -> Unit) : 사용자가 작성한 게시글 load +searchPost(search: String, callback: (ArrayList<Post>) -> Unit) : 제목에 검색어가 포함된 게시글 load

9) CatalogService

Attributes
Methods
+getImageCatalog(key: String, image: Uri): Call<Uri> : 카카오 비전 API를 사용하여 사진의 카테고리 설정

10) Post

Attributes
+nickname: String : 게시글 작성자의 닉네임 +title: String : 게시글의 제목 +content: String : 게시글의 내용 +catalog: String : 게시글의 카테고리 +uploadTime: String : 게시글 업로드 시간 +image: Uri : 게시글의 사진
Methods

11) Location

Attributes
+longitude: Double : 경도 +latitude: Double : 위도 +altitude : Double : 고도
Methods

12) MapViewModel

Attributes
+pageLiveData: MutableLiveData<Int> : 실시간 페이지 번호 +searchLiveData: MutableLiveData<String> : 사용자가 검색한 검색 값 -isSuccessLiveData: SingleLiveEvent<Any> : 검색 성공여부를 알리는 이벤트 +MapListLiveData: MutableLiveData<List<RecyclerViewMap>> : 검색 결과 화면에 보일 장소들의 리스트
Methods
+onSearch() : 검색 값으로 검색 +onPrevious() : 이전 페이지로 이동 +onNext() : 다음 페이지로 이동 -searchKeyword() : 입력받은 검색 값, 페이지 번호로 검색

13) MapRepository

Attributes
Methods
+searchKeyword(keyword: String, page: Int, onResult: (Boolean, Map?) -> Unit) : keyword와 페이지 번호로 검색한 결과를 반환

14) MapService

Attributes
Methods
+getSearchKeyword(key: String, query: String, page: Int): Call<Map> : 카카오 Map API를 사용하여 검색 결과 반환

15) Map

Attributes
+meta: PlaceMeta : 장소 메타데이터 +documents: List<Place> : 검색 결과
Methods

16) PlaceMeta

Attributes
+total_count: Int : 검색어에 검색된 문서 수 +pageable_count: Int : total_count 중 노출 가능 문서 수 +is_end: Boolean : 현재 페이지가 마지막 페이지인지 여부 +same_name: RegionInfo : 검색어의 지역 및 키워드 분석 정보
Methods

17) RegionInfo

Attributes
+region: List<String> : 검색어에서 인식된 지역의 리스트 +keyword: String : 검색어에서 지역 정보를 제외한 키워드 selected_region: String : 인식된 지역 리스트 중, 현재 검색에 사용된 지역 정보
Methods

18) Place

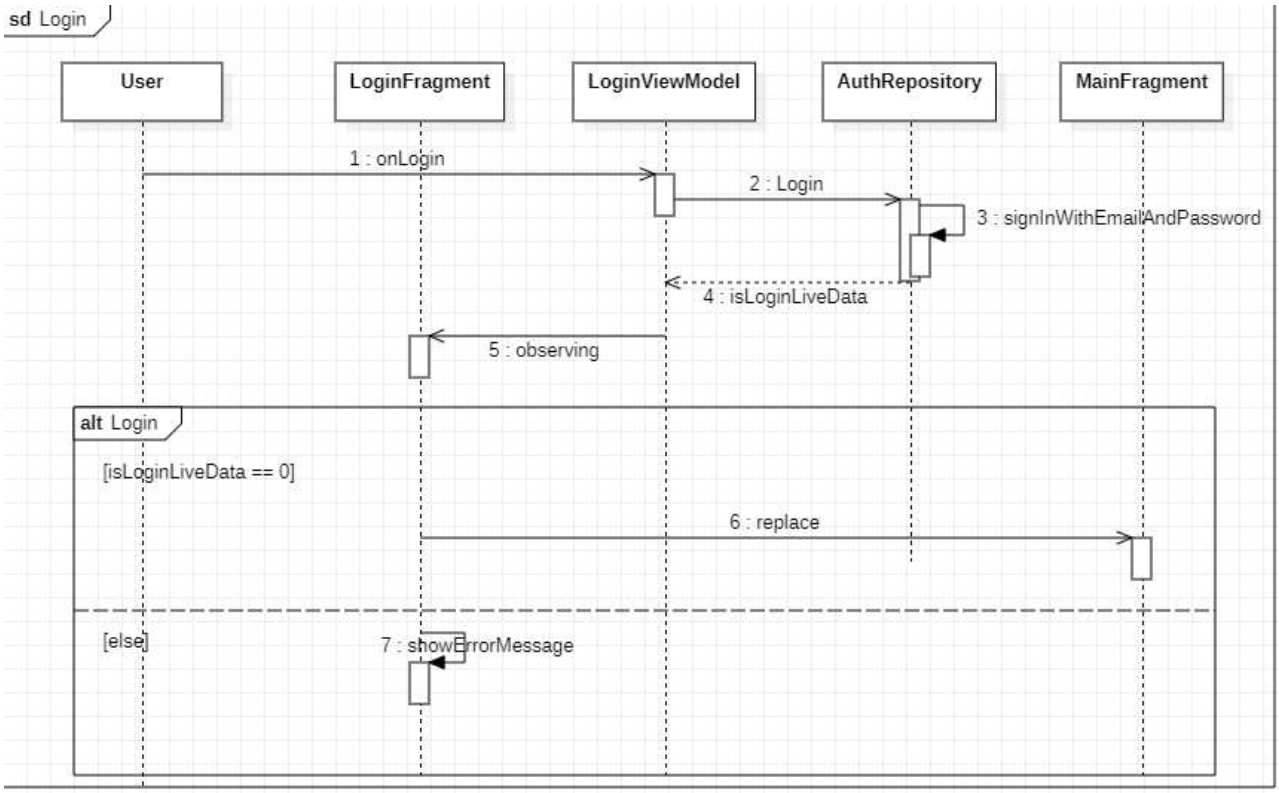
Attributes
+id: String : 장소 ID
+place_name: String ; 장소명
+category_name: String : 카테고리 이름
+category_group_code: String : 카테고리 그룹 코드
+category_group_name: String : 카테고리 그룹명
+phone: String : 전화번호
+address_name: String : 전체 지번 주소
+road_address_name: String : 전체 도로명 주소
+x: String : 경도
+y: String : 위도
+place_url: String : 장소 상세페이지 URL
+distanc: String : 중심좌표까지의 거리
Methods

19) RecyclerViewMap

Attributes
+name: String : 장소명
+road: String : 도로명
+address: String : 지번 주소
+x: Double : 경도
+y: Double : 위도
Methods

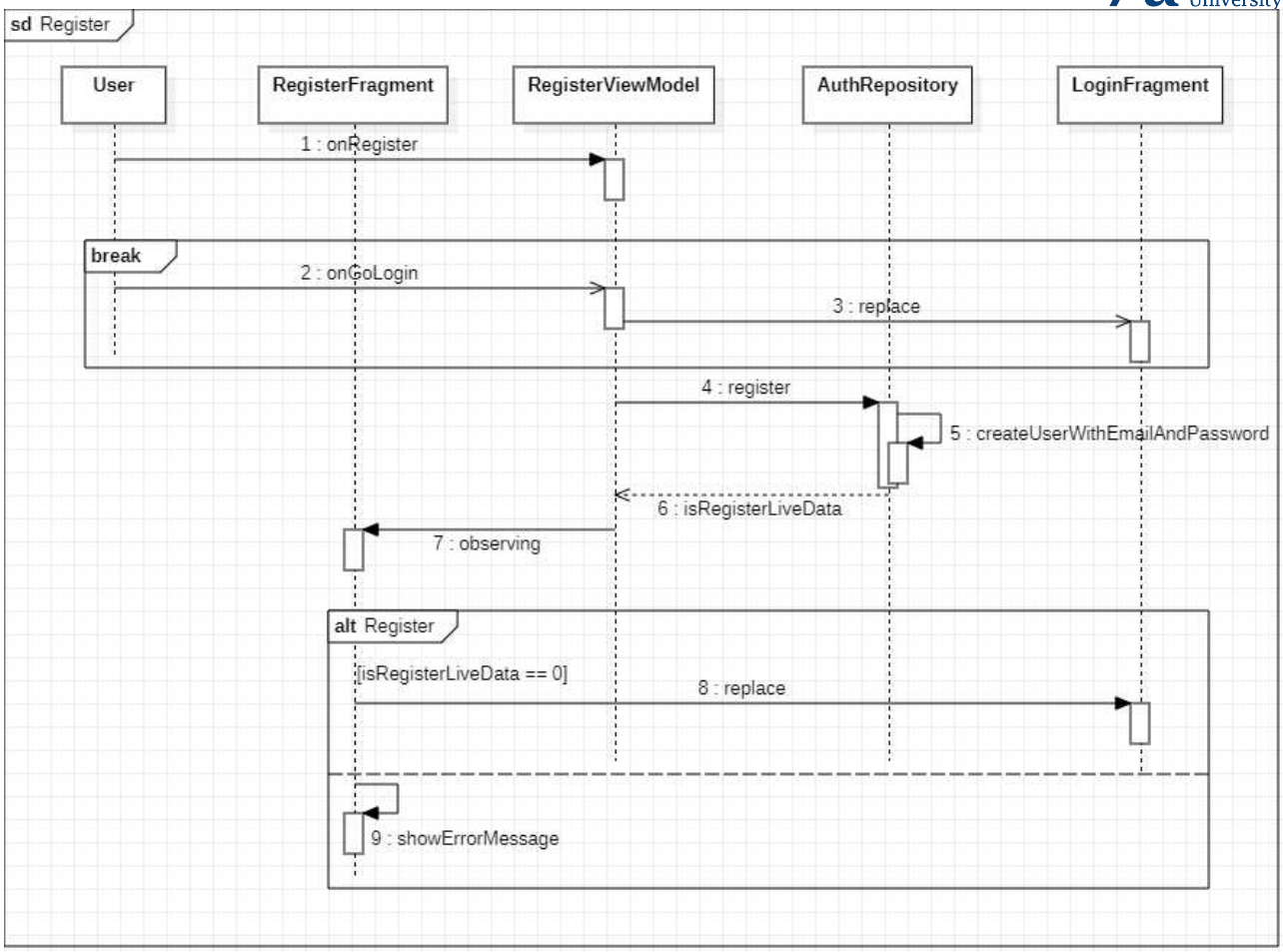
3. Sequence diagram

MVVM 패턴을 사용하면서 Databinding을 사용하였다. Databinding은 xml과 클래스의 데이터를 직접적으로 연결하여 유지관리가 더욱 쉬운 장점이 있다. 또한 LiveData를 사용하여 Observer를 사용하여 값이 바뀌는 것을 실시간으로 감지할 수 있는 장점이 있다.



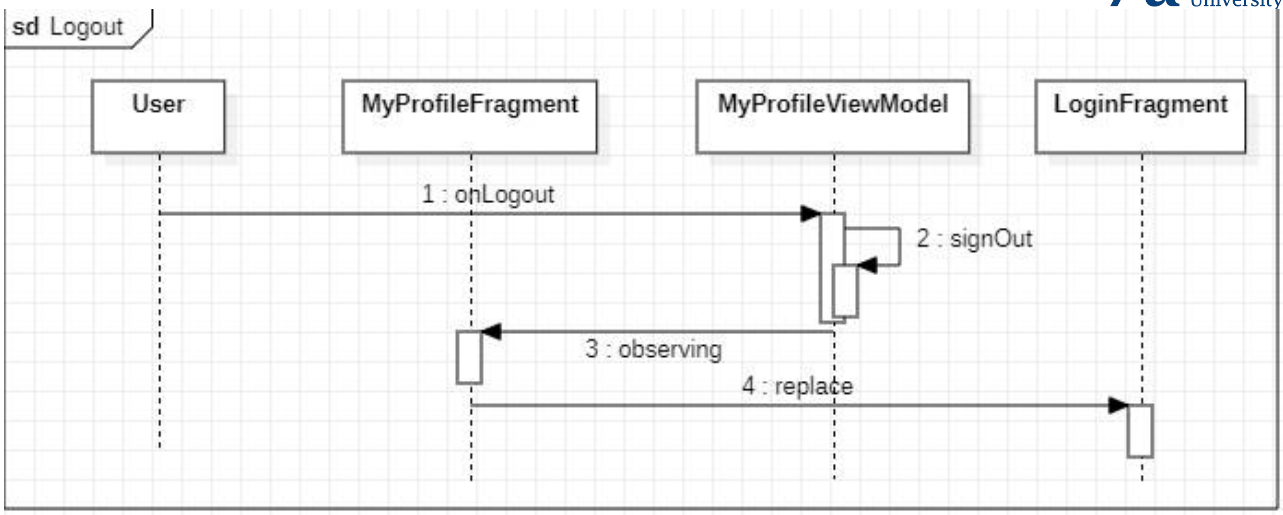
[그림 2] Login Sequence Diagram

시스템이 시작되면서 로그인 기능을 시작한다. 사용자가 로그인 정보를 입력하고 로그인 버튼을 누르면 LoginViewModel에서 값을 받아 Login을 실행한다. 그 후 AuthRepository에서 로그인을 시도하고 결괏값을 넘겨준다. 4번에서 LiveData의 값을 설정하여 LoginFragment에서 바뀐 값을 관찰하여 로그인 성공 시 MainFragment로의 화면 전환을, 실패 시 오류 메시지를 출력한다.



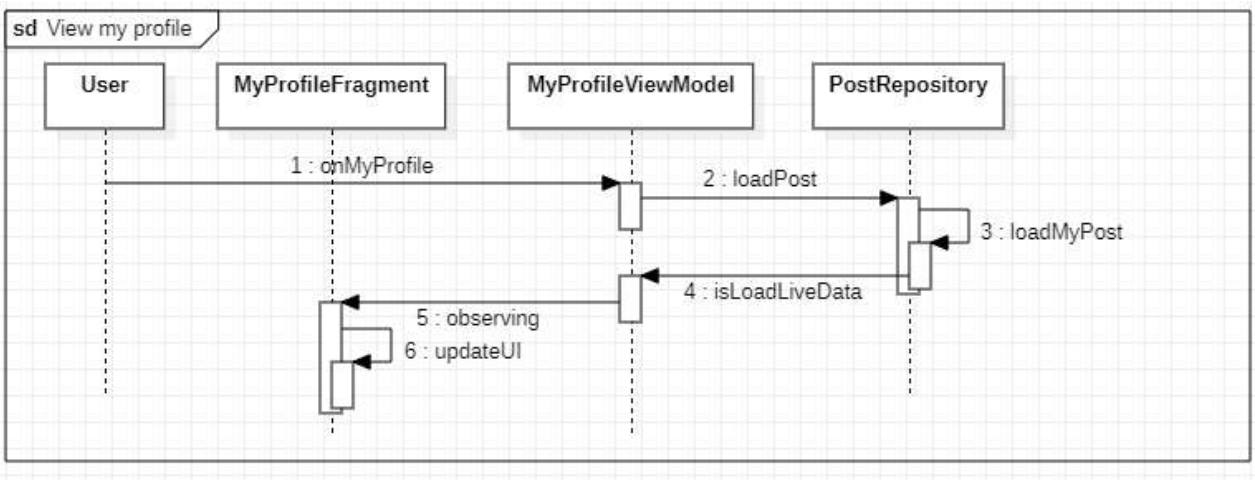
[그림 3] Register Sequence Diagram

회원가입 화면에서의 과정이다. 사용자가 회원가입을 중단하고 로그인 화면으로 이동하고 싶을 때 로그인 화면으로 버튼을 누르면 로그인 화면으로 전환한다. 사용자가 회원가입 정보를 입력하고 회원가입 버튼을 누르면 RegisterViewModel에서 값을 받아 register를 실행한다. 그 후 AuthRepository에서 회원가입을 실행하여 결괏값을 넘겨준다. 6번에서 LiveData의 값을 설정하여 RegisterFragment에서 바뀐 값에 따라 성공 시 로그인 화면으로 전환을, 실패 시 오류 메시지를 출력한다.



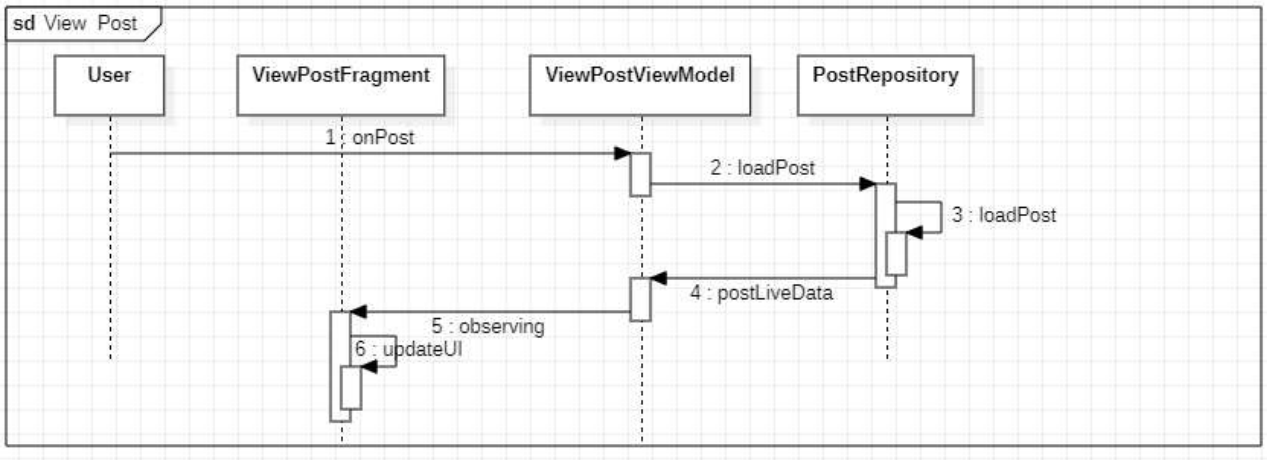
[그림 4] Logout Sequence Diagram

로그아웃을 하는 과정이다. 사용자가 MyProfile화면에서 로그아웃 버튼을 누르면 MyProfileViewModel에서 로그아웃을 하고 그 값을 MyProfileFragment에서 받아 다시 로그인 화면으로 전환한다.



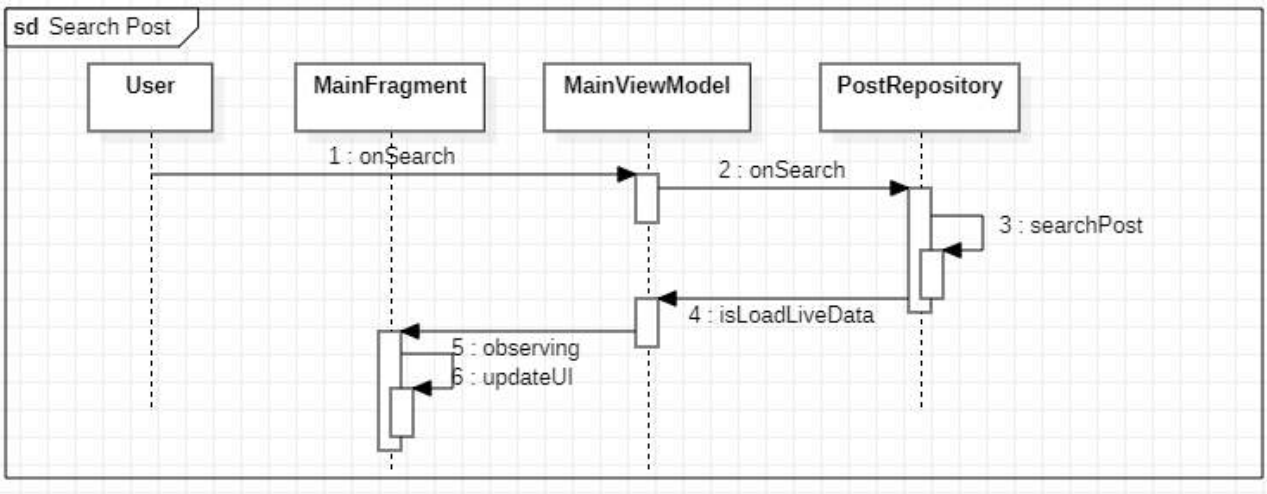
[그림 5] ViewMyProfile Sequence Diagram

내 프로필 화면에서의 과정이다. 사용자가 내 프로필 버튼을 누르면 MyProfileViewModel에서 사용자가 작성한 게시글 목록을 불러오며 4번에서 LiveData의 값을 설정하여 MyProfileFragment에서 화면을 새로고침한다.



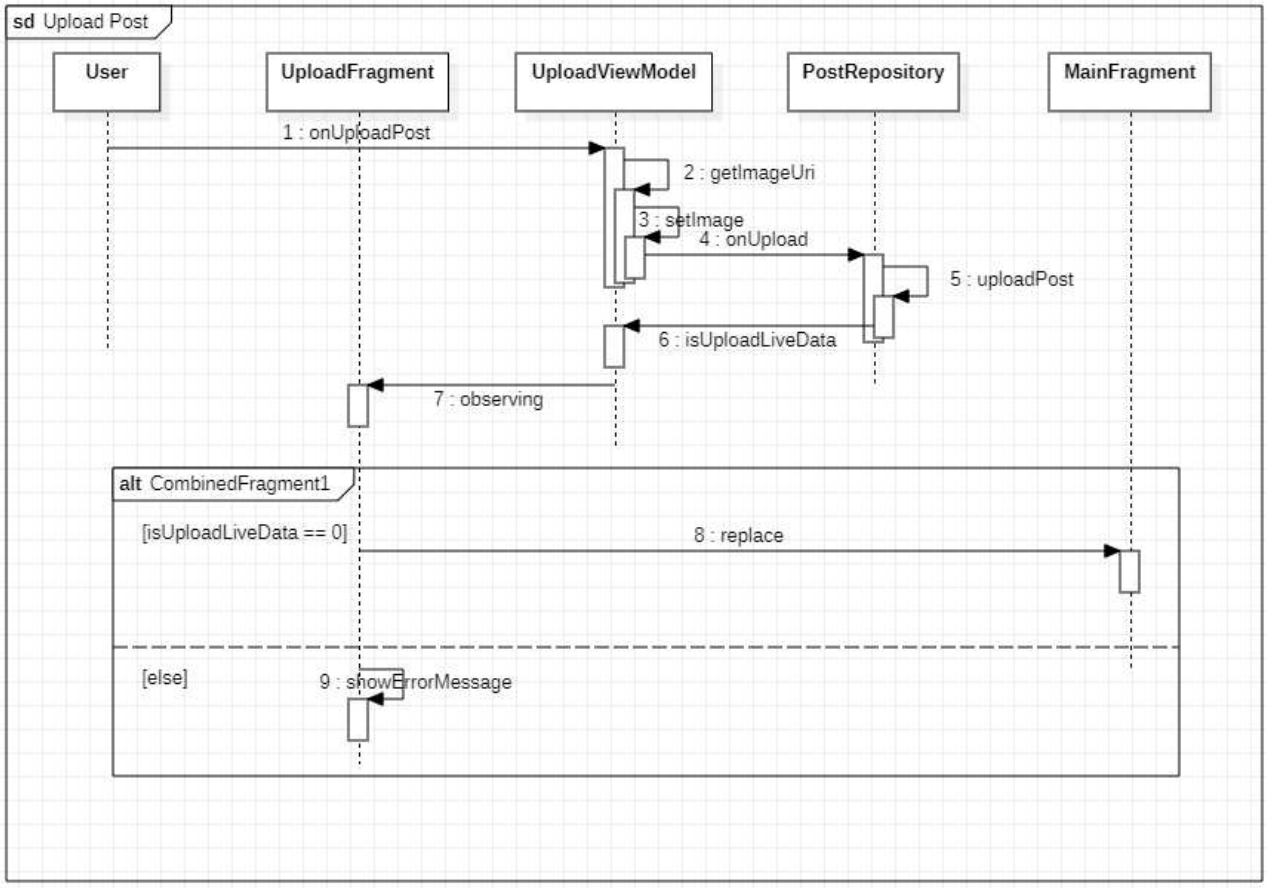
[그림 6] ViewPost Sequence Diagram

게시글을 클릭하여 자세히 볼 때의 과정이다. 사용자가 게시글을 클릭하면 ViewPostFragment에서 loadPost를 실행하여 글의 내용을 설정하도록 한다. ViewPostViewModel에서 loadPost를 실행하여 PostRepository에서 글의 정보를 4번의 LiveData에 설정한다. LiveData가 바뀜을 감지한 ViewPostFragment에서 화면을 새로고침한다.



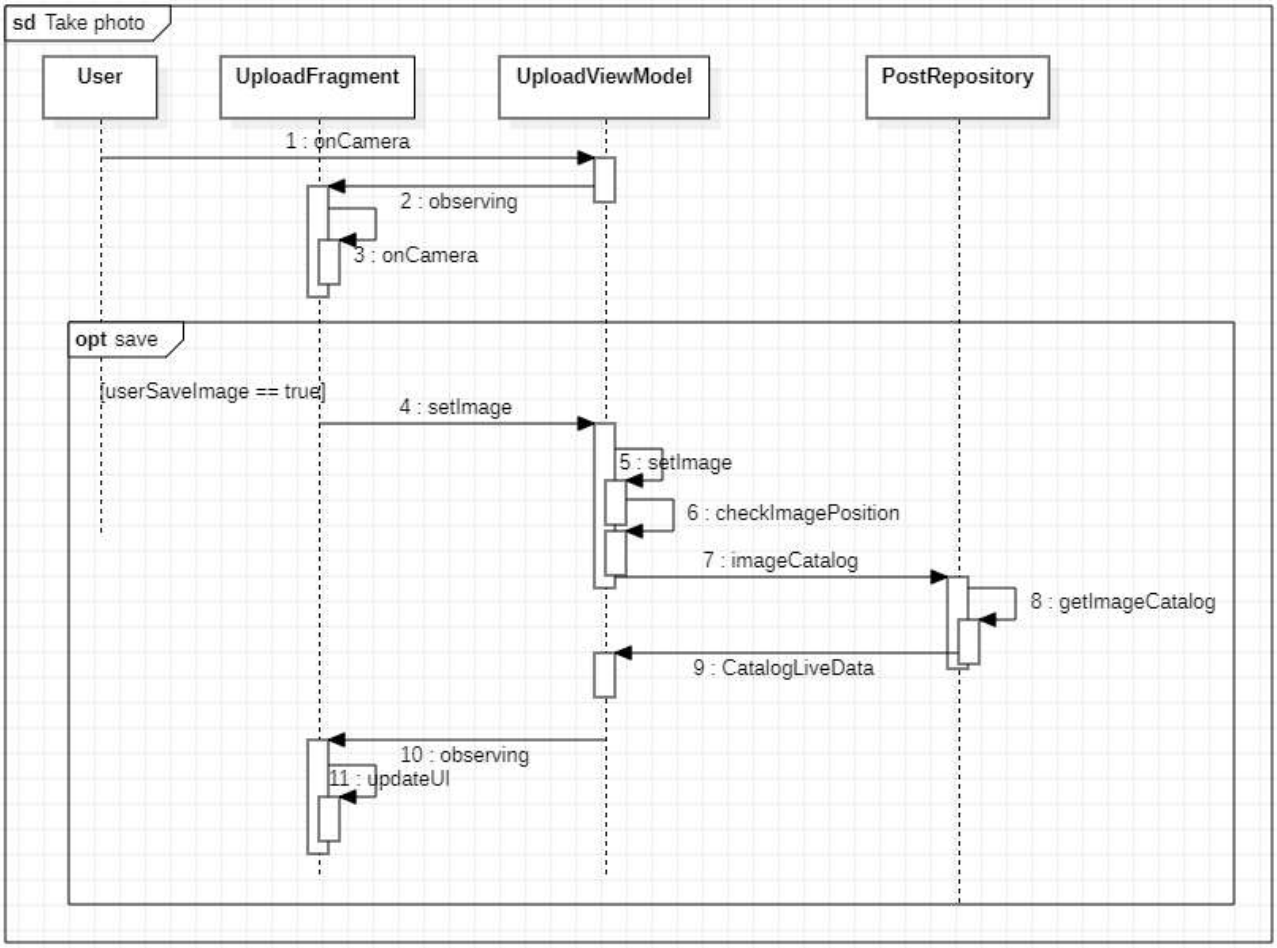
[그림 7] SearchPost Sequence Diagram

게시글을 검색하는 과정이다. 사용자가 검색어를 입력 후 검색 버튼을 누르면 MainViewModel에서 Search를 실행한다. 그 후 PostRepository에서 해당하는 검색 값을 반환하고 성공여부를 4번의 LiveData에 설정하여 값이 바뀜을 감지한 MainFragment에서 화면을 새로고침한다.



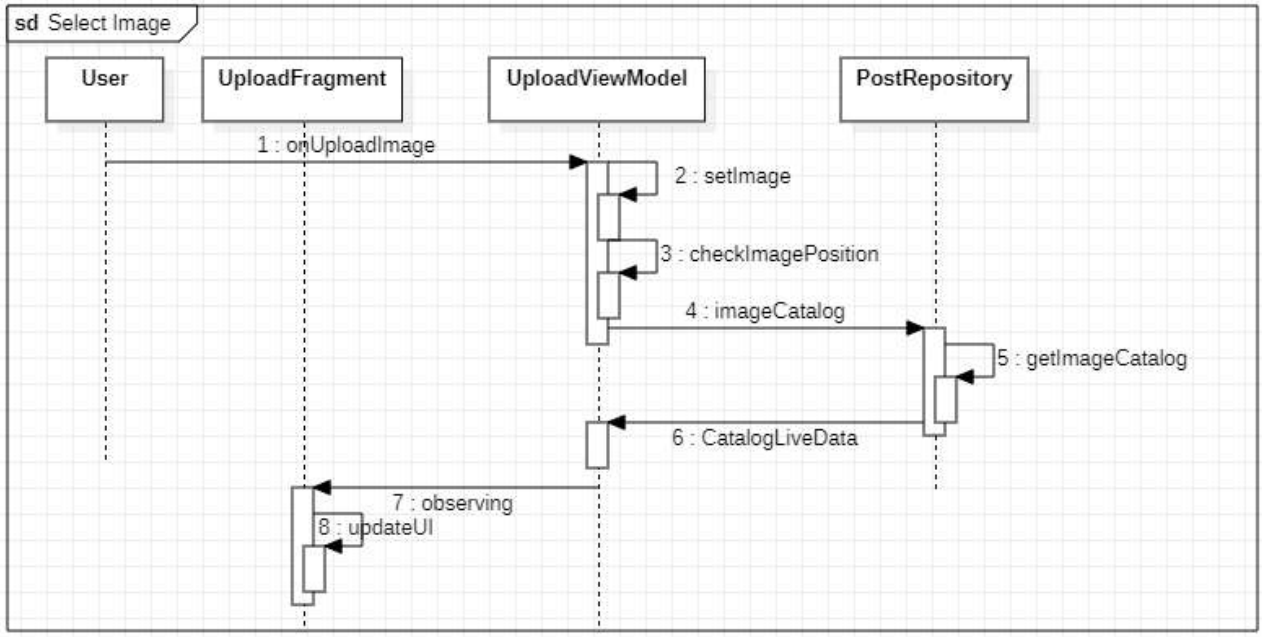
[그림 8] UploadPost Sequence Diagram

글을 등록하는 과정이다. 사용자가 메인화면에서 글 등록 버튼을 눌러 UploadViewModel에서 기본 입력 값을 입력 후, 사진을 선택하여 PostRepository에 글 등록을 요청한다. PostRepository에서 6번의 LiveData에 결괏값을 설정하여 UploadFragment에서 값을 감지 후 성공 시 MainFragment로의 전환을, 실패 시 에러메시지를 보여준다.



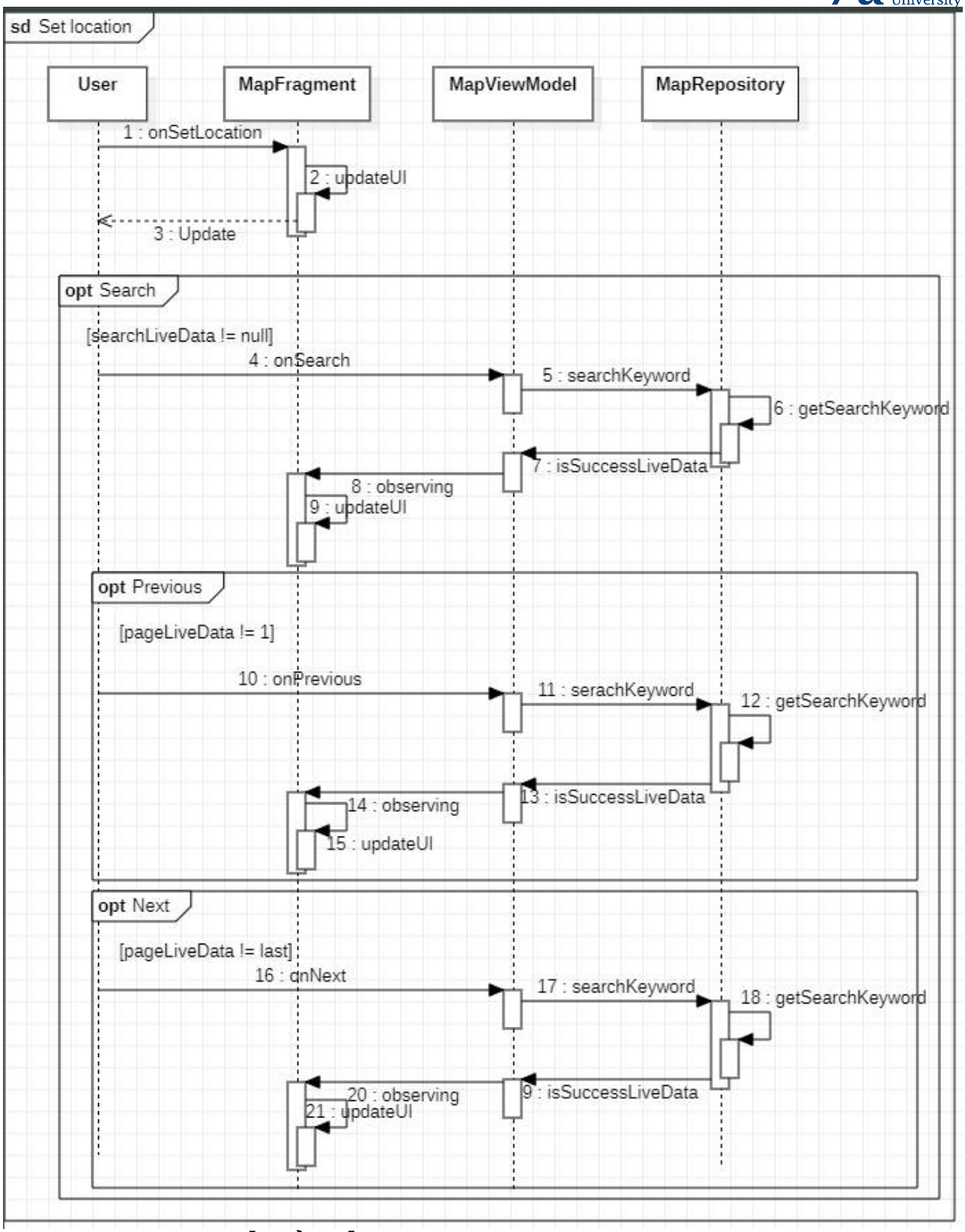
[그림 9] TakePhoto Sequence Diagram

카메라를 실행하는 과정이다. 사용자가 카메라 버튼을 누르면 카메라를 실행한다. 촬영 후 사진 등록을 원하는 경우 SetImage를 실행하여 사진을 게시글 작성 목록에 등록하고 위치정보 및 카테고리를 설정 후 UploadFragment를 업데이트한다.



[그림 10] SelectImage Sequence Diagram

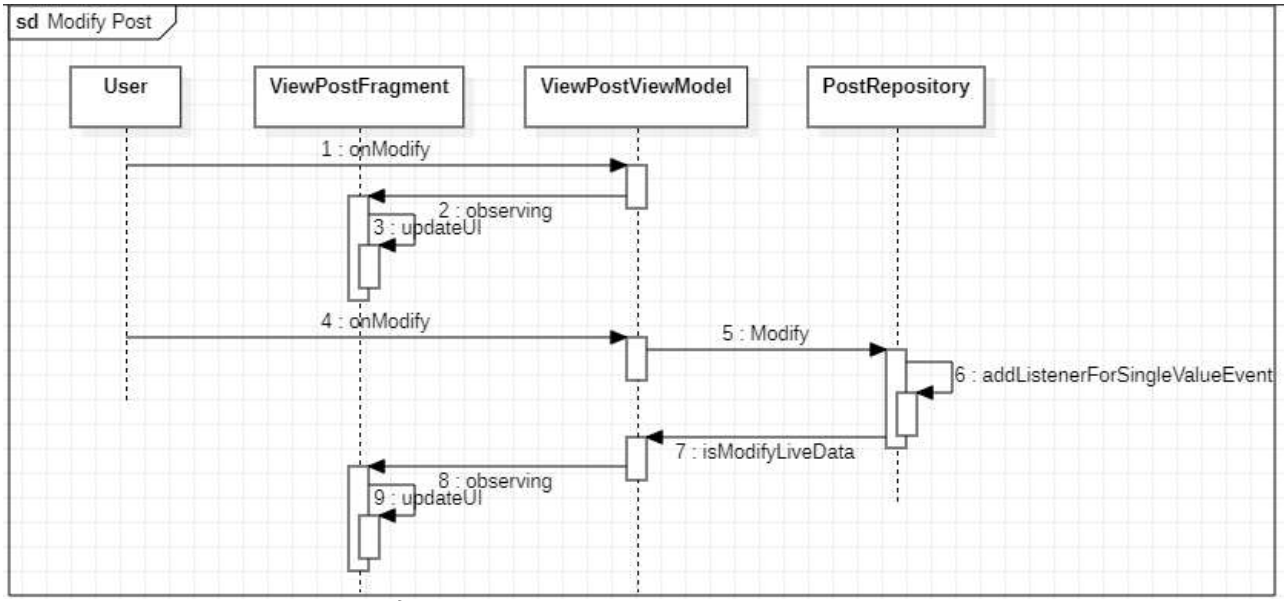
갤러리에서 사진을 선택하는 과정이다. 사용자가 사진 추가 버튼을 누르면 갤러리를 실행한다. 사진을 선택 후 checkImagePosition을 실행하여 사진이 위치정보가 포함되어있는지 확인한다. 그 후 imageCatalog를 실행하여 사진의 카테고리를 설정한다. PostRepository에서 카카오 비전 API를 이용하여 사진의 카테고리를 6번 Live Data에 설정한다. 그 후 Live Data의 값을 감지한 UploadFragment에서 화면을 업데이트한다.



[그림 11] SetLocation Sequence Diagram

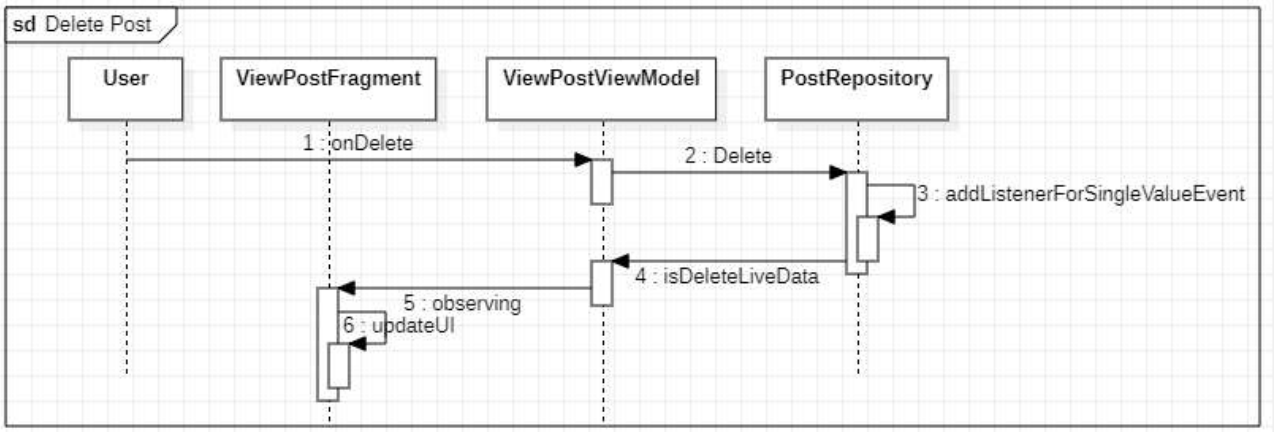
위치를 설정하는 과정이다. 사용자가 위치설정 버튼을 누르면 MapFragment에서 지도화면을 띄워 사용자에게 보여준다. 사용자가 검색을 하면 검색 값을

MapViewModel에서 받아 MapRepository에서 검색한다. 카카오 MAP API를 이용하여 검색값을 받아오고 성공여부를 7번의 LiveData에 설정하여 MapFragment에서 화면을 업데이트한다. 검색 후, 현재 페이지가 1이 아닐 경우 사용자는 이전 버튼을 누를 수 있으며 이전 버튼을 누를 시 한 페이지 전의 결과를 받아 화면에 업데이트한다. 또한 현재 페이지가 마지막 페이지가 아닐 경우 다음 버튼을 누를 수 있으며 다음 버튼을 누를 경우 한 페이지 다음 결과를 받아 화면에 업데이트한다.



[그림 12] ModifyPost Sequence Diagram

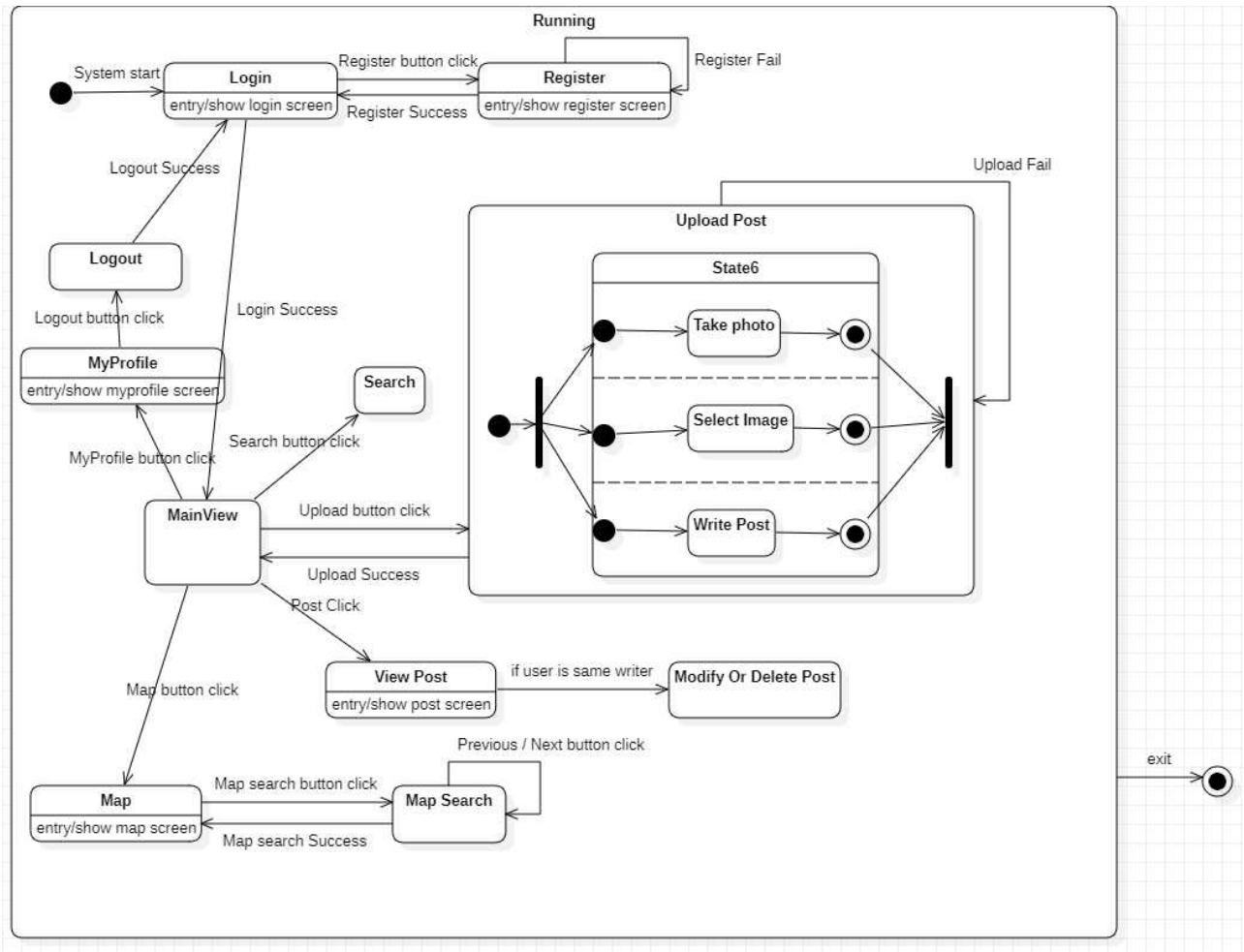
게시글을 수정하는 과정이다. 사용자가 게시글을 들어가 수정 버튼을 누르면 게시글 수정화면으로 바뀐다. 그 후 수정할 값을 입력하고 수정 버튼을 누르면 ViewPostViewModel에서 Modify를 실행하여 게시글을 수정한다. 수정 결과를 7번의 LiveData에 설정하여 바뀐 값을 감지하여 ViewPostFragment에서 화면을 업데이트한다.



[그림 13] DeletePost Sequence Diagram

게시글을 삭제하는 과정이다. 사용자가 게시글 삭제 버튼을 누르면 ViewPostViewModel에서 Delete를 실행하여 게시글을 삭제한다. 그 결괏값을 4번 LiveData에 값을 설정하여 바뀐 값을 감지하여 ViewPostFragment에서 화면을 업데이트한다.

4. State machine diagram



시스템 실행 시 Login 화면으로 시작하며 회원가입 버튼을 누를 시 회원가입 화면으로 이동한다. 회원가입이 성공하면 다시 로그인 화면으로, 실패하면 회원가입 화면에 남아있게 된다. 로그인 성공 시 메인화면으로 넘어가며 메인화면에서 내 프로필 버튼을 누를 시 사용자가 작성한 게시글 목록이 보이며 로그아웃 버튼을 누르면 로그아웃을 하여 다시 로그인 화면으로 넘어간다. 메인 화면에서 검색 값을 입력하고 검색 버튼을 누르면 검색 결과가 나온다. 업로드 버튼을 누를 시 업로드 동작을 수행하는데 사진 촬영, 사진 선택, 글 작성이 모두 끝나면 정상적인 수행인지 확인 후 성공 시 다시 메인 화면으로 이동한다. 하지만 실패 시 업로드 화면에 남아있게 된다. 그리고 게시글을 클릭하면 해당 게시글의 자세한 내용이 나오는 화면으로 넘어가고 작성자와 사용자가 일치할 경우 수정 및 삭제가 가능하다. 마지막으로 위치 정보 버튼을 클릭하면 지도 화면이 나오고 검색 값을 입력 후 검색하게 되면 검색결과가 화면에 반환되며 이전 화면, 다음 화면 버튼을 클릭하면 계속해서 검색을 하여 반환한다.

5. Implementation requirements

Android Studio 2021.2.1 이상, Android 6.0(SDK 23) 이상 시스템 구동이 가능하다. 안드로이드가 아닌 환경에서 작동 시 동작하지 않는 기능이 존재한다.

6. Glossary

Term	Description
MVVM	안드로이드 디자인 패턴 중 하나로, Model, View, ViewModel로 3가지로 구성된 패턴
메타데이터	데이터에 관한 구조화된 데이터
콜백함수	다른 함수의 인자로써 이용되는 함수

7. References

안드로이드 Databinding -
https://developer.android.com/topic/libraries/data-binding
안드로이드 Livedata -
https://developer.android.com/topic/libraries/architecture/livedata