3. Design

<멍멍걸음>

Number	22311989
Name	이수진
E-mail	ssjjlee22@gmail.com

[Revision history]

Revision date	Version #	Description	Author
06/06/2025	1.00	first draft	이수진
06/10/2025	1.01	Sequence diagram, State machine diagram, Class diagram 수정	이수진

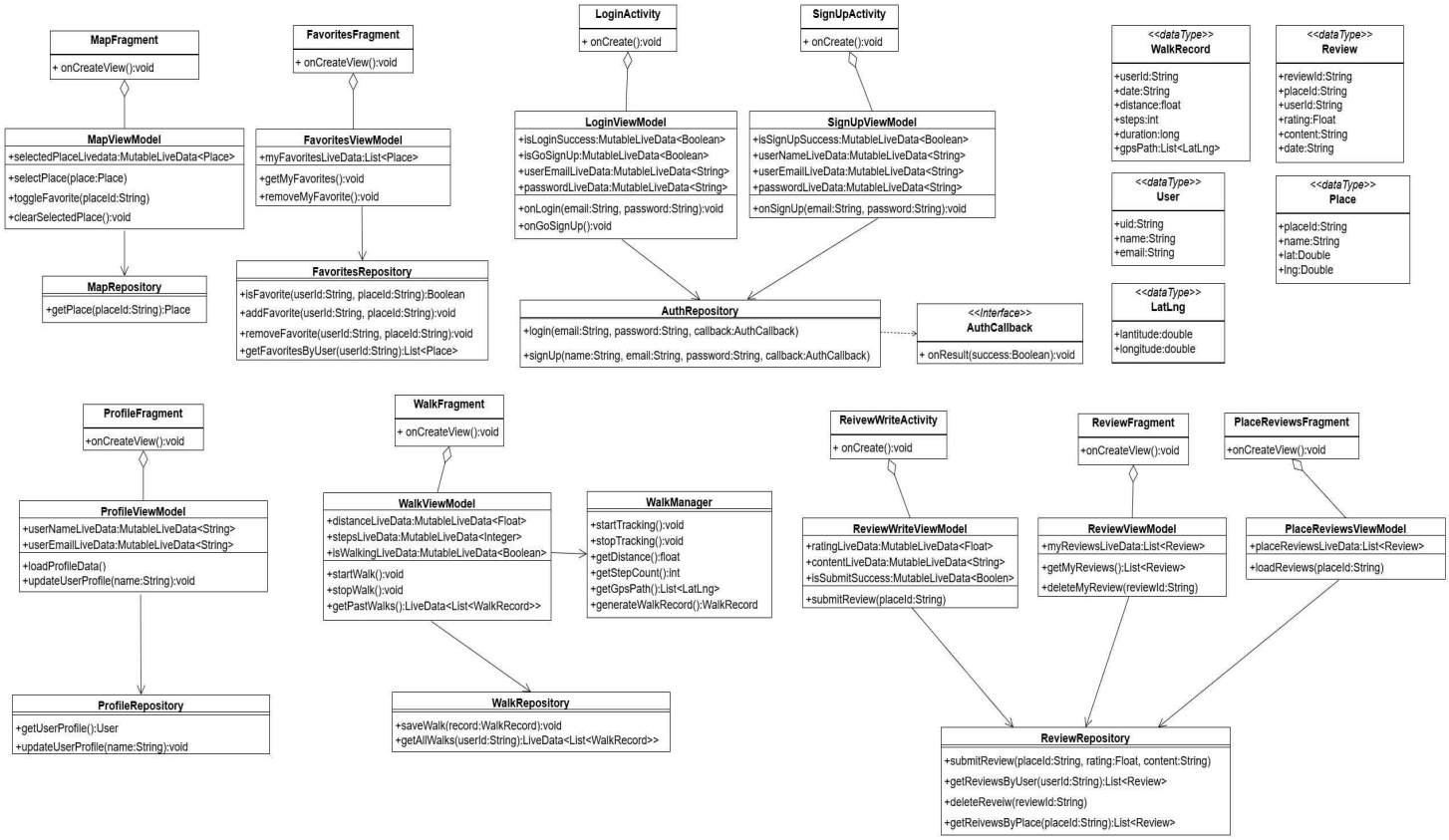
= Contents =

1. Introduction

본 문서는 강아지 산책 기록 앱 개발을 위한 설계 단계를 다룬 보고서이다. 사용자가 강아지와 산책한 경로, 시간, 거리, 걸음 수 등을 실시간으로 기록하고, 산책 장소에 대한 리뷰를 작성하거나 즐겨찾기를 추가하는 기능을 포함한다. 해당 기능에 맞춰 클래스 다이어그램, 시퀀스 다이어그램, 상태 머신 다이어그램을 작성하고 각 다이어그램에 대해 설명한다.

앱의 전체 구조는 MVVM(Model-View-ViewModel) 아키텍처 패턴을 기반으로 구성하였다. UI 로직, 비즈니스 로직, 데이터 처리 로직으로 분리함으로써 유지보수가 쉬우며 각 구성 요소가 독립적으로 역할을 수행하고 협력하도록 구현하였다.

2. Class diagram



[그림1] 클래스 다이어그램

클래스 다이어그램에서 UI를 구성하는 Activity/Fragment 클래스는 생략하였다.

1. User

Attributes	
uid:String	사용자 식별 ID
email:String	사용자의 이메일
name:String	사용자의 이름
Description	
사용자 정보를 저장하고 불러오기 위한 클래스이다.	

2. LoginViewModel

Attributes	
isLoginSuccess:MutableLiveData<Boolean>	로그인의 성공 여부
isGoSignUp:MutableLiveData<Booelan>	회원가입 화면 이동 여부

userEmailLiveData:MutableLiveData<String>	사용자가 입력한 이메일
passowrdLiveData:MutableLiveData<String>	사용자가 입력한 비밀번호
Methods	
onLogin(email:String, password:String):void	로그인 실행
onGoSignUp():void	회원가입 화면 이동
Description	
로그인 화면의 사용자 입력 및 상태를 관리하고 인증 처리 로직을 담당하는 클래스이다.	

3. SignUpViewModel

Attributes	
isSignUpSuccess:MutableLiveData<Boolean>	회원가입 성공 여부
userNameLiveData:MutableLiveData<String>	사용자가 입력한 이름
userEmailLiveData:MutableLiveData<String>	사용자가 입력한 이메일
passwordLiveData:MutableLiveData<String>	사용자가 입력한 비밀번호
Methods	
onSignUp(email:String, password:String):void	회원가입 실행
Description	
사용자 입력값(이름, 이메일, 비밀번호)을 관리하고, 회원가입 요청 처리 로직을 담당하는 클래스이다.	

4. AuthRepository

Methods	
login(email:String, password:String, callback:AuthCallback)	이메일과 비밀번호로 로그인을 시도한 후 결과를 콜백으로 반환
signUp(name:String, email:String, password:String, callback:AuthCallback)	이름, 이메일, 비밀번호로 회원가입을 시도한 후 결과를 콜백으로 반환
Description	
로그인 및 회원가입 기능을 외부 인증 서비스와 연동하여 처리하는 사용자 인증 기능을 담당하는 클래스이다.	

5. WalkRecord

Attributes	
userId:String	사용자의 ID
date:String	산책 날짜
distance:float	산책한 거리
steps:int	산책한 걸음 수
duration:long	산책한 시간
gpsPath:List<LatLng>	산책한 경로
content:String	산책 기록 메모
Description	

산책 기록에 대한 정보를 담고 있는 클래스이다.

6. WalkSharedViewModel

Attributes	
distanceLiveData:MutableLiveData<Float>	현재까지 산책한 거리
stepsLiveData:MutableLiveData<Integer>	현재까지 산책한 걸음 수
isWalkingLiveData:MutableData<Boolean>	산책 중인지 여부 확인
walkRecord:MutableLiveData<WalkRecord>	산책 기록 객체
Methods	
startWalk():void	산책 기록 시작
stopWalk():void	산책 기록 중단
updateMemo(content:String):void	산책 기록에 메모 추가하기
saveWalkRecord():void	산책 기록 저장
Description	
실시간 거리, 걸음 수, 산책 진행 여부 등을 관리하고 사용자의 산책 기록을 수정하고 저장하는 클래스이다.	

7. WalkManager

Methods	
startTracking():void	위치 추적 시작
stopTracking():void	위치 추적 종료
getDistance():float	실시간 거리 계산
getStepCount():int	실시간 걸음 수 계산
getGpsPath():List<LatLng>	경로 반환
generateWalkRecord():WalkRecord	WalkRecord 객체 생성
Description	
GPS, 시간, 걸음 수 등 산책 중 발생하는 데이터 수집 및 로직을 담당하는 클래스이다.	

8. WalkRepository

Methods	
saveWalk(record:WalkRecord):void	산책 기록을 DB에 저장
getAllWalks(userId:String):LiveData<List<WalkRecord>>	모든 산책 기록 불러오기
getWalksByDate(userId:String, date:String):LiveData<List<WalkRecord>>	특정 날짜의 산책 기록 불러오기
deleteWalkRecord(userId:String, recordId:String):void	특정 산책 기록 삭제
editWalkRecord(userId:String, recordId:String, newMemo:String):void	특정 산책 기록 수정
Description	

산책 기록(WalkRecord)을 저장하고 불러오는 역할을 담당하는 클래스이다.

9. WalkListViewModel

Attributes	
selectDateLiveData:MutableLiveData<String>	선택된 날짜
WalkRecordsByDate:LiveData<List<WalkRecord>>	특정 날짜에 대한 산책 기록 리스트
Methods	
getPastWalks():LiveData<List<WalkRecord>>	과거의 산책 기록 불러오기
setSelectedDate(date:String):void	선택된 날짜 저장하기
Description	
특정 날짜에 대한 산책 기록들을 출력하여 조회할 수 있도록 하는 클래스이다.	

11. DetailWalkViewModel

Attributes	
walkRecord:MutableLiveData<WalkRecord>	산책 기록 객체
Methods	
getWalkRecord():LiveData<WalkRecord>	산책 기록 불러오기
editWalkRecord():void	산책 기록 메모 수정하기
saveWalkRecord():void	산책 기록 저장
deleteWalkRecord():void	산책 기록 삭제
Description	
산책 기록 리스트의 각각의 산책 기록 객체를 관리하는 클래스이다.	

12. MapViewModel

Attributes	
selectedPlaceLiveData:MutableLiveData<Place>>	사용자가 선택한 장소
Methods	
selectPlace(place:Place):void	사용자가 선택한 장소 정보 불러오기
clearSelectedPlace():void	선택된 장소 초기화
toggleFavorite(placeId:String)	선택된 장소를 즐겨찾기 추가/해제
Description	
지도 화면에서 필요한 장소 데이터를 불러오고 관리하는 클래스이다.	

13. MapRepository

Methods	
getPlaces(userId:String):List<Place>	사용자가 즐겨찾기한 장소 목록 반환
Description	

즐겨찾기 관련 데이터를 가져오거나 저장하는 클래스이다.

14. ReviewWriteViewModel

Attributes	
ratingLiveData:MutableLiveData<Float>	사용자가 입력한 별점
contentLiveData:MutableLiveDAta<String>	사용자가 입력한 리뷰 내용
isSubmitSuccess:MutableLiveData<Boolen>	리뷰 작성 결과 여부
Methods	
submitReview(placeId:String)	리뷰 데이터 저장 요청
Description	
리뷰 작성 상태를 관리한다. 입력된 리뷰 정보를 처리하고 저장 요청을 하는 클래스이다.	

15. ReviewViewModel

Attributes	
myReviewsLiveData:List<Review>	사용자가 작성한 리뷰 목록
Methods	
getMyReviews():List<Review>	사용자의 리뷰 목록 불러오기
deleteMyReview(reviewId:String)	사용자의 리뷰 삭제
Description	
사용자가 작성한 리뷰 목록을 불러오고 삭제하는 역할을 담당한 클래스이다.	

16. PlaceReviewsViewModel

Attributes	
placeReivewsLiveData:List<Reivew>	특정 장소에 대한 리뷰 목록
selectedPlaceId:MutableLiveData<String>	선택된 장소의 ID
Methods	
loadReviews(placeId:String):void	특정 장소에 대한 리뷰 목록 불러오기
Description	
특정 장소에 대한 리뷰 목록을 관리하는 클래스이다.	

17 . ReviewRepository

Methods	
submitReview(placeId:String, rating:Float, content:String):List<Review>	리뷰 정보 저장 처리
getReviewsByUser(userId:String):List<Review>	사용자의 리뷰 목록 반환
deleteReview(userId:String, reviewId:String)	리뷰 삭제
getReviewsByPlace(placeId:String):List<Review>	장소의 리뷰 목록 반환
Description	
리뷰 관련 데이터를 가져오거나 저장하는 클래스이다.	

18. FavoritesViewModel

Attributes	
myFavoritesLiveData:List<Place>	사용자의 즐겨찾기 장소 목록
Methods	
getMyFavorites(userId:String):void	사용자의 즐겨찾기 장소 목록 불러오기
removeMyFavorite(placeId:String):void	즐겨찾기 목록에서 장소 삭제
Description	
즐겨찾기 장소 목록 데이터를 관리하고 불러오기 및 삭제 요청을 하는 클래스이다.	

19. FavoritesRepository

Methods	
isFavorite(userId:String, placeId:String):Boolean	해당 장소를 사용자가 즐겨찾기 한 상태인지 확인
addFavorite(userId:String, placeId:String):void	해당 장소를 즐겨찾기에 추가
removeFavorite(userId:String, placeId:String):void	해당 장소를 즐겨찾기에서 삭제
getFavoritesByUser(userId:String):List<Place>	사용자의 즐겨찾기 장소 목록을 반환
Description	
리뷰 관련 데이터를 가져오거나 저장하는 클래스이다.	

20. Place

Attributes	
placeId:String	장소 고유 ID
name:String	장소 이름
location:LatLng	장소의 위치
Description	
장소 정보를 나타내는 클래스이다.	

21. Review

Attributes	
reviewId:String	리뷰 고유 ID
placeId:String	해당 장소의 고유 ID
userId:String	리뷰 작성자 ID
rating:Float	별점
content:String	리뷰 내용
date:String	작성 일자
Description	
리뷰 정보를 나타내는 클래스이다.	

22. LatLng

Attributes

lantitude:double	장소의 위도
longitude:double	장소의 경도
Description	
위도와 경도 좌표를 저장하는 클래스이다.	

23. ProfileViewModel

Attributes	
userNameLiveData:MutableLiveData<String>	사용자 이름
userEmailLiveData:MutableLiveData<String>	사용자 이메일
Methods	
loadProfileData():void	사용자 정보 불러오기
updateUserProfile(name:String):void	사용자 정보 수정 요청
Description	
프로필 화면의 상태와 로직을 담당하며 사용자 정보 조회와 수정을 관리하는 클래스이다.	

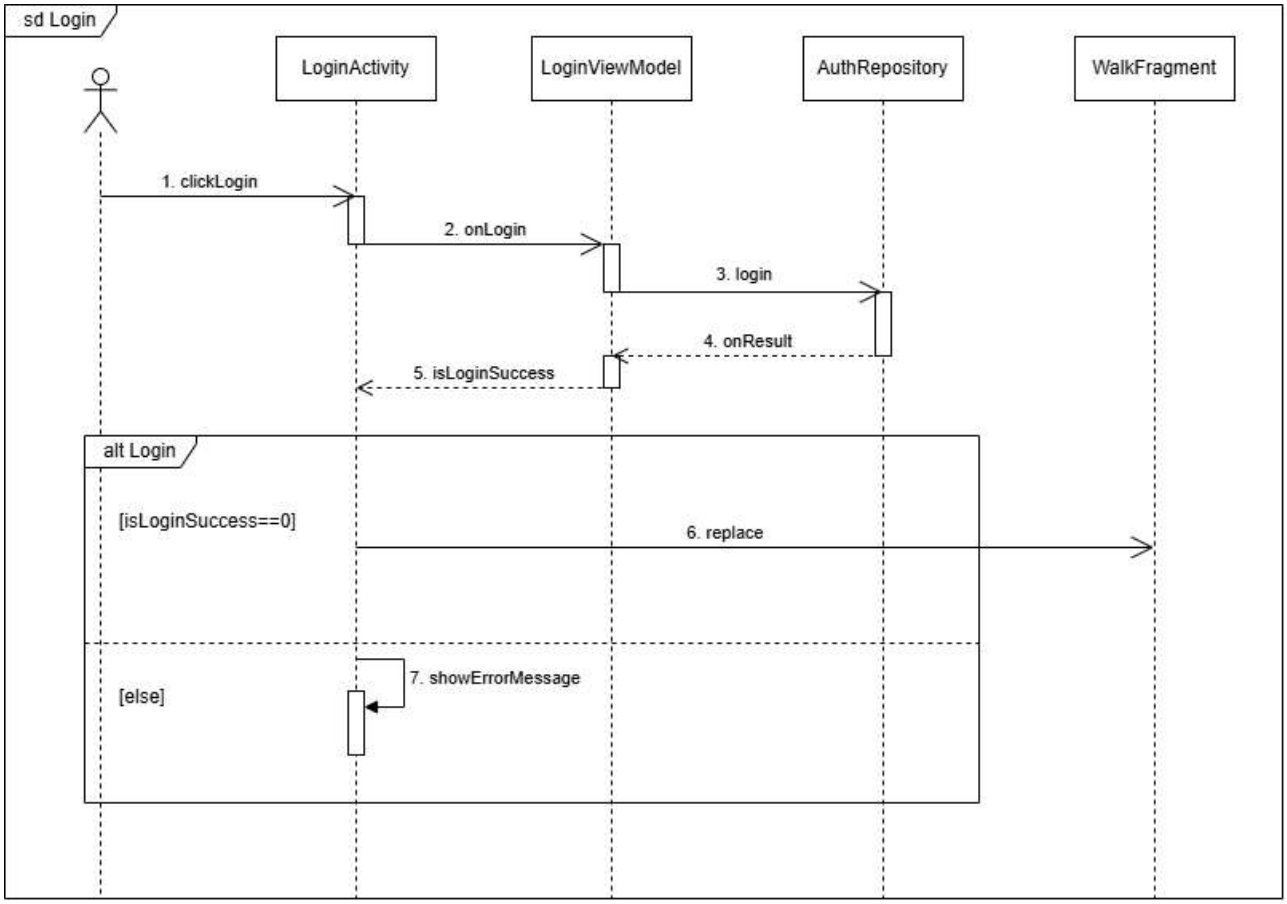
24. ProfileRepository

Methods	
getUserProfile():User	사용자 정보를 반환
updateUserProfile(name:String):void	사용자 정보 수정
Description	
사용자 프로필과 관련된 데이터 처리를 담당하는 클래스이다.	

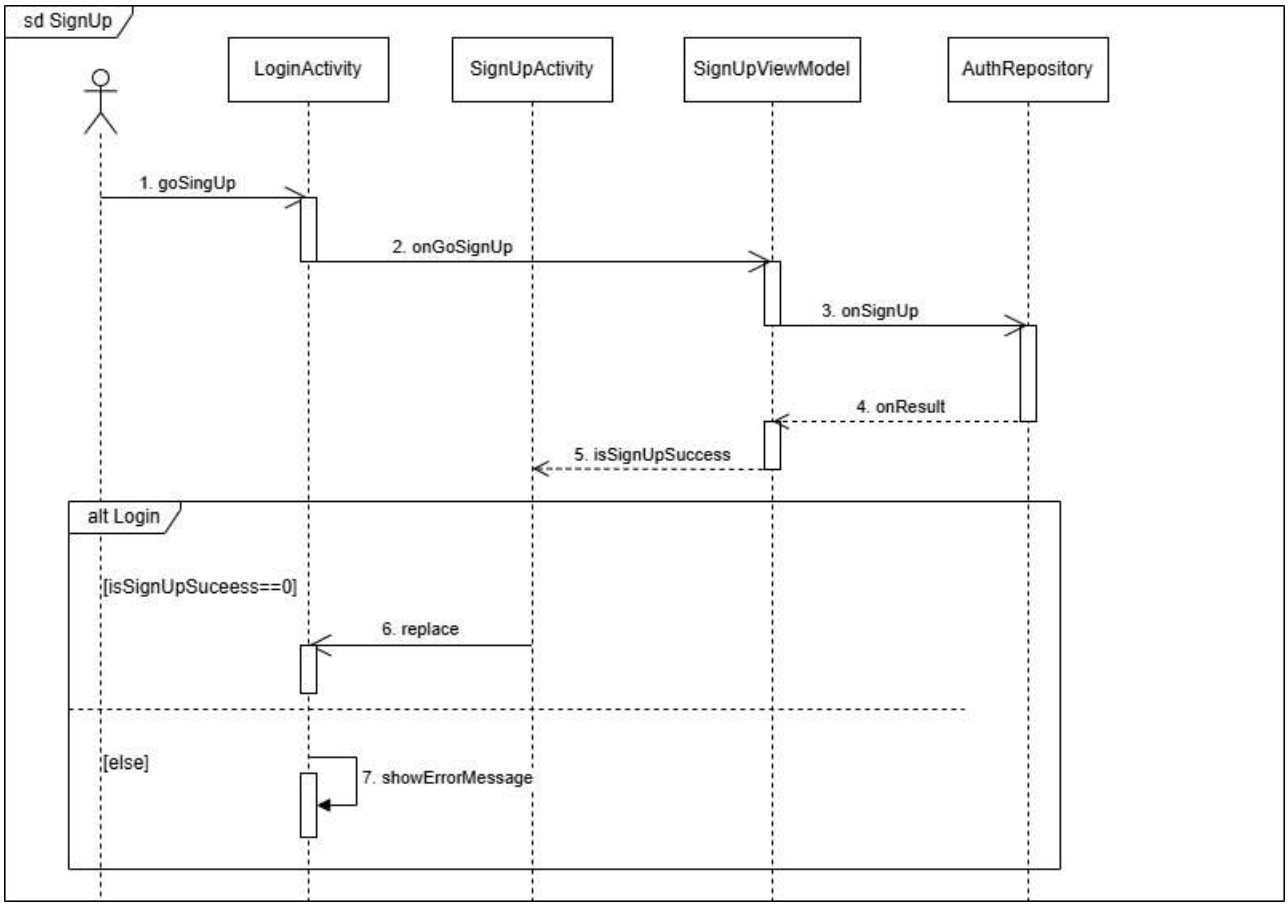
25. WalkPath

Attributes	
pathPoints:List<LatLng>	산책한 경로
Description	
산책 경로 전체를 나타내는 클래스이다.	

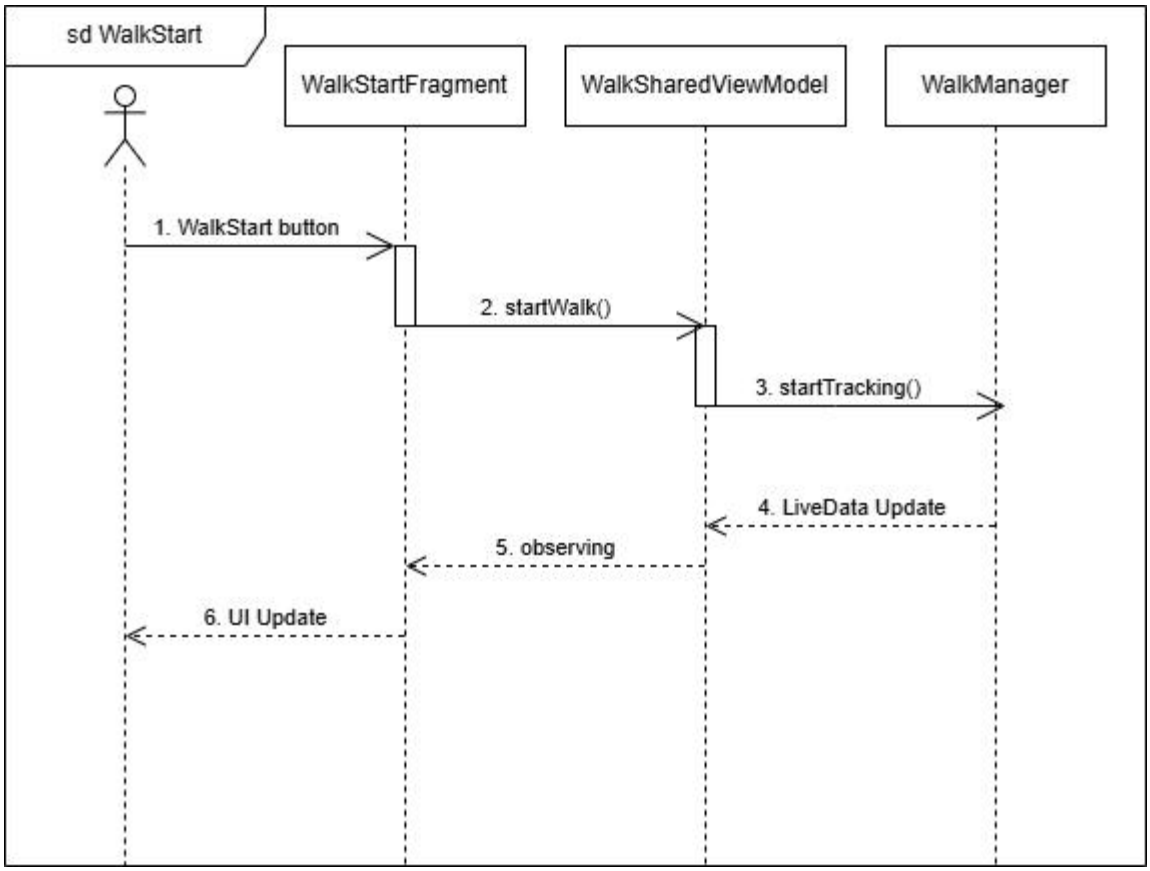
3. Sequence diagram



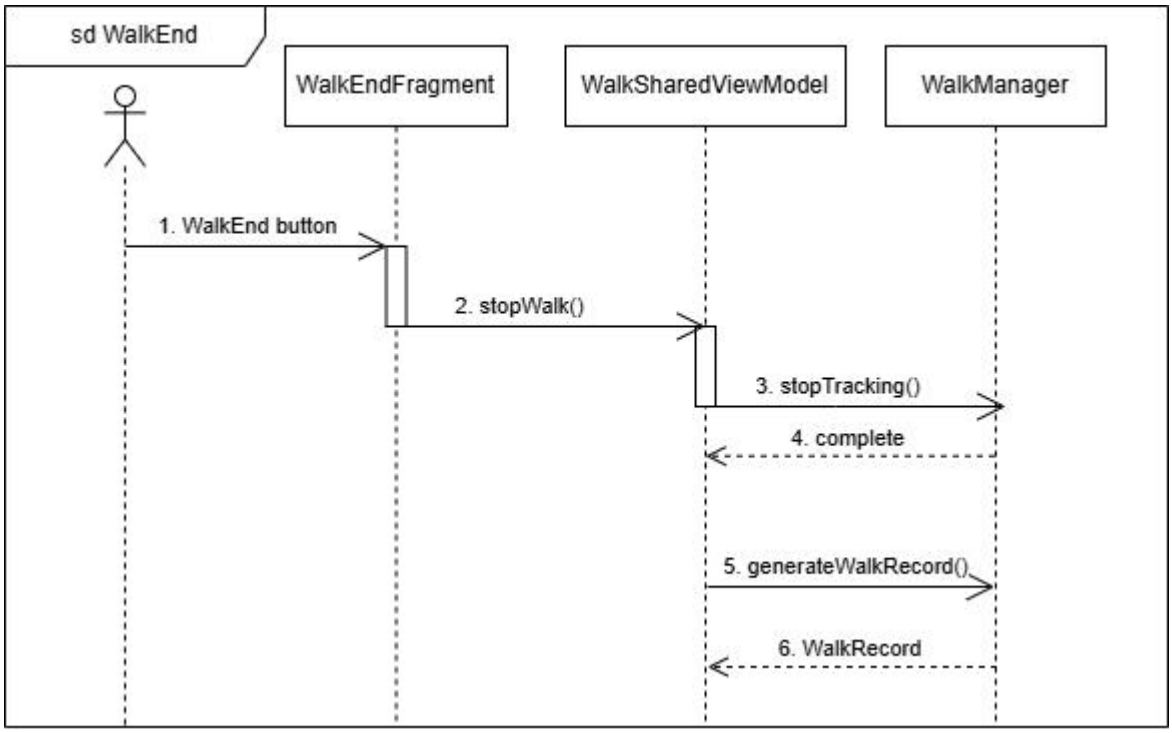
사용자는 이메일과 비밀번호를 입력하고 로그인 버튼을 클릭한다. LoginActivity에서 로그인 버튼 클릭 이벤트를 감지하고 LoginViewModel의 onLogin() 함수를 호출한다. login(email,password) 함수가 호출되고 콜백을 통해 결과를 받는다. LoginViewModel은 로그인 결과에 따라 LiveData 값을 설정한다. LoginActivity는 해당 LiveData를 observe하고 있기 때문에 값이 변경되면 자동으로 반응한다. 따라서 로그인에 성공했을 경우, 홈 화면으로 이동하고, 실패한 경우 에러 메시지를 표시한다.



사용자가 로그인 화면에서 회원가입 버튼을 클릭하면 SignUpActivity 화면으로 이동한다. 사용자가 회원가입 정보를 입력하고 회원가입 버튼을 누르면 onSignUp()이 호출되고 AuthRepository에서 회원가입을 실행하고 처리 결과를 onResult 콜백으로 반환한다. isSignUpSuccess 값에 따라 화면 흐름이 달라진다. 성공 시 로그인 화면으로 전환되고, 실패할 시 오류 메시지를 출력한다.

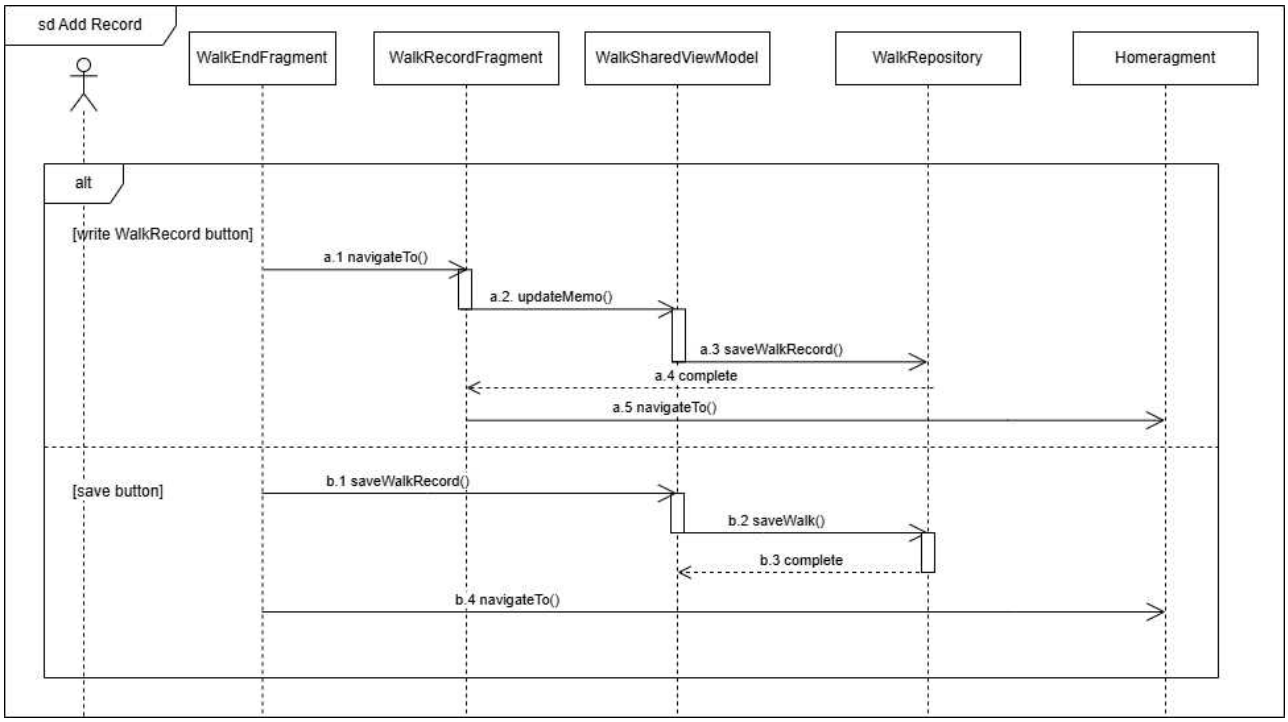


산책 시작 기능이 동작하는 과정이다. 사용자가 WalkStartFragment에서 '산책 시작' 버튼을 누른다. WalkStartFragment는 사용자 입력을 처리하여 WalkSharedViewModeldml startWalk() 메서드를 호출한다. ViewModel은 내부적으로 WalkManager의 startTracking()을 호출하여 위치 추적을 시작한다. WalkManager는 GPS 및 걸음 수 데이터를 수집하고 그 결과를 ViewModel의 LiveData 객체에 업데이트 한다. WalkStartFragment는 LiveData를 관찰하고 있는 상태이므로, 데이터가 변경되면 자동을 콜백을 받아 UI를 갱신한다.



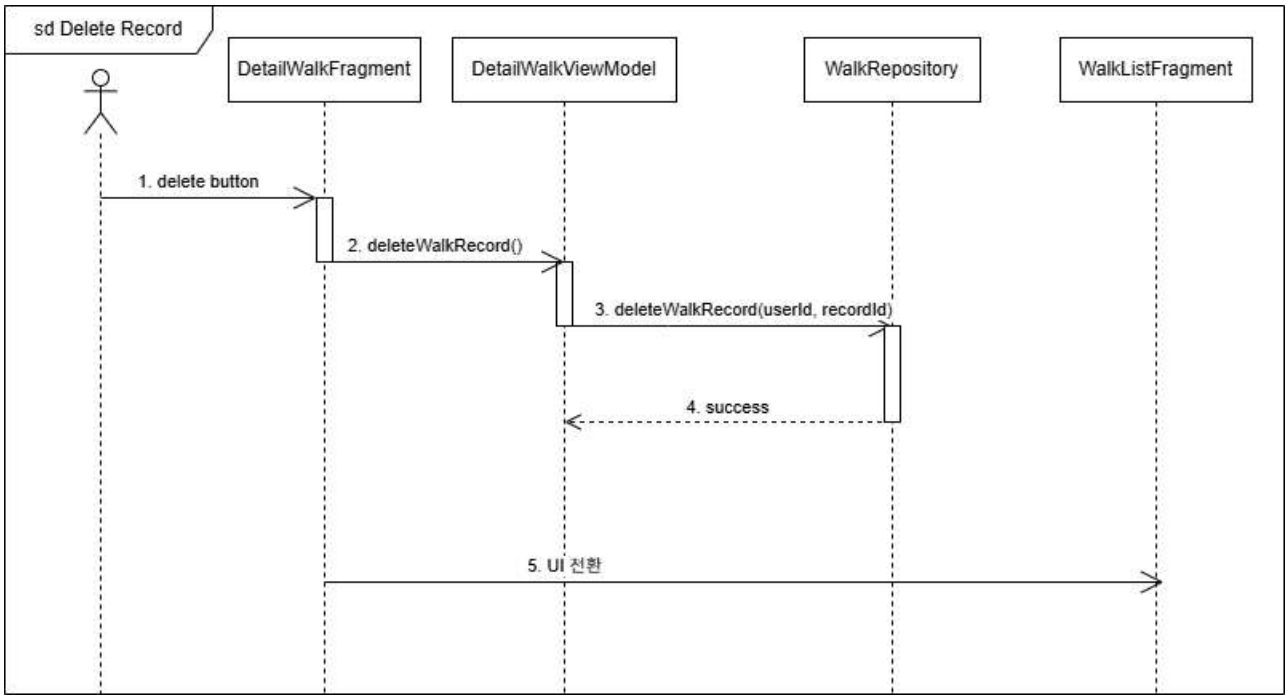
산책 종료 시점의 동작 과정을 표현한다. 사용자가 WalkEndFragment에서 '산책 종료' 버튼을 클릭한다. WalkEndFragment는 이를 처리하여 WalkSharedViewModel의 stopWalk() 메서드를 호출한다. ViewModel 내부에서 WalkManager의 stopTracking()이 호출되며 위치 추적이 중단된다. WalkManager는 추적 중이던 데이터 수집을 마무리하고 완료 신호를 반환한다.

이후 WalkSharedViewModel은 generateWalkRecord()를 호출하여 해당 산책에 대한 기록 데이터를 생성한다. 생성된 WalkRecord 객체는 ViewModel로 반환되어, 이후 사용자 선택을 기다리게 된다.

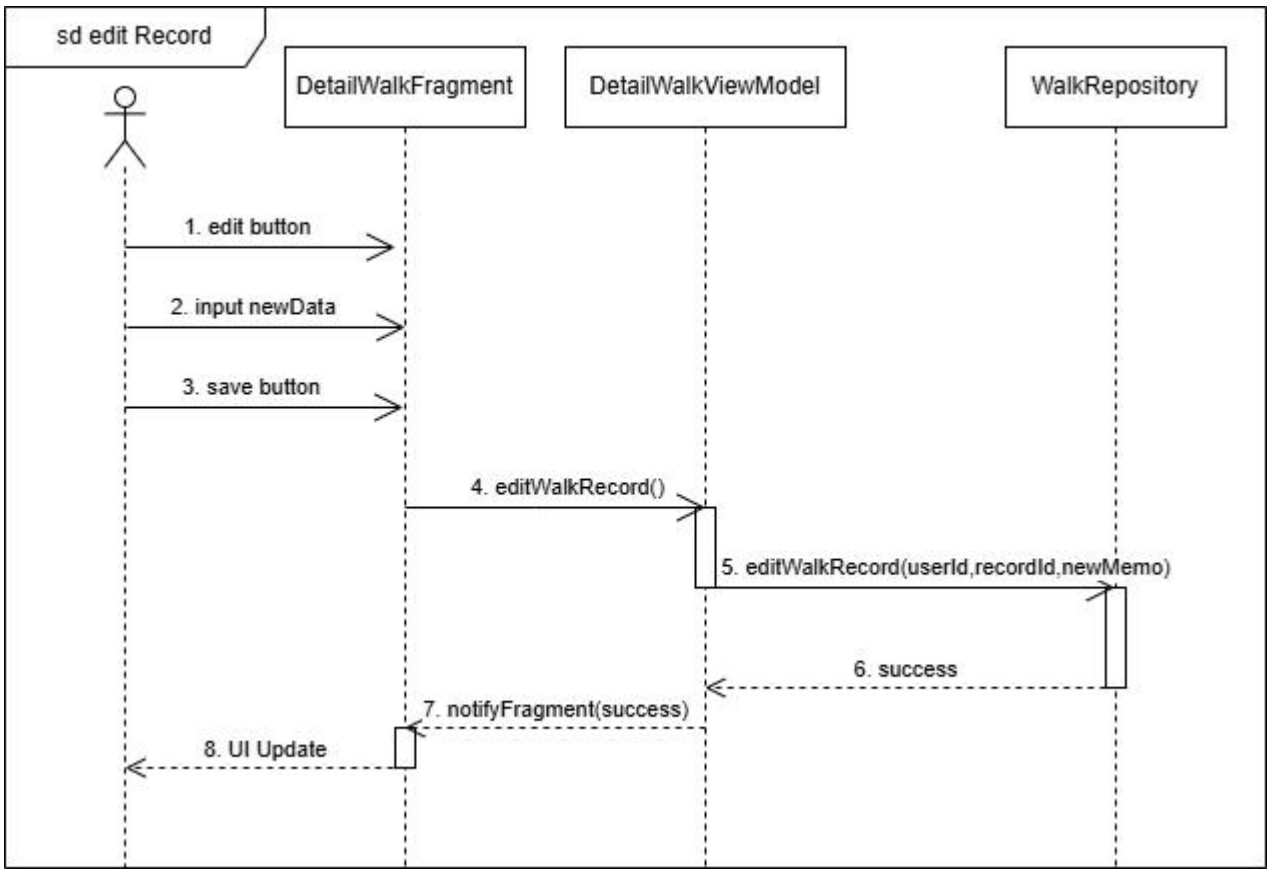


사용자가 산책을 종료한 뒤, 산책 기록을 저장하는 과정이다. 사용자가 WalkEndFragment에서 '산책 기록 쓰러가기' 버튼을 클릭하면 WalkRecordFragment로 화면이 전환된다. WalkRecordFragment에서 사용자가 메모를 작성하면 ViewModel의 updateMemo() 메서드를 통해 메모 내용이 WalkRecord에 반영된다. 사용자가 '저장' 버튼을 누르면 saveWalkRecord() 메서드를 통해 저장이 시작되며 ViewModel은 내부적으로 WalkRepository.saveWalk()를 호출하여 저장을 수행한다. 저장이 완료되면 ViewModel은 완료 메시지를 전송하고 이후 WalkRecordFragment는 홈 화면으로 전환한다.

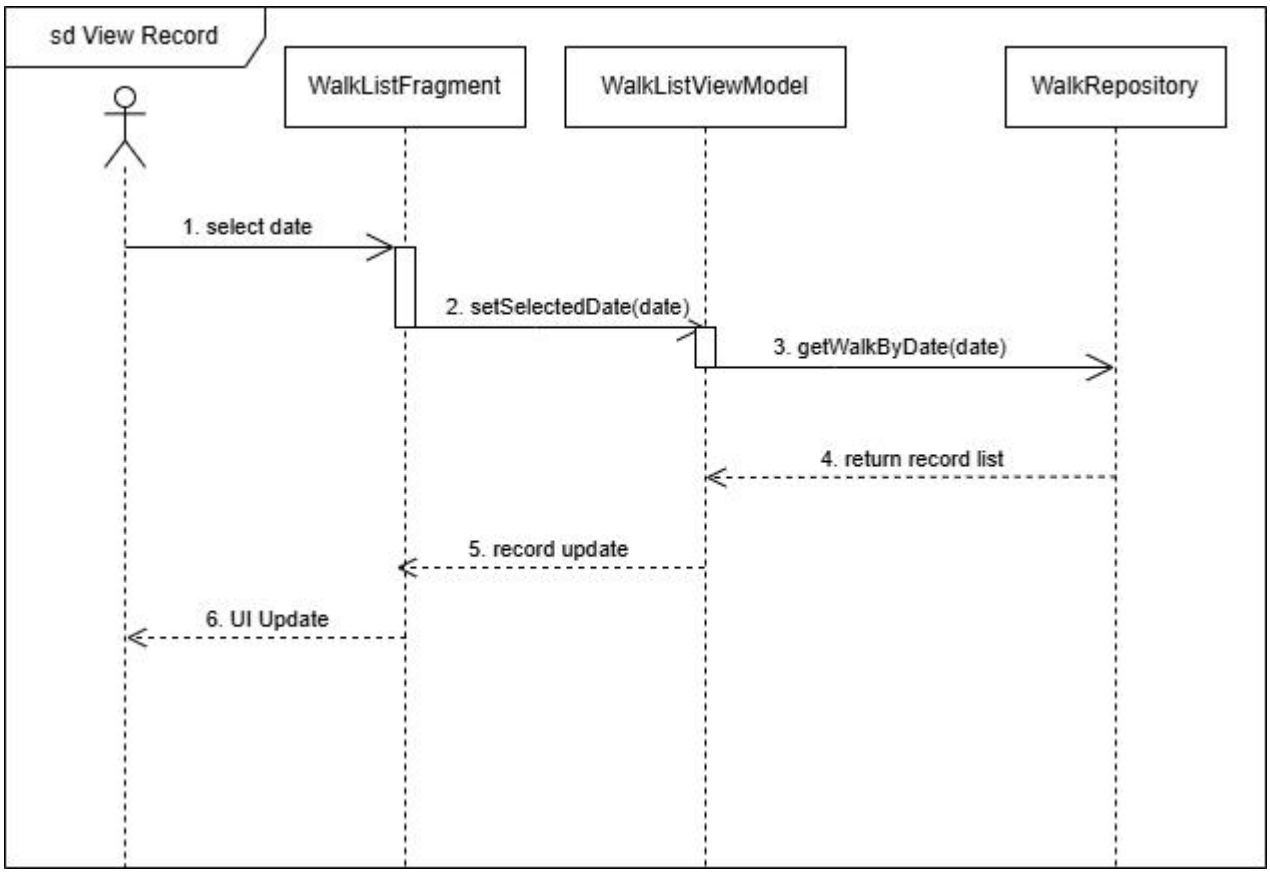
사용자가 산책을 종료한 뒤 메모 없이 바로 저장을 선택한다면 WalkEndFragment에서 곧바로 saveWalkRecord()가 호출된다.



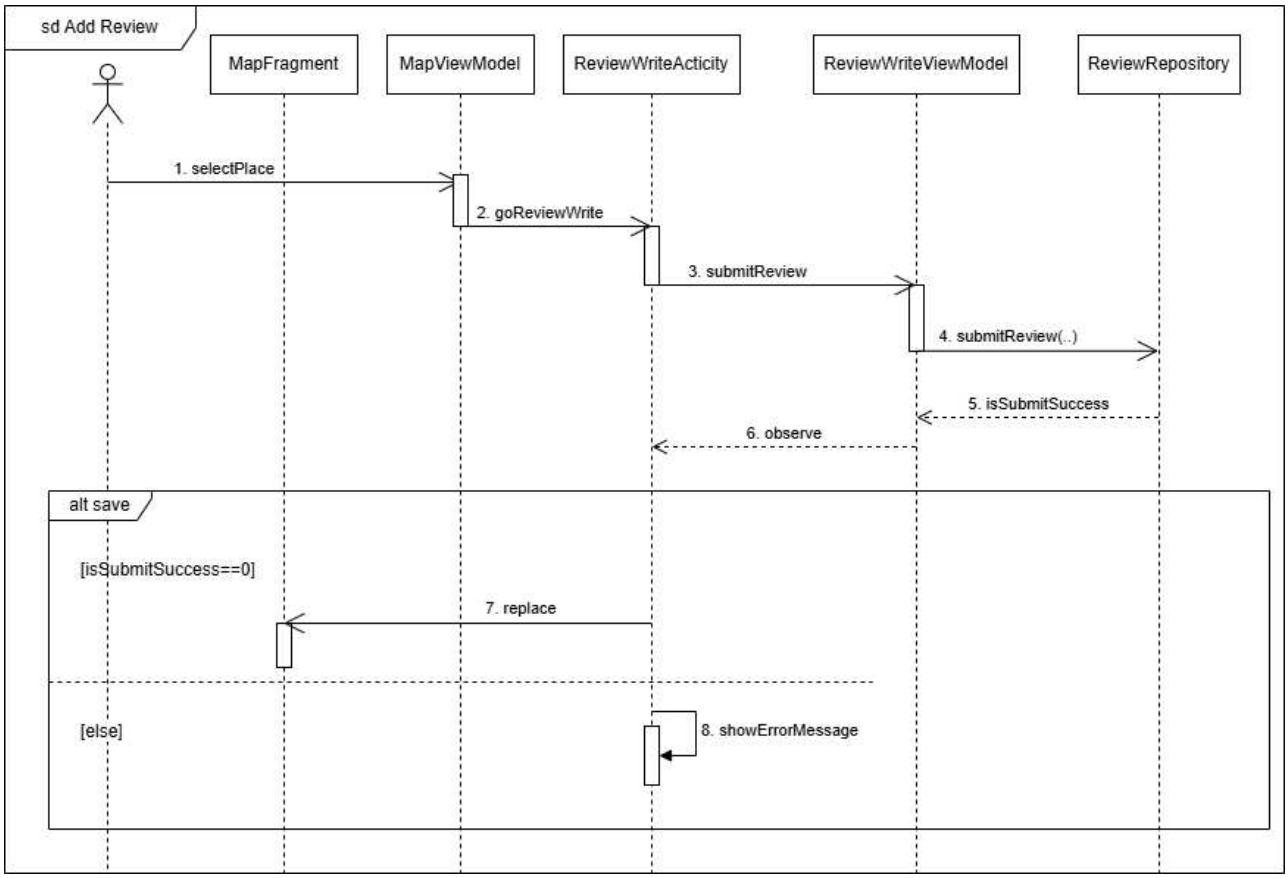
사용자가 산책 기록을 삭제하는 과정이다. 사용자가 DetailWalkFragment에서 삭제 버튼을 누르면 DetailWalkViewModel의 deleteWalkRecord() 메소드가 호출된다. 해당 메소드는 WalkRepository에 사용자 ID와 산책 기록 ID를 넘겨 삭제 요청을 한다. 삭제가 완료되면 WalkListFragment로 화면이 전환된다.



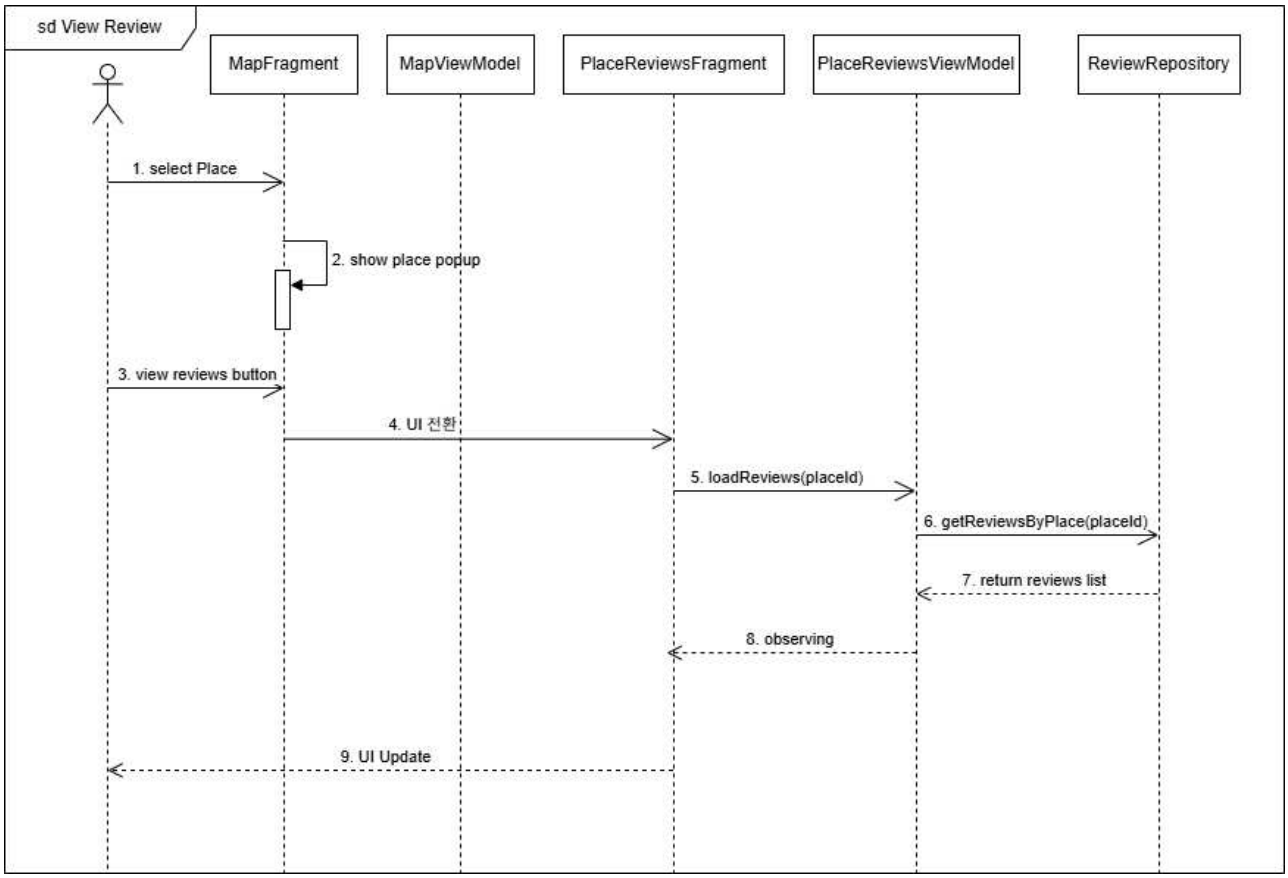
사용자가 산책 기록을 수정하는 과정이다. 사용자는 먼저 수정 버튼을 눌러 수정 화면으로 진입하고, 새로운 내용을 입력한 뒤 저장 버튼을 누른다. 이후 DetailWalkFragment는 ViewModel의 editWalkRecord() 메서드를 호출하고, ViewModel은 사용자 ID, 기록 ID, 수정된 메모 데이터를 포함해 WalkRepository에 수정 요청을 전달한다. 저장이 성공적으로 완료되면 ViewModel은 Fragment에 성공 여부를 알리고, Fragment는 이를 기반으로 UI를 갱신하여 사용자에게 수정 결과를 반영한다.



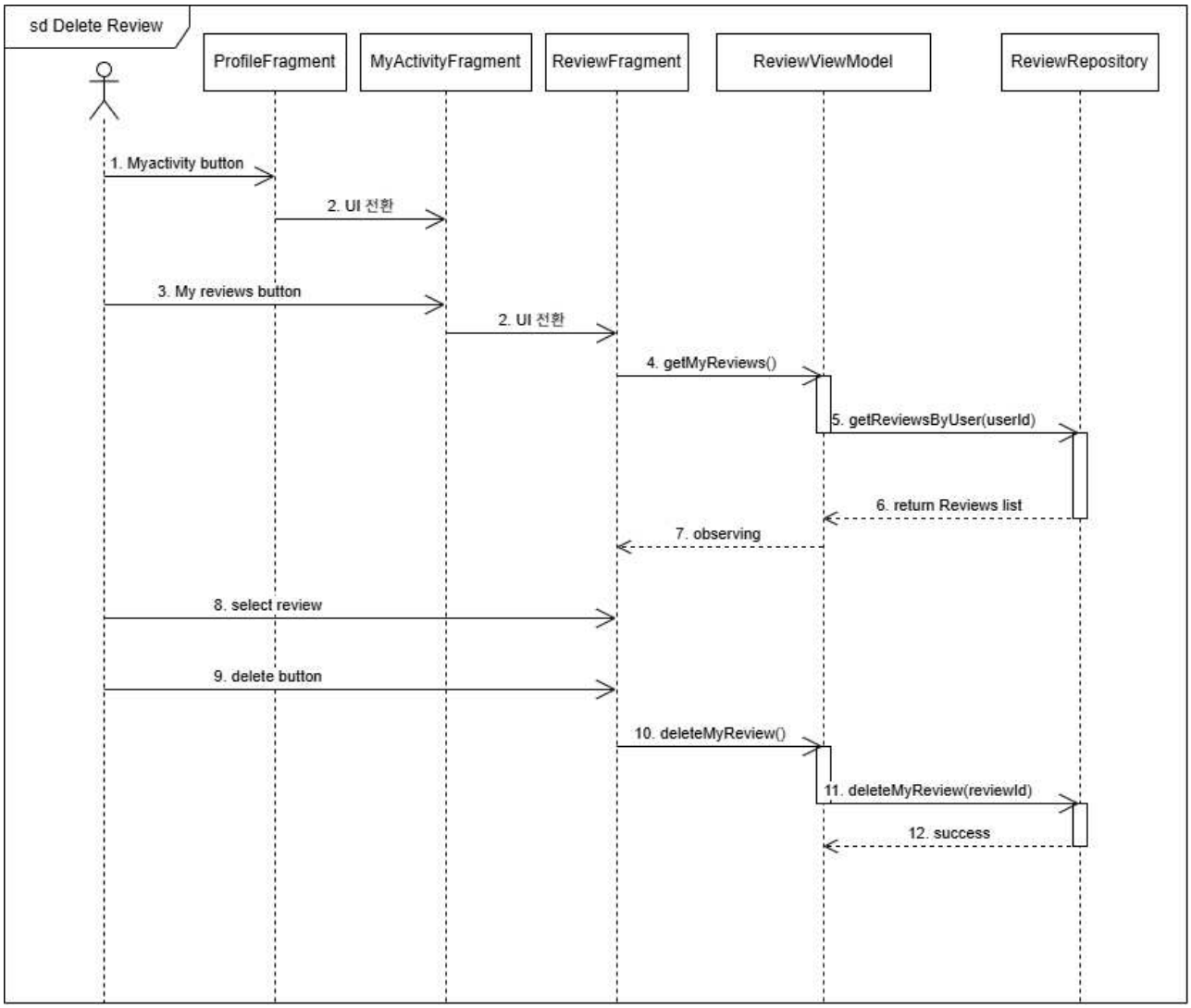
사용자가 특정 날짜의 산책기록을 조회하는 과정이다. 사용자는 홈 화면의 산책달력에서 날짜를 선택하고, WalkListFragment는 ViewModel의 setSelectedDate() 함수를 호출한다. ViewModel은 WalkRepository의 getWalksByDate() 함수를 호출하여 해당 날짜의 산책기록을 요청하고, 결과는 walkRecordsByDate LiveData에 반영된다. WalkListFragment는 이 값을 관찰하고 있어 자동으로 UI에 해당 날짜의 기록 목록이 표시된다.



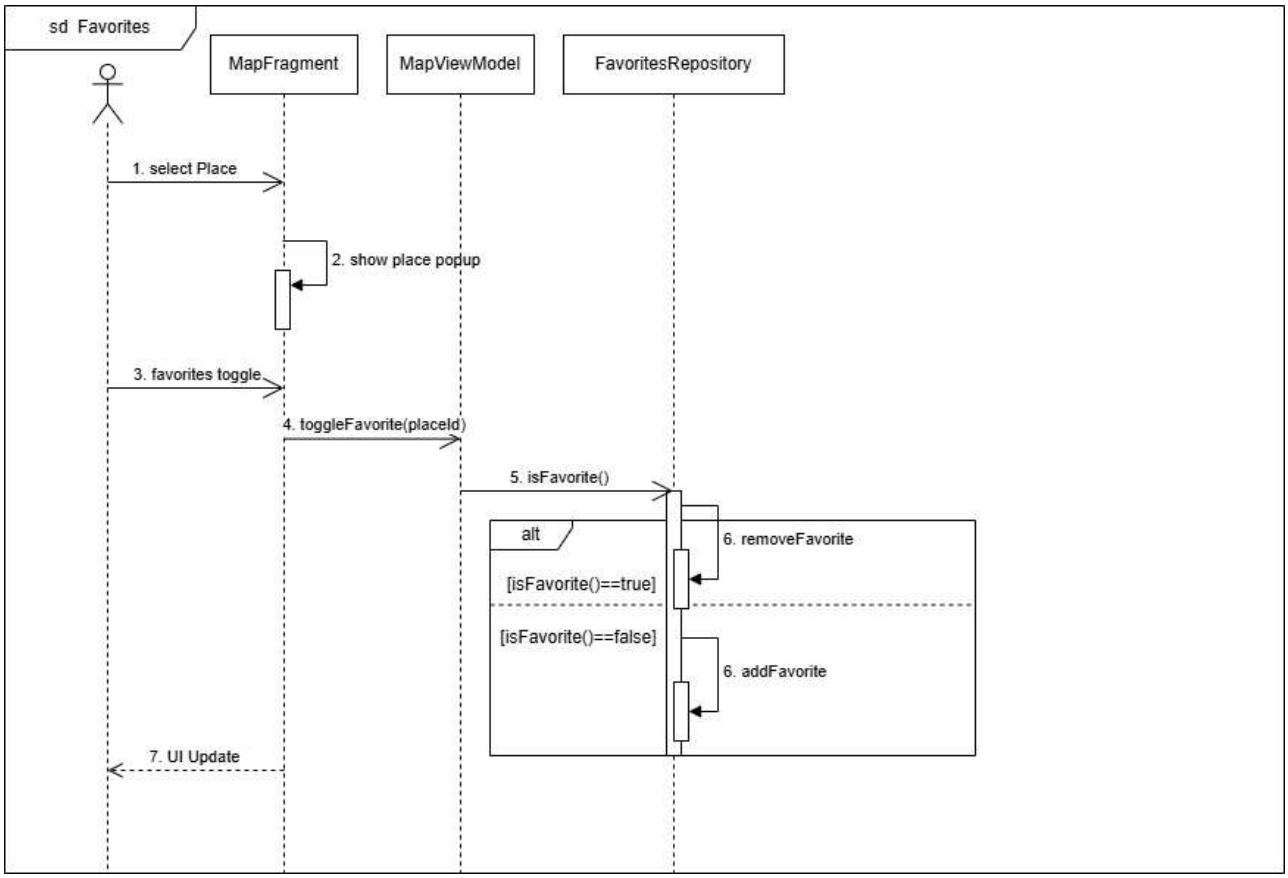
지도에서 특정 장소에 대한 리뷰를 작성하는 과정이다. 사용자는 지도 화면에서 특정 장소를 선택한다. 리뷰 쓰기 버튼을 누르면 ReviewWriteActivity로 화면이 전환된다. 사용자는 별점, 내용 등을 작성하고 등록 버튼을 누르면 submitReview()가 호출된다. ReviewRepository에 리뷰 정보를 저장하고 결과값을 isSubmitSuccess를 통해 받는다. ReviewWriteActivity는 해당 값에 따라 지도 화면으로 전환하거나 오류 메시지를 표시한다.



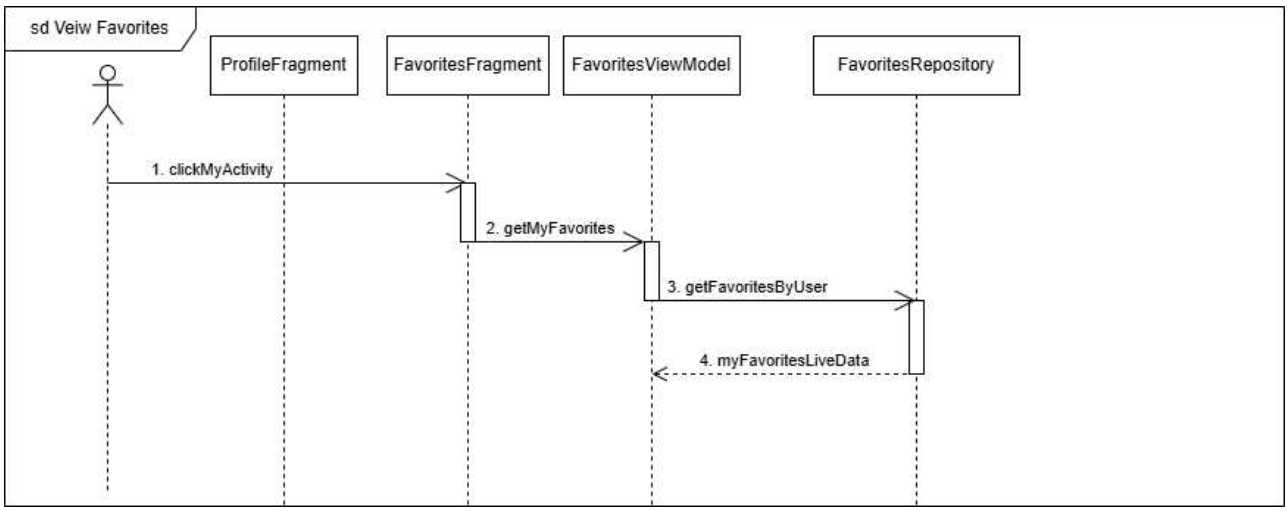
사용자가 지도에서 장소를 선택하여 해당 장소의 리뷰 목록을 조회하는 과정이다. 사용자가 장소 선택 시 MapFragment는 장소 정보를 팝업으로 표시한다.
사용자가 팝업에서 리뷰 보기 버튼을 누르면 PlaceReviewsFragment로 화면이 전환되며, 해당 Fragment는 PlaceReviewsViewModel의 loadReviews(placeId) 함수를 호출한다. getReviewsByPlace(placeId)`를 통해 리뷰 목록을 요청하고, 받은 결과는 LiveData에 반영되어 Fragment에서 관찰된다. 이를 통해 UI에는 자동으로 해당 장소의 리뷰 목록이 표시된다.



사용자가 자신이 작성한 리뷰를 지우는 과정이다. 사용자가 프로필 화면에서 '나의 활동 목록' 버튼을 누르면 MyActivityFragment 화면으로 전환된다. '나의 리뷰 목록' 버튼을 누르면 리뷰 목록 화면으로 전환된다. 삭제하려는 리뷰를 선택하고 삭제 버튼을 누르면 deleteMyReview() 함수를 호출하여 실행한다.

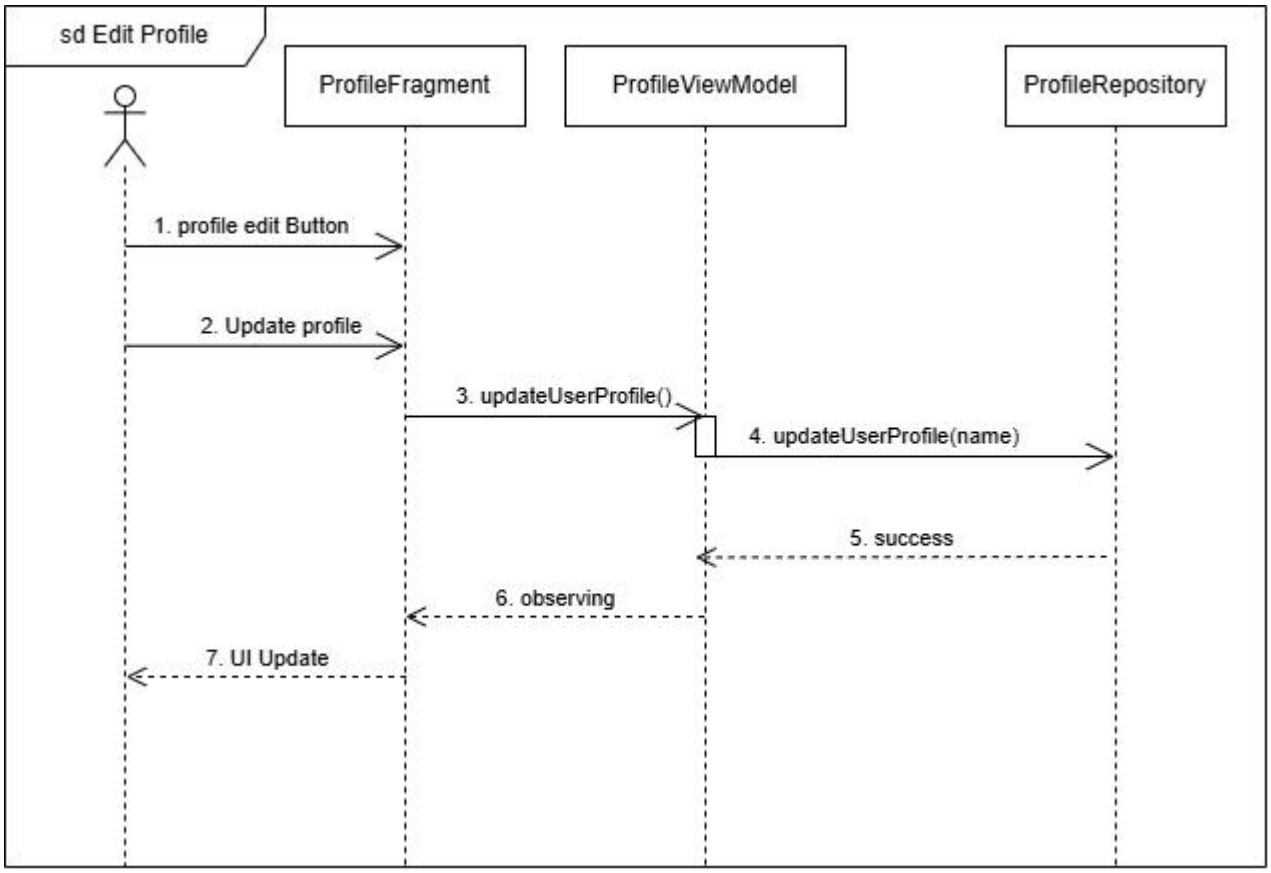


사용자가 지도에서 즐겨찾기 기능을 해제/설정하는 과정이다. 사용자가 장소를 선택하고 즐겨찾기 버튼을 누르면 MapViewModel의 toggleFavorite(placeId) 함수가 호출된다. ViewModel은 isFavorite()을 통해 해당 장소가 즐겨찾기 상태인지 확인한다.
결과가 true인 경우 removeFavorite()을, false인 경우 addFavorite()을 호출하여 상태를 반전시킨다. 처리 결과에 따라 UI는 즐겨찾기 아이콘 상태를 업데이트한다.



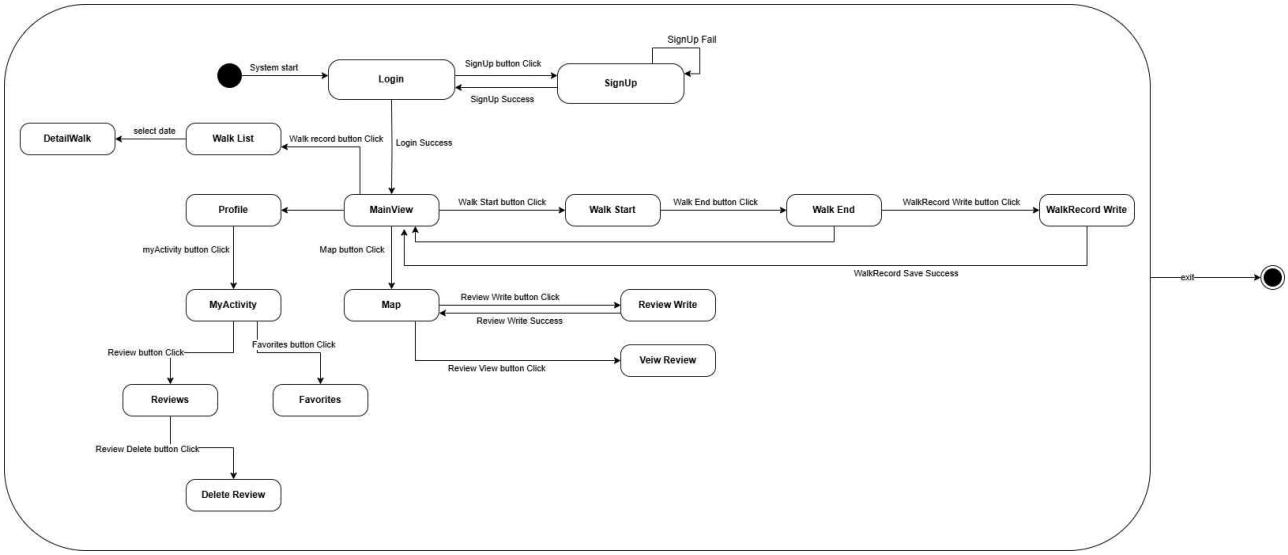
사용자가 자신이 저장한 즐겨찾기 장소를 조회하는 과정이다. 사용자는 프로필 화면에서 즐겨찾기 목록으로 이동한다. FavoritesViewModel의 getMyFavorites() 함수를 호출

하여 FavoriteRepository에서 getFavoritesByuser() 함수를 호출한다. 해당 함수의 결과 값은 myFavoritesLiveData에 반영되고 FavoritesFragment는 해당 값을 관찰하고 있어 자동으로 UI에 반영한다.



사용자가 자신의 프로필을 수정하는 과정이다. 프로필 화면에서 '프로필 수정' 버튼을 누르고 입력값을 수정한 뒤 완료를 누르면 updateUserProfile() 함수가 호출된다. DB에 저장된 사용자의 이름을 수정하고 ViewModel로 완료 메시지를 전달한다. ViewModel의 userNameLiveData를 수정하고 이는 자동으로 UI에 반영된다.

4. State machine diagram



사용자가 시스템 실행 시 로그인 화면에서 시작된다. 해당 화면에서 회원가입 버튼을 누르면 회원가입 화면으로 넘어간다. 로그인 화면에서 입력값을 작성하고 로그인 버튼을 누르면 메인화면으로 넘어간다. 메인화면에서는 산책 시작 기능을 활성화 할 수 있고 프로필 화면, 지도 화면으로 넘어갈 수 있다. 산책 시작 버튼을 누르면 산책 중 상태로 전환되고 GPS, 걸음 수 등 데이터를 수집한다. 산책 종료 버튼을 눌러야 데이터 수집이 중단되며, 사용자가 산책 정보를 가지고 추가적으로 글을 쓰는 경우에는 산책 기록 작성하기 화면으로 넘어간다. 산책 기록을 저장한 후에 다시 메인화면으로 전환된다. 만약 산책 기록에 글을 작성하지 않으면 저장 후 바로 메인화면으로 전환한다. 메인화면에서 지도 버튼을 누르면 지도 화면이 출력되며 사용자가 특정 장소를 선택하고 리뷰 쓰기 버튼을 누르면 해당 장소에 대한 리뷰를 쓸 수 있다. 사용자가 리뷰를 등록하면 다시 지도 화면으로 돌아온다. 지도 화면에서 장소를 선택하고 리뷰 보기 버튼을 누른다면 해당 장소에 등록된 리뷰 목록들을 화면에 출력한다. 메인화면에서 사용자가 프로필 화면으로 넘어가면 자신의 활동기록을 볼 수 있다. 나의 활동 기록 버튼을 누르면 나의 리뷰와 나의 즐겨찾기 버튼이 출력된다. 그 중 나의 리뷰 버튼을 누르면 사용자가 작성한 리뷰 목록들이 화면에 나타나고, 나의 즐겨찾기 버튼을 누르면 사용자가 등록한 즐겨찾기한 장소들이 화면에 나타난다.

5. Implementation requirements

본 애플리케이션은 Android OS 기반으로 개발되었으며, Android 8.0 (SDK 26) 이상 환경에서 시스템 구동이 가능하다. 실제 테스트는 Pixel 3a API 35 가상 디바이스에서 수행되었다.

6. Glossary

용어	설명
MVVM	사용자 인터페이스, 비즈니스 로직, 데이터 처리를 명확히 분리하는 아키텍처 패턴
LiveData	ViewModel에서 관리되는 관찰 가능한 데이터 객체. UI가 데이터 변화에 자동으로 반응할 수 있도록 도와준다.

7. References

1. LiveData

https://developer.android.com/topic/libraries/architecture/livedata?hl=ko

2. ViewModel

https://developer.android.com/topic/libraries/architecture/viewmodel?hl=ko