최종 문건

개념계획서

기초설계

화면설계서, 기능 다이어그램

상세설계

알고리즘, 슈도코드, 레이아웃

3조

김문희 김소연 김소지 문주영

3주차 개념계획서

개념계획서

캡스톤 개념계획서
[AR관련 개발]

1. 목표 및 필요성

1-1. 과제개요

과제명		증강현실을 이용한 위치기반 사진일기 어플리케이션 개발				
키워드	국문	증강현실, GPS				
	영문	AR(augmented reality), GPS				
과제 규모	연구비		연구기간	15 주	소요인력	4 명

1-2. 기술개발 필요성

- 기존 일기장 앱은 시간에 집중하여 ‘언제’ 어떤 일이 있었는지를 떠올리게 함.
- 또한 특정 공간에서 일어난 일을 떠올리게 하는 기능은 제공하지 않음.
-따라서 공간에 집중한 일기장 서비스를 제공하고자 함.
-특정 위치에 일기를 저장하여 해당 위치에 있을 때만 일기를 볼 수 있게 함으로써 ‘어디에서’ 어떤 일이 있었는지를 떠올리게 함.

1-3. 기술개발 목표와 내용

•목표

- AR과 일기장 연동

•일기에 저장된 위치 정보를 받아와 해당 위치에서 그 곳에 저장된 일기를 불러옴.

•내용

구분	내용
상세내역	o 메인 화면에서 주변 위치를 카메라로 비추면 AR로 그 위치에 저장된 일기가 들어있는 캡슐이 보임. o 캡슐을 누르면 저장된 일기 화면으로 전환됨. o 플로팅 버튼을 구현'+'버튼을 누르면 카메라 버튼과 갤러리 버튼을 띄움. 전자를 누를 경우 카메라로, 후자를 누를 경우 갤러리 화면으로 전환됨. o 카메라에서 사진을 찍거나 갤러리에서 사진을 선택한 이후 일기작성 화면으로 넘어가고 일기 작성이 완료되면 DB에 저장됨. o DB에는 위도, 경도, 작성일, 내용, 사진을 저장함. o 위치는 실시간으로 받아오며 일기를 저장한 위치에 접근하면 Push 알람이 발생. o 캡슐이 보이는 반경을 설정할 수 있음. (50m, 100m, 200m, 300m)
핵심기술 및 서비스	o 핵심 기술 - 증강현실 어플리케이션 개발 - GPS를 이용한 위치 인식 및 위치 정보 획득 o 적용범위(또는 서비스) - 일상 속 언제든 어디서나 추억을 남기고 싶은 사용자들

2. 기술 특성

2-1. 혁신성 및 차별성

-기존에 없는 AR 일기장 서비스 제공

-> 최초의 AR 일기장 서비스라는 점에서 의의가 있음.

-공간에 집중한 일기장 서비스 제공

-> 시간이 아닌 공간에 집중한 일기장 서비스를 제공함으로써 특정 위치에 있을 때 즉각적으로 기억을 떠올리게 할 수 있는 기능을 함.

3. 개발 전략

3-1. 기술개발 방법

개발 방법

- AR Core를 이용한 안드로이드 스튜디오 개발 환경 구축

-> 안드로이드 스튜디오 버전 : 3.1.1

- Unity를 이용한 캡슐모형 3D 모델링

-> Unity 버전 : 2018.1.0f2

- SQLite를 이용한 DB 구축

4. 기대 효과

4-1. 파급 효과

사회적 측면

- 바쁜 일상을 살아가는 현대인들이 잠시나마 현실 속에서 추억을 떠올릴 수 있는 효과를 기대.

5. 제안기관 정보

5-1 연구시설 및 연구 장비 보유 현황

연구시설·장비명	수량	활용도
개인 노트북	4	연구개발
갤럭시 S8+/갤럭시S9	1	연구개발

5-2 본 과제 참여연구원 현황

박사	석사	학사	기타	합계
0	0	4	0	4

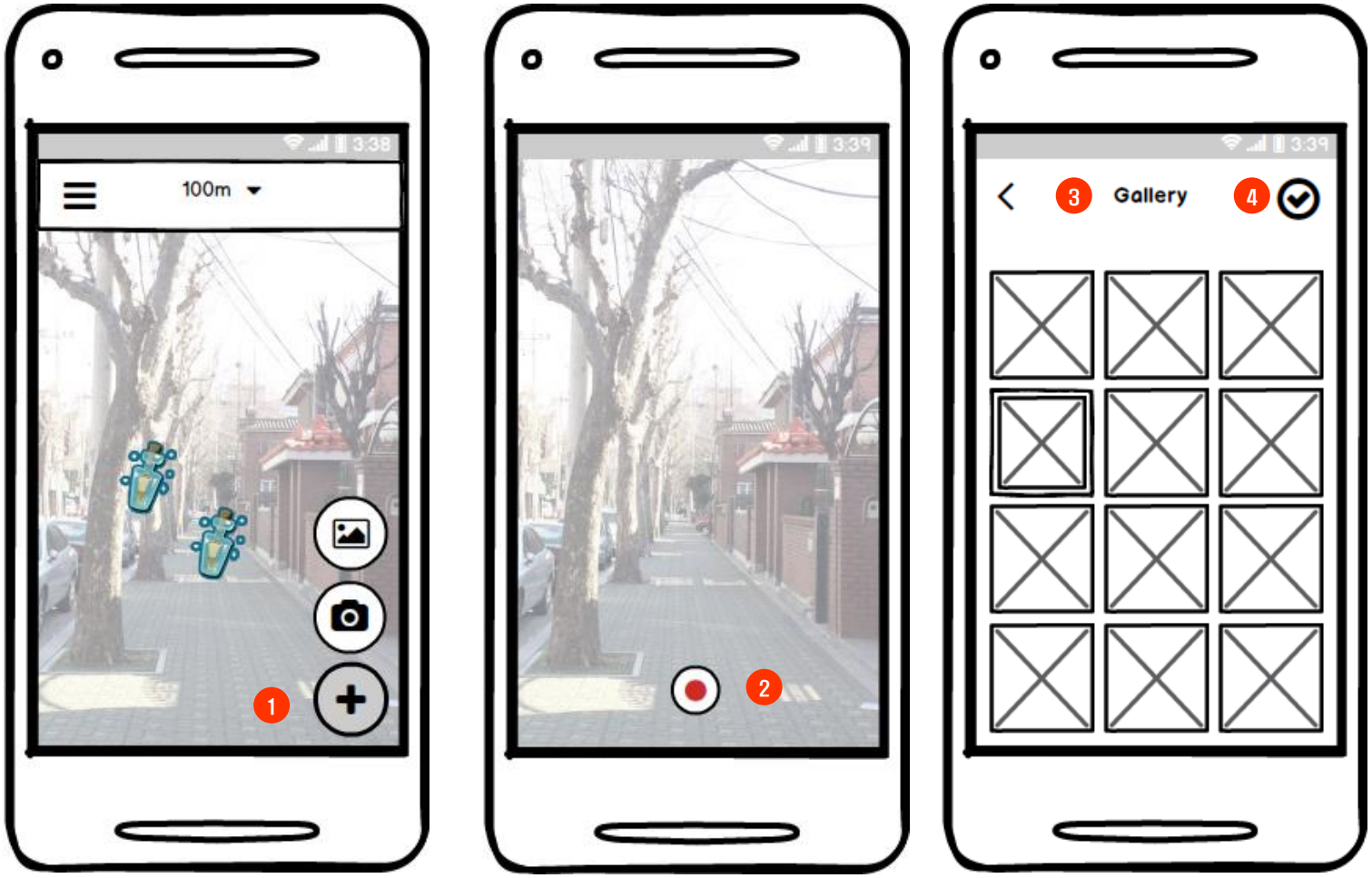
4주차 기초설계

화면설계서 및 다이어그램

화면명	메인 화면	최종 수정일		이전 수정일		6 Page
Location				page		Project 프로젝트명

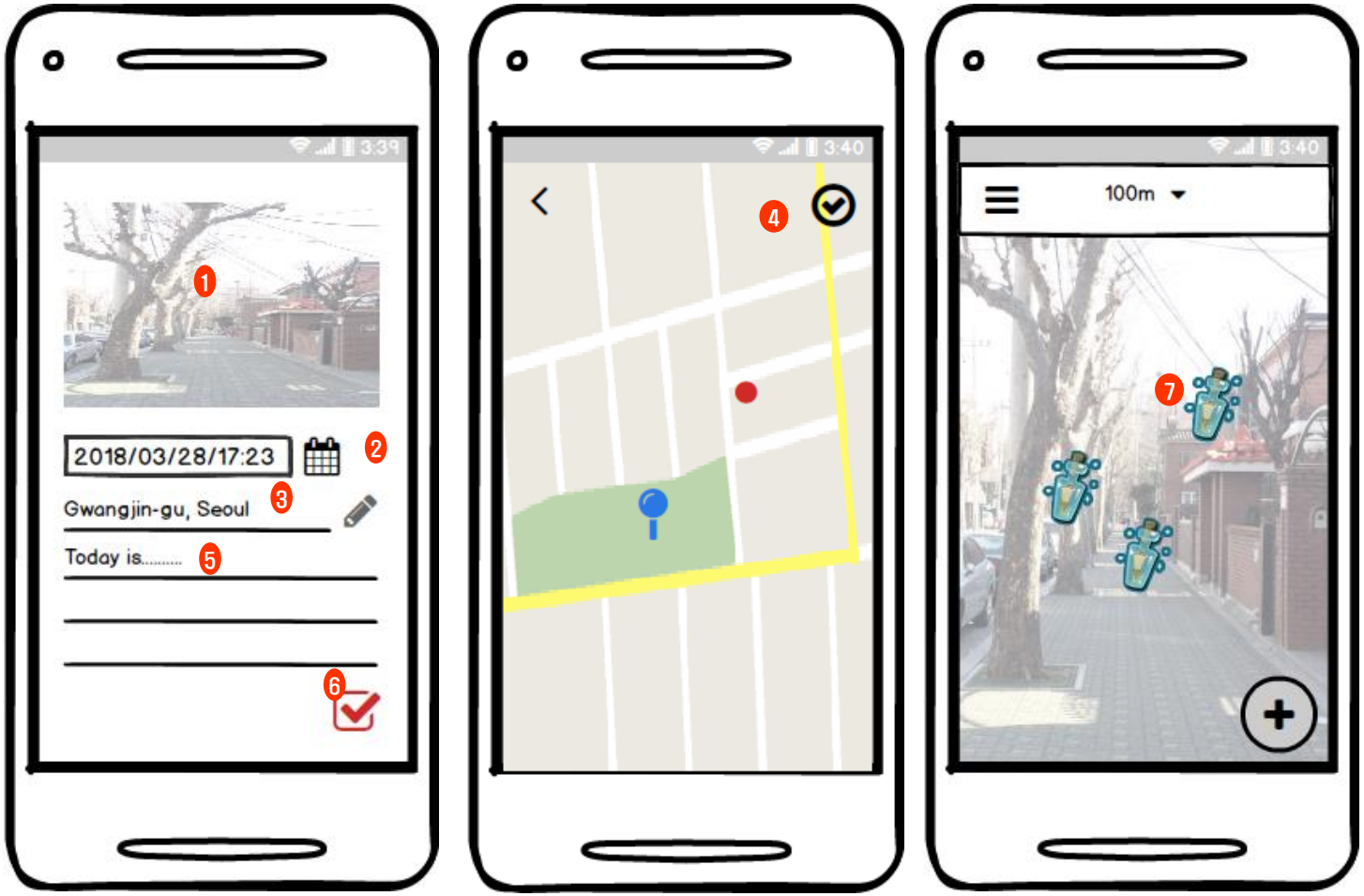
Description (화면설명)	
1	일기장 새로 만들기 버튼
2	저장된 일기장 클릭 시 저장된 일기장을 볼 수 있음.
3	아이콘을 인식할 수 있는 범위를 지정할 수 있음
4	장소를 저장한 날짜의 목록을 보여줌.
5	기억이 저장된 장소 근처에 있을 시 푸시 알람 발생
6	
7	
8	
9	
10	
11	
12	
13	
14	

화면명	1번 화면	최종 수정일		이전 수정일		7 Page
Location				page		Project 프로젝트명



Description (화면설명)	
1	플로팅 버튼 구현
2	카메라 버튼을 눌렀을 시의 화면.
3	갤러리 버튼을 눌렀을 시의 화면
4	사진 선택 후 일기를 쓰는 화면으로 전환
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

화면명	1번 화면	최종 수정일		이전 수정일		8 Page
Location				page		Project 프로젝트명



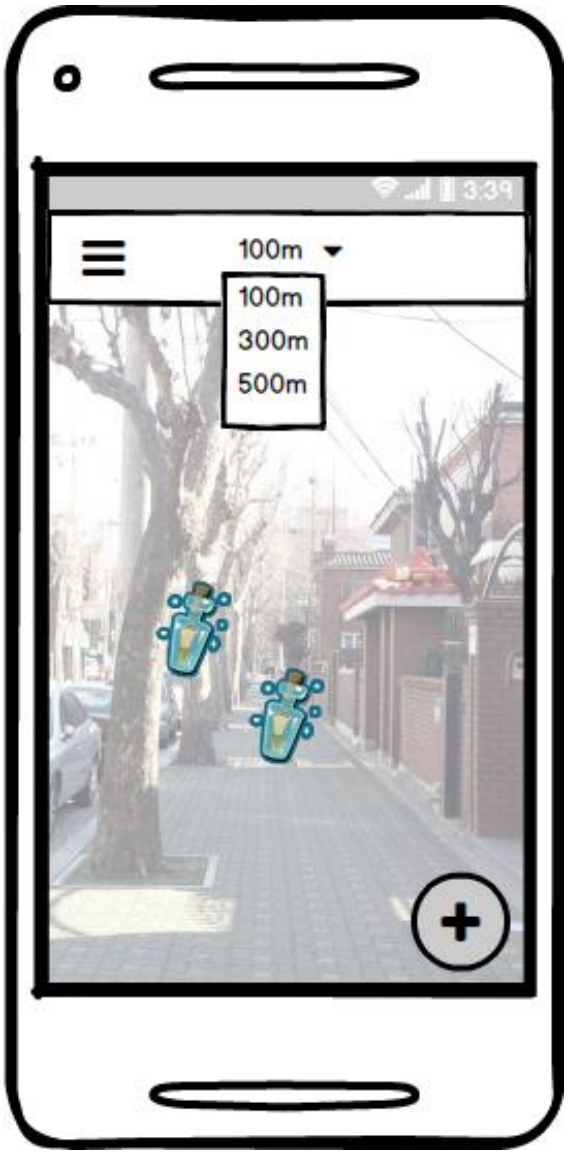
Description (화면설명)	
1	선택한 사진을 불러옴.
2	저장하고 싶은 날짜로 설정
3	저장하고 싶은 위치 설정
4	위치 설정 화면 체크 표시를 누르면 일기 작성 화면으로 넘어감
5	일기 작성
6	일기 저장
7	일기 저장 후 지정한 위치에 아이콘이 생성됨
8	
9	
10	
11	
12	
13	
14	

화면명	2번 화면	최종 수정일		이전 수정일		9 Page
Location				page		Project 프로젝트명

Description (화면설명)

1	저장된 아이콘을 클릭하여 일기 확인
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

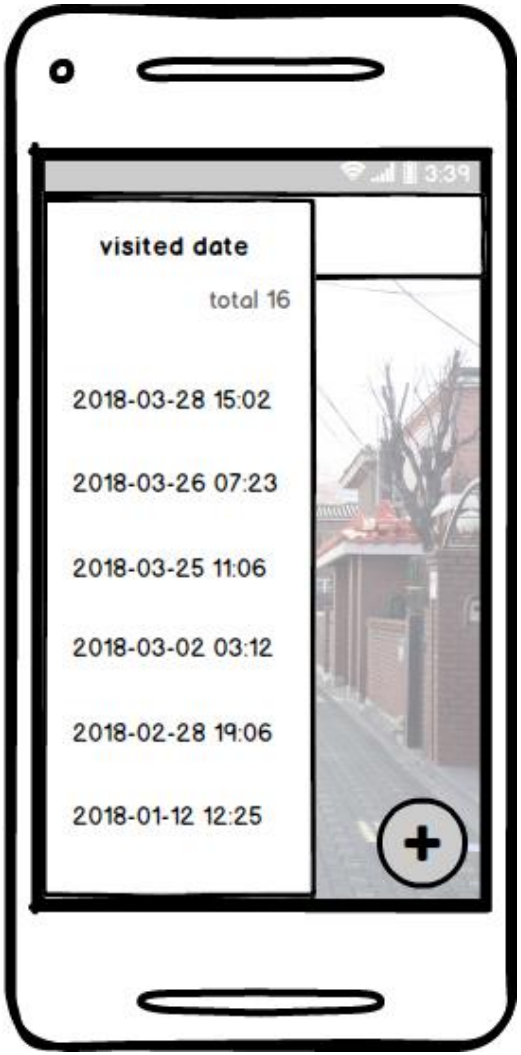
화면명	3번 화면	최종 수정일		이전 수정일		10 Page
Location				page		Project 프로젝트명



Description (화면설명)

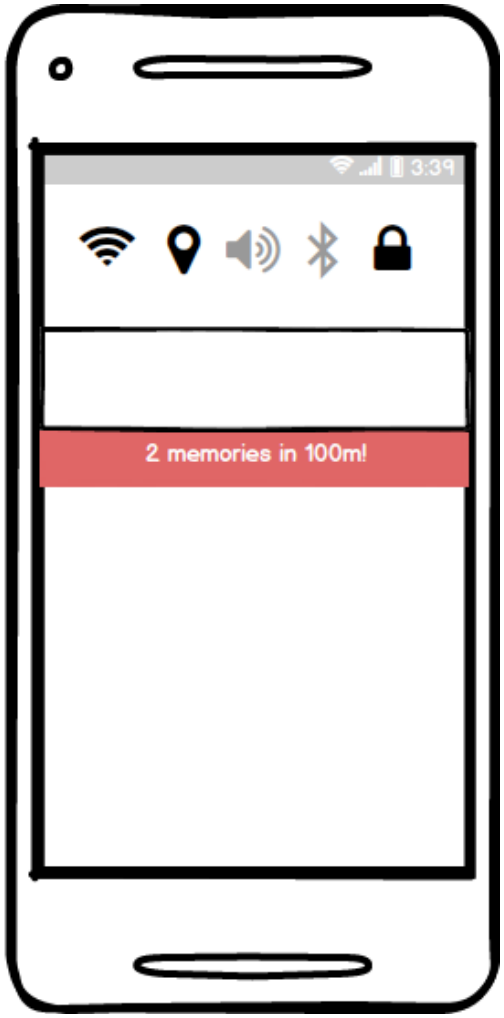
1	해당 위치에서 아이콘을 볼 수 있는 범위를 설정할 수 있음
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

화면명	4번 화면	최종 수정일		이전 수정일		11 Page
Location				page		Project: 프로젝트명



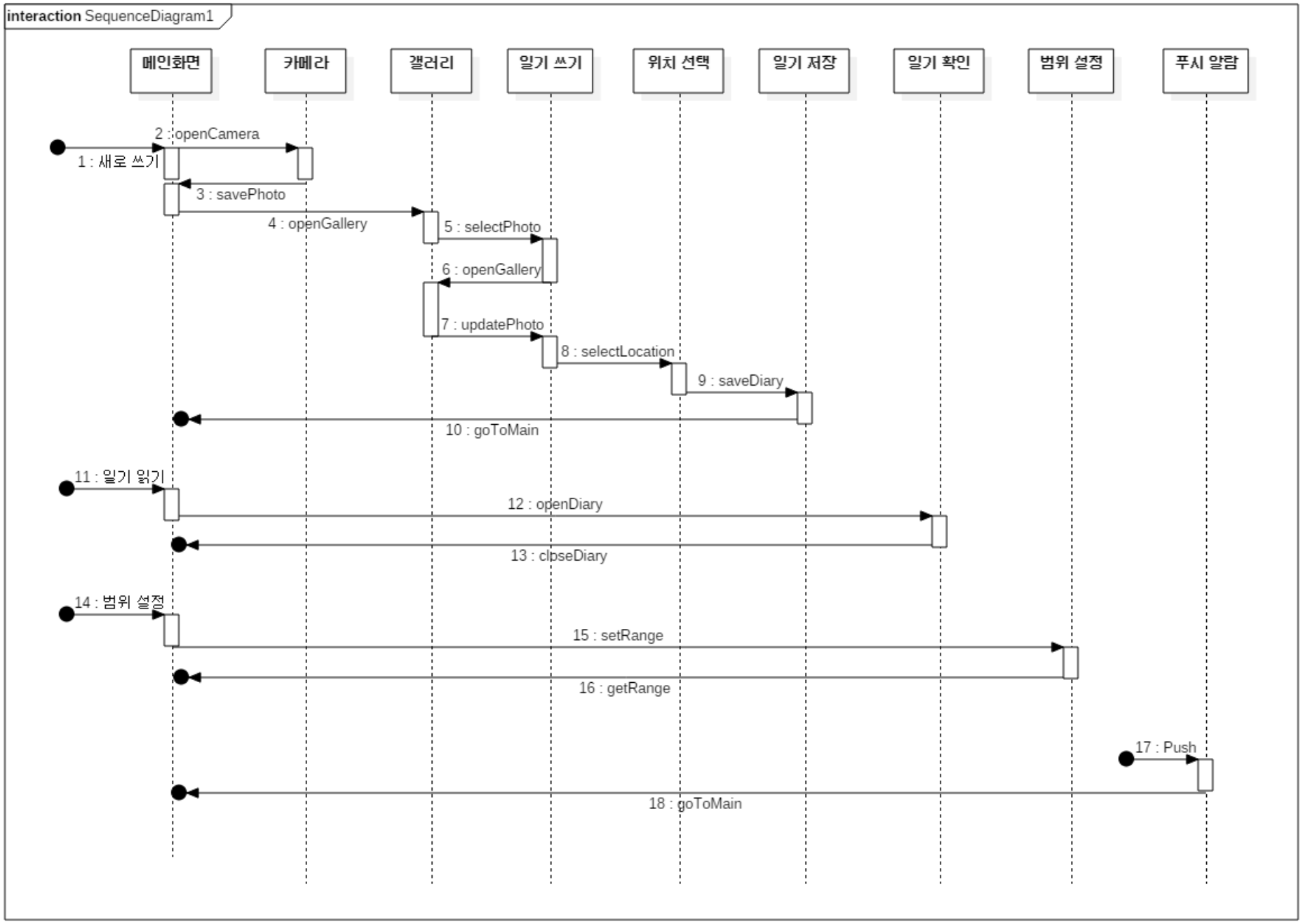
Description (화면설명)	
1	저장한 날짜를 확인할 수 있음. 확인만 할 수 있으며 저장된 일기의 개수를 확인할 수 있음.
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

화면명	5번 화면	최종 수정일		이전 수정일		12 Page
Location				page		Project 프로젝트명



Description (화면설명)

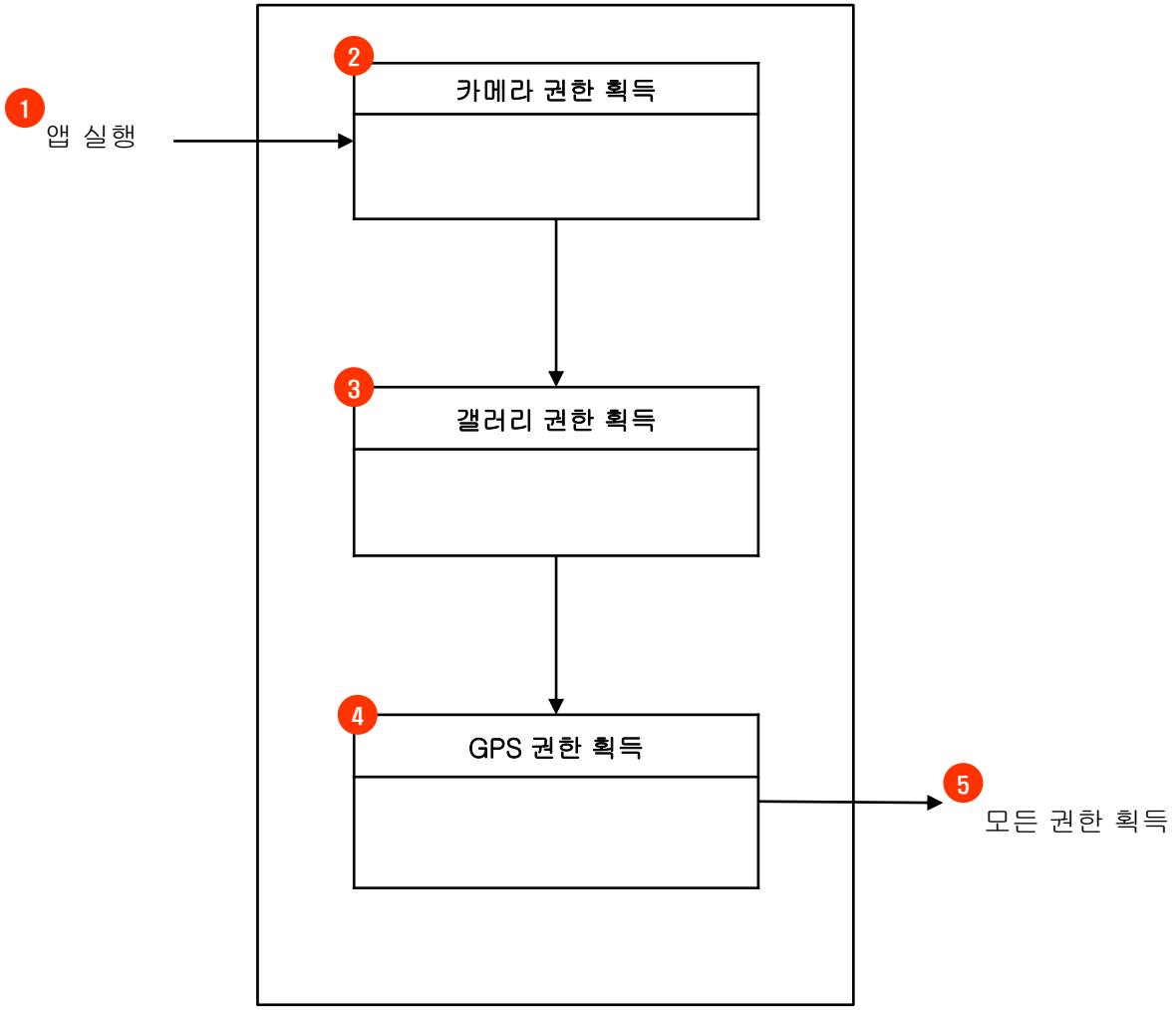
1	푸시 알람이 발생하였을 때의 화면
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	


interaction SequenceDiagram1
메인화면
카메라
갤러리
일기 쓰기
위치 선택
일기 저장
일기 확인
범위 설정
푸시 알람
1 : 새로 쓰기
2 : openCamera
3 : savePhoto
4 : openGallery
5 : selectPhoto
6 : openGallery
7 : updatePhoto
8 : selectLocation
9 : saveDiary
10 : goToMain
11 : 일기 읽기
12 : openDiary
13 : closeDiary
14 : 범위 설정
15 : setRange
16 : getRange
17 : Push
18 : goToMain

5주차 기초설계

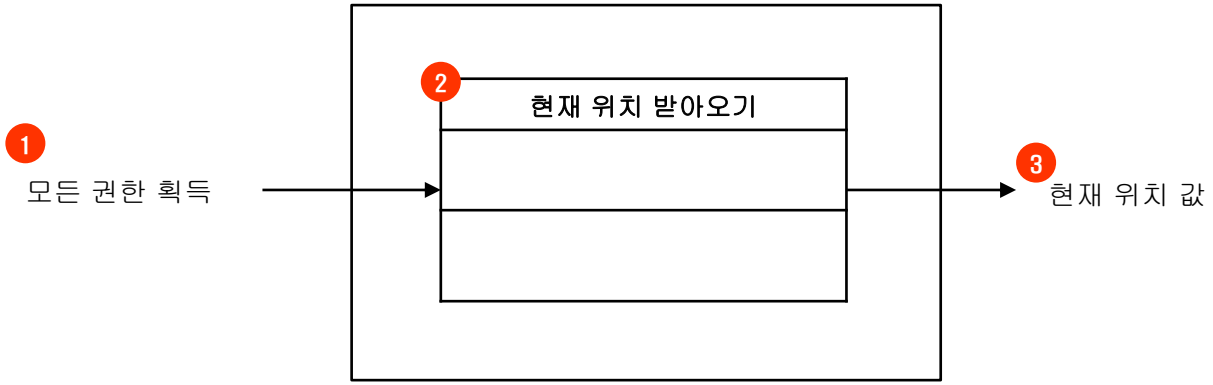
기능 다이어그램

화면명	권한 받아오기	최종 수정일	18-04-03	이전 수정일		15 Page
Location				page		Project 프로젝트명



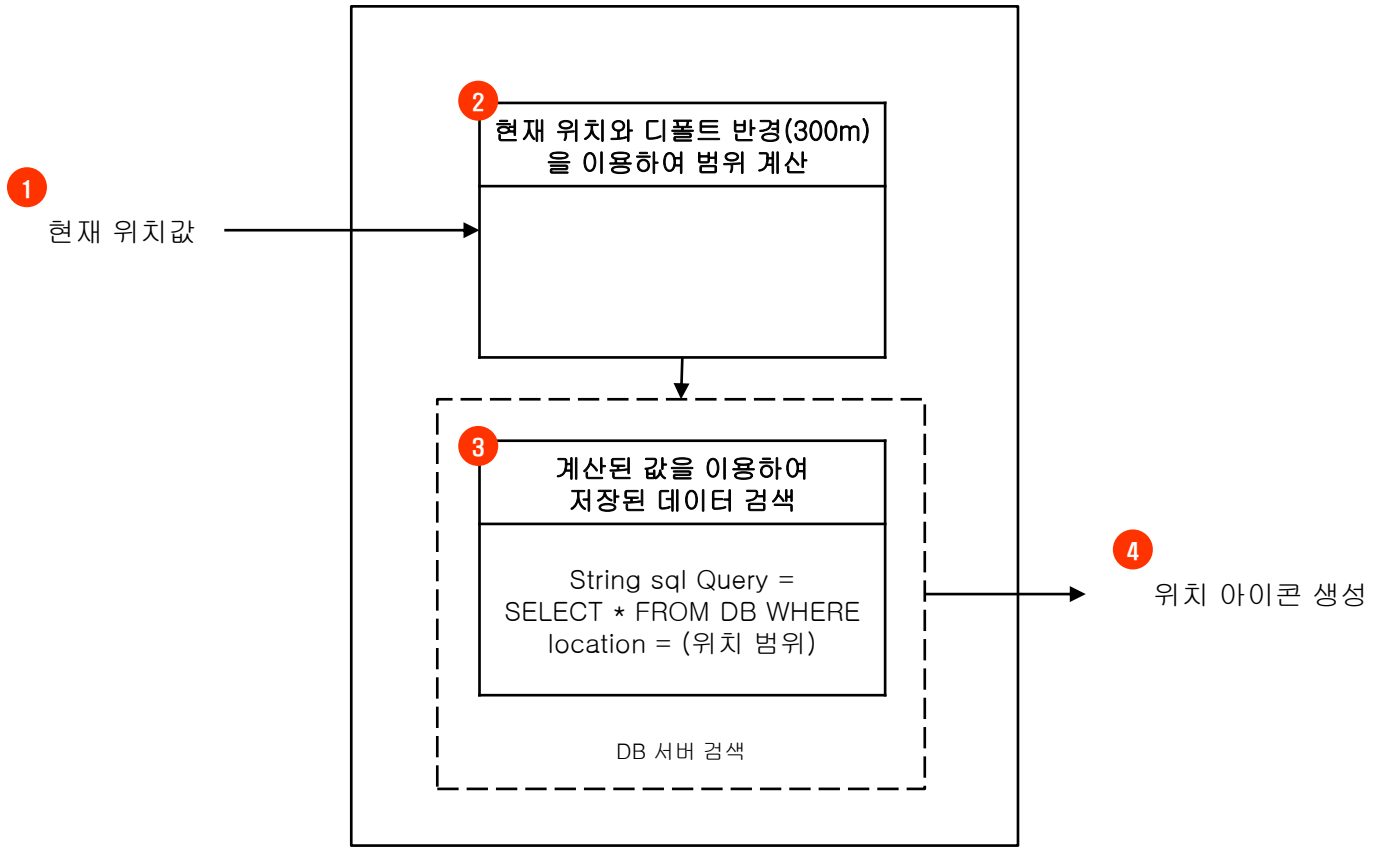
Description (화면설명)	
1	앱을 실행한다
2	카메라 권한이 있는지 확인한다
3	갤러리 권한이 있는지 확인한다
4	GPS 권한이 있는지 확인한다
5	모든 권한을 획득한다
6	
7	
8	
9	
10	
11	
12	
13	
14	

화면명	현재 위치 받아오기	최종 수정일	18-04-03	이전 수정일		16 Page
Location				page		Project: 프로젝트명



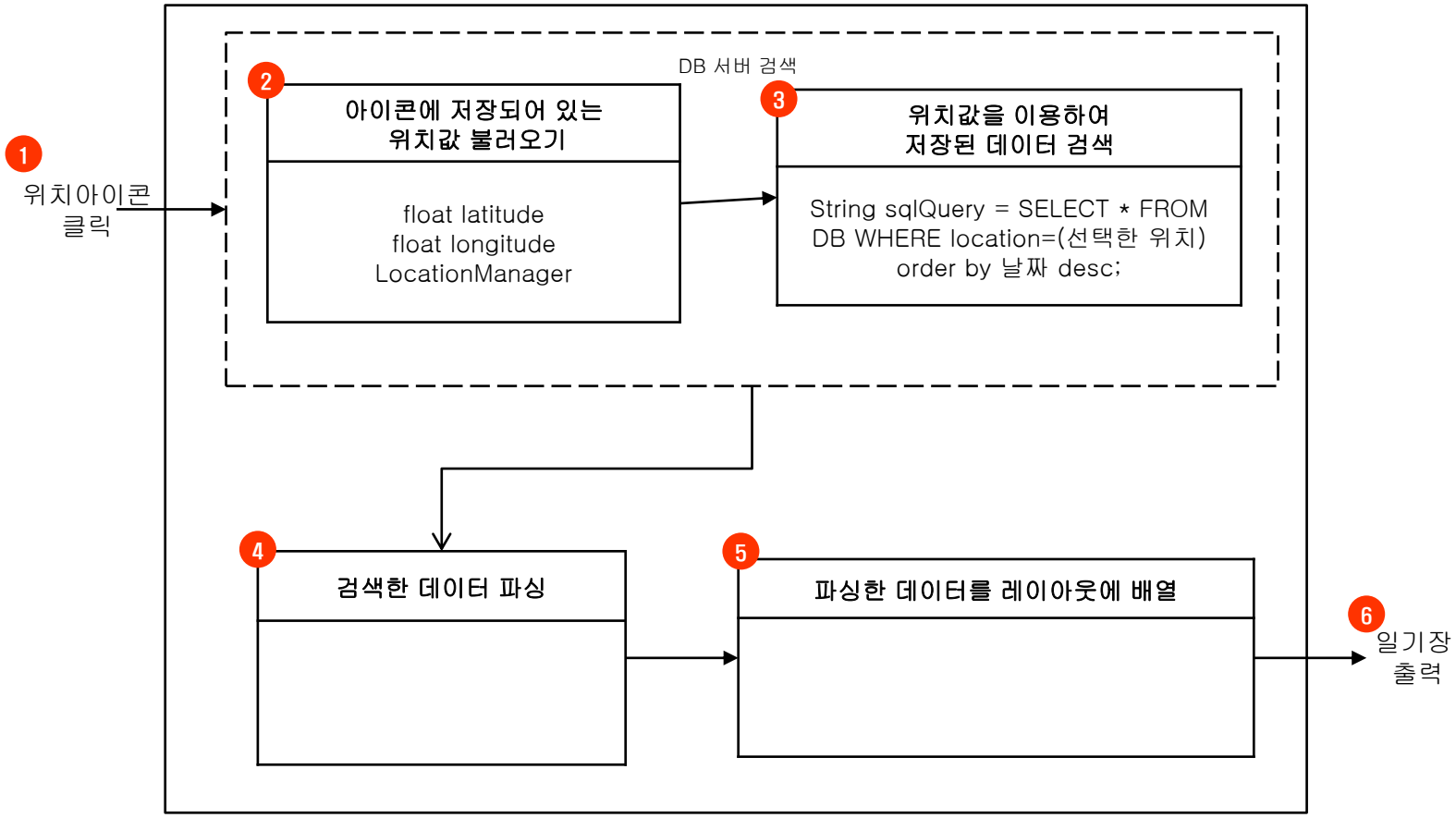
Description (화면설명)	
1	모든 권한을 획득하면, 현재 위치를 받아오기 기능을 실행한다.
2	현재 위치를 받아온다.
3	현재 위치값을 출력한다.
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

화면명	근처 저장된 위치 받아오기	최종 수정일	18-04-03	이전 수정일		17 Page
Location				page		Project 프로젝트명



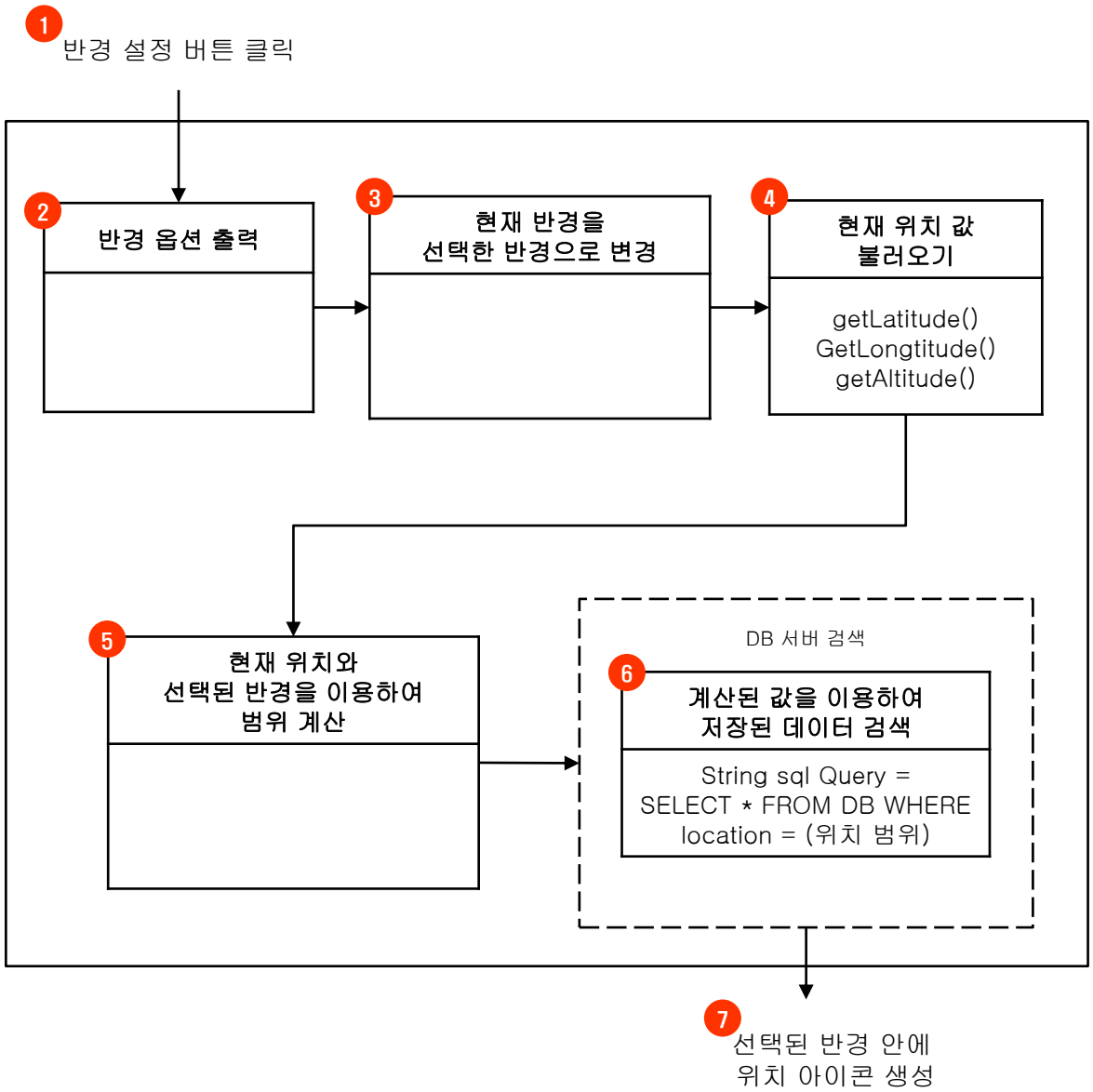
Description (화면설명)	
1	현재 위치값을 받아온다
2	현재 위치와 디폴트 반경(300m)를 이용하여 범위 계산.
3	계산된 값을 이용하여 저장된 데이터 검색
4	데이터가 있으면 위치 아이콘을 생성.
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

화면명	일기장 출력하기	최종 수정일	18-04-03	이전 수정일		18 Page
Location				page		Project 프로젝트명



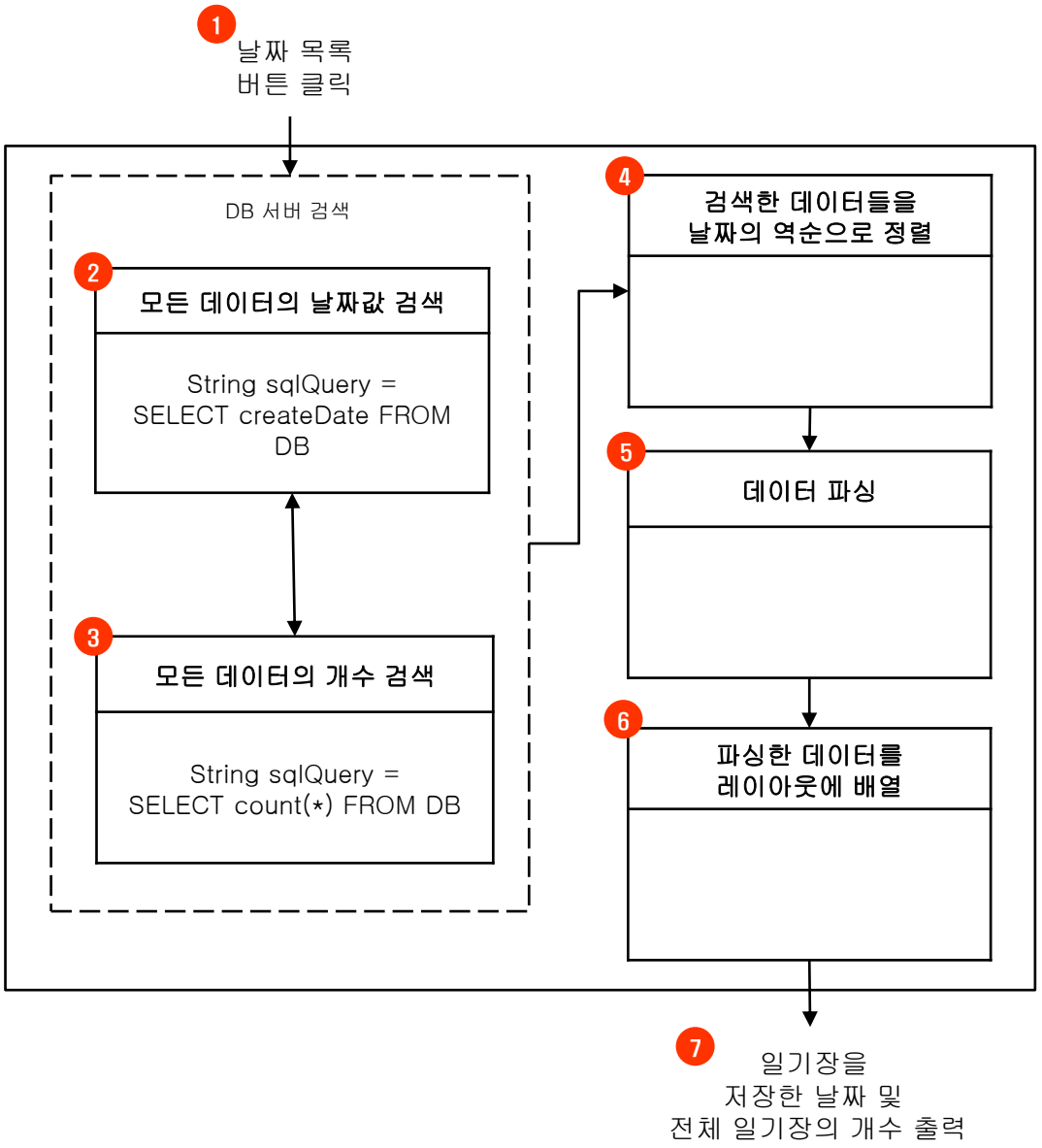
Description (화면설명)	
1	위치 아이콘 클릭
2	DB 서버에서 아이콘에 저장되어 있는 위치값(위도, 경도)을 불러옴.
3	DB 서버에서 위치값을 이용하여 저장된 데이터를 검색함
4	검색한 데이터를 파싱
5	파싱한 데이터를 레이아웃에 배열
6	일기장 출력
7	
8	
9	
10	
11	
12	

화면명	위치 범위 정하기	최종 수정일	18-04-03	이전 수정일		19 Page
Location				page		Project: 프로젝트명



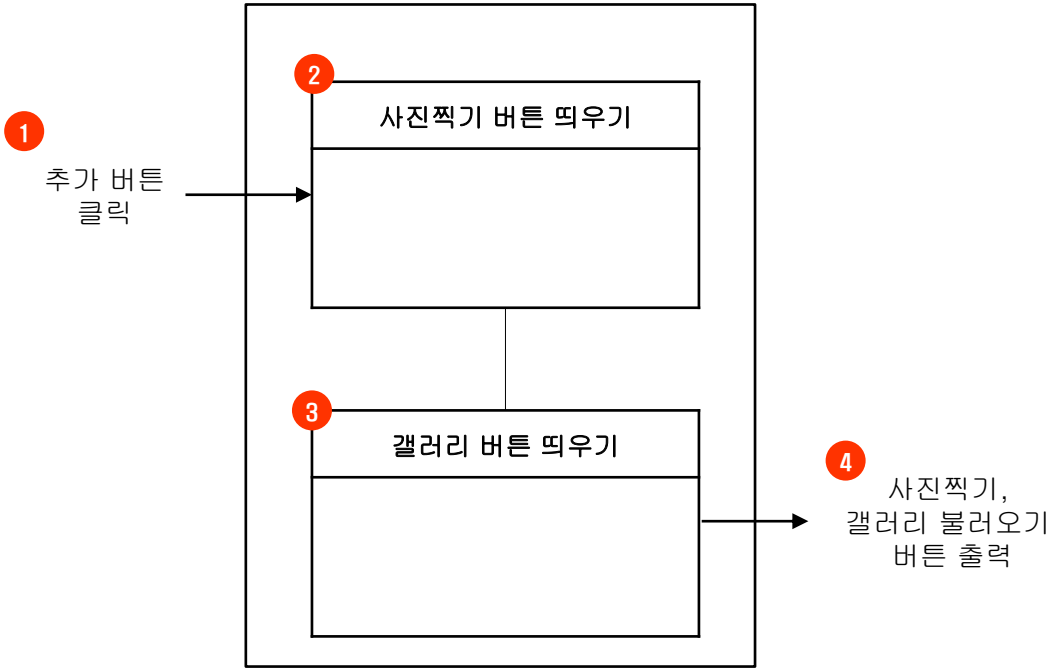
Description (화면설명)	
1	반경 설정 버튼 클릭
2	반경 설정 클릭 시 50m, 100m, 200m, 300m의 옵션창이 반경 텍스트 하단에 출력됨
3	현재 반경을 선택한 반경으로 변경함
4	현재 위치 값을 불러옴
5	현재 위치와 선택된 반경을 이용하여 위치 값 범위 계산
6	DB 서버에서 계산된 값을 이용하여 저장된 데이터 검색
7	계산된 위치에 해당하는 데이터 값이 있다면 선택된 반경 안에 위치 아이콘 생성
8	
9	
10	

화면명	저장한 날짜의 목록 보기	최종 수정일	18-04-03	이전 수정일		20 Page
Location				page		Project 프로젝트명



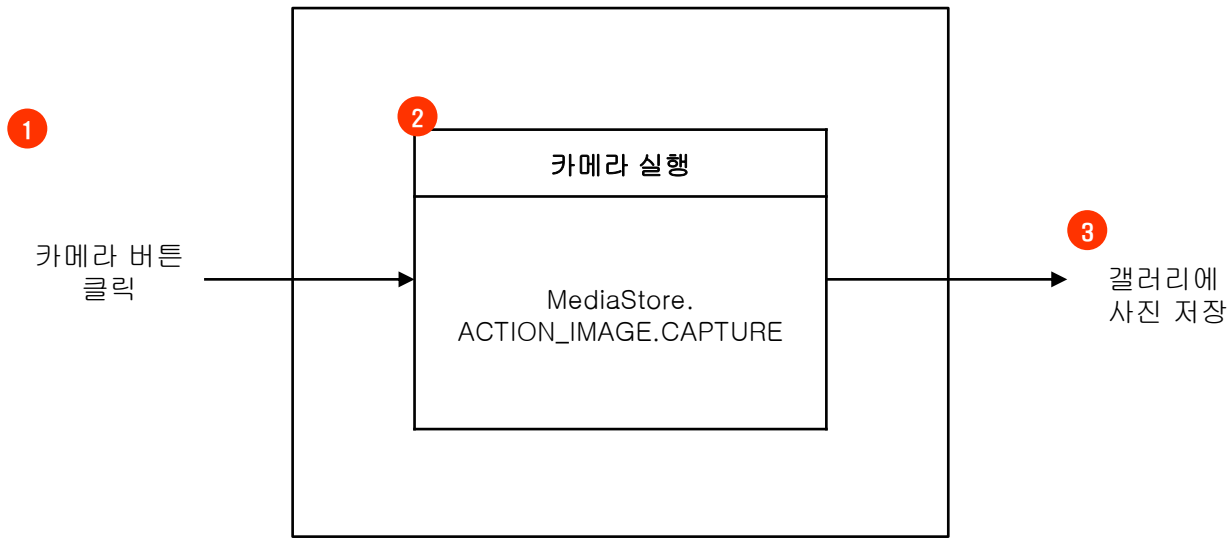
Description (화면설명)	
1	날짜 목록 버튼을 클릭
2	DB 서버에 저장된 모든 데이터의 날짜값을 검색.
3	DB 서버에 저장된 모든 데이터의 개수 검색
4	검색한 데이터들을 UI상에서 최신순으로 보이게 하기 위하여 날짜의 역순으로 정렬
5	검색한 데이터를 파싱
6	파싱한 데이터를 레이아웃에 배열
7	일기장을 저장한 날자 및 전체 일기장의 개수 출력
8	
9	
10	
11	
12	

화면명	플로팅 버튼 띄우기	최종 수정일	18-04-03	이전 수정일		21 Page
Location				page		Project 프로젝트명



Description (화면설명)	
1	추가 버튼 클릭
2	사진찍기 버튼이 출력됨.
3	갤러리 버튼이 출력됨.
4	사진찍기, 갤러리 불러오기 버튼 출력됨
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

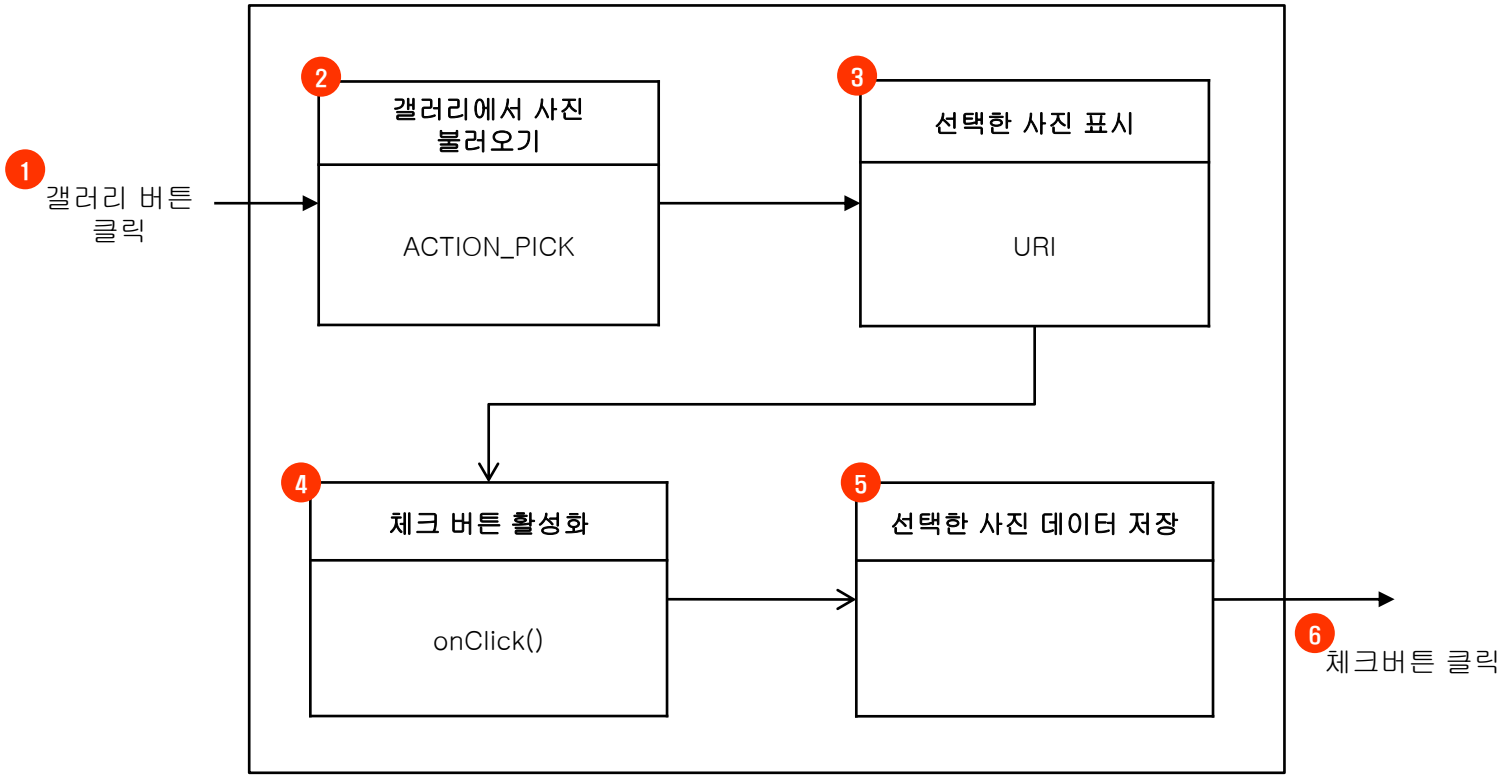
화면명	사진 찍기	최종 수정일	18-04-03	이전 수정일		22 Page
Location				page		Project 프로젝트명



Description (화면설명)	
1	메인화면의 카메라 버튼 클릭
2	내장된 기본 카메라앱 실행
3	갤러리에 사용자가 찍은 사진 저장
4	확인버튼 클릭
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

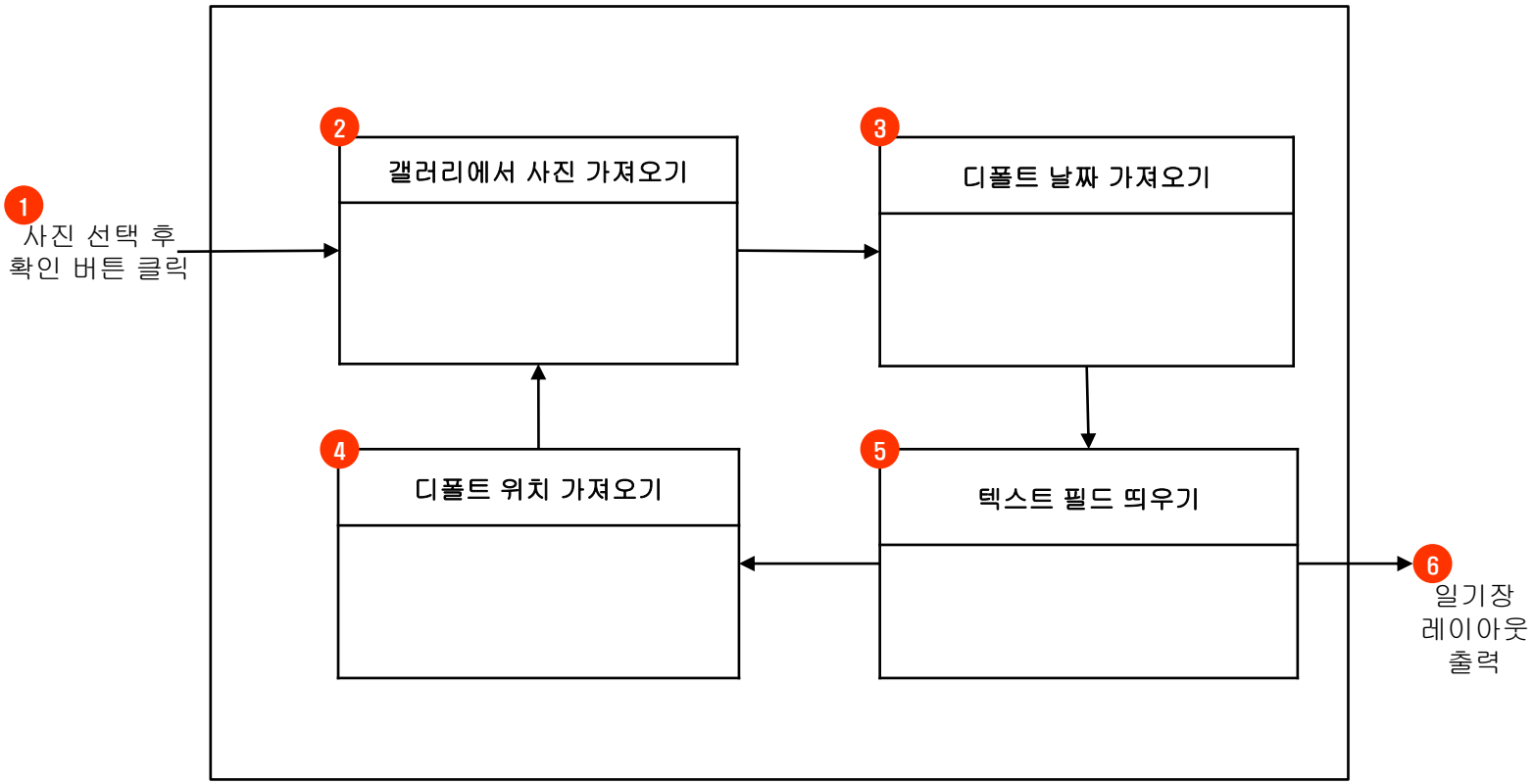
화면명	사진 선택하기	최종 수정일	18-04-03	이전 수정일	
Location				page	

Project	프로젝트명



Description (화면설명)	
1	메인화면의 갤러리 버튼 클릭
2	내장되어 있는 기본 갤러리 호출하여 화면에 띄우기
3	사용자가 선택한 한 장의 사진 표시하기
4	사용자가 사진을 선택하였다면 체크버튼 활성화시키기
5	선택한 사진에 대한 데이터를 저장
6	활성화된 체크버튼 클릭
7	
8	
9	
10	
11	
12	
13	

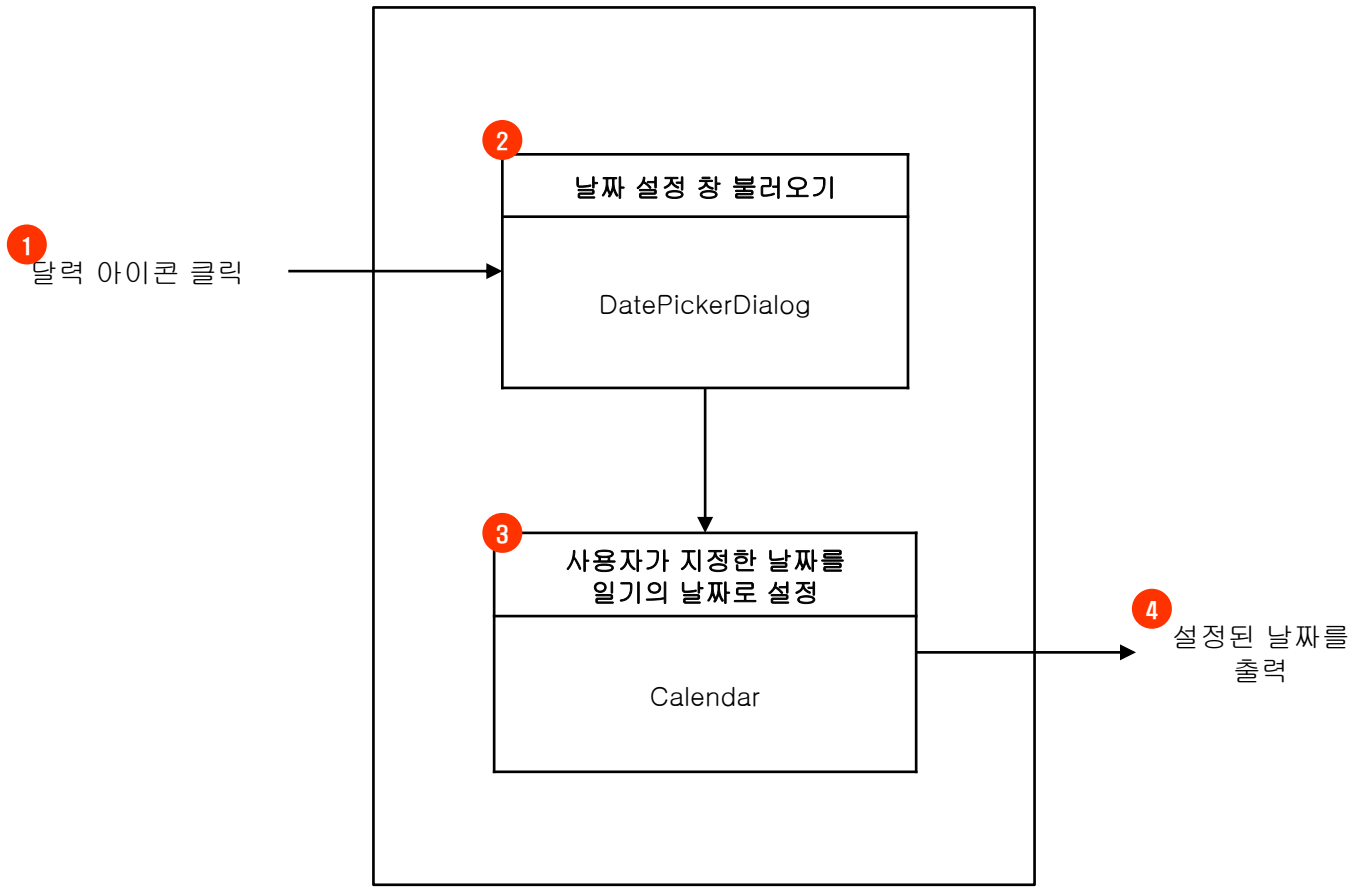
화면명	일기장 레이아웃 출력	최종 수정일	18-04-03	이전 수정일		24 Page
Location				page		Project 프로젝트명



Description (화면설명)

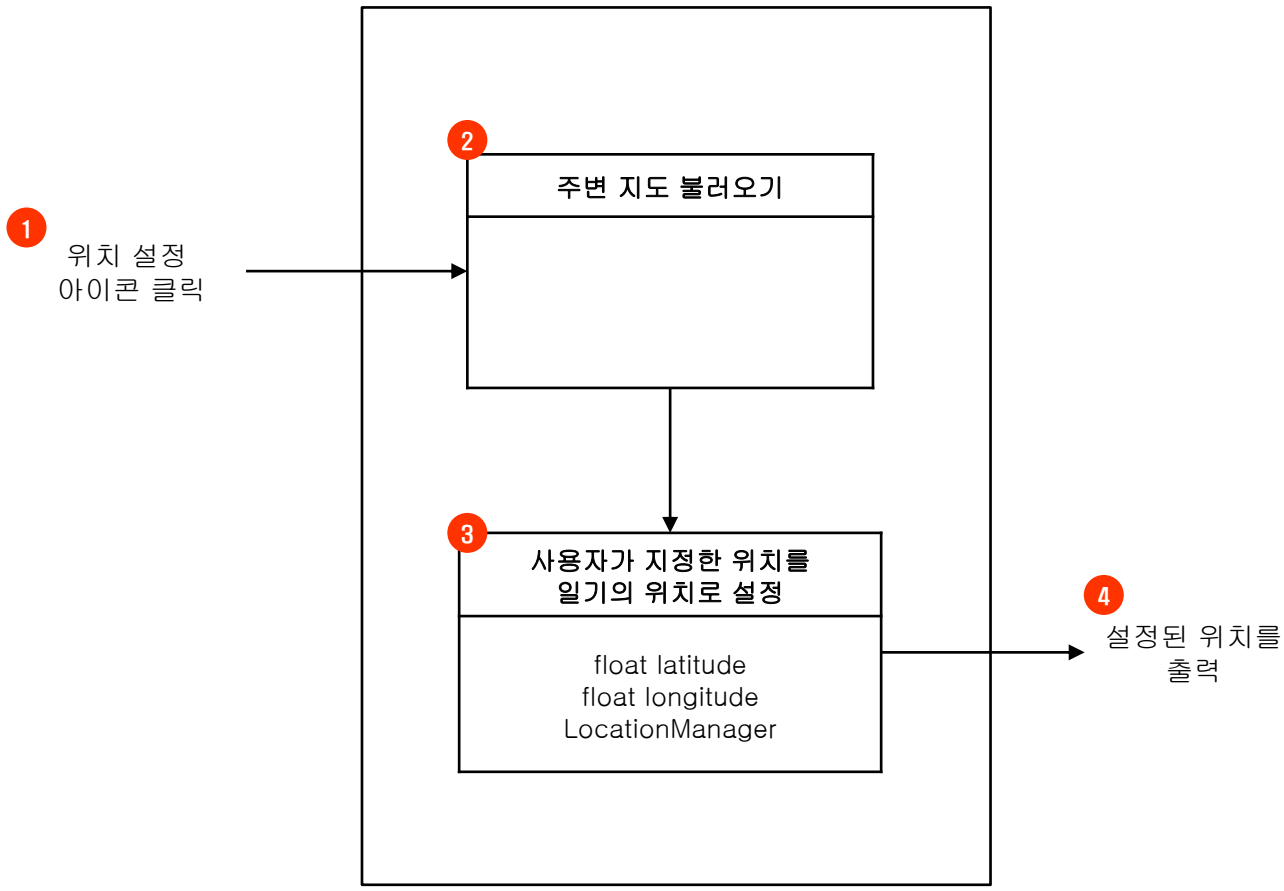
1	카메라>확인 또는 갤러리>확인 버튼을 클릭한다
2	갤러리에서 선택하거나 새로 찍은 사진 가져오기
3	디폴트 날짜값 가져오기
4	디폴트 위치값 가져오기
5	텍스트필드 띄우기
6	일기장 레이아웃 출력
7	
8	
9	
10	
11	
12	
13	
14	

화면명	날짜 설정하기	최종 수정일	18-04-03	이전 수정일		25 Page
Location				page		Project 프로젝트명



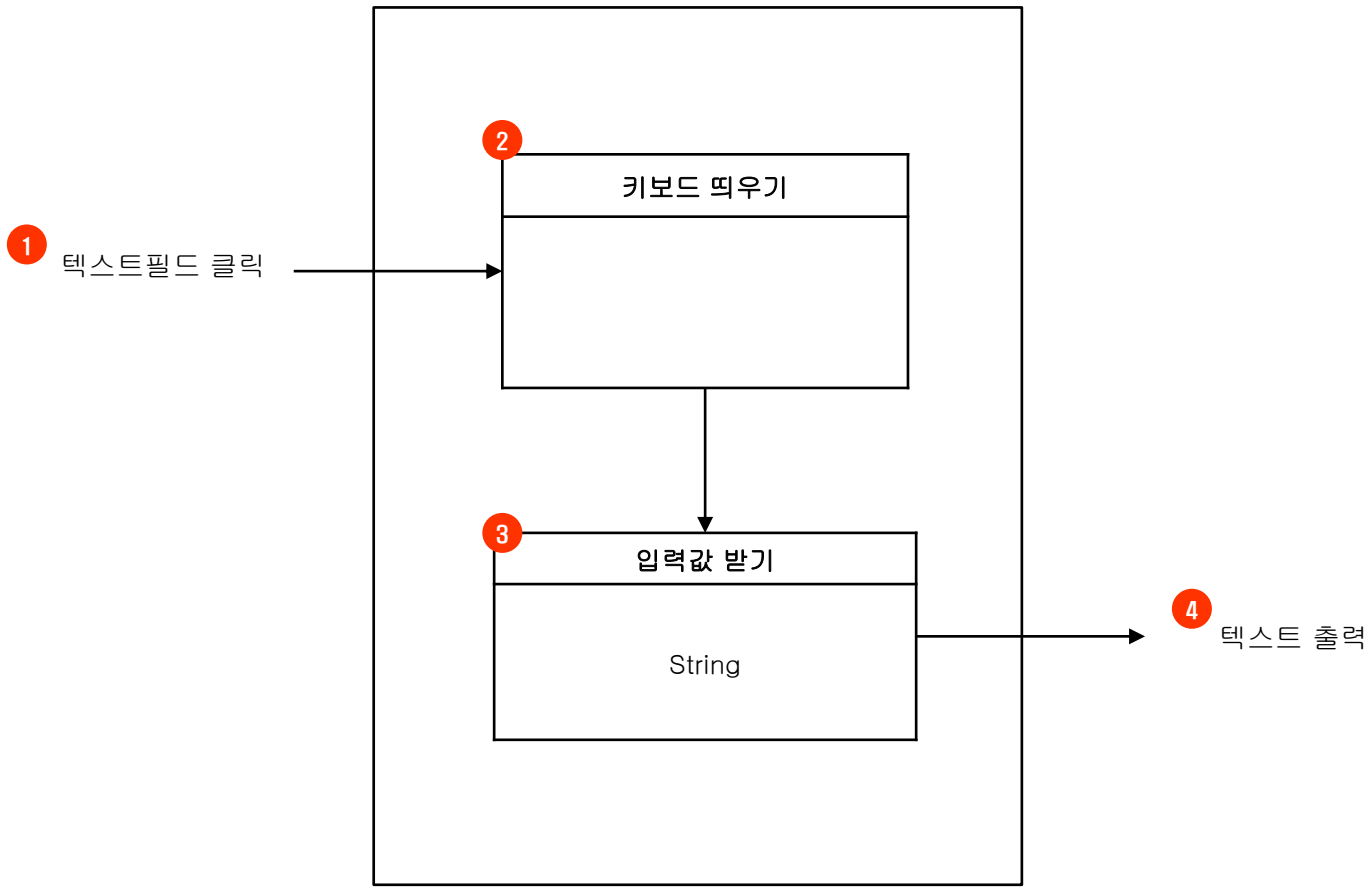
Description (화면설명)	
1	사용자가 날짜를 바꾸기 위해 달력 아이콘을 클릭한다
2	날짜를 설정할 수 있는 창을 불러온다
3	일기의 날짜를 사용자가 바꾼 날짜로 바꾼다
4	날짜 레이아웃에 사용자가 설정한 날짜로 바꿔서 출력한다
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

화면명	위치 설정하기	최종 수정일	18-04-03	이전 수정일		26 Page
Location				page		Project 프로젝트명



Description (화면설명)	
1	사용자가 위치를 바꾸기 위해 위치 정하기 아이콘을 클릭한다
2	위치를 설정할 수 있는 2D 지도를 불러온다
3	일기의 위치를 사용자가 바꾼 위치로 바꾼다
4	위치 레이아웃에 사용자가 설정한 위치로 바꿔서 출력한다
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

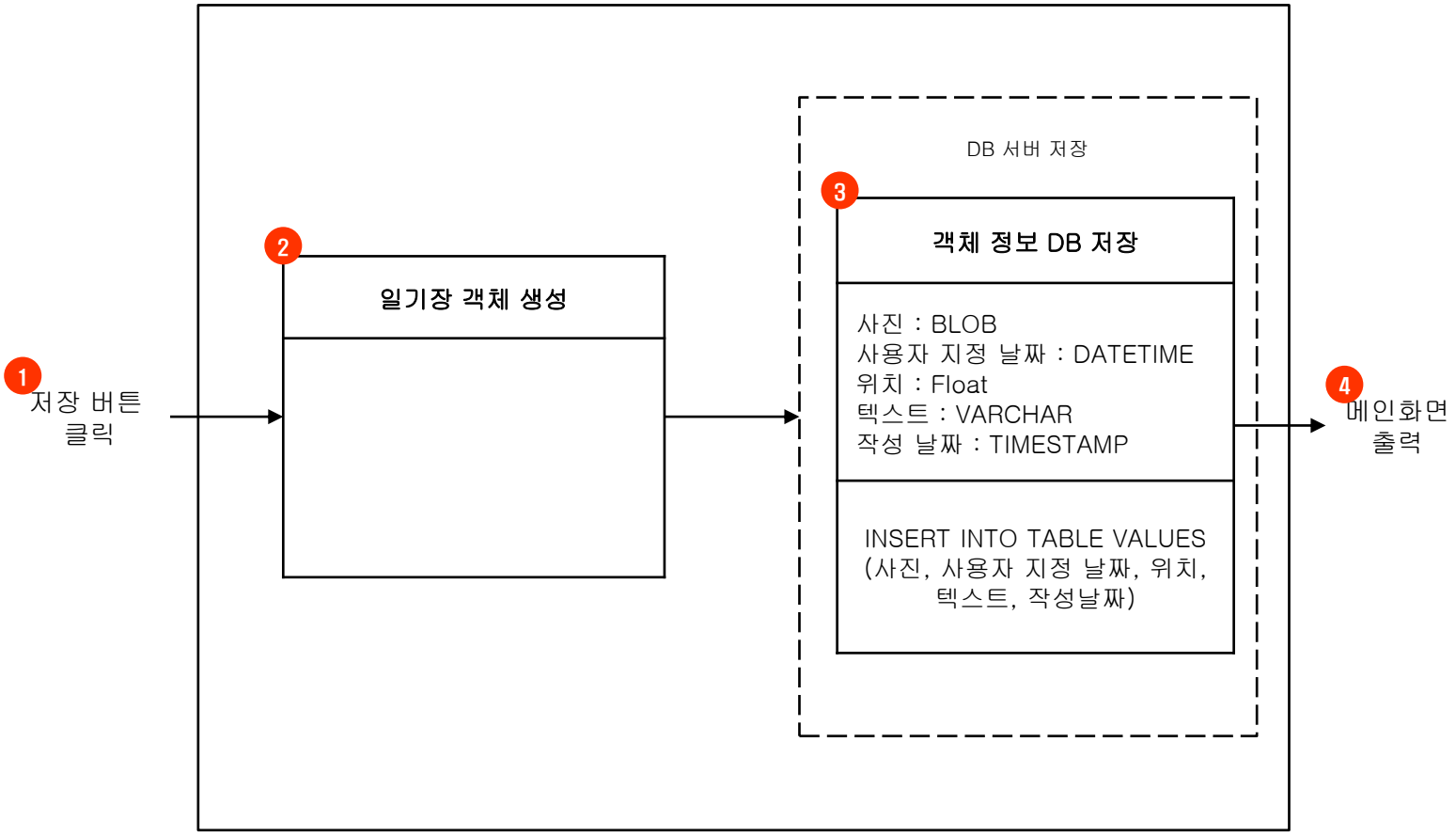
화면명	텍스트 작성하기	최종 수정일	18-04-03	이전 수정일		27 Page
Location				page		Project 프로젝트명



Description (화면설명)	
1	사용자가 일기를 쓰기 위해 텍스트필드를 클릭한다
2	키보드를 띄운다
3	키보드에 클릭된 값들을 불러온다
4	텍스트를 출력한다
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

화면명	일기 저장하기	최종 수정일	18-04-03	이전 수정일	
Location				page	

Project	프로젝트명



Description (화면설명)	
1	일기 작성 페이지에서 저장 버튼 클릭
2	사용자가 작성한 내용을 담을 객체를 생성
3	생성한 객체 정보 DB에 저장
4	DB에 저장 후 메인 화면 출력
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

6주차 · 7주차 상세설계

알고리즘 & 슈도코드 & 레이아웃

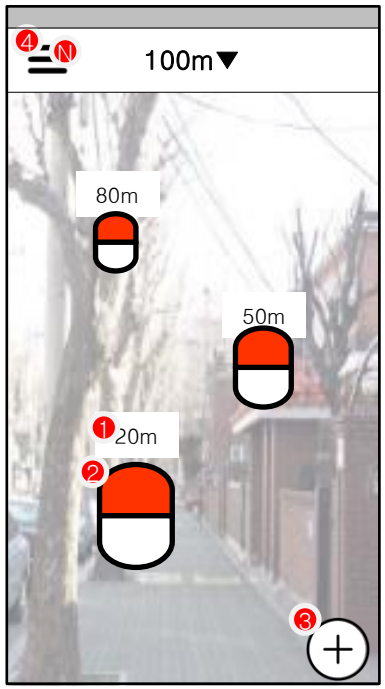
화면명	데이터베이스 구조	최종 수정일	18.04.20	이전 수정일	18.04.09

Capsule Table

NAME	TYPE	PK	NULL
INDEX (저장 시 순서대로 부여되는 번호)	INTEGER		
LATITUDE (위도)	DOUBLE	Y	
LONGITUDE (경도)	DOUBLE	Y	
CREATE_DATE (일기장을 작성한 날짜)	TIMESTAMP		
CONTENT (일기장에 작성한 내용)	TEXT		
PICTURE (일기장에 첨부한 사진)	BLOB		

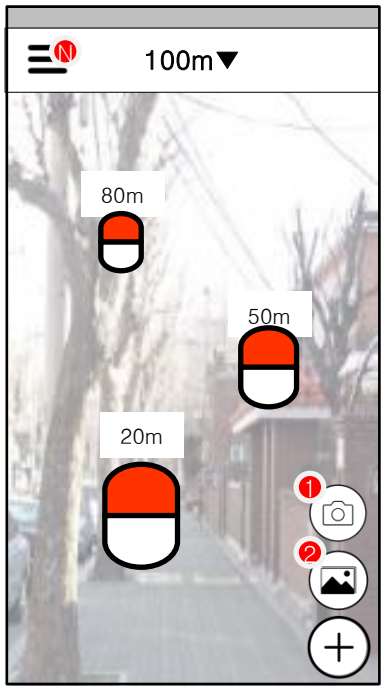
화면명	메인 화면/작성 날짜/푸시알림	최종 수정일		이전 수정일	

Title	Activity_main.xml



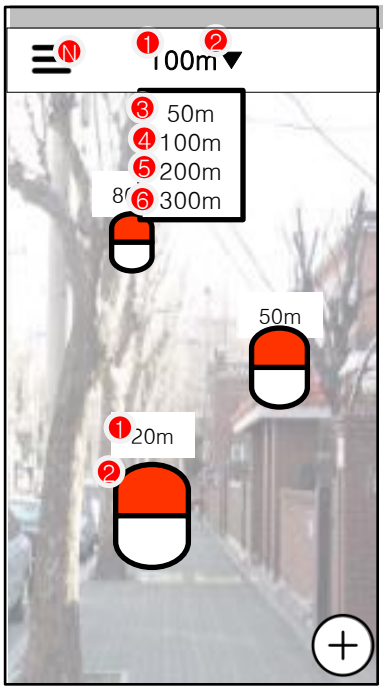
Description(화면설명)	
1	현재 위치에서부터 캡슐의 위치까지의 거리 / distanceText
2	일기장의 정보를 담고 있는 캡슐 모양의 앵커 / capsuleAnchor
3	클릭 시 카메라, 갤러리를 선택할 수 있는 플로팅 버튼 / floatingBtn
4	일기장 작성 날짜를 확인할 수 있는 메뉴 버튼 / openSaveDateBtn

Title	Activity_main.xml



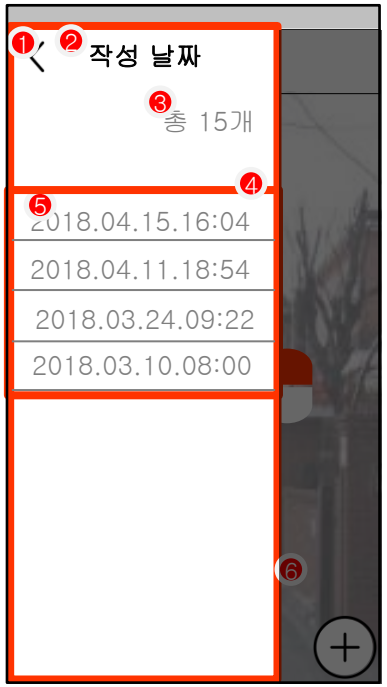
Description(화면설명)	
1	카메라 버튼 / CameraBtn
2	갤러리 버튼 / GalleryBtn
3	
4	
5	

Title	Activity_main.xml



Description(화면설명)	
1	캡슐을 찾을 현재 반경 텍스트 / CurrentRangeText
2	반경 리스트를 출력할 버튼 / rangeBtn
3	50m 반경 텍스트 / range50mText
4	100m 반경 텍스트 / range100mText
5	200m 반경 텍스트 / range200mText
6	300m 반경 텍스트 / range300mText

Title	Activity_drawer.xml



Description(화면설명)	
1	작성 날짜 리스트를 닫는 버튼 / closeSaveDataBtn
2	작성 날짜 텍스트 / saveDataText
3	총 작성한 캡슐의 개수 텍스트 / totalCapsuleText
4	작성한 날짜를 담을 리스트 / saveDataList
5	작성한 날짜 (리스트 어댑터의 아이템) / saveDataItem
6	작성 날짜 리스트를 담을 뷰 / drawerLayout

Title	Push

Description(화면설명)	
1	상단 바에 출력될 이미지 / pushSmallImage
2	알림 확인 시 출력될 이미지 / pushBigImage
3	푸시 알림 제목 / pushTitleText
4	푸시 알림 내용 / pushContentText
5	

화면명	카메라 실행/갤러리열기	최종 수정일		이전 수정일	

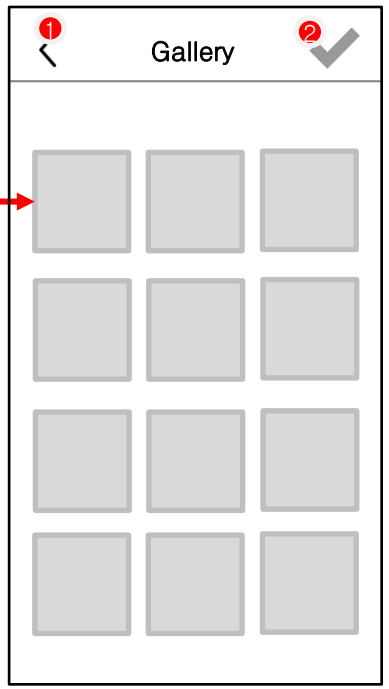
Title	

Description(화면설명)	
1	카메라불러오기
2	사진찍기버튼/takePhotoBtn
3	화면전환버튼/
4	
5	

Title	

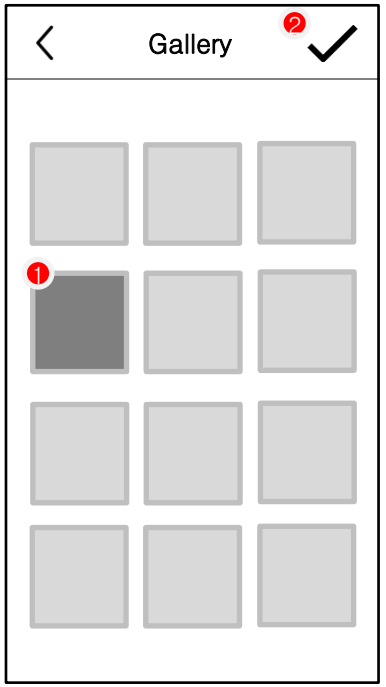
Description(화면설명)	
1	다시찍기 다이얼로그 출력 /RetryDialogue
2	다시찍기/RetryBtn
3	저장하기/SaveBtn
4	
5	

Title	



Description(화면설명)	
1	뒤로가기/ GoBackBtn
2	비활성체크버튼/checkOffBtn
3	기본갤러리출력
4	
5	

Title	



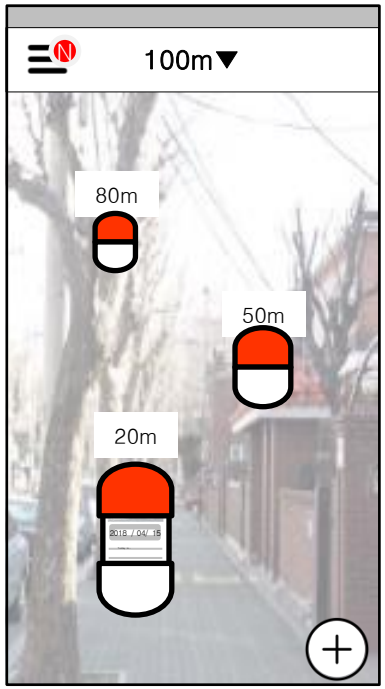
Description(화면설명)	
1	체크된 사진/selectedPhoto
2	활성화된체크버튼 /checkOnBtn
3	
4	
5	

화면명	일기 작성/일기 확인	최종 수정일		이전 수정일	

Title	writeDiary.xml

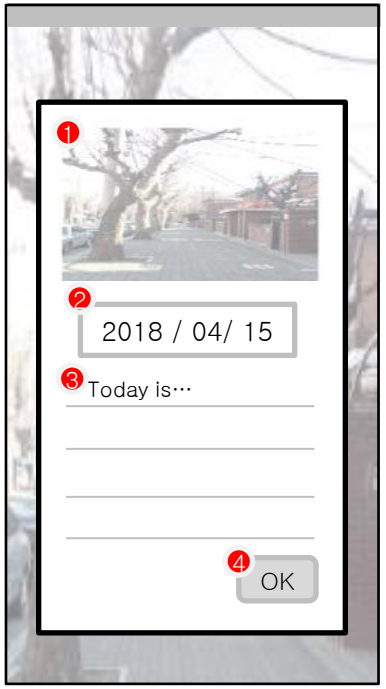
Description(화면설명)	
1	선택된 사진 띄우는 ImageView / selectedPictureImageView
2	날짜 보여주는 TextView / dateTextView
3	DatePickerDiaglog 불러오는 버튼 / changeDateBtn
4	텍스트 작성하는 EditText / writeContentEditText
5	저장하기 버튼/ saveDiaryBtn

Title	openDiary.xml



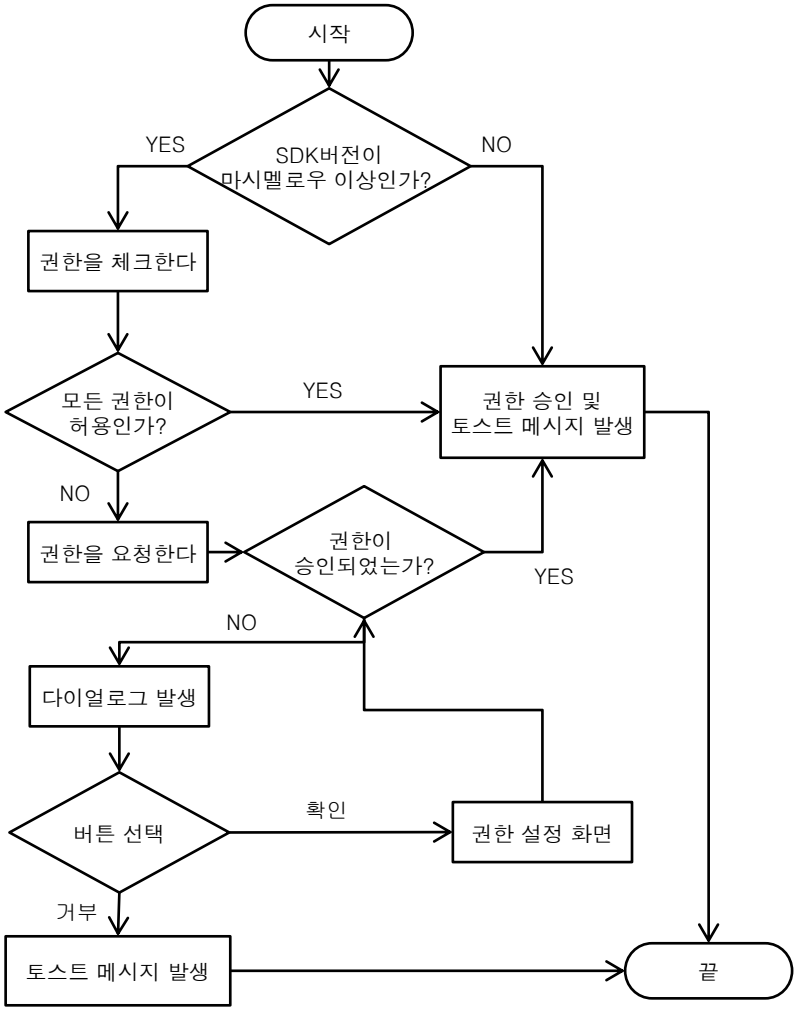
Description(화면설명)	
1	
2	
3	
4	
5	

Title	openDiary.xml

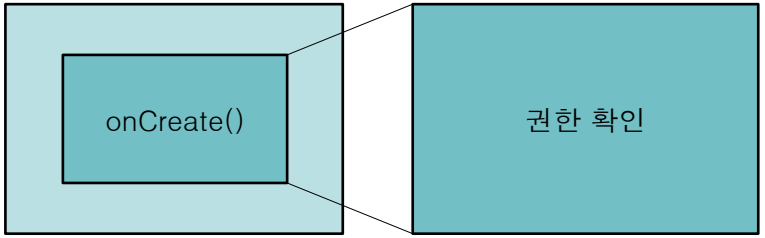


Description(화면설명)	
1	사진 출력하는 ImageView / showPictureImageView
2	날짜 출력하는 TextView / showDateTextView
3	텍스트 출력하는 TextView / showContentTextView
4	일기 닫기 버튼 / closeDiaryBtn

화면명	권한 받아오기	최종 수정일	18.04.10	이전 수정일	18.04.07



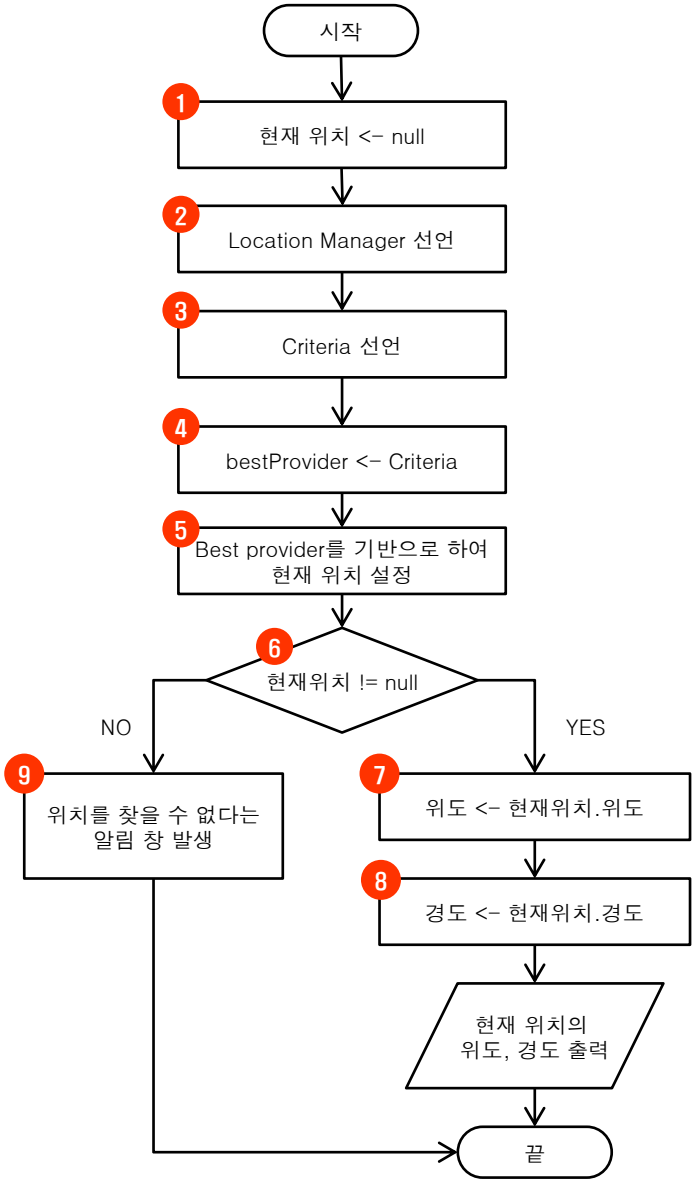
MainActivity.java



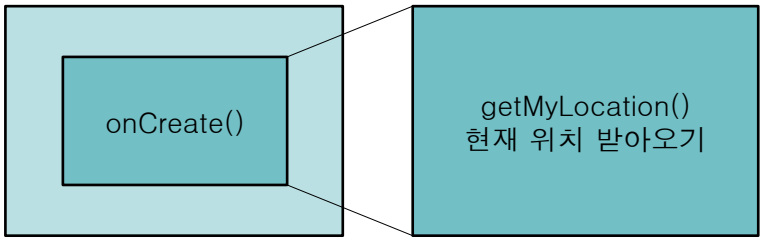
* 권한 불러오기 라이브러리인 TedPermisson Library를 사용함.
(https://github.com/ParkSangGwon/TedPermission)

```
1 // Permission 리스너 생성
Permission Listener permissionlistener = new PermissionListener() {

2  권한이 모두 허용되었을 경우{
                // 토스트 메시지 발생
                Toast.("반갑습니다");
                }

3  권한이 허용되지 않았을 경우{
                // 토스트 메시지 발생
                Toast.("권한이 거부되었습니다");
                }
};

TedPermission.with(this)
        4 // 리스너와 연결
          .setPermissonListener(permissionListener)

        5 // 알림 창에 쓰일 메시지 작성
          .setRationaleMessage
          ("공간캡슐을 실행하기 위해서는 카메라, 위치, 저장소 권한이 필요합니다.
          확인을 눌러주세요!");

        6 //요청할 권한 설정
          .setPermissions(카메라, 위치, 저장소);
```

화면명	현재 위치 받아오기	최종 수정일	18.04.15	이전 수정일	18.04.10

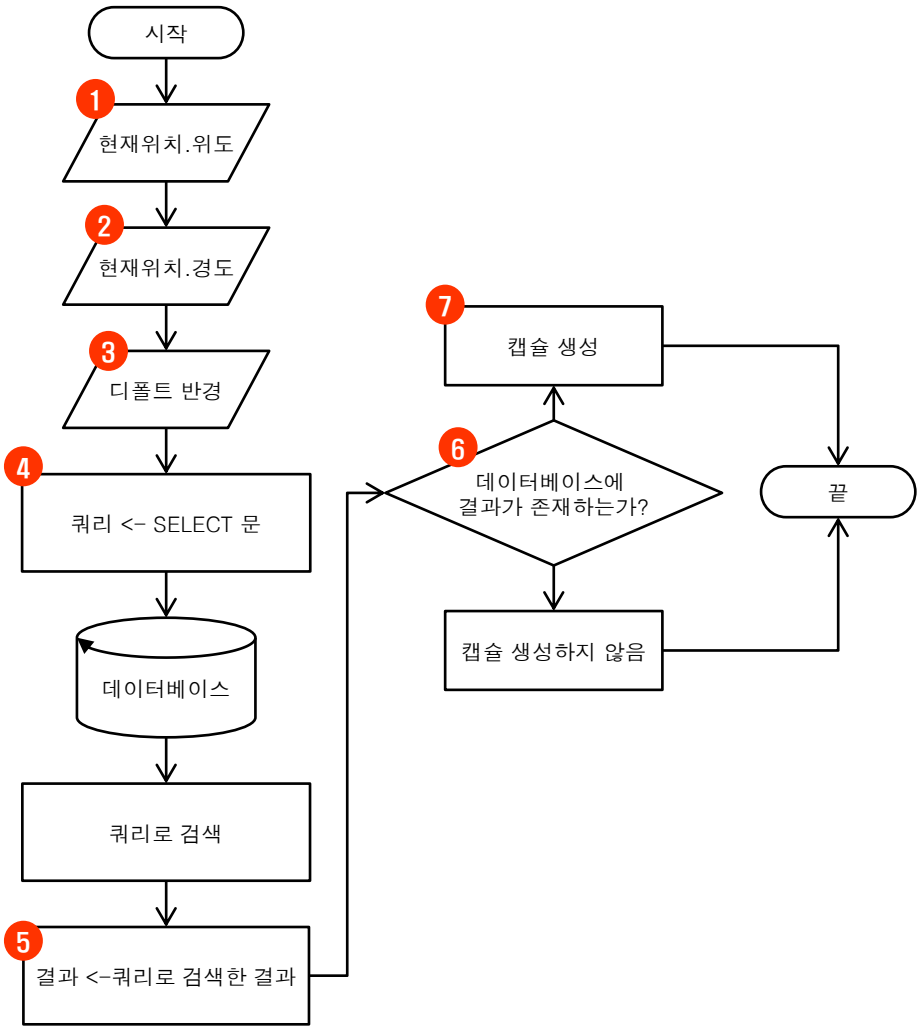


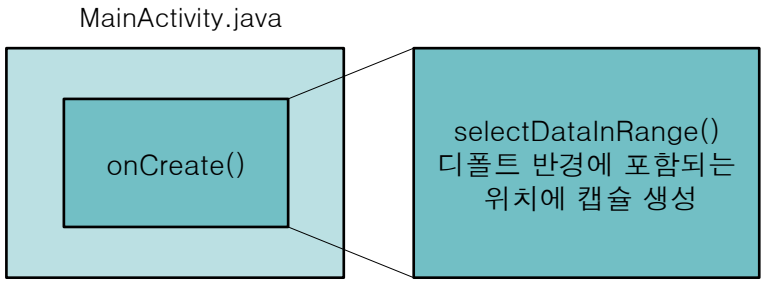
MainActivity.java




```
// 현재 위치를 반환하는 함수 정의
Location getMyLocation() {
1   현재위치 = null;     // 현재 위치 null로 초기화.

2   // Location Mangager 선언
    LocationManager locationManager = (LocationManager)
    this.getSystemService( Context.LOCATION_SERVICE);

3   // Criteria 선언
    Criteria criteria = new Criteria();
    // 정확도
    criteria.setAccuracy(Criteria.NO_REQUIREMENT);
    // 전원 소비량
    criteria.setPowerRequirement(Criteria.NO_REQUIREMENT);
    // 고도, 높이 값을 얻어 올지를 결정
    criteria.setAltitudeRequired(false);
    // provider 기본 정보(방위, 방향)
    criteria.setBearingRequired(false);
    // 속도
    criteria.setSpeedRequired(false);
    // 위치 정보를 얻어 오는데 들어가는 금전적 비용
    criteria.setCostAllowed(true);

4   // 최적의 위치 정보 프로바이더 선언
    bestProvider = getBestProvider(new Criteria(),true);

5   // 현재 위치 설정
    현재위치 = getLastKnownLocation( bestProvider );

6   // 현재위치가 null이 아닐 경우 위도와 경도를 받아옴
    if (현재위치 != null) {
7   double 위도 = 현재위치.getLongtitude();
8   double 경도 = 현재위치.getLatitude();
    }

9   // 현재위치가 null일 경우 토스트 메시지 발생
    else
    Toast.makeText('위치를 찾을 수 없습니다.");
```

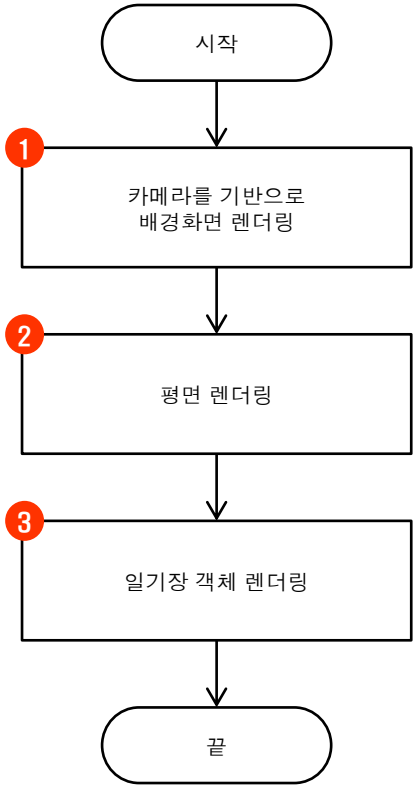
화면명	디폴트 반경에 포함되는 위치에 캡슐 생성	최종 수정일	18.04.15	이전 수정일	18.04.10





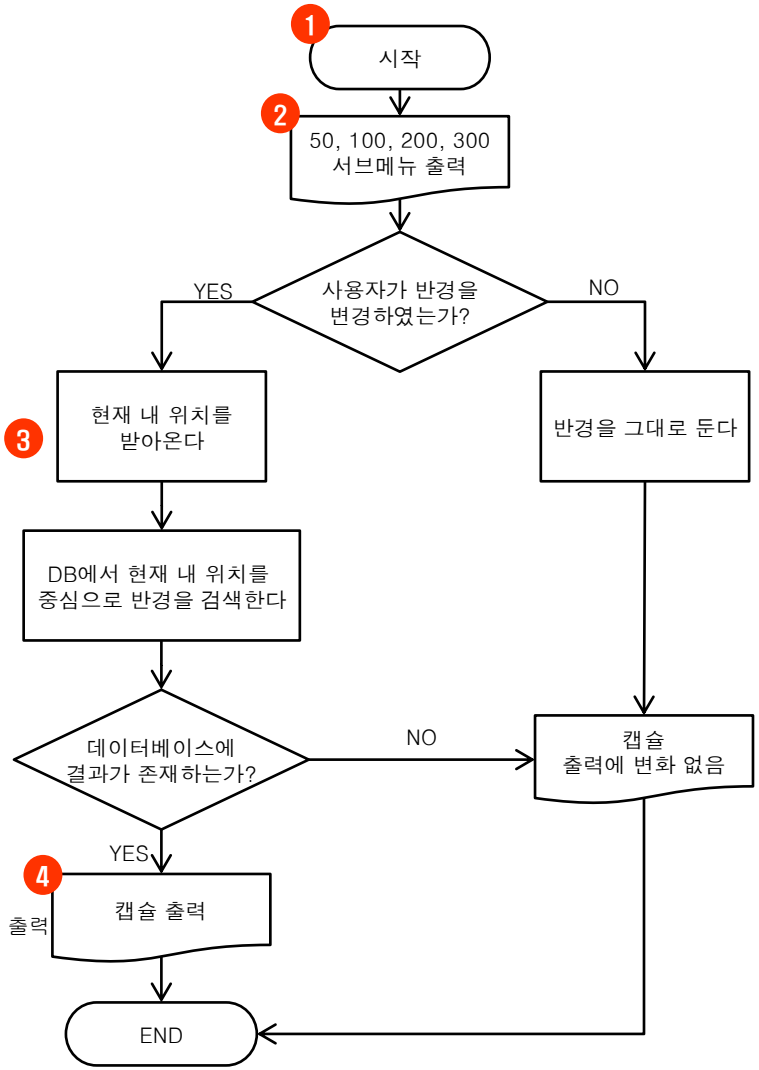

```
// 현재위치와 디폴트 반경 300m를 이용하여 범위 계산

// 리스트 캡슐 생성
ArrayList <캡슐> 캡슐리스트 = new ArrayList<>();

Int selectDataInRange(현재위치.위도, 현재위치.경도, 300m) {

    // SELECT 쿼리문 생성
    String 쿼리 = "SELECT *,
    (6371*acos(cos(radians(현재위치.위도))*cos(radians(데이터베이스.위도컬럼))
    *cos(radians(데이터베이스.경도컬럼)
    -radians(현재위치.경도))+sin(radians(현재위치.위도))
    *sin(radians(데이터베이스.위도컬럼))))
    AS 300m
    FROM 데이터베이스
    HAVING 300m <= 설정반경
    ORDER BY distance";

    // 쿼리문의 결과를 저장.
    // 결과는 현재 위치에서부터 300m까지의 범위 중
    데이터베이스에 저장된 데이터의 튜플임.
    Cursor 결과 = 데이터베이스.rawQuery(쿼리, null);

    // 데이터베이스에 결과가 존재한다면
    if (결과.moveToNext())
    Capsule capsule = new Capsule();
    capsule = 결과
    캡슐리스트.add(capsule);

    // 캡슐 생성
    createAnchor(캡슐리스트);
}
```

화면명	메인 화면 로딩	최종 수정일	18.04.15	이전 수정일	18.04.10



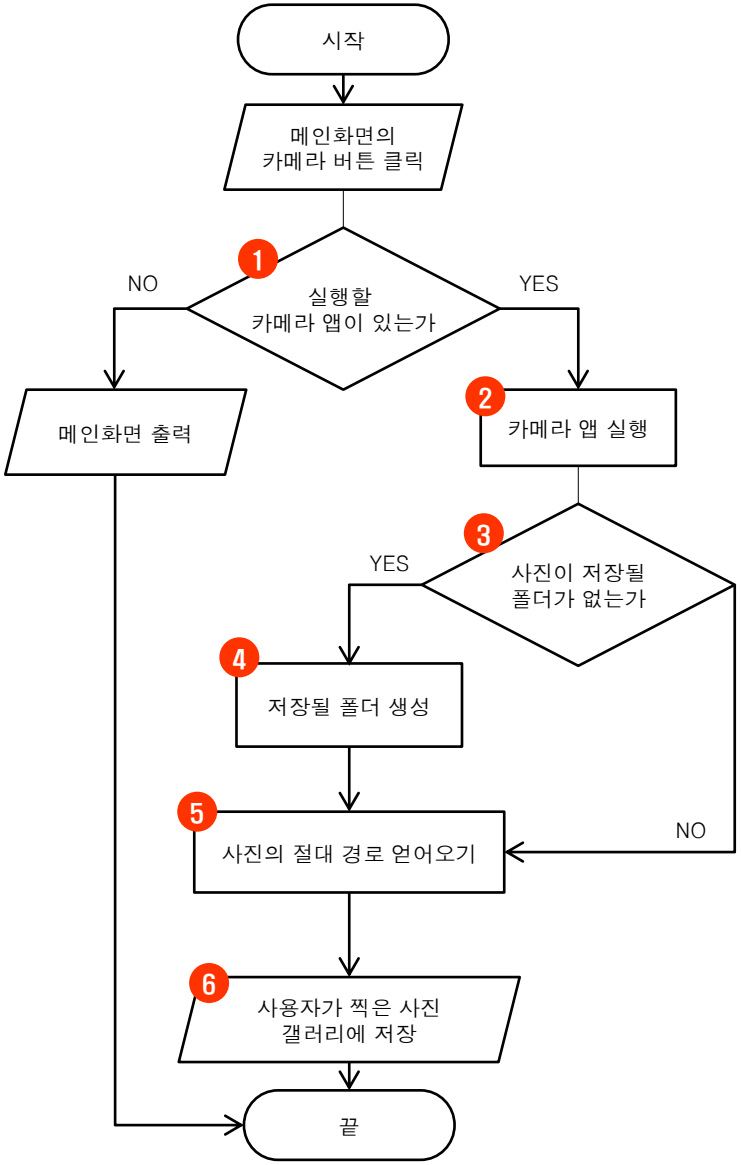

```
// 캡슐을 3D로 띄우기 위해 OpenGL의 함수 선언
public void onSurfaceCreate(GL10 gl, EGLConfig config) {

        // 배경 렌더링 <- arCore 코드에서 클래스 불러옴
1       backgroundRenderer.createOnGlThread(this);

        // 캡슐 렌더링
2       virtualObject.createOnGlThread(this, “캡슐.obj", “캡슐.png");
}

public void onDrawFrame(GL10 gl) {
        for (Anchor 캡슐 : 캡슐리스트) {
        anchor.getPose().toMatrix(anchorMatrix, 0); // 캡슐의 위치 가져오기
3
        // 캡슐 띄우기
}
}
```

화면명	반경설정하기	최종 수정일	18.04.10	이전 수정일	18.04.08



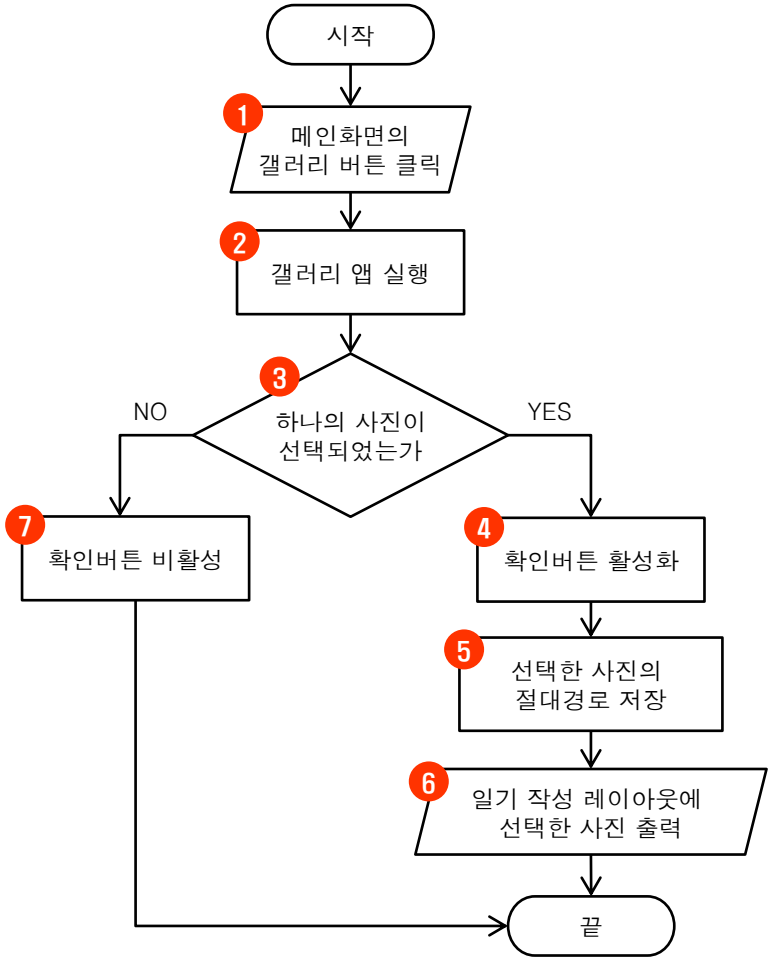

```
// 반경 설정하기

1 onClick(View v ) {
                반경 설정 옵션창 출력
  }
2 switch( v. getID() ) {
  Case 50m : getMyLocation(50m)
  Case 100m : getMyLocation(100m)
  Case 200m : getMyLocation(200m)
  Case 300m : getMyLocation(300m)
  }

  // 리스트 캡슐 생성
  ArrayList <캡슐> 캡슐리스트 = new ArrayList<>();

3 Int selectDataInRange(현재위치.위도, 현재위치.경도, 반경) {

                // SELECT 쿼리문 생성
                String 쿼리 = "SELECT *,
                (6371*acos(cos(radians(현재위치.위도))*cos(radians(데이터베이스.
  위도컬럼))
                *cos(radians(데이터베이스.경도컬럼)
                -radians(현재위치.경도))+sin(radians(현재위치.위도))
                *sin(radians(데이터베이스.위도컬럼))))
                AS 반경
                FROM 데이터베이스
                HAVING 반경
                ORDER BY distance";

                // 쿼리문의 결과를 저장.
                // 결과는 현재 위치에서부터 300m까지의 범위 중
                데이터베이스에 저장된 데이터의 튜플임.
                Cursor 결과 = 데이터베이스.rawQuery(쿼리, null);

4               // 데이터베이스에 결과가 존재한다면
                if (결과.moveToNext())
                Capsule capsule = new Capsule();
                capsule = 결과
                캡슐리스트.add(capsule);

                // 캡슐 생성
                createAnchor(캡슐리스트);
  }
```

화면명	카메라 앱 실행 및 저장하기	최종 수정일	18.04.10	이전 수정일	18.04.08



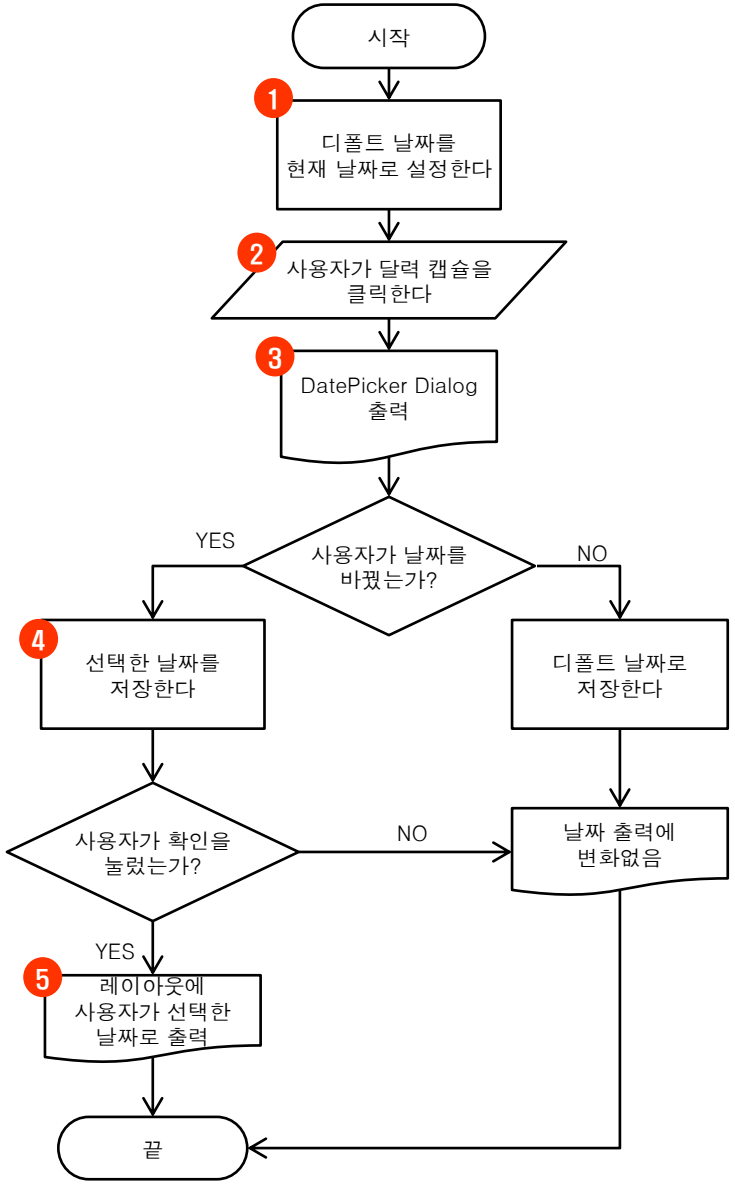

```
카메라 캡슐 버튼.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if( 실행할 카메라 앱이 있는가 ) {

            // 카메라 앱 실행
            Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

            if( 사진이 저장될 폴더가 없는가 ) {
                // 사진 파일이 저장될 폴더 생성
            }

            // 사진 파일의 절대 경로를 얻음
            사진 경로 = file.getAbsolutePath();

            // 찍은 사진 갤러리에 저장
            Intent mediaScanIntent =
                    new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);
            File f = new File(imagePath);
            Uri contentUri = Uri.fromFile(f);
            galleryIntent.setData(contentUri);
            this.sendBroadcast(galleryIntent);

        } else {
            // 메인화면으로
            goToMain();
        }
}});
```

화면명	갤러리에서 사진 선택하기	최종 수정일	18.04.10	이전 수정일	18.04.08




```
1 갤러리 캡슐 버튼.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
2       // 갤러리앱 실행
        Intent intent = new Intent(Intent.ACTION_GET_CONTENT);

        //Intent 시작 - 갤러리 앱을 실행하여 원하는 이미지를 선택할 수 있게 함
        startActivityForResult(Intent.createChooser(intent, "Select Picture",
                               PICK_IMAGE_REQUEST);

3       if( 하나의 사진이 선택되었는가 ) {

4       // 확인 버튼 활성화

5       //data에서 절대경로로 사진을 가져옴
        Uri uri = data.getData();

6       //일기 작성 레이아웃에 출력
        imgView.setImageBitmap(scaled);

        } else {

7       //확인 버튼 비활성화

    }
}});
```

화면명	날짜 설정하기	최종 수정일	18.04.10	이전 수정일	18.04.08



1
```
//현재 날짜 받아오기
onCreate(){
c= Calendar.getInstance();
int year = c.get(c.YEAR);
int month = c.get(c.MONTH);
int dayOfMonth = c.get(c.DAY_OF_MONTH);

연도 텍스트필트.setText(year);
월 텍스트필드.setText(month);
일 텍스트필드.setText(dayOfMonth);
}
```

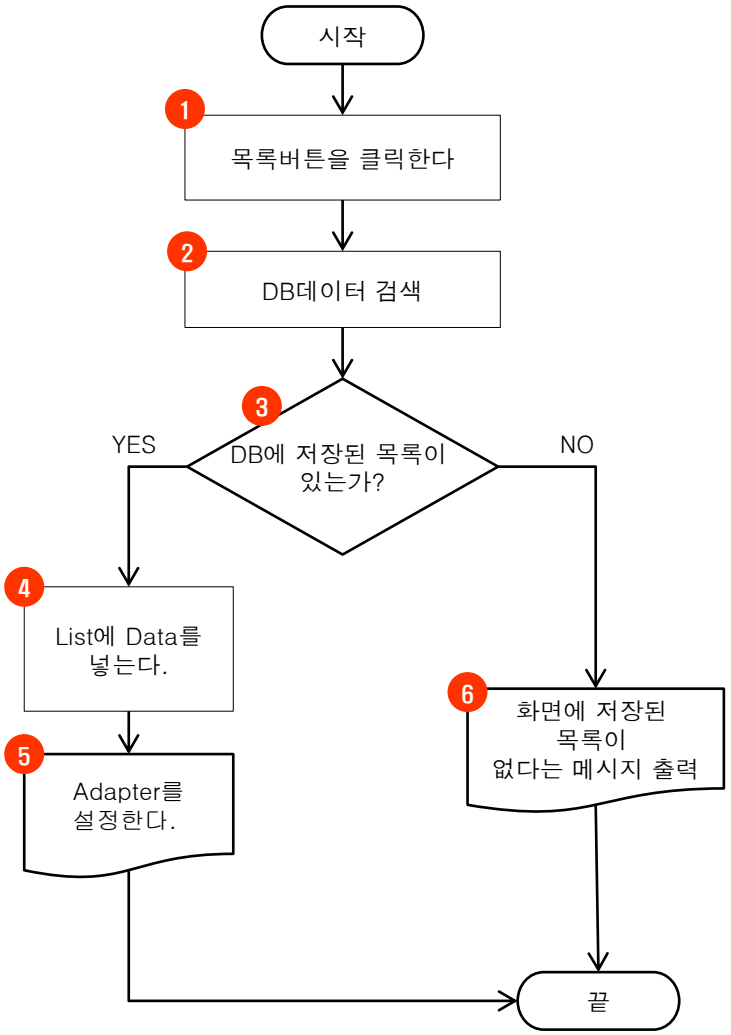
2
```
달력캡슐.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
```

3
```
    DatePicker dialog로 달력띄우기 //달력 출력

//날짜 변경 이벤트를 받기 위한 리스너
setOnDateSetListener {
//Set버튼 클릭해서 변경을 완료했을 때 호출
```

4
```
public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth){
mYear = year;
mMonth = monthOfYear;
mDay = dayOfMonth;
updataDisplayDate(mYear, mMonth, mDay);
}
```

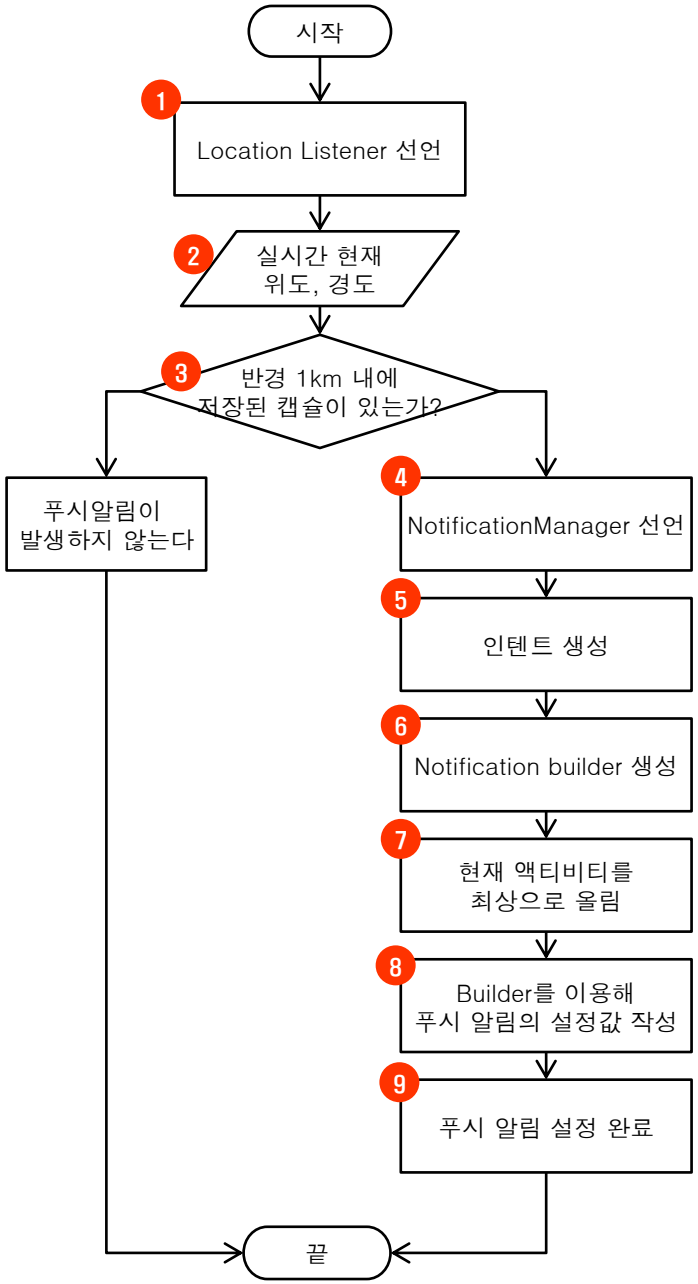
5
```
updataDisplayDate(연, 월, 일) {
// 레이아웃에 날짜 배열
연도 텍스트필트.setText(onDateSet.year);
월 텍스트필드.setText(onDateSetmonth);
일 텍스트필드.setText(onDateSetdayOfMonth);
}
```

화면명	저장한 날짜의 목록 보기	최종 수정일	18.04.09	이전 수정일	18.04.08




```
목록 버튼을 클릭한다
1 //데이터 가져오기 버튼 클릭 리스너를 만든다

날짜목록버튼.setOnClickListener(){

    onClick(View v){

        날짜목록.setVisibility(View.VISIBLE);

            2 // SELECT 쿼리문 생성
              String 쿼리 = "SELECT 날짜,
              FROM 데이터베이스

              // 쿼리문의 결과를 저장.
              Cursor 결과 = 데이터베이스.rawQuery(쿼리, null);

            3 // 데이터베이스에 결과가 존재한다면
              if (결과.moveToNext())
            4 List 날짜 = getDiaryTimeData();
              날짜.add(결과);

            5 listView.setAdapter(new TimeListAdapter(time,
              MainActivity.this));

            6 else
              Toast.makeText("저장된 목록이 없습니다.");
    }

};

Public List getDiaryTimeData(){};

Class TimeListAdapter extends BaseAdapter{};
```

화면명	푸시알림	최종 수정일	18.04.15	이전 수정일	

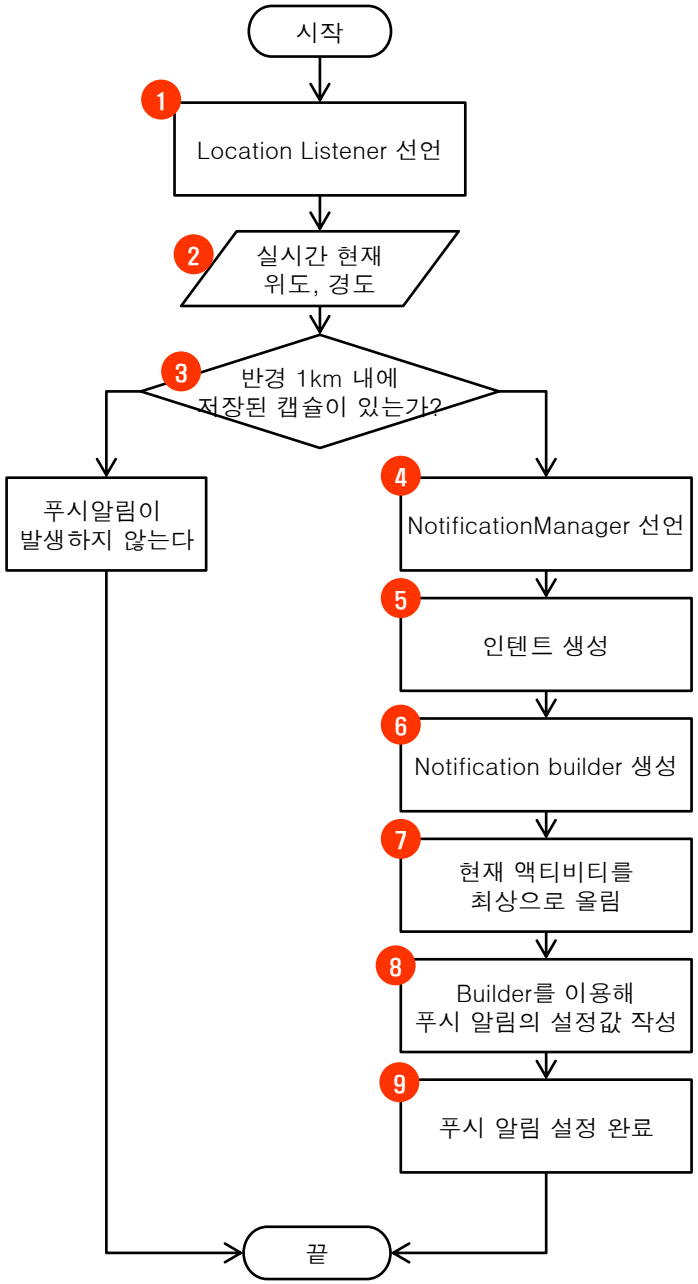


1
```
// 실시간 위치 받아오기
// location Listener 선언
private LocationListener locationListener = new LocationListener() {
```

2
```
// 현재 위치가 변할 때마다 위도, 경도를 새로 갱신함
onLocationChanged() {
double 위도 = location.getLatitude();
double 경도 = locattion.getLongitude();

...
}
};

// 반경 1km내에 캡슐이 있는지 검색
selectData(현재위치.위도, 현재위치.경도, 1km) {
        // SELECT 쿼리문 생성
        String 쿼리 = "SELECT *,
        (6371*acos(cos(radians(현재위치.위도))*cos(radians(데이터베이스.위도컬럼))
        *cos(radians(데이터베이스.경도컬럼)
        -radians(현재위치.경도))+sin(radians(현재위치.위도))
        *sin(radians(데이터베이스.위도컬럼))))

        AS 1000m
        FROM 데이터베이스
        HAVING 300m <= 설정반경
        ORDER BY distance";

        // 쿼리문의 결과를 저장.
        // 결과는 현재 위치에서부터 300m까지의 범위 중
        데이터베이스에 저장된 데이터의 튜플임.
        Cursor 결과 = 데이터베이스.rawQuery(쿼리, null);
```

3
```
        // 데이터베이스에 결과가 존재한다면
        if (결과.moveToNext())
```

화면명	푸시알림	최종 수정일	18.04.15	이전 수정일	



4 // NotificationManager 선언
NotificationManager notificatiionManger = (NotificaitionManager) MainActivity.this.
getSystemService(MainActivity.this.NOTIFICATION_SERVICE);

5 // 인텐트 생성
Intent intent = new Intent(MainActivity.this.getApplicationContext(),MainActivity.class);

6 // Notification builder 생성
Notification.Builder builder = new Notification.Builder(getApplicationContext());
intent.naddFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);

7 //현재 액티비티를 최상으로 올리고, 최상의 액티비티를 제외한 모든 액티비티를
없앰.
Intent.FLAG_ACTIVITY_CLEAR_TOP

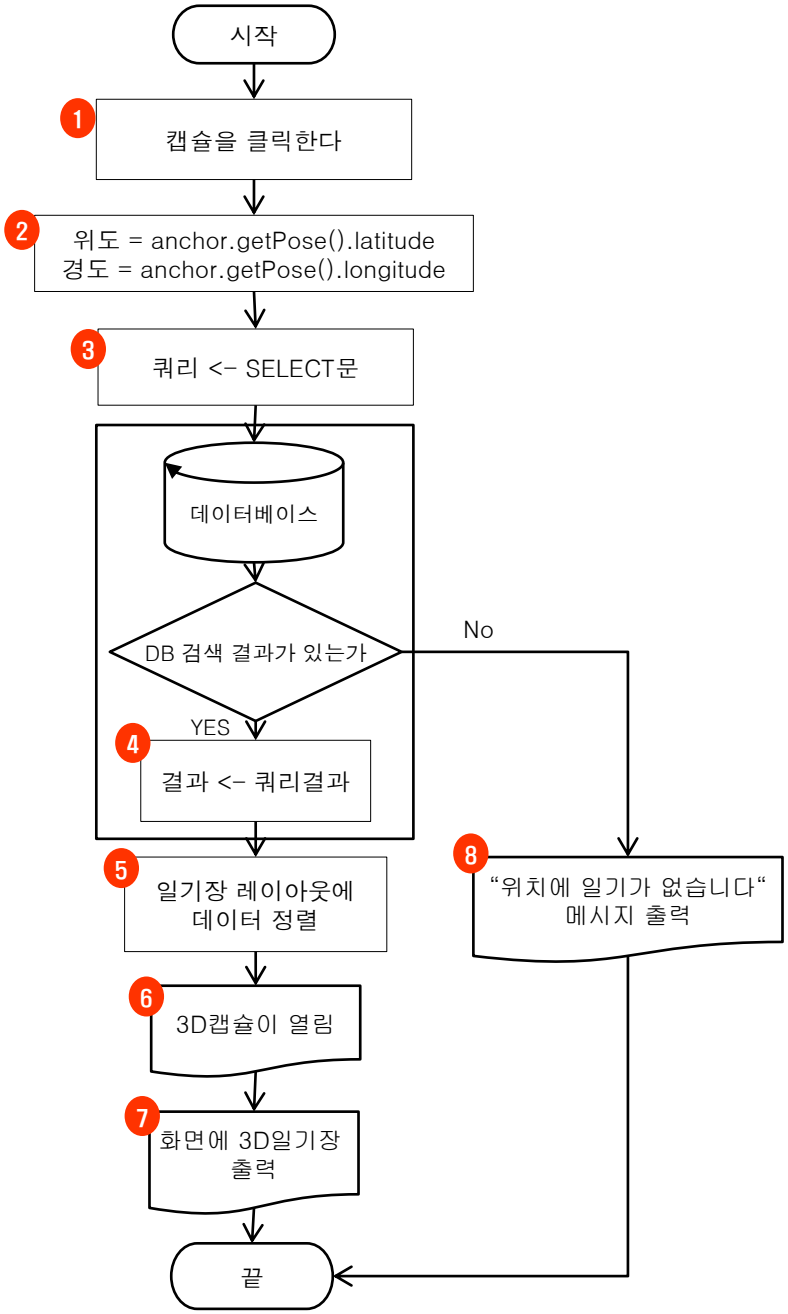
PendingIntent pendingNotificationIntent = PendingIntent.getActivity(MainActivity.this,0,
intent1,PendingIntent.FLAG_UPDATE_CURRENT);

8 builder.setSmallIcon(pushSmallImage
).setTicker("푸시 알림이 출력될 때 상단 문구").setWhen(System.currentTimeMillis())
.setNumber(1).setContentTitle(pushTitleText).setContentText(pushTitleText)
.setDefaults(Notification.DEFAULT_SOUND |
Notification.DEFAULT_VIBRATE).setContentIntent(pendingNotificationIntent).setAutoCancel(t
rue).setOngoing(true);

9 // 푸시 알림 설정 완료
notificationManager.notify(1, builder.build());

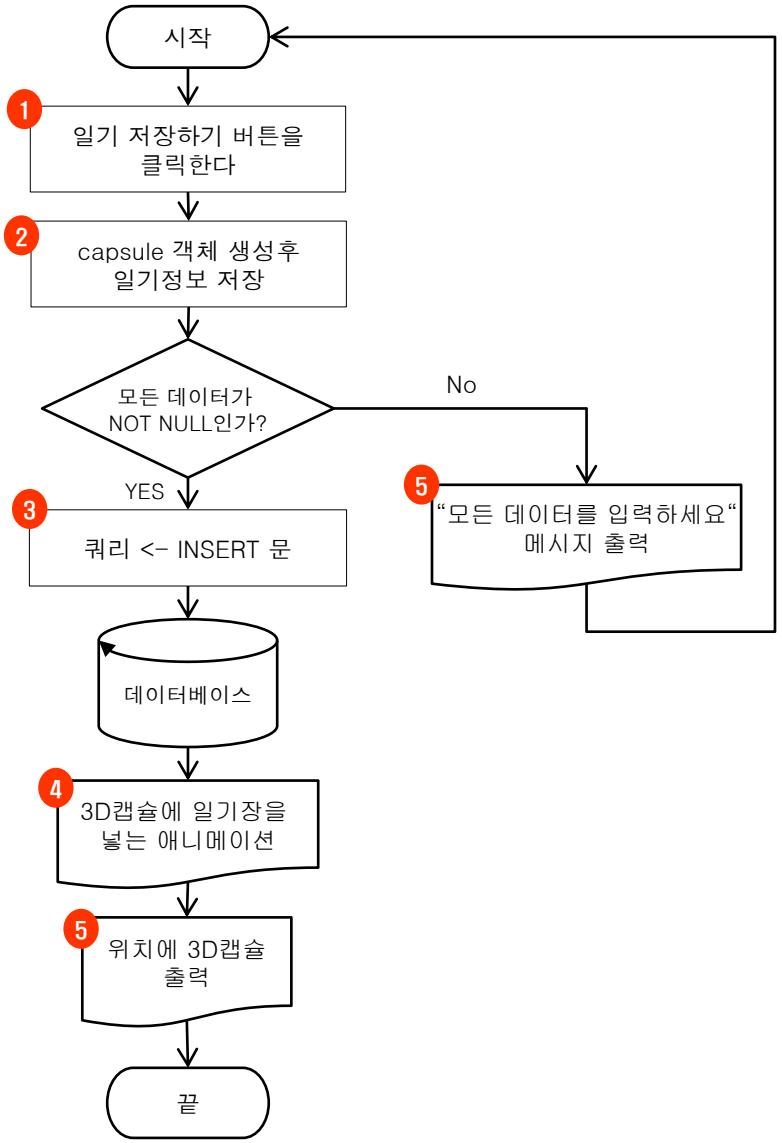
}

화면명	일기확인하기	최종 수정일	18.04.16	이전 수정일	18.04.15



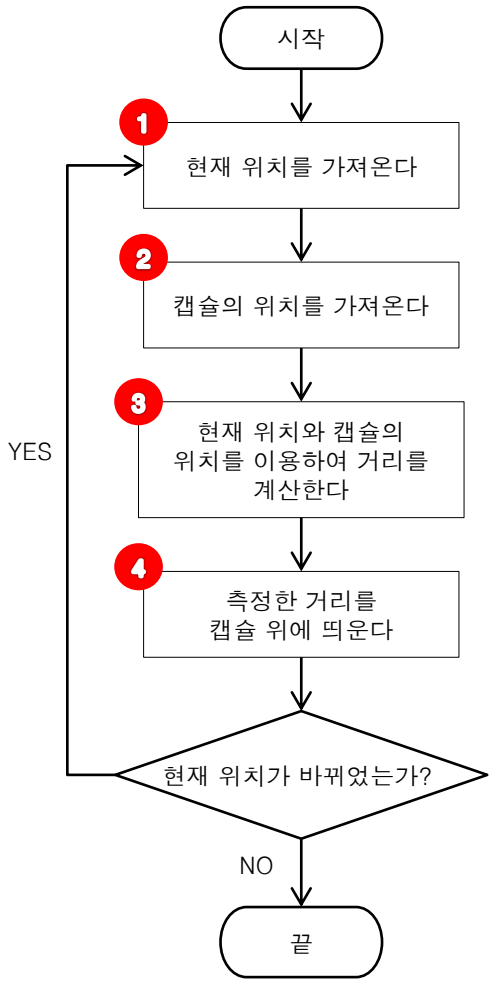

```
1 //사용자가 캡슐을 클릭한다
  캡슐.setOnClickListener(new View.OnClickListener() {
   public void onClick(View v) {

2 //캡슐(앵커)에 저장된 위도와 경도를 가져온다
  point_world = anchor.getPose().transformPoint(point_local);
  위도 = point_world.위도;
  경도 = point_world.경도;

  //DB에 위도, 경도로 SELECT
3 String 쿼리 = "SELECT * FROM CAPSULE TABLE
                 WHERE 위도="위도" AND 경도 = "경도"

  //결과에 DB 검색결과 저장
4 Cursor 결과 = 데이터베이스.rawQuery(쿼리, null);

5 if ( 결과 != null) {
  // 일기가 있으면 일기장 레이아웃에 데이터 넣고 출력
  showPictureImageView.setImageResource("결과.PICTURE");
  showDateTextView.setText("결과.USERDATE");
  showContentTextView.setText("결과.CONTENT");

6 //3D 캡슐 열림
7 //화면에 일기장 출력
  Display();

8 } else (결과 == NULL) {
  //일기가 없으면 "위치에 일기가 없습니다" 토스트 메시지 출력;
  Toast.makeToast(getApplicationContext(), "위치에 일기가 없습니다.",
  Toast.LENGTH_LONG).show();
  }

  };
```

화면명	일기저장하기	최종 수정일	18.04.	이전 수정일	18.04.15




```
1 // 사용자가 일기 저장 버튼을 클릭한다.
  saveDiaryBtn.setOnClickListener(new View.OnClickListener() {
  public void onClick(View v) {

2 // 객체 생성 후 일기 정보 저장
  Capsule capsule = new Capsule();
  capsule.setLatitude("위도");
  capsule.setLongtitude("경도");
  capsule.setCreateDate("작성 날짜");
  capsule.setPicture("사진");
  capsule.setContent("내용");

3 if (capsule 객체의 모든 속성이 NOT NULL이면) {
  // DB에 데이터 INSERT
  String 쿼리 = "INSERT INTO CAPSULE TABLE
  VALUES('인덱스','위도','경도','일기작성날짜','일기내용','사진');

5 } else {
  // 데이터에 NULL이 있으면 토스트 메시지 출력
  Toast.makeToast(getApplicationContext(), "내용을 모두 입력하세요",
  Toast.LENGTH_LONG).show();
  }

  //3D 캡슐 닫힘
4 goToMain(); //메인화면 출력
  }

  };
```

화면명	거리 측정 후 캡슐에 띄우기	최종 수정일	18.04.16	이전 수정일	18.04.15



1
```
//현재 위치 측정
Location 현재위치 = new Location("Point A");
현재위치.setLongitude(tAx);
현재위치.setLatitude(tAy);
```

2
```
//캡슐의 위치 측정
Location 캡슐의 위치 = new Location("Point B");
캡슐의 위치.setLongitude(tBx);
캡슐의 위치.setLatitude(tBy);
```

3
```
//현재 위치와 캡슐의 위치를 측정하여 거리를 계산한다.
int distance = (int) locationA.distanceTo(locationB);
```

4
```
//측정한 거리를 캡슐 위에 띄운다
 distanceText.setText(distance+"m");
```