**[야스장] 기술 문서**

**목차**

1. 문서 정보 (Document information)
2. 개요 (Overview)
3. 흐름도 (Flow chart)
4. 화면 설명 (Screen description)
5. 설계도 (Blueprint)
6. 소프트웨어 기능 (Functions and Requirements)
7. 소프트웨어 코드 (Code and Technology)
8. 소프트웨어 마케팅 (Marketing and Sales)
9. 로드 맵 (Road Map)

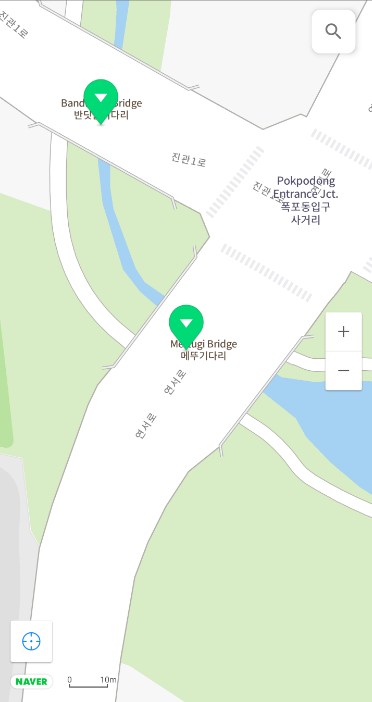
**문서 정보**

* 문서명 : [앱개발] 기술 문서
* 작성일 : 2021.05.03 ~ 2021.05.13
* 안드로이드 버전 : Android 4.1 (Jelly Bean)
* 테스트 버전 : Android 8.0 (Oreo)

**Copyright 2021. SungWon’s Friends., All Rights Reserved.**

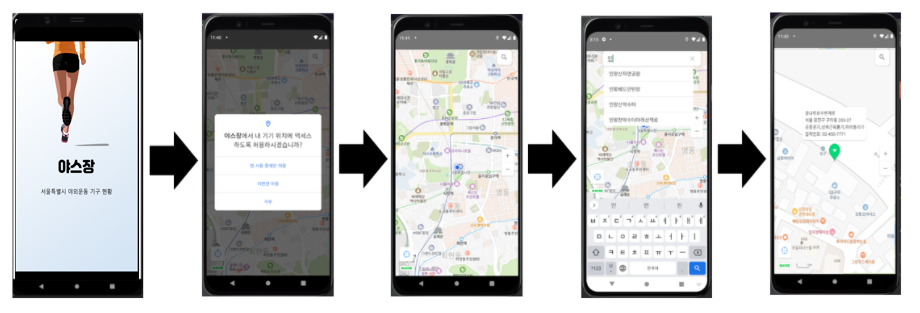
본 문서에 포함된 정보의 무단 배포, 복사 및 사용은 엄격히 금지됩니다.

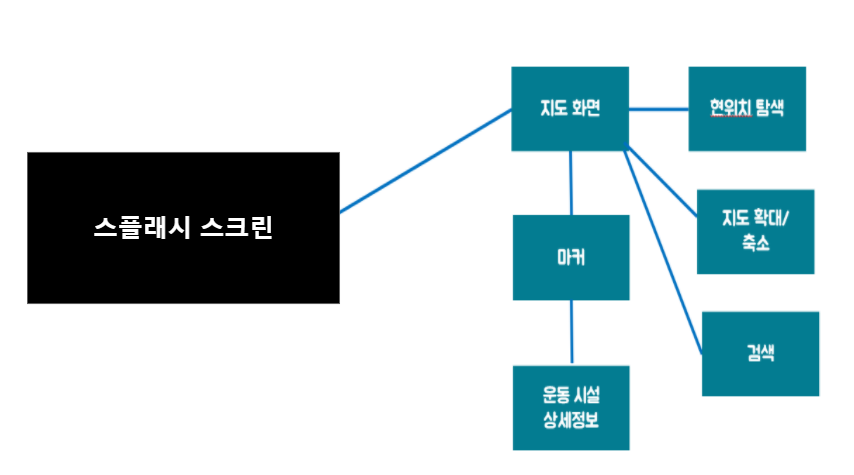
**개요**

****

이 문서는 안드로이드OS 기반의 야외 운동기구 위치 정보 제공 어플리케이션인 야스장의 소프트웨어 기능, 코드, 사용자 매뉴얼, 마케팅, 로드 맵에 대한 정보를 담고 있습니다. 유저들이 앱을 불편함없이 사용할 수 있고 원활하게 이용할 수 있도록 하는 것이 앱의 목표입니다.

**흐름도**

****

****

**화면 설명**

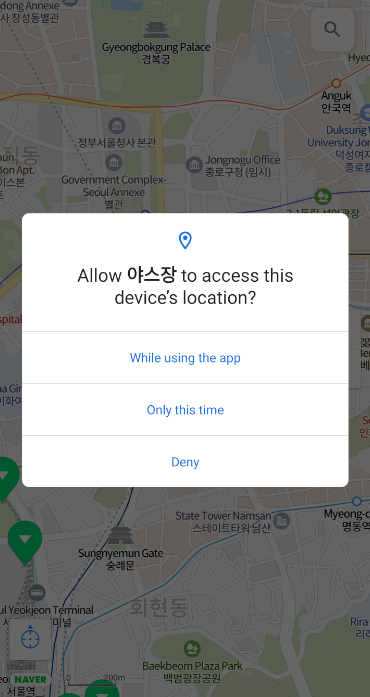
****

**스플래시 스크린**

**구성 요소: 이미지 1개, 텍스트 2개**

**이미지: 앱의 로고이다.**

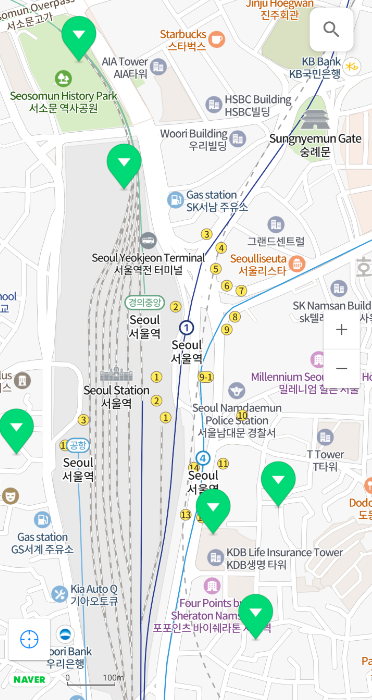
**텍스트: 앱의 이름, 부제목이 표시된다.**

****

**위치 권한 허용 화면**

**구성 요소: 팝업 1개**

**팝업: 앱 사용 중에만 허용 시 권한 허용을 추가로 할 필요가 없고, 이번만 허용 시 앱을 실행할 때마다 권한을 허용해야 한다. 거부를 누른다면 앱 이용이 불가능하다.**

****

**지도 화면**

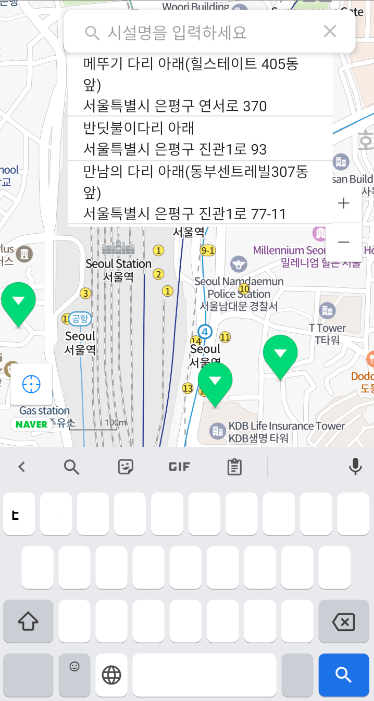
**구성 요소: 지도, 검색 버튼 1개, 확대/축소 버튼 1개, 현 위치 탐색 버튼 1개**

**지도: 지도 화면이 나온다.**

**검색 버튼: 운동 시설을 검색할 수 있는 검색 화면이 나타난다.**

**확대/축소 버튼: 지도를 확대, 축소 할 수 있다.**

**현 위치 탐색 버튼: 사용자의 현 위치로 이동한다.**

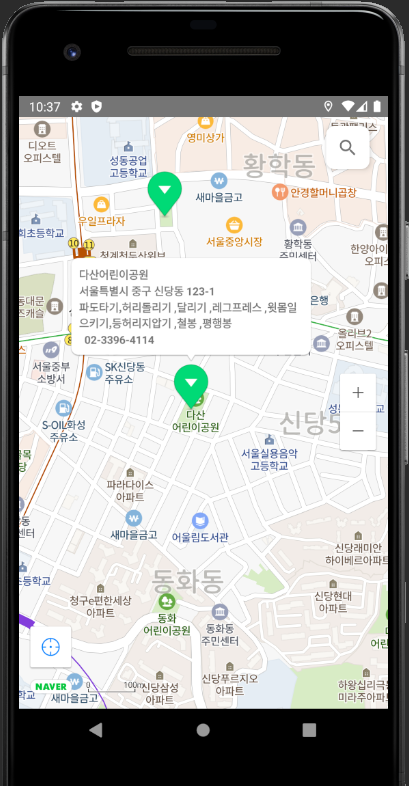
****

**검색 화면**

**구성 요소: 검색 바 1개, 검색 목록 바 1개**

**검색 바: 검색 하는 텍스트가 나타난다.**

**검색 목록 바: 운동 시설명이 나타난다.**

****

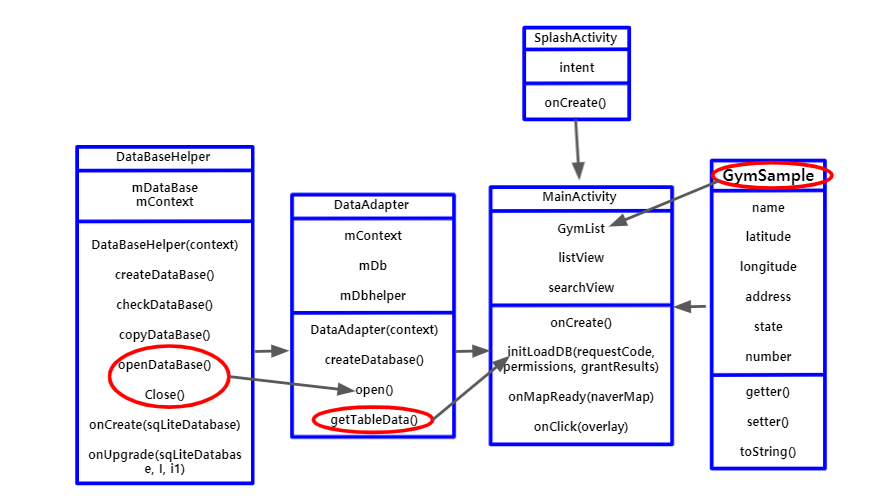
**운동 시설 상세 정보 화면**

**구성 요소: 마커, 팝업창 1개**

**마커: 운동 시설의 위치를 나타내주는 역할을 한다.**

**팝업창: 운동 시설명, 주소, 운동기구 종류, 담당 부서의 연락처가 텍스트로 표시된다.**

**설계도**

****

**소프트웨어 기능**

**(Functions and Requirements)**

1. 서울특별시 지역별 운동 시설 위치 검색
2. 현위치 탐색
3. 지도 확대/축소
4. 키워드 검색
5. 해당 운동 시설의 상세 정보 제공

**소프트웨어 코드**

**(Code and Technology)**

Xml AndroidManifest : 앱의 패키지 이름, 앱의 구성 요소, 앱이 시스템 또는 다른 앱의 보호된 부분에 액세스하기 위해 필요한 권한, 앱에 필요한 하드웨어 및 소프트웨어 기능을 정의한 xml.

Class SplashActivity : 앱 실행 시 2.5초간 뜨는 화면을 나타내는 클래스.

Class MainActivity : 지도와 마커, 검색창이 나와 야외 운동 시설을 확인하는 클래스.

Class GymSample : 야외 운동 시설의 정보를 저장하는 클래스.

Class DataBaseHelper : 데이터베이스를 열고 연결해주는 클래스.

Class DataBaseAdapter : 데이터베이스의 커서를 이용하여 데이터를 가져오는 클래스.

**Xml AndroidManifest**

****

**uses-permission : 앱이 올바르게 작동하기 위해 사용자가 반드시 부여해야 하는 시스템 권한입니다.**

**application : 애플리케이션을 선언합니다.**

**meta-data : 상위 구성 요소에 제공할 수 있는 추가적인 임의의 데이터 항목에 대한 이름-값 쌍입니다.**

**activity : 액티비티 구성 요소를 선언합니다.**

**intent-filter : 액티비티, 서비스 또는 Broadcast Receiver가 응답할 수 있는 인텐트 유형을 지정합니다.**

**action : 인텐트 필터에 작업을 추가합니다**

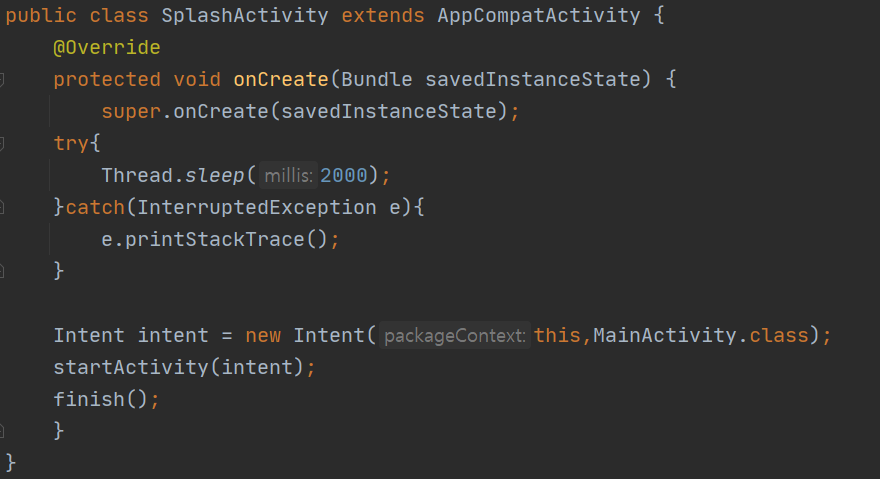
**category : 인텐트 필터에 카테고리 이름을 추가합니다.**

**android.permission.internet : 안드로이드 앱에서 인터넷으로 접속할 수 있도록 허용하는 권한입니다.**

**android.permission.ACCESS\_FINE\_LOCATION : 기기의 인터넷, 와이파이, GPS 정보를 활용해 정확한 위치를 가져오는 것입니다.**

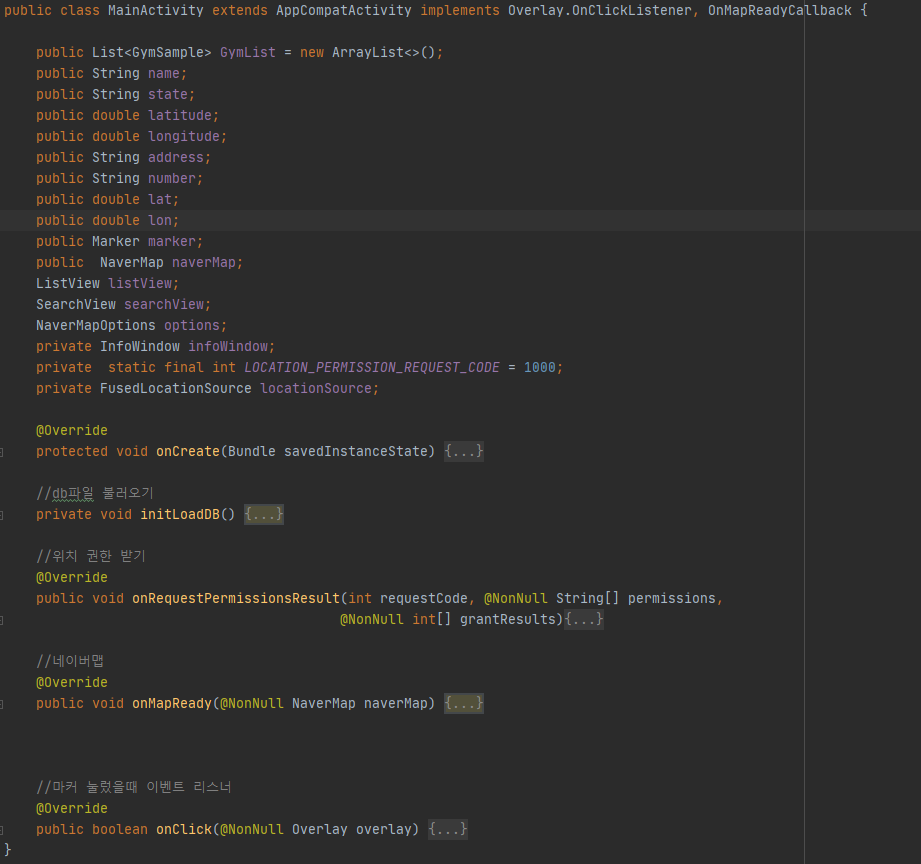
**Class SplashActivity**

**onCreate 함수**

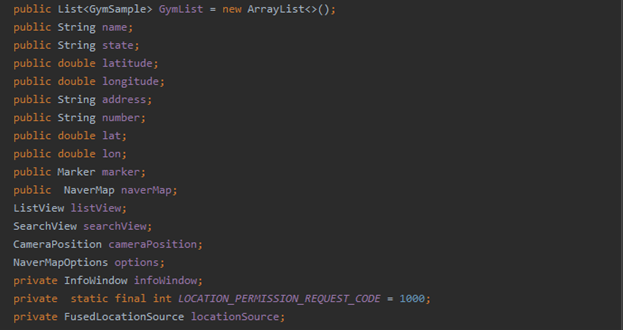


**스플래시 스크린을 2.5초간 보여주고 MainActivity로 전환한다.**

**startActivity(intent); : 지도 화면으로 전환을 해준다.**

**┗ MainActivity**

**지도를 띄우는 화면으로 운동기구 위치에 마커 생성, 마커 클릭 시 정보 표시, 검색의 기능을 수행한다.**

****

**┗ 변수들**

**위에서부터 운동기구의 정보를 넣어둘 ArrayList.**

**이름, 상태, 위도, 경도, 주소, 전화번호의 데이터를 담아둘 변수들.**

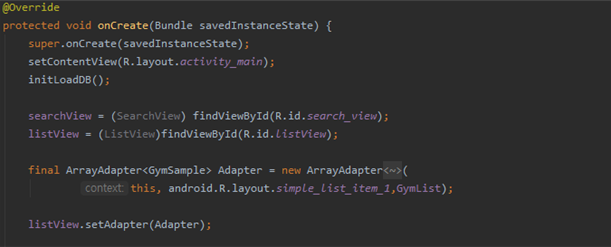
**지도에서 위치를 표시할 때 사용할 marker, 네이버 맵을 이용하기 위한 naverMap,**

**검색할 때 사용할 오브젝트인 ListView와 SearchView,**

**Marker를 눌렀을 때 정보를 표시하는 정보 창이 infoWindow,**

**GPS 사용 가능 여부를 묻는 LOCAION\_PERMISSION\_REQUEST\_CODE,**

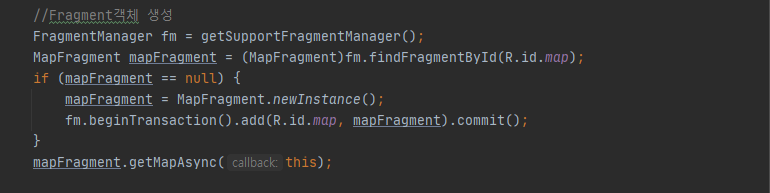
**내 위치를 담는 locationSource을 생성해 준다.**

****

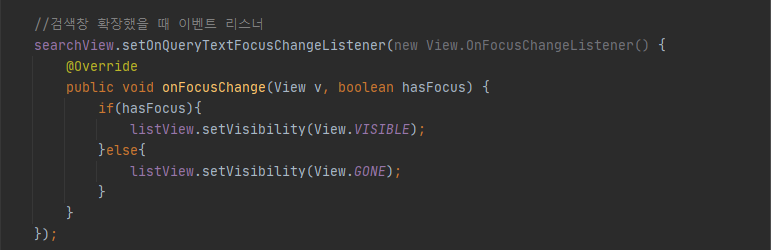
**┗ onCreate 함수**

**initLoadDB 함수를 실행하며 SearchView와 ListView 객체를 레이아웃에 연동시킨다.**

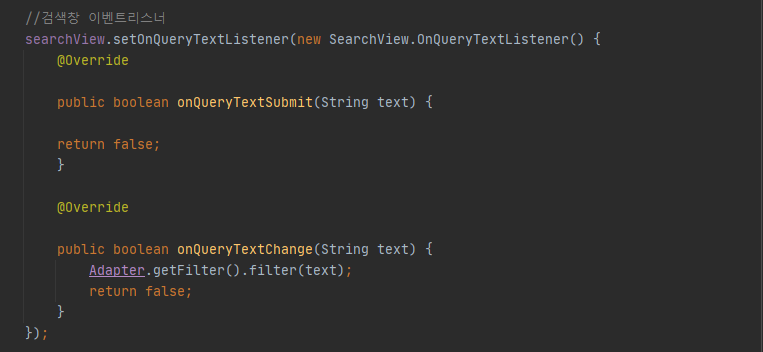
**ArrayList 객체를 ArrayAdapter 객체에 연결한 후 ListView 객체에 Adapter 객체를 연결한다.**

**┗ onCreate 함수**

**mapFragment 객체를 생성하고 mapFragment를 관리할 FragmentManager 객체도 하나 생성한다.**

**┗ onCreate 함수**

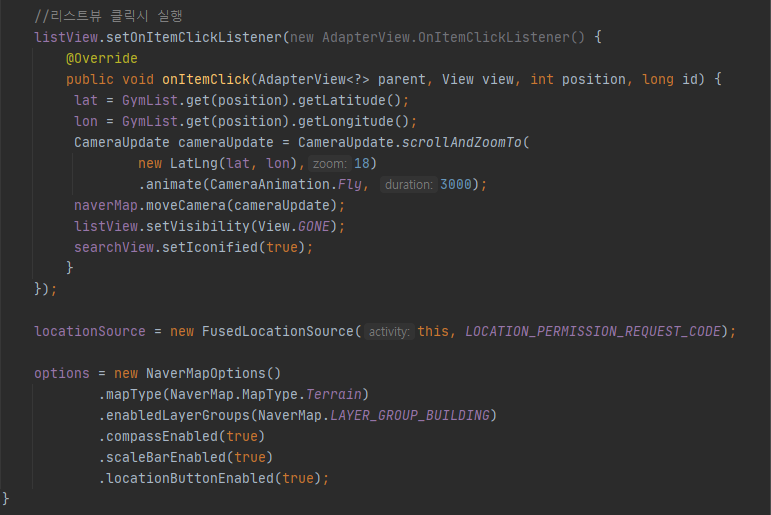
**searchview 클릭 시 listview를 보여주는 함수이다.**

**┗ onCreate 함수**

**SearchView를 이용해서 검색할 때 사용되는 함수들이다.**

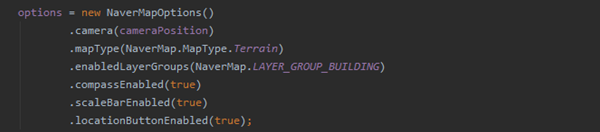
**onQueryTextSubmit 함수는 내용을 재출시 검색합니다.**

**onQueryTextChange 함수는 텍스트 변경 시 검색합니다.**

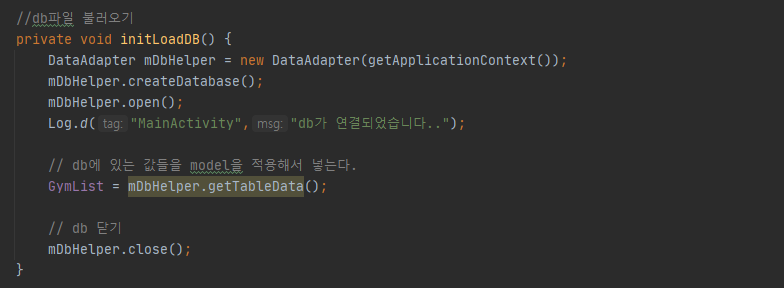
**┗ onCreate 함수**

**ListView 클릭 이벤트 함수로 클릭한 목록의 위도, 경도 값을 불러와 카메라 위치를 이동시키는 기능이다.**

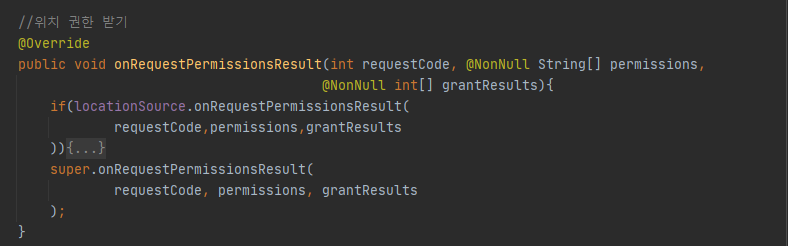
****

**┗ onCreate 함수**

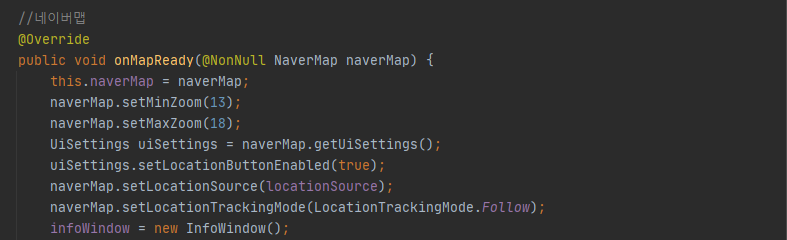
**현재 위치를 가져오기 위한 권한을 설정하는 locationSource를 명시하고 지도의 옵션을 명시해 준다.**

**┗ initLoadDB 함수**

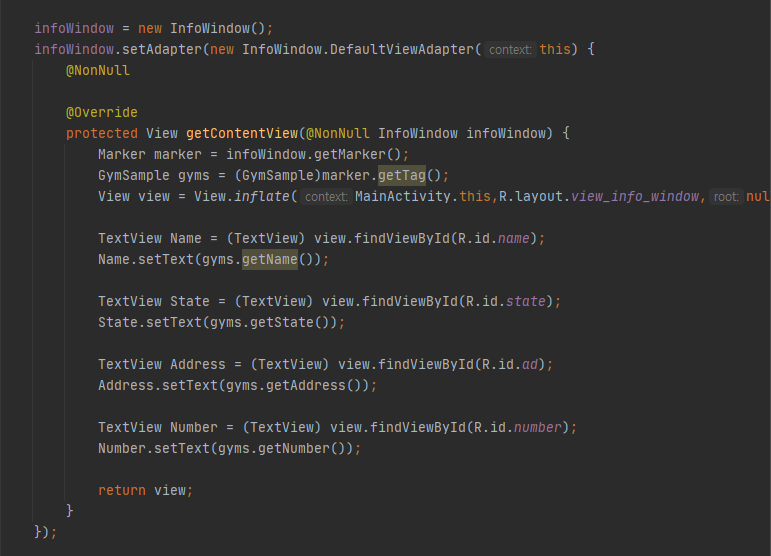
**DB에 연결하고 DB 내용을 GymList에 넣어주는 함수이다.**

**┗ onRequestPermissionResult 함수**

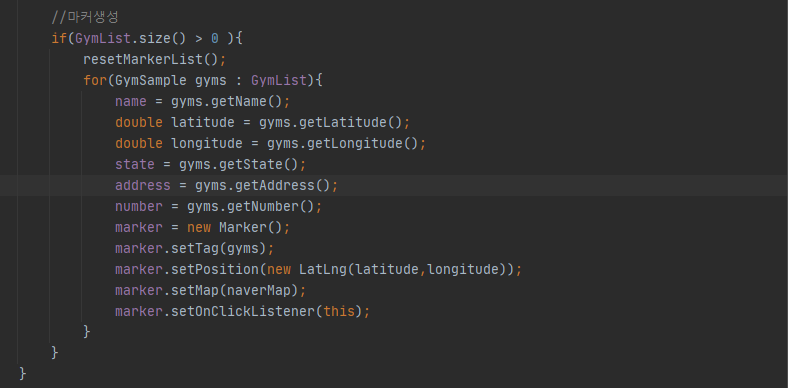
**이 앱을 사용하는데 필요한 접근 권한을 요구하는 함수이다.**

**┗ onMapReady 함수**

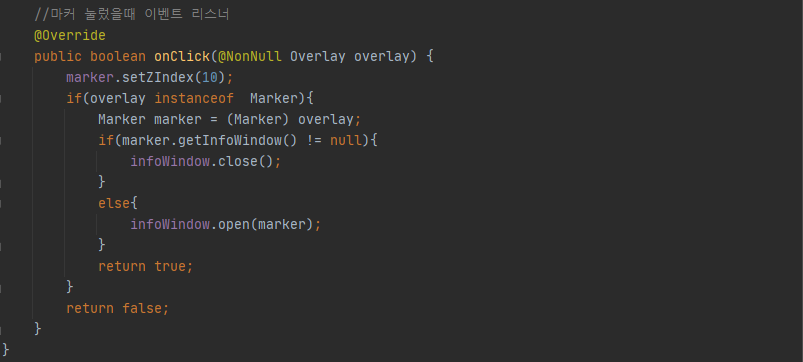
**네이버 맵 패키지 중 UI 설정을 관리하는 클래스를 이용해 UI를 설정해 준다.**

**┗ onMapReady 함수**

**맵에 마커를 생성하고 마커 클릭 시 이름, 주소, 상태, 업체 번호를 보여주는 텍스트를 설정하는 함수이다.**

**┗ onMapReady 함수**

**이름, 위도, 경도, 상태, 주소, 전화번호 데이터를 가져온 후 위도, 경도 값에 마커를 생성한다.**

**┗ onClick 함수**

**onMapReady 함수에서 설정한 텍스트를 마커 클릭 시 보여주는 함수이다.**

**Class GymSample**

****

**변수**

**Getter & Setter 함수**

**toString 함수**

**변수**

****

**Private는 접근제어자로 외부에 공개되지 않으며 외부에서 접근할 수 없습니다.**

**String name = 시설지명을 넣을 문자열 변수입니다.**

**double latitude = 위도를 넣을 실수형 변수입니다.**

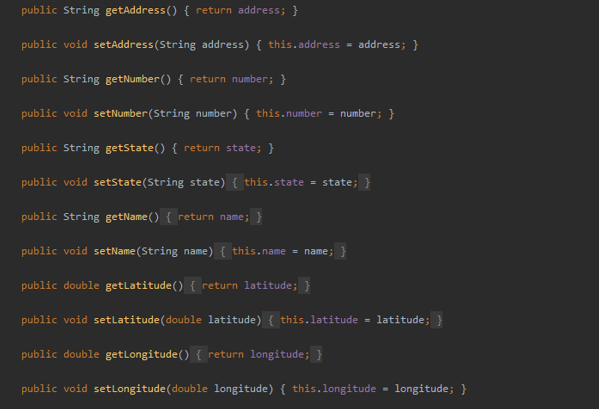
**double longitude = 경도를 넣을 실수형 변수입니다.**

**String address = 주소를 넣을 문자열 변수입니다.**

**String state = 시설물 현황을 넣을 문자열 변수입니다.**

**String number = 업체 번호를 넣을 문자열 변수입니다.**

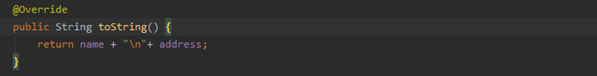
**Getter & Setter 함수**

****

**Private로 선언된 변수에 접근하기 위해 사용하는 함수입니다.**

**변수의 값을 정하는 set 변수명 함수와 변수의 값을 가져오는 get 변수명 함수입니다.**

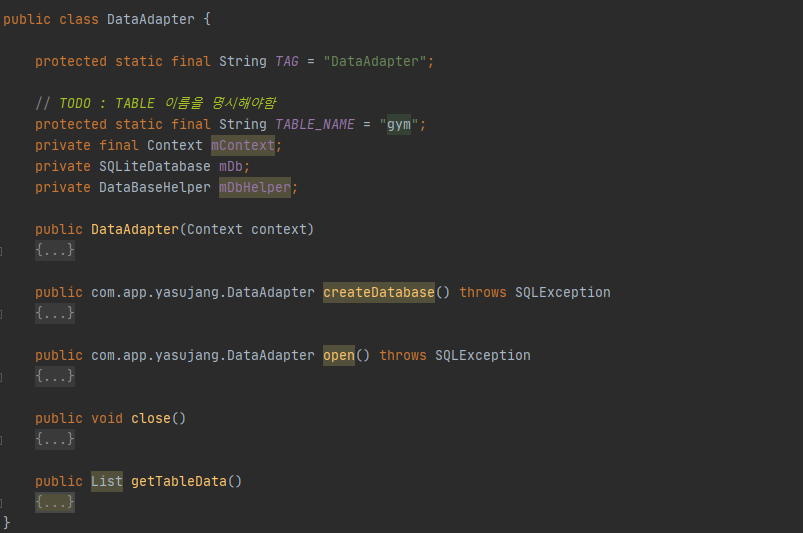
**toString 함수**

****

**변수의 값을 문자열로 반환하기 위하여 사용하는 toString 함수입니다.**

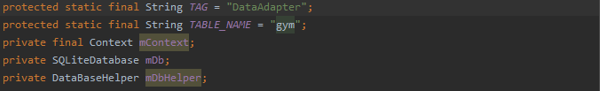
**시설지명과 주소를 문자열로 반환합니다.**

**Class DataAdapter**

****

**createDataBase();**

**변수 선언.**

****

**Public = class 외부에서 접근할 수 있는 접근자.**

**Private = class 외부에서 접근할 수 없는 접근자.**

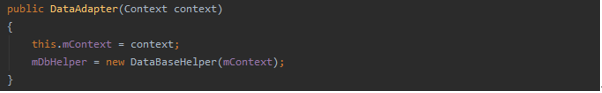
**String TAG = TAG 이름을 넣어줍니다.**

**String TABLE\_NAME =테이블명을 넣어 줍니다.**

**Context 데이터베이스 경로를 찾아줍니다.**

**SQLiteDAtabase= mdb로 지정해 줍니다.**

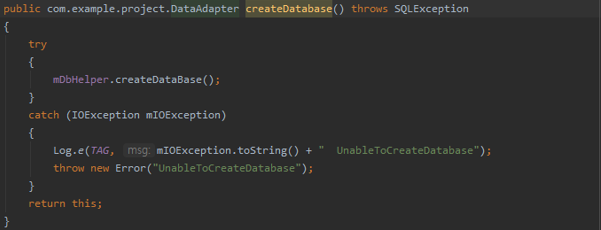
**DataBaseHelper=mdbHelper로 지정해 줍니다.**

****

**Context =데이터베이스 경로를 찾습니다.**

**DataAdapter 함수 = 클래스의 새 인스턴스를 초기화 시킵니다.**

**DataBaseHelper 함수 = 데이터베이스 생성 및 버전 관리를 합니다.**

****

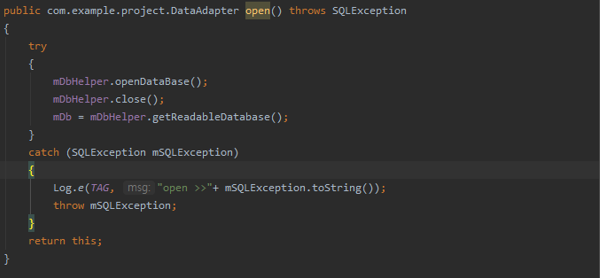
**createDatabase 함수= 서버에 데이터베이스를 만듭니다.**

**try 문을 사용해 예외 코드로 createDataBase()를 생성합니다.**

**IOException = 오류가 발생하는 경우 throw 됩니다.**

**try에서 오류가 발생하면 catch에서 UnableToCreateDatabase 문제를 발생시킵니다.**

**throw = 예외를 발생시킵니다.**

****

**Open 함수 = 데이터 베이스를 여는 함수입니다.**

**Try 블록에 mdbHelper.openDataBase(),**

**openDataBase 함수 = 열려 있는 기존 또는 새 관리자 데이터베이스를 나타내는 데이터베이스 개체입니다.**

**Close 함수 = db를 닫는 함수이다.**

**getReadableDatabase 함수 = 유를 읽기 전용으로 Open 하는 함수이다.**

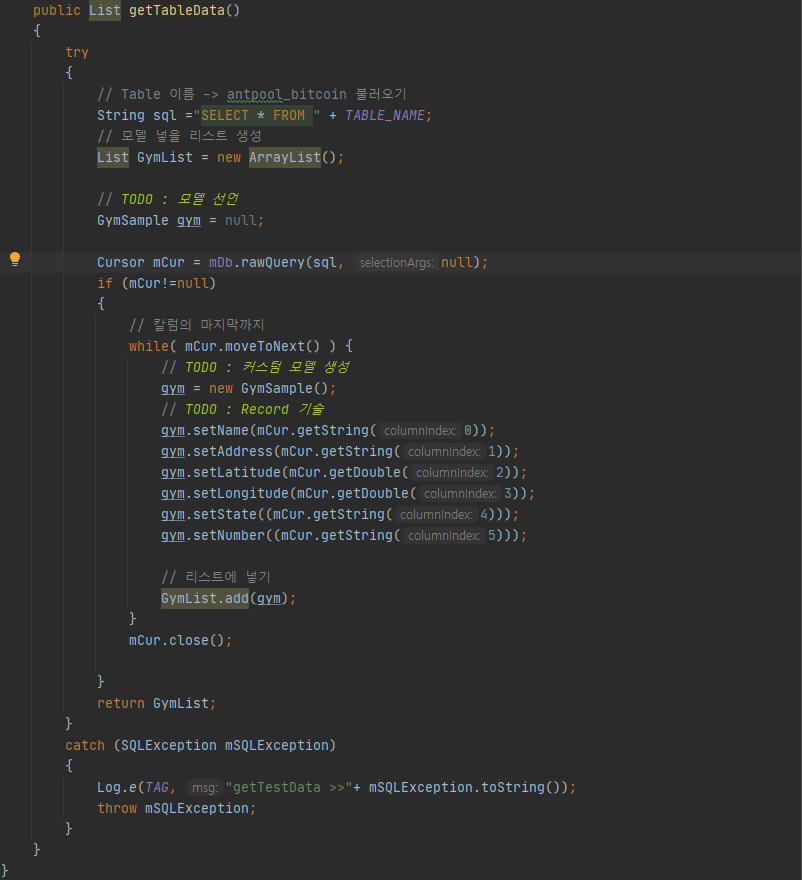
**SQLException = SQL Server에서 오류를 반환할 때 throw 되는 예외입니다.**

**catch 문에서 오류 발생 메시지를 띄워줍니다.**

**mSQLException.toString 함수 = sqlException 개체를 나타내는 문자열입니다.**

****

**Close를 선언하여 db를 닫습니다.**

****

**GetTableData 함수 = 쓰기 가능한 테이블 개체를 가져옵니다.**

**ArrayList() = 크기를 조절할 수 있는 배열입니다.**

**List GymList = new ArrayList(); = 모델을 넣을 리스트를 생성합니다.**

**Cursor = 데이터베이스에 저장되어 있는 테이블의 행을 참조하여 데이터로 가져오는 역할을 합니다.**

**If 문에서 mCur와 null이 같지 않다면 while 문을 작동시킵니다.**

**mCur.getString(), mCur.getDouble() 함수를 이용해서 gym 테이블의 실제 데이터를 가져옵니다.**

**rawQuery = 데이터베이스 결과값을 받아옵니다.**

**mCur.MoveToNext() =Cursor를 다음 행으로 이동시킨다.**

**catch 문을 이용해 오류 발생 문자를 발생시킵니다.**

**mSQLException.toString 함수 = sqlException 개체를 나타내는 문자열입니다.**

**Class DataBaseHelper**

****

**DataBaseHelper 클래스. DB의 이름과 경로를 통해 assets에 있는 DB를 사용할 것인지 등을 확인.**

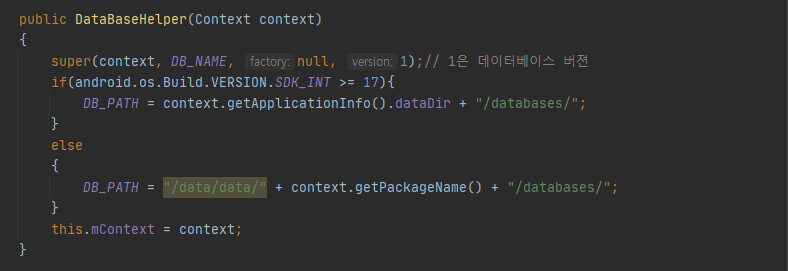
**다른 클래스에서 참조하므로 public으로 선언. SQLiteOpenHelper 클래스를 상속한다.**

**String DB\_PATH: DB 파일의 경로를 저장할 변수.**

**String DB\_NAME: DB 파일의 파일명을 저장할 변수.**

**SQLiteDatabase mDataBase: DB 파일을 저장할 변수.**

**Context mContext: DB 파일의 경로를 찾는데 사용할 변수**

****

**DataBaseHelper 함수.**

**super: 부모 클래스의 자원을 사용할 때 사용.**

**부모 클래스가 SQLiteOpenHelper이므로**

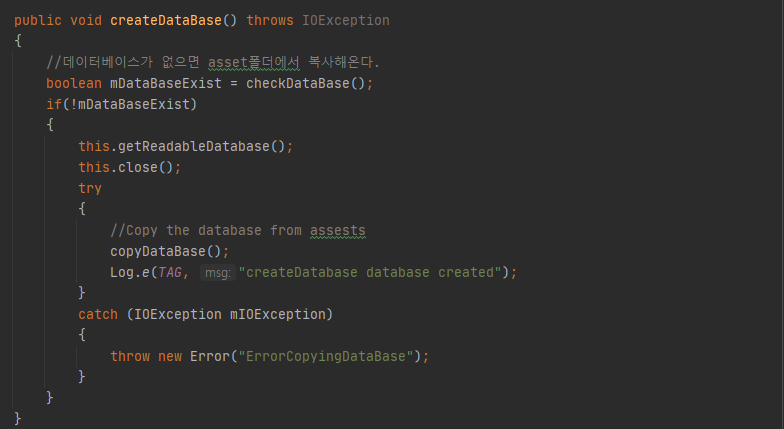
**SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version)에 맞게**

**super(context, name, factory, version) 매개변수를 맞춰서 작성.**

**SQLiteDatabase.CursorFactory: Cursor 쿼리를 호출할 때 하위 클래스를 반환하는 데 사용.**

**if 문에서 안드로이드의 버전을 체크한 다음 버전에 맞춰서 DB의 경로를 저장.**

**mContext에 경로 정보를 저장함.**

****

**createDataBase 함수.**

**주요 기능: DB가 없으면 asset 폴더에서 DB를 복사해 가져온다.**

**throws IOException: 입출력 예외 처리.**

**checkDataBase()를 boolean형의 mDataBaseExist 변수에 대입.**

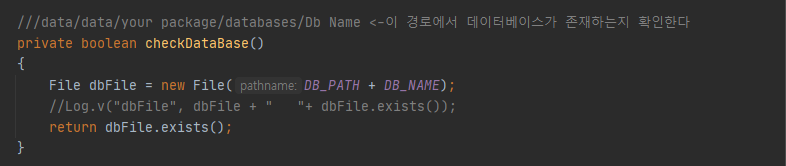
**if 문으로 mDataBaseExist가 false 인지 체크. checkDataBase()는 후술함.**

**getReadableDatabase(): DB를 생성 또는 연다.**

**try 문에서 copyDataBase()를 실행. copyDataBase()는 후술함.**

**copyDataBase()를 실행하고 DB를 생성했다는 로그를 띄움.**

**try 문에서 예외가 발생하면 catch로 넘겨서 오류 발생 메시지를 띄움.**

****

**checkDataBase 함수.**

**주요 기능: DB 파일 존재 유무 확인.**

**DB가 존재하는지 존재하지 않는지 확인을 해야 하기 때문에 boolean형으로 선언.**

**File: 파일 또는 디렉터리에 대한 정보를 제공하는 클래스.**

**File(String pathname): 주어진 문자열을 이름으로 갖는 File 인스턴스 생성.**

**exists(): 디렉터리의 존재 유무를 확인.**

**존재하면 true, 아니면 false를 return 함.**

****

**copyDataBase 함수**

**주요 기능: assets 폴더에서 DB 파일을 복사한다.**

**InputStream: 바이트 기반 입력 스트림의 최상위 클래스.**

**OutputStream: 바이트 기반 출력 스트림의 최상위 클래스.**

**Assets 폴더의 DB 파일을 불러와서 mInput에 저장.**

**outFileName에 DB 경로와 이름을 대입.**

**OutputStream 객체를 생성.**

**Byte 배열 mBuffer를 만들어서 DB의 임시 저장소로 사용.**

**mBuffer의 반환 값을 비교할 때 사용할 int형 변수 mLength 선언.**

**read(): 데이터를 읽는 메서드. 데이터를 다 읽으면 -1이 반환된다.**

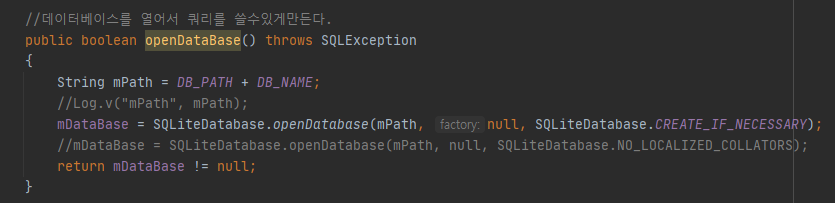
**write(): 데이터를 출력하는 메서드.**

**데이터를 읽으면서 mLength에 반환값을 대입. -1이 반환될 때까지 while 문을 반복한다.**

**mBuffer를 0부터 mLength까지 반복해서 출력.**

**flush(): 스트림의 버퍼에 있는 모든 내용을 출력.**

**close(): 입력 소스를 닫음으로써 사용하고 있던 자원 반환.**

****

**openDataBase 함수**

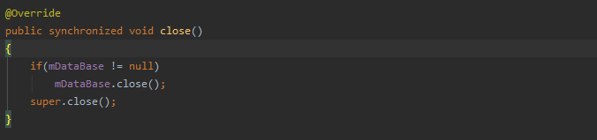
**주요 기능: DB를 열어서 쿼리를 쓸 수 있게 만든다.**

**boolean형으로 선언.**

**mPath에 DB를 저장.**

**CREATE\_IF\_NECESSARY: DB 파일이 아직 없는 경우 생성하기 위한 플래그.**

**DB를 열고 mDataBase는 비어 있는 값이 아님을 return 함.**

****

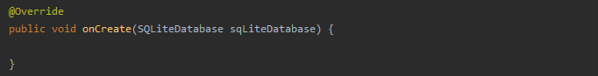
**Close 함수**

**Override로 close() 메서드를 재정의.**

**Synchronized: 다수의 스레드가 공통된 데이터에 읽고 쓰기를 할 때, 스레드 간의 간섭으로 인한 비정상적인 동작을 예방하기 위해 사용.**

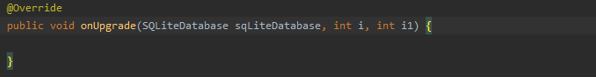
**If 문을 사용해서 mDataBase에 정보가 있을 때 mDataBase를 닫는다.**

**부모 클래스의 close() 메서드를 호출.**

****

**onCreate 함수**

**onCreate(SQLiteDatabase db): 데이터베이스가 처음 생성될 때 호출.**

****

**onUpgrade 함수**

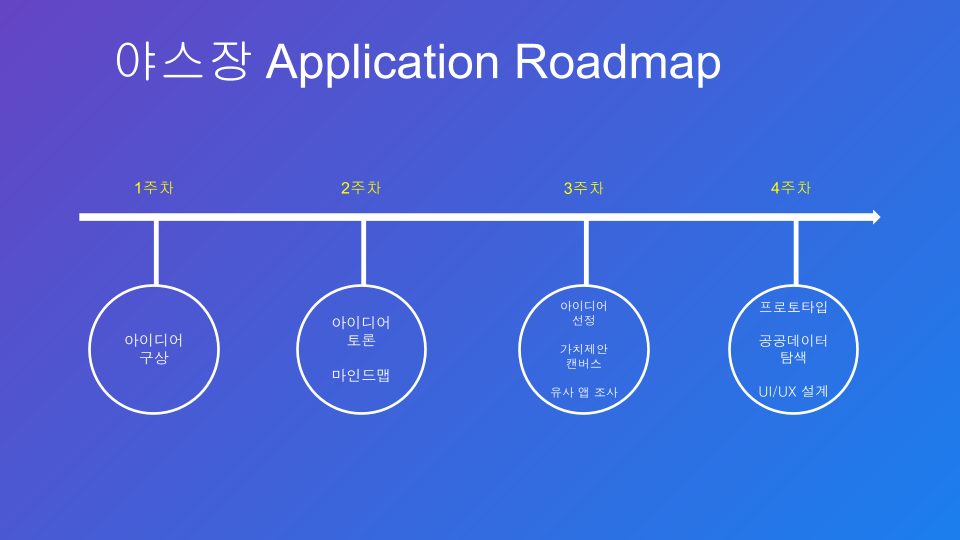
**onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion): 데이터베이스를 업그레이드해야 할 때 호출.**

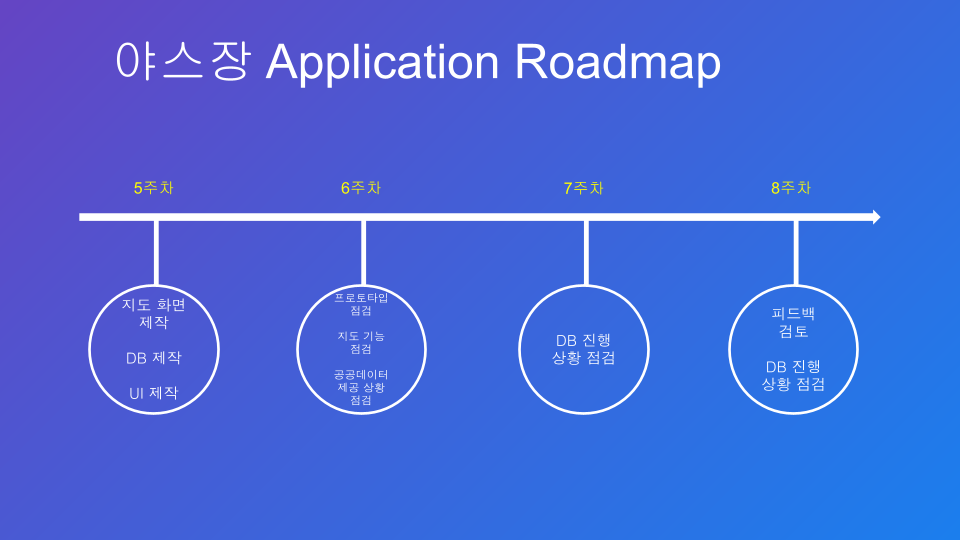
**소프트웨어 마케팅**

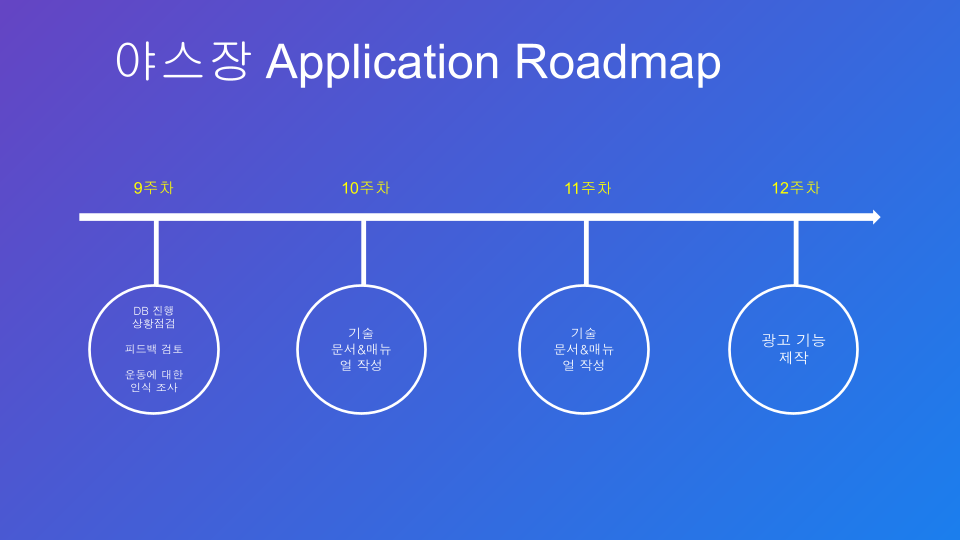
**(Marketing and Sales)**

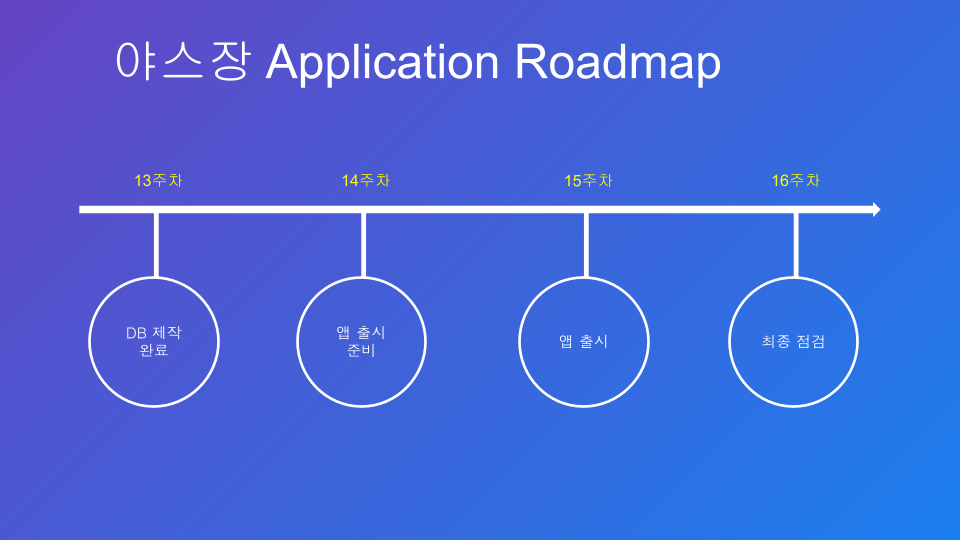
1. SNS(페이스북, 인스타그램, 카카오톡 오픈채팅방 등)를 이용해 홍보 및 소통.
2. 웹사이트, 블로그, 카페를 이용하여 홍보.
3. 에브리타임 등의 커뮤니티 사이트를 이용하여 홍보.
4. 유튜브 광고영상 제작.

**로드 맵 (Road Map)**





****

****