

폼생폼사

시스템 설계 문서

201714285 백충원

201713074 임예린

201714286 조예린

201713082 주지혜

목차

- 1.....시스템 개요
 - 1-1.....설명
 - 1-2.....기능적 요구사항
- 2.....시스템 전체 설계도
 - 2-1.....Layered architecture
 - 2-2.....프로토콜
 - 2-3.....DB Structure
- 3.....Client
 - 3-1.....클래스 다이어그램 (전체)
 - 3-2.....클래스 다이어그램 (상세)
 - 3-3.....클래스 info
- 4.....Server
 - 4-1.....클래스 다이어그램
 - 4-2.....클래스 info
- 5.....결과물
 - 5-1.....실제 실행 화면
 - 5-2.....실행 환경
 - 5-3.....실행 방식 설명

1 시스템 개요

1-1. 설명

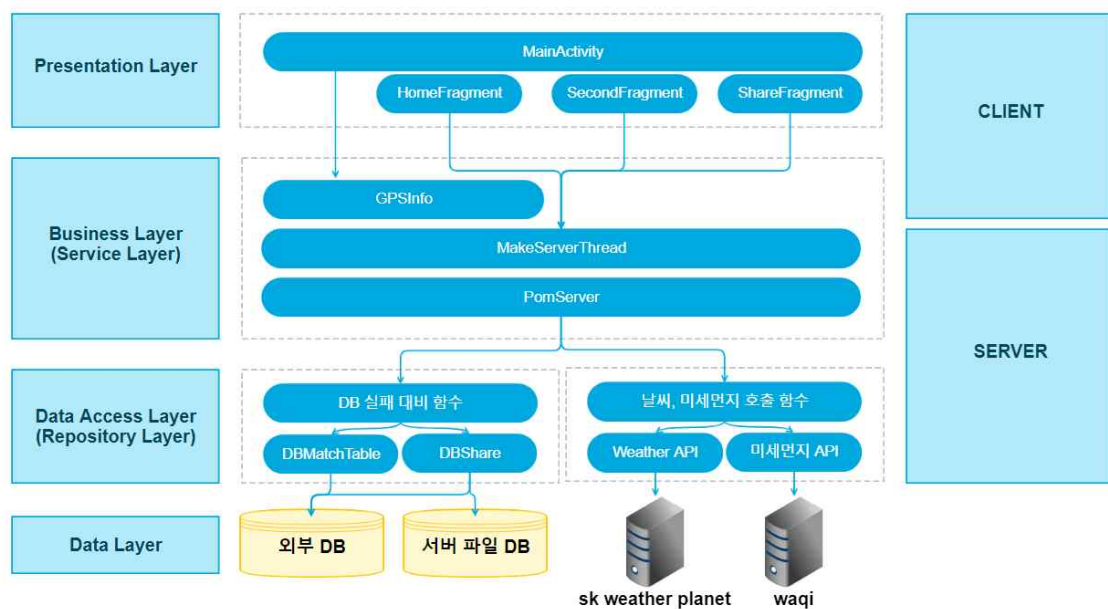
사용자의 위치에 해당하는 현재 날씨 정보와 미세먼지 정보를 활용해 우산과 마스크의 착용 여부를 알 수 있고 기온에 해당하는 적절한 옷을 추천 받고 옷 스타일링을 날씨 정보와 함께 다른 사람들과 공유할 수 있다.

1-2. 기능적 요구사항

- 사용자는 모바일 기기의 gps를 기반으로 해당 위치의 일기예보(현재 기온, 최고기온, 최저기온, 날씨정보, 강수확률)와 미세먼지 농도를 알 수 있다.
- 사용자는 상의, 하의, 외투 별로 현재 기온에 적합한 옷 추천을 받을 수 있다.
- 사용자는 상의, 하의, 외투 별로 오늘의 최고기온과 최저기온에 적합한 옷 추천을 받아 일교차에 대비할 수 있다.
- 사용자는 본인의 옷 스타일링 사진을 촬영해 날씨정보와 함께 다른 사람과 공유할 수 있다.

2. 시스템 전체 설계도

2-1. Layered architecture

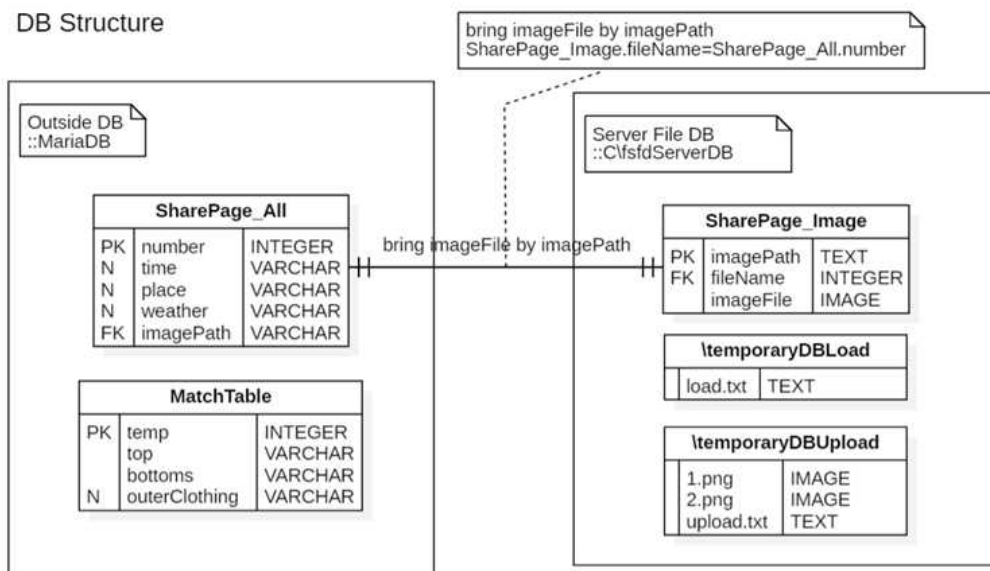


2-2. 프로토콜

-표기상의 엔터는 가독성을 위한 편의상의 엔터이다.

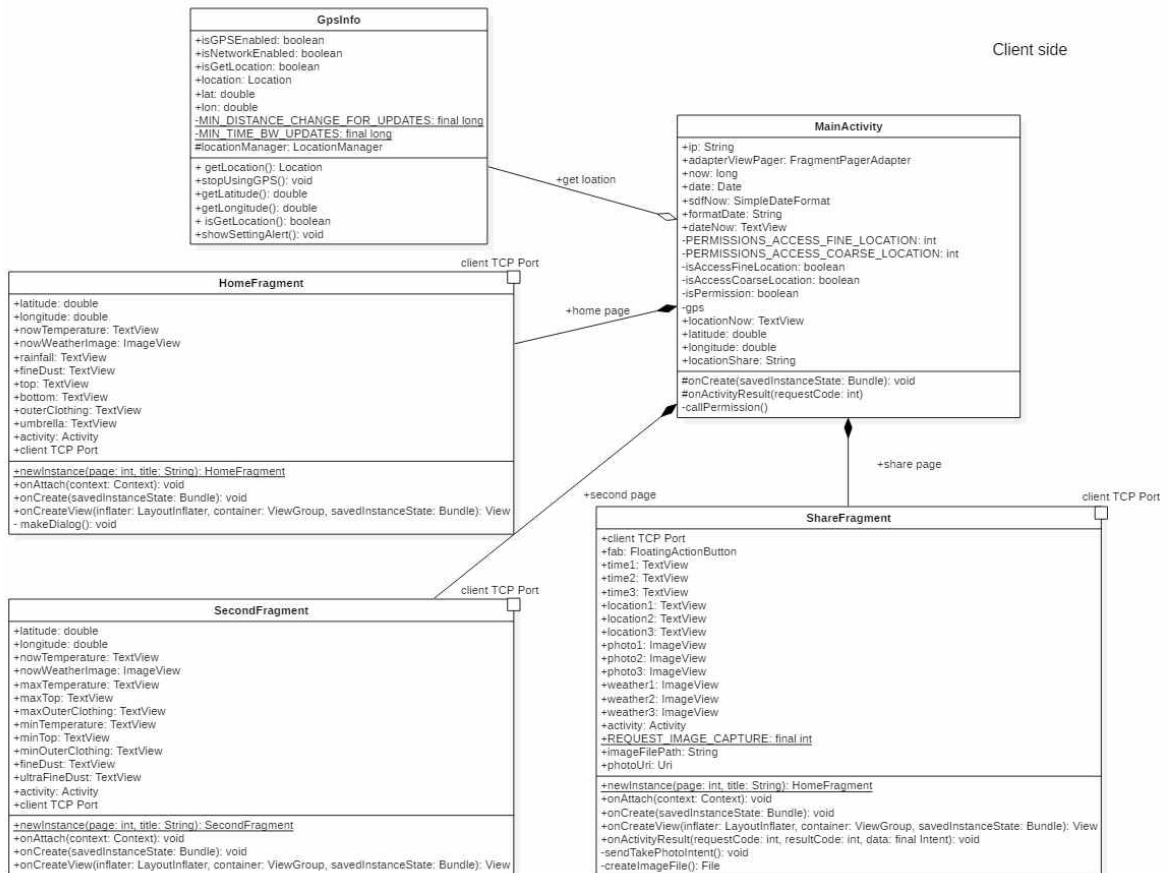
	앱 -> 서버	서버 -> 앱
홈 화면 로드	HOME:(위도):(경도);	HOME; (현재 기온):(현재 날씨); (강수량):(미세먼지); (상익):(하익):(외투);
두 번째 화면 로드	SECOND:(위도):(경도);	SECOND; (현재 기온):(현재 날씨); (최고기온):(최고_상익):(최고_외투); (최저기온):(최저_상익):(최저_외투); (미세먼지):(초미세먼지);
공유화면 로드	SHARE:R:0;	SHARE:R:(시간):(위치):(날씨); (사진);
	SHARE:R:1;	
	SHARE:R:2;	
공유 버튼 클릭	SHARE:W:(시간):(위치); (사진);	

2-3. DB Structure



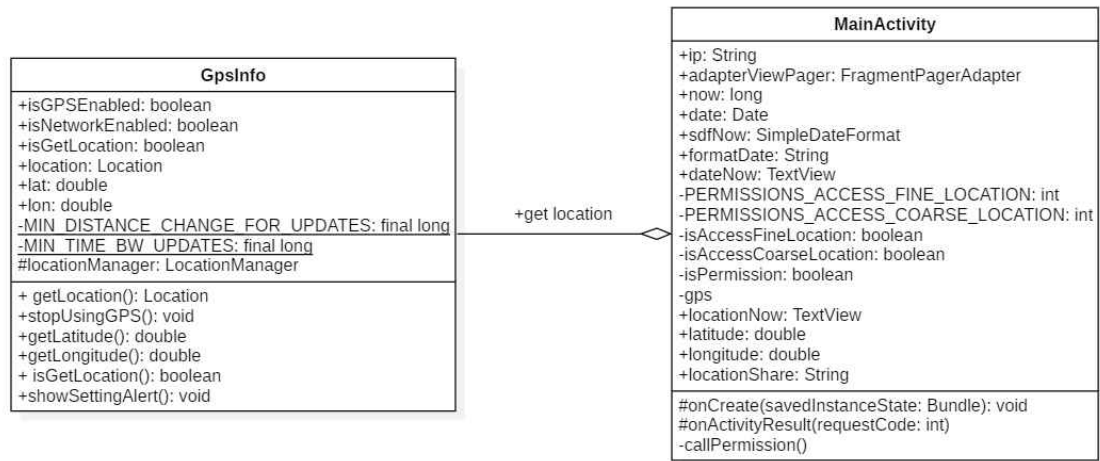
3. Client

3-1. 클래스 다이어그램 (전체)

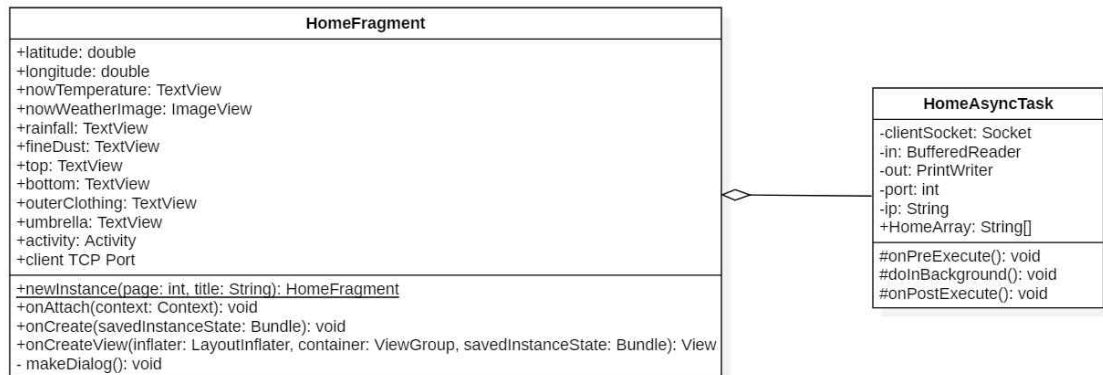


3-2. 클래스 다이어그램 (상세)

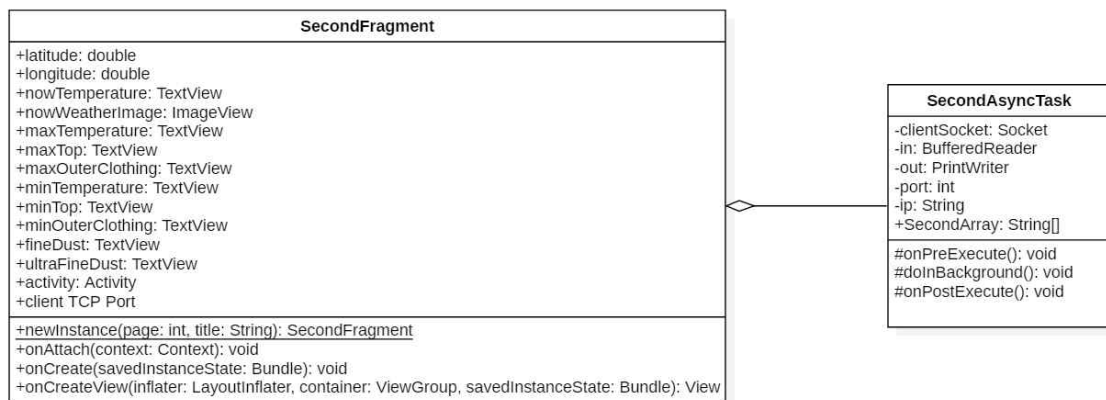
① MainActivity



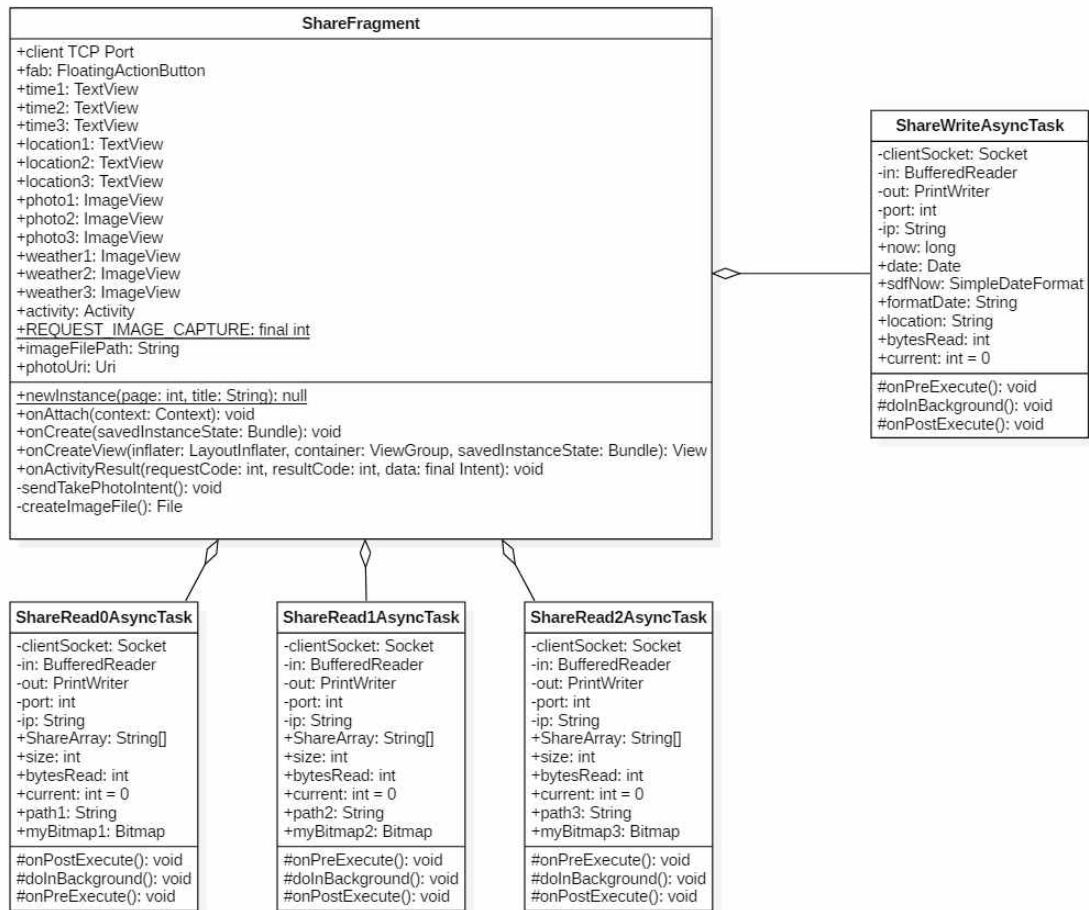
② HomeFragment



③ SecondFragment



④ ShareFragment



3-3. 클래스 info

GpsInfo	위치정보 처리
private final Context mContext boolean isGPSEnabled boolean isNetworkEnabled boolean isGetLocation Location location double lat double lon private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES private static final long MIN_TIME_BW_UPDATES protected LocationManager locationManager	안드로이드 컨텍스트 현재 GPS 사용유무 네트워크 사용유무 GPS 상태 값 위치정보 위도 경도 최소 GPS 업데이트 거리 최소 GPS 업데이트 시간 LocationManager
public Location getLocation() public void stopUsingGPS() public double getLatitude() public double getLongitude() public boolean isGetLocation() public void showSettingsAlert() private void makeDialog()	위치 정보 가져옴 Gps 종료 위도 값을 가져옴 경도 값을 가져옴 gps가 켜져 있는지 확인 makeDialog() 호출 gps 설정 팝업창

MainActivity	메인 화면. 세부화면 관리
public String ip FragmentPagerAdapter adapterViewPager long now Date date SimpleDateFormat sdfNow String formatDate TextView dateNow: private final int PERMISSIONS_ACCESS_FINE_LOCATION private final int PERMISSIONS_ACCESS_COARSE_LOCATION private boolean isPermission private GpsInfo gps TextView locationNow double latitude double longitude String locationShare	서버의 ip 주소 3개의 화면 관리자. 뷰페이저 현재 시간 현재 날짜 시간/날짜 형식 형식에 맞는 시간 값 화면에 띄울 시간/날짜 퍼미션체크 퍼미션체크 gps 권한 허가 여부 GpsInfo 객체 화면에 띄울 현재 위치 위도 경도 Share화면에서 사용할 위치
protected void onCreate(Bundle savedInstanceState) private void callPermission() protected void onActivityResult (int requestCode, int resultCode, Intent data)	메인 화면 UI 구성 gps 권한 설정 팝업 띄움 ShareFragment에서 찍은 사 진 데이터를 넘겨줌

SecondFragment	두 번째 화면
TextView nowTemperature; ImageView nowWeatherImage; TextView maxTemperature TextView maxTop TexttView maxOuterClothing TextView minTemperature TextView minTop TextView minOuterClothing TextView fineDust TextView ultraFineDust double latitude double longitude Activity activity	현재 기온 현재 날씨 오늘 최고 기온 최고기온일 때 추천 상의 최고기온일 때 추천 외투 오늘 최저기온 최저기온일 때 추천 상의 최저기온일 때 추천 외투 미세먼지 농도 초미세먼지 농도 위도 경도 MainActivity 변수 호출용
public static SecondFragment newInstance (int page, String title) public void onAttach(Context context) public void onCreate(Bundle savedInstanceState) public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)	프래그먼트 추가 액티비티에 프래그먼트 부착 프래그먼트 생성 프래그먼트 UI 구성

SecondAsyncTask	서버와의 통신을 위한 내부 클래스
private Socket clientSocket private BufferedReader in private PrintWriter out private int port private String ip = ((MainActivity) activity).ip; String[] SecondArray	클라이언트 소켓 서버로부터 메시지 수신 서버로 메시지 송신 서버 포트번호 서버 ip주소 서버로부터 받은 메시지
protected Void doInBackground(Void... voids) protected void onPostExecute(Void aVoid)	메시지 송수신, 처리 받은 메시지로 UI 구성

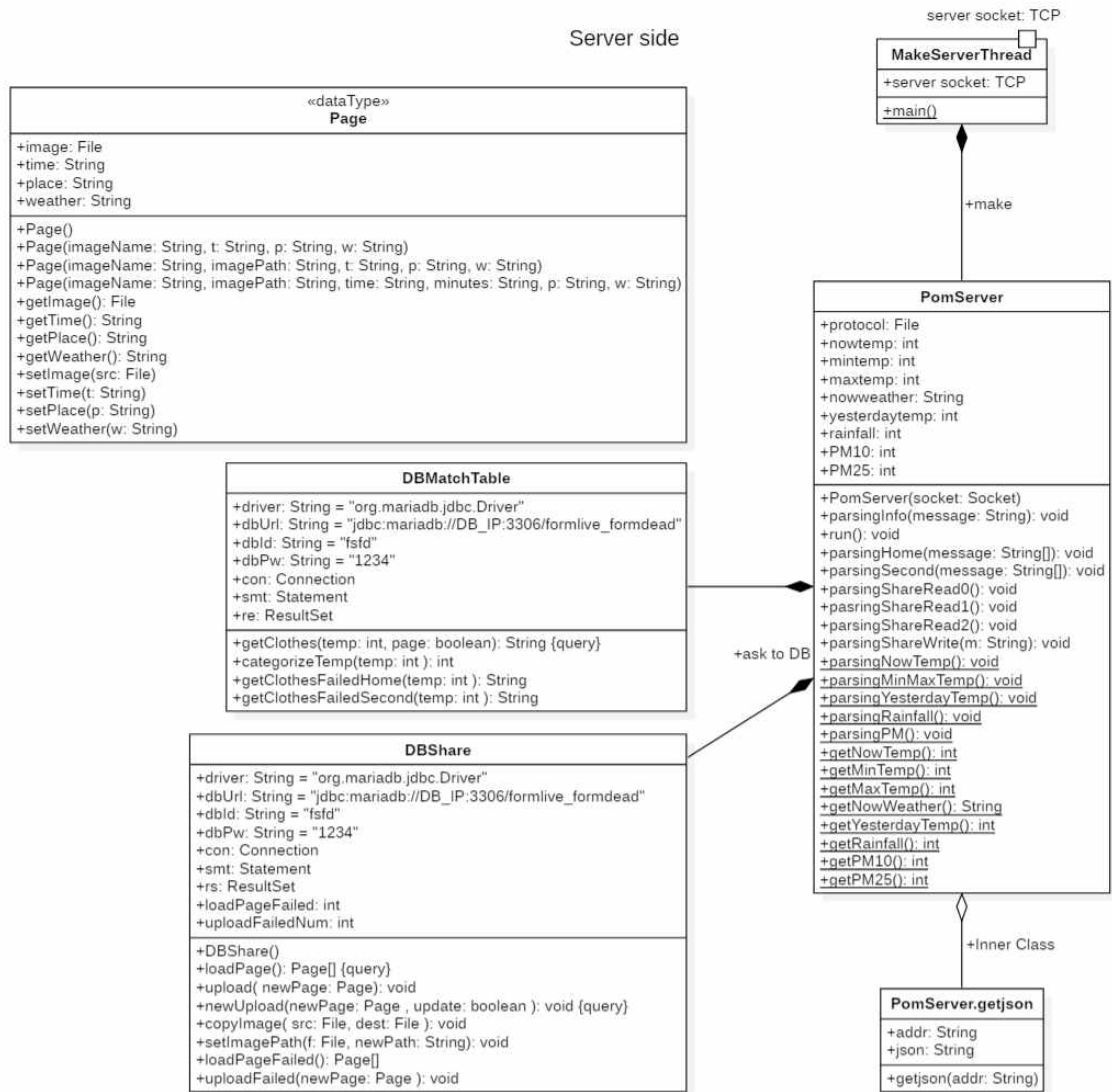
ShareFragment	공유 화면
FloatingActionButton fab TextView time1 TextView time2 TextView time3 TextView location1 TextView location2 TextView location3 ImageView photo1 ImageView photo2 ImageView photo3 ImageView weather1 ImageView weather2 ImageView weather3 Activity activity static final int REQUEST_IMAGE_CAPTURE private String imageFilePath private Uri photoUri	공유 버튼 공유 사용자1 시간(UI) 공유 사용자2 시간 공유 사용자3 시간 공유 사용자1 위치 공유 사용자2 위치 공유 사용자3 위치 공유 사용자1 사진 공유 사용자2 사진 공유 사용자3 사진 공유 사용자1 날씨 공유 사용자2 날씨 공유 사용자3 날씨 MainActivity 변수 호출용 requestCode 저장할 사진 경로 저장할 사진 Uri
public static ShareFragment newInstance (int page, String title) public void onAttach(Context context) public void onCreate(Bundle savedInstanceState) public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) private void sendTakePhotoIntent() private File createImageFile()	프래그먼트 추가 액티비티에 프래그먼트 부착 프래그먼트 생성 프래그먼트 UI 구성 카메라 화면 호출 찍은 사진 저장

ShareRead0AsyncTask (ShareRead2AsyncTask까지 모두 동일)	서버로부터 다른 사용자들의 게시물 가져오기
<pre>private Socket clientSocket private BufferedReader in private PrintWriter out private int port private String ip = ((MainActivity) activity).ip; String[] ShareArray int size int bytesRead int current = 0 String path1 Bitmap myBitmap1</pre>	클라이언트 소켓 서버로부터 메시지 수신 서버로 메시지 송신 서버 포트번호 서버 ip주소 서버로부터 받은 메시지 사진 크기 read로 읽은 바이트 수 현재까지 읽은 바이트 수 사용자 사진1을 저장할 경로 사용자 사진1 (UI)
<pre>protected Void doInBackground(Void... voids) protected void onPostExecute(Void aVoid)</pre>	메시지 송수신, 처리 받은 메시지로 UI 구성

ShareWriteTask	새 게시물 등록하기
<pre>private Socket clientSocket private BufferedReader in private PrintWriter out private int port private String ip = ((MainActivity) activity).ip; String[] ShareArray long now Date date SimpleDateFormat sdfNow String formatDate String location int bytesRead; int current = 0;</pre>	클라이언트 소켓 서버로부터 메시지 수신 서버로 메시지 송신 서버 포트번호 서버 ip주소 서버로부터 받은 메시지 현재 시간 현재 날짜 시간/날짜 형식 형식에 맞는 시간 값 현재 사용자 위치 read로 읽은 바이트 수 현재까지 읽은 바이트 수
<pre>protected Void doInBackground(Void... voids) protected void onPostExecute(Void aVoid)</pre>	메시지 송수신, 처리 받은 메시지로 UI 구성

4. Server

4-1. 클래스 다이어그램



4-2. 클래스 info

MakeServerThread	클라이언트 당 스레드를 생성, 할당
static void main (String[] args)	메인. 클라이언트 접속 시 스레드 생성

PomServer	정보 식별, 필요한 함수 호출. 서버 메인 기능
static String latitude static String lonitude private Socket socket BufferedReader bufferedReader PrintWriter printWriter int bytesRead int current	client의 현재 위도 (API 호출에 사용) client의 현재 경도 (API 호출에 사용) TCP 서버 소켓 client로부터 메시지 수신 client로 메시지 송신 읽은 바이트 수 (이미지 송신에 사용) 총 읽은 바이트 수 (이미지 송신에 사용)
public void run() void parsingInfo(String message) void parsingHome(String[] message) void parsingSecond(String[] message) void parsingShareRead0() void parsingShareRead1() void parsingShareRead2() void parsingShareWrite(String[] m)	client로부터 메시지 수신 client로부터 받은 메시지 파싱 Home 화면에 출력될 날씨, 옷 값 Second 화면에 출력될 날씨, 옷 값 첫 공유화면에 출력될 사진, 날씨 두 번째 공유화면에 출력될 사진, 날씨 세 번째 공유화면에 출력될 사진, 날씨 공유화면 이미지 파일,위치 받기

<<Data type>> Page	공유화면 페이지 구조체 클래스
Image image String time String place String weather	
Image getImage() String getTime() String getPlace() String getWeather() void setImage(Image img) void setTime(String t) void setPlace(String p) void setWeather(String w)	변수 image를 리턴 변수 time를 리턴 변수 place를 리턴 변수 weather를 리턴 변수 image에 값을 입력 변수 time에 값을 입력 변수 place에 값을 입력 변수 weather에 값을 입력

DBMatchTable	홈 화면 옷 매치테이블 관련
String driver String dbUrl String dbId String dbPw Connection con Statement smt ResultSet rs	드라이버 주소 DB 주소 DB ID DB 패스워드 연결 변수 상태 변수 결과 저장
String getClothes(int temp, boolean page) int categorizeTemp(int temp) String getClothesFailedHome(int temp) String getClothesFailedSecond(int temp)	매치테이블 정보 가져옴 (page 변수는 home과 second를 구분하기 위함) 온도 값을 매치테이블 정보에 맞게 분류한 후 리턴 실패 대비 함수 - 홈 화면 실패 대비 함수 - 두 번째 화면

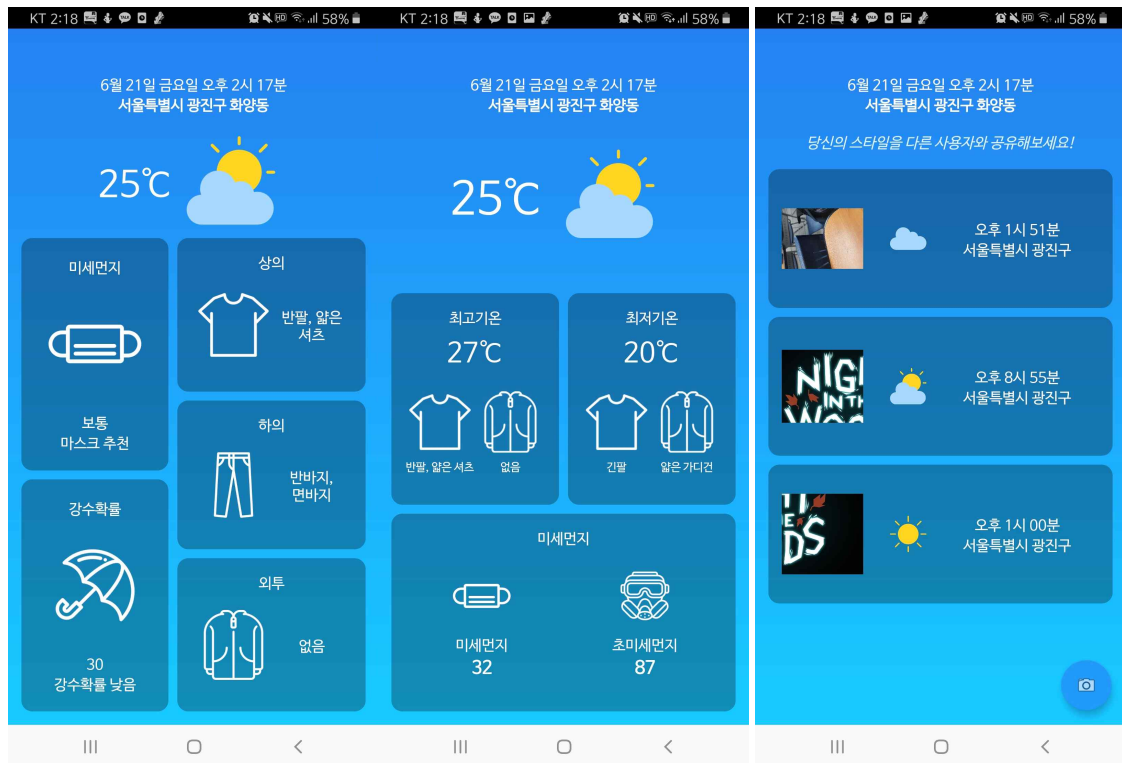
DBShare	공유 페이지 관련
String driver	드라이버 주소
String dbUrl	DB 주소
String dbId	DB ID
String dbPw	DB 패스워드
Connection con	연결 변수
Statement smt	상태 변수
ResultSet rs	결과 저장
int loadPageFailed	loadPage 함수 실패 시 지정하는 변수
int uploadFailedNum	업로드 실패 횟수
DBShare()	생성자
Page[] loadPage()	공유화면 페이지(최근 3개) 로드
void upload(Page newPage)	실패 대비 함수 (업로드 실패 횟수에 따라 일처리 다름)
void newUpload (Page newPage, boolean update)	업로드 함수
void copyImage(File src, File dest)	이미지를 src에서 dest로 복사
void setImagePath(File f, String newPath)	이미지 경로를 지정한다.
Page[] loadPageFailed()	페이지 로드 실패 대비 함수
void uploadFailed(Page newPage)	업로드 실패 대비 함수

WeatherAPI	날씨 관련(sk weather planet api 사용)
int nowtemp String nowweather int mintemp int maxtemp int yesterdaytemp int rainfall	현재 기온 (parsingNowTemperature()) 현재 날씨 요약 (parsingNowWeather()) 오늘의 최저 기온(parsingMinTemperature ()) 오늘의 최고 기온(parsingMaxTemperature ()) 어제의 기온 (parsingYesterdayWeather()) 현재 강수확률 (parsingRainfall())
void getjson() void parsingNowTemp() void parsingNowWeather() void parsingMinMaxTemp() void parsingYesterdayTemp() void parsingRainfall() int getNowTemp() int getMinTemp() int getMaxTemp() String getNowWeather() int getYesterdayTemp() int getRainfall()	api로부터 json 가져옴 현재 기온 값 파싱 현재 날씨 요약 값 파싱 (ex. 맑음, 흐림, 비 등) 오늘 최저, 최고기온 파싱 어제 기온 파싱 강수확률 파싱 현재 기온 값 리턴 오늘 최저 기온 리턴 오늘 최고 기온 리턴 현재 날씨 요약 값 리턴 어제 기온 리턴 강수확률 리턴

WeatherWQ_API	미세먼지 관련(waqi api 사용)
int PM10 int PM25	현재 미세먼지 농도 현재 초미세먼지 농도
void getjson() void parsingPM() int getPM10() int getPM25()	api로부터 json 가져옴 현재 미세먼지 & 초미세먼지 농도 파싱 현재 미세먼지 농도 리턴 현재 초미세먼지 농도 리턴

5 결과물

5-1. 실제 실행 화면



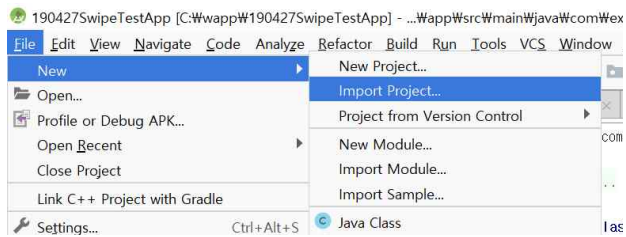
5-2. 실행 환경

- ① eclipse
- ② 안드로이드 스튜디오
- ③ 안드로이드 스마트폰
- ④ Maria DB
- ⑤ Maria DB Connector

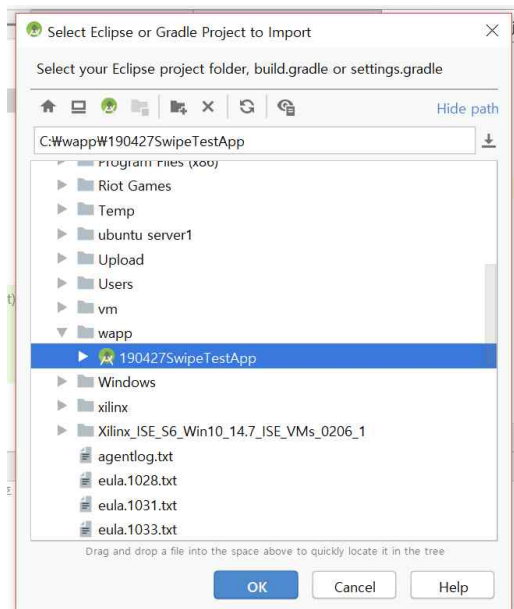
5-3. 실행 방식 설명

<CLIENT>

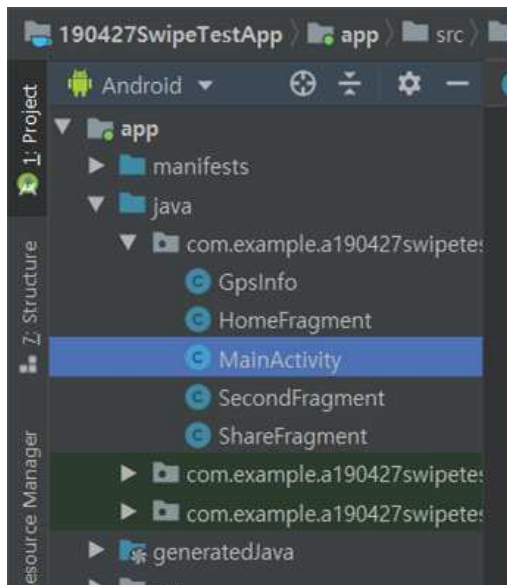
1. App.zip 압축 풀기
2. 안드로이드 스튜디오 실행
3. File - New - Import Project



4. 다운받은 안드로이드 스튜디오파일 경로 선택



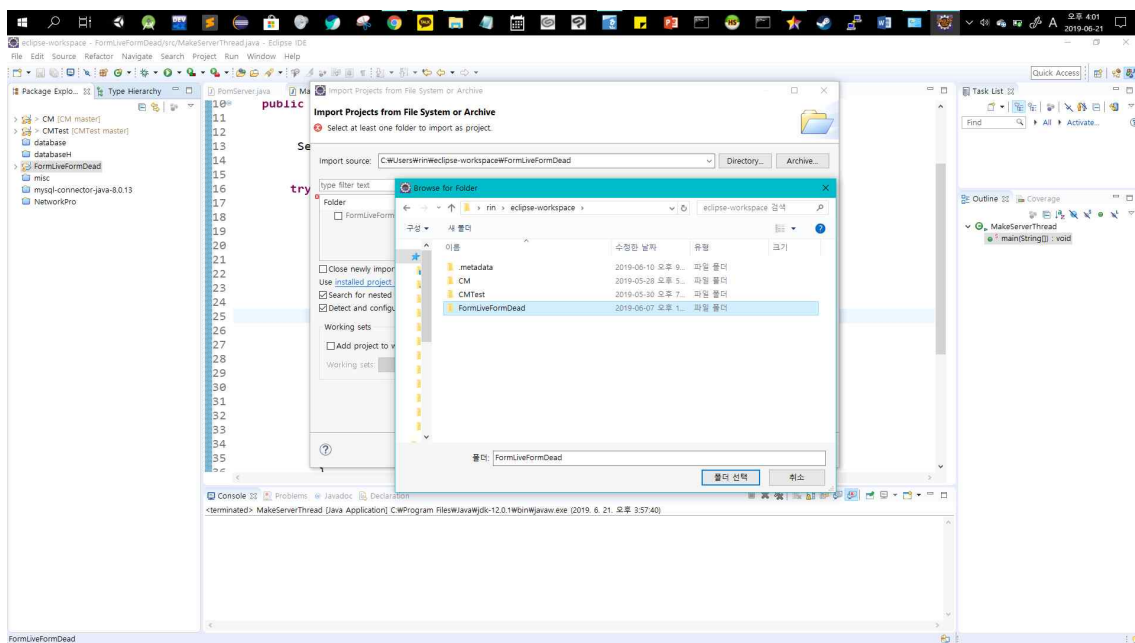
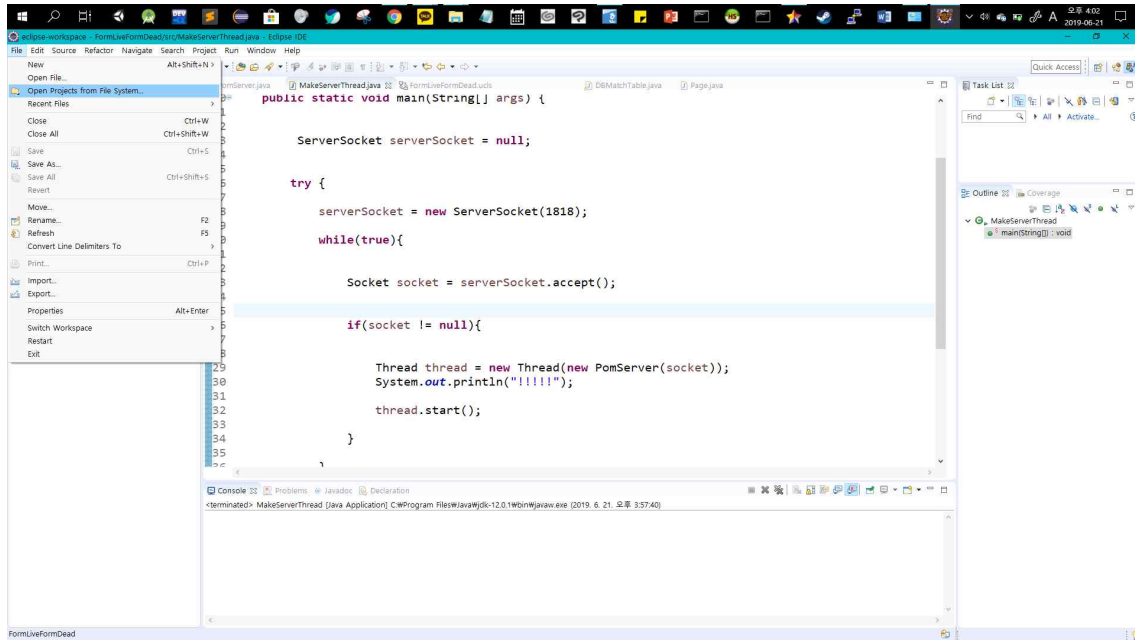
5. MainActivity.java 28번째 줄에서 ip주소를 서버 ip주소로 변경



```
23  
24 import me.relex.circleindicator.CircleIndicator;  
25  
26 public class MainActivity extends AppCompatActivity {  
27       
28     public String ip = "192,168,22,180"; //190612 TEST  
29  
30     // 뷰 페이지  
31     FragmentPagerAdapter adapterViewPager;  
32
```

<SERVER>

1. FormLiveFormDead.zip 파일 이클립스 workspace 경로에 압축해제
2. File-Open Projects from File System-폼생폼사 파일 선택



3. MariaDB 커넥터 삽입 (<https://thisisspear.tistory.com/4> 참고)
4. DBMatchTable과 DBShare 클래스의 ip주소(127.0.0.1)를 데이터베이스 서버(팀장 임예린 개인 노트북)의 ip주소로 변경

```

class DBMatchTable{

    String driver = "org.mariadb.jdbc.Driver";
    //DB=로컬호스트로 지정. 코드 이전 시 IP주소 바꿀 것
    String dbUrl="jdbc:mariadb://127.0.0.1:3306/formlive_formdead";
    String dbId="fsfd";
    String dbPw="1234";
    Connection con=null;
    Statement smt=null;
    //PreparedStatement pstmt=null;
    ResultSet rs=null;

class DBShare{

    //String IMAGEPATH="";

    String driver = "org.mariadb.jdbc.Driver";
    //DB=로컬호스트로 지정. 코드 이전 시 IP주소 바꿀 것
    String dbUrl="jdbc:mariadb://127.0.0.1:3306/formlive_formdead";
    String dbId="fsfd";
    String dbPw="1234";
    Connection con=null;
    Statement smt=null;
    //PreparedStatement pstmt=null;
    ResultSet rs=null;
    int loadPageFailed;
    int uploadFailedNum;

```