실시간 날씨 기반 스타일 추천 앱 그 그 그

백충원 임예린 조예린 주지혜

# CONTENTS

- 1 OVERVIEW
- 2 SYSTEM DESIGN
- 3 CLASS DIAGRAM

## **OVERVIEW**











공유 화면



#### 현재 사용자의 위치 정보 스마트폰 GPS권한 사용

#### 미세먼지 정보

WHO 기준에 따른 마스크 추천 필요 없음 / 마스크 추천 / KF80 이상 권장 / KF90 이상 권장

#### 우산 필요 여부 4단계

필요 없음 / 강수확률 낮음 / 우산 소지 추천 / 우산 필수



#### 현재 기온 및 날씨

SK weather planet API 사용

#### 실시간 날씨 기준 스타일 추천

상의 / 하의 / 외투 각각 추천 후드 / 기모 후드처럼 두께감도 함께 표기 매치 테이블(DB) 사용 (MariaDB)



#### 현재 기온 및 날씨

SK weather planet API 사용

## 오늘의 최고기온/최저기온 최고/최저기온일 때의 옷차림을 각각 추천

일교차 대비 가능

#### 미세먼지 정보

미세먼지 수치와 초미세먼지 수치를 함께 표기 waqi API 사용



#### 날씨에 맞는 스타일 공유 게시물

업로드 시간과 사용자의 위치, 날씨를 사진과 함께 표기 다른 사용자의 옷 선택에 도움 ShareReadnAsyncTask는각각 n번째 게시물을 서로 asynchronous방식으로 수신

#### 공유 버튼

사용자의 스타일을 사진으로 업로드 버튼 클릭 시 카메라 화면으로 이동 ShareWriteAsyncTask에서 현재 시간과 사용자의 위치와 함께 데이터를 서버로 전송

## OVERVIEW - 폼생폼사 어플 개발 환경



## OVERVIEW - 폼생폼사 서버 특징



#### 독립적인 기능

기능의 완전한 분리 기능 간 낮은 의존성으로 재활용 용이한 코드

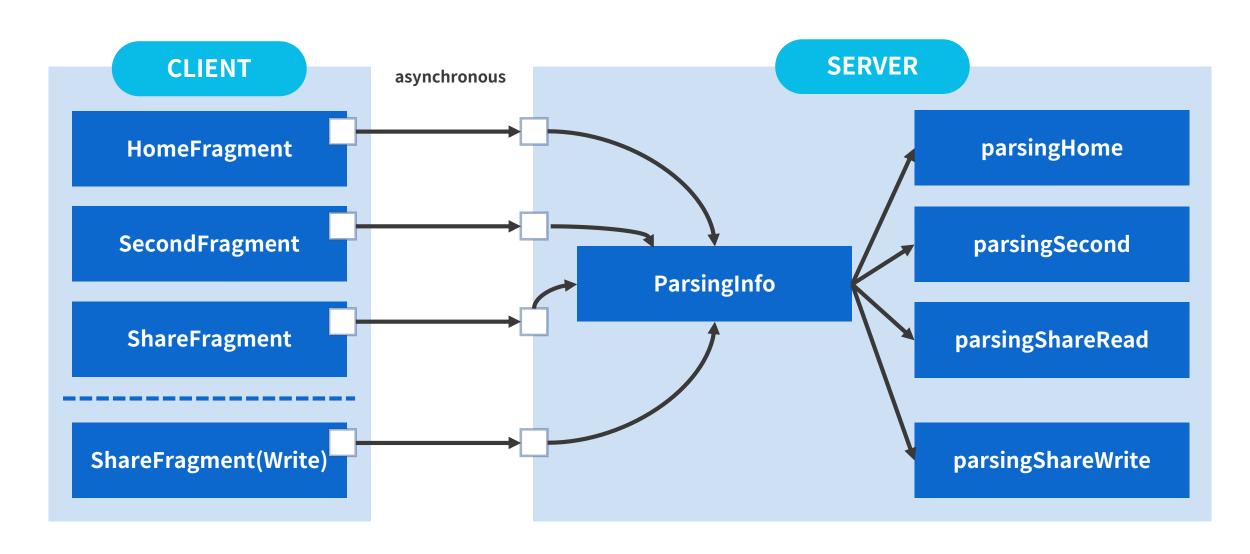
### 수업에서 배운 내용을 충실하게 사용한 JAVA SERVER

클라이언트가 보내는 메시지 당 하나의 스레드 생성 비동기식 통신 지원

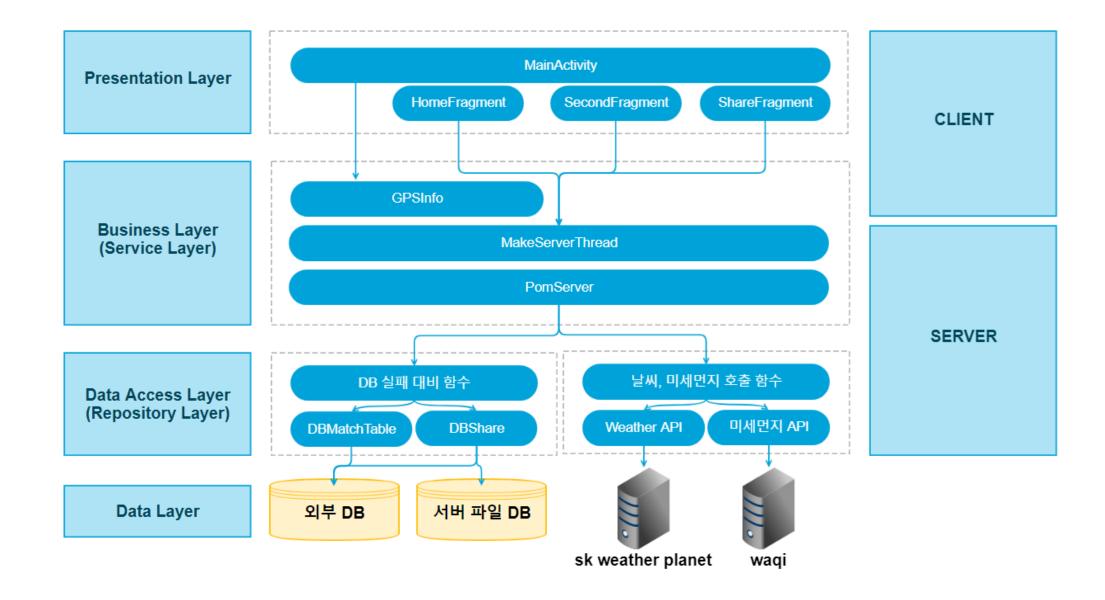
### 날씨, 미세먼지 정보 및 DB 지원

날씨 API 사용 MariaDB 및 서버 파일 DB 사용

## SYSTEM DESIGN



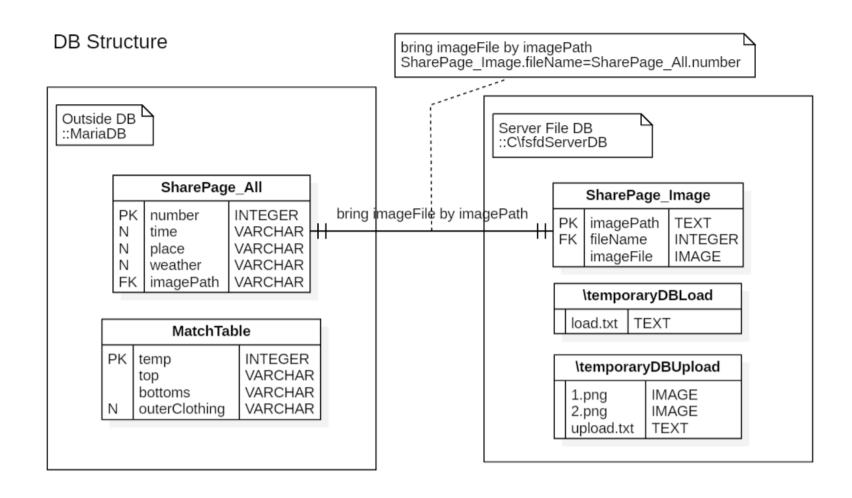
## **SYSTEM DESIGN** - Layered Architecture

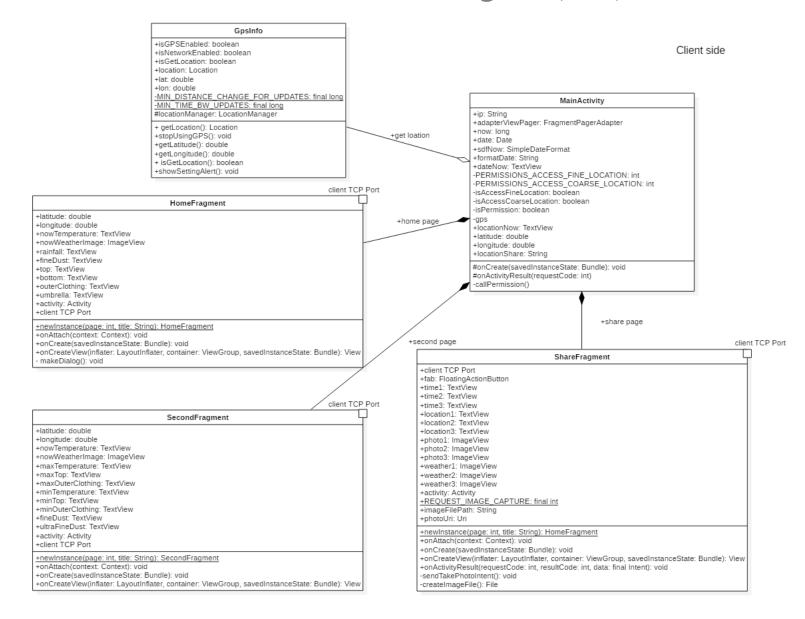


## SYSTEM DESIGN - Protocol

	앱 -> 서버	서버 -> 앱
홈 화면 로드	HOME;(위도);(경도);	HOME; (현재 기온);(현재 날씨); (강수량);(미세먼지); (상의);(하의);(외투);
두 번째 화면 로드	SECOND;(위도);(경도);	SECOND; (현재 기온);(현재 날씨); (최고기온);(최고_상의);(최고_외투); (최저기온);(최저_상의);(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);	

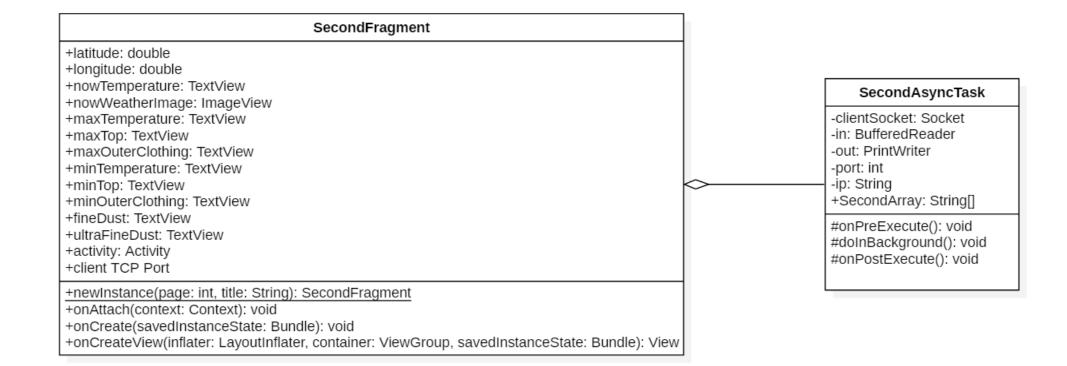
## **SYSTEM DESIGN** - DB structure

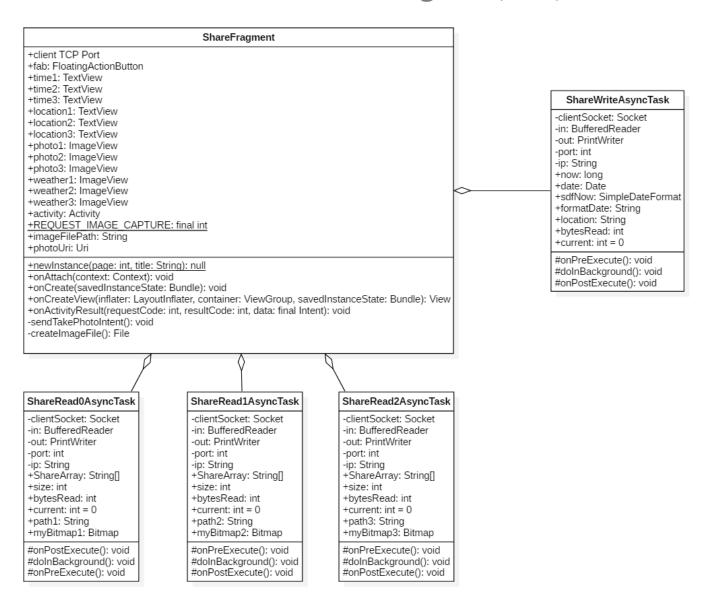




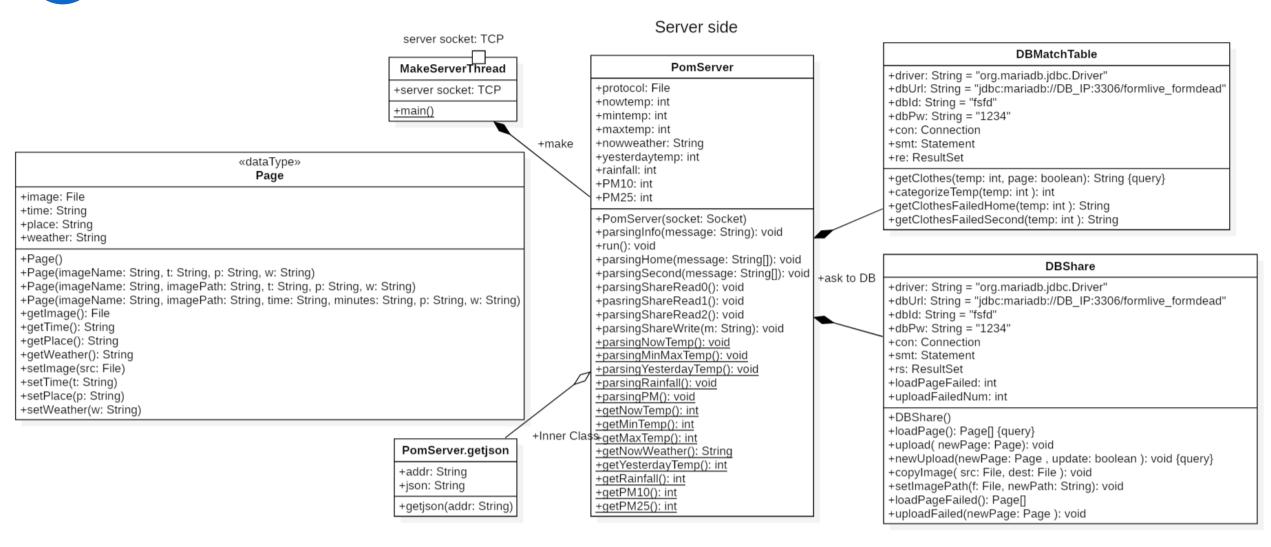
#### MainActivity +ip: String +adapterViewPager: FragmentPagerAdapter GpsInfo +now: long +isGPSEnabled: boolean +date: Date +isNetworkEnabled: boolean +sdfNow: SimpleDateFormat +isGetLocation: boolean +formatDate: String +location: Location +dateNow: TextView +lat: double -PERMISSIONS ACCESS FINE LOCATION: int +lon: double +get location -PERMISSIONS ACCESS COARSE LOCATION: int -MIN DISTANCE CHANGE FOR UPDATES: final long -isAccessFineLocation: boolean -MIN TIME BW UPDATES: final long -isAccessCoarseLocation: boolean #locationManager: LocationManager -isPermission: boolean + getLocation(): Location +locationNow: TextView +stopUsingGPS(): void +latitude: double +getLatitude(): double +longitude: double +getLongitude(): double +locationShare: String + isGetLocation(): boolean +showSettingAlert(): void #onCreate(savedInstanceState: Bundle): void #onActivityResult(requestCode: int) -callPermission()

#### HomeFragment +latitude: double +longitude: double +nowTemperature: TextView HomeAsyncTask +nowWeatherImage: ImageView +rainfall: TextView -clientSocket: Socket +fineDust: TextView -in: BufferedReader +top: TextView -out: PrintWriter +bottom: TextView -port: int +outerClothing: TextView -ip: String +umbrella: TextView +HomeArray: String[] +activity: Activity #onPreExecute(): void +client TCP Port #doInBackground(): void +newInstance(page: int, title: String): HomeFragment #onPostExecute(): void +onAttach(context: Context): void +onCreate(savedInstanceState: Bundle): void +onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): View - makeDialog(): void





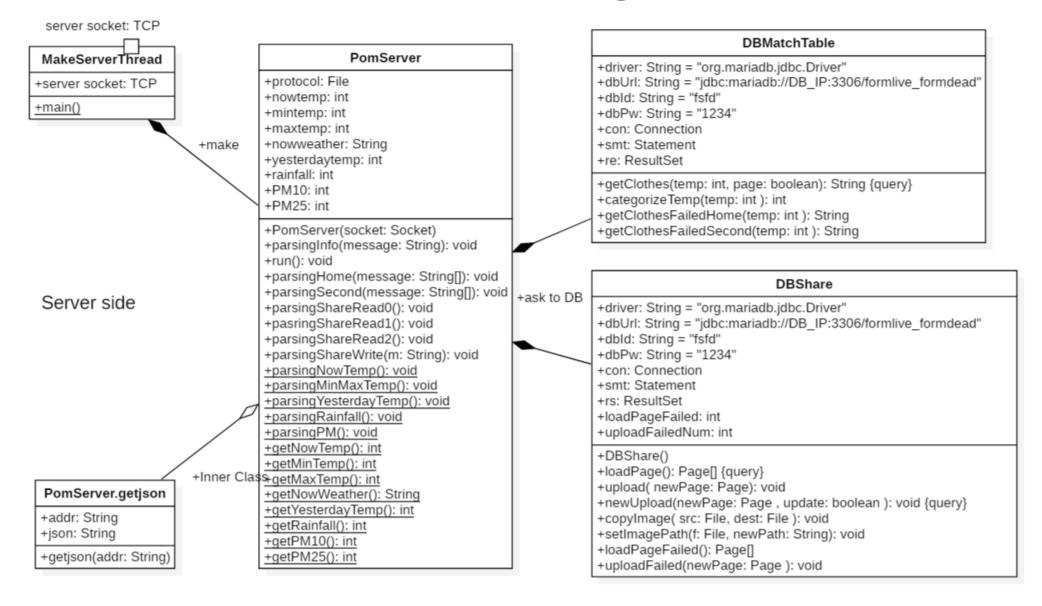
## CLASS DIAGRAM – Server side class diagram (전체)



## CLASS DIAGRAM - Server side class diagram (부분)

```
«dataType»
                                            Page
+image: File
+time: String
+place: String
+weather: String
+Page()
+Page(imageName: String, t: String, p: String, w: String)
+Page(imageName: String, imagePath: String, t: String, p: String, w: String)
+Page(imageName: String, imagePath: String, time: String, minutes: String, p: String, w: String)
+getImage(): File
+getTime(): String
+getPlace(): String
+getWeather(): String
+setImage(src: File)
+setTime(t: String)
+setPlace(p: String)
+setWeather(w: String)
```

## CLASS DIAGRAM – Server side class diagram (부분)



# THANK YQU