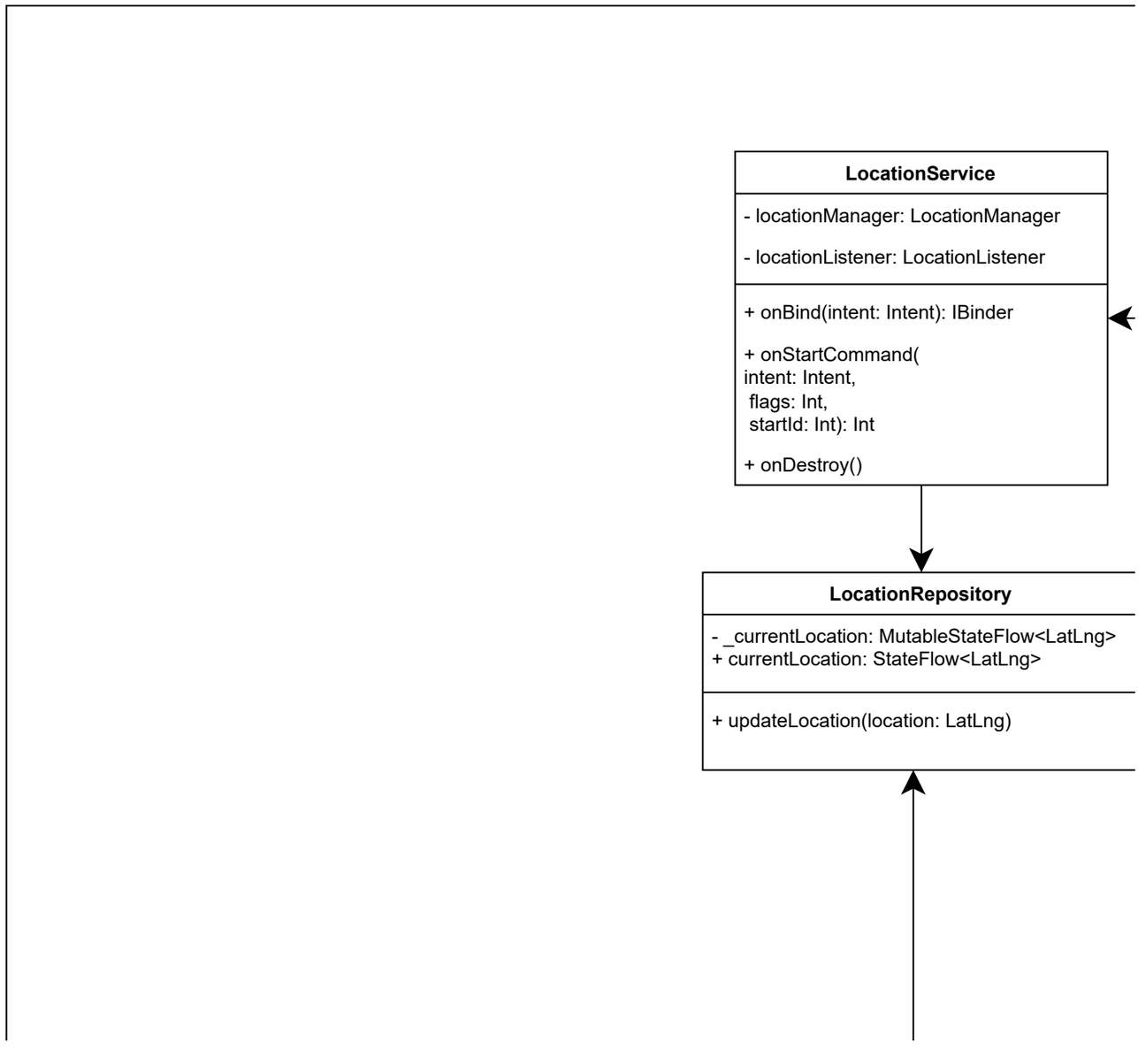
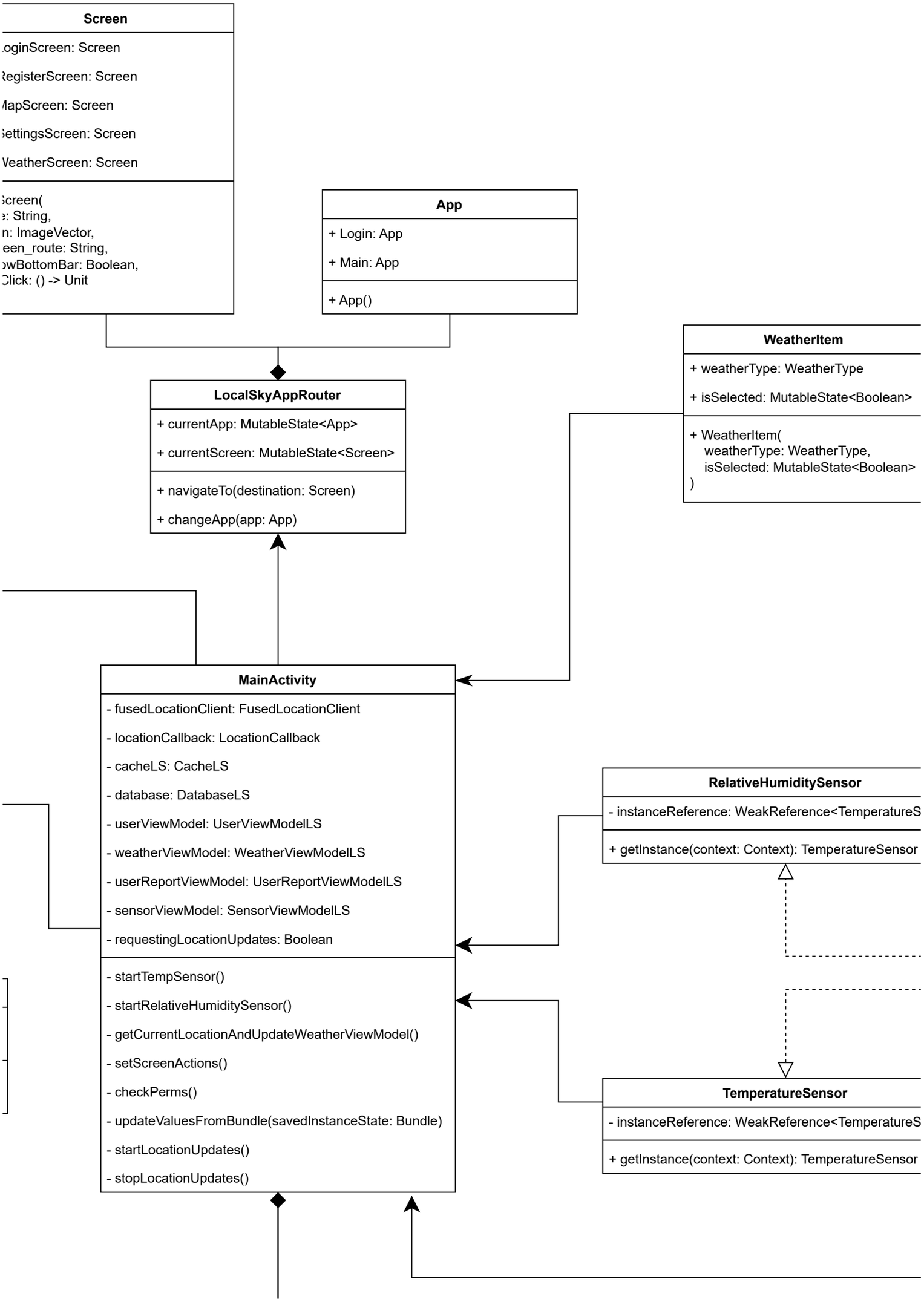
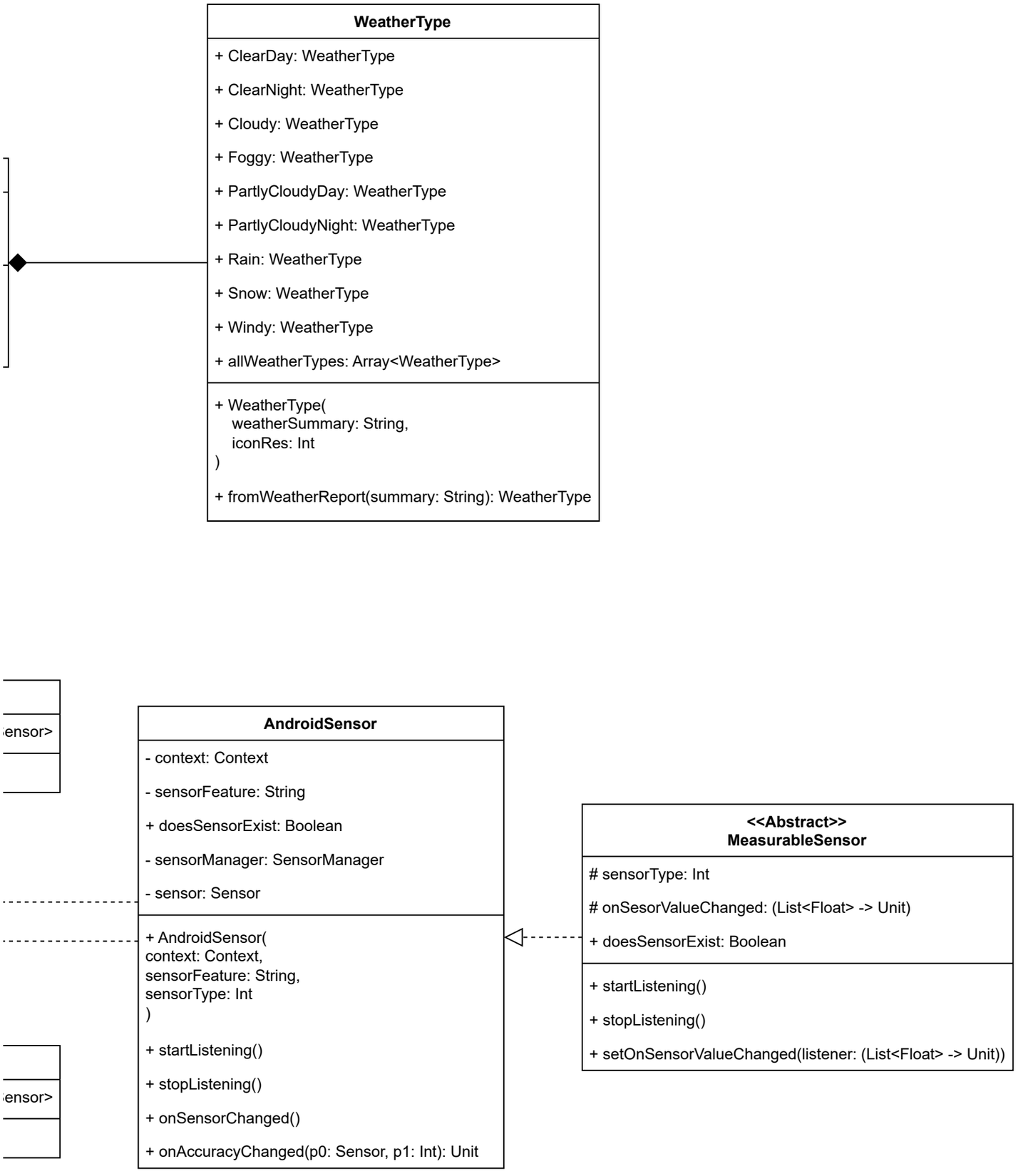
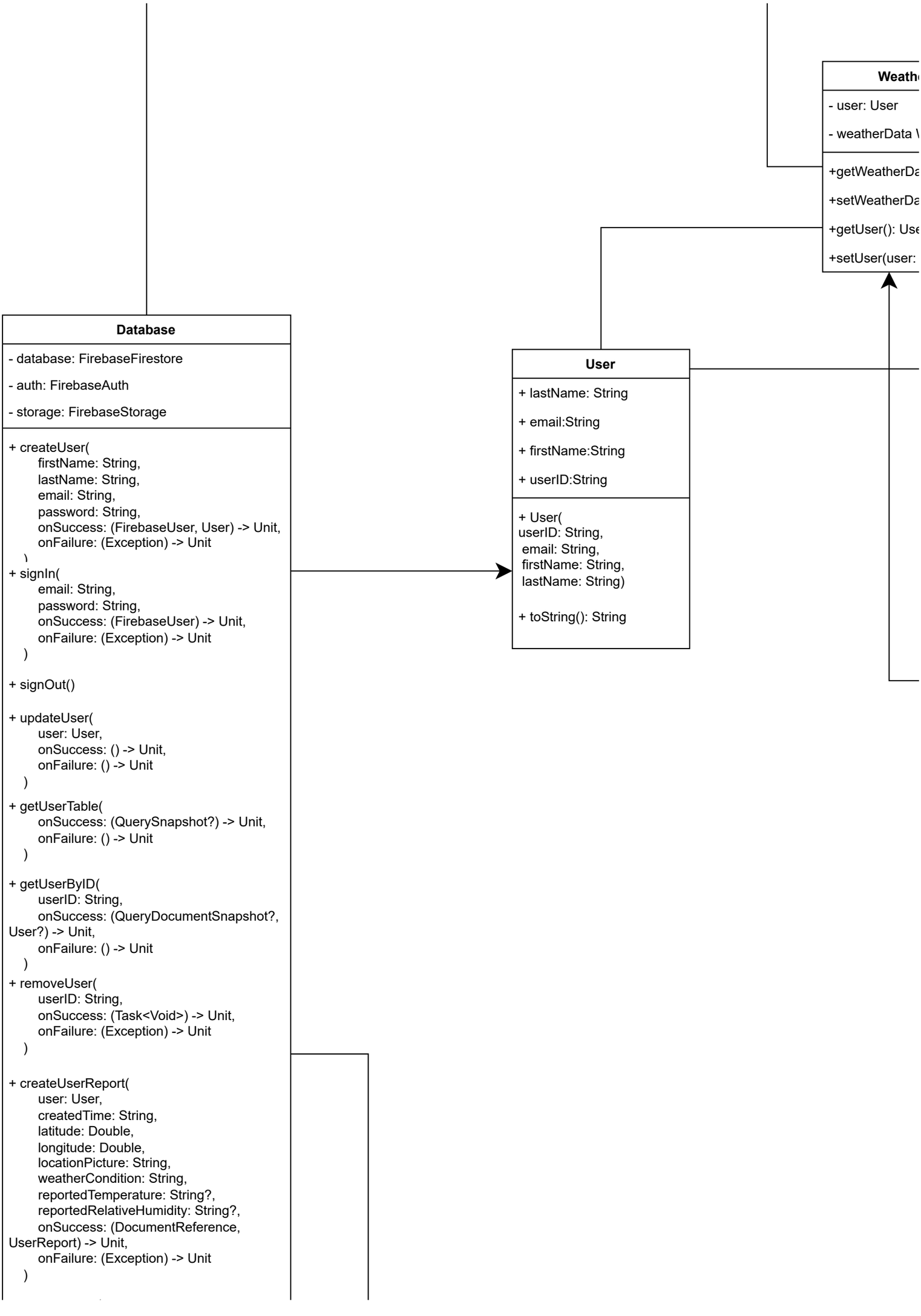
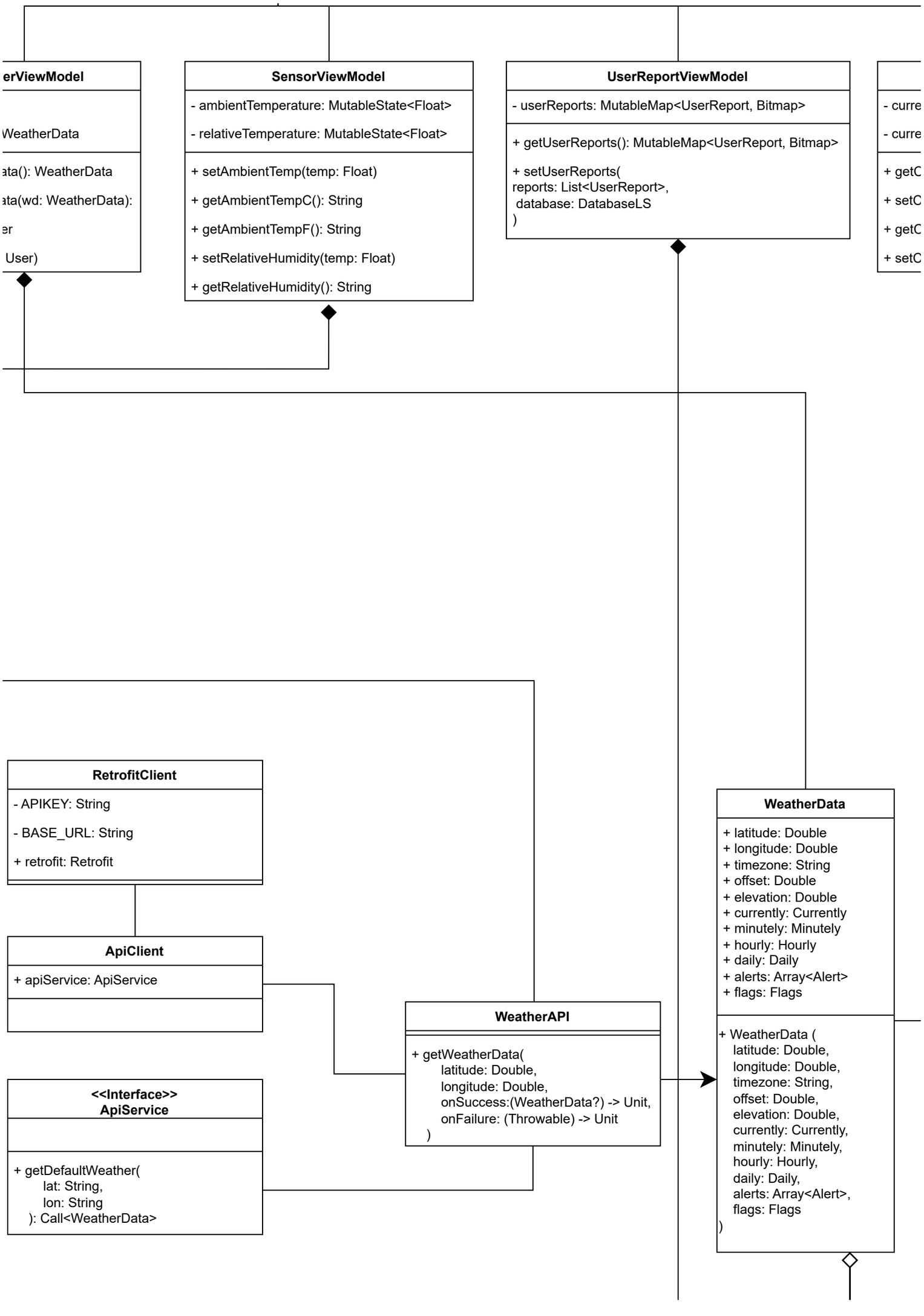
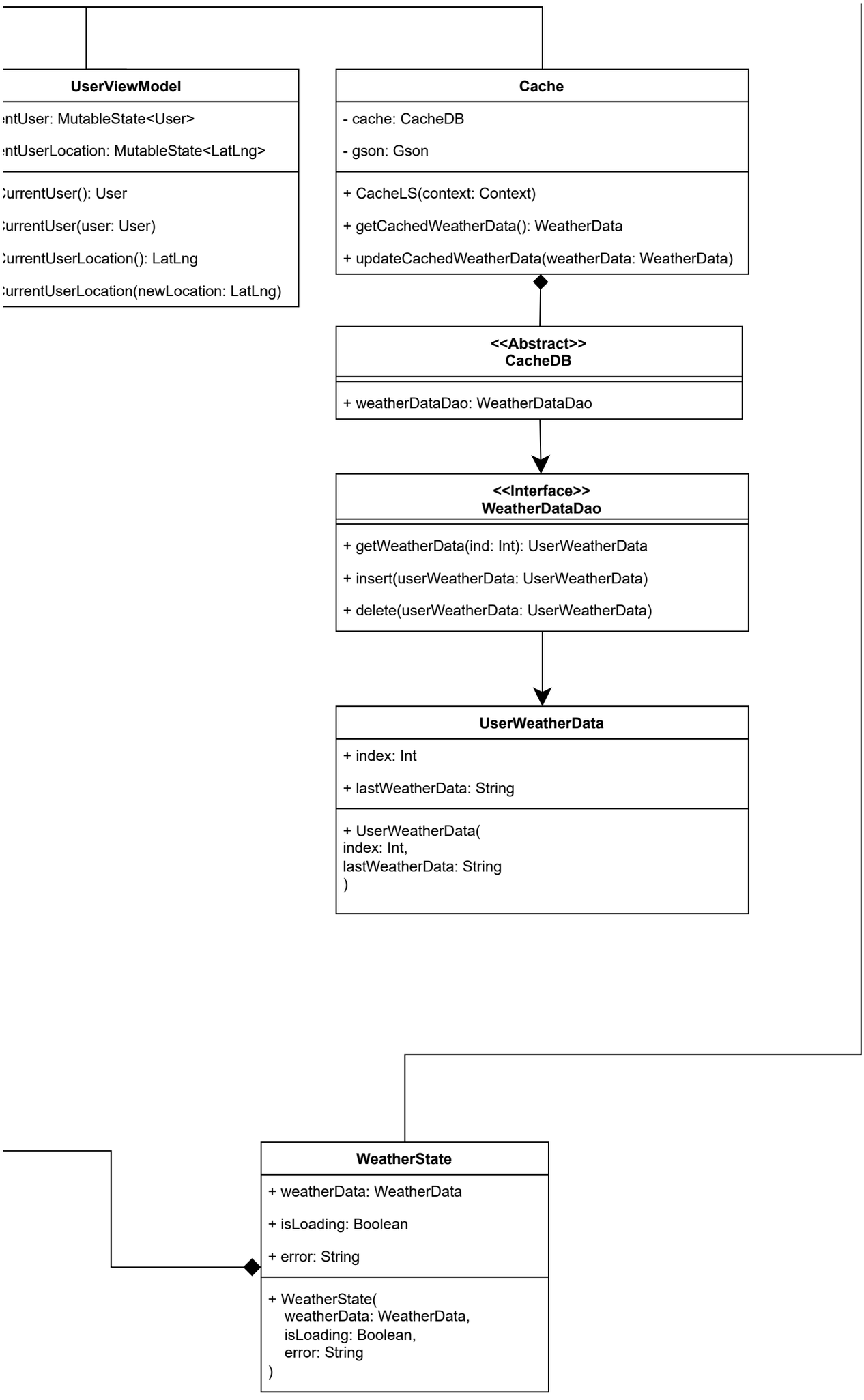

LocationService
- locationManager: LocationManager
- locationListener: LocationListener
+ onBind(intent: Intent): IBinder
+ onStartCommand(
intent: Intent,
flags: Int,
startId: Int): Int
+ onDestroy()
LocationRepository
- _currentLocation: MutableStateFlow<LatLng>
+ currentLocation: StateFlow<LatLng>
+ updateLocation(location: LatLng)

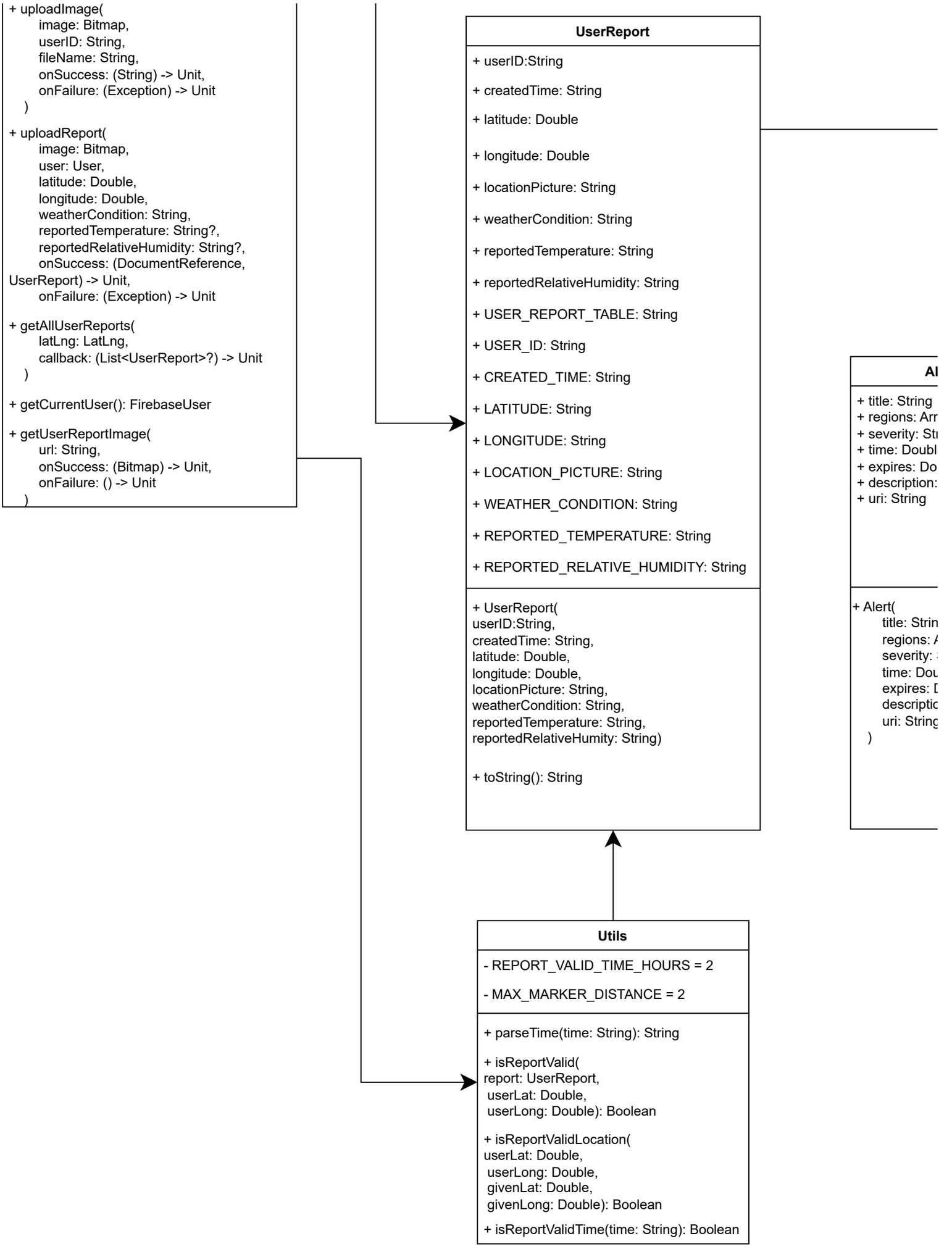

Screen
.oginScreen: Screen
Registerscreen: Screen
MapScreen: Screen
SettingsScreen: Screen
VeatherScreen: Screen
Screen(
e: String,
n: ImageVector,
een_route: String,
owBottomBar: Boolean,
Click: () -> Unit
App
+ Login: App
+ Main: App
+ App()
WeatherItem
+ weatherType: WeatherType
+ isSelected: MutableState<Boolean>
+ WeatherItem(
weatherType: WeatherType,
isSelected: MutableState<Boolean>
)
LocalSkyAppRouter
+ currentApp: MutableState<App>
+ currentScreen: MutableState<Screen>
+ navigateTo(destination: Screen)
+ changeApp(app: App)
MainActivity
- fusedLocationClient: FusedLocationClient
- locationCallback: LocationCallback
- cacheLS: CacheLS
- database: DatabaseLS
- userViewModel: UserViewModelLS
- weatherViewModel: WeatherViewModelLS
- userReportViewModel: UserReportViewModelLS
- sensorViewModel: SensorViewModelLS
- requestingLocationUpdates: Boolean
- startTempSensor()
- startRelativeHumiditySensor()
- getCurrentLocationAndUpdateWeatherViewModel()
- setScreenActions()
- checkPerms()
- updateValuesFromBundle(savedInstanceState: Bundle)
- startLocationUpdates()
- stopLocationUpdates()
RelativeHumiditySensor
- instanceReference: WeakReference<TemperatureS
+ getInstance(context: Context): TemperatureSensor
TemperatureSensor
- instanceReference: WeakReference<TemperatureS
+ getInstance(context: Context): TemperatureSensor

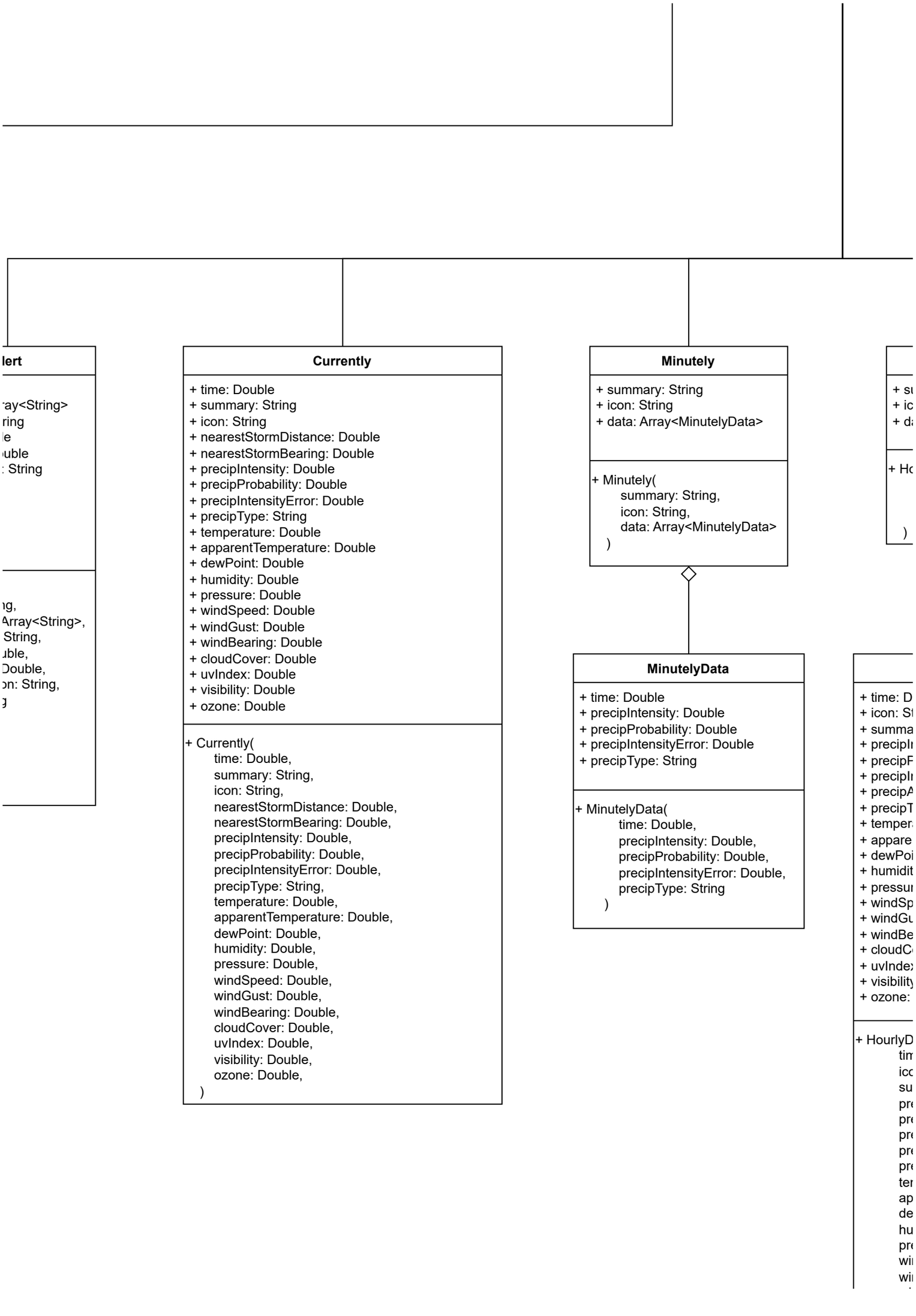

WeatherType
+ ClearDay: WeatherType
+ ClearNight: WeatherType
+ Cloudy: WeatherType
+ Foggy: WeatherType
+ PartlyCloudyDay: WeatherType
+ PartlyCloudyNight: WeatherType
+ Rain: WeatherType
+ Snow: WeatherType
+ Windy: WeatherType
+ allWeatherTypes: Array<WeatherType>
+ WeatherType(
weatherSummary: String,
iconRes: Int
)
+ fromWeatherReport(summary: String): WeatherType
AndroidSensor
- context: Context
- sensorFeature: String
+ doesSensorExist: Boolean
- sensorManager: SensorManager
- sensor: Sensor
+ AndroidSensor(
context: Context,
sensorFeature: String,
sensorType: Int
)
+ startListening()
+ stopListening()
+ onSensorChanged()
+ onAccuracyChanged(p0: Sensor, p1: Int): Unit
<<Abstract>>
MeasurableSensor
sensorType: Int
onSesorValueChanged: (List<Float> -> Unit)
+ doesSensorExist: Boolean
+ startListening()
+ stopListening()
+ setOnSensorValueChanged(listener: (List<Float> -> Unit))

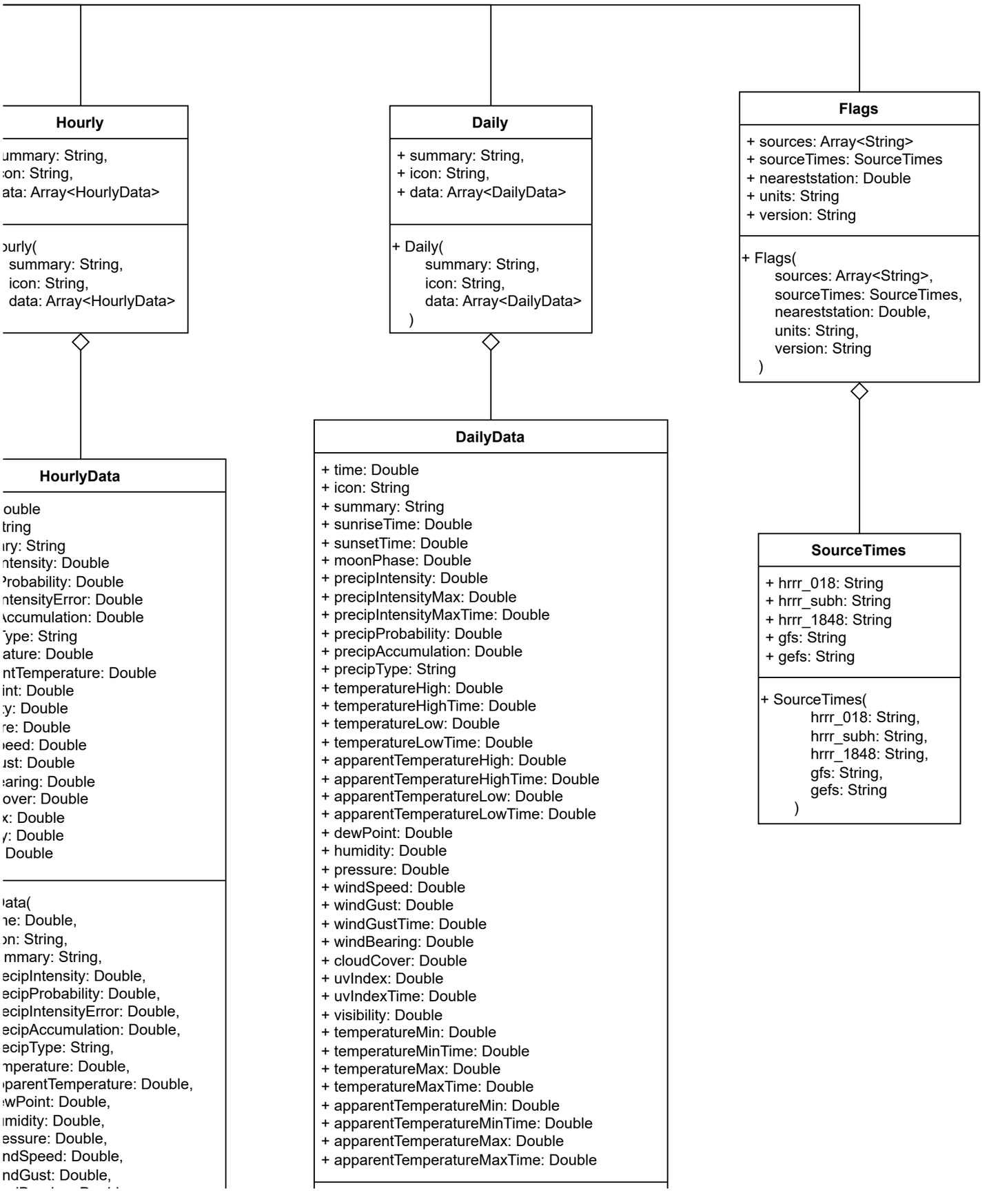

Weath
- user: User
- weatherData \
+getWeatherDa
+setWeatherDa
+getUser(): Use
+setUser(user:
Database
- database: FirebaseFirestore
- auth: FirebaseAuth
- storage: FirebaseStorage
+ createUser(
firstName: String,
lastName: String,
email: String,
password: String,
onSuccess: (FirebaseUser, User) -> Unit,
onFailure: (Exception) -> Unit
)
+ signIn(
email: String,
password: String,
onSuccess: (FirebaseUser) -> Unit,
onFailure: (Exception) -> Unit
)
+ signOut()
+ updateUser(
user: User,
onSuccess: () -> Unit,
onFailure: () -> Unit
)
+ getUserTable(
onSuccess: (QuerySnapshot?) -> Unit,
onFailure: () -> Unit
)
+ getUserByID(
userID: String,
onSuccess: (QueryDocumentSnapshot?,
User?) -> Unit,
onFailure: () -> Unit
)
+ removeUser(
userID: String,
onSuccess: (Task<Void>) -> Unit,
onFailure: (Exception) -> Unit
)
+ createUserReport(
user: User,
createdTime: String,
latitude: Double,
longitude: Double,
locationPicture: String,
weatherCondition: String,
reportedTemperature: String?,
reportedRelativeHumidity: String?,
onSuccess: (DocumentReference,
UserReport) -> Unit,
onFailure: (Exception) -> Unit
)
User
+ lastName: String
+ email:String
+ firstName:String
+ userID:String
+ User(
userID: String,
email: String,
firstName: String,
lastName: String)
+ toString(): String


erViewModel
WeatherData
ata(): WeatherData
ata(wd: WeatherData):
er
User)
SensorViewModel
- ambientTemperature: MutableState<Float>
- relativeTemperature: MutableState<Float>
+ setAmbientTemp(temp: Float)
+ getAmbientTempC(): String
+ getAmbientTempF(): String
+ setRelativeHumidity(temp: Float)
+ getRelativeHumidity(): String
UserReportViewModel
- userReports: MutableMap<UserReport, Bitmap>
+ getUserReports(): MutableMap<UserReport, Bitmap>
+ setUserReports(
reports: List<UserReport>,
database: DatabaseLS
)
- curre
- curre
+ getC
+ setC
+ getC
+ setC
RetrofitClient
- APIKEY: String
- BASE_URL: String
+ retrofit: Retrofit
ApiClient
+ apiService: ApiService
<<Interface>>
ApiService
+ getDefaultWeather(
lat: String,
lon: String
): Call<WeatherData>
WeatherAPI
+ getWeatherData(
latitude: Double,
longitude: Double,
onSuccess:(WeatherData?) -> Unit,
onFailure: (Throwable) -> Unit
)
WeatherData
+ latitude: Double
+ longitude: Double
+ timezone: String
+ offset: Double
+ elevation: Double
+ currently: Currently
+ minutely: Minutely
+ hourly: Hourly
+ daily: Daily
+ alerts: Array<Alert>
+ flags: Flags
+ WeatherData (
latitude: Double,
longitude: Double,
timezone: String,
offset: Double,
elevation: Double,
currently: Currently,
minutely: Minutely,
hourly: Hourly,
daily: Daily,
alerts: Array<Alert>,
flags: Flags
)


UserViewModel
ntUser: MutableState<User>
ntUserLocation: MutableState<LatLng>
urrentUser(): User
urrentUser(user: User)
urrentUserLocation(): LatLng
urrentUserLocation(newLocation: LatLng)
Cache
- cache: CacheDB
- gson: Gson
+ CacheLS(context: Context)
+ getCachedWeatherData(): WeatherData
+ updateCachedWeatherData(weatherData: WeatherData)
<<Abstract>>
CacheDB
+ weatherDataDao: WeatherDataDao
<<Interface>>
WeatherDataDao
+ getWeatherData(ind: Int): UserWeatherData
+ insert(userWeatherData: UserWeatherData)
+ delete(userWeatherData: UserWeatherData)
UserWeatherData
+ index: Int
+ lastWeatherData: String
+ UserWeatherData(
index: Int,
lastWeatherData: String
)
WeatherState
+ weatherData: WeatherData
+ isLoading: Boolean
+ error: String
+ WeatherState(
weatherData: WeatherData,
isLoading: Boolean,
error: String
)


+ uploadImage(
image: Bitmap,
userID: String,
fileName: String,
onSuccess: (String) -> Unit,
onFailure: (Exception) -> Unit
)
+ uploadReport(
image: Bitmap,
user: User,
latitude: Double,
longitude: Double,
weatherCondition: String,
reportedTemperature: String?,
reportedRelativeHumidity: String?,
onSuccess: (DocumentReference,
UserReport) -> Unit,
onFailure: (Exception) -> Unit
+ getAllUserReports(
latLng: LatLng,
callback: (List<UserReport>?) -> Unit
)
+ getCurrentUser(): FirebaseUser
+ getUserReportImage(
url: String,
onSuccess: (Bitmap) -> Unit,
onFailure: () -> Unit
)
UserReport
+ userID:String
+ createdTime: String
+ latitude: Double
+ longitude: Double
+ locationPicture: String
+ weatherCondition: String
+ reportedTemperature: String
+ reportedRelativeHumidity: String
+ USER_REPORT_TABLE: String
+ USER_ID: String
+ CREATED_TIME: String
+ LATITUDE: String
+ LONGITUDE: String
+ LOCATION_PICTURE: String
+ WEATHER_CONDITION: String
+ REPORTED_TEMPERATURE: String
+ REPORTED_RELATIVE_HUMIDITY: String
+ UserReport(
userID:String,
createdTime: String,
latitude: Double,
longitude: Double,
locationPicture: String,
weatherCondition: String,
reportedTemperature: String,
reportedRelativeHumity: String)
+ toString(): String
Al
+ title: String
+ regions: Arr
+ severity: St
+ time: Doubl
+ expires: Do
+ description:
+ uri: String
+ Alert(
title: Strin
regions: A
severity:
time: Dou
expires: D
descriptio
uri: String
)
Utils
- REPORT_VALID_TIME_HOURS = 2
- MAX_MARKER_DISTANCE = 2
+ parseTime(time: String): String
+ isReportValid(
report: UserReport,
userLat: Double,
userLong: Double): Boolean
+ isReportValidLocation(
userLat: Double,
userLong: Double,
givenLat: Double,
givenLong: Double): Boolean
+ isReportValidTime(time: String): Boolean


lert
ray<String>
ring
e
uble
: String
ıg,
Array<String>,
String,
uble,
Double,
on: String,
g
Currently
+ time: Double
+ summary: String
+ icon: String
+ nearestStormDistance: Double
+ nearestStormBearing: Double
+ precipIntensity: Double
+ precipProbability: Double
+ precipIntensityError: Double
+ precipType: String
+ temperature: Double
+ apparentTemperature: Double
+ dewPoint: Double
+ humidity: Double
+ pressure: Double
+ windSpeed: Double
+ windGust: Double
+ windBearing: Double
+ cloudCover: Double
+ uvIndex: Double
+ visibility: Double
+ ozone: Double
+ Currently(
time: Double,
summary: String,
icon: String,
nearestStormDistance: Double,
nearestStormBearing: Double,
precipIntensity: Double,
precipProbability: Double,
precipIntensityError: Double,
precipType: String,
temperature: Double,
apparentTemperature: Double,
dewPoint: Double,
humidity: Double,
pressure: Double,
windSpeed: Double,
windGust: Double,
windBearing: Double,
cloudCover: Double,
uvIndex: Double,
visibility: Double,
ozone: Double,
)
Minutely
+ summary: String
+ icon: String
+ data: Array<MinutelyData>
+ Minutely(
summary: String,
icon: String,
data: Array<MinutelyData>
)
MinutelyData
+ time: Double
+ precipIntensity: Double
+ precipProbability: Double
+ precipIntensityError: Double
+ precipType: String
+ MinutelyData(
time: Double,
precipIntensity: Double,
precipProbability: Double,
precipIntensityError: Double,
precipType: String
)
+ su
+ ic
+ da
+ Ho
)
+ time: D
+ icon: St
+ summa
+ precipI
+ precipF
+ precipI
+ precipA
+ precipT
+ temper
+ appare
+ dewPoi
+ humidit
+ pressur
+ windSp
+ windGu
+ windBe
+ cloudC
+ uvIndex
+ visibility
+ ozone:
+ HourlyD
tim
ico
su
pre
pre
pre
pre
pre
ter
ap
de
hu
pre
wi
wi


Hourly
ummary: String,
on: String,
ata: Array<HourlyData>
ourly(
summary: String,
icon: String,
data: Array<HourlyData>
HourlyData
ouble
tring
ry: String
ntensity: Double
robability: Double
ntensityError: Double
ccumulation: Double
ype: String
ature: Double
ntTemperature: Double
int: Double
y: Double
e: Double
eed: Double
st: Double
aring: Double
over: Double
x: Double
y: Double
Double
ata(
ne: Double,
on: String,
mmary: String,
ecipIntensity: Double,
ecipProbability: Double,
ecipIntensityError: Double,
ecipAccumulation: Double,
ecipType: String,
mperature: Double,
parentTemperature: Double,
wPoint: Double,
midity: Double,
essure: Double,
ndSpeed: Double,
ndGust: Double,
Daily
+ summary: String,
+ icon: String,
+ data: Array<DailyData>
+ Daily(
summary: String,
icon: String,
data: Array<DailyData>
)
DailyData
+ time: Double
+ icon: String
+ summary: String
+ sunriseTime: Double
+ sunsetTime: Double
+ moonPhase: Double
+ precipIntensity: Double
+ precipIntensityMax: Double
+ precipIntensityMaxTime: Double
+ precipProbability: Double
+ precipAccumulation: Double
+ precipType: String
+ temperatureHigh: Double
+ temperatureHighTime: Double
+ temperatureLow: Double
+ temperatureLowTime: Double
+ apparentTemperatureHigh: Double
+ apparentTemperatureHighTime: Double
+ apparentTemperatureLow: Double
+ apparentTemperatureLowTime: Double
+ dewPoint: Double
+ humidity: Double
+ pressure: Double
+ windSpeed: Double
+ windGust: Double
+ windGustTime: Double
+ windBearing: Double
+ cloudCover: Double
+ uvIndex: Double
+ uvIndexTime: Double
+ visibility: Double
+ temperatureMin: Double
+ temperatureMinTime: Double
+ temperatureMax: Double
+ temperatureMaxTime: Double
+ apparentTemperatureMin: Double
+ apparentTemperatureMinTime: Double
+ apparentTemperatureMax: Double
+ apparentTemperatureMaxTime: Double
Flags
+ sources: Array<String>
+ sourceTimes: SourceTimes
+ nearestStation: Double
+ units: String
+ version: String
+ Flags(
sources: Array<String>,
sourceTimes: SourceTimes,
nearestStation: Double,
units: String,
version: String
)
SourceTimes
+ hrrr_018: String
+ hrrr_subh: String
+ hrrr_1848: String
+ gfs: String
+ gefs: String
+ SourceTimes(
hrrr_018: String,
hrrr_subh: String,
hrrr_1848: String,
gfs: String,
gefs: String
)

wi
clo
uv
vis
oz
)

ndBearing: Double,
oudCover: Double,
Index: Double,
sibility: Double,
one: Double,

+ DailyData(
time: Double,
icon: String,
summary: String,
sunriseTime: Double,
sunsetTime: Double,
moonPhase: Double,
precipIntensity: Double,
precipIntensityMax: Double,
precipIntensityMaxTime: Double,
precipProbability: Double,
precipAccumulation: Double,
precipType: String,
temperatureHigh: Double,
temperatureHighTime: Double,
temperatureLow: Double,
temperatureLowTime: Double,
apparentTemperatureHigh: Double,
apparentTemperatureHighTime: Double,
apparentTemperatureLow: Double,
apparentTemperatureLowTime: Double,
dewPoint: Double,
humidity: Double,
pressure: Double,
windSpeed: Double,
windGust: Double,
windGustTime: Double,
windBearing: Double,
cloudCover: Double,
uvIndex: Double,
uvIndexTime: Double,
visibility: Double,
temperatureMin: Double,
temperatureMinTime: Double,
temperatureMax: Double,
temperatureMaxTime: Double,
apparentTemperatureMin: Double,
apparentTemperatureMinTime: Double,
apparentTemperatureMax: Double,
apparentTemperatureMaxTime: Double
)