**We would like you to make a weather application**

**Client Text:**

We would like you to make a weather application. The app should get the current GPS location of the device, make a web-service call to fetch the weather for that location (you can use any public API), parse the response and make a model, then display the relevant data on the UI. The app should be able to display a map with the user’s current location and weather. The user should be able to interact with the map (zoom, pan, etc), and after clicking on the map, all the relevant weather data for the user-clicked location should be displayed in an overlay on the map. It is up to the developer to decide how the data will be cached for the app (if necessary), but the app should work in portrait and landscape mode, have an intuitive UI following Android conventions, and the code should be documented and polished.

We would also like to see an SDK for the app. You’ll need to create an SDK and include all necessary parts like a readme, git-ignore, etc, an example of how to implement the SDK, and your weather app itself should use the SDK. It should be possible for us to use your SDK in another app without use of your weather app.  
  
Feel free to add any other features, like multi-day forecasts, weather of neighboring cities, or maybe even weather related news.

**Requirements:**

1. **App should get the current GPS location of the device**
2. **Make a web-service call to fetch the weather for that location (you can use any public API)**
   1. **Parse the response**
   2. **Make a model**
3. **Display the relevant data on the UI**
4. **App should be able to display a map with the user’s current location and weather**
5. *User should be able to interact with the map (zoom, pan, etc)*
6. *Map Click Action*
   1. *All the relevant weather data for the user-clicked location should be displayed in an overlay on the map*
7. **It is up to the developer to decide how the data will be cached for the app (if necessary) (YES)**
8. *App should work in portrait and landscape mode*
9. *Have an intuitive UI following Android conventions*

*Code should be documented and polished*

**SDK Requirements:**

1. Include all necessary parts like a readme, git-ignore, etc, an example of how to implement the SDK
2. Weather app itself should use the SDK
3. Should be possible for us to use your SDK in another app without use of your weather app

Optional Features:

1. Multi – Day Forecasts
2. Weather of Neighboring Cities
3. Weather Related News

Competing Applications Research:

* Native Android app displays current weather and 5 day forecast for 1 location
* The Weather Channel App
  + Current Location Weather
  + Weather News
  + Ads
  + Can create a personalized account
  + Animated weather layer
  + Multiple layer choices for weather map
  + Severe weather notifications
* Accuweather
  + Very simple interface, love it
    - Updated to follow latest Android recommendations
  + Initial page shows current weather, a minutecast, and what to expect weather wise in one sentence
  + Tabs for now, hourly, daily, maps and video news
  + Severe weather notifications
  + The map is easier to understand than Weather Channel App, it does choose to limit the number of layers
  + Can add multiple locations and switch between them
* Yahoo Weather
  + Severe weather notifications
  + Ability to add multiple cities
  + Presents simple information with pleasant background screen saving advanced details to be scrollable below this
  + Much simpler weather map than other app however it also displays less useful data
  + UI is simple, weather info is easily understandable with the exception of the weather map
  + Mostly a portal to their other applications
* Weather Underground
  + Social, gives users ability to confirm weather condition
  + Editable home page
  + The UI transitions are frustrating; many times they only display a blank page or actions have no obvious effect
  + The weather map is comparable to the rest with the exception it uses temperature annotations over the local area vs a heat map
  + UI is consistent with new android standards

Conclusion, most weather apps follow the same generic pattern. What I notice most is that these apps tie users to locations they know or popular tourist destinations. Giving weather info for those locations… which is exactly what they’re supposed to do. However, opportunities for making a more dynamic app would be to deviate from the idea that people only want to know when their city is about to be flooded. While the users’ location is ultimately the most important, I want the application to also be like an ideal weather finder. Instead of “hey a hurricane is about to destroy your livelihood” being the only notification a user can expect, let’s make it “Hey, sunny weather and blue skys at Hanging Rock this weekend”. Be more suggestive of where to find weather that the user prefers in addition to the basic data. That said, if needing inspiration Accuweather is my preference for its simplicity.

Decisions: Need to implement the SDK first, but before that I need to highlight functions of the sdk, research, develop plan and then iteratively work on functions independent of a UI by using tests.

SDK Questions:

What are important functions of the sdk?

* Accepts one or more gps locations
* Fetching weather data for each location
* Storing variable day weather data for multiple locations
* Produce the basic map view with animations, with implementable callbacks so applications can override functionality. Ideally users would only need to instantiate a “JDWeatherMapView” for this basic functionality to happen.
* Should have methods for dynamically retrieving weather data for locations over 1 - X day period

What additional functions would be nice to have?

* Automatically searching for weather in near-by locations that might be desirable to user
* Suggesting weekend travel locations based on user definable ideal weather
* Determining travel direction and determining coming weather

How can I implement core features?

* Most android devices have easily accessible gps information
* Overriding Google Maps API is an option for implementing a map interface
* ~~OpenWeatherMap seems to have the resources for current weather data, forecast data, historical data and a weather map layers option which could be used for developing map UI~~
  + [~~http://openweathermap.org/api~~](http://openweathermap.org/api)
* Will have to create simple annotations over google maps for the time being
* Utilizing SQLite would be ideal for caching multi day forecasts, I may have to save the OpenWeatherMap map layer files in a separate location depending on the implementation of this system

How can I implement additional features?

* Google Place API could provide the functionality for finding nearby locations to the one specified by a user
  + <https://developers.google.com/places/supported_types#table1>

Rough Plan:

Create a dedicated plan document for developing the SDK. Ideally I will break down the sdk into discrete implementable functions or activities. After each function or activity write tests to confirm it is operational. The plan should consider the timeframe and importance of delivering a final product. As such core features will be implemented first, the sdk will be delivered so that a test application may be created. After review of the test application and ensuring it meets minimal specified requirements I will then go back and add the additional features to the sdk. Once again, time permitting, I will deliver the sdk again and work on the updated sample application.

Ideally my production process will follow an Exploratory Lifecycle as identified by Disciplined Agile Delivery: <http://www.disciplinedagiledelivery.com/lifecycle/>

SDK Plan: DocumentB\_SDKPlan