We would like you to solve the following coding challenge.

Please read ALL of the content before you begin.

**Weather Alert**

**Assessment Criteria**

We analyse your coding challenge submission using several criteria, amongst these:

* Code quality
* Readability
* Relevant technology selection and implementation
* pragmatic design choices
* good testing practices
* consistent coding techniques
* evidence of some functional testing - (e.g. a reasonably robust deliverable)
* supportability
* the breadth of skills demonstrated

We expect to receive good quality code (some non-obvious bugs are ok of course) and unit tests. It is not expected that the challenge would take more than one or two evenings to complete.

Please consider that you may be required to reason about your solution and justify implementation and design choices you have made.

**The App - Weather Alert**

Implement an application that will allow the user to check the wind conditions and forecast at one or more locations around the world.

**User Interface**

The design of the UI is totally down to you. We expect you to choose a sensible user interface design and implement controls, views and layout in a way that will offer users a good experience. We expect the use of the latest technologies in the app and we expect support modern versions of the operating system – please make it clear if there are version restrictons on your submission. Note: we may test the app on different screen sizes (common ones) for your submission.

**The Requirements**

**Executive summary**

We require an app that allows a user to monitor current and/or forecast wind information from selected locations around the world.

**The Data Source**

In order to complete the coding challenge we would like you to use the openweather API. It is well documented with an up to date wiki offering help and advice. The API can be found here.

<http://openweathermap.org/API#weather>

All of the functionality within the requirements can be achieved using this service. You are free to use either the JSON or XML API.

**The Main View – Favourite locations**

This is the main interface of the application and the starting point when launching your app.

In this view the user can see previously added ‘favourite’ locations and add new ‘favourite’ locations for which they want to monitor wind information. How this is implemented in the user interface is up to you. Think about what data you can practically display here, about current or forecast wind information.

**Getting Locations**

Read the API documentation, you can search for Cities by substring.

Users will add locations they are interested to their favourites, you should bear this in mind when designing the user interface. Ensure you make good practical use of the data available.

There is no limit to how many favourite locations a user may add, you may implement a limit if required.

On Start-up, the application should check if the user has already chosen locations in previous sessions and ensure that they are presented to the user in this Main view.

When the application starts up for the very first time, or when no favourite locations have been selected, there will be no favourite data to display. In this situation the user interface should be clear and intuitive. As a minimum, an add station option should be present.

**The Wind forecast - detail view**

For any favourite location displayed in the main view the user should be provided with functionality allowing them to see full/further forecast information about this location

The UI design is totally down to you but bear in mind that

* The information must be clear and navigation should be intuitive.
* Some of the data may be better represented as images.  For example - wind direction, or wind speed, feel free to research this, the concepts of wind direction is quite straight-forwards. <http://en.wikipedia.org/wiki/Cardinal_direction>

Forecast data is well supported in the [openweathermap.org](http://openweathermap.org/) API. Remember we are interested in Wind forecast information.

The [openweathermap.org](http://openweathermap.org/) API is metered to prevent DOS and other abuse. More than 2000 API requests in one hour can result in a ban or, more typically, a throttle being put on subsequent requests. More details about this are available at the [openweathermap.org](http://openweathermap.org/) site, bear this in mind in your design.

**What to do when you are complete**

Please send us the solution as a shared link to a file hosted in any of the many file sharing services (dropbox,mediafire, [drop.io](http://drop.io/),etc) or to a source repository e.g. github or bitbucket. email may be problematic due to attachment size.

If you have problems or any questions about the challenge, please feel free to contact us.