**Cross-Platform Development Assessment Item 1 Report**

**Concept**

**Overview**

Have you ever been in a situation where you looked at the current forecast for your current location or the location you plan to travel to? Only for the weather forecast to be completely wrong, resulting in you getting soaked without your coat to save you! Or perhaps it claims that it is raining, only for it to be sunny, resulting in you having to lug your coat around?

Weatherly is a community based weather app that harnesses the power of users to report back on the actual conditions of weather based on their current locations. This provides real time feedback on the actual conditions of a location rather than just relying on predictions provided by the forecasting services. With Weatherly, gone are the days of guessing on whether the forecasted weather is true or not. Weatherly also provides a forecast just like regular weather apps that provides fast, timely updates to make sure you have all the information you need from multiple sources.

With Weatherly you’ll never be caught unprepared again!

Barcode:



**Requirements**

The weatherly app can be used in a variety of user scenarios, from travelling to a location such as going on holiday, or embarking on a typical daily commute to work. The target group for the app is users of any age group that travel often, stay over in remote locations for a few days, and/or users who prefer to travel with just the most essential items.

As a frequent commuter Omar wants to have a more accurate weather report so he can make an informed decision about what to take with him to university. As a frequent commuter Omar also wants the ability to save his favourite locations so he can keep track of their weather conditions.

Biff the businessman frequently travels to work and takes public transportation to get there, he prefers to travel lighter as it is more comfortable for his journey. As he only takes a briefcase with his work it doesn’t provide any space for an umbrella or coat for his travels. He often finds that after watching the morning weather forecast on TV before work, he takes his umbrella to work with him only to find that the weather is different than the forecast and as a result he has carried his umbrella around with him for nothing. Biff then searches for a better solution to this problem and stumbles upon Weatherly on the app store. Weatherly helps Biff to make a more informed decision about if he will need to take his umbrella to work all day. Biff uses weatherly one day that reports that his work location will be cloudy but dry, Biff then realises that the community report says others, with over 2000 users reporting that there is rain present. With this new knowledge he can now more easily justify taking the extra luggage with him to work as the community has reported rain is indeed present at his work location, he then is able to keep his suit dry and his hair tidy for his big conference meeting.

As such, the functional requirements of this app would be the use of a custom API hosted on a server which the app uses to provide the community feature of users reporting weather conditions in their location. The app also needs to tap into a weather API to pull information about the location the user is currently situated in, and to provide weather data for remote/added locations. It then needs to store those location locally, so the app can “remember” the locations added for the user, so they don’t need to continually input them. This data then needs to be displayed onto the app’s UI for users to view.

Non-functional requirements would pertain to making the community ratings quick and easy for users to participate in. A community based system that allows users to anonymously post updates on the current weather conditions of their current locations and to receive community updates on other locations.

**Competitors**

The two apps that Weatherly would compete with are BBC Weather (BBC, xxxx) and Met Office Weather Forecast (Met Office, xxxx).

The BBC Weather app’s main window consists of a full screen background that reacts based on the current weather conditions of a given location. The app provides weather information in great detail, ranging from reports of air quality to humidity and visibility conditions. As a result of all this detailed information, the main app’s window is densely populated without any logical placement of these elements, this leaves the window looking cluttered and potential confusing on first use (See Figure 1). One aspect of the main window’s UI is the increased emphasis the app puts on critical information, such as the current temperature, weather condition, and location are all larger in size and bold. This assists the critical information in standing out from the rest of the less relevant data in main window, which immediately draws more of the user’s attention (See Figure 1).

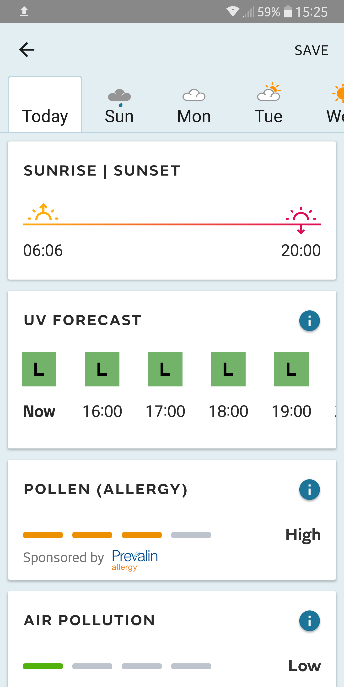
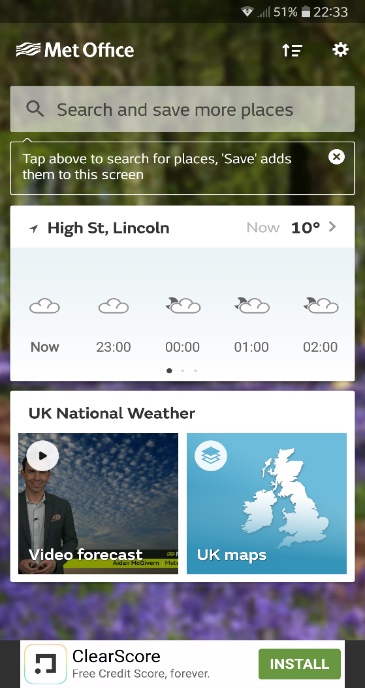


*Figure 1: BBC Weather app’s main window of a location’s weather condition (BBC, xxxx).*

The app’s functionality allows users to search for any location around the world and add it to a personal saved list, which saves the user time and adds to the convenience of the app. One feature that was not apparent was that the UI allows for swiping to the left to reveal an hourly forecast, there is no indication to show to users that this feature exists and is possible.

This app could have benefited from a onetime tutorial that introduces the app’s features to its users to get them more accustomed with the UI and functionality of the app. This is also reflected through some of the symbols the app employs that don’t entirely indicate what the data may be showing, for example the symbol that is employed for wind direction and speed is not entirely clear, nor does it tell the user what metric is being used. This is a negative element that Weatherly will try to avoid, by making all weather data easily readable and laid out in a logical fashion.

The 2nd app is MetOffice Weather Forecast (MetOffice, xxxx), its main window is more minimal in design compared to the BBC Weather app (BBC, xxxx) and shows a list of locations first, rather than current location conditions (See figure 2). From this interface the app prompts the user to use the search bar to find locations they may be interested in, which provides the user with direction, an improvement over the BBC Weather app (See figure 2).



*Figure 2: Main window of the MetOffice Weather Forecast app being displayed (left) and the current weather conditions of a location (right) (MetOffice, xxxx).*

The user experience of the app is an improvement over BBC Weather, with better spacing and clearer separation between contents which better breaks up the information and makes the window look cleaner (See figure 2). Weatherly will employ a similar approach of keeping the UI clean and information displayed in a logical fashion.

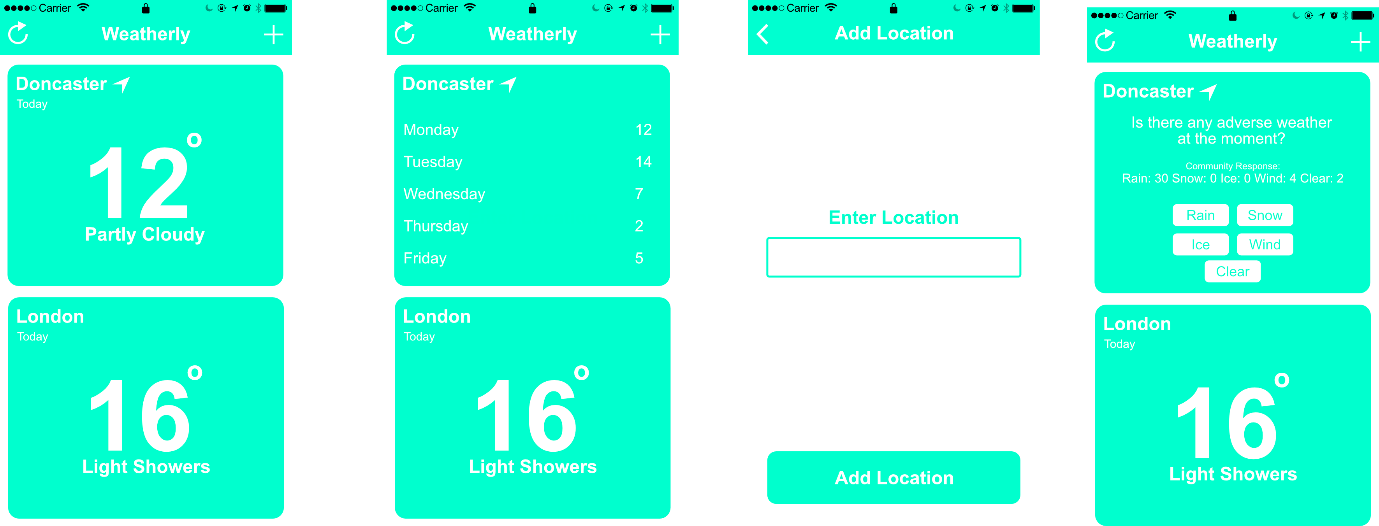
MetOffice Weather Forecast provides substantial detail, with the app harnessing the power of MetOffice’s API to provide rich detail about the current and future conditions of a location. Weatherly will also make use of a forecast API to gather information about a location’s current and future weather conditions, combined with the community features this will help users paint a more accurate image of the what weather is like at a given location. With both apps, Weatherly distinguishes from them with its unique community ratings feature that hopes to provide a more accurate weather forecast for locations by leveraging the power of the community.

**Prototyping**

Prototyping work was approached through the creation of non-interactive mock-ups using Affinity Designer (Affinity, xxxx). These mock-ups show the different windows the user can navigate to when using the app and shows the layout of the user interface. These mock-ups were then presented to fellow colleagues along with a SUS questionnaire to evaluate their effectiveness in certain areas of design and usability. The testers were also encouraged to suggest improvements and to say what they liked and disliked. This data would then be used to improve the designs of the prototype, hopefully leading towards a sufficient design that would then be used for the final app design and layout.

The first stage of prototyping involved an early concept that had most of the functional requirements situated in a large card that took up most of the display. The user would then swipe on the card to flip it over displaying more detailed information about the weather, such as a weekly look or wind speeds (See figure 3). Multiple cards can be added to the main window, forming a list of locations the user is interested in seeing.

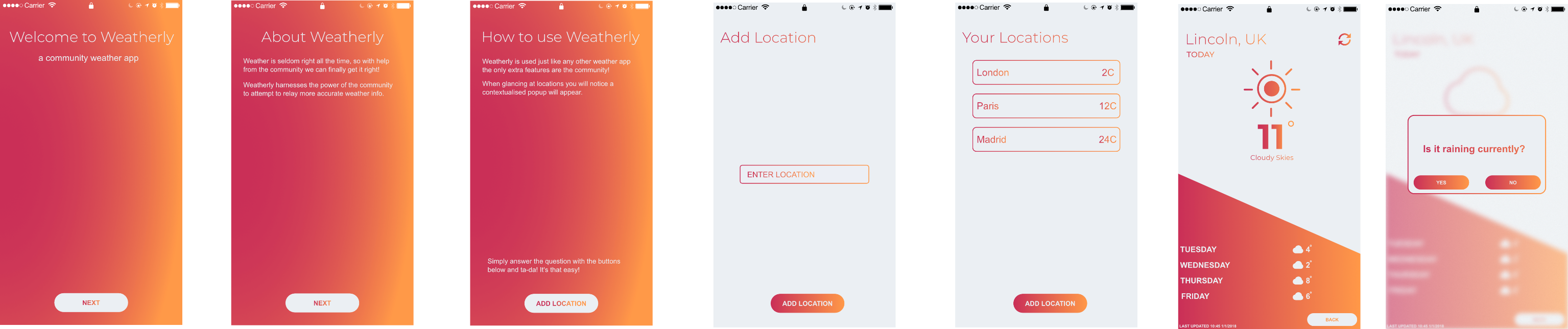
This design incorporated all the functional requirements of the app into the central card, performing a swipe gesture on the card reveals more information about the weekly forecast of the location. The current location’s card would include a community feature section at the end of the card’s content. This allows the user to see the community report on the weather in this given location and provides the user with the option to add to the community report by casting their vote on the current weather conditions of their current location.

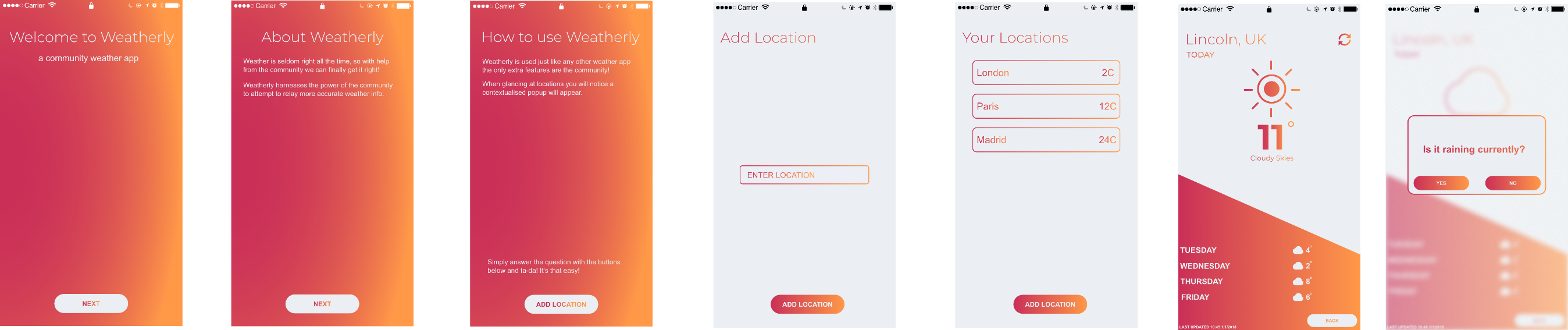


*Figure 3: Prototype 1’s various windows (Colours were not final).*

This design was not well received by user testing participants, as many noted that it was impractical to keep having to constantly flip the card to look between different pieces of information. The testers did not find it very easy to use because of this feature, this prompted a rethink on how the user interface was to be laid out but testers did rate consistency very favourably, but described that the community feature was not easily understood and could do with clarity (See figure 3). The enter location page was specifically acceptable to testers, providing verbal feedback that the placement of the elements on that page were great for one handed usage (with exception to the back button).

The next prototype had a drastically different layout from the previous, learning from the issues highlighted by the user testers. The cards layout was replaced with a much simpler list of locations, which users then tap on to take them to that location’s information page. While containing less weather data, the new layout has an improved design with all information being comfortably displayed in one window (See figure 4).





*Figure 4: The final prototype/mock-up that inspired the final app design.*

Following on with suggestions pointed out by user testers evaluating the last prototype, a tutorial was introduced into this prototype which helps user understand the basics of the app and how the community ratings features works. The tutorial was evaluated by fellow colleagues to determine its effectiveness, which was met with positive responses. Testers noted that the tutorial better helped them understand the community ratings feature, how it works, how to use it, and how the user may potentially benefit from this feature.

Another improvement over the original design is how the community ratings feature has been implemented. The feature now provides the user with a popup asking them about what the current weather conditions are like outside compared to the forecasted prediction. This method, while arguably more intrusive, means that the likelihood of users contributing and using the community ratings feature will increase significantly.

The popup improves over the original design by making it easier and quicker for the user to answer, rather than them swiping to the end of the card to find the feature. This improvement also aligns with the 3-click rule of usability, that implies that the average user will be left frustrated if they cannot find what they are looking for in 3 clicks (Porter, 2003).

As the app could be used in a variety of public/outdoor environments, extra care was given to one handed usage of the app, by making sure all critical elements are reachable even on larger display devices.

**Final App**

**Reflection**

Cross-Platform technologies are greatly assisting app developers in making it easier to deploy their apps for multiple platforms, with the advantage of maintaining a single codebase for all deployed platforms. This improves development time when creating apps and allows developers to reach out to as many users as possible by making their app accessible to all platforms.

This approach is not without its disadvantages,