Group 10's Usability Testing

Testing:

We conducted a Heuristic Evaluation with 3 evaluators from our development team, below are their evaluations:

1. Visibility of System Status:
    a. The app ran smoothly, with no actions taking time that would make me wonder whether the app froze or not.
    b. The app occasionally took a bit too long while searching locations, however, this was rare and did not impact my experience much.
    c. No issues with any indicators, the app's current state was clear at all times.

2. Match to Real World:
    a. The app uses familiar weather-related vocabulary and icons, such as the magnifying glass for searching.
    b. I'd like to see a compass showing the wind direction.
    c. The concept of a 'Home' feels weird in the context of a weather app.

3. User Control and Freedom:
    a. The home button is always visible, making it easy to return to the home screen.
    b. I can always use the home button or just quit the app and relaunch it.
    c. N/A

4. Consistency and Standards:
    a. Finding all icons in the bottom bar rather than in a side burger menu makes the app feel a little dated, but ultimately it is just a matter of preference.
    b. The design language is very clear and consistent.
    c. The app is consistent, however I found two typos in the UI.

5. Error Prevention:
    a. The app suggests locations as a form of autofill on the search screen.
    b. The app does not let you do anything that could cause an error, however that is just a result of the app not taking many possible inputs from users.
    c. I accidentally deleted a location from my Favourites and had to find it manually again instead of being able to undo the action.

6. Recognition > Recall:
    a. The app communicates clearly, with standard, easy-to-understand icons.
    b. Having the weather symbols in colour rather than plain white would help recognise the weather conditions at first glance.
    c. The 'add' button should be green and the 'remove' button should be red to avoid user confusion and improve recognition.

7. Flexibility for Efficient Use:
    a. The app allows users to bookmark their favourite locations.
    b. Since this is a simple mobile app, there are no complex inputs that the power-users would need to speed up.
    c. There is no such thing as an 'expert user' for a simple weather app.

Group 10's Usability Testing

8. Aesthetic and Minimalist Design:
    a. The design shows you just what it thinks you need, nothing more, nothing less. However, I would appreciate at least a bit of customizability.
    b. The home screen shows you the most important metrics (although it doesn't allow customisation), if you need anything else, you just scroll down and get the details.
    c. Very lightweight interface.

9. Error Recovery
    a. If your location cannot be found, the app informs you that it could not find anything.
    b. The search error screen should tell me to check for typos, I accidentally made one when searching for a location without and got confused when I was told that "we don't know where that is".
    c. The search results clearly inform you if your query is invalid.

10. Help and Documentation:
    a. The app should offer at least some minimal explanation about some of the metrics, e.g. what does "UV index: moderate" mean.
    b. Since this app is a very standard weather app in its core, there is no need for any additional documentation or a manual.
    c. No need for documentation since users are used to using weather apps.

Based on the reports, we made a few minor changes regarding the UI design, such as using more meaningful colours and eliminating typos. We also added a "Recommended" section that appears if the user location search fails. We've also added the option to undo your actions, which is a useful form of error prevention.

Deviations from our lo-fi design:

Our testing of earlier versions found that our animated backgrounds for different weather conditions distracted the users from the actual data, which ultimately made us replace the idea of a dynamic background with a static one picturing a mountain. At the beginning of our implementation, we came to the conclusion that our team had much greater experience with front-end development in React Native, and therefore chose to adopt this, keeping the originally planned Java backend connecting to Open-Meteo API. This allowed us to create the user experience in a more intuitive way, utilising the power of components. React Native apps are also generally considered to be lightweight with smaller load times, which allowed us to improve the user experience.