



Ashley, Daphne, Eliza, Leon

Introduction:

Existing weather forecasts provide tons of information, but everyone perceives temperatures differently. FitCast makes weather forecasts more comprehensive by directly forecasting your outfit instead, sending you personalized outfit suggestions tailored specifically with your body and comfort in mind.

Design tools:

We built our high-fidelity prototype with React Native and Expo. All navigation was done with Stack Navigation and Drawer Navigation through Expo Router. We used Apple's Xcode Simulator to test the app as we developed it.

Operating Instructions:



Download the Expo Go app, and login/create account
Scan this QR code and open the app in expo
Start FitCasting away!

Limitations:

- We do not actually store any user information or data in the backend, so we can't iterate on user feedback and fully factor in user preferences.
- We also don't ask for user information/preferences as much as we should. This is because most functionality is currently hard coded, so allowing the user to input/update their preferences would make the hard coded information

mismatch what is shown.

- In this version, the logging modals do not pop up at the right times. The idea is that a modal pop up when the user moves locations significantly, and when the time passes 9pm. In order to showcase how these pop ups work, we have them hardcoded to randomly appear, rather than the intended functionality.
- In this version, we have not integrated real weather data or real suggestions. Since weather only changes over longer periods of time, we would not be able to showcase the full potential of our app without hardcoding some weather events. That being said, we have actually implemented a **fully functional version of the FitCast app which fetches real time weather data from a weather API, and generates a true FitCast.** See that version by scanning the QR code.



Wizard of Oz Techniques:

Randomized Pop-ups

Our prototype randomly decides when and if to display a suggestion log/location log pop up. Since their true triggers (moving locations/being nighttime) are unlikely to happen while testing the app itself, we currently have it set up to randomize whether something pop ups and which log pops up.

Hard Coded Features for this Technique:

- Pop ups are randomly decided
- Clothing icons on location log → would usually be adapted to user needs/preferences
- Map location is set to Stanford d.school

AI Capabilities

Our prototype magically reveals clothing suggestions that a frequent user might see if they have already calibrated how they feel around these weather conditions. This is as we have hard coded a user's weather and clothing preferences.

Hard Coded Features for this Technique:

- User information → necessary so the rest of the hard coding aligns
- User weather preferences -> all interpretations of the weather based on

- 'previous' user interactions with the app
- Suggestions themselves → the text and the icons

Live Weather Updates

Our prototype simulates specific weather conditions in a hard-coded area. Upon opening the app, the user is shown simulated weather conditions in a hard-coded city, as well as hard coded weather statistics.

Hard Coded Features for this Technique:

- Current location
- Current local weather/statistics
- Current time
- Blocks of temperature and weather change

*Note, in the fully functional API fetched version of FitCast, the live weather updates are not hardcoded.