Interaction Design: Weather App

Task 2: Design

1. Brainstorming concepts and metaphors

Concepts:
Conceptual Design:

- ➔ Provide the current temperature
- ➔ Provide basic weather information (sunny, windy, rainy)
- ➔ Provide some means to teach children about weather to a limited extent (such as comparing temperature, activities to do in weather)
- ➔ Provide a basic list of activities to do in each weather
- ➔ Give a short weather forecast (up to ±3 days)
- ➔ Implement privacy-respecting measures.
- ➔ Display information in symbolic form, with familiar icons from everyday life.
- ➔ Allow customization

‘Field’ Scene / Stage

- Everything bit of information is part of the ‘scene’
- Temperature is on a sign with thermometer
- Settings screen is a rusted cog (for parents use only), clearly labeled so that parents recognize it. (‘Age’-gated -> Simple Maths sum)
- Grass & Sky

Forecast Navigation:

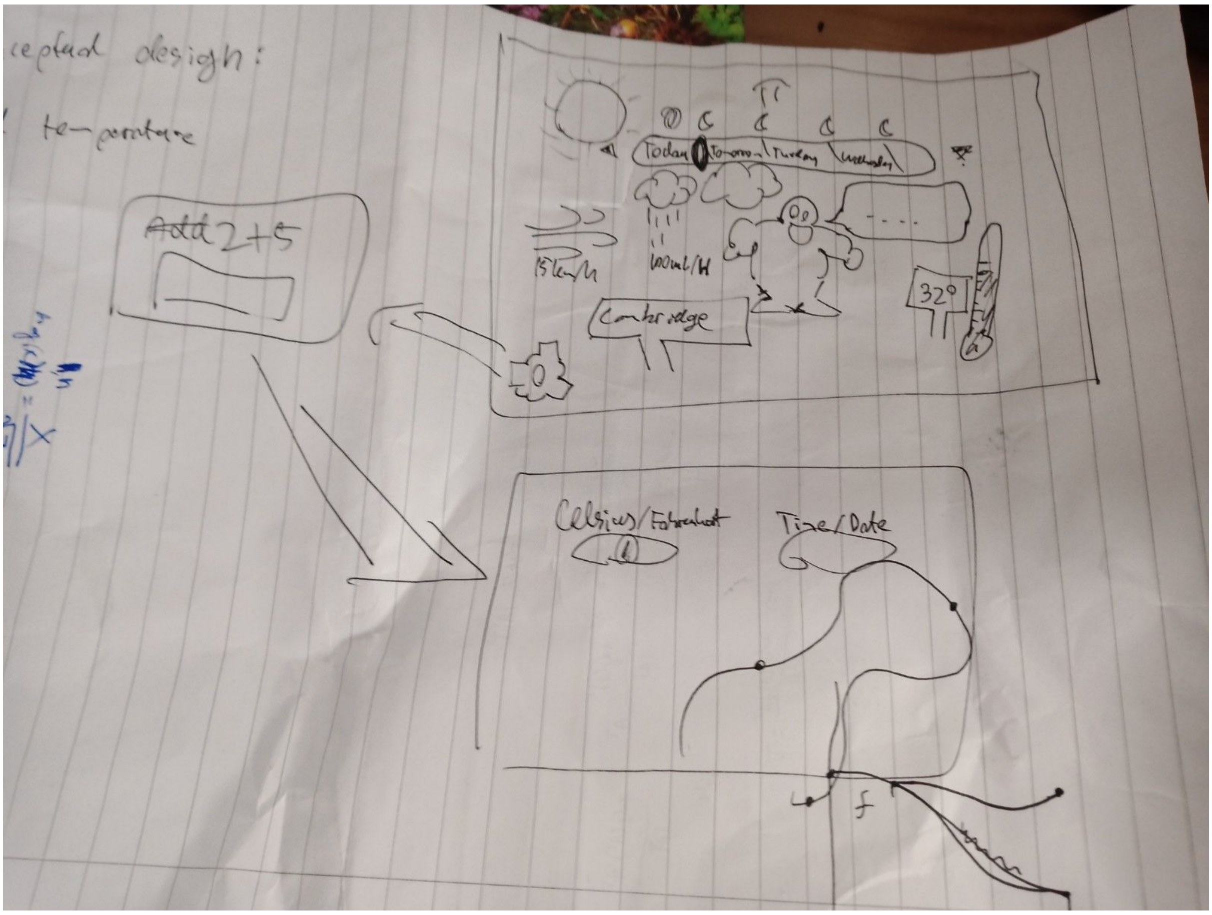
- ‘Slider’-style animations, transition between ‘weathers’ instead of day-night
    - Top Bar (top 15%-ish)
    - Today, Tomorrow, followed by Days of week
    - Up to +4 Days (5 Days in total)
    - Navigate both through sliding and directional buttons

Character / Cartoon:

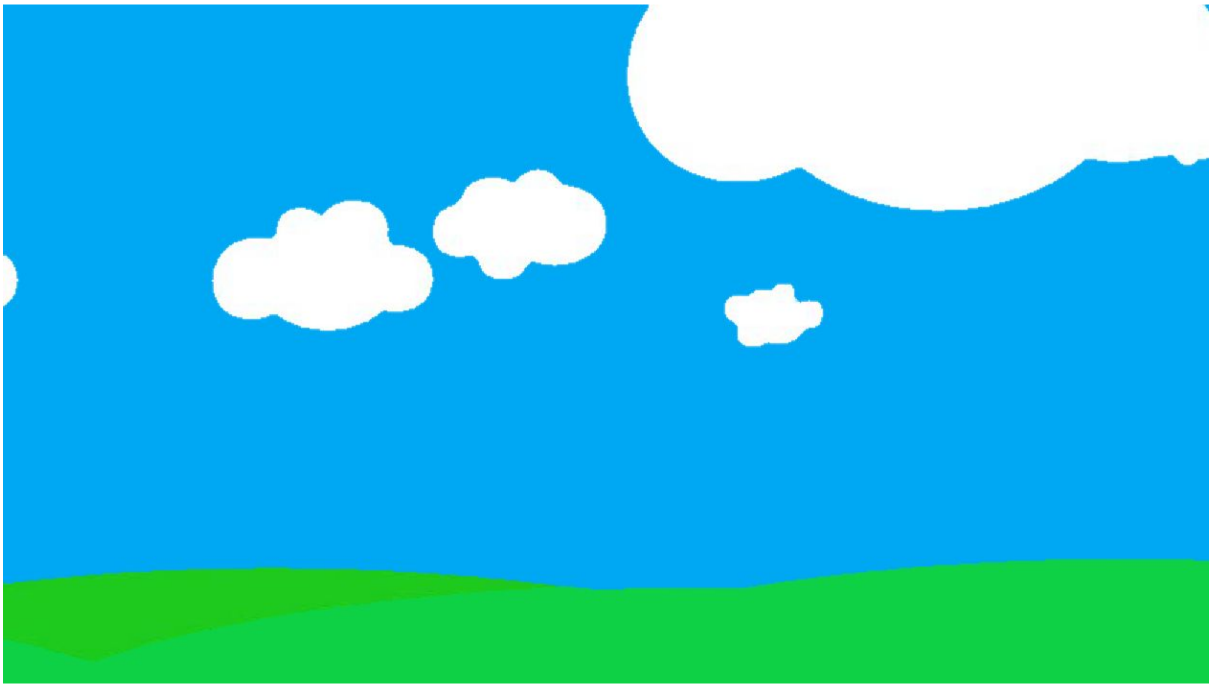
- Take up middle third of screen
- Attire changes according to weather (umbrella hat, shades, hat, coat, etc)
    - Sunny: Shades, Hat
    - Rain: Umbrella Hat
    - Windy: Coat
    - Snow: Skis
    - Cold: Jumper

Interaction

- Every weather object (sun/rain/wind) is tappable and character has an



Initial brainstorming layout.



Example of the comic drawing style that we intend to use.

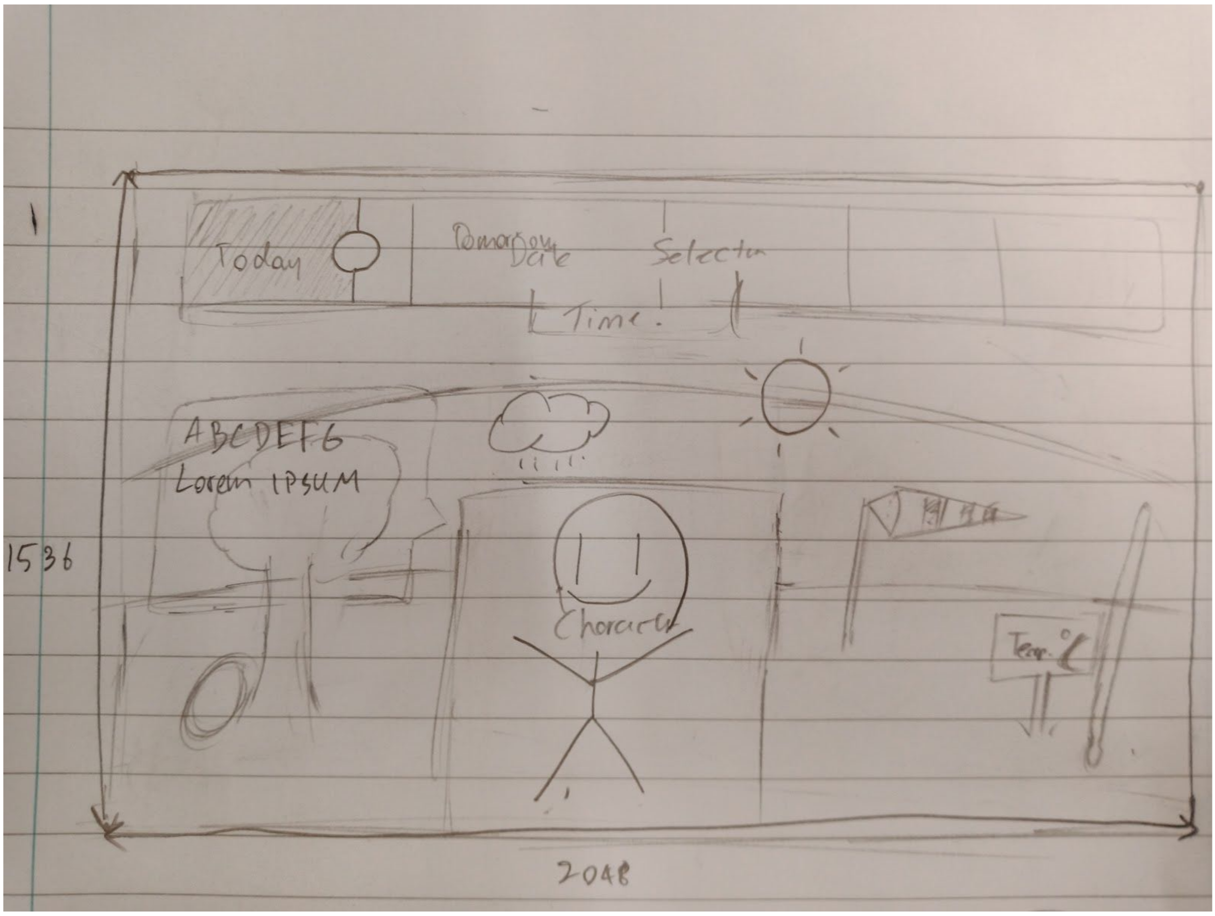
2. Design for phone / tablet

Our questionnaire showed a slightly higher usage rate for phones than tablets. However, the error bound is high, due to the low number of responses. Assuming a roughly equal usage rate, we assume that our decision does not affect the size of the user base. Education is a primary function of our app, and it is easier to implement on bigger screens.

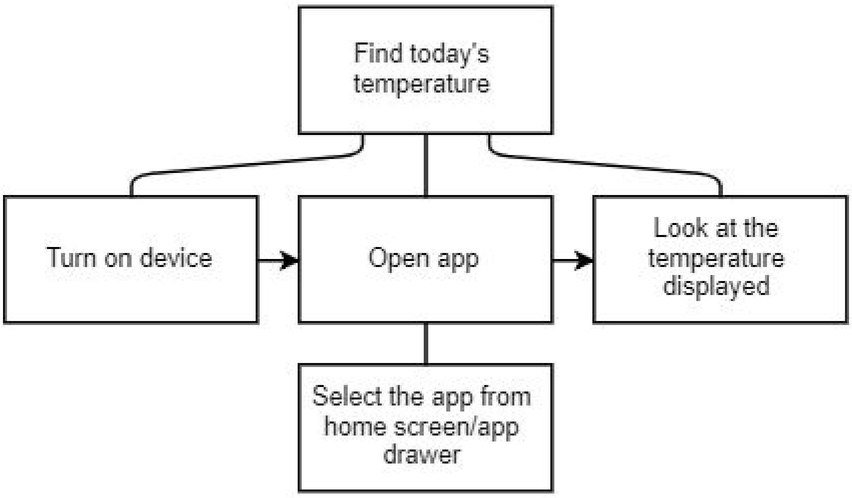
Furthermore, tablets have been used in schools as an education aid. This is an area where phones are less commonly used and are sometimes prohibited because of the distraction risk they pose. Since this is a potential scenario for our application to be used, we will opt for a tablet design.

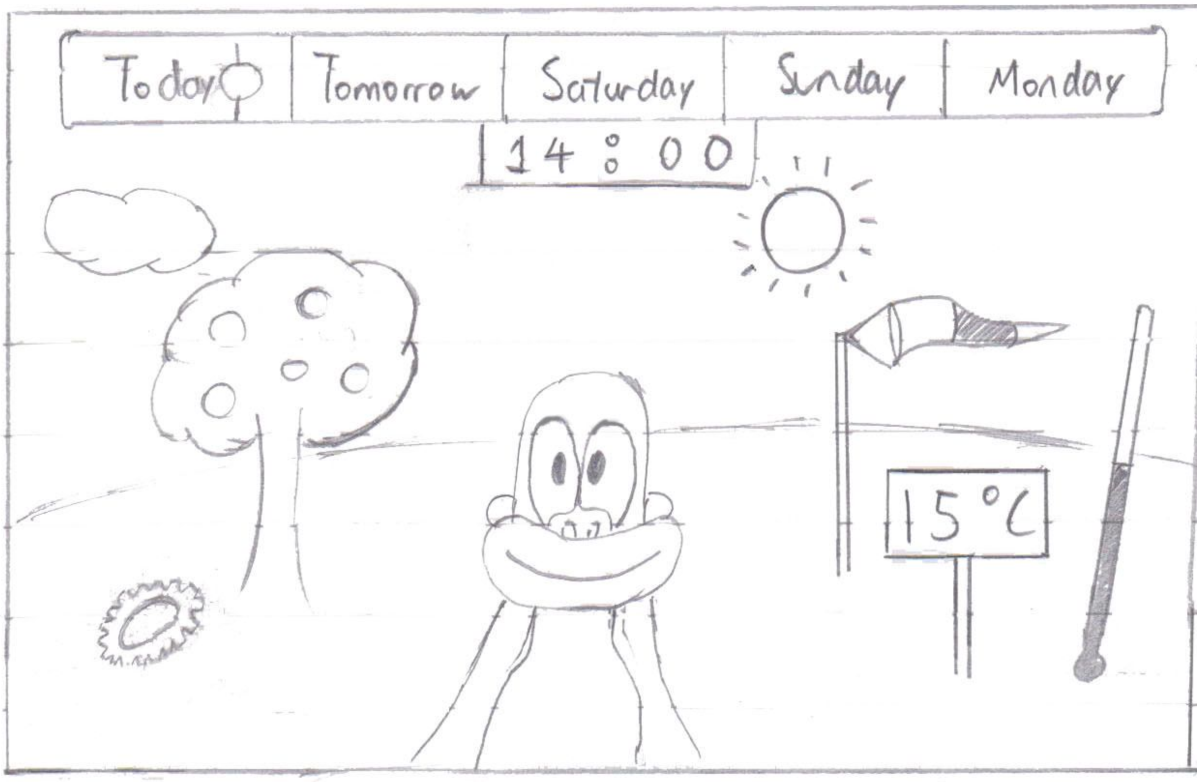
3. Screen Layout & Tasks

Initial screen



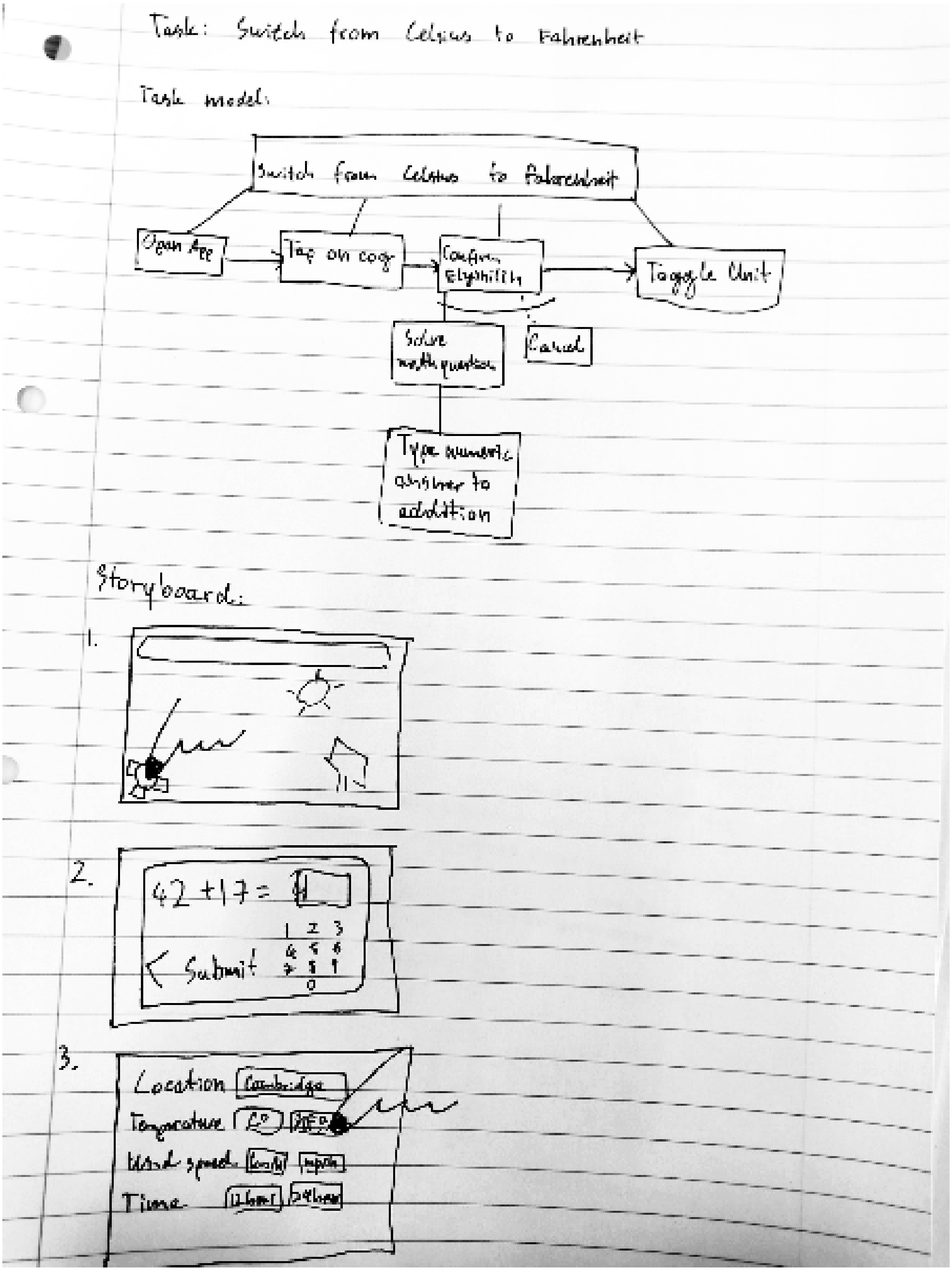
Task 1: Check Temperature


Find today's temperature
Turn on device
Open app
Look at the temperature displayed
Select the app from home screen/app drawer



(1) The user opens the app and can currently read the current temperature from the sign by the thermometer. Without any user intervention, the app will poll the weather API to get updated weather information.

Task 2: Switch from Celsius to Fahrenheit



Task: Switch from Celsius to Fahrenheit

Task model:

Storyboard:

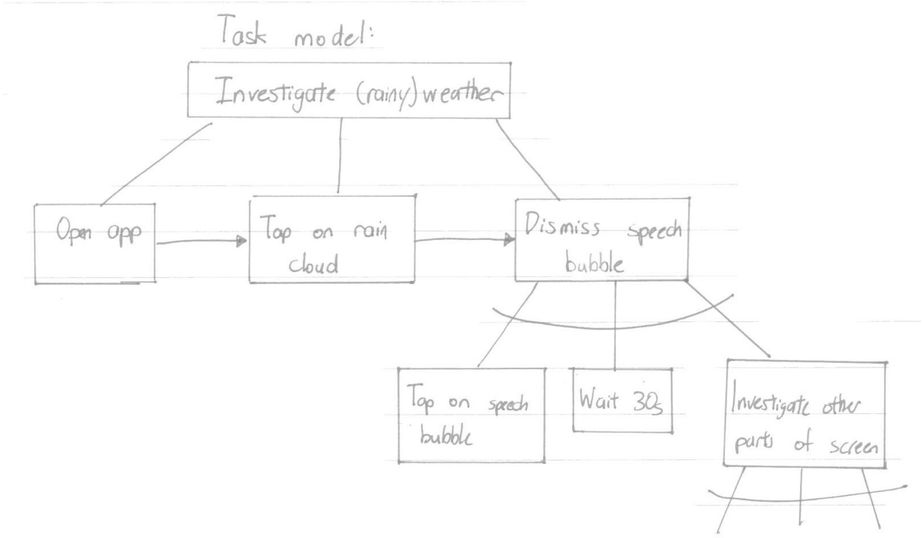
(1) The user opens the app

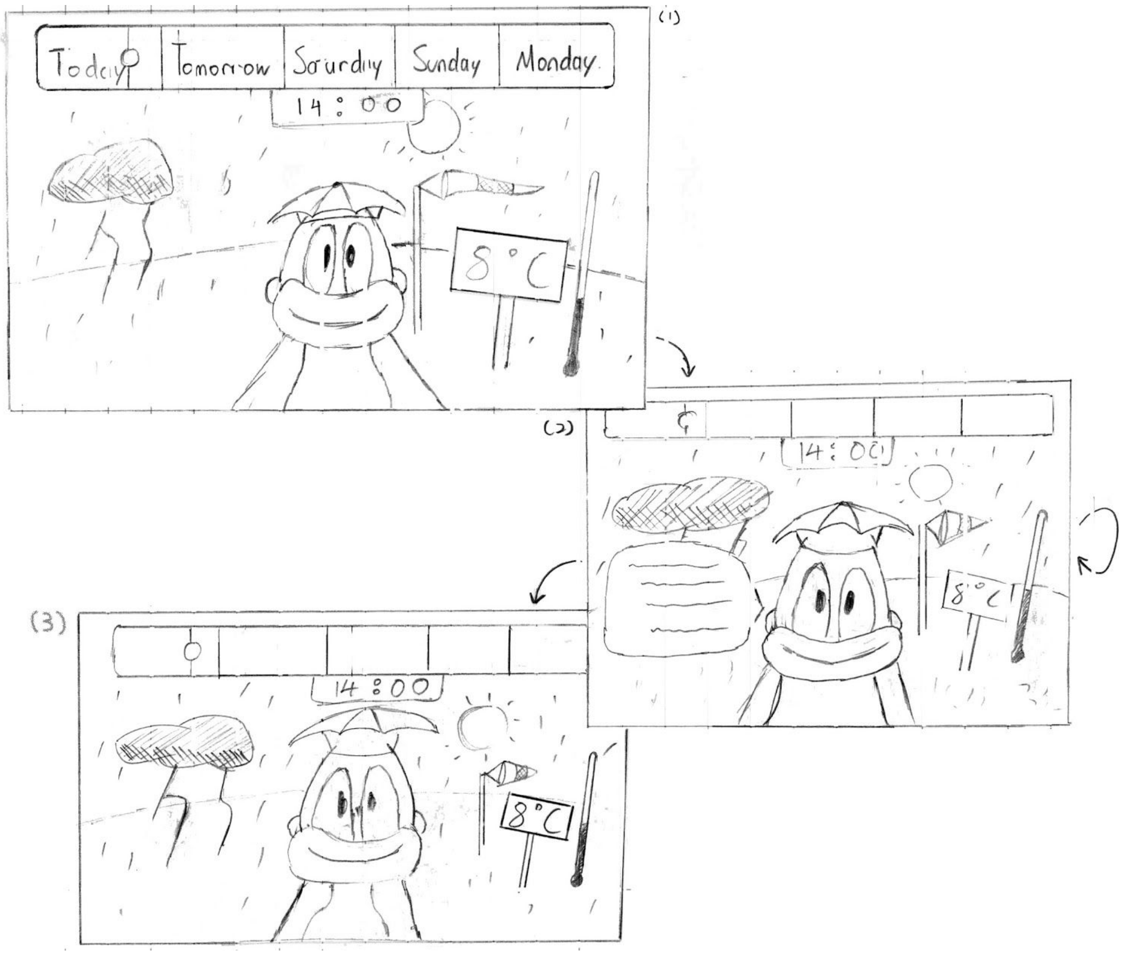
(2) They click on the cog in the lower-left corner to enter the settings menu

(3) To ensure that the user understands the settings that will be provided, the app asks a simple math question. Younger users will then be able to return to the main screen, while older users can answer the question and access the settings menu.

(4) The user toggles the temperature mode to switch between Celsius and Fahrenheit.

Task 3: Investigate rainy weather





(1) The user opens the app, and discovers that it is currently **raining**. Curious about the current weather condition, the user then taps on the rain cloud. This transitions into screen 2.

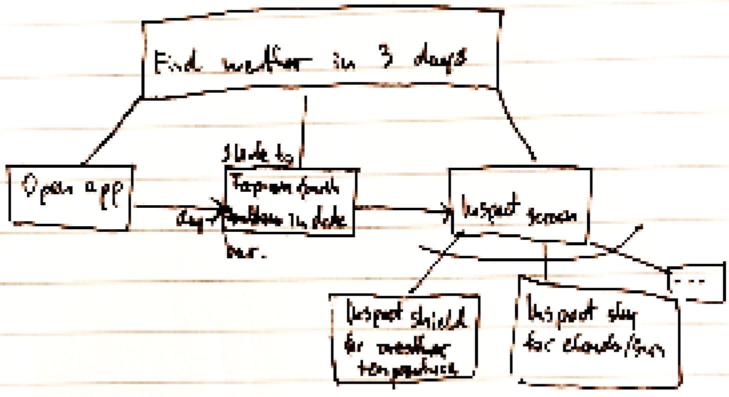
(2) A speech bubble appears to the left of the character, with interesting information about the rain. (e.g. how much rainfall, how long it may last, how it usually gets colder when it rains)

(3) The user is done reading the information. The text box will slowly fade away after 30 seconds (it is assumed that the user can read the text within 30s). Alternatively, the user can tap on the text box to dismiss it. Tapping on another feature (the wind speed, for example), will refresh the text box with information about the other feature.

Task 4: Show the weather in three days from now

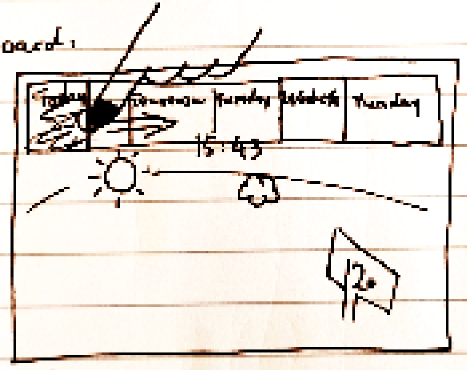
Task: ~~Find +~~ Show the weather in three days:

Task model:

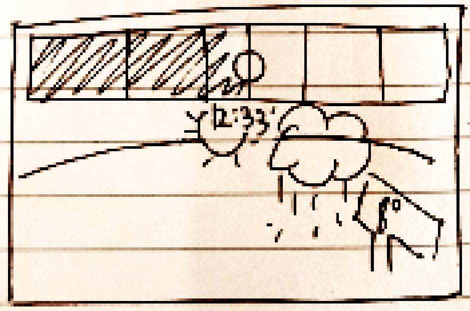

Find weather in 3 days
Open app
Slide to
Tap on fourth ~~button~~ day in date bar.
Inspect screen
Inspect shield for weather temperature
Inspect sky for clouds/Sun
...

Storyboard:

1.



2.



(1) The user opens the app and uses the slider provided on the date bar in the upper screen to slide to the desired date and time

(2) The slider is fully animated, meaning that the displayed weather will change to reflect the predicted weather as the slider is moved.

4. Lo-fi to Hi-fi

The hi-fi design will be implemented as a Java application using the Swing GUI library.

For the backend, we need to be able to read the weather information from the API. Using this, we can construct the relevant weather UI elements as well as update the character's clothing and thermometer.

For the frontend, we must design separate classes for the weather UI elements (sun, clouds, storm clouds etc.). There is a potential risk here of feature creep: the ease of adding new classes could result in too much development time being spent here as well as a cluttered UI if all of those weather elements should be present at once. For this reason, we limit ourselves to only using one weather element at once. (No combinations of fog/snow etc) At this stage we also need to add in animations and speech to our character. This will increase engagement with our audience and can be used as a learning tool.

Task Distribution:

- Charlie (Chieftain Artist) - Developing graphics 1
- Robert Hönig - Developing graphics 2
- Justin Hou - Developing graphics 3
-
- James Hogge - being flakey App Layout
- Jake Hillion - Weather logic