Mobile NAT

August, 2018 | v1.5

NERDERY | Proprietary and Confidential

Mobile NAT - Introduction

For this code challenge, you will be creating a simple weather application named **Umbrella.** The application will download both the current conditions and an hour by hour forecast from Dark Sky. This application is intended to only be released in the United States.

We intend for this to take less than 8 hours. Please don't spend more than 12 hours because, frankly, we value your time. If you think you may go over, we recommend commenting on how you'd accomplish certain tasks. At a minimum, you should have implemented a zip code input, show current and hourly weather conditions, and highlight the high and low temperatures for the day.

Additionally, please locally commit your code changes regularly throughout the challenge.

In this code challenge we will be paying particular attention to the following items:

- **Design fidelity:** Can you accurately and efficiently implement the design as specified in the art comps?

- **Functionality:** Does the application meet the technical requirements and work reliably?

- **Architecture:** How do you structure your application and its classes?
  Would the application be extensible?
  How do you encapsulate data parsing and access?

- **Coding practices and use of IDE:** How do you organize your files and groups?
  What practices do you adhere to make the code accessible and usable to other developers?
  How is source control used within the application?

Mobile NAT - Getting Started

Basics

We created a base project with classes to expedite development. Feel free to not use the classes that we provide, but they're there to help you out.

- **iOS:** https://codechallenge@git.nerdery.com/scm/bravo/ios.codechallenge.git
- **Android:** https://codechallenge@git.nerdery.com/scm/bravo/android.codechallenge.git

u: codechallenge
p: paVuEsarZ3vy3yobjVez

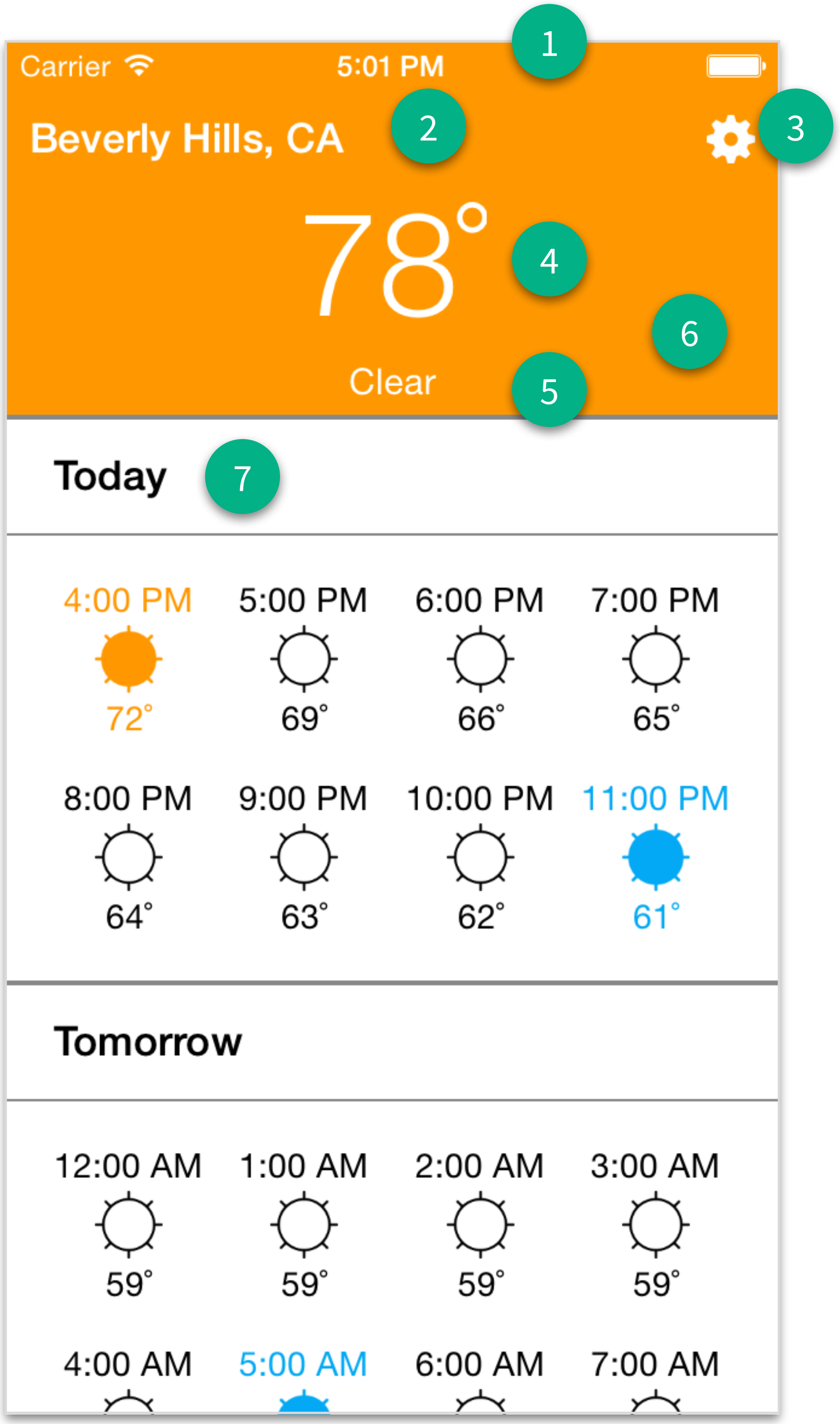
Dark Sky provides free trial access to their API for developers. You can sign up for an API key at: https://darksky.net/dev

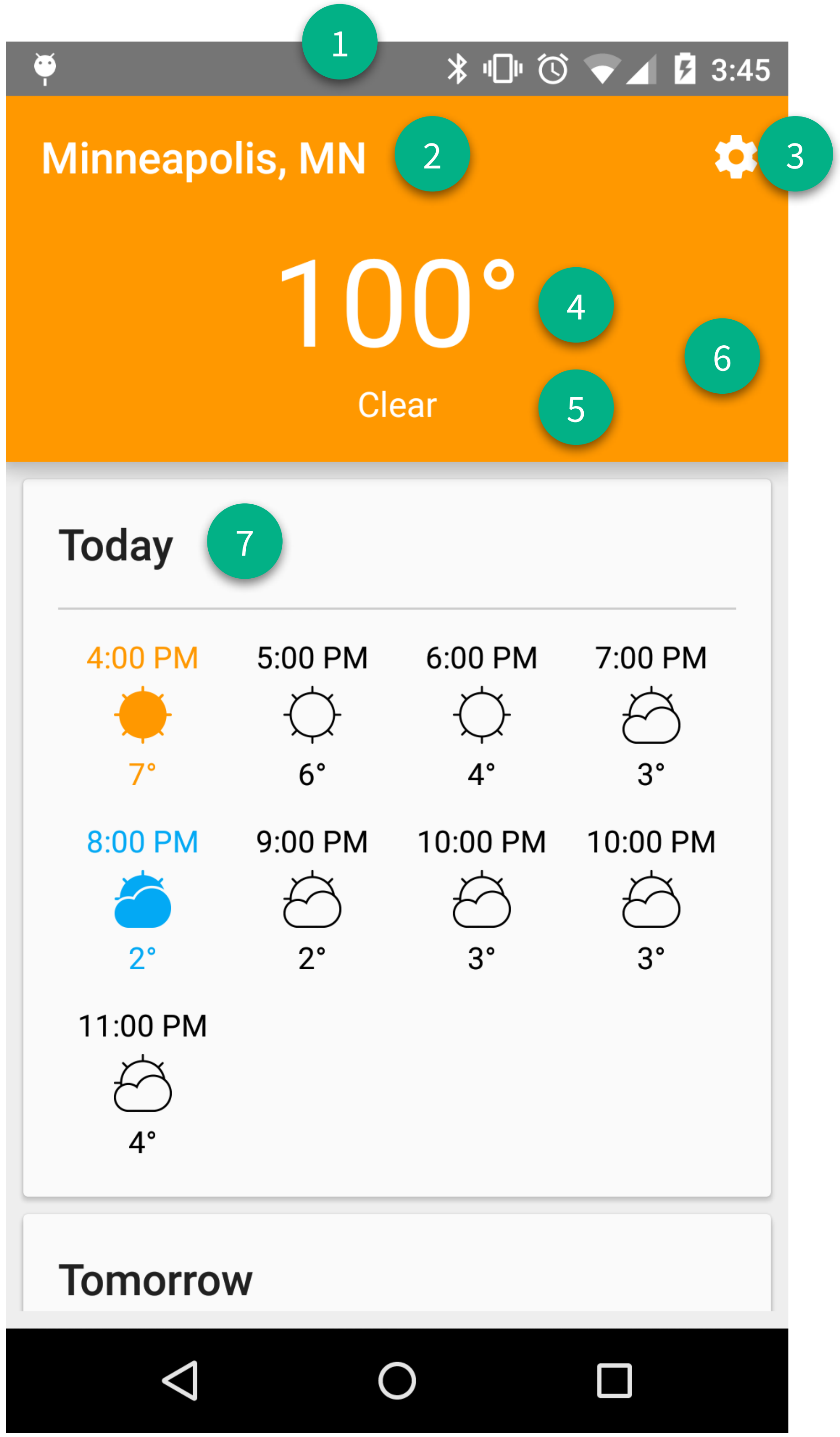
Application Requirements

The application should be built with the latest public SDK and can target the latest public release of the OS.

- **iOS:** Application can be written in either Swift or Objective-C.
- **Android:** Application can be written in either Kotlin or Java.

Mobile NAT - 1.0 Hourly Display





Interactive & UI Notes

1. On the application coming into the foreground:
a. If ZIP code is already entered, refresh the weather data.
b. If a ZIP code has not been provided, or there is an error fetching weather for the zip: direct the user to screen **2.0 Settings**.

2. The application only needs to support weather data for US cities (and their respective states).

3. On click, it displays the **2.0 Settings** screen.

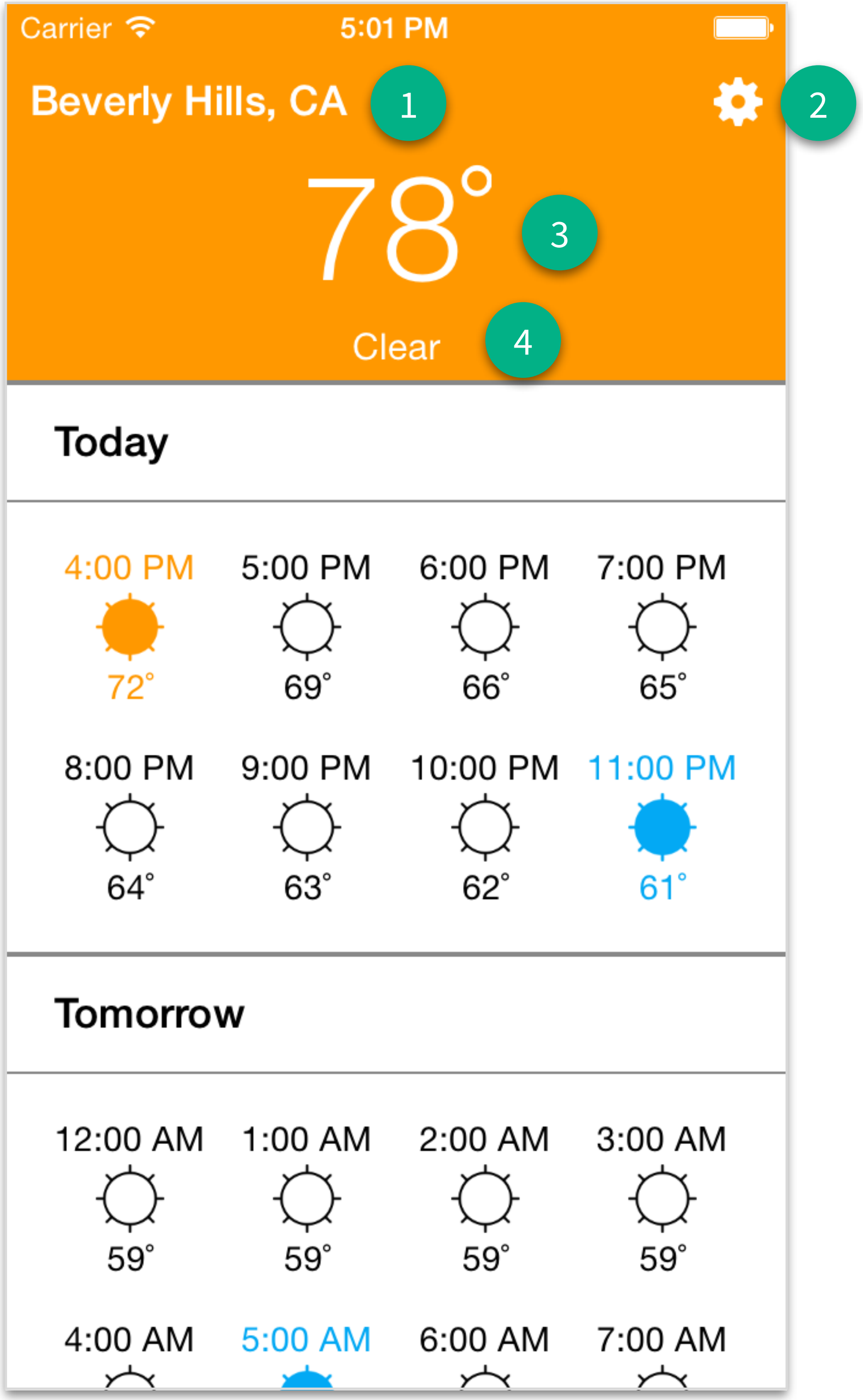
4. Display the current temperature for the selected city and in the selected format, Fahrenheit or Celsius (see **2.0 Settings**), rounded to the nearest whole number.

5. Display a condition summary e.g., “Clear”, “Overcast”.

6. For temperates equal or above 60° F/16° C, use a warm color (0xFF9800) for the background color. For temperatures below, use a cool color (0x03A9F4) as shown in **1.1 Hourly Display (Variation)**. The default color for when there is no ZIP Code entered is the warm color.

7. List the hourly weather for today and tomorrow.
a. Provide ability to scroll.
b. Highlight the lowest (0x03A9F4 color) and highest (0xFF9800 color) temps for each day. If there is a tie for high and low, highlight 1st occurence. If high and low are the same hour, do not use a tint color.

Mobile NAT - 1.0 Hourly Display [cont.]



Beverly Hills, CA

78°

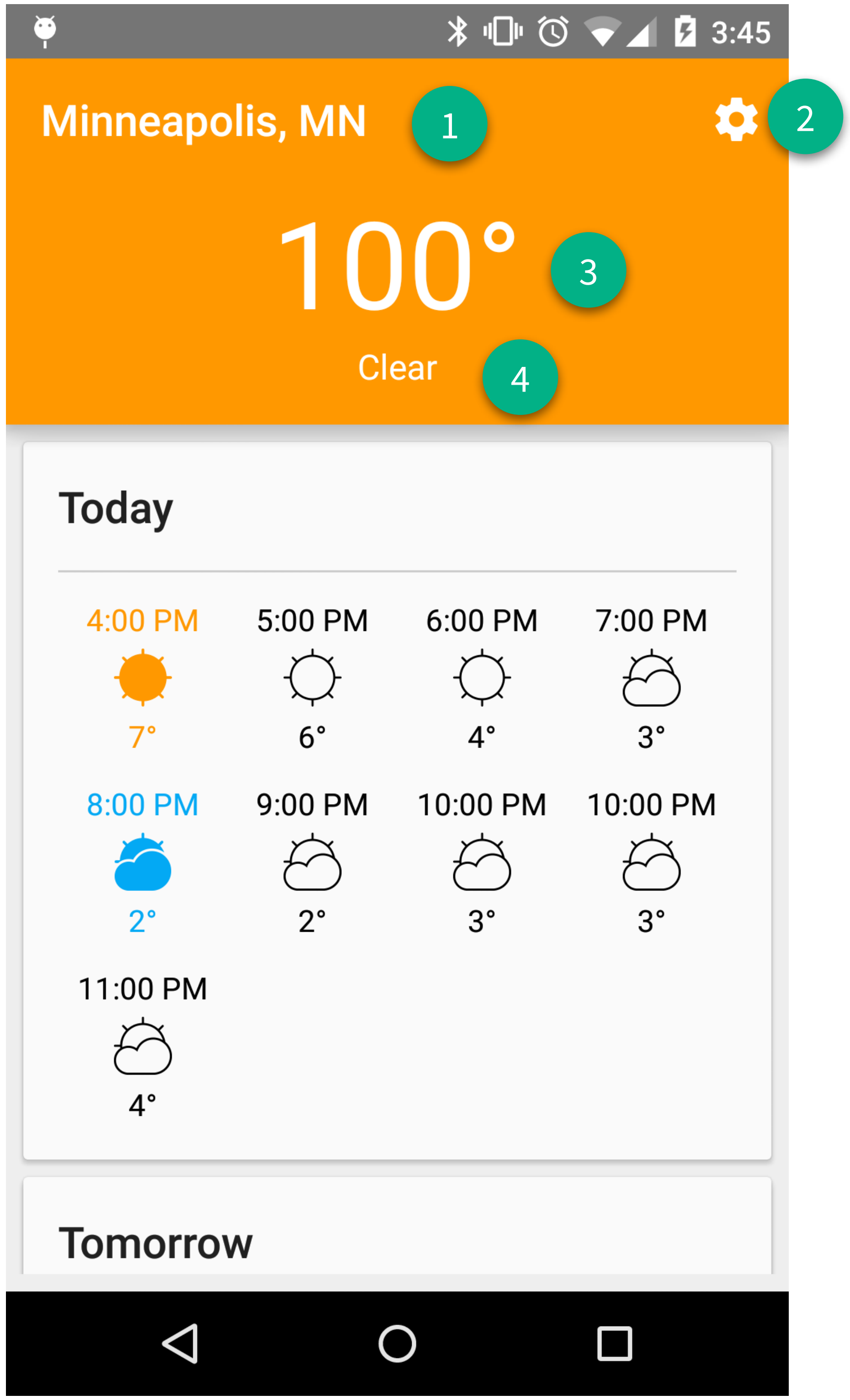
Clear

Today

4:00 PM	5:00 PM	6:00 PM	7:00 PM
72°	69°	66°	65°
8:00 PM	9:00 PM	10:00 PM	11:00 PM
64°	63°	62°	61°

Tomorrow

12:00 AM	1:00 AM	2:00 AM	3:00 AM
59°	59°	59°	59°
4:00 AM	5:00 AM	6:00 AM	7:00 AM



Minneapolis, MN

100°

Clear

Today

4:00 PM	5:00 PM	6:00 PM	7:00 PM
7°	6°	4°	3°
8:00 PM	9:00 PM	10:00 PM	10:00 PM
2°	2°	3°	3°
11:00 PM			
4°			

Tomorrow

Visual Design Notes

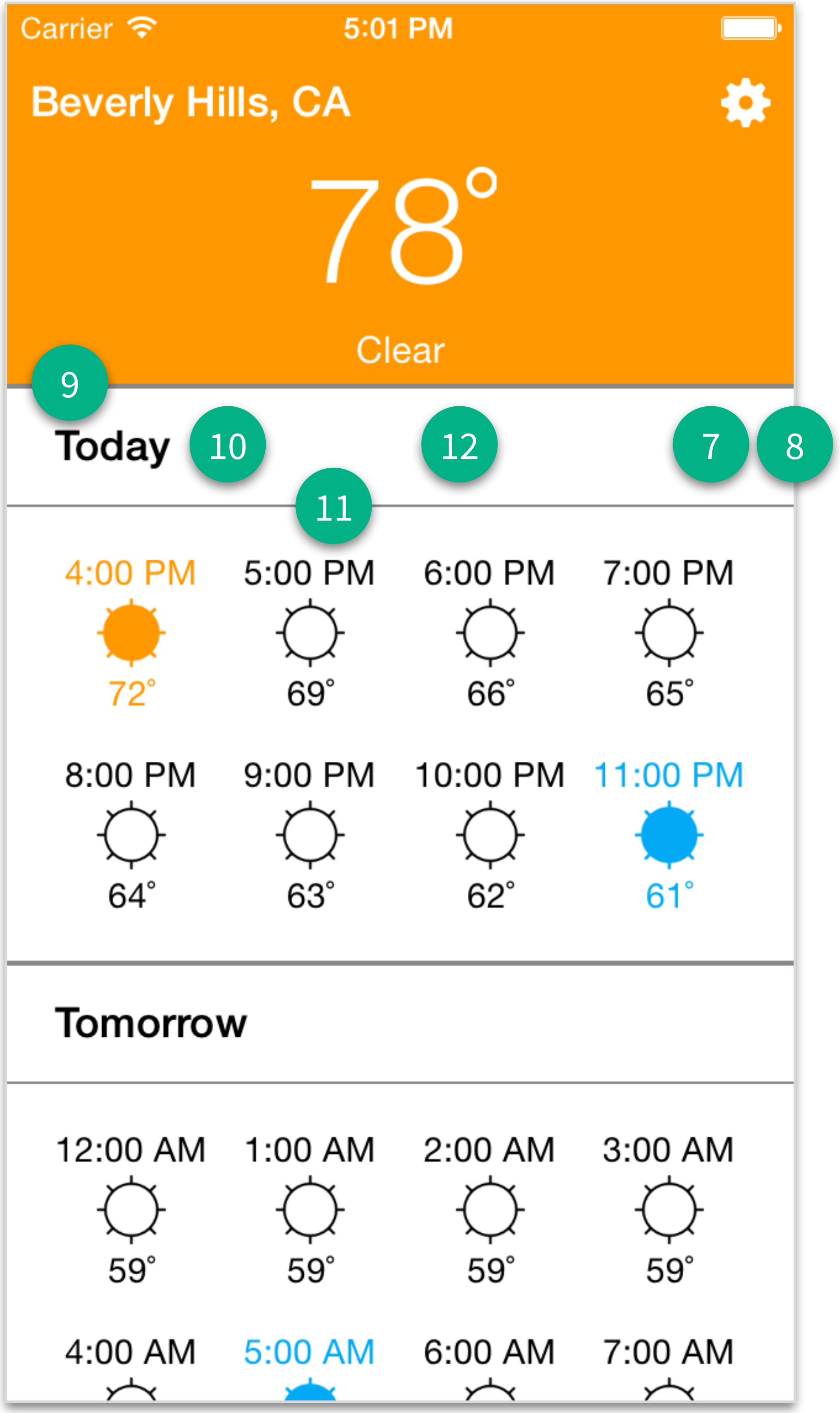
1. iOS: The city/state label should be a headline font. It should be 30 points from the top of the screen. It should be 10 points from left of the screen.
Android: The city/state label should use standard ActionBar title styling.

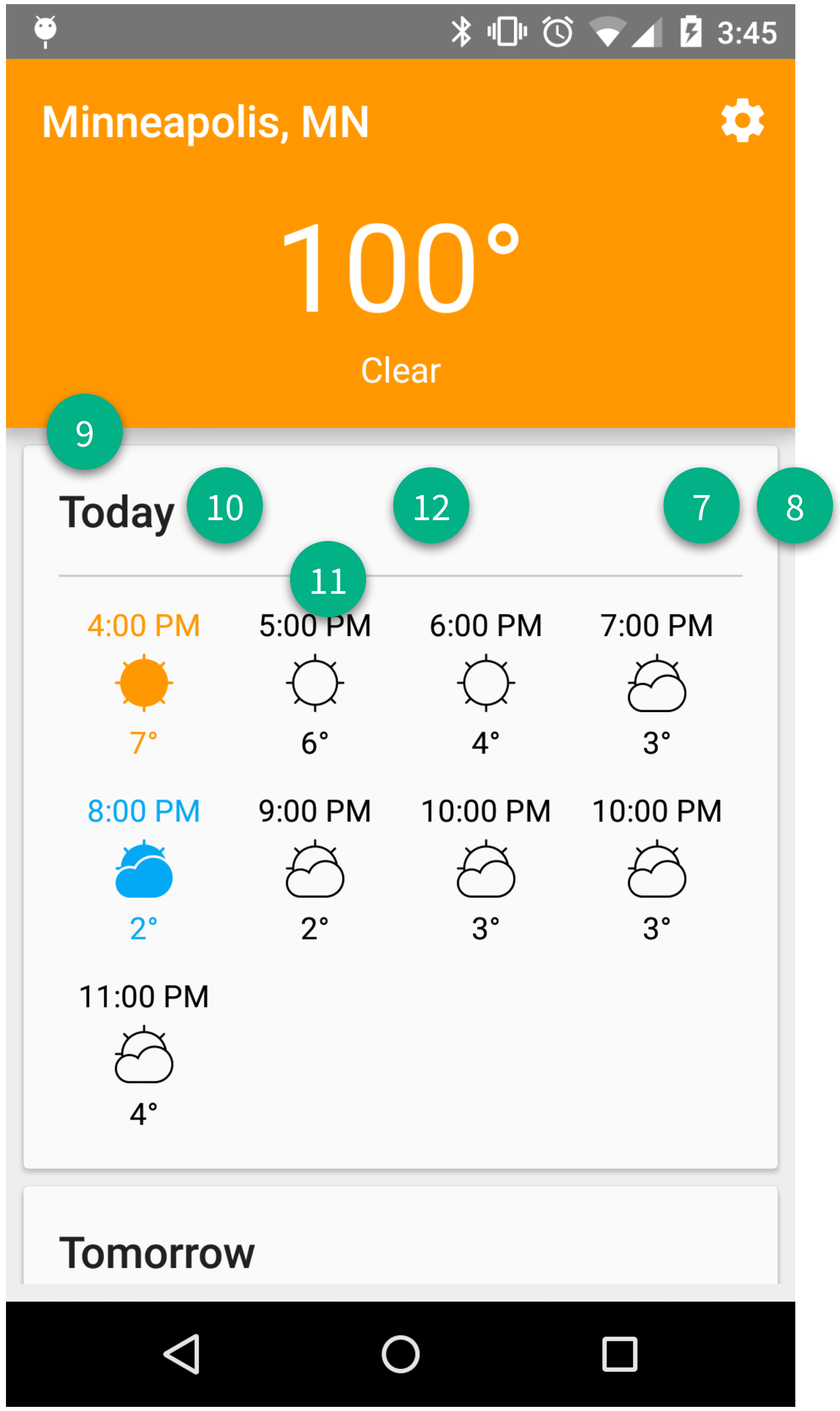
2. iOS: The settings cog (image: settingsCog) should be aligned vertically with the city/state label. It should be 10 points from the right of the screen.
Android: The settings cog (image: ic_settings) should be presented as a standard ActionBar menu item.

3. The current temperature label should be horizontally centered.
iOS: It should be a light weight font at 60 points. It should be 5 points under the city/state label.
Android: It should utilize the "Display 3" text style from the Material design specifications.

4. The current conditions label should be horizontally centered.
iOS: It should be a subheadline, 5 points under the current temperature label. There should be 5 points of space between the bottom of the label and the bottom of the current conditions box.
Android: It should utilize the "Subhead" text style from the Material design specifications. There should be 16dp of space between the bottom of the label and the bottom of the current conditions box.

Mobile NAT - 1.0 Hourly Display [cont. 2]





Visual Design Notes [cont.]

7. iOS: The section insets for the scrollable area is 20 points on all sides. The minimum space for cells is 10 points. The minimum spacing for lines is 20 points.
Android: The section insets for the scrollable area is 8dp on all sides. There should be an 8dp gutter between each card. Card content should be inset by 16dp.

8. The scrollable area should take the remaining space that the current conditions box doesn't use.

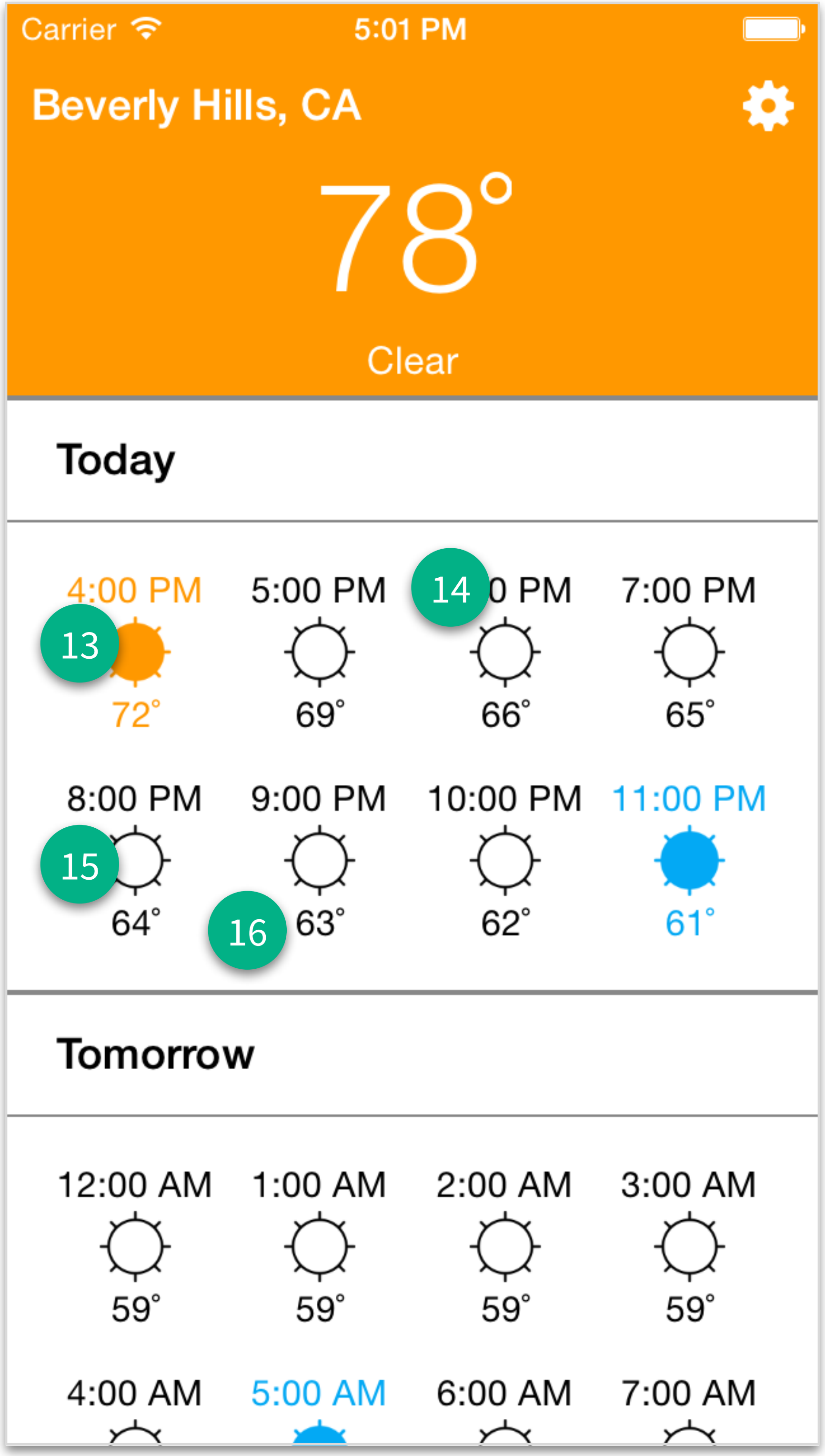
9. iOS: The top of the daily group has a 2 point tall line. It should span the full width of the group. It should be 0x888888 color.
Android: Forecast cards should follow the Material design specifications for cards, including card shadows and corner radii.

10. The day header label should be the relative date where applicable (today, tomorrow, etc).
iOS: It should be 20 points left of the scrollable area and centered vertically. It should be a headline and black.
Android: It should be black and use the "Title" text style form the Material design specifications.

11. iOS: The line under the day header label should be 1 point tall and full width. It also should be 0x888888 color.
Android: The line beneath the day header is a 1dp tall divider that spans the width of the card (excluding card padding). It should be 0xCCCCCC.

12. iOS: The height of the section header should be 50 points.
Android: There should be 16dp of spacing between the divider line and the hourly cells.

Mobile NAT - 1.0 Hourly Display [cont. 3]



Carrier 5:01 PM

Beverly Hills, CA

78°

Clear

Today

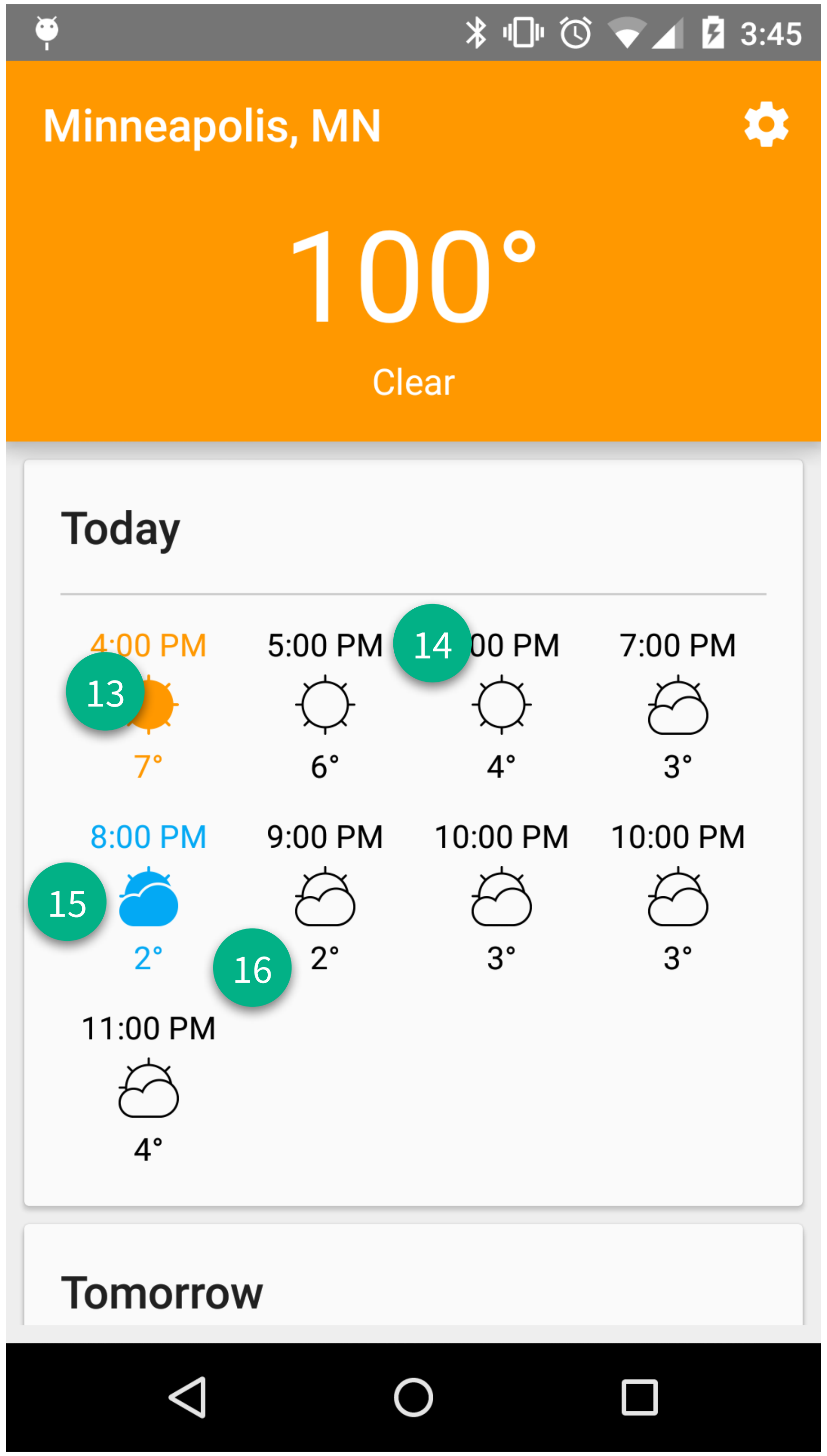
4:00 PM 72° | 5:00 PM 69° | 6:00 PM 66° | 7:00 PM 65°

8:00 PM 64° | 9:00 PM 63° | 10:00 PM 62° | 11:00 PM 61°

Tomorrow

12:00 AM 59° | 1:00 AM 59° | 2:00 AM 59° | 3:00 AM 59°

4:00 AM | 5:00 AM | 6:00 AM | 7:00 AM



3:45

Minneapolis, MN

100°

Clear

Today

4:00 PM 7° | 5:00 PM 6° | 6:00 PM 4° | 7:00 PM 3°

8:00 PM 2° | 9:00 PM 2° | 10:00 PM 3° | 10:00 PM 3°

11:00 PM 4°

Tomorrow

Visual Design Notes [cont. 2]

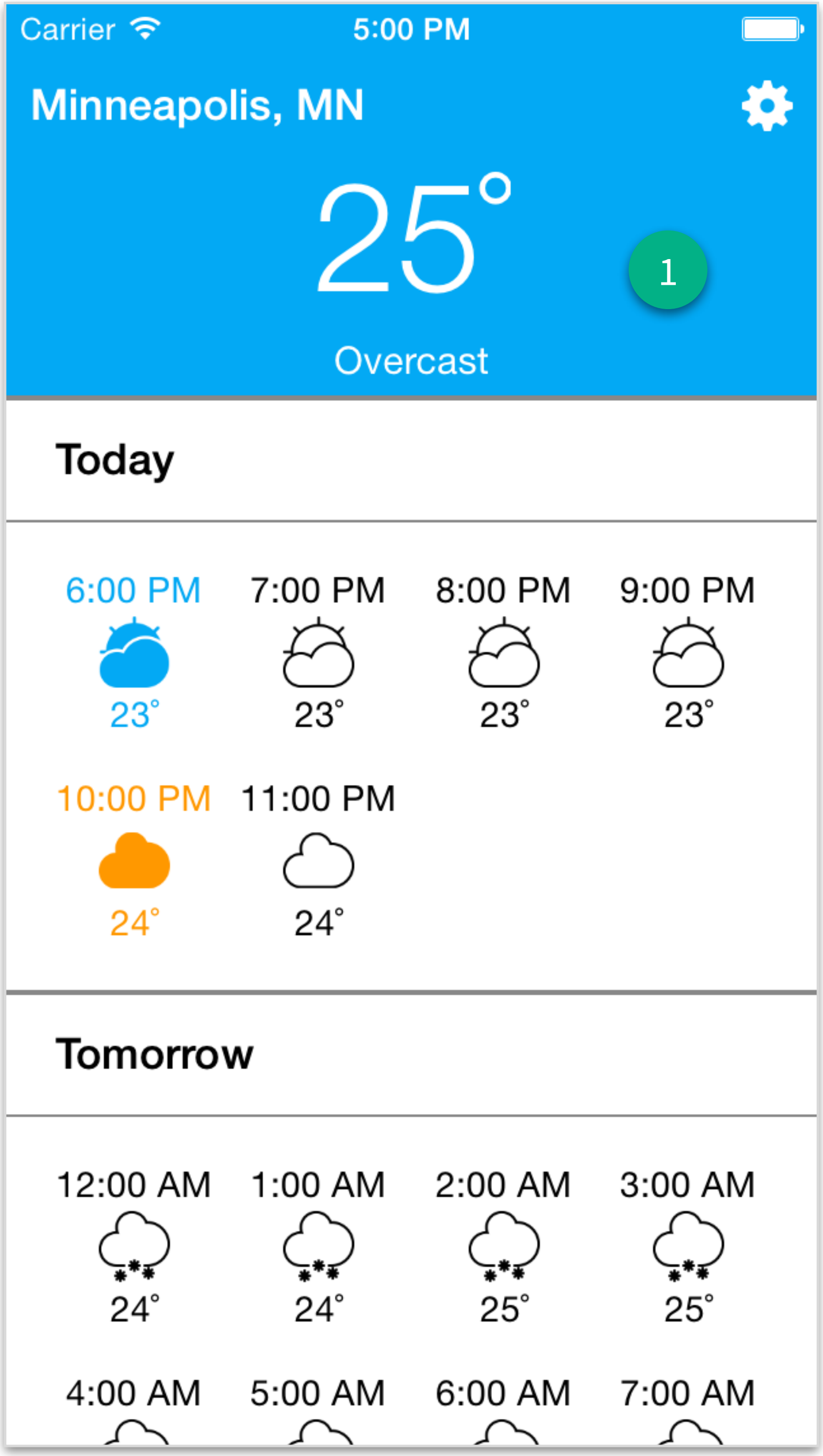
13. iOS: The weather cells should be 62 points x 62 points.
Android: Cells should wrap their contents' width and height. There should be a 12dp gutter between rows of cells.

14. The weather cell time should be a short style (e.g. 12:00 AM).
iOS: It should be 12 point system font and 2 points above the icon.
Android: It should utilize the "Body 1" style from the Material design specifications.

15. Since Dark Sky doesn't provide icon images, included in the checkout is a class that will provide the remote URL string for the weather icon. It should not be scaled or stretched.
iOS: It should be 28 points by 28 points and be vertically and horizontally centered in the cell.
Android: It should have 4dp of spacing on all sides.

16. The temperature should be horizontally centered. It should always be rounded to a whole number.
iOS: It should be 12 point system font and 2 points below the icon.
Android: It should utilize the "Body 1" style from the Material design specifications.

Mobile NAT - 1.1 Hourly Display (Variation)



Carrier 5:00 PM

Minneapolis, MN

25°

1

Overcast

Today

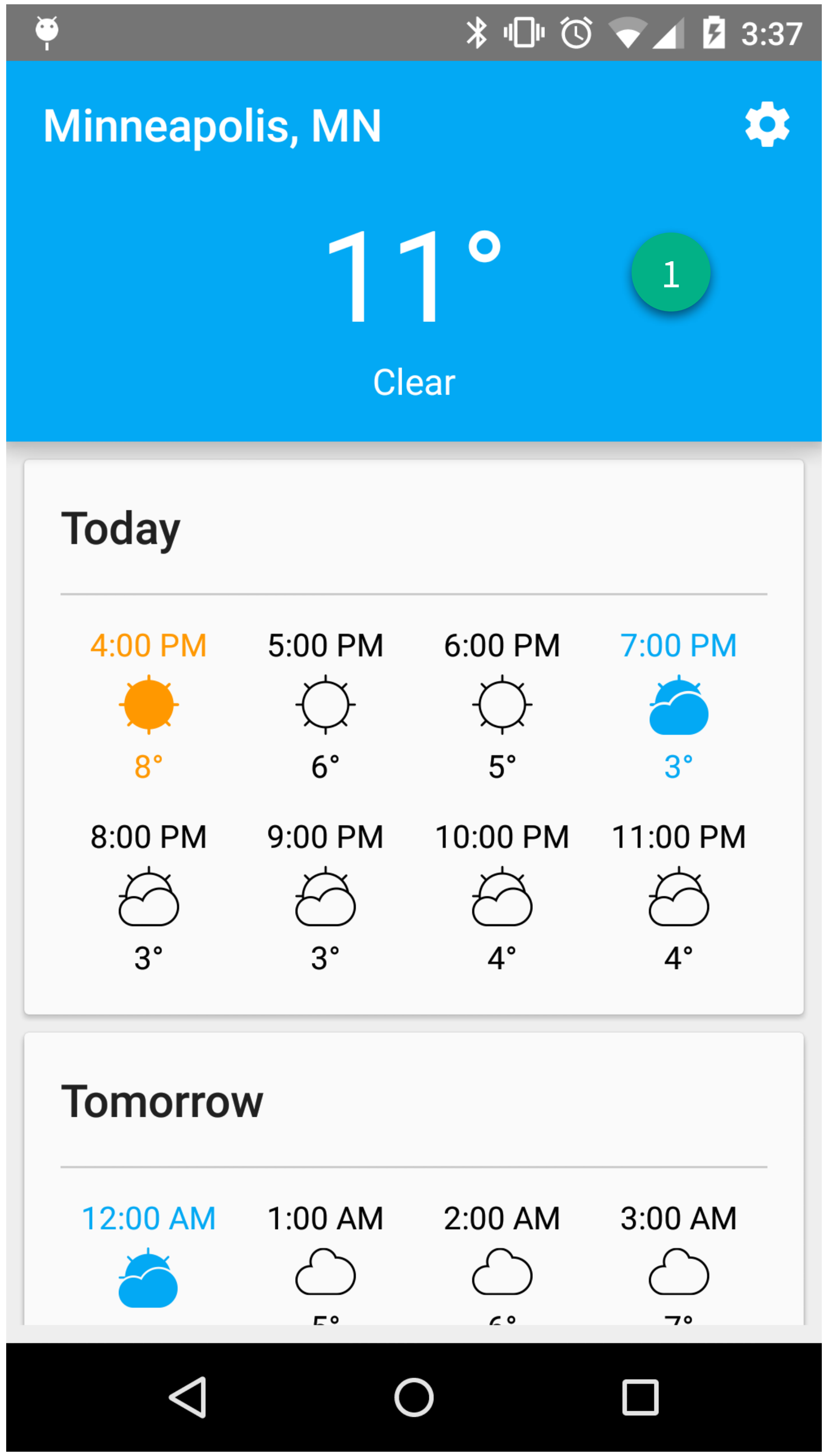
6:00 PM 23° | 7:00 PM 23° | 8:00 PM 23° | 9:00 PM 23°

10:00 PM 24° | 11:00 PM 24°

Tomorrow

12:00 AM 24° | 1:00 AM 24° | 2:00 AM 25° | 3:00 AM 25°

4:00 AM | 5:00 AM | 6:00 AM | 7:00 AM



3:37

Minneapolis, MN

11°

1

Clear

Today

4:00 PM 8° | 5:00 PM 6° | 6:00 PM 5° | 7:00 PM 3°

8:00 PM 3° | 9:00 PM 3° | 10:00 PM 4° | 11:00 PM 4°

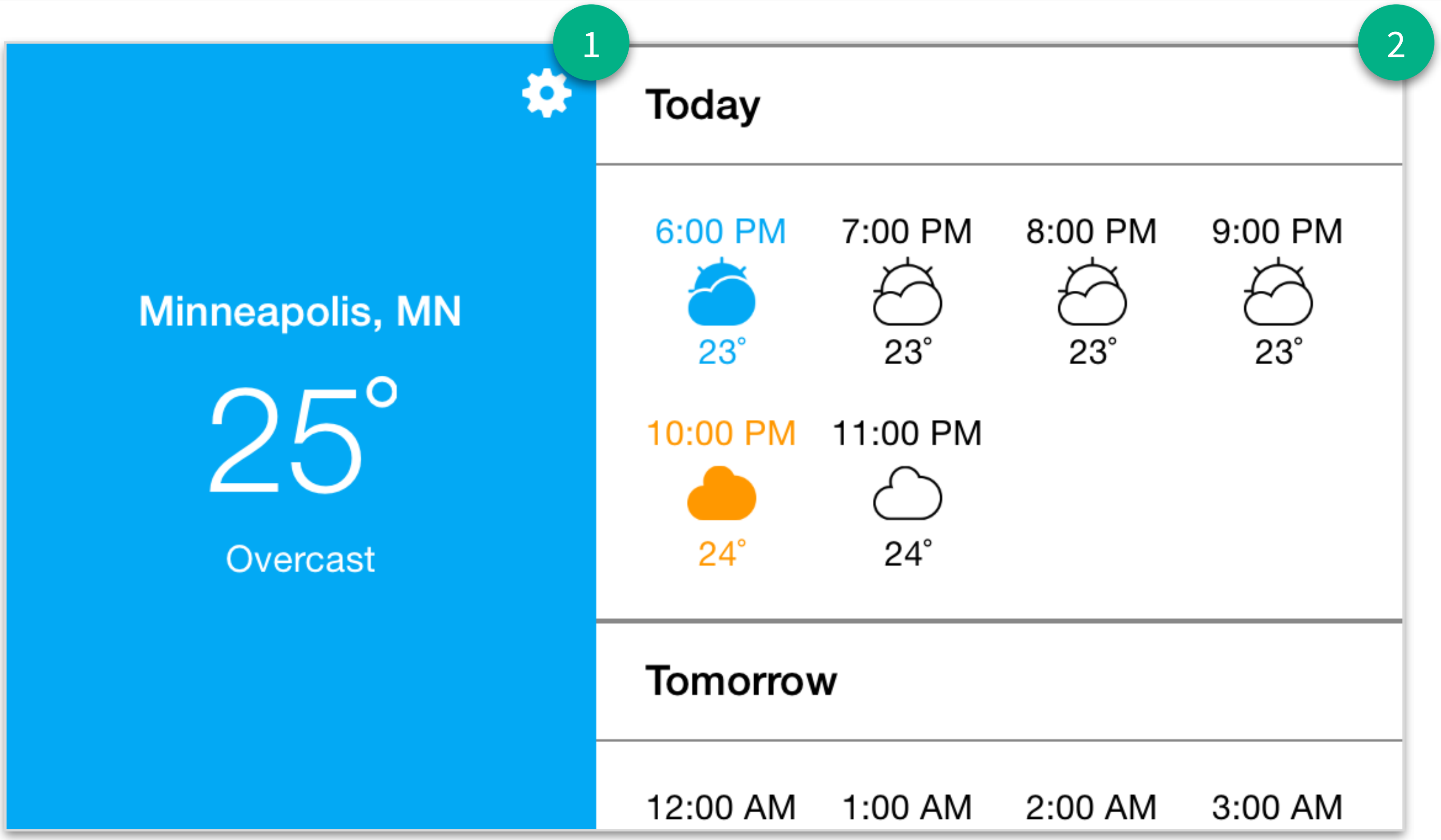
Tomorrow

12:00 AM | 1:00 AM 5° | 2:00 AM 6° | 3:00 AM 7°

Interactive & UI Notes

1. Note how the background color is the cool color (0x03A9F4) because the temperature is below 60° F/16° C.

Mobile NAT - 1.2 Hourly Display (Landscape)



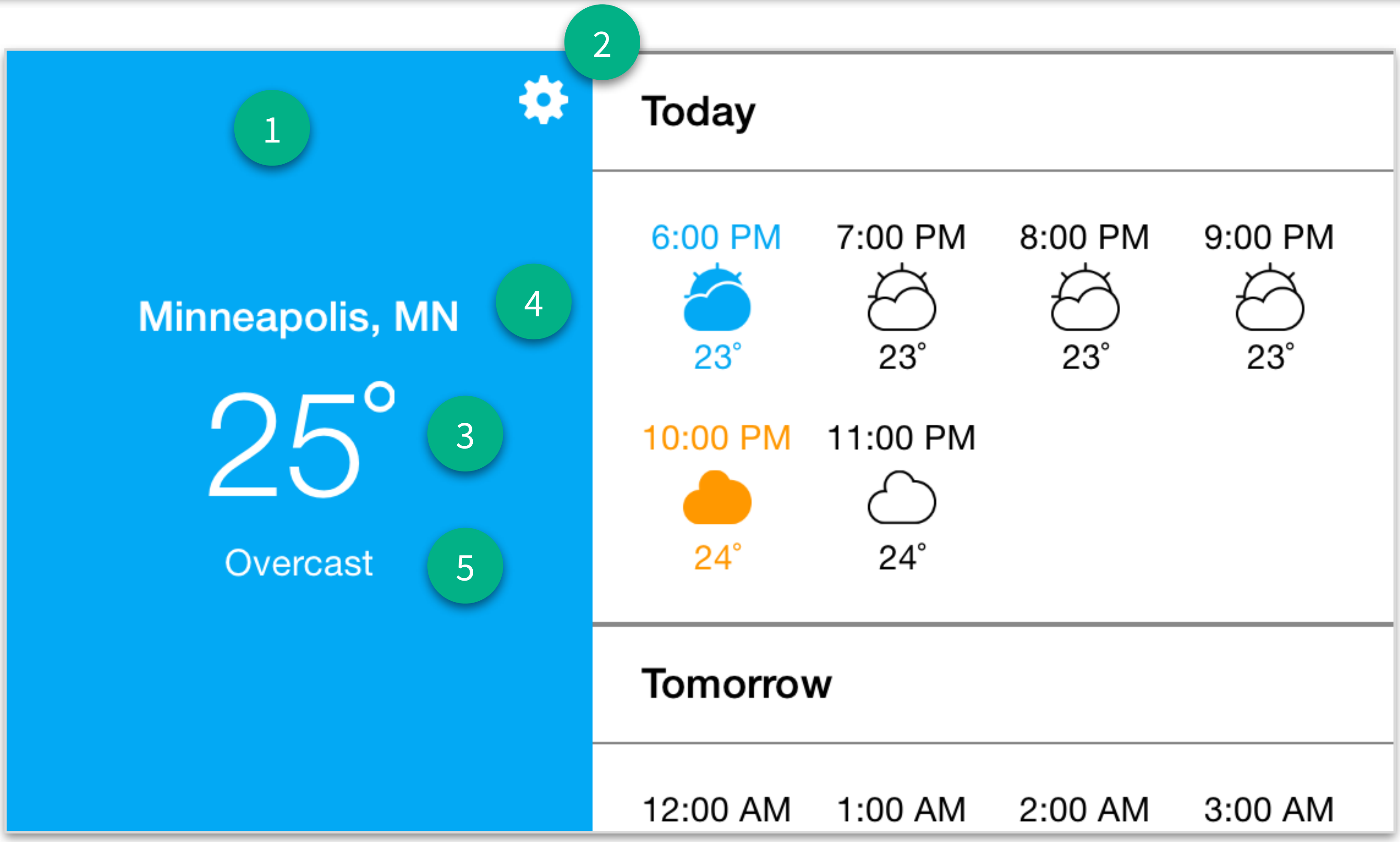
The Android NAT has no specific design requirements for landscape, but it is expected that the app maintains stability across rotations.

Interactive & UI Notes

1. iOS: Reflow top and bottom sections to be left-to-right aligned. Center align the city, current temperate and weather condition.

2. iOS: Reflow hourly data to the right. Still enable scrolling of the section vertically, while keeping the left section fixed.

Mobile NAT - 1.2 Hourly Display (Landscape) [cont.]



The Android NAT has no specific design requirements for landscape, but it is expected that the app maintains stability across rotations.

Visual Design Notes

1. iOS: The background box should always be 240 points wide.

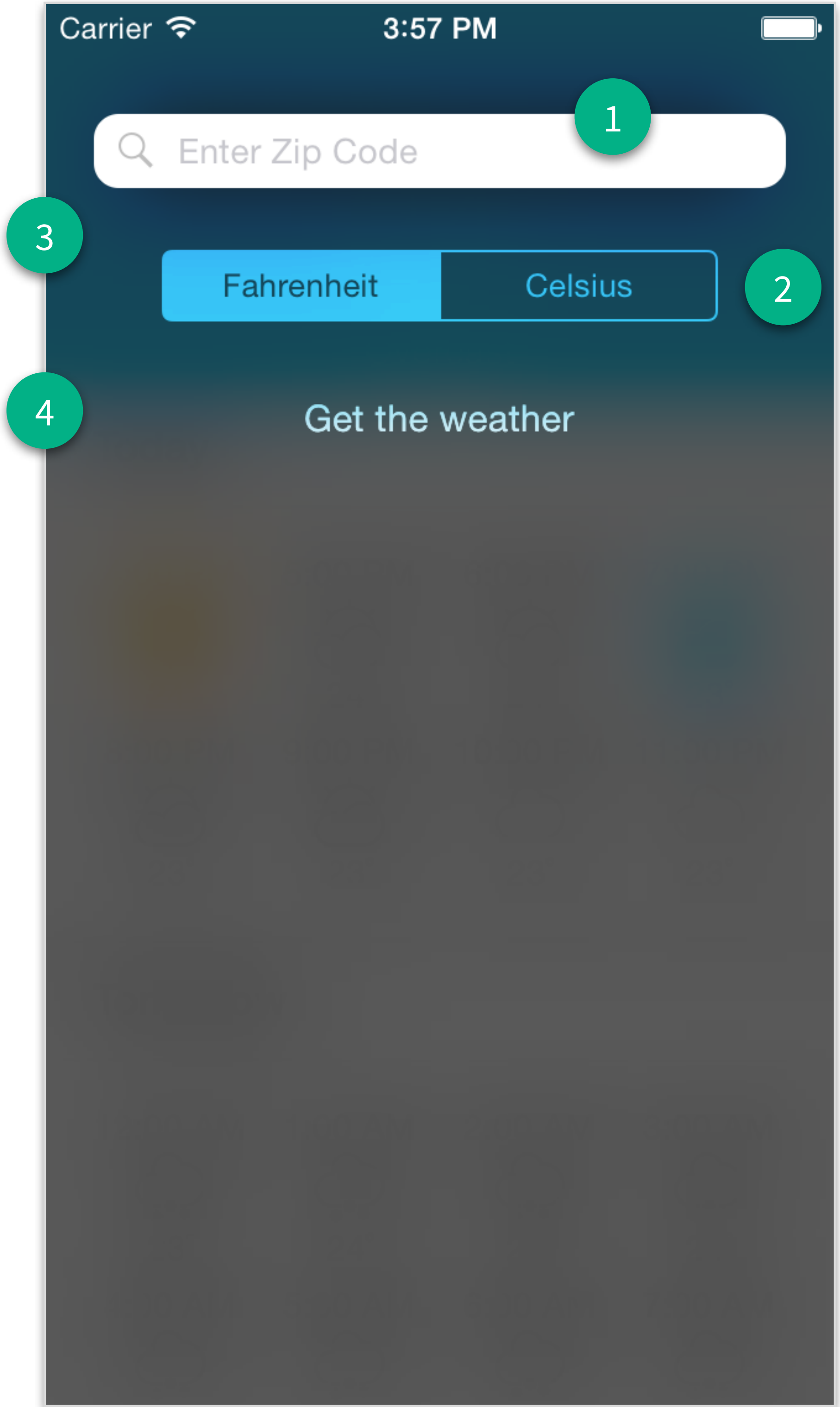
2. iOS: The settings cog should be 10 points from the right and top of the background.

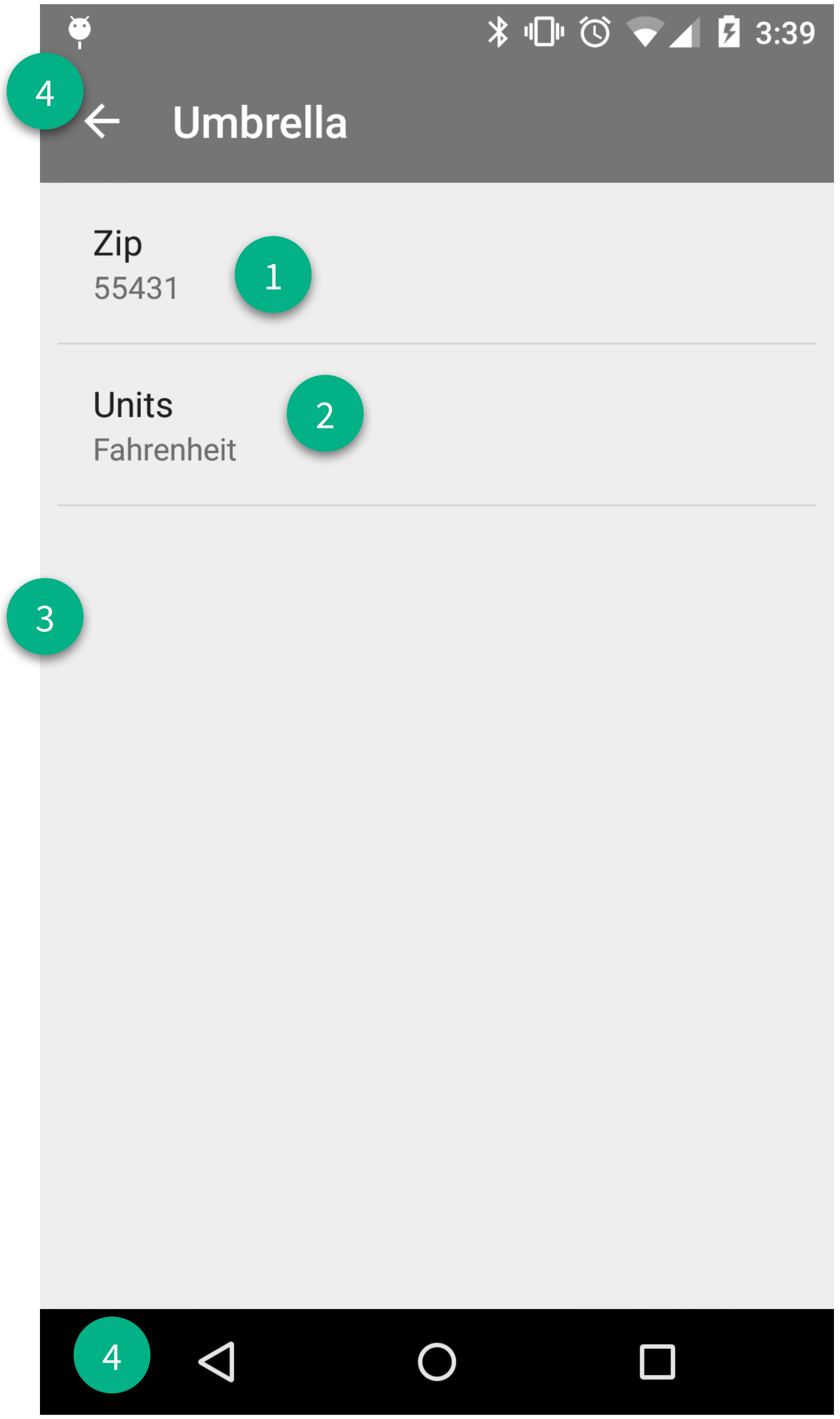
3. iOS: The current temperature should be center aligned both vertically and horizontally.

4. iOS: The city/state label should be a 10 points above the current temperature label and horizontally centered.

5. iOS: The current conditions label should be 10 points under the current temperature label and horizontally centered.

Mobile NAT - 2.0 Settings





Zip
55431

Units
Fahrenheit

Interactive & UI Notes

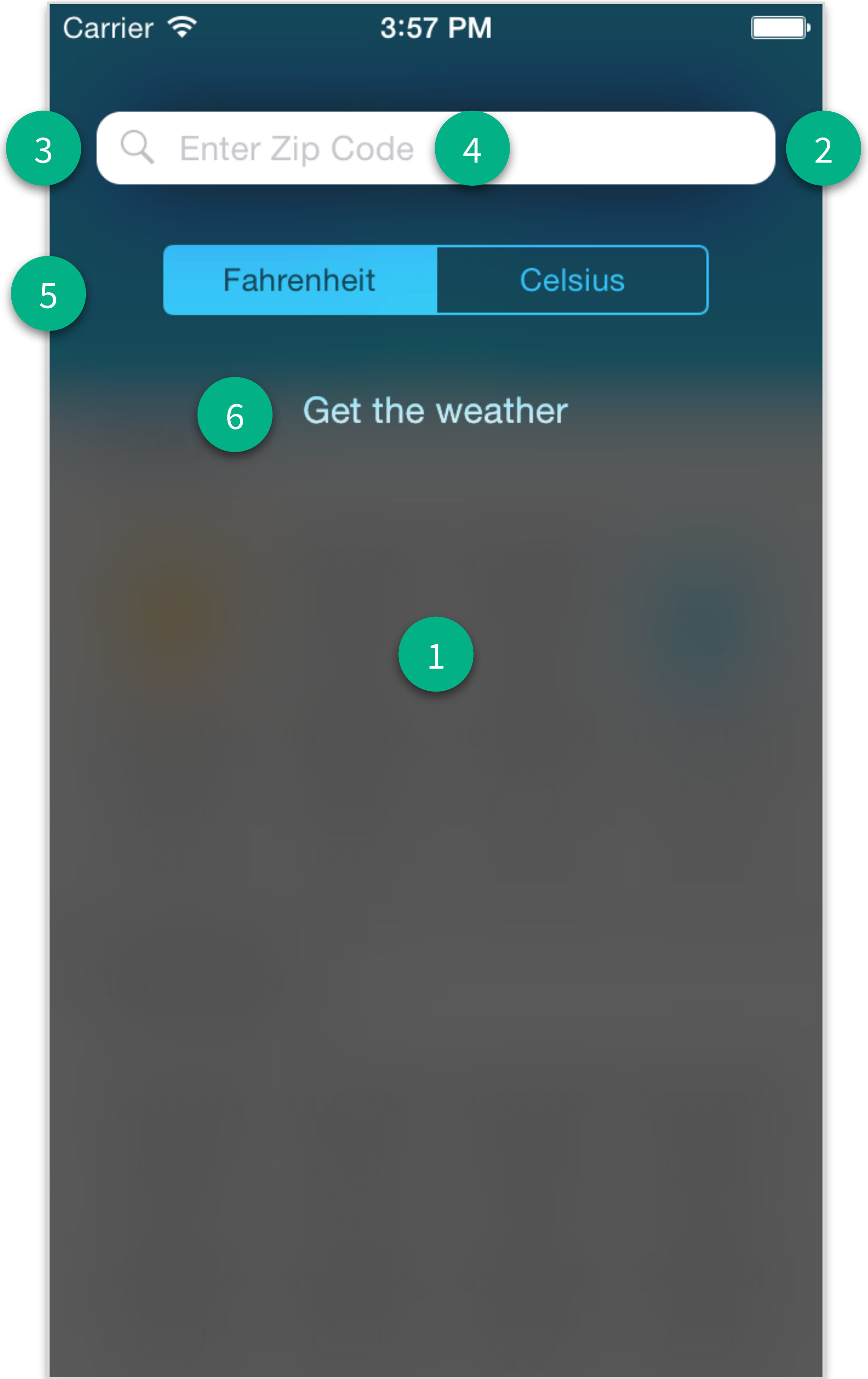
1. Users should be able to enter a 5-digit numeric ZIP code. We are assuming this application is only available to those in the United States.
On Android: The ZIP preference should open dialog that allows the user to input a ZIP code.

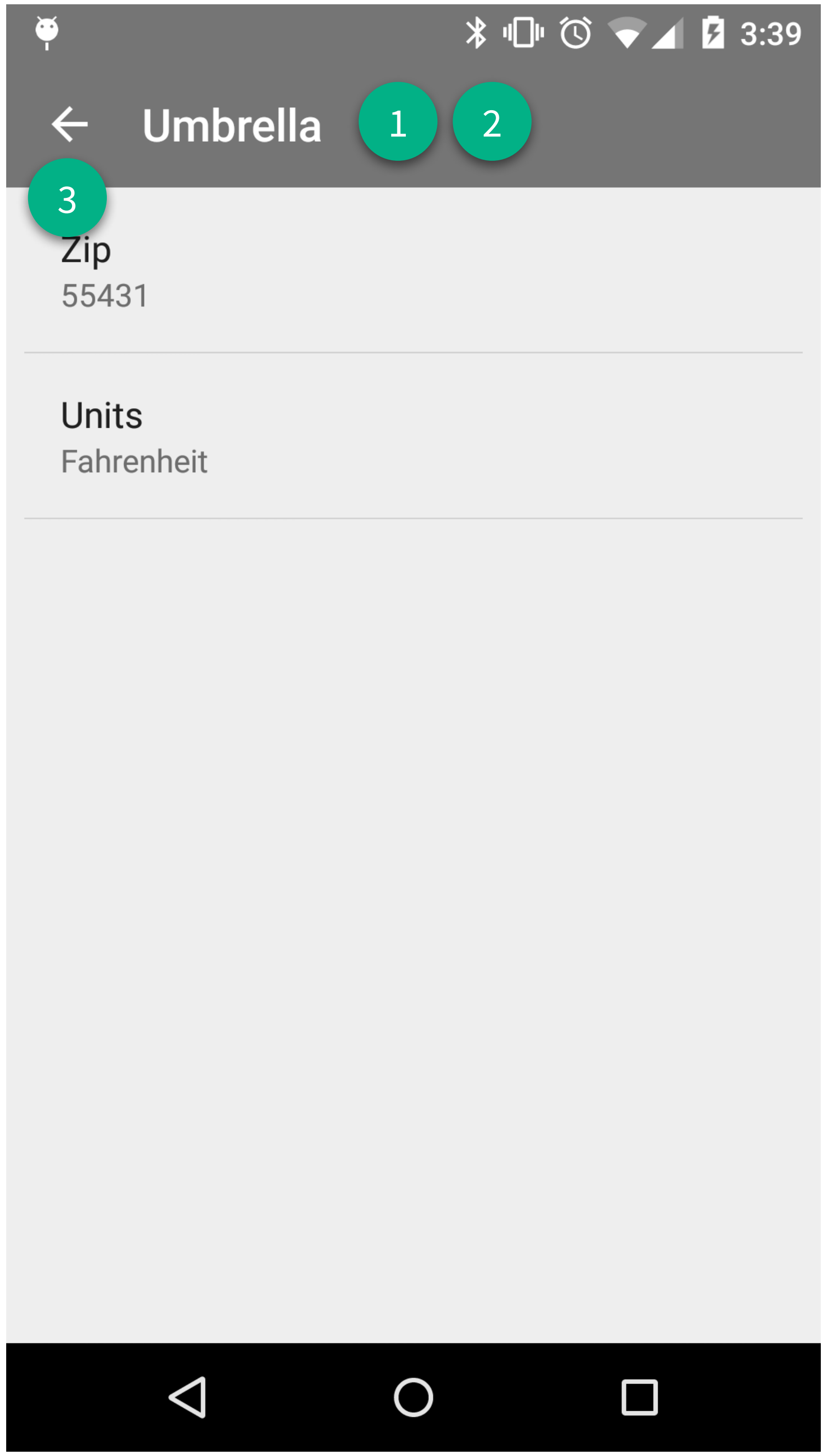
2. User should be able to select between Fahrenheit (Imperial) and Celsius (Metric) as the units of measure for temperature.
On Android: The Units preferences should open a dialog that allows the user to select between Fahrenheit and Celsius.

3. Any changes in this overlay screen are auto remembered by the application.

4. iOS: Clicking “Get the weather” takes the user to **1.0 Hourly Display**.
Android: Clicking either the up indicator or back button takes the user to **1.0 Hourly Display**.

Mobile NAT - 2.0 Settings [cont.]





Zip
55431

Units
Fahrenheit

Visual Design Notes

iOS:

1. The background of the settings screen should be the weather screen blurred with a dark effect. Tapping on the blurred area should dismiss this view.

2. The zip code entry should be 15 points from the left and right of the screen. It should be 45 points from the top of the screen. The background is white and has a corner radius of 10 points.

3. The search magnifying glass should be 10 points from the left of the background and vertically centered in the field. The icon is provided in the base project (settingsCog).

4. There should be 10 points of padding on the left and right side of the zip code text field.

5. The segmented control should be 25 points from the bottom of the zip field. It should be horizontally centered and 225 points wide. It should use a vibrancy effect.

6. The "Get the weather" button should be centered in the container and 25 points from the bottom of the segmented control. It should use a vibrancy effect.

Android:

1. Android should use the standard Preferences screen.
2. The ActionBar should be present on the settings screen, and should be #757575.
3. The ActionBar should display a standard up indicator.

Thank you!