# **CMPE 277 – Lab 3: My Weather iOS App**

# **Groups for Lab 3: 1**

# **By:**

# **Phat Truong, Tri Ninh, Tuan Ung and Vimmi Swami**

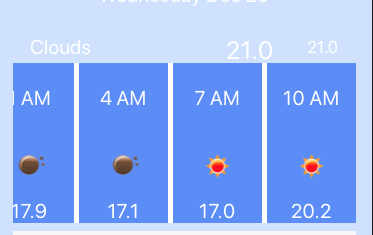
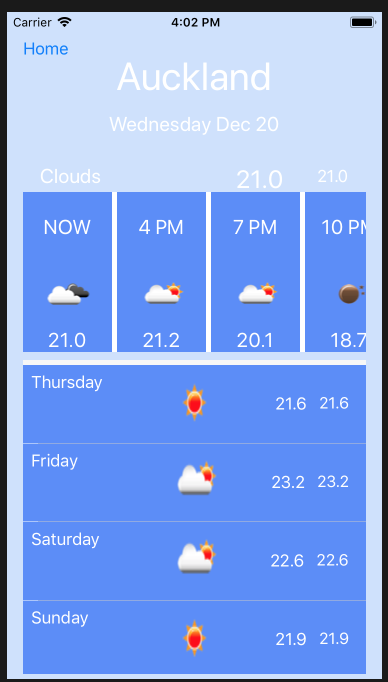
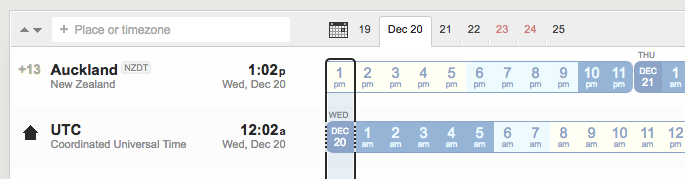
#### **City View**

1. Content:

The main City detail view shows name of the city, Day with date, current weather and temperatures: Max (bigger font) and Min (smaller font) for the day as Headings in white.

For 24 hour we have horizontal scrollbar, showing city’s Hour, weather image and Avg temperature.

For this Example, 24 hour starts with NOW(1pm) and ends at tomorrow’s 10AM ( as 10+3= 13, i.e., 1PM, completing 24 cycle.)



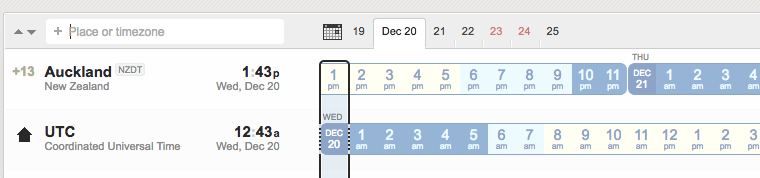


Above two images show Auckland’s Today’s (left) and 5days 3hr (right) weather data.

Based on first screenshot comparing UTC and Auckland’s time, at that time, Auckland’s time in UTC’s was 12:02 am. As UTC time was already past 12am, the weather 5day 3hr json will start from 3am UTC, which can be seen in its payload screenshot. Due to which, our app, updates first data of “NOW” same as Today’s weather data, and rest of the them start taking from 5days 3hr data. This can be confirmed by the values seen on Auckland’s detail view vs weather data of Today and 5days 3hr. We have confined precision to 1 decimal place, hence rounding takes place in the temperature final values.

For next 4days data, we find what is corresponding date in UTC for the city’s tomorrow’s noon. We then compare that with nearest 3 multiple match in HOUR number and fetch data from 5days 3hr weather data accordingly.

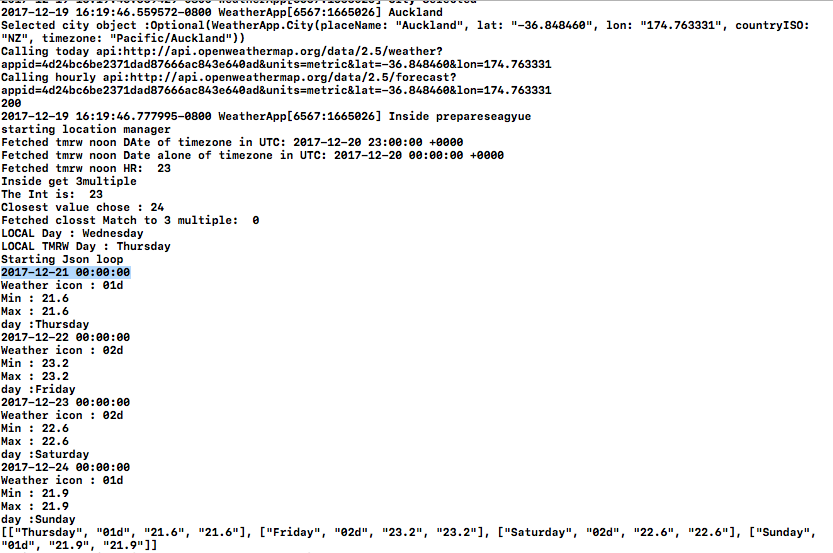
As it can be understood from below for Auckland, today noon (Dec 20) corresponds to UTC’s Dec 19’s 11 pm, so Auckland’s tomorrow Noon will correspond to UTC Dec 20 11pm. Now as 11pm (23) HOUR matches closely to 3’s multiple 24 HOUR, 24 HOUR is nothing but Dec 21 00:00 (early morning). Hence, we will fetch our data from 5days 3hr weather json starting from timestamp of Dec 21 00:00



Below is a screenshot of our print messages in app, which verifies above logic. The 4 days values in the end corresponds to dates:

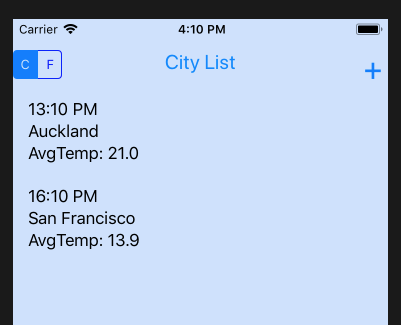
2017-12-21 00:00:00, 2017-12-22 00:00:00, 2017-12-23 00:00:00

& 2017-12-24 00:00:00

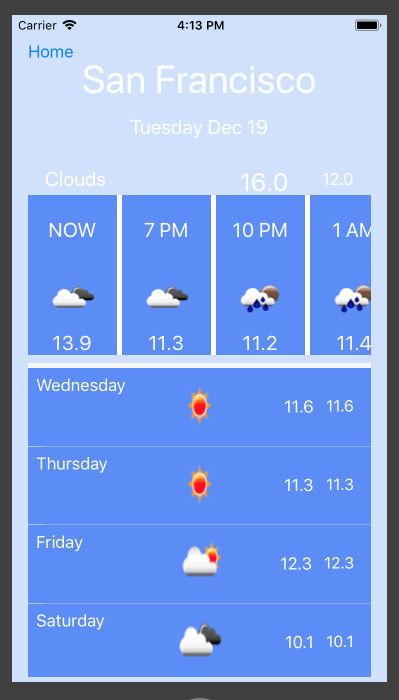


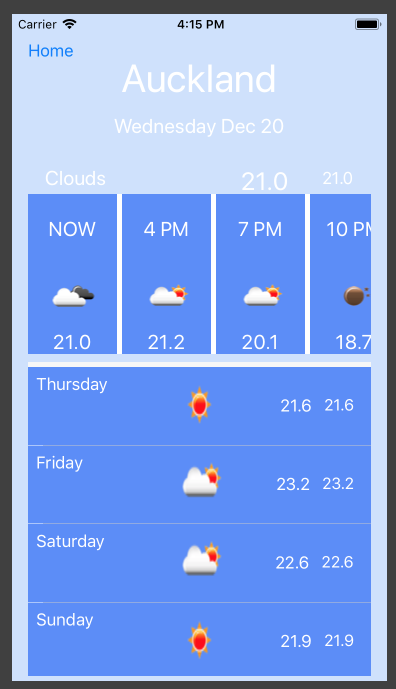
1. Navigation

The app provides swipe left and swipe right functionality on City detail view to view next available city w.r.t the direction of swipe. As can be seen from below initial City List view, Auckland is first and San Francisco is last.



We tap on Auckland-> to its Detail View. There we swipe from right–to-left, as expected, the view transitions from Auckland to SF. As we load weather data before loading the next view, the loaded city view may take some time to update the data in its view.



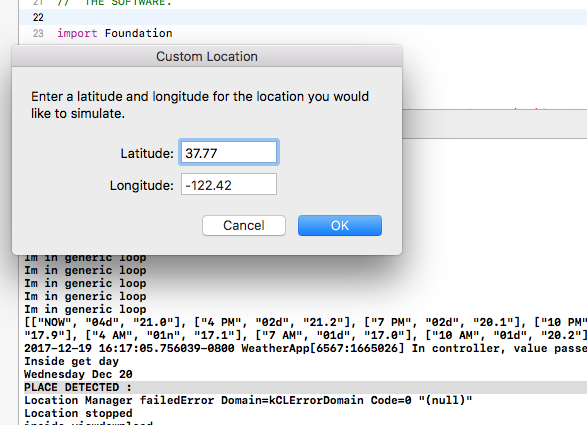


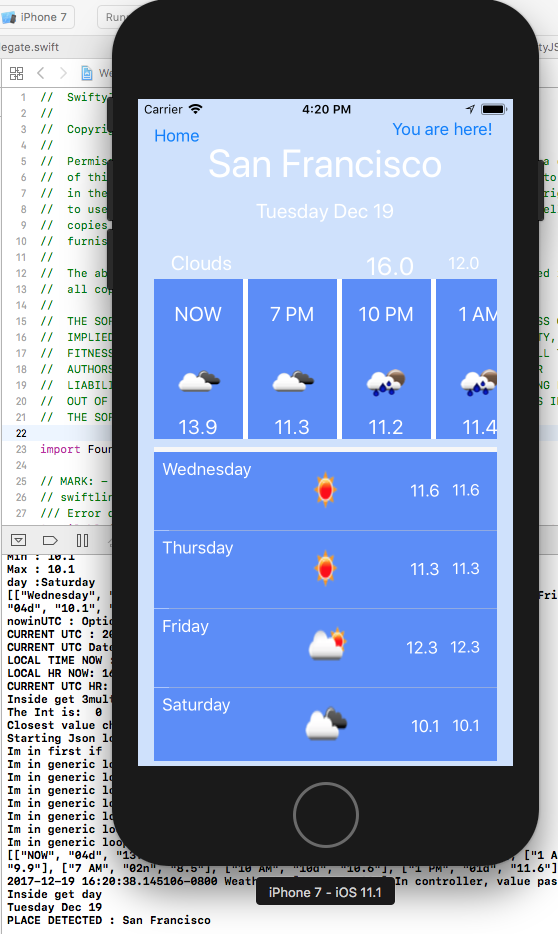
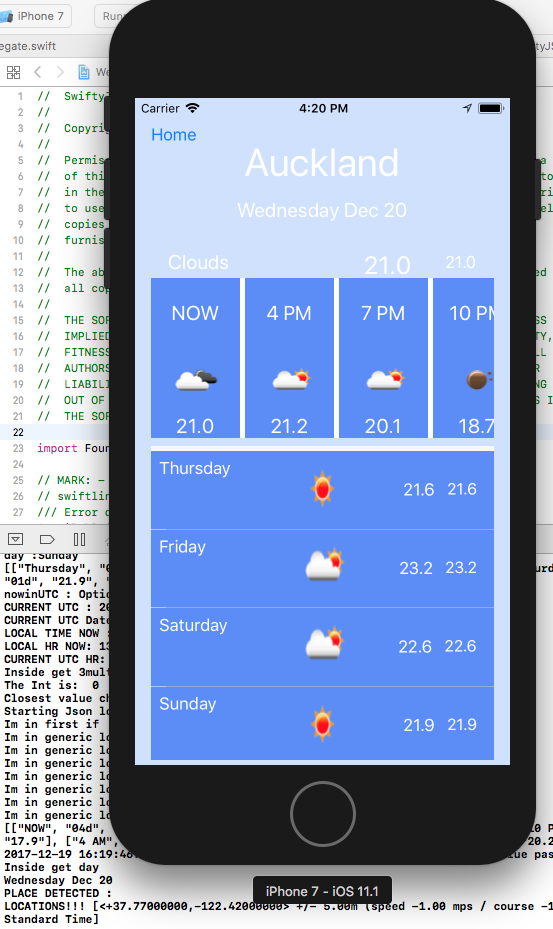
Above shows similar action as above, but now we swipe left-to-right from San Francisco and we get Auckland. The app won’t do anything if the view is last in its list for the swiped direction. For example, nothing will happen if we swipe again from right-to-left on San Francisco, as its last in list. Similarly, nothing will happen if we swipe left-to-right from Auckland as it’s the first in list.

1. Support for the current city

Below screenshot shows that we are changing device location to that of San Francisco.

The highlighted text in screenshot- ”PLACE DETECTED” shows null before clicking Ok.



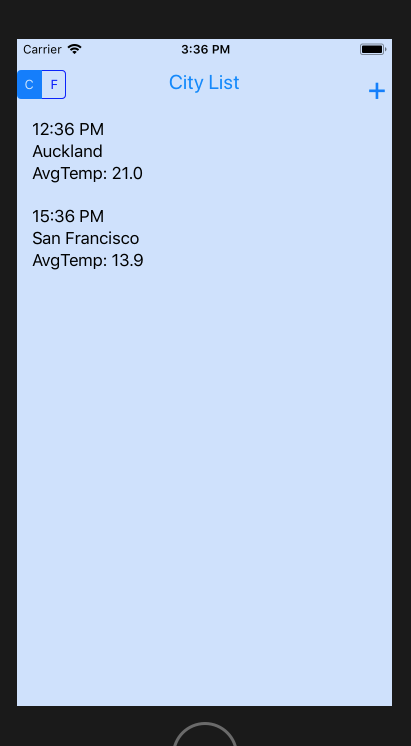


The location detection **starts** in City Detail view controller. As can be seen from left screenshot, Auckland detail view shows location arrow on top right corner of device. Just below the device, the debugging message under “PLACE DETECTED” text, fetches inputted location coordinates “LOCATION!! …..”. This ensures that the location update has worked.

Now to verify app functionality, we swipe to San Francisco detail view and we find “**You are here!”** text on top corner of the app (above the city name). This can be verified by the debug message below device screenshot showing message: “PLACE DETECTED: San Francisco”.

#### **City List View**

1. Shows a list of cities, where each city is rendered in a row, with info such as current local time of that city, city name, and current temperature.

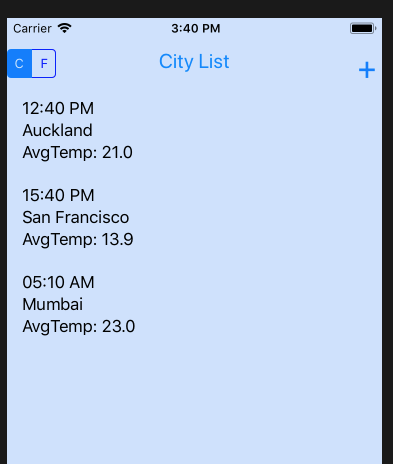
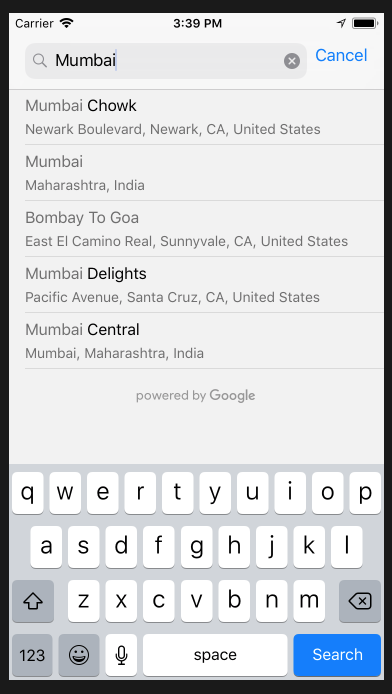
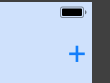


The City List view also shows C/F segmented button to set preference of C/F for temperatures.

The **+** button allows user to add to the city list. City list view reloads data for each city on view load, i.e., coming to City list view or adding to city List will reload data for each city. The UI is kept very simple for easy for user interactions.

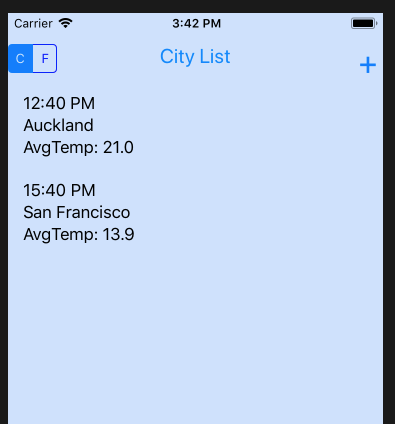
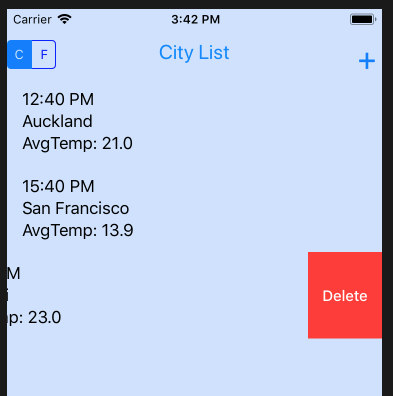
1. Provide ways for the user to add and delete cities, one at a time.
   1. Adding City

For Adding, click on **+** button on top right corner in City List view. Google place API integrated with app will then provide a view to search and select a city. In this case, we selected Mumbai city. On selection, the app then verifies if the Open Weather API and Timezone API successfully return response for the selected city. In case of any error, a popup alert will be shown to user with error message. If successful, the city will be appended to the list. This can be seen in right most screenshot, that Mumbai is successful added, and its corresponding data is displayed with other cities updated data too.



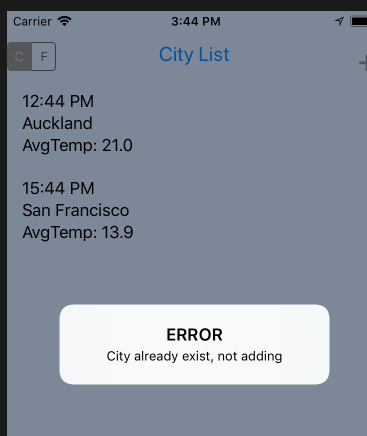
2. Deleting City

For deletion, just swipe right-to-left on the city to be deleted. As shown below, Mumbai added in above step is now being deleted and final City List doesn’t show Mumbai.



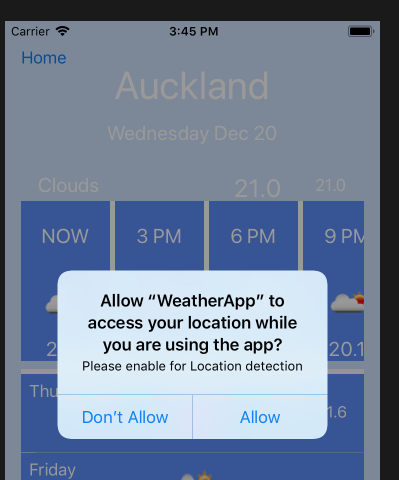
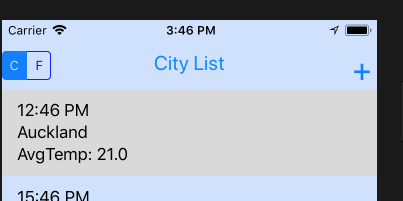
3. Not adding identical cities to the list:

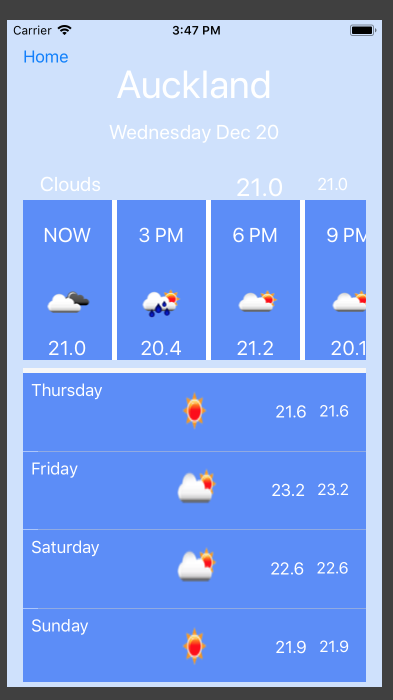
When user tries to add an existing city to the list, below pop-up message is shown to the user. It will hide on its own in 2 seconds.



4. Touch any specific city navigates the app to the City View.

From City List view, once we tap on a city, the background color changes, and the city detail view is shown. When the app is launched initially, it asks for permission to access location. If allowed, it starts monitoring. If not allowed, it doesn’t monitor location and requires manual permissions to be given through app settings.

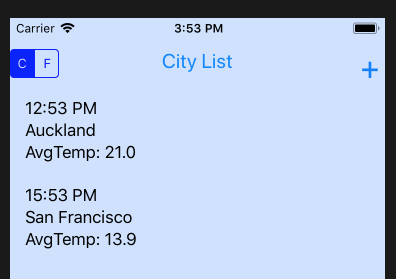
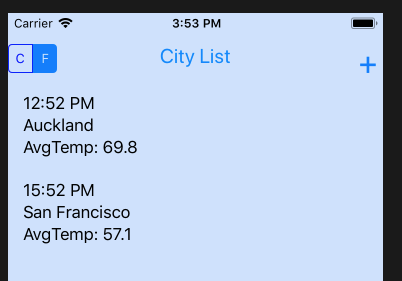
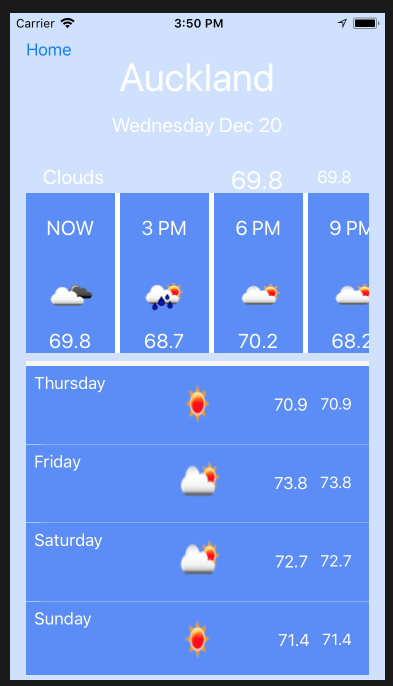
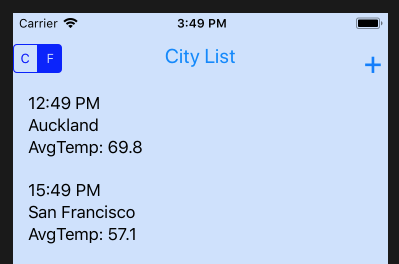
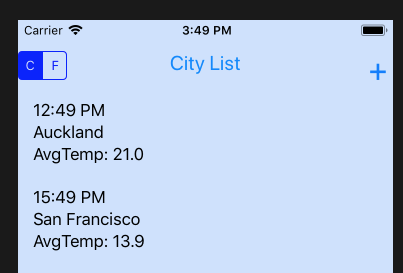




#### **Setting View**

1. The Setting View allows the user to specify the following preferences

Fahrenheit (F) vs Celsius (C).: The default is C. On change of preference, the selection is highlighted and the view is updated. In our app, we calculate temperature to decimal precession 1, if Fahrenheit is selected. As seen from below screenshots, the temperature values are updated on selecting Fahrenheit.



Above we can see that Fahrenheit is still selected in City List view after we clicked on HOME button from City Detail view. We then change to Celsius, and the City List view is refreshed with temperatures in Celsius.