

## CS 442: Mobile Applications Development

### Assignment 3 – Visual Crossing Weather App (300 pts)

Uses: Multiple Activities, Internet API, RecyclerViews, Option-Menus, Android Volley, JSON Data, Time conversion, Dialogs, Images

#### App Highlights:

- This app displays a variety of weather data for a specified location – current weather, hourly forecast (48 hours), and daily forecast (15 days).
- The app is made up of 2 activities – the home weather screen, and the daily forecast screen.
- The home weather screen *will* need an alternate layout for landscape orientation (see images). The daily forecast screen *will not* need an alternate landscape layout – the same layout should work in any orientation.
- The data that makes up the current weather, hourly forecast, and daily forecast are detailed in the images in this document.
- The units (C/F) can be toggled between imperial and metric by tapping an options-menu icon.
- The daily forecast is displayed by tapping the calendar icon in the options-menu.
- The location for the displayed weather can be changed by tapping the location icon in the options-menu.



#### Home screen Portrait & Landscape layouts

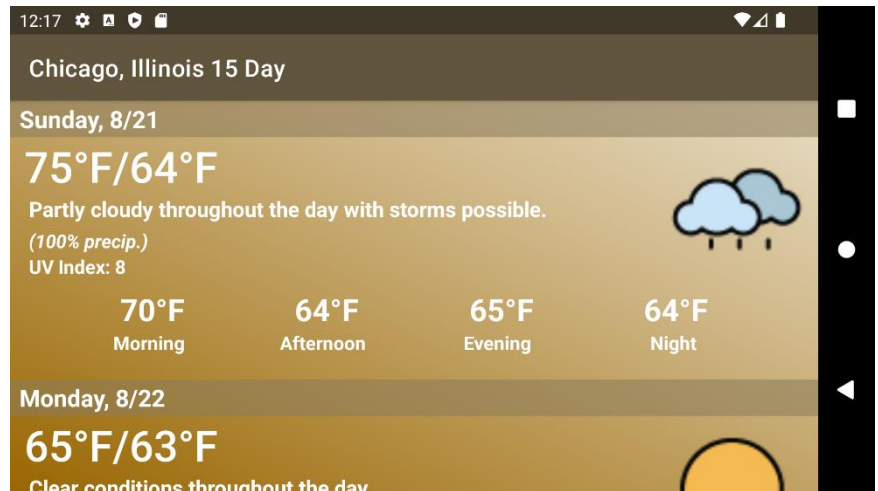
*Content details are shown later*





### 7-Day weather Portrait & Landscape layouts

*Content details are shown later*



#### A) Internet Data:

Weather data for this app will come from Visual Crossing Weather Data & API (<https://www.visualcrossing.com/>).

You will need to set up a free account and get an API key to use this API. You can create your account at <https://www.visualcrossing.com/sign-up>. Follow the instructions there to set up your free account. When asked your reason for setting up an account, you can say something like "Signing up for education".

Once you have your account, log in, click on "Account" (upper-right). You can find your API key on that account page, in the "Your Details" section, under the "Key" label. Your free account has a limit of 1000 calls/day.

We will use their "Timeline Weather" API endpoint (documentation at <https://www.visualcrossing.com/resources/documentation/weather-api/timeline-weather-api/>).

That call will look like:

`https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/<Location string>?unitGroup=us&lang=en&key=<your API key>`

- Location string: Location to use to obtain weather, Like "Dallas, TX", "Seattle, WA", etc (use "Chicago, IL" as the initial default latitude value)



- unitGroup: us or metric
- Your API key: your Visual Crossing API key

The JSON results of this endpoint call look like the below. Data we will make use of is highlighted in yellow. The unhighlighted data can be ignored.

```
{
  "queryCost": 1,
  "latitude": 41.8843,
  "longitude": -87.6324,
  "resolvedAddress": "Chicago, IL, United States",
  "address": "Chicago, IL",
  "timezone": "America/Chicago",
  "tzoffset": -5,
  "description": "Warming up with a chance of rain Thursday.",
  "days": [{
    "datetime": "2022-09-29",
    "datetimeEpoch": 1664427600,
    "tempmax": 63.6,
    "tempmin": 46.2,
    "temp": 54.3,
    "feelslikemax": 63.6,
    "feelslikemin": 44.3,
    "feelslike": 54,
    "dew": 43.1,
    "humidity": 66.6,
    "precip": 0,
    "precipprob": 0,
    "precipcover": 0,
    "preciptype": null,
    "snow": 0,
    "snowdepth": 0,
    "windgust": 13.2,
    "windspeed": 11,
    "winddir": 153.1,
    "pressure": 1031,
    "cloudcover": 40.9,
    "visibility": 12.3,
    "solarradiation": 110.8,
    "solarenergy": 9.6,
    "uvindex": 6,
    "severerisk": 10,
    "sunrise": "06:45:21",
    "sunriseEpoch": 1664451921,
    "sunset": "18:35:36",
    "sunsetEpoch": 1664494536,
    "moonphase": 0.07,
    "conditions": "Partially cloudy",
    "description": "Partly cloudy throughout the day.",
    "icon": "partly-cloudy-day",
    "stations": ["AP878", "KORD", "KMDW", "F1983", "KPWK"],
    "source": "comb",
    "hours": [{
      "datetime": "00:00:00",
      "datetimeEpoch": 1664427600,
      "temp": 50.5,
      "feelslike": 50.5,
      "humidity": 65.07,
      "dew": 39.2,
      "precip": 0,
```

```

        "precipprob": 0,
        "snow": 0,
        "snowdepth": 0,
        "preciptype": null,
        "windgust": 13.2,
        "windspeed": 3,
        "winddir": 45,
        "pressure": 1032.7,
        "visibility": 9.9,
        "cloudcover": 22.1,
        "solarradiation": 0,
        "solarenergy": null,
        "uvindex": 0,
        "severerisk": 10,
        "conditions": "Partially cloudy",
        "icon": "partly-cloudy-night",
        "stations": ["KORD", "KMDW", "KPWK"],
        "source": "obs"
    }, {
        ... 23 more "hours" records here
    }
  ], {
    ... 15 more "days" records here
  },
  "alerts": [],
  "stations": { ... },
  "currentConditions": {
    "datetime": "14:05:00",
    "datetimeEpoch": 1664478300,
    "temp": 58.4,
    "feelslike": 58.4,
    "humidity": 64.6,
    "dew": 46.5,
    "precip": 0,
    "precipprob": null,
    "snow": 0,
    "snowdepth": 0,
    "preciptype": null,
    "windgust": 11.2,
    "windspeed": 4.6,
    "winddir": 99,
    "pressure": 1031.9,
    "visibility": 9.9,
    "cloudcover": 25,
    "solarradiation": 52,
    "solarenergy": 0.2,
    "uvindex": 1,
    "conditions": "Partially cloudy",
    "icon": "partly-cloudy-day",
    "stations": ["UR352", "KMDW", "45198_maritime"],
    "sunrise": "06:45:21",
    "sunriseEpoch": 1664451921,
    "sunset": "18:35:36",
    "sunsetEpoch": 1664494536,
    "moonphase": 0.07
  }
}

```

## B) Application/Behavior Diagrams:

JSON field name is shown in parentheses after each entry below.

Full Timeline API documented at

<https://www.visualcrossing.com/resources/documentation/weather-api/timeline-weather-api/>

### 1) App Main screen

Location name (address)

Current Date/Time  
(currentConditions.datetimeEpoch)

Temperature  
(currentConditions.temp)

Feels Like (currentConditions.feelslike)

Humidity (currentConditions.humidity)

UV Index (currentConditions.uvindex)

Morning Temp  
(days[0].hours[8].temp)

Afternoon Temp  
(days[0].hours[13].temp)

48 Hourly Weather RecyclerView (see below)

Sunrise  
(currentConditions.sunriseEpoch)

Options Menu (3 items):

- Toggle units
- Show daily forecast
- Change location

Weather Icon  
(currentConditions.icon)

Weather description and cloud cover (currentConditions.conditions and currentConditions.cloudcover)

Winds (currentConditions.windir, currentConditions.windspeed, currentConditions.windgust)

Visibility  
(currentConditions.visibility)

Night Temp  
(days[0].hours[23].temp)

Evening Temp  
(days[0].hours[17].temp)

Sunset  
(currentConditions.sunsetEpoch)



Daily Weather RecyclerView daily entries. Current day should show "Today", subsequent days use day name. Start with hours after the current hour. Show hourly weather for today and 3 days forward (days[0].hours[n thru 23], days[1-3].hours[0-23])

Day ("Today", "Day name", "Day name", etc.). For example: "Today", "Friday", "Saturday".  
(days[n].hours[n].datetimeEpoch)

Time HH:MM am/pm  
(days[n].hours[n].datetimeEpoch)

Weather icon  
(days[n].hours[n].icon)

Temperature  
(days[n].hours[n].temp)

Description  
(days[n].hours[n].conditions)



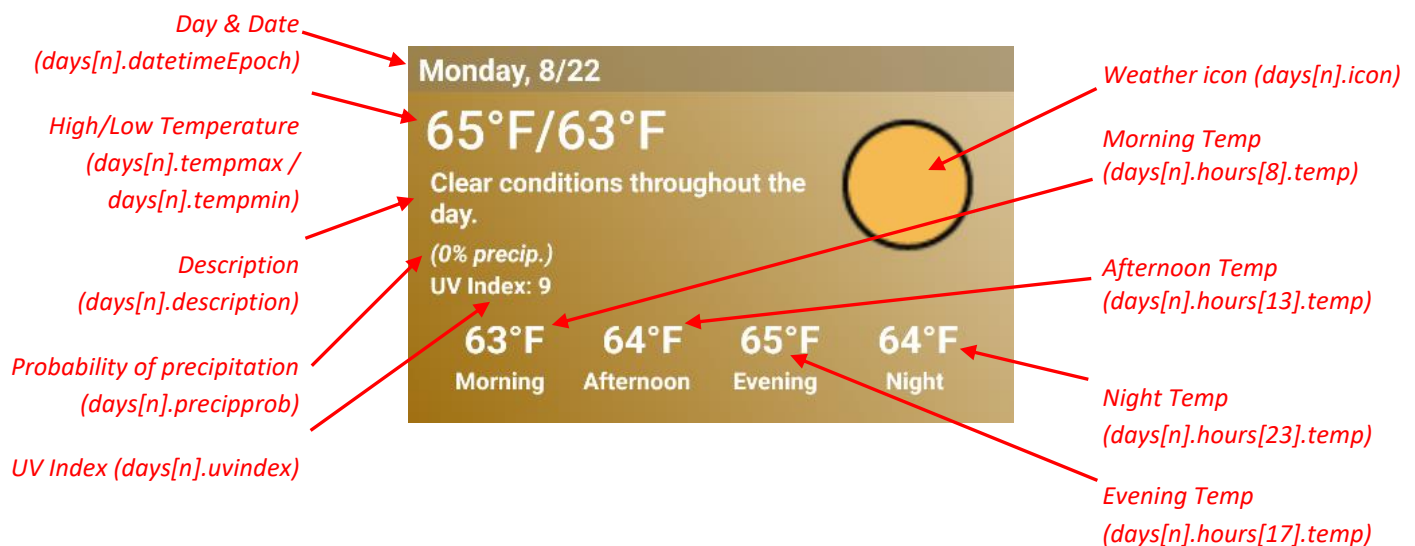


## 2) 7-Day weather Portrait & Landscape



Location

Day weather data RecyclerView. One entry per day (see below).





## 3) Options Menu

- Change Units  
Selecting this menu item should toggle the selected measurement unit from “us” to “metric” or from “metric” to “us”. When changed, the Visual Crossing Timeline API endpoint should be called again, specifying the newly selected unit. The change unit menu item’s icon should be changed to match the selected unit (°F for “us”, °C for “metric” – icon images provided)
- Daily Forecast  
Selecting this menu item should open the daily forecast activity (passing the daily forecast data content).
- Change Location  
Selecting this menu item should display an AlertDialog that allows the user to enter a new location, as shown below. Upon tapping OK, the specified location should be used to call the Visual Crossing Timeline API endpoint to get the new weather data.

Enter a Location  
For US locations, enter as 'City', or 'City, State'  
For international locations enter as 'City, Country'  
CANCEL OK

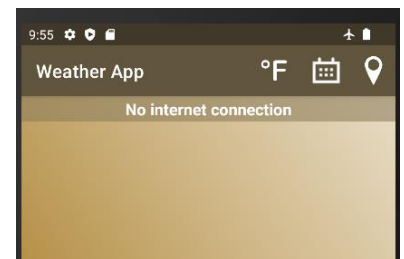
## 4) Weather icons

Weather icons found in the JSON weather data are provided as a string code (clear-day, partly-cloudy-night, rain, etc). These codes correspond to image file names (image files are provided to you). These icon names are used to access these provided images within your project (in the drawable resource folder). The following code shows how to do this:

```
// Get resource ID of desired icon
String icon = <the icon string as described above>
icon = icon.replace("-", "_"); // Replace all dashes with underscores
int iconID =
    anActivity.getResources().getIdentifier(icon, "drawable", anActivity.getPackageName());
if (iconID == 0) {
    Log.d(TAG, "parseCurrentRecord: CANNOT FIND ICON " + icon);
}
```

## 5) Handling no-network situations

When the device has no network connection, the app cannot access the Visual Crossing API site. In those situations (when the app attempts to access the internet without a connection), the home screen’s data/time text view should show “No internet connection”. When there is no network connection, the options-menu selections should *not* function. (Displaying a Toast message indicating that the function cannot be used when there is no network connection is fine in this situation).



```
private boolean hasNetworkConnection() {
    ConnectivityManager connectivityManager = getSystemService(ConnectivityManager.class);
    NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo();
    return (networkInfo != null && networkInfo.isConnectedOrConnecting());
}
```

Note – you must declare the ACCESS\_NETWORK\_STATE permission (in addition to INTERNET permission) in the project manifest!

## 6) Time Conversions

Date/Time ("Epoch" data fields) values are provided as the current time in seconds. These need to be converted to actual Date objects, then formatted as needed. This can be done as follows:

```
long datetimeEpoch = 1664427600;
Date dateTime = new Date(datetimeEpoch * 1000); // Java time values need milliseconds

SimpleDateFormat fullDate =
    new SimpleDateFormat("EEE MMM dd h:mm a, yyyy", Locale.getDefault());
SimpleDateFormat timeOnly = new SimpleDateFormat("h:mm a", Locale.getDefault());
SimpleDateFormat dayDate = new SimpleDateFormat("EEEE MM/dd", Locale.getDefault());

String fullDateStr = fullDate.format(dateTime); // Thu Sep 29 12:00 AM, 2022
String timeOnlyStr = timeOnly.format(dateTime); // 12:00 AM
String dayDateStr = dayDate.format(dateTime); // Thursday 09/29
```

## 7) Converting Wind Direction

The wind direction (winddir) is provided by the API in degrees (0-359 degrees). The app should display this in a more conventional fashion, using compass points (N, SE, W, etc). This can be derived by specifying what degrees correspond to these compass points. For purposes of this application, you can use the following:

```
private String getDirection(double degrees) {
    if (degrees >= 337.5 || degrees < 22.5)
        return "N";
    if (degrees >= 22.5 && degrees < 67.5)
        return "NE";
    if (degrees >= 67.5 && degrees < 112.5)
        return "E";
    if (degrees >= 112.5 && degrees < 157.5)
        return "SE";
    if (degrees >= 157.5 && degrees < 202.5)
        return "S";
    if (degrees >= 202.5 && degrees < 247.5)
        return "SW";
    if (degrees >= 247.5 && degrees < 292.5)
        return "W";
    if (degrees >= 292.5 && degrees < 337.5)
        return "NW";
    return "X"; // We'll use 'X' as the default if we get a bad value
}
```



## 8) Additional provided images



daily.png



location.png



units\_c.png



units\_f.png



background.png



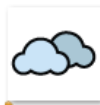
alert.png



clear\_day.png



clear\_night.png



cloudy.png



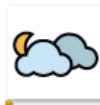
fog.png



hail.png



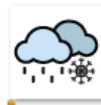
partly\_cloudy\_day.png



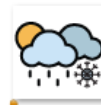
partly\_cloudy\_night.png



rain.png



rain\_snow.png



rain\_snow\_showers\_day.png



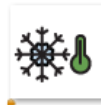
rain\_snow\_showers\_night.png



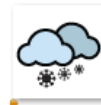
showers\_day.png



showers\_night.png



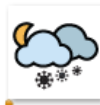
sleet.png



snow.png



snow\_showers\_day.png



snow\_showers\_night.png



thunder.png



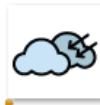
thunder\_rain.png



thunder\_showers\_day.png



thunder\_showers\_night.png



wind.png

## 9) Extra Credit Options

There are 2 extra credit options in this project – Incorporating “Swipe Refresh”, and “Saving User Settings”.

- (+15 pts) Swipe Refresh – Add swipe-refresh capability (to the home screen only) to reload the weather data for the current location. If there is no network connection, do nothing – leave the existing weather data in place and indicate the no-network situation in a Toast message.
- (+15 pts) Saving User Settings – Save the currently selected location latitude and longitude and the current units selection so that when the app is restarted, the weather displayed is for the saved location, and is displayed in the saved measurement units. Note, the “units” options-menu and all displayed temperature values should correctly display the currently selected measurement units.

### Assignment Assistance

The TAs for our course are available to assist you with your assignment if needed. Questions on assignment requirements and course concepts can be sent to the instructor.

### Submissions & Grading

- 1) Submissions must consist of your zipped project folder (*please execute Build =>Clean Project before generating the zip file*).
- 2) Submissions should reflect the concepts and practices we cover in class, and the requirements specified in this document.
- 3) Late submissions will be penalized by 10% per week late. (i.e., from one second late to 1 week late: 10% penalty, from one week late to 2 weeks late: 20% penalty, etc.). **NO SUBMISSIONS CAN BE MADE BEYOND 2 WEEKS LATE.**
- 4) Grading will be based upon the presence and proper functionality of *all features and behaviors* described in this document. NOTE: All descriptions and images constitute the requirements for this assignment.

### NOTE

**This assignment is worth 300 points. This means (for example) that if you get 89% on this assignment, your recorded score will be:**

**(89% \* 300 points = 267 points)**

***Note that this also means that the 10% late submission penalty will be 10% \* 300 points = 30 points.***

*If you do not understand anything in this handout, please ask.*

*Otherwise, the assumption is that you understand the content.*

***Unsure? Ask!***