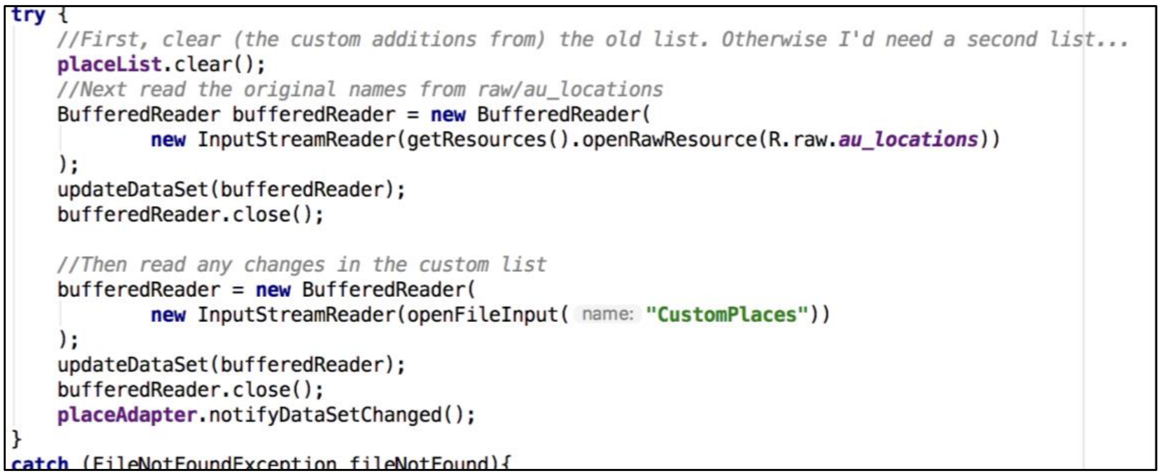
Task 9.1P

Working with Persistent Data, again

1. Add a Custom Geolocation

Code Snippet: File Load

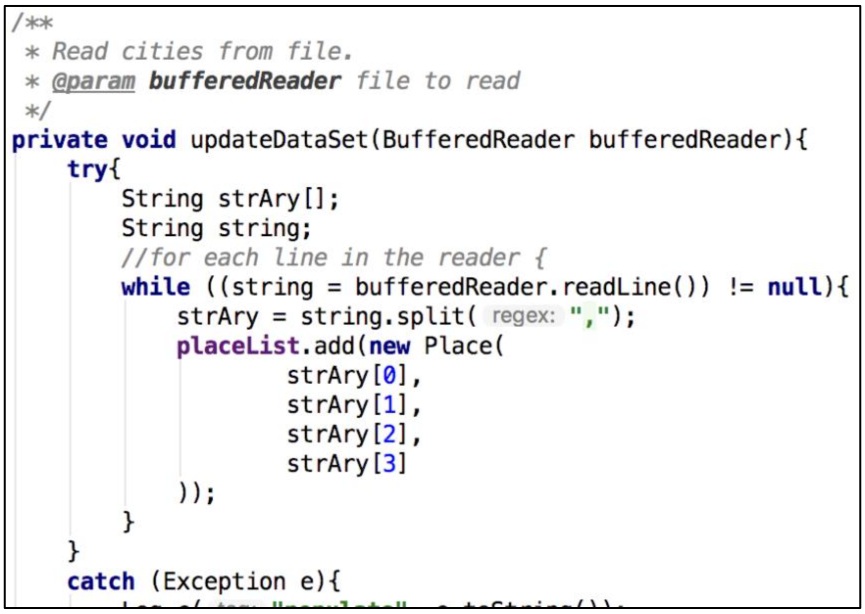
Since you can't write to res/raw, I figured it would be best to load the saved file separately. This lead to me combining the two loaders into a method and just passing it different buffered readers:



```java
try {
    //First, clear (the custom additions from) the old list. Otherwise I'd need a second list...
    placeList.clear();
    //Next read the original names from raw/au_locations
    BufferedReader bufferedReader = new BufferedReader(
            new InputStreamReader(getResources().openRawResource(R.raw.au_locations))
    );
    updateDataSet(bufferedReader);
    bufferedReader.close();

    //Then read any changes in the custom list
    bufferedReader = new BufferedReader(
            new InputStreamReader(openFileInput( name: "CustomPlaces"))
    );
    updateDataSet(bufferedReader);
    bufferedReader.close();
    placeAdapter.notifyDataSetChanged();
}
catch (FileNotFoundException fileNotFound){
```

The method then proceeds as usual:

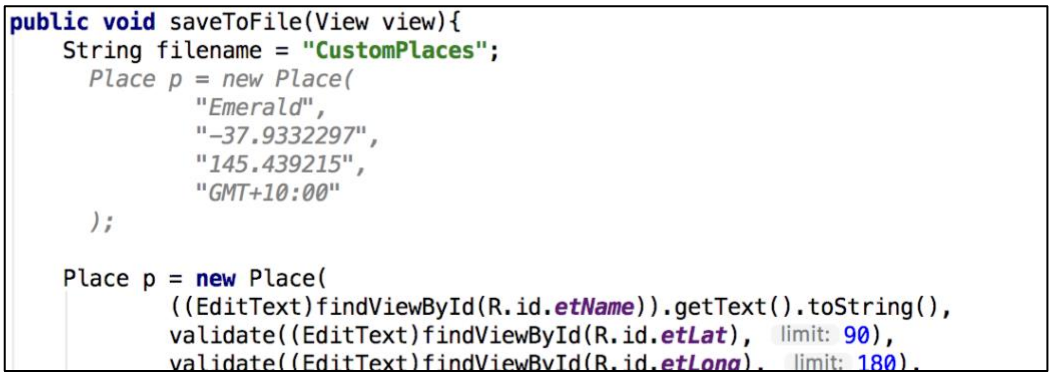


```java
/**
 * Read cities from file.
 * @param bufferedReader file to read
 */
private void updateDataSet(BufferedReader bufferedReader){
    try{
        String strAry[];
        String string;
        //for each line in the reader {
        while ((string = bufferedReader.readLine()) != null){
            strAry = string.split( regex: ",");
            placeList.add(new Place(
                    strAry[0],
                    strAry[1],
                    strAry[2],
                    strAry[3]
            ));
        }
    }
    catch (Exception e){
        [illegible]
```

'place Adapter' is then told the dataset is updated afterwards.

Code Snippet: File Write

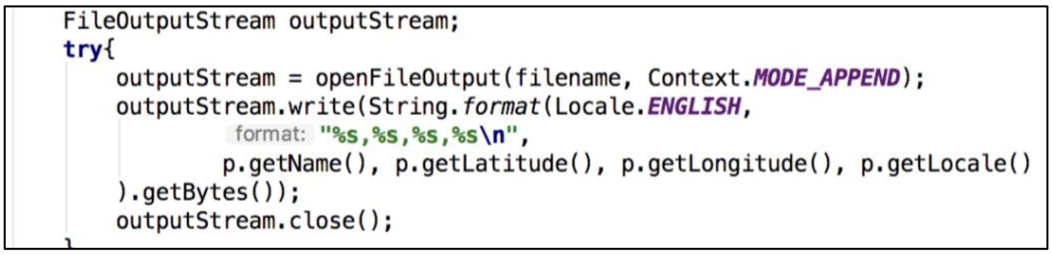
'Open file output' creates a file if one matching the filename isn't found, which saves some error handling. A 'Place' object (Name, Lat, Long, time zone) is then created based on the values in the fields, minor validation is applied and the location is then saved:



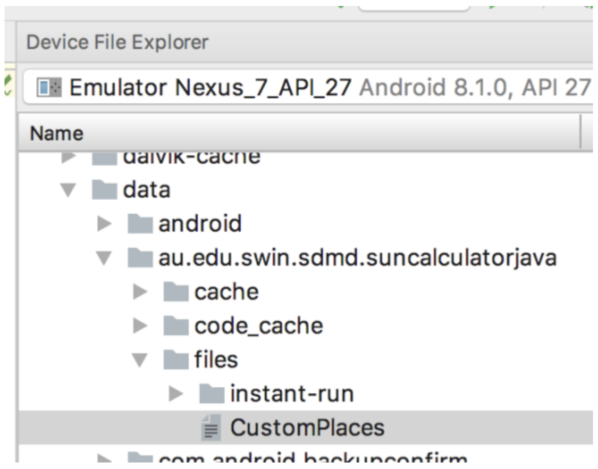
```java
public void saveToFile(View view){
    String filename = "CustomPlaces";
      Place p = new Place(
              "Emerald",
              "-37.9332297",
              "145.439215",
              "GMT+10:00"
      );

    Place p = new Place(
            ((EditText)findViewById(R.id.etName)).getText().toString(),
            validate((EditText)findViewById(R.id.etLat),  limit: 90),
            validate((EditText)findViewById(R.id.etLong),  limit: 180),
```

…



```java
    FileOutputStream outputStream;
    try{
        outputStream = openFileOutput(filename, Context.MODE_APPEND);
        outputStream.write(String.format(Locale.ENGLISH,
                 format: "%s,%s,%s,%s\n",
                p.getName(), p.getLatitude(), p.getLongitude(), p.getLocale()
        ).getBytes());
        outputStream.close();
    }
```

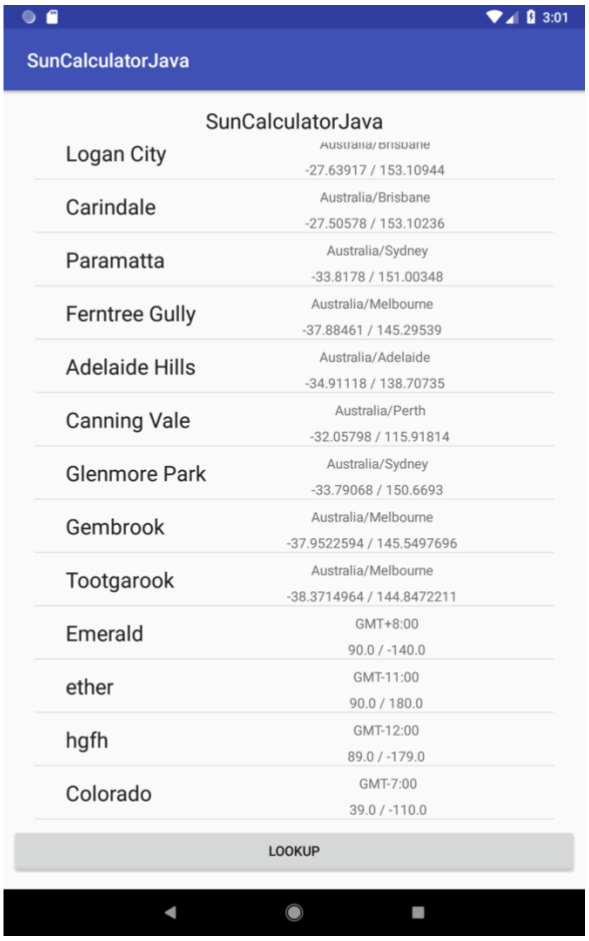


```
Device File Explorer
Emulator Nexus_7_API_27 Android 8.1.0, API 27
Name
  ▶ dalvik-cache
  ▼ data
      ▶ android
      ▼ au.edu.swin.sdmd.suncalculatorjava
          ▶ cache
          ▶ code_cache
          ▼ files
              ▶ instant-run
                CustomPlaces
      ▶ com.android.backupconfirm
```

Curiously, I found an issue with the sun time calculator while testing the validation – if one inputs random coordinates that don't match their time zone, the calculator couldn't handle it and closes before the user gets to see any visible effect from their tap.
On the other hand, the validation at least prevents the program from stopping if the Latitude is more than +/- 90, or Longitude is more than +/-180.

Appendix: Screenshots:

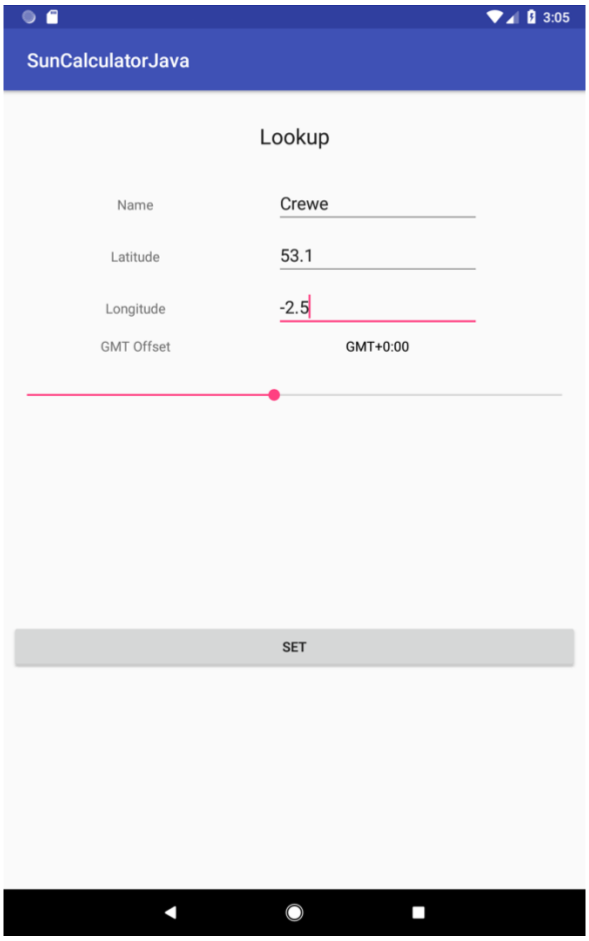
Let’s add the English city of Crewe to the calculator:



SunCalculatorJava

SunCalculatorJava

Logan City	Australia/Brisbane -27.63917 / 153.10944
Carindale	Australia/Brisbane -27.50578 / 153.10236
Paramatta	Australia/Sydney -33.8178 / 151.00348
Ferntree Gully	Australia/Melbourne -37.88461 / 145.29539
Adelaide Hills	Australia/Adelaide -34.91118 / 138.70735
Canning Vale	Australia/Perth -32.05798 / 115.91814
Glenmore Park	Australia/Sydney -33.79068 / 150.6693
Gembrook	Australia/Melbourne -37.9522594 / 145.5497696
Tootgarook	Australia/Melbourne -38.3714964 / 144.8472211
Emerald	GMT+8:00 90.0 / -140.0
ether	GMT-11:00 90.0 / 180.0
hgfh	GMT-12:00 89.0 / -179.0
Colorado	GMT-7:00 39.0 / -110.0

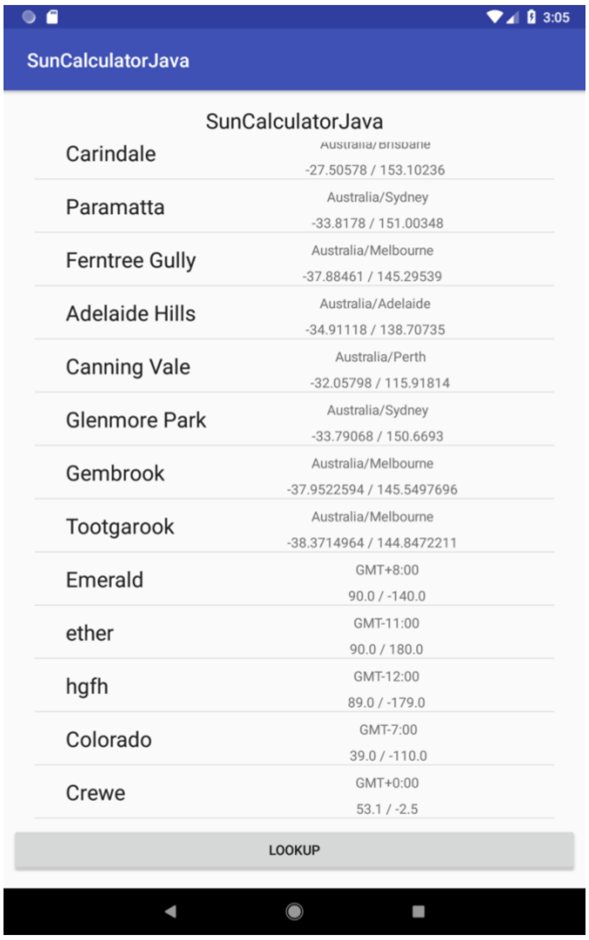


SunCalculatorJava

Lookup

Name	Crewe
Latitude	53.1
Longitude	-2.5
GMT Offset	GMT+0:00

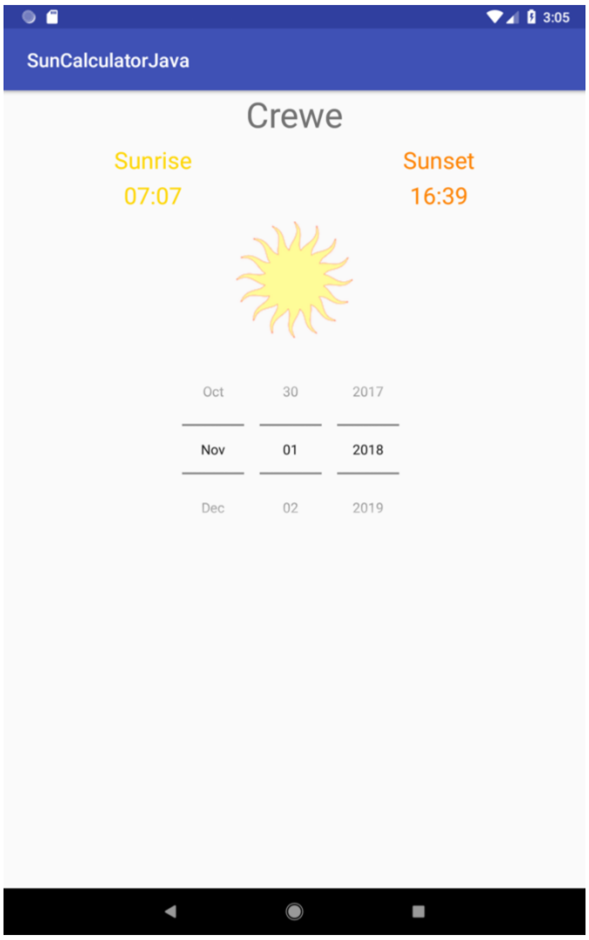
Left: Crewe isn’t on our list. (Some test cities are, they can be ignored for now)
Right: Google Maps tells us it’s at about 53.1 Latitude and -2.5 Longitude. Still within GMT+0.



SunCalculatorJava

SunCalculatorJava

Carindale	Australia/Brisbane	-27.50578 / 153.10236
Paramatta	Australia/Sydney	-33.8178 / 151.00348
Ferntree Gully	Australia/Melbourne	-37.88461 / 145.29539
Adelaide Hills	Australia/Adelaide	-34.91118 / 138.70735
Canning Vale	Australia/Perth	-32.05798 / 115.91814
Glenmore Park	Australia/Sydney	-33.79068 / 150.6693
Gembrook	Australia/Melbourne	-37.9522594 / 145.5497696
Tootgarook	Australia/Melbourne	-38.3714964 / 144.8472211
Emerald	GMT+8:00	90.0 / -140.0
ether	GMT-11:00	90.0 / 180.0
hgfh	GMT-12:00	89.0 / -179.0
Colorado	GMT-7:00	39.0 / -110.0
Crewe	GMT+0:00	53.1 / -2.5



SunCalculatorJava

Crewe

Sunrise 07:07

Sunset 16:39

Left: Pressing 'Set' takes us back to the main screen, and there's Crewe!
Right: And the local sunrise/set times.