

Mobile Dev 5.1P

Daniel Coady (102084174)

13/10/2019

Screenshots

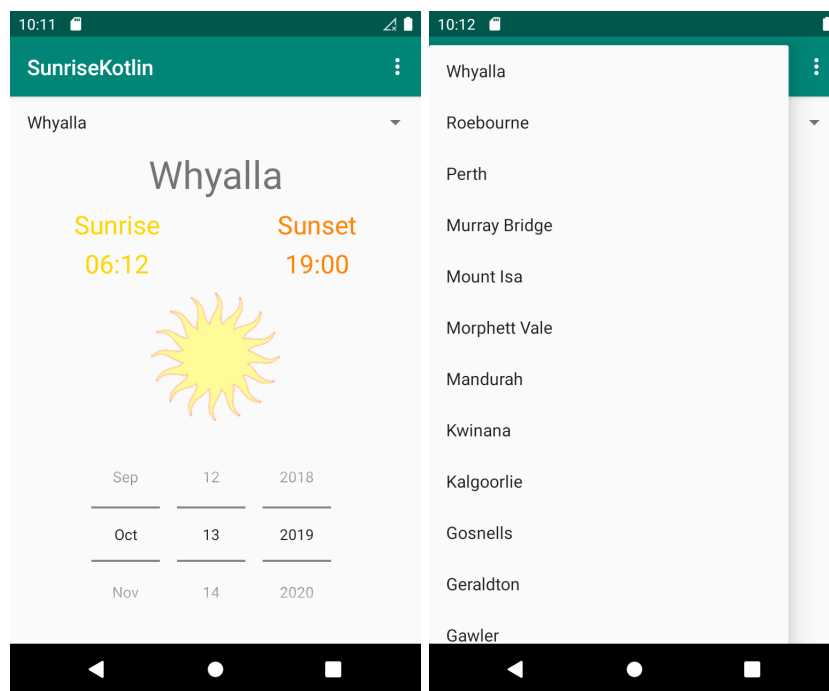


Figure 1: The main screen of the application

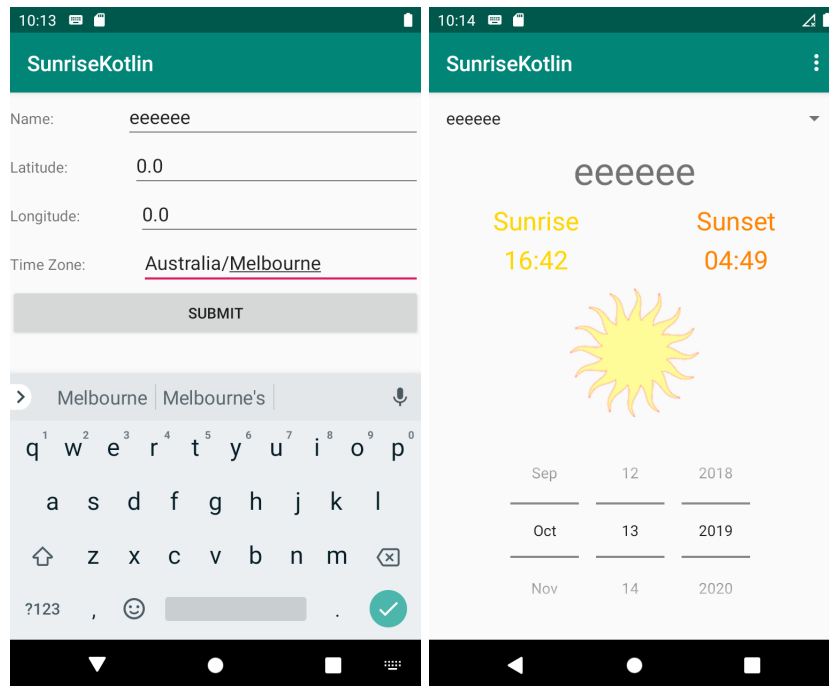


Figure 2: Adding a location to the application

Code

```
private fun populateDropDown(year: Int, month: Int, day: Int) {
    // reading in the file
    var options = ArrayList<String>()
    val file = openFileInput( name: "locations.txt")
    val reader = InputStreamReader(file)
    val buffReader = BufferedReader(reader)

    // populate the drop down menu
    var line = buffReader.readLine()
    while (line != null) {
        options.add(line.split( ...delimiters: ",") [0])

        line = buffReader.readLine()
    }
    location_selection.adapter = ArrayAdapter( context: this, android.R.layout.simple_list_item_1, options)

    location_selection.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
        override fun onNothingSelected(parent: AdapterView<*>?) {
        }

        override fun onItemSelected(parent: AdapterView<*>?, view: View?, position: Int, id: Long) {
            // is this a really bad way to do things? yes. i should have created an object for
            // all of the locations that i could then iterate through. you know what though? i'm
            // not being marked for efficiency and good code so it's stayin like this, if only
            // because it's easier as i bodge this all together
            val name = location_selection.selectedItem as String
            val lat: Double
            val lon: Double
            val timezone: String

            // read file for our location
            val file = openFileInput( name: "locations.txt")
            val reader = InputStreamReader(file)
            val buffReader = BufferedReader(reader)
            var line = buffReader.readLine()
            while (line != null) {
                if (line.split( ...delimiters: ",") [0] == name)
                    break // will break if we have a match
                line = buffReader.readLine()
            }

            lat = line.split( ...delimiters: ",") [1].toDouble()
            lon = line.split( ...delimiters: ",") [2].toDouble()
            timezone = line.split( ...delimiters: ",") [3]

            updateLocation(year, month, day, name, lat, lon, timezone)
        }
    }
}
```

Figure 3: Loading options and handling selections

```
fun save(_v: View) {  
    val name = txtName.text.toString()  
    val lat = txtLatitude.text.toString()  
    val lon = txtLongitude.text.toString()  
    val tz = txtTimezone.text.toString()  
    val data = name + " " + lat + " " + lon + " " + tz + "\n"  
  
    val file = File(filesDir, child: "locations.txt")  
    val fos = FileOutputStream(file, append: true)  
    fos.write(data.toByteArray())  
    finish()  
}
```

Figure 4: Saving a new location