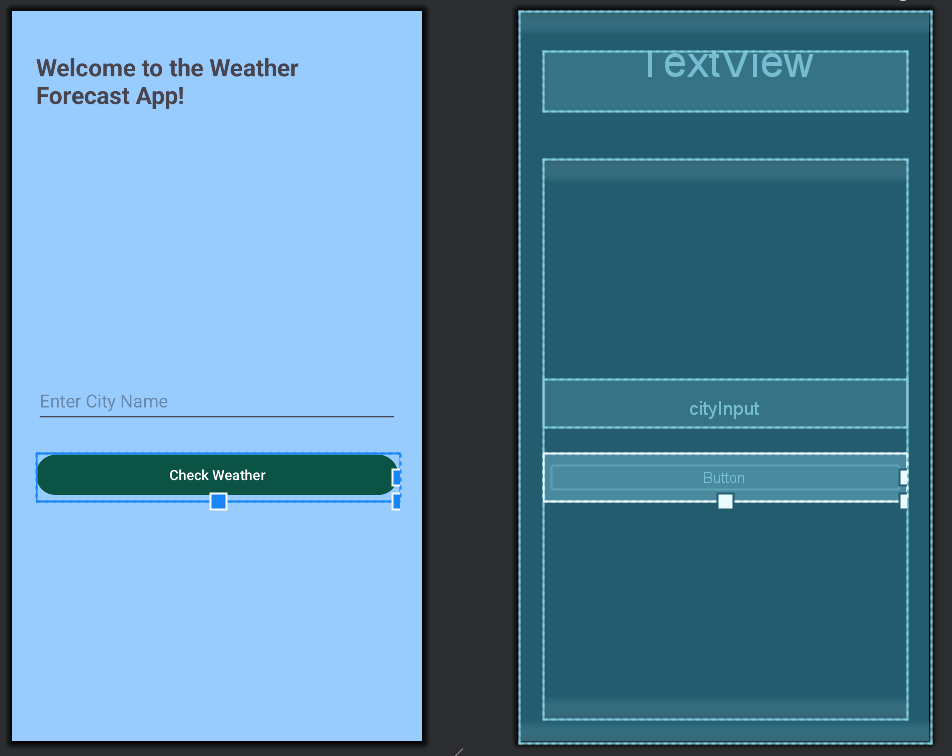
CS377 Final Project - Part 1

By: Samantha Jurado & Andrew Angus

Part 1: Initial Setup and Progress

For Part 1 of your final project, submit the following:

* **Group Members:** Samantha Jurado & Andrew Angus; Samantha will do initial idea coding work and background information, Andrew will help with connecting and UI implementation.We will switch jobs as needed.
* **Physical Phone Availability:** We have access to Phone # 1 from what we checked out from the class.
* **Chosen REST API:** We have chosen Open-Meteo as our rest API, which we will use to draw on the current weather forecast information and display it for our weather app. The goal is to have the user input which city they’re looking up weather forecast information for, and then have it appear on the next page for display.
* **Basic UI:** Create a simple user interface in Android Studio. Create screenshots, Kotlin, and xml files for the user interface to demonstrate the app’s initial design and layout. (Please See Below!)

The first screen would be our welcome page, where eventually we’ll add more images, but for now this works:  


activity\_main.xml:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="24dp"

android:background="#98cbfe"

>

<!-- Title at the top -->

<!-- Centered content -->

<TextView

android:id="@+id/welcomeText"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="16dp"

android:layout\_marginBottom="48dp"

android:text="Welcome to the Weather Forecast App!"

android:textSize="24sp"

android:textStyle="bold"

android:textColor="#0b5345"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1"

android:gravity="center"

android:orientation="vertical">

<EditText

android:id="@+id/cityInput"

android:layout\_width="match\_parent"

android:layout\_height="48dp"

android:hint="Enter City Name"

android:inputType="text"

android:layout\_marginBottom="26dp" />

<Button

android:id="@+id/submitButton"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

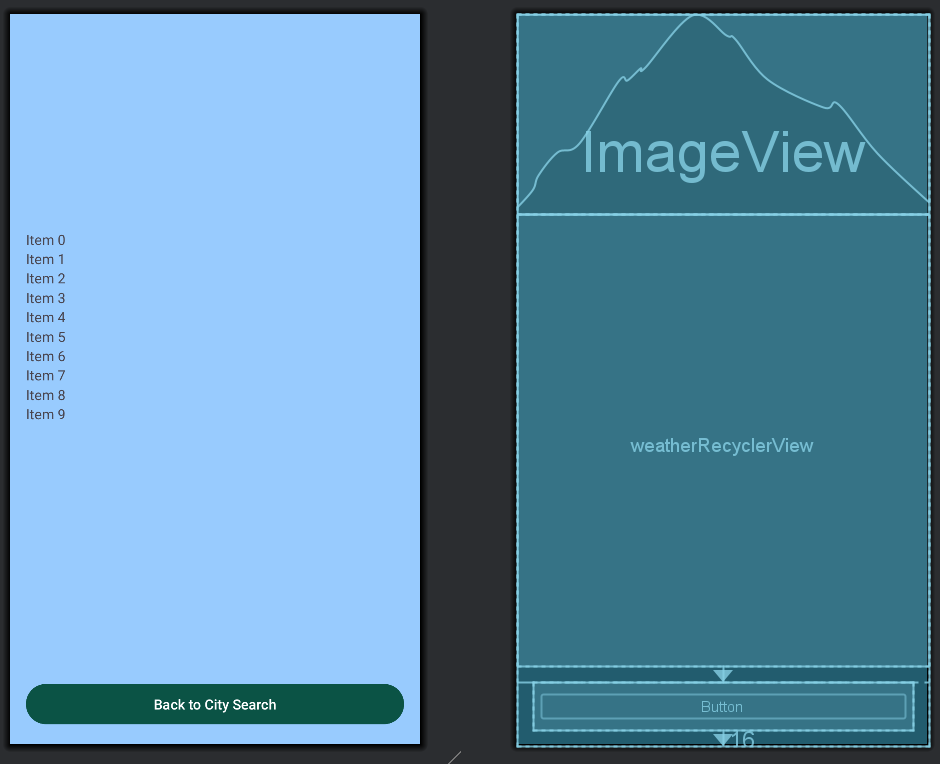
android:backgroundTint="@color/darkGreen"

android:text="Check Weather" />

</LinearLayout>

</LinearLayout>

Next, the second page currently looks like this, but the plan would be to add more images, and every ‘Item0’, and ‘Item1’ placeholder will be replaced with the forecast for the next week, with an image of the weather next to the specific date and weather forecast.



activity\_weather.xml code:

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#98cbfe">

<!-- Weather Icon/Image -->

<ImageView

android:id="@+id/weatherImage"

android:layout\_width="match\_parent"

android:layout\_height="200dp"

android:scaleType="centerInside"

android:contentDescription="Weather Image"

android:layout\_alignParentTop="true" />

<!-- Forecast List -->

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/weatherRecyclerView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/weatherImage"

android:layout\_above="@+id/backButton"

android:padding="16dp"

android:clipToPadding="false" />

<!-- Back Button -->

<Button

android:id="@+id/backButton"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:backgroundTint="@color/darkGreen"

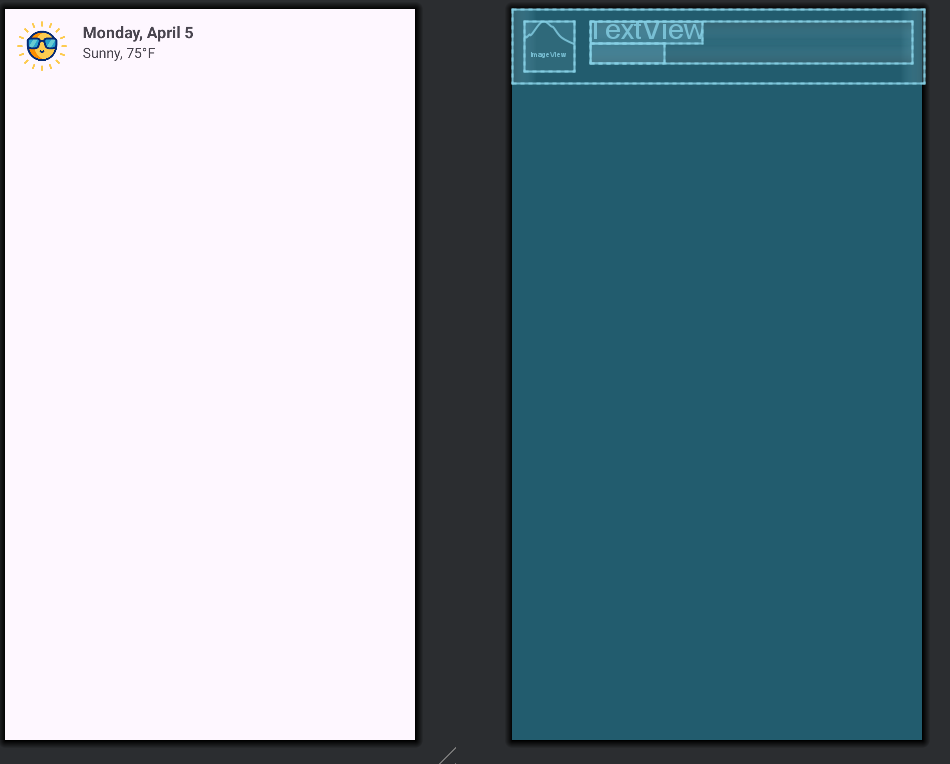
android:text="Back to City Search"

android:layout\_alignParentBottom="true"

android:layout\_margin="16dp" />

</RelativeLayout>

Last would be the actual image that we’re envisioning for the specific ‘Item0’ placeholders. There’s a solid chance we’ll change things up as we go, but for now, this is what we currently have as an idea:



item\_weather.xml:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:padding="12dp">

<!-- Weather Icon -->

<ImageView

android:id="@+id/weatherIcon"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:src="@drawable/ic\_sun"

android:contentDescription="Weather Icon"

android:layout\_marginEnd="16dp" />

<!-- Date and Weather Text -->

<LinearLayout

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:orientation="vertical">

<TextView

android:id="@+id/dateText"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Monday, April 5"

android:textSize="16sp"

android:textStyle="bold" />

<TextView

android:id="@+id/weatherDescription"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Sunny, 75°F"

android:textSize="14sp"/>

</LinearLayout>

</LinearLayout>

Our current Kotlin code, which needs to be edited and merged with our chosen API for further functionality:

WeatherAdapter.kt:  
package com.jurado.finalproject

import android.os.Bundle

import android.view.LayoutInflater

import android.view.View

import android.view.ViewGroup

import android.widget.ImageView

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

import androidx.recyclerview.widget.LinearLayoutManager

import androidx.recyclerview.widget.RecyclerView

class WeatherAdapter(private val weatherList: List<WeatherData>) :

RecyclerView.Adapter<WeatherAdapter.WeatherViewHolder>() {

class WeatherViewHolder(view: View) : RecyclerView.ViewHolder(view) {

val dateText: TextView = view.findViewById(R.id.*dateText*)

val weatherDescription: TextView = view.findViewById(R.id.*weatherDescription*)

val weatherIcon: ImageView = view.findViewById(R.id.*weatherIcon*)

}

override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): WeatherViewHolder {

val view = LayoutInflater.from(parent.*context*)

.inflate(R.layout.*item\_weather*, parent, false)

return WeatherViewHolder(view)

}

override fun onBindViewHolder(holder: WeatherViewHolder, position: Int) {

val weather = weatherList[position]

// Show "Day0", "Day1", etc. instead of the date

holder.dateText.*text* = "Day$position" // This will display 'Day0', 'Day1', etc.

holder.weatherDescription.*text* = "${weather.condition}, ${weather.temperature}°F"

// Change the icon dynamically based on weather condition

val iconRes = when (weather.condition.*lowercase*()) {

"sunny" -> R.drawable.*ic\_sun*

"cloudy" -> R.drawable.*ic\_cloud*

"rainy" -> R.drawable.*ic\_rain*

else -> R.drawable.*ic\_unknown* // Fallback icon

}

holder.weatherIcon.setImageResource(iconRes)

}

override fun getItemCount(): Int = weatherList.size

}

class WeatherActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.*activity\_weather*)

val weatherList = *listOf*(

WeatherData("Monday, April 5", "Sunny", 75),

WeatherData("Tuesday, April 6", "Cloudy", 68),

WeatherData("Wednesday, April 7", "Rainy", 60)

)

// Set the top weather image dynamically

val weatherImage = findViewById<ImageView>(R.id.*weatherImage*)

val firstWeatherCondition = weatherList.*firstOrNull*()?.condition ?: "unknown"

val iconRes = when (firstWeatherCondition.*lowercase*()) {

"sunny" -> R.drawable.*ic\_sun*

"cloudy" -> R.drawable.*ic\_cloud*

"rainy" -> R.drawable.*ic\_rain*

else -> R.drawable.*ic\_unknown*

}

weatherImage.setImageResource(iconRes)

// Set up RecyclerView

val recyclerView = findViewById<RecyclerView>(R.id.*weatherRecyclerView*)

recyclerView.*layoutManager* = LinearLayoutManager(this)

recyclerView.*adapter* = WeatherAdapter(weatherList)

}

}

data class WeatherData(val date: String, val condition: String, val temperature: Int)

MainActivity.kt:

package com.jurado.finalproject

import android.content.Intent

import android.os.Bundle

import android.widget.Button

import androidx.activity.enableEdgeToEdge

import androidx.appcompat.app.AppCompatActivity

import androidx.core.view.ViewCompat

import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

*enableEdgeToEdge*()

setContentView(R.layout.*activity\_main*)

// Insets padding (leave this as-is)

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.*main*)) **{** v, insets **->**

val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())

v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)

insets

**}**

// Add this block to handle button click

val checkWeatherButton = findViewById<Button>(R.id.*submitButton*)

checkWeatherButton.setOnClickListener **{**

val intent = Intent(this, WeatherActivity::class.*java*)

startActivity(intent)

**}**

}

}