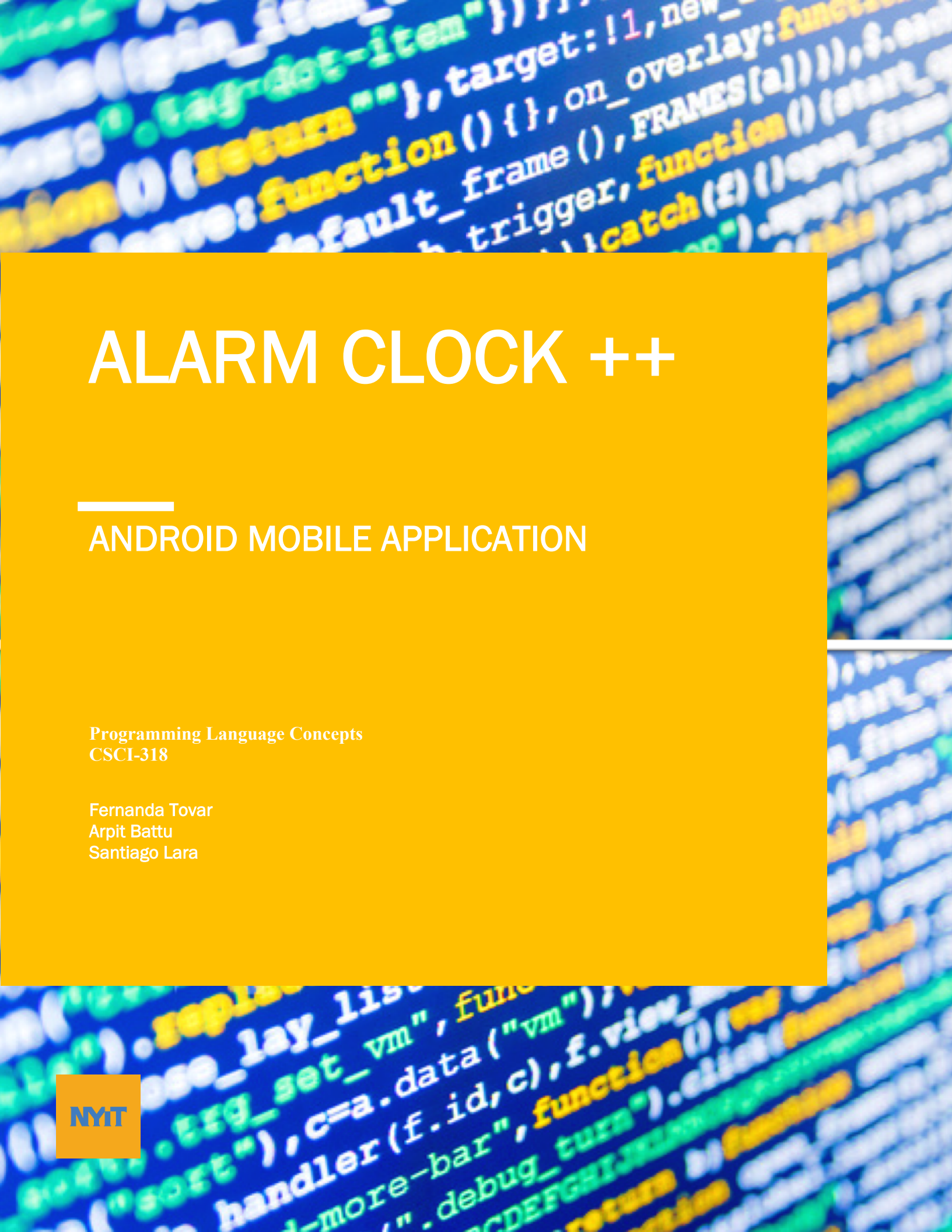


ALARM CLOCK ++

ANDROID MOBILE APPLICATION

Programming Language Concepts
CSCI-318

Fernanda Tovar
Arpit Battu
Santiago Lara

NYIT

FINAL REPORT

BACKGROUND 

Alarm Clock Application have been on a growing trend with people becoming more interested in their sleep patterns and ways to force themselves up. There are millions of applications on Google's Play Store that have a plethora of features such as sleep tracking abilities, challenges to turn off alarm, custom alarm tones/music and much more. Our Android mobile application focuses on what happens before the alarm goes off in the morning. Our focus is to bring something new and exciting like bringing a "smart" alarm that suggest a waking up time based on the person's inputs.

This project challenges us to create a mobile application that is useful for the everyday busy college student. As college students, we are overwhelmed by the loads of assignments, our work schedules and other activities. We spend restless nights and setting alarms according to the next day's schedule is the last thing that we want to do. Therefore, this application will maximize our sleep time and relax our busy minds. It's important to create an app where it sets what time to wake up based on factors such as the time to get ready and commute time. The end goal is for the user to input the time they need to be at the designated location and then the app will calculate the rest to set an accurate alarm time. You won't need to spend extra time thinking about what time to set the alarm to wake up to. All you will have to worry about is getting a good night sleep, which overall improves the productivity of college students.

WHAT SOFWARE WAS USED 

Our primary platform that we use to code was Android Studio. We use an Google Pixel 2 phone emulator to run the code and see the output. We coded with Java.

FEATURES 

- Sets an alarm based on:
  - How long it takes you to get ready.
  - Commute time from starting to end location.
    - Auto-complete address
- See all alarms
  - Displays all alarms that have been set.
    - Customize days
    - Customize alarm ring
    - Customize labels
    - Snooze Alarm
    - Dismiss Alarm
- Timer
- Stopwatch
- World clock

HOW IT WORKS

Our application heavily rely on Google Maps APIS and the Android Developers alarm clock provider. The commute time and address finder is based on Google Maps. This GPS system will give us an accurate time of how long it takes from point A to point B. It will already consider the traffic and delays, so all we must do is add on the user input of the destination and what transportation mode they use.
The Android Developers alarm clock class already configured most of the algorithm. All we had to do was implement the user inputs on how long it will take for them to get ready and what time they need to arrive at that location. Therefore, the app's final goal is to calculate what time the alarm needs to be set. The class created by Android developer sets the alarm based on the calculation from the equation below.

The two equations below show the general formula needed for an accurate alarm time:

Time needed = transportation time + get ready time + 10 minutes extra

Alarm= ETA - time needed

RESEARCH AND DOCUMENATION

Google Maps Platform

.
Maps SDK for Android, you can add maps based on Google Maps data to your application. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. You can also use API calls to add markers, polygons, and overlays to a basic map, and to change the user's view of a particular map area.

The Distance Matrix API is a service that provides travel distance and time for a matrix of origins and destinations, based on the recommended route between start and end points.

The Places SDK for Android allows you to build location-aware apps that respond contextually to the local businesses and other places near the device. This means you can build rich apps based on places that mean something to the user, to complement the straightforward geographic-based services offered by the services. Specifically we used Place Autocomplete to automatically fill in the name and/or address of a place as a user types.

Android Developer

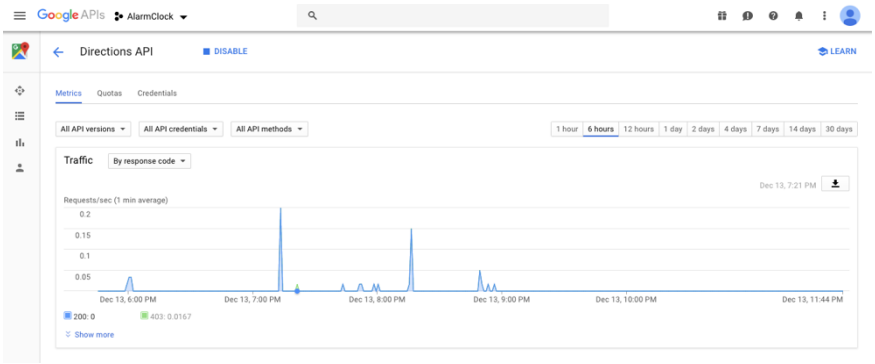
Alarm clock provider contains an Intent action and extras that can be used to start an Activity to set a new alarm or timer in an alarm clock application. Applications that wish to receive the ACTION_SET_ALARM and ACTION_SET_TIMER Intents should create an activity to handle the Intent that requires the permission com.android.alarm.permission.SET_ALARM.

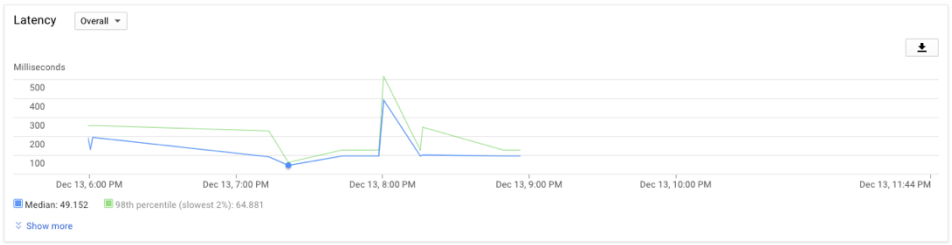
GeoCoding is the process of transforming a street address or other description of a location into a latitude, longitude coordinate. It was used before we started to implement the distance matrix API provided by Google Maps which at the end should have been more efficient.

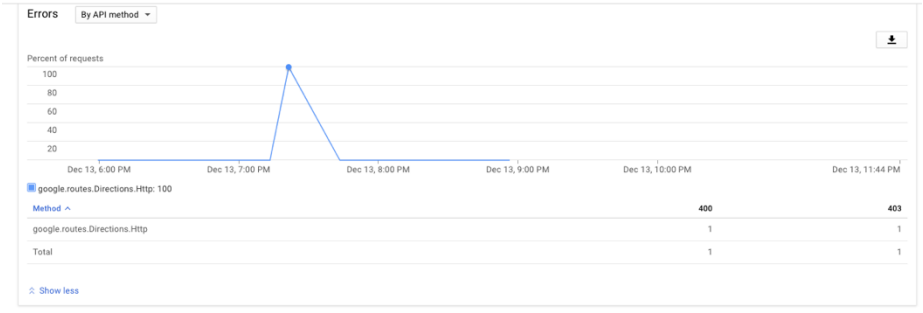
THINGS WE LEARNED

API

We learned how to get an API Key for our Android Application on a certain website. This is needed to access the Google Maps servers. We implemented this key into the <string> element in the google_maps_api.xml file in android studio. This key allowed us to further enable other APIs needed for Google Maps. For example, Maps SDK for Android, Places SDK for Android, Distance Matrix API.







XML

We had fun designing the application's look. This included layout and background. We wanted something fun and vibrant to showcase the inputs we request from the user. Each class has its own XML file attached to it. Each file comes with attributes that can be associated to a value used in the java class. The xml file is designed with a design section and a text section. They work cohesively to update any changes. With this it offers buttons (used to go to next page), spinner methods (pull down menus for transportation mode and for picking hour, minutes, pm/am), and text boxes (for headings).

XML: Buttons

```
<activity
    android:name="MyOtherActivity"
    android:label="@string/app_name">
</activity>
```

All activities must be declared in manifest, even if they do not have an intent filter assigned to them.

Figure 1 Other activity declared in manifest

```
Button btn = (Button)findViewById(R.id.open_activity_button);

btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            startActivity(new Intent(MainActivity.this, MyOtherActivity.class));
        }
});
```

Figure 2 Main activity

XML: Spinner Methods

```
package example.javatpoint.com.spinner;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements
        AdapterView.OnItemSelectedListener {
    String[] country = { "India", "USA", "China", "Japan", "Other"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
       //Getting the instance of Spinner and applying OnItemSelectedListener on it
        Spinner spin = (Spinner) findViewById(R.id.spinner);
        spin.setOnItemSelectedListener(this);

        //Creating the ArrayAdapter instance having the country list
        ArrayAdapter aa = new ArrayAdapter(this,android.R.layout.simple_spinner_
        aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_it
        //Setting the ArrayAdapter data on the Spinner
        spin.setAdapter(aa);

    }

    //Performing action onItemSelected and onNothing selected
    @Override
    public void onItemSelected(AdapterView<?
> arg0, View arg1, int position, long id) {
        Toast.makeText(getApplicationContext(),country[position] , Toast.LENGTH_L
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) {
        // TODO Auto-generated method stub
    }
}
```

Passing Java objects

Both figures were used in order to communicate between the different activities or classes. It was a way to connect the application.

```
// inside first activity
Intent intent = new Intent(MainActivity.this, DetailActivity.class);
intent.putExtra("title", "The Godfather");
intent.putExtra("year", 1972);

// inside second activity
Intent intent = getIntent();
String title = intent.getStringExtra("title");
Integer year = intent.getIntExtra("year");
```

Figure 3 example on how to pass objects between activities

In your Client class.

```
public void setString(String yourString){
    variableToStoreStringInClass = yourString;
}
```

And in your MainActivity, where you want to send the string to the Client.

```
Client yourClient = new Client();
yourClient.setString("Enter the string");
```

Figure 4example on how to pass objects from a activity to a non-activity class

Gradle

In Android studio, this is a custom build tool used to build android packages (apk file) by managing dependencies and proving custom build logic.

build.gradle

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.example.arpit.alarmclock2"
        minSdkVersion 23
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}
```

```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation 'com.google.android.gms:play-services-location:16.0.0'
    implementation 'com.google.android.gms:play-services-places:16.0.0'
    implementation 'com.google.android.gms:play-services:12.0.1'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.google.android.gms:play-services-maps:16.0.0'
    implementation 'junit:junit:4.12'
    implementation 'com.google.firebase:firebase-core:16.0.6'
    //noinspection GradleCompatible
    implementation 'com.android.support.test:runner:1.0.2'
    implementation 'com.android.support:appcompat-v7:28.0.0'
    androidTestImplementation('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.google.android.gms:play-services:12.0.1'
    implementation 'com.android.support:support-v4:28.0.0'
    implementation 'com.google.maps.android:android-maps-utils:0.5+'
}

allprojects {
    repositories {
        //...
    }

    subprojects {
        project.configurations.all {
            resolutionStrategy.eachDependency { details ->
                if (details.requested.group == 'com.google.android.gms'
                        && !details.requested.name.contains('multidex')) {
                    details.useVersion "12.0.1"
                }
            }
        }
    }
}
```

Manifest

AndroidManifest.xml. Manifest file for an android application is a resource file which contains all the details needed by the android system about the application. It is a key file that works as a bridge between the android developer and the android platform.

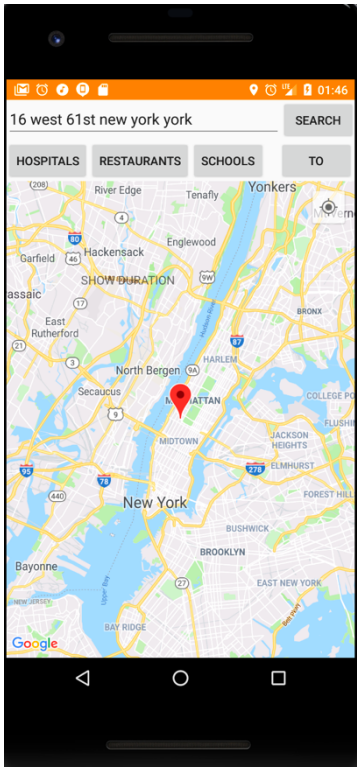
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.arpit.alarmclock2">

    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the 'MyLocation' functionality.
    -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="com.android.alarm.permission.SET_ALARM"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".DestPage"
            android:label="@string/app_name" />
        <activity
            android:name=".MethodPage"
            android:label="@string/app_name" />
        <activity
            android:name=".TimePage"
            android:label="@string/app_name" />
        <activity
            android:name=".ReadyPage"
```

```
            android:label="@string/app_name" />

        <!--
             The API key for Google Maps-based APIs is defined as a string resource.
             (See the file "res/values/google_maps_api.xml").
             Note that the API key is linked to the encryption key used to sign the APK.
             You need a different API key for each encryption key, including the release key that is used to
             sign the APK for publishing.
             You can define the keys for the debug and release targets in src/debug/ and src/release/.
        -->
        <meta-data
            android:name="AA_DB_NAME"
            android:value="local_test_db.db" />
        <meta-data
            android:name="AA_DB_VERSION"
            android:value="9" />
        <meta-data
            android:name="AA_MODELS"
            android:value="aaeu.app.datalayer.Alert , aaeu.app.datalayer.Area" />

        <!-- without this meta-data below the app don't crash but sais API key wrong -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />
        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />

        <activity android:name=".ResultView" />

        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />

        <activity
            android:name=".MapsActivity"
            android:label="@string/title_activity_maps" />
        <activity android:name=".SetAlarm"></activity>
    </application>

</manifest>
```

DIFFICULTIES WE ENCOUNTERED

Unfortunately due to a lack of time, we came across a lot of issues and setbacks. There was constant trial and error. We used stack overflow to answer many questions and fix many errors we got. We used GitHub and YouTube tutorials to learn and incorporate source code. The whole team was not experienced with Android Studio, so this ultimately set us back because we had to learn everything from scratch. Without the adequate time, we couldn't add features like incorporating Google Calendar to suggest alarms based on one's schedule. We also wanted to incorporate the constantly changing commute time based on traffic from Google Maps to update the alarm.



Features that we needed and couldn't implement at the end was the method of transportation. This was key to accurately produce a real commute time. By default, it is set to driving, so all users would only know how long it takes to get there based on driving traffic. In addition, we couldn't calculate the minutes so that meant we can only demo the change in hours and not minutes. Ultimately, the team struggle to find a way to implement the distance and duration needed for the calculation that provided an alarm time. We showed that we can get the coordinates on the result view page and that we can display the map as show above. In order to get the program running, we had to ask the user , the commute time instead of having Google Maps implement that for us.

REFERENCES

Google Maps Platform documentation
https://developers.google.com/maps/documentation/

Android Developer: Alarm Clock
https://developers.google.com/maps/documentation/

Alarm Clock source code- YouTube video
https://www.youtube.com/watch?v=LsKjw-IJQpI
Maps- Github source code
https://github.com/priyankapakhale/GoogleMapsNearbyPlacesDemo/blob/master/app/src/main/java/com/example/priyanka/mapsdemo/MapsActivity.java

Geocoding
https://developer.android.com/reference/android/location/Geocoder

How to pass objects between activities
https://en.proft.me/2017/02/28/pass-object-between-activities-android-parcelable/

How to pass objects from activity to class
https://stackoverflow.com/questions/45137811/passing-string-value-from-activity-class-to-a-non-activity-class

Gradle
https://stackoverflow.com/questions/16754643/what-is-gradle-in-android-studio

Android Manifest
https://javapapers.com/android/android-manifest/

Button
https://stackoverflow.com/questions/24610527/how-do-i-get-a-button-to-open-another-activity-in-android-studio
Spinner Method
https://www.javatpoint.com/android-spinner-example

WHAT EACH MEMBER WORKED ON

Fernanda Tovar

- Bringing Report together
- Alarm Clock class

Arpit Battu

- Editing presentation and report
- Geocoding class and Maps classes

Santiago Lara

- Creating presentation
- Setting up layouts with XML files that include buttons, spinner methods (hour, minute and transportation)

APPLICATION SCREENSHOTS

Figure 1 homepage with 2 options. First one leads to figure 2 and ends with Figure 8. Second one leads directly to Figure 8.

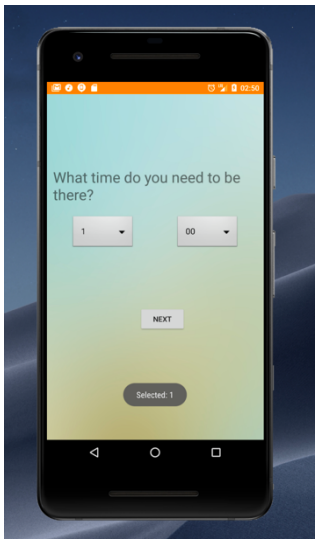


Figure 2 input: time to be at location

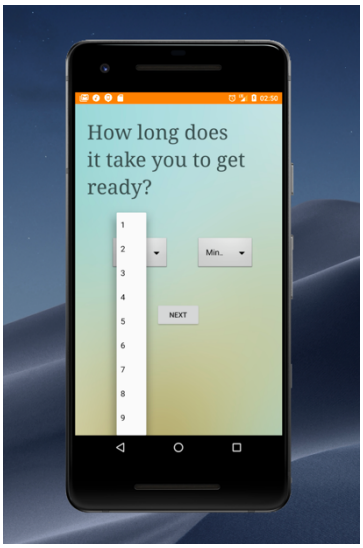


Figure 3 input: time to get ready

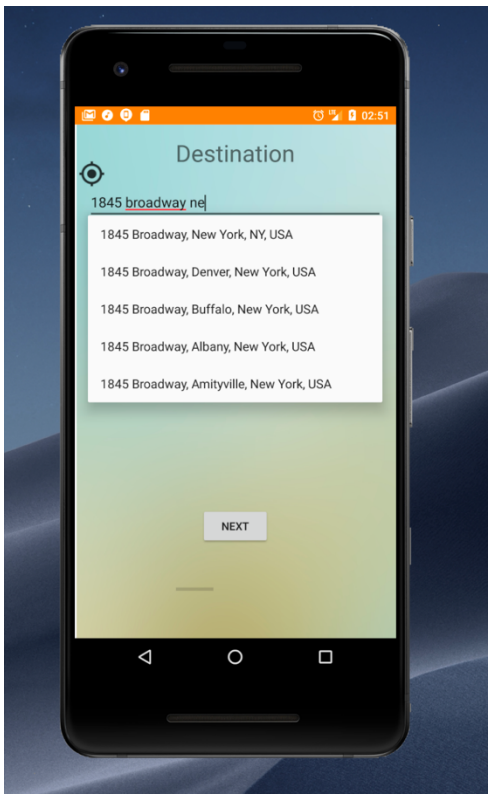


Figure 4 input: To/ from address with autocomplete

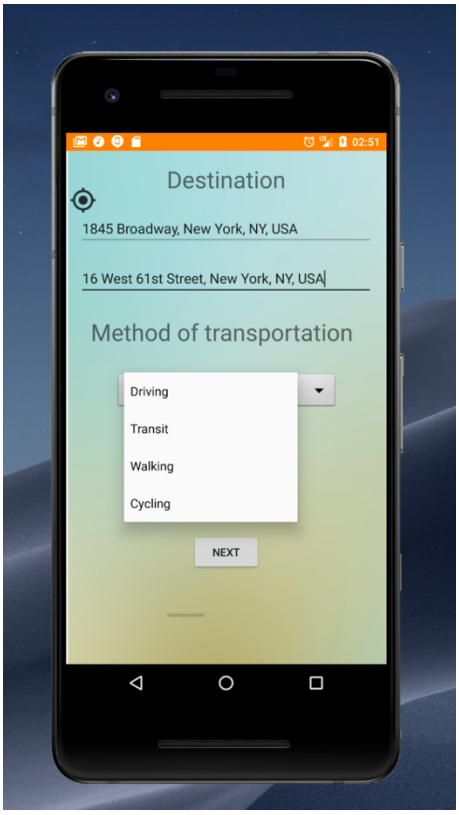


Figure 5 method of transportation

Figure 6 Filler: Page to display long/lat coordinates

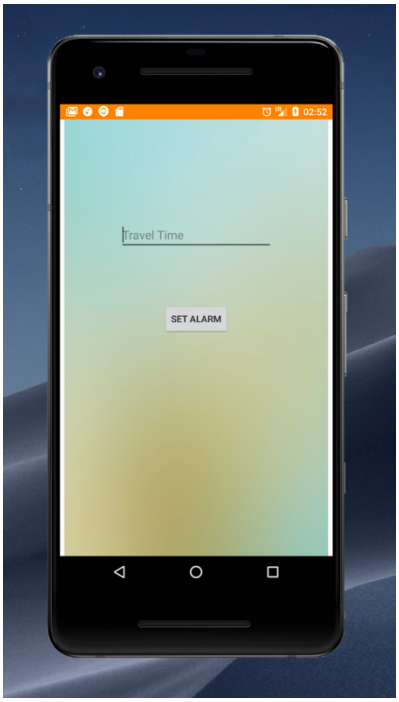


Figure 7 filler: substitute for lack of maps to implement automatic commute time

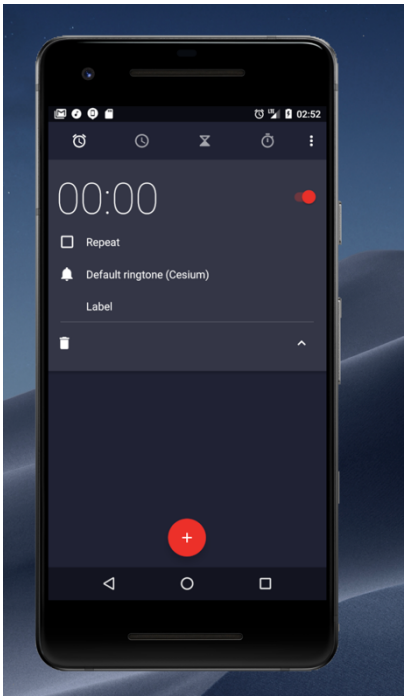


Figure 7 displays all alarms, pick ringtone, repeat days

CODE SCREENSHOTS

Activities

MainActivity.java

This main activity ties all the activities together.

```
package com.example.arpit.alarmclock2;

import android.app.Dialog;
import android.content.Intent;
import android.os.Bundle;
import android.provider.AlarmClock;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.ImageButton;
import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GoogleApiAvailability;

import java.util.Map;


public class MainActivity extends AppCompatActivity {
    private static ImageButton newAlarmButton;
    private static ImageButton seeAlarmsButton;
    private static final String TAG = "MainActivity";

    private static final int ERROR_DIALOG_REQUEST = 9001;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if(isServicesOK()){
            init();
        }


        OnClickButtonListenerMaps();
```

```
        OnClickButtonListenerAllAlarms();
    }


    public boolean isServicesOK(){
        Log.d(TAG, "isServicesOK: checking google services version");

        int available =
GoogleApiAvailability.getInstance().isGooglePlayServicesAvailable(MainActivity.this);

        if(available == ConnectionResult.SUCCESS){
            //everything is fine and the user can make map requests
            Log.d(TAG, "isServicesOK: Google Play Services is working");
            return true;
        }
        else if(GoogleApiAvailability.getInstance().isUserResolvableError(available)){
            //an error occured but we can resolve it
            Log.d(TAG, "isServicesOK: an error occured but we can fix it");
            Dialog dialog = GoogleApiAvailability.getInstance().getErrorDialog(MainActivity.this, available,
ERROR_DIALOG_REQUEST);
            dialog.show();
        }else{
            Toast.makeText(this, "You can't make map requests", Toast.LENGTH_SHORT).show();
        }
        return false;
    }




    private void init(){
        newAlarmButton = findViewById(R.id.newAlarmButton);
        newAlarmButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, TimePage.class);
                startActivity(intent);
            }
        });
    }
    public void OnClickButtonListenerMaps() {
        newAlarmButton = findViewById(R.id.mapsButton);
        newAlarmButton.setOnClickListener(
                new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Intent intent = new Intent(MainActivity.this, MapsActivity.class);
                        startActivity(intent);
                    }
                }
        );
```

```
    }
    public void OnClickButtonListenerAllAlarms() {
        final int hour = 0;
        final int min = 0;
        seeAlarmsButton = findViewById(R.id.seeAlarmsButton);
        seeAlarmsButton.setOnClickListener(
                new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                       // Intent intent = new Intent(MainActivity.this, MapsActivity.class);
                        //startActivity(intent);
                        Intent intent = new Intent(AlarmClock.ACTION_SET_ALARM);
                        intent.putExtra(AlarmClock.EXTRA_HOUR, hour);
                        intent.putExtra(AlarmClock.EXTRA_MINUTES, min);
                        startActivity(intent);
                    }
                }
        );

    }

}
```

MapsActivity.java

This maps activity integrates GoogleMaps and displays a map and the other features.

```
package com.example.arpit.alarmclock2;

import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.os.Build;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;
import android.support.v4.content.ContextCompat;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;
```

```
import com.google.android.gms.common.GoogleApiAvailability;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.LocationListener;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;

import java.io.IOException;
import java.util.List;



public class MapsActivity extends FragmentActivity implements OnMapReadyCallback,
        GoogleApiClient.ConnectionCallbacks,
        GoogleApiClient.OnConnectionFailedListener,
        LocationListener,
        GoogleMap.OnMarkerClickListener,
        GoogleMap.OnMarkerDragListener
{

    private GoogleMap mMap;
    GoogleApiClient mGoogleApiClient;
    Location mLastLocation;
    Marker mCurrLocationMarker;
    LocationRequest mLocationRequest;
    int PROXIMITY_RADIUS = 10000;
    double latitude, longitude;
    double end_latitude, end_longitude;
    public String duration, distance;
    public TextView durationText;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);

        if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            checkLocationPermission();
        }

        //Check if Google Play Services Available or not
        if (!CheckGooglePlayServices()) {
            Log.d("onCreate", "Finishing test case since Google Play Services are not available");
            finish();
```

```
    }
    else {
        Log.d("onCreate","Google Play Services available.");
    }

    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);
    mapFragment.getMapAsync(this);

    GetDirectionsData d = new GetDirectionsData();
    duration = d.getDurationMethod();



}

private boolean CheckGooglePlayServices() {
    GoogleApiAvailability googleAPI = GoogleApiAvailability.getInstance();
    int result = googleAPI.isGooglePlayServicesAvailable(this);
    if(result != ConnectionResult.SUCCESS) {
        if(googleAPI.isUserResolvableError(result)) {
            googleAPI.getErrorDialog(this, result,
                    0).show();
        }
        return false;
    }
    return true;
}


/**
 * Manipulates the map once available.
 * This callback is triggered when the map is ready to be used.
 * This is where we can add markers or lines, add listeners or move the camera. In this case,
 * we just add a marker near Sydney, Australia.
 * If Google Play services is not installed on the device, the user will be prompted to install
 * it inside the SupportMapFragment. This method will only be triggered once the user has
 * installed Google Play services and returned to the app.
 */
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    //Initialize Google Play Services
    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (ContextCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)
                == PackageManager.PERMISSION_GRANTED) {
            buildGoogleApiClient();
            mMap.setMyLocationEnabled(true);
        }
```

```
    } else {
        buildGoogleApiClient();
        mMap.setMyLocationEnabled(true);
    }

    mMap.setOnMarkerDragListener(this);
    mMap.setOnMarkerClickListener(this);



}



protected synchronized void buildGoogleApiClient() {
    mGoogleApiClient = new GoogleApiClient.Builder(this)
            .addConnectionCallbacks(this)
            .addOnConnectionFailedListener(this)
            .addApi(LocationServices.API)
            .build();
    mGoogleApiClient.connect();
}

public void onClick(View v)
{
    Object dataTransfer[] = new Object[2];
    GetNearbyPlacesData getNearbyPlacesData = new GetNearbyPlacesData();


    switch(v.getId()) {
        case R.id.B_search: {
            EditText tf_location = (EditText) findViewById(R.id.TF_location);
            String location = tf_location.getText().toString();
            List<Address> addressList = null;
            MarkerOptions markerOptions = new MarkerOptions();
            Log.d("location = ", location);

            if (!location.equals("")) {
                Geocoder geocoder = new Geocoder(this);
                try {
                    addressList = geocoder.getFromLocationName(location, 5);

                } catch (IOException e) {
                    e.printStackTrace();
                }

                if (addressList != null) {
                    for (int i = 0; i < addressList.size(); i++) {
                        Address myAddress = addressList.get(i);
                        LatLng latLng = new LatLng(myAddress.getLatitude(), myAddress.getLongitude());
                        markerOptions.position(latLng);
                        mMap.addMarker(markerOptions);
```

```
                    mMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
                }
            }

        }
    }
    break;
    case R.id.B_hospital:
        mMap.clear();
        String hospital = "hospital";
        String url = getUrl(latitude, longitude, hospital);

        dataTransfer[0] = mMap;
        dataTransfer[1] = url;

        getNearbyPlacesData.execute(dataTransfer);
        Toast.makeText(MapsActivity.this, "Showing Nearby Hospitals",
Toast.LENGTH_LONG).show();
        break;

    case R.id.B_restaurant:
        mMap.clear();
        dataTransfer = new Object[2];
        String restaurant = "restaurant";
        url = getUrl(latitude, longitude, restaurant);
        getNearbyPlacesData = new GetNearbyPlacesData();
        dataTransfer[0] = mMap;
        dataTransfer[1] = url;

        getNearbyPlacesData.execute(dataTransfer);
        Toast.makeText(MapsActivity.this, "Showing Nearby Hospitals",
Toast.LENGTH_LONG).show();
        break;
    case R.id.B_school:
        mMap.clear();
        String school = "school";
        dataTransfer = new Object[2];
        url = getUrl(latitude, longitude, school);
        getNearbyPlacesData = new GetNearbyPlacesData();
        dataTransfer[0] = mMap;
        dataTransfer[1] = url;

        getNearbyPlacesData.execute(dataTransfer);
        Toast.makeText(MapsActivity.this, "Showing Nearby Hospitals",
Toast.LENGTH_LONG).show();
        break;

    case R.id.B_to:
        dataTransfer = new Object[3];
        url = getDirectionsUrl();
        GetDirectionsData getDirectionsData = new GetDirectionsData();
        dataTransfer[0] = mMap;
```

```
                dataTransfer[1] = url;
                dataTransfer[2] = new LatLng(end_latitude, end_longitude);
                getDirectionsData.execute(dataTransfer);

                break;

            }
        }

        private String getDirectionsUrl()
        {
            StringBuilder googleDirectionsUrl = new
StringBuilder("https://maps.googleapis.com/maps/api/directions/json?");
            googleDirectionsUrl.append("origin="+latitude+","+longitude);
            googleDirectionsUrl.append("&destination="+end_latitude+","+end_longitude);
            googleDirectionsUrl.append("&key="+"AIzaSyCGCKFAzPiIDGhyniiKT1wu5G-6ygDqtq8");

            return googleDirectionsUrl.toString();
        }

        private String getUrl(double latitude, double longitude, String nearbyPlace)
        {
            StringBuilder googlePlacesUrl = new
StringBuilder("https://maps.googleapis.com/maps/api/directions/json?");
            googlePlacesUrl.append("location=" + latitude + "," + longitude);
            googlePlacesUrl.append("&radius=" + PROXIMITY_RADIUS);
            googlePlacesUrl.append("&type=" + nearbyPlace);
            googlePlacesUrl.append("&sensor=true");
            googlePlacesUrl.append("&key=" + "AIzaSyCGCKFAzPiIDGhyniiKT1wu5G-6ygDqtq8");
            Log.d("getUrl", googlePlacesUrl.toString());
            return (googlePlacesUrl.toString());
        }



        @Override
        public void onConnected(Bundle bundle) {
            mLocationRequest = new LocationRequest();
            mLocationRequest.setInterval(1000);
            mLocationRequest.setFastestInterval(1000);
            mLocationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY)
;
            if (ContextCompat.checkSelfPermission(this,
                    Manifest.permission.ACCESS_FINE_LOCATION)
                    == PackageManager.PERMISSION_GRANTED) {
                LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient,
mLocationRequest, this);
            }
        }
```

```
    @Override
    public void onConnectionSuspended(int i) {

    }

    @Override
    public void onLocationChanged(Location location) {
        Log.d("onLocationChanged", "entered");

        mLastLocation = location;
        if (mCurrLocationMarker != null) {
            mCurrLocationMarker.remove();
        }

        latitude = location.getLatitude();
        longitude = location.getLongitude();


        LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());
        MarkerOptions markerOptions = new MarkerOptions();
        markerOptions.position(latLng);
        markerOptions.draggable(true);
        markerOptions.title("Current Position");
        markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_MA
GENTA));
        mCurrLocationMarker = mMap.addMarker(markerOptions);

        //move map camera
        mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
        mMap.animateCamera(CameraUpdateFactory.zoomTo(11));


        Toast.makeText(MapsActivity.this,"Your Current Location", Toast.LENGTH_LONG).show();


        //stop location updates
        if (mGoogleApiClient != null) {
            LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient, this);
            Log.d("onLocationChanged", "Removing Location Updates");
        }

    }

    @Override
    public void onConnectionFailed(ConnectionResult connectionResult) {

    }

    public static final int MY_PERMISSIONS_REQUEST_LOCATION = 99;
    public boolean checkLocationPermission(){
        if (ContextCompat.checkSelfPermission(this,
```

```
            Manifest.permission.ACCESS_FINE_LOCATION)
            != PackageManager.PERMISSION_GRANTED) {

        // Asking user if explanation is needed
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,
                Manifest.permission.ACCESS_FINE_LOCATION)) {

            // Show an explanation to the user *asynchronously* -- don't block
            // this thread waiting for the user's response! After the user
            // sees the explanation, try again to request the permission.

            //Prompt the user once explanation has been shown
            ActivityCompat.requestPermissions(this,
                    new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                    MY_PERMISSIONS_REQUEST_LOCATION);


        } else {
            // No explanation needed, we can request the permission.
            ActivityCompat.requestPermissions(this,
                    new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                    MY_PERMISSIONS_REQUEST_LOCATION);
        }
        return false;
    } else {
        return true;
    }
}

@Override
public void onRequestPermissionsResult(int requestCode,
                                       String permissions[], int[] grantResults) {
    switch (requestCode) {
        case MY_PERMISSIONS_REQUEST_LOCATION: {
            // If request is cancelled, the result arrays are empty.
            if (grantResults.length > 0
                    && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

                // permission was granted. Do the
                // contacts-related task you need to do.
                if (ContextCompat.checkSelfPermission(this,
                        Manifest.permission.ACCESS_FINE_LOCATION)
                        == PackageManager.PERMISSION_GRANTED) {

                    if (mGoogleApiClient == null) {
                        buildGoogleApiClient();
                    }
                    mMap.setMyLocationEnabled(true);
                }

            } else {
```

```
                // Permission denied, Disable the functionality that depends on this permission.
                Toast.makeText(this, "permission denied", Toast.LENGTH_LONG).show();
            }
            return;
        }

    }
}


@Override
public boolean onMarkerClick(Marker marker) {
    marker.setDraggable(true);
    return false;
}

@Override
public void onMarkerDragStart(Marker marker) {

}

@Override
public void onMarkerDrag(Marker marker) {

}

@Override
public void onMarkerDragEnd(Marker marker) {
    end_latitude = marker.getPosition().latitude;
    end_longitude =  marker.getPosition().longitude;

    Log.d("end_lat",""+end_latitude);
    Log.d("end_lng",""+end_longitude);
}


}
```

DestPage.java

This destination page asks the user for to/ from addresses and method of transportation.

```
package com.example.arpit.alarmclock2;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
```

```
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.PendingResult;
import com.google.android.gms.common.api.ResultCallback;
import com.google.android.gms.location.places.Place;
import com.google.android.gms.location.places.PlaceBuffer;
import com.google.android.gms.location.places.Places;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.LatLngBounds;



public class DestPage extends AppCompatActivity implements
        GoogleApiClient.OnConnectionFailedListener,
        GoogleApiClient.ConnectionCallbacks {
    private static Button destNextButton;
    private static final int GOOGLE_API_CLIENT_ID = 0;
    private static String LOG_TAG = "DestPage";
    private AutoCompleteTextView fromDest;
    private AutoCompleteTextView destAddress;
    private TextView destText;
    private GoogleApiClient mGoogleApiClient;
    private PlaceArrayAdapter mPlaceArrayAdapter;
    private static final LatLngBounds BOUNDS_MOUNTAIN_VIEW = new LatLngBounds(
            new LatLng(37.398160, -122.180831), new LatLng(37.430610, -121.972090));

    private String hourTime, minTime, ampmTime
            ,hourReady, minReady;


    @Override
    protected void onCreate(Bundle savedInstance) {
        super.onCreate(savedInstance);
        setContentView(R.layout.dest_page);

        Bundle bundle = getIntent().getExtras();
        hourTime = bundle.getString("hourTime");
        minTime = bundle.getString("minTime");
```

```
        ampmTime = bundle.getString("ampmTime");

        hourReady = bundle.getString("hourReady");
        minReady = bundle.getString("minReady");

        mGoogleApiClient = new GoogleApiClient.Builder(DestPage.this)
                .addApi(Places.GEO_DATA_API)
                .enableAutoManage(this, GOOGLE_API_CLIENT_ID, this)
                .addConnectionCallbacks(this)
                .build();
        fromDest = (AutoCompleteTextView) findViewById(R.id
                .fromDest);
        destAddress = (AutoCompleteTextView) findViewById(R.id
                .destAddress);
        fromDest.setThreshold(3);
        destAddress.setThreshold(3);
        destText = (TextView) findViewById(R.id.destText);
        fromDest.setOnItemClickListener(mAutocompleteClickListener);
        mPlaceArrayAdapter = new PlaceArrayAdapter(this, android.R.layout.simple_list_item_1,
                BOUNDS_MOUNTAIN_VIEW, null);
        fromDest.setAdapter(mPlaceArrayAdapter);
        destAddress.setAdapter(mPlaceArrayAdapter);

        OnClickListener();
        //get the spinner from the xml.
        Spinner transSpinner = findViewById(R.id.methodSpinner);
        //create a list of items for the spinner.
        String[] transMethod = new String[]{"Driving", "Transit", "Walking", "Cycling"};
        //create an adapter to describe how the items are displayed, adapters are used in several places in
android.
        //There are multiple variations of this, but this is the basic variant.
        ArrayAdapter<String> transAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, transMethod);
        //set the spinners adapter to the previously created one.
        transSpinner.setAdapter(transAdapter);

        //fer:  retrieve mode  to use in calculate distance
       // String mode=bundle.getString("Transportation mode");


    }

    public void OnClickListener(){
        destNextButton = findViewById(R.id.destNextButton);
        destNextButton.setOnClickListener(
                new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
```

```
                Bundle bundle = getIntent().getExtras();
                String address = bundle.getString("address");
                //testText.setText(address);
                Intent intent = new Intent(DestPage.this, ResultView.class);
                //Intent sentData = new Intent(DestPage.this, ResultView.class);
               // startActivity(intent);
                intent.putExtra("hourTime", hourTime);
                intent.putExtra("minTime", minTime);
                intent.putExtra("ampmTime", ampmTime);
                intent.putExtra("hourReady", hourReady);
                intent.putExtra("minReady", minReady);
                String fromDestText = fromDest.getText().toString();
                String destAddressText = destAddress.getText().toString();
                intent.putExtra("fromDest", fromDestText);
                intent.putExtra("destAddress", destAddressText);
                //startActivity(sentData);
                startActivity(intent);



            }
        }
    );
}

private AdapterView.OnItemClickListener mAutocompleteClickListener
        = new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        final PlaceArrayAdapter.PlaceAutocomplete item = mPlaceArrayAdapter.getItem(position);
        final String placeId = String.valueOf(item.placeId);
        Log.i(LOG_TAG, "Selected: " + item.description);
        PendingResult<PlaceBuffer> placeResult = Places.GeoDataApi
                .getPlaceById(mGoogleApiClient, placeId);
        placeResult.setResultCallback(mUpdatePlaceDetailsCallback);
        Log.i(LOG_TAG, "Fetching details for ID: " + item.placeId);
    }
};
private ResultCallback<PlaceBuffer> mUpdatePlaceDetailsCallback
        = new ResultCallback<PlaceBuffer>() {
    @Override
    public void onResult(PlaceBuffer places) {
        if (!places.getStatus().isSuccess()) {
            Log.e(LOG_TAG, "Place query did not complete. Error: " +
                    places.getStatus().toString());
            places.release(); //fer
            return;
        }
        // Selecting the first object buffer.
```

```
            final Place place = places.get(0);
            CharSequence attributions = places.getAttributions();

        }
    };

    @Override
    public void onConnected(Bundle bundle) {
        mPlaceArrayAdapter.setGoogleApiClient(mGoogleApiClient);
        Log.i(LOG_TAG, "Google Places API connected.");

    }

    @Override
    public void onConnectionFailed(ConnectionResult connectionResult) {
        Log.e(LOG_TAG, "Google Places API connection failed with error code: "
                + connectionResult.getErrorCode());

        Toast.makeText(this,
                "Google Places API connection failed with error code:" +
                        connectionResult.getErrorCode(),
                Toast.LENGTH_LONG).show();
    }

    @Override
    public void onConnectionSuspended(int i) {
        mPlaceArrayAdapter.setGoogleApiClient(null);
        Log.e(LOG_TAG, "Google Places API connection suspended.");
    }
}
```

ResultView.java

This result view page displays the values from the geocoding class (display coordinates) and what the user picked.

```
package com.example.arpit.alarmclock2;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class ResultView extends AppCompatActivity {
    private String hourTime, minTime, ampmTime
```

```
                ,hourReady, minReady
                ,fromDest, destAddress;

private TextView hourTimeText, minTimeText, amTimeText
                ,hourReadyText, minReadyText
                ,fromDestText, destTextView
                ,addressTV, latLongTV, editText;

private Button addressButton, nextButton;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.result_view);
    Bundle bundle = getIntent().getExtras();
    hourTime = bundle.getString("hourTime");
    minTime = bundle.getString("minTime");
   // ampmTime = bundle.getString("ampmTime");

    hourReady = bundle.getString("hourReady");
    minReady = bundle.getString("minReady");

    fromDest = bundle.getString("fromDest");
    destAddress = bundle.getString("destAddress");

    hourTimeText = (TextView) findViewById(R.id.hourTimeText);
    minTimeText = (TextView) findViewById(R.id.minTimeText);
    //amTimeText = (TextView) findViewById(R.id.amTimeText);

    hourReadyText = (TextView) findViewById(R.id.hourReadyText);
    minReadyText = (TextView) findViewById(R.id.minReadyText);

    fromDestText = (TextView) findViewById(R.id.fromDestText);
    destTextView = (TextView) findViewById(R.id.destTextView);

    hourTimeText.setText("Hour: "+hourTime);
    minTimeText.setText("Minute: "+ minTime);
    // amTimeText.setText(ampmTime);

    hourReadyText.setText("Hour: "+hourReady);
    minReadyText.setText("Minute: "+minReady);

    fromDestText.setText(fromDest);
    destTextView.setText(destAddress);


    addressButton = (Button) findViewById(R.id.addressButton);
    nextButton = (Button) findViewById(R.id.nextButton);
    addressButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View arg0) {
```

```
            String fromAddress = fromDest;
            String toAddress = destAddress;

            GeocodingLocation locationAddress = new GeocodingLocation();
            locationAddress.getFromAddressFromLocation(fromAddress,
                    getApplicationContext(), new FromGeocoderHandler());
            locationAddress.getToAddressFromLocation(destAddress,
                    getApplicationContext(), new ToGeocoderHandler());


        }
    });

    nextButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View arg0) {

            Intent intent = new Intent(ResultView.this, SetAlarm.class);
            intent.putExtra("hourTimeString", hourTime);
            intent.putExtra("minTimeString", minTime);
            intent.putExtra("hourReadyString", hourReady);
            intent.putExtra("minReadyString", minReady);
            startActivity(intent);

        }
    });

}

private class FromGeocoderHandler extends Handler {
    @Override
    public void handleMessage(Message message) {
        String locationAddress;
        switch (message.what) {
            case 1:
                Bundle bundle = message.getData();
                locationAddress = bundle.getString("address");
                break;
            default:
                locationAddress = null;
        }
        fromDestText.setText(locationAddress);
    }
}
private class ToGeocoderHandler extends Handler {
    @Override
    public void handleMessage(Message message) {
        String locationAddress;
        switch (message.what) {
            case 1:
                Bundle bundle = message.getData();
                locationAddress = bundle.getString("address");
```

```
                break;
            default:
                locationAddress = null;
        }
        destTextView.setText(locationAddress);
    }
  }


}
```

MethodPage.java

This method page displays a spinner for different modes of transportations.

```
package com.example.arpit.alarmclock2;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;

public class MethodPage extends AppCompatActivity{
    private static Button methodNextButton;
    public TextView testText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.method_page);
        OnClickListener();

        testText = findViewById(R.id.testText);

        //get the spinner from the xml.
        Spinner transSpinner = findViewById(R.id.methodSpinner);
        //create a list of items for the spinner.
        String[] transMethod = new String[]{"Driving", "Transit", "Walking", "Cycling"};
        //create an adapter to describe how the items are displayed, adapters are used in several places
in android.
        //There are multiple variations of this, but this is the basic variant.
        ArrayAdapter<String> transAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, transMethod);
        //set the spinners adapter to the previously created one.
```

```
        transSpinner.setAdapter(transAdapter);
    }
    public void OnClickListener(){
        methodNextButton = findViewById(R.id.methodNextButton);
        methodNextButton.setOnClickListener(
                new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        /*
                        Bundle bundle = getIntent().getExtras();
                        String address = bundle.getString("address");
                        testText.setText(address); */
                        Intent intent = new Intent(MethodPage.this, TimePage.class);
                        startActivity(intent);
                    }
                }
        );
    }
}
```

PlaceArrayAdapter.java

This extends the next java class.

```
package com.example.arpit.alarmclock2;

import android.content.Context;
import android.util.Log;
import android.widget.ArrayAdapter;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.Toast;

import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.PendingResult;
import com.google.android.gms.common.api.Status;
import com.google.android.gms.location.places.AutocompleteFilter;
import com.google.android.gms.location.places.AutocompletePrediction;
import com.google.android.gms.location.places.AutocompletePredictionBuffer;
import com.google.android.gms.location.places.Places;
import com.google.android.gms.maps.model.LatLngBounds;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.concurrent.TimeUnit;

public class PlaceArrayAdapter extends ArrayAdapter<PlaceArrayAdapter.PlaceAutocomplete>
implements Filterable {
```

```
private static final String TAG = "PlaceArrayAdapter";
private GoogleApiClient mGoogleApiClient;
private AutocompleteFilter mPlaceFilter;
private LatLngBounds mBounds;
private ArrayList<PlaceAutocomplete> mResultList;


public PlaceArrayAdapter(Context context, int resource, LatLngBounds bounds,
                         AutocompleteFilter filter) {
    super(context, resource);
    mBounds = bounds;
    mPlaceFilter = filter;
}

public void setGoogleApiClient(GoogleApiClient googleApiClient) {
    if (googleApiClient == null || !googleApiClient.isConnected()) {
        mGoogleApiClient = null;
    } else {
        mGoogleApiClient = googleApiClient;
    }
}

@Override
public int getCount() {
    return mResultList.size();
}

@Override
public PlaceAutocomplete getItem(int position) {
    return mResultList.get(position);
}


@Override
public Filter getFilter() {
    Filter filter = new Filter() {
        @Override
        protected FilterResults performFiltering(CharSequence constraint) {

            FilterResults results = new FilterResults();
            if (constraint != null) {
                mResultList = getPredictions(constraint);

                if (mResultList != null) {
                    results.values = mResultList;
                    results.count = mResultList.size();
                }
            }

            return results;
        }
```

```
        @Override
        protected void publishResults(CharSequence constraint, FilterResults results) {
            if (results != null && results.count > 0) {
                // The API returned at least one result, update the data.
                notifyDataSetChanged();
            } else {
                // The API did not return any results, invalidate the data set.
                notifyDataSetInvalidated();
            }
        }
    };
    return filter;
}


private ArrayList<PlaceAutocomplete> getPredictions(CharSequence constraint) {

    if (mGoogleApiClient != null) {

        Log.i(TAG, "Executing autocomplete query for: " + constraint);

        PendingResult<AutocompletePredictionBuffer> results = Places.GeoDataApi
                .getAutocompletePredictions(mGoogleApiClient, constraint.toString(),
                        mBounds, mPlaceFilter);

        // Wait for predictions, set the timeout.
        AutocompletePredictionBuffer autocompletePredictions = results
                .await(60, TimeUnit.SECONDS);

        final Status status = autocompletePredictions.getStatus();
        if (!status.isSuccess()) {
            Toast.makeText(getContext(), "Error: " + status.toString(),
                    Toast.LENGTH_SHORT).show();
            Log.e(TAG, "Error getting place predictions: " + status
                    .toString());
            autocompletePredictions.release();
            return null;
        }

        Log.i(TAG, "Query completed. Received " + autocompletePredictions.getCount()
                + " predictions.");
        Iterator<AutocompletePrediction> iterator = autocompletePredictions.iterator();
        ArrayList resultList = new ArrayList<>(autocompletePredictions.getCount());
        while (iterator.hasNext()) {
            AutocompletePrediction prediction = iterator.next();
            resultList.add(new PlaceAutocomplete(prediction.getPlaceId(),
                    prediction.getFullText(null)));
        }
        // Buffer release
        autocompletePredictions.release();
        return resultList;
```

```
        }

        Log.e(TAG, "Google API client is not connected.");
        return null;
    }

    class PlaceAutocomplete {

        public CharSequence placeId;
        public CharSequence description;

        PlaceAutocomplete(CharSequence placeId, CharSequence description) {
            this.placeId = placeId;
            this.description = description;
        }

        @Override
        public String toString() {
            return description.toString();
        }
    }
}
```

PlaceAutocompleteAdapter.java

This placeAutocompleteAdapter takes care of suggesting addresses as the user types in.

```
package com.example.arpit.alarmclock2;

import android.content.Context;
import android.graphics.Typeface;
import android.text.style.CharacterStyle;
import android.text.style.StyleSpan;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.PendingResult;
import com.google.android.gms.common.api.Status;
import com.google.android.gms.common.data.DataBufferUtils;
import com.google.android.gms.location.places.AutocompleteFilter;
import com.google.android.gms.location.places.AutocompletePrediction;
import com.google.android.gms.location.places.AutocompletePredictionBuffer;
```

```
import com.google.android.gms.location.places.Places;
import com.google.android.gms.maps.model.LatLngBounds;

import java.util.ArrayList;
import java.util.concurrent.TimeUnit;

/**
 * Adapter that handles Autocomplete requests from the Places Geo Data API.
 * {@link AutocompletePrediction} results from the API are frozen and stored directly in this
 * adapter. (See {@link AutocompletePrediction#freeze()}.)
 * <p>
 * Note that this adapter requires a valid {@link
com.google.android.gms.common.api.GoogleApiClient}.
 * The API client must be maintained in the encapsulating Activity, including all lifecycle and
 * connection states. The API client must be connected with the {@link Places#GEO_DATA_API}
API.
 */
public class PlaceAutocompleteAdapter
        extends ArrayAdapter<AutocompletePrediction> implements Filterable {

    private static final String TAG = "PlaceAutoCompleteAd";
    private static final CharacterStyle STYLE_BOLD = new StyleSpan(Typeface.BOLD);
    /**
     * Current results returned by this adapter.
     */
    private ArrayList<AutocompletePrediction> mResultList;

    /**
     * Handles autocomplete requests.
     */
    private GoogleApiClient mGoogleApiClient;

    /**
     * The bounds used for Places Geo Data autocomplete API requests.
     */
    private LatLngBounds mBounds;

    /**
     * The autocomplete filter used to restrict queries to a specific set of place types.
     */
    private AutocompleteFilter mPlaceFilter;

    /**
     * Initializes with a resource for text rows and autocomplete query bounds.
     *
     * @see android.widget.ArrayAdapter#ArrayAdapter(android.content.Context, int)
     */
    public PlaceAutocompleteAdapter(Context context, GoogleApiClient googleApiClient,
                                    LatLngBounds bounds, AutocompleteFilter filter) {
        super(context, android.R.layout.simple_expandable_list_item_2, android.R.id.text1);
        mGoogleApiClient = googleApiClient;
        mBounds = bounds;
```

```
    mPlaceFilter = filter;
}

/**
 * Sets the bounds for all subsequent queries.
 */
public void setBounds(LatLngBounds bounds) {
    mBounds = bounds;
}

/**
 * Returns the number of results received in the last autocomplete query.
 */
@Override
public int getCount() {
    return mResultList.size();
}

/**
 * Returns an item from the last autocomplete query.
 */
@Override
public AutocompletePrediction getItem(int position) {
    return mResultList.get(position);
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View row = super.getView(position, convertView, parent);

    // Sets the primary and secondary text for a row.
    // Note that getPrimaryText() and getSecondaryText() return a CharSequence that may contain
    // styling based on the given CharacterStyle.

    AutocompletePrediction item = getItem(position);

    TextView textView1 = (TextView) row.findViewById(android.R.id.text1);
    TextView textView2 = (TextView) row.findViewById(android.R.id.text2);
    textView1.setText(item.getPrimaryText(STYLE_BOLD));
    textView2.setText(item.getSecondaryText(STYLE_BOLD));

    return row;
}

/**
 * Returns the filter for the current set of autocomplete results.
 */
@Override
public Filter getFilter() {
    return new Filter() {
        @Override
        protected FilterResults performFiltering(CharSequence constraint) {
```

```
            FilterResults results = new FilterResults();

            // We need a separate list to store the results, since
            // this is run asynchronously.
            ArrayList<AutocompletePrediction> filterData = new ArrayList<>();

            // Skip the autocomplete query if no constraints are given.
            if (constraint != null) {
                // Query the autocomplete API for the (constraint) search string.
                filterData = getAutocomplete(constraint);
            }

            results.values = filterData;
            if (filterData != null) {
                results.count = filterData.size();
            } else {
                results.count = 0;
            }

            return results;
        }

        @Override
        protected void publishResults(CharSequence constraint, FilterResults results) {

            if (results != null && results.count > 0) {
                // The API returned at least one result, update the data.
                mResultList = (ArrayList<AutocompletePrediction>) results.values;
                notifyDataSetChanged();
            } else {
                // The API did not return any results, invalidate the data set.
                notifyDataSetInvalidated();
            }
        }

        @Override
        public CharSequence convertResultToString(Object resultValue) {
            // Override this method to display a readable result in the AutocompleteTextView
            // when clicked.
            if (resultValue instanceof AutocompletePrediction) {
                return ((AutocompletePrediction) resultValue).getFullText(null);
            } else {
                return super.convertResultToString(resultValue);
            }
        }
    };
}

/**
 * Submits an autocomplete query to the Places Geo Data Autocomplete API.
 * Results are returned as frozen AutocompletePrediction objects, ready to be cached.
 * objects to store the Place ID and description that the API returns.
```

```
     * Returns an empty list if no results were found.
     * Returns null if the API client is not available or the query did not complete
     * successfully.
     * This method MUST be called off the main UI thread, as it will block until data is returned
     * from the API, which may include a network request.
     *
     * @param constraint Autocomplete query string
     * @return Results from the autocomplete API or null if the query was not successful.
     * @see Places#GEO_DATA_API#getAutocomplete(CharSequence)
     * @see AutocompletePrediction#freeze()
     */
    private ArrayList<AutocompletePrediction> getAutocomplete(CharSequence constraint) {
        if (mGoogleApiClient.isConnected()) {
            Log.i(TAG, "Starting autocomplete query for: " + constraint);

            // Submit the query to the autocomplete API and retrieve a PendingResult that will
            // contain the results when the query completes.
            PendingResult<AutocompletePredictionBuffer> results =
                    Places.GeoDataApi
                            .getAutocompletePredictions(mGoogleApiClient, constraint.toString(),
                                    mBounds, mPlaceFilter);

            // This method should have been called off the main UI thread. Block and wait for at most 60s
            // for a result from the API.
            AutocompletePredictionBuffer autocompletePredictions = results
                    .await(60, TimeUnit.SECONDS);

            // Confirm that the query completed successfully, otherwise return null
            final Status status = autocompletePredictions.getStatus();
            if (!status.isSuccess()) {
                Toast.makeText(getContext(), "Error contacting API: " + status.toString(),
                        Toast.LENGTH_SHORT).show();
                Log.e(TAG, "Error getting autocomplete prediction API call: " + status.toString());
                autocompletePredictions.release();
                return null;
            }

            Log.i(TAG, "Query completed. Received " + autocompletePredictions.getCount()
                    + " predictions.");

            // Freeze the results immutable representation that can be stored safely.
            return DataBufferUtils.freezeAndClose(autocompletePredictions);
        }
        Log.e(TAG, "Google API client is not connected for autocomplete query.");
        return null;
    }


}
```

ReadyPage.java

This readypage asks the user how long it takes them to get ready.

```
package com.example.arpit.alarmclock2;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;

public class ReadyPage extends AppCompatActivity implements
AdapterView.OnItemSelectedListener {

    private String hourTime, minTime, ampmTime
            ,hourReady, minReady
            ,fromDest, destAddress;



    public static Button readyButton;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.ready_page);

        Bundle bundle = getIntent().getExtras();
        hourTime = bundle.getString("hourTime");
        minTime = bundle.getString("minTime");
        ampmTime = bundle.getString("ampmTime");

        //get the spinner from the xml.
        readyButton = findViewById(R.id.readyButton);
        final Spinner readyHourSpinner = findViewById(R.id.readyHourSpinner);
        final Spinner readyMinSpinner = findViewById(R.id.readyMinSpinner);

        //create a list of items for the spinner.
        String[] hourSpinner = new String[]{"Hours", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11",
"12"};
        String[] minSpinner = new String[]{"Minutes", "00", "15", "30", "45"};
        //create an adapter to describe how the items are displayed, adapters are used in several places
in android.
        //There are multiple variations of this, but this is the basic variant.
        ArrayAdapter<String> hourAdapter = new ArrayAdapter<>(this,
```

```
android.R.layout.simple_spinner_dropdown_item, hourSpinner);
        ArrayAdapter<String> minAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, minSpinner);
        //set the spinners adapter to the previously created one.
        readyHourSpinner.setAdapter(hourAdapter);
        readyHourSpinner.setOnItemSelectedListener(this);

        readyMinSpinner.setAdapter(minAdapter);
        readyMinSpinner.setOnItemSelectedListener(this);

        //OnClickButtonListener();

        readyButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(ReadyPage.this, DestPage.class);
                Intent sentData = new Intent(ReadyPage.this, DestPage.class);
                intent.putExtra("hourTime", hourTime);
                intent.putExtra("minTime", minTime);
                intent.putExtra("ampmTime", ampmTime);
                intent.putExtra("hourReady", String.valueOf(readyHourSpinner.getSelectedItem()));
                intent.putExtra("minReady", String.valueOf(readyMinSpinner.getSelectedItem()));
                //intent.putExtra("ampmTime", String.valueOf(ampmTimeSpinner.getSelectedItem()));
                //startActivity(sentData);
                startActivity(intent);
            }
        });
    }

    public void onItemSelected(AdapterView<?> parent, View view, int position, long id){
        String item = parent.getItemAtPosition(position).toString();
        Toast.makeText(parent.getContext(), "Selected: " + item, Toast.LENGTH_LONG).show();
    }

    public void onNothingSelected(AdapterView<?> arg0) {

    }


    }
```

SetAlarm.java

This sets an alarm based on the user inputs and calculates an automatic alarm.

```
package com.example.arpit.alarmclock2;

import android.annotation.SuppressLint;
```

```
import android.content.Intent;
import android.provider.AlarmClock;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class SetAlarm extends AppCompatActivity {


    EditText mHourEditText;
    EditText mMinuteEditText;

    Button mSetAlarmButton;

    public String hourTimeString, minTimeString
            ,hourReadyString, minReadyString;

    public int hourTimeInt, minTimeInt
            ,hourReadyInt, minReadyInt;

    public int alarmHourInt, alarmMinInt;

    public EditText travelTimeText;

    public int travelTimeInt;



    @SuppressLint("WrongViewCast")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.set_alarm);

        Bundle bundle = getIntent().getExtras();
        hourTimeString = bundle.getString("hourTimeString");
        minTimeString = bundle.getString("minTimeString");
        hourReadyString = bundle.getString("hourReadyString");
        minReadyString = bundle.getString("minReadyString");

        hourTimeInt = Integer.parseInt(hourTimeString);
        minTimeInt = Integer.parseInt(minTimeString);
        hourReadyInt = Integer.parseInt(hourReadyString);
        minReadyInt = Integer.parseInt(minReadyString);
        //minReadyInt = 0;



        travelTimeText = (EditText) findViewById(R.id.travelTimeText);
```

```
        //mHourEditText = (EditText)findViewById(R.id.travelTimeText);
       // mMinuteEditText=(EditText)findViewById(R.id.minute_edit_text);

        mSetAlarmButton=(Button)findViewById(R.id.set_alarm_button);
        mSetAlarmButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
//              // int hour = Integer.parseInt(mHourEditText.getText().toString());
                //int minute = Integer.parseInt(mMinuteEditText.getText().toString());
                travelTimeInt = Integer.parseInt(String.valueOf(travelTimeText.getText()));
                int latency = (hourTimeInt - hourReadyInt);
                alarmHourInt = ((latency) - (travelTimeInt));
                alarmMinInt = (minTimeInt - minReadyInt);



                Intent intent = new Intent(AlarmClock.ACTION_SET_ALARM);
                intent.putExtra(AlarmClock.EXTRA_HOUR, alarmHourInt);
                intent.putExtra(AlarmClock.EXTRA_MINUTES, alarmMinInt);
                startActivity(intent);



                //if (hour < 24 && minute <= 60) {
                   // startActivity(intent);

               // }
            }
        });


    }


    }
```

TimePage.java

This time page shows the spinner method for the pull down menu of the hour and minute.

```
package com.example.arpit.alarmclock2;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
```

```
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;

public class TimePage extends AppCompatActivity implements AdapterView.OnItemSelectedListener
{


    private static Button timeNextButton;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.time_page);
        OnClickButtonListenerHour();
        OnClickButtonListenerMin();
        //OnClickButtonListenerAMPM();
        OnClickButtonListenerSwitch();

        //get the spinner from the xml.
        final Spinner hourTimeSpinner = findViewById(R.id.hourTimeSpinner);
        timeNextButton = findViewById(R.id.timeNextButton);
        hourTimeSpinner.setOnItemSelectedListener(this);//
        //create a list of items for the spinner.
        String[] hourTimeMethod = new String[]{"Hour", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"11", "12", "13", "14", "15","16","17","18","19","20","21","22","23","24"};
        //create an adapter to describe how the items are displayed, adapters are used in several places
in android.
        //There are multiple variations of this, but this is the basic variant.
        final ArrayAdapter<String> hourTimeAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, hourTimeMethod);
        //set the spinners adapter to the previously created one.
        hourTimeSpinner.setAdapter(hourTimeAdapter);

        //get the spinner from the xml.
        final Spinner minTimeSpinner = findViewById(R.id.minTimeSpinner);
        //create a list of items for the spinner.
        String[] minTimeMethod = new String[]{"Minute", "00", "05", "10", "15", "20", "30", "45", "50",
"55"};
        //create an adapter to describe how the items are displayed, adapters are used in several places
in android.
        //There are multiple variations of this, but this is the basic variant.
        ArrayAdapter<String> minTimeAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, minTimeMethod);
        //set the spinners adapter to the previously created one.
        minTimeSpinner.setAdapter(minTimeAdapter);
        minTimeSpinner.setOnItemSelectedListener(this);



        timeNextButton.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(TimePage.this, ReadyPage.class);
            Intent sentData = new Intent(TimePage.this, ReadyPage.class);
            intent.putExtra("hourTime", hourTimeSpinner.getSelectedItem().toString());
            intent.putExtra("minTime", minTimeSpinner.getSelectedItem().toString());
          //  intent.putExtra("ampmTime", ampmTimeSpinner.getSelectedItem().toString());
            //startActivity(sentData);
            startActivity(intent);
        }
    });


}

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
    String item = parent.getItemAtPosition(position).toString();
    Toast.makeText(parent.getContext(), "Selected: " + item, Toast.LENGTH_LONG).show();
}

public void onNothingSelected(AdapterView<?> arg0) {

}

public void OnClickButtonListenerHour() {


}

public void OnClickButtonListenerMin() {

}




public void OnClickButtonListenerSwitch() {
    timeNextButton = findViewById(R.id.timeNextButton);
    timeNextButton.setOnClickListener(
            new View.OnClickListener() {
                @Override
                public void onClick(View v) {


                    Intent intent = new Intent(TimePage.this, ReadyPage.class);
                    startActivity(intent);
                }
            }
    );

}
}
```

Maps

All the classes in Maps was used to experiment and learn more on how to integrate distance and duration.

GetDirectionsData.java

```
package com.example.arpit.alarmclock2;

import android.os.AsyncTask;

import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

import java.io.IOException;
import java.util.HashMap;


public class GetDirectionsData extends AsyncTask<Object,String,String> {

    GoogleMap mMap;
    String url;
    String googleDirectionsData;
    String duration, distance, durationGet;
    LatLng latLng;


    //TextView durationText;
    @Override
    protected String doInBackground(Object... objects) {

        mMap = (GoogleMap)objects[0];
        url = (String)objects[1];
        latLng = (LatLng)objects[2];



        DownloadUrl downloadUrl = new DownloadUrl();
        try {
            googleDirectionsData = downloadUrl.readUrl(url);
        } catch (IOException e) {
            e.printStackTrace();
        }
```

```
        return googleDirectionsData;
    }

    @Override
    protected void onPostExecute(String s) {


        GetDirectionsData d = new GetDirectionsData();
        HashMap<String, String> directionsList;
        DataParser parser = new DataParser();
        directionsList = parser.parseDirections(s);
        duration = directionsList.get("duration");
        distance = directionsList.get("distance");
        d.setDurationMethod(duration);



        mMap.clear();
        //durationText.setText(duration);
        MarkerOptions markerOptions = new MarkerOptions();
        markerOptions.position(latLng);
        markerOptions.draggable(true);
        markerOptions.title("Duration ="+duration);
        markerOptions.snippet("Distance = "+distance);

        mMap.addMarker(markerOptions);

    }

    public void setDurationMethod(String dur){
         durationGet = dur;
    }

    public String getDurationMethod(){
        return duration;

    }

}
```

GetNearbyPlacesData.java

```
package com.example.arpit.alarmclock2;

import android.os.AsyncTask;
```

```
import android.util.Log;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

import java.io.IOException;
import java.util.HashMap;
import java.util.List;

public class GetNearbyPlacesData extends AsyncTask<Object, String, String> {

    String googlePlacesData;
    GoogleMap mMap;
    String url;

    @Override
    protected String doInBackground(Object... objects) {
        mMap = (GoogleMap)objects[0];
        url = (String)objects[1];

        DownloadUrl downloadUrl = new DownloadUrl();
        try {
            googlePlacesData = downloadUrl.readUrl(url);
        } catch (IOException e) {
            e.printStackTrace();
        }

        return googlePlacesData;
    }

    @Override
    protected void onPostExecute(String s) {
        List<HashMap<String, String>> nearbyPlaceList = null;
        DataParser parser = new DataParser();
        nearbyPlaceList = parser.parse(s);
        showNearbyPlaces(nearbyPlaceList);
    }

    private void showNearbyPlaces(List<HashMap<String,String>> nearbyPlaceList)
    {
        for(int i = 0;i<nearbyPlaceList.size() ; i++)
        {
            MarkerOptions markerOptions = new MarkerOptions();
            HashMap<String , String> googlePlace = nearbyPlaceList.get(i);
            Log.d("onPostExecute","Entered into showing locations");

            String placeName = googlePlace.get("place_name");
            String vicinity = googlePlace.get("vicinity");
            double lat = Double.parseDouble( googlePlace.get("lat") );
```

```
            double lng = Double.parseDouble( googlePlace.get("lng"));

            LatLng latLng = new LatLng(lat, lng);
            markerOptions.position(latLng);
            markerOptions.title(placeName +" : "+ vicinity);
            markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_B
LUE));
            mMap.addMarker(markerOptions);
            mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
            mMap.animateCamera(CameraUpdateFactory.zoomTo(10));



        }


    }

}
```

DataParser.java

```
package com.example.arpit.alarmclock2;

import android.util.Log;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;



class DataParser {



    private HashMap<String, String> getDuration(JSONArray googleDirectionsJson){

        HashMap<String, String> googleDirectionMap = new HashMap<>();
        String duration = "";
        String distance = "";

        Log.d("json response", googleDirectionsJson.toString());
        try {
            duration =
```

```
googleDirectionsJson.getJSONObject(0).getJSONObject("duration").getString("text");
            distance =
googleDirectionsJson.getJSONObject(0).getJSONObject("distance").getString("text");
            System.out.println("OOOOOoooooOOOOOOooooooooOOOooOOOO"+duration);

             googleDirectionMap.put("duration", duration);
             googleDirectionMap.put("distance", distance);


        } catch (JSONException e) {
            e.printStackTrace();
        }

        return googleDirectionMap;
    }


    private HashMap<String, String> getPlace(JSONObject googlePlaceJson) {
        HashMap<String, String> googlePlacesMap = new HashMap<>();
        String placeName = "-NA-";
        String vicinity = "-NA-";
        String latitude = "";
        String longitude = "";
        String reference = "";
        Log.d("getPlace", "Entered");


        try {
            if (!googlePlaceJson.isNull("name")) {

                placeName = googlePlaceJson.getString("name");

            }
            if (!googlePlaceJson.isNull("vicinity")) {
                vicinity = googlePlaceJson.getString("vicinity");

            }
            latitude =
googlePlaceJson.getJSONObject("geometry").getJSONObject("location").getString("lat");
            longitude =
googlePlaceJson.getJSONObject("geometry").getJSONObject("location").getString("lng");

            reference = googlePlaceJson.getString("reference");

            googlePlacesMap.put("place_name", placeName);
            googlePlacesMap.put("vicinity", vicinity);
            googlePlacesMap.put("lat", latitude);
            googlePlacesMap.put("lng", longitude);
            googlePlacesMap.put("reference", reference);


            Log.d("getPlace", "Putting Places");
```

```
    } catch (JSONException e) {
        e.printStackTrace();
    }

    return googlePlacesMap;
}


private List<HashMap<String, String>> getPlaces(JSONArray jsonArray) {
    int count = jsonArray.length();
    List<HashMap<String, String>> placesList = new ArrayList<>();
    HashMap<String, String> placeMap = null;
    Log.d("Places", "getPlaces");

    for (int i = 0; i < count; i++) {
        try {
            placeMap = getPlace((JSONObject) jsonArray.get(i));
            placesList.add(placeMap);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    return placesList;

}

public List<HashMap<String, String>> parse(String jsonData) {
    JSONArray jsonArray = null;
    JSONObject jsonObject;

    try {
        Log.d("Places", "parse");

        jsonObject = new JSONObject(jsonData);
        jsonArray = jsonObject.getJSONArray("results");

    } catch (JSONException e) {
        e.printStackTrace();
    }

    return getPlaces(jsonArray);
}


public String[] getPaths(JSONArray googleStepsJson) {
    int count = googleStepsJson.length();
    String[] polylines = new String[count];

    for (int i = 0; i < count; i++) {
        try {
```

```
                polylines[i] = getPath(googleStepsJson.getJSONObject(i));
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }

        return polylines;
    }

    public String getPath(JSONObject googlePathJson) {
        String polyline = "";
        try {
            polyline = googlePathJson.getJSONObject("polyline").getString("points");
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return polyline;
    }

    public HashMap<String,String> parseDirections(String jsonData){
        JSONArray jsonArray = null;
        JSONObject jsonObject;

        try {
            jsonObject = new JSONObject(jsonData);
            jsonArray = jsonObject.getJSONArray("routes");

        } catch (JSONException e) {
            e.printStackTrace();
        }

        return getDuration(jsonArray);

    }


}
```

DownloadUrl.java

```
package com.example.arpit.alarmclock2;

import android.util.Log;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
```

```
import java.net.MalformedURLException;
import java.net.URL;


public class DownloadUrl {

    public String readUrl(String myUrl) throws IOException
    {
        String data = "";
        InputStream inputStream = null;
        HttpURLConnection urlConnection = null;
        try {
            URL url = new URL(myUrl);
            urlConnection = (HttpURLConnection) url.openConnection();
            urlConnection.connect();

            inputStream = urlConnection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(inputStream));
            StringBuffer sb = new StringBuffer();

            String line = "";
            while((line = br.readLine()) != null)
            {
                sb.append(line);

            }

            data = sb.toString();
            Log.d("downloadUrl", data.toString());

            br.close();

        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        finally {
            if(inputStream != null)
                inputStream.close();
            urlConnection.disconnect();
        }

        Log.d("data downlaod",data);
        return data;

    }
}
```

Geocoding

This class enable us to convert the string addresses into their corresponding coordinates.

GeocodingLocation.java

```
package com.example.arpit.alarmclock2;
import android.content.Context;
import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.util.Log;

import java.io.IOException;
import java.util.List;
import java.util.Locale;

public class GeocodingLocation {

    private static final String TAG = "GeocodingLocation";

    public static void getFromAddressFromLocation(final String locationAddress,
                                  final Context context, final Handler handler) {
        Thread thread = new Thread() {
            @Override
            public void run() {
                Geocoder geocoder = new Geocoder(context, Locale.getDefault());
                String result = null;
                try {
                    List
                            addressList = geocoder.getFromLocationName(locationAddress, 1);
                    if (addressList != null && addressList.size() > 0) {
                        Address address = (Address) addressList.get(0);
                        StringBuilder sb = new StringBuilder();
                        sb.append(address.getLatitude()).append("\n");
                        sb.append(address.getLongitude()).append("\n");
                        result = sb.toString();
                    }
                } catch (IOException e) {
                    Log.e(TAG, "Unable to connect to Geocoder", e);
                } finally {
                    Message message = Message.obtain();
                    message.setTarget(handler);
                    if (result != null) {
                        message.what = 1;
                        Bundle bundle = new Bundle();
                        result = "Address: " + locationAddress +
                                "\n\nLatitude and Longitude :\n" + result;
                        bundle.putString("address", result);
```

```
                message.setData(bundle);
            } else {
                message.what = 1;
                Bundle bundle = new Bundle();
                result = "Address: " + locationAddress +
                        "\n Unable to get Latitude and Longitude for this address location.";
                bundle.putString("address", result);
                message.setData(bundle);
            }
            message.sendToTarget();
        }
    }
};
thread.start();
}

public static void getToAddressFromLocation(final String locationAddress,
                                final Context context, final Handler handler) {
    Thread thread = new Thread() {
        @Override
        public void run() {
            Geocoder geocoder = new Geocoder(context, Locale.getDefault());
            String result = null;
            try {
                List
                        addressList = geocoder.getFromLocationName(locationAddress, 1);
                if (addressList != null && addressList.size() > 0) {
                    Address address = (Address) addressList.get(0);
                    StringBuilder sb = new StringBuilder();
                    sb.append(address.getLatitude()).append("\n");
                    sb.append(address.getLongitude()).append("\n");
                    result = sb.toString();
                }
            } catch (IOException e) {
                Log.e(TAG, "Unable to connect to Geocoder", e);
            } finally {
                Message message = Message.obtain();
                message.setTarget(handler);
                if (result != null) {
                    message.what = 1;
                    Bundle bundle = new Bundle();
                    result = "Address: " + locationAddress +
                            "\n\nLatitude and Longitude :\n" + result;
                    bundle.putString("address", result);
                    message.setData(bundle);
                } else {
                    message.what = 1;
                    Bundle bundle = new Bundle();
                    result = "Address: " + locationAddress +
                            "\n Unable to get Latitude and Longitude for this address location.";
                    bundle.putString("address", result);
                    message.setData(bundle);
```

```
                }
                message.sendToTarget();
            }
        }
    };
    thread.start();
  }
}
```

Layouts

All the XML files were used to design the application according to its corresponding activity page.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/frameLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:theme="@style/AppTheme">
    tools:context=".MainActivity"
    tools:layout_editor_absoluteY="25dp">

    <ImageView
        android:id="@+id/backgroundPic"
        android:layout_width="wrap_content"
        android:layout_height="821dp"
        android:contentDescription="@string/todo"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.145"
        app:srcCompat="@mipmap/blurry_background" />

    <TextView
        android:id="@+id/welcomeTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="40dp"
        android:fontFamily="serif"
        android:text="@string/app_name"
        android:textSize="50sp"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/backgroundPic" />

<ImageButton
    android:id="@+id/newAlarmButton"
    android:layout_width="336dp"
    android:layout_height="0dp"
    android:layout_marginTop="285dp"
    android:layout_marginBottom="95dp"
    app:layout_constraintBottom_toTopOf="@+id/seeAlarmsButton"
    app:layout_constraintStart_toStartOf="@+id/backgroundPic"
    app:layout_constraintTop_toTopOf="@id/backgroundPic"
    app:srcCompat="@mipmap/button_background"
    tools:srcCompat="@mipmap/button_background" />

<ImageButton
    android:id="@+id/seeAlarmsButton"
    android:layout_width="336dp"
    android:layout_height="0dp"
    android:layout_marginBottom="151dp"
    android:contentDescription="TODO"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="@+id/backgroundPic"
    app:layout_constraintTop_toBottomOf="@+id/newAlarmButton"
    app:srcCompat="@mipmap/button_background"
    tools:srcCompat="@mipmap/button_background" />

<ImageButton
    android:id="@+id/mapsButton"
    android:layout_width="51dp"
    android:layout_height="0dp"
    android:layout_marginBottom="4dp"
    android:contentDescription="TODO"
    app:layout_constraintBottom_toBottomOf="parent"
    tools:layout_editor_absoluteX="7dp"
    tools:srcCompat="@mipmap/ic_gps" />

<TextView
    android:id="@+id/createAlarmText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Create a new alarm"
    android:textSize="30sp"
    app:layout_constraintBottom_toBottomOf="@+id/newAlarmButton"
    app:layout_constraintEnd_toEndOf="@+id/newAlarmButton"
    app:layout_constraintStart_toStartOf="@+id/newAlarmButton"
    app:layout_constraintTop_toTopOf="@+id/newAlarmButton" />

<TextView
    android:id="@+id/seeAlarmText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:text="See your set alarms"
        android:textSize="30sp"
        app:layout_constraintBottom_toBottomOf="@+id/seeAlarmsButton"
        app:layout_constraintEnd_toEndOf="@+id/seeAlarmsButton"
        app:layout_constraintStart_toStartOf="@+id/seeAlarmsButton"
        app:layout_constraintTop_toTopOf="@+id/seeAlarmsButton" />

</android.support.constraint.ConstraintLayout>
```

activity_maps.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:weightSum="1">

        <EditText
            android:id="@+id/TF_location"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:layout_toStartOf="@+id/B_search"
            android:layout_weight="1.01"
            android:ems="10"
            android:inputType="textPersonName" />

        <Button
            android:id="@+id/B_search"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBottom="@+id/TF_location"
            android:layout_alignParentEnd="true"
            android:onClick="onClick"
            android:text="Search" />


    </LinearLayout>

    <fragment xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:map="http://schemas.android.com/apk/res-auto"
```

```
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="300dp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true"
    tools:context="com.example.priyanka.mapsnearbyplaces.MapsActivity"
    android:layout_below="@+id/B_hospital" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

</LinearLayout>

<Button
    android:id="@+id/B_hospital"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/linearLayout"
    android:onClick="onClick"
    android:text="Hospitals" />

<Button
    android:id="@+id/B_restaurant"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/linearLayout"
    android:layout_toEndOf="@+id/B_hospital"
    android:onClick="onClick"
    android:text="Restaurants" />

<Button
    android:id="@+id/B_school"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/linearLayout"
    android:layout_toEndOf="@+id/B_restaurant"
    android:onClick="onClick"
    android:text="Schools" />

<Button
    android:id="@+id/B_to"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/map"
    android:layout_marginStart="16dp"
    android:layout_toEndOf="@+id/B_school"
    android:onClick="onClick"
    android:text="To" />
```

```
    <TextView
        android:id="@+id/durationText"
        android:layout_width="127dp"
        android:layout_height="45dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginStart="88dp"
        android:layout_marginTop="203dp"
        android:allowUndo="false"
        android:autoText="false"
        android:text="SHOW DURATION" />

</RelativeLayout>
```

dest_page.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/backgroundDest"
        android:layout_width="500dp"
        android:layout_height="900dp"
        android:layout_marginStart="8dp"
        android:adjustViewBounds="true"
        android:cropToPadding="true"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.322"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@mipmap/blurry_background" />

    <TextView
        android:id="@+id/destText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:text="Destination"
        android:textSize="30dp"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/backgroundDest" />

<TextView
    android:id="@+id/methodText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="212dp"
    android:text="Method of transportation"
    android:textSize="30dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Spinner
    android:id="@+id/methodSpinner"
    android:layout_width="285dp"
    android:layout_height="49dp"
    android:layout_marginTop="32dp"
    android:background="@android:drawable/btn_dropdown"
    android:spinnerMode="dropdown"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.505"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/methodText" />

<Button
    android:id="@+id/destNextButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="120dp"
    android:text="Next"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<ProgressBar
    android:id="@+id/progressBar"
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="130dp"
    android:layout_marginBottom="55dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<AutoCompleteTextView
    android:id="@+id/fromDest"
    android:layout_width="0dp"
```

```
        android:layout_height="42dp"
        android:layout_marginStart="17dp"
        android:layout_marginTop="24dp"
        android:layout_marginEnd="17dp"
        android:hint="Current Location"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/destText" />

    <AutoCompleteTextView
        android:id="@+id/destAddress"
        android:layout_width="0dp"
        android:layout_height="43dp"
        android:layout_marginStart="17dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="17dp"
        android:layout_marginBottom="8dp"
        android:hint="Final Destination"
        app:layout_constraintBottom_toTopOf="@+id/methodText"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/fromDest"
        app:layout_constraintVertical_bias="0.423" />

    <ImageView
        android:id="@+id/ic_gps"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="6dp"
        android:layout_marginBottom="1dp"
        app:layout_constraintBottom_toTopOf="@+id/fromDest"
        app:layout_constraintStart_toStartOf="parent"
        app:srcCompat="@mipmap/ic_gps" />


</android.support.constraint.ConstraintLayout>
```

method_page.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
```

```
    android:id="@+id/backgroundDest"
    android:layout_width="500dp"
    android:layout_height="900dp"
    android:layout_marginStart="8dp"
    android:adjustViewBounds="true"
    android:cropToPadding="true"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.532"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@mipmap/blurry_background" />

<TextView
    android:id="@+id/methodText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="36dp"
    android:layout_marginTop="136dp"
    android:text="Method of transportation"
    android:textSize="30dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Spinner
    android:id="@+id/methodSpinner"
    android:layout_width="285dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="56dp"
    android:background="@android:drawable/btn_dropdown"
    android:spinnerMode="dropdown"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/methodText" />

<Button
    android:id="@+id/methodNextButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="19dp"
    android:layout_marginBottom="8dp"
    android:text="Next"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/methodSpinner" />

<TextView
    android:id="@+id/testText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="148dp"
    android:layout_marginTop="52dp"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

result_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ResultView"
    tools:layout_editor_absoluteY="81dp">

    <ImageView
        android:id="@+id/backgroundPic"
        android:layout_width="wrap_content"
        android:layout_height="821dp"
        android:contentDescription="@string/todo"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.145"
        app:srcCompat="@mipmap/blurry_background" />

    <TextView
        android:id="@+id/travelTimeText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="44dp"
        android:text="Time Activity"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/hourTimeText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="7dp"
        android:layout_marginTop="9dp"
```

```
    android:text="hour"
    app:layout_constraintStart_toStartOf="@+id/travelTimeText"
    app:layout_constraintTop_toBottomOf="@+id/travelTimeText" />

<TextView
    android:id="@+id/minTimeText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp"
    android:text="min "
    app:layout_constraintStart_toStartOf="@+id/hourTimeText"
    app:layout_constraintTop_toBottomOf="@+id/hourTimeText" />

<TextView
    android:id="@+id/readyActivityText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="40dp"
    android:text="Ready Activity"
    android:textSize="18sp"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/minTimeText" />

<TextView
    android:id="@+id/hourReadyText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:layout_marginTop="20dp"
    android:text="hour"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/readyActivityText" />

<TextView
    android:id="@+id/minReadyText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:layout_marginTop="16dp"
    android:text="min "
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/hourReadyText" />

<TextView
    android:id="@+id/destTextResult"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="19dp"
    android:text="Destination Cords"
    android:textSize="24sp"
```

```
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/minReadyText" />

    <TextView
        android:id="@+id/fromDestText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="35dp"
        android:layout_marginTop="18dp"
        android:text="From"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/destTextResult" />

    <TextView
        android:id="@+id/destTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="35dp"
        android:layout_marginBottom="17dp"
        android:text="Dest"
        app:layout_constraintBottom_toTopOf="@+id/addressButton"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/addressButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="15dp"
        android:layout_marginBottom="12dp"
        android:text="Get Cords"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/nextButton"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/nextButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="13dp"
        android:layout_marginBottom="12dp"
        android:text="NEXT"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/addressButton" />

</android.support.constraint.ConstraintLayout>
```

ready_page.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/readyBackground"
        android:layout_width="546dp"
        android:layout_height="888dp"
        android:contentDescription="TODO"
        app:layout_constraintBaseline_toBaselineOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:srcCompat="@mipmap/blurry_background" />

    <TextView
        android:id="@+id/welcomeTextView"
        android:layout_width="370dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginTop="28dp"
        android:fontFamily="serif"
        android:text="@string/ready_text"
        android:textColor="@color/colorAccent"
        android:textSize="40sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/readyBackground" />

    <Spinner
        android:id="@+id/readyMinSpinner"
        android:layout_width="114dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="264dp"
        android:background="@android:drawable/btn_dropdown"
        android:spinnerMode="dropdown"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/readyHourSpinner"
        app:layout_constraintTop_toTopOf="@id/readyBackground" />

    <Spinner
        android:id="@+id/readyHourSpinner"
        android:layout_width="114dp"
        android:layout_height="wrap_content"
```

```
        android:layout_marginStart="16dp"
        android:layout_marginTop="264dp"
        android:background="@android:drawable/btn_dropdown"
        android:spinnerMode="dropdown"
        app:layout_constraintEnd_toStartOf="@+id/readyMinSpinner"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@id/readyBackground" />

    <Button
        android:id="@+id/readyButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="216dp"
        android:text="Next"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />


</android.support.constraint.ConstraintLayout>
```

set_alarm.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/backgroundPic"
        android:layout_width="398dp"
        android:layout_height="853dp"
        android:layout_alignParentTop="true"
        android:layout_marginStart="3dp"
        android:layout_marginEnd="3dp"
        android:contentDescription="@string/todo"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@mipmap/blurry_background" />
```

```
    <Button
        android:id="@+id/set_alarm_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/travelTimeText"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginTop="80dp"
        android:text="set alarm"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/travelTimeText" />

    <EditText
        android:id="@+id/travelTimeText"
        android:layout_width="230dp"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="152dp"
        android:ems="10"
        android:hint="Travel Time"
        android:inputType="time"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

time_page.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="500dp"
        android:layout_height="900dp"
        android:layout_marginStart="8dp"
        android:adjustViewBounds="true"
        android:cropToPadding="true"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.733"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@mipmap/blurry_background" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="12dp"
        android:layout_marginTop="139dp"
        android:text="What time do you need to be there?"
        android:textSize="28dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Spinner
        android:id="@+id/hourTimeSpinner"
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="32dp"
        android:background="@android:drawable/btn_dropdown"
        android:spinnerMode="dropdown"
        app:layout_constraintEnd_toStartOf="@+id/minTimeSpinner"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <Spinner
        android:id="@+id/minTimeSpinner"
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:background="@android:drawable/btn_dropdown"
        android:spinnerMode="dropdown"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/hourTimeSpinner"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <Button
        android:id="@+id/timeNextButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="108dp"
        android:text="Next"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.543"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/hourTimeSpinner" />

</android.support.constraint.ConstraintLayout>
```