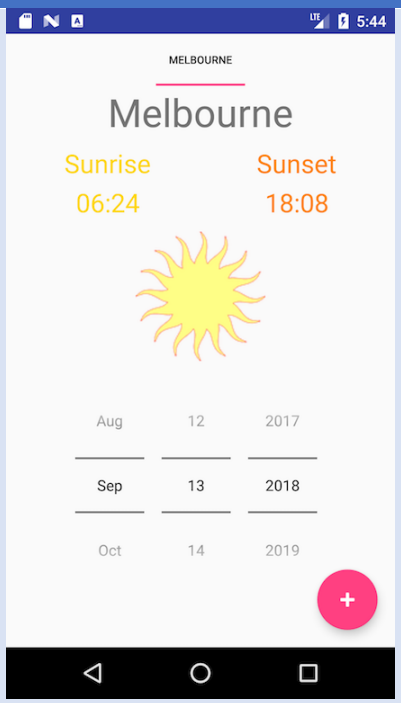
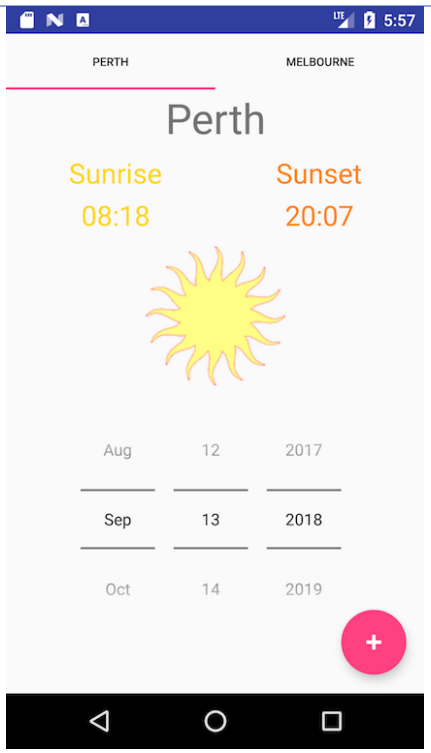
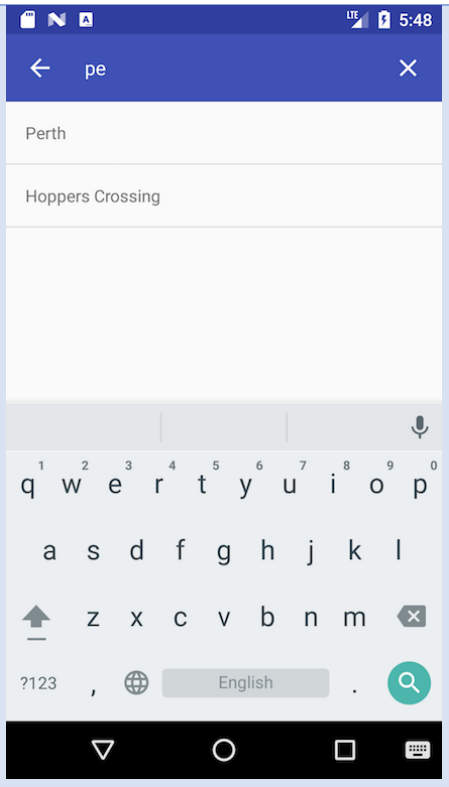
Software Development for Mobile Devices

Submission for Assignment A8.1P

App Screenshot

Main screen

Screen	
Home Screen (Default)	

Home Screen (Add)	
Add Location Screen	

Main Layout (activity_main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"

    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <android.support.v4.view.ViewPager
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/pager"
        android:layout_gravity="top|center"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <android.support.design.widget.TabLayout
            android:id="@+id/tabLayout"
            style="@style/MyCustomTabLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="top" />
    </android.support.v4.view.ViewPager>

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/btnAddLocation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="16dp"
        app:srcCompat="@android:drawable/ic_input_add"
        android:tint="@android:color/white"
        app:layout_anchorGravity="bottom|right|end"
        android:onClick="floatButtonClick"/>

</android.support.design.widget.CoordinatorLayout>
```

Sunset Fragment(sunset_fragment.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:id="@+id/locationTV"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Melbourne, AU"
        android:textSize="36sp"
        android:gravity="center"/>


    <TableLayout
        android:id="@+id/TableLayout01"
        android:layout_width="fill_parent"
```

```
        android:layout_height="wrap_content"
        android:stretchColumns="0,1"
        android:gravity="center"
        android:padding="5sp">
        <TableRow
            android:id="@+id/TableRow01"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:padding="2sp"
                android:text="Sunrise"
                android:textColor="#ffd700"
                android:textSize="24sp"/>
            <TextView
                android:padding="2sp"
                android:gravity="center"
                android:textColor="#ff8400"
                android:textSize="24sp"
                android:text="Sunset"
                android:layout_height="wrap_content"
                android:layout_width="wrap_content"/>
        </TableRow>

        <TableRow android:id="@+id/TableRow02" android:layout_width="match_parent"
android:layout_height="wrap_content">
            <TextView android:padding="2sp" android:textSize="24sp"
android:text="00:00" android:textColor="#ffd700" android:gravity="center"
android:layout_height="wrap_content" android:id="@+id/sunriseTimeTV"
android:layout_width="wrap_content"></TextView>
            <TextView android:padding="2sp" android:textSize="24sp"
android:text="00:00" android:textColor="#ff8400" android:gravity="center"
android:layout_height="wrap_content" android:id="@+id/sunsetTimeTV"
android:layout_width="wrap_content"></TextView>
        </TableRow>



    </TableLayout>



    <ImageView android:id="@+id/ImageView01" android:layout_width="match_parent"
        android:src="@drawable/sunpic"
        android:padding="4dp" android:layout_height="wrap_content"/>

    <DatePicker
        android:id="@+id/datePicker"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="5sp"
        android:calendarViewShown="false"
        android:datePickerMode="spinner"/>

</LinearLayout>
```

Add Location Layout (activity_temperature_conversion.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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".AddLocationActivity">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/colorPrimary">

    </android.support.v7.widget.Toolbar>

    <com.miguelcatalan.materialsearchview.MaterialSearchView
        android:id="@+id/search_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:searchBackground="@color/colorPrimary"
        app:searchSuggestionBackground="@color/colorPrimary"
        app:searchCloseIcon="@drawable/ic_action_navigation_close_inverted"
        app:searchBackIcon="@drawable/ic_action_navigation_arrow_back_inverted"
        app:searchSuggestionIcon="@drawable/ic_suggestion"
        android:textColor="#FFFFFF"
        android:textColorHint="#FFF">


    </com.miguelcatalan.materialsearchview.MaterialSearchView>

    <android.support.v7.widget.RecyclerView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/toolbar">

    </android.support.v7.widget.RecyclerView>


</RelativeLayout>
```

MainActivity

```
public class MainActivity extends AppCompatActivity {

    FragmentAdapter adapterViewPager;
    private List<GeoLocation> australiaLocations = new ArrayList<>();
    FloatingActionButton btnAdd;
    ViewPager vpPager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initData();
        btnAdd = findViewById(R.id.btnAddLocation);

        vpPager = (ViewPager) findViewById(R.id.pager);
```

```
        adapterViewPager = new FragmentAdapter(getSupportFragmentManager(),
australiaLocations);
        vpPager.setAdapter(adapterViewPager);
        vpPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {

            // This method will be invoked when a new page becomes selected.
            @Override
            public void onPageSelected(int position) {

            }

            // This method will be invoked when the current page is scrolled
            @Override
            public void onPageScrolled(int position, float positionOffset, int
positionOffsetPixels) {
                // Code goes here
            }

            // Called when the scroll state changes:
            // SCROLL_STATE_IDLE, SCROLL_STATE_DRAGGING, SCROLL_STATE_SETTLING
            @Override
            public void onPageScrollStateChanged(int state) {
                // Code goes here
            }
        });
        TabLayout tabLayout = findViewById(R.id.tabLayout);
        if (australiaLocations.size() > 4) {
            tabLayout.setTabMode(TabLayout.MODE_SCROLLABLE);
        } else {
            tabLayout.setTabMode(TabLayout.MODE_FIXED);
        }
        tabLayout.setupWithViewPager(vpPager);
    }

    public void addLocation(GeoLocation location) {
        this.australiaLocations.add(location);
    }

    private void initData() {
        TimeZone tz = TimeZone.getDefault();
        addLocation(new GeoLocation("Melbourne", -37.813629, 144.963058,tz));
    }

    public void floatButtonClick(View view) {
        Intent intent = new Intent(this, AddLocationActivity.class);
        startActivityForResult(intent, 1);
    }

    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == 1) {
            if(resultCode == RESULT_OK) {
               GeoLocation location = data.getExtras().getParcelable("geoLocation");
               if (this.australiaLocations.size() > 4) {
                   this.australiaLocations.remove(this.australiaLocations.size()-1);
               }

                this.australiaLocations.add(0,location);
                adapterViewPager = new FragmentAdapter(getSupportFragmentManager(),
australiaLocations);
                vpPager.setAdapter(adapterViewPager);
```

```
            }
        }
    }
}
```

AddLocationActivity

```
public class AddLocationActivity extends AppCompatActivity {

    MaterialSearchView searchView;
    List<GeoLocation> geoLocations = new ArrayList<>();
    RecyclerView recyclerView;
    GeolocationAdapter mAdapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_location);
        initializeUI();
        try {
            mapDataToObject(readDataFromFile());
        } catch (IOException ex) {
            Log.e("error", ex.toString());
        }

    }

    private void initializeUI() {
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle("Search Locations");
        toolbar.setTitleTextColor(Color.parseColor("#ffffff"));
        searchView = findViewById(R.id.search_view);
        searchView.setOnQueryTextListener(new MaterialSearchView.OnQueryTextListener()
{
            @Override
            public boolean onQueryTextSubmit(String query) {
                return false;
            }

            @Override
            public boolean onQueryTextChange(String newText) {
                if (newText != null && !newText.isEmpty()) {
                   ArrayList<GeoLocation> searchLocation =
findLocationByName(newText);
                    mAdapter = new GeolocationAdapter(searchLocation,
AddLocationActivity.this);
                    mAdapter.notifyDataSetChanged();
                    recyclerView.setAdapter(mAdapter);
                } else {

                    mAdapter = new GeolocationAdapter(geoLocations,
AddLocationActivity.this);
                    mAdapter.notifyDataSetChanged();
                    recyclerView.setAdapter(mAdapter);
                }
                return true;
            }
        });

        recyclerView = (RecyclerView) findViewById(R.id.listView);
```

```
        mAdapter = new GeolocationAdapter(geoLocations, this);
        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(getApplicationContext());
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setItemAnimator(new DefaultItemAnimator());
        recyclerView.addItemDecoration(new DividerItemDecoration(this,
LinearLayoutManager.VERTICAL));
        recyclerView.setAdapter(mAdapter);

    }

    private ArrayList<GeoLocation> findLocationByName(String name) {
        ArrayList<GeoLocation> tLocation = new ArrayList<>();
        for (GeoLocation location: geoLocations) {
            if (location.getLocationName().toLowerCase().contains(name.toLowerCase()))
{
                tLocation.add(location);
            }
        }
        return tLocation;
    }
    private ArrayList<String> readDataFromFile() throws IOException {
        ArrayList<String> records = new ArrayList<String>();
        BufferedReader reader = new BufferedReader(
                new InputStreamReader(getAssets().open("au_locations.txt")));
        String line;
        while ((line = reader.readLine()) != null)
        {
            // Skip the comment notation
            if (!line.contains("//")) {
                records.add(line);
            }
        }
        reader.close();
        return records;
    }

    private void mapDataToObject(ArrayList<String> string) {
        for (String line: string) {
            String[] row = line.split(",");
            GeoLocation geoLocation = new GeoLocation(row[0], Double.valueOf(row[1]),
Double.valueOf(row[2]), TimeZone.getDefault());
            this.geoLocations.add(geoLocation);
        }
        mAdapter.notifyDataSetChanged();
    }



    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_item, menu);
        MenuItem item = menu.findItem(R.id.action_search);
        searchView.setMenuItem(item);
        return true;
    }
}
```

Sunset Fragment

```
public class SunsetFragment extends android.support.v4.app.Fragment {

    private GeoLocation geoLocation;

    private View rootView;

    public GeoLocation getGeoLocation() {
        return geoLocation;
    }

    public void setGeoLocation(GeoLocation geoLocation) {
        this.geoLocation = geoLocation;
    }

    public static SunsetFragment newInstance(GeoLocation geoLocation) {
        SunsetFragment fragmentFirst = new SunsetFragment();
        fragmentFirst.setGeoLocation(geoLocation);
        return fragmentFirst;
    }

    // Store instance variables based on arguments passed
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    // Inflate the view for the fragment based on layout XML
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        rootView = inflater.inflate(R.layout.sunset_fragment, container, false);
        initializeUI(rootView);
        return rootView;
    }

    private void initializeUI(View view) {
        TextView locationTV = view.findViewById(R.id.locationTV);
        locationTV.setText(geoLocation.getLocationName());
        DatePicker dp = view.findViewById(R.id.datePicker);
        Calendar cal = Calendar.getInstance();
        int year = cal.get(Calendar.YEAR);
        int month = cal.get(Calendar.MONTH);
        int day = cal.get(Calendar.DAY_OF_MONTH);
        dp.init(year,month,day,dateChangeHandler); // setup initial values and reg.
handler
        updateTime(view,year, month, day);
    }

    private void updateTime(View view,int year, int monthOfYear, int dayOfMonth) {
//        TimeZone tz = TimeZone.getDefault();
//        GeoLocation geolocation = new GeoLocation("Melbourne", -37.50, 145.01, tz);
        AstronomicalCalendar ac = new AstronomicalCalendar(geoLocation);
        ac.getCalendar().set(year, monthOfYear, dayOfMonth);
        Date srise = ac.getSunrise();
        Date sset = ac.getSunset();

        SimpleDateFormat sdf = new SimpleDateFormat("HH:mm");

        TextView sunriseTV = view.findViewById(R.id.sunriseTimeTV);
        TextView sunsetTV = view.findViewById(R.id.sunsetTimeTV);
        Log.d("SUNRISE Unformatted", srise+"");
```

```
        sunriseTV.setText(sdf.format(srise));
        sunsetTV.setText(sdf.format(sset));
    }
    DatePicker.OnDateChangedListener dateChangeHandler = new
DatePicker.OnDateChangedListener()
    {
        public void onDateChanged(DatePicker dp, int year, int monthOfYear, int
dayOfMonth)
        {
            updateTime(rootView, year, monthOfYear, dayOfMonth);
        }
    };

}
```