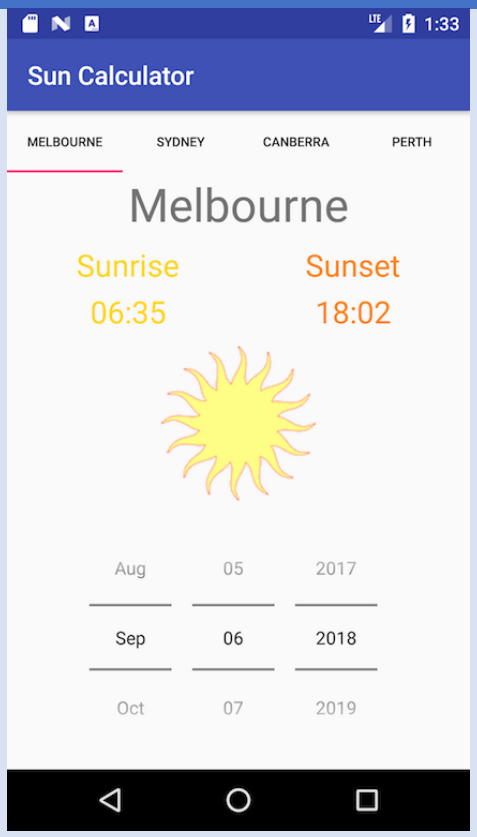
Software Development for Mobile Devices

Submission for Assignment A5.3P

Refactoring the Sun

After refactoring the project based on the specification, the app is now having a tab bar on the top of the screen. A new fragment will be created when the user swipes to a new tab. For example, if the user swipe to Sydney Region, a new fragment will be created and calculate sun activities based on the geolocation.

App Screen

Activity	Screen
MainActivity	

After refactoring, the MainActivity includes only the ViewPager, TabLayout and a FragmentAdapter to manipulate Fragments.

MainActivity

```
public class MainActivity extends AppCompatActivity {

    FragmentAdapter adapterViewPager;
    private List<GeoLocation> australiaLocations = new ArrayList<>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initData();
        ViewPager 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) {
//                Toast.makeText(MainActivity.this,
//                        "Selected page position: " + position,
Toast.LENGTH_SHORT).show();
            }

            // 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));
        addLocation(new GeoLocation("Sydney", -33.868820, 151.209290,tz));
        addLocation(new GeoLocation("Canberra", -35.280937, 149.130005,tz));
```

```
        addLocation(new GeoLocation("Perth", -31.950527, 115.860458,tz));
    }

}
```

Fragment Adapter

```
public class FragmentAdapter extends FragmentStatePagerAdapter {
    List<GeoLocation> australiaLocations;

    public FragmentAdapter(FragmentManager fm, List<GeoLocation> australiaLocations) {
        super(fm);
        this.australiaLocations = australiaLocations;
    }

    @Override
    public Fragment getItem(int position) {
        return SunsetFragment.newInstance(australiaLocations.get(position));
    }

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

    @Override
    public CharSequence getPageTitle(int position) {
        return australiaLocations.get(position).getLocationName();
    }

}
```

A SunsetFragment uses to setup a fragment based on the provided geolocation.

SunsetFragment

```
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);
        }
    };

}
```