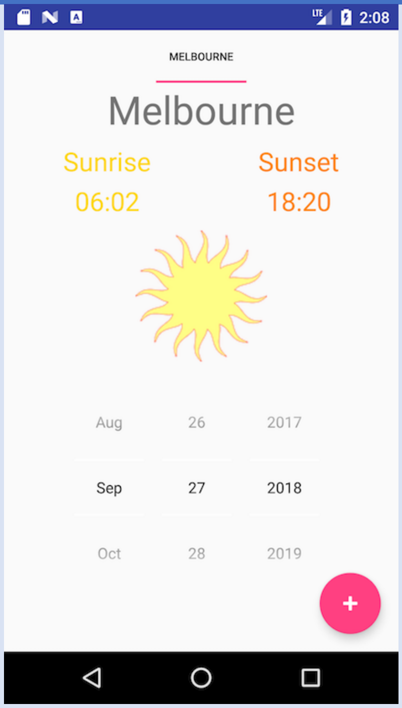
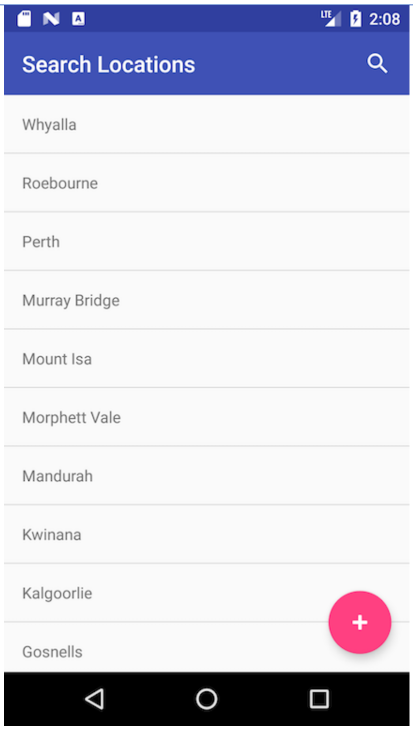
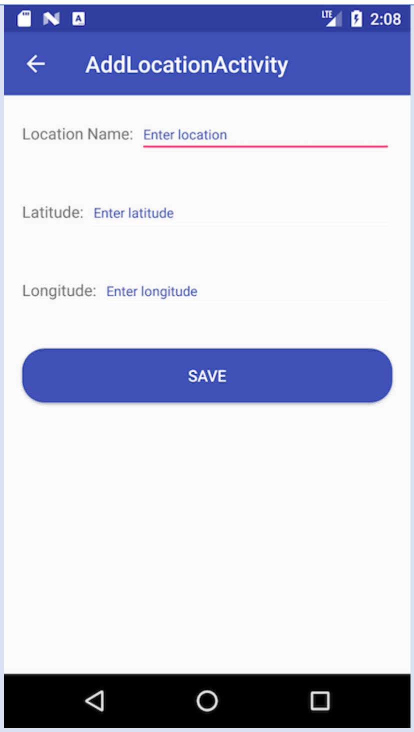
Software Development for Mobile Devices

Submission for Assignment A9.1P

App Screen Shot

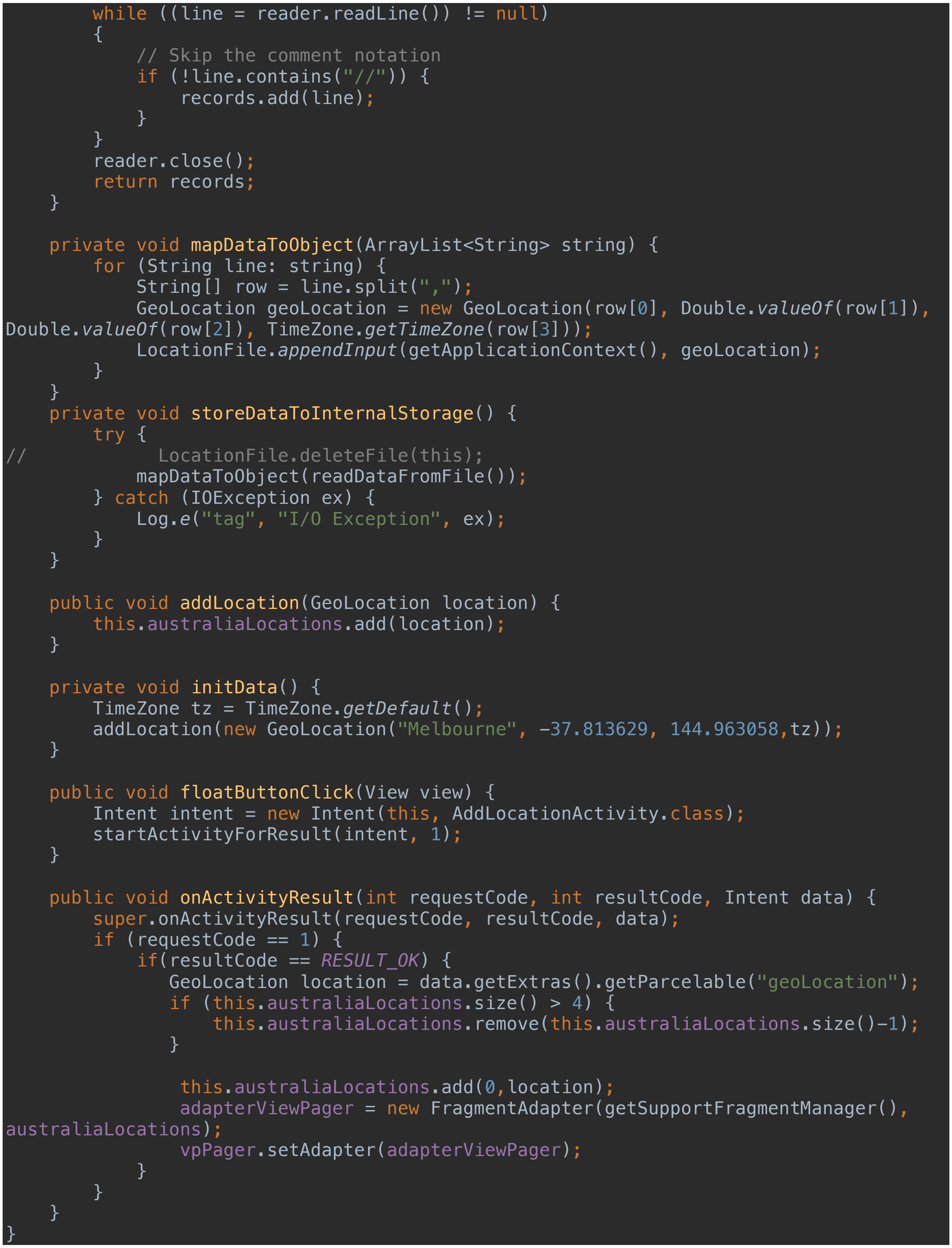
Screen Name	Screen
Home Screen	 Melbourne Sunrise 06:02 Sunset 18:20 Aug 26 2017 Sep 27 2018 Oct 28 2019

Search Screen	 2:08 Search Locations Whyalla Roebourne Perth Murray Bridge Mount Isa Morphett Vale Mandurah Kwinana Kalgoorlie Gosnells +
Add Location Screen	 2:08 AddLocationActivity Location Name: Enter location Latitude: Enter latitude Longitude: Enter longitude SAVE

Source Code

Main Activity

```
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();
        storeDataToInternalStorage();
        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);
    }


    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.getTimeZone(row[3]));
            LocationFile.appendInput(getApplicationContext(), geoLocation);
        }
    }
    private void storeDataToInternalStorage() {
        try {
//            LocationFile.deleteFile(this);
            mapDataToObject(readDataFromFile());
        } catch (IOException ex) {
            Log.e("tag", "I/O Exception", ex);
        }
    }

    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);
            }
        }
    }
}
```

Search Screen – AddLocationActivity

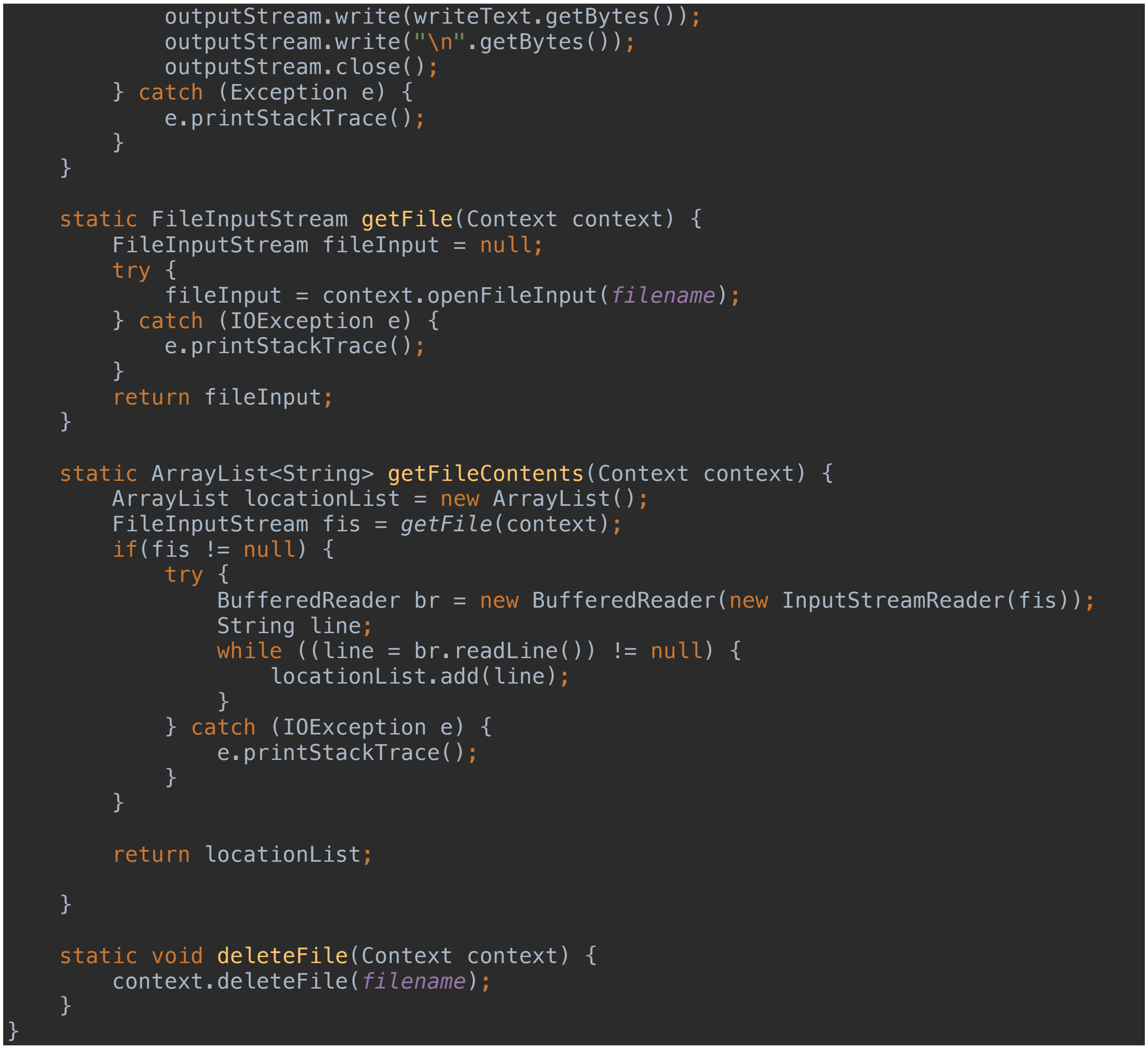
```
public class AddLocationActivity extends AppCompatActivity {

    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();
        mapDataToObject((ArrayList<String>)
LocationFile.getFileContents(getApplicationContext()));
    }

    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 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.getTimeZone(row[3]));
            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;
    }

    public void addGeoLocation(View view) {
        // TODO: Intent to add geolocation activity
        Intent intent = new Intent(this, AddGeoLocation.class);
        startActivityForResult(intent, 1);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(resultCode == RESULT_OK) {
            if (requestCode == 1) {
                mapDataToObject((ArrayList<String>)
LocationFile.getFileContents(getApplicationContext()));
            }
        }

    }

}
```

Add Location – AddGeoLocation

```
public class AddGeoLocation extends AppCompatActivity {

    EditText edtLocationName;
    EditText edtLatitude;
    EditText edtLongitude;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_geo_location);
        Toolbar toolbar =  findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
        setupUI();
    }

    private void setupUI() {
        edtLocationName = findViewById(R.id.edtLocation);
        edtLatitude = findViewById(R.id.edtLatitude);
        edtLongitude = findViewById(R.id.edtLongitude);
    }

    public void saveButtonClick(View view) {
        // TODO: Save data to file and send a call back to previous activity finish
        if (!edtLongitude.getText().toString().equals("") &&
                !edtLatitude.getText().toString().equals("") &&
                !edtLocationName.getText().toString().equals("")) {
            GeoLocation geoLocation = new
GeoLocation(edtLocationName.getText().toString(),
                    Float.valueOf(edtLatitude.getText().toString()),
Float.valueOf(edtLongitude.getText().toString()), TimeZone.getTimeZone("GMT"));
            LocationFile.appendInput(this, geoLocation);
            setResult(RESULT_OK);
            this.finish();
        } else {
            Toast.makeText(this, "Please input value", Toast.LENGTH_SHORT).show();
        }

    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == android.R.id.home) {
            this.finish();
        }
        return super.onOptionsItemSelected(item);
    }
}
```

LocationFile

```
public class LocationFile {
    private static String filename = "au_location.txt";
//

    static void appendInput(Context context, GeoLocation location) {
        FileOutputStream outputStream;

        try {
            outputStream = context.openFileOutput(filename, Context.MODE_APPEND);
            String writeText = location.getLocationName()+ "," +
location.getLatitude() + "," + location.getLongitude() + "," + location.getTimeZone();
```



```
            outputStream.write(writeText.getBytes());
            outputStream.write("\n".getBytes());
            outputStream.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    static FileInputStream getFile(Context context) {
        FileInputStream fileInput = null;
        try {
            fileInput = context.openFileInput(filename);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return fileInput;
    }

    static ArrayList<String> getFileContents(Context context) {
        ArrayList locationList = new ArrayList();
        FileInputStream fis = getFile(context);
        if(fis != null) {
            try {
                BufferedReader br = new BufferedReader(new InputStreamReader(fis));
                String line;
                while ((line = br.readLine()) != null) {
                    locationList.add(line);
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        return locationList;

    }

    static void deleteFile(Context context) {
        context.deleteFile(filename);
    }
}
```