9/10/2020

Android Assignment

BCA - 5

Shashwat Tripathi
A71004818016

1. What is the importance of AndroidManifest.xml?

This file includes essential details about your application that the OS needs to know to actually able to run your app. This file contains app’s package name, all your app activities, permission constraints, the app launcher, the intent filter that basically tells the OS to start the activity it is enclosed within when the launcher icon is clicked and hardware & software features the app requires which determines device compatibility to run the app.

2. What is an Activity? Explain the four important states of the Activity.

An activity is a core android class which is responsible for generating app’s user interface and handles user to app interaction (input events). Every android app launches with one of its activity from the activity bundle set and this is handled by the intent filter in the manifest file. Every activity has a layout has two files;

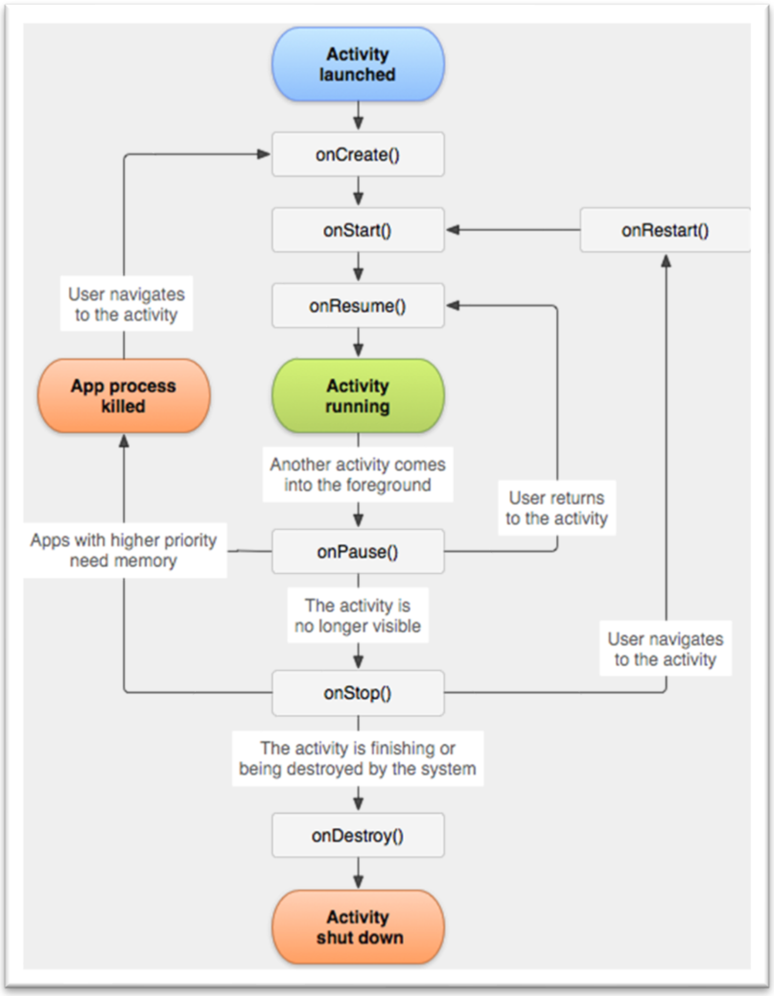
- The layout file(activit.xml) which takes care about “how the app is going to look like” -the core UI which is nothing but multiple view tags (Button,TextView,ImageView etc) put together in a layout container.
- The actual activity(activity.java/kt) file that interacts with the layout file which is inflated during the process called layout inflation which was triggered when the activity was started. This process basically “converts the views defined in the layout file into java/kotlin view objects” in memory. Once this process is executed successfully developer can dynamically modify the layout file along side can generated several logical codes on listening to specific set of events when triggered. For example when client clicks a particular button on the screen then the corresponding logic that resides the java/kotlin file will handle the event and respond to the client accordingly.

The time when user taps the app icon to the time when app initially loads and eventually when user exits out of the app, the app transitioned between several states during the whole process. This process is also known as “Activity Lifecycle”, Let’s see what they are:

1. Active state: When the app initially loads, the main activity is started and the UI is generated as per the layout XMLs(Inflation Process). onCreate() callback is invoked here

and as the app is live and running (onStart() phase), it is said to be in an ACTIVE state.

2. Paused state: When the user encounters some service within the app that partially stops the app and resumes when intended input is being provided by the user(situations like permissions, dialog box etc). This is called as Pause state(onPause()).
3. Stop state: When the user switch to other app without exiting from the existing one then the state of the current app is saved in memory, so the user can continue(onResume()) from where it left. This is when onStop() callback is invoked.
4. Destroy state: When the user exits out of the app completely or due to some issue the app crashes. The app is said to be in Destroy state or simply onDestroy() callback is invoked.


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

A simplified illustration of the activity lifecycle.

3. Differentiate between an Explicit and Implicit Intent.

Implicit Intent	Explicit Intent
Here we are not specifying any external activity but rather just specifying the file type & action type. Then OS filters out apps from the device that can handle such intent request.	We explicitly call another activity from the package by mentioning the app context and the java class file to be invoked.
Here we parse URI(data) to generate results.	We can even pass data between activities using PUT_EXTRA & GET_EXTRA.

Syntax:
Intent imp_intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
startActivity(imp_intent);
	Syntax:
	Intent exp_intent = new Intent(getApplicationContext(), SecondActivityName.class);
	startActivity(exp_intent);

Eg: creating an app that can take picture using camera or like opening a web page link in a browser etc.	Eg: creating an app that takes input using on activity and then displaying results in other activity.

4.Develop an android application to implement Location Based services.

MainActivity.kt

```kotlin
package com.example.permissionoverview

import android.Manifest
import android.content.pm.PackageManager
import android.os.Bundle
import android.util.Log
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import com.google.android.gms.location.FusedLocationProviderClient
import com.google.android.gms.location.LocationServices
import kotlinx.android.synthetic.main.activity_main.*

//**** Very Very Important!
//NOTE: before running this app open Google maps app in your AVD, so that there is at l
east some confirmed location, and then test your app.

class MainActivity : AppCompatActivity() {
    private val permissionCode = 101 //request code id for the FINE location request
    private lateinit var fusedLocationClient: FusedLocationProviderClient
    private lateinit var latitudeTV: TextView
    private lateinit var longitudeTV: TextView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        latitudeTV = findViewById(R.id.latitutdeTV)
        longitudeTV = findViewById(R.id.longitudeTV)
        fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)

        locationBtn.setOnClickListener {

            if(hasLocationPermission())
                getLastLocation()
            else
                requestLocation()
        }
    }

    private fun getLastLocation(){
        if (ActivityCompat.checkSelfPermission(
                this,
                Manifest.permission.ACCESS_FINE_LOCATION
            ) == PackageManager.PERMISSION_GRANTED
        ) {
            fusedLocationClient.lastLocation
                .addOnSuccessListener(
                    this
                ) { location ->
                    // Got last known location. In some rare situations this can be null.
                    if (location != null) {
                        latitudeTV.text = location.latitude.toString()
                        longitudeTV.text = location.longitude.toString()
                        Log.d("Location: ", location.toString())
                    }
                    Log.d("Location: ", "null")
                }
        }else
            requestLocation()

    }


   private fun hasLocationPermission()  =  ActivityCompat.checkSelfPermission(
       this,
       Manifest.permission.ACCESS_FINE_LOCATION
   ) == PackageManager.PERMISSION_GRANTED

    private fun requestLocation(){
       var permissionsToRequest = mutableListOf<String>(); //initializing permission list

       //if location permission is not granted
        if(!hasLocationPermission()){
            permissionsToRequest.add(Manifest.permission.ACCESS_FINE_LOCATION)

            //in case user have denied the request earlier when asked
            if(ActivityCompat.shouldShowRequestPermissionRationale(
                    this,
                    Manifest.permission.ACCESS_FINE_LOCATION
                )){
                Toast.makeText(
                    applicationContext,
                    "Allow location service to access this feature!",
                    Toast.LENGTH_LONG
                ).show()
            }
            //if asked for the first time
            ActivityCompat.requestPermissions(
                this,
                permissionsToRequest.toTypedArray(),
                permissionCode
            )
        }
        else{
            Toast.makeText(
                applicationContext,
                "Already have location permission!",
                       Toast.LENGTH_SHORT
                   ).show()
               }
           }

           //in-order to check if user have either allowed or denied the request
           //we will override onRequestPermissionResult()

           override fun onRequestPermissionsResult(
               requestCode: Int,
               permissions: Array<out String>,
               grantResults: IntArray
           ) {
               super.onRequestPermissionsResult(requestCode, permissions, grantResults)

               if(requestCode == permissionCode && grantResults.isNotEmpty())
               {
                   //permission granted
                   //as we have just one permission to ask so looping is not required
                   if(grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                       Toast.makeText(
                           applicationContext,
                           "Location permission granted!",
                           Toast.LENGTH_LONG
                       ).show()
                       getLastLocation()
                   }
                   //permission not granted
                   else
                       Toast.makeText(applicationContext, "Location permission denied!"
, Toast.LENGTH_LONG).show()
               }
           }
       }
```

Myapplication.java

```java
package com.example.permissionoverview;

//This class holds a global list of location
//This will extend Application class as super class
//So the list is available to every other class in this application
//We setting up this class as singleton class: at most one instance of this class
//NOTE: don't forget to add this class name as "android:name= className" in the manifes
t file, otherwise while typecasting the context it will throw error

import android.app.Application;
import android.location.Address;
import android.location.Location;

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

public class MyApplication extends Application {


    //location list: is accessible to every file in this app via class instance
    private List<Location> myLocations;

    //initializing this class as singleton class
    public  static  MyApplication singleton_Instance;

    //return class instance
    public MyApplication getInstance(){
       return singleton_Instance;
    }

    //Kind of constructor that creates the instance of  "MyApplication" class type;
    public void onCreate(){
        super.onCreate();
        singleton_Instance = this;
        //this: MyApplication

        //initializing list
        myLocations = new ArrayList<>();
    }

    public List<Location> getMyLocations() {
        return myLocations;
    }

    public void setMyLocations(List<Location> myLocations) {
        this.myLocations = myLocations;
    }

}
```

TrackUserActivity.java

```java
package com.example.permissionoverview;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.Intent;
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.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Switch;
import android.widget.TextView;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;

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

public class TrackUserActivity extends AppCompatActivity {
    private static final int PERIMISSIONS_FINE_LCOATION = 101;
    //References to UI elements
    TextView tv_lat, tv_long, tv_alt, tv_speed, tv_address, tv_accuracy, tv_sensor, tv_update, tv_wayPointCounts;
    Switch loc_update, sensor_update;
    Button btn_newWayPoint,btn_showWayPointList, btn_mapIntent;

    //Google's API (Location Service)
    FusedLocationProviderClient FLPC;

    //Location Request is config file for all settings belongs to FLPC
    LocationRequest LOC_RQ;

    //current location tracker
    Location current_loc;

    //List of saved locations
    List<Location> savedLocations;


    //location callback for loc_updates
    LocationCallback loc_callBack;

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

        //Buttons
        btn_newWayPoint = findViewById(R.id.btn_newWayPoint);  //adds current/new loc to the global list
        btn_showWayPointList = findViewById(R.id.btn_showWayPointList); //display all loc present on the global list
        btn_mapIntent = findViewById(R.id.btn_intent_map); //display all loc on map

        //TVs
        tv_lat = findViewById(R.id.tv_latitude);
        tv_long = findViewById(R.id.tv_longitude);
        tv_alt = findViewById(R.id.tv_altitude);
        tv_speed = findViewById(R.id.tv_speed);
        tv_address = findViewById(R.id.tv_address);
        tv_accuracy = findViewById(R.id.tv_accuracy);
        tv_sensor = findViewById(R.id.tv_labelsensor);
        tv_update = findViewById(R.id.tv_labelupdates);
        tv_wayPointCounts =  findViewById(R.id.tv_wayPoint);

        //Switches
        loc_update = findViewById(R.id.sw_locationsupdates);
        sensor_update = findViewById(R.id.sw_gps);

        //Setting all properties for location req obj
        LOC_RQ = new LocationRequest();
        LOC_RQ.setInterval(1000 * 30);
        LOC_RQ.setFastestInterval(1000 * 5);
        LOC_RQ.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY);

        //loc_callBacks: is triggered whenever the update interval is met.
        loc_callBack = new LocationCallback() {

            @Override
            public void onLocationResult(LocationResult locationResult) {
                super.onLocationResult(locationResult);

                //save the location
                Location current_loc = locationResult.getLastLocation();
                updateUI(current_loc);
            }
        };
        //sensor update listener
        sensor_update.setOnClickListener(new View.OnClickListener() {
            @Override
                  public void onClick(View view) {
                      if (sensor_update.isChecked()) {
                          //more accuracy
                          LOC_RQ.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

                          tv_sensor.setText("Using GPS Sensors");
                      } else {
                          LOC_RQ.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY);
                          tv_sensor.setText("Cellular Towers + WIFI");
                      }

                  }
              });

              //location update listener
              loc_update.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
            public void onClick(View view) {
                if (loc_update.isChecked())
                    startLocationUpdates();
                else
                    stopLocationUpdates();
            }
        });

        //adding current loc to the location global list
        btn_newWayPoint.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //get the gps location

                //add the new location to the global list

                MyApplication myApp = (MyApplication)getApplicationContext();
                //here we basically typecasting the application context wrt the class we instantiating
                //in our case "MyApplication" is the class

                //now we are getting the global list from the class using its instance so that we can add several locs in it.
                savedLocations = myApp.getMyLocations();  //return the global list
                savedLocations.add(current_loc);  //adding loc to the list
                //current_loc is updated by updateGPS() method
            }
        });

        //show the global list using intent
        btn_showWayPointList.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent loc_list_intent= new Intent(getApplicationContext(),ShowSavedLocationsList.class);
                startActivity(loc_list_intent);
            }
        });

        //map intent
        btn_mapIntent.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent mapIntent = new Intent(getApplicationContext(), MapsActivity.class);
                startActivity(mapIntent);
            }
        });
        updateGPS();
    }

    private void startLocationUpdates() {
        tv_update.setText("Tracking Enabled");
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
            FLPC.requestLocationUpdates(LOC_RQ, loc_callBack, null);
            //loc_callBacks is for updating our UI with updated location co-ordinates
            updateGPS();
```

```
                }
            }

            private void stopLocationUpdates() {
                String mssg = "Not tracking location";
                tv_update.setText("Tracking Disabled");
                tv_address.setText(mssg);
                tv_long.setText(mssg);
                tv_alt.setText(mssg);
                tv_accuracy.setText(mssg);
                tv_lat.setText(mssg);
                tv_speed.setText(mssg);

                //Dismissing the update location service
                FLPC.removeLocationUpdates(loc_callBack);
                //basically saying we don't want any location update from loc_callback method
            }

            private void updateGPS(){
                //gets user permission to track latest location
                //get current loc from fused client
                //update UI: with all req loc info

                //Initializing fuse client obj
                FLPC = LocationServices.getFusedLocationProviderClient(this);
                if(ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED){
                    //user has granted permission
                    FLPC.getLastLocation().addOnSuccessListener(this, new OnSuccessListener<Location>() {
                        @Override
                        public void onSuccess(Location user_loc) {
                            //we got permissions.
                            //here we can access all sorts of data and update UI
                            updateUI(user_loc);

                            //storing the loc to the global var
                            current_loc = user_loc;

                        }
                    });
                }
                else{
                    //permission not granted yet
                    //request permission from user

                    //1- checking if the API level of device is 23 or higher
                    //M: Marshmallow
                    if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
                        requestPermissions(new String[] {Manifest.permission.ACCESS_FINE_LOCATION}, PERIMISSIONS_FINE_LCOATION);
                    }
                }

            }

            //accepts the current location obj in order to fetch all req data
            private void updateUI(Location current_loc) {
                tv_long.setText(String.valueOf(current_loc.getLongitude()));
                tv_lat.setText(String.valueOf(current_loc.getLatitude()));
```

```java
            tv_accuracy.setText(String.valueOf(current_loc.getAccuracy()));

            //not every phone has the ability to calc the altitude
            if(current_loc.hasAltitude())
                tv_alt.setText(String.valueOf(current_loc.getAltitude()));
            else
                tv_alt.setText("Not Available!");

            //not every phone has the ability to calc the speed
            if(current_loc.hasSpeed())
                tv_speed.setText(String.valueOf(current_loc.getSpeed()));
            else
                tv_speed.setText("Not Available!");

           //address??
            Geocoder geo_loc_obj = new Geocoder(getApplicationContext());
            try{
                //this list can hold one or many addresses as per the given loc_lat & loc_long
               List<Address> addresses = geo_loc_obj.getFromLocation(current_loc.getLatitude(),current_loc.getLongitude(),1);
                tv_address.setText(addresses.get(0).getAddressLine(0));

            }
            catch(Exception e){
                tv_address.setText("Unable to fetch address!");
                Log.d("Loc Address", e.toString());
            }

            //to show total waypoints/crumbs/locations present in the global list
            try {
                tv_wayPointCounts.setText(Integer.toString(savedLocations.size()));

            }
            catch (Exception e){
                tv_wayPointCounts.setText(R.string.defaultVal);
                Log.d("WayPoint_Lenght:", "Initially the global loc list will be zero, it returns the null point exception");
            }


        }

        // this is one of many methods belongs to the base class: AppCompatActivity

        //it basically tells the program to trigger this method after permissions have been granted!
        @Override
        public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
            super.onRequestPermissionsResult(requestCode, permissions, grantResults);

            //we specifically looking for the grantResults[] wrt fine_loc_req
            switch(requestCode){
                case PERIMISSIONS_FINE_LCOATION:
                    if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                        updateGPS();
                        Log.d("Loc Permission:"," Accepted!");
                    }
                    else{
                        Log.d("Loc Permission:"," Denied!");
                    }
                    break;
            }
        }
    }
```

MapsActivity.java

```
package com.example.permissionoverview;

import androidx.fragment.app.FragmentActivity;

import android.location.Location;
import android.os.Bundle;

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

import java.util.List;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    List<Location> allSavedLocations;

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


        //getting the single class's global location list
        MyApplication myApp = (MyApplication)getApplicationContext();
        allSavedLocations = myApp.getMyLocations();
    }

    /**
     * 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
```

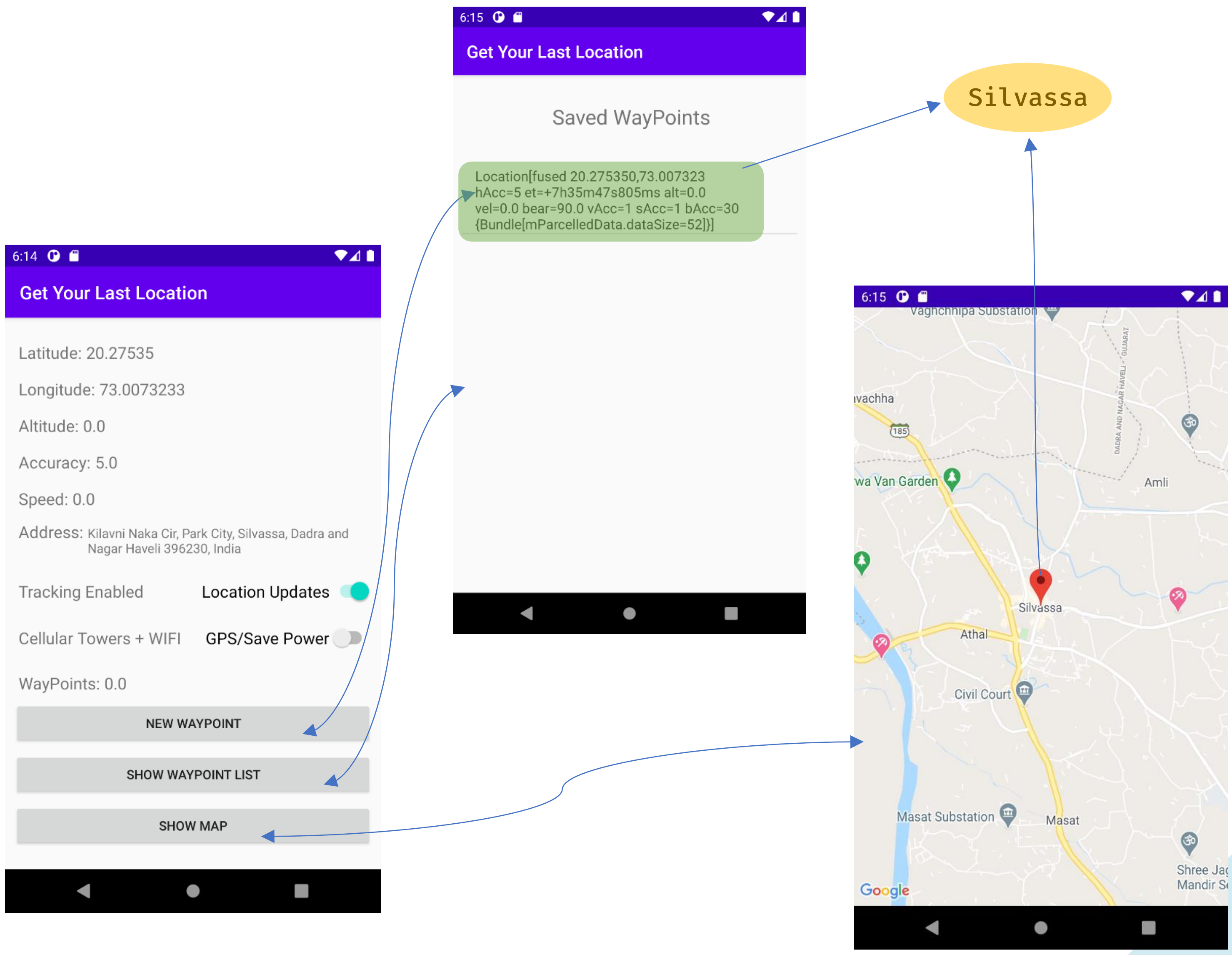
```java
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        // Add a marker in Sydney and move the camera
         LatLng sydney = new LatLng(-34, 151);
        //mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
        //mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));

        LatLng lastLocation = sydney; //considering sydney as default last location
        //iterating through our saved location list
        //and creating marker for each loc
        for(Location loc: allSavedLocations){
            LatLng current_loc = new LatLng(loc.getLatitude(),loc.getLongitude());

            //here we creating the pin/marker to point at the specified location
            MarkerOptions marker = new MarkerOptions();
            marker.position(current_loc);
            marker.title("Lat:"+ loc.getLatitude()+"Lng:"+loc.getLongitude());
            mMap.addMarker(marker);

            //updated lastLoc with current_loc;
            lastLocation = current_loc;
        }

        //now zooming into our last location point
        mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(lastLocation,13.5f)); //12.0 is the zoom intensity(float type)
        //higher the zoom intensity bigger the zoom scope!

```

ShowSavedLocation.java

```java
package com.example.permissionoverview;

import androidx.appcompat.app.AppCompatActivity;

import android.location.Location;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;


import java.util.List;

public class ShowSavedLocationsList extends AppCompatActivity {
 ListView lv_savedLocations;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_show_saved_locations_list);

        //List view UI target
        lv_savedLocations = findViewById(R.id.lv_saved_loc_list);

        //creating the singleton class instance inorder to access the global loc list
        MyApplication myApp = (MyApplication)getApplicationContext();

        //getting the copy of the actual list
        List<Location> allSavedLocations = myApp.getMyLocations(); //returns the list copy
        //this is also our resource/data that is to be injected in the list view

        lv_savedLocations.setAdapter(new ArrayAdapter<Location>(this,android.R.layout.simple_list_item_1,allSavedLocations));

    }
}
```

Snapshots:

Tracking current location, saving it as a way point & displaying it on google maps:



6:14

Get Your Last Location

Latitude: 20.27535

Longitude: 73.0073233

Altitude: 0.0

Accuracy: 5.0

Speed: 0.0

Address: Kilavni Naka Cir, Park City, Silvassa, Dadra and Nagar Haveli 396230, India

Tracking Enabled — Location Updates (on)

Cellular Towers + WIFI — GPS/Save Power (off)

WayPoints: 0.0

NEW WAYPOINT

SHOW WAYPOINT LIST

SHOW MAP

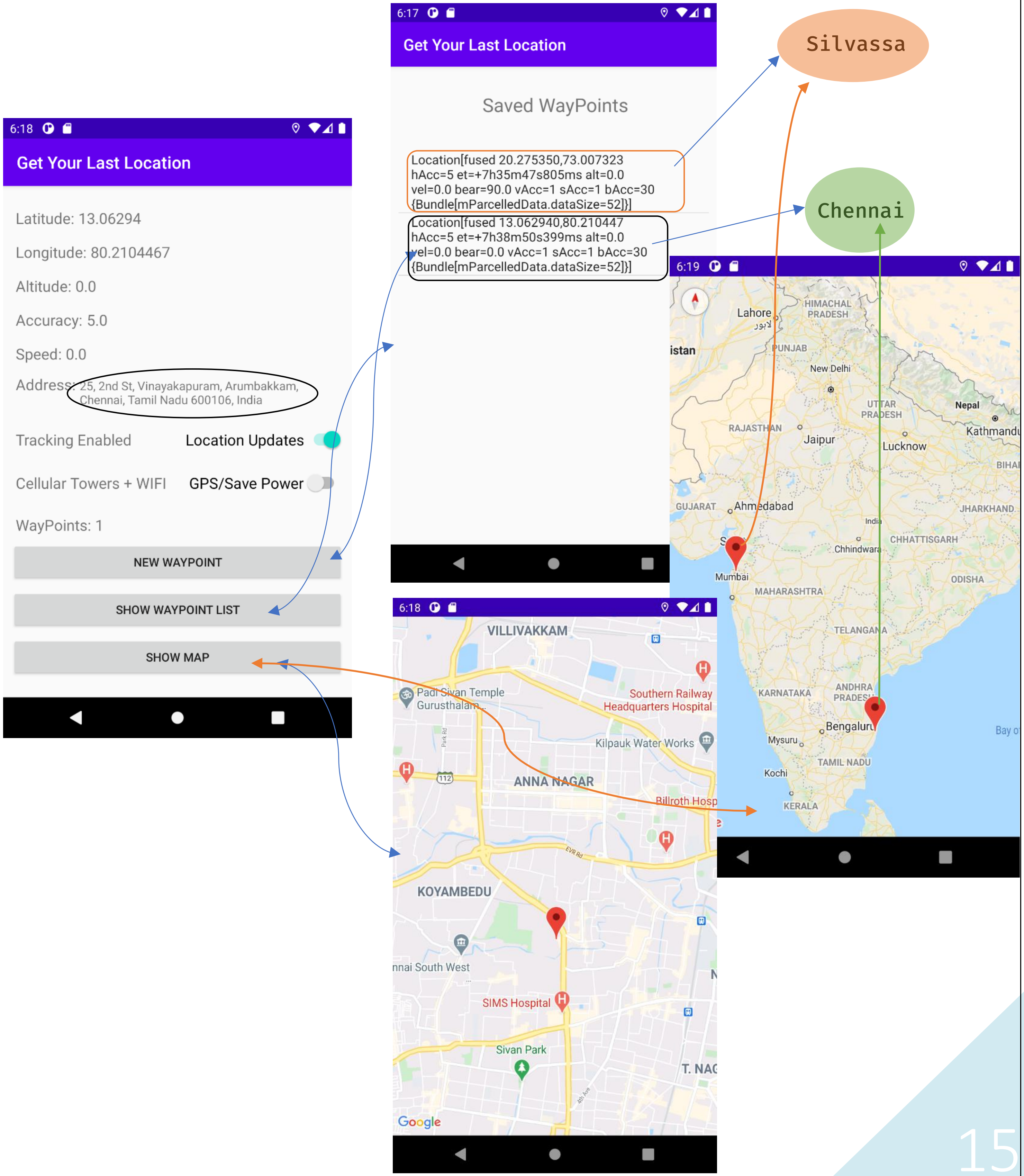
6:15

Get Your Last Location

Saved WayPoints

Location[fused 20.275350,73.007323 hAcc=5 et=+7h35m47s805ms alt=0.0 vel=0.0 bear=90.0 vAcc=1 sAcc=1 bAcc=30 {Bundle[mParcelledData.dataSize=52]}]

Changing location from Silvassa(current) to Chennai(new), setting new way point and then displaying all waypoints on the google map:



6:18

Get Your Last Location

Latitude: 13.06294

Longitude: 80.2104467

Altitude: 0.0

Accuracy: 5.0

Speed: 0.0

Address: 25, 2nd St, Vinayakapuram, Arumbakkam, Chennai, Tamil Nadu 600106, India

Tracking Enabled — Location Updates

Cellular Towers + WIFI — GPS/Save Power

WayPoints: 1

NEW WAYPOINT

SHOW WAYPOINT LIST

SHOW MAP

6:17

Get Your Last Location

Saved WayPoints

```
Location[fused 20.275350,73.007323
hAcc=5 et=+7h35m47s805ms alt=0.0
vel=0.0 bear=90.0 vAcc=1 sAcc=1 bAcc=30
{Bundle[mParcelledData.dataSize=52]}]
```

```
Location[fused 13.062940,80.210447
hAcc=5 et=+7h38m50s399ms alt=0.0
vel=0.0 bear=0.0 vAcc=1 sAcc=1 bAcc=30
{Bundle[mParcelledData.dataSize=52]}]
```

6:18

6:19

5. Develop an android application to show reading and writing data into a file using streams.

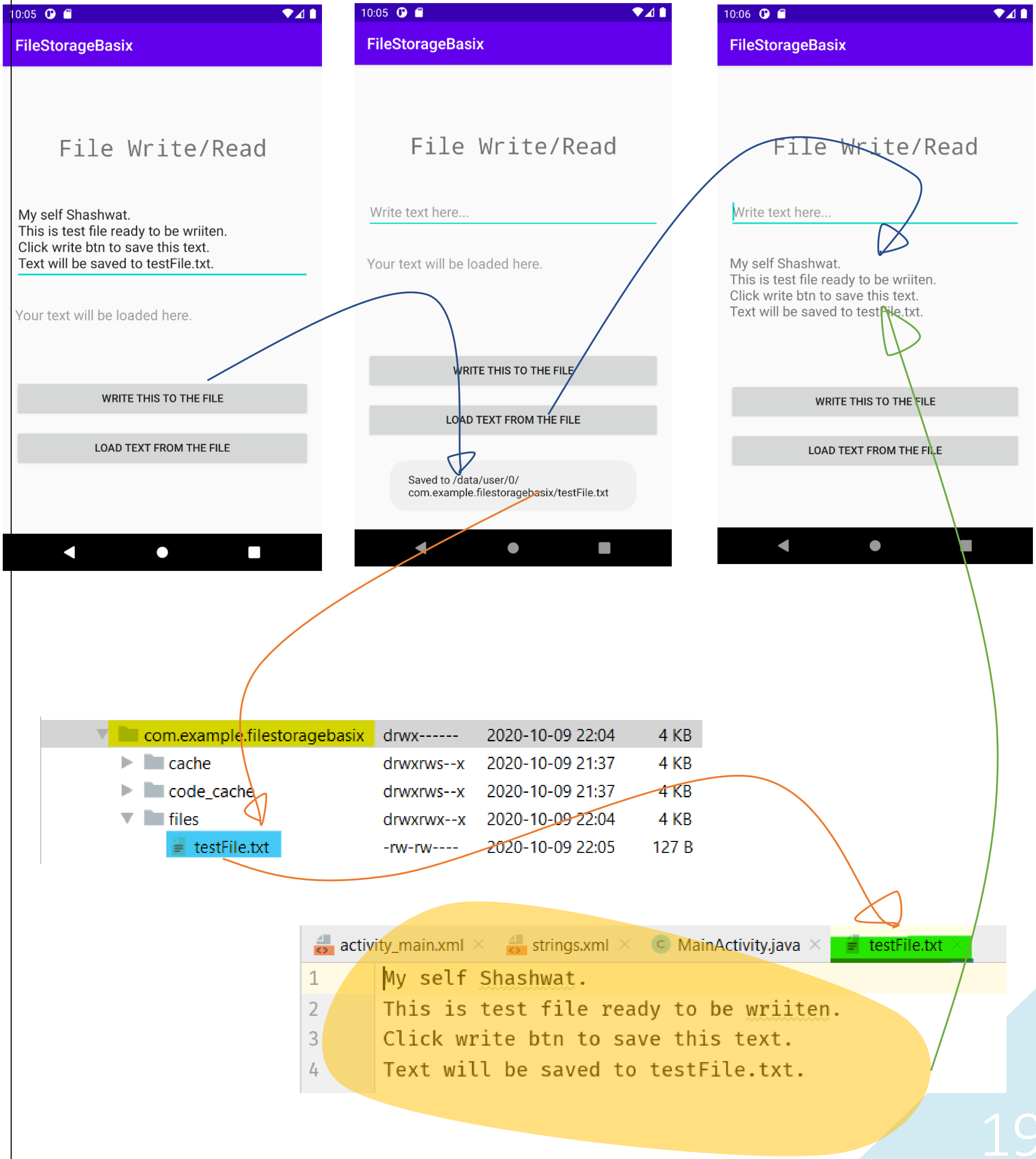
Actvity_man.java

```java
package com.example.filestoragebasix;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity {

    private static final String FILE_NAME = "testFile.txt";
    private EditText content;
    private TextView loadContent;

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

        content = findViewById(R.id.et_writeFile);
        loadContent = findViewById(R.id.tv_loadFile);

        //read or load
        findViewById(R.id.btn_loadFile).setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View view) {
                try {
                    loadFile();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });

        //write or save
        findViewById(R.id.btn_wrtieFile).setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                try {
                    saveFile();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });

    }

    public void saveFile() throws IOException {
        String entered_txt = content.getText().toString();
        FileOutputStream fos = openFileOutput(FILE_NAME,MODE_PRIVATE);
        fos.write(entered_txt.getBytes());

        content.getText().clear();
        Toast.makeText(getApplicationContext(),"Saved to "+getDataDir()+"/"+FILE_NAME,Toast.LENGTH_LONG).show();
        fos.close();
    }

    public void loadFile() throws IOException {
        FileInputStream fis = openFileInput(FILE_NAME);
        InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader bfr = new BufferedReader(isr);
        StringBuilder str_bldr = new StringBuilder();
        String text;

        while((text = bfr.readLine()) != null){
            str_bldr.append(text).append("\n");
        }
        loadContent.setText(str_bldr.toString());
        fis.close();
    }
}
```

Activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    android:padding="16dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tv_loadFile"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:hint="@string/tv_loadFile_hint"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/et_writeFile" />

    <EditText
        android:id="@+id/et_writeFile"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="44dp"
        android:ems="10"
        android:gravity="start|top"
        android:hint="@string/et_writeFile"
        android:inputType="textMultiLine"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/appTitle" />

    <TextView
        android:id="@+id/appTitle"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="68dp"
        android:fontFamily="monospace"
        android:text="@string/appTitle_txt"
        android:textAlignment="center"
        android:textSize="30sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btn_loadFile"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/btn_loadFile_txt"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btn_wrtieFile" />

    <Button
        android:id="@+id/btn_wrtieFile"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="@string/btn_writeFile_txt"
        app:layout_constraintBottom_toTopOf="@+id/btn_loadFile"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_loadFile"
        app:layout_constraintVertical_chainStyle="packed" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Snapshots:



My self Shashwat.
This is test file ready to be wriiten.
Click write btn to save this text.
Text will be saved to testFile.txt.

Saved to /data/user/0/com.example.filestoragebasix/testFile.txt

My self Shashwat.
This is test file ready to be wriiten.
Click write btn to save this text.
Text will be saved to testFile.txt.

com.example.filestoragebasix	drwx------	2020-10-09 22:04	4 KB
cache	drwxrws--x	2020-10-09 21:37	4 KB
code_cache	drwxrws--x	2020-10-09 21:37	4 KB
files	drwxrwx--x	2020-10-09 22:04	4 KB
testFile.txt	-rw-rw----	2020-10-09 22:05	127 B

```
My self Shashwat.
This is test file ready to be wriiten.
Click write btn to save this text.
Text will be saved to testFile.txt.
```