Appendix

Appendix A

- mobile application Screen
    - Start Page Interface

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

   <Button
      android:id="@+id/button1"
      android:layout_width="250dp"
      android:layout_height="50dp"
      android:background="@drawable/button_sample1"
      android:layout_alignParentBottom="true"
      android:layout_centerHorizontal="true"
      android:text="@string/ok"
      android:clickable="true"
      android:onClick="okay"
      android:layout_marginBottom="10dp" />

   <ImageView
      android:src="@drawable/ic_launcher"
      android:layout_height="300dp"
      android:layout_width="200dp"

      android:layout_below="@+id/app_name"
      android:layout_centerHorizontal="true" />
   <TextView
      android:id="@+id/app_name"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="40dp"
      android:text="@string/friend_locator"
      android:fontFamily="sans-serif-condensed"
      android:textAppearance="?android:attr/textAppearanceLarge"
      android:textColor="#ff8f06"
      android:textSize="40sp"
      android:textStyle="bold"
      android:typeface="monospace" />
```

```
</RelativeLayout>
```

- Login Interface

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingBottom="@dimen/activity_vertical_margin"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   tools:context=".Login_page" >

   <EditText
      android:id="@+id/password"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerHorizontal="true"
      android:layout_centerVertical="true"
      android:ems="10"
      android:inputType="textPassword"
      android:hint="@string/password" >


   </EditText>

   <EditText
      android:id="@+id/username"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_above="@+id/password"
      android:layout_alignLeft="@+id/password"
      android:layout_alignRight="@+id/password"
      android:ems="10"
      android:layout_marginBottom="5dp"
      android:hint="@string/username" >

      <requestFocus />
```

```
  </EditText>

  <Button
    android:id="@+id/login"
    android:layout_width="100dp"
    android:layout_height="40dp"
    android:layout_alignLeft="@+id/password"
    android:layout_below="@+id/password"
    android:layout_marginTop="20dp"
    android:text="@string/login"
    android:background="#ff8f06"
    android:clickable="true"
    android:onClick="login" />

  <Button
    android:id="@+id/button2"
    android:layout_width="100dp"
    android:layout_height="40dp"
    android:layout_alignBaseline="@+id/login"
    android:layout_alignBottom="@+id/login"
    android:layout_alignRight="@+id/password"
    android:text="@string/signup"
    android:background="#ff8f06"
    android:clickable="true"
    android:onClick="signup" />

</RelativeLayout>
```

- Signup Interface

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

  <EditText
    android:id="@+id/password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:ems="10"
```

```
    android:hint="@string/password"
    android:inputType="textPassword" >


</EditText>

<EditText
    android:id="@+id/username"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/password"
    android:layout_alignLeft="@+id/password"
    android:layout_alignRight="@+id/password"
    android:ems="10"
    android:layout_marginBottom="5dp"
    android:hint="@string/username" >

    <requestFocus />
</EditText>

<EditText
    android:id="@+id/email"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/password"
    android:layout_alignRight="@+id/password"
    android:layout_below="@+id/password"
    android:ems="10"
    android:hint="@string/email"
    android:inputType="textEmailAddress"
    android:layout_marginTop="5dp" />

<Button
    android:id="@+id/sign_up"
    android:layout_width="100dp"
    android:layout_height="40dp"
    android:layout_below="@+id/email"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:text="@string/signup"
    android:background="#ff8f06"
    android:clickable="true"
    android:onClick="sign_up" />
```

```
</RelativeLayout>
```

- Main Activity Interface

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingBottom="@dimen/activity_vertical_margin"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   tools:context=".G_Maps" >



   <fragment
      android:id="@+id/map"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      class="com.google.android.gms.maps.MapFragment" />

</RelativeLayout>
```

- Friends Page Interface

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

   <ListView
      android:id="@+id/friends_list"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:cacheColorHint="@android:color/transparent"
      android:background="#ff8f06" >
   </ListView>
```

```
</RelativeLayout>
```

- Profile Page Interface

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

   <TextView
      android:id="@+id/user"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:textIsSelectable="true"
      android:layout_marginTop="5dp"
      android:textAppearance="?android:attr/textAppearanceLarge" />

   <TextView
      android:id="@+id/email"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@+id/user"
      android:layout_centerHorizontal="true"
      android:textIsSelectable="true"
      android:layout_marginTop="1dp"
      android:textAppearance="?android:attr/textAppearanceSmall" />

   <EditText
      android:id="@+id/search"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentLeft="true"
      android:layout_below="@+id/email"
      android:layout_marginLeft="10dp"
      android:layout_marginTop="5dp"
      android:layout_marginRight="5dp"
      android:hint="@string/search"
      android:layout_toLeftOf="@+id/search_button"
```

```
    android:ems="10"
    android:textColorHint="@android:color/black" />

  <Button
    android:id="@+id/search_button"
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:layout_alignBottom="@+id/search"
    android:layout_alignParentRight="true"
    android:layout_alignTop="@+id/search"
    android:layout_marginRight="10dp"
    android:background="@drawable/search_new"
    android:clickable="true"
    android:onClick="search_method" />

  <ListView
    android:id="@+id/search_list"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/search"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="5dp"
    android:background="#ff8f06"
    android:cacheColorHint="@android:color/transparent" >

  </ListView>

</RelativeLayout>
```

- Main Page Interface

```
<?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" >

  <TabHost
    android:id="@android:id/tabhost"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
```

```
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TabWidget
      android:id="@android:id/tabs"
      android:layout_width="match_parent"
      android:layout_height="wrap_content" >
    </TabWidget>

    <FrameLayout
      android:id="@android:id/tabcontent"
      android:layout_width="match_parent"
      android:layout_height="match_parent" >

      <LinearLayout
        android:id="@+id/tab1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <TabWidget
          android:id="@android:id/tabs"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content" >
        </TabWidget>

      </LinearLayout>

      <LinearLayout
        android:id="@+id/tab2"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >
      </LinearLayout>

      <LinearLayout
        android:id="@+id/tab3"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >
      </LinearLayout>
    </FrameLayout>
  </LinearLayout>
</TabHost>
```

```
</LinearLayout>
```

- Notification Page Interface

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

   <TextView
      android:id="@+id/Noti_text"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="5dp"
      android:text="@string/notification"
      android:textAppearance="?android:attr/textAppearanceMedium" />

   <ListView
      android:id="@+id/list_notification"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_below="@+id/Noti_text"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="10dp"
      android:cacheColorHint="@android:color/transparent"
      android:background="#ff8f06" >
   </ListView>

</RelativeLayout>
```

Appendix B

- Database Activity Handler

```
package com.amira.googlemaps;

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

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class DatabaseHandlerActivity extends SQLiteOpenHelper
{

  // All Static variables
  // Database Version
  private static final int DATABASE_VERSION = 1;


  // Database Name
  private static final String DATABASE_NAME ="friendslocator";

  // Contacts table name
  private static final String TABLE_LOCATIONS = "flocation";

  // Contacts Table Columns names
  private static final String KEY_ID = "id";
  private static final String KEY_LAT = "latitude";
  private static final String KEY_LON = "longitude";
  private static final String KEY_TIME = "time";


  public DatabaseHandlerActivity(Context context)
  {
    super(context, DATABASE_NAME, null,DATABASE_VERSION);
    // TODO Auto-generated constructor stub
  }


  @Override
  public void onCreate(SQLiteDatabase db)
  {
    // TODO Auto-generated method stub


    //creating table values

    String CREATE_CONTACTS_TABLE = "CREATE TABLE " + TABLE_LOCATIONS + "("
    + KEY_ID + " INTEGER PRIMARY KEY," + KEY_LAT + " TEXT,"
    + KEY_LON + " TEXT," + KEY_TIME + " TEXT" + ")";
    db.execSQL(CREATE_CONTACTS_TABLE);
  }

  //Upgrading table
  @Override
  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // TODO Auto-generated method stub
```

```
    // Drop older table if existed
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_LOCATIONS);

    // Create tables again
    onCreate(db);

  }


  // Adding new contact
  public void addLocation(Locations_viewer location)
  {
    SQLiteDatabase db = this.getWritableDatabase();// making writable database

    ContentValues values = new ContentValues();
    values.put(KEY_LAT, location.getLat()); // latitude
    values.put(KEY_LON, location.getLong()); // longitude
    values.put(KEY_TIME, location.getTime());

    // Inserting Row
    db.insert(TABLE_LOCATIONS, null, values);
    db.close(); // Closing database connection
  }

  // Getting single contact
  public Locations_viewer getLocation(int id)
  {
    SQLiteDatabase db = this.getReadableDatabase();

    Cursor cursor = db.query(TABLE_LOCATIONS, new String[] { KEY_ID,KEY_LAT, KEY_LON,KEY_TIME }, KEY_ID + "=?",
        new String[] { String.valueOf(id) }, null, null, null, null);
    if (cursor != null)
      cursor.moveToFirst();

    Locations_viewer contact = new Locations_viewer(Integer.parseInt(cursor.getString(0)),cursor.getString(1),
cursor.getString(2),cursor.getString(3));
    // return contact
    return contact;
  }

  // Getting All Contacts
  public List<Locations_viewer> getAllLocations() {
    List<Locations_viewer> contactList = new ArrayList<Locations_viewer>();

    // Select All Query
    String selectQuery = "SELECT  * FROM " + TABLE_LOCATIONS;

    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);

    // looping through all rows and adding to list
    if (cursor.moveToFirst()) {
      do {
```

```
        Locations_viewer location = new Locations_viewer();
        location.setID(Integer.parseInt(cursor.getString(0)));
        location.setLAt(cursor.getString(1));
        location.setLongi(cursor.getString(2));
        location.setTime(cursor.getString(3));

        // Adding contact to list
        contactList.add(location);
      } while (cursor.moveToNext());
    }

    // return contact list
    return contactList;
}


// Getting contacts Count
public int getLocationsCount()
{
    String countQuery = "SELECT  * FROM " + TABLE_LOCATIONS;

    SQLiteDatabase db = this.getReadableDatabase();//database get readable
    Cursor cursor = db.rawQuery(countQuery, null);
    cursor.close();

    // return count
    return cursor.getCount();
}


// Updating single contact
public int updateLocation(Locations_viewer locations_viewer)
{
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(KEY_LAT, locations_viewer.getLat());
    values.put(KEY_LON, locations_viewer.getLong());
    values.put(KEY_TIME, locations_viewer.getTime());
    Log.d("Latitude",locations_viewer.getLat());
    Log.d("Longitude",locations_viewer.getLong());
    Log.d("Time", locations_viewer.getTime());

    // updating row
    return db.update(TABLE_LOCATIONS, values, KEY_ID + " = ?",new String[] { String.valueOf(locations_viewer.getID()) });
}

// Deleting single contact
public void deleteLocation(Locations_viewer locations_viewer) {SQLiteDatabase db = this.getWritableDatabase();
db.delete(TABLE_LOCATIONS, KEY_ID + " = ?",new String[] { String.valueOf(locations_viewer.getID()) });
db.close();


}
```

```
  public void deleteAll()
  {
    SQLiteDatabase db=this.getWritableDatabase();
    db.delete(TABLE_LOCATIONS, null, null);
  }
  public void delete_byID(int id)
  {
    SQLiteDatabase db=this.getWritableDatabase();
     db.delete(TABLE_LOCATIONS, KEY_ID+"="+id, null);
    }



}
```

- First_Page

```
package com.amira.googlemaps;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class First_Page extends Activity
{

   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      // TODO Auto-generated method stub
      super.onCreate(savedInstanceState);
      setContentView(R.layout.first_page);
   }
   public void okay(View v)
   {
      Intent i = new Intent(getApplicationContext(),Login_page.class);
      startActivity(i);
   }
}
```

- Friends_Page

```
package com.amira.googlemaps;

import java.util.List;

import com.parse.FindCallback;
import com.parse.ParseException;
import com.parse.ParseObject;
import com.parse.ParseQuery;
import com.parse.ParseUser;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
```

```
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

public class Friends_page extends Activity
{

   ProgressDialog myPd_ring;
   ListView friends_list;
   List<ParseObject> loc_friend;
   String lat, lon;
   String  Time;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      // TODO Auto-generated method stub
      super.onCreate(savedInstanceState);
      setContentView(R.layout.friends_page);
      friends_list = (ListView)findViewById(R.id.friends_list);
      final ArrayAdapter<String> my_adapter = new
ArrayAdapter<String>(getApplicationContext(), android.R.layout.simple_list_item_1);



      myPd_ring=ProgressDialog.show(Friends_page.this, "Please wait", "Loading please
wait..", true);
      myPd_ring.setCancelable(true);
      new Thread(new Runnable() {
         @Override
         public void run() {
            // TODO Auto-generated method stub
            try
            {
               String our_Id=ParseUser.getCurrentUser().getUsername();
               ParseQuery friends =new ParseQuery("Friends");
               friends.whereEqualTo("UserId", our_Id);
               friends.findInBackground(new FindCallback() {

                  @Override
                  public void done(List<ParseObject> arg0, ParseException arg1) {
                     my_adapter.clear();
                     // TODO Auto-generated method stub
                     for(ParseObject obj:arg0)
                     {
                        my_adapter.add(obj.getString("FriendId"));
                     }

                     my_adapter.notifyDataSetChanged();
                     friends_list.setAdapter(my_adapter);
                     myPd_ring.cancel();
                  }
               });


               Thread.sleep(5000);
            }catch(Exception e){}

            //          myPd_ring.cancel();
         }
```

```
      }).start();


      friends_list.setOnItemClickListener(new OnItemClickListener() {

         @Override
         public void onItemClick(AdapterView<?> arg0, View arg1, int position,
               long arg3) {
            // TODO Auto-generated method stub

            String friend = friends_list.getItemAtPosition(position).toString();
            ParseQuery location = ParseUser.getQuery();
            location.whereEqualTo("username", friend);
            try {
               loc_friend=location.find();
            } catch (ParseException e) {
               // TODO Auto-generated catch block
               e.printStackTrace();
            }

            for(ParseObject obj:loc_friend)
            {
               lat=obj.getString("Latitude");
               lon=obj.getString("Longitude");
               Time =  obj.getString("Time");

            }
//          DateFormat df = new SimpleDateFormat("MM/dd/yyyy HH:mm");
//          String new_Time = df.format(Time);
            Toast.makeText(getApplicationContext(), "Time ="+Time,
Toast.LENGTH_SHORT).show();

            Intent i = new Intent(Friends_page.this,G_Maps_Friends.class);
            i.putExtra("lat", lat);
            i.putExtra("lon", lon);
            i.putExtra("time", Time);
            startActivity(i);


         }
      });
   }

   @Override
   public void onBackPressed() {
      // TODO Auto-generated method stub
      //    super.onBackPressed();
       new AlertDialog.Builder(Friends_page.this)
      .setTitle("Are you sure,")
      .setMessage("Want to exit?")
      .setNegativeButton("No", new DialogInterface.OnClickListener() {

         @Override
         public void onClick(DialogInterface dialog, int which) {
            // TODO Auto-generated method stub
            dialog.dismiss();
         }
      })
      .setPositiveButton("Yes", new DialogInterface.OnClickListener() {

         @Override
         public void onClick(DialogInterface dialog_main, int which) {
            // TODO Auto-generated method stub
```

```
            dialog_main.dismiss();
            finish();
         }
      }).show();
   }
}
```

- GMaps

```
package com.amira.googlemaps;

import java.util.List;

import android.os.Build;
import android.os.Bundle;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import android.annotation.TargetApi;
import android.app.Activity;
import android.view.Menu;

@TargetApi(Build.VERSION_CODES.HONEYCOMB)
public class G_Maps extends Activity {

   private GoogleMap mMap;
   Double latitude, longitude;
   LatLng CIU;
   List<Locations_viewer> location_list;

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


      MapFragment mpFragment= ((MapFragment)
getFragmentManager().findFragmentById(R.id.map));
        mMap=mpFragment.getMap();
        mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);

      final DatabaseHandlerActivity db = new
DatabaseHandlerActivity(getApplicationContext());
      location_list = db.getAllLocations();
      db.close();

      if(location_list.isEmpty())
      {
//       Toast.makeText(getApplicationContext(), "Pre-Location",
Toast.LENGTH_SHORT).show();
//       Calendar c = Calendar.getInstance();
//       SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
//         String formattedDate = df.format(c.getTime());
```

```
         CIU = new LatLng(10.523067500000000000, 76.222210599999920000);
         MapFragment mpFragment2= ((MapFragment)
getFragmentManager().findFragmentById(R.id.map));
           mMap=mpFragment2.getMap();
           mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
           mMap.addMarker(new MarkerOptions()
           .position(CIU).title("Location").snippet("Latitude=10.52306750,
Longitude=76.22221059999992"));


      }
      else{

         for(Locations_viewer locations_viewer:location_list)
         {
            latitude = Double.parseDouble(locations_viewer.getLat());
            longitude= Double.parseDouble(locations_viewer.getLong());
            CIU = new LatLng(latitude, longitude);
             mMap.addMarker(new MarkerOptions()

.position(CIU).title("Location").snippet("Latitude="+locations_viewer.getLat()+",
Longitude="+locations_viewer.getLong()));
         }
      }


       mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(CIU, 10));
       mMap.animateCamera(CameraUpdateFactory.zoomTo(10),2000,null);
   }

   @Override
   public boolean onCreateOptionsMenu(Menu menu) {
      // Inflate the menu; this adds items to the action bar if it is present.
      getMenuInflater().inflate(R.menu.main, menu);
      return true;
   }

}
```

- GMaps Friends

```
package com.amira.googlemaps;

import android.os.Build;
import android.os.Bundle;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import android.annotation.TargetApi;
import android.app.Activity;

@TargetApi(Build.VERSION_CODES.HONEYCOMB)
public class G_Maps_Friends extends Activity {

   private GoogleMap mMap;
   Double latitude, longitude;
   String Time;
   LatLng CIU;


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

      String lat = getIntent().getStringExtra("lat");
      String lon = getIntent().getStringExtra("lon");
      Time = getIntent().getStringExtra("time");
      latitude = Double.parseDouble(lat);
      longitude = Double.parseDouble(lon);



      MapFragment mpFragment= ((MapFragment)
getFragmentManager().findFragmentById(R.id.map));
        mMap=mpFragment.getMap();
        mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
        CIU = new LatLng(latitude, longitude);
      MapFragment mpFragment2= ((MapFragment)
getFragmentManager().findFragmentById(R.id.map));
        mMap=mpFragment2.getMap();
        mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);

        mMap.addMarker(new MarkerOptions()
        .position(CIU).title("Location").snippet("Latitude="+lat+"  Longitude="+lon));



       mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(CIU, 10));
       mMap.animateCamera(CameraUpdateFactory.zoomTo(10),2000,null);
   }



}
```

- Location Viewer

```
package com.amira.googlemaps;

public class Locations_viewer
{

   int id;
   String Latitude;
   String Longitude;
   String Time;

   //empty constructor
   public Locations_viewer()
   {

   }

   public Locations_viewer(int _id,String lat,String longi,String time)
   {
      this.id=_id;
      this.Latitude=lat;
      this.Longitude=longi;
      this.Time=time;
   }

   public Locations_viewer(String lat,String longi,String time)
   {
      this.Latitude=lat;
      this.Longitude=longi;
      this.Time=time;
   }

   public int getID(){
      return this.id;
   }

   // setting id
   public void setID(int id){
      this.id = id;
   }

   // getting name
   public String getLat(){
      return this.Latitude;
   }

   // setting lat
   public void setLAt(String lat)
   {
      this.Latitude = lat;
   }

   // getting long
   public String getLong(){
      return this.Longitude;
   }

   // setting phone number
   public void setLongi(String Longi){
      this.Longitude = Longi;
```

```
    }
    // getting time
       public String getTime(){
          return this.Time;
       }

       // setting time
       public void setTime(String time){
          this.Time = time;
       }
}
```

- Login_Page

```
package com.amira.googlemaps;

import com.parse.LogInCallback;
import com.parse.ParseException;
import com.parse.ParseUser;

import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class Login_page extends Activity
{

   EditText userField, passField;
   String username, password;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);

      ParseUser currentUser = ParseUser.getCurrentUser();
      if (currentUser != null) {
         // do stuff with the user
         Intent i=new Intent(getApplicationContext(),MainPage.class);
         startActivity(i);
      }
      else
      {
         setContentView(R.layout.login_page);
         userField = (EditText)findViewById(R.id.username);
         passField = (EditText)findViewById(R.id.password);
      }


   }



   public void login(View v)
   {
```

```
    username = userField.getText().toString().trim();
    password = passField.getText().toString().trim();

    final ProgressDialog mDialog = new ProgressDialog(this);
    mDialog.setMessage("Login Progress..");

    mDialog.show();

    ParseUser.logInInBackground(username, password,
          new LogInCallback() {

       @Override
       public void done(ParseUser user, ParseException e) {
          if (user != null) {
             // Hooray! The user is logged in.

             mDialog.cancel();
             Toast toast = Toast.makeText(
                   getApplicationContext(),
                   "Successfull Login! ",
                   Toast.LENGTH_LONG);
             toast.show();

             Intent signupScreen = new Intent(getApplicationContext(),
                   MainPage.class);
             startActivity(signupScreen);



          } else {

             mDialog.cancel();
             // Signup failed. Look at the ParseException
             // to see what happened.

             AlertDialog.Builder builder = new AlertDialog.Builder(Login_page.this);
             builder.setMessage(e.getMessage())
             .setCancelable(false)

             .setNegativeButton("Ok!", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                   dialog.cancel();
                }
             });
             AlertDialog alert = builder.create();
             alert.show();

          }

       }


    });



}
public void signup(View v)
{
    Intent i = new Intent(getApplicationContext(),SignUp.class);
    startActivity(i);
```

```
    }
}
```

- Main_Page

```
package com.amira.googlemaps;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

import com.parse.ParseUser;

import android.app.TabActivity;
import android.content.Context;
import android.content.Intent;
import android.content.res.Resources;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TabHost;
import android.widget.Toast;

@SuppressWarnings("deprecation")
public class MainPage extends TabActivity
{

  LocationManager locmngr;
  Location our_location;
  Double L1 = 10.000886500000000000;  // setting an initial latitude.
  Double L2 = 76.299572799999960000;
  LocationListener loclis;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.mainpage);

    Resources re=getResources();
    TabHost th=getTabHost();
    TabHost.TabSpec spec;
    Intent i;



    locmngr=(LocationManager)getSystemService(Context.LOCATION_SERVICE);
    loclis =new MyLocationListner();
    our_location=locmngr.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
    locmngr.requestLocationUpdates( LocationManager.NETWORK_PROVIDER, 0, 0,loclis);
```

```
  i=new Intent(this,Profile.class).addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

  spec = th.newTabSpec("tab1").setIndicator("",re.getDrawable(R.drawable.users_newone)).setContent(i);
  th.addTab(spec);

  i=new Intent(this,Friends_page.class).addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

  spec = th.newTabSpec("tab2").setIndicator("",re.getDrawable(R.drawable.friend_list_new)).setContent(i);
  th.addTab(spec);

  i=new Intent(this,Notification_new.class).addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

  spec = th.newTabSpec("tab3").setIndicator("",re.getDrawable(R.drawable.notification    )).setContent(i);
  th.addTab(spec);

  th.setCurrentTab(0);

}



@Override
public boolean onCreateOptionsMenu(Menu menu)
{
  // Inflate the menu; this adds items to the action bar if it is present.
  getMenuInflater().inflate(R.menu.new_menu, menu);
  return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
  // TODO Auto-generated method stub
  switch(item.getItemId())
  {
  case R.id.logout_menu:
  {
    ParseUser.logOut();
    Toast.makeText(this, "Please Log In, to continue", Toast.LENGTH_SHORT).show();
    Intent i =new Intent(getApplicationContext(),Login_page.class);
    startActivity(i);
    MainPage.this.finish();
    break;
  }

  case R.id.history_menu:
    Intent i = new Intent(this,G_Maps.class);
    startActivity(i);
    break;

  }
```

```
  return true;
}


public class MyLocationListner implements LocationListener{

  @Override
  public void onLocationChanged(Location location)
  {
    // TODO Auto-generated method stub
    L1=location.getLatitude();
    L2=location.getLongitude();
    String currentDateTimeString = DateFormat.getDateTimeInstance().format(new Date());
    //      Toast.makeText(getApplicationContext(), currentDateTimeString, Toast.LENGTH_SHORT).show();
    DatabaseHandlerActivity db = new DatabaseHandlerActivity(getApplicationContext());
    db.addLocation(new Locations_viewer(L1.toString(), L2.toString(), currentDateTimeString));

          Toast.makeText(getApplicationContext(), "New Location   "+"Latitude ="+L1+
              "   Longitude ="+L2, Toast.LENGTH_LONG).show();


    Calendar c = Calendar.getInstance();
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
     String formattedDate = df.format(c.getTime());


    ParseUser.getCurrentUser().put("Latitude", L1.toString());
    ParseUser.getCurrentUser().put("Longitude", L2.toString());

    ParseUser.getCurrentUser().put("Time", formattedDate);


    ParseUser.getCurrentUser().saveInBackground();

    //      Toast.makeText(getApplicationContext(), "Latitude ="+L1, Toast.LENGTH_LONG).show();
    //      Toast.makeText(getApplicationContext(), "Longitude ="+L2, Toast.LENGTH_LONG).show();



  }

  @Override
  public void onProviderDisabled(String provider) {
    // TODO Auto-generated method stub

  }

  @Override
  public void onProviderEnabled(String provider) {
    // TODO Auto-generated method stub

  }

  @Override
```

```
    public void onStatusChanged(String provider, int status, Bundle extras) {
      // TODO Auto-generated method stub

    }
  }


}
```

- Notification

```
package com.amira.googlemaps;

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

import com.parse.FindCallback;
import com.parse.ParseException;
import com.parse.ParseObject;
import com.parse.ParseQuery;
import com.parse.ParseUser;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;


public class Notification_new extends Activity {

    ArrayAdapter<String> noti_adapter;
    ProgressDialog myPd_ring;
    ListView list_notification;
    TextView Noti_text;
    ArrayList<ParseObject> obj_notification = new ArrayList<ParseObject>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.notification);
        noti_adapter = new ArrayAdapter<String>(getApplicationContext(),
android.R.layout.simple_list_item_1);
        list_notification = (ListView)findViewById(R.id.list_notification);
        Noti_text = (TextView)findViewById(R.id.Noti_text);

        myPd_ring=ProgressDialog.show(Notification_new.this, "Please wait", "Loading
please wait..", true);
        myPd_ring.setCancelable(true);
        new Thread(new Runnable() {

            @Override
            public void run() {
```

```
            // TODO Auto-generated method stub

            try {
               ParseQuery myquery = new ParseQuery("Request");
               myquery.whereEqualTo("UserId",
ParseUser.getCurrentUser().getUsername());
               myquery.findInBackground(new FindCallback() {

                  @Override
                  public void done(List<ParseObject> arg0, ParseException arg1) {
                     // TODO Auto-generated method stub
                     for(ParseObject obj:arg0)
                     {
                        noti_adapter.add(obj.getString("RequestingId"));
                        obj_notification.add(obj);
                     }

                     noti_adapter.notifyDataSetChanged();

                     if(noti_adapter.isEmpty())
                     {
                        Noti_text.setText("No Notifications!");
                     }

                     list_notification.setAdapter(noti_adapter);


                     myPd_ring.dismiss();

                  }
               });

               Thread.sleep(5000);
            } catch (InterruptedException e) {
               // TODO Auto-generated catch block
               e.printStackTrace();
            }

         }

      }).start();




      list_notification.setOnItemClickListener(new OnItemClickListener() {

         @Override
         public void onItemClick(AdapterView<?> arg0, View arg1, final int position,
long arg3)
         {
            // TODO Auto-generated method stub

            AlertDialog.Builder builder = new
AlertDialog.Builder(Notification_new.this);
            builder.setMessage("Would you like to add
"+noti_adapter.getItem(position)+" as your friend?")
            .setCancelable(false)

            .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
```

```
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    // TODO Auto-generated method stub

                    String our_Name = ParseUser.getCurrentUser().getUsername();
                    ParseObject Request = new ParseObject("Friends");
                    Request.put("UserId",our_Name);
                    Request.put("FriendId", noti_adapter.getItem(position));
                    Request.saveEventually();
                    ParseObject Request_relation = new ParseObject("Friends");
                    Request_relation.put("UserId", noti_adapter.getItem(position));
                    Request_relation.put("FriendId", our_Name);
                    Request_relation.saveInBackground();
                    Toast.makeText(getApplicationContext(), "You are now friends with
"+noti_adapter.getItem(position), Toast.LENGTH_SHORT).show();


                }
            })
            .setNegativeButton("No", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    dialog.cancel();
                }

            });
            AlertDialog alert = builder.create();
            alert.show();

            obj_notification.get(position).deleteEventually();

        }
    });



    }

}
```

- Parse Aplication

```
package com.amira.googlemaps;

import com.parse.Parse;
import android.app.Application;

public class ParseApplication extends Application {
   @Override
   public void onCreate() {
      // TODO Auto-generated method stub
      super.onCreate();
      Parse.initialize(this, "cOF2YCY3vHAJeG0Y2fkVDxW4TXXcqjpkZfXCKnU5",
            "We0vjt4xAh2MSGy3Xxofq0q3CpnqXpF8QNSDpCtr");
   }

}
```

- Profile

```
package com.amira.googlemaps;

import java.util.ArrayList;
```

```
import java.util.List;

import com.parse.FindCallback;
import com.parse.ParseException;
import com.parse.ParseObject;
import com.parse.ParseQuery;
import com.parse.ParseUser;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class Profile extends Activity
{

   Boolean req_status=false;
   Boolean noti_status=false;
   String username, email;
   TextView name, mail;
   EditText search;
   ListView SearchList;
   ArrayAdapter<String> Adapter;
   // ArrayList<String> friends_Id = new ArrayList<String>();
   ProgressDialog myPd_ring;
   volatile Boolean check_delay=true;



   @Override
   protected void onCreate(Bundle savedInstanceState) {
      // TODO Auto-generated method stub
      super.onCreate(savedInstanceState);
      setContentView(R.layout.profile_page);

      search = (EditText)findViewById(R.id.search);
      SearchList=(ListView)findViewById(R.id.search_list);

      name = (TextView)findViewById(R.id.user);
      mail = (TextView)findViewById(R.id.email);

      username = ParseUser.getCurrentUser().getUsername();
      email = ParseUser.getCurrentUser().getEmail();



      name.setText(username);
      mail.setText(email);

   }
   public void search_method(View v)
   {
```

```
      Adapter=new ArrayAdapter<String>(getApplicationContext(),
android.R.layout.simple_list_item_1);
      if(Adapter.isEmpty()==false)
         Adapter.clear();
      final String search_item=search.getText().toString().trim();



      myPd_ring=ProgressDialog.show(Profile.this, "Please wait", "Loading please
wait..", true);
      myPd_ring.setCancelable(true);
      new Thread(new Runnable() {
         @Override
         public void run() {
            // TODO Auto-generated method stub
            try
            {
               ParseQuery match =ParseUser.getQuery();
               match.whereEqualTo("username", search_item);
               match.findInBackground(new FindCallback() {
                  public void done(List<ParseObject> objects, ParseException e) {
                     if (e == null) {
                        // The query was successful.
                        for(ParseObject i: objects){

                           Adapter.add(i.getString("username"));


                        }

                        set_search_list();
                     } else {
                        // Something went wrong.
                     }
                  }


               });

               Thread.sleep(5000);
            }catch(Exception e){}

            //          myPd_ring.cancel();
         }
      }).start();


   }

   public void set_search_list()
   {




      if(Adapter.isEmpty()){

         myPd_ring.cancel();
         AlertDialog.Builder builder = new AlertDialog.Builder(Profile.this);
         builder.setMessage("No Results Found")
         .setCancelable(false)
```

```
        .setNegativeButton("Ok!", new DialogInterface.OnClickListener() {
           public void onClick(DialogInterface dialog, int id) {
              dialog.cancel();
           }
        });
        AlertDialog alert = builder.create();
        alert.show();

    }

    myPd_ring.cancel();
    Adapter.notifyDataSetChanged();


    SearchList.setAdapter(Adapter);

    SearchList.setOnItemClickListener(new OnItemClickListener() {

        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1,  int position,
              long arg3) {
           // TODO Auto-generated method stub


           String our_name = ParseUser.getCurrentUser().getUsername();



           //Checking whether friend is already added
           Check_friend(position,our_name);


        }


    });
}



    public Boolean Check_notification(int position,String our_name)
    {

        ParseQuery noti_1 = new ParseQuery("Request");
        noti_1.whereEqualTo("UserId", Adapter.getItem(position));
        noti_1.whereEqualTo("RequstingId", our_name);


        ParseQuery noti_2 = new ParseQuery("Request");
        noti_2.whereEqualTo("UserId", our_name);
        noti_2.whereEqualTo("RequstingId", Adapter.getItem(position));

        List<ParseQuery> queries = new ArrayList<ParseQuery>();
        queries.add(noti_1);
        queries.add(noti_2);

        ParseQuery mainQuery = ParseQuery.or(queries);
        mainQuery.findInBackground(new FindCallback() {
           public void done(List<ParseObject> results, ParseException e) {
```

```
                noti_status=results.isEmpty();//true if empty


            }

        });


        return noti_status;
    }


    public void Check_friend(final int position, final String our_name)
    {
        final ProgressDialog obj_dlg=new ProgressDialog(Profile.this);
        obj_dlg.setMessage("Please wait");
        obj_dlg.setCancelable(true);
        obj_dlg.show();
        new Thread(new Runnable() {

            @Override
            public void run() {
                // TODO Auto-generated method stub
                ParseQuery friend_query = new ParseQuery("Friends");

                friend_query.whereEqualTo("UserId", our_name);
                friend_query.whereEqualTo("FriendId", Adapter.getItem(position));
                friend_query.findInBackground(new FindCallback() {

                    @Override
                    public void done(List<ParseObject> friends, ParseException arg1) {
                        // TODO Auto-generated method stub


                        req_status=friends.isEmpty();  //true if empty
//                      Toast.makeText(getApplicationContext(), "friends status=
"+req_status.toString(), Toast.LENGTH_LONG).show();



                        if(req_status==false)
                        {
                            obj_dlg.cancel();
                            AlertDialog.Builder build = new
AlertDialog.Builder(Profile.this);
                            build.setMessage("You are already friends with
"+Adapter.getItem(position)+"!")
                            .setCancelable(false)
                            .setPositiveButton("Ok", new DialogInterface.OnClickListener()
{

                                @Override
                                public void onClick(DialogInterface dia, int which) {
                                    // TODO Auto-generated method stub
                                    dia.dismiss();
                                }
                            }).show();
                        }

                        else{
                            obj_dlg.cancel();
```

```
                        send_request(position);
                    }


                }




            });


            }
        }).start();



    }


    public void send_request(final int position)
    {
        AlertDialog.Builder builder = new AlertDialog.Builder(Profile.this);
        builder.setMessage("Do you want to add "+Adapter.getItem(position)+" as your
friend?")
        .setCancelable(false)

        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int which) {
                // TODO Auto-generated method stub
                String our_Name = ParseUser.getCurrentUser().getUsername();
                ParseObject Request = new ParseObject("Request");


                Request.put("UserId",Adapter.getItem(position));
                Request.put("RequestingId", our_Name);
                Request.saveInBackground();
                Toast.makeText(getApplicationContext(), "Request sent",
Toast.LENGTH_SHORT).show();


            }
        })
        .setNegativeButton("No", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
            }

        });
        AlertDialog alert = builder.create();
        alert.show();
    }


    @Override
    public void onBackPressed() {
```

```
        // TODO Auto-generated method stub
        //    super.onBackPressed();
        new AlertDialog.Builder(Profile.this)
        .setTitle("Are you sure,")
        .setMessage("Want to exit?")
        .setNegativeButton("No", new DialogInterface.OnClickListener() {

           @Override
           public void onClick(DialogInterface dialog, int which) {
              // TODO Auto-generated method stub
              dialog.dismiss();
           }
        })
        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {

           @Override
           public void onClick(DialogInterface dialog_main, int which) {
              // TODO Auto-generated method stub
              dialog_main.dismiss();
              finish();
           }
        }).show();
     }
  }
```

- Sign_up

```
package com.amira.googlemaps;

import java.text.SimpleDateFormat;
import java.util.Calendar;

import com.parse.ParseException;
import com.parse.ParseUser;
import com.parse.SignUpCallback;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class SignUp extends Activity
{
   EditText usernameField, passwordField, emailField;
   String username, password, email;

   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      // TODO Auto-generated method stub
      super.onCreate(savedInstanceState);
      setContentView(R.layout.signup_page);
      usernameField = (EditText)findViewById(R.id.username);
      passwordField = (EditText)findViewById(R.id.password);
```

```
      emailField = (EditText)findViewById(R.id.email);
   }
   public void sign_up(View v)
   {
      if (usernameField.getText().length()>0) {
         if (passwordField.getText().length()>0) {
            if (emailField.getText().length()>0) {


            username = usernameField.getText().toString().trim();
            password = passwordField.getText().toString().trim();
            email = emailField.getText().toString().trim();


            ParseUser user = new ParseUser();
            user.setUsername(username);
            user.setPassword(password);
            user.setEmail(email);
            user.put("Latitude", "10.523067500000000000");
            user.put("Longitude", "76.222210599999920000");


            Calendar c = Calendar.getInstance();
            SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
              String formattedDate = df.format(c.getTime());


            user.put("Time", formattedDate);

//          user.put("SimSerialNumber", simSerialNumber);
//          user.put("UniqueNumber", uniqueNumber);

            final ProgressDialog mDialog = new ProgressDialog(
                  SignUp.this);
            mDialog.setMessage("Signing Up..");

            mDialog.show();

            user.signUpInBackground(new SignUpCallback() {
               public void done(ParseException e) {

                  mDialog.cancel();

                  if (e == null) {
                     Toast toast = Toast.makeText(
                           getApplicationContext(),
                           "Successful signUp! ",
                           Toast.LENGTH_LONG);
                     toast.show();
                     Intent signupScreen = new Intent(
                           getApplicationContext(),
                           Login_page.class);
                     startActivity(signupScreen);
                     // Hooray! Let them use the app now.

                  } else {

                     AlertDialog.Builder builder = new AlertDialog.Builder(
                           SignUp.this);
                     builder.setMessage(e.getMessage())
                           .setCancelable(false)
```

```
                        .setNegativeButton(
                              "Ok!",
                              new DialogInterface.OnClickListener() {
                                 public void onClick(
                                       DialogInterface dialog,
                                       int id) {
                                    dialog.cancel();
                                 }
                              });
                        AlertDialog alert = builder.create();
                        alert.show();

                        // Sign up didn't succeed. Look at the
                        // ParseException
                        // to figure out what went wrong

                     }
                  }
               });
            }
               else
               {
                  Toast toast = Toast.makeText(getApplicationContext(),
                        "Please enter an email! ", Toast.LENGTH_LONG);
                  toast.show();
               }

            }
               else {
               Toast toast = Toast.makeText(getApplicationContext(),
                     "Please enter a password! ", Toast.LENGTH_LONG);
               toast.show();

            }

         } else {
            Toast toast = Toast.makeText(getApplicationContext(),
                  "Please enter a valid username! ",
                  Toast.LENGTH_LONG);
            toast.show();
         }


      }
}
```

Refrences:

[1] Mahmood, F.M. ; Bin Abdul Salam, Z.A "A conceptual framework for personalized location-based Services (LBS) tourism mobile application leveraging semantic web to enhance tourism experience," Advance Computing Conference (IACC), 2013 IEEE 3rd International Page(s):287 - 291 Print ISBN: 978-1-4673-4527-9
[2] Federal Communication Commission 911 services website, accessed August 24, 2006, http://www.fcc.gov/911/enhanced/
[3] The Parlay Group, http://www.parlay.org
[4] The Open Mobile Alliance (OMA), "The Mobile Location Protocol," http://www.openmobilealliance.org
[5] Sean J. Barbeau, Miguel A. Labrador, Philip L. Winters, Rafael Pérez, and Nevine Labib Georggi, A General Architecture in Support of Interactive, Multimedia, Location-Based Mobile Applications, IEEE Communications Magazine • November 2006
[6] J. Hightower and G. Borriello, "A Survey and Taxonomy of Location Systems for Ubiquitous Computing," IEEE Computer, vol. 34, no. 8, 2001, pp. 57–66.
[7] A. Kupper, Location-Based Services: Fundamentals and Operation, New York: Wiley, 2005.
[8] H. Karl and A. Willig, Protocols and Architectures for Wireless Sensor Networks, John Wiley, 2005.
[9] Sun Microsystems, "Java Remote Method Invocation Specification," 1998.
[10] M. Spanoudakis et al., "Extensible Platform for Location Based Services Provisioning," Proc. 4th Int'l. Conference Web Information Systems Engineering Workshop. (WISEW), 2004.
[11] A. Krevl, M. Ciglaric, "A framework for developing distributed location based applications", IPDPS, 2006, Parallel and Distributed Processing Symposium, International, Parallel and Distributed Processing Symposium, International 2006, pp. 294, doi:10.1109/IPDPS.2006.1639551
[12] Ch.ChakradharaRao et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (6) , 2013, 979-982 "GPS Based Vehicle Navigation System Using Google Maps".
[13] https://parse.com/docs/android/guide
[14] http://www.tutorialspoint.com/android/