

MOBILE COMPUTING PROJECT – SAFETY APPLICATION FOR EMERGENCY
JAVA CODE

DatabaseHandler.java :

```
package com.example.safety;

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

public class DatabaseHandler extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "mylist.db" ;
    public static final String TABLE_NAME = "mylist_data";
    public static final String COL1 = "ID";
    public static final String COL2 = "ITEM1";

    public DatabaseHandler(Context context) {super(context,DATABASE_NAME,null,1);}
    @Override
    public void onCreate(SQLiteDatabase db){

        String createTable = "CREATE TABLE "+TABLE_NAME+"(ID INTEGER PRIMARY KEY AUTOINCREMENT, "+"ITEM1
TEXT)";
        db.execSQL( createTable );

    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){

        String a = "DROP TABLE IF EXISTS " +TABLE_NAME;
        db.execSQL(a);
        onCreate( db );

    }
    public boolean addData (String item1){
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues( );
        contentValues.put( COL2,item1 );

        long result = db.insert( TABLE_NAME,null,contentValues );

        if(result==-1){
            return false;
        }
        else{
            return true;
        }
    }
    public Cursor getListContents(){
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor data = db.rawQuery("SELECT * FROM "+TABLE_NAME,null);
        return data;
    }
}
```

MainActivity.java :

```
package com.example.safety;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        new Handler( ).postDelayed( new Runnable(){
            @Override
            public void run(){
                if(MainActivity.this==null){
                    return;
                }
                Intent intent = new Intent( getApplicationContext(), MainActivity2.class);
                startActivity(intent);
                finish();
            }
        },3000);
    }
}
```

MainActivity2.java :

```
package com.example.safety;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.DialogInterface;
import android.content.pm.PackageManager;
import android.content.pm.ProviderInfo;
import android.database.Cursor;
import android.location.Criteria;
import android.location.Location;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.provider.Settings;
import android.view.View;
import android.widget.Button;
import android.content.Intent;
import android.location.LocationManager;
import android.widget.ImageView;
```

```

import android.widget.Toast;
//import android.permission.CALL_PHONE;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.location.LocationSettingsRequest;
import com.google.android.gms.tasks.OnSuccessListener;

import java.security.Provider;
import java.util.ArrayList;

public class MainActivity2 extends AppCompatActivity {

    Button b1, b2;
    ImageView imageView;
    private FusedLocationProviderClient fusedLocationProviderClient;
    DatabaseHandler myDB;
    private final int REQUEST_CHECK_CODE = 8989;
    private LocationSettingsRequest.Builder builder;
    String x = "", y = "";
    private static final int REQUEST_LOCATION = 1;

    LocationManager locationManager;
    Intent mIntent;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        b1 = findViewById(R.id.button);
        b2 = findViewById(R.id.button2);
        imageView = findViewById(R.id.chat);

        fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this);

        myDB = new DatabaseHandler(this);
        final MediaPlayer mp = MediaPlayer.create(getApplicationContext(), R.raw.emergency_alarm);

        locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
        if (!locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
            onGPS();
        } else {
            startTrack();
        }
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(getApplicationContext(), Register.class);
                startActivity(i);
            }
        });

        b2.setOnLongClickListener(new View.OnLongClickListener() {
            @Override
            public boolean onLongClick(View view) {

```

```

        mp.start();
        Toast.makeText(getApplicationContext(), "PANIC BUTTON STARTED", Toast.LENGTH_SHORT).show();
        return false;
    }
});

b2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        loadData();
    }
});

imageView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(), MainActivity3.class);
        startActivity(intent);
    }
});
}

```

```

private void loadData() {
    ArrayList<String> thelist = new ArrayList<>();
    Cursor data = myDB.getListContents();
    if (data.getCount() == 0) {
        Toast.makeText(this, "no content to show.", Toast.LENGTH_SHORT).show();
    } else {
        String msg = "I NEED HELP.MY LOCATION IS LATITUDE:" + x + " AND LONGITUDE:" + y + "\n ";

        msg += "http://maps.google.com/?q=" + x + "," + y;
        String number = "";
        while (data.moveToNext()) {
            thelist.add(data.getString(1));
            number = number + data.getString(1) + (data.isLast() ? "" : ";");
            call();
        }
        if (!thelist.isEmpty()) {
            sendSms(number, msg, true);
        }
    }
}

```

```

private void sendSms(String number, String msg, boolean b) {
    Uri uri = Uri.parse("smsto:" + number);
    Intent smsIntent = new Intent(Intent.ACTION_SENDTO, uri);

    smsIntent.putExtra("sms_body", msg);
    //smsIntent.setType("vnd.android-dir/mms-sms");
    startActivity(smsIntent);
}

```

```

private void call() {
    Intent i = new Intent(Intent.ACTION_CALL);
}

```

```

i.setData(Uri.parse("tel:9842928762"));

if (ContextCompat.checkSelfPermission(getApplicationContext(), Manifest.permission.CALL_PHONE) ==
PackageManager.PERMISSION_GRANTED) {
    startActivity(i);
} else {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        requestPermissions(new String[]{Manifest.permission.CALL_PHONE}, 1);
    }
}
}

private void startTrack() {

    if (ActivityCompat.checkSelfPermission(MainActivity2.this, Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(MainActivity2.this,
            Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {

        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
REQUEST_LOCATION);

    } else {
        fusedLocationProviderClient.getLastLocation().addOnSuccessListener(this, new
OnSuccessListener<Location>() {
            @Override
            public void onSuccess(Location location) {
                if (location != null) {
                    double lat = location.getLatitude();
                    double lon = location.getLongitude();
                    x = String.valueOf(lat);
                    y = String.valueOf(lon);
                }
            }
        });
        /*Location locationGPS = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        if(locationGPS!=null){
            double lat = locationGPS.getLatitude();
            double lon = locationGPS.getLongitude();
            x = String.valueOf(lat);
            y = String.valueOf( lon );
        }*/

    }

}

private void onGPS(){
    final AlertDialog.Builder builder = new AlertDialog.Builder( this );
    builder.setMessage( "Enable GPS" ).setCancelable( false ).setPositiveButton("yes", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            startActivity( new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS ) );
        }
    }).setNegativeButton("NO", new DialogInterface.OnClickListener() {
        @Override

```

```

        public void onClick(DialogInterface dialogInterface, int i) {
            dialogInterface.cancel();
        }
    });
    final AlertDialog alertDialog = builder.create();
    alertDialog.show();
}
}

```

MainActivity3.java :

```

package com.example.safety;
import org.alicebot.ab.*;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.content.res.AssetManager;
import android.os.Bundle;
import android.os.Environment;
import android.text.TextUtils;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.Toast;
import java.util.*;
import java.io.*;

import com.example.safety.Adapter.ChatMessageAdapter;
import com.example.safety.Model.ChatMessage;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.karumi.dexter.Dexter;
import com.karumi.dexter.MultiplePermissionsReport;
import com.karumi.dexter.PermissionToken;
import com.karumi.dexter.listener.DexterError;
import com.karumi.dexter.listener.PermissionRequest;
import com.karumi.dexter.listener.PermissionRequestErrorListener;
import com.karumi.dexter.listener.multi.MultiplePermissionsListener;

import java.util.ArrayList;

public class MainActivity3 extends AppCompatActivity {

    ListView listView;
    FloatingActionButton btnSend;
    EditText edtTextMsg;
    ImageView imageView;

    public Bot bot;
    public static Chat chat;
    private ChatMessageAdapter adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

setContentView(R.layout.activity_main3);

listView = findViewById(R.id.listView);
btnSend = findViewById(R.id.btnSend);
edtTextMsg = findViewById(R.id.editTextMsg);
imageView = findViewById(R.id.imageView);

adapter = new ChatMessageAdapter(this, new ArrayList<ChatMessage>());
listView.setAdapter(adapter);

btnSend.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String message = edtTextMsg.getText().toString();
        String response = chat.multisentenceRespond(message);
        System.out.println(message);
        System.out.println(response);
        if(TextUtils.isEmpty(message)){
            Toast.makeText(MainActivity3.this,"Please enter a query..",Toast.LENGTH_SHORT).show();
            return;
        }
        sendMessage(message);
        botsReply(response);
        edtTextMsg.setText("");
        listView.setSelection(adapter.getCount() - 1 );
    }
});

Dexter.withActivity(this)
    .withPermissions(
        Manifest.permission.WRITE_EXTERNAL_STORAGE,
        Manifest.permission.READ_EXTERNAL_STORAGE
    ).withListener(new MultiplePermissionsListener(){
        @Override
        public void onPermissionsChecked(MultiplePermissionsReport report){
            if(report.areAllPermissionsGranted()){
                custom();
                Toast.makeText(MainActivity3.this,"Permission granted",Toast.LENGTH_SHORT).show();
            }
            if(report.isAnyPermissionPermanentlyDenied()){
                Toast.makeText(MainActivity3.this,"Please grant all permission",Toast.LENGTH_SHORT).show();
            }
        }
        @Override
        public void onPermissionRationaleShouldBeShown(List<PermissionRequest> permissions,
PermissionToken token){
            token.continuePermissionRequest();
        }
    }).withErrorListener(new PermissionRequestErrorListener() {
        @Override
        public void onError(DexterError error) {
            Toast.makeText(MainActivity3.this,""+error,Toast.LENGTH_SHORT).show();
        }
    }).onSameThread().check();
}

private void botsReply(String response) {

```

```

    ChatMessage chatMessage = new ChatMessage(response,false,false);
    adapter.add(chatMessage);
}

private void sendMessage(String message) {
    ChatMessage chatMessage = new ChatMessage(message,true,false);
    adapter.add(chatMessage);
}

private void custom(){
    boolean available = isSDCARDAvailable();

    AssetManager assets = getResources().getAssets();
    File filename = new File(Environment.getExternalStorageDirectory().toString()+"/TBC/bots/bot");

    boolean makeFile = filename.mkdirs();
    if(filename.exists()){
        try{
            for(String dir : assets.list("bot")){
                File subDir = new File(filename.getPath()+"/"+dir);
                boolean subDir_check = subDir.mkdirs();

                for(String file : assets.list("bot/"+dir)){
                    File newfile = new File(filename.getPath()+"/"+dir+"/"+file);
                    if(newfile.exists()){
                        continue;
                    }
                    InputStream in;
                    OutputStream out;

                    in = assets.open("bot/"+dir+"/"+file);
                    out = new FileOutputStream(filename.getPath()+"/"+dir+"/"+file);

                    copyFile(in,out);
                    in.close();
                    out.flush();
                    out.close();
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    MagicStrings.root_path = Environment.getExternalStorageDirectory().toString()+"/TBC";
    AIMLProcessor.extension = new PCAIMLProcessorExtension();

    bot = new Bot("bot",MagicStrings.root_path,"chat");
    chat = new Chat(bot);
}

private void copyFile(InputStream in, OutputStream out) throws IOException{
    byte [] buffer = new byte[1024];
    int read;
    while((read = in.read(buffer))!= -1){
        out.write(buffer,0,read);
    }
}

```



```

    }

    private boolean isSDCARDAvailable() {
        return Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED)? true : false;
    }
}

```

Register.java :

```

package com.example.safety;

import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;

public class Register extends AppCompatActivity {
    Button b1;
    Button b2,b3;
    EditText e1;
    ListView listView;
    SQLiteOpenHelper sl;
    SQLiteDatabase sqldb;
    DatabaseHandler myDB;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        e1 = findViewById( R.id.phone );
        b1 = findViewById( R.id.add );
        listView = findViewById(R.id.list);
        b2 = findViewById( R.id.delete );
        b3 = findViewById( R.id.view );

        myDB = new DatabaseHandler(this );

        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String sr = e1.getText().toString();
                addData(sr);
                Toast.makeText(Register.this,"Data added",Toast.LENGTH_SHORT).show();
                e1.setText("");
            }
        });
    }
}

```

```

b2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        sqllitedb = myDB.getWritableDatabase();
        String x = e1.getText().toString();
        DeleteData(x);
        Toast.makeText(Register.this,"Data deleted",Toast.LENGTH_SHORT).show();
    }
});

b3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        loadData();
    }
});

}
private void loadData(){
    ArrayList<String> theList = new ArrayList<>( );
    Cursor data = myDB.getListContents();
    if(data.getCount()==0){
        Toast.makeText( Register.this,"There is no content",Toast.LENGTH_SHORT).show();

    }
    else{
        while(data.moveToNext()){
            theList.add( data.getString(1) );
            ListAdapter listAdapter = new ArrayAdapter<>(this,android.R.layout.simple_list_item_1,theList );
            listView.setAdapter( listAdapter );
        }
    }
}
private void addData(String newEntry ){
    boolean insertData = myDB.addData( newEntry );
    if(insertData==true){
        Toast.makeText(Register.this,"Data added",Toast.LENGTH_SHORT).show();
    }
    else{
        Toast.makeText(Register.this,"Unsuccessful",Toast.LENGTH_SHORT).show();
    }
}
private boolean DeleteData(String x){
    return sqllitedb.delete(DatabaseHandler.TABLE_NAME, DatabaseHandler.COL2+ "=?", new String[]{x})>0;
}
}

```

Adapter

ChatMessageAdapter.java :

```
package com.example.safety.Adapter;
```

```
import android.content.Context;
import android.view.LayoutInflater;
```

```

import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;
import android.widget.Toast;

import java.util.List;

import com.example.safety.Model.ChatMessage;
import com.example.safety.R;

public class ChatMessageAdapter extends ArrayAdapter<ChatMessage> {

    private static final int MY_MESSAGE = 0, OTHER_MESSAGE = 1;

    public ChatMessageAdapter(Context context, List<ChatMessage> data){
        super(context, R.layout.users_message,data);
    }

    @Override
    public int getViewTypeCount(){
        return 2;
    }

    @Override
    public int getItemViewType(int position){
        ChatMessage item = getItem(position);

        if(item.isMine() && !item.isImage()){
            return MY_MESSAGE;
        }
        else {
            return OTHER_MESSAGE;
        }
    }

    @Override
    public View getView(int position, View itemView, ViewGroup parent){
        int viewType = getItemViewType(position);

        if(viewType == MY_MESSAGE){
            itemView = LayoutInflater.from(getContext()).inflate(R.layout.users_message,parent,false);
            TextView textView = itemView.findViewById(R.id.text);
            textView.setText(getItem(position).getContent());
        }
        else if(viewType == OTHER_MESSAGE){
            itemView = LayoutInflater.from(getContext()).inflate(R.layout.bot_message,parent,false);
            TextView textView = itemView.findViewById(R.id.text);
            textView.setText(getItem(position).getContent());
        }

        itemView.findViewById(R.id.chatMessageView).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(getContext(),"Clicked",Toast.LENGTH_LONG).show();
            }
        });
    }
}

```

```
        return itemView;
    }
}
```

ChatMessage.java :

```
package com.example.safety.Model;
```

```
public class ChatMessage {
    private boolean isImage, isMine;
    private String content;

    public ChatMessage(String message, boolean mine, boolean image){
        content = message;
        isMine = mine;
        isImage = image;
    }

    public String getContent(){
        return content;
    }

    public void setContent(String content){
        this.content = content;
    }

    public boolean isImage() {
        return isImage;
    }

    public boolean isMine() {
        return isMine;
    }

    public void setIsMine(boolean mine) {
        isMine = mine;
    }

    public void setIsImage(boolean image) {
        isImage = image;
    }
}
```