

**'Do Not Disturb Incoming Call Manager'**

*A project report submitted in partial fulfillment of the requirements for the degree  
Of*

**Bachelor of Computer Applications (BCA)  
OF  
North Eastern Hill University (NEHU)  
2017**



*Submitted by*

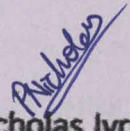
Name : AIBOKLANG DKHAR  
Roll No. : P1500061  
Reg. No. : 10461 of 2014-15



**DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS  
SHILLONG COLLEGE  
SHILLONG 793003  
MEGHALAYA.**

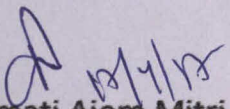
# CERTIFICATE

This is to certify that the Project work titled '*Do Not Disturb Incoming Call Manager*' is a bonafide work done by *AIBOKLANG DKHAR*, University Roll No.: *P1500061*, Registration No.: *10461 of 2014-15*, under my guidance during the final year of the course.

  
Sir. Nicholas Jyrwa

Internal Guide

Project seminar was held on *17.04.2017* at Shillong College, Shillong.

  
Shrimati Aiom Mitri

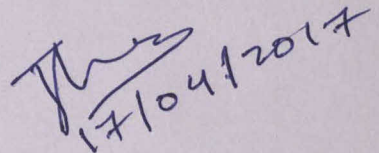
HOD

Dept. Of Comp. Sci.&App.

Shillong College

Shillong, Meghalaya.

793003

  
17/04/2017  
External examiner

## Acknowledgement

I would like to take the opportunity to express my deepest thanks to my guide Sir. Nicholas Jyrwa for helping me and encouraging me in doing this project.

I would also like to thank the staff of the Department of Computer Science for helping me in every way they can.

Finally, I would like to thank the students of our Department, especially Shawn L. Khongwir and Macdonald Kharsahnoh for their help, for their friendship and for encouraging me to do this project.

Thanking You,  
AIBOKLANG DKHAR



## Table of Contents

1.	Introduction.	Page()
2.	Synopsis.	Page()
3.	System Existing Applications/Proposed Application.	Page()
4.	User Requirements Hardware/Software Specifications.	Page()
5.	DFD Diagram Level 0/Level 1	Page()
6.	Flowchart.	Page()
7.	Feature Enhancements	Page()
8.	Conclusion.	Page()
9.	Bibliography	Page()
10.	Source Code.	Page()
11.	Sample Input / Output Screens.	Page()

## 1. Introduction

It is an application for mobile phones using the operating system Android whose function is to automatically Reject calls to contacts that has previously been defined as blocked in a time frame, both times as daily indicating that you are sleeping (or busy).

## 2. Synopsis

**Project Title:** 'Do Not Disturb Incoming Call Manager'

**Project Technology:** Android

**Scopes of Project:**

It allows user to define which calls to be blocked, allows editing and deleting blocked contact list.

It Blocks and Rejects Calls Automatically to the numbers that have been defined in the Blocked List.





**Preliminary Investigation**

The idea about the project is got from the requirement for not to be disturbed by a call from others at some specific time. The application is for android phone, so the development to be done using android technology. In order to block (reject) the incoming call, we have to get the number and check that whether it falls in Blocking criteria, if it is then it will automatically Block the incoming call.

**Selecting the project development Strategy**

Since the technology used is android for this application and android uses java as base language for programs I have selected the Object Oriented System Development (OOSD) as the development strategy. Again the entities involved in this application can be considered as Objects and we can also store those Objects in the database. So I have used the Database Engine SQLite.

### 3. System Study

	Proposed Application		Existing Applications	
	Do Not Disturb Incoming Call Manager	Call Blocker	SIP Trunk Call Manager	Do Not Disturb
ICONS				
OS	Android	Android	Android	Android
INTERNET	Not Required	Not Required	Not Required	Not Required
SUBSCRIPTION	Not Required	Not Required	Required	Not Required
SIZE	1.81MB	1.64MB	0.88MB	731KB
RAM	85MB(max)	147MB(max)	109MB(max)	112MB(max)
ANDROID VERSION	4.1 Above(Jelly Bean)above	4.4(Kitkat)above	4.4(Kitkat)above	5.0(Lollipop)
REJECT CALLS	YES	NO	NO	NO
FEATURES	Reject Calls Automatically	Silent phone state	Forwarded calls to another number	Silent Call Notification

### 4. User's Requirements:

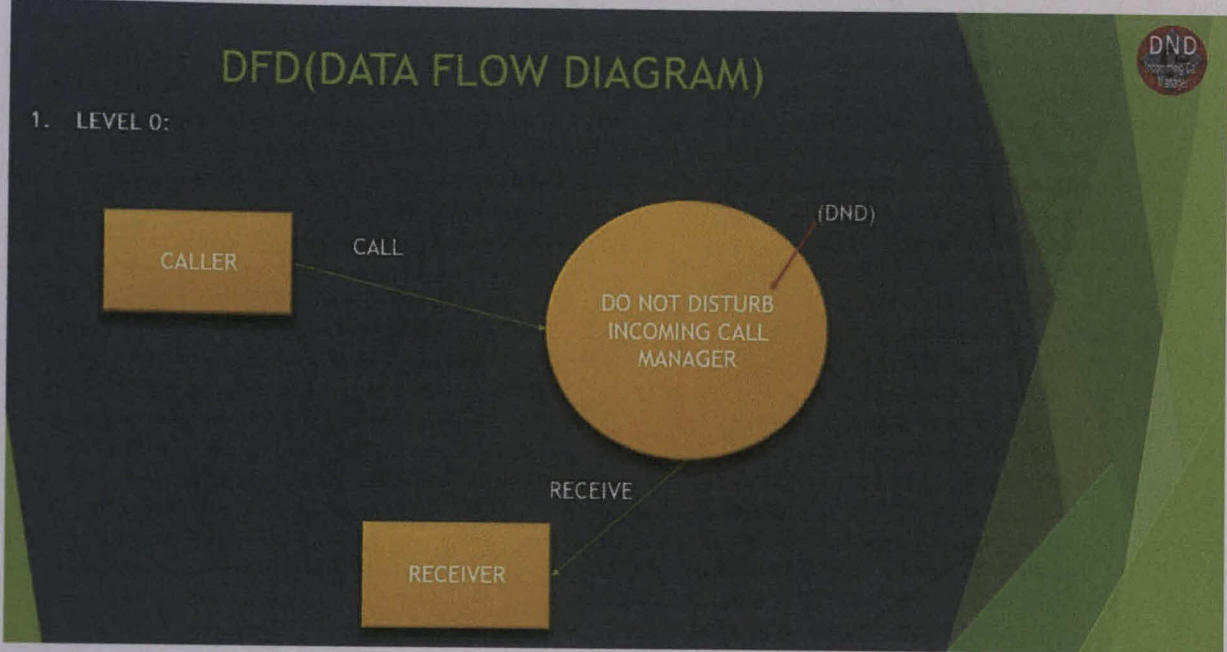
Hardware Requirements: Android Mobile Phones

Software Requirements:

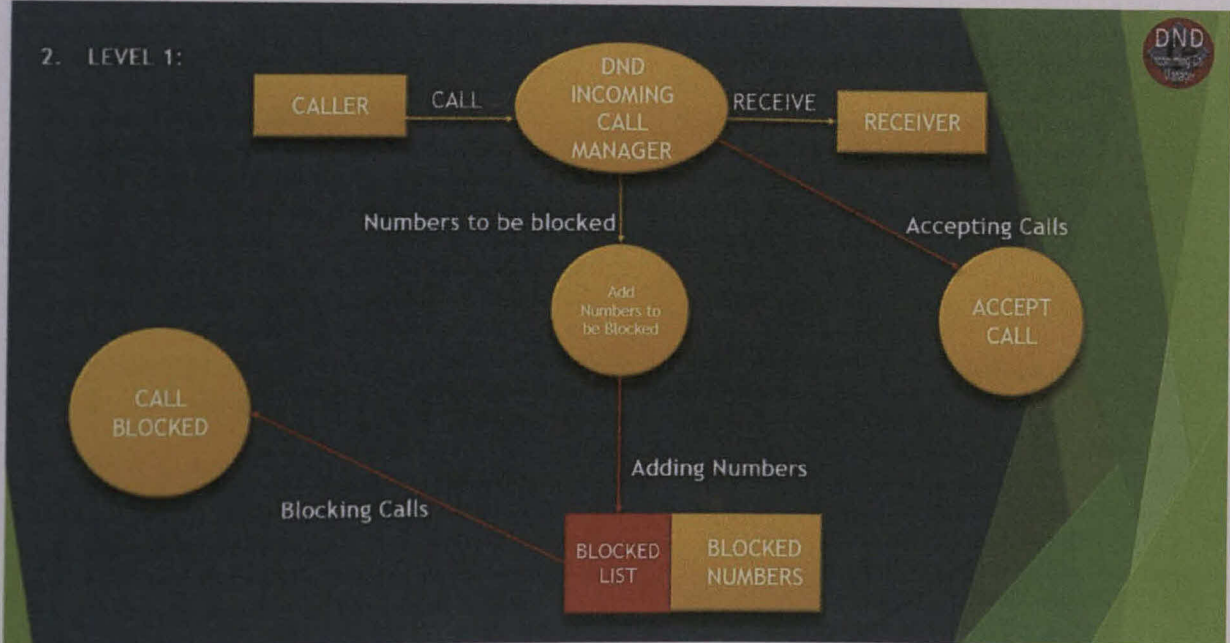
1. OS – Android 4.1 above (Jelly Bean).
2. RAM – 215MB minimum
3. Storage – 12.11MB
4. System Requirements – Single Core above

5. DFD Diagram:

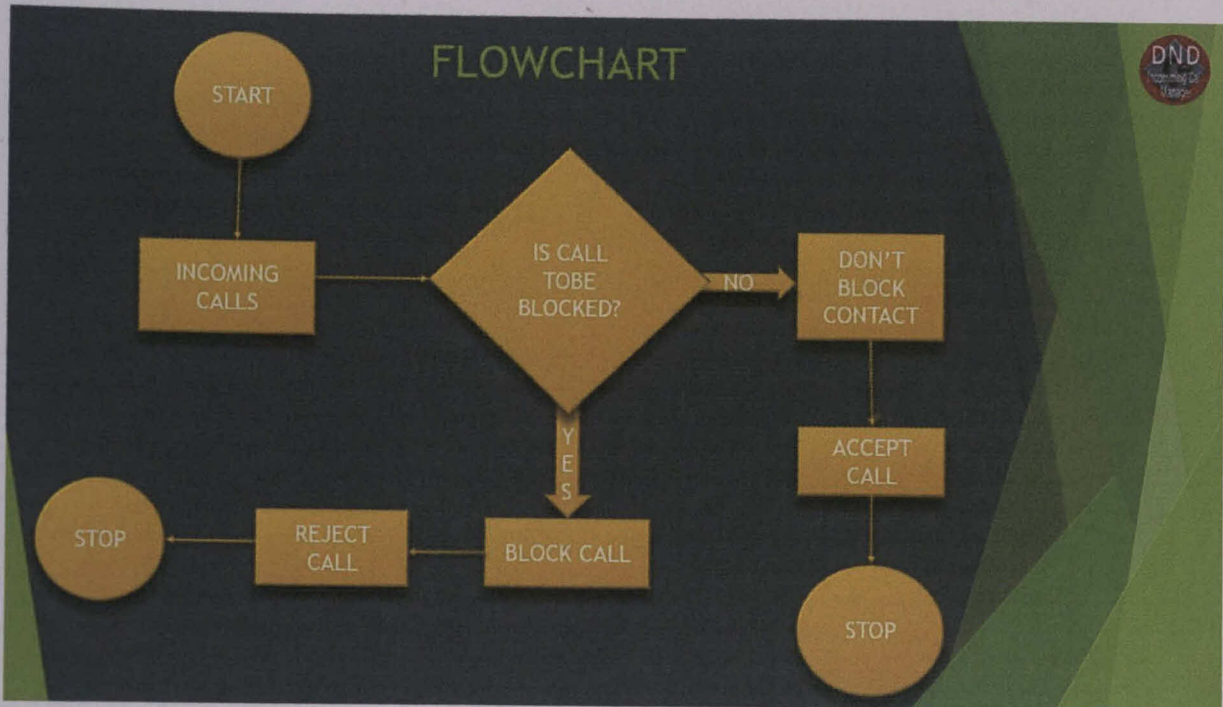
(i) Level 0:



(ii) Level 1:



## 6. Flowchart:



## 7. Feature Enhancements:

1. Automatically Reject Calls.
2. User Friendly.
3. Less Memory Required.
4. Anyone can use without Subscription.

**8. Conclusion:** The Application will allow user to be able to keep privacy between them and the people whom they usually gets disturbed often.



## 9. Bibliography:

1. <https://developer.android.com/training/index.html>
2. <http://stackoverflow.com/questions/15297840/populate-listview-from-arraylist-of-objects>
3. <http://stackoverflow.com/questions/9904426/how-to-block-a-mobile-number-call-and-message-receiving-in-android-application-d>
4. <https://developer.android.com/reference/android/telephony/TelephonyManager.html>
5. [https://www.tutorialspoint.com/android/android\\_sqlite\\_database.htm](https://www.tutorialspoint.com/android/android_sqlite_database.htm)
6. <https://www.google.co.in/amp/mobilesiri.com/android-sqlite-database-tutorial-using-android-studio/amp/>

10. Source Code:

# **XML FILES**

## ACTIVITY\_MAIN.XML

### CODES:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="@drawable/p4"
    android:weightSum="1">

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:text="BLOCKED LIST"
        android:id="@+id/textView"
        android:textColor="#ffffff" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:id="@+id/ImageView"
        android:layout_gravity="center_horizontal"
        android:background="@drawable/aibok"/>

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="350dp"
        android:orientation="vertical"
        android:padding="10dp" >

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <ListView
                android:id="@+id/listview"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

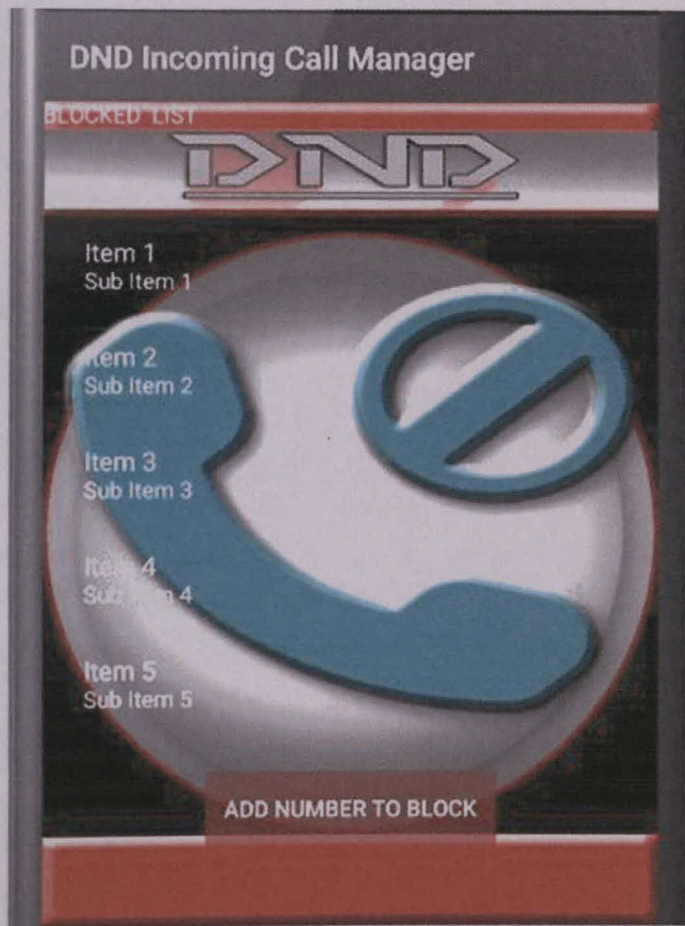
        </TableRow>

    </TableLayout>
```

```
<Button
  android:id="@+id/add_blacklist_btn"
  android:layout_width="180dp"
  android:layout_height="wrap_content"
  android:text="Add Number To Block"
  android:background="#b45b1010"
  android:layout_column="18"
  android:layout_gravity="center_horizontal"
  android:textColor="#ffffff" />
```

```
</LinearLayout>
```

**DESIGN:**



## ACTIVITY\_ADD\_TO\_BLOCKLIST.XML

### CODES:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="@drawable/p4"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="top|center_horizontal"
        android:text="Add Phone Number"
        android:textSize="22sp"
        android:textColor="#ffffff" />

    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="left"
        android:orientation="vertical"
        android:padding="10dip" >

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            </TableRow>

            <TableRow
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:backgroundTint="#ffffff">

            </TableRow>

            <TableRow
                android:layout_width="match_parent"
                android:layout_height="match_parent">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
```



```
android:layout_marginTop="10dip"  
android:text="Country code "  
android:textSize="15sp"  
android:textColor="#ffffff"  
android:layout_column="0" />
```

```
<EditText
```

```
android:layout_width="fill_parent"  
android:layout_height="wrap_content"  
android:layout_marginLeft="0dip"  
android:layout_weight="1"  
android:id="@+id/country_code_et"  
android:inputType="number"  
android:layout_column="1"  
android:layout_span="5"  
android:textColor="#ffffff" />
```

```
</TableRow>
```

```
<TableRow
```

```
android:layout_width="match_parent"  
android:layout_height="wrap_content" >
```

```
<TextView
```

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginTop="10dip"  
android:text="Phone Number "  
android:textSize="15sp"  
android:textColor="#ffffff" />
```

```
<EditText
```

```
android:layout_width="fill_parent"  
android:layout_height="wrap_content"  
android:layout_marginLeft="0dip"  
android:layout_weight="1"  
android:id="@+id/phone_et"  
android:inputType="number"  
android:textColor="#ffffff"  
android:layout_span="5" />
```

```
</TableRow>
```

```
</TableLayout>
```

```
<RelativeLayout
```

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:orientation="horizontal"  
android:padding="20dip"  
>
```

```
<Button
```

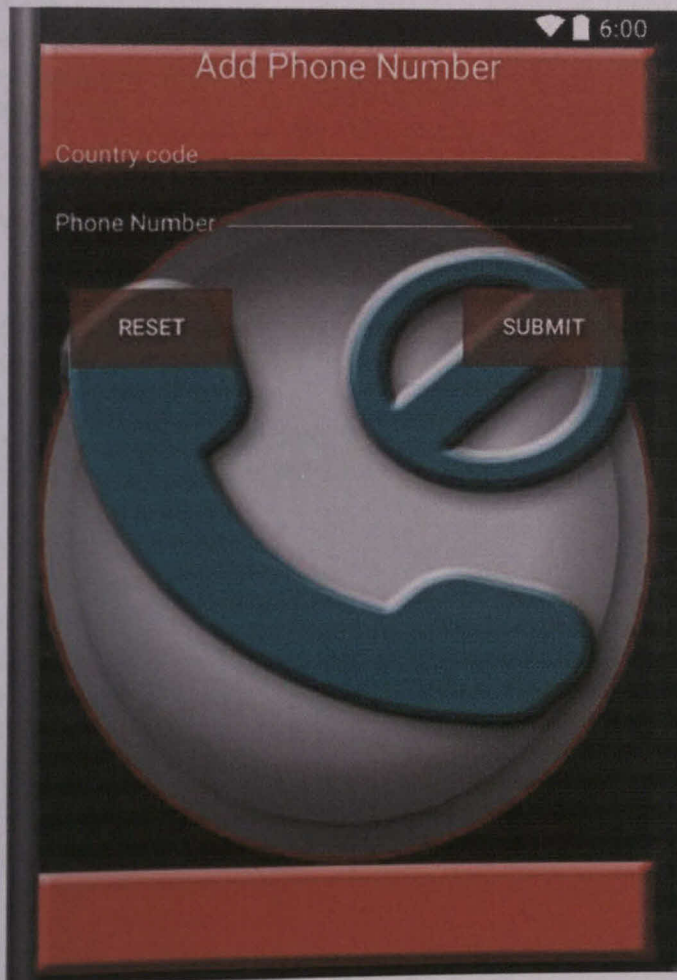
```
android:id="@+id/reset_btn"  
android:layout_width="100dp"
```

```
android:layout_height="wrap_content"  
android:layout_gravity="top"  
android:text="Reset"  
android:background="#b45b1010"  
android:textColor="#ffffff" />
```

```
<Button  
  android:id="@+id/submit_btn"  
  android:layout_width="100dp"  
  android:layout_height="wrap_content"  
  android:layout_gravity="left"  
  android:text="Submit"  
  android:background="#b45b1010"  
  android:textColor="#ffffff"  
  android:layout_alignTop="@+id/reset_btn"  
  android:layout_alignParentRight="true"  
  android:layout_alignParentEnd="true" />  
</RelativeLayout>
```

```
</LinearLayout>
```

**DESIGN:**



## LIST\_ITEM.XML

## CODES:

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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:orientation="horizontal">

    <TableRow
        android:layout_width="fill_parent"
        android:background="#6f6c67"
        android:layout_gravity="center"
        android:layout_height="wrap_content" >

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:layout_marginTop="10dip"
            android:text="Sl. No(s)"
            android:id="@+id/serial_tv"
            android:textColor="#980000"
            android:background="#e7e5d9"
            android:layout_margin="1dip"
            android:padding="3dip"
            android:gravity="center"
            android:layout_weight="2"
            android:textSize="15sp" />

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:layout_marginTop="10dip"
            android:text="Blocked Phone Number() "
            android:id="@+id/phone_number_tv"
            android:textColor="#980000"
            android:background="#e7e5d9"
            android:layout_margin="1dip"
            android:padding="3dip"
            android:gravity="center"
            android:layout_weight="1"
            android:textSize="15sp" />

    </TableRow>

</LinearLayout>
```



**DESIGN:**

Sl. No(s)	Blocked Phone Number()
[Redacted Content]	

## ANDROIDMANIFEST.XML

### CODES:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.callmgr.dnd.dndcallmanager">

    <uses-permission android:name="android.permission.CALL_PHONE" />

    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />

    <uses-permission android:name="android.permission.READ_PHONE_STATE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="DND Incoming Call Manager"
        android:theme="@style/Theme.AppCompat.DayNight.NoActionBar" >
        <activity
            android:name=".MainActivity"
            android:label="DND Incoming Call Manager" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <receiver android:name=".CallBarring">
            <intent-filter android:priority="100" >
                <action android:name="android.intent.action.PHONE_STATE" />
            </intent-filter>
        </receiver>

        <activity
            android:name=".AddToBlocklistActivity"
            android:label="DND Incoming Call Manager"
            />
    </application>

    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />

</manifest>
```

**JAVA**

**FILES**

## MAIN ACTIVITY.JAVA



## CODES:

```
package com.callmgr.dnd.dndcallmanager;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;

import java.util.List;

public class MainActivity extends AppCompatActivity implements
OnClickListener, OnItemLongClickListener

{
    private Button add_blacklist_btn;
    public ListView listview;

    private BlacklistDAO blacklistDao;

    public static List<Blacklist> blockList;

    private int selectedRecordPosition = -1;

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

        add_blacklist_btn = (Button) findViewById(R.id.add_blacklist_btn);
        add_blacklist_btn.setOnClickListener(this);

        listview = (ListView) findViewById(R.id.listview);

        final LayoutInflater inflater = (LayoutInflater)
this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
```

```

        final View rowView = inflater.inflate(R.layout.list_item,
listview, false);

        listview.addHeaderView(rowView);

        listview.setOnItemLongClickListener(this);
    }

    private void populateNoRecordMsg()
    {

        if(blockList.size() == 0)
        {
            final TextView tv = new TextView(this);
            tv.setPadding(5, 5, 5, 5);
            tv.setTextSize(15);
            tv.setText("No Record Found !!");
            listview.addFooterView(tv);
        }
    }

    @Override
    public void onClick(View v)
    {
        if (v == add_blacklist_btn)
        {
            startActivity(new Intent(MainActivity.this,
AddToBlocklistActivity.class));
        }
    }

    @Override
    public boolean onItemLongClick(AdapterView<?> parent, View view,
int position, long id)

    {
        if (position > 0)
        {
            selectedRecordPosition = position - 1;
            showDialog();
        }
        return true;
    }

    @Override
    protected void onResume()
    {
        super.onResume();

        blacklistDao = new BlacklistDAO(this);
    }

```

```

        blockList = blackListDao.getAllBlacklist();

        if(listview.getChildCount() > 1)

listview.removeFooterView(listview.getChildAt(listview.getChildCount
() - 1));

listview.setAdapter(new CustomArrayAdapter(this, R.layout.list_item,
blockList));

        populateNoRecordMsg();
    }

    private void showDialog()
    {

        final AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);

        alertDialogBuilder.setMessage("Are you Really want to
delete the selected record ?");

        alertDialogBuilder.setPositiveButton("Delete",new
DialogInterface.OnClickListener()

        {

            @Override
            public void onClick(DialogInterface arg0, int
arg1)
            {

                try

                {

blackListDao.delete(blockList.get(selectedRecordPosition));

blackList.remove(selectedRecordPosition);
                listview.invalidateViews();

                selectedRecordPosition = -1;
                populateNoRecordMsg();
                } catch (Exception e)

                {

                    e.printStackTrace();
                }
            }
        });

        alertDialogBuilder.setNegativeButton("Cancel",new
DialogInterface.OnClickListener()

```

```

        {
            @Override
            public void onClick(DialogInterface dialog, int
which) {
                }
            });

            final AlertDialog alertDialog =
alertDialogBuilder.create();

            alertDialog.show();
        }

public class CustomArrayAdapter extends ArrayAdapter<String>
{
    private LayoutInflater inflater;

    private List<Blacklist> records;

    public CustomArrayAdapter(Context context, int resource,
@SuppressWarnings("rawtypes") List objects)
    {
        super(context, resource, objects);

        this.records = objects;
        inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public View getView(int position, View convertView,
ViewGroup parent) {

        if(convertView == null)
            convertView = inflater.inflate(R.layout.list_item,
parent, false);

        final Blacklist phoneNumber = records.get(position);

        ((TextView)convertView.findViewById(R.id.serial_tv)).setText("
" + (position + 1));

        ((TextView)convertView.findViewById(R.id.phone_number_tv)).set
Text(phoneNumber.phoneNumber);
        return convertView;
    }
}

```

}

}

}



## ITELEPHONY



### CODES:

```
package com.android.internal.telephony;
```

```
/**
```

```
 * Created by HP on 15-12-2016.
```

```
 */
```

```
public interface ITelephony
```

```
{
```

```
    boolean endCall();
```

```
    void answerRingingCall();
```

```
    void silenceRinger();
```

```
}
```

**DATABASEHELPER.JAVA****CODES:**

```
package com.callmgr.dnd.dndcallmanager;

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

public class DatabaseHelper extends SQLiteOpenHelper
{
    private static final String DATABASE_NAME = "call_blocker.db";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_BLACKLIST = "blacklist";
    private static final String TABLE_CREATE = "create table " +
TABLE_BLACKLIST + "( id "
        + " integer primary key autoincrement, phone_number text
not null);";

    public DatabaseHelper(Context context)
    {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db)
    {
        db.execSQL(TABLE_CREATE);
    }

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

## CALLBARRING.JAVA

## CODES:

```

package com.callmgr.dnd.dndcallmanager;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.telephony.TelephonyManager;
import com.android.internal.telephony.ITelephony;
import java.lang.reflect.Method;

public class CallBarring extends BroadcastReceiver
{
    private String number;

    @Override
    public void onReceive(Context context, Intent intent)
    {
        if
(!intent.getAction().equals("android.intent.action.PHONE_STATE"))
            return;

        else
        {
            number =
intent.getStringExtra(TelephonyManager.EXTRA_INCOMING_NUMBER);

            if(MainActivity.blockList.contains(new
Blacklist(number)))
            {
                disconnectPhoneItelephony(context);
                return;
            }
        }
    }

    @SuppressWarnings({ "rawtypes", "unchecked" })

    private void disconnectPhoneItelephony(Context context)
    {
        ITelephony telephonyService;
        TelephonyManager telephony = (TelephonyManager)
            context.getSystemService(Context.TELEPHONY_SERVICE);

        try
        {
            Class c = Class.forName(telephony.getClass().getName());
            Method m = c.getDeclaredMethod("getITelephony");
            m.setAccessible(true);
            telephonyService = (ITelephony) m.invoke(telephony);
            telephonyService.endCall();
        }
    }
}

```

```
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
```

**BLACKLISTDAO.JAVA****CODES:**

```
package com.callmgr.dnd.dndcallmanager;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import java.util.ArrayList;
import java.util.List;

public class BlacklistDAO

(
    private SQLiteDatabase database;
    private DatabaseHelper dbHelper;

    public BlacklistDAO(Context context)
    {
        dbHelper = new DatabaseHelper(context);
        open();
    }

    private void open() throws SQLException
    {
        database = dbHelper.getWritableDatabase();
    }

    public void close()
    {
        dbHelper.close();
    }

    public Blacklist create(final Blacklist blackList)
    {
        final ContentValues values = new ContentValues();
        values.put("phone_number", blackList.phoneNumber);

        final long id =
database.insert(DatabaseHelper.TABLE_BLACKLIST , null, values);

        blackList.id = id;
        return blackList;
    }
}
```

```
public void delete(final Blacklist blacklist)
{
    database.delete(DatabaseHelper.TABLE_BLACKLIST,
"phone_number = '" + blacklist.phoneNumber + "'", null);
}

public List<Blacklist> getAllBlacklist()
{
    final List<Blacklist> blacklistNumbers = new
ArrayList<Blacklist>();

    final Cursor cursor =
database.query(DatabaseHelper.TABLE_BLACKLIST, new
String[]{"id", "phone_number"}, null, null, null, null, null);

    cursor.moveToFirst();

    while (!cursor.isAfterLast())
    {
        final Blacklist number = new Blacklist();

        number.id = cursor.getLong(0);
        number.phoneNumber = cursor.getString(1);

        blacklistNumbers.add(number);

        cursor.moveToNext();
    }
    return blacklistNumbers;
}
}
```

**BLACKLIST.JAVA****CODES:**

```
package com.callmgr.dnd.dndcallmanager;

public class Blacklist
{
    public long id;
    public String phoneNumber;

    public Blacklist()
    {
    }

    public Blacklist(final String phoneNumber)
    {
        this.phoneNumber = phoneNumber;
    }

    @Override
    public boolean equals(final Object obj)
    {
        if(obj.getClass().isInstance(new Blacklist()))
        {
            final Blacklist bl = (Blacklist) obj;

            if(bl.phoneNumber.equalsIgnoreCase(this.phoneNumber))
                return true;
        }
        return false;
    }
}
```

## ADDTOBLOCKLISTACTIVITY.JAVA

## CODES:

```

package com.callmgr.dnd.dndcallmanager;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class AddToBlocklistActivity extends Activity implements
OnClickListener
{
    private EditText country_code_et, phone_et;
    private Button reset_btn, submit_btn;

    private BlacklistDAO blackListDao;

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

        blackListDao = new BlacklistDAO(this);

        country_code_et = (EditText) findViewById(R.id.country_code_et);
        phone_et = (EditText) findViewById(R.id.phone_et);
        reset_btn = (Button) findViewById(R.id.reset_btn);
        submit_btn = (Button) findViewById(R.id.submit_btn);

        reset_btn.setOnClickListener(this);
        submit_btn.setOnClickListener(this);
    }

    @Override
    public void onClick(View v)
    {
        if(v == submit_btn)
        {
            if(country_code_et.getText().toString().trim().length() > 0 &&
phone_et.getText().toString().trim().length() > 0)
            {
                final Blacklist phone = new Blacklist();

                phone.phoneNumber = "+" +
country_code_et.getText().toString() +
phone_et.getText().toString();
            }
        }
    }
}

```



```

        blacklistDao.create(phone);
        showDialog();
    }
    else
    {
        showMessageDialog("All fields are mandatory !!");
    }
    else if(v == reset_btn)
    {
        reset();
    }
}

private void reset()
{
    country_code_et.setText("");
    phone_et.setText("");
}

private void showDialog()
{
    final AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);

    alertDialogBuilder.setMessage("Phone Number added to block list
successfully !!");

    alertDialogBuilder.setPositiveButton("Add More", new
DialogInterface.OnClickListener()
{

@Override
public void onClick(DialogInterface arg0, int arg1)
{
    reset();
}
});

    alertDialogBuilder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener()
{
@Override
public void onClick(DialogInterface dialog, int which)
{
    finish();
}
});

    final AlertDialog alertDialog = alertDialogBuilder.create();

    alertDialog.show();
}

private void showMessageDialog(final String message)

```

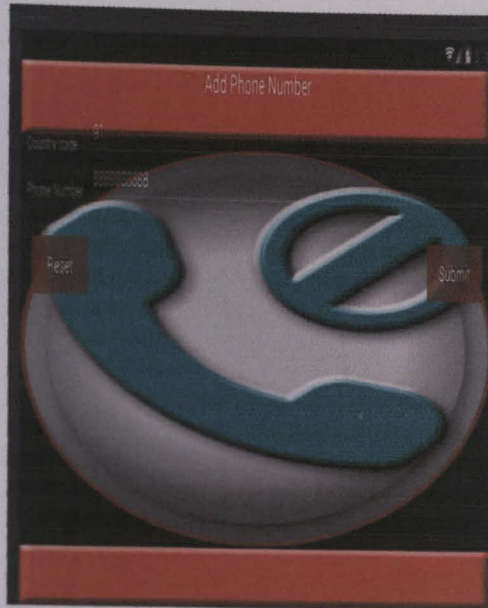


```
{  
    final AlertDialog.Builder alertDialogBuilder = new  
AlertDialog.Builder(this);  
    alertDialogBuilder.setMessage(message);  
    final AlertDialog alertDialog = alertDialogBuilder.create();  
    alertDialog.show();  
}  
}
```



11. Sample Input/ Output Screens:

Input: fig (1) Adding Number To Block List



Output: fig (2) Blocked Numbers

