Development of Smart City Dustbin Mobile Application for Android

Abdulraouf Hakeem Ishola¹,², O. Oyebanji³
1, 3- Department of Computer, Science National Open University, Nigeria.
2- Department of Mechatronics and System Engineering, Abubakar Tafawa Balewa University PMB 0248, Bauchi, Nigeria

ABSTRACT:
One of the problem facing developing countries and particularly an urban city like Lagos in Nigeria is the issue of poor waste management by citizen. This definitely affects the standard of living of both the humans being and animals. Spread of diseases, foul smell in the air and unpleasant look of the city are few of its consequences. The cause of this can be trace to either non availability of Dustbins within the vicinity or their addresses and locations are not known. In the development of smart city dustbin mobile application for android, such problems are taking in to consideration. Its result generates a mobile Android app called Smart Dustbin that have Database which contain the proposed names and addresses for different dustbins. These are displayed on the screen of any Android mobile Smartphone’s the app is installed on and can then be used to locate the dustbin within the user’s vicinity. If a particular address is not descriptive enough or the precise location of the dustbin is proofing difficult for the user, such dustbin can be click. This will launch a goggle map, search with the address and bring the precise location of the dustbin with other relevant information that will guide the user.

KEYWORD: Application, Smartphone’s, Dustbin, Android, Code.

1. INTRODUCTION
Development of cities around the world comes along with numerous improvements, few of which are population growth, high standard of living and improve economic activities. These as well have some unpleasant consequences such as traffic congestion, solid waste pollution, high crime rate, limited infrastructures and many more that need proactive and effective management. These necessitate for the use of Information and Communication technology, giving birth to the concept of smart city. Environmental pollution and waste management in a smart city can be effectively implemented by developing mobile application that can access the location and availability of internet and Wi-Fi enable Dustbins, Lagos Nigeria is our research target area due to it population.

The use of information and communication technologies increase operational efficiency within the public and improves both the quality of government services and citizen welfare, such as automation system and internet of things (IoT) are driving smart city adoption [1]. However, In June 2016, Lagos state government signed a smart city initiated memorandum of understanding with Dubai Holdings LLC, owners of Smart City (Dubai) LLC, “to develop sustainable, smart, globally connected knowledge-based communities that drive a knowledge economy” [2]. Smart city initiatives aim to monitor and address environmental concerns such as climate change and air pollution. Sanitation can also be improved with smart technology, be it using internet-connected trash cans and IoT-enabled fleet management systems for waste collection and removal [3 and 4]. This project proposed the design and development of the client layer of a smart Dustbin for an Android platform. The project develops the mobile application that work on android platform which requires programming in XML and Java programming language and incorporated with SQLite written database. The application uses relevant data collected from citydustbins to populate the User Interface which can be used to dispose waste properly.

1.2. Waste Management
Waste can be defined as solid substances generated as a result of human activities, and, being no longer of value for the respective economic, physiological or technological process, are removed from it [5]. They are usually termed as materials that have been used up either in the household, industries, agricultural environment or public domain. Effect of waste left unmanaged is not desirable as lot of them are not biodegradable and can litter around for years. Waste
management is the process of treating solid wastes and offers variety of solutions for recycling items that don’t belong to trash. It is about how garbage can be used as a valuable resource. Waste management is something that each and every household and business owner in the world needs [6]. Waste management disposes of the products and substances that you use in a safe and efficient manner. According to [7] waste management or waste disposal is all the activities and actions required to manage waste from its inception to its final disposal. This includes collection things, transport, treatment and disposal of waste together with monitoring and regulation. It also encompasses the legal and regulatory framework that relates to waste management encompassing guidance on recycling etc.[8].

Waste management have hierarchy which by arranging from the most preferred option to least preferred option are source reduction and reuse, recycling and composting, waste-to-energy and lastly, treatment and disposal [9]. All this options need effective collection for implementation with the use of dustbins.

2. MATERIALS

2.1. Android

Android is an operating system with open source software [10]. Then, it was primarily developed for touch screen mobile devices such as Smartphone’s then further developed with LED to Android TV for Televisions, Android Auto for Cars and Wear OS for Wrist Watches. The operating system had a multiple releases with current version being 9 “pie”.

2.2. Android Platform (OS)

Mobile applications can be developed through various platforms known as operating systems, mobile operating system is a software that permit the smooth running of programs and applications on devices such as Smartphone’s, Tablets, Wristwatches etc. Such OS are built by different companies. Example of such includes Apple iOS, Google Android, Research in motion’s Blackberry OS, Nokia’s Symbian, Hewlett-Packard’s web OS (formerly Palm OS) and Microsoft’s windows. [10]

2.3. Integrated Development Environment (IDE)

An Integrated Development Environment is a software suite that consolidates basic tools required to write and test software. It helps in the selection, deployment, integration and management of various tools needed by developers in a single package. Tools such as text editors, code libraries, compilers and test platforms. This is meant to enhance the productivity of software developer.[11] There are various types of IDEs which can be use base on the programming language such as Apples iOS, IDE was used for the development support to Apple’s Swift programming language, Xcode and Appcode are example of such IDEs for iOS. For Google’s Android applications, Eclipse and Android Studio (AS) are example of suitable IDEs because they support Java and android App.[12]

Fig. 1. Project structure.

3. METHODOLOGY

3.1. Stages to Build an Android Application

I. Stage one; a great imagination leads to a great App (Decide the feature of your App)
II. Stage two; identify (to create a successful mobile App)
III. Stage three; Design your APP
IV. Stage four; identify approach to develop the APP- Native, web or hybrid
V. Develop a prototype
VI. Integrate an appropriate analytics tools
VII. Identify beta-testers listen to their feedback and integrate relevant one
VIII. Release/delay the APP
IX. Capture the metrics
X. Update your APP with improvement and new features

Fig. 2 explains the stages of developing a mobile APP for competitive market.
3.2. Programming Code for Implementation of Mobile Smart Dustbin App

3.2.1. build.gradle (module: app) code

Gradle Scripts consist of files that are automatically generated by AS itself and to process the program compilation. It manages dependencies and provides custom build logic in order to generate an Android Application Package (APK) files that can be signed and pushed to the devices where it gets executed. For this research, Smart Dustbin, target Sdk Version is 28 and minimum Sdk version is 15 as seen below.

```java
/**
 * Smart dustbin
 * An Msc research by Abdulraouf Hakeem Ishola
 * co contributor Benjamin Bello
 * s under the supervision of Dr. Oyebanji Omotayo
 */

apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.example.android.dustbin"
        minSdkVersion 15
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles
                getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
    sourceSets {
        main {
            assets.srcDirs = ['src/main/assets', 'src/main/assets/']
        }
    }
    dependencies {
        implementation fileTree(dir: 'libs', include: ['*.jar'])
        implementation 'com.android.support:appcompat-v7:28.0.0'
        implementation 'com.android.support.constraint:constraint-layout:1.0.2'
        implementation 'com.readystatesoftware.sqliteasset:sqiliteassethelper:2.0.1'
        testImplementation 'junit:junit:4.12'
        androidTestImplementation 'com.android.support.test:runner:1.0.1'
        androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
    }
}
```

Fig. 2. Developing Stage block diagram.
3.2.3. Androidmanifest.xml file code
Manifest file has an xml files called AndroidManifest.xml. This file presents essential information about the app to the Android system. These are information the system must have before it can run any of the app’s code. Few of such information are package name, label, launching icon and theme which can be fetching from its source code displayed below.

```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.android.dustbin">
  <application allowBackup="true"
    icon="@drawable/bin_icon"
    label="@string/app_name"
    roundIcon="@mipmap/ic_launcher_round"
    supportsRtl="true"
    theme="@style/AppTheme">
    <activity name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
    <activity name=".DustbinList"/>
    <provider
      authorities="com.example.android.dustbin"
      name=".data.DustbinProvider"
      exported="false"/>
  </application>
</manifest>
```

3.2.4. Activity_main.xml code
Activity_main.xml contains an ImageView with id dustbin_icon that hold space for a dustbin image and lastly is a button with id findBinButton that is coded in MainActivity.java to launch dustbinList activity when clicked.

```xml
<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"
  tools:context=".MainActivity">
  <TextView
    android:id="@+id/quoteTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:textColor="@android:color/black"
    android:textSize="24sp"
    android:textStyle="italic|bold"
    android:padding="24dp"
    android:paddingBottom="64dp"/>
  <ImageView
    android:id="@+id/dustbin_icon"
    android:layout_width="132dp"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:layout_gravity="center"
    android:layout_marginLeft="16dp"
    android:layout_marginStart="16dp"
    tools:ignore="ContentDescription"/>
  <Button
    android:id="@+id/findBinButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
</LinearLayout>
```
3.2.5. Dustbinlist.java code

DustbinList.java is a inflate second layout called activity_dustbin_list and set view content on it when the APP was launched to find dustbins button in activity_main for the address of a dustbin from your environment when you click ListView it will search on goggle map and locate, direct and give you the time interval need to navigate to dispose your waste materials.

/*
 * Smart Dustbin
 * An MSc project by Abdulraouf Hakeem Ishola
 * contributor Benjamin Bello
 * under the supervision of Dr. Oyebanji
 */

package com.example.android.dustbin;

import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import com.example.android.dustbin.data.dustbinContract;
import com.example.android.dustbin.data.dustbinContract.DustbinEntry;
import com.example.android.dustbin.data.dustbinDbHelper;

/**
 * Displays list of Dustbins that were stored in the app database.
 */
public class DustbinList extends AppCompatActivity {

    // Defining the ListView which will be populated with the dustbin data
    private ListView listview;

    // Database helper that will provide us access to the database
    private dustbinDbHelper mDbHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dustbin_list);
        listview = findViewById(R.id.lvItems);
        mDbHelper = new dustbinDbHelper(this);

        // To access our database, we instantiate our subclass of SQLiteOpenHelper
        // and pass the context, which is the current activity.
        mDbHelper = new dustbinDbHelper(this);

        //setting click listener on ListView to launch google map
        listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                //select address to use for search in google map intent
                String selectedAddress = ((TextView) view.findViewById(R.id.address_textView)).getText().toString();
                Uri gmmIntentUri = Uri.parse("google.navigation:q=" + selectedAddress);
                Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
                mapIntent.setPackage("com.google.android.apps.maps");
                if (mapIntent.resolveActivity(getPackageManager()) != null) {
                    startActivity(mapIntent);
                }
            }
        });
    }
}
3.2.6. ACTIVITY_DUSTBIN_LIST. xml CODE

The activity_dustbin_list.xml source code is a
LinearLayout xml layout resources file which has a
ListView child view with id “lvItems” that present its
list of items in a vertical orientation.

```xml
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".DustbinList">

    <ListView
        android:id="@+id/lvItems"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_height="match_parent">
    </ListView>
</LinearLayout>
```

3.2.7. DustbinCursorAdapter.java code

DustbinCursorAdapter.java is a Java class file and
as the name implies, it implements the adaptation of
cursor data from the app database on the appropriate
TextView in List_item.xml Layout resources file. The
Adapter created inflates new views, retrieve names and
addresses from cursor and bind them each on separate
view individually.

```java
package com.example.android.dustbin;
import android.content.Context;
import android.database.Cursor;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.CursorAdapter;
import android.widget.TextView;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.ViewGroup;
import android.widget.ViewGroup.LayoutParams;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.Text
public class DustbinsCursorAdapter extends CursorAdapter {
    public DustbinsCursorAdapter(Context context, Cursor c) {
        super(context, c, 0);
    }

    //Method to inflate a new view and return it
    @Override
    public View newView(Context context, Cursor cursor, ViewGroup parent) {
        return LayoutInflater.from(context).inflate(R.layout.list_item, parent, false);
    }

    //Method to bind all data to a given view
    @Override
    public void bindView(View view, Context context, Cursor c) {
        TextView tvName = view.findViewById(R.id.name_textView);
        TextView tvAddress = view.findViewById(R.id.address_textView);

        String name = cursor.getString(cursor.getColumnIndexOrThrow(DustbinContract.DustbinEntry.COLUMN_DUSTBIN_NAME));
        String address = cursor.getString(cursor.getColumnIndexOrThrow(DustbinContract.DustbinEntry.COLUMN_DUSTBIN_ADDRESS));

        tvName.setText(name);
        tvAddress.setText(address);
    }
}

3.2.9. DustbinContract.java
DustbinContract.java is a Java class that defines constant values needed in our app. Most of which are used for database implementation and few for content provider.

package com.example.android.dustbin.data;
import android.net.Uri;

<?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">
    <TextView
        android:id="@+id/name_textView"
        android:layout_width="wrap_content"
        android:padding="16dp"
        android:layout_height="wrap_content"
        android:textColor="#232323"/>
    <TextView
        android:id="@+id/address_textView"
        android:layout_width="wrap_content"
        android:padding="16dp"
        android:layout_height="wrap_content"
        android:textColor="#934747"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:fontFamily="sans-serif"/>
</LinearLayout>

3.2.8. List_item.xml
xml is a layout resource file with source code that helps create the adapter as needed. It has a LinearLayout with vertical orientation and two TextView. Textview
import android.provider.BaseColumns;

public final class dustbinContract {
    // empty constructor
    private dustbinContract(){
    }

    // inner class that define constant values for the dustbin database.
    public static final class DustbinEntry implements BaseColumns {
        // Name of database table for dustbin
        public final static String TABLE_NAME = "myDustbinsTb";

        // Unique ID number for the dustbin (only for use in the database table). Type: INTEGER
        public final static String _ID = BaseColumns._ID;

        // Name of the dustbin. Type: TEXT
        public final static String COLUMN_DUSTBIN_NAME = "name";

        // Address of dustbins. Type: TEXT
        public final static String COLUMN_DUSTBIN_ADDRESS = "address";

        // URI build up
        private final static Uri BASE_CONTENT_URI = Uri.parse("content://com.example.android.dustbin");

        public final static Uri CONTENT_URI = Uri.withAppendedPath(BASE_CONTENT_URI, PATH_DUSTBINS);

        public static final String CONTENT_AUTHORITY = "com.example.android.dustbin";

        public final static String PATH_DUSTBINS = "dustbins";
    }

    3.2.10. DustbinProvider.java
    Asset such as databases needs protection because it possesses carefully organized data whose positions must not be tempered with. DustbinProvider does this by giving access to the database content from authorized authority only. If there is an attempt from an authority that does not match the authorized one, it throws IllegalArgumentException saying "cannot query unknown URI"
3.2.11. DustbinDbHelper.java

This is a Java class that is used to manage the app database by extending SQLiteAssetHelper class which is a helper class from which dustbinDbHelper class inherited all its behaviour and work by introducing dependency.

```java
package com.example.android.dustbin.data;
import android.content.Context;
import com.readystatesoftware.sqliteasset.SQLiteAssetHelper;

/**
 * A class to help setup our SQLite written Database "myDustbins.db" with initial version 1
 * It comes with its dependency from "com.readystatesoftware.sqliteasset.SQLiteAssetHelper".
 */
public class dustbinDbHelper extends SQLiteAssetHelper{
    private static final String DATABASE_NAME = "myDustbins.db";
    private static final int DATABASE_VERSION = 1;
    public dustbinDbHelper(Context context){
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
}
```

3.2.12. Asset folder and database

Asset folder is an app’s databases folder which contains a directory folder this app (Smart dustbin) database is an SQLite written file named myDustbins.db. This contain the location of where the dustbin is been kept or being positioning for better waste management in selected areas in Lagos Nigeria and two location in Oyo Nigeria. myDustbins.db contain a single table called myDustbinsTb that have three (3) columns and twenty(20) rows as shown in table 1. _id, name and address are the columns names that describe details of dustbins. _id is an auto generated and auto-incremented integer that are unique to each dustbin as it serves as the primary key of the table. Name is the tag to identify individual dustbin and address describe the specific locations. The rows indicate the number of dustbin present and each row give the full description of each dustbin.

<table>
<thead>
<tr>
<th>_id</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ogodonto001</td>
<td>Ogolonto Bustop Ikorodu</td>
</tr>
<tr>
<td>2</td>
<td>Agric001</td>
<td>Agric bustop Terminal, Ikorodu</td>
</tr>
<tr>
<td></td>
<td>Ebute003</td>
<td>Ebute Ultra Mordern Complex</td>
</tr>
<tr>
<td>4</td>
<td>Ebute001</td>
<td>Ebute Amusement Cente</td>
</tr>
<tr>
<td>5</td>
<td>VON</td>
<td>VON Alogba Junction</td>
</tr>
<tr>
<td>6</td>
<td>ALOGBA</td>
<td>Alhaja Sideeqat Ahmad Mejida Centre Mosque</td>
</tr>
<tr>
<td>7</td>
<td>Ogodonto002</td>
<td>Justrite Superstore Ikorodu</td>
</tr>
<tr>
<td>8</td>
<td>NOUN</td>
<td>National Open University of Nigeria</td>
</tr>
<tr>
<td>9</td>
<td>SilverBird</td>
<td>SilverBird Taxi Park</td>
</tr>
<tr>
<td>10</td>
<td>Bonny Camp</td>
<td>Bonny Camp Bus Stop</td>
</tr>
<tr>
<td>11</td>
<td>Obalende001</td>
<td>Obalende Bus Park</td>
</tr>
<tr>
<td>12</td>
<td>Obalende002</td>
<td>BRT Park Obalende</td>
</tr>
<tr>
<td>13</td>
<td>Alausa</td>
<td>Alausa Secretariat 2nd Bus Stop</td>
</tr>
<tr>
<td>14</td>
<td>Ikeja</td>
<td>Ikeja Along Bussstop</td>
</tr>
<tr>
<td>15</td>
<td>Ketu</td>
<td>Ketu Bus-Stop</td>
</tr>
<tr>
<td>16</td>
<td>Mushin</td>
<td>Mushin Post Office</td>
</tr>
<tr>
<td>17</td>
<td>Surulere001</td>
<td>National Stadium, Lagos</td>
</tr>
<tr>
<td>18</td>
<td>Surulere002</td>
<td>National Stadium Police Station</td>
</tr>
<tr>
<td>19</td>
<td>Takie</td>
<td>Takie Motor Park</td>
</tr>
<tr>
<td>20</td>
<td>LAUTECH</td>
<td>Ladoke University, Ogbomoso</td>
</tr>
</tbody>
</table>
3.3. RES File

Res directory contains all resources need for the app layout (explained earlier), mipmap, values and drawable but the drawable folder host two (2) image resources of the app (bin_icon.png and bin_icon_launcher.png), mipmap for different launcher icon densities and in values subdirectory, files such as colors.xml, strings.xml and style.xml are housed.

4. EXPERIMENTAL RESULTS

4.1 Compilation, Installation and Launch Icon

Compilation and installation of codes, assets and resources from smart dustbin package was test run on an emulator Nexus 6 Api 25 (Android 7.1.1) and as well on an Infinix Hot 4 pro (X556) which is an android phone with 5.5-inch touch display and Android 6.0 (marshmallow) operating system. The result of this generated a 1.5Mb Android Application Package (APK) that have two activities and a launch icon. The launch icon is a rounded edge square shaped icon with a green dustbin image (the bin_icon_launcher.png designed by Robin Weather all and downloaded from IconArchive) and app name “Smart Dustbin” centrally positioned under it. When you click on it to launched app, it will display the first activity.

4.1.1. First activity

The first activity launched after a click on the launched icon is activity_main which display a quote, a dustbin image and a button labelled FIND DUSTBINS as shown in Fig. 3. A click on the button launched the second activity.

4.1.2. Second activity

The second activity launched is the activity_dustbin_list that present a scrollable list of dustbin names and addresses. It contains twenty (20) different dustbins. Any specific dustbin with an address closer or of interest to the user can be located. Otherwise if the address or location is not descriptive enough to the user then it can be clicked. The click launches google map and search for the location of the dustbin with its address.

Google map

Google map activity launched have lot of features to that help user locate the dustbin selected. The features are:
1. The exact distance in kilometre from the user location to the selected dustbin
2. A prescribed direction and the fastest route to it
3. The expected time needed for the journey depending on the means of transport used
4. monitoring of user progress while approaching target with the help of audio instruction

However, Fig. 6 explained a dustbin located at Alhaja sideeqat Ahmad mosque it was 1.2km away from the
user and it will take him two (2) minute to reach there at fastest route. Also Fig. 7 explained the user is at Voice of Nigeria and locate a dustbin at Ogolonto bus stop which is 4.5km far from him and it will take him 12 minute to reach there at the fastest route. While Fig. 8 explained the user located at Ikorodu area in Lagos locate a dustbin at national stadium surelere Lagos with a distance of 28km while it will take him 46 minute to reach at fastest route.

![Google Map Screenshot of Alhaja Sideeqat Mosque dustbin.](image1)

Fig. 6: Google Map Screenshot of Alhaja Sideeqat Mosque dustbin.

![Google Map Screenshot of Ogolonto Bus Stop dustbin.](image2)

Fig. 7: Google Map Screenshot of Ogolonto Bus Stop dustbin.

![Google Map Screenshot for National Stadium Surelere dustbin.](image3)

Fig. 8: Google Map Screenshot for National Stadium Surelere dustbin.

5. CONCLUSION

The development of smart city dustbin mobile application for android is done to enhance the effectiveness of solid waste management. Environmental pollution is causing a lot of distress not only to humans but also animals, driving many animal species to endangerment and even extinction. As Smartphone’s are handy to all and mobile in usage, bringing dustbin addresses, location and descriptive details of the populate area that makes waste disposal easier. It will improve their orientation toward littering the environment.

REFERENCES


