GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

METRO AND BUS SERVICE APP

A Report for the Evaluation 3 of Project 2

Submitted by

SNIGDHA(1613101743 / 16SCSE101375)

in partial fulfillment for the award of the degree of

Bachelor of Technology

IN

Computer Science and Engineering

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Dr. J.N.Singh,Professor

APRIL / MAY- 2020

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **“METRO AND BUS SERVICE APP”** is the bonafide work of **“SNIGDHA (1613101743)”** who carried out the project work under my supervision.

SIGNATURE OF HEAD

DR. MUNISH SHABARWAL,
PhD(Management),PhD(CS)
Professor & Dean,
School of computing Science & Engineering

SIGNATURE OF SUPERVISIOR

Dr.J.N Singh,
Professor
School of computing Science & Engineering

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO.
1.	Abstract	1
2.	Introduction	1-2
3.	Existing System	2
4.	Proposed system	2-3
5.	Coding and implementations	3-13
6.	Output / Result / Screenshot	13-16
7.	Conclusion/Future Enhancement	16-17
8.	References	18

ABSTRACT

The purpose of this project is to create an android application which can be used by the people residing in Delhi.Many android applications are also present but each has some cons in it. I have tried to remove all such types of bugs..In our daily life we want to save our time and money.with the help of “METRO AND BUS SERVICE APP” we can save out time as it shows minimal distance and minimal money required to travel from one place to another.

This android application not only works on online mode but also we can use it in offline mode.There is also an emergency tool given which will help you to make your journey easy.There is also a toll free number option given which you can use if you are stuck anywhere in the app.The helpline number option is also available to call the police or to call an Ambulance.Thus,the main aim of this app is to create a user-friendly application which can be used easily.

The technology which we have used are- SQLite,Android and Java.

This Application provides us to enable the easy travelling in Delhi and Noida without using the internet and the Maps or GPS in mobile .It solves one major problem that is if we have to find out our way through we need to use GPS for which we need an internet connection. In contradiction to this app works offline and provides all the information without an internet connection.

INTRODUCTION

This project aims at developing an android applications i.e. a mobile app to enable people to find out their way while travelling in Delhi. The app gives the basic details of the various ways to travel in Delhi that mainly are metro and DTC bus services.

It includes:

1. Description of the metro services including the various stations, routes and the lines(blue ,green ,yellow ,violet etc.) and also enables planning of these routes.

2. A brief description of the bus services. About the stations and the source and start of a particular bus route and all the important information that is necessary to use the service.

3. Enables tracking routes on the metro map.

4. Customer care calling: it enables users to contact the emergency helpline numbers like
   a. Police
   b. Ambulance
   c. Fire station.

1 Technology

- Java
- Android
- SQLite

2. Advantages

- It is easy to use.
- It provides customer care support.
- It can be used offline.
- It is user friendly.

EXISTING SYSTEM

In the existing system of “Metro and Bus” application there was a need of Internet connection even the android applications used to hang often .There was also no customer care service available and we have to install two separate applications which was quite time and storage taking.If we want to go from DTC buses then we have to install a different application whereas for metro we need another.

Thus,these all were the drawbacks of existing system which I have tried to overcome with the help of an android application named as”Metro and Bus service App”.

PURPOSE OF THE SYSTEM

The purpose of this android app is to make it easy to travel in Delhi for those who are not much familiar with the options. Along with providing the details of the various ways to go from one place to another, it also provides a calling feature to any of the customer care if the user is stuck in any problem and saves the user form the hectic task of searching numbers and following the whole procedure to make a call instead an emergency call can be at the single click of a button.It also works fine in an offline mode.The app is safe to use and it doesn’t require more storage.

PROBLEM STATEMENT

This Application provides us to enable the easy travelling in Delhi and Noida without using the internet and the Maps or GPS in mobile .

It solves one major problem that is if we have to find out our way through we need to use GPS for which we need an internet connection. In contradiction to this app works offline and provides all the information without an internet connection.

METHODOLGY-ADOPTED

I followed a 5-Step Development Methodology to develop this Android Application named Delhi Route Planner.

1. Study and Learning: Studying java and android and gaining the basic knowledge of the software like eclipse and databases.

2. Layout: Finding out the details to be shown and then designing a UI to best suit the need using the UI designing feature provided by the android.

3. Development: Once it was finalized how the UI was to be designed and the number and type of activities and screen they were designed and developed one after the another.

4. Implementation: Once the basic designing was done I tested the app for my mobile and ensured that all the features are working properly.

5. Demo: The final step is to give a demo of the Delhi route planner.

CODE

BusRouteDetail.java

```
package com.planner;
```

```
import java.io.IOException;
import java.util.ArrayList;

import android.app.Activity;
import android.database.Cursor;
import android.database.sqlite.SQLiteException;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.Toast;

public class BusRouteDetail extends Activity
{
    ImageView _back;
    Spinner _RouteSpinner;
    Button _fetchDetail, _toMainMenu;
    EditText _showDetail;
    String selectedRoute;
    ArrayAdapter<String> adapt;
    ArrayList<String> BusRoutes = new ArrayList<String>();

public void onCreate(Bundle bun){
        super.onCreate(bun);
        setContentView(R.layout.busroutedetail);
        populateFrom();
        try{
            fillSpinner();
        }catch(Exception ae){
            Log.w("BusRouteDetail",""+ae.getMessage());
        }
}

public void populateFrom(){
    _showDetail =(EditText)findViewById(R.id.DRDeditboxshowdetail);
    _showDetail.setFocusable(false);
    _showDetail.setEnabled(false);

//  _back = (ImageView)findViewById(R.id.DRDimageviewback);
//  _back.setOnClickListener(this);

    _RouteSpinner = (Spinner)findViewById(R.id.DRDspinnerroutes);
    _RouteSpinner.setOnItemSelectedListener(new OnItemSelectedListener() {

        public void onItemSelected(AdapterView<?> parent, View view, int pos,
                long arg3) {
            selectedRoute = parent.getItemAtPosition(pos).toString();
        }

        public void onNothingSelected(AdapterView<?> arg0) {
            // TODO Auto-generated method stub

        }

    });

    _fetchDetail = (Button)findViewById(R.id.DRDbuttonfetchroutedetail);
    _fetchDetail.setOnClickListener(new OnClickListener() {

        public void onClick(View v) {
            try{
                getRouteDetail();
            }catch(Exception ae){
                Log.w("Error in Fetching Route Detail",""+ae.getMessage());
            }

        }
    });
//  _toMainMenu = (Button)findViewById(R.id.DRDbuttonmainmenu);
    //_toMainMenu.setOnClickListener(this);

}


private void fillSpinner()throws SQLiteException, IOException {

     DBHelperBusRouteDetail db = new DBHelperBusRouteDetail(this);
     db.openDataBase();
     Cursor cur=db.getRoutesName();

     if(cur.moveToFirst())
        {
            do
            {
                String str=cur.getString(0);
                BusRoutes.add(str);
            }while(cur.moveToNext());
            adapt = new ArrayAdapter<String>(getApplicationContext(),
android.R.layout.simple_spinner_dropdown_item,BusRoutes);
        _RouteSpinner.setAdapter(adapt);

        }
     cur.close();
}
private void getRouteDetail() throws SQLiteException, IOException{
     String Starts, Ends,  MiddleStops = "No Stops", Length,  Detail = "";

     DBHelperBusRouteDetail db = new DBHelperBusRouteDetail(this);
     db.openDataBase();
     Cursor cur=db.getRouteDetail(selectedRoute);

     if(cur.moveToFirst())
        {
            do
            {
                Starts = cur.getString(1);
                Ends = cur.getString(2);
                MiddleStops = cur.getString(3);
                Length = cur.getString(4);
                //Toast.makeText(getApplicationContext(), "Working in loop",
Toast.LENGTH_SHORT).show();
            }while(cur.moveToNext());


        Detail =" ****** Route No: "+selectedRoute+"***** \n"
                +"Route Length: "+Length+" KM \n"+
                "Starts At:   "+Starts+"\n"
          + "Ends At:   "+Ends+"\n"
          +"Middle Stops Are : \n"+MiddleStops+"\n \n \n";

        //Toast.makeText(getApplicationContext(), "Working after Detail Specified",
100).show();
        _showDetail.setText(Detail);


        }
     cur.close();

}
}
```

MetroStationDetail.java

```
package com.planner;

import java.io.IOException;
import java.util.ArrayList;

import android.app.Activity;
import android.database.Cursor;
import android.database.sqlite.SQLiteException;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

public class MetroStationDetail extends Activity  {

    Spinner _stations;
    Button _fetchDetail;
    Button _xxxxx;
    String StationSelected;
    TextView _LineStatus;
    EditText _showDetail;
    ArrayList<String> stationname= new ArrayList<String>();
    ArrayAdapter<String> adapt;

    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.metrostationdetail);
        initilizeComponents();
        try{
            getStations();
        }catch(Exception ae) {

        }

    }
    public void initilizeComponents(){
        _LineStatus = (TextView)findViewById(R.id.MSDtextviewlinestatus);
        _stations = (Spinner)findViewById(R.id.MSDspinnerstations);
        _stations.setOnItemSelectedListener(new OnItemSelectedListener() {

        public void onItemSelected(AdapterView<?> parent, View view, int pos, long
id)
        {
            StationSelected = parent.getItemAtPosition(pos).toString();

        }
        public void onNothingSelected(AdapterView<?> parent){

        }
    });

        _fetchDetail = (Button)findViewById(R.id.MSDbuttonfetchdetail);
        _fetchDetail.setOnClickListener(new OnClickListener() {


            public void onClick(View v) {
                try{
                    getStationsDetail();
                }catch(Exception ae){
                    Log.w("","error"+ae.getMessage());
                }

            }
        });
        _showDetail = (EditText)findViewById(R.id.MSDedittextshowdetail);


    }
```

```
        public void getStations()throws SQLiteException, IOException{
                DBHelperMetro db = new DBHelperMetro(this);
                  db.openDataBase();
                  Cursor cur=db.getStations();

                  if(cur.moveToFirst())
                        {
                                do
                                {
                                        String str=cur.getString(1);
                                        stationname.add(str);
                                }while(cur.moveToNext());
                                adapt = new ArrayAdapter<String>(getApplicationContext(),
android.R.layout.simple_spinner_dropdown_item,stationname);
                        _stations.setAdapter(adapt);

                        }
            }

        public void getStationsDetail()throws SQLiteException, IOException{
                String line, nearby, Detail = "";
                DBHelperMetro db = new DBHelperMetro(this);
                  db.openDataBase();
                  Cursor c = db.getStationDetail(StationSelected);
                  if(c.moveToFirst())
                        {
                                do
                                {
                                        line = c.getString(2);
                                        nearby = c.getString(3);
                                }while(c.moveToNext());
                                _LineStatus.setText("   "+line + "");
                        Detail =" ****"+StationSelected+"***** \n\n"+
                                        "Nearby Stations : "+nearby+"\n";

                                _showDetail.setText(Detail);
                        }
}

}
DBHelperRoteDetail.java
package com.planner;

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class DBHelperBusRouteDetail extends SQLiteOpenHelper {
        private static String DB_NAME = "dtcstops";
        private static String DB_PATH = "/data/data/com.planner/databases/";
        private SQLiteDatabase dbTcr;
        private final Context myContext;
        private static final int DATABASE_VERSION = 1;

public DBHelperBusRouteDetail(Context context){
                super(context, DB_NAME, null, DATABASE_VERSION);
                this.myContext = context;
        }
        public boolean checkDatabase(){
                SQLiteDatabase checkDB = null;
                try{
                        String mypath = DB_PATH +DB_NAME;
                        checkDB = SQLiteDatabase.openDatabase(mypath, null,
SQLiteDatabase.OPEN_READWRITE);
                        Log.e("Test 1", "Table Opened ");
                }catch(Exception ae){
                        Log.e("Test 1", "Error : Unable to open database"+"."+ae.toString());
                }
                if(checkDB!=null){
                        checkDB.close();
                }
                return checkDB !=null ? true:false;
        }

        public void createDatabase(){
                boolean dbExist = checkDatabase();
                if(dbExist)     {
                        Log.e("test","createDatabase[] - db exists");
                        this.getWritableDatabase();
                }else{
                        Log.e("test","createDatabase[] - db not exists");
                        this.getReadableDatabase();
              try {
                copyDatabase();
                Log.e("test","createDatabase[] - copied db");
              }catch (IOException e){
                Log.e("test","createDatabase[] - copied not db");
              }
              SQLiteDatabase checkDB = null;
              String myPath = DB_PATH + DB_NAME;
              checkDB = SQLiteDatabase.openDatabase(myPath,
null,SQLiteDatabase.OPEN_READWRITE);
```

```
        }
    }

    public void copyDatabase() throws IOException, SQLiteException {
        Log.e("tofu","copy database");
        InputStream myInput = myContext.getAssets().open(DB_NAME);

     String outFileName = DB_PATH + DB_NAME;
     OutputStream myOutput = new FileOutputStream(outFileName);

     //transfer bytes from the inputfile to the outputfile
     byte[] buffer = new byte[1024];
     int length;
     while ((length = myInput.read(buffer))>0){
        myOutput.write(buffer, 0, length);
   }
     myOutput.flush();
     myOutput.close();
     myInput.close();
    }

    public void openDataBase() throws SQLiteException, IOException{
        this.createDatabase();
        try {
         String myPath = DB_PATH + DB_NAME;
         dbTcr = SQLiteDatabase.openDatabase(myPath,
null,SQLiteDatabase.OPEN_READONLY);
        }catch(SQLException sqle){
            throw sqle;
        }
 }

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

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // TODO Auto-generated method stub
    Log.e("dbUpgrade","OLD Version = "+oldVersion+"");
    Log.e("dbUpgrade","New Version = "+newVersion+"");
    myContext.deleteDatabase(DB_NAME);
    Log.e("dbUpgrade","deleted"+DB_NAME);
}

public Cursor getRoutesName(){
    String str="select * from dtcstops";
    return dbTcr.rawQuery(str, null);
```

```
}
public Cursor getRouteDetail(String routenum){
        String str = "select * from dtcstops where RouteNo = '"+routenum+"'";
        return dbTcr.rawQuery(str, null);
}


}
```

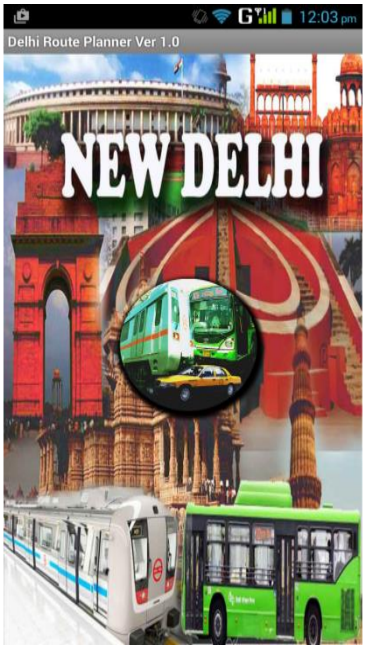
MainScreen.java

```
package com.planner;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.sax.StartElementListener;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.animation.AccelerateDecelerateInterpolator;
import android.view.animation.RotateAnimation;
import android.view.animation.ScaleAnimation;
import android.view.animation.TranslateAnimation;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;


public class MainScreen extends Activity implements OnClickListener{
        Button btnDtc , btnMetro ,btnUtilities, btnmaps;
        ImageView exitImg;
        TextView contactDeveloper;



        public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.mainscreenlayout);
      System.out.println("2 ..... Control at Mainscreen First Line " );
      btnMetro = (Button)findViewById(R.id.buttonMetro);
      btnMetro.setOnClickListener(this);
        btnDtc = (Button)findViewById(R.id.buttonDtc);
        btnDtc.setOnClickListener(this);

      exitImg = (ImageView)findViewById(R.id.Mainexitimageview);
      exitImg.setOnClickListener(this);

  }

        public void onClick(View itemclicked) {
```

```
        try{
            switch(itemclicked.getId()){
            case R.id.buttonDtc:
                Intent godtcscreen = new Intent(getApplicationContext(), DtcServiceScreen.class);
                startActivity(godtcscreen);
                break;
            case R.id.buttonMetro:
                Intent gometroservices = new Intent(getApplicationContext(), MetroServiceScreen.class);
                startActivity(gometroservic gometroservices);
                break;

            case R.id.Mainexitimageview:
                Toast.makeText(getApplicationContext(), "App Will Exit.. Thanks for using", Toast.LENGTH_SHORT).show();
                finish();
                break;
            }
    }catch(Exception ae){
        Toast.makeText(getApplicationContext(), "Error :"+ae.getMessage(), Toast.LENGTH_SHORT).show();
        Intent gomain = new Intent(getApplicationContext(),SplashActivity.class);
        finish();
        startActivity(gomain);
        }
    }

    /*public void myAnimationText(){

     TranslateAnimation sc=new TranslateAnimation( 0, 0, 50, 0) ;
     sc.setInterpolator(new AccelerateDecelerateInterpolator());
         sc.setDuration(1000);
         exitImg.startAnimation(sc);



  }



    public void DTCAnimation(){
        TranslateAnimation sc=new TranslateAnimation( 0, 0, -100, 0) ;
        sc.setInterpolator(new AccelerateDecelerateInterpolator());
        sc.setDuration(1500);
        btnDtc.startAnimation(sc);

    }

    public void MetroAnimation(){
        TranslateAnimation sc=new TranslateAnimation( 0, 0, -100, 0) ;
        sc.setInterpolator(new AccelerateDecelerateInterpolator());
        sc.setDuration(1500);
```

```
                btnMetro.startAnimation(sc);
        }

        public void TaxiAnimation(){
                TranslateAnimation sc=new TranslateAnimation( 0, 0, -100, 0) ;
                sc.setInterpolator(new AccelerateDecelerateInterpolator());
                sc.setDuration(1500);
                btnUtilities.startAnimation(sc);
        }
        public void LocationAnimation(){
                TranslateAnimation sc=new TranslateAnimation( 0, 0, -100, 0) ;
                sc.setInterpolator(new AccelerateDecelerateInterpolator());
                sc.setDuration(1500);
                btnmaps.startAnimation(sc);
        }
        */
}
```

OUTPUT



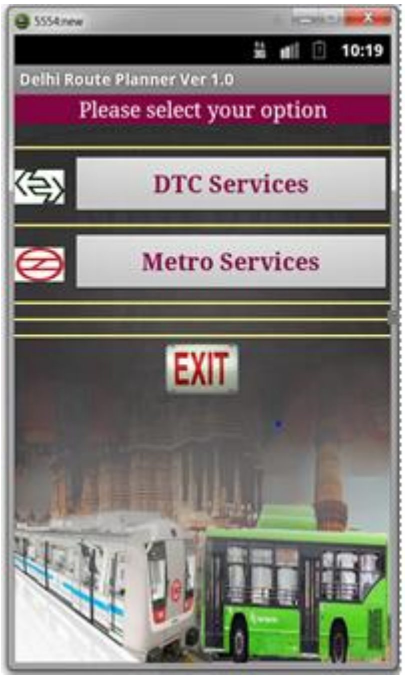


Fig 1: Home page: the main screen of the app. This is called the splash activity which opens up as soon as the app starts up. Before the main activity is displayed or loaded an animation of the logo icon placed in the center of the screen is shown after which the main screen is loaded and options are given for the user to select from among the two options namely for DTC services and Metro Services .

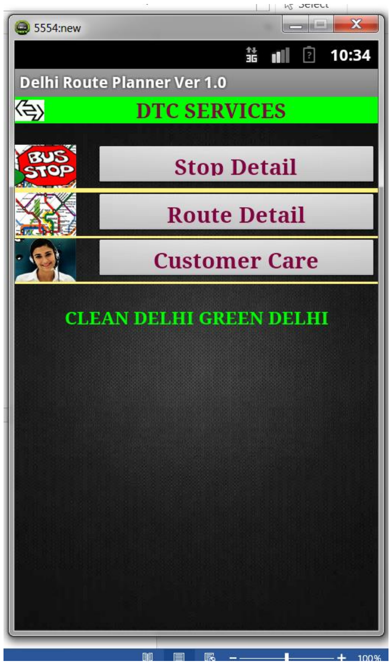
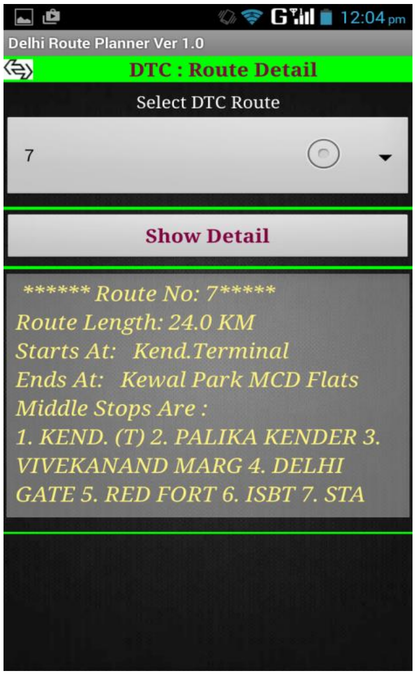
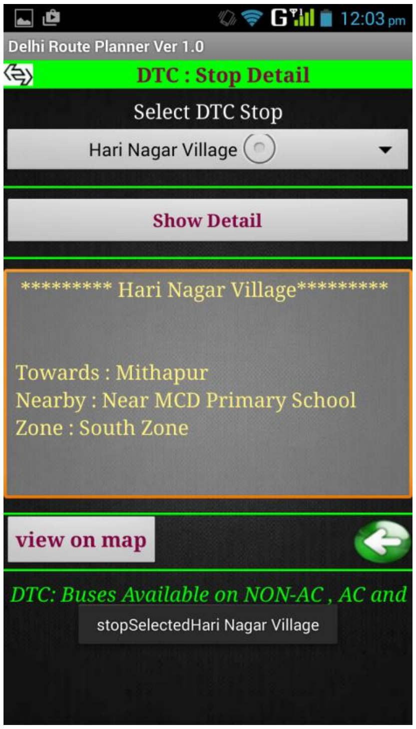


Fig 2: DTC service screen
When user selects DTS service from the main
Screen of the app..



```
****** Route No: 7*****
Route Length: 24.0 KM
Starts At:   Kend.Terminal
Ends At:   Kewal Park MCD Flats
Middle Stops Are :
1. KEND. (T) 2. PALIKA KENDER 3.
VIVEKANAND MARG 4. DELHI
GATE 5. RED FORT 6. ISBT 7. STA
```

Fig 3: Route Detail screen:
Here the user can get information about
the bus route.



Delhi Route Planner Ver 1.0

DTC : Stop Detail

Select DTC Stop

Hari Nagar Village

Show Detail

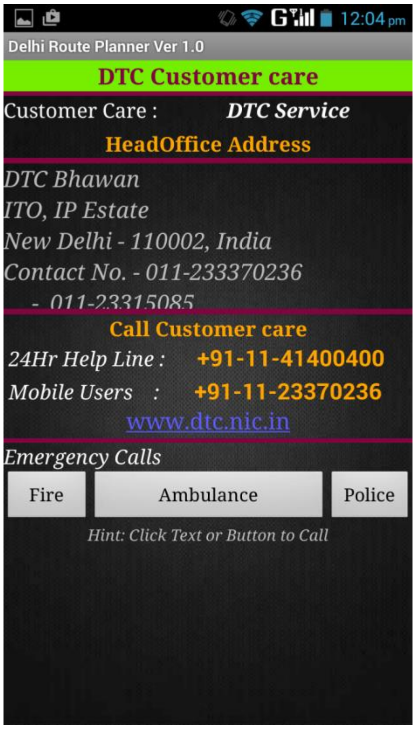
********* Hari Nagar Village*********

Towards : Mithapur
Nearby : Near MCD Primary School
Zone : South Zone

view on map

DTC: Buses Available on NON-AC , AC and

stopSelectedHari Nagar Village



Delhi Route Planner Ver 1.0

DTC Customer care

Customer Care : DTC Service

HeadOffice Address

DTC Bhawan
ITO, IP Estate
New Delhi - 110002, India
Contact No. - 011-233370236
- 011-23315085

Call Customer care

24Hr Help Line : +91-11-41400400
Mobile Users : +91-11-23370236
www.dtc.nic.in

Emergency Calls

Fire | Ambulance | Police

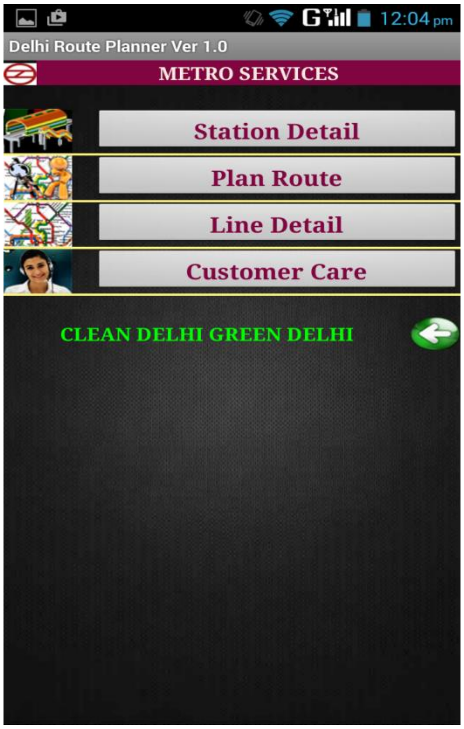
Hint: Click Text or Button to Call

Fig 4:Stop Detail screen:

Here the user can select a stop and gets information about its location and route.

Fig 5:DTC customer care

From this user can get three Helpline Numbers in any emergency.



Delhi Route Planner Ver 1.0

METRO SERVICES

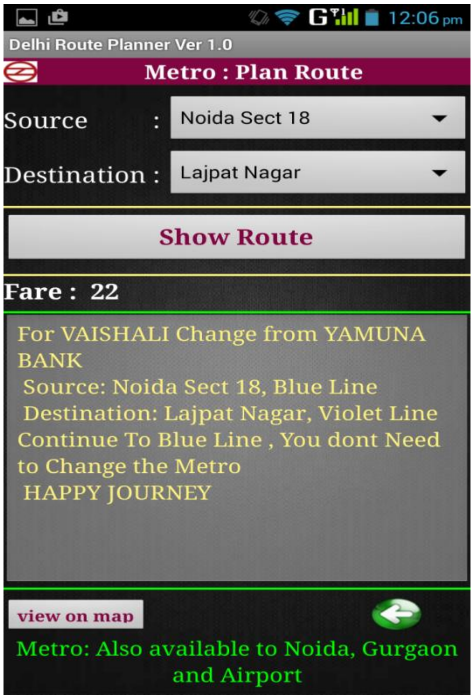
Station Detail

Plan Route

Line Detail

Customer Care

CLEAN DELHI GREEN DELHI



Delhi Route Planner Ver 1.0

Metro : Plan Route

Source : Noida Sect 18

Destination : Lajpat Nagar

Show Route

Fare : 22

For VAISHALI Change from YAMUNA BANK
Source: Noida Sect 18, Blue Line
Destination: Lajpat Nagar, Violet Line
Continue To Blue Line , You dont Need to Change the Metro
HAPPY JOURNEY

view on map

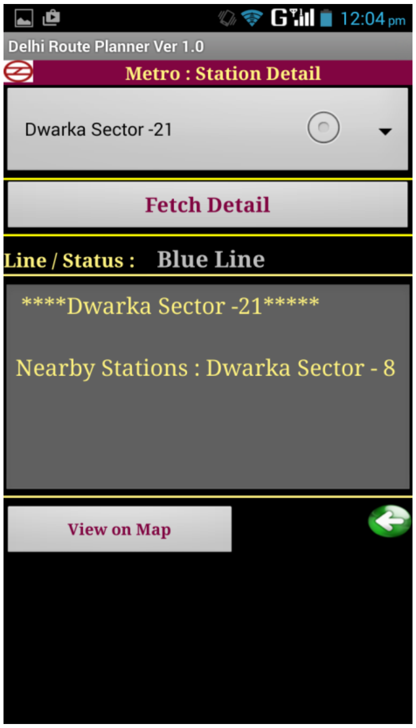
Metro: Also available to Noida, Gurgaon and Airport

Fig 6: Metro service screen

Here the user can select the options which they want to check.

Fig 7:Metro route plan

Showing the details of a route when user select Source and Destination.



12:04 pm

Delhi Route Planner Ver 1.0

Metro : Station Detail

Dwarka Sector -21

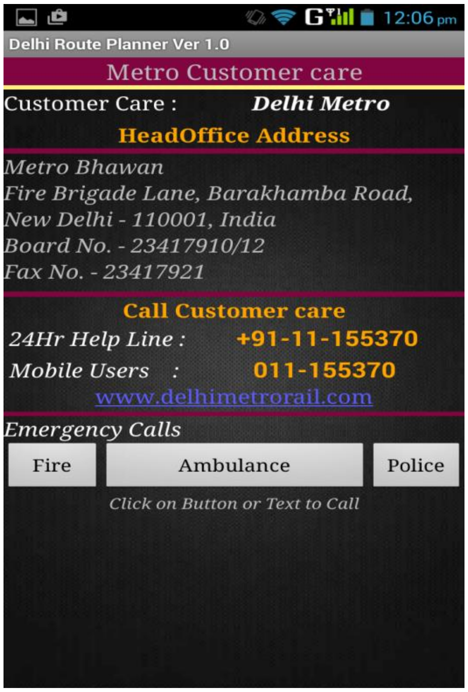
Fetch Detail

Line / Status : Blue Line

****Dwarka Sector -21*****

Nearby Stations : Dwarka Sector - 8

View on Map



12:06 pm

Delhi Route Planner Ver 1.0

Metro Customer care

Customer Care : *Delhi Metro*

HeadOffice Address

Metro Bhawan
Fire Brigade Lane, Barakhamba Road,
New Delhi - 110001, India
Board No. - 23417910/12
Fax No. - 23417921

Call Customer care

24Hr Help Line : +91-11-155370
Mobile Users : 011-155370
www.delhimetrorail.com

Emergency Calls

Fire | Ambulance | Police

Click on Button or Text to Call

Fig:8 Metro Station Details

It'll fetch the details for the metro

Fig:9 Metro customer care

It'll help the user for any service.

IMPLIFICATION FOR FUTURE SCOPE

In future the application can be extended and developed to further enable the users to find out their current location on the map and guide them to the destination.

Also as it is now limited to Delhi it can be extended for the nearby areas and cover the whole Delhi-NCR region and other cities as well.

It can be used to provide the information about various government offices in Delhi. Apart from metro and bus services the app can also be used to gain information about the taxi services available in Delhi and can further be modified to book cabs and travel with comfort in the capital.

Also one more feature that can be probably added is ability to track the location of the user and to provide further guidance on the Google maps. This can be done using the Location manager or GPS feature of the android and a user's location can be easily tracked.

CONCLUSION AND RECOMMENDATION

The android application fulfills the purpose of its creation by providing the basic help and guidance that it was expected to do. As proposed it does makes route planning easy and allows users to choose among from the various options available and choose that best suits them. It is easily accessible and very easy to handle and use. It can be modified and developed further to serve many other purposes such as location tracking and cab bookings etc. its major advantage is that all the information is available to the front end users offline and at easy button clicks.

REFERENCES

[1] The complete Reference Java by Herbert Schildt(7th edition).

[2] Beginning android application development by Wei-Meng Lee.

[3] www.tutorialspoint.com

[4] http://www.javatpoint.com/android-sqlite-tutorial.

[5] http://www.javatpoint.com/android-tutorial.

[6] http://www.javatpoint.com/android-what-where-and-why.

[7] https://www.android.com/history/.

[8] www.**android**central.com/**android-versions**