

**MAJOR PROJECT REPORT**  
**ON**  
**“Delhi Bus Navigator”**

A MAJOR PROJECT REPORT SUBMITTED IN PARTIAL FULLFILLMENT FOR THE  
REQUIREMENT FOR THE AWARD OF THREE YEAR DIPLOMA

**IN**  
**COMPUTER ENGINEERING**

UNDER THE GUIDANCE

OF

**Mr. H S BHATIA**

(PROJECT IN-CHARGE)



**AMBEDKAR INSTITUTE OF TECHNOLOGY**

SHAKARPUR DELHI-92

**SUBMITTED BY:-**

BITTOO

1302051010

**UNDER SUPERVISION OF: -**

MR. H S BHATIA

(H.O.D of COMPUTER ENGG)

# **Delhi Bus Navigator**

A MAJOR PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENT FOR THE AWARD OF THREE YEAR DIPLOMA

IN

COMPUTER ENGINEERING

**AMBEDKAR INSITUTE OF TECHNOLOGY**

**SHAKARPUR, DELHI-92**

SESSION: JAN 2016-JUNE 2016

**SUBMITTED TO:**

Mr. H S BHATIA

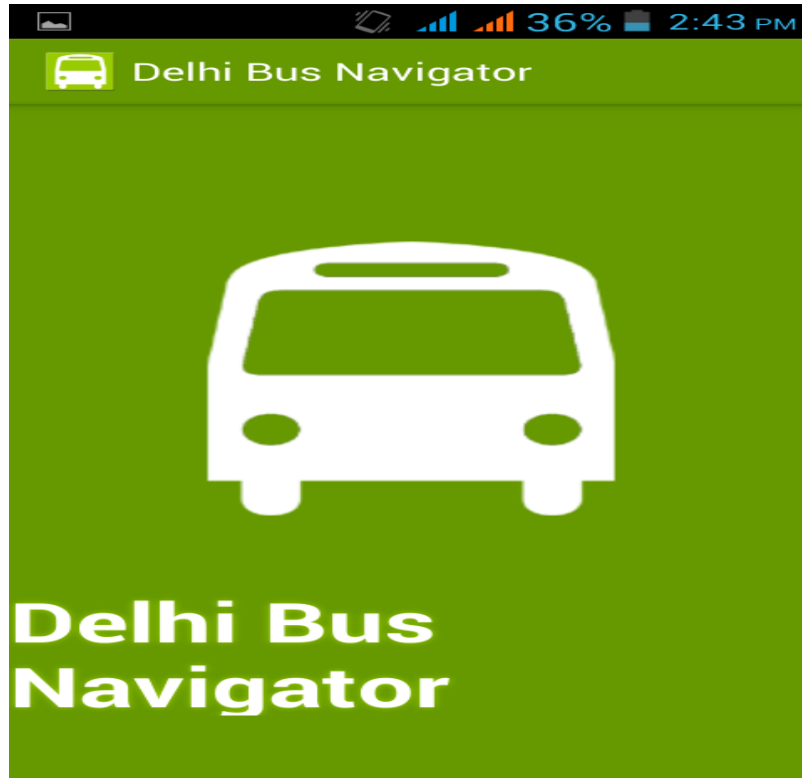
(Project In-Charge)

**SUBMITTED BY:**

BITTOO

(1302051010)

## “DELHI BUS NAVIGATOR”



## **CERTIFICATE**

This is to be certifying that this entitled “**Delhi Bus Navigator**” being submitted by **BITTOO**, B.T.E Roll Number **1302051010** in the partial fulfillment of the requirement for the diploma in COMPUTER ENGG in the department of **COMPUTER ENGG, Ambedkar Institute of Technology** is a record of bonafide work done by him under my supervision and guidance. It is also certified that the dissertation has not been submitted elsewhere for any other degree.

Under the guidance of:-

**Mr. H S BHATIA**

(Project In-Charge)

## **ACKNOWLEDGEMENT**

I take this opportunity to express my gratitude to my project guide **Mr. H S BHATIA** for his endeavor encouragement and support throughout this endeavor. His insight and expertise in this field motivated and supported me during the duration of this project. It is my privilege and honor to have worked under his supervision, his invaluable guidance and helpful discussion in every stage if this project really helped me in materializing this project. Without his constructive direction and invaluable advice, this work would not have been completed

I would also like to take this opportunity to present us sincere regards to **Mr. H S BHATIA** (In-charge of major project), **COMPUTER ENGINEERING**, Ambedkar Institute of Technology Delhi, for this. My gratitude is also extended to all teaching and nonteaching staff for their unwavering encouragement and support in our pursuit for academics. I wish to express my deepest love for my parents & family, whose endless love, understanding, and support during all these years has been the greatest assets in my life.

# **SELF DECLARATION OF STUDENT**

This project is submitted as partial fulfillment of the requirement of DIPLOMA IN **COMPUTER ENGINEERING** of **AMBEDKAR INSTITUTE OF TECHNOLOGY SHAKARPUR NEW DELHI:- 110092** affiliated to BTE DELHI, under the guidance of **Mr. H.S Bhatia**, Head of Computer Engineering Department, **AMBEDKAR INSTITUTE OF TECHNOLOGY, Shakarpur Delhi:-110092**.

I hereby declare that present project report on “**DELHI BUS NAVIGATOR**” is partially original and a bona fide work done by me and wherever the matter has been replicated with or without modification the same has been specially mentioned with the reasons for its usage.

BITTOO

BTE Roll No. 1302051010

Computer Engineering

Final Year (2016)

# **ABSTRACT**

People love to travel places around. Whenever they get time, they visit the neighborhood, some heritage point, educational places, historical places, amusement parks etc.

The problem arises are, where to go? When to go? What is the specialty of that particular place? How to reach there? What all other facilities are available in the nearby area? Most of the times these questions remain answerless.

While planning for outing, deciding where to go is the first and the toughest part of the planning. Sometimes you get the answer and sometimes you don't. Also there is a situation when you are deciding, thoughts occur, is it appropriate to visit that place, today?

And Their a problem arise to where to go and how to go ,Sometime it is more difficult to know destination route for local people through Bus so finding Bus number to go destination place .If you ask people but it is not sufficient that information to want to go.

Sometimes that person forgot about that information told by another person .

The information you need to have is the historic value or USP of the place you are deciding to visit. Best commutation facility, how to reach there.

Generally, you don't get all these information at a single place. You have to google about the place, read some blogs about the reviews and description of that place. Doing such things make it messy and consumes a lot of time to actually know the facts. People demand a solution to these multiple sources of information. They need a single platform to have all such information there, together and organized.

Also they need is that the information should be handy and not to be vague. The information should be easily portable form place to place. And carrying that information should not look awkward.



## **Objective**

This project aims at developing software that allows users to see the information regarding the BUS ROUTE . The project is developed, keeping in mind, the problems of every user have i.e. slow internet connection. To avoid that problem, we have built this application for offline use and user can access it anywhere, even while roaming. User has offline data of the BUS NO. which are categorized according to the type of interests place. People love to travel places by bus. Whenever they get time, they visit the neighborhood, some heritage point, educational places, historical places, amusement parks by the bus .

The problem arises are, where to go? When to go? What is the specialty of that particular take bus ? How to reach there? What all other facilities are available in the different bus route? Most of the times these questions remain answerless.

So this Android App will help to:-

- To provide better service to the users.
- To share information regarding latest happenings at that place.
- The time and money of the users is saved.
- To provide users a platform to access data in a fast and easy way.
- To let user know the specialty of the place.

# CONTENTS

## **TITLE**

---

- 1. Certificate**
- 2. Acknowledgement**
- 3. Abstract**
- 4. Objective**

### **Chapter 1:- Introduction**

- 1.1.Problem Statement**
- 1.2.Proposed solution**
- 1.3.Deliverables**

### **Chapter 2:- Project Description**

- 2.1. Feasibility Studies**
- 2.2 Introduction**
- 2.2. System Specification**
- 2.3. Methodology and Tools used**
- 2.4. Assumption and dependency**
- 2.5. User Characteristics**
- 2.6 Architecture**

### **Chapter 3:- Functionality**

- 3.1. Class Diagram**
- 3.2. Data Flow Diagram**
- 3.3. Use Case Diagram**
- 3.4 Flow Chart**

### **Chapter 4: - Testing**

- 4.1. Testing Methodology**
- 4.2. Unit Testing**
- 4.3. Integration Testing**
- 4.4. System Testing**
- 4.5. Functional Testing**
- 4.6. Test Cases**

### **Chapter 5:- Conclusion and Reference**

- 4.1. Conclusion**
- 4.2. References/Bibliography**

### **Appendix**

- A. Source Code**
- B. Screen Shots**

## 1.1. PROBLEM STATEMENT

People love to travel places around. Whenever they get time, they visit the neighborhood, some heritage point, educational places, historical places, amusement parks etc.

The problem arises are, where to go? When to go? What is the specialty of that particular place? How to reach there? What all other facilities are available in the nearby area? Most of the times these questions remain answerless.

While planning for outing, deciding where to go is the first and the toughest part of the planning. Sometimes you get the answer and sometimes you don't. Also there is a situation when you are deciding, thoughts occur, is it appropriate to visit that place, today?

And Their a problem arise to where to go and how to go ,Sometime it is more difficult to know destination route for local people through Bus so finding Bus number to go destination place .If you ask people but it is not sufficient that information to want to go.

Sometimes that person forgot about that information told by another person .

**There are lots of problem also arise :-**

- No internet connection
- Google map
- Time consuming
- Costly

The information you need to have is the historic value or USP of the place you are deciding to visit. Best commutation facility, how to reach there.

Generally, you don't get all these information at a single place. You have to google about the place, read some blogs about the reviews and description of that place. Doing such things make it messy and consumes a lot of time to actually know the facts. People demand a solution to these multiple sources of information. They need a single platform to have all such information there, together and organized.

Also they need is that the information should be handy and not to be vague. The information should be easily portable form place to place. And carrying that information should not look awkward.

## **1.2. PROPOSED SOLUTION**

### **JUSTIFICATION AND NEED FOR THE SYSTEM**

- To provide better service to the users.
- To share information regarding latest happenings at that place.
- The time and money of the users is saved.
- To provide users a platform to access data in a fast and easy way.
- To let user know the specialty of the place.

### **ADVANTAGES OF THE PROPOSED SYSTEM**

- Reduces the effort and time in gathering the information.
- Provides a complete record.
- Everything at single place.
- No need to use google and other internet apps simultaneously.
- Categories according to type of the place.

## **1.3 DELIVERABLES**

- Table of contents
- Use Case Diagram
- Data flow diagrams
- Test Plan

## **2.1 FEASIBILITY STUDY**

Feasibility study is a preliminary study undertaken before the real work of a project starts to ascertain the likelihood of the project's success. It is an analysis of all possible solutions to a problem and a recommendation on the best solution to use. It involves evaluating how the solution will fit into the corporation.

It is used to determine if the project should get the go-ahead. If the project is to proceed, the feasibility study will produce a project plan and budget estimates for the future stages of development.

### **2.1.1 Technical Feasibility**

The key customer benefits kept in mind while envisaging this architecture were:

- Higher maintainability, extensibility and configurability
- Improved performance and scalability
- Lower Cost of Ownership
- Better productivity
- Lower business risk
- System Performance
- System Interfaces
- Development Processes
- Risk Assessment
- Failure Immunity
- Customer Support
- Security

### **2.1.2 Economical Feasibility**

Cost Benefit Analysis was done in this stage. Following activities were performed during this stage:-

- Each phase of the project was analyzed for the cost involved in it.
- This was calculated based upon resources and infrastructure used.
- Benefits of each phase which were the end products were analyzed and listed.
- Both cost involved and benefits obtained were compared to the details to get the final result.

### **2.1.3 Operational Feasibility**

How well the solution will work in the organization and how the end-users and managers feel about the system.

This is important because a workable solution can be thrown away because of end-user or management doesn't want the system. Therefore usability is another important factor.

**Usability analysis** is often performed with a working prototype of the proposed system. Test of system's user interfaces and measured in how easy they are to learn and to use and how they support the desired productivity levels of users. Easy to learn, use and user satisfaction are other things which are considered here.

### **2.1.4 Other Feasibility Dimensions**

Scheduling Feasibility was one of the dimensions. Measure of how reasonable the project timetable is. Schedule can be mandatory or desirable. It's better to deliver a properly functioning information system later than to deliver an error-prone.

- Measure of how reasonable the project timetable is.
- Work allocation was finalized.
- Time Value for money was analyze

## 2.2 INTRODUCTION

There are buses made available for passengers travelling distances, but not many passengers have complete information about these buses. Complete information namely the number of buses that go to the required destination, bus numbers,, the routes through which the bus would pass the bus to reach.

The proposed system deals with overcoming the problems stated above. The system is an Android application that gives necessary information about all the buses travelling in Delhi. This information overcomes the problems faced in the previously built application “Delhi Bus Navigator”. The platform chosen for this kind of system is Android, reason being Android Operating System has come up on a very large scale and is owned by almost every second person. Also, Android is a user friendly platform, thereby enabling ease of access for all the users.

This application has been developed using IDE(Android Studio) with ADT (Android Development Tools) and Android SDK(Software Development Kit).[3]

There are a number of constraints that need to be satisfied. A few of them may be stated as follows:

1. The phone should not lag each time any route or bus number has been requested for.
2. The platform used must be Android only.
3. All the bus numbers must be stored in the database and retrieved whenever asked for.
4. The application must have information about all the routes in Delhi.
5. The application must be updated with the addition of new buses along with the bus numbers as well as new routes.
6. The application must be user friendly enough for the user to understand it and operate it.
7. The application must not need internet while displaying only the routes. It must be done in offline mode.
8. The Android versions must support the application.

## **2.3 SYSTEM SPECIFICATIONS**

### **2.3.1 Hardware Requirements**

#### **■ Programmer**

- A computer with Core i3 2.3GHz or higher processor, 4GB RAM.
- Minimum 100 GB Hard Disk Space recommended.
- 1024X768 Pixels Screen Resolution for proper viewing of Screens.

#### **■ User**

- A Smart Phone with full features
- Minimum 100 MB storage space and 512 MB RAM
- 1024X768 Pixels Screen Resolution for proper viewing of Screens

### **2.3.2 Software Requirements**

#### **Programmer**

- |                      |   |                      |
|----------------------|---|----------------------|
| ■ Operating System   | : | Windows 7 Home Basic |
| ■ Scripting language | : | ANDROID              |
| ■ IDE & ADT Bundle   | : | ECLIPSE JUNO         |
| ■ Photoshop Tool     | : | ADOBE PHOTOSHOP CS6  |

#### **User**

- |                    |   |                        |
|--------------------|---|------------------------|
| ■ Operating System | : | ANDROID Jelly Bean 4.3 |
|--------------------|---|------------------------|



## 2.3 METHODOLOGY AND TOOLS USED

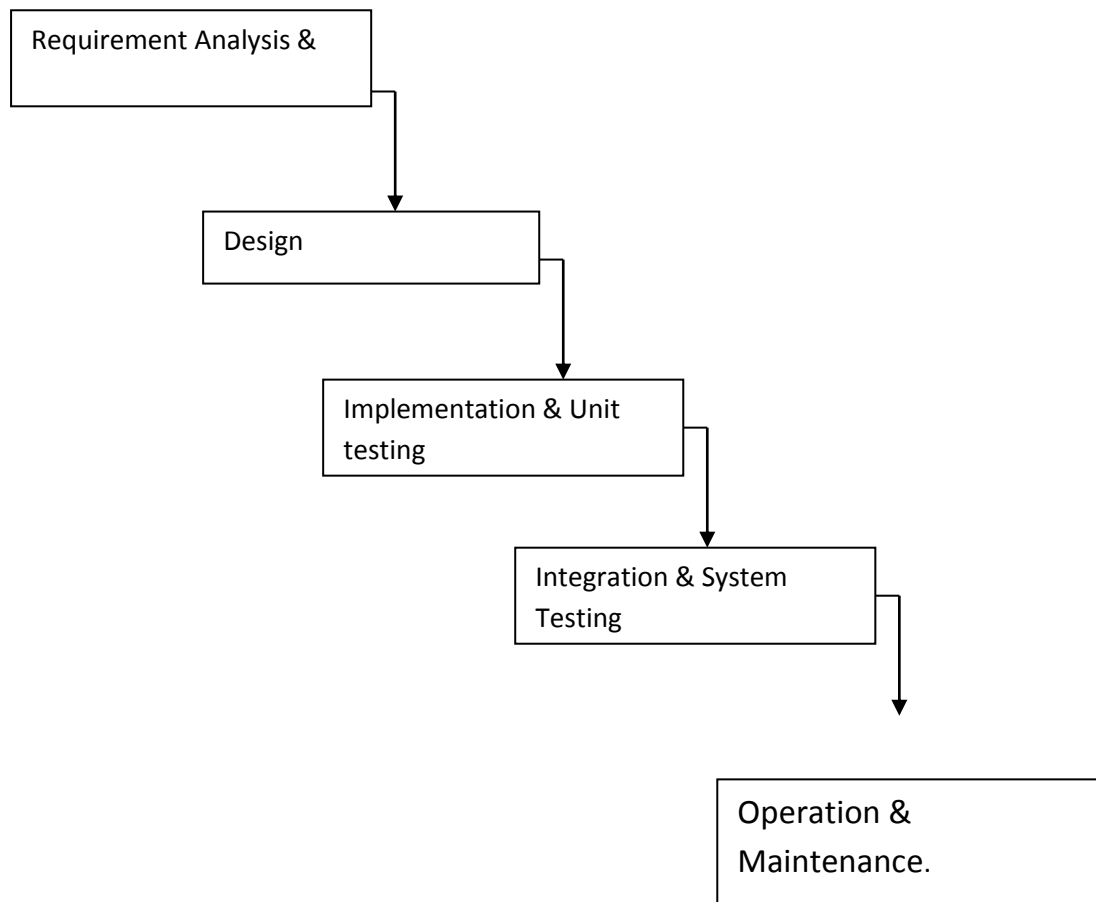
The model that is basically being followed is the WATERFALL MODEL, which states that the phases are organized in a linear order. First of all the feasibility study is done. Once the part is over the Requirement Analysis and Project Planning begins. The design starts after the requirements analysis is complete and the coding begins after the design is complete. Once the coding is completed, the testing is done. In this model the sequence of activities performed in software development project are:

- Requirement Analysis
- Project Planning
- System Design
- Coding
- Unit Testing
- System Integration and Testing

Here the linear ordering of these activities is critical .Output of one phase is the input of another phase. The output of each phase is to be consistent with overall requirement of the system.

Some of the qualities of spiral model are also incorporated like after Interface designing the user was asked to validate the design as per the requirements. Interaction with the user was also done from time to time for identifying further requirements.

WATERFALL Model was being chosen because all the requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.



**Fig. 1 - Various Stages of WATERFALL MODEL**

### **2.3.1 Requirement Analysis & Specification Phase**

The goal of this phase is to understand the exact requirements of the customer and to document them properly. The requirements describe the “what” of the system, not the “how”. This phase produces a large document, written in a natural language, contains a description of what the system will do without describing how it will be done. The resultant document is known as Software Requirement Specification (SRS).

### **2.3.2 Design Phase**

The goal of this phase is to transform the requirements specification into a structure that is suitable for implementation in some programming language.

Here, overall software architecture is defined, and the high level and detailed design work is performed. This work is documented and known as software design description (SDD) document.

### **2.3.3 Implementation & Unit testing Phase**

During this phase, design is implemented. If SDD is complete, the implementation or coding phase proceeds smoothly.

During Testing, the major activities are centered on the examination and modification of the code. Initially small modules are tested in isolation from the rest of the software product.

### **2.3.4 Integration & System testing Phase**

This is a very important phase. Effective testing will contribute to the delivery of higher quality software products, more satisfied users, lower maintenance costs, and more accurate and reliable results. It is a very expensive activity and consumes one third to one-half of the cost of a typical developments project.

As we know, the purpose of unit testing is to determine that each independent module is correctly implemented. This gives a little chance to determine that the interface between modules is also correct, and for this reason integration testing of the entire system is done whereas software is part of the system. This is essential to build confidence in the developers before software is delivered to the customer or released in the market.

### **2.3.5 Operation & Maintenance phase**

Software maintenance is a task that every development group has to face, when the software is delivered to the customer's site, installed and is operational. Therefore, release of software inaugurates the operation and maintenance phase of the life cycle. The time spent and effort required to keep the software operational after is very significant. Despite the fact that it is very important and challenging task; it is routinely the poorly managed headache that nobody wants to face.

Software maintenance is a very broad activity that includes error correction, enhancement of capabilities and optimization. The purpose of this phase is to preserve the value of the software overtime.

### 2.3.6 Technology Used / Tools Selection

The strength of any project depends upon the technology on which the project is based. Today we are living in a world where technologies related to computer science are evolving every day new technologies are taking an edge over the older ones. Every new technology provides new benefits, but only small part of them remains in the competitive world.

## ANDROID CONCEPT

### 2.3.6.1. Java Concept

Java is a programming language originally developed by James Gosling at Sun Microsystems (now part of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere." Java is currently one of the most popular programming languages in use, particularly for client-server web applications.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java and GNU Class path.

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java byte code, instead of directly to platform-specific machine code. Java byte code instructions are analogous to machine code, but are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware. End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets.

Standardized libraries provide a generic way to access host-specific features such as graphics, threading, and networking.

A major benefit of using byte code is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native

executable would. Just-in-Time compilers were introduced from an early stage that compile byte codes to machine code during runtime.

Sun Microsystems officially licensed the Java Standard Edition platform for Linux, Mac OS X, and Solaris. In the past Sun licensed Java to Microsoft but the license expired without renewal. Because Windows does not ship with a Java software platform, a network of third-party vendors and licensees develop them for Windows and other operating system/hardware platforms.

Sun's trademark license for usage of the Java brand insists that all implementations be "compatible". This resulted in a legal dispute with Microsoft after Sun claimed that the Microsoft implementation did not support RMI or JNI and had added platform-specific features of their own. Sun sued in 1997, and in 2001 won a settlement of US\$20 million, as well as a court order enforcing the terms of the license from Sun. As a result, Microsoft no longer ships Java with Windows, and in recent versions of Windows, Internet Explorer cannot support Java applets without a third-party plugin. Sun, and others, have made available free Java run-time systems for those and other versions of Windows.

Platform-independent Java is essential to the Java EE strategy, and an even more rigorous validation is required to certify an implementation. This environment enables portable server-side applications, such as Web services, Java Servlets, and Enterprise JavaBeans, as well as with embedded systems based on OSGI, using Embedded Java environments.

### **2.3.6.2. XML Concept**

**Extensible Markup Language (XML)** is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. It is defined by the W3C's XML 1.0 Specification and by several other related specifications, all of which are free open standards.

The design goals of XML emphasize simplicity, generality and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while many application programming interfaces (APIs) have been developed to aid the processing of XML data.

The XML specification defines an XML document as a well-formed text – meaning that it satisfies a list of syntax rules provided in the specification. Some key points in the fairly lengthy list include:

- The document contains only properly encoded legal Unicode characters
- None of the special syntax characters such as < and & appear except when performing their markup-delineation roles
- The begin, end, and empty-element tags that delimit the elements are correctly nested, with none missing and none overlapping

- The element tags are case-sensitive; the beginning and end tags must match exactly.
- Tag names cannot contain any of the characters `!"#$%&'()*+,-/;<=>?@[\\^`{|}~`, nor a space character, and cannot start with `-`, `.`, or a numeric digit.
- A single "root" element contains all the other elements.

The definition of an *XML document* excludes texts that contain violations of well-formed rules; they are simply not XML. An XML processor that encounters such a violation is required to report such errors and to cease normal processing. This policy, occasionally referred to as "[draconian](#) error handling," stands in notable contrast to the behavior of programs that process [HTML](#), which are designed to produce a reasonable result even in the presence of severe markup errors.<sup>[17]</sup> XML's policy in this area has been criticized as a violation of [Pastel's law](#) ("Be conservative in what you send; be liberal in what you accept").

The XML specification defines a [valid XML document](#) as a [well-formed XML document](#) which also conforms to the rules of a [Document Type Definition](#) (DTD).

## 2.4 ASSUMPTIONS & DEPENDENCIES

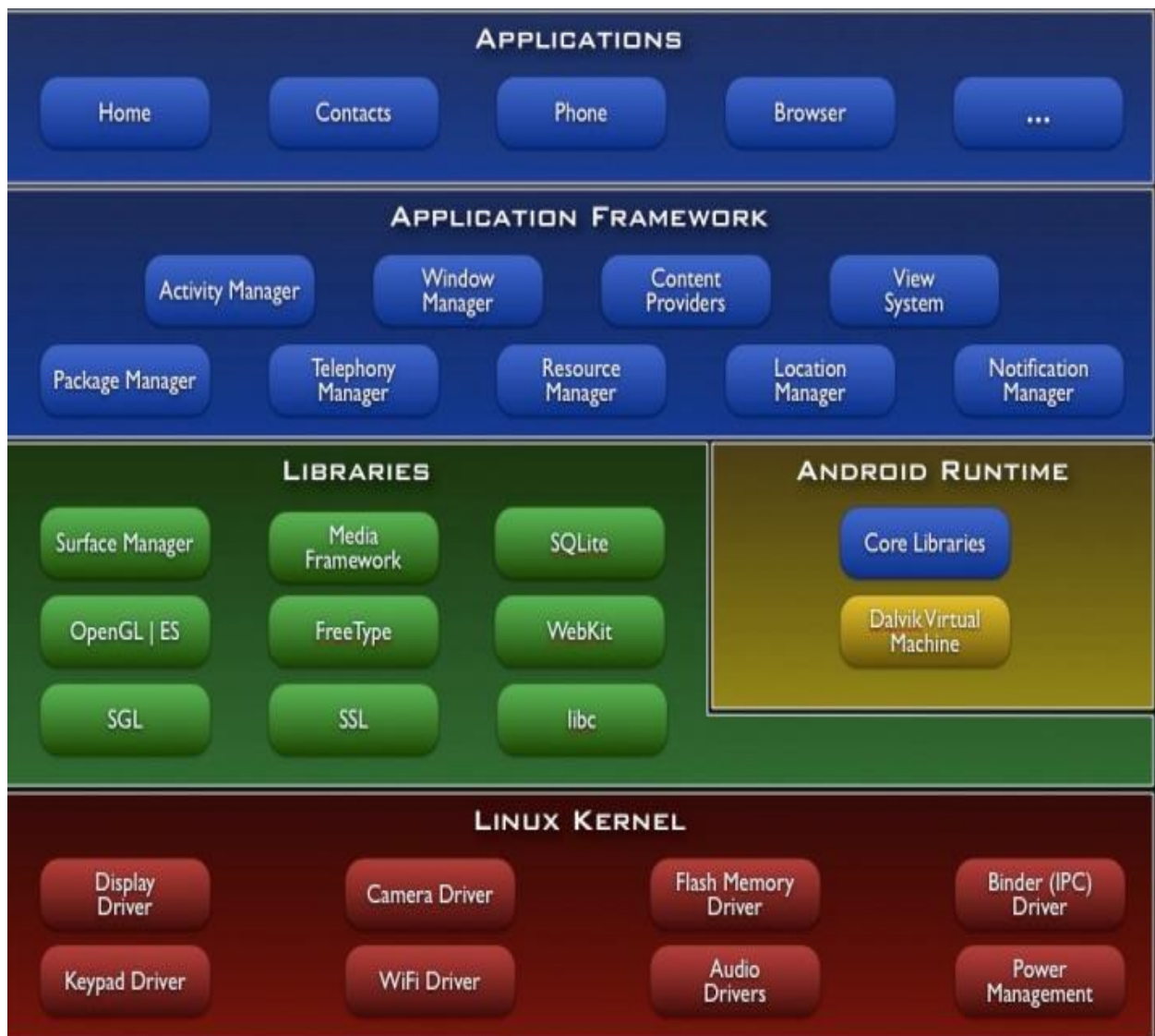
- User must know internet browsing fundamentals
- User must be having internet access.

## 2.5 USER CHARACTERISTICS

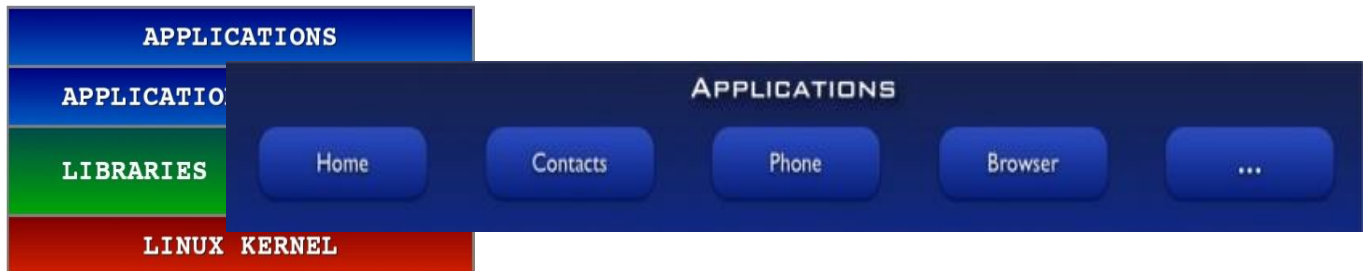
- User must know internet browsing fundamentals
- Registered user must have a valid registration id obtained by registering with the website and must remember his password
- Administrator must remember his secure password

# Architecture

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.



## Android S/W Stack – Application



You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Games etc.

## Android S/W Stack – App Framework (Cont)



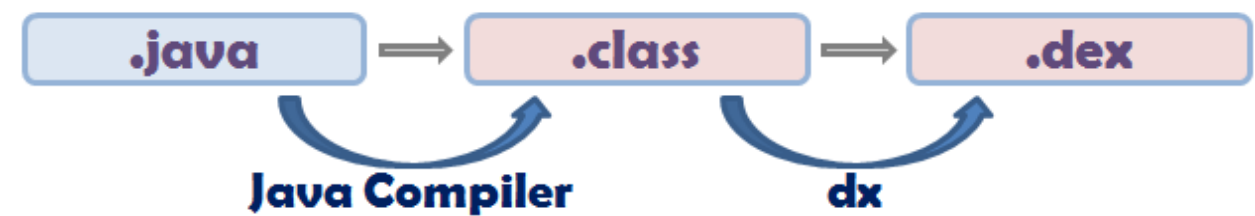
The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.



## Android S/W Stack – Runtime



This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called **Dalvik Virtual Machine** which is a kind of Java Virtual Machine specially designed and optimized for Android. The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine. The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.



## Android S/W Stack – Libraries



On top of Linux kernel there is a set of libraries including open-

**Libraries**

**Explanation**

source  
e  
Web  
brows  
er  
engin  
e  
WebK  
it,

well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

SQLite	It is used to access data published by content providers & includes SQLite DB management classes.
SSL	It is used to provide internet security.
OpenGL	It is used to provide java interface to the OpenGL/ES3D graphics rendering API.
Webkit	It is the browser engine used to display internet content

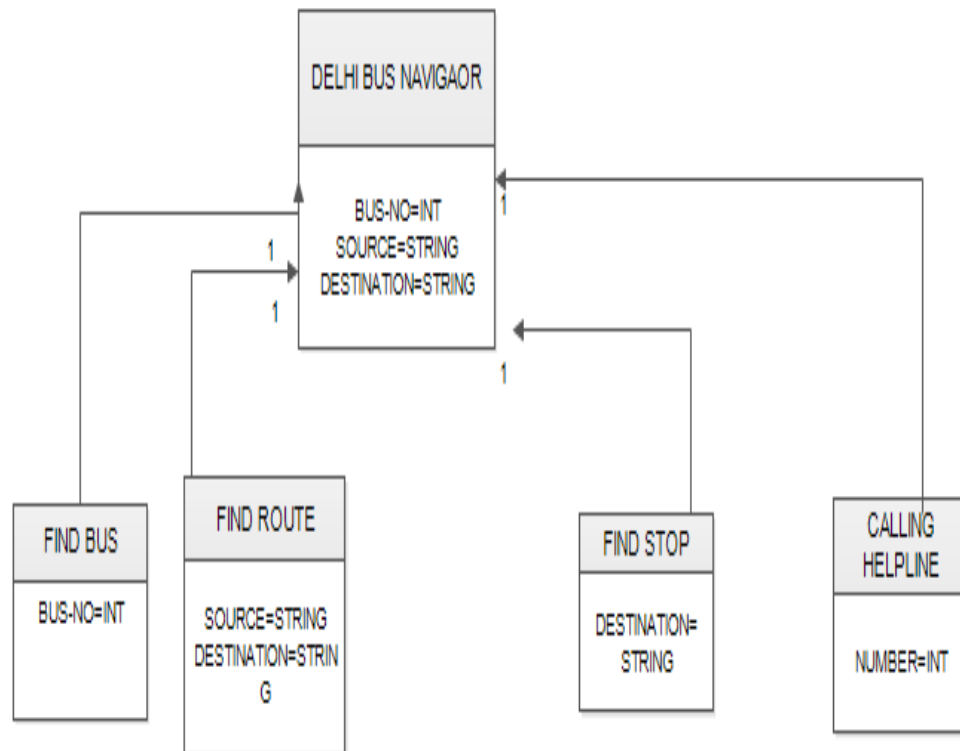
## **Android S/W Stack – Linux Kernel**



At the bottom of the layers is Linux - Linux 2.6 with approximately 115 patches. This provides basic system functionality like process management, memory management, device management like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

### **3.1. CLASS DIAGRAM**

## CLASS DIAGRAM



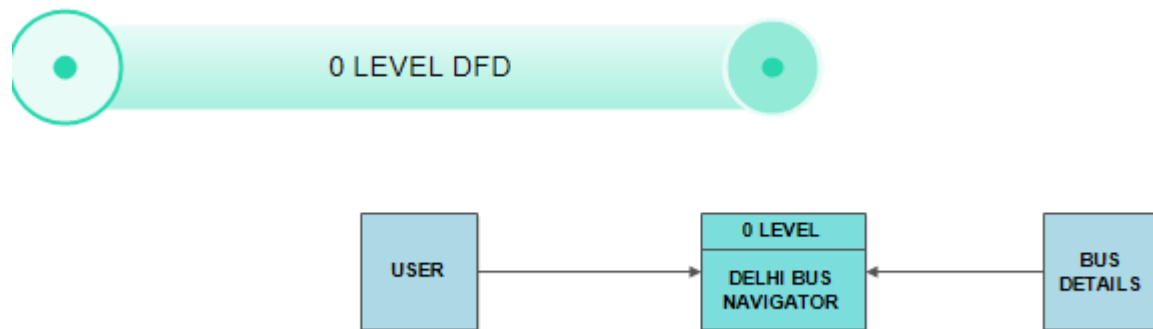
**Fig 2 – CLASS Diagram**

### 3.2. DATA FLOW DIAGRAM

**Data flow diagrams** (DFDs) are one of the three essential perspectives of SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a systems evolution.

- The components of a data flow diagram (DFD) are:
  - Processes
  - Flows
  - Stores
  - Terminators (sometimes called sources and sinks)

### **O – LEVEL DFD of the safe connect**



**Fig 3 – DF Diagram**

## 1 – LEVEL DFD of message transferring

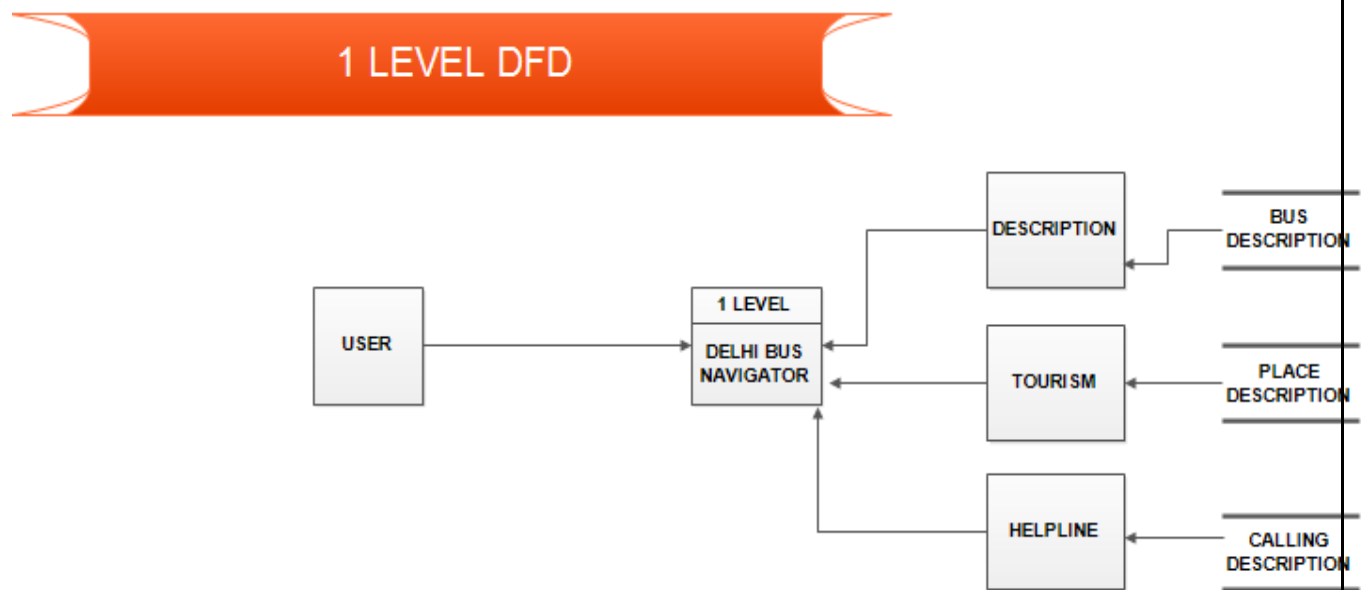
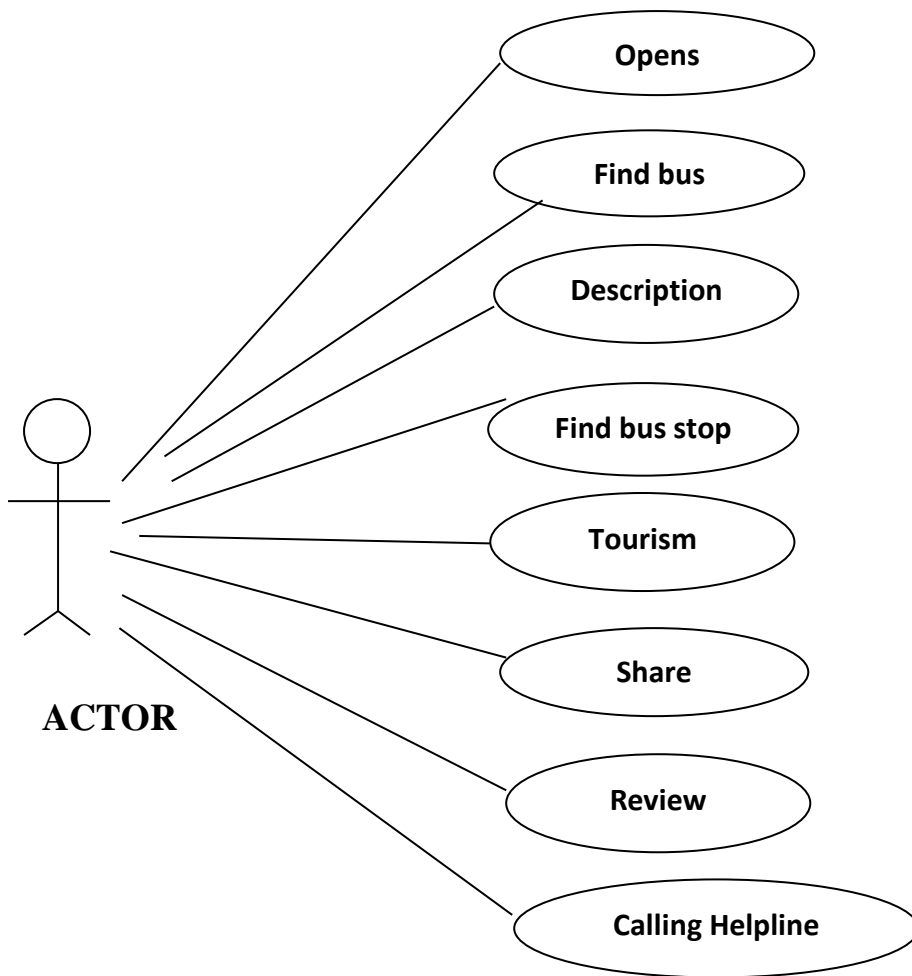


Fig 4 – DFD Diagram

### 3.4. USE CASE DIAGRAM



**Fig 5 – Use Case Diagram**

#### **Use case description**

**Actor:** - User

**Use case:** - Opens, find bus ,Share,About,Reviwe,calling Helpline,find stop.



### 3.5 FLOW CHART

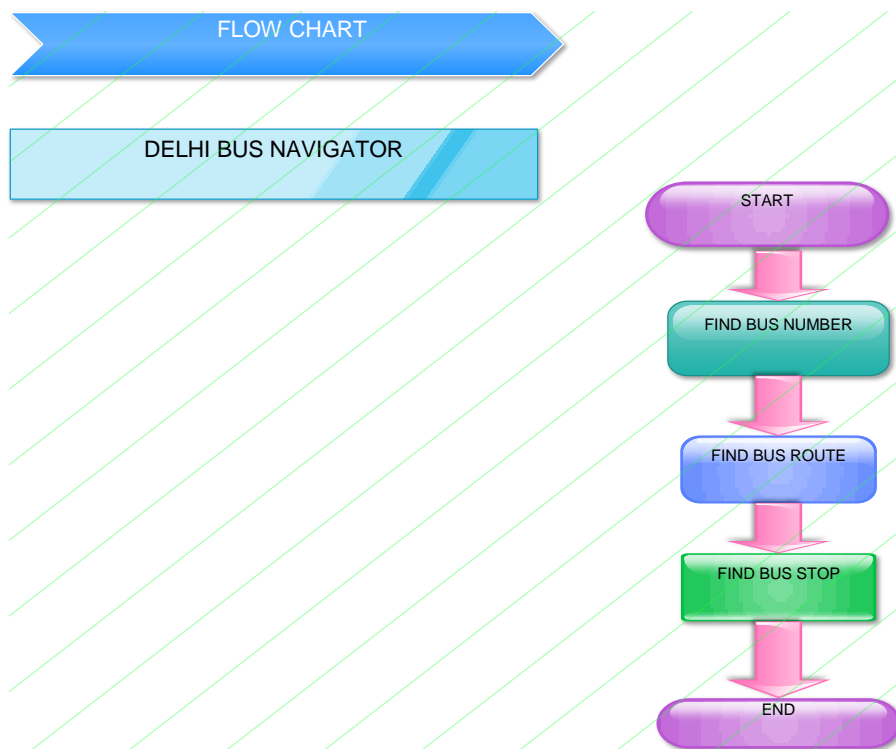


Fig-6 flow chart

## **4.1 TESTING METHODOLOGY**

Software testing is the process used to help identify the correctness, completeness, security and quality of developed computer software. Testing is a set of activities that can be planned in advance and conducted systematically. Testing can never completely establish the correctness of arbitrary computer software. In other words, testing is criticism or comparison that is comparing the actual value with an expected one.

Testing is the process of exercising the software item to detect the differences between its behavior and the desired behavior as stipulated by the requirements specifications- 'what is' and 'what should be'. To achieve a very high standard in quality of delivery, a comprehensive and planned testing will be carried out during project execution. The Testing phase follows the coding and unit-testing phase. Testing a program basically consists of providing the program with set of test inputs or the test cases and observing whether the program behaves in the normal and expected manner. The condition under which the programs behaves in an unexpected manner and deviates from its normal course is noted for debugging and correction.

The objective is to design tests that systematically uncover different classes of errors and do so with a minimum amount of time and effort Secondary benefits include

Demonstration of software functions appearing to be working according to specification. Performance requirements appear to have been met Data collection during testing provides a good indication of software reliability and some indication of software quality.

Testing cannot show the absence of defects, it can only show that software defects are present.

## 4.2 UNIT TESTING

**Unit testing** is a procedure used to validate that the individual modules or units of source code or functions are working properly. Ideally, each test case is independent from the others; mock objects can be used to assist testing a module or a piece of a module in isolation. Unit testing is typically done by the developers and not by end-users.

The purpose of unit testing is to identify and correct as many internal logic errors as possible. Unit tests would be repeatable and may be conducted at any point in the implementation process in accordance with the approved unit test plan for the module I project. The goal for unit testing by developers is to perform selected path testing in which every affected branch is navigated in all possible directions at least once and every affected line of code is executed at least once. The developer would do unit testing at time of coding.

## 4.3 INTEGRATION SYSTEM

**Integration testing** is a procedure used to validate that the modules or units of source code or functions are working properly. Ideally, module is integrated which is independent from the others; mock objects can be used to assist testing a module or a piece of a module in isolation. Integration testing is typically done by the developers and by end-users.

## 4.4 SYSTEM TESTING

System testing is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of Black box testing, and as such, should require no knowledge of the inner design of the code or logic. It requires testing the system as a whole. Testing of the system requirements and the resilience of the software. Integration testing constituents checking of the integrated unit technically and functionally achieves its purpose. Pre-requisite to System Testing - Integration testing is completed and specifications for each node.

## **4.5 FUNCTIONAL TESTING**

The objective of this test is to ensure that each element of the application meets the functional requirements as outlined in system specifications. Functional testing covers the aspects of the system executing functions it is supposed to execute-including user commands, data manipulation, searches and business processes, user screens, and integrations. Functional testing covers the obvious surface type of functions, as well as the back-end operations (such as security and how upgrades affect the system). Before executing the system test cases in full, a limited functional testing will be performed with a subset of system test case where the system will be run on two (or may be more is to be decided) business days and covering end-to-end two (or may be more is to be decided) event types. This is done to verify if all the components of the system is installed properly and to do a basic functionality testing. This will conclude high-level testing. It will be followed by detailed-level tests, which will aim to test the individual processes and data flows.

## **5.1. CONCLSUION**

This application is user friendly. Various categories are provided to help in deciding the type of place user want to visit. Modes of commutation and food junction nearby are also mentioned. Image of the place is also shown to let user have the idea of how that place looks. It provides efficient and quicker service to all users. It provides easy access to the database. It provides flexibility to accommodate future needs. It eliminates duplication of works and provides a convenient and an effective information system

## **5.2. REFERENCES AND BIBILOGRAPHY**

1. For FACEBOOK SHARE - <https://developers.facebook.com/docs/android/share#linkshare>
2. For ImageSswitcher  
<http://developer.android.com/reference/android/widget/ImageSwitcher.html>
3. For Photoshop tutorials - <http://www.adobephotoshoptutorials.com/free/index.php?cat=13>

## APPENDIX A: SOURCE CODE

### Splash Activity.java

```
package com.sas.delhibusnavigator;

import android.app.Activity;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.Menu;
import android.widget.Toast;

public class SplashActivity extends Activity implements Runnable {

    public final String database = "run.db";
    public final static String table = "run";
    public final static String Column_val = "val";

    static SQLiteDatabase check;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        Thread timer = new Thread(this);
        timer.start();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.splash, menu);
        return true;
    }

    @Override
    public void run() {
        try {
            Thread.sleep(1500);
        } catch (InterruptedException e) {
            e.printStackTrace();
        } finally {
            // Intent startingPoint = new
            Intent("com.sas.delhibusnavigator.MAINACTIVITY");
            //startActivity(Intent.createChooser(startingPoint,"check"));
        }
        Intent startingPoint = new Intent("com.sas.delhibusnavigator.MAINACTIVITY");
        startActivity(startingPoint);
    }

    @Override
    protected void onPause() {
        super.onPause();
        finish();
    }

    @Override
    protected void onResume() {
        check = openOrCreateDatabase(database,
        SQLiteDatabase.CREATE_IF_NECESSARY|SQLiteDatabase.OPEN_READWRITE, null);
        check.execSQL("CREATE TABLE IF NOT EXISTS " + table + " (" +
```

```

        Column_val + " TEXT NOT NULL);");
    super.onResume();
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK)
        return true;
    else
        return super.onKeyDown(keyCode, event);
}
}

```

## Route Activity.java

```

package com.sas.delhibusnavigator;

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.database.SQLException;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

import java.io.IOException;
import java.util.LinkedList;

public class RouteActivity extends Activity {

    String bno;

    TextView tvBNo, tvSource, tvDestination;

    ListView lv;

    //AdView av;

    DatabaseHelper myDbHelper;

    LinkedList<String> ll;

    ArrayAdapter<String> aa;

```

```

//private InterstitialAd interstitial;

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

    //interstitial = new InterstitialAd(this, "");
    //interstitial.loadAd(new AdRequest());
    //interstitial.setAdListener(this);

    Intent t = getIntent();
    bno = t.getStringExtra("bno");
    tvBNo = (TextView) findViewById(R.id.etBNo);
    tvSource = (TextView) findViewById(R.id.etSource);
    tvDestination = (TextView) findViewById(R.id.etDestination);
    lv = (ListView) findViewById(R.id.lvRoute);
    //av = (AdView) findViewById(R.id.avRoute);
    myDbHelper = new DatabaseHelper(this);
    try {
        myDbHelper.createDatabase();
    } catch (IOException e) {
        throw new Error("Unable to create database.");
    }
    try {
        myDbHelper.openDatabase();
    } catch (SQLException sqle) {
        throw sqle;
    }
    Cursor c = myDbHelper.getSD(bno);
    c.moveToFirst();

```



```

        tvBNo.setText(bno);

        tvSource.setText(c.getString(1));

        tvDestination.setText(c.getString(2));

        c.close();

        ll = myDbHelper.getStops(bno);

        aa = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, ll);

        lv.setAdapter(aa);

        //av.loadAd(new AdRequest());
    }

    private void setupActionBar() {
        getActionBar().setDisplayHomeAsUpEnabled(true);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.route, menu);

        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                this.finish();

                return true;
        }

        return super.onOptionsItemSelected(item);
    }

    @Override
    protected void onDestroy() {

```

```
        super.onDestroy();  
    }  
}
```

### View Route Activity.java

```
package com.sas.delhibusnavigator;  
  
import android.app.Activity;  
import android.content.Intent;  
import android.database.SQLException;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.AdapterView.OnItemClickListener;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.io.IOException;  
import java.util.LinkedList;  
  
public class ViewRouteActivity extends Activity {  
  
    String source = null, destination = null;  
    TextView tvSource, tvDestination;  
    public static TextView tvLabel;  
    ListView lv;  
    public static LinkedList<String> ll = new LinkedList<String>();  
    ArrayAdapter<String> aa;  
    DatabaseHelper myDbHelper;
```

```

    SQLiteDatabase myDb;

    //AdView av;

    //private InterstitialAd interstitial;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_view_route);

        setupActionBar();

        Intent t = getIntent();

        source = t.getStringExtra("source");

        destination = t.getStringExtra("destination");

        myDbHelper = new DatabaseHelper(this);

        try {

            myDbHelper.createDatabase();

        } catch (IOException e) {

            throw new Error("Unable to create database.");

        }

        try {

            myDbHelper.openDatabase();

        } catch (SQLException sqle) {

            throw sqle;

        }

        tvSource = (TextView) findViewById(R.id.tvSource);

        tvDestination = (TextView) findViewById(R.id.tvDestination);

        tvLabel = (TextView) findViewById(R.id.tvLabel);

        lv = (ListView) findViewById(R.id.lvBusRoute);

        ll.clear();

```

```

        lv.setAdapter(null);

        tvSource.setText(source);
        tvDestination.setText(destination);

        findPath(source, destination);

        aa = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, ll);
        lv.setAdapter(aa);
        lv.setOnItemClickListener(new OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long
arg3) {

                String bn = lv.getItemAtPosition(arg2).toString();

                String[] bus = bn.split(" : ");

                Intent i = new Intent(getApplicationContext(), RouteActivity.class);
                i.putExtra("bno", bus[0]);

                startActivity(i);

            }

        });

        //av = (AdView) findViewById(R.id.avViewRoute);
        //av.loadAd(new AdRequest());
    }

    private void setupActionBar() {

        getActionBar().setDisplayHomeAsUpEnabled(true);
    }

    public void findPath(String start, String end) {

        if (myDbHelper.getDirectRoute(start, end)) {

            tvLabel.setText(getResources().getString(R.string.searchSuccess));

            return;
        }
    }

```

```

        Toast.makeText(this, getResources().getString(R.string.searchFail),
        Toast.LENGTH_LONG).show();

        this.finish();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.view_route, menu);

        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                this.finish();

                return true;
        }

        return super.onOptionsItemSelected(item);
    }

    @Override
    protected void onDestroy() {

        super.onDestroy();
    }
}

```

### **About Activity.java**

```

package com.sas.delhibusnavigator;

```

```

import android.app.Activity;
import android.os.Bundle;
import android.support.v4.app.NavUtils;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuItem;
import android.webkit.WebView;

public class AboutActivity extends Activity {

    WebView wv;

    String url;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_about);

        setupActionBar();

        url = "file:///android_asset/about.html";

        wv = (WebView) findViewById(R.id.wvAbout);

        wv.loadUrl(url);

    }

    private void setupActionBar() {

        getActionBar().setDisplayHomeAsUpEnabled(true);

        getActionBar().setSubtitle(getResources().getString(R.string.app_about));

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

```

```

        getMenuInflater().inflate(R.menu.about, menu);

        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                NavUtils.navigateUpFromSameTask(this);

                return true;
        }

        return super.onOptionsItemSelected(item);
    }

    @Override

    public boolean onKeyDown(int keyCode, KeyEvent event) {
        return super.onKeyDown(keyCode, event);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
    }
}

```

### CustomListAdapter.java

```
package com.sas.delhibusnavigator;
```

```

import android.app.Activity;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ArrayAdapter;

import android.widget.TextView;


import java.util.ArrayList;


public class CustomListAdapter extends ArrayAdapter<String> {

    Activity context;

    ArrayList<String> bno = new ArrayList<String>();

    ArrayList<String> source = new ArrayList<String>();

    ArrayList<String> destination = new ArrayList<String>();


    public CustomListAdapter(Activity context, ArrayList<String> bno,
ArrayList<String> source, ArrayList<String> destination) {

        super(context, R.layout.list_item, bno);

        this.context = context;

        this.bno = bno;

        this.source = source;

        this.destination = destination;
    }


    @Override

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

        View rowView = context.getLayoutInflater().inflate(R.layout.list_item, null,
true);

        TextView txtBNo = (TextView) rowView.findViewById(R.id.tvBNo);

        TextView txtSource = (TextView) rowView.findViewById(R.id.tvSource);

        TextView txtDestination = (TextView)
rowView.findViewById(R.id.tvDestination);

```



```

        txtBNo.setText(bno.get(position).toString());

        txtSource.setText(rowView.getResources().getString(R.string.etSource) + " " +
source.get(position).toString());

txtDestination.setText(rowView.getResources().getString(R.string.etDestination) + "
" + destination.get(position).toString());

        return rowView;

    }

}

```

### DisclaimerActivity.java

```

package com.sas.delhibusnavigator;

import android.net.Uri;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.webkit.WebView;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.support.v4.app.NavUtils;

public class DisclaimerActivity extends Activity implements OnItemClickListener{

    public ListView lv1;

```

```

    public String[] helpline = {"Women's Helpline" ,"Toll Free Number", "DTC Central
Control Room"
    ,"DTC East Control Room", "DTC West Control Room", "DTC North Control
Room",
    "DTC South Control Room", "DTC Rural Control Room"};

```

```

@Override

```

```

protected void onCreate(Bundle savedInstanceState) {

```

```

    super.onCreate(savedInstanceState);

```

```

    setContentView(R.layout.activity_disclaimer);

```

```

    setupActionBar();

```

```

    lv1 = (ListView) findViewById(R.id.lv1);

```

```

    lv1.setAdapter(new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, helpline));

```

```

    lv1.setOnItemClickListener((OnItemClickListener) this);}

```

```

private void setupActionBar() {

```

```

    getActionBar().setDisplayHomeAsUpEnabled(true);

```

```

getActionBar().setSubtitle(getResources().getString(R.string.app_disclaimer));

```

```

}

```

```

@Override

```

```

public boolean onCreateOptionsMenu(Menu menu) {

```

```

    getMenuInflater().inflate(R.menu.disclaimer, menu);

```

```

    return true;

```

```

}

```

```

@Override

```

```

public boolean onOptionsItemSelected(MenuItem item) {

```

```

    switch (item.getItemId()) {

```

```

        case android.R.id.home:
            NavUtils.navigateUpFromSameTask(this);

            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    @Override

    public boolean onKeyDown(int keyCode, KeyEvent event) {
        // interstitial.stopLoading();

        return super.onKeyDown(keyCode, event);
    }

    @Override

    protected void onDestroy() {
        super.onDestroy();
    }

    @Override

    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        switch(arg2){

            case 0:
                Intent callIntent = new Intent(Intent.ACTION_CALL);
                callIntent.setData(Uri.parse("tel:1091"));
                startActivity(callIntent);
                break;

            case 1:
                Intent callIntent1 = new Intent(Intent.ACTION_CALL);
                callIntent1.setData(Uri.parse("tel:1800118181"));

```

```
        startActivity(callIntent1);

        break;

    case 2:

        Intent callIntent2 = new Intent(Intent.ACTION_CALL);
        callIntent2.setData(Uri.parse("tel:9818999098"));
        startActivity(callIntent2);

        break;

    case 3:

        Intent callIntent3 = new Intent(Intent.ACTION_CALL);
        callIntent3.setData(Uri.parse("tel:9818999108"));
        startActivity(callIntent3);

        break;

    case 4:

        Intent callIntent4 = new Intent(Intent.ACTION_CALL);
        callIntent4.setData(Uri.parse("tel:9818999105"));
        startActivity(callIntent4);

        break;

    case 5:

        Intent callIntent5 = new Intent(Intent.ACTION_CALL);
        callIntent5.setData(Uri.parse("tel:9871383511"));
        startActivity(callIntent5);

        break;

    case 6:

        Intent callIntent6 = new Intent(Intent.ACTION_CALL);
        callIntent6.setData(Uri.parse("tel:9871383525"));
        startActivity(callIntent6);

        break;

    case 7:

        Intent callIntent7 = new Intent(Intent.ACTION_CALL);
        callIntent7.setData(Uri.parse("tel:9871383523"));
        startActivity(callIntent7);

        break;

}
```

```

    }

}

EclairMotionEvent.java

package com.sas.delhibusnavigator;

import android.view.MotionEvent;

public class EclairMotionEvent extends WrapMotionEvent {

    protected EclairMotionEvent(MotionEvent event) {
        super(event);
    }

    public float getX(int pointerIndex) {
        return event.getX(pointerIndex);
    }

    public float getY(int pointerIndex) {
        return event.getY(pointerIndex);
    }

    public int getPointerCount() {
        return event.getPointerCount();
    }

    public int getPointerId(int pointerIndex) {
        return event.getPointerId(pointerIndex);
    }
}

```

## PhotoSlideShowActivity.java

```
package com.sas.delhibusnavigator;

import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.view.GestureDetector;
import android.view.GestureDetector.SimpleOnGestureListener;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.animation.AccelerateInterpolator;
import android.view.animation.Animation;
import android.view.animation.TranslateAnimation;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.ViewFlipper;

import java.io.File;

public class PhotoSlideShowActivity extends Activity implements OnClickListener {

    private static final int SWIPE_MIN_DISTANCE = 120;
    private static final int SWIPE_MAX_OFF_PATH = 250;
    private static final int SWIPE_THRESHOLD_VELOCITY = 200;
    private GestureDetector gestureDetector;
    View.OnTouchListener gestureListener;
    ViewFlipper imageFrame;
```

```

RelativeLayout slideShowBtn;

Handler handler;

Runnable runnable;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.slide_show);

    imageFrame = (ViewFlipper) findViewById(R.id.imageFrames);


    File parentFolder = new File(Environment.getExternalStorageDirectory()

        .getAbsolutePath() + "/Pictures/CameraSample");

    addFlipperImages(imageFrame, parentFolder);

    gestureDetector = new GestureDetector(new MyGestureDetector());

    gestureListener = new View.OnTouchListener() {

        public boolean onTouch(View v, MotionEvent event) {

            if (event.getAction() >= MotionEvent.ACTION_POINTER_2_DOWN) {

                try {

                    catch (Exception e) {

                        e.getMessage();

                    }

                    return true;

                } else if (gestureDetector.onTouchEvent(event))

                    return true;

                else

                    return false;

            }

        };

        handler = new Handler();

        imageFrame.setOnClickListener(PhotoSlideShowActivity.this);

        imageFrame.setOnTouchListener(gestureListener);

```

```

        slideShowBtn = (RelativeLayout) findViewById(R.id.slideShowBtn);

        slideShowBtn.setOnClickListener(new OnClickListener() {

            @Override

            public void onClick(View arg0) {

                runnable = new Runnable() {

                    @Override

                    public void run() {

                        handler.postDelayed(runnable, 3000);

                        imageFrame.showNext();

                    }

                };

                handler.postDelayed(runnable, 500);

            }

        });

    }

private void addFlipperImages(ViewFlipper flipper, File parent) {

    int imageCount = parent.listFiles().length;

    RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(

        RelativeLayout.LayoutParams.FILL_PARENT,

        RelativeLayout.LayoutParams.FILL_PARENT);

    for (int count = 0; count < imageCount - 1; count++) {

        ImageView imageView = new ImageView(this);

        String file = parent.listFiles()[count].getAbsolutePath();

        Bitmap imbm = BitmapFactory.decodeFile(file);

        imageView.setImageBitmap(imbm);

        imageView.setLayoutParams(params);

        flipper.addView(imageView);

    }

```



```
}
```

```
class MyGestureDetector extends SimpleOnGestureListener {  
    @SuppressWarnings("static-access")  
    @Override  
    public boolean onSingleTapConfirmed(MotionEvent e) {  
        slideShowBtn = (RelativeLayout) findViewById(R.id.slideShowBtn);  
        slideShowBtn.setVisibility(slideShowBtn.VISIBLE);  
        handler.removeCallbacks(runnable);  
        runnable = new Runnable() {  
            @Override  
            public void run() {  
                slideShowBtn.setVisibility(slideShowBtn.INVISIBLE);  
            }  
        };  
        handler.postDelayed(runnable, 2000);  
        return true;  
    }  
  
    @SuppressWarnings("static-access")  
    @Override  
    public boolean onSingleTapUp(MotionEvent e) {  
        slideShowBtn = (RelativeLayout) findViewById(R.id.slideShowBtn);  
        slideShowBtn.setVisibility(slideShowBtn.VISIBLE);  
        handler.removeCallbacks(runnable);  
        runnable = new Runnable() {  
            @Override  
            public void run() {  
                slideShowBtn.setVisibility(slideShowBtn.INVISIBLE);  
            }  
        }  
    }  
}
```

```

    };

    handler.postDelayed(runnable, 2000);

    return true;
}

@Override
public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX,
    float velocityY) {
    try {
        if (Math.abs(e1.getY() - e2.getY()) > SWIPE_MAX_OFF_PATH)
            return false;

        // right to left swipe
        if (e1.getX() - e2.getX() > SWIPE_MIN_DISTANCE
            && Math.abs(velocityX) > SWIPE_THRESHOLD_VELOCITY) {
            handler.removeCallbacks(runnable);

            imageFrame.setInAnimation(inFromRightAnimation());
            imageFrame.setOutAnimation(outToLeftAnimation());

            imageFrame.showNext();
        } else if (e2.getX() - e1.getX() > SWIPE_MIN_DISTANCE
            && Math.abs(velocityX) > SWIPE_THRESHOLD_VELOCITY) {
            handler.removeCallbacks(runnable);

            imageFrame.setInAnimation(inFromLeftAnimation());
            imageFrame.setOutAnimation(outToRightAnimation());

            imageFrame.showPrevious();
        }
    } catch (Exception e) {
        // nothing
    }

    return false;
}
}

```

```

@Override

public void onClick(View view) {

}

private Animation inFromRightAnimation() {

    Animation inFromRight = new TranslateAnimation(

        Animation.RELATIVE_TO_PARENT, +1.2f,

        Animation.RELATIVE_TO_PARENT, 0.0f,

        Animation.RELATIVE_TO_PARENT, 0.0f,

        Animation.RELATIVE_TO_PARENT, 0.0f);

    inFromRight.setDuration(500);

    inFromRight.setInterpolator(new AccelerateInterpolator());

    return inFromRight;

}

private Animation outToLeftAnimation() {

    Animation outtoLeft = new TranslateAnimation(

        Animation.RELATIVE_TO_PARENT, 0.0f,

        Animation.RELATIVE_TO_PARENT, -1.2f,

        Animation.RELATIVE_TO_PARENT, 0.0f,

        Animation.RELATIVE_TO_PARENT, 0.0f);

    outtoLeft.setDuration(500);

    outtoLeft.setInterpolator(new AccelerateInterpolator());

    return outtoLeft;

}

private Animation inFromLeftAnimation() {

    Animation inFromLeft = new TranslateAnimation(

        Animation.RELATIVE_TO_PARENT, -1.2f,

        Animation.RELATIVE_TO_PARENT, 0.0f,

        Animation.RELATIVE_TO_PARENT, 0.0f,

        Animation.RELATIVE_TO_PARENT, 0.0f);

```

```

        inFromLeft.setDuration(500);

        inFromLeft.setInterpolator(new AccelerateInterpolator());

        return inFromLeft;
    }

    private Animation outToRightAnimation() {
        Animation outtoRight = new TranslateAnimation(
            Animation.RELATIVE_TO_PARENT, 0.0f,
            Animation.RELATIVE_TO_PARENT, +1.2f,
            Animation.RELATIVE_TO_PARENT, 0.0f,
            Animation.RELATIVE_TO_PARENT, 0.0f);

        outtoRight.setDuration(500);

        outtoRight.setInterpolator(new AccelerateInterpolator());

        return outtoRight;
    }
}

```

### **TouchImageView.java**

```

package com.sas.delhibusnavigator;

import android.content.Context;

import android.graphics.Bitmap;

import android.graphics.Matrix;

import android.graphics.PointF;

import android.util.Log;

import android.view.MotionEvent;

import android.view.View;

```

```
import android.widget.ImageView;

public class TouchImageView extends ImageView {

    private static final String TAG = "Touch";

    Matrix matrix = new Matrix();

    Matrix savedMatrix = new Matrix();

    // We can be in one of these 3 states

    static final int NONE = 0;

    static final int DRAG = 1;

    static final int ZOOM = 2;

    int mode = NONE;

    // Remember some things for zooming

    PointF start = new PointF();

    PointF mid = new PointF();

    float oldDist = 1f;

    Context context;
```

```

public TouchImageView(Context context) {

    super(context);

    super.setClickable(true);

    this.context = context;

    matrix.setTranslate(1f, 1f);

    setImageMatrix(matrix);

    setScaleType(ScaleType.MATRIX);

    setOnTouchListener(new OnTouchListener() {

        @Override

        public boolean onTouch(View v, MotionEvent rawEvent) {

            WrapMotionEvent event = WrapMotionEvent.wrap(rawEvent);

            // Handle touch events here...

            switch (event.getAction() & MotionEvent.ACTION_MASK) {

                case MotionEvent.ACTION_DOWN:

```

```

        savedMatrix.set(matrix);

        start.set(event.getX(), event.getY());

        Log.d(TAG, "mode=DRAG");

        mode = DRAG;

        break;

    case MotionEvent.ACTION_POINTER_DOWN:

        oldDist = (float) spacing(event);

        Log.d(TAG, "oldDist=" + oldDist);

        if (oldDist > 10f) {

            savedMatrix.set(matrix);

            midPoint(mid, event);

            mode = ZOOM;

            Log.d(TAG, "mode=ZOOM");

        }

        break;

    case MotionEvent.ACTION_UP:

        int xDiff = (int) Math.abs(event.getX() - start.x);

        int yDiff = (int) Math.abs(event.getY() - start.y);

        if (xDiff < 8 && yDiff < 8){

            performClick();

        }

```

```

        case MotionEvent.ACTION_POINTER_UP:

            mode = NONE;

            Log.d(TAG, "mode=NONE");

            break;

        case MotionEvent.ACTION_MOVE:

            if (mode == DRAG) {

                // ...

                matrix.set(savedMatrix);

                matrix.postTranslate(event.getX() - start.x, event.getY() -
start.y);

            } else if (mode == ZOOM) {

                float newDist = (float) spacing(event);

                Log.d(TAG, "newDist=" + newDist);

                if (newDist > 10f) {

                    matrix.set(savedMatrix);

                    float scale = newDist / oldDist;

                    matrix.postScale(scale, scale, mid.x, mid.y);

                }

            }

            break;

    }

```



```

        setImageMatrix(matrix);

        return true; // indicate event was handled

    }

});

}

public void setImage(Bitmap bm, int displayWidth, int displayHeight) {

    super.setImageBitmap(bm);

    //Fit to screen.

    float scale;

    if ((displayHeight / bm.getHeight()) >= (displayWidth / bm.getWidth())){

        scale = (float)displayWidth / (float)bm.getWidth();

    } else {

        scale = (float)displayHeight / (float)bm.getHeight();

    }

    savedMatrix.set(matrix);

    matrix.set(savedMatrix);

```

```

        matrix.postScale(scale, scale, mid.x, mid.y);

        setImageMatrix(matrix);

        // Center the image

        float redundantYSpace = (float)displayHeight - (scale *
(float)bm.getHeight());

        float redundantXSpace = (float)displayWidth - (scale *
(float)bm.getWidth());

        redundantYSpace /= (float)2;

        redundantXSpace /= (float)2;

        savedMatrix.set(matrix);

        matrix.set(savedMatrix);

        matrix.postTranslate(redundantXSpace, redundantYSpace);

        setImageMatrix(matrix);
    }

    /** Show an event in the LogCat view, for debugging */

```

```

@SuppressWarnings("unused")

private void dumpEvent(WrapMotionEvent event) {

    // ...

    String names[] = { "DOWN", "UP", "MOVE", "CANCEL", "OUTSIDE",

        "POINTER_DOWN", "POINTER_UP", "7?", "8?", "9?" };

    StringBuilder sb = new StringBuilder();

    int action = event.getAction();

    int actionCode = action & MotionEvent.ACTION_MASK;

    sb.append("event ACTION_").append(names[actionCode]);

    if (actionCode == MotionEvent.ACTION_POINTER_DOWN

        || actionCode == MotionEvent.ACTION_POINTER_UP) {

        sb.append("(pid ").append(

            action >> MotionEvent.ACTION_POINTER_ID_SHIFT);

        sb.append(")");

    }

    sb.append("[");

    for (int i = 0; i < event.getPointerCount(); i++) {

        sb.append("#").append(i);

        sb.append("(pid ").append(event.getPointerId(i));

        sb.append(")=").append((int) event.getX(i));

        sb.append(",").append((int) event.getY(i));

```

```

        if (i + 1 < event.getPointerCount())

            sb.append(";");

    }

    sb.append("]");

    Log.d(TAG, sb.toString());

}

/** Determine the space between the first two fingers */

private double spacing(WrapMotionEvent event) {

    // ...

    float x = event.getX(0) - event.getX(1);

    float y = event.getY(0) - event.getY(1);

    return Math.sqrt(x * x + y * y);

}

/** Calculate the mid point of the first two fingers */

private void midPoint(PointF point, WrapMotionEvent event) {

    // ...

    float x = event.getX(0) + event.getX(1);

    float y = event.getY(0) + event.getY(1);

    point.set(x / 2, y / 2);

```

```
    }  
  
}
```

### **TourismActivity.java**

```
package com.sas.delhibusnavigator;  
  
import com.sas.delhibusnavigator.TourismActivity;  
import com.sas.delhibusnavigator.R;  
import com.sas.delhibusnavigator.TouchImageView;  
import com.sas.delhibusnavigator.TourismActivity.ImageAdapter;  
  
import android.app.Activity;  
import android.content.Context;  
import android.content.res.TypedArray;  
import android.os.Bundle;  
import android.view.Gravity;  
import android.view.View;  
import android.view.ViewGroup;  
import android.view.ViewGroup.LayoutParams;  
import android.widget.AdapterView;  
import android.widget.AdapterView.OnItemClickListener;  
import android.widget.BaseAdapter;  
import android.widget.Gallery;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.Toast;  
  
public class TourismActivity extends Activity {  
    Integer[] pics = {R.drawable.akshardham_temple, R.drawable.ansal_plaza,  
        R.drawable.appu_ghar, R.drawable.art_heritage, R.drawable.chandni_chowk,  
        R.drawable.dilli_haat_ina, R.drawable.garhi_lalit_kala_artist_studio,  
        R.drawable.greater_kailash_market, R.drawable.gurudwara_bangla_sahib,  
        R.drawable.humayun_tomb, R.drawable.india_gate,  
        R.drawable.indira_gandhi_memorial, R.drawable.iskcon_temple,
```

```

        R.drawable.jama_masjid , R.drawable.jantar_mantar,
        R.drawable.lodhi_garden , R.drawable.lotus_temple,
        R.drawable.national_zoological_park , R.drawable.nehru_park,
        R.drawable.parliament_house , R.drawable.pragati_maidan,
        R.drawable.purana_qila, R.drawable.qutab_minar ,
        R.drawable.raj_ghat, R.drawable.rajpath ,
        R.drawable.rashtrapati_bhavan, R.drawable.red_fort,
        R.drawable.the_village_gallery, R.drawable.tibet_house };

LinearLayout imageView;

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.tour);

    try {
        // InputStream in = (new URL("www.google.com").openStream());
    } catch (Exception e) {
        e.getMessage();
    }

    Gallery ga = (Gallery) findViewById(R.id.Gallery01);
    ga.setAdapter(new ImageAdapter(this));

    imageView = (LinearLayout) findViewById(R.id.ImageView01);
    ga.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
            long arg3) {

            String details="";

            switch(arg2)
            { case 0 : details="AKSHARDHAM TEMPLE \n\nDescription : The Akshardham
constructed of recent times and was inaugurated temple Bochasanvasi Swaminarayan
Sanstha " +

```

"on the banks of the River Yamuna, Open timings 9 am to 8 pm(Monday closed).The temple stretches over an area of 100 acres and was completed in two years." +

"This temple has awesome musical fountain which started in the evening. ";

break;

case 1 : details="ANSAL PLAZA \n\nDescription : Ansal Plaza offers the customers a world class shopping experience located near South Extensio in South Delhi. " +

"This huge plaza is built on 35 acres of land.An auditorium with a stage is located at the centre while the 45-feet high splendid Ansal Plaza is built around it." +

" The Plaza has a French glass curtain wall that to keep away ultraviolet and other harmful radiation.";

break;

case 2 : details="APPU GHAR \n\nDescription : Appu Ghar was an amusement park located in the Pragati Maidan in Delhi. This was the first amusement park of India, " +

"and was inaugurated by Late Shri Rajiv Gandhi, who was the Prime Minister of India. Appu Ghar closed on 17th Feb 2008. Appu Ghar was opened on November 19, 1984 " +

"and was named from the name 'Appu' which was the mascot of the 1982 Asian Games. It stretched over 15.5 acres of land. ";

break;

case 3 : details="ART HERITAGE \n\nDescription : A respected name in the art world was founded in 1977 by Roshen Alkazi. The gallery holds regular exhibitions throughout " +

"the year of some of the best known artists in India.205, Triveni Kala Sangam, Tansen Marg, New Delhi.Phone.:23718833. Timing: 11.00 am to 7.00 pm .";

break;

case 4 : details="CHANDNI CHOWK \n\nDescription : Chandni Chowk is one of the oldest market places that,it has still retained its charm.Chandni Chowk is located opposite the Red Fort. " +

"At one end of the Chandni Chowk, one which Mosque, been built by the wives of Shah Jahan. At the other end of Chandni Chowk, is the old police station or the Kotwali.It is famous for shopping all types of goods";

break;

case 5 : details="DILLI HAAT INA \n\nDescription : Dilli Haat stands for the variety Indian foods and customs in a single roomy enclosed space of six acres. A permanent and improvised " +

"adaptation of a traditional village haat, it is actually a mixture of a food plaza and craft marketplace, located in the locality of South Delhi. ";

break;

case 6 : details="GARHI LALIT KALA ARTIST STUDIO \n\nDescription : It is a Place worth visiting place if you are interested in the live processes of art.

This studio is a part of and is run, The Lalit Kala Academy. " +

"Well known artists from India and abroad often hold workshops and illustrated lectures here.Kala Kutir, East of Kailash, New Delhi. Phone.:- 26432225 Timing: 10.00 am to 6.00 pm .";

break;

case 7 : details="GREATER KAILASH MARKET \n\nDescription : The Greater Kailash area of South Delhi has it all; lavish cozy homes, elegant dresses, brisk crowd and bountiful markets. " +

"One of the most colorful and posh markets of Delhi, this place is a real paradise. Magnificent show rooms and retail outlets selling designer wears, restaurants & easy accessibility give the " +

"Greater Kailash market the edge.";

break;

case 8 : details="GURUDWARA BANGLA SAHIB \n\nDescription : Gurudwara Bangla Sahib is located next to Gol Dak Khana near the Connaught Place. This place of Sikh worship is open to people of all faiths, castes or creeds. " +

"The premises house a sacred pond in which devotees bathe. They believe that this would wash off their misdeeds and thus allow them to attain peace of mind.";

break;

case 9 : details="HUMAYUN TOMB \n\nDescription :It is located near the crossing of Mathura road and Lodhi road, this magnificent garden tomb is the first substantial example of Mughal architecture in India." +

"It was built in 1565 A.D. nine years after the death of Humayun, by his senior widow Bega Begam. Inside the walled enclosure the most notable features are the garden squares (chaharbagh) with pathways water channels.";

break;

case 10 : details="INDIA GATE \n\nDescription : At the centre of New Delhi stands the 42 m Triomphe like archway in the middle of a crossroad.It commemorates the 70,000 peoples fighting for the British Army during the World War I. " +

"The memorial bears the names of more than 13,516 British and Indian soldiers killed in the Northwestern war of 1919. The monument was dedicated to the nation and known by Amar Jawan Jyoti";

break;

case 11 : details="INDIRA GANDHI MEMORIAL \n\nDescription : Indira Gandhi Memorial Museum was the residence of the former Prime Minister of India. It was later converted into a museum. " +

"One can see the collection of rare photographs of the Nationalist movement, the personal moments of the Nehru-Gandhi family and her childhood. Location : No. 1, Safderjang Road, New Delhi- 110 011. Timing: 9.30am To 5:00 pm Closed: Monday.";

break;

case 12 : details="ISKCON TEMPLE \n\nDescription : The ISKCON Temple is located at Hare Krishna Hill, Sant Nagar, East of Kailash,New Delhi,DL-110065. Phone.:- 011-26235133 and was completed in 1998 as a complex of temples. " +



"This temple has been built on a hilly terrain and is dedicated to the Lord Krishna. A great sense of calm, stillness envelope you once you enter the premises. Many spiritual activities are carried out in the temple premises throughout the year.";

break;

case 13 : details="JAMA MASJID \n\nDescription : Jama Masjid, is located in Chandni Chowk, New Delhi, Phone.:-011-23365358 and it was commissioned to be constructed by the Mughal Emperor Shah Jahan. " +

"It holds the distinction of being one of the biggest and the most well known mosque of Old Delhi. Due to its setting at a very prominent center in Old Delhi, a lot of visitors visit it right through the year. ";

break;

case 14 : details="JANTAR MANTAR \n\nDescription : Jantar Mantar is located at Sansad Marg, Janpath, Connaught Place, New Delhi, DL-110001 Phone:- 011-23365358 Open: All days ,Timings: 6:00 am - 7:00 pm Entry Fee: Rs.5(Indians) ,Rs.100(foreigners)." +

"Jantar Mantar(jantra- instruments, mantra- formulae) was constructed in 1724.";

break;

case 15 : details="LODHI GARDEN \n\nDescription : The Lodhi Gardens is a recreational area in Delhi, located at Amrita Shergill Ln, Lodhi Gardens, Lodi Estate, New Delhi, DL-110003 Timing: 6.00 AM - 7.00 PM and situated between Khan Market and Safdarjung's Lodhi Road. " +

"In the middle of Lodhi and Sayyid aristocratic beautiful gardens is the Bara Gumbad or the 'Big Dome' and Sheesh'mirror dome'";

break;

case 16 : details="LOTUS TEMPLE \n\nDescription : Lotus Temple is situated at Lotus Temple Rd, Shambhu Dayal Bagh, Bahapur, New Delhi, Delhi 110019 Pnone.:- 011-23389326 Timing: 9:30 am - 5:30 pm. It is incredible architectures of the faith. The temple has been constructed to resemble a lotus flower. " +

"The huge lotus flower has been made out of marble, dolomite, cement, and sand. The place is known for its spotlessly clean environment.";

break;

case 17 : details="NATIONAL ZOOLOGICAL PARK \n\nDescription : It is located near the Old Fort, Mathura Rd, New Delhi, DL-110003 TIMINGS: 9:00am - 4:30pm Monday closed, Entrance Fee: Indians: Rs.10(Adults) , Rs.5(Children from 5-12 years) and Foreigners: Rs.50(Adults)." +

"It stretches across 214 acres of land. The park is home to several species of animals as well as various types of vegetation. It offers a natural environment to over 2,000 birds and animals";

break;

case 18 : details="NEHRU PARK \n\nDescription : Nehru Park, Delhi, is a large park situated at Vinay Marg, Chanakyapuri Diplomatic Enclave of New Delhi, DL-110021. Named after India's first Prime Minister, " +

"Jawaharlal Nehru, the park is spread over an area of 80 acres, close to the heart of the city, and was established in 1969.";

break;

case 19 : details="PARLIAMENT HOUSE \n\nDescription : The Parliament House is located at Lok Sabha Marg, Gokul Nagar,Pandit Pant Marg Area, Central Secretariat, New Delhi, DL-110001. " +

"The Parliament House (Sansad Bhavan) is a circular building designed by the British architects Sir Edwin Lutyens and Sir Herbert Baker in 1912-1913";

break;

case 20 : details="PRAGATI MAIDAN \n\nDescription :It is located at Mathura Rd, Pragati Maidan New Delhi,DL-110002.It is a venue for large exhibitions and conventions in New Delhi, " +

"and with 72,000 sq. metres of exhibition space,it is Delhi's largest exhibition centre. It is owned and managed by India Trade Promotion Organization (ITPO), Govt. of India.";

break;

case 21 : details="PURANA QILA \n\nDescription : Purana Qila or the Old Fort is situated at Pragati Maidan,Mathura Rd,New Delhi,Delhi-110003 Phone:- 011-23365358. Purana Qila standing stoically amidst wild greenery." +

"Built on the site of the most ancient of the numerous cities of Delhi, Indraprastha, Purana Qila is roughly rectangular in shape having a circuit of nearly two kilometers.";

break;

case 22 : details="QUTAB MINAR \n\nDescription : Qutab Minar is located near Mehrauli,New Delhi Open Timings: 6:00 am - 6:00 pm.It is a soaring, 73 m-high tower of victory & is made of red sandstone and marble, " +

"built in 1193 by Qutab-ud-din Aibak immediately after the defeat of Delhi's last Hindu kingdom. The tower has five distinct storeys, each marked by a projecting balcony and tapers from a 15 m diameter at the base to just 2.5 m at the top. ";

break;

case 23 : details="RAJ GHAT \n\nDescription : Raj Ghat is located on the banks of Yamuna River Open timing: 5:30 am to 7 pm and was built as a cenotaph for honouring Mahatma Gandhi. There is a grave black marble podium at this site which is the spot of cremation of Mahatma Gandhi done on 31st January 1948";

break;

case 24 : details="RAJPATH \n\nDescription : RajPath is said commissioned India Gate and its surrounding area on Rajpath, RajRoad Open Timing: 12:00 AM - 12:00 PM. It is the traditional avenue of there public Indian Republic day parade passes every year. " +

"It boasts of decorative parks, pools and gardens along extends from the India Gate to Vijay Chowk, and Bhavan and National Stadium at two opposite ends.";

break;

case 25 : details="RASHTRAPATI BHAVAN \n\nDescription : Rashtrapati Bhavan(Presidential Residence) is the official home of the President of India. It is situated in the Raisina Hills,Delhi Phone:- 011-23013287, open Timings 9:00am to 4:00pm(Fri-Sun). " +

"It may refer to only the mansion (the 340-room main building) that has the President's official residence, halls, guest rooms and offices. The

```

main palace building was formerly known as Viceroy's House.";

        break;

        case 26 : details="RED FORT \n\nDescription : The Red sandstone walls
of massive Red Fort (Lal Qila) rise 33-m above the clamour of Old Delhi as a
reminder of the magnificent power and emperors. The pomp of the Mughal walls, built
in 1638. " +

                "The main gate, Lahore Gate, is one of the emotional and
symbolic focal points of the modern Indian nation and on crowd major attracts
Independence Day";

        break;

        case 27 : details="THE VILLAGE GALLERY \n\nDescription : This gallery
has done innovative shows in graphic design, drawings and sketches by Satyajit Ray,
Mono-rints by women inmates of Tihar Jail and Haku Shah's homage to Gandhiji with
straw, " +

                "rags and cotton wool.14, Hauz Khas Village, New Delhi. Phone.:
26853860Timing: 10.30 am to 6.30 pm";

        break;

        case 28 : details="TIBET HOUSE \n\nDescription : The Tibet House is
located at Lodhi Rd,Institutional Area,Lodi Colony,New Delhi,DL-110003 Phone:- 011-
24611515 Open: 9:30 am To 5:30 pm Closed: Sat & Sun and is a brilliant information
offering a quick picture of the Tibetan history. " +

                "The Museum of Tibet house, holds an assortment of Tibetan
artifacts that brought to India by Dalai Lama, when he escaped from the Tibetan
lands.";

        break;

    }

    Toast.makeText(getApplicationContext(),details, Toast.LENGTH_SHORT).show();

    //Toast.makeText(getApplicationContext(),"You have selected picture " + (arg2
+ 1)

    //          + " of Delhi", Toast.LENGTH_SHORT).show();

    try {

        imageView.removeAllViews();

    } catch (Exception e) {

        e.getMessage();

    }

    TouchImageView touchImageView = new TouchImageView(

        TourismActivity.this);

    touchImageView.setImageResource(pics[arg2]);

```

```

        LayoutParams lp=new LayoutParams (LayoutParams.FILL_PARENT,
LayoutParams.FILL_PARENT);

imageView.setGravity(Gravity.CENTER_HORIZONTAL|Gravity.CENTER_VERTICAL);

        touchImageView.setLayoutParams(lp);

        imageView.addView(touchImageView);

    }

});

}

public class ImageAdapter extends BaseAdapter {

    private Context ctx;

    int imageBackground;

    public ImageAdapter(Context c) {

        ctx = c;

        TypedArray ta = obtainStyledAttributes(R.styleable.Gallery1);

        imageBackground = ta.getResourceId(

            R.styleable.Gallery1_android_galleryItemBackground, 1);

        ta.recycle();

    }

    @Override

    public int getCount() {

        return pics.length;

    }

    @Override

    public Object getItem(int arg0) {

```

```

        return arg0;
    }

    @Override
    public long getItemId(int arg0) {

        return arg0;
    }

    @Override
    public View getView(int arg0, View arg1, ViewGroup arg2) {
        ImageView iv = new ImageView(ctx);
        iv.setImageResource(pics[arg0]);
        iv.setScaleType(ImageView.ScaleType.FIT_XY);
        iv.setLayoutParams(new Gallery.LayoutParams(150, 120));
        iv.setBackgroundResource(imageBackground);
        return iv;
    }
}

```

### WrapMotionEvent.java

```

package com.sas.delhibusnavigator;

import android.view.MotionEvent;

public class WrapMotionEvent {
    protected MotionEvent event;

    protected WrapMotionEvent(MotionEvent event) {
        this.event = event;
    }
}

```

```

}

static public WrapMotionEvent wrap(MotionEvent event) {
    try {
        return new EclairMotionEvent(event);
    } catch (VerifyError e) {
        return new WrapMotionEvent(event);
    }
}

public int getAction() {
    return event.getAction();
}

public float getX() {
    return event.getX();
}

public float getX(int pointerIndex) {
    verifyPointerIndex(pointerIndex);
    return getX();
}

public float getY() {
    return event.getY();
}

public float getY(int pointerIndex) {
    verifyPointerIndex(pointerIndex);
    return getY();
}

public int getPointerCount() {

```

```

        return 1;
    }

    public int getPointerId(int pointerIndex) {
        verifyPointerIndex(pointerIndex);

        return 0;
    }

    private void verifyPointerIndex(int pointerIndex) {
        if (pointerIndex > 0) {
            throw new IllegalArgumentException(
                "Invalid pointer index for Donut/Cupcake");
        }
    }
}

```

### MainActivity.java

```

package com.sas.delhibusnavigator;

import java.io.IOException;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.Locale;
import android.app.ActionBar;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.FragmentTransaction;
import android.content.ContentValues;
import android.content.DialogInterface;
import android.content.Intent;

```

```

import android.database.Cursor;

import android.database.SQLException;

import android.os.Bundle;

import android.speech.RecognizerIntent;

import android.support.v4.app.Fragment;

import android.support.v4.app.FragmentActivity;

import android.support.v4.app.FragmentManager;

import android.support.v4.app.FragmentPagerAdapter;

import android.support.v4.view.ViewPager;

import android.text.Editable;

import android.text.TextWatcher;

import android.view.KeyEvent;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.view.View.OnClickListener;

import android.view.ViewGroup;

import android.view.inputmethod.InputMethodManager;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

import android.widget.ArrayAdapter;

import android.widget.AutoCompleteTextView;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.ListView;

import android.widget.Toast;


public class MainActivity extends FragmentActivity implements ActionBar.TabListener
{

    SectionsPagerAdapter mSectionsPagerAdapter;

```



```

ViewPager mViewPager;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    final ActionBar actionBar = getActionBar();

    actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);

    mSectionsPagerAdapter = new SectionsPagerAdapter(
        getSupportFragmentManager());

    mViewPager = (ViewPager) findViewById(R.id.pager);

    mViewPager.setAdapter(mSectionsPagerAdapter);

    mViewPager.setOnPageChangeListener(new ViewPager.SimpleOnPageChangeListener()
    {

        @Override

        public void onPageSelected(int position) {

            actionBar.setSelectedNavigationItem(position);

        }

    });

    for (int i = 0; i < mSectionsPagerAdapter.getCount(); i++) {

        actionBar.addTab(actionBar.newTab().setText(mSectionsPagerAdapter.getPageTitle(i)).
            setTabListener(this));

    }

    // Toast t=Toast.makeText(getApplicationContext(), "here", Toast.LENGTH_LONG);

    //t.show();

```

```

Intent x = new Intent("com.sas.delhibusnavigator.TUTORIALACTIVITY");

String query = "SELECT * FROM " + SplashActivity.table;

Cursor c = SplashActivity.check.rawQuery(query, null);

if(c.getCount() == 0) {

    c.close();

    ContentValues values = new ContentValues();

    values.put(SplashActivity.Column_val, "1");

    SplashActivity.check.insert(SplashActivity.table, null, values);

    x.putExtra("val", 1);

    startActivity(x);

} else {

    c.moveToFirst();

    int val = Integer.parseInt("" + c.getString(0));

    if (val <= 2) {

        x.putExtra("val", val);

        startActivity(x);

    }

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

    getMenuInflater().inflate(R.menu.main, menu);

    return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {

        case R.id.rate:

            Toast.makeText(this, "Rate Us", Toast.LENGTH_LONG).show();


```

```

        return true;

    case R.id.share:

        Intent localIntent = new Intent("android.intent.action.SEND");

        localIntent.setType("text/plain");

        localIntent.putExtra("android.intent.extra.SUBJECT", "Delhi Bus (DTC) Navigator");

        localIntent.putExtra("android.intent.extra.TEXT", "New to Delhi? Having difficulty in travelling? Now, travel across DELHI (NCR) through DTC Buses with 'Delhi Bus Navigator' - Get detail about every Bus Route of the DTC Network.");

        startActivity(Intent.createChooser(localIntent, "Share Via"));

        return true;

    case R.id.about:

        Intent about = new Intent("com.sas.delhibusnavigator.ABOUTACTIVITY");

        startActivity(about);

        return true;

    case R.id.disclaimer:

        Intent disclaimer = new Intent("com.sas.delhibusnavigator.DISCLAIMERACTIVITY");

        startActivity(disclaimer);

        return true;

    case R.id.tourism:

        Intent tour = new Intent("com.sas.delhibusnavigator.TOURISMACTIVITY");

        startActivity(tour);

        return true;

    case R.id.exit:

        this.finish();

        return true;

    default:

        return super.onOptionsItemSelected(item);

    }

}

@Override

public boolean onKeyDown(int keyCode, KeyEvent event) {

```

```

        if (keyCode == KeyEvent.KEYCODE_BACK) {

            new AlertDialog.Builder(this)

                .setIcon(R.drawable.ic_action_alert)

                .setTitle(R.string.dialog_title)

                .setMessage(R.string.dialog_message)

                .setPositiveButton(R.string.yes, new DialogInterface.OnClickListener() {

                    public void onClick(DialogInterface dialog, int id) {

                        MainActivity.this.finish();

                    }

                })

                .setNegativeButton(R.string.no, null)

                .setCancelable(false)

                .show();

        }

        return super.onKeyDown(keyCode, event);
    }

    @Override

    protected void onDestroy() {

        super.onDestroy();

    }

    @Override

    public void onTabSelected(ActionBar.Tab tab, FragmentTransaction
fragmentTransaction) {

        mViewPager.setCurrentItem(tab.getPosition());

    }

    @Override

    public void onTabUnselected(ActionBar.Tab tab,

        FragmentTransaction fragmentTransaction) {

    }

```

```

@Override

public void onTabReselected(ActionBar.Tab tab,
                             FragmentTransaction fragmentTransaction) {

}

public class SectionsPagerAdapter extends FragmentPagerAdapter {

    public SectionsPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        Fragment fragment = new DummySectionFragment();
        Bundle args = new Bundle();
        args.putInt(DummySectionFragment.ARG_SECTION_NUMBER, position);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public int getCount() {
        return 3;
    }

    @Override
    public CharSequence getPageTitle(int position) {
        Locale l = Locale.getDefault();
        switch (position) {
            case 0:
                return getString(R.string.title_section1).toUpperCase(l);
            case 1:
                return getString(R.string.title_section2).toUpperCase(l);

```

```

        case 2:

            return getString(R.string.title_section3).toUpperCase();

        }

        return null;
    }
}

public static class DummySectionFragment extends Fragment {

    public static final String ARG_SECTION_NUMBER = "section_number";

    int position;

    View rootView;

    EditText etSearch;

    ListView lv;

    CustomListAdapter clAdapter;

    DatabaseHelper myDbHelper;

    ArrayList<String> bno = new ArrayList<String>();

    ArrayList<String> source = new ArrayList<String>();

    ArrayList<String> destination = new ArrayList<String>();

    AutoCompleteTextView actv1, actv2;

    LinkedList<String> ll;

    ArrayAdapter<String> arrayAdapter;

    Button btn;

    ImageButton ibSrc, ibDest ;

    AutoCompleteTextView actv3;

    ImageButton bFind;

    ListView lvStops;

    LinkedList<String> llSearch, llStops = new LinkedList<String>();

    ArrayAdapter<String> aa1, aa2;

    public DummySectionFragment() {

```

```

    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {

        switch(requestCode) {

            case 100:

                if(resultCode==Activity.RESULT_OK) {

                    ArrayList<String> matches =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

                    String first = matches.get(0);

                    String uc = first.toUpperCase();

                    actv1.setText(uc);


                }

                break;

            case 150:

                if(resultCode==Activity.RESULT_OK) {

                    ArrayList<String> matches =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

                    String first = matches.get(0);

                    String uc = first.toUpperCase();

                    actv2.setText(uc);

                }

                break;

            default:

                break;

        }

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {

        position = getArguments().getInt(ARG_SECTION_NUMBER);

        myDbHelper = new DatabaseHelper(getActivity());

```

```

try {
    myDbHelper.createDatabase();
} catch (IOException e) {
    throw new Error("Unable to create database.");
}

try {
    myDbHelper.openDatabase();
} catch (SQLException sqle) {
    throw sqle;
}

switch (position) {
case 0:
    rootView = inflater.inflate(R.layout.fragment_main_dummy1, container,
false);

    etSearch = (EditText) rootView.findViewById(R.id.etSearch1);
    lv = (ListView) rootView.findViewById(R.id.lv1);
    Cursor c = myDbHelper.getAllBusNumber();
    c.moveToFirst();
    for(int i = 0; i <= c.getCount() - 1; i++) {
        bno.add(c.getString(0));
        source.add(c.getString(1));
        destination.add(c.getString(2));
        c.moveToNext();
    }
    c.close();

    clAdapter = new CustomListAdapter(getActivity(), bno, source,
destination);
    lv.setAdapter(clAdapter);
    lv.setOnItemClickListener(new OnItemClickListener() {

        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
long arg3) {
            Intent i = new Intent("com.sas.delhibusnavigator.ROUTEACTIVITY");
            i.putExtra("bno", "" + lv.getItemAtPosition(arg2));

```



```

        startActivity(i);
    }
});

etSearch.addTextChangedListener(new TextWatcher() {

    @Override
    public void onTextChanged(CharSequence arg0, int arg1, int arg2, int
arg3) {

        try {

            bno.clear();

            source.clear();

            destination.clear();

            clAdapter.clear();

            Cursor c;

            lv.setAdapter(null);

            if (etSearch.getText().toString().contentEquals("")) {

                c = myDbHelper.getAllBusNumber();

                c.moveToFirst();

                for(int i = 0; i <= c.getCount() - 1; i++) {

                    bno.add(c.getString(0));

                    source.add(c.getString(1));

                    destination.add(c.getString(2));

                    c.moveToNext();

                }

                c.close();

            } else {

                String search = etSearch.getText().toString();

                c = myDbHelper.searchBusNumber(search);

                c.moveToFirst();

                for(int i = 0; i <= c.getCount() - 1; i++) {

                    bno.add(c.getString(0));

                    source.add(c.getString(1));

                    destination.add(c.getString(2));

                    c.moveToNext();

                }

            }

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

});

```

```

        }

        c.close();

    }

    clAdapter = new CustomListAdapter(getActivity(), bno, source,
destination);

    lv.setAdapter(clAdapter);

} catch (Exception e) {

    Toast.makeText(getActivity(), "ERROR: " + e.getMessage(),
Toast.LENGTH_SHORT).show();

}

}

@Override

public void beforeTextChanged(CharSequence arg0, int arg1, int arg2,
int arg3) {

}

@Override

public void afterTextChanged(Editable arg0) {

}

});

//av1 = (AdView) rootView.findViewById(R.id.av1);

//av1.loadAd(new AdRequest());

//interstitial.loadAd(new AdRequest());

break;

case 1:

    rootView = inflater.inflate(R.layout.fragment_main_dummy2, container,
false);

    actv1 = (AutoCompleteTextView) rootView.findViewById(R.id.actvSource);

    actv2 = (AutoCompleteTextView)
rootView.findViewById(R.id.actvDestination);

    btn = (Button) rootView.findViewById(R.id.bFind);

    ibSrc =(ImageButton) rootView.findViewById(R.id.speakSource);

    ibDest =(ImageButton) rootView.findViewById(R.id.speakDestination);

    ll = myDbHelper.getAutoFill();

```

```

        arrayAdapter = new ArrayAdapter<String>(getActivity(),
android.R.layout.simple_dropdown_item_1line, ll);

        actv1.setAdapter(arrayAdapter);

        actv2.setAdapter(arrayAdapter);

        btn.setOnClickListener(new OnClickListener() {

            @Override

            public void onClick(View v) {

                InputMethodManager localInputMethodManager = (InputMethodManager)
getActivity().getSystemService("input_method");

                localInputMethodManager.hideSoftInputFromWindow(actv1.getWindowToken(), 0);

                localInputMethodManager.hideSoftInputFromWindow(actv2.getWindowToken(), 0);

                if (actv1.getText().toString().equals("")) &&
actv2.getText().toString().equals("")) {

                    Toast.makeText(getActivity(), "Source and Destination cannot
be blank", Toast.LENGTH_SHORT).show();

                } else if (actv1.getText().toString().equals("")) {

                    Toast.makeText(getActivity(), "Source cannot be blank",
Toast.LENGTH_SHORT).show();

                } else if (actv2.getText().toString().equals("")) {

                    Toast.makeText(getActivity(), "Destination cannot be blank",
Toast.LENGTH_SHORT).show();

                } else if
(actv1.getText().toString().equalsIgnoreCase(actv2.getText().toString())) {

                    Toast.makeText(getActivity(), "Source and Destination cannot
be same", Toast.LENGTH_LONG).show();

                } else {

                    Intent i = new
Intent("com.sas.delhibusnavigator.VIEWROUTEACTIVITY");

                    i.putExtra("source", actv1.getText().toString());

                    i.putExtra("destination", actv2.getText().toString());

                    startActivity(i);

                }

            }

        });

        ibSrc.setOnClickListener(new OnClickListener() {

```

```

        @Override
        public void onClick(View arg0) {

            Intent i1 = new Intent(Intent.ACTION_RECOGNIZE_SPEECH);

            i1.putExtra(Intent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);

            i1.putExtra(Intent.EXTRA_PROMPT, "Speak The Source!!");

            startActivityForResult(i1,100);

        }

    });

    ibDest.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View arg0) {

            Intent i2 = new Intent(Intent.ACTION_RECOGNIZE_SPEECH);

            i2.putExtra(Intent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);

            i2.putExtra(Intent.EXTRA_PROMPT, "Speak The
Destination!!");

            startActivityForResult(i2,150);

        }

    });

    break;
case 2:
    rootView = inflater.inflate(R.layout.fragment_main_dummy3, container,
false);

    actv3 = (AutoCompleteTextView) rootView.findViewById(R.id.actvStop);

```

```

        bFind = (ImageButton) rootView.findViewById(R.id.bFindStop);

        lvStops = (ListView) rootView.findViewById(R.id.lvStops);

        llSearch = myDbHelper.getAutoStop();

        aa1 = new ArrayAdapter<String>(getActivity(),
android.R.layout.simple_dropdown_item_1line, llSearch);

        actv3.setAdapter(aa1);

        aa2 = new ArrayAdapter<String>(getActivity(),
android.R.layout.simple_list_item_1, llStops);

        bFind.setOnClickListener(new OnClickListener() {

            @Override

            public void onClick(View v) {

                String stop = actv3.getText().toString();

                InputMethodManager imm = (InputMethodManager)
getActivity().getSystemService("input_method");

                imm.hideSoftInputFromWindow(actv3.getWindowToken(), 0);

                llStops.clear();

                aa2.clear();

                lvStops.setAdapter(null);

                if (stop.contentEquals(""))

                    Toast.makeText(getActivity(), "Enter Stop Name",
Toast.LENGTH_LONG).show();

                else {

                    llStops = myDbHelper.searchStop(stop);

                    if (llStops.isEmpty())

                        Toast.makeText(getActivity(), "Nothing found",
Toast.LENGTH_LONG).show();

                    else {

                        aa2 = new ArrayAdapter<String>(getActivity(),
android.R.layout.simple_list_item_1, llStops);

                        lvStops.setAdapter(aa2);

                    }

                }

            }

        });

        lvStops.setOnItemClickListener(new OnItemClickListener() {

```

```

        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
long arg3) {

            Intent i = new Intent("com.sas.delhibusnavigator.ROUTEACTIVITY");

            i.putExtra("bno", "" +
lvStops.getItemAtPosition(arg2).toString());

            startActivity(i);

        }

    });

    break;

default:

    break;

}

return rootView;

}

}

```

### DatabaseHelper.java

```

package com.sas.delhibusnavigator;

import android.annotation.SuppressLint;
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;

import java.io.File;

```

```

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.util.LinkedList;

@SuppressLint("SdCardPath")

public class DatabaseHelper extends SQLiteOpenHelper {

    private SQLiteDatabase myDatabase;

    private final Context myContext;

    private static final String DATABASE_NAME = "bus.mp3";

    public final static String DATABASE_PATH =
"/data/data/com.sas.delhibusnavigator/databases/";

    public static final int DATABASE_VERSION = 1;

    Cursor c;

    int countcheck;

    int countd;

    int countf;

    int counts;

    String[] destList = null;

    String destbus = null;

    String[] list;

    String[] list1;

    String[] listf;

    String[] sourceList = null;

    int tempk;

    public DatabaseHelper(Context context) {

        super(context, DATABASE_NAME, null, DATABASE_VERSION);

        this.myContext = context;

    }

```

```

public void createDatabase() throws IOException {

    boolean dbExist = checkDatabase();

    if(dbExist) {

        Log.v("DB Exists", "db exists");

    }

    boolean dbExist1 = checkDatabase();

    if(!dbExist1) {

        this.getReadableDatabase();

        try {

            this.close();

            copyDatabase();

        } catch (IOException e) {

            throw new Error("Error copying Database");

        }

    }

}

private boolean checkDatabase() {

    boolean checkDB = false;

    try {

        String myPath = DATABASE_PATH + DATABASE_NAME;

        File dbFile = new File(myPath);

        checkDB = dbFile.exists();

    } catch (SQLException e) {

    }

    return checkDB;

}

```



```

private void copyDatabase() throws IOException {

    String outFileName = DATABASE_PATH + DATABASE_NAME;

    OutputStream myOutput = new FileOutputStream(outFileName);

    InputStream myInput = myContext.getAssets().open(DATABASE_NAME);

    byte[] buffer = new byte[1024];
    int length;

    while ((length = myInput.read(buffer)) > 0) {
        myOutput.write(buffer, 0, length);
    }

    myInput.close();
    myOutput.flush();
    myOutput.close();
}

public void db_delete() {
    File file = new File(DATABASE_PATH + DATABASE_NAME);
    if (file.exists()) {
        file.delete();
        System.out.println("Delete Database File");
    }
}

public void openDatabase() throws SQLException {
    String myPath = DATABASE_PATH + DATABASE_NAME;

    myDatabase = SQLiteDatabase.openDatabase(myPath, null,
SQLiteDatabase.OPEN_READWRITE);
}

public synchronized void closeDatabase() throws SQLException {
    if (myDatabase != null)

```

```

        myDatabase.close();

        super.close();
    }

    public Cursor getAllBusNumber() {
        String query = "SELECT * FROM BusRoute ORDER BY BusNumber";
        Cursor c = myDatabase.rawQuery(query, null);
        return c;
    }

    public Cursor searchBusNumber(String s) {
        String query = "SELECT * FROM BusRoute WHERE BusNumber LIKE '%" + s + "%'
ORDER BY BusNumber";
        Cursor c = myDatabase.rawQuery(query, null);
        return c;
    }

    public LinkedList<String> getAutoFill() {
        String query = "SELECT DISTINCT Stop FROM BusStop";
        LinkedList<String> ll = new LinkedList<String>();
        Cursor c = myDatabase.rawQuery(query, null);
        c.moveToFirst();
        for(int i = 0; i <= c.getCount() - 1; i++) {
            ll.add(c.getString(0));
            c.moveToNext();
        }
        c.close();
        return ll;
    }

    public Cursor getSD(String bno) {
        String query = "SELECT * FROM BusRoute WHERE BusNumber='" + bno + "'";
        Cursor c = myDatabase.rawQuery(query, null);
        return c;
    }

```

```
}
```

```
public LinkedList<String> getStops(String bno) {  
    String query = "SELECT * FROM BusStop WHERE BusNumber='" + bno + "'";  
    LinkedList<String> item = new LinkedList<String>();  
    Cursor c = myDatabase.rawQuery(query, null);  
    c.moveToFirst();  
    for(int i = 0; i <= c.getCount() - 1; i++) {  
        item.add(c.getString(1));  
        c.moveToNext();  
    }  
    c.close();  
    return item;  
}
```

```
public LinkedList<String> getAutoStop() {  
    String query = "SELECT DISTINCT Stop FROM BusStop";  
    LinkedList<String> ll = new LinkedList<String>();  
    Cursor c = myDatabase.rawQuery(query, null);  
    c.moveToFirst();  
    for(int i = 0; i <= c.getCount() - 1; i++) {  
        ll.add(c.getString(0));  
        c.moveToNext();  
    }  
    c.close();  
    return ll;  
}
```

```
public LinkedList<String> searchStop(String stop) {  
    String query = "SELECT BusNumber FROM BusStop WHERE Stop='" + stop + "'";  
    LinkedList<String> ll = new LinkedList<String>();  
    Cursor c = myDatabase.rawQuery(query, null);  
    c.moveToFirst();
```

```

        for(int i = 0; i <= c.getCount() - 1; i++) {

            ll.add(c.getString(0));

            c.moveToNext();

        }

        c.close();

        return ll;

    }

    public boolean getDirectRoute(String start, String end) {

        String query = "SELECT DISTINCT BusNumber FROM BusStop WHERE BusNumber IN
        (SELECT BusNumber FROM BusStop WHERE Stop='" + start + "' INTERSECT SELECT
        BusNumber FROM BusStop WHERE Stop ='" + end + "') ";

        Cursor c = myDatabase.rawQuery(query, null);

        int i = c.getCount();

        while(true) {

            if (!c.moveToNext()) {

                if (i != 0)

                    break;

                return false;

            }

            String str = c.getString(c.getColumnIndex("BusNumber")) + " : " + start +
            " to " + end;

            ViewRouteActivity.ll.add(str);

        }

        return true;

    }

    @Override

    public void onCreate(SQLiteDatabase db) {

    }

    @Override

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

        if (newVersion > oldVersion) {

```

```

        Log.v("Database Upgrade", "Database version higher than old.");

        db_delete();
    }
}
}

```

### WrapMotionEvent.java

```

package com.sas.delhibusnavigator;

import android.view.MotionEvent;

public class WrapMotionEvent {
    protected MotionEvent event;

    protected WrapMotionEvent(MotionEvent event) {
        this.event = event;
    }

    static public WrapMotionEvent wrap(MotionEvent event) {
        try {
            return new EclairMotionEvent(event);
        } catch (VerifyError e) {
            return new WrapMotionEvent(event);
        }
    }

    public int getAction() {
        return event.getAction();
    }

    public float getX() {
        return event.getX();
    }
}

```

```

    }

    public float getX(int pointerIndex) {
        verifyPointerIndex(pointerIndex);

        return getX();
    }

    public float getY() {
        return event.getY();
    }

    public float getY(int pointerIndex) {
        verifyPointerIndex(pointerIndex);

        return getY();
    }

    public int getPointerCount() {
        return 1;
    }

    public int getPointerId(int pointerIndex) {
        verifyPointerIndex(pointerIndex);

        return 0;
    }

    private void verifyPointerIndex(int pointerIndex) {
        if (pointerIndex > 0) {
            throw new IllegalArgumentException(
                "Invalid pointer index for Donut/Cupcake");
        }
    }
}

```

### AboutActivity.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"

    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    tools:context=".AboutActivity" >


    <WebView

        android:id="@+id/wvAbout"

        android:layout_width="match_parent"

        android:layout_height="0dp"

        android:layout_weight="1" />


</LinearLayout>
```

### DisclaimerActivity.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"

    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    android:weightSum="9"

    tools:context=".DisclaimerActivity" >
```

```

<TextView
    android:id="@+id/etSearch1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"
    android:textSize="25dp"
    android:textColor="#ffffff"
    android:background="#a8d71c"
    android:text="@string/call_helpline" />

<ListView
    android:id="@+id/lv1"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="8" />

</LinearLayout>

```

### MainActivity.XML

```

<android.support.v4.view.ViewPager
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/pager"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" />

```



## TutorialActivity.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"
    android:background="#000000"
    android:alpha="0.7"
    tools:context=".TutorialActivity" >
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_vertical" >
```

```
<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:textSize="25sp"
    android:textColor="#FFFFFF"
    android:textStyle="bold"
    android:gravity="right"
    android:text="@string/tu_top"
    android:shadowColor="#FFFFFF"
    android:shadowDx="0.0"
    android:shadowDy="0.0"
    android:shadowRadius="20" />
```

```
<ImageView
```

```

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:paddingRight="50dp"

        android:src="@drawable/rarrow"

        android:shadowColor="#FFFFFF"

        android:shadowDx="0.0"

        android:shadowDy="0.0"

        android:shadowRadius="20" />

```

```

</LinearLayout>

```

```

<LinearLayout

        android:layout_width="match_parent"

        android:layout_height="wrap_content" >

```

```

    <ImageView

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:paddingTop="50dp"

        android:layout_weight="1"

        android:src="@drawable/uarrow"

        android:shadowColor="#FFFFFF"

        android:shadowDx="0.0"

        android:shadowDy="0.0"

        android:shadowRadius="20" />

```

```

    <ImageView

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:paddingTop="50dp"

        android:layout_weight="1"

        android:src="@drawable/uarrow"

        android:shadowColor="#FFFFFF"

```

```

        android:shadowDx="0.0"

        android:shadowDy="0.0"

        android:shadowRadius="20" />

<ImageView

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:paddingTop="50dp"

    android:layout_weight="1"

    android:src="@drawable/uarrow"

    android:shadowColor="#FFFFFF"

    android:shadowDx="0.0"

    android:shadowDy="0.0"

    android:shadowRadius="20" />

</LinearLayout>

<LinearLayout

    android:layout_width="match_parent"

    android:layout_height="wrap_content" >

    <TextView

        android:layout_width="0dp"

        android:layout_height="wrap_content"

        android:layout_weight="1"

        android:textColor="#FFFFFF"

        android:textSize="20sp"

        android:textStyle="bold"

        android:gravity="center"

        android:text="@string/tu_tab1"

        android:shadowColor="#FFFFFF"

        android:shadowDx="0.0"

        android:shadowDy="0.0"

```

```

        android:shadowRadius="20" />

<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:textStyle="bold"
    android:gravity="center"
    android:text="@string/tu_tab2"
    android:shadowColor="#FFFFFF"
    android:shadowDx="0.0"
    android:shadowDy="0.0"
    android:shadowRadius="20" />

<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:textStyle="bold"
    android:gravity="center"
    android:text="@string/tu_tab3"
    android:shadowColor="#FFFFFF"
    android:shadowDx="0.0"
    android:shadowDy="0.0"
    android:shadowRadius="20" />

</LinearLayout>

<LinearLayout

```

```

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:paddingTop="50dp" >

<ImageView

    android:id="@+id/imageView1"

    android:layout_width="0dp"

    android:layout_height="wrap_content"

    android:layout_weight="1"

    android:src="@drawable/larrow"

    android:shadowColor="#FFFFFF"

    android:shadowDx="0.0"

    android:shadowDy="0.0"

    android:shadowRadius="20" />

<TextView

    android:id="@+id/textView1"

    android:layout_width="0dp"

    android:layout_height="wrap_content"

    android:layout_weight="1"

    android:textColor="#FFFFFF"

    android:textStyle="bold"

    android:textSize="18sp"

    android:gravity="center"

    android:text="@string/tu_msg"

    android:shadowColor="#FFFFFF"

    android:shadowDx="0.0"

    android:shadowDy="0.0"

    android:shadowRadius="20" />

<ImageView

    android:layout_width="0dp"

    android:layout_height="wrap_content"

```

```

        android:layout_weight="1"

        android:src="@drawable/rarrow"

        android:shadowColor="#FFFFFF"

        android:shadowDx="0.0"

        android:shadowDy="0.0"

        android:shadowRadius="20" />

</LinearLayout>

<LinearLayout

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:orientation="vertical"

    android:paddingBottom="50dp"

    android:gravity="bottom|right" >

    <Button

        android:id="@+id/button1"

        android:layout_width="50dp"

        android:layout_height="wrap_content"

        android:text="@string/tu_btn"

        android:shadowColor="#FFFFFF"

        android:shadowDx="0.0"

        android:shadowDy="0.0"

        android:shadowRadius="20" />

    </LinearLayout>

</LinearLayout>

Fragment_main_dummy1.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"

android:paddingLeft="@dimen/activity_horizontal_margin"

android:paddingRight="@dimen/activity_horizontal_margin"

android:paddingTop="@dimen/activity_vertical_margin"

tools:context=".MainActivity$DummySectionFragment" >

```

```

<EditText

    android:id="@+id/etSearch1"

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:hint="@string/etSearch" />

```

```

<ListView

    android:id="@+id/lv1"

    android:layout_width="match_parent"

    android:layout_height="0dp"

    android:layout_weight="1" />

```

```

</LinearLayout>

```

#### Fragment\_main\_dummy2.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:paddingLeft="@dimen/activity_horizontal_margin"

```

```
android:paddingRight="@dimen/activity_horizontal_margin"

android:paddingTop="@dimen/activity_vertical_margin"

android:orientation="vertical"

android:weightSum="9"

tools:context=".MainActivity$DummySectionFragment" >
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"

    android:layout_height="0dp"

    android:layout_weight="1"

    android:weightSum="3"

    android:orientation="horizontal" >
```

```
<AutoCompleteTextView
```

```
    android:id="@+id/actvSource"

    android:layout_weight="2.5"

    android:completionThreshold="3"

    android:layout_width="0dp"

    android:layout_height="wrap_content"

    android:layout_gravity="bottom"

    android:hint="@string/actvSource" />
```



```
<ImageButton

    android:layout_width="0dp"

    android:layout_height="match_parent"

    android:layout_weight="0.5"

    android:background="@drawable/spee"

    android:id="@+id/speakSource"

    android:layout_gravity="bottom"

/>
```

```
</LinearLayout>
```

```
<LinearLayout

    android:layout_width="match_parent"

    android:layout_height="0dp"

    android:layout_weight="1"

    android:weightSum="3"

    android:orientation="horizontal" >
```

```
<AutoCompleteTextView

    android:id="@+id/actvDestination"
```

```
        android:completionThreshold="3"

        android:layout_width="0dp"

        android:layout_height="match_parent"

        android:layout_weight="2.5"

        android:layout_gravity="center"

        android:hint="@string/actvDestination" />
```

```
<ImageButton
```

```
        android:layout_width="0dp"

        android:layout_height="wrap_content"

        android:layout_weight="0.5"

        android:id="@+id/speakDestination"

        android:background="@drawable/spee"

        />
```

```
</LinearLayout>
```

```
<Button
```

```
        android:id="@+id/bFind"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"
```

```

        android:text="@string/bFind" />

</LinearLayout>

```

#### Fragment\_main\_dummy3.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"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity$DummySectionFragment" >

    <LinearLayout

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <AutoCompleteTextView

            android:id="@+id/actvStop"

            android:completionThreshold="1"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:hint="@string/actvStop" />

        <ImageButton

            android:id="@+id/bFindStop"

```

```

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:src="@drawable/ic_action_search" />

</LinearLayout>

<ListView
    android:id="@+id/lvStops"

    android:layout_width="match_parent"

    android:layout_height="0dp"

    android:layout_weight="1" >

</ListView>

</LinearLayout>

List_Item.XML
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:padding="2dp"

    >

    <TableRow>

        <TextView

            android:id="@+id/tvBNo"

            android:layout_width="wrap_content"

            android:layout_height="wrap_content"

            android:textSize="21sp"

            android:textStyle="bold"/>

        </TableRow>

```

```

<TableRow>

    <TextView

        android:id="@+id/tvSource"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_marginLeft="10dp"

        android:textSize="16sp" />

</TableRow>


<TableRow>

    <TextView

        android:id="@+id/tvDestination"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_marginLeft="10dp"

        android:textSize="16sp" />

</TableRow>


</TableLayout>

Slide_show.XML

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

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="fill_parent"

    android:layout_height="fill_parent"

    android:orientation="vertical" >

    <ViewFlipper

        android:id="@+id/imageFrames"

        android:layout_width="fill_parent"

        android:layout_height="fill_parent"

        android:background="@android:drawable/screen_background_dark" >

```

```

</ViewFlipper>

<RelativeLayout

    android:id="@+id/slideShowBtn"

    android:layout_width="fill_parent"

    android:layout_height="40dp"

    android:layout_alignParentBottom="true"

    android:gravity="center" android:visibility="invisible">

    <RelativeLayout

        android:layout_width="fill_parent"

        android:layout_height="fill_parent"

        android:background="#33FFFFFF"

        android:gravity="center"

        android:paddingLeft="1dp"

        android:paddingRight="1dp"

        android:paddingTop="1dp" >

        <RelativeLayout

            android:layout_width="fill_parent"

            android:layout_height="fill_parent"

            android:background="#66000000"

            android:gravity="center" >

            <TextView

                android:layout_width="wrap_content"

                android:layout_height="fill_parent"

                android:gravity="center"

                android:text="Slideshow"

                android:textSize="18dp" />

            </RelativeLayout>

        </RelativeLayout>

    </RelativeLayout>

```

```
</RelativeLayout>
```

### TOUR.XML

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

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:orientation="vertical"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="fill_parent" android:padding="5dp"
```

```
>
```

```
<Gallery
```

```
    android:id="@+id/Gallery01"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="90dp"></Gallery>
```

```
<LinearLayout
```

```
    android:id="@+id/ImageView01"
```

```
    android:orientation="vertical"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="fill_parent">
```

```
</LinearLayout>
```

```
</LinearLayout>
```

## Splash Activity.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"

  android:background="#669900"

  tools:context=".SplashActivity" >
```

```
<ImageView

  android:layout_width="wrap_content"

  android:layout_height="wrap_content"

  android:src="@drawable/splash"

  android:paddingTop="100dp"

  android:layout_gravity="center"

  android:shadowColor="#99CC00"

  android:shadowDx="0.0"

  android:shadowDy="0.0"

  android:shadowRadius="20" />
```

```
<LinearLayout

  android:layout_width="match_parent"

  android:layout_height="match_parent"

  android:orientation="vertical"

  android:gravity="center|bottom"

  android:paddingBottom="50dp" >
```

```
<TextView

  android:layout_width="wrap_content"

  android:layout_height="wrap_content"

  android:paddingTop="50dp"
```



```
    android:layout_gravity="center"
    android:textSize="35sp"
    android:textStyle="bold"
    android:text="@string/app_name"
    android:textColor="#FFFFFF"
    android:shadowColor="#99CC00"
    android:shadowDx="0.0"
    android:shadowDy="0.0"
    android:shadowRadius="8" />
```

<TextView

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingTop="5dp"
    android:textStyle="bold"
    android:text="@string/ver"
    android:textColor="#FFFFFF"
    android:shadowColor="#99CC00"
    android:shadowDx="0.0"
    android:shadowDy="0.0"
    android:shadowRadius="8" />
```

<TextView

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="5dp"
    android:textStyle="bold"
    android:text="@string/dev"
    android:textColor="#FFFFFF"
    android:shadowColor="#99CC00"
    android:shadowDx="0.0"
    android:shadowDy="0.0"
    android:shadowRadius="8" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

## Route Activity.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"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
    tools:context=".RouteActivity" >
```

```
    <LinearLayout
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
```

```
        android:layout_margin="@dimen/activity_vertical_margin"
```

```
        android:orientation="horizontal"
```

```
        android:weightSum="3" >
```

```
            <TextView
```

```
                android:layout_width="0dp"
```

```
                android:layout_height="wrap_content"
```

```
                android:text="@string/bno"
```

```
                android:textSize="18sp"
```

```
                android:textStyle="bold"
```

```
                android:layout_weight="1" />
```

```

<TextView
    android:id="@+id/etBNo"

    android:layout_width="0dp"

    android:layout_height="wrap_content"

    android:textSize="18sp"

    android:layout_weight="2" />

</LinearLayout>

<LinearLayout

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:layout_margin="@dimen/activity_vertical_margin"

    android:orientation="horizontal"

    android:weightSum="3" >

    <TextView

        android:layout_width="0dp"

        android:layout_height="wrap_content"

        android:text="@string/etSource"

        android:textSize="18sp"

        android:textStyle="bold"

        android:layout_weight="1" />

    <TextView

        android:id="@+id/etSource"

        android:layout_width="0dp"

        android:layout_height="wrap_content"

        android:textSize="18sp"

        android:layout_weight="2" />

</LinearLayout>

```

```

<LinearLayout

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:layout_margin="@dimen/activity_vertical_margin"

    android:orientation="horizontal"

    android:weightSum="3" >

    <TextView

        android:layout_width="0dp"

        android:layout_height="wrap_content"

        android:text="@string/etDestination"

        android:textSize="18sp"

        android:textStyle="bold"

        android:layout_weight="1" />

    <TextView

        android:id="@+id/etDestination"

        android:layout_width="0dp"

        android:layout_height="wrap_content"

        android:textSize="18sp"

        android:layout_weight="2" />

</LinearLayout>

<TextView

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:padding="5dp"

    android:background="#000000"

    android:gravity="center"

    android:alpha="0.5"

    android:text="@string/tvList"

    android:textColor="#FFFFFF"

```

```

        android:textStyle="bold" />

<ListView

    android:id="@+id/lvRoute"

    android:layout_width="match_parent"

    android:layout_height="0dp"

    android:layout_weight="1"

    android:listSelector="@android:color/transparent" />

</LinearLayout>

```

## View Route Activity.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"

    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    tools:context=".ViewRouteActivity" >

    <LinearLayout

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:layout_margin="@dimen/activity_vertical_margin"

        android:orientation="horizontal"

        android:weightSum="3" >

        <TextView

            android:layout_width="0dp"

```

```

        android:layout_height="wrap_content"

        android:layout_weight="1"

        android:text="@string/etSource"

        android:textSize="18sp"

        android:textStyle="bold" />

<TextView

    android:id="@+id/tvSource"

    android:layout_width="0dp"

    android:layout_height="wrap_content"

    android:layout_weight="2"

    android:textSize="18sp" />

</LinearLayout>

<LinearLayout

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:layout_margin="@dimen/activity_vertical_margin"

    android:orientation="horizontal"

    android:weightSum="3" >

    <TextView

        android:layout_width="0dp"

        android:layout_height="wrap_content"

        android:layout_weight="1"

        android:text="@string/etDestination"

        android:textSize="18sp"

        android:textStyle="bold" />

    <TextView

        android:id="@+id/tvDestination"

        android:layout_width="0dp"

```

```

        android:layout_height="wrap_content"

        android:layout_weight="2"

        android:textSize="18sp" />

</LinearLayout>

<TextView
    android:id="@+id/tvLabel"

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:padding="@dimen/activity_vertical_margin"

    android:gravity="center"

    android:background="#000000"

    android:alpha="0.6"

    android:textColor="#FFFFFF"

    android:textStyle="bold" />

<ListView
    android:id="@+id/lvBusRoute"

    android:layout_width="match_parent"

    android:layout_height="0dp"

    android:layout_weight="1" />

</LinearLayout>

```

## Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.sas.delhibusnavigator"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="11"
        android:targetSdkVersion="19" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/Theme" >
        <activity
            android:name="com.sas.delhibusnavigator.SplashActivity"
            android:label="@string/app_name"
            android:screenOrientation="portrait"
            >

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

        <activity
            android:name="com.sas.delhibusnavigator.MainActivity"
            android:label="@string/app_name"
            android:screenOrientation="portrait" >
            <intent-filter>
                <action android:name="com.sas.delhibusnavigator.MAINACTIVITY" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.sas.delhibusnavigator.ViewRouteActivity"
            android:label="@string/app_name"
            android:parentActivityName="com.sas.delhibusnavigator.MainActivity"
            android:screenOrientation="portrait" >
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="com.sas.delhibusnavigator.MainActivity" />

            <intent-filter>
                <action android:name="com.sas.delhibusnavigator.VIEWROUTEACTIVITY"

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.sas.delhibusnavigator.RouteActivity"
            android:label="@string/app_name"
            android:parentActivityName="com.sas.delhibusnavigator.MainActivity"
            android:screenOrientation="portrait" >
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="com.sas.delhibusnavigator.MainActivity" />
        </activity>
    </application>
</manifest>
```



```

        <intent-filter>
            <action android:name="com.sas.delhibusnavigator.ROUTEACTIVITY" />

            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
    <activity
        android:name="com.sas.delhibusnavigator.AboutActivity"
        android:label="@string/app_name"
        android:parentActivityName="com.sas.delhibusnavigator.MainActivity"
        android:screenOrientation="portrait" >
        <meta-data
            android:name="android.support.PARENT_ACTIVITY"
            android:value="com.sas.delhibusnavigator.MainActivity" />

        <intent-filter>
            <action android:name="com.sas.delhibusnavigator.ABOUTACTIVITY" />

            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
    <activity
        android:name="com.sas.delhibusnavigator.DisclaimerActivity"
        android:label="@string/app_name"
        android:parentActivityName="com.sas.delhibusnavigator.MainActivity"
        android:screenOrientation="portrait" >
        <meta-data
            android:name="android.support.PARENT_ACTIVITY"
            android:value="com.sas.delhibusnavigator.MainActivity" />

        <intent-filter>
            <action
android:name="com.sas.delhibusnavigator.DISCLAIMERACTIVITY"/>

            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
    <activity
        android:name="com.sas.delhibusnavigator.TourismActivity"
        android:label="@string/app_name"
        android:parentActivityName="com.sas.delhibusnavigator.MainActivity"
        android:screenOrientation="portrait" >
        <meta-data
            android:name="android.support.PARENT_ACTIVITY"
            android:value="com.sas.delhibusnavigator.MainActivity" />

        <intent-filter>
            <action android:name="com.sas.delhibusnavigator.TOURISMACTIVITY" />

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

    <activity
        android:name="com.sas.delhibusnavigator.TutorialActivity"
        android:label="@string/app_name"
        android:theme="@android:style/Theme.Translucent.NoTitleBar" >
        <intent-filter>
            <action android:name="com.sas.delhibusnavigator.TUTORIALACTIVITY"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>
    </activity>
</application></manifest>

```

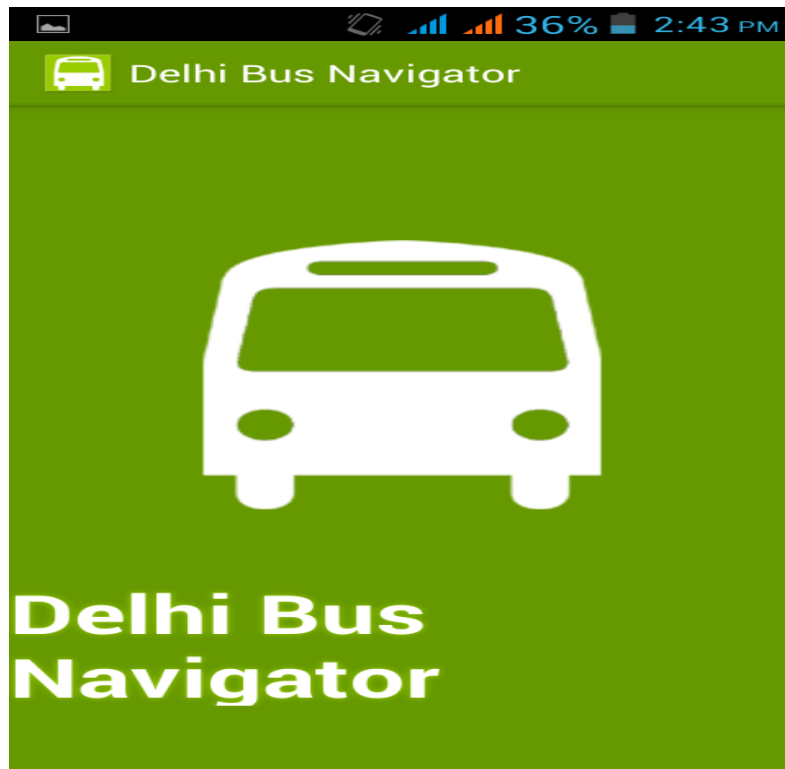
## String file

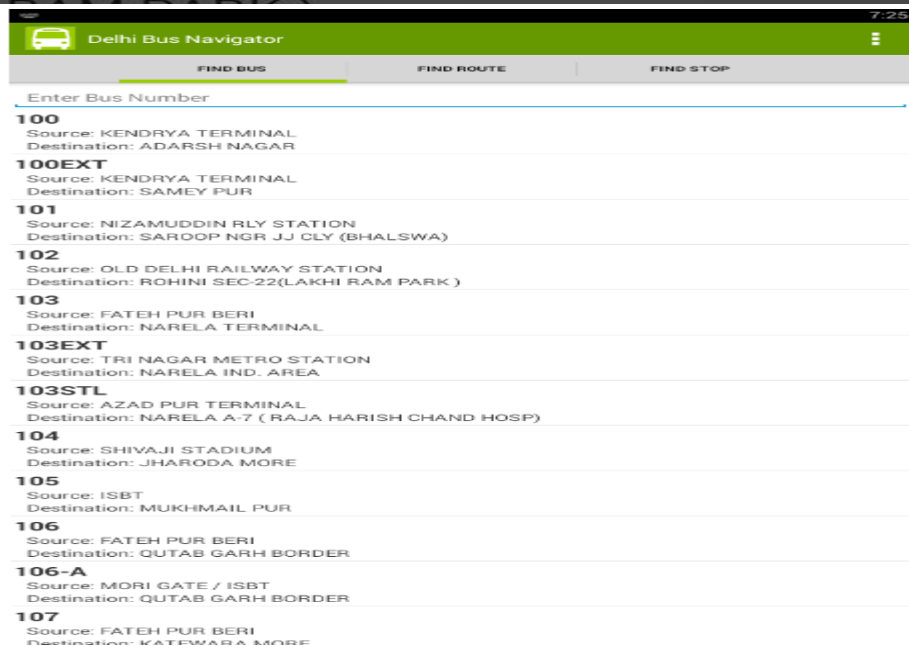
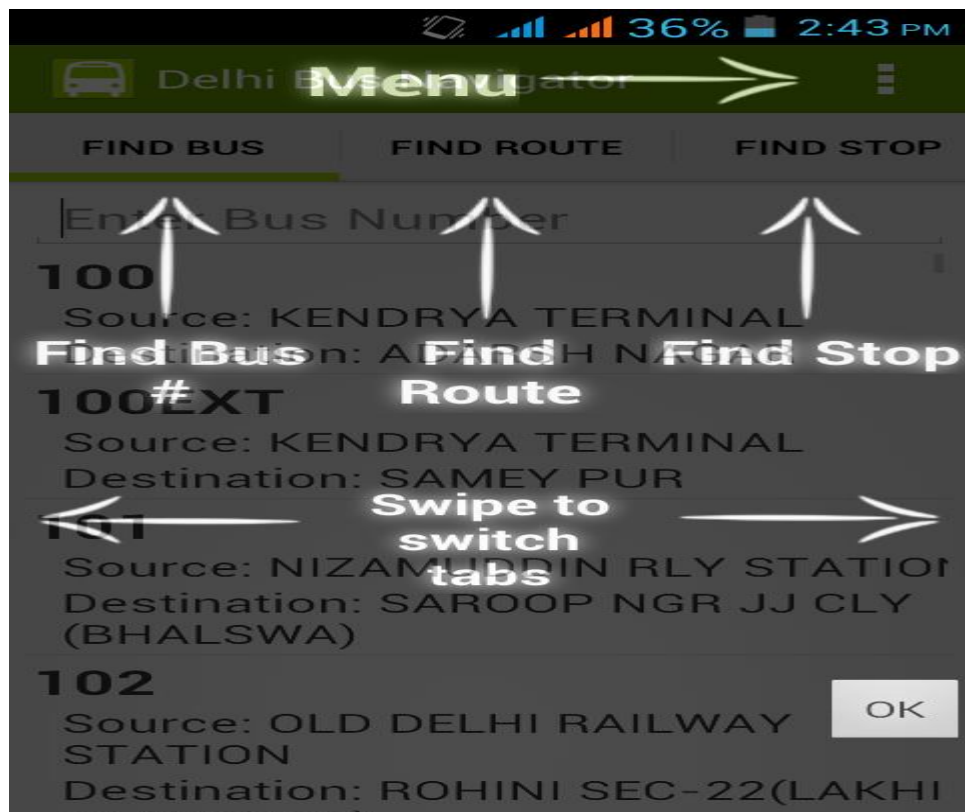
```
<?xml version="1.0" encoding="utf-8"?>
<resources>

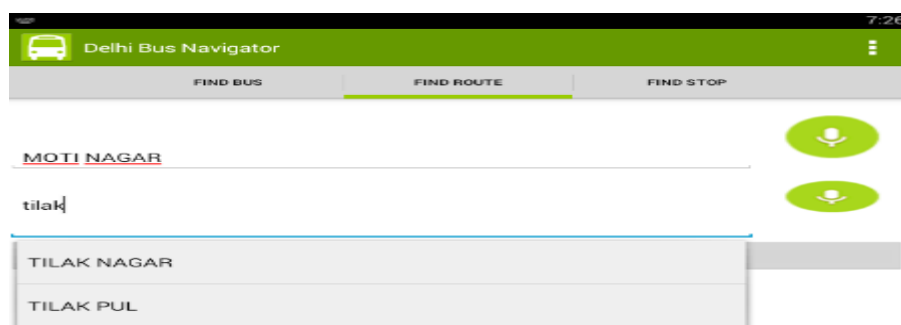
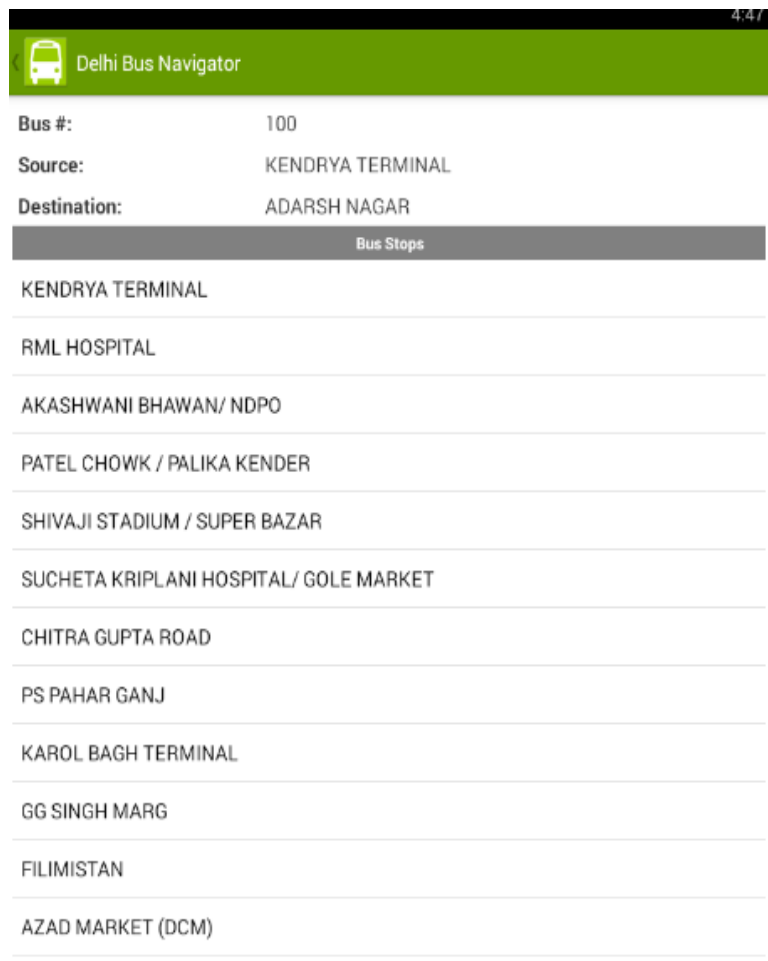
    <string name="app_name">Delhi Bus Navigator</string>
    <string name="ver">Version: 1.0</string>
    <string name="dev"></string>
    <string name="app_about">About Us</string>
    <string name="app_disclaimer">Call Helpline</string>
    <string name="call_helpline">DTC Helpline Numbers</string>
    <string name="action_more">More</string>
    <string name="action_rate">Rate</string>
    <string name="action_share">Share</string>
    <string name="action_about">About Us</string>
    <string name="action_disclaimer">Call DTC Helpline</string>
    <string name="action_tourism">Tourism</string>
    <string name="action_exit">Exit</string>
    <string name="title_section1">Find Bus</string>
    <string name="title_section2">Find Route</string>
    <string name="title_section3">Find Stop</string>
    <string name="etSearch">Enter Bus Number</string>
    <string name="actvSource">Enter Source</string>
    <string name="actvDestination">Enter Destination</string>
    <string name="actvStop">Enter Stop</string>
    <string name="bFind">Find Route</string>
    <string name="bno">Bus #: </string>
    <string name="etSource">Source: </string>
    <string name="etDestination">Destination: </string>
    <string name="source">Source</string>
    <string name="destination">Destination</string>
    <string name="searchSuccess">List of available Bus Routes</string>
    <string name="searchFail">Sorry! No Direct Bus Route found.</string>
    <string name="tvList">Bus Stops</string>
    <string name="dialog_title">Alert</string>
    <string name="dialog_message">Are you sure you want to quit the
application?</string>
    <string name="yes">Quit</string>
    <string name="no">Return</string>
    <string name="tu_top">Menu</string>
    <string name="tu_tab1">Find Bus #</string>
    <string name="tu_tab2">Find Route</string>
    <string name="tu_tab3">Find Stop</string>
    <string name="tu_msg">Swipe to switch tabs</string>
    <string name="tu_btn">OK</string>

</resources>
```

APPENDIX B: ScreenShot







Delhi Bus Navigator

7:27

FIND BUS

FIND ROUTE

FIND STOP

MOTI NAGAR

TILAK NAGAR

Find Route

Delhi Bus Navigator	
Source:	MOTI NAGAR
Destination:	TILAK NAGAR
List of available Bus Routes	
234EXT : MOTI NAGAR to TILAK NAGAR	
308 : MOTI NAGAR to TILAK NAGAR	
726 : MOTI NAGAR to TILAK NAGAR	
726A : MOTI NAGAR to TILAK NAGAR	
801 : MOTI NAGAR to TILAK NAGAR	
807 : MOTI NAGAR to TILAK NAGAR	
810 : MOTI NAGAR to TILAK NAGAR	
816 : MOTI NAGAR to TILAK NAGAR	
817 : MOTI NAGAR to TILAK NAGAR	
818 : MOTI NAGAR to TILAK NAGAR	
823 : MOTI NAGAR to TILAK NAGAR	
828 A : MOTI NAGAR to TILAK NAGAR	
832 : MOTI NAGAR to TILAK NAGAR	
847 : MOTI NAGAR to TILAK NAGAR	
851 : MOTI NAGAR to TILAK NAGAR	
852 : MOTI NAGAR to TILAK NAGAR	
854 : MOTI NAGAR to TILAK NAGAR	

