

Android Application development for importing Vehicles to Ethiopia Tracking system

Zekir Ahmed Awol

Thesis
Business Information Technology
2016



Degree programme

Author Zekir Ahmed Awol	Group or year of entry 2011
Title of report Android Application development for importing Vehicles to Ethiopia Tracking system	Number of pages and appendices 26 + 12
supervisor Hinkula Juha	
<p>Mobile Application have become common in many business sector in the last few decades. The two giant application stores are Google play for Android device platform and Apple store for iOS, others such as Amazon App store, BlackBerry world, windows phone store and Samsung Apps are some sort of distribution platform for mobile device. In many companies, for instance, to serve their customers easily they developed their own application.</p> <p>The purpose of this thesis was to develop Android mobile Application that assist in Tracking vehicle arrival status for Dry Port Service Enterprise in Ethiopia.</p> <p>The thesis has been done not only Front-End which is the application development but also database design and connection of the Back-End or server side. The server side connection is done using PHP web services.</p> <p>The application was made on android studio platform tools. The effectiveness of the application was evaluated and worked out as it is programmed and gradually it will be improved.</p>	
Keywords Android, Ethiopia, Dry Port, vehicle Tracking, ESLSE, DPSE,ESL	

Table of contents

Abbreviations	1
1 Introduction	2
1.1 Company Background	2
1.2 Objective of the project	3
1.3 Scope of the project	4
2 Requirement Analysis	5
2.1 Functional Requirement.....	5
2.2 Software Requirement	5
3 Application Architecture and Design	6
3.1 Application Architecture	6
3.1.1 Front-End side	7
3.1.2 Back-End side	7
3.2 Application Design	8
3.2.1 Use Case Diagram	8
3.2.2 Class Diagram	9
4 Application Development Environment	13
4.1 Installation of Android studio	13
4.2 Creating an Android Project.....	14
5 Implementation	16
5.1 Database Management System.....	16
5.1.1 MySQL for Excel.....	17
5.1.2 Database Design	18
5.2 Graphical user Interface.....	19
5.2.1 Home screen	19
5.2.2 List of Chassis screen	20
5.2.3 Vehicle Arrival Report screen	21
6 Testing	23
6.1 Unit Test.....	23
7 Summary and Conclusion of the thesis	25

7.1 Summary.....	25
7.2 Conclusion	25
8 Future work	26
Bibliography	27
Appendix A- Application source code	29
Appendix A.1. – EvtsMainActivity	29
Appendix A.2. – DryPortLocation	35
Appendix A.3. - ServiceHandler	36
Appendix A.4 – ListChassisActivity	37
Appendix A.5 – VehicleDetailinfo	38
Appendix A.6. – VehicleArrivalInfoActivity	40
Appendix B.1. – Application Android Manifest xml file.....	41
Appendix C.1. - Layout xml file	42
Appendix C.2. – list_chassis xml file	43
Appendix D. PHP Code	44

Abbreviations

ESLSE	Ethiopian Shipping Lines and Logistics Services Enterprise
ESL	Ethiopian Shipping Lines
B/L	Bill of Lading
DPSE	Dry Port Service Enterprise
VIN	Vehicle Identification Number
JSON	JavaScript Object Notation
ADT	Android Development Tools
SDK	Software Development Kit
IDE	Integrated Development Environment
ER	Entity Relationship
SQL	Structured Query Language

1 Introduction

1.1 Company Background

Ethiopian Shipping Lines and Logistics Services Enterprise (ESLSE) formerly known as Ethiopian Shipping Lines (ESL) was established half a century ago (1964) as Share Company with the total capital investment 3,750,00 birr of which 51% of the share had American based company is called Taurus Investment Inc. of Washington DC. In 1969/70 five years, later the government of Ethiopia secured fully and from onwards it operates under the Ethiopian Supervising Authority. (SeaRates LP ,2016.)

In 2011 November, ESLCE restructured to split its services in to three different Enterprise. These sectors are Ethiopian Shipping Lines, Ethiopia Maritime, and Transit services and Ethiopian Dry Port Enterprise. Each Enterprise has their own mandate to take care of organizing, planning, managing, arranging the transit and transporting shipments until delivery to the importers or customers. (ESL ,2016a.)

ESLSE uses the import and export trade in Djibouti port after Ethio-Eritrea war broke out. However, there was many complaining from importers that their goods delay in the arrival of Djibouti port. To overcome the delay problem, to save foreign currency and to avoid the demurrage costs the ESLSE had arranged dry port in two regional state namely Modjo dry port and Semera dry port in Oromia regional state and Afar regional state respectively in 2007. Later after six years Ethiopian Dry Port Service Enterprise (DPSE) established additional two dry ports to minimize the congestion in Modjo dry port. Furthermore, DPSE has set up more dry ports almost in every regional state such as in Gelan Dry Port, Diredewa Dry Port, Kombolcha Dry Port, Mekelle Dry Port, Kality Branch (Comet), Babogaya Maritime Trg. Institute.

Modjo dry port and Gelan dry port is the biggest port that accommodate all the customers imported vehicles. If the customers use multimodal transport service (single administrative document), then his/her products will be shipped from the port of the origin to the point of destination through Djibouti port. The other method is unimodal transport service, it is type of single mode of transportation for instance sea, road, rail, or air freight of cargo. For this type of services to be done, there will be many operators and agreements involved. In unimodal system, the service ends at the port of Djibouti if the imported goods came by sea, and then the consignee/importer has the right to choose the transporter and make agreement to process his/her own goods. (ESLSE, 2016)

The study shows that the customers' satisfaction of using multi modal transport services is very low, this may arise many reasons. (Evaluation of Performances of Intermodal Import-Export Freight Transport System in Ethiopia, 2015.) The main reason is the delay of shipment from Djibouti port to Ethiopia dry port, the other is customs administrators does not have efficient way delivery to customers or importers. The most important part they missed it is delivery of information to customers is not satisfactory. This thesis is mainly focus on this part.

1.2 Objective of the project

The aim of this thesis is to develop android mobile Application for one part of Ethiopian shipping And Logistics Services Enterprise (ESLS) sectors is called Ethiopian Dry Port Enterprise which is the one responsible to take care of imported vehicles coming across the globe. However, the importers are informed the vehicles status through direct calling from the responsible authority that the car is whether it is already on dry port or the importer must check the websites of the ESLS if they post it. This Application help to vehicles importers to check whether their vehicle Arrived or not on the dry port.

The application will also reduce the work overload and will satisfy the customers' services.

1.3 Scope of the project

This project mainly focus on imported vehicles and those which are arrived in Ethiopia dry port, and those who are using the multi modal transport services. The customers must know their vehicles chassis as it is stated in the document of bill of Lading, otherwise They must know the bill of lading number to use this application.

This project also not only develop the android application development but also its design the database and connection of the Back-End.

2 Requirement Analysis

2.1 Functional Requirement

- MySQL database must be updated as the vehicle arrive in dry port of regional state of Ethiopia.
- Customers or users must first select the dry port which stated as his/her bill of Lading
- Customers or users must have vehicle identification number (VIN) also called chassis number if he or she want to search by VIN.
- Customers or users must have Bill of Lading number if he or she want to search based on it.
- The application works only for those who have android mobile device.

2.2 Software Requirement

The following are the software requirement to achieve the application

- Operating system: can be windows 7, 8,10. Mac and Linux can be others option. But in my case I used window 10.
- Language: Android is based on Java language
- Database: MySQL, and MySQL for Excel
- Technologies used: XML, JSON, Excel, MySQL, Java, ADT, SDK
- Tools: I used Android studio version 2.2, but It can be used version greater than version 1.0
- Server: XAMPP Server used as local server

3 Application Architecture and Design

This section will be illustrated the architecture and design part.

3.1 Application Architecture

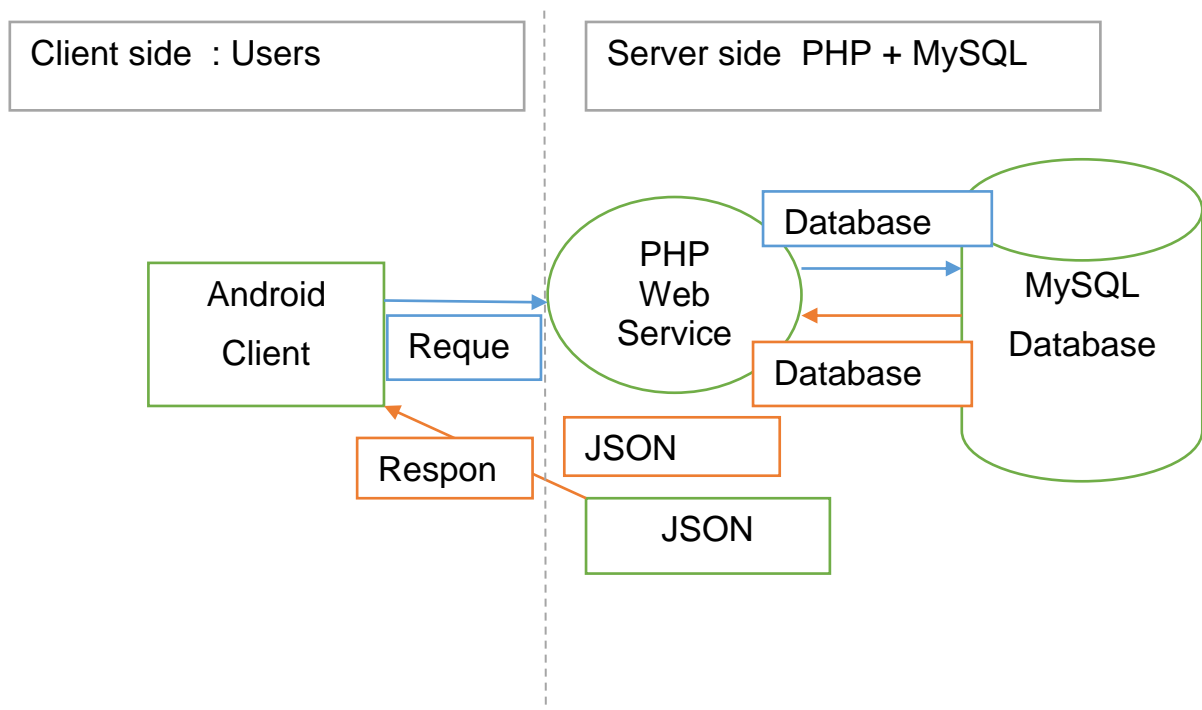


Figure 1. Architecture of the Application

When the Android Client in other words the person, who install the application in his android mobile device execute, it requests a PHP script from the web server and the PHP script will fetch data from the MySQL database by making database query. Based on the query request the database return data to PHP web services. finally, the PHP encode it into JSON format and send it to the android client. After getting the encode the android application parse the data and display it on mobile devices. As the fig 1 shown. Android client operates in the client side where as PHP and MySQL both are running in webserver which is server side. (Connect Android with PHP ,2016.)

3.1.1 Front-End side

On the front-End, the android application development using official Android studio version 2.0, XML for application design or User interface design. Coding brings the application functions to life. some of coding example are as shown below

- **Activities:** There can be many different activities or class within an application, occurring at different times, each of which gives your users a way to interact with your application. Activities for this application or class is discuss in detail in the application design chapter.
- **Manifest.** The Manifest defines the matching basic characteristics of your application, defining each component from the ground up. (Guide to Android Development ,2016.)

3.1.2 Back-End side

On the back-End side, server side PHP and MySQL is used for this application which is compulsory to work the application properly.

PHP or other web services that support the data flow from client to server and vice versa, Translating Data from database to JSON format and JSON to database is the prominent task in Mobile application.

Data is stored in MySQL format and PHP scripts can be used which to access MySQL databases. (PHP& MySQL, 2016.)

The PHP code for this application is shown in Appendix D. whereas the database design is discussed in the chapter 5.

3.2 Application Design

This Section will be discussed the Application Design Use Case Diagram and Class Diagram

3.2.1 Use Case Diagram

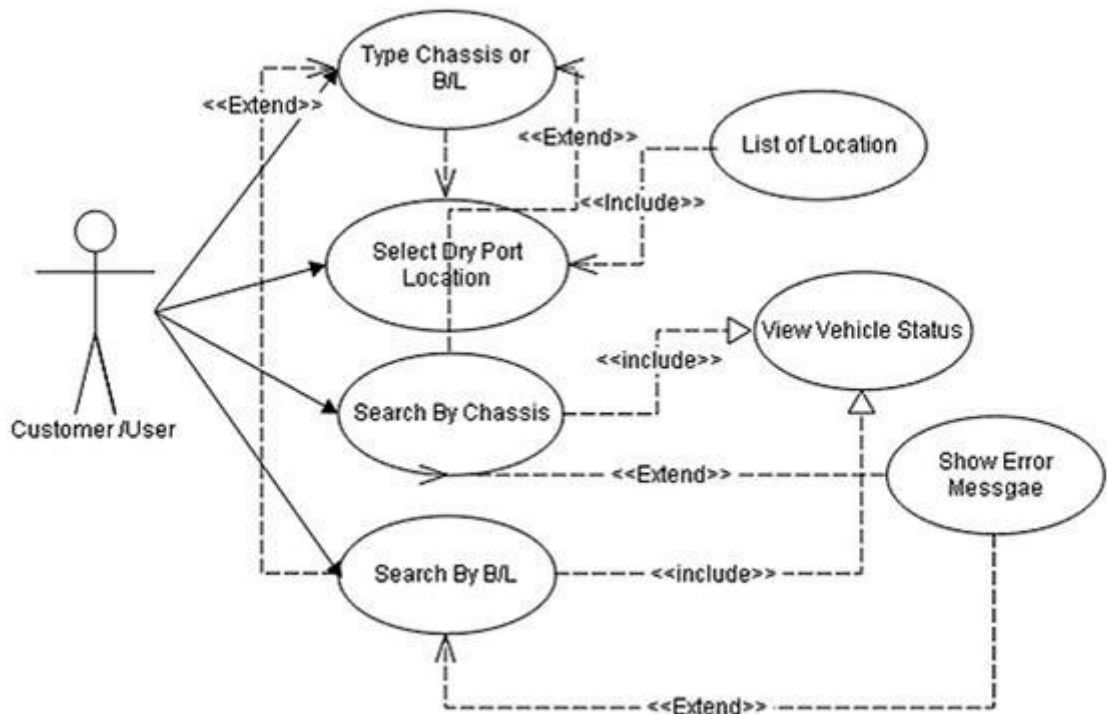


Figure 2. Use case Activity Diagram for Customers or Users

- The Customers need to first select the dry port location before typing chassis number or Bill of Lading Number (B/L).
- The system allows to type the chassis no or B/L.
- The customers click either the Search by Chassis or B/L, If the Customers Typed incorrect data then the system displayed Error Message. Otherwise It display the Vehicle status.

3.2.2 Class Diagram

The Class Diagram as shown below is defined using a rectangular with three section, the upper part describes the class name, the middle part describes the Fields, and the bottom section shows the Methods or the operation. The class name is always required. The diagram shows also how each class related with the other class.

The application is made with six important class, such as EvtsMainActivity, Service Handler, DryPortLocation, VehicleDetailinfo, List-ChassisActivity, VehicleArrivalInfoActivity.

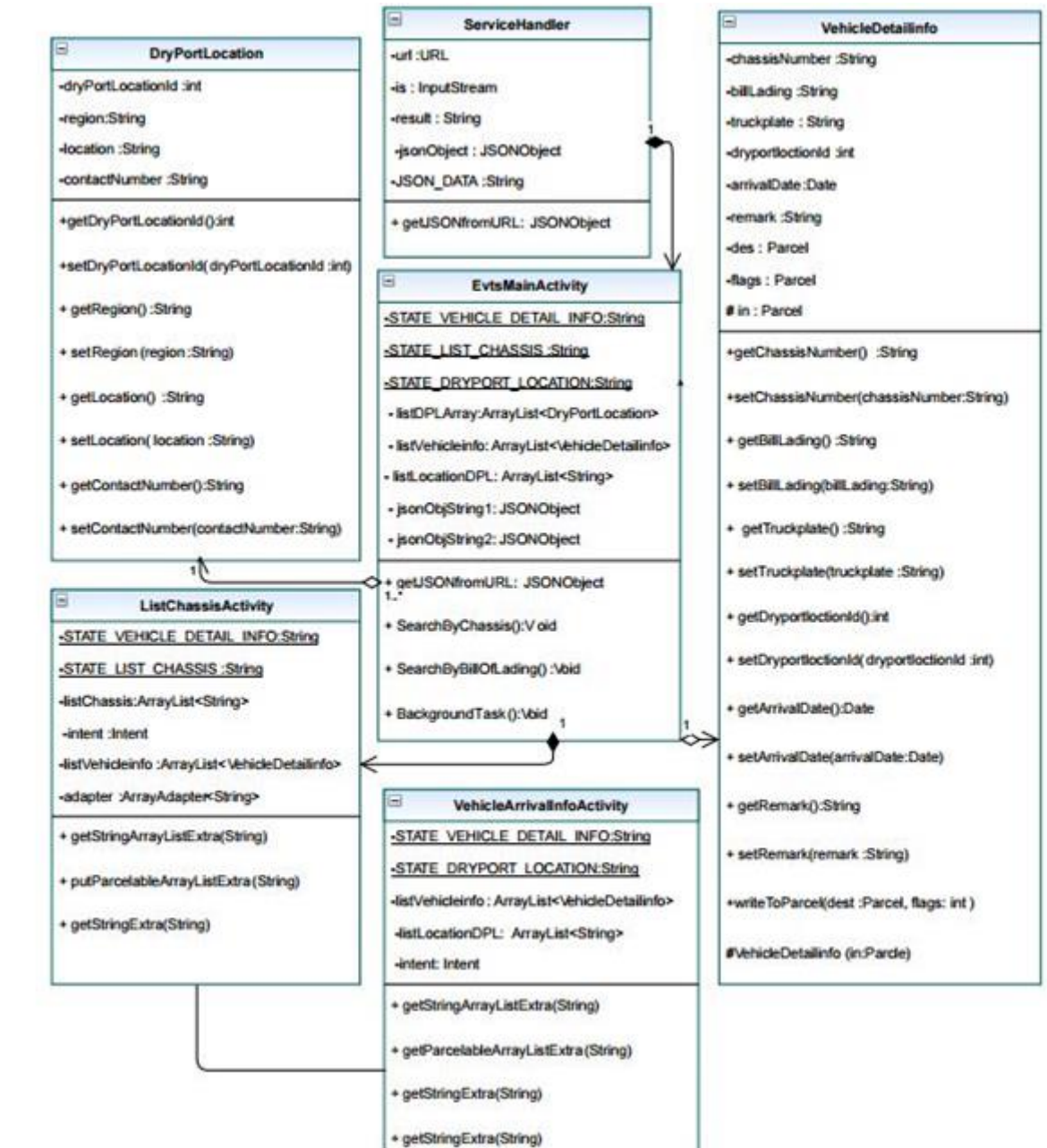


Figure 3. Class Diagram

EvtsMainActivity Class

This Class is the main class of the application, In the run time system start our program by calling this class first, and then calls the other methods required to run the program.

To know this class is the main, it is defined in the program manifest xml file.

```

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

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

```

In the EvtsMainActivity we are storing the JSON Nodes such as STATE_VEHICLE_DETAIL_INFO, STATE_CHASSIS_LIST, STATE_DRYPORT_LOCATION in the static variables.

This class used **Async task** method which has three methods within it.

- **onPostExecute ()**: It used for setting up the task for instate starting a progress bar.
- **doInBackground ()**: It Makes connections and receives JSON responses from the server using service handler class.
- **onPostExecute ()**: In this section, we can do user interface handling with the response data, or basically assign the response to specific variable types.

(Stack Overflow,2016.)

After the implementation of these, the other two Functions SearchByChassis () and SearchByBillOfLading (), gives the result.

ServiceHandlers Class

In ServiceHandler class, most commonly known as JSON Parser class. It gets the JSON data from the URL using **HttpPost** and it converts the response to JSON Object.

DryPortLocation and VehicleDetailinfo Class

Both class are model class, it helps us to model the data in the application. Once we make the fields are decided, we could generate getter and settler method. In Android studio (Alt +Insert) gives the option for auto generate.

ListChassisActivity Class

In this class, it displays the list of chassis that are searched by the users and populate to in the List View. When the users made a press any of the element, it passes to the new class called VehicleArrivalInfoActivity using Intent object.

VehicleArrivalInfoActivity

This class mainly show us the detail vehicle arrival information. It gets location String Array List by using getStringArrayListExtra () Method and get vehicle arrival information Array list using getParcelableArrayListExtra () method. Finally display the result in Text View.

4 Application Development Environment

This android application was made using Android studio 2.2 tools, its latest release official integrated development environment (IDE) by Google. It is open source available under the Apache license 2.0. The reason I used Android studio over other like Eclipse is because, Android studio already has ADT Plugin tools included in it but Eclipse does not. You can get all up-to-date version of the android studio from the official website. For detail information please refer the Android developer website at <https://developer.android.com/studio/features.html>. (Android developer 2016a.)

4.1 Installation of Android studio

For Android studio installation, we need to install Java development kit (JDK) version eight is required If not installed before. Once it is ready, we navigate to the official Android development home page at <https://developer.android.com/studio/index.html>. (Android developer 2016b.)

First, we download it .exe file from the link above and then we launch the executable file. After all steps finish it is ready to create new project file.

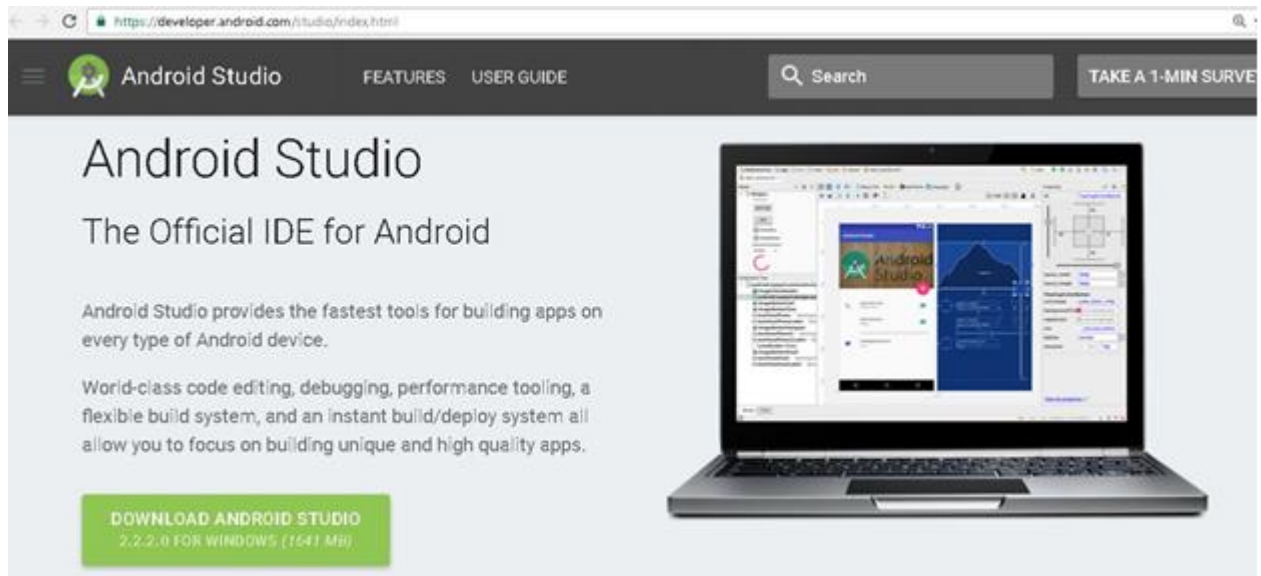


Figure 4. Android Studio Download Page

4.2 Creating an Android Project

If you launch the executable file the first time, welcome to Android studio window, click start a new Android project or once you opened project before, Select ->New Project.

In New project page, you will find Application Name and company name, you can edit as wish. then Click Next.

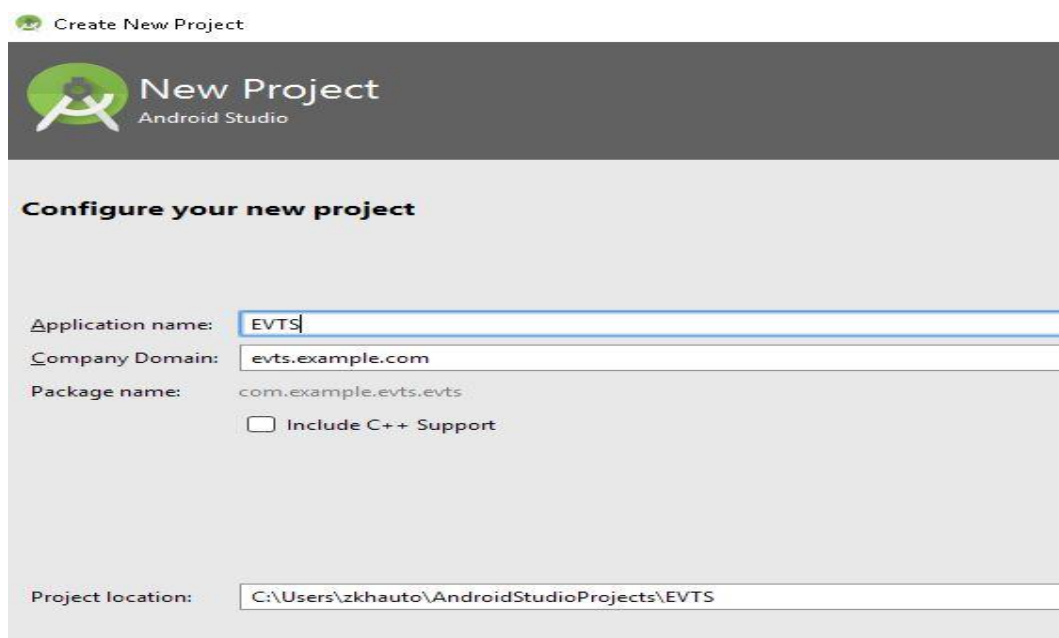


Figure 5. How to Create New Page

The Next step you will find “Target Android Devices” Screen and the minimum Required SDK. Usually Android studio put the recommendation, but it is possible to change the value best of the interest.

The next step is in the “Add an Activity to Mobile” screen, Select Empty Activity and press Next and keep the default “customize the Activity” screen and finish.

Since Android studio integrate ADT and SDK tools, unlike Eclipse no need to install separately. After Installation, you can see the version and update information by navigating Tools->Android->SDK manager.

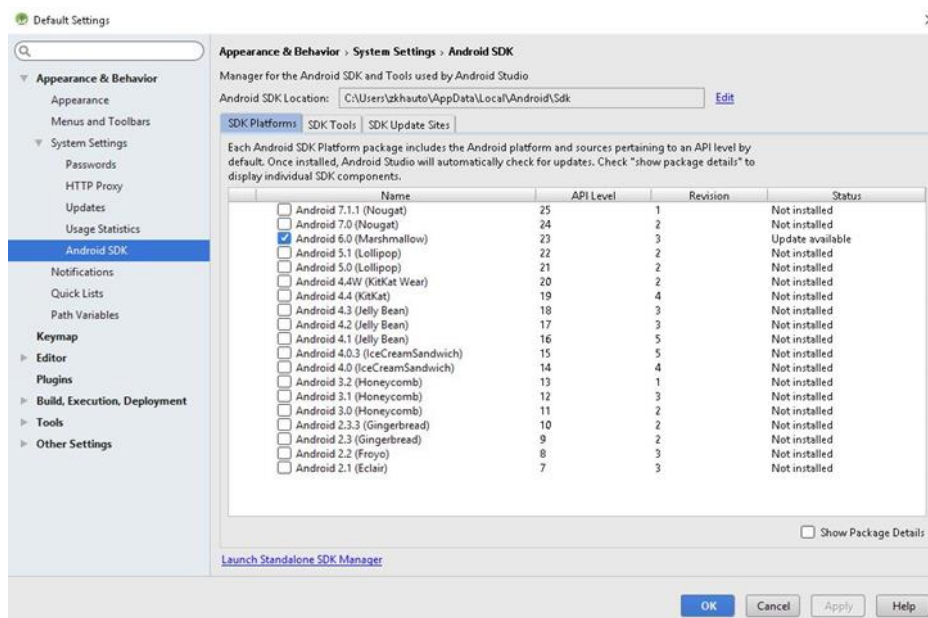


Figure 6. Android SDK Manager

5 Implementation

This section Explain How database is inserted to MySQL database management system and How data retrieve from application, and what it looks Graphical User interface. How the users access the data.

5.1 Database Management System

This Application developed based on the data from Ethiopian shipping Line, from my data collection I got information that They store data as Excel sheet, Once the vehicle is arrived in one of the dry port. They notify for the customers by calling direct phone or they do text message. They uploaded the data in their website to let the customer to see their cars. (ESL ,2016b.)

Table 1. Sample data Gelan Vehicle Arrival Report

Gelan Vehicle Arrival Report (on and before Jan 09,2016)				
NO	CHASSI NO	B/L Number	Trucks Plate No	Arrival Date
1	VNKKG96370A323795	ACA140-RTM-MDJ052	505D55	01-09-16
2	JT121JK2200008362	055148-044 MODJO	505D55	01-09-16
3	VNKKV183X0A266660	ACA140-RTM-MDJ051	505D55	01-09-16
4	JTDKV123503119622	ASI140MOD528	505D55	01-09-16
5	JTDKZ28EX00018483	ASI140MOD516	505D55	01-09-16
6	MMBCNK7401D015794	GUR012-RTM-MDJ039	75412/22712	01-09-16
7	VNKKV18310A431768	GUR012-RTM-MDJ012	75412/22712	01-09-16
8	JT121JK2100004659	NUS012MOD551	75412/22712	01-09-16
9	VNKKG96300A011561	NUS012MOD563	75412/22712	01-09-16
10	MMBCNK7404D006389	055148-005 GELAN	75412/22712	01-09-16
11	MMBCNKB407D013032	055148-001	75412/22712	01-09-16
12	JT121LK2200035983	055148-001	75412/22712	01-09-16
13	MMBCNK7403D005580	055148-001	75412/22712	01-09-16
14	JTDBW23E243032334	NSSCB15005610	61165/682	01-09-16
15	JTDBW23E143041879	NSSCB15005610	61165/682	01-09-16
16	AHTFR22G406010360	15106BM03	2962/01643	01-09-16
17	VNKKV18300A308155	ACA140-RTM-MDJ055	2962/01643	01-09-16
18	SCP100391807	NSSCB15005425	2962/01643	01-09-16
19	KSP902001777	NSSCB15005425	2962/01643	01-09-16
20	KSP905145379	NSSCB15005425	2962/01643	01-09-16

5.1.1 MySQL for Excel

As mentioned before there is no direct way of retrieving the excel data through application. So, what is the solution? The solution is first we must create a database and insert the excel data into it. This can be done basically by installing MySQL for excel and Export excel data to our server where our database is hosted. It allows you to edit the MySQL data directly from within Excel.

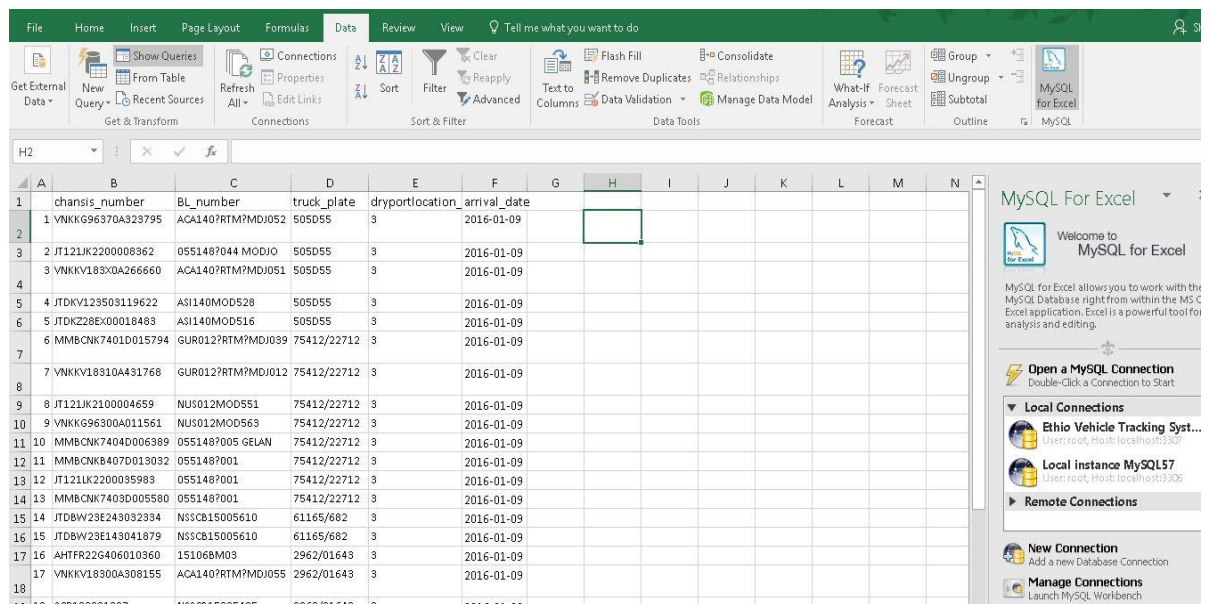


Figure 7. Excel Sheet with MYSQL for Excel

To connect to our remote server, press on New Connection, then it gives us a chance to fill the authentication. After the connection is successful, it enables us to edit, append, update and delete.

MySQL Instance Connection X

Connection Name: Type a name for the connection

Connection Method: Standard (TCP/IP) v Method to use to connect to the RDBMS

Connection Status: Unknown

Parameters

Hostname: Port: Name or IP address of the server host - TCP/IP port.

Username: Name of the user to connect with.

Password: The user's password, stored in a secured vault.

Default Schema: The default schema, leave blank to select it later.

Advanced

☐ Use ANSI quotes to quote identifiers. If enabled this option overwrites the server side settings.

☐ Use compression protocol. Select this option for WAN connections

Test Connection Cancel OK

Figure 8. MySQL Instance Connection

5.1.2 Database Design

The database used for this application is as early defined MySQL Database, it is included in XAMPP Apache server and hosted locally. The database name is Vehicle_tracking which has two table, namely dryport_location and vehicle_status. For instance, to access the database it is possible use the link as <http://localhost/phpmyadmin>. It is possible to connect a remote database with valid credentials by changing the hostname parameter. (XAMMP ,2016.)

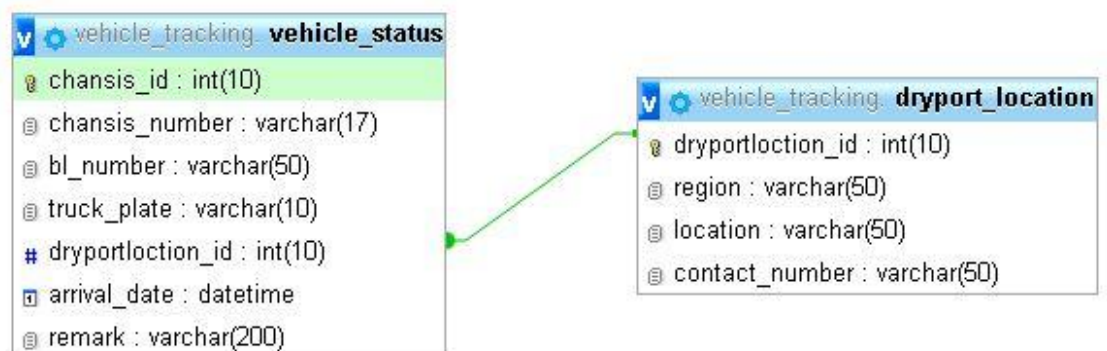


Figure 9. Entity relationship(ER) Diagram

5.2 Graphical user Interface

This Application is made with simple user interface; it's not complicated to understand how it works. Hints are given to make to more understandable and to show the user where needs the correct input.

The Application has three important screens, such as

- Home screen
- List of Chassis screen
- Vehicle Arrival Report screen

5.2.1 Home screen

A home screen or Start screen is the main screen of the application. It consists of Edit Text box where the user must type chassis or Bill of Lading (B/L).it has an option to select dry port location from spinner dropdown, once the customer or user put the two-necessary information on the above, there is two button option that give a result based on the input. These buttons are "Search by Chassis" and "Search by B/L". If the user press "Search by Chassis" the result shows in Text View on the same home screen below "Search By B/L", but if the user press "Search by B/L", it goes to next List of Chassis screen.

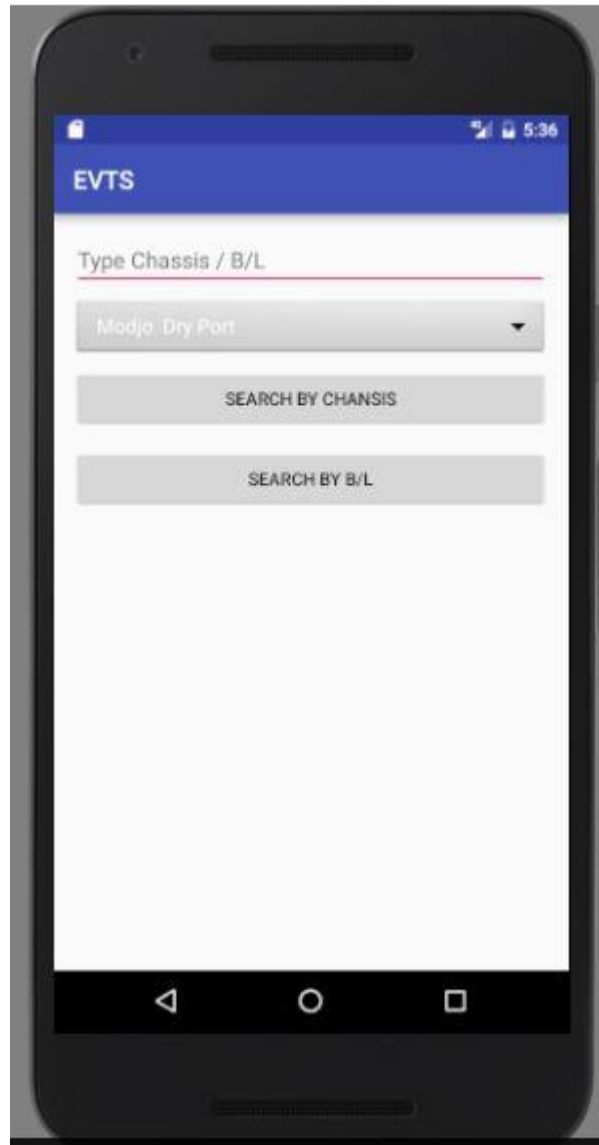


Figure 10. Home Screen of the application

5.2.2 List of Chassis screen

This screen shows us the list of chassis where the users type bill of lading number in edit Text box and press the “Search by B/L” button. The reason this screen is required because there might be more than one chassis number for a single bill of lading number, for instance from the data, bill of lading number “NSSCB15005592” has two chassis number such as “JTDBW23E140072325” and “JTDBW23E14007236”. once the user select

the chassis, it will display the “Vehicle Arrival Report” in new Screen called “Vehicle Arrival Report”.

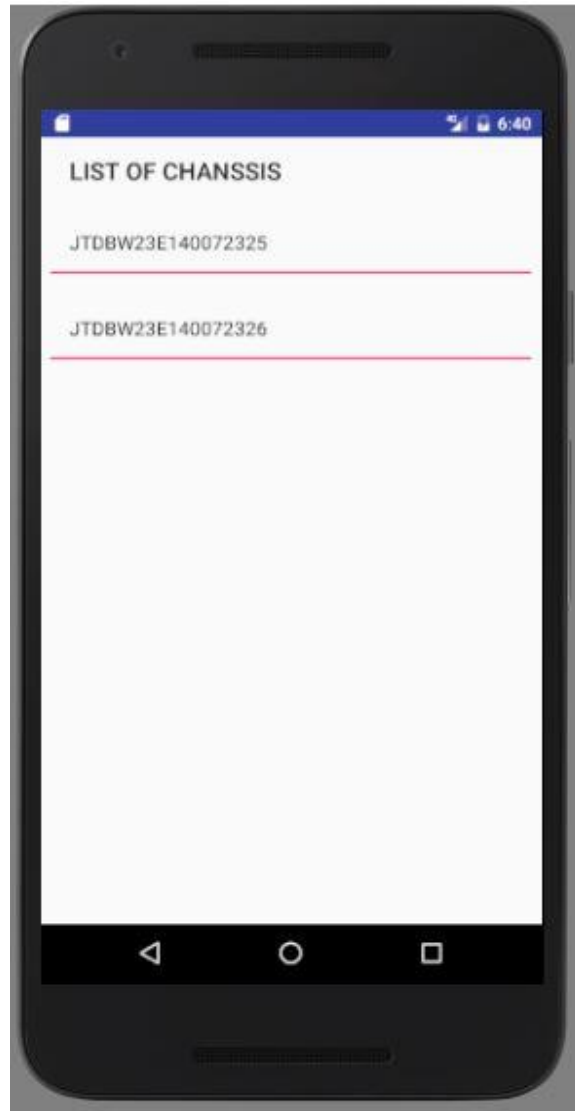


Figure 11. List of Chassis screen of the application

5.2.3 Vehicle Arrival Report screen

This screen appears as users press a chassis from List of chassis number screen, it will show the detail vehicle arrival information namely chassis number, bill of lading, arrival date, truck plate number, which is giving more information that the vehicle is brought to the dry port by the truck which its plate number is mention on it. And, display the contact number for whom

you can ask more information about it. The last one is the remark which explain if something happens on the vehicle.



Figure 12. Vehicle Arrival report screen of the Application

6 Testing

It is a common and necessary method which is conducted after the application is out of the development phase, it requires testing by different responsible organization. The purpose is to find errors or another defect related to functionality. For this application, I selected a simple unit test technique because as it needs a detailed information of the internal program, design, and code.

6.1 Unit Test

A unit is the smallest testable part of an application for instance certain functions, classes, procedures, interfaces. It helps us to determine failures in our code at early stage. It typically has a single input or a few inputs to verify if the functions are returning the correct values as we expecting.

In this application, I used six test cases as input value to verify the application whether it works as it has been expected or not. If there is any error on the functions, classes then it is easy to fix it right way.

Unit Testing is best method to assure that the existing functionality is not broken when additional functionality is Implemented.

Table 2. Unit Testing

No.	Module Name	Test case	Expected Result	Actual Result	Status (pass/Fail)
1	Home screen	display all the expect widget	Edit Text	edit Text	pass
			dropdown spinner	dropdown spinner	
			Search by Chassis Button	Search By chassis btn	
			Search by B/L Button	Search by B/L btn	
			Text View	Text View	
2	Home screen	Type on Edit text field correct chassis and press Search by chassis Button	display detail vehicle arrival report	display detail vehicle arrival report	pass
3	Home screen	Type on Edit text field chassis with lower and capital case letters and press Search by chassis Button	display detail vehicle arrival report	display detail vehicle arrival report	pass
4	Home screen	Type on edit text field chassis number with wrong input and press search by chassis btn	display error message	display error message	pass
5	Home screen	Type on Edit text field B/L and press search by B/L	List of Chassis Number	List of Chassis Number	pass
6	List of chassis screen	Select one chassis and press	display detail vehicle arrival report on new screen	display detail vehicle arrival report on new screen	pass

As shown from the above table the status result is pass, which indicate the application works properly as it has been expected.

7 Summary and Conclusion of the thesis

In this section, I summarize the main objective and the purpose of this application. I also discuss the result in conclusion part.

7.1 Summary

The main objective of this application is to make easier for vehicles importers in Ethiopia to track their car on Arrival status or not. Once the database is up to date then the users able to access the vehicle status whether it is on Arrival on dry port or not. I accomplished installing the development environment, designing the database, coding, and Testing the application. This thesis documented the process and the foremost steps taken. The objective is fulfilled.

7.2 Conclusion

This application has given me a significant confidence and competence on Android development Environment.

This application is limited to only Android users, it is essential to develop similar version for iOS (Mobile operating system for Apple formerly known as iPhone OS).

This application access the data from my local server, sample data is taken from Ethiopian Shipping lines websites. To publish online, the database connection must be moved from local server to their company server.

Finally, this application is the first version, it needs many improvement regarding to UI, and some additional things can be considered.

8 Future work

As I have mentioned before this application is the first version, so that in the update version It can be included the following possible extensions.

- User Interface improvement for instance Toolbar can be added in the future, it consists of Navigation button, A brand logo image, etc.
- It can also possible to add map which display the location of the dry port .so that the user can navigate from where he/she is.
- Possible to change the database design based on the additional set up.
- It can possible to consider the phone call from the application.

Bibliography

The list of references used.

Android developers 2016a.HardWare Requirement. URL:

<https://developer.android.com/studio/index.html>.

Accessed :25 Jul 2016.

Addis Fortune 2016a.Completion of Modjo Dry Port Expansion One Month Overdue .URL:

http://www.addisfortune.com/Vol_10_No_572_Archive/Completion%20of%20Modjo%20Dry%20Port%20Expansion%20One%20Month%20Overdue.htm

Accessed: 19 Jul 2016.

Connect Android with PHP.URL: <http://techlovejump.com/connect-android-with-php-mysql-and-json/> Accessed : 5 Nov 2016.

Ethiopian Shipping Line Company Profile (ESL),2016b.

URL:<http://www.ethiopianshippinglines.com.et/node/63>. Accessed: 15 Jul 2016.

ESL 2016a. Ethiopian Shipping Line Company Profile. URL:

<http://www.ethiopianshippinglines.com.et/companyprofile>. Accessed 20 Sept 2016.

Evaluation of Performances of Intermodal Import-Export Freight Transport System in Ethiopia ,2016. URL:

<http://www.scirp.org/journal/PaperDownload.aspx?paperID=53749> Accessed: 25 Sept 2016.

Guide to Android Development ,2016. A Guide to Android Development: What to Know URL: <https://www.upwork.com/hiring/mobile/android-development-what-to-know-who-to-hire> Accessed:1 Dec 2016

JSON 2016.What is JSON?URL: <http://json.com> .Accessed :15 Oct 2016.

PHP 2016.What is PHP? URL: <http://php.net/manual/en/intro-whatcando.php>.

Accessed: 15 Oct 2016.

PHP& MySQL 2016. How to Connect Android with PHP, MySQL, and JSON – Example. URL: <http://techlovejump.com/connect-android-with-php-mysql-and-json/> Accessed: 5 Nov 2016.

SeaRates LP 2016. Background of the company

URL:<https://www.searates.com/sealine/esl.htm> Accessed :20 Jul 2016.

Stack Overflow 2016. AsyncTask Android example.

URL:<http://stackoverflow.com/questions/9671546/asynctask-android-example>.

Accessed: 10 Nov 2016.

Xampp 2016.What is XAMPP? URL:

<https://www.apachefriends.org/index.html> Accessed:02 Oct 2016.

Appendix A- Application source code

Appendix A.1. – EvtsMainActivity

```
package com.example.evts.evts;

import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class EvtsMainActivity extends AppCompatActivity {

    private static final String STATE_VEHICLE_DETAIL_INFO =
"state_vehicle_detail_info";
    private static final String STATE_CHASSIS_LIST = "listChassis";
    private static final String STATE_DRYPORT_LOCATION =
"state_dryport_location";
    private static final String STATE_DRYPORT_DETAIL_INFO =
"state_dryport_detail_info";
    JSONObject jsonObjString1;
    JSONObject jsonObjString2;
    int spPosition;
    ListView listView;
    int dryportlocation_id;

    //ServiceHandler jParser = new ServiceHandler();
    ArrayList<String> listDpL = new ArrayList<>();
    ArrayList<DryPortLocation> listDPLArray = new ArrayList<>();
    ArrayList<String> listLocationDPL = new ArrayList<>();
    //=====
    ArrayList<VehicleDetailinfo> listVehicleinfo = new ArrayList<>();
    ArrayList<String> listChassis = new ArrayList<>();
    String[] values;

    String usrSearch = "";

    String location;
    String contact;

    // url to make request /URL to get JSON Array
    private static String url_dryport =
"http://192.168.0.102/ethiovehicle/dryportlocation.php";
    private static String url_vehicleStatus =
```

```

"http://192.168.0.102/ethiovehicle/vehiclestatus.php";

// JSON Node names
private static final String TAG_DRYPORTARRAY = "dryportarray";
private static final String TAG_DRYPORTLOCATION_ID = "dryportlocation_id";
private static final String TAG_REGION = "region";
private static final String TAG_LOCATION = "location";
private static final String TAG_CONTACT_NUMBER = "contact_number";
//=====
private static final String TAG_VehicleReport = "VehicleReport";
private static final String TAG_ChassisNumber = "chassis_number";
private static final String TAG_BillLading = "bl_number";
private static final String TAG_Truckplate = "truck_plate";
private static final String TAG_ArrivalDate = "arrival_date";
private static final String TAG_Remark = "remark";

// dryportarray JSONArray
JSONArray dryportarray = null;
JSONArray vehicleArray = null;

//=====
//public static final String PREFS_NAME="MyPrefsFile";
//    TextView textView = (TextView) findViewById(R.id.textView);

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_evts_main);
    /* Toolbar mytoolbar=(Toolbar) findViewById(R.id.custom_toolbar);
    setSupportActionBar(mytoolbar);
    getSupportActionBar().setTitle("ETHIO VEHICLE TRACKING SYSTEM");*/

    if (savedInstanceState != null) {

        listVehicleinfo = savedInstanceState.getParcelableArrayList(STATE_VEHICLE_DETAIL_INFO);
        listLocationDPL = savedInstanceState.getStringArrayList(STATE_DRYPORT_LOCATION);
        listDPLArray = savedInstanceState.getParcelableArrayList(STATE_DRYPORT_DETAIL_INFO);

    }

    SearchByChassis();
    SearchByBillOfLading();

    new BackgroundTask().execute();
}

@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putParcelableArrayList(STATE_VEHICLE_DETAIL_INFO, listVehicleinfo);
    outState.putStringArrayList(STATE_DRYPORT_LOCATION, listLocationDPL);
}

private void SearchByBillOfLading() {

    Button btnSearchByBL = (Button) findViewById(R.id.btnSearchByBL);

    btnSearchByBL.setOnClickListener(new View.OnClickListener() {
        @Override

```

```

        public void onClick(View v) {
            EditText inputUsr = (EditText) findViewById(R.id.editTxtSearchByChassisBL);
            final TextView textView = (TextView) findViewById(R.id.textView);

            ArrayList<String> listChassis1 = new ArrayList<>();

            if (inputUsr.toString() != null) {

                for (int i = 0; i < listVehicleinfo.size(); i++) {

                    dryportloction_id = listVehicleinfo.get(i).getDryportloctionId();

                    if (dryportloction_id == (spPosition + 1)) {

                        String myEditTxt = inputUsr.getText().toString();
                        String chassis = listVehicleinfo.get(i).getChassisNumber();
                        String billOfLading = listVehicleinfo.get(i).getBillLading();

                        
//*****
/* Date arriveDate = listVehicleinfo.get(i).getArrivalDate();
String remark = listVehicleinfo.get(i).getRemark();
String dryPortLocation = listLocationDPL.get(spPosition);
String phone = listDPLArray.get(spPosition).getContactNumber(); */
//*****

                        listView = (ListView) findViewById(R.id.listChassis);

                        ArrayList<Integer> myList = new ArrayList<>();
                        myList.add(i);

                        if (myEditTxt.equalsIgnoreCase(billOfLading)) {

                            listChassis1.add(chassis);

                            Intent intent = new Intent(EvtsMainActivity.this,
ListChassisActivity.class);
                            intent.putStringArrayListExtra(STATE_CHASSIS_LIST, listChassis1);
                            intent.putParcelableArrayListExtra(STATE_VEHICLE_DETAIL_INFO, listVehicleinfo);
                            intent.putExtra(STATE_DRYPORT_DETAIL_INFO, listDPLArray);
                            intent.putExtra(STATE_DRYPORT_LOCATION, listLocationDPL);

                            intent.putExtra("myEditTxt", myEditTxt);
                            startActivity(intent);

                        }

                    }

                }

            } else {

                textView.setText("INCORRECT ENTRY");
            }
        }
    }
}

```

```

    }

    });
}

private void SearchByChassis() {

    Button btnSearchByChassis = (Button) findViewById(R.id.btnSearchByChassis);

    btnSearchByChassis.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            EditText inputUsr = (EditText) findViewById(R.id.editTxtSearchByChassisBL);
            final TextView textView = (TextView) findViewById(R.id.textView);

            if (inputUsr.toString() != null) {
                for (int i = 0; i < listVehicleinfo.size(); i++) {

                    dryportloction_id = listVehi-
cleinfo.get(i).getDryportloctionId();

                    if (dryportloction_id == (spPosition + 1)) {

                        String myEditTxt = inputUsr.getText().toString();
                        String chassis = listVehi-
cleinfo.get(i).getChassisNumber();
                        String billOfLading = listVehi-
cleinfo.get(i).getBillLading();
                        Date arriveDate = listVehi-
cleinfo.get(i).getArrivalDate();
                        String remark = listVehicleinfo.get(i).getRemark();
                        String dryPortLocation = listLoca-
tionDPL.get(spPosition);
                        String phone = listDPLAr-
ray.get(spPosition).getContactNumber();

                        if (myEditTxt.equalsIgnoreCase(chassis) || myEdit-
Txt.equalsIgnoreCase(billOfLading)) {

                            textView.setText("chassis :" + chassis + "\n" +
"BillOfLading :" + billOfLading + "\n"
                                + "Arrival Date :" + arriveDate + "\n" +
"Dry port Location :" + dryPortLocation + "\n"
                                + "Contact Number : " + phone + "\n" +
"Remark :" + remark + "\n");
                        }

                    }

                } else {
                    textView.setText("Incorrect Entry");
                }

            }

        });
}

```

```

    }

    //GET JSON FILE
    private class BackgroundTask extends AsyncTask<Void, Void, Void> {

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

        @Override
        protected Void doInBackground(Void... param) {

            ServiceHandler jParser1 = new ServiceHandler();
            ServiceHandler jParser2 = new ServiceHandler();
            //Getting Dry port JSON String FROM URL
            jsonObjString1 = jParser1.getJSONfromURL(url_dryport);
            // Get Vehicle Detail Info Json String From URL
            jsonObjString2 = jParser2.getJSONfromURL(url_vehicleStatus);

            try {
                vehicleArray = jsonObjString2.getJSONArray(TAG_VehicleReport);

                for (int i = 0; i < vehicleArray.length(); i++) {

                    VehicleDetailinfo vdi = new VehicleDetailinfo();

                    JSONObject jObjVehicle = vehicleArray.getJSONObject(i);

                    vdi.setDryportlocationId(jObjVehicle.getInt(TAG_DRYPORTLOCATION_ID));

                    vdi.setChassisNumber(jObjVehicle.getString(TAG_ChassisNumber));
                    vdi.setBillLading(jObjVehicle.getString(TAG_BillLading));
                    vdi.setTruckplate(jObjVehicle.getString(TAG_Truckplate));

                    String dateStr = jObjVehicle.getString(TAG_ArrivalDate);
                    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");

                    Date arrivalDate = sdf.parse(dateStr);
                    vdi.setArrivalDate(arrivalDate);
                    vdi.setRemark(jObjVehicle.optString(TAG_Remark));
                    listVehicleinfo.add(vdi);
                    listChassis.add(jObjVehicle.getString(TAG_ChassisNumber));

                    //then

                }
            } catch (JSONException e) {
                Log.e("Log_TAG", "VehicleArray String Error !" + e.toString());
            } catch (ParseException e) {
                Log.e("Log_TAG", "Date Format Error !" + e.toString());
            }

            try {
                dryportarray = jsonObjString1.getJSONArray(TAG_DRYPORTARRAY);

                //looping through all dryPortArray

                for (int i = 0; i < dryportarray.length(); i++) {

                    JSONObject jOb = dryportarray.getJSONObject(i);
                    DryPortLocation dpl = new DryPortLocation();

```

```

        dpl.setDryPortLocationId(jOb.optInt(TAG_DRYPORTLOCATION_ID));
        dpl.setRegion(jOb.optString(TAG_REGION));
        dpl.setLocation(jOb.optString(TAG_LOCATION));
        dpl.setContactNumber(jOb.optString(TAG_CONTACT_NUMBER));
        listDPLArray.add(dpl);
        listLocationDPL.add(jOb.optString(TAG_LOCATION));

    }

    } catch (JSONException e) {
        Log.e("Log_Tag", "Getting Array of Dryport not OK!" +
e.toString());
    }

    return null;
}

@Override
protected void onPostExecute(Void s) {

    final Spinner sp = (Spinner) findViewById(R.id.spinnerDropDown);

    ArrayAdapter<String> dataAdapter1 = new ArrayAdapter<
String>(getApplicationContext(),
        android.R.layout.simple_spinner_item, listLocationDPL);
    dataAdapt-
er1.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    sp.setAdapter(dataAdapter1);

    //=====added=====
    sp.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {

        // On selecting a spinner item
        String item = listLocationDPL.get(position);

        spPosition = sp.getSelectedItemPosition();

        //PositionDisplay.setText("Item Position is = " + Hold );
        EditText editText = (EditText) findViewBy-
Id(R.id.editTxtSearchByChansisBL);

        // Showing selected spinner item

        //Toast.makeText(this,"Selected Location : " + item ,).show();
        TextView textView = (TextView) findViewById(R.id.textView);
        textView.setText("Item position is =" + spPosition);
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

});
//=====End Here=====00
}

```

```
}  
}
```

Appendix A.2. – DryPortLocation

```
package com.example.evts.evts;  
  
import android.os.Parcel;  
import android.os.Parcelable;  
  
/**  
 * Created by Zekir on 10/3/2016.  
 */  
  
public class DryPortLocation implements Parcelable{  
    private int dryPortLocationId;  
    private String region;  
    private String location;  
    private String contactNumber;  
  
    protected DryPortLocation(Parcel in) {  
        dryPortLocationId = in.readInt();  
        region = in.readString();  
        location = in.readString();  
        contactNumber = in.readString();  
    }  
  
    public static final Creator<DryPortLocation> CREATOR = new Crea-  
tor<DryPortLocation>() {  
        @Override  
        public DryPortLocation createFromParcel(Parcel in) {  
            return new DryPortLocation(in);  
        }  
  
        @Override  
        public DryPortLocation[] newArray(int size) {  
            return new DryPortLocation[size];  
        }  
    };  
  
    public int getDryPortLocationId() {  
        return dryPortLocationId;  
    }  
  
    public void setDryPortLocationId(int dryPortLocationId) {  
        this.dryPortLocationId = dryPortLocationId;  
    }  
  
    public String getRegion() {  
        return region;  
    }  
  
    public void setRegion(String region) {  
        this.region = region;  
    }  
  
    public String getLocation() {  
        return location;  
    }  
  
    public void setLocation(String location) {  
        this.location = location;  
    }  
  
    public String getContactNumber() {  
        return contactNumber;  
    }  
}
```

```

    public void setContactNumber(String contactNumber) {
        this.contactNumber = contactNumber;
    }

    public DryPortLocation() {
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {

        dest.writeInt(dryPortLocationId);
        dest.writeString(region);
        dest.writeString(location);
        dest.writeString(contactNumber);
    }
}

```

Appendix A.3. - ServiceHandler

```

package com.example.evts.evts;

import android.util.Log;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

/**
 * Created by zekir on 10/29/2016.
 */

//JSON Parser class

public class ServiceHandler {

    public ServiceHandler() {
    }

    public static JSONObject getJSONfromURL(String urls) {
        URL url;
        InputStream is = null;
        String result = "";
        JSONObject jsonObject = null;
        String JSON_DATA;

        try {

            //Get JSON Data from URL
            url = new URL(urls);
            HttpURLConnection httpURLConnection = (HttpURLConnection)
            url.openConnection();

```



```

        InputStream inputStream = httpURLConnection.getInputStream();
        //Convert response to String
        BufferedReader bufferedReader = new BufferedReader(new In-
putStreamReader(inputStream));
        StringBuilder stringBuilder = new StringBuilder();

        while ((JSON_DATA = bufferedReader.readLine()) != null) {

            stringBuilder.append(JSON_DATA + "\n");
        }
        bufferedReader.close();
        inputStream.close();
        httpURLConnection.disconnect();
        result = stringBuilder.toString();

        jsonObject = new JSONObject(result);

    } catch (MalformedURLException e) {

        Log.e("log_tag", "Error in Http connection" + e.toString());

    } catch (IOException e) {
        Log.e("Log_tag", "Error Converting the result" + e.toString());
    } catch (JSONException e) {
        Log.e("Log_tag", "Error parsing data" + e.toString());
    }
    return jsonObject;
}
}

```

Appendix A.4 – ListChassisActivity

```

package com.example.evts.evts;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import java.util.ArrayList;

/**
 * Created by zekir on 11/10/2016.
 */

public class ListChassisActivity extends Activity {

    private static final String STATE_VEHICLE_DETAIL_INFO =
"state_vehicle_detail_info";
    private static final String STATE_DRYPORT_DETAIL_INFO =
"state_dryport_datail_info";
    private static final String STATE_LIST_CHASSIS = "listChassis";
    ListView listView;
    ArrayList<String> listChassis = new ArrayList<>();
    ArrayList<Integer> getMyList = new ArrayList<>();
    String getMyEditText;

    ArrayList<DryPortLocation> listDPLArray = new ArrayList<>();
    ArrayList<VehicleDetailinfo> listVehicleinfo = new Ar-

```

```

rayList<VehicleDetailinfo>());

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.list_chassis);

        listView = (ListView) findViewById(R.id.listChassis);
        //EditText inputUsr = (EditText) findViewById(R.id.editTxtSearchByChansisBL);

        Intent intent = getIntent();
        listVehicleinfo = intent.getParcelableArrayListExtra(STATE_VEHICLE_DETAIL_INFO);
        listDPLArray = intent.getParcelableArrayListExtra(STATE_DRYPORT_DETAIL_INFO);

        // final ArrayList<VehicleDetailinfo> listVehicleinfo
        =getIntent().getParcelableArrayListExtra("VehicleDetailinfo");

        listChassis = intent.getStringArrayListExtra(STATE_LIST_CHASSIS);

        //=====TESTING-----

        final ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, listChassis);
        listView.setAdapter(adapter);

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Intent intent1 = new Intent(ListChassisActivity.this, VehicleArrivalInfoActivity.class);
                String selectedValue = adapter.getItem(position);
                intent1.putExtra(selectedValue, selectedValue);

                intent1.putParcelableArrayListExtra(STATE_VEHICLE_DETAIL_INFO, listVehicleinfo);
                intent1.putParcelableArrayListExtra(STATE_DRYPORT_DETAIL_INFO, listDPLArray);
                startActivity(intent1);

            }

        });

    }
}

```

Appendix A.5 – VehicleDetailinfo

```

package com.example.evts.evts;

import android.os.Parcel;
import android.os.Parcelable;

import java.util.Date;

/**
 * Created by zekir on 11/8/2016.
 */

```

```

public class VehicleDetailinfo implements Parcelable {

    private String chassisNumber;
    private String billLading;
    private String truckplate;
    private int dryportloctionId;
    private Date arrivalDate;
    private String remark;

    protected VehicleDetailinfo(Parcel in) {
        chassisNumber = in.readString();
        billLading = in.readString();
        truckplate = in.readString();
        dryportloctionId = in.readInt();
        arrivalDate = new Date(in.readLong());
        remark = in.readString();
    }

    public static final Creator<VehicleDetailinfo> CREATOR = new Creator<VehicleDetailinfo>() {
        @Override
        public VehicleDetailinfo createFromParcel(Parcel in) {
            return new VehicleDetailinfo(in);
        }

        @Override
        public VehicleDetailinfo[] newArray(int size) {
            return new VehicleDetailinfo[size];
        }
    };

    public String getChassisNumber() {
        return chassisNumber;
    }

    public void setChassisNumber(String chassisNumber) {
        this.chassisNumber = chassisNumber;
    }

    public String getBillLading() {
        return billLading;
    }

    public void setBillLading(String billLading) {
        this.billLading = billLading;
    }

    public String getTruckplate() {
        return truckplate;
    }

    public void setTruckplate(String truckplate) {
        this.truckplate = truckplate;
    }

    public int getDryportloctionId() {
        return dryportloctionId;
    }

    public void setDryportloctionId(int dryportloctionId) {
        this.dryportloctionId = dryportloctionId;
    }

    public Date getArrivalDate() {
        return arrivalDate;
    }
}

```

```

    public void setArrivalDate(Date arrivalDate) {
        this.arrivalDate = arrivalDate;
    }

    public String getRemark() {
        return remark;
    }

    public void setRemark(String remark) {
        this.remark = remark;
    }

    public VehicleDetailinfo() {
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(chassisNumber);
        dest.writeString(billLading);
        dest.writeString(truckplate);
        dest.writeInt(dryportloctionId);
        dest.writeLong(arrivalDate.getTime());
        dest.writeString(remark);
    }
}

```

Appendix A.6. – VehicleArrivalInfoActivity

```

package com.example.evts.evts;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.Date;

/**
 * Created by zekir on 11/12/2016.
 */

public class VehicleArrivalInfoActivity extends Activity {

    private static final String STATE_VEHICLE_DETAIL_INFO =
"state_vehicle_detail_info";
    private static final String STATE_DRYPORT_DETAIL_INFO =
"state_dryport_datail_info";

    ArrayList<VehicleDetailinfo> listVehicleinfo = new Ar-
rayList<VehicleDetailinfo>();
    ArrayList<DryPortLocation> listDPLArray = new ArrayList<>();

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

        Intent intent = getIntent();
    }
}

```

```

        listDPLArray = intent.getParcelableArrayListExtra(STATE_DRYPORT_DETAIL_INFO);

        listVehicleinfo = getIntent().getParcelableArrayListExtra(STATE_VEHICLE_DETAIL_INFO);

        TextView txtView = (TextView) findViewById(R.id.txtViewArrivalInfo);

        String selectedChassis = intent.getStringExtra("selectedValue");

        for (int i = 0; i < listVehicleinfo.size(); i++) {

            int t = listVehicleinfo.get(i).getDryportloctionId() - 1;
            String chassis = listVehicleinfo.get(i).getChassisNumber();
            String billOfLading = listVehicleinfo.get(i).getBillLading();
            Date arriveDate = listVehicleinfo.get(i).getArrivalDate();
            String remark = listVehicleinfo.get(i).getRemark();
            String dryPortLocation = listDPLArray.get(t).getLocation();

            String contactNumber = listDPLArray.get(t).getContactNumber();
            String truckPlate = listVehicleinfo.get(i).getTruckplate();

            if (selectedChassis.equalsIgnoreCase(chassis)) {

                txtView.setText("chassis :" + chassis + "\n" + "BillOfLading :" +
                    billOfLading + "\n"
                        + "Arrival Date :" + arriveDate + "\n" + "Truck Plate : "
                    + truckPlate + "\n" + "Dry port Location :" + dryPortLocation + "\n"
                        + "Contact Number : " + contactNumber + "\n" + "Remark :"
                    + remark + "\n");
            }

        }
    }
}

```

Appendix B.1. – Application Android Manifest xml file

```

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

    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".EvtsMainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ListChassisActivity"
            android:label="@string/List_chassis"/>
        <activity android:name=".VehicleArrivalInfoActivity" />
    </application>

```

</manifest>

Appendix C.1. - Layout xml file

activity_evts_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_evts_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.evts.evts.EvtsMainActivity">

    <EditText
        android:layout_width="0dp"
        android:layout_height="43dp"
        android:inputType="text|textAutoCorrect|textPersonName"
        android:ems="10"
        android:id="@+id/editTxtSearchByChansisBL"
        android:layout_marginStart="16dp"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginLeft="16dp"
        android:layout_marginEnd="16dp"
        app:layout_constraintRight_toRightOf="parent"
        android:layout_marginRight="16dp"
        android:layout_marginTop="16dp"
        app:layout_constraintTop_toTopOf="parent"
        android:textColorLink="?attr/colorControlActivated"
        android:textAllCaps="false"
        android:hint="Type Chassis / B/L" />

    <Button
        android:text="Search By Chansis"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:id="@+id/btnSearchByChansis"
        android:layout_marginEnd="16dp"
        app:layout_constraintRight_toRightOf="parent"
        android:layout_marginRight="16dp"
        android:layout_marginStart="16dp"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="8dp"
        app:layout_constraintTop_toBottomOf="@+id/spinnerDropDown" />

    <Spinner
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:id="@+id/spinnerDropDown"
        android:prompt="@string/spinner_title"
        android:layout_marginStart="16dp"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginLeft="16dp"
        android:layout_marginEnd="16dp"
        app:layout_constraintRight_toRightOf="parent"
        android:layout_marginRight="16dp"
        app:layout_constraintTop_toBottomOf="@+id/editTxtSearchByChansisBL"
        android:elevation="0dp"
        android:background="@android:drawable/btn_dropdown"
        android:theme="@style/Widget.AppCompat.ListView"
```

```

        android:dropDownWidth="match_parent"
        android:keepScreenOn="false" />

<Button
    android:text="Search By B/L"
    android:layout_width="0dp"
    android:layout_height="48dp"
    android:id="@+id/btnSearchByBL"
    android:layout_marginTop="16dp"
    app:layout_constraintTop_toBottomOf="@+id/btnSearchByChansis"
    android:layout_marginStart="16dp"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginLeft="16dp"
    android:layout_marginEnd="16dp"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginRight="16dp"
    app:layout_constraintHorizontal_bias="0.55" />

<TextView
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:id="@+id/textView"
    android:layout_marginStart="16dp"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginLeft="16dp"
    android:layout_marginEnd="16dp"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginRight="16dp"
    android:layout_marginTop="16dp"
    app:layout_constraintTop_toBottomOf="@+id/btnSearchByBL"
    app:layout_constraintHorizontal_bias="0.33"
    app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginBottom="16dp" />
</android.support.constraint.ConstraintLayout>

```

Appendix C.2. – list_chassis xml file

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.Toolbar
        android:layout_width="0dp"
        android:layout_height="56dp"
        android:theme="@android:style/Theme.DeviceDefault.Light"
        android:minHeight="?attr/actionBarSize"
        android:id="@+id/toolbar"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        app:title="@string/List_chassis"
        app:titleMargins="2dp" />

    <ListView
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:id="@+id/listChassis"

```

```

        style="@style/Widget.AppCompat.ListView.DropDown"
        android:headerDividersEnabled="true"
        android:footerDividersEnabled="true"
        android:layout_marginStart="8dp"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginEnd="8dp"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        android:layout_marginBottom="8dp"
        android:choiceMode="singleChoice"
        android:layout_marginTop="8dp"
        app:layout_constraintTop_toBottomOf="@+id/toolbar"
        android:scrollIndicators="right|end"
        app:layout_constraintVertical_bias="0.62"
        android:divider="?android:attr/actionModeSplitBackground" />
</android.support.constraint.ConstraintLayout>

```

Appendix D. PHP Code

```
<? php
```

```
    //open connection to mysql db
```

```
    $connection =
```

```
    mysqli_connect("127.0.0.1:3307","root","Password","vehicle_tracking") or
    die("Error " . mysqli_error($connection));
```

```
    //fetch table rows from mysql db
```

```
    $sql = "select * from vehicle_status";
```

```
    $result = mysqli_query($connection, $sql) or die("Error in Selecting " .
    mysqli_error($connection));
```

```
    //create an array
```

```
    $emparray = array();
```

```
    while($row =mysqli_fetch_assoc($result))
```

```
    {
```

```
        $emparray['VehicleReport'][] = $row;
```

```
    }
```

```
    echo json_encode($emparray);
```

```
    //close the db connection
```

```
    mysqli_close($connection);
```

```
?>
```