A REPORT

ON

AMAZON INDIA SHOPPING ANDROID APP

By

Theegela N V B Sri Teja **2012A7PS019G**

Prepared in partial fulfillment of the
Practice School-II Course

AT

AMAZON DEVELOPMENT CENTER, BANGALORE
A Practice School II Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

MARCH, 2016

A REPORT

ON

AMAZON INDIA SHOPPING ANDROID APP

By

Theegela N V B Sri Teja **2012A7PS019G**

Prepared in partial fulfillment of the
Practice School-II Course

AT

AMAZON DEVELOPMENT CENTER, BANGALORE
A Practice School II Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

MARCH, 2016

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)

Practice School Division

Station: Amazon Development Center **Centre:** Bangalore

Duration: 6 Months **Date of Start:** 11th January, 2016

Date of Submission: 15th March, 2016

Title of the Project: Amazon India Shopping Android App

Name: Theegela N V B Sri Teja **ID No:** 2012A7PS019G
Discipline: B.E. (Hons) Computer Science

Name(s) and: Mr. Aman Gupta (Mentor)
Designation(s) Mr. Amit Arora (Manager)
of the expert(s)

Name(s) of the: Mr. Vishwanathan Hariharan
PS Faculty

Key Words: Mobile, Android, Shopping

Project Areas: Android Application Development

Abstract: With the penetration of smartphones increasing in India, and with the internet on smartphones becoming much more affordable and accessible, the e-commerce market on mobile is exponentially increasing. The goal of this project is to make the Amazon India App better by adding various features to improve the overall customer experience of the app.

Signature of Student Signature of PS Faculty

Date: 15th March, 2016

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)

PRACTICE SCHOOL DIVISION

Response Option Sheet

Station: Amazon Development Center **Center**: Bangalore

Name: Theegela N V B Sri Teja **ID No**: 2012A7PS019G

Title of the Project: Amazon India Shopping Android App

Usefulness of the project to the on-campus courses of study in various disciplines. Project should be scrutinized keeping in view the following response options. Write Course No. and Course Name against the option under which the project comes.

Code No.	Response Option	Course No.(s) & Name
1.	A new course can be designed out of this project.	NO
2.	The project can help modification of the course content of some of the existing Courses	NO
3.	The project can be used directly in some of the existing Compulsory Discipline Courses (CDC)/ Discipline Courses Other than Compulsory (DCOC)/ Emerging Area (EA), etc. Courses	NO
4.	The project can be used in preparatory courses like Analysis and Application Oriented Courses (AAOC)/ Engineering Science (ES)/ Technical Art (TA) and Core Courses.	NO
5.	This project cannot come under any of the above mentioned options as it relates to the professional work of the host organization.	YES

Signature of Student
Date: 15th March, 2016

Signature of Faculty

ACKNOWLEDGEMENT

The successful completion of various parts of this project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my heartfelt gratitude to the people who have been instrumental in the successful completion of this project.

I want to start by thanking my mentor, **Mr. Aman Gupta**, whose constant supervision was invaluable. His knowledge in this domain and his commitment towards work were always a source of motivation for me. I would also like to thank my manager, **Mr. Amit Arora**, and my senior manager **Mr. Sanket Murarka** for their encouragement throughout the project. I am deeply indebted to the employees of the Mobile development team, particularly **Ms. Poornima, Mr. Ashish Chavan,** my fellow intern **Ms. Charu Roy** for their unending patience and support all along.

I am thankful to **Amazon** for providing me with an opportunity to work on latest tools and technology and alongside such talented people.

I am also thankful to my PS instructor, **Mr. Vishwanathan Hariharan**, who has always been supportive and always guided us in the right direction. He made our PS experience hassle free, and ensured that no one had any problems.

Lastly, I would like to thank my family and all my friends for their suggestions and constant motivation which were imperative to the successful completion of the project.

TABLE OF CONTENTS

INTRODUCTION

Amazon App, also known as the mShop app internally, allows you to conveniently shop on-the-go; browse & search for your desired products and brands, view recommendations & customer reviews, add to wish list & cart, complete the purchase using the available payment options and track the status of your orders. New customers can sign up for an account using the Amazon App, while existing customers can login using their Amazon.in credentials to access the same account that they use on the website.

The users can shop anywhere with ease and confidence from millions of products across a broad range of categories including Books; eBooks & Tablets; Movies & TV shows; Kindle Devices, Computers & Accessories; Mobiles & Accessories; Consumer Electronics; Toys & Games; Baby Products; Health and Personal Care products; Watches; Fashion Jewelry; Home & Kitchen products; Beauty Products; Video Games; Music; Luggage & Bags; Handbags & Clutches; Sports, Fitness & Outdoor equipment; Shoes; Men’s, Women’s & Kid’s wear; Pet Supplies and more.

My task through the internship is to add features to the mShop app, to make the customer experience better. Some of the features added so far include Play Store Rating System, Search Suggestions and re-implementing the menu in the mShop app.

SCOPE AND OBJECTIVE OF THE PROJECT

With the penetration of smartphones increasing in India, and with the internet on smartphones becoming much more affordable and accessible, the e-commerce market on mobile is exponentially increasing. The goal of this project is to make the Amazon India App better by adding various features to improve the overall customer experience of the app.

The scope of this project is to add various features to the Amazon Shopping app or the mShop app to improve the overall customer experience on the mobile app. The mobile team does user research and comes up with requirements of features to be added. My task is to implement new features according to the requirements given. Several features like Play Store Rating System, search suggestions and hamburger menu have been implemented so far.

TECHNICAL SPECIFICATIONS

What is Android?

Android is an open source mobile operating system that combines and builds upon parts of many different open source projects. What does this mean to you as a developer? You have access to the source code of the platform that is running on the phone. This can help you better understand how interface controls and the various other pieces work. If you happen to find a bug, you can also submit a patch for the issue, though this is a more advanced practice. Google has also pulled together a large group of companies (called the Open Handset Alliance) that both contribute to and use the Android OS in their hardware devices. This means that there is industry-wide support for Google's OS, promising wide adoption across well-known vendors.

Why Android?

There are many advantages to developing for the Android platform:

- **Zero startup costs to begin development.** The development tools for the platform are free to download, and Google only charges a small fee to distribute applications on the Android Market.
- **Freedom to innovate.** The Android OS is an open-source platform based on the Linux kernel and multiple open-source libraries. In addition to building applications to run on Android devices, developers are free to contribute to or extend the platform as well.
- **Freedom to collaborate.** Android developers are not required to sign an NDA and are encouraged to collaborate and share source code with each other. According to a survey by Black Duck Software, the number of open source mobile apps and libraries grew at a rate of 168% from 2008 to 2009, faster on Android than any other platform. This means more code that you can reuse in your own projects to bring them to market much faster.

- **Open distribution model.** Very few restrictions are placed on the content or functionality allowed in Google's Android Market, and developers are free to distribute their applications through other distribution channels as well.
- **Multi-platform support.** There are a wide variety of hardware devices powered by the Android OS, including many different phones and tablet computers. Development for the platform can occur on Windows, Mac OS or Linux.
- **Multi-carrier support.** A large number of telecom carriers currently offer Android powered phones.

Developer Workflow

To develop apps for Android, you use a set of tools that are included in Android Studio. In addition to using the tools from Android Studio, you can also access most of the SDK tools from the command line. Developing with Android Studio is the preferred method because it can directly invoke the tools that you need while developing applications.

App Workflow

The basic steps for developing applications (with or without Android Studio) are shown in figure 1. The development steps encompass four development phases, which include:

- Environment Setup

During this phase you install and set up your development environment. You also create Android Virtual Devices (AVDs) and connect hardware devices on which you can install your applications.

See Managing Virtual Devices and Using Hardware Devices for more information.

- Project Setup and Development

During this phase you set up and develop your Android Studio project and application modules, which contain all of the source code and resource files for your application. For more information, see Create an Android project.

- Building, Debugging and Testing

During this phase you build your project into a debuggable .apk package(s) that you can install and run on the emulator or an Android-powered device. Android Studio uses a build system based on Gradle that provides flexibility, customized build variants, dependency resolution, and much more. If you're using another IDE, you can build your project using Gradle and install it on a device using adb. For more information, see Build and run your application.

Next, with Android Studio you debug your application using the Android Device Monitor and device log messages (logcat) along with the IntelliJ IDEA intelligent coding features. You can also use a JDWP-compliant debugger along with the debugging and logging tools that are provided with the Android SDK. For more information see Debug your application with the SDK debugging and logging tools.

Last, you test your application using various Android SDK testing tools. For more information, see Test your application with the Testing and Instrumentation framework.

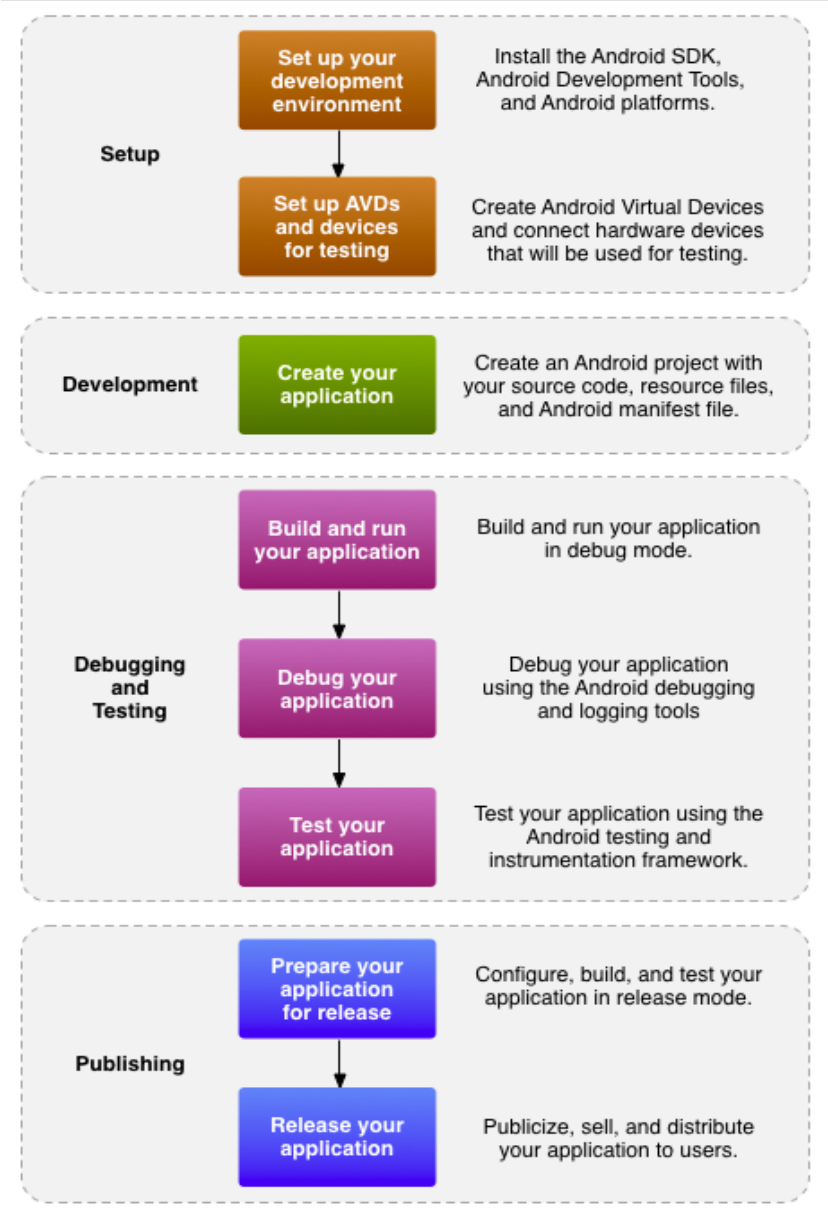

Setup
Set up your development environment
Install the Android SDK, Android Development Tools, and Android platforms.
Set up AVDs and devices for testing
Create Android Virtual Devices and connect hardware devices that will be used for testing.
Development
Create your application
Create an Android project with your source code, resource files, and Android manifest file.
Debugging and Testing
Build and run your application
Build and run your application in debug mode.
Debug your application
Debug your application using the Android debugging and logging tools
Test your application
Test your application using the Android testing and instrumentation framework.
Publishing
Prepare your application for release
Configure, build, and test your application in release mode.
Release your application
Publicize, sell, and distribute your application to users.

Figure 1

FEATURE IMPLEMENTATIONS

Navigation Drawer

The navigation drawer is a panel that displays the app's main navigation options on the left edge of the screen. It is hidden most of the time, but is revealed when the user swipes a finger from the left edge of the screen or, while at the top level of the app, the user touches the app icon in the action bar.To add a navigation drawer, declare your user interface with a DrawerLayout object as the root view of your layout. Inside the DrawerLayout, add one view that contains the main content for the screen (your primary layout when the drawer is hidden) and another view that contains the contents of the navigation drawer.For example, the following layout uses a DrawerLayout with two child views: a FrameLayout to contain the main content (populated by a Fragment at runtime), and a ListView for the navigation drawer.

```
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <!-- The main content view -->
    <FrameLayout
        android:id="@+id/content_frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
    <!-- The navigation drawer -->
    <ListView android:id="@+id/left_drawer"
        android:layout_width="240dp"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:choiceMode="singleChoice"
        android:divider="@android:color/transparent"
        android:dividerHeight="0dp"
        android:background="#111"/>
</android.support.v4.widget.DrawerLayout>
```

In your activity, one of the first things to do is initialize the navigation drawer's list of items. How you do so depends on the content of your app, but a navigation drawer often consists of a ListView, so the list should be populated by an Adapter (such as ArrayAdapter or SimpleCursorAdapter).

For example, here's how you can initialize the navigation list with a string array:

```
public class MainActivity extends Activity {
    private String[] mPlanetTitles;
    private DrawerLayout mDrawerLayout;
    private ListView mDrawerList;
    ...

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

        mPlanetTitles = getResources().getStringArray(R.array.planets_array);
        mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);
        mDrawerList = (ListView) findViewById(R.id.left_drawer);

        // Set the adapter for the list view
        mDrawerList.setAdapter(new ArrayAdapter<String>(this,
                R.layout.drawer_list_item, mPlanetTitles));
        // Set the list's click listener
        mDrawerList.setOnItemClickListener(new DrawerItemClickListener());

        ...
    }
}
```

AutocompleteTextView

An editable text view that shows completion suggestions automatically while the user is typing. The list of suggestions is displayed in a drop down menu from which the user can choose an item to replace the content of the edit box with.

The drop down can be dismissed at any time by pressing the back key or, if no item is selected in the drop down, by pressing the enter/dpad center key.

The list of suggestions is obtained from a data adapter and appears only after a given number of characters defined by the threshold.

The following code snippet shows how to create a text view which suggests various countries names while the user is typing:

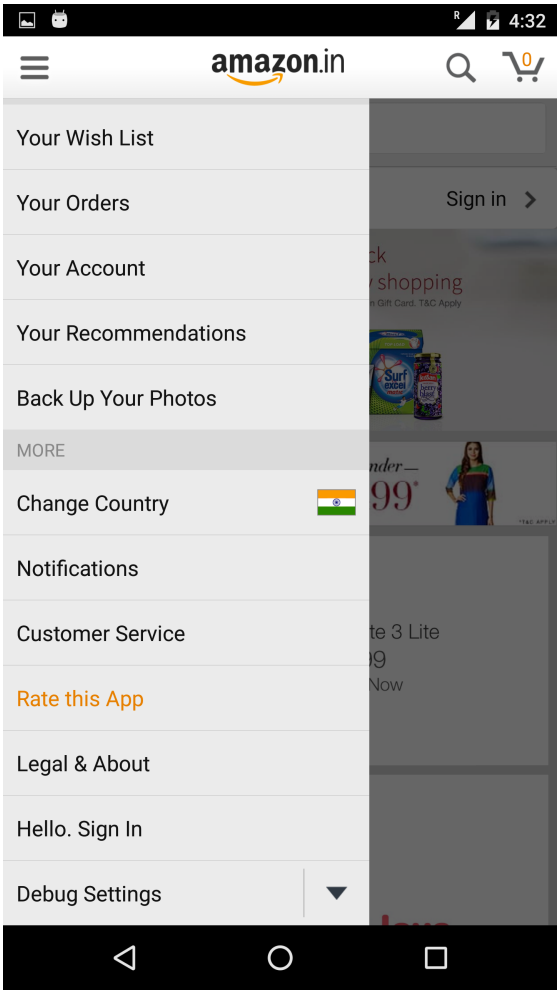
```
public class CountriesActivity extends Activity {
     protected void onCreate(Bundle icicle) {
         super.onCreate(icicle);
         setContentView(R.layout.countries);

         ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
                 android.R.layout.simple_dropdown_item_1line, COUNTRIES);
         AutoCompleteTextView textView = (AutoCompleteTextView)
                 findViewById(R.id.countries_list);
         textView.setAdapter(adapter);
     }

     private static final String[] COUNTRIES = new String[] {
         "Belgium", "France", "Italy", "Germany", "Spain"
     };
 }
```

RESULTS

The results of the implementations of the above mentioned elements, resulted in adding several features to the production app of mShop or the Amazon India Shopping app. The screenshots of the implementations are shown below:



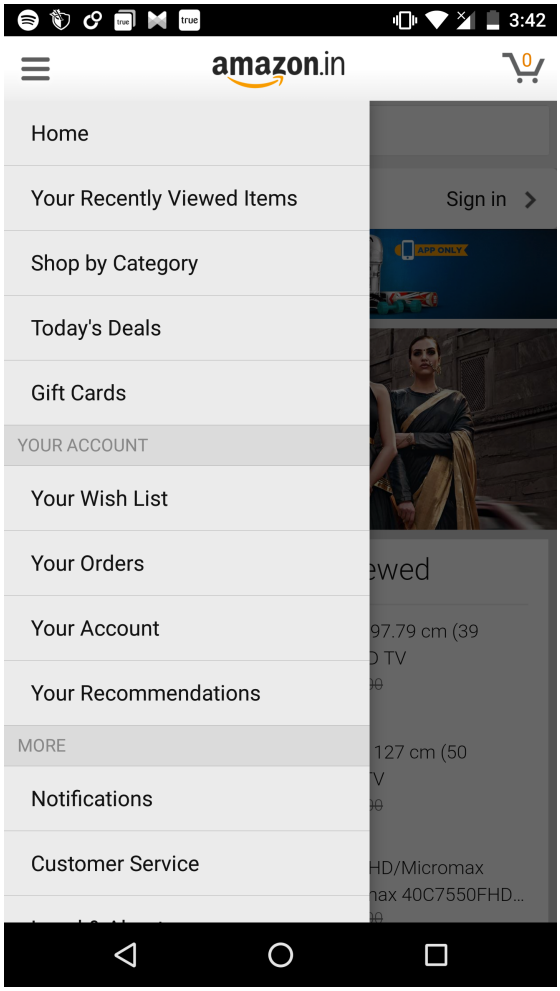


Fig 2 (left): Added a “Rate this App” entry which takes the user to the Play Store listing when clicked.
Fig 3 (right): Created a new menu called the hamburger menu, which slides out from the left.

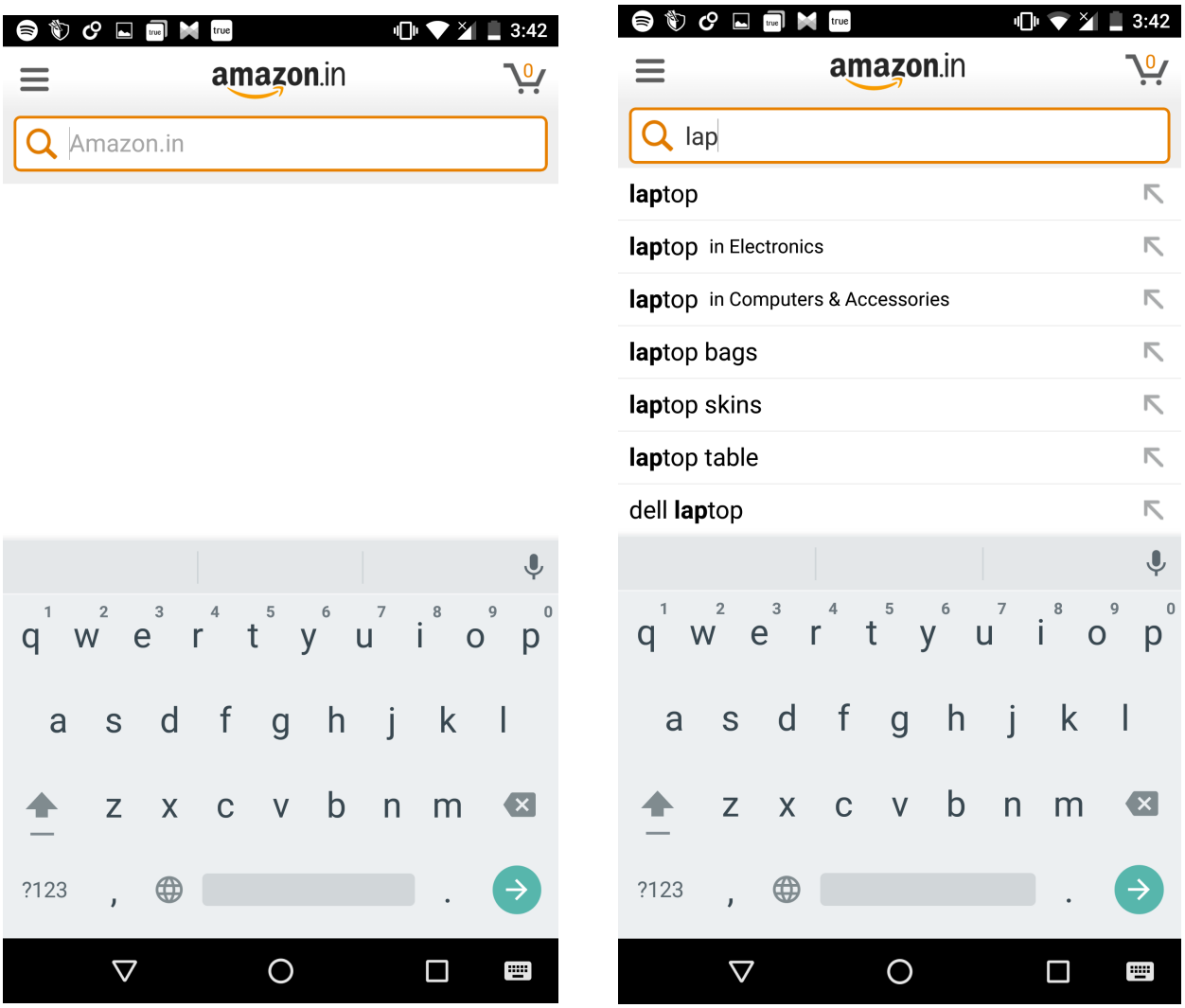


Fig 4 and 5: Implementation of Search Suggestions, which suggest the users based on the partial query entered.

CONCLUSION

The purpose of this report can be considered reached. It shows how various elements of Android SDK like the drawer layout, AutoCompleteTextView can be used to create powerful tools and widgets in the app, which improves the overall customer experience.

The project involved many more things, and could not be included in this report to maintain confidentiality. The report barely touches the surface of the technologies and tools in building these features. Also, many features that were implemented could not be described in this report to maintain confidentiality. Overall, the project has been of great interest so far, and proved to be a great learning experience.

REFERENCES

1. http://developer.android.com/reference/android/widget/AutoCompleteTextView.html

2. http://makovkastar.github.io/blog/2014/04/12/android-autocompletetextview-with-suggestions-from-a-web-service/

3. https://www.codeofaninja.com/2013/11/android-autocompletetextview-example-sqlite-database.html

4. http://www.androidinterview.com/how-to-use-android-autocompletetextview-example/

5. http://developer.android.com/guide/topics/providers/content-providers.html

6. http://codetheory.in/android-navigation-drawer/