1. INTRODUCTION

1.1. PROBLEM DEFINITION:

The **“Eatery Finder”** is an android mobile application where users can install the application and register with their username, Email id, phone no, password. Users can then login into the site with registered username and password and can access the services of the application like

- Search for any kind of restaurant situated at current location.
- Users can long press the location of restaurant in the map that enables a bookmark of the corresponding eatery.
- Users can view menus, pictures, phone numbers, directions, user reviews, and all the other information you need to choose one among many by selecting the relevant bookmark.

1.2 EXISTING SYSTEM

In this contemporary situation, many applications are providing a best way to find the foodie bays which enables the users to eat at (or) order online. In this context, we can quote “Zomato” as an example which is easy-to-use social restaurant finder app that lets us to explore all dining options in the city.

However, these kinds of applications are restricted to affluent restaurants which are not quite suitable to abundant set of people especially college goers.

On top of that, these substantial eateries may be located at the heart of the city which leads to trouble for the people who live in distant places.

1.3 PROPOSED SYSTEM

To resolve such issues, this application is intended to discover localized eateries that are reasonable, not-so-familiar and where the fresh and delicious food is available which would be an added an added advantage. With the help of this strategy, the hotels located in deserted and remote areas comes into limelight and thus results in decline of strength of population going to opulent restaurants and encourages local vendors to set and grow their business.

1.4 SOFTWARE AND HARDWARE SPECIFICATION

Software:

Operating system	:	API19: Android 4.4.2 (or) later
Emulator	:	Android SDK Emulator
Development Tools	:	NetBeans IDE, Android Studio with SDK Manager

Hardware:

Device	:	Mobile Phone [Android Enabled]
Processor	:	1 GHZ +
RAM	:	2 GB
SD Memory	:	1 GB

2. LITERATURE

2.1 Android

Operating Systems have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android.

One of the most widely used mobile OS these days is **ANDROID**. **Android** is a software bunch comprising not only operating system but also middleware and key applications. Android Inc. was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later Android Inc. was acquired by Google in 2005. After original release there have been number of updates in the original version of Android.

Android is a powerful Operating System supporting many applications in Smart Phones. These applications make life more comfortable and advanced for the users. Hardware that supports Android are mainly based on ARM architecture platform.

Android comes with an Android market which is an online software store. It was developed by Google. It allows Android users to select, and download applications developed by third party developers and use them. There are around 2.0 lack+ games, application, and widgets available on the market for users.

Android applications are written in java programming language. Android is available as open source for developers to develop applications which can be further used for selling in android market. There are around 200000 applications developed for android with over 3 billion+ downloads. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. For software development, Android provides **Android SDK** (Software development kit).

Applications

These are the basics of Android applications:

• Android applications are composed of one or more application components (activities, services, content providers, and broadcast receivers)

• Each component performs a different role in the overall application behavior, and each one can be activated individually (even by other applications)

• The manifest file must declare all components in the application and should also declare all application requirements, such as the minimum version of Android required, and any hardware configurations required

Non-code application resources (images, strings, layout files, etc.) should include alternatives for different device configurations (such as different strings for different languages)

Google, for software development and application development, had launched two competitions ADC1 and ADC2 for the most innovative applications for Android. It offered prizes of USD 10 million combined in ADC1 and 2. ADC1 was launched in January 2008 and ADC 2 was launched in May 2009. These competitions helped Google a lot in making Android better, more user friendly, advanced, and interactive.

We will be glad to know that we can start your Android application development on either of the following

Operating systems:

➔ Microsoft Windows XP or later version.

➔ Mac OS X 10.5.8 or later version with Intel chip.

➔ Linux including GNU C Library 2.7 or later.

Second point is that all the required tools to develop Android applications are freely available and can be downloaded from the Web. Following is the list of software's we will need before you start your Android application programming.

➔ Java JDK5 or JDK6

➔ Android SDK

➔ NetBeans IDE for Java Developers (optional)

➔ Android Development Tools (ADT) (optional)

Here last two components are optional and if we are working on Windows machine then these components make our life easy while doing Java based application development. So let us have a look how to proceed to set required environment.

Step 1 - Setup Java Development Kit (JDK)

You can download the latest version of Java JDK from Oracle's Java site. You will find instructions for installing JDK in downloaded files, follow the given instructions to install and configure the setup.

Finally set PATH and JAVA_HOME environment variables to refer to the directory that contains java and javac, typically java_install_dir/bin and java_install_dir respectively.

If we are running Windows and installed the JDK in C:\jdk1.6.0_15, we would have to put the following line in your C:\autoexec.bat file.

```
set PATH=C:\jdk1.6.0_15\bin;%PATH%
set JAVA_HOME=C:\jdk1.6.0_15
```

Alternatively, we could also right-click on My Computer, select Properties, then Advanced, then Environment Variables. Then, you would update the PATH value and press the OK button.

On Linux, if the SDK is installed in /usr/local/jdk1.6.0_15 and you use the C shell, you would put the following code into your .cshrc file.

```
setenv PATH /usr/local/jdk1.6.0_15/bin:$PATH
setenv JAVA_HOME /usr/local/jdk1.6.0_15
```

Alternatively, if we use an Integrated Development Environment (IDE) NetBeans, then it will know automatically where you have installed your Java.

Step 2 – Install NetBeans IDE

Go to **https://netbeans.org/downloads/**.You can also choose to download and use the platform-independent zip file. Click the Download button for the download option that you want to install. Save the installer file to your system. For Windows, the installer executable file has the .exe extension. Double-click the installer file to run it. At the Welcome page of the installation, click Next. Accept the license agreement and click Next. At the NetBeans IDE installation page, do the following [2]:

1. Accept the default installation directory for the NetBeans IDE or specify another directory.
2. Accept the default JDK installation to use with the NetBeans IDE

Step 3 - Install Android Studio

Navigate to the Android developers' site to install Android Studio. This page will automatically detect your operating system.
Accept the terms and conditions to start the download. Double-click the downloaded file and follow all the prompts [3].

Step 4 – Install Android System Images & Tools

Open Android Studio and click the SDK Manager button on the toolbar. After the SDK Manager has fully loaded, close Android Studio.
We are building for Android 4.4.2 so make sure the following packages are checked under the Tools section:

- Android SDK Tools rev 22.6.3
- Android Platform-tools rev 19.0.1
- Android SDK Build-tools rev 19.1
- Check these under the Android 4.4.2(API 19) section:
- Intel x86 Atom System Image
- Google APIs (x86 System Image)
- And check these under the Extras section:
- Android Support Repository
- Android Support Library Click Install

Accept the licenses that appear for each section to enable download.

Step 5 - Create Android Virtual Device

To test your Android applications you will need a virtual Android device. So before we start writing our code, let us create an Android virtual device. Launch Android AVD Manager using Android menu options **Tools > Android > AVD Manager.** You can also start the AVD Manager from the command line by calling the android tool with the avd options, from the **<sdk>/tools/** directory. You can create as many AVDs as you would like to use with the Android Emulator.
To create an AVD based on an existing device definition:

1 From the main screen, click **Create Virtual Device**.
2 In the Select Hardware window, select a device configuration, such as Nexus 6, then, click **Next**.
3 Select the desired system version for the AVD and click **Next**.
4 Verify the configuration settings, then click **Finish**.

If necessary, click **Show Advanced Settings** to select a custom skin for the hardware profile and adjust other hardware settings.To launch the AVD in the Android Emulator, click the launch button ▶ in the list of AVDs.

2.2 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 (Figure 2.1)[4]. These four layers are described as follows:

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.

Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine Web Kit, 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.

Android Libraries:

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access. A summary of some key core Android libraries available to the Android developer is as follows −

- **android.app** − Provides access to the application model and is the cornerstone of all Android applications.

- **android.content** − Facilitates content access, publishing and messaging between applications and application components.

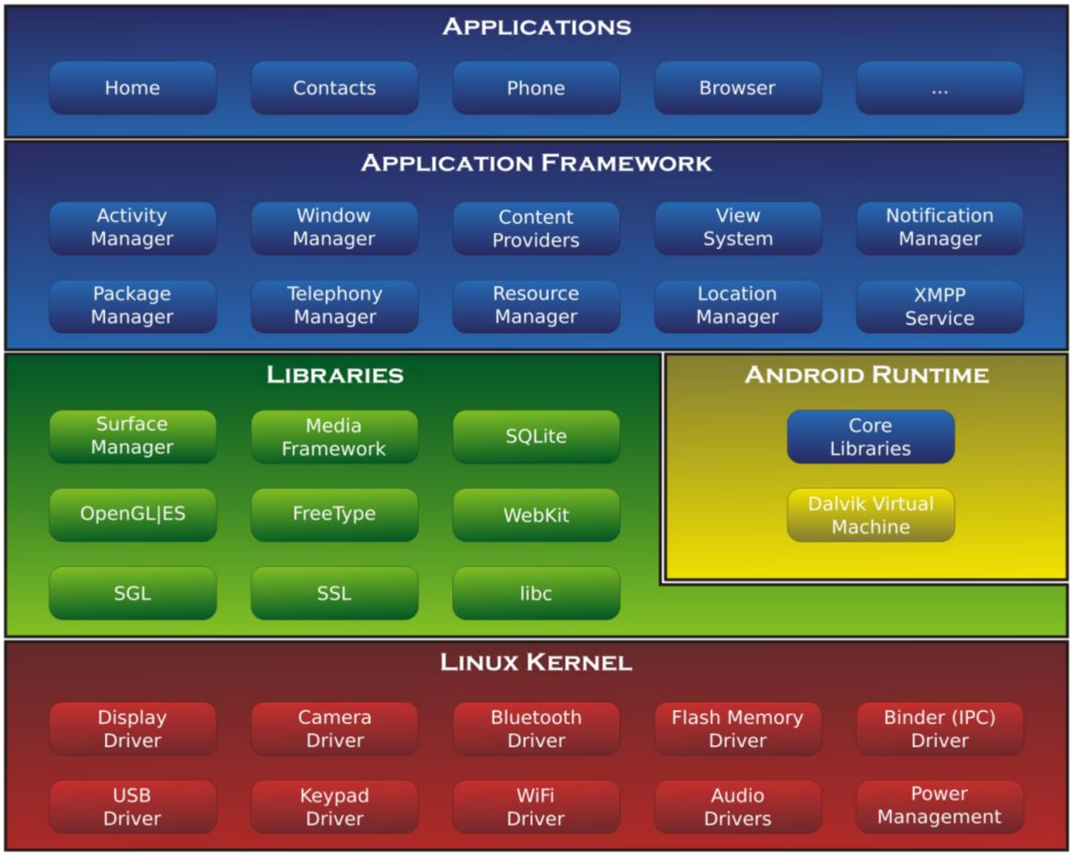


Figure 2.1 Android Architecture

- **android.support --** The Android Support Library package contains several libraries. Each of these libraries supports a specific range of Android platform versions and set of features.

- **android.os** − Provides applications with access to standard operating system services including messages, system services and inter-process communication.

- **android.view** – The fundamental building blocks of application user interfaces.

- **android.widget** − A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.

Android 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.

Application Framework

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.
The Android framework includes the following key services –

- **Activity Manager** − Controls all aspects of the application lifecycle and activity stack.
- **Content Providers** − Allows applications to publish and share data with other applications.
- **Resource Manager** − Provides access to non-code embedded resources such as strings, color settings and user interface layouts.
- **Notifications Manager** − Allows applications to display alerts and notifications to the user.
- **View System** − An extensible set of views used to create application user interfaces.

Applications

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, and Games etc.

Application Components:

Application components are the essential building blocks of an Android application. These components are loosely coupled by the application manifest file AndroidManifest.xml that describes each component of the application and how they interact.

There are following two main components that can be used within an Android application:

Components	**Description**
Activities	They dictate the UI and handle the user interaction to the smart phone screen
Content Providers	They handle data and database management issues.

Activities

An activity represents a single screen with a user interface. For example, an email application might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched. An activity is implemented as a subclass of Activity class as follows:

```
public class MainActivity extends Activity { }
```

Content Providers

A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the ContentResolver class. The data may be stored in the file system, the database or somewhere else entirely.

A content provider is implemented as a subclass of ContentProvider class and must implement a standard set of APIs that enable other applications to perform transactions.

```
public class MyContentProvider extends ContentProvider {
        public void onCreate(){ }        }
```

Additional Components

There are additional components which will be used in the construction of above mentioned entities, their logic, and wiring between them. These components are:

Components	Description
Fragments	Represents a portion of user interface in an Activity.
Views	UI elements that are drawn on-screen including buttons, lists forms etc.
Layouts	View hierarchies that control screen format and appearance of the views.
Intents	Messages wiring components together.
Resources	External elements, such as strings, constants and drawable pictures.
Manifest	Configuration file for the application.

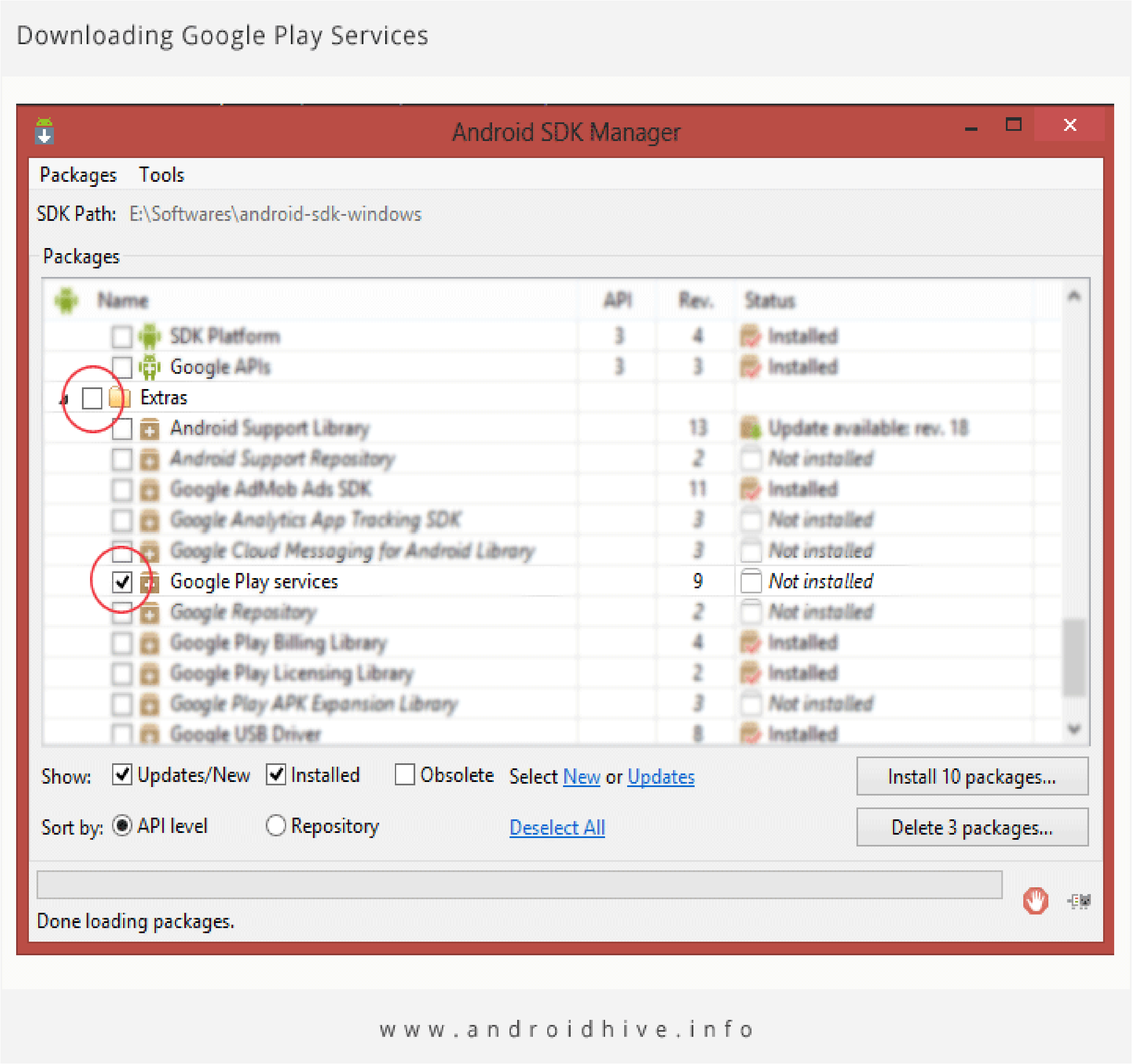
2.3 GOOGLE MAPS

Android allows us to integrate google maps in our application. You can show any location on the map, or can show different routes on the map etc. You can also customize the map according to your choices.

2.3.1 Downloading Google Play Services

Google made new Maps V2 API as a part of “Google play services” SDK. So, before we start developing maps we need to download google play services from SDK manger. You can open SDK manager either from Eclipse or from android sdk folder.

Open **Eclipse ⇒ Windows ⇒ Android SDK Manager** and check whether you have already downloaded Google Play Services or not under **Extras** section. If not select play services and install the package.



SDK Path: E:\Softwares\android-sdk-windows

Packages

Name	API	Rev.	Status
□ SDK Platform	3	4	Installed
□ Google APIs	3	3	Installed
□ Extras			
□ Android Support Library		13	Update available: rev. 18
□ Android Support Repository		2	Not installed
□ Google AdMob Ads SDK		11	Installed
□ Google Analytics App Tracking SDK		3	Not installed
□ Google Cloud Messaging for Android Library		3	Not installed
☑ Google Play services		9	Not installed
□ Google Repository		2	Not installed
□ Google Play Billing Library		4	Installed
□ Google Play Licensing Library		2	Installed
□ Google Play APK Expansion Library		3	Not installed
□ Google USB Driver		8	Installed

Show: ☑ Updates/New ☑ Installed □ Obsolete Select New or Updates [Install 10 packages...]

Sort by: ● API level ○ Repository Deselect All [Delete 3 packages...]

Done loading packages.

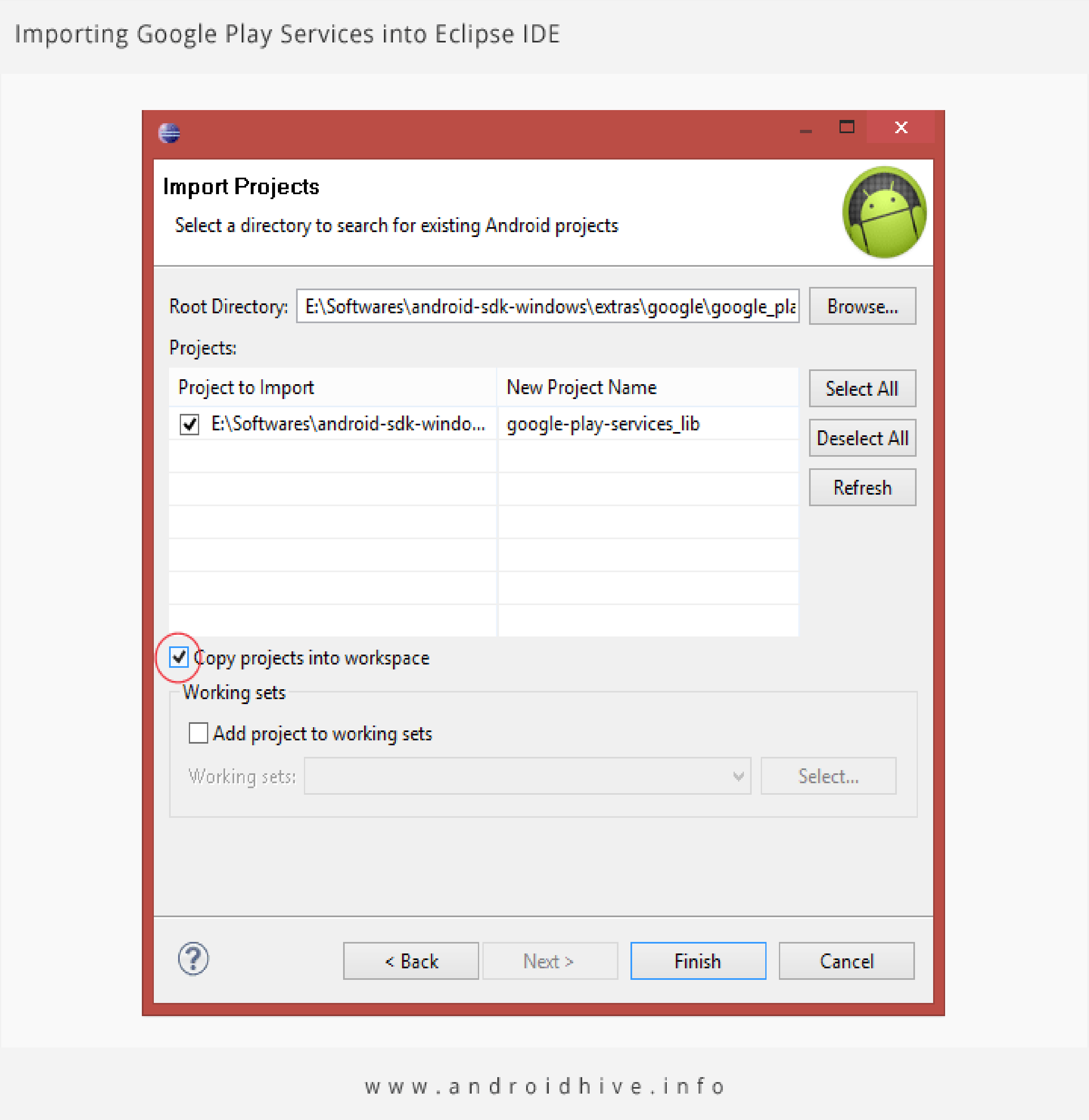
2.3.2. Importing Google Play Services into Eclipse

After downloading play services, we need to import it to Eclipse which will be used as a library for our maps project.

1. In Eclipse go to **File ⇒ Import ⇒ Android ⇒ Existing Android Code into Workspace**
2. Click on Browse and select Google Play Services project from your android sdk folder. You can locate play services library project from

android-sdk-windows\extras\google\google_play_services\libproject\google-play-services_lib

3. Importantly while importing check **Copy projects into workspace** option as shown in the below image.



Importing Google Play Services into Eclipse IDE

Import Projects

Select a directory to search for existing Android projects

Root Directory: E:\Softwares\android-sdk-windows\extras\google\google_pla

Projects:

Project to Import	New Project Name
☑ E:\Softwares\android-sdk-windo...	google-play-services_lib

☑ Copy projects into workspace

Working sets

□ Add project to working sets

Working sets:

2.3.3. Getting the Google Maps API key

1. Same as in maps v1 we need to generate SHA-1 fingerprint using java `keytool`. Open your terminal and execute the following command to generate SHA-1 fingerprint.

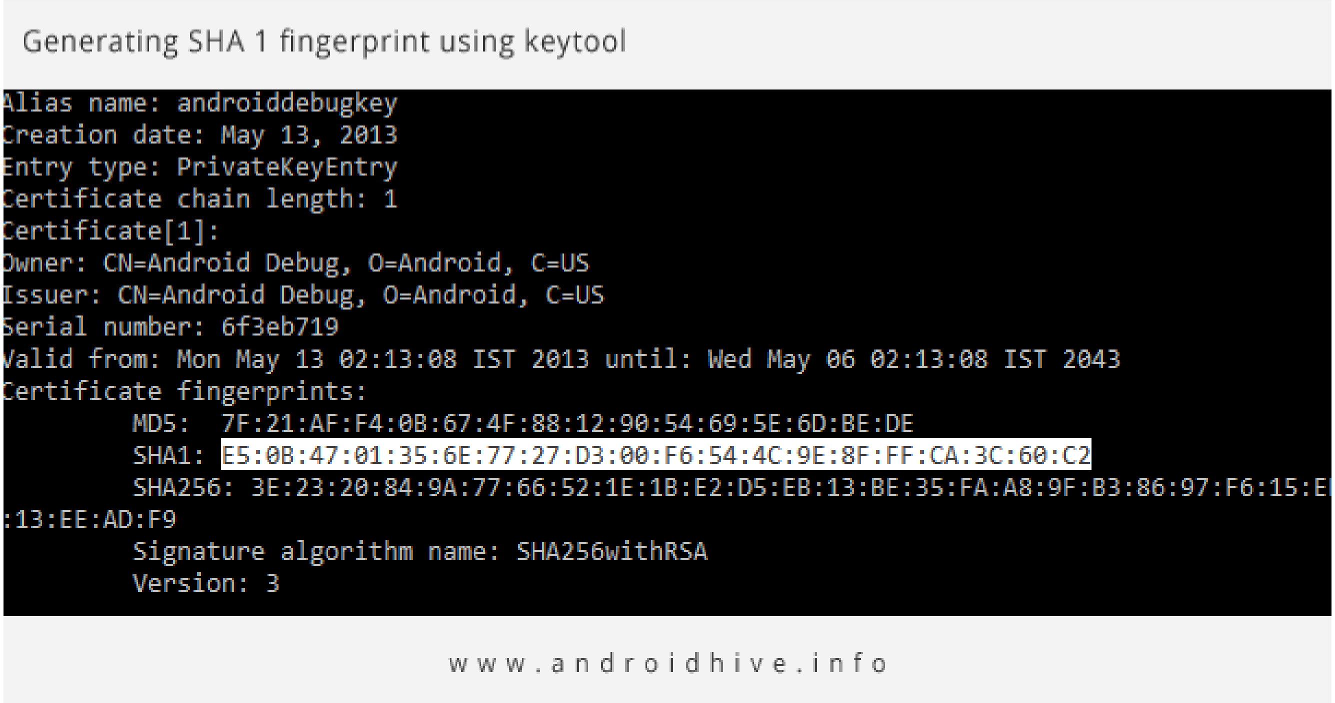
On Windows

keytool -list -v -keystore "%USERPROFILE%\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass

On Linux or Mac OS

keytool -list -v -keystore ~/.android/debug.keystore -alias androiddebugkey -storepass android -keypass android

In the output you can see SHA 1 finger print.



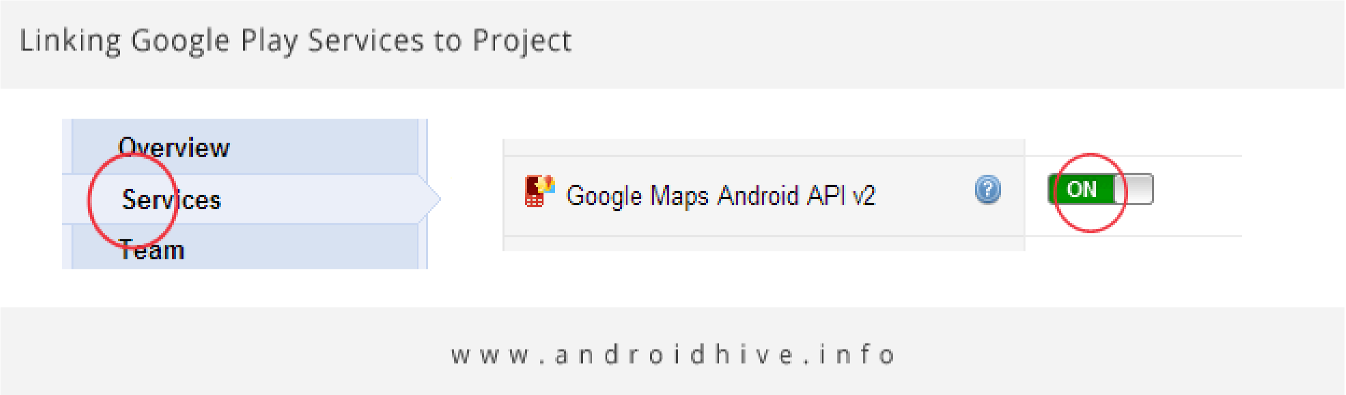
Generating SHA 1 fingerprint using keytool

```
Alias name: androiddebugkey
Creation date: May 13, 2013
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 6f3eb719
Valid from: Mon May 13 02:13:08 IST 2013 until: Wed May 06 02:13:08 IST 2043
Certificate fingerprints:
         MD5:  7F:21:AF:F4:0B:67:4F:88:12:90:54:69:5E:6D:BE:DE
         SHA1: E5:0B:47:01:35:6E:77:27:D3:00:F6:54:4C:9E:8F:FF:CA:3C:60:C2
         SHA256: 3E:23:20:84:9A:77:66:52:1E:1B:E2:D5:EB:13:BE:35:FA:A8:9F:B3:86:97:F6:15:E
:13:EE:AD:F9
         Signature algorithm name: SHA256withRSA
         Version: 3
```

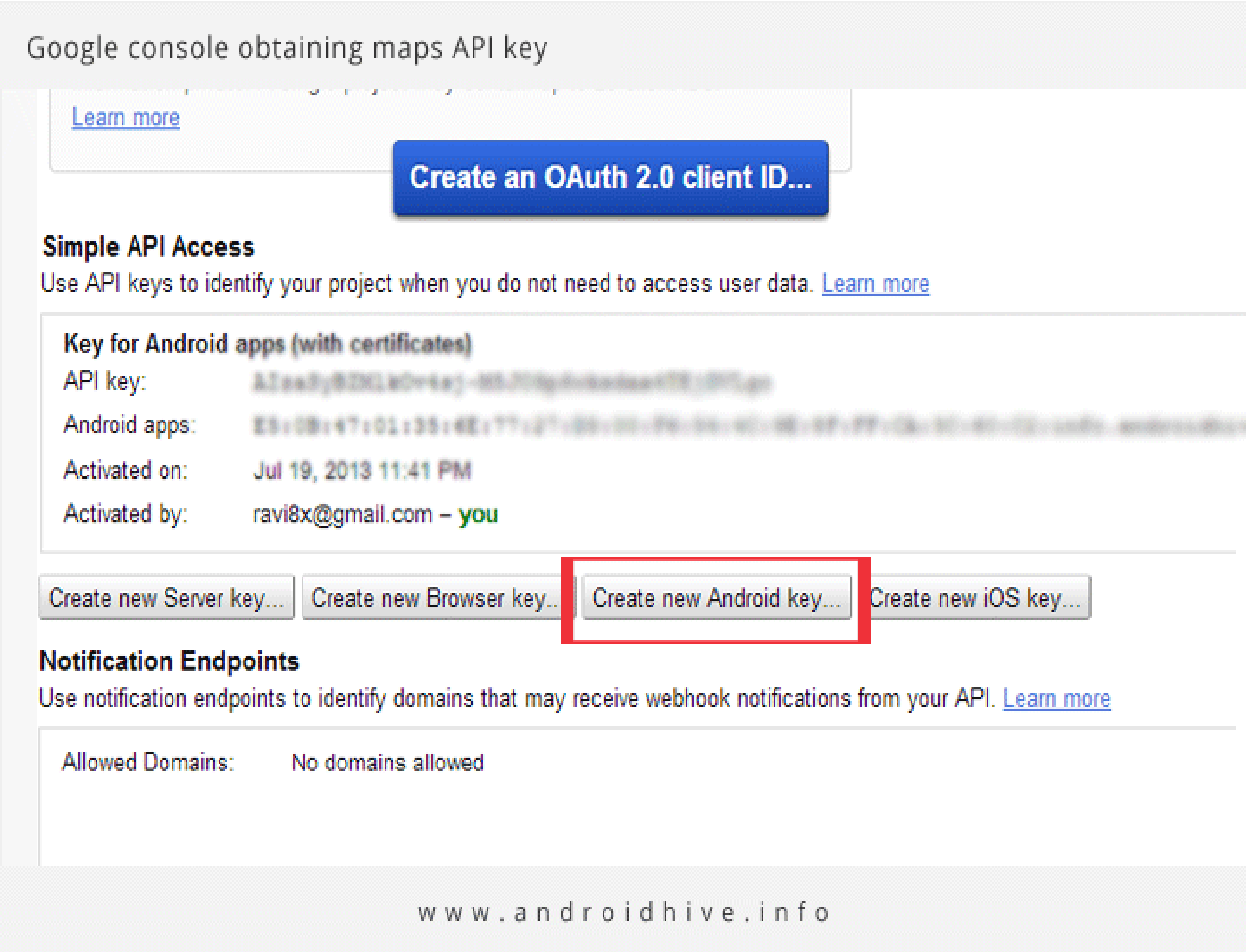
www.androidhive.info

2. Now open Google API Console

3. Select **Services** on left side and turn on **Google Maps Android API v2**


Linking Google Play Services to Project
Overview
Services
Team
Google Maps Android API v2
ON
www.androidhive.info

4. Now select **API Access** on left side and on the right side click on **Create new Android key…**


Google console obtaining maps API key
Learn more
Create an OAuth 2.0 client ID...

Simple API Access

Use API keys to identify your project when you do not need to access user data. Learn more

Key for Android apps (with certificates)

API key:

Android apps:

Activated on: Jul 19, 2013 11:41 PM

Activated by: ravi8x@gmail.com – you

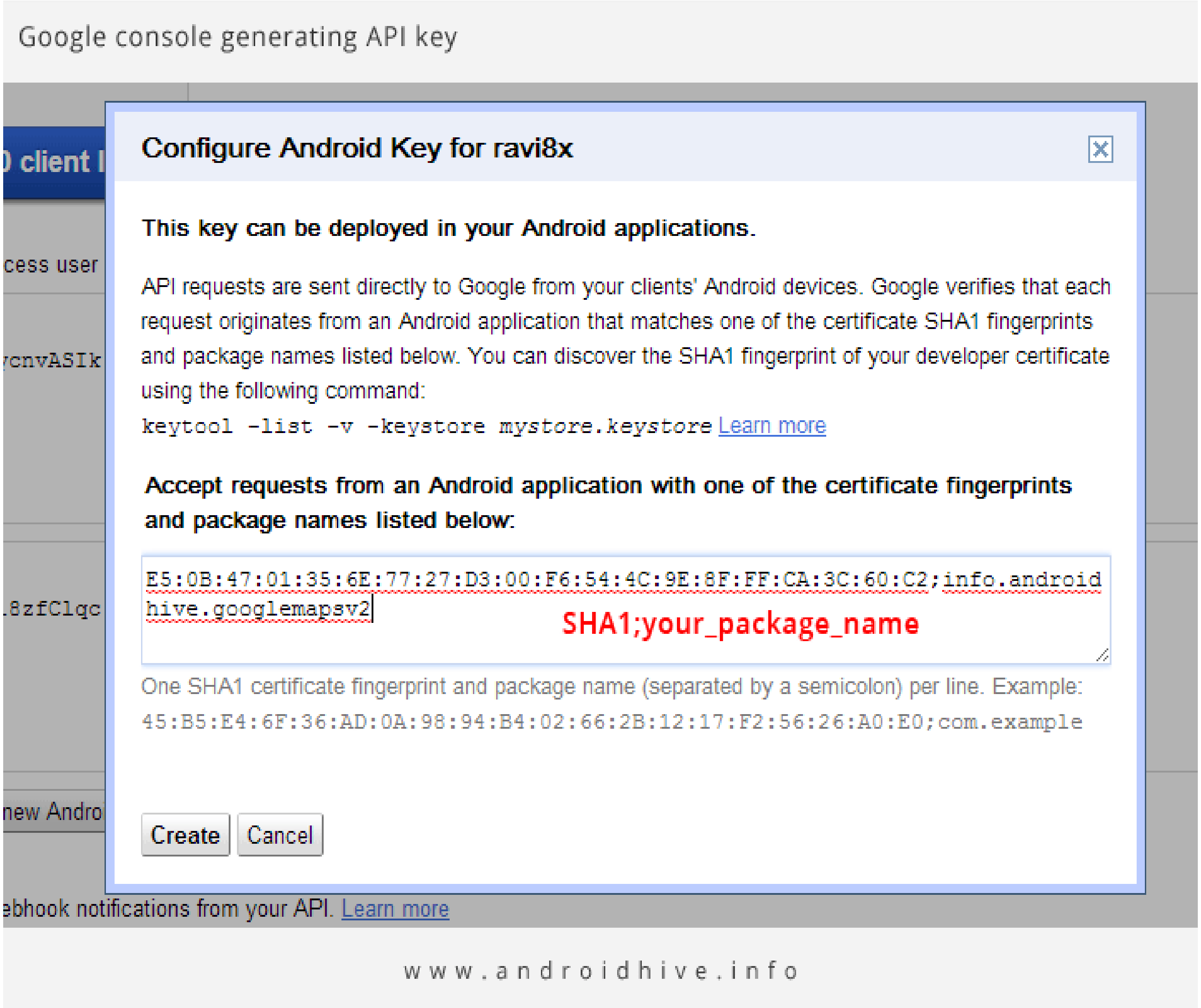
Create new Server key...
Create new Browser key...
Create new Android key...
Create new iOS key...

Notification Endpoints

Use notification endpoints to identify domains that may receive webhook notifications from your API. Learn more

Allowed Domains: No domains allowed

5. It will pop up a window asking the SHA1 and package name. Enter your SHA 1 and your **android project package name** separated by semicolon; and click on create.



Google console generating API key

Configure Android Key for ravi8x

This key can be deployed in your Android applications.

API requests are sent directly to Google from your clients' Android devices. Google verifies that each request originates from an Android application that matches one of the certificate SHA1 fingerprints and package names listed below. You can discover the SHA1 fingerprint of your developer certificate using the following command:

```
keytool -list -v -keystore mystore.keystore
```

Learn more

Accept requests from an Android application with one of the certificate fingerprints and package names listed below:

```
E5:0B:47:01:35:6E:77:27:D3:00:F6:54:4C:9E:8F:FF:CA:3C:60:C2;info.android
hive.googlemapsv2
```

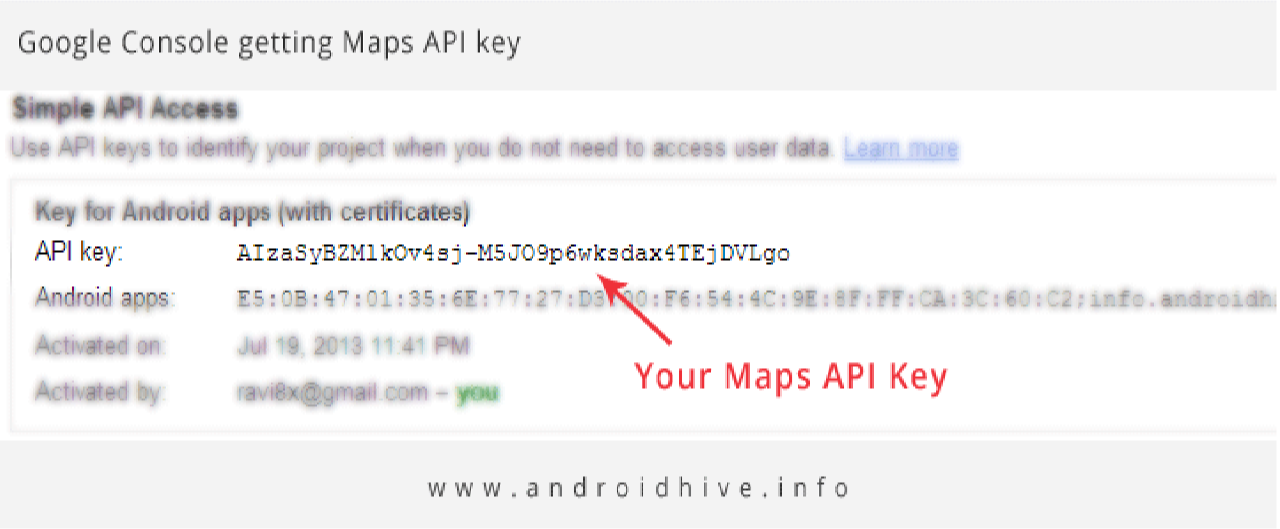
SHA1;your_package_name

One SHA1 certificate fingerprint and package name (separated by a semicolon) per line. Example:

```
45:B5:E4:6F:36:AD:0A:98:94:B4:02:66:2B:12:17:F2:56:26:A0:E0;com.example
```

BE:03:E1:44:39:7B:E8:17:02:9F:7F:B7:98:82:EA:DF:84:D0:FB:6A;info.androidhive.googlemapsv2

And note down the API key which required later in our project.



Google Console getting Maps API key

Simple API Access

Use API keys to identify your project when you do not need to access user data. Learn more

Key for Android apps (with certificates)

API key: AIzaSyBZMlkOv4sj-M5JO9p6wksdax4TEjDVLgo

Android apps: E5:0B:47:01:35:6E:77:27:D3:00:F6:54:4C:9E:8F:FF:CA:3C:60:C2;info.androidh

Activated on: Jul 19, 2013 11:41 PM

Activated by: ravi8x@gmail.com – you

Your Maps API Key

www.androidhive.info

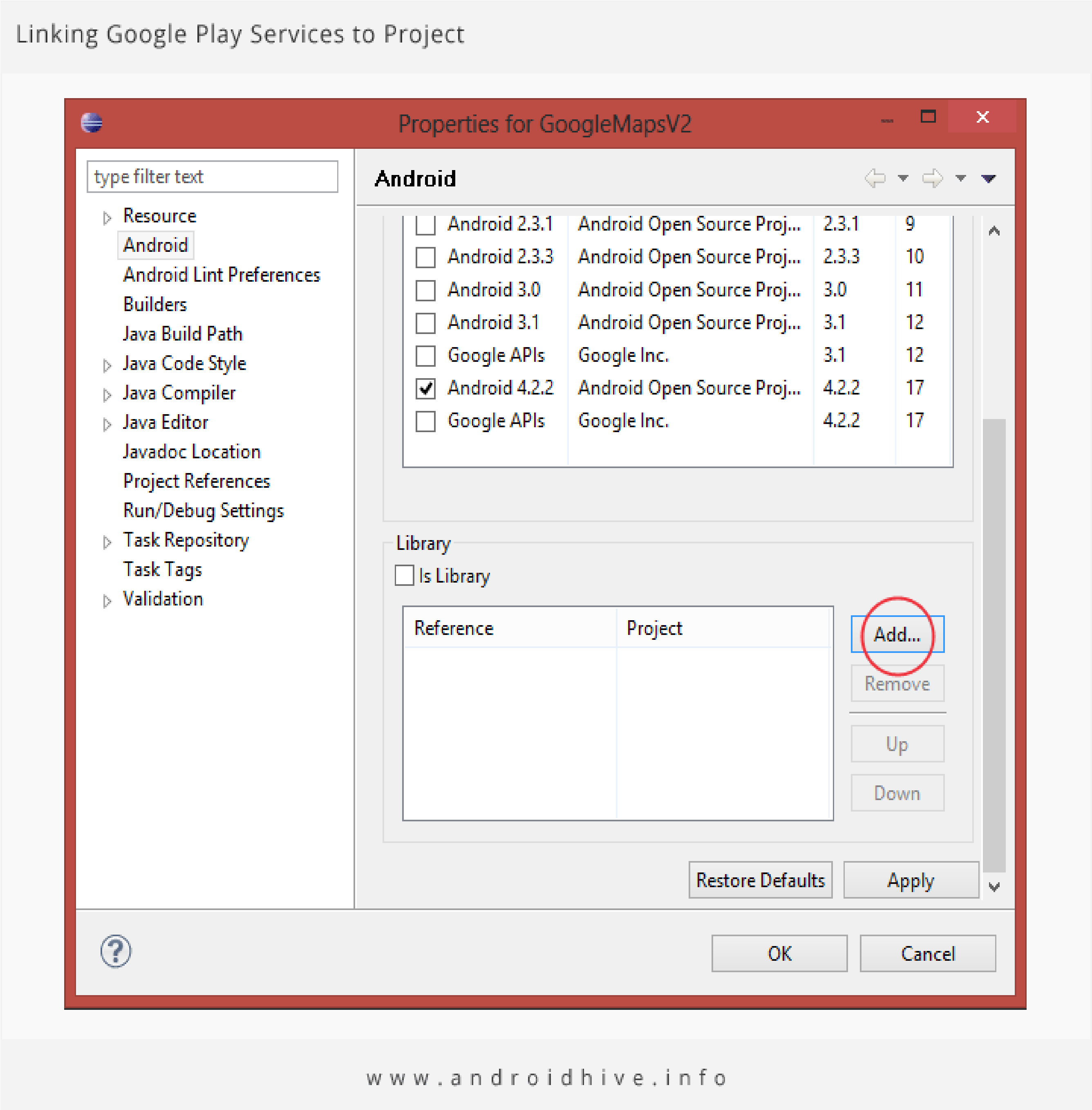
2.4.4. Creating new Project

After completing required configuration, It's time to start our project.

1. In Eclipse create a new project by going to **File ⇒ New ⇒ Android Application Project** and fill required details. I kept my project name as Google maps V2 and package name as info.androidhive.info.

2. Now we need to use Google Play Services project as a library to use project. So **right click** on project and select **properties**. In the properties window on left side select **Android**. On the right you can see a **Add** button under library section. Click it and select **google play services** project which we imported previously.

Linking Google Play Services to Project



Properties for GoogleMapsV2

Android

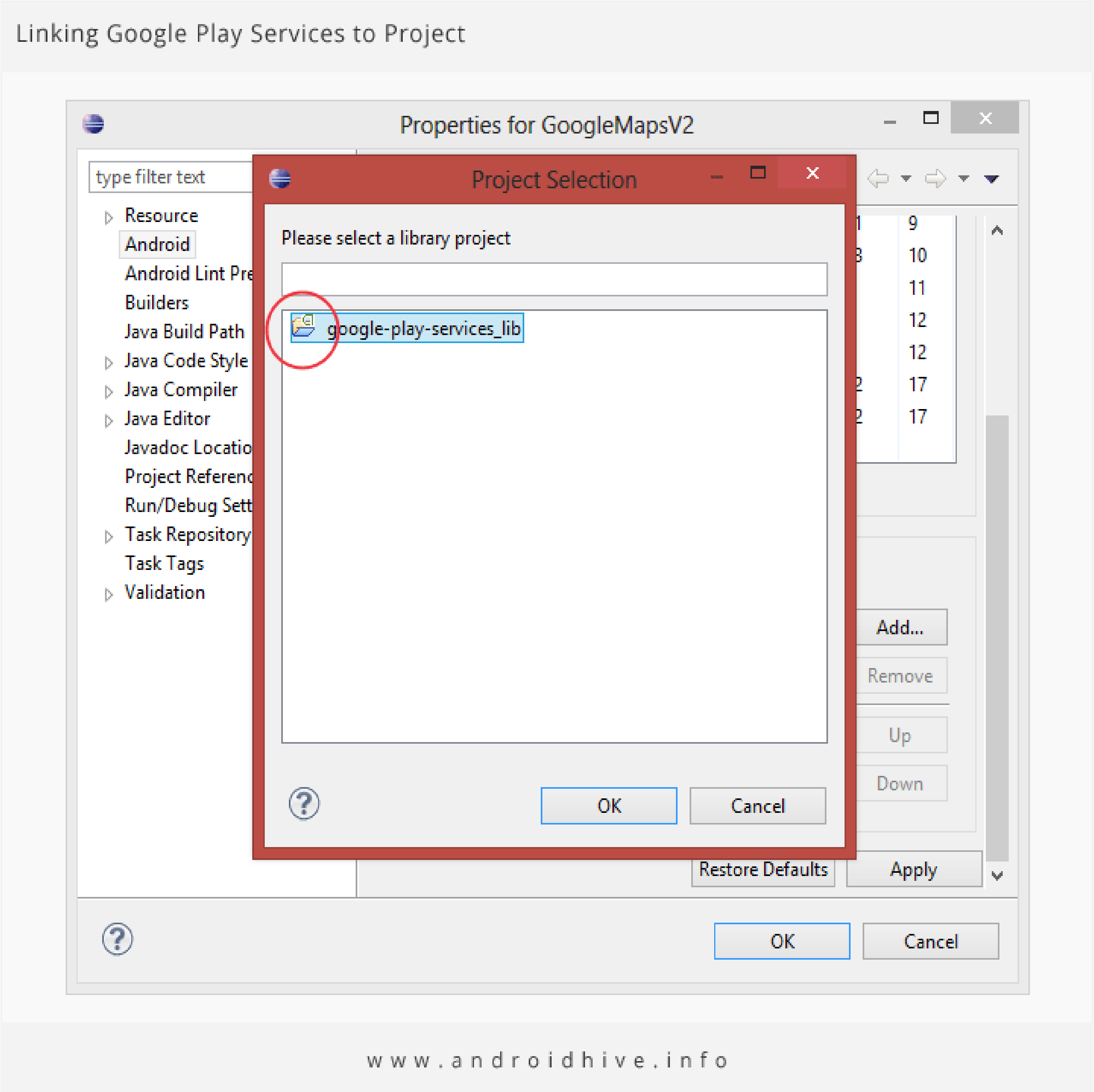
□	Android 2.3.1	Android Open Source Proj...	2.3.1	9
□	Android 2.3.3	Android Open Source Proj...	2.3.3	10
□	Android 3.0	Android Open Source Proj...	3.0	11
□	Android 3.1	Android Open Source Proj...	3.1	12
□	Google APIs	Google Inc.	3.1	12
☑	Android 4.2.2	Android Open Source Proj...	4.2.2	17
□	Google APIs	Google Inc.	4.2.2	17

Library

□ Is Library

Reference	Project

www.androidhive.info

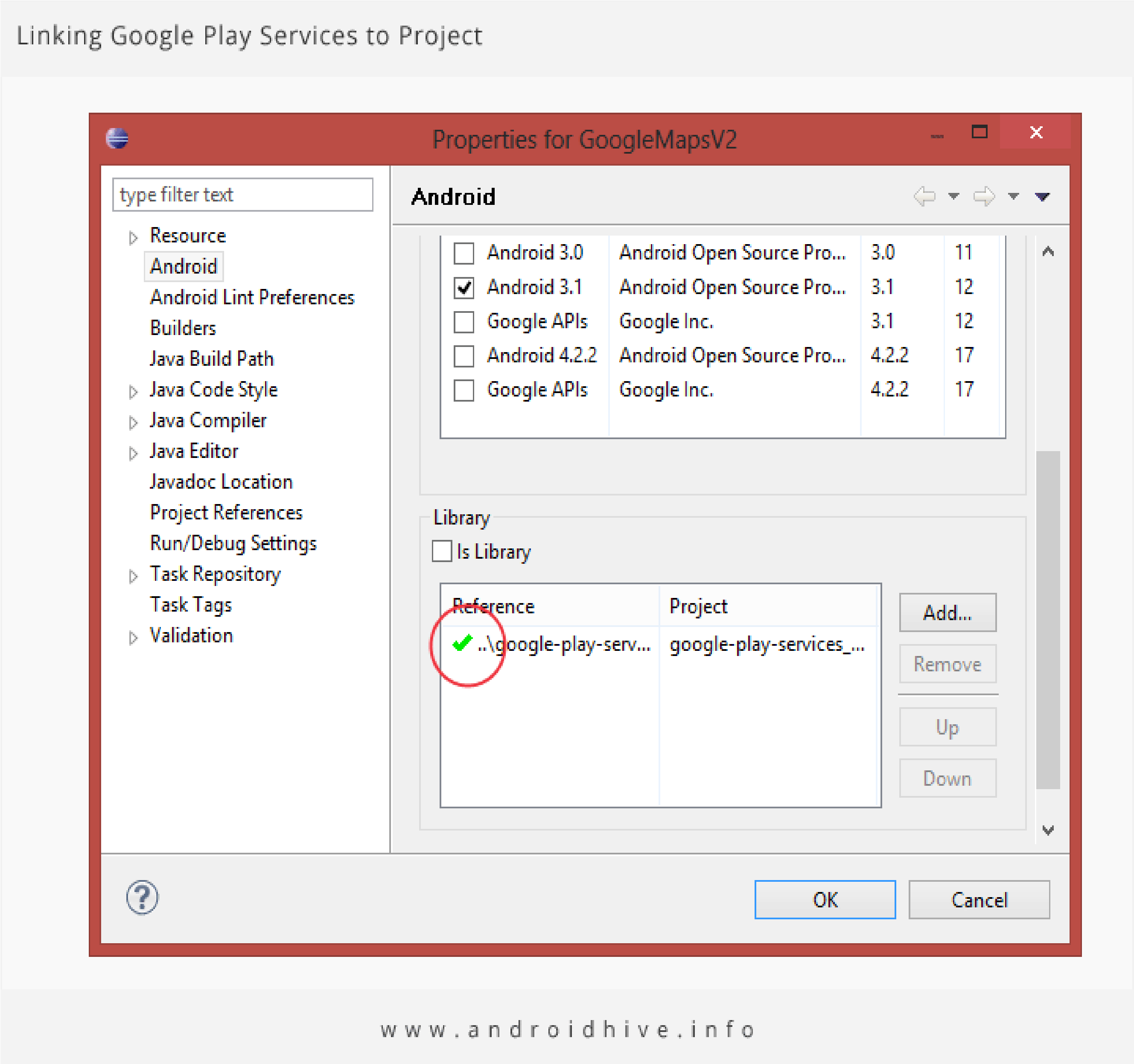


Linking Google Play Services to Project

Project Selection

Please select a library project

google-play-services_lib



Linking Google Play Services to Project

□ Android 3.0	Android Open Source Pro...	3.0	11
☑ Android 3.1	Android Open Source Pro...	3.1	12
□ Google APIs	Google Inc.	3.1	12
□ Android 4.2.2	Android Open Source Pro...	4.2.2	17
□ Google APIs	Google Inc.	4.2.2	17

Library

□ Is Library

Reference	Project
✓ ..\google-play-serv...	google-play-services_...

www.androidhive.info

3. Add the Map Key in the manifest file. Open **AndroidManifest.xml** file and add the following code before tag. Replace **android:value** with your map key which you got from google console.

```
<!-- Goolge Maps API Key -->
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
     android:value="AIzaSyBZMlkOv4sj-M5JO9p6wksdax4TEjDVLgo" />
```

4. Google maps needs following permissions and features.

ACCESS_NETWORK_STATE – To check network state whether data can be downloaded or not.

INTERNET – To check internet connection status.

WRITE_EXTERNAL_STORAGE - To write to external storage as google maps store map data in external storage.

ACCESS_COARSE_LOCATION – To determine user's location using WIFI and mobile cell data.

ACCESS_FINE_LOCATION - To determine user's location using GPS.

OPENGL ES V2 - Required for Google Maps V2.

3. DESIGN AND ANALYSIS

3.1 Unified Modeling Language (UML)

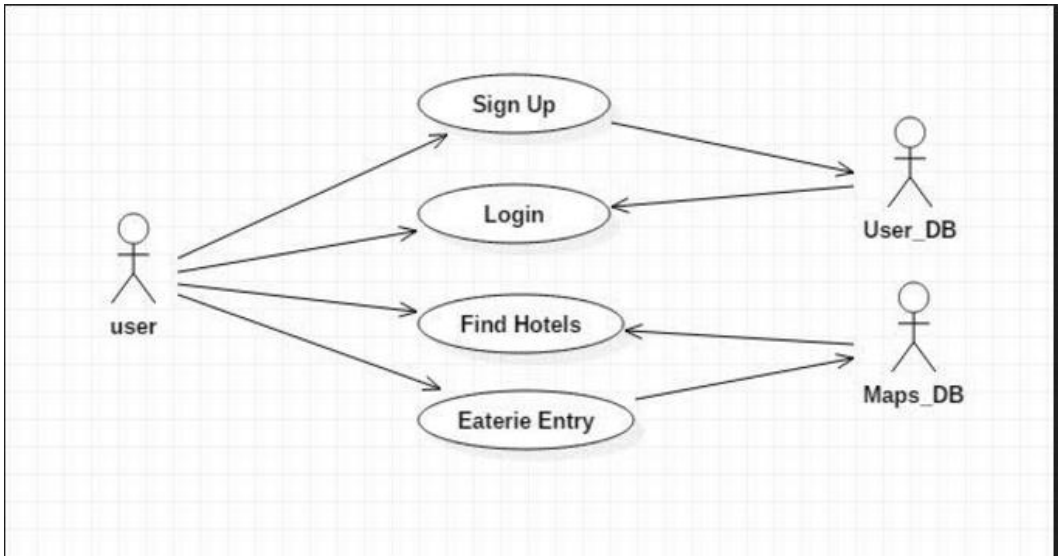
In this chapter we are going to use UML to describe and generate a blueprint for our system. UML will be used for the following purposes.

- **Visualization** – This is used as a standardized tool to write and clearly specify a model for the system being developed.
- **Specifying**– Models built using UML are precise, unambiguous and complete that leave no element of doubt for the developer.
- **Constructing** – UML is object oriented in nature which means that after generation of these you can directly convert to code.
- **Documenting** – These UML diagrams also serve as documentation for the system’s architecture and intrinsic details.

3.2 Assumptions and Dependencies

- The end-user must have a mobile phone.
- The mobile phone used by the end-user must have access to the internet.

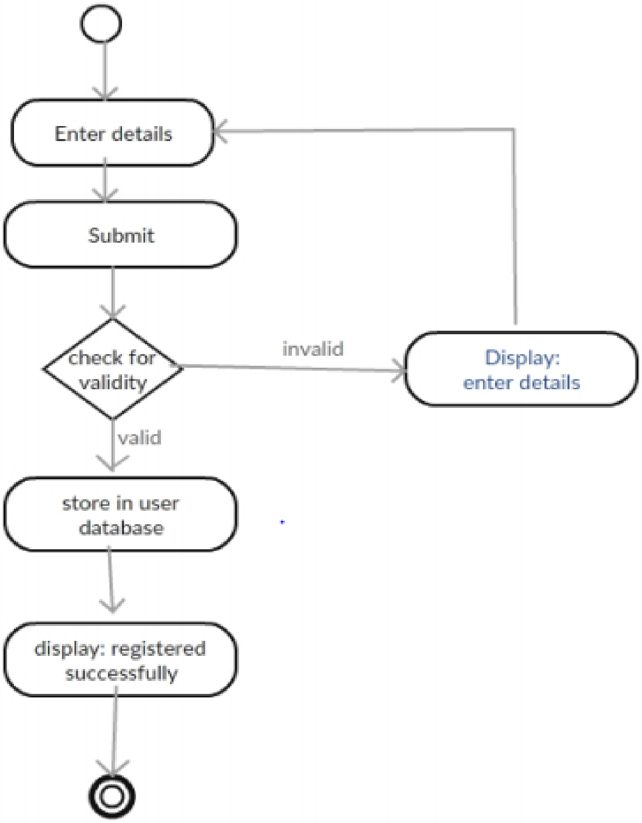
Usecase Diagram:



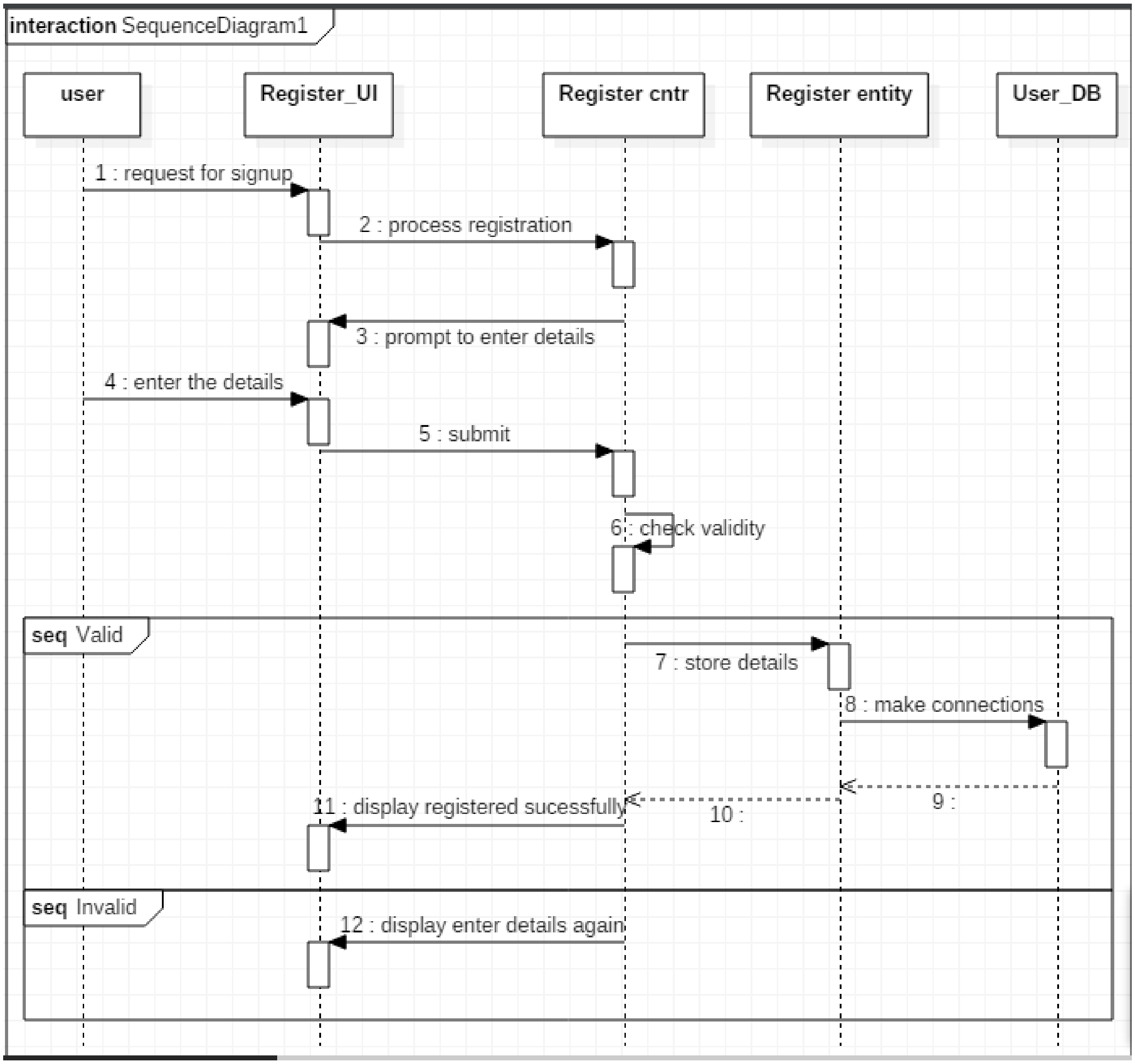
3.3 Registration (Sign Up)

Use Case ID:	1
Use Case Name:	Registration (Sign Up)
Actors:	User
Description:	The required fields to be filled by user are set and all the entered details by the user are stored in the user database.
Pre conditions:	User hasn’t registered before i.e new user
Post conditions:	Displays “registered successfully” message and login page is opened.
Normal Flow:	1. The valid user details have to be entered 2. The prescribed details are stored in the user database 3.A unique username is generated along with password.
Exceptions:	If invalid details are entered then system prompts the user to re enter the details. If the user is not new then system displays “username is already registered”

Activity Diagram:



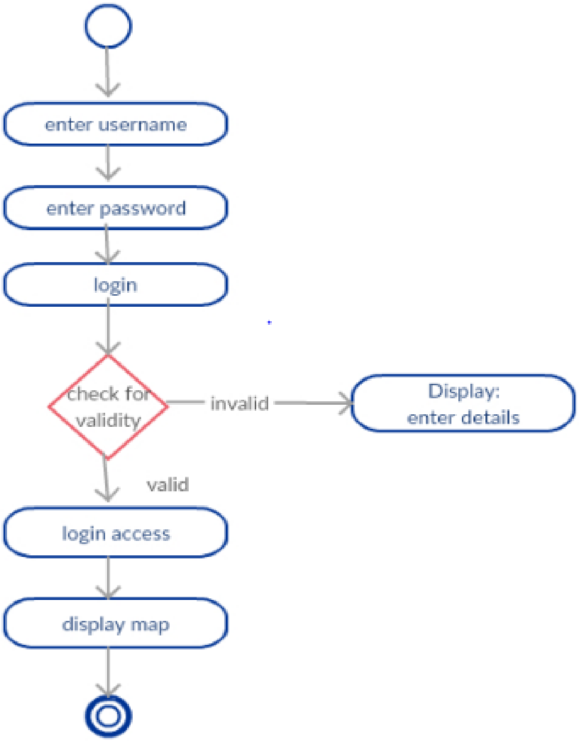
Sequence Diagram:



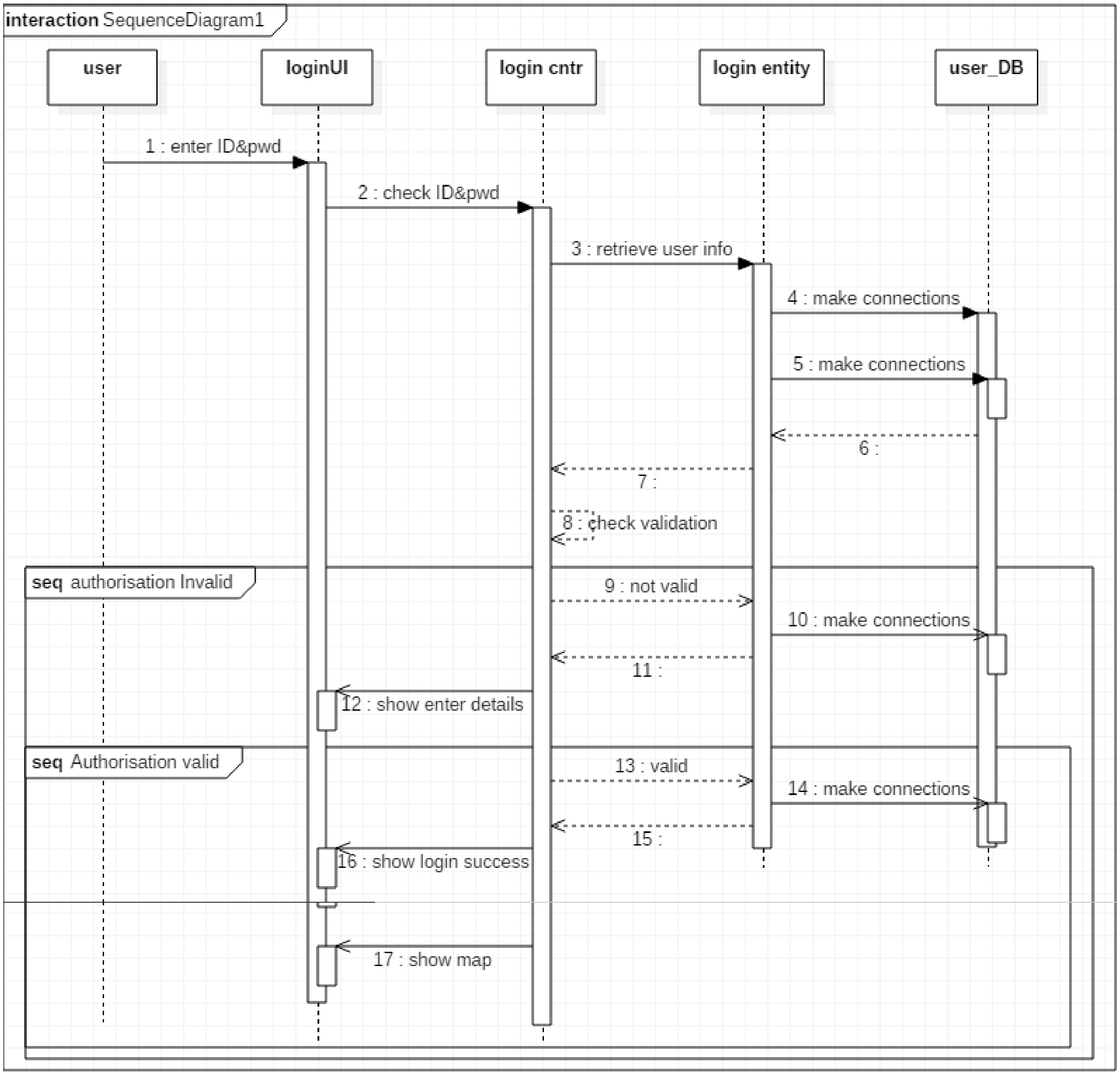
3.4 Login

Use Case ID:	1
Use Case Name:	Login
Actors:	User
Description:	User can access the services of application through Username and password and can view his account.
Pre conditions:	User has to register before
Post conditions:	map is displayed.
Normal Flow:	1. The valid username and password details have to be entered 2. user account is displayed
Exceptions:	1.If invalid details are entered then system prompts the user to re enter the details.

Activity diagram:



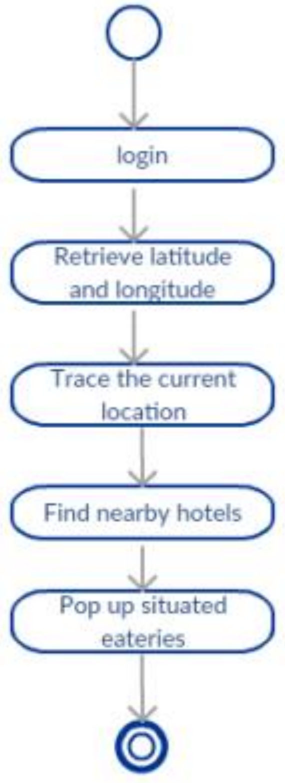
Sequence diagram:



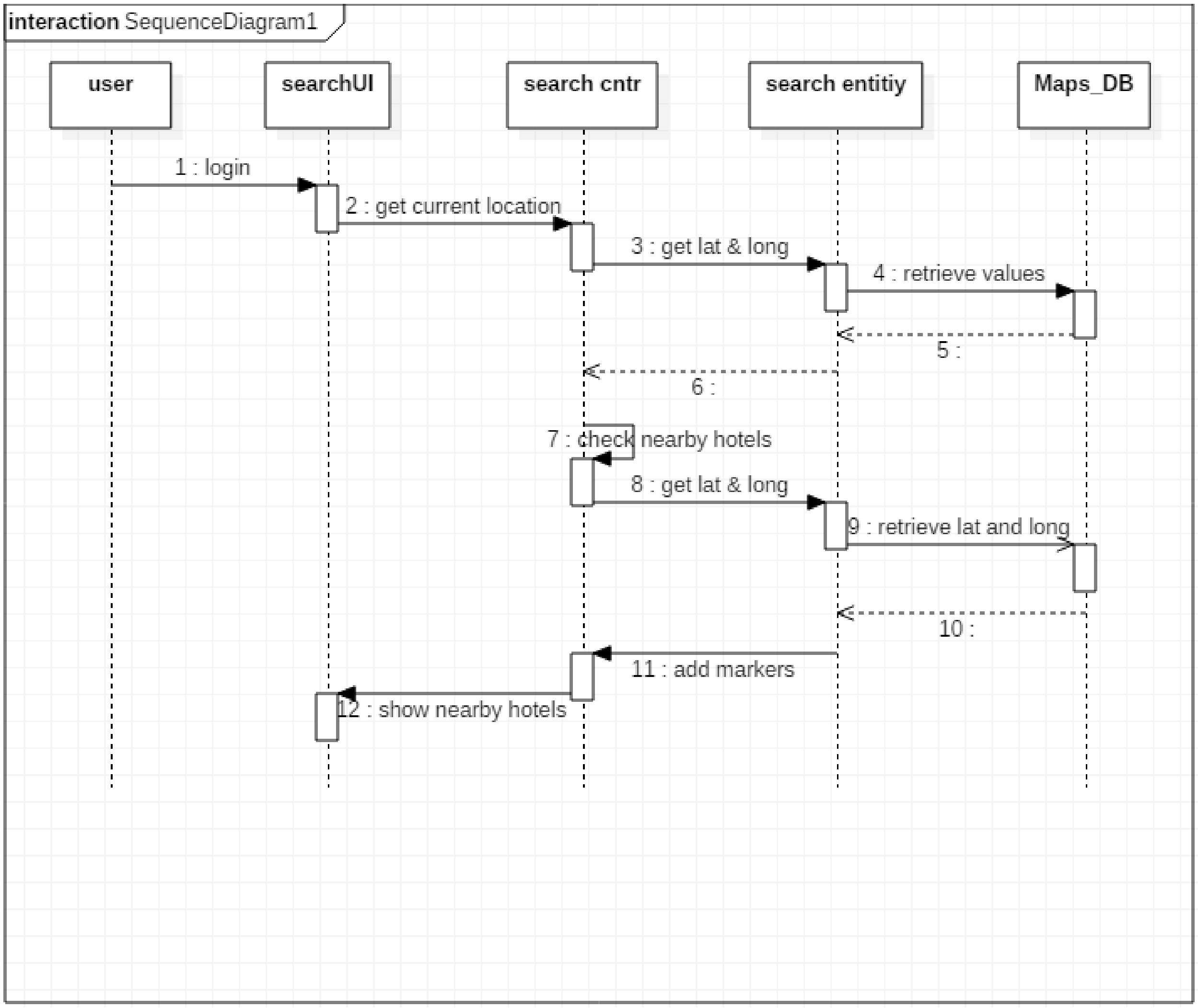
3.3 Findhotels

Use Case ID:	3
Use Case Name:	Findhotels
Actors:	User
Description:	User can discover any kind of eateries like hotels, coffee shops, canteens, dhabas and other local hotels which are situated around current location
Pre conditions:	User needs to login before exploring any food outlets
Post conditions:	Restaurants gets popped up around the present location
Normal Flow:	1. when user login into his account, the system trace the location of user by retrieving the corresponding latitude and longitude from Google maps 2. Through which, current location gets traced and stationed eateries are popped up.

Activity diagram:



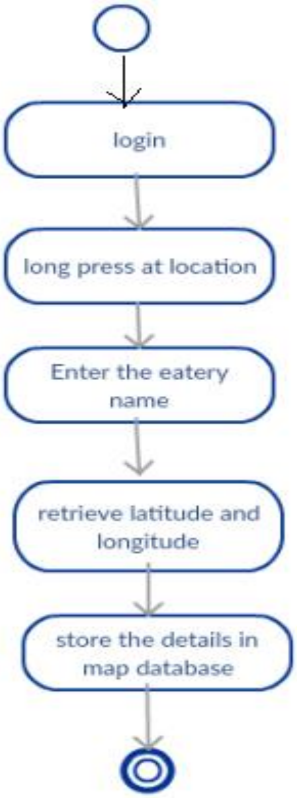
Sequence diagram:



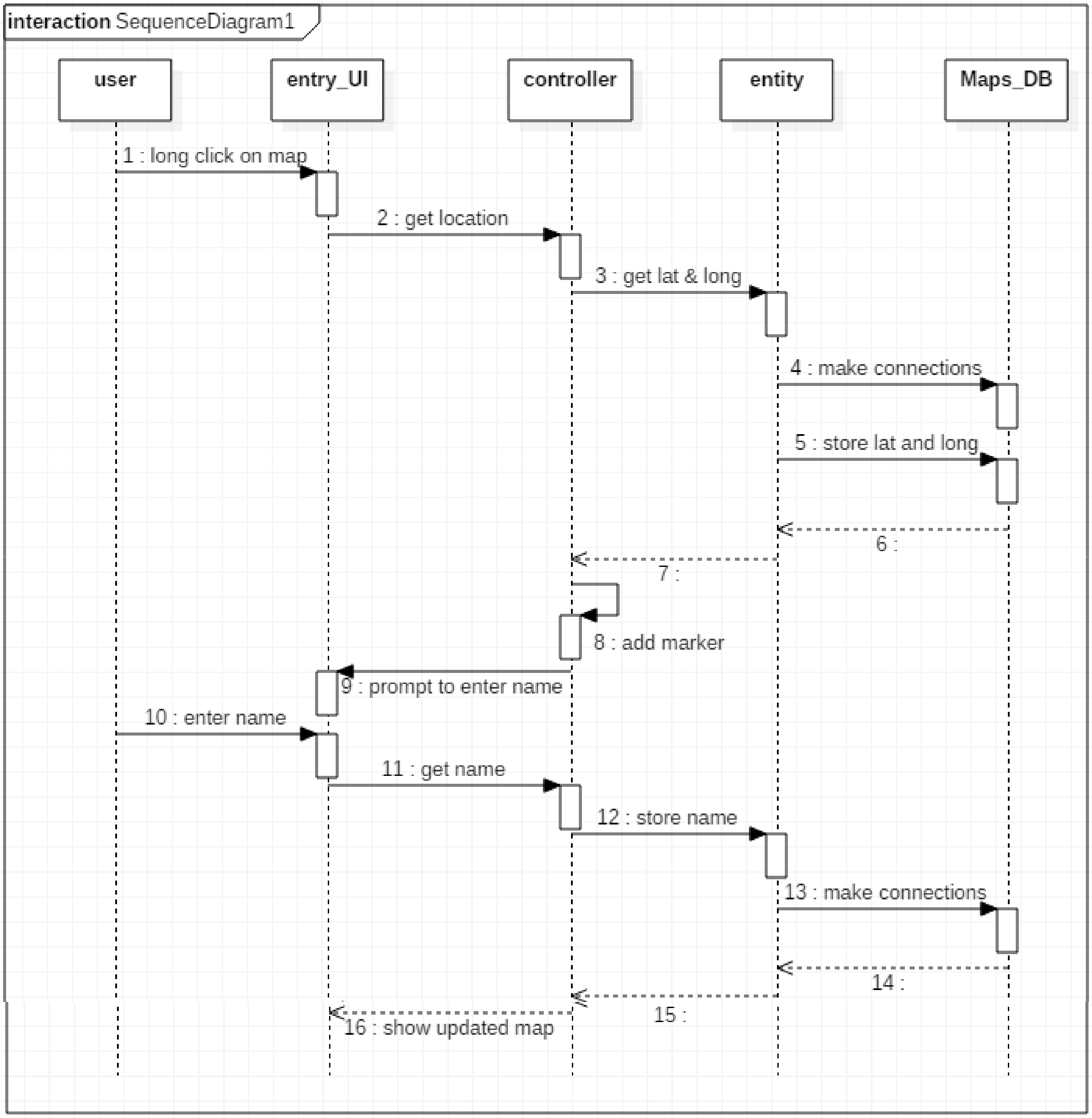
4.4: Eaterie Entry

Use Case ID:	4
Use Case Name:	Eaterie Entry
Actors:	User
Description:	User can long press the location over the map which provides a book mark for the specified eatery.
Pre conditions:	The restaurant which is to be marked must be present at the specified location.
Post conditions:	The details of the corresponding eatery details are stored in the maps database.
Normal Flow:	1.User can enable a bookmark for the specified eatery through long press on the location over the map. 2. The latitude and longitude of the location gets tracked from Google maps and are stored in the maps database. 3.Later on, when user search for same location the marked entry gets popped up along with existing ones.

Activity Diagram:


login
long press at location
Enter the eatery name
retrieve latitude and longitude
store the details in map database

Sequence Diagram:



4.TESTING AND RESULTS

The following screen appears as soon as the user opens the application as shown in figure 4.0.

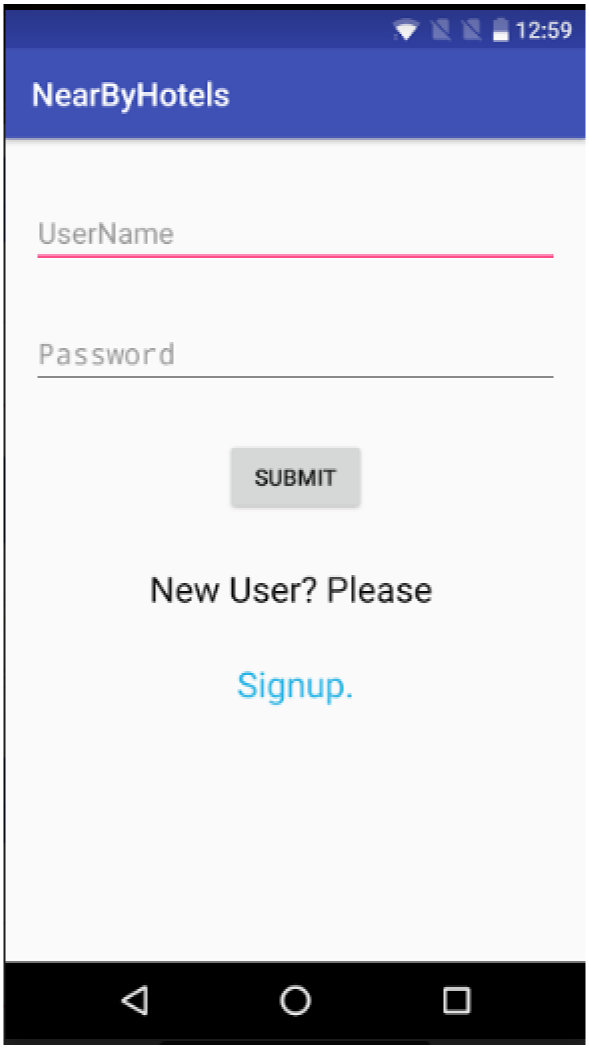
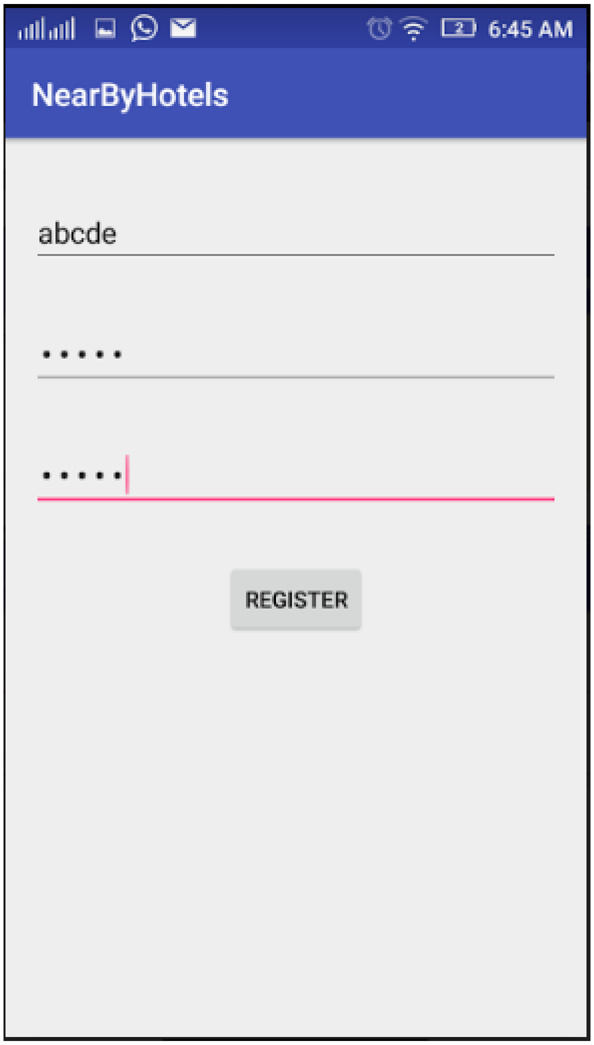


Figure 4.0

4.1.Register

If the user hasn't registered before , on selecting "Sign up" the following screen appears as shown in figure 4.1.1



6:45 AM

NearByHotels

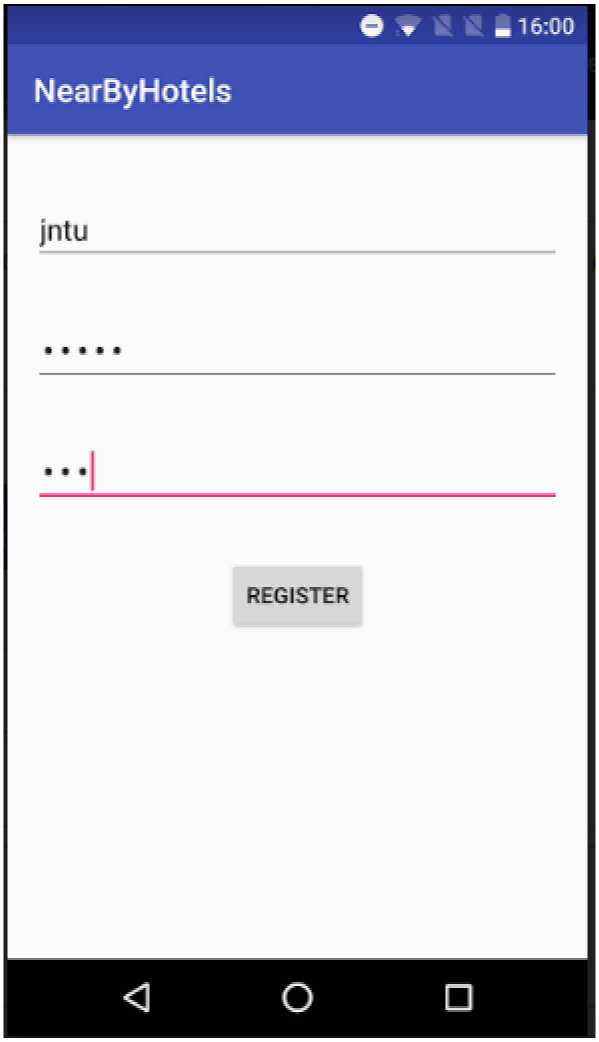
abcde

•••••

•••••

Figure 4.1.1

If the invalid details like different passwords are entered then the following screen appears as shown in figure 4.1.2

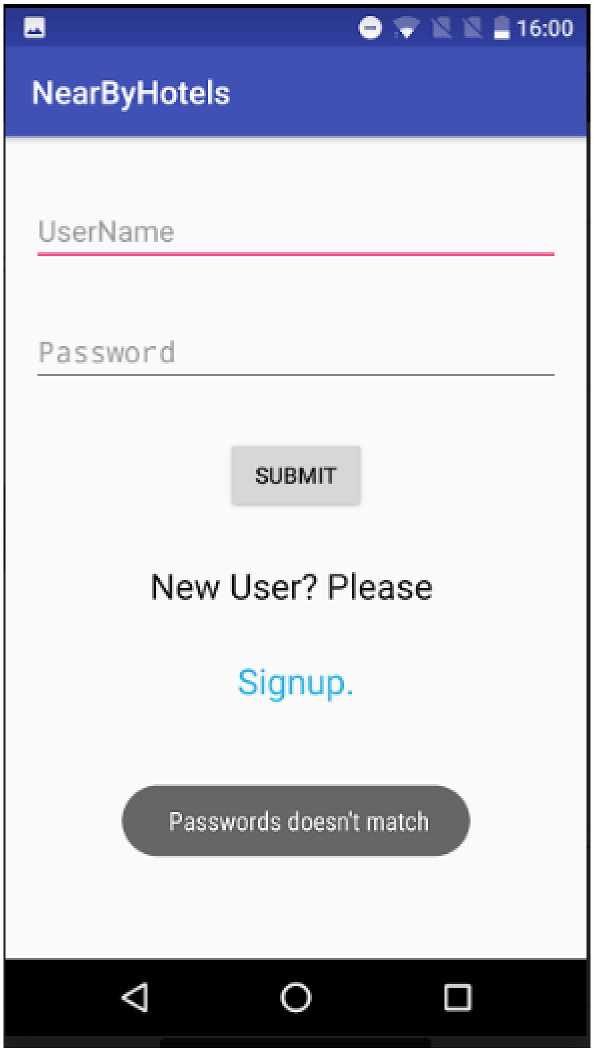

NearByHotels

jntu

•••••

•••

Figure 4.1.2.1


NearByHotels

UserName

Password

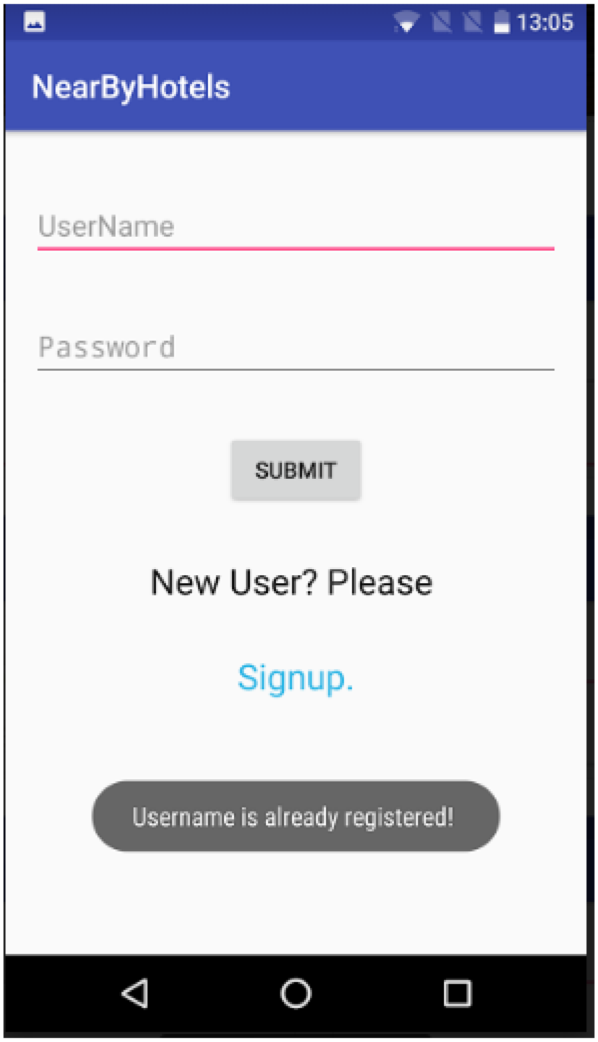
New User? Please

Signup.

Passwords doesn't match

figure 4.1.2.2

If the user has already registered, who is not a new user then following screen appears as shown in figure 4.1.3



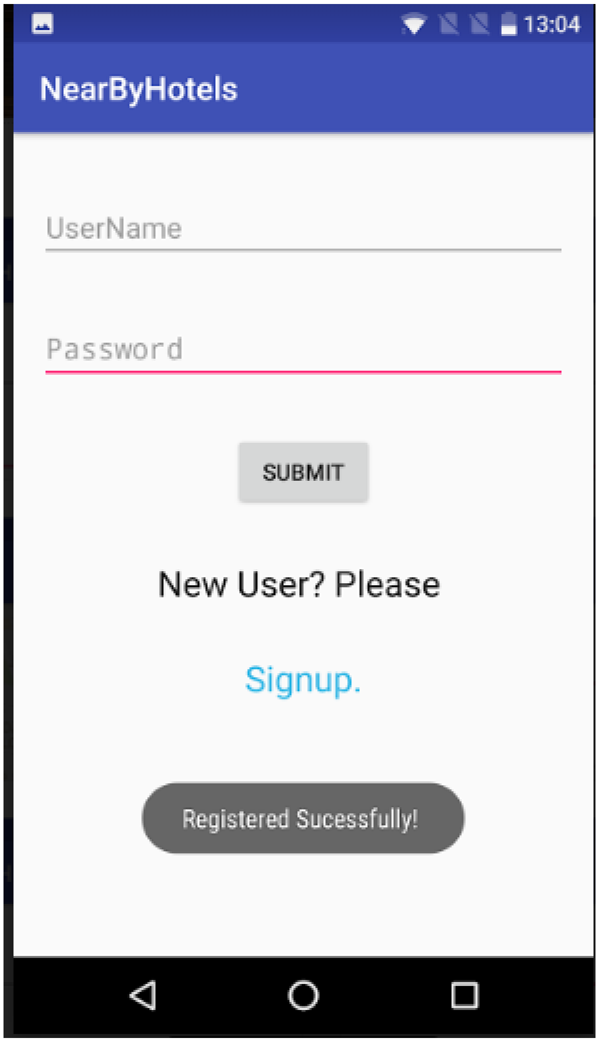
New User? Please

Signup.

Username is already registered!

Figure 4.1.2

After filling up details in the registration form, the following screen appears as shown in figure 4.1.4



New User? Please

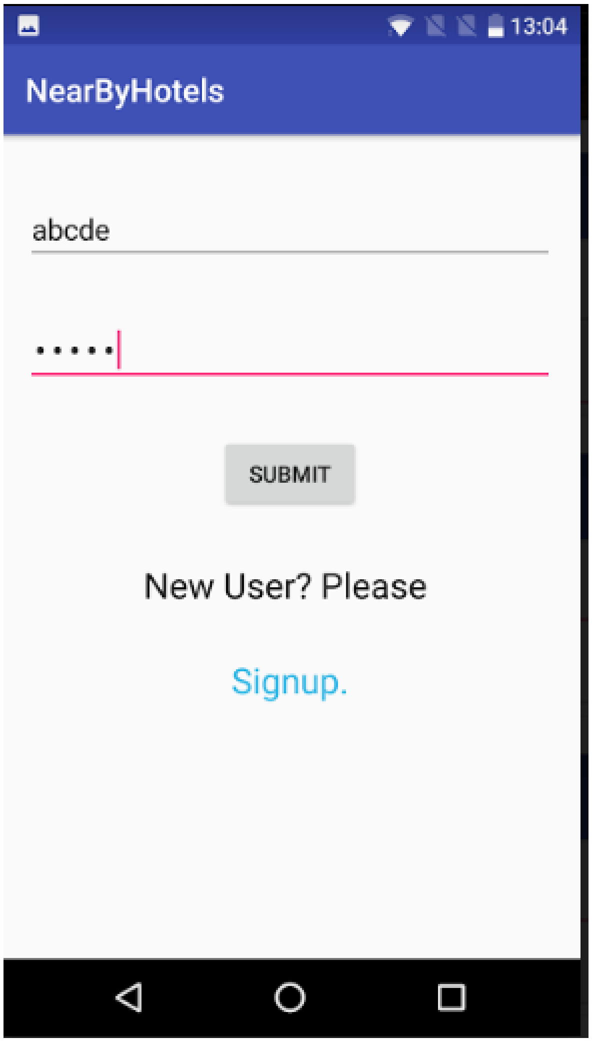
Signup.

Registered Sucessfully!

Figure 4.1.4

4.2 Login

After registration, the login screen appears as shown in figure 4.2.1



NearByHotels

abcde

•••••

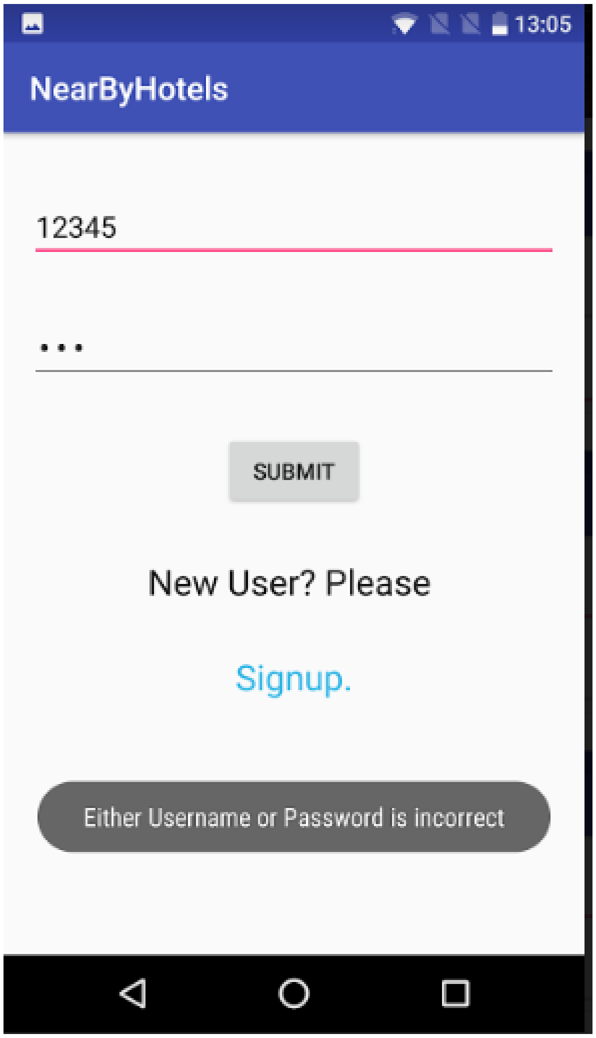
New User? Please

Signup.

Figure 4.2.1

If the user enters invalid details , then the following screen appears as shown

in figure 4.2.2



13:05

NearByHotels

12345

•••

SUBMIT

New User? Please

Signup.

Either Username or Password is incorrect

Figure 4.2.2

4.3 FindHotels

When user login into his account, the following screen showing map appears as which represents current location as shown in figure 4.3.1

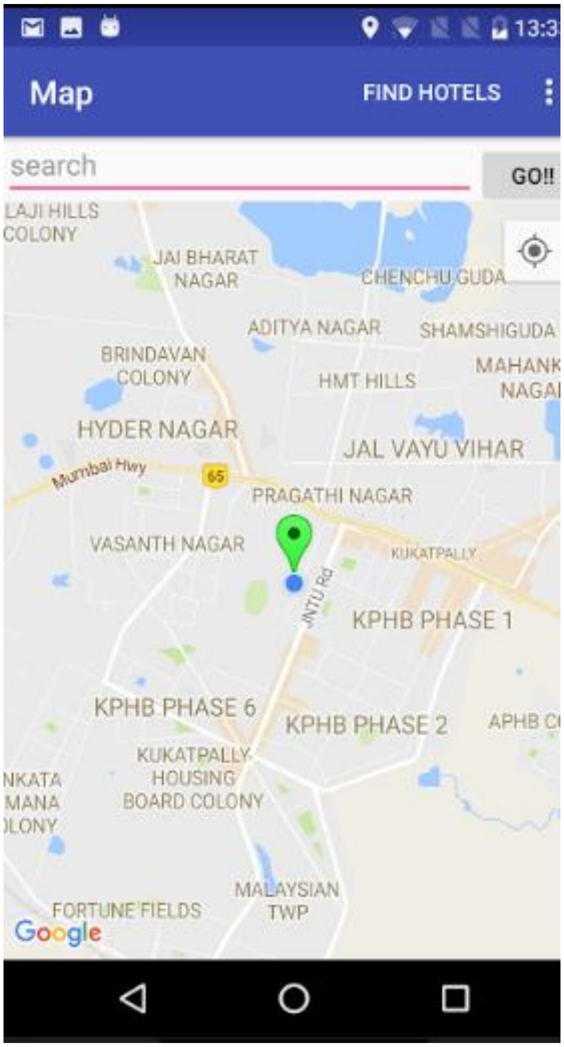


figure 4.3.1

when user press the “find Hotels” option , the screen appears as shown in figure 4.3.2.where near by eateries gets popped up.

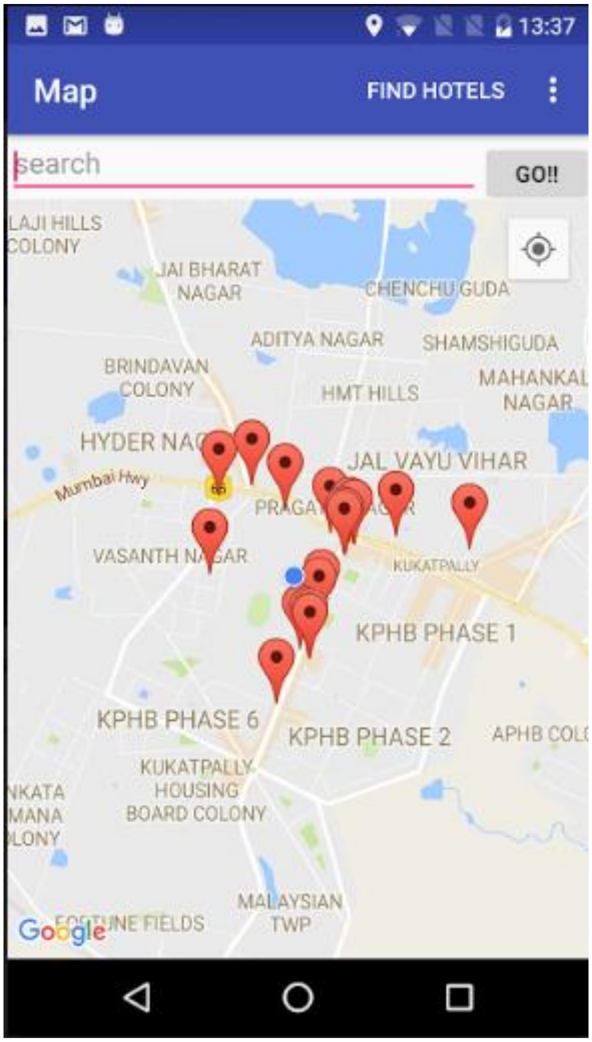
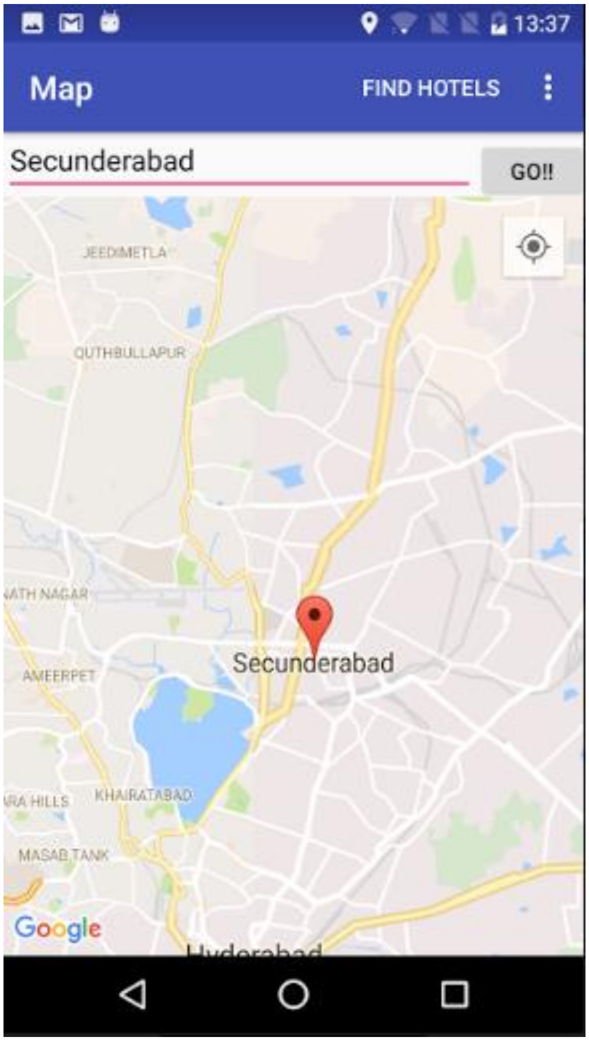


Figure 4.3.2

Whenever user search for specified location , the following screen appears as shown in figure 4.3.3

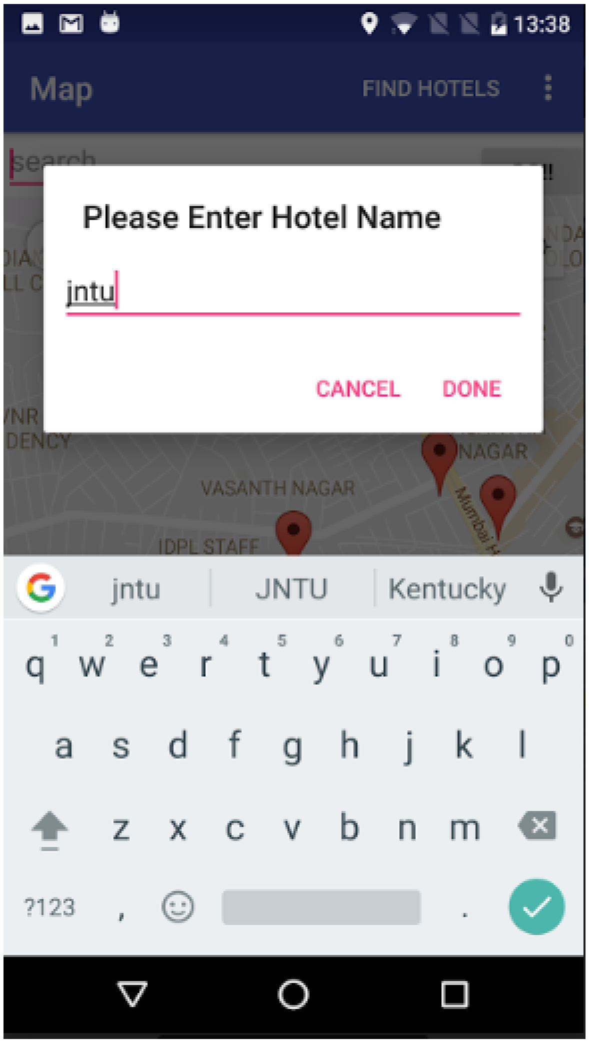


Secunderabad

Figure 4.3.3

4.4 Eaterie Entry

When user does the long press on location over the map, the following screen appears as shown in figure 4.4.1.



Please Enter Hotel Name

jntu

Figure 4.4.1

When user enter the details of the restaurant , this enables a bookmark at appropriate location and the following screen appears as shown in figure 4.4.2

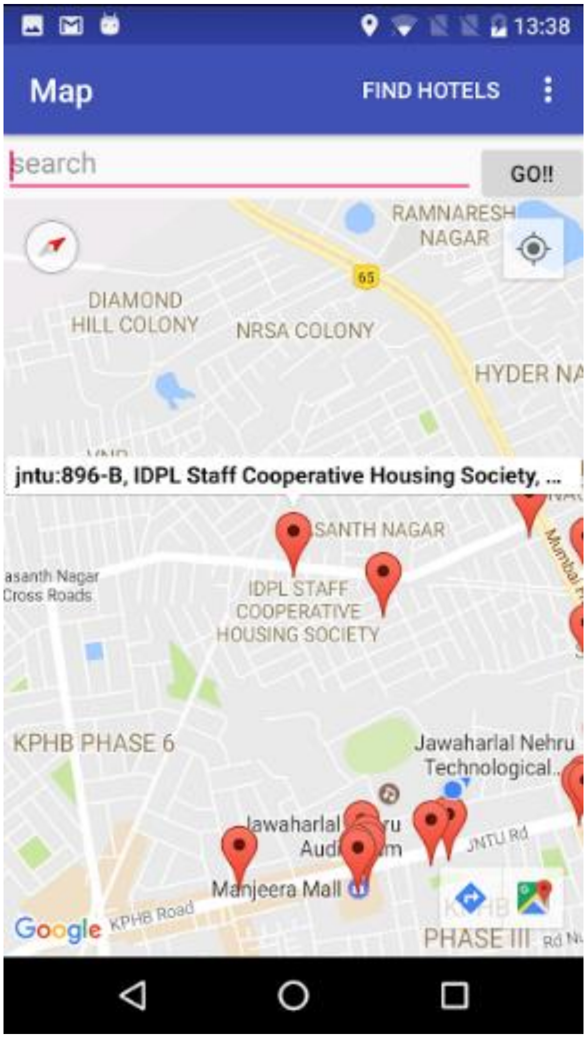


Figure 4.4.2

User can view the directions by pressing the icon that is encircled in the following screen as shown in figure 4.4.3 and the resultant screen appears as shown in figure 4.4.4

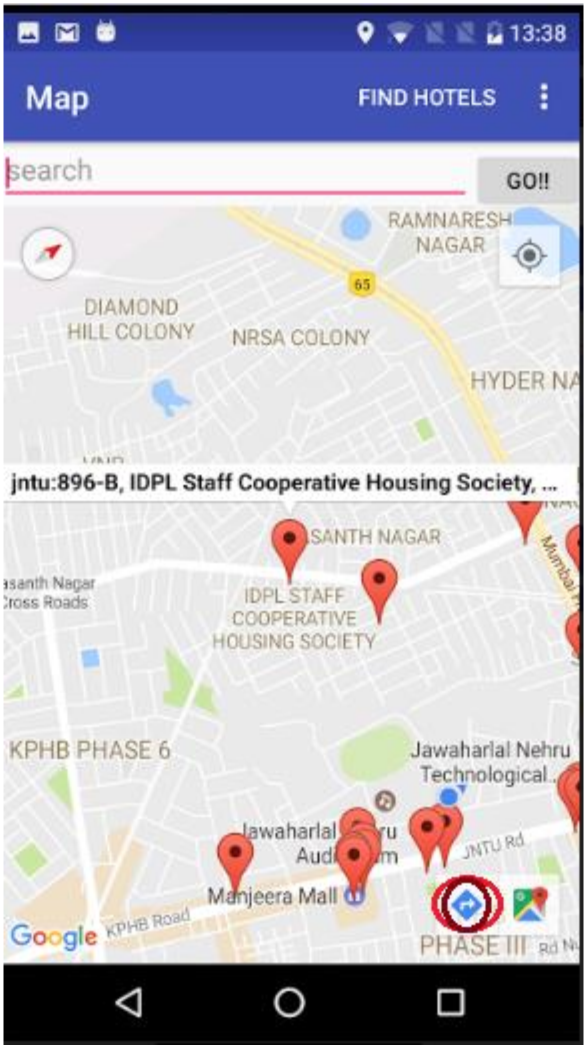


Figure 4.4.3

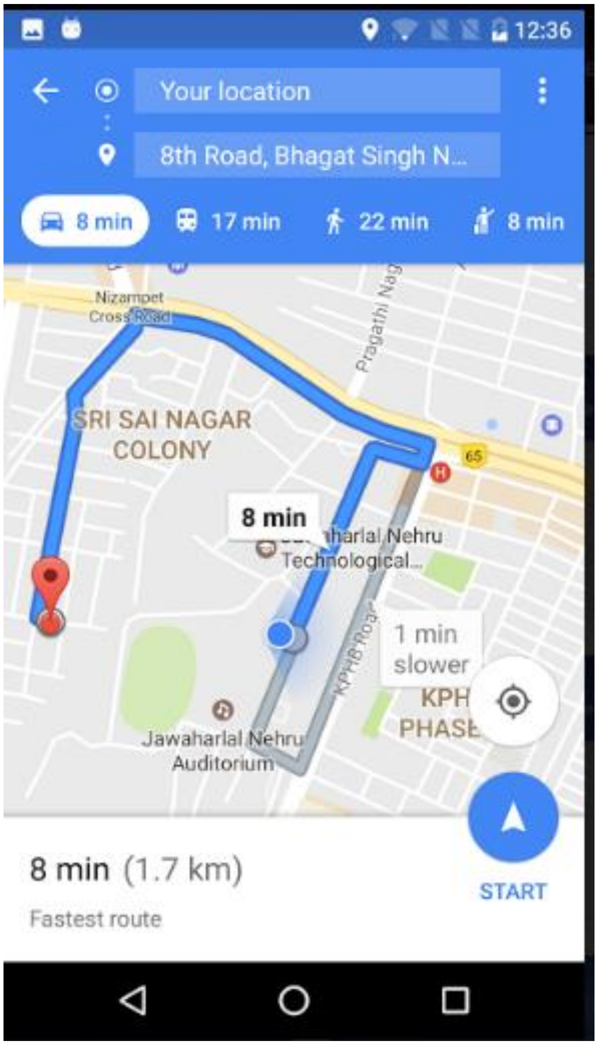


figure 4.4.4

4.5 Logout

When user logs out from the application , the screen appears as shown in figure 4.6.

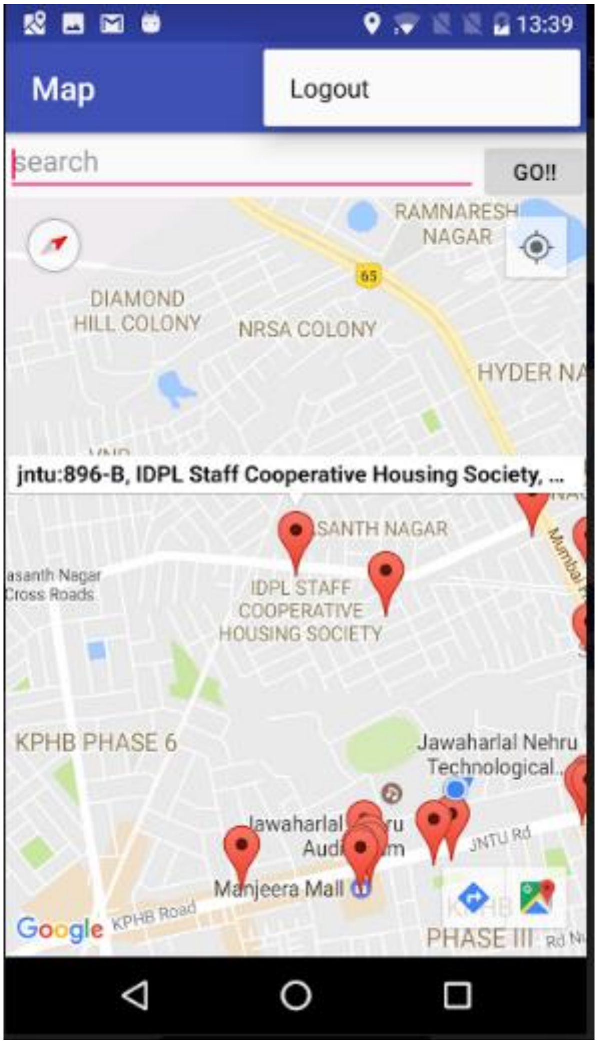


Figure 4.6

5. DATABASE

5.1 USER DATABASE

5.1.1 LOGIN AND REGISTRATION

SNO	Username	Password
1	Lakshmi kolluru	13011A0518
2	Revanth	13011A0569
3	Sahana	13011A0537
4	Keshav	Shailesh

5.2 MAPS DATABASE

5.2.2 FIND HOTELS AND EATERIE ENTRY

SNO	Place	Vicinity	Latitude	Longitude
1	Punjabi Dhabha	NH65, Ashok Nagar, , Kukatpally Housing board Colony,	17.498731,	78.389935
2	Ragavendhra Mess	Sardar Patel Nagar, IDPL Staff Cooperative Housing Society, Kukatpally Housing Board Colony,	17.498557,	78.391593
3	Balayya chai bandi	SY-132, Mumbai Highway, Hyder Nagar, Kukatpally, Jai Bharat Nagar, Hyder Nagar,	17.498958,	78.382813
4	Sri Sai ram Chinese food	Necklace Rd, Central Secretariat, Khairatabad,	17.38723	78.5234
5	Devi chat bhandar	Peddhammapuram, nalgonda	17.4352	79.6567

6.CONCLUSION

User can have facility to login through social networking sites. Apart from finding the eateries nearby current location we can also extend the feature to explore the foodie bays at specified locations. Administrator need to have privilege to verify and store the details of the corresponding entry into the database marked by the user.

7. REFERENCES

[1] Motivation about the Restaurant finder application

https://www.zomato.com/india

http://pokemongo.com

[2] **Android Installation**

http://developer.android.com/sdk/installing/index.html

[3] Android Tutorials

http://developer.android.com/tools/devices/managingavds.html

http://www.tutorialspoint.com/android/android_application_components.html

https://developers.google.com/maps/documentation/android-api/start

https://androidhive.com