INDEX

1. ABSTRACT

Modern hand held devices such as smart phones and PDAs have become increasingly powerful in recent years. Dramatic breakthroughs in processing power along with the number of extra features included in these devices have opened the doors to a wide range of commercial possibilities. In particular, most cell phones regularly include cameras, processors comparable to PCs from only a few years ago, and internet access. However, even with all these added abilities, there are few applications that allow much passing of the environmental information and location based services.

As mobile devices become more like PCs they will come to replace objects we tend to carry around such as checkbooks, credit cards, cameras, planners, mp3 players, etc. In short, we will be using them to accomplish our daily tasks. One application that falls into this category is the Places Around Me Application developed for the Google Android Phones.

The prime objective of “Places Around Me Application” is to create a fully fledged Android application which could locate the location of the user and places in his/her vicinity. The application helps the user to find hospitals, restaurants, shopping malls or any other required facility within the specified range by providing the radius feature which allows the user to specify the circle of distance around which the user can view all the places around along with the distance from the current location. The user can also map the location of the destination place on Google Maps rendered to the user on the phone & find the path from his current location to that location. Not only does the

application help the user in finding places around him/her but also allows the user to share the location to others via social networking sites or electronic message or mail.

The Project is developed in Java Programming Language by using the Eclipse Ganymede Integrated Development Environment (IDE). We use the Android Software Development Kit (SDK) which includes a variety of custom tools that help us develop mobile applications on the Android platform. The most important of these are the Android Emulator and the Android Development Tools (ADT) plug-in for Eclipse.

2. LIST OF ABBREVIATIONS

GPS	Global Positioning System
API	Application Programming Interface
HTC	High Tech Corporation
AOSP	Android Open Source Project
SDK	Software Development Kit
ADT	Android Development Toolkit
AVD	Android Virtual Device
NLP	Network Location Provider
UI	User Interface
XML	Extensible Markup Language
SOAP	Simple Object Access Protocol
JSON	JavaScript Object Notation

3. INTRODUCTION

3.1 Literature Review

Android is the emerging device in today's world. It has occupied large percentage in the mobile market. It functions not only to make calls but also makes the life of the user easy with its various feature. The Places Around Me Application is developed with keeping in view the present requirements of an android phone in the competitive world. The project uses the GPS tracking system to locate the position of the user. The application uses the Google Maps to navigate from one place to another. This application is not only meant to find places in the vicinity but also connect with other people using electronic media

3.2 Motivation

The main motivation for the project was to develop an application which would make life easy. The Android Platform was preferred because it is one of the fastest growing mobile operating systems on the market and is an open source development. This project allowed to gain an understanding of how some of the built in frameworks can be utilized to develop application. Furthermore, this project demonstrates how mobile applications can contribute to improve in lifestyle of user with all the information of the world on a hand with a single click

3.3 Birth Of Android

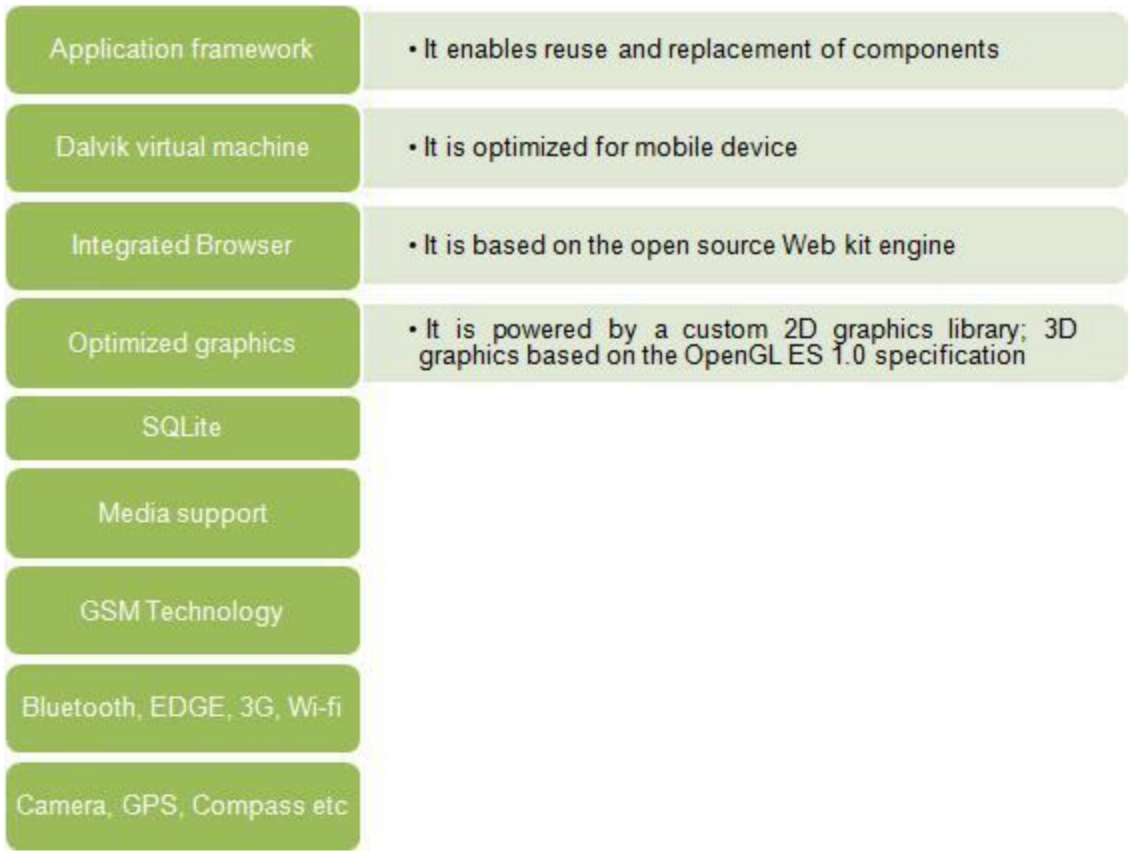
An Android Device

Android is a mobile phone operating system. It was originally developed by Android Inc, which was acquired by Google in July 2005. Today, development is overseen by the Android Open Source Project (AOSP), led by Google. The AOSP is "tasked with the maintenance and further development of Android". As of the 3rd Quarter of 2010 Android has a market share of 25% making it the second most popular phone operating system of the market (second only to Nokia's Symbian). This is a major rise from the 3.5% share Android had in 3Q2009.

Android, Inc. was founded in Palo Alto, California, United States in October 2003 by Andy Rubin (co-founder of Danger), Rich Miner (co-founder of Wildfire Communications, Inc.), Nick Sears (once VP at T-Mobile),and Chris White (headed design and interface development at WebTV)to develop, in Rubin's words "...smarter mobile devices that are more aware of its owner's location and preferences. Despite the obvious past accomplishments of the founders and early employees, Android Inc. operated secretly, revealing only that it was working on software for mobile phones. That same year, Rubin ran out of money. Steve Perlman, a close friend of Rubin, brought him $10,000 in cash in an envelope and refused a stake in the company.

Google acquired Android Inc. on August 17, 2005, making Android Inc. a wholly owned subsidiary of Google. Key employees of Android Inc., including Andy Rubin, Rich Miner and Chris White, stayed at the company after the acquisition .

3.4. Features



Application framework	• It enables reuse and replacement of components
Dalvik virtual machine	• It is optimized for mobile device
Integrated Browser	• It is based on the open source Web kit engine
Optimized graphics	• It is powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification
SQLite	
Media support	
GSM Technology	
Bluetooth, EDGE, 3G, Wi-fi	
Camera, GPS, Compass etc	

Android Operating System Feature

World is contracting with the growth of mobile phone technology. As the number of users is increasing day by day, facilities are also increasing. Starting with simple regular handsets which were used just for making phone calls, mobiles have changed our lives and have become part of it. Now they are not used just for making calls but they have innumerable uses and can be used as a Camera , Music player, Tablet PC, T.V. , Web browser etc . And with the new technologies, new software and operating systems are required.

In the present world market, Android is very popular among the user. Today the mobiles are not only thought to be used in the just for communication but also as a mini computer which has the computing and manipulating capacity as that of computers. The wide use and popularity of the android is due to the various features available in it.

3.4.1 Application Framework

It is used to write application for Android .Unlike other embedded mobile environment ,Android application are all equal, for instance ,an application which come with the phone are no different than those that any developer writes. The framework is supported by numerous open source libraries such as openssl, SQLite and libc. It is also supported by the android core libraries. From the point of security, the application have only those abilities that owner of the mobile gives them at the installation time.

3.4.2 Dalvik Virtual Machine

It is extremely low memory based virtual machine, which was designed especially for Android to run on embedded systems and work well in low power situations. It is also tuned to CPU attributes .The DVM creates a special file format (.DEX) that is created through build time post processing. Conversion between Java classes and .DEX is done by included “dx” tool

3.4.3 Integrated Browser

Google made a choice on WebKit as open source web browser. They added a two pass layout and frame flattening . Two pass layout loads a page without waiting for blocking elements such as external CSS or JavaScript.

3.4.4 Optimized Graphics

An Android has 2D graphics library and 3D graphics based on OpenGL ES 1.0.

3.4.5 SQLite

SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.

SQLite is a popular choice as embedded database for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems, among others. SQLite has many bindings to programming languages.

3.4.5 Java Virtual Machine

Java virtual machine (JVM) is a virtual machine that can execute Java byte code. It is the code execution component of the Java software platform. Sun Microsystems has stated that there are over 5.5 billion JVM-enabled devices. JVMs are most often implemented to run on an existing operating system, but can also be implemented to run directly on hardware.

3.4.6 Development Environment

The Development Environment includes a device emulator , tools for debugging ,memory and performance profiling, a plug-in for the eclipse IDE. There are a number of hardware dependent features, for instance , a huge media and connections support, GPS, GSM telephony. A great work was done for the developers to start work with Android using device emulator ,tools for debugging and plug-in for eclipse.

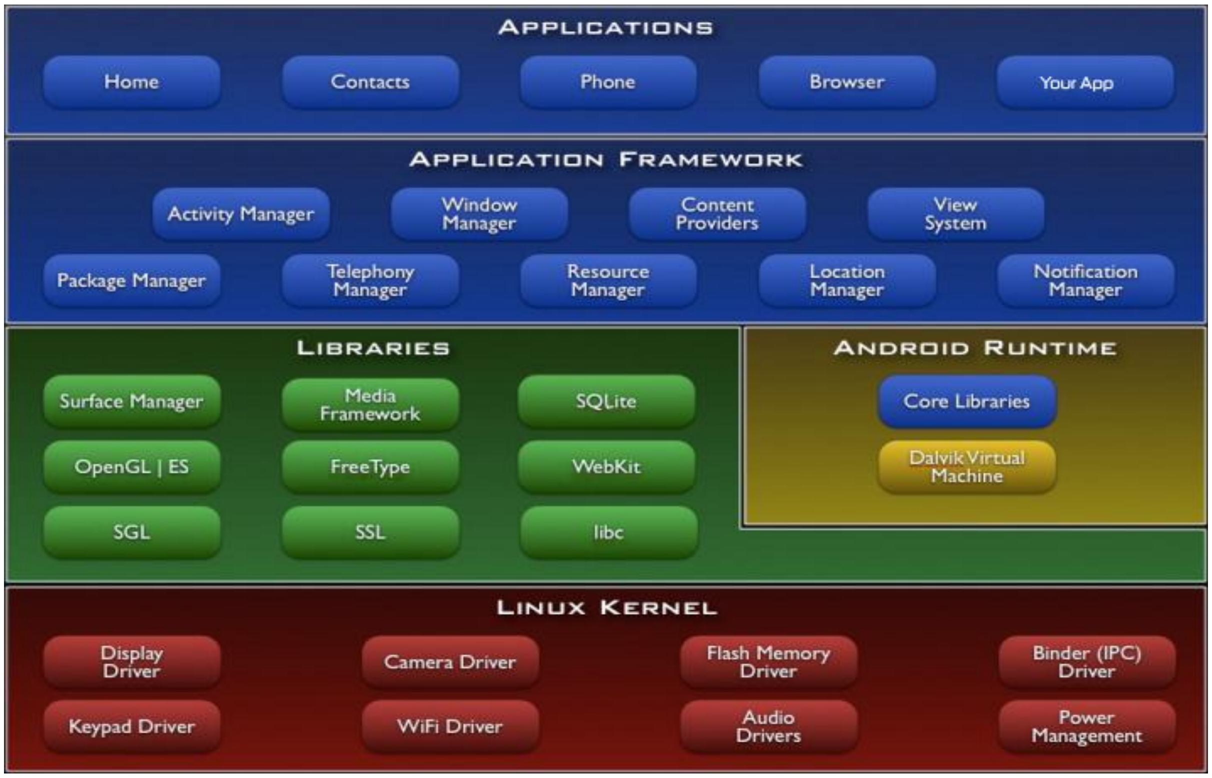
The different version of the android operating system that are launched are as follows

Version	Code name	Release date	API level	Distribution
5.1.x	Lollipop	March 9, 2015	22	2.6%
5.0.0–5.0.2		November 3, 2014	21	15.5%
4.4.0–4.4.4	KitKat	October 31, 2013	19	39.3%
4.3.x	Jelly Bean	July 24, 2013	18	4.7%
4.2.x		November 13, 2012	17	15.9%
4.1.x		July 9, 2012	16	13%
4.0.3–4.0.4	Ice Cream Sandwich	December 16, 2011	15	4.1%
2.3.3–2.3.7	Gingerbread	February 9, 2011	10	4.6%
2.2	Froyo	May 20, 2010	8	0.3%

Different Versions of Android OS

3.5. Architecture

Android is based on the Linux Kernel. Android Developers are able to access all the components of the Application Framework used by core applications when creating an application.



Android OS architecture

The above figure shows the diagram of Android Architecture. The Android OS can be referred to as a software stack of different layers, where each layer is a group of several program components. Together it includes operating system, middleware and important applications. Each layer in the architecture provides different services to the layer just above it.

3.6. Objectives

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.

As an Internship Project, the application developed is a "Places Around Me "This application includes the features of Google API to mark the present location of the individual and displays the list of places nearby. This application even allows the user to share the location or navigate from one place to another using Google Maps.

The application includes

- Geo-code specific Google maps and other related features design
- Optimum path finding and route highlighting with other related features design

The basic objectives of this projects are:

- To understand and realize the real world project , understand the problems associated in developing it and explore the appropriate solution to it.

- To understand the application development in android based system.
- To be familiar with the Google maps API and understand the features available on it.
- To think in a way so as to make the life simpler and easier with the available technology and resources.

The specific objectives are:

- To facilitate the user of android to get the location of required facilities through this app.
- To view all places around the user
- To let the user choose the circle of radius within which all the places are displayed.
- To navigate from one place to another using Google Maps
- To share the location

4. REQUIREMENTS SPECIFICATION

Requirement analysis is the first thing to be done in starting any kind of project. Requirement analysis is important and preliminary factor of software development process. It helps to identify the feasibility of project and many more. Hence in order to make our project real and to realize the project in a practical way, we first analyze the requirement of our project. Specifically the requirements of our project are tabulated as:

- Identify the current location of the user
- List out places on the basis of distance
- Share the location

4.1. Hardware Requirements

- A Mobile Phone with Android Operating System and GPS facility and access to Google Maps

4.2. Software Requirements

- Eclipse Operating System
- Android Software Development Kit
- JAVA programming language
- JAVA Development Kit
- Android Development Tools(ADT)
- Google Play Services Packages

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plug-in, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and Net Beans IDE also supports Android development via a plug-in. Also SDK provides several interface and drag and drop features in it. We have used several features available in it as per our need and the requirement.

4.3 Background Theories

4.3.1 Maps, Geocoding and Location Based Services

One of the defining features of mobile phones is their portability, so it's not surprising that some of the most enticing Android features are the services that let us find, contextualize, and map physical locations.

We can create map-based Activities using Google Maps as a User Interface element. We have full access to the map, allowing us to control display settings, alter the zoom level, and move the centered location. In order to display the map in the application ,we can use API provided. There are several API,s related . However Google Maps API is very popular and are extensively used. Google API has the several features available in it which we can use it through simple effort and in an easier way. For this purpose of using Google map API in our application we need to generate the MD5 hash key through the command line and send it to the Google Inc. which in turn provides the API key. This API key is used in the application to display the map.

Google maps has its own symbols and patterns . We may come across the situation that we need to add some markers and symbols on the map. In that situation Google API provides a tool named Overlay. Using Overlays, within the project ,we can annotate maps and handle user input to provide map contextualized information and functionality. We can use any number of overlays on the map according to the need and specification. Overlays are a way to add annotations and click handling to Map Views. Each Overlay lets you draw 2D primitives including text, lines, images and shapes directly onto a canvas, which is then overlaid onto a Map View. We can add several Overlays onto a single map. All the Overlays assigned to a Map View are added as layers, with newer layers potentially obscuring older ones. User clicks are passed through the stack until they are either handled by an Overlay or registered as a click on the Map View itself.

Location-Based Services (LBS) — the services that let us find the device's current location. They include technologies like GPS and Google's cell-based location technology. We can specify which location-sensing technology to use explicitly by name, or implicitly by defining a set of criteria in terms of accuracy, cost, and other requirements. Google map API has the in built library through which we can find the device current location .Further it has the ability to listen the location change of the device. Whenever the location of the device is changed from a particular location, it provides the notification of location changed through the features available in it. Location-based services (LBS) is an umbrella term used to describe the different technologies used to find the device's current location. The two main LBS elements are:

- Location Manager Provides hooks to the location-based services.
- Location Providers Each of which represents a different location-finding technology used to determine the device's current location.

Using the Location Manager, we can:

- Obtain your current location.
- Track movement.
- Set proximity alerts for detecting movement into and out of a specifi ed area.

Maps and location-based services use latitude and longitude to pinpoint geographic locations, but users are more likely to think in terms of an address. Android provides a Geo coder that supports forward and reverse geo coding. Using the Geo coder, We can convert back and forth between latitude/longitude values and real-world addresses. Forward Geocoding is the conversion of address provided in to the latitude and longitude whereas Reverse Geocoding is the conversion latitude and longitude in to the address.

Used together, the mapping, geocoding, and location-based services provide a powerful toolkit for incorporating phone's native mobility into mobile applications.

4.3.2. JavaScript Object Notation

JSON (JavaScript Object Notation) is a text-based open standard designed for human readable independent data interchange. JSON is limited to text and numeric values. Binary values are not supported. JSON is a subset of the JavaScript Specification (ECME-Script) and it is therefore directly supported in JavaScript. Data structures in JSON are based on key / value pairs. The key is a string, the value can be a numerical value, a Boolean value (true or false) or an object. The JSON format is often used for serializing and transmitting structured data over a network connection. It is used primarily to transmit data between a server and web application, serving as an alternative to XML.

4.3.2.1 JSON Data Types and Syntax

JSON's basic types are:

- Number (double precision floating-point format in JavaScript, generally depends on implementation)
- String (double-quoted Unicode, with backslash escaping)
- Boolean (true or false)
- Array (an ordered sequence of values, comma-separated and enclosed in square brackets; the values do not need to be of the same type)

- Object (an unordered collection of key: value pairs with the ':' character separating the key and the value, comma-separated and enclosed in curly braces; the keys must be strings and should be distinct from each other)
- null (empty)

An JSON object is a set of key / value pairs which starts with "{" and ends with "}".

```
{
{
 firstName:'Lars',
 lastName:'Vogel',
 address: {
   street:'Examplestr.',
   number: '31'
 }
}
```

JSON arrays are one or more values surrounded by [] and separated by ",".

```
[
{
 firstName:'Lars',
 lastName:'Vogel',
 address: {
   street:'Examplestr.',
   number: '31'
 }
}
,
{
 firstName:'Jack',
 lastName:'Hack',
 address: {
   street:'Examplestr.',
   number: '31'
 }
}
]
```

4.3.2.2 Parsing through RESTful Services and JSON Responses

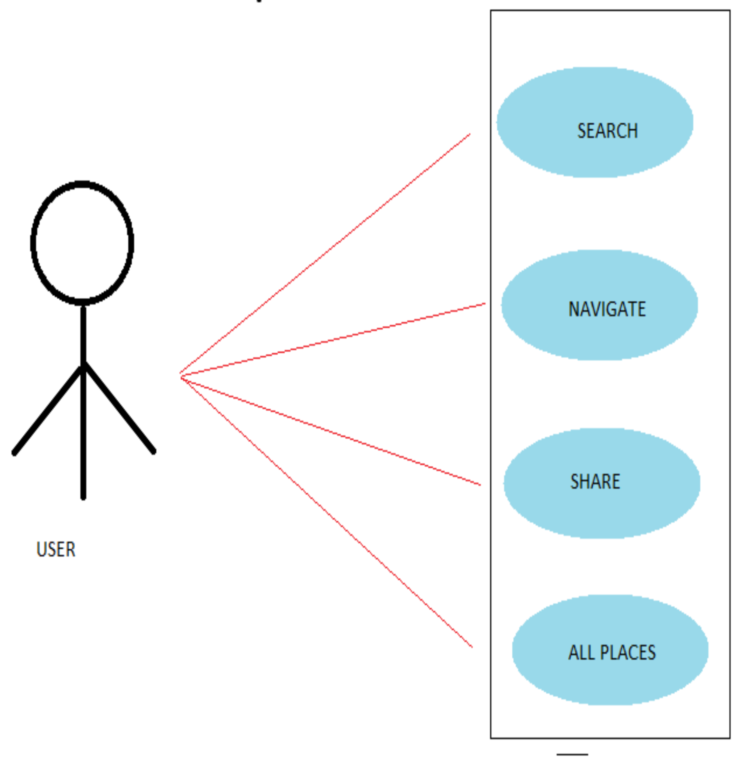
Roy Fielding's explanation of the meaning of REST was "Representational State Transfer " is intended to evoke an image of how a well-designed Web application behaves: a network of web pages (a virtual state-machine), where the user progresses through an application by selecting links (state transitions), resulting in the next page (representing the next state of the application) being transferred to the user and rendered for their use." It is for creating large scale of network system architecture style.

Motivation of REST was to create an architectural model how the Web service should work, such that it could serve as the guiding framework for the Web protocol's standards. REST has been applied to describe the desired Web architecture, helped to identify existing problems, to compare alternative solutions, and ensure that protocol extensions would not violate the core constraints that make the Web successful. But standards are usable even though REST is not a standard. For example:

- HTTP
- URL
- XML/JSON/HTML /etc
- Text/xml, object/json, text/html, etc

5. SYSTEM DESIGN AND IMPLEMENTATION

5.1 Use Case Diagrams



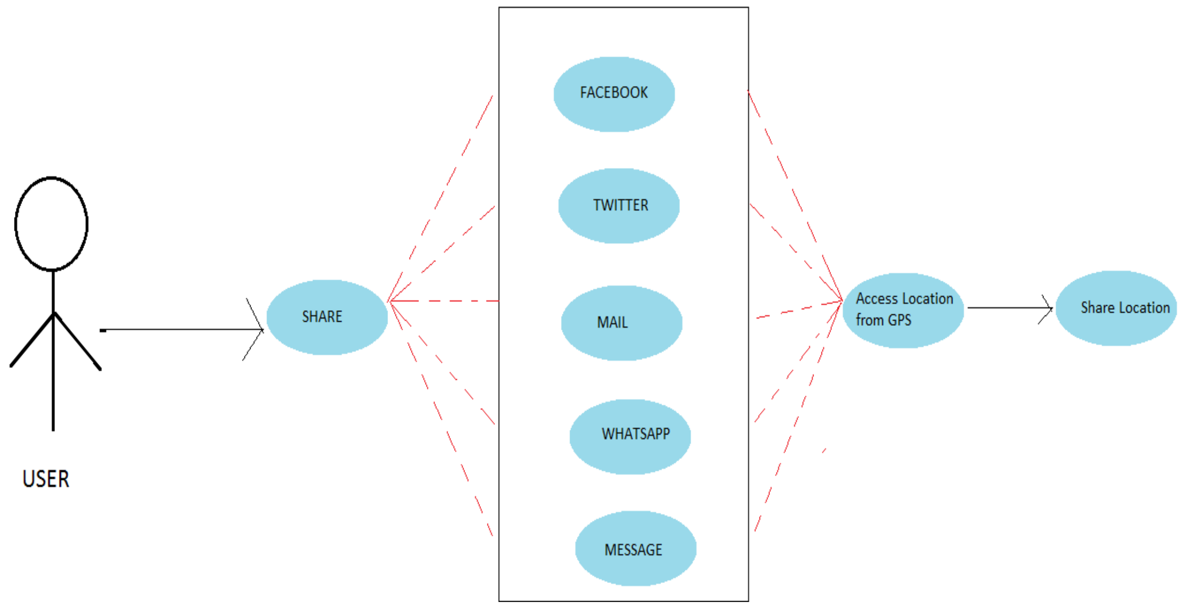
The various options a user has while accessing the application are shown above.

1. Search for any place

2. Navigate from one place to another

3. Share location with others

4. Know all the places around,



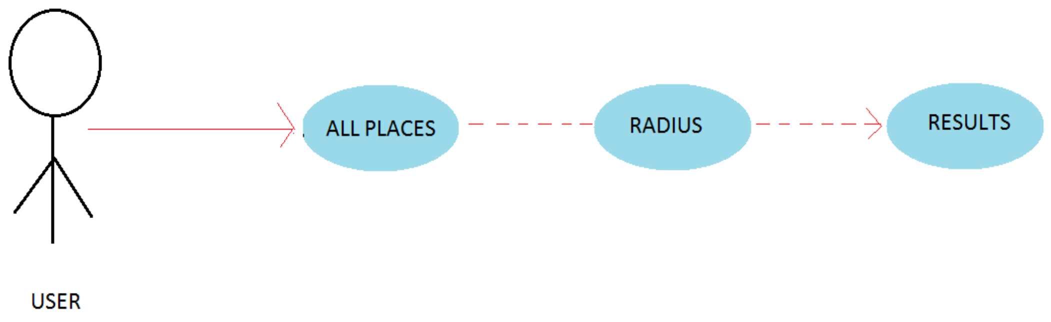
The Use Case Diagram for the user to share the location is shown above.

When the user wants to share, he is provided with various options. When the medium to share is selected, the application accesses the location implicitly and the location is returned as a link to the user. In this project, the share function is written implicitly.

The source code for share is written as follows

```
share.setOnClickListener(new View.OnClickListener()
 {
     @Override
     public void onClick(View v)
    {
      Double latitude = places.get(pos).getLocation().latitude;
      Double longitude =  places.get(pos).getLocation().longitude;
```

```
        String uri = "http://maps.google.com/maps?saddr=" + latitude+ "," +
                                        longitude;
        Intent sharingIntent = new Intent(android.content.Intent.ACTION_SEND);
                                sharingIntent.setType("text/plain");
        String ShareSub = "Here is my location";
        sharingIntent.putExtra(android.content.Intent.EXTRA_SUBJECT,
                                        ShareSub);
         sharingIntent.putExtra(android.content.Intent.EXTRA_TEXT, uri);
         startActivity(Intent.createChooser(sharingIntent, "Share via"));
        dismiss();
  }
}
```



The above shows the diagram when the user wishes to see all places around. The application provides the user the facility to choose the radius of distance. All the places which within this vicinity are shown as results with distance.

The source code can be given as follows:

```
setContentView(R.layout.activity_places);
                mPlaces = (ListView) findViewById(R.id.listView1);
                mSeekBar=(SeekBar)findViewById(R.id.seekBar1);
                mSeekBar.setMax(10000);
                mSeekBar.setProgress(1000);
                radius=(TextView)findViewById(R.id.radius);
                results=(TextView)findViewById(R.id.total);
                results.setText("Total places found : "+places.size());
```

5.2 Source Code

5.2.1 PlacesActivity Java Code

```
package com.imissionlabs.placesaroundme;
import java.io.*;
import java.net.*;
import java.util.*;
import android.*;
import org.json.*;
import com.google.android.gms.maps.model.LatLng;

public class PlacesActivity extends Activity
 {
        private ListView mPlaces;
        private ArrayList<Place> places = new ArrayList<Place>();
        private String query;
       private String URL =
"https://api.foursquare.com/v2/venues/search?client_id=5AS5TOFS3IG0RSN1KC4
LHC4ZAZ3NV1LBO0Q5X0CKXFBP1N2G&client_secret=NUVRKDB2105Y11O
Q5XRT25AOBW2KUNIVY3M1CZSIX52FVOGF&v=20130815&llimit=50";
```

```
private int pos;
private SeekBar mSeekBar;
private TextView radius,results;
private String radiusValue="1000";
@Override
protected void onCreate(Bundle savedInstanceState)
{
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_places);
        mPlaces = (ListView) findViewById(R.id.listView1);
        mSeekBar=(SeekBar)findViewById(R.id.seekBar1);
        mSeekBar.setMax(10000);
        mSeekBar.setProgress(1000);
        radius=(TextView)findViewById(R.id.radius);
        results=(TextView)findViewById(R.id.total);
        results.setText("Total places found : "+places.size());
        radius.setText("Radius : 1000 m");
        LatLng location = getIntent().getParcelableExtra("location");
        query = getIntent().getStringExtra("query");
        URL = URL + "&ll=" + location.latitude + "," + location.longitude
                        + "&query=" + query.trim()+"&radius=";

        GetPlacesActivity task = new GetPlacesActivity();
        task.execute();
        mPlaces.setOnItemClickListener(new OnItemClickListener()
        {
                @Override
                   public void onItemClick(AdapterView<?> arg0, View arg1, int
                                                                    arg2,long arg3)
                 {
                        pos=arg2;
            CustomAlertDialog d=new CustomAlertDialog(PlacesActivity.this);
                        d.show();
                }
        });
mSeekBar.setOnSeekBarChangeListener(newOnSeekBarChangeListener()
{
              @Override
                public void onStopTrackingTouch(SeekBar seekBar)
               {
                        // TODO Auto-generated method stub
                        places.clear();
                        GetPlacesActivity task = new GetPlacesActivity();
                        task.execute();


                }
                @Override
                public void onStartTrackingTouch(SeekBar seekBar) {
                        // TODO Auto-generated method stub

                }
               @Override
                public void onProgressChanged(SeekBar seekBar, int progress,
                                boolean fromUser)
               {
                        // TODO Auto-generated method stub
                        radiusValue=""+progress;

                        radius.setText("Radius :"+progress+" m");
                }
        });
}
public class CustomAlertDialog extends AlertDialog
 {
        private TextView navigate, share;
        public CustomAlertDialog(Context context)
         {
                super(context);
 }
@Override
protected void onCreate(Bundle savedInstanceState)
{
        super.onCreate(savedInstanceState);
        setContentView(R.layout.custom_alert_inapp);
        navigate = (TextView) findViewById(R.id.navigate);
        share = (TextView) findViewById(R.id.share);
        navigate.setOnClickListener(new View.OnClickListener()
        {
                @Override
                public void onClick(View v)
              {
                  Double latitude = places.get(pos).getLocation().latitude;
              Double longitude =  places.get(pos).getLocation().longitude;
Uri gmmIntentUri = Uri.parse("google.navigation:q="+latitude+","+longitude);
                  Intent mapIntent = new Intent(Intent.ACTION_VIEW,
                                                        gmmIntentUri);
                  mapIntent.setPackage("com.google.android.apps.maps");
                        startActivity(mapIntent);
                        dismiss();
                }
        });
```

```
                    share.setOnClickListener(new View.OnClickListener()
                  {
                          @Override
                            public void onClick(View v)
                            {
                              Double latitude = places.get(pos).getLocation().latitude;
                       Double longitude =  places.get(pos).getLocation().longitude;
                     String uri = "http://maps.google.com/maps?saddr=" + latitude
                                                            + "," + longitude;
    Intent sharingIntent = new Intent(android.content.Intent.ACTION_SEND);
                                  sharingIntent.setType("text/plain");
                      String ShareSub = "Here is my location";

   sharingIntent.putExtra(android.content.Intent.EXTRA_SUBJECT,
                                                 ShareSub);
   sharingIntent.putExtra(android.content.Intent.EXTRA_TEXT, uri);
   startActivity(Intent.createChooser(sharingIntent, "Share via"));
   dismiss();
                            }
                    });
   }
}
   private class GetPlacesActivity extends AsyncTask<Void, Void, Void>
```

```
{
        private ProgressDialog dialog;
        @Override
         protected void onPreExecute()
         {
                super.onPreExecute();
                dialog = ProgressDialog.show(PlacesActivity.this, null,
                                "Getting neary locations");
                dialog.setCancelable(false);
         }
         @Override
         protected Void doInBackground(Void... params)
         {
               String response = request(URL+radiusValue).toString();
                 try {
                        places.clear();
                        JSONObject obj = new JSONObject(response);
         JSONArray venues = obj.getJSONObject("response").getJSONArray(
                                                                  "venues");
                        for (int i = 0; i < venues.length(); i++)
                           {
                               JSONObject venue = venues.getJSONObject(i);
                               Place place = new Place();
```

```
                    place.setName(venue.getString("name"));
                JSONObject location = venue.getJSONObject("location");
                    StringBuilder sb = new StringBuilder();
                    sb.append(location.has("address") ? location
                    .getString("address") : "Address not available");
                    sb.append(location.has("crossStreet") ? location
                            .getString("crossStreet") : "");
                    place.setAddress(sb.toString());
                 place.setLocation(newLatLng(location.getDouble("lat"),
                            location.getDouble("lng")));
                    place.setDistance(location.getInt("distance"));
                    places.add(place);
           }
          }
         catch (JSONException e)
         {
             // TODO Auto-generated catch block
             e.printStackTrace();
          }
          return null;
     }
     @Override
     protected void onPostExecute(Void result)
```

```
        {
                super.onPostExecute(result);
                dialog.dismiss();
                System.out.println("SIZE :"+places.size());
                if (places.size() > 0) {
                        mPlaces.setVisibility(View.VISIBLE);
                        mPlaces.setAdapter(new PlacesAdpater());
                        results.setText("Total places found : "+places.size());

                } else {
                        mPlaces.setVisibility(View.GONE);
                        results.setText("Sorry! No places found");
                }
        }
}
private StringBuffer request(String urlString)
{
        System.out.println(urlString);
        StringBuffer chaine = new StringBuffer("");
        try {
                URL url = new URL(urlString);
                HttpURLConnection connection = (HttpURLConnection) url
                                .openConnection();
                        connection.setRequestMethod("GET");
                        connection.setDoInput(true);
                        connection.connect();
                        InputStream inputStream = connection.getInputStream();
                    BufferedReader rd = new BufferedReader(new InputStreamReader(
                                    inputStream));
                        String line = "";
                        while ((line = rd.readLine()) != null) {
                              chaine.append(line);
                        }
                  } catch (IOException e) {
                        // writing exception to log
                        e.printStackTrace();
                  }
                  return chaine;
}
private class PlacesAdpater extends BaseAdapter
 {
                  @Override
                  public int getCount() {
                        // TODO Auto-generated method stub
                        return places.size();
                  }
```

```
		@Override
		public Object getItem(int arg0) {
			return null;
		}
		@Override
		public long getItemId(int arg0) {
			// TODO Auto-generated method stub
			return 0;
		}

		@SuppressLint("ViewHolder")
		@Override
		public View getView(int arg0, View arg1, ViewGroup arg2)
		  {
			Place place = places.get(arg0);
			arg1                                                    =
LayoutInflater.from(PlacesActivity.this).inflate(R.layout.item_places, null);
			TextView name = (TextView) arg1.findViewById(R.id.place_name);
		TextView address = (TextView) arg1.findViewById(R.id.place_address);
	TextView distance = (TextView) arg1.findViewById(R.id.place_distance);
			if (place.getDistance() > 1000)
			{
				distance.setText(place.getDistance() / 1000 + " km");
```

```
                }
               else {
                        distance.setText(place.getDistance() + " m");
                }
                name.setText(place.getName());
                address.setText(place.getAddress());
                return arg1;
        }
    }
}
```

5.2.2 Place Java Code

```
package com.imissionlabs.placesaroundme;

import com.google.android.gms.maps.model.LatLng;

public class Place {

    private String name;
    private String address;
    private int distance;
    private LatLng location;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getAddress() {
        return address;
```

```
	}
	public void setAddress(String address) {
		this.address = address;
	}
	public int getDistance() {
		return distance;
	}
	public void setDistance(int distance) {
		this.distance = distance;
	}
	public LatLng getLocation() {
		return location;
	}
	public void setLocation(LatLng location) {
		this.location = location;
	}

}
```

5.2.3 HomeScreenActivity Java Code

```
package com.imissionlabs.placesaroundme;
import android.*;
import com.google.android.gms.maps.*;

public class HomeScreenActivity extends FragmentActivity implements
		OnMyLocationChangeListener, OnClickListener
{

	private GoogleMap map;
	private ProgressDialog dialog;
	private LatLng myLocation;
	private Marker m;
	private Button shareLocation;
	private ImageView search;
	private EditText searchQuery;
	@Override
	protected void onCreate(Bundle arg0) {
		super.onCreate(arg0);
		setContentView(R.layout.activity_main);
		map = ((SupportMapFragment) getSupportFragmentManager()
				.findFragmentById(R.id.map)).getMap();
		shareLocation = (Button) findViewById(R.id.sharelocation);
		searchQuery=(EditText) findViewById(R.id.searchTxt);
		search=(ImageView)findViewById(R.id.search);
		map.setMyLocationEnabled(true);
		turnGPSOn();
		map.setOnMyLocationChangeListener(this);
		dialog = ProgressDialog.show(this, null, "Fetching current location");
		dialog.setCancelable(false);
		shareLocation.setOnClickListener(this);
		search.setOnClickListener(this);

	}

	private void turnGPSOn() {
		String provider = Settings.Secure.getString(getContentResolver(),
				Settings.Secure.LOCATION_PROVIDERS_ALLOWED);

		if (!provider.contains("gps")) { // if gps is disabled
			final Intent poke = new Intent();
			poke.setClassName("com.android.settings",

	"com.android.settings.widget.SettingsAppWidgetProvider");
			poke.addCategory(Intent.CATEGORY_ALTERNATIVE);
			poke.setData(Uri.parse("3"));
			sendBroadcast(poke);
		}
	}

	@Override
	public void onMyLocationChange(Location location) {
		if (m != null) {
			m.remove();
			m = null;
		}
```

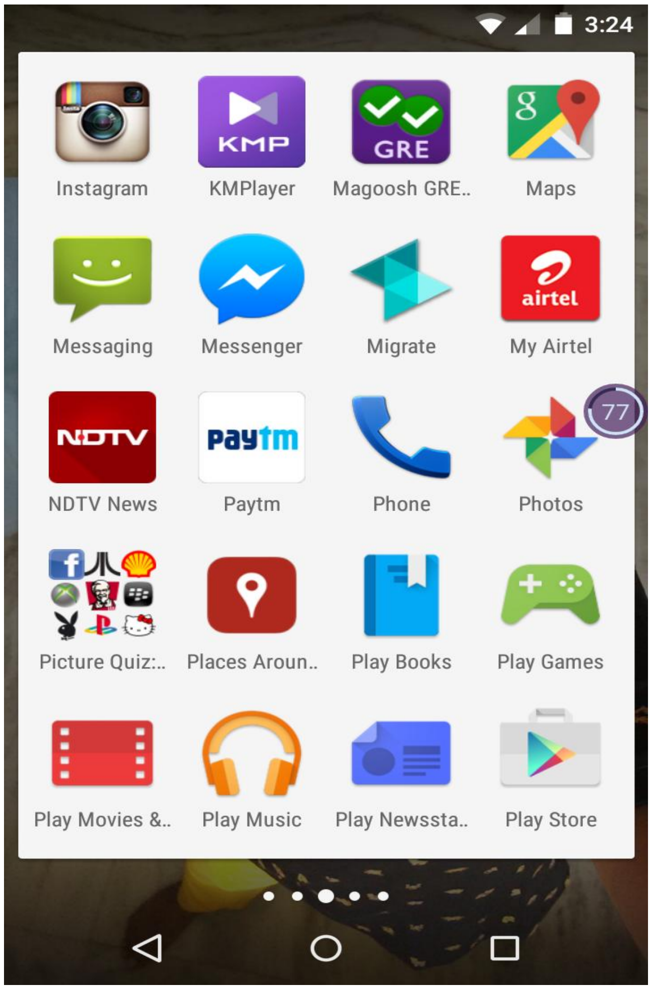
```
                myLocation = new LatLng(location.getLatitude(),
location.getLongitude());
                MarkerOptions markerOptions = new MarkerOptions();
                markerOptions.position(myLocation);
                markerOptions.icon(BitmapDescriptorFactory

        .defaultMarker(BitmapDescriptorFactory.HUE_GREEN));
                markerOptions.title("Hey! I am here..");
                m = map.addMarker(markerOptions);
                m.showInfoWindow();

        map.animateCamera(CameraUpdateFactory.newLatLngZoom(myLocation,
15));
                dialog.dismiss();
        }

        @Override
        public void onClick(View v) {
                switch (v.getId()) {
                case R.id.sharelocation:
                        if (myLocation != null) {

                                Double latitude = myLocation.latitude;
                                Double longitude = myLocation.longitude;

                                String uri = "http://maps.google.com/maps?saddr=" +
latitude + "," + longitude;

                                Intent sharingIntent = new Intent(
                                                android.content.Intent.ACTION_SEND);
                                sharingIntent.setType("text/plain");
                                String ShareSub = "Here is my location";

        sharingIntent.putExtra(android.content.Intent.EXTRA_SUBJECT,
                                                ShareSub);
        sharingIntent.putExtra(android.content.Intent.EXTRA_TEXT, uri);
        startActivity(Intent.createChooser(sharingIntent, "Share via"));
                        }
                        break;
        case R.id.search:
                        Intent intent=new Intent(HomeScreenActivity.this,
PlacesActivity.class);
                        intent.putExtra("location", myLocation);
                        intent.putExtra("query", searchQuery.getText().toString());
                        startActivity(intent);
                        break;

                default:
                        break;
                }
        }
}
```

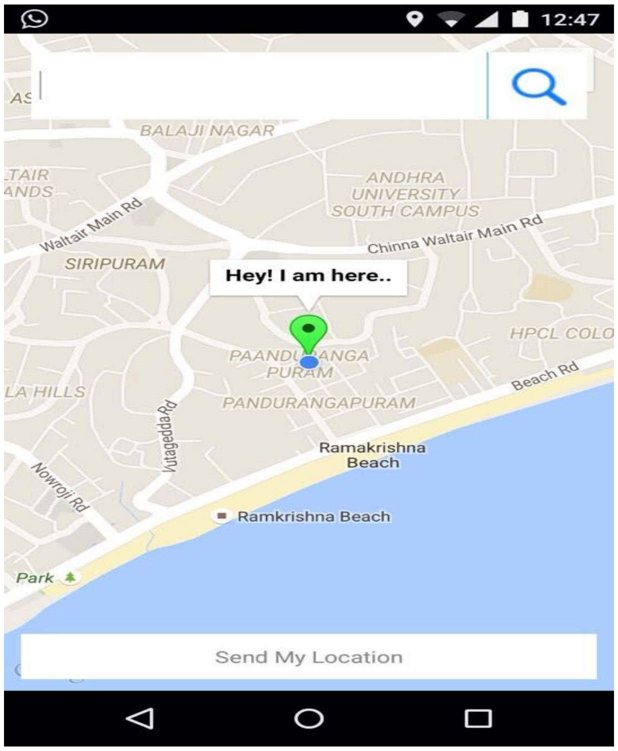
5.2.4 AndroidManifest XML Code

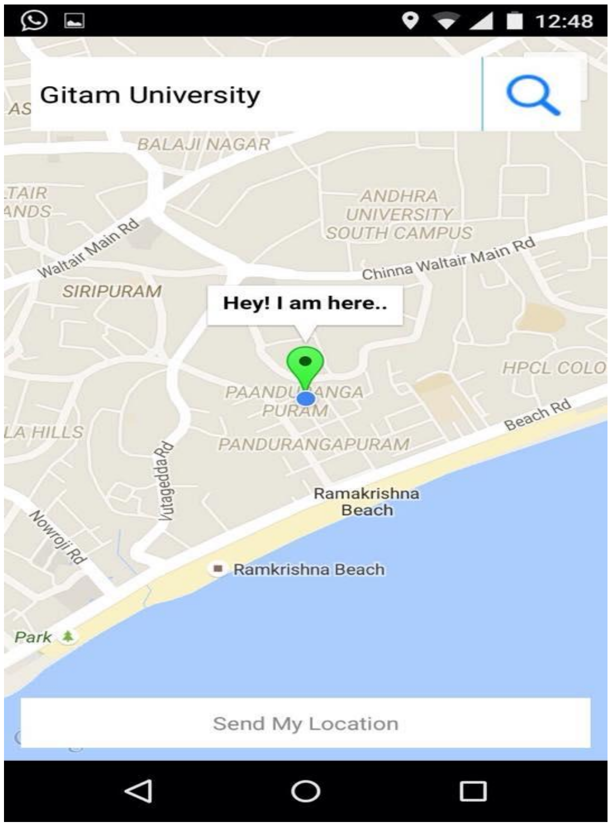
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.imissionlabs.placesaroundme"
    android:versionCode="1"
    android:versionName="1.0" >

<uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="21" />

    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <meta-data
            android:name="com.google.android.maps.v2.API_KEY"
            android:value="AIzaSyDK46slbXEf-AqaEFlBKgxTF2VgfRv4Rvk" />
        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />

        <activity
            android:name="HomeScreenActivity"
            android:theme="@android:style/Theme.Black.NoTitleBar" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="PlacesActivity">

        </activity>
    </application>

</manifest>
```

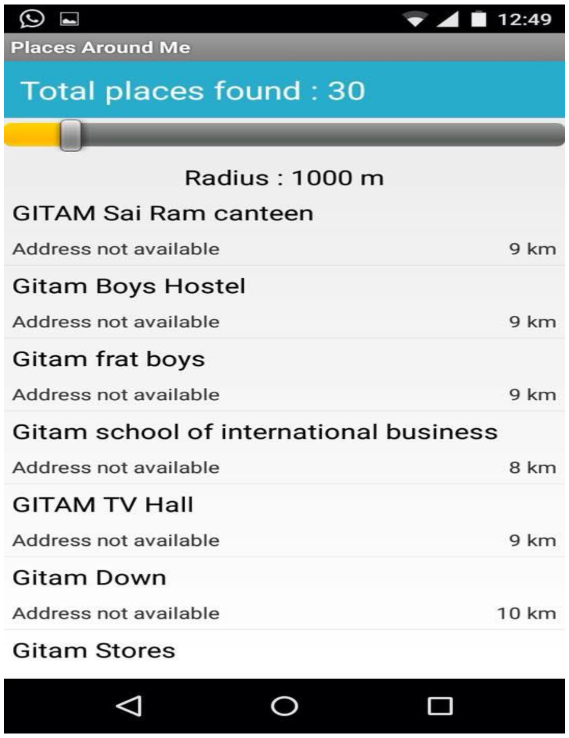
5. RESULTS

5.1 Application Interface Screenshots




12:47
BALAJI NAGAR
ANDHRA UNIVERSITY SOUTH CAMPUS
Waltair Main Rd
Chinna Waltair Main Rd
SIRIPURAM
Hey! I am here..
HPCL COLO
PAANDURANGA PURAM
PANDURANGAPURAM
Beach Rd
Vutagedda Rd
Ramakrishna Beach
Nowroji Rd
Ramkrishna Beach
Park
Send My Location


12:48
Gitam University
BALAJI NAGAR
ANDHRA UNIVERSITY SOUTH CAMPUS
Waltair Main Rd
Chinna Waltair Main Rd
SIRIPURAM
Hey! I am here..
HPCL COLO
PAANDURANGA PURAM
PANDURANGAPURAM
Beach Rd
Vutagedda Rd
Ramakrishna Beach
Nowroji Rd
Ramkrishna Beach
Park
Send My Location



12:49

Places Around Me

Total places found : 30

Radius : 1000 m

GITAM Sai Ram canteen
Address not available 9 km

Gitam Boys Hostel
Address not available 9 km

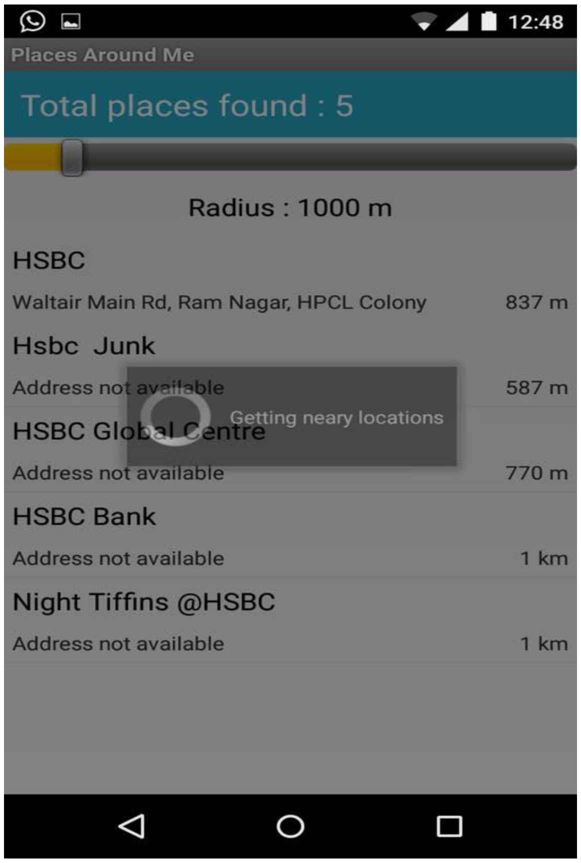
Gitam frat boys
Address not available 9 km

Gitam school of international business
Address not available 8 km

GITAM TV Hall
Address not available 9 km

Gitam Down
Address not available 10 km

Gitam Stores



12:48

Places Around Me

Total places found : 5

Radius : 1000 m

HSBC
Waltair Main Rd, Ram Nagar, HPCL Colony 837 m

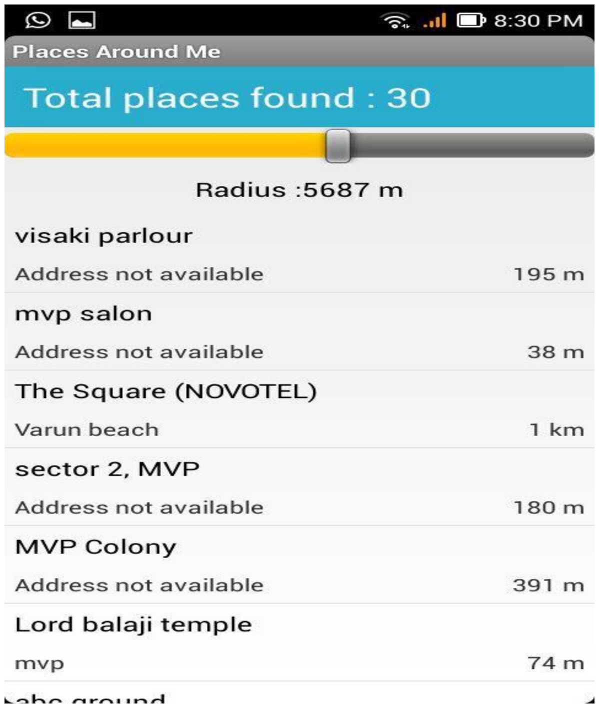
Hsbc Junk
Address not available 587 m

HSBC Global Centre
Address not available 770 m

HSBC Bank
Address not available 1 km

Night Tiffins @HSBC
Address not available 1 km

Getting neary locations



8:30 PM

Places Around Me

Total places found : 30

Radius :5687 m

visaki parlour
Address not available 195 m

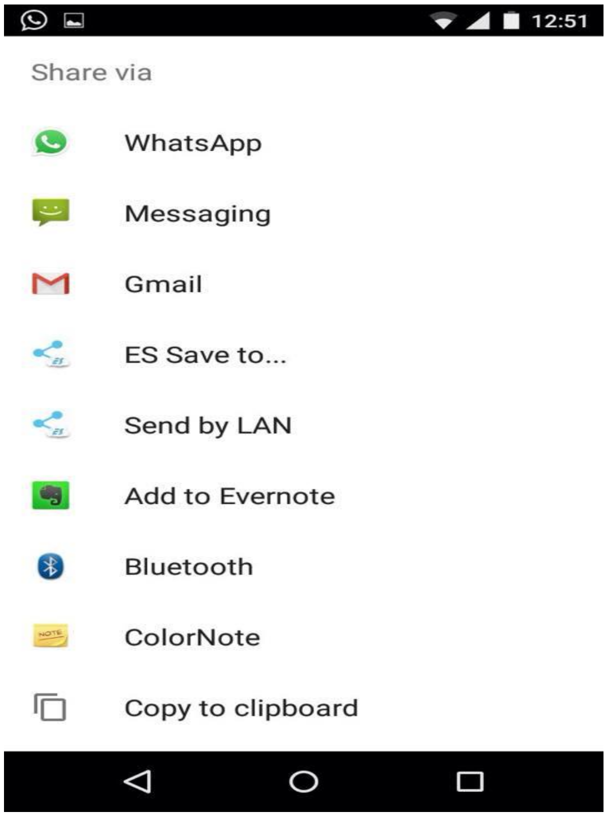
mvp salon
Address not available 38 m

The Square (NOVOTEL)
Varun beach 1 km

sector 2, MVP
Address not available 180 m

MVP Colony
Address not available 391 m

Lord balaji temple
mvp 74 m



12:51

Share via

WhatsApp

Messaging

Gmail

ES Save to...

Send by LAN

Add to Evernote

Bluetooth

ColorNote

Copy to clipboard

6. CONCLUSION

Best efforts have been put into the project to make it real and user friendly. This application is very identical in its kind, unlike other applications that are existent on the Play Store. This will be highly appreciated and will be handy for any visitors to a place where they don't have any clue of their surroundings. The Android Application ***"Places Around Me"*** was successfully concluded by meeting the requirements specified by the software requirement document.