1. INTRODUCTION

Navigation/Map applications for Smart phones are quite useful in the day-to-day life. There are lots of applications available in market which provides efficient and user friendly navigation to the user. Most of the applications are successful in assisting the user with his current location and providing directions to particular destination for outdoor locations. In most of the scenarios this is achieved using the GPS unit of the Smartphone. But accurate navigation while not in line of sight with GPS satellites is still a challenge. There is a limitation for smart phones to locate their exact position while in covered areas such as shopping malls, airports, railway stations, multi storied buildings, apartments. There are in door navigation systems available in the market which uses Bluetooth, Wi-Fi, AGPS or RFID. Bluetooth requires expensive receivers and the accuracy of Bluetooth navigation depends upon the number of cells used. Wi-Fi also demands expensive access points for indoor navigation. AGPS uses network assistance servers for indoor navigation

Using AGPS technique, accuracy is very much limited because of approximation. It involves infrastructure cost for provider and the user. RFID requires active tags for indoor navigation, where the accuracy is directly proportional to the number of active tags used. Active tags are self-powered and hence costly. Also close pass by is required to sense RFIDs and even the user need to be aware of the RFID position. Most of the existing solutions are far from providing an accurate and cost effective indoor navigation.

2. OBJECTIVES

QR code (abbreviated from **Quick Response Code**) is the trademark for a type of matrix barcode (or two-dimensional barcode) first designed for the automotive industry in Japan. A barcode is a machine-readable optical label that contains information about the item to which it is attached. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte / binary, and kanji) to efficiently store data; extensions may also be used.

The QR Code system has become popular outside the automotive industry due to its fast readability and greater storage capacity compared to standard UPC barcodes. Applications include product tracking, item identification, time tracking, document management, general marketing, and much more.

A QR code consists of black modules (square dots) arranged in a square grid on a white background, which can be read by an imaging device (such as a camera) and processed using Reed–Solomon error correction until the image can be appropriately interpreted. The required data are then extracted from patterns present in both horizontal and vertical components of the image.

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 Purpose of this document

This SRS describes the function and the performance allocated to our product. It provides a reference for the validation of the final product. SRS provides an overview of the product including functional and non-functional requirements, abbreviations used, product and functions etc.

3.2 Scope of Developing Project

This web application has many scopes for future expansion. In our project we provide,

- An accurate navigation inside a closed building
- To Provide the location details to the user
- Finding shortest path to the destination.

3.3 General Description

Some of the important features that this software should deliver are,

- Floor plan which shows the shortest path from source to destination.
- Distance that he should travel.
- Zooming and zoom out the floor plan.

4. SYSTEM STUDY

4.1 EXISTING SYSTEM

There are lots of applications available in market which provides efficient and user friendly navigation to the user. Most of the applications are successful in assisting the user with his current location and providing directions to particular destination for outdoor locations. In most of the scenarios this is achieved using the GPS unit of the Smartphone. But accurate navigation while not in line of sight with GPS satellites is still a challenge. There is a limitation for smart phones to locate their exact position while in covered areas such as shopping malls, airports, railway stations, multi storied buildings, apartments. There are in door navigation systems available in the market which uses Bluetooth, Wi-Fi, AGPS or RFID. Bluetooth requires expensive receivers and the accuracy of Bluetooth navigation depends upon the number of cells used. Wi-Fi also demands expensive access points for indoor navigation. AGPS uses network assistance servers for indoor navigation

4.2 PROPOSED SYSTEM

QR code (abbreviated from **Quick Response Code**) is the trademark for a type of matrix barcode. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte / binary, and kanji) to efficiently store data; extensions may also be used. A QR code consists of black modules (square dots) arranged in a square grid on a white background, which can be read by an imaging device (such as a camera) and processed using Reed–Solomon error correction until the image can be appropriately interpreted. The required data are then extracted from patterns present in both horizontal and vertical components of the image.

4.3 FEASIBILITY AND CRITICAL FACTORS

A feasibility study is a test of system proposal according to its workability, impact on the organization, ability to meet user needs and effective use of resources.

The objective of feasibility study is not to solve the problem, but to acquire a sense of its scope. During the study, the problem definition is crystallized and aspects of problem to be included in the system are determined, consequently costs and benefits are estimated with greater detail at this state. The result of feasibility study is a system formal proposal. This is simply a form of documenting or detailing the nature and scope of proposed solution. The proposed summarizes what is known and what going to be done.

The key considerations involved in feasibility analysis are technical, economic and operational.

Technical feasibility

Technical feasibility is the most important of all types of feasibility analysis. Technical feasibility deals with hardware as well as software requirements. An idea from the outline design to system requirements in terms of inputs, outputs, files and procedures is drawn and type of hardware, software and the methods required for running the system are analyzed. Keeping in mind of the above considerations, the resource availability at his company was observed. It was found that the company has the sufficient resources to develop the current product; hence the system is technically feasible.

Economic feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of the software, more commonly known as the cost or benefits analysis. The procedure is to determine the benefits and savings that are expected from a candidate system and compare them with cost. If the benefits outweigh cost, the decision is made to design and implement the system; otherwise further alternatives have to be made. Here it is seen that no new hardware or software is needed for the development of the system. Hence the project is economically feasible for development in this system.

Schedule feasibility

Schedule feasibility is concerned with the completion of the project development within the fixed time span. It is an important factor as it can affect other factors like machine availability, tools, cost development and delay in the development of other systems. Besides these, this project is assigned to the student as an academic exercise to be completed within a fixed period of time.

Operational feasibility

The purpose of the operational feasibility study is to determine whether the new system would be used if it is developed and implemented? Will there be a resistance from users that will undermine the possible application benefits? From the output of the meeting that was held with the system user, it was found that all of them support the development of new system. The positive response from them encouraged in building such a system.

5. SYSTEM DESIGN

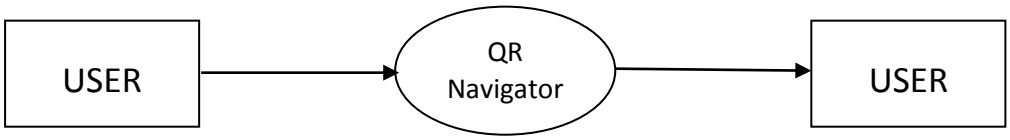
5.1 MODULE

There are 3 modules in this project.

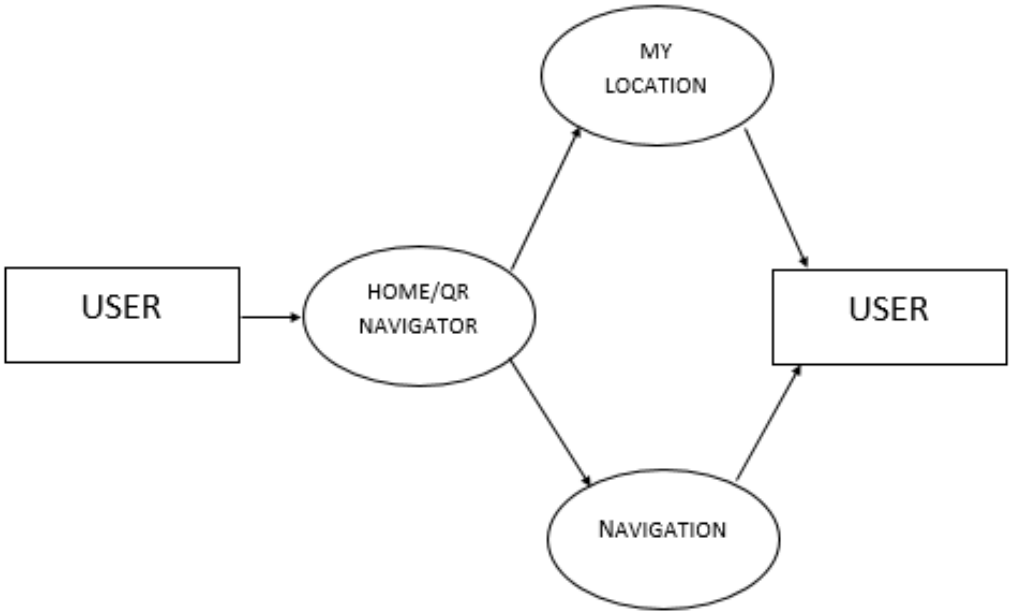
- QR CODE SCANNING
  - User scans the QR Code
  - QR codes are two dimensional codes where the data is encoded in an optically readable format
  - Navigation application in the mobile uses the camera to read frames continuously
  - Using camera of 2MB or more to scan QR Code
- FINDING SHORTEST PATH
  - Executing Dijikstra's Algorithm
  - Finding shortest path from the graph
- DISPLAYING MAP
  - Once a QR code is found, the application decodes the QR code and obtains the floor map and location details
  - It decodes the geo location details from QR code and points the location in the Map

5.2 DATA FLOW DIAGRAMS

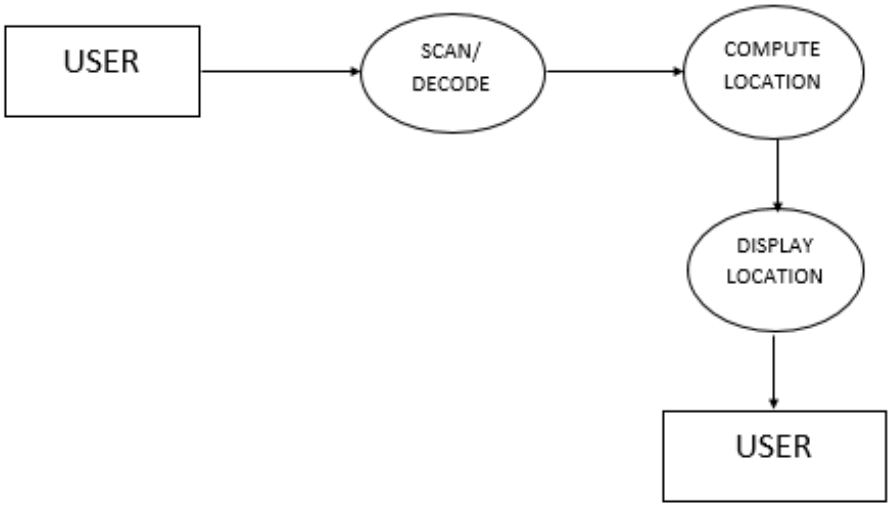
LEVEL 0



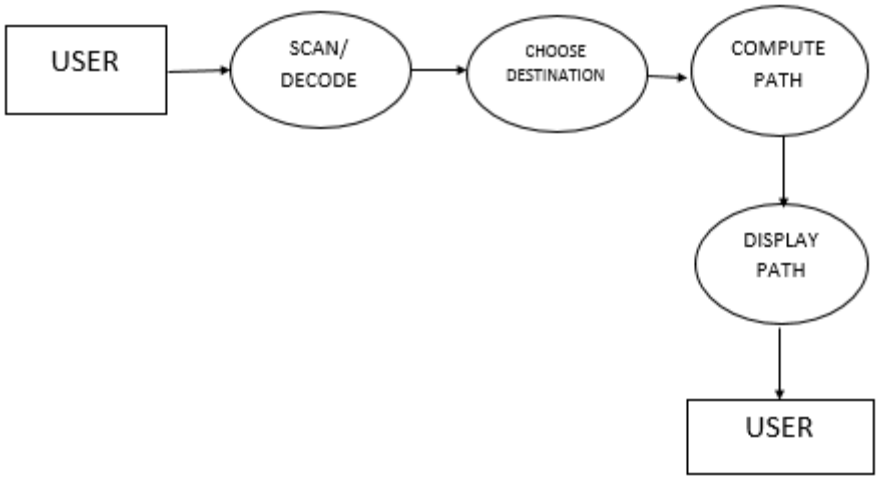
LEVEL 1



LEVEL 2 FOR PROCESS 1.1



LEVEL 2 FOR PROCESS 1.2

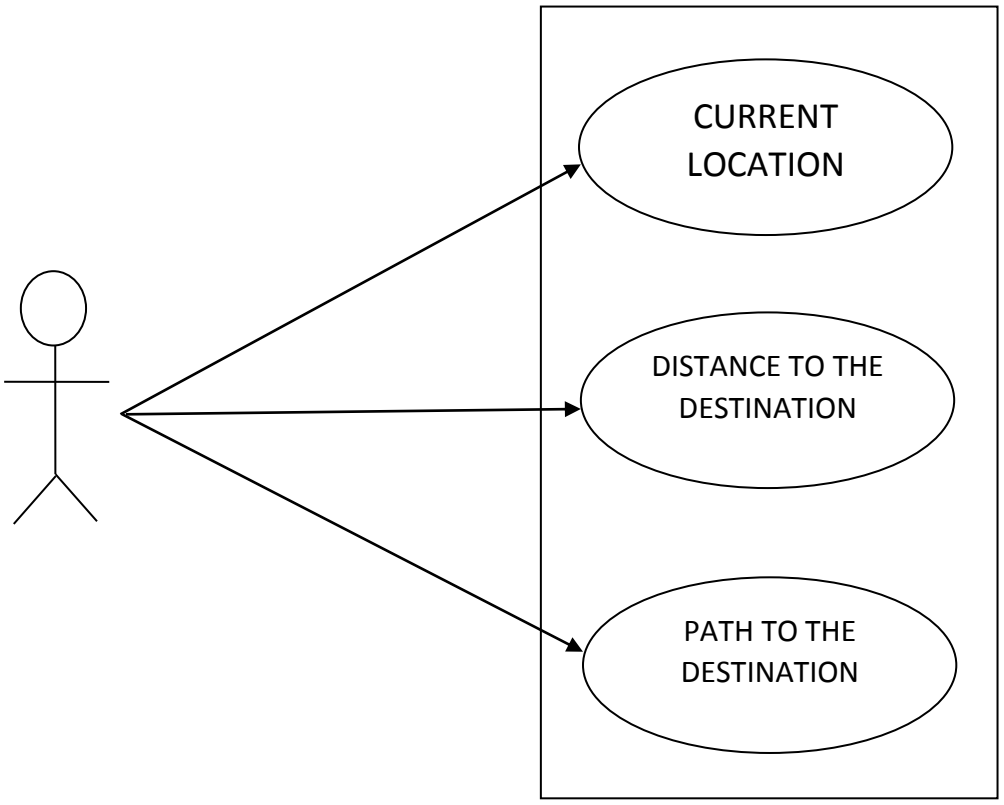


6. USE CASE DIAGRAM

Use case is a set of scenarios that describing an interaction between a user and a system. The two main component of use case diagram are use cases and actors. They are helpful in exposing requirements and planning the projects. During the initial stage of project, most use cases should be defined. But as the project continues more might become visible, the use case diagram for showing the project is that it measures the size of the software. It reduces users need for training and operating the system.

In use case diagram, we all know, the main two things are the actors and the actions. Here the actors is the user. Apart from the actors there are actions to be performed by these actors. These actions are shown in oval.

Use case diagrams are relatively easy UML diagram. They are helpful in exposing requirements and planning of the project. During the initial stage of a project most use cases should be defined, but as the project continues more might become visible. A use case is an external view of the system that represents some action the user might perform in order to complete a task. A use case diagram displays the relationship among actors and use cases.



7. DETAILED DESIGN

7.1 MODULE DESCRIPTION

7.1.1 QR CODE SCANNING

Originally designed for industrial uses, QR codes have become common in consumer advertising. Typically, a smartphone is used as a QR code scanner, displaying the code and converting it to some useful form (such as a standard URL for a website, thereby obviating the need for a user to type it into a web browser). The QR code has become a focus of advertising strategy, since it provides a way to access a brand's website more quickly than by manually entering a URL. Beyond mere convenience to the consumer, the importance of this capability is that it increases the conversion rate (the chance that contact with the advertisement will convert to a sale), by coaxing interested prospects further down the conversion funnel with little delay or effort, bringing the viewer to the advertiser's website immediately, where a longer and more targeted sales pitch may lose the viewer's interest.

Although initially used to track parts in vehicle manufacturing, QR codes are now (as of 2012) used over a much wider range of applications, including commercial tracking, entertainment and transport ticketing, product/loyalty marketing (examples: mobile couponing where a company's discounted and percent discount can be captured using a QR code decoder which is a mobile app, or storing a company's information such as address and related information alongside its alpha-numeric text data as can be seen in Yellow Pages directory), and in-store product labeling. It can also be used in storing personal information for use by organizations. An example of this is Philippines National Bureau of Investigation (NBI) where NBI clearances now come with a QR code. Many of these applications target mobile-phone users (via mobile tagging). Users may receive text, add a vCard contact to their device, open a URI, or compose an e-mail or text message after scanning QR codes. They can generate and print their own QR codes for others to scan and use by visiting one of several pay or free QR code-generating sites or apps. Google had a popular API to generate QR codes, and apps for scanning QR codes can be found on nearly all smartphone devices.

7.1.2 Dijikstra's Algorithm

Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It was conceived by computer scientist Edsger W. Dijkstra in 1956 and published in 1959. The algorithm exists in many variants; Dijkstra's original variant found the shortest path between two nodes, but a more common variant fixes a single node as the "source" node and finds shortest paths from the source to all other nodes in the graph, producing a shortest path tree.
Dijkstra's original algorithm does not use a min-priority queue and runs in time $O(|V|^2)$ (where $|V|$ is the number of nodes). The idea of this algorithm is also given in (Leyzorek et al. 1957). The implementation based on a min-priority queue implemented by a Fibonacci heap and running in $O(|E|+|V| \log |V|)$ (where $|E|$ is the number of edges) is due to (Fredman & Tarjan 1984). This is asymptotically the fastest known single-source shortest-path algorithm for arbitrary directed graphs with unbounded non-negative weights.

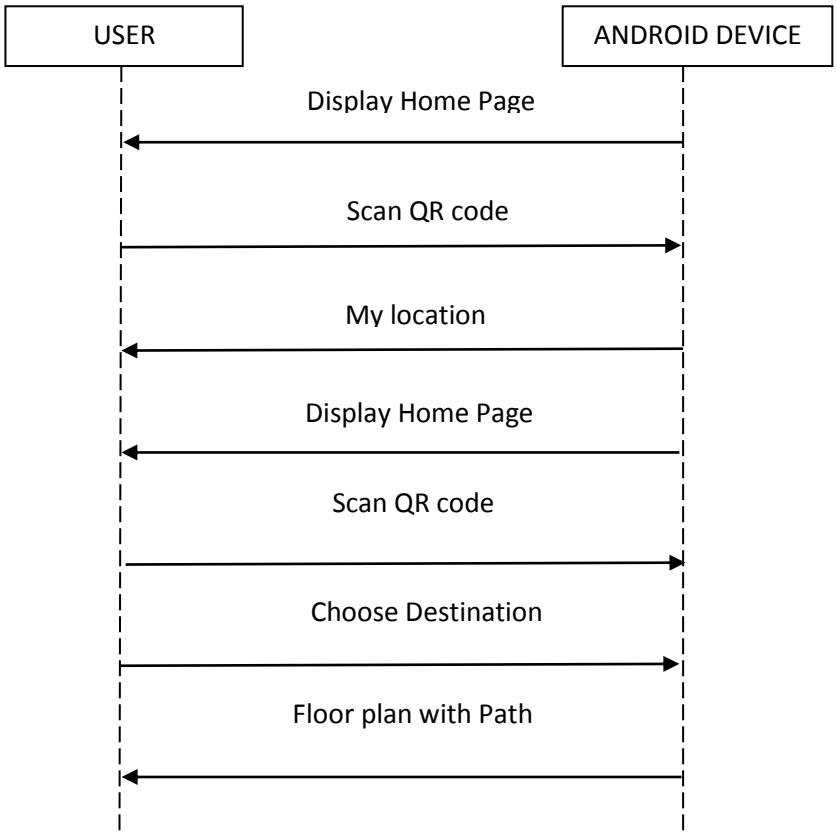
- DDISPLAYING MAP
  - Once a QR code is found, the application decodes the QR code and obtains the floor map and location details
  - It decodes the geo location details from QR code and points the location in the Map.

7.2 SEQUENCE DIAGRAM

7.2.1 PURPOSE

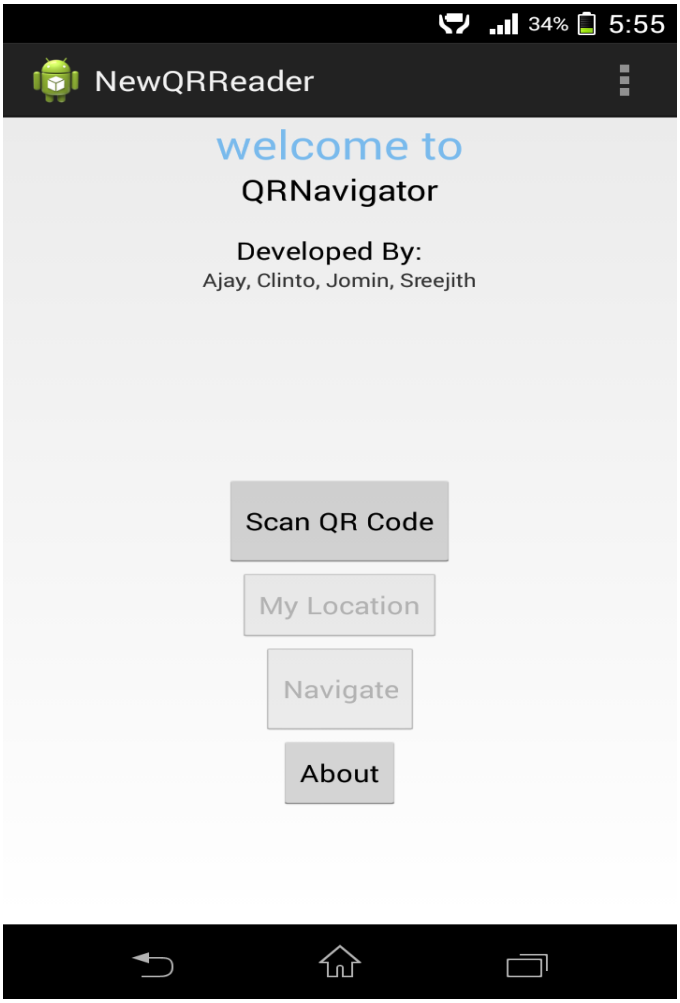
Sequence diagram tell the manner in which interaction takes place and the style in which it is executed. They generally show the sequence in which the diagram demonstrates how objects are statically connected. It demonstrates the behaviour of an object in a use case by describing the actions they performed. The object is represented in rectangle and existence of objects is shown by vertical lines. The main purpose of a sequence diagram is to define event sequences that results in some desired outcome. The focus is less on events themselves and

more on the order in which events occur; nevertheless, most sequence diagrams will communicate what information are sent between a system's objects is information along the horizontal well as the order in which they occur. The diagram conveys this information along the horizontal and vertical dimensions: the vertical dimensions shows, top down, the time sequence of instructions as they occur, and the horizontal dimension shows, left to right, the object instances that the instructions are sent to. Here in this there are some rules to draw the diagram. They are explained as follows: When drawing a sequence diagram, lifeline notation elements are placed across the top of the diagram. Lifelines represent either roles or object instances that participate in the sequence being modelled. Lifelines are drawn as a box with a dashed line descending from the centre of the bottom edge. The lifeline's name is placed inside the box. It shows the way the different components of project interact with each other from starting to end .The invocation of methods in each object, and the order in which the invocation occurs is captured in a Sequence diagram. This makes the Sequence diagram a very useful tool to easily represent the dynamic behaviour of a system.

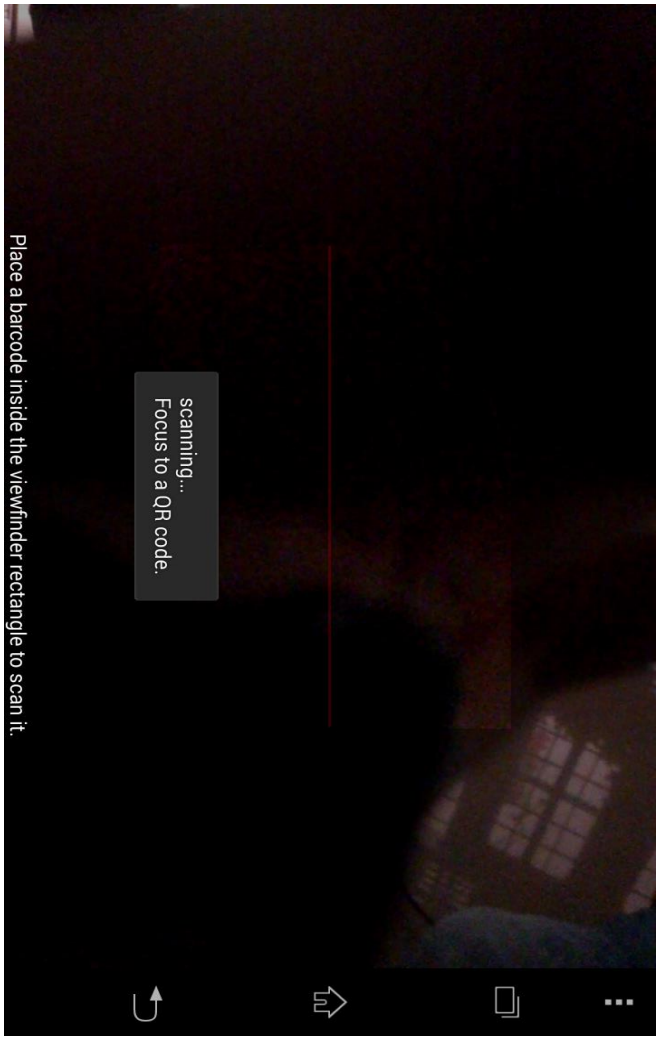


8. SCREEN SHOTS

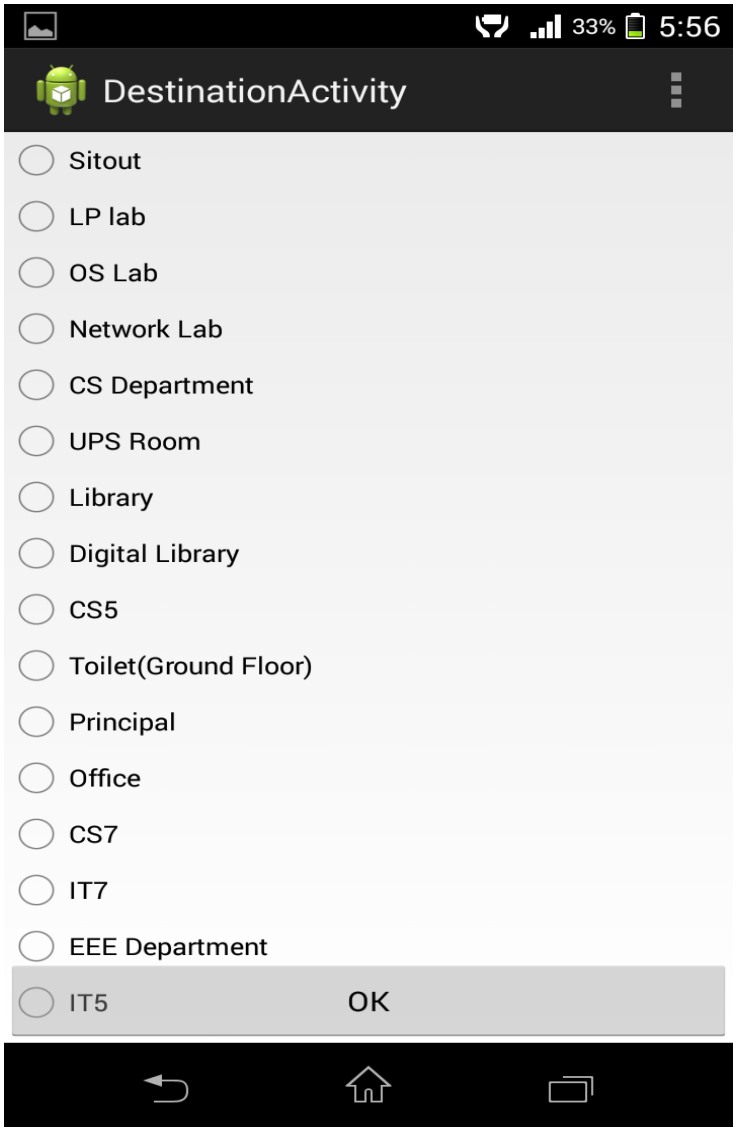
8.1 WELCOME PAGE



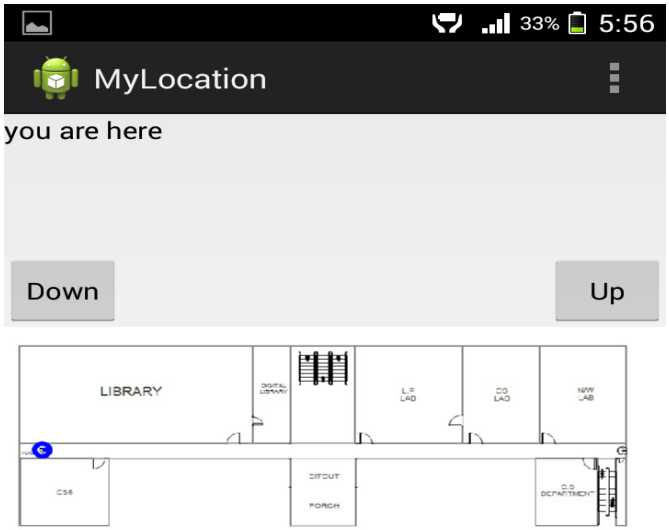
8.2 SCANNING


Place a barcode inside the viewfinder rectangle to scan it.
scanning...
Focus to a QR code.

8.3SELECT DESTINATION

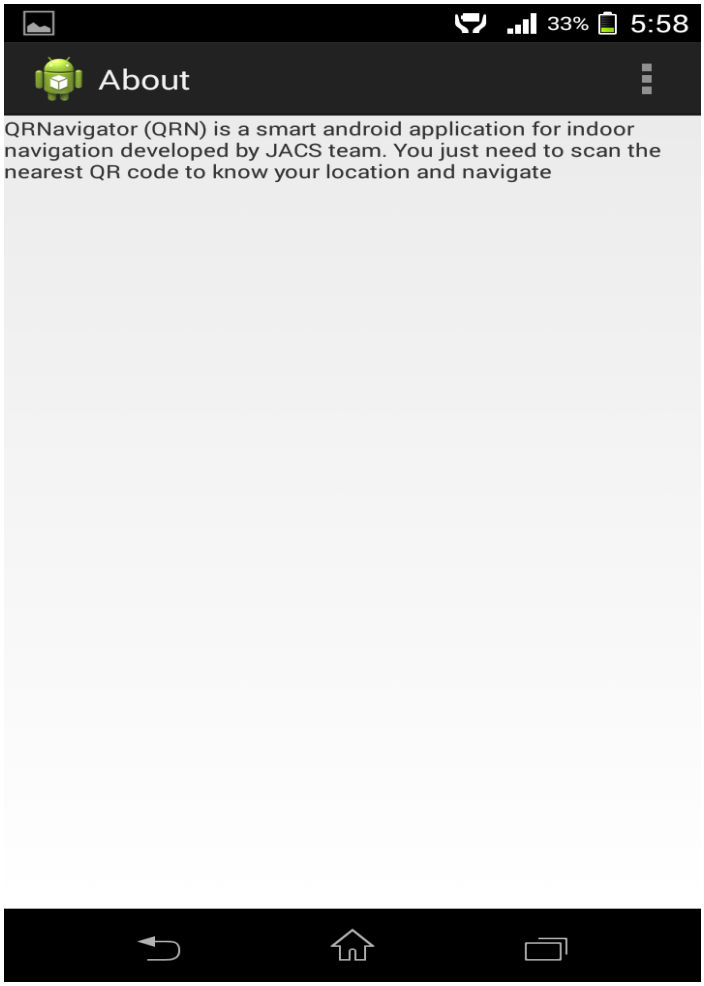


8.4 FLOOR PLAN



8.5 DIRECTIONS

8.6 ABOUT


33% 5:58
About
QRNavigator (QRN) is a smart android application for indoor navigation developed by JACS team. You just need to scan the nearest QR code to know your location and navigate

9. SYSTEM IMPLEMENTATION

9.1 Software Requirements

Operating system : Android

Platform : Smart phone

9.2 Hardware Requirements

Memory size : 256 MB RAM

Storage : 5 MB (SD Card or Phone)

Keyboard : Virtual keyboard with 102 keys

Device : Android Mobile with 2 MB Camera

9.3 Familiarization with the tools

About Android and Android programming

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touchscreen input, it also has been used in game consoles, digital cameras, regular PCs (e.g. the HP Slate 21) and other electronics.

10. SOURCE CODE

10.1 Main Activity

```
package com.example.newqrreader;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import com.google.zxing.client.android.CaptureActivity;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends Activity {
	static String source="",myflr="",mypix="",mynof="";
	String tmp="",tmp1="";
	int i=0,j=0;
	Button btn_scan,btn_navigate,btn_myloc,btn_abt;
	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_main);
		btn_scan=(Button)findViewById(R.id.button_scan);
		btn_navigate=(Button)findViewById(R.id.button_navi);
```

```
btn_myloc=(Button)findViewById(R.id.myloc);
btn_abt=(Button)findViewById(R.id.abt);
btn_navigate.setEnabled(false);
btn_myloc.setEnabled(false);
btn_abt.setOnClickListener(new OnClickListener() {
	@Override
	public void onClick(View arg0) {
	Intent callIntent = new Intent(getApplicationContext(), About.class);
		startActivity(callIntent);
		}
});
btn_myloc.setOnClickListener(new OnClickListener() {
	@Override
	public void onClick(View arg0) {
		//to read pixels from pixel file
FileReader fin2 = null;
tmp1="";
tmp="";
		try {
			fin2 = new
FileReader("/mnt/sdcard/qrnavigator/file2.txt");
		} catch (FileNotFoundException e) {
				e.printStackTrace();
		}
BufferedReader pixelFile = new BufferedReader(fin2);
		try {
			while((tmp = pixelFile.readLine())!=null)
			{
				tmp1+=tmp;
			}
```

```
                        } catch (IOException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }
                        mynof=tmp1.substring(0, 2);
                        MyClipboardManager obj1 = new MyClipboardManager();
                        source = obj1.readFromClipboard(getBaseContext());
                i=tmp1.lastIndexOf("V"+source+"-");
                j=tmp1.indexOf("-", i);
                myflr=tmp1.substring(j+1, j+3);
                        mypix=tmp1.substring(i-8, i-1);
                        Toast.makeText(getApplicationContext(), myflr+"\n"+mypix,
Toast.LENGTH_LONG).show();
        Intent callIntent = new Intent(getApplicationContext(), MyLocation.class);
                        startActivity(callIntent);
                }
        });
        scanQRimage();
        selectDestination();
    }

    public void scanQRimage(){

        btn_scan.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
            Toast.makeText(getApplicationContext(), "scanning...\nFocus to a QR
code.", Toast.LENGTH_LONG).show();
Intent callIntent = new Intent("com.google.zxing.client.android.SCAN");
                        startActivity(callIntent);
                        btn_myloc.setEnabled(true);
                        btn_navigate.setEnabled(true);
```

```
                }
        });
    }
    public void selectDestination(){
        btn_navigate.setVisibility(View.VISIBLE);
        btn_navigate.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
                MyClipboardManager obj = new MyClipboardManager();
                source = obj.readFromClipboard(getBaseContext());
                Toast.makeText(getApplicationContext(), "destination:
"+source, Toast.LENGTH_LONG).show();
                Intent intent=new Intent(getApplicationContext(),
DestinationActivity.class);
                startActivity(intent);
            }
        });
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
            getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

10.2 Destination Activity

```
package com.example.newqrreader;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
```

```
import java.io.IOException;
import android.R.integer;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.view.ViewGroup;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;
import android.widget.RadioGroup.OnCheckedChangeListener;

public class DestinationActivity extends Activity {
	static int destination=0,pos=0;
	Button btn_ok;
	RadioGroup rdgr;
	static int[] placeid=new int[100];
	int[] x=new int[100];
	int[] y=new int[100];
	String tmp="",tmp1="",tmp2="";
	String node="",pixel="",output="";
	static int cflr=-1,nof=0;
	static String dist="",filecnt = "",pathlist="",xyz="";
	static String[] path = new String[100];
	static String[] nodelist = new String[100];
	static String[] destinations = new String[100];
	int i=0,k=0,j=0,p=0,flr=0,maxdstno=0;
```

```
float m1=0,m2=0;
double ang1=0,ang2=0;
public static int ip2=0,ip1=0;
MainActivity ma=new MainActivity();
@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_destination);
        btn_ok=(Button)findViewById(R.id.button_destok);
        cflr=-1;
FileReader fin1 = null;
        try {
                fin1 = new FileReader("/mnt/sdcard/qrnavigator/file3.txt");
        } catch (FileNotFoundException e) {
                e.printStackTrace();
        }
BufferedReader graphFile = new BufferedReader(fin1);
        try {
                maxdstno=Integer.parseInt(graphFile.readLine());
        } catch (NumberFormatException e1) {
                e1.printStackTrace();
        } catch (IOException e1) {
                e1.printStackTrace();
        }
   ViewGroup hourButtonLayout = (ViewGroup) findViewById(R.id.hour_radio_group);
          for (int i = 0; i<maxdstno; i++) {
            RadioButton button = new RadioButton(this);
            button.setId(i);
            try {
                 destinations[i]=graphFile.readLine();
```

```
                                button.setText(destinations[i]);
                                placeid[i]=Integer.parseInt(graphFile.readLine());
                        } catch (IOException e) {
                                e.printStackTrace();
                    hourButtonLayout.addView(button);}
                   final RadioGroup rg = (RadioGroup)findViewById(R.id.hour_radio_group);
                rg.setOnCheckedChangeListener(new OnCheckedChangeListener() {
                        @Override
                public void onCheckedChanged(RadioGroup group, int checkedId) {
                        pos=rg.indexOfChild(findViewById(checkedId));
                        destination = placeid[pos];
                        Toast.makeText(getBaseContext(), "You have Clicked Radio
"+Integer.toString(destination), Toast.LENGTH_SHORT).show();
                                }
                        });
                        funcCollegelayout();
        }
        public void funcCollegelayout(){
                btn_ok.setOnClickListener(new OnClickListener() {
                        @Override
                        public void onClick(View v) {
                tmp1="";
                tmp="";
                FileReader fin1 = null;
                                try {
                 fin1 = new FileReader("/mnt/sdcard/qrnavigator/file1.txt");
                                } catch (FileNotFoundException e) {
                                        e.printStackTrace();
                                }
                BufferedReader graphFile = new BufferedReader(fin1);
```

```
                        try {
                                while((tmp = graphFile.readLine())!=null)
                                {
                                        tmp1+=tmp;
                                }
                        } catch (IOException e) {
                              e.printStackTrace();
                        }
                        filecnt=tmp1;
                        nof = Integer.parseInt(tmp1.substring(0, 2));
                        Dijkstra d=new Dijkstra();
                        int ip1 = Integer.parseInt(ma.source);
                        int ip2 = destination;
                        output = d.newMain(ip1,ip2);
                           FileReader fin2 = null;
        tmp1="";
        tmp="";
                        try {
                        fin2 = new FileReader("/mnt/sdcard/qrnavigator/file2.txt");
                        } catch (FileNotFoundException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }
        BufferedReader pixelFile = new BufferedReader(fin2);
                        try {
                                while((tmp = pixelFile.readLine())!=null)
                                {
                                        tmp1+=tmp;
                                }
                        } catch (IOException e) {
```

```
                    e.printStackTrace();
                }
            pixel = tmp1
            for(i=0;output.charAt(i)!='[';i++);
                dist=output.substring(0, i-1);
                    for(k=0;k<100;k++)
                        path[k]="";
                   for(++i,p=0;output.charAt(i)!=']';i++)
                {
                    node+=output.charAt(i);
                if(output.charAt(i+1)==','|output.charAt(i+1)==']')
                {
                    nodelist[j]=node;
                    j++;
                    if(cflr==-1)
                        cflr=Integer.parseInt(output.substring(i-1, i+1));
                    if(flr != Integer.parseInt(output.substring(i-1, i+1))&cflr!=-1)
                    {
                        k = pixel.lastIndexOf(node);
                        path[flr]+="ctx.lineTo("+pixel.substring(k-8, k-1)+");\n";
                        flr = Integer.parseInt(output.substring(i-1, i+1));
                    }
                    if(path[flr]=="")
                    {
                        k = pixel.lastIndexOf(node);
                    path[flr]+="ctx.arc("+pixel.substring(k-8, k-1)+",10,0,7);\n";
                    path[flr]+="ctx.moveTo("+pixel.substring(k-8, k-1)+");\n";
                        x[p]=Integer.parseInt(pixel.substring(k-8, k-5));
                        y[p]=Integer.parseInt(pixel.substring(k-4, k-1));
                        p++;
```

```
                    node="";
                }
                else
                {
                k = pixel.lastIndexOf(node);
                path[flr]+="ctx.lineTo("+pixel.substring(k-8, k-1)+");\n";
                x[p]=Integer.parseInt(pixel.substring(k-8, k-5));
                y[p]=Integer.parseInt(pixel.substring(k-4, k-1));
                p++;
                node="";
                }
                if(output.charAt(i+1)==',')
                    i+=2;
            }
            }
            pathlist="";
            String prevs="",current="";
            for(i=2;i<p;i++)
            {
            ang1=Math.toDegrees(Math.atan2((y[i-1]-y[i-2]),(x[i-1]-x[i-2])));
            ang2=Math.toDegrees(Math.atan2((y[i]-y[i-1]),(x[i]-x[i-1])));
                if(ang1==ang2)
                    current="go straight...\n";
                else if(ang1==0)
                {
                    if(ang2<0&&ang2>-180)
                        current="turn left...\n";
                    else
                        current="turn right...\n";
                }
```

```
else if(ang1>0&&ang2>0)
{
if(ang1<ang2)
        current="turn right...\n";
else if(ang1>ang2)
        current="turn left..\n";
else
        current="go straightbalbal...\n";
}
else if(ang1<0&&ang2<0)
{
        if(ang1>ang2)
                current="turn right...\n";
        else if(ang1<ang2)
                current="turn left..\n";
        else
                current="go straighhfghnt...\n";
}
else if(ang1<0&&ang1>=-180)
{
        if(ang2>=0&&ang2<=90)
                current="turn right...\n";
        else
                current="turn left...\n";
}
else if(ang2<0&&ang2>=-180)
{
        if(ang1>=0&&ang1<=90)
                current="turn left...\n";
        else
```

```
                                        current="turn right...\n";
                        }
                        else if(ang2==0)
                        {
                                if(ang1>0&&ang1<180)
                                        current="turn left...\n";
                                else
                                        current="turn right...\n";
                        }
                        if(prevs=="go straight...\n"&&current=="go straight...\n")
                        {
                        }
                        else
                        {
                                prevs=current;
                                pathlist+=current;
                        }
                    }
                Intent intent=new Intent(getApplicationContext(), Draw.class);
                        startActivity(intent);
                    }
            });
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
            getMenuInflater().inflate(R.menu.activity_destination, menu);
            return true;
    }
}
```

10.3 College Layout Activity

```
package com.example.newqrreader;
import android.os.Bundle;
import android.os.Environment;
import android.app.Activity;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Bitmap;
import android.graphics.Matrix;
import android.graphics.PointF;
import android.util.FloatMath;
import android.util.Log;
import android.view.Menu;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ImageView;
import android.widget.ZoomControls;

public class CollegeLayoutActivity extends Activity {
        final String TAG="YOUR_TAG_NAME";
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_college_layout);
                TouchImageView obj=new TouchImageView(this);
                obj.setImageResource(R.drawable.full);
                obj.setMaxZoom(4f);
```

```
                setContentView(obj);
        }
        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
                // Inflate the menu; this adds items to the action bar if it is present.
                getMenuInflater().inflate(R.menu.activity_college_layout, menu);
                return true;
        }
}
```

10.4 Clipboard Manager

```
package com.example.newqrreader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import android.annotation.SuppressLint;
import android.content.ClipData;
import android.content.ClipboardManager;
import android.content.ContentResolver;
import android.content.Context;
import android.content.Intent;
import android.content.res.AssetFileDescriptor;
import android.net.Uri;
import android.util.Log;

public class MyClipboardManager {
        int message;
   @SuppressLint("NewApi")
```

```
@SuppressWarnings("deprecation")
public boolean copyToClipboard(Context context, String text) {
    try {
        int sdk = android.os.Build.VERSION.SDK_INT;
        if (sdk < android.os.Build.VERSION_CODES.HONEYCOMB) {
        android.text.ClipboardManager clipboard = (android.text.ClipboardManager) context
                .getSystemService(context.CLIPBOARD_SERVICE);
            clipboard.setText(text);
        } else {
            android.content.ClipboardManager clipboard = (android.content.ClipboardManager) context
                .getSystemService(context.CLIPBOARD_SERVICE);
            android.content.ClipData clip = android.content.ClipData
                .newPlainText(
                    context.getResources().getString(message), text);
            clipboard.setPrimaryClip(clip);
        }
        return true;
    } catch (Exception e) {
        return false;
    }
}

@SuppressLint("NewApi")
public String readFromClipboard(Context context) {
    int sdk = android.os.Build.VERSION.SDK_INT;
    if (sdk < android.os.Build.VERSION_CODES.HONEYCOMB) {
        android.text.ClipboardManager clipboard = (android.text.ClipboardManager) context
            .getSystemService(context.CLIPBOARD_SERVICE);
        return clipboard.getText().toString();
```

```
    } else {
        ClipboardManager clipboard = (ClipboardManager) context
                .getSystemService(Context.CLIPBOARD_SERVICE);
        ContentResolver cr = context.getContentResolver();
        ClipData clip = clipboard.getPrimaryClip();
        if (clip != null) {
            String text = null;
            String title = null;
            ClipData.Item item = clip.getItemAt(0);
            Uri uri = item.getUri();
            if (text == null) {
                text = coerceToText(context, item).toString();
            }
            return text;
        }
    }
    return "";
}
@SuppressLint("NewApi")
public CharSequence coerceToText(Context context, ClipData.Item item) {
    CharSequence text = item.getText();
    if (text != null) {
        return text;
    }
    Uri uri = item.getUri();
    if (uri != null) {
      FileInputStream stream = null;
        try {
                    AssetFileDescriptor descr = context.getContentResolver()
                .openTypedAssetFileDescriptor(uri, "text/*", null);
```

```
            stream = descr.createInputStream();
            InputStreamReader reader = new InputStreamReader(stream,
                  "UTF-8");
            StringBuilder builder = new StringBuilder(128);
            char[] buffer = new char[8192];
            int len;
            while ((len = reader.read(buffer)) > 0) {
               builder.append(buffer, 0, len);
            }
            return builder.toString();
         } catch (FileNotFoundException e) {
         } catch (IOException e) {
            // Something bad has happened.
            Log.w("ClippedData", "Failure loading text", e);
            return e.toString();
         } finally {
            if (stream != null) {
               try {
                  stream.close();
               } catch (IOException e) {
               }
            }
         }
      return uri.toString();
      }
      Intent intent = item.getIntent();
      if (intent != null) {
         return intent.toUri(Intent.URI_INTENT_SCHEME);
      }
      return "";
```

```
   }
}
```

10.5 Dijekstra

```
package com.example.newqrreader;
import java.io.*;
import java.util.PriorityQueue;
import java.util.List;
import java.util.ArrayList;
import java.util.Collections;
import android.provider.SyncStateContract.Constants;

class Vertex implements Comparable<Vertex>
{
   public final String name;
   public Edge[] adjacencies;
   public double minDistance = Double.POSITIVE_INFINITY;
   public Vertex previous;
   public Vertex(String argName) { name = argName; }
   public String toString() { return name; }
   public int compareTo(Vertex other)
   {
      return Double.compare(minDistance, other.minDistance);
   }
}
class Edge
{
   public final Vertex target;
```

```
   public final double weight;
   public Edge(Vertex argTarget, double argWeight)
   { target = argTarget; weight = argWeight; }
}
public class Dijkstra
{
   public static void computePaths(Vertex source)
   {
      source.minDistance = 0.;
      PriorityQueue<Vertex> vertexQueue = new PriorityQueue<Vertex>();
          vertexQueue.add(source);
          while (!vertexQueue.isEmpty()) {
              Vertex u = vertexQueue.poll();
         for (Edge e : u.adjacencies)
         {
            Vertex v = e.target;
            double weight = e.weight;
            double distanceThroughU = u.minDistance + weight;
                    if (distanceThroughU < v.minDistance) {
                       vertexQueue.remove(v);
                       v.minDistance = distanceThroughU ;
                       v.previous = u;
                       vertexQueue.add(v);
                    }
         }
      }
   }
   public static List<Vertex> getShortestPathTo(Vertex target)
   {
      List<Vertex> path = new ArrayList<Vertex>();
```

```
    for (Vertex vertex = target; vertex != null; vertex = vertex.previous)
        path.add(vertex);
    Collections.reverse(path);
    return path;
  }
  public String newMain(int inp1, int inp2)
  {
        int max=100;
        DestinationActivity m = new DestinationActivity();
        String files = m.filecnt;
        String tmp1="",str="";
        int i=3,j=0,k=0;
        int[] to = new int[100];
        int[] dist = new int[100];
    Vertex[] ver = new Vertex[100];
 for(i=3;files.charAt(i)!='[';i++)
    {
        if(files.charAt(i)=='_')
        {
          ver[j] = new Vertex(tmp1);
          tmp1="";
          j++;
        }
        else
        tmp1+=files.charAt(i);
    }
    for(;j<max;j++)
    {
       ver[j] = new Vertex("nil");
    }
```

```
j=0;k=0;
for(++i;files.charAt(i)!=']';i++)
{
if(files.charAt(i)=='_')
{
          to[j]=Integer.parseInt(tmp1);
  tmp1="";
}
else if(files.charAt(i)=='+')
{
          dist[j]=Integer.parseInt(tmp1);
  tmp1="";
  j++;
}
else if(files.charAt(i)=='%')
{
          dist[j]=Integer.parseInt(tmp1);
  tmp1="";
  j++;
ver[k].adjacencies = new Edge[]{ new Edge(ver[to[0]], dist[0]),
        new Edge(ver[to[1]], dist[1]),
      new Edge(ver[to[2]], dist[2]), new Edge(ver[to[3]], dist[3]) };
k++;
j=0;
}
else
tmp1+=files.charAt(i);
}
for(;k<max;k++)
{
```

```
        ver[k].adjacencies = new Edge[]{ new Edge(ver[99], 9999),
              new Edge(ver[99], 9999),
             new Edge(ver[99], 9999), new Edge(ver[99], 9999) };
        }
        Vertex[] vertices = new Vertex[100];
        for(i=0;i<max;i++)
                vertices[i]=ver[i];
    computePaths(ver[inp1]);
    i=0;
    for (Vertex v : vertices)
        {
           System.out.println("Distance to " + v + ": " + v.minDistance);
           List<Vertex> path = getShortestPathTo(v);
           System.out.println("Path: " + path);
           if(i==inp2)
           {
           str+=v.minDistance;
           str+=path;
           }
           i++;
        }
       return str;
  }
}
```

Direction list

```
package com.example.newqrreader;
import android.os.Bundle;
import android.app.Activity;
```

```
import android.view.Menu;
import android.widget.Button;
import android.widget.TextView;

public class DirectionsList extends Activity {
	DestinationActivity dd = new DestinationActivity();
	TextView txt;
	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_directions_list);
		txt=(TextView)findViewById(R.id.drctnlist);
		txt.setText(dd.pathlist);
	}
	@Override
	public boolean onCreateOptionsMenu(Menu menu) {
	getMenuInflater().inflate(R.menu.activity_directions_list, menu);
		return true;
	}
}
```

Draw

```
package com.example.newqrreader;
import android.R.integer;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.test.suitebuilder.annotation.MediumTest;
```

```
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.webkit.WebView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class Draw extends Activity {
	String Html="",html_1="",html_2="",html_3="",dist="";
	String[] path= new String[10];
	int flr=0,nof=0;
	Button up_btn,down_btn,dirlist;
	WebView wb;
	TextView dist_display;
	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_draw);
		up_btn=(Button)findViewById(R.id.up);
		down_btn=(Button)findViewById(R.id.down);
		dirlist=(Button)findViewById(R.id.dirlist);
		wb=(WebView)findViewById(R.id.webView1);
		dist_display=(TextView)findViewById(R.id.distance_display);
		DestinationActivity mm = new DestinationActivity();
		path=mm.path;
		nof=mm.nof;
		dist=mm.dist;
		flr=mm.cflr;
```

```
dist_display.setText("distance to "+mm.destinations[mm.pos]+" is : "+dist+" meters.");
final String fileName = "file:///mnt/sdcard/floor";
down_btn.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View arg0) {
if(flr>0)
{
--flr;
Html= html_1+path[flr]+html_2+Integer.toString(flr)+html_3;
wb.loadDataWithBaseURL("", Html, "text/html", "UTF-8", Html);
}
}
});
up_btn.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View arg0) {
if(flr<nof)
{
++flr;
Html= html_1+path[flr]+html_2+Integer.toString(flr)+html_3;
wb.loadDataWithBaseURL("", Html, "text/html", "UTF-8", Html);
}
}
});
dirlist.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View arg0) {
// TODO Auto-generated method stub
```

```
                    Intent intent=new Intent(getApplicationContext(),
DirectionsList.class);
                        startActivity(intent);
                    }
                });



        //moveUp();
        //moveDown();




        html_1="<html><body>"
                    + "<canvas id="+'"'+"myCanvas"+'"'+" width="+'"'+"1000"+'"'+"
height="+'"'+"1000"+'"'+" ></canvas>"
                    +"<script>"
                    + " var canvas = document.getElementById('myCanvas');"
                    + "var ctx = canvas.getContext('2d');"
                    + "var imageObj = new Image();"
                    + " imageObj.onload = function() {"
                    + "ctx.drawImage(imageObj, 0, 0);"
                    + "ctx.beginPath();";



        html_2="ctx.lineWidth = 7;"
                    + "ctx.strokeStyle = 'blue';"
                    + "ctx.stroke();"
                    + "};"
                    + "imageObj.src = 'file:///mnt/sdcard/qrnavigator/floor";
        html_3=".jpg';"
```

```
                + "</script>"
                + "</body>"
                + "</html> ";

        Html= html_1+path[mm.cflr]+html_2+Integer.toString(mm.cflr)+html_3;

        wb.getSettings().setBuiltInZoomControls(true);
        //wb.getSettings().setDisplayZoomControls(false);
    wb.getSettings().setSupportZoom(true);
    wb.getSettings().setJavaScriptEnabled(true);
    wb.getSettings().setLoadWithOverviewMode(true);
    wb.getSettings().setUseWideViewPort(true);
    wb.getSettings().setAppCacheEnabled(true);
    //vw.getSettings().setPluginState(PluginState.ON);
        wb.getSettings().setAllowFileAccess(true);

        wb.loadDataWithBaseURL("", Html, "text/html", "UTF-8", Html);



    }



    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.activity_draw, menu);
```

```
                return true;
        }

}
```

10.6My Location

```
package com.example.newqrreader;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.webkit.WebView;
import android.widget.Button;
import android.widget.TextView;

    public class MyLocation extends Activity {
        String Html="",html_1="",html_2="",html_3="",dist="";
        String[] path= new String[10];
        int flr=0,nof=0,i=0;
        Button up_btn,down_btn,dirlist;
        WebView wb;
        TextView dist_display;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_my_location);
                up_btn=(Button)findViewById(R.id.up);
                down_btn=(Button)findViewById(R.id.down);
```

```
wb=(WebView)findViewById(R.id.webView1);
dist_display=(TextView)findViewById(R.id.distance_display);
for(i=0;i<10;i++)
	path[i]="";
MainActivity ma1=new MainActivity();
path[Integer.parseInt(ma1.myflr)]="ctx.moveTo("+ma1.mypix+");
ctx.arc("+ma1.mypix+",10,0,7);";
nof=Integer.parseInt(ma1.mynof);
flr=Integer.parseInt(ma1.myflr);
dist_display.setText("you are here");
final String fileName = "file:///mnt/sdcard/floor";
	down_btn.setOnClickListener(new OnClickListener() {
		@Override
		public void onClick(View arg0) {
			if(flr>0)
			{
				--flr;
				Html=
html_1+path[flr]+html_2+Integer.toString(flr)+html_3;

    wb.loadDataWithBaseURL("", Html, "text/html", "UTF-8", Html);
			}
		}
	});
	up_btn.setOnClickListener(new OnClickListener() {
		@Override
		public void onClick(View arg0) {
			// TODO Auto-generated method stub
			if(flr<nof)
			{
				++flr;
```

```
                                        Html=
html_1+path[flr]+html_2+Integer.toString(flr)+html_3;
    wb.loadDataWithBaseURL("", Html, "text/html", "UTF-8", Html);
                                    }
                                }
                            });
                html_1="<html><body>"
                                + "<canvas id="+""+"myCanvas"+""+"
width="+""+"1000"+""+" height="+""+"1000"+""+" ></canvas>"
                                +"<script>"
                                + " var canvas = document.getElementById('myCanvas');"
                                + "var ctx = canvas.getContext('2d');"
                                + "var imageObj = new Image();"
                                + " imageObj.onload = function() {"
                                + "ctx.drawImage(imageObj, 0, 0);"
                                + "ctx.beginPath();";
                html_2="ctx.lineWidth = 7;"
                                + "ctx.strokeStyle = 'blue';"
                                + "ctx.stroke();"
                                + "};"
                                + "imageObj.src = 'file:///mnt/sdcard/qrnavigator/floor";
                html_3=".jpg';"
                                + "</script>"
                                + "</body>"
                                + "</html> ";
                Html= html_1+path[flr]+html_2+Integer.toString(flr)+html_3;
                wb.getSettings().setBuiltInZoomControls(true);
                //wb.getSettings().setDisplayZoomControls(false);
        wb.getSettings().setSupportZoom(true);
        wb.getSettings().setJavaScriptEnabled(true);
        wb.getSettings().setLoadWithOverviewMode(true);
        wb.getSettings().setUseWideViewPort(true);
```

```
 wb.getSettings().setAppCacheEnabled(true);
 //vw.getSettings().setPluginState(PluginState.ON);
        wb.getSettings().setAllowFileAccess(true);
        wb.loadDataWithBaseURL("", Html, "text/html", "UTF-8", Html);
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_my_location, menu);
        return true;
}
                                                }
```

11. CONCLUSION

QR codes could be the cheapest and easiest positioning method for an indoor navigation system. Main advantage of this approach is that it is cost effective for the service provider. Users/visitors do not have to make any investment for this indoor navigation. Complexity and time to implement is less for this approach. There are no additional configurations which the users have to maintain for indoor navigation. Access to navigation system even for moderately equipped phones with Satellite communication is avoided (which is not possible in closed environments) for getting the location information.

12. REFERENCES

[1]. Rahul Raj C P, Seshu Babu Tolety, Catheine Immaculate, "QR code based navigation system for closed building using smart phones", Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), pp. 641 – 644, 2013

[2] Christian LUKIANTO and Harald STERNBERG. 2011. Overview of Current Indoor Navigation Techniques and Implementation Studies .FIG Working week, Germany

[3] Yue Liu. 2008. Recognition of QR code with mobile hones.University

[4] Anwar A K. 2009. Indoor location tracking using AGPS. University of Jinan, Hannover.

[5]www.stackoverflow.com

[6]www.codeforprojects.com

[7]www.w3schools.com