DECLARATION

We hereby declare that the project work entitled - **Smart System for the blind** is an authentic record of our work carried out as requirements for final year project for the award of B.tech degree in Computer Science Engineering at Symbiosis Institute of Technology Pune, affiliated to Symbiosis International University Pune, under the guidance of Mrs. Poorva Agarwal, during January 2016 to May 2016.

G. Sai Rohan Reddy	**Harshad Shinde**	**Supriya Shinde**	**Tushar Mane**
12070121115	13070121254	13070121255	13070121257

Date: _____

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

Mrs. Poorva Agarwal
Assistant Professor
CS and IT Department
SIT, Pune

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the completion of our project work. Whatever we have done is only due to such guidance and assistance and we would not forget to thank them.

First and foremost, we are expressing our thankfulness and praise to Symbiosis Institute of Technology, Pune and Department of Computer Science for giving us this wonderful opportunity to undergo B.Tech Project Work, helping us to learn and get a great experience.

We would like to thank our Prof. Poorva Agarwal for her valuable guidance, supervising this work and helpful suggestions.

We owe our profound gratitude and special thanks to Prof. Rahul Joshi who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep us on the correct path by his untiring assistance, direction, encouragement, comments, suggestions, continuous guidance, support, ideas and constructive criticism throughout this project work.

We heartily thank our project coordinator, Prof. Kalyani Kadam for cooperating with us and giving us her valuable time and information.

At last we are grateful to our respected teachers Department of Computer Science SIT, Pune, family and friends for their help, encouragement and co-operation during the project work.

G. Sai Rohan Reddy

Harshad Shinde

Supriya Shinde

Tushar Mane

ABSTRACT

"Smart System for the Blind"

In recent years, blind people have been a lot of facing in directing and guiding themselves for walking.

They have not been able to walk properly onto public without proper guidance or with the help a person. This makes them dependent on someone else.

We have thought something interesting on this issue and bought up a solution to this problem. We would be planning to use embedded systems, sensors, a smart phone and few other hardware devices to make an integrated system to help and guide the blind in directions and obstacle detection.

We will be needing the smart phone for supporting an application that we would be making and a pair of shoes where we will be fitting in 2 sensors, an Arduino Nano micro controller, breadboard and a Bluetooth HC-05 module.

Keywords - Arduino, embedded system, bluetooth HC-05 module

LIST OF FIGURES

1. LITERATURE SURVEY

Research phase is very crucial for the success of any project. The capabilities and strengths of a project depend on how strong the research is. We devoted 40% of our time towards research on various sensors that can be used, how the arduino macro works and how communication can be done via Bluetooth module.

- **Optimum Energy Management System**

  This system is used as a way to manage the consumption of energy in an optimal way. It defines an easy way to save energy by controlling the switching of loads in a room only on the basis of the number of persons entering the room. The project uses IR sensors to sense the persons entering and leaving the room and accordingly the control unit controls the switching of the load.

- **Distance Measurement by ultrasonic sensor**

  Ultrasonic sensor can be used to measure distance of any object from a certain position. The sensor emits ultrasonic waves which are reflected by the object. The time taken by the waves to travel back and forth is calculated and multiplied with velocity of sound to get the distance measurement.

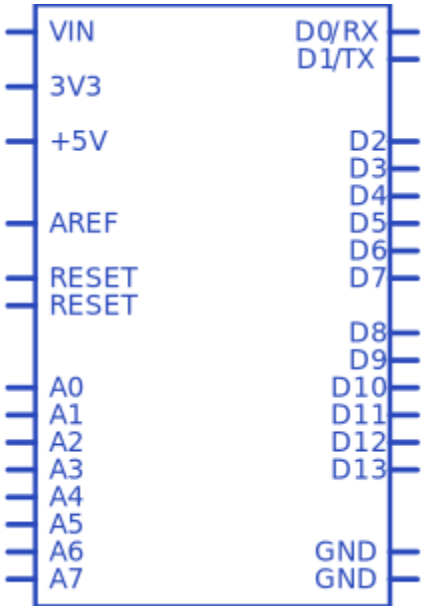
- **Practical Use of Arduino Micro Controller**

  We have made quite some observations on how arduino works in many other embedded system projects including sensors. How the data is integrated to this board and how communication is done with the Arduino. The use of digital and analog ports in it and what type of output is generated from them.

2. INTRODUCTION

2.1. OVERVIEW & PROBLEM DEFINITION

Embedded Systems have become a common usage now a days in day to day applications. Use of technology is vast in recent times. From whatever tools and knowledge we had, making a smart system that operates for the blind users was a possibility. A blind person on his own self cannot walk conveniently or know which direction to head to. It has become too much problematic for him/her to perform regular activities like transporting and walking.



WHY ARDUINO NANO?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects. Nano is one of the micro controllers designed by Arduino which serves the sole purpose of fitting into our project mainframe. Nano is easy to attach physically to the shoes which we will be using for our project.

2.2. PROBLEM STATEMENT

"We aim to design an integrated system for the users (blind) so that it helps them in guiding for directions i.e. movement and direction from one location to another. At the same time, the system will also be helping them in detecting any person or objects nearby so that the user can be alerted before collision."

2.3. CONTRIBUTION

This project aims to support a user in directions and guide them for obstacle detections. This can be really used practically, and it can help a lot of people who are in need. We can even start setting up this system by putting the idea up at a blind school or any foundation for users.

It will not only make the user self-reliable and independent, but also aim to make them more confident about travelling and going from place to place.

2.4. FEATURES OF THE PROJECT

- Obstacle Detection
- Map guiding
- Speech to Text for taking input
- Text to Speech for giving an output

3. PLATFORM / TECHNOLOGY USED

- Arduino IDE:

- JDK

- SDK

- Android Studio

- Google maps API

4. SRS (SOFTWARE REQUIREMENT SPECIFCATION)

4.1 Overview

This project on a whole is a system which does multiple tasks using the inter-communication between various components and also has a real time working. It does not fail to detect any object that passes by and also helps in navigation of the user. The wirings, sensors, boards and chips will be soldered within the shoes.

4.2 External Interface Requirements

4.2.1 User Interfaces

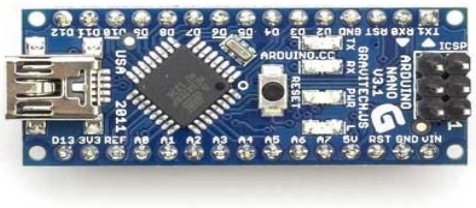
There can practically be no interface that supports for a user who cannot see, therefore we will be assigning any one of the buttons of the phone dedicated to start the app. Whenever the app starts, the user can give his speech and further processing is done.

4.2.2 Hardware Interfaces

- **Arduino Nano**

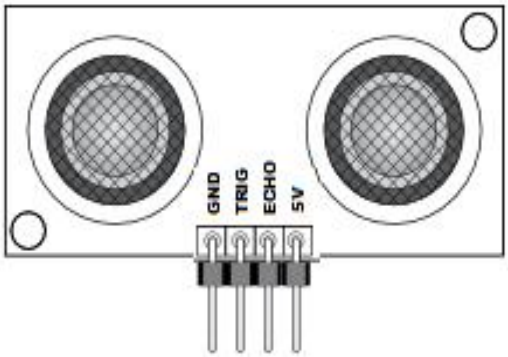
  Arduino is common term for a software company, project, and user community, that designs and manufactures computer open-source hardware, open-source software, and microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices.

  The project is based on microcontroller board designs, produced by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can interface to various expansion boards (termed shields) and other circuits. The boards feature serial communication interfaces, including Universal Serial Bus (USB) on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) based on a programming language named Processing



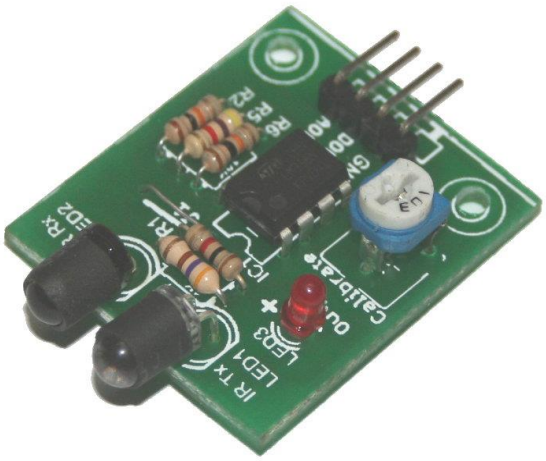
- **Ultrasonic/Ping sensor**

Ultrasonic transducers are transducers that convert ultrasound waves to electrical signals or vice versa. Those that both transmit and receive may also be called ultrasound transceivers; many ultrasound sensors besides being sensors are indeed transceivers because they can both sense and transmit. These devices work on a principle similar to that of transducers used in radar and sonar systems, which evaluate attributes of a target by interpreting the echoes from radio or sound waves, respectively. Active ultrasonic sensors generate high-frequency sound waves and evaluate the echo which is received back by the sensor, measuring the time interval between sending the signal and receiving the echo to determine the distance to an object. Passive ultrasonic sensors are basically microphones that detect ultrasonic noise that is present under certain conditions, convert it to an electrical signal, and report it to a computer/device. This would be placed ahead in one of the shoes.



- **Obstacle sensor**

  Use and function of this is similar to that of Ultrasonic sensor. The difference is that ultrasonic sensor by default transmits digital signals i.e. only in the formal of 0 & 1. Where as an obstacle sensor gives a analog output, it gives the exact distance of the object lying nearby. Two of them are used on both of shoes.



- **Battery**

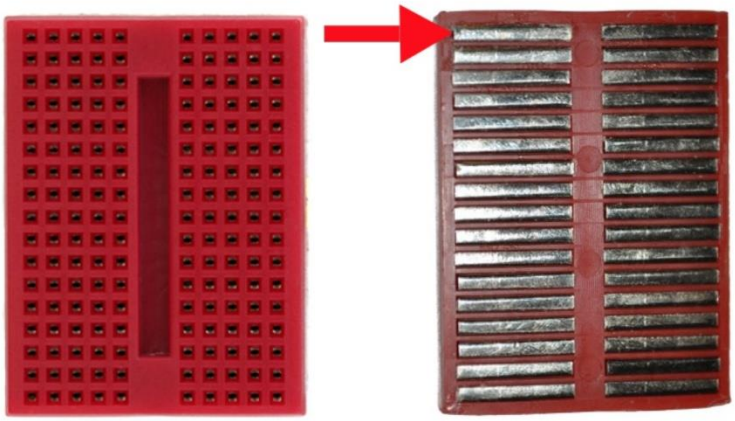
  We would be using a 5 volts, 1 amp battery to suffice for our electricity purposes. This battery is replaceable and also rechargeable.

- **Wires**

  Male to male, Male to female, female to female wirings are used to connect the components.

- **Breadboard**

  A breadboard is a construction base for prototyping of electronics. It is used to divide/integrate a signal of the system.



4.2.3 Software Interfaces

- **Arduino IDE**

  The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and provides simple one-click mechanism to compile and load programs to an Arduino board.

  The Arduino IDE supports the languages C and C++ using special rules to organize code. The Arduino IDE supplies a software library called Wiring from the Wiring project, which provides many common input and output procedures. A typical Arduino C/C++ sketch consist of two functions that are compiled and linked with a program stub main() into an executable cyclic executive program:

setup(): a function that runs once at the start of a program and that can initialize settings.

loop(): a function called repeatedly until the board powers off.

- **JDK**

  The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris ,Linux, Mac OS X or Windows. The JDK includes a private JVM and a few other resources to finish the development of a Java Application. Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK).

- **SDK**

  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. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows XP or later. As of March 2015, the SDK is not available on Android itself, but the software development is possible by using specialized Android applications.

- **Android Studio**

  Android Studio is the official integrated development environment (IDE) for developing for the Android platform.

- **Google maps API**

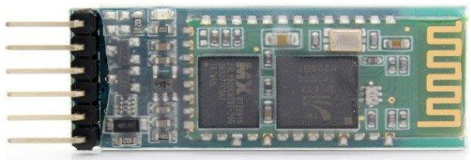
  Google Maps is a desktop web mapping service developed by Google. It offers satellite imagery, street maps, 360° panoramic views of streets (Street View), real-time traffic conditions (Google Traffic), and route planning for traveling by foot, car, bicycle (in beta), or public transportation.

4.2.4 Communication Interfaces

- **Bluetooth Module - HC05**

  HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

  Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.



- **Jumper Cables**

  This includes male to female, male to male and female to female wires for interconnection.

4.3 Assumptions and Dependencies

Assumptions

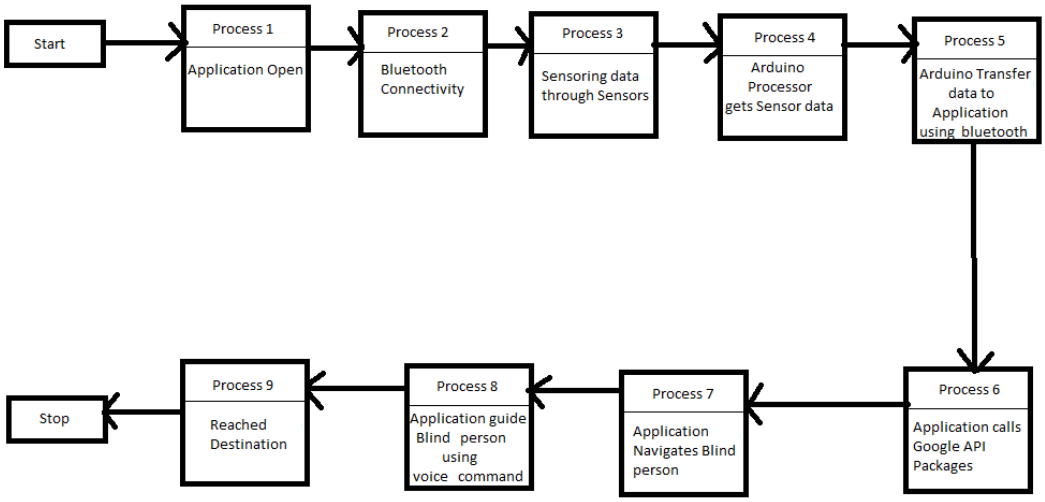
- We will be soldering the wires and the whole system into shoes, so the pretty clear assumption here is that there would not be any loose connection between any of the components or wires.
- We also are assuming that the text to speech and speech conversion is made properly and ideally no problems regarding those are faced.

Dependencies

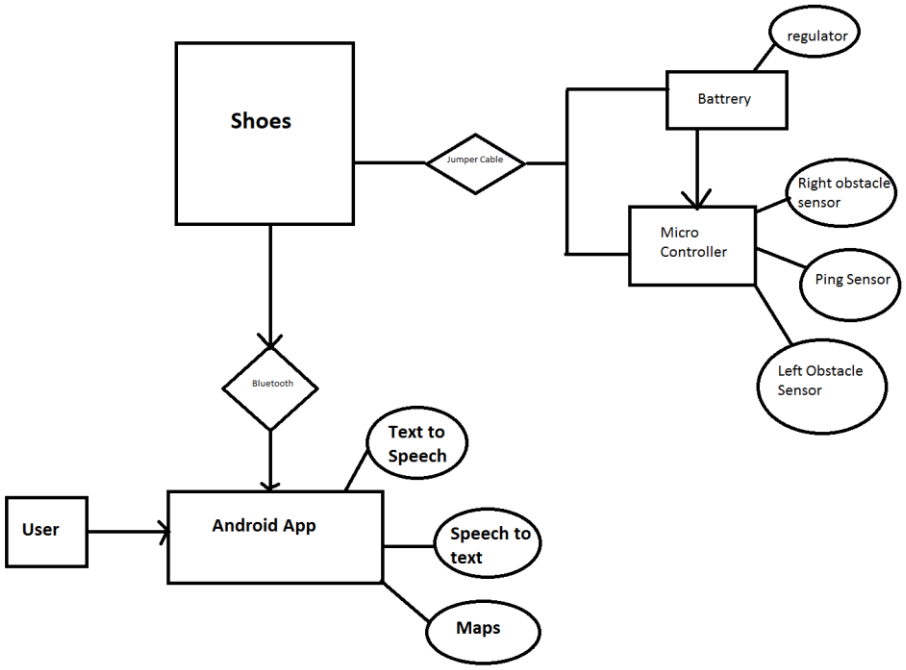
- The application's outcome solely depends on the sensors reading. Any mistake can result in a wrong detection or no detection of an obstacle.
- The Bluetooth module always has a lot of dependencies, in case there is a communication failure, there will be a problem with the system.

5. High Level Design

5.1 Data Flow Diagram

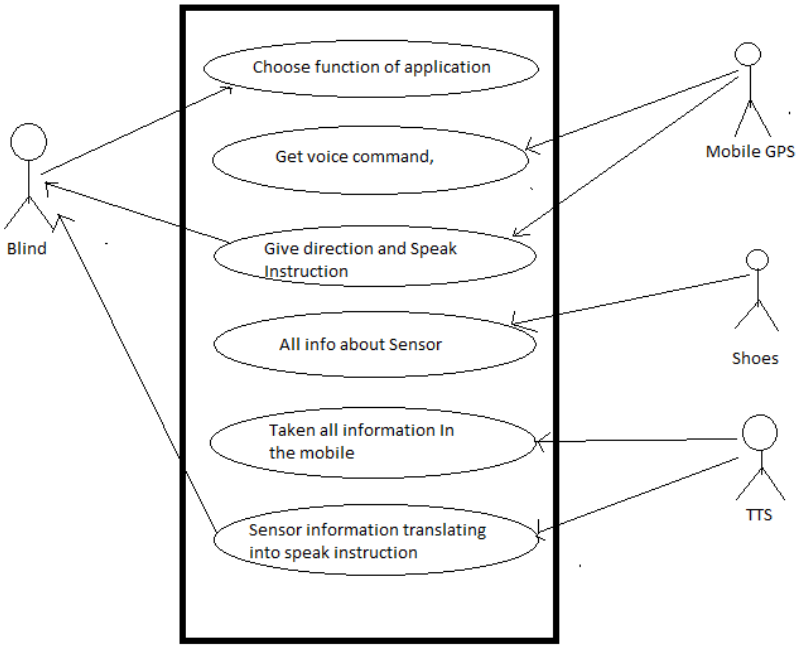


5.2 Entity Relationship Diagram

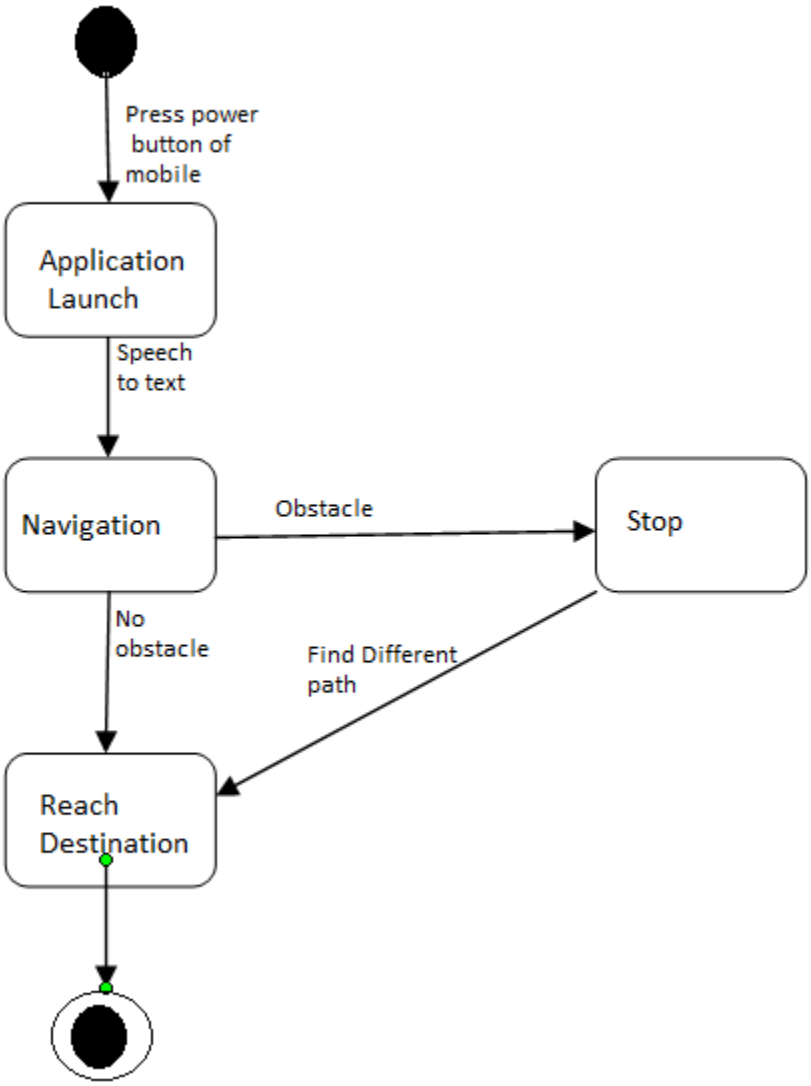


5.3 UML Diagrams

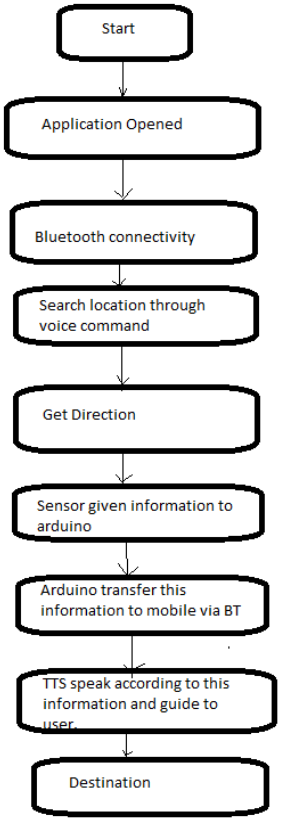
5.3.1 Use Case Diagram



5.3.2 State Transition Diagram

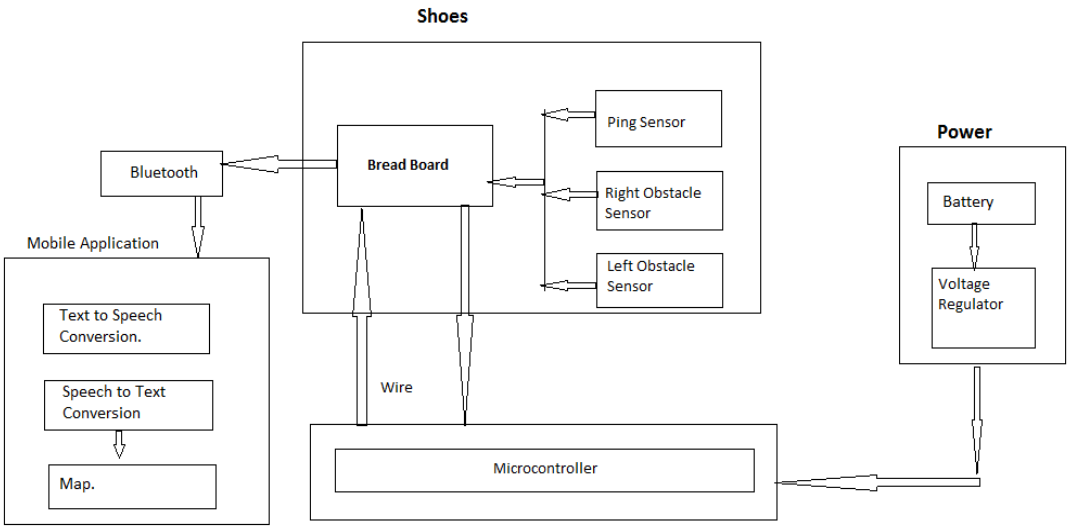


5.3.3 Activity Diagram



6. Project Plan

6.1 Block Diagram



6.2 RMMM PLAN

The goal of **the risk mitigation, monitoring and management plan** is to identify as many as potential risks as possible.

Risk Description

- ✓ **Project risks:** Identifies potential code, personnel, resource and requirement problems and their impact on the project. It threatens the project plan.

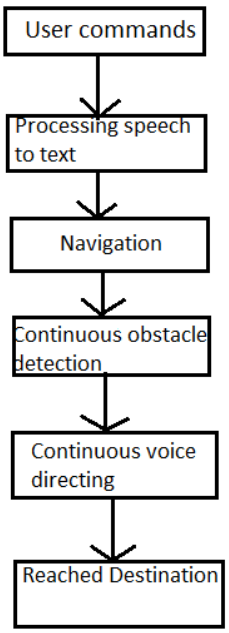
- ✓ **Technical risks:** Identifies potential design, implementation, interface, verification and maintenance problems. Technical risks threaten quality and timeliness of the software to be produced. If a technical risk becomes a reality, Implementation may become difficult or impossible.

- ✓ **Network risks:** This includes the network failure and other network related issues.

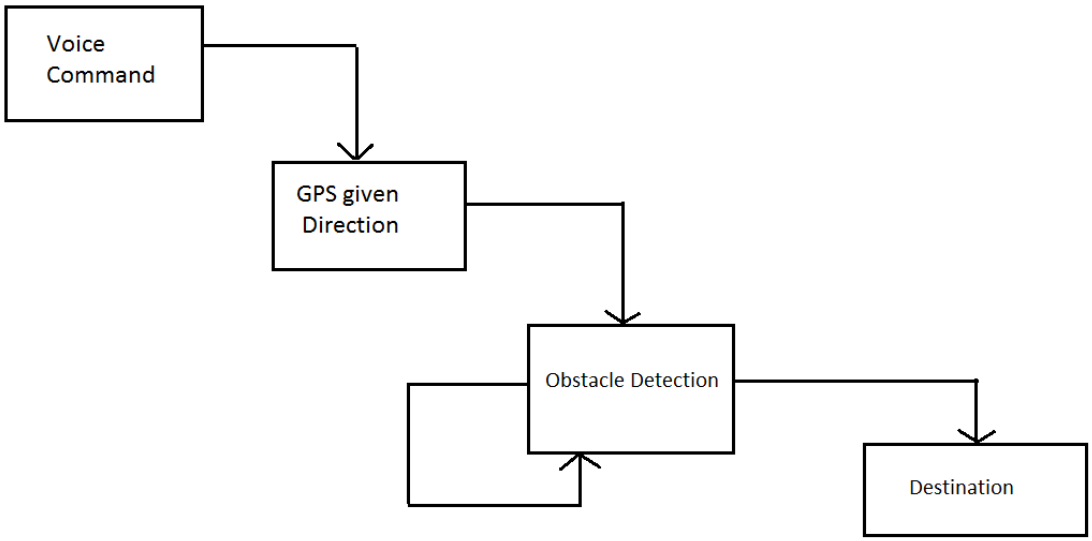
- ✓ **Support risk:** The degree of uncertainty that the resultant software will be easy to correct, adapt and enhance.

7. Implementation

7.1 System Model



7.2 Process Flow



7.3 Working

7.3.1 User communication with the system

The application gets initiated within a click any of the buttons of the smart phone. The app is ready to take input from the user. The user calls out for a statement saying "navigate to symbiosis institute of technology".

Next the application processes this speech and parses it, converts to text using STT(speech to text) and then makes sense of it. Then, it will launch the Google maps application with the source as a default "Your location" and destination as per the user's instruction.

7.3.2 Continuous Obstacle Detection

As the application keeps directing the user to the destination, we also should be detecting obstacles on the way. There are 2 sensors attached to the front and side of the shoe and it keeps checking for objects. This data is sent to the arduino nano micro controller which integrates it. Based on the inputs from the sensors, nano keeps sending signals continuously to the app via Bluetooth.

We would be receiving an exact analog signal (precise distance of object from the ping sensor) and digital output from the obstacle sensor. All sensors are connected to the digital pins of the arduino.

7.3.3 Application commands to the user

Whenever there is an obstacle detection signal from the arduino, the application leaves a command determining whether the object is on the left, right, ahead or on both sides.
This command is given using TTS (text to speech).

Meanwhile, Google maps keeps giving voice commands to the user regarding the direction to move and how far the destination is and when to take a turn.

7.4 Deployment & Soldering

Soldering is done to keep the jumper cables fixed and all the sensors, micro controller, bread board, Bluetooth module at place. Shoes are for a little rough usage, hence soldering is required to maintain shape.

The whole system is connected to a 5amp battery which is also attached to the shoe.

7.4.1 Suitable Range of Sensors

We need to adjust the sensor detection range in such a way that the range is neither too less as it would result in collision. Neither should it be too high as it will result in multiple detections even in the case of no collision. Hence, we should be selecting an optimum range for the sensors.

7.5 Software System Attributes

7.5.1 Reliability

We can totally tell that the whole system is reliable and works perfectly fine. The sensors are tested and they work on a real time basis. The reaction time and response time, procession speed, everything is quick enough before the user comes across an object. Exception handling is also done in the android code, where there are more chances of a bug.

7.5.2 Availability

The whole system is available to any person who has a smart phone and the app installed in it. The other components have to bought as one. The availability factor here would be mobile internet for the access of google maps. Otherwise, everything is pretty much available at all times. Most of the components are attached within the shoes.

7.5.3 Security

The whole system is completely secure. The arduino code is run upon the processor and it has a volatile memory. It does not get flushed until a new code is run over it. The code cannot be accessed by connection the microprocessor to another personal computer. Except for using internet for google maps, everything is done internally. So there is not much of a security threat.

7.5.4 Maintainability

Application code is cohesive and has a recognizable functionality. Each hardware part of the system is very much maintainable and also replaceable. There is no wear and tear of any of the components. There would only be a small issue of the battery. But once the battery is over, it can be recharged or even replaced with a new one. So again, maintainability is up to the mark. All the hardware components are of good quality and do not go down that easily.

7.5.5 Portability

The whole system/project is meant to be portable, it constitutes of a system which guides the user on the move. Libraries are platform independent. The arduino code supports C and C++ languages. It works on all android devies with bluetooth.

8. VALIDATION OF SOFTWARE

8.1. OVERVIEW

Testing is an investigation conducted about the quality of the product or service under test. It can be stated as the process of validating and verifying that a computer program/application/product:

- meets the requirements that guided its design and development,
- works as expected

Success of any GUI application depends on how it interacts with user through its user interface, how the user actions are performed to access application's features and whether application responds in functionally correct manner. An application with incorrect behavior or invalid user interaction can lead to huge problems.

GUI testing is a process to test application's user interface and to detect if application is functionally correct. GUI testing involves carrying set of tasks and comparing the result of same with the expected output and ability to repeat same set of tasks multiple times with different data input and same level of accuracy.

Implementing GUI testing for our application early in the software development cycle speeds up development improves quality and reduces risks. GUI Testing can be performed manually with a human tester.

8.2. TEST CASES

1. **Test Name:** Application Closure

   Test Description:

   - Launch your application
   - Navigate throughout the application, and then close the application through the button

   Result:

   - There wasn‘t any unexpected behavior during the closing process.

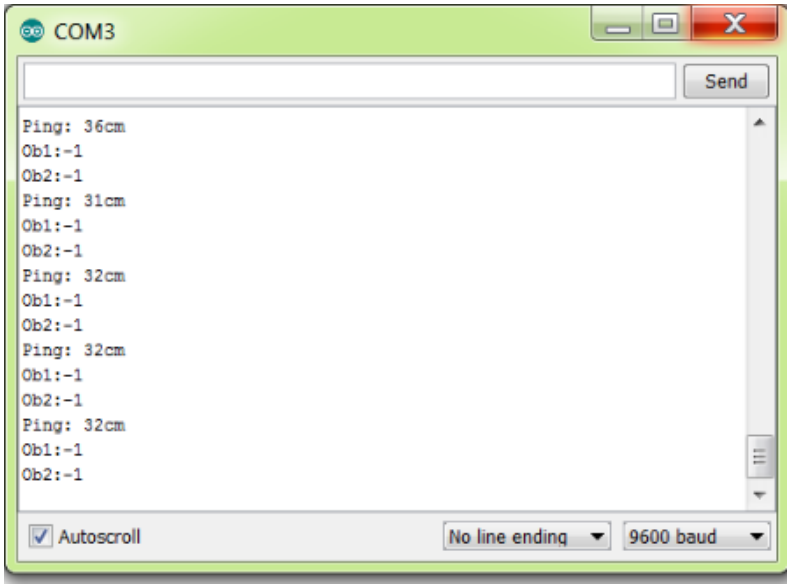
2. **Test Name:** Application Responsiveness

   Test Description:

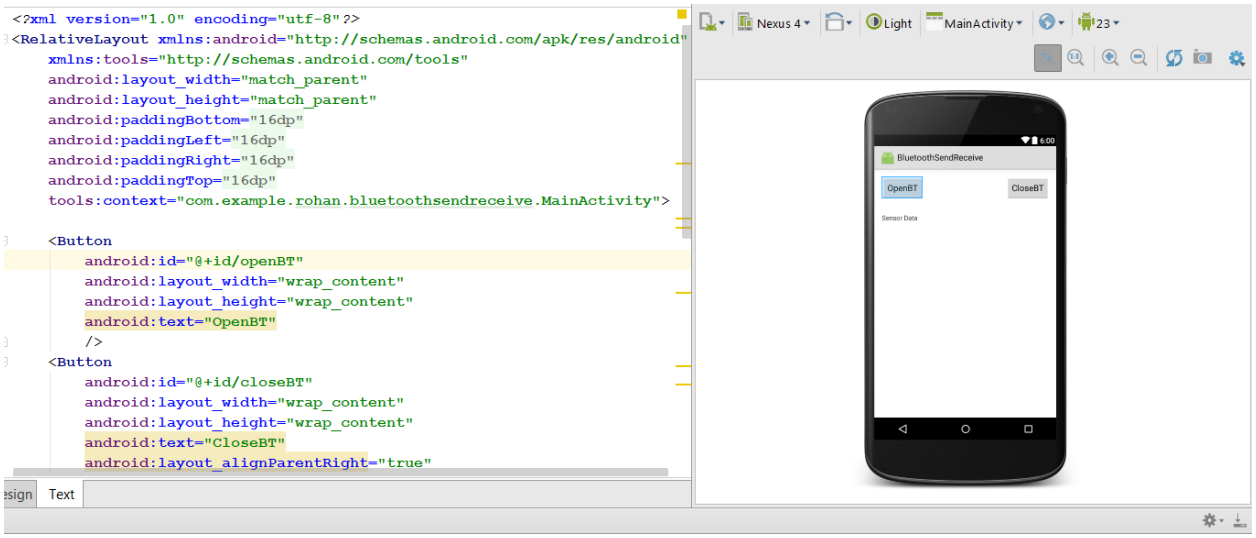
   - Launch your application
   - Thoroughly test the application features and functionality.
   - Verify that the application does not become unresponsive for more than three seconds.

9. Results & Analysis

9.1 Arduino IDE output



9.2 Android Studio screenshot



9.3 Sample Code

9.3.1 Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.rohan.bluetoothsendreceive">
    <uses-permission android:name="android.permission.BLUETOOTH" />

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

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

</manifest>
```

9.3.2 Main Activity

```
package com.example.rohan.bluetoothsendreceive;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.net.Uri;
import android.os.Handler;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.Locale;
import java.util.Set;
import java.util.UUID;
```

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener,TextToSpeech.OnInitListener  {

    Button  open,close;
    TextView out;
    EditText input;

    BluetoothAdapter ba;
    BluetoothDevice bd;
    BluetoothSocket bs;

    InputStream is;


    Thread workerThread;
    byte[] readBuffer;
    int readBufferPosition;

    volatile boolean stopWorker;

    String speechdata = "";
    TextToSpeech tts;



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

        out=(TextView)findViewById(R.id.data);
        open=(Button)findViewById(R.id.openBT);
        close=(Button)findViewById(R.id.closeBT);

        open.setOnClickListener(this);
        close.setOnClickListener(this);


        tts = new TextToSpeech(this, this);
    }


    @Override
    public void onClick(View v) {
        if(v.getId()==R.id.openBT)
        {
            try {
                findBT();
                openBT();
            }catch (Exception e){}
        }
        else if(v.getId()==R.id.closeBT)
        {
            try
            {
                closeBT();
            }catch (Exception e){}
```

```
        }
        else
        {
            try {
                String data = input.getText().toString();

            }catch (Exception e){}
        }
    }
    void findBT()throws Exception
    {

        ba=BluetoothAdapter.getDefaultAdapter();
        if(!ba.isEnabled())
        {
            Intent i1=new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivity(i1);

        }
        if(ba.isEnabled())
        {
            Set<BluetoothDevice> paired=ba.getBondedDevices();
            if(paired.size()>0)
            {
                for(BluetoothDevice dev:paired)
                {
                    if(dev.getName().equals("HC-05"))
                    {

                        bd=dev;

                        break;
                    }
                }
            }
        }



    }
    void openBT()throws Exception
    {
        UUID uuid= UUID.fromString("1101-0000-1000-8000-00805f9b34fb");
        bs=bd.createRfcommSocketToServiceRecord(uuid);
        bs.connect();

        is=bs.getInputStream();
        recieve();
        Toast.makeText(this, "Socket Opened", Toast.LENGTH_LONG).show();
    }



    void recieve() throws Exception
    {
        final Handler handler = new Handler();
        final byte delimiter = 32;

        stopWorker = false;
        readBufferPosition = 0;
```

```
        readBuffer = new byte[1024];
        workerThread = new Thread(new Runnable()
        {
            public void run()
            {
                while(!Thread.currentThread().isInterrupted() && !stopWorker)
                {
                    try
                    {
                        int bytesAvailable = is.available();
                        if(bytesAvailable > 0)
                        {
                            byte[] packetBytes = new byte[bytesAvailable];
                            is.read(packetBytes);
                            for(int i=0;i<bytesAvailable;i++)
                            {
                                byte b = packetBytes[i];
                                if(b == delimiter)
                                {
                                    byte[] encodedBytes = new
byte[readBufferPosition];
                                    System.arraycopy(readBuffer, 0, encodedBytes,
0, encodedBytes.length);
                                    final String data = new String(encodedBytes,
"US-ASCII");
                                    readBufferPosition = 0;

                                    handler.post(new Runnable()
                                    {
                                        public void run()
                                        {
                                            out.setText(data);
                                            if(data.equals("1")) {
                                            speechdata = "Obstacle on front";

                                            speak();
                                        }
                                           else if(data.equals("2")) {
                                                speechdata = "Obstacle on left";

                                                speak();
                                            }
                                            if(data.equals("3")) {
                                                speechdata = "Obstacle on right";

                                                speak();
                                            }
                                            if(data.equals("4")) {
                                                speechdata = "Obstacle on both
sides";


                                                speak();
                                            }


                                        }
                                    });
                                }
                                else
                                {
```

```
                                    readBuffer[readBufferPosition++] = b;
                                }
                            }
                        }
                    }
                    catch (Exception ex)
                    {
                        stopWorker = true;
                    }
                }
            }
        });

        workerThread.start();


    }


    void callme()
    {
        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, "en-US");
        try {
            startActivityForResult(intent, 1);
            speechdata = "";
        } catch (ActivityNotFoundException a) {
            Toast t = Toast.makeText(getApplicationContext(),
                    "Opps! Your device doesn't support Speech to Text",
                    Toast.LENGTH_SHORT);
            t.show();
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
        super.onActivityResult(requestCode, resultCode, data);
        switch (requestCode) {
            case 1: {
                if (resultCode == RESULT_OK && null != data) {
                    ArrayList<String> text = data

.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

                    speechdata = text.get(0);

                    Toast.makeText(this, speechdata, Toast.LENGTH_SHORT).show();

                    if(speechdata.contains("restaurant")) {

                        Uri gmmIntentUri = Uri.parse("geo:0,0?q=restaurants");
                        Intent mapIntent = new Intent(Intent.ACTION_VIEW,
gmmIntentUri);
                        mapIntent.setPackage("com.google.android.apps.maps");
                        startActivity(mapIntent);
                    }
                    else if(speechdata.contains("college")) {
                        Uri gmmIntentUri =
Uri.parse("google.navigation:q=Symbiosis+Institute+of+Technology,+Pune+India");
```

```
                        Intent mapIntent = new Intent(Intent.ACTION_VIEW,
gmmIntentUri);
                        mapIntent.setPackage("com.google.android.apps.maps");
                        startActivity(mapIntent);


                    }
                    else if(speechdata.contains("food court")) {
                        Uri gmmIntentUri =
Uri.parse("google.navigation:q=ccd+Symbiosis+Institute+of+Technology,+Lavale");
                        Intent mapIntent = new Intent(Intent.ACTION_VIEW,
gmmIntentUri);
                        mapIntent.setPackage("com.google.android.apps.maps");
                        startActivity(mapIntent);


                    }
                    else if(speechdata.contains("s i b m")) {
                        Uri gmmIntentUri =
Uri.parse("google.navigation:q=Symbiosis+International+University,+Lavale");
                        Intent mapIntent = new Intent(Intent.ACTION_VIEW,
gmmIntentUri);
                        mapIntent.setPackage("com.google.android.apps.maps");
                        startActivity(mapIntent);


                    }
                    else if(speechdata.contains("D block")) {
                        Uri gmmIntentUri =
Uri.parse("google.navigation:q=d+block+Symbiosis+Institute+of+Technology,+Lavale")
;
                        Intent mapIntent = new Intent(Intent.ACTION_VIEW,
gmmIntentUri);
                        mapIntent.setPackage("com.google.android.apps.maps");
                        startActivity(mapIntent);


                    }
                    else if(speechdata.contains("Lupin")) {
                        Uri gmmIntentUri =
Uri.parse("google.navigation:q=Lupin,+Lavale");
                        Intent mapIntent = new Intent(Intent.ACTION_VIEW,
gmmIntentUri);
                        mapIntent.setPackage("com.google.android.apps.maps");
                        startActivity(mapIntent);


                    }
                }
            }
        }
    }


    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if (keyCode == KeyEvent.KEYCODE_VOLUME_DOWN) {

            callme();
            return true;
        }
```

```
        else if(keyCode == KeyEvent.KEYCODE_VOLUME_UP)
        {
            try {
                findBT();
                openBT();
            }
            catch (Exception e){}
        }


        return super.onKeyDown(keyCode, event);
    }


            void closeBT()throws Exception
    {
        stopWorker = true;
        is.close();
        bs.close();

        Toast.makeText(this,"Socket closed", Toast.LENGTH_LONG).show();

    }

    @Override
    public void onInit(int status) {
        // TODO Auto-generated method stub

        if (status == TextToSpeech.SUCCESS) {

            int result = tts.setLanguage(Locale.US);

            // tts.setPitch(5); // set pitch level

            // tts.setSpeechRate(2); // set speech speed rate

            if (result == TextToSpeech.LANG_MISSING_DATA
                    || result == TextToSpeech.LANG_NOT_SUPPORTED) {
                Log.e("TTS", "Language is not supported");
            } else {

                speak();
            }

        } else {
            Log.e("TTS", "Initilization Failed");
        }

    }
    void speak()
    {


        tts.speak(speechdata, TextToSpeech.QUEUE_FLUSH, null);

    }


}
```

9.3.3 Arduino Code

```
#include <SoftwareSerial.h>
#include <NewPing.h>
#define TRIGGER_PIN  9
#define ECHO_PIN     8
#define MAX_DISTANCE 200

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
int tx = 7;
int rx = 6;

SoftwareSerial bluetooth(tx, rx);

void setup()
{
  Serial.begin(9600);
  bluetooth.begin(9600);
  pinMode(13,1);
  pinMode(2,0);
   pinMode(3,0);
}

void loop()
{

 delay(50);
   int uS = sonar.ping();
  Serial.print("Ping: ");
  int cm=uS / US_ROUNDTRIP_CM;
  Serial.print(cm);
  Serial.println("cm");

   int a=digitalRead(2);
    int b=digitalRead(3);
  Serial.print("Ob1:-");
  Serial.println(a);
  Serial.print("Ob2:-");
  Serial.println(b);
```

```
 if(cm<15&&cm>0)
{
  bluetooth.print("1 ");
     delay(1500);
}
 else if(cm>15)
 {
 bluetooth.print("0 ");
   delay(1500);
 }
 else if(a==0&&b==0)
 {
 bluetooth.print("4 ");
  delay(1500);
 }
  else if(a==0)
 {
 bluetooth.print("2 ");
   delay(1500);
 }
 else if(b==0)
 {
 bluetooth.print("3 ");
  delay(1500);
 }
 }
```

10. Conclusion and Future Scope

- The directions given to the user are directing to a limited number of places. This has been designed such a way because a blind user cannot really travel long distances via foot. But still a future scope would be to add all places to the application.

- Only 2 sensors were used in the system, we can add another one to the other shoe for better detection. Connection for the sensors can be done by either assigning wireless ones or by a meshed wire setup.

- Detection of objects is done on a real time basis.

- So, finally we can conclude that a proper functioning system is made for helping the users in navigation and obstacle sensing. This will not only help them in detection, but also make them independent for transportation.

Further study can also include other uses embedded inside the application and wireless system, sensors with a better range etc. We can also add additional commands given by the application regarding which location is the user currently in and nearby places from current location.

11. References

- https://www.**arduino**.cc/en/Main/Software (for arduino IDE)
- developer.**android**.com/sdk/index.html (for android studio and environment)
- www.wikipedia.org (for initial references and study)
- https://www.parallax.com/sites/default/files/.../28015-**PING**-Detect-Distance.pdf
- www.coregravity.com/html/detecting_**obstacle**_with_ir__in.html
- wiki.iteadstudio.com/Serial_Port_**Bluetooth**_Module_(Master/Slave)_:_**HC-05**