Introduction

- This project is an android application that is aimed at locating nearby places such as hospitals, medical stores, departmental stores, bus stops, etc. for the user.
- The key feature of this project is the development of a voice operated navigation system easily accessible to visually impaired people.
- It is based on the needs of unsighted ones while it can be operated by other people also.
- The development of the project focuses on blind people providing an easy to access user interface allowing its user to directly operate the application.

Objectives

- **Locating the nearest desired destination.**
- **Tracing the shortest and feasible path to destination.**
- **Navigating till the destination.**
- **Voice input to specify the destination.**
- **Smart stick sensor to avoid obstacles in the path.**
- **Smart app gestures to open the android application.**

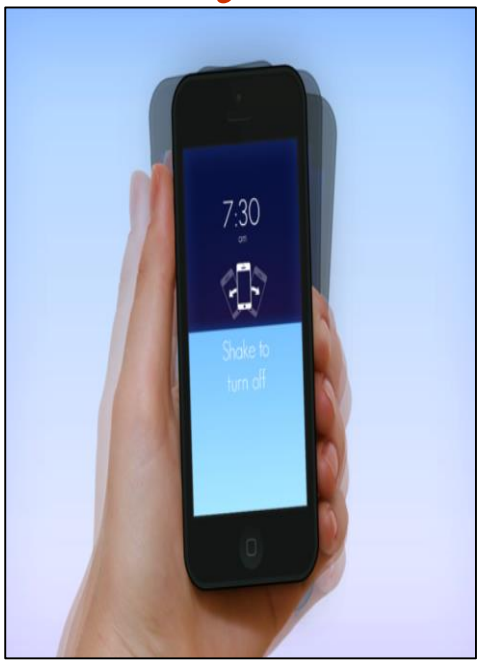
Project Modules

The project consists of the following four modules.

- **Gesture module.**
- **Navigation module.**
- **Speech-to-Text module.**
- **Smart Stick Sensor module.**

Gesture Module

The module allows the user to open the application by simply shaking the mobile device and double tapping on the home screen of the device. The gesture module facilitates the easy usability.



Shake the mobile device. The device motion is detected by the device Accelerometer.

Double tap on the device screen to open the navigation screen.

Gesture Module

How it works?

- Once the application is installed in the android device, the device requires a reboot.
- Once the reboot completes, the application starts listening to the shake event made by shaking the device.
- The Accelerometer detects the shake and a notification is generated alerting the user to double tap on the screen in order to open the app.
- When the user taps twice on the screen, the double tap gesture is recognised and a response is generated.
- The response is in the form of the execution of the Navigation module.

Gesture Module

The key steps involved in starting the android application using the gesture module consists of the following steps:

- *Generation of broadcast message by the system.*
- *Detection of shake event by the Accelerometer.*
- *A notification to alert the user to double tap on the screen.*
- *Double –Tap gesture detection.*

Gesture Module: Broadcast Message Generation

A broadcast receiver is an android component that simply responds to broadcast messages from other applications or from the system itself. These messages are also called intents.

When the reboot process completes, an intent is generated which is responded to by the registered broadcast receiver. The following broadcast receivers are used in this module:

"android.intent.action.BOOT_COMPLETED“

"android.intent.action.QUICKBOOT_POWERON"

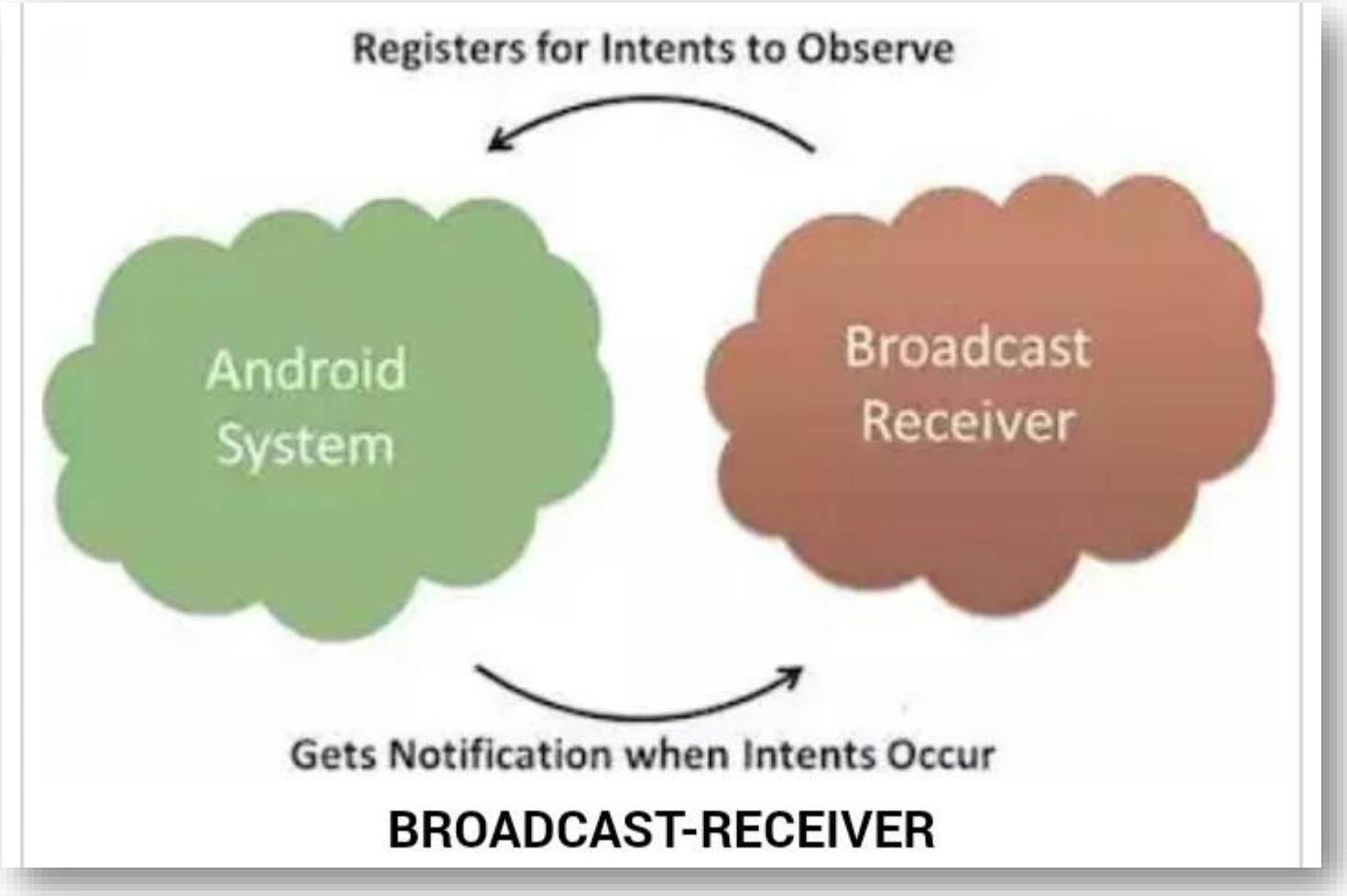
Gesture Module: Broadcast Message Generation

There are two important steps to make Broadcast Receiver work for the system broadcast intents:

Creating Broadcast receiver: This is implemented as a subclass of *BroadcastReceiver class* overriding the *onReceive()* method where each object is received as an Intent object parameter.

Registering Broadcast Receiver: An application listens for specific broadcast intents by registering a broadcast receiver in AndroidManifest.xml file.

Gesture Module: Broadcast Message Generation



Gesture Module: Detection of Shake Event by the Accelerometer

- The linear acceleration sensor or the Accelerometer provides a three-dimensional vector representing acceleration along each device axis, excluding gravity. We can use this value to perform gesture detection.
- The accelerometer is a good sensor to use if we are monitoring device motion. Almost every Android-powered handset and tablet has an accelerometer, and it uses about 10 times less power than the other motion sensors.
- Accelerometers use the standard sensor coordinate system.
- Conceptually, an acceleration sensor determines the acceleration that is applied to a device (Ad) by measuring the forces that are applied to the sensor itself (Fs) using the following relationship:
- $Ad = - \sum F_s / mass$

Gesture Module: Alert Notification

Once the application is ready to be opened after the detection of the shake event, a notification is generated to let him know that he can now double tap on the screen to open the app.

A notification appears with a notification icon in the notification area. To create the notification, *NotificationManager* **class is used which can be received from the context like an activity via the** *getSystemService()* **method.**

Gesture Module: Alert Notification

- **The *Notification.Builder* provides an builder interface to create an Notification object.**

- **Providing the notification a custom sound using:** *notify.sound = Uri.parse("android.resource://+ this.getPackageName() + "/" + R.raw.begin);*

- **When the notification is generated, the phone vibrates:**
  notify.defaults |= Notification.DEFAULT_VIBRATE

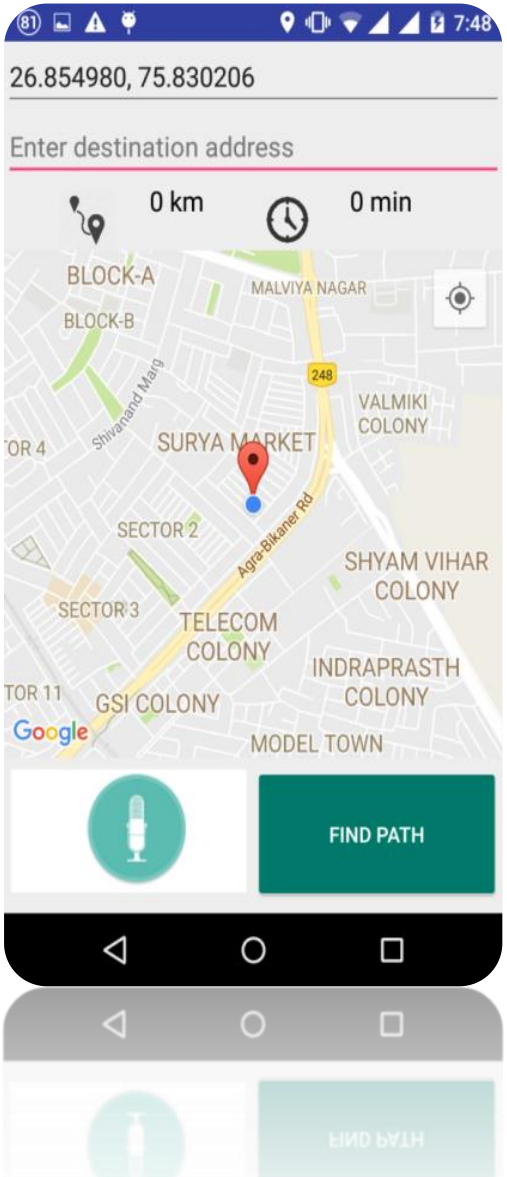
Gesture Module: Double-Tap Gesture Detection

The double-tap gesture is detected using the code which is written in *GestureDetect.java* file which extends the Activity class and implements the following interfaces:

- *GestureDetector.OnGestureListener*
- *GestureDetector.OnDoubleTapListener*

On double tap, the navigation activity opens for starting the navigation.

Navigation Screen



Navigation Module

- **This is the core module of app , and will enable our special users to find nearest destination along with a feasible walking path.**
- **This will provide a real time navigation with precise turning distance , coverage of circles , flyovers and is also enabled with automatic redirecting (if user deviates from its path).**
- **It uses GOOGLE MAPS API , and has inbuilt Distance Calculator and Total Duration Calculator**

Navigation Module: How it works?

- **Current location of user is detected using GPS of phone in terms of geo coordinates .**

```
double lati=0.0;
double longi=0.0;
GPSTracker tracker = new GPSTracker(this);
if (!tracker.canGetLocation()) {
tracker.showSettingsAlert();
}
else {
        lati = tracker.getLatitude();
       longi = tracker.getLongitude();
      }
```

- **Now , user enters the desired destination using speech to text module .**

- **Once the destination is entered and user press Find Path button a request is generated to find path and geo coding takes place .**

```
try {
   new DirectionFinder(this, origin,destination).execute();
    }
catch (UnsupportedEncodingException e) {          e.printStackTrace();
    }
```

- **Direction Finder class :**

```
return DIRECTION_URL_API + "origin=" + urlOrigin + "&destination=" + urlDestination +
    "&key=" + GOOGLE_API_KEY ;
```

- **On long click : An android long click listner is called which calls the Google Map Intent with source and destination parameters**

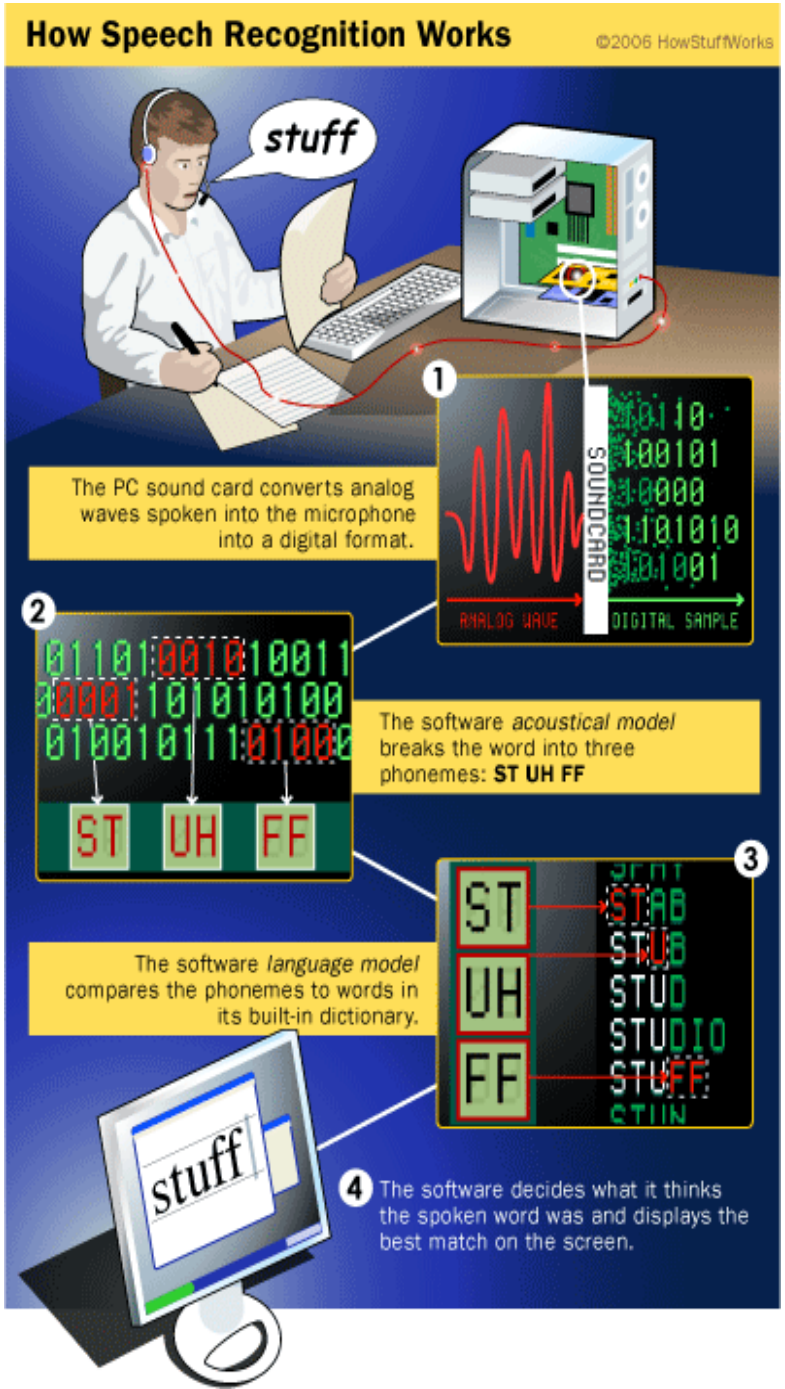
```
btnFindPath.setOnLongClickListener(new View.OnLongClickListener()
{
  public boolean onLongClick(View v) {
      sendCallMap();
      return true;    }
```

- **Send call map function:**

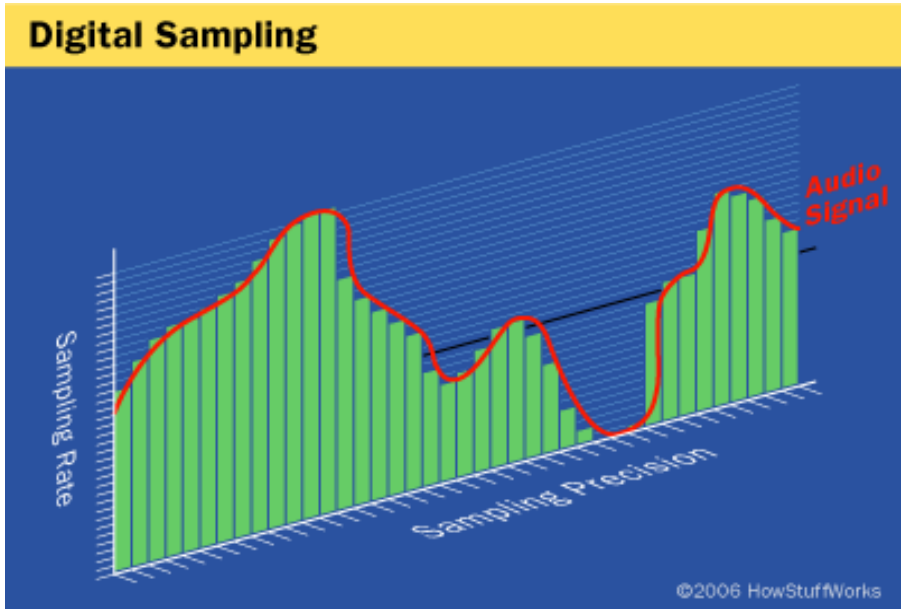
```
String Urll = new DirectionFinder(this, origin, destination).createUrll();
Uri gmmIntentUri = Uri.parse("google.navigation:q="+Urll+"&mode=w");
Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
mapIntent.setPackage("com.google.android.apps.maps");          startActivity(mapIntent);
```

Speech to text Module

- **This helps the user to enter destination through his own voice. This Module comprises of Google speech to Text API which works on deep learning method called Long short-term memory (LSTM)**
- **An ADC translates the analog waves of your voice into digital data by sampling the sound. The higher the sampling and precision rates, the higher the quality. To convert speech to on-screen text or a computer command, a computer has to go through several complex steps. When you speak, you create vibrations in the air.**



The **analog-to-digital converter (ADC)** translates this analog wave into digital data that the computer can understand.



Smart Stick Sensor Module

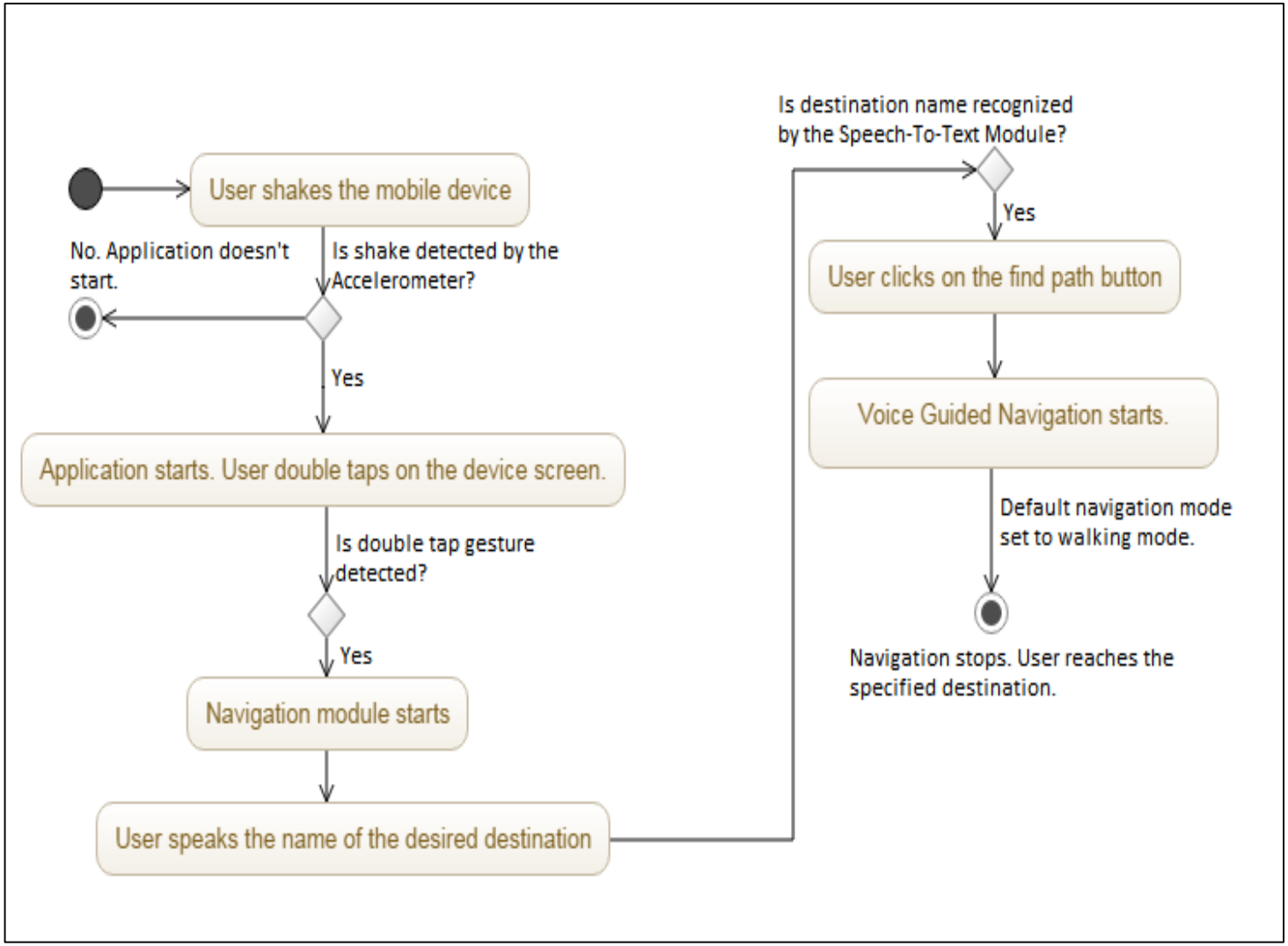
- As name suggests , smart stick is basically blind stick mounted with proximity sensors which will enable user to detect upcoming obstacles in real time.
- This sensor has a range of 3-4 ft. and also has an inbuilt buzzer which alarms users of forthcoming hurdles.

Smart Stick Sensor Module

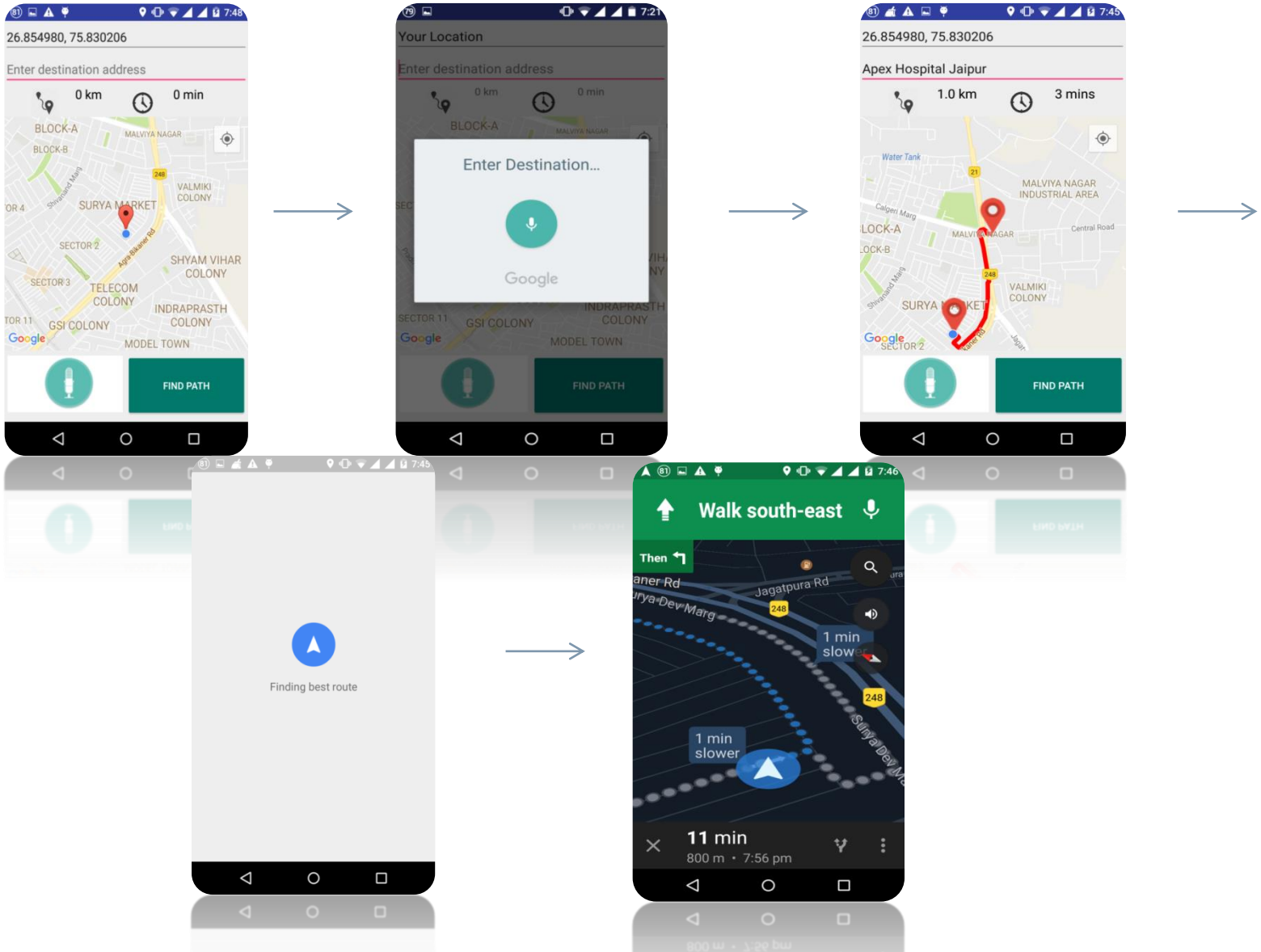
Smart Stick Sensor Module: How it works?

- **Ultrasonic Sensors**
- **These sensors emit radio or ultrasonic waves that bounce off of objects coming is user's path , much like some animals use echolocation. The returning waves are then registered and analysed by a microcontroller . By measuring the time it took the wave to return to the sensor the frequency of beeping is decided . If the object is far from the user the buzzer gives a low frequeacy sound , as soon as the user reaches near obstacle the frequency continuously increases intimating user to stop or change direction**

Activity Diagram



Voice Guided Navigation Process



References

- The following web pages have been referenced while implementing the above modules:
- [1] https://material.google.com/
- [2] https://github.com/googlemaps/
- [3] https://developer.android.content/BroadcastReceiver.html/
- [4] https://developer.android.com/guide/topics/sensors/sensors_motion.html/
- [5] https://developer.android.com/reference/android/location/Geocoder.html/
- [6] https://developers.google.com/maps/documentation/android-api/
- [7] https://developer.android.com/guide/topics/ui/notifiers/notifications.html
- [8] https://developer.android.com/reference/android/speech/package-summary.html
- [9] https://developer.android.com/guide/topics/permissions/index.html
- [10] https://developer.android.com/training/gestures/detector.html

Thank You