**1. INTRODUCTION**

'Smart System for blind people' is a project being done for the purpose of navigating and helping any blind person walk. Few main components in this project would include Sensors, Bluetooth, Smart Phone, Shoes to attach the sensors, Arduino Nano etc. This system will help a person navigate to a place by taking the speech and converting to text, then processing it and also vice versa. It will also help in detection of any object that comes close to the user.

**1.1 Purpose**

The purpose of this document is to give a detailed description of the requirements for 'Smart system for blind people'. It will illustrate the purpose and complete declaration for the development of the system. It will also explain the constraints, interface and user interactions. This document intended to be proposed to the customer/user before use and a reference for developing further into the project.

**1.2 Scope**

With the increasing technology and modernization of the world, many handicapped people find it tough to cope up with it. Why not we make use of the same technology and make this world a better place to be in? Further, this project will enable blind people to get along on their own without needing any help from others. This could really mean a big thing to many users. This system uses ultrasonic and obstacle sensors to detect any object close by. A blind man cannot use a phone's interface, so we will provide an app that runs with a click of a button and start taking commands from the user and then navigate or guide him accordingly. This kind of project will be extremely helpful and make the user's life very comfortable.

**1.3 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)

**1.4 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.

**2. Overall Description**

**2.1 Process Model**

**2.2 External Interface Requirements**

**2.2.1 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.

**2.2.2 Hardware Interfaces**

* **Arduino Nano**

Arduino is common term for a software company, project, and user community, that designs and manufactures computer [open-source hardware](https://en.wikipedia.org/wiki/Open-source_hardware), [open-source software](https://en.wikipedia.org/wiki/Open-source_software), and [microcontroller](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/USB)) on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) based on a programming language named [Processing](https://en.wikipedia.org/wiki/Processing_(programming_language))

* **Ultrasonic/Ping sensor**

Ultrasonic transducers are [transducers](https://en.wikipedia.org/wiki/Transducer) that convert [ultrasound](https://en.wikipedia.org/wiki/Ultrasound) waves to [electrical signals](https://en.wikipedia.org/wiki/Signal_(electrical_engineering)) or vice versa. Those that both transmit and receive may also be called ultrasound transceivers; many ultrasound sensors besides being [sensors](https://en.wikipedia.org/wiki/Sensor) are indeed transceivers because they can both sense and transmit. These devices work on a principle similar to that of transducers used in [radar](https://en.wikipedia.org/wiki/Radar) and [sonar](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/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. Whereas 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](https://en.wikipedia.org/wiki/Prototype) of [electronics](https://en.wikipedia.org/wiki/Electronic_circuit). It is used to divide/integrate a signal of the system.

**2.2.3 Software Interfaces**

* **Arduino IDE**

The Arduino project provides the Arduino [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE), which is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application written in the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It originated from the IDE for the languages [Processing](https://en.wikipedia.org/wiki/Processing_(programming_language)) and [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)). 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](https://en.wikipedia.org/wiki/Syntax_highlighting), [brace matching](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules to organize code. The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/Java_SE), [Java EE](https://en.wikipedia.org/wiki/Java_EE) or [Java ME](https://en.wikipedia.org/wiki/Java_ME) platforms released by [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation) in the form of a binary product aimed at [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) developers on [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) ,[Linux](https://en.wikipedia.org/wiki/Linux), [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) or [Windows](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/Java_(software_platform)) platform, it has been by far the most widely used Software Development Kit ([SDK](https://en.wikipedia.org/wiki/Software_development_kit)).

* **SDK**

The Android [software development kit](https://en.wikipedia.org/wiki/Software_development_kit) (SDK) includes a comprehensive set of development tools. These include a [debugger](https://en.wikipedia.org/wiki/Debugger), [libraries](https://en.wikipedia.org/wiki/Software_library), a handset [emulator](https://en.wikipedia.org/wiki/Emulator) based on [QEMU](https://en.wikipedia.org/wiki/QEMU), documentation, sample code, and tutorials. Currently supported development platforms include computers running [Linux](https://en.wikipedia.org/wiki/Linux_kernel) (any modern desktop [Linux distribution](https://en.wikipedia.org/wiki/List_of_Linux_distributions)), [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) 10.5.8 or later, and [Windows XP](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for developing for the [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) platform.

* **Google maps API**

Google Maps is a desktop [web mapping](https://en.wikipedia.org/wiki/Web_mapping) service developed by [Google](https://en.wikipedia.org/wiki/Google). It offers [satellite imagery](https://en.wikipedia.org/wiki/Satellite_imagery), street maps, 360° panoramic views of streets ([Street View](https://en.wikipedia.org/wiki/Google_Street_View)), real-time traffic conditions ([Google Traffic](https://en.wikipedia.org/wiki/Google_Traffic)), and [route planning](https://en.wikipedia.org/wiki/Route_planner) for traveling by foot, car, bicycle (in [beta](https://en.wikipedia.org/wiki/Beta_test)), or [public transportation](https://en.wikipedia.org/wiki/Public_transportation).

**2.2.4 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.

**2.3 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.

**3. Project Design**

**3.1 Analysis Models**

**3.1.1 Entity Relationship Diagram**

**3.1.2 State Transition Diagram**

**3.2 Software System Attributes**

**3.2.1 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.

**3.2.2 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.

**3.2.3 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.

**3.2.4 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.

**3.2.5 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 devices with Bluetooth.