15SE401M – Multi Disciplinary Design

SEMESTER- VII

ACADEMIC YEAR: 2019-20

NAME: Shreayan Chaudhary

REGISTER NUMBER: RA1611020010011

NAME: Ankit Sahu

REGISTER NUMBER: RA1611020010095

NAME: Paras Sibal

REGISTER NUMBER: RA1611020010055

PROGRAM: B. Tech – SWE

SUBMITTED TO: C.G ANUPAMA

DEPARTMENT OF SOFTWARE ENGINEERING

Faculty of Engineering and Technology

SRM IST

(Under section 3 of the UGC act, 1956)

Acknowledgement

We would like to take this opportunity to thank the **Director of E&T, Dr. Muthmamizehelvan** and **Head of Department of Software Engineering, Dr. C. Lakshmi,** for making the Multi-Disciplinary Design a part of the curriculum and possible for the students.

We also convey our immense gratitude for **Mrs. C.G. Anupma, Assistant Professor (S.G), Department of Software Engineering** who gave us constant inspiration and suggestions throughout the project work. Under her guidance, we were able to complete the project on time. He has not only helped us in the project but has also been a great mentor to us.

Lastly, we would like to thank the entire Software Engineering Department for the constant support and co-operation throughout out project

TABLE OF CONTENTS

1. Abstract

In today's world, the disabled people are being treated very differently as compared to other beings. They often face difficulties while performing day to day work. The primary goal of this project is to address the given problem and help those with disabilities so they can live a comfortable life. The purpose of our product is to guide the visually impaired people so that they can avoid the obstacles that come in their way making their day to day life easier. The product will capture the videos of the surroundings of the user and guide him accordingly. It will be paired to the smartphone of the user so that navigation, and accessibility of the phone will increase. It will also be connected to the headphones of the user so that the text will be converted in the form of speech and the user will be able to "hear" his/her surroundings.

2. Software Requirements Specification

2.1 Introduction

The main objective of this Software Requirements Specification (SRS) is to give a brief idea and facilitate the project (Smart Assistant for Visually Impaired) which is carefully designed and implemented to facilitate the visually blind people who are being treated very differently as compared to other beings. This project will address the problem of the visually impaired people and help those with disabilities so they can live a comfortable life.

2.1.1 Purpose

The purpose of our model is to guide the visually impaired people so that they can avoid the obstacles. This will help those with disabilities so they can live a comfortable and making their day to day life easier.

2.1.2 Document Conventions

This document follows MLA (Modem Language Association) format. Bold-faced text has been used to emphasize section and sub-section headings. Highlighting is to point out words in the glossary and italicized text is used to label and recognize diagrams.

2.1.3 Intended Audience and Reading Suggestion

This document is a prototype to be read by the development teams, project managers, marketing staff, testers & documentation team. The target audience for this project are the visually impaired people. Our model will be guiding them so that they can easily avoid all these obstacles in front of them.

2.1.4 Product Scope

The purpose of our model is to ease day to day tasks for the visually impaired people and people having poor vision. Our model will enable the user to perform activities and will act like a

guiding agent to them. Our model will operate within the distance on 20 metres and will use frequency less than 16 KHz for ultrasonic sensor.

2.2 Overall Description

2.2.1 Product Perspective

The project will use a headband connected to the phone and the earphones of the user. The camera will capture a video of the environment as perceived by the user i.e. the peripheral vision of the user. The model will then get screenshots from the video at regular intervals. We will train our model to detect various objects like walls, cars, people, and other frequent objects. Our model will be guiding them so that they can easily avoid all these obstacles in front of them.

2.2.2 Product Function

- Detect obstacles
- Guide the user agent

2.2.3 User Classes and Characteristics

The user classes are:

- End user – Visually Blind People
- Admin – System
- Admin – Model

2.2.4 Operating Environment

Minimum system requirements are:

- Android Lollipop (version-5.1.3) and above.
- Qualcomm Snapdragon 430 processor (1.2GHz) and above.
- 3 GB RAM and above.
- 20MB storage space on device.
- 5MP rear camera.
- GPS enabled smartphone.
- Earphones.

2.2.5 Design and Implementation Constrains

It is assumed that the user has an Android smartphone with the following specifications:

- Detects objects up to a distance of 20 m.
- Requires well lit areas to function.
- Battery needs to be charged enough to work.

2.2.6 User Documentation

A user manual will be provided with the product. The user manual will contain detailed instructions to use the product and the technical specifications of the product. Non-technical as well as experienced people can use the user manual.

2.3 External Interface Requirement

2.3.1 User Interfaces

The device is voice enabled so that the visually blind people can interface with device guiding the user.

2.3.2 Hardware Interfaces

- There is a LDR sensor whose resistivity is a function of the incident electromagnetic radiation. Hence, they are light sensitive devices and will illuminate light during night alerting audience of user presence.
- The device also contains an Ultrasonic sensor with frequency of 16 KHz which sends t high frequency beam to detect obstacles.
- Camera of at least 5 megapixels is used to capture screenshots of peripheral environment.

2.3.3 Software Interfaces

The product involves the use of a Machine learning concepts with python language. The device uses YOLO object detection to detect obstacles.

2.4 System Features

- **Dark Mode**
  We have used LDR sensors which depends on the photosensitivity of light and will illuminate during the night-time. This will help the user to use the device even at night.

- **GPS Tracking**
  Our product will use GPS tracker equipped in user's phone which will track the user's location in real time and will help him reach from the source to the destination with ease. It will only work in places where the GPS signal is available.

- **Object Detection**
  Our product will predict the objects in front of the user which are potentially obstacles in the user's way. These objects may be cars, wall, other people, pets, street lights etc. We have used Ultrasonic sensors to measure the distance between the objects and the user. We will also be using YOLO object detection on the video to detect the object. YOLO outperforms top detection methods like DPM and R-CNN by a wide margin.

2.5 Other Non-Functional Requirement

2.5.1 Performance Requirement

The application must be free from any sort of lags that is any task being performed must be completed at that very particular instance. Most of the high end mobile devices should be able to support all major functions of the application. More over the application should be immune to frequent crashes.

2.5.2 Safety Requirement

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up and reconstructs a more current state by reapplying or redoing the operations from the backed up log, up to the time of failure.

2.5.4 Software Quality Attributes

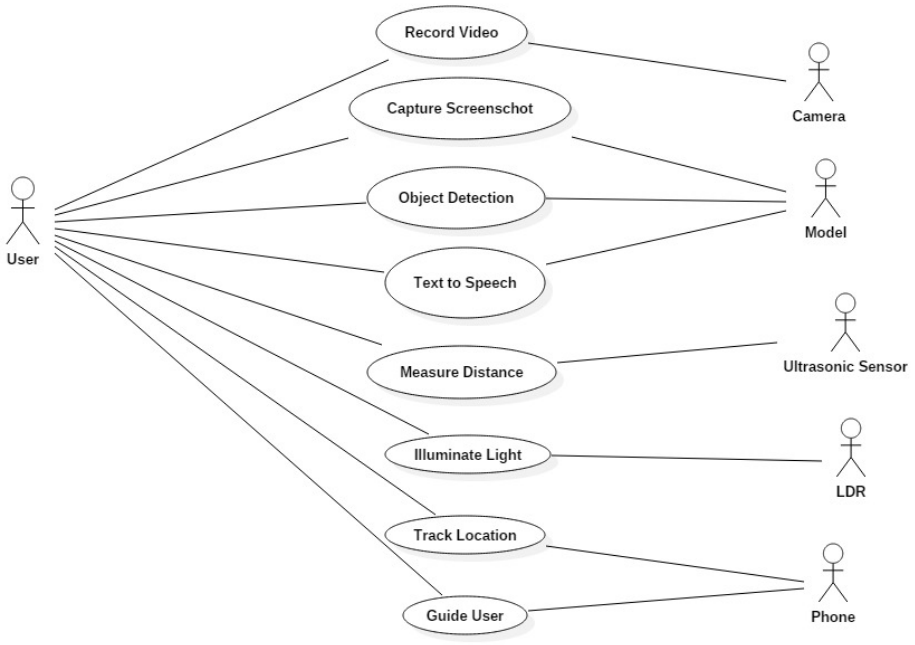
Our model has a very secure security system, immune to most attacks on the applications' database. It can identify its user's voice hence the issue of resolving forgotten passwords, addressing account lockouts, and expiring inactive sessions shall be solved by a two-way Authentication process wherein the user receives an Email as well as an OTP on the registered mobile number to regain access.

2.5.5 Business Rules

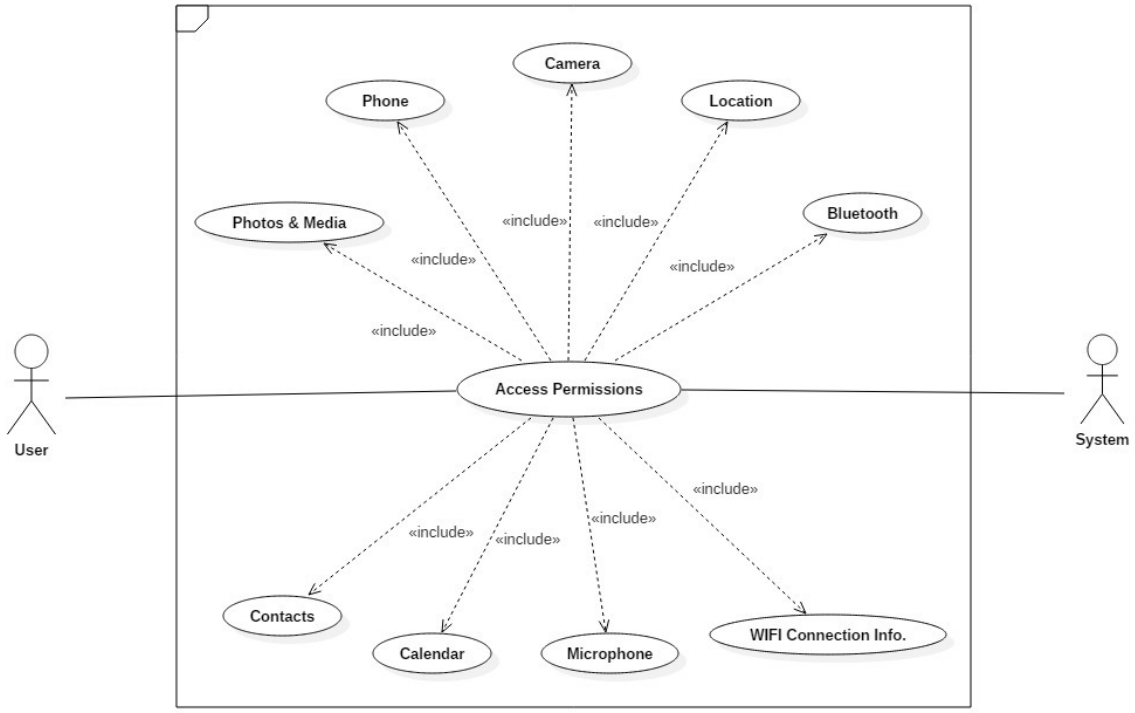
- All prices are inclusive of GST
- The price of the application may vary from region to region depending on the taxes.
- The application may not be available in some countries due to the laws pertaining to that country.
- The price of the product is very reasonable and affordable to everyone.
- The management provides all the hardware.

3 Software Design Document

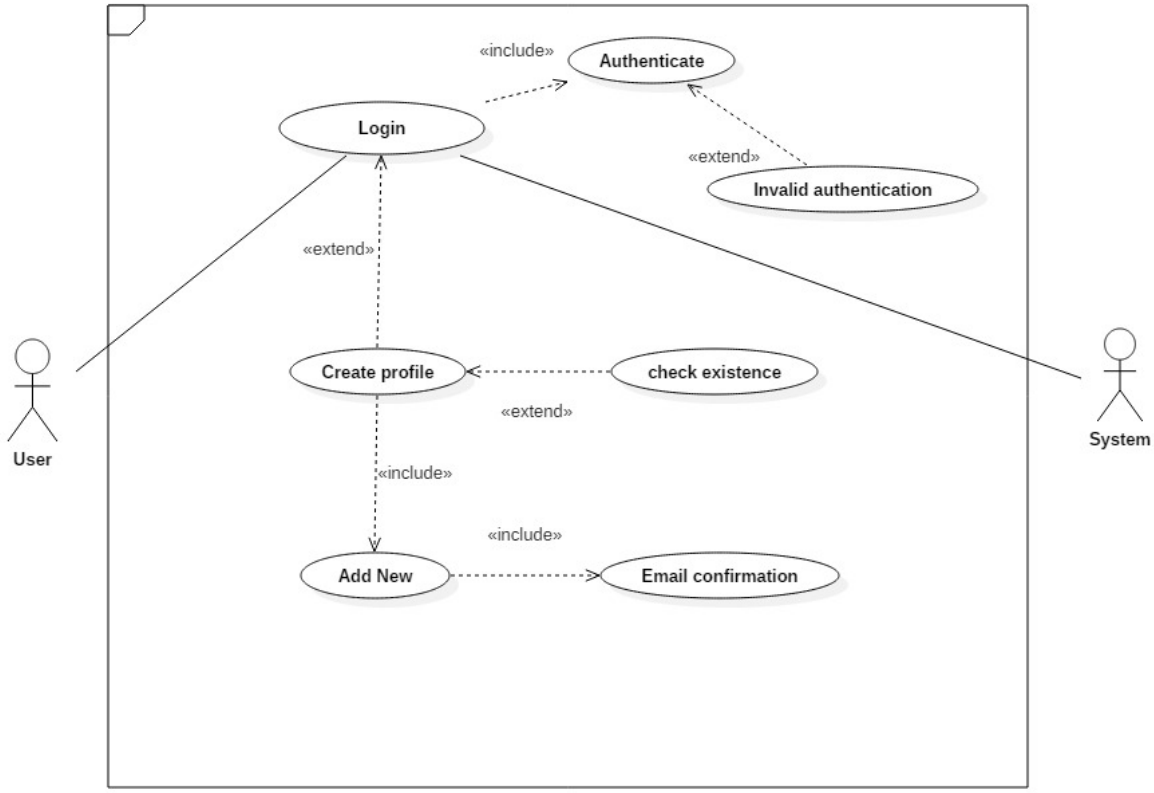
3.1 Use Case Diagram



According to the requirement, we have made three use case diagrams. In the above use case diagram, we have five actors which are, user, model, phone, camera, and sensor. We have shown the interaction of the actors with the use cases which are, record video, capture screenshots, object detection, text to speech track location and guide user. The sensor actor will perform actions like measure distance and illuminate light.



The second use case diagram has only two actors which is the user and system, and it shows the interactions between them. The user and system needs to access permission for accessing the camera, location, phone, and others.



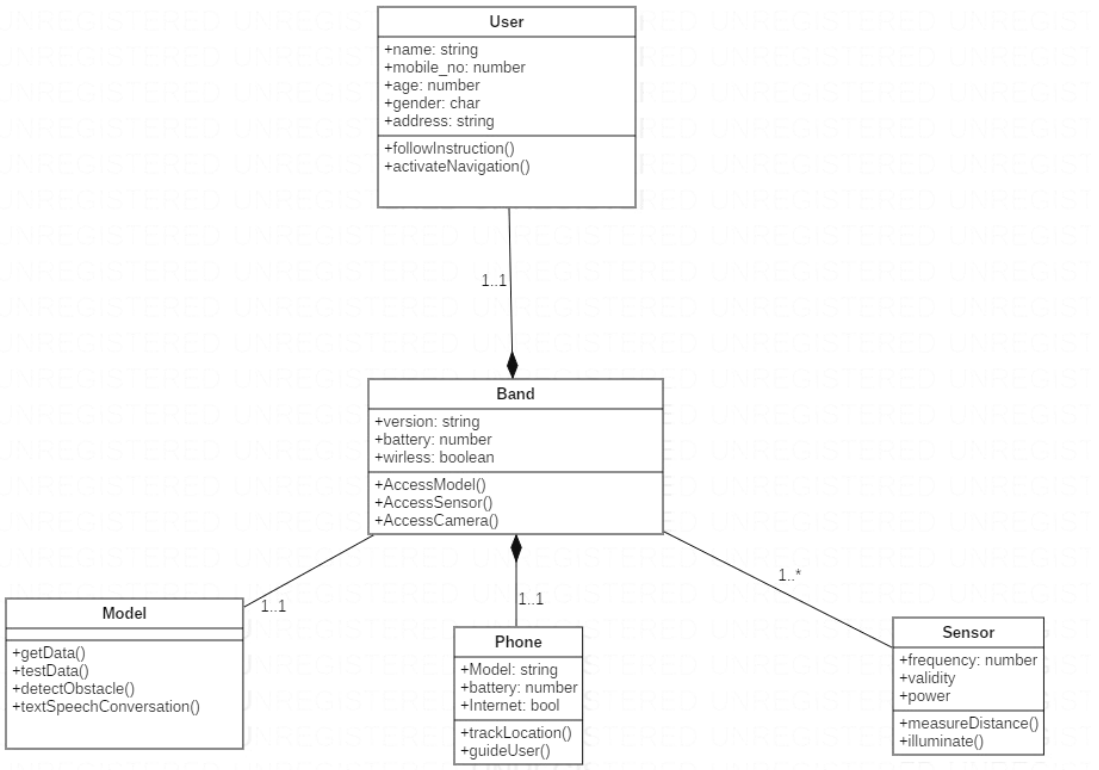
The third use case diagram has only two actors which is the user and system, and it shows the interactions between them. This shows that the user need to sign up and then login to access the functionalities. The system will guide user through the process.

Use Case Name:	**Headband**
Actor(s):	user, model, phone, camera, and sensor
Summary Description:	The complete model will detect obstacles and guide path for visually impaired.
Priority:	Must Have
Status:	Medium Level of details
Pre-Condition:	The user must login in to the system
Post-Condition(s):	The sensors are activated and the user will be guided

Use Case Name:	**Headband permissions**
Actor(s):	user, system
Summary Description:	The user and system needs to access permission for accessing the camera, location, phone, and others.
Priority:	Must Have
Status:	Medium Level of details
Pre-Condition:	The user must login in to the system
Post-Condition(s):	The user will be allowed to use and access the system functions

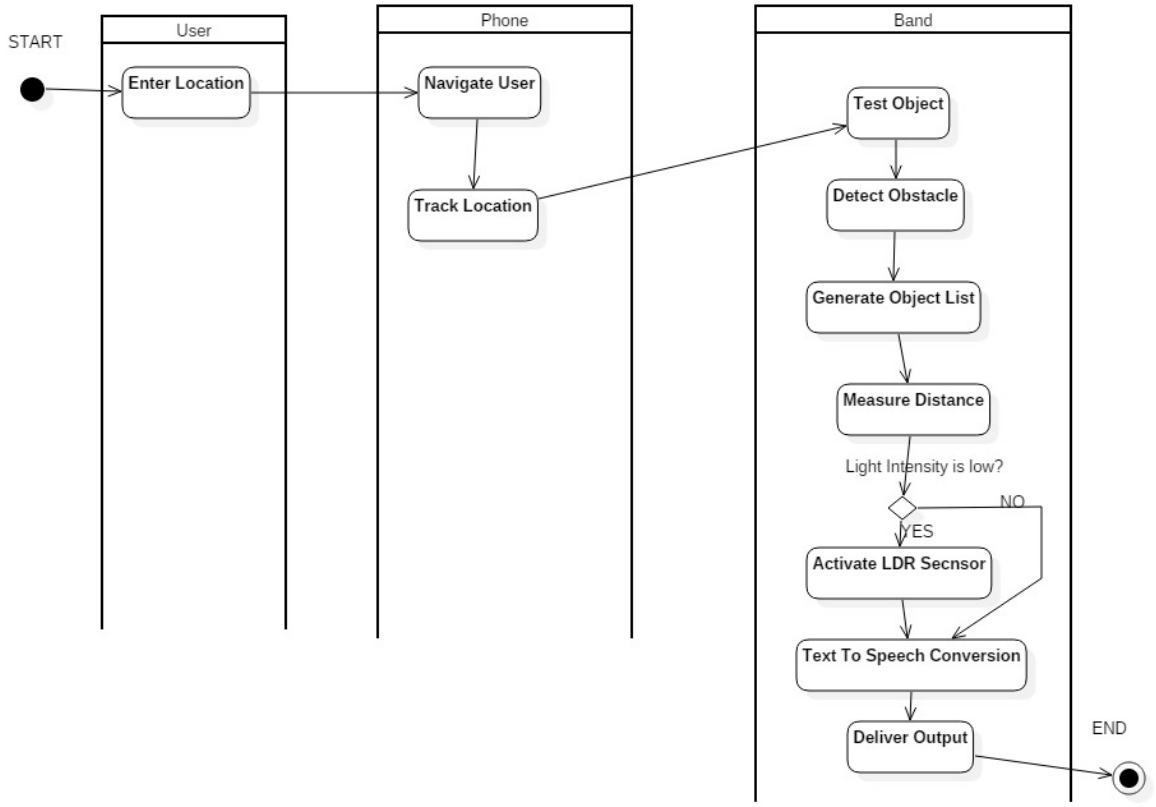
Use Case Name:	**Authentication of user's profile**
Actor(s):	user, system
Summary Description:	It shows the interactions between them. This shows that the user need to sign up and then login to access the functionalities. The system will guide user through the process.
Priority:	Must Have
Status:	Medium Level of details
Pre-Condition:	The user must sign up in to the system
Post-Condition(s):	The user will be logged in to access functions

3.2 Class Diagram



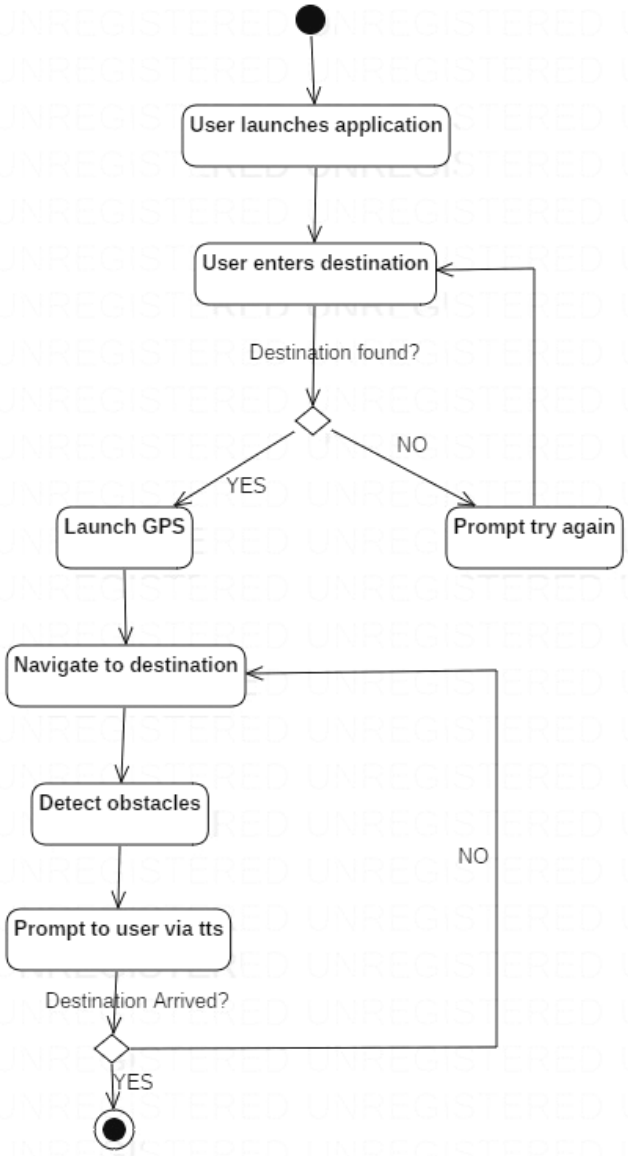
This above diagram is the class diagram for our project. A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The different classes in our model is that of User, Model, Phone, Band, and Sensor. Their attributes and relations have been shown and the relationship between the different classes have been shown as well

3.3 Activity Diagram



Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. The activity diagram starts from the detection of water level and carries on until finding the alternate route, it shows all the activities involved in our project.

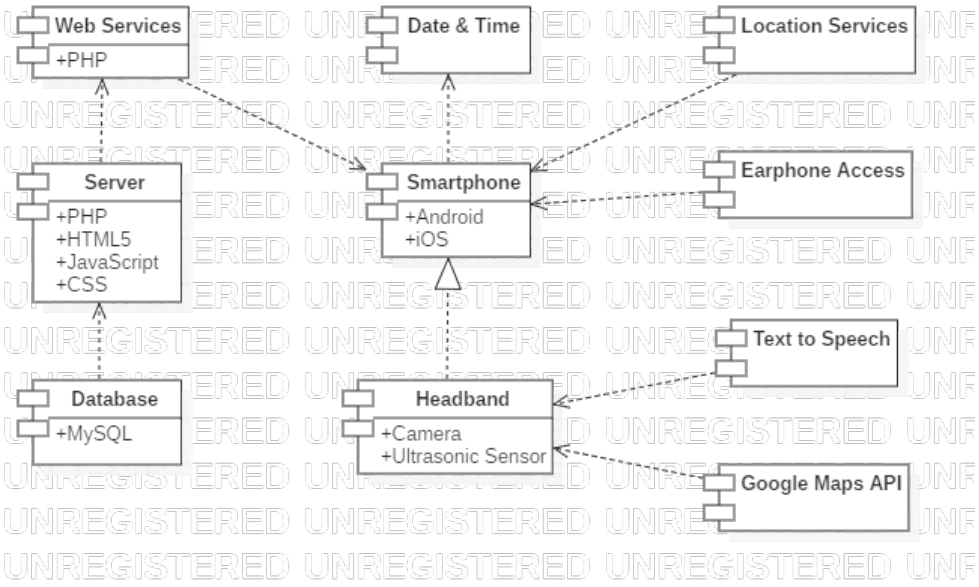
3.4 State Chart Diagram



A state chart diagram is a type of diagram to describe the behaviour of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics. The different states in our

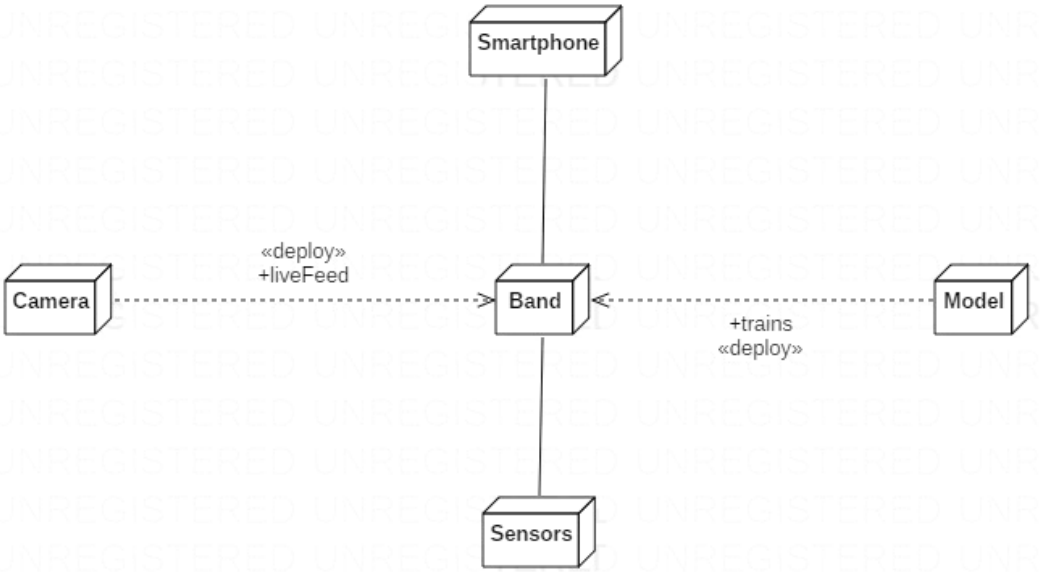
state diagram are water detection, alert route detection, direction follow up, destination and car evacuation.

3.5 Component Diagram



A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development. The physical components of the component diagram are location services, web services, server, smartphone, earphones access, text to speech, database, google maps, and headband.

3.6 Deployment Diagram



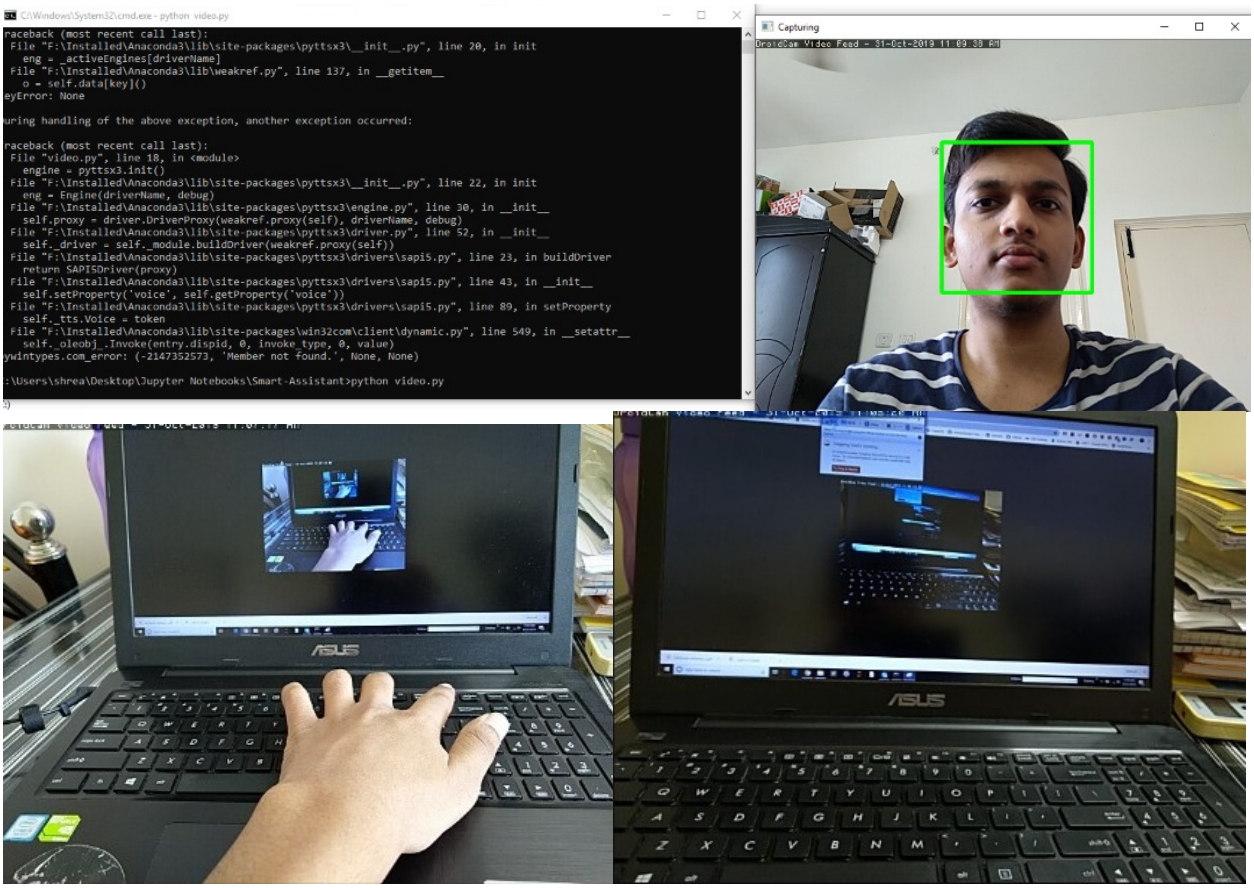
A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. The software part of our project includes the smartphone and the model whereas the hardware part includes the camera and sensors.

4. Code

```
#!/usr/bin/env python

# coding: utf-8

# In[1]:

# Importing the OpenCV library

import cv2

import time

# Importing python text ot speech library

import pyttsx3

# In[2]:

# Initializing the engine

engine = pyttsx3.init()

# In[3]:

# Initializing video

# video = cv2.VideoCapture('http://192.168.1.2:4747/video/mjpegfeed?640x480')

video = cv2.VideoCapture(0)

# In[4]:

# Creating a CascadeClassifier

face_cascade = cv2.CascadeClassifier("models/haarcascade_frontalface_default.xml")

# body_cascade = cv2.CascadeClassifier("haarcascade_fullbody.xml")

# numberplate_cascade = cv2.CascadeClassifier("haarcascade_licence_plate_rus_16stages.xml")

# In[5]:
```

```
# Initialize the flag variable

flag = 1

# In[6]:

while True:

    flag+=1

    check, frame = video.read()

    # gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    face_found = 0

    # Search co-ordinates of face

    faces = face_cascade.detectMultiScale(frame, scaleFactor = 1.05, minNeighbors = 5)

    for x,y,w,h in faces:

        face_found = 1

        img = cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 3)

    cv2.imshow('Capturing', frame)

    # Waiting for 1ms

    key = cv2.waitKey(1)

    # Break if user presses quit

    if(key==ord('q')):

        break

# In[7]:

if(face_found==1):

    engine.say("Careful, there is a person ahead!")
```

```
    engine.runAndWait()
else:
    engine.say("Your path is clear!")
    engine.runAndWait()
# In[8]:
print(flag)
# In[9]:
video.release()
# In [10]:
cv2.destroyAllWindows()
```



5. Conclusion

With the proposed architecture, if constructed with at most accuracy, the blind people will able to move from one place to another without others help, which leads to increase autonomy for the blind. The developed smart stick that is incorporated with multiple sensors will help in navigating the way while walking and keep alarming the person if any sign of danger or inconvenience is detected. The developed prototype gives good results in detecting obstacles paced at distance in front of the user; it will be real boon for the blind. At the same time global positioning system (GPS) can be linked with the voice stick for navigation, so that person can know his current position and distance from the destination which will be informed to users through voice instruction

6. Reference:

1. https://edutechlearners.com/object-oriented-system-development-by-ali-bahrami/
2. https://belitsoft.com/custom-application-development-services/software-requirements-specification-document-example-international-standard
3. http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf
4. https://www.python.org/
5. https://keras.io/
6. https://www.cv/foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf
7. https://www.csie.ntu.edu.tw/~r95004/Refactoring_improving_the_design_of_existing_code.pdf
8. https://www.numpy.org/
9. https://www.amazon.in/Learn-Python-3-Hard-Way/dp/9352865103
10. http://index-of.es/Varios 2/Hands%20on%20Machine%20Learning%20with%20Scikit%20Learn%20and%20Tensorflow.pdf