



THIRD EYE

A MOBILE APPLICATION FOR VISUALLY IMPAIRED PEOPLE TO ASSIST WITH DAILY TASKS

22_23-J 83

OUR TEAM



Ms. Uthpala Samarakoon
Supervisor



Ms. Suriyaa Kumari
Co-supervisor



M.C.J. Costa
IT17088138



A.D. Amarasinghe
IT19133614



K.D.J. Jayasekara
IT19103600



A.M.N.D. Aththanayake
IT19981918

RESEARCH PROBLEM

There's no better product or software that can easily used by the visually impaired people to do their daily tasks easily.

- Limitations and problems in existing mobile applications
- Unaffordable high cost
- Security vulnerabilities and restrictions
- Don't fulfil blind people requirements (navigation, objects recognitions)
- Evaluating color blindness type does not exist
- Real time support not existing
- Usability issues

OBJECTIVES

MAIN OBJECTIVE

Implement an application that can guide and assist blind people and visually troubled individuals. They can have support in which areas they have difficulties. Their difficulties can be overcome by this system which is fully automated with a built-in voice assistant and utilities that can support each of these said groups.

OBJECTIVES

SUB OBJECTIVES

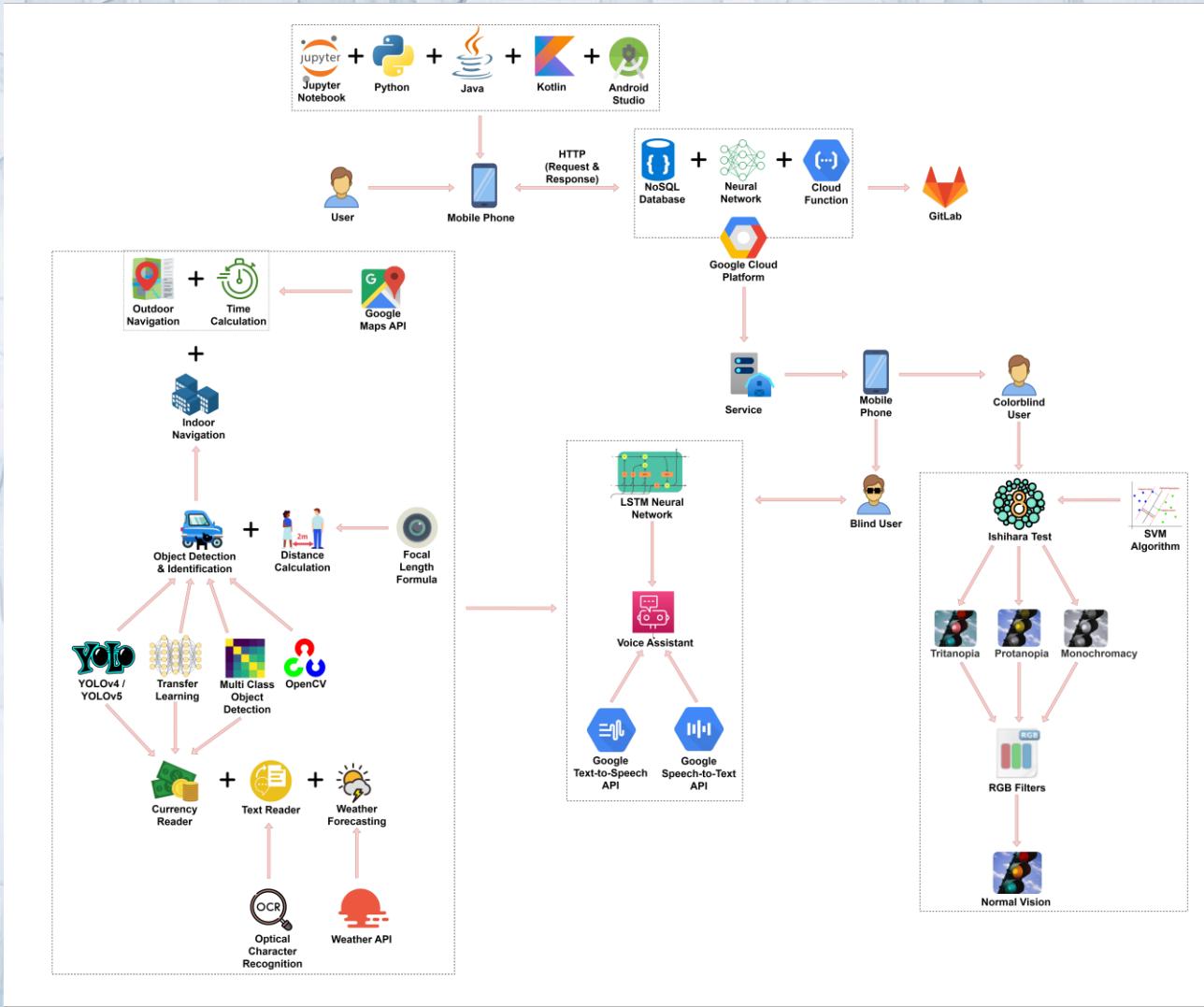
1. Implement real time map navigation with a voice assistant support and suggestions to navigations where the distance can be reduced.
2. Implement real time text capturing and reading to assist blind users to understand languages. While weather forecasting can support when navigating through destination.
3. Implement object identifying when navigating through the journey to assist with obstacles, road signs as well as distances calculating to forecast collision detection.
4. Implement utilities where a colorblind user can minimize the difficulties they face when trying to identify color related elements for day-to-day activities. Support colorblind types by testing and evaluating which type of colorblindness is present and supporting throughout.

HARDWARE SOLUTION



- The back camera is used to capture the visual data
- Wireless/wired headphones are used to listen to the voice commands
- The microphone is used to give the commands

SYSTEM OVERVIEW DIAGRAM





M.C.J. Costa
IT17088138
Information Technology

RESEARCH PROBLEM

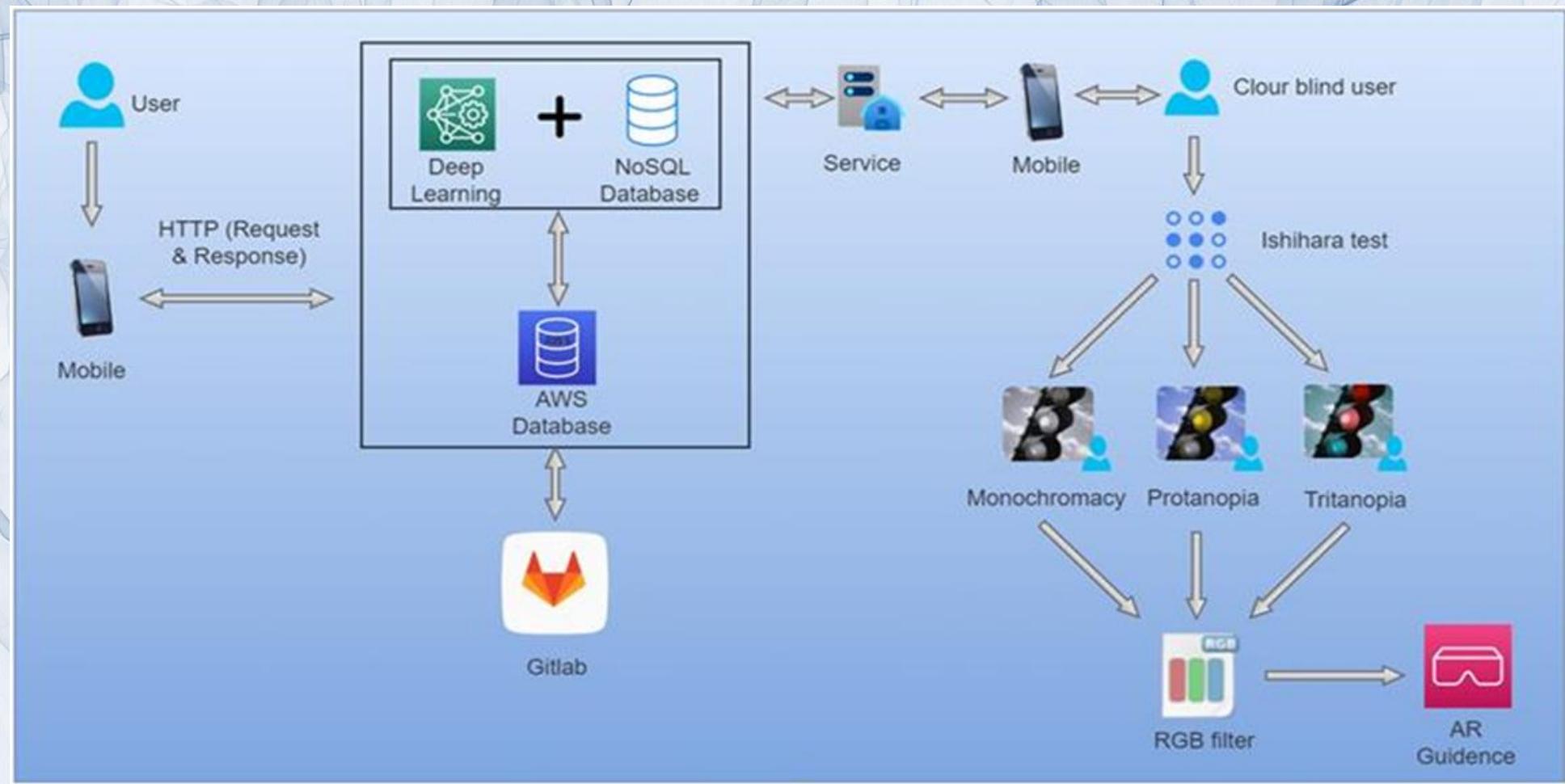
- Diagnosing a person what type of colorblindness they have.
- No current dataset supporting the users with correct color filter through the mobile device.
- Creating model to support Ishihara test.
- Create a mobile application to support the colorblind person.

PROPOSED SOLUTION

Implement real time support application which helps the colorblind people to get the QoL by using the smartphone accurately.

- Add a display filter depending on the evaluated result.
- Guide the user with augmented reality.
- To evaluate the type of colorblindness, create “Ishihara” tests.

COMPONENT OVERVIEW DIAGRAM

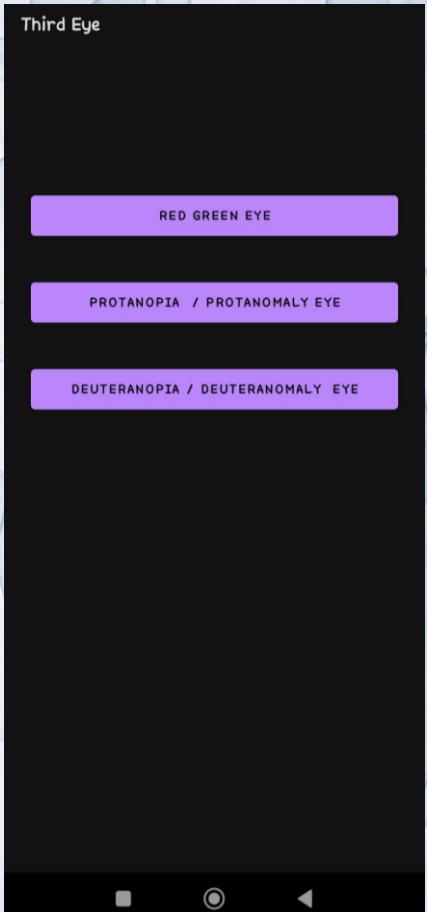
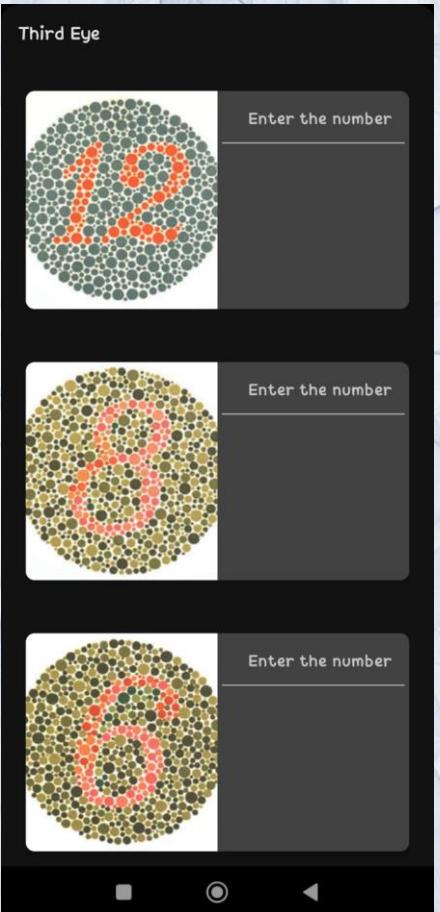


CURRENT PROGRESS 100%

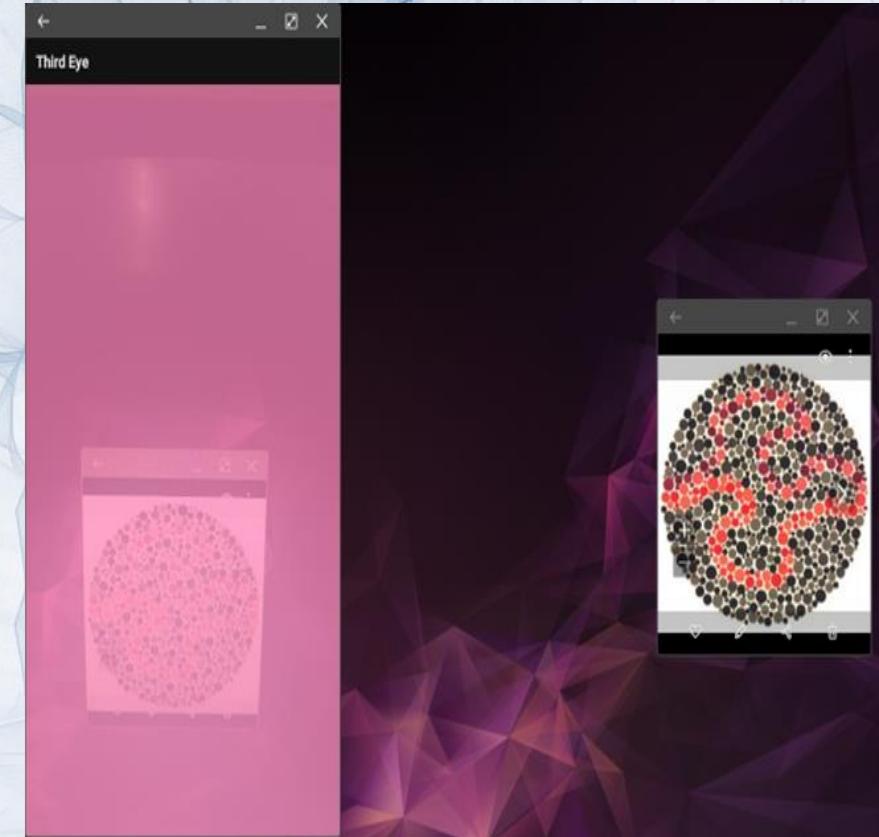
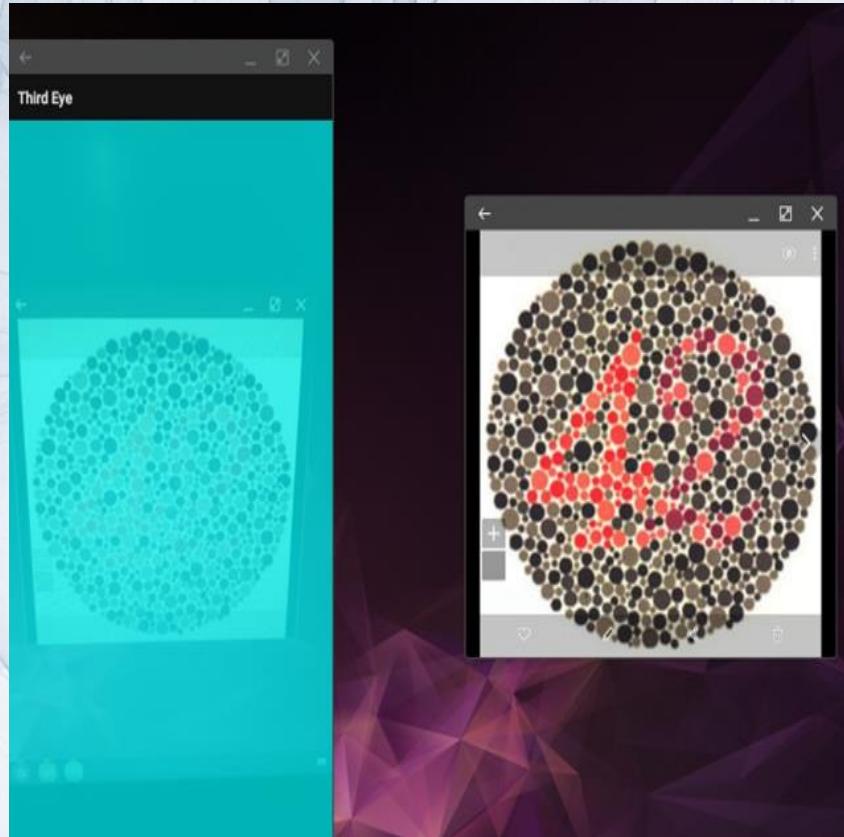
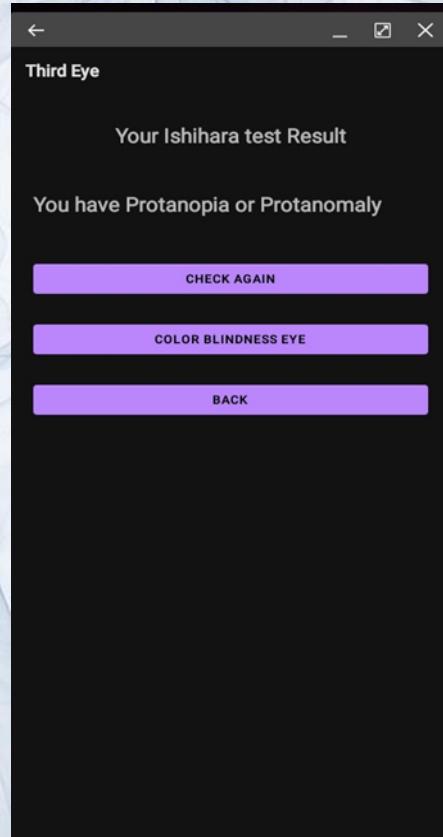


- ✓ Collected Ishihara diagnostics plates and data.
- ✓ Implemented a dataset using a gathered dataset and trained it.
- ✓ Created a custom model to identify different type of colorblind diagnosis.
- ✓ Developed a custom script to get the user input to finalize the color blindness conditions.
- ✓ Used the test results to implement a custom filters for user's mobile.
- ✓ Created and developed the model in cloud to use it anywhere with a lesser number of performance lag.
- ✓ Created a mobile application with UI to handle the Ishihara test.
- ✓ Improved UX when Ishihara test result response after populated.
- ✓ Implemented camera filter.

TESTING



TESTING



TESTING

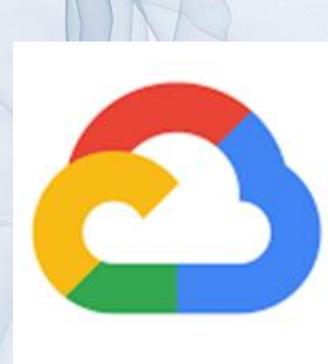
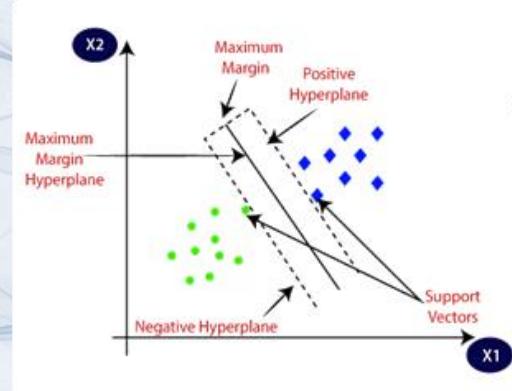


RISK MITIGATION

- Create a custom data set because no current data set.
- Training the model to display Ishihara test results with user's inputs.
- When training the model, the number of Ishihara plates were not significant. enough to get an accurate prediction for some color blindness types.
- When training the machine learning model what type of model needs to be considered (classification).
- Choosing between support vector classification and binary classification to get the accurate prediction.

TOOLS & TECHNOLOGIES

- Kotlin
- Pytorch
- Jupyter notebook
- Anaconda environment
- Java
- Google Cloud Platform
- Android Studio
- Python



PROOF OF WORK

```
IshiharaTest.java
49     // Declaring String variables for storing user inputs
50     private String Card1_text,Card2_text,Card3_text,Card4_text,Card5_text,Card6_text,Card7_text,Card8_text;
51
52     // Declaring Integer variables for storing the id of radiobutton that user have selected
53     private int selectedId1,selectedId2,selectedId3,selectedId4,selectedId5,selectedId6,selectedId7,selectedId8;
54
55     @Override
56     protected void onCreate(Bundle savedInstanceState) {
57         super.onCreate(savedInstanceState);
58         setContentView(R.layout.activity_ishihara_test);
59
60         // Start Initializing TextViews of the cards
61         textViewCard1 = findViewById(R.id.textViewCard1);
62         textViewCard2 = findViewById(R.id.textViewCard2);
63         textViewCard3 = findViewById(R.id.textViewCard3);
64         textViewCard4 = findViewById(R.id.textViewCard4);
65         textViewCard5 = findViewById(R.id.textViewCard5);
66         textViewCard6 = findViewById(R.id.textViewCard6);
67         textViewCard7 = findViewById(R.id.textViewCard7);
68         textViewCard8 = findViewById(R.id.textViewCard8);
69         textViewCard9 = findViewById(R.id.textViewCard9);
70         textViewCard10 = findViewById(R.id.textViewCard10);
71         textViewCard11 = findViewById(R.id.textViewCard11);
72         textViewCard12 = findViewById(R.id.textViewCard12);
73         textViewCard13 = findViewById(R.id.textViewCard13);
74         textViewCard14 = findViewById(R.id.textViewCard14);
75         textViewCard15 = findViewById(R.id.textViewCard15);
76         textViewCard16 = findViewById(R.id.textViewCard16);
77         textViewCard17 = findViewById(R.id.textViewCard17);
78         textViewCard22 = findViewById(R.id.textViewCard22);
79         textViewCard23 = findViewById(R.id.textViewCard23);
```

```
IshiharaTest.java x IshiharaResult.java x
1 package com.sliit.third_eye_app;
2
3 import ...
4
5 public class IshiharaResult extends AppCompatActivity {
6
7     TextView textView;
8
9     Button btn_checkAgain,btn_goto_cbc,btn_goback_activity;
10
11     @SuppressLint("MissingInflatedId")
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_ishihara_result);
16
17         textView = findViewById(R.id.textView);
18         btn_checkAgain = findViewById(R.id.btn_checkAgain);
19         btn_goto_cbc = findViewById(R.id.btn_goto_cbc);
20         btn_goback_activity = findViewById(R.id.btn_goback_activity);
21
22         Intent intent = getIntent();
23         String result = intent.getStringExtra( name: "result");
24
25         textView.setText(result);
26
27         btn_checkAgain.setOnClickListener(new View.OnClickListener() {
28             @Override
29             public void onClick(View v) {
30
31                 startNewActivity( context: IshiharaResult.this,IshiharaTest.class);
32
33             }
34
35         });
36
37     }
38
39 }
```

PROOF OF WORK

✓ ishiharatest 2nd gen (Deployed at Apr 19, 2023, 4:35:48 AM) URL: <https://ishiharatest-37h3nbeslq-as.a.run.app> ⚡ ⓘ ⓘ

Powered by Cloud Run ⓘ
ishiharatest

METRICS	DETAILS	SOURCE	VARIABLES	TRIGGER	PERMISSIONS	LOGS	TESTING
Runtime: Python 3.9	Entry point: hello_http	Source location: gcf-v2-sources-853429710779-asia-southeast1/ishiharatest/function-source.zip					

[DOWNLOAD ZIP](#)

main.py

```
1 # Import the necessary libraries
2 import numpy as np
3 import functions_framework
4 from google.cloud import storage
5 import pickle
6 import json
7
8 @functions_framework.http
9 def hello_http(request):
10     """HTTP Cloud Function.
11     Args:
12         request (flask.Request): The request object.
13         <https://flask.palletsprojects.com/en/1.1.x/api/#incoming-request-data>
14     Returns:
15         The response text, or any set of values that can be turned into a
16         Response object using `make_response`
17         <https://flask.palletsprojects.com/en/1.1.x/api/#flask.make\_response>.
18     """
19
20     #accessing gc storage bucket
21     storage_client = storage.Client()
22     bucket = storage_client.get_bucket('thirdeyeai')
23     blobmodel = bucket.blob('ishiharalv.pkl')
24
25     blobmodel.download_to_filename('/tmp/ishiharalv.pkl')
26
27     # Load the model from disk
```



A.D. Amarasinghe
IT19133614
Information Technology

RESEARCH PROBLEM

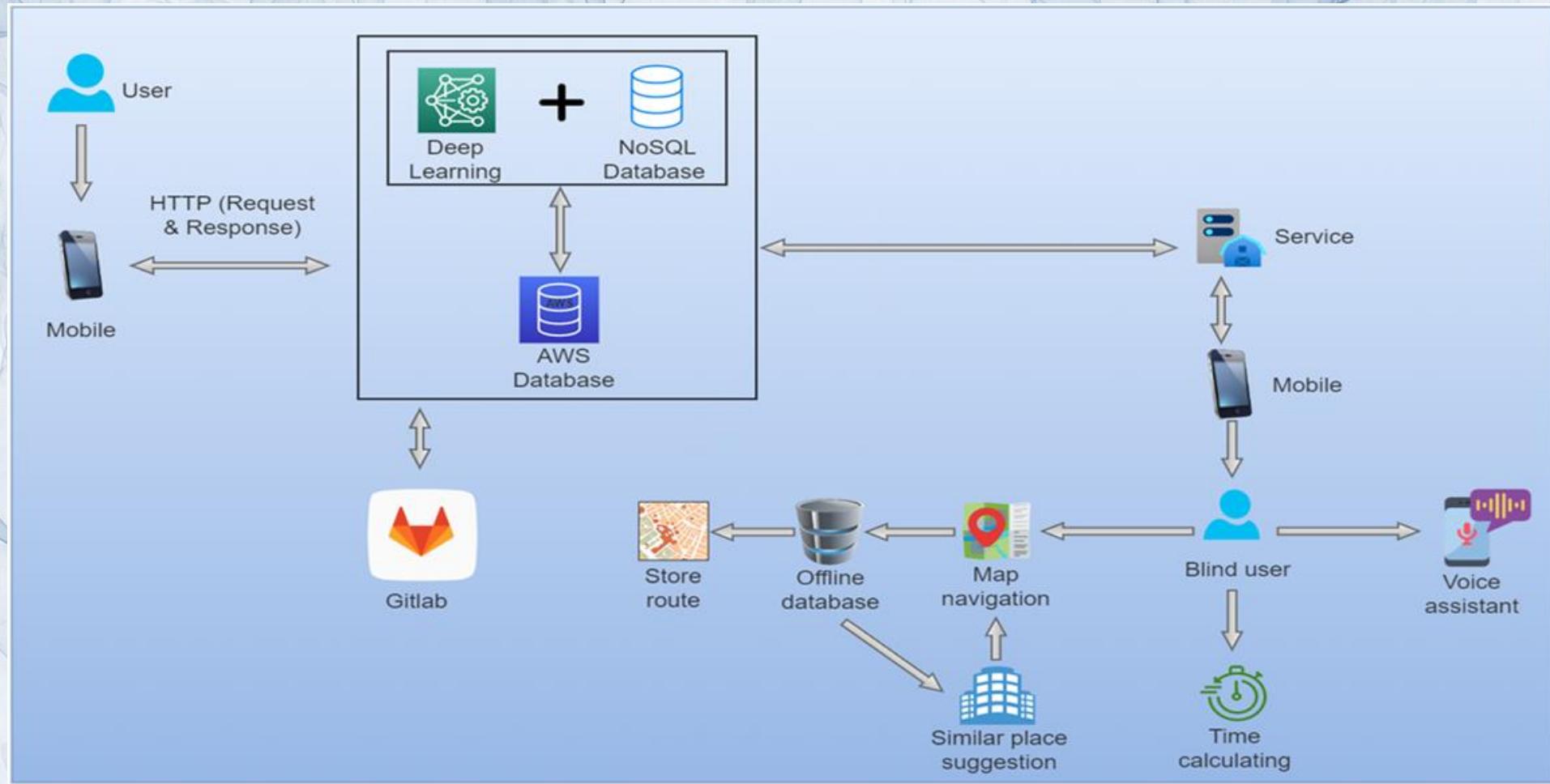
There's no proper mobile application for visually impaired people to help with both indoor and outdoor navigation. And BVI people prefer voice-command-based applications rather than finger taps to accomplish their needs.

PROPOSED SOLUTION

Implement real time navigation with a voice assistant support and suggestions to navigations where the distance can be reduced.

- **Implementation of the voice assistant to be effective, clear, and accurate when it guides the user.**
- **Accurate map navigation for both indoor and outdoor.**
- **Suggest similar destinations and calculate the time to reach the destination.**

COMPONENT OVERVIEW DIAGRAM



CURRENT PROGRESS 100%



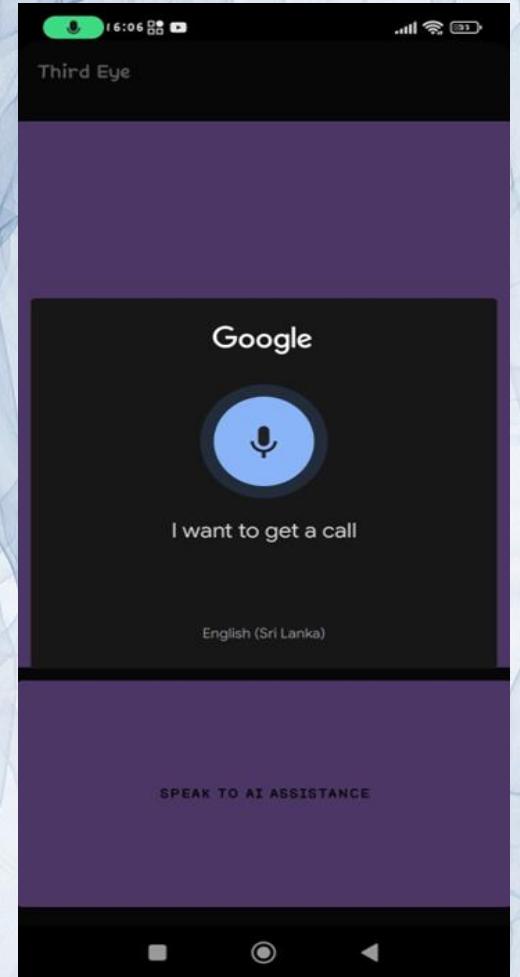
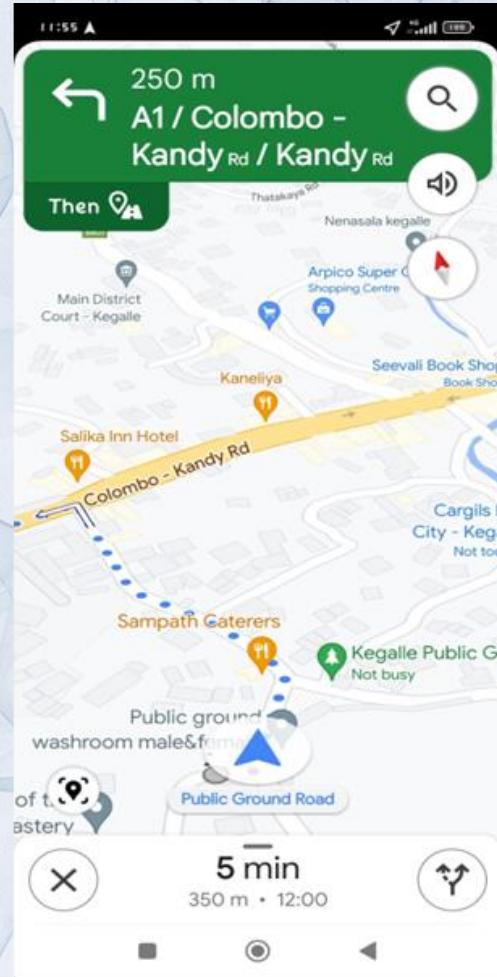
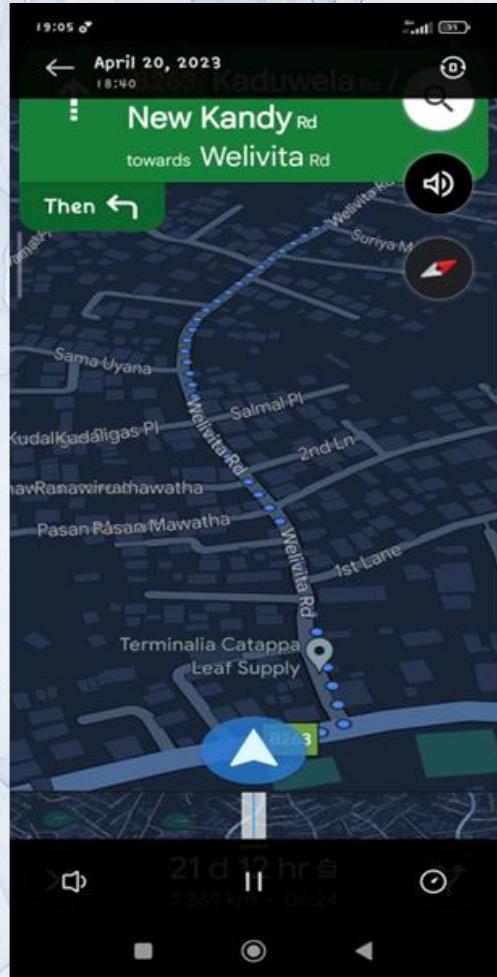
- ✓ Implemented an intelligent chatbot using LSTM deep neural network algorithm which can give random responses for trained questions.
- ✓ Created and trained the dataset in 'JSON' format defining different requests and responses from server to client and vice versa.
- ✓ Created and developed the model in cloud to use it anywhere to reduce the performance lag.
- ✓ Implemented the android mobile application to properly work the text to speech function of the implemented chatbot.
- ✓ Implemented the outdoor navigation function with the AI assistant's support using GPS based map.

CURRENT PROGRESS 100%

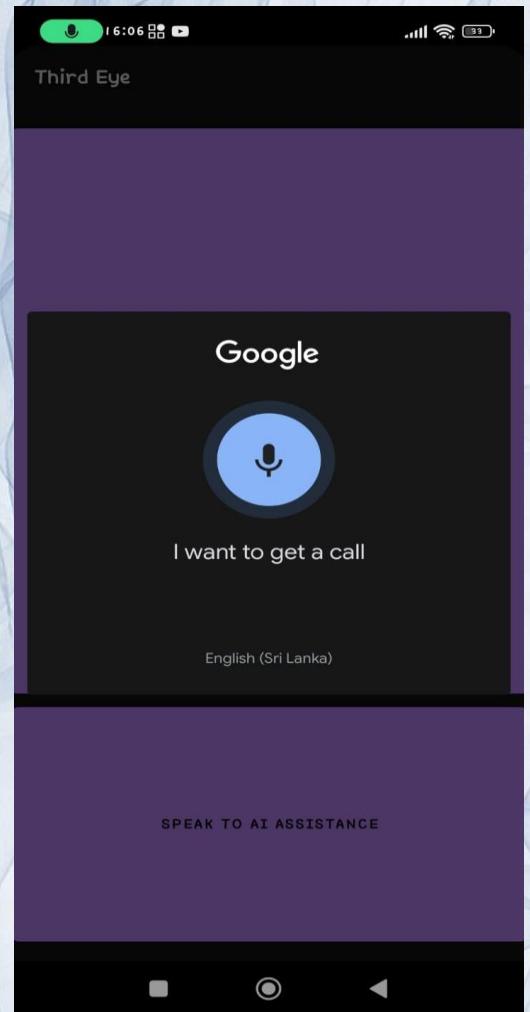
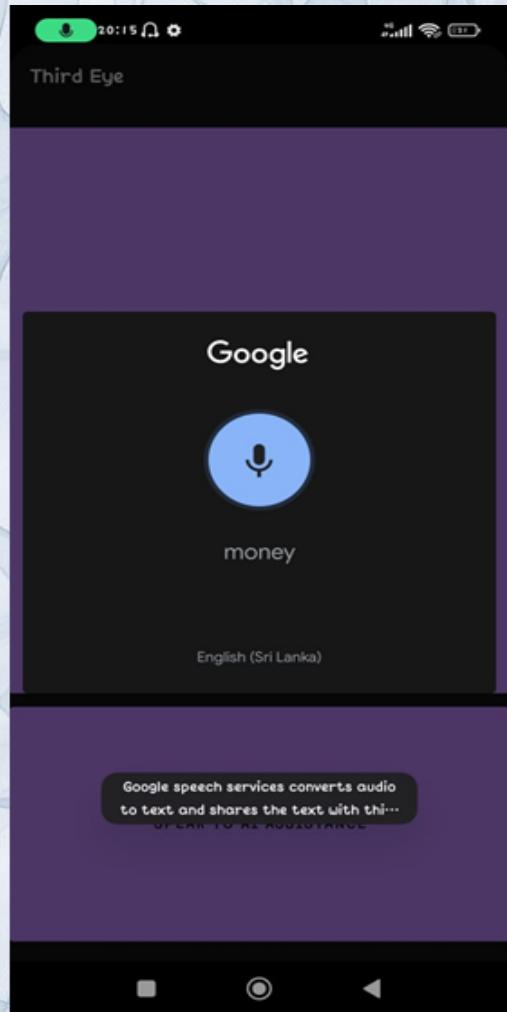
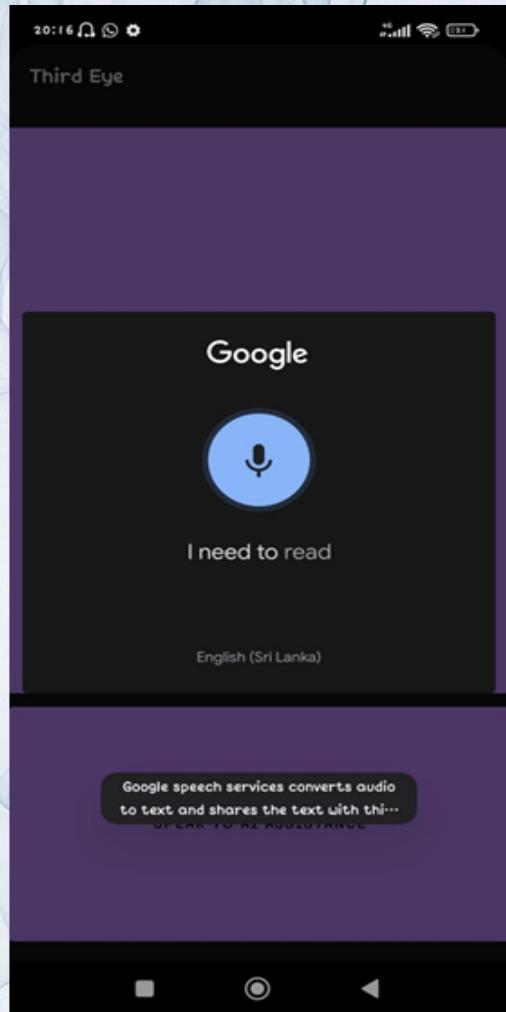
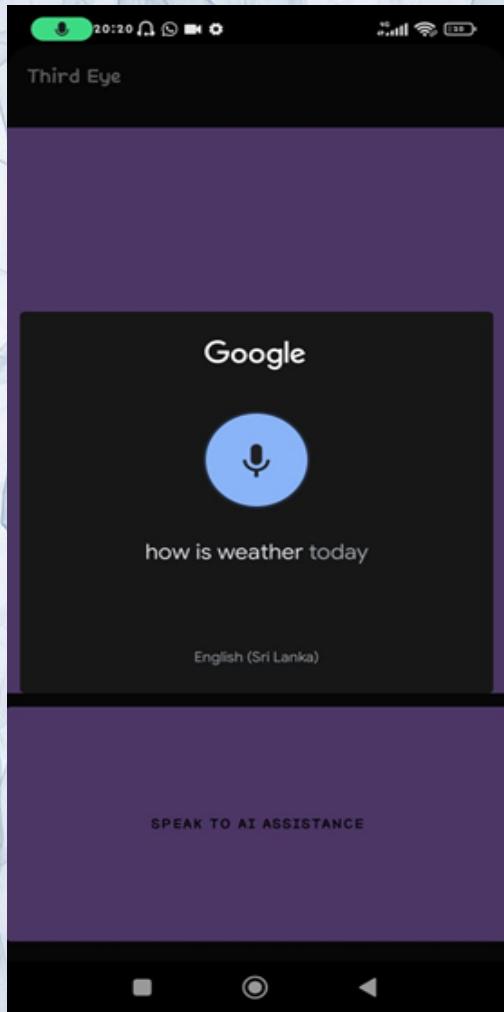


- ✓ Implemented the connectivity between the chatbot and object detection function to work the indoor navigation functionality.
- ✓ Implemented the connectivity between the chatbot and the other components. (text reader, currency reader, weather reader, turn on flashlight).
- ✓ Implemented a functionality to take voice calls in an emergent situation as further implementation.

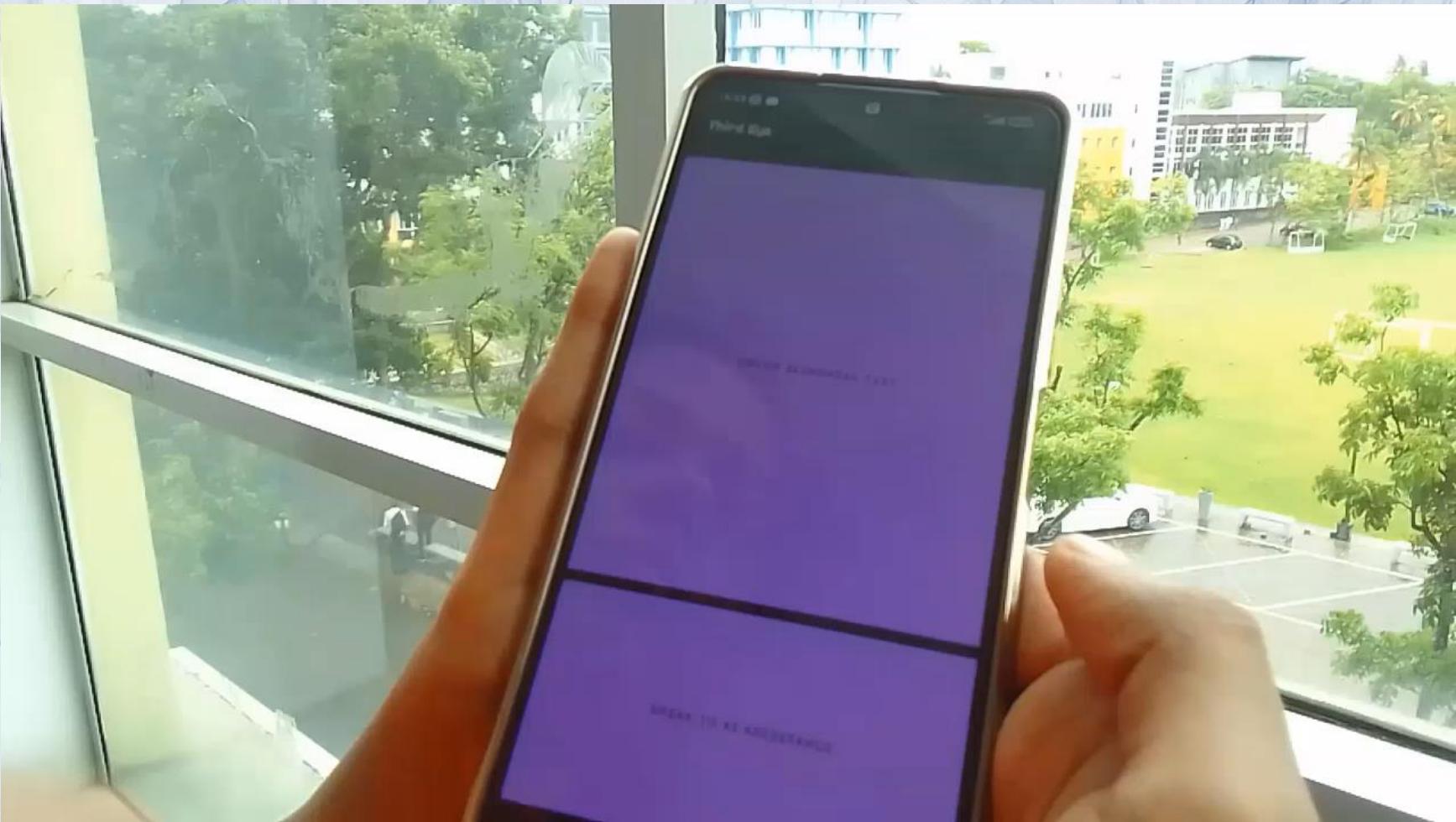
TESTING



TESTING



TESTING



RISK MITIGATION

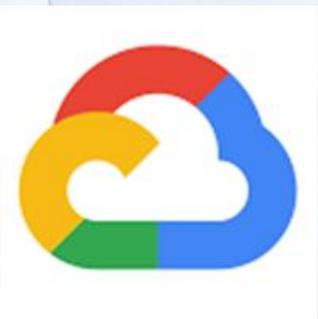
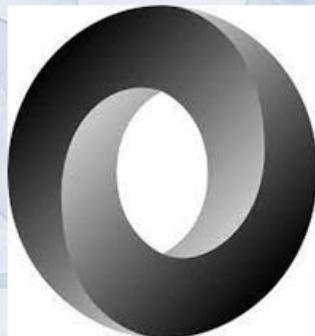
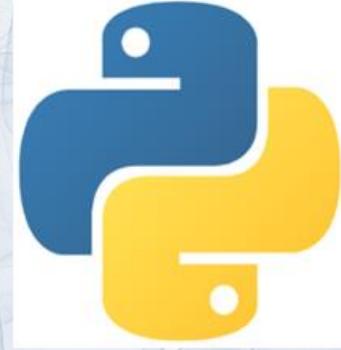
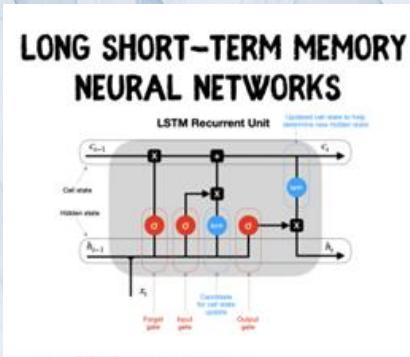
- Indoor navigation is impossible to do practically because of the signal block, and weakness in indoor areas.
- And, without entering a floor plan or a blueprint to the system, an application can't navigate the user to a destination.
- Augmented Reality based options give a partial solution, but it is useless to a blind person because he can't see the directions displaying in the screen.
- The solution is to combine both voice assistant and object detection functions together to do the indoor navigation.
- Because object identifier detect vary range of obstacles in user's walking path.

RISK MITIGATION

- When the voice assistant connected with the object detection component, it informed the user about the detected object in an endless loop.
- So corrected it so that it read as once.
- Joined new features like taking a phone call and turning on a flashlight that can be used for both object detection and text reading in dark environments.

TOOLS & TECHNOLOGIES

- LSTM neural network
- Jupyter notebook
- Anaconda environment
- NLTK tokenizer
- Json
- Google Cloud Platform
- Android Studio
- Java / Kotlin



PROOF OF WORK

```
// function for listing
private void Listen_AIAssistant(int rqcode) {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, "en-LK");
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Start Speaking");
    startActivityForResult(intent, rqcode);
}

// function for getting results
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == 100 && resultCode == RESULT_OK) {
        input = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS).get(0);
        postrequest(input);
        Log.d(TAG, "onActivityResult: " + input);
    } else if (requestCode == 101 && resultCode == RESULT_OK) {
        input = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS).get(0);
        open_navigation(input);
    }
}
```

```
// function for http post request by using volley
private void postrequest(String RQmsg) {

    String url = "https://thirdeye-37h3nbelsq-el.a.run.app";

    RequestQueue queue = newRequestQueue(this);

    JsonObjectRequest request = new JsonObjectRequest(Request.Method.POST, url, null, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {

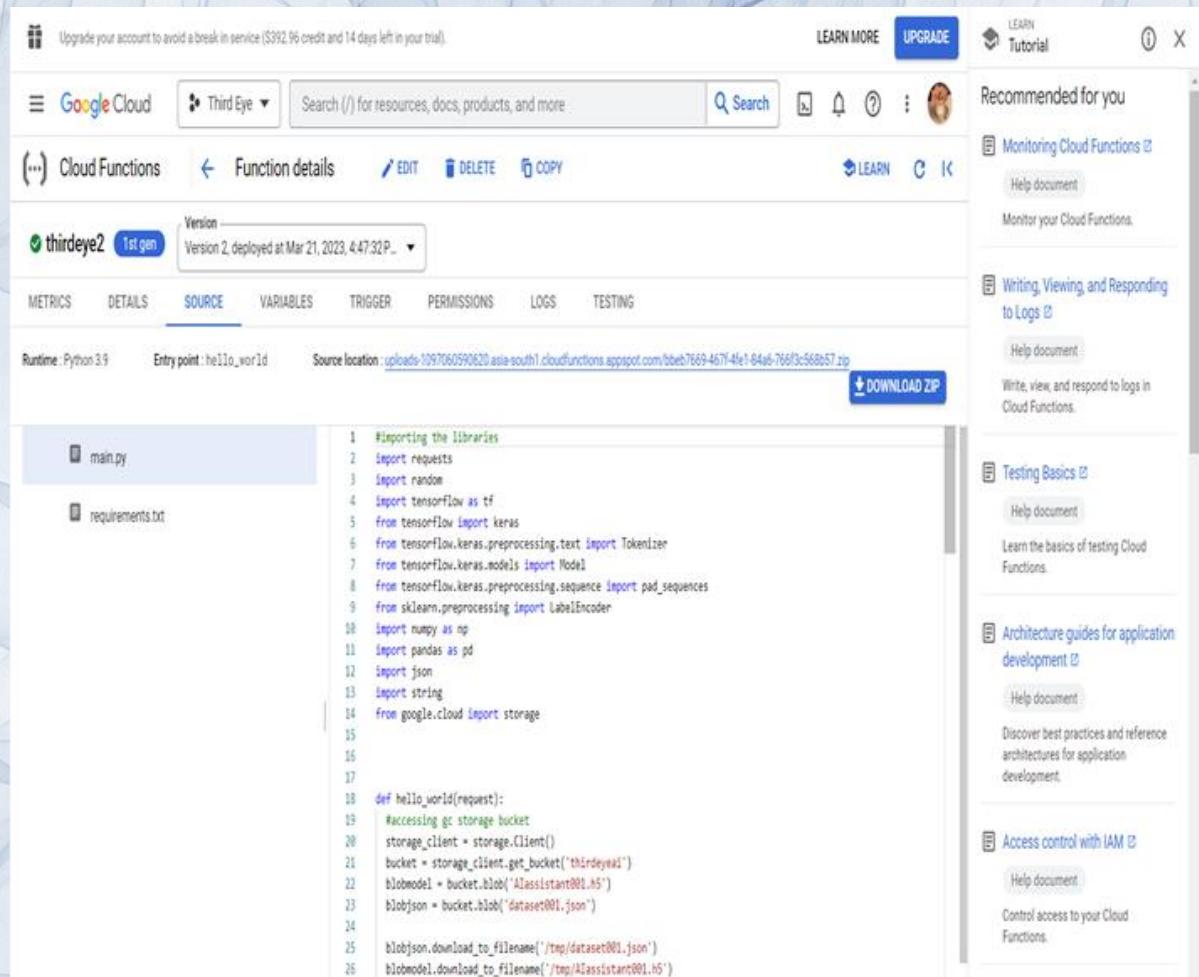
            try {
                resmsg = response.getString("msg");
                AI_Brain(resmsg);

            } catch (JSONException e) {

                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            resmsg = "sorry network error occurs. ";
            AI_Brain(resmsg);
        }
    }) {
        @Override
        public String getBodyContentType() {
            return "application/json; charset=utf-8";
        }
    }
}
```

PROOF OF WORK

```
private void AI_Brain(String airesmsg) {  
  
    // switch cases are used to check responses by AI and get decisions to execute functions.  
    switch (airesmsg) {  
        case "sure. Ishihara test opening":  
        case "Here is isihihara test":  
            Speak_AIAssistant(airesmsg);  
            break;  
        case "sure. Where do u want to go?":  
            Speak_AIAssistant(airesmsg);  
            handler.postDelayed(new Runnable() {  
                @Override  
                public void run() {  
                    Listen_for_place();  
                }  
            }, 3000);  
  
            break;  
        case "sure. please show the page to mobile camera that you want to read":  
            Speak_AIAssistant(airesmsg);  
            open_Ocr();  
            break;  
        case "sure. currency reader opening. please show the money to camera one by one ":  
            Speak_AIAssistant(airesmsg);  
            open_CurrencyDetection();  
            break;  
        case "Opening Third eye object detection":  
            Speak_AIAssistant(airesmsg);  
            open_DistanceEstimation();  
            break;  
        case "Opening weather":  
            Speak_AIAssistant(weather);  
            break;  
        default:  
            Speak_AIAssistant(airesmsg);  
    }  
}
```



The screenshot shows the Google Cloud Functions interface. The top navigation bar includes 'Google Cloud', 'Third Eye', 'Search (/) for resources, docs, products, and more', and a user profile icon. The main title is 'Function details' for 'thirdeye2' (1st gen). Below it, the version is listed as 'Version 2, deployed at Mar 21, 2023, 4:47:32 P...'. The interface has tabs for METRICS, DETAILS, SOURCE, VARIABLES, TRIGGER, PERMISSIONS, LOGS, and TESTING. The SOURCE tab is selected, showing the runtime as Python 3.9 and the entry point as 'hello_world'. The source location is 'uploads:1097060590820.asia-south1.cloudfunctions.appspot.com/bbeb7669-4674-4e1-84a6-766f3c568b57.zip'. A 'DOWNLOAD ZIP' button is available. The code editor displays two files: 'main.py' and 'requirements.txt'. The 'main.py' file contains the following Python code:

```
1 #Importing the libraries  
2 import requests  
3 import random  
4 import tensorflow as tf  
5 from tensorflow import keras  
6 from tensorflow.keras.preprocessing.text import Tokenizer  
7 from tensorflow.keras.models import Model  
8 from tensorflow.keras.preprocessing.sequence import pad_sequences  
9 from sklearn.preprocessing import LabelEncoder  
10 import numpy as np  
11 import pandas as pd  
12 import json  
13 import string  
14 from google.cloud import storage  
15  
16  
17  
18 def hello_world(request):  
19     #Accessing gc storage bucket  
20     storage_client = storage.Client()  
21     bucket = storage_client.get_bucket("thirdeyeai")  
22     blobmodel = bucket.blob("AIassistant001.5")  
23     blobjson = bucket.blob("dataset001.json")  
24  
25     blobjson.download_to_filename('/tmp/dataset001.json')  
26     blobmodel.download_to_filename('/tmp/AIassistant001.5')
```



K.D.J. Jayasekara
IT19103600
Information Technology

RESEARCH PROBLEM

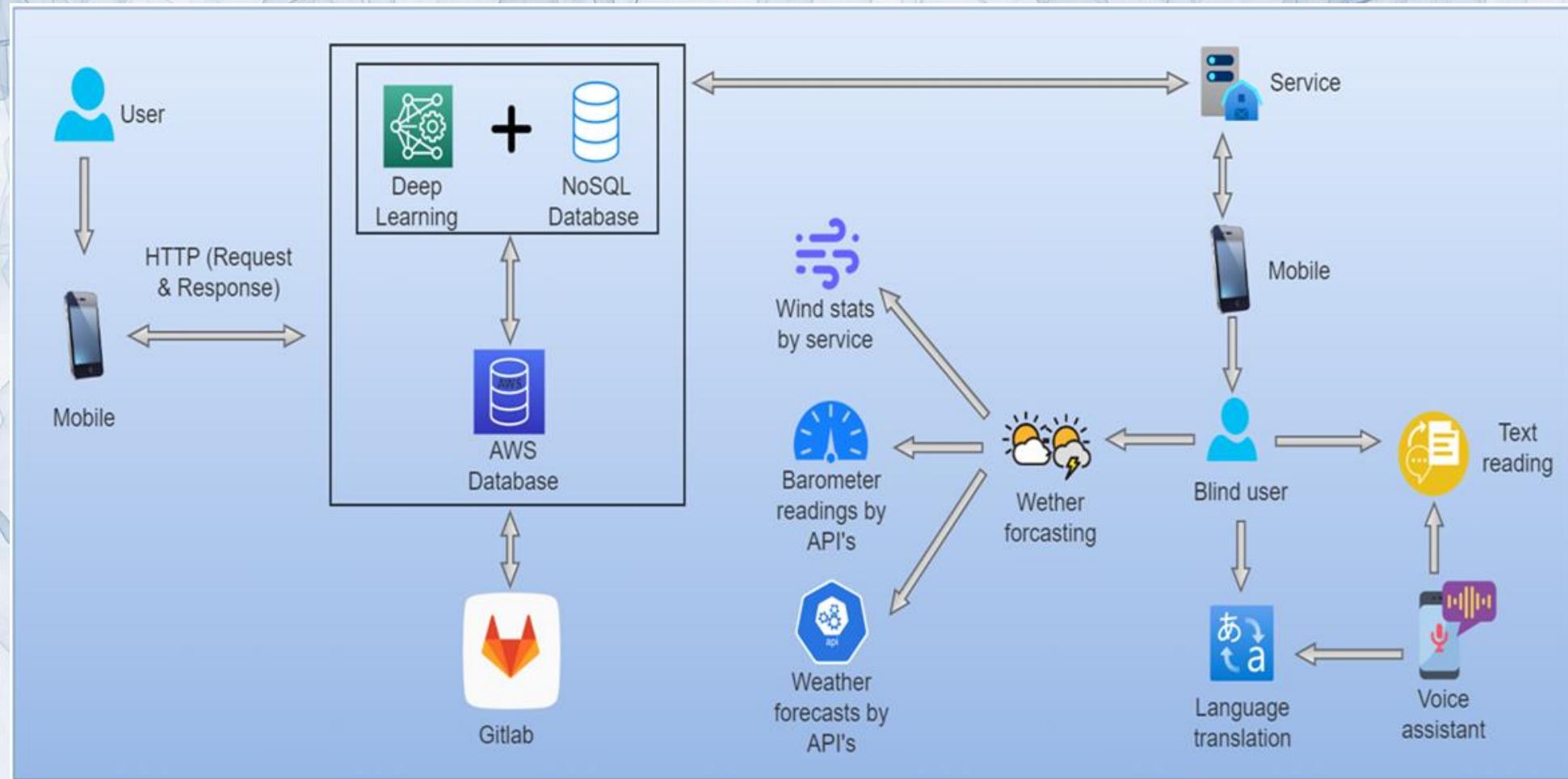
- Aiding visually impaired people to identifying currency notes and coins accurately real-time.
- Training a machine learning to identify the accurate outcome.
- Creating a system that gives accurate weather prediction and text recognition output via an AI voice assistance.
- Creating a mobile application that supports text recognition, currency detection and weather prediction.

PROPOSED SOLUTION

Implement real time mobile application to be assistive for blind people by implementing a text reader, weather prediction and currency reader.

- **Implementing real time text reading and translating functionalities**
- **Implementing a currency notes and coins identifier**
- **A reliable and accurate weather forecasting system**

COMPONENT OVERVIEW DIAGRAM

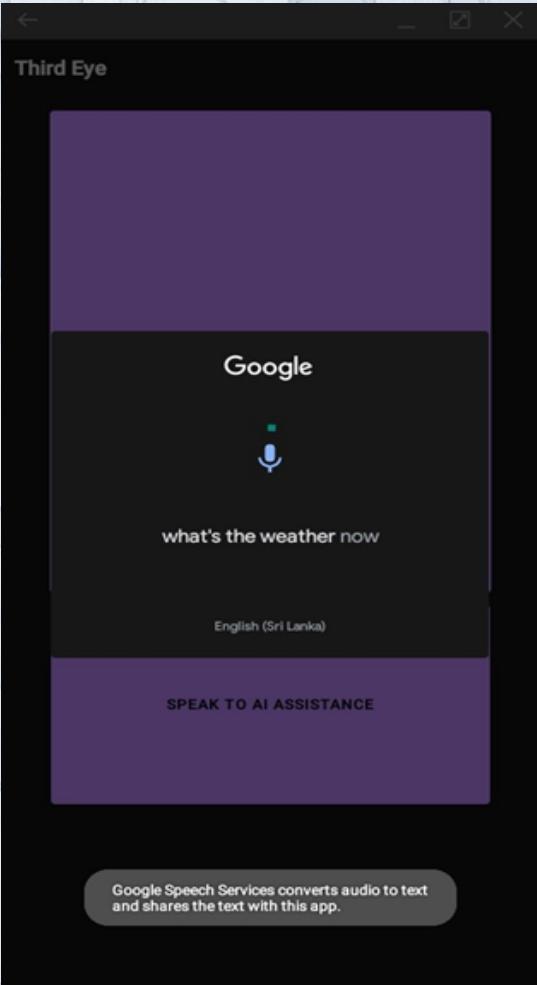
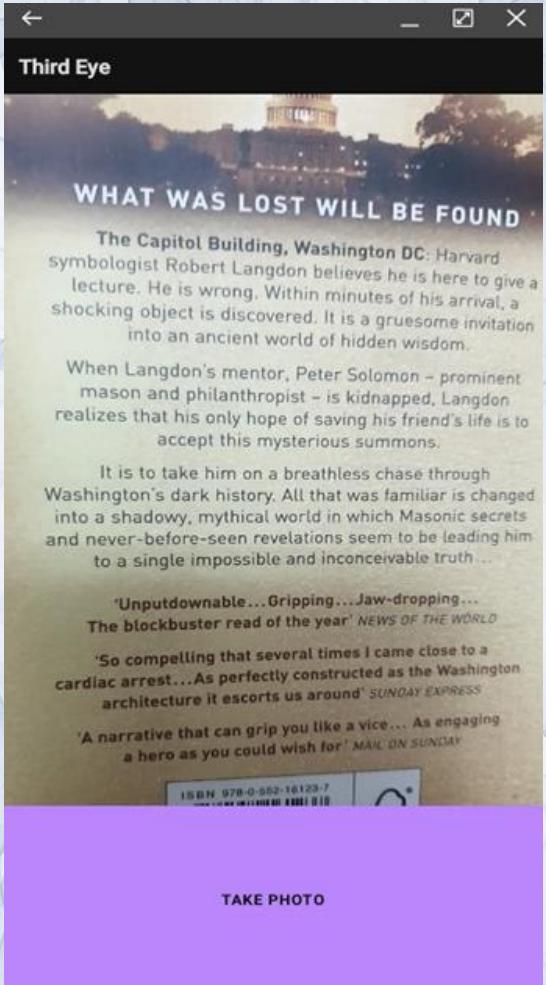


CURRENT PROGRESS 100%

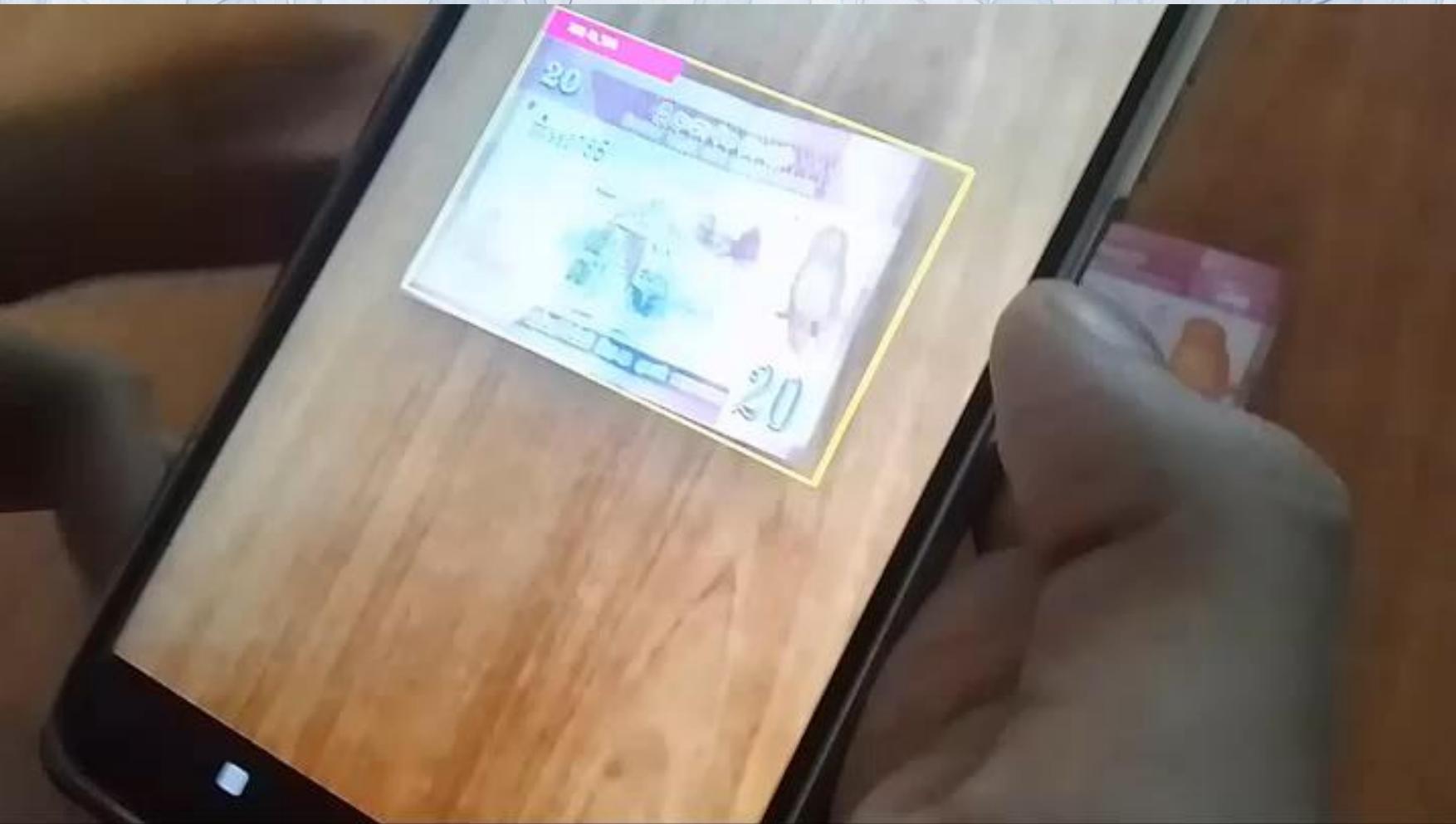


- ✓ Collected coins and notes images with different angles for each note and coin to be later be used as the trained-test split in the model and train the model with test images.
- ✓ Custom YOLOV5 model which uses compound detection model to train the custom data set to identify given notes and coins in real-time.
- ✓ Convert custom model to Pytorch version to train and deploy model to mobile device.
- ✓ Implementing and training an OCR model to identify text on any source with the support of the best pre trained customized text recognition model in Google ML kit.
- ✓ Implemented a weather forecasting system with the support of a weather API.
- ✓ Using Pytorch light and Google ML kit to implemented the model to a mobile application.
- ✓ Integrated the currency recognition to the main application.

TESTING



TESTING

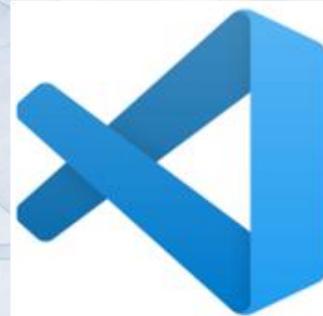


RISK MITIGATION

- To implement considering which ML method to get results when identifying currency notes.
- Using YOLOV4 training model with custom made data set of images.
- Using YOLOV5 instead of YOLOV4 to enhance performs.
- Inaccurate coin identification, therefore expanding data set and different model for coins.

TOOLS & TECHNOLOGIES

- YOLO5
- Pytorch
- Kotlin
- VS Code
- Java / Kotlin
- Android Studio
- Multi class object detection
- Optical Character Recognizer

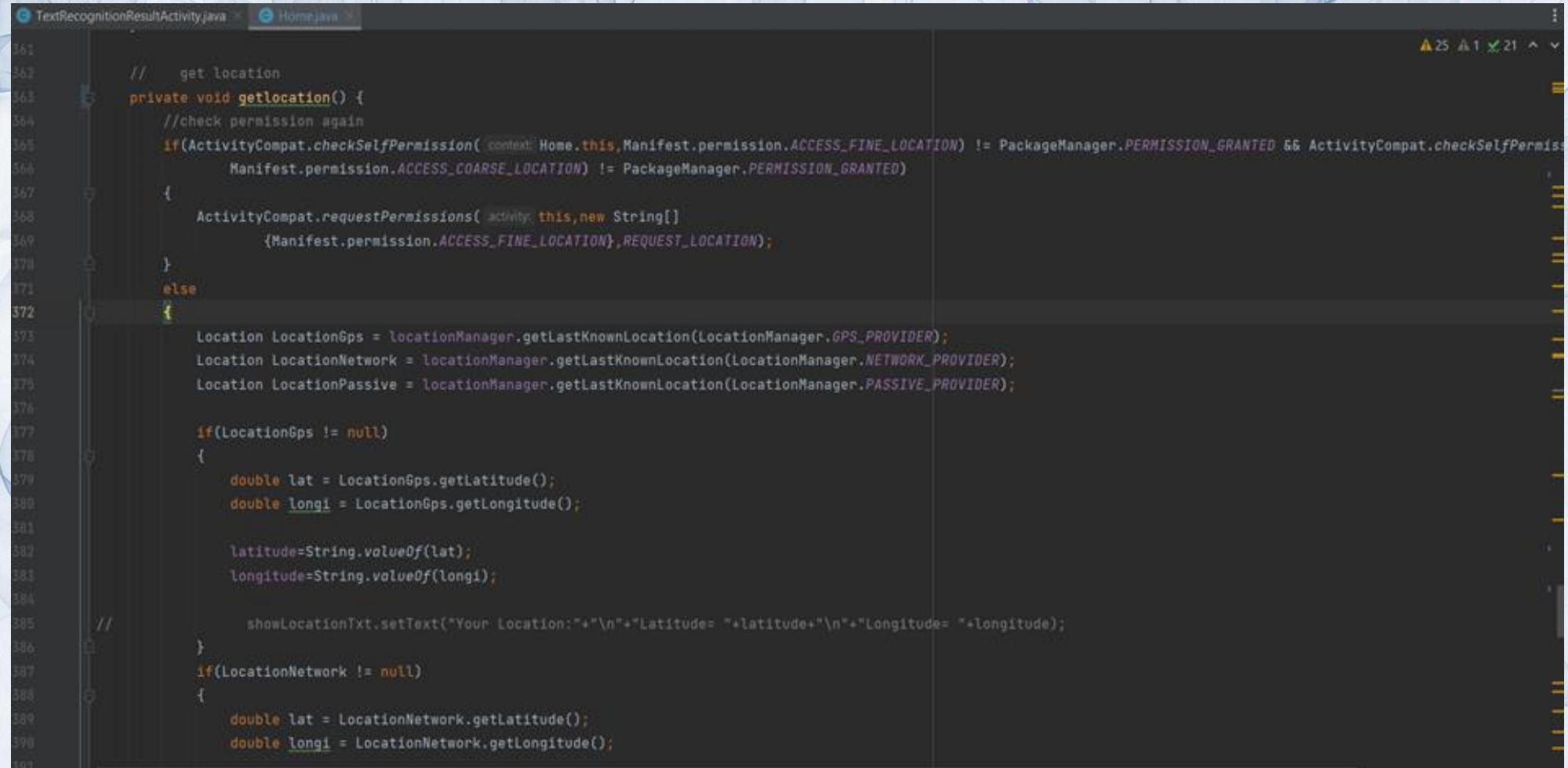


PROOF OF WORK

```
TextRecognitionResultActivity.java
66
67     textRecognizer = TextRecognition.getClient(TextRecognizerOptions.DEFAULT_OPTIONS);
68
69     btnCheckagain = (Button) findViewById(R.id.btnCheckagain);
70
71     btnCheckagain.setOnClickListener(new View.OnClickListener() {
72
73         @Override
74         public void onClick(View v) {
75             Intent intent = new Intent(getApplicationContext(),TextRecognitionCameraActivity.class);
76             startActivity(intent);
77         }
78     });
79
80     try{
81
82         piclink = getIntent().getStringExtra("piclink");
83
84         Uri.parse("content://media/*piclink");
85         String[] projection = { MediaStore.Images.Media.DATA };
86         Cursor cursor = getContentResolver().query(imageUri, projection, selection: null, selectionArgs: null, sortOrder: null);
87
88         if (cursor != null && cursor.moveToFirst()) {
89             int columnIndex = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
90             String imagePath = cursor.getString(columnIndex);
91
92             if (imagePath != null && !imagePath.isEmpty()) {
93                 Bitmap bitmap2 = BitmapFactory.decodeFile(imagePath);
94                 imageView.setImageBitmap(bitmap2);
95             }
96
97         }
98
99         cursor.close();
100    }
```

```
TextRecognitionResultActivity.java
88
89     @Override
90     @WorkerThread
91     @Nullable
92     protected AnalysisResult analyzeImage(ImageProxy image, int rotationDegrees) {
93
94         final ArrayList<Result> results;
95
96         try {
97             if (mModule == null) {
98                 mModule = LiteModuleLoader.load(Home.assetFilePath(getApplicationContext(), assetName: "money4best.torchscript.ptl"));
99             }
100
101             Bitmap bitmag = imgToBitmap(image.getImage());
102             Matrix matrix = new Matrix();
103             matrix.postRotate( degrees: 90.0f );
104             bitmap = Bitmap.createBitmap(bitmag, 0, 0, bitmag.getWidth(), bitmag.getHeight(), matrix, false: true);
105             Bitmap resizedBitmap = Bitmap.createScaledBitmap(bitmap, PrePostProcessor.mInputWidth, PrePostProcessor.mInputHeight, false: true);
106
107             final Tensor inputTensor = TensorImageUtils.bitmapToFloat32Tensor(resizedBitmap, PrePostProcessor.NO_MEAN_RGB, PrePostProcessor.NO_STD_RGB);
108             IValue[] outputTuple = mModule.forward(IValue.from(inputTensor)).toTuple();
109             IValue[] outputtuple = new IValue[ ] { IValue.from(inputTensor) };
110             final Tensor outputTensor = outputTuple[0].toTensor();
111             final float[] outputs = outputTensor.getDataAsFloatArray();
112
113             float imgScaleX = (float) bitmap.getWidth() / PrePostProcessor.mInputWidth;
114             float imgScaleY = (float) bitmap.getHeight() / PrePostProcessor.mInputHeight;
115             float ivScaleX = (float) mResultView.getWidth() / bitmap.getWidth();
116             float ivScaleY = (float) mResultView.getHeight() / bitmap.getHeight();
117
118             results = PrePostProcessor.outputsToNMSPredictions(outputs, imgScaleX, imgScaleY, ivScaleX, ivScaleY, mark: 0, start: 0);
119         }
```

PROOF OF WORK



The screenshot shows an Android Studio code editor with the file `TextRecognitionResultActivity.java` open. The code is written in Java and handles location retrieval. It includes logic to check for location permissions, request them if necessary, and then retrieve the last known location from GPS, Network, and Passive providers. The retrieved latitude and longitude are then displayed in a text view.

```
361  
362     //  get location  
363     private void getLocation() {  
364         //check permission again  
365         if(ActivityCompat.checkSelfPermission(context: Home.this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(context: Home.this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)  
366         {  
367             ActivityCompat.requestPermissions(activity: this, new String[]  
368                 {Manifest.permission.ACCESS_FINE_LOCATION}, REQUEST_LOCATION);  
369         }  
370         else  
371         {  
372             Location LocationGps = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);  
373             Location LocationNetwork = locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);  
374             Location LocationPassive = locationManager.getLastKnownLocation(LocationManager.PASSIVE_PROVIDER);  
375  
376             if(LocationGps != null)  
377             {  
378                 double lat = LocationGps.getLatitude();  
379                 double longi = LocationGps.getLongitude();  
380  
381                 latitude=String.valueOf(lat);  
382                 longitude=String.valueOf(longi);  
383  
384                 showLocationTxt.setText("Your Location:"+"\n"+ "Latitude= "+latitude+"\n"+ "Longitude= "+longitude);  
385             }  
386             if(LocationNetwork != null)  
387             {  
388                 double lat = LocationNetwork.getLatitude();  
389                 double longi = LocationNetwork.getLongitude();  
390             }  
391         }  
392     }  
393 }
```



A.M.N.D. Aththanayake
IT19981918
Information Technology

RESEARCH PROBLEM

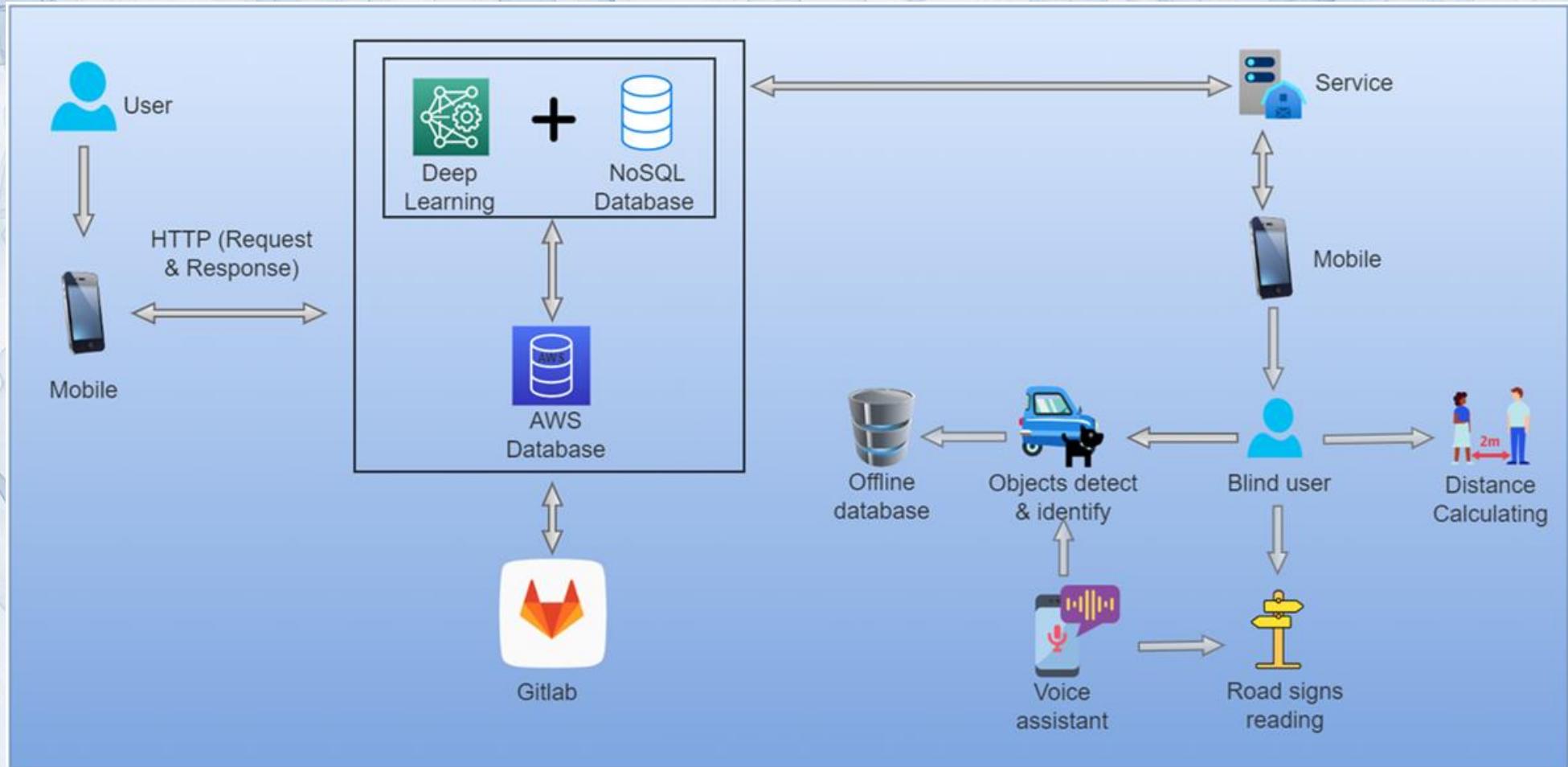
Existing mobile apps for object detection for BVI's provide limited information about the captured objects, such as an alert sound. It is much needed to know what the app detected and how far is it located to be more prepare before an accident happens.

PROPOSED SOLUTION

Implement object identifying when navigating through the journey to assist with obstacles as well as distances calculating to forecast collision detection.

- Detect and recognize in a vary range of objects.
- Calculate distance between the user and obstacle.
- Work properly in different background conditions.

COMPONENT OVERVIEW DIAGRAM

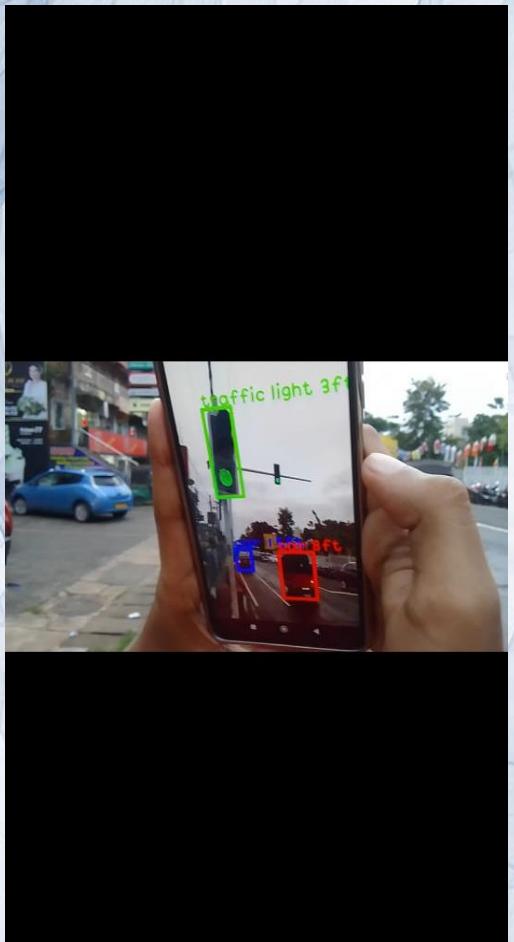
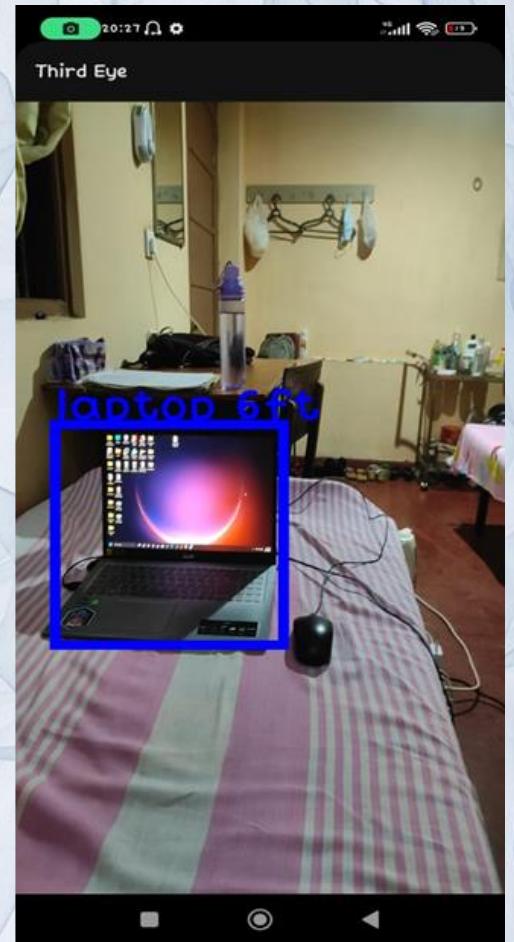
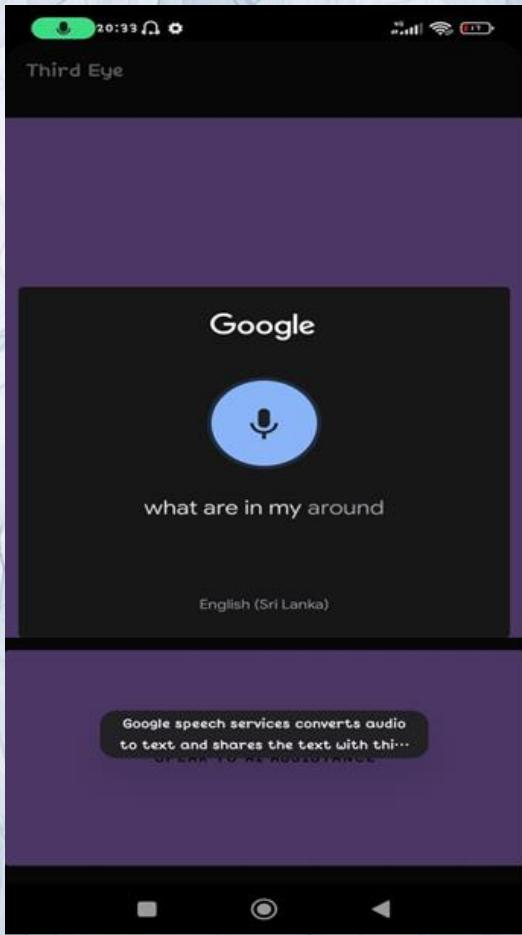


CURRENT PROGRESS 100%

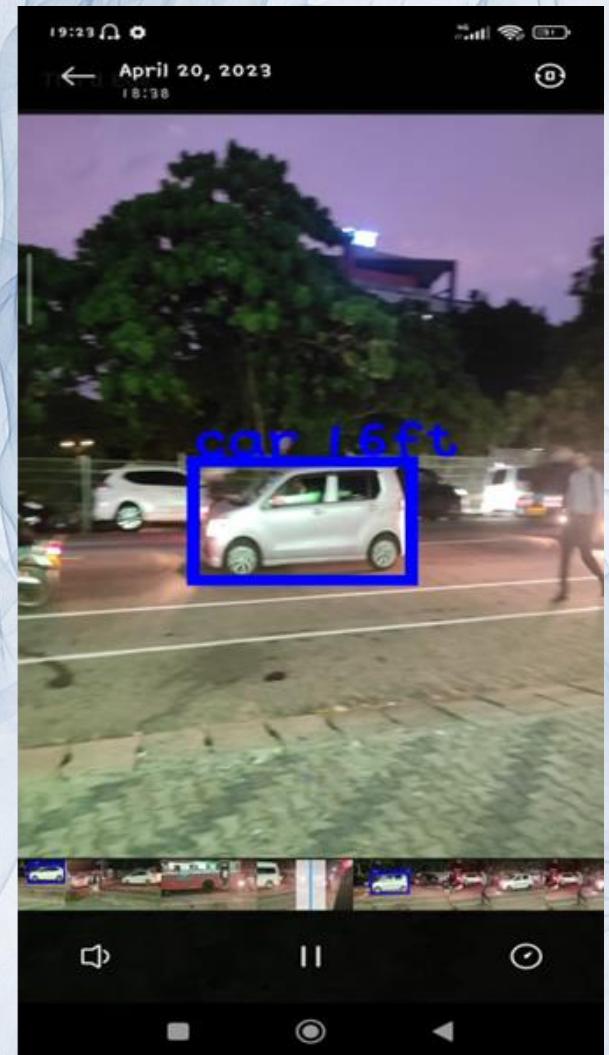


- ✓ Implemented a model to detect and identify objects with the accuracy value.
- ✓ Implemented a formula using focal length and actual length to calculate the distance to a person, frame by frame with real time capturing.
- ✓ The model has trained to identify vary range of objects.
- ✓ Improved the formula to calculate the distance for other objects.
- ✓ Implemented the model to detect objects even in low light conditions.
- ✓ Improved the accuracy of the model by modifying the formula.
- ✓ Improved the formula that can calculate the distance for more objects.
- ✓ Tested the function with the users.

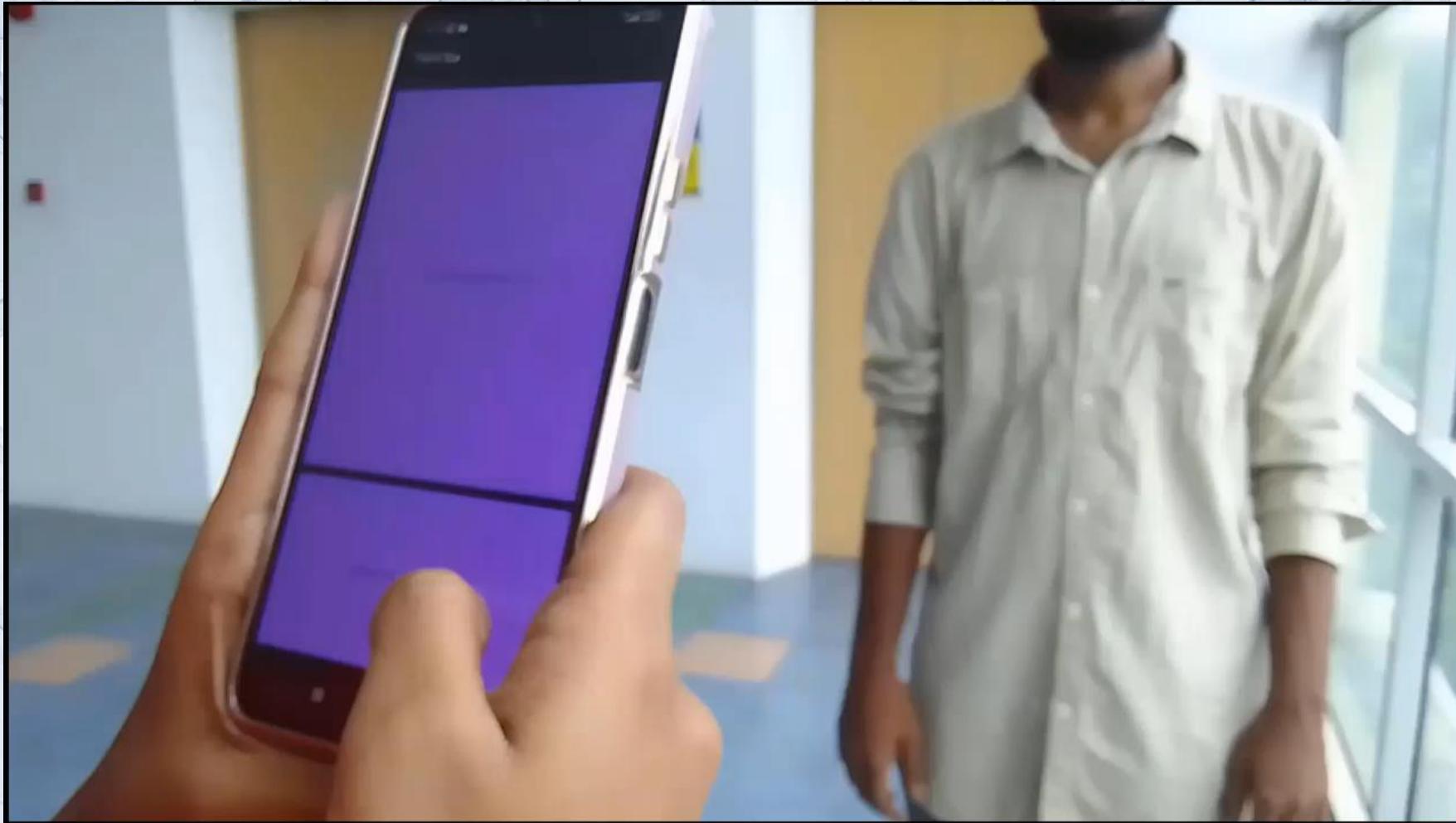
TESTING



TESTING



TESTING

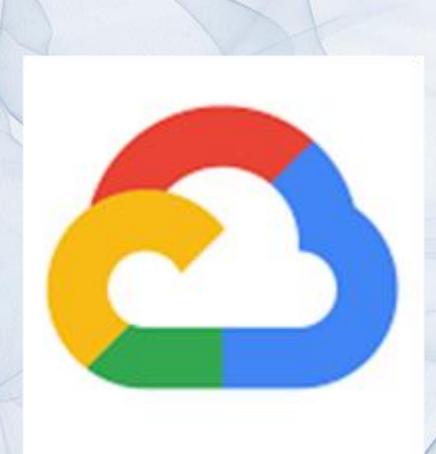
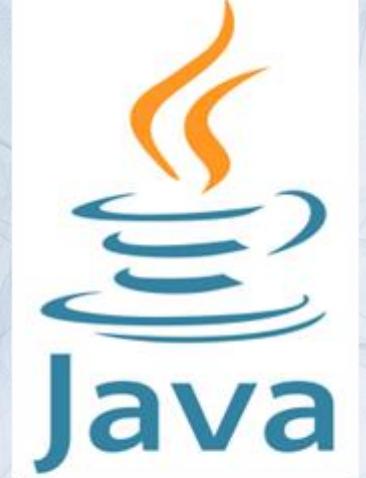
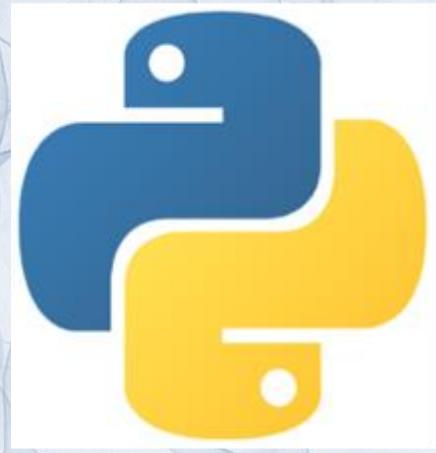


RISK MITIGATION

- Normally distance to an object can't be calculated by trigonometry only using the camera lens. It needed external sensors or IoT devices.
- So instead of the proposed trigonometric formula, had to use 'Focal Length' formula to calculate the distance considering the camera lens, and reflection of the object.
- Calculation of the distance frame-by-frame can be endless loop to the voice assistant and to the user.
- When informing the user of the detected object, it works as an endless loop that distracts the listener.
- To avoid it, the second option was to change the model to notify the detected object only once.

TOOLS & TECHNOLOGIES

- YOLO4 / YOLO5
- TensorFlow tflite
- Transfer Learning
- Multi class object detection
- OpenCV
- Google Cloud Platform
- PyCharm
- Android Studio
- Java / Kotlin



PROOF OF WORK

```
private lateinit var tts: TextToSpeech

lateinit var labels: List<String>

var colors = listOf<Int>(
    Color.BLUE,
    Color.GREEN,
    Color.RED,
    Color.CYAN,
    Color.GRAY,
    Color.BLACK,
    Color.DKGRAY,
    Color.MAGENTA,
    Color.YELLOW
)

val paint = Paint()
lateinit var imageProcessor: ImageProcessor
lateinit var bitmap: Bitmap
lateinit var imageView: ImageView
lateinit var cameraDevice: CameraDevice
lateinit var handler: Handler
lateinit var cameraManager: CameraManager
lateinit var textureView: TextureView
lateinit var model: SsdMobileNetV1Metadata1

// Set camera parameters
private val focallength = 26.0 // Replace with your camera's focal length
private val imageSensorWidth =
    6.0 // Replace with your camera's image sensor width in millimeters
```

```
textureView.surfaceTextureListener = object : TextureView.SurfaceTextureListener {
    override fun onSurfaceTextureAvailable(p0: SurfaceTexture, p1: Int, p2: Int) {
        open_camera()
    }

    override fun onSurfaceTextureSizeChanged(p0: SurfaceTexture, p1: Int, p2: Int) {
        // TODO("Not yet implemented")
    }

    override fun onSurfaceTextureDestroyed(p0: SurfaceTexture): Boolean {
        return false
    }

    override fun onSurfaceTextureUpdated(p0: SurfaceTexture) {
        bitmap = textureView.bitmap!!
        // Creates inputs for reference.
        var image = TensorImage.fromBitmap(bitmap)
        image = imageProcessor.process(image)

        // Runs model inference and gets result.
        val outputs = model.process(image)
        val locations = outputs.locationsAsTensorBuffer.floatArray
        val classes = outputs.classesAsTensorBuffer.floatArray
        val scores = outputs.scoresAsTensorBuffer.floatArray
        val numberofDetections = outputs.numberofDetectionsAsTensorBuffer.floatArray

        var mutable = bitmap.copy(Bitmap.Config.ARGB_8888, true)
        val canvas = Canvas(mutable)

        val h = mutable.height
        val w = mutable.width
```

PROOF OF WORK

```
scores.forEachIndexed { index, fl ->
    x = index
    x *= 4

    if (fl > 0.68) {
        paint.setColor(colors.get(index))
        paint.style = Paint.Style.STROKE
        canvas.drawRect(
            RectF(
                locations.get(x + 1) * w,
                locations.get(x) * h,
                locations.get(x + 3) * w,
                locations.get(x + 2) * h
            ), paint
        )
    }

    val boundingBoxCenterX = (locations.get(x + 1) * w) + (300f / 2)

    // Calculate distance to camera

    if (labels.get(classes.get(index).toInt()) == "person") {
        var distance = calculateDistanceToCamera( objectWidth: 305f, boundingBoxCenterX, imageWidth: 300)
        val distanceInInt = (distance / 305).toInt()

        paint.style = Paint.Style.FILL
        canvas.drawText(
            labels.get(

```

```
) + " " + distanceInInt.toString() + "ft",
                locations.get(x + 1) * w,
                locations.get(x) * h,
                paint
            )
        speakAIAssistant(
            text: labels.get(
                classes.get(index).toInt()
            ) + "is" + distanceInInt.toString() + "ft in front of you"
        )
    }

    } else if(labels.get(classes.get(index).toInt()) == "bicycle"){

        var distance = calculateDistanceToCamera( objectWidth: 500f, boundingBoxCenterX, imageWidth: 300)
        val distanceInInt = (distance / 305).toInt()

        paint.style = Paint.Style.FILL
        canvas.drawText(
            labels.get(
                classes.get(index).toInt()
            ) + " " + distanceInInt.toString() + "ft",

```

PRODUCT REQUIREMENTS

NON-FUNCTIONAL REQUIREMENTS

- Availability
- Reliability
- Usability
- Accuracy
- Speed
- Performance

SYSTEM REQUIREMENTS

- Smartphone should have at least a 20MP back camera.
- Proposed mobile app should work on Android platform.
- Android version 8 is required.

USER REQUIREMENTS

- Users should be able to maintain trust in the accuracy of data.
- The system should be supportive according to the user's demands & requests.
- The app should provide real-time information without any latency.
- The application should be easily handled by the user.

MARKETING PLAN

ONBOARDING FREE

- Voice assistant-based map navigation
- Object detection and identification for limited objects
- Text reading for English language
- Currency notes reader
- Ishihara test and available color filters

MONTHLY
\$14.99/mo

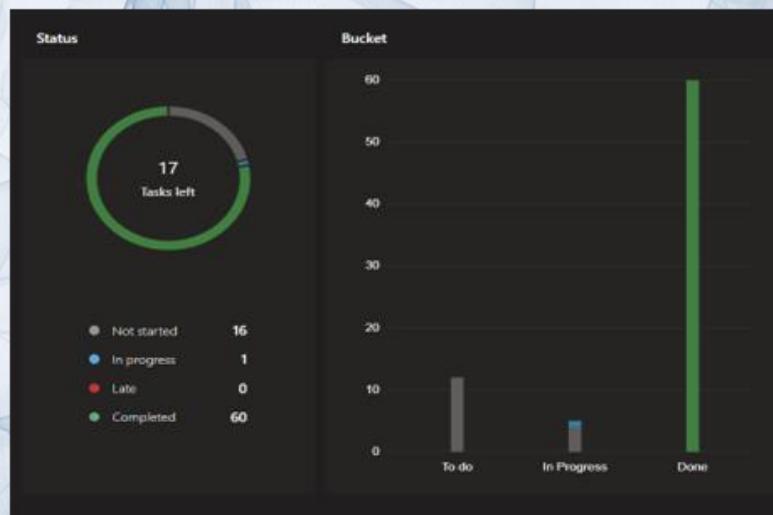
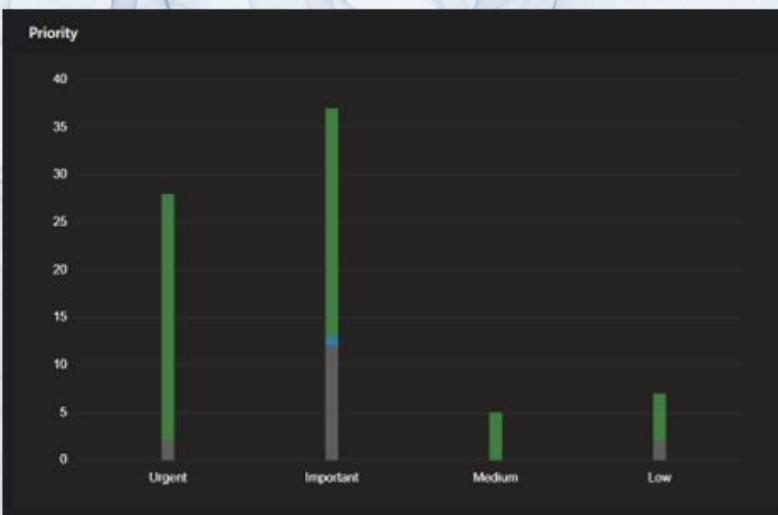
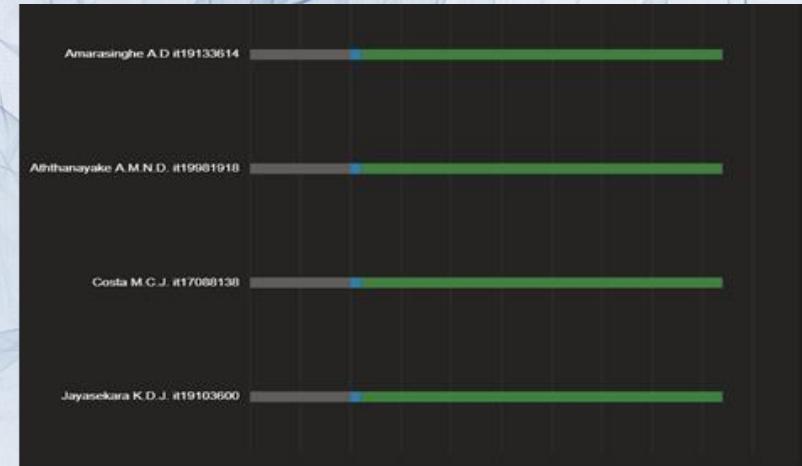
- Sinhala voice assistant
- Text reader to English and Sinhala
- Augmented Reality based color filters

ANNUALY
\$19.99/mo

- Bus and train schedules
- Object detection and identification for unlimited objects

BEST PRACTICES

Task title	Completed by	Priority	Completed %	Bucket
New task	SLIIT	Low	0%	To do
PPV presentation day	Amarasinghe A.D.I...	High	100%	Done
practice for PPV presentation	Amarasinghe A.D.I...	High	100%	Done
create PPV presentation slides	Amarasinghe A.D.I...	High	100%	Done
commit changes to Colab	Amarasinghe A.D.I...	High	100%	Done
design app UI	Amarasinghe A.D.I...	High	100%	Done
integrate the system	Amarasinghe A.D.I...	High	100%	Done
develop the combined mobile application	Amarasinghe A.D.I...	High	100%	Done
convert web system to a mobile app	Amarasinghe A.D.I...	High	100%	Done
convert web system to a mobile app	Amarasinghe A.D.I...	High	100%	Done
implement weather API and OCR model	Amarasinghe A.D.I...	High	100%	Done
improve the currency reading model	Amarasinghe A.D.I...	High	100%	Done
implement the voice assistant with the connectivity of Google map API	Amarasinghe A.D.I...	High	100%	Done
implement the image capturing method to calculate distance	Amarasinghe A.D.I...	High	100%	Done
improve object identifier distance with focal length formula	Amarasinghe A.D.I...	High	100%	Done
improve the voice-assistant model with other components	Amarasinghe A.D.I...	High	100%	Done



Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	
26	27	28	29	30	31	1	publish th...
2	3	4	5	6	7	8	publish the r...
9	10	11	12	13	14	15	publish the r...
16	17	18	19	20	21	22	publish the r...
23	24	25	26	27	28	29	publish the r...
30	1	2	3	4	5	6	finalize the mobile app submit confere...
							submit conference details

BEST PRACTICES

21 Apr, 2023 1 commit

 Merge branch 'general/develop_2' into 'master' [...](#)
Chathuranga Costa authored 1 hour ago

20 Apr, 2023 28 commits

 Merge branch 'dev/nimesha' into 'general/develop_2' [...](#)
Chathuranga Costa authored 11 hours ago

 updated labels and all objects
nimesha00 authored 12 hours ago

 DistanceEstimation activity
nimesha00 authored 12 hours ago

 object detection file added
nimesha00 authored 12 hours ago

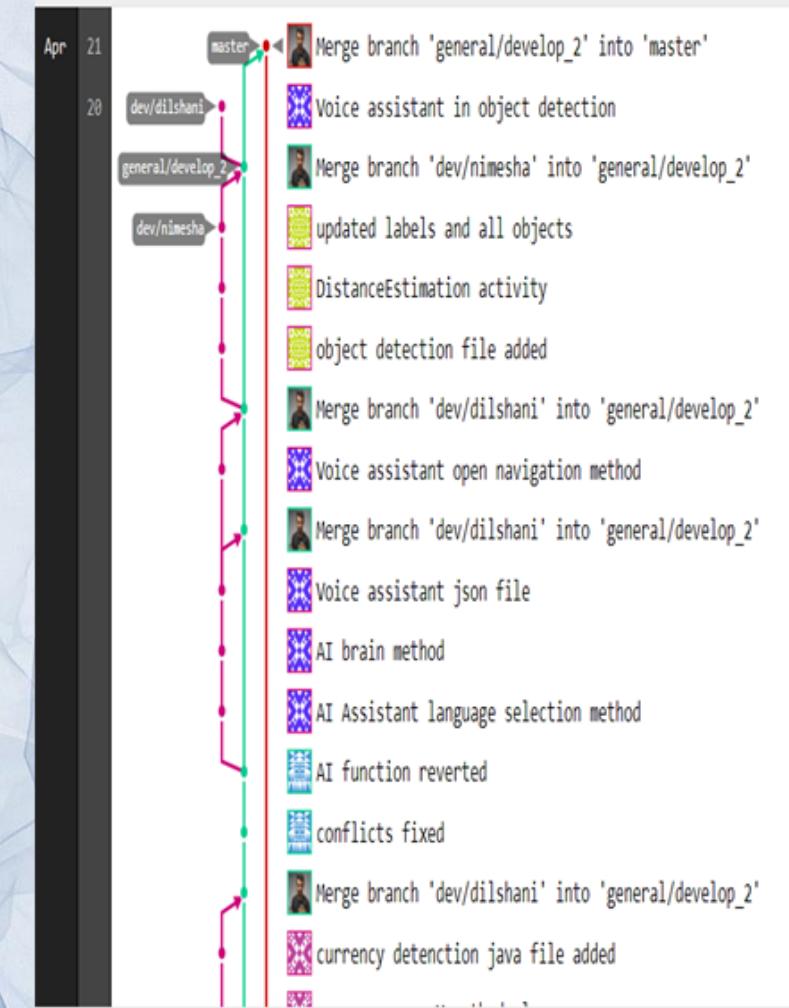
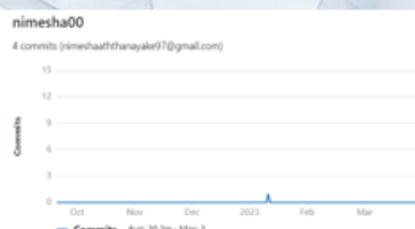
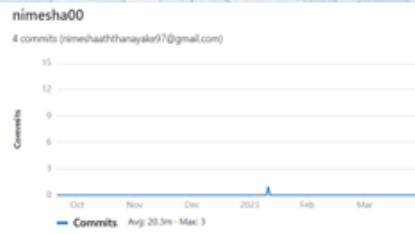
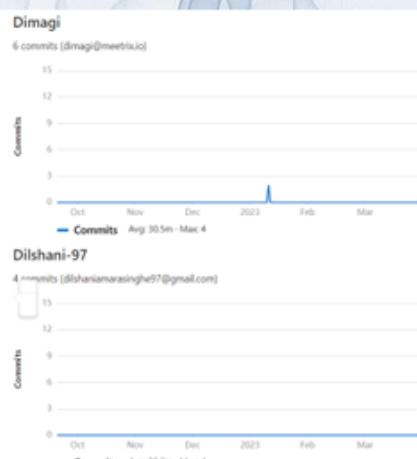
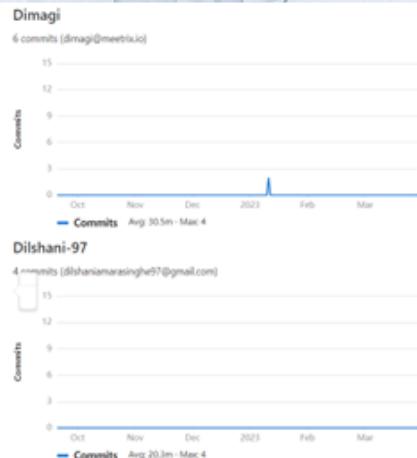
 Merge branch 'dev/dilshani' into 'general/develop_2' [...](#)
Chathuranga Costa authored 12 hours ago

 Voice assistant open navigation method
Dilshani-97 authored 12 hours ago

 Merge branch 'dev/dilshani' into 'general/develop_2' [...](#)
Chathuranga Costa authored 12 hours ago

 Voice assistant json file
Dilshani-97 authored 12 hours ago

 AI brain method
Dilshani-97 authored 13 hours ago



REFERENCES

- <https://www.educative.io/courses/build-your-own-chatbot-inpython/qAA5mKY19Gp>
- <https://towardsdatascience.com/reading-color-blindness-charts-deeplearning-and-computer-vision-a8c824dd71cd>
- <https://pyimagesearch.com/2017/09/11/object-detection-with-deeplearning-and-opencv/>
- <https://datascienceparichay.com/article/distance-between-two-pointspython/>
- <https://phdservices.org/currency-recognition-using-opencv-python/>

SPECIAL THANKS TO



**Sri Lanka Council for the Blind - Rajagiriya
R.M.H.P. Gunarathna (The Secretary)
including all the council members and the
staff members**

RESEARCH PANEL



Dr. Jayantha Amararachchi



Ms. Thamali Kelegama



Ms. Poorna Panduwawala



THANK YOU !!!